

Fire-Extinguishing Autonomous Robot



Smokey the Bot

EEL5666C – Intelligent Machine Design Lab

Matthew Bellman

Instructors: A. Antonio Arroyo, Ph.D.; Eric M. Schwartz, Ph.D.

TAs: Mike Pridgen, Thomas Vermeer

Executive Summary

In order to learn the basics of robotics design, I chose to build my own autonomous robot in the Intelligent Machine Design Lab (IMDL) at the University of Florida (UF). My robot's task was simple – autonomously find and extinguish a flame. The application for robots like “Smokey the Bot” is obvious, but robotic firefighters have yet to become commonplace in local fire departments. The autonomous platform developed in this course introduced me to hands-on robotics design and creation, and may serve as a platform for further investigation into firefighting robots. “Smokey” is composed of a cylindrical wooden frame with two motors positioned symmetrically, allowing for differential drive capabilities. Obstacle detection and avoidance is achieved by forward mounted sonar range finders, which communicates with a Pridgen Vermeer microcontroller via ADC. The microcontroller served as Smokey's brain, and communicated commands to the motor driver, toy fire truck sirens and flame-extinguishing fans while receiving information from the sonar and a Hamamatsu UVTRON flame detector.

One of Smokey's basic behaviors is obstacle detection and avoidance. Obstacle detection begins when Smokey is activated via two toggle switches, one for each power supply. The smaller, 6-AA NiMH battery pack provides power to the microcontroller, and the larger, 8-AA NiMH pack powers the motor driver, flame detector, and fans. The ultrasonic range finding sonar sensor detects obstacles, and sends an analog signal to the microcontroller's ADC port. Smokey's behavior depends on the value read from the sensor, and it provides sufficient obstacle avoidance at any desired threshold between 1 – 5 feet away from the sensor. Although only one sonar sensor was used, the robot's cylindrical frame and differential drive helped to avoid collisions by effortlessly turning away from obstacles.

Smokey's other primary behavior is flame finding/extinguishing. While obstacle avoiding, Smokey is programmed to stop every 5 seconds to turn and scan the surrounding area for flames. The UVTRON flame detector and driving circuit are mounted on top of the robot, and are equipped with a cardboard tube, aligned with the front of the robot, that the flame sensor sits inside of. Electrical tape covers the forward facing end, but a slit has been left open to allow only a narrow field of view for the sensor, directly in front of the robot. This makes finding the flame easy for Smokey – it must simply detect the flame and drive forward until it encounters the flame source (candle). Once the flame detector “sees” a flame, Smokey's sirens and lights activate (relay switch) and it drives quickly and in a straight line to the flame. Occasional misses occurred when Smokey drove forward but was not aligned with the flame. Once the sonar detects the candle, Smokey stops, activates the forward-facing computer fans by switching a relay, and blows the candle out. When the fans stop, the sirens and lights deactivate and Smokey begins searching for another flame.

Table of Contents

Executive Summary	2
Abstract	4
Introduction	5
Integrated System	6
Mobile Platform	7
Actuation	8
Sensors	11
Behaviors	13
Conclusion	14
Appendix	15

Abstract

Autonomous robots that can complete hazardous tasks are an important step forward in minimizing the risk that humans must take during emergency situations. This paper proposes an autonomous flame-extinguishing robot that can navigate an area while detecting obstacles and avoiding collisions, search for a flame using a flame sensor, and extinguish the flame with a fan. These tasks are accomplished by an integrated system of components including motors, sensors, and a microcontroller, all assembled together on a mobile platform. Included in this paper are descriptions of how the system functions, an Appendix of AVR code, and suggestions for continued work.

Introduction

After abandoning earlier, more complex robot project ideas, I decided on the simple task of extinguishing a flame. I have seen videos of flame-extinguishing competitions, in which the competing robots autonomously navigate a series of rooms and corridors to locate a flame and extinguish it. Wanting to do something similar, I opted for a more robust design, with the intent of finding and extinguishing the flame as fast as possible. My vision for Smokey's final performance was a robot that could navigate an area quickly and smoothly while scanning for a flame. I hoped that Smokey would be able to track and quickly extinguish multiple flames.

