Chapter 5. CPU

5.1 Power Supply Specifications

5.1.1 AC power supply

Model Item		G7M–DR10A G7M–DT10A	G7M–DR20A G7M–DT20A	G7M–DR30A G7M–DT30A	G7M–DR40A G7M–DT40A	G7M–DR60A G7M–DT60A
Input	Voltage	AC85 ~ 264V				
	Frequency	50 / 60 Hz (47 ~ 63 Hz)				
	Current	0.5A(AC110V)/0.25A(AC220V)			0.6A(AC110V)/0.3A(AC220V)	
	Inrush current	30 A or less				
	Efficiency	65% or higher (rated input/load)				
	Fuse	1A/Slow/AC250V			2A/Slow/AC250V	
	Dropout tolerance	20 ms or less				
Output (1)	Voltage	DC 5V	DC 5V		DC 5V	
	Current	1.0 A	1.2 A		2.0 A	
Output (2)	Voltage	DC24V				
	Current	0.2 A				
Output status indication		PWR LED On when normal output status				

5.1.2 DC power supply

Model Item		G7M–DR10A/DC	G7M–DR20A/DC	G7M–DR30A/DC	G7M–DR40A/DC	G7M–DR60A/DC
Input	Voltage	DC10.2 ~ 28.8V				
	Current	1.2A(DC12V) / 0.6A(DC24V)			1.8A(DC12V) / 0.6A(DC24V)	
	Inrush current	60 A or less			70 A or less	
	Efficiency	60% or higher (rated input/load)				
	Fuse	5A/Slow/50V				
	Dropout tolerance	1 ms or less				
Output (1)	Voltage	DC 5V	DC 5V		DC 5V	
	Current	1.0 A	1.2 A		2.0 A	
Output status indication		PWR LED On at normal output status				

5.2 CPU Specifications

The following table shows the general specifications of the GLOFA-GM7 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)	Immediate input/output is available by 'direct I/O' function
Programming language		Instruction List Ladder Diagram Sequential Function Chart	
Number of instructions	Operator	LD: 13, IL: 21	
	Basic function	138	
	Basic function block	11	
	Special function block	Each special module have their own special function blocks	
Processing speed	Operator	0.5	
	Basic function	Refer to Appendix 3	
	Basic function block		
Programming memory capacity		68K bytes	Including parameter (Approx. 4k byte)
I/O points		10 points expansion unit : input 6 points/output 4 points 20 points base unit : input 12points/output 8point 30 points base unit : input 18points/output 12point 40 points base unit : input 24points/output 16point 60 points base unit : input 36points/output 24point	Max 2 expansion units can be attached to a base unit
Data memory	Direct variable area	2k to 8k bytes	Adjustable with parameter setting
	Symbolic variable area	32 k bytes-Direct variable area	
Timer		No limitations in points Time range:0. 001 to 4294967.295 sec (1193 hours)	1point occupies 20 bytes of Symbolic variable area
Counter		No Limitations in points Counting range: -32768 to +32767	1point occupies 8bytes of symbolic variable area
Operation modes		RUN, STOP, PAUSE and DEBUG	
Data protection method at power failure		Set to 'Retain' variables at data declaration	
Number of program blocks		128	

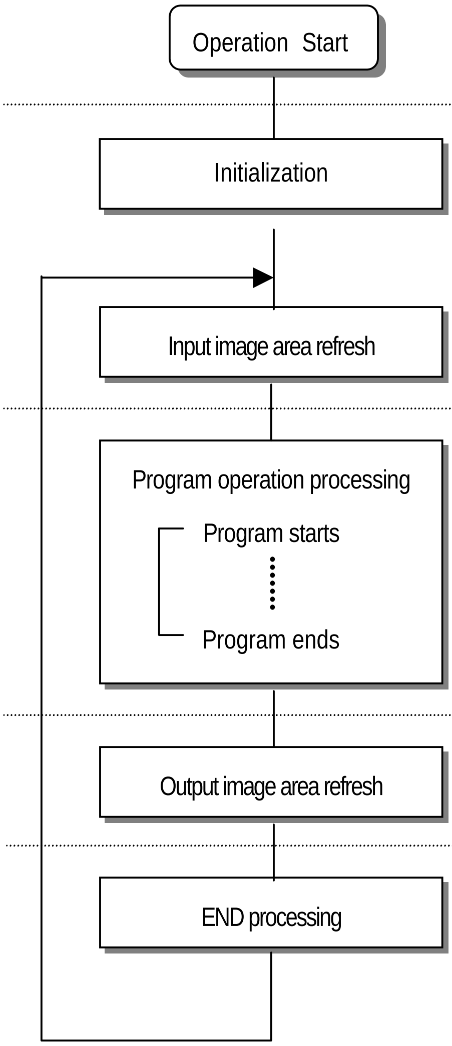
Items			Specifications	Remarks
Program type	Scan		100	
	Time-driven interrupt task		8	
	External interrupt task		8 Total 8 pieces are usable.	
	High speed counter task		1	
	Inside interrupt task		8	
	Initialization task		1 (_INIT)	
Internal Function	PID control function		Function block control, auto tuning, forced output, adjustable operation scan time, forward/reverse operation control	
	Cnet I/F Function		GLOFA exclusive protocol support MODBUS protocol support User's protocol support	Common use with GMWIN port
	High-speed counter	Capacity	1 phase : 16 kHz, 1 channel 2 phase : 8 kHz,1 channel	
		Counter function	It has 3diffferant counter function as following; 1 phase, up/down by program 1 phase, up/down by B phase input 2 phase, up/down by phase difference	
		Multiplication function	Multiplication : 1, 2, or 4 (adjustable)	
		Data comparison function	Execute a task program when the elapsed counter value reaches to the preset value	
	Pulse catch		Minimum pulse width : 0.2msec, 8 points	
	Pulse output		2khz, 1point	Transistor output only
	External interrupt		8points, 0.4ms	
	Input filter		0~15ms	
Weight (g)	G7M-DR20A		480	
	G7M-DR30A		551	
	G7M-DR40A		670	
	G7M-DR60A		844	
	G7E-DR10A		228	

5.3 Operation Processing

5.3.1 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 reset ▶ Execution of self-diagnosis ▶ Data clear ▶ I/O address allocation or type
Input image area refresh	• Input part conditions are read and stored into the input image area before start the processing of a program
Program operation processing Program starts Program ends	• Program is sequentially executed from the first step to the last step Program operation processing
Output image area refresh	• The contents stored in the output image area is output to output part when operation processing of a program is finished.
END processing	• Stage for return processing after the CPU part has finished 1 scan. The END processing following processing is executed. - Self-diagnosis - Change 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 GM7 series, can be set to between 0.001 to 4294967.29 sec. This operation is used to process operation with a constant cycle.

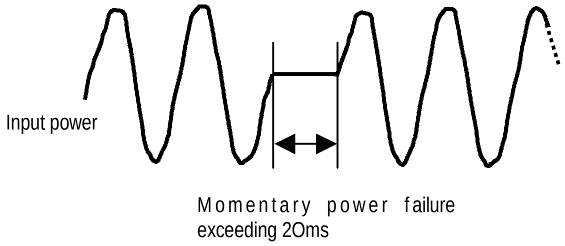
3) Event driven interrupt operation method

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 of those urgent conditions is called interrupt signal. The GM7 CPU has two kind of interrupt operation methods, which are internal and external interrupt signal methods.

5.3.2 Operation processing at momentary power failure occurrence

The CPU detects any momentary power failure when the input line voltage to the power supply falls down below the defined value. When the CPU detects any momentary power failure, the following operations will be executed:

1) Momentary power failure within 20 ms



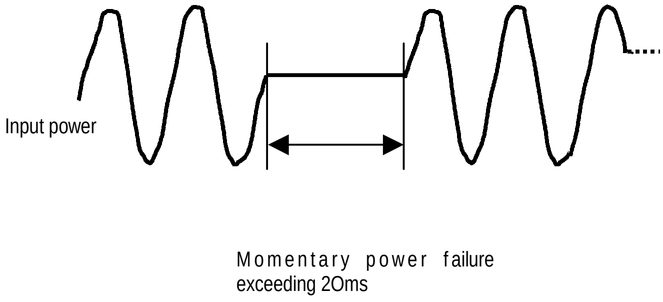
(1) The operation processing is stopped with the output retained.

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

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

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

2) 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).

5.3.3 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) Scan time = Scan program processing time + Task program processing time + PLC internal processing time

• Scan program processing time = The processing time used to process a user program that is not specified to a ta sk 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 + Communi cations service processing time

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

2) Flag

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)

5.3.4 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.

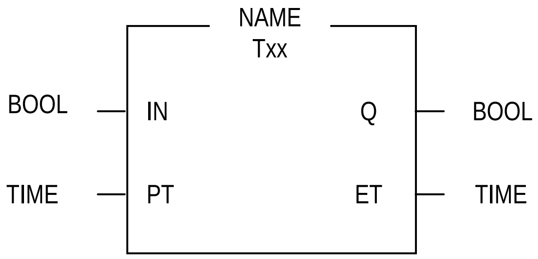
4) In order to clear watchdog error, using manual reset switch, restarting the PLC or mode change to STOP mode are available.

REMARK

Setting range of watchdog : 1~ 5,000ms(unit : 1ms)

5.3.5 Timer Processing

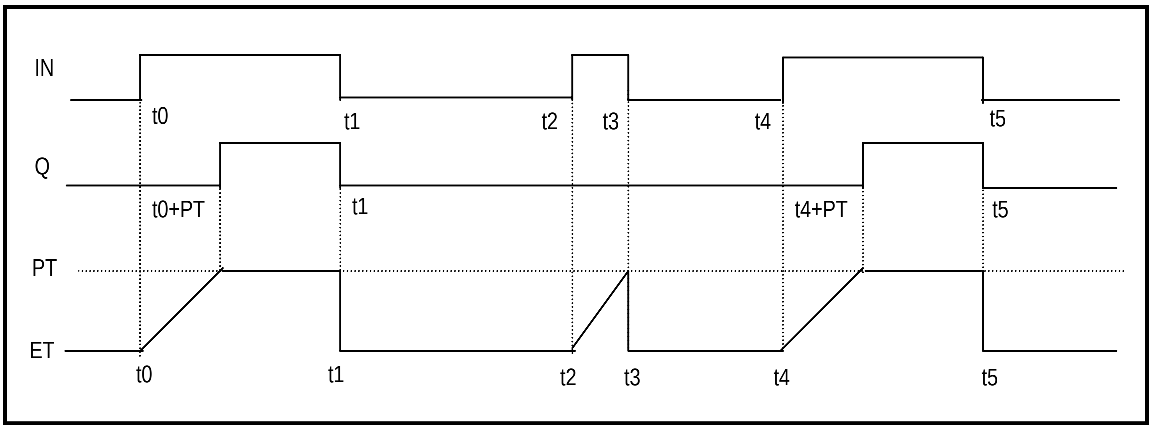
The CPU timer is an incremental timer, which increases its present value according to the measuring time. Three types of On Delay Timer (TON), Off Delay Timer (TOF) and Pulse Timer (TP) are available. Its measuring range is 0.001 to 4,294,967,295 sec (1,193 hours) by 1 ms. for details, refer to “GLOFA-GM programming”.



