User's Manual

IMO Programmable Logic Controller
G6 Series

IMO Precision Controls Ltd

Chapter 1.   GENERAL

Chapter 2.   SYSTEM CONFIGURATION

Chapter 3.   GENERAL SPECIFICATION

Chapter 4.   CPU MODULE

Chapter 5.   BATTERY

Chapter. 6   USING THE USER PROGRAM IN FLASH MEMORY

Chapter. 7   DIGITAL INPUT AND OUTPUT MODULES

Chapter 8.   POWER SUPPLY MODULE

Chapter 9.   BASE BOARD

Chapter 10.   INSTALLATION AND WIRING

Chapter 11.   MAINTENANCE

Chapter 12.   TROUBLESHOOTING

Chapter 13.   Dedicated Cnet communication for GM6

APPENDICES

Chapter 2.   SYSTEM CONFIGURATION

The GLOFA-GM6 series has various modules suitable to configuration of the basic, computer link and network systems.

This chapter describes the configuration and features of each system.

2.1   Overall Configuration

The following shows the overall configuration of the GLOFA-GM6 series.

Battery

CPU Module

RS-232C Cable

GMW

GMWIN

Discket

Power Supply Module (GM6-PAF   )

Base board(GM6-B0   M)

Input Module (G6I-   )

Output Module (G6Q-   A)

Special Module (G6F-   )

Communication Module (G6L-   )
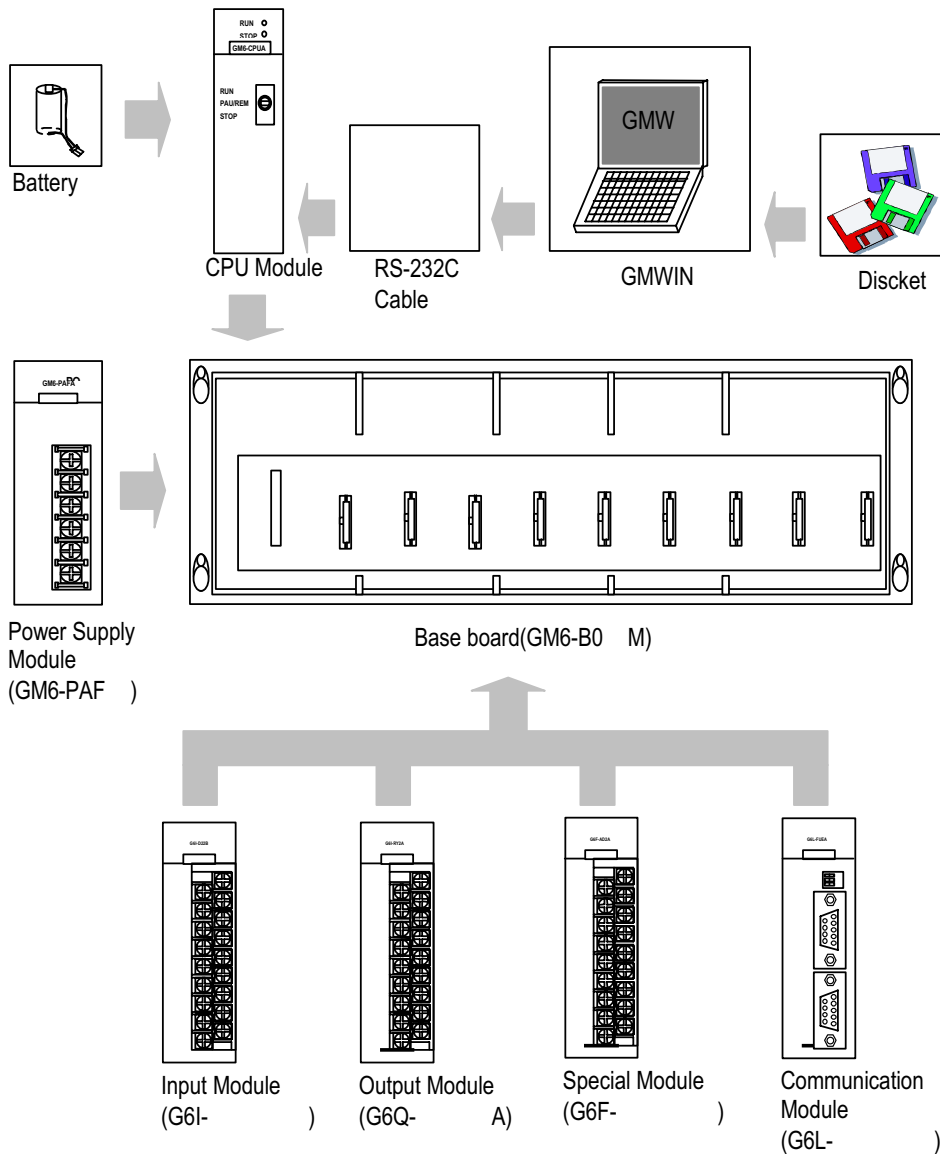
2.2   Product List

The following table shows product list of GLOFA-GM6 series.

2.2.1   GM6 series Configuration

Items	Models	Description	Remarks
CPU module	GM6-CPUA	• Maximum I/O points: 256	
Digital input module	G6I-D22A	• 16-point 24 VDC input module(current source & sink input)	
	G6I-D22B	• 16-point 24 VDC input module(current source input)	
	G6I-D24A	• 32-point 24 VDC input module(current source & sink input)	
	G6I-D24B	• 32-point 24 VDC input module(current source input)	
	G6I-A11A	• 8-point 110 VAC input module	
	G6I-A21A	• 8-point 220 VAC input module	
Digital output module	G6Q-RY2A	• 16-point relay output module(2A)	
	G6Q-TR2A	• 16-point transistor output module(0.5A, sink output)	
	G6Q-TR4A	• 32-point transistor output module(0.1A, sink output)	
	G6Q-SS1A	• 8-point triac output module(1A)	
Main base unit	GM6-B04M	• Up to 4 I/O modules can be mounted.	
	GM6-B06M	• Up to 6 I/O modules can be mounted.	
	GM6-B08M	• Up to 8 I/O modules can be mounted.	
Power supply module	GM6-PAFA	Free Voltage • 5 VDC : 2 A, 24 VDC : 0.3 A	AC100 ~ 240V
	GM6-PAFB	" • 5 VDC : 2 A • +15 VDC : 0.5 A, -15VDC : 0.2 A	
Special modules	A/D conversion module G6F-AD2A	• Voltage/current input : 4 channels • DC -10 to 10V / DC -20 to 20 mA	
	D/A conversion module G6F-DA1A	• Voltage/current input : 2 channels • DC -10 to 10V / DC -4 to 20 mA	
	High speed counter module G6F-HSCA	• Counting range: 0 to 16,777,215(24 bit binary) • 50 kHz, 1 channel	
Communication modules	Fnet I/F module G6L-FUEA	• For Fnet I/F • 1 Mbps base band • For twisted cable	
	Computer Link module G6L-CUEB	• RS-232C	
	G6L-CUEC	• RS422	
Others	Dust Proof Module GM6-DMMA	• Protect empty slot for dust	

2.3 System Configuration Types

System configuration is classified into 3 types that Basic system, Computer link system executing data communications between the CPU module and a computer by use of a computer link module(G6L-CUEB/C) and Network system controlling the PLC and remote I/O modules.

2.3.1 Basic System

The following describes basic system.

Example of System configuration	Slot number, POWER, CPU table (see figure)
	(The above figure shows the configuration where 16-input/output modules are loaded.)
Maximum number of Input/Output modules	8 modules
Maximum number of Input/Output points	• 16-point module mounted: 128 points • 32-point module mounted: 256 points
Configuration units	CPU module — GM6-CPUA Power Supply module — GM6-PAFA, GM6-PAFB Basic Base Unit — GM6-B04/06/08M I/O module — G6I-□□□□ / G6Q-□□□□
I/O number allocation	64 points are allocated to each slot in a base board whatever it is empty or not. There's no limitation for the location and the number of special modules on base board. Special modules do not have fixed I/O numbers while a fixed I/O number is allocated to a digital I/O module. A dedicated function block controls a special module and memory is allocated automatically.

2.3.2 Computer Link System

Computer Link System communicates data between the CPU module and peripheral devices like a computer or a printer by use of RS-232C and RS-422(or RS-485)interface of the computer link module.

The G6L-CUEB or G6L-CUEC are the computer link module for GM6 series. For details of computer link module, refer to related User's Manual.

2.3.3   Network System

The Network system adapted in the GLOFA series a Fnet system that satisfies the IEC/ISA field bus specifications. Fnet system as a network system is used for data communications between CPU modules and control of remote I/O modules so that distribution of control and concentration of supervision could be easy. For details, refer to Fnet system user's manual.

Chapter 3. GENERAL SPECIFICATION

3.1 General specifications

The following shows the general specifications of the GLOFA-GM series.

No	Item	Specifications				References
1	Operating ambient temperature	0 ~ 55 °C				
2	Storage ambient temperature	-25 ~ +75 °C				
3	Operating ambient humidity	5 ~ 95%RH, non-condensing.				
4	Storage ambient humidity	5 ~ 95%RH, non-condensing.				
5	Vibration	Occasional vibration				IEC 1131-2
		Frequency	Acceleration	Amplitude	Sweep count	
		10≤ f<57 Hz	-	0.075 mm	10 times per axis, on X,Y, Z axis	
		57≤f≤150 Hz	9.8 m/s^2{1 G}	-		
		Continuous vibration				
		Frequency	Acceleration	Amplitude		
		10≤f <57 Hz	-	0.035 mm		
		57≤f≤150 Hz	4.9 m/s^2 {0.5G}	-		
6	Shocks	Maximum shock acceleration: 147 m/s^2{15G} Duration time: 11 ms Pulse wave: half sine pulse (3 shocks per axis, on X,Y,Z axis)				IEC 1131-2
7	Noise Immunity	Square wave Impulse Noise	± 1,500 V			
		Electronic discharge	Voltage : 4 kV			IEC 1131-2, IEC 801-3
		Radiated electromagnetic field noise	27 ~ 500 MHz, 10 V/m			IEC 1131-2, IEC 801-3
		Fast transient/burst noise	Item	Power supply	Digital I/O (>24V)	Digital I/O (<24V) Analog I/O interface	IEC 1131-2, IEC 801-4
			Voltage	2 kV	1 kV	0.25 kV	
8	Operating ambience	Free of corrosive gases and excessive dust.				IEC 1131-2
9	Altitude	2,000 m or less				
10	Pollution	2				
11	Cooling method	Air-cooling				

REMARK

1) IEC(International Electromechanical Commission) :   An international   civilian   institute who   establishes international standards in area of   electric's and electronics.

2)Pollution : An indicator which indicates pollution degree which determine insulation performance of equipment. Pollution 2 means that non-conductive pollution usually occurs but temporal conduction occurs with condensing

Chapter 4. CPU MODULE

4.1   Performance specifications

The following shows the general specifications of the GLOFA-GM series.

Items		Specifications	Remarks
Operation method		Cyclic operation of stored program, Interrupt task operation	
I/O control method		Scan synchronized batch processing method(Refresh method)	
Programming language		Ladder Diagram(LD) Instruction List(IL) Sequential Function Chart(SFC)	
Number of instructions	Operator	LD : 13, IL : 21	
	Basic function	194	
	Basic function block	11	
	Special function block	Each special module have their own special function blocks	
Processing speed	Operator	Refer to Appendix 3.	
	Basic function		
	Basic function block		
Programming memory capacity		68 k bytes(17 k steps)	
I/O points		256 points	
Data memory	Direct variable area	2 to 8 k bytes	
	Symbolic variable area	30 k bytes – Direct variable area	
Timer		No limitations in points. Time range : 0.01 to 4294967.29 sec(1193 hours)	1 point occupies 20 bytes of symbolic variable area.
Counter		No limitations in points Counting range: -32768 to +32767	1 point occupies 8 bytes of symbolic variable area.
Program types	Numbers of program blocks	100	
	Initialization programs	1 (_INIT)	
	Task Programs	Time driven tasks	8	Total : 8 (The type of task is variable, however, total numbers of tasks is 8.)
		External interrupt tasks	8	
		Internal task	8	
Operation modes		RUN, STOP, PAUSE and DEBUG	
Restart modes		Cold, Warm	
Self-diagnostic functions		Watch dog timer, Memory error detection, I/O error detection, Battery error detection, Power supply error detection, etc.	
Data protection method at power failure		Set to 'Retain' variables at data declaration.	
Internal current consumption		0.15A	
Weight		0.11 Kg	

4.2 Operation Processing

4.2.1 Operation Processing Method

1) Cyclic operation

A PLC program is sequentially executed from the first step to the last step, which is called scan.

This sequential processing is called cyclic operation. Cyclic operation of the PLC continues as long as conditions do not change for interrupt processing during program execution.

This processing is classified into the following stages.

Stages	Processing
Operation Start	-
Initialization	• Stage for the start of a scan processing. it is executed only one time when the power is applied or reset is executed. It executes the following processing. ‣ I/O modules reset ‣ Execution of self-diagnosis ‣ Data clear ‣ I/O module address allocation or type registration
Input image area refresh	• Input module conditions are read and stored into the input image area before operation processing of a program.
Program operation processing Program start ~ Program end	• Program is sequentially executed from the first step to the last step
Output image area refresh	• The contents stored in the output image area is output to output modules when operation processing of a program is finished.
END processing	• Stage for return processing after the CPU module has finished 1 scan. The following processing are executed. ‣ Self-diagnosis ‣ Change of the present values of timer and counter, etc. ‣ Processing data communications between computer link module and communications module. ‣ Checking the switch for mode setting.

2) Time driven interrupt operation method

In time driven interrupt operation method, operations are processed not repeatedly but at every pre-set interval. Interval, in the GM6 CPU module, can be set to between 0.01 to 4294967.29 sec. This operation is used to process operation with a constant cycle.

3) Event driven interrupt operation method

If a situation occurs which is requested to be urgently processed during execution of a PLC program, this operation method processes immediately the operation which corresponds to interrupt program.   The signal which informs the CPU module of those urgent conditions is called interrupt signal. The GM6 CPU module has two kind of interrupt operation methods, which are internal and external interrupt signal methods.

4.2.2   Operation processing at momentary power failure occurrence

The CPU module detects any momentary power failure when the input line voltage to the power supply module falls down below the defined value.

When the CPU module detects any momentary power failure, the following operations will be executed.

1) Momentary power failure within 20 ms

input power

Momentary power failure within 20 ms
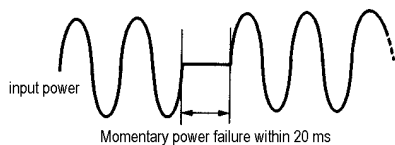
(1) The operation processing is stopped with the output retained.

(2) The operation processing is resumed when normal status is restored.

(3) The output voltage of the power supply module retains the defined value.

(4) The watch dog timer(WDT) keeps timing and interrupt timing normally while the operations is at a stop.

2) Momentary power failure exceeding 20 ms

input power

Momentary power failure exceeding 20 ms
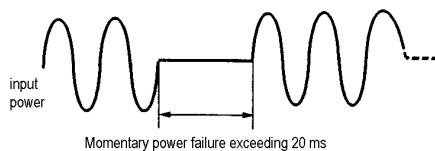
• The re-start processing is executed as the power is applied.

REMARK

1) Momentary power failure

The PLC defining power failure is a state that the voltage of power has been lowered outside the allowable variation range of it. The momentary power failure is a power failure of short interval(several to tens ms).

4.2.3 Scan Time

The processing time from a 0 step to the next 0 step is called scan time.

1) Expression for scan time

Scan time is the addition value of the processing time of scan program that the user has written, of the task program processing time   and the PLC internal processing time.

(1) <u>Scan time = Scan program processing time + Task program processing time + PLC internal processing time</u>

- Scan program processing time = The processing time used to process a user program that is not specified to a task program.

- Task program processing time = Total of the processing times of task programs executed during one scan.

- PLC internal processing time = Self-diagnosis time + I/O refresh time + Internal data processing time + Communications service processing time

(2) Scan time differs in accordance with the execution or non-execution of task programs and communications processing, etc.

2) Flag

(1) Scan time is stored in the following system flag area.

- _SCAN_MAX : Maximum scan time (unit : 1 ms)

- _SCAN_MIN : Minimum scan time (unit : 1 ms)

- _SCAN_CUR : Current scan time (unit : 1 ms)

4.2.4 Scan Watchdog Timer

1) Watchdog timer is used to detect a delay of abnormal operation of sequence program.
   (Watchdog time is set in menu of basic parameter of GMWIN.)

2) When watchdog timer detects an exceeding of preset watchdog time, the operation of PLC is stopped
   Immediately and all output is off.

3) If an exceeding of preset watchdog time is expected in sequence program, use ' WDT_RST' function.
   'WDT_RST' function make elapsed watchdog time as zero.

4) In order to clear watchdog error, using manual reset switch, restarting the PLC and mode change to STOP
   mode are available.

REMARK

Setting range of watchdog   :   1 ~ 65,535ms(1ms base)

4.2.5 Timer Processing

The CPU module timer is on incremental timer which increase its present value according to the measuring time. Three types of On Delay Timer(TON), Off Delay Timer(TOF) and Pulse Timer(TP)   are available.

Its measuring range is 0.001 to 4,294,967,295 sec (1,193 hours) by 1 ms. For details, refer to ' GLOFA-GM Programming'.
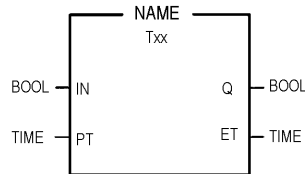
1)   **On Delay Timer Process Time Change and Contact On/Off)**

Timer Process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the Timer output contact turns on.

On Delay Timer Timing Diagram is shown as below.
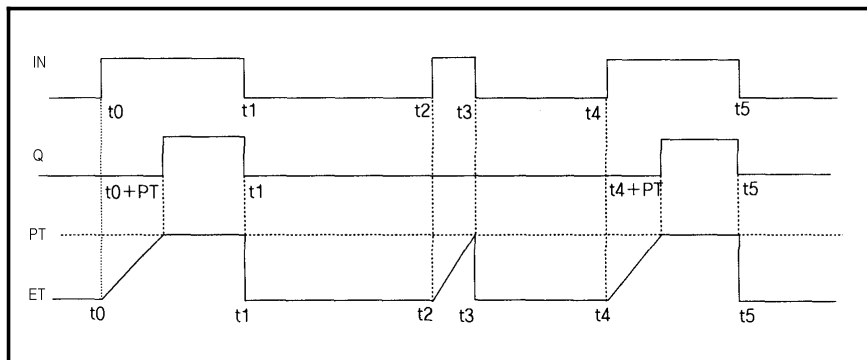
2)   **Off Delay Timer Process Time Change and Contact On/Off**

• If input condition turns on, timer output contact(Q) turns on. If input condition turns off, timer process time change starts.

• The process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the contact (Q) turns off. The following diagram shows Off Delay Timer Timing.
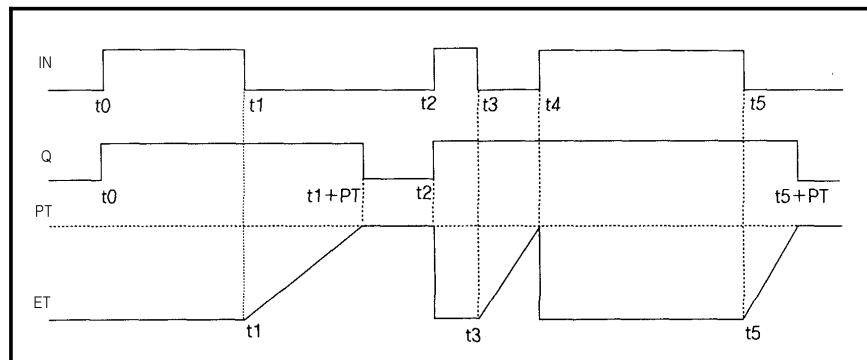
3)   Pulse Timer Process Time Change and Contact On/Off

If input condition turns on, output contact (Q) turns on.

The process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the contact (Q) turns off.

The contact turns off after the setting time regardless of input condition off status.

The following diagram shows pulse timer timing.
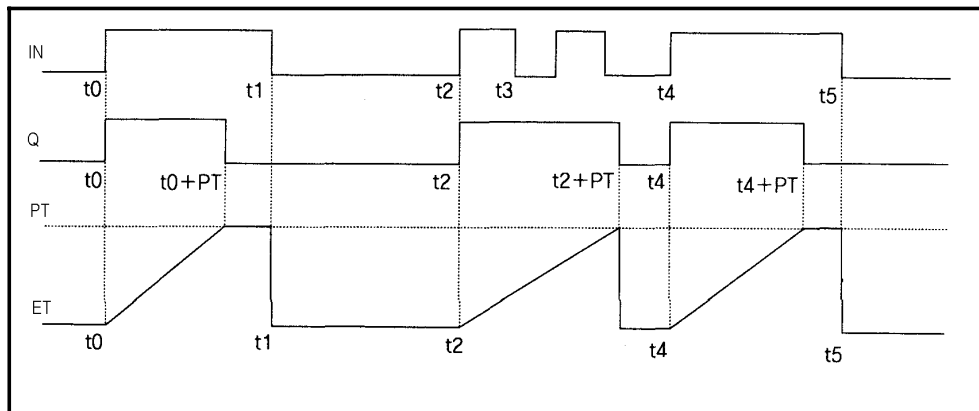
4)   Timer error

The maximum timer error is ' 1 scan time + time from the start of scan to execution of the timer function block".

4.2.6 Counter Processing

The CPU module counter increment/decrement the present counting value by the detection of rising edge(off→on) of input signal. Three types of counter are increment counter, Decrement counter and Increment-Decrement Counter. For details, refer to 'GLOFA – GM Programming'.

- The Increment counter is a counter which increment the present counting value
- The Decrement counter is a counter which decrement the present counting value
- The Increment-Decrement counter is a counter which compares the counting values of two input conditions.

1)　Counter Present Value Change and Contact On/Off

　　(1)　Increment Counter

- It should have Input condition (CU), reset condition (R) and setting value (PV).

```
              ┌──── NAME ────┐
              │      CTU      │
              │              │
BOOL  ────►   │ CU        Q  │  ─── BOOL
              │              │
BOOL  ───     │ R            │
              │              │
INT   ───     │ PV       CV  │  ─── INT
              └──────────────┘
```
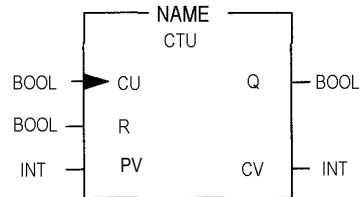
- If the counting value (CV) increments and reaches the setting value(PV) the output contact (Q) turns on.

　When the reset signal is turn on, the counting value is set to '0' and the output contact (Q) turns off.

　　(2)　Decrement Counter

- It should have input condition (CD), load (LD) and setting value (PV).

```
              ┌──── NAME ────┐
              │      CTD      │
              │              │
BOOL  ────►   │ CD        Q  │  ─── BOOL
              │              │
BOOL  ───     │ LD           │
              │              │
INT   ───     │ PV       CV  │  ─── INT
              └──────────────┘
```
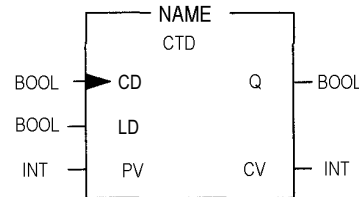
- If the counting value (CV) decrements and reaches '0', the output contact (Q) turns on.

　If the load(LD) signal is turned on, the counting value is set to the setting value and the output contact (Q) turns off.

(3)   Increment/Decrement Counter

- It should have Increment input condition (CU), Decrement input condition (CD), load (LD) and setting value (PV).

```
              ┌──── NAME ────┐
              │     CTUD     │
BOOL─┤ CU          QU ├─ BOOL
BOOL─┤ CD          QD ├─ BOOL
BOOL─┤ R               │
BOOL─┤ LD              │
 INT─┤ PV          CV ├─ INT
```
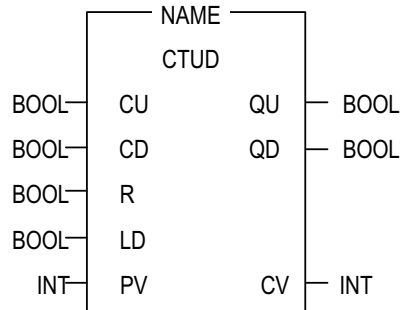
- If reset signal(R) turns on, counting value (CV) is set to '0'.
- If load signal(LD) turns on, counting value is set to setting value(PV).
- It is increased by 1at the rising edge of increment input(CU) and decreased by 1 at the edge of decrement input(CD). If counting value(CV) is equal or larger than setting value(PV),QU will be on, and if counting value(CV) is equal or less than setting value(PV),QD will be on.

2)   Counting speed

- The counting speed is decided by scan time and it will be counted when on time or off time of input condition is larger than each scan time.

Max. Counting speed (Cmax.) = n / 100   1 / ts   [pps]   [n : Duty(%),   ts : scan time(s)]

- Duty is percent of on time / off time.

```
          on
      ┌──────┐        ┌──────┐
      │      │   off  │      │
──────┘      └────────┘      └──────

      ←─ T1 ─→          ←─ T2 ─→
```
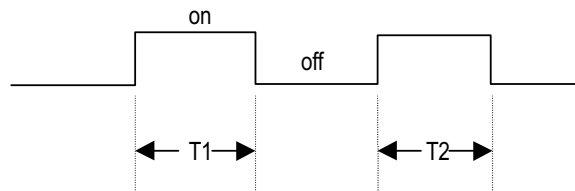
T1   T2 : n = T1 / (T1+T2)   100 [%]
T1   T2 : n = T2 / (T1+T2)   100 [%]

4.3 Program

4.3.1 Program Configuration

A program consists of all of the function elements that is needed to execute a particular control. It is to be stored in the internal RAM of the CPU module or the flash memory of the memory module.

The function elements are classified as below.

Function Elements	Processing Operation
Initialization program	• Executed when the power is applied or the CPU operation is transited to the RUN mode. • Executes the initial/fixes data setting for execution of scan program and the initialization of peripheral devices on special modules.
Scan program	• Processes the constantly repeated signals which are executed every scan.
Time driven task program	• When the following time conditional processing is required the program is executed complying with the time interval setting. ▶ In case that the processing need a shorter interval than that of average one scan processing time. ▶ In case that the processing need a longer interval than that of average one scan processing time. ▶ In case that the processing should be executed by the specified time interval.
Event driven task program	• A shorter processing is executed for internal or external interrupt.

4.3.2 Program Execution Procedure

The followings explain the program execution procedure when the power is applied or the mode setting switch of CPU module is in the RUN status.

Program operation processing is executed as the procedure given below

```
       ┌────────────────────┐
       │  Operation start   │
       └─────────┬──────────┘
                 │
       ┌─────────┴──────────┐         • Executed when the power has been applied
       │                    │◄·········  or the CPU operation is in the Run mode
       │ Initialization     │
       │ program            │         • Restart operation is executed complying with
       └─────────┬──────────┘            the initialization task(_INIT, HINIT)
                 │                *1
       ┌─────────┴──────────┐   ┌────────────────────┐
   ┌──►│                    │   │ External task program│
   │   │  Scan program      │◄─►│ Time driven task    │◄·······  Executed only when the condition
   │   │                    │   │ program             │            has been satisfied.
   │   └─────────┬──────────┘   └────────────────────┘
   │             │
   │             │              ┌────────────────────┐
   │             │              │                    │◄·······  Executed only when the condition
   │             ◄─────────────►│ Internal task program│          has been satisfied.
   │             │              └────────────────────┘
   │   ┌─────────┴──────────┐
   │   │  END processing    │
   │   └─────────┬──────────┘
   └─────────────┘
```

┌───┐
│ **REMARK**   │
│ 1)   *1 : In the GLOFA PLC, the time driven interrupt task programs and event driven interrupt task │
│   programs are called task program.   │
│   Event driven programs are classified into single task(internal interrupt) or interrupt task (external │
│   interrupt) according to the S/W and H/W interrupt signaling method. │
└───┘

1) Initialization program

(1) Function

- The Initialization program initializes the program to execute scan and task programs.
- The initialization can be executed with the restart mode which has been specified for program.

(2) Restart mode execution conditions

- The initialization tasks can be specified as below complying with the purpose of the initialization task.
  ‣ Program for Cold/ Worm restart started by the _INIT task

(3) Cold/ Warm Restart program

- The initialization program specified to _INIT task is executed with cold or warm restart mode when the operation starts.
- This initialization program executes the operations repeatedly until the setting conditions are satisfied(that is, until the Flag_INIT_DONE in the initialization program turns on). However, the I/O refresh is still executed.

(4) Flag

- _INIT_RUN flag is on during executing the initialization program.

2) Scan program

(1) Function

- In order to process signals which repeats constantly, the program executes its sequential operation repeatedly from the first step to the end step.
- If the interrupt task execution condition has been satisfied by a time driven task or event driven task module during scan program execution, the program that is under execution will be temporary stopped and the corresponding task program will be executed.
- If the scan program has been completely executed, the single task(internal interrupt) execution condition will be checked and the corresponding task program will be executed.

(2) configuration

- Up to 100 scan programs can be used.
  (If task programs are used, the usable number is reduced as many as that of the used task programs)
- Program has been not specified to initialization or task program when writing that program, it will be automatically specified to scan program.
- Scan program has lowest execution priority and the priorities of scan program are determined their registration sequence in the GMWIN screen when writing those programs.

3) Task program

(1) Function

- In order to process internal/ external signal which occurs periodically or non-periodically, the task program temporarily stop the operation of scan program and processes first the corresponding function

(2) Types

- Task programs are classified into the three types as below.
  ‣ Time driven task program : Up to 8 programs are applicable
  ‣ Single (internal) task program : Up to 8 programs are applicable
  ‣ Interrupt (external) task program : Up to 8 programs are applicable
- Time driven task program
  ‣ The program is executed by the time internal set before
- Single (internal) task program
  ‣ The corresponding program will be executed at the rising edge and on state of internal contact in the program.
  ‣ The detection of the start up condition will be executed after the scan program has been processed.
- Interrupt (external) task program
  ‣ The program is executed according to the external signal a input to the interrupt module

REMARK
1) Refer to section 4.3.3 task for details of task program.

4.3.3 Task

The followings explain the program structure and tasks of the GMWIN, that is, the GLOFA-GM programming S/W, in order to give an understanding of the task function
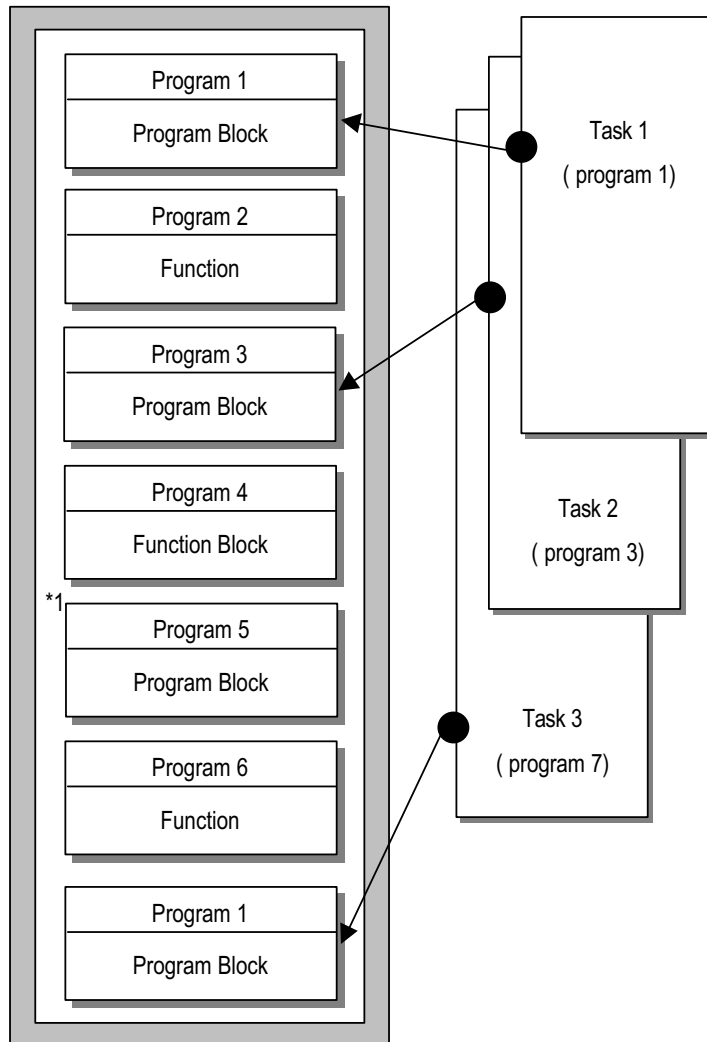
REMARK

1) A task executes the some function as the control panel which are used to execute programs. Each task consists of one or more program blocks in the three types of program. Those programs are called task programs A program to which a task has not been specified as marked with '*1' will be automatically specified to scan program

1) Task types and functions

The following table show the types and functions of tasks

Type Specifications	Time driven task	External interrupt task	Internal interrupt task
Number [1]	8	8	8
Start up condition	Time driven interrupt (up to 4,294,967.29sec by the 10msec)	At the rising edge of input contact on the designated slot	The rising edge or on state of the BOOL variable data which has been specified of buffer data
Detection and execution	Executed periodically as setting time	Immediately executed when an edge occurs in the interrupt module	Executed with edge detection after scan program has been finished
Detection delay time	Up to 1msec delay	Maximum 1msec delay + Input module delay(Within 3msec	Delayed for the same time as maximum scan time
Execution priority	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7	Level 0 to 7

 * 1) Up to 8 task programs are available.

2) Task program processing Method

The following explains the common processing method and instructions for task programs

(1) Task program characteristics

- The task program will be executed when a execution condition is satisfied while the scan program is repeatedly processed at every scan. Be sure to consider that point when writing a task program

- For example, if a timer and a counter have been used in a 10 sec cycle time driven task program, the timer can occur up to 10 sec error and an input which has been changed within 10 sec will not be counted because the counter checks its input status every 10 sec

(2) Execution priority

- The higher priority task program will be executed firstly.

- If a newly invoked task has higher priority than that of existing tasks which are under execution, they are temporary stopped and task has higher priority will be executed.

- When determining the priority of a task program, consider the characteristics, importance and urgency of the program

(3) Processing delay time

The following factors influence on the processing delay of task program, consider the characteristics, importance and urgency of the program

- Task detection delay (Refer to the detailed description of each task)

- Execution delay due to the execution of prior task programs

- Delay due to the execution of higher priority task programs while executing task programs

(4) Relationship of task program to initialization or scan program

- User defined tasks will not start while the initialization task program is being executed.

- As scan program has the lowest priority, if a task is invoked the scan program will be stopped and the task programs will be processed prior to them. Therefore, if tasks are invoked many times or concentrated sometimes the scan time may be extended abnormally. Be cautious when setting task conditions.

(5) Protection of the programs under execution from task programs

- If problems can be occur in case that program lose its execution continuousness by the task programs which have higher proprieties, the execution of task programs can be partly perverted For program protection, use the DI function(Task program start-up disable) or EI function(task program start-up enable)

3) Time driven task program processing method

The followings explain the processing method of a task program when its task condition(start-up condition) has been set to be driven by time.

(1) Settings that have to be set for the task

- Set the task execution cycle and its priority which are used as start-up conditions for the task programs to be executed. Priority number will be task number.

(2) Time driven task processing

- The corresponding time driven interrupt task program will be executed every setting time internal (execution cycle).

(3) Precautions for using the time driven task program

- While a time driven task program is being executed or ready for its execution, if a same priority task program has been invoked to be executed the newly invoked task will be ignored, the representative task collision warning flag (_TASK_ERR) will be set to ON, the detailed system error flag(_TC_BMAP[n] will be set to ON at its corresponding location and occurrence time of the time driven tasks whose execution requests have been ignored will be written at its corresponding location of the flag _TC_CNT[n].

- The timer that invokes the execution request for time driven task programs will be incremented only when the operation mode is in the RUN mode

  If the RUN mode has been changed into the PAUSE mode while operating with the RUN mode, and then the operation mode has been changed again into the RUN mode, the operation time spent with the PAUSE mode will be ignored.

- When setting the execution cycle for a time driven task program, be cautious that execution requests for many time driven task programs can occur. If four time driven task programs of cycle 2, 4, 10 and 20sec are used, four execution requests will occur every 20 sec and scan time can be momentarily extended.

4) External contact program processing method

The following explains in the case that the task(start-up condition) of a task program has been set to an external input signal.

(1) Settings that have to be set for the input module

- A contact of input module can be used as interrupt input.

(2) Settings that have to be set for the task

- Set the contact No. of input module and priority for the task that will be used as start-up conditions of the task programs to be executed. Priority will be the task number.

(3) External contact task processing

- The CPU module checks the occurrence of interrupt input every 1ms and executes the task program which are designated by the contact at which the signal has been occurred.

(4) Precautions for using an external contact task.

- While a task program which are designated by an input module having interrupt input, contact is being executed or ready for its execution, if an execution request of a task program has been occurred to the same input contact then the newly invoked task will be ignored, the representative task collision warning flag(_TASK_ERR) will be set to ON, the detailed system error flag(_TC_BAMP[n]) will be set to ON at its corresponding location and the occurrence time of the external task whose execution request has been congested.

- Execution request for a task program can be accepted only when the operation mode is in the RUN mode. That is, if the RUN mode has been changed into the PAUSE mode while operating with the RUN mode and the operation mode has been changed into the RUN mode again, all execution requests occurred during the operation with the PAUSE mode will be ignored.

5) Internal task program processing method

The following explains the processing method when the task (start-up condition) of a task program has been set to the contact of direct variable area(I, Q or M) or automatic variable area.

(1) Settings that have to be set for the task.

- Set the contact No. of input module and priority for the task that will be used as start-up conditions of the task programs to be executed. Priority will be the task number.

(2) Internal contact task processing

- After the execution of scan program has been completed in the CPU module, the internal contacts that are the start-up conditions of the task program will be checked and the internal task programs where rising edge or on state has been occurred will be executed in accordance with its parameter.

(3) Precautions when using an internal task program.

- The internal task program is executed when scan program has finished its execution. Therefore, though the execution condition for the internal task program has been invoked in the scan program or task program(time driven, external) the task (start-up condition) will not be immediately executed but will be executed when scan program has finished its execution.

• If execution of an internal task program is requested, the execution conditions will be checked when scan program has finished its execution. Therefore, if an internal task execution conditions, during 'one' scan, has been occurred and disappeared (if the specified contact has been turned from OFF to ON, and then from ON to OFF) by scan program or (time driven or external) task program the task will not be executed as the execution condition can not be detected at the time that execution conditions are being checked.

6) Task processing at momentary power failure

• In case of the power failure of 20 ms or less, the ready tasks before the power failure will be executed, a time driven task will be invoked with calculation of the power failure time, and time driven tasks invoked repeatedly before the power failure will be ignored.

7) Examination on task program

After writing down a task program, be sure to examine the following items.

(1) Task setting has been correctly done?

If tasks are invoked more frequently than necessary or several tasks are invoked simultaneously within one scan, the scan time become longer and irregular. In case that the task setting cannot be changed, check the maximum scan time.

(2) Task priorities are properly arranged?

The lower priority tasks still may not be processed after its time due to delay by higher priority tasks. In some cases, if the prior tasks have been delayed and next task occurs task collision can occur. Set the priority with due consideration of items such as urgency and execution time of a task.

(3) Task programs are written as shortly as possible?

If execution time of a task program is long, the scan time may become longer and irregular and also collision of task programs may occur. Therefore, write task programs as shortly as possible.

(4) Protection of lower priority programs against higher priority program isn't needed during execution of those programs.

If the priority of a task program (or a scan program) has been set to lower priority and other tasks must not interrupt during its execution, use the function 'DI' and 'EI' to protect the program partly. When processing global variables used commonly in other programs, special modules or communications modules, problems can occur.

REMARK

1) For examination on processing speed of scan program and task program, refer to the 'Scan time Calculation Example in the Section 4.2.3 'Scan Time'.

---

8) Example of program configuration and processing

When the task and program have been registered as below,

- Task registration : T_SLOW (interval : T#10ms, priority : = 0)

　　　　　　　　　　　　　PROC_1 (single : %MX0, priority : = 3)

　　　　　　　　　　　　　E_INT1 (interrupt : 0, priority : = 0)

- program registration : program → P0

　　　　　　　　　　　　　program → P1 with the task T_SLOW

　　　　　　　　　　　　　program → P2 with the task PROC_1

　　　　　　　　　　　　　program → P3 with the task E_INT1

If program execution time is equal to external interrupt occurrence time :

- Execution time for each program : P0 = 17 ms, P1 = 2 ms, P2 = 7 ms, P3 = 2 ms
- Interrupt E_INT occurrence time : Occurred at the 6, 7, 20 ms after the operation started.
- PROC_1 : Invoked during execution of scan program

Program execution is shown as below.
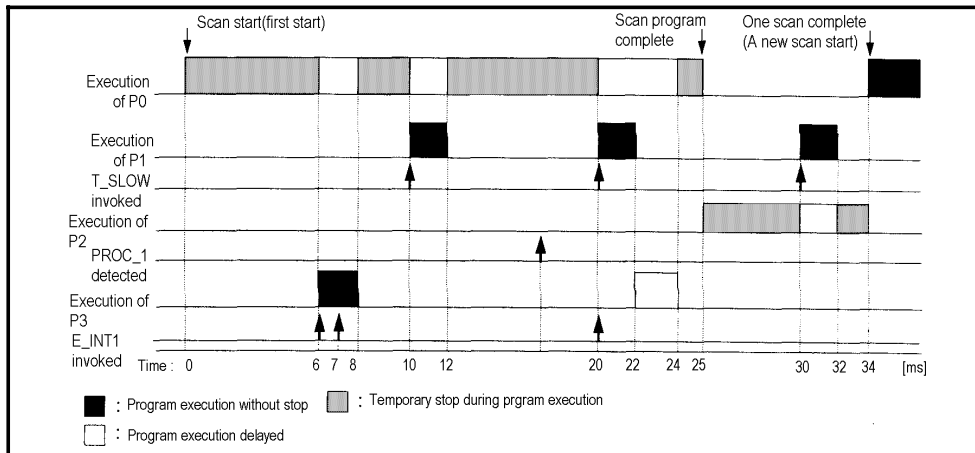
- Processing with time

　0 [ms] : Scan starts and the scan program P0 starts its execution.

　0 to 6 [ms] : The program P0 is being executed.

　6 to 8 [ms] : Execution request for P3 is input, and P0 is stopped and P3 is executed. Execution request for
　　　　　　　P1 by E_INT1 at the 7 [ms] is ignored as the P2 is being executed.

　8 to 10 [ms] : P3 finishes its execution and the P0 stopped continues its execution.

　10 to 12 [ms] : P0 is stopped and P1 is executed due to execution request for P1.

　12 to 20 [ms] : P2 finishes its execution and the P0 stopped continues its execution.

　20 [ms] : Execution requests for P1 and P3 are simultaneously exist, but the higher priority P1 is executed
　　　　　　and P3 is ready for its execution.

　20 to 22 [ms] : P0 is stopped and P1 is executed.

　22 to 24 [ms] : P1 finishes its execution and the higher priority P3 is executed before P0.

　24 to 25 [ms] : P3 finishes its execution and the P0 stopped completes its execution.

　25 [ms] : Execution request for P2 is checked at the finish time of the scan program (P0) and P2 is executed.

　25 to 30 [ms] : The program P2 is executed.

　30 to 32 [ms] : Execution request for P1 is input and P2 is stopped and P1 finishes its execution.

　32 to 34 [ms] : P1 finishes its execution and the P2 stopped finishes its execution.

　34 [ms] : A new scan starts. (P0 starts its execution.)

4.3.4 Error Handling

1) Error Classification

Errors occur due to various causes such as PLC system defect, system configuration fault or abnormal operation result. Errors are classified into fatal error mode, which stops system operation for system stability, and ordinary error mode, which continues system operation with informing the user of its error warning.

The main factors that occurs the PLC system error are given as followings.

- PLC hardware defect
- System configuration error
- Operation error during execution of the user programs
- External device malfunction

2) Operation mode at error occurrence

In case of error occurrence, the PLC system write the error contents the corresponding flags and stops or continues its operation complying with its operation mode.

