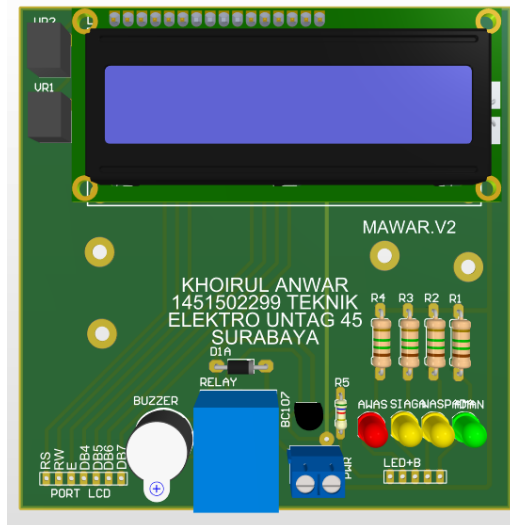
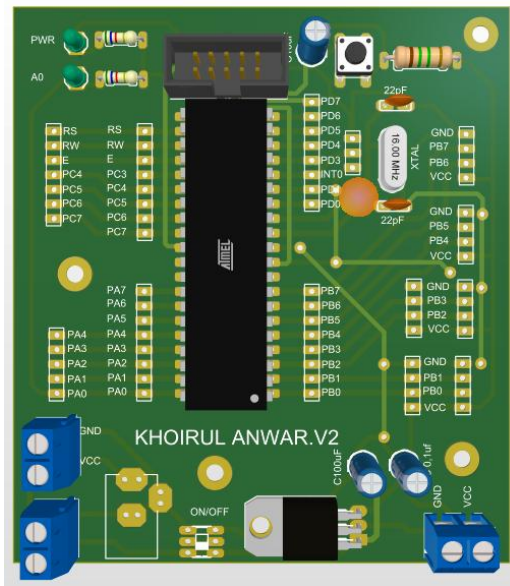
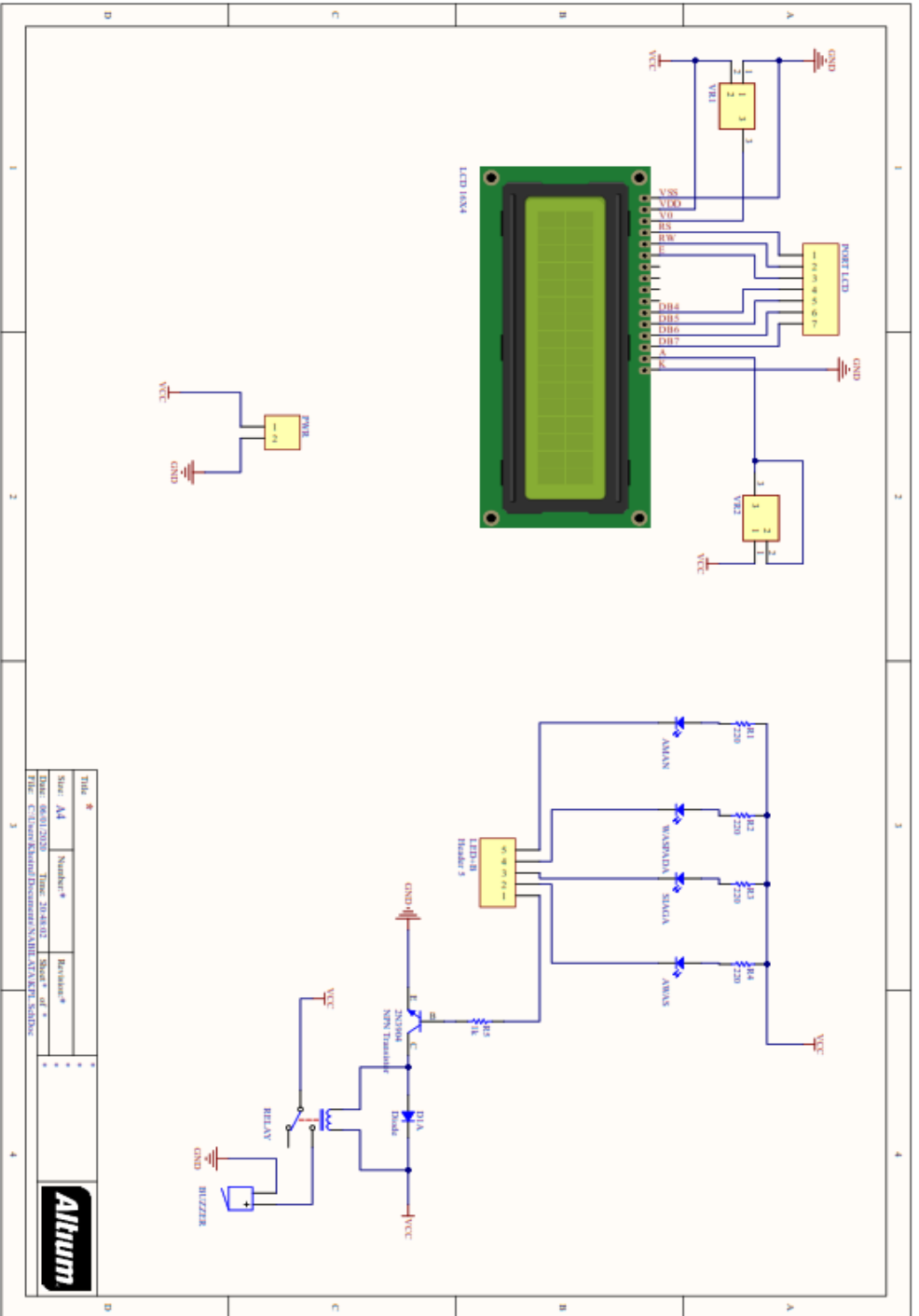