This paper contains the components, functions, and theory behind Smokey's behaviors. Detailed descriptions of mobile platform design and sensor capabilities are also described. The Appendix contains the code that ran on Smokey's microcontroller and delegated its functions.

Integrated System

The robot uses an integrated system of components to achieve its objective. The system's organization is shown in Figure 1.

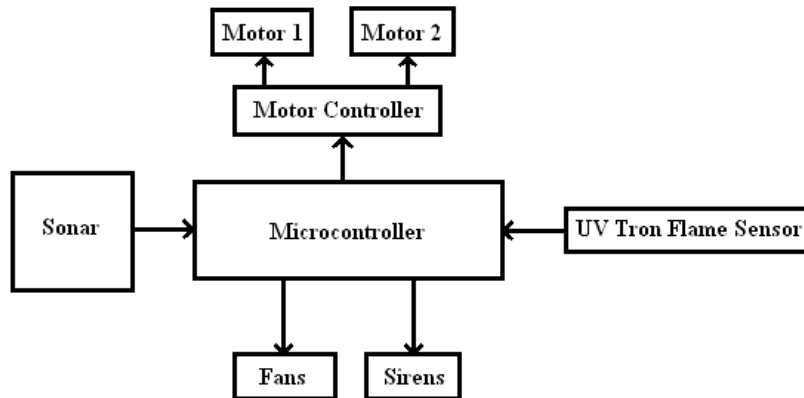


Figure 1: Shown in the figure are the various components of the robot, with arrows indicating the direction of information flow between components.

Descriptions of how each component functions within the system are given below:

- **Microcontroller** – A Pridgen Vermeer Robotics Xmega128 Board controls the robot. The board receives sensor data and outputs pulse-width modulated (PWM) signals to drive the motors appropriately. When the robot finds a flame, the microcontroller sends an “ON” signal to the flame extinguisher, and once the flame is out, turns it off.
- **Motor controller** – Since the microcontroller itself cannot power the 12V electric motors, a Sabertooth 2X5 Regenerative Dual Channel Motor Controller (lynxmotion.com) allows the motors to be driven by a PWM signal from the microcontroller.
- **Sonar** – The robot detects objects with three Maxbotix LV-EZ1 ultrasonic range finders. The sensors communicate data through the A/D ports of the microcontroller, which reads a voltage proportional to the distance measured from each sensor.
- **Flame sensor** – A Hamamatsu UVTRON R2868 flame sensor is mounted on top of the robot. The sensor’s periphery is physically blocked so that it has a very narrow field of view. This sensor communicates with the microcontroller through a driving circuit, which outputs a 5V square wave with a 10 ms pulse width to the control board.
- **Flame extinguisher** – Once the flame sensor indicates that a flame is near, the microcontroller turns two forward mounted fans to blow the flame out.
- **Lights and sirens** – When a flame is detected, lights and sirens from a toy fire truck activate. When the fans shut off and the flame is out, the lights and sirens deactivate.
- **Battery packs (not pictured)** – Two rechargeable NiMH battery packs power all onboard electronics.

Mobile Platform

All of the robot’s components are carried and supported by a simple, cylindrical frame made of wood. The motor assembly is mounted to the bottom of the frame along with two stick-on slides

that help support the robot. The microcontroller is contained within the structure of the robot. The sonar sensor is mounted on the frame's forward perimeter, allowing for almost 180° of obstacle detection in front of the robot. The flame sensor and driving circuit are mounted to the top of the mobile platform, and positioned such that the sensor's field of view is directly in front of the robot.

The “rolling chassis,” that is the frame and motor assembly without any sensors or other electronics, of the robot was designed using SolidWorks. The completed SolidWorks assembly is shown below in Fig. 2. The platform design was kept as simple as possible to minimize potential frustrations. The circular outer perimeter of the robot allows it to scan the surrounding area uniformly by simply spinning in place.