1) On Delay Timer : Process Time Change and Contact On/Off

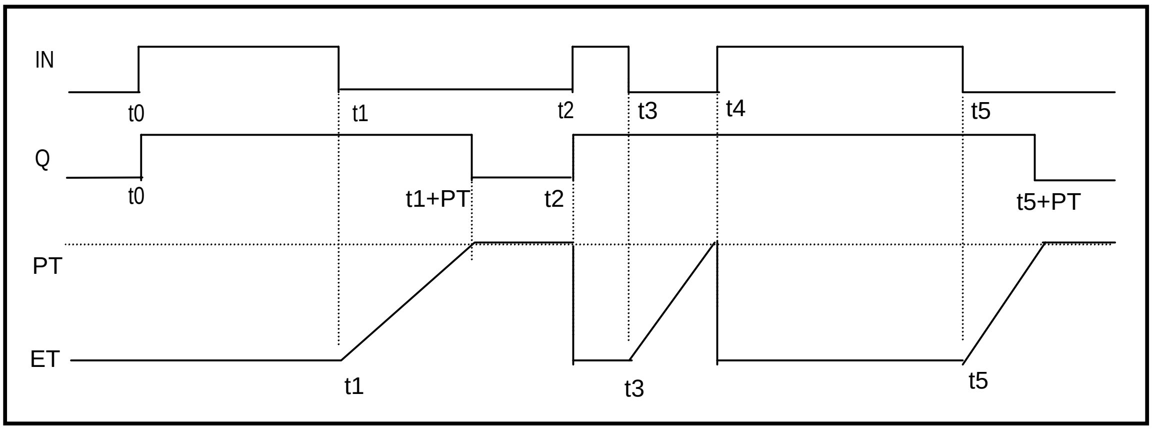
Timer Process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the Timer output contact turns on.

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 starts increasing.
- 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.



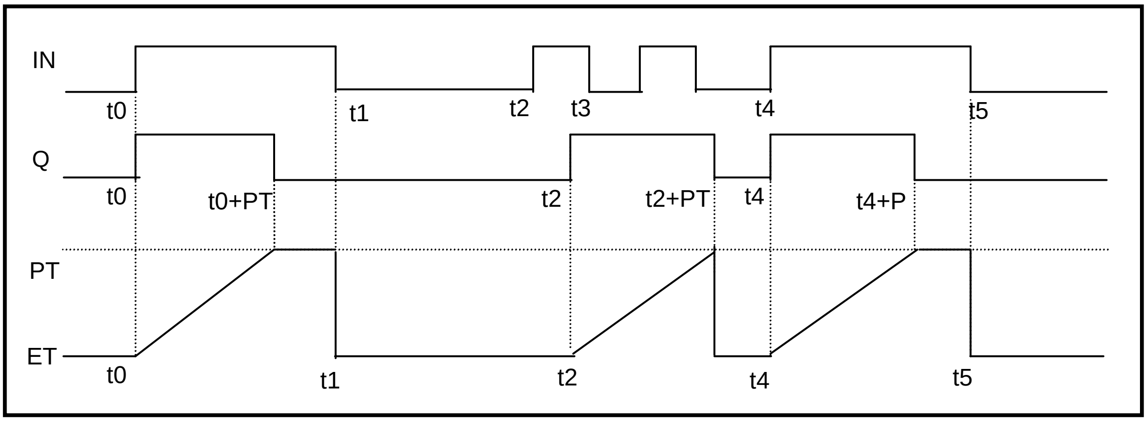
3) Pulse Timer Process Time Change and Contact On/Off

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 reac hes 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'

5.3.6 Counter Processing

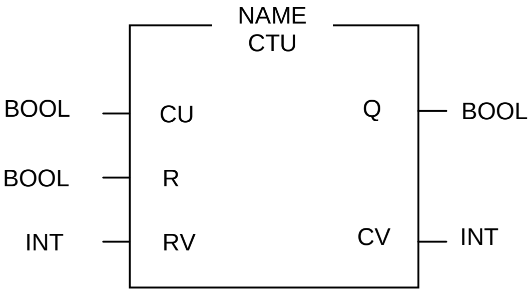
The CPU part counter increase/decrease the present counting value by the detection of rising edge (off→ on) of input signal. Three types of counter are increment counter, Decrement counter and Increment/Decrement Counter. For details, refer to GLOFA — GM Programming'.

- 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

Increment Counter

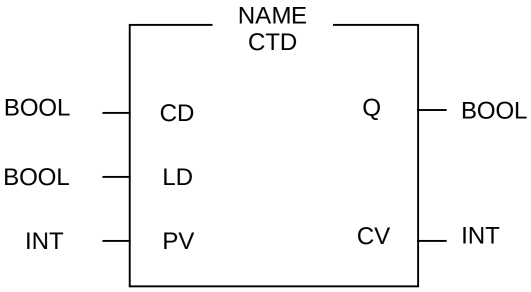
- It should have Input condition (CU), reset condition (R) and setting value (PV).



- 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

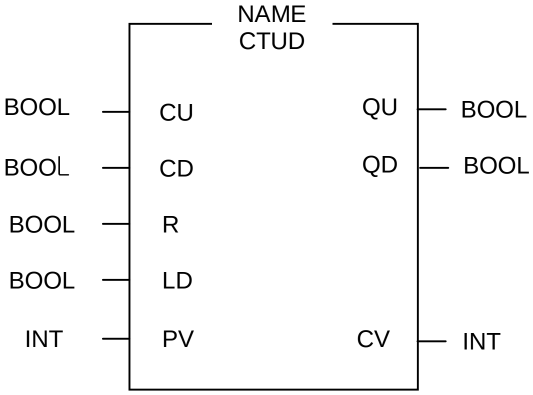
- It should have input condition (CD), load (LD) and setting value (PV).



- 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.

Increment/Decrement Counter

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



- 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 1 at 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

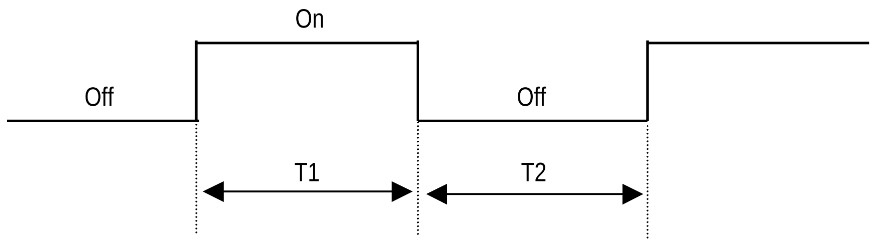
- The counting speed is decided by scan time and it will be counted when on time or off time of input condition is larger than each scan time.

Max. Counting speed Cmax = N/100 * 1/ts (pps/s)

n: duty (%)

ts: scan time(s)

- Duty (n) is the percentage (%) of On/Off of the input signal.



$Ti \leq T2, n= T1/(T1+T2) \times 100$ (%)

$Ti > 12, n= T2/(T1+T2) \times 100$ (%)

