Antarmuka & Peripheral

Lampu Lalu Lintas

Nama : Irawan

NIM : 0901128621057

Kelas : SK7B

Dosen Pengampuh : Rendyansyah S.Kom., M.T.

Jurusan Sistem Komputer

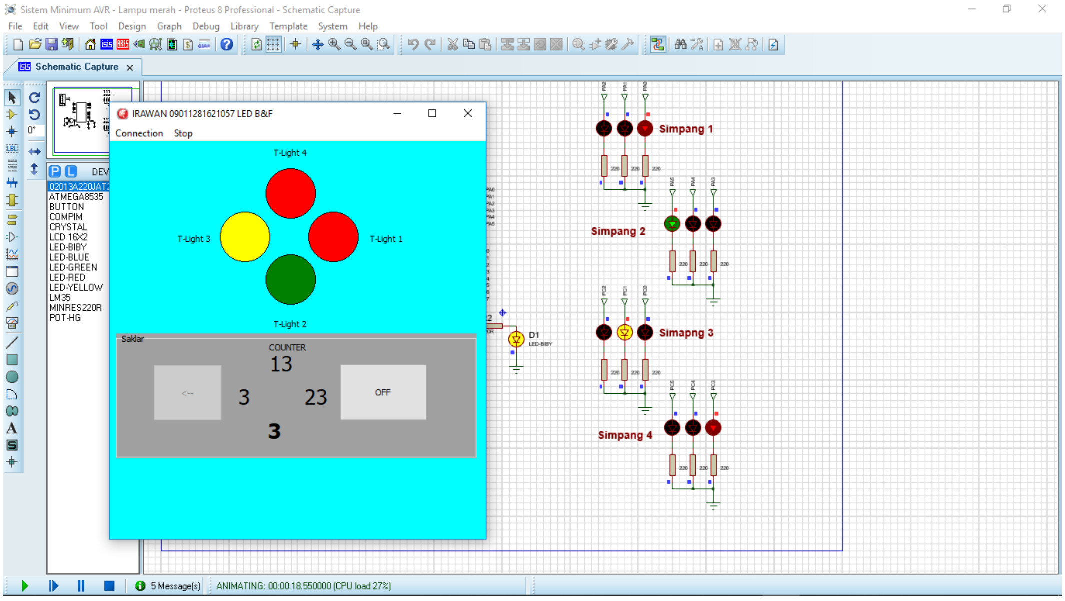
Fakultas Ilmu Komputer

Universitas Sriwijaya

2018

Tugas :

Antarmuka:



Code CVAVR:

```
/*****************************************************

This program was created by the

CodeWizardAVR V3.12 Advanced

Automatic Program Generator

© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com


Project :

Version :

Date    : 03/10/2019

Author  :

Company :
```

```
Comments:



Chip type               : ATmega8535
Program type            : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 128
*****************************************************/

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

// Declare your global variables here
int angka(int bilangan)
{
   int s,p,h;

   p=bilangan/10;
   s=bilangan%10;

   h=(p*16)+s;
   return h;
```

```
}
void main(void)
{
int a;
// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
```

```
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);


// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);


// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;


// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected
// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (0<<CS11) | (0<<CS10);

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: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;
```

```
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) |
(0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) |
(0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
```

```
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) |
(1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)
      {
      // Place your code here
      scanf("%d",&a);
      PORTA=a;
      scanf("%d",&a);
      PORTC=a;
      };

}
```