(1) PLC hardware defect

The system enters into the STOP state if a fatal error such as the CPU module defect has occurred, and continues its operation if an ordinary error such as battery error has occurred.

(2) System configuration error

This error occurs when the PLC hardware configuration differs from the configuration defined in the software. The system enter into the STOP state.

(3) Operation error during execution of the user programs

If the numeric operation error of these errors occurs during execution of the user programs, its contents are marked on the error flags and the system continues its operation. If operation time overruns the watch dog time or I/O modules loaded are not normally controlled, the system enters into the STOP state.

(4) External device malfunction

The PLC user program detects malfunctions of external devices. If a fatal error is detected the system enters into the STOP state, and if an ordinary error is detected the system continues its operation.

REMARK

1) In occurrence of a fatal error the state is to be stored in the representative system error flags, and an ordinary error in the representative system warning flags.
2) For details of flags, refer to Appendix 2. Flag List.

4.3.5 Precautions when using special modules

This system offers convenience and high performance in using special modules compared with the existing methods. Therefore, take some precautions when composing the system. Check the system after the following items have been thoroughly understood.

1) Special module programming

(1) Special function block is offered for each special module to make programs concise and to prevent errors in writing down the user program.

(2) Function blocks are largely of two types. 'Initialization' function block for initializing special modules and 'control' function block for control of the operations of special modules. Function block functions as an interface between the user program data and the special modules. As it includes the function that watches the operation status of special modules and indicates the error status, other separate error detection program does not have to be written.

(For detailed description of function block, refer to the User's Manuals of special modules and GLOFA-GM instructions.)

2) Special Module Initialization

This means to define the operations of a special module. It is done with 'initialization' function block. Generally, it specifies the data range to used channel, resolution or filtering method, etc. It defines the hardware characteristics and only one time execution at system start is sufficient.

REMARK

1) As the initialization should be finished before the scan program starts its execution, its program should be written in the restart program (initialization task program).

---

3) Control of special modules

In control the operations of special modules, write the program using function blocks which correspond to the operations that have to be controlled. These function blocks can locate at any place within the program.

REMARK

1) If a power failure occurs in the base unit where special units are loaded, special modules data are removed. Therefore, data should be newly written down in the program.

---

4) Restart Program Example

(1) System Configuration

The followings give an example for writing the initialization program of the system where a special module has been loaded onto its basic base unit shown as below figure.

The followings describe an example for writing the 'cold/warm restart program' and 'scan program' for the scan program where the 'D/A 02' outputs data every scan and the 'D/A 03' outputs data only when the data has been changed.

P O W E R	C P U	D C 3 2	A / D 0 1	D / A 0 2	D / A 0 3	D C 3 2	R Y 3 2	

DC32 : 32-point DC input module
A/D : A/D conversion module
D/A : D/A conversion module
RY32 : 32-point relay output module
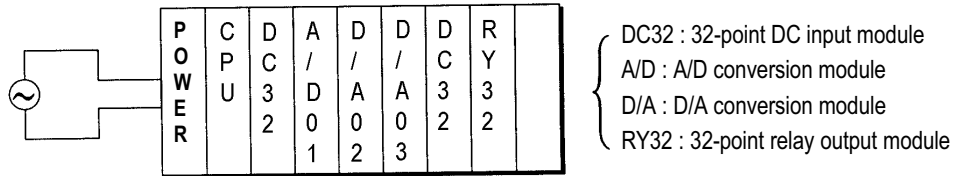
- As cold/warm restart makes the whole system restart, the 'cold/warm restart program' consists of only initialization program of special module.

(2) program

- Project Configuration : Restart.prj

● Program : cw_rst.src (cold/warm restart initialization program)

Comment ── Initialization of special module at cold/ warm restart, which are loaded onto the basic and extension base units.

Comment ── #### The module A/D 01 intialization ####

Line 2

```
                              ┌─ AD01_INI ─┐
Line 3   ─────────────────────│ REQ    DONE│
                              │            │
Line 4              0 ─┤ BASE    STAT├─ AD01_STAT
         Specifying the loaded Base No          Indicating the error status during initialization
                                                function block execution
Line 5              1 ─┤ SLOT     ACT├─ AD01_ACT
         Specifying the loaded Slot No          Indicating the run channels during initialization
Line 6         AD01_CH ─┤ CH          │         function block execution
         Specifying the used channel No
Line 7         AD01_DT ─┤ DATA        │
         Specifying the data type       TYPE
Line 8         AD01_FE ─┤ FILT        │
         Filter enable/disble specification _ EN
Line 9         AD01_FV ─┤ FILT        │
         Setting the filter value         _VAL
Line 10        AD01_AE ─┤ AVG_        │
         Average processing enable        EN
Line 11  /disable specification AD01_AS ─┤ AVG_│
         Specifying count/ time averaging SEL
Line 12        AD01_NT ─┤ NUM/        │
         Setting count/ time average value TIME
                              └────────────┘
```

Comment ── #### The module D/A 02 intialization ####

```
                              ┌─ DA02_INI ─┐
                              │   DA4INI   │
Line 15  ─────────────────────│ REQ    DONE│
                              │            │
Line 16             0 ─┤ BASE    STAT├─ DA02_STAT
         Specifying the loaded base No          Indicating the error status during initialization
                                                function block execution
Line 17             2 ─┤ SLOT     ACT├─ DA02_ACT
         Specifying the loaded slot No           Indicating the run channels during initialization
Line 18          DA_CH ─┤ CH          │         function block execution
         Specifying the used channel
Line 19        DA02_DT ─┤ DATA        │
         Specifying the data type       TYPE
Line 20  Specifying the D/A conversion DA02_SE ─┤ SEL│
         module output status when the
         CPU modules is in the STOP state
                              └────────────┘
```

Comment ── #### The module D/A 03 intialization ####

```
                              ┌─ DA03_INI ─┐
                              │   DA4INI   │
Line 22  ─────────────────────│ REQ    DONE│
                              │            │
Line 23             0 ─┤ BASE    STAT├─ DA03_STAT
         Specifying the loaded base No          Indicating the error status during initialization
                                                function block execution
Line 24             3 ─┤ SLOT     ACT├─ DA03_ACT
         Specifying the loaded slot No           Indicating the run channels during initialization
Line 25        DA03_CH ─┤ CH          │         function block execution
         Specifying the used channel
Line 26        DA03_DT ─┤ DATA        │
         Specifying the data type       TYPE
Line 27        DA03_SE ─┤ SEL         │
         Specifying the D/A conversion
         module output status when the
         CPU modules is in the STOP state
                              └────────────┘
```

Line 29

```
Line 30   AD01_INI      DA02_INI     DA03_INI                          _INIT_DON
          DONE          DONE         DONE                              E
         ──┤ ├──────────┤ ├──────────┤ ├──────────────────────────────( )──
         The initialization  The initialization  The initialization         Initialization program
         function block      function block      function block             complete
         AD01_INI has normally DA02_INI has normally DA03_INI has normally
         finished            finished             finished
```

● Program : scan.src (scan program)

Comment — #### Reading the data of the moduel
A/D 01 every scan ####

```
                            ┌── AD01_RD ──┐                        AD01_RD_0
                            │   AD4RD     │                            K
Line  1                     │ REQ    DONE │─                        ─( )─
                            │             │                        The A/D conversion
                            │             │                        writing has normally
                        0 ──│ BASE   STAT │─ AD01_STAT             finished
Line  2                     │             │      E
     Specifying the loaded Base No        │  Indicating the error status during the reading
                            │             │  function block execution
                        1 ──│ SLOT   DATA │─ AD01_DT
Line  3                     │             │  Indicating the A/D conversion value data of
     Specifying the loaded Slot No        │  the used channels
              AD01_CH ──────│ CH          │
Line  4                     └─────────────┘
     Specifying the used channel No

Line  5
```

Comment — #### Reading the data of the moduel
D/A 02 every scan ####

```
                            ┌── DA02_WR ──┐                        DA02_WR_0
                            │   DA4AWR    │                            K
Line  7                     │ REQ    DONE │─                        ─( )─
                            │             │                        The D/A conversion
                            │             │                        writing has normally
                        0 ──│ BASE   STAT │─ DA02_STAT             finished
Line  8                     │             │  Indicating the error status during writing
     Specifying the loaded Base No        │  function block execution
                        2 ──│ SLOT   DATA │─ DA02_ACT
Line  9                     │             │  '   'ng the run channels during writing
     Specifying the loaded Slot No        │  STAT ↑ block execution
           DA02_CH ──────── │ CH          │
Line 10                     │             │
     Specifying the used channel No       │
           DA02_DT ──────── │ DATA        │
Line 11                     └─────────────┘
     Inputting the digital data to be
     set

Line 12
```

Comment — #### Writing the channel data to the module D/A 03
only when the writing command is applied ####

Comment — #### Using the MOV function at hot restart only
when the data restore is needed ####

Comment   The 'DA03_OUT', which is declared as a resource   STAT   STAT
          global variable

```
     DA03_DATA                                          ┌── DA03_WR ──┐                 DA03_WR_0
     _CHANGE          ┌── MOVE ──┐                       │   DA4AWR    │                     K
Line 16 ─┤ ├──────────│ EN    END│───────────────────── │ REQ    DONE │─               ─( )─
                      │          │                       │             │                The D/A conversion
                      │          │                       │             │                writing has normally
     DA03_DT ─────────│ IN1   OUT│─ DA03_OUT         0 ──│ BASE   STAT │─ DA03_STAT     finished
Line 17               └──────────┘  Specifying the loaded Base No      │  Indicating the error status during writing
                                                                       │  function block execution
                                                   3 ──│ SLOT    ACT │─ DA03_ACT
Line 18                               Specifying the loaded Slot No    │   Indicating the run channels during writing
                                                                       │   function block execution
                                          DA03_CH ── │ CH          │
Line 19                               Specifying the used channel No   │
                                          DA03_OUT ──│ DATA        │
Line 20                               Inputting the digital data to be └───┘
                                      set

Line 21
```

4.4   Operation Modes

The CPU module operates in one of the four modes - the RUN, STOP, PAUSE and DEBUG mode.
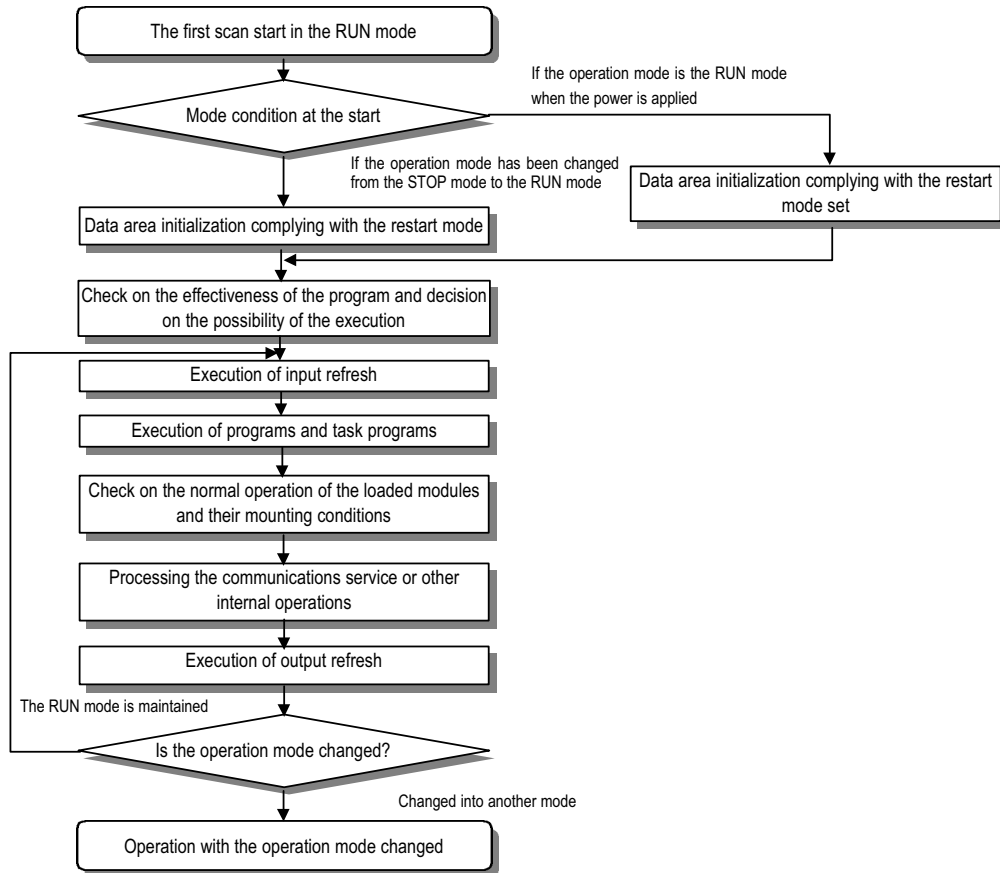The following describes the PLC operation processing in each operation mode.

4.4.1 RUN mode

In this mode, programs are normally operated.

```
┌──────────────────────────────────────┐
│  The first scan start in the RUN mode  │
└──────────────────────────────────────┘
                  │
                  ▼                              If the operation mode is the RUN mode
        ◇ Mode condition at the start ◇──────────  when the power is applied
                  │                                                    │
                  │   If the operation mode has been changed           ▼
                  │   from the STOP mode to the RUN mode    ┌────────────────────────────────┐
                  ▼                                         │ Data area initialization complying │
┌────────────────────────────────────────┐                │ with the restart mode set          │
│ Data area initialization complying with │                └────────────────────────────────┘
│ the restart mode                         │                                │
└────────────────────────────────────────┘◄───────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│ Check on the effectiveness of the program and decision │
│ on the possibility of the execution                    │
└────────────────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│       Execution of input refresh         │
└────────────────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│    Execution of programs and task programs │
└────────────────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│ Check on the normal operation of the loaded modules │
│ and their mounting conditions                       │
└────────────────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│ Processing the communications service or other │
│ internal operations                            │
└────────────────────────────────────────┘
                  │
                  ▼
┌────────────────────────────────────────┐
│       Execution of output refresh        │
└────────────────────────────────────────┘
                  │
The RUN mode is maintained
                  ▼
        ◇ Is the operation mode changed? ◇
                  │
                  │   Changed into another mode
                  ▼
┌────────────────────────────────────────┐
│  Operation with the operation mode changed │
└────────────────────────────────────────┘
```

1) Processing when the operation mode changes.

Initialization of data area is executed when the first scan starts.
(1) If the PLC is in the RUN mode when applying the power :
(2) If the operation mode has been changed into from the STOP mode into the RUN mode : the initialization is executed complying with the restart mode set. (cold / warm / hot)
(3) The possibility of execution of the program is decided with check on its effectiveness.

2) Operation processing contents

I/O refresh and program operation are executed.
(1) Task programs are executed with the detection of their start-up conditions.
(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.
(3) Communications service or other internal operations are processed.

4.4.2 STOP mode

In this mode, programs are not operated.

1) Processing when the operation mode changes

The output image area is cleared and output refresh is executed.

2) Operation processing contents

(1) I/O refresh is executed.
(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.
(3) Communications service or other internal operations are processed.

4.4.3 PAUSE mode

In this mode, the program operation is temporarily stopped. If it returns to the RUN mode, the operation continues from the state before the stop.

1) Processing when the operation mode changes

Data area clear and input image clear are not executed and the operating conditions just before the mode change is maintain.

2) Operation processing contents

(1) I/O refresh is executed.
(2) Normal or abnormal operation and mounting conditions of the loaded module are checked.
(3) Communications service or other internal operations are processed.

4.4.4 DEBUG mode

In this mode, errors of a program are searched and the operation sequence is traced. Changing into this mode is only possible in the STOP mode. In this mode, a program can be checked with examination on its execution state and contents of each data.

1) Processing when the operation mode changes

(1) Data area is initialized at the starting time of the mode change complying with the restart mode, which has been set on the parameters.
(2) The output image area is cleared and output refresh is executed.

2) Operation processing contents

(1) I/O refresh is executed by one time every scan.
(2) Communications service or other internal operations are processed.

3) Debug operation conditions

• Two or more of the following four operation conditions can be simultaneously specified.

Operation conditions	Description
Executed by the one operation unit, (step over)	If an operation command is ordered, the system operates one operation unit and stops.
Executed to the specified breakpoint.	• If break step is specified in the program, the operation stops at those step before execution. • Up to 8 breakpoints can be specified.
Executed according to the contact state	If the contact area to be watched and the condition (Read, Write, Value) where the operation has to stop are specified, the operation stops when the specified operation occurs at the specified contact.(after execution)
Executed by the specified scan number.	If the number of scan that will be operated is specified, the operation stops after it has operated by the specified scan number.

4) Operation method

(1) Execute the operation after the debug operation conditions have been set in the GMWIN.

(2) In task programs, each task can be specified to operation enable/disable.(For detailed operation method, refer to the GMWIN User's Manual Chapter 9.

4.4.5 Operation mode change

1) Operation mode change methods

The following method are used to change the operation mode.

(1) Change by the mode setting switch of CPU module.

(2) Change by the GMWIN connected with the CPU module communications port.

(3) Change by the GMWIN connected to the remote CPU module through Fnet.

(4) Change by the user's command using FAM or computer link module, etc.

(5) Change by the 'STOP function', 'ESTOP function' during program execution.

2) Operation mode change by the mode setting switch of CPU module

The following shows the operation mode change by the mode setting switch of CPU module.

Mode setting switch position	Operation mode
RUN	Local RUN
STOP	Local STOP
STOP → PAU/REM	Remote STOP
PAU/REM → RUN [1]	Local RUN
RUN → PAU/REM [2]	Local PAUSE / Remote RUN
PAU/REM → STOP	Local STOP

REMARK

1) If the operation mode changes from RUN mode to local RUN mode by the mode setting switch, the PLC operates continuously without stop.
2) If Local PAUSE disable(or Local PAUSE enable) is set by parameter in GMWIN, it operated as Remote RUN(or Local PAUSE).

3) Remote operation mode change

Remote operation mode change is available only when the operation mode is set to the remote STOP
mode (i.e., the mode setting switch position is in the 'STOP →PAU/REM').

Mode setting switch position	Mode Change	Mode change by the GMWIN	Mode change using FAM or computer link, etc.
PAU/REM	Remote STOP → Remote RUN	❍	❍
	Remote STOP → Remote PAUSE	×	×
	Remote STOP → DEBUG	❍	❍
	Remote RUN → Remote PAUSE	❍	❍
	Remote RUN → Remote STOP	❍	❍
	Remote RUN → DEBUG	×	×
	Remote PAUSE → Remote RUN	❍	❍
	Remote PAUSE → Remote STOP	❍	❍
	Remote PAUSE → Remote DEBUG	×	×
	DEBUG → Remote STOP	❍	❍
	DEBUG → Remote RUN	×	×
	DEBUG → Remote PAUSE	×	×

4) Remote operation mode change enable/disable

It is possible to disable the mode change for system protection so that some parts of the operation mode
sources cannot change the mode. If remote operation mode change has been disabled, the operation
mode change is possible only by the mode setting switch and GMWIN. To enable the remote operation
change, set the parameter 'Enabling the PLC control by communications' to enable. (For details, refer to
the Appendix 1. System Definitions)

4.5 Functions

4.5.1 Restart mode

The restart mode defines how to initialize variables and the system and how to operate in the RUN mode when the system starts its operation with the RUN mode by re-application of the power or mode change. Two restart modes, cold and warm restart are available and the execution condition for each restart mode is given below.
(For details, refer to the '4.5.1 Basic Parameters Edit' of the GMWIN User's Manual Section 4.5 Parameters Edit.

1) Cold Restart
   (1) It is executed when the restart mode parameter has been set to the cold restart mode.
   (2) All data are cleared with '0' and only the variables to which their initial value has been defined will be set to their initial value.
   (3) Though the parameter has been set to the warm restart mode, cold restart will be executed at the first execution of a program after it has been changed.
   (4) In case of selection 'Reset' command in the GMWIN, it restarts in accordance with setting in parameter and in case of selection 'Overall Reset' command, it restarts as cold restart mode.

2) Warm Restart
   (1) It is executed when the restart mode parameter has been set to the warm restart mode.
   (2) A data which set as retain & initial will be retain and a data which set as initial value will be set with default value during the warm restart. All other data will be cleared with '0'.
   (3) Though the parameter has been set to the warm restart mode, cold restart will be executed at the first execution of a program after it has been stopped due to its down load and error.
   (4) Though the parameter has been set to the warm restart mode, cold restart will be executed if data contents are abnormal (i.e., the data does not remain at a power failure)

• Restart mode is executed as the figure given below when the power has been re-applied during execution of the CPU module
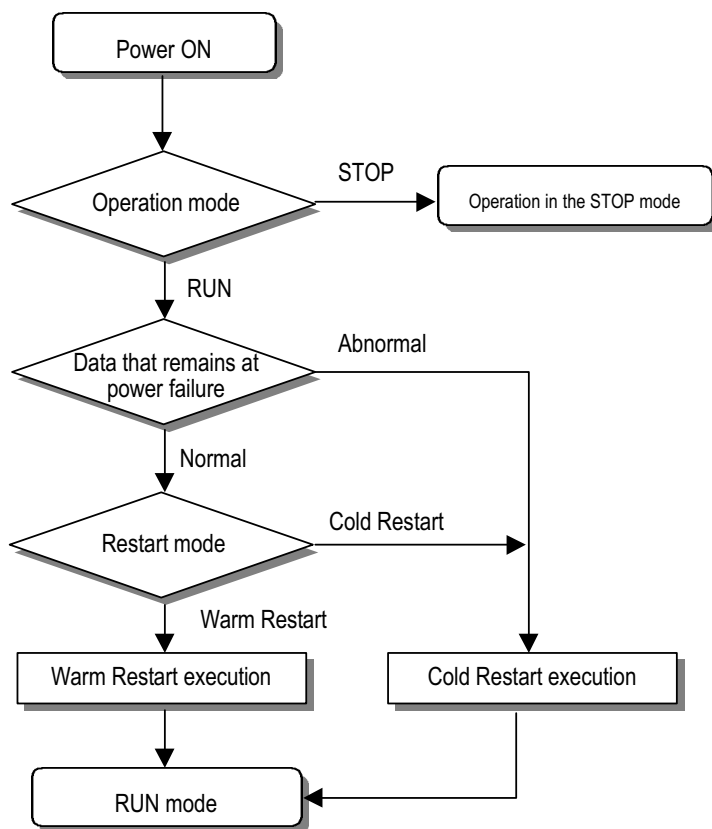
4) Data initialization according to the restart mode

The variables relating to the restart mode are classified into three types, i.e, default variable, initialization variable and retain variable. The following table shows the initialization method for each type variable.

Mode / Variable type	Cold	Warm
Default	Initialized with ' 0'	Initialized with ' 0'
Retain	Initialized with ' 0'	Previous value is retained
Initialization	Initialized with the user defined value	Initialized with the user defined value
Retain & Initialization	Initialized with the user defined value	Previous value is retained

---

REMARK

1) Definitions
   (1)Default variable :   A variable whose initial value is not defined or previous value will not be retained.
   (2)Initialization variable : A variable whose initial value is defined.
   (3)Retain variable : A variable whose previous value will be retained.

4.5.2 Self-diagnosis

1) Functions

(1) The self-diagnosis function permits the CPU module to detect its own errors.

(2) Self-diagnosis is carried out when the PLC power supply is turned on and when an error occurs the PLC is in the RUN state. If an error is detected, the system stops operation to prevent faulty PLC operation.

2) Error flag

If an error occurs, it will be stored to the following flags and the STOP LED flickers.
- Representative system error flag : _CNT_ER
- Representative system warning flag : _CNF_WAR

REMARK

1) Refer to 12.5 Error Code List of Chapter 12. Troubleshooting for details of contents of self-diagnosis and corrective actions.

4.5.3 Remote function

The CPU module can be controlled by external operations (from GMWIN and computer link module, etc.). For remote operation, set the mode setting switch of CPU module to remote position.

1) Remote RUN/STOP

(1) The remote RUN/STOP permits external operations to RUN/STOP the CPU module under the condition that the mode setting switch of CPU module is in the remote position.

(2) This function is convenient when the CPU module is located on the place where it is difficult to control the CPU module or the user want to control the CPU module in the control panel from outside.

2) Remote PAUSE

(1) The remote PAUSE permits external operations to execute PAUSE operations under the condition that the mode setting switch of CPU module is in the remote position. The PAUSE operations stop the CPU module operation processing while maintaining the On/Off state of the output module.

(2) This function is convenient when the user wants to maintain the ON state of the output module under the condition the CPU module has been stopped.

3) Remote DEBUG

(1) This function permits external operations to execute DEBUG operations under the condition that the mode setting switch of CPU module is in the remote position. The DEBUG operations execute programs complying with the specified operation conditions.

(2) This function is convenient when program execution or contents of any data are checked for debugging of the program.

4) Remote reset

(1) This function permits remote operations to reset the CPU module, which locates in the place where direct operations cannot be applied, when an error has occurred.

REMARK
1) For remote function operations, refer to the GMWIN User's Manual Chapter 7. On-line.

4.5.4 I/O Force On/Off function

1) Force On/Off setting method

Force on/off setting is applied to input area and output area.
Force on/off should be set for each input and output, the setting operates from the time that 'Force I/O setting enable' is set.
This setting can be done when I/O modules are not really loaded.

2) Force on/off Processing timing and method

(1) Force Input

- After data have been read from input modules, at the time of input refresh the data of the junctions which have been set to force on/off will be replaced with force setting data to change the input image area. And then, the user program will be executed with real input data and force setting data.

(2) Force output

- When a user program has finished its execution the output image area has the operation results. At the time of output refresh the data of the junctions which have been set to force on/off will be replaced with force setting data and the replaced data will be output. However, the force on/off setting does not change the output image area data while it changes the input image area data.

(3) Force on/off processing area

- Input/output areas for force on/off setting are larger than the real I/O areas. If remote I/O is specified using this area, the force on/off function is as just available in it as in the basic I/O areas.

(4) Precautions

- Turning the power off and on, change of the operation mode or operation by reset switch(GM3) does not change the previous force on/off setting data. They remain within the CPU module and operation is executed with the same data.
- Force I/O data will not be cleared even in the STOP mode.
- If a program is downloaded or its backup breaks, the force on/off setting data will be cleared. The operating program in memory differs from the program in the flash memory so that if operation restarts with the program in the flash memory the on/off setting data will be also cleared.
- When setting new data, disable every I/O settings using the setting data 'clear' function and set the new data.

REMARK

1) For detailed operation, refer to the GMWIN User's Manual Chapter 7 'Force I/O setting.

4.5.5 Direct I/O Operation function

This function is usefully available when an input junction state is directly read during execution of a program and used in the operation, or the operation result is directly output to an output junction.

1) Direct input

- Direct input is executed by use of the 'DIRECT_IN' function. If this function is used, the input image area will be directly updated and applied to the continuing operations.

2) Direct output

- Direct output is executed by use of the 'DIRECT_O' function. If this function is used, the data of the output image area, which has the operation results by the time, will be directly output to the direct output module.

3) Force on/off

- Force on/off settings are still effective when processing direct I/O.

REMARK

1) For detailed direct I/O functions, refer to the GLOFA-GM commands.

4.5.6 External Device Error Diagnosis function

Flags are given for the user to implement easily the program in which the error detection of external devices and system stop and warning are coded. By use of these flags, error indication of external devices is possible without complex programming and monitoring of the error location can be done without special tools (GMWIN, etc.) or source programs.
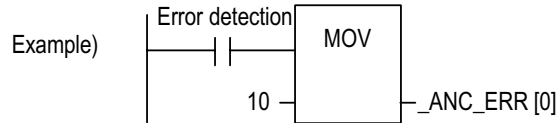
1) External device fault detection and classification

(1) The user program detects external device faults. The faults are classified into fatal fault(error), where the PLC stops its operation, and ordinary fault(warning), where operation continues.

(2) The flag ANC_ERR[n] is used to indicate error. The flag ANC_WN[n] is used to indicate warning.
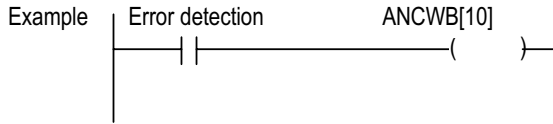
2) External Device Fatal-fault (Error) Processing.

(1) If an error of external device is detected and the error type, where other value than 0 is used, is written to the system flag ANC_ERR[n], the flag will checked at the time that scan program finishes its execution. If an error is indicated on the flag, it will be also indicated on the _ANNUN_ER of the representative system error flag _CNF_ER, the PLC turns all output modules off and the error state will be same as the PLC self-diagnosis.

(2) The user can know the cause of error by use of the GMWIN, and also by direct monitoring of the flag _ANC_ERR[n].

(3) As the flag _ANC_ERR[n] has sixteen elements(n : 0 to 15), the user can classify error states largely.

User defined error No.   can be written to the elements. A number of 1 to 65535 is usable.

Example)

```
        Error detection
                  ┌──────┐
   ────────┤ ├────┤ MOV  │
                  │      │
            10 ───┤      ├─── _ANC_ERR [0]
                  └──────┘
```

3) External device Ordinary-fault (Warning) Processing.

(1) If a warning of external device is detected and the corresponding flag of the system flag _ANC_WB[n]
   is set to on, the flag will checked from the _ANC_WB[0] at the time that scan program finishes its
   execution. If an error is indicated on the flag, it will be also indicated on the _ANNUN_WR of the
   representative system warning flag _CNF_WAR. External device waning numbers will be written to
   from _ANC_WAR[0] to _ANC_WAR[7] according to   occurrence sequence.

(2) The user can know the cause of error by use of the GMWIN, and also by direct monitoring of the flags
   _ANC_WAR[n] and _ANC_WB[n].

(3) If an external device waning is removed, that is, the elements of _ANC_WAR[n] are released from
   warning, the corresponding _ANC_WAR[n] will be automatically cleared. If all element flags are
   cleared, the flag _ANNUN_WR of the system flag _CNF_WAR will be reset.

```
Example ┬ Error detection              ANCWB[10]          ┬
        │    ──┤ ├──────────────────────( )──            │
        │                                                 │
        │                                                 │
```

_ANNUN_WR	=	1
_ANC _WAR[0]	=	10
_ANC _WAR[1]	=	0
_ANC _WAR[2]	=	0
_ANC _WAR[3]	=	0
_ANC _WAR[4]	=	0
_ANC _WAR[5]	=	0
_ANC _WAR[6]	=	0
_ANC _WAR[7]	=	0

If the user program had detected a system fault and set _ANC_WB[10] to ON, the states of _ANNUN_WR and _ANN_WAR [0..7] will be shown as left after the scan has been finished

_ANNUN_WR	=	1
_ANC _WAR[0]	=	10
_ANC _WAR[1]	=	0
_ANC _WAR[2]	=	0
_ANC _WAR[3]	=	0
_ANC _WAR[4]	=	0
_ANC _WAR[5]	=	0
_ANC _WAR[6]	=	0
_ANC _WAR[7]	=	0

After the next scan has been finished, if the numbers 1, 2, 3, 10, 15 , 40 , 50, 60 and 75 of _ANC_WB[n] are tuned on _ANC_WAR[n] will be shown as left

As the number 10 has turned on (has occurred) in the previous scan, though the number 10 has lower priority than the numbers 1, 2 and 3, it will be the lower element of _ANC_WAR[n]. The _ANC_WB[75] is not indicated as it is turned on and the warning that occurred before has written to the _ANC_WAR[n].

_ANNUN_WR	=	1
_ANC _WAR[0]	=	10
_ANC _WAR[1]	=	0
_ANC _WAR[2]	=	0
_ANC _WAR[3]	=	0
_ANC _WAR[4]	=	0
_ANC _WAR[5]	=	0
_ANC _WAR[6]	=	0
_ANC _WAR[7]	=	0

After the next scan has been finished, if the numbers 1, 2, 3, 10, 15 , 40 , 50, 60 and 75 of _ANC_WB[n] are tuned on _ANC_WAR[n] will be shown as left.

The No. 10 warning has been released the content of _ANC_WAR[0] will be cleared and the contents of _ANC_WAR[1..7] will shift into the lower elements. The content of _ANC_WAR[7] will has been cleared by the shifting and the content of _ANC_WB[75] will be written to _ANC_WAR[7].

_ANNUN_WR	=	1
_ANC _WAR[0]	=	10
_ANC _WAR[1]	=	0
_ANC _WAR[2]	=	0
_ANC _WAR[3]	=	0
_ANC _WAR[4]	=	0
_ANC _WAR[5]	=	0
_ANC _WAR[6]	=	0
_ANC _WAR[7]	=	0

If all warnings indicated on the _ANC_WB[n] are released during operation, the _ANNUN_WR and _ANC_WAR[n] will be shown as left.

4.6 Memory Configuration

The CPU module includes two types of memory that are available by the user. One is program memory which is used to store the user programs written to implement a system by the user. The other is data memory which stores data during operation.

1) Program memory configuration

The table given below shows the contents to be stored and the storage capacity of program memory.

Item	Memory Capacity
Overall program memory area	68 k bytes
Parameter area • Basic parameter area • I/O parameter area • High speed link parameter area • Interrupt setting information area	2 k bytes
Program area • Scan program area • Task program area • User defined function/function block area • Standard library area • Access variable are • Variable initialization information area • Protective variable specification information area	66 k bytes

2) Data memory Configuration

The table given below shows the contents to be stored and the storage capacity of program memory.

Item	Memory Capacity
Overall data memory area	32 k bytes
System area • I/O information table • Force I/O table	1 k bytes
System flag area	1.5 k bytes
Input image area (%IX)	128 k bytes
Output image area (%QX)	128 k bytes
Direct variable area (%M)	2 to 8 k bytes
Symbolic variable area (maximum)	29 k bytes – the size of direct variable area
Stack area	3 k bytes

3) Purpose

(1) System area

it used to store the self-created data of the CPU module for system management and GMWIN system control data.

(2) System flag area

it used to user flags and system flags. The user operates it with flag name.

(3) Input image area

it used to store input data read from input modules. Overall size is %IX0.0.0 to %IX1.7.63.

(4) Output image area

It used to store operation results. The stored data are automatically output to output modules. Overall size is %QX0.0.0 to %QX1.7.63.

(5) Direct variable area

The user can use this area to access direct memory data through the variable names such as %MX0, %MB0, %MW0 and %MD0, which was pre-defined by the system. Memory size is defined when program is made by user and it refers to 'App1. System Definitions'.

(6) Symbolic variable area

It used to store the variables that the user created, that is, whose names the user defined when writing a program.   Global variables and instance memory are located in this area. The variables used in program blocks locates in the 'PB instance memory' of the program, and the memory used in function block locates in the 'FB instance memory'.

The maximum size of the PB instance memory is 32 Kbytes. If the used size overruns the maximum size, divide the program blocks or use global variables.

4.7　I/O No. Allocation Method

1) I/O No. allocation means to give an address to each module in order to read data from input modules and output data to output modules.

2) Fixed 64 points are allocated to each module for I/O points.

3) Fixed 64 points are allocated regardless of mounting/dismounting or type of modules.
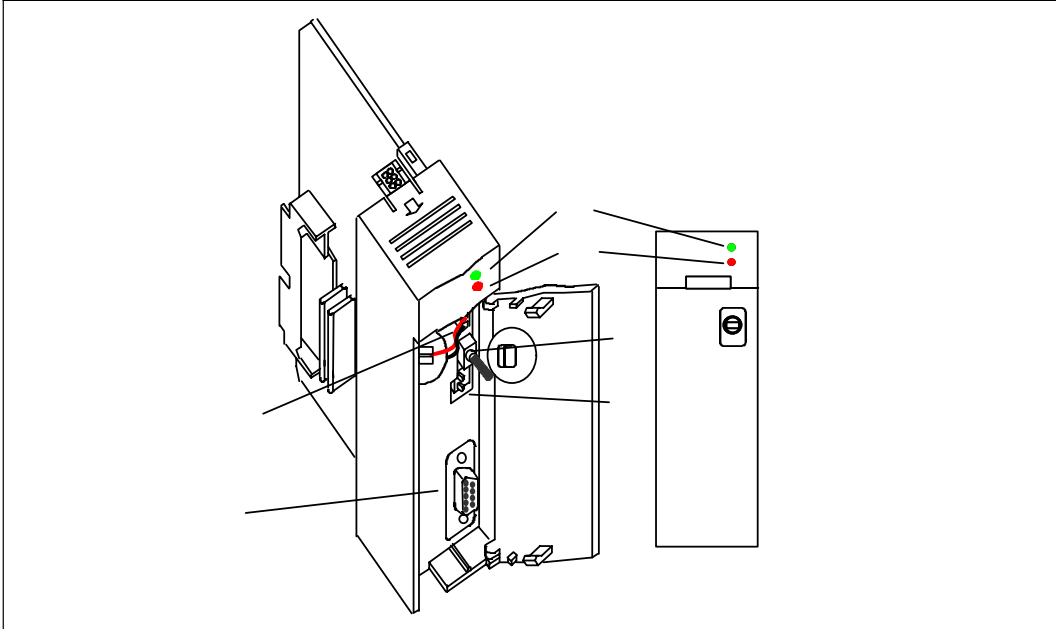
4) The following shows I/O No. allocation method.

```
Input   : %   I   X   0.   0.   0
Output : %   Q   X   0.   1.   15
```

Contact number on I/O module
0 ~ 63

Slot number of Base
0 ~ 7
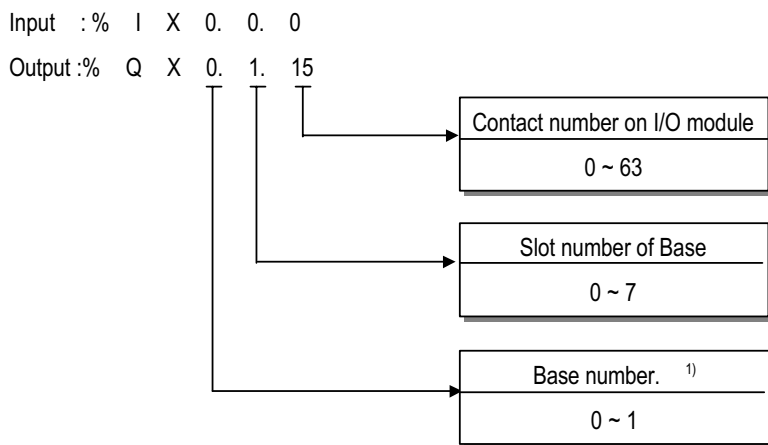
Base number.   1)
0 ~ 1

REMARK

　1) Although there is no expansion base, A base having more than 8 I/O slot which has a plan to develop set by 1 as base number.

4.8 Names of Parts

The following describes the names and functions of parts of the CPU module.

No.	Name	Function

1	RUN LED	Indicates the operation status of the CPU module.
		• On : when the CPU module operates with the mode setting switch in the local or remote RUN state.
		• Off : when the followings occur
		  The voltage is not normally supplied to the SPU module.
		  The mode setting switch is in the STOP or PAU/REM state.
		  An error which makes operation stop is detected.

2	STOP LED	• On : when the mode setting switch is in the local or remote STOP state.
		• Off : when the followings occur
		  The mode setting switch is in the local RUN or local PAUSE state.
		  The operation state is in the RUM/PAUSE/DEBUG state.
		• Flickering : when an error is detected by self-diagnosis during operation.

3	Mode setting switch	Sets the operation mode of the CPU module. .
		• RUN : Program operation is executed.
		• STOP : Program operation is temporarily stopped.
		• PAU/REM :
		    PAUSE : Program operation is temporarily stopped.
		    REMOTE: Used for the remote operation

4	DIP S/W for flash memory	See chap 6.
5	Battery installing connector	It used to connect to the backup battery.
6	RS-232C connector	It used to connect to peripheral devices(GMWIN, etc.)

REMARK

The followings shows the LED status complying with the operation mode, and the operation mode complying with the position of the mode setting switch.

1) LED status complying with the operation mode

Operation Mode	LED Status		
	RUN	STOP	REMOTE
Local Run	On	Off	Off
Local Stop	Off	On	Off
Local Pause	Off	Off	Off
Remote Run	On	Off	On
Remote Stop	Off	On	On
Remote Pause, Remote Debug	Off	Off	On

2) Operation mode complying with the position of the mode setting switch.

Position of Mode switch	Operation Mode
STOP $\longrightarrow$ PAU/REM	Remote Stop
PAU/REM $\longrightarrow$ RUN	Local Run
RUN $\longrightarrow$ PAU/REM	Local Pause [1]
PAU/REM $\longrightarrow$ STOP	Local Stop

● Change of remote mode is available only after the operation mode has entered into the remote STOP mode.

caution   1) In case of local pause disable, it operated as Remote Run.

Chapter 5.   BATTERY

5.1   Specifications

| Item	Specifications
Normal voltage	3.0 VDC
Warranty life time	5 years
Application	Programs and data backup, and RTC runs in power failure
Specifications	Lithium Battery, 3 V
External dimension (mm)	$\Phi 14.5 \times 26$

5.2   Handling Instructions

1) Do not heat or solder its terminals.

2) Do not measure its voltage with a tester or short circuit.

3) Do not disassemble.

5.3   Battery Replacement

Backup battery needs periodic exchange. When the battery exchange, it should be done at power on, otherwise some or all data will be lost.

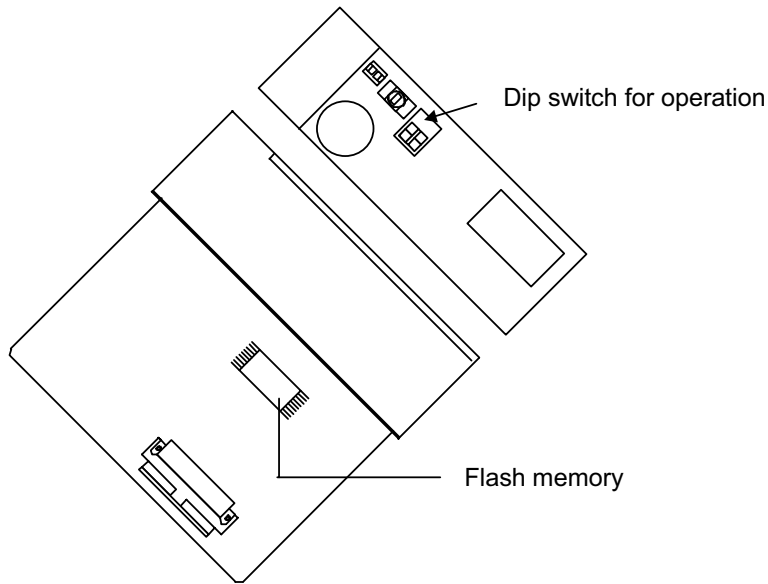The following shows the battery replacement procedure.
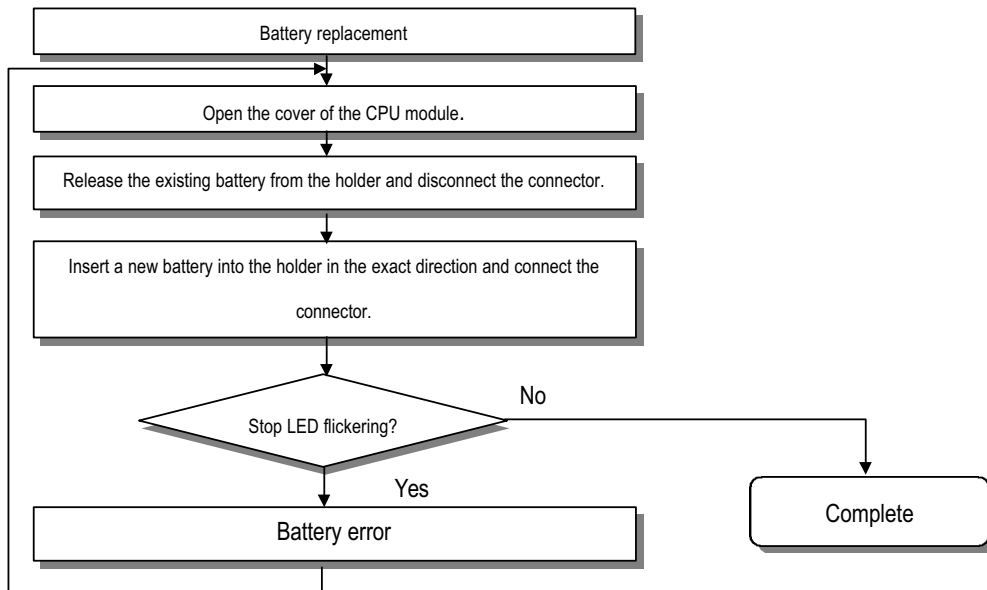
Chapter. 6   USING THE USER PROGRAM IN FLASH MEMORY

This chapter describes user program storage and operation it.

