

# Appendix C

## AutoMax Task Execution

In AutoMax applications that incorporate multi-tasking, task execution is determined by eligibility to run. A task's eligibility, in turn, is determined by its priority, scan time, and any links to other tasks built into the task. This appendix describes task execution for AutoMax application tasks. Refer to the appropriate Configuration and Programming instruction manual for a description of the task execution for UDC tasks.

### Priority

Task priority refers to the relative importance of a task in the application. Tasks whose execution is more critical to the operation of the controlled machinery should be assigned higher priority. Priority can range from 4 (highest) to 11 (lowest). Priority is specified by the programmer when adding a task to the rack.

### Scan Time

Scan time refers to how often the task is scheduled to be executed. It is usually measured in ticks. You can assign a tick rate for each AutoMax Processor. The tick rate can range from 0.5 ms to 10.0 ms. The default tick rate is 5.5 ms. See section 6.1 of this manual for more information. For example, an AutoMax task with a 20 tick scan time (using the default tick rate) means that the task is scheduled to start every 110ms.

Scan time is specified differently for PC/Ladder Logic, Control Block, and BASIC tasks. In each case, before setting the scan time, it is important to know approximately how long the task takes to execute. For example, a task that takes 20ms to execute cannot be assigned a scan time of 2 ticks because a 20ms task cannot be started every 11ms. This situation would result in an overlap error (error code 14 would be displayed on the Processor and all tasks in the rack would be stopped).

Scan time for PC/Ladder Logic tasks is set using the Ladder Editor. The Program Properties window is used by the programmer to enter the scan time. The screen display will indicate the approximate execution time of the task, determined by the Executive software. The AutoMax Ladder Logic language instruction manual contains execution time estimates for each type of Ladder Logic operation.

For AutoMax Control Block tasks, the scan time is specified in ticks using the SCAN\_LOOP statement. The approximate execution time of the task can be calculated using the execution time estimates found in the AutoMax Control Block language instruction manual (J-3676).

The scan time for BASIC tasks is usually specified in the START EVERY statement. The START EVERY statement, which is optional, allows the programmer to use seconds, minutes, and hours, as well as ticks, as the time unit. Note that there is no simple method of estimating execution time for BASIC tasks ahead of time. One method of estimating execution time for a specific BASIC task is to include a statement that turns on a DC output at the beginning of the task (after the START EVERY statement, if used) and turns it off at the very end of the task. Then the task is put into run with no other tasks running, and an oscilloscope is used to measure the time that the output is on.

## Synchronizing Tasks

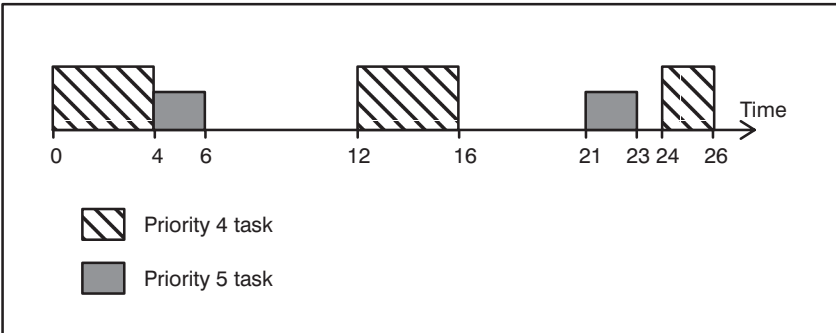
The implications of scan time and priority are particularly important in applications that require tasks to share data or to be synchronized based on hardware or software events. Priority, however, cannot be used to synchronize tasks on different Processors. This is because each Processor controls only the scheduling of the tasks that reside on it. Therefore, the concept of task priority is relevant only for the Processor on which the tasks reside. Task synchronization between AutoMax Processors and UDC modules is handled by interrupts. See the DPS Configuration and Programming instruction manual for more information about UDC tasks.

In addition to priority, there are other methods of synchronizing tasks that work with both tasks on the same Processor and tasks on different Processors. In BASIC tasks, statements such as DELAY, WAIT, SET, OPEN, and INPUT can be used for this purpose. In Control Block tasks, the EVENT parameter in the (required) SCAN\_LOOP statement is used to synchronize tasks. PC/Ladder Logic tasks use the EVENT coil for task synchronization.

