LAMPIRAN

Listing Program :

```
#include <mega8535.h>
#include <delay.h>

unsigned char keypad='',scrn=0;

bit
send_at,send_atcmgf,send_atcmgfask,send_atcnmi,send_atcnmiask,s
end_atandw,rst_modem,send_atcmgs;

unsigned char provider,nominal,nominal_range;
unsigned char number[16]={' ',' ',' ',' ',' ',' ',' ',' ','
',' ',' ',' ',' ',' ',' '};
unsigned char eeprom pin[5]={' ',' ',' ',' ',' '};
unsigned char eeprom server[16]={' ',' ',' ',' ',' ',' ',' ','
',' ',' ',' ',' ',' ',' ',' ',' '};
unsigned char eeprom counter=0;
unsigned char number_idx,pin_idx,server_idx;
bit confirm;

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 104
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
```

```
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
   {
   rx_buffer[rx_wr_index]=data;
   if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
   if (++rx_counter == RX_BUFFER_SIZE)
      {
      rx_counter=0;
      rx_buffer_overflow=1;
      };
   };
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
//  Func7=In  Func6=In  Func5=In  Func4=In  Func3=In  Func2=In
Func1=In Func0=In
//  State7=T  State6=T  State5=T  State4=T  State3=T  State2=T
State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
//  Func7=In  Func6=In  Func5=In  Func4=In  Func3=In  Func2=In
Func1=In Func0=In
//  State7=T  State6=T  State5=T  State4=T  State3=T  State2=T
State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=Out  Func6=Out  Func5=Out  Func4=Out  Func3=In  Func2=In
Func1=In Func0=In
//  State7=0  State6=0  State5=0  State4=0  State3=P  State2=P
State1=P State0=P
PORTC=0x0F;
DDRC=0xF0;

// Port D initialization
//  Func7=In  Func6=In  Func5=In  Func4=In  Func3=In  Func2=Out
Func1=In Func0=In
//  State7=T  State6=T  State5=T  State4=T  State3=T  State2=1
State1=T State0=T
PORTD=0x04;
DDRD=0x04;

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

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

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

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

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

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

// LCD module initialization
lcd_init(16);
```

```
// Global enable interrupts
#asm("sei")

//start: initializing parameter
lcd_clear();
lcd_putsf("System StartUp");
delay_ms(1000);

lcd_clear();
lcd_putsf("Initialize Param");
lcd_gotoxy(0,1);
lcd_putsf("Please Wait     ");
delay_ms(30000);
//start:set modem sms receiving mode
//printf("AT+CNMI=1,2,0,0,0");
putchar('A');
delay_ms(5);
putchar('T');
delay_ms(5);
putchar('+');
delay_ms(5);
putchar('C');
delay_ms(5);
putchar('N');
delay_ms(5);
putchar('M');
delay_ms(5);
putchar('I');
delay_ms(5);
putchar('=');
delay_ms(5);
putchar('1');
delay_ms(5);
putchar(',');
delay_ms(5);
putchar('2');
delay_ms(5);
putchar(',');
delay_ms(5);
putchar('0');
delay_ms(5);
putchar(',');
delay_ms(5);
putchar('0');
delay_ms(5);
putchar(',');
delay_ms(5);
putchar('0');
delay_ms(5);
putchar(13);
delay_ms(5);
```

```
//end:set modem sms receiving mode
//end: initializing parameter

while (1)
      {
       // Place your code here

      //start: processing data from keypad
      if((PINC.0==0)||(PINC.1==0)||(PINC.2==0)||(PINC.3==0))
      {
       PORTC.7=0; PORTC.6=1; PORTC.5=1; PORTC.4=1;
       delay_ms(10);
       if(PINC.0==0)
