MEMORANDUM

To: Charlie Refvem, Lecturer, Department of Mechanical Engineering, Cal Poly SLO
crefvem@calpoly.edu

From: Michael Shokoohi

Date: 10/23/2025

RE: **Mecha 02**

For this assignment I chose the simplist task structure possible which included collection_task running at a period of 50ms priority 2 and serial_task running at a period of 100ms priority 1. The fsm for collection_task includes states setup, idle and run while publish includes idle and publishing. The shares between tasks includes good_to_publish, start_test, collecting_data which essentially act as Boolean acknowledgements from one task to another reflecting what step they should move to. The collection has a faster period and priority because when a step response is being measured it is critical for that task to run in order to get the necessary data, while the publishing task is much lower priority because it simply empties the queues to the UART connection and doesn't allow for another test to be conducted until it is finished anyway. The queues used are 100 elements in length holding all data for time, right and left position, right and left velocity.

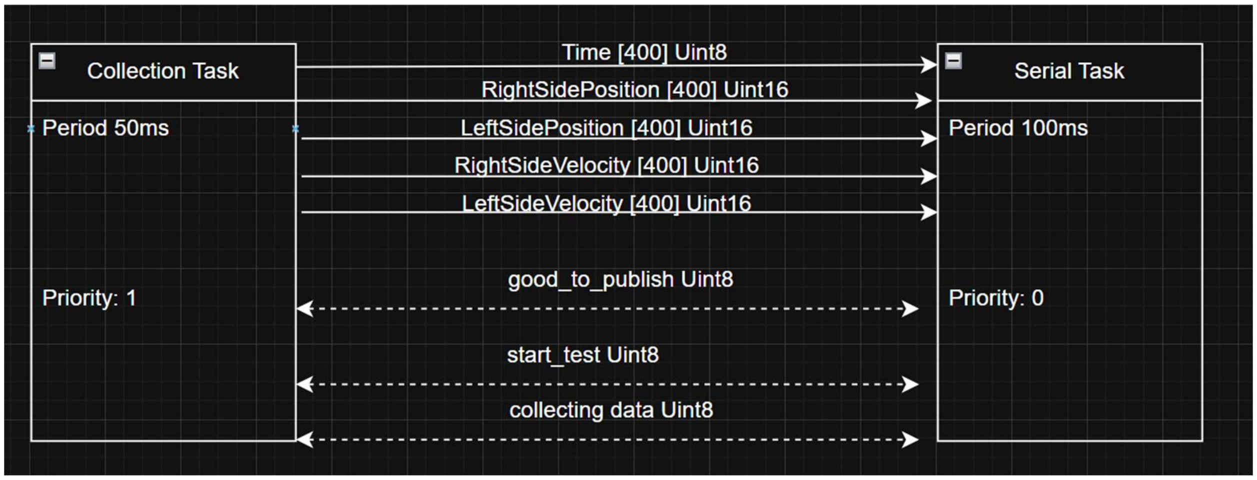


Figure 1. Shown is the Task diagram used for checking user input, collecting data, and publishing it to the serial connection.