Lampiran 1 Hardware Blind Spot





Lampiran 3 Program Sistem Blind Spot

/******

This program was produced by the

CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project : TUGAS AKHIR

Version : 1.0

Date : 05/01/2020

Author : KHOIRUL ANWAR

Company : UNIVERSITAS 17 AGUSTUS 1945 SURABAYA

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 16,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

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

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

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#asm
```

```
.equ __lcd_port=0x15 ; PORTC

#endasm

// Alphanumeric LCD functions

#include <alcd.h>

int frekuensi=0;

float pulsa;

char temp[8];

unsigned int count=0;

unsigned int count1=0;

unsigned int count2=0;

unsigned int count3=0;

unsigned char jrk;

unsigned char jrk1;

unsigned char jrk2;

unsigned char jrk3;

unsigned char blind_spot;

unsigned char blind_spot1;

unsigned char blind_spot2;

unsigned char blind_spot3;

unsigned char kata1[16];

unsigned char kata2[16];

unsigned char kata3[16];

unsigned char kata4[16];

unsigned char kata5[16];

unsigned char kata6[16];

unsigned char kata7[16];
```

```
unsigned char kata8[16];

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here

    frekuensi++;
}

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Reinitialize Timer1 value
    TCNT1H=0xD23A >> 8;
    TCNT1L=0xD23A & 0xff;

    // Place your code here

    pulsa=(float)frekuensi*60/15;//formula perhitungan kecepatan
    frekuensi=0;
}

void k_k()
{
    ftoa(pulsa,1,temp);

    if(pulsa <20)
    {
        PORTA.4=1;
    }
    else
```

```

    {
        PORTA.4=0;
    }

}

//SENSOR DEPAN
unsigned char depan()
{
    count=0;

    //pemberian pemicu berupa pulsa positif minimal 10us
    PORTB.1=1;
    delay_us(5);
    PORTB.1=0;

    //menunggu hingga ECHO = 1
    while(PINB.0==0){};

