/\*\*\*\*\*\*\*\*Alpha-Numeric Keypad\*\*\*\*\*\*\*\*\*/

void lcd\_initialize(void);

void lcd\_wcmd(char i);

void lcd\_wdata(char j);

void lcd\_busy\_check(void);

void lcd\_goto\_xy();

void lcd\_write\_text(char txt[40]);

void clear(void);

void uart\_send(char a);

void get\_key(void);

void run\_key\_function(void);

void update\_display(void);

void gsm\_send\_command(char gsm\_command[10]);

void gsm\_send\_sms(void);

char d=0,column=0,temp=0,key=17,char\_index=0,temp1=0,temp2=0,character[4],char\_type=1,prev\_key=0;

char cursor\_position=1,roll=1,roll\_count=0,a,b,temporary;

char message[50],temp4,data\_index;

void KEY\_PRESS() iv IVT\_ADDR\_INT0 ics ICS\_AUTO

{

PORTC=0X00;

TCCR2=0X00;

get\_key(); //Identify the key with previous procedure

lcd\_wcmd(14); //Show cursor

if((key%4)==0)

{

run\_key\_function(); //Execute the function of pressed key

}

else

{

if(key==prev\_key)

{

run\_key\_function(); //Execute the function of pressed key

}

else

{

char\_index=0;

if(roll==0)

{

cursor\_position++;

data\_index++;

run\_key\_function(); //Execute the function of pressed key

}

else

{

run\_key\_function(); //Execute the function of pressed key

}

}

message[data\_index]=character[char\_type];

lcd\_goto\_xy();

lcd\_wdata(message[data\_index]);

lcd\_goto\_xy();

TCCR2=0X0F; //START COUNTER TO INCREMENT CURSOR POSITION

roll=0;

}

if((key==14)||(char\_type==0))

{

roll\_count=28;

}

prev\_key=key;

PORTA|=0X0F; //Enable all the columns

}

void main()

{

PORTB=0X00; DDRB=0X00;

PORTC=0X00; DDRC=0X00;

PORTA=0XFF; DDRA=0XFF;

PORTD=0XFF; DDRD=0X00;

TCCR1B=0X00;

lcd\_initialize();

lcd\_busy\_check();

cursor\_position=68;

lcd\_goto\_xy();

lcd\_write\_text("ABC");

cursor\_position=1;

GICR=0X40; MCUCR=0X03;

OCR2=0XFF; TIMSK|=0X80;

SREG|=0X80;

cursor\_position=1;

lcd\_wcmd(0X0E);

lcd\_goto\_xy();

while(1)

{

}

}

void KEY\_ROLL() iv IVT\_ADDR\_TIMER2\_COMP ics ICS\_AUTO

{

roll\_count++;

if(roll\_count==30)

{

TCCR2=0X00;

char\_index=0;

cursor\_position++;

lcd\_goto\_xy();

data\_index++;

key=21;

roll=1;

roll\_count=0;

}

else

{

TCCR2=0X0F;

}

}

void get\_key(void)

{

for(column=1,temp=1;column<=4;temp\*=2,column++)

{

PORTA&=0XF0; //Disable all columns without disturbing remaining pins

PORTA|=temp; //Enable particular column leaving remaining port pins

switch(PINC&0X1F) //Read rows data and identify the

{

case 0X01:{ //Row 1

key=column;

}break;

case 0X02:{ //Row 2

key=4+column;

}break;

case 0X04:{ //Row 3

key=8+column;

}break;

case 0X08:{ //Row 4

key=12+column;

}break;

case 0X10:{ //Row 5

key=16+column;

}break;

}

do{}while((PINC&0X1F)!=0); //Wait for key release

}

}

void run\_key\_function(void)

{

switch(key)

{

case 1:{ //KEY 1

character[0]=49; //1

character[1]='.';

character[2]='.';

character[3]='.';

}break;

case 2:{ //KEY 2

character[0]=50; //2

character[1]=65+char\_index; //A

character[2]=97+char\_index; //a

character[3]=','; //,

char\_index++;

char\_index%=3;

}break;

case 3:{ //KEY 3

character[0]=51; //3

character[1]=68+char\_index; //D

character[2]=100+char\_index; //d

character[3]='?'; //?

char\_index+=1;

char\_index%=3;

}break;

case 4:{ //KEY 4

cursor\_position--;

data\_index--;

message[data\_index]=0X00;

lcd\_goto\_xy();

lcd\_wdata(message[data\_index]);

lcd\_goto\_xy();

roll=1;

}break;

case 5:{ //KEY 5

character[0]=52; //4

character[1]=71+char\_index; //G

character[2]=103+char\_index; //g

character[3]=';'; //;

char\_index+=1;

char\_index%=3;

}break;

case 6:{ //KEY 6

character[0]=53; //5

character[1]=74+char\_index; //J

character[2]=106+char\_index; //j

character[3]='!'; //!

char\_index+=1;

char\_index%=3;

}break;

case 7:{ //KEY 7

character[0]=54; //6

character[1]=77+char\_index; //M

character[2]=109+char\_index; //m

character[3]='@'; //@

char\_index+=1;

char\_index%=3;

}break;

case 8:{ //KEY 8

char\_type++;

char\_type%=4;

temporary=cursor\_position;

cursor\_position=68;

lcd\_goto\_xy();

if(char\_type==0)