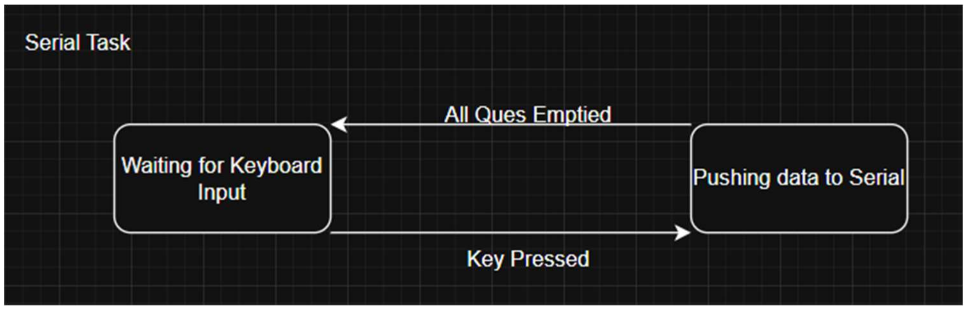


Figure 2. Shown are the states present within the serial task

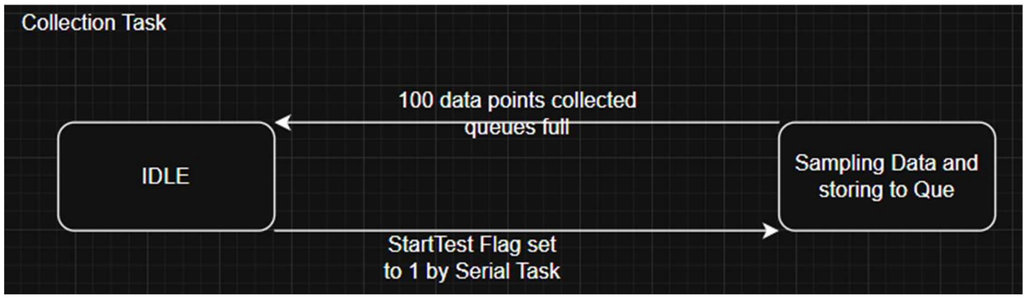
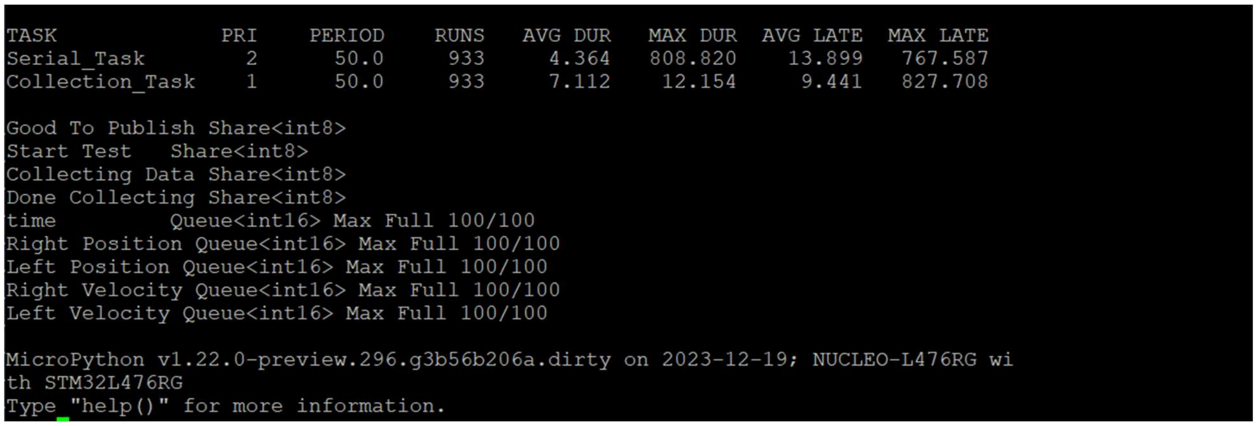

Collection Task
100 data points collected queues full
IDLE
Sampling Data and storing to Que
StartTest Flag set to 1 by Serial Task

Figure 3. Shown are the states present in the Collection Task.

Share name	Data type	Purpose
Good_to_publish	8 bit	Allows collection task to fill all queues before letting the serial task remove and process values from the queues
Start_test	8 bit	Allows the serial task to tell collection task to begin step response test (turns true when user input/ running automated script occurs)
Collecting_data	8 bit	This share is similar to good_to_publish but only allows the serial task to change into the publishing state not actually publish the data. This is important because its effect allows for the serial task to stop monitoring for user input.