5.4 Program

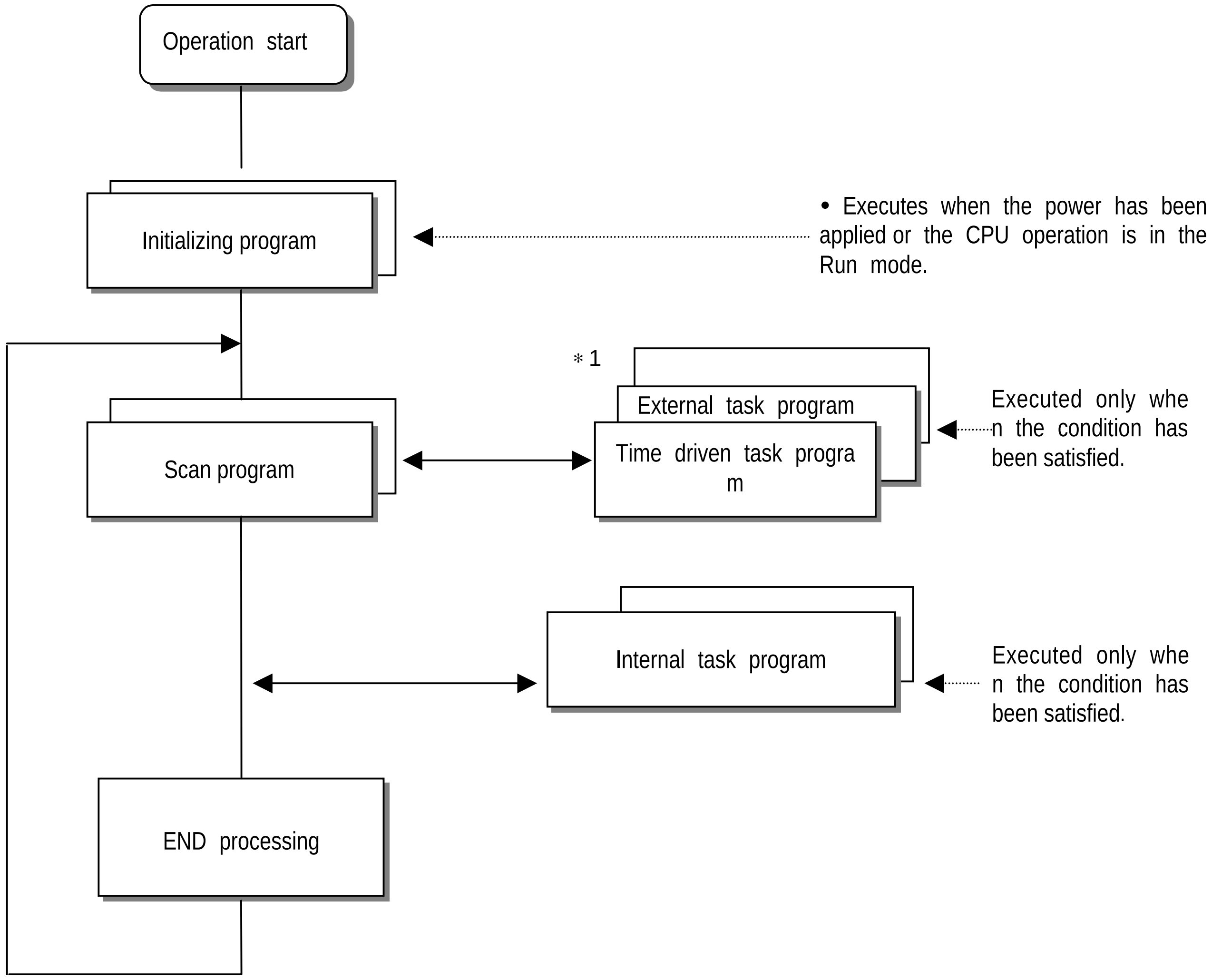
5.4.1 Program Configuration

A program consists of all of the function elements that are needed to execute a particular control. It is to be stored in the internal RAM of the CPU part or the flash memory. 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 of the processing need a shorter interval than that of average scan processing time. ▶ In case of the processing needs a longer interval than that of average 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.

5.4.2 Program Execution Procedure

The followings explain the program execution procedure when the power is applied or the mode-setting switch of CPU part is in the RUN status. Program operation processing is executed as the procedure given below:


Operation start
Initializing program
• Executes when the power has been applied or the CPU operation is in the Run mode.
＊1
External task program
Time driven task progra m
Executed only whe n the condition has been satisfied.
Scan program
Internal task program
Executed only whe n the condition has been satisfied.
END processing

REMARK

＊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.

(2) 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.

(3) Flag

- _INIT_RUN flag is on during executing the initialization program.

2) Scan program

(1) Function

- In order to process signal, 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 during scan program execution, the program that is under execution will be temporary stopped and the corresponding task program will be executed.

(3) 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-periodicity the task program temporarily stop the operation of scan program and processes first the corresponding function.

(2) Types

- Task programs are classified into the four 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
  - ▶ High speed counter task program: only 1 program is 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 proces sed.
- Interrupt (external) task program
  - ▶ The program is executed according to the external signal a input to the interrupt module
- High-speed counter task program
  - ▶ The program is executed according to speed level.

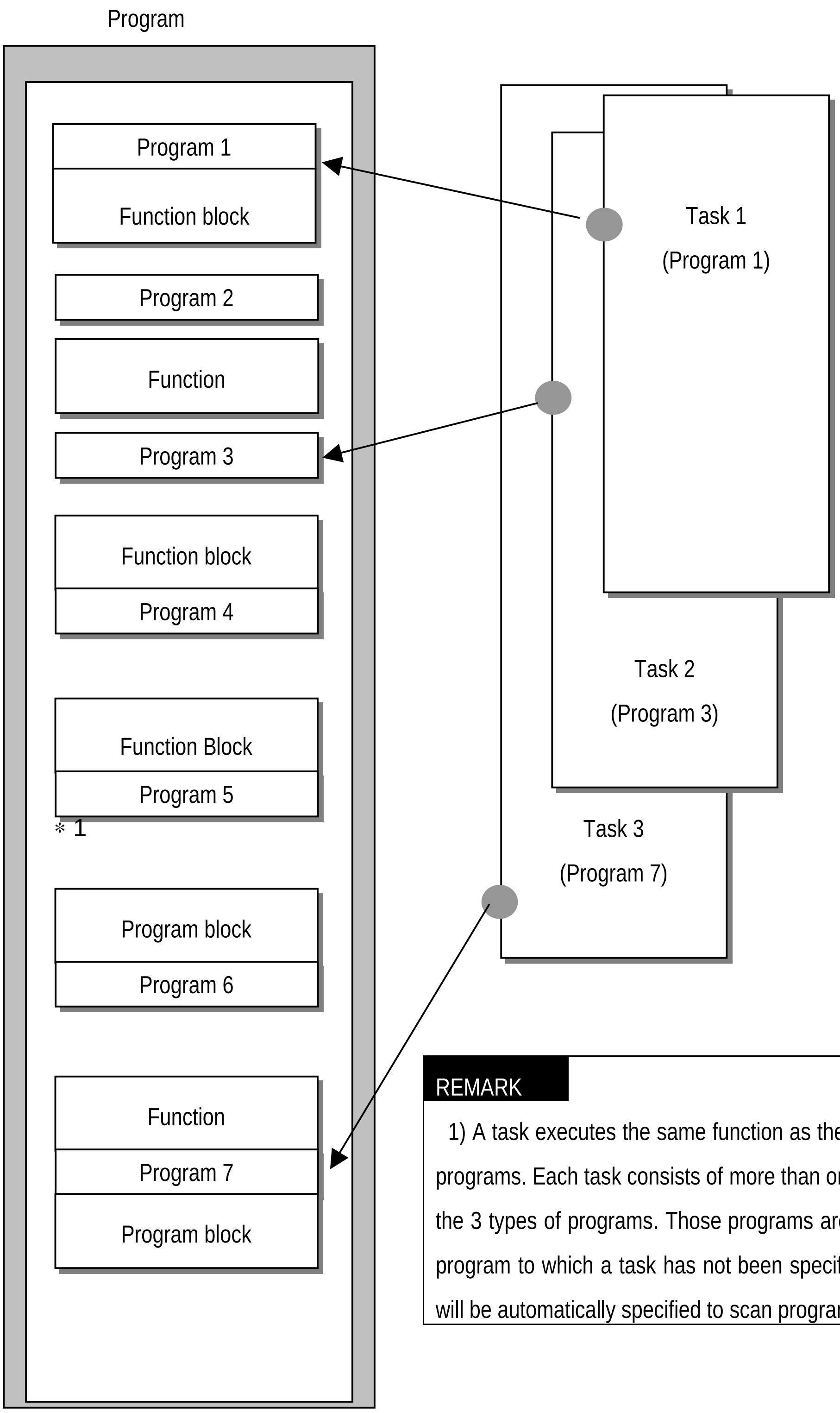
Remark

1) Refer to section 5.3.3 “Task” for details of task program.

2) GM7 series uses separate input program to manage sign of interrupt. Refer to section 5.3.3 “Task” for details of task program.

5.4.3 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. (Refer to GIMWIN section for details of GMWIN program)



REMARK

1) A task executes the same function as the control panel executing programs. Each task consists of more than one program blocks out of the 3 types of programs. Those programs are called task program. 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 shows the types and functions of tasks.

Type Size	Time driven task	External interrupt task	Internal interrupt task	High speed task
Number	8	8	8	1
Start up condition	Time driven interrupt (up to 4,294,967.29 sec by the 10 ms)	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.	Using CHSC_SET F/B, select a set value.
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.	When reaches the SV, it executes.
Detection delay time	Up to 1 ms delay	0.4ms	Delayed for the same time as max. scan time.	Delayed up to 1 ms.
Execution priority	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)	Level 0 to 7 (Level 0 has highest priority)

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 an execution condition is satisfied while the scan program is r epeatedly 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, th e 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 a re 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

REMARK

The priority for GM7 can’t be set as the same. If it is set as the same, an error will occur.

(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 white 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)
- Use 'DI' function where program needs protection and 'EI' function where program needs cancellation. After the scan program ends of the running program, automatically it becomes permissible. Initialization program doesn't get influences from 'DI and EI.'

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 drive 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 (TASKERR) will be set to ON, the detailed system error flag (JC 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 wh

en 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 20 sec are used, four execution requests will occur every 20 sec and scan time can be momentarily extended.

4) External contact program processing method

In GM7series, it is different from GM1/2/3/4 to use normal digital input task program, not a separate interrupt input module. 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 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) External contact task processing

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

(3) Precautions for using an external contact task.

- Input interrupt that is possible to set is up to %IX0.0.0~%IX0.0.7.
- 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], TC_CNT[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(l, 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 ris ing 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 pro gram(time driven, external) the task (start-up condition) will not be immediately executed but will be exec uted when scans program has finished its execution.
- If execution of an internal task program is requested, the execution conditions will be checked when sca n program has finished its execution. Therefore, if an internal task execution conditions, during 'One' sca n, 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.

REMARK

1) When an action must continuously be executed according to the related contact point set as a start-up condition, select a level.

6) Execution of high-speed task program

GM7 series uses general digital input contact point to count high-speed pulse, not a separate high-speed pulse input module. Setting a task (startup condition) as the same with the one of the high-speed pulse input will be explained.

(1) Conditions to be set for a task

- Set the priority on the tasks that are startup conditions for the task program to be executed. Then a task number will automatically be added in the priority order.

(2) Processing the high speed counter task

- When CHSC_SET F/B of the program assigns a set value, the task program whose set value matches with the counted value of the pulse that is input in a high speed is executed.

(3) Precautions for using high speed counter task program

- The task can be used only with CHSC_SET F/B.
- High speed input counter can be used without CHSC_SET F/B.
- Even though the operation is PAUSE mode, counted value rises. But this can executes the task program. When

the operation is RUN mode, the task is executed.

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. If the task setting cannot be changed, check the maximum scans 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.

