Final Report

by Mingwei Liu

Robot Name: Danner

Course Name: EEL5666 Intelligent Machine Design Lab

Instructors: Dr. A. Antonio Arroyo, Dr. Eric M. Schwartz

TAs: Devin Hughes, Tim Martin, Ryan Stevens, Josh Weaver

Table of Content

1. Abstraction

A robot is usually an electro-mechanical machine. It can be guided by computer or electronic programming, and also be able to do tasks on its own. The job of robots is to assist human beings to finish some dangerous tasks such as firefighting or construction. To build a robot needs a combination of knowledge such as programming, mechanics, control theory, materials science, etc. In this project, a line-following robot is going to be built to demonstrate the basic concept of robot design.

2. Executive Summary

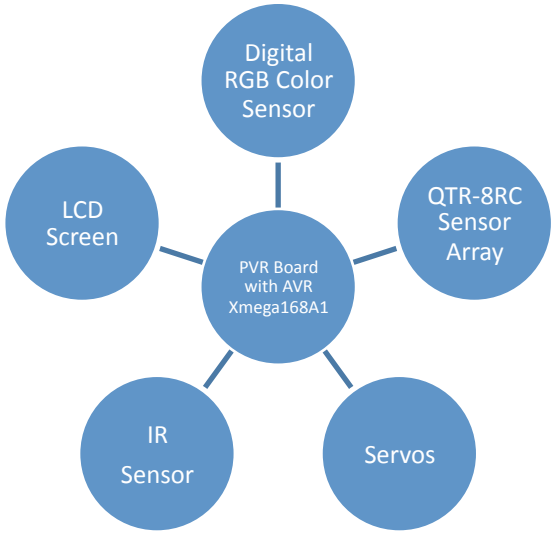
In this project, I built a robot which can follow a black line on a white platform. This can be implemented by using 8 photoresistors array. First, these sensors are set to be output. After delaying for a period of time and setting ports input agian, this array will recieve light from surroundings and transfers it into digital signal. Since black and white has different reflection coefficient, the robot can use this to distinguish whether it's on a line or not. The delay time is an important parameter because it can determine the sensitivity of the sensor. After experiment several times, I found 200us is the perfect delay time for my robot. Besides tracking the black line, the robot can also detect colors of an object. This can be achieved by using a distance IR sensor and a RGB color detection sensor. The IR sensor can calculate the distance of an obstacle in front. The robot can stop at certain point and use RGB color sensor to read from it. IR sensor can be easily used through ADC port. However, RGB color sensor has to use TWI communication protocol to obtain color because the color sensor contains much more information than IR sensor. For actuation part, at first I used two gear motors and a dual motor driver to control it. However, after changing the platform of my robot, I found the motors were not powerful enough so I replaced them with two servos. The robot and patrol on the map, going through every routine. When it detects object it will distinguish whether this object is red or blue. If the line ends, it will turn around and move on. The destination is a white area. The robot can go back after reaching the destination.

3. Introduction

The name of my robot is Danner. Danner is a line following and patrol robot. Its job is to detect victims when danger happens. It can search an area and go through every routine of this area. Whenever it detects victims in red color it will flash its LED to alarm. If it detects blue ones it will flash blue to say it's safe here. After completing

all the paths and rescuing all victims, it will reach its destination. Danner also has the ability to calculate the shortest routine to go back to the starting point without detouring. Danner can deal with sharp turns and can finish patrolling in a short period of time.

4. Integrated System



- 8 Photoresistor sensor array will help the robot to track the line.
- IR switches can calculate the distance to make robot stop when detects objects.
- RGB color detection sensor will be used to distinguish the red colored victim.
- 6 AA batteries are used as power supply.
- 2 LEDs
- LCD screen can show the status of IR and color sensors which are used in debugging process.

5. Sensors

■ 8 IR sensors-QTR-8RC Reflectance Sensor Array

This sensor module has 8 IR LED/phototransistor pairs mounted on a 0.375" pitch, making it a great detector for a line-following robot. Pairs of LEDs are arranged in series to halve current consumption, and a MOSFET allows the LEDs to be turned off for additional sensing or power-savings options. Each sensor provides a separate digital I/O-measurable output.

■ Sensors-Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm

- operating voltage: 4.5 V to 5.5 V
- average current consumption: 30 mA (typical)
- distance measuring range: 10 cm to 80 cm (4" to 32")
- output type: analog voltage
- output voltage differential over distance range: 1.9 V (typical)
- response time: 38 ± 10 ms
- package size: 29.5×13.0×13.5 mm (1.16×0.5×0.53")
- weight: 3.5 g (0.12 oz)

■ **Digital RGB Color Sensor-ADJD-S371**

- Four channel integrated light to digital converter (Red, Green, Blue and Clear).
- 10 bit digital output resolution
- Independent gain selection for each channel
- Wide sensitivity coverage: 0.1 klux - 100 klux
- Two wire serial communication
- Built in oscillator/selectable external clock

6. Actuation

- Servo Motors

  Servo motors, or servost ,are three-wire DC motors used extensively in the toy and model airplane industries, and in the steering on a radio-controlled car. This type of assembly incorporates a DC motor, a gear-train, limit stops beyond which the shaft cannot turn, a potentiometer for position feedback, and an integrated circuit for position control. We can use PMW to control the speed and direction of the servos.

