**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 1\_\_\_\_\_\_\_\_\_\_

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

6.1 Mechanical Detailed Design.

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

Insert Image HERE

Figure 6.1: 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.

6.2 Electrical Detailed Design

Below in Figure 6.2 is the electrical design. The circuitry supports 16 LEDs using shift registers and an arduino. The infrared sensor is also shown. The resistors are 330 ohm resistors and keep the LEDs from being overpowered. Each LED is RGB and takes an input controlling each color separately.

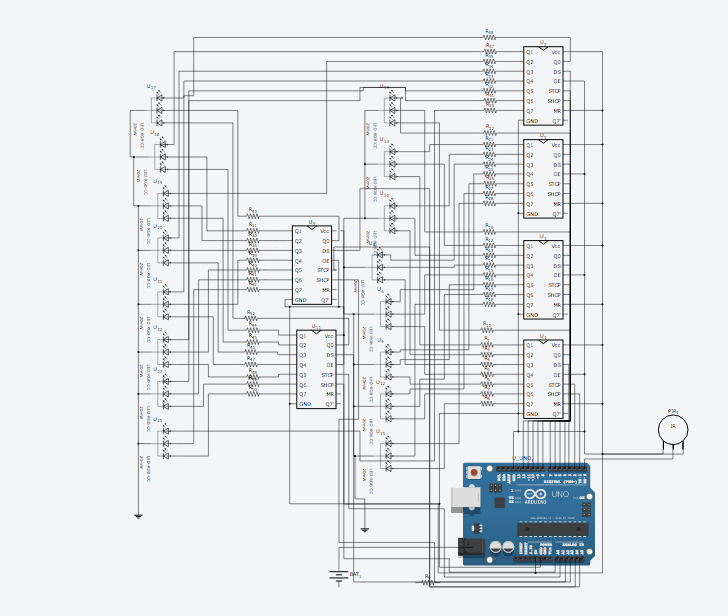


Figure 6.2: Electrical Design

6.3 Software Design

Below in Figure 6.3 is shown the overall flow diagram which the program follows. Each block represents a function call whose parameters are shown below in the tables following figure 3.

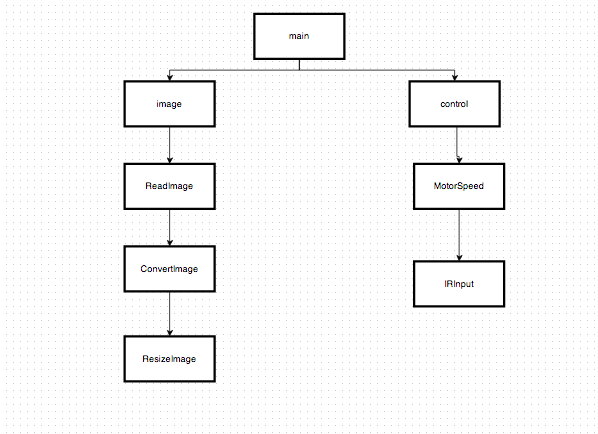


Figure 6.3: Software Flow Diagram

|  |  |
| --- | --- |
| Module Name | main() |
| Module Type | Coordination |
| Input Arguments | None |
| Output Arguments | None |
| Descriptions | The main function calls image() to read the input file from Bluetooth or USB and resize the image and convert it to the proper format, and control() to control the motor speed. |
| Modules Invoked | image, control |

Table 6.1: Main()

The main function in Table 6.1 calls image() to read the input file from Bluetooth or USB and resize the image and convert it to the proper format, and control() to control the motor speed.

|  |  |
| --- | --- |
| Module Name | image() |
| Module Type | Input and Output |
| Input Arguments | None. |
| Output Arguments | Image to LED’s |
| Descriptions | The image function calls ReadImage() |
| Modules Invoked | ReadImage |

Table 6.2: Image()

The image() module shown in Table 6.2 calls ReadImage() in order to accept an image from the user.

|  |  |
| --- | --- |
| Module Name | ReadImage() |
| Module Type | Input and Output |
| Input Arguments | Image File |
| Output Arguments | Image |
| Descriptions | Receives any image from user and then calls ConvertImage |
| Modules Invoked | ConvertImage |

Table 6.3: ReadImage()

The ReadImage() Module in Table 6.3 Receives any image from user and then calls ConvertImage to convert the image into a jpeg format.

|  |  |
| --- | --- |
| Module Name | ConvertImage() |
| Module Type | Input and Output |
| Input Arguments | Image |
| Output Arguments | Converted Image |
| Descriptions | Receives image from user and converts the image to jpeg format then sends the Converted Image to ResizeImage |
| Modules Invoked | ResizeImage |

Table 6.4: ConvertImage()

The ConvertImage() module shown in Table 6.4 receives image from user and converts the image to jpeg format then sends the Converted Image to ResizeImage.

|  |  |
| --- | --- |
| Module Name | ResizeImage() |
| Module Type | Input and Output |
| Input Arguments | Converted Image |
| Output Arguments | Resized Image |
| Descriptions | Takes the Converted Image and resizes the image to 64 x110 pixels and returns it to the Image function |
| Modules Invoked | None. |

Table 6.5: ResizeImage()

The ResizeImage Module shown in Table 6.5 takes the Converted Image and resizes the image to 64 x110 pixels and returns it to the Image function.

|  |  |
| --- | --- |
| Module Name | Control() |
| Module Type | Input and Output |
| Input Arguments | None. |
| Output Arguments | None |
| Descriptions | The control function calls the MotorSpeed function to determine the speed of the motor |
| Modules Invoked | MotorSpeed |

Table 6.6: Control()

The control function module shown in Table 6.6 calls the MotorSpeed function to determine the speed of the motor.

|  |  |
| --- | --- |
| Module Name | MotorSpeed() |
| Module Type | Input and Output |
| Input Arguments | None. |
| Output Arguments | Speed |
| Descriptions | The MotorSpeed Function calls the IRSensor Function to determine the speed to of the motor. |
| Modules Invoked | IRInput |

Table 6.7: MotorSpeed()

The MotorSpeed module shown in Table 6.7 calls the IRSensor Function to determine the speed to of the motor.

|  |  |
| --- | --- |
| Module Name | IRInput() |
| Module Type | Input and Output |
| Input Arguments | IR sensor |
| Output Arguments | Position |
| Descriptions | Recieves data from IR sensor and calculates the speed |
| Modules Invoked | None. |

Table 6.8: IRSensor()

The IRInput Module shown in Table 6.8 recieves data from IR sensor and calculates the angular velocity of the motor.

6.9 Algorithm flowchart