

53.0 EXECUTION TIME ESTIMATES

Execution time can be affected by remark statements within a task. There are two forms of the remark statement: REM and I. Remark statements using "REM" are stripped off before the task is downloaded and, therefore, do not affect execution time. However, remark statements using "I" do require execution time when the task runs on the Processor. For tasks that will run on 6010/6011 Processors, use a value of 40 microseconds as the time for each "I" remark. For tasks that will run on 7010 Processors, use a value of 9.88 microseconds as the time for each "I" remark. For tasks that will run on UDC modules, use 2.37 microseconds as the time for each "I" remark. The amount of text that follows the "I" does not affect the execution time; no text requires the same time as a full line of text. For computing CPU usage for the task, only those remark "I" statements that occur after the SCAN_LOOP block need be considered. Those that occur before the SCAN_LOOP block are executed only once and, therefore, need not be considered.

53.1 AutoMax Processor and AutoMax PC3000 Control Block Tasks

The execution times for all control block statements programmed in a particular AutoMax task can be summed with the SCAN_LOOP/END time to estimate the total task execution time. This value divided by the scan time (TICKS * tick rate in seconds) gives the estimated CPU usage for the control block task. Refer to Table 4 for AutoMax Processor block execution times. Refer to Table 6 for AutoMax PC3000 block execution times. Note that the times listed are valid for Version 3.4 and later of the AutoMax Programming Executive software.

53.2 UDC Tasks

The execution times for all Control Block statements programmed into a particular UDC Control Block tasks can be summed with the SCAN_LOOP/END time to estimate execution time for that UDC Control Block task. If two Control Block tasks will be running on a particular UDC module, the estimated execution times of both tasks must be added together. The total execution time divided by the tick rate (.5 milliseconds) will equal the minimum value you should enter for the TICKS parameter in the SCAN_LOOP block. **Note that you must enter the same value for the TICKS parameter in the SCAN_LOOP block for both tasks.** The maximum value you can enter for the TICKS parameter is 20 (10 milliseconds). Refer to Table 5. Note that the times listed are valid for Version 3.4A of the AutoMax Programming Executive software.

Table 4 - Maximum Execution Time Summary - AutoMax Tasks

Block Name	7010 Processor Maximum Time (μsec)	6010/6011 Processor Maximum Time (μsec)
ABSOLUTE_VALUE	9.16	41
ALARM	11.16 + n(2.48)	44 + n(11)
AMPLIFIER	13.16 + k(3.12)	62 + k(13.8)
AND	10.80 + j(1.88)	41 + j(8.3)
BIT_SELECT	26.04 + n(1.20)	118 + n(6.5)
COMPARE	11.24 + n(1.16)	45 + n(5.5)
COUNTER	15.64	61
DIFFERENCE	10.80	44
DIFF_LAG	33.04	154
END	13.82 + t(2.0) + i(1.6) + b(2.2) + d(2.5)	60 + t(3.2) + i(2.8) + b(4.3) + d(4.9)
FUNCTION	23.56	93
INTEGRATE	39.14	193
INVERTER	10.84	48
LAG	26.52	118
LATCH	15.04	68
LEAD_LAG	68.60	284
LIMIT	18.20	69
MOVE	10.40 + p(2.32)	39 + p(11)
MULTIPLY_DIVIDE	13.76	71
NOTCH	85.16	393
OR	10.24 + j(1.88)	41 + j(8.3)
PACK_BITS	24.28 + j(1.44)	110 + j(5)
PID	124.00	515
PROP_INT	40.32	193
PULSE_MULT	18.32	86
RAMP	27.72	122
READ_BITS	24.32 + n(1.12)	106 + n(7.3)
READ_WORDS	25.76 + n(1.28)	111 + n(5.6)
RUNNING_AVERAGE	28.64	125
SAMPLED_AVERAGE	29.00	125
S_CURVE	47.00	282
SCALE	22.48	98
SCAN_LOOP	80.00 + t(1.6) + i(2.5) + b(3.8) + d(4.1)	326 + t(2.4) + i(4.4) + b(6.8) + d(7.2)
SEARCH	19.80 + m(0.68)	74 + m(3)
SELECT	11.36 + j(2.52)	44 + j(12.2)
SHIFT_BITS	18.28 + p(2.04)	92 + p(19.7)
SHIFT_WORDS	17.80 + p(1.72)	89 + p(13.3)
SUMMER	11.28	44
SWITCH	11.52	47
TRANSITION	11.36	48
UNPACK_BITS	23.40 + n(1.44)	106 + n(7.3)
WRITE_BITS	25.56 + j(1.36)	103 + j(6)
WRITE_WORDS	25.52 + j(0.36)	110 + j(1.9)
I<REM>	9.88	40

b = number of common boolean variables referenced by the task
 d = number of common double integer variables referenced by the task
 i = number of common integer variables referenced by the task
 j = total number of inputs programmed
 k = total number of input pairs programmed
 m = number of elements searched
 n = total number of outputs programmed
 p = total number of input/output pairs programmed
 t = total number of common integer, boolean, and double integer variables referenced by the task

