Smoke Eater

An Autonomous Fire Fighting Robot

Written Report
By
David Morneault
EEL 5666
August 9, 2000

Table of Contents

Abstract

Smoke Eater is an autonomous robot that randomly searches for fires and then extinguishes them once they are found. It is equipped with a UV detector for flame detection and IR detectors and bump sensors to avoid all obstacles in its way. It also has CdS cells to find certain areas in the building that it resides in. When it reaches a fire it extinguishes it by spraying it with water, then it continues its search for fires.

Executive Summary

Smoke Eater is an autonomous robot that randomly searches for fires and then extinguishes them once they are found. Smoke Eater is equipped with a UV detector, IR detectors, bump switches, CdS cells, and a water pump to accomplish his goal.

To implement Smoke Eater's flame detection behavior, I used a ultra-violet detector. This allows Smoke Eater to detect a wavelength of light that is only characteristic of fires. The UV detector has a 40? range of detection. It can detect flames up to ten feet. Once it has detected a flame it stops and sprays the fire with water provided by the water pump.

Smoke Eater must roam through a building with unknown obstacles, which he must avoid. IR detectors and IR emitters implement his obstacle avoidance. The IR detectors read the signal sent out by the emitters if there is an object in front of the sensors. The stronger the signal the closer the object is. Once there is an object detected Smoke Eater turns away in a direction that is free of obstacles.

There are also bump switches circling Smoke Eater. These are in case the IR sensors fail. This may seem redundant but a good system has redundancy built into it. Just like the IR sensors when an object is detected then Smoke Eater turns in a direction which is clear of objects.

By integrating all these sensors with The Motorola 68HC11 I was able to build an autonomous robot that would roam a building, search for fires and extinguish them.

Introduction

Background

Household and industrial fires cause millions of dollars in damage each year. Most fires start out fairly small and could be put out easily with the right equipment or notification of the fires presence. The most common system used today to extinguish fires in industrial type buildings is a sprinkler system. This system can be affective, but it also damages most of the surrounding goods. An autonomous robot would only spray the area on fire, therefore minimizing the damage. With an autonomous fire-fighting robot like Smoke Eater around an industrial fire or a house fire could be stopped at its early stage therefore preventing major damage caused to the building or surrounding area and saving a lot of money.

Scope

Smoke Eater is an autonomous robot, which detects fires and extinguishes them. It was designed to navigate its way through a simulated miniature house with simple obstacles place throughout the house. It's goal is to find a candle placed at random in the house and put out the flame then find its way back to "home base".

Specifications

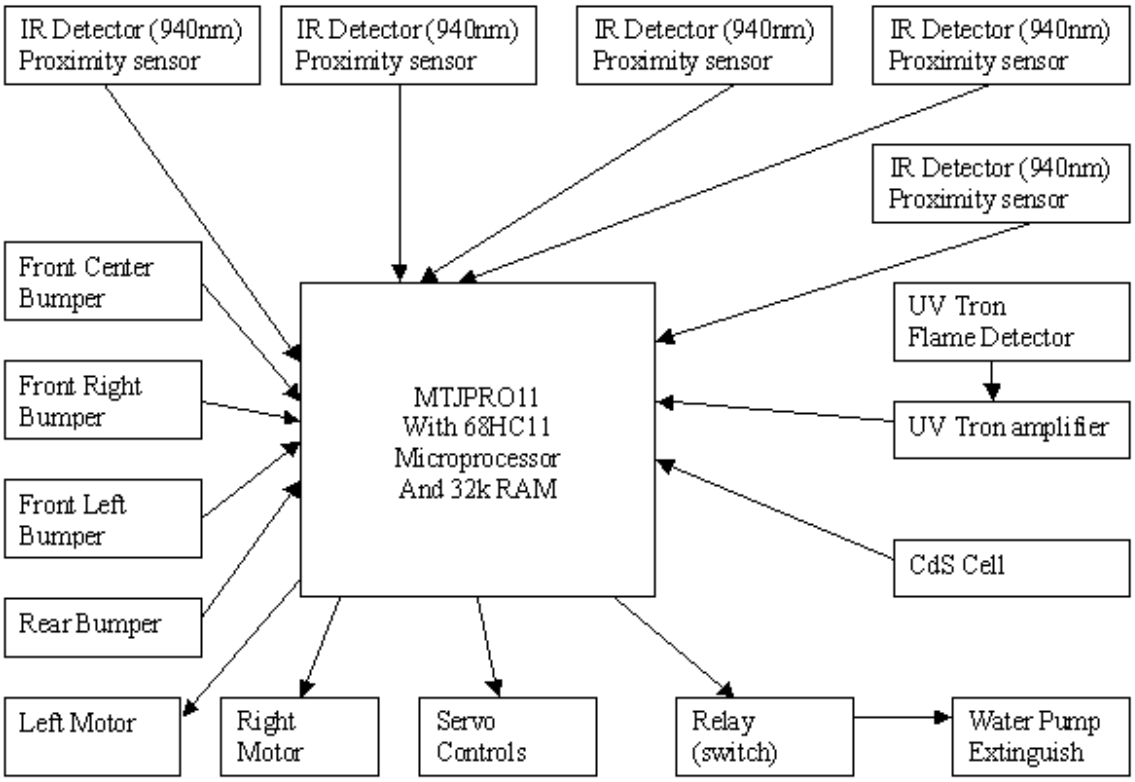
The robot cannot be bigger than 12.5 inches by 12.5 inches. It cannot use dead reckoning either. It also must be completely autonomous.