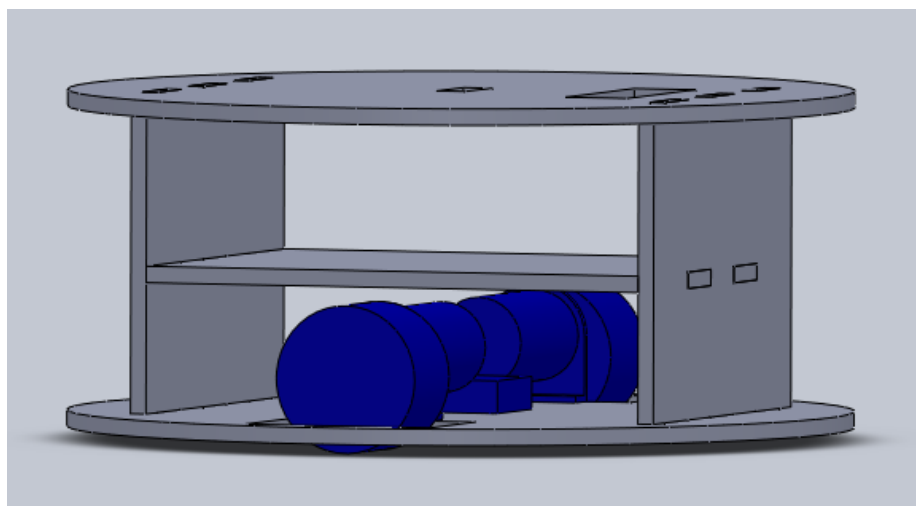


Figure 2: SolidWorks assembly of “rolling chassis,” with gray parts and blue parts being the frame and motor assemblies, respectively.

A scroll saw was used to cut the five assembly pieces out of one-half inch thick oak board. Fabrication by hand took only a few hours to do and cost little. The wooden pieces were fastened together by simply cut tabs, making assembly and disassembly quick and easy. Holes were cut into various faces of the platform to hold sensors in place and to allow wires to pass through the structure.

Actuation

Two spur gear head motors purchased from Lynxmotion.com drive the robot towards its objective. The motors are collinear and their axis passes through the center of the mobile platform, so that when both motors are driving at the same speed, the robot moves forward in a

straight line. To turn, a differential drive system is employed, where the rate and direction of turn is determined by the magnitude and sign of the difference between motor speeds. Referring to Figure 3, if the speed of Motor 1 > Motor 2, the robot will turn left, and if the speed of Motor 1 < Motor 2, the robot will turn right. When a turn of specific degree is required, the motor speeds are set so that the robot turns at a constant rate for an amount of time depending on the degree of the turn (about 30°/s for normal turn speed).

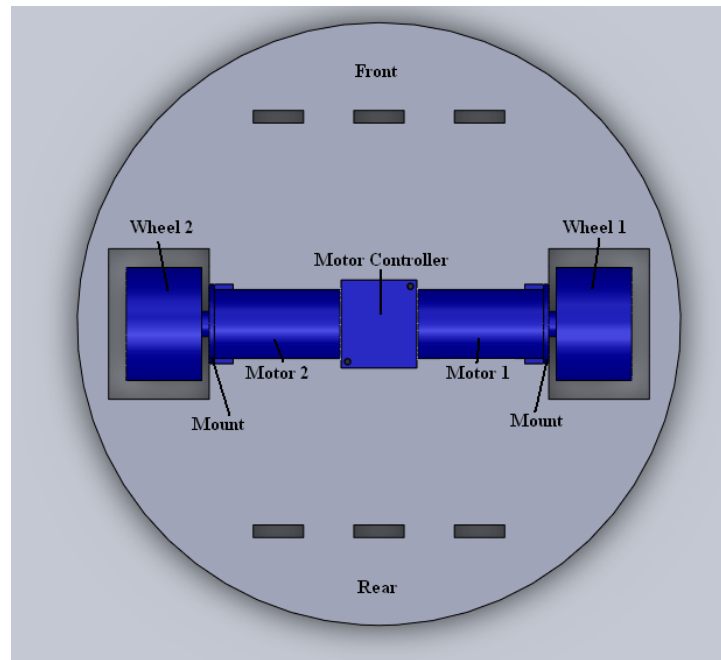


Figure 3: Mobile platform base (viewed from above in SolidWorks) with mounted motor assembly: motor controller, motors, mounts and wheels.