{

lcd\_write\_text("123");

}

else if(char\_type==1)

{

lcd\_write\_text("ABC");

}

else if(char\_type==2)

{

lcd\_write\_text("abc");

}

else if(char\_type==3)

{

lcd\_write\_text(".,?");

}

cursor\_position=temporary;

lcd\_goto\_xy();

}break;

case 9:{ //KEY 9

character[0]=55; //7

character[1]=80+char\_index; //P

character[2]=112+char\_index; //p

character[3]='&'; //&

char\_index+=1;

char\_index%=4;

}break;

case 10:{ //KEY 10

character[0]=56; //8

character[1]=84+char\_index; //T

character[2]=116+char\_index; //t

character[3]='('; //(

char\_index+=1;

char\_index%=3;

}break;

case 11:{ //KEY 11

character[0]=57; //9

character[1]=87+char\_index; //W

character[2]=119+char\_index; //w

character[3]=')'; //)

char\_index+=1;

char\_index%=4;

}break;

case 12:{ //KEY 12

lcd\_wcmd(1);

cursor\_position=68;

lcd\_goto\_xy();

data\_index=0;

char\_type=1;

lcd\_write\_text("ABC");

cursor\_position=1;

lcd\_goto\_xy();

}break;

case 13:{ //KEY 13

character[0]=42; //\*

character[1]=42; //\*

character[2]=42; //\*

character[3]=42; //\*

}break;

case 14:{ //KEY 14

character[0]=48; //0

character[1]=32; //SPACE

character[2]=32; //SPACE

character[3]=32; //SPACE

}break;

case 15:{ //KEY 15

character[0]=35; //#

character[1]=35; //#

character[2]=35; //#

character[3]=35; //#

}break;

case 16:{ //KEY 16

message[data\_index]=13;

lcd\_wcmd(1);

cursor\_position=1;

lcd\_goto\_xy();

lcd\_write\_text("Message:");

cursor\_position=21;

lcd\_goto\_xy();

for(data\_index=0;message[data\_index]!=13;data\_index++)

{

lcd\_wdata(message[data\_index]);

cursor\_position++;

lcd\_goto\_xy();

}

lcd\_wcmd(0X0C);

}break;

}

}

void lcd\_wcmd(char i)

{

lcd\_busy\_check();

PORTA&=0XEF;

PORTA&=0XDF;

PORTB=0;

DDRB=0XFF;

PORTA|=0X40;

Delay\_us(10);

PORTB=i;

PORTA&=0XBF;

}

void lcd\_wdata(char j)

{

lcd\_busy\_check();

PORTA|=0X10;

PORTA&=0XDF;

PORTA|=0X40;

Delay\_us(10);

DDRB=0XFF;

PORTB=j;

PORTA&=0XBF;

Delay\_us(30);

}

void lcd\_goto\_xy()

{

lcd\_busy\_check();

if(cursor\_position<20)

{

b=1;

a=cursor\_position%20;

}

else if(cursor\_position==20)

{

b=1;

a=20;

}

else if(cursor\_position<40)

{

b=2;

a=cursor\_position%20;

}

else if(cursor\_position==40)

{

b=2;

a=20;

}

else if(cursor\_position<60)

{

b=3;

a=cursor\_position%20;

}

else if(cursor\_position==60)

{

b=3;

a=20;

}

else if(cursor\_position<80)

{

b=4;

a=cursor\_position%20;

}

else if(cursor\_position==80)

{

b=4;

a=20;

}

switch(b)

{

case 1:{

a=a+127; //LINE 1 FIRST CHARACTER IS AT 128 FROM DATASHEET

// d=a;

lcd\_wcmd(a);

}break;

case 2:{

a=a+191;

// d=a;

lcd\_wcmd(a);

}break;

case 3:{

a=a+147;

// d=a;

lcd\_wcmd(a);

}break;

case 4:{

a=a+211;

// d=a;

lcd\_wcmd(a);

}break;

}

}

void uart\_send(char a)

{

do{}while((UCSRA&0X20)==0);

UCSRA|=0X40;

UDR=a;

do{}while((UCSRA&0X40)==0);

}

void lcd\_initialize(void)

{

lcd\_wcmd(0X38); //FUNCTION SET

lcd\_wcmd(0X0E); //DISPLAY CURSOR

lcd\_wcmd(0X01); //CLEAR

lcd\_wcmd(0X06); //SHIFT CURSOR FOR EVERY DDRAM UPDATE

}

void lcd\_write\_text(char txt[40])

{

char p;

for(p=0;txt[p]!='\0';p++)

{

lcd\_wdata(txt[p]);

}

}

void clear(void)

{

PORTA&=0X8F;

/\*PORTA4\_BIT=0;

PORTA5\_BIT=0;

PORTA6\_BIT=0;\*/

}

void lcd\_busy\_check(void)

{

Delay\_ms(5);

//RS=0;

/\*PORTA&=0XEF;

//RW=1;

PORTA|=0X20;

//E=0;

PORTA&=0XBF;

PORTB=0X00;

DDRB=0X00;

\_delay\_us(100);

//E=1;

PORTA|=0X40;

\_delay\_us(100);

//E=0;

PORTA&=0XBF;

do

{}while(PINB&0X80!=0);\*/

}