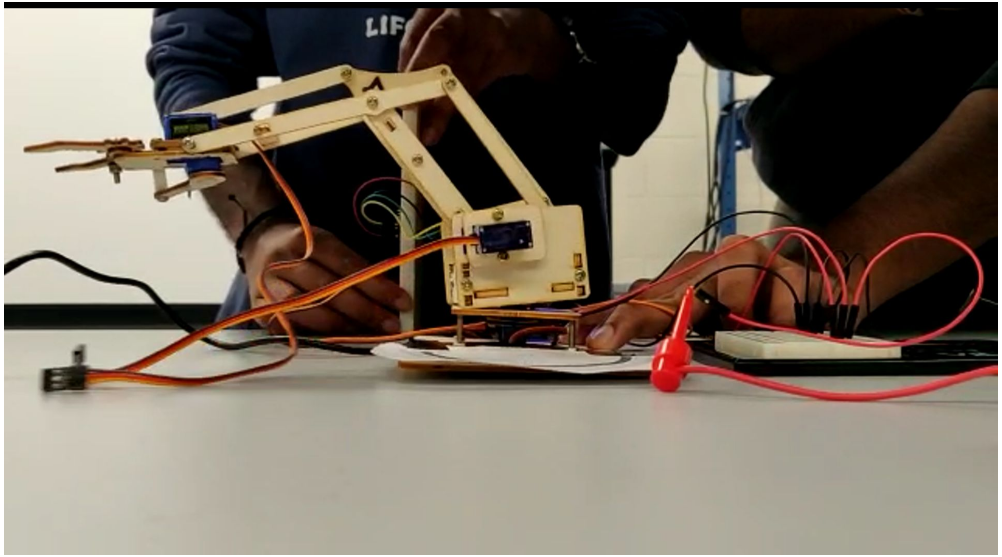
ECEN 5623 - RTES

Motion Controlled ROBOT ARM

Vishnu Dodballapur and Pradyumna Gudluru

Project Overview

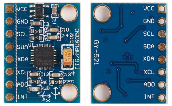
- Motion Controlled Robotic Arm
    - Robot base and arm move based on pitch and roll detected by accelerometer
- Arm functions like a crane
- Mostly recreational



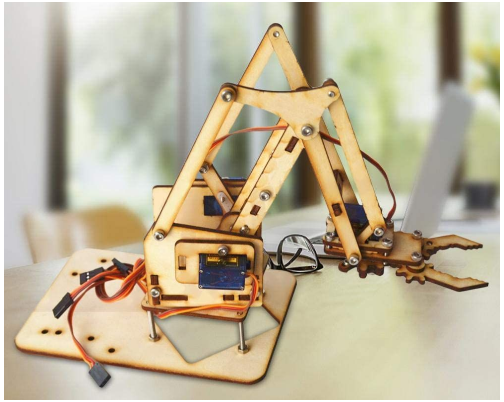
Hardware

- Microcontroller: Tiva Launchpad

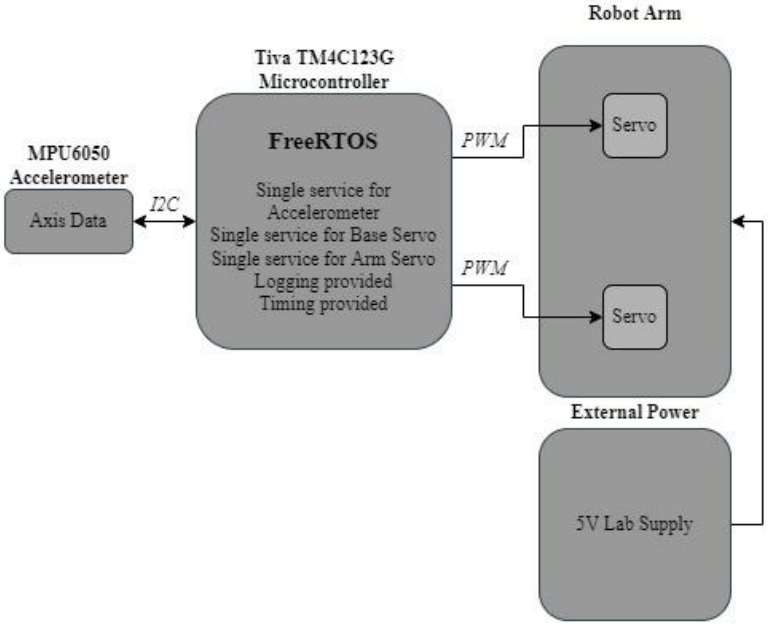
    - Runs FreeRTOS
- Accelerometer: MPU6050



