TOSHIBA

UM-TS02N**-E001

PROGRAMMABLE CONTROLLER

PROSEC T2N

USER'S MANUAL

- Basic Hardware and Function -

Contents

Toshiba Corporation

Important Information

Misuse of this equipment can result in property damage or human injury.
Because controlled system applications vary widely, you should satisfy yourself as to the acceptability of this equipment for your intended purpose.
In no event will Toshiba Corporation be responsible or liable for either indirect or consequential damage or injury that may result from the use of this equipment.

No patent liability is assumed by Toshiba Corporation with respect to use of information, illustrations, circuits, equipment or examples of application in this publication.

Toshiba Corporation reserves the right to make changes and improvements to this publication and/or related products at any time without notice. No obligation shall be incurred other than as noted in this publication.

PROSEC and TOSLINE are registered trademarks of TOSHIBA Corporation.

IBM and Windows are registered trademarks of International Business Machines Corporation.
MS-DOS and Windows are registered trademarks of Microsoft Corporation.
Ethernet is a registered trademark of Xerox Corporation.
DeviceNet is a trademark of the Open DeviceNet Vender Association, Inc.

Publication number: UM-TS02N∗∗-E001
1st edition December 1997

CE Marking

The Programmable Controller PROSEC T2N (hereafter called T2N) complies with the requirements of the EMC Directive 89/336/EEC and the Low Voltage Directive 72/23/EEC under the condition of use according to the instructions described in this manual.
The contents of the conformity are shown below.

Application of : **EMC -** 89/336/EEC(as amended by 91/263/EEC and 92/31/EEC)
Council Directive : **LVD -** 72/23/EEC

Manufacture's Name : TOSHIBA CORPORATION, FUCHU WORKS

Manufacture's address : 1, TOSHIBA-CHO,
FUCHU-SHI, TOKYO 183,
JAPAN

declares, that the product

Product Name : Programmable Controller , T2N Series

Model Number : TPU215N*S,TPU235N*S,TPU245N*S,TBU228N*S

Product Options : All

conforms to the following Product Specifications:

EMC	**: Radiated Interference**	EN55011 Class A Group 1
	Mains Interference	EN55011 Class A Group 1
	Radiated Susceptibility	ENV50140
	Conducted RFI Susceptibility	ENV50141, IEC1000-4-6
	Electrostatic Discharge	IEC1000-4-2
	Electrical Fast Transient	IEC1000-4-4

LVD : **EN61131-2 : 1995**

3,10 Dielectric Properties
4 Mechanical Requirements

Safety Precautions

This manual is prepared for users of Toshiba’s Programmable Controller T2N.
Read this manual thoroughly before using the T2N. Also, keep this manual and related manuals so that you can read them anytime while the T2N is in operation.

General Information

1. The T2N has been designed and manufactured for use in an industrial environment. However, the T2N is not intended to be used for systems which may endanger human life. Consult Toshiba if you intend to use the T2N for a special application, such as transportation machines, medical apparatus, aviation and space systems, nuclear controls, submarine systems, etc.

2. The T2N has been manufactured under strict quality control. However, to keep safety of overall automated system, fail-safe systems should be considered outside the T2N.

3. In installation, wiring, operation and maintenance of the T2N, it is assumed that the users have general knowledge of industrial electric control systems.
   If this product is handled or operated improperly, electrical shock, fire or damage to this product could result.

4. This manual has been written for users who are familiar with Programmable Controllers and industrial control equipment. Contact Toshiba if you have any questions about this manual.

5. Sample programs and circuits described in this manual are provided for explaining the operations and applications of the T2N. You should test completely if you use them as a part of your application system.

Hazard Classifications

In this manual, the following two hazard classifications are used to explain the safety precautions.

⚠ WARNING Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices.

Even a precaution is classified as CAUTION, it may cause serious results depending on the situation. Observe all the safety precautions described on this manual.

Safety Precautions

Safety Precautions

Installation:

⚠ CAUTION

1. Excess temperature, humidity, vibration, shocks, or dusty and corrosive gas environment can cause electrical shock, fire or malfunction. Install and use the T2N and related equipment in the environment described in this manual.

2. Improper installation directions or insufficient installation can cause fire or the units to drop. Install the T2N and related equipment in accordance with the instructions described in this manual.

3. Turn off power before installing or removing any units, modules, racks or terminal blocks. Failure to do so can cause electrical shock or damage to the T2N and related equipment.

4. Entering wire scraps or other foreign debris into to the T2N and related equipment can cause fire or malfunction. Pay attention to prevent entering them into the T2N and related equipment during installation and wiring.

5. Turn off power immediately if the T2N or related equipment is emitting smoke or odor. Operation under such situation can cause fire or electrical shock. Also unauthorized repairing will cause fire or serious accidents. Do not attempt to repair. Contact Toshiba for repairing.

6. The T2N must be installed in an enclosure. The user should consider to prevent contact with careless touch to the live parts of this product in during operation or maintenance.

7. The Protective ground terminal of the T2N must be connected to an external protective earth.

8. The computer which is connected to the T2N must be connected to an external protective earth properly.

9. The external cables, including for data transmission, which are prepared by the user are outside of the scope of this document.

Safety Precautions

Wiring:

⚠ CAUTION

1. Turn off power before wiring to minimize the risk of electrical shock.

2. Exposed conductive parts of wire can cause electrical shock. Use crimp-style terminals with insulating sheath or insulating tape to cover the conductive parts. Also close the terminal covers securely on the terminal blocks when wiring has been completed.

3. Operation without grounding may cause electrical shock or malfunction. Connect the ground terminal on the T2N to the system ground.

4. Applying excess power voltage to the T2N can cause explosion or fire. Apply power of the specified ratings described in the manual.

5. Improper wiring can cause fire, electrical shock or malfunction. Observe local regulations on wiring and grounding.

Operation:

⚠ WARNING

1. Configure emergency stop and safety interlocking circuits outside the T2N. Otherwise, malfunction of the T2N can cause injury or serious accidents.

⚠ CAUTION

2. Operate the T2N and the related modules with closing the terminal covers. Keep hands away from terminals while power on, to avoid the risk of electrical shock.

3. When you attempt to perform force outputs, RUN/HALT controls, etc. during operation, carefully check for safety.

4. Turn on power to the T2N before turning on power to the loads. Failure to do so may cause unexpected behavior of the loads.

5. Do not use any modules of the T2N for the purpose other than specified. This can cause electrical shock or injury.

6. Do not modify the T2N and related equipment in hardware nor software. This can cause fire, electrical shock or injury.

7. Configure the external circuit so that the external power required for output modules and power to the loads are switched on/off simultaneously.
   Also, turn off power to the loads before turning off power to the T2N.

Safety Precautions

Operation (continued):

⚠ CAUTION

8. Install fuses appropriate to the load current in the external circuits for the outputs. Failure to do so can cause fire in case of load over-current.

9. Check for proper connections on wires, connectors and modules. Insufficient contact can cause malfunction or damage to the T2N and related equipment.

Maintenance:

⚠ WARNING

1. Do not charge, disassemble, dispose in a fire nor short-circuit the batteries. It can be cause explosion or fire. Observe local regulations for disposal of them.

⚠ CAUTION

2. Turn off power before removing or replacing units, modules, terminal blocks or wires. Failure to do so can cause electrical shock or damage to the T2N and related equipment.

3. Replace a blown fuse with a specified one. Failure to do so can cause fire or damage to the T2N.

4. Do not insert your finger into the rack’s ventilation hole during power on. This can cause electrical shock.

5. Do not disassemble the T2N because there are hazardous voltage parts inside.

6. Perform daily checks, periodical checks and cleaning to maintain the system in normal condition and to prevent unnecessary troubles.

7. Check by referring “Troubleshooting” section of this manual when operating improperly. Contact Toshiba for repairing if the T2N or related equipment is failed. Toshiba will not guarantee proper operation nor safety for unauthorized repairing.

8. The contact reliability of the output relays will reduce if the switching exceeds the specified life. Replace the unit or module if exceeded.

9. Replace batteries in accordance with instructions described in the manual.
Failure to do so can cause system accidents.

Markings used on the T2N and in this manual

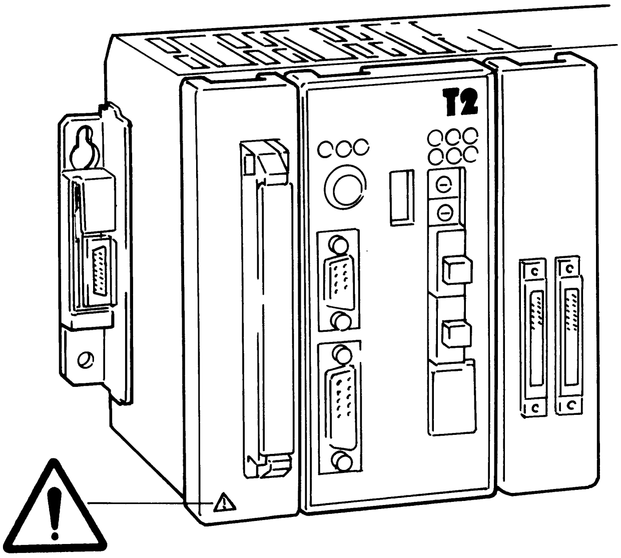
Warning Mark on the T2N

This is the warning mark for dangerous location. It is attached to the equipment in positions where there is a risk of electric shock and in positions where there is a risk damage to the equipment through wrong wiring.
Take the following precautions where there is this mark.

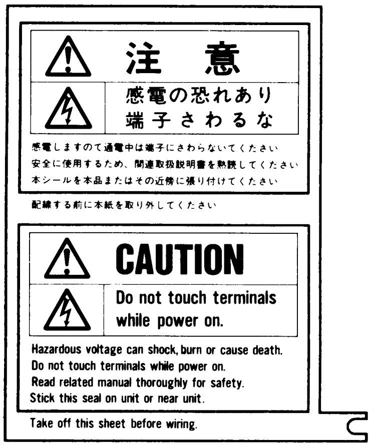
(1) Keep hands away from terminals ,especially the input terminall of power supply while power on, to avoid the risk of electrical shock.

(2) Turn off power before installing or removing modules, terminal blocks or wires.

(3) Applying excess power voltage to the T2N can cause explosion or fire. Apply power of the specified ratings described in this manual.



Safety Label



⚠ 注　意

⚠ 感電の恐れあり
端子さわるな

感電しますので通電中は端子にさわらないでください
安全に使用するため、関連取扱説明書を熟読してください
本シールを本品またはその近傍に張り付けてください

配線する前に本紙を取り外してください

⚠ CAUTION

⚠ Do not touch terminals while power on.

Hazardous voltage can shock, burn or cause death.
Do not touch terminals while power on.
Read related manual thoroughly for safety.
Stick this seal on unit or near unit.

Take off this sheet before wiring.

The safety label as shown on the left is attached to the power terminal of the T2N.

Remove the mount paper before wiring.

Peel off the label from the mount paper and stick it near the power terminals where it can be readily seen.

> NOTE
> ▼△▼
> This mark is printed in places in this manual which should always be read carefully.
> Read them carefully.

About This Manual

This manual has been prepared for first-time users of Toshiba’s Programmable Controller T2N to enable a full understanding of the configuration of the equipment, and to enable the user to obtain the maximum benefits of the equipment.

This manual introduces the T2N system configuration, and explains the specifications, installation and wiring for T2N’s basic hardware. This manual provides the information for designing T2N user program, such as T2N internal operation, memory configuration, I/O allocation. Information for maintenance and troubleshooting are also provided in this manual.

The specifications of the enhanced communication function, and how to use them, are explained in separate manual. Read T2E/T2N User’s Manual-Enhanced communication function. (UM-TS02E**-E003)
In addition, the T2N’s computer link function is also covered by separate manual. Read T-series Computer Link Operation Manual for details.

Related Manuals

The following related manuals are available for T2N. Besides this manual, read the following manuals for your better understanding.

T2N User’s Manual - Basic Hardware and Function	**- UM-TS02N**-E001**
T2E/T2N User’s Manual - Enhanced communication function	**- UM-TS02E**-E003**
TOSLINE-S20LP T2N/T3H Stations Instruction Manual	**-6F3B0356**
Built-in Ethernet Module for T2N (PU235N/245N) Instruction Manual	**-6F3B0362**
T-series Instruction Set	**- UM-TS03***-E004**
T-PDS for Windows Basic Operation Manual	**- UM-TS03***-E038**
T-PDS Basic Operation Manual (MS-DOS)	**- UM-TS03***-E006**
T-PDS Command Reference Manual (MS-DOS)	**- UM-TS03***-E007**
T-PDS Ver.2.0 Expanded Functions (MS-DOS)	**- UM-TS03***-E028**
T-Series Handy Programmer (HP911) Operation Manual	**- UM-TS03***-E025**
T-series Computer Link Operation Manual	**- UM-TS03***-E008**
1 Axis positioning controller Manual	**- UM-EX100**-E011**
T2 Communication Interface Module (CF211) Manual	**- UM-TS02***-E013**
T2/EX100 Computer Link Module (CL11) Manual	**- UG-TS02***-E015**
TOSLINE-S20 User’s Manual	**- UM-TLS20**-E001**
TOSLINE-F10 User’s Manual	**- UM-TLF10**-E001**

NOTE

Other than the listed above, some T2N elated manuals for special I/O modules and data transmission modules are available. Contact Toshiba for more information.

Terminology

The following is a list of abbreviations and acronyms used in this manual.

µs	microsecond
ASCII	American Standard Code For Information Interchange
AWG	American Wire Gage
BCC	Block Check Code
CCW	Counter-Clockwise
CPU	Central Processing Unit
CW	Clockwise
EEPROM	Electrically Erasable Programmable Read Only Memory
H	hexadecimal (when it appears in front of an alphanumeric string)
I/O	Input/Output
LED	Light Emitting Diode
LSB	Least Significant Bit
ms	millisecond
MSB	Most Significant Bit
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
Vac	AC voltage
Vdc	DC voltage

Contents

4. INSTALLATION AND WIRING

5. MAINTENANCE AND CHECKS

6. TROUBLESHOOTING

PART 2 FUNCTIONS

1. OVERVIEW

2. INTERNAL OPERATIONS

3. USER PROGRAM RUNNING CONTROL

4. PERIPHERAL MEMORY SUPPORT

5. RAS FUNCTION

PART 3 USER PROGRAMS

1. OVERVIEW

2. USER PROGRAM CONFIGURATION

3. USER DATA

4. I/O ALLOCATION

5. PROGRAM LANGUAGE

Features

Networking & Communication PLC   Easy to use high technology

There are three types of CPUs in the T2N.

PU215N : Standard type
PU235N : Standard type + built-in Ethernet
PU245N : Standard type + built-in Ethernet + built-in TOSLINE-S20LP

■Networking

●Ethernet connection

The T2N CPU modules (PU235N and PU245N) have built-in Ethernet interface (10Base-T). Through the Ethernet, the T2N can communicate with higher level controllers (computer, workstation, etc.) or other PLCs including T2N.

●TOSLINE-S20LP connection

The T2N CPU module (PU245N) has built-in TOSLINE-S20LP (Loop version) in the CPU module. TOSLINE-S20LP is a high-reliability double-loop fiber optic network.

●High speed industrial LAN

The T2N can be connected to Toshiba’s high speed industrial LANs (Local Area Networks) TOSLINE-S20 and TOSLINE-F10. The TOSLINE series are suited for real time control data linkage. Through these networks, the T2N can exchange data with Toshiba’s various equipment, such as, DCS system, other T-series PLCs, Inverters, etc.

●DeviceNet support

A DeviceNet scanner module is available for the T2N. The DeviceNet scanner module can read/write data to any other manufacturer’s OVDA certified devices (I/O blocks, Inverters to include Toshiba’s G3, air manifold, sensors, etc.).

■Enhanced communication

The T2N has RS-232C/RS485 serial communication port. Either interface can be selected. Using this port, one of the following communication functions can be used.

●Computer link mode	:Connection with higher level computer, MMI/SCADA system, modem, etc.
●Data link mode	:Easy data linkage between two T2Ns or T2E .
●Free ASCII mode	:Active communication between serial ASCII devices. (bar code readers, etc.).

■Programmer port function

The T2N’s RS-232C programmer port supports the T-series computer link protocol. This results in easy connection to a higher level computer, an operator interface unit, a modem, etc.

■ High speed processing

The T2N excels at applications where high speed processing is required.

- 0.33μs/contact • 0.44μs/coil
- 1.2μs/16-bit transfer• 1.63μs/16-bit addition

■ Advanced instruction set

The T2N offers 24 basic relay-ladder instructions and 192 function block instructions, including the following.

- Arithmetic operation
- PID/ramp/integral
- Averaging/filtering
- Data manipulations
- Subroutine call
- ASCII↔Hex conversion
- Trigonometric functions
- For-Next loop
- Floating-point math

■ Two programming Languages

The T2N supports two programming languages: Ladder Diagram (LD) and Sequential Function Chart (SFC). By selecting the appropriate language, or combination of the two, program development time can be greatly reduced.

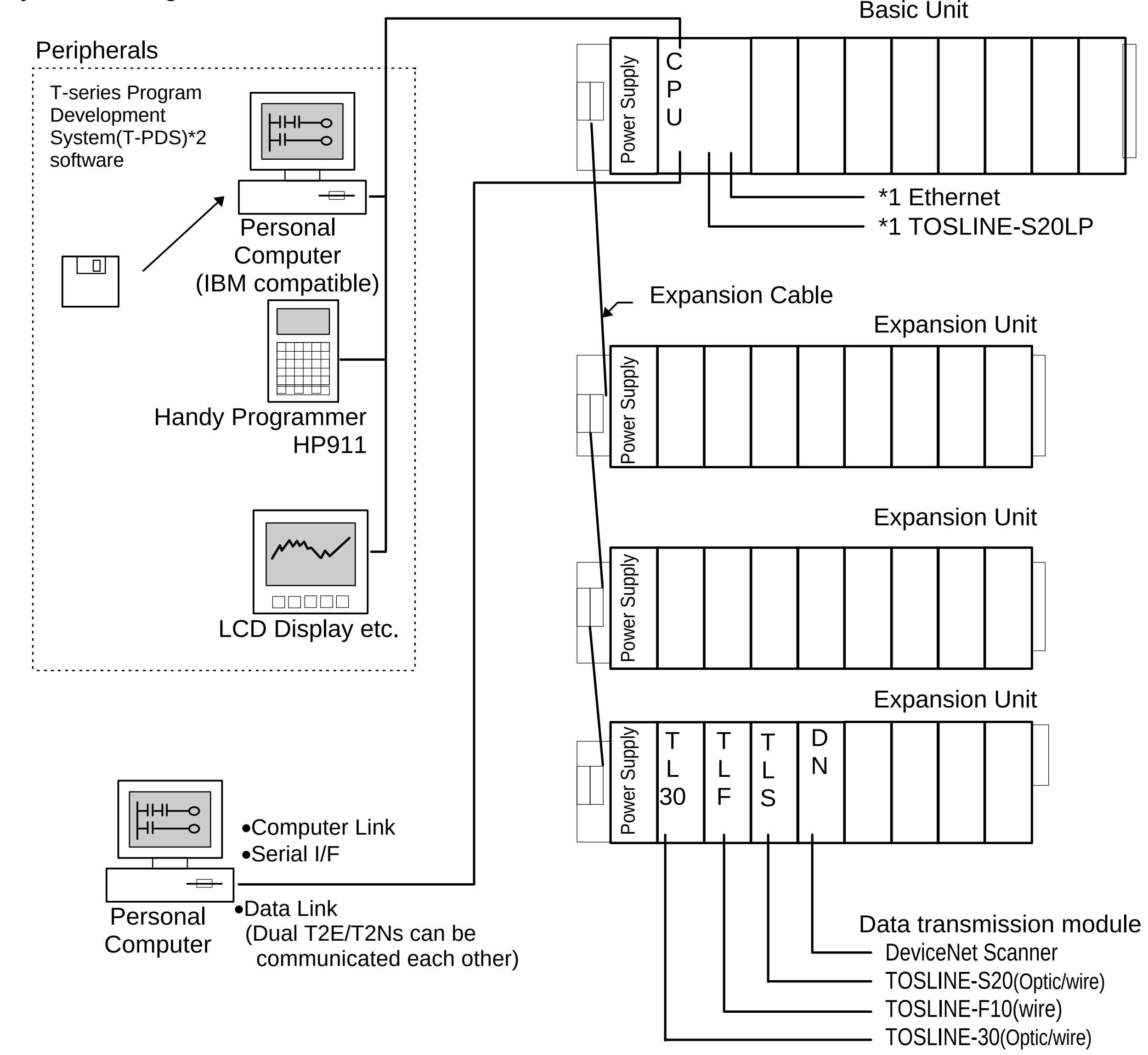
■ Sufficient capacity

The T2N has 24K steps program memory and controls up to 2,048 I/O points.

PART1

HARDWARE

1.1 System Configuration



Up to a maximum of 3 expansion units can be connected.

Note *1:	Ethernet	which is built-in PU235N and PU245N
	TOSLINE-S20LP	which is built-in PU245N

*2: T-PDS (MS-DOS) software V2.06 or later is available for the T2N.
T-PDS (Windows) software V1.2 or later is available for the T2N.

•Basic Configuration

Rack	BU228N
	BU268
	BU266
Power Supply Module	PS261
	PS31
CPU Module	PU215N
	PU235N
	PU245N
Expansion Cable	CAR3
	CAR5
	CAR7
	CS2RF

•I/O module

DC input	DI31
	DI32
	DI235
AC input	IN51
	IN61
Transistor output	DO31
	DO32
	DO235
	DO233P
Triac output	AC61
Relay output	RO61
	RO62
Analog input	AI21
	AI31
	AI22
	AI32
Analog output	AO31
	AO22
	AO32
Pulse input	PI21
Positioning module	MC11
Serial interface	CF211

•Data transmission Module

DeviceNet scanner	DN211
TOSLINE-S20	SN221
	SN222A
TOSLINE-F10	MS211
	RS211
TOSLINE-30	LK11
	LK12

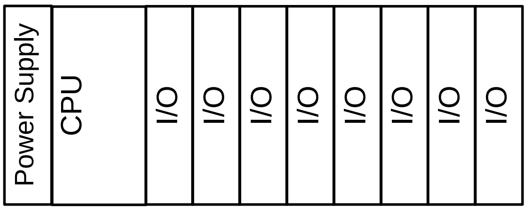
•Programming Tool

T-PDS (for MS-DOS)	MM33I1
T-PDS (for Windows)	MW33E1
Handy Programmer	HP911

Minimum and Maximum configuration are shown on next page.
As mentioned in Section 1.5 ,the following racks are available.

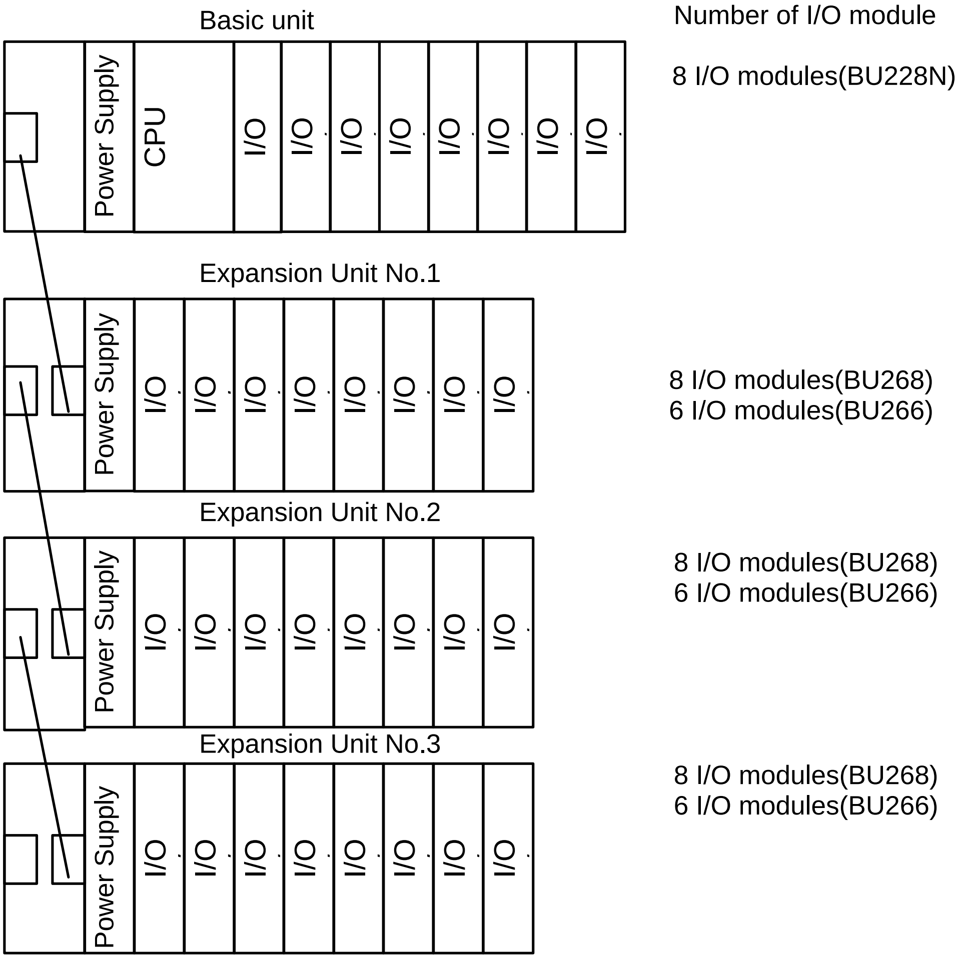
Part Number	Application	Number of Module Installation	Remarks
TBU228N*S	Dedicated to the Basic unit	8	
TBU268**S	For expansion unit	8	
TBU266**S	For expansion unit	6	

1)Minimum Configuration



Basic unit with 8 I/O modules(BU228N)

2)Maximum Configuration

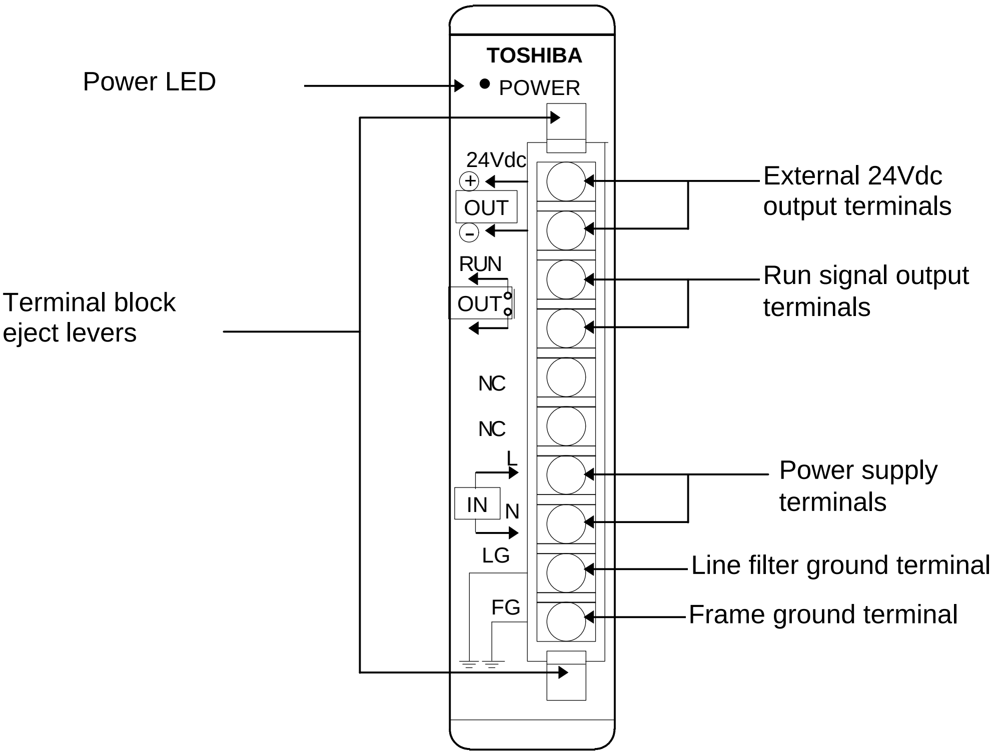


- Up to a maximum of 3 expansion units can be connected.
- There is no limit on combinations of the types of the rack.
- When one BU228N and three BU268 are used ,a maximum of 32 I/O modules can be controlled by the T2N CPU.

1.2 Power Supply Module

Power supply modules are mounted on the left -end slots of all units. There are two types according to the power voltage. Select one as required.

Model	Power Voltage	Output Rating	Weight
TPS261**S	100 - 240Vac (+10%/-15%)	Internal 5V power supply : 2.5A (max.) External power supply : 24V,+10%/-10% 0.5A (max.) (Internal + external total 15W or less)	Approx. 300g
EX10*MPS31	24Vdc (+20%/-15%)		


TOSHIBA
POWER
Power LED
24Vdc
+
OUT
-
RUN
OUT
NC
NC
L
IN
N
LG
FG
Terminal block eject levers
External 24Vdc output terminals
Run signal output terminals
Power supply terminals
Line filter ground terminal
Frame ground terminal

- External 24Vdc Output Terminals
  These terminals can be used to supply 24Vdc power to external devices such as sensors or relay output modules. 24Vdc(±10%)-0.5A(max.)
- Run Signal Output Terminals
  When the T2N is in the operating mode(RUN), built-in contact is closed.
  240Vac(+10%)/24Vdc(+20%)-2A(max.)
  (Can also be used on expansion units)
- Power Supply Terminals
  Connect to the power supply line. (See 4.7 Wiring the power supply).
- Line Filter Ground Terminal / Frame Ground Terminal
  These are grounding terminals. (See 4.6 Grounding methods).

NOTE

▼△▼

1. The maximum rated output of the power supply module is 15W, this includes the internal 5Vdc and external 24Vdc output combined. Configure the system, referring to the Module Current Consumption Table on the next page, so that the following equation is satisfied.
   15W≥5V × Total 5V current (max. 2.5A) + 24V × external 24V current (max. 0.5V)

2. Do not connect the external 24V supply terminals to the other power supply systems, and do not run the wiring over long distances.

3. This power supply module is dedicated power supply for the T2N and T2E/T2/EX100. Do not use it by itself for other purposes.

Module Current Consumption Table

Name	Model	Internal 5Vdc	External 24Vdc	Weight (approx.)
CPU	PU215N	800mA or less	-	300g
CPU(Ether)	PU235N	1500mA or less	-	400g
CPU(Ether+TOSLINE-S20LP)	PU245N	2000mA or less	-	400g
Rack	BU228N	50mA or less	-	1600g
	BU268	50mA or less	-	1500g
	BU266	50mA or less	-	1400g
16-point DC/AC input(12-24V)	DI31	15mA or less	-	200g
32-point DC input(24V)	DI32	80mA or less	-	200g
64-point DC input(24V)	DI235	100mA or less	-	250g
16-point AC input(100-120V)	IN51	15mA or less	-	250g
16-point AC input(200-240V)	IN61	15mA or less	-	250g
12-point relay output	RO61	50mA or less	DC24V, 140mA	250g
8-point isolated relay output	RO62	40mA or less	DC24V, 100mA	250g
16-point transistor output	DO31	60mA or less	DC5-24V, 35mA	200g
32-point transistor output	DO32	250mA or less	DC5-24V, 100mA	200g
64-point transistor output	DO235	250mA or less	-	250g
16-point transistor (PNP)	DO233P	60mA or less	-	200g
12-point triac output	AC61	300mA or less	-	200g
4ch analog input (8bit) (4-20mA/1-5V)	AI21	50mA or less	DC12/24V, 50mA	200g
4ch analog input (12bit) (4-20mA/1-5V)	AI22	50mA or less	DC24V, 50mA	200g
4ch analog input (8bit) (0-10V)	AI31	50mA or less	DC12/24V, 50mA	200g
4ch analog input (12bit) (±10V)	AI32	50mA or less	DC24V, 50mA	200g
2ch analog output (8bit) (4-20mA/1-5V/0-10V)	AO31	70mA or less	DC24V, 90mA	200g
2ch analog output (12bit) (4-20mA/1-5V)	AO22	170mA or less	DC24V, 90mA	200g
2ch analog output (12bit) (±10V)	AO32	170mA or less	DC24V, 90mA	200g
1ch pulse input	PI21	80mA or less	-	200g
Position control	MC11	200mA or less	DC12/24V, 100mA	250g
Serial Interface	CF211	550mA or less	-	200g
TOSLINE-30(wire)	LK11	250mA or less	-	200g
TOSLINE-30(optical)	LK12	200mA or less	-	200g
TOSLINE-S20(wire)	SN221	600mA or less	-	250g
TOSLINE-S20(optical)	SN222A	700mA or less	-	250g
TOSLINE-F10(Master Station)	MS211	600mA or less	-	250g
TOSLINE-F10(Remote Station)	RS211	600mA or less	-	250g
Devicenet scanner	DN211	500mA or less	-	200g

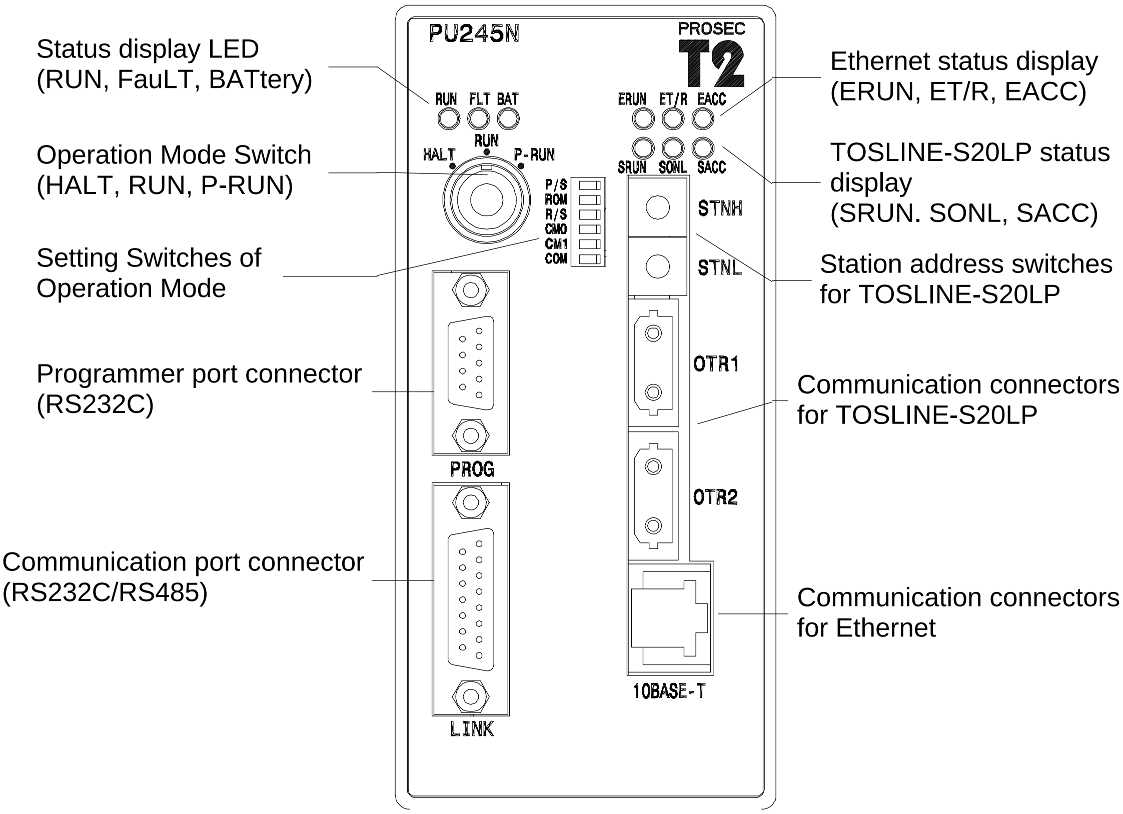
NOTE
▼△▼

The external 24Vdc in the Table are not power supplies for input/output signals. They are the power supplies required for module operation.

1.3 CPU Module

There are three types of CPU modules with functions as shown below.

Type	Specification
PU215N	RAM(battery back-up) + EEPROM, User program 23.5k step, ladder, SFC, real time clock, enhanced communication function
PU235N	PU215's function + Ethernet
PU245N	PU215's function + Ethernet + TOSLINE-S20LP



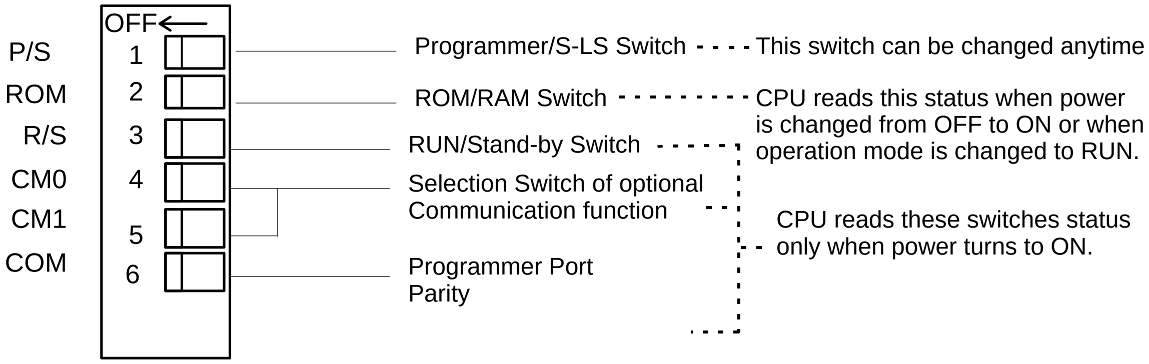
Status display LEDs : Show operation states of the T2N

RUN (Green)	Lit	Operating state (RUN Mode)
	Blink	HOLD Mode
	Out	Stopped state (HALT Mode) or Error Mode
FAULT (Red)	Lit	CPU abnormal
	Blink	Program abnormal
	Out	Normal
BAT (Green)	Lit	Battery normal
	Blink	Battery getting abnormal
	Out	Battery abnormal or no battery

Status display LEDs : Show operation states of the Ethernet/TOSLINE-S20LP
For details of operation mode, see Ethernet/TOSLINE-S20LP user's manual.
These all LED's color is green.

ERUN	Operating state (Ethernet)
ET/R	Data transmission/receive state (Ethernet)
EACC	T2N's accessing state for Ethernet
SRUN	Operating state (TOSLINE-S20LP)
SONL	Online state of TOSLINE-S20LP
SACC	T2N's accessing state for TOSLINE-S20LP

- Setting Switches of Operation Mode
  These switches are provided on the CPU front panel. They control the following functions.


OFF
P/S 1
ROM 2
R/S 3
CM0 4
CM1 5
COM 6
Programmer/S-LS Switch
This switch can be changed anytime
ROM/RAM Switch
CPU reads this status when power is changed from OFF to ON or when operation mode is changed to RUN.
RUN/Stand-by Switch
Selection Switch of optional Communication function
CPU reads these switches status only when power turns to ON.
Programmer Port Parity

(1) Programmer/S-LS(selection) Switch

Setting Position SW.1	Function	CPU operation (Communication support on Programmer port)
OFF	T-PDS	T-PDS direct connection mode : Operate as the Programmer.
ON	S-LS	S-LS connection mode : Operate as the tool for setting parameter for TL-S20LP.Or operate as the remote programmer T-PDS via TL-S20LP.

NOTE
▼△▼

Programmer/S-LS (selection) switch is set to Programmer at the factory.

(2) ROM/RAM Switch

Setting Position SW.2	Function	CPU operation at power up and at the beginning of the RUN mode
OFF	ROM	Starts up after the content of the EEPROM has been transferred to the RAM. (Transfer is not executed when P-RUN is selected.)
ON	RAM	Starts up on the content of the RAM. (No program transfer)

NOTE
▼△▼

The ROM/RAM switch is set to ROM at the factory.

(3) RUN/Stand-by Switch

Setting Position SW.3	Function	Operation Mode Switch	Mode after power up	Remarks
OFF	Automatic RUN	HALT	HALT	
		RUN	RUN	Automatic RUN start occurs.
ON	Stand-by	HALT	HALT	
		RUN		Starts up in HALT mode. Ready to start operation by RUN command from the programmer or by shifting the operation mode switch.(→HALT→RUN)

NOTE
▼△▼

The RUN/Stand-by switch is set to RUN at the factory.

(4)(5)Selection Switch of optional Communication function

Setting Position		Function	Operation
SW.4	SW.5		
OFF	OFF	Computer Link	The T2N can communicate with a master computer using T-series computer link protocol.
ON	OFF	Data Link	The T2N executes data link with other T2N.
OFF	ON	Free ASCII	The T2N can communicate with external devices using ASCII code.
ON	ON	Reserved	No operation.

NOTE
▼△▼

1. These switches are set to computer link function at the factory.
2. For details of the operation mode, see Section 1.4.

(6)Programmer Port Parity

Setting Position SW.6	Function	Remarks
OFF	Odd Parity	8 bit Data, 9600bps, Data length is 11bit.
ON	No Parity	8 bit Data, 9600bps, Data length is 10bit.

NOTE
▼△▼

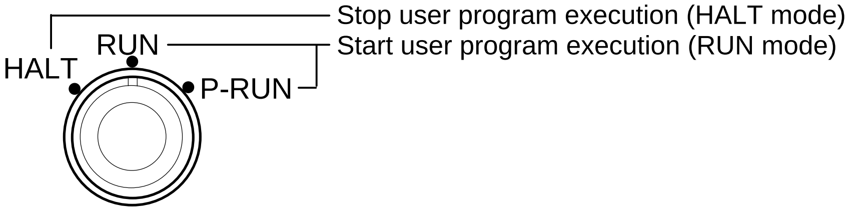
The Programmer Port Parity switch is set to Odd Parity at the factory.

The T2N can connect to Modem by using this switch.
Control signals (CTS,DTR,etc) should be set to No Use at the modem side.

Response of the T2N can be delayed on the programmer port using SW38 (Programmer port response delay mode register).

- Operation Mode Switch

  The Operation Mode Switch is provided on the CPU module.
  This switch controls T2N operation (RUN/P-RUN/HALT).


Stop user program execution (HALT mode)
RUN
Start user program execution (RUN mode)
HALT
P-RUN

CPU status is shown below after power up or after the operation mode is changed to RUN from HALT.

Setting Position	User Program	Operation Mode	Initial Load Program exe : executed - :not executed		Memory Protection	Operation Mode Change by the programmer
HALT	Stopped	HALT	SW.2:OFF	exe	No protect	not available
			SW.2:ON	-		
RUN	Executed	RUN	SW.2:OFF	exe	No protect	available
			SW.2:ON	-		
P-RUN			SW.2:OFF	-	protect	
			SW.2:ON			

As shown the above table, initial load (program transfer into EEPROM from RAM) performs in the RUN mode when setting SW.2 to OFF except P-RUN position.

NOTE

▼△▼

1. The operation mode switch is set to HALT at the factory.
2. “P-RUN” is state that the operation mode switch set to P-RUN.
   The user program and the first half of data register (D0000 to D2047) are in the write protect mode and user can’t write or change them.
3. Normally, the programming is carried out in the HALT mode.
4. When shifting to the RUN mode with the ROM/RAM switch in the ROM position , operation will commerce after program transfer has been executed. (that is, it is called initial load.)
5. For details of the operation mode, see Part2, Section 2.3.
6. When the operation mode switch is changed quickly from HALT to P-RUN, program transfer is interrupted.
   Turn to P-RUN after the RUN LED is lit.
7. The RAM is backed up by battery of the T2N. When the battery voltage drops and the T2N can’t keep retentive area in the RAM, CPU clears all the retentive data.
   Then CPU checks user program BCC. If error is detected, CPU registered error.

- Programmer Port
  The programmer (T-PDS or HP911) is connected to this programmer port. Connector type of CPU side is female , 9-pin D-SUB connector.

  The T2N's RS232C programmer port can accept the computer link protocol (data read/write). This results in easy connection to a higher level computer, an operator interface unit, etc. directly.
  General specifications and the connector pin assignment of programmer port are shown below.
  For details of T-series computer link protocol, see T-series User's manual - Computer Link (UM-TS03***-E008).

Communication specifications

Item	Specifications
Interface	Conforms to RS232C
Configuration	One to One
Transmission distance	15m max.
Transmission speed	9600bps (fixed)
Frame format	Start bit 1bit Data 8bit Parity odd/none (selected by SW.6) stop bit 1bit
Supported command	DR (Data Read) DW (Data Write) ST (Status read)

Pin assignment of programmer port

Signals	No. of pins	Symbols	Direction
Transmission data	3	TXD	T2N → Host
Receive data	2	RXD	T2N ← Host
Signal ground	5	SG	T2N -- Host
Request To Send	7	RTS	T2N → Host
Clear To Send	8	CTS	T2N ← Host

NOTE
▼△▼

Other pins except the above table should not be connected.

1.4 Communication port

The T2N supports enhanced communication functions. Either RS485 or RS232C can be selected for this communication port by using selection switches.

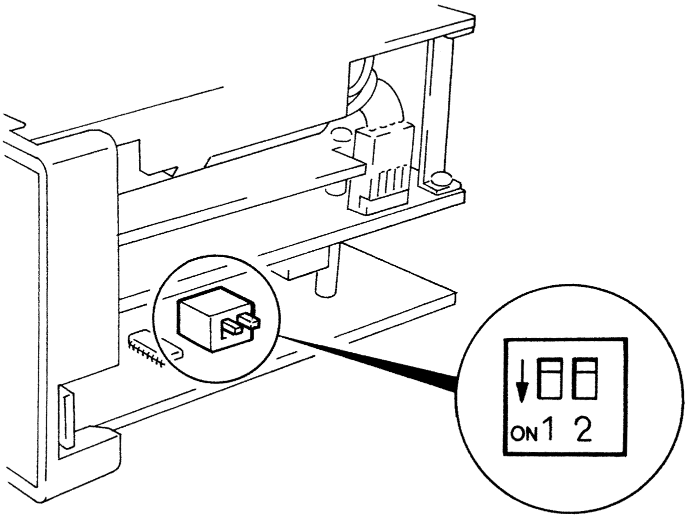
There are three communication functions in the T2N.
One of them can be selected by setting swithches CM0/CM1 (SW.4/SW.5).

Function	Operation
Computer Link	This performs to connect between a Host computer and up to 32 T2Ns, using RS485. (one to one ,using RS232C)
Data Link	This performs to connect two T2E/T2Ns. They share 32W data with each other.
Free ASCII	This performs to connect between the external devices such as inverter, etc. and the T2N.

For details of these functions and usage method, see T2E/T2N User's manual - Enhanced communication function -(UM-TS02E**-E003).

- Switch settinf of communication port.

Setting Position Dip-SW.1	Communication port interface.
OFF	Conforms to RS485.
ON	Conforms to RS232C.

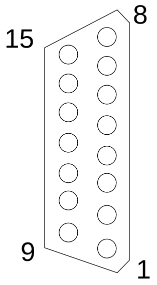


NOTE
▼△▼
These swithes are set to RS485 at the factory.

- Pin assignment of communication port (LINK).

Signals	No. of pins	Symbols	Interface	Direction
Transmission data	5	TXD	RS232C	T2N→Host
Receive data	12	RXD	RS232C	T2N←Host
Request To Send	6	RTS	RS232C	T2N→Host
Clear To Send	14	CTS	RS232C	T2N←Host
Signal Ground	7,8,15	SG	RS232C/485	T2N - Host
Transmission data A	3	TXA	RS485	T2N→Host
Transmission data B	11	TXB	RS485	T2N→Host
Receive data A	2	RXA	RS485	T2N←Host
Receive data B	10	RXB	RS485	T2N←Host
No Connection	1,4,9,13	NC		

D-SUB 15pin connector.

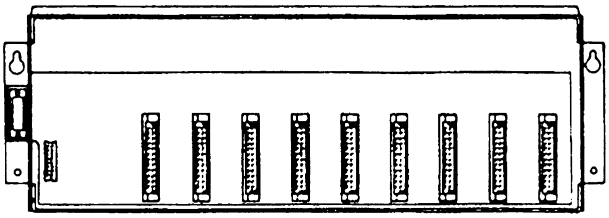


1.5 Racks

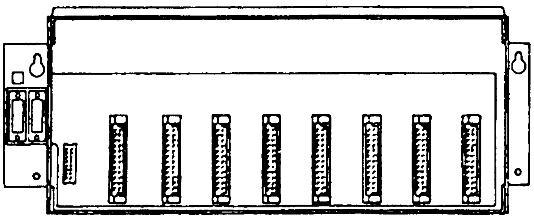
As mentioned in Section 1.1, the rack is available in the three types as follows. The BU228N is for the basic unit, and the BU268/BU266 are for the expansion unit.

Type	Application	Number of Module Installation	Remarks
BU228N	Dedicated to the Basic unit	8	
BU268	For expansion unit	8	
BU266	For expansion unit	6	

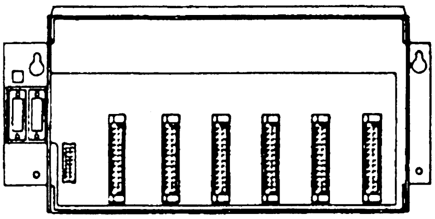
BU228N



BU268



BU266



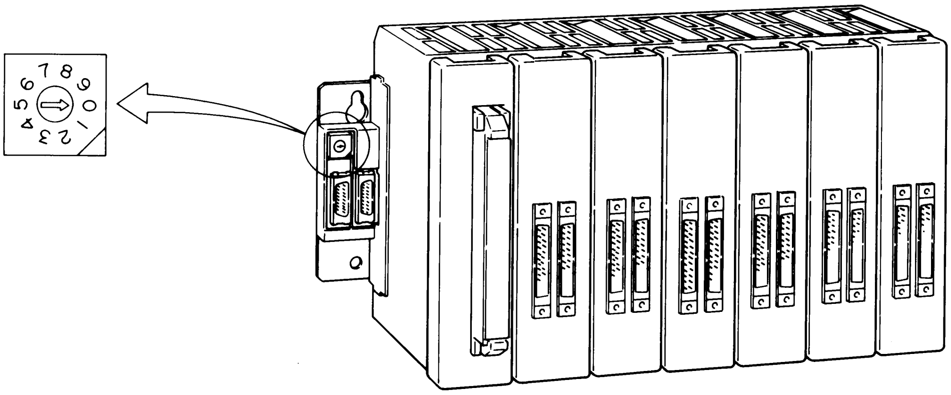
NOTE ▼△▼

1. Two expansion connectors are provided in the BU268 and the BU266. The right side connector is for connecting the basic unit/previous expansion unit.
   The left side connector is for connecting the next expansion unit.

2. Use a blind slot cover (EX10*ABP1) to prevent debris from entering in the rack where no I/O modules is mounted .

- Setting the Unit No.

When using the BU268 or the BU266 for dedicated expansion units, set the Unit No. before operating. The setting is carried out by a rotary switch in the upper part of the expansion connector on the left hand side of the rack.



The rack used for :	Switch Setting
Expansion Units	Set in the order 1>2>3, starting from the unit closest to the basic unit

NOTE
▼△▼

1.Switches will be set at 0 at the factory.
2.Be careful not to duplicate Unit Nos. on units.
3.Do not use setting 4 - 9, as these are not for use.

1.6 Expansion Cables

These are used for connecting the basic unit and the expansion units. They are available in the following four lengths.

Type	Length
CAR3	30cm
CAR5	50cm
CAR7	70cm
CS2RF	1.5m

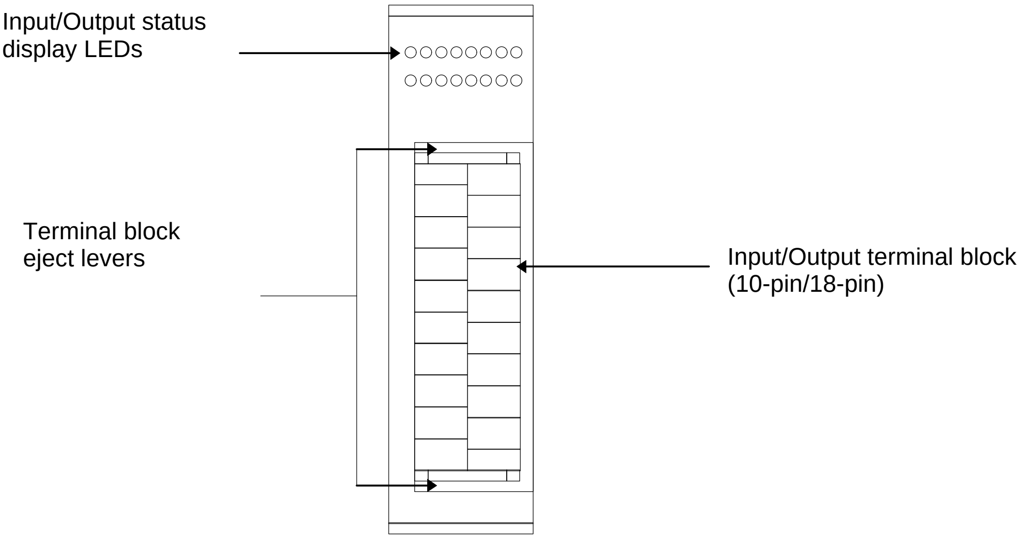
NOTE
▼△▼

The maximum cable length between units is 1.5m.
The maximum total cable length is 4.5m.

1.7 I/O Modules

Various types of I/O modules are available for the T2N, as shown in the following table. Thus, it can respond to a wide variety of applications.

I/O modules can be mounted in any slot in the rack, and in any order. (See Section 4.8 for recommended arrangements)


Input/Output status display LEDs
Terminal block eject levers
Input/Output terminal block (10-pin/18-pin)

Type	Description	Specification
DI31	DC/AC input	16-point (16 points per common),12-24V dc/ac
DI32	DC input	32-point (8 points per common),24Vdc
DI235	DC input	64-point (8 points per common),24Vdc
IN51	AC input	16-point (16 points per common),100-120Vac
IN61		16-point (16 points per common),200-240Vac
RO61	Relay output	12-point (4 points per common), 240Vac(+10%)/DC24V(+20%), 2A/point,4A/4 points common (max.)
RO62		8-point (each point isolated), 240Vac(+10%)/DC24V(+20%), 2A/point (max.)
DO31	Transistor output	16-point (16 points per common),5-24Vdc 1A/point, 1.2A/4 points (max.)
DO32		32-point (8 points per common),5-24Vdc 0.1A/point, 0.8A/8 points common (max.)
DO235		64-point (8 points per common),5-24Vdc 0.1A/point, 0.8A/8 points common (max.)
DO233P		16-point (16 points per common),12-24Vdc
AC61	Triac output	12-point (4 points per common),100-240Vac 0.5A/point, 0.6A/2-element SSR (max.)
AI21	Analog input (8bit)	4-channel (not isolated between channels), 1-5V/4-20mA, 8bit resolution
AI31		4-channel (not isolated between channels), 0-10V ,8bit resolution
AI22	Analog input (12bit)	4-channel (not isolated between channels), 1-5V/4-20mA, 12bit resolution
AI32		4-channel (not isolated between channels), -10V - +10V ,12bit resolution

Type	Description	Specification
AO31	Analog output (8bit)	2-channel (not isolated between channels), 1-5V/0-10V/4-20mA, 8bit resolution
AO22	Analog output (12bit)	4-channel (not isolated between channels), 4-20mA /1-5V,12bit resolution
AO32		4-channel (not isolated between channels), -10V - +10V,12bit resolution
PI21	Pulse input	1-channel (two phase, with zero marker), 5/12Vdc, 100kpps (max), 24bit counter
MC11	Single-axis positioning	1 axis, 100kpps(max.), position data memory capacity 64 points
CF211	Serial Interface	RS-232C 1port, Common memory 160W×2

NOTE
▼△▼

For detailed specifications of each I/O module, see Section 2.3 I/O Module Specifications.

1.8 Data Transmission Module

By applying the following 4 types of data transmission module according to the system requirements, the T2N can configure the flexible and efficient control systems.

- TOSLINE-F10

PLC to PLC data linkage and remote I/O systems are configured by the TOSLINE-F10 data transmission equipment.

Up to 8 T2 stations can be mounted in any slots, in the same way as I/O modules.

	MS211/RS211 (High-speed setting)	MS211/RS211 (Long-distance setting)
Topology	Bus (twisted-pair cable)	
Transmission speed	750kbps	250kbps
Transmission Distance	500m	1km
Number of stations	max. 32stations	
Transmission capacity	32 words (L/LW)	
Response speed	7ms(when 32 words)	12ms(when 32 words)

- TOSLINE-S20

The TOSLINE-S20 is a Local Area Network (LAN) for factory automation systems. It can achieve high-speed data linkage between PLCs and communication between industrial computers.
One T2 station can be mounted in any slot, in the same way as an I/O module.
SN221 and SN222A are allocated as optional card in the T2N.
Therefore READ/WRITE instructions should be used to access data of TOSLINE-S20 modules, because the link registers are not assigned.

	SN221 (Co-Axial Cable)	SN222A (Optical Fiber Cable)
Topology	Bus	
Transmission speed	2 Mbps	
Transmission Distance	1km	10km (1km between stations)
Number of stations	Max. 64 stations	
Response speed	Minimum 5ms in scan transmission	

- TOSLINE-30

The T2N can use the TOSLINE-30.
It is effective when connecting EX series systems to the T2N.
Up to 4 T2 stations can be mounted in any slots, in the same way as I/O modules.
The link relay/register(Z/W) is assigned for the TOSLINE-30, the same as the TOSLINE-S20LP. If the TOSLINE-S20LP is used together with the TOSLINE-30, the link registers assigned to the TOSLINE-30 (starting with W0000) should not allocate for the TOSLINE-S20LP.

	LK11 (Twisted-pair Cable)	LK12 (Optical Fiber Cable)
Topology	Bus	Star
Transmission speed	187.5kbps	375kbps
Transmission Distance	1km	2km (1km between stations)
Number of stations	MAX. 17 stations	MAX. 16 stations
Transmission capacity	8/16/32 words W0000-W0127	
Response speed	25ms(when 32words)	19.2ms(when 32words)

- DeviceNet (Scanner)

The DeviceNet is a field network. It can achieve data linkage between the T2N and any ODVA certified devices.
The T2N can use the DeviceNet scanner module.
It is effective for connecting DeviceNet systems to the T2N.

The T2N stations can be mounted in any slots, in the same way as for I/O modules.

	DN211 (Twisted-pair Cable)		
Topology	Bus		
Transmission speed	125kHz	250kHz	500kHz
Transmission Distance	500m	250m	100m
Number of stations	Max. 64 stations		

1.9 Built-in networking

The T2N has built-in networking as follows.

Type	TOSLINE-S20LP	Ethernet
PU215N	-	-
PU235N	-	Built-in
PU245N	Built-in	Built-in

For details of networking, see the following manuals.

Ethernet	Built-in Ethernet Module for T2N (PU235N/245N) Instruction Manual	6F3B0362
TOSLINE-S20LP	TOSLINE-S20LP T2N / T3H Stations Instruction Manual	6F3B0356

- Ethernet

The Ethernet is a standard Local Area Network (LAN) not only for OA but also for factory automation systems.

It can achieve high-speed data linkage between PLCs and communication with higher level computers (workstation/personal computer). The T2N supports T-series computer link function, PLC data link function and socket interface function.
The T2N has 10BASE-T interface for the Ethernet.

		Specifications	
Topology		Bus	
Transmission speed		10Mbps	
Transmission Distance	between nodes	200m(one HUB)	1700m(four HUBs)
	between segments	100m	
Number of nodes		Max. 1 node/1 segment	
Communication Function		(1) T-series computer link	
		(2) T-series PLC link	
		(3) Socket interface	

- TOSLINE-S20LP

The TOSLINE-S20LP is a Local Area Network (LAN).
It can achieve high-speed data linkage between PLCs.
The TOSLINE-S20LP is a double-loop system using optical fibers and can continue data linkage if any problem is occured in one loop.
Transmission data capacity is expanded to 4k words.

		Specifications
Topology		Double-Loop
Transmission speed		2Mbps
Transmission Distance	between stations	1km (4km using the optical repeater)
	total	30km
Number of stations		Max. 64 stations/loop
Transmission capacity		4,096words W0000-W2047 (Z/W) and any 2048word registers (using XFER instruction)
Response speed		Min. 5ms in scan transmission
Communication Function		(1) Scan transmission
		(2) Message transmission

2.1 General Specification

Item		Specification	Remarks
Power Supply	Rated Voltage	(1)100-240Vac	PS261
		(2)24Vdc	PS31
	Voltage Fluctuation Range	(1)85 - 264Vac	PS261
		(2)20.4 - 28.8Vdc	PS31
	Power Supply Frequency	(1)50/60Hz	
	Frequency Fluctuation Range	(1)47 - 63Hz	
	Retentive power interruption	10ms or less (at maximum load for one power supply module)	
	Power consumption	(1)53VA or less	PS261
		(2)22W or less	PS31
	Inrush current	(1)15A(at 100Vac) or less 35A(at 240Vac) or less	PS261
		(2)30A/10ms or less	PS31
Insulation resistance		10MΩor more (between power terminals and ground terminals)	
Withstand voltage		1500Vac - 1minute	*1
Ambient temperature		0 to 55°C	operation
		-20 to 75°C	storage
Ambient humidity		20~90%RH no condensation	
Atmosphere		No corrosive gases Sulphurous acid gas 0.05ppm or less Hydrogen sulphide 0.01ppm or less	
Dust		10mg/m^3 or less	
Vibration immunity		16.7Hz-3mm p-p (3 mutually perpendicular awes)	
Shock immunity		98m/s^2 (10g) (3 shocks per axis, on 3 mutually perpendicular awes)	
Noise immunity		1000Vp-p /1µs Complied for EMC Directive of CE marking	
Grounding		Grounding resistance 100Ω or less	
Construction		Installed in control panel	
Cooling		Natural air cooling	

Notes *1 Insulated Circuits

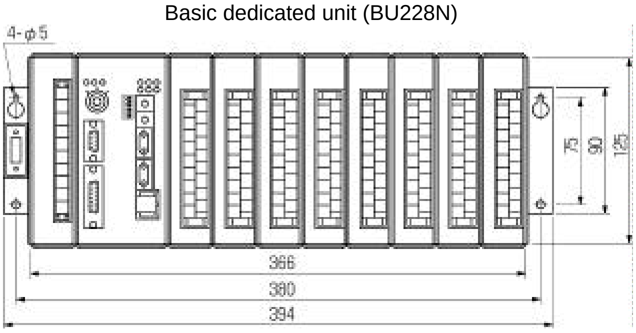
- between Power supply circuit and I/O circuit
- between Accessible metal parts and Power supply circuit
- between Accessible metal parts and I/O circuit
- between SELV circuit and Power supply
- between SELV circuit and I/O circuit

Accessible metal parts: Racks, Protective ground terminal, etc.
SELV (Safety Extra Low Voltage) circuit: Internal logic circuit

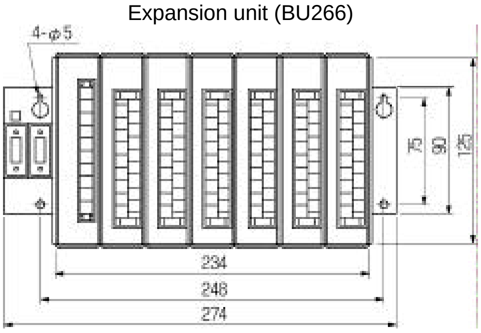
The accessible metal parts of the peripherals which are connected to the programmable controller by the standard cable are connected to the Protective ground terminal, or double insulated.