        keypad='R';
       if(PINC.1==0)
        keypad='3';
       if(PINC.2==0)
        keypad='2';
       if(PINC.3==0)
        keypad='1';
       PORTC.7=1; PORTC.6=0; PORTC.5=1; PORTC.4=1;
       delay_ms(10);
       if(PINC.0==0)
        keypad='M';
       if(PINC.1==0)
        keypad='6';
       if(PINC.2==0)
        keypad='5';
       if(PINC.3==0)
        keypad='4';
       PORTC.7=1; PORTC.6=1; PORTC.5=0; PORTC.4=1;
       delay_ms(10);
       if(PINC.0==0)
        keypad='U';
       if(PINC.1==0)
        keypad='9';
       if(PINC.2==0)
        keypad='8';
       if(PINC.3==0)
        keypad='7';
       PORTC.7=1; PORTC.6=1; PORTC.5=1; PORTC.4=0;
       delay_ms(10);
       if(PINC.0==0)
        keypad='D';
       if(PINC.1==0)
        keypad='E';
       if(PINC.2==0)
        keypad='0';
       if(PINC.3==0)
        keypad='C';
       PORTC.7=0; PORTC.6=0; PORTC.5=0; PORTC.4=0;
```

```
while((PINC.0==0)||(PINC.1==0)||(PINC.2==0)||(PINC.3==0));
       delay_ms(50);
      }
      //end: processing data from keypad

      //start: display to LCD
      if(scrn==0)       //provider screen
      {
       lcd_clear();
       lcd_putsf("Entry Provider :");
       lcd_gotoxy(0,1);
       if(provider==0)
        lcd_putsf("None            ");
       if(provider==1)
        lcd_putsf("XL              ");
       if(provider==2)
        lcd_putsf("IM3             ");
       if(provider==3)
        lcd_putsf("Mentari         ");
       if(provider==4)
        lcd_putsf("StarOne         ");
       if(provider==5)
        lcd_putsf("Simpati         ");
       if(provider==6)
        lcd_putsf("As              ");
       if(provider==7)
        lcd_putsf("Fleksi          ");
       if(provider==8)
        lcd_putsf("Smart           ");
       if(provider==9)
        lcd_putsf("Three           ");
       if(provider==10)
        lcd_putsf("Axis            ");
       if(provider==11)
        lcd_putsf("Fren            ");
       if(provider==12)
        lcd_putsf("Esia            ");
      }
      if(scrn==1)       //manual screen
      {
       lcd_clear();
       lcd_putsf("Manual Scrn :");
       lcd_gotoxy(0,1);
       if(send_at==1)
       {
        lcd_putsf("Send AT        ");
        delay_ms(500);
        send_at=0;
       }
       if(send_atcmgf==1)
       {
```

```
  lcd_putsf("Send AT+CMGF    ");
  delay_ms(500);
  send_atcmgf=0;
 }
 if(send_atcmgfask==1)
 {
  lcd_putsf("Send AT+CMGF?   ");
  delay_ms(500);
  send_atcmgfask=0;
 }
 if(send_atcnmi==1)
 {
  lcd_putsf("Send AT+CNMI    ");
  delay_ms(500);
  send_atcnmi=0;
 }
 if(send_atcnmiask==1)
 {
  lcd_putsf("Send AT+CNMI?   ");
  delay_ms(500);
  send_atcnmiask=0;
 }
 if(send_atandw==1)
 {
  lcd_putsf("Send AT&W       ");
  delay_ms(500);
  send_atandw=0;
 }
 if(rst_modem==1)
 {
  lcd_putsf("Reset Modem     ");
  delay_ms(500);
  rst_modem=0;
 }
 if(send_atcmgs==1)
 {
  lcd_putsf("Send AT+CMGS    ");
  delay_ms(500);
  send_atcmgs=0;
 }
}
if(scrn==2)       //nominal screen
{
 lcd_clear();
 lcd_putsf("Entry Nominal :");
 lcd_gotoxy(0,1);

 if((provider>0)&&(nominal==0))
  lcd_putsf("None            ");

 if(((provider>=1)&&(provider<=4))&&(nominal==1))
  lcd_putsf("IDR 5000        ");
```

```
       if(((provider>=1)&&(provider<=4))&&(nominal==2))
        lcd_putsf("IDR 10000       ");
