Final Report

FIREMAN

Date: 3/22/11
Student Name: **Jiecheng Chen**
TAs: Tim Martin
Ryan Stevens
Devin Hughes
Josh Weaver
Instructors: Dr. A. Antonio Arroyo
Dr. Eric M. Schwartz

University of Florida
Department of Electrical and Computer Engineering
EEL 4665/5666
Intelligent Machines Design Laboratory

Table of contents

Abstract

This robot is a fire-fighting robot. It can put out fire with water while fire source is found. It uses two wheels to run on the ground. Also, it can avoid obstacles and find fire source in a close space. What is more, I will make this robot to find the fire source in a relatively short time.

Introduction

I decide to keep the robot chassis as simple as possible to achieve the tasks it needed to. After watching some videos about flame extinguishing robot and some papers and report about this robot, I have got the basic approach of how to design this robot. The robot needs to be: light in weight, small and an easy shape to move around the table and avoid obstacles and cheap to manufacture. Also, my robot can find and extinguishing the flame as fast as possible. It could search for the area in a fast way and then scan and locate the flame.

This paper includes the components, basic functions and algorithm about Fireman's behavior.

Integrated System

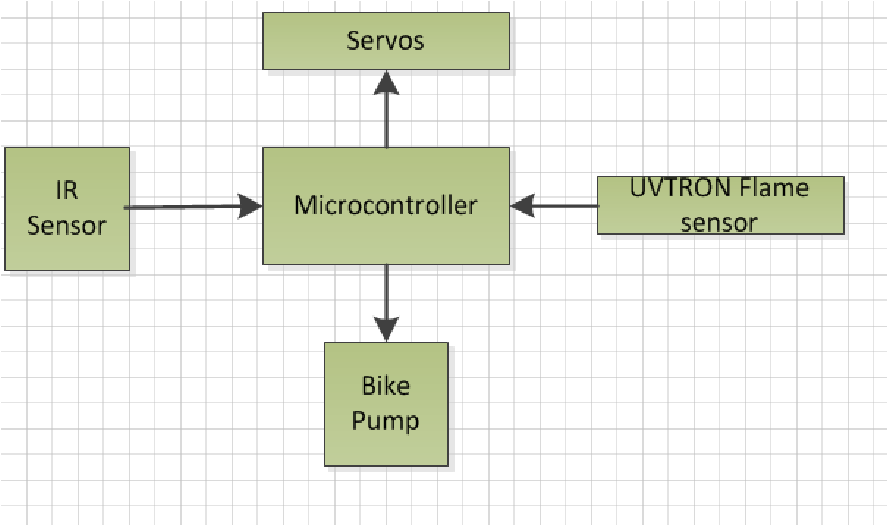

Fig 1. various components of the robot

The following is how each component functions within the system:

- Microcontroller: A Pridgen Vermeer Robotics Xmega128 board is used in this robot. The board can receive IR sensor data and outputs PWM signals to LCD.

Also, when the flame is detected, the sensor will send a high logical signal to PORTB, and then the board will detect the flame.

- IR Sensor: To implement the obstacle avoidance, the robot uses two IR sensors in front of it. The sensors communicate data via the A/D ports of the microcontroller and microcontroller can feedback the distance to LCD.
- Flame sensor: A Hamamatsu UVTRON R2868 flame sensor is located on the top of the robot. This sensor communicates with microcontroller through a driver board. When flame is detected, a 5V pulses will be generated from the pulse out pin.
- Flame extinguisher: Fan When the flame is detected in front of the robot, the microcontroller will turn on the fan and put off the fire.
- Battery packs: 6 Nimh AA batteries

The whole model of the Fireman

The chassis has two layers. In the bottom of the lower layer, a circular shape with two drive wheels should be used. Battery pack and motors are located on the second layer. On the upper layer, the PVR board and some sensors, such as bump sensor, IR sensor and flame sensors will be placed here. In addition, above the robot there will be a device such as a fan or a pump to get ready for putting out the fire. I will try to install “arm” in this robot that it can lift a cup of water and use it to put out the fire.

