ENGI1020 – Lab 4 Logbook

|  |  |  |
| --- | --- | --- |
| Name | Lab & Date | Input & Output Devices |
| Anton Guaman | Lab 4  November 8, 2020 | Input: Rotary Dial  Output: Servomotor |

* 0 Preparation
  + - Summary of lab task in your own words
    - Summary of preparation (no need to repeat entirely)

In the lab preparation I selected the Rotor Motor project. For the design part of the preparation we had to create a flow chart that from the rotary dial it will take 10 samples and average them in a certain amount of time using the sleep. Then through if and elif statements the average will be compared to a set value of the threshold of the rotary dial. This will manipulate the output in the rotary dial. For the testing part of the procedure the ways I thought that could disrupt it is by moving the rotary dial to the extremes so 0 and 1023.

For the lab we implemented our codes by parts. First, we created a for loop to calculate the average from the rotary dial and stored it in a variable so we can use it for the second part of our code. In the second part of our code we used if and elif statements to compare the average value calculated to a threshold value set of the rotary dial. The conditions were set to move the servo motor 5 degrees weather it is clockwise or counterclockwise depending on our if and elif statements. This value was also stored in a variable so we could use it on the servomotor command to move it to a certain angle. We also set the servomotor value at 90. The 5 degrees will be added or subtracted from that servomotor value depending on the if and elif statements. Finally, we combined these two parts into a while loop.

* 1 Implementation
  + 1.1 Converting Design to Implementation
    - Notes about how you took your design and implemented it
    - DO NOT just insert your code (though if it is useful to include lines of code to explain, you can)

As mentioned before our code is divided into two parts and then combined using a while loop. For the first part of the code we used a for loop to calculate the average of the rotary dial for ten samples and stored it in the variable named Avg. First, we create a variable named averagevalue that stores 0 which after the for loop will have the sum of the rotary dial values. Inside the for loop we create the variable readvalue that will store one value at a time from the input(analog\_read(1)). The input values will be taken every 0.1 seconds using the sleep function for the duration of the for loop which is 10.

readvalue = analog\_read(1)

Every 0.1 seconds the new input value stored in readvalue will be added to the averagevalue.

averagevalue += readvalue

After this averagevalue will have stored the sum of 10 values from the input. Then to calculate the average we will divide the averagevalue/10 and that value will be stored in the variable Avg.

Avg = averagevalue/10

For the second part of the code we will use if and elif statements to manipulate the servomotor. We begin by establishing an intial value for the servomotor in the variable SMsetting which stores 90. Also, we create a variable that stores the threshold value from the rotary dial in the variable RDthreshold = 500.

Now we create the if and elif statements. For the first one if Avg is <= RDthreshold the SMsetting value should increase by 5.

﻿ if Avg <= RDthreshold:

SMsetting += 5

Otherwise in the second statement if Avg is >= RDthreshold the SMsetting value decrease by 5.

elif Avg > RDthreshold:

SMsetting -= 5

Then we use the servomotor command to store the SMsetting value.

﻿servo\_move(5, SMsetting)

Finally, we combine these two parts into one by using the while loop. Every part mentioned has to be indented inside the while loop. But there are some exceptions that need to be placed outside and above the while loop. This will allow the code to run without errors.

﻿SMsetting = 90

RDthreshold = 500

﻿servo\_move(5,0)

* + 1.2. Errors Encountered While Implementing
    - Notes about errors encountered in Python interpreter or syntax problems
    - Include notes about solution
    - If useful, include names of errors

No errors encountered in the implementation. Although, I always say it that I always go through my code before testing verifying my variables are written correctly throughout the code.