       if(((provider>=1)&&(provider<=4))&&(nominal==3))
        lcd_putsf("IDR 25000       ");
       if(((provider>=1)&&(provider<=4))&&(nominal==4))
        lcd_putsf("IDR 50000       ");
       if(((provider>=1)&&(provider<=4))&&(nominal==5))
        lcd_putsf("IDR 100000      ");


if((((provider>=5)&&(provider<=8))||(provider==10)||(provider==
11))&&(nominal==1))
        lcd_putsf("IDR 5000        ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==2))
        lcd_putsf("IDR 10000       ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==3))
        lcd_putsf("IDR 20000       ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==4))
        lcd_putsf("IDR 50000       ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==5))
        lcd_putsf("IDR 100000      ");

       if(((provider==9)||(provider==12))&&(nominal==1))
        lcd_putsf("IDR 5000        ");
       if(((provider==9)||(provider==12))&&(nominal==2))
        lcd_putsf("IDR 10000       ");
       if(((provider==9)||(provider==12))&&(nominal==3))
        lcd_putsf("IDR 20000       ");
       if(((provider==9)||(provider==12))&&(nominal==4))
        lcd_putsf("IDR 25000       ");
       if(((provider==9)||(provider==12))&&(nominal==5))
        lcd_putsf("IDR 50000       ");
       if(((provider==9)||(provider==12))&&(nominal==6))
        lcd_putsf("IDR 100000      ");
      }
      if(scrn==3)       //number screen
      {
       lcd_clear();
       lcd_putsf("Entry Number :  ");
       lcd_gotoxy(0,1);
       lcd_putchar(number[0]);
       lcd_putchar(number[1]);
       lcd_putchar(number[2]);
       lcd_putchar(number[3]);
```

```
 lcd_putchar(number[4]);
 lcd_putchar(number[5]);
 lcd_putchar(number[6]);
 lcd_putchar(number[7]);
 lcd_putchar(number[8]);
 lcd_putchar(number[9]);
 lcd_putchar(number[10]);
 lcd_putchar(number[11]);
 lcd_putchar(number[12]);
 lcd_putchar(number[13]);
 lcd_putchar(number[14]);
 lcd_putchar(number[15]);
}
if(scrn==4)       //confirm screen
{
 lcd_clear();
 lcd_gotoxy(0,0);
 if(provider==0)
  lcd_putsf("None   ");
 if(provider==1)
  lcd_putsf("XL     ");
 if(provider==2)
  lcd_putsf("IM3    ");
 if(provider==3)
  lcd_putsf("Mentari");
 if(provider==4)
  lcd_putsf("StarOne");
 if(provider==5)
  lcd_putsf("Simpati");
 if(provider==6)
  lcd_putsf("As     ");
 if(provider==7)
  lcd_putsf("Fleksi ");
 if(provider==8)
  lcd_putsf("Smart  ");
 if(provider==9)
  lcd_putsf("Three  ");
 if(provider==10)
  lcd_putsf("Axis   ");
 if(provider==11)
  lcd_putsf("Fren   ");
 if(provider==12)
  lcd_putsf("Esia   ");

 lcd_gotoxy(8,0);
 if(((provider>=1)&&(provider<=4))&&(nominal==1))
  lcd_putsf("IDR 5K  ");
 if(((provider>=1)&&(provider<=4))&&(nominal==2))
  lcd_putsf("IDR 10K ");
 if(((provider>=1)&&(provider<=4))&&(nominal==3))
  lcd_putsf("IDR 25K ");
 if(((provider>=1)&&(provider<=4))&&(nominal==4))
```

```
        lcd_putsf("IDR 50K ");
       if(((provider>=1)&&(provider<=4))&&(nominal==5))
        lcd_putsf("IDR 100K");


if((((provider>=5)&&(provider<=8))||(provider==10)||(provider==
11))&&(nominal==1))
        lcd_putsf("IDR 5K  ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==2))
        lcd_putsf("IDR 10K ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==3))
        lcd_putsf("IDR 20K ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==4))
        lcd_putsf("IDR 50K ");