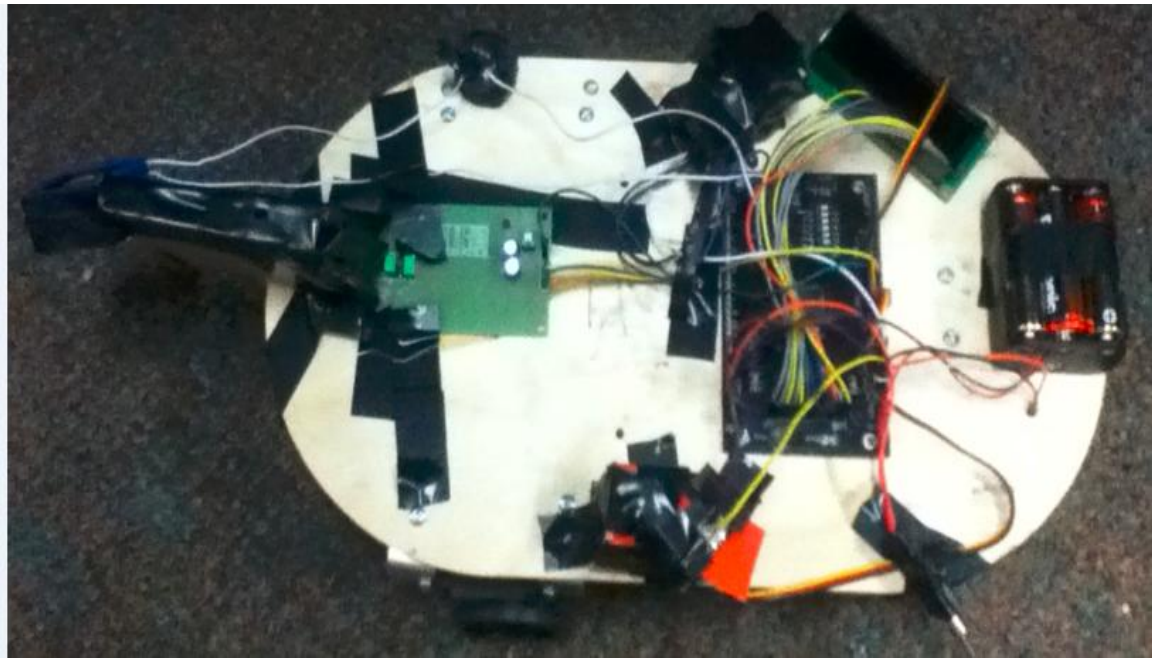

Fig 2. What Fireman will look like

Actuation

Fireman will use two robot servos with attached wheels for actuation. Both of the servos are in the same speed. When the left servo turns forward and the right servo turns back, it will make a right turn. When the left servo turns back and the right servo turns forward, it will make a left turn. In this robot, I use HS-311 as my servo. But this servo can only make a 180 degree. In order to make a 360 degree, I need to hack the servo. Here is the link of tutorials how to hack HS-311 servo: http://www.ge.uiuc.edu/s-burns/HS311hack.

Fig 3. HS-311 Servo

Sensors

To implement the obstacle avoidance function, I choose the Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F. This sensor can detect the distance from 20 to 150cm. I used two sensors in my robot. Mounted on the front of my robot, this sensor allows for an 180 degree field of view for obstacle avoidance.

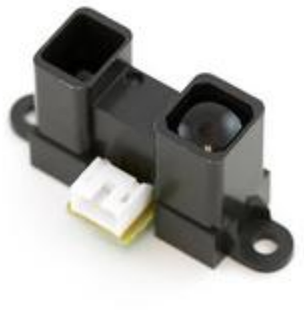

Fig 4. Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F

Also, to detect the flame, I use a Hamamatsu UVTRON R2868 flame sensor. The on/off detector makes use of the photoelectric effect of metal and gas multiplication effect to sense ultraviolet radiation between 185 nm and 260 nm in wavelength. Mounted in the middle and top of the robot, it can let the sensor to detect the flame directly.