2.2 External dimensions

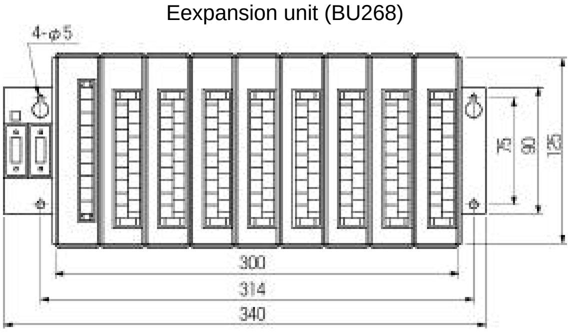
Basic dedicated unit (BU228N)



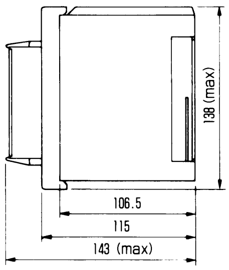
Expansion unit (BU266)



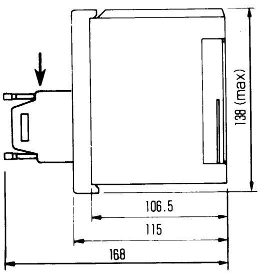
Eexpansion unit (BU268)



When 16-point I/O module installed

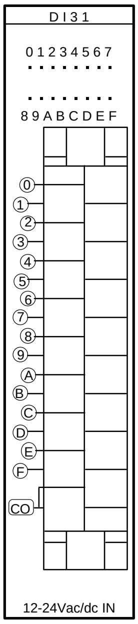


When 32/64-point I/O, MC11 installed



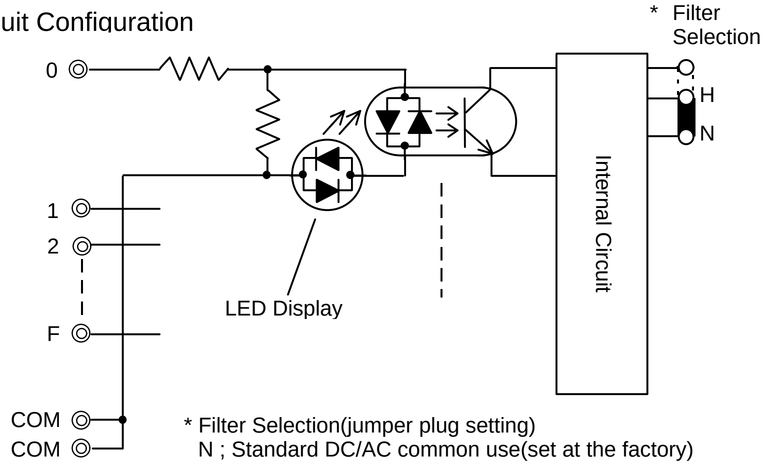
2.3 I/O Module Specifications

16-point DC/AC input



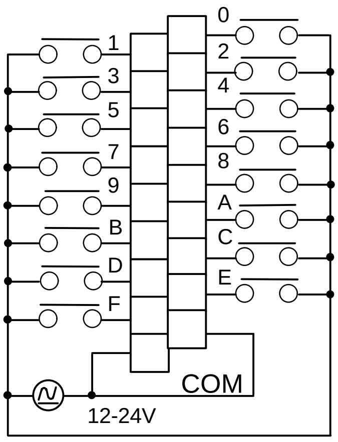
Item		DI31 (EX10*MDI31)
Input Voltage Range		$12 - 24\text{V}^{+10\%}_{-15\%}$ dc/ac(50/60Hz)
Minimum ON Voltage		9.6V or more
Maximum OFF Voltage		3.6V or less (leak current 0.7mA or less)
Input Current(Type)		Approx.8mA (at 24Vdc)
No. of input point		16 points/common
ON Delay	N Mode	10ms or less (dc) / 20ms or less (ac)
	H Mode	1.5ms or less (dc)
OFF Delay	N Mode	10ms or less (dc) / 15ms or less (ac)
	H Mode	1.5ms or less (dc)
Withstand Voltage		1500Vac / 1minute
Current Consumption		15mA (5Vdc) or less
Weight		Approx. 200g

Circuit Configuration

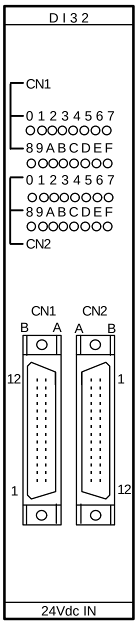


* Filter Selection(jumper plug setting)
N ; Standard DC/AC common use(set at the factory)
H ; High-speed DC input only

Terminal Connections

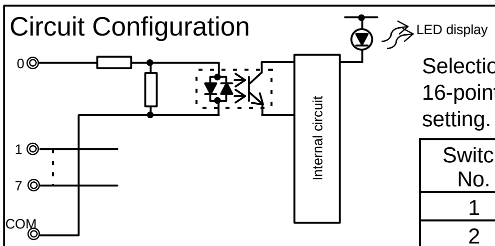


32-point DC input



Item		DI32 (EX10*MD132)
Input Voltage Range		24Vdc ± 10% - 15%
Minimum ON Voltage		18.0V
Maximum OFF Voltage		6.0V
Input Current		Approx 5mA(at 24Vdc)
Number of Input point		32points
ON Delay	N Mode	10msec or less
	H Mode	1.5msec or less
OFF Delay	N Mode	10msec or less
	H Mode	1.5msec or less
External Connections		2 x 24pin connectors
Common Configuration	Number of commons	4
	Number of Input points per Common	8 points
	Common Polarity	No Polarity
Withstand voltage		1500Vac / 1 minute
Current Consumption		80mA (5Vdc) or less
Weight		Approx. 200 g

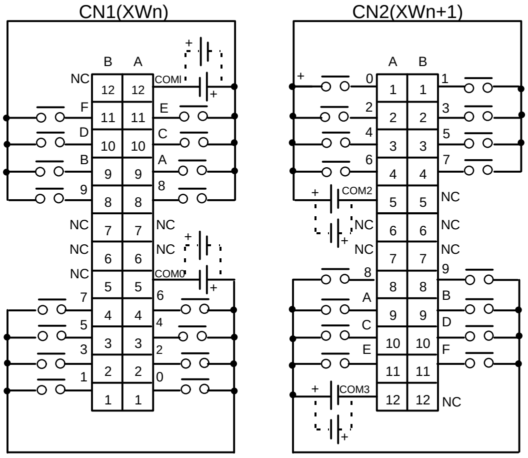
Circuit Configuration



Selection of the filter constant can be set in 16-point units (CN1, CN2) by DIP switch setting. (set on N mode at the factory)

Switch No.	OFF	ON	Application
1	N mode	H mode	CN1(XWn)
2	N mode	H mode	CN1(XWn+1)

Terminal Connections



NOTE

Connectors on the Module: FCN-365P024-AU (made by Fujitsu)

Cable side connectors:

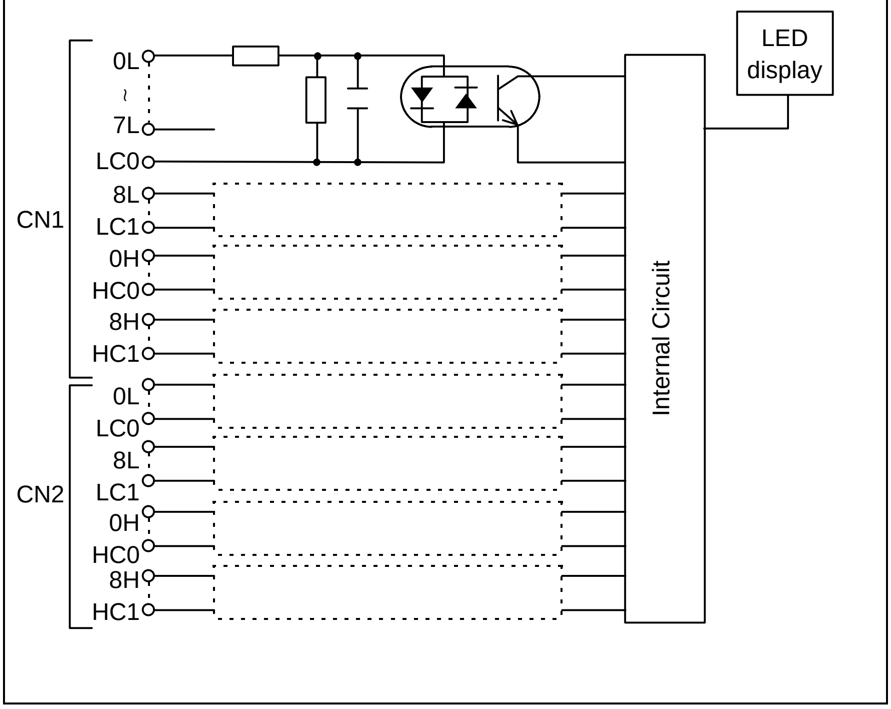
Soldering type (standard attached)

Connector FCN-361J024-AU (made by Fujitsu)

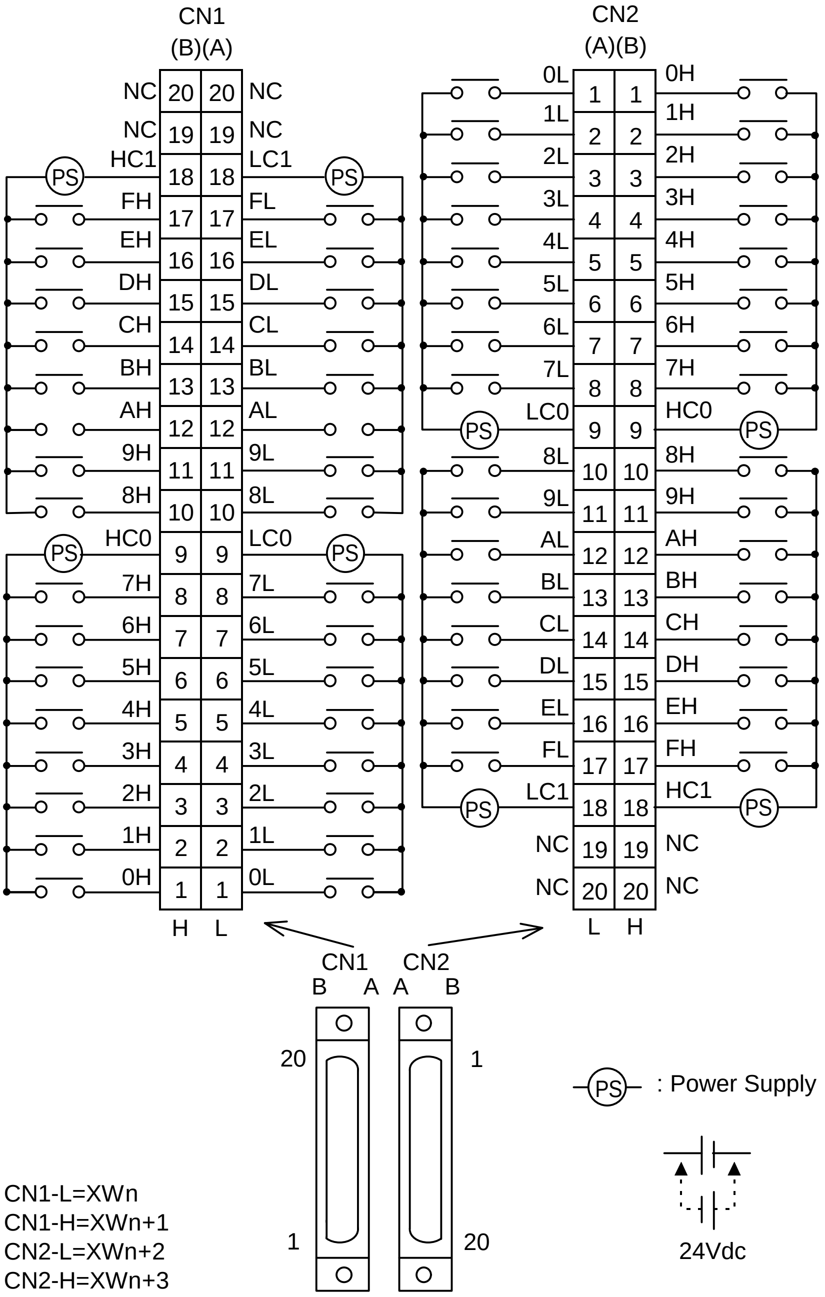
Connector cover FCN-360C024-E (made by Fujitsu)

64-point DC input.

Item		DI235
Input Voltage Range		24Vdc $^{+10\%}_{-15\%}$
Input Current		Apporx.4mA(at 24Vdc)
Input Impedance		5.8kΩ(24Vdc)
Minimum ON Voltage		16V
Maximum OFF Voltage		5V
ON delay		10ms or less than
OFF delay		15ms or less than
External Connections		2 × 40 pin connectors
Common Configuration	Number of Commons	8
	Number of Input Points per Common	8 points
	Common Polarity	No Polarity
Derating Condition		See next page
Current Consumption		100mA (5Vdc) or less
Withstand voltage		1500Vac/ 1 minute
Weight		Approx. 250g
Circuit Configuration		



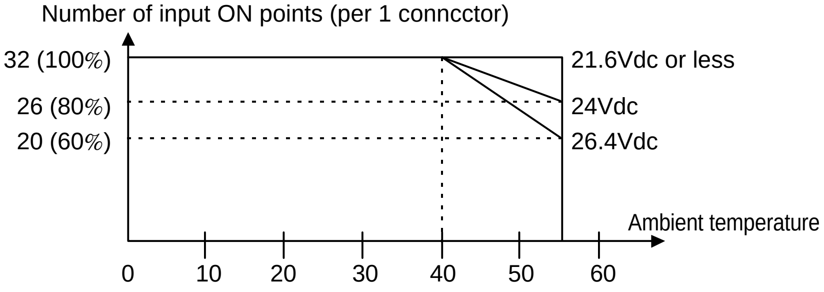
Terminal Connections



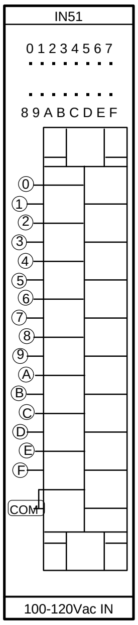
CN1-L=XWn
CN1-H=XWn+1
CN2-L=XWn+2
CN2-H=XWn+3

Connectors on the Module	:	FCN-365P040/AU (made by Fujitsu)
Cable side connectors	:	Soldering type (standard attached)
Connector	:	FCN-361J040-AU (made by Fujitsu)
Connector cover	:	FCN-360C040-E (made by Fujitsu)

Derating Condition

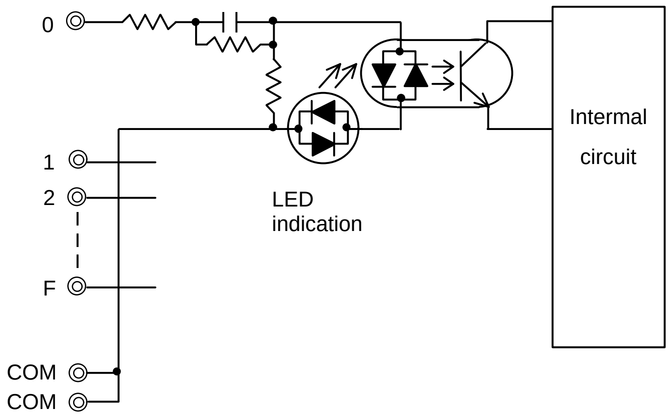


16-point AC input

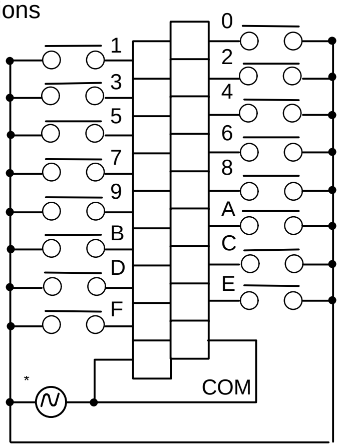


Item	IN51 (EX10*MIN51)	IN61 (EX10*MIN61)
Input Voltage Range (Sine wave)	100-120Vac + 10% - 15% (50/60Hz)	200-240Vac + 10% - 15% (50/60Hz)
Minimum ON voltage (Sine wave)	80Vac or more	160Vac or more
Maximum OFF voltage (Sine wave)	30Vac or less (leak current 2mA or less)	60Vac or less (leak current 2mA or less)
Input Current(Sine wave)	Approx 7mA (100V-50Hz)	Approx 6mA (200V-50Hz)
Number of Input Points	16 points (single common)	16 points (single common)
ON Delay (Sine wave)	20mS or less	20mS or less
OFF Delay (Sine wave)	15mS or less	15mS or less
Voltage Insulation	1500Vac / 1 minute	1500Vac / 1 minute
Current Consumption	15mA (5Vdc) or less	15mA (5Vdc) or less
Weight	Approx 250g	Approx 250g

Circuit Configuration

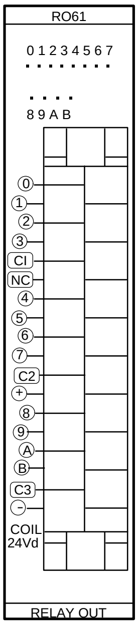


Terminal connections



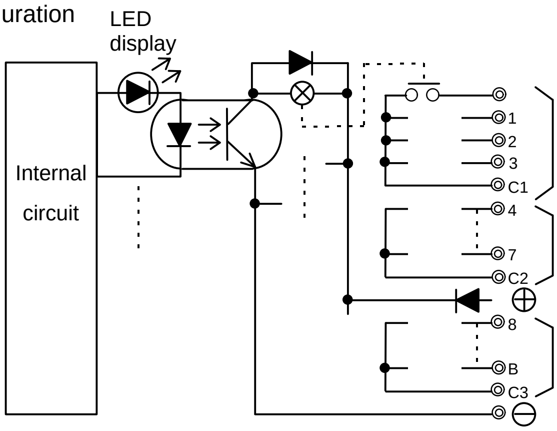
* IN51 : 100-120Vac (50/60HZ)
IN61 : 200-240Vac (50/60HZ)

12-point Relay Output

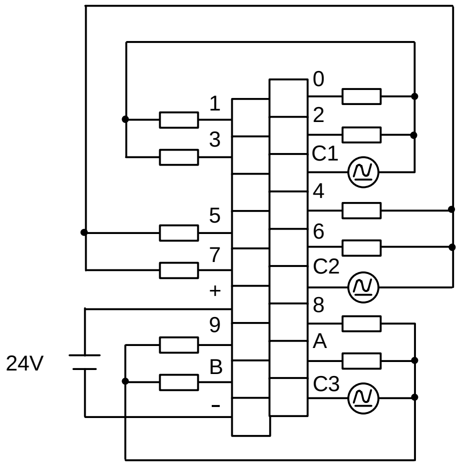


Item	RO61 (EX10*MRO61)
Load voltage	24Vdc, +20%(MAX)/240Vac, + 10%(MAX)
Maximum load	2A/point (resistive load), 1A/point (inductive load), 4A/4points common
Minmum load	50mW (5V or more)
Number of output points	12 points (4 points / common)
ON delay	10ms or less
OFF delay	15ms or less
Leakage current When OFF	0mA
Withstand voltage	1500Vac / 1 minute
Current consumption	50mA (5Vdc) or less
External relay Coil Power required	24Vdc +/- 10% - 140mA/all points ON (10mA/point)
Weight	Approx 250g

Circuit Configuration



Terminal Connections

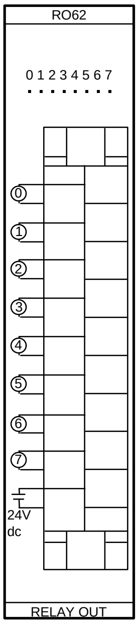


NOTE

1.ON/OFF life of relays: Electrical 100,000 times
Mechanical 20 million times

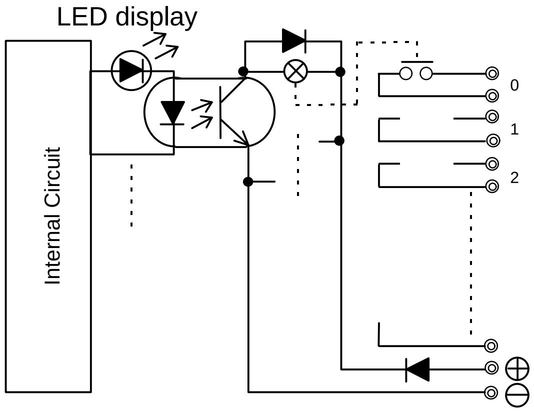
2.No overload protection fuses are built into this module.Therefore connect fuses externally suitable to the current capacity.

8-point Isolated Relay Output

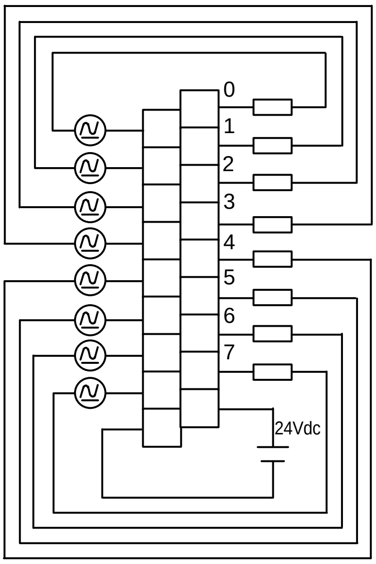


Item	RO62 (EX10*MRO62)
Load voltage	24Vdc, +20%(MAX)/240Vac, +10%(MAX)
Maximum load	2A/point (resistive load), 1A/point (inductive load)
Minimum load	50mW (5V or more)
Number of output points	8 points (each point isolated)
ON Delay	10msec or less
OFF Delay	15msec or less
Leakage current When OFF	0mA
Voltage insulation	1500Vac / 1 minute
Current consumption	40mA (5Vdc) or less
External Relay Coil Power Required	24Vdc +/– 10% - 100mA/all points ON (10mA/point)
Weight	Approx 250g

Circuit Configuration



Terminal Connections

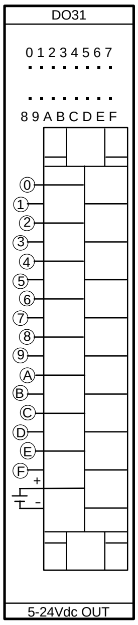


NOTE

1.ON/OFF life of relays: Electrical 100,000 times
Mechanical 20 million time

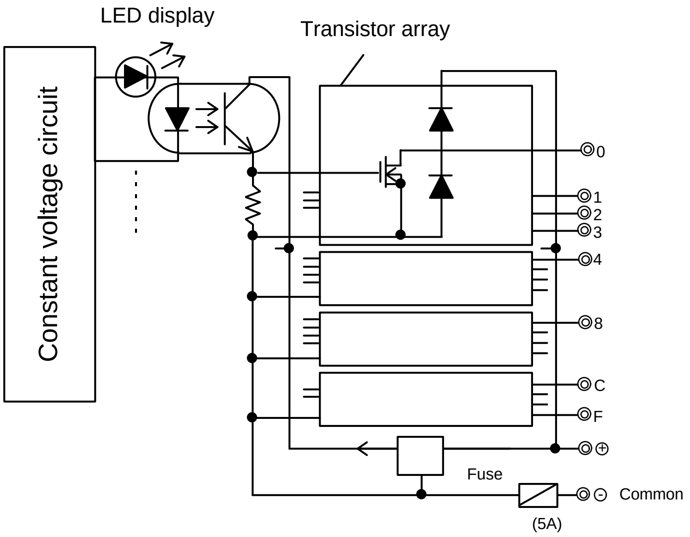
2.No overload protection fuses are built into this module. Therefore connect fuses externally suitable to the current capacity.

16-point Transistor Output

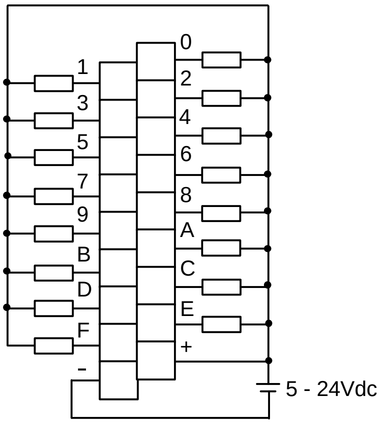


Item	DO31 (EX10*MDO31)
Load power supply	5-24Vdc+10%/-5% (Internal current consumption 35mA or less)
Output ON current	1A/point (external power supply 7V or more) 0.3A/point (external power supply less than 7V) 1.2A/4 points (4-element transistor array)
Output ON resistance	1.5Ω or less
Number of output points	16 points (single minus common)
On Delay	1ms or less
OFF Delay	1ms or less
Leak current when Output OFF	0.1mA or less
Voltage insulation	1500Vac/ 1 minute
Current consumption	60mA (5Vdc) or less
Weight	Approx 200g

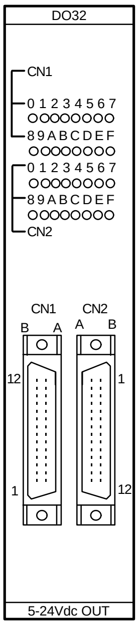
Circuit Configuration



Terminal Connections

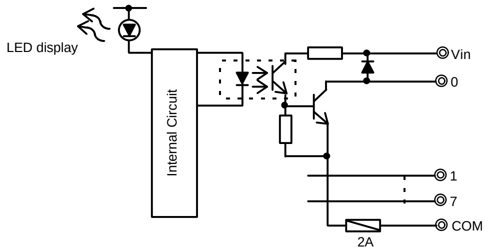


32-point Transistor Output

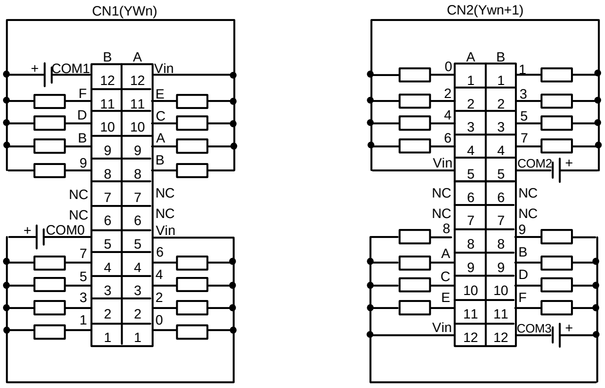


Item		DO32 (EX10*MDO32)
Load voltage		5-24Vdc +10%/-5%
Output ON current		100mA/point (when load voltage 24V) 20mA/point (when load voltage 5V) 800mA/common
Saturation voltage when ON		0.4V or less
Number of output points		32 points
Output type		Current sinking
ON Delay		1msec or less
OFF Delay		2msec or less
Leakage current When OFF		0.1mA or less
External connections		2 x 24 pin connectors
Common configuration	Number of Common	4
	Number of output points per common	8 points
	Common polarity	minus common
Withstand voltage		1500Vac / 1 minute
Current consumption		250mA (5Vdc) or less
Built-in fuse		2A/common x 4
Weight		Approx 250g

Circuit Configuration



Terminal Connections



NOTE

Connectors on the Module : FCN-365P024-AU (made by Fujitsu)

Cable side connectors

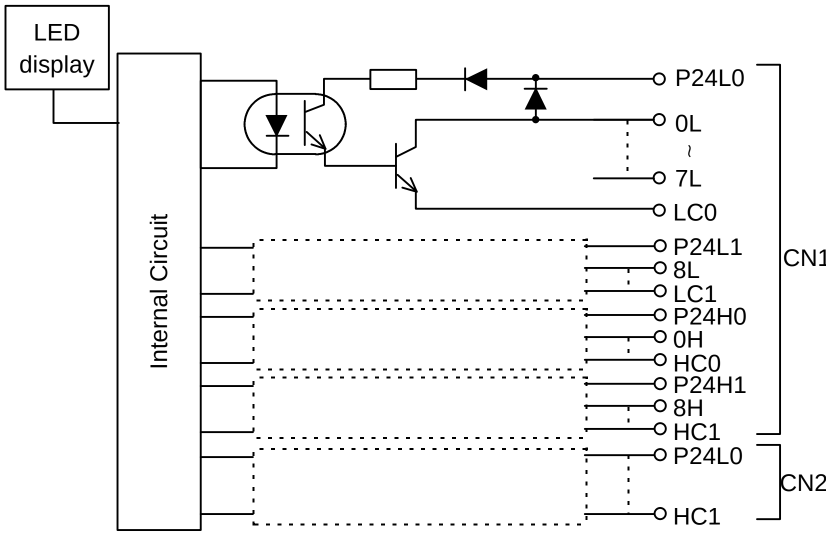
Soldering type (standard attached)

Connector : FCN-361J024-AU (made by Fujitsu)

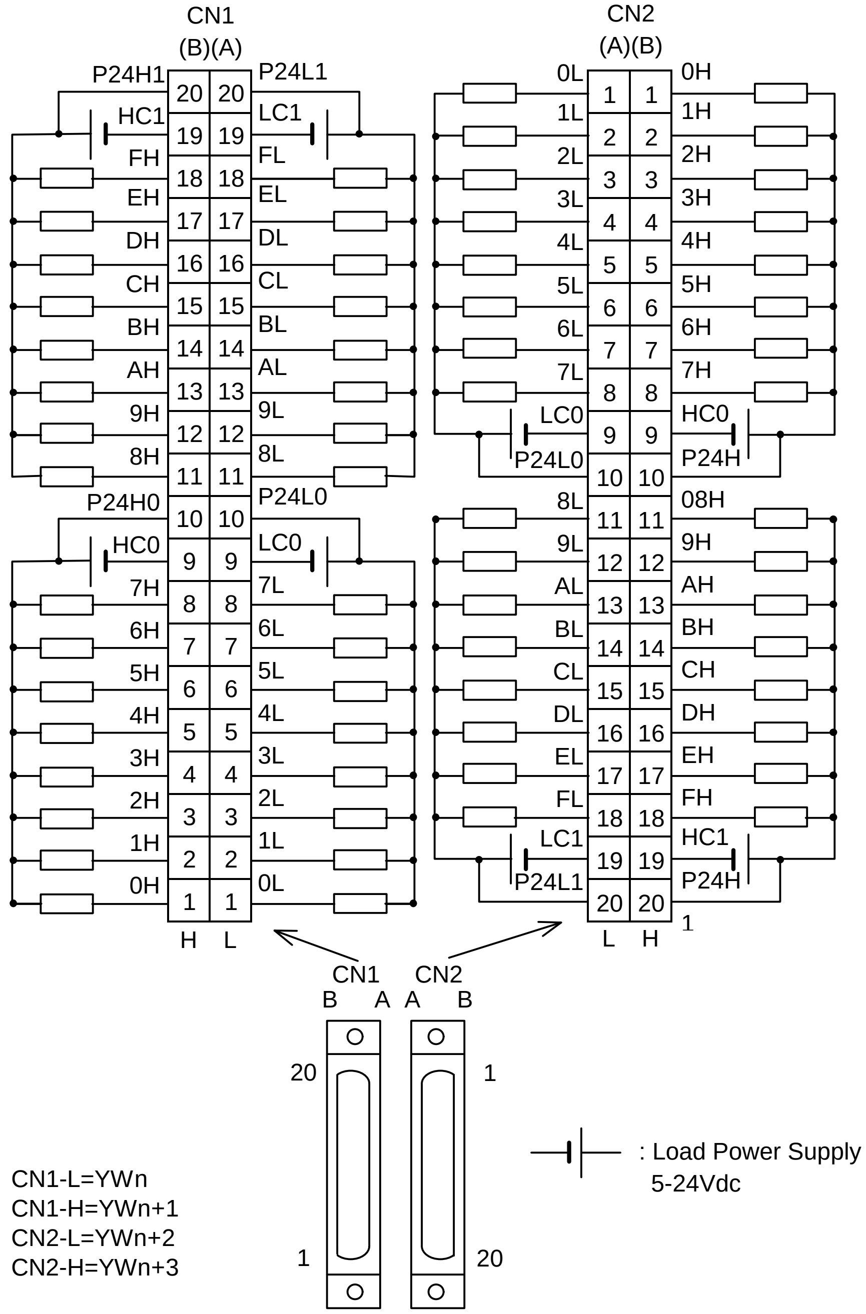
Connector cover : FCN-360C024-E (made by Fujitsu)

64-point Transistor Output

Item		DO235
Output type		Current sinking
Number of output		64 points
Load Voltage		5-24Vdc +10%/-5%
Output ON Current		0.1A/point (9.6 - 26.4Vdc) 0.05A/point (4.5 - 9.5Vdc)
Saturation voltage when ON		0.4V or less
Leakage current when OFF		0.1mA or less (24Vdc)
ON delay		1ms or less
OFF delay		1ms or less
External Connections		2 x 40pin connectors
Common Configuration	Number of Common	8
	Number of Output points per common	8 points
	Common Polainty	minus
Current Consumption		250mA (5Vdc) or less
Withstand voltage		1500Vac/1 minute
Built-in fuse		none
Surge Protection		Diode
Weight		Approx. 250g
Circuit Configuration		



Terminal Connections

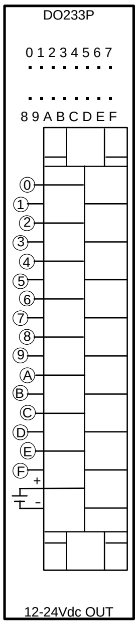


: Load Power Supply
5-24Vdc

CN1-L=YWn
CN1-H=YWn+1
CN2-L=YWn+2
CN2-H=YWn+3

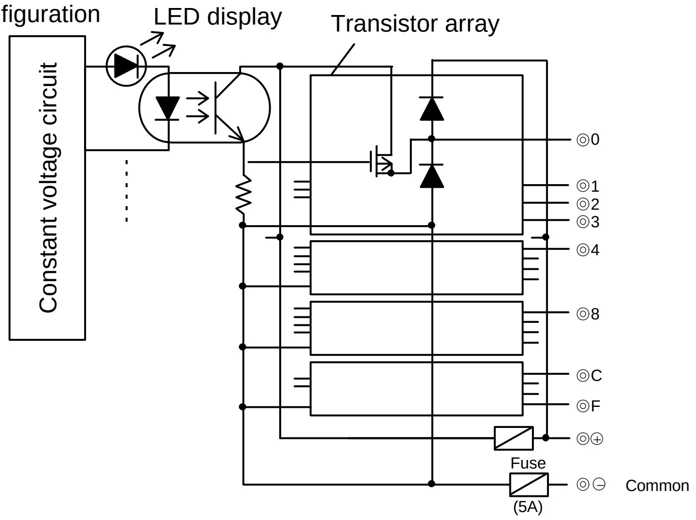
Connectors on the Module	:	FCN-365P040/AU (made by Fujitsu)
Cable side connectors	:	Soldering type (standard attached)
Connector	:	FCN-361J040-AU (made by Fujitsu)
Connector cover	:	FCN-360C040-E (made by Fujitsu)

16-point Transistor Output

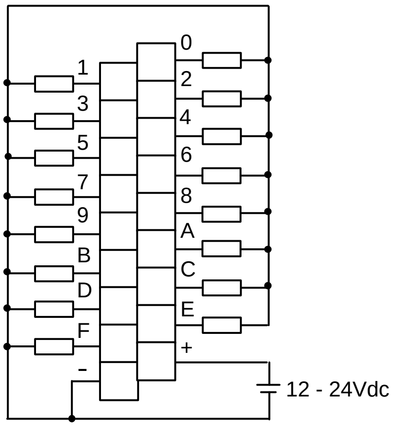


Item	DO233P
Load power supply	(Internal current consumption 35mA or less)
Output ON current	1A/point (external power supply 7V or more) 1.2A/4 points (4-element transistor array)
Output ON resistance	1.5Ω or less
Number of output points	16 points (single plus common)
On Delay	1ms or less
OFF Delay	1ms or less
Leak current when Output OFF	0.1mA or less
Voltage insulation	1500Vac/ 1 minute
Current consumption	60mA (5Vdc) or less
Weight	Approx 200g

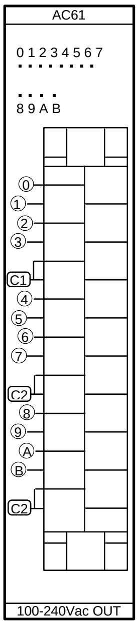
Circuit Configuration



Terminal Connections

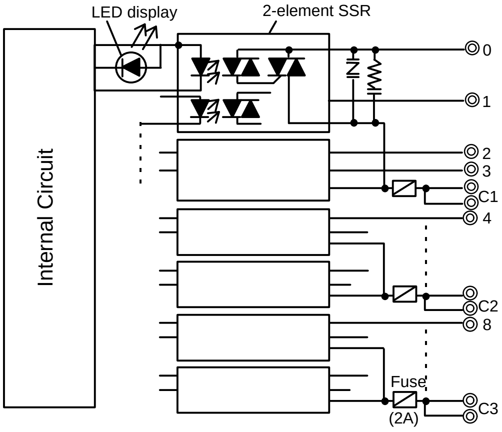


12-point Triac Output

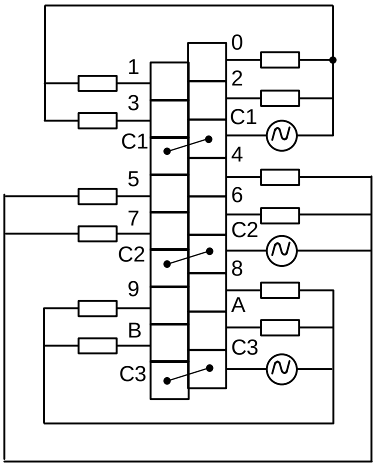


Item	AC61 (EX10*MAC61)
Load voltage	100-240Vac +10%/-5%(50/60Hz sine wave)
Output ON current	0.5A/point, 0.6A(2-element SSR)
Saturated ON voltage	1.5V or less (0.3A load)
Number of output points	12 points (4 points / common)
ON Delay	1msec or less
OFF Delay	1/2 cycle of load power supply + 1msec or less
Leakage Current When OFF	1.2mA (100Vac) or less, 3mA (240Vac) or less
Withstand voltage	1500Vac / 1 minute
Current consumption	300mA (5Vdc) or less (20mA/point)
Weight	Approx. 250g

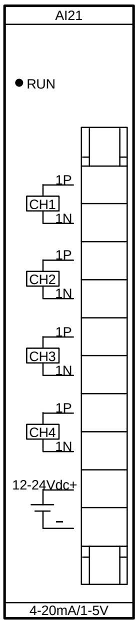
Circuit Configuration



Terminal Connections

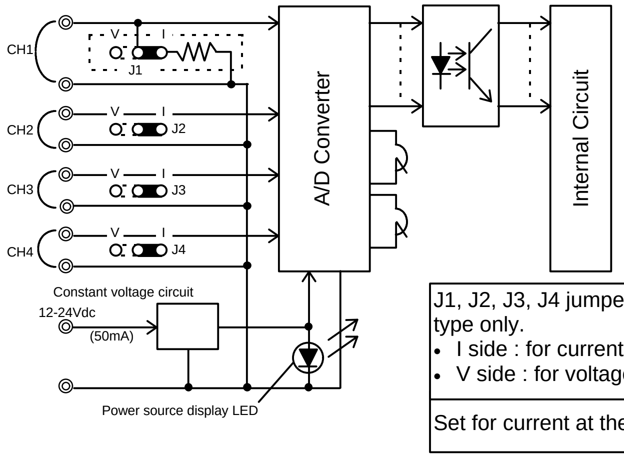


4-Channel Analog Input Input(8-bit)



Item	A121 (EX10*MA121)	A131 (EX10*MA131)
Input range	1 - 5V or 4 - 20mA	0 - 10V
Input Impedance	1 - 5V : 500KΩ or more 4 - 20mA : 250Ω	500KΩ or more
Number of input points	4 points (N side common)	4 points (N side common)
Resolution	1 - 5V : 0 - 250 4 - 20mA : 0 - 250	0 - 10V : 0 - 250
Overall Accuracy	±1% (FS)	±1% (FS)
4 point (N side common)	Approx. 1msec	Approx. 1msec
Wire breakage detection	Only possible for 4-20mA	-
External Power Supply failure detection	Yes	Yes
Withstand voltage	500Vac / 1 minute	500Vac / 1 minute
Current consumption	50mA 5Vdc or less	50mA 5Vdc or less
External power required	12 - 24Vdc±10% - 50mA	12 - 24Vdc±10% - 50mA
Weight	Approx. 200g	Approx. 200g

Circuit Configuration

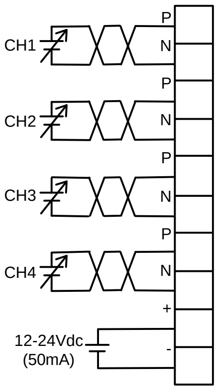


J1, J2, J3, J4 jumper plugs are AI21 type only.
- I side : for current (4 - 20mA)
- V side : for voltage (1-5V)

Set for current at the factory

The AI21 type is set to current input (4-20mA) at the factory.
For voltage (1-5V) input, set J1 - J4 to the V side.

Terminal Connections

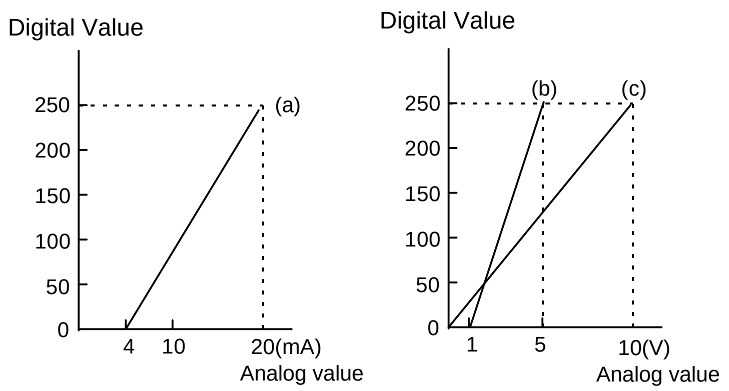


Isolate external power supply line (12 - 24Vdc) from other signal lines.

Use shielded twisted-pair cables for analog signals, and ground the shields securely.

4-Channel Analog Input (8-bit) (Continued)

A/D Conversion



(a) 4-20mA range : D = 15.625 x A-62.5
(b) 1-5V range : D = 62.5 x A-62.5
(c) 0-10V range : D = 25 x A

D = Digital Value
A = Analog Value

Data Format

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
XW	B	*	*	*	*	*	*	*	D	D	D	D	D	D	D	D

D : Data bit (8 bits)
0-250 (H00-HFA)
(Over-voltage/over-current input 255 counts)

B : External line abnormal detection bit
0 = Normal

1 =Abnormal
When all data bits are 0 ••Current input line open
When all data bits are 1 ••External power is OFF

* : Always 0

4 words occupied
(X4W)

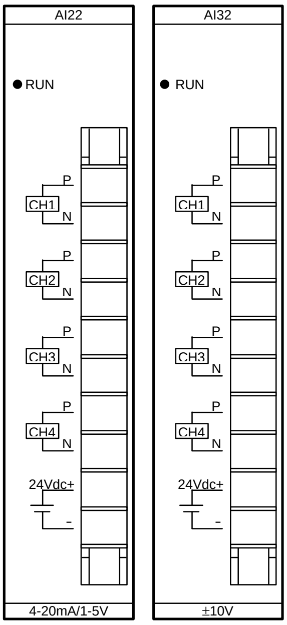
	F O
XWn	CH1
XWn+1	CH2
XWn+2	CH3
XWn+3	CH4

NOTE

1.In the voltage input specification, when there is an open-circuit between the input terminals, the data bits do not become 0. (They become indeterminate between 1 and 250).

2.It is recommended that unused channels be shorted between the input terminals.

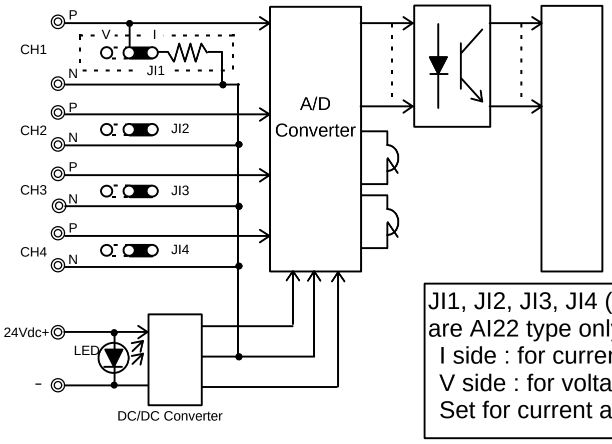
4-Channel Analog Input (12-bit)



Item	AI22 (EX10*MA122)	AI32 (EX10*MA132)
Input Range	1 - 5V or 4 - 20mA	-10 - +10V
Input Impedance	1 - 5V : 1MΩ or more 4 - 20mA : 250Ω	1MΩ or more
Number of Input Points	4 points (N side common)	4 points (N side common)
Overall Accuracy	± 0.5% : 25ºC ± 1%FS / 0 - 55ºC	± 0.5% : 25ºC ± 1%FS / 0 - 55ºC
Resolution	1 - 5V : 0 - 4000 4 - 20mA : 0 - 4000	-10 - +10C : -2000 - 2000
Conversion Cycle	Approx. 9.6msec/4 channels	Approx. 9.6mS/4 channels
Wire Breakage Detection	Only possible for 4 - 20mA	
External Power Supply Break Detection	Yes	Yes
Withstand Voltage	1500Vac / 1 minute	1500Vac / 1 minute
Current consumption	50mA (5Vdc) or less	50mA (5Vdc) or less
External Power Required	24Vdc ±10% - 50mA	24Vdc ±10% - 50mA
Weight	Approx. 200g	Approx. 200g

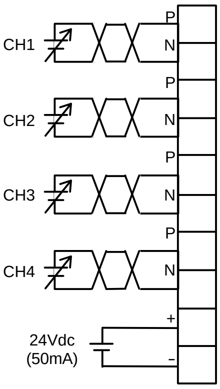
The AI22 type is set to current input (4-20mA) at the factory. When using voltage (1-5V) input, reset the jumper plugs

Circuit Configuration



JI1, JI2, JI3, JI4 (jumper plugs) are AI22 type only
I side : for current (4 - 20mA)
V side : for voltage (1 - 5V)
Set for current at the factory

Terminal Connections

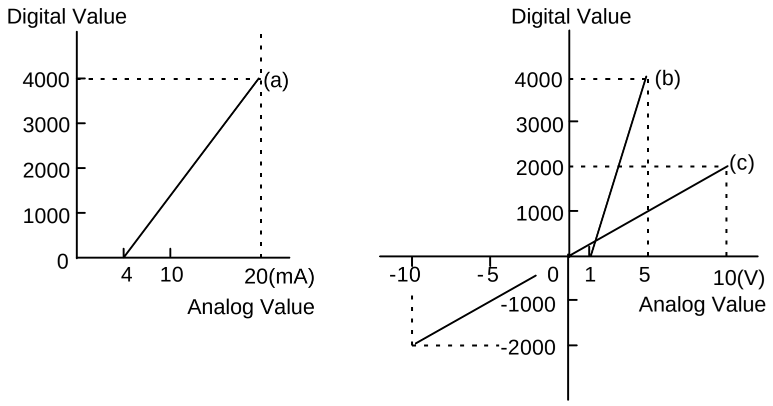


Isolate external power supply line (24Vdc) from other signal lines.

Use shielded twisted-pair cables for analog signals, and ground the shields securely.

4-Channel Analog Input (12-bit) (Continued)

A/D Conversion



(a) 4-20mA range : D = 250 x A-1000
(b) 1-5V range : D = 1000 x A-1000
(c) ±10V range : D = 200 x A

D = Digital value
A = Analog value

Data Format (Input occupies 4 words)

- 4-20mA/1-5V

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
B	*	*	*	D	D	D	D	D	D	D	D	D	D	D	D

D : Data bit (12 bits)
0-4000 (H0000-H0FA0)
B : External line abnormal detection bit
0 = Normal
1 = Abnormal
(Current input line open, or external power OFF)

* : Always 0

- ±10V

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D

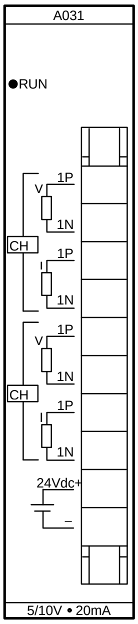
S : Sign bit
0 = Positive
1 = Negative
D : Data bit (11 bits)
- 2000 - 2000 (HF830 - H07D0)
2's complement if negative

NOTE

1. In the voltage input specification, when there is an open-circuit between the input terminals, the data bits do not become 0.

2. It is recommended that unused channels be shorted between the input terminals.

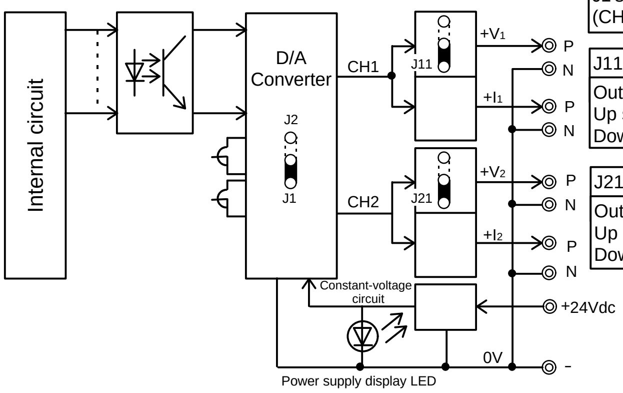
2-Channel Analog Output (8-bit)



Item	AO31 (EX10*MA031)
Output Range	1 - 5V, 4 - 20mA paired output
	0 - 10V
	0 - 5V
Load Impedance	5V full-scale terminal : 5KΩ or more
	10V full scale terminal: 10KΩ or more
	20mA full-scale terminal : 600Ω or less
Number of Output Points	2 points (each voltage, current pair) (each N side common)
Resolution	0 - 250 (full-scale)
Overall Accuracy	± 1% (FS)
Conversion Cycle	Approx. 1msec
External Power Supply Break Failure	No
Voltage Insulation	1500Vac / 1 minute
Current Consumption	70mA (5Vdc) or less
External Power Supply	24Vdc ± 10%-90mA
Weight	Approx. 200g

The A031 type is set to 1-5V, 4-20mA setting at the factory. Refer to the circuit configuration for other settings.

Circuit Configuration

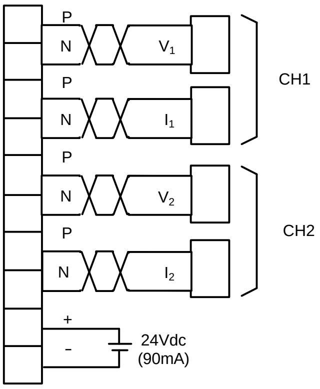


J1, J2 switching jumper
J2 short : Reference value 0V (0mA)
J1 short : Reference value 1V (4mA)
(CH1, CH2 operation)

J11 switching jumper
Output voltage level for CH1
Up side short : 5V
Down side short : 10V

J21 switching jumper
Output voltage level for CH2
Up side short : 5V
Down side short : 10V

Terminal Connections

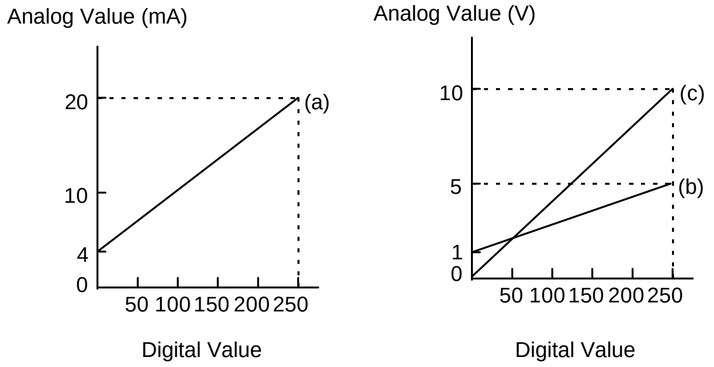


Isolate external power supply line (24Vdc) from other signal lines.

Use shielded twisted-pair cables for analog signals.

2-Channel Analog Output (8-bit) (Continued)

D/A Conversion



(a) 4-20mA range : A = 0.064 x D + 4
(b) 1-5V range : A = 0.016 x D + 1
(c) 0-10V range : A = 0.04 x D

A = Analog value
D = Digital value

Data Format

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
YW	*	*	*	*	*	*	*	*	D	D	D	D	D	D	D	D

D : Data bit (8 bits)
0-250 (H00-HFA)
* : No effect (No effect on D/A conversion)

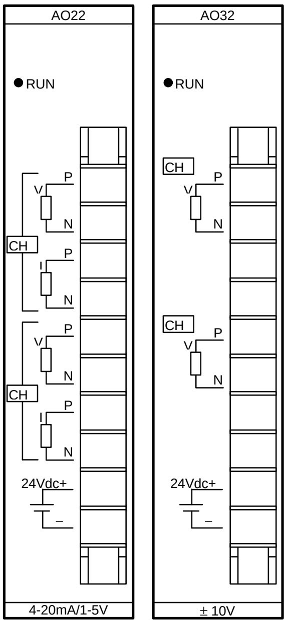
2 words occupied
(Y2W)

	F 0
YWn	CH1
Ywn+1	CH2

NOTE

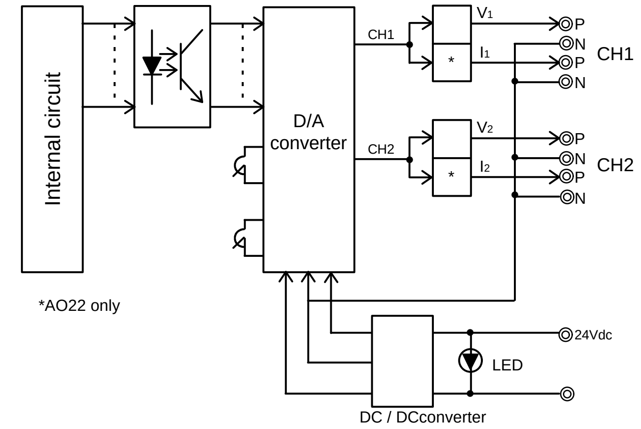
When executing direct output, two registers (both channels) should be specified to output.

2-Channel Analog Output (12-bit)

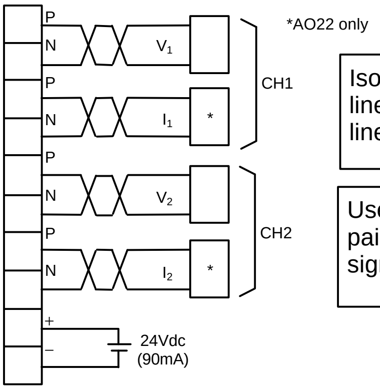


Item	AO22 (EX10*MAO22)	AO32 (EX10*MAO32)
Output Range	1 - 5V or 4 - 20mA	–10 – + 10V
Load Impedance	1-5V : 5KΩ or more 4-20mA : 600Ω or less	5KΩ or more
Number of Output Points	2 points (each N side common) (voltage, current pair)	2 points (each N side common)
Resolution	1~5V : 0~4000 4~20mA : 0~4000	–10~ + 10V :–2000~2000
Overall Accuracy	± 0.5%FS/25°C ± 1% FS/0~55°C	± 0.5%FS/25°C ± 1% FS/0~55°C
Conversion Cycle	Approx. 1mS	Approx. 1mS
External Power Supply FailureDetection	No	No
Withstand voltage	1500Vac/ 1 minute	1500Vac / 1 minute
Current Consumption	170mA (5Vdc) or less	170mA (5Vdc) or less
External Power Required	24Vdc ± 10%-90mA	24Vdc ± 10%-90mA
Weight	Approx. 200g	Approx. 200g

Circuit Configuration



Terminal Connections

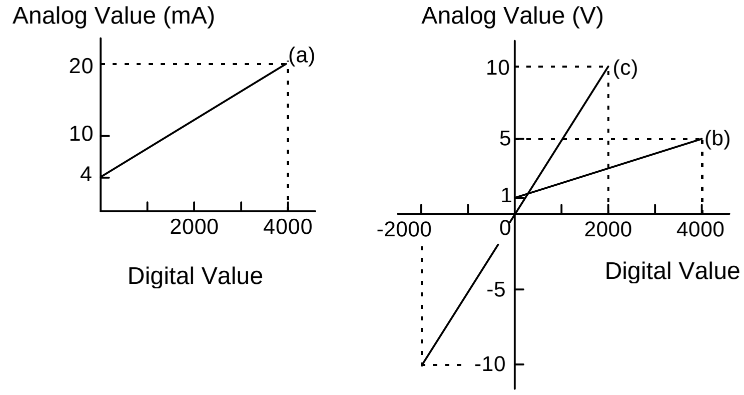


Isolate external 24Vdc line from other signal lines.

Use shielded twisted-pair cables for analog signals.

2-Channel Analog Output (12-bit) (Continued)

D/A Conversion



(a) 4-20mA range : A = 0.004 x D + 4
(b) 1-5V range : A = 0.001 x D + 1
(c) ±10V range : A = 0.005 x D

A = Analog value
D = Digital value

Data Format (Output occupies 2 words)

- 4-20mA / 1-50

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
YW	*	*	*	*	D	D	D	D	D	D	D	D	D	D	D	D

D : Data bit (12 bits)
0-4000 (H0000-H0FA0)
* : No effect (No effect on D/A conversion)

- ± 10V

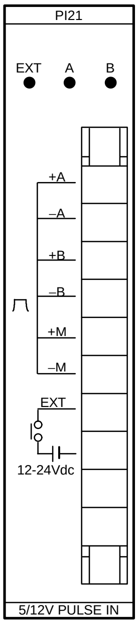
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
YW	*	*	*	*	S	D	D	D	D	D	D	D	D	D	D	D

S : Sign bit
0 = Positive
1 = Negative
D : Data bit (11 bits)
- 2000 - 2000 (HF830 - H07D0)
2's complement if negative

NOTE

When executing direct output, two registers (both channels) should be specified to output

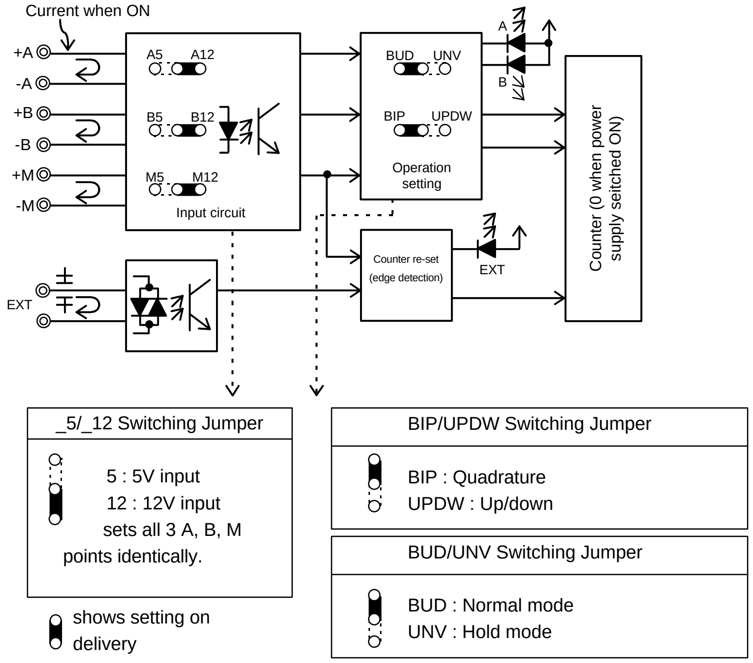
Single-Channel Pulse Input



At the factory setting, the PI21 is set to 12V input, quadrature and normal counter operation. See the Circuit Configuration for other settings

Item		PI21 (EX10*MP121)
Input Voltage Range	A, B, M	12V ± 10% / -5% (12V setting), 5V +10% / -5% (5V setting)
	EXT	12~24Vdc +10% -15%
Minimum ON Voltage	A, B, M	9V (12V setting), 3.5V (5V setting)
	EXT	9.6V
Maximum OFF Voltage	A, B, M	2V (12V setting), 1V (5V setting)
	EXT	3.6V
Input Current	A, B, M	12V-7.5mA (12V setting), 5V-10mA (5V setting)
	EXT	24V-10mA, 12V-5mA
Number of Input Points		1 point phase A, B, M and ETX
Pulse Counting Speed		100Kpps (max) (pulse-width 4μ sec or more)
Counter Configuration		24-bit binary
Pulse Input Mode	Quadrature	Phase A, B (90 degree phase shift), up/down
	Up/Down	Phase A: count up / phase B: count down
Counter Operation Mode	Normal	Counter clears at simultaneous ON timing of phase M and EXT input (edge), always executes count apart from this.
	Hold	Executes count only when both phase M and EXT input are ON, count stops when either is OFF (Count value maintained). (Counter clear is at the same timing as the Normal Mode).
External (EXT) Input Operation		Counter clears at simultaneous ON timing of phase M and EXT input (edge)
External (EXT) Input Delay		ON-OFF, OFF-ON each 5mS or less
Withstand Voltage		1500Vac / 1 minute (but except between each of the A, B, M phases)
Current Consumption		80mA (5Vdc) or less
Weight		Approx. 200g

Circuit Configuration



_5/_12 Switching Jumper

5 : 5V input
12 : 12V input
sets all 3 A, B, M points identically.

shows setting on delivery

BIP/UPDW Switching Jumper

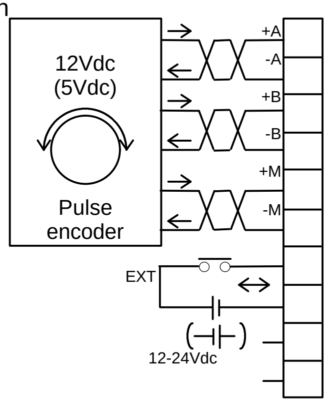
BIP : Quadrature
UPDW : Up/down

BUD/UNV Switching Jumper

BUD : Normal mode
UNV : Hold mode

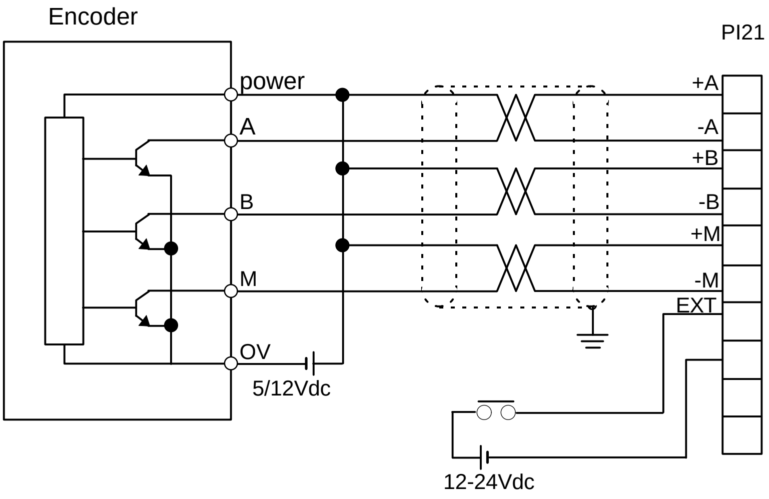
Single-Channel Pulse Input (Continued)

Terminal Connection

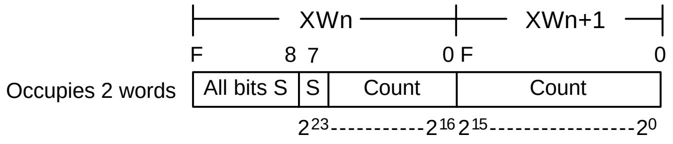


Use shielded twisted-pair cables for pulse signals, and ground the shields securely.

Wiring Example



Data Format



For the count values (24 bits), the lower 16 bits are read as XWn + 1, and the upper 8 bits as the bit 0 to bit 7.

Count = -8388608 to 8388609 (or 0 to 16777215 if upper 8 bits are masked).

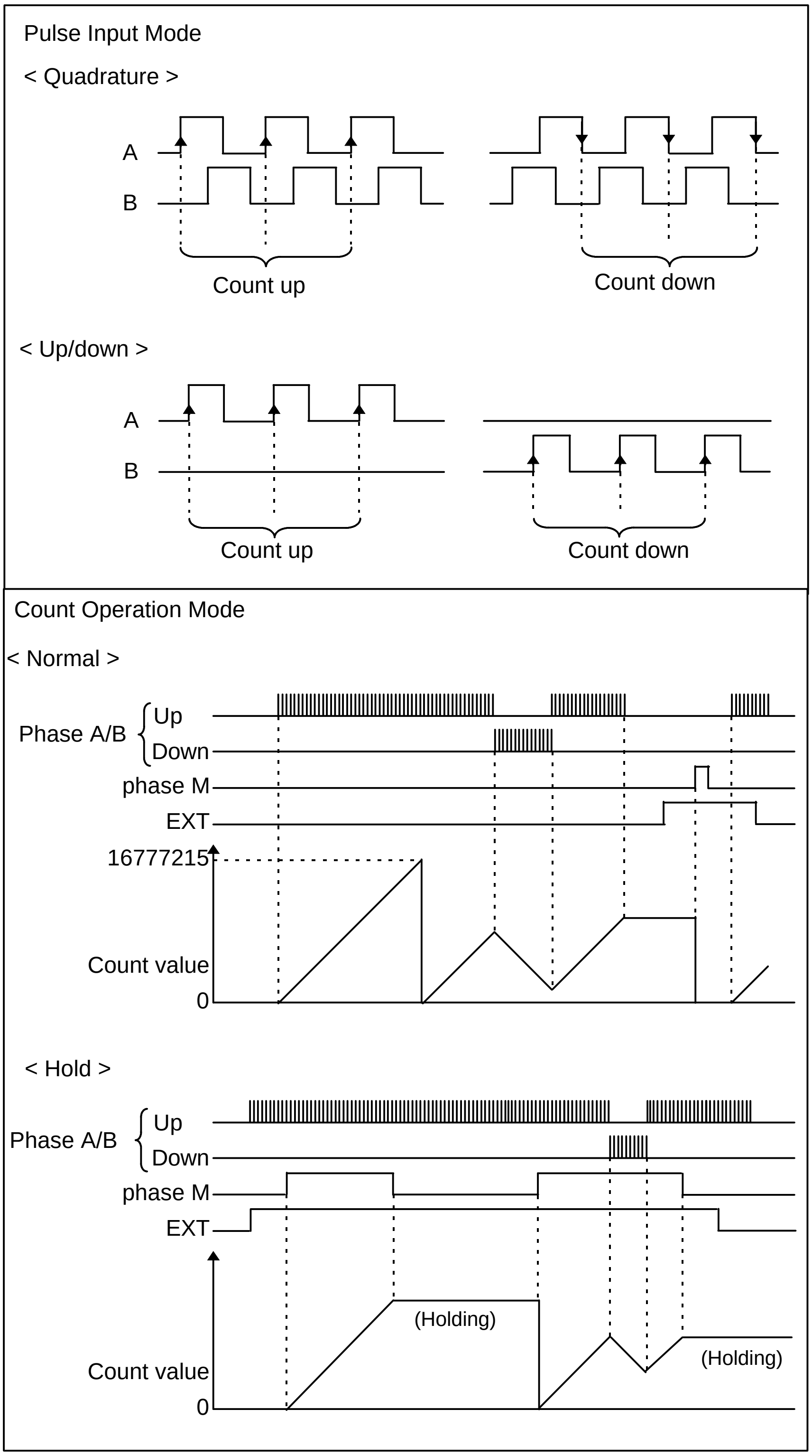
NOTE

Bit 8 to bit F of XWn are the same as bit 7 of XWn (sign bit). In order to match the double-length register configuration of the T2N, use after exchanging the upper word and the lower word.

