**EGR 302 – Engineering Design and Documentation**

**Deliverable 6: Detailed Design**

Team Name: \_\_\_\_\_\_\_\_\_\_\_\_RPMs\_\_\_\_\_\_\_\_\_

Team Members: \_\_\_\_\_\_\_\_Jordan Ziegler\_\_\_\_\_\_

\_\_\_\_\_\_\_\_Timothy Decious\_\_\_\_

\_\_\_\_\_\_\_\_Brian Ungermann\_\_\_\_

\_\_\_\_\_\_\_\_Austin Hulen\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_Dylan Shanahan\_\_\_\_\_

\_\_\_\_\_\_\_\_Josh Hemsley\_\_\_\_\_\_\_

Client’s Name: \_\_\_\_\_\_\_\_Dr. Xu\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Version: \_\_\_\_\_\_\_\_\_Version 2\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_April 4, 2014\_\_\_\_\_\_\_\_\_

6.1 Mechanical Detailed Design.

Below in Figure 6.1 is the mechanical design are all the moving parts that make up the structure of our device.

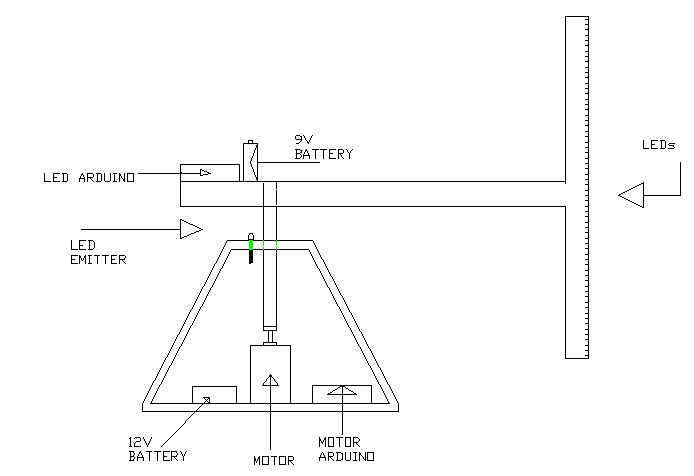


Figure 6.1: complete mechanical design

The mechanical diagram shows the overall view and the dimensioning of the device. The lower portion includes the base of the device which has a 12V battery, a motor, the motor arduino, and the infrared emitter LED. The upper portion includes the rotating arm and the led wand. The wand is the vertical component and the rotating arm is the horizontal portion of the mechanical design. In the design, the arduino and the battery on the rotating arm are designed to counterbalance with the wand.

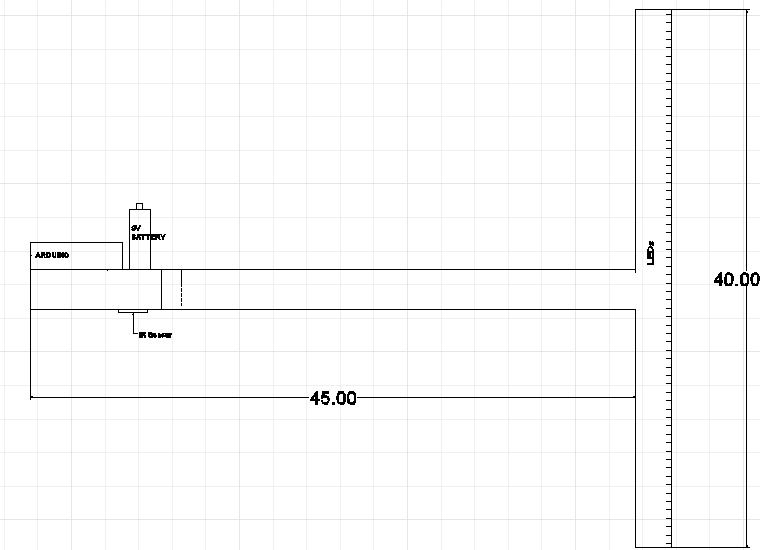


Figure 6.2: Overall Arm Design

The figure above in figure 6.2 demonstrates the dimensions of the rotating arm and the wand filled with LEDs. The next two portions will go into greater detail on each of these.