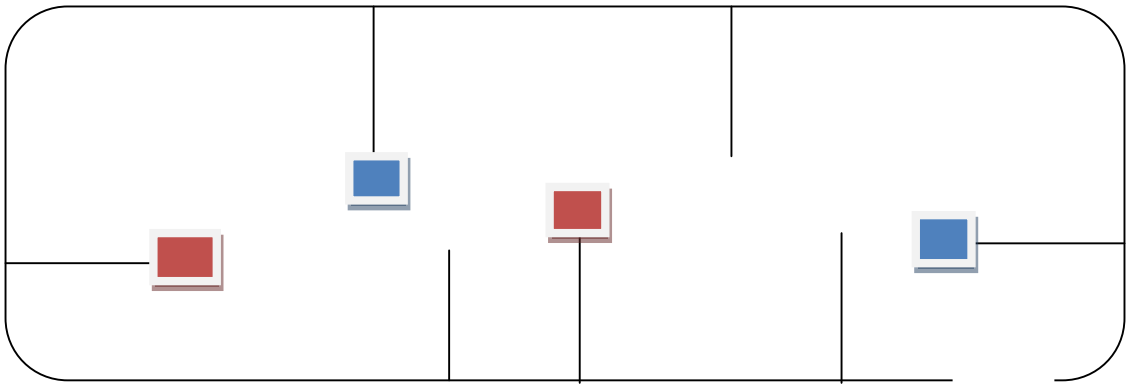
- 2 wheels-Pololu 42x19mm Wheel and Encoder Set

  This set includes a pair of 42×19mm wheels, a pair of extended brackets, and two matching encoders. Just pick a pair of micro metal gearmotors to complete your

feedback-enabled drive system.

7. Behaviors

- Searching all the area
- Avoid obstacles in front of it
- Distinguish red and blue, flashing corresponding LEDs
- Turn around if the line is finished
- After searching one branch, can continue go forward along the main road.
- Go back to the starting point with the shortest routine(did not implemented on Media's day)



8 Algorithm

Basic line following algorithm:

L4	L3	L2	L1	R1	R2	R3	R4

```
if (L==R){
        ServoC0(speed);
        ServoC1(speed);
        }
```

```
else if (R>L) {
            ServoC0(speed);
            ServoC1(-speed);
             }
else if (R<L) {
             ServoC0(-speed);
            ServoC1(speed);
               }
```

Petrol line following algorithm (pseudo code):

```
While(1){
If (IR detects object in certain range){
   Stop and detect colors, flash LEDs for several seconds;
After doing all these turn round;
}
else{
     if (detects all white){
      turn around
      }
    else if(detects a right sharp turn){
       turn right and set flag equal to 1;
      }
    else if(detects a left sharp turn){
       turn left and set flag equal to 2；
      }
    else if(detects all back which means it encounters a “T” intersections){
         if(flag==1)
               turn right;
         else if (flag==2)
               turn left;
    else
         use basic line-following algorithm
      }
 }
```

9. Experimental Layout and Results

The robot works fine in line-following part but not quite accurate when distinguishing colors. The color sensor has to be contact tightly with the object but IR sensor is not designed accurate for this purpose.

10. Conclusion

As I describe in the Experimental Layout and Results, color detecting was not successfully implemented. I think one approach to improve this situation is to replace IR with sonar sensor because sonar has higher accuracy compared with IR sensor. In this way, the robot can stop just in front of the object and contact with the object closely. The color sensor will be more accurate in this way. Another way to improve the behavior of my robot is to add returning part. After patrolling all the routines and reaches its destination, it can go back to the starting point directly without detouring. Actually I have written this part in a while loop. However, I could not find a proper way to make the program break from the first loop and enter the second loop. If the color sensor could be more accurate, I would use green as the signal for destination. When the robot detects green, it knows it reaches the destination and finishes the task and go back to the starting point directly. In this project I would thank Dr. Arroyo and Dr. Schwartz and all the TAs. I also want to thank Chien-Chih (Paul) Chao who helped me with RGB color sensor and Ruibiao Song who helped me hack my servos.

