

Information needed to analyze Real-Time code includes:

1. All mutually exclusive hardware accesses, data accesses, kernel accesses
2. All places where interrupts are disabled
3. All shared data
 - Used by every task
 - Duration of critical section
4. All task precedence constraints
5. Additional task timing constraints (jitter, end-to-end requirements)
6. All task priorities
7. How your compiler:
 - Represents all data types (native sizes)
 - Writes to memory – identify all non-atomic writes
8. Worst case stack (and heap) usage, as well as usage of other shared, limited resources
9. Worst case execution time of every task
10. All task timing parameters:
 - Release times
 - Execution times
 - Deadlines
 - Periods
 - Self-suspend times
 - Non-preemption times
11. Execution time and stack depth of all library routines used
12. Kernel particulars
 - where are interrupts disabled, and for how long
 - resource usage: stack, any other resource
 - execution time of all calls used
 - non-preemptible system calls
 - context switch time
 - scheduling algorithm

Highlighted items are required even if there are no critical timing requirements!