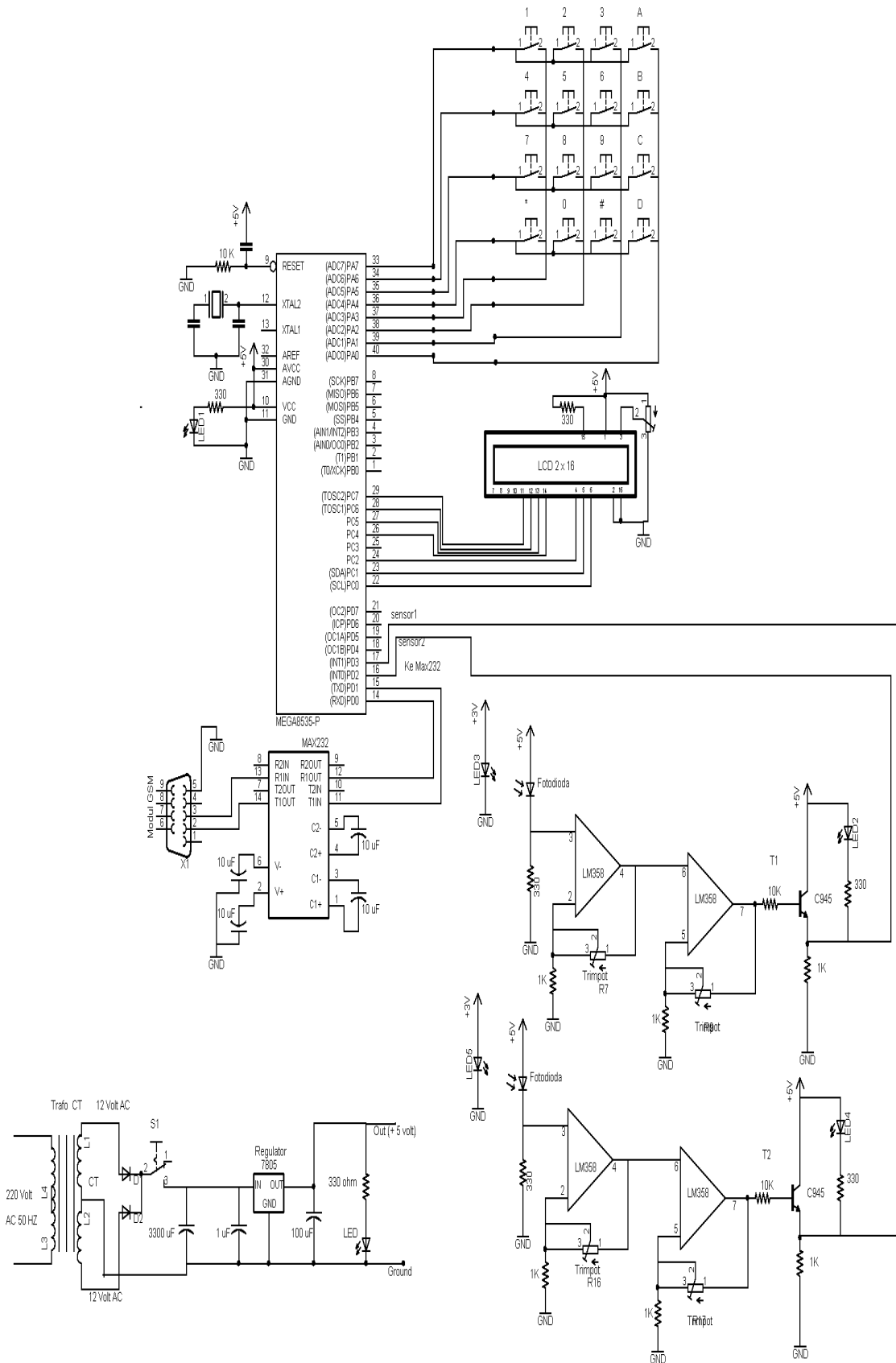


LAMPIRAN A RANGKAIAN LENGKAP



LAMPIRAN B
PROGRAM DI MIKROKONTROLER ATMEGA 8535

/******

This program was produced by the

CodeWizardAVR V1.25.8 Standard

Automatic Program Generator

© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 17/12/2011

Author : F4CG

Company : F4CG

Comments:

Chip type : ATmega8535

Program type : Application

Clock frequency : 11,059200 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 128

*****/

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

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x15 ;PORTC
```

```
#endasm
```

```

#include <lcd.h>
#include <stdio.h>
#include <delay.h>
int indikator2, kondisi, hitung, nilai1, nilai2, nilai3, indktor, temp, alamat;
int ps1, ps2, ps3, ps4, ps5, ps6, ps7, ps8, ps9, ps10, ps11, ps12, ps13;
float kec_m, kec_km, detik, jarak, v_maks;
unsigned char eeprom *pointer_eeprom;
unsigned char data1[16];

#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 8
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;
};
if (indikator2==12) {ps13=data; indikator2=13; }
if (indikator2==11) {ps12=data; indikator2=12; }
if (indikator2==10) {ps11=data; indikator2=11; }
if (indikator2==9) {ps10=data; indikator2=10; }
if (indikator2==8) {ps9=data; indikator2=9; }
if (indikator2==7) {ps8=data; indikator2=8; }
if (indikator2==6) {ps7=data; indikator2=7; }
if (indikator2==5) {ps6=data; indikator2=6; }
if (indikator2==4) {ps5=data; indikator2=5; }
if (indikator2==3) {ps4=data; indikator2=4; }
if (indikator2==2) {ps3=data; indikator2=3; }
if (indikator2==1) {ps2=data; indikator2=2; }
if (indikator2==0) {ps1=data; indikator2=1; }
ps11=data;
};

```

```

}

#ifdef _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 <Baca_keypad.c>;

// Declare your global variables here
void baca()
{
indktor=1;
while (indktor==1)
{
PORTA.0=0;
PORTA.1=1;
PORTA.2=1;
}
}

```

```

PORTA.3=1;
delay_ms(2);
if (PINA.4==0) {temp=1; indktor=0;}
if (PINA.5==0) {temp=2; indktor=0;}
if (PINA.6==0) {temp=3; indktor=0;}
PORTA.0=1;
PORTA.1=0;
PORTA.2=1;
PORTA.3=1;
delay_ms(2);
if (PINA.4==0) {temp=4; indktor=0;}
if (PINA.5==0) {temp=5; indktor=0;}
if (PINA.6==0) {temp=6; indktor=0;}
PORTA.0=1;
PORTA.1=1;
PORTA.2=0;
PORTA.3=1;
delay_ms(2);
if (PINA.4==0) {temp=7; indktor=0;}
if (PINA.5==0) {temp=8; indktor=0;}
if (PINA.6==0) {temp=9; indktor=0;}
PORTA.0=1;
PORTA.1=1;
PORTA.2=1;
PORTA.3=0;
delay_ms(2);
if (PINA.5==0) {temp=0; indktor=0;}
PORTA.3=1;
}
}

```

```

void main(void)

```

```

{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=Out
Func0=Out
// State7=P State6=P State5=P State4=P State3=1 State2=1 State1=1 State0=1
PORTA=0xFF;
DDRA=0x0F;

// 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=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
PORTC=0x00;
DDRC=0x00;

// 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=P State2=P State1=T State0=T
PORTD=0x0C;
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;
```

```

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")
indikator2=0;
kondisi=0;
hitung=0;
pointer_eeprom=0;
alamat=*pointer_eeprom;
if (alamat==255.00)
{
    pointer_eeprom=0;    // alamat penyimpanan data kec max
    *pointer_eeprom=001;

    pointer_eeprom=0;
    pointer_eeprom=pointer_eeprom+1;    // alamat penyimpanan data jarak
    *pointer_eeprom=5;
}
pointer_eeprom=0;
v_maks=*pointer_eeprom;

pointer_eeprom=0;
pointer_eeprom=pointer_eeprom+1;
jarak=*pointer_eeprom;
jarak=jarak/10;
while (1)
{
    // Place your code here
    if (ps8==82 && ps9==101 && ps10==97)
    {
        kondisi=1;
    }
}

```

