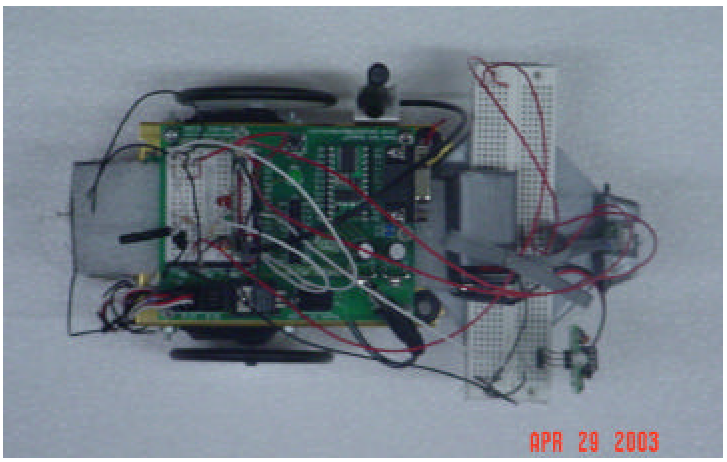
MAE 476 / 576
Mechatronics

FINAL PROJECT

MINE DETECTING ROBOT



MAY 8, 2003

DEMONSTRATION – APRIL 30, 2003

GROUP B:

DAVID PERICAK
2797 2725
JAIRAM RAMASWAMY
3067 3752
SRINIVAS SUNDARAGOPAL
3045 9561
TALIB BHABHRAWALA
2978 6208
VAMSI KRISHNA
3010 3495

TABLE OF CONTENTS

ABSTRACT

The purpose of this project was to get a feel of distributed sensing and control of a wheeled mobile robot. We tried to simulate a mine detecting robot which is capable of detecting mines in a war like scenario and sends back the approximate location back to the base station. A abase station was set up to command the mobile robot and a mobile robot with various sensors mounted on it is used to detect the mines and send back the location of the mines.

INTRODUCTION

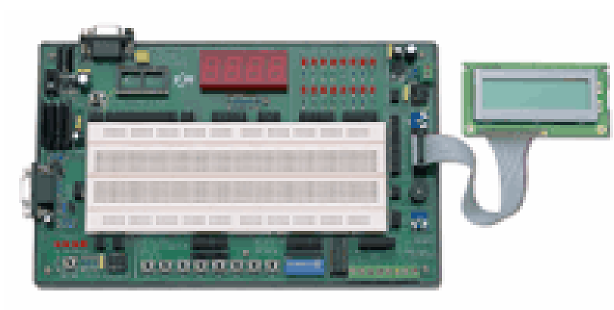
In today's world robots are omnipresent, they are used virtually in every area man can think of. Their importance is more felt when dealing in harsh environments, where it is not safe for humans because of the dangers present. One such area where these robots are of immense help is in the detection of mines in a mine field.

In the current project we tried to simulate the minefield and with the help of a wheeled robot, mines were detected. The concept of distributed computing was used effectively to achieve the desired goals. Even partitioning of the tasks amongst the various processors also was implemented. One of the processor resided on the base station and the second processor resided on the mobile robot. The mobile robot was controlled from the base station. The mobile robot goes in to the mine field and detects the mines and sends back the relevant data back to the base station. In order for continuous flow of data between the two processors a communication link was set up. This was done using the RF and IR communication (Though in the real life situation only RF would be preferred). The robot was developed to perform in three modes Autonomous mode, systematic grid search mode and tele operated mode

HARDWARE USED

The hardware that was used came from the Stamp Works kit

Stamp Works Kit

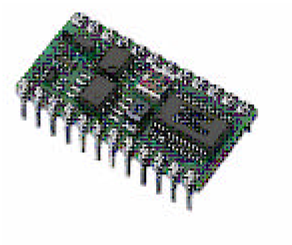


Hardware used included:

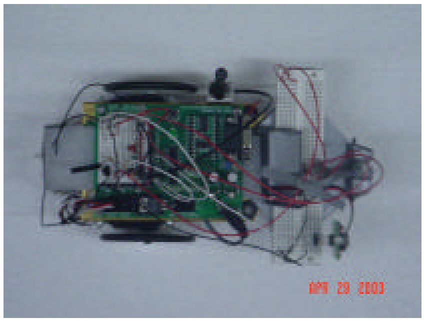
- NX-1000 BASIC Stamp Experiment Board
- BASIC Stamp 2 module (BS2-IC)
- BOE-BOT.
- 2 row x 16 character Hitachi-compatible parallel LCD with custom manufactured cable
- Digital multimeter with two probes
- Wire cutter / wire stripper / pliers
- Serial cable
- Computer - Furnas 811
- Batteries - 9V and AA
- BOE BOT
- Infrared transmitter

- Infrared receiver
- RF transmitter
- RF receiver
- RF antenna
- Digital IR Range finding sensor(GP2D12)
- Photo resistors.

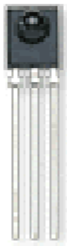
Snap shots of major Hardware:



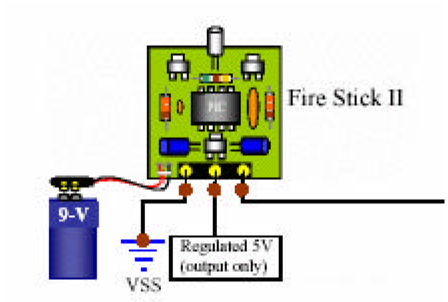
Basic Stamp II



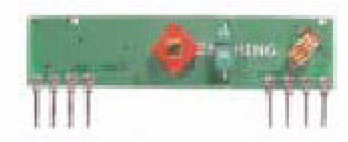
BOE BOT fully assembled



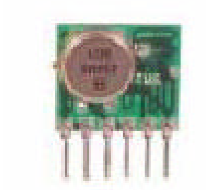
Infrared Receiver



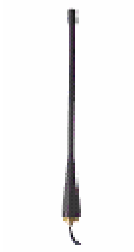
Fire Stick Infrared Transmitter



RWS 434 RF Receiver



TWS 434 RF Transmitter



RF antenna

HARDWARE DESCRIPTION

Some of the special hardware we used to assure successful completion of the project.

Boe-Bot Kit

We use a Boe-Bot Kit as the mobile robot, which has wheels with two servomotors available for movement. It is a tricycle model, with two disk wheels located at left and right sides, and ball-wheel at the back.

The different assembly that we have made is adding a photoresistor at the bottom of the sponge cardboard system in front of the WMR. The figure shows the WMR as shown in the assembled robot.

Fire Stick 2 Infrared Transmitter

The IR transmitter as shown in the picture in the previous page required a 9V battery source. It can transmit serial data by connecting to a single I/O data pin on a microcontroller. It has a 3-pin header that makes connection to the BASIC Stamp, PIC or other controller circuit quick & easy. The receiver is a Panasonic PNA4602M, which is capable for serial data reception at baud rates up to 2400.

The circuit connections as shown in the figure are relatively simple. They are just connected to I/O pins and voltage sources. We chose this communication interface because both the hardware and software interface is easy.

TWS-434 Transmitter

This small transmitter can output up to 8mW at 433.92 MHz. It can operate in the range of about 400 ft. outdoors, or about 200 ft. indoors. It can go through most walls. The operation voltage can be from 1.5 to 12 V and it accepts both linear and digital input. Figure in the previous page shows the transmitter.

RWS-434 Receiver

This receiver is also operating at 433.92 MHz with 4.5 – 5.5 V DC. Its sensitivity is 3 V, and it can have both linear and digital outputs.

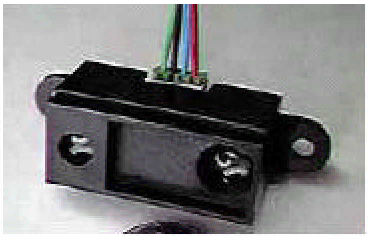
RF Antenna

The RF transmitter and receiver above need to connect to antennas when they are used for long-range detection. The antenna is fed through the base with RG-174 coax cable which may be soldered directly to the TWS-434 or RWS-434 antenna connections or circuit board. The length of the antenna body is 7-inches, the coax is 8-inches. These antennas simplify placement of all the 433.92MHz RF modules inside small project enclosures, and provide maximum operating distance for the TWS, RWS, RXLC and TXLC RF modules.

