CODE:

TX:

#include <LPC214x.H>

#define GPIO\_Port0s\_IODIR IODIR0

#define GPIO\_Port1s\_IODIR IODIR1

#define Set\_Port0s IOSET0

#define Clear\_Port0s IOCLR0

#define Set\_Port1s IOSET1

#define Clear\_Port1s IOCLR1

#define Port0\_Set IOPIN0

#define Port1\_Set IOPIN1

#define IR (1<<5)

#define PIR (1<<6)

#define Gas (1<<7)

#define buzzer (1<<10)

# define inputs (IR|PIR|Gas)

# define outputs buzzer

#include"lcd.c"

#include"serial\_uart0.c"

#include"I2C.c"

#include"robo.c"

void Delay(unsigned int );

void Delay\_cam(unsigned int time);

void Motor\_Init(void);

int PinStatus\_Port(unsigned char ,unsigned char);

void DTMF\_Data\_Decode(void);

void Passwd\_DTMF(void);

void Check\_Sensors(void);

unsigned int pinstate;

unsigned char x,LCD\_CLEAR=0x01;

/\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_output=1 input=0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*/

int main (void)

{

GPIO\_Port1s\_IODIR=(LCD\_Data|RS|EN);

GPIO\_Port0s\_IODIR=~(inputs);

GPIO\_Port0s\_IODIR= (outputs);

Lcd\_Init();

Init\_UART0(9600);

I2C\_Init();

MEMS\_Init();

Project\_Label();

while (1)

{

Check\_Sensors();

}

}

PinStatus\_Port(unsigned char port,unsigned char pin)

{

if(port==0)

{

x=(Port0\_Set& (1<<pin))?1:0;

}

if(port==1)

{

x=(Port1\_Set& (1<<pin))?1:0;

}

return x;

}

// void Delay(unsigned int time)

// {

// unsigned int i,j;

// for(i=0;i<time;i++)

// for(j=0;j<25000;j++);

void Project\_Label(void)

{

Lcd\_Data\_Str(1,1,"Indoor Intrusion");

Lcd\_Data\_Str(2,1,"Detecting System");

Delay(300);

Lcd\_Data\_Chr(0,0,0,LCD\_CLEAR);

}

if(PinStatus\_Port(0,5))

{

UART0\_TX\_Str ("I");

Lcd\_Data\_Str(2,1,"IR Area Distrub ");

Set\_Port0s=buzzer;

Delay(200);

Clear\_Port0s=buzzer;

}

if(!PinStatus\_Port(0,6))

{

UART0\_TX\_Str ("P");

Lcd\_Data\_Str(2,1,"PIR Area Disturb");

Set\_Port0s=buzzer;

Delay(200);

Clear\_Port0s=buzzer;

}

if(!PinStatus\_Port(0,7))

{

UART0\_TX\_Str ("G");

Lcd\_Data\_Str(2,1,"Gas Detected ");

Set\_Port0s=buzzer;

Delay(200);

Clear\_Port0s=buzzer;

}

else

{

Lcd\_Data\_Str(2,1,"All Sensor Clr ");

Delay(200);

}

}

void Delay\_cam(unsigned int time)

{

unsigned int i,j;

for(i=0;i<time;i++)

for(j=0;j<1200;j++);

}

/\*

#define RS (1<<24) //24

#define EN (1<<25) //25

#define LCD\_Data (0xFF)<<16

unsigned char LCD\_C=0,LCD\_D=1,Wr\_Data=1;

int Delay(unsigned int );

void Lcd\_Init(void);

int Lcd\_Data\_Chr(unsigned char ,unsigned char ,unsigned char ,unsigned char);

int Lcd\_Data\_Str(unsigned char ,unsigned char ,unsigned char temp[]);

int Lcd\_Wr(unsigned char );

\*/

/\*

void Lcd\_Init(void)

{

unsigned char LCD\_2\_LINE=0x38;

unsigned char LCD\_CLEAR=0X01;

unsigned char DISPLAY\_ON=0X0E;

unsigned char LCD\_CURSOR\_OFF=0x0C;