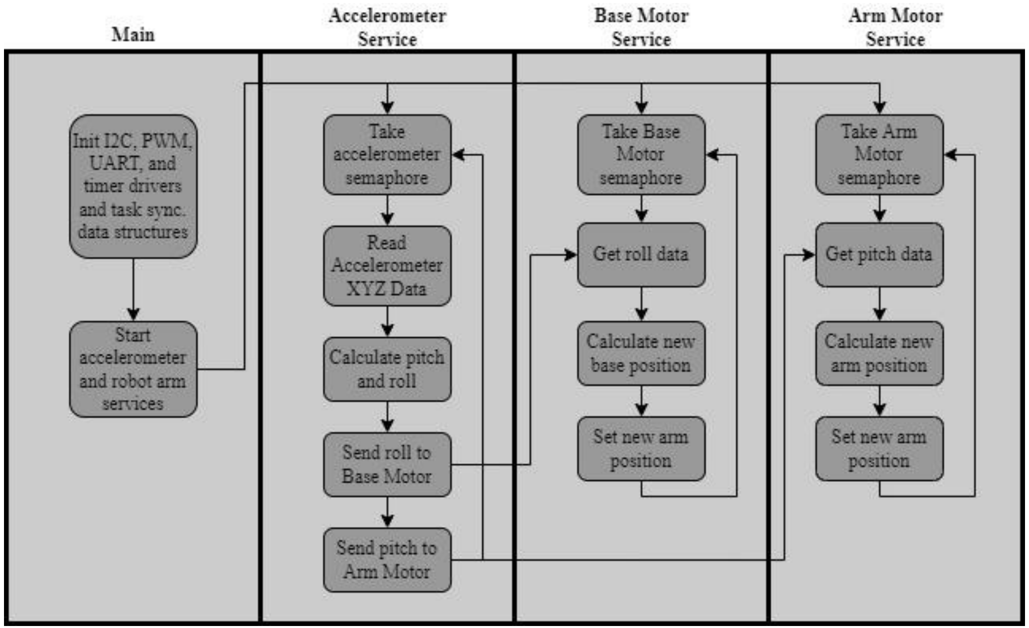
    - I2C Communications
- Robot Arm: SNAM1500 4 DOF Wood Robotic Mechanical Arm
    - Driven by SG90 Servo Motors



Block Diagram



Software Flow Diagram



Capability Requirements

- System will run FreeRTOS
- Processor will communicate with accelerometer via I2C
- Processor will set position of robot elements with 50Hz PWM signal
- Processor will read XYZ values from the accelerometer
- Processor will convert XYZ values to pitch and roll
- Processor will set position of robot base depending on roll
- Processor will set position of robot arm depending on pitch
- Base of robot will have 180 degrees of motion
- Arm of robot will have 60 degrees of motion

- Stretch Goal: Processor and accelerometer will enter low power mode when waiting or sleeping
- Stretch Goal: Will engage third servo on claw to grab object off of button press

Real Time Requirements

- Decided on end-to-end response time of 500ms for robot arm build with commercially available parts based off of the following paper
  - https://smartech.gatech.edu/handle/1853/37380
- Accelerometer has input latency of 10ms
  - Takes 10ms to get a new reading
- Servos take 100ms to move 60 degrees
  - Worst case movement is 300ms to move 180 degrees
- Effective Deadline for all Services: 500ms - 300ms - 10ms = 190ms