8) Example of program configuration and processing

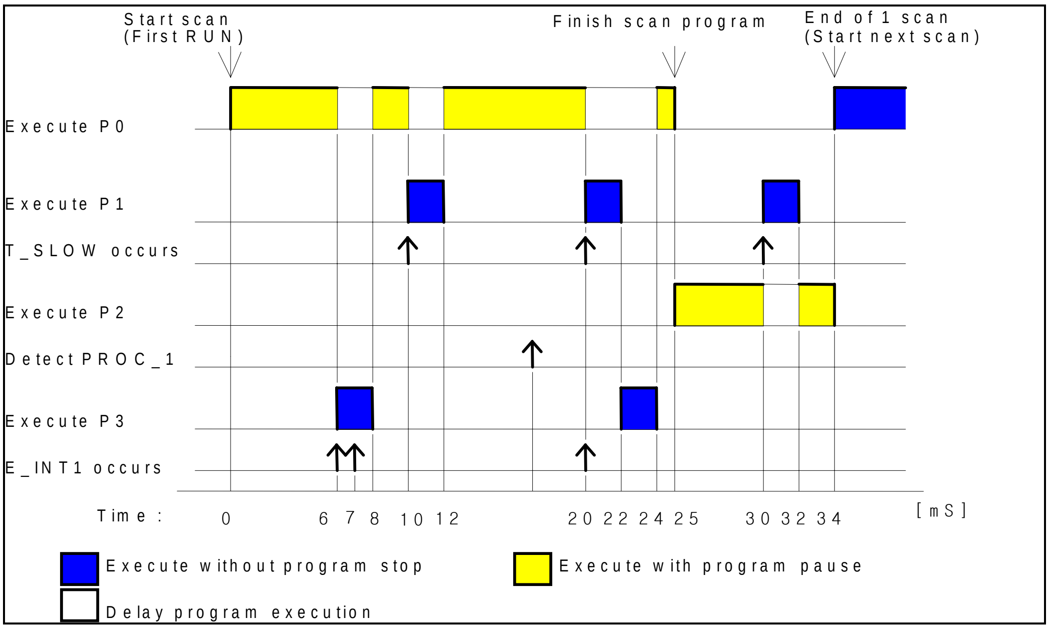
When the task and program have been registered as below,

•	Task registration :	T_SLOW (interval T#10ms, priority = 0)
		PROC.9 (internal contact point: %MX0, priority = 3)
		E_INT1 (external contact point: %IX0.0.1, priority = 2)
•	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 EINT occurrence time: Occurred at the 6, 7, 20 ms after the operation started.
- PROC_1 occurrence: Invoked during the execution of scan program

Program execution is shown as below.



Processing with time

Time(ms)	Processing
0	Scan starts and the scan program P0 starts its execution.
0~6	The program P0 is being executed.
6~8	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.
8~10	P3 finishes its execution and the P0 stopped continues its execution.
10~12	P0 is stopped and P1 is executed due to execution request for P1.
12~20	P2 finishes its execution and the P0 stopped continues its execution.
20	Execution requests for P1 and P3 are simultaneously exist, but the higher priority P1 is executed and P3 is ready for its execution.
20~22	P0 is stopped and P1 is executed.
22~24	P1 finishes its execution and the higher priority P3 is executed before P0.
24~25	P3 finishes its execution and the P0 stopped completes its execution.
25	Execution request for P2 is checked at the finish time of the scan program (P0) and P2 is executed.
25~30	The program P2 is executed.
30~32	Execution request for P1 is input and P2 is stopped and P1 finishes its execution.
32~34	P1 finishes its execution and the P2 stopped finishes its execution.
34	A new scan starts. (P0 starts its execution.)

5.4.4 Error Handling

1) Error Classification

Errors occur due to various causes such as PLC system defect, system configuration fault or abnormal ope ration result. Errors are classified into fatal error mode, which stops system operation for system stability, a nd 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 GM7 series. The system enters into the STOP state.

(3) Operation error during execution of the user programs

It the numeric operation error of these errors occurs during execution of the user programs, its contents ar e marked on the error flags and the system continues its operation. If operation time overruns the watchdo g 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 ente rs 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.

5.4.5 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 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.

2) Control of special modules

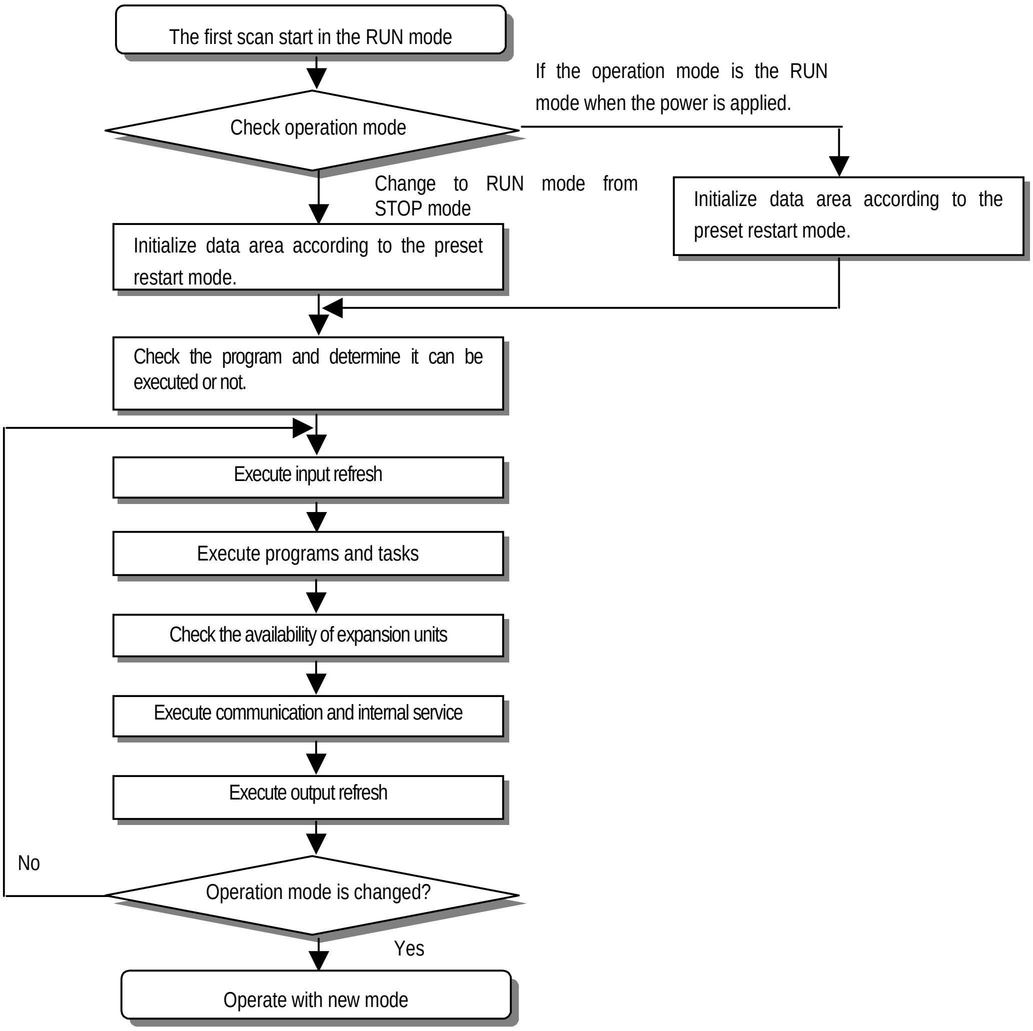
In controlling the operations of special modules, write the program using function blocks, which correspond to the operations that have to be controlled. These function blocks can locate at any place within the program.

5.5 Operation Modes

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

5.5.1 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 when the power is applied.
Check operation mode
Change to RUN mode from STOP mode
Initialize data area according to the preset restart mode.
Initialize data area according to the preset restart mode.
Check the program and determine it can be executed or not.
Execute input refresh
Execute programs and tasks
Check the availability of expansion units
Execute communication and internal service
Execute output refresh
No
Operation mode is changed?
Yes
Operate with new mode

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 initializati on 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 refreshes 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.

5.5.2 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.

5.5.3 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 and input image are not cleared 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.

5.5.4 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 sta te 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 (step over)	If an operation command is ordered, the system operates one operation unit 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. .• U to 8 break points 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.

5.5.5 Operation mode change

1) Operation mode change methods

The following method is 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
PAU / REM → STOP	Local STOP

REMARK

1) ∗ 1: If the operation mode changes from RUN mode to local RUN mode by the mode setting switch, the PLC operates continuously without stopping.

2) * 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	0	0
	Remote STOP → Remote PAUSE	X	X
	Remote STOP → DEBUG	0	0
	Remote RUN → Remote PAUSE	0	0
	Remote RUN → Remote STOP	0	0
	Remote RUN → DEBUG	X	X
	Remote PAUSE → Remote RUN	0	0
	Remote PAUSE → Remote STOP	0	0
	Remote PAUSE → Remote DEBUG	X	X
	DEBUG → Remote STOP	0	0
	DEBUG → Remote RUN	X	X
	DEBUG → Remote PAUSE	X	X

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)

5.6 Functions

5.6.1 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 as '0' and only variables of which initial value has been defined will be set as 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

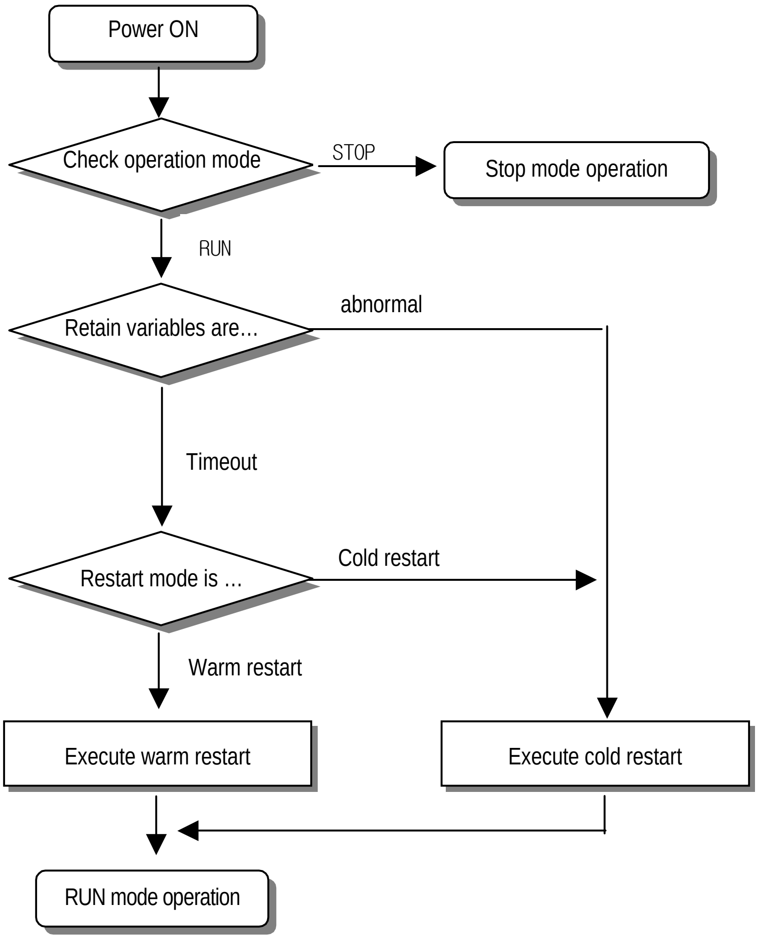
(1) It is executed when the restart mode parameter has been set to the warm restart mode.