```

lcd_gotoxy(0,0);
lcd_putsf("My Project");
lcd_gotoxy(0,1);
lcd_putsf("Sistem Ready");
delay_ms(1000);
while (kondisi==1)
{
if (PIND.3==0)
{
while (PIND.2==1) {delay_ms(2); hitung=hitung+1;}
delay_ms(50);
pointer_eeprom=0;
pointer_eeprom=pointer_eeprom+1;
jarak=*pointer_eeprom;
jarak=jarak/10;
detik=hitung;
detik=detik/500;
kec_m=jarak/detik;
lcd_clear();
sprintf(data1,"t=%0.03fs",detik);
lcd_gotoxy(0,0);
lcd_puts(data1);
sprintf(data1,"S=%0.01f",jarak);
lcd_gotoxy(10,0);
lcd_puts(data1);
sprintf(data1,"v=%0.05f m/dtk",kec_m);
lcd_gotoxy(0,1);
lcd_puts(data1);
delay_ms(5000);

kec_km=kec_m*3.6;
lcd_clear();
sprintf(data1,"t=%0.03fs",detik);

```

```

lcd_gotoxy(0,0);
lcd_puts(data1);
sprintf(data1,"S=%0.01f",jarak);
lcd_gotoxy(10,0);
lcd_puts(data1);
sprintf(data1,"v=%0.05f Km/jam",kec_km);
lcd_gotoxy(0,1);
lcd_puts(data1);

if (kec_km>=v_maks)
{
printf("at+cmgs=");
putchar("");
printf("+6285359710965");
putchar("");
printf(",");
delay_ms(1000);
putchar(13);
delay_ms(1000);
printf("Kecepatan Melampaui Batas v=%0.3f Km/jam\r\n", kec_km);
delay_ms(1000);
putchar(26);
putchar(13);
delay_ms(1000);
}
delay_ms(5000);

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("My Project");
lcd_gotoxy(0,1);
lcd_putsf("Sistem Ready");
hitung=0;

```

```

delay_ms(100);
}

PORTA.0=0;
delay_ms(2);
if (PINA.7==0)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf(" Setting v Maks");
lcd_gotoxy(0,1);
lcd_putsf("v Maks= ");
  baca();
  nilai1=temp;
  sprintf(data1,"%d",nilai1);
  lcd_gotoxy(8,1);
  lcd_puts(data1);
  delay_ms(1000);
  baca();
  nilai2=temp;
  sprintf(data1,"%d",nilai2);
  lcd_gotoxy(9,1);
  lcd_puts(data1);
  delay_ms(1000);
  baca();
  nilai3=temp;
  sprintf(data1,"%d",nilai3);
  lcd_gotoxy(10,1);
  lcd_puts(data1);
  delay_ms(1000);
  nilai1=(nilai1*100)+(nilai2*10)+nilai3;
  v_maks=nilai1;
  pointer_eeprom=0;      // alamat penyimpanan data kec max

```

```

*pointer_eeprom=nilai1;
PORTA.0=1;
sprintf(data1,"%d",nilai1);
lcd_gotoxy(13,1);
lcd_puts(data1);
delay_ms(3000);
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf(" Setting Jarak");
lcd_gotoxy(0,1);
lcd_putsf("Jarak= ");
  baca();
  nilai1=temp;
  sprintf(data1,"%d",nilai1);
  lcd_gotoxy(7,1);
  lcd_puts(data1);
  delay_ms(1000);
  baca();
  nilai2=temp;
  sprintf(data1,"%d",nilai2);
  lcd_gotoxy(8,1);
  lcd_puts(data1);
  delay_ms(1000);
  baca();
  nilai3=temp;
  sprintf(data1,"%d",nilai3);
  lcd_gotoxy(9,1);
  lcd_puts(data1);
  delay_ms(1000);
  nilai1=(nilai1*100)+(nilai2*10)+nilai3;
  pointer_eeprom=0;
  pointer_eeprom=pointer_eeprom+1;    // alamat penyimpanan data jarak
*pointer_eeprom=nilai1;

```

