

Final Report

Jared Pasik

Grill Mate

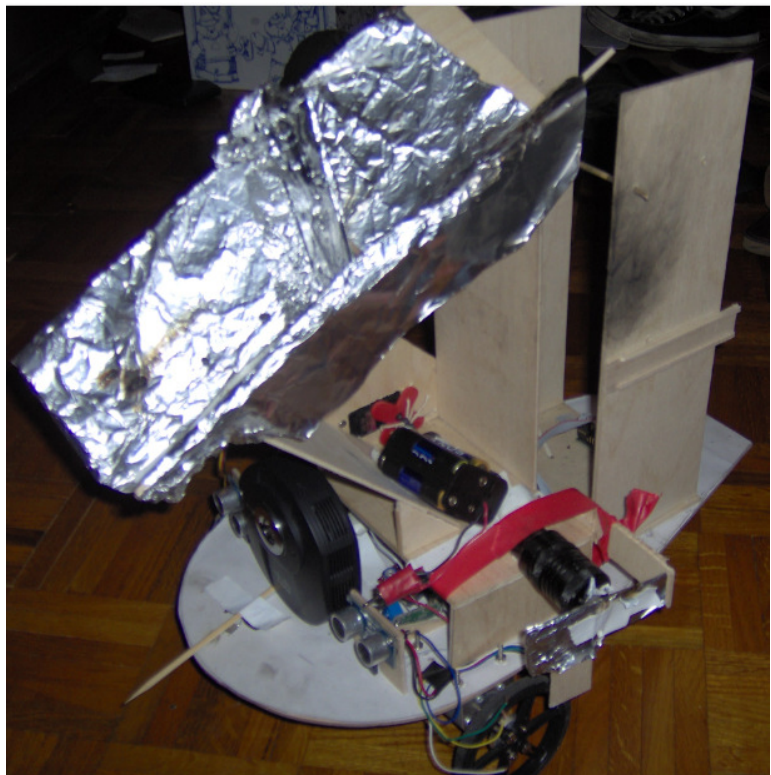
EEL 4665

For Dr. A. Antonio Arroyo and Dr. Eric M. Schwartz

Sean Frucht, Devin Hughes, Tim Martin, Ryan Stevens, and Josh Weaver

Table of Contents

Abstract	3
Executive Summary	4
Introduction	5
Integrated Systems	6
Mobile Platform	7
Actuation	8
Sensors	10
Behaviors	12
Experimental Layout and Results	14
Conclusion	16
Documentation	17
Appendix A: AVR code	18



Abstract

This report is a guide to the design and testing of a robot that makes use of a laser to light a grill. The robot uses an IP webcam coupled with OpenCV software to locate a grill and sonar to avoid objects while looking for the grill. Once locating the grill a bump switch is triggered and the laser is turned on to light a fuse which will ignite the onboard charcoal. This charcoal is, shortly after ignition, deposited the grill where other charcoal await.

The robot was successful in avoiding obstacles and “seeing” the color red although this was not accurately demonstrated during Media Day. The primary issue remaining to tackle is the connection of the Xbee antennae so information can be relayed from the webcam to the microprocessor. Suggested future work is also outlined in this document.

Executive Summary

A robot was designed to locate and light a charcoal grill. It was to do this by using a wireless IP webcam to determine if the grill was in front of the robot. While looking maneuvering toward the grill any obstacles in the way of the robot will be avoided with the use of sonar to measure the distance between the nearest object and the robot.

Once the robot has detected the grill the webcam will communicate with the robot to determine which direction the robot now need to go to reach the grill. The webcam will be used to generate three signals: turn left, turn right, and stay straight. Each will be given a numerical value which will be interpreted and sent to the motor controllers.

When the robot gets close enough to the grill that the coal will fall from the chute into the grill a bump sensor on the front of the grill will be pressed causing the robot to stop moving. With the robots motion stopped the board will activate a servo that will turn the key of a laser into the armed position. With the power button already activated the laser will fire a beam at the nearby match head. After a few seconds the match will ignite and the fuse will be followed to the awaiting coals on the chute. After a brief period of time the board will activate a second servo along the chute. This servo will remove the gate from the chute allowing the light coals to fall into the grill below.