DIRRS Infrared Ranger Sensor

The short-range infrared (IR) detector with detection range of about 10 cm to 80 cm (4” to 31.5”). This compact device consists of an IR transmitter, receiver, optics, filter, detection and amplification circuitry. It is highly resistant from the environment condition, especially ambient light, and some variations of the surface reflectivity of the detected object.

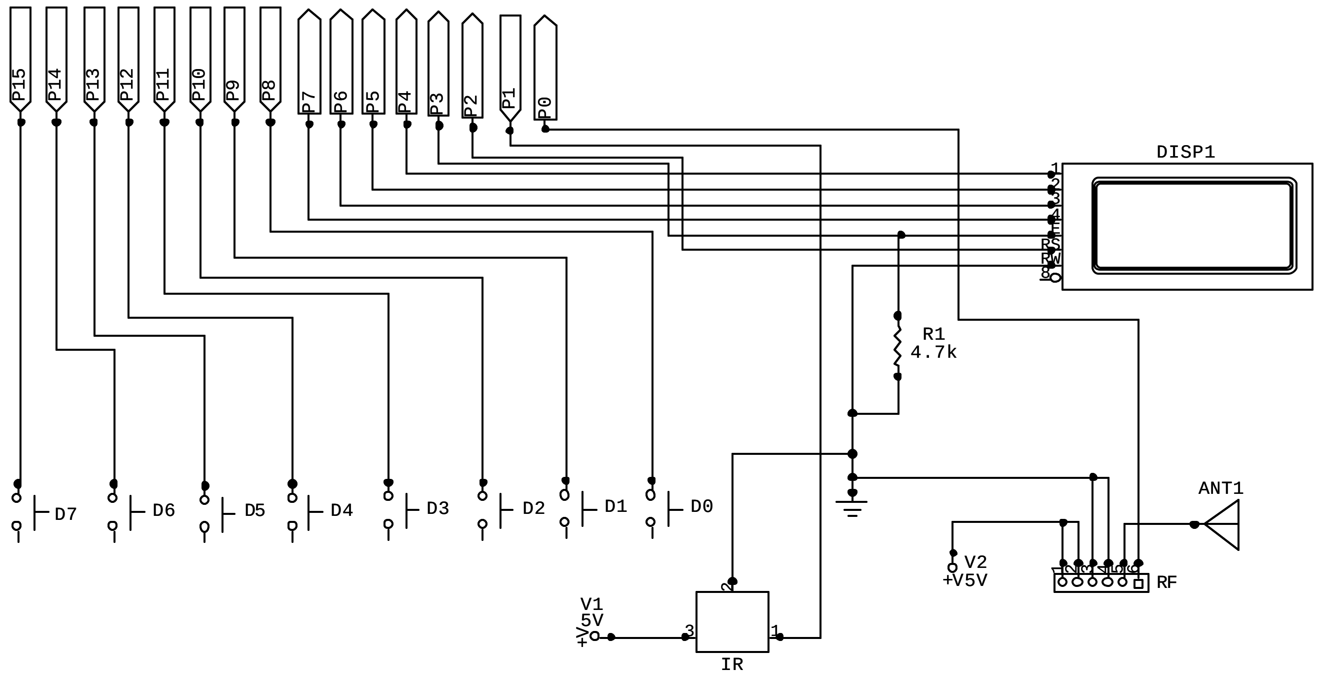
Calibration is required for comparing the output numerical values of the device and the corresponding distance.



DIRRS Infrared Ranger

CIRCUIT DIAGRAMS

BASE STATION CIRCUITRY



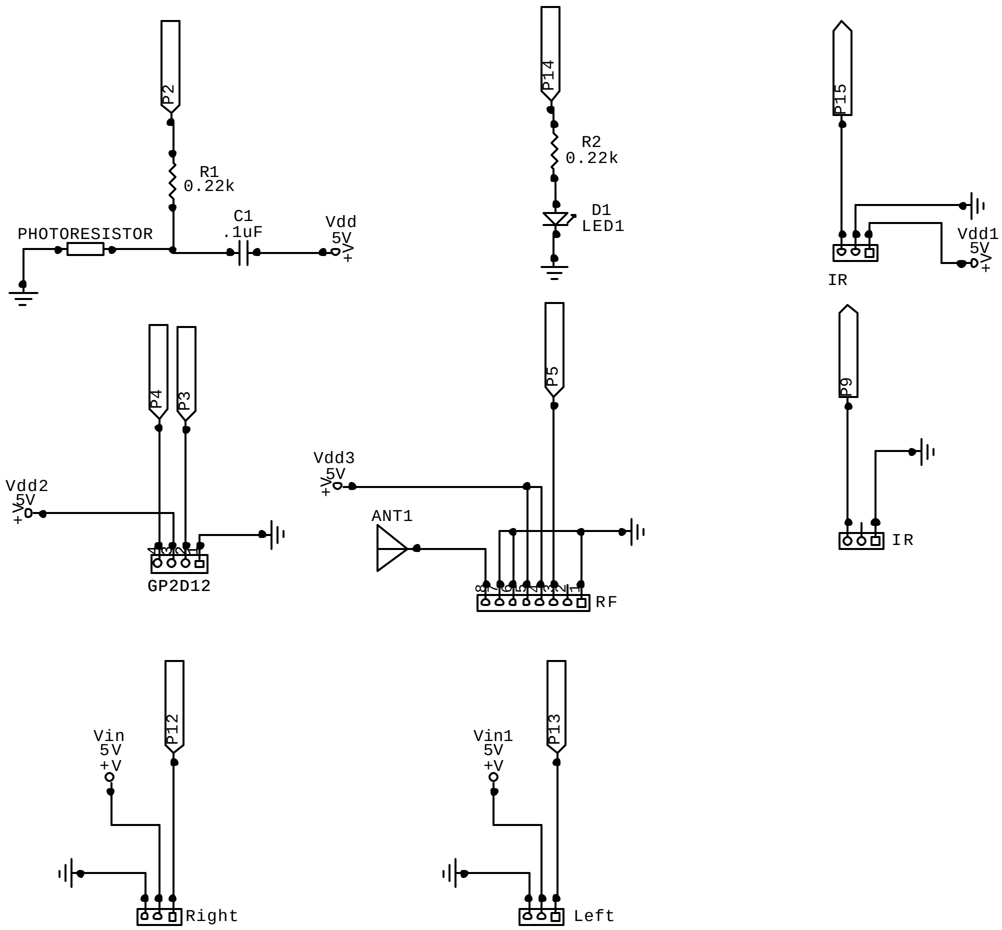
IR : IR Receiver

RF : RF Transmitter

DISP1 : LCD Display

ANT1 : RF Antenna.

MOBILE STATION CIRCUITRY:



Object detection circuitry:

IR : Receiver(Connected to Pin 15)

LED1 : Transmitter.

Distance measurement circuitry

GP2D12: Digital IR Ranging Sensor.

Wheels Circuitry:

Right : Right wheel

Left : Left wheel.

Communications:

RF : RF Receiver.

IR : IR Transmitter

ANT1 : Antenna.

VARIOUS MODES OF OPERATION

The different modes of operation are:

a. *Systematic Grid Mode*

- BOE BOT will automatically maneuver down a straight path till a mine (simulated as a black spot) is detected. Once a mine is detected the BOT will stop and inform the base station about the presence of mine. The base station can request the location of the mine and asks it to proceed further or can call the robot back and ask the robot to trace a mine in a separate lane.

b. *Manual Mode*

- The robot can be operated manually from the base station. This mode can also be used during the systematic grid search mode in order to straighten its out its path. The manual controls that the BOT possess are left and right turning and also forward and backwards movements.

c. *Autonomous Mode*

- Using the Digital IR Ranging Sensor the BOT is able to detect any obstacles which come in its path. By detecting the distance from obstacles the BOT is able to maneuver around it.

