**PIC PROGRAME**

#include<pic.h>

#define PCTX RB4

#define PCRX RB6

#define GSMTX RB5

#define GSMRX RB7

void transmit(char);

void transmiti(char);

void delay(int x);

void data(char);

void stringdisplay(const char \*);

void stringtransmit(const char \*s);

void stringtransmiti(const char \*s);

void command(char);

void delay2(int x);

char msgread();

void gsm\_initial();

char recieve(void);

int flag=0,flag1=0;

void main()

{

char a,b,s,g;

int i,count=0,temp=0;

TRISB=0X00;

PORTB=0X00;

TRISC = 0x80;

TXSTA = 0x24;

RCSTA = 0x90;

TRISD=0X00;

SPBRG = 25;

TRISA=0X07;

ADCON0=0X01;

ADCON1=0X00;

RC5=1;

command (0X01);

command(0X80);

command(0X38);

command(0X06);

command(0X0E);

PCTX = 0;

PCRX = 0;

GSMTX = 1;

GSMRX= 1;

gsm\_initial();

CHS0=0;

CHS1=1;

CHS2=0;

ADGO=1;

while (ADGO==1);

s=ADRESH;

if(s>=100)

{ PCTX = 0;

PCRX = 0;

GSMTX = 1;

GSMRX= 1;

stringdisplay("manual");

delay(10);

while(1)

{

g=msgread();

switch (g)

{

case '1':

{

RB0=1;

RB2=0;

RB1=0;

RB3=0;

RC4=0;

RC5=1;

delay(60);

break;

}

case '2':

{

RB1=0;

RB3=0;

RB0=1;

RB2=1;

RC4=0;

RC5=1;

delay(60);

break;

}

case '3':

{

RB0=1;

RB1=0;

RB2=0;

RB3=0;

RC4=1;

RC5=0;

delay(200);

delay(200);

delay(200);

RC5=1;

flag=1;

PCTX=0;

PCRX=0;

GSMTX = 1;

GSMRX= 1;

transmit(0x0d);

stringtransmit("AT+CMGS=");

transmit('"');

stringtransmit("+919645433341");

transmit('"');

transmit(0x0d);//similar to enter key

delay(10);

stringtransmit("baby is not sleeping");

transmit(0x1a);//message terminate(similar to control z)

transmit(0x1a);

transmit(0x1a);

transmit(0x1a);

delay(20);

break;

}

default :

{

RB0=0;

RB1=0;

RB2=0;

RB3=0;

RC4=0;

RC5=1;

PCTX=0;

PCRX=0;

GSMTX = 1;

GSMRX= 1;

transmit(0x0d);

stringtransmit("AT+CMGS=");

transmit('"');

stringtransmit("+919645433341");

transmit('"');

transmit(0x0d);//similar to enter key

delay(10);

stringtransmit("you entered a wrong command");

transmit(0x1a);//message terminate(similar to controlz)

transmit(0x1a);

transmit(0x1a);

transmit(0x1a);

delay(20);

CHS0=0;

CHS1=0;

CHS2=0;

temp=0;

for(i=1;i<10;i++)

{

ADGO=1;

while (ADGO==1);

a=ADRESH;

temp=temp+a;

}

a=temp/10;

a=5\*a\*100/255;//CALIBRATION FOR TEMP SENSOr

count=count+1;

if(count%2==0)

{

command(0Xc0);

CHS0=1;

CHS1=0;

CHS2=0;

ADGO=1;

while (ADGO==1);

stringdisplay("");

b=ADRESH;

stringdisplay("temp=");

data(a/100+48);

data((a/10)%10+48);

data((a%10)+48);

data(',');

stringdisplay("moi=");

data(b/100+48);

data((b/10)%10+48);

data((b%10)+48);

command (0X01);

delay(20);

}

else

{

command(0X80);

//a=ADRESH;

//a=5\*a\*100/255;//CALIBRATION FOR TEMP SENSOR

CHS0=1;

CHS1=0;

CHS2=0;

ADGO=1;

while (ADGO==1);

b=ADRESH;

stringdisplay("temp=");

data(a/100+48);

data((a/10)%10+48);

data((a%10)+48);

data(',');

stringdisplay("moi=");

data(b/100+48);

data((b/10)%10+48);

data((b%10)+48);

command(0X01);

delay(20);

}

}

}

}

else

{

PCTX = 1;

PCRX = 1;

GSMTX = 0;

GSMRX= 0;

GIE=1;

PEIE=1;

RCIE=1;

delay(10);

stringdisplay("automatic");

delay(10);

while(1)

{

delay(10);

CHS0=0;

CHS1=0;

CHS2=0;

temp=0;

for(i=1;i<10;i++)

{

ADGO=1;

while (ADGO==1);

a=ADRESH;

temp=temp+a;

}

a=temp/10;

a=5\*a\*100/255;//CALIBRATION FOR TEMP SENSOr

count=count+1;

if(count%2==0)

{

command(0Xc0);

CHS0=1;

CHS1=0;

CHS2=0;

ADGO=1;

while (ADGO==1);

stringdisplay("");

b=ADRESH;

stringdisplay("temp=");

data(a/100+48);

data((a/10)%10+48);

data((a%10)+48);

data(',');

stringdisplay("moi=");

data(b/100+48);

data((b/10)%10+48);

data((b%10)+48);

command (0X01);

delay(20);