Everything about this robot works except for the serial communication between the two Xbee antennae. The signal is successfully sent but cannot be received at this time. For this reason, the robot is unable to locate the grill on its own. The image detection software allows the computer running the software to know where the grill is but the computer/Xmega connection is nonexistent.

Future work into this robot will consist of fixing the Xbee connection so that the grill will be able to achieve its goal of locating the grill.

Introduction

While hosting a party or group gathering during the warmer months food is often required to satisfy your hungry guests. The easiest thing would be to call for pizza and have it delivered. This can become quite pricy as the summer rolls along. For this reason starting up the grill might prove a more cost effective option for feeding the masses. The only issue is that if the game is on inside you have to be outside. A way around this is to leave the grill unattended while it comes to heat, which is a bad idea for many reasons. To combat this unfortunate obstacle I propose a robot capable of starting a grill and alarming everyone if the fire has gotten out of control.

The robot will need to navigate outdoors as it will be setting fire charcoal. Outdoor navigation will be done using sonar devices because IR sensors are rendered ineffective due to the sun. The robot will locate the grill using a mounted IP camera. As the grill is approached the grill will interact with a bump sensor mounted at the front of the robot. Once activated, the bump sensor will cause the robot to cease object avoidance. A mounted laser will be fired igniting a fuse which will in turn be used to light charcoal.

The design process was first mapped out so that a clear goal could be achieved. This also allowed for easy reference to make sure that nothing extraneous was added. With a goal in mind and steps to get there laid out the next step was to develop a platform. Then actuation became necessary to make the platform mobile. This was followed by that addition of sensors to the platform so that the robot would know about its environment and be able to make decisions. Finally, the robot was coded with behaviors that would allow for the accomplishment of the design goals. Along the way experiments were done to determine the effectiveness of the various parts.

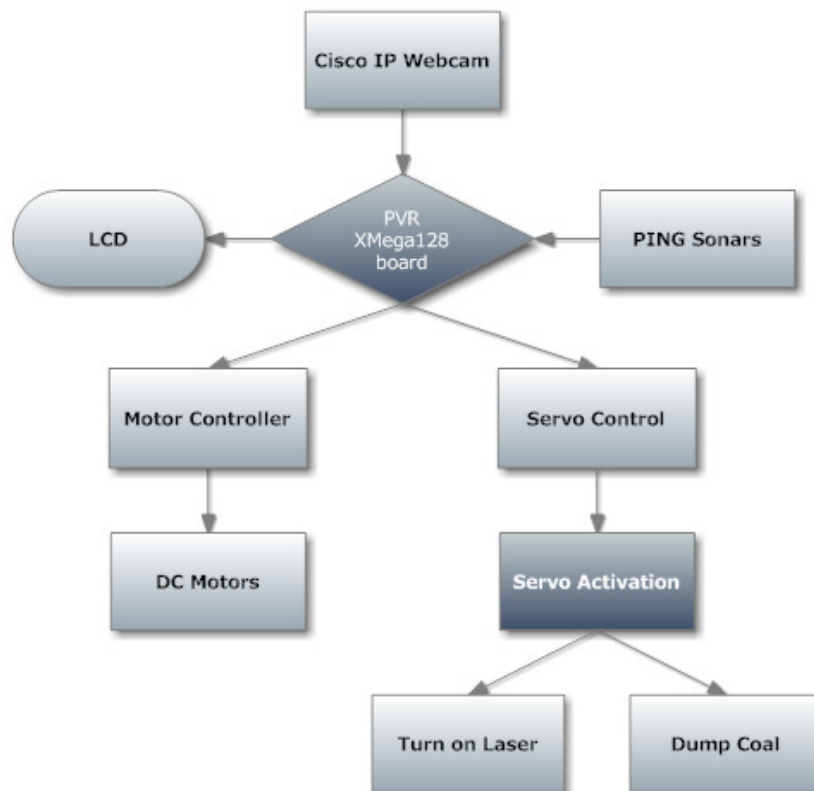
Integrated System

The robot, Grill Mate, will be controlled by the PVR XMega128 microprocessor. It will receive data from the dual PING sonars and the Cisco IP Webcam. The sonars will be used to object avoid while the IP webcam searches out the grill. The sonars are connected to ports B and Q on the XMega board.

The webcam communicates with the board via two Xbee chips. The first will be connected to the computer running the OpenCV software and the other will be connected to port F on the XMega board. This allows for the XMega board to determine which direction to head while moving toward the grill.

