# Individual Lab Report 4

Guillermo Manuel Cidre

Team G – Bob's builders

Teammates: Eric Newhall, Michael O’Connor, Christian Heaney-Secord

IRL04

2/26/15

Individual Work:

For this week, I implemented the control of two stepper motors, a DC motor, and the electromagnet for the addition of the part placer subsystem. Originaly, our GUI program could only transmit two values from two sliders on the window. The sliders were implemented in order to control motors. So for this week, I had to expand upon the GUI by the addition of a third slider for determining which direction the DC motor spins and a button that toggles the electromagnet on or off. Such addition vastly changes our state diagram for out input serial handler for the arduino. The new diagram can be shown on Figure 1. The new layout of the GUI can be shown in Figure 2. The arduino code for this system demo is on Figure 3

In addition, I implemented very basic serial communication between the raspberry pi and the arduino. Basically, the arduino connected to the raspberry pi through its usb port. So, the arduino communicated with the raspberry pi the same way the arduino communicated with the GUI program. The raspberry pi, then, initiated serial connection through python. The code for this part is on Figure 4.

Challenges / Issues:

The main challenge that I had was troubleshooting everything once the project was set up. Initially, the goal was to leave specific pins untouched in the arduino. So, I interfere with other pins like pin 0 and pin 1. The issue was that those pins were preprogramed to handle serial components in the machine. So they failed to write the proper values in our program. Another factor that contributed to our troubleshooting was the fact that our power supply only provided up to half an Amp instead of 4 amps of current. This caused components to not work correctly if more than one is running at the same time.  These two factors took a while to fix. But it gave two possible problems. The first problem is that we needed to get a power source. The second problem is that a bigger arduino will be needed soon for the other motors that we need to hook up since we had to use some of the specific pins that were supposed to be untouched. These problems will hopefully be solved as soon as we receive the new pieces.

Team Work:

Michael O'Connor and Christian Heaney-Secord worked on assembling the arm of the part placer. In addition, they helped develop the CAD model of our design. They also helped in the attempt of applying a new power source to the project in order to replace the one as specified in the challenges section.

Eric Newhal helped me properly wire the two stepper motors, the DC motor, and the electromagnet into the protoboard. In addition, he helped improved the part placer after it was assembled. He also helped in the attempt of applying the new power source.

Future Plans:

We have access to the new power source but we do not know how to operate it yet. So, I plan to help in fixing part of the power source. In addition, I plan to expand my GUI again for the implementation of the flux subsystem. Finally, I plan to help on the midterm presentation for our group.