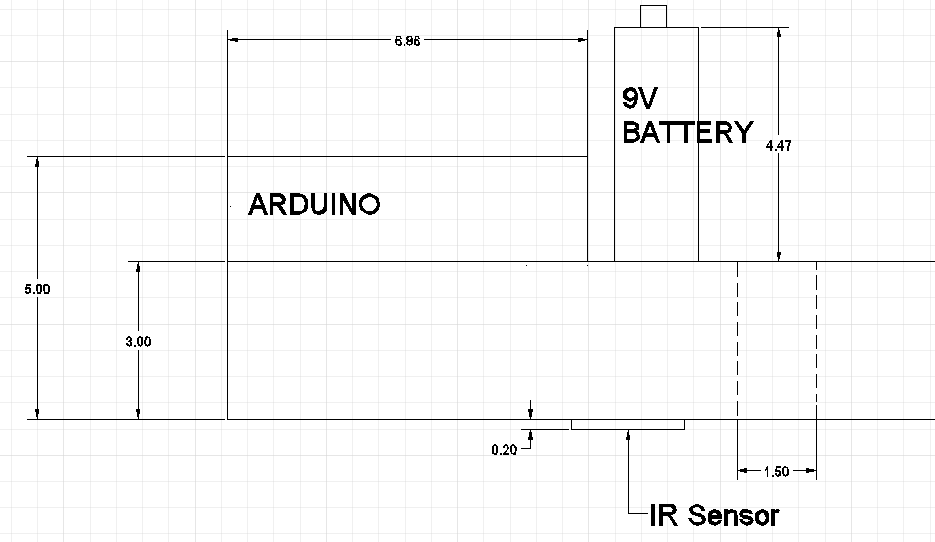


Figure 6.3: Section of LED Wand

This section shows the arduino and the 9V battery in greater detail along with the dimensions of the hole for the rotating shaft to be inserted.

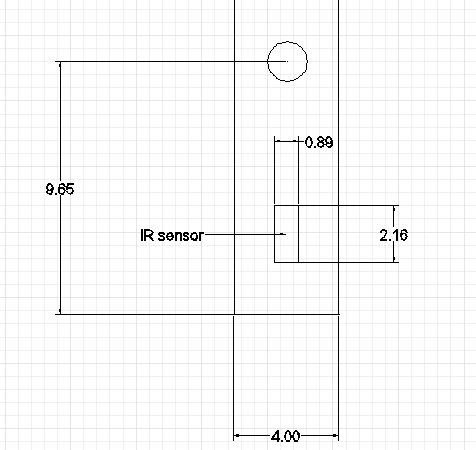


Figure 6.4 Dimensions of arm from below.

This portion is a view from beneath the rotating arm. It includes parts of the dimensioning not included in the previous sections.

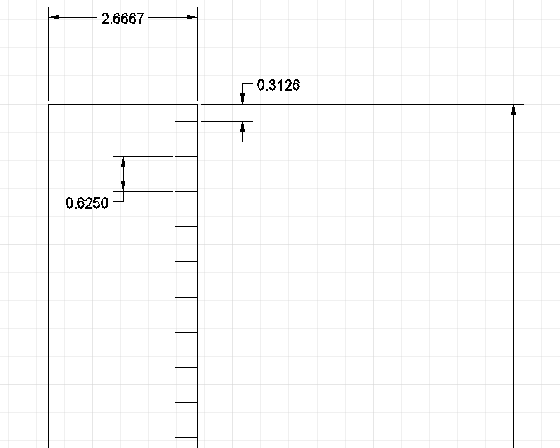


Figure 6.5: Close up of the End of the Arm

In this part, the small horizontal lines symbolize the LEDs. There are a total of 64 spread evenly across the wand.

