

William Otto Goethals  
TAs : Adam  
Sara  
Instructors: A. A Arroyo  
E. M. Schwartz

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

# Final Report

Casaroba

## Table of Contents

1. Abstract.....	3
2. Executive Summary.....	4
3. Introduction.....	5
4. Integrated System.....	5
5. Mobile Platform.....	7
6. Actuation.....	8
7. Sensors.....	8
8. Behaviors.....	16
9. Experimental Results.....	17
10. Conclusion.....	19
11. Documentation.....	20
12. Appendices.....	21

**Abstract**

The robot I have designed is Casaroba, an automated Casanova. The robot's main objective is to detect humans, determine their gender and obtain information about the person.

## **Executive Summary**

Casaroba is a human interacting robot that uses several sensors to find and obtain information from people.

An H-Bridge drives the two motors that give Casaroba motion. All controlling signals originate from the MAVRIC-IIB micro controller and all sensors send their data here. During movement the robot periodically scans to the right and left and uses sonar to detect obstacles in front of it and to turn away from them when they are too close.

During the scanning for objects the robot is also using a motion detector to scan for heat signals that would indicate a person is in front of it. Once a detection has occurred Casaroba will drive towards the person periodically checking the motion sensor to ensure the person has not turned and is in fact in front of the robot.

Once the sonar has indicated that the person is within certain proximity the text to speech chip is initialized and the robot begins to speak to the person. The robot then asks the gender of the person and then if it can obtain the person's number. If the person replies true the voice recognition will signal to the micro controller the response for true. The person is then asked to input their number on the keypad as the robot reads out the number for accuracy.

After number acquisition the robot turns back to return the code to its primary user, after a keypad code verification. The robot then asks if the user would like him to go back out to find friends or not. Each of the responses elicits different behaviors from Casaroba and most are rather humorous.

## **Introduction**

The only thing that really seems important to me is spending time with the opposite sex. Thus Casaroba was born, a robot that finds girl, always has a working line (who can reject a robot?), and will bring the numbers back to me. I find his purposes entertaining and useful and his systems challenging, which drive me to work on him.

The goal of Casaroba is a robot that is able to detect people, mainly girls, and obtain their telephone numbers via a keypad. The most important part of the design is the voice recognition and speech capabilities of the robot. The voice recognition is speaker independent so any person will be understood. These systems are integrated onto a micro controller that can only use sonar, detect bumps, and have a pleasing look (because who would give an ugly robot a number?).

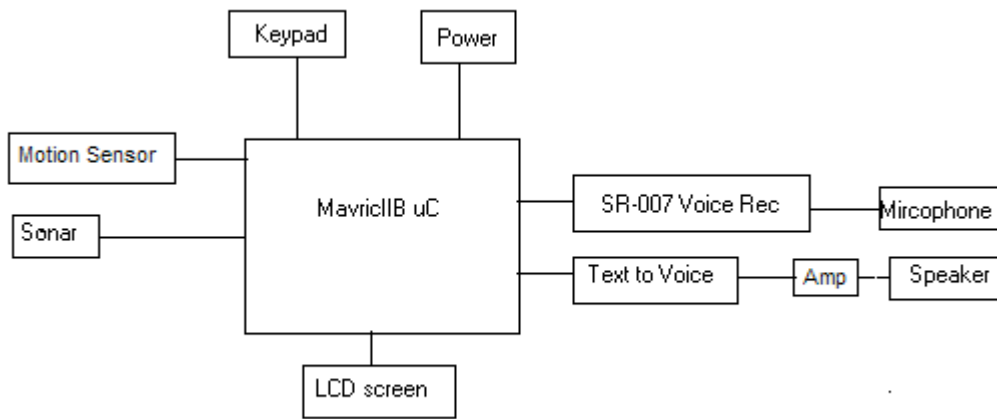
## **Integrated System**

The MAVRIC-IIB micro controller is the brain behind the design. The sensors input data into the controller who then determines what response is needed for said inputs and outputs the controlling signals to achieve the desired response.

The sensors work on a priority system, in which the bump sensors are the highest priority, regardless of the rest of the sensors. If no bumps are detected the robot uses sonar to determine how close walls and other objects are, and to avoid them. After these two behaviors the system will use the pyro-electric sensor to find humans. If one is found they are closed in on and voice recognition and speech are initiated with the SR-07 and the text to speech chip. This system is interfaced with people via a microphone and speaker. Once a successful exchange has occurred between the robot and a person, the

person will be asked to input their number on a keypad. The number will be stored in memory and displaced on an LCD display to ensure correctness.

The figure below shows the important systems connected to the micro controller. The second figure shows the MAVRIC-IIB board and where the connections to the system occur.



