Wesley Myers
18-578
Team F
Individual Lab Report
February 22, 2012

Individual Progress

I created two diagrams to assist us with this Lab. The first was what the finite state machine should look like. This is illustrated in Figure 1. We had discussed what we wanted, but I visualized it so that we could start by the weekend. The next thing I did was create a diagram for a lab demonstration rig. I gave this to Rick and he took it from there. This is illustrated in Figure 2.

This week my main focus was making progress on the CMUcam4. The personal goal I set for myself was being able to locate a target for the demonstration. I was making progress with the code I was developing when I was contacted by a CMUcam4 developer. As it turns out, he is writing an Arduino CMUcam4 library. This library is under development, so there are still minor issues with it. Either way I was a suitable candidate for testing and using it.

I essentially transferred to this new library resource. It has many of the CMUcam4 functions prototyped, but not all of them are working. The library made the communication between the Arduino and CMUcam4 very simple. I did have to download a Google developer Arduino application that is different from the standard Arduino application. This version unfortunately uses a different file extension, so old code needed to be transferred over to this new application.

Another thing that I found out was that the camera on the CMUcam4 was damaged. It took some very strange pictures that I assumed were due to some noise since what I had was a development board. I consulted with the CMUcam4 developers and I found out that the cell phone camera on the board was defective. I promptly went and got a new camera for the CMUcam4. After testing, I found that there was still some minor noise due to the development board. After bargaining, I acquired a new board that is getting currently made by Sparkfun and a company in Europe. This new board has no noise and operates effectively. A sample of the new image taken with the CMUcam4 is shown in Figure 8.

To help me with testing, I created a test bench with some spare cardboard and some plastic. Curtis Layton consulted me and he said that the background for the targets would be black. Curtis also gave me the tape used on the actual targets so that I can make my own small-scale versions. I spray painted the cardboard black and cut out targets. Figure 3 illustrates the board with the targets.

In order to make sure I have a consistent angle with the CMUcam4, I made a stationary structure to support it. This enables a consistent angle every time the camera is operated. Figure 7 illustrates the rigging.

I assisted in final integration with the demonstration rig. The rig is illustrated in Figure 4. The final system is illustrated in Figure 5.

I created an LED array to visually illustrate the state of the machine. This was helpful for testing and demonstrations. Since the Arduino UNO only has one Serial port for communicating to the CMUcam4 and to the laptop, any print statements also go to the CMUcam4, which can cause problems. The LEDs allowed me to see errors and results. This is shown in the code section of the ILR. For demonstrations, it gives a good visual cue to people observing the system. The LED array is illustrated in Figure 6.

Challenges

Consistent lighting for the targets has been a challenge. I tested in many different locations to see different lighting situations. It is a bit of a painful process because I must take an image with the CMUcam4 and then open it up on my computer to check the RGB values. After that, I must program it to the camera and test it. Often the ranges for the Red, Green, and Blue must be tweaked. I decided that putting the target under the lab bench lamp provided consistent light during day and night time conditions. This is illustrated in Figure 3.

Another challenge was that Richard got sick Sunday night. This pushed responsibility for our Monday meeting on the rest of the team. Emergency communication was needed to determine what could and couldn't be done by the mid-semester project demonstration. Because he was sick, he wasn't able to laser cut the acrylic until Tuesday. A lot of the assembly was pushed to Tuesday night.

I think another large challenge the team faced was writing a rather large amount of code and having multiple people working on it. There were some tense moments between some members of my team. Luckily I was able keep away from that by working on the CMUcam4 code. I only touched FSM code when I added in the camera functionality.

Teamwork

This week we split up the work that suited our strengths. SooHyun and David were tasked with getting the Finite State Machine working. Most of what needed to be done was already written. It was a matter of integrating code together into a FSM format.

Rick was tasked with getting the demonstration rig assembled. He was able to use parts meant for the final product in our demonstration rig. It gave us an idea

of how everything worked in relation to mounting and weight. Doing this opened our eyes to the fact that just these few components cause a lot of weight and causes sagging in the acrylic board that supports them.

My task was to work on the CMUcam4. This is really a one-person job, however David assisted for an hour over the weekend so that he could learn more about it.

At the end we came together to assemble everything for our final demo.