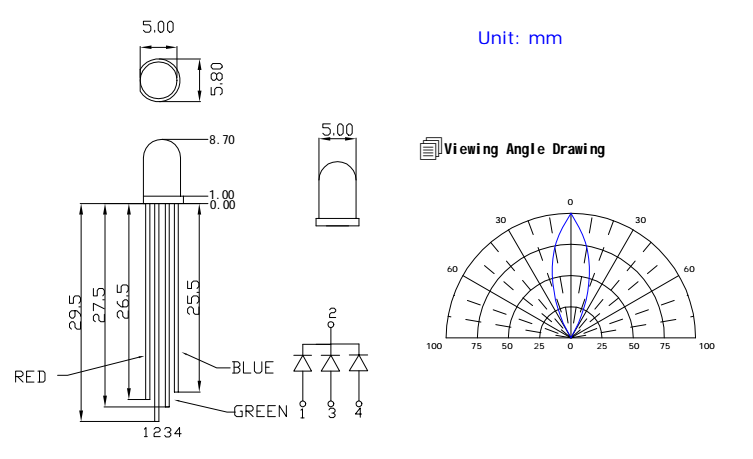


Figure 6.7 LED design from datasheet

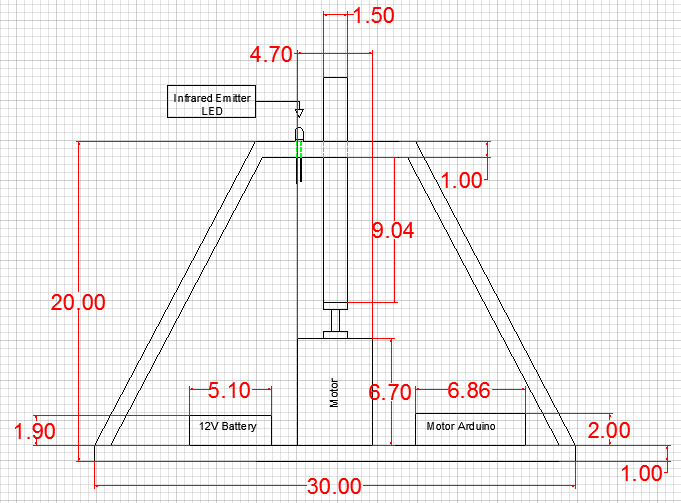


Figure 6.8: Overall Base Design

The above figure shows the dimensioning for the base of the E-Wand. This includes the motor, the motor arduino, the battery, and the infrared emitter.

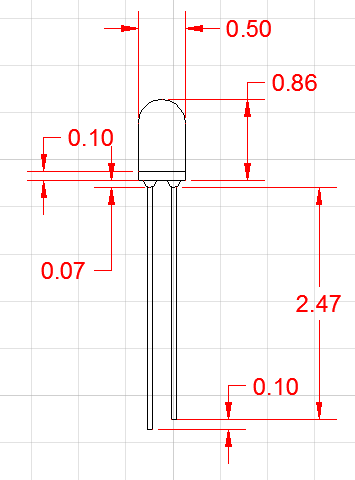


Figure 6.9: Infrared LED Emitter Design

The above figure shows the dimensioning for the LED Emitter.

6.2 Electrical Detailed Design

Below in Figure 2 is the LED electrical design. The design is somewhat limited for simplicity showing only the circuitry for one color. The circuitry for the other two colors is identical to this one with the exception of using different pin locations. The resistors are 330 ohm resistors and are included in order to keep the LEDs from being overpowered.