Real Time Services, Ti, Ci, Di

- Service 1: Accelerometer Service (Higher priority service)
  - Ci: 2110µs
  - Ti: 25ms
  - Di: 25ms
- Service 2: Base Motor Service (Lesser priority service)
  - Ci: 134.5µs
  - Ti: 50ms
  - Di: 50ms
- Service 3: Arm Motor Service (Lesser priority service)
  - Ci: 138µs
  - Ti: 50ms
  - Di: 50ms

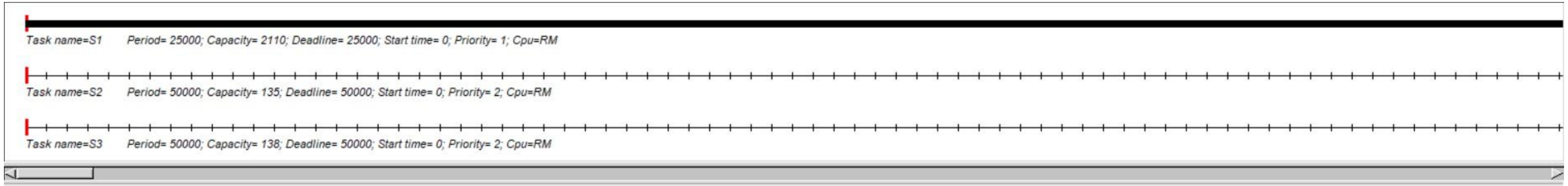
WCET

- Service 1: Accelerometer Service
  - Computation includes I2C communication, pitch and roll calculations (floating point math!), place value in message queues
  - Calculated to be 2110µs
- Service 2: Base Motor Service
  - Computation includes receiving from message queue, converting roll angle to duty cycle, and setting servo position
  - Calculated to be 134.5µs
- Service 3: Arm Motor Service
  - Computation includes receiving from message queue, restricting range of motion to 60 degrees, converting pitch angle to duty cycle, and setting servo position
  - Calculated to be 138µs

Verification Plan

- Timing
  - Logging via UARTprintf() using hardware timer at 1µs resolution
- Accelerometer Data
  - Logging of pitch and roll via UARTprintf()
- Robot Angle
  - Protractor cut-out to verify correct angles

Feasibility Analysis



Scheduling simulation, Processor RM :

```
- Number of context switches :  3
- Number of preemptions :  0

- Task response time computed from simulation :
    S1 => 2110/worst
    S2 => 2383/worst
    S3 => 2248/worst
- No deadline missed in the computed scheduling : the task set seems to be schedulable.
```

