Individual Lab Report 3

Guillermo Manuel Cidre

Team G – Bob's builders

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

IRL03

2/19/14

Individual Work:

For this week, I mainly focused on modifying our last lab code to get the stepper motor working for our system demo in the lab. The stepper motor which we were using for our demo is different than the stepper motor used in the lab itself. So Eric and I had to figure out the proper wiring for it and properly apply it to our previous lab. All other sensors and motors were carefully removed from the board because they were no longer needed. In addition, I proofread the code and fixed series of simple bugs that began to appear in the previous lab. The code for this lab is on Figure 2. The state diagram for the input serial handling of the code is on figure 1.

Challenges / Issues:

The main challenge that I had was troubleshooting everything once the project was set up. When we first tested the new stepper motor with our code, the stepper motor failed to rotate and instead nudge clockwise then counterclockwise. So, we basically spend an hour believing that our wiring for our new motor was incorrect. After testing our all possible connections and thinking more clearly, we found that the structure of our design was flawed, or that the hole that the motor was supposed to peek out of collided sharply with one edge of the motor shaft. This was easily fixed by loosing some of the screws attached to motor.

An issue I experience was the fact that my GUI program can only run on my computer. It turned out that my executeable for the program requires to be linked with up to 10 DLL files that were installed along with QT programming. My other teammates don't have this files and therefore cannot run it. This did not impact us this week. But when working with more than one subsystem at once, we would have to rely upon more than one computer.

Team Work:

Eric Newhal helped me properly wire the new motor into the protoboard. In addition, he helped set up the sequence diagrams correctly. I mainly only pointed out some mistakes while he fixed them. Then, he helped loosened some of the screws in order to fixe the issue introduced in our main challenge.

Michael O'Connor and Christian Heaney-Secord worked on creating the tray, railing system, and some hoppers for prototypes. They mainly achieved it through the use of a 3D printer. They also attached the different components together to make the structure of our project that we are gonna demo.

Future Plans:

From the second issue of my work this week, I plan to make the GUI program standalone so that more computers can be utilized for different subsystems. I also plan to fix typos from our design proposal that was copied unto the website. These things may possibly not be needed right now, but it is better to get them done early. I also plan to contribute in testing the electromagnetism and helping set up part of the upper subsystem hopefully for the next demo.

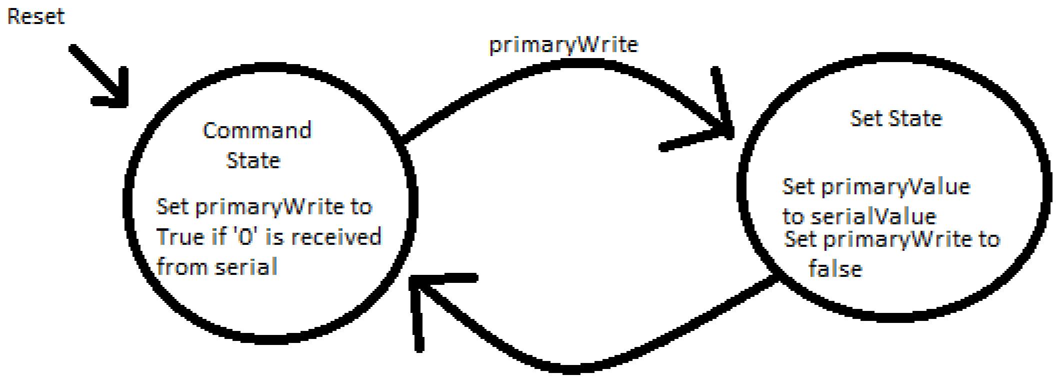

Figure 1: input handling state diagram

```
int IRPin = A0;
float IRDistance = 0.0;
int stepPin = 6;
int dirPin = 5;
int enPin = 2;
int toggle = 0;
float stepDistance = 0.0;

unsigned char serialValue = 0;
float primaryValue = 0; //for GUI
boolean primaryWrite = 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);
}
```

```
void loop() {
  if(Serial.available()) //Read value and decide what to do with it
  {
    serialValue = Serial.read();
    if(primaryWrite)
    {
      primaryValue = serialValue;
      primaryValue = serialValue;
      primaryWrite = false;
    }
    else if(serialValue == 30)
    {
      primaryWrite = true; //primaryValue = next serial reading
    }
  } //attributed sensor readings

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

  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);
  }
  delay(1); //For fast speed


  Serial.flush();
}
```

Figure 2: Code for system demo