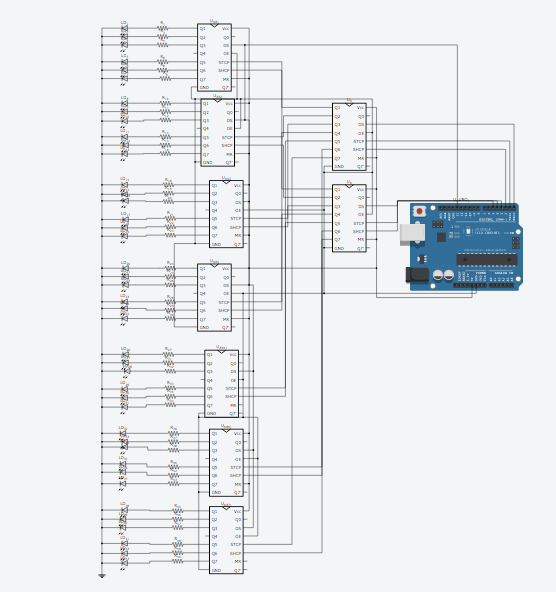


Figure 2: LED Circuit Design

Two shift registers are connected to the arduino pins for control. These shift registers select the LEDs that are driven through 7 other shift registers. A zoomed in picture is shown below in Figure 3 to give a better picture of the shift register connections. Vcc and MR is connected to 5V, DS is connected to pin 13.

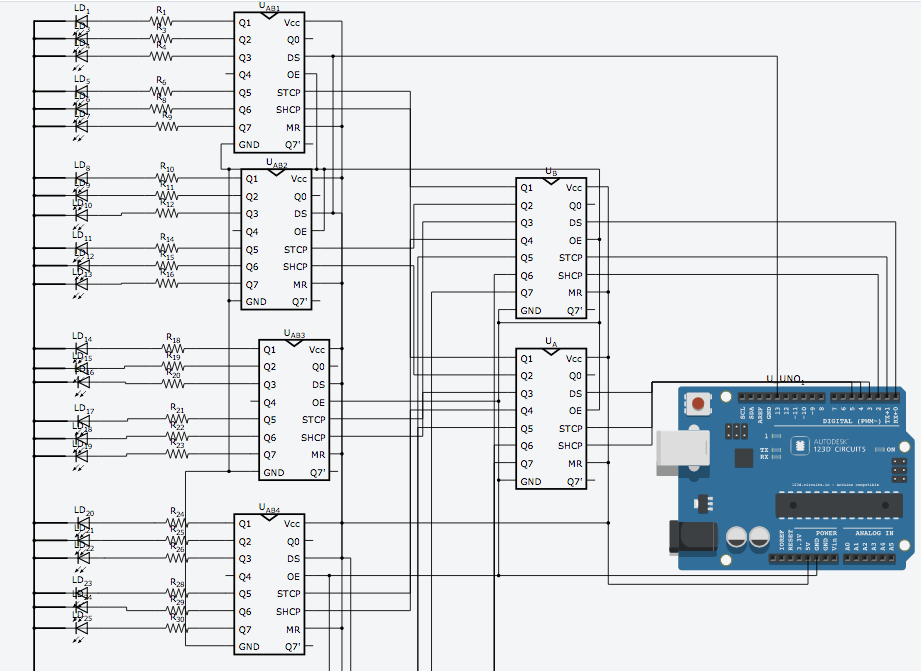


Figure 3: LED Shift Registers

Each of the LEDs is controlled by the bank of 7 shift registers. These shift registers are controlled by the two primary shift registers which are connected to pins 1-6. Pins 1, 2 and 3 are connected to the top shift register, while Pins 4, 5 and 6 are connected to the lower shift register. Figure 5 below shows the truth table for the LEDs. Shown are the LEDs and the pins that control them. Each LED is listed with the pinouts required to turn it on. Only one LED can be turned on at a single time.

Shown below in Figure 4 is the portion of the LED arduino that controls the IR sensor.