The microprocessor will direct the motor controllers by toggling ports J and F high and low depending on the desired direction.

Based upon how close the grill is to Grill Mate the laser will turn on to ignite the fuse which will light some charcoal. This charcoal will be released via the coal dump apparatus into the grill where additional coals are waiting to be ignited.



With this setup Grill Mate will be able to locate a grill, avoid obstacles while searching and heading toward the grill, and ignite the grill without any outside assistance.

Mobile Platform

For this autonomous grill starter the platform will have to be relatively robust to carry some charcoal to the grill. The charcoal will also have a number of matches inside to be used as the ignition source. It will also need an arm with the ability to ignite the grill. The lighting of the matches will be done with a laser pointer, specifically a green laser pointer with 200mW output. With the laser pointer focused on the coals and matches the temperatures will get high enough to combust the matches and in turn the charcoal.

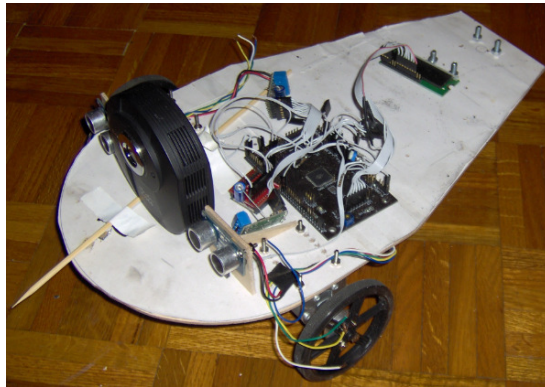


Figure 1. Robot platform. Also shown are all of the motors and drivers as well as the sensors.

The platform is made from 1/8" thick board. It was cut free hand using an electric saw and drill. Tolerances were not kept tight.

Actuation

Wheels will be suitable for transporting Grill Mate from his resting location to the grill. They are a better choice than legs because legs may not be able to support the weight of the charcoal without high torque servos. The robot will roll on two large wheels placed in the back and one caster wheel placed in the front. The front wheels will be 90mm in height while the height of the caster will be roughly 3in. They will have to be small enough to keep the center of mass far enough back that the robot will not tip over during or after dumping the charcoal into the grill.



Figure 2. 90mm Pololu tires.

Servos will be used to fire the laser at the fuse as well as dump the coal into the grill. These applications require less torque and instantaneous power allowing for the use of servos. The servo activating the laser will only need to move 45° to cause the laser to fire. It will hold the position for approximately 30 seconds then the servo will go to its original position and turn off the laser. The servo dumping the charcoal will also rotate 45° so that the arm can be removed from the chute and the coals can be dumped into the grill. The Hitec HS-311 servo receives a PWM signal from PORTC and PORTD for its application in Grill Mate



Figure 3. Hitec HS-311 standard servos.

The wheels will be moved using a DC motor rather than a servo. This is because a larger amount of torque will be required to move the robot with its full payload. Also, these motors will have to be capable of forward and reverse motion so that obstacle avoidance can be used while the IP cam is searching for the grill.

I bought encoded wheels from Pololu thinking that I might save myself some time to have someone else put the wires onto the wheels and maybe I would be able to make some use out of the revolution counting tool. But as it turned out I spent about \$15 more than I needed to just to have four extra wires per motor.

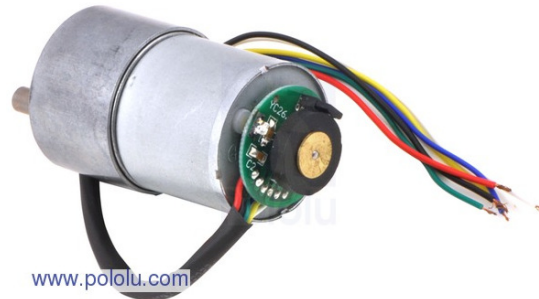


Figure 4. 131:1 Metal Gearmotor 37Dx57L mm with 64 CPR Encoder.

Motor encoders were needed to actively control the motors. The encoders would control if the motor spun in the forward or the reverse direction by determining if the direction port of the encoder was set high or low. A low signal to the port would cause the motors to operate forward while a high signal would cause the motors to operate in reverse. The motors could be stopped by declaring the direction pin to be an input. The motor controllers power the motors with a PWM signal from PORTD. The direction that the motors need to turn is determined by toggling the output of pin 0 on PORTF high and low.