Table 5 - Maximum Execution Time Summary - UDC Tasks

Block Name	UDC Module Maximum Time (μsec)
ABSOLUTE_VALUE	8.30
ALARM	10.10 + n(1.80)
AMPLIFIER	15.30 + k(1.80)
AND	10.60 + j(1.45)
COMPARE	7.50 + n(2.25)
COUNTER	16.90
DIFFERENCE	7.97
DIFF_LAG	41.50
END	130.00 + i(2.40) + b(2.50) + d(3.30) + z
FUNCTION	20.30
INTEGRATE	58.60
INVERTER	9.40
LAG	41.00
LAGN	39.25 + o(4.00)
LEADN	39.25 + o(4.00)
LATCH	14.85
LEAD_LAG	39.70
LIMIT	15.05
MOVE	10.40 + p(1.80)
MULTIPLY_DIVIDE	13.80
NOTCHN	64.25 + o(20.00)
OR	8.90 + j(1.45)
PACK_BITS	16.16 + n(1.25)
PROP_INT	70.85
PULSE_MULT	18.96
RAMP	24.96
RUNNING_AVERAGE	26.94
SAMPLED_AVERAGE	25.94
S_CURVE	35.01
SCALE	18.30
SCAN_LOOP	350.00 + i(2.20) + b(2.00) + d(3.30) + x
SELECT	12.50 + j(1.50)
SUMMER	10.00
SWITCH	12.55
TACHLOSS_OVERSPEED	22.00
THERMAL_OVERLOAD	46.10
TRANSITION	11.90
UNPACK_BITS	14.40 + n(1.90)
I<REM>	2.37

b = number of common boolean variables referenced by the task
d = number of common double integer variables referenced by the task
i = number of common integer variables referenced by the task
j = total number of inputs programmed
k = total number of input pairs programmed
n = total number of outputs programmed
o = "order" of filter
p = total number of input/output pairs programmed
x = avg. feedback message overhead per task; add 312 μsec when only 1 task executes.
z = avg. setpoint message overhead per task; add 100 μsec when only 1 task executes

Table 5 - Maximum Execution Time Summary - AutoMax PC3000 Tasks

Block Name	PC3000 Maximum Time (μsec)
ABSOLUTE_VALUE	6.44
ALARM	$7.8 + n(1.76)$
AMPLIFIER	$9.2 + k(2.20)$
AND	$7.56 + j(1.32)$
BIT SELECT	$18.24 + n(0.84)$
COMPARE	$7.88 + n(0.80)$
COUNTER	10.96
DIFFERENCE	7.56
DIFF LAG	23.12
END	$9.68 + t(1.40) + i(1.12) + b(1.56) + d(1.76)$
FUNCTION	16.48
INTEGRATE	27.40
INVERTER	7.6.0
LAG	18.56
LATCH	10.52
LEAD LAG	48.04
LIMIT	12.76
MOVE	$7.28 + P(1.64)$
MULTIPLY_DIVIDE	9.64
NOTCH	59.60
OR	$7.16 + j(1.32)$
PACK_BITS	$17.00 + j(1.00)$
PID	86.80
PROP INT	28.24
PULSE_MULT	12.84
RAMP	19.40
READ BITS	$17.04 + n(0.80)$
READ_WORDS	$18.04 + n(0.92)$
RUNNING_AVERAGE	20.04
SAMPLED_AVERAGE	20.32
S_CURVE	32.92
SCALE	15.72
SCAN LOOP	$56.00 + t(1.76) + b(2.68) + d(2.88)$
SEARCH	$13.88 + m(0.48)$
SELECT	$7.96 + j(1.76)$
SHIFT BITS	$12.80 + p(1.44)$
SHIFT_WORDS	$12.48 + p(1.20)$
SUMMER	7.88
SWITCH	8.08
TRANSITION	7.96
UNPACK BITS	$16.40 + n(1.00)$
WRITE BITS	$17.88 + j(0.96)$
WRITE_WORDS	$17.88 + j(0.28)$
I<REM>	6.92

b = number of common boolean variables referenced by the task
d = number of common double integer variables referenced by the task
i = number of common integer variables referenced by the task
j = total number of inputs programmed
k = total number of input pairs programmed
m = number of elements searched
n = total number of outputs programmed
p = total number of input/output pairs programmed
t = total number of common integer, boolean, and double integer variables referenced by the task