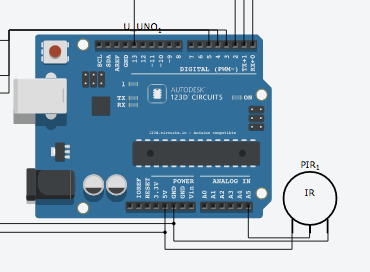


Figure 4: IR sensor Connected to LED Arduino

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pin 1 | Pin 2 | Pin 3 | Pin 4 | Pin 5 | Pin 6 | Pin 13 |
| LED1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| LED2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| LED3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| LED5 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| LED6 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| LED7 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| LED8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| LED9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| LED10 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| LED12 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| LED13 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| LED14 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| LED15 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| LED16 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| LED17 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| LED19 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| LED20 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| LED21 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| LED22 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| LED23 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| LED24 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| LED26 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| LED27 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| LED28 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| LED29 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| LED30 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| LED31 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| LED33 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| LED34 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| LED35 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| LED36 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| LED37 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| LED38 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| LED40 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| LED41 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| LED42 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| LED43 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| LED44 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| LED45 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| LED47 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| LED48 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| LED49 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 5: LED truth table

Below in Figure 6 is shown the motor arduino design. The motor is shown as a block and connected to Pin 0. The emitting diode is shown as a diode and connected to Pin 1.

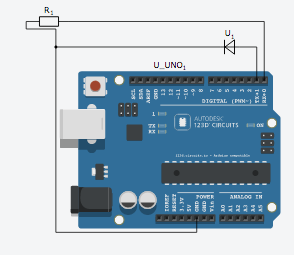


Figure 6: Motor Arduino Design

6.3 Software Design

Below in Figure 3 is shown the overall flow diagram of Arduino 1 which the program follows. Each block represents a function call whose parameters are shown below in the tables following Figure 3.

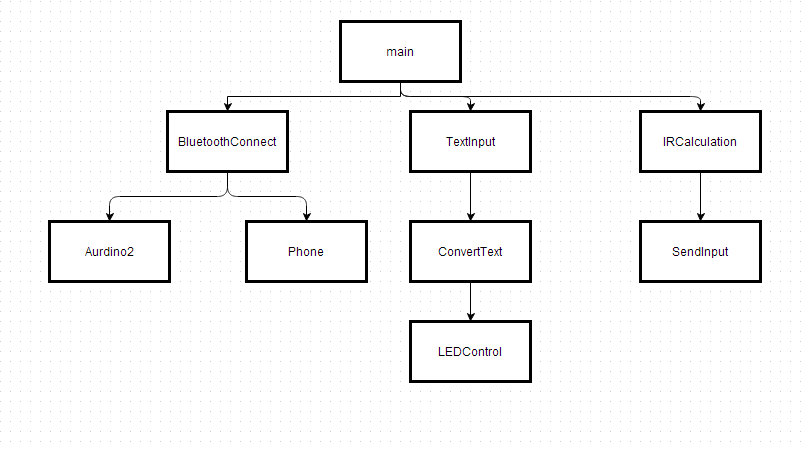


Figure 3: Software Diagram of Arduino 1.

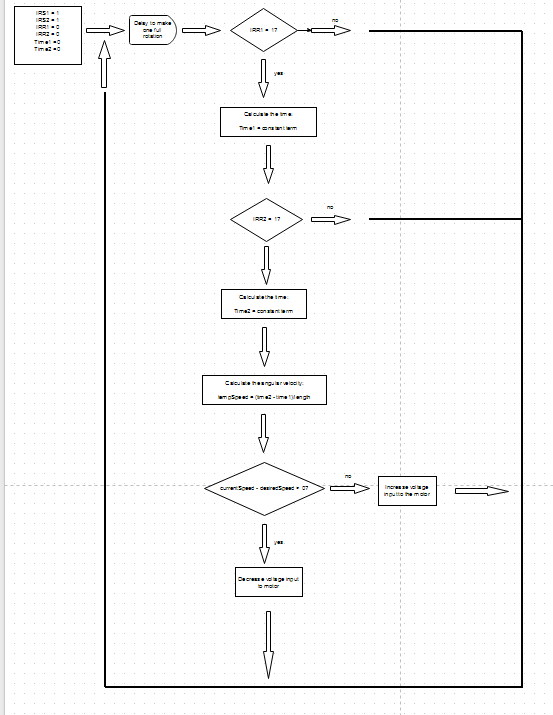


Figure 4: Flow Diagram

The flow diagram in Figure 4 shows how the software reacts to situations and locations and controls the LEDs. In figure 5 below the software for the first Arduino is shown to calculate the speed of the motor and determine the position in order to light up LEDs.