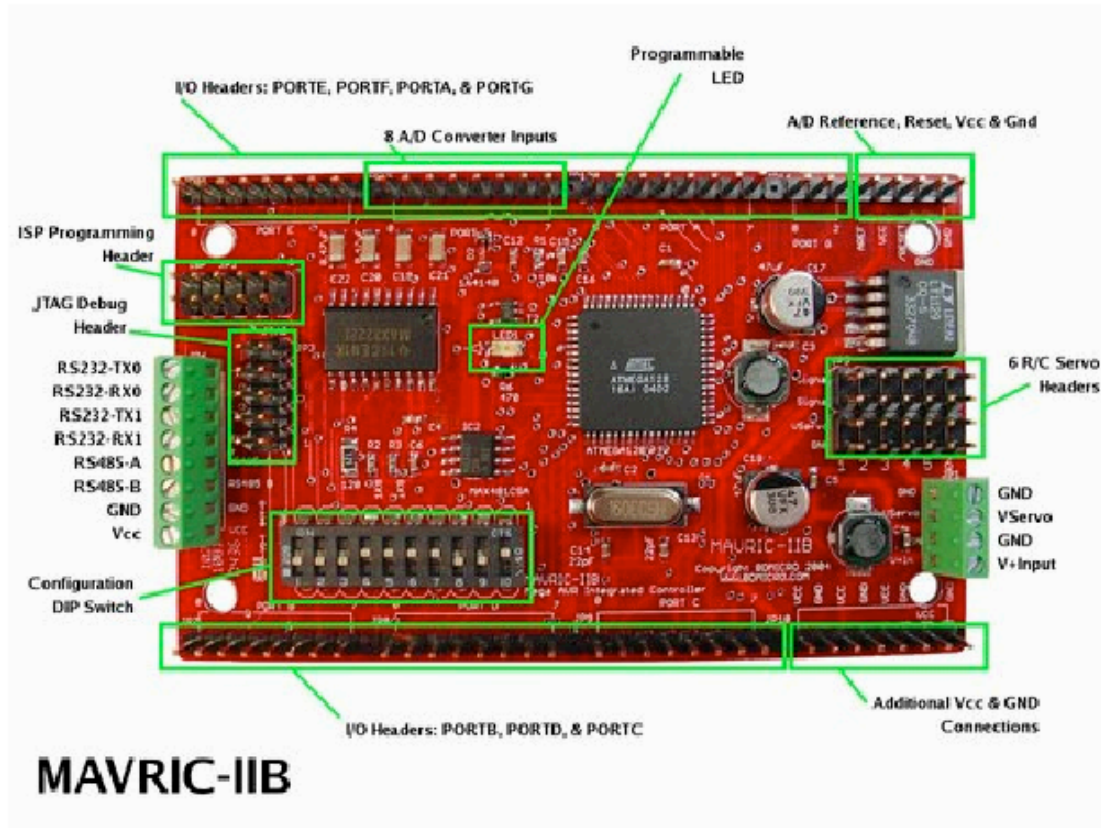


Fig.1: Mavric-IIB board

## Mobile Platform

The main body of the robot rests on a 2 wheel based structure with a pivot at the top of the platform. The rest of the body is a car casing, with the various sensors slightly protruding from the body of the car. The sonar sticks out to mimic front lights while the keypad, LCD screen, and speaker are pointed up from the top of the car.

The initial car design was too small for all the components and lead to a mess of parts held on by Velcro to the body of the car. Also the original design was not that attractive. The process of switching platforms was a much larger process than anticipated.

## **Actuation**

Two motors are used to drive the robot, and a pyro-electric detector finds humans. The detector is placed on the top of the robot (roughly 10 inches high). The servo ordered is the GWS S03N servo from pololu that produce 3.40/47 **kg-cm/oz-in** of torque. The motors, hubs, universal mounts, wheels and H-bridge to drive the motors were all ordered from lynxmotion.com.

## **Sensors**

### **1) Sonar**

#### Description

The SRF05 was obtained through Acroname and is made by Devantech, this ultrasonic ranging solution is inexpensive and easy to use. The sensor is directed to send out a 8 cycle sonic burst and then waits for the echo to determine how close an object is to the sensor. The time from the signal being sent to the return echo will determine how far away an object is.

#### Purpose

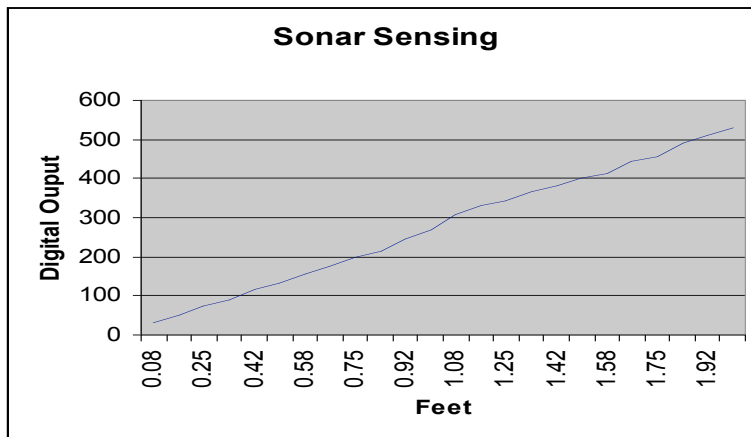
The sonar is used in Casaroba to detect and avoid objects, and to move towards a potential human. The sonar is mounted on the front of the robot and when an object is within a foot of the robot, it will turn to avoid the object. When a person is detected the sonar will try to zone in on the target in the direction of detection.