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==5))
        lcd_putsf("IDR 100K");

       if(((provider==9)||(provider==12))&&(nominal==1))
        lcd_putsf("IDR 5K  ");
       if(((provider==9)||(provider==12))&&(nominal==2))
        lcd_putsf("IDR 10K ");
       if(((provider==9)||(provider==12))&&(nominal==3))
        lcd_putsf("IDR 20K ");
       if(((provider==9)||(provider==12))&&(nominal==4))
        lcd_putsf("IDR 25K ");
       if(((provider==9)||(provider==12))&&(nominal==5))
        lcd_putsf("IDR 50K ");
       if(((provider==9)||(provider==12))&&(nominal==6))
        lcd_putsf("IDR 100K");

       lcd_gotoxy(0,1);
       lcd_putchar(number[0]);
       lcd_putchar(number[1]);
       lcd_putchar(number[2]);
       lcd_putchar(number[3]);
       lcd_putchar(number[4]);
       lcd_putchar(number[5]);
       lcd_putchar(number[6]);
       lcd_putchar(number[7]);
       lcd_putchar(number[8]);
       lcd_putchar(number[9]);
       lcd_putchar(number[10]);
       lcd_putchar(number[11]);
       lcd_putchar(number[12]);
       lcd_putchar(number[13]);
       lcd_putchar(number[14]);
       lcd_putchar(number[15]);
      }
      if(scrn==5)       //data status screen
      {
       lcd_clear();
       lcd_putsf("Please Wait");
       lcd_gotoxy(0,1);
       if((confirm==0)||(number[10]=='        ')||(server[10]=='
')||(pin[3]==' '))
        lcd_putsf("Cancelling Data");
       if((confirm==1)&&(number[10]!='        ')&&(server[10]!='
')&&(pin[3]!=' '))
       {
        lcd_putsf("Processing Data");
        counter++;

        //start:sending data to the modem
        //printf("AT+CMGS=");
        //putchar('"');
        putchar('A');
        delay_ms(5);
        putchar('T');
        delay_ms(5);
        putchar('+');
        delay_ms(5);
        putchar('C');
        delay_ms(5);
        putchar('M');
        delay_ms(5);
        putchar('G');
        delay_ms(5);
        putchar('S');
        delay_ms(5);
        putchar('=');
        delay_ms(5);

        putchar('"');
        delay_ms(5);

        for(server_idx=0;server[server_idx]!=' ';server_idx++)
        {
         putchar(server[server_idx]);
         delay_ms(5);
        }
        putchar('"');
        delay_ms(5);

        putchar(13);
        delay_ms(5);
```

```
if(provider==1)
{
 //printf("xr");
 putchar('x');
 delay_ms(5);
 putchar('r');
 delay_ms(5);
}
if(provider==2)
{
 //putchar('i');
 putchar('i');
 delay_ms(5);
}
if(provider==3)
{
 //putchar('m');
 putchar('m');
 delay_ms(5);
}
if(provider==4)
{
 //printf("st");
 putchar('s');
 delay_ms(5);
 putchar('t');
 delay_ms(5);
}
if(provider==5)
{
 //printf("se");
 putchar('s');
 delay_ms(5);
 putchar('e');
 delay_ms(5);
}
if(provider==6)
{
 //printf("ase");
 putchar('a');
 delay_ms(5);
 putchar('s');
 delay_ms(5);
 putchar('e');
 delay_ms(5);
}
if(provider==7)
{
 //putchar('f');
 putchar('f');
 delay_ms(5);
}
```

```
if(provider==8)
{
 //printf("sm");
 putchar('s');
 delay_ms(5);
 putchar('m');
 delay_ms(5);
}
if(provider==9)
{
 //putchar('t');
 putchar('t');
 delay_ms(5);
}
if(provider==10)
{
 //printf("ax");
 putchar('a');
 delay_ms(5);
 putchar('x');
 delay_ms(5);
}
if(provider==11)
{
 //printf("fr");
 putchar('f');
 delay_ms(5);
 putchar('r');
 delay_ms(5);
}
if(provider==12)
{
 //putchar('e');
 putchar('e');