* + 1.3 Details and Facts from Implementation to Remember
    - Details about concepts that you learned from implementation
    - Technical details about interpreter, Arduino equipment, engi1020 module related to implementation
    - Anything else you think will demonstrate your learning and will be useful to remember for future

Everything went well in the implementation. Something new that I learned and good to remember is the call command for the servomotor.

servo\_move( , )

* 2 Testing
  + 2.1. Test Plan and Results
    - Should be in table format!!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Input Manipulation** | **Output**  **Expected** | **Output**  **Observed** | **Investigation** |
| **Test 1** | **How did you**  **change the**  **input?**  I did not change the input I left the rotary dial at 0. | **What did you**  **expect to**  **observe?**  I expect to observe the servomotor to increase by 5. The first value will be 95. I expect this because the Avg will be < than the RDthreshold. Causing the output to increase by 5. I expect it to be 95 since the SMsetting is 90 plus 5 it will be 95. | **What did you**  **observe?**  I observed what I expected. The Avg was less than the RDthreshold so the servomotor increased by 5 and the first value was 95. | **Did they match?**  **If not, what did**  **you do (can**  **reference**  **Section 2.2)**  Yes, they did match! |
| **Test 2** | I moved the rotary dial to its max which would be 1023. | I expect to observe the servomotor to decrease by 5. The first value will be 85. I expect this because the Avg will be > than the RDthreshold. Causing the output to decrease by 5. I expect it to be 85 since the SMsetting is 90 minus 5 it will be 85. | I observed what I expected. The Avg was greater than the RDthreshold so the servomotor decreased by 5 and the first value was 85. | Yes, they did match! |
| **Test 3** | I was moving the rotary dial so I could test that the if and elif statements were working. The values that the rotary dial should pass is greater than 500 and less than 500 which is the RDthreshold value. | When moving the rotary dial above 500 or equal to 500 I expect the servomotor to decrease by 5. And when moving the rotary dial less than 500 I expect the servomotor to increase by 5. | When I moved the rotary dial above 500 the servomotor to decrease by 5. When I moved the rotary dial below 500 the servomotor increased by 5. | Yes, they did match! |

* 2.2. Errors Encountered while Testing
  + - Details about semantics or syntax errors discovered through testing
    - Expanding on “Investigation Column” above

No errors encountered while testing. Although we implemented something else. While testing when the servo motor value reached 0 or 180 the value will continue to decrease or increase past the limits of the rotary dial. To prevent this an if and elif statement was implmeneted.

﻿if SMsetting >= 180:

SMsetting = 180

elif SMsetting <= 0:

SMsetting = 0

If the SMsetting >= 180 the SMsetting will become 180 preventing the value of the SMsetting to go beyond the servomotor limits. The same thing will happen if SMsetting <= 0 the SMsetting will become 0 preventing the value of the SMsetting to go beyond the servomotor limits.

* + 2.3. Details and Facts from Testing to Remember
    - Details about concepts that you learned from testing
    - Technical details about interpreter, Arduino equipment, engi1020 module related to testing
    - Anything else you think will demonstrate your learning and will be useful to remember for future

Everything went well in the testing. Something new that I learned and good to remember is the call command for the servomotor.

servo\_move( , )

Also, I learned how to print these types of problems into the console. This was super helpful while doing the tests since I could see through the consoles the values of the servomotor and the average values of the rotary dial.

﻿print("AvgRead: {0}, RDsetting: {1}".format(Avg, SMsetting))

* 3. Reflection and Conclusion
  + 3.1. Reflection Question Answers
    - While other parts of the lab logbook should be in jot-note form, reflection question answers should be complete sentences.

1. Describe another situation where having a loop inside a loop would be useful.

For instance, when you need to do multiple loops that are connected with each other. Also, it can include if, elif, and else statements. All of these parts of code need to be executed more than once. An outer loop will be greater to execute more than once the inner loop or loops with if, elif and else statements that have a relationship.

1. Describe a situation where having a loop inside a loop would NOT be useful.

When the programmer or user wants to execute only once a loop an outer loop will not be useful.

1. As you tested your individual parts, what did you find challenging?

The first part of the code I did not have any problems testing, it worked fine. On the other hand, the second part of the code brought difficulties since the second part needed to be connected to the first part in order to work. This was challenging at first but then when I figured it out the testing went well. I knew that for testing part 2 I needed to connect it to part 1.

1. When you tested everything together, what changes did you need to make?

The testing of everything together went well. I just added the servo\_move(5,0) outside the while loop since the servomotor wasn’t moving when testing it. After this everything worked fine.

* + 3.2 Additional conclusion Notes