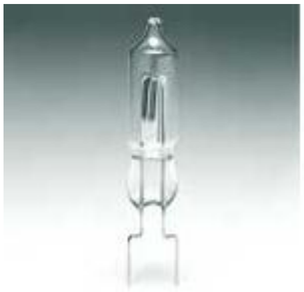

Fig 5. UVTRON R2868 flame sensor

Also, I use R2868 flame detector driver board to connect with the UVTRON flame sensor. The driver board provides the necessary operating voltage for the Hamamatsu flame sensor with a DC input from 3.6- 10V. Under operating conditions, when the sensor detects the ultraviolet radiations emitted from a flame within operating range(up to 7 meters), an 5V pluses will be generated from the Pulse Out Pin, the period and amplitude of the pulses depends on various factors, such as distance of the flame detected and the intensity of the ultraviolet radiations emitted. The stronger the ultraviolet radiations is, the shorter the pluse out period.

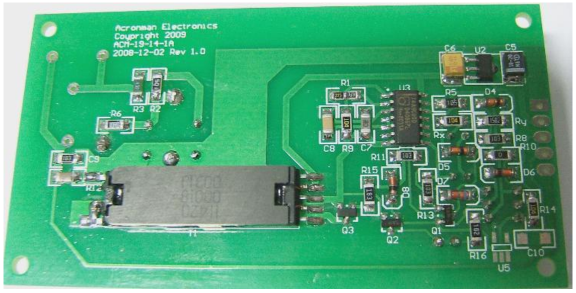

Fig 6. R2868 flame detector driver board

When ultraviolet radiation from a flame is detected by the flame sensor, the driving circuit outputs a square wave with a 10ms pulse width. The microcontroller is programmed so that when a flame is detected, one of the analog port pins is pulled high, indicating the presence of a flame and activating the appropriate behavior. Because I just use one flame sensor, so if the flame is not detected, the robot will spin in space, scanning the area around it. When the flame is detected, the robot will move forward to the flame.

Behaviors

Fireman will exhibit some special behaviors. Its basic behavior is that it can avoid obstacles, which is made by IR sensor. When the sensor detects a close obstacle, it will send some signals to the microcontroller and ask the microcontroller to change the direction so that the robot will turn away from the object and continue on its path. After a period of time, if the flame is not detected, the robot will stop and search the flame around the area for the reason that I only use one UVTRON and its detection view is relatively narrow.

When the flame is found, Fireman will drive towards the flame and it will stop in front of the flame. Then Fireman will extinguish the flame using a fan. In this design, After extinguishing the flame, Fireman will restart the whole routine and begin to look for another flame.

Experimental Layout

Fireman will be placed in a closed space containing at least two bottles of flames. Around them there will be some random obstacles. The fireman will follow search for the fire and after it extinguishes the right flame, it will return to where it comes from.

Conclusion

Fireman is an autonomous fire-fighting robot. It is finished in about three months. Fireman is able to avoid obstacles automatically while looking for a flame. This function is implemented by IR sensor. Also, the flame detection is implemented via UVTRON R2868 flame sensor. When Fireman sees the flame, it will drive forwards toward the flame and once it is located in front of the flame, it will stop. Then Fireman will send a command to let the microcontroller to run the servo, which is connected to the bike pump and blow the flame out.

Indeed, lots of work should be done in the future. First, I will add some Sonar to my robot because it has a more accurate detection. Also, one more flame sensor is needed in this robot in order to broaden the view of the sensor. In that way the flame can be detected more directly and the robot does not have to spin and scan the flame. Also, I would change the platform and make it smaller so that it can make a turn more quickly.

