FreeRTOS Demo Robot

Anthony Savitt & Jacob Carver

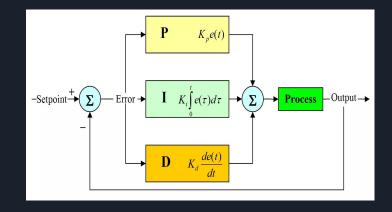
FreeRTOS Features Used

- Tasks
- Queues
- Semaphores
- Timers
- ISR Management
- FreeRTOS+Tracealyzer
 - Data Acquisition

Tasks to Perform and Optimize

- 1. Get distance data from LIDAR sensor
 - a. Read data into buffer from 8 bytes
 - b. Return buffer

- 2. Calculate pwm value from PID loop
 - a. Our target distance: 40cm
 - b. Max distance before full speed: 90cm



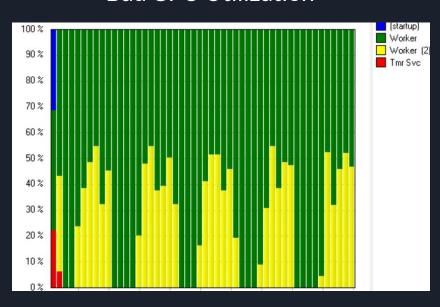
3. Perform PWM

Experiment #1 Basic Tasks and Queues

Minimal task use



Bad CPU Utilization

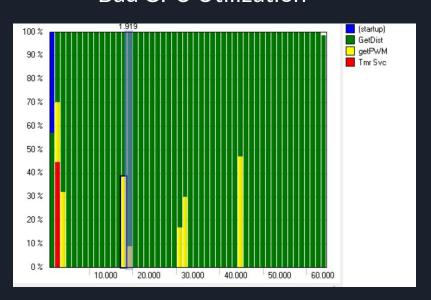


Experiment #2 Improved Tasks and Queues

Minimal task use



Bad CPU Utilization

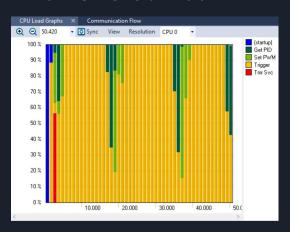


Experiment #3 Callbacks & Semaphores

Lots of Fragmentation



Bad CPU Utilization



Test Conditions:

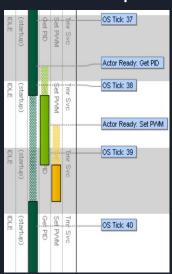
- Utilized USART callback mode
- Use of Semaphores for callback
- Separate Data Queues

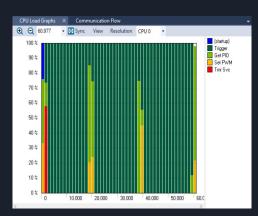
Results:

- Improved synchronization.
- Reduced program complexity
- CPU utilization is still inefficient
- Task execution order is chaotic.
- Set PWM Exec: 773µs
- Set PWM Response: 2.169ms

Experiment #4 Semaphores w/ Priorities

Improved Execution Order





Test Conditions:

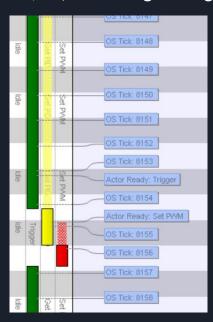
- PID set to Priority 2
- PWM set to Priority 2
- RX set to Priority 1(RX < PWM & PID)

Results:

- Vital PWM and PID management tasks are no longer interrupted mid execution.
- CPU utilization was not addressed in this experiment.
- SetPWM Exec: 444µs
- SetPWM Response: 787μs
- SetPWM Fragmentation: 1

Experiment #5 Disabling Preemption

PID Task (Yellow, P2) is Waiting on Trigger(Green, P1)



Test Conditions:

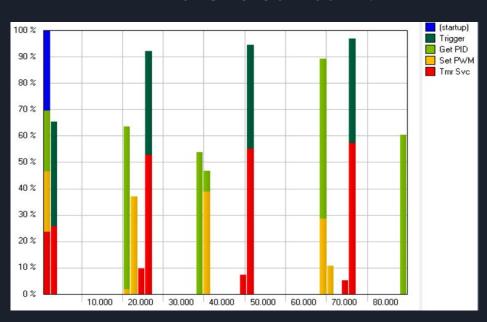
- Same as Experiment #4
- Disabled Preemption in FreeRTOS settings.

Results:

- Preemptive scheduling is a must
- Nothing we didn't expect

Experiment #6 Timers(Best Result)

Minimal CPU Utilization!



Test Conditions:

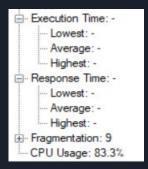
- Added timer to RX
- Removed Semaphores
- Removed unneeded single RX Trigger jobs
- Everything else is the same as Experiment #4

Experiment #6 Timers(Best Result)

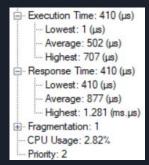
Results:

- IDLE task at 83.3% CPU Usage!
- Plenty of room for other tasks to be added in.
- RX data acquisition rate can be arbitrarily set depending on needs.

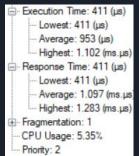
IDLE



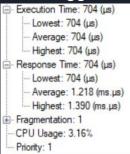
Set PWM



Get PID

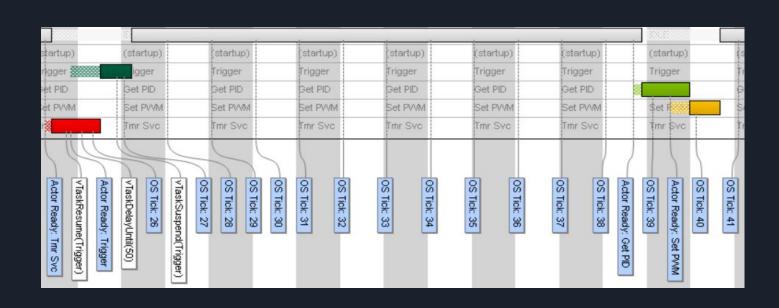


RX Trigger



Experiment #6 Timers(Best Result)

Lots of Available IDLE Time!



Emergency Slide (ONLY IF NECESSARY)