It took a long time to properly code the encoders so that they would move in the forward and reverse directions. This was primarily due to my shortcomings with the AVR Studio code and my knowledge of microcontrollers in general. I had been using the wrong pin to send my direction signal to the encoder. As a result, the motor only had one speed and direction until a week later when I figured out my issue.

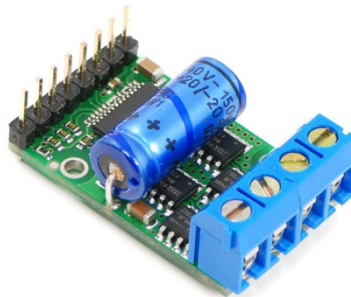


Figure 5. Pololu High-Power motor driver 36v9.

Sensors

A tilt sensor will be used to stop the object avoidance once the grill has been detected by the IP cam. It will be placed in front of the robot so that it can run into the grill and start the coal lighting process. If the robot has detected the grill then the tilt sensor will act as a bump sensor.

This was left out of the final design. The solution proved to be annoyingly difficult so I used a bamboo skewer to press a bump switch to activate the laser servo and ultimately the servo to dump the light charcoal.

Feedback will be received through a mounted LCD display. This will primarily be used to determine if all of the other sensors and actuators are behaving as they should. During object detection and igniting the charcoal the LCD display will not be read manually. The LCD display is mounted to the LCD port on the XMEGA with the exception that the Vcc power is received from PORTC for the higher voltage.

I had trouble getting the LCD display to work because I did not look at the minimum voltage requirement. To correct this I had to get a potentiometer and wire it into the 5V output of one of the PWM channels.



Figure 6. Lumex Alphanumeric 16x2 LCD display.

Bump sensors will be my last resort against front end collisions with objects. They should not be used, but if they are they will cause the robot to reverse in an attempt to get away from the object it has encountered.

Only one bump sensor was included in the final design. It was used as a switch to alert the robot that it had arrived at the grill and was wired into PORTH.

Sonar will be the way the robot navigates through the outdoors until it locates the grill. There will be two sonars placed at the front of the robot where they will read the distances between themselves and the nearby objects. Two sonars are necessary because each has an effective window of 20° in all directions. To effectively canvas the front of the robot at least two sonars

will have to work together. The sonars were wired into PORTC to receive power and ground and into PORTB and PORTQ to send and receive their signal.

Objects will be avoided if they come within 8 inches of the front of the robot. This will be an acceptable distance because the robot will not be moving at great speeds. In this way, sonar will be used, in part, to control the motor encoders. If the webcam determines that the grill is in front of the robot then the 8 inch limit will be ignored and the grill will become the primary target.



Figure 7. Parallax PING))) Ultrasonic Sonar.

The responsibility of detecting the grill will belong to the IP webcam. The webcam will use OpenCV to analyze the images “seen” by the webcam and determine if a grill had been in front of the robot. This is the most important sensor because all actions besides object avoidance hinge on the robot’s ability to locate the grill.

To do this the color of the grill is what the webcam is looking for. Once a red object enters the view OpenCV sends out the direction of the grill to the Xmega through coupled Xbee antennae. The code will take this character and interpret it as a “left”, “right”, or “go straight” command. The Xbee is powered by the 3.3V output and receives the serial data through pins 2 and 3 of PORTF.

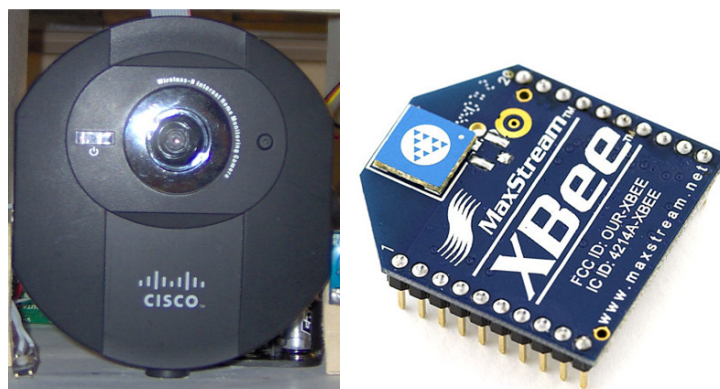


Figure 8. Linksys WVC80N IP webcam and Xbee antennae.