## Experiments

Given below in Chart 1 is the data taken from the SRF05 mounted on Casaroba. The digital times indicate how many times the code goes through a wait loop following the rising edge of the echo pulse. The time for each echo detection is roughly equivalent to the digital display times 5us plus the time from the falling edge of the trigger pulse (about 100us seconds). The formula to determine the distance where 340mps is the speed of sound at sea level is then:

$$\text{Distance} = (100\text{us} + 5\text{us} * \text{DisplayValue}) * 340\text{mps}$$

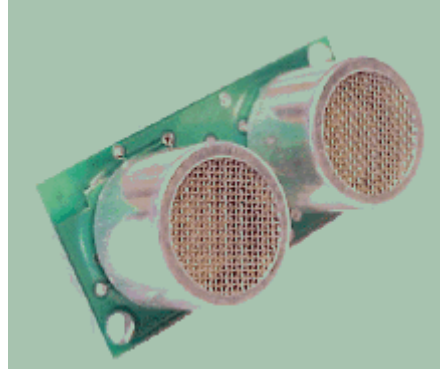


The values used in Chart 1 below are determined from actual experiment as code delays and non-ideal conditions might affect the distance formula. The max value of the sensor is 4053 cycles, which roughly corresponds to 30ms from the time the trigger pulse is dropped. This value indicates that the sensor did not receive an echo before the signal timed out. The data indicates this occurs at approximately 14 feet or a little over 4 meters.

Graph 1 shows the data of distance versus digital display.

<b>Sonar Data</b>		
<b>Feet</b>	<b>inches</b>	<b>digital display</b>
0.083333	1	33
0.166667	2	50
0.25	3	74
0.333333	4	91
0.416667	5	116
0.5	6	132
0.583333	7	155
0.666667	8	174
0.75	9	199
0.833333	10	216
0.916667	11	247
1	12	268
1.083333	13	309
1.166667	14	333
1.25	15	342
1.333333	16	365
1.416667	17	383
1.5	18	400
1.583333	19	414
1.666667	20	443
1.75	21	454
1.833333	22	492
1.916667	23	510
2	24	531
2.5	30	686
3	36	822
3.5	42	957
4	48	1089
4.5	54	1130
5	60	1121
13.5	162	3907
14	168	4053

**Chart 1**



**SRF05**

## **2) Bump Sensors**

### Description

Bump sensors are simple devices that provide conduction between two nodes when pressure is applied to the device. By applying voltage to one side of the device and a resistor to ground on the other side the voltage over the resistor can be read to indicate the device has been tripped.

### Purpose

The purpose the bump sensors is to indicate to the robot it is pressed against an object and needs to move away from the area the bump sensor is located at. This is a back up system if the sonar was not about to see an object or it turns into a corner or wall.

### Analysis

The different resistance values put to different bump sensor over a common resistor to ground will yield different analog signals. Using A/D conversation the precise voltage will let the mirco-controller know which bump sensor is being used.

### **3) Pyroelectric**

#### Description

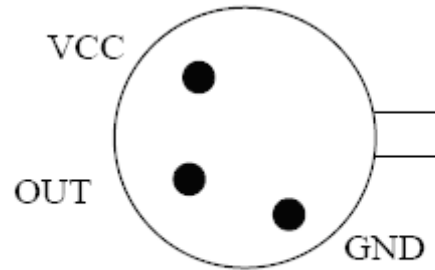
The pyroelectric, also called a passive infrared sensor, detects changes of heat radiated from various sources. When heat change is detected the sensor responds by allowing more or less current through the device. The pyroelectric is most commonly used in motion sensors and in detection for living animals. This unit has the model number KDS-4G1, however, no datasheet could be tracked down on the unit.

#### Purpose

In Casaroba the pyroelectric is used to find humans to interact with. The sensor will be mounted on the top of the robot and can be rotated by a servo to locate stationary people, or mounted stationary to find moving people.

#### Analysis

This device came with no data sheet and all information was obtained via research on similar devices and experimentation. The pin out was determined and is illustrated in the figure below. Similar devices warn of currents above 20  $\mu\text{A}$  and as such a large current limiting resistor of 1 Mega-ohm is used. The output is tied to ground to produce an output voltage. The change in heat will cause the device to allow more or less current to flow, resulting in a higher or lower voltage on the output reference.



#### **Pyroelectric Pin Out**

The analysis thus far shows large voltage changes due to lights but smaller spikes for humans in the order of a few hundred milli-volts.

#### **4) Voice Recognition**

##### Description

The SR-07 voice recognition kit was produced by SI Images, Inc. This kit is a programmable device that allows 20 1.92 second words or 40 words at .96 seconds to be recorded and recognized. The HM2007 chip, the basis for the kit, analyzes sections of the words and determines matches. The kit is continuously listening and can be set up as a stand alone module or interfacing with a microcontroller. A headset is provided for recording and speaking, and a 3 volt battery backs up the memory when the device is not in operation mode.

The figure below, from SI Images, Inc. shows the device and all it's components.