(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 va riable 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 & Initializati on	Initialized with the user defined v alue	Previous value is retained.

REMARK

Definitions of variable

(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.

5.6.2 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 i s 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 11.5 'Error Code List of Chapter 11'. Troubleshooting for details of contents of self-diagnosis and corrective actions.

5.6.3 Remote function

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

1) Remote RUNISTOP

(1) The remote RUN/STOP permits external operations to RUN/STOP the CPU module under the condition that t he mode selling 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 m ode-setting switch of CPU module is in the remote position. The PAUSE operations stop the CPU module operatio n 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 s etting switch of CPU module is in the remote position. The DEBUG operations execute programs complying with t he specified operation conditions.

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

5.6.4 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, changes 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.

5.6.5 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.

5.6.6 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

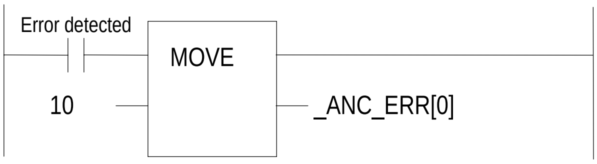
(1) If an error of external device is detected and the error type, where other value than 0 is used, is written to the system flag ANC_ERR[n], the flag will checked at the time that scan program finishes its execution. If an error is indicated on the flag, it will be also indicated on the _ANNUN_ER of 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)



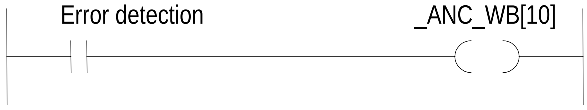
3) External device Ordinary-fault (Warning) Processing

(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)



_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 _ANCWB[10] to ON, the states of _ANNUNWR 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] = 1
_ANC_WAR[2] = 2
_ANC_WAR[3] = 3
_ANC_WAR[4] = 15
_ANC_WAR[5] = 40
_ANC_WAR[6] = 50
_ANC_WAR[7] = 60

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 sho

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 _ANCWAR[n]. The _ANC_WB[75] is n ot indicated as it is turned on and the warning that occurred before ha s written to the _ANCWARIn1.

_ANNUN_WR = 1
_ANC_WAR[0] = 1
_ANC_WAR[1] = 2
_ANC_WAR[2] = 3
_ANC_WAR[3] = 15
_ANC_WAR[4] = 40
_ANC_WAR[5] = 50
_ANC_WAR[6] = 60
_ANC_WAR[7] = 75

After the next scan has been finished, if the numbers 1, 2, 3,10,15 4 0 , 50, 60 and 75 of _ANC_WB[n] are tuned on _ANC_WAR[n] will b

The No. 10 warning has been released the content of _ANCWAR[0] wi ll be cleared and the contents of _ANC_WAR[1..7] will shift into the lo wer elements. The content of _AN&WAR[7] will has been cleared by th e shifting and the content of _AN&WB[751 will be written to ANCWAR [7].

_ANNUN_WR = 0
_ANC_WAR[0] = 0
_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 operati on, the ANNUN_WR and _ANC_WAR[n] will be shown as left.

5.7 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 sto res 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 byte
Parameter area • Basic parameter area • High speed link parameter area • interrupt setting information area	2.0 Kbytes
Program area • Scan program area • Task program area • User defined function/function block area • Standard library area • Variable initialization information area • Protective variable specification information area	66 Kbytes

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 Kbytes
System area • I/O information table • Force I/O table	1 Kbytes
System flag area	1.5 Kbytes
Input image area (%IX)	128 byte
Output image area %QX	128 byte
Direct variable area %M	2 ~ 8 Kbytes
Symbolic variable area (maximum)	29 Kbytes – the size of direct va riable area

3) Purpose

(1) System area

It used to save the self-producing data of the CPU module for the system management and GMWIN system control data.

(2) System flag area

It used to save the user flags and system flags. The user operates it by flag names.

(3) Input image area

It used to save input data read from input modules. Overall size is %IX0.0.0~%IX0.7.63. Only %QX0.0.0~%QX0.3.63 can be used as a real input domain but the other unused domain can be used as convenience, especially remote output data for communication can be saved here as convenience.

(4) Output image area

It used to save operation results that are automatically output through the output device. Overall data size is %QX0.9.9~%QX0.7.63. In GM7, only %QX0.0.0~%QX0.3.63 can be used as a real input domain but the other unused domain can be used as convenience, especially remote output data for communication can be saved here as convenience.

(5) Direct variance area

The user can use this area to access direct memory data, using the variable names such as %MX0, %MB0, %MW0, %MD0, which was defined in advance by the system. Memory size is defined when the user makes program. Refers to “system definitions” for the variable area available to use according to the setting.

(6) Symbolic variable area

It used to save the variables that when the user creates a program or when the user defines a global variables, is automatically allocated its memory. The variables used in program blocks are located in the ‘PB instance memory’ of the related program, and the memory used in the function block is located in the ‘FB instance memory.’

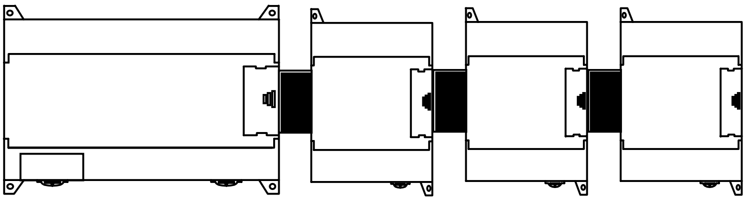
5.8 I/O No. Allocation Method

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.

The following shows an example of I/O No. allocation method.

	Base unit(20~60 points)	Expansion module(10 points)	Expansion module(AD mix)	Expansion module(10 points)
Input	%IX0.0.0 ~ %IX0.0.35	%IX0.1.0 ~ %IX0.1.5	-	%IX0.3.0 ~ %IX0.3.5
Output	%QX0.0.0 ~ %QX0.0.23	%QX0.1.0 ~ %QX0.1.3	-	%QX0.3.0 ~%QX0.3.3

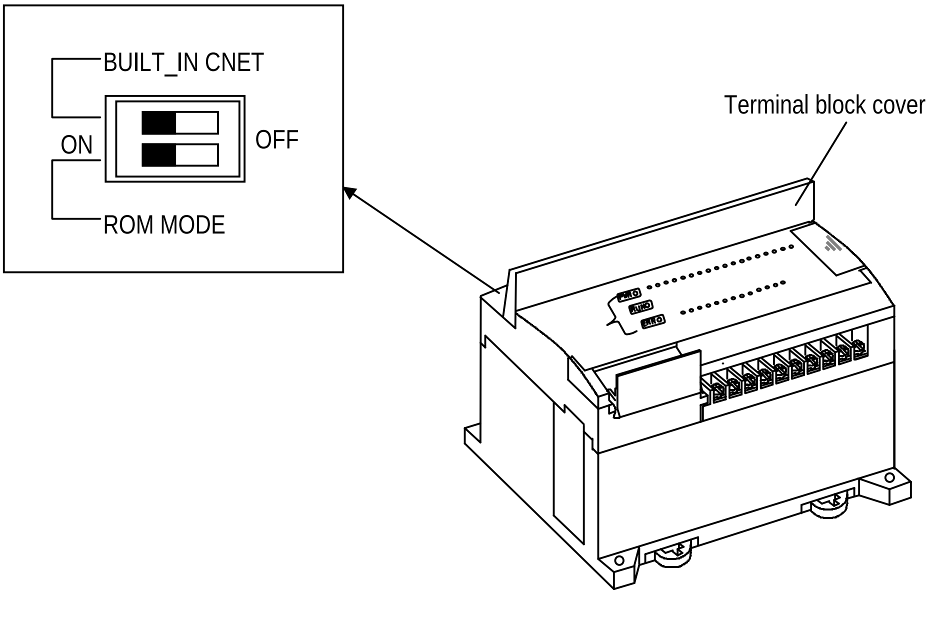


5.9 Built-in Flash Memory

GM7 series includes a built-in flash memory to store user program. Also, user can set the PLC automatically executes the user program of flash memory when the PLC is turned on. It is similar with the ROM operation of other PLCs, but it is different that no external memory is required.

5.9.1 Structure

You can see dip switches as shown when you open I/O terminal block cover.

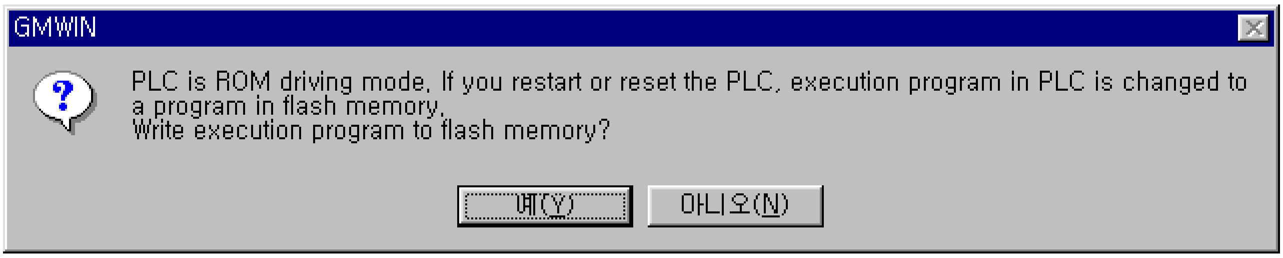


5.9.2 Usage

1) Set the base unit to the STOP mode.