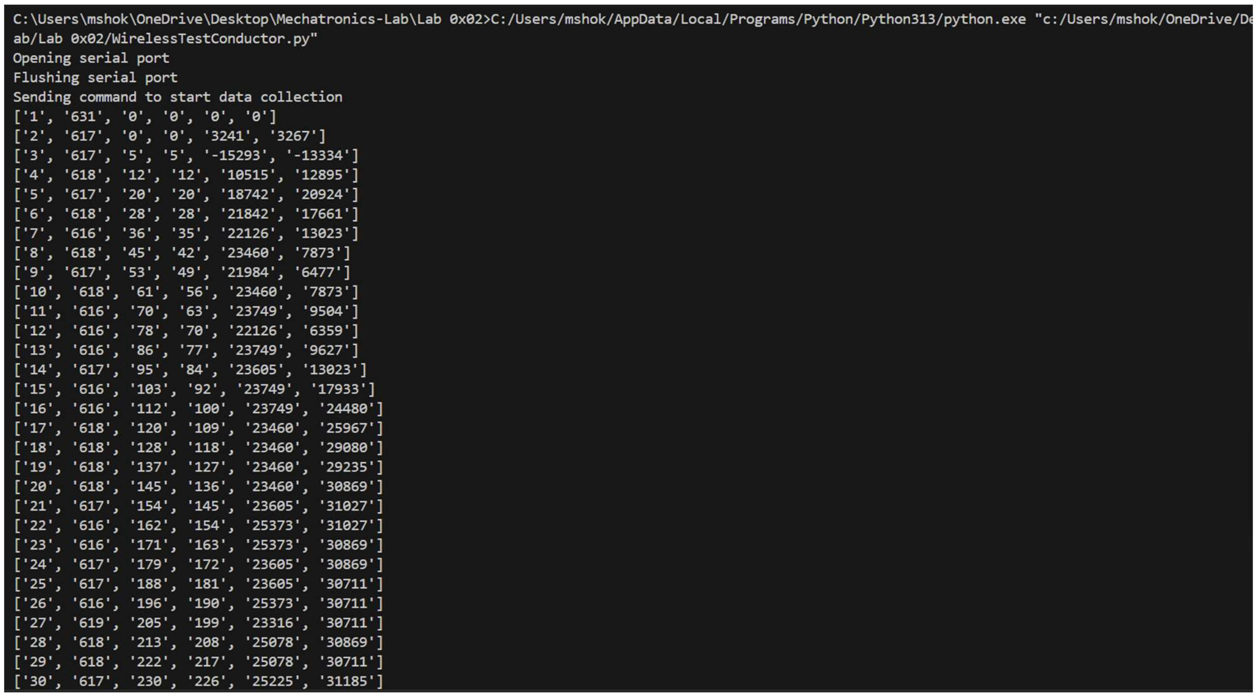


```
TASK             PRI    PERIOD    RUNS   AVG DUR   MAX DUR  AVG LATE  MAX LATE
Serial_Task        2      50.0     933     4.364   808.820    13.899   767.587
Collection_Task    1      50.0     933     7.112    12.154     9.441   827.708

Good To Publish Share<int8>
Start Test   Share<int8>
Collecting Data Share<int8>
Done Collecting Share<int8>
time         Queue<int16> Max Full 100/100
Right Position Queue<int16> Max Full 100/100
Left Position Queue<int16> Max Full 100/100
Right Velocity Queue<int16> Max Full 100/100
Left Velocity Queue<int16> Max Full 100/100

MicroPython v1.22.0-preview.296.g3b56b206a.dirty on 2023-12-19; NUCLEO-L476RG wi
th STM32L476RG
Type "help()" for more information.
```

Figure 4. Shown is the task profile printed to console from the scheduler. Not all of my tasks have run on time but I believe that to be an artifact of the initialization/configuration setup states for each task taking longer than expected. Overall from the user perspective it does provide the expected behavior.



```
C:\Users\mshok\OneDrive\Desktop\Mechatronics-Lab\Lab 0x02>C:/Users/mshok/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/mshok/OneDrive/De
ab/Lab 0x02/WirelessTestConductor.py"
Opening serial port
Flushing serial port
Sending command to start data collection
['1', '631', '0', '0', '0', '0']
['2', '617', '0', '0', '3241', '3267']
['3', '617', '5', '5', '-15293', '-13334']
['4', '618', '12', '12', '10515', '12895']
['5', '617', '20', '20', '18742', '20924']
['6', '618', '28', '28', '21842', '17661']
['7', '616', '36', '35', '22126', '13023']
['8', '618', '45', '42', '23460', '7873']
['9', '617', '53', '49', '21984', '6477']
['10', '618', '61', '56', '23460', '7873']
['11', '616', '70', '63', '23749', '9504']
['12', '616', '78', '70', '22126', '6359']
['13', '616', '86', '77', '23749', '9627']
['14', '617', '95', '84', '23605', '13023']
['15', '616', '103', '92', '23749', '17933']
['16', '616', '112', '100', '23749', '24480']
['17', '618', '120', '109', '23460', '25967']
['18', '618', '128', '118', '23460', '29080']
['19', '618', '137', '127', '23460', '29235']
['20', '618', '145', '136', '23460', '30869']
['21', '617', '154', '145', '23605', '31027']
['22', '616', '162', '154', '25373', '31027']
['23', '616', '171', '163', '25373', '30869']
['24', '617', '179', '172', '23605', '30869']
['25', '617', '188', '181', '23605', '30711']
['26', '616', '196', '190', '25373', '30711']
['27', '619', '205', '199', '23316', '30711']
['28', '618', '213', '208', '25078', '30869']
['29', '618', '222', '217', '25078', '30711']
['30', '617', '230', '226', '25225', '31185']
```