Flash memory is used to store a user program and installed in PLC.

6.1   Structure

Dip switch for operation

Flash memory

6.2   How to use

Read / Write is available to flash memory in accordance with selection of DIP switch.

Selection of DIP switch for flash memory	Operation
ON	PLC is operated by the program in flash memory when power on or PLC reset.
ON	PLC recognize that no program is in flash memory.

(Caution : Lower switch should be at the off position.)

User program can be written to flash memory at the PLC stop mode and then the selection of switch is ignored.

Chapter. 7   DIGITAL INPUT AND OUTPUT MODULES

7.1   Notes on Selecting Input and Output Modules

The followings describe instructions for selection of digital I/O modules that will be used in the GLOFA-GM6 series.

1) The types of digital input are current sink input and current source input.
   When selecting DC input modules consider the specifications of those input devices as the wiring method of the external input power supply varies complying with the type of digital input.
   In the GM6 series, the types are dedicated source input and source/sink common DC input.

2) Maximum simultaneous input points differs with the type of a module. Check the specifications of the input module to be applied before use.

3) Use transistor or triac output modules with a load that is frequently opened and closed or with an inductive load as, in those cases, the life span of a relay output module will become shorter than specified.
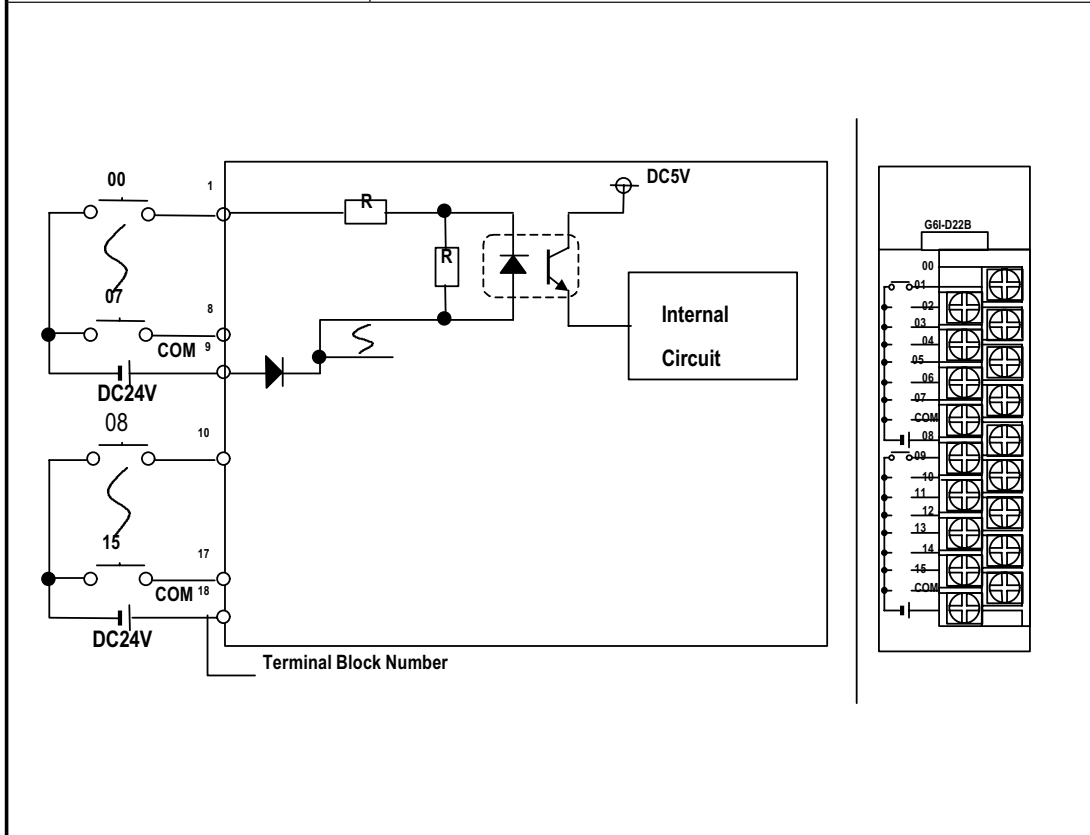
7.2 Digital Input Module Specifications

7.2.1 16-points 24 VDC input module (source/sink type)

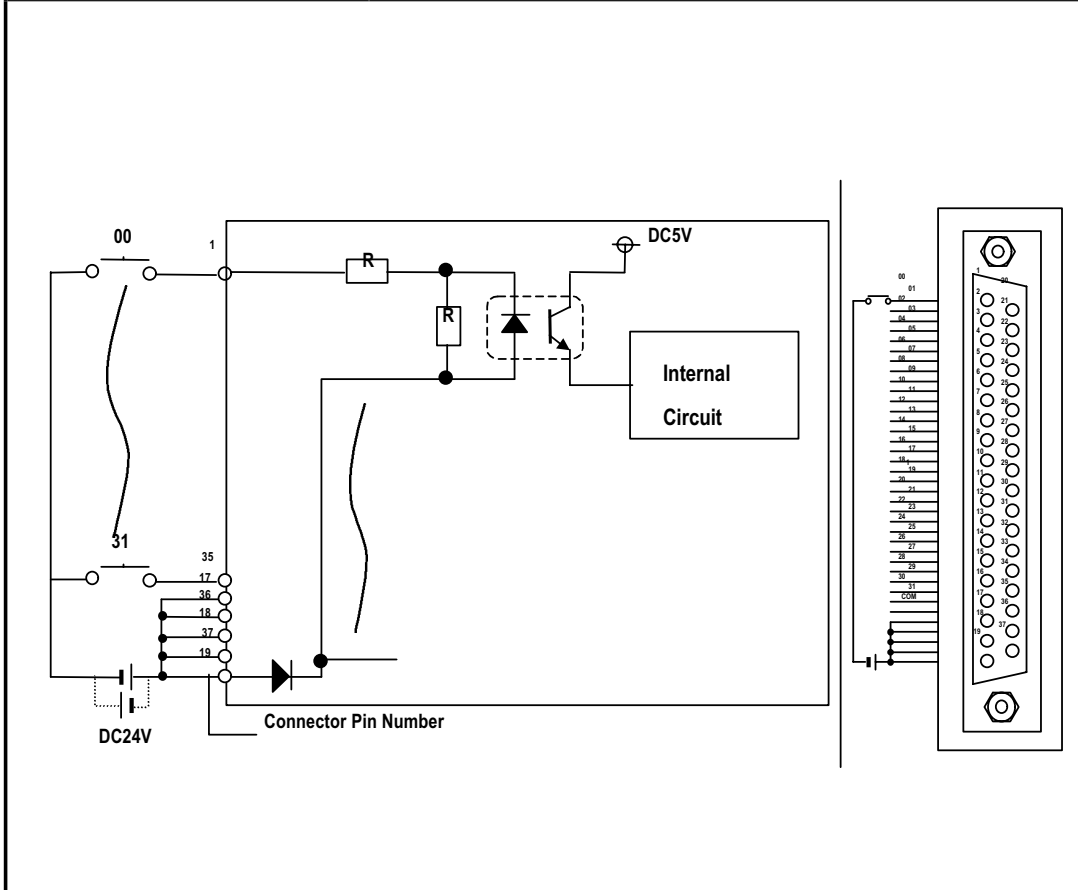| Specifications \ Model	DC Input Module G6I-D22A
Number of input points	16 points
Insulation method	Photo coupler
Rated input voltage	24 VDC
Rated input current	7 mA
Operating voltage range	20.4 VDC to 28.8 VDC (ripple: less than 5%)
Maximum simultaneous input points	100%(8 points/COM) simultaneously ON
ON voltage/ON current	15 VDC or higher/4.3 mA or higher
OFF voltage/OFF current	5 VDC or lower/1.7 mA or lower
Input impedance	Approx. 3.3 kΩ
Response time OFF → ON	1.2 ~ 4.8 ms or less
Response time ON → OFF	1.2 ~ 4.8 ms or less
Common terminal	8 points/COM
Internal current consumption	70 mA
Operating indicator	LED turns on at ON state of input
External connections	18-points terminal block connector(M3 × 6 screws)
Weight	0.15 kg
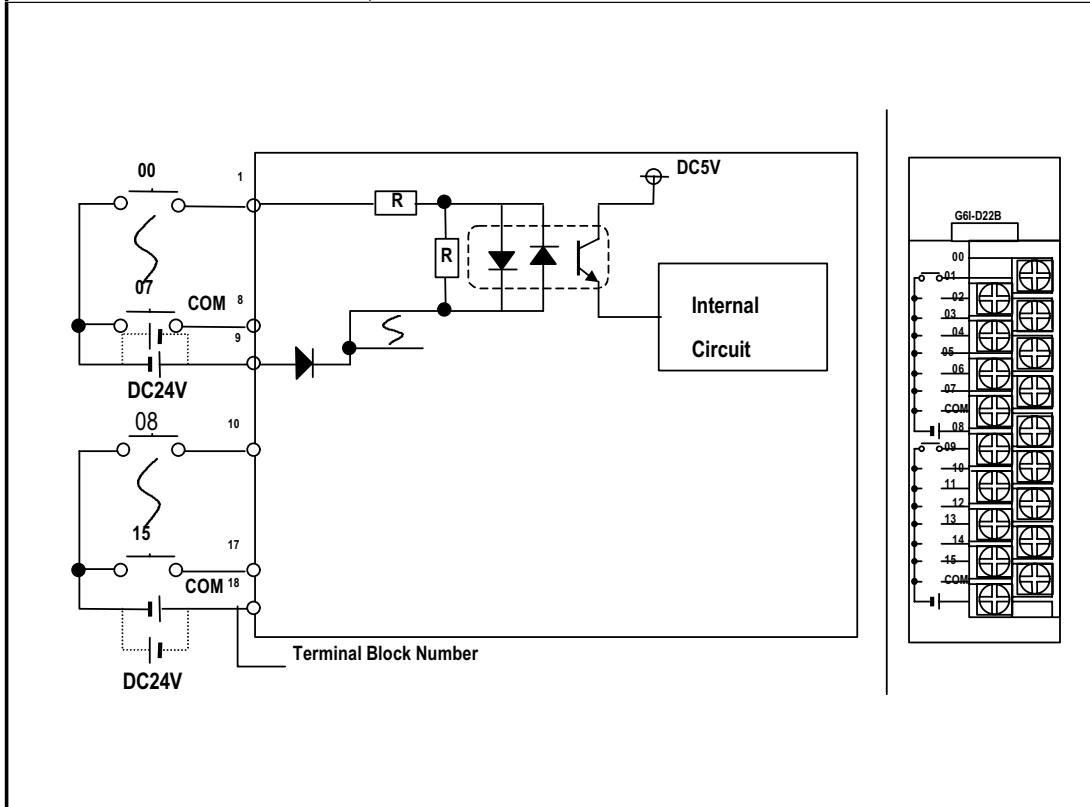
7.2.2   16-points 24 VDC input module (source type)

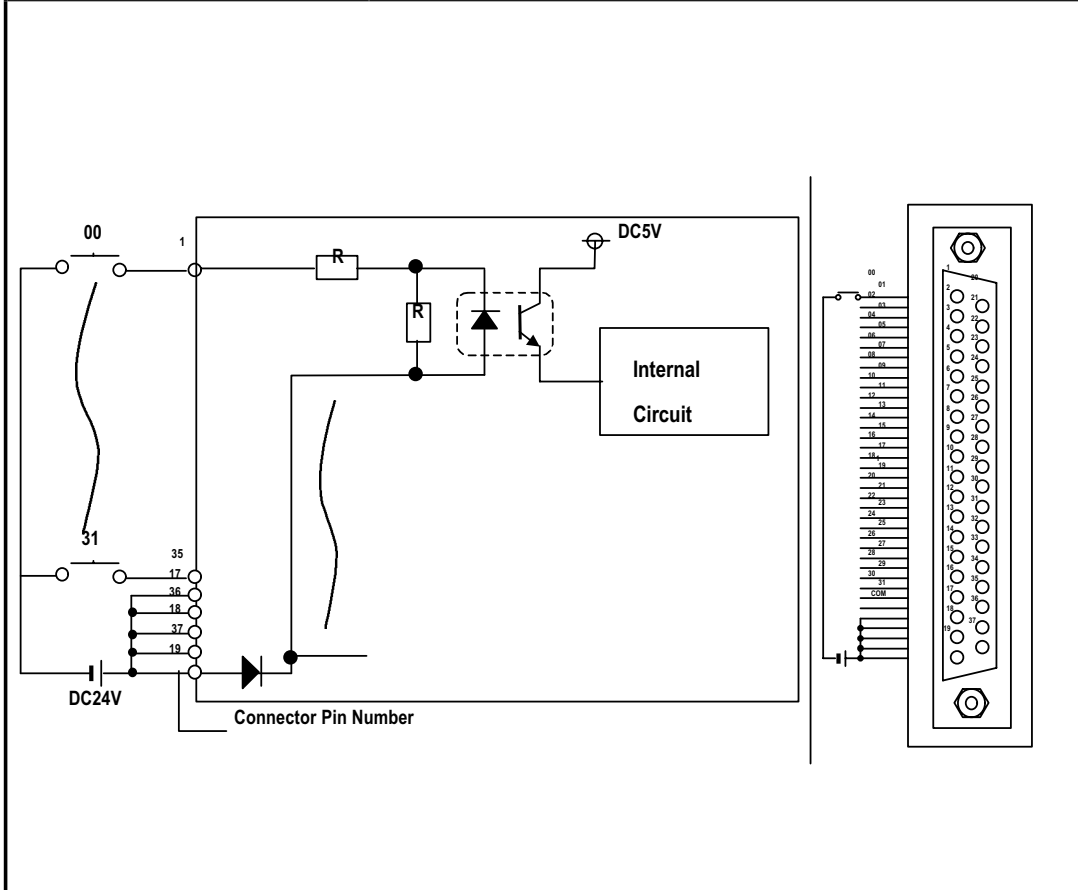| Specifications \ Models	DC Input Module
	G6I-D22B
Number of input points	16 points
Insulation method	Photo coupler
Rated input voltage	24 VDC
Rated input current	7 mA
Operating voltage range	20.4 to 28.8 VDC (ripple: less than 5%)
Maximum simultaneous input points	100% (8 points/COM) simultaneously ON
ON voltage/ON current	15 VDC or higher/4.3 mA or higher
OFF voltage/OFF current	5 VDC or lower/1.7 mA or lower
Input impedance	Approx. 3.3 kΩ
Response time OFF $\rightarrow$ ON	1.2 ~ 4.8 ms or less
Response time ON $\rightarrow$ OFF	1.2 ~ 4.8 ms or less
Common terminal	8 points/COM
Internal current consumption	70 mA
Operating indicator	LED turns on at ON state of input
External connections	18-point terminal block connector(M3 $\times$ 6 screws)
Weight	0.15 kg

7.2.3   32-points 24 VDC input module (source/sink type)

Model Specifications	DC Input Module G6I-D24A
Number of input points	32 points
Insulation method	Photo coupler
Rated input voltage	24 VDC
Rated input current	7 mA
Operating voltage range	20.4 to 28.8 VDC (ripple: less than 5%)
Maximum simultaneous input points	60% simultaneously ON
ON voltage/ON current	15 VDC or higher/4.3 mA or higher
OFF voltage/OFF current	5 VDC or lower/1.7 mA or lower
Input impedance	Approx. 3.3 kΩ
Response time OFF → ON	1.2 ~ 4.8 ms or less
Response time ON → OFF	1.2 ~ 4.8 ms or less
Common terminal	32 points/COM
Internal current consumption	75 mA
Operating indicator	LED turns on at ON state of input
External connections	37-point terminal block connector(M3 × 6 screws)
Weight	0.11 kg

7.2.4   32-points 24 VDC input module (source type)

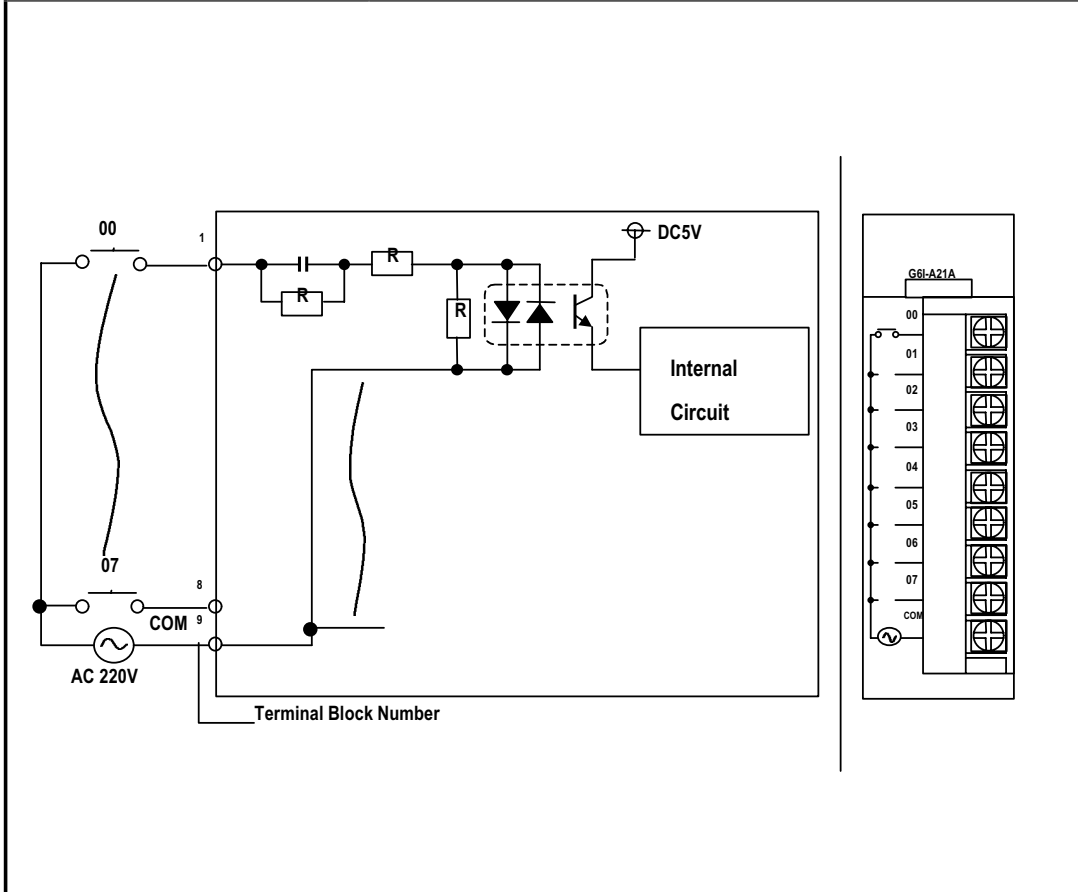| Specifications \\ Model	DC Input Module
	G6I-D24B
Number of input points	32 points
Insulation method	Photo coupler
Rated input voltage	24 VDC
Rated input current	7 mA
Operating voltage range	20.4 to 28.8 VDC (ripple: less than 5%)
Maximum simultaneous input points	60% simultaneously ON
ON voltage/ON current	15 VDC or higher/4.3 mA or higher
OFF voltage/OFF current	5 VDC or lower/1.7 mA or lower
Input impedance	Approx. 3.3 kΩ
Response time OFF → ON	1.2 ~ 4.8 ms or less
Response time ON → OFF	1.2 ~ 4.8 ms or less
Common terminal	32 points/COM
Internal current consumption	75 mA
Operating indicator	LED turns on at ON state of input
External connections	37-point terminal block connector(M3 × 6 screws)
Weight	0.11 kg

7.2.5   8-points 110 VAC input module

Models Specifications		AC Input Module
		G6I-A11A
Number of input points		8 points
Insulation method		Photo coupler
Rated input voltage		100 to 120 VAC (50/60 Hz)
Rated input current		7 mA (110 VAC / 60 Hz)
Operating voltage range		85 to 132 VAC (50/60 Hz $\pm$ 3 Hz)
Maximum simultaneous input points		100%(8 points / COM) simultaneously ON
Inrush current		300 mA, 0.3 ms or lower (132 VAC)
ON voltage/ON current		80 VAC or higher/5 mA or higher
OFF voltage/OFF current		30 VAC or lower/2 mA or lower
Input impedance		Approx. 10 kΩ
Response time	OFF $\rightarrow$ ON	15 ms or less
	ON $\rightarrow$ OFF	25 ms or less
Common terminal		8 points/COM
Internal current consumption		60 mA
Operating indicator		LED turns on at ON state of input
External connections		9-point terminal block connector(M3 $\times$ 6 screws)
Weight		0.14 kg

7.2.6   8-points 220 VAC input module

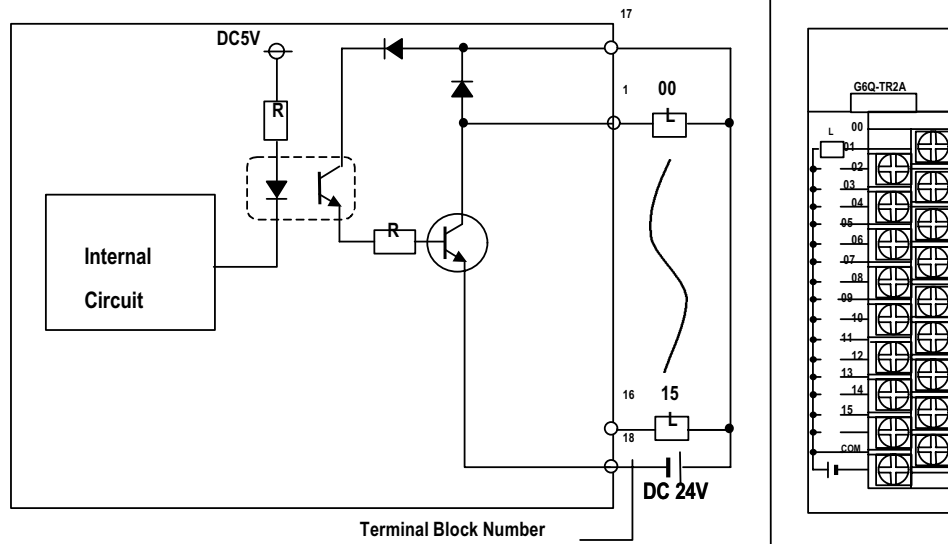| Model Specifications		GM4
		G6I-A21A
Number of input points		8 points
Insulation method		Photo coupler
Rated input voltage		200 to 240 VAC (50/60 Hz)
Rated input current		11 mA (220 VAC / 60 Hz)
Operating voltage range		85 to 264 VAC (50/60 Hz $\pm$ 3 Hz)
Maximum simultaneous input points		100%(8 points / COM) simultaneously ON
Surge input current		600 mA, 0.12 ms or lower (264 VAC)
ON voltage/ON current		80 VAC or higher/5 mA or higher
OFF voltage/OFF current		30 VAC or lower/2 mA or lower
Input impedance		Approx. 10 kΩ
Response time	OFF $\rightarrow$ ON	15 ms or less
	ON $\rightarrow$ OFF	25 ms or less
Common terminal		8 points/COM
Internal current consumption		60 mA
Operating indicator		LED turns on at ON state of input
External connections		9-point terminal block connector(M3 $\times$ 6 screws)
Weight		0.14 kg

7.3 Digital Output Module Specifications

7.3.1   16-point relay output module

Specifications \ Models	Relay Output Module
	G6Q-RY2A
Number of output points	16 points
Insulation method	Photo coupler
Rated load voltage & current	24 VDC 2A(resistance)/point, 5 A/ COM
	220 VAC 2A(COSΨ = `1)/point, 5A/COM
Minimum load voltage/current	5 VDC / 1 mA
Maximum load voltage/current	250 VAC, 110 VDC
Maximum switching frequency	1200 times per hour
Surge absorber	None
Service life — Mechanical	20 million times or more
Service life — Electrical	Rated load voltage/current 100000 times or more
	200 VAC 1.5 A, 240 VAC 1 A (COSΨ = 0.7) : 100000 times or more
	200 VAC 1 A, 240 VAC 0.5 A (COSΨ = 0.35) : 100000 times or more
	24 VAC 1.5 A, 100 VDC 0.1 A (L/R= 7 ms) : 100000 times or more
Response time — Off → On	10 ms or less
Response time — On → Off	12 ms or less
Common terminal arrangement	8 points/COM
Internal current consumption	415mA
Operation indicator	LED turns on at ON state of output
External connections	18-point terminal block connector (M3 × 6 screws)
Weight	0.19 kg

7.3.2 16-point transistor output module (sink type)

| Models Specifications		Transistor Output Module
		G6Q-TR2A
Number of output points		16 points
Insulation method		Photo coupler
Rated load voltage/current		12/24 VDC
Operating load voltage range		10.2 to 26.4 VDC
Maximum load current		0.5 A /point, 4 A / COM
Off leakage current		0.1 mA
Maximum inrush current		4 A, 10 ms or less
Maximum voltage drop at ON circuit		1.5 VDC(0.5A)
Surge absorber		Clamp Diode
Response time	Off → On	2 ms or less
	On → Off	2 ms or less
Common terminal arrangement		16 points/COM
Internal current consumption		185 mA
External power supply	Voltage	24 VDC ± 10 % (ripple voltage : 4VP-P or less)
	Current	30 mA or less (all points ON)
Operation indicator		LED turns on at ON state of output
External connections		18-point terminal block connector (M3 × 6 screws)
Weight		0.18 kg

7.3.3   32-point transistor output module (sink type)

Specifications \ Models		Transistor Output Module
		G6Q-TR4A
Number of output points		32 points
Insulation method		Photo coupler
Rated load voltage/current		12/24 VDC
Operating load voltage range		10.2 to 26.4 VDC
Maximum load current		0.1 A / point, 2 A /COM
Off leakage current		0.1 mA or less
Maximum inrush current		4 A, 10 ms or less
Maximum voltage drop at ON circuit		1.0 VDC
Surge absorber		None
Response time	Off → On	2 ms or less
	On → Off	2 ms or less
Common terminal arrangement		32 points/COM
Internal current consumption		139 mA
External power supply	Voltage	24 VDC ± 10 % (ripple voltage : 4VP-P or less)
	Current	36 mA or less (24 VDC/COM)
Operation indicator		LED turns on at ON state of output
External connections		37-pin D Sub-connector
Weight		0.11 kg
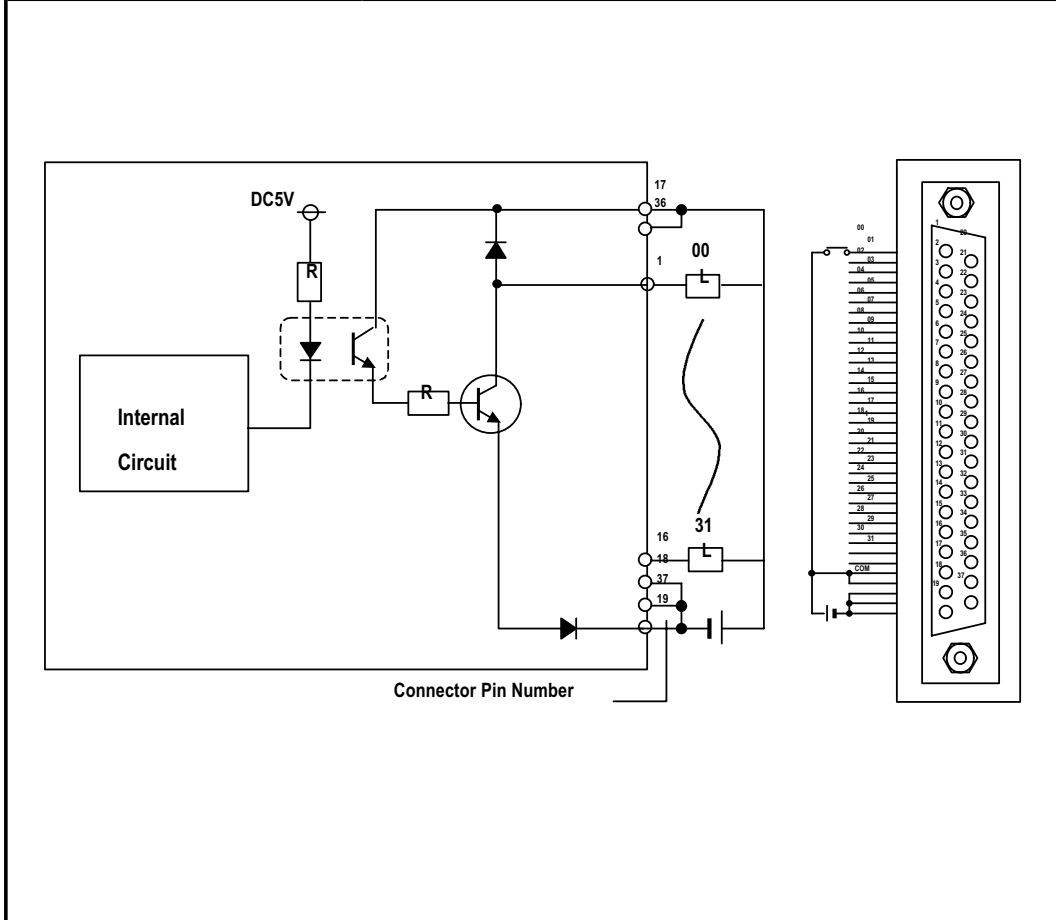
7.3.4   8-point triac output module

| Models Specifications	Triac Output Module G6Q-SS1A
Number of output points	8 points
Insulation method	Photo coupler
Rated load voltage	100 to 240 VAC (50 to 60 Hz)
Minimum load voltage	264 VAC
Maximum load current	1 A / point, 4 A / 1 COM
Minimum load current	20 mA
Off leakage current	2.5 mA (220 VAC, 60 Hz)
Maximum inrush current	40 A, 10 ms or less
Maximum voltage drop at ON circuit	1.5 VAC or less (2 A)
Surge absorber	Varistor (387 to 473 V), C.R absorber
Response time — Off → On	1ms or less
Response time — On → Off	1ms + 0.5 cycle or less
Common terminal arrangement	8 points/COM
Internal current consumption	210 mA
Operation indicator	LED turns on at ON state of output
External connections	9-point terminal block connector (M3 × 6 screws)
Weight	0.16 kg
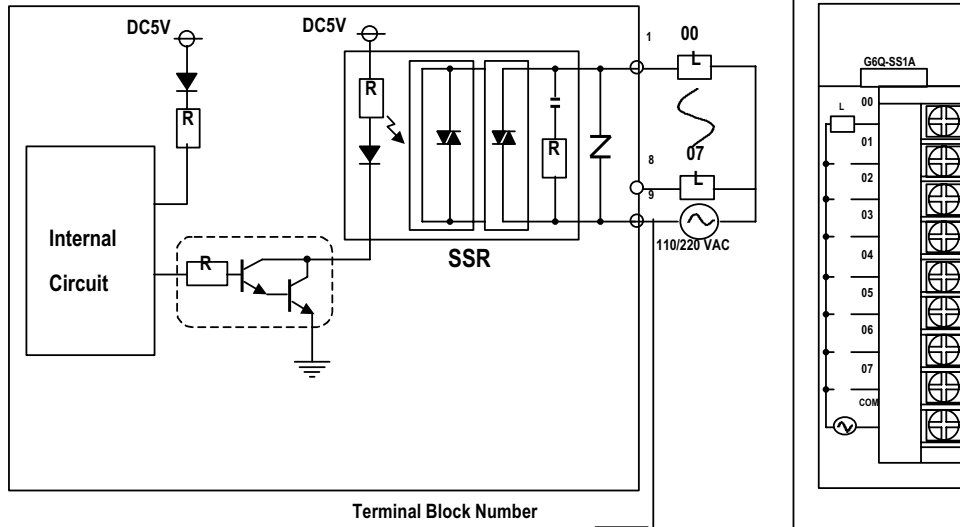
Chapter 8.   POWER SUPPLY MODULE

This chapter describes the selection method, type and specifications of the power supply module.

8.1   Selection of power supply module

Selection of the power supply module is determined by the total current consumption of digital input modules, special modules and communications modules, etc. whose powers are supplied by the power supply module.

If total load overrun the rated output capacity, the system will not normally operate. When configuring a system, select a power supply module with due consideration of current consumption of each module.

1) Current consumption GM6 series modules (unit: mA)

Modules	Models	Current Consumption	Modules	Models	Current Consumption
CPU module	GM6-CPUA	100	A/D conversion module	G6F-AD2A	50
24 VDC input module	G6I-D22A	70	D/A conversion module	G6F-DA2V	50
	G6I-D22B	70		G6F-DA2I	50
	G6I-D24A	75	High speed counting module	G6F-HSCA	300
	G6I-D24B	75	Computer link module	G6L-CUEB	108
110 VAC input module	G6I-A11A	41		G6L-CUEC	138
220 VAC input module	G6I-A21A	41	Fnet I/F module	G6L-FUEA	182
Relay output module	G6Q-RY2A	415			
Transistor output module	G6Q-TR2A	185			
	G6Q-TR4A	139			
Triac output module	G6Q-SS1A	210			

8.2 Specifications

	Item	GM6-PAFA	GM6-PAFB
Input	Input voltage	85 to 264 VAC	85 to 264 VAC
	Input frequency	50 / 60 Hz (47 to 63 Hz)	
	Input current	0.7 / 0.35 A	0.7 / 0.35 A
	Inrush current	30 A or less	
	Efficiency	70% or more (rated load, 110/220 VAC)	
	Input fuse	250 VAC / 2A	
	Allowable momentary power failure	20 ms or less	
Output	Output voltage	5 VDC	
	Output current	5 VDC : 2 A 24 VDC : 0.3 A	5 VDC : 2 A +15 VDC : 0.5 A -15VDC : 0.2 A
	Over-current protection	5 VDC : 2.2 A or more 24 VDC : 0.33 A or more	5 VDC : 2.2 A +15 VDC : 0.55 A -15VDC : 0.22 A
Voltage status indicator		LED turns On at normal output voltage.	
Used wire specifications		0.75 to 2 mm^2	
Weight		0.32 kg	

REMARK

To use A/D and D/A modules (G6F-AD2A, G6F-DA2V, G6F-DA2I), choose the GM6-PAFB power module.

8.3 Names of Parts

The followings describe names of parts and their purposes of the power supply module.
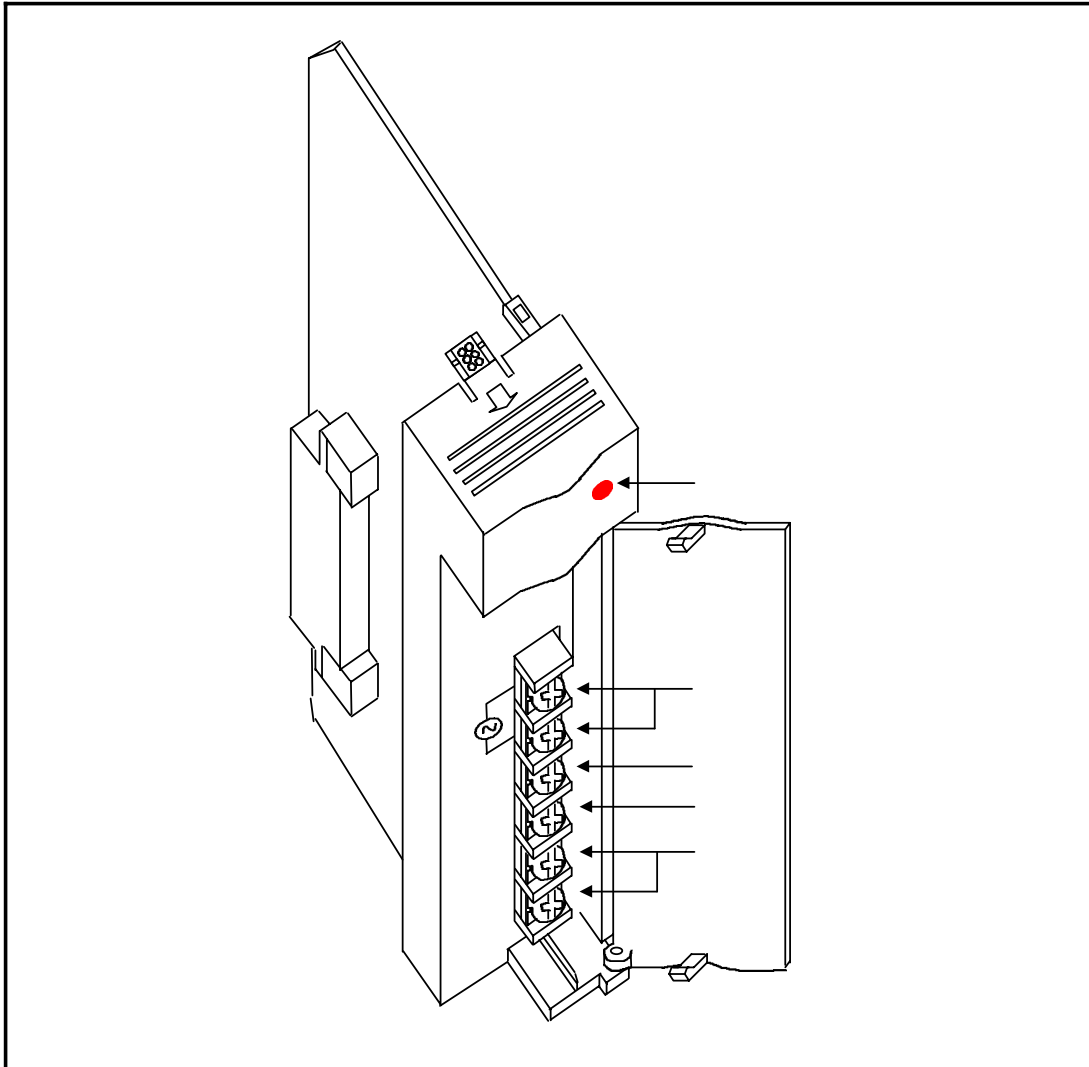
No.	Name	Purpose		
1	Power LED	It used to indicate the 5 VDC power supply.		
2	Power input terminal	It used to connect 110 or 220 VAC power.		
3	LG terminal	Line Ground		
4	FG terminal	Frame Ground		
5	24 VDC and DC24G terminal	GM6-PAFA	It used the 24 VDC power to supply to the other module	
	No connection	GM6-PAFB	No connection	

Chapter 9.   BASE BOARD

9.1   Specifications

1) GM3

Models\ Items	GM6-B04M	GM6-B06M	GM6-B08M
Mounting I/O modules	4 modules	6 modules	8 modules
Outer dimensions (mm)	$244 \times 110 \times 62$	$314 \times 110 \times 62$	$384 \times 110 \times 62$
Panel installation hole size	$\phi4.5$ (for M4 screw)		
Weight (kg)	0.24	0.35	0.75

9.2 Names of Parts
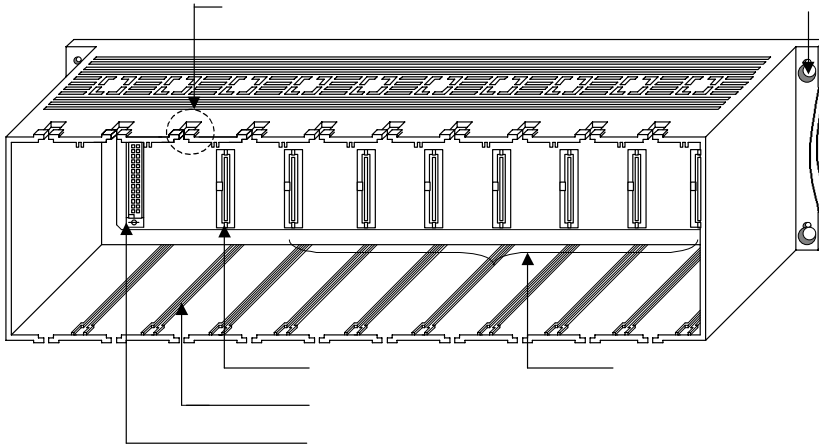
Chapter 10.   INSTALLATION AND WIRING

10.1   Installation

10.1.1   Installation Environment

This unit has high reliability regardless of its installation environment, but be sure to check the following for system reliability and stability.

1)   Environment requirements

Avoid installing this unit in locations which are subjected or exposed to :

(1) Water leakage and dust.

(2) Continuous shocks or vibrations.

(3) Direct sunlight.

(4) Dew condensation due to rapid temperature change.

(5) Higher or lower temperatures outside the range of 0 to 55 °C

(6) Relative humidity outside the range of 5 to 95 %

(7) Corrosive or flammable gases

2)   Precautions during installing.

(1) During drilling or wiring, do not allow any wire scraps to enter into the PLC.

(2) Install it on locations that are convenient for operation.

(3) Make sure that it is not located on the same panel that high voltage equipment located..

(4) Make sure that the distance from the walls of duct and external equipment be 50 mm or more.

(5)Be sure to be grounded to locations that have good ambient noise immunity.

3)   Heat protection design of control box

(1) When installing the PLC in a closed control box, be sure to design heat protection of control box with consideration of the heat generated by the PLC itself and other devices.

(2)It is recommended that filters or closed heat exchangers be used.

The following shows the procedure for calculating the PLC system power consumption.

1) PLC system power consumption block diagram
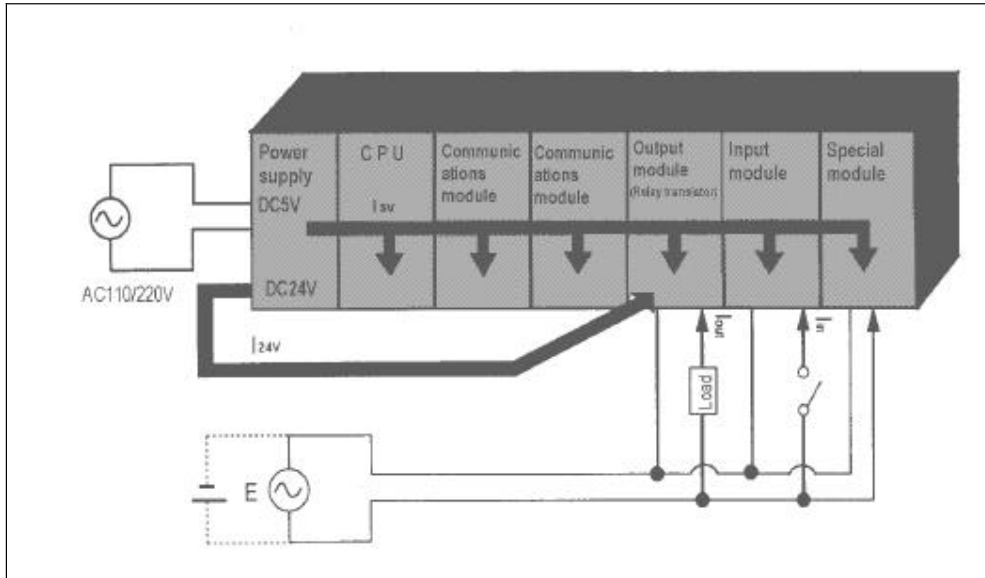
2) Power consumption of each part

(1) Power consumption of a power supply module

Approximately 70% of the power supply module current is converted into power and 30% of that 70% dissipated as heat, i.e., 3/7 of the output power is actually used.

- $Wpw = 3/7\ \{(I_{5V} \times 5) + (I_{24V} \times 24)\}$ (W)

where,　I_{5V} =　　5 VDC circuit current consumption of each module

I_{24V} =　　24 VDC circuit average current consumption of output modules (with points simultaneously switched ON). Not for 24 VDC power supplied from external or power supply modules that has no 24 VDC output.

(2) Total 5 VDC power consumption

The total power consumption of all modules is the power of the 5 VDC output circuit of the power supply module.