 delay_ms(5);
}

if(((provider>=1)&&(provider<=4))&&(nominal==1))
{
 //putchar('5');
 putchar('5');
 delay_ms(5);
}
if(((provider>=1)&&(provider<=4))&&(nominal==2))
{
 //printf("10");
 putchar('1');
 delay_ms(5);
 putchar('0');
 delay_ms(5);
}
if(((provider>=1)&&(provider<=4))&&(nominal==3))
```

```
        {
         //printf("25");
         putchar('2');
         delay_ms(5);
         putchar('5');
         delay_ms(5);
        }
        if(((provider>=1)&&(provider<=4))&&(nominal==4))
        {
         //printf("50");
         putchar('5');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }
        if(((provider>=1)&&(provider<=4))&&(nominal==5))
        {
         //printf("100");
         putchar('1');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }


if((((provider>=5)&&(provider<=8))||(provider==10)||(provider==
11))&&(nominal==1))
        {
         //putchar('5');
         putchar('5');
         delay_ms(5);
        }

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==2))
        {
         //printf("10");
         putchar('1');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==3))
        {
         //printf("20");
         putchar('2');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==4))
        {
         //printf("50");
         putchar('5');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }

if(((provider>=5)&&(provider<=8)||(provider==10)||(provider==11
))&&(nominal==5))
        {
         //printf("100");
         putchar('1');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }

        if(((provider==9)||(provider==12))&&(nominal==1))
        {
         //putchar('5');
         putchar('5');
         delay_ms(5);
        }
        if(((provider==9)||(provider==12))&&(nominal==2))
        {
         //printf("10");
         putchar('1');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }
        if(((provider==9)||(provider==12))&&(nominal==3))
        {
         //printf("20");
         putchar('2');
         delay_ms(5);
         putchar('0');
         delay_ms(5);
        }
        if(((provider==9)||(provider==12))&&(nominal==4))
        {
         //printf("25");
         putchar('2');
         delay_ms(5);
```

```
  putchar('5');
  delay_ms(5);
 }
 if(((provider==9)||(provider==12))&&(nominal==5))
 {
  //printf("50");
  putchar('5');
  delay_ms(5);
  putchar('0');
  delay_ms(5);
 }
 if(((provider==9)||(provider==12))&&(nominal==6))
 {
  //printf("100");
  putchar('1');
  delay_ms(5);
  putchar('0');
  delay_ms(5);
  putchar('0');
  delay_ms(5);
 }

 putchar('.');
 delay_ms(5);

 for(number_idx=0;number[number_idx]!=' ';number_idx++)
 {
  putchar(number[number_idx]);
  delay_ms(5);
 }

 putchar('.');
 delay_ms(5);

 for(pin_idx=0;pin[pin_idx]!=' ';pin_idx++)
 {
  putchar(pin[pin_idx]);
  delay_ms(5);
 }

 putchar(0x1A);
 delay_ms(5);
 putchar(0x08);
 delay_ms(5);
 //end: sending data to the modem

}

//start: reset the register
provider=0;
nominal=0;
for(number_idx=0;number_idx<16;number_idx++)
```

```
  number[number_idx]=' ';
 number_idx=0;
 server_idx=0;
 pin_idx=0;
 confirm=0;
 //end: reset the register

 delay_ms(1000);
 scrn=0;
}
if(scrn==6)       //setting pin screen
{
 lcd_clear();
 lcd_putsf("Entry Pin :  ");
 lcd_gotoxy(0,1);
 lcd_putchar(pin[0]);
 lcd_putchar(pin[1]);
 lcd_putchar(pin[2]);