Figure 5. Shown is the data outputted from the Nucleo and received to the Bluetooth COM port on PC

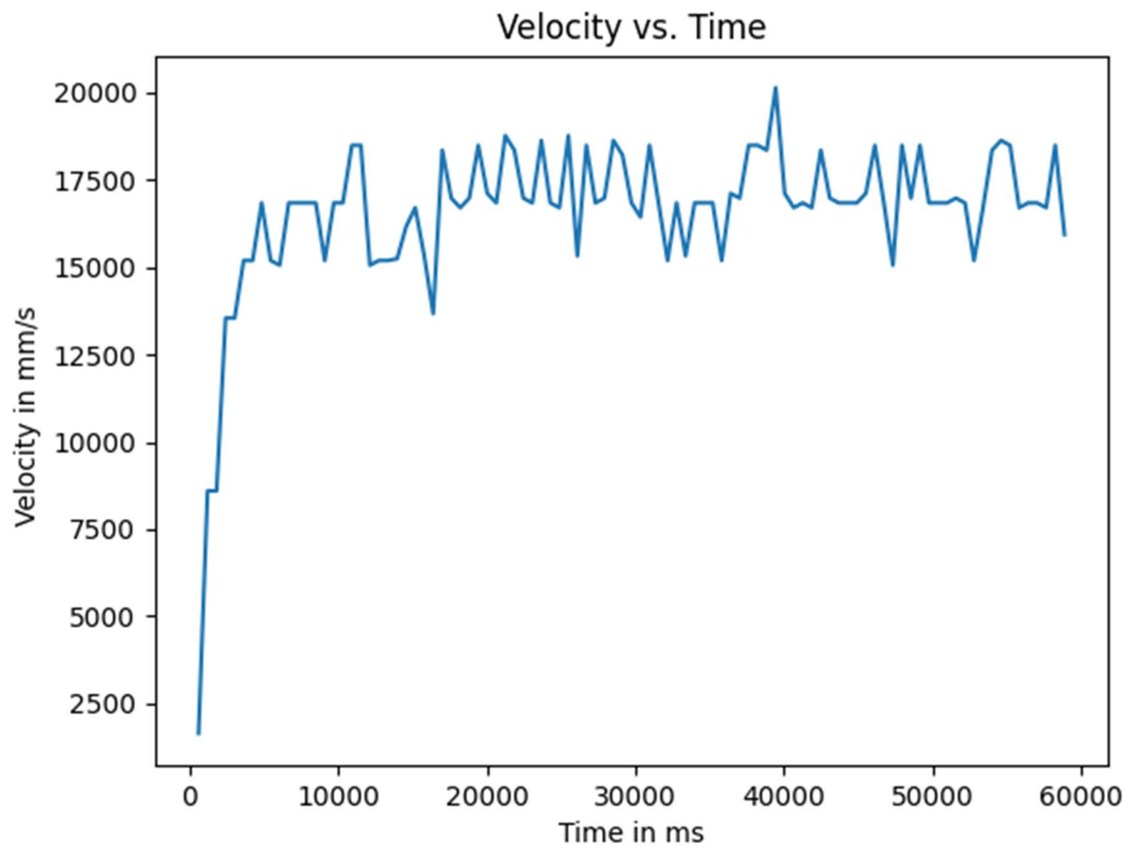


Figure 6. Shown is a plot of the data captured during a step response test

Source files used:

Motor.py, encoder.py (hardware drivers)
Main.py defines shares, queues, tasks and allows the scheduler to run
Tasks.py this is where the tasks themselves are defines along with their fsm
WirelessTestConductor.py (This is the automated PC script that runs tests and plots data)