2) There are 2 ways to use writing program into the built-in flash memory of base unit.

(1) While ROM_MODE switch is On, if the program is written on the basic unit, the following message at GMWIN shows.



PLC is ROM driving mode. If you restart or reset the PLC, execution program in PLC is changed to a program in flash memory.
Write execution program to flash memory?

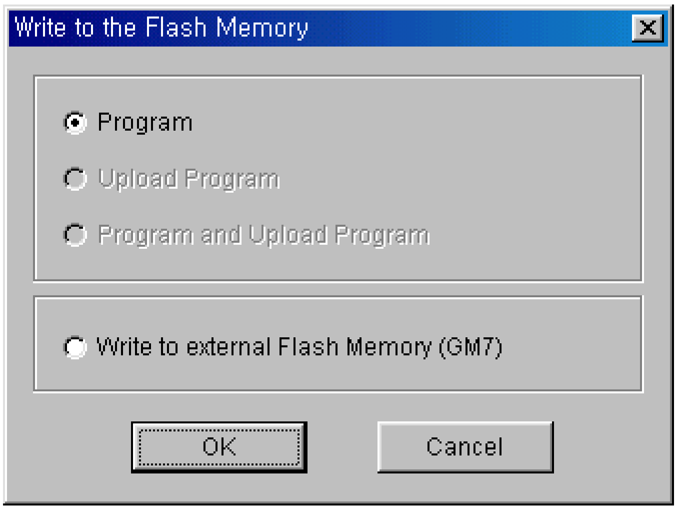
- ◆ If you want to input downloaded program into flash memory, press 'y.'

REMARK

Above message shows only when ***Online – Writing – Parameter and program*** is chosen in menu.

(2) To save in the flash memory, lest not write on the program, it's as follows:

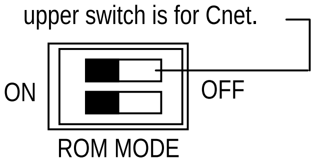
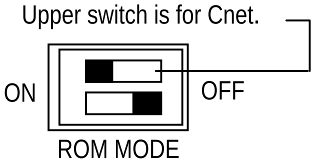
- ◆ If ***Online – Flash memory – Writing on flash memory*** is chosen, the following window shows.



● Program
○ Upload Program
○ Program and Upload Program
○ Write to external Flash Memory (GM7)

- ◆ Choose title to be saved in the flash memory and press 'y.'

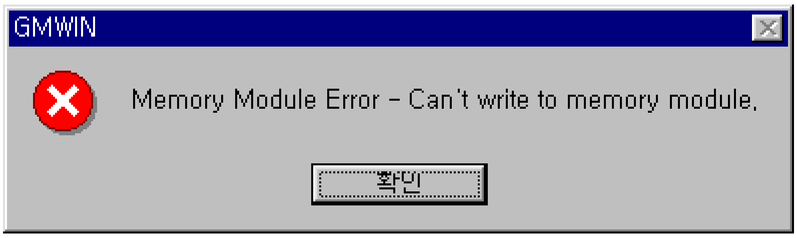
3) When you reset the PLC system, it works according to the dip switch for operating flash memory.

Dip switch position	Description
 	When power is on, the program saved in the flash memory operates.
 	CPU recognizes that there is no program in the flash memory, and starts to drive program from RAM.

REMARKS

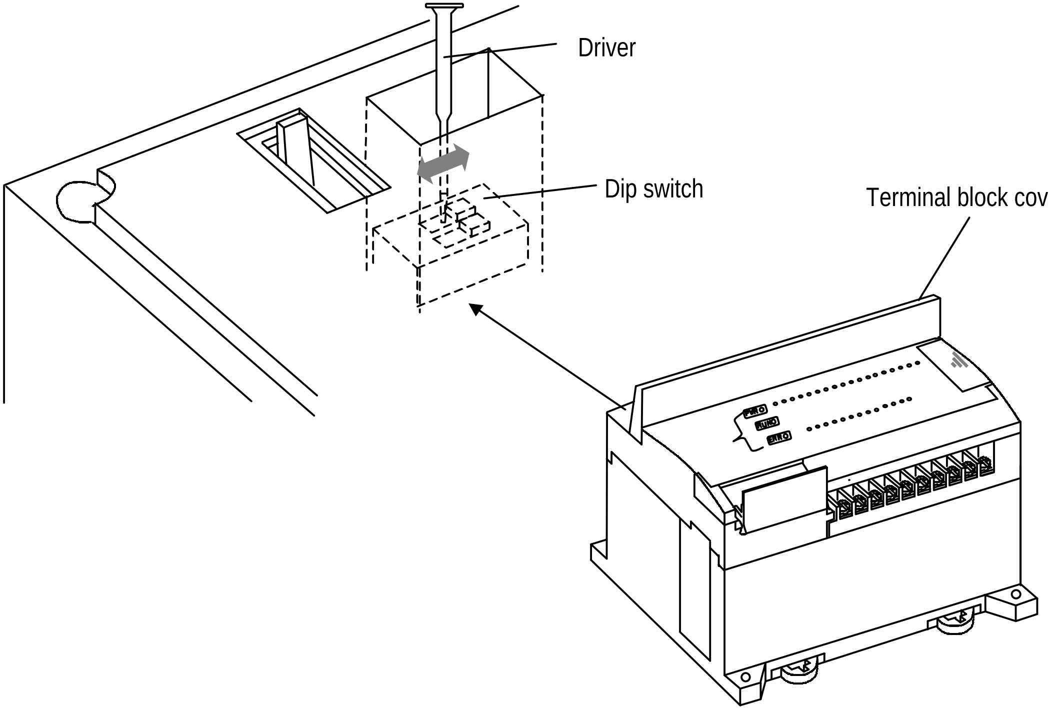
1) The flash memory dipswitch has no relationship with saving the program.

When you save the program in the flash memory, set STOP mode as the operation mode of the basic units. If you try to save when the mode is RUN, the following window shows.


GMWIN

Memory Module Error - Can't write to memory module.

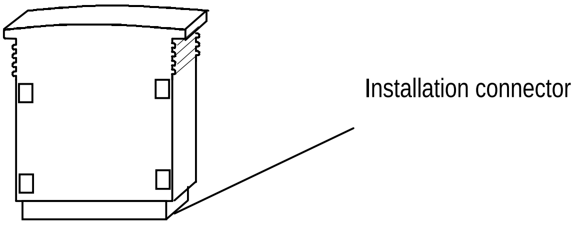
3) Dip switch for flash memory operation is placed in deep place to prevent a mistaken operation caused by terminal block cover, etc. Use a small driver to operate it.



5.10 External Memory Module

GM7 series supplies external memory module for the user to save programs safely or download a program on the system and use it in case of a program is damaged.

5.10.1 Structure



5.10.2 Usage

1) Saving the user's program on the external memory module.

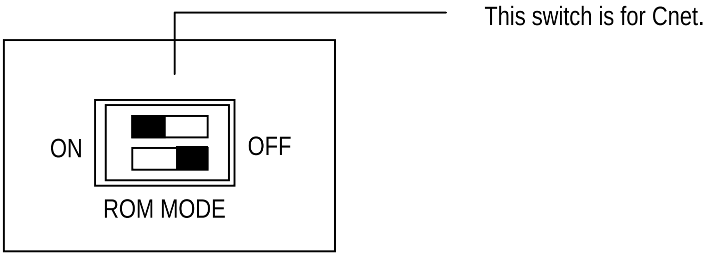
Turn the power of the base unit off.

Install the memory module.

◆ When only basic unit is used: Connect to the expansion connector of the basic unit.

◆ When expansion unit is used: Connect to the expansion connector of the last connected expansion unit.

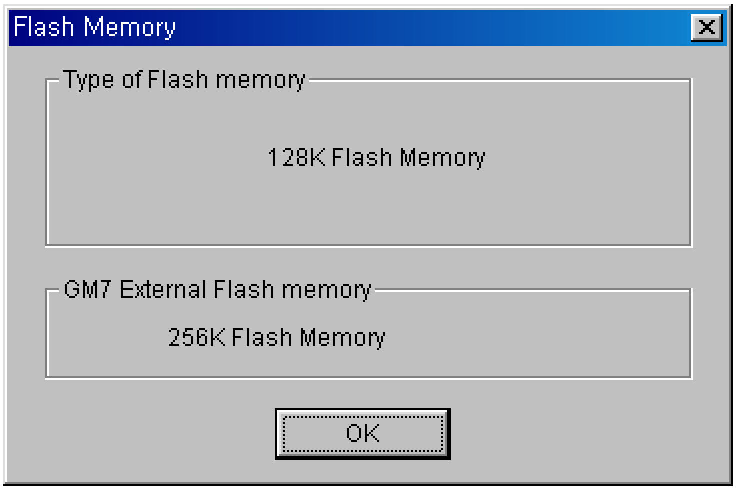
Turn the dip switch for ROM mode setting of the base unit to OFF.



(4) Turn the power of the base unit on.

(5) Connect GMWIN and PLC.

(6) Select ***Online – Flash memory – Type information*** to confirm the flash memory size and installation of the memory module.



Flash Memory

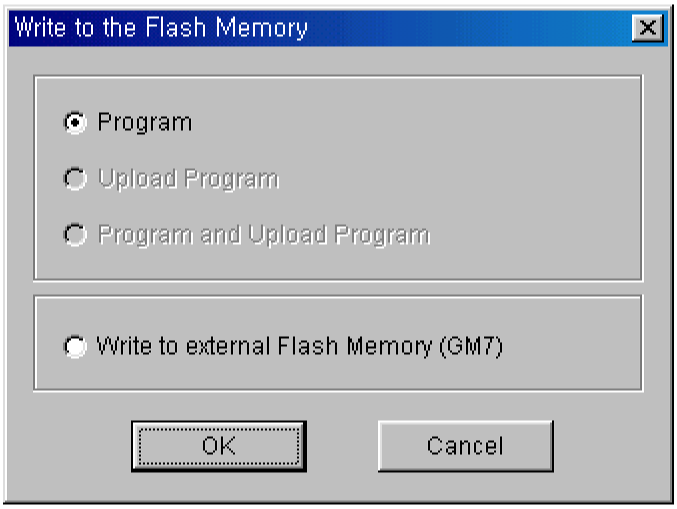
Type of Flash memory

128K Flash Memory

GM7 External Flash memory

256K Flash Memory

(7) Choose ***Online – Flash memory – Writing*** in menu, and the following message box will displayed.