Behaviors

Grill Mate makes use of several behaviors as it looks to light the grill. It starts by practicing obstacle avoidance. It can be seen that the robot is doing this task by looking at the LEDs on top of the sonars. The obstacle avoidance is controlled by the code shown in the Appendix. It determines how close the nearest object is to its right and left side and either turns away, backs away, or keeps its current bearing. The implemented distance threshold is 1000. This corresponds to approximately 10in directly in front of the sonar.

During obstacle avoidance, Grill Mate will be searching for the grill which will be seen using color detection and thresholding in OpenCV. The robot is looking for the color red; all other colors will be dismissed as background and unimportant to the current task.

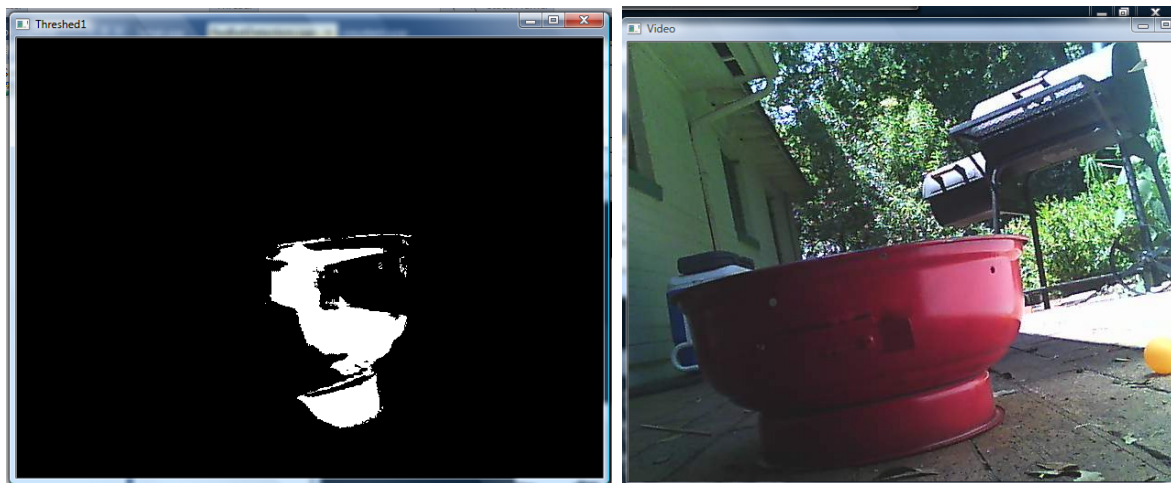


Figure 9. OpenCV image processing of webcam image with thresholding.

Once the grill is seen, if the grill location is returned as 2, Grill Mate then the obstacle avoidance is temporarily suspended allowing for full concentration of the navigation to the grill. If the location of the grill in the OpenCV window is determined to be between 300 and 400 then the grill is in front of the robot and the value 9 will be sent to the microprocessor. If it is less than 300, a value of 7 is returned and the robot will turn right until the grill is again in front. Likewise, if the grill location is greater than 400, a value of 8 is passed and the robot will turn left.

```
50.1 KB received
frame loadedposition <351,321,9>
Saving to 'snapshot.cgi.jpeg'...
50.1 KB received
frame loadedposition <352,319,9>
Saving to 'snapshot.cgi.jpeg'...
50.1 KB received
frame loadedposition <348,316,9>
Saving to 'snapshot.cgi.jpeg'...
50 KB received
frame loadedposition <348,318,9>
Saving to 'snapshot.cgi.jpeg'...
50.3 KB received in 1 seconds (50.3 KB/sec)
frame loadedposition <346,317,9>
```

Figure 10. Print out of OpenCV color recognition program. Position is given as (position x, position y, and return value).

When the grill is reached the fore mounted bump sensor is activated and the servo controlling the laser is turned 45° to arm the laser. The laser beam combusts a match which is attached to a short fuse leading to the chute where a few coals sit. After 3 minutes, long enough for the fuse to burn and the coals to be ignited, the coals in the chute are released and allowed to fall into the grill. If the coals were allowed to burn long enough they will not be put out by the fall and will ignite the coals sitting in the grill.