(Example : Store count value in the double-length register of D0101 • D0100)

```
├─[XWn      MOV D0101]─┤
├─[XWn+1  MOV D0100]─┤
```

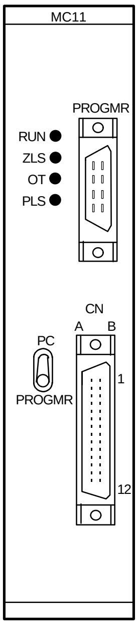
Single-Channel Pulse Input (Continued)



NOTE

When executing direct input, two registers should be specified to input.

Single-Axis Position Control



Item			MC11 (EX10*MMC11)
Number of Control Axes			1 axis
Control Units			Pulse, inch, mm, etc.
Control Range			± 999,999
Point Data Capacity			64 points
Maximum Speed			200kpps
Operating Speed Selection			Origin return speed, Maximum speed, Minimum speed
Acceleration/deceleration System			Automatic trapezoidal / triangular system
Acceleration/deceleration Time			0 - 26 seconds
Backlash Compensation			0 - 1000 pulses
Zero Position Offset Amount			± 999,999 command units
Dwell Time			0 - 99 seconds
I/O occupancy points			X + Y 4 W (64 bits)
Parameter Storage			EEPROM
External Input	Input Voltage		12/24Vdc (zero marker: 5/12/24V)
	Input Current		10mA (when 24V input)
	ON/OFF Voltage		9.6V / 3.2V
	ON/OFF Delay		5msec (zero marker: 1msec)
External Output	Pulse Output	Mode Switch Setting	.CW/CCW error counter clear .PULSE/DIR (pulse/direction), error counter clear
		Output Method	Open collector (5-24V, 50mA)
		ON/OFF Delay	2µS
	RUN Output	Output Method	Open collector (5-24V, 50mA)
		Operation	ON during normal operation
Current Consumption		Internal	200mA 400mA- (when HP connected)-5Vdc
		External	100mA-12 / 24Vdc

Connector Arrangement

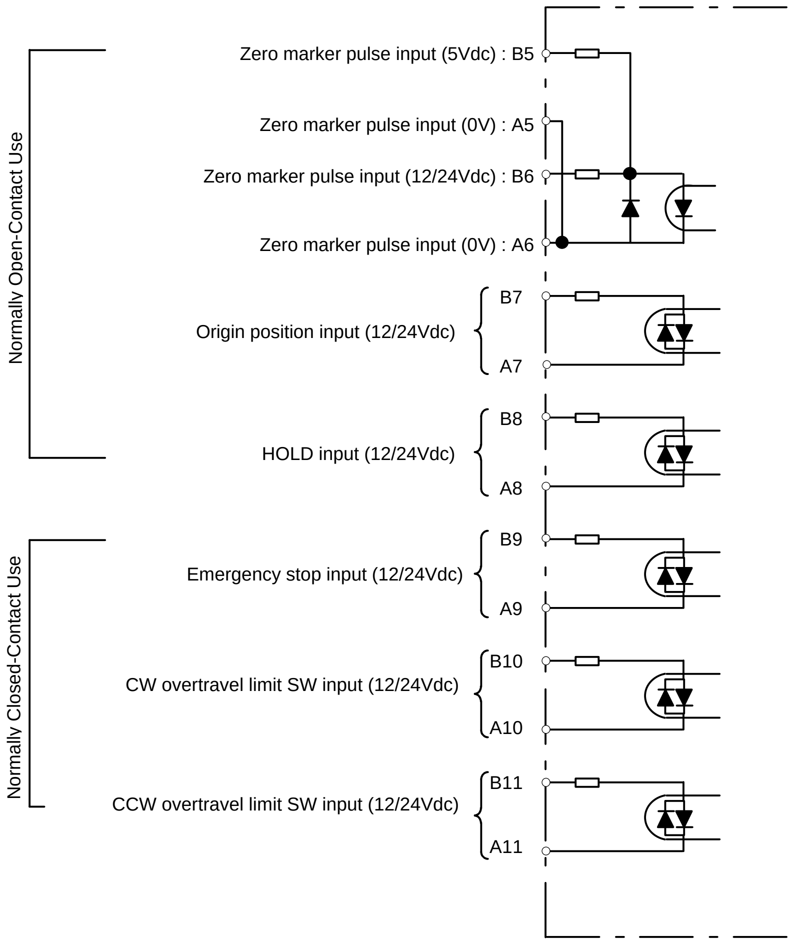
	A	B	
HUN output (0V)	1	1	RUN output
CW pulse/pulse output (0V)	2	2	CW pulse/pulse output
CCW pulse/direction output (0V)	3	3	CCW pulse/direction output
Error counter clear output (0V)	4	4	Error counter clear output
Zero marker pulse input (0V)	5	5	Zero marker pulse input (DC5V)
Zero marker pulse input (0V)	6	6	Zero marker pulse input (DC12/24V)
Ongin position input	7	7	Ongin position input(DC12/24V)
HOLD input	8	8	HOLD input (DC12/24V)
Emergency stop input	9	9	Emergency stop input (DC12/24V)
CW side overtravel limit SW input	10	10	CW side overtravel limit SW input (DC12/24V)
CCW side overtravel limit SW input	11	11	CCW side overtravel limit SW input (DC12/24V)
Extemal power supply (0V)	12	12	Extemal power supply (DC12/24V)

NOTE
▼△▼

Connector on the Module:
FCN-365P024-AU (made by Fujitsu)
Cables side connectors
Soldering type (standard attached)
Connector FCN-361J024-AU (made by Fujitsu)
Connector cover FCN-360Co24-E (made by Fujitsu)

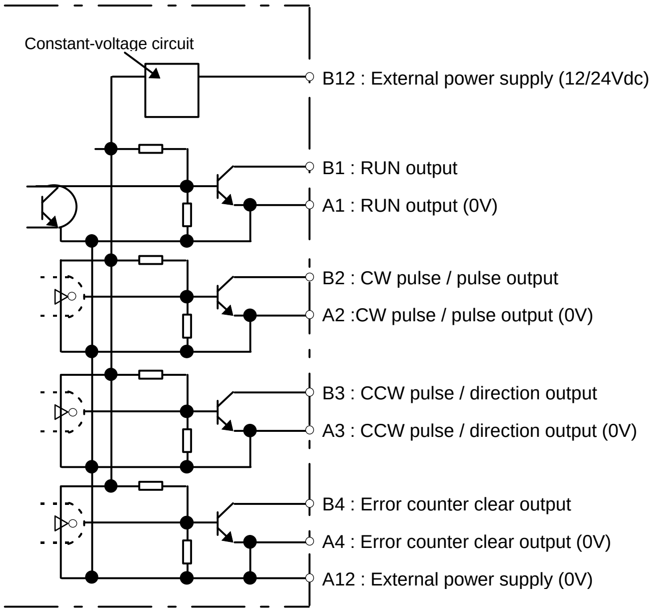
Single-Axis Position Control (Continued)

Input Circuit


Zero marker pulse input (5Vdc) : B5
Zero marker pulse input (0V) : A5
Zero marker pulse input (12/24Vdc) : B6
Zero marker pulse input (0V) : A6
Origin position input (12/24Vdc)
B7
A7
HOLD input (12/24Vdc)
B8
A8
Emergency stop input (12/24Vdc)
B9
A9
CW overtravel limit SW input (12/24Vdc)
B10
A10
CCW overtravel limit SW input (12/24Vdc)
B11
A11
Normally Open-Contact Use
Normally Closed-Contact Use

Note : For the Zero marker pulse input use only one of 5Vdc or 12/24Vdc.

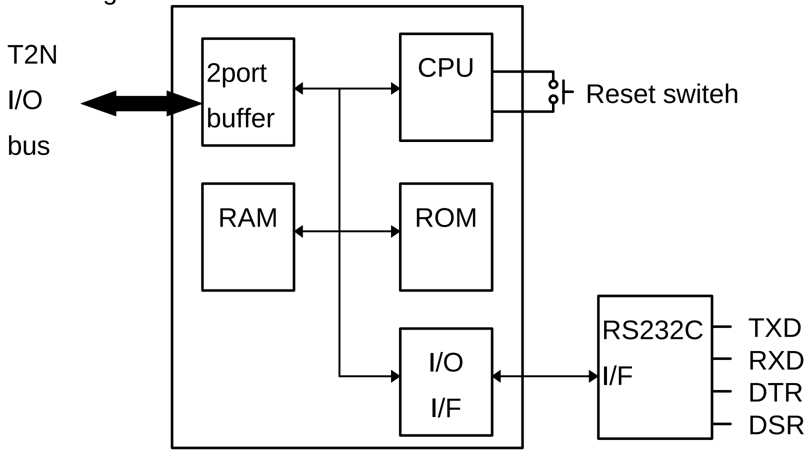
Output Circuit


Constant-voltage circuit
B12 : External power supply (12/24Vdc)
B1 : RUN output
A1 : RUN output (0V)
B2 : CW pulse / pulse output
A2 :CW pulse / pulse output (0V)
B3 : CCW pulse / direction output
A3 : CCW pulse / direction output (0V)
B4 : Error counter clear output
A4 : Error counter clear output (0V)
A12 : External power supply (0V)

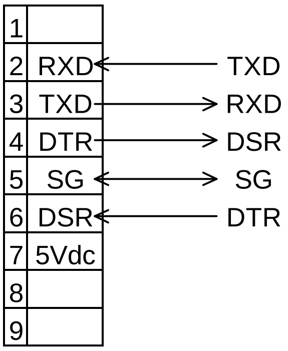
Serial Communication Interface

Item	CF211 (TCF211**S)
Common memory	160 words × 2
Transmission mode	Full-duplex
Synchronizing	asynchrounus (Start-stop method)
Interface	Conforms to RS232C 1CH
Transmission Code	ASCII
Frame Format	Start bit : 1bit Data : 7 or 8bits Parity : even/odd/none When none parity is selected,the data bit length must be 8 bits. Stop bit : 1 or 2bit When 2 stop bits is selected,the data bit length must be 7 bits.
Transmission Speed	300,600,1200,2400,4800,9600,19200bps
I/O occupancy points	i X+Y 4W
Data exchange system	By READ/WRITE instructions of the T2E
LED Display	Transmission data
Isolation	none(between RS232C Port and internal circuit)
Current Consumption	550mA or less

Circuit Configuration



External Connections



D-SUB 9pin(female)

3.1 Input Modules Application Precautions

(1) Minimum ON/OFF time of input signal

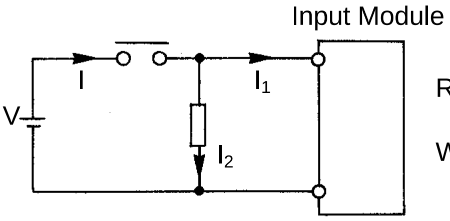
The conditions for guaranteed reading of the ON/OFF states of the input signal are:

Input ON time ≧ ON delay time + 1 scan time

Input OFF time ≧ OFF delay time + 1 scan time

Therefore, be sure to use longer times than these for the ON/OFF times of the input signal.

(2) There are some contacts for which the reliability of contact cannot be guaranteed at the specified input current, depending on the contacts. In such cases, install an external bleeder resistor and pass a dummy current.

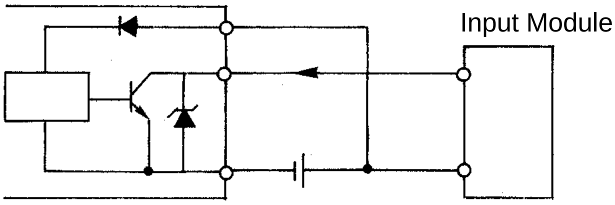


$$R=\frac{V}{I-I_1}$$

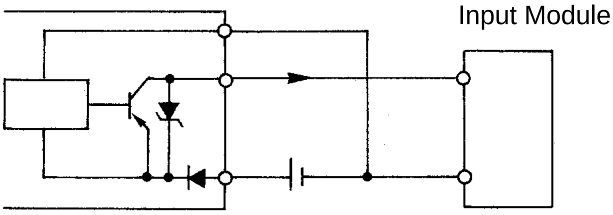
$$\text{Wattage } P> \frac{V^2}{R} \times 3$$

(3) The following are examples of connection with transistor output equipment (such as proximity switches).

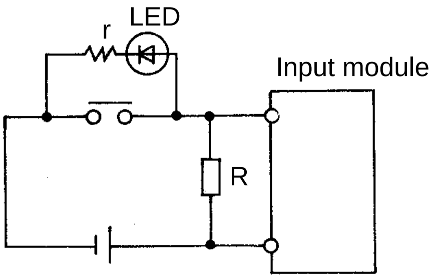
- NPN open collector type (+common)



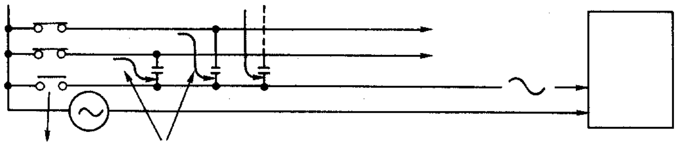
- PNP open collector type (-common)



(4) When using a switch with an LED, if the LED-lighting current flows even when the switch is OFF, it sometimes cannot be recognised as OFF. In this case install bleeder resistor R and decrease the input impedance.



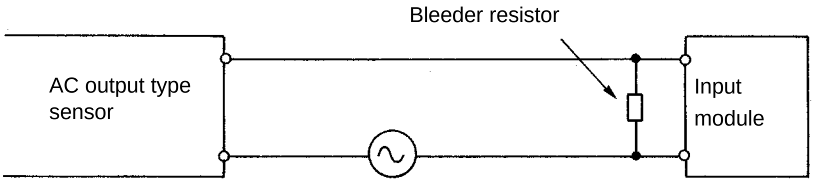
(5) When applying an AC input signal, if the external cables are long or if the number of cores of a multi-core cable is large, a current induced from the charged wires will flow in the open wire, depending on the mutual capacitance between the cables. Sometimes a voltage may be generated which reaches the ON level despite the contacts being open. In this case, the general method is to decrease the input impedance and lower the input ON level due to the induced current. Install a resistor or a resistor + capacitor between input and common, or use a multi-core shielded cable with a small cable capacitance.



Contacts open Current induced from charged wires flow through cable capacitance

Current induced from charged wires flows through cable capacitance
For the maximum distance of external cable to an AC input module, it is necessary either to take the above countermeasure within 100m length in the condition that, out of 20 cores, 19 are charged wires and 1 is an open wire, or to check whether the input voltage in this condition is less than the OFF voltage. When handling large numbers of AC input signals, precautions such as the above are required. Therefore, taking account of cost of the system as a whole, one method is to study the interface by DC signals.

(6) When connecting AC output type sensors, sometimes it is not possible to detect the OFF state due to the leak current when the sensor is OFF.
In this case, counter by installing an external bleeder resistor as shown in the drawing below.



Select bleeder resistors using the following points as a guide.

1) When the sensor is OFF, the voltage between the input terminals must be less than the residual voltage in the OFF state
2) The current when the sensor is ON within the allowable value for the sensor
3) Determine the wattage of the bleeder resistor by making an allowance of approximately 3 times the current when the sensor is ON.

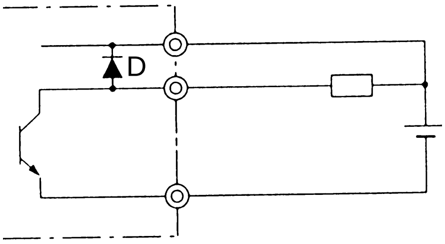
3.2 DC Output Module Application Precautions

(1) The DC output module needs an external power supply to drive output transistors. For each common, connect the load power to the approprite terminal.(For details, see 2.3 I/O module specifications)
If the wrong polarity of the power supply to the terminal is connected,the module will be damaged. Check the polarity before connection.

(2) Protection coordination against over-current of DC output module

Type of module	Protection
DO31 (16-point output)	A fuse of 5A per common(16 points) is built in this DC output module. For an overload and load short-circuit, the transistor will not be protected. This fuse,however,protects the DC output module and the external cable from burn-out.
DO32 (32-point output)	A fuse of 2A per common(8 points) is built in this DC output module. For an overload and load short-circuit, the transistor will not be protected. This fuse,however,protects the DC output module and the external cable from burn-out.
DO235 (64-point output)	The load short-circuit,etc. will cause burn-out of the module and external cable,because a protection fuse is not built in this DC output module. Therfore, install an appropriate fuse on the outside to prevent accidental burn-out.

(3) A diode as shown in the figure below is built in to protect the transistor from transient overvoltage.

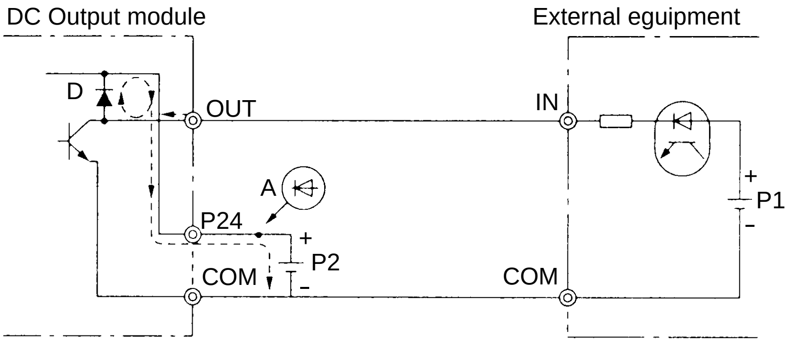


D: Bypasses transient overvoltage to the power supply and suppresses the voltage between the collector and emitter of the transistor.

(4) Pay attention to the following cases caused by the diode mentioned in (3).

〈Case1〉

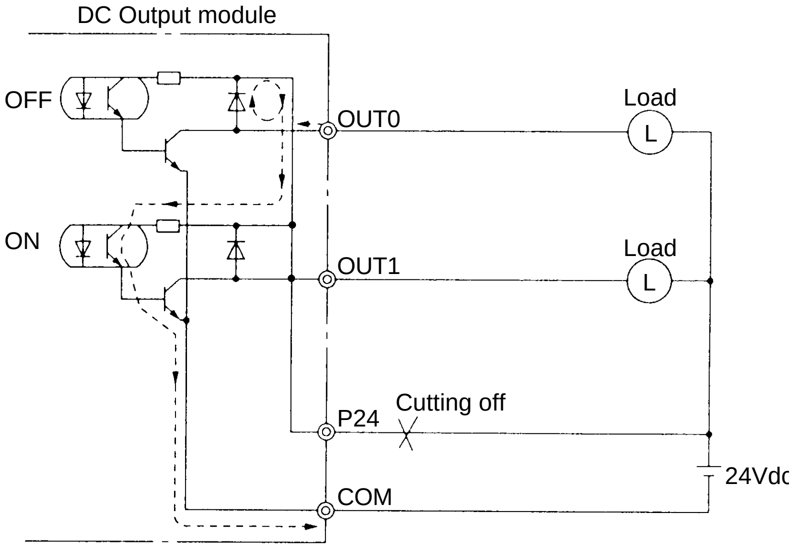
When connecting to the external equipment in which DC power supply is provided from internal,if the voltage of P1 is higher than one of P2 or if P2 turns to OFF when P1 is ON, the external equipment may turn to ON in spite of the output status.
Because a load current flows through D:a diode.



A power supply should be basically used for P1 and P2.
In the above case,insert a diode for preventing a back current in the position A so that a back current is prevented from flowing into P2.

〈Case2〉

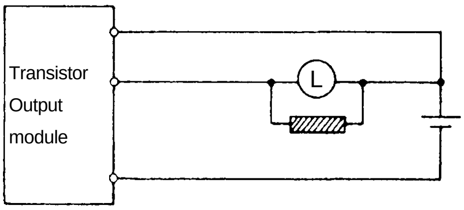
When the power cable connected to the P24 terminal is cut off, external load may be driven because the load current of OFF output circuit(OUT0) flows through the diode and the transistor of ON output circuit(OUT1) as shown below.



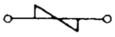
Pay attention for preventing a P24 line from cutting off.

(5) If a capacitive load is connected, rush current will flow when output is charged to ON.
At that time, necessary measures must be taken to protect the output transistor from being destroyed by the rush current.
To limit the rush current there are two effective measures. One is to connect a resistor to the load in series. The other is to apply dummy current to the load by conncting a resistor between the output terminals.

(6) If an inductive load is connected, transient overvoltage will occur when the output is changed to OFF.
This surge voltage will be absorbed into the diode D mentioned before so that the transistor will be protected. However,if the output cable is installed closely to other signal lines, the surge voltage may cause other problem. In this case, install a flywheel diode in parallel with the inductive load (as near as possible to the load).



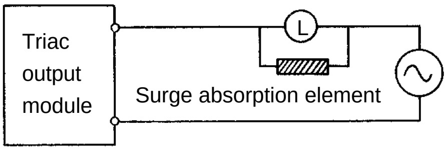
A suitable surge absorption element should be selected according to the application.

Flywheel diode (for voltage clamping)		Peak inverse voltage: 3 times the power supply voltage or more Forward current: Load current or more
Varistor (for voltage clamping)		Rated voltage about twice the maximum(peak) power supply voltage.
Snubber(CR) voltage circuit(for high frequency attenuation)		R:0.5 - 1Ω per 1V coil voltage C:0.5 - 1μF per 1A coil current (Non-polar capacitor)

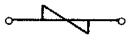
3.3 Triac Output Module Application Precautions

(1) Over-Current Protection Coordination
One 2A fuse per 4 output points is mounted in the triac output module. Although, taking account of protecting elements by the fuse blowing even in load short circuits, when the fuse blows the semiconductors are subjected to considerable damage. Therefore, take care in handling and wiring so that short circuits will not occur.

(2) Output Surge Protection



A suitable surge absorption element should be selected according to the application.

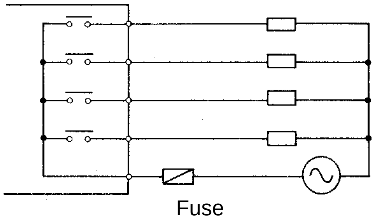
1. Varistor (for voltage clamping) (peak) voltage		Rated voltage about 1.2 times the maximum power supply
2. Snubber (CR) circuit (for high-frequency attenuation)		R : 0.5-1Ω per 1V coil voltage C : 0.5-1µF per 1A coil current (Non-polar capacitor)

3.4 Relay Output Module Application Precautions

(1) It is necessary to supply a +24V power supply to the internal control circuit of the relay output module. Therefore, connect a 24Vdc + 10% power supply between the + and - terminals.

(2) No overload protection fuse is built into the relay output module. Therefore, always install a fuse suited to the current capacity.

12-point relay output module

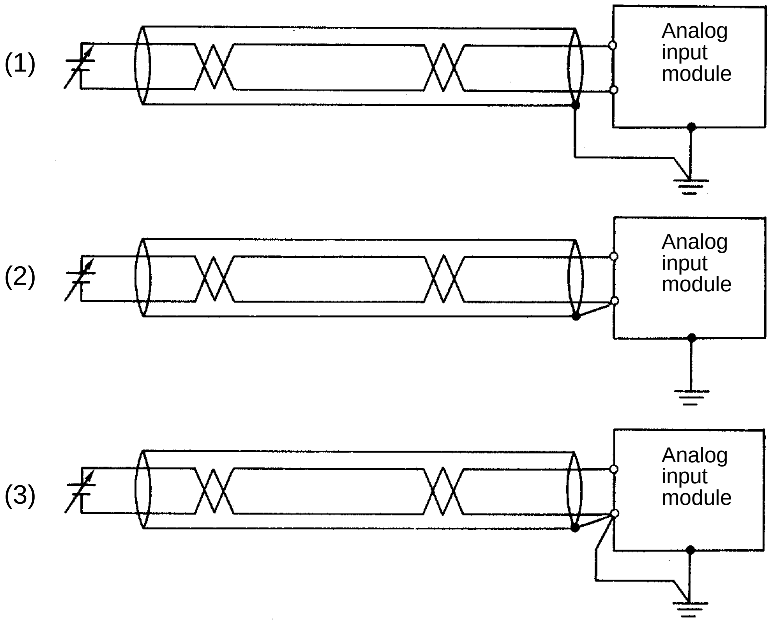


NOTE
▼△▼
Please note that, if it is not protected with a fuse, the module pattern will burn out when there is a load short-circuit.

(3) Output Surge Protection
The installation of a surge absorption element for the induced load, as described in the paragraphs on the transistor output module and the triac output module, is recommended.

3.5 Analog Input Module Application Precautions

(1) Use a shielded twisted-pair cable for the analog input signal line, and wire over the minimum distance. Carry out the grounding of the cable shield on the analog input side(the T2N side).(1) in the dwawing below is the basic. Sometimes, operation is more stable if the wiring is as in (2)or(3).

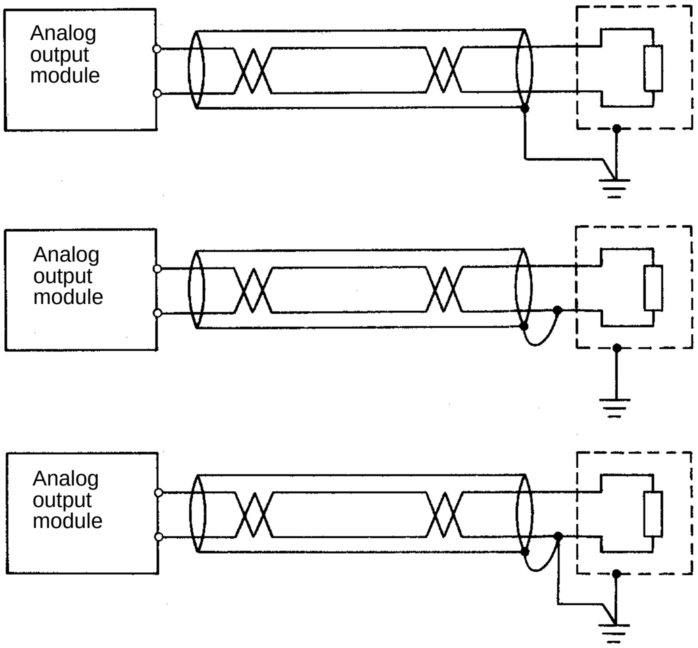


(2) Sometimes the conversion values are unstable,depending on the voltage state of the external 24Vdc power supply.
If the conversion result is not stable, make the external power supply for analog use a dedicated power supply.
Use of the 24Vdc external supply power source of the T2N power supply module is recommended.

(3) All intrinsically shielded cables are fitted with ferrite choke adjacent to analog inputs/outputs, and must have ground connections to grounded metalwork within 5cm for applying to the EMC Directive.

3.6 Analog Output Module Application Precaution

(1) Use a shielded twisted-pair cable for the analog output signal line, and wire over the minimum distance. Carry out the grounding of the cable shield on the load side. (1) in the drawing below is the basic.
Sometimes, operation is more stable if the wiring is as in (2) or (3).



(2) Sometimes the conversion values are unstable, depending on the voltage state of the external 24Vdc power supply.
If the conversion result is not stable, make the external power supply for analog use a dedicated power supply.
Use of the 24Vdc external supply power source of the T2N power supply module is recommended.

(3) All intrinsically shielded cables are occasionally fitted with ferrite choke adjacent to analog inputs/outputs, must have ground connections to grounded metalwork within 5cm for applying to the EMC directive.

4.1 Operating Enviroment

When installing the T2N,avoid the following locations.

(1) Where the ambient temperature exceeds the 0-55°C range.

(2) Where the relative humidity exceeds the 20-90% range.

(3) Where there is condensation due to sharp temperature variations.

(4) Locations subject to vibration in excess of the permissible value.

(5) Locations subject to shock in excess of the permissible value.

(6) Where there are corrosive gases or flammable gases.

(7) Where there is dust,salinity or iron particles.

(8) Locations exposed to direct sunlight.

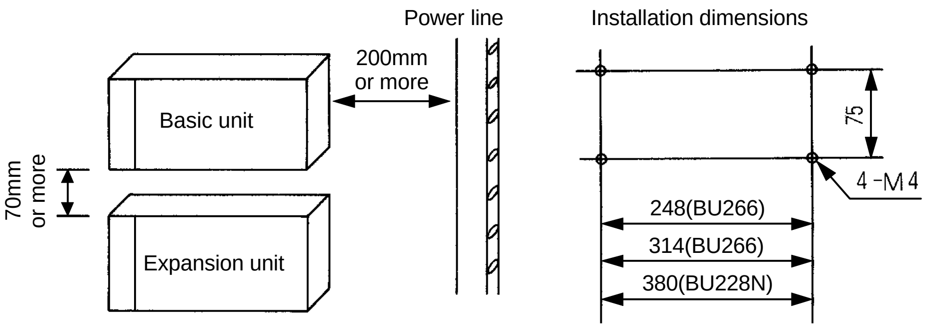
When installing the panel which houses the T2N, take note of following items.

(1) Install as far away as possible from high-voltage panels and power panels.(200mm or more)

(2) When there are high-frequency machines or equipment,securely ground the housing panel.

(3) When using the same channel base as other panels,ensure there are no leakage current from the other panels and equipment.

4.2 Installing the Rack



Installation Precautions

(1) Since the T2N is not of dust-proof construction, install it in a dust-proof control panel.

(2) Avoid installing the units directly above equipment which generates large amounts of heat (such as heaters, transformers and large capacity resistors).

(3) Taking account of safety in maintenance and operation, either isolate at least 200mm from high-voltage equipment and power equipment, or separate by shielding, such as steel plate.

(4) Separate at least 200mm from high-voltage lines and high power lines.

(5) For ventilation, leave an air space of at least 70mm around the units.

(6) In paticular, in the vicinity of high-voltage and power equipment, it is necessary to give consideration to grounding.(See 4.5 Grounding)

(7) In the units, the power supply modules are always positioned on the left-hand side. Install them vertically on the mounting frame.
Do not install them in the direction except it.

(8) Mount the units securely, using the rack mounting screws of M4 size.
(Screws torque : approx.1.47N · m=15kgf · cm)

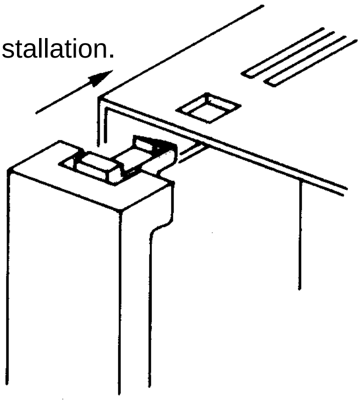
4.3 Mounting the Modules

Always mount the power supply module in the left end slot of the rack.
Also, mount the CPU module in the slot next to the power supply module of the basic unit.

Execute the following procedure for module installation.

(1) Taking care that they are securely inserted in the slots of the base unit, install modules in sequence starting with the power source module at the left end.

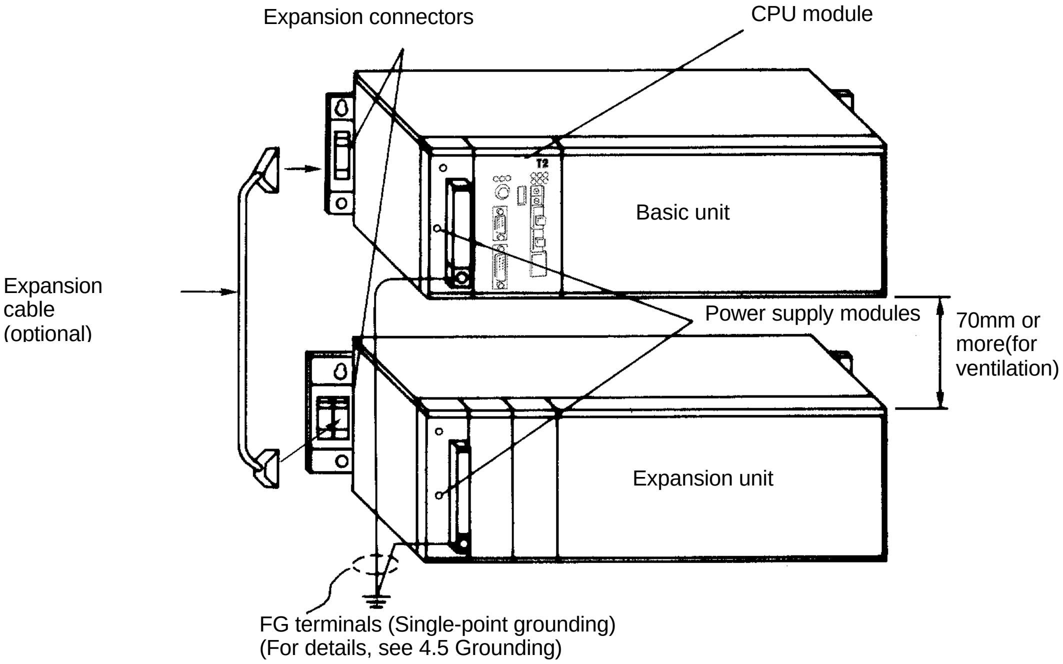
(2) Push securely until the front panel of the module clicks into the base unit.



> NOTE
> ▼△▼
>
> 1.For safety, always turn power OFF before installing and removing modules or installing and removing terminal blocks.
>
> 2.After installing the modules, secure the unit so that it can be mounted vertically, including when in transport.

4.4 Connecting the Expansion Unit

Up to three expansion units (8-slot or 6-slot) can be connected in the T2N.
For the expansion units, use BU268 (8-slot) or BU266 (6-slot).


Expansion connectors
CPU module
T2
Basic unit
Expansion cable (optional)
Power supply modules
70mm or more(for ventilation)
Expansion unit
FG terminals (Single-point grounding)
(For details, see 4.5 Grounding)

> NOTE
> ▼△▼
>
> 2 expansion connectors are fitted on the 5-slot I/O type (BU266) and the 7-slot I/O type (BU268) racks. The right-hand connector is for input from the previous unit, the left-hand is for output to the next unit.

NOTE ▼△▼

1.Separate the expansion cables as far as possible from other cables. In particular, isolate them at least 200mm from power lines.

2.4 types of expansion cables are available - 0.3m, 0.5m, 0.7m and 1.5m. Select according to the positions of the units.

4.5 Grounding

Grounding Point

It is advisable, for the grounding of electronic devices to carry out dedicated grounding which is isolated from that of power systems, and to carry out single-point grounding between 2 or more electronic devices. In the T2N, noise-proofing is designed which takes the actual application into account, and it has a satisfactory noise-immunity without carrying out grounding of the device itself. However, as a precaution, correct grounding is recommended from the viewpoint of reliability.

Carrying out grounding, check against the following criteria.

(1) The electronic equipment case must not become a path for a ground current. (High-frequency currents are particularly harmful)

(2) Equalise the ground potentials when 2 or more units of electronic equipment are to be connected. (Single-point grounding is best)

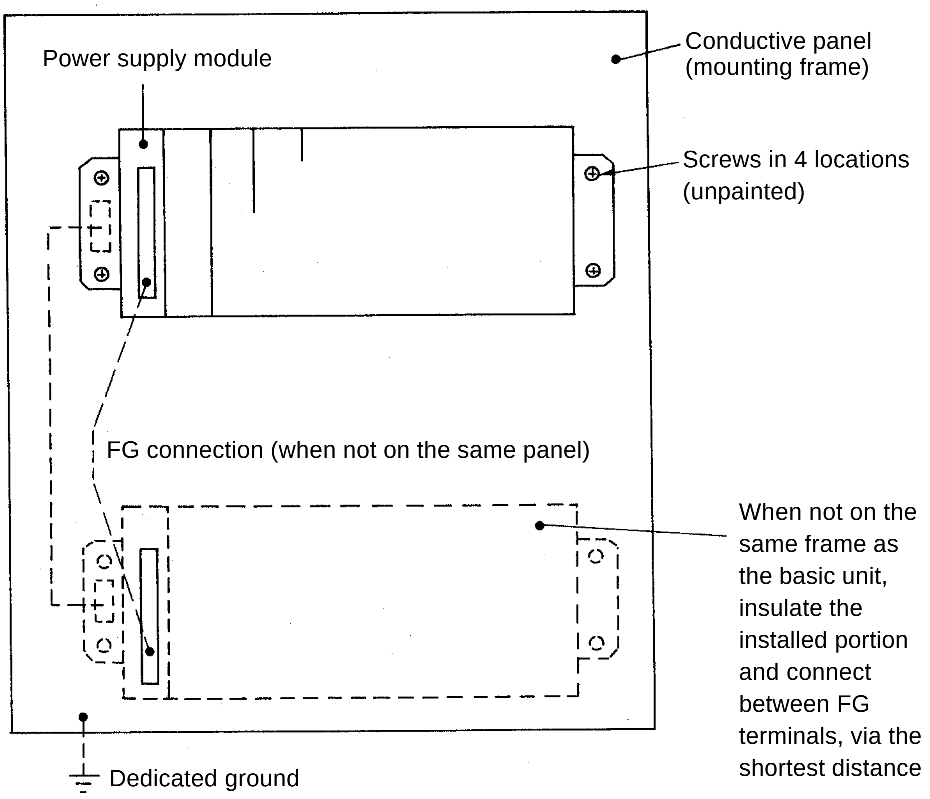
(3) Do not connect to power system earths. (High-frequency isolation is necessary)

(4) Do not connect to unstable earths. (parts with unstable impedance such as painted screws, and parts subject to vibration)

4.6 Grounding Methods

(1) Installation of a Conductive Panel.

When the mounting frame itself has good conductivity, and is not in common use with the earths of other power systems, proceed as below.

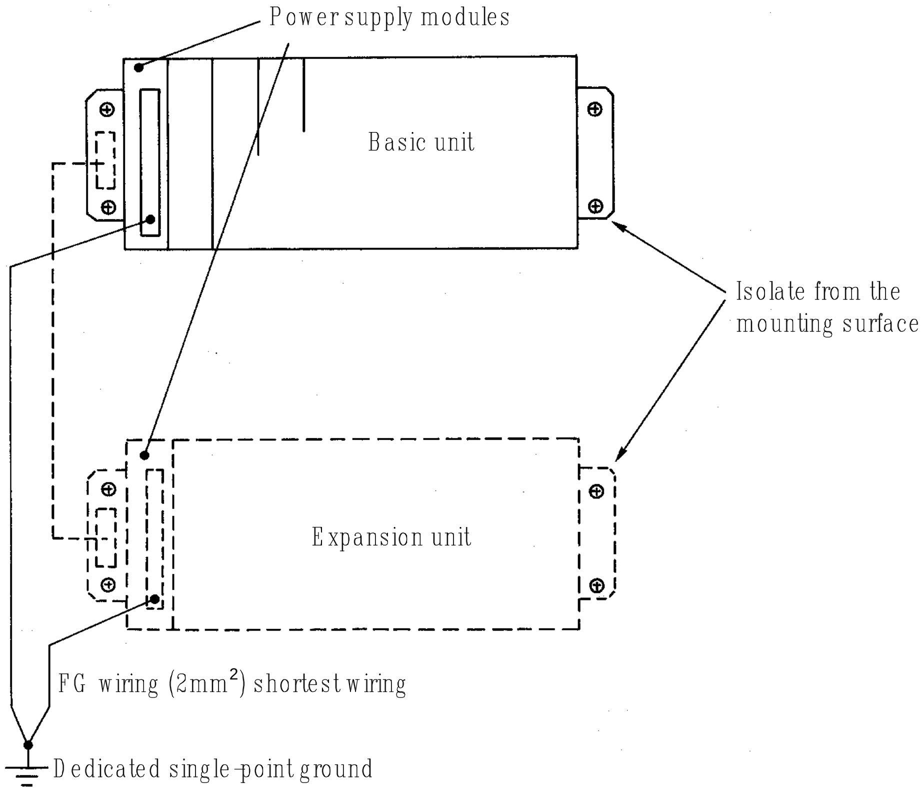

Power supply module
Conductive panel (mounting frame)
Screws in 4 locations (unpainted)
FG connection (when not on the same panel)
When not on the same frame as the basic unit, insulate the installed portion and connect between FG terminals, via the shortest distance
Dedicated ground

NOTE ▼△▼

When the mounting frame does not possess good conductivity, or when the frameground is a power system earth or is an unstable ground, use the insulated installation shown in (2).

(2) Isolated Installation

When the mounting frame has an unstable earth potential, or when it is not conductive, securely mount the unit with isolation, as shown below.


Power supply modules
Basic unit
Isolate from the mounting surface
Expansion unit
FG wiring (2mm²) shortest wiring
Dedicated single-point ground

(3) When there is no grounding point

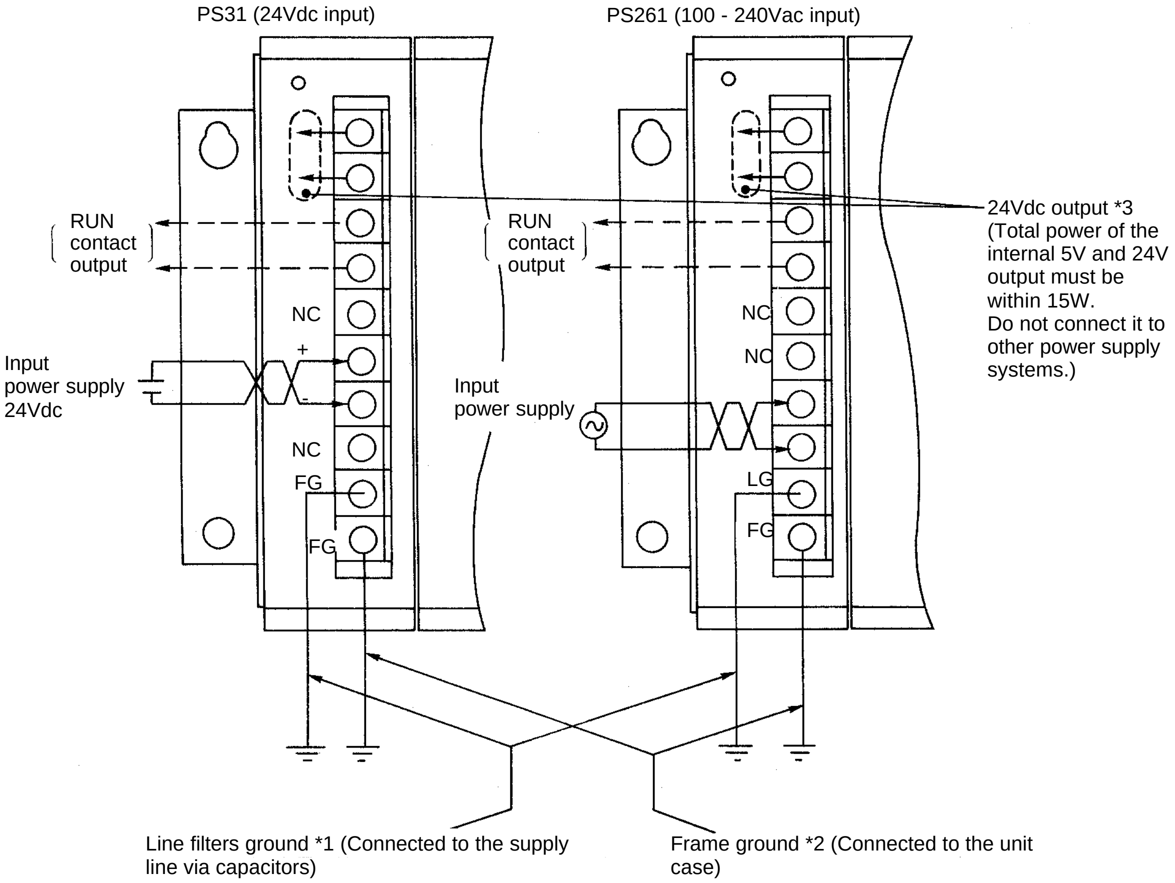
When suitable grounding is not available (no dedicated grounding point), mount by the method of (2) Isolated Installation.

There is no problem with operation if there is mutual connection between the FG terminals of the T2N itself. However, for safety, carry out single-point grounding via an impedance* close to the frame.

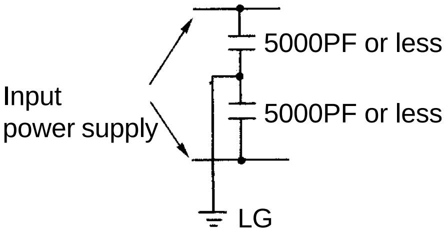
*; When a resistor	:	Ground the frame via 1W-1Ω approx.
When an inductance	:	Ground to the frame via 2A-100μH approx.

4.7 Wiring the Power Supply

Wire the external power supply to the T2N power supply module in the following manner.
When using expansion units, arrange for power to be supplied simultaneouly to the basic unit and the expansion units (or to the expansion units before the basic unit).


PS31 (24Vdc input)
PS261 (100 - 240Vac input)
RUN contact output
RUN contact output
NC
+
-
NC
FG
FG
Input power supply 24Vdc
Input power supply
NC
NC
LG
FG
24Vdc output *3 (Total power of the internal 5V and 24V output must be within 15W. Do not connect it to other power supply systems.)
Line filters ground *1 (Connected to the supply line via capacitors)
Frame ground *2 (Connected to the unit case)

*1 : Line Filter Ground(LG)



Normally, the LG and FG terminals are shorted.
However, depending on grounding environment (such as when there is a problem with leakage current or when the power supply ground is separate), open the LG terminal or provide a dedicated ground.

*2 : Frame Gounds(FG)
See 4.5 Grounding for details.

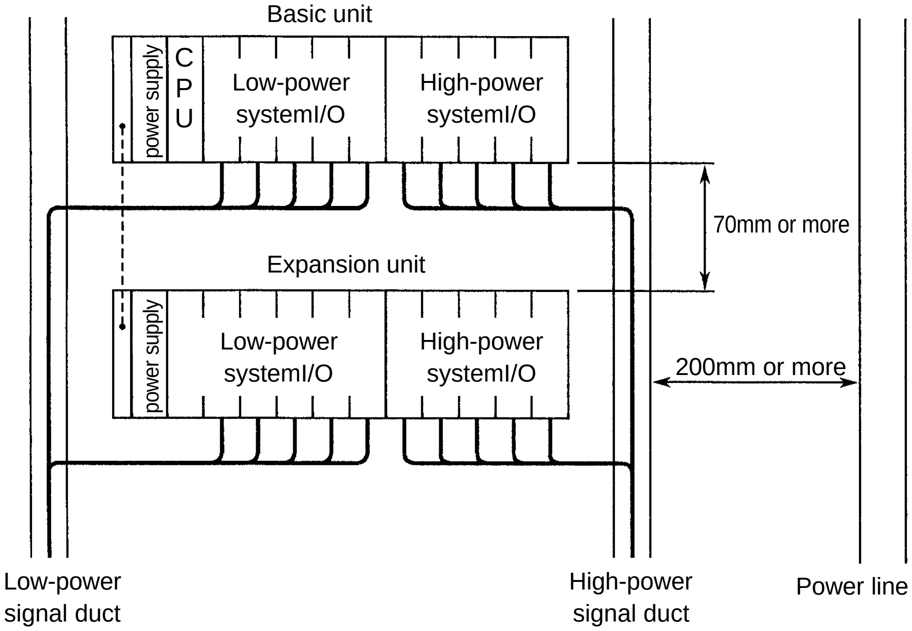
*3 : 24Vdc output
The 24Vdc power cables must be suppressed with ferrite cores, immediately adjacent to the power supply module(s).

NOTE
▼△▼

Use crimp-style terminals with sheaths as far as possible for wiring to the power supply module. When it is not possible to use crimp-style terminals with sheaths, cover with insulating tape so that the conductive parts are not exposed.

4.8 I/O Wiring

Pay attention to the following points when mounting and wiring the I/O modules.



Low-Power System I/O
DC input module
Analog input module
Analog output module
Pulse input module
Positioning module
Serial Interface
Transmission module

High-Power System I/O
AC input module
DC output module
AC output module
Relay output module

(1) For the positioning of the I/O modules, arrange the low-power system I/O to the left and high-power system I/O to the right, and keep the wiring separate.

(2) The gap between units should be at least 70mm for maintenance and ventilation.

(3) Separate by at least 200mm from power lines and power equipment, or shield with a steel plate (the steel plate must be grounded).

(4) For the sizes of input/output wiring, see the Table below.

Numbers of Points in Modules	Wire Sizes to Use
16-point module	0.5 ~ 1.25mm^2
32/64-point module	0.1 ~ 0.3mm^2

However, for common lines, use a thicker size which takes account of the current capacity. Also, for cables outside the panel, the use of cable of at least 1.25mm^2 is recommended to keep the impedance low.

(5) The terminal screws are M3.5. For suitable crimp-style terminals, use terminals with width 7mm or less for M3.5 screws.

(6) Both inside and outside the panel, always avoid wiring input/output signal lines in bundles with, in proximity, or in parallel with high-voltage lines and power lines. When separation is difficult, use multi-core shielded cable depending on the type of input/output signals, and make a single-point ground for the shield at the service entrance in the panel (in the cases of AC I/O,DC I/O and relay output modules).

(7) Pay attention to 3.Application Precautions for I/O Modules.

⚠ CAUTION

1.Turn off power before wiring to minimize the risk of electrical shock.

2.Use crimp-style terminals with sheaths as far as possible for wiring to the power supply module. When it is not possible to use crimp-style terminals with sheaths, cover with insulating tape so that the conductive parts are
not exposed.

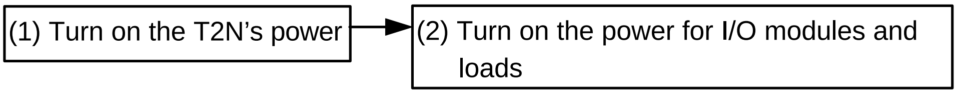
3.Operation without grounding may cause electrical shock or malfunction. Connect the ground terminal on the T2N to the system ground.

4.Apply power of the specified rating.
Applying excess power voltage to the T2N can cause explosion or fire.

4.9 Power up/down Sequence

Turn on power or turn off power of the T2N according to the following sequence so that the T2N is used safely and securely.

(Power up sequence)



(1) Turn on the T2N’s power

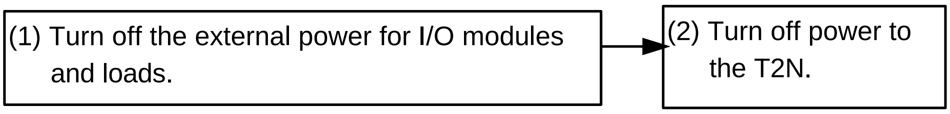
(2) Turn on the power for I/O modules and loads

(1) Turn on power of the T2N at first. When using expansion units, arrange for power to be supplied simultaneously to the basic unit and the expansion units.
Use the same power lines for them.
If it is difficult, turn on power of the expansion units before turing on that of the basic unit.

(2) Turn on power for I/O modules and loads simultaneously.
Use the same power lines for them.

If it is difficult, turn on the external power for I/O modules before turing on the power for the loads.

(Power down sequence)



(1) Turn off the external power for I/O modules and loads.

(2) Turn off power to the T2N.

(1) Turn off the external power for I/O modules and loads simultaneously.
Or turn off power for loads and turn off power for I/O modules in turn.

(2) Turn off power of the T2N.

When using expansion units, turn off power of the basic unit and the expansion units simulataneously.

Or turn off power of the expansion units after turning off power of the basic unit.

⚠ CAUTION

1.Configure the external circuit so that the external power required for output modules and power to the loads are switched off simultaneously. Also, turn off to the loads before turning off power to the T2N.

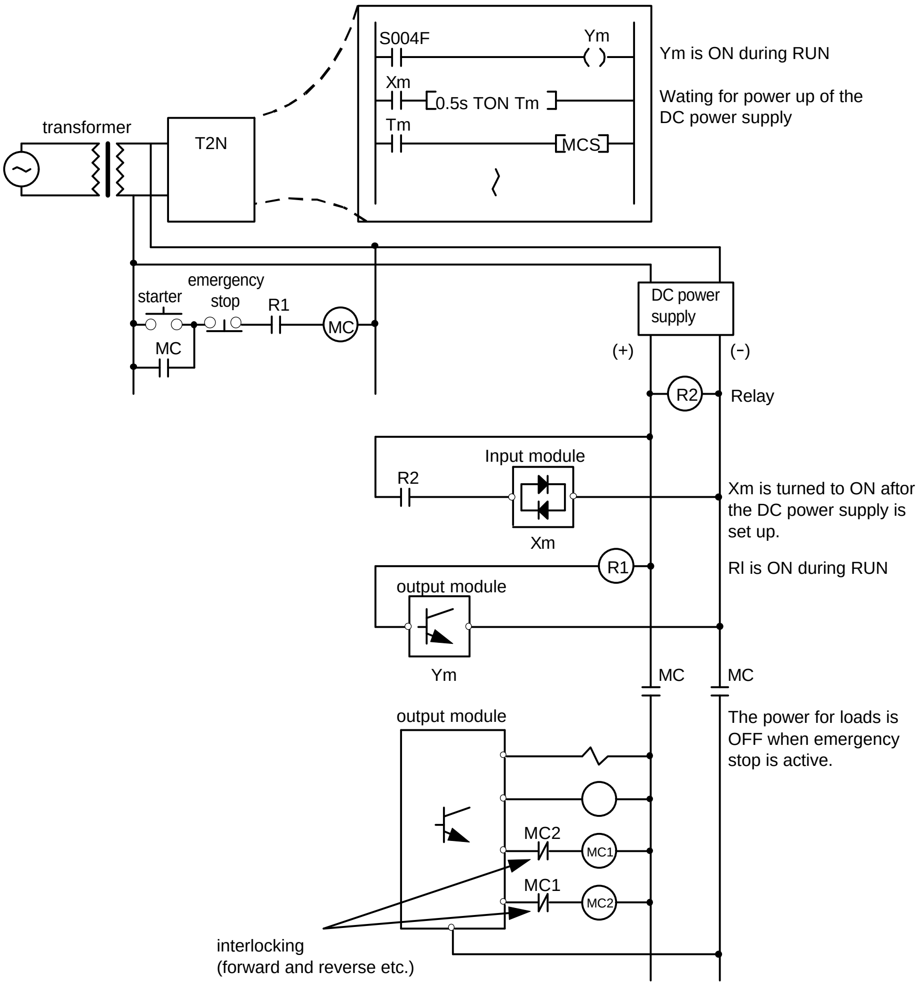
4.10 Safty circuit

Configure emergency stop and safty interlocking circuits out of the T2N against faulty of the T2N or breaking wires.

Emergency stop circuit

Safty interlocking circuit (forward and reverse etc.)

(Example)



Ym is ON during RUN

Wating for power up of the DC power supply

Xm is turned to ON aftor the DC power supply is set up.

Rl is ON during RUN

The power for loads is OFF when emergency stop is active.

interlocking (forward and reverse etc.)

5.1 Daily Checks

Recommended daily checks for optimum system performance

Item	Content of Check	Countermeasure when Abnormal

Check the LEDs on the front of the power supply modules and CPU module	POWER(Red)
	:Lit when 5V power supply normal
	RUN(Green)
	:Lit when operating normally
	FLT(Red)
	:Out when CPU and I/O normal
	BAT(Green)
	:Lit when battery voltage normal
		When the state of an LED is not normal, follow the procedure in 6.Trouble-shooting.

Check the LED displays of the input modules (Digital input)	The corresponding LED should be lit when an external input signal is ON.	•Check whether the input voltage is within the specified value. •Check whether there is any slack in the input terminal block. •Check whether the module is securely mounted.
Check the LED displays of the output modules (Digital output)	When the output is ON, the corresponding LED should be lit and the corresponding external load should operate.	•Check whether the external load voltage is within the specified value •Check the built-in fuse. •Check whether there is any slack in the output terminal block. •Check whether the module is securely mounted.
Check the switch position on the CPU module	Operation is executed when the operation mode switch(HALT/RUN) is to Run.	Put the switch to the specified position.

See Ethernet/TOSLINE-S20LP of the T2N user's manual regarding status display LEDs for built-in networking.

NOTE
▼△▼

When a serious error (such as system RAM abnormal) is detected after power is switched ON, the FAULT LED on the CPU will blink. In this condition, communication with programmer cannnot be executed. When this condition does not change even if the power is switched ON again, exchange CPU modules.

5.2 Periodic Checks

Check the following items periodically (about once every 6 months).Check also when the operating conditions/enviroment change.

Item	Check	Criteria
Power Supply	Power supply voltage (measure at the module's power supply terminals.)	85 ~ 264Vac 20.4 ~ 28.8Vdc
	Is there any slack in the power supply terminal block screws ?	Must not be loose.
	Is there any damage to the wires and cables ?	Must not be damaged.
Mounting Condition	Is the basic unit firmly secured ?	Must not be any slack or play.
	Are the expansion units firmly secured ?	Must not be any slack or play.
	Is each module firmly installed ?	Must not be any slack or play.
	Is there any slack in the expansion cable connctors or any damage to the cables ?	Must not be any slack or damage.
Programmer	Is there any problem with the programmer functions ?	Execute simple operations.
	Is there any slack in the connectors or damage to the cable.	Must not be any slack or damage.
I/O Modules	Measure the voltage at each I/O terminal block.	Must be within the specified values.
	Check the input state display LEDs.	Must light when normal.
	Check the output state display LEDs.	Must light when normal.
	Are the I/O terminal blocks firmly secured ?	Must not be any slack or play.
	Is there any slack in the terminal screws, or is there any risk of adjacent terminals touching each other ?	There must not be any slack or risk of mutual contact.
	Is there any damage to the wires and cables ?	Must not be any damage.
Environment	Check that temperature, humidity, vibration,dust,etc are within the specified values.	Must be within the general specifications.
Program	Check that the contents of the basic program and the master program (kept on a floppy disk or the like) agree.	Contents must agree when carrying out a comparative check
Battery	Does the battery require changing ? (The battery installation date is recorded on the optional card cover).	A change is recommended if 2 years have elapsed.
	Are the battery connctors firmly connected ?	Must not be any slack.

⚠ CAUTION

1. Turn off power before checking voltage on terminals. Failure to do so can cause electrical shock.
2. Do not modify the T2N in hardware nor software. This can cause fire, electrical shock or injury.

5.3 Spare Parts to Keep in Stock

The following items are recommended minimum spares. These will allow operation to be resumed immediately in the event of any failure.

Part	Quantity	Remarks
I/O modules	One of each type used	For the relay contact output, the contact life must be taken into account. See Section 2.3.
Fuses	Number to be used	See Section 5.5.
Batteries	1	For emergency use. See Section 5.4.
CPU modules included with optional cards	1	Keep a minimum of 1 each to reduce down-time to the minimum.
Power supply modules	1	
Programmer	1set	Useful for detecting the cause of a failure.
Master programs	As required	Keep on FD or the like.

⚠ CAUTION

Do not touch activated terminals of I/O modules and units. Keep the terminal covers closed during power ON.
This can cause electrical shock or injury.

NOTE

▼△▼

Store batteries in a cool (25°C or less) place as self-descharge is greater at high temperatures.

5.4 Battery Replacement

Normally the program and 2kwords data register are kept in the built-in EEPROM of the T2N. On the other hand, other retentive registers' value and real time clock's value can be kept in the RAM with a battery.

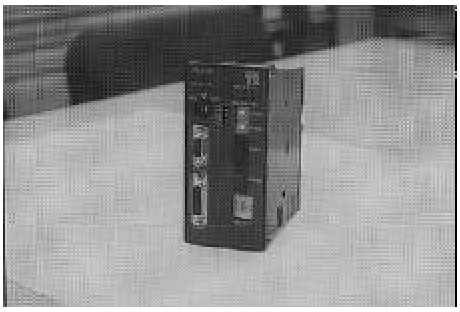
A lithium battery is used, which has little self-discharge and can be used over a wide temperature range. Therefore, during its period of use, and in paticular when the non-conducting time is long, it can be used with assurance.

The date of istallation of the battery is recorded on the CPU side panel.

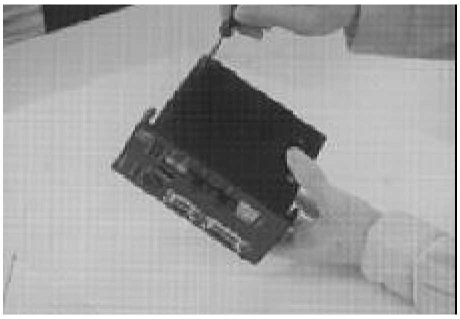
Under normal use, it is recommended that the battery should be replaced every 2 years. Check the date of installation and replace using the following procedure.

There is an LED(BAT) which indicates normal battery voltage on the front of the CPU module. This is lit when normal. When this LED flashes or is out, the battery comes into life. Therefore replace the battery within 14days. (It is recommended that the power supply should not be switched OFF until actual replacement, in order to protect the program.)

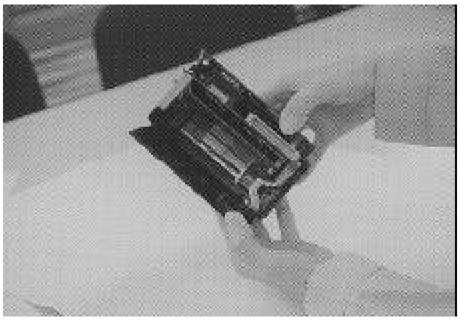
Remove the CPU module from the basic unit after turning power OFF.



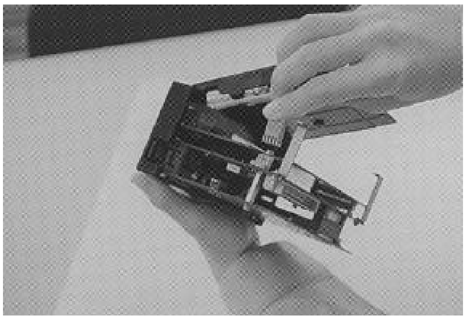
Loosen the screws of the side panel which mounts a battery with CPU module.



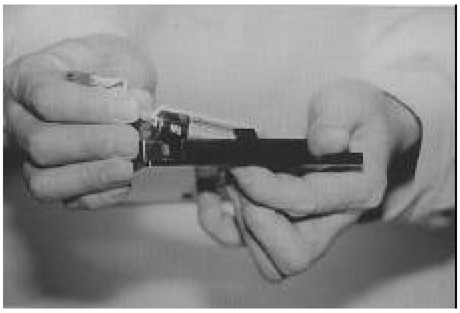
Remove the CPU side panel.



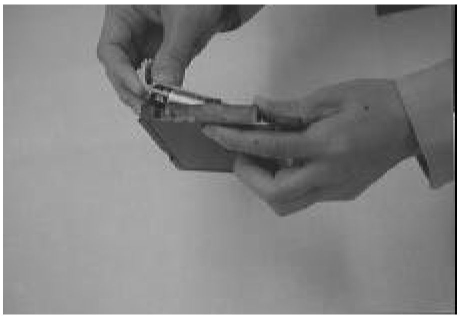
Pull off a connector of the battery.



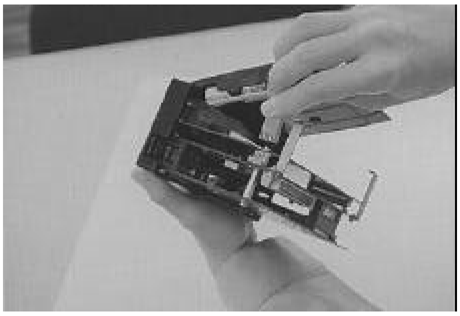
Inserting a finger into the hole under battery holder, take off a battery like pushing out.



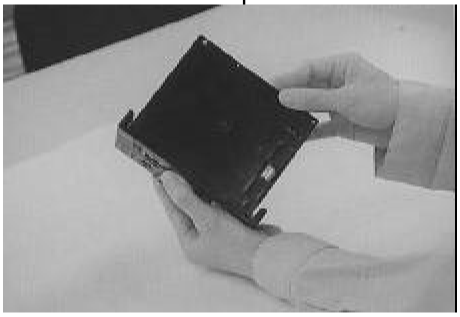
Insert a new battery into the holder, supporting it from the hole by a finger.



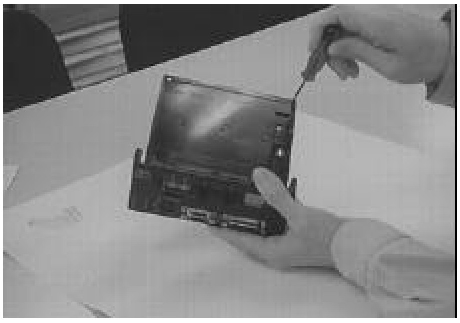
Connect the new battery's connector.



Corresponding the side panel's holes for screws with position of metal plates, connect CPU module and side panel.



By screwing, connect both CPU module and side panel.