- $W_{5V} = I_{5V} \times 5$ (W)

(3) Average 24 VDC power consumption (with points simultaneously switched ON)

The total power consumption of all modules is the average power of the 24 VDC output circuit of the power supply module.

- $W_{24V} = I24_{V} \times 24$ (W)

(4) Average power consumption by voltage drop of output modules (with points simultaneously switched ON)

- $Wout = Iout \times Vdrop \times$ output points $\times$ the rate of points switched on simultaneously (W)

$\begin{cases} \text{Iout : output current (actual operating current) (A)} \\ \text{Vdrop : voltage dropped across each output load (V)} \end{cases}$

(5)   Average power consumption of input circuits if input modules (with points simultaneously switched ON)

- Win = Iin $\times$ E $\times$ input points $\times$ the rate of points switched on simultaneously (W)

$\left\{ \begin{array}{l} \text{Iin : input current (effective value for AC) (A)} \\ \text{E: input voltage (actual operating voltage) (V)} \end{array} \right.$

(6)   Power consumption of the special module power supply

- Ws = $I_{5V} \times 5 + I_{24V} \times 24 + I_{100V} \times 100$ (W)

The sum of the above values is the power consumption of the entire PLC system.

- W = $W_{PW} + W_{5V} + W_{24V} + W_{OUT} + W_{IN} + W_{S}$ (W)

Check the temperature rise within the control panel with calculation of that total power consumption(W).

The temperature rise in the control panel is expressed as:

T = W/UA [$^\circ$C]

$\left\{ \begin{array}{l} \text{W : Power consumption of the entire PLC system (obtained as shown above)} \\ \text{A : Control panel inside surface area (m2)} \\ \text{U : 6 (if the control panel temperature is controlled by a fan, etc.)} \end{array} \right.$

4 (if control panel air is not circulated)

10.1.2 Handling Instructions

To installing the temperature-measuring resistor input module, be sure to check the following:

- Do not drop it off, and make sure that strong shock should not be applied.
- Do not unload the PCB from its case. It can cause faults.
- During wiring, be sure to check any foreign matter like wire scraps should not enter into the upper side of the PLC. If any foreign matter has entered into it, always eliminate it.
- Do not load or unload the module while the power supply is being connected.

1) I/O module handling instructions

The followings explains instructions for handling or installing the input module.

(1)   I/O module specifications re-check

Re-check the input voltage for the input module. If a voltage over the maximum switching capacity is applied, it can cause faults, destruction or fire.

(2) Used wire

Select the wire with due consideration of ambient temperature and rated current. Its minimum specifications should be AWG22(0.3 mm^2) or more.

(3) Environment

When wiring the I/O module, if it locates near a device generating an cause short circuit, destruction or malfunction.

(4) Polarity

Before applying the power to a module that has polarities, be sure to check its polarities.

(5) Wiring

- Wiring I/O wires with high voltage cable or power supply line can cause malfunction or disorder.
- Be sure that any wire does not pass across during input LED (I/O status will not be clearly identified.
- If an inductive load has been connected to output module, connect parallel surge killer or diode to a load. Connect the cathode part of diode to the + part of the power supply.
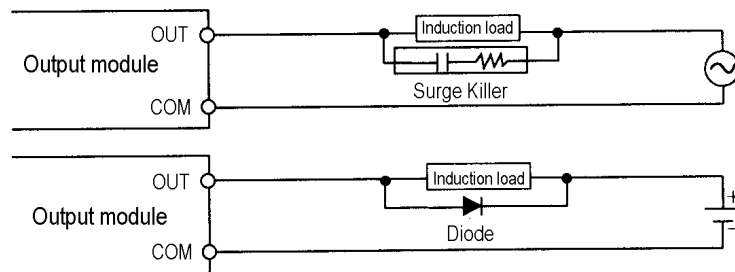
(6) Terminal block

Check its fixing. During drilling or wiring, do not allow any wire scraps to enter into the PLC. It can cause malfunction and fault.

(7) Be cautious that strong shock does not applied to the I/O module. Do not separate the PCB from its case.

2) Base board mounting instructions

The following explains instructions for mounting the PLC onto the control panel.

(1) Allow sufficient distance from the upper part of the module for easy module replacement.

(2) Do not mount the PLC in a vertical or horizontal position because it affects on ventilation.

(3) Do not mount the base board together with a large-sized electromagnetic contact or no-fuse breaker, which produces vibration, on the same panel. Mount them on different panels, or keep the base board away from such a vibration source.

(4) Mount the wire duct as it is needed.

If the clearances are less than those in Fig 10.1, follow the instructions shown below.

- If the wire duct is mounted on the upper part of the PLC, make the wiring duct clearance 50mm or less for good ventilation. Also, allow the distance enough to press the hook in the upper part from the upper part of the PLC.

- If the wire duct is mounted on the lower part of the PLC, make optic or coaxial cables contact it and consider the minimum diameter of the cable.

(5) To protect the PLC from radiating noise or heat, allow 100 mm or more clearances between it and parts. Left or right clearance and clearance from other device in the left or right side should be 50 mm or more.
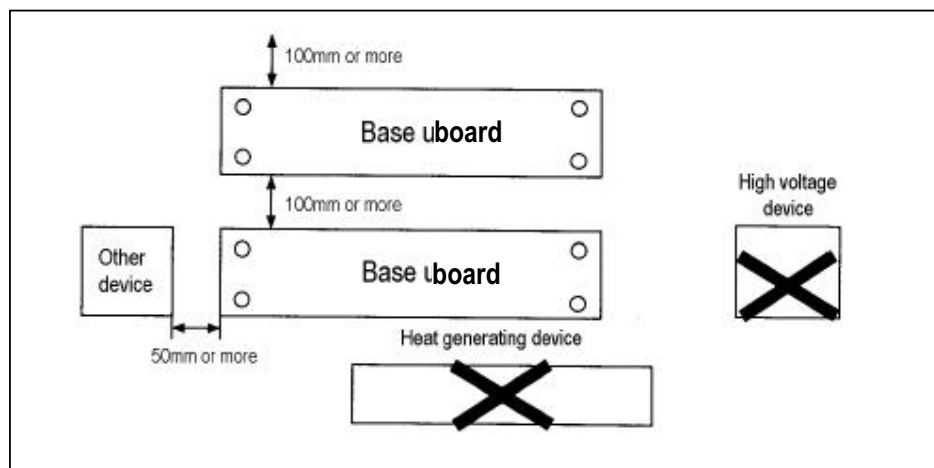
[Fig. 10.1] PLC mounting
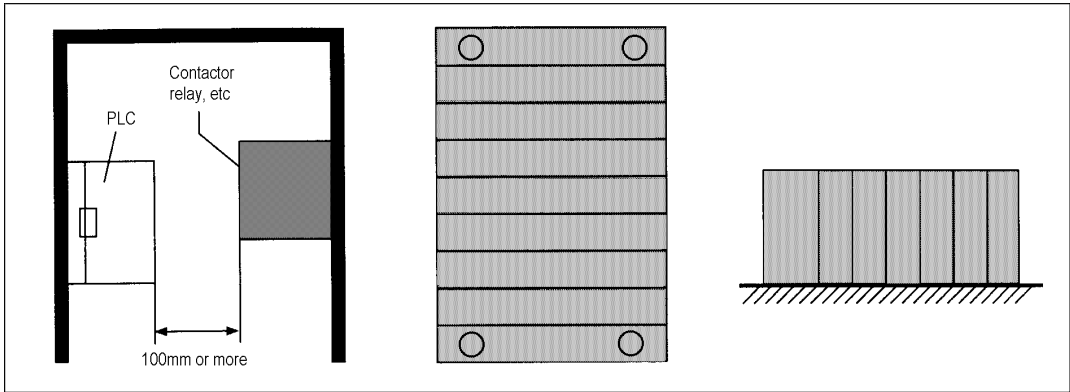
[Fig.10.2] Clearance from the front device   [Fig. 10.3] Vertical mounting   [Fig 10.4] Horizontal mounting

10.1.3   Mounting and Dismounting of module

The following explains the mounting and dismounting of various modules.

1) Module mounting

- Insert the module to mounting slot with sliding guide.
- Check that the module is firmly mounted onto the base board.

Locking part for Hook

Hook

Sliding
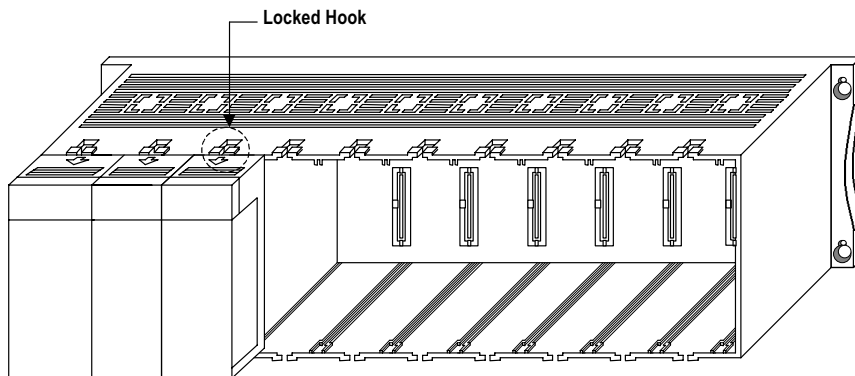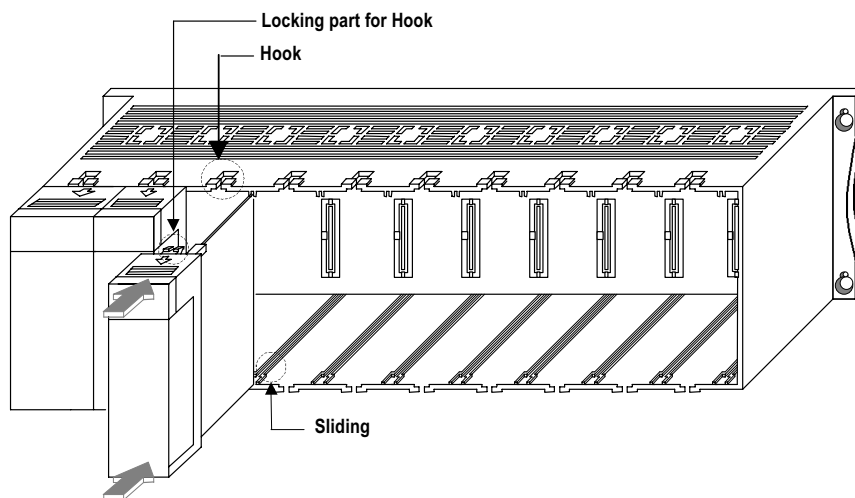
Locked Hook

2) Module dismounting

- First, push the locked hook(   ) and pull the module with direction of arrow   .
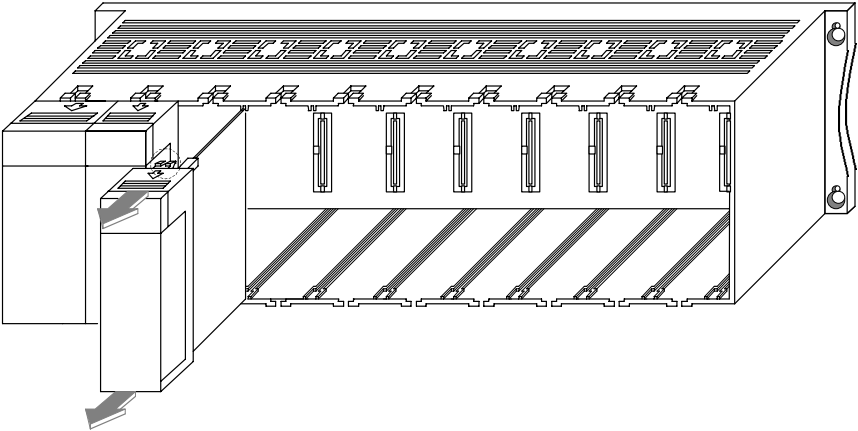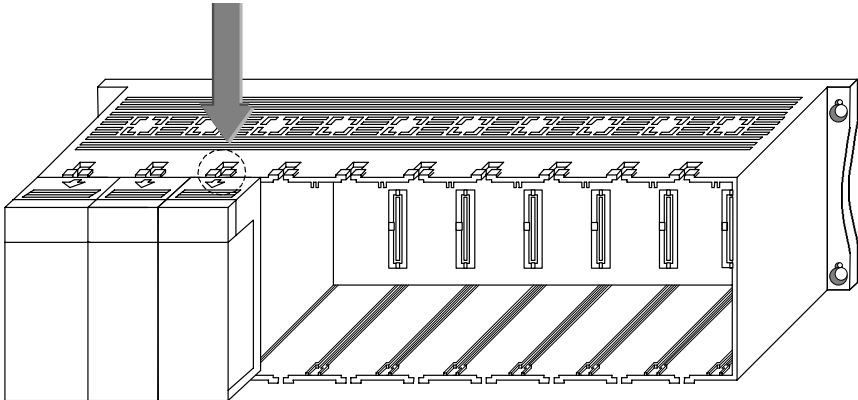
10.2 Wiring

The followings explains the wiring instructions for use of the system.

10.2.1   Power Supply Wiring

1) When voltage fluctuations are larger than the specified value, connect a constant-voltage transformer.
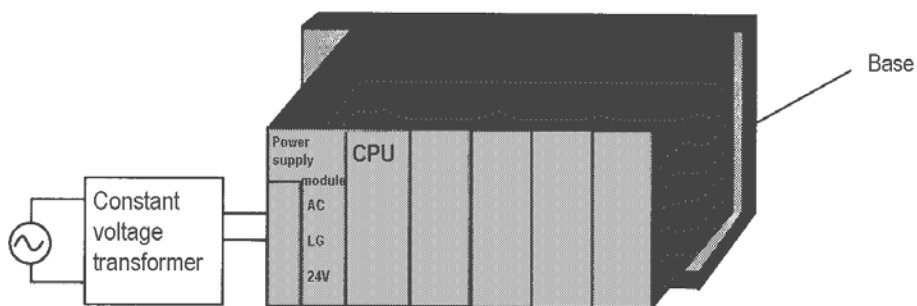
2)   Use a power supply which generates minimal noise across wire and across PLC and ground. (When excessive noise is generated, connect an insulating transformer)

3)   When wiring, separate the PLC power supply from the I/O and power device as shown below.
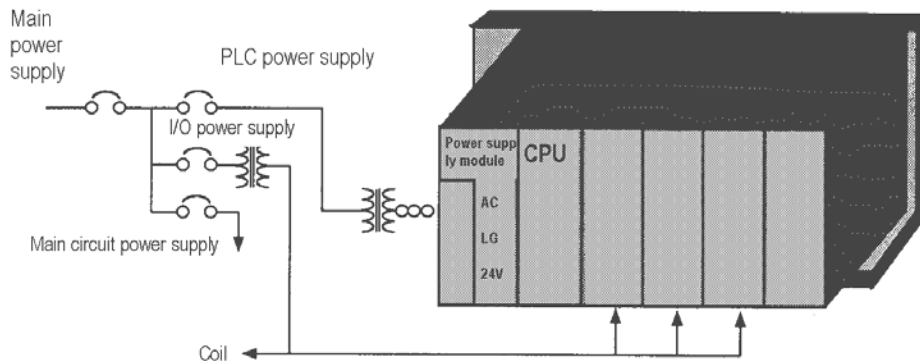
4) Notes on using 24 VDC output of the power supply module

- To protect the power supply modules, do not supply one I/O module with 24 VDDC from several power supply modules connected in parallel.

- If 24 VDC output capacity is sufficient for one power supply module, supply 24 VDC from the external 24 VDC power supply as shown below.
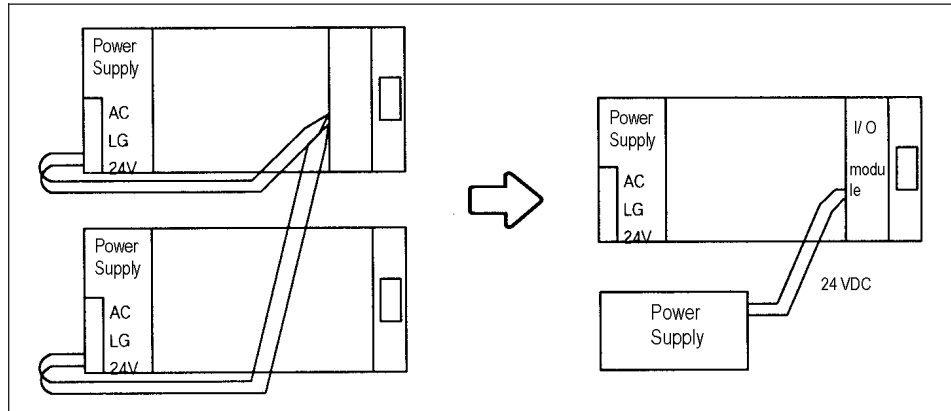
5) Twist the 110 VAC, 220 VAC, and 24 VDC cables as closely as possible. Connect modules with the shortest possible wire lengths.

7) To minimize voltage drop, use the thickest (max. 2 mm^2) wires possible for the 100 VAC, 200VAC and 24 VDC cables.

8) Do not bundles the 100 VAC and 24 VDC cables with main-circuit(high voltage, large current) wires or the I/O signal wires. If possible, provide more than 100 mm distance between the cables and wires.

8) As a lightning-protection measure, connect a surge absorber as shown below.

Surge absorber for lightening
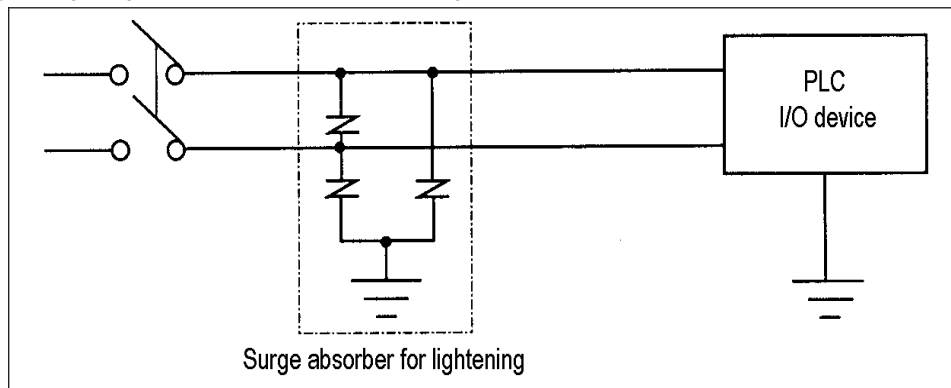
REMARK
1) Ground the surge absorber (E1) and the PLC(E2) separately from each other.
2) Select a surge absorber making allowances for power voltage rises.

9) Use a insulating transformer or noise filter for protection against noise .

10) Twist every input power supply wires as closely as possible. Do not allow the transformer or noise filter across the duct.

10.2.2 Input and Output Devices Wiring

1) Applicable size of wire for I/O wiring is 0.3 to 2 mm^2. However, it is recommended to use wire of 0.3mm^2 for convenience.

2) Separate the input and output lines.

3) I/O signal wires must be at least 100 mm away from high voltage and large current main circuit wires.

4) When the I/O signal wires cannot be separated from the main circuit wires and power wires, ground on the PLC side with batch-shielded cables.
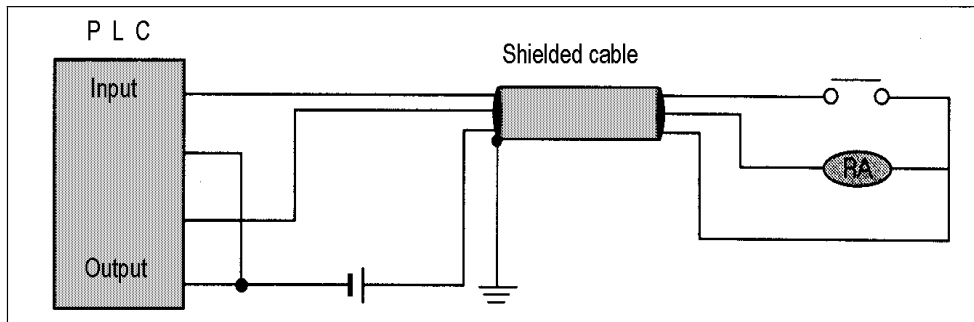
5) If wiring has been done with a piping, ground the piping.

6) Separate the 24 VDC I/O cables from the 110 VAC and 220 VAC cables.

7) If wiring over 200 m or longer distance, problems can be caused by leakage currents due to line capacity. Refer to the Section 12.4 Examples.

10.2.3 Grounding

1) This PLC has sufficient protection against noise, so it can be used without grounding except for special much noise. However, when grounding it should be done conforming to below items.

2) Ground the PLC as independently as possible. Class 3 grounding should be used (grounding resistance 100 Ω or less).

3) When independent grounding is impossible, use the joint grounding method as shown in the figure below (B).
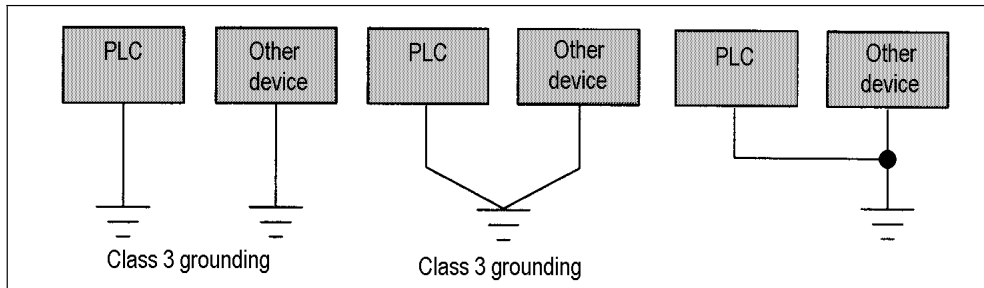
(A) Independent grounding : Best   (B) Joint grounding : Good   (C) Joint grounding : Not allowed

4) Use 2 mm^2 or more wire for grounding line. Make the distance as short as possible with the grounding point located to nearest to the PLC.

5) Ground LG (Power Supply Module) separately with FG (Base board).
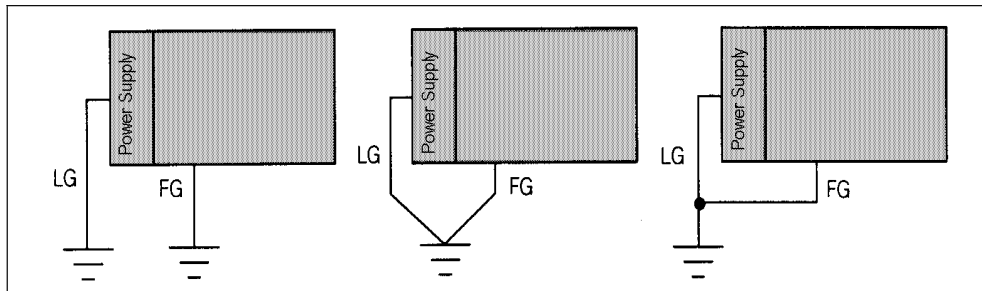
(A) Independent grounding : BEST   (B) Joint grounding : GOOD   (C) Joint Grounding : Not Allowed

6) If a malfunction occurs depend on grounding point, separate FG (Base Board) with ground.

10.2.4 Cable Specifications for wiring

Kinds of external connection	Cable Specifications ( )	
	Minimum	Maximum
Digital Input	0.18 (AWG 24)	1.5 (AWG16)
Digital Output	0.18 (AWG24)	2.0 (AWG14)
Analog Input/Output	0.18 (AWG24)	1.5 (AWG16)
Communication	0.18 (AWG24)	1.5 (AWG16)
Main Power	1.5 (AWG16)	2.5 (AWG12)
Grounding	1.5 (AWG16)	2.5 (AWG12)

Chapter 11.   MAINTENANCE

Be sure to perform daily and periodic maintenance and inspection in order to maintain the PLC in the best conditions.

11.1   Maintenance and Inspection

The I/O module mainly consist of semiconductor devices and its service life is semi-permanent. However, periodic inspection is requested for ambient environment may cause damage to the devices. When inspecting one or two times per six months, check the following items.

Check Items		Judgment	Corrective Actions
Ambient environment	Temperature	0 to +55°C	Adjust the operating temperature and humidity with the defined range
	Humidity	5 to 95%RH	
	Vibration	No vibration	Use vibration resisting rubber or the vibration prevention method
Play of modules		No play allowed	Securely enrage the hook
Connecting conditions of terminal screws		No loose allowed	Retighten terminal screws
Change rate of input voltage		-15% to 15%	Hold it with the allowable range
Spare parts		Check the number of spare parts and their storage conditions	Cover the shortage and improve the storage condition

11.2   Daily Inspection

The following table shows the inspection and items which are to be checked daily

Check Items		Check points	Judgment	Corrective Actions
Base unit mounting conditions		Check for loose mounting screws	The base unit should be securely mounted	Retighten Screws
Mounting conditions of I/O modules		• Check if the hook is securely engaged • Check if the upper cover is securely mounted	The hook should be securely engaged	Securely engage the hook
Connecting conditions of terminal block or extension cable		Check for loose terminal screws	Screws should not be loose	Retighten terminal screws
		Check the distance between solderless terminals	Proper clearance should be provided	Correct
		Check connectors of extension cable	Connectors should not be loose	Correct
Indic atin g LED	Power LED	Check that the LED is ON	ON(OFF indicates an error)	See chapter 12
	Run LED	Check that the LED is ON during Run	ON(ON or flickering indicates an error)	"
	Stop LED	Check that the LED is OFF during Run	OFF(ON indicates an error)	"
	Input LED	Check that the LED turns ON and OFF	ON when input is ON, OFF when input is off	"
	Output LED	Check that the LED turns ON and OFF	ON when output is ON. OFF when output is OFF	"

11.3   Periodic Inspection

Check the following items once or twice every six months, and perform the needed corrective actions.

Check Items		Checking Methods	Judgment	Corrective Actions
Ambient environment	temperature	Measure with thermometer and hygrometer Measure corrosive gas	0 to 55°C	
	Ambient humidity		5 to 95% RH	
	Ambience		There should be no corrosive gases	
PLC conditions	Looseness, play	Move the unit	The module should be mounted securely	Retighten screws
	Ingress of dust or foreign material	Visual check	No dust or foreign material	
Connecting conditions	Loose terminal screws	Re-tighten	Screws should not be loose	Retighten
	Distance between terminals	Visual check	Proper clearance	Correct
	Loose connector	Visual check	Connectors should not be loose	Retighten connector mounting screws
Line voltage check		Measure voltage across 110/ 220 VAC terminal	85 to 132VAC 170 to 264VAC	Change supply power
Battery		Check battery replacement time and battery capacity reduction	• Check total power failure time and the specified source life • Battery capacity reduction should not be indicated	If battery capacity reduction is not indicated, Change the battery when specified service life is exceeded
Fuse		Visual check	No melting disconnection	If fuse melting disconnection, change the fuse periodically because a surge current can cause heat

Chapter 12. TROUBLE SHOOTING

The following explains contents, diagnosis and corrective actions for various errors that can occur during system operation

12.1   Basic Procedures of Troubleshooting

System reliability not only depends on reliable equipment but also on short down-times in the event of faults.

The short discovery and corrective action is needed for speedy operation of system.

The following shows the basic instructions for troubleshooting.

1) Visual checks

Check the following points

- Machine motion(In stop and operating status)
- Power ON or OFF
- Status of I/O devices
- Condition of wiring (I/O wires, extension and communications cables)
- Display states of various indicators(such as POWER LED, RUN LED, STOP LED and I/O LED). After checking them, connect the peripheral devices and check the operation status of the PLC and the program contents.

2) Trouble Check

Observe any change in the error conditions during the following.

- Set the mode setting switch to the STOP position, and then turn the power ON and OFF

3) Narrow down the possible causes of the trouble

Deduce where the fault lies, i. e:

- Inside or outside of the PLC
- I/O module or another module
- PLC program ?

12.2 Troubleshooting

This section explains the procedure for determining the cause of troubles as well as the errors and corrective actions for the error codes.

Occurrence of error	
Is the power LED turned OFF?	Flowchart used when the POWER LED is turned OFF
Is the stop LED flickering ?	Flowchart used when the STOP LED is flickering
Are the RUN and STOP LED turned OFF?	Flowchart used when the RUN and STOP LED is turned OFF
I/O module dose not operate properly	Flowchart used when the output load of the output module dose not turn on
Program cannot be written	Flowchart used when a program cannot be written to the PLC

12.2.1   Troubleshooting flowchart used when the POWER LED turns OFF.

The following flowchart explains corrective action procedure used when the power is all lied or the POWER
LED turns OFF during operation

```
                    ┌─────────────────────────┐
                    │  Power LED is turned OFF │
                    └─────────────────────────┘
                                 │
                                 ▼
                          ╱──────────╲          No      ┌──────────────────────┐
                         ╱ Is the power ╲────────────────▶│ Apply the power supply│
                         ╲ supply operating╱              └──────────────────────┘
                          ╲──────────╱                                │
                               │Yes                                   ▼
                               │            No          ╱──────────╲         Yes
                               │◀───────────────────────╱ Does the Power╲──────────▶
                               ▼                         ╲ LED turn ON? ╱
                          ╱──────────╲          No      ┌──────────────────────────┐
                         ╱ Is the line  ╲────────────────▶│ See the supply power to  │
                         ╲ voltage 85 to  ╱                │  within the rated power  │
                         ╲ 264VAC?      ╱                └──────────────────────────┘
                          ╲──────────╱                                │
                               │Yes                                   ▼
                               │            No          ╱──────────╲         Yes
                               │◀───────────────────────╱ Does the Power╲──────────▶
                               ▼                         ╲ LED turn ON? ╱
                          ╱──────────╲          Yes     ┌──────────────────────┐
                         ╱ Is Fuse      ╲────────────────▶│   Replace the fuse   │
                         ╲ disconnected?╱                └──────────────────────┘
                          ╲──────────╱                                │
                               │No                                    ▼
                               │            No          ╱──────────╲         Yes
                               │◀───────────────────────╱ Does the Power╲──────────▶
                               ▼                         ╲ LED turn ON? ╱
                          ╱──────────╲          No      ┌──────────────────────────────┐
                         ╱ Is the power ╲────────────────▶│ Fix the power supply module   │
                         ╲ supply module  ╱               │          correctly           │
                         ╲ Fixed to the base?╱            └──────────────────────────────┘
                          ╲──────────╱                                │
                               │Yes                                   ▼
                               │            No          ╱──────────╲         Yes
                               │◀───────────────────────╱ Does the Power╲──────────▶
                               ▼                         ╲ LED turn ON? ╱
                          ╱──────────╲          Yes     ┌──────────────────────────────┐
                         ╱ Over current ╲────────────────▶│1) Eliminate the excess current│
                         ╲ protection     ╱               │2) Switch the input power OFF, │
                         ╲ Device activated?╱             │   then ON                    │
                          ╲──────────╱                    └──────────────────────────────┘
                               │No                                    │
                               │            No          ╱──────────╲         Yes
                               │◀───────────────────────╱ Does the Power╲──────────▶
                               ▼                         ╲ LED turn ON? ╱
            ┌──────────────────────────────────┐
            │ Write down the troubleshooting   │         ┌──────────┐
            │ questionnaires and contact the   │         │ Complete │
            │ nearest service center           │         └──────────┘
            └──────────────────────────────────┘
```
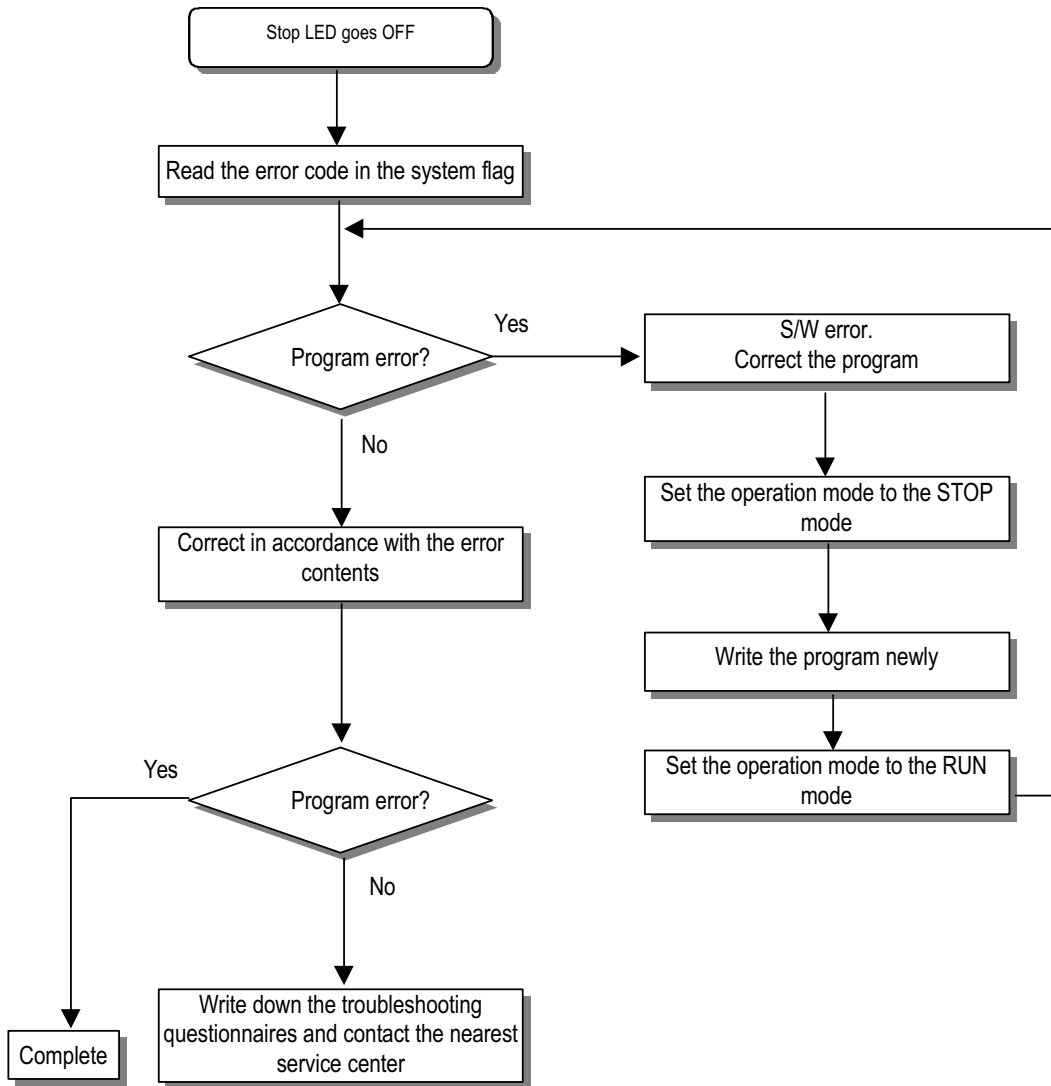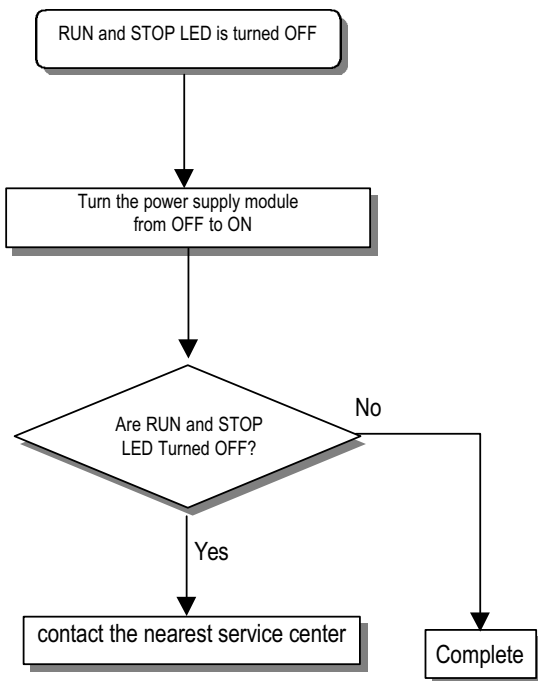
12.2.2   Troubleshooting flowchart used when the STOP LED is flickering

The following flowchart explains corrective action procedure use when the power is applied starts or the STOP LED is flickering during operation

```
                    ┌─────────────────────────┐
                    │    Stop LED goes OFF     │
                    └─────────────────────────┘
                                 │
                                 ▼
          ┌──────────────────────────────────────┐
          │  Read the error code in the system flag │
          └──────────────────────────────────────┘
                                 │                                    ◄─────────────┐
                                 ▼                                                  │
                             ╱───────╲          Yes      ┌─────────────────────┐    │
                            ╱  Program  ╲ ─────────────►  │    S/W error.       │    │
                            ╲  error?  ╱                  │  Correct the program │    │
                             ╲───────╱                    └─────────────────────┘    │
                                 │ No                              │                  │
                                 ▼                                 ▼                  │
          ┌──────────────────────────────┐          ┌─────────────────────────┐     │
          │  Correct in accordance with   │          │ Set the operation mode to │    │
          │      the error contents       │          │    the STOP mode         │     │
          └──────────────────────────────┘          └─────────────────────────┘     │
                                 │                                 │                  │
                                 │                                 ▼                  │
                                 │                      ┌─────────────────────────┐   │
                                 │                      │  Write the program newly │   │
                                 │                      └─────────────────────────┘   │
                                 ▼                                 │                  │
         Yes                 ╱───────╲                             ▼                  │
    ◄────────────────────── ╱  Program  ╲          ┌─────────────────────────┐       │
                            ╲  error?  ╱           │ Set the operation mode to │──────┘
                             ╲───────╱            │    the RUN mode          │
                                 │ No              └─────────────────────────┘
                                 ▼
    ┌──────────┐   ┌──────────────────────────────────┐
    │ Complete │   │  Write down the troubleshooting   │
    └──────────┘   │ questionnaires and contact the    │
                   │      nearest service center       │
                   └──────────────────────────────────┘
```
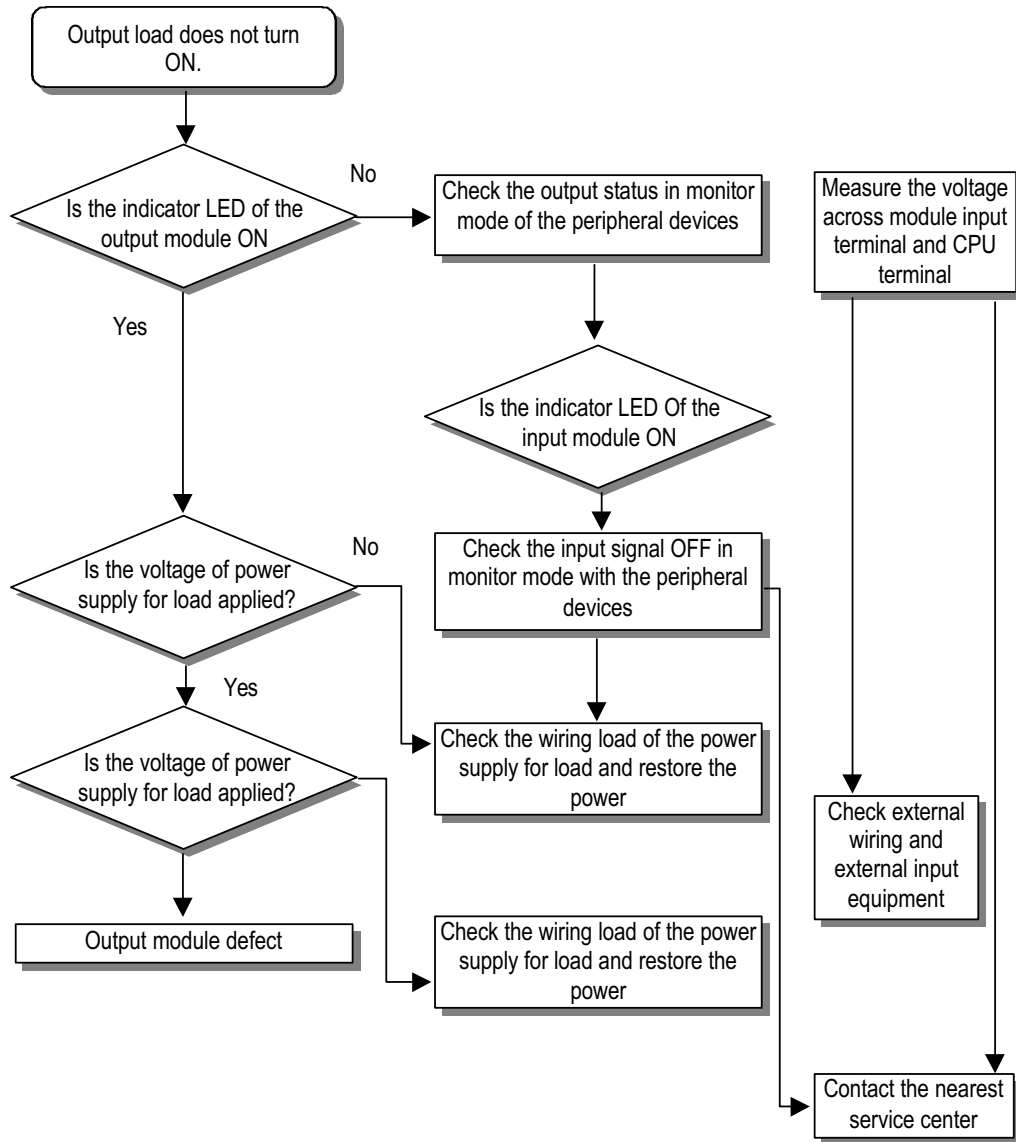
12.2.3   Troubleshooting flowchart used when the RUN and STOP LEDs turns off.