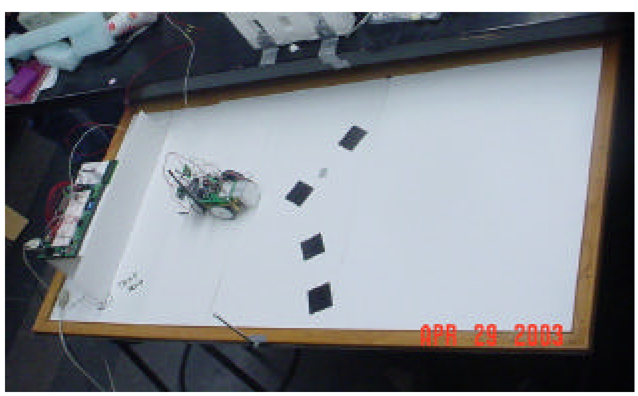
PROCEDURE

The purpose of the project is to detect mines in a predetermined area and reporting the location to the base station. The robot also avoids collision. The robot has the additional capacity to navigate through a maze.

APPROACH

- Mines were simulated by black spots.
- Photo-resistors were used to identify the black spots in a white background
- IR LED's are used for obstacle avoidance during manual mode and grid search mode.
- IR ranging sensors were used to measure the distance as they were more accurate and easier to calibrate (no inconsistencies with respect to surface, inclination of the object etc.).
- The ranging sensor also allowed us to set up a fairly accurate system for obstacle detection and autonomous navigation.

We adopted a bottom-up approach and built in the robot from a subsystem level. The following were the various stages in the development of the system.



- **Assembly of the Mobile Robot**: We began the project by assembling the mobile BOE BOT robot. This involved fixing the wheel and trying to get it properly aligned and giving it a motion.

- **Communication** : We first worked on the communication between Mobile robot and Base station. The type of communication we chose to use was wireless rather than wired. The two types of communication that we used were IR and RF. The IR communication is used for communicating from mobile robot to base station. The RF communication channel is used to send instructions from base station to mobile robot. The RF was the more reliable communication as it was not affected by line of sight in any way. The issue of line of sight was considerably resolved by placing the receiver at a height as though it was a satellite. This achieved very good results.

- **Calibration:** We also had to do the calibration for getting the wheels take right angle turns and the desired angular turns. The calibration was one of the most toughest and time consuming job as it changed with new batteries, with rough surface and also the surrounding frequency leak ins. The calibration also involved the calculation of the distance of the mobile robot from the base station using DIRRS.

- **Mine Detection**: We used a photoresistor which required detecting the mines and this involved a little modification in the software so that it sends out the correct signal back to the base station.

- **Obstacle Detection**: We tried using Whiskers and tried to test it but finally arrived at a decision that it would not serve any purpose other than finding and sensing obstacles. Instead the photoresistor was a better choice.

- **Platform Setup**: The mine field was simulated on a hard board. The advantage of using the board was that the calibration of the wheels can be done in a more accurate manner.

INTEGRATION OF SUBSYSTEMS

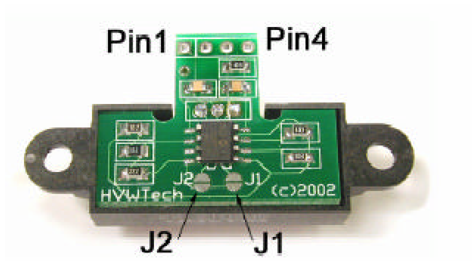
After the initial set of systems we still had a set of decoupled systems which were working fine independently. However to get the desired output from the robot, it meant coupling all the above systems at both software and hardware level. Our attempt was to create a seamless integration between all the systems.

EXPLANATION OF SUBSYSTEMS

DISTANCE MEASUREMENT

Sensor used: Digital Infra Red Ranging Sensor Plus GP2D12 [1].

Figure:



Connection Table of the Sensor:

Pin	Symbol	Connect to
1	GND	Ground
2	Vin	Output pin of Microcontroller
3	Vcc	+5V DC
4	Vout	Input pin of Microcontroller

WORKING

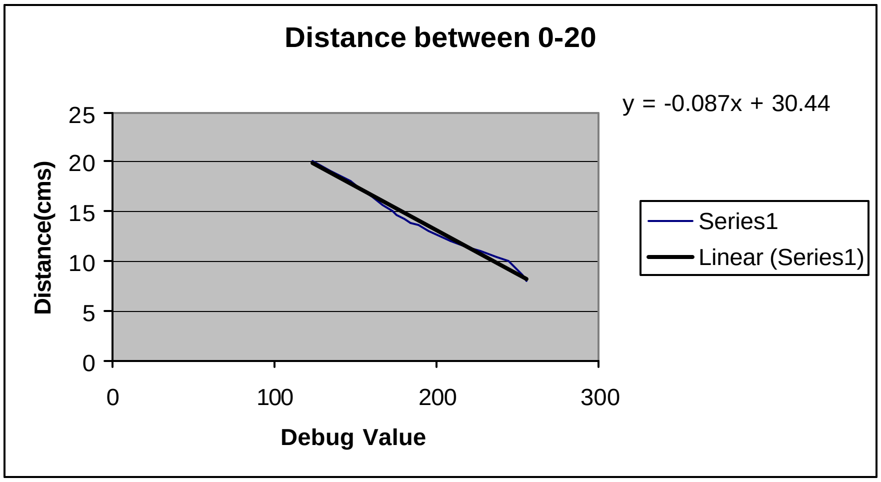
GP2D12 uses an array of photodiodes (called a Position Sensitive Detector, or PSD) and some simple optics to detect distance. An infra-red diode emits a modulated beam; the beam hits an object and a portion of the light is reflected back through the receiver optics and strikes the PSD. When light hits the PSD, it hits one of the 'taps' and causes current to flow out each end of the resistor, forming a voltage divider. As an object moves closer or farther from the sensor, incoming light hits a different 'tap' causing the current coming out each end of the resistor to change. These currents are compared and a voltage proportional to the position of the 'tap' (and hence the distance of the object) is generated.

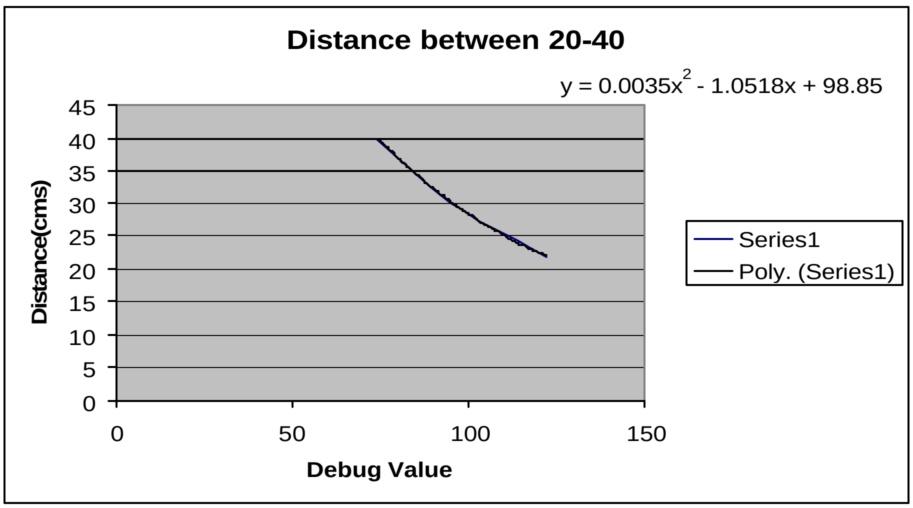
CALIBRATION

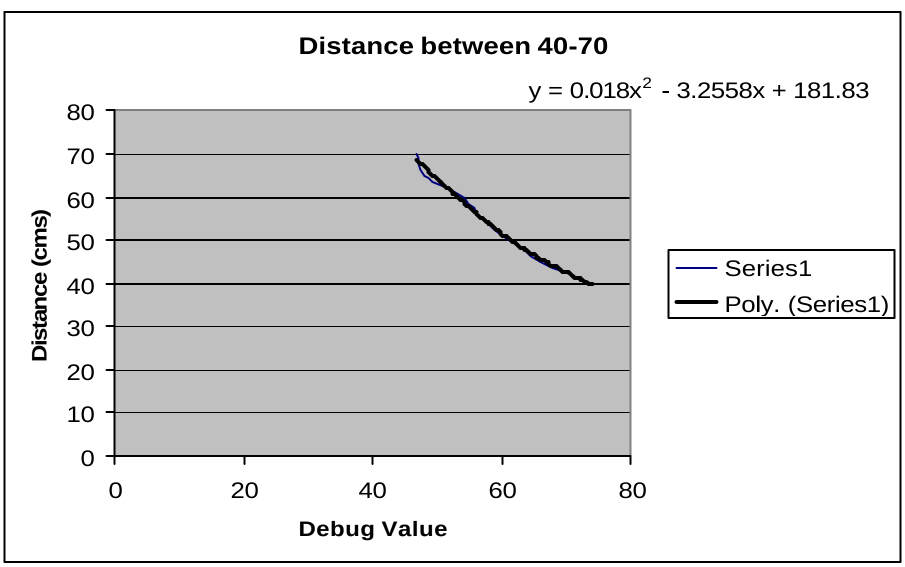
The calibration for IR Ranging sensor is as shown below. The sensor was calibrated for the distances before being mounted on the actual position. The sensor gives a distance from 0- 80 cms which corresponds to 0-255 units. The sensor was calibrated by placing a obstacle (white) in front of it noting the values in debug window.