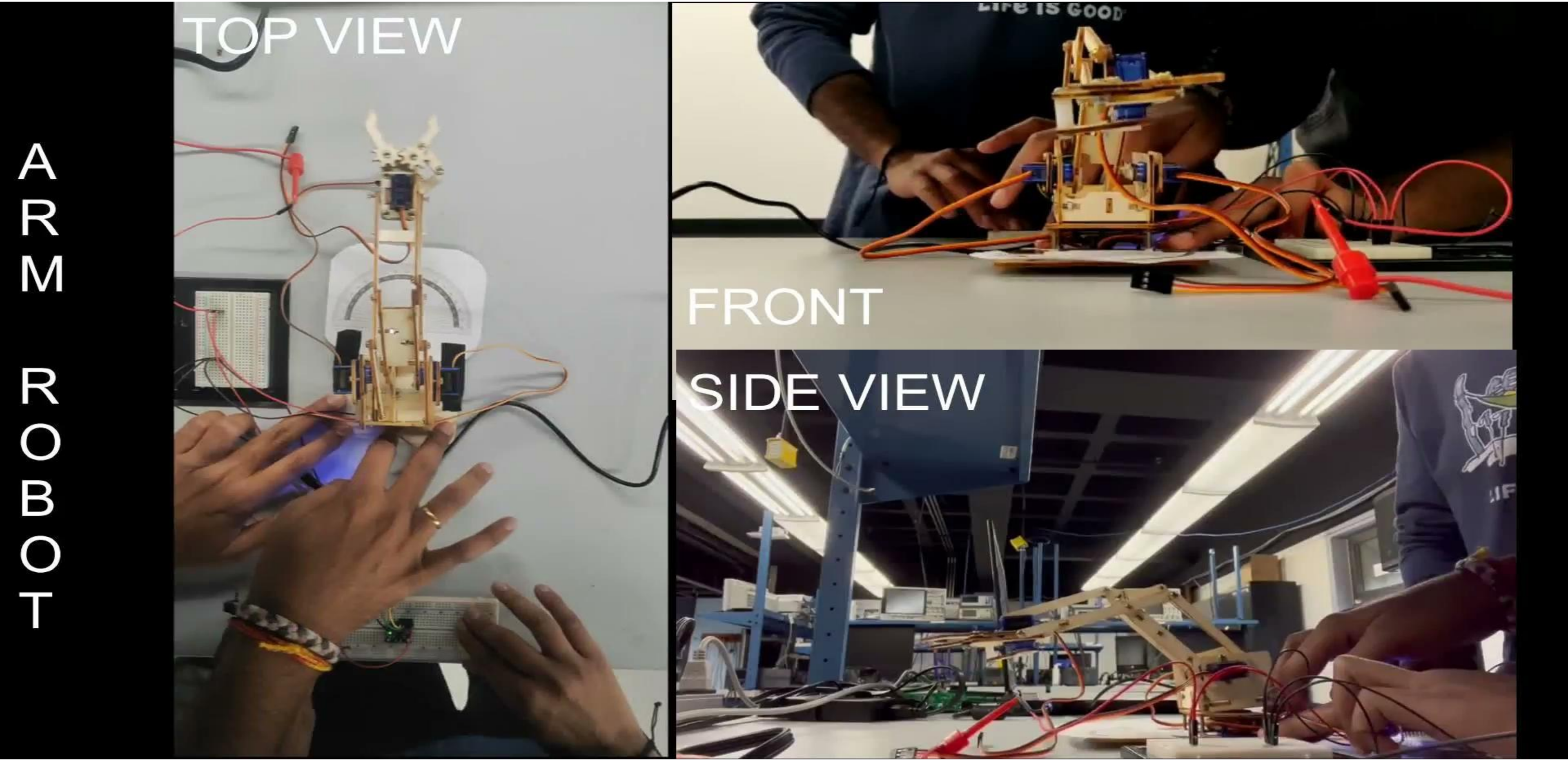
Scheduling feasibility, Processor RM :

```
1) Feasibility test based on the processor utilization factor :

- The base period is 50000 (see [18], page 5).
- 45507 units of time are unused in the base period.
- Processor utilization factor with deadline is 0.08986 (see [1], page 6).
- Processor utilization factor with period is 0.08986 (see [1], page 6).
- In the preemptive case, with RM, the task set is schedulable because the processor utilization factor 0.08986 is equal or less than 1.00000 (see [19], page 13).


2) Feasibility  test based on worst case task response time :

- Bound on task response time :  (see [2], page 3, equation 4).
    S2 => 2383
    S3 => 2248
    S1 => 2110
- All task deadlines will be met : the task set is schedulable.
```

Feasibility Analysis

- Per Cheddar, system is schedulable
- CPU Utilization U=0.08985
- To be feasible by RM LUB, U must be less than 0.7797, the RM LUB for 3 services
    - System is feasible!