Appendix

// The following is the code I use in the media day.

```
#include <avr/io.h>
#include "PVR.h"

//Initiate variables
int flame=0;
int sonar=1000;
int speed=25;
int count=0;
int time=0;
int range=0;
int lrange=0;
int rrange=0;

void main(void)
{
   //initialization
  xmegaInit();              //setup XMega
  delayInit();              //setup delay functions
  ServoCInit();              //setup PORTC Servos
  ADCAInit();              //setup PORTA analog readings
  lcdInit();              //setup LCD on PORTK
  PORTH_DIR=0b11111110;
 // PORTQ_DIR|=0x01;

// int v;

int crange=0;
int v=0;
int flag = 1;
while(1)
{
        v=PORTH_IN&0x01;
        int i=0;
        int lrange=0;
        int rrange=0;

loop1:
        ServoC1(-90);
        ServoC2(-90);
        for(int time1=0;time1<2000;time1++)
        for (int k=0;k<1000;k++)
                {

                        v=PORTH_IN&0x01;
                        lrange=ADCA2();
                        rrange=ADCA4();
                        lcdInt(v);
                        lcdData(0x01);

                        lcdData(0x01);

              while(lrange>3900)
                        {
                        lrange=ADCA2();
              ServoC2(50);
                        }




              while(rrange>3900)
                        {
                        rrange=ADCA4();
                        ServoC2(50);

                        }


                        if(v==1)
                        {
                                  stop();
                                  goto loop2;
                        }

              }
    stop();

     for(int j=0;j<30;j++)
     { lrange=ADCA2();
       rrange=ADCA4();
crange=ADCA0();
       lcdInt(crange);
       lcdData(0x01);
       forward();
    while(lrange>3900)
                        {
                        lrange=ADCA2();
              ServoC2(-50);
              ServoC1(50);
                        }





              while(rrange>3900)
```

```
                        {
                        rrange=ADCA4();

                        ServoC2(50);
                  ServoC1(-50);
                        }




    }


        }

loop2:
stop();
for(int i=0;i<1000;i++)
for(int j=0;j<1000;j++)
{}
//forward2();
//delay_ms(500);

for(int j=0;j<300;j++)
{ServoC1(20);
ServoC2(-20);
lrange=ADCA2();
          rrange=ADCA4();
     crange=ADCA0();

        while(lrange>3900)
                        {
                        lrange=ADCA2();
                ServoC2(-50);
                ServoC1(50);
                        }





                while(rrange>3900)
                        {
                        rrange=ADCA4();

                        ServoC2(50);
                  ServoC1(-50);
                        }

crange=ADCA0();
 if(crange>3550)
 goto loop5;
  if(crange>3900)
 goto loop4;
}
stop();
for(int i=0;i<1000;i++)
for(int j=0;j<1000;j++)
{         }


while(1)
{
        v=PORTH_IN&0x01;
        int i=0;
        int lrange=0;
        int rrange=0;
                        ServoC1(-10);
        ServoC2(-10);
                        v=PORTH_IN&0x01;
                //      lrange=ADCA2();
                //      rrange=ADCA4();
                //      lcdInt(v);
                //      lcdData(0x01);

                //      lcdData(0x01)



                        if(v==1)
                        {
                                stop();
                                goto loop2;
                        }

}
loop5:
ServoC1(20);
ServoC2(-20);
while(1)
{
crange=ADCA0();
  if(crange>3950)
 goto loop4;
}

 loop4:
 stop();
 delay_ms(30000);


}
```

```
void spinToFlame(void)
{
ServoC2(50);
ServoC1(50);
}

void forward(void)
{ServoC1(90);
ServoC2(-90);
}
void forward2(void)
{ServoC1(-360);
 ServoC2(-360);
 }

void stop(void)
{
ServoC1(0);
ServoC2(0);
}
```