Future Work

The work with the CMUcam4 has made great strides. This code shows that we can distinguish and locate the targets. This will be used to help build the final system.

As mentioned earlier, our demonstration rig helped us understand the physical system better. The things we have learned from this will assist us in creating our final system. In fact we have already made modifications in the design from what we have learned.

This lab was of great use since it helped us understand how everything works together. Not only from a physical point of view, but also a coding point of view.

As for next week, I am again tasked with the CMUcam4. I was asked to be able to distinguish between all three colors. Rick is tasked with getting the mechanical parts assembled. SooHyun and David are tasked with writing new FSM code for our system.

Figures

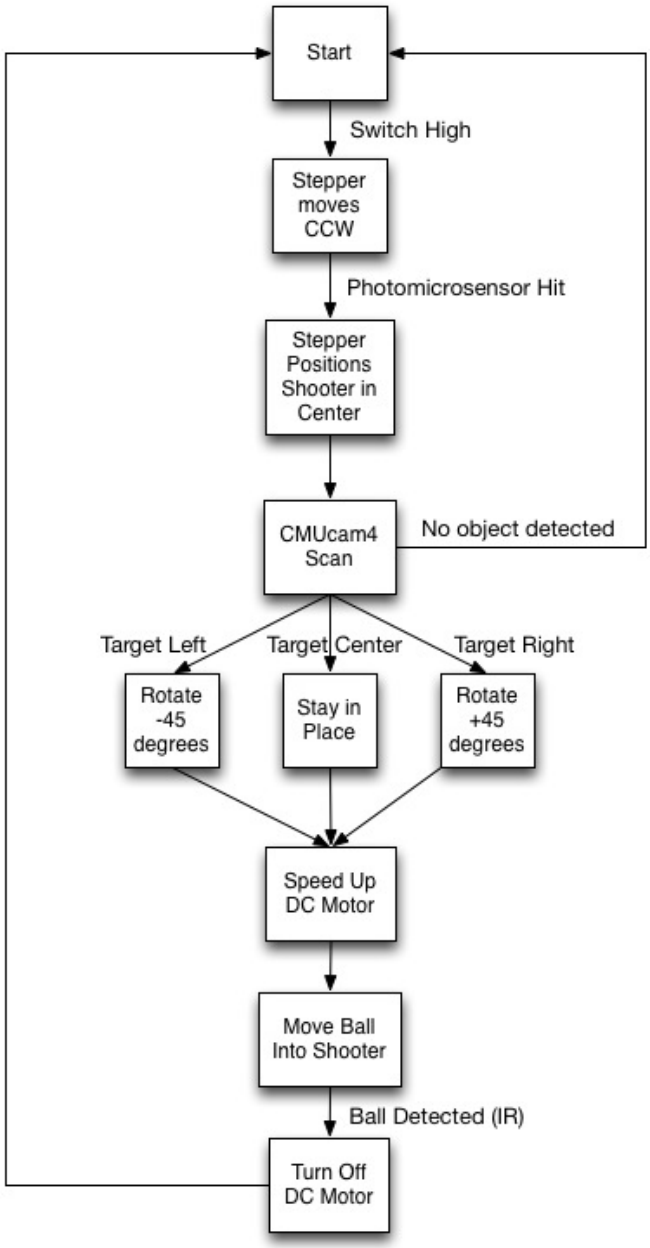

Figure 1. State Machine for Lab

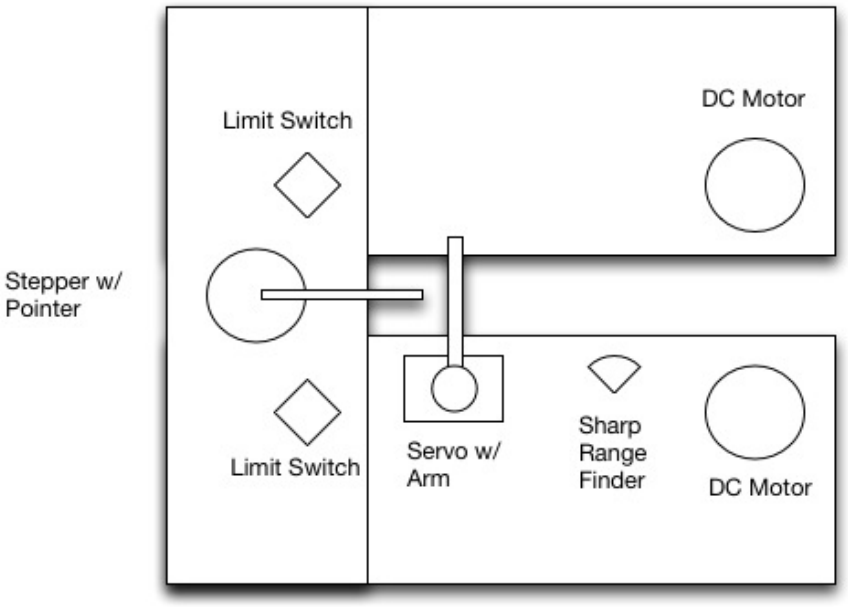

Figure 2. Demonstration Rig for Lab

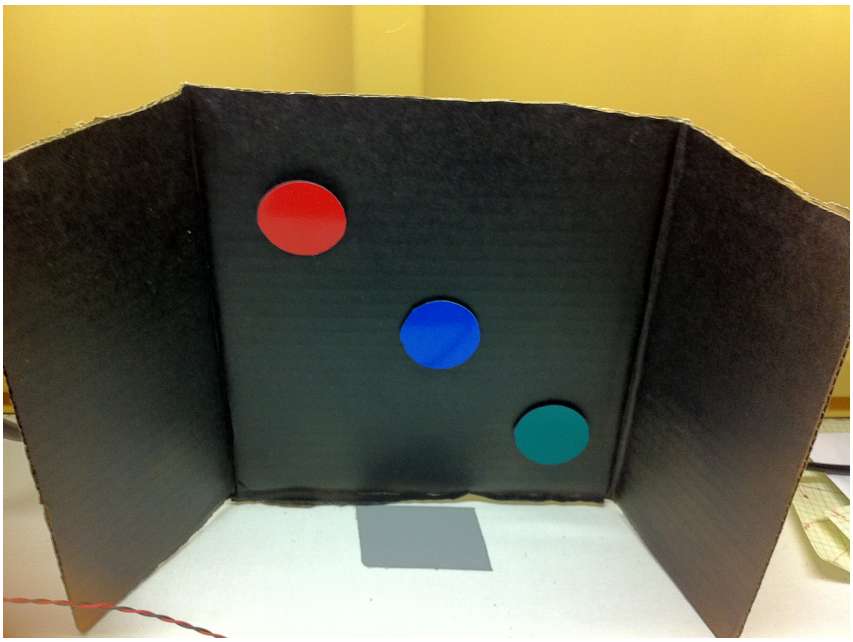

Figure 3. Targeting Test Bench

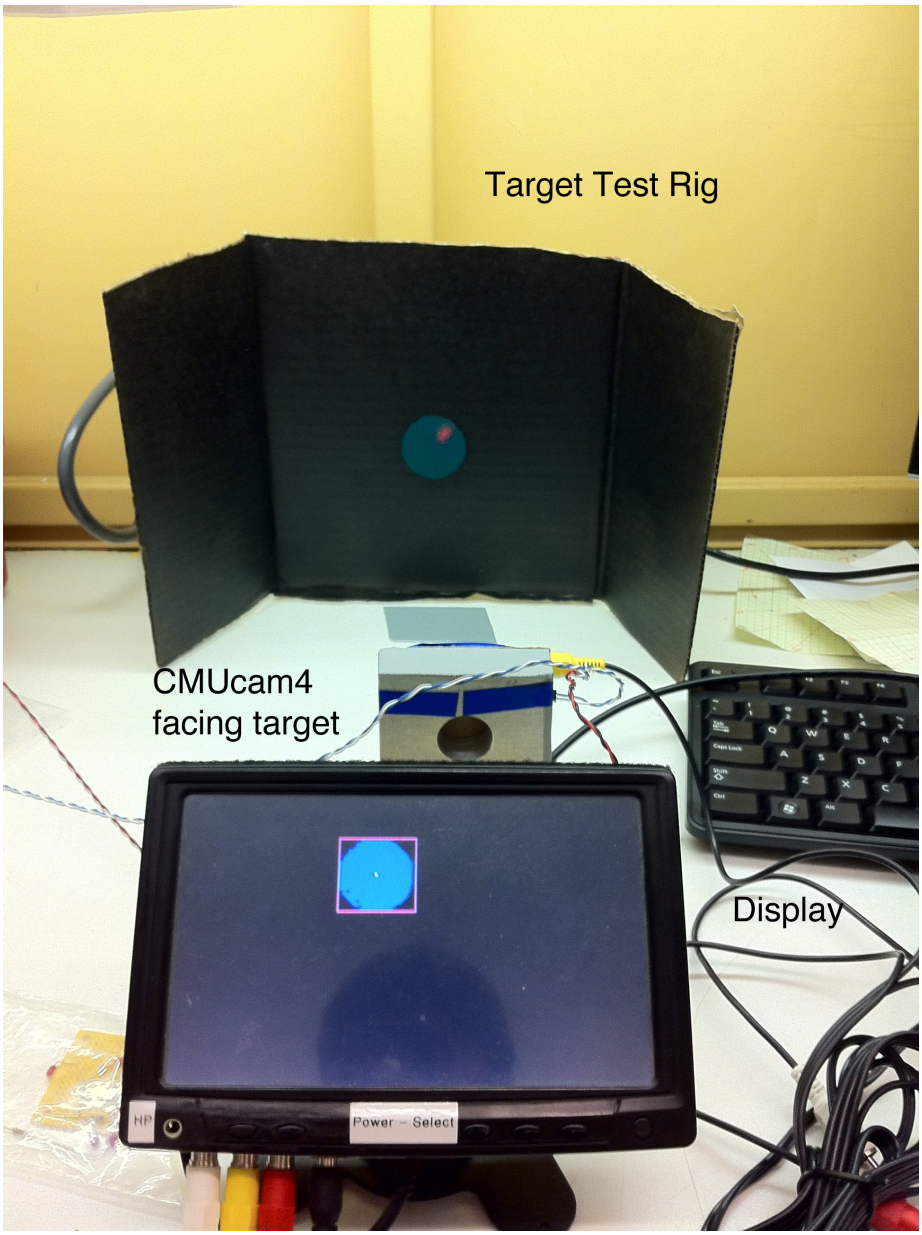

Figure 4. Demonstration Test Setup

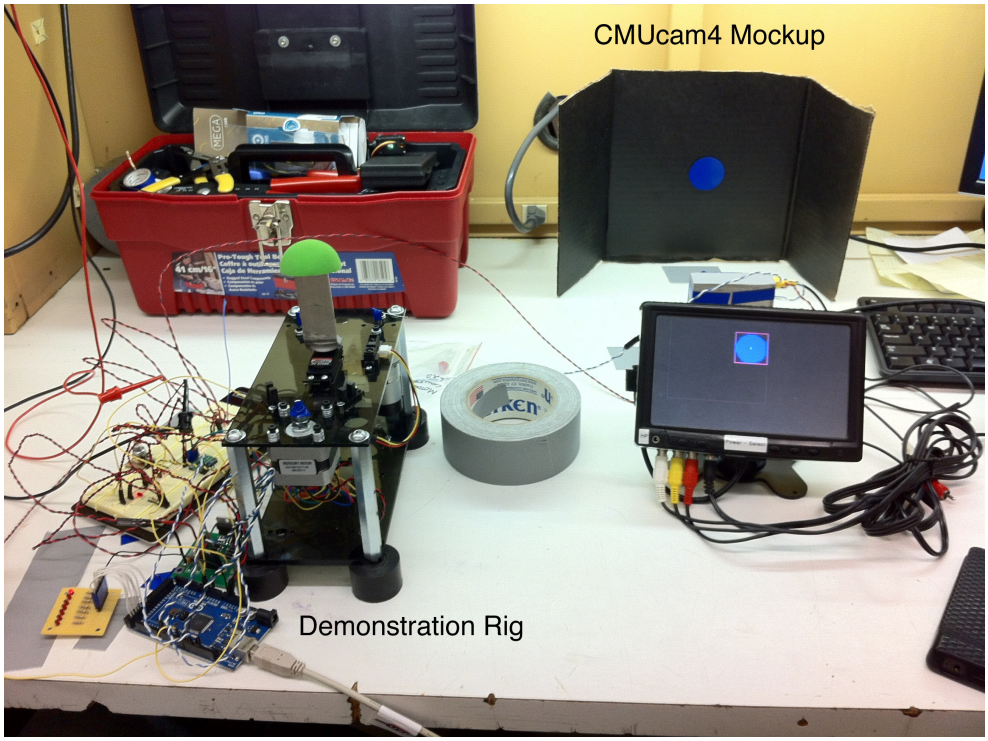

Figure 5. Full Lab 3 Demonstration Setup

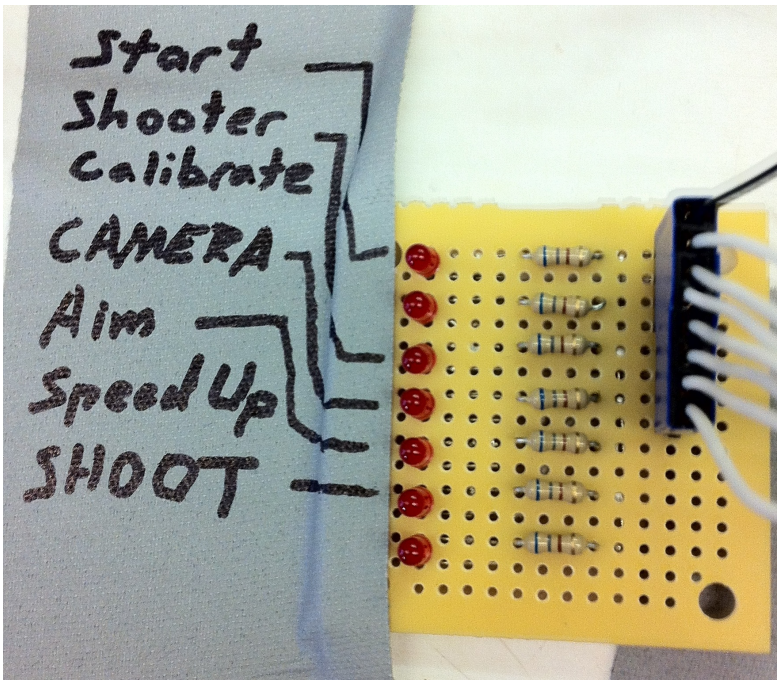

Figure 6. LEDs for State Machine State Cue

Figure 7. CMUcam4 Mounting

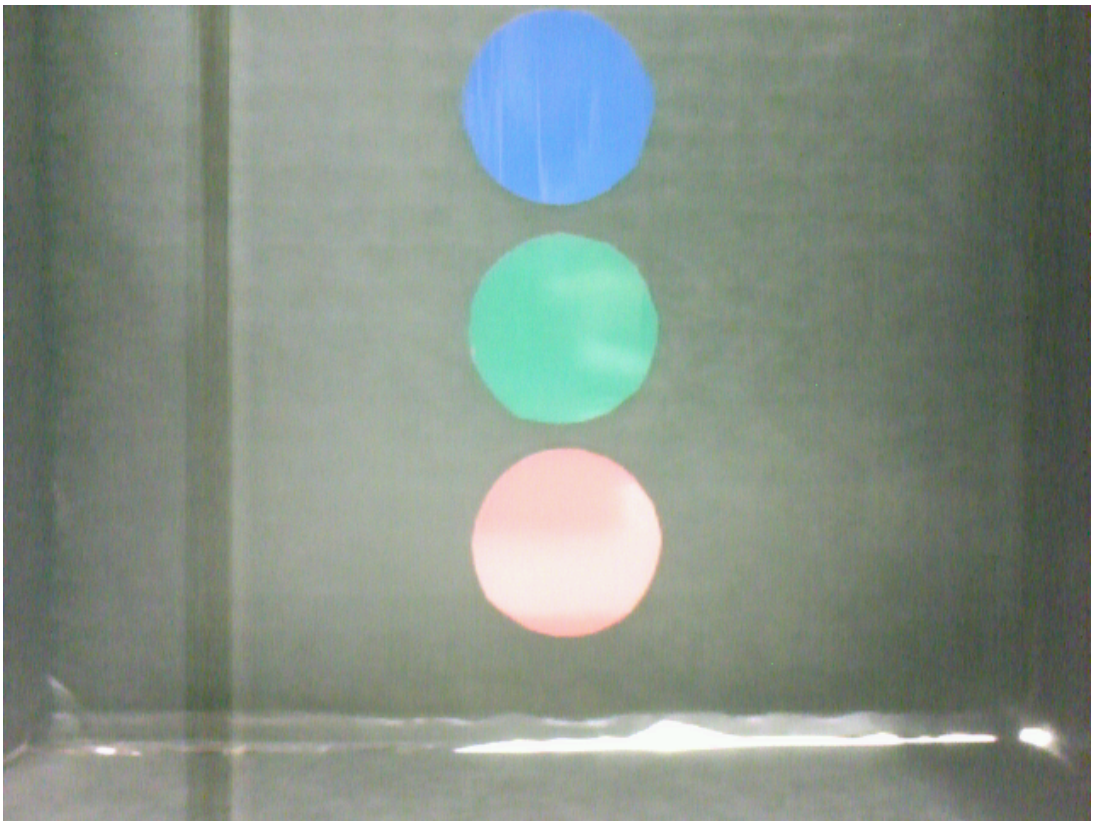

Figure 8. Image taken from New CMUcam4

Code

Below is code from CMUcam4 testing with new Arduino libraries. A similar version of this code was included in final demonstration version.