Integrated Systems

Description of System

Smoke Eater is designed to use Motorola's 68HC11 microprocessor on a MTJPRO11 board designed by Mekatronics. Included in my robot are four IR emitters and five detectors for proximity detection and avoidance, ten bump switches for collision avoidance, a UV Tron made by Hamamatsu for flame detection and a DC water pump and reservoir to extinguish the fire. It will also have two CdS photoresistor cells to detect for a white ring on the ground known as "Home Base" in the fire-fighting robot contest. Smoke Eater will be driven by two hacked Futaba servos. There will also be another servo to move the nozzle of the extinguisher up and down the fire to eliminate any inaccuracy in the extinguisher.

Operation



The robot uses all eight pins from the analog port. It also uses four pins from the input port A.

Mobile Platform

Scope

Smoke Eater must orient himself through a simulated house or something similar, therefore he must avoid foreign objects and not get stuck on them. I choose to use a Talrik platform designed by Mekatronics. I choose a round platform because it will

maneuver much better than a rectangle or square robot, or any other shape. Also this platform is large enough that it will accommodate for all of my sensors, motors, and water for the extinguishing system.

Specifications

I placed the wheel axis in the center of the robot, this allows for more accurate turns in a round platform. The motors are placed on the underside of the robot along with the IR emitters and the wheels. I was able to find wheel hubs that screw into the servo horn. This made sure that the wheels were directly in the center of the horn and were aligned. By screwing the wheels into the servo horn it made them more secure and easier to install and uninstall them on the robot. I placed the MTJPRO11 suspended over the hole in the middle of the Talrik platform for easy access from both the bottom through the hole and the top of the robot. The Talrik design will also allow me to place my extinguisher nozzle and servo in front of the robot.

Actuation

Scope

Smoke Eater will need to carry a decent amount of weight considering all of its on board hardware. The main pieces weighing the robot down are the two battery packs, water reservoir, water pump and three servos. The third servo is placed near the front of the robot to perform the task of moving the extinguisher nozzle up and down as it spays the fire. The water pump must pump water at least a foot away to extinguish the fire.

Specifications

There are two driving wheels attached to a 100-oz./in. hacked servo (Futaba S9303) each. There is also a caster wheel place towards the back of the robot for balance. I also installed a stabilizing bar in the front so the robot would not get stuck leaning forward, which it sometimes did during operation. I used a hacked servo because it already comes with motor drivers. The hack was performed by removing the tab inside the gearbox that stopped the horn from turning in 360? of motion.

The servo (Futaba S148) that moves the extinguisher nozzle will move approximately 40 degrees. This corresponds to 3000 to 3400 in my program where I control the servo. This will move up and down for two seconds.

The DC water pump that I used is from a car windshield washer pump. I choose this because it uses DC power and would be easy to implement on the robot.