Three different regions are identified and calibration is done differently for these regions. This is done because of the inherent computational limitations of the Basic Stamp in computing.

Distance	**Debug Value**
8	255
10	244
12	209
14	184
15	172
18	147
20	123
22	122
25	111
30	95
35	84
40	74
45	66
50	61
55	57
60	54
65	48
70	47
80	40
85	37


Distance between 0-20
y = -0.087x + 30.44
Distance(cms)
Debug Value
Series1
Linear (Series1)
0
5
10
15
20
25
0
100
200
300


Distance between 20-40
y = 0.0035x2 - 1.0518x + 98.85
Distance(cms)
Debug Value
Series1
Poly. (Series1)
0
5
10
15
20
25
30
35
40
45
0
50
100
150


Distance between 40-70
y = 0.018x2 - 3.2558x + 181.83
Distance (cms)
Debug Value
Series1
Poly. (Series1)
0
10
20
30
40
50
60
70
80
0
20
40
60
80

COMMUNICATION SYSTEMS

We decided up on wire less communication as this plays a major role in real life situations and it gives us a wide operating range. Two way communications was setup using RF communication and IR communication.

IR Communication

The asynchronous serial IR communication was achieved by using Fire-Stick II. The Fire stick was placed on the mobile robot while the IR receiver was placed on the Base station. The connections are made as shown in the circuitry diagram and pictorial diagrams are as shown in the hardware section.

IR communication works on the principle of line of sight. As long as the transmitter is in line of sight of the receiver there will be a proper communication between the base station and mobile robot. In order to achieve this receiver needs to be mounted at a very high position, which ensures that it will catch the signals being send by the transmitter. More detail insight of the IR Communication is available at [2].

RF Communication

The asynchronous serial RF communication was achieved by using RF transmitter with antenna and RF receiver with antenna. The transmitter was mounted on the base station and receiver was mounted on the mobile robot. It works basically on the radio signal frequencies. More detail insight of the RF Communication is available at [3].

OBJECT DETECTION SYSTEM

The Boe-Bot uses infrared LED's .They emit infrared, and in some cases, the infrared reflects off objects, and bounces back in the direction of the Boe-Bot. The eyes of the Boe-Bot are the infrared detectors. The infrared detectors send signals to the BASIC Stamp indicating whether or not they detect infrared reflected off an object.

The IR detectors have built-in optical filters that allow very little light except the 980 nm. infrared that we want to detect onto its internal photodiode sensor. The infrared detector also has an electronic filter that only allows signals around 38.5 kHz to pass
through. In other words, the detector is only looking for infrared flashed on and off at 38,500 times per second. This prevents interference from common IR interference sources such as sunlight and indoor lighting. Sunlight is DC interference (0 Hz), and house lighting tends to flash on and off at either 100 or 120 Hz, depending

on the main power source in the country where you reside. Since 120 Hz is way outside the electronic filter's 38.5 kHz band pass frequency, it is, for all practical purposes, completely ignored by the IR detectors. Circuitry for this is shown in the circuits page.

MINE DETECTION SYSTEM

A photo resistor is a light-dependent resistor (LDR) that covers the spectral sensitivity similar to that of the human eye. The active elements of these photo resistors are made of Cadmium Sulfide (CdS). Light enters into the semiconductor layer applied to a ceramic substrate and produces free charge carriers. A defined electrical resistance is produced that is inversely proportional to the illumination intensity. In other words, darkness produces high resistance, and high illumination produces very small amounts of resistance.

When implementing it to detect mine, the concept was simple that when it encounters a black spot amidst a white the value theoretically should turn 0. Despite us trying to cover it completely it didn't go to 0 as we couldn't keep out ambient light completely.

However we observed that it gave the incorrect value consistently hence we could easily calibrate it for our purpose. Such sort of manipulating of code can actually resolve some hardware problems and vice versa.

WORKING OF THE CODE

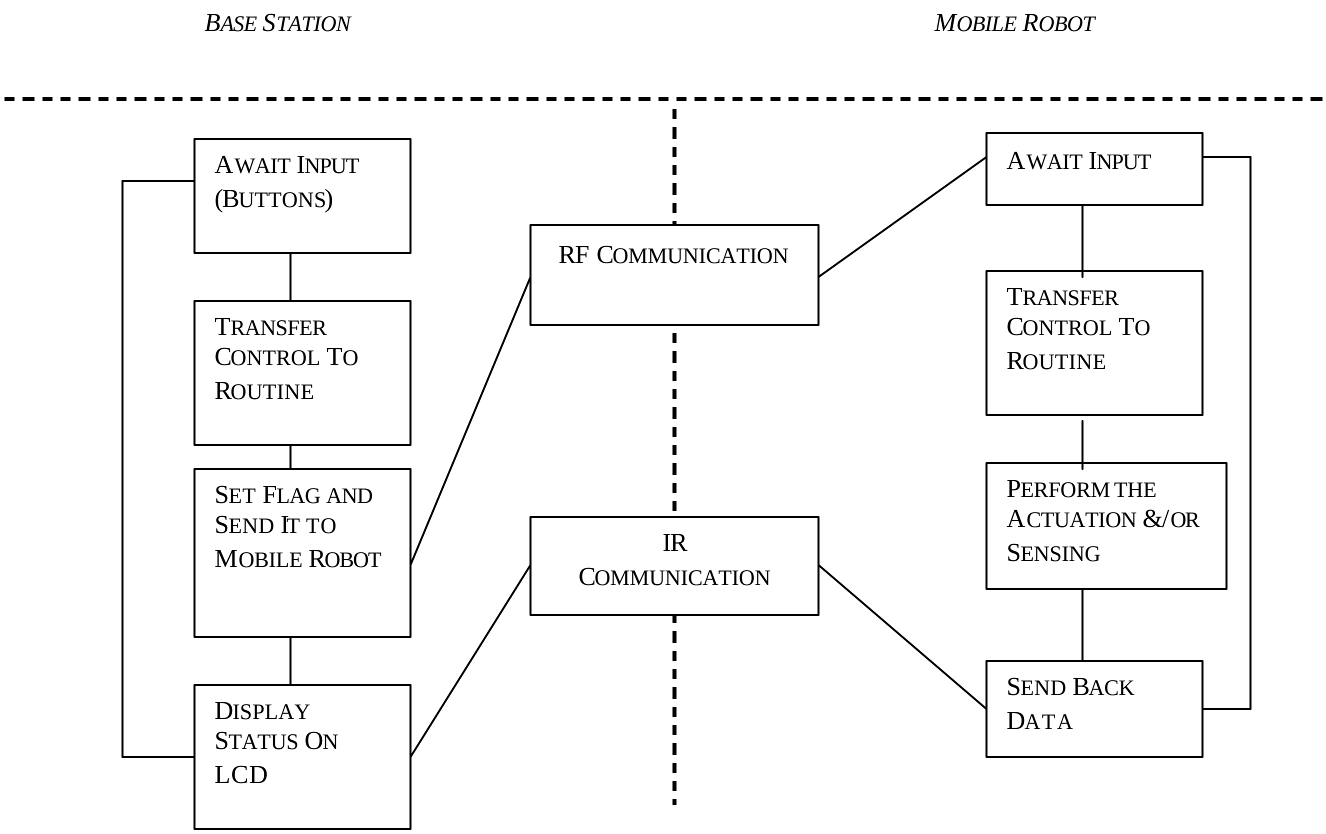
As expected there are two sets of codes for each one of the processors on board the base station and the mobile robot.