##### Purpose

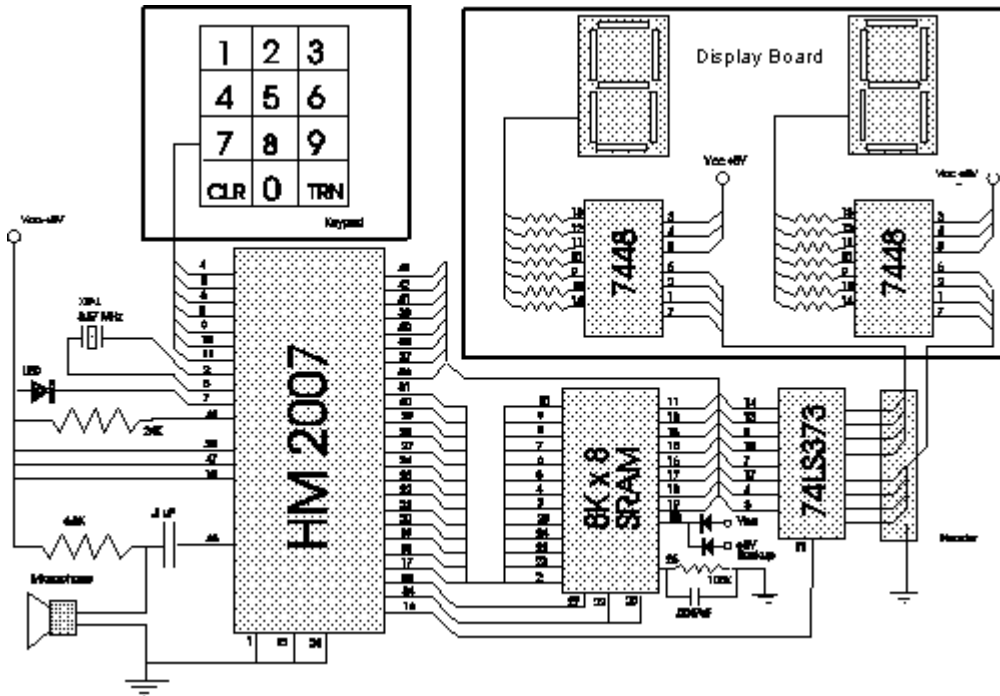
The purpose of this device is to determine from the speaker whether they are male or female and possible verification of information. Once Casaroba finds a person the robot will ask questions and interpret the responses with the SR-07 kit. By recording several instances of each word I hope to reduce the risk of misinterpretation.

### Experiment

Several trials of the SR-07 indicate that many conditions make the device perform poorly. Words with fewer syllables and those that sound alike are the most often misclassified. When the microphone is not at the correct distance or there is ambient noise often there are several misses.

Since the microphone is continuously listening when a person talks too rapidly it will listen at the middle of a word and not identify it correctly.

Under controlled conditions the results were satisfactory but not overly impressive.



## 5) Keypad

### Description

The keypad allows the user to push in a button and from the decoding of the 8 outputs the button pressed can be determined. The depression of a button connects the wire laying in the column direction to the wire in the row direction. By scanning the device that has all the columns tied low with a high voltage, the shorted output will indicate which button has been pressed.

### Purpose

The purpose of the keypad is to allow the user to enter their number and store it to be used by the user in the future. This is a crucial function of Casaroba, the key useful information that he is obtaining.

## Analysis

This device works as expected with no delays or problems so far seen.

## **Behaviors**

### **1. Object avoidance**

Object detected with the sonar that are within a 9 inch distance not emitting heat as detected by the pyroelectro are avoided.

### **2. Human Detection**

Small heat changes from the pyro indicate a person and stops the turning robot in that direction. There are several behavior modes for detecting people. Casaroba currently runs a program that continually checks for the person, which allows the robot to find people who still move after the initial detection. The other mode when there are many people around is for a single check after detection. This insures the robot is going straight and if there are lots of people are around it insures that the robot does not keep changing targets.

### **3. Speaking and Listening**

Using the text to voice chip in conjunction with the amplifier and speaker Casaroba asks questions and listens for answers. The robot will determine gender and ask to obtain a number.

### **4. Number Acquisition**



Once a human is detected and positively replies that they will give their phone number the keypad takes in the data and returns it to Otto.

## **5. Path Retracing**

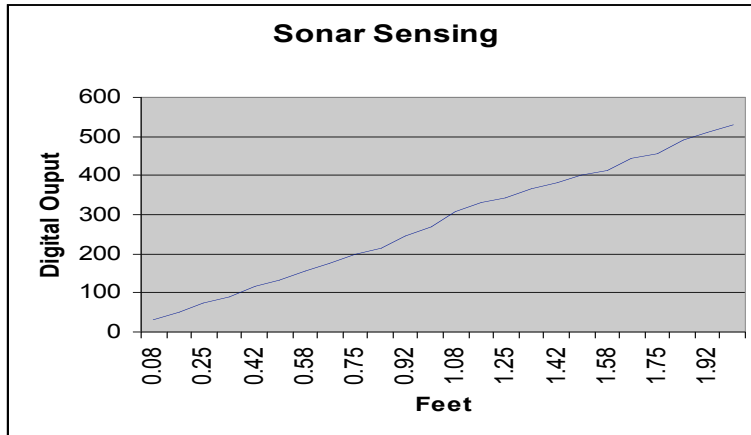
Using the human detection routine Casaroba remembers which way he has turned and the distance he has gone and once a number is collected he follows the same path to his origin.