The following flowchart explains corrective action procedure use when the power is applied starts or the RUN and STOP LED is turned OFF is flickering during operation

```
          ┌──────────────────────────────────┐
          │  RUN and STOP LED is turned OFF   │
          └──────────────────────────────────┘
                          │
                          ▼
          ┌──────────────────────────────────┐
          │   Turn the power supply module    │
          │         from OFF to ON            │
          └──────────────────────────────────┘
                          │
                          ▼
                    ◇ Are RUN and STOP        No
                    ◇ LED Turned OFF?   ─────────────┐
                          │                          │
                         Yes                         │
                          ▼                          ▼
     ┌──────────────────────────────┐        ┌──────────────┐
     │ contact the nearest service  │        │   Complete   │
     │           center             │        └──────────────┘
     └──────────────────────────────┘
```

12.2.4 Troubleshooting flowchart used when the output load of the output module does not turns on.

The following flowchart explains corrective action procedure used when the output load of the output module does not turn ON during operation

```
              ┌────────────────────────┐
              │  Output load does not turn  │
              │           ON.           │
              └────────────────────────┘
                          │
                          ▼
                                        No    ┌────────────────────────┐      ┌────────────────────────┐
         ◇ Is the indicator LED of the  ──────▶│ Check the output status in monitor │   │ Measure the voltage │
         ◇    output module ON                 │ mode of the peripheral devices │      │ across module input │
                                               └────────────────────────┘      │ terminal and CPU terminal │
                  │ Yes                                    │                    └────────────────────────┘
                                                           ▼
                                               ◇ Is the indicator LED Of the
                                               ◇    input module ON
                                                           │
                  │                                        ▼
                                        No     ┌────────────────────────┐
         ◇ Is the voltage of power   ──────────▶│ Check the input signal OFF in │
         ◇ supply for load applied?            │ monitor mode with the peripheral │
                                               │        devices          │
                  │ Yes                                    │
                                                           ▼
         ◇ Is the voltage of power              ┌────────────────────────┐      ┌────────────────────────┐
         ◇ supply for load applied?            │ Check the wiring load of the power │  │ Check external wiring and │
                                               │ supply for load and restore the │    │ external input equipment │
                  │                            │        power            │      └────────────────────────┘
                  ▼                            └────────────────────────┘
         ┌────────────────────────┐           ┌────────────────────────┐
         │  Output module defect   │           │ Check the wiring load of the power │
         └────────────────────────┘           │ supply for load and restore the │
                                               │        power            │
                                               └────────────────────────┘
                                               ┌────────────────────────┐
                                               │ Contact the nearest service center │
                                               └────────────────────────┘
```

REMARK

1) If the input or load signals are not switched OFF, see Section 12.4.1

12.2.5   Troubleshooting flowchart used when a program cannot be written to the CPU module.

The following flowchart shows the corrective action procedure used when a program cannot be written to the PLC module

```
         ┌────────────────────────────────────────┐
         │  Program cannot be written to the PC CPU │
         └────────────────────────────────────────┘
                           │
                           ▼
              ◇ Is the mode setting switch      No     ┌──────────────────────────────┐
                set the remote STOP          ────────▶ │ Set the mode setting switch to the │
                mode ◇                                 │ remote STOP mode and execute   │
                           │                           │ the program write              │
                          Yes                          └──────────────────────────────┘
                           │
                           ▼
              ◇ Is the STOP LED                Yes     ┌──────────────────────────────┐
                flickering? ◇               ────────▶ │ Read the error code using the  │
                                                       │ peripheral devices and correct the │
                                                       │ contents.                      │
                                                       └──────────────────────────────┘
```
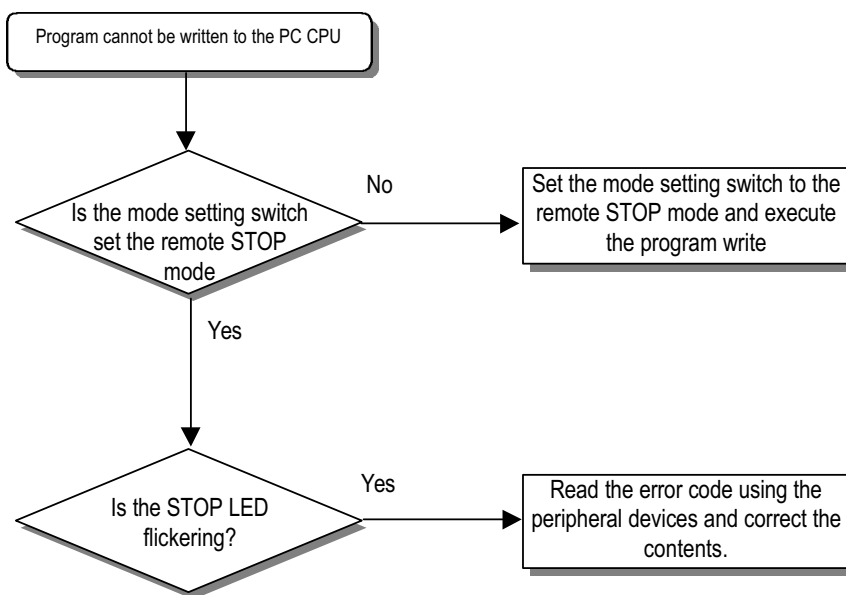
12.3   Troubleshooting Questionnaire

When problems have been met during operation of the GM6 series PLC, please write down this questionnaires and contact the service center via telephone or facsimile

• For errors relating to special or communications modules, use the questionnaire included in the user's Manual of the unit

1. Telephone & FAX No.   Tel) _____

                                    FAX) _____

2. Used Equipment   (   )

3. Details of used Equipment

   - CPU module : - OS version No.(   ),   - Serial No.(   )

   - GMWIN version No. used to compile programs

4. General description of the device or system used as the control object

5. Operations used by the CPU module

   - Operation by the mode setting switch(   ),

   - Operation by the GMWIN or communications.(   )

6. Is the STOP LED of the CPU module turned ON?   Yes(   ),   No(   )

7. GMWIN error message :

8. Used initialization program : initialization program(   )

9. History of corrective actions for the error message in the article 7.

10. Other tried corrective actions

11. Error character sties

   • Repetitive(   ) : Periodic(   ), Related to a particular sequence(   ), Related to environment(   )

   • Sometimes(   ) : General error assurance interval

12. Detailed Description of error contents :

13. Configuration Diagram for the applied system :

12.4   Troubleshooting Examples

Possible troubles with various circuits and their corrective actions are explained.
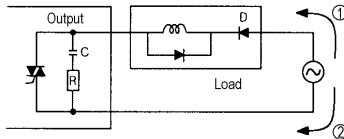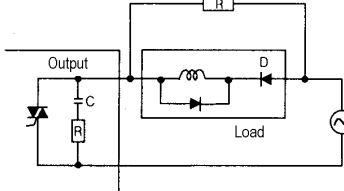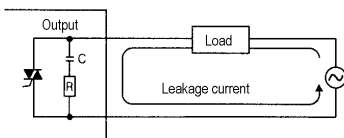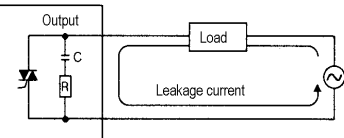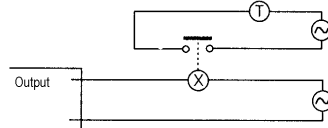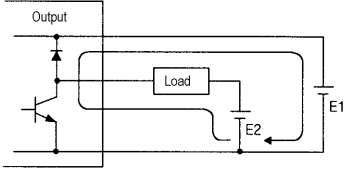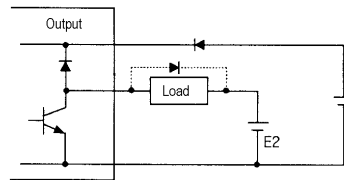
12.4.1   Input circuit troubles and corrective actions

The followings describe possible troubles with input circuits, as well as corrective actions.
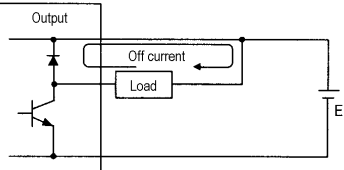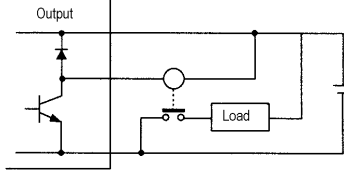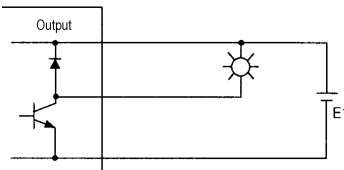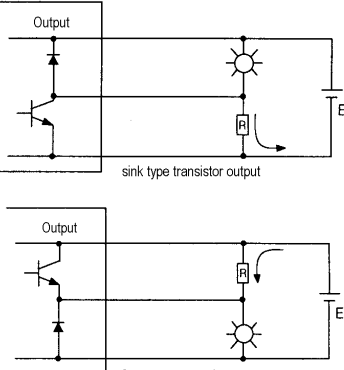
Condition	Cause	Corrective Action
Input signal close not turn OFF	Leakage current of external device (such as a drive by non-contact switch)	• Connect an appropriate register and capacity which will make the voltage across the terminals of the input module lower than
Input signal does not turn OFF	Leakage current of external device (Drive by a limit switch with neon lamp)	• C and R values are determined by the leakage current value - Reminded value C : 0.1 ~ 0.47 R : 47 ~ 120Ω (1/2W) Or make up another independent display circuit
Input signal does not turn OFF	Leakage current due to line capacity of wiring cable	• Power supply is located on the external device side as shown below
Input signal does not turn OFF	Leakage current of external device (Drive by switch with LED indicator)	• Connect an appropriate register which will make the voltage across input module terminal and common higher than the OFF voltage, as shown below
Input signal does not turn OFF	• Sneak current due to the use of two different power supplies • E1 > E2, Sneaked	• Use only one power supply • Connect a sneak current prevention diode, as shown below

12.4.2   Output circuit troubles and corrective actions

The following desires possible troubles with output circuits, as well as corrective actions

Condition	Cause	Corrective Action
When the output is Off, excessive voltage is applied to the load	• Load is half-wave rectified inside (in some cases, it is true of a solenoid) • When the polarity of the power supply is as shown in ①, C is charged. When the polarity is as shown in②, the voltage charged in C plus the line voltage are applied across D. Max voltage is approx. If a resistor is used in this way, it does not pose a problem to the output element. But it may make the performance of the diode(D), which is built in the load, drop to cause problems	• Connect registers of tens to hundreds kΩ across the load in parallel
The load does not turn OFF	• Leakage current by surge absorbing circuit which is connected to output element in parallel 	• Connect C and R across the load, which are of registers of tens kΩ When the wiring distance from the output module to the load is long, there may be a leakage current due to the line capacity
When the load is C-R type timer, time constant fluctuates	• Leakage current by surge absorbing circuit which is connected to output element in parallel 	• Drive the relay using a contact and drive the C-R type timer using the since contact • Use other timer than the C-R contact Some timers have half-ware rectified internal circuits therefore, be cautious.
The load does not turn OFF	• Sneak current due to the use of two different power supplies • E1 < E2 : sneak current • E1 is switched Off and E2 is switched ON : sneak current	• Use only one power supply • Connect a sneak current prevention diode(Figure below) If the load is the relay, etc, connect a counter-electromotive voltage absorbing code as show by the dot line

Output circuit troubles and corrective actions(continued)

Condition	Cause	Corrective Action
The load off response time is long	• Over current at Off state [The large solenoid current fluidic load (L/R is large) such as is directly driven with the transistor output • The off response time can be delayed by one or more second as some loads make the current flow across the diode at the off time of the transistor output	• Insert a small L/R magnetic contact and drive the load using the same contact
Output transistor is destroyed	Surge current of the white lamp A surge current of 10 times or more when turned ON.	• To suppress the surge current make the dark current of 1/3 to 1/5 rated current flow sink type transistor output Source type transistor output

12.5   Error code list

Error code	Cause	Corrective Action	Operation status	STOP LED Flickering cycle	Diagnosis time	Re-start mode
2	OS ROM error	Contact the service center if it reactively occurs when the power is re-applied.	Defect	0.4 sec	When power is applied	-
3	OS ROM error	"	Defect	0.4 sec	When power is applied	-
4	RTC fault	"	Defect	0.4 sec	When power is applied	-
5	Dedicated processor fault	"	Defect	0.4 sec	When power is applied	-
6	Program memory fault	"	Defect	0.4 sec	When power is applied	-
7	Data memory fault	"	Defect	0.4 sec	When power is applied	-
10	Watch dog error due to OS program congestion	RE-apply the power	Reset	0.4 sec	During run	Cold
20	Program memory backup error	Replace the battery if it has error check the program after cc-loading it, and if an error is detected replace the CPU module	STOP	0.4 sec	When power is applied	Cold
21	Memory module defect	Check and correct the memory module mounting condition Re-apply the power and if an error occurs, replace the memory module	STOP	0.4 sec	When power is applied	Cold
22	Memory module program fault	Correct the memory module program and re-operate the system	STOP	0.4 sec	Change into the RUN mode	Cold
23	An normal program	Re-load the program and start it	STOP	0.4 sec	Change into the RUN mode	Cold
30	Inconsistency between the specified modules by parameters and the loaded modules	Module type inconsistency error Refer to the flags(_IO_TYER,_IO_ DEER_N, _IO_TYER [n]) and correct the incorrective slot, and re-start the system	STOP	0.4 sec	Change into the RUN mode	Cold
31	Module dismounting or additional mounting during run	Module mounting/ dismounting error Refer to the flags(_IO_DEER,_IO_ DEER_N, IO_DEER [n]) and correct the in corrective slot, and re-start the system	STOP	0.4 sec	When scan completes	Cold
32	Fuse disconnection during run	Fuse disconnection error Refer to the flags(_FUSE_ER,_FUSE _ER_N, FUSE_ER[n]) and correct the in corrective slot, and re-start the system	STOP	0.4 sec	When scan completes	Cold
33	Abnormal I/D module data access during run	I/O module read/ write error Refer to the flags(IO_RWER, _IP_RWER_N, _IO_RWER [n]) and restart the system	STOP	0.4 sec	When scan completes During execution of program	cold
34	Abnormal special/ link module data access during run	Special/ link module interface error Refer to the flags(_SP_IFER,_IP_ IFER_N,_IP_IFER [n]) and restart the system	STOP	0.4 sec	When power is applied When scan completes During execution of program	cold
40	During run, Scan time over than the scan delay time specified by parameters	Check the scan delay time specified by parameters and correct the parameters or the program, and then re-start the program	STOP	0.4 sec	During execution of program	cold
41	Unreadable instructions in the user program	Re-load the program and re-start it	STOP	0.4 sec	During execution of program	cold
50	External device fatal error	Refer to the external device fatal error flags(_ANNUN_ER,_ANC_ERR[n]) and correct the fault devices and then re-start the system	STOP	0.4 sec	When scan completes	cold
60	The 'E-STOP' function has been executed	Correct the program so that the error elements that invoked the 'E_STOP' function can be eliminated in the program and re-start the system(Cold re-start)	STOP	-	During execution of program	cold
100	Communications module configuration error	If the number of computer 4communications module is included, then adjust the maximum number with in 8	STOP	0.4 sec	When power is applied	cold
101	Special/ Communications module initialization failure	Adjust the number of high speed communications modules loaded	STOP	0.4 sec	When power is applied	cold
500	Data memory backup error	If the battery has no error	RUN	-	When power is applied	cold
501	RTC data error	If the battery has no error, re-set the time using the GMWIN	RUN	2 sec	When power is applied When scan completes	-
502	Lower battery voltage	Replace the battery which the power is being applied.	RUN	4 sec	When power is applied When scan completes	-

Chapter 13. Dedicated Cnet communication for GM6

13.1   Introduction

The GM6 CPU module provides some basic Cnet communication functions without Cnet module. Although all functions of Cnet module are not supported, it will be very useful functions for users to perform simple Cnet communication. If your needs are read/write variables (I,Q,M devices) and Monitoring, you don't need to buy Cnet module. It will save your money and slot for Cnet module.

The Cnet functions provided by CPU (A-type) module are as following;

- Individual read instruction
- Continuous read instruction
- Individual write instruction
- Continuous write instruction
- Monitoring variables registration
- Monitoring execution
- 1:1 communication only (dedicated protocol)
- RS-232 communication only

> **Note**   Because the dedicated Cnet communication for GM6 does not support all functions of Cnet module, there are some limitations as following comparison with using Cnet module.
>
> 1) At the pressing time, the RS-422 protocol is not supported. Only RS-232C protocol is supported. (RS422 protocol will be available with the next version of GM6 CPU.)
>
> 2) Only the 1:1 communication is available. The 1:N communication (multi-drop) which have Master & Slave station will be available with the next version of GM6 CPU.
>
> 3) Because the GM6 CPUA module has only one serial port supports RS-232C, the general RS-232C cable can not be used. Also the cable for Cnet module can not be used with GM6 CPUA module. See the chapter 13.3 of this manual for the detailed pin assign for GM6 CPUA module.

13.2 The example of system configuration

Generally, the system configuration have two types; the 1:1 communication with PC and the connection with monitoring device (like PMU).

The configuration when connected to PC

   : With this configuration, the communication program of PC can be a user's own program (written in C or other programming language) or a commercial software like FAM or CIMON.
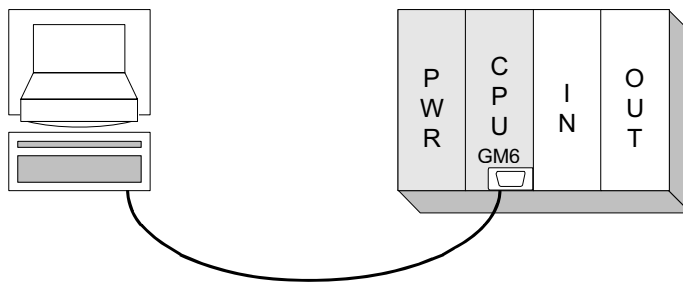
(Cnet connection with PC)

The configuration when connected to PMU
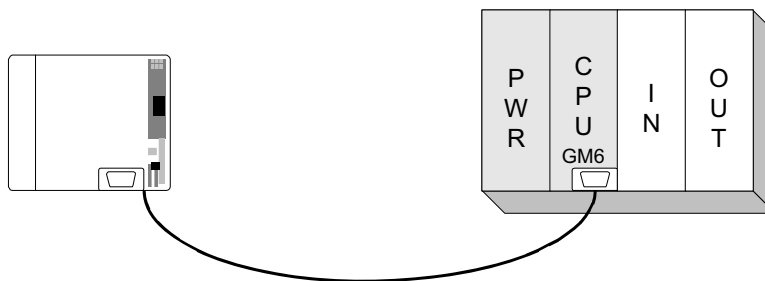
(Cnet connection with PMU)

13.3 The pin assignment of RS-232C connector of the GM6 dedicated Cnet communication

The 1:1 connection with PC

<The pin assignment of RS232C connector which are used the connection of PC and GM6 CPU>
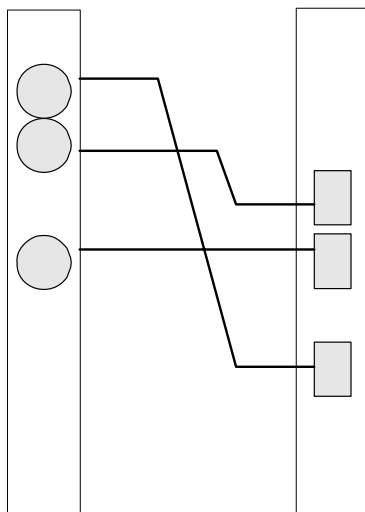
The 1:1 connection with the monitoring unit like PMU

<The pin assignment of RS232C connector which are used the connection of PMU and GM6 CPU>
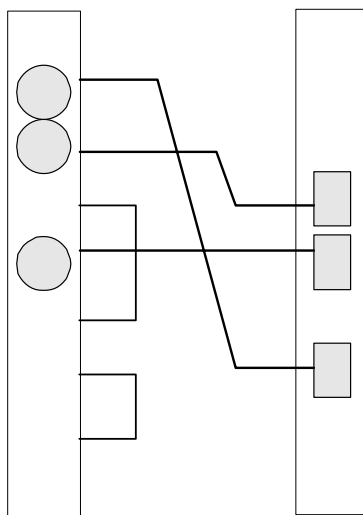
13.4   Frame structure

1)   Basic structure of frame

(1)   Request frame(external communication devices   Cnet module)

(Max. 256 Bytes)

Header (ENQ)	Station No.	Command	Type of command	Structurized data area	Tail (EOT)	Frame check(BCC)

(2)   ACK response frame(Cnet module   external communication devices, when data is normally received)

(Max. 256 Bytes)

Header (ENQ)	Station No.	Command	Type of command	Structurized data area or null	Tail (ETX)	Frame check(BCC)

(3)   NAK response frame(Cnet module   external communication devices, when data is abnormally received)

(Max. 256 Bytes)

Header (NAK)	Station No.	Command	Type of command	Error code (ASCII 4 Bytes)	Tail (ETX)	Frame check(BCC)

Remark

The contents of the code used are as below Table. Control characters are importantly used during serial communication, so they must be well acquainted.

Table 13.1   Control characters

Code	Hex value	Original word	Contents
ENQ(Header)	H05	Enquire	Start code of request frame
ACK(Header)	H06	Acknowledge	Start code of ACK response frame
NAK(Header)	H15	Not acknowledge	Start code of NAK response frame
EOT(Tail)	H04	End of text	End ASCII code of request frame
ETX(Tail)	H03	End Text	End ASCII code of response frame

Remark

Numerical data of all frames is ASCII code of hex value as long as there is not any definition. The contents that is indicated into hex-decimals are as follows :

- Station number

- Command type in case that command type is numerical(means data type) when main commands are R(r) and W(w).

- All items indicating data size of data area structurized.

- Command type(register number) for monitor register and execution command M(n).

- All contents of data

- Frame number of domain

Remark

For hex-decimal data, 'H' such as H01, H12345, H34, H12, or H89AB indicates that the data is a type of hex-decimal.

---

2)   Sequence of command frame

(1)   Sequence of command request frame

ENQ	Station No.	Command	Formatted data	EOT	BCC	(PLC ACK response)

ACK	Station No.	Command	Data or null	ETX	BCC

NAK	Station No.	Command	Error code	ETX	BCC

(PLC NAK response)
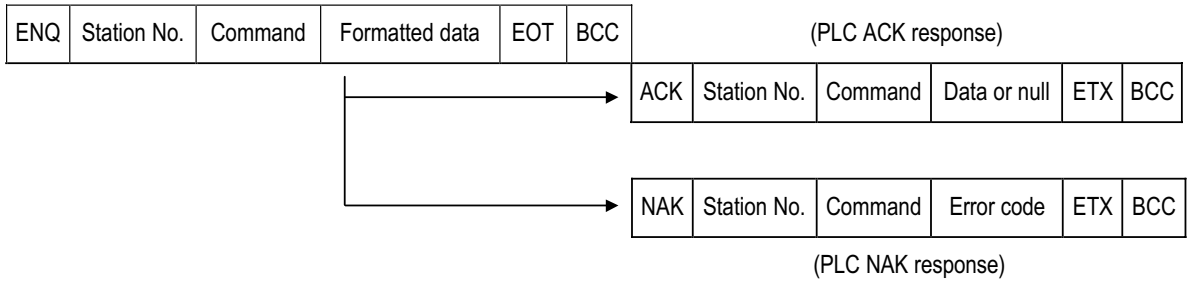
(2)   Sequence of Download/upload frame

ENQ	Station No.	Start Command	Data	EOT	BCC

ACK	Station No.	Command	Data or null	ETX	BCC

ENQ	Station No.	Command	Formatted data	EOT	BCC	(Down/upload command frame No. H0001)

ACK	Station No.	Command	Data or null	ETX	BCC

.
.
.
.

ENQ	Station No.	Command	Formatted data	EOT	BCC	(Down/upload end command frame No. HFFFF)

ACK	Station No.	Command	Data or null	ETX	BCC
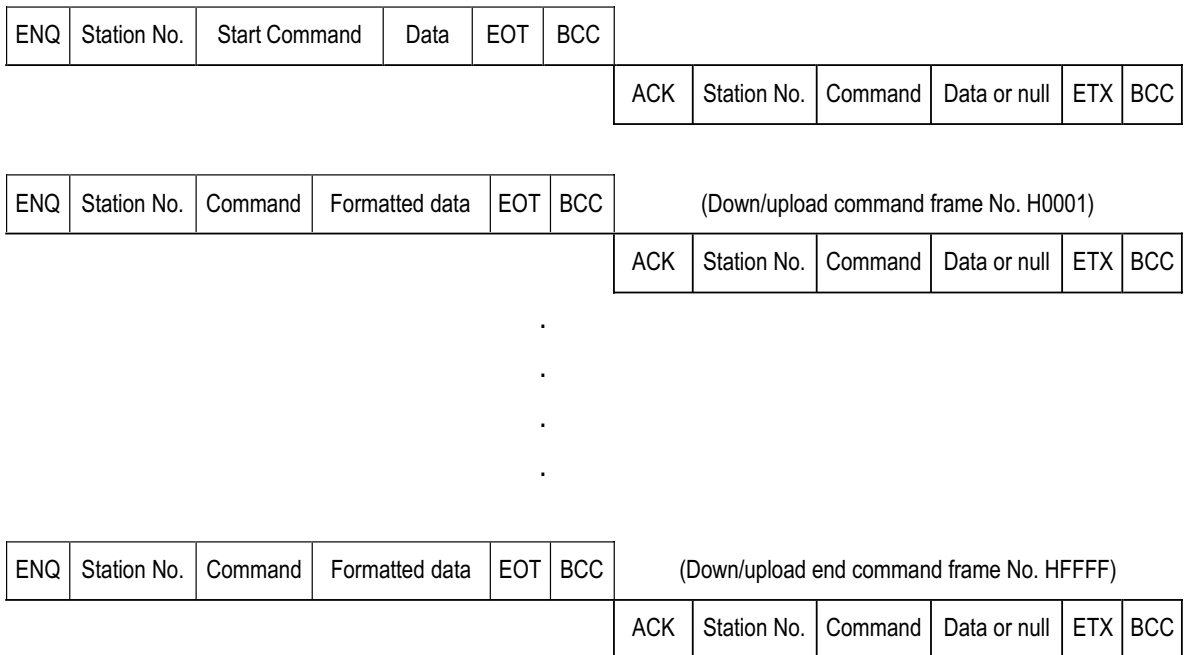
13.5   List of commands

Commands used in dedicated communication service are as below Table :

[Table 13.2   List of commands]

		Command			Contents
		Main command		Command type	
		Sign	ASCII code	Sign	ASCII code	
Direct var. Reading	Indivi.	r (R)	H72 (H52)	SS	5353	Reads direct variables of Bit, Byte, Word, Dword, and Lword type.
	Contin.	r (R)	H72 (H52)	SB	5342	Reads direct variables of Byte, Word, Dword, and Lword type in block unit. (Continuous reading Bit is unavailable)
Named Var. Reading[1]	Indivi.	r (R)	H72 (H52)	H00-H14	3030-3134	Reads data according to data type of named variable. (Variable to be read must be one registered in access variable area.)
	Array	r (R)	H72 (H52)	H15-H27	3135-3237	Reads data of array named variable. (Variable to be read must be one registered in access variable area.)
Direct var. Writing	Indivi.	w(W)	H77 (H57)	SS	5353	Writes data to direct variable of Bit, Byte, Word, Dword, Lword type.
	Contin.	w(W)	H77 (H57)	SB	5342	Writes data to direct variable of Byte, Word, Dword, Lword type in block unit. (Continuous reading Bit is unavailable)
Named Var. Writing[2]	Indivi.	w(W)	H77 (H57)	H00-H14	3030-3134	Writes variable of each data type using variable name. (Variable to be read must be one registered in access variable area.)
	Array	w(W)	H77 (H57)	H15-H27	3135-3237	Writes data to array named variable. (Variable to be read must be one registered in access variable area.)
Monitor Var. Register		x(X)	H78 (H58)	H00-H31	3030-3331	Register variable to be monitored. If registered variable is named one, variable to be read must be one registered in access variable area.
Monitor Execution		y(Y)	H79 (H59)	H00-H31	3030-3331	Carries out the registered variable to monitor.

Remark

In the main command, the capital and small letter have different meaning. In other field, however, it doesn't care letters are capital or small. For example, %mW100 and %mw100 are exactly same command.

---

[1] The CPU-A type does not support this function.

13.6   Data type

When direct variables and named variables are read/written, attention must be paid to data type of direct and named variables.

1)   Data type of direct variables

Memory device type of GLOFA GM PLC : M(Internal memory), Q(Output), I(Input)

Memory device type of GLOFA GK PLC : P, M, L, K, C, D, T, S, F

Data type for direct variables is indicated next to direct variable indicating character '%'.

Table 13.3   List of data types of direct variables

Data type	Ind. charac..	Example of use
BIT	X(58H)	%MX0, %QX0.0.0, %IX0.0.0, %PX0, %LX0, %FX0
BYTE	B(42H)	%MB10, %QB0.0.0, % IB0.0.0
WORD	W(57H)	%MW10, %QW0.0.0, % IW0.0.0, %PW0, %LW0, %FW0, %DW0
DOUBLE WORD	D(44H)	%MD10, %QD0.0.0, % ID0.0.0

Remark

1) The read/write of named variables will be available with the next version of CPU-A type.

---

[2] The CPU-A type does not support this function.

13.7	Execution of commands(Ex.)

1)   Separately reading(RSS) direct variables

(1)   Introduction

This is a function that reads PLC device memory directly specified in accord with memory data type.

Separate device memory can be read up to 4 at a time.

(2)   Request format(PC-->PLC)

Format name	Header	Station No.	Command	Command type	Number of blocks	Variable length	Variable name	....	Tail	Fame check
Frame (Ex.)	ENQ	H20 [1]	R(r)	SS	H01	H06	%MW100		EOT	BCC
ASCII value	H05	H3230	H52 (72)	H5353	H3031	H3036	H254D5731 30 30		H04	

1 block(Setting can be repeated up to 4 block)

- BCC : When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII and added to BCC.

- Number of blocks : This specifies how much of the blocks composed of '[Variable length][Variable name]' are in this request format. This can be set up to 4. Therefore, the value of [Number of blocks] must be H01(ASCII value:3031)-H04(ASCII value:3034).

- Variable length(Name length of direct variable) : This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).

- Direct variables : Address to be actually read is entered. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.

Remark

1) Numerical data of frame(Ex.) is hex value, and 'H' is unnecessary during preparing real frame.

Direct variables available according to PLC type are as follows :

Table 13.4   Type of direct variables

Type	BOOL	Byte	WORD	DOUBLE WORD	LONG WORD
GM1	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM2	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM3	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	--
GM4	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	--
GM5	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	--
GM6	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	--

For how to specify the area of each device in GLOFA GM and GK series, see GLOFA PLC technical data.

Remark

Device data type of each must be same. If data type of the first block is WORD, and the second block is DOUBLE

WORD, error occurs.

---

(3)   Response format(for PLC of ACK response)

Format name	Header	Stati-on No.	Command	Comma-nd type	Number of blocks	Variable length	Data	....	Tail	Fame check
Frame (Ex.)	ACK	H20	R(r)	SS	H01	H02	HA9F3		ETX	BCC
ASCII value	H06	H3230	H52(72)	H5353	H3031	H3032	H41394633		H04	

1 block(Max. 4 blocks)

- Station number, commands, type of command, and number of blocks are the same as computer request format.

- BCC :   When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each
  to ASCII values from ACK to ETX is converted into ASCII, added to BCC, and sent.

- Number of data means Byte number of hex type, and is converted into ASCII. This number is determined
  according to memory type(X,B,W,D,L) included in direct variable name of computer request format.

Table 13.5   Number of data according to variables

	Available direct variable	**Number of data**
BOOL(X)	%MX,%QX,%IX,%(P,M,L,K,F,T,C,D,S)X	1(Only lowest bit of these is available)
Byte(B)	%MB,%QB,%IB,%(P,M,L,K,F,T,C,D,S)W	1
WORD(W)	%MW,%QW,%IW,%(P,M,L,K,F,T,C,D,S)W	2
DOUBLE WORD(D)	%MD,%QD,%ID,%(P,M,L,K,F,T,C,D,S)W	4

- In data area, there are the values of hex data converted to ASCII code.

Ex.1

The fact that number of data is H04(ASCII code value:H3034) means that there is hex data of 4 Bytes in data(DOUBLE

WORD). Hex data of 4 Bytes is converted into ASCII code in data.

Ex.2

If number of data is H04 and the data is H12345678, ASCII code converted value of this is "31 32 33 34 35 36 37 38",

and this contents is entered in data area. Namely, highest value is first, lowest value is last.

Remark

If data type is BOOL, data read is indicated by one Byte of hex. Namely, if Bit value is 0, it indicated by H00, and if 1, by

H01.

---

(4)   Response format(for NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Byte)	Tail	Frame check
Frame(Ex.)	NAK	H20	R(r)	SS	H1132	ETX	BCC
ASCII value	H15	H3230	H52(72)	H5353	H31313332	H03	

- Station number, commands, and type of command are the same as computer request format.

- BCC : When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each

     to ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.

- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see Appendix

    'B. Error Code Table'.

(5)   Example of use

- This example supposes when 1 WORD from %MW20 of station No.1 and 1 WORD from %QW0.2.1 address are read. Also it is supposed that H1234 is entered in %MW20, and data of H5678 is entered in %QW0.2.1.

(Computer request format)

Format name	Header	Statio n No.	Command	Comma- nd type	Number of blocks	Variabl e length	Variable name	Variable length	Variable name	Tail	BCC
Frame (Ex.)	ENQ	H01	R(r)	SS	H02	H05	%MW20	H08	%QW0.2.1	EOT	BCC
ASCII value	H05	H3031	H52(72)	H5353	H3032	H3035	H254D5 73230	H3038	H255157 302E322 E31	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Comman d	Comman d type	Number of blocks	Numbe r of data	Data	Number of data	Data	Tail	BCC
Frame (Ex.)	ACK	H01	R(r)	SS	H02	H02	H1234	H02	H5678	ETX	BCC
ASCII value	H06	H3031	H52(72)	H5353	H3032	H3032	H3132 3334	H3032	H3536 3738	H03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Command type	Error code	Tail	BCC
Frame (Ex.)	NAK	H01	R(r)	SS	Error code(2)	ETX	BCC
ASCII value	H15	H3031	H52(72)	H5353	Error code(4)	H03	

2)   Continuous reading(RSB) of direct variable

(1)   Introduction

This is a function that reads the PLC device memory directly specified in accord with memory data type. With this, data is read from specified address as much as specified.

(2)   Request format

Format name	Header	Station No.	Command	Command type	Variable length	Variable name	Number of data (Max.120 Bytes)	Tail	Frame check
Frame (Ex.)	ENQ	H10	R(r)	SB	H06	%MD100	H05	EOT	BCC
ASCII value	H05	H3130	H52 (72)	H5342	H3036	H254D44 313030	H3035	H04	

Remark

Number of data specifies the number according to the type of direct variable. Namely, if the data type of direct variable is double word, and number of data is 5, it means that read 5 DOUBLE WORDs.

- BCC :   When command is one of lower case(r), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC.

- Name length of direct variable : This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).

- Direct variables :   Address to be actually read is entered in this. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered. Continuous reading of direct variables available according to PLC type are as follows :

Table 13.6   Readable continuous variable area

	BOOL	Byte	WORD	DOUBLE WORD	LONG WORD
GM1	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM2	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM3	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM4	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM5	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM6	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL

(3) For PLC ACK response after execution of command

Format name	Header	Station No.	Command	Command type	Number of data	Data	Tail	Frame check
Frame (Ex.)	ACK	H10	R(r)	SB	H14	H1122334455667788 99AABBC CDDEEFF1122334455	EOT	BCC
ASCII value	H06	H3130	H52(72)	H5342	H3134	H313132323333343435353636 373738383939414142424343434 4444545464631313232333334 343535	H03	

- Station number, main commands, and type of command are the same as computer request format.

- *BCC :   When main command is lower case(like 'r'), only one lower byte of the value resulted by adding ASCII*
    values from ACK to ETX is converted into ASCII, added to BCC, and sent.
    When main command is upper case(like 'r'), BCC is not used.

- Number of data means Byte number of hex type, and is converted into ASCII. This number is determined by multiplying the data number of computer request format by the data size(in below Table) according to memory type(B,W,D,L) included in direct variable name of computer request format.

Table 13.7   Available direct variables

	Available direct variable	Number of data
Byte(B)	%MB,%QB,%IB	1
WORD(W)	%MW,%QW,%IW,%(P,M,L,K,F,T,C,D,S)W	2
DOUBLE WORD(D)	%MD,%QD,%ID	4

Ex.1

When memory type included in direct variable name of computer request format is W(WORD), and data number of computer request format is 03, data number of PLC ACK response after execution of command is indicated by

---

- In data area, the value converted from hex data to ASCII code is entered.

Ex.2

In just above example, when data contents of 3 WORDs are 1234, 5678, and 9ABC in order, actual ASCII code converted values are 31323334 35363738 39414243, and the contents is to be entered in data area.

---

(4)   Response format(for PLC NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Bytes)	Tail	Frame check
Frame (Ex.)	NAK	H10	R(r)	SB	H1132	ETX	BCC
ASCII value	H15	H3130	H52(72)	H5342	H31313332	H03	

- Station number, commands, and type of command are the same as computer request format.
- *BCC : When main command is lower case(like 'r'), only one lower byte of the value resulted by adding ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.*
  When main command is upper case(like 'r'), BCC is not used.
- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see Appendix 'B. Error Code Table'.

(5)   Example of use

This example supposes when 2 DOUBLE WORDs from %MD0 of station No.10 are read. Also it is supposed

that the following data are entered in %MD0 and %MD1 :

%MD0 = H12345678

%MD1 = H9ABCDEF0

(Computer request format)

Format name	Header	Station No.	Command	Command type	Variable length	Variable name	Number of data	Tail	BCC
Frame (Ex.)	ENQ	H0A	R(r)	SB	H04	%MD0	H02	EOT	BCC
ASCII value	H05	H3041	H52(72)	H5342	H3034	H254D4430	H3032	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Command	Command type	Number of blocks	Number of data	Data	Tail	BCC
Frame (Ex.)	ACK	H0A	R(r)	SB	H01	H08	12345678 9ABCDEF0	ETX	BCC
ASCII value	H06	H3041	H52(72)	H5342	H3031	H3038	H3132333435363738394142434445 4630	03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Command type	Error code	Tail	BCC
Frame (Ex.)	NAK	H0A	R(r)	SB	Error code(2)	ETX	BCC
ASCII value	H15	H3041	H52(72)	H5342	Error code(4)	H03	

3)   Separate writing of direct variable (WSS)

(1)   Introduction

This is a function that directly specifies PLC device memory and writes in accord with data type. Device memory can be separately written up to 4 memories at a time.

(2)   Request format

Format name	Headr	Statin No.	Command	Command type	Number of blocks	Variable length	Variable name	Data	....	Tail	Frame check
Frame (Ex.)	ENQ	H20	W(w)	SS	H01	H06	%MW100	H00E2		EOT	BCC
ASCII value	H05	H3230	H57 (77)	H5354	H3031	H3036	H254D57 313030	H3030 4532		H04	

1 blocks(can be repeatedly set up to 4 blocks)

- BCC : When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC, and sent.
- Number of blocks : This specifies how much of the blocks composed of '[Variable length][Variable name]' are in this request format. This can set up to 4 blocks. Therefore, the value of [Number of blocks] must be H01(ASCII value:3031)-H04(ASCII value:3034).
- Variable length(Name length of direct variable) : This indicates the number of the name's characters that registered in direct variable of PLC, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).
- Direct variable : This is an address of variable to be actually read. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered.
- Data : If the value to be written in %MW100 area is H A, the data format must be H000A. If the value to be written in %MD100 area is H A, the data format must be H0000000A. In data area, the ASCII value converted from hex data is entered.

The following shows direct variables available according to PLC type.

Type	BOOL	Byte	WORD	DOUBLE WORD
GM1/2	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID
GM3/4/5	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID
GM4	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID
GM5	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID
GM6	%MX,%QX,%IX	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID

Ex.1

If type of data to be currently written is DOUBLE WORD, the data is H12345678, ASCII code converted value of this is

"3132333435363738", and this content must be entered in data area. Namely, most significant value must be sent first,

least significant value must be last.

Remark

1) Device data types of each blocks must be same.

2) If data type is BOOL, the data to be written is indicated by 1 Byte of hex. Namely, if Bit value is 0, it must be

indicated by H00(3030), and if 1, by H01(3031).

(3)   Response format(for ACK response)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame(Ex.)	ACK	H20	W(w)	SS	ETX	BCC
ASCII value	H06	H3230	H57(77)	H5353	H03	

- Station number, commands, and type of command are the same as computer request format.

- BCC : When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte

    each to ASCII values from ACK to ETX is converted into ASCII, added to BCC, and sent.

(4)   Response format(for NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Bytes)	Tail	Frame check
Frame(Ex.)	NAK	H20	W(w)	SS	H4252	ETX	BCC
ASCII value	H15	H3230	H57(77)	H5353	H34323532	H03	

- Station number, commands, and type of command are the same as computer request format.

- BCC : When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte

    each to ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.

- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see

    Appendix 'B. Error Code Table'.

(5)   Example of use

This supposes that "H00FF" is written in %MW230 address.

(Computer request format)

Format name	Head-er	Station No.	Command	Command type	Number of blocks	Variable name length	Variable name	Data	Tail	BCC
Frame (Ex.)	ENQ	H01	W(w)	SS	H01	H06	%MW230	H00FF	EOT	BCC
ASCII value	H05	H3031	H57(77)	H5353	H3031	H3036	H254D573 23330	H303046 46	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Command	Command type			Tail	BCC
Frame (Ex.)	ACK	H01	W(w)	SS			ETX	BCC
ASCII value	H06	H3031	H57(77)	H5353			H03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Command type	Error code	Tail	BCC
Frame (Ex.)	NAK	H01	W(w)	SS	Error code(2)	ETX	BCC
ASCII value	H15	H3031	H57(77)	H5353	Error code(4)	H03	

4)   Continuous writing of direct variable(WSB)

(1)   Introduction

This is a function that directly specifies PLC device memory and continuously writes data from specified address as much as specified length.

(2)   Request format

Form-at name	Head-er	Stat-ion No.	Comm-and	Comm-and type	Variable length	Variable name	Number of data (Max.120 Bytes)	Data	Tail	Frame check
Frame (Ex.)	ENQ	H10	W(w)	SB	H06	%MD100	H01	H11112222	EOT	BCC
ASCII value	H05	H3130	H57 (77)	H5342	H3036	H254D44 313030	H3031	H3131313 132323232	H04	

Remark

1)   Number of data specifies the number according to the type of direct variable. Namely, if the data type of direct variable is DOUBLE WORD, and number of data is 5, it means that write 5 DOUBLE WORDs.

- BCC :   When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC.
- Protocol of continuous writing function of direct variable has not [Number of blocks].
- Name length of direct variable : This indicates the number of name's characters that means direct variable, which is allowable up to 16 characters. This value is one of ASCII converted from hex type, and the range is from H01(ASCII value:3031) to H10(ASCII value:3130).
- Direct variables : Address to be actually read is entered in this. This must be ASCII value within 16 characters, and in this name, digits, upper/lower case, '%' and '.' only are allowable to be entered. Direct variables available according to PLC type are as follows :

	BOOL	Byte	WORD	DOUBLE WORD	LONG WORD
GM1	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM2	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM3	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM4	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM5	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL
GM6	--	%MB,%QB,%IB	%MW,%QW,%IW	%MD,%QD,%ID	%ML,%QL,%IL

(3)   Request format(for ACK response)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Ex.)	ACK	H10	W(w)	SB	ETX	BCC
ASCII value	H06	H3130	H57(77)	H5342	H03	

- Station number, command and command type are the same as computer request format.

- BCC : When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte
  each to ASCII values from ACK to ETX is converted into ASCII, added to BCC, and sent.

(4)   Response format(for PLC NAK response)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Bytes)	Tail	Frame check
Frame(Ex.)	ENQ	H10	W(w)	SB	H1132	EOT	BCC
ASCII value	H05	H3130	H57(77)	H5342	H31313332	H03	

- Station number, command and command type are the same as computer request format.

- BCC : When command is one of lower case(w), only one lower byte of the value resulted by adding 1 Byte
  each to ASCII values from ACK to ETX is converted into ASCII, added to BCC, and sent.

- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see
  Appendix 'B. Error Code Table'.

(5)   Example of use

This supposes that HAA15056F is written in %QD0.0.0 of No.1 address.

(Computer request format)

Format name	Head-er	Station No.	Comm-and	Command type	Variable length	Variable name	Number of data	Data	Tail	Frame check
Frame (Ex.)	ENQ	H01	W(w)	SB	H08	%QD0.0.0	H01	HAA150 56F	EOT	BCC
ASCII value	H05	H3031	H57(77)	H5342	H3038	H254442302 E302E30	H3031	H414131 3503536 46	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Command	Command type	Tail	Frame check
Frame (Ex.)	ACK	H01	W(w)	SB	ETX	BCC
ASCII value	H06	H3031	H57(77)	H5342	H03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Command type	Error code	Tail	Frame check
Frame (Ex.)	NAK	01	W(w)	SB	Error code(2)	ETX	BCC
ASCII value	H15	H3031	H57(77)	H5342	Error code(4)	H03	

5)   Monitor register(X##)

(1)   Introduction

Monitor register can separately register up to 32 in combination with actual variable reading command, and carries out the registered one through monitor command after registration.

(2)   Request format

Format name	Header	Station No.	Command	Register No.	Register format	Tail	Frame check
Frame (Ex.)	ENQ	H10	X(x)	H1F	See register format	EOT	BCC
ASCII value	H05	H3130	H58(78)	H3146	[ ]	H04	

- BCC : When command is one of lower case(x), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC, and sent.
- Register No. : This can be registered up to 32(0-31, H00-H1F), and if an already registered No. is registered again, the one of current execution is registered.
- Register format : This is used to before EOT in command of formats of separate reading of direct variable, continuous reading, and named variable reading.

Register format : Register format of request formats must select and use only one of the followings.

Separate reading of direct variable

RSS	Number of blocks(2 Bytes)	Variable length(2 Bytes)	Variable name(16 Bytes)	....

1 block(Max. 4 blocks)

Continuous reading of direct variable

RSB	Variable length(2 Bytes)	Variable name(16 Bytes)	Number of data

1 block(Max. 4 blocks)

(3)   Response format(for PLC ACK response)

Format name	Header	Station No.	Command	Register No.	Tail	Frame check
Frame(Ex.)	ACK	H10	X(x)	H1F	ETX	BCC
ASCII value	H06	H3130	H58(78)	H3146	H03	

- Station number, command and resister No. are the same as computer request format.
- BCC : When command is one of lower case(x), only one lower byte of the value resulted by adding 1 Byte each
  to ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.