 lcd_putchar(pin[3]);
}
if(scrn==7)       //setting server number screen
{
 lcd_clear();
 lcd_putsf("Server Number : ");
 lcd_gotoxy(0,1);
 lcd_putchar(server[0]);
 lcd_putchar(server[1]);
 lcd_putchar(server[2]);
 lcd_putchar(server[3]);
 lcd_putchar(server[4]);
 lcd_putchar(server[5]);
 lcd_putchar(server[6]);
 lcd_putchar(server[7]);
 lcd_putchar(server[8]);
 lcd_putchar(server[9]);
 lcd_putchar(server[10]);
 lcd_putchar(server[11]);
 lcd_putchar(server[12]);
 lcd_putchar(server[13]);
 lcd_putchar(server[14]);
 lcd_putchar(server[15]);
}
if(scrn==8)       //transaction counter screen
{
 lcd_clear();
 lcd_putsf("Sending Cntr :");
 lcd_gotoxy(0,1);
 lcd_putchar(0x30+(counter/100));
 lcd_putchar(0x30+((counter%100)/10));
 lcd_putchar(0x30+((counter%100)%10));
}
//end: display to LCD
```

```
//start: accept data from keypad
if(keypad!='')
{
 //lcd_putchar(keypad);
 if(scrn==0)      //provider screen
 {
  if((keypad=='U')&&(provider<12))
   provider++;
  if((keypad=='D')&&(provider>0))
   provider--;
  if((keypad=='E')&&(provider>0))
   scrn=2;
  if(keypad=='M')
   scrn=6;
  if(keypad=='C')
   scrn=8;
  if(keypad=='R')
   scrn=1;
  keypad='';
 }
 if(scrn==1)      //manual screen
 {
  if(keypad=='1')
  {
   send_at=1;
   //start:testing modem's ATcommand
   //printf("AT");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:testing modem's ATcommand
  }
  if(keypad=='2')
  {
   send_atcmgf=1;
   //start:set modem to text mode
   //printf("AT+CMGF=1");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('+');
   delay_ms(5);
   putchar('C');
   delay_ms(5);
   putchar('M');
   delay_ms(5);
   putchar('G');
   delay_ms(5);
   putchar('F');
   delay_ms(5);
   putchar('=');
   delay_ms(5);
   putchar('1');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:set modem to text mode
  }
  if(keypad=='3')
  {
   send_atcmgfask=1;
   //start:ask for modem text mode
   //printf("AT+CMGF?");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('+');
   delay_ms(5);
   putchar('C');
   delay_ms(5);
   putchar('M');
   delay_ms(5);
   putchar('G');
   delay_ms(5);
   putchar('F');
   delay_ms(5);
   putchar('?');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:ask for modem text mode
  }
  if(keypad=='4')
  {
   send_atcnmi=1;
   //start:set modem sms receiving mode
   //printf("AT+CNMI=1,2,0,0,0");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('+');
   delay_ms(5);
   putchar('C');
   delay_ms(5);
   putchar('N');
   delay_ms(5);
   putchar('M');
   delay_ms(5);
   putchar('I');
   delay_ms(5);
   putchar('=');
   delay_ms(5);
   putchar('1');
   delay_ms(5);
   putchar(',');
   delay_ms(5);
   putchar('2');
   delay_ms(5);
   putchar(',');
   delay_ms(5);
   putchar('0');
   delay_ms(5);
   putchar(',');
   delay_ms(5);
   putchar('0');
   delay_ms(5);
   putchar(',');
   delay_ms(5);
   putchar('0');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:set modem sms receiving mode
  }
  if(keypad=='5')
  {
   send_atcnmiask=1;
   //start:ask for modem sms receiving mode
   //printf("AT+CNMI?");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('+');
   delay_ms(5);
   putchar('C');
   delay_ms(5);
   putchar('N');
   delay_ms(5);
   putchar('M');
   delay_ms(5);
   putchar('I');
   delay_ms(5);
   putchar('?');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:ask for modem sms receiving mode