## **Experimental Layout and Results**

### **1. Sonar Experiment**

Given below in Chart 1 is the data taken from the SRF05 mounted on Casaroba. The digital times indicate how many times the code goes through a wait loop following the rising edge of the echo pulse. The time for each echo detection is roughly equivalent to the digital display times 5us plus the time from the falling edge of the trigger pulse (about 100us seconds). The formula to determine the distance where 340mps is the speed of sound at sea level is then:

$$\text{Distance} = (100\text{us} + 5\text{us} * \text{DisplayValue}) * 340\text{mps}$$



The values used in Chart 1 below are determined from actual experiment as code delays and non-ideal conditions might affect the distance formula. The max value of the sensor is 4053 cycles, which roughly corresponds to 30ms from the time the trigger pulse is dropped. This value indicates that the sensor did not receive an echo before the signal timed out. The data indicates this occurs at approximately 14 feet or a little over 4 meters.

Graph 1 shows the data of distance versus digital display.

Sonar Data		
Feet	inches	digital display
0.083333	1	33
0.166667	2	50
0.25	3	74
0.333333	4	91
0.416667	5	116
0.5	6	132
0.583333	7	155
0.666667	8	174
0.75	9	199
0.833333	10	216
0.916667	11	247
1	12	268
1.083333	13	309
1.166667	14	333
1.25	15	342
1.333333	16	365
1.416667	17	383
1.5	18	400
1.583333	19	414

1.666667	20	443
1.75	21	454
1.833333	22	492
1.916667	23	510
2	24	531
2.5	30	686
3	36	822
3.5	42	957
4	48	1089
4.5	54	1130
5	60	1121
13.5	162	3907
14	168	4053

**Chart 1**

## **2. Voice Recognition Experiment**

Several trials of the SR-07 indicate that many conditions make the device perform poorly. Words with fewer syllables and those that sound alike are the most often misclassified. When the microphone is not at the correct distance or there is ambient noise often there are several misses.

Since the microphone is continuously listening when a person talks too rapidly it will listen at the middle of a word and not identify it correctly.

Under controlled conditions the results were satisfactory but not overly impressive.

## **Conclusion**

Casaroba, while not my first choice for name, was a complete success. Every area worked in the manner it was first designed to and some sensors better than expected.

The motion sensor is highly sensitive and allows finds people quickly and accurately. The first set of sonar detectors were destroyed with the help of bad wiring, but

the second set proved more durable and work great. The keypad and LCD were simple to implement and there were never any real problems with them.

Voice recognition proved to be the most difficult obstacle in the robot. Distance from the microphone always decreases the accuracy of detection and an initial faulty microphone caused much headache. The problem was finally solved with a wireless microphone which seems to do the job.

To follow this work you need to buy all the supplies that are listed in the sensor and platform sections and then go item by item in testing the parts until full integration with my beautiful code can be implemented.

Future work would be to clean up the programming a little, making the retracing back to the point of origin more efficient with use of the TCNT. But as far as hardware and looks go, I see no way to improve. If I had to start this project over I would have gone to a larger platform early.

## **Documentation**

[1] Acroname Inc., <http://www.acroname.com/index.html>

[2] Ihns, R. Michael "Final Report" EEL5666C Final report, Department of Electrical and Computer Engineering, University of Florida, Spring 2005.

[3] Images SI, Inc.,

<http://www.imagesco.com/articles/hm2007/SpeechRecognitionTutorial02.html>

## Appendices

### CODE

```
/******
```

```
This program was assisted by the  
CodeWizardAVR V1.24.6 Standard  
Automatic Program Generator  
© Copyright 1998-2005 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com  
e-mail:office@hpinfotech.com
```

```
Project :  
Version :  
Date   : 4/15/2006  
Author : Job  
Company : Earth  
Comments:
```

```
Chip type      : ATmega128  
Program type   : Application  
Clock frequency : 16.000000 MHz  
Memory model   : Small  
External SRAM size : 0  
Data Stack size : 1024  
*****/
```

```
#include <mega128.h>  
  
// Alphanumeric LCD Module functions  
#asm  
    .equ __lcd_port=0x15 ;PORTC  
#endasm  
#include <lcd.h>  
#include <delay.h>  
#include <stdlib.h>  
int lookSonar(int blah);  
void avoidObstacle(void);  
int gotoPerson(void);  
  
// Standard Input/Output functions  
#include <stdio.h>
```



```

if(foundPerson == 1)//you found a person
{
    PORTD=0x30;

    putchar(0x00);
    putsf("I see you");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}

    PORTD = 0x3A;
    toPersonTime =gotoPerson();
    firstForward += toPersonTime;
    //stop
    PORTD = 0x30;    //stop right where you are
    forwardTime += temp; //still need to record how much you just turned

    return 1;    //return that you found a person
}
else
{
    forwardTime += temp;
}
//go right
lcd_putsf("right 400ms");
PORTD = 0x36;

//looking for people as moving right
temp = lookSonar(70);
if(foundPerson == 1)
{
    PORTD=0x30;

    putchar(0x00);
    putsf("I see you");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}

    PORTD = 0x39; // got to go back a little bit
    delay_ms(200);
    PORTD = 0x3A;
    toPersonTime =gotoPerson();
    firstForward += toPersonTime;
    //stop
    PORTD=0x30;
    rightTime += temp;
    return 1;
}
else
{
    //rightTime += temp; bad move
}

lcd_clear();