(4)   Response format(for PLC NAK response)

Format name	Header	Station No.	Command	Register No.	Error code(Hex 2 Bytes)	Tail	Frame check
Frame (Ex.)	ACK	H10	X(x)	H1F	H1132	ETX	BCC
ASCII value	H06	H3130	H58(78)	H3146	H31313332	H03	

- Station number, main commands, and resister No. are the same as computer request format.
- BCC : When command is one of lower case(x), only one lower byte of the value resulted by adding 1 Byte each
  to ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.
- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see Appendix
  'A2. Error Code Table'.

(5)   Example of use

This supposes that the variable which data type of station No.'1' is UINT and the variable name is "ASDF" is
monitor-registered with No.'1'.

(Computer request format)

Format name	Head-er	Stat-ion No.	Comm-and	Regist-er No.	Register format				Tail	Frame check
					R##	Number of blocks	Variable length	Variable name		
Frame(Ex.)	ENQ	H01	X(x)	H01	R0A	H01	H04	ASDF	EOT	BCC
ASCII value	H05	H3031	H58(78)	H3031	H523041	H3031	H3034	H41534446	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Command	Register No.	Tail	Frame check
Frame(Ex.)	ACK	H01	X(x)	H01	ETX	BCC
ASCII value	H06	H3031	H58(78)	H3031	H03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Command type	Error code (Hex 2 Bytes)	Tail	Frame check
Frame(Ex.)	NAK	H01	X(x)	H01	Error code(2)	ETX	BCC
ASCII value	H15	H3031	H58(78)	H3031	Error code(4)	H03	

6)   Monitor execution(Y##)

(1)   Introduction

This is a function that carries out the writing of the variable registered by monitor register. This also specifies

registered No. and carries out the writing of the variable registered in the No.

(2)   Request format

Format name	Header	Station No.	Command	Register No.	Tail	Frame check
Frame(Ex.)	ENQ	H10	Y(y)	H1F	EOT	BCC
ASCII value	H05	H3130	H59(79)	H3146	H03	

- Register No. uses the same No. as the No. registered during monitor register for monitor execution.
- BCC :   When main command is one of lower case(y), only one lower byte of the value resulted by adding 1

Byte each to ASCII values from ENQ to EOT is converted into ASCII, added to BCC, and sent.

- In computer request format, register No. can be set to 00-31(H00-H1F).

(3)   Response format(for PLC ACK response)

① In case that the register format of register No. is the separate reading of direct variable

Format name	Header	Station No.	Command	Register No.	Number of blocks	Number of data	Data	Tail	Frame check
Frame (Ex.)	ACK	H10	Y(y)	H1F	H01	H04	H9183AABB	ETX	BCC
ASCII value	H06	H3130	H59(79)	H313F	H3031	H3034	H3931383341 414242	H03	

② In case that the register format of register No. is the continuous reading of direct variable

Format name	Header	Station No.	Command	Register No.	Number of data	Data	Tail	Frame check
Frame (Ex.)	ACK	H10	Y(y)	H1F	H04	H9183AABB	ETX	BCC
ASCII value	H06	H3130	H59(79)	H313F	H3034	H3931383341414242	H03	

③   In case that the register format of register No. is the reading of named variable

Format name	Header	Station No.	Command	Register No.	Number of blocks	Number of data	Data	Tail	Frame check
Frame (Ex.)	ACK	H10	Y(y)	H1F	H01	H04	H9183AABB	ETX	BCC
ASCII value	H06	H3130	H59(79)	H313F	H3031	H3034	H3931383341 414242	H03	

- Data format such as number of blocks and number of data is the same as the contents of variable writing.

- Station number, commands, and register No. are the same as computer request format.

- BCC : When main command is one of lower case(y), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from ACK to ETX is converted into ASCII, added to BCC, and sent.

(4)   Response format(for PLC NAK response)

Format name	Header	Station No.	Command	Register No.	Error code (Hex 2 Bytes)	Tail	Frame check
Frame (Ex.)	NAK	H10	Y(y)	H1F	H1132	ETX	BCC
ASCII value	H15	H3130	H59(79)	H3146	H31313332	H03	

- Station number, commands, and register No. are the same as computer request format.

- BCC : When command is one of lower case(y), only one lower byte of the value resulted by adding 1 Byte each to ASCII values from NAK to ETX is converted into ASCII, added to BCC, and sent.

- Error code is hex and 2 Bytes(ASCII code, 4 Bytes), which indicates type of error. For the details, see Appendix 'A2. Error Code Table'.

(5)　Example of use

　　This supposes that reading the variable registered with register No.'1' in station No.'1' is carried out. It is also

　　supposed that the one registered is a named variable reading, the number of blocks is 1, and the data type is

　　DINT.

(Computer request format)

Format name	Header	Station No.	Command	Register No.	Tail	Frame check
Frame(Ex.)	ENQ	H01	Y(y)	H01	EOT	BCC
ASCII value	H05	H3031	H59(79)	H3031	H04	

(For PLC ACK response after execution of command)

Format name	Header	Station No.	Command	Register No.	Number of blocks	Number of data	Data	Tail	Frame check
Frame (Ex.)	ACK	H01	Y(y)	H01	H01	H04	H23422339	ETX	BCC
ASCII value	H06	H3031	H59(79)	H3031	H3031	H3034	H3233343232 333339	H03	

(For PLC NAK response after execution of command)

Format name	Header	Station No.	Command	Register No.	Error code	Tail	Frame check
Frame(Ex.)	NAK	H01	Y(y)	H01	Error code(2)	ETX	BCC
ASCII value	H15	H3031	H59(79)	H3031	Error code(4)	H03	

13.8   Error code during NAK occurrence(for GM6 dedicated communication)

Error code	Error type	Contents	Action to take
H0001	PLC system error	Interface with PLC impossible	Power On/Off
H0011	Data error	* Error occurred when ASCII data value is converted into digits	Check whether another character than upper and lower cases('%',' ',' '), and digits has been used, correct, and execute again.
H0021	Instruction error	* Using wrong instruction	Inspect instruction
H0031	Instruction type error	* Instruction is used in wrong type	Inspect instruction type
H1132	Device memory error	* Wrong specified device memory	Inspect device type
H1232	Data size error	* Execution data number exceeding 120 Bytes	Correct data length
H2432	Data type error	* Data type mismatch with actual variable	Equalize variable and data type of PLC program
H7132	Variable request format error	*'%' is missing	Inspect format, correct, and then execute again.
H2232	Area exceeding error	* M,I,Q area exceeding error	Inspect area difinition and execute again
H0190	Monitor execution error	Registered number exceeding range	Execute again after adjusting monitor registration number to 31 or less.
H0290	Monitor registration error	Registered number exceeding range	Execute again after adjusting monitor registration number to 31 or less.
H6001	Syntax error_6001	Not available instruction is used	
H6010	Syntax error_6010	Over-run, Frame error	Check the system is in stop mode
H6020	Syntax error_6020	Time out error	Check the connection of RS-232C port
H6030	Syntax error_6030	Instructino syntax error	Check each frame has ENQ, EOT
H6040	Syntax error_6040	Text of one frame exceed 256byte	Devide the text into several frames as a text does not exceed 256 byte
H6050	Syntax error_6050	BCC error	Check the BCC is correct

Chapter 14   The RS422/485 communication of GM6-CPUB

14 The RS422/485 communication of GM6-CPUB

14.1 Introductions

1) The GM6-CPUB module can be used as the master station of RS422/485 network and applicable for the 1:N network of G series PLCs and/or PC.

2) To operate the GM6-CPUB as the master station, basic parameters and high speed link parameters should be set properly.

3) The dedicated G Series Cnet protocol is used for transmission control.

4) The GM6-CPUA and GM6-CPUC does not support the master station function.

14.2 Features

1) Max. 64 high speed link items can be assigned.

2) Max. 32 stations can be linked.

3) According to the parameter setting, the operation mode and error code of slave stations is stored at the relevant flag.

4) The communication status can be monitored with the monitoring function of GMWIN software.

14.3 Parameter setup

To start RS422/485 communication,

- The CPU module type should be a B-type CPU. (GM6-CPUB)

- Set the communication parameters of the 'Basic Parameters' of GMWIN software.

- Set the 'High speed link 1' of the 'High Speed Link Parameters'

- Enable the high speed link 1 with 'Link Enable…' menu.

1) Basic parameter setup

a) Station number : Assign the station number of master station in the range of 0 ~ 31

b) Baud rate : Select the communication as 9600, 19200 or 38400 bps.

c) Master/Slave : Only GM6-CPUB can be set as master station. If the CPU is selected as master station, the network type of high speed link 1 is automatically set as G Series 422/485.

d) Timeout : Set the period that the interval until a timeout error occurs. The default value is 500msec and minimum value is 10msec (1×10msec).

e) Read status of slave PLC : If check this item, the master station reads the status of slave PLCs and store the status at the corresponding flags.

2) High speed link parameter setup
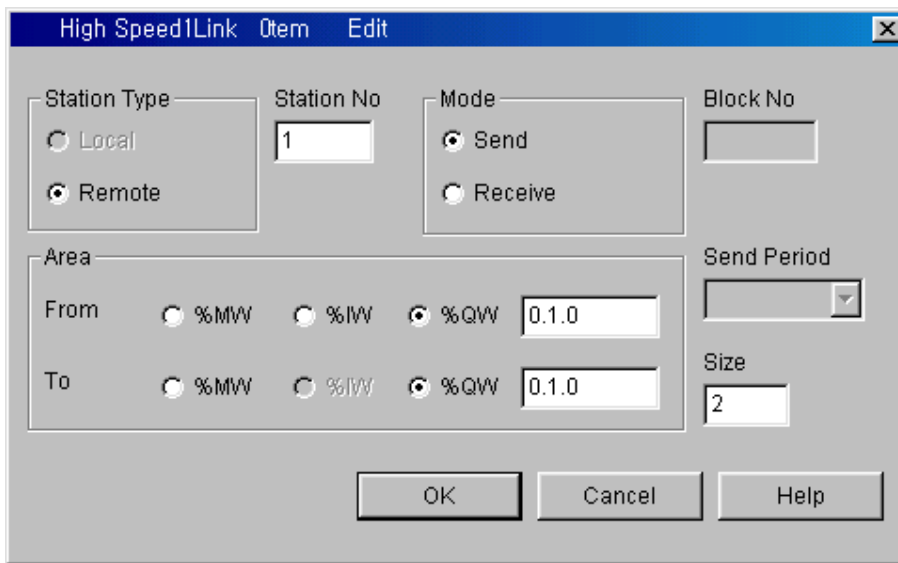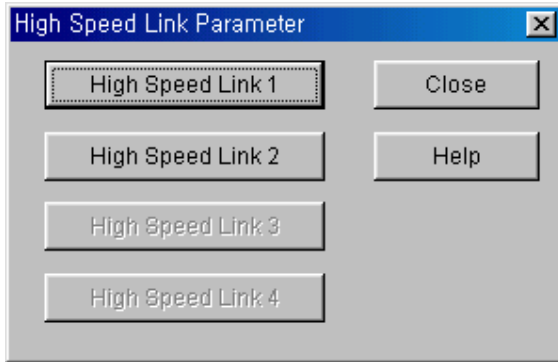
a) Only the 'High speed link 1' can be set as G Series 422/485 network type.

b) The setup is similar as the high speed link parameter setup with other communication modules such as Fnet module.

- Max. 64 items can be assigned.

- The size of data block is assigned by the unit of word, and the Max. size is 60 words.

- Area setup

    Send   From : I / Q / M   To : Q / M

    Receive   From : I / Q / M   To : Q / M

14.4 The status flag

1) Communication error counter flag

- Flag name : _M422_ERR_CNT[n] (Array_Byte Type, n = 0 ~ 31)

- Description

Each byte of the '_M422_ERR_CNT[n]' array indicates how many times communication errors occurred at the relevant station. For example, the _M422_ERR_CNT[5] is the error counter of station 5.

2) The error code

- Flag name : _M422_ERR[n] (Array_Byte Type, n = 0 ~ 31)

- Description

0 : No error   1 : Timeout error   2 : NAK

3) Operation mode and error of slave station

- Flag name : _S422_STATE[n] (Array_Byte Type, n = 0 ~ 31)

- Description

Bit 0 : Indicates an error of slave PLC. (0 : No error, 1 : Error occurred)

Bit 1 ~ Bit 3 : Reserved

Bit 4 ~ Bit 7 : Indicates the operation mode of slave PLC

Bit 4 : STOP   Bit 5 : RUN

Bit 6 : PAUSE   Bit 7 : DEBUG

4) The status flag of master station

- Flag name : _M422_STATE (Byte Type, n = 0 ~ 31)

- Description

Bit 0 : Turn on when the CPU module is assigned as master station but it is not B type (GM6-CPUB)

Bit 1 : Turn on when the master station number of basic parameter setting is duplicated with one of the slave station numbers of high speed link parameters setting.

Bit 2 : Turn on when the M area of high speed link parameter setting is out of the range.

5) The scan time of RS422/485 communication

- Description   _M422_SCAN_MAX (Time Type) : The maximum scan time

_M422_SCAN_MIN (Time Type) : The minimum scan time

_M422_SCAN_CUR (Time Type) : The current scan time

Remark

Scan time : A total time of the processing time of the all parameter settings. (From the execution of the first parameter setting to the next execution)

14.5 Monitoring

Users can monitor the communication status of RS422/485 network with the monitor function of the GMWIN software. The high speed link parameter 1 monitoring screen is used for monitoring the RS422/485 network status.

- The CPU module should be a B-type, and assigned as master station in the basic parameter setting. Otherwise, the monitor screen will show the status of high speed link service.

- In the monitoring screen, the following flags are shown;

Master PLC parameter	_M422_STATE (On / Off)
The scan time of communication	_M422_SCAN_MAX (Maximum scan time)
	_M422_SCAN_MIN (Minimum scan time)
	_M422_SCAN_CUR (Current scan time)
No., Type, From, To, Size	The contents of high speed link 1 parameters
Error counter and code	_M422_ERR_CNT, _M422_ERR
Slave PLC	_S422_STATE

14.6 Communication method and termination resistor

1) Data type

      Data bit   : 8 bits

      Stop bit   : 1 bit

      Parity   : None

2) Communication speed (Baud rate) : Selectable one of 9600, 19200, 38400 bps

3) Termination resistor

    When use a long cable for connecting two or more PLCs, a termination resistor should be connected at the both ends of network. Otherwise, the communication can be disturbed by the reflected wave of cable. The termination resistor should be 1/2W grade and have the equivalent resistance with the characteristic impedance of cable. (When use the RS-422 protocol, connect two termination resistors between SDA and SDB, RDA and RDB. With the RS-485 protocol, connect a termination resistor between RDA and RDB, or SDA and SDB.)

14.7 RS-422/485 pin assignment

1)   The RS-422 network is connected with 5-pin connector. The following table shows the name, and description of each pins and direction of signal.

Pin No.	MASTER	Signal direction	SLAVE
1	RDA	←	SDA
2	RDB	←	SDB
3	SDA	→	RDA
4	SDB	→	RDB
5	SG	←→	SG

2)   When using RS-485 interface, connect cable as RS-422 interface, then interconnect RDA and SDA, RDB and SDB. With the RS-485 interface, the send / receive signals share one line and communication is performed as half-duplex method.

Chapter 15   The PID function

15 The PID functions

15.1 Introductions

This chapter will provide information about the built-in PID (Proportional Integral Differential) function of B and C type CPU module. (GM6-CPUB and GM6-CPUC) The GM6 series does not have separated PID module like GM4 series, and the PID function is integrated into the CPU module (B and C type)

The PID control means a control action in order to keep the object at a set value (SV). It compares the SV with a sensor measured value (PV : Present Value) and when a difference between them (E : the deviation) is detected, the controller output the manipulate value (MV) to the actuator to eliminate the difference. The PID control consists of three control actions that are proportional (P), integral (I), and differential (D).

The characteristics of the PID function of GM6 is as following;

- the PID function is integrated into the CPU module. Therefore, all PID control action can be performed with F/B (Function Block) without any separated PID module.
- Forward / reverse operations are available
- P operation, PI operation, PID operation and On/Off operation can be selected easily.
- The manual output (the user-defined forced output) is available.
- By proper parameter setting, it can keep stable operation regardless of external disturbance.
- The operation scan time (the interval that PID controller gets a sampling data from actuator) is changeable for optimizing to the system characteristics.

15.2 PID control

15.2.1   Control actions

15.2.1.1   Proportional operation (P operation)

1) P action means a control action that obtain a manipulate value which is proportional to the deviation (E : the difference between SV and PV)

2) The deviation (E) is obtained by multiplying a reference value to the actual difference between SV and PV. It prevents the deviation from a sudden change or alteration caused by external disturbance. The formula of deviation is as following;

$$MV = Kp \times [b \times SV - PV]$$

Kp	: the proportional constant (gain)
b	: reference value
SV	: set value
PV	: present value

3) If the Kp is too large, the PV reaches to the SV swiftly, but it may causes a bad effect like oscillations shown in the Fig. 2.1.

4) If the Kp is too small, oscillation will not occur. However, the PV reaches to the SV slowly and an offset may appear between PV and SV shown in the Fig. 2.2.

5) The manipulation value (MV) varies from 0 to 4,000. User can define the maximum value of MV (MV_MAX) and minimum value (MV_MIN) within the range 0 ~ 4,000.

6) When an offset remains after the system is stabilized, the PV can be reached to the SV by adding a certain value. This value is called as bias value, and user can define the bias value with GM-WIN software.
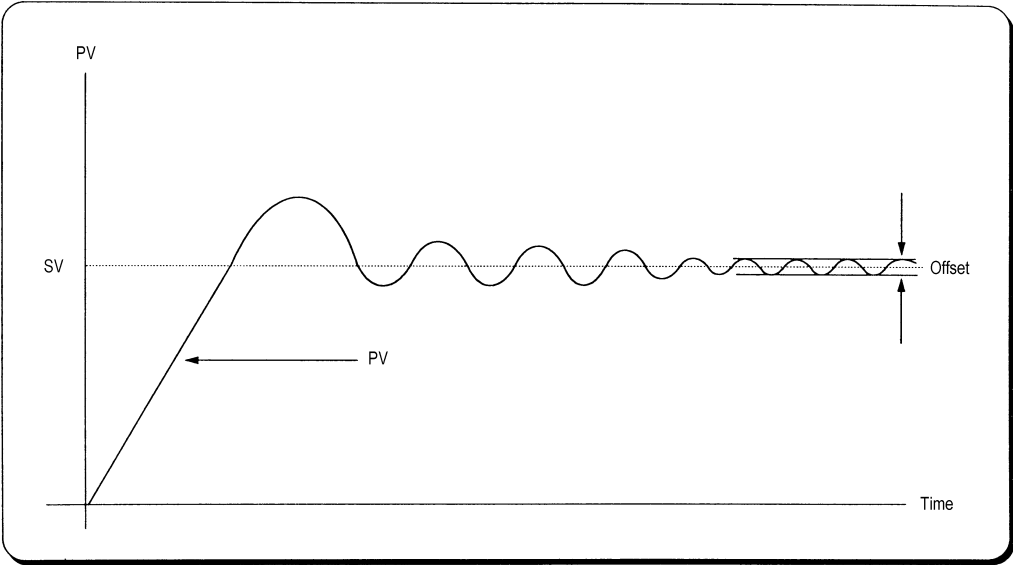
Fig. 2.1 When the proportional constant (Kp) is large
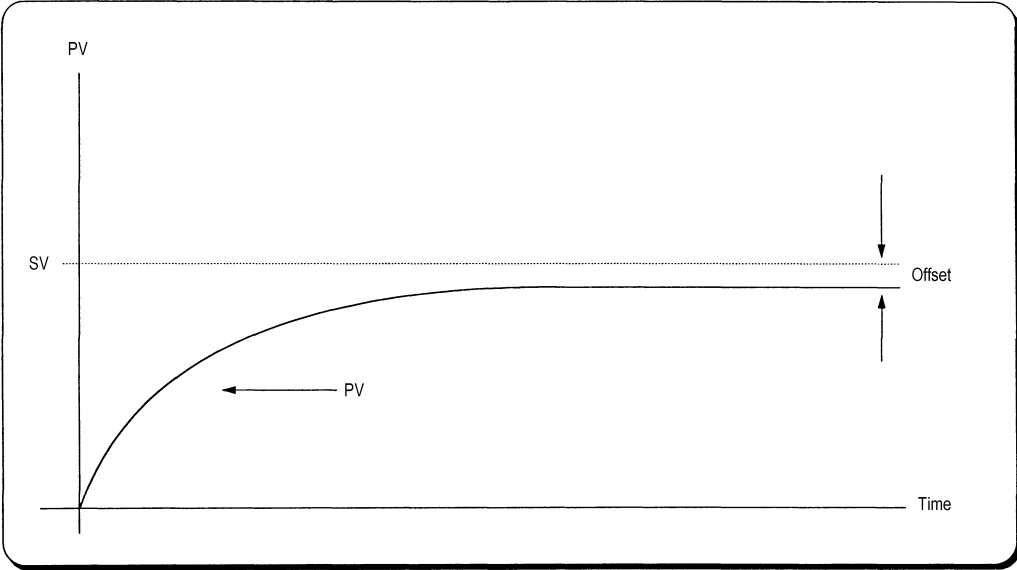
Fig. 2.1 When the proportional constant (Kp) is small

15.2.1.2   Integral operation (I action)

1) With integral operation, the manipulate value (MV) is increased or decreased continuously in accordance time in order to eliminate the deviation between the SV and PV. When the deviation is very small, the proportional operation can not produce a proper manipulate value and an offset remains between PV and SV. The integral operation can eliminate the offset value even the deviation is very small.

2) The period of the time from when the deviation has occurred in I action to when the MV of I action become that of P action is called Integration time and represented as Ki.

3) Integral action when a constant deviation has occurred is shown as the following Fig. 2.4.
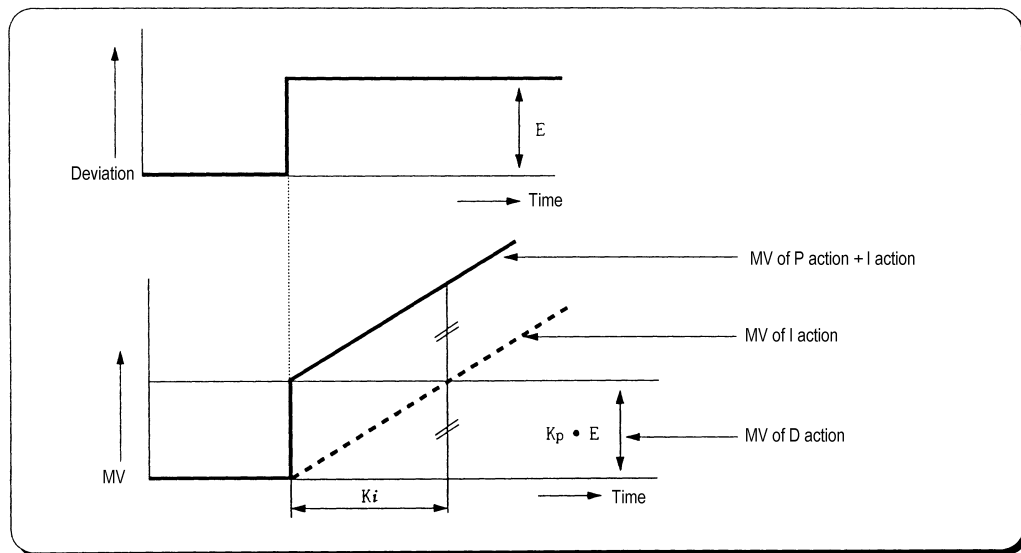
Fig. 2.4   The integral action with constant deviation

4) The expression of I action is as following;

$$MV = \frac{Kp}{Ti} \int E dt$$

As shown in the expression, Integral action can be made stronger or weaker by adjusting integration time (K*i*) in I action.

That is, the more the integration time (the longer the integration time) as shown in Fig. 2.5, the lesser the quantity added to or subtracted from the MV and the longer the time needed for the PV to reach the SV.

As shown in Fig. 2.6, when the integration time given is short the PV will approach the SV in short time since the quantity added or subtracted become increased. But, If the integration time is too short then oscillations occur, therefore, the proper P and I value is requested.

5) Integral action is used in either PI action in which P action combines with I action or PID action in which P and D actions combine with I action.
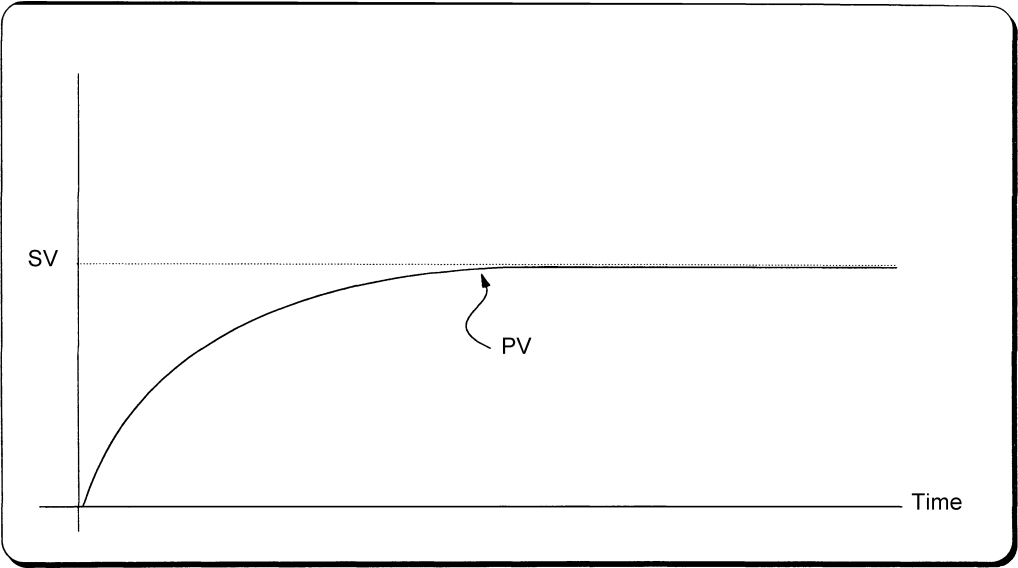
Fig. 2.5   The system response when a long integration time given
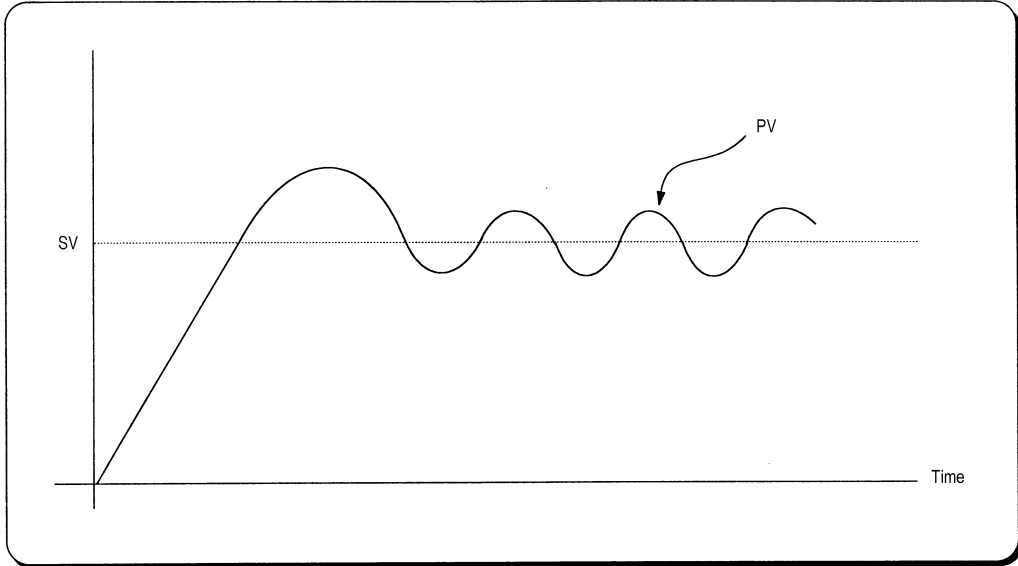
Fig. 2.6   The system response when a short integration time given

15.2.1.3   Derivative operation (D action)

(1) When a deviation occurs due to alteration of SV or external disturbances, D action restrains the changes of the deviation by producing MV which is proportioned with the change velocity (a velocity whose deviation changes at every constant interval) in order to eliminate the deviation.

▸ D action gives quick response to control action and has an effect to reduce swiftly the deviation by applying a large control action (in the direction that the deviation will be eliminated) at the earlier time that the deviation occurs.

▸ D action can prevent the large changes of control object due to external conditions.

(2) The period of time from when the deviation has occurred to when the MV of D action become the MV of P action is called derivative time and represented as Kd.

(3) The D action when a constant deviation occurred is shown as Fig. 2.7.
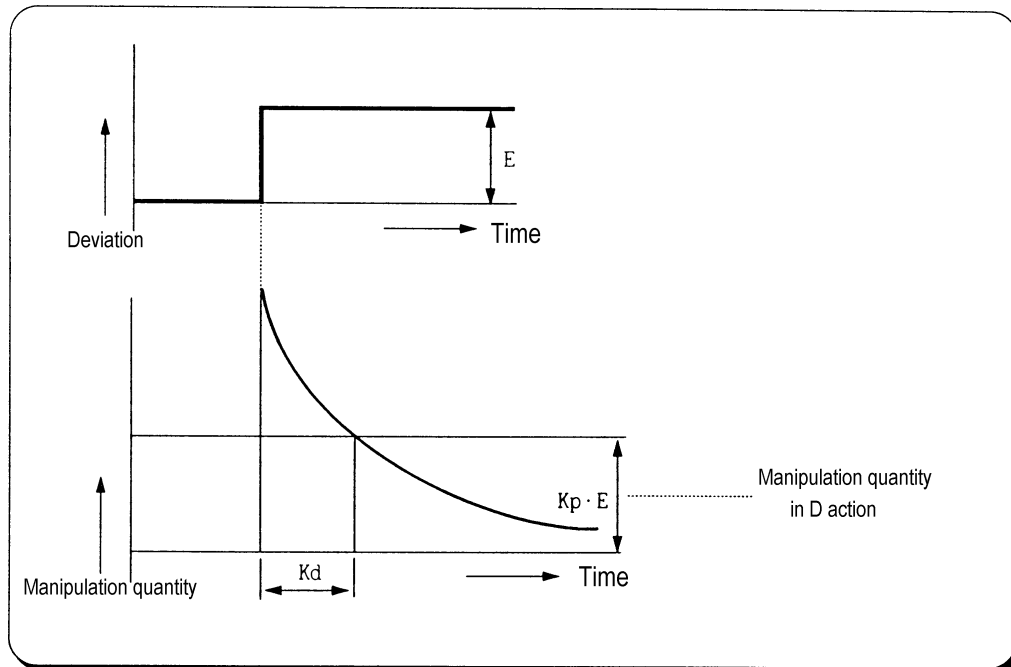
Fig. 2-7   Derivative action with a constant deviation

(4) The expression of D action is as following;

$$MV = Kp \times Td \frac{dE}{dt}$$

(5) Derivative action is used only in PID action in which P and I actions combine with

D action.

15.2.1.4   PID action

1) PID action controls the control object with the manipulation quantity produced by (P+I+D) action

2) PID action when a given deviation has occurred is shown as the following Fig. 2.8.
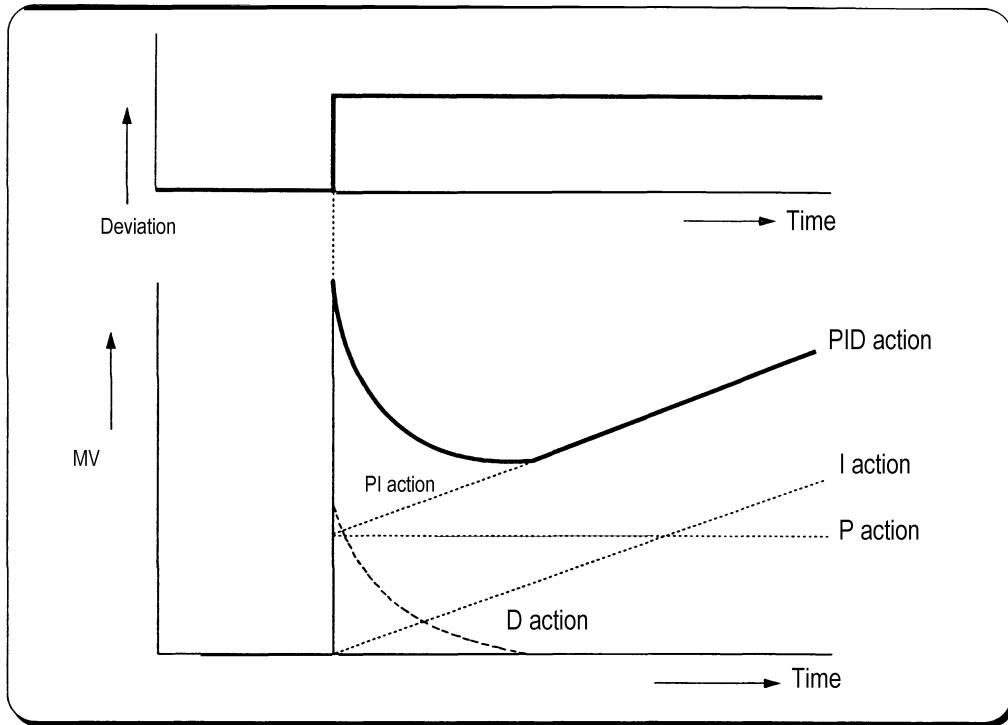
Fig. 2-8   PID action with a constant deviation

15.2.1.5   Forward / Reverse action

1) PID control has two kind of action, forward action and reverse action. The forward action makes the PV reaches to SV by outputting a positive MV when the PV is less than SV.

2) A diagram in which forward and reverse actions are drawn using MV, PV and SV is shown as Fig. 2.9.
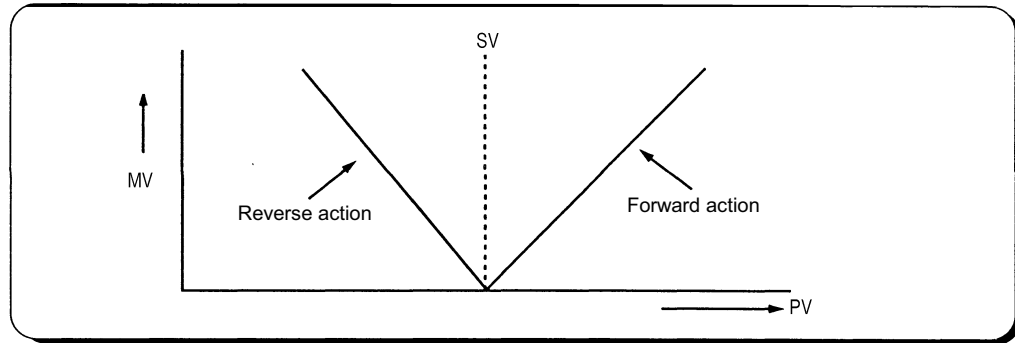
Fig. 2-9 MV of forward / reverse action

3) Fig 2.10 shows examples of process control by forward and reverse actions, respectively.
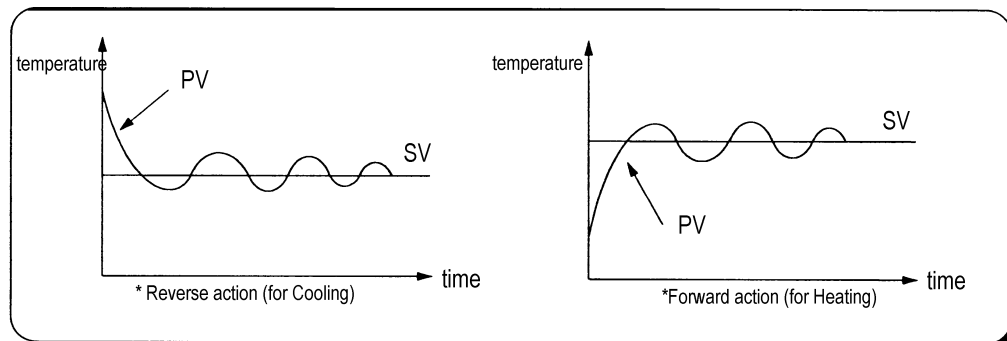
Fig. 2-10 PV of forward / reverse action

15.2.1.6   Reference value

In general feedback control system shown as the Figure 2-10, the deviation value is obtained by the difference of PV and SV. P, I, and D operations are performed based on this deviation value. However, each of P, I, and D operations use different deviation values according to the characteristics of each control actions. The expression of PID control is as following;

$$MV = K\left[Ep + \frac{1}{Ti}\int_0^t Ei(s)ds + Td\frac{dEd}{dt} \right]$$

MV	: Manipulate value
K	: Proportional gain
Ti	: Integral time
Td	: Derivative time
Ep	: Deviation value for proportional action
Ei	: Deviation value for integral action
Ed	: Deviation value for derivative action

The deviation values of P, I, and D action is described as following equations;

$$Ep = b \times SV - PV$$
$$Ei = SV - PV$$
$$Ed = -PV$$

The b of the first equation is called as reference value. It can be varied according to the load disturbance of measurement noise.
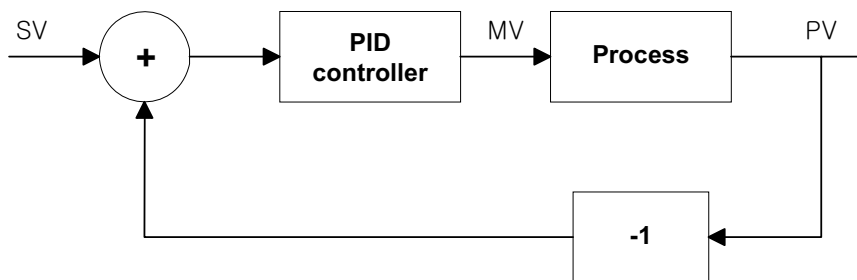
Fig. 2-10   Diagram of simple feedback system

The figure 2.11 shows the variation of PV according to the several different reference values (b). As shown in the Fig. 2.11, the small reference value produces small deviation value, and it makes the control system response be slow.

In general, control system is required to be adaptable to various external / internal changes. Especially, it should shows a stable transient response with the sudden change of the SV to be robust to load disturbances and/or measurement noise.
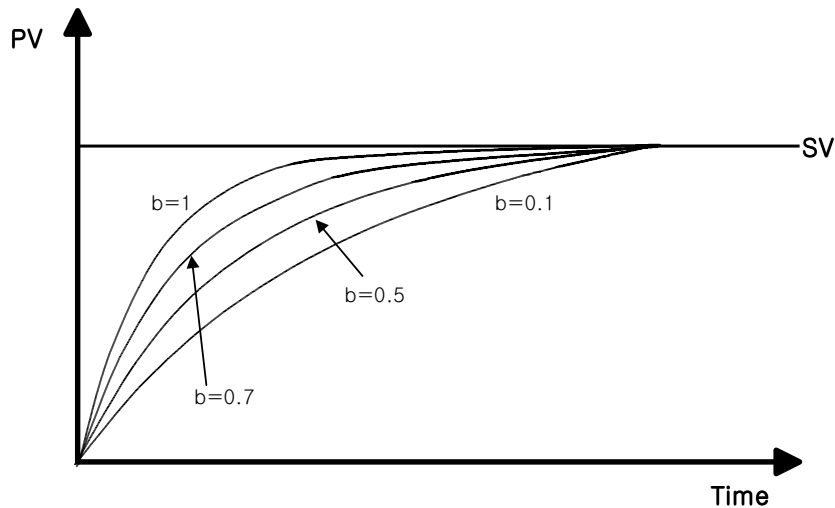
Figure 2-11 The PI control with several reference values

15.2.1.7   Integral windup

All devices to be controlled, actuator, has limitation of operation. The motor has speed limit, the valve can not flow over the maximum value. When the control system has wide PV range, the PV can be over the maximum output value of actuator. At this time, the actuator keeps the maximum output regardless the change of PV while the PV is over the maximum output value of actuator. It can shorten the lifetime of actuator.

When the I control action is used, the deviation term is integrated continuously. It makes the output of I control action very large, especially when the response characteristic of system is slow.

This situation that the output of actuator is saturated, is called as 'windup'. It takes a long time that the actuator returns to normal operating state after the windup was occurred.

The Fig. 2-12 shows the PV and MV of PI control system when the windup occurs. As shown as the Fig. 2-12, the actuator is saturated because of the large initial deviation. The integral term increase until the PV reaches to the SV (deviation = 0), and then start to decrease while the PV is larger than SV (deviation < 0). However, the MV keeps the saturated status until the integral term is small enough to cancel the windup of actuator. As the result of the windup, the actuator will output positive value for a while after the PV reached to the SV, and the system show a large overshoot. A large initial deviation, load disturbance, or mis-operation of devices can cause windup of actuator.

MV (without windup) ————————————
MV (with windup) **————————————**
Integral term ·····························
Proportional term —·—·—·—·—·—·—·—·—
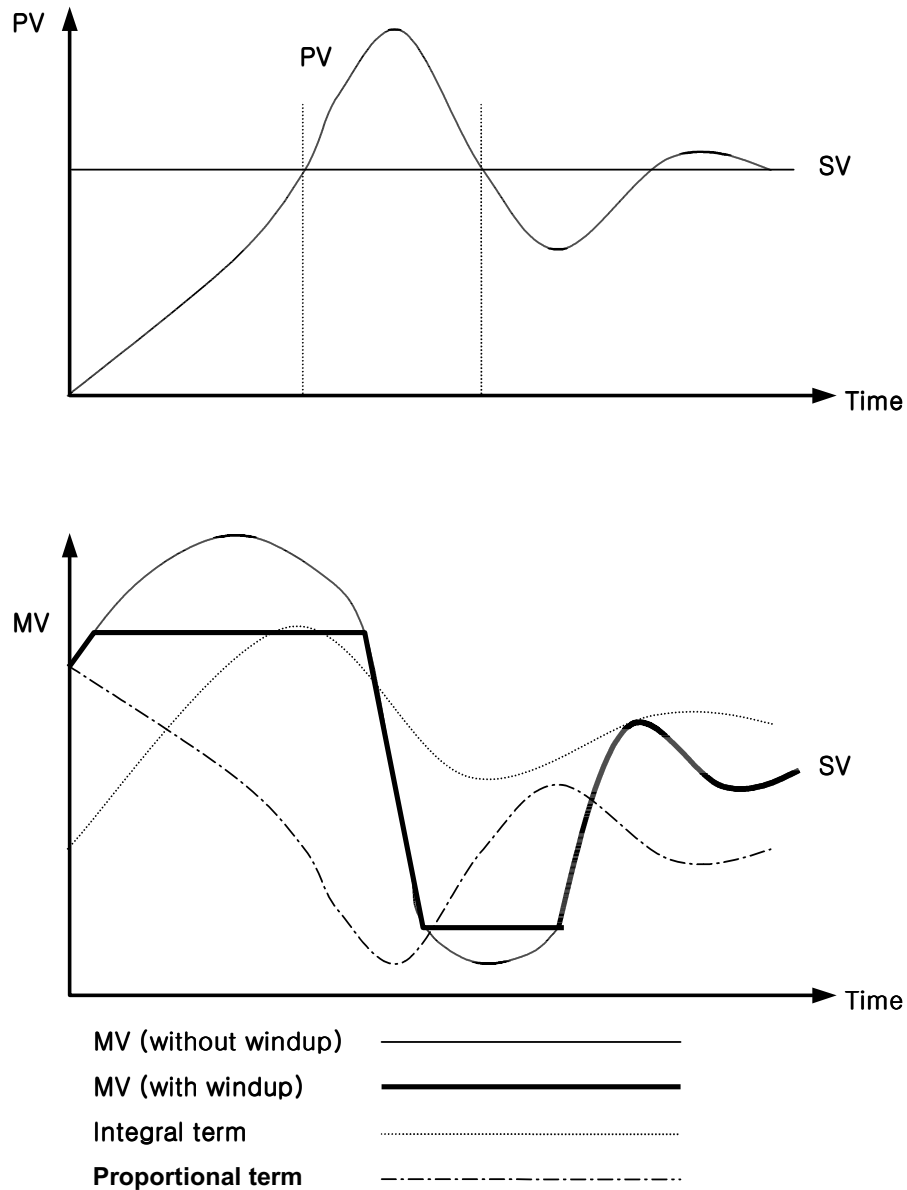
There are several methods to avoid the windup of actuator. The most popular two methods are adding another feedback system to actuator, and using the model of actuator. The Fig. 2-13 shows the block diagram of the anti-windup control system using the actuator model.