```

    jarak=nilai1;
    jarak=jarak/10;
    PORTA.1=1;
    sprintf(data1,"%0.01f",jarak);
    lcd_gotoxy(11,1);
    lcd_puts(data1);
    delay_ms(3000);
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("My Project");
    lcd_gotoxy(0,1);
    lcd_putsf("Sistem Ready");
    delay_ms(100);
} }}
else
{
    lcd_gotoxy(0,0);
    lcd_putchar(ps1);
    lcd_gotoxy(1,0);
    lcd_putchar(ps2);
    lcd_gotoxy(2,0);
    lcd_putchar(ps3);
    lcd_gotoxy(3,0);
    lcd_putchar(ps4);
    lcd_gotoxy(4,0);
    lcd_putchar(ps5);
    lcd_gotoxy(5,0);
    lcd_putchar(ps6);
    lcd_gotoxy(6,0);
    lcd_putchar(ps7);
    lcd_gotoxy(7,0);
    lcd_putchar(ps8);
    lcd_gotoxy(8,0);

```

```
    lcd_putchar(ps9);
    lcd_gotoxy(9,0);
    lcd_putchar(ps10);
    lcd_gotoxy(10,0);
    lcd_putchar(ps11);
    lcd_gotoxy(11,0);
    lcd_putchar(ps12);
    lcd_gotoxy(12,0);
    lcd_putchar(ps13);
    indikator2=0;

    sprintf(data1,"%d",ps8);
    lcd_gotoxy(0,1);
    lcd_puts(data1);

    sprintf(data1,"%d",ps9);
    lcd_gotoxy(6,1);
    lcd_puts(data1);

    sprintf(data1,"%d",ps10);
    lcd_gotoxy(11,1);
    lcd_puts(data1);
    delay_ms(300);
    lcd_clear();
}

};

}
```


LAMPIRAN C PPROGRAM VISUAL BASIC

```
=====
Public con As New ADODB.Connection
Public rssms As ADODB.Recordset
```

```
Public Sub connectDatabase()
Set con = New ADODB.Connection
Set rssms = New ADODB.Recordset
con.Open           "DRIVER={MySQL           ODBC           3.51
Driver};DESC=;DATABASE=smsd;SERVER=localhost;UID=root;PASSWORD=;P
ORT=3306;OPTION=;STMT=;"
End Sub
```

```
=====
Private Sub Command1_Click()

Unload Me

End Sub
```

```
Private Sub Form_Load()

Dim indeks As Integer

Timer1.Enabled = True

Call connectDatabase

rssms.CursorLocation = adUseClient

rssms.Open "select ReceivingDateTime, TextDecoded from inbox", con,
adOpenStatic, adLockOptimistic

Set DataGrid1.DataSource = rssms.DataSource

DataGrid1.Columns(1).Width = 5700

DataGrid1.Columns(0).Width = 2450

End Sub
```

```
Private Sub Timer1_Timer()

Call connectDatabase
```

```
rssms.CursorLocation = adUseClient
rssms.Open "select ReceivingDateTime, TextDecoded from inbox ORDER by
ReceivingDateTime ASC", con, adOpenStatic, adLockOptimistic
Set DataGrid1.DataSource = rssms.DataSource
DataGrid1.Columns(1).Width = 5700
DataGrid1.Columns(0).Width = 2450
'Me.DataGrid1.Columns(1).Width = Len(Me.DataGrid1.Columns(1).Text)
rssms.MoveLast
Text1.Text = rssms(0)
Text2.Text = rssms(1)
End Sub
```