}

else

{

command(0X80);

//a=ADRESH;

//a=5\*a\*100/255;//CALIBRATION FOR TEMP SENSOr

CHS0=1;

CHS1=0;

CHS2=0;

ADGO=1;

while (ADGO==1);

b=ADRESH;

stringdisplay("temp=");

data(a/100+48);

data((a/10)%10+48);

data((a%10)+48);

data(',');

stringdisplay("moi=");

data(b/100+48);

data((b/10)%10+48);

data((b%10)+48);

command(0X01);

delay(20);

}

if(a>=35 || b>=130 )

{

RC4=1;

PCTX=0;

PCRX=0;

GSMTX = 1;

GSMRX= 1;

delay(40);

flag1=1;

}

else

{

flag1=0;

if(flag==0 && flag1==0)

RC4=0;

transmit(a/100+48);

transmit((a/10)%10+48);

transmit((a%10)+48);

transmit(',');

transmit(b/100+48);

transmit((b/10)%10+48);

transmit((b%10)+48);

transmit(',');

delay(10);

}

}

}

}

void transmit(char data)

{

TXREG = data;

while(TXIF==0);

TXIF=0;

}

void transmiti(char data)

{

TXREG = data;

while(TXIF==0);

TXIF=0;

}

void interrupt recV(void)

{

char d;

GIE=0;PEIE=0;RCIE=0;

if(RCIF==1)

{ //while(1)

{

d = recieve();

//transmit(d);

switch (d)

{

case '1':

{

RB0=1;

RB2=0;

RB1=0;

RB3=0;

RC4=0;

RC5=1;

delay2(60);

break;

}

case '2':

{

RB1=0;

RB3=0;

RB0=1;

RB2=1;

RC4=0;

RC5=1;

delay2(60);

break;

}

case '3':

{

RB0=1;

RB1=0;

RB2=0;

RB3=0;

RC4=1;

delay2(50);

RC5=0;

delay2(200);

delay2(200);

delay2(200);

RC5=1;

flag=1;

PCTX=0;

PCRX=0;

GSMTX = 1;

GSMRX= 1;

transmiti(0x0d);

stringtransmiti("AT+CMGS=");

transmiti('"');

stringtransmiti("+919645433341");

transmiti('"');

transmiti(0x0d);//similar to enter key

delay2(10);

stringtransmiti("baby is not sleeping");

transmiti(0x1a);//message terminate(similar to controlz)

transmiti(0x1a);

transmiti(0x1a);

transmiti(0x1a);

delay2(40);

PCTX=1;

PCRX=1;

GSMTX = 0;

GSMRX= 0;

break;

}

default :

{ RB0=0;

RB1=0;

RB2=0;

RB3=0;

RC4=0;

RC5=0;

PCTX=0;

PCRX=0;

GSMTX = 1;

GSMRX= 1;

transmiti(0x0d);

stringtransmiti("AT+CMGS=");

transmiti('"');

stringtransmiti("+919645433341");

transmiti('"');

transmiti(0x0d);//similar to enter key

delay2(10);

stringtransmiti("you entered a wrong command");

transmiti(0x1a);//message terminate(similar to controlz)

transmiti(0x1a);

transmiti(0x1a);

transmiti(0x1a);

PCTX=1;

PCRX=1;

GSMTX = 0;

GSMRX= 0;

}

}

delay2(70);

}

GIE=1; PEIE=1; RCIE=1;RCIF=0;

}

}

char recieve(void)

{

while(RCIF == 0);

RCIF = 0;

return(RCREG);

}

void data(char data)

{

PORTD=data;

RC0=1;

RC1=0;

RC2=1;

delay(5);

RC2=0;

}

void command(char data)

{

PORTD=data;

RC0=0;

RC1=0;

RC2=1;

delay(5);

RC2=0;

}

void stringdisplay(const char \*s)

{

while(\*s)

{

data (\*s++);

delay(5);

}

}

void stringtransmit(const char \*s)

{

while(\*s)

{

transmit (\*s++);

delay(5);

}

}

void stringtransmiti(const char \*s)

{

while(\*s)

{

transmiti (\*s++);

delay2(5);

}

}

void delay(int x)

{

int i,j;

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

for(j=0;j<1000;j++);

}

void delay2(int x)

{

int i,j;

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

for(j=0;j<1000;j++);

}

char msgread()

{

char r;

while( recieve()!='+');

command(0x01);

stringdisplay("msg rec");

delay(5);

stringtransmit("AT+CMGR=1");

transmit(0x0d);

while(recieve()!='\*');

r=recieve();

delay(5);

stringtransmit("AT+CMGD=1");

transmit(0x0d);

return r;

}

void gsm\_initial()

{

transmit(0x0d);

stringtransmit("AT");//starting

transmit(0x0d);

delay(5);

stringtransmit("AT+CMGF=1");//message format

transmit(0x0d);

delay(5);

stringtransmit("AT+CSAS");//message save settings

transmit(0x0d);

delay(5);

stringtransmit("AT+CMGD=1");//message delete

transmit(0x0d);

}