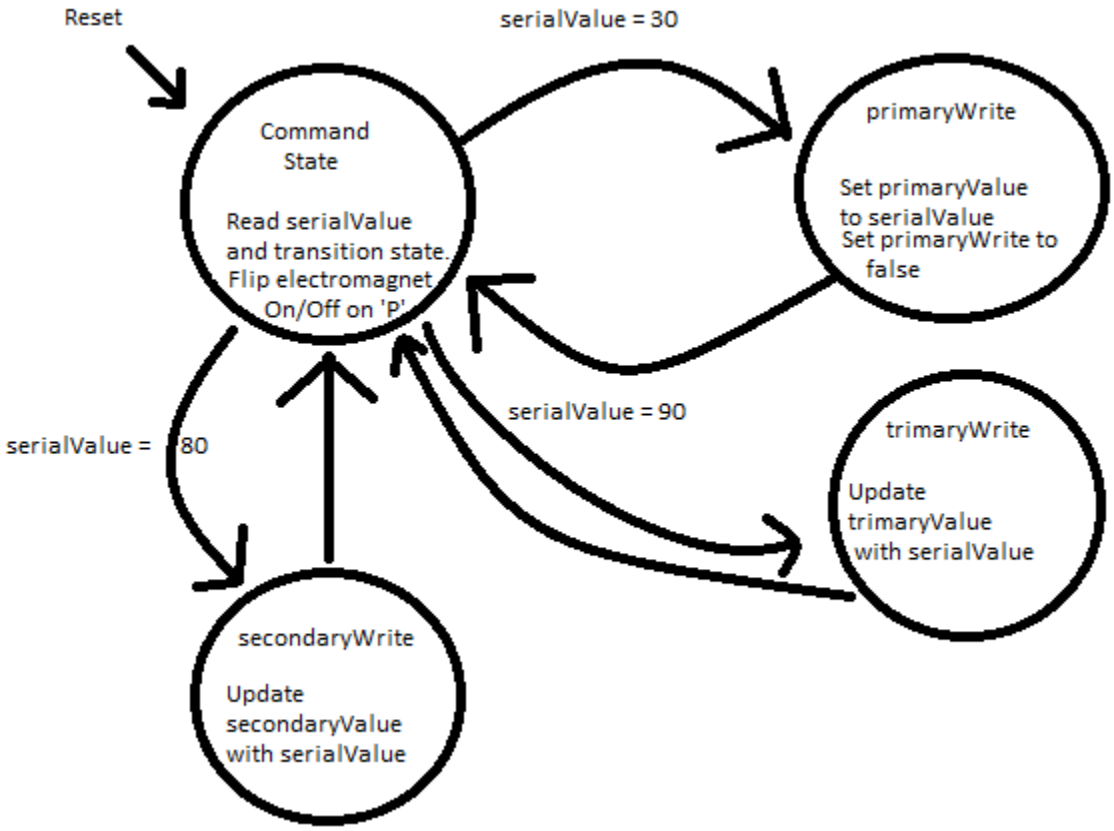


Figure 1: input handling state diagram

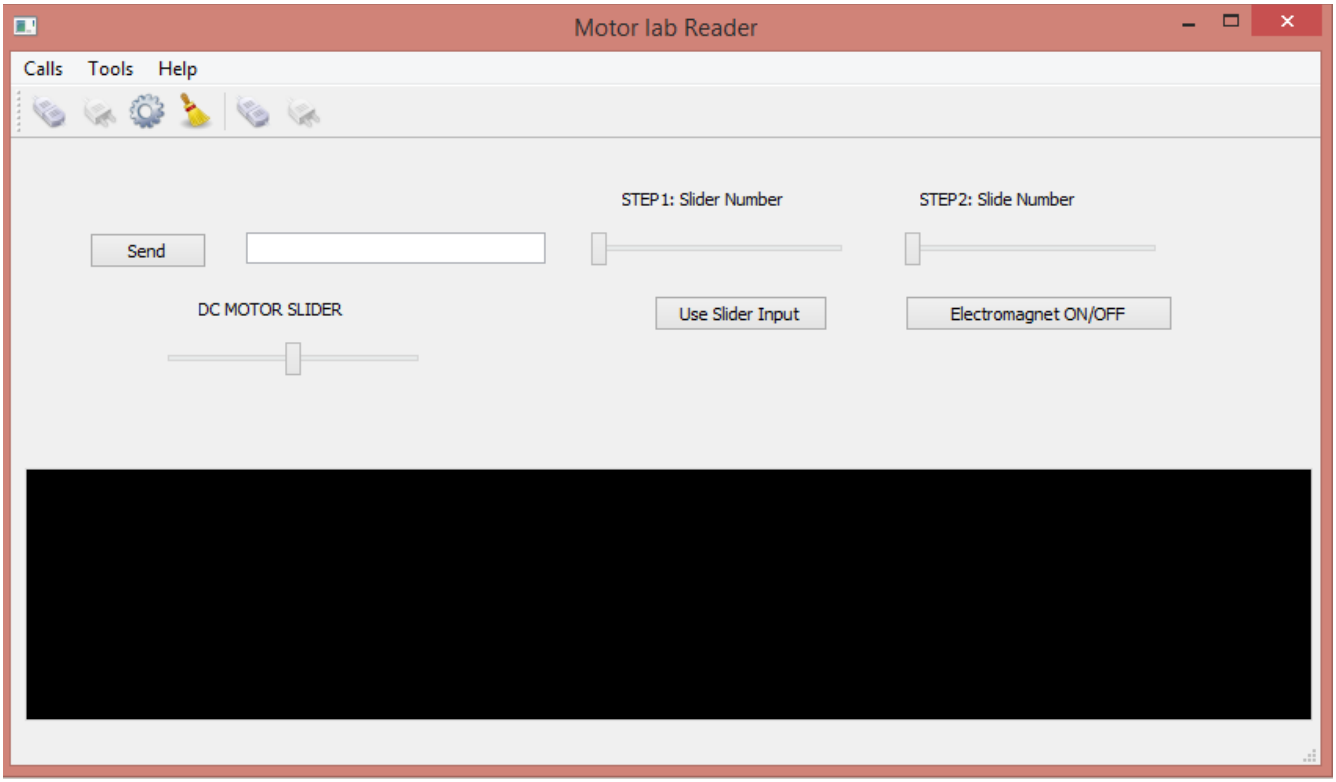


Figure 2: Visual representation of the new expanded GUI program

```
float IRDistance = 0.0;
int stepPin = 6;
int dirPin = 5;
int enPin = 9;
int toggle = 0;
float stepDistance = 0.0;

//Second Motor need to access pin values
float IRDistance2 = 0.0;
int stepPin2 = 4;
int dirPin2 = 10;
int enPin2 = 12;
int toggle2 = 0;
float stepDistance2 = 0.0;

//For our DC motor
float potAngle = 0.0;
//int DCMotorAngle = 0;
int DCMotorPinA = 7;
int DCMotorPinB = 8;
```

```
//etc
const int limitsw = 13;
const int elePin = 2;
int ele = 0;




unsigned char serialValue = 0;
float primaryValue = 0; //for GUI
boolean primaryWrite = false;

float secondaryValue = 0;
boolean secondaryWrite = false;

unsigned char trimaryValue = 1;
boolean trimaryWrite = false;

void setup() {
   Serial.begin(9600);
   while (!Serial) { //Copy
    ; // wait for serial port to connect. Needed for Leonardo only
  }
   pinMode(stepPin,OUTPUT);
   pinMode(dirPin,OUTPUT);
   pinMode(enPin,OUTPUT);

   pinMode(stepPin2,OUTPUT);
   pinMode(dirPin2,OUTPUT);
   pinMode(enPin2,OUTPUT);

   pinMode(limitsw, INPUT);
   pinMode(elePin, OUTPUT);

   pinMode(DCMotorPinA, OUTPUT);
   pinMode(DCMotorPinB, OUTPUT);
}

void loop() {
  if(Serial.available()) //Read value and decide what to do with it
```

```
{
  serialValue = Serial.read();
  if(primaryWrite)
  {
    primaryValue = serialValue;
    primaryWrite = false;
  }
  else if(secondaryWrite)
  {
    secondaryValue = serialValue;
    secondaryWrite = false;
  }
  else if(trimaryWrite)
  {
    trimaryValue = serialValue;
    trimaryWrite = false;
  }
  else if(serialValue == 90)
  {
    trimaryWrite = true;
  }
  else if(serialValue == 30)
  {
    primaryWrite = true; //primaryValue = next serial reading
  }
  else if(serialValue == 80)
  {
    secondaryWrite = true;
  }
  else if(serialValue == 50)
  {
    ele = (!ele) & 0x01;
  }
} //attributed sensor readings

//Stepper Motor controlled by IR sensor
IRDistance = (primaryValue*2.5)/(255*2.5); //approx distance in cm
IRDistance2 = (secondaryValue*2.5)/(255*2.5);

//Motor 1
if(stepDistance < IRDistance-.005 && IRDistance < 2.0){
   digitalWrite(enPin, LOW);
```

```
    digitalWrite(dirPin, HIGH);