As shown in the Fig. 2-13, the anti-windup system feedback the multiplication of gain (1/Tt) and Es to the input of integral term. The Es is obtained as the difference value between actuator output (U) and manipulation value of PID controller (MV). The Tt of the feedback gain is tracking time constant, and it is in inverse proportion with the resetting speed of integral term. Smaller Tt will cancel the windup of actuator faster, but too small Tt can cause anti-windup operation in derivative operation. The Fig. 2-14 shows several Tt value and PV in the PI control system.
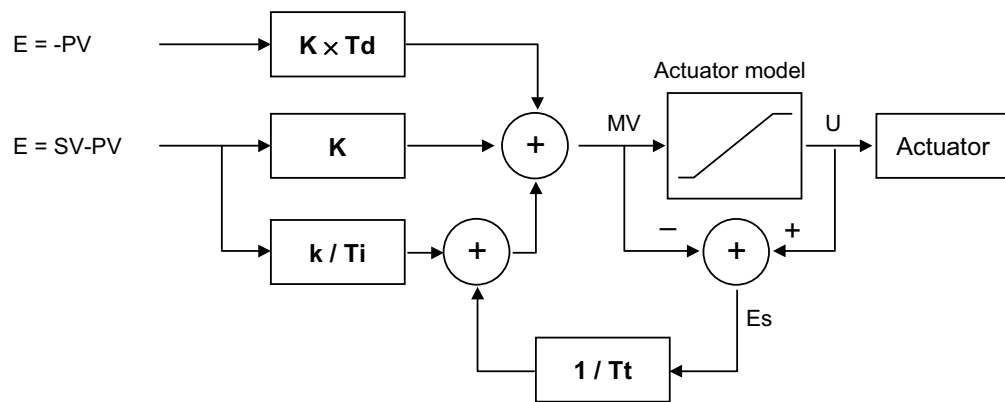
Fig. 2-13   The block diagram of anti-windup control system
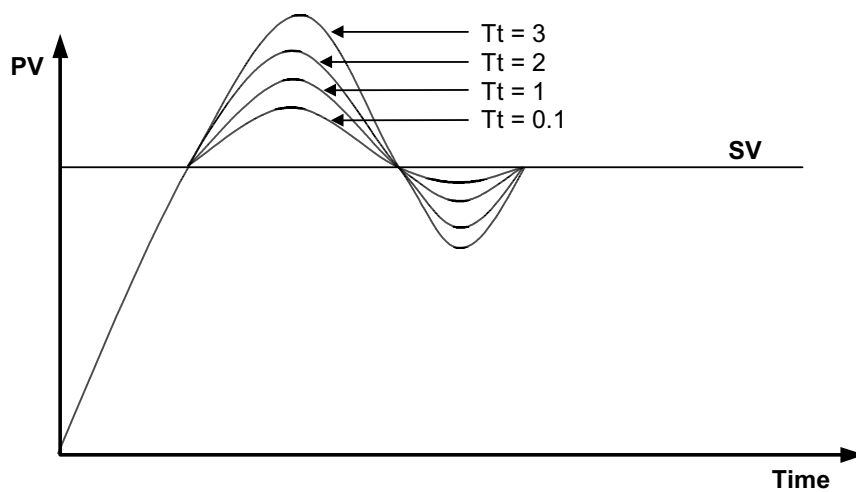
Fig. 2-14   The PV output characteristics with different Tt values.

15.2.2 Realization of PID control on the PLC

In this chapter, it will described that how to get the digitized formula of the P, I, and D terms. Then, the pseudo code of PID control will be shown.

15.2.2.1   P control

The digitized formula of P control is as following;

$$P(n) = K\big[b \times SV(n) - PV(n)\big]$$

n : sampling number

K : proportional gain constant

b : reference value

SV : set value

PV : present value

15.2.2.2   I control

The continuous formula of I control is as following;

$$I(t) = \frac{K}{Ti} \int_0^t e(s)\,ds$$

I(t) : integral term

K : proportional gain constant

Ti : integral time

e(s) : deviation value

By deviation about t, we can obtain;

$$\frac{dI}{dt} = \frac{K}{Ti} e$$

e = (SV – PV) : deviation value

The digitized formula is as following;

$$\frac{I(n+1) - I(n)}{h} = \frac{K}{Ti} e(n)$$

h : sampling period

$$I(n+1) = I(n) + \frac{Kh}{Ti} e(n)$$

15.2.2.3   D control

The continuous formula of derivative term is as following;

$$\frac{Td}{N} \times \frac{d}{dt} D + D = -KTd \frac{dy}{dt}$$

N : high frequency noise depression ration

y : the object to be controlled (PV)

The digitized formula is as following (Use Tustin approximation method)

$$D(n) = \frac{2Td - hN}{2Td + hN} D(n-1) - \frac{2KTdN}{2Td + hN} \left[y(n) - y(n-1) \right]$$

15.2.2.4   Pseudo code of PID control

The pseudo code of PID control is as following;

Step 1 : Get constants that are used for PID operation

$$Bi = K \times \frac{h}{Ti} \qquad \text{: integral gain}$$

$$Ad = \frac{(2 \times Td - N \times h)}{(2 \times Td + N \times h)} \qquad \text{: derivation gain}$$

$$Bd = \frac{(2 \times K \times N \times Td)}{(2 \times Td + N \times h)}$$

$$A0 = \frac{h}{Tt} \qquad \text{: anti-windup gain}$$

Step 2 : Read SV and PV value

PV = adin(ch1)

Step 3: Calculate the proportional term.

P = K × (b × SV – PV)

Step 4 : Update the derivative term. (initial value of D = 0)

D = As × D – Bd × (PV – PV_old)

Step 5 : Calculate the MV. (initial value of I = 0)

MV = P + I + D

Step 6 : Check the actuator is saturated or not.

U = sat(MV, U_low, U_high)

Step 7 : Output the MV value to the D/A module

Step 8 : Update the integral term.

I = I + bi × (SV – PV) + A0 × (U – MV)

Step 9 : Update the PV_old value.

PV_old = PV

15.3 Function blocks
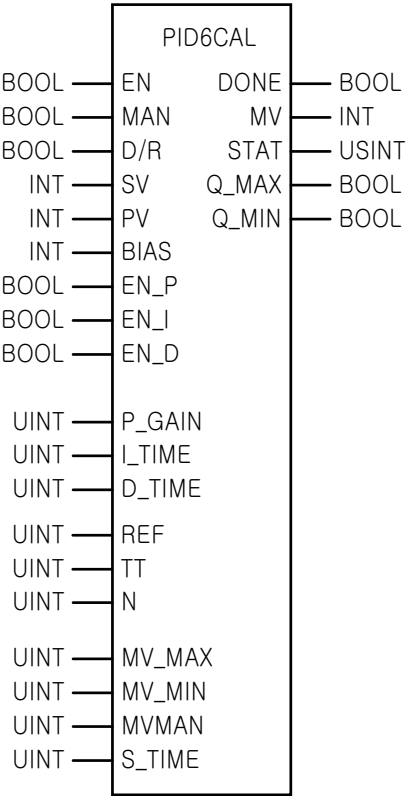
For the PID operation of GM6-CPUB and GM6-CPUC, following 2 function blocks are included in the GMWIN software. (version 3.2 or later)

No	Name	Description
1	PID6CAL	Perform the PID operation
2	PID6AT	Perform the auto tuning operation

Remarks

1. GM6 PID function blocks do not support array type.

2. Refer the GMWIN manual for the registration and running of function block.

3. GM6-CPUA does not support PID operation.

15.3.1 The function block for PID operation (PID6CAL)

Function block	Description

PID6CAL
 	
BOOL — EN DONE — BOOL
BOOL — MAN MV — INT
BOOL — D/R STAT — USINT
INT — SV Q_MAX — BOOL
INT — PV Q_MIN — BOOL
INT — BIAS
BOOL — EN_P
BOOL — EN_I
BOOL — EN_D
 	
UINT — P_GAIN
UINT — I_TIME
UINT — D_TIME
 	
UINT — REF
UINT — TT
UINT — N
 	
UINT — MV_MAX
UINT — MV_MIN
UINT — MVMAN
UINT — S_TIME
	Input
	  EN : enable signal of the PID6CAL F/B
	  MAN : manual operation mode
	        (0 : auto, 1 : manual)
	  D / R : select direction of operation
	        (0 : forward, 1 : reverse)
	  SV : set value data input
	        (input range : 0 ~ 4000)
	  PV : present value data input
	  BIAS : feed forward or offset value input for
	        disturbance compensation
	        (input range : 0 ~ 4000)
	  EN_P : enable signal of proportional control
	        (0 : disable, 1 : enable)
	  EN_I : enable signal of integral control
	        (0 : disable, 1 : enable)
	  EN_D : enable signal of derivative control
	        (0 : disable, 1 : enable)
	  P_GAIN : the proportional gain constant
	        (range : 0.01 ~ 100.00)
	  I_TIME : the integration time
	        (range : 0.0 ~ 2000.0)
	  D_TIME : the deviation time
	        (range : 0.0 ~ 2000.0)
	  MV_MAX : the maximum value of MV
	        (range : 0 ~ 4000)
	  MV_MIN : the minimum value of MV
	        (range : 0 ~ 4000)
	  MVMAN : the input data of manual operation mode
	        (range : 0 ~ 4000)
	  S_TIME : operation scan time
	        (range : 0.1 ~ 10)
	  REF : the reference value
	        (range : 0.1 ~ 1)
	  TT : tracking time constant
	        (range : 0.01 ~ 10.00)
	  N : high frequency noise depression ratio
	        (range : 1 ~ 10)
	
	Output
	  DONE : completion flag of PID operation
	  MV : output manipulation value
	        (range : 0 ~ 4000)
	  STAT : error code output
	  Q_MAX : shows MV is limited with maximum value
	  Q_MIN : shows MV is limited with minimum value

1) SV (setting value : the designated value) and PV (process value : present value) of GM6 PID operation have the range 0 ~ 4000. The range is set with the consideration of the resolution of A/D and D/A module of GM6 series (12 bits) and offset value.

2) The BIAS data is used for the compensation of offset in the proportional control.

3) In GM6-CPUB and GM6-CPUC, only the following 4 operation modes are available. Other operation modes, such as PD or I, are not permitted.

No.	EN_P	EN_I	EN_D	Operation
1	1 (enable)	0 (disable)	0 (disable)	P operation
2	1 (enable)	1 (enable)	0 (disable)	PI operation
3	1 (enable)	1 (enable)	1 (enable)	PID operation
4	0 (disable)	0 (disable)	0 (disable)	On/Off operation

4) The GM6 CPU module can handle only integer, not the floating point type. Therefore, to enhance the accuracy of PID operation, the PID6CAL function block is designed to input the P_GAIN data as the 100 times scaled up. For example, if the designated P_GAIN is 98, actual input data of P_GAIN should be 9800. If the designated P_GAIN is 10.99, input 1099 to the P_GAIN.

5) I_TIME and D_TIME are 10 times scaled up. For example, input 18894 if the designated I_TIME value is 1889.4. The range of actual input is 0 ~ 20000.

6) S_TIME is the period of reading data (sampling), and also 10 times scaled up. Generally, it should be synchronized with external trigger input (EN input of function block) to perform proper PID operation. The range of sampling time is 0.1 ~ 10 seconds, and actual input range is 0 ~ 100.

7) REF may be useful parameter according to the control system type, especially velocity, pressure, or flux control system. The REF input is also 10 times scaled up, and the actual range is 0 ~ 10.

8) TT (tracking time constant) parameter is used to cancel anti-windup operation. The range of TT is 0.01 ~ 10 and the actual input range that are 100 times scaled up is 0 ~ 1000.

9) N (high frequency noise depression ratio) parameter is used for derivative control operation, and shows the ratio of high frequency noise depression. If there is a lot of high frequency noise in the control system, select the N value as higher value. Otherwise, leave the N parameter as 1. The range of N is 0 ~ 10 and it is not scaled up, so input the designated value directly.
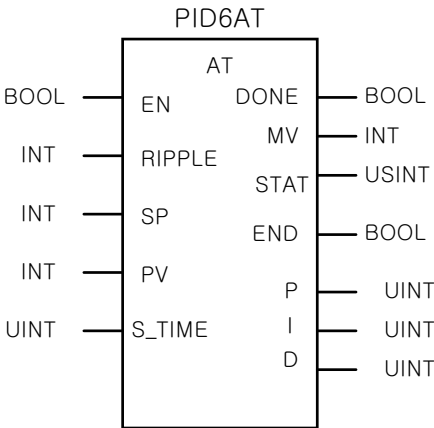
15.3.2 The error code of PID6CAL F/B

The following table shows error codes and descriptions of PID6CAL function block.

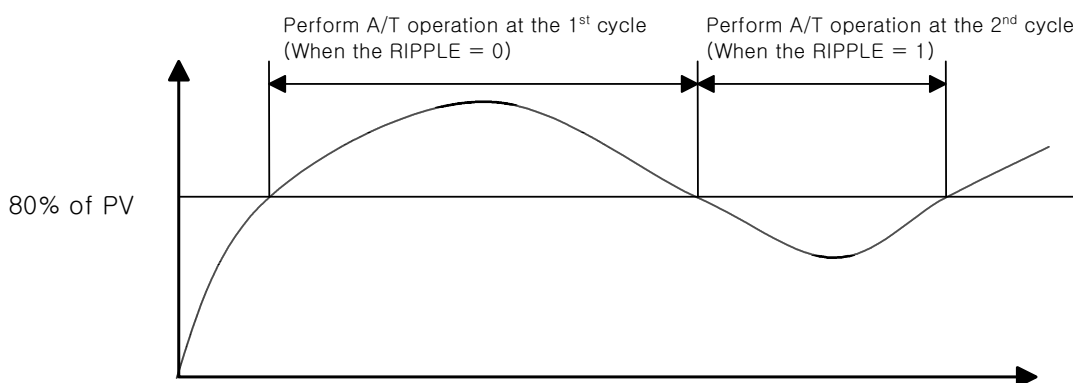| Error code (STAT output)	Type	Description	Countermeasure
0	Local	Normal operation	
1		SV is out of range	Change the SV within 0 ~ 4000
2		MVMAN is out of range	Change the MVMAN within 0 ~ 4000
3		P_GAIN is out of range	Change the P_GAIN within 0 ~ 10000
4		I_TIME is out of range	Change the I_TIME within 0 ~ 20000
5		D_TIME is out of range	Change the D_TIME within 0 ~ 20000
6		S_TIME is out of range	Change the S_TIME within 0 ~ 100
7		REF is out of range	Change the REF within 0 ~ 10
8		TT is out of range	Change the TT within 0 ~ 1000
9		N is out of range	Change the N within 0 ~ 1000
10		EN_I and/or EN_D is set as 1 when EN_P is 0	Only P, PI, and PID controls are available with GM6-CPUB and GM6-CPUC. Please change the setting of EN_P, EN_I, and EN_D by reference to the chapter 15.3.1.
40		CPU type is mismatched	Replace the CPU module with GM6-CPUB or GM6-CPUC.

Remarks

1. Please be careful to input 100 times scaled up values for P_GAIN and TT.

2. I_TIME, D_TIME, S_TIME, and REF are 10 times scaled up, not 100 times.

15.3.3 Auto tuning function block (PID6AT)

Function block	Description

PID6AT
 	
AT
BOOL — EN   DONE — BOOL
            MV — INT
INT — RIPPLE
          STAT — USINT
INT — SP   END — BOOL
INT — PV
                P — UINT
UINT — S_TIME   I — UINT
                D — UINT
	Input
	  EN : enable input of function block
	  SV : set value (goal value) data input
	        (range : 0 ~ 4000)
	  PV : present value input
	        (range : 0 ~ 4000)
	  S_TIME : scan time input (sampling interval)
	            (range : 0 ~ 100)
	  RIPPLE : select the wave form to be used for auto
	            tuning operation. Select 1 in general case.
	Output
	  DONE : Turn on whenever the auto tuning operation
	            is completed.
	  END : Turns on when the F/B operation is
	          completed with no error, and keep the status
	          until next F/B execution
	  STAT : shows the error code
	  MV : the manipulated value of current loop on which
	        the auto tuning operation is performed.
	        (range : 0 ~ 4000)
	  P : the proportional gain constant obtained by auto
	      tuning operation. (range : 0.01 ~ 100.00)
	  I : the integral time constant obtained by auto tuning
	      operation.
	  D : the derivative time constant obtained by auto
	      tuning operation

1) SV (setting value : the designated value) and PV (process value : present value) of GM6 PID operation have the range 0 ~ 4000. The range is set with the consideration of the resolution of A/D and D/A module of GM6 series (12 bits) and offset value. When setting the SV or PV, please be careful convert the analog value of control object (temperature, velocity, etc.) to digital value that are the output of A/D convert module. For example, assume that PID control is used for temperature control with Pt100 (operation range : 0 °C ~ 250 °C), and the goal value is 100 °C. The equivalent digital output of A/D module (voltage output range : 1 ~ 5V) is 1600 if the A/D module outputs 0 (1V) with 0 °C, and 4000(5V) with 250 °C. Therefore, the input of SV should be 1600, not 2.

2) S_TIME is the period of reading data (sampling), and 10 times scaled up for more precious operation. Generally, it should be synchronized with external trigger input (EN input of function block) to perform proper PID operation. The range of sampling time is 0.1 ~ 10 seconds, and actual input range is 0 ~ 100.

3) The GM6-CPUB and GM6-CPUC module perform auto-tuning operation based on the frequency response method. PID parameters are obtained by On/Off operation during 1 cycle of PV variation. The RIPPLE parameter shows at which cycle the CPU module will perform auto-tuning operation. If 0 is selected, the CPU will get PID parameters during the first cycle of PV variation. If 1 is selected, the second cycle will be used. (refer Fig. 12-15 for detailed information) Other choice of RIPPLE parameter is not allowed. In general case, select 1 for proper auto-tuning operation. The On/Off operation will be occur at the 80% of PV value.

15.3.4 Error codes of auto-tuning function block (PID6AT)

The following table shows error codes and descriptions of PID6AT function block.

Error code (STAT output)	Type	Description	Countermeasure
0	Local	Normal operation	
1		SV is out of range	Change the SV within 0 ~ 4000
2		PV is out of range	It may caused by fault of A/D module. Check the A/D module.
3		S_TIME is out of range	Change the S_TIME within 0 ~ 100
4		CPU type is mismatched	Replace the CPU module with GM6-CPUB or GM6-CPUC.

15.4 Programming

15.4.1 System configuration
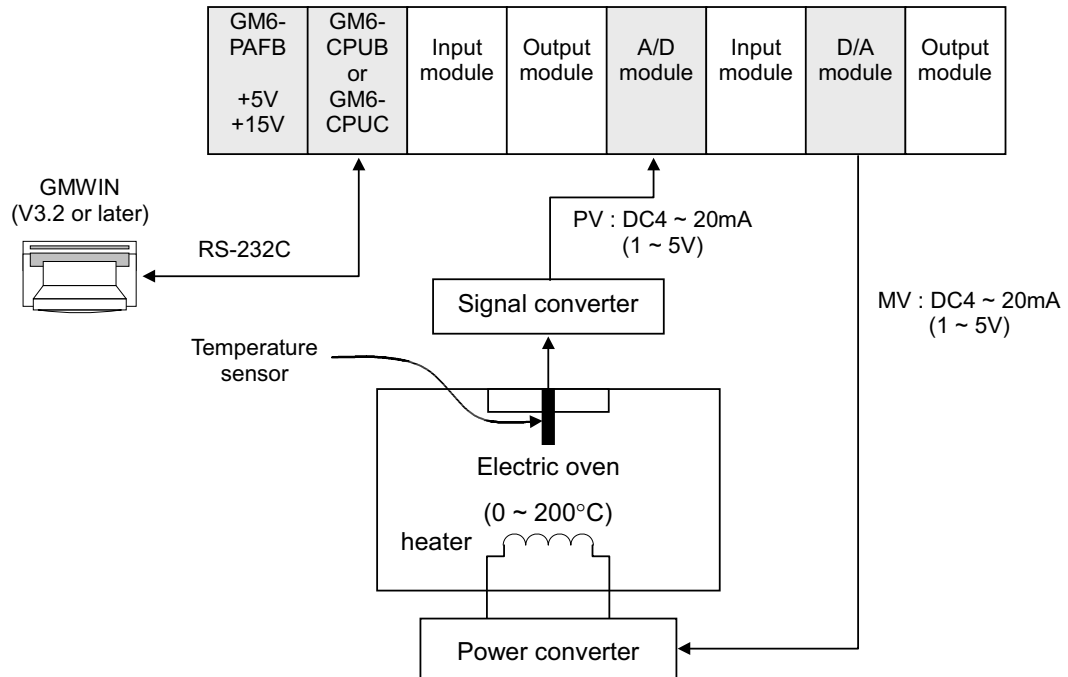
15.4.2 Initial setting

1) PID operation parameters

    a) Auto / Manual operation setting   : Auto

    b) Forward / Reverse operation   : Forward

    c) SV setting   : 1600 (100°C)

    d) BIAS setting   : 0 (If only P control is used, input proper value other 0)

    e) EN_P, EN_I, EN_D setting   : EN_P=1, EN_I=1, EN_D=1 (PID operation)

    f) REF, TT, N   : REF=10, TT=5-, N=1

    g) MV_MAX, MV_MIN, MVMAN   : MV_MAX=4000, MC_MIN=0, MAMAN=2000

    h) S_TIME   : S_TIME=100 (sampling time = 10 seconds)

2) Auto-tuning parameters

    a) PV setting   : 1600 (100°C)

    b) S_TIME   : S_TIME=100 (sampling time = 10 seconds)

3) A/D module setting

   a) Channel setting   : use channel 0

   b) Output data type   : $-48 \sim 4047$

   c) Input processing   : Sampling

4) D/A module setting

   a) Channel setting   : use channel 0
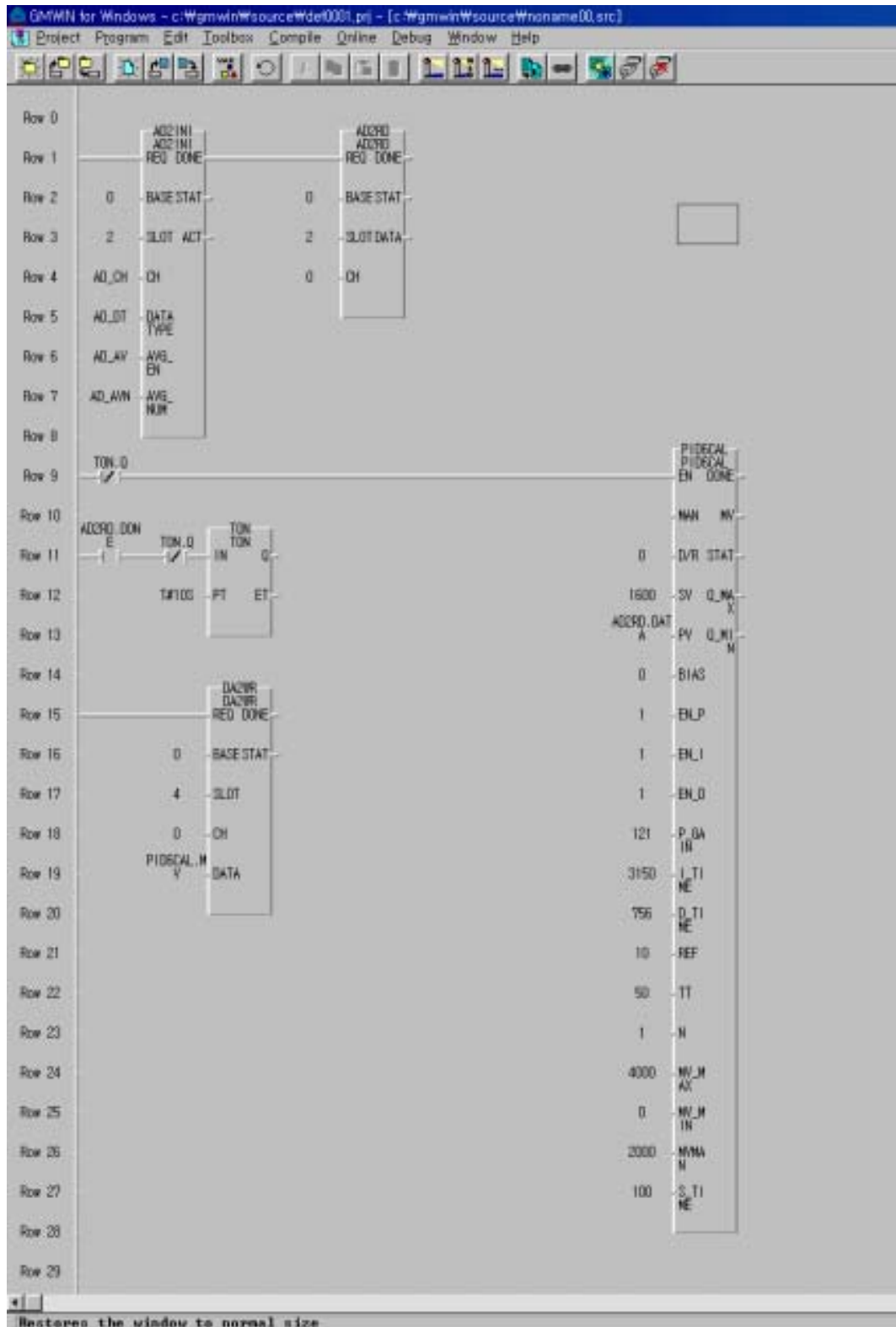
15.4.3 Program description

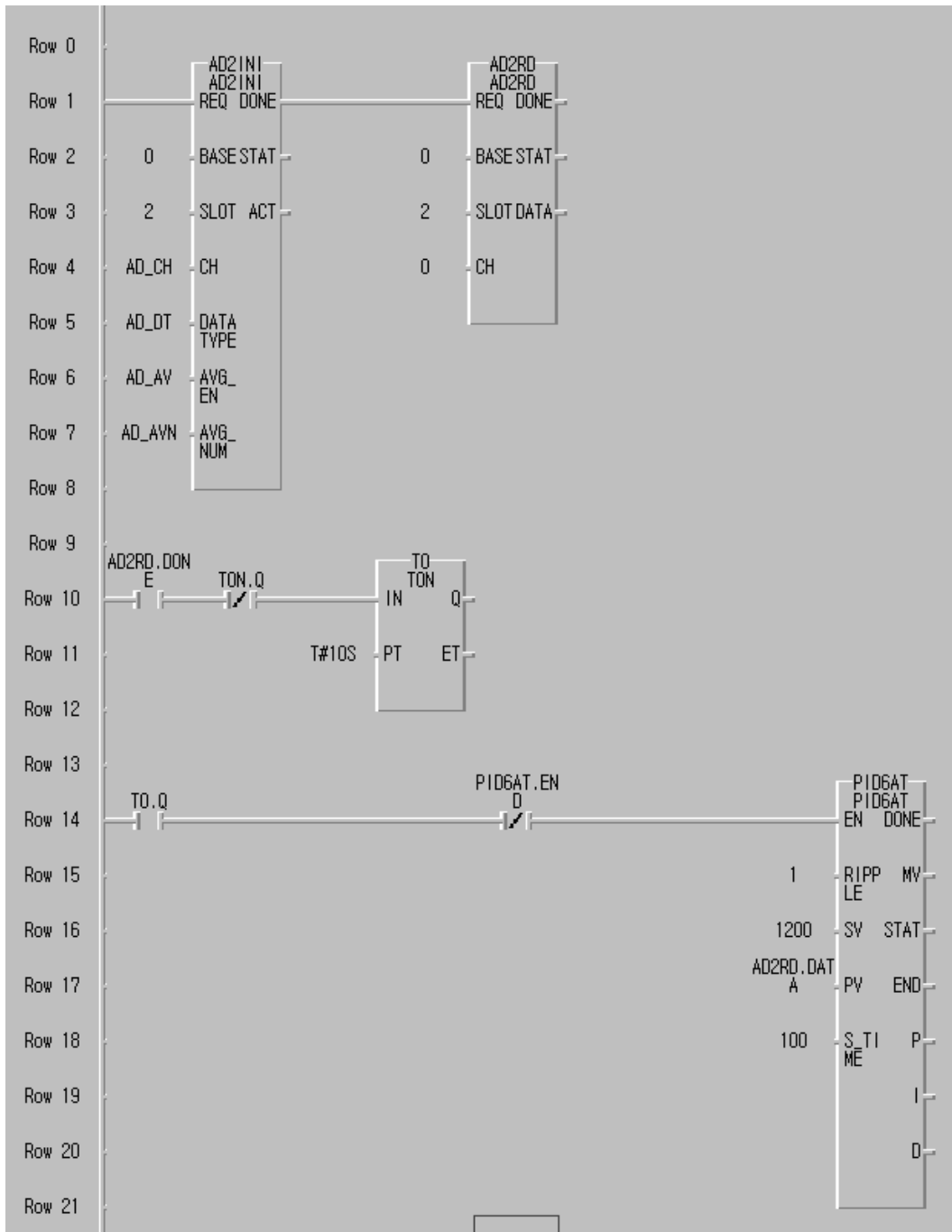#### 15.4.3.1　Use only PID operation (without A/T function)

1) Convert the measured temperature (0 ~ 250°C) to current signal (4 ~ 20mA), and input the current signal to the channel 0 of A/D module. Then, the A/D module converts the analog signal to digital value (0 ~ 4000)

2) PID6CAL function block will calculate manipulate value (MV : 0 ~ 4000) based on PID parameter settings (P_GAIN, I_TIME, D_TIME, etc.) and PV from A/D module. Then, the calculated MV is output to the channel 0 of D/A module.

3) D/A module will convert the MV (0 ~ 4000) to analog signal (4 ~ 20mA) and output to the actuator (power converter).

15.4.3.2　Use PID operation with A/T function

1) Convert the measured temperature (0 ~ 250°C) to current signal (4 ~ 20mA), and input the current signal to the channel 0 of A/D module. Then, the A/D module converts the analog signal to digital value (0 ~ 4000)

2) A/T function block will calculate manipulate value (MV : 0 ~ 4000) based on the SV and PV from A/D module. Simultaneously, the A/T module will calculate P,I and D parameters.

3) The END output of A/T module will be 1 when the A/T operation is completed. Then, PID module will start operation with PID parameters that are calculated by A/T module.

4) D/A module will convert the MV (0 ~ 4000) to analog signal (4 ~ 20mA) and output to the actuator (power converter).
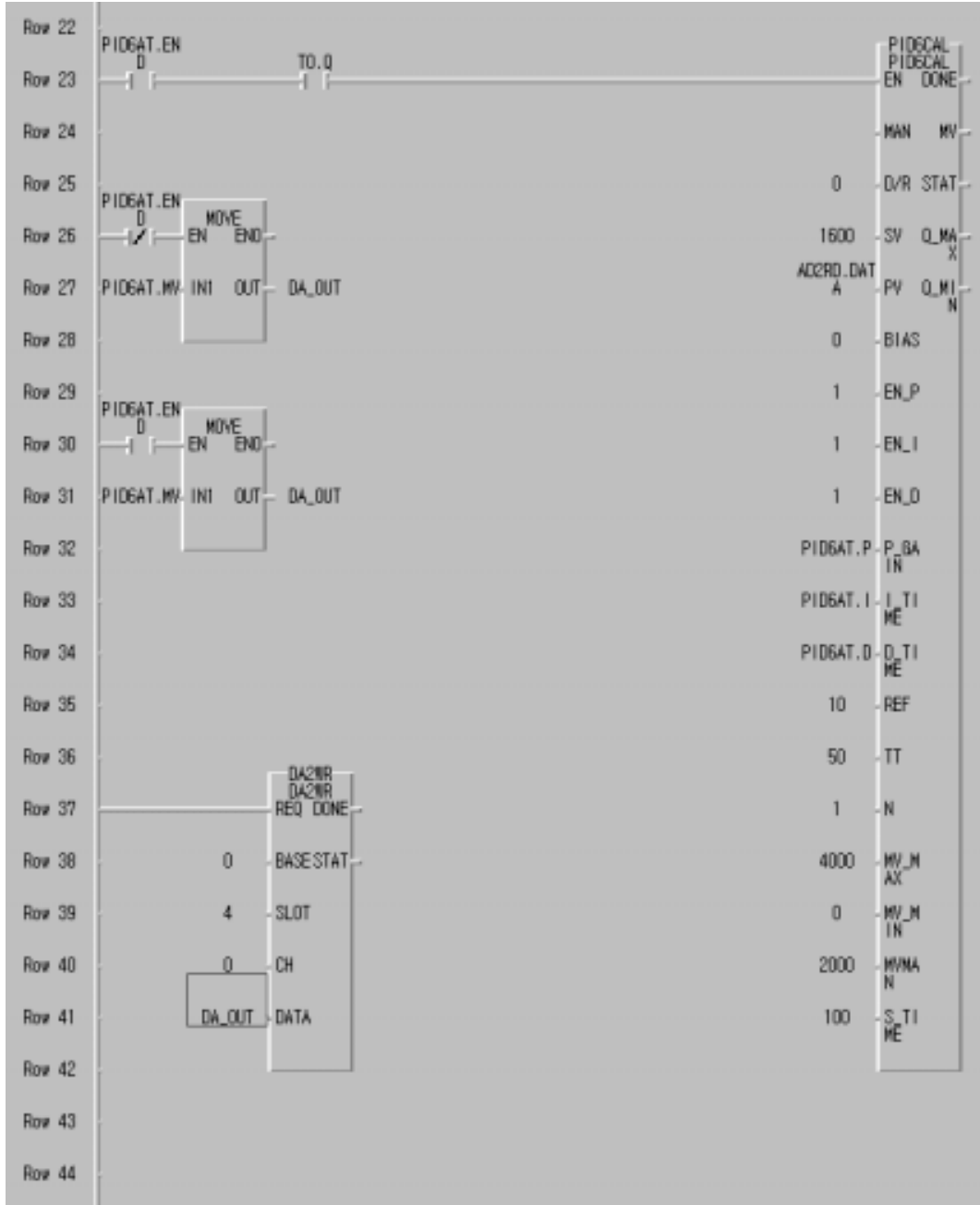
[Example program of 15.4.3.1]

[Example program of 15.4.3.2]

```
Row 0
                    AD2INI                        AD2RD
                    AD2INI                        AD2RD
Row 1               REQ  DONE                     REQ  DONE

Row 2        0      BASE STAT          0          BASE STAT

Row 3        2      SLOT  ACT          2          SLOT DATA

Row 4      AD_CH    CH                 0          CH

Row 5      AD_DT    DATA
                    TYPE
Row 6      AD_AV    AVG_
                    EN
Row 7      AD_AVN   AVG_
                    NUM
Row 8

Row 9     AD2RD.DON                    TO
               E       TON.Q          TON
Row 10       ┤ ├       ┤/├        IN        Q

Row 11                       T#10S  PT       ET

Row 12

Row 13                                  PID6AT.EN             PID6AT
             TO.Q                           D                 PID6AT
Row 14       ┤ ├                          ┤/├               EN  DONE

Row 15                                                1      RIPP   MV
                                                            LE
Row 16                                              1200     SV   STAT

Row 17                                           AD2RD.DAT
                                                     A       PV   END
Row 18                                              100      S_TI  P
                                                            ME
Row 19                                                             I

Row 20                                                             D

Row 21
```

(continue to next page)

15-25

[Example program of 15.4.3.2] (continued)

Row 22		
	PID6AT.EN	PID6CAL
	D	PID6CAL
Row 23	—┤ ├———————┤├├—	EN DONE—
Row 24		—MAN MV—
Row 25	PID6AT.EN	0 —D/R STAT—
	D MOVE	
Row 26	—┤/├—EN END—	1600 —SV Q_MA—
		X
Row 27	PID6AT.MV—IN1 OUT—DA_OUT	AD2RD.DAT —PV Q_MI—
		A N
Row 28		0 —BIAS
Row 29	PID6AT.EN	1 —EN_P
	D MOVE	
Row 30	—┤ ├—EN END—	1 —EN_I
Row 31	PID6AT.MV—IN1 OUT—DA_OUT	1 —EN_D
Row 32		PID6AT.P —P_GA
		IN
Row 33		PID6AT.I —I_TI
		ME
Row 34		PID6AT.D —D_TI
		ME
Row 35		10 —REF
Row 36		50 —TT
	DA2WR	
	DA2WR	
Row 37	REQ DONE—	1 —N
Row 38	0 —BASESTAT—	4000 —MV_M
		AX
Row 39	4 —SLOT	0 —MV_M
		IN
Row 40	0 —CH	2000 —MVMA
		N
Row 41	DA_OUT —DATA	100 —S_TI
		ME
Row 42		
Row 43		
Row 44		

Chapter 16　Built-in high speed counter of GM6-CPUC

16. Built-in high speed counter of GM6-CPUC

16.1. Introductions

This chapter describes the specification, handling, and programming of built-in high speed counter of GM6-CPUC module. The built-in high speed counter of GM6-CPUC (Hereafter called HSC) has the following features;

- 3 counter functions as followings

    - 1-phase up / down counter   : Up / down is selected by user program

    - 1-phase up / down counter   : Up / down is selected by external B phase input

    - 2-phase up / down counter   : Up / down is automatically selected by the phase difference between phase A and B.

- Multiplication (1, 2, or 4) with 2-phase counter

    - 2-phase pulse input multiplied by one : Counts the pulse at the leading edge of phase A.

    - 2-phase pulse input multiplied by two : Counts the pulse at the leading / falling edge of phase A.

    - 2-phase pulse input multiplied by four : Counts the pulse at the leading / falling edge of phase A and B

16.2. Performance specifications

Items		Specifications
Input signal	Types	Phase A, Phase B, Preset
	Rated level	24VDC (13mA)
	Signal type	Voltage input
Counting range		0 ~ 16,777,215 (Binary 24 bits)
Max. counting speed		50k pps
Up / Down selection	1-phase	Sequence program or B-phase input
	2-phase	Auto-select by phase difference of phase A and B
Multiplication		1, 2, or 4
Preset input		Sequence program or external preset input

16.3. Input specifications

16.3.1. Function of input terminals

Items		Specifications
A / B phase	Rated input	24VDC (13mA)
	On voltage	14VDC or higher
	Off voltage	2.5VDC or lower
Preset input	Rated input	24VDC (10mA)
	On voltage	19VDC or higher
	Off voltage	6V or lower
	On delay time	Less than 1.5ms
	Off delay time	Less than 2ms

16.3.2. Names of wiring terminals

No. of terminal	Input signal
1	A phase input
2	B phase input
3	COM
4	Preset input
5	Preset COM
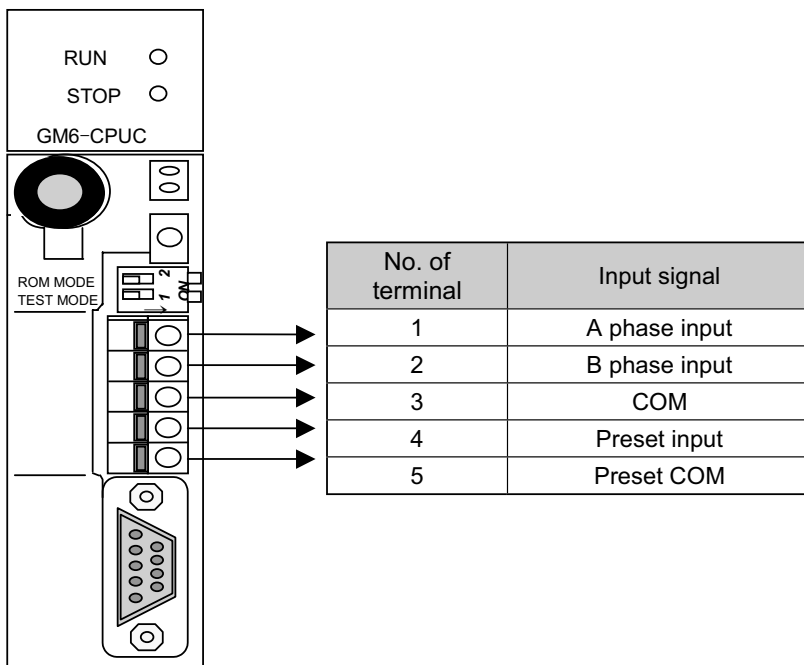
16.3.3. External interface circuit

	Internal circuit	No. of terminal	Signal type	Operation voltage	
Input	3.3KΩ 820Ω	1	A-phase pulse input 24VDC	ON	14 ~ 26.4 VDC
				OFF	Less than 2.5VDC
		2	B-phase pulse input 24VDC	ON	14 ~ 26.4 VDC
				OFF	Less than 2.5VDC
		3	COM		
Input	3.3KΩ 270Ω	4	Preset input 24V	ON	19 ~ 26.4 V
				OFF	6 V or less
		5	Preset COM		

16.4. Wiring

16.4.1. Wiring instructions

A high speed pulse input is sensitive to the external noise and should be handled with special care. When wiring the built-in high speed counter of GM6-CPUC, take the following precautions against wiring noise.

1) Be sure to use shielded twisted pair cables. Also provide Class 3 grounding.
2) Do not run a twisted pair cable in parallel with power cables or other I/O lines which may generate noise.
3) Before applying a power source for pulse generator, be sure to use a noise-protected power supply.
4) For 1-phase input, connect the count input signal only to the phase A input; for 2-phase input, connect to phases A and B.

16.4.2. Wiring examples

1) Voltage output pulse generator
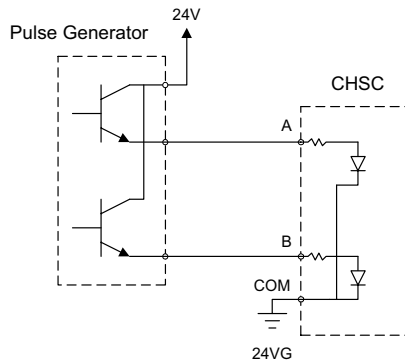
2) Open collector output pulse generator
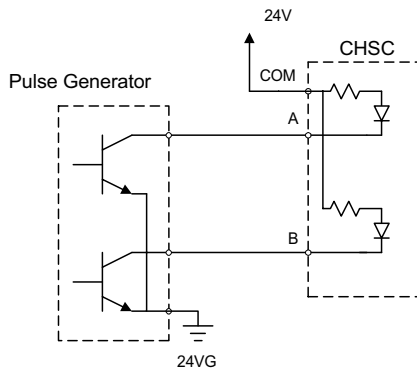
16.5. Programming

16.5.1. Function block (F/B)

CHSC_WR

CHSC_WR

GM1	GM2	GM3	GM4	GM5	GM6
					●
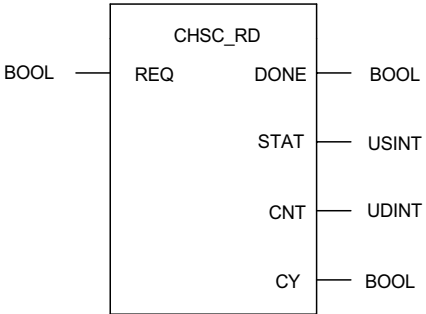
FUNCTION BLOCK	Description

![CHSC_WR function block]
 	
CHSC_WR
 	
BOOL — REQ   DONE — BOOL
 	
BOOL — PHS   STAT — USINT
 	
USINT — MULT
 	
BOOL — U/D_ I/E
 	
BOOL — CY_R
 	
BOOL — DOWN
 	
BOOL — CT_E
 	
BOOL — PRE_ I/E
	Input REQ : Request signal of F/B execution
	        PHS : Operation modes selection
	               0 (1-phase counter), 1(2-phase counter)
	        MULT : Assign the multiplication factor
	               (MULT = 1, 2, or 4)
	        U/D_I/E : Assign the count direction (up/down)
	                  selector
	                 0 : Set by sequence program
	                 1 : Set by B-phase input signal
	                      (1:up-count, 0:down-count)
	        CY_R : Carry reset signal (1: reset).
	        DOWN : Select the count direction (0:up/1:down)
	                when the counter is set as 1-phase counter
	                and up/down is selected by sequence
	                program. (PHS=0 & U/D_I/E=0)
	        CT_E : Counter enable signal
	               (0 : Counter disable, 1 : Enable)
	        PRE_I/E : Assign PRESET input
	                 0 : PRESET by sequence program
	                 1 : PRESET by external input at the
	                      PRESET terminal
	
	Output
	        DONE : Turns on after the F/B is executed with no
	                error.
	        STAT : Indicate the operation status of F/B

- The MULT input will be dummy input when the HSC is set as 1-phase counter (PHS = 0). When the HSC is set as 2-phase counter, the U/D_I/E and DOWN input will be dummy input. (PHS = 1)

- The current value of HSC will be cleared as 0 when the CT_E (counter enable) is 0.