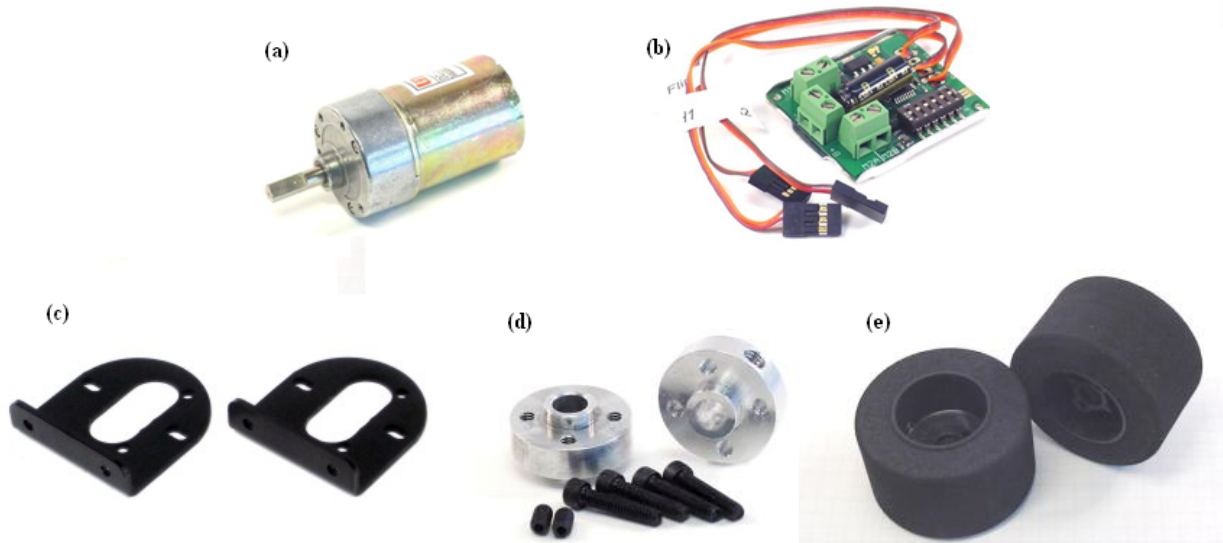


Figure 4: Motor assembly: (a) GHM-13 spur gear head motor, (b) Sabertooth dual channel motor controller, (c) aluminum motor mounts, (d) 6 mm aluminum hubs, (e) 2.5 inch diameter neoprene foam tires. (www.lynxmotion.com)

At a rated load of 1.50 kg-cm, the GHM-13 motors rotate at 132 RPM, and along with 2.5" diameter foam wheels drive the robot at a speed of about 1.5 feet per second. These motors carry built-in gearboxes, giving the motor a 50:1 gear reduction and a stall torque of 16.7 kg-cm with a corresponding stall current of 3.4 Amps. These specifications and more are shown in Figure 5.