Write to the Flash Memory

● Program

○ Upload Program

○ Program and Upload Program

○ Write to external Flash Memory (GM7)

(8) Choose an item to be saved in the flash memory and press ‘OK.’

(9) Turn the power of the base unit off.

(10) Remove the external memory module.

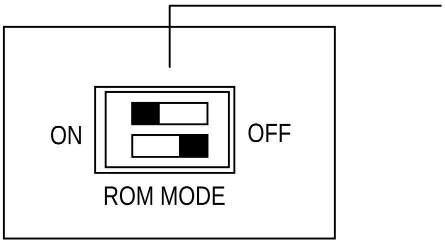
Through the above steps a user can save a program into the external memory module.

2) Run the PLC with a program of external memory module

(1) Turn the power of the base unit off.

(2) Install the memory module (When only base unit is used, connect to the expansion connector of the base unit. And when expansion unit is used, connect to the expansion connector of the last connected expansion unit).

(3) Set the dip switch for ROM mode setting of the base unit to OFF position.



This switch is for Cnet.

(4) Turn on the power of the base unit.

(5) As RUN LED and ERR. LED are on, the contents of the memory module is transferred into the program area of the base unit and ROM operation area of the flash memory. (It may take about 15 sec.)

(6) Operate according to the set operation mode.

(7) Turn off the power of the basic unit.

(8) Remove the memory module.

(9) Turn the power on.

Through the above steps the user can operate the PLC with program stored in the external memory module.

REMARK

1) When the PLC is operated with the external memory module, it always operates with cold restart.

2) Remove after the program transfer is finished.

5.11 Battery

1) Specifications

Item	Specifications
Normal voltage	DC 3.0 V
Warranty life time	5 years
Application	Programs and data backup, and RTC runs in power failure
Specifications	Lithium Battery, 3V
External dimension (mm)	ϕ 14.5 X 26

2) Handling Instructions

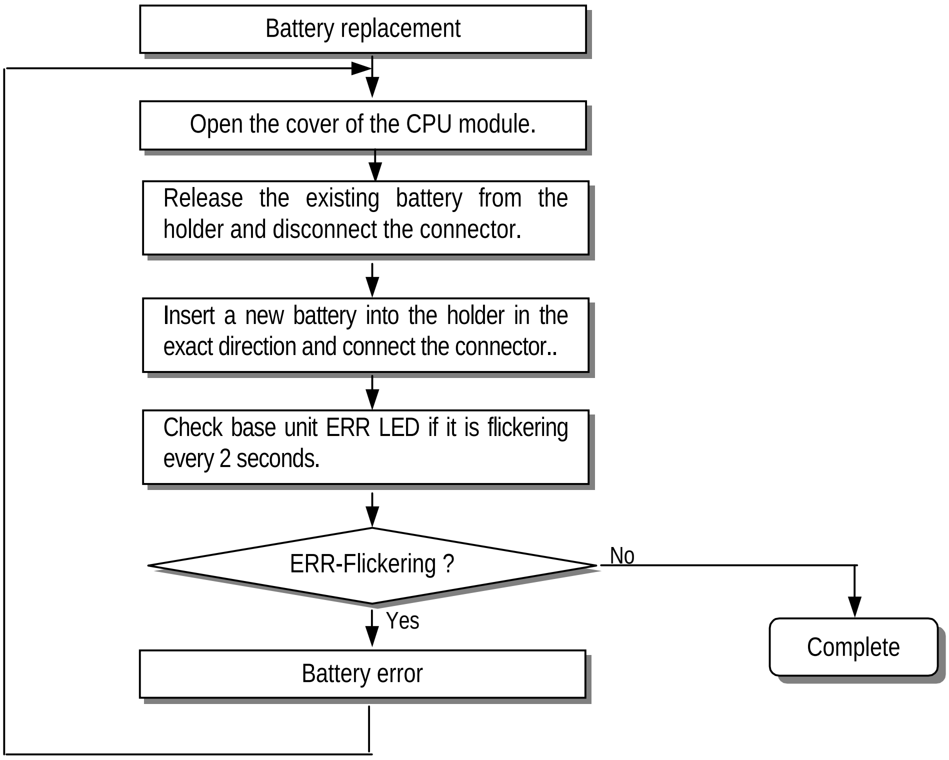
(1) Don’t heat or solder its terminals.

(2) Don’t measure its voltage with a tester or short circuit.

(3) Don’t disassemble.

3) Battery Replacement

Backup battery needs periodic exchange. In case of battery replacement at power off, the built-in super capacitor backup the program and retain variables about 30 minutes. However, it is recommended to complete the battery replacement as soon as possible, or turn on the base unit during battery replacement.


Battery replacement
Open the cover of the CPU module.
Release the existing battery from the holder and disconnect the connector.
Insert a new battery into the holder in the exact direction and connect the connector..
Check base unit ERR LED if it is flickering every 2 seconds.
ERR-Flickering ?
No
Yes
Complete
Battery error

5.12 RTC module

The G7E-RTCA module GM7 provides RTC(Real Time Clock) function for GM7 series. The RTC function can be used for time-scheduling control or recording an error occurrence time. The RTC data is updated into system operation status flag per every scan.

1)Functions

The RTC module will send the RTC data to main unit per every scan.
By the super capacitor back-up, the RTC module keep operating while the power is off.

2)RTC data

Item	Data
Year	Upper 2 digit of year data
	Lower 2 digit of year data
Month	1 ~ 12
Date	1 ~ 31
Hour	0 ~ 23 (24 hour)
Minute	0 ~ 59
Second	0 ~ 59
Day	0 ~ 6 (Monday:0 ~ Sunday:6)
Century	Indicate upper 2 digit of year data

(1)Accuracy

Max. ± 2.2 sec / 1 day (At 25 °C)

(2)RTC data back-up time

200 hours (At 25 °C)

(3)Read/Write of RTC data

Select the menu 'Online – PLC Information' win GNWIN software.

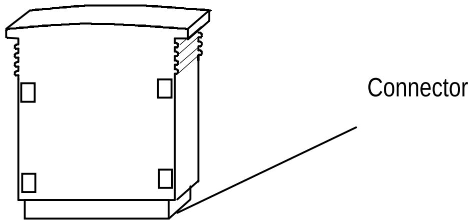
NOTE

1) The RTC module is sold with no initial RTC data setting. Be sure to input the RTC data when use a RTC module first time.

2) The RTC module may show abnormal operation when a improper RTC data is written.
Example) 14(Month) 32(Date) 25(Hour)
When the super capacitor is fully discharged, the RTC module may stop operation or RTC data may be broken. To remove error status, re-write RTC data by GMWIN software.
The system flag _RTC_ERR of _CNF_WAR will turn on when a RTC data error occurred. This flag will turn off automatically when the error is cleared.

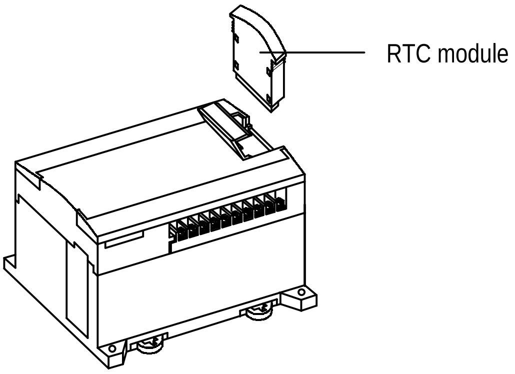
3) Shape of module and connection method

(1) Shape of module

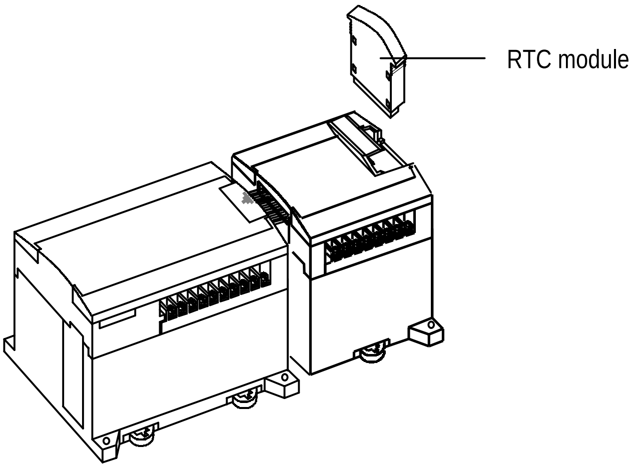


(2) Connection method
Insert the RTC module into the expansion connector of the base/expansion module at last position.

