**Week 4 Lab 2 Report**

**Name: Anurag S. Chatterjee**

**Lab Group: B03**

**Task 1:**

Using the formula: :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **At Rest** | **Pushed Up** | **Pushed Down** | **Pushed Left** | **Pushed Right** |
| **VRx** | 353= **1.14V** | 353= **1.14V** | 353= **1.14V** | 8= **0.026V** | 704= **2.27V** |
| **VRy** | 353= **1.14V** | 8= **0.026V** | 704= **2.27V** | 353= **1.14V** | 353= **1.14V** |

**Task 2:**

The I2C address is 0x48, so I substituted that value in my code given. The 16-bit Config register values sent in the first version of the program is used to control the operating mode, input selection, data rate, full-scale range, and comparator modes. Based on the MSB numbers in the ADS1115 datasheet pages 28-29 in the 14:12 Bit section and in the MUX[2:0] field, change the input multiplexer configuration to change our reading from analog-in pin A0 to analog-in pin A1. If we want to read from analog-pin A0, change MSB to Wire.write(0b11000000) and to read from analog-pin A1, change MSB to Wire.write(0b11010000). If we change these, the joystick will behave accordingly, so if we read from A0 i.e., Wire.write(0b11000000) //MSB Of Config Register, the VRx will work (joystick left and right from 0 to 3.32 volts – 0V pulled to left and 3.32V pulled to right) and if we read from A1 i.e. Wire.write(0b11010000) //MSB Of Config Register, the VRy will work (joystick up and down from 0 to 3.32 volts – 0V pulled to up and 3.32V pulled to down).

**Task 3:**

|  |  |  |  |
| --- | --- | --- | --- |
| Setting Up The Voltages In The X And Y Directions:    I have created VRx and VRy voltages and voltageread() is a function I made which will read and output the voltage from the ADS1115\_MUX\_Channel. I will then feed ADS1115\_COMP\_0\_GND to VRx as it is connected to A0 and A1 for VRy. I initialized Pushbutton as D3 as input earlier in my code and  BUTTONSTATE reads the status of the button. The outputs function takes in the VRx, VRy and Button inputs to read the final outputs. | |  |  | | --- | --- | | If-Else Statement:    The outputs function is the if-else statement under void loop() which will read the status of the joystick being pushed up, down, left, right, up-left, up-right, down-left and down-right based on the output voltages of VRx and VRy based on what I’ve observed in task 1, as well as prints out the state of the button too as pushed or not pushed at all. |  | |

**Task 4:**

**A picture containing text

Description automatically generated**

I initialized ledState as a Boolean variable to be LOW as bool ledState = LOW while filling up my ssid and password names. Then, I created a new function named toggleLED to handle the new endpoint server.on("/toggleLED", toggleLED) which I called upon later in the void setup (void) block where I also initialized the LED to be in D4 pin as pinMode(D4, OUTPUT). Everything else is the same as the WebServer.ino code. In my function, the ledState != ledState will convert the ledStatus to HIGH so if the LED\_STATUS (a new variable created) is LOW, then turn the LED On (as ledState != ledState so LOW initially will become HIGH now), and print this out as String(LED\_STATUS) to be “LED On!”, or else, turn the LED Off and print this out.

**Task 5:**

A picture containing graphical user interface

Description automatically generated

The sensor I used was a simple PushButton which can be used for doorbells for example to ring the doorbell. I initialised a variable called pushBUTTON which will read the current state of the button – on whether it was pushed or not. I have connected this button to pin D3 as digitalRead(D3). The code is almost the same as the HTTP Request code with the only difference being that I initialized 4 url that would send back data to the laptop on whether the button is pushed or not. The data I sent the current status of the button as pushed (represented by 1) or not pushed (represented by 0), as well as a string to read the current status of the button – if pushed, it will print out “Button Is Pushed” or else, “Button Is Not Pushed”.

|  |  |
| --- | --- |
|  |  |