Lcd\_Data\_Chr(0,0,0,LCD\_2\_LINE);

Delay(5);

Lcd\_Data\_Chr(0,0,0,DISPLAY\_ON);

Delay(5);

Lcd\_Data\_Chr(0,0,0,LCD\_CURSOR\_OFF);

Delay(5);

Lcd\_Data\_Chr(0,0,0,LCD\_CLEAR);

Delay(5);

}

Lcd\_Data\_Chr(unsigned char RS1 ,unsigned char line,unsigned char position,unsigned char temp1)

{

unsigned char x;

if(RS1==0)

{

Set\_Port1s= (temp1<<16)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

if(RS1==1)

{

if(line==1)

{

x=0x7f+position;

Set\_Port1s= (x<<16)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

if(line==2)

{

x=0xbf+position;

Set\_Port1s= (x<<16)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

Set\_Port1s= (temp1<<16)&LCD\_Data;

Lcd\_Wr(LCD\_D);

Clear\_Port1s=LCD\_Data;

Delay(15);

}

}

Lcd\_Data\_Str(unsigned char line1,unsigned char position,unsigned char temp[])

{

unsigned int p;

unsigned char t;

if(line1==1)

{

p=0x7f+position;

Lcd\_Data\_Chr(0,0,0,(Set\_Port1s=(p<<16)&LCD\_Data));

}

if(line1==2)

{

p=0xbf+position;

Lcd\_Data\_Chr(0,0,0,(Set\_Port1s=(p<<16)&LCD\_Data));

}

while(temp[t]!='\0')

{

Set\_Port1s= (temp[t]<<16)&LCD\_Data;

Lcd\_Wr(LCD\_D);

t++;

Clear\_Port1s=LCD\_Data;

Delay(15);

}

}

Lcd\_Wr(unsigned char r)

{

if(r==1)

{

Set\_Port1s= RS;

Set\_Port1s= EN;

Delay(1);

Clear\_Port1s= EN;

}

if(r==0)

{

Clear\_Port1s= RS;

Set\_Port1s= EN;

Delay(1);

Clear\_Port1s= EN;

}

}

Delay(unsigned int time)

{

unsigned int i,j;

for(i=0;i<time;i++)

for(j=0;j<25000;j++);

}

\*/

#define RS (1<<16)

#define EN (1<<17)

#define LCD\_Data (0XFF)<<18 //16----18------25

//#define LCD\_Data (0xFFFFFFFF)<<24

unsigned char LCD\_C=0,LCD\_D=1,Wr\_Data=1;

void Delay(unsigned int );

void Lcd\_Init(void);

int Lcd\_Data\_Chr(unsigned char ,unsigned char ,unsigned char ,unsigned char);

int Lcd\_Data\_Str(unsigned char ,unsigned char ,unsigned char temp[]);

int Lcd\_Wr(unsigned char );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Lcd\_Data\_(0-cmd:1-data,line no,position,char to disp on LCD);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Lcd\_Init(void)

{

unsigned char LCD\_2\_LINE=0x38;

unsigned char LCD\_CLEAR=0X01;

unsigned char DISPLAY\_ON=0X0E;

unsigned char LCD\_CURSOR\_OFF=0x0C;

Lcd\_Data\_Chr(0,0,0,LCD\_2\_LINE);

Lcd\_Data\_Chr(0,0,0,DISPLAY\_ON);

Lcd\_Data\_Chr(0,0,0,LCD\_CURSOR\_OFF);

Lcd\_Data\_Chr(0,0,0,LCD\_CLEAR);

}

Lcd\_Data\_Chr(unsigned char RS1 ,unsigned char lines,unsigned char position,unsigned char temp1)

{

unsigned char x;

if(RS1==0)

{

Set\_Port1s= (temp1<<18)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

if(RS1==1)

{

if(lines==1)

{

x=0x7f+position;

Set\_Port1s= (x<<18)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

if(lines==2)