Sensors

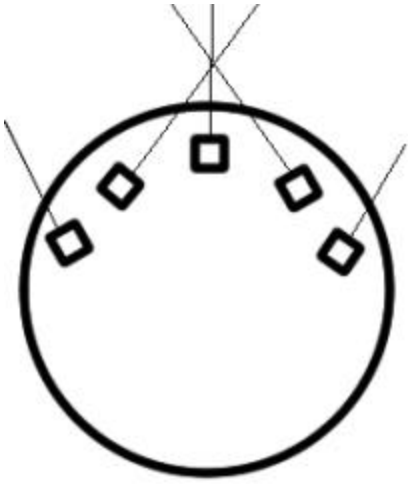
Scope

Smoke Eater must avoid the walls and any objects in its path while it must also detect if a fire is present. It must also detect a piece of white tape on the ground, which will be place at the candle and at the home base. The robot must start out in home base and must put the candle while part of it has crossed the white line by the candle; this prevents a

robot from spraying the fire from far distances. It will detect the white tape with CdS photoresistors.

IR Emitters/Detectors

To avoid any objects in its path I have placed four IR emitters and five IR detectors along the front of the robot creating a semi-circle of IR detectors. Here is the layout I used:



This layout gave the robot a better of what was in front of him by crossing the two IR detectors. Also I put two more on both sides to detect for anything immediately on its side that the others would not catch.

Bump Detectors

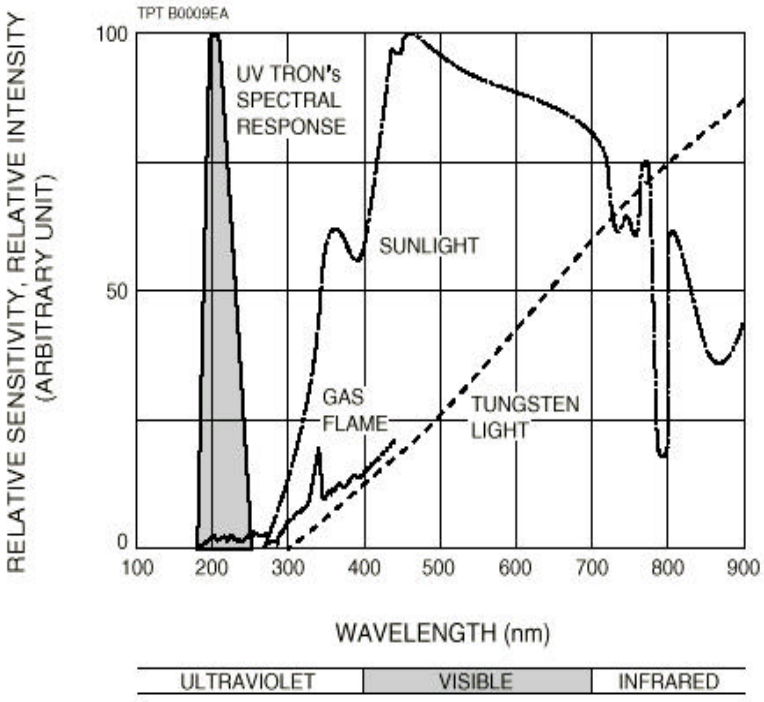
There are bump switches, which will detect if the robot has run into anything. When something is in its way or it bumps into something the robot will turn in a random direction and precede forward. Here are the readings I read from the bump switches:

Location of Switch	Analog Value
Front Left	76-78
Front Center	43-45
Front Right	19-20
Rear	123-125

UV Detector

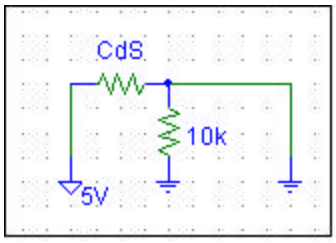
For the fire detection sensor I am using a UV Tron part number R2868 from Hamamtsu. I also bought the circuitry that they sell along with the UV Tron, which is part number C3704. This detects ultraviolet light in the range of 180 nm to 260 nm. It is located on top of the bridge of the robot. I positioned the UV Tron so that it would only sense flames at a short distance. I did this by placing the UV Tron flat and covering it with a tube. This allowed light to enter the UV Tron only from the tip where the maximum range is about ten feet. I also removed the voltage regulator (NEC 78L05 806D) from the circuit board on the C3704. This increased the voltage to the chips on the board to five volts, which made them respond a lot quicker. Otherwise there was a 0.7-volt drop from the regulator, which gave less voltage to the chips on the board therefore reducing the response time. I also put a 1 ?F capacitor on the circuit board (C3704) to increase the pulse rate of the signal to one second. Otherwise the program would not always catch the signal. Here is the spectral response of the UV Tron:

Figure 1: UV TRON's Spectral Response and Various Light Sources



Home Base / Candle Detector

The CdS photoresistors change resistances according to what color is being reflected to the resistor, this will detect the white tape on the ground in front of the candle and around the home base. I used a 10k ohm resistor in my circuit for the CdS cell. Here is the circuit I used:



Behaviors

Obstacle Avoidance

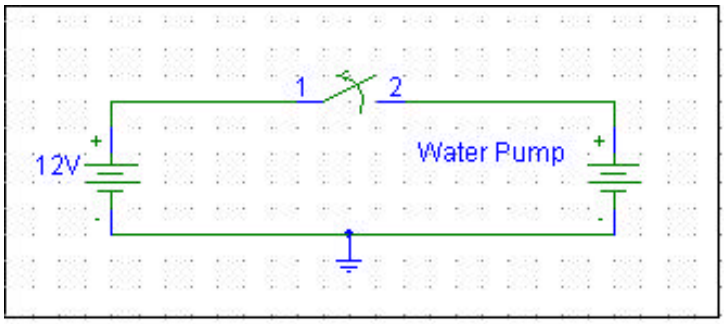
Smoke Eater will avoid running into objects by using the IR emitters and detectors. As a backup method of obstacle avoidance there are bump sensors placed around the body. When an obstacle is detected then Smoke Eater will then turn in a direction that is clear of objects and move forward again.

Fire Detection

In this behavior Smoke Eater will move forward for three seconds then turn in a full circle scanning the room for fires then move forward again. When a flame is detected then it will move in the direction of the flame until it detects the white tape in front of the candle. When the tape is found then it will extinguish the fire.

Fire Extinguishing

When the robot is in front of the fire and is on the white tape then it will spray water out of the nozzle while the servo is moving the nozzle up and down. The water pump only stays on for two seconds, then it looks to see if the fire is out, if not it keeps spraying, though if the fire is out then it searches for more fires to extinguish. I used a Reed Relay part number 273-075 from Radio Shack. This relay acts as a switch triggered by five volts. This enabled me to use pin 6 on port A from the 68HC11 to turn it on. When the signal went high it connected the circuit below:



Experimental Layout and Results

Once I finished building the platform I started to test Smoke Eater. I first edited the sensory testing program from the MIL lab and edited it to include all of my sensors. This allowed me to see all five IR detectors the CdS cell, the bump sensors and the UV Tron at once. I then recorded all the values of the sensors, which I needed to write my program. I used a white piece of cardboard to measure the IR detectors since the area where I would be testing it would have white walls. I also measured the CdS cell against a white background and a black background.

The UV Tron was a little more difficult. I had to write a separate program that would simulate what the robot would do during its routine. Then I measure how fast it took to detect flames at different distances. After recording that data I rewrote the program routine to according to the time it took to detect the flame and react so that it would be exactly centered in front of the flame when it detected the candle. The variable I used was the length of the covering on the UV Tron. By varying this length I was able to adjust the angle of detection, which was directly related to the response time.

Conclusion

Building Smoke Eater was challenging but it was also a lot of fun. This project incorporated many of the skills I have learned over the past few years in my electrical engineering classes and it was exciting to put them to use. If I were to build it again I would definitely try to build it on an aluminum platform because the wood that I used became a little to light weight towards the end when it has everything loaded on it. I am planning on entering the Trinity College Fire Fighting contest next year(2001). To make Smoke Eater more accurate in detecting and pinpointing flames I want to integrate two UV Trons on the platform. I would recommend this class to any engineering student because of the hands-on experience is like no other class I have taken.