The base station robot has all the functionalities to control the robot. The code is highly dependent on a set of flags which are set and reset on commencement and wrap up of an operation. On setting an appropriate flag it sends out the information to the mobile robot which is awaiting instructions. Once this is done the code waits for any response or update from the robot in terms of IR communication.

Whilst on the mobile robot all the actuation and sensing takes place. It keeps on moving or collecting data using sensor and when a desired information is received e.g. mine is detected it sends it back to the base station and awaits further instructions.

When we tried to initiate the variables for actuation on the base station there seemed to be a lag in resending data for every loop and hence the robot moved in jerks. This was logical as there was extra pause induced in the system via the communication channel.

LOGICAL FLOW OF THE CODE



WORKING OF THE ROBOT

The first task the robot was made to do was to detect mines; this type of mine detection was accomplished by using a photo resistor as discussed. Once a mine was detected the BOT would automatically stop and with the press of another button the distance from the base station was reported. The distance was reported using an infrared detector. After the mine was detected it was time for the BOT to return back to its original position and continue the same process in a different row. By pressing the automatic mode button again the BOT would return to its original position and move over one row and await another command. At this point you are ready to continue the process of mine detection again. If for some reason the BOT is not lined up and ready to go for mine detection in the second row you are able to give it manual commands to get it to turn left, right or move forward or backward. If for some reason there is an obstacle in the way the BOT would also be able to maneuver around the obstacle while still detecting mines. If the BOT detected a mine while maneuvering around an obstacle the BOT would automatically stop.

The User Manual section would give a feel of how the entire system operates.

WORKING OF CODE FOR THE AUTONOMOUS MODE

Once the robot is activated in the autonomous mode it starts to measure distance in front of it and if it is above a threshold of 20cm it continues in that direction. As soon as the obstacle comes inside 20cm the evasive mode is activated. It immediately takes a turn of 90 degrees and starts compensating for the turn by turning left in increments of 15 degrees while still checking for obstacles. As soon as it finds a clear way it follows it.

This logic gave it enough functionality to find its way out of mazes provided distance measurements are being taken correctly i.e. the measurement is surface dependent. Certain types of surfaces give incorrect results (will be discussed in issues).

USER MANUAL

The robot initially waits for a button press input. Depending on the input, the robot switches to one of the modes. The LCD display shows

Waiting…..

ROUTINE

Manual Mode

The robot indicates the direction of travel by updating the LCD display. For eg.

Left

Trigger

Pressing Buttons D0 to D3

Options within mode

Pressing D0 navigates the robot in the forward direction, D1 in the Reverse Direction, D2 to the left and D3 to the right.

Other modes possible from this loop

Can switch to the Detect Mine routine by pressing buttons D4 to D6 and revert back here. Can also go to the autonomous mode from here, but since the robot is on its own, avoiding obstacles and locating mines, detection of a mine only triggers the next process (User input on mine detection)

ROUTINE

SYSTEMATIC SEARCH MODE

The LCD displays the mine detect mode by displaying.

Detecting Mine

The display is updated to exhibit the current status like “Mine detected” and “Returning to base”.

Trigger

Pressing Buttons D4 to D6

Options within mode

Pressing D4 sets it to perform a grid search systematically looking for mines

D5 is to measure the distance relative to the base station when a mine has been detected. D6 commands the robot to the base station.

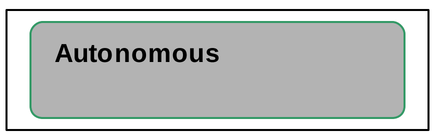
Other modes possible from this loop

Can switch to the Manual routine by pressing buttons D0 to D3 and revert back here. Can also go to the autonomous mode from here, but since the robot is on its own, avoiding obstacles and locating mines, detection of a mine only triggers the next process (User input on mine detection).

ROUTINE

Autonomous Search Manual

The LCD displays the following to show the robot's current mode



The user can know from the LCD display if the mine path is clear or if an object is detected by the messages "Area is clear" or "Object detected"

Trigger

Pressing Button D7

Options within mode

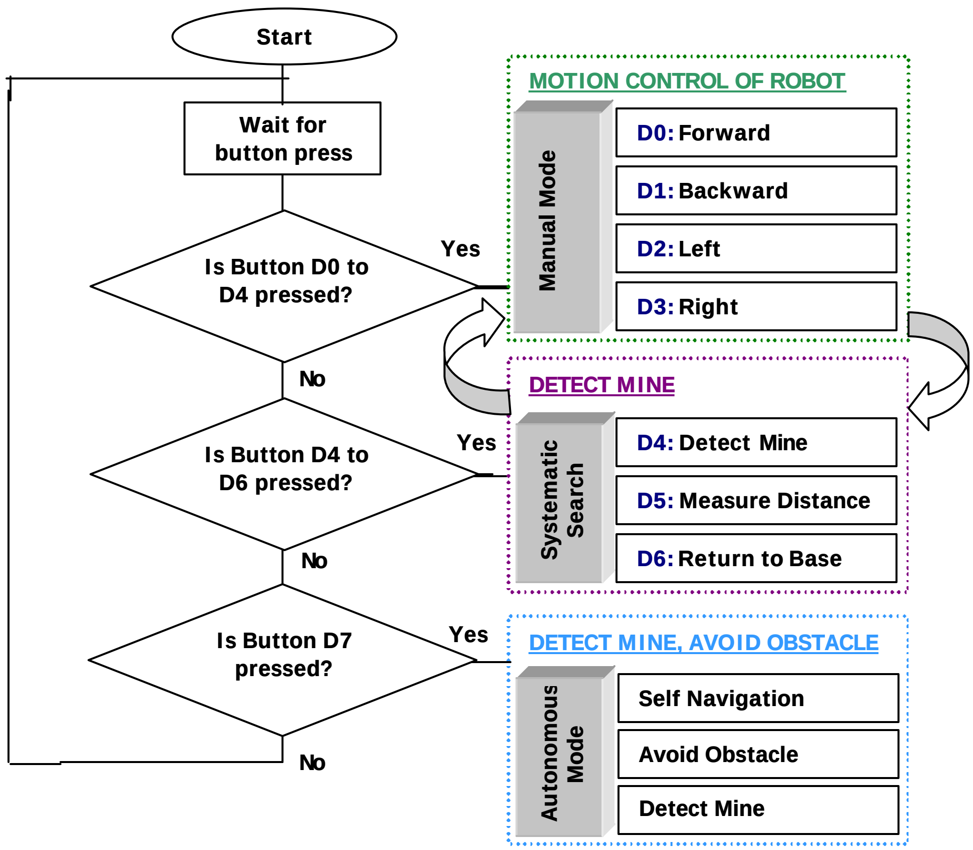
The robot is on its own in this mode, sensing obstacles and avoiding them, taking the free path and locates the mines.

Other modes possible from this loop

On detection of mine, the user has the choice to trigger any of the three modes, the manual mode or the systematic search mode or to restart the autonomous working.