CHSC_RD

Read the current value and operation status of HSC
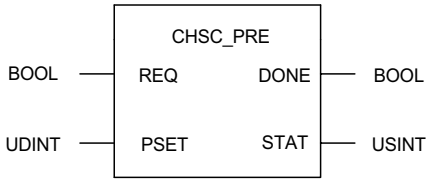
GM1	GM2	GM3	GM4	GM5	GM6
					●

FUNCTION BLOCK	Description

CHSC_RD
 	
BOOL — REQ   DONE — BOOL
 	
STAT — USINT
 	
CNT — UDINT
 	
CY — BOOL
	Input REQ : Request signal for F/B execution
	
	Output
	     DONE : Turns on after the F/B is executed with no error.
	     STAT : Indicates the operation status of F/B
	      CNT : The current value of HSC
	           (0 ~ 16,777,215)
	       CY : Carry flag (0 : OFF, 1 : ON)

CHSC_PRE

CHSC_PRE

Set the preset value of HSC

GM1	GM2	GM3	GM4	GM5	GM6
					●
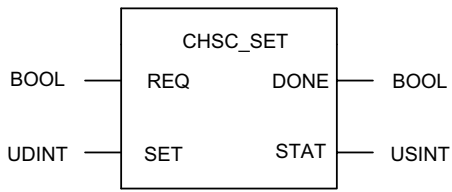
FUNCTION BLOCK	Description

CHSC_PRE
 	
BOOL — REQ   DONE — BOOL
 	
UDINT — PSET   STAT — USINT
	Input REQ : Request signal for F/B execution
	     PSET : Set the preset value (0 ~ 16,777,215)
	
	Output
	     DONE : Turns on after the F/B is executed with no
	         error.
	     STAT : Indicates the operation status of F/B

- When the PRE_I/E is set as 0 (Preset input by sequence program), the current value of HSC is changed as the assigned preset value with the rising edge of REQ input.
- When the PRE_I/E is set as 1 (Preset input by external preset input), the current value of HSC is changed as the assigned preset value with the rising edge of external preset input. At this time, the REQ input of CHSC_PRE is ignored.
- The CY output is set off while the CHSC_PRE F/B is executing.
- The CHSC_PRE F/B is disabled while the CT_E input of CHSC_WR F/B is 0 (Counter disabled).

CHSC_SET

CHSC_SET

Assign a setting value to be compared

with the current value of HSC

GM1	GM2	GM3	GM4	GM5	GM6
					●

FUNCTION BLOCK	Description

CHSC_SET
 	
BOOL ── REQ   DONE ── BOOL
 	
UDINT ── SET   STAT ── USINT
	Input REQ : Request signal for F/B execution
	          SET : Set a setting value (0 ~ 16,777,215)
	
	Output
	          DONE : Turns on after the F/B is executed with no
	                    error.
	          STAT : Indicates the operation status of F/B

Run a task program when the current value of HSC reaches to the setting value.

To run a task program, define a high speed counter task program as following figure,

and write a task program.
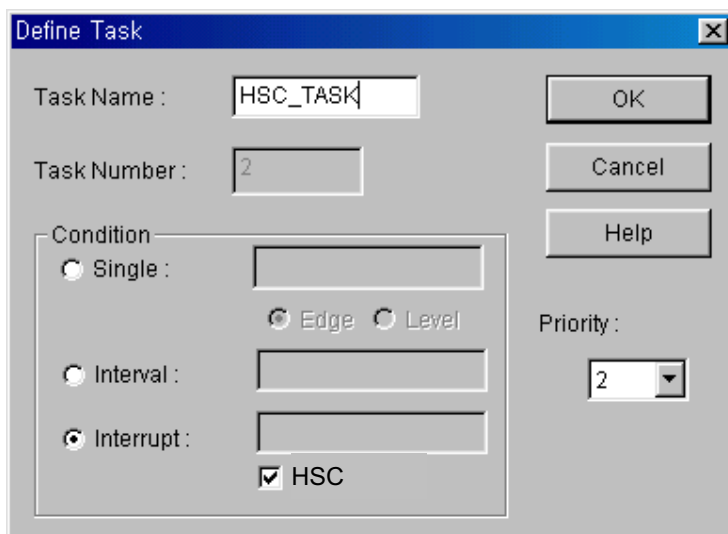
16.5.2 Error code of F/B

The following table shows error codes appear at the STAT output.

Error code	Description
00	No error
01	Built-in high speed counter is not found (GM6-CPUA, GM6-CPUB CPU module)
02	Input data error at MULT input of CHSC_WR (2 Phase Mode 에서 1, 2, 4 이외의 숫자일 때)
03	PSET (CHSC_PRE) or SET (CHSC_SET) is out of specified range (0 ~ 16,777,215).
04	Execute Preset command while the HSC is disabled status

Appendix 1. System Definitions

1) Basic Parameters

The basic parameters are necessary for operation of the PLC and used to allocate memory, set the restart mode and set the scan watch dog time, etc.
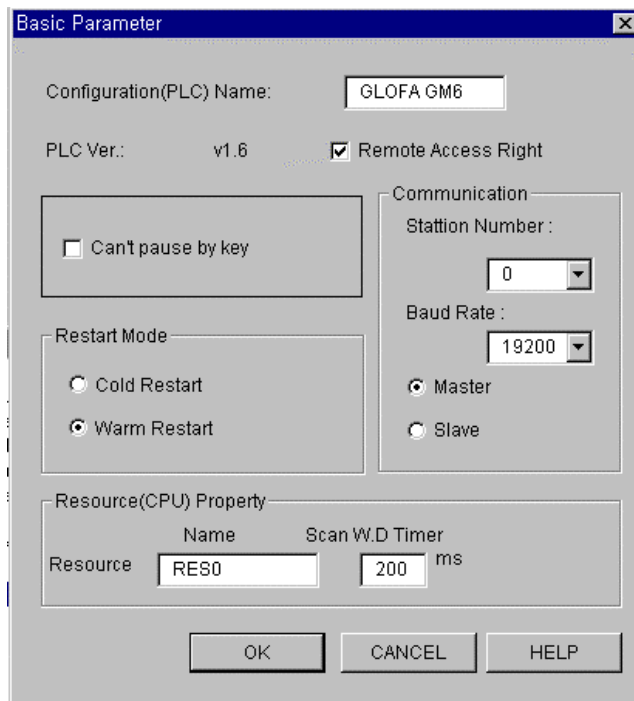
(1) Configuration (PLC) Name

- It is a representative name for the PLC system. It is used to designate this PLC system when a network system is configured using communication modules.

(2) Enabling/Disabling the control of the PLC via communications

- This parameter is used to enable or disable the remote control of this PLC system through the FAM or computer link module, etc. except for the GMWIN. If this parameter has been set to enable, change of the operation mode and download of programs are available via communications.

(3) Restart Mode

- This parameter is used to set the restart mode in the PLC system. When the system re-starts, one of the 'cold restart' or 'warm restart' is selected in compliance with the parameter setting.

(4) Resource (CPU) Name

- Resource Name is the name that each CPU module configuring the PLC has. When configuring a network system the name is used to designate each CPU module that is used the system.
- Only one CPU module can be mounted in the GM3/4 series, therefore, only the resource 0 is valid.

(5) Scan Watch Dog Time

- This parameter is used to set the maximum allowable execution time of an user program in order to supervisor   its normal or abnormal operation.
- Only one CPU module can be mounted in the GM3/4 series, therefore, scan watch dog is valid to only the resource 0.

(6) Unable to Pause by mode setting switch

- Set : When switch mode is changed from run to pause/remote (RUN → PAU/REM),
  PLC is operated as Local Pause mode.
- Default (do not set) : When switch mode is changed from run to pause/remote (RUN → PAU/REM),
  PLC is operated as Remote RUN mode.

2) I/O Configuration Parameters

These parameters are used to set the configuration of a system that will be operated. They set the modules that will be mounted and operated onto their own slot in the base unit. If a parameter that has been set and the real mounted module are different, the operation will not be executed. When writing a new project I/O configuration parameters will be all set to default (DEF_MODULE).

If I/O configuration parameters are set to default, the operation starts on the basis of the configuration of the real mounted module when the power is applied. Therefore, though a power failure had occurred during normal operation or the system configuration had been changed due to slip-out of a mounted module, operation starts and continues when the power has been re-applied because the system considers that it is a normal operation state. To prevent this error, be sure to set correctly the I/O configuration parameters complying with the real modules that shall be mounted and operated.

<I/O Parameters Setting List>
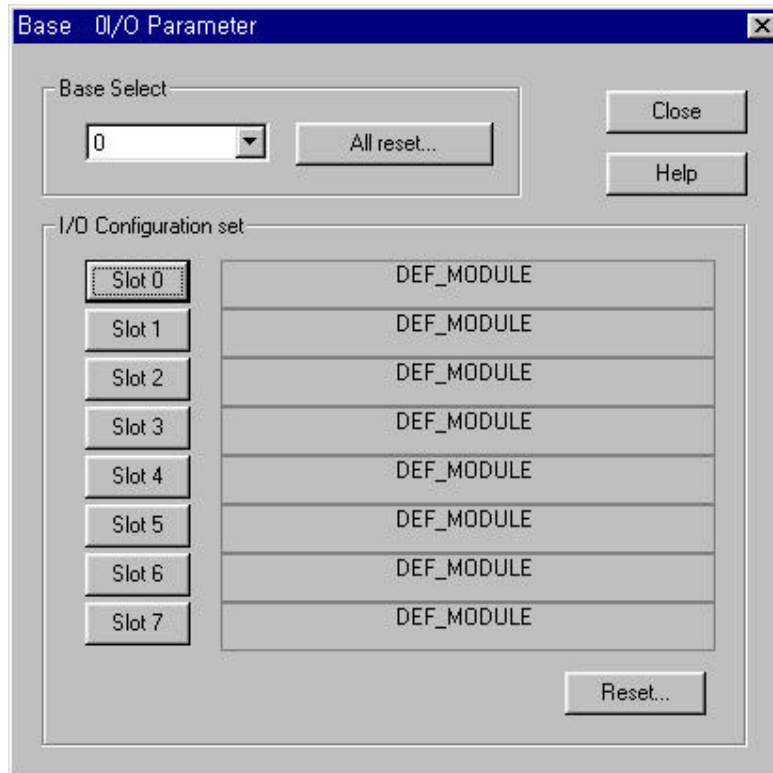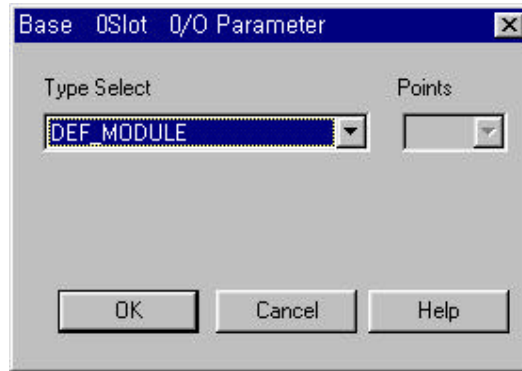
| Keywords	Description	Applicable Modules
DC input	DC input module	G6I-D22A(16 points), G6I-D24A(32 points), G6I-D22B(16 points) G6I-D24B(32 points)
110 VAC input	110 VAC input module	G6I-A11A(8 points)
220 VAC input	220 VAC input module	G6I-A21A(8 points)
Relay output	Relay output module	G6Q-RY2A(16 points)
SSR output	Triac output module	G6Q-SS1A(8 points)
TR output	Transistor output	G6Q-TR2A(16 points), G6Q-TR4A(32 points)
A/D	A/D conversion module	G6F-AD2A(4 channels)
DAV, DAI	D/A conversion module	G6F-DA2V(4channels, voltage type) G6F-DA2I(4channels, current type)
HSC	High speed counting module	G6F-HSCA(1 channels)
GLOFA Fnet	Fnet I/F module	G6L-FUEA
GLOFA Cnet	Cnet I/F module	G6L-CUEB, G6I-CUEC
DEF_I	All input modules	G6I-D22A(16 points), G6I-D24A(32 points), G6I-D22B(16 points) G6I-D24B(32 points), G6I-A11A(8 points), G6I-A21A(8 points)
DEF_O	All output modules	G6Q-RY2A(16 points), G6Q-SS1A(8 points), G6Q-TR2A(16 points), G6Q-TR4A(32 points)
DEF_IO	All mixed I/O modules	-
DEF_SP	All communications / special modules	• All special modules • All communications modules
DEF_MODULE	All modules	• All input modules • All output modules • All mixed I/O modules • All special modules • All communications modules
DEF_EMPTY	Empty slot	–

3) Communications Parameters

These high speed link parameters are used to set the opposite station for data communications, data and communications cycle when communicating a defined data repeatedly through communication modules.

(For detailed descriptions, refer to the User's Manual relating to data communications)
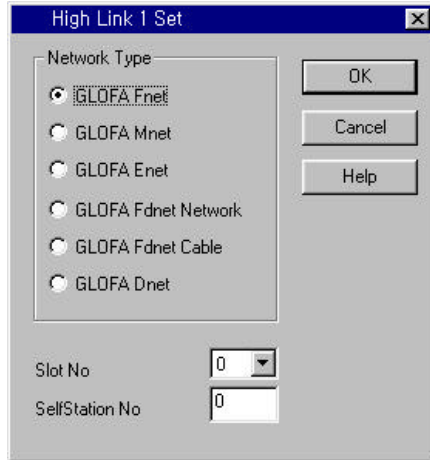
(1) Network type : Used to set the type of the communications module

(2) Slot No. : Location number of slot where the communications module has been mounted.

(3) Local No. : Local number of the module which executes high speed link communications.
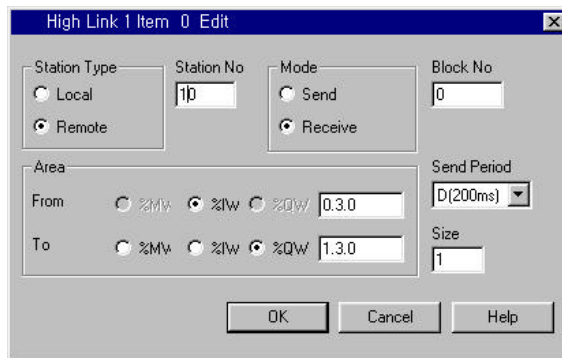
(1) Station type : Type of the communications module in the opposite station. Local or remote will be set.

(2) Station No. : Used to indicate the station that has invoked data during communications.

(3) Mode :   Used to set the communications mode to Send or Receive.

(4) Block No. : Designating number for identification of a data block in the same communications module.

(5) Data communications cycle : Used to set the cycle of sending and receiving of data.

(6) Area: I, Q and M areas should be set by the decimal number or word.

(7) Size : Number of words that will be sent and received.

3) Representative System Warning Flag List

Keyword	Type	Bit No.	Name	Description
_CNF _WAR	WORD	Representative keyword	System warning	This flag treats the below warning flags relating to continuous operation in batch.
_D_BCK_ER	BOOL	Bit 1	Data backup error	This flag indicates
_AB_SD_ER	BOOL	Bit 3	Abnormal shutdown	This flag indicates that the program had been stopped during restore from power failure due to causes such as power off, and then cold restart has been executed and the continuous operation which retains the data is impossible. Usable in the initialization program. Automatically reset when the initialization program has finished. (The same things given above will be applied when the program has been stopped by the 'ESTOP' function)
_TASK_ERR	BOOL	Bit 4	Task collision (plus cycle and external tasks)	This flag indicates that task collision has occurred as execution request for a same task had been repeatedly invoked. (Refer to the flag _TC_BMAP[n] and _TC_CNT[n])
_BAT_ERR	BOOL	Bit 5	Battery fault	This flag detects and indicates that the voltage of the battery, which is used to backup user programs and data memory, is lower than the defined value.
_ANNUN_WR	BOOL	Bit 6	External device warning detection	This representative flag indicates that the user program has detected an ordinary fault of external devices and has written it to the flag _ANC_WB [n].
–	–	Bit 7	–	–
_HSPMT1_ER	BOOL	Bit 8	High speed link parameter 1 error	This representative flag detects error of each high speed link parameter when the high link has been enabled and indicates that high speed link cannot be executed. It will be reset when the high speed link is disabled.
_HSPMT2_ER	BOOL	Bit 9	High speed link parameter 2 error	
_HSPMT3_ER	BOOL	Bit 10	High speed link parameter 3 error	
_HSPMT4_ER	BOOL	Bit 11	High speed link parameter 4 error	

4) Detailed System Error and Warning Flag List

Keyword	Type	Data setting range	Name	Description
_IO_TYER_N	UINT	0 to 15	The number of slot whose module type is inconsistent.	This flag detects that I/O configuration parameters of each slot differ from the real loaded module configuration or a particular module is loaded onto the slot where modules cannot be loaded, and indicates the lowest slot No. of the detected slot numbers.
_IO_TYERR[n]	BYTE	n: 0 to 1	The location of slot where module type is inconsistent.	This flag detects that I/O configuration parameters of each slot differ from the real loaded module configuration or a particular module is loaded onto the slot where modules cannot be loaded, and indicates the slot locations in the bit map of base units.
_IO_DEER_N	UINT	0 to 15	The number of slot where module mounting/dismounting error occurred.	This flag detects that module configuration of each slot has been changed, that is, module mounting/dismounting error has been occurred, and indicates the lowest slot No. of the detected slot numbers.
_IO_DEERR[n]	BYTE	n: 0 to 1	The location of slot where module mounting/dismounting error occurred.	This flag detects that module configuration of each slot has been changed, that is, module mounting/dismounting error has been occurred, and indicates the slot locations in the bit map of base units.
_FUSE_ER_N	UINT	0 to 15	The number of slot where fuse breaks.	This flag detects that fuses of fuse-mounted modules has broken, and indicates the lowest slot No. of the detected slot numbers.
_FUSE_ERR[n]	BYTE	n: 0 to 1	The location of slot where fuse breaks.	This flag detects that fuses of fuse-mounted modules has broken, and indicates the slot locations in the bit map of base units.
_IO_RWER_N	UINT	0 to 15	The number of slot where I/O module read/write occurred.	This flag detects that input modules of a slot cannot be normally read from or written to, and indicates the lowest slot No. of the detected slot numbers.
_IO_RWERR[n]	BYTE	n: 0 to 1	The location of slot where I/O module read/write occurred.	This flag detects that input modules of a slot cannot be normally read from or written to, and indicates the slot locations in the bit map of base units.
_IP_IFER_N	UINT	0 to 15	Special/link module interface error slot No.	This flag detects that initialization cannot be executed for special or link module of a slot, or normal interface is impossible due to module malfunction, and indicates the lowest slot No. of the detected slot numbers.
_IP_IFERR[n]	BYTE	n: o to 1	Special/link module interface error location	This flag detects that initialization cannot be executed for special or link module of a slot, or normal interface is impossible due to module malfunction, , and indicates the slot locations in the bit map of base units.
_ANC_ERR[n]	UINT	n : 0 to 7	External device fatal error	This flag detects fatal error of external devices and its content is written to this flag. A number that identifies error type will be written to each of the sixteen locations. (The number 0 is not allowed)
_ANC_WAR[n]	UINT	n : 0 to 7	External device ordinary error	If the user program indicates a warning on the flag _ANC_WB[n], the bit locations are sequentially written to _ANC_WAR[n] from _ANC_WAR[0] complying with their occurrence sequence.
_ANC_WB[n]	BIT	n: 0 to 127	External device ordinary error bit map	The user program detects ordinary error of external device and the errors are indicated on a bit map. (The number 0 is not allowed)
_TC_BMAP[n]	BIT	n : 0 to 7	Task collision bit map	The flag detects that task collision has occurred because, while a task was being executed or ready for execution, an execution request has occurred for the same task, indicates the errors on a bit map.
_TC_CNT[n]	UINT	n : 0 to 7	Task collision counter	This flag detects task collision occurrence time for each task when executing a user program, indicates the task collision occurrence time.

4) Detailed System Error and Warning Flag List (continued)

Keyword	Type	Data setting range	Name	Description
_BAT_ER_TM	DATE & TIME	—	Batter voltage drop time	The first detection date and time of battery voltage drop are written to this flag. It will be reset if the battery voltage has been restored.
_AC_F_CNT	UINT	0 to 65535	Momentary power failure occurrence count	The accumulated momentary power failure occurrence times during operation in the RUN mode is written to this flag.
_AC_F_TM[n]	DATE & TIME	n : 0 to 15	Momentary power failure history	The times of the latest sixteen momentary power failures are written.
_ERR_HIS[n]		n : 0 to 15	Error history	The times and error codes of the latest sixteen errors are written to this flag. • Stop time : DATE & TIME (8 bytes) • Error code : UINT (2 bytes)
_MODE_HIS[n]		n : 0 to 15	Operation mode change history	The times, operation modes and restart modes of the latest sixteen operation mode changes are written to this flag • Change time : DATE & TIME (8 bytes) • Operation mode : UINT (2 bytes) • Restart : UINT (2 bytes)

* Write is available in user programs.

5) System Operation status Information Flag List

Keyword	Type	Data setting range	Name	Description
_CPU_TYPE	Unit	0 to 16	System type	GM1 : 0, GM2 : 1, (GM3 : 2, GM4 : 3, GM% : 4) (FSM : 5,6), Twofold : 16
_VER_NUM	Unit	-	O/S version No.	System O/S version No.
_MEM_TYPE	Unit	1 to 5	Memory module type	Type of program memory module (0: Unloading state, type : 0 to 5)
_SYS_STATE	Word	Representative keyword	PLC mode and operation status	System operation mode and operation state information
		Bit 0	Local control	Operation mode change is possible only by mode change switch or GMWIIN
		Bit 1	STOP	CPU module operation state
		Bit 2	RUN	
		Bit 3	PAUSE	
		Bit 4	DEBUG	
		Bit 5	Operation mode change factor	Operation mode change by mode change switch
		Bit 6	Operation mode change factor	Operation mode change by GMWIN
		Bit 7	Operation mode change factor	Operation mode change by remote GMWIN
		Bit 8	Operation mode change factor	Operation mode change by communications
		Bit 9	STOP by STOP function	Operation in the RUN mode is stopped by STOP function after the scan has finished
		Bit 10	Force input	Input junction force On/Off is being executed.
		Bit 11	Force output	Output junction force On/Off is being executed
		Bit 12	STOP by ESTOP function	Operation in the RUN mode is directly stopped by ESTOP function.
		Bit 13	-	-
		Bit 14	During monitoring	External monitoring is being executed for programs or variables
		Bit 15	Remote mode ON	Operation in the remote mode
_GMWIN_CNF	Byte	Representative keyword	GMWIN connection state	Connection state between CPU module and GMWIN
		Bit 0	Local GMWIN connection	Local GMWIN connection state
		Bit 1	Remote GMWIN connection	Remote GMWIN connection state
		Bit 2	Remote communications connection	Remote communications connection state
_RST_TY	Byte	Representative keyword	Restart mode information	Restart type of program which is being executed in present. (History)
		Bit 0	Cold restart	See the Section 4.5.1
		Bit 1	Warm restart	
		Bit 2	Hot restart	
_INIT_RUN	Bool	-	During initialization	An initialization program written by the user is being executed
_SCAN_MAX	Unit	-	Maximum scan time (ms)	Maximum scan time is written during operation.
_SCAN_MIN	Unit	-	Minimum scan time (ms)	Minimum scan time is written during operation.
_SCAN_CUR	Unit	-	Present scan time (ms)	Present scan time is continuously updated during operation.
_RTC_TIME[n]	BCD	N : 0 to 7	Present time	BCD data of present time of RTC (Example : 96-01-12-00-00-00-XX) _RTC _TIME[0] : year, _RTC _TIME[1] : month, _RTC _TIME[2] : day, _RTC _TIME[3] : hour, _RTC _TIME[4] : minute, _RTC _TIME[5] : second, _RTC _TIME[6] : day of the week, _RTC _TIME[7] : unused Day of the week : 0 : Mon., 1: Tue., 2: Wed., 3:Thur., 4:Fri., 5: Sat., 6:Sun.
_SYS_ERR	Unit	Error code	Error type	See the Section 12.5 Error Code List

6) System Configuration status Information Flag

(1) User Program Status Information

Keyword	Type	Data setting range	Name	Description
_DOMAN_ST	BYTE	Representative keyword	System S/W configuration information	GM1 : 0, GM2 : 1, (GM3 : 2, GM4 : 3, GM% : 4) (FSM : 5,6), Twofold : 16
		Bit 0	Basic parameter error	Checks and indicates Basic parameter error
		Bit 1	I/O configuration parameter error	Checks and indicates I/O configuration parameter error
		Bit 2	Program error	Checks and indicates Program error
		Bit 3	Access variable error	Checks and indicates Access variable error
		Bit 4	High speed link parameter error	Checks and indicates High speed link parameter error

(2) Operation Mode change switch Status Information

Keyword	Type	Data Setting range	Name	Description
_KEY_STATE	BYTE	Representative keyword	Mode setting switch position	Indicates the state mode setting switch of CPU module
		Bit 0	KEY_STOP	Indicates that the mode setting switch is in the STOP state.
		Bit 1	KEY_RUN	Indicates that the mode setting switch is in the RUN state.
		Bit 2	KEY_PAUSE/REMOTE	Indicates that the mode setting switch is in the PAUSE/REMOTE state.

(3) I/O Module Installation Status Information

Keyword	Type	Data Setting range	Name	Description
_IO_INSTALL[n]	BYTE	n : 0 to 1	I/O module installation location	Locations of slots where I/O modules are loaded are indicated in the bitmap of base units.

7) Communications Flag

- GLOFA Mnet / Fnet / Cnet Flag List

(1) Communication Module Information Flag List

● n is the number of slot where a communications module is loaded. (n = 0 to 7)

Keyword	Type	Applicable Net	Name	Description
_CnVERNO	UINT	Mnet/Fnet/Cnet	Communications module version No.	● Communications module O/S version No.
_CnSTNOH _CnSTNOL	UINT UDINT	Mnet/Fnet/Cnet	Communications module station No.	● Indicates the number which is set on communications module station switch. Mnet : MAC station No. marked on the front of communication module. Fnet : Station switch No. marked on the front of communications module. Cnet : Station No. set by the frame editor _CnSTNOH : Station No. set on the side of RS-232C _CnSTNOL : Station No. set on the side of RS-422
_CnTXECNT	UINT	Mnet/Fnet/Cnet	Communications frame sending error	● Increments by one whenever sending error of communications frame occurs. ● Connection condition of network is evaluated by this value. ● In Cnet, this value is the sum of errors occurred during receiving through RS-232 and RS-422.
_CnRXECNT	UINT	Mnet/Fnet/Cnet	Communications frame receiving error	● Increments by one whenever communications service fails. ● Connection condition of network is evaluated by this value. Overall network communications quantity and program stability are also evaluated by this value.
_CnSVCFCNT	UINT	Mnet/Fnet/Cnet	Communications service processing error	● Indicates the maximum time that is spent until every station connected to network has the token at least one time and sends a sending frame.
_CnSCANAV	UINT	Mnet/Fnet/Cnet	Maximum communications scan time (unit : 1 ms)	● Indicates the average time that is spent until every station connected to network has the token at least one time and sends a sending frame.
_CnSCANMN	UINT	Mnet/Fnet/Cnet	Average communications scan time (unit : 1 ms)	● Indicates the minimum time that is spent until every station connected to network has the token at least one time and sends a sending frame.
_CnLINF	UINT	Mnet/Fnet/Cnet	Minimum communications scan time (unit : 1 ms)	● Indicates operation state of communications module with a word.
_CnLNKMOD	BIT 15		Operation mode (RUN=1, TEST=0)	● Indicates that operation mode of communications module is in the normal operation mode or test mode.
_CnINRING	BIT 14		In-ring (IN_RING = 1)	● Indicates that the communications module can communicates(IN_RING = 1) with other station or not.
_CnIFERR	BIT 13		Interface error (error = 1)	● Indicates that interface with communications modules has been stopped.
_CnSVBSY	BIT 12		Insufficient common RAM (Insufficient = 1)	● Indicates that service cannot be offered due to insufficient common RAM. ● Indicates communications module hardware defect or system O/S error.
_CnCRDER	BIT 11		Communications module system error (error = 1)	
_NETn_LIV[k] (k = 0 to 63, k:Station No.)	BIT ARRAY	Fnet	Stations connected to the network (1=connected, 0=disconnected)	● Indicates whether k remote station or local PLC is connected to the network or not. The state value is written to each bit. These values shows present state of the network. (Write is disabled)
_NETn_RST[k] (k = 0 to 63, k:Station No.)	BIT ARRAY	Fnet	Re-connection of a station (1=re-connected, 0=no changed condition)	● Indicates re-connected stations, which had been disconnected before, on a bitmap. Because this value has been replaced with '1' when re-connected, the user program has to clear this value with '0' so that next re-connection can be detected. (Write is enabled)
_NETn_232[k] (k = 0 to 63, k:Station No.)	BIT ARRAY	Cnet	The indication that the user defined frame has been received. Indicated at each setting No. (Received = 1).	● When a receiving frame is received through RS-232C while the part of RS-232C in Cnet is operating in the user-defined mode, the bit corresponding to setting No. is turned ON. If RCV_MSG F/B has read that, that bit will be cleared with 0.
_NETn_422[k] (k = 0 to 63, k:Station No.)	BIT ARRAY	Cnet	The indication that the user defined frame has been received. Indicated at each setting No. (Received = 1).	● When a receiving frame is received through RS-422 while the part of RS-232C in Cnet is operating in the user-defined mode, the bit corresponding to setting No. is turned ON. If RCV_MSG F/B has read that, that bit will be cleared with 0.

(1) Communications Module Information Flag List (continued)

Keyword	Type	Applicable Net	Name	Description
_FSMn_reset	BIT	Fnet	Remote I/O station S/W reset	• Requests reset for remote I/O station (Write is enabled) Request can be done individually or wholly complying with the settings in the FSMn_st_no.
_FSMn_io_reset	BIT	Fnet	Remote I/O station digital output reset	• Requests output reset for remote I/O station (Write is enabled) • Request can be done individually or wholly complying with the settings in the FSMn_st_no.
_FSMn_hs_reset	BIT	Fnet	Remote I/O station high speed link information initialization	• If a momentary power failure occurs in the remote I/O station, the operation mode bit of high speed link information turns off and link trouble has the value 1. If the bit is turned on to clear that bit, the operation mode bit turns on and link trouble is cleared with 0. • Request can be done individually or wholly complying with the settings in the FSMn_st_no.
_FSMn_st_no	USINT		Numbers of I/O stations where _FSMn_reset, _FSMn_io_reset and _FSMn_hs_reset will be executed. (Write is enabled)	• Sets the numbers of I/O stations where _FSMn_reset, _FSMn_io_reset and _FSMn_hs_reset will be executed. (Write is enabled) • 00 to 63 ➡ individual station No. setting • 255 ➡ Whole station No. setting

(2) Detailed High Speed Link Information Flag List

Keyword	Type	Applicable Net	Name	Description
_HSmRLINK	Bit	Fnet/Mnet	High speed link RUN link information	• Indicates that all stations are normally operating complying with the parameter set in the high speed link. This flag turns on under the following conditions. 1) All stations set in the parameter are in the RUN mode and have no error, and 2) All blocks set in the parameter normally communicate, and 3) The parameter set in all stations, which are set in the parameter, normally communicate. • Once this flag is turned on, it maintains that state as long as link enable does not make that state stopped.
_HSmLTRBL	Bit	Fnet/Mnet	High speed link trouble information	• This flag turns on when, under the condition that _HSmRLINK is turned on, communications of the stations and data blocks set in the parameter is under the following conditions. 1) A station set in the parameter is not in the RUN mode, or 2) A station set in the parameter has an error, or 3) The communications of data blocks set in the parameter does not normally operate. • This flag turns on if the above conditions 1), 2) and 3) occur. If those conditions are restored, it will turn off again.
_HSmSTATE[k] (k = 0 to 63, k:Station No.)	Bit Array	Fnet/Mnet	K Data Block overall communications state information	• Indicates overall communications state of every blocks of the parameters set. _HSmSTATE[k] = _HSmMOD[k] & _HSmTRX[k] & _HSmERR[k]
_HSmMOD[k] (k = 0 to 63, k:Station No.)	Bit Array	Fnet/Mnet	K Data Block setting stations mode information. (RUN = 1, others =-0)	• Indicates the operation modes of stations set the K data block of parameters.
_HSmTRX[k] (k = 0 to 63, k:Station No.)	Bit Array	Fnet/Mnet	K Data Block communications state information (Normal = 1, abnormal = 0)	• Indicates that communications of the K data block of parameters are normally operating as set or not. .
_HSmERR[k] (k = 0 to 63, k:Station No.)	Bit Array	Fnet/Mnet	K Data Block setting stations state information. (Normal = 1, abnormal = 0)	• Indicates that the stations set in the K data block of parameters have an error or not.

Appendix 3.   Function/Function Block List

1) Function List

Name	Function	Size of PB (byte) *1	Size of library (byte) *2	Processing speed (μsec) *3 GM6
ABS (int)	Absolute value operation	24	—	1.2
ADD(int)	Addition	24	—	1.7
AND (word)	Logical multiplication	16	—	4.3
DIV(int)	Division	32	—	32.9
DIV(dint)	Division	32	—	62.9
EQ (int)	'Equality' comparison	20	—	1.6
LIMIT(int)	To output upper and lower limits	48	794	11.8
MAX(int)	To output the maximum input value	48	738	12.9
MOVE	To copy data	8		1.0
MUL(dint)	Multiplication	24		65.9
MUL (int)	Multiplication	24		35.9
MUX (int)	To output a selected input value	56	682	15.8
MUX(dint)	To output a selected input value	84	682	53.2
ROL	To rotate left	40	160	9.7
BCD_TO_DINT	Conversion of BCD type into DINT type	12	300	273.9
BCD_TO_INT	Conversion of BCD type into INT type	12	200	111.9
BCD_TO_SINT	Conversion of BCD type into SINT type	12	140	40.9
BYTE_TO_SINT	Conversion of BCD type into SINT type	8		0.4
DATE_TO_STRING	Conversion of DATE type into string	48	458	205.9
DINT_TO_INT	Conversion of DINT type into INT type	8		1.3
DINT_TO_BCD	Conversion of DINT type into BCD type	12	278	446.9
DT_TO_DATE	Conversion of DT type into DATE type	16		3.3
DT_TO_TOD	Conversion of DT type into TOD type	16	12	4.1
DT_TO_STRING	Conversion of DT type into string	48	780	524.9
DWORD_TO_WORD	Conversion of DWORD type into WORD type	8		1.3
INT_TO_DINT	Conversion of INT type into DINT type	12		0.9
INT_TO_BCD	Conversion of INT type into BCD type	12	180	129.9
NUM_TO_STRING (int)	Conversion of number into string	52	808	159.9
SINT_TO_BCD	Conversion of SINT type into BCD type	12	140	67.9
STRING_TO_INT	Conversion of string into INT type	16	1308	281.9
CONCAT	To concatenate strings	72	248	54.9
DELETE	To delete string	68	298	63.9
EQ	'Equality' comparison	20	788	38.3
FIND	To find a string	40	222	73.9
INSERT	To insert a string	68	524	418.9
LEFT	To obtain the left part of a string	56	158	33.4
LEN	To obtain the length of a string	16	48	17.5
LIMIT (str)	To output upper or lower limits	80	794	80.9
MAX (str)	To output the maximum input value	76	738	68.4
MID	To obtain the middle part of a string	64	236	47.1
REPLACE	To replace a string with another	73	584	97.9
RIGHT	To obtain the right part of a string	56	226	53.9
ADD_TIME (time)	Time addition	40	280	11.6
DIV_TIME(i1 = time)	Time division	40	266	67.9

REMARK

1) The items marked with '*' has following meaning.
  *1 : The size of the program memory which a program occupies when it uses the function once.
  *2 : The size of the program memory which a program occupies only one time though it uses the function many times.
  *3 : of IL programs (2 input variables, 10 strings)
2) The above shows the function list when programs are written with IL(Instruction List) language. If programs are written with LD(Ladder diagram), the following differences occur.
  (1)   16 byte will be added to the size of the PB.
  (2)   In non-execution, 0.4 will be added to the processing speed. In execution, 0.8 μsec will be added.

2) Function Block List

Name	Function	Size of PB (byte) *1	Size of library		Processing speed (μsec)	
			Size (byte) *2	Size of instance memory *3	GM3	GM4
CTU	Addition counter	72	110	6	10.2	12.8
CTUD	Addition/subtraction counter	112	186	6	15.6	18.4
F_TRIG	Descending edge detection	40	38	1	5.7	6.6
RS	Preference reset table	48	72	2	7.5	8.7
TON	ON delay timer	56	200	2000	8.5	11.1

REMARK

1) The items marked with '✳' has following meaning.
   *1 : The size of the program memory which a program occupies when it uses the function once.
    *2 : The size of the program memory which a program occupies only one time though it uses the function many times.
    *3 :   The size of the program memory which a program occupies whenever it uses the function block once.
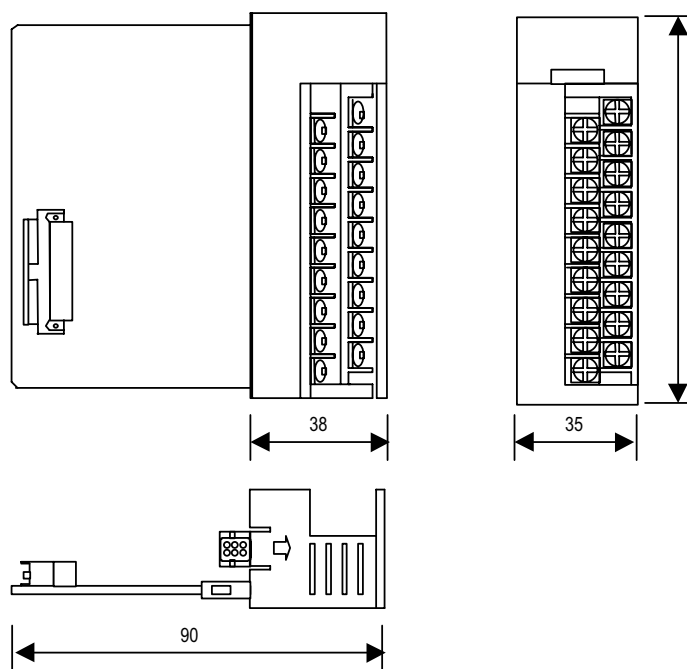2) The occupied memory size and processing speed of IL programs are same as LD programs.
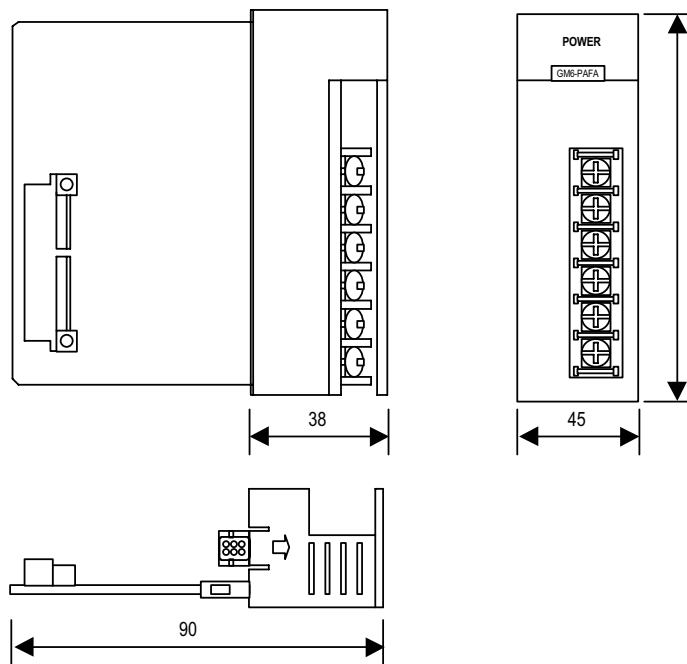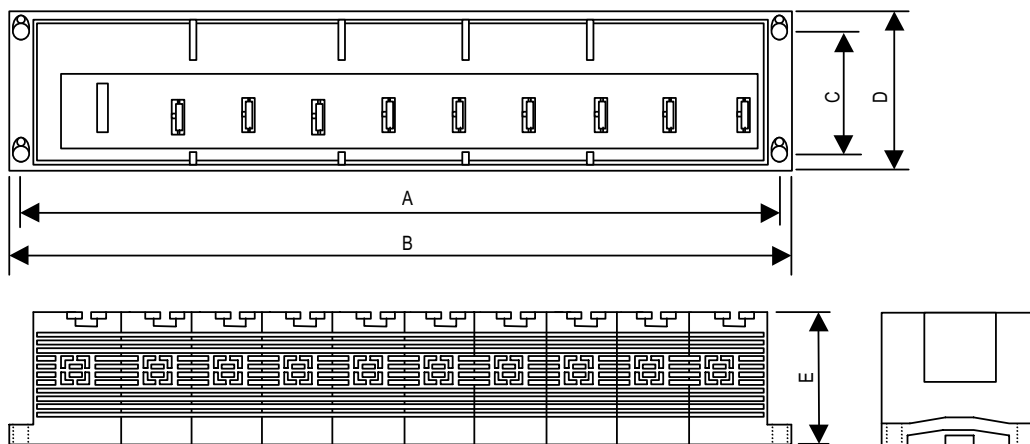
Appendix 4. Dimensions (Unit : mm)

1) CPU module

38

35

90

2)I/O Module

38

35

90
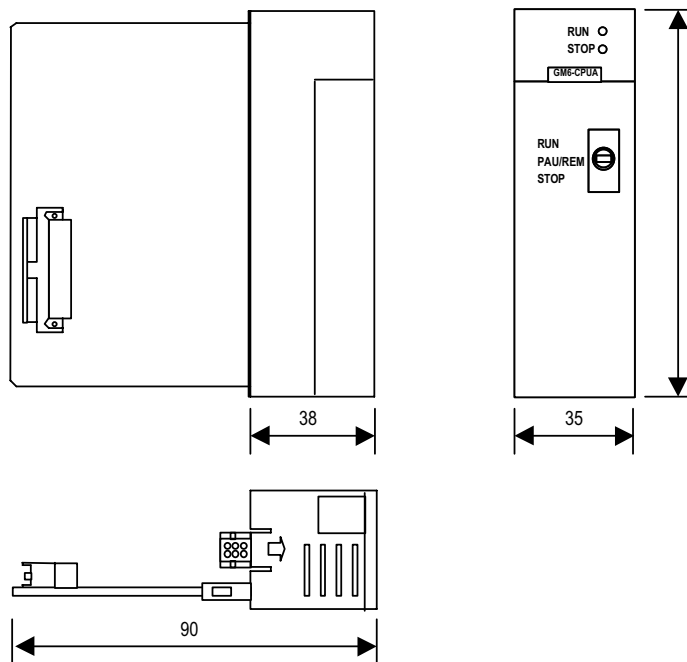
3) Power Supply Module

4) Basic/Extension Base Unit

(Unit : mm)

	A	**B**	**C**	**D**	**E**
GM6-B04M	230.5	244	92.5	110	62
GM6-B06M	300.5	314	92.5	110	62
GM6-B08M	370.5	384	92.5	110	62