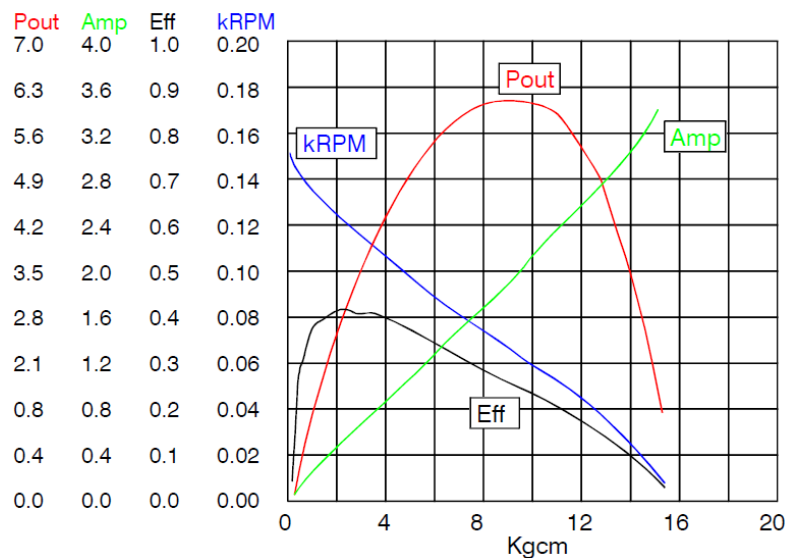


Figure 5: GHM-13 12vdc gear head motor specifications -- Power output (W), Pout (red); Current draw (A), Amp (green); Motor speed (kRPM), kRPM (blue); Efficiency, Eff (black). (www.lynxmotion.com)

Sensors

A Maxbotix LV-EZ1 ultrasonic range finding sensor (Fig. 6) sends out high frequency (42 kHz) sound waves that bounce off of objects and return to the sensor. Mounted on the front of the robot, this sensor allows for an almost 180° field of view for obstacle detection in front of the robot. The manufacturer reports that the ultrasonic range finders can detect objects from 6 – 254 inches away with 1 inch resolution. Range readings are made at 20 Hz, with a serial (RS232), analog, or PWM output. The analog output of the sensor gives a voltage proportional to the distance measured ($V_{cc}/512$ inch), although the actual distance measured is irrelevant for this application.

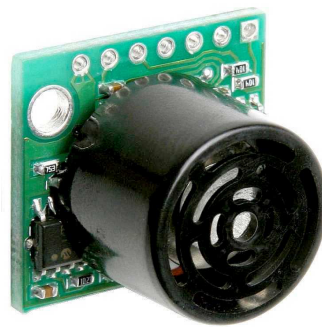


Figure 6: Maxbotix LV-EZ1 ultrasonic range finding sensor. (www.maxbotix.com)

The performance of the Maxbotix LV-EZ1 with a 3.3V supply fell below expectations; although, better performance was achieved with a 5V supply, taken from the PWM ports of the PVR microcontroller. When the analog output of the sensor is connected directly to the ADC port of the microcontroller, the values read from the port range from 277 – 3605. If an object is detected within 12 inches of the sensor, the value read from the ADC port will be 277. Values of 384 – 405 are read from 12 inches to about 36 inches out from the sensor. The values continue to increase with range, but the quoted resolution of 1 inch is not accurate. Realistically, these sensors have a resolution about 1 foot, and an approximately 160° horizontal field of view. Despite the lack of resolution in measuring distance, this ultrasonic sonar sensor provide enough information to detect almost any obstacle within 1 foot of the robot's front side. I only used one sensor, but with its field of view and mounting position on my robot, it was able to provide successful obstacle avoidance, and very rarely touched any kind of obstacle. The microcontroller is programmed to activate the robot's obstacle avoidance behavior when the sonar sensor reports a value of 400, indicating that an object is close and a collision is probable.

A Hamamatsu UVTRON R2868 flame sensor (Fig. 7) completes the objective of finding the flame. The ON/OFF detector makes use of the photoelectric effect of metal and the gas multiplication effect to sense ultraviolet radiation between 185 nm and 260 nm in wavelength, the same range of ultraviolet wavelengths emitted by flames. The UVTRON is mounted on top of the robot, on a cardboard tube with some electrical tape covering most of one end narrows the

sensors field of view enough for a line-of-sight detection. This lets the robot know that the flame is directly in front of it when detected. Unfortunately, I was not able to narrow the field of view enough and this caused problems. Smokey detected the flame sooner than it should have, and caused it to miss by a few inches. After a lot of testing, I found that the best distance for detection of this kind was about 2 m. Any farther than that and the sensor would not receive enough fire-light to signal detection of the flame.



Figure 7: Hamamatsu UVTRON R2868 Flame Sensor. (www.hamamatsu.com)

The R2868 requires a large supply voltage (400V maximum), so a Hamamatsu UVTRON C3704 driving circuit (Fig. 8) is used to power the sensor. When the sensor and the driving circuit are used together, the flame of a cigarette lighter can be detected from about 5 meters away.