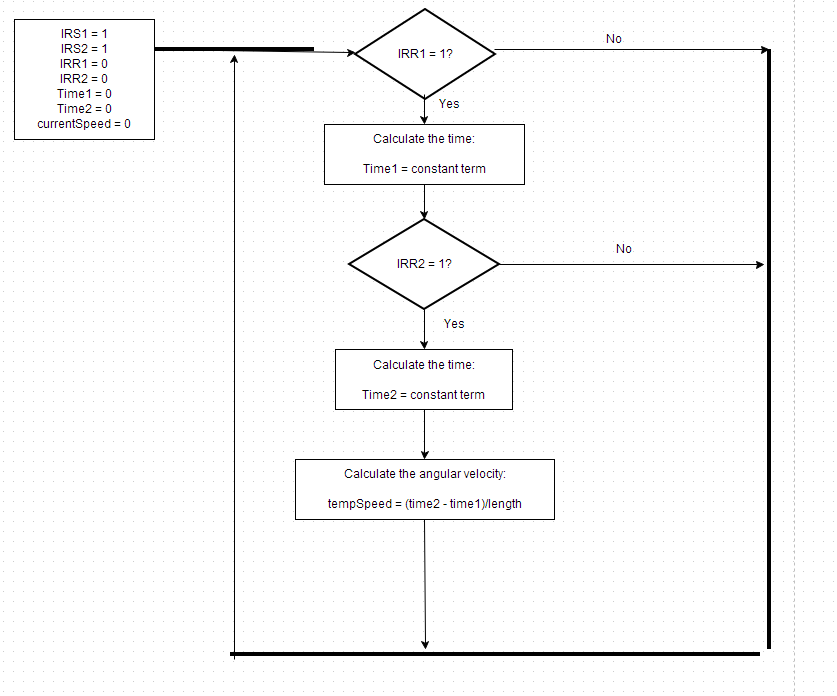


Figure 5: LED arduino flow diagram

Figure 6 below shows the control system for the motor arduino.

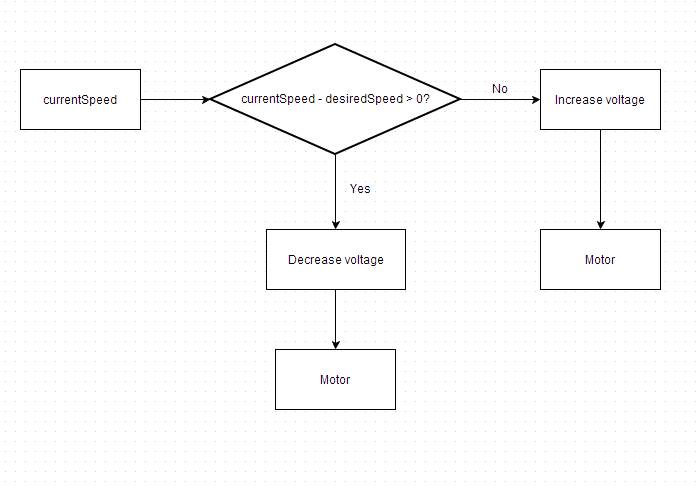


Figure 6.

|  |  |
| --- | --- |
| Module Name | main() |
| Module Type | Coordination |
| Input Arguments | None |
| Output Arguments | None |
| Descriptions | The main function calls BluetoothConnect() to conncet to Arduino2 and to the users phone. TextInput() is then called to receive the text the user wants to display and then converts the text to be displayed onto the LED wand. IRCalculation() is called to calculate the speed of the wand and sends the data to Arduino2 to adjust the motor speed. |
| Modules Invoked | BluetoothConnect, TextInput, IRCalculation |

Table 6.1: Main()

The main function in Table 6.1 gets a user input of text over Bluetooth and maps it out to the LED wand, and controls the motor speed.