    toggle = (!toggle) & 0x01;
    digitalWrite(stepPin, toggle);
    stepDistance += .001;
  }else if(stepDistance > IRDistance+.005 && IRDistance < 2.0){
    digitalWrite(enPin, LOW);
    digitalWrite(dirPin, LOW);
    toggle = (!toggle) & 0x01;
    digitalWrite(stepPin, toggle);
    stepDistance -= .001;
  }
  else {
    digitalWrite(enPin, HIGH);
  }

  //Motor 2
  if(stepDistance2 < IRDistance2-.005 && IRDistance2 < 2.0){
    digitalWrite(enPin2, LOW);
    digitalWrite(dirPin2, HIGH);
    toggle2 = (!toggle2) & 0x01;
    digitalWrite(stepPin2, toggle2);
    stepDistance2 += .001;
  }else if(stepDistance2 > IRDistance2+.005 && IRDistance2 < 2.0){
    digitalWrite(enPin2, LOW);
    digitalWrite(dirPin2, LOW);
    toggle2 = (!toggle2) & 0x01;
    digitalWrite(stepPin2, toggle2);
    stepDistance2 -= .001;
  }
  else {
    digitalWrite(enPin2, HIGH);
    digitalWrite(dirPin2, LOW);
  }

  //DC Motor Implementation(TO DO)
 // if(digitalRead(limitsw) == LOW)
  //{
    switch(trimaryValue) {
      case 0:
        //turn left
        digitalWrite(DCMotorPinA,LOW);
        digitalWrite(DCMotorPinB,HIGH);
```

```
        break;
      case 1:
        digitalWrite(DCMotorPinA,LOW);
        digitalWrite(DCMotorPinB,LOW);
        break;
      case 2:
        digitalWrite(DCMotorPinA,HIGH);
        digitalWrite(DCMotorPinB,LOW);

    }
  //}
//  else
//  {
//        digitalWrite(DCMotorPinA,LOW);
//        digitalWrite(DCMotorPinB,LOW);
//  }

//  if(potAngle < DCMotorAngle - 8.0){
//    digitalWrite(DCMotorPinA,LOW);
//    digitalWrite(DCMotorPinB,HIGH);
//  }else if(potAngle > DCMotorAngle + 8.0){
//    digitalWrite(DCMotorPinA,HIGH);
//    digitalWrite(DCMotorPinB,LOW);
//  }else{
//    digitalWrite(DCMotorPinA,LOW);
//    digitalWrite(DCMotorPinB,LOW);
//  }


   //electro magnetism
   digitalWrite(elePin, ele);

  delay(1); //For fast speed


  Serial.flush();
}
```

Figure 3: Arduino code for system demo

```
const int ledPin = 13;
```

```
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available()) {
    switch(Serial.read()){
      case '0':
        digitalWrite(ledPin, LOW);
        break;
      case '1':
        digitalWrite(ledPin, HIGH);
    }
  }
  Serial.println("Hello World");
  delay(2000);
}
```

Figure 4: Arduino code for raspberry pi interface.