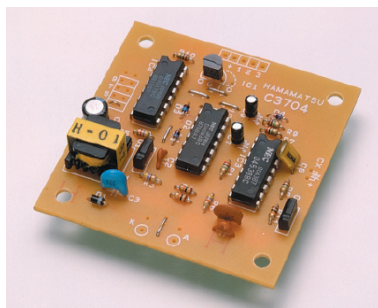


Figure 8: Hamamatsu UVTRON C3704 Driving Circuit. (www.hamamatsu.com)

When ultraviolet radiation from a flame is detected by the flame sensor, the driving circuit outputs a square wave with a 10 ms 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. Since the field of view of the sensor is directly in front of the robot, the robot can spin in place, scanning the area around it, until a flame is detected. When a flame is detected, the robot must simply drive forward, keeping the flame directly in front of it until it reaches the target.

Behaviors

Smokey's basic behavior is obstacle detection and avoidance, made possible by its range finding sonar sensor. When the sensor detects a close object, the microcontroller commands the motors to change speed so that the robot turns away from the obstacle and continues on its path. For example, if Smokey drives forward toward a wall, it will stop one motor while maintaining speed on the other, causing a turn away from the wall. After a set amount of time, Smokey will stop and spin in place, scanning the surrounding area for a flame. If Smokey sees a flame, the code (Appendix) switches to flame finding.

The UVTRON flame sensor is responsible for the flame finding behavior. If a flame is detected, Smokey is assumed to be facing the flame, because of the sensor's field of view, and drives forward toward the flame. Also, when a flame is detected, Smokey's lights and sirens are activated. During this behavior, an obstacle detected by the middle sonar sensor is assumed to be the flame source (candle or lighter), so that if the robot detects an object directly in front of it while still being able to see the flame, then the flame has been found and Smokey is ready to extinguish it. Once in position, the microcontroller activates the forward-facing fans. The fans run for a predetermined amount of time, then turn off, along with the lights and sirens. Smokey then restarts the whole routine and begins searching for another flame.

Conclusion

Smokey the Bot is a successful autonomous fire-extinguishing robot, made in only a few short months out of wood, wires and electronics. Smokey is able to autonomously avoid obstacles while searching for a flame through use of an ultrasonic range finding sonar sensor, and can detect a flame with a flame sensor made for fire alarms by Hamamatsu. When Smokey sees the flame, its sirens and lights are activated and its motors speed it forward towards the flame in the manner of a fire truck rushing to save a burning building. Once it has found the flame and stops in front of it, Smokey's forward-facing computer fans activate and blow the flame out. Smokey usually only fails because of an early flame detection when searching, which results in a miss of a few inches.

Future work to improve this robot would include implementation of more accurate range finders, multiple servo-mounted flame detectors for more sophisticated flame tracking, and a motor-encoder system for dead reckoning. I would also redesign the mobile platform to be lighter and more compact, mainly to minimize motor power requirements. Later in the semester, I dismantled a paintball gun and devised a way to have Smokey fire CO₂ to extinguish the flame, but unfortunately, time restraints cancelled implementation.

Smokey the Bot successfully completed the simple task of extinguishing a flame autonomously.

Appendix

```
// The following code was created by Matthew Bellman for IMDL Spring 2010 at
// the University of Florida. This code controls Smokey the Bot's autonomous
// obstacle detection and avoidance and flame finding and extinguishing.
// This was the code used on Media Day for a successful demonstration.
```