    //mencacah untuk mengukur waktu lamanya pulsa 1 dari ECHO
    while(PINB.0==1)
    {
        count++;
        delay_us(1);
    }

    //mengkonversikan hasil cacahan menjadi jarak dalam cm
    jrk=((float)count/29.034/2);

    return jrk;
}

//SENSOR BELAKANG

```

```

unsigned char belakang()
{
    count1=0;
    //pemberian pemicu berupa pulsa positif minimal 10us
    PORTB.3=1;
    delay_us(5);
    PORTB.3=0;
    //menunggu hingga ECHO = 1
    while(PINB.2==0){};
    //mencacah untuk mengukur waktu lamanya pulsa 1 dari ECHO
    while(PINB.2==1)
    {
        count1++;
        delay_us(1);
    }
    //mengkonversikan hasil cacahan menjadi jarak dalam cm
    jrk1=((float)count1/29.034/2);
    return jrk1;
}

```

```
//SENSOR KIRI
```

```

unsigned char kiri()
{
    count2=0;
    //pemberian pemicu berupa pulsa positif minimal 10us

```



```

PORTB.5=1;
delay_us(5);
PORTB.5=0;

//menunggu hingga ECHO = 1
while(PINB.4==0){};

//mencacah untuk mengukur waktu lamanya pulsa 1 dari ECHO
while(PINB.4==1)
{
count2++;
delay_us(1);
}

//mengkonversikan hasil cacahan menjadi jarak dalam cm
jrk2=((float)count2/29.034/2);

return jrk2;
}

//SENSOR KANAN
unsigned char kanan()
{
count3=0;

//pemberian pemicu berupa pulsa positif minimal 10us
PORTB.7=1;
delay_us(5);
PORTB.7=0;

//menunggu hingga ECHO = 1
while(PINB.6==0){};

//mencacah untuk mengukur waktu lamanya pulsa 1 dari ECHO

```

```

        while(PINB.6==1)
        {
            count3++;
            delay_us(1);
        }

        //mengkonversikan hasil cacahan menjadi jarak dalam cm
        jrk3=((float)count3/29.034/2);
        return jrk3;
    }

// 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=Out Func3=Out Func2=Out Func1=Out
    Func0=Out

    // State7=T State6=T State5=T State4=0 State3=0 State2=0 State1=0 State0=0

    PORTA=0x00;

    DDRA=0x1F;

    // Port B 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

    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=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=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 250,000 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
```

```
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x03;
TCNT1H=0xD2;
TCNT1L=0x3A;
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=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
```

```
// INT2: Off

GICR|=0x40;

MCUCR=0x02;

MCUCSR=0x00;

GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x04;

// USART initialization

// USART disabled

UCSRB=0x00;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// ADC initialization

// ADC disabled

ADCSRA=0x00;

// SPI initialization

// SPI disabled

SPCR=0x00;

// TWI initialization

// TWI disabled

TWCR=0x00;

// Alphanumeric LCD initialization

// Connections are specified in the
```

```
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("KHOIRUL ANWAR");
lcd_gotoxy(0,1);
lcd_putsf("NBI 1451502299");
delay_ms(100);
lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("=TUGAS AKHIR=");
lcd_gotoxy(3,1);
lcd_putsf("BLIND SPOT");
delay_ms(100);
// Global enable interrupts
#asm("sei")
while (1)
{
    // Place your code here
```

```
blind_spot=depan();  
sprintf(kata1,"%1d cm",blind_spot);  
if (blind_spot <5)  
{  
    sprintf(kata2,"AWAS");  
    PORTA.0=0; PORTA.1=1; PORTA.2=1; PORTA.3=1; k_k();  
}  
else  
{  
    sprintf(kata2,"SIAGA");  
    PORTA.0=1;PORTA.1=0; PORTA.2=1; PORTA.3=1; PORTA.4=1;  
}  
if (blind_spot >15 )  
{  
    sprintf(kata2,"WASPADA");  
    PORTA.0=1; PORTA.1=1; PORTA.2=0; PORTA.3=1; PORTA.4=1;  
}  
if (blind_spot >20)  
{  
    sprintf(kata2,"AMAN");  
    PORTA.0=1; PORTA.1=1; PORTA.2=1; PORTA.3=0; PORTA.4=1;  
};  
  