- When use base unit only



- When use base unit and expansion unit

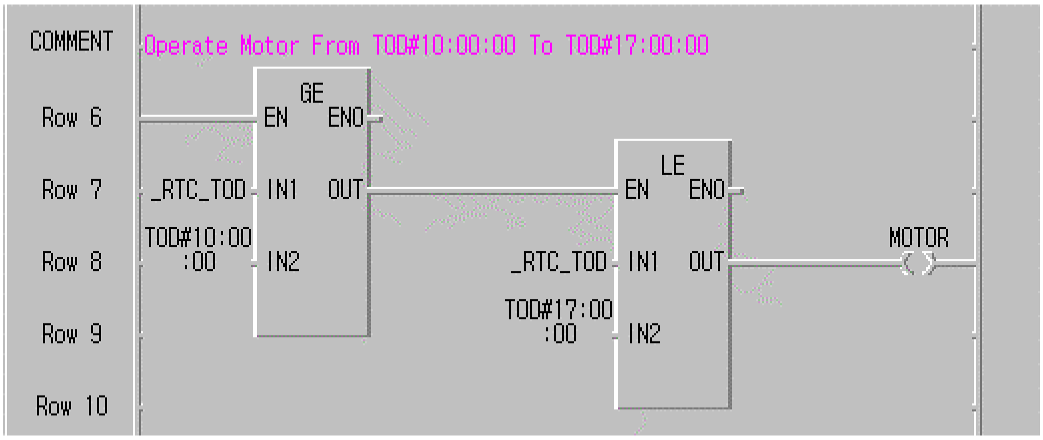


4) Read RTC data

Example : 1998. 12. 22. 19:37:46, Tuesday

Keyword	Type	Name	Description	Data
_RTC_TOD	TOD	RTC present time	Time Data	TOD#19:37:46
_RTC_WEEK	UINT	RTC present day	Day data *(0: Monday, 1:Thuesday, 2: Wednesday, 3: Thursday, 4: Friday, 5: Saturday, 6:Sunday)	1
_INT_DATE	DATE	RTC present date	Date Data of standard format (Reference date – Jan. 1, 1984)	D#1998-12-22
_RTC_ERR	BOOL	RTC data error	This flag indicates that RTC DATA error.	0
_RTC_TIME[n] * n : 0 to 7	BCD	Present time	BCD data of present time of RTC _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] : century Day of the week : 0 : Mon., 1: Tue., 2: Wed., 3:Thur., 4:Fri., 5: Sat., 6:Sun.	_RTC _TIME[0] : 16#98 _RTC _TIME[1]: 16#12 _RTC _TIME[2] : 16#22 _RTC _TIME[3] : 16#19 _RTC _TIME[4] : 16#37 _RTC _TIME[5] : 16#46 _RTC _TIME[6] : 16#1 _RTC _TIME[7] : 16#19

Example Program

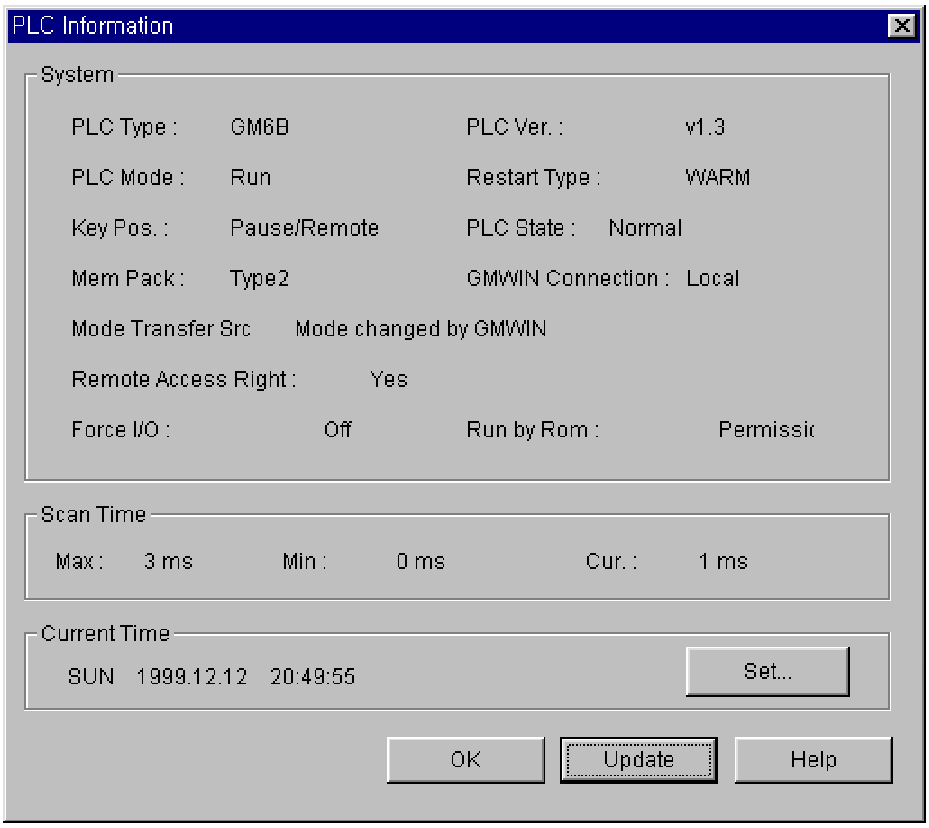


5) Write RTC data

There is two ways to write new RTC data to the CPU.

The first one is using GMWIN.

- Select **Online-PLC Information-System Info..** in the pull-down menu.



PLC Information

System

PLC Type : GM6B PLC Ver. : v1.3

PLC Mode : Run Restart Type : WARM

Key Pos. : Pause/Remote PLC State : Normal

Mem Pack : Type2 GMWIN Connection : Local

Mode Transfer Src Mode changed by GMWIN

Remote Access Right : Yes

Force I/O : Off Run by Rom : Permissi(

Scan Time

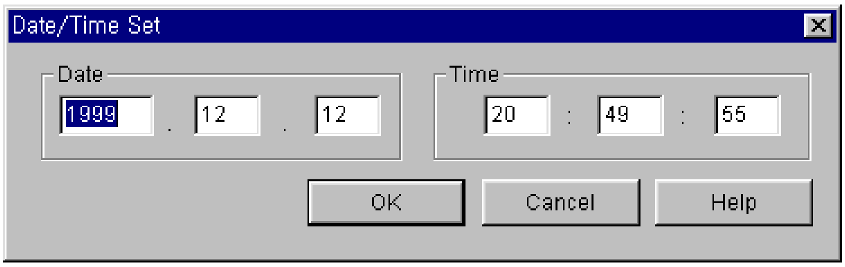
Max : 3 ms Min : 0 ms Cur. : 1 ms

Current Time

SUN 1999.12.12 20:49:55

If you want to setup or edit current time,

- Select **Set...** button of current time in **PLC Information** dialog box.



Date/Time Set

Date 1999 . 12 . 12

Time 20 : 49 : 55

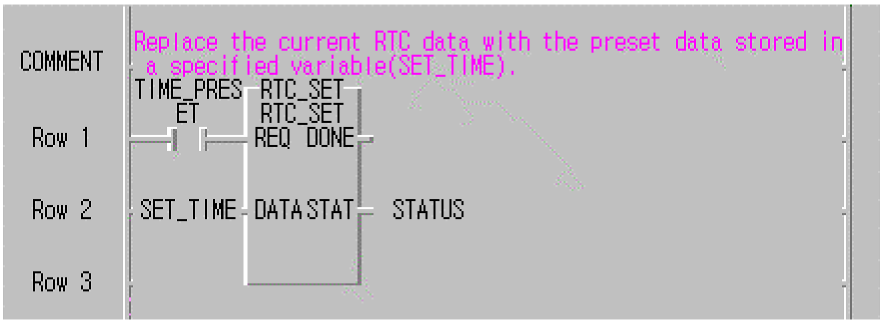
- Setup Date and Time in **Date-Time Set** dialog box.

For detailed information, refer the user's manual of GMWIN.

The second one is write sequence program. By executing a F/B(RTC_SET), user can replace the current RTC data with the preset data stored in a specified variable. The following is an example program.

Example The preset RTC data*: 1999. 1. 17. 11:53:24, Sunday*

When the ‘TIME_PRESET’ bit is switched on, the new data in ‘SET_TIME’ will be moved to ‘_RTC_TIME’.



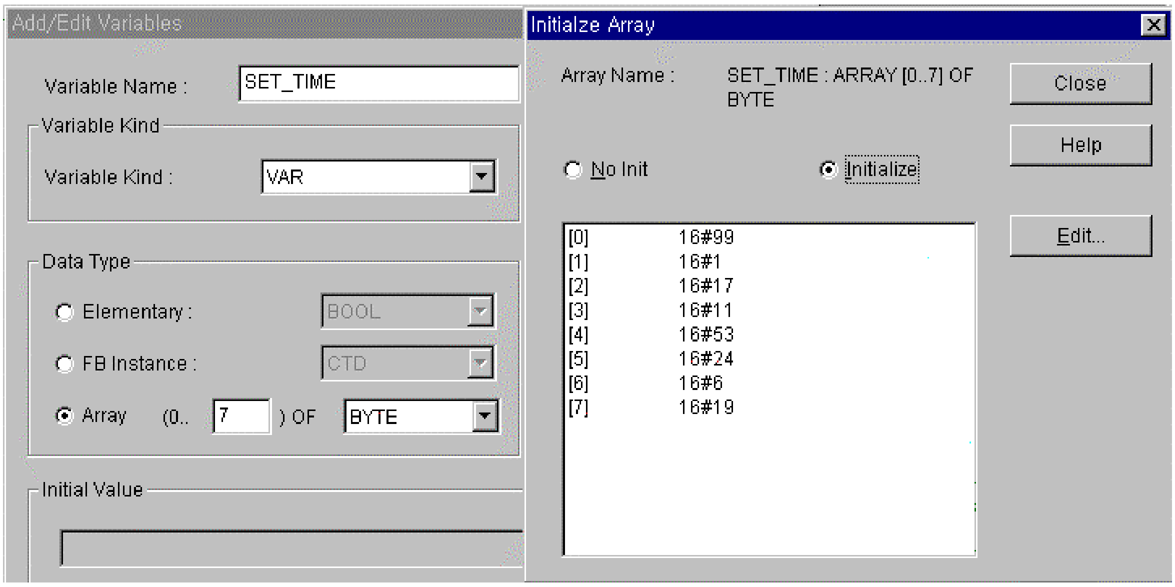
COMMENT Replace the current RTC data with the preset data stored in a specified variable(SET_TIME).

Row 1 TIME_PRESET RTC_SET RTC_SET REQ DONE

Row 2 SET_TIME DATA STAT STATUS

Row 3

*SET_TIME Variable Setting



Add/Edit Variables

Variable Name : SET_TIME

Variable Kind

Variable Kind : VAR

Data Type

○ Elementary : BOOL

○ FB Instance : CTD

● Array (0.. 7) OF BYTE

Initial Value

Initialze Array

Array Name : SET_TIME : ARRAY [0..7] OF BYTE

○ No Init ● Initialize

[0]	16#99
[1]	16#1
[2]	16#17
[3]	16#11
[4]	16#53
[5]	16#24
[6]	16#6
[7]	16#19

Error code of F/B

The following table shows error codes appear at the STAT output.

Error code	Description
00	No error
01	RTC Module is not found * Insert the RTC module into the expansion connector
02	A improper RTC data is written. Ex) 14(Month) 32(Date) 25(Hour) * Please write a correct RTC data