Software:

Flow Chart:



Source Code:

```
'*********************************************************
'{$STAMP BS2}
'*********************************************************

'**********************************************************
'Code For a base station which controls the action of a
'mobile robot which mimics the performance of a mine detector.

'This was implemented as a final project for MAE 576 mechatronics
'by Batch B Talib,Vamsi,Srinivas,David, Jairam.
'For Queries please contact tsb2@buffalo.edu
'**********************************************************

'*********************************************************
'Program for Base Station Functionalities.
'*********************************************************


'**********************************************************
'Declarations
'**********************************************************

time           VAR    byte

DAT            VAR    word   'Data send by Wireless Communication

synch          VAR    Byte  'Variable For Authentication

junk           VAR    Byte

'**********************************************************
'General Declarations
'**********************************************************

i                     VAR   Word

distance              VAR   Word

Obj_Detect_Right      VAR   Bit

Obj_Detect_Left       VAR   Bit

DetectRight           VAR   Bit

DetectLeft            VAR   Bit

MoveLeft              VAR   Bit

MoveRight             VAR   Bit

MoveStraight          VAR   Bit

counter               VAR   Word

measure               VAR   Bit

auto                  VAR   Bit

d                     VAR   Byte

base_return           VAR   Bit

base_returned         VAR   Bit

countpulses           VAR   Word

'*****************************************************************************

'*******************************************************************************
'LCD Variables
```

```
'***************************************************************************
E            CON    3             ' LCD Enable pin (1 = enabled) FOR DIR H
RS           CON    2             ' Register Select (1 = char)FOR DIR H
LCDbus       VAR    OutB   ' 4-BITLCD data bus (if connecting to 8-15) pins
ClrLCD       CON    $01    ' clear the LCD
CrsrHm       CON    $02    ' move cursor to home position
CrsrLf       CON    $10    ' move cursor left
CrsrRt       CON    $14    ' move cursor right
DispLf       CON    $18    ' shift displayed chars left
DispRt       CON    $1C    ' shift displayed chars right
DDRam        CON    $80    ' Display Data RAM control
MoveCrsr     CON    %10000000     'Move Cursor to this position
CGRam        CON    $40        'Custom Character RAM
Line2        CON    $C0
char         VAR    Byte   ' character sent to LCD
index        VAR    Byte   ' loop counter
photodetect  VAR    Word
mine         VAR    Bit
Mine_Detect  VAR    Bit
fwd          VAR    Bit
bwd          VAR    Bit
lft          VAR    Bit
rt           VAR    Bit

'***************************************************************************
'Messages
'***************************************************************************
Msg1         DATA "Waiting........",0
Msg2         DATA "Forward........",0
Msg3         DATA "Backward       ",0
Msg4         DATA "Left          ",0
Msg5         DATA "Right         ",0
Msg10        DATA "Autonomous     ",0
Msg11        DATA "MeasureDistance",0
Msg12        DATA "Detecting  Mine",0
Msg13        DATA "Mine   Detected",0
Msg14        DATA "Area  is  Clear",0
Msg15        DATA "Object Detected",0
Msg16        DATA "Returning  Base",0
Msg17        DATA "Ready  To    Go",0
```

```
'******************************************************************************

'*********************************************************
'Button Control Declarations
'*********************************************************
Btn0            CON 8

Btn1            CON 9

Btn2            CON 10

Btn3            CON 11

Btn4            CON 12

Btn5            CON 13

Btn6            CON 14

Btn7            CON 15

swData          VAR Byte
'*********************************************************
'Initializing Variable
'*********************************************************

swData = 0

time   = 250

DIRH   =%00000000

DIRL   =%11111111

synch = "Z"

Obj_Detect_Right=1

Obj_Detect_Left=1

DetectRight=1

DetectLeft=1

MoveLeft=0

MoveRight=0

MoveStraight=0

distance=0

measure=0

auto=0

mine=0

base_return=0

base_returned=0

countpulses=0

fwd=0

bwd=0

lft=0

rt=0
'***********************************************************
'Main Subroutine Where It Awaits a Button To Be Pressed
'***********************************************************
```

```
Main:
        Obj_Detect_Right=1  ‘Reseting Appropriate Flags
        Obj_Detect_Left=1
        DetectRight=1
        DetectLeft=1
        measure=0
        auto=0
        mine=0
        fwd=0
        bwd=0
        lft=0
        rt=0
        base_return=0
        base_returned=0

        char = ClrLCD
        GOSUB LCD_Command
        index = Msg1
        GOSUB ReadCharacter
        PAUSE 5

Waiting:
         ‘  Buttons To Initiate activities
        BUTTON Btn0,0,255,10,swData,1,Forward
        BUTTON Btn1,0,255,10,swData,1,Backward
        BUTTON Btn2,0,255,10,swData,1,Left
        BUTTON Btn3,0,255,10,swData,1,Right
        BUTTON Btn4,0,255,10,swData,1,Detect_Mine
        BUTTON Btn5,0,255,10,swData,1,Measure_D
        BUTTON Btn6,0,255,10,swData,1,Return_Base
        BUTTON Btn7,0,255,10,swData,1,Autonomous
GOTO Waiting
'*********************************
'Forward Manual Move
'*********************************

Forward:

index=Msg2  ‘Display Message on the LCD
GOSUB ReadCharacter
```

```
fwd=1

'Send Instruction to mobile robot

serout[0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]
        '

GOTO Main

'********************************
'Backward Manual Move
'********************************

Backward:

Index = Msg3

GOSUB ReadCharacter

debug "back"

bwd = 1

serout 0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]

GOTO Main

'************************
'Left Manual Move
'**********************

Left:

index=Msg4

GOSUB ReadCharacter

lft=1

serout 0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]

GOTO Main


'***************************
'Right Manual Move
'***************************


Right:

index=Msg5

GOSUB ReadCharacter

rt=1

serout 0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]

GOTO Main


'*************************************************************************************
'******Autonomous Mode******

'This subroutine Kicks the the robot into autonomous mode
'and relinquishes control of the base station.
'During This Time it checks for mines as well and reports progress to the Base
'Station which is subsequently displayed on the LCD.

'**************************************************************************************

Autonomous:

index=Msg10
```

```
GOSUB ReadCharacter
auto=1 ‘ Set Flag

serout  0,16780,[junk,synch,measure,auto,mine,base_return]  ‘Sends  Out  Command  TO  GO
Autonomous

serin 1,813,1000,Nodata9,[ wait ("CB"),DetectRight,photodetect]  ‘Waits For Mine to be
detected

Nodata9:   ‘If No data is received it terminate the wait.

PAUSE 1000

auto=0 ‘Reset Flag

GOTO Main


'*********************************************************************
'This Routine Measures Distance Using the IR ranging sensor and Displays it
‘ On the LCD.

'********************************************************************

Measure_D :

distance=0

measure=1

index=Msg11

GOSUB ReadCharacter

serout 0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]

measure=0

serin 1,813,5000,Nodat,[wait ("CB"),distance]

Nodat:

d.Nib1 = distance/10//10  ‘Separates the Digit For Displaying on the LCD

d.Nib0 = distance/1//10

GOSUB DisplayDistance

PAUSE 2000

index=0

GOTO Main


'**************************************************************
'This Mode Starts the Straight Search For Detecting a Mine.
'**************************************************************

Detect_Mine :

debug"mine"

mine=1

index=Msg12

GOSUB ReadCharacter

serout 0,16780,[junk,synch,measure,auto,mine,base_return]

mine=0

serin 1,813,[ wait ("CB"),Mine_Detect,countpulses]
```

```
‘Waits till mine is detected also countpulses keeps track of distance Traveled

IF Mine_Detect=1 THEN M_Detected  ‘Conditional Statements if Mine Or Object is Detected

IF Mine_Detect=0 THEN O_Detected

PAUSE 500

GOTO Main

'*****************************************************************
‘If Robot Reports back a mine it Displays the Same on The LCD
‘*****************************************************************

M_Detected:

PAUSE 500

index=Msg13

GOSUB ReadCharacter

PAUSE 1000

mine=0

GOTO Main

'*****************************************************************
‘If Robot Reports back an object it Displays the Same on The LCD
‘*****************************************************************

O_Detected:

PAUSE 500

index=Msg15

GOSUB ReadCharacter

PAUSE 1000

mine=0

GOTO Backward

'*****************************************************************
'It Signals the Robot to Return Base
'*****************************************************************

Return_Base:

index=Msg16

GOSUB ReadCharacter

base_return=1

‘Sends the Appropriate Pulses to the Robot

serout 0,16780,[junk,synch,measure,auto,mine,base_return,countpulses,fwd,bwd,lft,rt]

serin 1,813,[ wait ("CB"),base_returned]

PAUSE 500

index=Msg17

GOSUB ReadCharacter

PAUSE 1000

countpulses=0
```

```
GOTO Main

'***************************************************
 ‘Routine TO Display Distance on the LCD
'***************************************************


DisplayDistance:

        char = MoveCrsr + 64
        GOSUB LCD_Command

        char = d.Nib1 + 48
        GOSUB LCD_Write

        char = d.Nib0 + 48
        GOSUB LCD_Write


RETURN


'*****************************************************************************************
****
'LCD Read The character from Messages and Sets the cursor and all in the LCD
'*****************************************************************************************
****


ReadCharacter:

        char=CrsrHm

        GOSUB LCD_Command

        Read_Char:

                READ index, char ' get character from EEPROM

                IF (char = 0) THEN Msg_Done ' if 0, message is complete

                GOSUB LCD_Write ' write the character

                index = index + 1 ' point to next character

                GOTO Read_Char ' go get it

                Msg_Done: ' the message is complete

                index=0

RETURN ' do it all over


'*****************************************************************************************
******
'Initialize The LCD
'*****************************************************************************************
******

LCD_Init:

        PAUSE 500      ' let the LCD settle

        LCDbus = %0011        ' 8-bit mode

        PULSOUT E, 1

        PAUSE 5

        PULSOUT E, 1
```

```
        PULSOUT E, 1
        LCDbus = %0010        ' 4-bit mode
        PULSOUT E, 1

        char = %00001100 ' disp on, crsr off, blink off
        GOSUB LCD_Command
        char = %00000110 ' inc crsr, no disp shift
        GOSUB LCD_Command
RETURN
'************************************************************
'Low the RS to Write to The LCD
'************************************************************
LCD_Command:
        LOW RS ' enter command mode

'*************************************************************
'Write to the LCD
'*************************************************************

LCD_Write:
        LCDbus = char.HighNib ' output high nibble
