

## 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

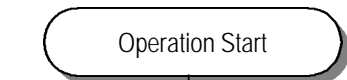
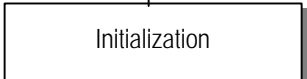
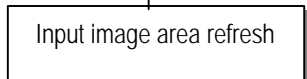
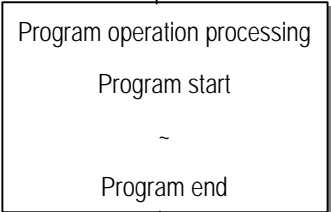
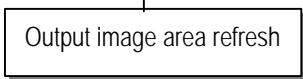
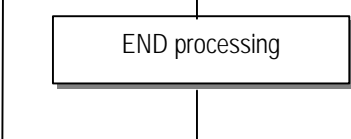
### 4.2.1 Operation Processing Method

#### 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
	-
	<ul style="list-style-type: none"> <li>• 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.                             <ul style="list-style-type: none"> <li>▶ I/O modules reset</li> <li>▶ Execution of self-diagnosis</li> <li>▶ Data clear</li> <li>▶ I/O module address allocation or type registration</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>• Input module conditions are read and stored into the input image area before operation processing of a program.</li> </ul>
	<ul style="list-style-type: none"> <li>• Program is sequentially executed from the first step to the last step</li> </ul>
	<ul style="list-style-type: none"> <li>• The contents stored in the output image area is output to output modules when operation processing of a program is finished.</li> </ul>
	<ul style="list-style-type: none"> <li>• Stage for return processing after the CPU module has finished 1 scan. The following processing are executed.                             <ul style="list-style-type: none"> <li>▶ Self-diagnosis</li> <li>▶ Change of the present values of timer and counter, etc.</li> <li>▶ Processing data communications between computer link module and communications module.</li> <li>▶ Checking the switch for mode setting.</li> </ul> </li> </ul>

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

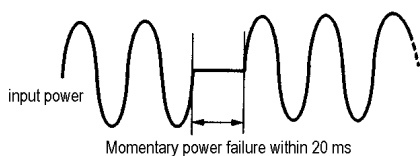
If a situation occurs which is requested to be urgently processed during execution of a PLC program, this operation method processes immediately the operation which corresponds to interrupt program. The signal which informs the CPU module of those urgent conditions is called interrupt signal. The GM6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

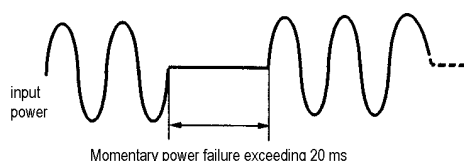
When the CPU module detects any momentary power failure, the following operations will be executed.

1) Momentary power failure within 20 ms



- (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



- 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) 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 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.

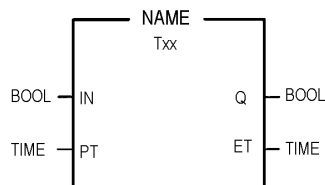
4) In order to clear watchdog error, using manual reset switch, restarting the PLC and mode change to STOP mode are available.

#### REMARK

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

### 4.2.5 Timer Processing

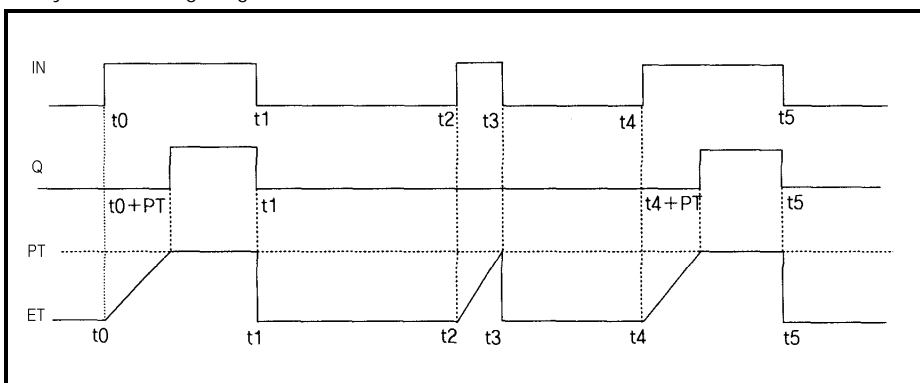
The CPU module timer is an incremental timer which increases its present value according to the measuring time. Three types of On Delay Timer (TON), Off Delay Timer (TOF) and Pulse Timer (TP) are available. Its measuring range is 0.001 to 4,294,967,295 sec (1,193 hours) by 1 ms. For details, refer to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