```

```

lcd_putsf("try the left");
PORTD = 0x39;
temp=lookSonar(125);
if(foundPerson == 1)
{
PORTD=0x30;
left = 1;
putchar(0x00);
putsf("I see you");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}

PORTD = 0x36; // got to go back a little bit
delay_ms(200);
PORTD = 0x3A;
toPersonTime =gotoPerson();
firstForward += toPersonTime;
//stop
PORTD=0x30;
leftTime += temp;
return 1;
}
else
{
// leftTime += temp; bad move
}
lcd_clear();
lcd_putsf("back right");
PORTD = 0x36;
temp=lookSonar(55);
if(foundPerson == 1)
{
PORTD=0x30;
left = 1;
putchar(0x00);
putsf("I see you");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}

PORTD = 0x39; // got to go back a little bit
delay_ms(200);
PORTD = 0x3A;
toPersonTime =gotoPerson();
firstForward += toPersonTime;
//stop
PORTD=0x30;
leftTime += (100 - temp); //this needs to be total time in loop -temp
return 1;
}
else
{
//rightTime += temp; bad move
}
}

```



```

}

int checkPyro(void)
{
  int number;
  number = read_adc(2);

  if(number > 150)
  {
    //call find person
    return 1;
  }
  else if (number < 110)
  {
    //call find person
    return 1;
  }

}
}

```

```

int lookSonar(int threshold)
{
  int time;
  int gotoTime = 0;
  int pyro = 0;

  time = 0;
  //total = time + threshold;

  while(time < threshold)
  {
    pyro = checkPyro();

    if(pyro == 0)
    {
      time++;
      i=0;
      PORTD.6 = 1;
      delay_us(10);
      PORTD.6 = 0;
      while(PIND.7 == 0)
      {

        delay_us(20);
      }

      while(PIND.7 == 1)
      {
        i++;

```

```

        delay_us(5);
    }

    if(i < 100)
    {
        PORTD = 0x30; //stopping the robot
        object = 1;
        avoidObstacle();
        return time; //found a person
    }

}
else if(pyro == 1)
{

    foundPerson =1;
    /* PORTD = 0x3A; //go towards the signal
    gotoTime = gotoPerson();
    forwardTime += gotoTime;
    */

    return time;
}

}
object = 0;
return time; // no person
}

void avoidObstacle(void)
{
    //basically avoid objects by turning 50 degrees to the right every time, then storing num of right turns
    in variable numRightAvoidanceTurns
    lcd_putsf("right 400ms");
    PORTD = 0x36;
    delay_ms(400);
    numAvoidanceRightTurns++;
}

int gotoPerson(void)
{
    int moving = 0;
    int count;
    int t;
    int check=0;
    int loopy = 0;

    int temp;
    while(1)
    {
        count = 0;
        check = 0;

```

```

while(count < 40)
{
    count++;
    moving++;
    t=0;
    PORTD.6 = 1;
    delay_us(10);
    PORTD.6 = 0;
    while(PIND.7 == 0)
    {

        delay_us(20);
    }

    while(PIND.7 == 1)
    {
        t++;
        delay_us(5);
    }

    if(t < 200)
    {
        PORTD = 0x30; //stopping the robot
        return moving; //found a person
    }
}
while(check == 0)
{
    PORTD = 0x36;

    loopy = 0;
    while((check == 0) && (loopy <80))
    {
        loopy++;
        delay_ms(5);
        check = checkPyro();
    }
    if(check == 1)
    {
        PORTD =0x3A;
    }

    if(check == 0)
    {
        PORTD = 0x39;

        loopy = 0;
        while((check == 0) && (loopy <160))
        {
            loopy++;
            delay_ms(5);
            check = checkPyro();
        }
    }
}

```

```

    }
    if(check == 1)
    {
    PORTD =0x3A;
    }

    if(check == 0)
    {
    PORTD = 0x36;

    loopy =0;
    while((check == 0) && (loopy <80))
    {
        loopy++;
        delay_ms(5);
        check = checkPyro();

    }

    if(check == 1)
    {
    PORTD =0x3A;
    }

    if(check == 0)
    {
    PORTD = 0x39;

    loopy = 0;
    while((check == 0) && (loopy <160))
    {
        loopy++;
        delay_ms(5);
        check = checkPyro();

    }
    if(check == 1)
    {
    PORTD =0x3A;
    }
    }
    }
}

int voiceRec(void)
{
int loop = 0;

while(loop ==0)
{

```

```

if((PINA == 0x09) || (PINA ==0x10) || (PINA ==0x11) || (PINA ==0x12))
{
putchar(0x00);
putsf(" True");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
loop=1;
return 3;
}
else if((PINA == 0x13) || (PINA ==0x14) || (PINA ==0x15) || (PINA ==0x16))
{
putchar(0x00);
putsf(" False");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
loop=1;
return 4;
}
}

```

```

}

```

```

int voiceRecSex(void)

```

```

{
int loop = 0;

```

```

while(loop ==0)
{

```

```

if((PINA == 0x01) || (PINA ==0x02) || (PINA ==0x03) || (PINA ==0x04))
{
putchar(0x00);
putsf(" Guy");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
loop=1;
return 2;
}