Code Delphi:

```
unit Unit2;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Vcl.StdCtrls, Vcl.ExtCtrls,
  CPort;
```

```
type
led_array = array[0..8]of integer;
  TForm2 = class(TForm)
    MainMenu1: TMainMenu;
    Connection1: TMenuItem;
    Setting1: TMenuItem;
    Connect1: TMenuItem;
    Stop1: TMenuItem;
    Exit1: TMenuItem;
    ComPort1: TComPort;
    GroupBox1: TGroupBox;
    Button1: TButton;
    Button2: TButton;
    Timer1: TTimer;
    Label1: TLabel;
    Label2: TLabel;
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Label3: TLabel;
    Timer2: TTimer;
    Timer3: TTimer;
    Timer4: TTimer;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Timer5: TTimer;
    procedure Exit1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Setting1Click(Sender: TObject);
    procedure Connect1Click(Sender: TObject);
    procedure status_led(led:led_array);
    procedure Stop1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure Timer3Timer(Sender: TObject);
    procedure Timer4Timer(Sender: TObject);
    procedure Timer5Timer(Sender: TObject);


  private
    { Private declarations }
  public
   { Public declarations }
```

```
    left:integer;
   right:integer;
   sat:integer;
   pul:integer;
   mer:integer;
   kun:integer;
   hij:integer;
   sw1:integer;
   sw2:integer;
   sw3:integer;
   sw4:integer;
   send1:integer;
   send2:integer;
   send3:integer;
   send4:integer;
  end;

var
  Form2: TForm2;
  hitung1:integer;
  hitung2:integer;
  hitung3:integer;
  hitung4:integer;
implementation
```

```
{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
if button1.Caption='<--' then
Begin
hitung1:=10;
hitung2:=10;
hitung3:=20;
hitung4:=30;
sw1:=3;
Timer1.Enabled:=True;
Timer2.Enabled:=True;
Timer3.Enabled:=True;
Timer4.Enabled:=True;
Timer5.Enabled:=True;
button1.Enabled:=False;
button2.Enabled:=True;
mer:=30;
end;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
if button2.Caption='OFF' then
```

```
Begin

Timer1.Enabled:=False;

Timer2.Enabled:=False;

Timer3.Enabled:=False;

Timer4.Enabled:=False;

Timer5.Enabled:=False;

button1.Enabled:=True;

button2.Enabled:=False;

end;

end;



procedure TForm2.Connect1Click(Sender: TObject);

begin

Comport1.Open();

Comport1.Connected := True

end;


procedure TForm2.Exit1Click(Sender: TObject);

begin

    Application.Terminate

end;


procedure TForm2.FormCreate(Sender: TObject);

begin
```

```
Button2.Enabled:=False;
hitung1:=0;
end;

procedure TForm2.Setting1Click(Sender: TObject);
begin
comport1.ShowSetupDialog()
end;

function pangkat(b:integer):Integer;
var
i,a:Integer;
begin
  a:=1;
  for i := b downto 1 do
  begin
    a:=a*2;
  end;
  pangkat:=a;
end;

procedure TForm2.status_led(led:led_array);
begin
  if led[0]=1 then
  begin
    Shape1.Brush.Color:=clRed;
  end
  else
  begin
    Shape1.Brush.Color:=clWhite;
  end;

  if led[1]=1 then
  begin
    Shape2.Brush.Color:=clRed;
  end
  else
  begin
    Shape2.Brush.Color:=clWhite;
  end;

  if led[2]=1 then
  begin
    Shape3.Brush.Color:=clRed;
  end
  else
  begin
    Shape3.Brush.Color:=clWhite;
  end;
```

```
  if led[3]=1 then
  begin
    Shape4.Brush.Color:=clRed;
  end
  else
  begin
    Shape4.Brush.Color:=clWhite;