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

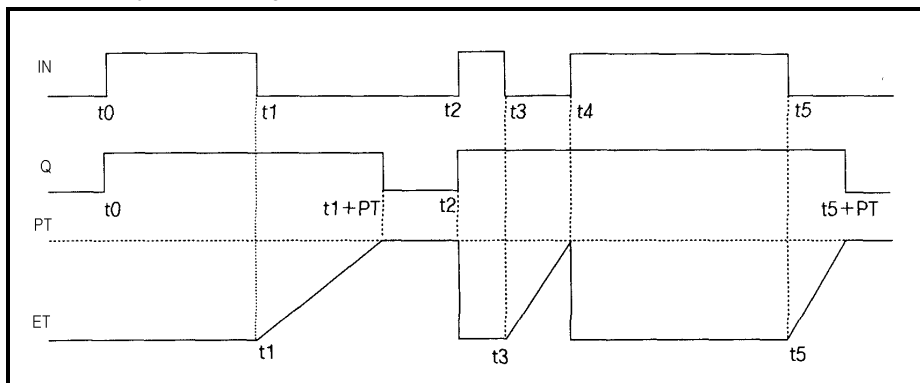
Timer Process time is newly changed when the timer function block is executed. When the process time reaches the setting time (process time = setting time), the Timer output contact turns on.

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 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.



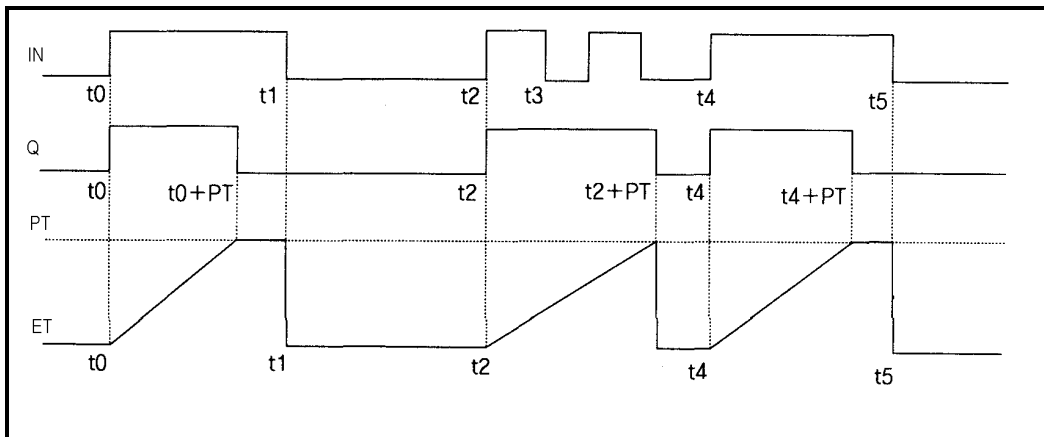
3) **Pulse Timer Process Time Change and Contact On/Off**

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

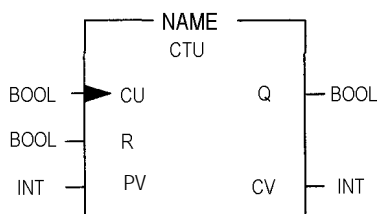
The CPU module counter increment/decrement the present counting value by the detection of rising edge(off→on) of input signal. Three types of counter are increment counter, Decrement counter and Increment-Decrement Counter. For details, refer to ‘GLOFA –GM Programming’.

- 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).

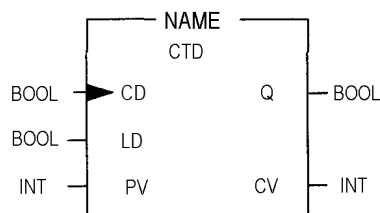


- If the counting value (CV) increments and reaches the setting value(PV) the output contact (Q) turns on.

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).

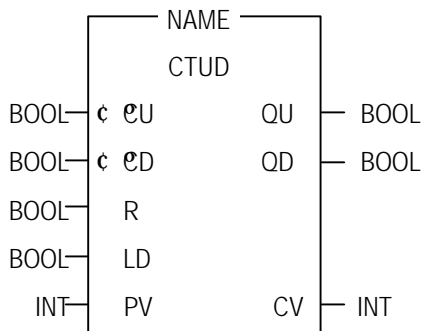


- If the counting value (CV) decrements and reaches '0', the output contact (Q) turns on.