```

```

else if((PINA == 0x05) || (PINA ==0x06) || (PINA ==0x07) || (PINA ==0x08))
{
putchar(0x00);
putsf(" Woman");
woman = 1;
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
loop=1;
return 1;
}
}

```

```

    }
    }
}

char getKey(void)
{
    char key;
    button = 0;

    while (button == 0) // while (SW2 not pressed)
    {
        if (counter == 0)
        {
            PORTB = 0x40;
            counter = counter + 1;
        }
        else if (counter == 1)
        {
            PORTB = 0x80;
            counter = counter + 1;
        }
        else if (counter == 2)
        {
            PORTB = 0x20;
            counter = counter + 1;
        }
        else if (counter == 3)
        {
            PORTB = 0x10;
            counter = counter + 1;
        }
        else if (counter == 4)
        {
            counter = 0;
        }
    }

    if (PINB.0 == 1) // Row 1 is Pressed
    {
        button = 1;

        if(1 == 0)
        {
        }
        else if (PORTB.6 == 1)
        {
            out= '0';
        }
        else if (PORTB.7 == 1)
        {
            out= '*';
        }
        else if (PORTB.5 == 1)

```

```

{
  out= '#';
}
else if (PORTB.4 == 1)
{
  out= 'D';
}
}
else if (PINB.1 == 1) // Row 1 is Pressed
{
  button = 1;

  if(1 == 0)
  {
  }
  else if (PORTB.6 == 1)
  {
    out= '8';
  }
  else if (PORTB.7 == 1)
  {
    out= '7';
  }
  else if (PORTB.5 == 1)
  {
    out= '9';
  }
  else if (PORTB.4 == 1)
  {
    out= 'C';
  }
}
}
else if (PINB.2 == 1) // Row 1 is Pressed
{
  button = 1;

  if(1 == 0)
  {
  }
  else if (PORTB.6 == 1)
  {
    out= '5';
  }
  else if (PORTB.7 == 1)
  {
    out= '4';
  }
  else if (PORTB.5 == 1)
  {
    out= '6';
  }
  else if (PORTB.4 == 1)
  {
    out= 'B';
  }
}
}
}

```

```

else if (PINB.3 == 1) // Row 1 is Pressed
{
  button = 1;

  if(1 == 0)
  {

  }
  else if (PORTB.6 == 1)
  {
    out= '2';
  }
  else if (PORTB.7 == 1)
  {
    out= '1';
  }
  else if (PORTB.5 == 1)
  {
    out= '3';
  }
  else if (PORTB.4 == 1)
  {
    out= 'A';
  }
}
}

  lcd_putchar(out);
  putchar(0x00);
  putchar(out);
  putchar(0xAA);
  while(PINE.5 == 0){}
  while(PINE.5 == 1){}
  delay_ms(200);
  while(( PINB.0 != 0) || (PINB.1 != 0) || (PINB.2 != 0) || ( PINB.3 != 0) )
  {

  }

  key = out;

return key;
}

void driveBackS(void)
{
  int delayForward;
  int delayRight;
  int delayLeft;
  int delayRightAvoid;
  int count = 0;
  int t;
  //go 180 degrees back the way you came

```



```

PORTD= 0x39;
delay_ms(1400);
PORTD= 0x3A;
delay_ms(1000);
PORTD= 0x30; //stop

//rotate 45 degrees either right or left
if(left == 1) //if he went left, rotate 45 degrees right to get back
{
    PORTD = 0x36; //go right
    delay_ms(200);
}
else //if he went right, rotate 45 degrees left to get back
{
    PORTD = 0x39;
    delay_ms(400);
}

while(count < 120)
{
    count++;

    t=0;
    PORTD.6 = 1;
    delay_us(10);
    PORTD.6 = 0;
    while(PIND.7 == 0)
    {

        delay_us(20);
    }

    while(PIND.7 == 1)
    {
        t++;
        delay_us(5);
    }

    if(t < 200)
    {
        PORTD = 0x36; //turning right
        delay_ms(200); //found a person
        PORTD = 0x3A;
    }
    delay_ms(5);
}

/*delay_ms(5*firstForward);
PORTD= 0x30; //stop

if(rightTime >0)
{ //go left, to backtrack from right, check conversion
    PORTD = 0x39;
    delayRight = 800/85 * rightTime;
}
}

```

```

    delay_ms(delayRight);
}
if(leftTime >0)
{
    //go right, to backtrack from left, check conversion
    PORTD = 0x36;
    delayLeft = 400/50 * leftTime;
    delay_ms(delayLeft );
}
if(numAvoidanceRightTurns%8 > 0) //if did not turn right 360 degrees
{
    //go left the number of times you went right
    PORTD = 0x39;
    delayRightAvoid = (numAvoidanceRightTurns * 600);
    delay_ms( delayRightAvoid );
}
if(forwardTime >0) //CAUTION: might need to make the ints in here doubles
{
    //go forward, here is a conversion from sonar time to delay time, check this!!
    PORTD = 0x3A;
    delayForward = 1000/200 * forwardTime;
    delay_ms(delayForward);
}
*/
//STOP YOU ARE BACK TO BILL
PORTD = 0x30;
}