Experimental Layout and Results

To test the effective distance of the laser in lighting matches I tested it under three different conditions. First with one “out-of-the-box” match then with one “tipped” match finally with a combination of matches from the first two categories.

Table 1. Distance of laser from matches and time it took to ignite.

Distance (in)	Out-of-the-box (s)	Tipped (s)
6	7	7
9	8	9
12	10	9

This data showed that there is no appreciable difference between the out-of-the-box matches and the black tipped matches as far as combustion. The closer the laser is to the matches the easier it is to keep the beam focused on a single point on the match. This is accredited as the reason that the distance was a factor. Still, in the final design closer was selected as better than far away and the match was placed less than an inch away from the front face of the laser.

To test the tilt sensor a servo was tethered to the tilt sensor so that when the sensors connection was broken the servo would be activated. The tilt sensor was attached to a dowel rod so that the angle could be accurately measured.

No data was recovered from this experiment because of inconsistent data. The connection would seemingly spontaneously appear and break. No additional testing was done using this apparatus because it was determined to be too sensitive and not reliable especially as a trigger for the laser and coal dump apparatus.

The sonar was tested by placing book parallel to the back and a predetermined distance away from the sensor. The angle of the object from the center of the sonar was altered in increments of 10° starting from zero displacement until the object was no longer detected. The distance and angle from center of the sonar were also recorded. This was done to test the visual envelope of the sonar and to determine where it should be mounted on the robot to optimize the ability to avoid obstacles.

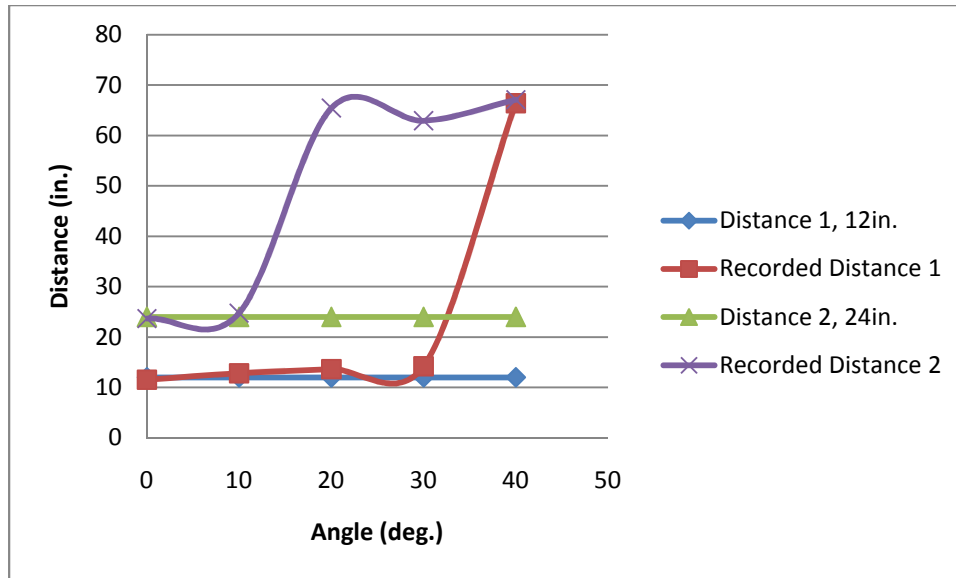


Figure 11. Sonar test run at two different distances in front of the sensor.

The data showed that the closer the object was to the sonar the wider the effective range of the sonar. The spikes in the data reflect where the sonar was unable to detect the object. The accuracy of the distance remained quite constant throughout the experiment. This should be acceptable for my application because I will be analyzing a distance less than 12in. The data suggests that for my desired operation condition of 10in that the sonar will be able to accurately detect object within 30° of the sonar's centerline.

Conclusion

For Media Day my robot was capable of object avoidance, detecting the grill, and lighting the grill. Media Day did however not go well for Grill Mate and the object avoidance was not accurately demonstrated and neither was the grill detection. It was later determined that the reason for object avoidance failing was due to the left sonar's signal pin falling out. Grill Mate needed some assistance in lighting the grill as the coal dump servo activated early and the fuse died out on its way to the coals.