Appendices

Source Code:

```
void main(void)
{
    xmegaInit();                    //setup XMega
    delayInit();                    //setup delay functions
    ServoCInit();                   //setup PORTC Servos
    ADCAInit();                     //setup PORTA analong readings

    PORTCFG.MPCMASK = 0xFF;
    TWI_MasterInit(&twiMaster,      //setup portF to twi communication
                   &TWIF,
                   TWI_MASTER_INTLVL_LO_gc,
                   TWI_BAUDSETTING);

    adjd_init();

     int value=0;
    int L=0, R=0;
     int range1;
     int speed=50;
    int delaytime=200;
     int flag;
     int color;

    PORTQ_DIR=0xff;



    while(1)
   {

    PORTQ_OUT=0;
    PORTH_DIR|=0xFF;
    PORTH_OUT|=0xFF;
    delay_ms(10);
    PORTH_DIR&=0x00;
    delay_us(400);
    value=PORTH_IN;

    L=value&0b00001111;
```

```
R=value&0b11110000;
R=R>>4;
L=sb(L);

range1=ADCA0();

if ( range1>4090){
        ServoC0(-5);
       ServoC1(5);
      delay_ms(1000);
        color=adjd_read();
        delay_ms(100);
        if (color==1){
          PORTQ_OUT=1;
        delay_ms(3000);}
        else if (color==2){
          PORTQ_OUT=4;
        delay_ms(3000);}

        ServoC0(speed);
       ServoC1(-speed);
      delay_ms(delaytime);
          while(1){
      PORTH_DIR|=0xFF;
           PORTH_OUT|=0xFF;
            delay_ms(10);
            PORTH_DIR&=0x00;
           delay_us(400);
           value=PORTH_IN;

                  L=value&0b00001111;
                  R=value&0b11110000;
                  R=R>>4;
                  L=sb(L);
                if((L!=0)&&(R!=0))
                        break;
                        }

  }

else{
       if((L==0)&&(R==0)){        //all white turn around
                ServoC0(speed);
                 ServoC1(-speed);
```

```
while(1){
 PORTH_DIR|=0xFF;
    PORTH_OUT|=0xFF;
    delay_ms(10);
    PORTH_DIR&=0x00;
    delay_us(400);
      value=PORTH_IN;

    L=value&0b00001111;
    R=value&0b11110000;
    R=R>>4;
    L=sb(L);
 if((L!=0)&&(R!=0))
          break;
          }
}

else if((L==0x0f)&&(R==0x0f)){// all black
                    if (flag==1){
                        ServoC0(speed);
                            ServoC1(-speed);
                          delay_ms(delaytime);
                        ServoC0(speed);
                            ServoC1(speed);
                       delay_ms(delaytime);
                          ServoC0(speed);
                          ServoC1(-speed);
                         delay_ms(delaytime);}
                    else if (flag==2){
                        ServoC0(-speed);
                          ServoC1(speed);
                         delay_ms(delaytime);
                           ServoC0(speed);
                            ServoC1(speed);
                        delay_ms(delaytime);
                      ServoC0(-speed);
                           ServoC1(speed);
                          delay_ms(delaytime); }
                    else{
                           ServoC0(-5);
                            ServoC1(5);
                        }
                }
```

```
                    else
if(((L&0b00001000)==0b00001000)&&((R&0b000001000)==0)){
  //intersection LEFT
                                ServoC0(-speed);
                                ServoC1(speed);
                              delay_ms(delaytime);
                              ServoC0(speed);
                                  ServoC1(speed);
                              delay_ms(delaytime);
                               ServoC0(-speed);
                                ServoC1(speed);
                               delay_ms(delaytime);
                               flag=2;
                                 }

                     else
if(((R&0b00001000)==0b00001000)&&((L&0b000001000)==0)){
  //intersection RIGHT
                                 ServoC0(speed);
                                  ServoC1(-speed);
                                delay_ms(delaytime);
                               ServoC0(speed);
                                     ServoC1(speed);
                                delay_ms(delaytime);
                               ServoC0(speed);
                                  ServoC1(-speed);
                                delay_ms(delaytime);
                                flag=1;
                                 }

                    else {
                           if (L==R)
                           {
                              ServoC0(speed);
                              ServoC1(speed);
                            }
                         else if (R>L)
                           {
                              ServoC0(speed);
                             ServoC1(-speed);
                           }
                          else if (R<L)
                           {
                               ServoC0(-speed);
```

```
                                ServoC1(speed);
                               }
                        }
                    }

            }
                //the petrol part is finished, now begin return part
             /* while(1){

                  PORTH_DIR|=0xFF;
                    PORTH_OUT|=0xFF;
                    delay_ms(10);
                    PORTH_DIR&=0x00;
                    delay_us(400);
                    value=PORTH_IN;

                    L=value&0b00001111;
                    R=value&0b11110000;
                    R=R>>4;
                    L=sb(L);


if(((L&0b00001000)==0b00001000)&&((R&0b000001000)==0)){
  //intersection LEFT
                    ServoC0(speed);
                    ServoC1(-speed);
                   delay_ms(200);

                   }

                    else
if(((R&0b00001000)==0b00001000)&&((L&0b000001000)==0)){
  //intersection RIGHT
                    ServoC0(-speed);
                    ServoC1(speed);
                   delay_ms(200);
                   }
                  else {
                               if (L==R)
                               {
                                  ServoC0(speed);
                                  ServoC1(speed);
                                }
                       else if (R>L)
```

```
                {
                  ServoC0(speed);
                 ServoC1(-speed);
                }
            else if (R<L)
                {
                   ServoC0(-speed);
                  ServoC1(speed);
                 }
          }
        }*/
}
```