        PULSOUT E, 1 ' strobe the Enable line
        LCDbus = char.LowNib ' output low nibble
        PULSOUT E, 1
        HIGH RS ' return to character mode
RETURN

'****************************************************************************************
***********
```

```
'**************************************************************************************
***********
'**************************************************************************************
***********
'CODE FOR MOBILE ROBOT
'**************************************************************************************
***********
'**************************************************************************************
***********

'{$STAMP BS2}

'*****************************************
'Declarations
'*****************************************

pulse_count    var    word ' Declare a variable for counting.

baud           con    813 ' 1200 Baud, (NON-INVERTED)

f              var    word ' Declare a variable for counting.

b              var    word ' Declare a variable for counting.

p1             var    word

synch          VAR    Byte

junk           VAR    Byte

v              var    byte

distance       VAR    byte   'Variable used to store DIRRS result

countpulses    VAR    Word   ' Counters

counter        VAR    Word

i              VAR    Byte

T              VAR    Byte

'***************************************************************
'Variables For DIRRS
'***************************************************************

clock          CON    3

datain         CON    4

OUTPUT         clock  'Pin 0 connects to Pin 2 of DIRRS Plus

INPUT          datain 'Pin 1 connects to Pin 4 of DIRRS Plus

'**************************************************************
'Declaration of Flags
'**************************************************************

right_IR_det   var     bit

measure_mode   VAR    bit

auto_mode      VAR    bit

photo          VAR    word

MoveRight      VAR    Bit

MoveLeft       VAR    Bit

dist           VAR    byte

mined          VAR    Bit
```

```
mine_d         VAR    Bit

basereturn     VAR    Bit

back           VAR    Bit

fwd            VAR    Bit

bwd            VAR    Bit

lft            VAR    Bit

rt             VAR    Bit

'****************************************************
' Initialization
'***************************************************
low 12 ' Set P12 and 13 to output-low.

low 13

synch = "Z"

junk = 126

right_IR_det=1

distance=0

output  1      ' Set all I/O lines sending freqout

output   13    ' signals to function as outputs

‘*****Initializing Flags************************

measure_mode=0

auto_mode=0

MoveRight=0

MoveLeft=0

mined=0

mine_d=0

basereturn=0

countpulses=0

fwd=0

bwd=0

lft=0

rt=0

'******************************************************
'Main Routine
'******************************************************
Main:

‘Reset Variables


       fwd=0

       bwd=0

       lft=0

       rt=0
```

```
        distance=0
        counter=0
        countpulses=0
        dist=0
        measure_mode=0
        auto_mode=0
        mine_d=0
        basereturn=0
        ‘Waits for command from base station to proceed
        SERIN 16780,[WAIT(SYNCH),measure_mode,auto_mode,mine_d,basereturn,
         countpulses,fwd,bwd,lft,rt]

        ‘Goes to Appropriate Routine According to flag being set from the base station.
        IF measure_mode=1 THEN Measure_Distance
        IF auto_mode=1 THEN Autonomous_Temp
        IF mine_d=1 THEN Mine_Detect
        IF basereturn=1 THEN Return_to_base
        IF fwd=1 THEN Forward
        IF bwd=1 THEN Backward
        IF lft=1 THEN Left
        IF rt=1 THEN Right

GOTO Main
END
'*******************************************************************************************
***********
'This routine is activated by the base station to tell the robot to conduct a
‘straight line search for mines.
‘During this routine it checks for mines and objects as well.
'*******************************************************************************************
***********
Mine_Detect:
        right_IR_det=1   ‘Sets Flag
        mined=0
        countpulses=countpulses+1
        f=904        ‘Moves Forward
        b=500
        pulsout 12, f
        pulsout 13, b
        '*********************** Detecting a Mine ***************
        high 2
```

```
        pause 5

        rctime 2,1,photo

        photo=photo/100 'Calibrated Output For

        '************************Detecting an Object*************

        freqout 14, 1, 38850

        right_IR_det = in15

        debug ?right_IR_det

        IF right_IR_det=0 THEN Ob_Detected

        IF photo>=10   THEN   Mine_Detected

GOTO Mine_Detect

'**************************************************************
'If Mine is Detected the Base Station is Notified
'**************************************************************
 Mine_Detected:

        mined=1

        SEROUT 9,17197 ,["CB",mined,countpulses]

GOTO Main

'********************************************************
' If Object is Detected the Base Station is Notified
'******************************************************
Ob_Detected:

        mined=0

        SEROUT 9,17197 ,["CB",mined,countpulses]

GOTO Main

'********************************************************************
'Puts Robot in a autonomous mode where it can search for mines and
'Intelligently navigate as well.
'********************************************************************

Autonomous_Temp:

GOSUB  Measure_Distance_one   'Repeating  Measurement  of  Distance  To  assure  a  correct
reading is taken
GOSUB Measure_Distance_one    'initially
GOSUB Measure_Distance_one


GOSUB here

Autonomous:

        GOSUB Measure_Distance_one

        here:

                IF dist>20 THEN straight

                IF dist<=20 THEN decide

GOTO Autonomous

'*************************************************************************************
'This routine is part of autonomous routine where it directs the bot to go straight
'if there are no obstacles.
'*************************************************************************************

Straight:
```

```
        f=500

        b=904

        p1=20

        FOR pulse_count=1 TO p1

                pulsout 12, f

                pulsout 13, b

        NEXT

GOSUB Autonomous

'***************************************************************************
'Decision is take here whether it has to spin or go straight after taking
‘Distance Measurements.
'***************************************************************************

        decide:

                GOSUB spin   % Spins to avoid an obstacle

        measure_again:

                pause 100

                GOSUB Measure_Distance_one

                GOSUB Look  % If nothing is close by it goes straight

                for i= 1 to 10

                        GOSUB Look

                        GOSUB spin1

                        GOSUB Measure_Distance_one

                        PAUSE 500
                NEXT

GOTO Autonomous

'**********************************************
'Spins Right by 90 degrees
'*************************************************
spin:

        f=1000

        b=1000

        p1=30

        FOR pulse_count=1 TO p1

                pulsout 12, f

                pulsout 13, b

        NEXT
GOTO measure_again

'************************************************************************************
'Recovers from the Spin in small increments and spanning the area at the same time
'************************************************************************************
spin1:

        f=500

        b=500
```

```
        p1=10
FOR pulse_count=1 TO p1
        pulsout 12, f
        pulsout 13, b
NEXT
RETURN
'*******************************************************
'Looks For Distance Satisfaction
'*******************************************************
Look:
        IF dist>20 THEN straight
RETURN
'******************************************************
'Subroutine which Initiates Measurement of Distance
'******************************************************

Measure_Distance:

GOSUB Measure_Distance_one
GOSUB Measure_Distance_one
GOSUB Measure_Distance_one

SEROUT 9,17197,["CB",dist]

GOTO Main
'******************************************************
'Measure_Distance_One
'******************************************************
Measure_Distance_one:
        High clock            'Start Vin must have high-low transition
        Pause 3               'Wait for high to be recognized
        Low clock
        For T= 1 to 100       'Begin 30 ms wait
               PAUSE 1
               If IN4=1 Then JUMP_first
                              'If Vout is HIGH, measurement is complete -jump
        Next                  'Measurement not complete, wait another 1 ms
JUMP_first:
        Shiftin datain,clock,2,[distance\8]
        PAUSE 500
        “Calibration of Distance According to the value of DIRRS
        IF distance <=67 THEN calib11
        IF distance <=109 THEN calib21
        IF distance<=147  THEN calib31
        IF distance <=200 THEN calib41
        IF distance<=255  THEN calib51
RETURN
```

```
' Calibration formula

calib11:

        dist=1974-5375/100*distance+(distance*distance*375/1000)

        IF dist>80 THEN set
RETURN

calib21:

        dist=275-(447*distance/100)+(distance*distance*2/100)
RETURN

calib31:

        dist = 98-distance+(distance*distance*3/1000)
RETURN

calib41:

        dist =12

RETURN

calib51:
        dist=10

RETURN

set:

dist=80

RETURN

'**********************************************
'Initiates Return_to_base
'**********************************************

Return_to_base:

        f=500

        b=904     '1000

        pulsout 12, f

        pulsout 13, b

        counter=counter+1

        IF counter >= countpulses THEN GOBACK 'tune to make it come back to the same loc.

        back=0

        GOTO Return_to_base

'**********************************************
'Stops When it reaches back and switches lanes .
'**********************************************

        PAUSE 2000

        back =1

        f=1000

        b=1000

        'rt turn

        p1=95

        FOR Pulse_Count=1 TO p1
```

```
		pulsout 12, f
		pulsout 13, b
	NEXT
	PAUSE 2000
	f=1000
	b=500
	p1=105  'step size
	FOR Pulse_Count=1 TO p1
		pulsout 12, f
		pulsout 13, b
	NEXT
	PAUSE 2000
	f=500          'back to position
	b=500
	p1=150
	FOR Pulse_Count=1 TO p1
		pulsout 12, f
		pulsout 13, b
	NEXT
	SEROUT 9,17197 ,["CB",back]
	back=0
GOTO Main
'*****************************************************
'Routines Used When Manually Operated From The base station
'*****************************************************

'**********************************************
'Forward
'*********************************************
Forward:
	f=904
	b=500
	p1=25
	FOR pulse_count=1 TO p1
	pulsout 12, f
	pulsout 13, b
	next
GOTO Main
'**********************************************
'Backward
'*********************************************
Backward:
	f=500
```

```
        b=904

        p1=25

        FOR pulse_count=1 TO p1

        pulsout 12, f

        pulsout 13, b

        next

GOTO Main
'**********************************************
'Left
'*********************************************
Left:
        f=500

        b=500

        p1=10

        FOR pulse_count=1 TO p1

        pulsout 12, f

        pulsout 13, b
        next

GOTO Main
'**********************************************
'Right
'*********************************************
Right:

        f=904

        b=904

        p1=10

        FOR pulse_count=1 TO p1

        pulsout 12, f

        pulsout 13, b

        next

GOTO Main
```