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

```
//Initiate variables
int flame = 0;
int sonar = 0;
int speed = 25;
int count = 0;
int time = 0;
```

```
//Main function
void main(void)
```

```
{
    xmegaInit();           //setup XMega
    delayInit();           //setup delay functions
    MotorClnit();          //setup PORTC Servos
    ADCAInit();            //setup PORTA analog readings
    lcdInit();             //setup LCD on PORTK
    PORTB_DIR = 0x00;      //set B0 (flame) as input
    PORTQ_DIR |= 0x01;     //set Q0 (LED) as output
    PORTJ_DIR |= 0x02;     //set J1 (fans) as output
    PORTH_DIR |= 0x01;     //set H0 (lights/sirens) as output
```

```
    top: //restart point after one flame extinguishing cycle
```

```
    speed = 25;            //reset speed to initial value
    LED(1);                //LED ON when active
    delay_ms(2000);        //wait 2 seconds after power to begin
```

```
    while(1) //routine start
    {
```

```
        checkSonar(); //returns sonar range readings
        checkFlame(); //returns flame (1 if no flame, else 0)
```

```
        //when time has reached 1000...
        if (time == 1000)
        {
```

```

        time = 0;                //...reset time to 0
    }

    //when no flame is detected...
    if(flame == 1)
    {
        //when no obstacles detected and time under 500...
        if((sonar > 400)&&(time < 500))
        {
            forward();           //...drive forward
            time++;              //...move time forward
        }
        //when time is between 500 and 1000...
        else if((time >= 500)&&(time<1000))
        {
            spinToFlame(); //...spin until flame detected
            time++;        //...move time forward
        }
        else
        {
            spin(1);        //spin to avoid obstacle
            delay_ms(100); //wait 100 ms
        }
    }

    //when flame detected...
    else
    {
        sirens(); //...turn sirens and lights on

        while(1)
        {
            checkSonar(); //...check for candle

            //when no obstacles close...
            if (sonar > 400)
            {
                speed = 50; //...increase speed
                forward(); //...drive forward
            }
            //when candle seen...
            else
            {
                delay_ms(1400); //...keep forward until close to candle
                stop(); //stop
                fan(2000); //turn fans on for 5 seconds
                sirens(); //turn sirens off when fans stop
                LED(0); //LED OFF: routine end
            }
        }
    }
}

```



```

                                goto top;           //restart routine
                                }
                            }
                    }
            }

//This function reads the ADCA port for the sonar sensors output.
void checkSonar(void)
{
    sonar = ADCA0();
}

//This function checks B0 for high signal from flame detector.
void checkFlame(void)
{
    flame = PORTB_IN & 0x01;
}

//This function drives the motors forward (speed-2 to account for drift).
void forward(void)
{
    MotorC0(speed);
    MotorC1(speed-2);
    return;
}

//This function stops the motors and delays 10 ms.
void stop(void)
{
    MotorC0(0);
    MotorC1(0);
    delay_ms(10);
    return;
}

//This function spins the robot in a specified direction.
void spin(int direction) //direction = 0 for left, 1 for right
{
    checkFlame();

    //when no flame is detected...
    if(flame == 1)
    {
        //...turn left
        if(direction == 0)
        {
            MotorC0(-speed);

```

```

        MotorC1(+speed);
    }
    //...turn right
    else
    {
        MotorC0(+speed);
        MotorC1(-speed);
    }
}
//when a flame is detected...
else
{
    stop(); //stop
}
return;
}

```

//This function spins Smokey slowly and stops it when flame detected

```

void spinToFlame(void)
{

```

```

    checkFlame();

    //when a flame is detected..
    if (flame == 0)
    {
        stop(); //stop
        return;
    }
    //when no flame detected...
    else
    {
        MotorC0(-12); //spin slowly
        MotorC1(12);
    }
}

```

//This function activates the fans for a number of milliseconds (PORT J)

```

void fan(int ms)
{

```

```

    lcdGoto(0,0);
    lcdString("Fan ON ");
    PORTJ_OUT = 2;

    delay_ms(ms);

    PORTJ_OUT = 0;

```

```

        lcdGoto(0,0);
        lcdString("Fan OFF ");

        delay_ms(ms);
    }

//This function turns the board LED on and off (PORT Q)
void LED(int power)    //power = 1 for ON, 0 for OFF
{
    PORTQ_OUT = power;
}

//This function turns sirens on first call, off second (PORT H)
void sirens(void)
{
    PORTH_OUT = 1;
    delay_ms(100);
    PORTH_OUT = 0;
}

```