{

x=0xbf+position;

Set\_Port1s= (x<<18)&LCD\_Data;

Lcd\_Wr(LCD\_C);

Clear\_Port1s=LCD\_Data;

}

Set\_Port1s= (temp1<<18)&LCD\_Data;

Lcd\_Wr(LCD\_D);

Clear\_Port1s=LCD\_Data;

}

}

Lcd\_Data\_Str(unsigned char line1,unsigned char position,unsigned char temp[])

{

unsigned int p;

unsigned char t;

if(line1==1)

{

p=0x7f+position;

Lcd\_Data\_Chr(0,0,0,(Set\_Port1s=(p<<18)&LCD\_Data));

}

if(line1==2)

{

p=0xbf+position;

Lcd\_Data\_Chr(0,0,0,(Set\_Port1s=(p<<18)&LCD\_Data));

}

while(temp[t]!='\0')

{

Set\_Port1s= (temp[t]<<18)&LCD\_Data;

Lcd\_Wr(LCD\_D);

t++;

Clear\_Port1s=LCD\_Data;

// Delay(5);

}

}

Lcd\_Wr(unsigned char r)

{

if(r==1)

{

Set\_Port1s= RS;

Set\_Port1s= EN;

Delay(1);

Clear\_Port1s= EN;

}

if(r==0)

{

Clear\_Port1s= RS;

Set\_Port1s= EN;

Delay(1);

Clear\_Port1s= EN;

}

}

void Delay(unsigned int time)

{

unsigned int i,j;

for(i=0;i<time;i++)

for(j=0;j<25000;j++);

}

RX:

#include <LPC2103.H>

#include <string.h>

#define GPIO\_Port0s\_IODIR IODIR

#define Set\_Port0s IOSET

#define Clear\_Port0s IOCLR

#define Port0\_Set IOPIN

#define Restg 0

#define Restd 1

#include "LCD.c"

#include "Serial\_Uart0.c"

#include "Serial\_Uart1.c"

#include "GSM.c"

#include "App.c"

#define Buzzer (1<<7)

int PinStatus\_Port(unsigned char ,unsigned int);

void Check\_Sensors(void);

unsigned char x;

main()

{

GPIO\_Port0s\_IODIR = (LCD\_Data|RS|EN|Buzzer); //output=1 // input=0

Clear\_Port0s=Buzzer;

Lcd\_Init();

Init\_UART0 (9600);

Init\_UART1 (9600);

Init\_UART0\_Interrupt();

GSM\_S900();

Project\_Label();

while(1)

{

Check\_Sensors();

}

}

PinStatus\_Port(unsigned char port,unsigned int pin)

{

if(port==0)

{

x=(Port0\_Set& (1<<pin))?1:0;

}

return x;

}

void Project\_Label(void)

{

Lcd\_Data\_Str(1,1,"Indoor Intrusion");

Lcd\_Data\_Str(2,1,"Detecting System");

Delay(300);

Lcd\_Data\_Chr(0,0,0,LCD\_CLEAR);

Lcd\_Data\_Str(2,1,"All Sensors Clr ");

}

#define Buzzer (1<<7)

int PinStatus\_Port(unsigned char ,unsigned int);

//0-wet

unsigned char iflag1=0,pflag1=0,gflag1=0,mflag1=0,RxData;

void Check\_Sensors(void)

{

RxData=UART1\_RX\_Chr();

if(RxData=='I')

{

Lcd\_Data\_Str(1,1,"IR Distrubed");

Set\_Port0s=Buzzer;

Delay(200);

Clear\_Port0s=Buzzer;

if(iflag1==0)

{

iflag1=1;

Sending\_Message\_For\_Stored\_Misscall("Indoor Intrusion\r\n","Alert....! IR Area Disturbed Pls Check");

}

}

else if(RxData=='P')

{

Lcd\_Data\_Str(1,1,"PIR Distrubed");

Set\_Port0s=Buzzer;

Delay(200);