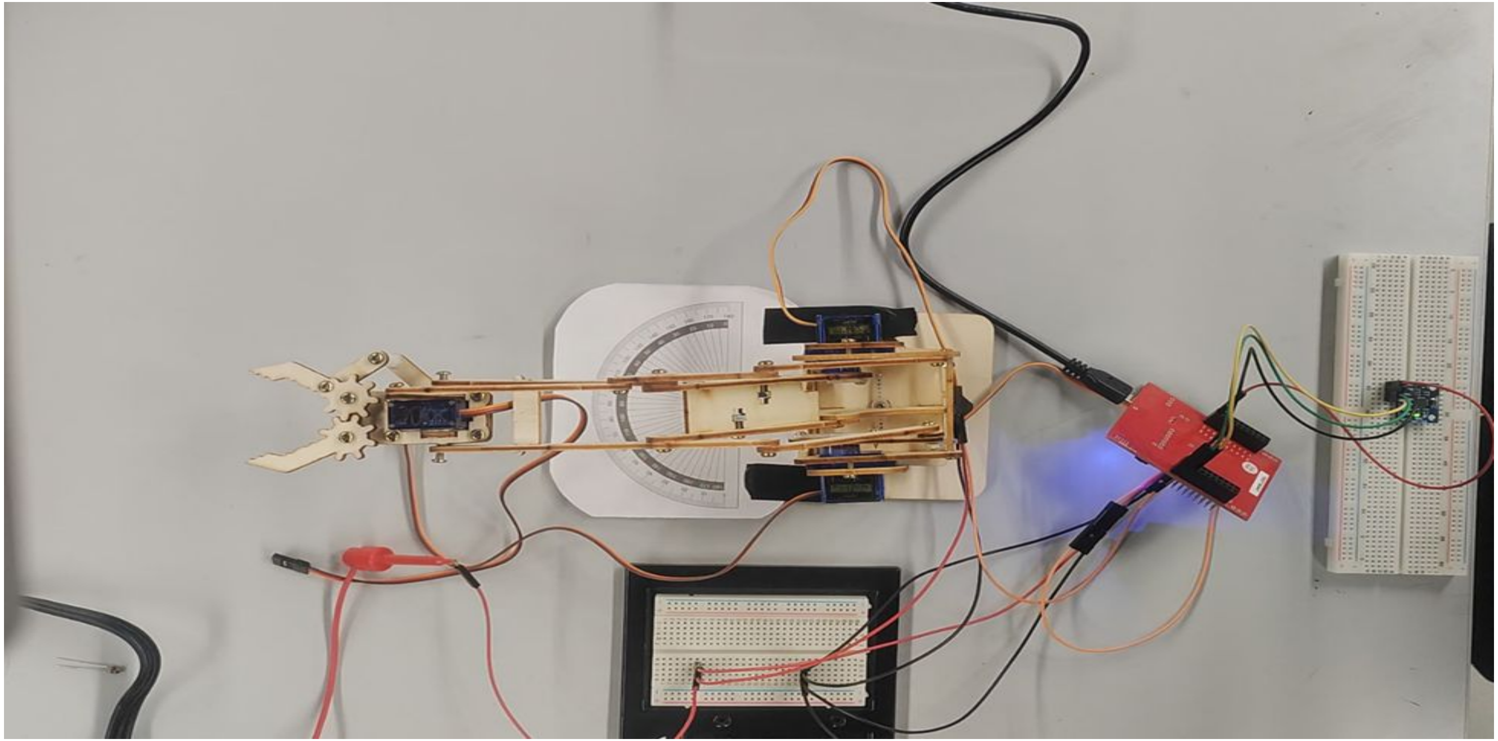
Proof of Concept Implementation



Proof of Concept Verification

```
MPU6050 Task Start: 5209 ms.
MPU6050 Read: 1672 us.
MPU6050 Task Start: 5237 ms.
MPU6050 Read: 1670 us.
Base Motor Task Start: 5237 ms.
Base Motor Set Angle: 16 us.
Roll: 20 degrees.
Arm Motor Task Start: 5248 ms.
Arm Motor Set Angle: 17 us.
Pitch: -11 degrees.
MPU6050 Task Start: 5263 ms.
MPU6050 Read: 1672 us.
MPU6050 Task Start: 5291 ms.
MPU6050 Read: 1671 us.
Base Motor Task Start: 5291 ms.
Base Motor Set Angle: 16 us.
Roll: 70 degrees.
Arm Motor Task Start: 5302 ms.
Arm Motor Set Angle: 16 us.
Pitch: -5 degrees.
```

Proof of Concept Verification



Summary and Lessons Learned

- I/O Latency does not factor into deadlines or WCET!
- Not advisable to use wood for fast moving parts
- TI driverlib provides helpful abstractions
- Hardware timers are harder to set up, but provide better resolution and are more reliable

Thank you!