Documentation

Mekatronix. www.mekatronix.com Infrared emitters and detectors, MTJPRO11 board, and information on servo hack and infrared hack.

IMDL. www.mil.ufl.edu/IMDL. Information on the process of the design and building of a robot.

Radio Shack. Reed Relay, and extra circuit board.

Hamamatsu Corporation. www.usa.hamamatsu.com. UV Tron (R2868), circuitry for UV Tron (C3704).

George's Hardware. 34th Street and University Ave. Wheel hubs and wheels.

Tower Hobbies. www.towerhobbies.com. 100oz./in. servos (Futaba S9303), and 22oz./in. servo (Futaba S148).

Pep Boys. Newberry Road and 75th Street. Water pump.

Wal-Mart. Butler Plaza. Rechargeable batteries and water reservoir.

Appendix

```
/*****************************************************************************
*
*       Title:          Smoke15
*       Programmer:     David Morneault
*       Date:           July 26,2000
*       Version:        1.5
*
*       Description: Demo version.  This program will randomly search a
*       house, and about every 3 seconds will scan the room spinning 360
*       degrees.  When a flame is detected then it will extinguish it for 2
*       seconds.  If the flame is still present it will spray it again if
*       not then it will continue to search for flames.
*
*
*****************************************************************************/

/***************************** Includes ******************************/

#include <analog.h>
#include <clocktjp.h>
#include <motortjp.h>
```

```
#include <servotjp.h>

#include <mil.h>

#include <hc11.h>

#include <serialtp.h>



/************************ End of Includes ******************************/

/************************** Constants *********************************/



#define IRE_ON  *(unsigned char *)(0x7000) = 0x07

#define IRE_OFF       *(unsigned char *)(0x7000) = 0x00



#define LEFT_MOTOR 1

#define RIGHT_MOTOR 0

#define BUMPER analog(0)

#define CENTER_IR analog(4)

#define RIGHTM_IR analog(6)

#define LEFTM_IR analog(5)
```

```
#define RIGHT_IR analog(1)

#define LEFT_IR analog(7)

#define UV analog(2)

#define CDS1 analog(3)


/************************* End of Constants ***************************/



void main(void)

/****************************** Main ***********************************/

{


/***********Variables****************/


int       irr;

int       irlm;

int       irc;

int       irrm;

int       irl;

int       rspeed;

int       lspeed;

int i;

int num;
```

```
int k;
int       l;
int       lm;
int c;
int rm;
int r;
int m;
int rot;
int p;

/**************Initializations***************/

        init_analog();
        init_clocktjp();
        init_motortjp();
        init_servotjp();


        IRE_ON;

/*****************IR detector levels***********/

        lm = 121;
```

```
        l = 121;
        c = 128;
        rm = 121;
        r = 121;



        i = 1;
        num = 1;

/***************Start of infinite loop***************/

while(1)
{

/*First check for fire*/


        CLEAR_BIT(PORTA,0x40)           /*turn off water pump*/

/***************Flame Detection**********************/

        if (UV > 100)
```

```
{

motorp(RIGHT_MOTOR, 0);
motorp(LEFT_MOTOR, 0);

SET_BIT(PORTA,0x40)                    /*turn on water pump*/

for (p=1; p < 3; p++)          /*loop for controlling the front servo*/
{



rot = 3000;

for (m=1; m < 73; m++)
{
rot = rot + 5;

servo(1,rot);
wait(8);
}

rot = 3400;
```

```
        for (m=1; m < 73; m++)
        {
        rot = rot - 5;

        servo(1,rot);
        wait(8);
        }
        }                                                   /*end servo controlling
loop*/