Clear\_Port0s=Buzzer;

if(pflag1==0)

{

pflag1=1;

Sending\_Message\_For\_Stored\_Misscall("Indoor Intrusion\r\n","Alert....! PIR Area Disturbed Pls Check");

}

}

else if(RxData=='G')

{

Lcd\_Data\_Str(1,1,"GAS Detected ");

Set\_Port0s=Buzzer;

Delay(200);

Clear\_Port0s=Buzzer;

if(gflag1==0)

{

gflag1=1;

Sending\_Message\_For\_Stored\_Misscall("Indoor Intrusion\r\n","Alert....! Gas Leak Detected Pls Check");

}

}

else if(RxData=='M')

{

Lcd\_Data\_Str(1,1,"MEMs Distrubed");

Set\_Port0s=Buzzer;

Delay(200);

Clear\_Port0s=Buzzer;

if(mflag1==0)

{

mflag1=1;

Sending\_Message\_For\_Stored\_Misscall("Indoor Intrusion\r\n","Alert....! MEMs Area Disturbed Pls Check");

}

}

else

{

iflag1=0;

pflag1=0;

gflag1=0;

mflag1=0;

Lcd\_Data\_Str(1,1," All Sensors Clr");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chk\_for\_Arrvd\_Messg--->Collct\_Messg--->Read\_Messg

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Check\_Message(void)

{

if(Message\_Read\_flag==1)

{

Message\_Read\_flag=0;

Message\_Reading();

}

}

//////////////////////////////////////////////////////////////

void Message\_Collect(void)

{

while(UART0\_RX\_Chr()!='S');

while(UART0\_RX\_Chr()!='M');

while(UART0\_RX\_Chr()!=',');

for(j=0;j<2;j++)

{

s[j]=UART0\_RX\_Chr();

}

}

///////////////////////////////////////////////////////////////

void Message\_Reading(void)

{

Disable\_UART0\_Interrupt();

Lcd\_Data\_Str(1,1,"Msg Arivd ");

if((s[0]<=0x39)||(s[0]==0x31&&s[1]==0x39))

{

UART0\_TX\_Str(" AT+CMGR=");

UART0\_TX\_Chr(s[0]);

UART0\_TX\_Chr(s[1]);

UART0\_TX\_Chr(0x0D);

while(UART0\_RX\_Chr()!='+');

while(UART0\_RX\_Chr()!='R');

while(UART0\_RX\_Chr()!='E');

while(UART0\_RX\_Chr()!='A');

while(UART0\_RX\_Chr()!='D');

while(UART0\_RX\_Chr()!=',');

for(j=0;j<15;j++)

{

mess\_phone[j]=UART0\_RX\_Chr();

}

while(UART0\_RX\_Chr()!='+');

while(UART0\_RX\_Chr()!=0x0a);

for(j=0;j<15;j++)

{

mess[j]=UART0\_RX\_Chr();

if(mess[j]=='\r')

{

mess[j]=0x00;

break;

}

}

if((s[0]==0x31)&&(s[1]==0x30))

{

Message\_Deleting();

} // if

Lcd\_Data\_Str(1,1,mess\_phone);

Lcd\_Data\_Str(2,1,mess);

Delay(150);

Message\_String\_Compare();

Project\_Label();

Enable\_UART0\_Interrupt();

}//messg\_read

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Alert\_Sending\_Message\_For\_Misscall

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Sending\_Message\_For\_Stored\_Misscall(unsigned char \*MessBdy,unsigned char \*Mess)

{

Disable\_UART0\_Interrupt();

Lcd\_Data\_Str(2,1,"message sending");

UART0\_TX\_Str ("AT\r\n");

Delay(gsm\_delay);

UART0\_TX\_Str ("AT+CMGS=");

UART0\_TX\_Str(misscall\_phone);

UART0\_TX\_Chr(0x0D);

UART0\_TX\_Chr(0x0A);

while(UART0\_RX\_Chr()!='>');

UART0\_TX\_Str(MessBdy);

UART0\_TX\_Str(Mess);