- Disposal of the battery

Dispose of the battery in the same way as general-purpose dry batteries.
There is a risk of explosion if dismantled or burnt.
If the + and - of a lithium battery are allowed to short, this may lead to igniting and fumes.
Don not cut the lead wires and do not dismantle the connectors.

NOTE

▼△▼

1.The battery should be replaced with power OFF. The time with battery removed should be within 5 minutes. If the battery is kept removed for a long time, the contents of the RAM memory will be erased, so please take care.

2.When connector of the battery is disconnected, an LED (BAT) is out. Any error message about the battery can’t be registered in the Event History Table in this case.

3.When handling the battery, take care of the following points.

* The voltage is not compatible with manganese dry batteries and alkali batteries. Do not use these as substitutes.
* Never let the + and - of the battery be shorted.
* Never dismantle batteries, overheat them or put them into a fire.
* Never try to charge a battery. This is not possible.

4.Do not use a battery which has been stocked more than 3 years since manufactured date.

5.The battery is a dedicated product with lead wires and connectors attached. Order it from Toshiba.(Product Code:EX25SER6)

5.5 Fuse Replacement

The following fuses are used in the T2N modules.

These fuses are recommended minimum spares and will allow operation to be resumed immediately in the event of any failure.

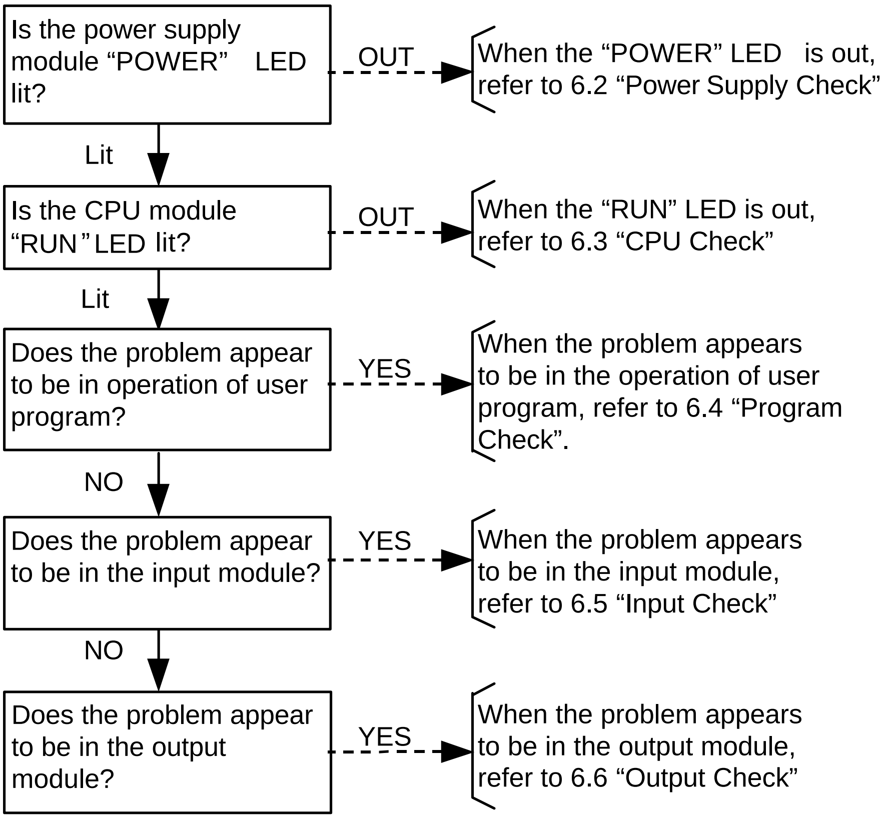
Module		Fuse Rating	Model	Quantity
Power Supply	PS31	Glass tube 125V-2A(normal fusion)	EX10*SFB20	1
	PS261	Glass tube 250V-3A(normal fusion)	TFU923*AS	1
Output	DO31	Glass tube 250V-5A(quick fusion)	EX10*SFA50	1
	DO32	Glass tube 250V-2A(quick fusion)	EX10*SFA20	4
	AC61	Glass tube 250V-2A(normal fusion)	EX10*SFC20	3

6.1 Troubleshooting Procedure

When a problem occurs in the system, having first thoroughly understood the content of the problem, it is important to determine whether the cause lies on the mechanical side or on the control system (PLC) side.

Also, the cause of one problem frequently gives rise to secondary problems. Therefore it is important clearly to determine the cause of the problem by considering the system as a whole.

When the problem is considered to be in the T2N itself or in the input/ output of the T2E, first check the following items.


Is the power supply module "POWER" LED lit?
OUT
When the "POWER" LED is out, refer to 6.2 "Power Supply Check"
Lit
Is the CPU module "RUN" LED lit?
OUT
When the "RUN" LED is out, refer to 6.3 "CPU Check"
Lit
Does the problem appear to be in operation of user program?
YES
When the problem appears to be in the operation of user program, refer to 6.4 "Program Check".
NO
Does the problem appear to be in the input module?
YES
When the problem appears to be in the input module, refer to 6.5 "Input Check"
NO
Does the problem appear to be in the output module?
YES
When the problem appears to be in the output module, refer to 6.6 "Output Check"

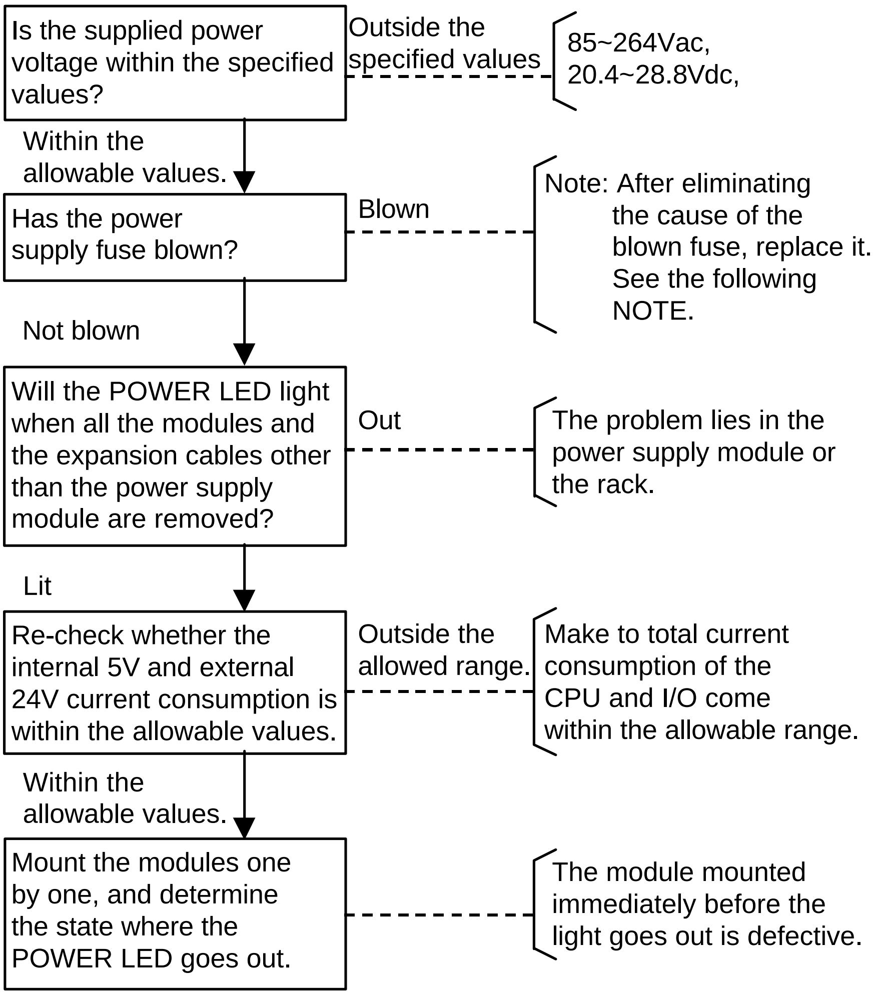
When the problem is temporary, and when the problem occurs with the synchronisation of system/mechanical operations, the influence of the external environment (such as noise and power fluctuations) may be considered to be the cause. Since the items to check in this case are collated in paragraph 6.7, carry out a check referring to that paragraph.

NOTE ▼△▼

When the cause cannot be determined by the above checks, consult Toshiba.

6.2 Power Supply Check

The following is a flow-chart of checks for use when the POWER LED does not light even when the power to the T2N is switched ON, or when a power supply failure occurs after some specified time.


Is the supplied power voltage within the specified values?
Outside the specified values
85~264Vac, 20.4~28.8Vdc,
Within the allowable values.
Has the power supply fuse blown?
Blown
Note: After eliminating the cause of the blown fuse, replace it. See the following NOTE.
Not blown
Will the POWER LED light when all the modules and the expansion cables other than the power supply module are removed?
Out
The problem lies in the power supply module or the rack.
Lit
Re-check whether the internal 5V and external 24V current consumption is within the allowable values.
Outside the allowed range.
Make to total current consumption of the CPU and I/O come within the allowable range.
Within the allowable values.
Mount the modules one by one, and determine the state where the POWER LED goes out.
The module mounted immediately before the light goes out is defective.

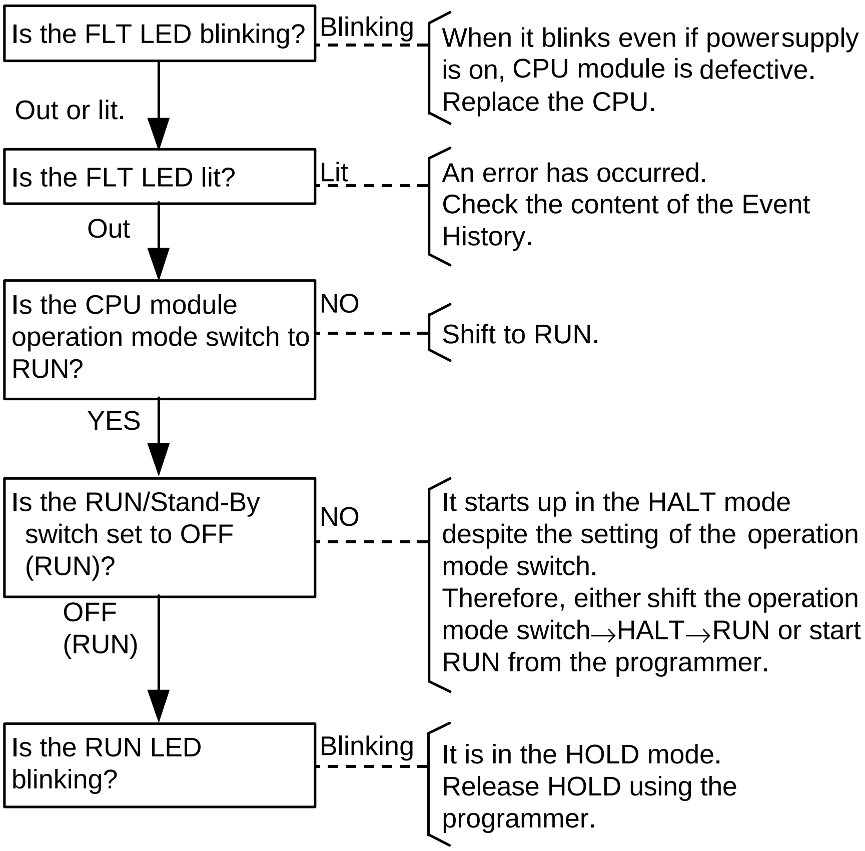
NOTE
▼△▼

1.When carrying out the above checks, always check each step after switching the power supply OFF again.

2.When a fuse has blown, always determine the cause of the blown fuse and eliminate it. If the fuse is replaced and the power supply is switched ON again without eliminating the cause, there is a risk of progressive damage to the module.
When the cause of the blown fuse cannot be determined, consult Toshiba without replacing the fuse.

6.3 CPU Check

When the “POWER” LED of the power supply module is lit, but the “RUN” LED of the CPU module is out, check the following items.


Is the FLT LED blinking?
Blinking
When it blinks even if power supply is on, CPU module is defective. Replace the CPU.
Out or lit.
Is the FLT LED lit?
Lit
An error has occurred. Check the content of the Event History.
Out
Is the CPU module operation mode switch to RUN?
NO
Shift to RUN.
YES
Is the RUN/Stand-By switch set to OFF (RUN)?
NO
It starts up in the HALT mode despite the setting of the operation mode switch. Therefore, either shift the operation mode switch→HALT→RUN or start RUN from the programmer.
OFF (RUN)
Is the RUN LED blinking?
Blinking
It is in the HOLD mode. Release HOLD using the programmer.

6.4 Program Check

When the control operation does not operate properly although the program is executed, check the following items.

(1) Is there an output to the same coil or register at 2 or more places in 1 scan, or, is there an overlap of the device for coil instruction and function block instruction?

(2) Is there an attempt to input a signal which changes faster than the scanning cycle?

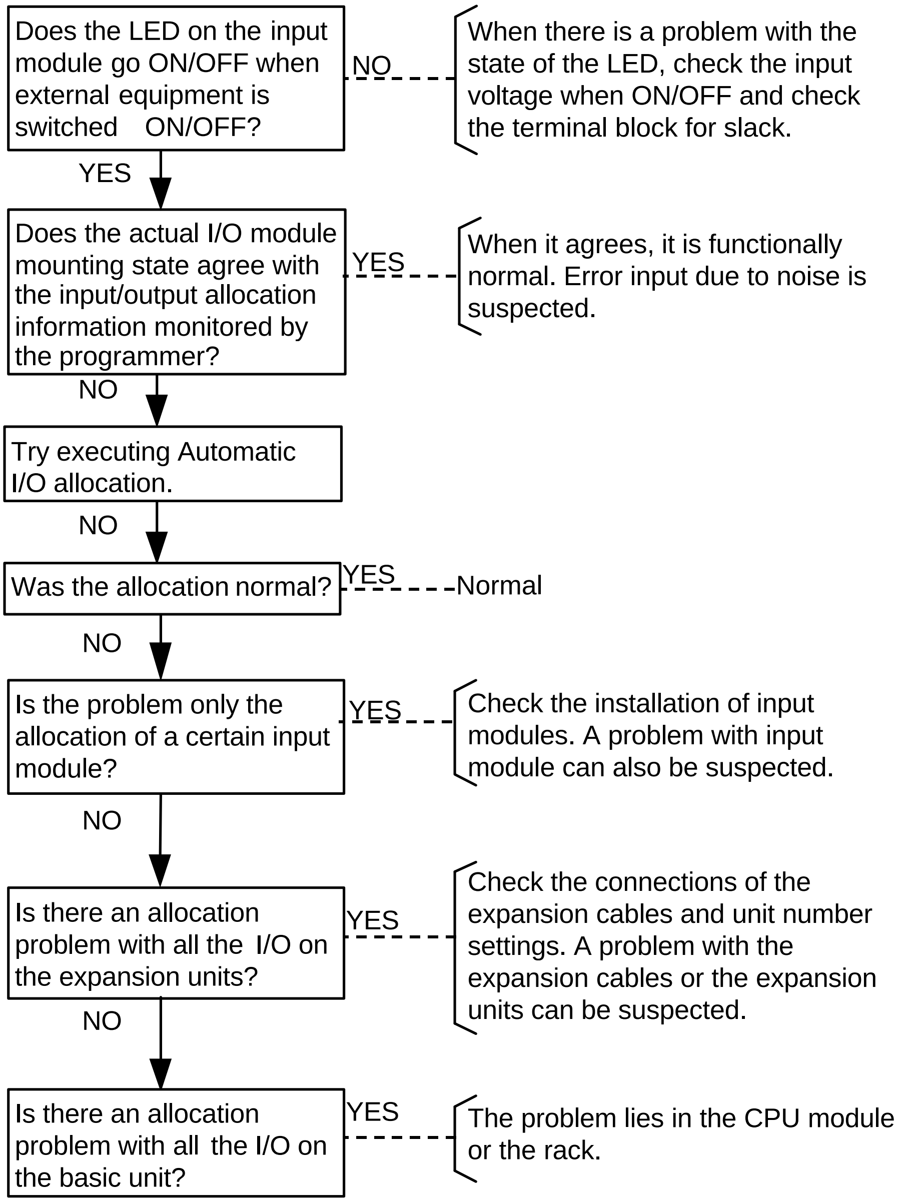
(3) Is the same timer register or counter register being used for multiple timer instructions or counter instructions?

(4) When interrupt is in use, is a device or register operating during the interrupt program which affects the operation of the main program?

(5) Is any EEPROM error occured?(S0039, S0051 is 1.)
In this case, carry out EEPROM write command by the programmer.

6.5 Input Check

When unable to read the input signal correctly although the program is being executed, check the following items.



Does the LED on the input module go ON/OFF when external equipment is switched ON/OFF?

NO – When there is a problem with the state of the LED, check the input voltage when ON/OFF and check the terminal block for slack.

YES

Does the actual I/O module mounting state agree with the input/output allocation information monitored by the programmer?

YES – When it agrees, it is functionally normal. Error input due to noise is suspected.

NO

Try executing Automatic I/O allocation.

NO

Was the allocation normal?

YES – Normal

NO

Is the problem only the allocation of a certain input module?

YES – Check the installation of input modules. A problem with input module can also be suspected.

NO

Is there an allocation problem with all the I/O on the expansion units?

YES – Check the connections of the expansion cables and unit number settings. A problem with the expansion cables or the expansion units can be suspected.

NO

Is there an allocation problem with all the I/O on the basic unit?

YES – The problem lies in the CPU module or the rack.

6.6 Output Check

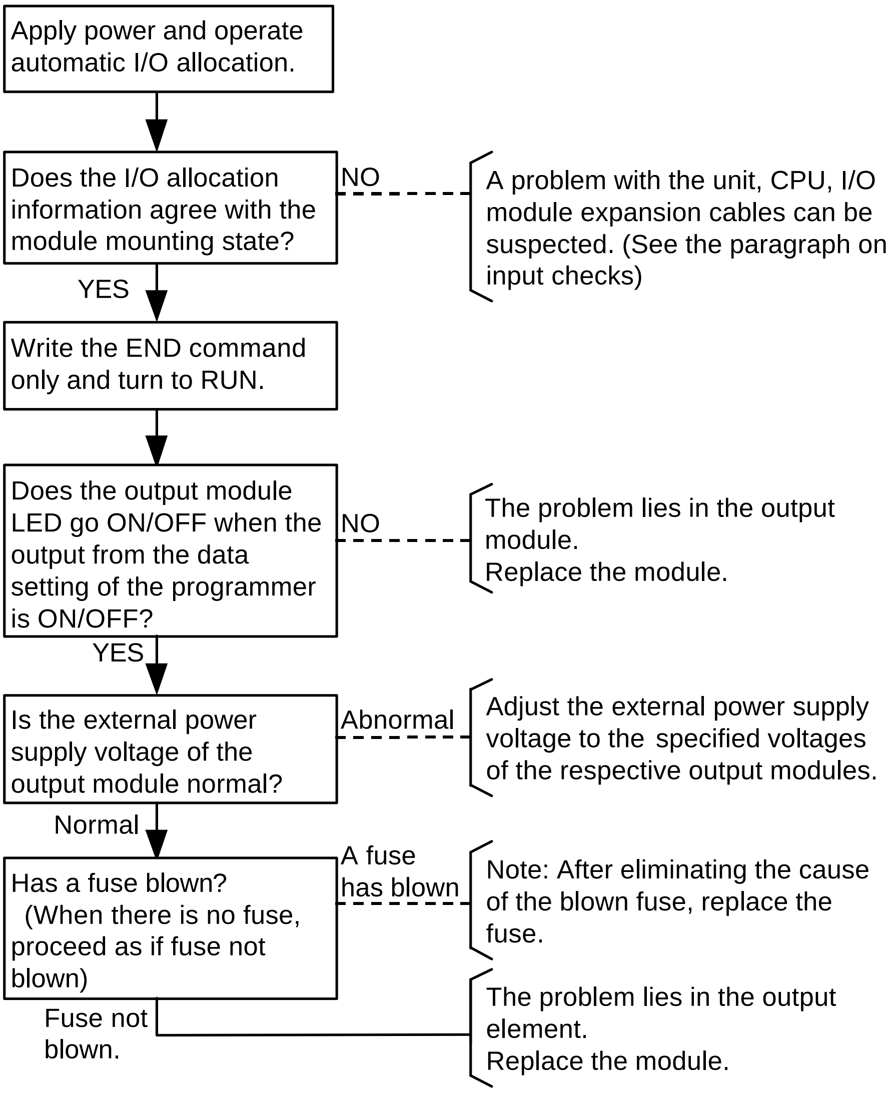
When there is a problem with the actual operation of output equipment although correctly outputting to registers and devices on the program, carry out the following in preparation for checks.

(1) Save the program. (In a floppy disk, EEPROM, etc)

(2) Clear the CPU memory.

(3) Put the ROM/RAM shift switch to RAM (ON).

After making the above preparations, carry out checks in the following sequence.


Apply power and operate automatic I/O allocation.
Does the I/O allocation information agree with the module mounting state?
NO
A problem with the unit, CPU, I/O module expansion cables can be suspected. (See the paragraph on input checks)
YES
Write the END command only and turn to RUN.
Does the output module LED go ON/OFF when the output from the data setting of the programmer is ON/OFF?
NO
The problem lies in the output module. Replace the module.
YES
Is the external power supply voltage of the output module normal?
Abnormal
Adjust the external power supply voltage to the specified voltages of the respective output modules.
Normal
Has a fuse blown? (When there is no fuse, proceed as if fuse not blown)
A fuse has blown
Note: After eliminating the cause of the blown fuse, replace the fuse.
Fuse not blown.
The problem lies in the output element. Replace the module.

NOTE ▼△▼

When a fuse blows, always investigate and eliminate the cause. If the fuse is replaced without eliminating the cause and the power supply is switched ON again, there is a risk of progressive damage to the module. When the cause of the blown fuse cannot be determined, consult Toshiba without replacing the fuse.

6.7 Faults Due to External Problems

When a problem with the T2N system occurs as one of the following phenomena, external factors should be suspected.

(1) When the problem occurs in synchronisation with the operation of input/output equipment
In this case, there is a possibility that the cause is noise generated when the output equipment switches ON/OFF. Apply the noise counter-measures described in Section 3 Application Precautions for I/O Modules.

(2) When the problem occurs in synchronisation with the operation of peripheral power equipment and high-frequency equipment
In this case, the effect of noise induced in the input/output signal lines may be suspected. Also, depending on the power supply system and the grounding system, the cause may sometimes be surges or voltage fluctuations in the power supply and fluctuations in the ground potential. Check with the Notes described in Section 4 Installation and Wiring. Depending on the case, one method is to try the effect of disconnecting the ground.

(3) When the problem occurs in synchronisation with the operation of machinery, the effect of vibration may be considered. Check the state of installation of units/modules and, at the same time, study vibration countermeasures, such as the use of vibration-proofing rubber.

(4) When similar problems re-occur even after replacing faulty modules, thoroughly check that there is no risk of entry of metal particles or drops of water.

Apart from the above causes, if, for instance, the ambient temperature exceeds the specified range, stable operation of the system cannot be guaranteed. Take thorough precautions over the environmental conditions.

6.8 List of Items for Self-Diagnostic Check

When the T2N CPU has detected a problem through self-diagnosis, it registers in the Event History Table one of the error messages (and associated information) shown in the Table on the following pages. When the details of the problem are such that it is not possible to continue operation, the CPU switches all the outputs to OFF, and stops the operation. (Error Down)

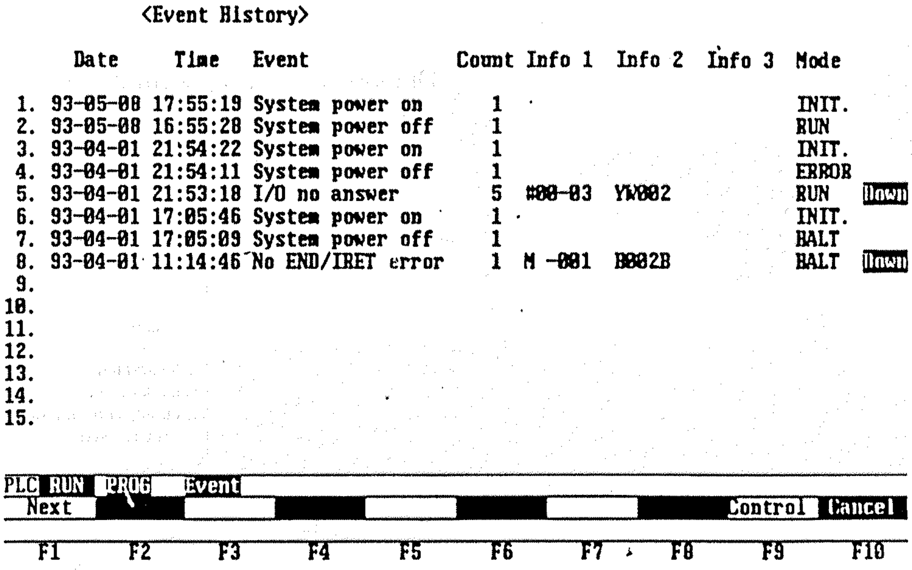
The latest 30 error messages and the times of their occurrence are stored in the Event History Table, and these can be displayed on the programmer. The times when any error were occured, can be recorded while the RAM and calendar are maintained by a capacitor and battery in the T2N. (Power supply ON/OFF can also be registered)

When the T2N system has been stopped by Error Down, first connect the programmer and make it display the Event History Table, then check the details of the error.

The following is the procedure for making the programmer display the Event History.

(1) Connect the T2N CPU module and the programmer (T-PDS) by a dedicated cable.
(2) Switch ON the power supply of the programmer (T-PDS). (The power supply of the T2N system should also be ON)
(3) Start up T-PDS by keying-in TPDS [Enter] from the programmer (T-PDS).
(4) If some key (any key) is pressed, the T-PDS initial menu screen will be displayed. At this time, “Receive Time Out” should not be displayed.
(5) In this state, if S and E are keyed-in, the Event History will be displayed.

(Example of Event History display screen)



```
           <Event History>

     Date      Time    Event                Count Info 1  Info 2  Info 3  Mode

 1. 93-05-08 17:55:19 System power on         1                           INIT.
 2. 93-05-08 16:55:28 System power off        1                           RUN
 3. 93-04-01 21:54:22 System power on         1                           INIT.
 4. 93-04-01 21:54:11 System power off        1                           ERROR
 5. 93-04-01 21:53:18 I/O no answer           5   #00-03  YW002           RUN   Down
 6. 93-04-01 17:05:46 System power on         1                           INIT.
 7. 93-04-01 17:05:09 System power off        1                           HALT
 8. 93-04-01 11:14:46 No END/IRET error       1   M -001  B002B           HALT  Down
 9.
10.
11.
12.
13.
14.
15.

PLC RUN PROG Event
Next                                                          Control  Cancel
 F1     F2     F3     F4     F5     F6     F7     F8     F9     F10
```

*)The Event History can be registered even if initial set of the calendar is not executed. In this case, the Date and Time displays will be shown as “??-??-?? ??:??:??”.

When “Receive Time Out” is displayed in Step (4) above, communication between the programmer and the T2N system has not been established. When the FLT LED on the CPU module is blinking, there is a malfunction in the CPU module. When this state does not change even if the power supply of the T2N system is switched ON again, replace the CPU module. When “Receive Time Out” is displayed in states without FLT blinking, check the Connection method of the setup options of the programmer (T-PDS) and the connection state of the connector cable. When there is no problem with the environmental setting or the connector cable, a malfunction of communication circuit in the T2N system or the programmer is suspected.

When the Event History has been displayed, check the registered error message (“Event”) (No.1 is the latest registered details).

On the next and following pages, the error messages and associated information registered in the Event History, related special relays, LED display states after the event and their meaning are collated. When an error occurs, check its cause and take the necessary steps.

In the “Error Down” state, operations such as program correction will not be possible. Therefore, carry out operations such as correction after executing “Error Reset” from the programmer. In order to start up RUN again, either shift the operation mode switch to RUN after first shifting it to HALT, or execute the Operation command from the programmer.

NOTE
▼△▼

If RUN is started in the state in which the ROM/RAM shift switch has been set to ROM (other than when the operation mode switch is P-RUN), the contents of the EEPROM will be transferred to the RAM memory, and any corrected contents of the RAM memory will be erased. Therefore, when setting to RUN after correcting the program in order to check its operation, start up RUN after executing “program write” by a T-PDS Memory Management menu.

In the Table on the next and following pages, the meanings of the symbols for the system LED displays are as follows:-

● :Lit
○ :Out
◑ :Blinking
— :No effect on state.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
Power Supply	System power on							Power supply ON (No error)
	System power off							Power supply OFF (No error)
Memory	RAM check error	Generated address	Error data	Test data	S0004 S0012	○	●	A fault has been detected by a read/write check of the user data memory (RAM). When the state does not change through switching ON the power supply again, replace the CPU module.
	Program BCC error	BCC error data			S0006 S0030	○	●	A fault has been detected by a BCC check of the user program memory (RAM) After executing Memory Clear, re-load the program.
	Batt voltage drop				S000F	—	—	A voltage drop has been found in the RAM memory back-up battery when the power supply is ON. (BATT LED out. No Error Down.) Replace the battery.
	EEPROM BCC error	BCC error data			S0004 S0013	○	●	A BCC fault has been detected in the user program in the EEPROM when transferring from EEPROM to RAM (when carrying out Inital Load, etc). (Transfer not executed). After checking the program, rewrite to EEPROM
	EEPROM warning	Number of times of writing exceeded			S0007	—	—	Writing to the EEPROM has exceeded life (100,000 times). (No Error Down). Hereafter, the possibility of an EEPROM write fault occurring is high. Therefore replace CPU module.
	EEPROM write error				S0039	—	—	Any error has been occured during writing data to an EEPROM. (Included with operation of XFER instruction) Carry out EEPROM write command by the programmer again.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
CPU	Sys RAM check err	Generated address	Error data	Test data		○	◑	A fault has been detected by a read/write check of the system memory (RAM). When the state is not changed even by switching on the power supply again, replace the CPU module.
	Sys ROM BCC error	BCC error data				○	◑	A fault has been detected by a BCC check of the system ROM. When the state is not changed even by switching on the power supply again, replace the CPU module.
	Peripheral LSI error	Error code				○	◑	A fault has been detected by a check of the peripheral control LSI in the CPU module. When the state is not changed even by switching on the power supply again, replace the CPU module.
	Clock-Calendar error					—	—	A fault has been detected in the built-in calendar LSI data. (No Error Down) When the error is generated even when the calendar is reset, replace the CPU module.
	Illegal sys interrupt	Interrupt generated address 1	Interrupt generated address 2			—	—	An unregistered interrupt request has been received by the CPU module. (No Error Down) If it appears to be generated frequently, replace the CPU module.
	WD timer error	Generated address 1	Generated address 2		S0004 S001F	○	●	A watchdog timer fault has been detected. If it appears to be generated frequently, replace the CPU module.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
I/O	I/O bus error	Unit No.	Data		S0005 S0020	○	●	A fault has been detected by an I/O bus check. Remove all the I/O modules and switch ON the power supply again. When an error is generated even so, replace in the sequence rack CPU. When the error is restored by switching ON the power supply again, switch the power supply OFF and insert I/O modules one by one, switching the power supply ON each time. Replace the I/O module which generated the error.
	I/O mismatch	Unit No. -Slot No.	Register		S0005 S0021	○	●	The input/output allocation information and the mounted state of the I/Os differ. Set the input/output allocation correctly.
	I/O no answer	Unit No. -Slot No.	Register		S0005 S0022	○	●	An I/O module has not been mounted in a slot allocated to I/O. Mount an I/O, or start up in the RUN-F (forced operation) mode.
	I/O parity error	Unit No. -Slot No.	Register No.		S0005 S0023	○	●	A parity error has been detected when data is transferred to an I/O module. Check if the I/O modules are installed properly.
	Duplicate I/O reg	Unit No. -Slot No.	Register		S0005 S0021	○	●	A duplication has been detected in the allocation of I/O modules to the input/output register. Re-set so that the unit first register assignment is not duplicated.
	Illegal I/O reg	Unit No. -Slot No.	Register		S0005 S0021	○	●	The allocation of I/O modules to the input/output register has exceeded 128W. Reduce the I/O module allocation.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
Processing	LP function error	Error code	Error data		S0004 S0015	○	●	A fault has been detected in the language processor for (LP). When the state does not change even on starting up again, replace the CPU module.
	LP execution timeout				S0004 S0015	○	●	The operation of the language processor (LP) is not completed within the specified time. When the state does not change even on starting up again, replace the CPU module.
	Scan time over	Scan time			S0006 S0031	○	●	The scan time exceeds 200ms. Shorten the scan time or use the “WDT” instruction
Program	No END /IRET error	Program type - Block No.	Address in block		S0006 S0030	○	●	No "END" instruction has been programmed in the main program or the sub program, or no "IRET" instruction as been programmed in the interrupt program. Insert the "END", or the "IRET" instruction.
	Pair inst error	Program type - Block No.	Address in block		S0006 S0030	○	●	There is a fault in the method of using instruction combinations MCS/R and JCS/R. Check that the MCS/R and JCS/R command combinations are correct.
	Operand	Program type- Block No.	Address in block		S0006 S0030	○	●	There is a fault in the operand assignment for the Coil instruction or the FUN instruction. Check whether an input (X) is allocated to an output operand.
	Invalid program	Program type - Block No.			S0006 S0030	○	●	A fault has been detected in the program control information. After executing Memory Clear, reload the program.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
Program	Jump target error	Program type - Block No.	Address in block	Jump label No.	S0006 S0030	○	●	The “LBL” instruction for the label No. designated by a “JUMP” instruction has not been programmed in the same program type. Or a “LBL” instruction is programmed on a point before by the “JUMP” instruction. (Backward jump) Program the “LBL” instruction in a regular position.
	No sub entry	Program type - Block No.	Address in block	Sub-routine No.	S0006 S0030	○	●	The “SUBR” instruction for the sub-routine No. designated by a “CALL” instruction has not been programmed. Program the “SUBR” Instruction.
	No RET error	Program type - Block No.	Address in block	Sub-routine No.	S0006 S0030	○	●	No “RET” instruction has been programmed in the sub-routine. Program the “RET” instruction
	Sub nesting err	Program type - Block No.	Address in block	Sub-routine No.	S0006 S0030	○	●	Sub-routine nesting has exceeded 6 layers. Alter the program so that sub-routine nesting is 6 layers or less.
	Loop nesting error	Program type - Block No.	Address in block		S0006 S0030	○	●	“FOR”, “NEXT” instruction nesting has exceeded 6 layers. Alter the program so that “FOR”, “NEXT” instruction nesting is 6 layers or less.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
Program	SFC step No. error	Program type - Block No.	Step No.		S0006 S0030	○	●	Is there multiple use of step Nos. in the SFC program, or do the steps No. designated by the initial step and the end step not agree?. Change the step Nos. or check the step No. of the end step.
	SFC marco No. err	Program type - Block No.	Macro No.		S0006 S0030	○	●	There is multiple use of a macro No. Or the same macro program is called in 2 or more places. Change the macro Nos. Or arrange for the macro program to be called in only 1 place.
	No SFC macro entry	Program type - Block No.	Macro No.		S0006 S0030	○	●	The macro program of the macro No. designated by a macro step has not been programmed. Check whether the macro program has been programmed, or whether the macro program No. is not in error.
	SFC jump label err	Program type - Block No.	SFC jump label No.		S0006 S0030	○	●	There is multiple use of an SFC jump label No.Change the SFC jump label No.
	No SFC jump label	Program type - Block No.	SFC label No.		S0006 S0030	○	●	The SFC label instruction for a jump label No. designated by an SFC jump instruction has not been programmed. Program the SFC label instruction.
	Duplicate SFC No.	Program type - Block No.	SFC program No.		S0006 S0030	○	●	There is multiple use of an SFC program No. Change the SFC program No.
	Invalid SFC prog	Program type - Block No.			S0006 S0030	○	●	The initial step/end step or end, or the macro/macro end, do not correspond. Alter the program so that the initial step/end step or end, or the macro/ macro end correspond.

Classification	Error Message and Associated Information				Related Special Relays	CPU LED Display		Meaning of Error and Countermeasures
	Error Message	Information 1	Information 2	Information 3.		RUN	FLT	
Program	Illegal inst	Program type - Block No.	Address in block		S0006 S0060	○	●	An illegal instruction has been detected in a program. After Memory Clear, re-load the program.
	Invalid Fun Inst	Program type - Block No.	Address in block	FUN instruction	S0006 S0030	○	●	An instruction has been detected which is not supported by the T2N. Erase the relevant instruction.
	Boundary error	Program type - Block No.	Address in block	FUN instruction No.	S0064 or S0065	—	—	The index value when qualifying the index by a FUN instruction has exceeded the register No. limit. (No Error Down) Change the program so that the index value comes within the register No. limit.
	Duplicate entry No.	Program type - Block No.	Address in block	Entry No.	S0006 S0030	○	●	There is multiple designation of the entry No. of an LBL instruction and an SUBR instruction. Set the entry Nos. so that there is no overlap.

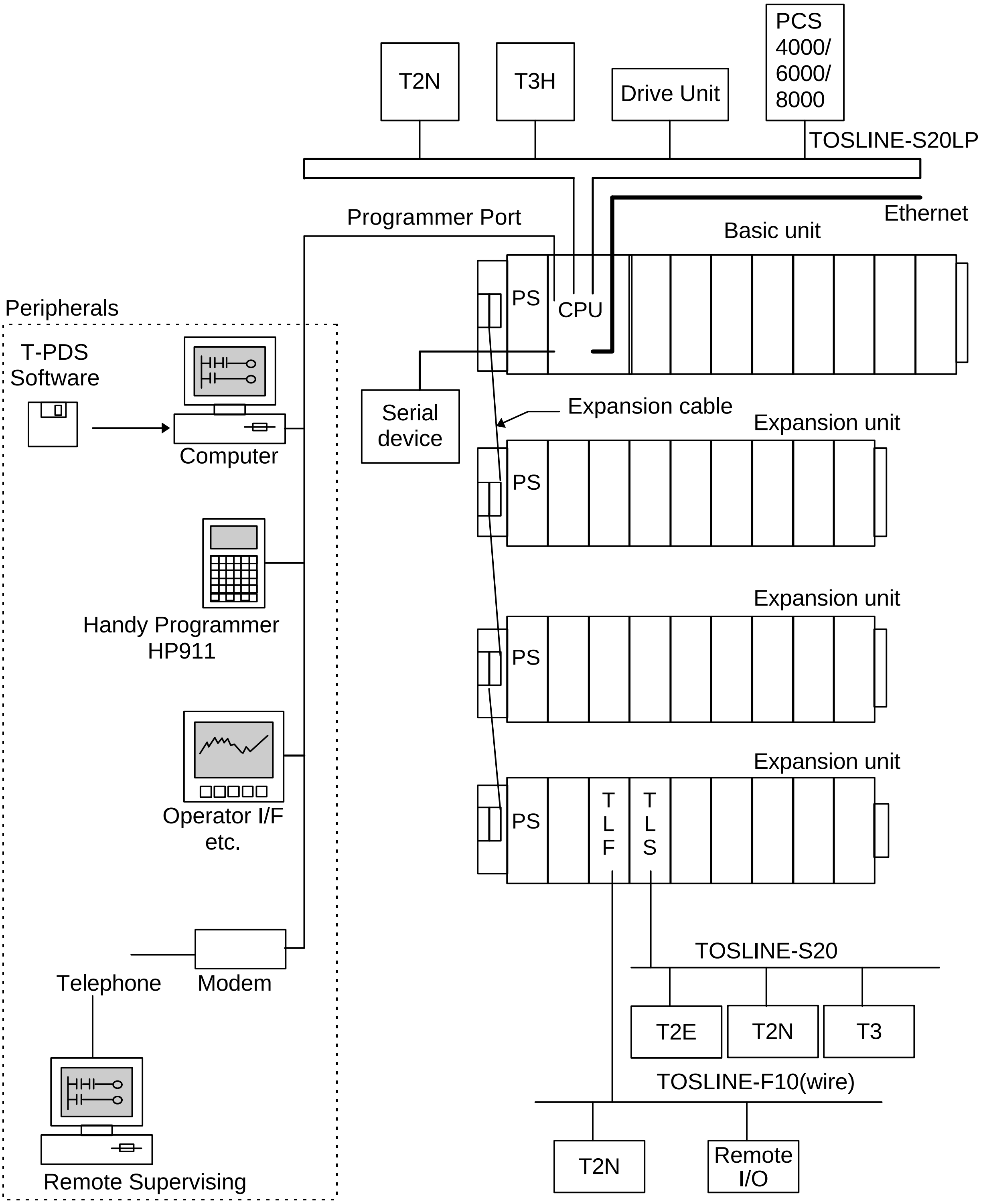
PART2

FUNCTIONS

1.1 T2N System Configuration

The T2N system configuration is shown in the figure below. Part 2 explains the T2N system functions, concentrating on the T2N CPU functions.

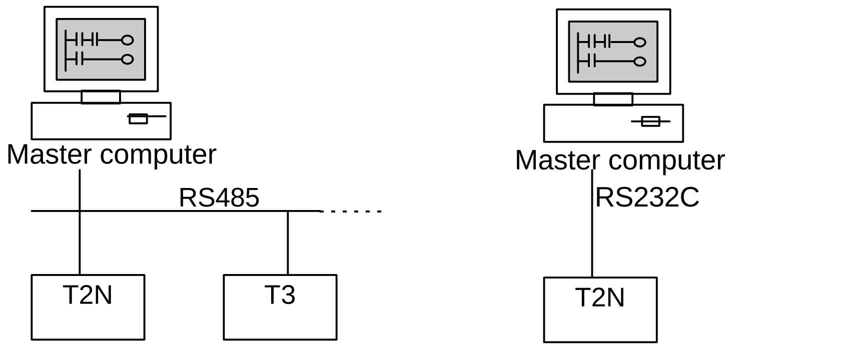
Example.1 System configuration



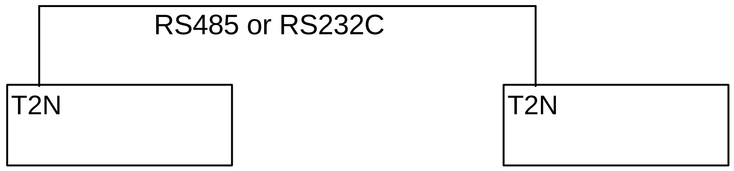
Example.2 Computer Link Function

One-to-N configuration
(Using RS485)

One-to-one configuration
(Using RS232C)



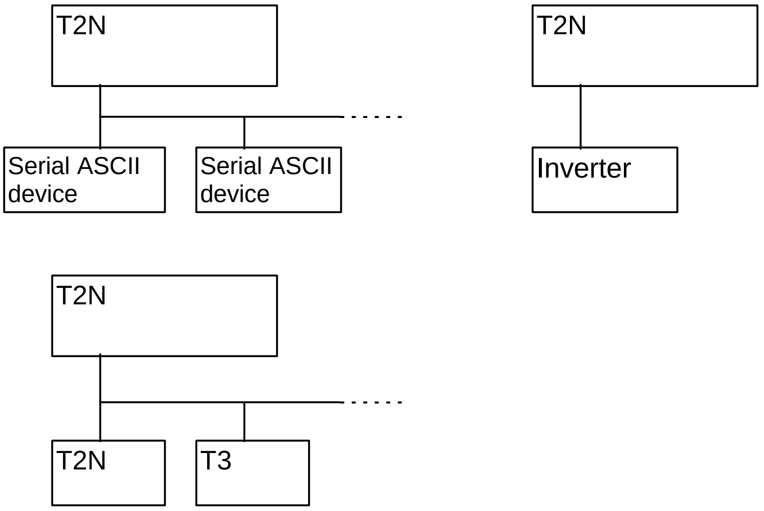
Example.3 Data Link Function



Example.4 Free ASCII Function

One-to-N configuration
(Using RS485)

One-to-one configuration
(Using RS232C)



1.2 Functional Specifications

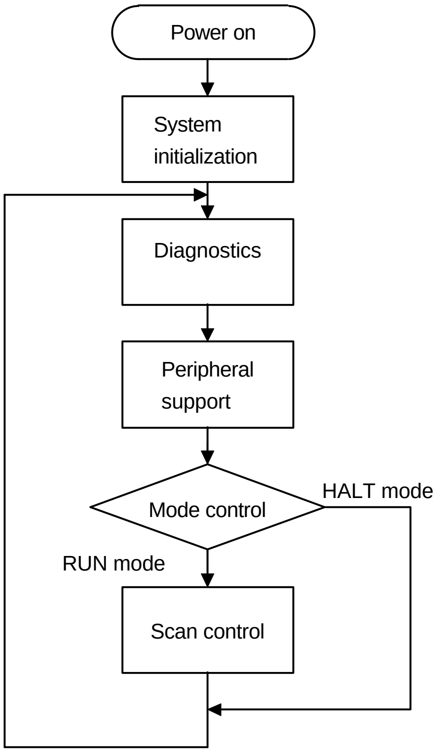
Item		Specifications
Control Method		Stored program, cyclic scan system
I/O Method		Batch I/O(refresh), Direct I/O, or combination
Number of I/O points		2,048 points / 128 words
User Program	Program language	Ladder diagram (relay symbol + function block) SFC (Sequential Function Chart)
	Program capacity	23.5K steps
	Memory	Main memory : RAM (battery back up) Optional memory : EEPROM
	RAM memory back-up	Built-in battery (more than 5 years/ 25°C)
	Instructions	Basic ladder instructions:24, function block instructions:192 ● transfer(single length/double length/registertable) ● arithmetic calculation(single length/double length/binary/BCD) ● logical operation(single length/double length/binary/BCD) ● comparison(single length/double length,sign/unsign) ● program control(jump/FOR-NEXT/subroutine and others) ● function(limit/trigonometric integral/PID/function generator and others) ● conversion(ASCII/BCD/7 segment/HEX-ASCII/ASCII-HEX floating point math function and others) ● Other functions
	Execution speed	0.33µs / contact,0.44µs / coil 1.2µs / transfer,1.6µs / addition
Scanning system		Floating scan/constant scan (interval : 10-200ms, 10ms units)
Multitasking		1 main program, 1sub program 1 timer interrupt (5-1000ms, 5ms units), 4 I/O interrupt
User data	I/O device/register	2,048 points/128words (X / Y, XW / YW batch I/O) (I / O, IW / OW direct I/O) (1 word is 16 bit.)
	Auxiliary device/register	4,096 points / 256words (R/RW)
	Special device/register	4,096 points / 256words (S/SW)
	Timer device/register	512 points (T./T) (T000-T063 : 10ms) (T064-T511 : 0.1sec.)
	Counter device/register	512 points (C./C)
	Data register	8192 words (D)
	Link device/register	16,000 points / 2,048words (Z/W) (for TOSLINE-S20,TOSLINE-30)
	Link relay/register	4,096 points / 256words (L/LW) (for TOSLINE-F10)
	File register	1,024words (F)
	Expanded file register	24,576words (8,192words∗ 3banks accessed by using XFER instruction)
	Index register	I, J, K (total 3words)
	Retentive memoory	User specified for RW,T,C and D
RAS	Diagnosis	Battery level, I/O bus check, I/O respomse, I/O parity, Watch dog timer, illegal instruction, LP check, others
	Monitoring	Event history record, scantime measurment, others
	Debugging	Online trace monitor, force, sampling trace, status latch, others

NOTE
▼△▼

Use READ/WRITE instruction in order to access transmission data of TOSLINE-S20 in the T2N.
The T2N doesn’t support automatic data refresh of TOSLINE-S20.

2.1 Basic Internal Operation Flow

The T2N basic operation flow chart is shown below.



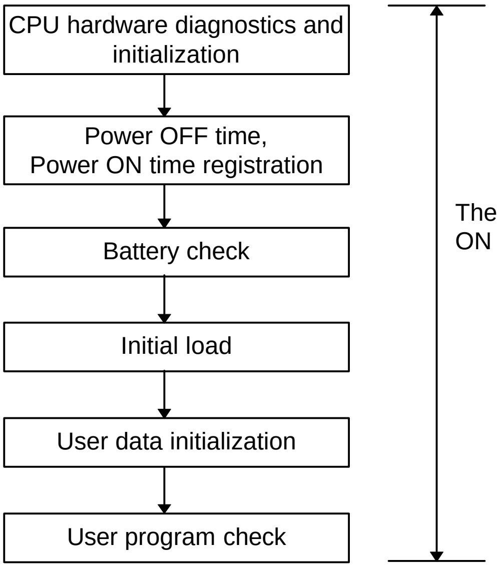
T2N performs diagnostics following power on and the first system initialization. In the absence of abnormalities, peripheral support is processed. However, if the programmer and the computer link is not required, this operation is not executed.

Next, if the RUN mode transitional condition is fulfilled, the scan control begins. The scan control is the basic function of the T2N for the user program execution operation. And if the RUN mode transitional condition is not fulfilled, T2N enters HALT mode and does not execute the program.

The details of these processes are explained in this section. Also, the diagnostics are explained in 5 RAS functions.

2.2 System Initialization

The system initialization is performed after power ON. That is, Hardware diagnostics and initialization followed by system initialization as follows.
The sequence of process is shown below.


CPU hardware diagnostics and initialization
Power OFF time, Power ON time registration
Battery check
Initial load
User data initialization
User program check

The system initialization from power ON to completion :
approx. 2 seconds
(without initial load)
approx. 2-3 seconds
(with initial load)

CPU hardware diagnostics and initialization
The system ROM check, the system RAM check and initial set up, the peripheral LSI check and initial set up, the calendar LSI and the language processor (LP) check take place.

Power OFF time, Power ON time registration
The last time the power was switched OFF is registered in the event history table, and the present date and time of Power ON read from the calendar LSI is recorded. Also, the special register (SW0007-SW013) are set into the present date and time.

Battery cheak
The battery voltage is checked for the user program and the user data back up. If the battery voltage is lower than the specified value a message is recorded in the event hystory table ‘batt voltage drop’ together with the special relay battery alarm flag (S000F) setting.

Initial load
The initial load means the term for the transfer of the contents of the user program and the first 2K words of the data register (D0000-D2047), from the peripheral memory (EEPROM) to the main memory (RAM), prior to running the user program.
The initial load is performed or not, depending on the position of operation mode switch and operation mode setting switches (DIP switches) when the power is turned ON.
The performance table of initial load is shown below.

Performance table of initial load(power on)

Operation mode switch	ROM/RAM switch (DIP SW.2)	Initial load performance
HALT	OFF(ROM)	performed
	ON(RAM)	not performed
RUN	OFF(ROM)	performed
	ON(RAM)	not performed
P-RUN	OFF(ROM)	
	ON(RAM)	

NOTE

▼△▼

When the initial load is performed, the contents of EEPROM is transfered to RAM. That is, the contents of RAM is overwritten. Therefore it is necessary to write to EEPROM before power OFF when the user program is changed.

User data initialization

The user data (registers and devices) is initialized according to the conditions in the following table. :

Register/Device				Initialization
Input register/device (XW/X)	Force area			retained
	Other area except the above			0 clear
Output register/device (YW/Y)	Force area			retained
	Other area except the above			0 clear
Link register/device (W/Z)	Force transmission area			retained
	Other area except the above			0 clear
Link relay register/relay (LW/L)	Force transmission area			retained
	Other area except the above			0 clear
Special register/device (SW/S)	SW0-063	CPU specified		Initialization
		User specified		retained
	SW064-			0 clear
File register(F)				retained
Expanded file register				0 clear
Index register(I,J,K)				0 clear
Auxiliary register/device (RW/R)	Specified retentive area			retained
	Force area			retained
	Other area except the above			0 clear
Timer register/device (T/T.)	Specified retentive area			retained
	Other area except the above			0 clear
Counter register/device (C/C.)	Specified retentive area			retained
	Other area except the above			0 clear
Data register (D)	Specified retentive area			retained
	Other area except the above	D0-D2047	Normal	0 clear
			P-RUN	retained
		D2048-		0 clear

NOTE

▼△▼

Refer to 5.6 Debug Support Function for forced functions.
Refer to Part.3 2.2 for power failure support specification.

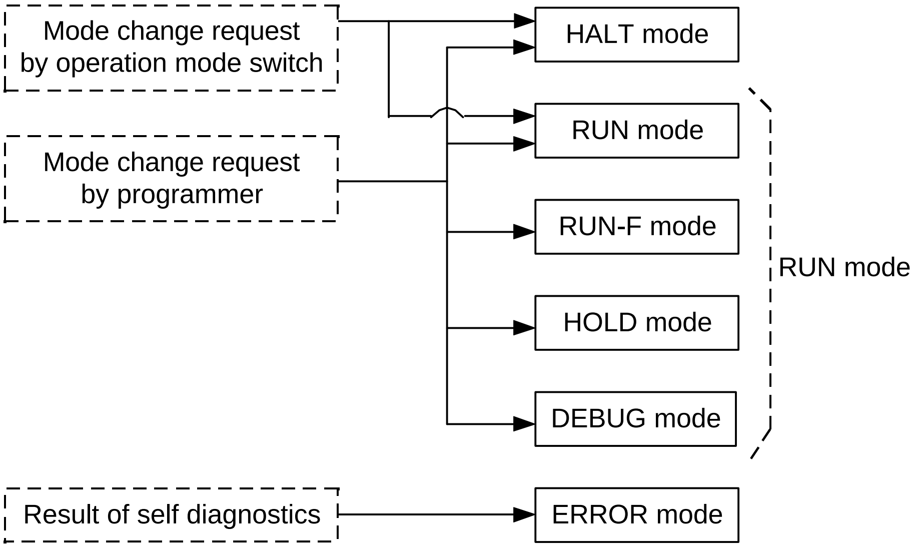
User program check

The content of the user program is checked by the main memory (RAM) on BCC.

2.3 Mode Control

The T2N operation mode is selected according to the status of the mode switch on the CPU module and mode change requests from the peripherals (programmer,computer link,data transmission system).

The T2N operation mode is basically divided into three, the RUN mode, the HALT mode and the ERROR mode. The ERROR mode is when diagnostic checks conclude that normal operation can not continue.
The figure below shows basic mode transition. Also, within the RUN mode, other than the usual RUN mode, there are also RUN-F, HOLD and DEBUG modes mainly for debugging.


Mode change request by operation mode switch
Mode change request by programmer
Result of self diagnostics
HALT mode
RUN mode
RUN-F mode
HOLD mode
DEBUG mode
ERROR mode
RUN mode

The operation of each mode and the mode transition condition are shown below.

HALT : External all output OFF, user program execution and I/O processing halted. In HALT mode, the mode control is performed periodically (every 50ms). Peripheral support and self-diagostics are performed for the idle time. This is the mode for user to edit/change user program.

RUN : After initial load (when neccesary), user data initialization, I/O module setting up, user program check and scan mode decisions, the T2N goes into RUN mode.
Mode control, batch I/O, timer update and user program execution are repeatedly performed in RUN mode. This operation is called scan control.
There are two kinds of scanning system, the floating scan repeats program execution continuously and the constant scan repeats program execution constantly according to user specified time. The selection is called scan mode selection.
Scan control is explained in detail in 2.4 and 3.

RUN-F : This is the forced run mode. It differ from the above RUN mode in that scan control begins even if the allocated I/O modules are not actually mounted. (If other modules are mounted instead, the mode does not run.)
Otherwise the action is the same as the above RUN mode.

HOLD : This is the scan hold mode. Only the batch I/O is run, but the timer update and the user program execution are halted. The scan mode continues previous scan mode.
The I/O test can be done by the data monitor and data set function.

DEBUG : This is the program debugging mode. Program bebugging functions (single step execution, single rung execution, run N scan execution, break point set up, prohibition of external I/O update, etc.) can be used this mode.
Refer to 5.6 for details of debug function.

ERROR : This is the error mode. The T2N goes to this ERROR mode when an error is detected in the self diagnostic checks, operation cannot be resumed by the prescribed retry action and operation cannot be continued correctly. In ERROR mode, all output are OFF and the error reset command from the programmer is effective (the error reset command will recover from ERROR mode to HALT mode). Refer to 5. RAS Functions for details of diagnostic.

The transition conditions for each mode are shown below.

- When power turns to ON

Operation mode switch	RUN/stand-by switch (DIP SW.3)	mode transition factor	operation mode after transition
RUN	OFF(RUN)	power on	RUN
	ON(Stand-by)	power on	HALT
HALT	—	power on	HALT

- When operation mode switch is changed

mode before transition		mode transition factor	operation mode after transition
operation mode	operation mode switch		
HALT	HALT	mode switch → RUN/ P-RUN	RUN
—	RUN	mode switch → HALT	HALT

- When command is executed by peripherals (it is available only in RUN position of operation mode switch.)

mode before transition		mode transition factor	operation mode after transition
operation mode	operation mode switch		
—	RUN	command HALT	HALT
HALT	RUN	command RUN	RUN
	RUN	command Force RUN	RUN-F
	RUN	command DEBUG	D-HALT
RUN/RUN-F	RUN	command HOLD	HOLD
HOLD	RUN	command HOLD cancel	RUN/RUN-F (to previous mode)
D-HALT	RUN	Debug function	D-RUN
D-RUN	RUN	detection break point or command Force Stop	D-STOP
D-STOP	RUN	Debug function	D-RUN
	RUN	command D-HALT	D-HALT
ERROR	—	command ERROR RESET	HALT

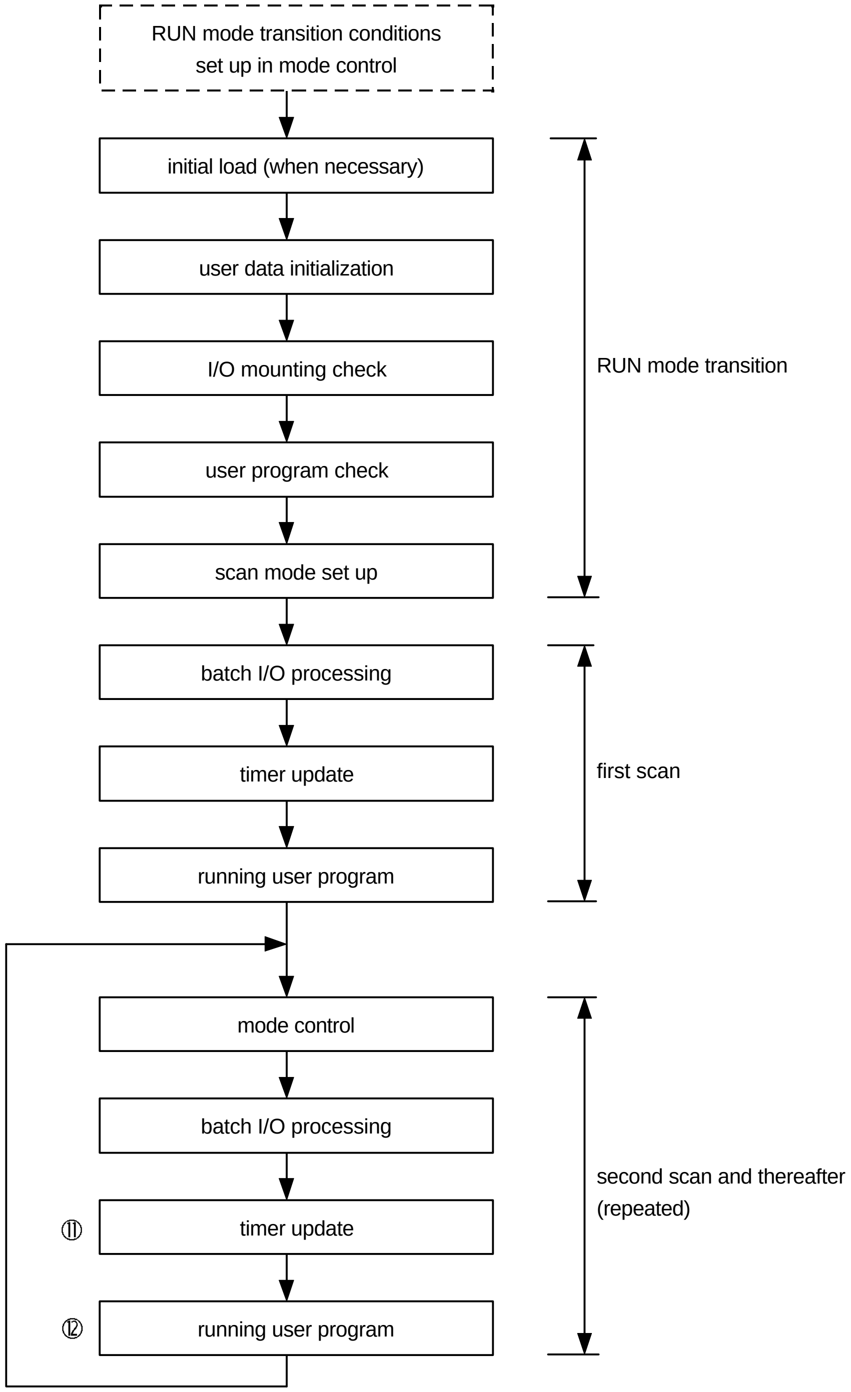
In the above table, the symbol '—' means that mode transition is performed independing on switchs' setting or previous operation mode.

NOTE
▼△▼

Refer to 5.6 for details of bedug mode. The FLT LED of the T2N CPU module is blinking in DEBUG mode.

2.4 Scan control

As explained in 2.3, when the RUN mode transition conditions are set up, the initial load (when necessary), the user data initialization, the I/O mounting check, the program check and scan mode selection are performed, and scan control begins. In scan control, mode control, batch I/O processing, timer update and user program executions are repeated. The following diagram shows the scan control flow chart.



Initial load

When RUN start up is taking place, with RUN position (the operation mode switch) and the ROM/RAM switch (Dip SW.2) on the front of the CPU module is OFF (ROM), the T2N will transfer the contents of the user program and the first 2K words of the data register (D0000 to D2047) from peripheral memory (EEPROM) to the main memory (RAM).

- The initial load is not performed if the user program is written in the EEPROM but the contents thereof are destroyed (BCC error detection). The T2N goes to ERROR mode.

User data initialization

User data initialization takes place after transfer from HALT mode to RUN mode.

Refer to 2.2 System initialization for details of initialization.

I/O mounting check

The I/O module mounting is checked basing on the I/O allocation information.

Refer to 5.RAS functions for details.

User program check

A BCC check is performed on the user program in the main memory (RAM).

Refer to 5.RAS functions for details.

Scan mode set up

Setting of the scan mode (floating scan or constant scan) is performed.

The scan mode is explained in 2.4.1.

, Batch I/O processing

The data exchange between the I/O image table (I/O register/device) and the I/O module is performed on the basis of the I/O allocation information. Data exchange with the data transmission module (TOSLINE-S20LP, TOSLINE-F10, TOSLINE-30) also takes place. The first scan is input only.

Batch I/O processing is explained in 2.4.2.

⑪

, Timer update

The timer register using the timer instruction is updated and the special relay timing relay (S0040 - S0047) is updated.

Timer update is explained in 2.4.3.

⑫

, Running user program

The user program instructions are executed in sequence from the beginning to the END instruction. Here, the user program consists of a main program and sub program.

When the interrupt conditions are set up, interrupt programs halt other operations and are activated immediately.

The user program running control is explained in detail in section 3.

Mode control

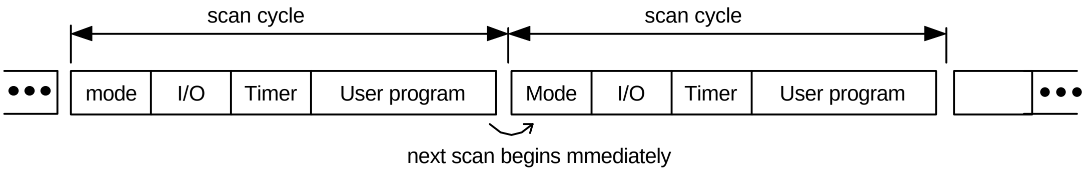
Checks the mode switch and for mode change commands from the programmer and changes operation mode. Also, scan timing control, measurement of the scan cycle and the user program running time are performed.

2.4.1 Scan mode

In the T2N the scan mode enables a choice of floating scan and constant scan.

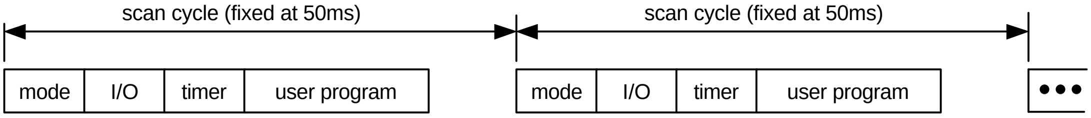
The floating scan mode is that, immediately after one scan is complete the next scan commences. It is the shortest scan cycle but the scan cycle varies according to the user program running state.

The action of the floating scan is shown in the following diagram.



The constant scan mode has a specified time cycle for scanning. The set up range of the cycle is 10 - 200ms (10ms units). Use this scan cycle to avoid variation in scan intervals.

The action of the constant scan when the cycle is fixed at 50ms is shown in the following diagram.