ISSUES & DISCUSSION

As we discussed earlier the working of the decoupled systems was fine, the systems were picture perfect. The issues started when we actually attempted to implement the entire system .However it was a great learning experience and troubleshooting made us understand the specifics of the subsystems which we might have otherwise overlooked.

- Firstly the IR communication didn’t work when ever line of sight was not maintained; this was very uncomfortable as it restricted our space of operation. This was resolved by mounting the receiver as a satellite looming over the area of operation and the transmitter was turned upwards to communicate with it.

- The IR ranging sensor worked fine on the base station but when we put it on the mobile robot, it stopped working. After a lot of debugging we figured that DIRRS didn’t work with BS2e which was on the robot .Hence we had to change microprocessors.

- Secondly the IR ranging sensor seemed to work fine for an hour and then started giving erratic values. This was essentially because it consumed batteries at a very high rate and hence batteries simply died down as far as the ranging sensor was concerned, it still managed to move the mobile robot pretty well.

- Calibration was a major issue. Every time there was a slight change in the ambient conditions or if something was added on the robot, the motion calibration changed. This was resolved simply by repeating calibrations and developing the entire system before calibrating it.

- Another aspect of concern was that maintaining the right pauses for measuring or use of sensors; this was a programming issue and was solved after considerable trial and error.

- In order to use IR Ranging Sensor, the sensor has to be mounted perfectly horizontal and the object whose distance its measuring should have a nearly if not perfectly vertical surface, else it was giving an erroneous results. Care was taken to incorporate these two in to the system.

CONCLUSION

A distributed sensing framework was successfully implemented. Further some an application was associated with it which was one of our primary concerns, that what we create should make some practical sense. Mine detection infact seems like a probable application where such a system would be ideal.

Further the concepts and issues pertaining to integration of varied subsystem were understood and the project was successfully implemented.

CONTRIBUTION OF EACH MEMBER

Team members were:

David Pericak
Jairam Ramaswamy
Srinivas Sundaragopal
Talib Bhabhrawala
Vamsi Krishna

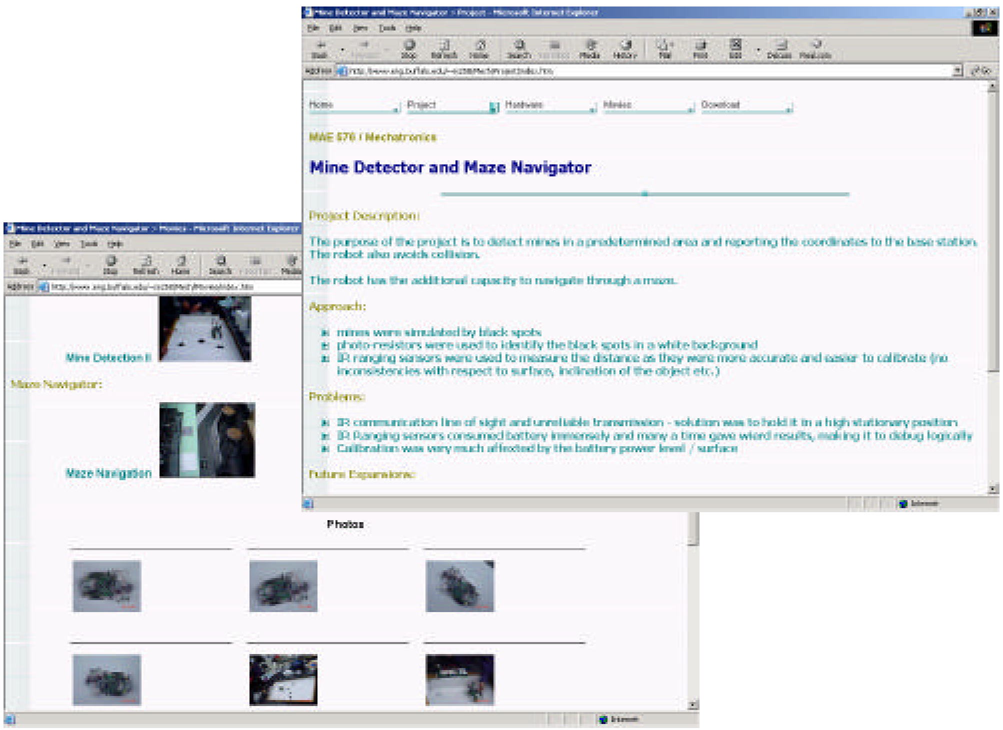
This being a team based final project, and the project specifics were open ended, we did not modularize the project. We worked on developing and testing all the features of the robot, then came up with our project proposal and built the robot in a bottom-up approach. All the team members were present and actively participated in each stage of the project. We really performed as a team and it is difficult to assess each person's individual contribution.

WEBSITE

The project web page is available at

http://www.eng.buffalo.edu/~ss258/Mech/Movies/index.htm

We developed a website to highlight the practical aspects and the approaches. It also has hardware specifications, photographs and videos of the model for download.



REFERENCES

[1]: www.HVWTech.com/dirrsplus.htm

[2]: http://www.rentron.com/FS-2.htm

[3]: http://www.rentron.com/Stamp_RF.htm

[4]: Basic Stamp Manual (General Reference)

[5]: BOE BOT Stamp Works Manual (General Reference)