The greatest limitation to the work that I've done was my own general lack of knowledge. Nearly everything was done in trial and error. This resulted in the project taking much longer to complete than I had anticipated and, ultimately, an incomplete project.

The laser and the color filtering capabilities of OpenCV and the ease of implementation were the two things that exceeded my expectations. The laser was able to work every time lighting the match when placed directly in the beam. The only issue with ignition was when the laser operator could not keep the beam focused on a single point. The robot did not have this problem. Color detection and thresholding in OpenCV allowed me to see the grill very clearly and was robust enough to filter out most other near-red colors (ie shades of orange).

If I were to start all over I would probably utilize hacked servos instead of motors for the wheels. The motors operated at different speeds every time. They were so unpredictable that during my demo all Grill Mate could do was spin in a circle because one motor was spinning that much faster than the other. Additionally, I would get a separate battery pack for the motors since they were the single largest power drain. I would change the body construction to some light weight metal, likely aluminum. It is very likely that Grill Mate will catch fire at some point if this upgrade is not implemented.

Documentation

PING Sonar: <http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/28015-PING-v1.6.pdf>

Motor Driver: <http://www.pololu.com/file/0J165/FDS5672.pdf>

And <http://www.pololu.com/catalog/product/756/resources>

Motor: <http://www.pololu.com/catalog/product/1447/specs>

LCD: <http://www.lumex.com/specs/LCM-S01602DTR%20M.pdf>

OpenCV: <http://opencv.willowgarage.com/wiki/>

Color Detection: <http://www.andol.info/wp-content/uploads/2009/02/colorrecg.pdf>

Appendix A: AVR code

```
//XMega code:
//Assistance:    Sonar: Thank you Terrance Tai
//              Serial: Thank you Eric Bennett
//              Xbee Communication: Thank you Eric Bennett
//              Vision: Thank you Justin Goll
//              And a BIG Thank you to all of the TAs! My robot didn't work but I'm proud of it!
#include <avr/io.h>
#include "avr_compiler.h"
#include "usart_driver.h"
#include "PVR.h"
#include <stdio.h>
#include <util/delay.h>
#include "stdlib.h"

#define USART USARTF0
#define USART_BAUD 11500
#define SERIAL_UBBRVAL(baud) (((F_CPU / 16) + (baud / 2)) / (baud)) - 1

#define SAMPLE_SIZE 32/////////originally 32, leave it here.
USART_data_t USART_data;

///////// Serial Code Stuff (from Steven) ////////////

void usart_initialize(void)
{
    //pin 3 output
    PORTF.DIRSET = PIN3_bm;
    //pin2 input
    PORTF.DIRCLR = PIN2_bm;
    USART_InterruptDriver_Initialize(&USART_data, &USART, USART_DREINTLVL(3));
    //usartc0, 8 data bits, no parity, 1 stop bit
    USART_Format_Set(&USART, USART_CHSIZE_8BIT_gc, USART_PMODE_DISABLED_gc,
false);
    //ENABLE INTERRUPT
    USART_RxdInterruptLevel_Set(USART_data.usart, USART_RXCINTLVL(3));
    //set baud rate
```

```

USART_Baudrate_Set(&USART, 17, 0); ////////////////

//ENABLE RX AND TX
USART_Rx_Enable(&USART);
USART_Tx_Enable(&USART);
//Enabel PMIC Interrupt level low
//PMIC.CTRL |= PCMIC_LOLVLEX_bm;

//enable global interrupts
//sei();
}

inline void usart_tx_byte(char DataByte)
{
    int txrxVal = 0;

    while(1)
    {
        txrxVal = USARTF0_STATUS;
        txrxVal &= 0x20;;
        if(txrxVal == 0x20)
        {
            USARTF0_DATA = DataByte;
            break;
        }
    }
}

inline void usart_tx_string(char *StringPtr)
{
    int i = 0;
    while(StringPtr[i] != 0) //while not null terminator
    {
        usart_tx_byte(StringPtr[i]);
        i++;
    }
}

```

```

inline char usart_rx_byte(void)
{
    int txrxVal = USARTF0_STATUS;
    txrxVal &= 0x80;
    char data;
    //// used to be char
    if(txrxVal == 0x80)
    {
        data = USARTF0_DATA;
    }
    else
    {
        data = 0;
    }
    return data;
}