|  |  |
| --- | --- |
| Module Name | BluetoothConnect() |
| Module Type | Input and Output |
| Input Arguments | None. |
| Output Arguments | None. |
| Descriptions | This function calls Arduino2() and Phone() to create a bluetooth connection between the Arduino’s and the user’s phone. |
| Modules Invoked | Arduino2, Phone |

Table 6.2: BluetoothConnect()

The BluetoothConnect() module shown in Table 6.2 calls Arduino2() and Phone() in order to establish a bluetooth connection with the other Arduino and the user’s phone.

|  |  |
| --- | --- |
| Module Name | Arduino2() |
| Module Type | Input and Output |
| Input Arguments | Bluetooth address to the Arduino2 is inputted |
| Output Arguments | Bluetooth Connection |
| Descriptions | Creates a connection with Arduino2 with its address and this connection is then used to send the speed of the motor to Arduino2 |
| Modules Invoked | None. |

Table 6.3: Arduino2()

The Arduino2() Module in Table 6.3 creates a connection between the Arduino 1 and 2 so they can communicate and send information back and forth.

|  |  |
| --- | --- |
| Module Name | Phone() |
| Module Type | Input and Output |
| Input Arguments | Bluetooth to phone |
| Output Arguments | Bluetooth Connection |
| Descriptions | Creates a bluetooth connection to the user’s phone which allows the user to then input a text and send it to the Arduino to display on the LED wand. |
| Modules Invoked | None. |

Table 6.4: Phone()

The Phone() module shown in Table 6.4 creates a connection between the Arduino 1 and the user’s phone so the user can send the message they want displayed on the LED wand.

|  |  |
| --- | --- |
| Module Name | TextInput() |
| Module Type | Input and Output |
| Input Arguments | userText-text from the user that was sent over bluetooth |
| Output Arguments | userText |
| Descriptions | Receives the text from the user and sends this text to ConvertText() to have the text mapped to the LED |
| Modules Invoked | ConvertText |

Table 6.5: TextInput()

The TextInput() Module shown in Table 6.5 receives the text from the user over bluetooth and sends it to the ConvertText() module.

|  |  |
| --- | --- |
| Module Name | ConvertText() |
| Module Type | Input and Output |
| Input Arguments | userText |
| Output Arguments | mappedLEDs |
| Descriptions | Receives the inputted text from the user and maps the text to the LED’s |
| Modules Invoked | LEDControl |

Table 6.6: ConvertText()

The ConvertText() function module shown in Table 6.6 receives the user text and maps it to the LED’s.

|  |  |
| --- | --- |
| Module Name | LEDControl() |
| Module Type | Input and Output |
| Input Arguments | mappedLEDs |
| Output Arguments | None. |
| Descriptions | Takes the mapped LEDs and displays them on 64 x110 pixels and decides what LED’s need to be lit up at what time as the LED wand spins. |
| Modules Invoked | None. |

Table 6.7: LEDControl()

The LEDContorl module shown in Table 6.7 uses the mapped LED’s and sends them to be displayed on the LED wand.

|  |  |
| --- | --- |
| Module Name | IRCalculation() |
| Module Type | Input and Output |
| Input Arguments | IRSensor |
| Output Arguments | IRReadings |
| Descriptions | This function records the IR sensor and uses it to calculate the angular velocity of the arm and sends this information to SendInput() |
| Modules Invoked | SendInput |

Table 6.8: IRCalculation()

The IRCalculation() Module shown in Table 6.8 receives data from IR sensor and calculates the angular velocity of the motor.

|  |  |
| --- | --- |
| Module Name | SendInput() |
| Module Type | Input and Output |
| Input Arguments | IRReadings |
| Output Arguments | Speed |
| Descriptions | This sends the new speed that the motor needs to operate at to Arduino2 through the bluetooth connection |
| Modules Invoked | None |

Table 6.9: SendInput()

The SendInput() module takes the calculated speed the motor needs to be moving and sends it to the Arduino 2 through bluetooth.