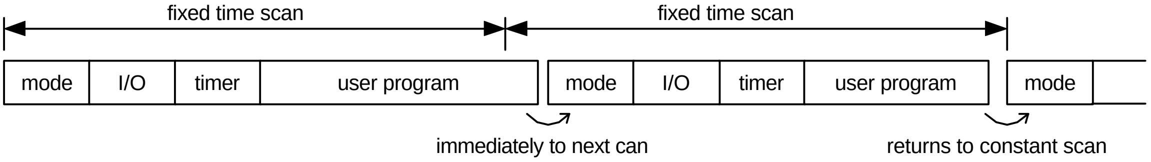


Scan mode selection is performed by setting up the scan cycle in the system information menu of the programmer.

To select floating scan, do not set up a scan time (leave blank).

With the constant scan the scan time can be set up within the range 10 - 200ms (10ms units).

Note) In the constant scan if the time for one scan is exceeded in a specified cycle it becomes a floating scan, and the fixed time scan retard flag (special relay - S0008) comes ON. Also, when the scan time reverts to within the specified cycle the scan cycle returns to the original constant scan.



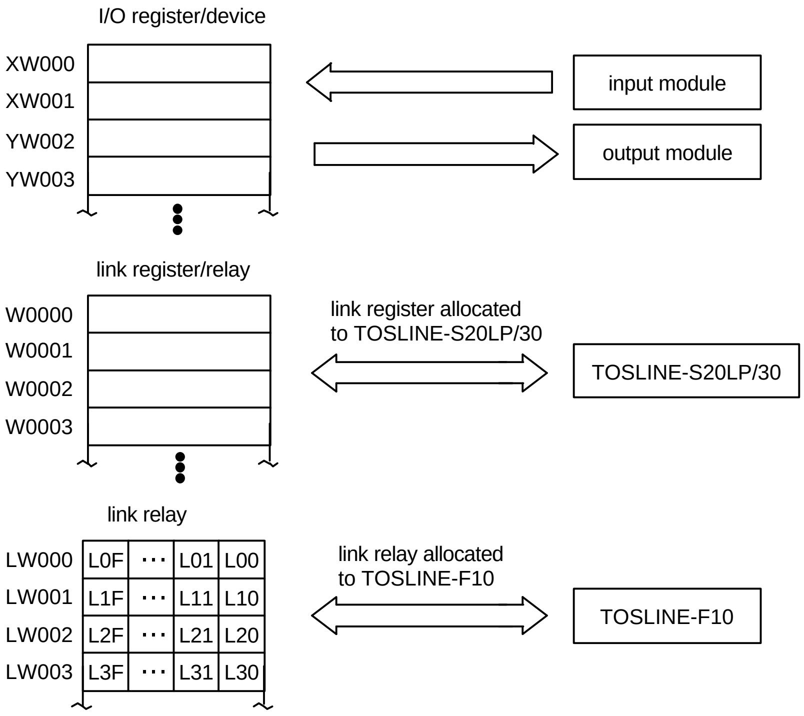
2.4.2 Batch I/O processing

The status of the external input signals from the input module is read onto the I/O register/device (XW/X), the output register/device (YW/Y) status is output to the output module. This process takes place before user program execution and is done in batches, hence name batch I/O processing. Batch I/O processing proceeds as follows.

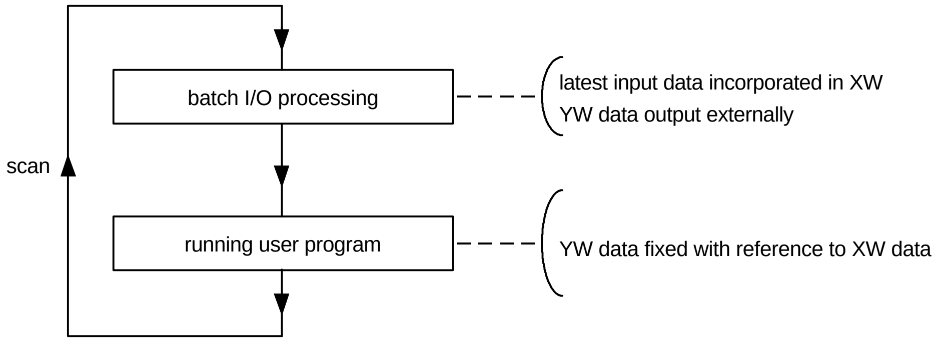
Batch inputting ··· signals from the input module with no i specification on I/O allocation and input registers/devices (XW/X) which are not forced

Batch outputting ··· output register/device (YW/Y) corresponding to output modules with no i specification on I/O allocation

Also, the data transmission module (TOSLINE-S20LP, TOSLINE-F10, TOSLINE-30) and the link register relay (W/Z and LW/L) within the CPU module run the data exchange.



If we consider the T2N operation simply from the viewpoint of external signal exchange, batch I/O processing and user program execution can be considered to be repeated continuously, as shown in the following diagram.


batch I/O processing
latest input data incorporated in XW
YW data output externally
scan
running user program
YW data fixed with reference to XW data

So basically, this has the advantage that high speed scans can take place so that I/O module data is not exchanged during user program execution and also it is easy to create program logic which prevents XW data changing during user program execution. This method is called the batch I/O processing method (refresh method).

There is also another method of the T2N operation whereby I/O module data exchange takes place during user program execution, using IW/I instead of XW/X and OW/O instead of YW/Y. This method is called the direct I/O processing method. It is recommended that the I/O modules used in direct I/O are inhibited the batch I/O (they have i specification on I/O allocation) so shorten the time for batch I/O processing.

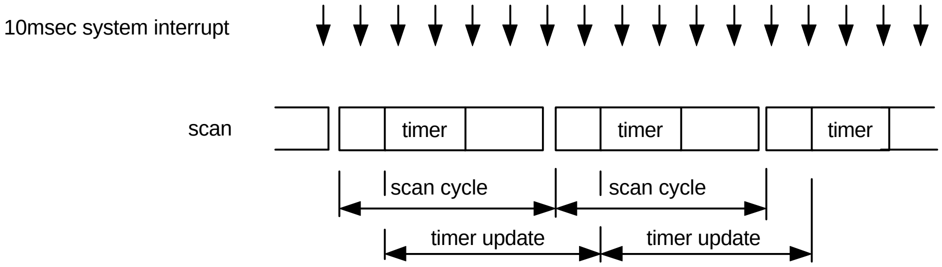
NOTE

1. Use the following criteria for batch I/O processing time.
2. input (XW) } ... approximately 45µs/register
3. output (YW) }
4. link(W) ... 8 µs/register
5. (LW) ... 22µs/register
6. I/O modules with i specification on I/O allocation (iX, iY, iX + Y) are not part of batch I/O processing. Refer to Part 3 for I/O allocation.
7. Forced input device (X), link register relay (Z), and link relay (L) are not part of batch I/O processing. The force function is explained in section 5.
8. Refer to the data transmission module manual for the allocation of the link register/relay (W/Z and L/LW) to the data transmission module.
9. With the direct I/O processing method, output is in register units even when the bit (O) is specified. Refer to Part 3 for direct I/O registers.

2.4.3 Timer update

The timer register used in the timer instruction is updated (increased), and the timing relay within the special relays (S0040 - S0047) is updated.

- updating the timer register



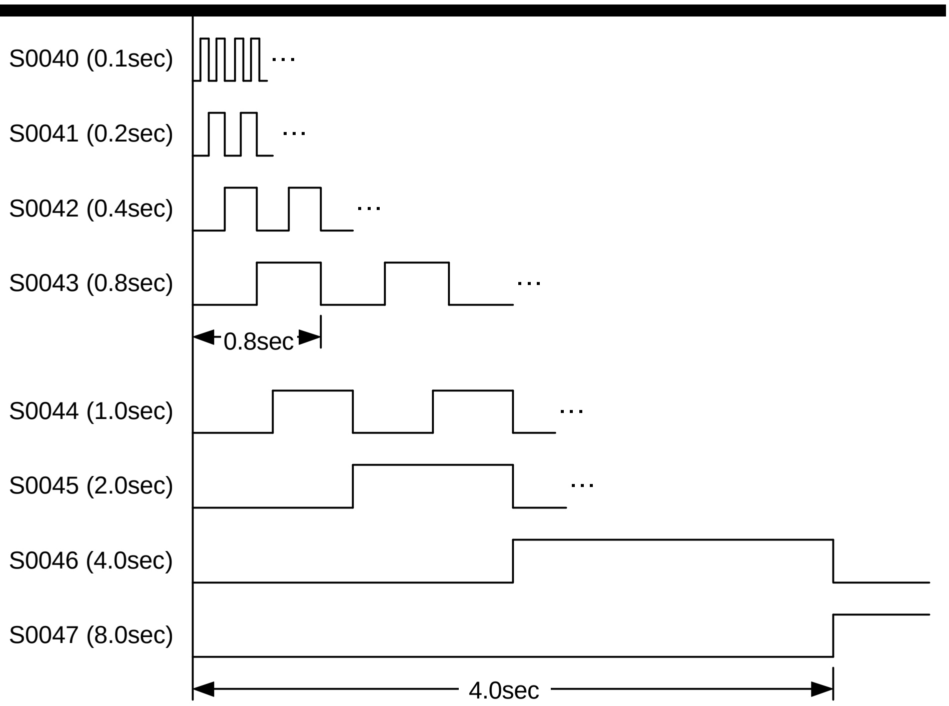
The number of system interrupts which occur during the timer update cycle (≒scan cycle) are counted and these counts are added up in the timer register which is started up by the timer instruction (TON, TOF, SS, TRG).

The 10msec system interrupt is used for the timer update. The timer reset and the time up processing are performed when running the timer instruction.

timer classification	timer register (timer device)	preset range	Notes
0.01 second timer	T000~T063 (T.000~T.063)	0~32767 (0~327.67 seconds)	on delay timer (TON) off delay timer (TOF) single shot timer (SS) timer trigger (TRG)
0.1 second timer	T064~T511 (T.064~T.511)	0~32767 (0~3276.7 seconds)	

*) Take the criteria for the time for performing the timer register update as follows.
22µs/timer register (update time)

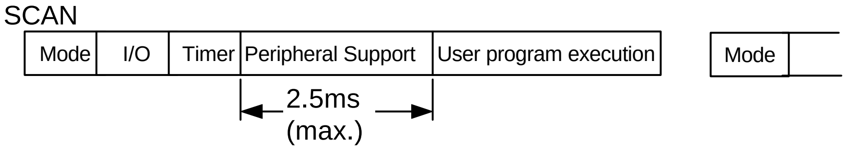
- Timing relay update
  The timing relay (S0040 - S0047) ON/OFF status is controlled by using the 10msec system interrupt. The binary counter is configured as shown on the next page. (When RUN is started up, they are all OFF.)



2.5 Peripheral support

Peripheral support processing interprets the request commands from the periphrals (programmer, computer link, data transmission module), process the requests and responds.

Peripheral support processing time is limited up to 2.5ms per one scan so that scan time is as constant as possible. If it takes more than 2.5ms to process peripheral support, this processing is stopped once within 2.5ms and remained processing is continuously performed in the next scan.



*) When 2 or more request commands are received simultaneously from the request source, the processing priority is as follows.

Programmer Port > Communication port > TOSLINE-S20LP
(Computer Link)

As for data link processing, it may take 0.5ms at the worst case in addition to the above.

<Peripheral support priority mode>

When special relay S158 is ON, peripheral support processing time is not limited and takes place in one scan.

It results in swift response for the peripherals although the scan time is extended.

<Computer link response delay mode>

Response of the T2N can be delayed on the communication port using SW57.

2.6 Programming support functions

The programming support functions form part of the functions realised as a result of peripheral support processing. Details of the programming support functions are explained in separate manuals for the programmer. The explanation here relates to an overview of the functions and their relation to the T2N operation modes.

(1) Memory clear
When the memory clear command is received, the content of the user program memory (RAM) is initialized and the content of the user data memory (RAM) is cleared to 0.

(2) I/O automatic allocation
When the I/O automatic allocation command is received, the types of I/O modules mounted are read and the I/O allocation information is stored on the system information. (System information is in the user program memory.)

(3) Reading the I/O allocation information
The I/O allocation information is read from the system information, and sent to the peripherals.

(4) Writing I/O allocation information
I/O allocation information received from peripherals is stored on the system information.

(5) Reading the system information
The system information (program ID, retentive memory specification, number of steps used, scan mode specification, other) is read and sent to the peripherals.

(6) Writing system information
The system information (user set up items) received from the peripherals is stored in the system information.

(7) Reading the program
In response to a request from peripherals, a specified range of instructions is read from the user program memory, and sent to the peripherals.

(8) Writing the program
A specified range of instructions is received from peripherals and written onto the user program memory. After writing, a BCC (check code) correction is carried out immediately.

(9) On-line program change

A BCC (check code) correction is carried out immediately after rewriting the content of the user program memory (adding / changing / inserting / deleting) in RUN mode. This action is performed after completion of one scan, so the scan cycle is extended while this is processed.

Changing the program on-line is subject to the following restrictions.

- You can not change the number or running order of instructions which are related to the program execution (see below).

  END, MCS, MCR, JCS, JCR, JUMP, LBL, FOR, NEXT, CALL, SUBR, RET, IRET

- You can not change SFC structure in the SFC program section, but you can change the action part corresponding to a step and a transitional condition. (Ladder diagram part).

(10) Batch reading of program

The content of the user program memory (including the system information) is read and sent to the peripherals.
It is used for the program uploading (T2N→Programmer→FD).

(11) Batch writing the program

The user program (including the system information) is received from the peripherals and stored in the user program memory.
It is used for the program download (FD→programmer→T2N).

(12) Search

The instruction operand specified by the peripherals is extracted from the user program memory and sent the address to the peripherals.

(13) Program check

When the program check command is received the user program syntax is checked. The results of this check are sent to the peripherals.

(14) Reading data

The specified data is read from the user data memory in response to a request from the peripherals, and sent to the peripherals.

(15) Writing data
User data address and data content received from the periphrals is stored in the user data memory.

(16) Program reading from the EEPROM
The checked EEPROM contents are transferred to the user program memory and user data memory (RW, T, C, D) of the main memory (RAM).

(17) Program writing to the EEPROM
The content of the user program memory and user data memory (RW, T, C, D) of the main memory (RAM) are transferred to the EEPROM.

The execution conditions for these functions are shown below.

Function	Execution conditions
Reading the I/O allocation information	Possible always in any mode except in the case of communication impossible with the periphrals when detecting error in the initialization
Reading the system information	
Reading the program	
Reading data	
Batch reading the program	Possible except in ERROR mode
Search	
Program check	Possible in HALT mode
Program writing to EEPROM	
Memory clear	Possible when in HALT mode except when operation mode switch is P-RUN
I/O automatic allocation	
Writing the I/O allocation information	
Writing the system information	
Writing the program	
Batch writing the program	
Program reading the EEPROM	
On-line program changing	Possible except in ERROR mode and except in P-RUN
Wriitng data	Possible except in ERROR mode, however writing into D0000-D2047 is prohibited in P-RUN.

3.1 Program classification

The T2N can run several different types of program, main program, sub program and interrupt program in parallel (this function is called the multitask function). This function can be used to realize the optimal response time for each application.
The programs are classified into the following 3 types, there are a total of 7 programs.

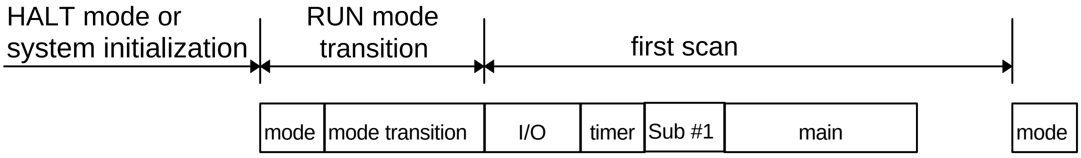
- Main program (one)
  This program is executed every scan and forms the main part of the scan.
- Sub program (one)
  This program is called the sub program #1. When RUN starts up, it is executed once only before the main program and after batch I/O processing and timer update.
- Interrupt program (five)
  When the interrupt condition is set up, the interrupt program stops other operations and is executed immediately. A total of 5 are provided : one program which starts up at user specified intervals (fixed cycle timer interrupt program) and
  4 programs which start up according to interrupt signals from I/O modules with an interrupt function (I/O interrupt programs) (No.1 - No.4).
  By using the timer interrupt, it results in taking place time critical control and effective multitask control. By using I/O interrupts, I/O responses can take place without affecting the scan cycle:

Sub program #1 and the interrupt program running method and the execution conditions are explained in this section.

3.2 Sub program control

When RUN starts up, sub program #1 is run once only before the main program is executed on the first scan. Therefore, use sub program #1 as the initial setting program at the starting of the operation.

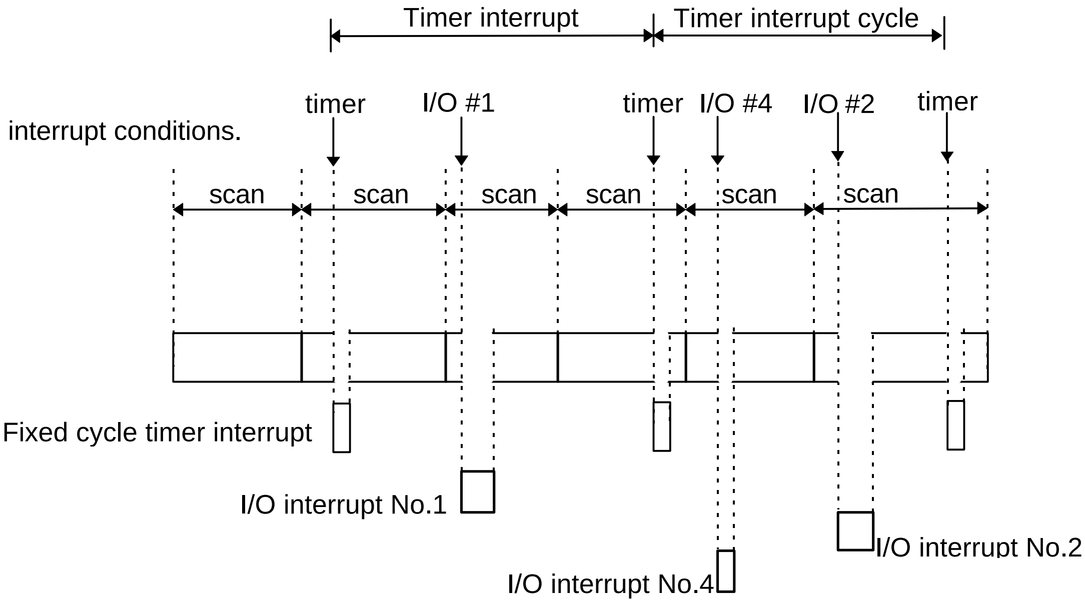
The first scan operation is shown in the following diagram.



3.3 Interrupt program control

When the interrupt condition is set up, the interrupt program stops other operations and is executed immediately. As shown below, you can register one timer interrupt program which starts up according to a cycle set up in system information and 4 I/O interrupt programs which start up according to interrupt signals from I/O modules with an interrupt function* (No.1 - No.4).

Interrupt program	Operation
Fixed cycle timer interrupt	Runs acording to the user specified interruption cycle time in system information. The interruption cycle time is set at 5 to 1,000ms (units 5ms)
I/O interrupt No.1 I/O interrupt No.4	I/O interrupt programs run by interrupt signals generated from I/O modules with interrupt function*



*) I/O modules with an interrupt function is under development.

(1) Priority control of interrupt program

When several interrupt conditions occur simultaneously, the programs are run in the order of priority shown in the following table (the lower the numerical value the higher the level of prriority). Also, if other interrupt conditions occur during an interrupt program run, the interrput conditions are put on hold during running of the interrput program, and after it is completed, they are run in order of priority.

Interrupt program	Level of priority	Level of priority within classification
Timer interrupt	0	-
I/O interrupt No.1	1	0
I/O interrupt No.2		1
I/O interrupt No.3		2
I/O interrupt No.4		3

The timer interrupt has the highest level of priority, followed by the I/O interrupt programs in order.

With respect to the level of priority for I/O interrupt, the I/O interrupt form the I/O module with interrupt function nearest the CPU has the highest level of priority. Refer to (3) below regarding the correspondence between interrupt programs and I/O modules.

(2) Interrupt enable/disable

You can switch between interrupt enable and disable by executing DI instruction (interrupt disable) or EI instruction (interrupt enable) in the user program. The interrupt request is hold during interrupt is prohibited after executing the DI instruction. After the EI instruction is executed and this request is permitted.

Also, interrupt is prohibited in the first scan after the transition to RUN mode and it is permitted from the second scan.

(3) Allocation of I/O interrupt program

The I/O interrupt with the lowest number corresponds to the I/O module with interrupt function nearest the CPU.

NOTE ▼△▼

- I/O modules with interrupt function is under development.

4.1 EEPROM Support

The contents of the user program and the register data (D) can be stored in the EEPROM and they can be read into the main memory (RAM) by the initial load function in the T2N or programmer operation (for maintanance).
The register data (D) in the EEPROM can be written to internal registers or be read out from them in the T2N during RUN operation by using special instruction XFER. The T2N can run without batteries and can be recovered easily from error down if user program is destroyed by using the EEPROM.
The following functions are available with the EEPROM.

Function	Operation	Execution condition
Initial load	Transfers the contents of the EEPROM to the user program memory and data registers (D0000-2047) in the main memory (RAM). However when the mode is transited from HALT to RUN, other registers except retentive specified memory is cleared.	At system initialization when power is turned on and ROM/RAM switch is set to ROM (not executed in the P-RUN mode)
		At transition to the RUN mode when transited to the RUN mode and ROM/RAM switch is set to ROM (not executed in the P-RUN mode)
Read/write the data registers in EEPROM	Reads out the data registers (D) in the EEPROM and stores in the main memory by user program. Writes the specified data of the main memory into the data registers in the EEPROM by user program.	Accessed by expanded data transfer instruction (XFER).
Write EEPROM (by programmer)	Writes the contents of the user program (including the system information) and the data registers(D), the timer registers(T), the counter registers(C) and the auxiliary relay registers(RW) in the main memory(RAM) into the EEPROM.	Executed by programmer command “Program write to IC card EEPROM” in the HALT mode
Read EEPROM (by programmer)	Transfers the contents OF the EEPROM to the user program (including the system information) and the data registers(D), the timer registers(T), the counter registers(C) and the auxiliary relay registers(RW) in the main memory(RAM).	Executed by programmer command “Program read from IC card EEPROM” in the HALT mode (not executed in the P-RUN mode)

*The P-RUN mode is when the operation switch is P-RUN.

NOTE
▼△▼

1.Refer to 2.2 System initialization and 2.4 Scan control for details of the initial load function.

2.The EEPROM can be written up to 100,000 times(guaranteed) depending on the hardware. The EEPROM alarm flag(S0039) comes ON if the EEPROM is written more than the limitation. Thereafter operation is not guaranteed.
S0039 may come ON when power is turned OFF during writing into EEPROM. In this case, carry out Write EEPROM command once. Nevertherless if S0039 is ON, it is recommended that the CPU module should be changed.

5.1 Overview

The meaning of RAS is Reliability, Availability and Serviceability, the RAS function is the general term used for the functions installed in the T2N which increase the reliability and serviceability of the applied systems and support the operation of the system.

This section explains the self-diagnostic functions installed in the T2N, the maintenance functions, the debugging functions and the system checks which can be run by the T2N user.

5.2 Diagnostics

The T2N runs checks on itself. The details of these self-diagnostics which are designed to prevent abnormal operation, the timing of the diagnosis and procedure when malfunctions are detected are shown below.

In building up the system, consider the system operation safety should a in case of a T2N shut down (fail safe) and the system operation backup function.

In the following explanation, error registration means the storing of the details of the error and the time when it occurred on the event history table; error down means that all the outputs turn OFF and ERROR mode is entered; alarm means that the erroris registered, the special relay is set, and running is continued.

(1) Diagnostics at system initialization (when power supply is turned on)

Diagnostics	Diagnostics details	Processing when error detected
System ROM BCC check	BCC check on the correctness of the system ROM	Error registration takes place, FLT LED flash. (Programmer communication impossible)
System RAM check	The system RAM read/write is checked.	Error registration takes place, the FLT LED flashes. (Programmer communication impossible)
Peripheral LSI check	Peripheral LSI checked for normal initialization. (Read back check)	Error registration takes place, the FLT LED flashes. (Programmer communication impossible)
LP check	LP (language processor) is checked for normal initialization.	Error registration takes place, ERROR mode is entered. (Error reset command invalid)
User program memory check	BCC check on the correctness of the content of the user program memory. (Checked after initial load when peripheral memory is present)	Error registration takes place, ERROR mode is entered.
User data memory check	The user data memory read/write is checked.	Error registration takes place, ERROR mode is entered. (Error reset command invalid)
Peripheral memory check	BCC check on initial loading of the peripheral memory (EEPROM).	Error registration takes place, ERROR mode is entered.

Calendar LSI check	The accuracy of the data read from the calendar LSI (date and time) is checked, the data is set in the special register.	Alarm. Until the calendar is reset, the date and time data (in the special register) are HFF.
Battery check	The voltage of the memory backup battery is checked.	Alarm. If the user program memory BCC is normal, it starts up normally. (However, user data without in retentive memory specification is not guaranteed.)

(2) RUN start up diagnostics

Diagnostics	Diagnostics details	Processing when error detected
I/O verify check	The I/O allocation information and the I/O modules mounted are verified, to check that they agree.	Error registration, error down. However, when start up is activated by a command from the programmer a message is displayed, it remains in HALT mode and no error registration takes place.
I/O bus check	Checks that I/O bus is correct.	Error registration, error down. However, when start up is activated by a command from the programmer a message is displayed, it remains in HALT mode and no error registration takes place.
I/O response check	Checks that response when I/O module is accessed is within specified response time limits.	Error registration, error down. However, when start up is activated by a command from the programmer a message is displayed, it remains in HALT mode and no error registration takes place.
Program check	User program syntax is checked.	Error registration, error down. However, when start up is activated by a command from the programmer a message is displayed, it remains in HALT mode and no error registration takes place.

(3) Diagnostics during scanning

Diagnostics	Diagnostics details	Processing when error detected
I/O bus check	Checks that I/O bus is normal. (at batch I/O processing)	Error registration then error down. (However, after a fixed number of retries, only registration takes place; no error down.)
I/O response check	Checks that response when I/O module is accessed is within specified response time limits. (At batch I/O processing and at direct I/O instruction)	Error registration then error down. (However, after recovered by retries, only registration takes place; no error down.)

I/O bus parity check	Bus parity is checked when the I/O module is accessed. (At batch I/O processing and direct I/O instruction)	Error registration then error down. (However, recovere by retries, only registration takes place; no error down.)
LP function check	Test program run in LP (language processor) and checked for correct results. (When running the user program)	Error registration then error down. (However, recovered by retries, only registration takes place; no error down.)
LP illegal instruction detection check	Checks whether or not illegal command detected in LP (language processor). (When running the user program)	Error registration then error down.
Scan time over check	Checks that scan cycle does not exceed set value (200ms). However, set value can be changed by user instruction (WDT). (When running the user program)	Error registration then error down.

(4) Diagnostics during normal running (take place in background)

Diagnostics	Diagnostics details	Processing when error detected
Watchdog timer check	Watchdog timer system runaway check. (Set at 350ms)	Error registration, transition to ERROR mode after system reset.
User memory check	User memory (RAM) read/write checked.	Error down after error register (with retry)
Battery check	Memory backup battery voltage checked.	Alarm
Calendar LSI check	Date and time data read from calendar LSI every 300ms, accuracy checked, data set in special register.	Alarm. Until calendar reset, date and time data are HFF.

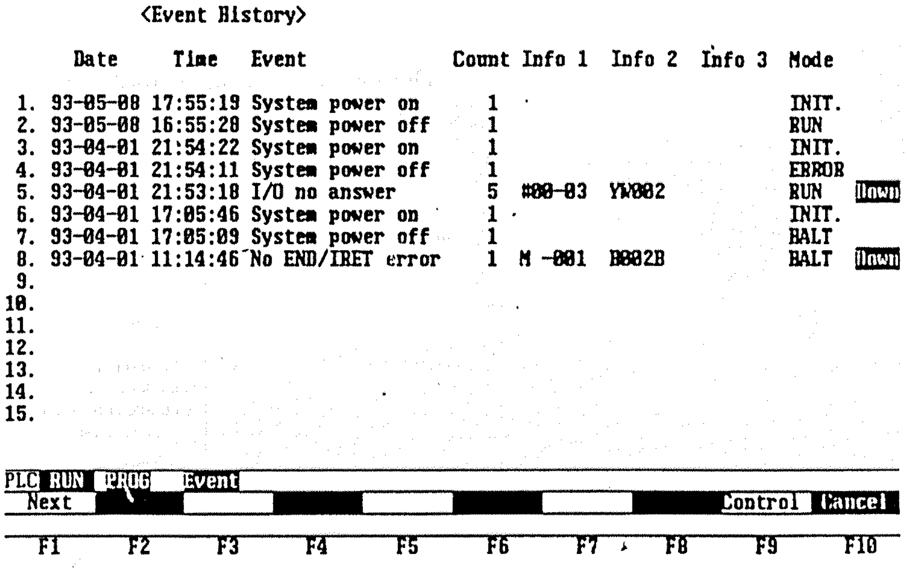
NOTE
▼△▼

For details of registration in the event history table when a error occurs and the special relay addresses that are set, refer to Part 1, section 6.

5.3 Event history

When an error is detected by the T2N diagnostics the details and time of occurrence are registered in the event history table (besides errors, the times power ON/OFF are also registered). The 30 recentest occurrences of errors are registered in the event history table. As new data is registered, the data registered previously is shifted down in sequence, and the oldest data is deleted.

Use the event history table for maintenance since with the programmer connected you can display and details as on the following diagram. The details on the event history table are stored until executing the event history clear command or the memory clear command from the programmer.



```
        <Event History>

      Date     Time   Event            Count Info 1  Info 2  Info 3  Mode

 1. 93-05-08 17:55:19 System power on     1                          INIT.
 2. 93-05-08 16:55:20 System power off    1                          RUN
 3. 93-04-01 21:54:22 System power on     1                          INIT.
 4. 93-04-01 21:54:11 System power off    1                          ERROR
 5. 93-04-01 21:53:10 I/O no answer       5  #00-03  YW002           RUN   Down
 6. 93-04-01 17:05:46 System power on     1                          INIT.
 7. 93-04-01 17:05:09 System power off    1                          HALT
 8. 93-04-01 11:14:46 No END/IRET error   1  M -001  B002B           HALT  Down
 9.
10.
11.
12.
13.
14.
15.

PLC RUN  PROG   Event
 Next                                                          Control Cancel
  F1     F2     F3     F4     F5     F6     F7     F8     F9     F10
```

*) When the calendar is not set initially or the calendar data is not backed up, event history is registered with the date and time displayed as “??-??-?? ??:??:??”.

The meaning of each item on the screen above is as follows.

(1) Number (1-30)
Indicates order of occurrence. Number one is the recentest.

(2) Date (year-month-day)
Indicates the date of occurrence. This is shown as “??-??-??” if the calendar data malfunctions.

(3) Time (hours:minutes:seconds)
Indicates the time of occurrence. This is shown as “??-??-??” if the calendar data malfunctions.

(4) Event
Indicates the what sort of error has been detected. (System power on indicates when system power is turned on and system power off when system power is turned off.)

(5) Count
Indicates the number of times the error was detected. For example, an error is detected during a process, the retry is repeated 3 times, the malfunction does not change and it goes to error down. This is indicated as a count of 4 and DOWN is displayed under the Mode.

(6) Information 1, Information 2, Information 3
Indicates supplementary information regarding malfunction. For example, with an I/O malfunction the I/O module position (unit No, slot No) where the malfunction occurred and the read/write register address etc are indicated.

(7) Mode
Indicates the actual mode when the error was detected. Also displays DOWN when error down occurs. On the mode display, INIT. indicates system initialization after power is turned on.

*) Refer to Part 1 section 6 for display details of detected errors and methods of proceeding.

5.4 Memory Protect Function

Memory Protect function is effective when the operation switch on the front of the CPU module is P-RUN. This is called memory protection.

The following operations cannot be carried out by programmer in the memory protection. The message “Memory protected” will be displayed on the programmer screen if you try to do so.

The following operations are prohibited in the memory protection.

(1) Memory clear

(2) I/O automatic allocation

(3) Write I/O allocation information

(4) Write system information

(5) Program editing (incliding on-line changes)

(6) Program download to the T2N from FDD etc.

(7) Program read from EEPROM (including initial load)

(8) Write data to first 2K words of data register (D0000-D2047)

The memory protect function can prevent the program from being destroyed due to incorrect operation of the programmer.

NOTE ▼△▼

Memory Protect function is effective when the operation switch on the front of the CPU module is P-RUN.

5.5 Execution status monitoring

The T2N support functions to monitor the status of T2N scan control, are as follows. (Refer to separate manuals for the programmer for these operation.)

(1) Execution time measurement function

Measures the following execution times. This data can be verified by reading the programmer.

- Scan cycle ... present value, maximum value, minimum value (1ms units)
- Main program execution time ... present value, maximum value, minimum value (1ms units)
- Sub program execution time (sub program No. 1) ... present value, maximum value, minimum value are all the same values (1ms units)
- Timer interrupt execution time ... latest value, maximum value, minimum value (0.1ms units)

NOTE
▼△▼

1.The scan cycle value includes the scan overhead and all interrupts occurring during the scan.
2.With the main program and the sub program execution times the interrupt time for any interrupts occurring are excluded.

(2) On-line trace function

This function traces the status during program execution and displays on the programmer screen (power flow display, register value display) in the circuit range being monitored by the programmer.
Since this displays data from the point in time that the instruction is executed rather than at the end of a scan cycle, it is also useful for program debugging.

(3) Status monitor function

Collects and displays the status of up to 8 points of devices/registers specified using the auxiliary display functions of the programmer on-line trace screen, immediately after the point in time when the above on-line trace is run.

(4) Sampling trace function
Collects data from specified devices/registers when the sampling conditions set by the programmer are realized, and stores it in the sampling buffer. Also, the number of sampling data may be selected:

3 registers + 8 devices ... 2048 times

7 registers + 8 devices ... 1024 times

The evaluation of the sampling trace conditions and the data collection are executed at the bottom of the scan.
The sampling data read by the programmer can be displayed in timing charts.

(5) Status latch function
Transfers specified device/register data in batches to the latch data store area when the latch conditions set by the programmer are realised or when the latch instruction is run.
The latch conditions are evaluated and data collected at the bottom of the scan. However, when the latch instruction is run, the data is collected when the instruction is executed. Latched data can be displayed on the programmer.

NOTE
▼△▼

The sampling buffer size setting is not needed in the T2N.

The T2N has built-in sampling buffer (8K words).

5.6 Debug Support Function

The T2N support the following functions enable the user program to be debugged. (Refer to separate manuals for programmers for operation of these.)

(1) Input force/coil force function

Batch input data is not updated in the input force specified register/device.

The registers/devices which can be specified for forced input are the input register/device(XW/X), link register/relay (W/Z) in the receiver area and link register/relay (LW/L) in the receiver area.

On the other hand, coil force specified coil instruction can not be processed when the program is running, so despite the run state of the circuit, the coil device maintains its previous state. The coil force devices which can be specified as forced coil are the output device (Y), the auxiliary relay (R), the transmitter area link register relay (Z), and the transmitter area link relay (L).

Simulated input and simulated output are made possible by the combined use of the input force/coil force function and the data setting function.

(2) Constant operand change function

This function enables to change the constant values of timer and counter instructions (preset values) and the constant values used in function instructions in on-line mode (during RUN) .

The constant values for the timer and the counter can also be changed while in memory protect mode (P-RUN).

(3) On-line program change function

This function enables to change the user program on-line (during RUN). The changes are made after completion of one scan, so it extends the inter-scan cycle.

On-line program change is subject to the following conditions.

- You cannot make changes to the number or order of run control related instructions (below).
  END, MCS, MCR, JCS, JCR, JUMP, LBL, FOR, NEXT, CALL, SUBR, RET, IRET

- You cannot change the SFC structure in the SFC program section, but you can change the action or transition condition (ladder diagram) which relate to step or transition.

The following functions are available only when in the DEBUG mode.

(4) Single step execution function
Starts and halts in units of one instruction. The trace of run state is displayed on the screen being monitored by the programmer.

(5) Single rung execution function
Starts and halts in units of one rung. The trace of run state is displayed on the screen being monitored by the programmer.

(6) N scan execution function
Starts and halts only with respect to the number of times the specified scan is run. The trace of run state is displayed on the screen being monitored by the programmer.

(7) Break point set up function
Starts and halts up to the instruction which is specified as the break point. The break point can be set in one location only. The trace of run state is displayed on the screen being monitored by the programmer.

(8) I/O simulation
No batch I/O processing is not performed during scan control. Also if you run direct I/O instruction, the data exchange with the I/O module does not take place, and the image table (XW/YW) data is used.
This is used when the program debug is run and is not output to the external output. The input state can be set up from the programmer. Also the run state is displayed as on-line trace.

(9) Trace back function
The on-line trace information of the latest 5 scan is stored in the DEBUG mode except single step execution funtion and single rung execution function.
This function is effective to check the data changes of each scan.

(Refer to T-PDS commnad reference manual for details of operation in the DEBUG mode.)

5.7 System diagnostics

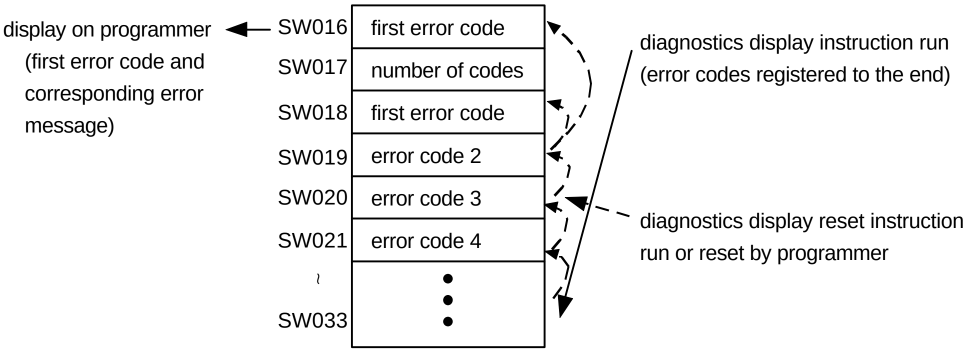
The following functions are provided for diagnosis of operation/status. The system can be monitored easily using of these functions.

(1) Diagnostics display function

Use of the diagnostics display instruction (DIAG) in the user program, the relevant error code (1-64) and error message (maximum 12 characters per message) can be displayed on the programmer screen. Also, the error code generated is stored in the special register (SW016-SW033) in order of generation up to a maximum of 16 codes and the annunciator relay (S0340-S037F) corresponding to the error code goes ON. It is possible to use the special register/relay to display the error code on an external display monitor.

The error code registered can be reset one at a time (shift up after erased) using the programmer or by the diagnostics display reset instruction (DIAR).

This function may also be used effectively in conjunction with the bit pattern check and the sequence time over detection mentioned below. (Refer to details of diagnosis display command in other manual on instruction set.)



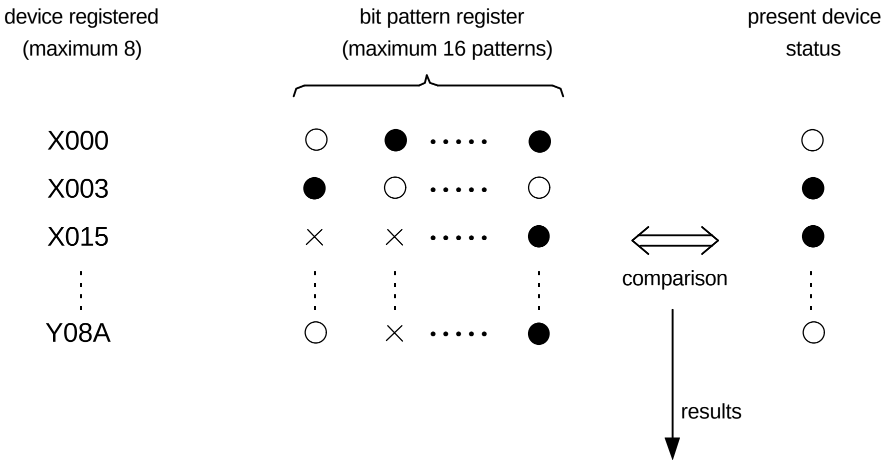
When error codes are registered, for example 3, 10, 29, 58, each corresponding annunciator relay, S0342, S0349, S035C, S0379 comes ON.

(Annunciator relay)

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
SW034	16	15	14	13	12	11	**10**	9	8	7	6	5	4	**3**	2	1
SW035	32	31	30	**29**	28	27	26	25	24	23	22	21	20	19	18	17
SW036	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
SW037	64	63	62	61	60	59	**58**	57	56	55	54	53	52	51	50	49

(2) Bit pattern check function

This function checks that the device ON/OFF status for a number of devices are in the normal combinations (pattern). For example, checks that not more than 2 from device 1, 2 and 3 are ON simultaneously. When a maximum of 8 devices are registered up to a maximum of 16 patterns are possible. The check is carried out immediately before starting a scan, the results are reflected in the special relay S0142.



agrees with the register pattern ... S0142 = 0
does not agree with the register pattern ... S0142 = 1

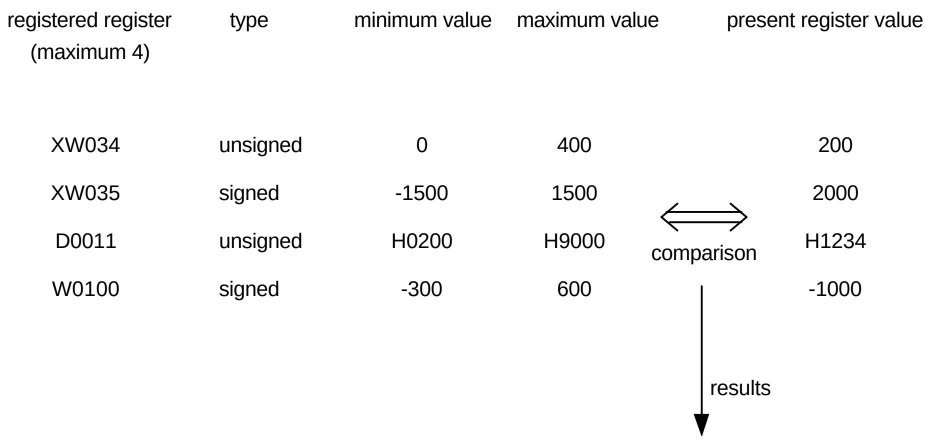
In the pattern registration, OFF is shown as○ , ON is shown as● and do not care is shown as X.

The device and bit pattern registration takes place in programmer system diagnosis mode.

(3) Register value validity check function

This function checks that the register value is within the specified numerical value range. There can be up to a maximum of 4 registers, a minimum and maximum value is registered for each. Also, it is possible to select the register value to be taken as an integer (signed) or as a positive integer (unsigned).

The check is carried out immediately before starting a scan, the results are stored in the special relay S0143-S0146 (within the range: 0, outside the range: 1).



registered register (maximum 4)	type	minimum value	maximum value	present register value
XW034	unsigned	0	400	200
XW035	signed	-1500	1500	2000
D0011	unsigned	H0200	H9000	H1234
W0100	signed	-300	600	-1000

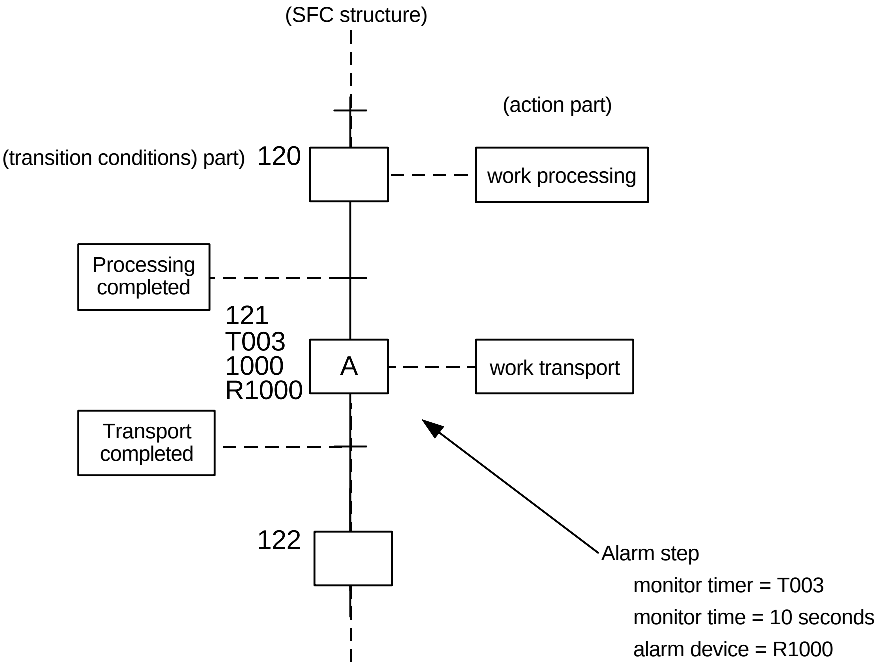
register 1 (XW034).... S0143 = 0
register 2 (XW035).... S0144 = 1
register 3 (D0011)..... S0145 = 0
register 4 (W0100).... S0146 = 1

The register and the numerical value range are registered in programmer system diagnosis mode.

(4) Sequence time over detection function

The alarm step is provided for one of SFC (sequential function chart) instructions. This Alarm step turns ON the specified device when the following transition is not come true within the preset time from the start of the step.

This function allows easy detection of operation holds up in sequential control process.



With the above example, if the transport has not been completed (work arrived signal ON etc) within 10 seconds, the specified alarm device (R1000) comes ON. By this means a malfunction generated by the work drive or the sensor can be detected.

Refer to Part 3 of this manual and the other instruction set manual for explanation with respect to SFC.

PART 3
USER PROGRAMS

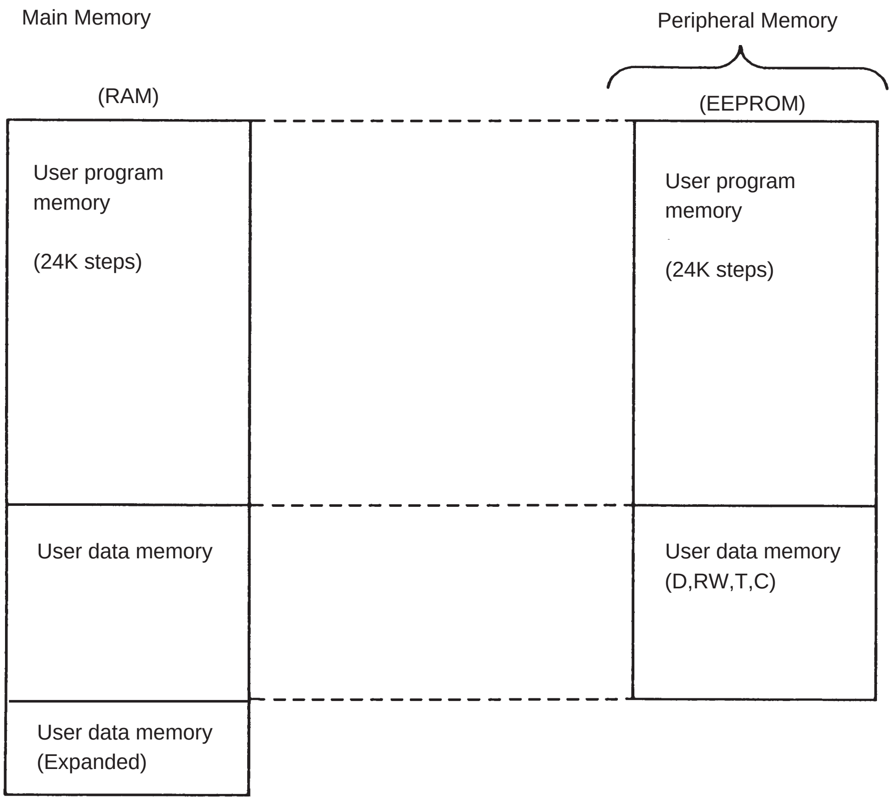
1.1 Aims of Part 3

The main functions of the T2N are to store the user program, to execute the stored user program and to control and monitor the operation/state of machines/processes which are the result of such execution.
The user program is a series of instructions describing operation sequences, operation conditions, data processing and the interface with the operator for achieving the requested machines/processes control. It is stored in the user program memory. The execution of the user program is the sequential performance of the processes of reading user data in which external input/output data and control parameters are stored, processing the respective instructions and storing the results of this in the user data memory.

Part 2 described the types of processing which are executed by the T2N internally, how the user program is executed, and also the internal configuration of the T2N and the types of functions which the T2N supports to maintain the machines/processes which are controlled by the T2N in the correct state. Part 3 describes the required information for creating user programs, that is to say detailed user data, detail of the input/output allocation and the programming languages. Also, the user program configuration in the T2N is described in order.

1.2 User Memory Configuration

The following diagram shows the user memory configuration of the T2N.



The memory which can be used by user is called user memory. The user memory can be divided by configuration into main memory and peripheral memory. And the user memory can be devided by function into user program memory and user data memory.

The main memory is a built-in RAM memory with battery backed up. On the other hand, the peripheral memory is a memory configured by EEPROM. The peripheral memory can be used as back up for main memory (user program and register data) .

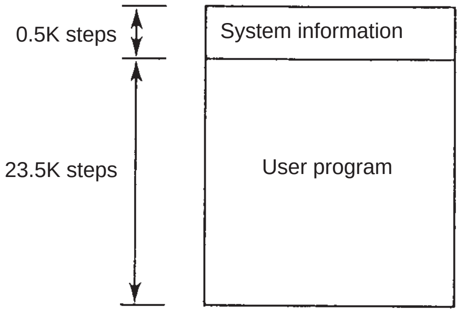
The user program memory has a capacity of 24K steps (step is a unit for instruction storage), and stores the user program configured by a series of instructions.

The user data memory stores variable data for user program execution. It is separated by function into input/output registers, data registers, etc.

2.1 Overview

The user program memory can be divided into the system information storage area and the user program storage area, as shown below.

User Program Memory Configuration



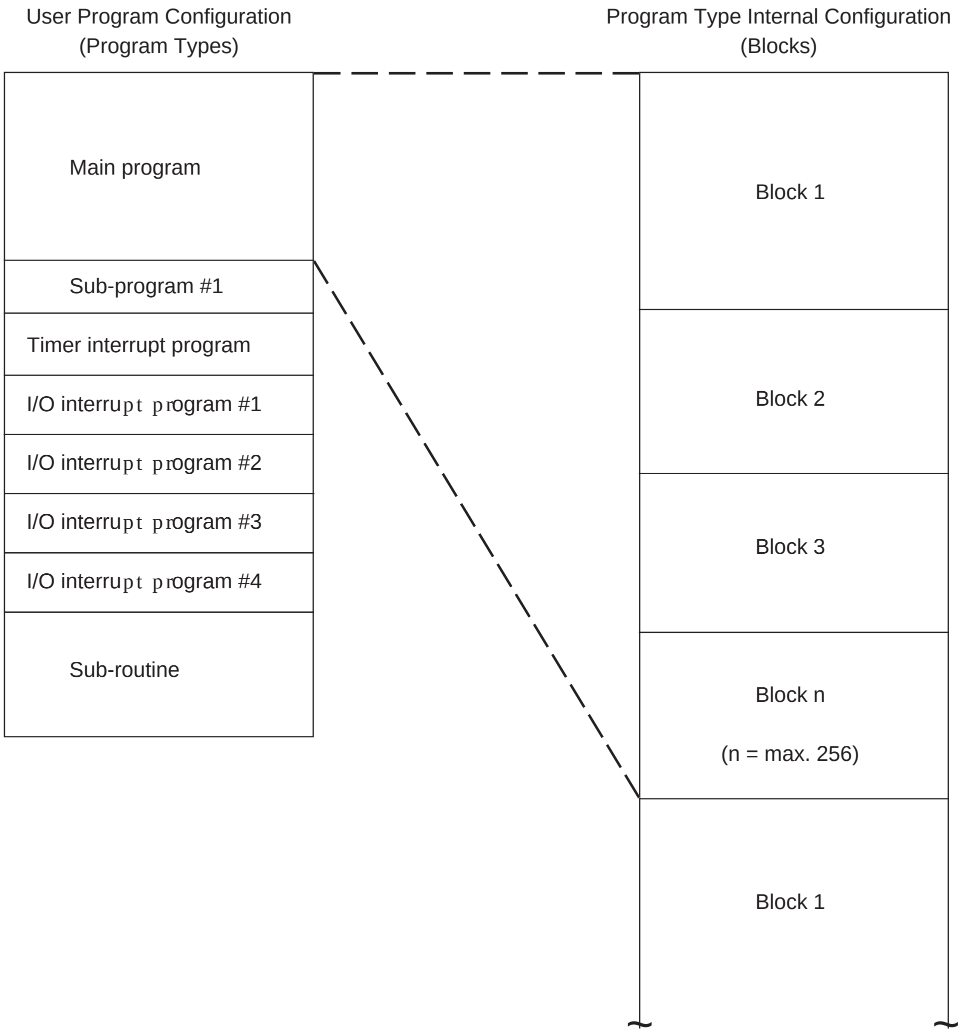
System information is the area which stores execution control parameters for executing the user program and user program management information, and it always occupies 0.5K steps.

The user program is divided by the program types into main program, sub-program, interrupt programs and sub-routines, depending on the function.

Of these, the main program is the core of the user program.

On the other hand, when it is difficult to achieve the requested control functions by the main program alone, sub-program and interrupt programs are used as required, but need not be provided.

Also, sub-routines are used when repetition of the same process in a program is required, or in order to see the program more easily by making one function into a block, but may not be provided if not required.



Also, in each program type, the user program is arranged by units called 'blocks'.

Internally, a block definition label is present at the head of each block. The program type, block number and programming language information are in the block definition label (there is no need for the user to be concerned with the block definition label).

Although the 2 programming languages of ladder diagram and SFC can be used in combination in the T2N, only 1 language can be used in any 1 block.

NOTE
▼△▼

1. In each program type and block, there is no limit to the program capacity (number of steps). The only limit is the total capacity (23.5K steps).
2. The block numbers need not be consecutive. In other words, there may be vacant blocks in the sequence.

2.2 System Information

System information is the area which stores execution control parameters and user program management information when executing a user program, and occupies 0.5K of the user program memory. The following details are included in system information.

(1) Program ID
This is the user program identification. A setting of up to 10 alphanumeric characters can be set. The program ID can be registered/monitored on the system information screen of the programmer.

(2) System Comments
These are comments attached to the user program. A setting of up to 30 alphanumeric characters can be set. The system comments can be registered/monitored on the system information screen of the programmer.

(3) Memory Capacity
This stores the memory type (user program capacity/data register capacity). Since the system side will be automatically registered, registration by the user is not required. The memory capacity can be monitored on the system information screen of the programmer.

(4) Steps Used
This stores the number of steps used in the user program. Since the system side performs automatic up-dating every time a user program is written, registration by the user is not required. The number of steps used can be monitored on the system information screen of the programmer.

(5) PLC Type
This stores the model type. Since the system side performs automatic registration, registration by the user is not required. The PLC type can be monitored on the system information screen of the programmer.

(6) Program Size Setting
The T2N is fixed at 24K steps. The program capacity can be registered/monitored on the system information screen of the programmer. (no need to set for T2N)

(7) Sampling Buffer Setting
This performs the setting and registration of the storage capacity of the sampling data from the sampling trace function. In the T2N, it is fixed at 8K words. The sampling buffer capacity can be monitored on the system information screen of the programmer. (Setting is not needed in the T2N.)

(8) Retentive memory area Designation
This sets and registers the address limits for the auxiliary register (RW), timer register(T), counter register(C) and data register(D) which retain pre-power cut data out of the user data when there is a power cut/power restoration. The limits registered here are outside the subjects of the user initialization process. For each of these registers, the limits from the leading address(0) to the designated address are the retentive memory areas. The power cut retention limit designations can be registered /monitored on the system information screen of the programmer.

(9) 10ms Timer Range Setting
This is invalid (setting is not needed.) in the T2N.

(10) Start Mode
This is invalid (setting is not needed.) in the T2N.

(11) Scan Time Setting
This sets and registerd the scan mode (floating/constant). When no scan time is registered (blank), the mode becomes the floating scan mode. When a numerical value is set for the scan time, the mode becomes a constant scan mode which takes that time as the scan cycle. The setting for the scan cycle is 10-200ms (in 10ms units). The scan time setting can be registered/monitored on the system information screen of the programmer.

(12) Sub-Program Execution Time
This is invalid (setting is not needed.) in the T2N.

(13) Fixed cycle Timer Interrupt Interval
This sets and registers the interrupt cycle of the time interrupt program. The setting limits are 5-1000ms (in 5ms units). The fixed timer interrupt cycle can be registered /monitored on the system information screen of the programmer.

(14) Computer Link Parameters

This sets and registers the parameters when using enhanced communication functions (Computer link, Data link, Free ASCII port). These parameters can be registered/monitored on the system information screen of the programmer.

The parameter items and their setting limits are as follows.

i) Computer link, Free ASCII port

• Station No.	...	1-32 (initial value = 1)
• Baud rate (bps)	...	300, 600, 1200, 2400, 4800, 9600, 19200(initial value = 9600)
• Parity	...	None, odd, even (initial value = odd)
• Data length(bits)	...	7,8 (initial value = 8)
• Stop bit	...	1,2 (initial value = 1)

ii) Data link

• Station No.	...	1 (initial value = 1)	: Master station
	...	2 (-32)	: Slave station

(15) Input/Output Allocation Information

This stores input/output allocation information and unit leading address designation information. This information is created either by executing the automatic I/O allocation command or by setting and registering an I/O module type for each slot (manual I/O allocation on the I/O allocation information screen of the programmer.)

(16) Network Assignment Information

Information on the link register areas allocated to data transmission stations (TOSLINE-S20LP, TOSLINE-F10) and information on the data input/output methods are stored here. The network assignment information can be registered/monitored on the network assignment information screen of the programmer.

2.3 User Program

The user program is composed of each of the program types of main program, sub-program #1, interrupt program (Timer and I/O #1-#4) and sub-routines. Of these program types, a main program must always be present. However, the other program types may not be present at all if they are not used. Therefore, needless to say, a user program can be configured with a main program only.

Also, among the program types, the programs can be divided into units called ‘blocks’. Block division is required in the following cases.

- * When using languages other than ladder diagrams (1 language/ block)
- * When creating multiple SFC programs (1 SFC/block, see Section 5.3)
- * When block division by control function units makes the program easier to see.

There are no restrictions on program capacities (numbers of steps) by program types and blocks. (Except in the case of SFC)

As block numbers, 1 to 256 are available. However, the block numbers need not be consecutive. When executing the program, the program is executed in sequence from the block with the lowest number.

In programming, the program type and block number is designated by the program read function of the programmer, and the specified portion is displayed on the screen. Then, the required program editing can be performed.

NOTE
▼△▼

Whether it is possible to use ladder diagram and SFC is shown below by program types.

Program Type	Ladder	SFC
Main program	Yes	Yes
Interrupt program	Yes	No
Sub-program	Yes	No
Sub-routine	Yes	No

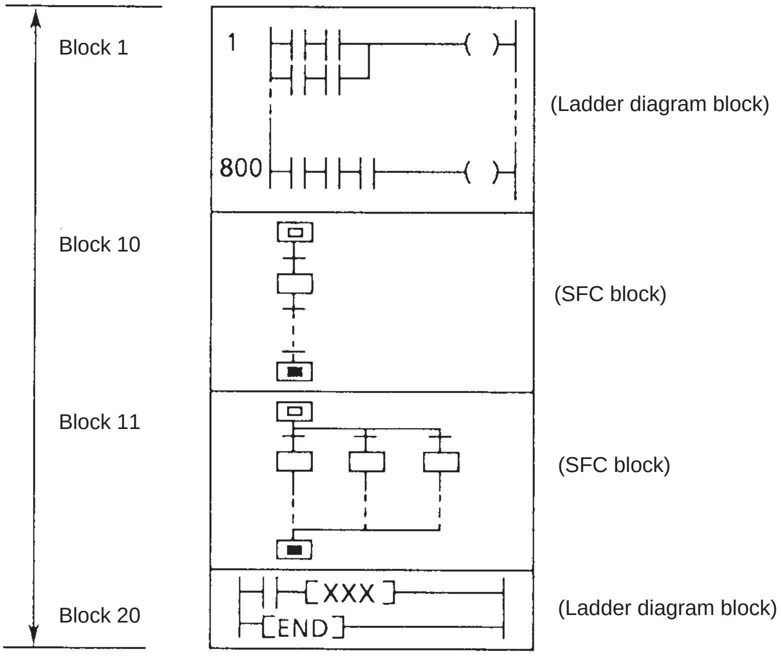
2.3.1 Main Program

The main program is the portion which is the core of the user program and is always executed every scan.

The main program must be ended by an END instruction.

Although instructions may be present after the END instruction, these portions will not be executed. (However, they count in the number of steps used)

(Example of Main Program Configuration)



2.3.2 Sub-Program

Sub-program #1 can be created which is executed only once at the head of the first scan when the T2N starts to RUN.

Therefore, when used for the initial value setting of registers, in order that it may not be programmed in the main program, it also has the effect of reducing the total scan time.

The sub-program #1 must be ended by an END instruction.

2.3.3 Interrupt Program

There are a total of 5 types of interrupt program. These are 1 fixed-cycle interrupt program which is executed cyclically with specified cycle time in system information, and 4 I/O interrupt programs (#1 - #4) which are started by interrupt signals from I/O modules with interrupt functions.

- Fixed cycle timer interrupt program
  This is executed cyclically with a cycle which is registered in the system information. The allowable setting is 5 to 1000ms (5ms units). When no cycle is registered (blank), it is not executed.
- I/O interrupt programs (#1 - #4)
  These are started by interrupt signals generated by I/O modules with the corresponding interrupt functions. The coordination between the interrupt program numbers and the I/O modules with interrupt function is fixed as shown below.

NOTE
▼△▼

For details of interrupt program operation, see Part 2 Section 3.3.

When automatic I/O allocation is executed in the state with interrupt I/O* mounted, for coordination between the interrupt program munber and the interrupt I/O, the lower number I/O interrupt programs are allocated in sequence from the interrupt I/O closest to the CPU.

*) I/O modules with interrupt are under development

Each interrupt program must be ended by an IRET instruction.

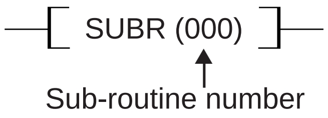
2.3.4 Sub-Routines

When it is necessary to execute repetitions of the same process in a program, this process can be registered as a sub-routine. This sub-routine can be executed by calling it (this is referred to as ‘sub-routine calling’) at the required location. By this means, the number of program steps can be reduced and, at the same time, the program becomes easier to see since the functions have been put in order.

Sub-routines can be called from other program types (main program, sub-program, interrupt program) and from other sub-routines (they can also be called from the action part portion of SFC).

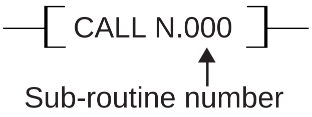
A sub-routine must be started by a SUBR instruction, and be ended by a RET instruction.
A maximum of 256 sub-routines can be programmed.

It is necessary to assign a sub-routine number to the SUBR instruction (sub-routine entry instruction). The limits of effective numbers are from 0 to 255.

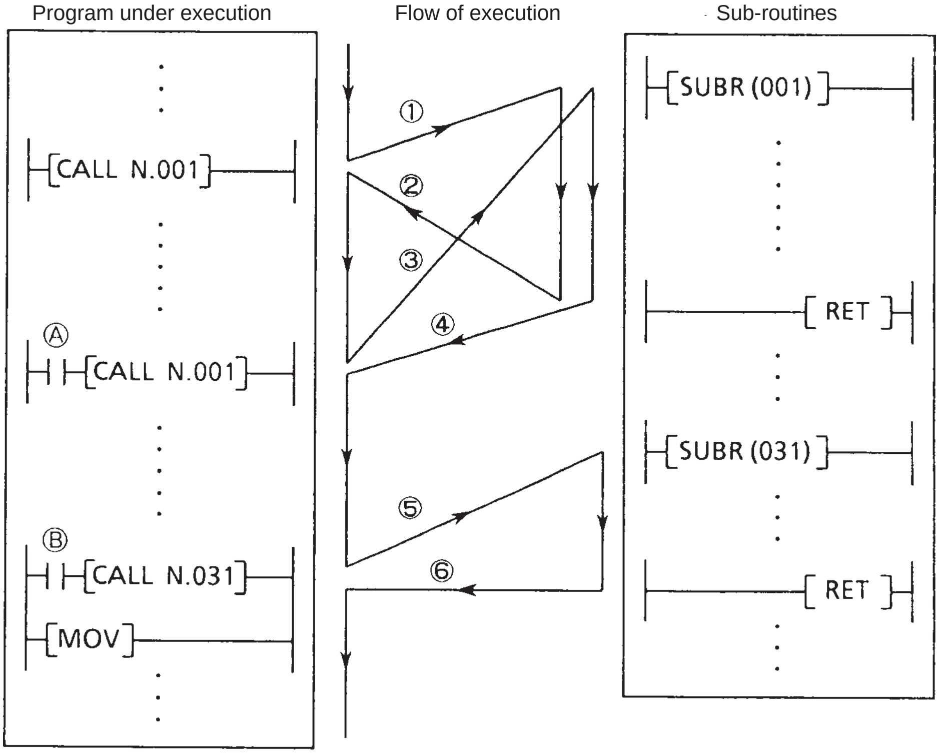


The RET instruction (sub-routine return instruction) has no sub-routine number.