```

```

#####

```

```

####
void main(void)
{
// Declare your local variables here
int son;
int size = 7;
char phoneNum[8]; // +1 for the null terminated c string
int i = 0;
char billsCode;
char s;
int word;
int phone;
int reject = 0;

PORTA=0x00;
DDRA=0x0A;

PORTB=0x00;
DDRB=0xF0;

PORTC=0x00;

```

```

DDRC=0x00;

PORTD=0x00;
DDRD=0x11;

PORTE=0x00;
DDRE=0x00;

PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x01;
TCNT0=0x00;
OCR0=0x00;

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;

```

```

TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 2400
UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x01;
UBRR0L=0xA0;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;

// LCD module initialization
lcd_init(20);

#####
#####
while (1)
{
//Sonar and driving
//STEP 1: CALL findAPerson() method
//find person, drive towards

//*****
*****
_lcd_ready();
delay_ms(50);

```

```

lcd_putsf("world come here");
delay_ms(5000);
lcd_clear();

PORTE=0x10;
delay_ms(1000);

PORTD=0x30; //enable motors
PORTD=0x3A; // go forward young one

son= findPersonS();

//*****
*****
//setting the not reset on TtoV

//STEP 2: Talk to the person, ask guy/girl?
lcd_clear();
lcd_putsf("go ahead and work");

putchar(0x01);
putchar('7');
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

putchar(0x03);
putchar('0');
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

putchar(0x02);
putchar('1');
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

putchar(0x00);
putsf("Hello, cas aroba is here.");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

putchar(0x00);
putsf("I'm sorry, I am blind, are you a guy or woman?");
putchar(0xAA);
while(PINE.5 == 0){}

```

```

while(PINE.5 == 1){}
delay_ms(1000);

//TO DO: STEP 3: Determine male/female here
word = voiceRecSex();
delay_ms(500);

if(word == 1)
{
//STEP 4: Ask for his/her phone number.
//TO DO: Add conditional If female

putchar(0x00);
putsf("You are a hot mama, can I holla?");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

}
else if(word == 2)
{

putchar(0x03);
putchar('1');
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

putchar(0x00);
putsf("You are a very handsome man, you must work out");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

}

putchar(0x00);
putsf("Can I have your number, true or false?");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
//TO DO: STEP 3: Determine male/female here
phone = voiceRec();
delay_ms(500);

if(phone == 3)
{
//STEP 4: Ask for his/her phone number.
//TO DO: Add conditional If female

```

```

putchar(0x00);
putsf("Please enter your 7 digit number firmly into my keypad");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

//STEP 3: Save the # to String
for(i = 0; i<size; i++)
{
    phoneNum[i] = getKey();
}
phoneNum[size] = '\0';

//Say thank you for phone number
putchar(0x00);
putsf("Thank you");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

}
else if(phone == 4)
{
    putchar(0x00);
    putsf("Foley told me that you were a snob, I don't need this");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);
    reject =1;
}

//STEP 4: Drive Back to Bill
driveBackS();

if(reject == 0)
{
//STEP 5: Receive Bill's code and check it
putchar(0x00);
putsf("Are you Otto, true or false?");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(2000); //delay for answer

phone = voiceRec();
delay_ms(500);

if(phone == 3)

```

```

{

putchar(0x00);
putsf("Please enter your code");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);

billsCode = getKey();
if(billsCode == '1')
{

if(woman == 1)
{
//STEP 6: Say # back to Bill
putchar(0x00);
putsf("Here is a Hottie for you.");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
}
else
{
//STEP 6: Say # back to Bill
putchar(0x00);
putsf("I met this guy who hit on me, this is his number");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
}

}

for(i = 0; i<size; i++)
{
putchar(0x00);
putchar(phoneNum[i]);
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
}
}
else
{
//You are NOT Bill
putchar(0x00);
putsf("You are not Otto");
putchar(0xAA);
while(PINE.5 == 0){}
while(PINE.5 == 1){}
delay_ms(1000);
}
}

```



```

    }
    else if(phone == 4)
    {

    putchar(0x00);
    putsf("Tell William I will give him a number if he gets me a lift kit");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

    }

    putchar(0x00);
    putsf("Do you want me to go find some more friends?");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(2000);

    phone = voiceRec();
    delay_ms(500);

    if(phone == 3)
    {
    //STEP 4: Ask for his/her phone number.
    //TO DO: Add conditional If female

    putchar(0x00);
    putsf("OK, later Otto");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

    PORTD= 0x39;
    delay_ms(1400);
    PORTD=0x30;

    }
    else if(phone == 4)
    {
    putchar(0x00);
    putsf("Alright then, Circle time!");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

    while(1)
    {
    PORTD=0x39;

```

```

    delay_ms(3000);
    PORTD=0x36;
    putchar(0x00);
    putsf("I'm getting dizzy, turn me off");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

}

}

}
else
{
    putchar(0x00);
    putsf("I got rejected, I will keep trying Otto");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

    putchar(0x00);
    putsf("Put me back in the game");
    putchar(0xAA);
    while(PINE.5 == 0){}
    while(PINE.5 == 1){}
    delay_ms(1000);

    reject = 0;
}
}

```

```

forward = 0;
left = 0;
word=0;
woman = 0; //if woman = 0, you met a man, else you met a women
object =0;
forwardTime =0;
rightTime =0;
leftTime = 0;
numAvoidanceRightTurns = 0;
foundPerson = 0;

}
}

```