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

(3) Increment/Decrement Counter

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



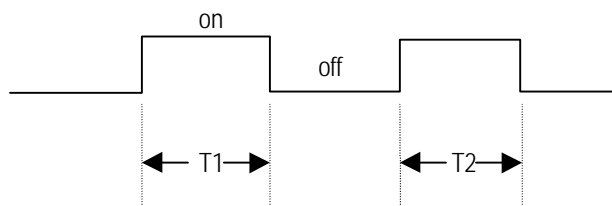
- If reset signal(R) turns on, counting value (CV) is set to '0'.
- If load signal(LD) turns on, counting value is set to setting value(PV).
- It is increased by 1 at the rising edge of increment input(CU) and decreased by 1 at the edge of decrement input(CD). If counting value(CV) is equal or larger 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.

$$\text{Max. Counting speed (Cmax.)} = n / 100 \times 1 / t_s \text{ [pps]} \quad [n : \text{Duty(\%)}, t_s : \text{scan time(s)}]$$

- Duty is percent of on time / off time.



$$T1 \text{ } \hat{\wedge} T2 : n = T1 / (T1+T2) \text{ } \times 100 \text{ [%]}$$

$$T1 \text{ } \wedge T2 : n = T2 / (T1+T2) \text{ } \times 100 \text{ [%]}$$

## 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.

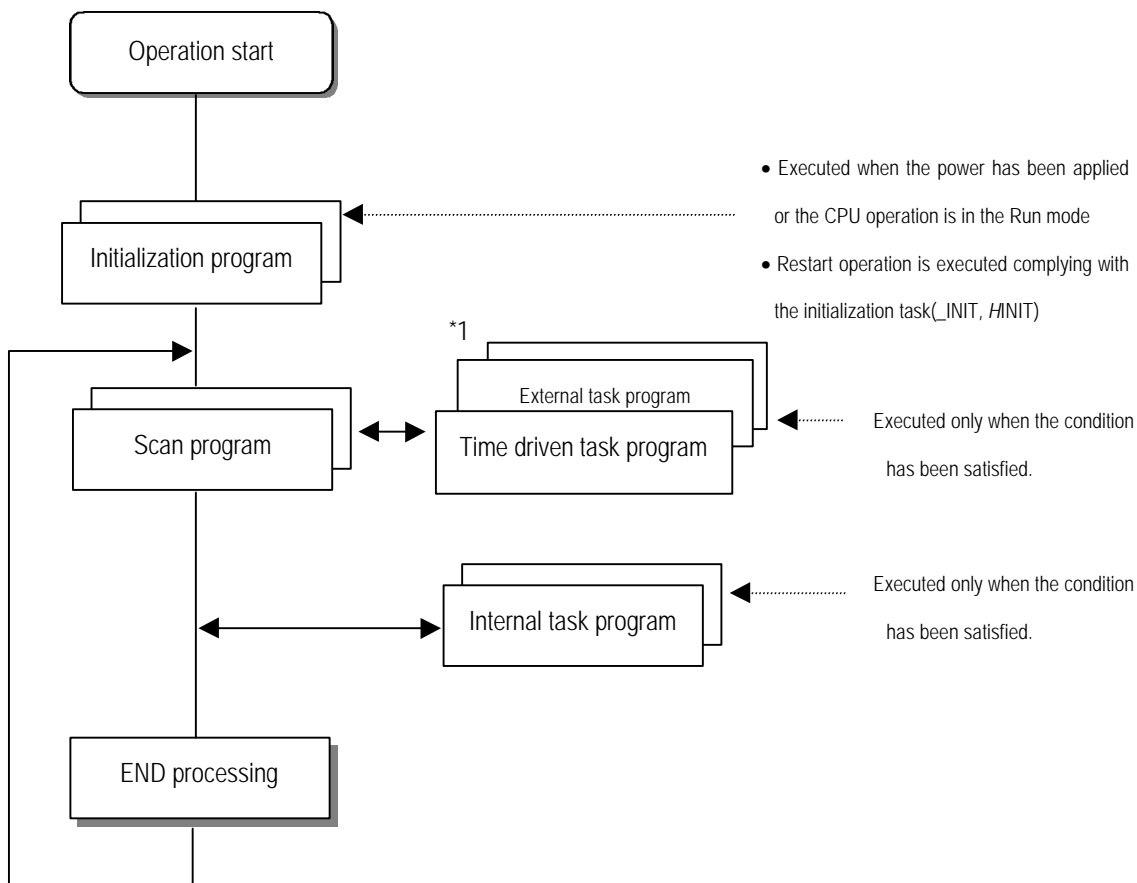
The function elements are classified as below.