The instruction which calls a registered sub-routine is the CALL instruction (sub-routine call instruction) of ladder diagrams. The CALL instruction requires the number of the sub-routine it calls.



The following is an execution sequence when sub-routines are included.



① By the sub-routine 001 CALL instruction execution, the execution shifts to sub-routine 001

② When it has proceeded to the RET instruction, the execution returns to the instruction following the CALL instruction in ①

③ When device Ⓐ is ON, the CALL instruction is executed, and the execution shifts to sub-routine 001

④ When it has proceeded to the RET instruction, the execution returns to the instruction following the CALL instruction in ③

⑤ When device Ⓑ is ON, the CALL instruction is executed, and the execution shifts to sub-routine 031

⑥ When it has proceeded to the RET instruction, the execution returns to the instruction following the CALL instruction in ⑤ (the MOV instruction in this example)

NOTE
▼△▼

1. Multiple sub-routines can be programmed in a block. However for execution monitor by programmer, 1 sub-routine on 1 block is recommended.
2. SFC cannot be used in a sub-routine.
3. Other sub-routines can be called from a sub-routine (nesting), up to 6 layers.
4. Since the operation will become abnormal in cases such as calling the same sub-routine during the execution of a sub-routine, take care that the following do not occur.
   * The case of an interrupt occurring during the execution of a subroutine by the main program and the same sub-routine being called in the interrupt

3.1 Overview

The area which stores the external input/output data, current values of timer or counter which are used in user program and the values of the variables for data processing is called the ‘user data’ area.

For user data, the storage location of the data is expressed by a combination of ‘function type’ and a sequence of numbers which starts from 0 (this is called the ‘address’).

Example) XW 005

Address 005 (in this case it is the register address)
Function type XW = Input register

To say that the content of XW005 is 100 is to say that the numerical value 100 is stored in a location in the user data memory indicated by XW005.

Also, user data is divided into registers and devices according to the type of data to be stored. (Although the expression ‘relay’ is also used, a relay should be regarded as one type of device)

A ‘register’ is area which stores 16 bits of data (provided it is a positive integer, the register can express any numerical value from 0 to 65535) and it is expressed as a combination of a function type and a register address. (the register address is a decimal number)

Example) D 1024

Register address (decimal number)
Function type D = Data register

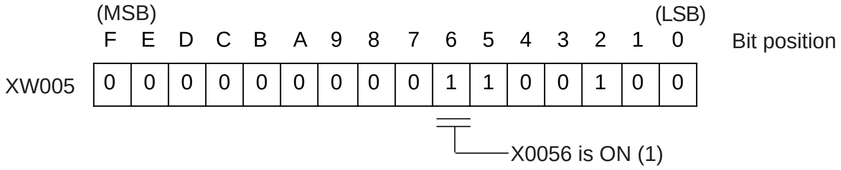
On the other hand a ‘device’ is an area which stores 1 bit of data (it expresses 1 or 0, in other words ON or OFF), and it is expressed as a combination of a function type and a device address. However, a device does not use an independent memory area. It is allocated as 1 bit in the 16 bits of the corresponding register. Therefore, the device address is expressed in the form of the corresponding register address + bit position.

Example) X 005 6

Bit position (there are 16 positions 0-F)
Register address (decimal number)
Function type X = Input device (corresponds to input register XW

The correspondence between register data and device data should be considered as follows.

Example) When it is said that the content of XW005 is 100, since the decimal number 100 is expressed as 1100100 in binary notation, this indicates that each of the bits of XW005 will be as follows.



	(MSB) F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	(LSB) 0	Bit position
XW005	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	

At this time, the data of device X0056 corresponding to bit position “6” of XW005 is 1, that is to say X0056 is ON.

The correspondence of registers and devices is shown by function types.

- Input device (X) ... corresponds to 1 bit of input register (XW)
- Output device (Y) ... corresponds to 1 bit of output register (YW)
- Auxiliary device (R) ... corresponds to 1 bit of auxiliary register (RW)
- Special device (S) ... corresponds to 1 bit of special register (SW)
- Link device (Z) ... corresponds to 1 bit of link register (W)
- (but only in the leading 1000 words)
  Link relay (L) ... corresponds to 1 bit of link register (LW)

The treatment of the other devices, I, O, T. and C., is slightly different. It is described in detail in Section 3.2.

The following Table shows the types of registers and devices and their address ranges. Their functions and methods of use are described in Section 3.2.

Function Type	Type Code	Address Range	Quantity	Expression Example
Input register	XW	000~127	Total 128 words	XW001
Output register	YW			YW034
Direct input register	IW			IW001
Direct output register	OW			OW034
Input device	X	0000~127F	Total 2048 points	X001A
Output device	Y			Y0348
Direct input device	I			I0012
Direct output device	O			O0340
Auxiliary register	RW	000~255	256 words	RW100
Auxiliary device	R	0000~255F	4096 points	R1001
Special register	SW	000~255	256 words	SW014
Special device	S	0000~255F	4096 points	S0140
Timer register	T	000~511	512 words	T030
Timer device	T.	000~511	512 points	T.030
Counter register	C	000~511	512 words	C199
Counter device	C.	000~511	512 points	C.199
Data register	D	0000~8191	8192 words	D4055
Link register	W	0000~2047	2048 words	W0200
Link device	Z	0000~999F	16000 points	Z2001
Link relay register	LW	0000~255	256 words	LW123
Link relay	L	0000~255F	4096 points	L123F
File register	F	0000~1023	1024 words	F0500
Index register	I	None	1 word	I
	J	None	1 word	J
	K	None	1 word	K
Expanded File register	-	accessing by XFER instraction	24576 words	-

NOTE

In the T2N, 1 word is treated as equal to 16 bits, and the number of registers is counted in word units.

3.2 Registers and Devices

The following Tables describe the functions and address ranges for each function type of registers and devices.

Input Registers and Input Devices

Codes	Input registers... XW Input devices ... X
Addresses	Input registers... 000-127 (128 words) Input devices ... 0000-127F (2048 points) } Common use as output registers/output devices
Functions	These are allocated in the input module as register units (word units) by performing input/output allocation. The signal state inputted to the input module is stored in the corresponding input register by batch input/output timing (except for modules which have the designation i attached when allocating). An input device expresses 1 bit of the corresponding input register. The data of input registers/input devices basically do not change during 1 scan. However, when executing a direct I/O instruction (FUN235), data is read from the corresponding input module when the instruction is executed and is stored in an input register/input device (XW/X). Thus, the data changes during the scan.

Output Registers and Output Devices

Codes	Output registers ... YW Output devices ... Y
Addresses	Output registers ... 000-127 (128 words) Output devices ... 0000-127F (2048 points) } Common use as output registers/output devices
Functions	These are allocated in the output module as register units (word units) by performing input/output allocation. The data stored in the output register is written to the corresponding output module by batch input/output timing, and the state of the output signal of the output module is determined (except for modules which have the designation i attached when allocating). An output device expresses 1 bit of an output register.

Direct Input Registers and Direct Input Devices

Codes	Direct input registers ... IW Direct input devices ... I
Addresses	Direct input registers ... 000-127 (correspond to input registers (XW)) Direct input devices ... 0000-127F (correspond to input devices (X))
Functions	Direct input registers/direct input devices do not themselves indicate specific memories. When the instruction word which uses these registers/ devices is executed, they operate and read data directly from the input module corresponding to the address. These registers/devices are used when using the T2N by the direct input/output system (direct system) and not the batch input/output system (refresh system). Example) I0000 ┤├ NO contact instruction of I0000 When executing the instruction, the bit data corresponding to X0000 is read from the input module corresponding to XW000, and the instruction is executed by this data. (The X0000 data is not affected) –[IW005 MOV RW100]– Transfer instruction from IW005 to RW100 When executing the instruction, the word data corresponding to XW005 is read from the input module corresponding to XW005 and is transferred to RW100. (The XW005 data is not affected)

Direct Output Registers and Direct Output Devices

Codes	Direct output registers ... OW Direct output devices ... O
Addresses	Direct input registers ... 000-127 (correspond to input registers (YW)) Direct input devices ... 0000-127F (correspond to input devices (Y))
Functions	When instructions are executed using direct output registers/direct output devices, data is stored in the corresponding output registers/output devices (YW/Y). Then, this output register (YW) data is written directly to the corresponding output module. These registers/devices are used when using the T2N by the direct input/output system (direct system) and not the batch intput/output system (refresh system). Example) O0020 ─( )─┤ Coil O0020 When the instruction is executed, the data (ON/OFF data) corresponding to the left link state is stored in Y0020. Then the 16-bit data of YW002 is written to the corresponding output module.

Auxiliary Registers and Auxiliary Devices

Codes	Auxiliary registers ... RW Auxiliary devices... R
Addresses	Output registers... 000-255 (256 words) Output devices... 0000-255F (corresponding to one bit in a register, 4096 points)
Functions	These are general purpose registers/general purpose devices which can be used for temporary storage of execution results during a program.An auxiliary register is used for storing 16-bit data. An auxiliary relay indicates 1 bit in an auxiliary register. Auxiliary registers/relays can be designated as retentive memory areas.

Special Registers and Special Devices

Codes	Special registers... SW Special devices... S
Addresses	Special registers... 000-255 (256 words) Special devices... 0000-255F (corresponding to one bit in a register, 4096 points)
Functions	These are registers/devices which have special functions such as fault flags (Error down/Warning) which are set when the CPU detects a malfunction; timing relays and clock calendar data (year, month, day, hour, minute, second, day of week) which are updated by the CPU; flags/data which the user sets for executing operational control of the sub-programs. For details, see the Table

Timer Registers and Timer Devices

Codes	Timer registers... T Timer devices... T.
Addresses	Timer registers... 000-511 (512 words) Timer devices... 000-511 (512 points)
Functions	The timer registers are used together with timer instructions (TON, TOF, SS, TRG), and store elapsed time (increment system) when the timer is operating. Also, the timer devices are linked to the operation of the timer registers with the same address, and store the output results of timer instructions. T000 to T063 works as 0.01 sec timers and T064 to T511 works as 0.1 sec timers. The timer registers can be designated as retentive memory areas.

Counter Registers and Counter Devices

Codes	Counter registers... C Counter devices... C.
Addresses	Counter registers... 000-511 (512 words) Counter devices... 000-511 (512 points)
Functions	The counter registers are used together with counter instructions (CNT, U/D), and store the count current value when the counter is operating. Also, the counter devices are linked to the operation of the counter registers with the same address, and store the output results of counter instructions. The counter registers can be designated as power cut retention areas.

Data Registers

Code	D
Addresses	0000-8191 (8192 words)
Functions	General-purpose registers which can be used for such purposes as a temporary memory for arithmetic results and the storage of control parameters. Apart from the fact that bit designation is not possible, they can be used in the same way as auxiliary registers. Data registers can be designated as retentive memory areas. Also, when a peripheral memory is used, D0000 - D2047 become subjects for the initial load. In the P-RUN mode, data writing to D0000-D2047 is prohibited.

Link Registers and Link Device (TOSLINE-S20LP/30)

Codes	Link registers... W Link devices... Z
Addresses	Link registers... 0000-2047 (2048 words) Link devices... 0000-999F, corresponding to the leading 1000 words of the register, 16000 points)
Functions	Used for a data link by the TOSLINE-S20LP or the TOSLINE - 30. For the leading 1000 words (W0000-W0999) of the link registers, bit designation is possible as link register relays (Z0000-Z999F). For areas not allocated to TOSLINE-S20LP/30 even when it is used, they can be used in the same way as auxiliary registers and data registers.

Link Registers and Link Relays (TOSLINE-F10)

Codes	Link registers... LW Link relays... L
Addresses	Link registers... 000-255 (256 words) Link relays... 0000-255F (4096)
Functions	Used as remote I/Os by the TOSLINE-F10. When TOSLINE-F10 is not used, they can be used in the same way as auxiliary relays.

File Registers

Code	F
Addresses	0000-1023 (1024 words)
Functions	Can be used in the same way as data registers for such as storing control parameters and storing field collection data. Bit designation is not possible. The whole file register area is retained for power off. The T2N has additional 24K words (8192 words x 3 banks) expanded file registers in the memory. The expanded file registers can be read/written by using expanded data transfer instruction (XFER). The expanded file registers are not retentive.

Index Registers

Codes	I, J, K (3 types, 3 words)
Addresses	None
Functions	When registers (apart from index registers) are used by instructions, apart from the normal address designation system (direct address designation, for instance D0100), indirect designation (indirect address designation, for instance D0100.I) is possible by using the index registers. (If, for instance the content of I is 5, D0100. I indicates D0105) For indirect address designation, see Section 3.4.

Tables of special registers/special relays are shown below.

Map of all the special registers

Register	Content
SW000	Operation mode, error flag, warning flag
SW001	Flag related to CPU error
SW002	Flag related to I/O error
SW003	Flag related to Program error
SW004	Timing relay
SW005	Carry flag, Error flag
SW006	Flag related to error during program execution
SW007 ≀ SW013	Clock calendar data (Year, month, day, hour, minute, second, day of the week)
SW014	Reserved (for future use)
SW015	Flag related to periphral support
SW016 ≀ SW033	Registration for Diagnosis display (system diagnosis)
SW034 ≀ SW037	Annunciator relay (system diagnosis)
SW038	Programmer port response delay mode
SW039	Interrupt program execution status
SW040	HOLD device
SW041	Sub-program execution status
SW042 ≀ SW056	Reserved (for future use)
SW057	Communication port response delay mode for computer link function
SW058 ≀ SW067	Reserved (for future use)
SW068	Related to ehanced communication function
SW069	Reserved (for future use)
SW070 ≀ SW077	TOSLINE-30 Scan healthy status
SW078 ≀ SW93	TOSLINE-F10 Command/status
SW94 ≀ SW109	TOSLINE-F10 Scan error map

Map of all the special registers

Register	Content
SW110	TOSLINE-S20LP Station status
SW111	Reserved (for future use)
SW112 ≀ SW115	TOSLINE-S20LP On line map
SW116 ≀ SW119	Reserved (for future use)
SW120 ≀ SW123	TOSLINE-S20LP Stand-by map
SW124 ≀ SW127	Reserved (for future use)
SW128 ≀ SW191	Reserved (for future use)
SW192 ≀ SW255	Reserved (for future use)

Special Device	Name	Function
S0000	Operation mode	0: Initializing 4: HOLD mode B: D-STOP 1: HALT mode 6: ERROR mode D: S-HALT 2: RUN mode 9: D-HALT E: S-RUN 3: RUN-F mode A: D-RUN F: S-STOP
S0001		
S0002		
S0003		
S0004	CPU error (Down)	ON when error occurs (OR condition of related flag in SW001)
S0005	I/O error (Down)	ON when error occurs (OR condition of related flag in SW002)
S0006	Program error (Down)	ON when error occurs (OR condition of related flag in SW003)
S0007	EEPROM number of writing times execeeded (Warning)	ON when EEPROM number of writing times 100,000 exceeded (operation contiunes)
S0008	Conststant scan delay (Warning)	ON when actual scan time exceeds the constant scan time setting
S0009		Reserved (for future use)
S000A	Calendar LSI error (Warning)	On when calendar timer data fault (operation continues)
S000B		Reserved (for future use)
S000C	TOSLINE-30 error (Warning)	ON when TOSLINE-30 error (operation continues)
S000D	TOSLINE-F10 error (Warning)	ON when TOSLINE F10 error (operation continues)
S000E	TOSLINE-20LP error (Warning)	ON when TOSLINE-S20LP error (operation continues)
S000F	Battery voltage low (Warning)	ON when battery voltage low (operation continues)
S0010	System ROM error (Down)	ON when when system ROM error
S0011	Sytem RAM error (Down)	ON when system RAM error
S0012	Program memory error (Down)	On when program memory (RAM) error
S0013	EEPROM error (Down)	ON when EEPROM error
S0014		Reserved (for future use)
S0015	LP error (Down)	ON when language processor (LP) error
S0016		Reserved (for future use)
S0017		
S0018		
S0019		
S001A		
S001B		
S001C		
S001E		
S001F		
S001E		
S001F	Watch-dog timer error (Down)	ON when watch-dog timer error occurs

*1) This area is for reference only (Do not write)
*2) The error flag becomes ON and is so maintained through the occurrence of a cause (it is re-set when RUN starts-up)

Special Device	Name	Function
S0020	I/O bus error (Down)	ON when I/O bus error
S0021	I/O mismatch error (Down)	ON when I/O mismatch error (allocation information and mounting state do not agree)
S0022	I/O response error (Down)	ON when no I/O response
S0023	I/O parity error (Down)	ON when I/O data parity error occurs.
S0024		Reserved (for future use)
S0025	I/O interrupt error (Warning)	ON when unused I/O interrupt occurs (operation continues)
S0026	Special module error (Warning)	ON when fault occurs in special module (operation continues)
S0027		Reserved (for future use)
S0028		
S0029		
S002A		
S002B		
S002C		
S002D		
S002E		
S002F		
S0030	Program error	ON when program fault occurs (OR condition of SW006-related flag)
S0031	Scan time error (Down)	ON when scan cycle exceeds the limit value
S0032		Reserved (for future use)
S0033		
S0034		
S0035		
S0036		
S0037		
S0038	EEPROM Initialization	OFF:Normal ON:Initializing
S0039	EEPROM Error	OFF:Normal ON:Error
S003A		Reserved (for future use)
S003B		
S003C		
S003D		
S003E		
S003F		

*1) This area is for reference only (Do not write)

*2) The error flag becomes ON and is so maintained through the occurrence of a cause (it is re-set when RUN starts-up)

Special Device	Name	Function	
S0040	Timing relay 0.1sec	0.05sec OFF/0.05sec ON (Cycle 0.1sec)	All OFF when RUN starts up
S0041	Timing relay 0.2sec	0.1sec OFF/0.1sec ON (Cycle 0.2sec)	
S0042	Timing relay 0.4sec	0.2sec OFF/0.2sec ON (Cycle 0.4sec)	
S0043	Timing relay 0.8sec	0.4sec OFF/0.4sec ON (Cycle 0.8sec)	
S0044	Timing relay 1.0sec	0.5sec OFF/0.5sec ON (Cycle 1.0sec)	
S0045	Timing relay 2.0sec	1.0sec OFF/1.0sec ON (Cycle 2.0sec)	
S0046	Timing relay 4.0sec	2.0sec OFF/2.0sec ON (Cycle 4.0sec)	
S0047	Timing relay 8.0sec	4.0sec OFF/4.0sec ON (Cycle 8.0sec)	
S0048		Reserved (for future use)	
S0049			
S004A			
S004B			
S004C			
S004D			
S004E	Alway OFF	Always OFF	
S004F	Alway ON	Always ON	
S0050	CF (carry flag)	Used by instructions with carry	
S0050	ERF (Error flag)	ON through error occurrence when executing instructions (linked with each error flag of SW006)	
S0052		Reserved (for future use)	
S0053			
S0054			
S0055			
S0056			
S0057			
S0058			
S0059			
S005A			
S005B			
S005C			
S005D			
S005E			
S005F			

*) This area (except for S0050, S0051) is for reference only (writing is ineffective)

Special Device	Name	Function
S0060	Illegal instruction detection (Down)	ON when illegal instruction detected
S0061		Reserved (for future use)
S0062		
S0063		
S0064	Boundary error (Warning)	ON when address range exceeded by indirect address designation (operation continues)
S0065	Address boundary error (Warning)	ON when destination (indirect) error by CALL instruction or JUMP instruction (operation continues)
S0066		Reserved (for future use)
S0067		
S0068	Division error (Warning)	ON when error occurs by division instruction (operation continues)
S0069	BCD data error (Warning)	ON when fault data detected by BCD instruction (operation continues)
S006A	Table operation error (Warning)	ON when table limits exceeded by table operation instruction (operation continues)
S006B	Encode error (Warning)	ON when error occurs by encode instruction (operation continues)
S006C	Address registration error (Warning)	ON when destination by CALL instruction or JUMP instruction unregistered (operation continues)
S006D	Nesting error (Warning)	ON when nesting exceeded by CALL instruction, FOR instruction or MCSn instruction (operation continues)
S006E		Reserved (for future use)
S006F		

*1) Becomes ON and is so maintained through the occurrence of a cause (it is re-set when RUN starts-up)

*2) Re-setting of warning flag executed by user program as required.

Special Register	Name	Function	
SW007	Calendar data (Year)	Last 2 digits of the calendar year (98, 99, 00, 01, ...)	The lower 8 bits are stored in BCD code
SW008	Calendar data (Month)	Month (01-12)	
SW009	Calendar data (Day)	Day (01-31)	
SW010	Calendar data (Hour)	Hour (00-23)	
SW011	Calendar data (Minute)	Minute (00-59)	
SW012	Calendar data (Second)	Second (00-59)	
SW013	Calendar data (Day of the week)	Day of the week (Sunday = 00, Monday = 01, ... Saturday = 06)	
SW014		Reserved (for future use)	

*1) The calendar data setting is performed by calendar setting instruction or by calendar setting operation by programmer. (It is ineffective to write data directly to the special registers)

*2) When the data cannot be read correctly due to a calendar LSI fault, these registers become H00FF.

*3) Calendar accuracy is ± 30 seconds/month.

Special Device	Name	Function
S0150		Reserved (for future use)
S0151		
S0152		
S0153		
S0154		
S0155		
S0156		
S0157		
S0158	Periphral support priority	Periphral support processing has been carried out in one scan when ON.
S0159		Reserved (for future use)
S015A		
S015B		
S015C		
S015D		
S015E		
S015F		

Special Register	Name	Function
SW016	First error code	· The designated error code (1-64) are stored in order of execution in SW018-SW033 (the earlier the code, the lower the address), and the number of registration (SW017) is updated. · The earliest error code occuring in the registered error codes (the content of SW018) is stored in the leading error code (SW016). · The registered error codes are cancelled one by one by the execution of the diagnostic display re-set instruction or by a re-set operation by the programmer. At this time, the number of registers is reduced by one and the storage positions of the error codes are shifted up.
SW017	Number of registration	
SW018	Error code first	
SW019	Error code (2)	
SW020	Error code (3)	
SW021	Error code (4)	
SW022	Error code (5)	
SW023	Error code (6)	
SW024	Error code (7)	
SW025	Error code (8)	
SW026	Error code (9)	
SW027	Error code (10)	
SW028	Error code (11)	
SW029	Error code (12)	
SW030	Error code (13)	
SW031	Error code (14)	
SW032	Error code (15)	
SW033	Error code (16)	

Special Device	Name	Function
S0340	Annunciator relay 1	・The annunciator relays corresponding to the error codes registered in SW018-SW033 become ON.
S0341	Annunciator relay 2	
S0342	Annunciator relay 3	
S0343	Annunciator relay 4	
S0344	Annunciator relay 5	
S0345	Annunciator relay 6	
S0346	Annunciator relay 7	
S0347	Annunciator relay 8	
S0348	Annunciator relay 9	
S0349	Annunciator relay 10	
S034A	Annunciator relay 11	
S034B	Annunciator relay 12	
S034C	Annunciator relay 13	
S034D	Annunciator relay 14	
S034E	Annunciator relay 15	
S034F	Annunciator relay 16	

Special Device	Name	Function
S0350	Annunciator relay 17	・The annunciator relays corresponding to the error codes registered in SW018-SW033 become ON
S0351	Annunciator relay 18	
S0352	Annunciator relay 19	
S0353	Annunciator relay 20	
S0354	Annunciator relay 21	
S0355	Annunciator relay 22	
S0356	Annunciator relay 23	
S0357	Annunciator relay 24	
S0358	Annunciator relay 25	
S0359	Annunciator relay 26	
S035A	Annunciator relay 27	
S035B	Annunciator relay 28	
S035C	Annunciator relay 29	
S035D	Annunciator relay 30	
S035E	Annunciator relay 31	
S035F	Annunciator relay 32	
S0360	Annunciator relay 33	
S0361	Annunciator relay 34	
S0362	Annunciator relay 35	
S0363	Annunciator relay 36	
S0364	Annunciator relay 37	
S0365	Annunciator relay 38	
S0366	Annunciator relay 39	
S0367	Annunciator relay 40	
S0368	Annunciator relay 41	
S0369	Annunciator relay 42	
S036A	Annunciator relay 43	
S036B	Annunciator relay 44	
S036C	Annunciator relay 45	
S036D	Annunciator relay 46	
S036E	Annunciator relay 47	
S036F	Annunciator relay 48	

Special Device	Name	Function
S0370	Annunciator relay 49	・The annunciator relays corresponding to the error codes registered in SW018-SW033 become ON
S0371	Annunciator relay 50	
S0372	Annunciator relay 51	
S0373	Annunciator relay 52	
S0374	Annunciator relay 53	
S0375	Annunciator relay 54	
S0376	Annunciator relay 55	
S0377	Annunciator relay 56	
S0378	Annunciator relay 57	
S0379	Annunciator relay 58	
S037A	Annunciator relay 59	
S037B	Annunciator relay 60	
S037C	Annunciator relay 61	
S037D	Annunciator relay 62	
S037E	Annunciator relay 63	
S037F	Annunciator relay 64	

Special Register	Name	Function
SW038	Programmer port response delay mode	The T2N sends back the response on the programmer port after waiting for specified time (value *10ms) specified value range : 0-30

Special Device	Name	Function
S0390	Timer interrupt execution status	ON during execution
S0391	I/O interrupt #1 execution status	ON during execution
S0392	I/O interrupt #2 execution status	ON during execution
S0393	I/O interrupt #3 execution status	ON during execution
S0394	I/O interrupt #4execution status	ON during execution
S0395		Reserved (for future use)
S0396		
S0397		
S0398		
S0399		
S039A		
S039B		
S039C		
S039D		
SO39E		
S036F		

Special Device	Name	Function
S0400		Reserved (for future use)
S0401	HOLD device	ON during HOLD mode (transition to HOLD even if ON by program)
S0402		Reserved (for future use)
S0403		
S0404		
S0405		
S0406		
S0407		
S0408		
S0409		
S040A		
S040B		
S040C		
S040D		
S040E		
S040F		
S0410	Sub-program #1 execution status	ON during sub-program #1 execution
S0411		Reserved (for future use)
S0412		
S0413		
S0414		
S0415		
S0416		
S0417		
S0418		
S0419		
S041A		
S041B		
S041C		
S041D		
S041E		
S041F		

Special Register	Name	Function
SW042 ≀ SW056		Reserved (for future use)
SW057	Communication port response delay mode	The T2N sends back the response on the communication port after waiting for specified time (value*10ms) in the computer link mode. specified value range :0-30
SW058 ≀ SW067		Reserved (for future use)

Special Device	Name	Function
S0680 S0681 S0682 S0683 S0684 S0685 S0686 S0687	End text for Free ASCII mode (trailing code)	Can be changed as the trailing code when in the Free ASCII mode Initial value = 0DH
S0688	Two wired system mode	Can connect to two wired system when this flag is ON in the Free ASCll mode
S0689	FIS communication status	ON when communication is normal in the FIS mode.
S068A	FIS start up flag	The FIS mode is started up when this flag is set to ON.
S068B	FIS connection status	ON when the FIS connection is completed normally.
S068C	FIS operation mode flag	ON : default setting mode OFF : user specified setting mode
S068D	Link partner‘s operation mode	ON when the link partner‘s operation mode is RUN in the data link mode.
S068E	Data link status	ON when communication is normal in the data link mode.
S068F	Free ASCII reset	The Free ASCII mode is reset when this flag is set to ON.

Special Register	Name	Function
SW069		Reserved (for future use)

Special Relay	Name	Function
S0700	Scan healty map for TOSLINE-30	ON when W0000 transmission normal
S0701		ON when W0001 transmission normal
S0702		ON when W0002 transmission normal
S0703		ON when W0003 transmission normal
S0704		ON when W0004 transmission normal
S0705		ON when W0005 transmission normal
S0706		ON when W0006 transmission normal
S0707		ON when W0007 transmission normal
S0708		ON when W0008 transmission normal
S0709		ON when W0009 transmission normal
S070A		ON when W0010 transmission normal
S070B		ON when W0011 transmission normal
S070C		ON when W0012 transmission normal
S070D		ON when W0013 transmission normal
S070E		ON when W0014 transmission normal
S070F		ON when W0015 transmission normal
S0710		ON when W0016 transmission normal
S0711		ON when W0017 transmission normal
S0712		ON when W0018 transmission normal
S0713		ON when W0019 transmission normal
S0714		ON when W0020 transmission normal
S0715		ON when W0021 transmission normal
S0716		ON when W0022 transmission normal
S0717		ON when W0023 transmission normal
S0718		ON when W0024 transmission normal
S0719		ON when W0025 transmission normal
S071A		ON when W0026 transmission normal
S071B		ON when W0027 transmission normal
S071C		ON when W0028 transmission normal
S071D		ON when W0029 transmission normal
S071E		ON when W0030 transmission normal
S071F		ON when W0031 transmission normal

Special Device	Name	Function
S0720	Scan healty map for TOSLINE-30	ON when W0032 transmission normal
S0721		ON when W0033 transmission normal
S0722		ON when W0034 transmission normal
S0723		ON when W0035 transmission normal
S0724		ON when W0036 transmission normal
S0725		ON when W0037 transmission normal
S0726		ON when W0038 transmission normal
S0727		ON when W0039 transmission normal
S0728		ON when W0040 transmission normal
S0729		ON when W0041 transmission normal
S072A		ON when W0042 transmission normal
S072B		ON when W0043 transmission normal
S072C		ON when W0044 transmission normal
S072D		ON when W0045 transmission normal
S072E		ON when W0046 transmission normal
S072F		ON when W0047 transmission normal
S0730		ON when W0048 transmission normal
S0731		ON when W0049 transmission normal
S0732		ON when W0050 transmission normal
S0733		ON when W0051 transmission normal
S0734		ON when W0052 transmission normal
S0735		ON when W0053 transmission normal
S0736		ON when W0054 transmission normal
S0737		ON when W0055 transmission normal
S0738		ON when W0056 transmission normal
S0739		ON when W0057 transmission normal
S073A		ON when W0058 transmission normal
S073B		ON when W0059 transmission normal
S073C		ON when W0060 transmission normal
S073D		ON when W0061 transmission normal
S073E		ON when W0062 transmission normal
S073F		ON when W0063 transmission normal

Special Relay	Name	Function
S0740	Scan healty map for TOSLINE-30	ON when W0064 transmission normal
S0741		ON when W0065 transmission normal
S0742		ON when W0066 transmission normal
S0743		ON when W0067 transmission normal
S0744		ON when W0068 transmission normal
S0745		ON when W0069 transmission normal
S0746		ON when W0070 transmission normal
S0747		ON when W0071 transmission normal
S0748		ON when W0072 transmission normal
S0749		ON when W0073 transmission normal
S074A		ON when W0074 transmission normal
S074B		ON when W0075 transmission normal
S074C		ON when W0076 transmission normal
S074D		ON when W0077 transmission normal
S074E		ON when W0078 transmission normal
S074F		ON when W0079 transmission normal
S0750		ON when W0080 transmission normal
S0751		ON when W0081 transmission normal
S0752		ON when W0082 transmission normal
S0753		ON when W0083 transmission normal
S0754		ON when W0084 transmission normal
S0755		ON when W0085 transmission normal
S0756		ON when W0086 transmission normal
S0757		ON when W0087 transmission normal
S0758		ON when W0088 transmission normal
S0759		ON when W0089 transmission normal
S075A		ON when W0090 transmission normal
S075B		ON when W0091 transmission normal
S075C		ON when W0092 transmission normal
S075D		ON when W0093 transmission normal
S075E		ON when W0094 transmission normal
S075F		ON when W0095 transmission normal

Special Device	Name	Function
S0760	Scan healty map for TOSLINE-30	ON when W0096 transmission normal
S0761		ON when W0097 transmission normal
S0762		ON when W0098 transmission normal
S0763		ON when W0099 transmission normal
S0764		ON when W0100 transmission normal
S0765		ON when W0101 transmission normal
S0766		ON when W0102 transmission normal
S0767		ON when W0103 transmission normal
S0768		ON when W0104 transmission normal
S0769		ON when W0105 transmission normal
S076A		ON when W0106 transmission normal
S076B		ON when W0107 transmission normal
S076C		ON when W0108 transmission normal
S076D		ON when W0109 transmission normal
S076E		ON when W0110 transmission normal
S076F		ON when W0111 transmission normal
S0770		ON when W0112 transmission normal
S0771		ON when W0113 transmission normal
S0772		ON when W0114 transmission normal
S0773		ON when W0115 transmission normal
S0774		ON when W0116 transmission normal
S0775		ON when W0117 transmission normal
S0776		ON when W0118 transmission normal
S0777		ON when W0119 transmission normal
S0778		ON when W0120 transmission normal
S0779		ON when W0121 transmission normal
S077A		ON when W0122 transmission normal
S077B		ON when W0123 transmission normal
S077C		ON when W0124 transmission normal
S077D		ON when W0125 transmission normal
S077E		ON when W0126 transmission normal
S077F		ON when W0127 transmission normal

Special Device	Name		Function
S0780	TOSLINE-F10 CH1 command	Transmission status	ON during transmission
S0781		Output inhibit status	ON when output inhibit mode
S0782		Re-configuration	ON during re-configuration
S0783			Reserved (for future use)
S0784		Scan transmission error	ON when scan transmission error occurs
S0785			Reserved (for future use)
S0786			
S0787			
S0788		Transmission stop	Transmission stop by setting ON
S0789		Output inhibit	Output inhibit by setting ON
S078A			Reserved (for future use)
S078B			
S078C			
S078D			
S078E			
S078F			
S0790	TOSLINE-F10 CH1 status	Transmission status	ON during transmission
S0791		Scan transmission	ON during scan transmission
S0792			Reserved (for future use)
S0793			
S0794			
S0795		MS operation mode	OFF:Normal mode ON:Test mode
S0796			Reserved (for future use)
S0797			
S0798			
S0799			
S079A			
S079B			
S079C			
S079D			
S079E			
S079F			

*) Refer to the TOSLINE-F10 manual for details.

Special Register	Name	Function
SW080	TOSLINE-F10 CH2 command	・Biti assignment in the register is the same as SW078 and SW079.
SW081	TOSLINE-F10 CH2 status	
SW082	TOSLINE-F10 CH3 command	
SW083	TOSLINE-F10 CH3 status	
SW084	TOSLINE-F10 CH4 command	
SW085	TOSLINE-F10 CH4 status	
SW086	TOSLINE-F10 CH5 command	
SW087	TOSLINE-F10 CH5 status	
SW088	TOSLINE-F10 CH6 command	
SW089	TOSLINE-F10 CH6 status	
SW090	TOSLINE-F10 CH7 command	
SW091	TOSLINE-F10 CH7 status	
SW092	TOSLINE-F10 CH8 command	
SW093	TOSLINE-F10 CH8 status	

Special Register	Name		Function
SW094	TOSLINE-F10 scan error map	LW000~LW015	・The corresponding bit comes ON when the LW register is not updated normally. ・The lowest adress of LW register corresponds to bit 0 in the SW register, and in the order.
SW095		LW016~LW031	
SW096		LW032~LW047	
SW097		LW048~LW063	
SW098		LW064~LW079	
SW099		LW080~LW095	
SW100		LW096~LW111	
SW101		LW112~LW127	
SW102		LW128~LW143	
SW103		LW144~LW159	
SW104		LW160~LW175	
SW105		LW176~LW191	
SW106		LW192~LW207	
SW107		LW208~LW223	
SW108		LW224~LW239	
SW109		LW240~LW255	

*) Refer to the TOSLINE-F10 manual for details.

Special Device	Name		Function
S1100	TOSLINE-S20LP station status	Test mode	ON when test mode
S1101			Reserved (for future use)
S1102			
S1103			
S1104		Master/slave	ON when master station
S1105		Scan inhibit	ON when scan transmission inhibit
S1106		Termination	ON when the terminated station runs
S1107		Repeat status	ON when repeat operation is executing
S1108		Loop2 transmission inhibit	ON when Loop2 transmission inhibit
S1109		Loop2 receive inhibit	ON when Loop2 receive inhibit
S110A		Loop1 transmission inhibit	ON when Loop1 transmission inhibit
S110B		Loop1 receive inhibit	ON when Loop1 receive inhibit
S110C		Online	ON when online mode
S110D		Standby	ON when standby mode
S110E		Offline	ON when offline mode
S110F		Down	ON when down mode
S1110			Reserved (for future use)
S1111			
S1112			
S1113			
S1114			
S1115			
S1116			
S1117			
S1118			
S1119			
S111A			
S111B			
S111C			
S111D			
S111E			
S111F			

*) Refer to the TOSLINE-S20LP manual for details.

Special Register	Name		Function
SW112	TOSLINE-S20LP Online map	station No.1~No.16	· The corresponding bit is ON when the station is online. · The lowest station number corresponds to bit 0 in the SW register, and in the order.
SW113		station No.17~No.32	
SW114		station No.33~No.48	
SW115		station No.49~No.64	
SW116			Reserved (for future use)
SW117			
SW118			
SW119			
SW120	TOSLINE-S20LP Standby map	station No.1~No.16	· The corresponding bit is ON when the station is standby. · The lowest station number corresponds to bit 0 in the SW register, and in the order.
SW121		station No.17~No.32	
SW122		station No.33~No.48	
SW123		station No.49~No.64	
SW124			Reserved (for future use)
SW125			
SW126			
SW127			

3.3 Processing Register Data

It has already been explained the a register is “a location which houses 16 bits of data”. In the T2N instruction words, the the following types of data can be processed using single registers or multiple consecutive registers.

- Unsigned integers (integers in the range 0 to 65535)
- Integers (integers in the range -32768 to 32767)
- BCD (integers in the range 0 to 9999 expressed by BCD code)
- Unsigned double-length integers (integers in the range 0 to 4294967295)
- Double-length integers (integers in the range -2147483648 to 2147483647)
- Double-length BCD (integers in the range 0 to 99999999 expressed by BCD code)
- Floating point data (real number in the range -3.40282×10^{38} to 3.40282×10^{38})

However, there are no dedicated registers corresponding to the types for processing these types of data. The processing of the register data varies according to which instruction word is used.

In other words, as shown in the following example, even when the same register is used, if the data type of the instruction word differs, the processing of the register data will also differ.

Example)

When the value of D0005 is HFFFF (hexadecimal FFFF):-

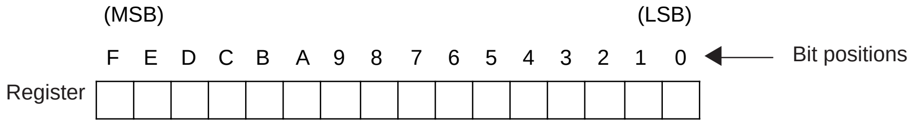
(1) In a comparison instruction (greater) without sign,
─[D0005 U > 100]─ decision output (ON when true)
The value of D0005 is regarded as 65535 (unsigned integer), therefore it is judged to be greater than the compared value (100) and the output of the instruction becomes ON.

(2) In a comparison instruction (greater) with sign,
─[D0005 > 100]─ decision output (ON when true)
The value of D0005 is regarded as -1 (integer), therefore it is judged not to be greater than the compared value (100) and the output of the instruction becomes OFF.

In this way, since there is no classification of registers by data type, it is possible to execute complex data operations provided their use is thoroughly understood. However, in order to make the program easier to see, it is recommended that registers be used by allocation by data types (1 register is processed by 1 data type) as far as possible.

(1) Unsigned Integer

This is a 16-bit unsigned integer expressed by 1 register. The bit configuration inside the register is as shown below



Bit 0 is the least significant bit (LSB), and bit F is the most significant bit (MSB). The processable numerical value ranges are as shown in the following Table.

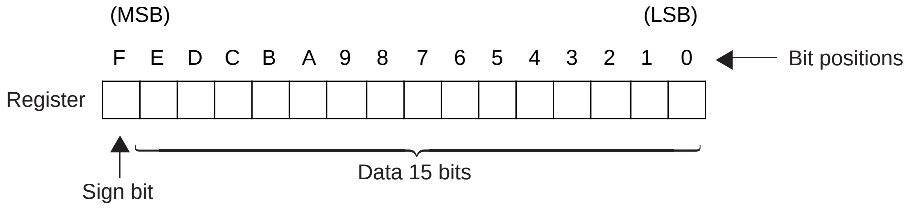
Numerical Value (Decimal)	Binary Expression	Hexadecimal Expression
65535	1111 1111 1111 1111	FFFF
65534	1111 1111 1111 1110	FFFE
∫	∫	∫
1	0000 0000 0000 0001	0001
0	0000 0000 0000 0000	0000

NOTE
▼△▼

When programming and when program monitoring, it is possible to shift between decimal numbers and hexadecimal numbers for displaying/setting register data. When using a hexadecimal display, “H” is attached before the numerical value.
Example) H89AB (hexadecimal 89AB)

(2) Integer

This is a 16-bit integer expressed by 1 register. A negative number is expressed by 2’s complements.



The numerical value is expressed by the 15 bits from bit 0 to bit E. Bit F expresses the sign (0 when positive, 1 when negative)

Processable numerical ranges and expression formats are shown in the following Table.

Numerical Value (Decimal)	Binary Expression	Hexadecimal Expression
32767	0111 1111 1111 1111	7FFF
32766	0111 1111 1111 1110	7FFE
∫	∫	∫
1	0000 0000 0000 0001	0001
1	0000 0000 0000 0000	0001
– 1	1111 1111 1111 1111	FFFF
∫	∫	∫
– 32767	1000 0000 0000 0001	8001
– 32768	1000 0000 0000 0000	8000

The numerical value when two complementary expressions are added together is a value in which the lower 16 bits are all 0.

Example)

```
         0111  1111  1111  1111      (Binary)  =  322767
         1000  0000  0000  0001      (Binary)  =  322767
    ------------------------------
    1    0000  0000  0000  0000
```

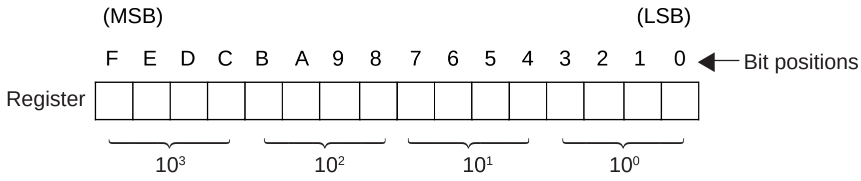
In calculation, the 2’s complements of a numerical value can be found by the operation of inverting each bit of that numerical value and adding 1.

Example)

```
0111  1111  1111  1111      (Binary)  =    32767
      (bit inversion)
1000  0000  0000  0001      (Binary)
         (add 1)
0000  0000  0000  0000      (Binary)  =  – 32767
```

(3) BCD

BCD is the abbreviation of Binary Coded Decimal. BCD expresses 1 digit (0 - 9) of a decimal number by 4 bits of a binary number. Therefore, 1 register can express the numerical value of a 4-digit decimal number.



Processable numerical ranges and expression formats are shown in the following Table.

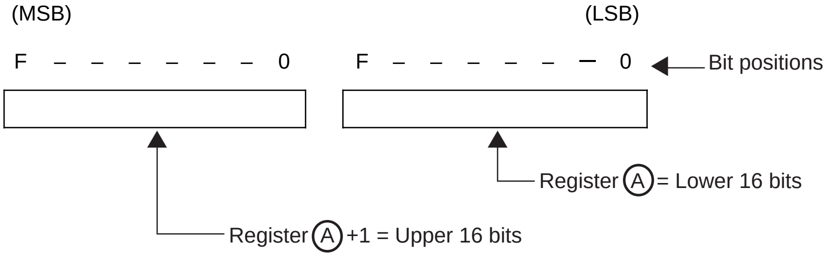
Numerical Value (Decimal)	Binary Expression	Hexadecimal Expression
9999	1001 1001 1001 1001	9999
9998	1001 1001 1001 1000	9998
∫	∫	∫
10	0000 0000 0001 0000	0010
9	0000 0000 0000 1001	0009
∫	∫	∫
1	0000 0000 0000 0001	0001
0	0000 0000 0000 0000	0000

NOTE

Basically, BCD is a data format used for data inputs from BCD-output type numerical setting devices and data outputs to BCD-input type numerical display devices. However, the T2N is provided with dedicated instructions which execute the 4 arithmetic calculations on BCD data as they stand.

(4) Unsigned Double-Length Integers

These are 32-bit unsigned integers which are expressed using 2 consecutive registers. In the case of double-length data, the registers are designated in the form Ⓐ + 1 • Ⓐ. Ⓐ indicates the lower 16 bits and Ⓐ + 1 shows the upper 16 bits. (Ⓐ + 1 is the register following register Ⓐ)



Example) When processing a Unsigned double-length integer in double length register D0201•D0200, D0200 becomes Ⓐ and D0201 becomes Ⓐ +1. D0200 becomes the lower side and D0201 becomes the upper side.
In programming, when D0200 is entered in the position which designates the instruction double-length operand, D0201•D0200 is automatically displayed.

The numerical value range in which unsigned double-length integers can be processed is shown in the Table on the following page.

Numerical Value	Hexadecimal Expression	
	Register Ⓐ+1	Register Ⓐ
4294967295	FFFF	FFFF
∫	∫	∫
65536	0001	0000
65535	0000	FFFF
∫	∫	∫
0	0000	0000

NOTE

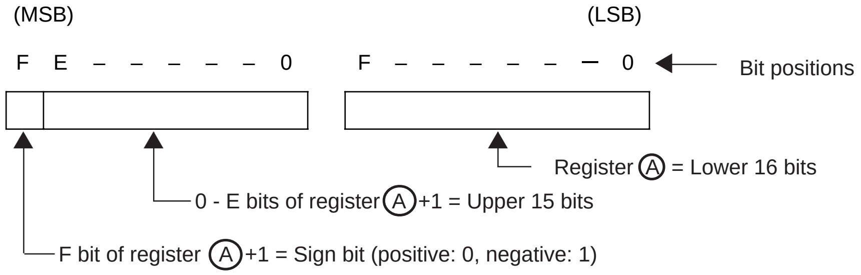
Both odd-numbered addresses and even-numbered addresses may be used as register A.

(5) Double-Length Integers

These are 32-bit integers which are expressed using 2 consecutive registers. Negative numbers are expressed by 2's complement. (See (2) ‘Integers’)

The registers are designated in the form Ⓐ+1•Ⓐ.

Ⓐ becomes the lower and Ⓐ+1 becomes the upper.


(MSB)
(LSB)
F E – – – – – 0
F – – – – – – 0
Bit positions
Register Ⓐ = Lower 16 bits
0 - E bits of register Ⓐ+1 = Upper 15 bits
F bit of register Ⓐ+1 = Sign bit (positive: 0, negative: 1)

The numerical value is expressed by the 31 bits from bit 0 of register Ⓐ to bit E of register Ⓐ +1. The sign is expressed by bit F of registerⒶ+1 (0 when positive, 1 when negative).

Example) When a double-length integer is processed by registers D1002•D1001, D1001 becomes Ⓐ and D1002 becomes Ⓐ+1, and D1001 is the lower and D1002 is the upper. Also, the sign is expressed by the bit F of D1002.

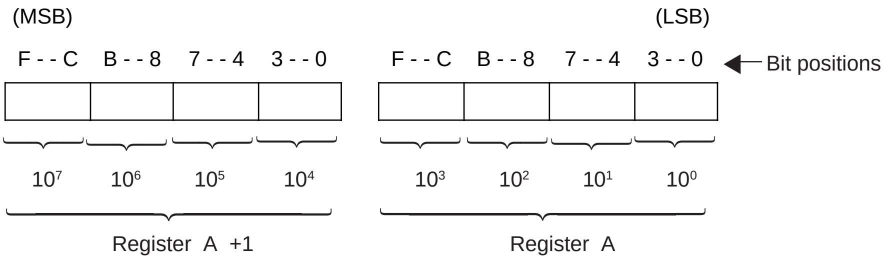
In programming, when D1001 is entered in the position which designates the instruction word double-length operand, D1002•D1001 is automatically displayed.

The numerical value range in which double-length integers can be processed is shown in the Table on the following page

Numerical Value	Hexadecimal Expression	
	Register A +1	Register A
2147483647	7FFF	FFFF
∫	∫	∫
65536	0001	0000
65535	0000	FFFF
∫	∫	∫
0	0000	0000
- 1	FFFF	FFFF
∫	∫	∫
- 65536	FFFF	0000
- 65537	FFFE	FFFF
∫	∫	∫
- 2147483648	8000	0000

(6) Double-Length BCD

This is 8-digit BCD data which is expressed by using consecutive registers.



The registers are designated in the form Ⓐ+1 Ⓐ, and Ⓐ becomes the lower 4 digits while Ⓐ+1 becomes the upper 4 digits.

Example) When processing a double-length BCD by registers XW001*XW000, XW000 becomes Ⓐ while XW001 becomes Ⓐ +1 and XW000 becomes the lower 4 digits while XW001 becomes the upper 4 digits.

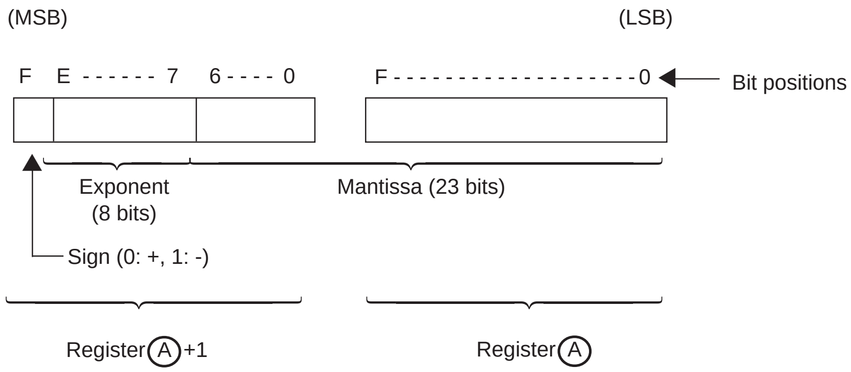
The following Table shows the numerical range and the expression format in which double-length BCD data can be processed.

Numerical Value	Hexadecimal Expression	
	Register A +1’	Register A
99999999	9999	9999
∫	∫	∫
1	0000	0001
0	0000	0000

(7) Floating Point Data

This is a real number which is expressed using 2 consecutive registers (32-bit).

The registers are designated in the form Ⓐ+1• Ⓐ. Internally, the following format is used. (conforms to IEEE754)



$$\text{Value} = (\text{Sign})1.(\text{Mantissa})\times 2^{(\text{Exponent-127})}$$

The floating point data is used with the following floating point instructions. Therefore, there is no need for user to consider the format.

- Conversions (Floating point↔Double-length integer)
- Floating point arithmetics
- Floating point comparisons

The following table shows the numerical range in which the floating point data can be processed.

Numerical Value	Expression	Remarks
3.40282×10^{38}	3.40282E38	Maximum
∫	∫	
1.17549×10^{-38}	1.17549E-38	Nearest to 0
0	0	
-1.17549×10^{-38}	- 1.17549E-38	Nearest to 0
∫	∫	
-3.40282×10^{38}	- 3.40282E38	Minimum

3.4 Index Modification

When registers are used by instructions, the method of directly designating the register address as shown in Example 1) below is called ‘direct addressing’.

As opposed to this, the method of indirectly designating the register by combination with the contents of the index registers (I, J, K) as shown in Example 2) below is called the ‘indirect addressing’. In particular, in this case, since the address is modified using an index register, this is called ‘index modification’.

Example 1)

```
─[ RW100 MOV D3500 ]─
```

Data transfer instruction
Transfer content of RW100 to D3500

Example 2)

```
          I                   J
─[ RW100 MVO D3500 ]─
```

Data transfer instruction (index modification attached)
Transfer content of RW(100 + I) to D(3500 + J)
(If I =3 and J = 200, the content of RW103 is transferred to D3700)

There are 3 types of index register - I, J and K. Each type processes 16-bit integers (-32768 to 32767). There are no particular differences in function between these 3 types of index register.

There is no special instruction for substituting values in these index registers. These are designated as normal transfer instructions or as destination for operation instructions.

Example 1) Substituting a constant in an index register

```
─[ 64 MOV I ]─   (Substitute 64 in index register I)

─[ - 2 MOV J ]─   (Substitute -2 in index register J)
```

Example 2) Substituting register data in an index register

```
─[ D0035 MOV K ]─(Substitute the value of D0035 in index register K)

─[ RW078 MOV I ]─   (Substitute the value of RW078 in index
                      register I)
```

Example 3) Substituting the result of an operation in an index register

–[RW200 - 30 → I]–

(Substitute the result of subtracting 30 from RW200 in I)

–[XW004 ENC (4) J]–

(Substitute the uppermost ON bit position of XW004 in J (encode))

NOTE
▼△▼

Although, basically, index registers are processed as single-length (16 bits), when, for instance, using an index register as the storage destination for a instruction which becomes double-length as the result of a multiplication instruction or the like, only the combinations J•I or K•J are effective. In this case, it becomes J•I by designating I in the double-length operand position, and J becomes upper while I becomes lower. In the same, by designating J, it becomes K•J, and K becomes upper while J becomes lower.
Example)

–[D1357 * 10 → J • I]–

The following are examples of registers in which index modification has been executed.

I
RW100

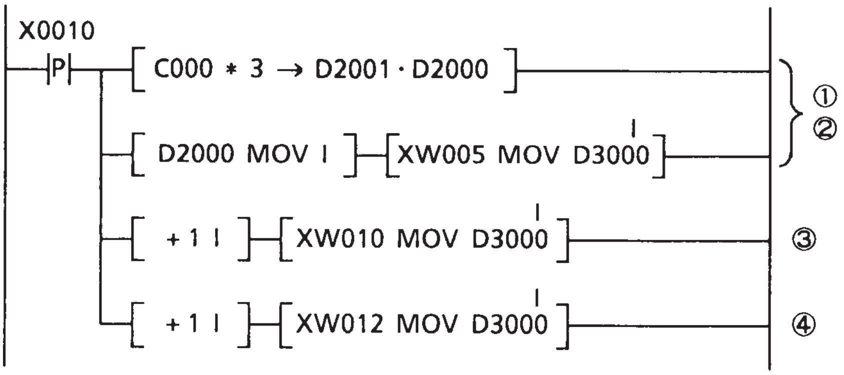
When I = 0,	expresses RW100
When I = 1,	expresses RW101
When I = -1,	expresses RW099
When I = 100,	expresses RW200
When I = -100,	expresses RW000

J
D0201 · D0200

When J = 0,	expresses D0201 · D0200
When J = 1,	expresses D0202 · D0201
When J = 2,	expresses D0203 · D0202
When J = -1,	expresses D0200 · D0199
When J = -2,	expresses D0199 · D0198

The following shows an example of the operation when index modification is applied to a program.

Example)



The following processing is carried out when X0010 changes from OFF to ON

① Substitute 3 times the value of the content of C000 in index register I
② Store content of XW005 in D(3000 + I)
③ Add 1 to the content of I and store content of XW010 in D(3000 + I)
④ Add a further 1 to the content of I and store content of XW012 in D(3000 + I)

Incidentally,

Ⓐ
─┤P├─is a positive pulse contact which becomes ON once only when device A starts-up from OFF to ON (up to the instruction execution of the following scan)

─[Ⓐ ⁕ Ⓑ → Ⓒ + 1• Ⓒ]─is multiplication instruction which multiplies Ⓐ by B and stores it in double-length register Ⓒ + 1• Ⓒ

─[+1 Ⓐ]─ is an increment instruction which adds 1 to the content of Ⓐ and stores it in Ⓐ

─[Ⓐ MOV Ⓑ]─ is a data transfer instruction which substitutes the content of Ⓐ in Ⓑ .

NOTE
▼△▼

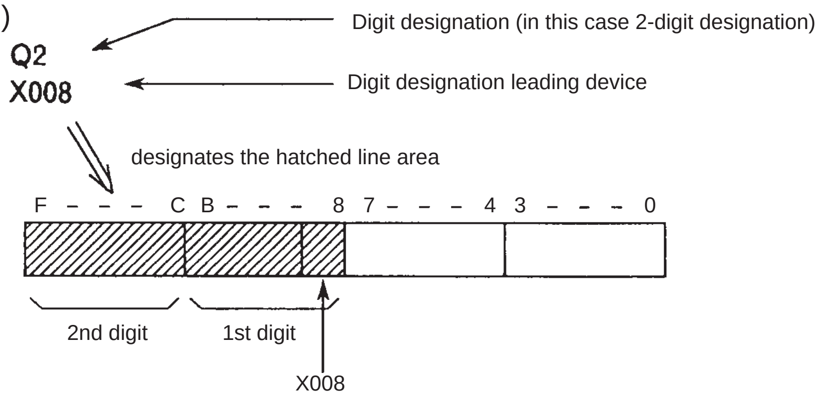
1. Substitutions of values to index registers and index modification may be carried out any number of times during a program. Therefore, normally, the program will be easier to see if a value substitution to an index register is executed immediately before index modification.
2. Be careful that the registers do not exceed the address range through index modification. When the results of index modification exceed the address range, the instruction becomes non-executable, and special relays (S0051 and S0064) which indicate ‘boundary error’ become ON.

3.5 Digit Designation

There is a method called 'digit designation' which is a special designation method for register data. 'Digit designation' treats 1 digit (4 bits) of a hexadecimal number as a data unit. It is a method of designation in which a number of digits from the designated devices (bit positions) are made the subject of data operation. Digit designation can be used by the transfer (FUN18) and data echange (FUN22) instructions.

In practice, in the case of the following Example, 2 digits from X0008 (that is to say, the upper 8 bits of XW000) become the subject of data operation.

Example)


Q2
X008
Digit designation (in this case 2-digit designation)
Digit designation leading device
designates the hatched line area
F – – – C B – – – 8 7 – – – 4 3 – – – 0
2nd digit
1st digit
X008

There are 5 types of digit designation - Q0, Q1, ..., Q4 which have the following significations

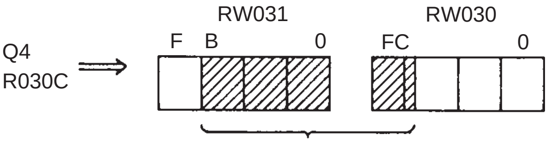
Q0 ... makes the designated device 1 bit the subject of data operation
Q1 ... makes 1 digit (4 bits) with the designated device as the lead the subject of data operation
Q2 ... makes 2 digits (8 bits) with the designated device as the lead the subject of data operation
Q3 ... makes 3 digits (12 bits) with the designated device as the lead the subject of data operation
Q4 ... makes 4 digits (16 bits) with the designated device as the lead the subject of data operation

NOTE
▼△▼

Q5 to Q8 cannot be used by the T2N.

In digit designation, when the area designated covers multiple registers, as shown below, the area is designated from the smaller address to the greater address.

Example)



The 16 bits R030C to R031B (R030C is the lowest position bit as a numerical value)

Below, the operation of digit designation is described for the case when digit designation is executed as a source operand (a register for executing a instruction using its data) and the case when digit designation is executed as a destination operand (a register which stores the result of instruction execution).

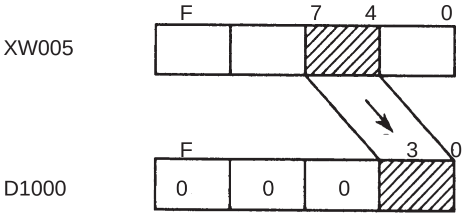
It is possible to carry out digit designation for both a reference operand and a transfer destination operand with 1 instruction.

(1) Digit Designation for MOV Instruction

Example 1)

```
     Q1
─[ X0054 MOV D1000 ]─   (data transfer instruction)
```

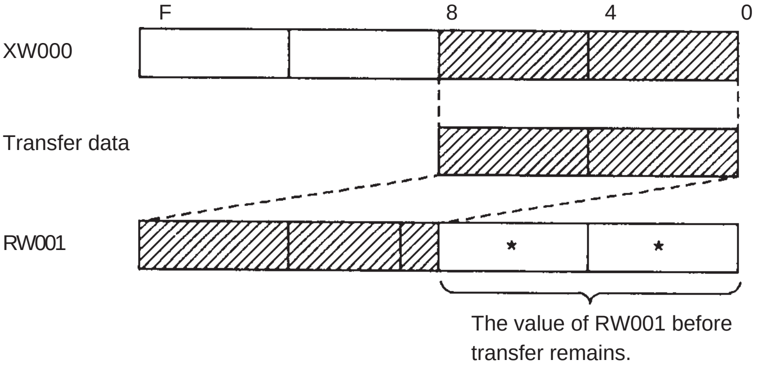
The tranfer data of 1 digit (4 bits) data starting with X0054 as the lower 4 bits, and apper 12 bits which are 0. Then, the transfer data is stored in D1000.



Example 2)

```
                 Q2
─[ XW000 MOV R10018 ]─   (data transfer instruction)
```

The data of the lower 2 digits (8 bits) of XW000 is transferred to the 2 digits (8 bits) which start from R0018.

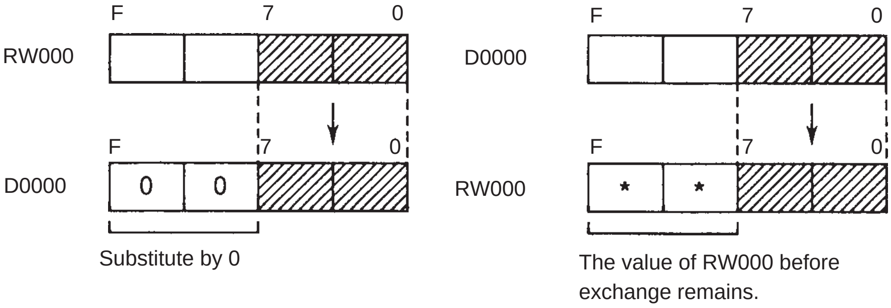


(2) Digit Designation for instruction Data Exchange

Example 3)

```
    Q2
─[ R0000   XCHG   D000 ]─   (data exchange instruction)
```

It is like a mixture of examples 1) and 2). 2 digits (8bits) datastarting with R000 are transferred to the lower 8 bits of D0000 and 8 bits data which are 0 are transferred to the upper 8 bits of D0000. At the same time, the lower 2 digits (8 bits) data of D000 are tranferred to the 2 digits (8 bits) stating with R0000.



NOTE ▼△▼

1. Be careful that the result of digit designation does not exceed the address range. When the result of digit designation exceeds the address range, the excess portion will be ignored a boundary error will occur.
2. A combination of digit designation and index modification can also be used.
   Example)

$$\overset{Q1\quad I}{R0000} \xrightarrow{\text{If I = H001C, it signifies the same}} \overset{Q1}{R001C}$$

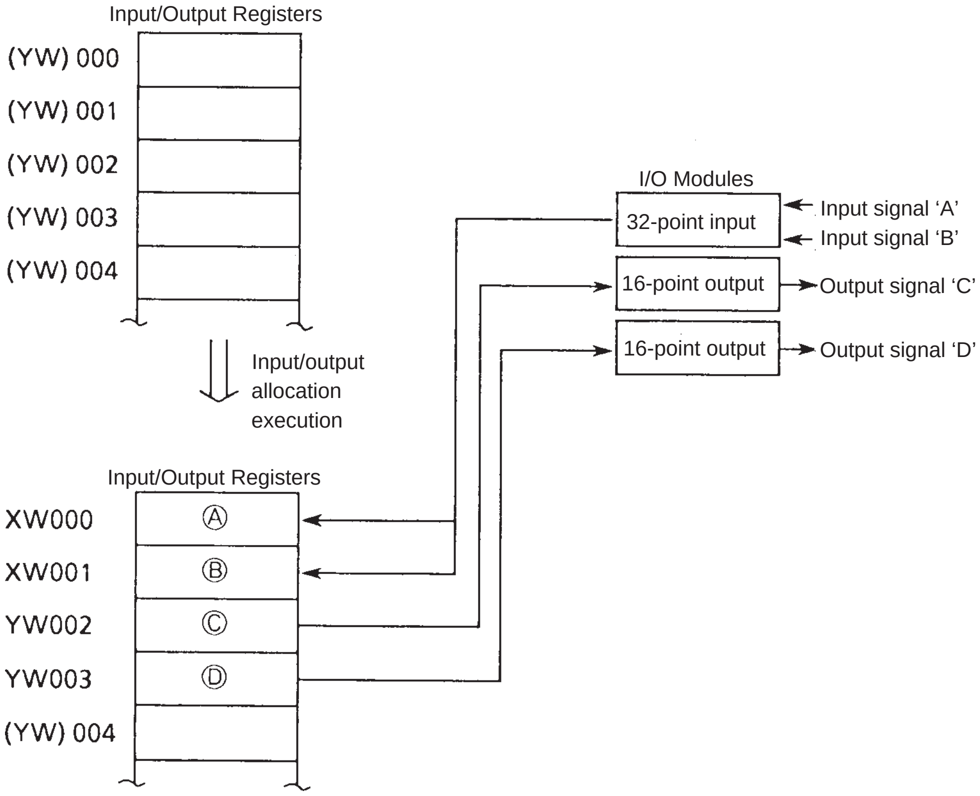
4.1 Overview

The state of external input signals inputted to T2N input modules is read via the input registers/devices (XW/X or IW/I) when scan control is executed. On the other hand, the output data determined in user program execution are outputted to output modules via output registers/ devices (YW/Y or OW/O) and outputs from the output modules to external loads are based on these data.

