

LAMPIRAN

/******

This program was produced by the
CodeWizardAVR V2.03.4 Standard
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :

Version :

Date : 9/16/2013

Author :

Company :

Comments:

Chip type : ATmega16
Program type : Application
Clock frequency : 11.059200 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

*****/

```
#include <mega16.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 64
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>
#include <delay.h>

```

```

// Declare your global variables here
char buffer[10]; //definisi buffer teks
int indexbuf=0;

void kirim_pesan (void)
{
printf("AT+CMGS=");    // perintah kirim sms
putchar("");
printf("085268011133"); // no HP tujuan
putchar("");
putchar(0x0D);
while(getchar()!=0x3E){}; // tunggu sampai muncul karakter ">"
delay_ms(5);
}

void main(void)
{
// Declare your local variables here
unsigned int c;
int a;

// 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=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D 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
PORTD=0x00;
DDRD=0x00;

// 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;

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

```



```
printf("ATE0"); // perintah agar tidak ada echo
putchar(0x0D); // enter
while(getchar() != 'K'){}; // tunggu sampai menerima K / OK
while(getchar() != 0x0A){}; // tunggu sampai newline / garis baru
```

```
printf("AT+CMGF=1"); //mode text
putchar(0x0D);
while(getchar() != 'K'){};
while(getchar() != 0x0A){};
```

```
printf("AT+CMGD=0,4"); // hapus semua sms
putchar(0x0D);
while(getchar() != 'K'){};
while(getchar() != 0x0A){};
```

```
printf("AT+CNMI=2,2"); // mode langsung baca sms
putchar(0x0D);
while(getchar() != 'K'){};
while(getchar() != 0x0A){};
```

```
PORTC=0XFF;
delay_ms(1000);
PORTC=0X00;
while (1)
{
// Place your code here
c = getchar();
if ( c == " ) // jika ga ada data masuk
{
}
else
```

```

    {
        if( c == '$') indexbuf=0; //jika triger $, maka mulai parsing (index array
buffer = 0 )
        buffer[indexbuf] = c; //masukkan teks sms ke buffer
        indexbuf++; //index array ditambah
    }

// Jika yg diterima sms "$ON 1", maka Lampu 1 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='1')
    {
        PORTC.0=1;
        for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
        kirim_pesan ();
        printf("Lampu 1 sudah dinyalakan");
        putchar(0x1A);          // CTRL + Z
        while(getchar()!='K'){};
    }

// Jika yg diterima sms "$ON 2", maka Lampu 2 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='2')
    {
        PORTC.1=1;
        for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
        kirim_pesan ();
        printf("Lampu 2 sudah dinyalakan");
        putchar(0x1A);          // CTRL + Z
        while(getchar()!='K'){};
    }

```

```

// Jika yg diterima sms "$ON 3", maka Lampu 3 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='3')
{
PORTC.2=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 3 sudah dinyalakan");
 putchar(0x1A);      // CTRL + Z
 while(getchar()!='K'){};
 }

```

```

// Jika yg diterima sms "$ON 4", maka Lampu 4 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='4')
{
PORTC.3=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 4 sudah dinyalakan");
 putchar(0x1A);      // CTRL + Z
 while(getchar()!='K'){};
 }

```

```

// Jika yg diterima sms "$ON 5", maka Lampu 5 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='5')
{
PORTC.4=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();

```

```

printf("Lampu 5 sudah dinyalakan");
putchar(0x1A);      // CTRL + Z
while(getchar()!='K'){};
}

```

```

// Jika yg diterima sms "$ON 6", maka Lampu 6 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='6')
{
PORTC.5=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
printf("Lampu 6 sudah dinyalakan");
putchar(0x1A);      // CTRL + Z
while(getchar()!='K'){};
}

```

```

// Jika yg diterima sms "$ON 7", maka Lampu 7 akan menyala
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='7')
{
PORTC.6=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
printf("Lampu 7 sudah dinyalakan");
putchar(0x1A);      // CTRL + Z
while(getchar()!='K'){};
}