//BELAKANG  
blind_spot1=belakang();  
sprintf(kata3,"%1d cm",blind_spot1);  
if (blind_spot1 <5)
```

```

    {
        sprintf(kata4,"AWAS");
        PORTA.0=0; PORTA.1=1; PORTA.2=1; PORTA.3=1; k_k();
    }
else
    {
        sprintf(kata4,"SIAGA");
        PORTA.0=1;PORTA.1=0; PORTA.2=1; PORTA.3=1; PORTA.4=1;
    }
if (blind_spot1 >15 )
    {
        sprintf(kata4,"WASPADA");
        PORTA.0=1; PORTA.1=1; PORTA.2=0; PORTA.3=1; PORTA.4=1;
    }
if (blind_spot1 >20)
    {
        sprintf(kata4,"AMAN");
        PORTA.0=1; PORTA.1=1; PORTA.2=1; PORTA.3=0; PORTA.4=1;
    };
//KIRI
blind_spot2=kiri();
sprintf(kata5,"%1d cm",blind_spot2);
if (blind_spot2 <5)
    {
        sprintf(kata6,"AWAS");
        PORTA.0=0; PORTA.1=1; PORTA.2=1; PORTA.3=1; k_k();
    }

```



```
    }  
else  
    {  
        sprintf(kata6,"SIAGA");  
        PORTA.0=1;PORTA.1=0; PORTA.2=1; PORTA.3=1; PORTA.4=1;  
    }  
if (blind_spot2 >15 )  
    {  
        sprintf(kata6,"WASPADA");  
        PORTA.0=1; PORTA.1=1; PORTA.2=0; PORTA.3=1; PORTA.4=1;  
    }  
if (blind_spot2 >20)  
    {  
        sprintf(kata6,"AMAN");  
        PORTA.0=1; PORTA.1=1; PORTA.2=1; PORTA.3=0; PORTA.4=1;  
    };  
//KANAN  
blind_spot3=kanan();  
sprintf(kata7,"%1d cm",blind_spot3);  
if (blind_spot3 <5)  
    {  
        sprintf(kata8,"AWAS");  
        PORTA.0=0; PORTA.1=1; PORTA.2=1; PORTA.3=1; k_k();  
    }  
else  
    {
```

```
    sprintf(kata8,"SIAGA");
    PORTA.0=1;PORTA.1=0; PORTA.2=1; PORTA.3=1; PORTA.4=1;
}
if (blind_spot3 >15 )
{
    sprintf(kata8,"WASPADA");
    PORTA.0=1; PORTA.1=1; PORTA.2=0; PORTA.3=1; PORTA.4=1;
}
if (blind_spot3 >20)
{
    sprintf(kata8,"AMAN");
    PORTA.0=1; PORTA.1=1; PORTA.2=1; PORTA.3=0; PORTA.4=1;
};

lcd_clear();
//DEPAN
lcd_gotoxy(6,0);
lcd_puts(kata2);
lcd_gotoxy(0,0);
lcd_puts("DPN =");
//BELAKANG
lcd_gotoxy(6,1);
lcd_puts(kata4);
lcd_gotoxy(0,1);
lcd_puts("BLK =");
//KIRI
lcd_gotoxy(6,2);
```

```
    lcd_puts(kata6);  
    lcd_gotoxy(0,2);  
    lcd_puts("KRI =");  
    //KANAN  
    lcd_gotoxy(6,3);  
    lcd_puts(kata8);  
    lcd_gotoxy(0,3);  
    lcd_puts("KNN =");  
    delay_ms(400);  
  
    }  
}
```