Input/output allocation is the execution of mapping between input registers/devices and input modules and of mapping between output registers/devices and output modules. In other words, physical devices called I/O modules are allocated to logic devices called registers/devices.

Input registers/devices and output registers/devices do not use their own independent memory areas. They use a series of memory areas which can be said to be input/output registers/devices (a register address range of 128words from 000 to 127).

By executing input/output allocation, function type determination is carried out by making addresses allocated to input modules input registers/devices and addresses allocated to output modules output registers/devices.



Note) Addresses not allocated to I/O modules are output (YW) internally.

The input/output registers are composed of 16bits. There are 128 input/output registers in the T2N. (Therefore 16 input/output signals are stored in one register)

The input/output register is expressed as follows in user program.

Input Register : XW□□□
Output Register : YW□□□

The above □□□ is address of the register (or it is called number of the register), decimal number from 000 to 127.

Also, each bit of input/output registers (it is called “Device”) is expressed as follows.

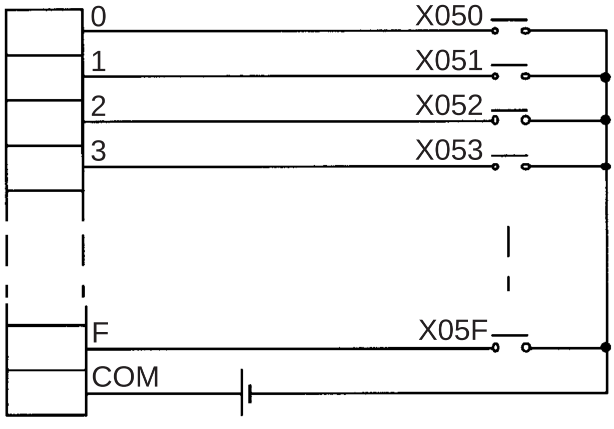
Bit in the input register (input device) : X□□□☒

Bit in the output register (output device) : Y□□□☒

The above □□□ is address of register and the above☒ is bit position of the register.
As for bit position, there are 16 positions (0,1,••••,9,A,B,C,D,E,F).

For example, the input devices (X050-X05F) are assigned corresponding to the input signals as shown below when the 16-point input module is allocated to input register XW05.

16-point input module

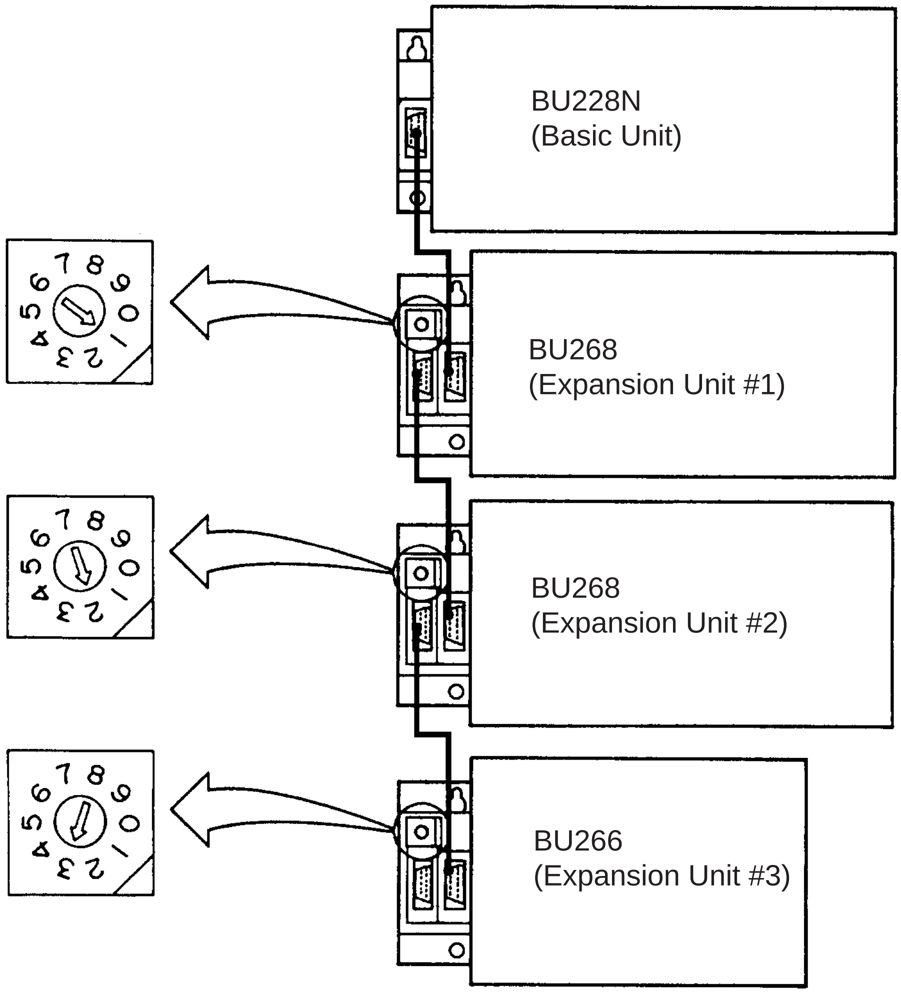


(XW05)

4.2 Setting of Base Unit No.

As explained in Part 1 Section 1.5, when the expansion units, set the Unit No. before operating. The setting is carried out by a rotary switch in the upper part of the expansion connector on the left hand side of the rack.

(Example)



The rack used for	Switch Setting
Expansion Units	Set in the order 1,2,3, starting from the unit closest to the basic unit

NOTE
▼△▼

1. Switches will be set at o at the factory.
2. Be careful not to duplicate Unit Nos. on units.
3. Do not use setting 4-9, as these are not for use.

4.3 Methods of Input/Output Allocation

The execution of input/output allocation can be said in other words to be the carrying out of the registration of input/output allocation information in system information. The T2N CPU checks whether the I/O modules are correctly mounted based on this input/output allocation information when RUN starts-up. Also, at the same time, the correspondence between the input/output registers (XW/YW) and the I/O modules is determined based on this input/output allocation information. On the other hand, the programmer reads this input/output allocation information when communicating with the T2N and recognizes the assignment whether input (XW) or output (YW) for every input/output register address.

There are 2 methods for the registration of input/output allocation information in system information. These are automatic I/O allocation and manual I/O allocation.

The registration of input/output allocation information is only available when the T2N is in the HALT mode but not in the ‘memory protect’ state (with the exception of the operation mode switch being to P-RUN).

Automatic I/O Allocation

This is a method of causing the T2N to execute the registration of input/output allocation information. It is carried out by selecting and executing the Autoset command on the I/O allocation screen of the programmer, (T-PDS).

When the automatic I/O allocation is executed, the T2N CPU reads out status of the I/O modules which are mounted (what type of module is mounted in which position) and registers the input/output allocation information.

Each I/O module has one of the module types shown on the following page.

NOTE
▼△▼

The PU235N has built-in Ethernet. The PU245N has both built-in Ethernet and TOSLINE-S20LP.
When the automatic I/O allocation is executed, the T2N CPU detects whether built-in network is or not automatically.
"ETHER" is displayed in the case of PU235N and "E/S" is displayed in the case of PU245N on the CPU of the input/output allocation information screen of the programmer.
Allocation of built-in network is not needed when manual I/O allocation.

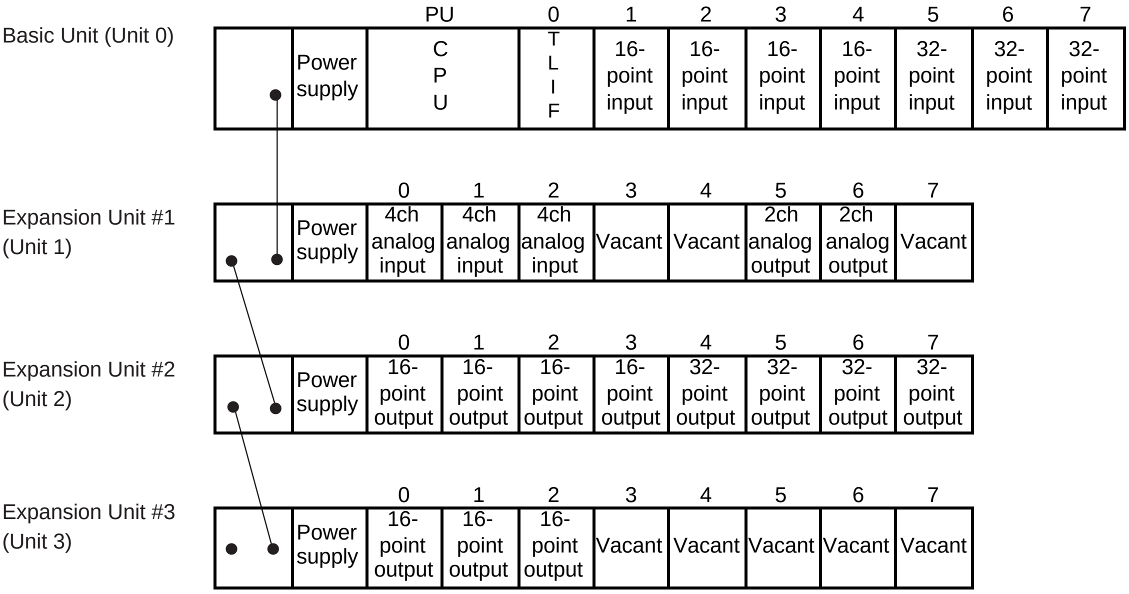
Type	Specification		Module Type
DI31	16-point 12-24V DC/AC input		X 1W
DI32	32-point 24VDC input		X 2W
DI235	64-point 24VDC input		X 4W
IN51	16-point 100-120VAV input		X 1W
IN61	16-point 200-240VAC input		X 1W
RO61	12-point relay output (240VAD/24VDC)		Y 1W
RO62	8-point islated relay output (240VAC/24VDC)		Y 1W
DO31	16-point transistor output (5-24VDC sink		Y 1W
DO32	32-point transistor output (5-24VDC) sink		Y 2W
DO235	64-point transistor output (5-24VDC) sink		Y 4W
DO233P	16-point transistor output (24VDC) source		Y 1W
AC61	12-point triac output (100-240VAC)		Y 1W
AI21	4ch analog input (4-20mA/1-5V)		X 4W
AI31	4ch analog input (0-10V)		X 4W
AI22	4ch analog input (4-20mA/1-5V)		X 4W
AI32	4ch analog input (? 10V)		X 4W
AO31	2ch analog output (5/10V, 20mA)		Y 2W
AO22	2ch analog output (4-20mA/1-5V)		Y 2W
AO32	2ch analog output (?10V)		Y 2W
PI21	1ch pulse input (5/12V)		X 2W
MC11	Single axis position control		X + Y 4W
CF211	Serial Interface		X + Y 4W *2
SN221	TOSLINE-S20 (co-axial cable)		TL-S
SN222	TOSLINE-S20 (optic fibre)		TL-S
MS211	TOSLINE-F10 master station (twisted pair)		TL-F
RS211	TOSLINE-F10 slare station (twisted pair)		TL-F
LK11	TOSLINE-30 (twisted pair)	Transmission capacity 8W setting	Z 8W
		Transmission capacity 16W setting	Z 16W
		Transmission capacity 32W setting	Z 32W
LK12	TOSLINE-30 (optic fibre)	Transmission capacity 8W setting	Z 8W
		Transmission capacity 16W setting	Z 16W
		Transmission capacity 32W setting	Z 32W
DN211	Device Net Scanner module		OPT
PU235N	built-in Ethernet		ETHER *1
PU245N	built-in Ethenet and TOSLINE-S20LP		E/S *1

*1) In the case of PU235N/PU245N, built-in network is automatically allocated.

*2) When executing automatic allocation in the state with a CF211 mounted in the unit, it is registered as X+Y 4W.
However it is necessary to modify this to iX+Y 4W in manual I/O allocation.

For instance, when automatic I/O allocation is executed with the I/O module mounting configuration shown below, the CPU reads the I/O module types which are mounted and creates input/output allocation information and it registers it in system information.

* Module Mounting Configuration



Basic Unit (Unit 0)

		PU	0	1	2	3	4	5	6	7
	Power supply	CPU	TL-IF	16-point input	16-point input	16-point input	16-point input	32-point input	32-point input	32-point input

Expansion Unit #1 (Unit 1)

		0	1	2	3	4	5	6	7
	Power supply	4ch analog input	4ch analog input	4ch analog input	Vacant	Vacant	2ch analog output	2ch analog output	Vacant

Expansion Unit #2 (Unit 2)

		0	1	2	3	4	5	6	7
	Power supply	16-point output	16-point output	16-point output	16-point output	32-point output	32-point output	32-point output	32-point output

Expansion Unit #3 (Unit 3)

		0	1	2	3	4	5	6	7
	Power supply	16-point output	16-point output	16-point output	Vacant	Vacant	Vacant	Vacant	Vacant

* Input/Output Allocation Information (PU245)

Unit 0		Unit 1		Unit 2		Unit 3	
Slot	Module Type	Slot	Module Type	Slot	Module Type	Slot	Module Type
PU	E/S	0	X 4W	0	Y 1W	0	Y 1W
0	TL-F	1	X 4W	1	Y 1W	1	Y 1W
1	X 1W	2	X 4W	2	Y 1W	2	Y1W
2	X 1W	3		3	Y 1W	3	
3	X 1W	4		4	Y 2W	4	
4	X 1W	5	Y 2W	5	Y 2W	6	
5	X 2W	6	Y 2W	6	Y 2W	6	
6	X 2W	7		7	Y 2W	7	
7	X 2W						

Manual I/O Allocation

This is the method by which the user edits the input/output allocation information on the I/O allocation information screen of the programmer (T-PDS) and writes it to the T2N. The manual I/O allocation is used in the following cases.

*When carrying out programming in a status in which the I/O modules are not fully mounted
*When it is desired to remove a special module from the the subjects of batch input/output
*When using the starting address setting function
*When allocating a specified number of registers to slot left vacant for future addition
*When carrying out off-line programming
*When using a serial interface module which requires iX +Y 4W

For manual I/O allocation, module types are set for each slot. The module types which can be set at this time are as shown below. Module types are expressed by combinations of function classifications and numbers of registers occupied. (except for MMR, TL-S, TL-F and OPT)

Function Classification	Number of Registers Occupied	Remarks
X	01, 02, 04,08	Input (batch input/output subject)
Y	01, 02, 04, 08	Output (batch input/output subject)
X + Y	02, 04, 08	Input + output (batch input/output subject)
iX	01, 02, 04, 08	Input (other than batch input/output subject)
iY	01, 02, 04, 08	Output (other than batch input/output subject)
iX + Y	02, 04, 08	Input + output (other than batch input/output subject)
Z	08, 16, 32	When TOSLINE-30 used
SP	01, 02, 04, 08	Space
MMR	–	Memory type (not used)
TL-S	–	For TOSLINE-S20
TL-F	–	For TOSLINE-F10
OPT	–	Option type

(1) Allocations to input/output modules are:- X and iX to input modules, Y and iY to output modules and X+Y and iX+Y to input/output mixed modules. The input/output registers which correspond to modules with the designation i attached are not included in batch input/output subjects.
(2) SP is used when allocating an arbitrary number of registers to a vacant slot.
(3) MMR is not used in the T2N.
(4) The TOSLINE-S20 modules are allocated as TL-S, but the T2N CPU does not exchange data automatically.
Use READ/WRITE instruction to access data.
(5) TL-F is allocated to data transfer device TOSLINE-F10.
(6) Z is allocated to data transfer device TOSLINE-30.

NOTE
▼△▼

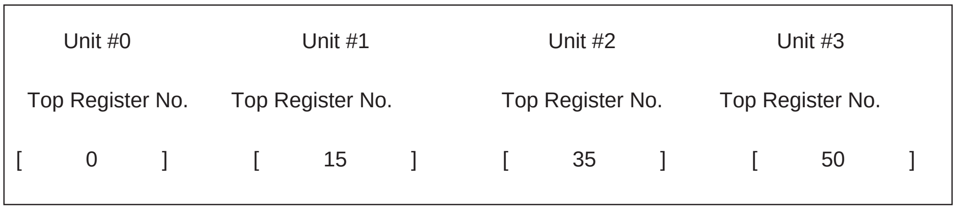
Input/output allocation information can be freely edited and registered by carrying out manual I/O allocation. However, it is necessary that the registered input/output allocation information and the I/O module mounting state should agree for starting-up RUN.
When executing the ‘forced run’command, operation (RUN-F mode) is possible even if the modules registered in the allocation information are not in the mounted state. However, in this case also, operation cannot be executed when a module of a different type to the registered module is mounted (I/O mismatch).

Unit Leading Address Setting

In manual I/O allocation, the starting register address (input/output registers) of each unit can be set and registered.

The register addresses can be arranged for every unit by using this function. Also, when an I/O module is added in a vacant slot in the future, it is possible to avoid affecting the register addresses of other units.

(Unit Starting Address Setting/Display Screenon T-PDS)



Unit #0	Unit #1	Unit #2	Unit #3
Top Register No.	Top Register No.	Top Register No.	Top Register No.
[0]	[15]	[35]	[50]

In the case of this screen example, address allocations can be carried out
from XW/YW000 for the basic unit
from XW/YW015 for expansion unit #1
from XW/YW035 for expansion unit #2
from XW/YW050 for expansion unit #3.

NOTE
▼△▼

Settings by which latter stage units become lower register addresses cannot be made.

4.4 Register and Module Correspondence

When input/output allocation information is registered by carrying out automatic allocation or individual allocation, equivalence between registersnand modules is automatically determined by the following rules.

(1) In any unit, allocation is the low address registers are allocated in sequence from the module at the left end.

(2) In a case when the unit leading address is not set (it is never set by automatic allocation), the registers are allocated in continuation from the previous stage unit.

(3) A slot for which a module type is not set (any vacant slot in automatic allocation is the same) does not occupy any registers

(4) The cases of basic/expansion type rack except 8 slots I/O also are handled in the same way as standard size rack (8 slots) for input/ output allocation,and they are regarded as having slots without settings in the latter portions of the unit. Therefore these portions do not occupy registers.

(5) Slots for which SP (space) is set, output registers are allocated internally by a number of set words.

(6) Modules for which Z, OPT, TL-S and TL-F are set do not occupy input/output registers (XW/YW).

(7) t/output registers which are not allocated to I/O modules become output registers (YW) in the programming. Thus, they can be used in the same way as auxiliary registers/relays (RW/R).

NOTE
▼△▼

For the allocation of link registers/link relays to data transmisson modules, see the separate manual for these modules.

The following Tables show the allocation of registers when input/output allocation information is registered.

Example)

* Input/Output Allocation Information (PU245N)

Unit 0		Unit 1		Unit 2		Unit 3	
Slot	Module Type	Slot	Module Type	Slot	Module Type	Slot	Module Type
PU	E/S	0	X 4W	0	Y 1W	0	Y 1W
0	TL-F	1	X 4W	1	Y 1W	1	Y 1W
1	X 1W	2	X 4W	2	Y 1W	2	Y1W
2	X 1W	3		3	Y 1W	3	
3	X 1W	4		4	Y 2W	4	
4	X 1W	5	Y 2W	5	Y 2W	6	
5	X 2W	6	Y 2W	6	Y 2W	6	
6	X 2W	7		7	Y 2W	7	
7	X 2W						

* Register Allocation

Unit 0		Unit 1		Unit 2		Unit 3	
Slot	Register	Slot	Register	Slot	Register	Slot	Register
PU		0	XW010~XW013	0	YW026	0	YW038
0	(Note)	1	XW014~XW017	1	YW027	1	YW039
1	XW000	2	XW018~XW021	2	YW028	2	YW040
2	XW001	3		3	YW029	3	
3	XW002	4		4	YW030, YW031	4	
4	XW003	5	YW022, YW023	5	YW032, YW033	5	
5	XW004, XW005	6	YW024,YW025	6	YW034, YW035	6	
6	XW006, XW007	7		7	YW036, YW037	7	
7	XW008, XW009						

(Note) LW000 to LW031 are alloctated to the TOSLINE-F10.

5.1 Overview

The T2N supports 2 types of programming language for the user programs - ladder diagram and SFC. Multiple programming languages can be used in mixed by a single user program by separating blocks of the program. Thus, the optimum program configuration for the control functions can be achieved.

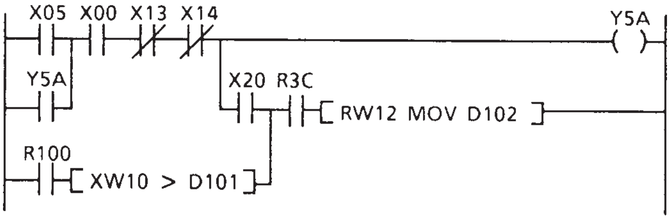
(1) Ladder Diagram

This is the language which is core programming language for the T2N. The program is configured by a combination of relay symbols and function blocks. This language is suitable for logic control and time control.

Relay Symbols ....These are no contact, NC contact, coil, etc.

Function Blocks ...These are box type instructions which express single functions. They can be freely positioned in a ladder diagram network by treating them in a similar way to relay contacts. The output of one function block can be connected to the input of another function block.

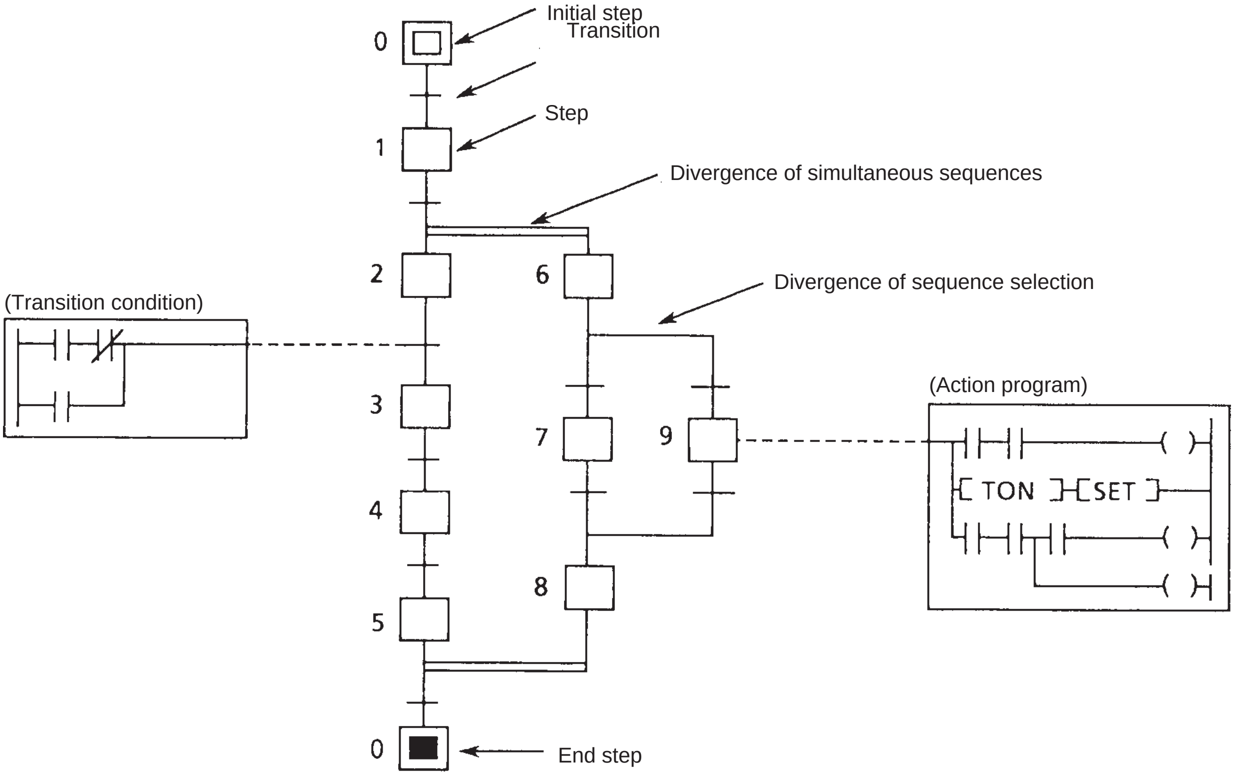
Example)



(2) SFC (Sequential Function Chart)

This is a programming language suitable for process stepping control (sequential control). Also, it is a language which makes the flow of control easy to see. Therefore, it is effective for program maintenance and standardization. SFC program is composed of structure part which shows the flow of control, action program which shows the operation of each step and transition condition parts which enable the process to advance. Action part and transition condition parts are produced by ladder diagrams. SFC can be considered as an execution control element for making a program easier to see by arranging the control processes and conditions rather than a single programming language.

(SFC Structure)



The flow of control advances downward from the initial step and, when it reaches the end step, it returns to the initial step. A step corresponds to an operational process, and there is an actionprogram corresponding to each step. The condition of shifting from one step to the next is called 'transition', and there is a transition condition corresponding to each transition. When the immediately preceding step of a transition is in the active state and the transition condition is ON, the state of the immediately preceding step is changed to inactive and the next step becomes active.

The following Table shows the programming languages which are usable for each program type/part.

Program Type/Block	Ladder Diagrams	SFC
Main program	❍	❍
Sub-program	❍	❍
Interrupt program	❍	X
Sub-routine	❍	X*
SFC active program part	❍	X*
SFC transition condition part	❍	X

❍ : Usable
X : Not usable

*) SFC can be made an hierarchical structure (other SFC can be made to correspond to 1 step of SFC). In this case a macro-step (equivalent to an SFC sub-routine) is used.

5.2 Ladder Diagram

Mixed use can be made of the two types of programming language, ladder diagram and SFC in the T2N. However, of these, ladder diagram is the basic language which must be present in the user program.

Here, the structure, execution sequence and general items of ladder diagram instructions are explained for ladder diagram programs.

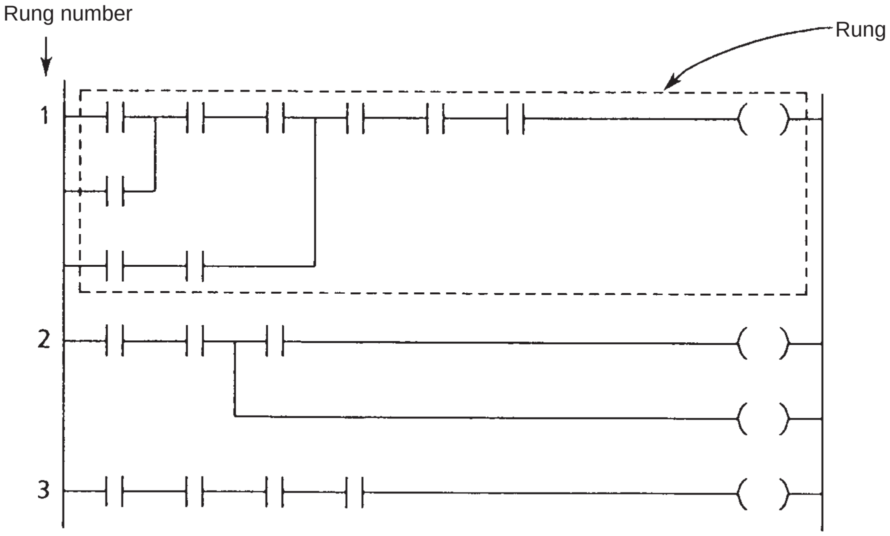
As explained before, a user program is registered by every functional type which is called a program type. Furthermore, in each program type the user program is registered by one or a multiple of units called 'blocks'.

Program Types ...	{ Main program, sub-program #1, timer interrupt program, I/O interrupt programs #1-#4, sub-routine.
Blocks	Blocks 1-256 (1 language/1 block).

When commencing programming in a block to be newly registered, that program is designated by the language which is used (this is called 'language designation').

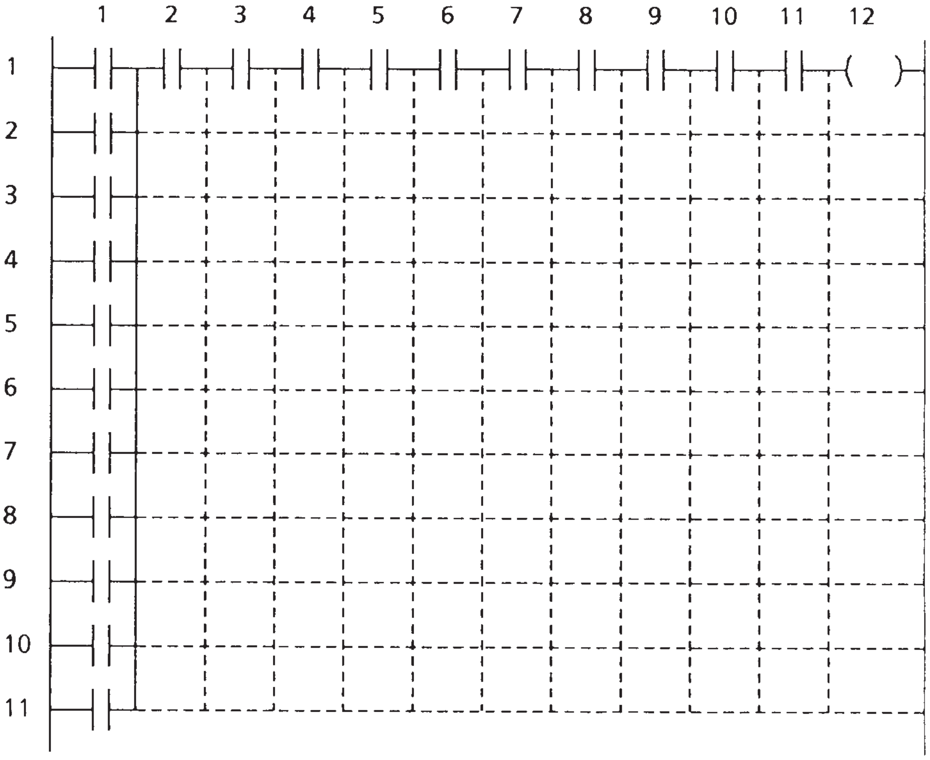
However, in the case of ladder diagram, the operation of language designation is not required (the default is ladder diagram).

The ladder diagram program in any one block is registered/arranged by units called 'rung'. A rung is defined as 1 network which is a combination of lines connected to each other, as shown below.



The rung numbers are a series of numbers (decimal numbers) starting from 1, and rung numbers cannot be skipped. There is no limit to the number of rungs.

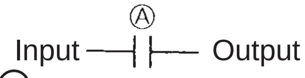
The size of any one rung is limited to 11 lines x 12 columns, as shown below.



Ladder diagram is a language which composes programs using relay symbols as a base in an image similar to a hard-wired relay sequence. In the T2N, in order to achieve an efficient data-processing program, ladder diagram which are combinations of relay symbols and function blocks are used.

Relay Symbols ... These are NO contacts, NC contacts, coils and contacts and coils to which special functions are given. Each of these is called an ‘instruction’. (Basic ladder instructions)

Example) NO contact



When device A is ON, the input side and the output side become conductive.

Viewed from the aspect of program execution, the operation is such that when the input is ON and the content of device A is also ON, the output will become ON.

Function Blocks .... These are expressed as boxes which each show 1 function. As types of function, there are data transfers, the four arithmetic operations, logic operations, comparative decisions, and various mathematical functions. Each of these is called an ‘instruction’. (Function instructions)

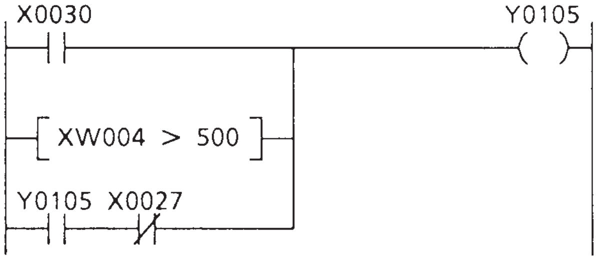
In a function block there are 1 or more inputs and 1 output. When a certain condition is satisfied by the input state, a specified function is executed and the ON/OFF of the output is determined by the result of execution.

Example 1) Addition Instruction

Input ─[Ⓐ + Ⓑ → Ⓒ]─ Output

When the input is ON the content of register Ⓐ and the content of registerⒷ are added and the result is stored in register Ⓒ. The output becomes ON if an overflow or an underflow is generated as the result of the addition.

Example 2) Combination of Relay Symbols and Function Blocks



When X0030 is ON or the content of XW004 exceeds 500, Y0105 becomes ON. When Y0105 stays on even if X0030 is OFF and the content of XW004 is 500 or less, Y0105 will become OFF when X0027 becomes ON.

NOTE
▼△▼

1. If a function block is considered as the operation of a ladder diagram, it can be regarded as a contact which has a special function. By carefully arranging the function blocks in the order of execution of instructions, complex control functions can be achieved by an easily understandable program.
2. A list of ladder diagram instructions is shown in Section 5.5. For the detailed specifications of each instruction, see the separate volume, ‘Instruction set Manual’.

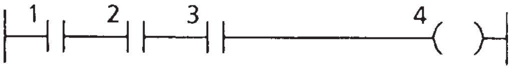
Instruction Execution Sequence

The instructions execution sequence in a block composed by ladder diagram are shown below.

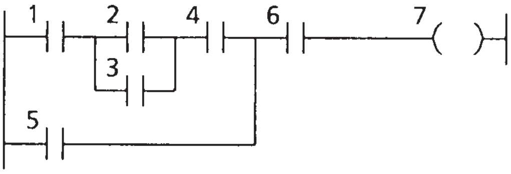
(1) They are executed in the sequence rung1, rung 2, rung 3 ... through to the final rung in the block (in the case of a block with an END instruction, through to the rung with the END instruction).

(2) They are executed according to the following rules in any one rung.

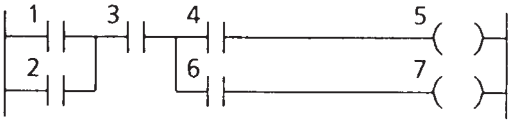
(1) When there is no vertical connection, they are executed from left to right.



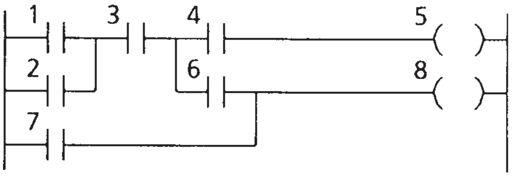
(2) When there is an OR connection, the OR logic portion is executed first.



(3) When there is a branch, they are executed in the order from the upper line to the lower line.



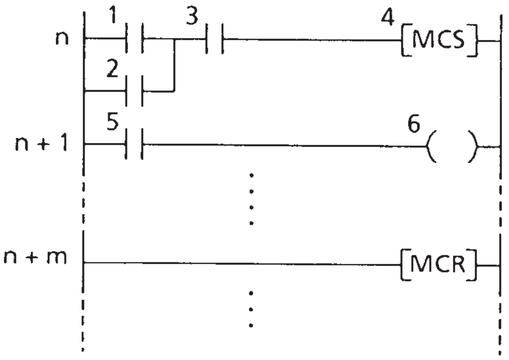
(4) A combination of (2) and (3) above



The instructions execution sequence in which function instructions are included also follows the above rules. However, for program execution control instructions, this will depend on the specification of each instruction.

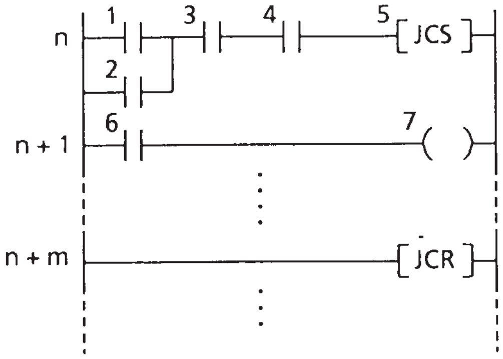
The following show the execution sequences in cases in which program execution control instructions are used.

* Master Control (MCS/MCR, MCSn/MCRn)



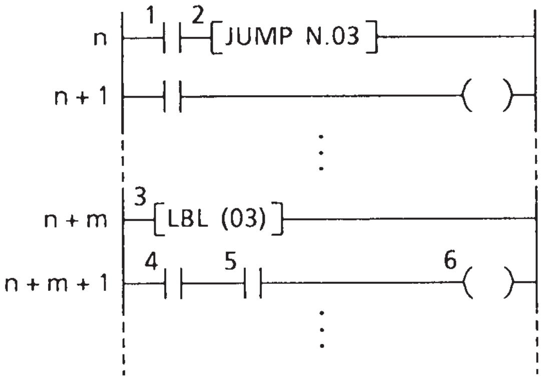
When the MCS input is ON, execution is normal.
When the MCS input is OFF, execution is by making the power rail from the rung following MCS to the rung of MCR OFF (the execution sequence is the same).

*Jump Control (JCS/JCR)



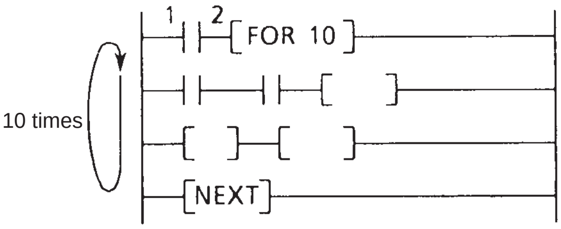
When the JCS input is ON, the instructions from the rung following JCS to the rung of JCR are read and skipped at high speed (instructions are only read and not executed). When the JCS input is OFF, execution is normal.

*Conditional Jump (JUMP/LBL)



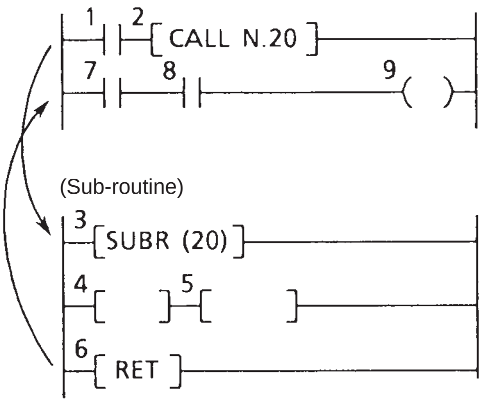
When the JUMP instruction input is ON, execution shifts to the rung following the LBL instruction with the designated label number (03 in the example on the left) (the numbers in the diagram on the left are the execution sequence at this time). When the JUMP instruction input is OFF, execution is normal.

*Repeat (FOR/NEXT)



When the FOR instruction input is ON, the instructions between FOR and NEXT are repeatedly executed the designated number of times (10 times in the example on the left), and when the designated number of times is reached, execution is shifted to the rung following the NEXT instruction. When the FOR instruction input is OFF, execution is normal.

*Sub-Routine (CALL/SUBR/RET)



When the CALL instruction input is ON, execution is shifted to the rung following the SUBR instruction with the designated sub-routine number (20 in the example in the left). When the RET instruction is reached, execution is returned to the instruction following the CALL instruction destination (the numbers in the diagram on the left are the execution sequence at this time). When the CALL instruction input is OFF, execution is normal.

General Information on Ladder Diagram Instructions

The general facts required for designing programs with ladder diagrams are listed below.

(1) In all program types, it is necessary to create at least one block by ladder diagram. In other words, the ends of the main program and each sub-program are judged by ladder diagram END instruction. Also, the end of each interrupt program is judged by a ladder diagram IRET instruction. Furthermore, it is necessary to compose the entry to and exit from a sub-routine by the ladder diagram SUBR instruction and RET instruction.

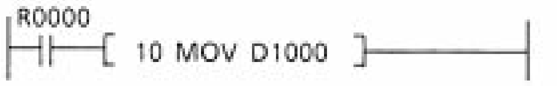
(2) The group of instructions which includes the timer instructions (4 types), counter instruction, jump control instruction, master control instruction and END instruction in the relay symbol type instructions is called the ‘basic ladder instructions’.

(3) Instructions other than the basic laddeer instructions are called ‘function instructions’. The function instructions have respective individual function numbers (FUN No.). Also, even if instructions have the same function number, selection of the execution conditions is possible as shown below. (There are some instructions which cannot be selected)

Normal ... Executed every scan while the instruction input is ON.
Edged ... Executed only in the scan in which the instruction input changes from OFF to ON.

Example) Data Transfer Instruction

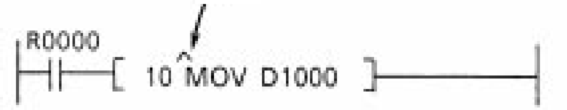
Normal



The MOV instruction (substitute 10 in D1000) is executed every scan while R0000 is ON.

Edged symbol

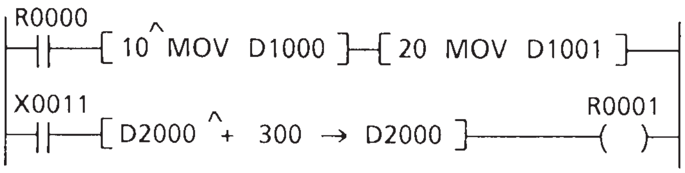
Edged



The MOV instruction (substitute 10 in D1000) is executed only in the scan in which R0000 changes from OFF to ON.

Any instructions cannot be positioned after (to the right of) a edged function instruction.

Example)

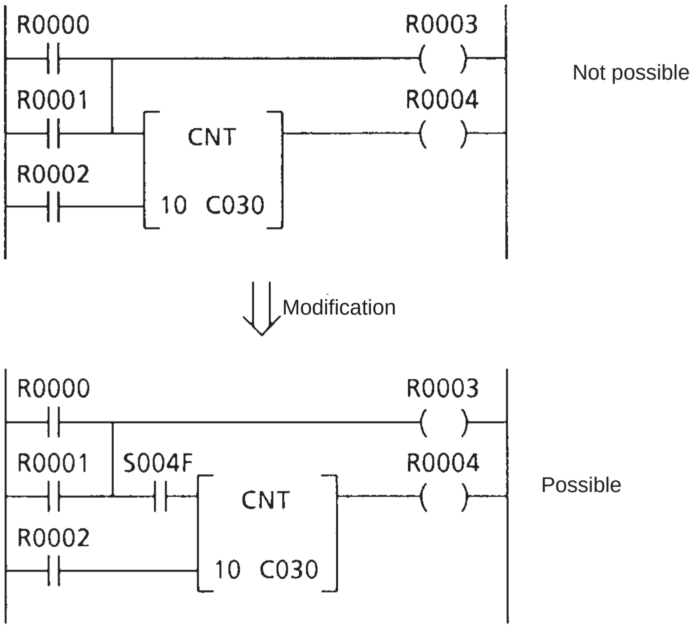


Neither of these two rung can be created.

(4) The number of steps required for one instruction differs depending on the type of instruction. Also, even with the same instruction, the number of steps occupied varies depending on whether digit designation is used in the operand, a constant or a register is used in a double-length operand, etc (1-10 steps/1 instruction).
Also, basically step numbers are not required for vertical connection lines and horizontal connection lines.

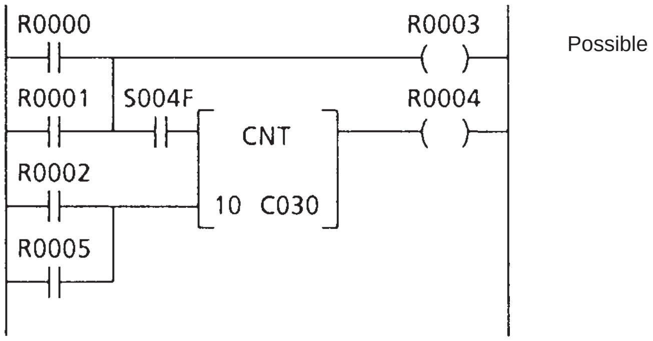
(5) In a instruction which has multiple inputs, a vertical connection line cannot be placed immediately before an input. In this case, insert a dummy contact (such as the NO contact of special relay S004F which is always ON) immediately before the input.

Example)



The above arrangement is not required for the lowest input of multiple inputs.

Example)



5.3 SFC

SFC is the abbreviation of Sequential Function Chart. This is a programming language suitable for process stepping control (sequential control). In the T2N, an SFC is applied in which the following functions are added to general SFC.

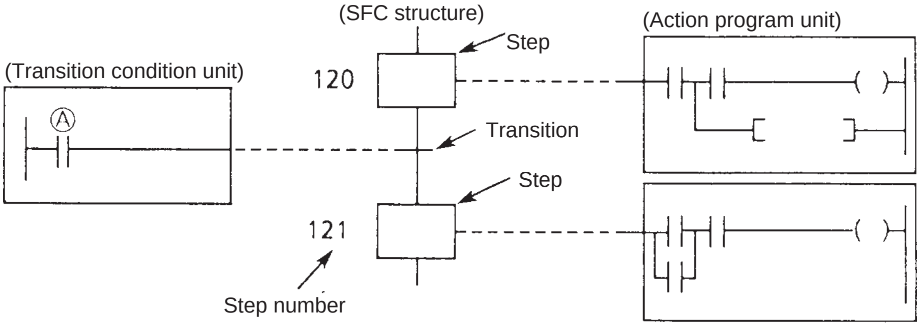
* Jump ... Moves the active state to an arbitrary step when a jump condition is satisfied.

* Step with waiting time... Even if the transition condition is satisfied, step transition is not carried out until a set time has elapsed. (wait step)

* Step with alam ... When transition to the following step is not carried out even if the set time has elapsed, the designated alarm device becomes ON.(Alam step)

SFC can be used in the main program. Here the overall composition of SFC, the composition factors of SFC and notes on program creation are described.

An SFC program is composed of SFC structure, action program parts and transition condition units.



An SFC structure regulates the flow of the control operation and has steps and transitions as its basic elements. A step is expressed by one box, as shown above. Each step has its own step number. Also, corresponding execution programs are annexed 1 to 1 to steps.

Steps have the two states of active and inactive. When a step is active, the power rail of the corresponding execution program will be in the live state (power rail ON). When a step is inactive, the power rail of the corresponding action program will be in in the cut off state (power rail OFF).

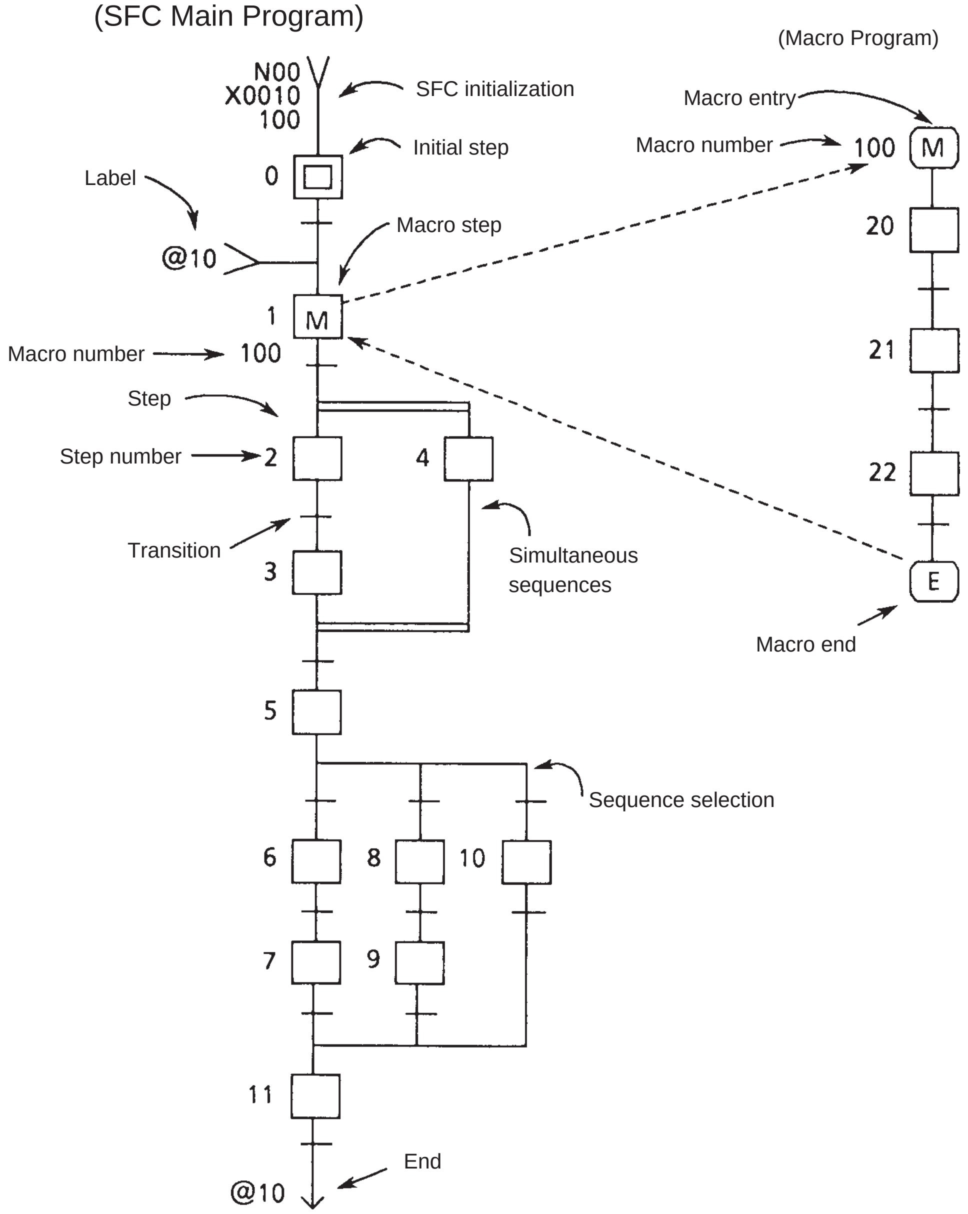
On the other hand, a transition is located between step and step, and expresses the conditions for transition of the active state from the step immediately before (upper step) to the following step (lower step). Corresponding transition conditions are annexed 1 to 1 to transitions.

For instance, in the diagram above, when step 120 is active, the execution program power rail corresponding to step 120 becomes ON. In this state, when device Ⓐ becomes ON, the transition conditions are satisfied, and step 120 becomes inactive and step 121 becomes active. In accompaniment to this, theaction program power rail corresponding to step 120 becomes OFF (executed as power rail OFF), and the action program power rail corresponding to step 121 becomes ON.

Overall Configuration

The following illustrates the overall configuration of an SFC program.

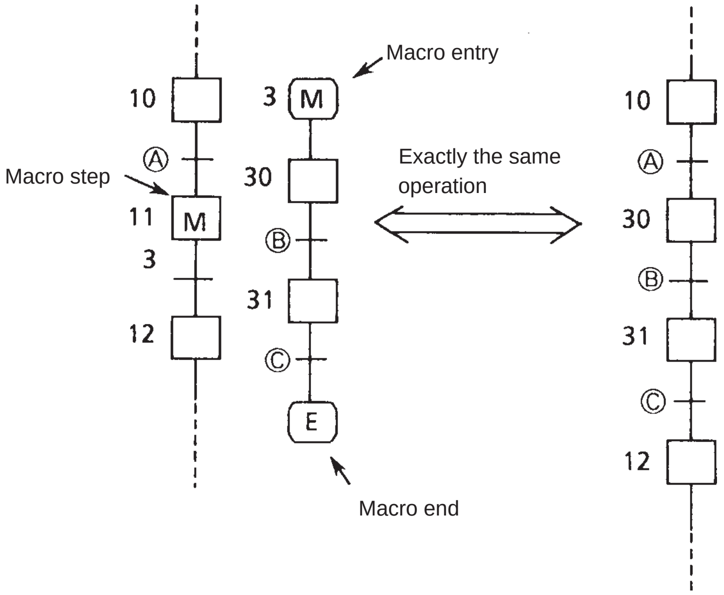
(SFC Main Program)



The overall SFC program cam be considered as divided into an SFC main program and a macro program.

The SFC main program has an initial step in its structure. In the T2N, a maximum of 64 SFC main programs can be cerated.

On the other hand a macro program is a sub-squence which starts from ‘macro entry’ and finishes at ‘macro end’. Each macro program has its own macro number, and corresponds 1 to 1 to macro steps which are present in the SFC main program or other macro programs. Macro programs are used for rendering the program easy to see by making the SFC program an hierarchical structure. In all, 128 macro programs can be created.



NOTE
▼△▼

(1) Macro steps can be used in macro programs (SFC multi-level hierarchy). There is no limit to the number of levels.

(2) Macro programs and macro steps must correspond 1 to 1. That is to say, macro steps designated with the same macro number cannot be used in multiple locations.

SFC programming becomes possible by designating blocks and then selecting SFC by language designation.

Only one SFC main program or one macro program can be created in 1 block. (1 SFC/block)

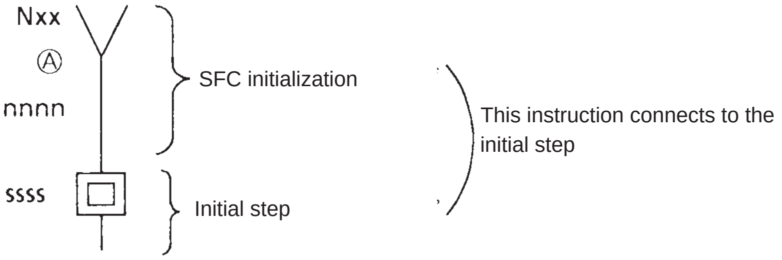
Also, the maximum number of SFC steps per block is 128.

SFC Composition Elements

The following is a description of the elements which compose an SFC program.

(1) SFC Initialization
This is the function which starts-up (makes active) the designated initial step by making the steps in a designated area inactive. Either of the two methods of an SFC instruction or a ladder diagram instruction is used. One SFC initialization is required for 1 SFC main program.

① SFC Instruction



Operands: xx = Program number (0-63)
Ⓐ = Start-up device (except T. and C.)
nnnn = Number of initialized steps (1-2048)

Function: When the device (with the exception of a timer device or acounter device) designated by A changes from OFF to ON, the number of steps following the initial step (ssss)which are designatedby nnnn (from step number ssss to ssss + nnnn - 1), are made inactive, and the initial step (ssss) is made active.

② Ladder Diagram Instruction

Input ─[SFIZ (nnnn) ssss]─ Output

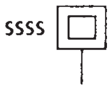
Operands: nnnn = Number of initialized steps (1-2048)
ssss = Step number of initial step (0-2047)

Function: When the input changes from OFF to ON, the initial step the number of steps designated by nnnn from the step number designated by ssss (from step number ssss to ssss + nnnn - 1), and made inactive, and the initial designated by ssss is made active.

(2) Initial Step

This is the step which indicates the start of an SFC main program. It has its own step number and can have an action program which corresponds 1 to 1.

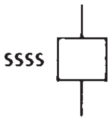
Only 1 initial step can be programmed in 1 block.



ssss = Step number (0-2047)

(3) Step

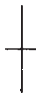
This expresses one unit of contral steps. The step has its own step numbers and has program which corresponds 1 to 1.



ssss = Step number (0-2047)

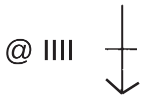
(4) Transition

This expresses the conditions for shifting the active state from a step to the following step. Transition has transition condition units which corresponds 1 to 1.



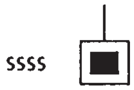
(5) SFC End

This expresses the end of an SFC main program. An SFC main program requires either this 'SFC end' or the 'end step' of (6). The 'end' has a transition condition which corresponds 1 to 1 and a return destination label number. When transition condition is satisfied with the step immediately before being in the active state, the step following the designation label is made active with making the step immediately before inactive. (This is the same operation as that described in 'jump' below).



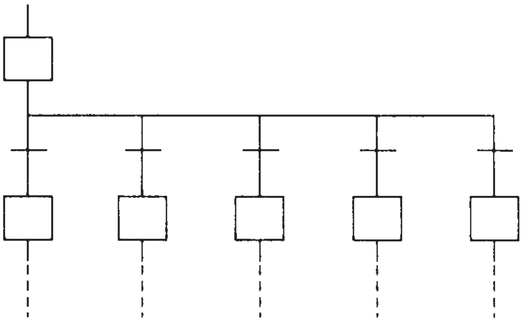
IIII = Label number (0-1023)

(6) End Step
This expresses the end of an SFC main program. An SFC main program requires either this ‘end step’ or the ‘SFC end’ of (5). The end step has the same step number as the initial step. When the immediately preceding transition condition is satisfied, the initial step returns to the active state.

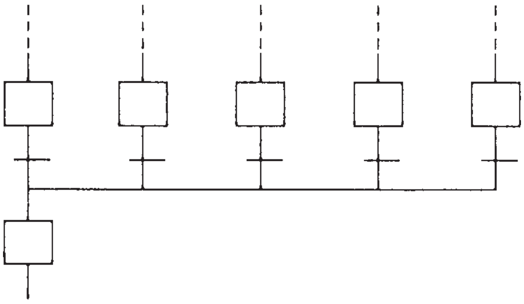


ssss = Initial step number (0-2047)

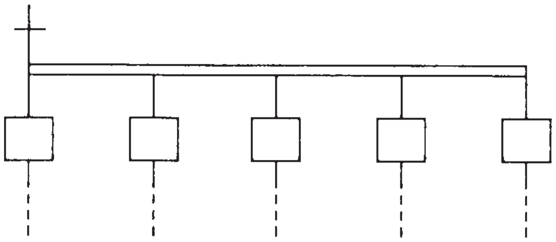
(7) Sequence Selection (divergence)
This transfers the active state to 1 step in which the transition condition is satisfied out of multiple connected steps. When the transition conditions are satisfied simultaneously, the step on the left has priority. (The number of branches is a maximum of 5 columns).



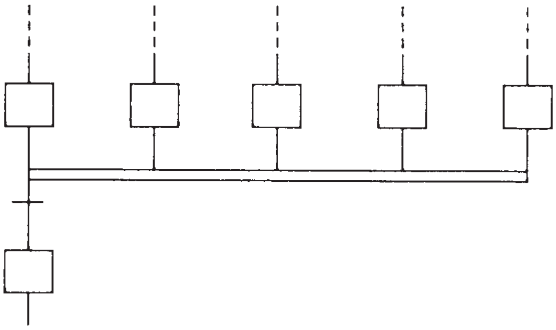
(8) Sequence Selection (convergence)
This collects into 1 step the paths diverged by above (7).



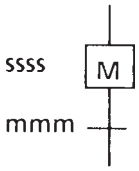
(9) Simultaneous Sequences (divergence)
After the immediately preceding transition condition is satisfied, this makes all the connected steps active. (The number of branches is a maximum of 5 columns).



(10) Simultaneous Sequences (convergence)
When all the immediately preceding steps are active and the transition condition is satisfied, this shifts the active state to the next step.



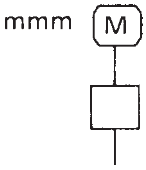
(11) Macro Step
A macro step corresponds to one macro program. When the immediately preceding transition condition is satisfied, this shifts the active state to macro program with the designated macro number. When the transition advances through the macro program and reaches the macro end, the active state is shifted to the step following the macro step. A macro step is accompanied by a dummy transition which has no transition condition (always true).



ssss = Step number (0-2047)
mmm = Macro number (0-127)

(12) Macro Entry

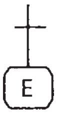
This expresses the start of a macro program. The macro entry has no action program. Steps are connected below the macro entry. Only 1 macro entry can be programmed in 1 block.



mmm = Macro number (0-127)

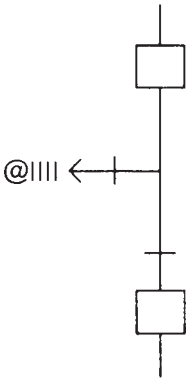
(13) Macro End

This expresses the end of a macro program. Macro end has transition condition which corresponds 1 to 1, and returns to the corresponding macro step when this transition condition is satisfied.



(14) SFC Jump

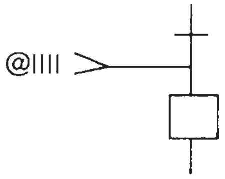
This expresses a jump to any arbitrary step. Jump has transition condition which corresponds 1 to 1, and jump destination label numbers. When the transition condition is satisfied, the active state jumps to the step following the designated label. When the jump transition condition and the transition condition for the following step are simultaneously satisfed, jump has priority.



llll = Label number (0-1023)

‘SFC Jump’ is located immediately after a step. Jumps with the same label number may be present in multiple locations.

(15) SFC Label
This expresses the return destination from an ‘SFC end’ and the jump destination from a ‘SFC jump’. Label is located immediately after transitions.

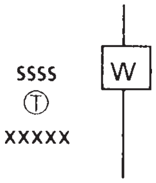


IIII = Label number (0-1023)

NOTE
▼△▼

Note that, when labels corresponding to ‘SFC Jump’ or ‘SFC End’ are not present, or when labels with the same label number are present in multiple locations, an error will occur when RUN starts-up.

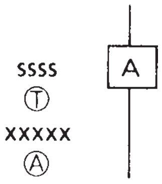
(16) Wait Step
This is a step which measures the time after becoming active and does not execute transition, even if the following transition condition is satisfied, until a set time has elapsed. It has an action program corresponding 1 to 1.



ssss = Step number (0-2047)
Ⓣ = Timer register (T000-T511)
xxxxx = Set time (0-65535)

(Note) T000-T063 are 0.01 second timers
T064-T511 are 0.1 second timers

(17) Alarm Step
This is a step which measures the time after becoming active and, when the transition condition is not satisfied within a set time, switches ON a designated alarm device. It has an action program corresponding 1 to 1. Also, when the transition condition is satisfied and the alarm step becomes inactive, the alarm device also becomes OFF.



ssss = Step number (0-2047)
Ⓣ = Timer register (T000-T511)
xxxxx = Set time (0-65535)
Ⓐ = Alarm device (other than X, T., C.)

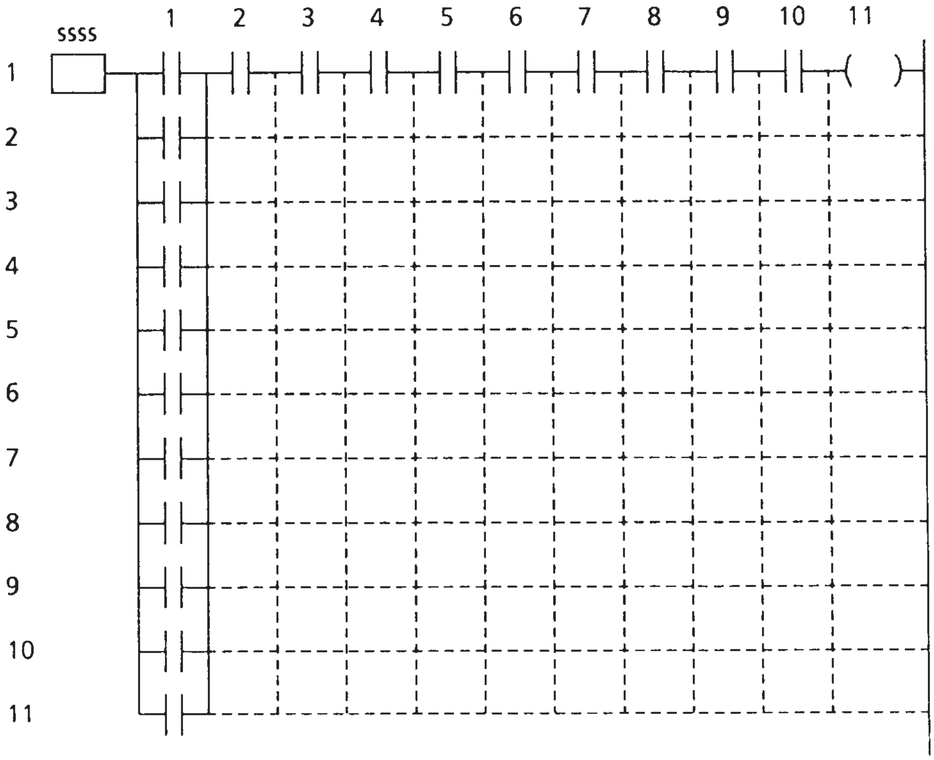
(Note) T000-T063 are 0.01 second timers
T064-T511 are 0.1 second timers

Action Program and Transition Condition

The action program corresponds to 1 step, and the transition condition corresponds to 1 transition.

These are programmed by ladder diagram.

(1) Action Program
The size of 1 action program is 11 lines x 11 columns as shown below, and the number of instruction steps is a maximum of 121 steps.