```

//////////////////// int used to be char

```

void main(void)
{
    xmegaInit();
    delayInit();
    ServoCInit();
    ServoDInit();
    ADCAInit();
    lcdInit();
    usart_initialize();
    lcdGoto(0,0);
    lcdString("Sonar Sonar ");
    lcdGoto(1,0);
    lcdString("0 0 ");
    int i1 = 0;
    int i2 = 0;
    int j = 0;
    int digits = 0;
    int base = 1;

```

```

//setup XMega
//setup delay functions
//setup PORTC Servos
//setup PORTD Servos

//setup LCD on PORTK

```

```

char usdel1[10] = {0};
char usdel2[10] = {0};
char ir0adc[10] = {0};
char ir1adc[10] = {0};
int ir0 = 0;
int ir1 = 0;
int go = 0;

char avalue;
int intvalue;

avalue = usart_rx_byte();
intvalue = avalue - '0';
lcdInt(intvalue);

PORTF_DIR |= 0xff;
PORTJ_DIR |= 0xff;
PORTH_DIR |= 0x03;

// ServoC5(-90);                               //servo to dump coal
ServoD5(80);                                   //servo to turn on laser

while(1) {
//      if(PORTH_IN & 1) {
//          go = 1;
//          while(go) {

                ir0 = ADCA0();
                ir1 = ADCA1();
                lcdGoto(1,0);
                PORTB_DIR |= 0x01;
                PORTQ_DIR |= 0x01;
                PORTB_OUT = 1;
                PORTQ_OUT = 1;
                _delay_us(1);
                PORTB_OUT = 0;
                PORTQ_OUT = 0;
                PORTB_DIR &= 0xFE;

```

```

PORTQ_DIR &= 0xFE;

i1=0;
i2=0;

MotorD0(2000);
MotorD2(1250);

while(!(PORTB_IN & 1) & !(PORTQ_IN & 1));
while((PORTB_IN & 1) | (PORTQ_IN & 1)) {
    _delay_us(1);
    if(PORTB_IN & 1) {
        i1++;
    }
    if(PORTQ_IN & 1) {
        i2++;
    }
}

if((i1 >= 1000) & (i2 >= 1000)) {
    PORTF_OUT = 0;
    PORTJ_OUT = 1;
}

if((i1 < 1000) & (i2 < 1000)) {
    PORTF_OUT = 1;
    PORTJ_OUT = 0;

    MotorD0(1250);
    MotorD2(1000);
}

// else if((i1 < 1000) | (i2 < 1000)) {
    MotorD0(-100);
    if(i1 < 1000) {
        PORTF_OUT = 1;
        //MotorD0(-100);
        _delay_us(500000);
    }
}

```

```

    }
    if(i2 < 1000) {
        PORTJ_OUT = 0;
        //MotorD2(-100);
        _delay_us(500000);
    }
}

if(intvalue == 2) {
    PORTF_OUT = 1;
    PORTJ_OUT = 0;
}
if(intvalue == 8) {
    PORTF_OUT == 0;
    PORTJ_OUT == 1;
}

}

LcdGoto(1,0);
sprintf(usdel1,"%d",i1);
LcdString(usdel1);
//
//
sprintf(usdel1, "%d", intvalue);
LcdString(usdel1);
base = 1;
digits = 0;
while(i1/base) {
    digits++;
    base = base*10;
}
for(j = 0; j < (7-digits);j++) {
    LcdString(" ");
}

}

sprintf(usdel2,"%d",i2);
LcdString(usdel2);
base = 1;
digits = 0;
while(i2/base) {
    digits++;
}

```

```

        base = base*10;
    }
    for(j = 0; j < (5-digits);j++) {
        lcdString(" ");
    }

    sprintf(ir1adc,"%d",ir1);
    base = 1;
    digits = 0;
    while(ir1/base) {
        digits++;
        base = base*10;
    }
    for(j = 0;j < (4-digits); j++) {
        lcdString(" ");
    }

    if(PORTH_IN & 1) {
        while(1) {
            MotorD0(-200);
            MotorD2(-200);
            go = 0;

            _delay_us(5000);
            ServoD5(-80);
            _delay_ms(180000);
            ServoC5(80);
        }
    }
}
// }
// else {}
// MotorD0(-100);
// MotorD2(-100);
}
}

```