        CLEAR_BIT(PORTA,0x40)          /*turn off water pump*/

        }/*end if UV > 100*/

/**********************End Flame Detection******************/

        CLEAR_BIT(PORTA,0x40)          /*turn off water pump*/

        i++;
        num = (i % 20);

        if ( num == 0)
        {
```

```
/**********************Start Flame Detection Sequence******/

        for (k=1; k < 14; k++)
        {

        if (UV > 100)
        {




        motorp(RIGHT_MOTOR, 0);
        motorp(LEFT_MOTOR, 0);

        SET_BIT(PORTA,0x40)                    /*turn on water pump*/

        for (p=1; p < 3; p++)
        {



        rot = 3000;

        for (m=1; m < 73; m++)
        {
```

```
rot = rot + 5;

servo(1,rot);
wait(8);
}

rot = 3400;

for (m=1; m < 73; m++)
{
rot = rot - 5;

servo(1,rot);
wait(8);
}
}

CLEAR_BIT(PORTA,0x40)                    /*turn off water pump*/

}/*end if UV > 100*/

CLEAR_BIT(PORTA,0x40)                    /*turn off water pump*/
```

```
        motorp(RIGHT_MOTOR, -100);
        motorp(LEFT_MOTOR, 100);

        wait(60);

        motorp(RIGHT_MOTOR, 0);
        motorp(LEFT_MOTOR, 0);

        wait(200);


        }/*end for loop*/

        }/*end if num == 0 */

/**********************End Flame Detection Sequence************/


        motorp(RIGHT_MOTOR, 100);
        motorp(LEFT_MOTOR, 100);

        CLEAR_BIT(PORTA,0x40)                   /*turn off water pump*/
```

```
/* Second check IR detectors */


        irlm = LEFTM_IR;
        irl = LEFT_IR;
        irc = CENTER_IR;
        irrm = RIGHTM_IR;
        irr = RIGHT_IR;


        if (((irlm > lm) || (irr > r)) && (irc > c) && ((irrm > rm) || (irl > l)))
        {
                lspeed = -100;
        }

        else if(((irlm > lm) || (irr > r)) && (irc > c) && ((irrm < rm) || (irl < l)))
        {
                lspeed = -100;
        }

        else if(((irlm > lm) || (irr > r)) && (irc < c) && ((irrm < rm) || (irl < l)))
        {
```

```
        lspeed = -100;
}

else
{
        lspeed = 100;
}

if (((irlm > lm) || (irr > r)) && (irc < c) && ((irrm > rm) || (irl > l)))
{
        rspeed = -100;
}


else if(((irlm < lm) || (irr < r)) && (irc > c) && ((irrm > rm) || (irl > l)))
{
        rspeed = -100;
}

else if(((irlm < lm) || (irr < r)) && (irc > c) && ((irrm < rm) || (irl < l)))
{
        rspeed = -100;
}
```

```
else if(((irlm < lm) || (irr < r)) && (irc < c) && ((irrm > rm) || (irl > l)))
{
        rspeed = -100;
}

else
{
rspeed = 100;
}

motorp(RIGHT_MOTOR, rspeed);
motorp(LEFT_MOTOR, lspeed);

wait(100);

motorp(RIGHT_MOTOR, 100);
motorp(LEFT_MOTOR, 100);


if (UV > 100)
{
```

```
motorp(RIGHT_MOTOR, 0);
motorp(LEFT_MOTOR, 0);

SET_BIT(PORTA,0x40)

for (p=1; p < 3; p++)
{



rot = 3000;

for (m=1; m < 73; m++)
{
rot = rot + 5;

servo(1,rot);
wait(8);
}

rot = 3400;

for (m=1; m < 73; m++)
{
```

```
        rot = rot - 5;


        servo(1,rot);
        wait(8);
        }
        }

        CLEAR_BIT(PORTA,0x40)


        }/*end if UV > 100*/



/* Third check the bump sensors */


        if ((BUMPER > 40) && (BUMPER < 50))
        {
                rspeed = -100;
                lspeed = 100;
        }

        else if ((BUMPER > 18) && (BUMPER < 30))
        {
```

```
        rspeed = 100;
        lspeed = -100;
}

else if ((BUMPER > 73) && (BUMPER < 99))
{
        rspeed = -100;
        lspeed = 100;
}

else
{
        rspeed = 100;
        lspeed = 100;
}


motorp(RIGHT_MOTOR, rspeed);
motorp(LEFT_MOTOR, lspeed);

wait(100);
```

```
        motorp(RIGHT_MOTOR, 100);
        motorp(LEFT_MOTOR, 100);



}/*end while loop for (1) */



}
/************************End of Main*********************************/
```