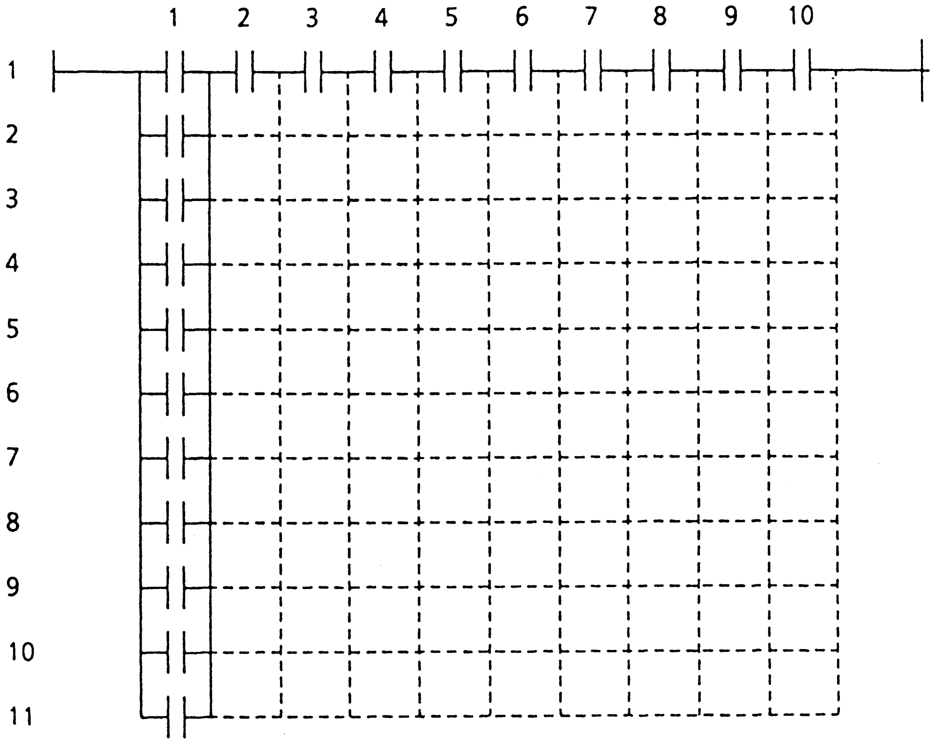


In a case when a larger size than the above is required as an action program, a sub-routine is used. (CALL instruction)
Even if there is no action corresponding to a step, this does not affect SFC operation. In this case, the step becomes a dummy step (a step which waits only the next transition condition will be satisfied).

In programming, by designating the step on the SFC screen and selecting the detail display mode, the monitor/edit screen for the action program corresponding to that step will appear.
In the case when the content of the action program is only 1 instruction out of the SET instruction, the RST instruction, coil, invert coil, positive pulse coil and negative transition-sensing coil, direct editing can be carried out without putting up the detail display screen. See the programmer (T-PDS) operation manual in a separate volume for this operation.

(2) Transition Condition

The size of 1 transition condition is 11 lines x 10 columns, and the number of instruction steps is a maximum of 110 steps.



When there is no transition condition corresponding to a certain transition, that transition condition is always regarded as true. (Dummy transition)

In programming, by designating the transition on the SFC screen and selecting the detail display mode, the monitor/edit screen for the transition condition corresponding to that transition will appear. In the case when the content of the transition condition is only 1 instruction of NO contact or NC contact, direct editing can be carried out without putting up the detail unit display screen. See the programmer (T-PDS) operation manual in a separate volume for this operation.

NOTE
▼△▼

The following execution control instructions cannot be used in action programs and transition conditions.

- * Jump (JSC/JCR, JUMP/LBL)
- * Master control (MCS/MCR, MCSn/MCRn)
- * End (END)
- * FOR - NEXT (FOR/NEXT)

The invert contact and various coil instructions cannot be used in transition conditions

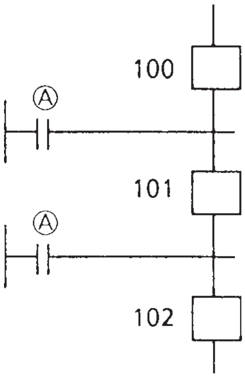
Execution System

The following shows the concept of the execution system in one SFC program.

(1) In one scan, evaluation of the transition condition, the step transition processing and the execution of the action program unit are sequentially operated.

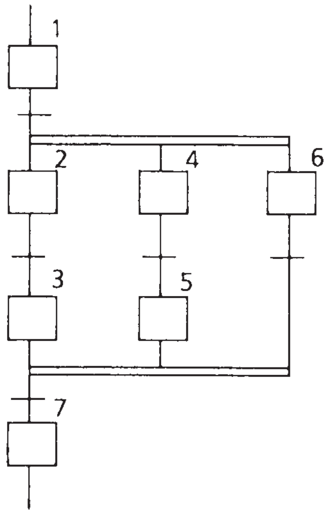
(2) Evaluation of the transition condition means the execution of the transition condition connected to an active step and carrying out a check for transition condition establishment. At this time, since evaluation is made only for active step, there are no multiple step transitions by 1 scan in consecutively connected steps.

For instance, as shown in the diagram on the right, in a program in which the transition condition from step 100 to 101 and the transition condition from step 101 to 102 are the same, step 100 becomes active in the previous scan, and when device Ⓐ has been switched ON in the present scan, there is transition to step 101 in the present scan. (Transition to step 102 will be from the next scan onward)



(3) Step transition processing means making the previous step inactive and the following step active if the transition condition is satisfied, based on the result of evaluation of the transition condition.

(4) Execution of the action program unit corresponding to the active step is carried out by switching the power rail ON, and executing the action program unit corresponding to the inactive step by switching the power rail OFF. At this time, as shown in the following diagram, the execution sequence is from top to bottom, and from left to right in branches.

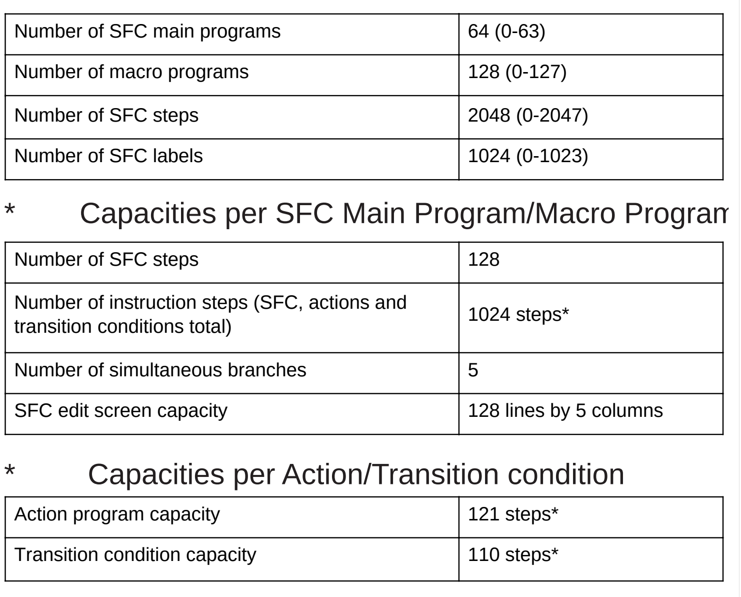


The numerals in the diagram show the execution sequence of the action programs.

Points to Note The following is a list of points to note when creating SFC programs.

(1) The capacity limits of SFC programs are set out in the following Tables. Be careful not to exceed these capacities.

* Overall Capacities (Maximum numbers which can be programmed in the T2N)



Number of SFC main programs	64 (0-63)
Number of macro programs	128 (0-127)
Number of SFC steps	2048 (0-2047)
Number of SFC labels	1024 (0-1023)

* Capacities per SFC Main Program/Macro Program

Number of SFC steps	128
Number of instruction steps (SFC, actions and transition conditions total)	1024 steps*
Number of simultaneous branches	5
SFC edit screen capacity	128 lines by 5 columns

* Capacities per Action/Transition condition

Action program capacity	121 steps*
Transition condition capacity	110 steps*

*) See 5.5 ‘List of Instructions’ for the required numbers of steps for SFC instructions and ladder diagram instructions.

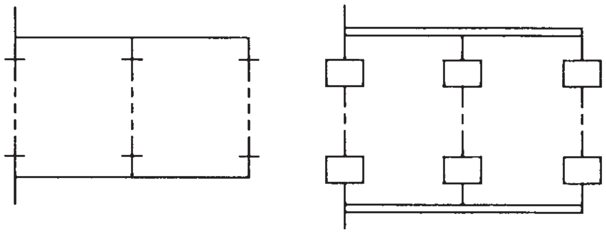
(2) The starting and re-setting of an SFC program is carried out by the SFC initialization instruction (SFC instruction/ladder diagram instruction). SFC initialization makes the steps in a designated area inactive and makes the initial step active. Therefore, the area of the steps designated by SFC initialization (the number of initialized steps) includes all the step numbers which are used in that SFC program (including macro programs as well). Take care that step numbers used in other SFC programs are not involved.

For instance, if the SFC initialization designation is 50 steps from step number 0 and step 50 is used in that SFC program, when SFC initialization is executed with step 50 in the active state, step 50 will remain active.

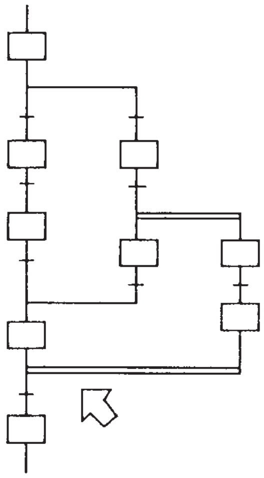
On the other hand, if the SFC initialization designation is 201 steps from step number 100 and step 300 is used in another SFC program, when SFC initialization is executed with step 300 in the active state, step 300 will become inactive without any condition.

(3) There is no limit to the step number sequence used in 1 SFC program (including macro programs). However, the initial step must be made the lowest step number in that sequence.
(See (2) above)

(4) A sequence selection diverges above transitions, and converges below transitions. Also, a simultaneous sequence diverges above a steps and converges below a steps.



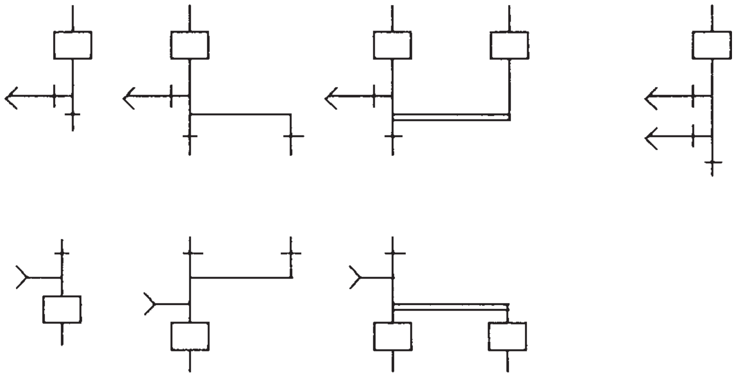
However, the divergence must end in a corresponding convergence. Therefore, programs such as the following are not allowed.



(5) The jump destination of a SFC jump may be either in the upward direction or in the downward direction, or it may be in another SFC program. Also, it is possible to jump to the outside from within a branch.

Since a SFC jump can be very freely used in this way, take thorough precautions so that the SFC logic will not become abnornal (so that multiple unrelated steps in a series of SFC will not become active) through jumping.

A SFC jump is always positioned immediately after a step. Also, although basically a SFC label is positioned immediately after a transition, it is positioned between the convergence line and the step in the case of a sequence selection (convergence).



(6) The states (active/inactive) of SFC steps are not power-cut retained.
When starting-up, all steps become inactive.

(7) The output of an SFC step can be controlled by sandwiching the SFC program block by ladder diagram master control (MCS/MCR).
When the input of MCS is OFF, the power rail of the action program corresponding to the active step also becomes OFF.
However, in the state, step transition is carried out.

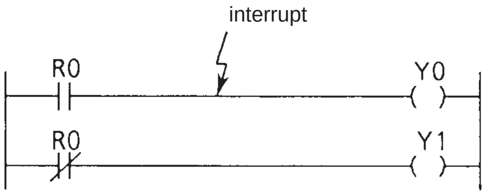
5.4 Programming Precautions

The T2N supports multi-task functions. When interrupt programs are used there is the possibility of the main program being interrupted by an interrupt program. Precautionary notes arising from this are given below, and should be taken into account when creating programs.

(1) Avoid using the same sub-routine in the main program and an interrupt program. When the main program exection is interrupted during a sub-routine is being executed and the same sub-routine is executed in that state, the results after re-starting are sometimes not as expected.

(2) There is no classification of user data (register/device) by program type. Therefore, take thorough precautions over individual data do that there is no erroneous mixed use between program types.

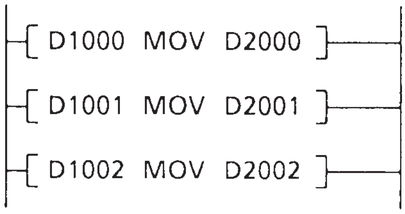
Example)



Interrupt occurs through the timing in the above diagram. And when the content of R0 is modified in the interrupt, the simultaneous ON (or the simultaneous OFF) of Y0 and Y1, which normally could not occur, happens.

(3) Try to execute the exchange of data between main and interrupt programs by 1 instruction, such as the data transfer instruction (MOV) or the table transfer instruction (TMOV) or by using the interrupt disble (DI) and the interrupt enable (EI) instruction Otherwise, the same thing as in (2) above may happen.

Example) Composition of the main program when transferring the three data, D1000, D1001 and D1002, from an interrupt program to the main program.

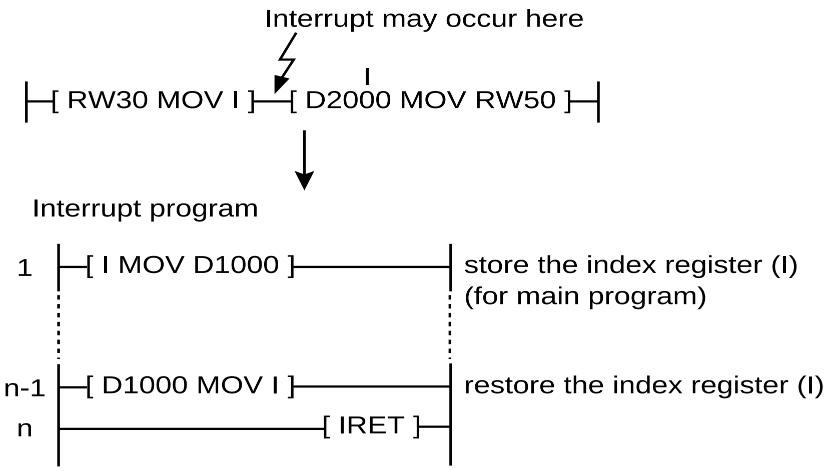


In the above program, when an interrupt occurs between instructions, synchronisation between D2000, D2001 and D2002 cannot be guaranteed. In this case, make 1 instruction by using the table transfer instruction, as follows.

```
|-[  D1000   TMOV   (3)   D2000  ]------|
```

(4) With respect to the index registers (I,J,K), the data of these registers are saved when interrupt occurs and restored when operation returns to main program automatically. However, beacuse of this, even if an index register is used only in an interrupt program, the data continuity of the index register between interrupt intervals is not kept.
In such case, use another register to store index value substitute the value into an index register in the interrupt program.

Example)



5.5 List of instructions

An instruction list is given in the sequence of ladder diagram instructions and SFC instructions on the next page and thereafter.

The groups in the list correspond to the group classifications of function instructions adopted by the programmer (T-PDS). (Except for SFC).

The required numbers of steps signify the size of memory required for storing these instructions. The showing of the required number of steps by a range such as 4-7, is because the number of steps changes due to the following conditions, even for the same instruction.

- When using digit designation, there is an increase of 1 step per 1 operand.
- When a constant is used in a double-length operand, there is an increase of 1 step.
- When executing index modification in a constant, there is an increase of 1 step.

NOTE
▼△▼

Here, an overview of each instruction is given. See the instruction manual in a separate volume for details.

The execution time shows the required time when the T2N fetches the instruction on memory and performs required operation.
The execution time shown on the next page and thereafter is normal case value. i.e. when no index modification, no digit designation and normal registers are used for each operand.
The execution time is subject to increase due to using index modification, digit designation, direct input register/device (IW/I), direct output register/device (OW/O) for each operand.

Ladder Diagram Instructions (Sequence Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Sequence instructions		NO contact	(A) ─┤ ├─	NO contact of device (A) (contact normally open)	1	0.33	
		NC contact	(A) ─┤/├─	NC contact of device (A) (contact normally closed open)	1	0.33	
		Transitional contact (rising)	─┤↑├─	Switches output ON only when input in the previous scan is OFF and the input for this scan is ON.	1	0.33	
		Transitional contact (falling)	(A) ─┤↓├─	Switches output ON only when input in the previous scan is ON and input in this scan is OFF.	1	0.33	
		Coil	(A) ─( )─┤	Switches device (A) on when input is ON.	1	0.44	
		Forced coil	(A) ×( )─┤	Retains state of device (A) when force is applied, regardless of whether input is ON or OFF.	1	0.33	
		Inverter	(A) ─┤I├─	Inverts the input state	1	0.22	
		Invert coil	(A) ─(I)─┤	Stores [data] in device (A), in reverse state to input.	1	0.44	
		Positive Transition-sensing contact	(A) ─┤P├─	Inverts the input state and stores it in device (A).	1	0.54	
		Negative Transition-sensing contact	─┤N├─	Turns output ON for 1 scan only, when input is ON and device (A) has been changed from ON to OFF.	1	0.54	
		Positive Transition-sensing coil	(A) ─(P)─┤	Turns device (A) ON for 1 scan only, when input has been changed from OFF to ON.	1	0.54	
		Negative Transition-sensing coil	(A) ─(N)─┤	Turns device (A) ON for 1 scan only, when input has been changed from ON to OFF.	1	0.54	
		Jump control set	─[JCS]─┤	Carries out high-speed skipping on instructions between JCS and JCR when JCS input is ON.	1	0.22	
		Jump control reset	├─[JCR]─┤		1	0.22	
		End	├─[END]─┤	Indicates end of main program and sub-program.	1	—	

Ladder Diagram Instructions (Sequence Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Sequence instructions		ON delay timer	-[(A) TON (B)]-	Turns output ON when set period specified by (A) has elapsed since input came ON. (B) is timer register.	2	0.44	
		OFF delay timer	-[(A) TOF (B)]-	Turns output OFF when set period specified by (A) has elapsed since input went OFF. (B) is timer register.	2	0.44	
		Single shot timer	-[(A) SS (B)]-	Turns output ON only for the set period, specified by (A), starting when input comes ON. (B) is timer register.	2	0.44	
		Counter	-[C CNT Q]- -[E (A) (B)]	When enable input (E) is ON, counts the number of times the count input (C) has come ON. When count value becomes equal to set value specifiedby (A), turns output (Q) ON. (B) is counter register.	2	0.44	
		Master control set	-[MCS]-|	Turns ON power rail between MCS and MCR when MCS input is ON.	1	0.22	
		Master control reset	|-[MCR]-|		1	0.22	
	134	Master control set (with nesting number	-[MCS n]-|	Turns on power rail to corresponding MCR when MCS input is ON. n is a nesting number. (1 - 7).	2	61	
	135	Master control reset (with nesting number	|-[MCR n]-|		2	58	
	148	Timer trigger	-[TRG (A)]-	When input is changed from OFF to ON, clears timer register specified by (A) and activates timer.	2	153	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Transfer instructions	18	Data transfer	–[(A) MOV (B)]–	Transfers contents of (A) to (B).	3~5	1.20	
	19	Double-length data transfer	–[(A)+1·(A) DMOV (B) + 1·(B)]–	Transfers contents of (A)+1 and (A) to (B)+1 and (B).	3~4	79	
	20	Invert and transfer	–[(A) NOT (B)]–	Transfers the bit-reversed data comprising the contents of (A) to (B).	3	60	
	21	Double-length invert and transfer	–[(A) + 1·(A) DNOT (B) +1·(B)]–	Transfers the bit-reversed data comprising the contents of (A)+1 and (A) to (B)+1 and (B).	3~4	82	
	22	Data exchange	–[(A) XCHG (B)]–	Exchanges the contents of (A) with the contents of (B).	3~5	144	
	23	Double-length data exchange	–[(A) + 1·(A) DXCH (B)+1·(B)]–	Exchanges the contents of (A)+ • (A)1 with the contents of (B)+1 • (B)	3	168	
	24	Table initialization	–[(A) TINZ (n) (B)]–	Initializes the contents of the table of size n, headed by (B), by the contents of (A).	4	134+2n	
	25	Table transfer	–[(A) TMOV (n) (B)]–	Transfers the contents of the table of size n, headed by (A), to the table headed by (B).	4	206+3.5n	
	26	Table invert and transfer	–[(A) TNOT (n) (B)]–	Transfers the bit-reversed data comprising the contents of the table of size n headed by (A) to the table headed by (B).	4	206+8.5n	
	27	Addition	–[(A) + (B) → (C)]–	Adds the contents of (B) to the contents of (A), and stores the result in (C).	4~6	1.63	
	28	Subtraction	–[(A) - (B) → (C)]–	Subtracts the contents of (B) from the contents of (A), and stores the result in (C).	4~6	1.63	
	29	Multiplication	–[(A) ❋ (B) → (C) +1·(C)]–	Multiplies the contents of (A) by the contents of (B) and stores the result in (C)+1 • (C).	4~6	84	
	30	Division	–[(A) / (B) → (C)]–	Divides the contents of (A) by the contents of (B), stores the quotient in (C), and the remainder in (C)+1.	4~6	95	
	31	Double-length addition	–[(A) +1·(A) D+(B) +1· (b) → (C)+1·(C)]–	Adds the contents of (B)+1 • (B) to the contents of (A)+1 and (A), and stores the result in (C)+1 • (C).	4~8	102	
	32	Double-length subtraction	–[(A) +1·(A) D–(B) +1· (b) → (C)+1·(C)]–	Subtracts the contents of (B)+1 and (B) from the content of (A)+1 • (A), and stores the result in (C)+1 • (C).	4~8	103	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Arithmetic operations	33	Double-length multiplication	-[(A) +1·(A) D❋·(B) +1· (B) → (C)+1·(C)]-	Multiplies the contents of (A)+1•(A) by the contents of (B)+1•(B), and stores the result in (C)+1, (C)+2, (C)+1•(C).	4~8	184	
	34	Double-length division	-[(A) +1·(A) D/·(B) +1· (B) → (C)+1·(C)]-	Divides the contents of (A)+1 and (A) by the contents of (B)+1•(B), and stores the quotient in (C)+1•(C) and the remainder in (C)+4•(C)+3.	4~8	170	
	35	Addition with carry	-[(A) +C (B) → (C)]-	Adds the contents of the carry flag and the contents of (B) to the contents of (A), and stores the result in (C). The carry flag changes according to the operation result.	4~6	87	
	36	Subtraction with carry	-[(A) –C (B) → (C)]-	Subtracts the contents of (B) and the contents of the carry flag from the contents of (A), and stores the result in (C). The carry flag changes according to the operation result.	4~6	87	
	37	Double-length addition with carry	-[(A) +1·(A) D+C (B) +1·(B) → (C)+1·(C)]-	Adds the contents of the carry flag to the contents of (A)+1•(A) and the contents of (B)+1•(B), and stores the result in (C)+1•C). The carry flag changes according to the operation result.	4~8	125	
	38	Double-length subtraction with carry	-[(A) +1·(A) D-C (B) +1·(B) → (C)+1·(C)]-	Subtracts the contents of (B)+1•(B) plus the contents of the carry flag from the contents of (A)+1 and (A), and stores the result in (C)+1•(C). The carry flag changes according to the operation result.	4~8	124	
	39	Unsigned multiplication	-[(A) U❋ (B) → (C) +1·(C)]-	Multiplies the contents of (A) by the contents of (B), and stores the result in (C)+1 (unsigned integer calculation).	4~6	82	
	40	Unsigned division	-[(A) U/ (B) → (C)]-	Divides the contents of (A) by the contents of (B), and stores the quotient in (C), and the remainder in (C)+1 (unsigned integer operation).	4~6	85	
	41	Unsigned double/single division	-[(A)+1·(A) DIV (B) → (C)]-	Divides the contents of (A)+1•(A) by the contents of (B), stores the quotient in (C), and the remainder in (C)+1 (unsigned integer operation).	4~7	111	
	43	Increment	-[+ 1 (A)]-	Increments the contents of (A) by 1.	2	52	
	44	Double-length increment	-[D + 1 (A) + 1·(A)]-	Increments the contents of (A)+1 and (A) by1.	2	81	
	45	Decrement	-[–1 (A)]-	Decrements the contents of (A) by 1.	2	52	
	46	Double-length decrement	-[D – 1 (A) +1·(A)]-	Decrements the contents of (A)+1•(A) by just 1.	2	81	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Arithmetic operations	208	Floating point addition	-[(A) +1· (A) F+(B)+1·(B) → (C) +1·(C)]-	Adds the floating point data of (A)+1 • (A) and (B)+1 •(B), and stores the result in (C)+1 • (C).	4	107	396µs (max)
	209	Floating point subtraction	-[(A) +1· (A) F-(B)+1·(B) → (C) +1·(C)]-	Subtracts the floating point data of (B)+1•(B)from (A)+1 • (A) ,and stores the result in (C)+1 • (C).	4	108	399µs (max)
	210	Floating point multiplication	-[(A) +1· (A) F*(B)+1·(B) → (C) +1·(C)]-	Multiplies the floating point data of (A)+1 • (A) by (B)+1•(B), and stores the result in (C)+1 • (C).	4	132	533µs (max)
	211	Floating point division	-[(A) +1·(A) F/(B)+1·(B) → (C) +1·(C)]-	Divides the floating point data of (A)+1 • (A) by (B)+1•(B), and stores the result in (C)+1 • (C).	4	133	728µs (max)
Logical operations	48	AN D	-[(A) AND (B) → (C)]-	Finds the logical AND of (A) and (B) and stores it in (C).	4~6	67	
	49	Double-length AND	-[(A) +1· DAND (B)+1·(B) → (C) +1·(C)]-	Finds the logical AND of (A)+1 and (A) and (B)+1 •(B) and stores it in (C)+1 and (C).	4~8	100	
	50	OR	-[(A) OR (B) → (C)]-	Finds the logical OR of (A) and (B) and stores in (C).	4~6	66	
	51	Double-length OR	-[(A) +1·(A) DOR (B)+1·(B) → (C)+1·(C)]-	Finds the logical OR of (A)+1 and (A) and (B)+1•(B) and stores it in (C)+1 and (C).	4~8	100	
	52	Exclusive OR	-[(A) EOR (B) → (C)]-	Finds the exclusive logical OR of (A) and (B) and stores it in (C).	4~6	66	
	53	Double-length exclusive OR	-[(A)+1·(A) DEOR (B)+1·(B) → (C)+1·(C)]-	Finds the exclusive logical OR of (A)+1• (A) and (B)+1 and (B) and stores it in (C)+1•(C).	4~8	100	
	54	Not exclusive OR	-[(A) ENR (B) → (C)]-	Fins the negative exclusive logical OR of (A) and (B) and stores it in (C).	4~6	66	
	55	Double-length Notexclusive OR	-[(A)+1·(A) DENR (B)+1·(B) → (C)+1·(C)]-	Finds the negative exclusive logical OR of (A)+1•(A) and (B)+1 and (B) and stores it in (C)+1•(C).	4~8	101	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (μs)	Remarks
Logical Operations	57	Table AND	-[(A) TAND (n) (B) → (C)]-	Finds the logical AND of the table of size n headed by (A) and the table of size n headed by (B), and stores it in the location headed by (C).	5	301+8.5n	
	58	Table OR	-[(A) TOR (n) (B) → (C)]-	Finds the logical OR of the table of size n headed by (A) and the table of size n headed by (B), and stores it in the location headed by (C).	5	301+8.5n	
	59	Table exclusive OR	-[(A) TEOR (n) (B) → (C)]-	Finds the exclusive lOR of the table of size n headed by (A) and the table of size n headed by (B), and stores it in the location headed by (C).	5	301+8.5n	
	60	Table Not exclusive OR	-[(A) TENR (n) (B) → (C)]-	Finds the NOT exclusive OR of the table of size n headed by (A) and the table of size n headed by (B) and stores it in the location headed by (C).	5	301+8.5n	
	64	Test	-[(A) TEST (B)]-	Turns the output ON if the logical AND of (A) and (B) is other than 0.	3~5	80	
	65	Double-length test	-[(A) +1·(A) DTST (B)+1·(B)]-	Turns the output ON if the logical AND of (A)+1•(A) plus (B)+1•(B) is other than 0.	3~7	113	
	66	Bit file bit test	-[(A) TTST (n) (B)]-	Decides the ON/OFF state of the (A)th bit of the bit table size n headed by (B).	5~6	148	
Shifts	68	1 bit shift right	-[SHR 1 (A)]-	Shifts the data in (A) 1 bit to the right (LSB direction) and stores the result in (A). The carry flag changes according to the result.	2	92	
	69	1 bit shift left	-[SHL 1 (A)]-	Shifts the data in (A) 1 bit to the left (MSB direction) and stores the result in (A). The carry flag changes according to the result.	2	92	
	70	1 bit shift right	-[(A) SHR n → (B)]-	Shifts the data in (A) n bits to the right (LSB direction) and stores the result in (B). The carry flag changes according to the result.	4~5	97	
	71	1 bit shift left	-[(A) SHL n → (B)]-	Shifts the data in (A) n bits to the left (MSB direction) and stores the result in (B). The carry flag changes according to the result.	4~5	97	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (μs)	Remarks
Shift	72	m bit file n bits shift right	[(A) TSHR (m) → (B)]	When (B) is a register: Takes the m-word table headed by (B), and shifts it to the right (low address direction) by the number of words indicated by (A). When (B) is a device: Takes the m-bit file headed by (B), and shifts it to the right (LSB direction) by the number of bits indicated by (A). The carry flag changes according to the result.	4~5	*	
	73	m bit file n bits shift left	[(A) TSHL (m) → (B)]	When (B) is a register: Takes the m-word table headed by (B), and shifts it to the left (high address direction) by the number of words indicated by (A). When (B) is a device: Takes the m-bit file headed by (B), and shifts it to the left (MSB direction) by the number of bits indicated by (A). The carry flag changes according to the result.	4~5	*	
	74	Shift register	D SR Q S (n) E (A)	If the enable input (E) is ON, then when the shift input (S) comes ON, the instruction takes the contents of the n devices headed by the device (A) and shifts them 1 bit to the left. The carry flag changes according to the result.	3	116	
	75	Bidirectional shift register	D DSR Q S (n) E L (A)	If the enable input (E) is ON, then when the shift input (S) comes ON, the instruction takes the contents of the n devices headed by the device (A) and shifts them 1 bit to the left or to the right (the shift direction depends on the state of the direction input (L)). The carry flag changes according to the result.	3	120	
	76	Device shift	[SFT (A)]	Takes the contents of the device ((A)-1) which immediatety precedes the device (A), stores them in (A), and sets (A)-1 to 0	2	94	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Rotate	78	1 bit rotate right	-[RTR 1 (A)]-	Rotates the data in (A) 1 bit to the right (LSB direction). The carry flag changes according to the result.	2	92	
	79	1 bit rotate left	-[RTL 1 (A)]-	Rotates the data in (A) 1 bit to the left (MSB direction). The carry flag changes according to the result.	2	92	
	80	n bits rotate right	-[(A) RTR n → (B)]-	Rotates the data in (A) n bits to the right (LSB direction). The carry flag changes according to the result.	4~5	97	
	81	n bits rotate left	-[(A) RTL n → (B)]-	Rotates the data in (A) n bits to the left (MSB direction). The carry flag changes according to the result.	4~5	97	
	82	m-it file n bits rotate right	-[(A) TRTR (m) (B)]-	When (B) is a register: Takes the table of m words, headed by (B), and rotates it to the right (low address direction) by the number of words specified by (A). When (B) is a device: Takes the bit file of m bits, headed by (B), and rotates it to the right (LSB direction) by the number of bits specified by (A). The carry flag changes according to the result.	4~5	*	
	83	m-bit file n bits rotate left	-[(A) TRTL (m) (B)]-	When (B) is a register: Takes the table of m words, headed by (B), and rotates it to the left (high address direction) by the number of words specified by (A). When (B) is a device: Takes the bit file of m bits, headed by (B), and rotates it to the left (MSB direction) by the number of bits specified by (A). The carry flag changes according to the result.	4~5	*	
	84	1 bit rotate right with carry	-[RRC 1 (A)]-	Rotates the data in (A) 1 bit to the right (LSB direction) including the carry flag. The carry flag changes according to the result.	2	102	

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Rotate	85	1 bit rotate left with carry	-[RLC 1 (A)]-	Rotates the data in (A) 1 bit to the left (MSB direction) including the carry flag. The carry flag changes according to the result.	2	102	
	86	n bits rotate right with carry	-[(A) RRC n → (B)]-	Rotates the data in (A) n bits to the right (LSB direction) including the carry flag, and stores the result in (B). The carry flag changes according to the result.	4~5	109	
	87	n bits rotate left with carry	-[(A) RLC n → (B)]-	Rotates the data in (A) n bits to the left (MSB direction), including the carry flag, and stores the result in (B). The carry flag changes according to the result.	4~5	109	
	88	m-bit file n bits rotate right with carry	-[(A) TRRC (m) (B)]-	If (B) is a register: Takes the table of m words headed by (B) and rotates it to the right (low address direction) by the number of words indicated by (A). (Same as register specification in FUN82.) If (B) is a device: Takes the bit file of m bits headed by (B), including the carry flag, and rotates it to the right (LSB direction) by the number of bits indicated by (A). The carry flag changes according to the result.	4~5	*	
	89	m-bit file n bits rotate left with carry	-[(A) TRLC (m) (B)]-	If (B) is a register: Takes the table of m words headed by (B) and rotates it to the left (high address direction) by the number of words indicated by (A). (Same as register specification in FUN83.) If (B) is a device: Takes the bit file of m bits headed by (B), including the carry flag, and rotates it to the left (MSB direction) by the number of bits indicated by (A). The carry flag changes according to the result.	4~5	*	
	90	Multiplexer	-[(A) MPX (n) (B) → (C)]-	Takes the contents of the (B)th register in the table of size n headed by the register (A), and stores them in the register (C).	7~9	139	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Rotate	91	Demultiplexer	┤[(A) DPX (n) (B) → (C)]├	Stores the contents of the register (A) in the (B)th register of the table of size n headed by the register (C).	7~9	154	
	92	Table -> bit transfer	┤[(A) TBM (n) (B) → (C)]├	Takes the (B)th bit from the head of the table of size n words headed by the register (A) and stores it in the device (C).	6~7	206	
	93	Bit -> table transfer	┤[(A) BTM (n) (B) → (C)]├	Takes the contents of the device (A) and stores them in the (B)th bit of the table of size n headed by the register (C).	6~7	189	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Compare	95	Bit file comparison	-[(A) TCMP (n) (B) → (C)]-	Compares the register tables starting from (A) and (B), and stores the non-matching bits in (C).	3~5	*	
	96	Greater than	-[(A) > (B)]-	Turns output ON if (A) > (B) (integer comparison).	3~5	61	
	97	Greater than or equal to	-[(A) > = (B)]-	Turns output ON if (A) (B) (integer comparison).	3~5	60	
	98	Equal to	-[(A) = (B)]-	Turn output ON if (A) = (B) (integer comparison).	3~5	60	
	99	Not equal to	-[(A) < > (B)]-	Turns output ON if (A) (B) (integer comparison).	3~5	60	
	100	Smaller than	-[(A) < (B)]-	Turns output ON if (A) < (B) (integer comparison).	3~5	61	
	101	Smaller than or equal to	-[(A) < = (B)]-	Turns output ON if (A) (B) (integer comparison).	3~5	61	
	102	Double-length greater than	-[(A) +1·(A) D> (B)+1·(B)]-	Turns output ON if (A)+1 and (A) > (B)+1•(B) (double- length integer comparison).	3~7	89	
	103	Double-length greater than or equal to	-[(A) +1·(A) D>= (B)+1·(B)]-	Turns output ON if (A)+1 and (A) (B)+1•(B) (double- length integer comparison).	3~7	88	
	104	Double-length equal to	-[(A) +1·(A) D= (B) +1·(B)]-	Turns output ON if (A)+1 and (A) = (B)+1•(B) (double- length integer comparison).	3~7	83	
	105	Double-length not equal to	-[(A) +1·(A) D< > (B)+1·(B)]-	Turns output ON if (A)+1 and (A) (B)+1•(B) (double- length integer comparison).	3~7	83	
	106	Double-length smaller than	-[(A) +1·(A) D< (B)+1·(B)]-	Turns output ON if (A)+1 and (A) < (B)+1•(B) (double- length integer comparison).	3~7	89	
	107	Double-length smaller than or equal to	-[(A) +1·(A) D< = (B)+1·(B)]-	Turns output ON if (A)+1 and (A) (B)+1•(B) (double- length integer comparison).	3~7	89	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Compare	108	Unsigned greater than	-[(A) U > (B)]-	Turns output ON if (A) > (B) (unsigned integer comparison).	3~5	61	
	109	Unsigned greater than or equal to	-[(A) U > = (B)]-	Turns output ON if (A) (B) (unsigned integer comparison).	3~5	61	
	110	Unsigned equal to	-[(A) U = (B)]-	Turns output ON if (A) = (B) (unsigned integer comparison).	3~5	61	
	111	Unsigned not equal to	-[(A) U < > (B)]-	Turns output ON if (A) (B) (unsigned integer comparison).	3~5	61	
	112	Unsigned smaller than	-[(A) U < (B)]-	Turns output ON if (A) < (B) (unsigned integer comparison).	3~5	61	
	113	Unsigned smaller than or equal to	-[(A) U < = (B)]-	Turns output ON if (A) (B) (unsigned integer comparison).	3~5	61	
	212	Floating point greater than	-[(A) +1·(A) F> (B)+1·(B)]-	Turns output ON if (A)+1•(A) > (B)+1•(B) (floating point data comparison).	3	84	208µs (max)
	213	Floating point greater than or equal	-[(A) +1·(A) F> = (B)+1·(B)]-	Turns output ON if (A)+1•(A) (B)+1•(B) (floating point data comparison).	3	84	206µs (max)
	214	Floating point equal	-[(A) +1·(A) F = (B)+1·(B)]-	Turns output ON if (A)+1•(A) = (B)+1•(B) (floating point data comparison).	3	83	158µs (max)
	215	Floating point not equal	-[(A) +1·(A) F< > (B)+1·(B)]-	Turns output ON if (A)+1•(A) (B)+1•(B) (floating point data comparison).	3	84	159µs (max)
	216	Floating point less than	-[(A) +1·(A) F< (B)+1·(B)]-	Turns output ON if (A)+1•(A) < (B)+1•(B) (floating point data comparison).	3	84	206µs (max)
	217	Floating point less than or equal	-[(A) +1·(A) F< = (B)+1·(B)]-	Turns output ON if (A)+1•(A) (B)+1•(B) (floating point data comparison).	3	84	208µs (max)

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Special data processing	114	Set device/register	-[SET (A)]-	If (A) is a device: Sets device (A) to ON.	2	93	
				If (A) is a register: Stores HFFFF in register (A).		78	
	115	Rerest device/register	-[RST (A)]-	If (A) is a device: Resets device (A) to OFF.	2	93	
				If (A) is a register: Stores 0 in register (A).		78	
	116	Table bit set	-[(A) TSET (n) (B)]-	From the bit file of n words, headed by the register (B), the instruction takes the bit in the location indicated by (A) and sets it to ON.	4~5	131	
	117	Table bit reset	-[(A) TRST (n) (B)]-	From the bit file of n words, headed by the register (B), the instruction takes the bit in the position indicated by (A) and resets it to OFF.	4~5	131	
	118	Set carry	-[SETC]-	Sets the carry flag.	1	41	
	119	Reset carry reset	-[RSTC]-	Resets the carry flag.	1	41	
	120	Encode	-[(A) ENC (n) (B)]-	In the bit file of size 2^n bits headed by (A), the instruction stores the uppermost ON bit position in register (B).	3~4	210	
	121	Decode	-[(A) DEC (n) (B)]-	Takes the bit file of size 2^n bits headed by (B), sets the bit position indicated by the lower n bits of register (A) to ON, and sets all the rest to OFF.	3~4	190	
	122	Bit count	-[(A) BC (B)]-	Counts the number of ON bits in the data in (A) and stores the result in (B).	3~4	170	
	123	Double-length bit count	-[(A) DBC (B)]-	Counts the number of ON bits in the double-length data in (A)+1•(A), and stores the result in (B).	3~5	285	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (μs)	Remarks
Special data processing	124	Data search	─[(A) SCH (n) (B) → (C)]─	Searches through data table of n words headed by (B) for data matching the contents of (A). Stores the number of matches in (C), and stores the lowest register address of the matching registers in (C)+1.	5~6	176	
	125	Push	─[(A) PUSH (n) (B) → (C)]─	Pushes the data in (A) into the table of n words headed by (C), and increments the value of (B) by 1.	5~6	147	
	126	Pop last	─[(A) POPL (n) (B) → (C)]─	Takes out the data pushed in last to the table of n words headed by (A) and stores it in (C). Also decrements the value of (B) by 1.	5	143	
	127	Pop first	─[(A) POPF (n) (B) → (C)]─	Takes out from the table of n words headed by (A) the data which was pushed in first, and stores it in (C). Also decrements the value of (B) by 1.	5	133	
	147	Flip-flop	S F/F Q R (A)	When the set input (S) is ON, the instruction sets the device (A) to ON; when the reset input (R) is ON, it resets the device (A) to OFF. (Reset takes priority)	2	73	
	149	Up-down counter	U U/D Q C E (A)	If the enable input (E) is ON, the instruction counts the number of times the count input (C) has come ON and stores it in the counter register (A). The selection of the count direction (increment/decrement) is made according to the state of the up/down selection input (U) (see below). ON: UP count (increment) OFF: DOWN count (decrement)	2	59	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Program control	128	Subroutine call	-[CALL N. nn]-	If the input is ON, the instruction calls the subroutine for the subroutine number nn.	2~3	92	
	129	Subroutine return	|-[RET]-|	Indicates the end of the subroutine.	1	73	
	130	Conditional jump	-[JUMP N. nn]-	If the input is ON, jumps directly to the label for the label number nn.	2~3	67	
	136	Jump label	|-[LBL (nn)]-|	Indicates the jump destination for the conditional jump.	2	40	
	132	FOR-NEXT loop	-[FOR n]-|	Executes the section from FOR to NEXT the number of times specified by n.	2	90	
	133	NEXT NEXT loop	|-[NEXT]-|		1	55	
	137	Subroutine entry	|-[SUBR (nn)]-|	Indicates the entrance to the subroutine (number nn).	2	40	
	138	STOP	-[STOP]-	Stops the program	1		
	140	Enable interrupt	-[EI]-	Enables execution of the interrupt program.	1	41	
	141	Disable interrupt	-[DI]-	Disables execution of the interrupt program.	1	61	
	142	Interrupt program end	|-[IRET]-|	Indicates the end of the interrupt program.	1	41	
	143	Watchdog timer reset	-[WDT n]-	Extends the scan time over detection time	2	68	
	144	Step sequence initialize	-[STIZ (n) (A)]-	Turns OFF the n devices headed by device (A), and turns (A) ON (activation of step sequence). These comprise one step sequence.	3	124	
	145	Step sequence input	|-[STIZ (A)]-	Turns output ON when input is ON and device (A) is ON.	2	97	
	146	Step sequence output	-[STIZ (A)]-|	When input is ON, the instruction turns OFF the devices with step sequence input instructions on the same route, and turns device (A) ON.	2	70	
	241	SFC initialize	-[SFIZ (n) (A)]-	When input is changed from OFF to ON, the instruction resets the n steps from the SFC step (A), and activates step (A) (activation of SFC).	3	113	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (μs)	Remarks
RAS	150	Diagnostic display	┤[DIAG (A) (B)]├	When input has changed from OFF to ON, the instruction records the error code indicated by (A) in the special register, and turns ON the corresponding annunciator relay. The error messages (max 12 characters) recorded in the register tables headed by (B) can be monitored on the peripheral devices.	3~4	116	
	151	Diagnostic display reset	┤[DIAR (A)]├	Erases the error code (A) from the error code list recorded by the diagnostic display instruction (FUN150) and from the annunciator relay.	2~3	96	
	152	Status latch set	┤[STLS]├	Takes the devices/registers (max 32) previously set by the programmer and stores them in the latch area.	1	416	
	153	Status latch reset	┤[STLR]├	Cancels the state of the status latch.	1	42	
	154	Set Calendar	┤[(A) CLDS]├	Takes the 6 words of data headed by the register A and sets them in the calendar LSI (date and time setting).	2	194	
	155	Calendar operation	┤[(A) CLNBD (B)]├	Subtracts the 6 words of date and time data headed by (A), from the current date and time, and stores the result in the 6 words starting with (B).	3	222	
	158	Drum sequencer	┤[(A) DRUM (n) (B) → (C) (m)]├	Compares the count value (B) with the count value setting table ((A)+2n onwards), then decides the step number and stores it in (B)+1. Using the data output pattern table (A), the instruction looks up the output pattern corresponding to this step number and outputs it to the bit table (C).	6	*	
	159	Cam sequencer	┤[(A) CAM (n) (B) → (C)]├	Compares the register (B) with the activation and deactivation setting value for table (A), and carries out ON/OFF control on the corresponding devices.	5	*	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Function	56	Moving average	-[(A) MAVE (n) (B) → (C)]-	Moves the data of (A) to the n-word data table headed by (B) and calculates the average value and stores it in (C).	5	237+18.4n	
	61	Digital Filter	-[(A) DFL (B) → (C)]-	Calculates the data of (A) by the value of (B) according to the filtering function and stores it in (C).	4	109	
	160	Upper limit	-[(A) UL (B) → (C)]-	Applies an upper limit to the contents of (A) using the value of (B), and stores the results in (C).	4~6	88	
	161	Lower limit	-[(A) LL (B) → (C)]-	Applies a lower limit to the contents of (A), using the value of (B), and stores the results in (C).	4~6	88	
	162	Maximum value	-[(A) MAX (n) (B)]-	Searches the n-word data table headed by (A) for the maximum value, stores the maximum value in (B), and stores the pointer with the maximum value in (B)+1.	4	150+18n	
	163	Minimum value	-[(A) MIN (n) (B)]-	Searches the n-word data table headed by (A) for the minimum value, stores the minimum value in (A), and stores the pointer with the minimum value in (B)+1.	4	150+18n	
	164	Average value	-[(A) AVE (n) (B)]-	Calculates the average value for the n-word data table headed by (A), and stores it in (B).	4	138+17n	
	165	Function generator	-[(A) FG (n) (B) → (C)]-	Using the function defined by the 2x n parameters headed by (B), finds the function value which takes the contents of (A) as its argument, and stores it in (C).	5~6	140	
	166	Dead band	-[(A) DB (B) → (C)]-	Finds the value which gives the dead band indicated by (B) for the contents of (A), and stores it in (C).	4~6	91	
	167	Square root	-[(A)+1·(A) RT (B)]-	Finds the square root of the double-length data (A)+1 and (A), and stores it in (B).	3~5	102	
	168	Integral	-[(A) INTG (B) → (C)]-	Calculates the integral for the value of (A) from the integral constant for (B)+1 and (B), and stores the result in (C)+1•(C).	4~5	154	
	169	Ramp function	-[(A) RAMP (B) → (C)]-	Generates the ramp function for the value of (A) the parameters starting with (B), and stores it in (C).	4~5	246	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Function	170	PID	-[(A) PID (B) → (C)]-	Carries out the PID calculation for the value of (A) by the parameters starting with (B), and stores it in (C).	4	475	
	171	Deviation square PID	-[(A) PID2 (B) → (C)]-	Carries out the deviation square PID calculation for the value of (A) using the parameters starting with (B), and stores it in (C).	4	*	
	156	Essential PID	-[(A) PID3 (B) → (C)]-	Carries out the essential PID calculation for the value of (A) using the parameters starting with (B), and stores it in (C)	4	*	
	172	Sine function (SIN)	-[(A) SIN (B)]-	Stores in (B) the value obtained by taking the angle (degree) obtained by dividing the value of (A) by 100 and multiplying its sine value by 10000.	3~4	110	
	173	Cosine function (COS)	-[(A) COS (B)]-	Stores in (B) the value obtained by taking the angle (degree) obtained by dividing the value of (A) by 100 and multiplying its cosine value by 10000.	3~4	111	
	174	Tangent function (TAN)	-[(A) TAN (B)]-	Stores in (B) the value obtained by taking the angle (degree) obtained by dividing the value of (A) by 100 and multiplying its tangent value by 10000.	3~4	*	
	175	Arc sine function (SIN-1)	-[(A) ASIN (B)]-	Divides the value of (A) by 10000, multiplies the arc sine value by 100, then stores it in (B).	3~4	78	
	176	Arc cosine function (COS-1)	-[(A) ACOS (B)]-	Divides the value of (A) by 10000, multiplies the arc cosine value by 100, then stores it in (B).	3~4	74	
	177	Arc tangent function (TAN-1)	-[(A) ATAN (B)]-	Divides the value of (A) by 10000, multiplies the arc tangent value by 100, then stores it in (B).	3~4	*	
	178	Exponential function	-[(A) EXP (B)+1·(B)]-	Finds the exponential of 1/1000 of the absolute value of (A) and stores it in (B)+1•(B).	3~4	*	
	179	Logarithm	-[(A) LOG (B)]-	Calculates the common logarithm of the absolute value of (A), multiplies it by 1000 and stores the result in (B).	3~4	*	

Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (μs)	Remarks
Conversion	62	HEX-ASCII conversion	-[(A) HTOA (n) (B)]-	Converts the HEX data in n registers headed by (A) into ASCII data and stores them in the registers headed by (B).	4	160+75.5n	
	63	ASCII-HEX conversion	-[(A) ATOH (n) (B)]-	Converts the ASCII data in n registers headed by (A) into HEX data and stores them in the registers headed by (B).	4	143+39.4n	
	180	Absolute value	-[(A) ABS (B)]-	Stores the absolute value of (A) in (B).	3~4	70	
	181	Double-length absolute value	-[(A) +1· (A) DABS (B) +1· (B)]-	Stores the absolute value of (A)+1 and (A) in (B)+1•(B).	3~5	103	
	182	2's complement	-[(A) NEG (B)]-	Stores the 2's complement of (A) in (B).	3~4	68	
	183	Double-length 2's complement	-[(A) +1· (A) DNEG (B) +1· (B)]-	Stores the 2's complement of (A)+1•(A) in (A)+1• (B).	3~5	103	
	184	Double lenght conversion	-[(A) DW (B) +1· (B)]-	Converts the signed data in (A) into double-length data, and stores in (B)+1•(B).	3~4	85	
	185	7-segment decode	-[(A) 7 SEG (B)]-	Converts the bottom 4 bits of (A) into 7-segment code, and code stores in (B).	3~4	73	
	186	ASCII conversion	-[(A) ASC (B)]-	Takes the alphanumerics (maximum 16 characters) indicated by (A) and converts them into ASCII code. Stores the result in the location headed by (B).	3~10	262	
	188	Binary conversion	-[(A) BIN (B)]-	Converts the BCD data in (A) into binary data and stores it in (B).	3~4	105	
	189	Double-length binary conversion	-[(A) +1· (A) DBIN (B) +1· (B)]-	Converts the double-length BCD data in (A)+1•(A) into binary data and stores it in (B)+1•(B).	3~5	175	
	190	BCD conversion	-[(A) BCD (B)]-	Converts the binary data in (A) into BCD data and stores in in (B).	3~4	101	
	191	Double-length BCD conversion	-[(A) +1·(A) DBCD (B) +1· (B)]-	Converts the binary data in (A)+1•(A) into BCD data and stores it in (B)+1•B).	3~5	169	
	204	Floating point conversion	-[(A) +1· (A) FLT (B) +1· (B)]-	Converts the double-length integer of (A)+1•(A) into floating point data and stores it in (B)+1•(B).	3~5	106	363μs (max)
	205	Fixed point conversion	-[(A) +1· (A) FIX (B) +1· (B)]-	Converts the floating point data of (A)+1•(A) into double-length integer data and stores it in (B)+1•(B).	3	96	320μs (max)

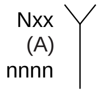
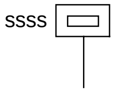
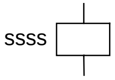
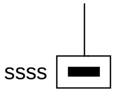
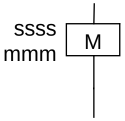
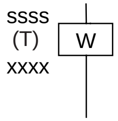
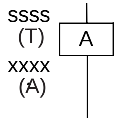
Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
BCD operation	192	BCD additon	–[(A) B + (B) → (C)]–	Carries out BCD addition of the contents of (A) and (B), and stores the result in (C).	4~6	205	
	193	BCD subtraction	–[(A) B – (B) → (C)]–	Subtracts the contents of (B) from the contents of (A) in BCD, and stores the result in (C).	4~6	197	
	194	BCD multiplication	–[(A) B ❋ (B) → (C) +1·(C)]–	Multiplies the contents of (A) and (B) together in BCD, and stores the result in (C)+1•(C).	4~6	247	
	195	BCD division	–[(A) B / (B) → (C)]–	Divides the contents of (A) by the contents of (B) in BCD, and stores the quotient in (C) and the remainder in (C)+1.	4~6	250	
	196	Double-length BCD addition	–[(A) +1·(A) DB + (B) +1·(B) → (C)+1· (C)]–	Adds the contents of (B)+1 •(B) to the contents of (A)+1 and (A) in BCD, and stores the result in (C)+1 •(C).	4~8	372	
	197	Double-length BCD subtraction	–[(A) +1·(A) DB – (B) +1·(B) → (C)+1· (C)]–	Subtracts the contents of (B)+1•(B) from the contents of (A)+1 and (A) in BCD, and stores the result in (C)+1•(C).	4~8	365	
	198	Double-length BCD multiplication	–[(A) +1·(A) DB* (B) +1·(B) → (C)+1· (C)]–	Multiplies the contents of (A)+1•(A) by the contents of (B)+1•(B) in BCD, and stores the result in (C)+3, (C)+2, (C)+1•(C).	4~8	672	
	199	Double-length BCD division	–[(A) +1·(A) DB / (B) +1·(B) → (C)+1· (C)]–	Divides the contents of (A)+1•(A) by the contents of (B)+1•(B) in BCD, and stores the quotient in (C)+1 •(C) and the remainder in (C)+3•(C)+2.	4~8	539	
	200	BCD addition with carry	–[(A) B +C (B) →(C)]–	Adds (B) plus the contents of the carry flag to (A) in BCD, and stores the result in (C). The carry flag changes according to the operation result.	4~6	222	
	201	BCD subtraction with carry	–[(A) B – C (B) → (C)]–	Subtracts (B) plus the contents of the carry flag from (A) in BCD, and stores the result in (C). The carry flag changes according to the operation result.	4~6	216	
	202	Double-length BCD addition with carry	–[(A) +1· (A) DB +C (B) +1·(B) → (C) +1· (C)]–	Adds the contents of (B)+1•(B), plus the contents of the carry flag, to (A)+1•(A) in BCD, and stores the result in (C)+1•(C). The carry flag changes according to the operation result.	4~8	390	
	203	Double-length BCD subtraction with carry	–[(A) +1· (A) DB – C (B) +1·(B) → (C) +1· (C)]–	Subtracts (B)+1 and (B) plus the contents of the carry flag from (A)+1•(A) in BCD, and stores the result in (C)+1•(C). The carry flag changes according to the operation result.	4~8	383	

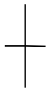
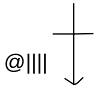
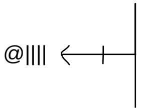
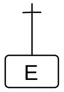
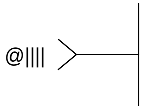
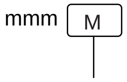
Ladder Diagram Instructions (Function Instructions)

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	Remarks
Input/ output	235	Direct I/O	–[I / O (n) (A)]–	Using the n-word register range headed by the input/output register (A), the instruction carries out input/output of data from/to the corresponding I/O module.	3	163+63n	
	236	Expanded data transfer	–[(A) XFER (B) → (C)]–	Transfers the word block of size (B) from the transfer source indirectly specified by the register (A) to the transfer destination indirectly specified by the register (C) or has special functions.	4	*	
	237	Special module data read	–[(A) READ (B) → (C)]–	Carries out data transfer from the expanded memory of the special module to the user register area.	4	430+5.6n	
	238	Special module data write	–[(A) WRITE (B) → (C)]–	Transfers the contents of the user register area to the expanded memory area of the special module.	4	427+10.6n	

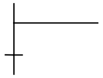
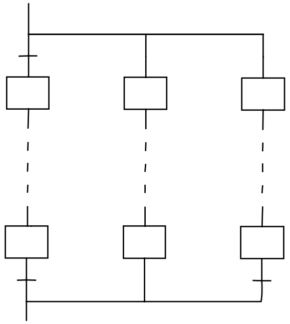
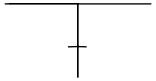
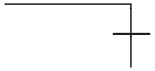
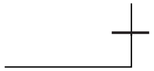
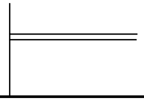
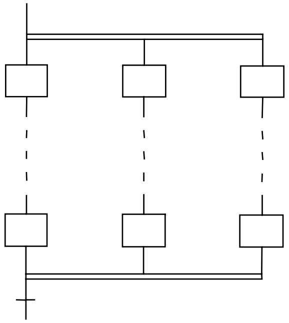
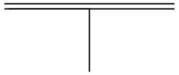
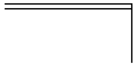
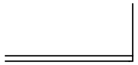
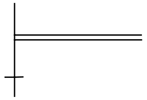
SFC Instructions

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	
						Inactive	Active
SFC initialize		SFC initialize		When the device (A) has changed from OFF to ON, the instruction inactivates the nnnn steps of the succeeding SFC program, and activates the initial step (SFC activation).	4	81	212+10.6 INT (hn/16–1)
SFC step		Initial step		Indicates the start of the SFC program and contains action program which correspond on a one-to-one basis. ssss is the step address.	2 (excluding action)	98	127
		Step		This is the single unit of control. It contains action program which correspond on a one-to-one basis. ssss is the step address.	1 (excluding action)	0.44	84
		End step		Indicates the end of the SFC program. Returns processing to the corresponding initial step when the immediately preceding transition condition holds true. ssss is the initial step address.	2	0.54	78
		Macro step		Corresponds on a one-to-one basis to the macro program indicated by mmm. ssss is the step address, and mmm is the macro address.	3	63	110
		Wait step		Even if the immediately preceding transition condition holds true, this instruction does not carry out the transition until the set period has elapsed. It has action program which correspond on a one-to-one basis. ssss is the step address, {T} is the timer register, and xxxx is the set period.	4 (excluding action	88	95
		Alarm step		Monitors the active period, and if the transition has not been made within the set period, sets the alarm device (A) to ON. Contains execution action program which correspond on a one-to-one basis. ssss is the step address, {T} is the timer register, and xxxx is the set period.	4 (excluding action)	105	115

SFC Instructions

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	
						Inactive	Active
Transition		Transition		ndicates the condition for transition between steps. Contains transition condition which correspond on a one-to-one basis.	1 (excluding condition)	0.32	96
		SFC End		Indicates end of SFC program. Jumps to the label indicated by IIII when the transition condition holds true. Contains transition condition which correspond on a one-to-one basis.	2 (excluding condition)	0.54	114
		SFC Jump		Indicates jump to desired step. Jumps to the step indicated by IIII when the condition holds true. Contains jump condition details which correspond on a one-to-one basis.	5 (excluding condition)	1.19	120
		Macro end		Indicates the end of the macro program. Contains transition condition which correspond on a one-to-one basis.	2 (excluding condition)	0.54	117
Label		SFC Label		Indicates the return destination from the SFC end, or the jump destination from the SFC jump.	2	55	117
		Macro entry		Indicates start of macro program.	1	0.43	0.43

SFC Instructions

Group	FUN No.	Name	Representation	Summary	Number of steps required	Execution time required (µs)	
						Inactive	Active
Sequence selection		Sequence selection Divergence (I)		From among several connected steps, activates the step for which the transition conditions hold true (left priority). 	2 x n-1 n is the branch count (Excluding transitions, steps, and individual details within the branch)	0.54	118
		Sequence selection Divergence (II)				0.54	118
		Sequence selection Divergence (III)				0.54	96
		Sequence selection Convergence				0.22	0.22
Simultaneous sequence		Simultaneous sequence Divergence (I)		Activates all the connected steps. 	n + 3 n is the branch count (Excluding transitions, steps, and individual details within the branch)	0.22	0.22
		Simultaneous sequence Divergence (II)				0.22	0.22
		Simultaneous sequence Divergence (III)				0.22	0.22
		Simultaneous sequence Convergence (I)				49	56
		Simultaneous sequence Convergence (II)				0.22	197

Supplementary information of instructions execution time

The instruction execution time in the T2N is subject to increase due to operand modification condition. (per one operand)

Operand condition	single-length	double-length
Index modification	58	140
Didit designation	54	–
Direct I/O (IW / OW)	94	172
Direct I/O with digit designation(IW / OW)	130	–

Supplementary information

Remarks No.		execution time (μs)
FNC018 (MOV)	Register to register	1.20
	Constant value to register	59
FNC027 (+)	Register + register	1.63
	Constant value + register	67
FNC028 (-)	Register - register	1.63
	Constant value - register	67

APPENDICES

Name	Specifications	Type	Part number

● Basic configuration

Rack	8-slot I/O dedicated basic unit (expansion connectable)	BU228N	TBU228N*S
	8-slot I/O expansion unit	BU268	TBU268**S
	6-slot I/O expansion unit	BU266	TBU266**S
Power supply module	100-240VAC	PS261	TPS261**S
	24VDC	PS31	EX10*MPS31
CPU module	23.5k step, calendar, EEPROM, 2port com	PU215N	TPU215N*S
	PU215N + Ethernet	PU235N	TPU235N*S
	PU215N + Ethernet + TOSLINE-S20LP	PU245N	TPU245N*S

● Input/output module

DC/AC input	16-point 12-24VDC/AC, 8mA	DI31	EX10*MD131
DC input	32-point 24VDC, 5mA	DI32	EX10*MD132
	64-point 24VDC, 4mA	DI235	TDI235**S
AC input	16-point 100-120VAC, 7mA	IN51	EX10*MIN51
	16-point 200 ~ 240VAC, 6mA	IN61	EX10*MIN61
Relay output	12-point 240VAC/24VDC, 2A/point (MAX)	RO61	EX10*MRO61
Isolated relay output	8-point 240VAC/24VDC, 2A/point(MAX)	RO62	EX10*MRO62
Transisitor output	16-point 5-24VDC, 1A/point (MAX) Sink type	DO31	EX10*MDO31
	32-point 5-24VDC, 100mA/point Sink type	DO32	EX10*MDO32
	64-point 5-24VDC, 100mA/point Sink type	DO235	TDO235**S
	16-point 12-24VDC, 1A/point (MAX) Source type	DO233P	TDO233P**S
Triac	12-point 100-240VAC, 0.5A/point (MAX)	AC61	EX10*MAC61
Analogue input	4ch, 4-20mA / 1-5V, 8bits resolution	AI21	EX10*MAI21
	4ch, 4-20mA / 1-5V, 12bits resolution	AI22	EX10*MAI22
	4ch, 0-10V, 8bits resolution	AI31	EX10*MAI31
	4ch, -10 - +10V, 12bits resolution	AI32	EX10*MAI32
Analogue output	2ch, 4-20mA / 0-5V / 1-5V / 0-10V, 8bits resolution	AO31	EX10*MAO31
	2ch, 4-20mA / 1-5V, 12bits resolution	AO22	EX10*MAO22
	2ch, -10 - +10V, 12bits resolution	AO32	EX10*MAO32
Pulse input	1ch, 5 / 12V, 100kpps (MAX), 24bit counter	PI21	EX10*MPI21
Position control	A pulse output type (MAX, 200kpps)	MC11	EX10*MMC11
Serial interface	RS232C 1ch, 160wards*2	CF211	TCF211**S

Ordering Information

● Data transmission module

TOSLINE-S20	Coaxial cable	SN221	SSN221*MS
	Optical fiber	SN222A	SSN222AMS
TOSLINE-F10	Twisted pair : Master station	MS221	FMS221AM
	Twisted pair : Slave station	RS211	FRS211AM
TOSLINE-30	Twisted pair	LK11	EX10*MLK11
	Optical fiber	LK12	EX10*MLK12
DeviceNet	DeviceNet Scanner module	DN211	TDN211**S

● Cables and others

Expansion	0.3m	CAR3	EX10*CAR3
	0.5m	CAR5	EX10*CAR5
	0.7m	CAR7	EX10*CAR7
	1.5m	CS2RF	TCS2RF*CS
Slot cover	Cover for empty slot : one slot length	-	EX10*ABP1

● peripherals

Programming software (T-PDS)	MS-DOS vorsion (English)	MM33I1	TMM33I1SS
	Windows version (English)	MW33E1	TMW33E1SS
Handy programmer	attached 2m cable between programmer and PC	HP911	THP911**S
TOSLINE-S20 support software (S-LS)	MS-DOS vorsion (English)	MM23I	SMM23I*SS
Programmer cable (for T-PDS, S-LS)	IBM-PC or compatible	CJ905	TCJ905*CS
RS232C/RS485 converter	RS232C/RS485 converter for computer link	ADP-6237B	EX25PADP6237B

● Accessories

CPU Battery	ER6	-	EX25SER6
Fuses	For PS31	-	EX10*SFB20
	For PS261	-	TFU923*AS
	For DO31	-	EX10*SFA50
	For DO32	-	EX10*SFA20
	For AC61	-	EX10*SFC20

TOSHIBA

TOSHIBA CORPORATION
Industrial Equipment Department
1-1, Shibaura 1-chome, Minato-ku
Tokyo 105-8001, JAPAN
Tel: 03-3457-4900   Fax: 03-5444-9268