  end;
end;

procedure TForm2.Stop1Click(Sender: TObject);
begin
Comport1.Close();
Comport1.Connected := False
end;

procedure TForm2.Timer1Timer(Sender: TObject);
var
  led:led_array;
   a,b:integer;
   nbit:integer;
   send:integer;

begin
 b:=hitung1;
```

```
b:=b-1;

hitung1:=b;

if (sw1=0)  then

begin

Shape1.Brush.Color:=clRed;

send1:=1;

if b<=5 then

begin

sw1:=1;

end

end

else if (sw1=1)  then

begin

Shape1.Brush.Color:=clYellow;

send1:=2;

if b<=1 then

begin

sw1:=2;

end

end

else if (sw1=2) or (sw1=3) then

begin

Shape1.Brush.Color:=clGreen;

send1:=4;

if (b=0) and (sw1=2) then
```

```
 begin
 hitung1:=10;
 sw1:=3;
 end
 else if (b=0) and (sw1=3) then
 begin
 hitung1:=mer;
 sw1:=0;
 Shape1.Brush.Color:=clRed;
 end
 end;
 label2.Caption:=InttoStr(b);
 //Comport1.WriteStr(IntToStr(send)+#13);
end;

procedure TForm2.Timer2Timer(Sender: TObject);
var
  led:led_array;
   a,b:integer;
   nbit:integer;
   send:integer;

begin
 b:=hitung2;
 b:=b-1;
```

```
hitung2:=b;
if (sw2=0)  then
begin
Shape2.Brush.Color:=clRed;
send2:=8;
if b<=5 then
begin
sw2:=1;
end
end
else if (sw2=1)  then
begin
Shape2.Brush.Color:=clYellow;
send2:=16;
if b=1 then
begin
sw2:=2;
end
end
else if (sw2=2) or (sw2=3) then
begin
Shape2.Brush.Color:=clGreen;
send2:=32;
if (b=0) and (sw2=2) then
begin
```

```
 hitung2:=10;
 sw2:=3;
 end
 else if (b=0) and (sw2=3) then
 begin
 hitung2:=mer;
 sw2:=0;
 Shape2.Brush.Color:=clRed;
 end
 end;
 label1.Caption:=InttoStr(b);
 //Comport1.WriteStr(IntToStr(send)+#13);
end;

procedure TForm2.Timer3Timer(Sender: TObject);
var
  led:led_array;
   a,b:integer;
   nbit:integer;
   send:integer;

begin
 b:=hitung3;
 b:=b-1;
 hitung3:=b;
```

```
if (sw3=0)  then

begin

Shape3.Brush.Color:=clRed;

send3:=1;

if b<=5 then

begin

sw3:=1;

end

end

else if (sw3=1)  then

begin

Shape3.Brush.Color:=clYellow;

send3:=2;

if b=1 then

begin

sw3:=2;

end

end

else if (sw3=2) or (sw3=3) then

begin

Shape3.Brush.Color:=clGreen;

send3:=4;

if (b=0) and (sw3=2) then

begin

hitung3:=10;
```

```
 sw3:=3;

 end

 else if (b=0) and (sw3=3) then

 begin

 hitung3:=mer;

 sw3:=0;

 Shape3.Brush.Color:=clRed;

 end

 end;

 label3.Caption:=InttoStr(b);

 //Comport1.WriteStr(IntToStr(send)+#13);

end;


procedure TForm2.Timer4Timer(Sender: TObject);

var

  led:led_array;

   a,b:integer;

   nbit:integer;

   send:integer;


begin

 b:=hitung4;

 b:=b-1;

 hitung4:=b;

 if (sw4=0)  then
```

```
begin
Shape4.Brush.Color:=clRed;
send4:=8;
if b<=5 then
begin
sw4:=1;
end
end
else if (sw4=1)  then
begin
Shape4.Brush.Color:=clYellow;
send4:=16;
if b=1 then
begin
sw4:=2;
end
end
else if (sw4=2) or (sw4=3) then
begin
Shape4.Brush.Color:=clGreen;
send4:=32;
if (b=0) and (sw4=2) then
begin
hitung4:=10;
sw4:=3;
```

```
 end
 else if (b=0) and (sw4=3) then
 begin
 hitung4:=mer;
 sw4:=0;
 Shape4.Brush.Color:=clRed;
 end
 end;
 label4.Caption:=InttoStr(b);
 //Comport1.WriteStr(IntToStr(send)+#13);
end;

procedure TForm2.Timer5Timer(Sender: TObject);
begin
 Comport1.WriteStr(IntToStr(send1+send2)+#13);
 //Comport1.WriteStr(IntToStr(send2)+#13);
 Comport1.WriteStr(IntToStr(send3+send4)+#13);
 //Comport1.WriteStr(IntToStr(send4)+#13);
end;

end.
```