  }
  if(keypad=='6')
  {
   send_atandw=1;
   //start:save modem setting's
   //printf("AT&W");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('&');
   delay_ms(5);
   putchar('W');
   delay_ms(5);
   putchar(13);
   delay_ms(5);
   //end:save modem setting's
  }
  if(keypad=='7')
  {
   rst_modem=1;
   //start: reset modem
   PORTD.2=0;
   delay_ms(500);
   PORTD.2=1;
   //end: reset modem
  }
  if(keypad=='8')
  {
   send_atcmgs=1;
   //start: test sending message to modem
   //printf("AT+CMGS=");
   putchar('A');
   delay_ms(5);
   putchar('T');
   delay_ms(5);
   putchar('+');
   delay_ms(5);
   putchar('C');
   delay_ms(5);
   putchar('M');
   delay_ms(5);
   putchar('G');
   delay_ms(5);
   putchar('S');
   delay_ms(5);
   putchar('=');
   delay_ms(5);

   putchar('"');
   delay_ms(5);
```

```
  for(server_idx=0;server[server_idx]!=' ';server_idx++)
  {
   putchar(server[server_idx]);
   delay_ms(5);
  }

  putchar('"');
  delay_ms(5);

  putchar(13);
  delay_ms(5);

  //printf("Test Sending Message To Modem");
  putchar('T');
  delay_ms(5);
  putchar('E');
  delay_ms(5);
  putchar('S');
  delay_ms(5);
  putchar('T');
  delay_ms(5);

  putchar(0x1A);
  delay_ms(5);
  putchar(0x08);
  delay_ms(5);
  //end: test sending message to modem
 }

 if(keypad=='R')
 {
  server_idx=0;
  scrn=0;
 }

 keypad='';
}
if(scrn==2)    //nominal screen
{
 if((provider==9)||(provider==12))
  nominal_range=6;
 else
  nominal_range=5;

 if((keypad=='U')&&(nominal<nominal_range))
  nominal++;
 if((keypad=='D')&&(nominal>0))
  nominal--;
 if((keypad=='E')&&(nominal>0))
  scrn=3;
 keypad='';
}
if(scrn==3)    //number screen
{
 if((keypad>='0')&&(keypad<='9'))
 {
  number[number_idx]=keypad;
  if(number_idx<14)
   number_idx++;
  else
   number_idx=0;
 }
 if(keypad=='R')
 {
  for(number_idx=0;number_idx<16;number_idx++)
   number[number_idx]=' ';
  number_idx=0;
 }
 if((keypad=='E')&&(number[10]!=' '))
  scrn=4;

 keypad='';
}
if(scrn==4)      //confirm screen
{
 if(keypad=='E')
 {
  confirm=1;
  scrn=5;
 }
 if(keypad=='C')
 {
  confirm=0;
  scrn=5;
 }

 keypad='';
}
if(scrn==6)      //setting pin screen
{
 if((keypad>='0')&&(keypad<='9'))
 {
  pin[pin_idx]=keypad;
  if(pin_idx<3)
   pin_idx++;
  else
   pin_idx=0;
 }
 if(keypad=='R')
 {
  for(pin_idx=0;pin_idx<16;pin_idx++)
   pin[pin_idx]=' ';
```

```
         pin_idx=0;
        }
        if((keypad=='E')&&(pin[3]!=' '))
        {
         pin_idx=0;
         scrn=7;
        }
        keypad='';
       }
       if(scrn==7)      //setting server number screen
       {
        if((keypad>='0')&&(keypad<='9'))
        {
         server[server_idx]=keypad;
         if(server_idx<14)
          server_idx++;
         else
          server_idx=0;
        }
        if(keypad=='R')
        {
         for(server_idx=0;server_idx<16;server_idx++)
          server[server_idx]=' ';
         server_idx=0;
        }
        if((keypad=='E')&&(server[10]!=' '))
        {
         server_idx=0;
         scrn=0;
        }
        keypad='';
       }
       if(scrn==8)      //transaction counter screen
       {
        if(keypad=='R')
         counter=0;
        if(keypad=='C')
         scrn=0;
        keypad='';
       }
      }
      //end: accept data from keypad

      delay_ms(100);

      };
}
```