Function Elements	Processing Operation
Initialization program	<ul style="list-style-type: none"> <li>• Executed when the power is applied or the CPU operation is transited to the RUN mode.</li> <li>• Executes the initial/fixes data setting for execution of scan program and the initialization of peripheral devices on special modules.</li> </ul>
Scan program	<ul style="list-style-type: none"> <li>• Processes the constantly repeated signals which are executed every scan.</li> </ul>
Time driven task program	<ul style="list-style-type: none"> <li>• When the following time conditional processing is required the program is executed complying with the time interval setting. <ul style="list-style-type: none"> <li>▶ In case that the processing need a shorter interval than that of average one scan processing time.</li> <li>▶ In case that the processing need a longer interval than that of average one scan processing time.</li> <li>▶ In case that the processing should be executed by the specified time interval.</li> </ul> </li> </ul>
Event driven task program	<ul style="list-style-type: none"> <li>• A shorter processing is executed for internal or external interrupt.</li> </ul>

### 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



**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/ Warm 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

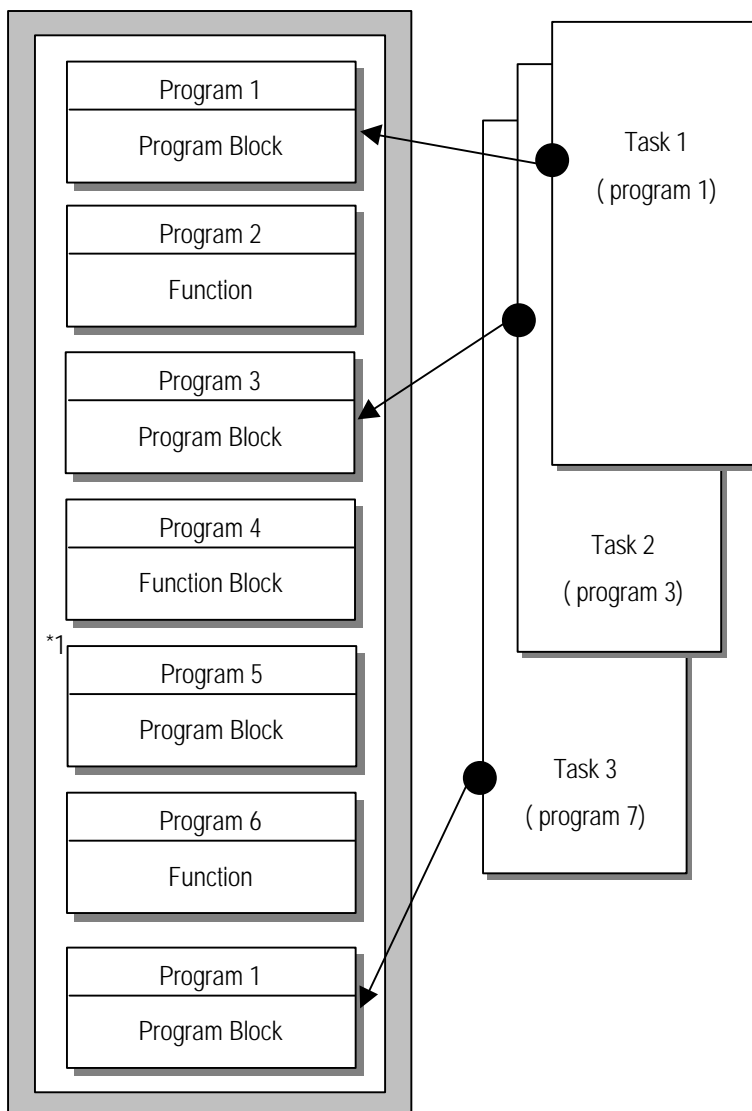
- Task programs are classified into the three types as below.
  - ▶ Time driven task program : Up to 8 programs are applicable
  - ▶ Single (internal) task program : Up to 8 programs are applicable
  - ▶ Interrupt (external) task program : Up to 8 programs are applicable
- Time driven task program
  - ▶ The program is executed by the time interval 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

<b>REMARK</b>
---------------

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 SW, 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 <sup>1)</sup>	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 occur in case that program lose its execution continuousness by the task programs which have higher priorities, 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 interval (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.