//GPS\_Values();

Delay(gsm\_delay);

UART0\_TX\_Chr(0x1a);

while(UART0\_RX\_Chr()!='O');

while(UART0\_RX\_Chr()!='K');

Lcd\_Data\_Str(2,1," message sent ");

Delay(300);

Project\_Label();

Enable\_UART0\_Interrupt();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Alert\_Sending\_Message\_For\_Received\_Message

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Sending\_Message\_For\_Messg\_Rev(unsigned char \*Mess)

{

Disable\_UART0\_Interrupt();

Lcd\_Data\_Str(2,1,"message sending");

UART0\_TX\_Str ("AT\r\n");

Delay(gsm\_delay);

UART0\_TX\_Str ("AT+CMGS=");

UART0\_TX\_Str(mess\_phone);

UART0\_TX\_Chr(0x0D);

UART0\_TX\_Chr(0x0A);

while(UART0\_RX\_Chr()!='>');

UART0\_TX\_Str(Mess);

//UART0\_TX\_Str(Gpslink);

Delay(gsm\_delay);

UART0\_TX\_Chr(0x1a);

while(UART0\_RX\_Chr()!='O');

while(UART0\_RX\_Chr()!='K');

Lcd\_Data\_Str(2,1," message sent ");

Delay(300);

Enable\_UART0\_Interrupt();

}

/\*----------------------------------------------------------------------------------------------------------

Sending Message For Particular no

----------------------------------------------------------------------------------------------------------\*/

void Message\_Send\_To\_PhNo(unsigned char \*phno,unsigned char \*mmdata)

{

Disable\_UART0\_Interrupt();

Lcd\_Data\_Str(2,1,"message sending");

UART0\_TX\_Str ("AT\r\n");

Delay(gsm\_delay);

UART0\_TX\_Str ("AT+CMGS=");

UART0\_TX\_Chr('"');

UART0\_TX\_Str ("+91");

UART0\_TX\_Str(phno);

UART0\_TX\_Chr('"');

UART0\_TX\_Chr(0x0D);

UART0\_TX\_Chr(0x0A);

while(UART0\_RX\_Chr()!='>');

UART0\_TX\_Str(mmdata);

UART0\_TX\_Chr(0x1a);

while(UART0\_RX\_Chr()!='O');

while(UART0\_RX\_Chr()!='K');

Lcd\_Data\_Str(2,1," message sent ");

Delay(300);

Enable\_UART0\_Interrupt();

}

/\*--------------------------------------------------------------------------------------------------

Message Deleting

--------------------------------------------------------------------------------------------------\*/

void Message\_Deleting(void)

{

Lcd\_Data\_Chr(0,0,0,LCD\_CLEAR);

Lcd\_Data\_Str(2,1,"Mess over load..");

Lcd\_Data\_Str(2,1,"Deleting............");

Disable\_UART0\_Interrupt();

for(j=0X31;j<=0X39;j++)

{

UART0\_TX\_Str("AT+CMGD=");

UART0\_TX\_Chr(j);

UART0\_TX\_Chr(0X0D);

UART0\_TX\_Chr(0X0A);

while(UART0\_RX\_Chr()!='O');

while(UART0\_RX\_Chr()!='K');

if(j==0x39)

{

UART0\_TX\_Str("AT+CMGD=");

UART0\_TX\_Chr(0x31);

UART0\_TX\_Chr(0x30);

UART0\_TX\_Chr(0X0D);

UART0\_TX\_Chr(0X0A);

while(UART0\_RX\_Chr()!='O');

while(UART0\_RX\_Chr()!='K');

Lcd\_Data\_Str(2,1," Deleted ");

Delay(300);

Enable\_UART0\_Interrupt();

}

}

}

/\*--------------------------------------------------------------------------------------------------\*/

/\*--------------------------------------------------------------------------------------------------\*/

void clear\_mess\_buffer(void)

{

for(j=0;j<16;j++) //

{

mess[j]=0x00;

}

}

}