## Examples of Task Execution

The scan time required for the highest priority task on the AutoMax Processor should be used as a guide to determine how tasks should be scheduled. It is also necessary to factor in system overhead, e.g., communication with the personal computer, when assigning priority and scan time. Total Processor utilization should be kept to 80-85%. Processor utilization can be monitored using the Info/Log option in the Executive software.

See the figure below for an example of task execution. Assume you have a priority 4 PC/Ladder task that needs to be executed approximately every 12 ticks and that takes 4 ticks to execute. Assume you also have a less critical priority 5 BASIC task that needs to be executed approximately every 17 ticks and takes 2 ticks to execute. The PC/Ladder task would start at time 0, then finish execution at 4 ticks. At this point the BASIC task would be eligible to run (its time 0).

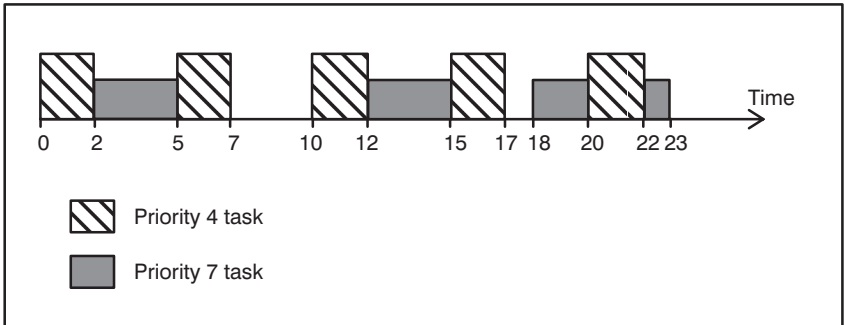


If there are any scheduling conflicts between tasks, i.e., more than one task is ready to run, priority takes precedence over scan time. A higher priority task that is scheduled to run can suspend a lower priority task, regardless of whether the lower priority task is ready to begin execution or has already begun executing. The suspended task is allowed to execute or continue execution from where it was suspended only after the higher priority task is finished.

Consider the following example. There are two tasks on a Processor: a priority 4 Control Block task that needs to be executed every 5 ticks and a priority 7 BASIC task that needs to be executed every 8 ticks. Assume the Control Block task requires 2 ticks to execute and the BASIC task requires 3 ticks to execute.

The Control Block task starts, and finishes executing 2 ticks later. At this point, the BASIC task is eligible to run (its time 0). When it finishes, the time elapsed from the point of view of the Control Block task is 5 ticks and the higher priority task can then execute again. At 10 ticks, however, both the Control Block and the BASIC task are scheduled to run. Because the Control Block task has a higher priority, it will execute first. The BASIC task will execute immediately afterwards.

Note that although the BASIC task was prevented from executing at the originally scheduled 10 ticks, it is still scheduled to run again at the next 8-tick interval, i.e., at 18 ticks. Two ticks into this scan, however, it will be suspended by the Control Block task which is scheduled to run at 20 ticks. After the Control Block task completes its scan, the BASIC task will continue the remainder of its scan.



## Initial Scan

In applications that require multiple interdependent tasks, the programmer needs to ensure that the structure of the individual tasks takes into account the possible timing lag between execution of different types of tasks. The first scan of the task is used to execute only the initialization code for the following types of tasks: all BASIC tasks with a START EVERY statement, all PC/Ladder Logic tasks, and all Control Block tasks.

Control Block initialization code consists of all statements before the SCAN\_LOOP statement. BASIC task initialization code consists of all statements before the START EVERY statement. PC/ladder Logic task initialization code is constructed by the Executive software.

BASIC tasks without a START EVERY statement do not have initialization code and are always executed in total (if they are not suspended by a higher priority task which is scheduled to run) during their first scan. Execution of BASIC tasks without a START EVERY statement is based only on their priority or their links to other tasks because there is no method of establishing scan time without a START EVERY statement. Note that this type of BASIC task is executed only once unless it includes a GOTO statement that results in a loop.