```
#include "CMUCam4.h"

#define led1 5
#define led2 6
#define led3 7
#define blob_left_led   8
#define blob_middle_led 9
#define blob_right_led   10
#define error_led       11

CMUCam4 camera;

unsigned char blob_x;
unsigned char blob_y;

int mode;
int error;

CMUCam4::CMUCAM4_color_range_t blueTracking;

CMUCam4::CMUCAM4_track_data_t trackingData;

void setup()
{
  camera.init();
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(blob_left_led, OUTPUT);
  pinMode(blob_middle_led, OUTPUT);
  pinMode(blob_right_led, OUTPUT);
  pinMode(error_led, OUTPUT);

  blueTracking.redLow = 40;
  blueTracking.redHigh = 120;
  blueTracking.greenLow = 160;
  blueTracking.greenHigh = 255;
  blueTracking.blueLow = 150;
  blueTracking.blueHigh = 230;

  trackingData.centroidX = 0;
  trackingData.centroidY = 0;
  trackingData.boundX1 = 0;
  trackingData.boundY1 = 0;
  trackingData.boundX2 = 0;
  trackingData.boundY2 = 0;
  trackingData.trackedPixels = 0;
  trackingData.trackingConfidence = 0;

  mode = 0;

}
```

```
void loop()
{
  if(mode == 0)  //send tracking information
  {
    camera.setTrackColor(&blueTracking);
    camera.startTrackColor(&blueTracking);

    delay(1000);  //give a second to settle

    mode = 1;
    digitalWrite(led1, HIGH);
  }
  else if(mode == 1)
  {
    //get tracking data, and note if error occured
    error = camera.getTrackData(&trackingData);

    if(error < 0)
      digitalWrite(error_led, HIGH);
    else
      digitalWrite(error_led, LOW);

    digitalWrite(led2, HIGH);
    delay(100);

    mode = 2;
  }
  else if(mode == 2)
  {
    //display information
    blob_x = trackingData.centroidX;
    blob_y = trackingData.centroidY;

    Serial.print("Error: ");
    Serial.println(error);

    Serial.print("Centroid X: ");
    Serial.println(blob_x);

    Serial.print("Centroid Y: ");
    Serial.println(blob_y);

    mode = 3;
  }
  else if(mode == 3)
  {
    //indicate with LED where target is
    if(blob_x < '2') //blob left (50)
    {
      digitalWrite(blob_left_led, HIGH);
    }
    else if(blob_x < 'n') //blob middle (110)
    {
      digitalWrite(blob_middle_led, HIGH);
    }
    else //blob_x < CMUCAM4_WINDOW_X_MAX
    {
      digitalWrite(blob_right_led, HIGH);
    }
  }
}
```