```

```

// Jika yg diterima sms "$ON 8", maka Lampu 8 akan menyala

```

```

if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='8')
{
PORTC.7=1;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 8 sudah dinyalakan");
 putchar(0x1A);      // CTRL + Z
 while(getchar()!='K'){};
}

```

// Jika yg diterima sms "\$ON ALL", maka Semua Lampu akan menyala

```

if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='N' && buffer[3]==' '
&& buffer[4]=='A' && buffer[5]=='L' && buffer[6]=='L')
{
PORTC=0XFF;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Semua Lampu sudah dinyalakan");
 putchar(0x1A);      // CTRL + Z
 while(getchar()!='K'){};
}

```

// Jika yg diterima sms "\$OFF 1", maka lampu 1 akan padam

```

if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='1')
{
PORTC.0=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 1 sudah dipadamkan");
}

```

```

putchar(0x1A);          // CTRL + Z
while(getchar()!='K'){};
}

// Jika yg diterima sms "$OFF 2", maka lampu 2 akan padam
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='2')
{
PORTC.1=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 2 sudah dipadamkan");
 putchar(0x1A);          // CTRL + Z
 while(getchar()!='K'){};
}

// Jika yg diterima sms "$OFF 3", maka lampu 3 akan padam
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='3')
{
PORTC.2=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 3 sudah dipadamkan");
 putchar(0x1A);          // CTRL + Z
 while(getchar()!='K'){};
}

// Jika yg diterima sms "$OFF 4", maka lampu 4 akan padam
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='4')

```

```

{
PORTC.3=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 4 sudah dipadamkan");
 putchar(0x1A);          // CTRL + Z
 while(getchar()!='K'){};
}

```

```

// Jika yg diterima sms "$OFF 5", maka lampu 5 akan padam
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='5')

```

```

{
PORTC.4=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 5 sudah dipadamkan");
 putchar(0x1A);          // CTRL + Z
 while(getchar()!='K'){};
}

```

```

// Jika yg diterima sms "$OFF 6", maka lampu 6 akan padam
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'
&& buffer[4]==' ' && buffer[5]=='6')

```

```

{
PORTC.5=0;
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
 kirim_pesan ();
 printf("Lampu 6 sudah dipadamkan");
 putchar(0x1A);          // CTRL + Z
 while(getchar()!='K'){};
}

```

```
}
```

```
// Jika yg diterima sms "$OFF 7", maka lampu 7 akan padam
```

```
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'  
&& buffer[4]==' ' && buffer[5]=='7')
```

```
{
```

```
PORTC.6=0;
```

```
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
```

```
 kirim_pesan ();
```

```
 printf("Lampu 7 sudah dipadamkan");
```

```
 putchar(0x1A); // CTRL + Z
```

```
 while(getchar()!='K'){};
```

```
}
```

```
// Jika yg diterima sms "$OFF 8", maka lampu 8 akan padam
```

```
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' &&buffer[3]=='F'  
&& buffer[4]==' ' && buffer[5]=='8')
```

```
{
```

```
PORTC.7=0;
```

```
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
```

```
 kirim_pesan ();
```

```
 printf("Lampu 8 sudah dipadamkan");
```

```
 putchar(0x1A); // CTRL + Z
```

```
 while(getchar()!='K'){};
```

```
}
```

```
// Jika yg diterima sms "$OFF ALL", maka semua lampu akan padam
```

```
if( buffer[0]=='$' && buffer[1]=='O' &&buffer[2]=='F' && buffer[3]=='F'  
&& buffer[4]==' ' && buffer[5]=='A' && buffer[6]=='L' && buffer[7]=='L')
```

```
{
```

```
PORTC=0X00;
```



```
for(a=0;a<10;a++) buffer[a]=0; // bersihkan buffer
    kirim_pesan ();
    printf("Semua lampu sudah dipadamkan");
    putchar(0x1A);          // CTRL + Z
    while(getchar()!='K'){};
    }
};
}
```