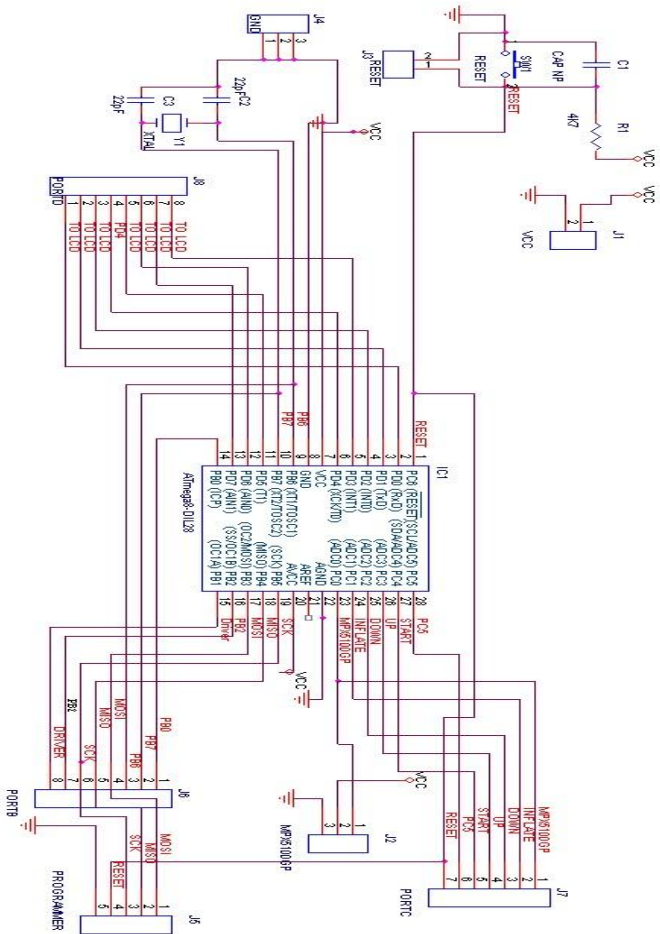


## DAFTAR PUSTAKA

- Atmel. (2013). *Datasheet Atmega* 8. <http://www.alldatasheet.com/datasheet-pdf/pdf/80247/ATMEL/ATMEGA8.html>. Diakses pada tanggal 21 September 2016 Pukul 07.00 WIB.
- Depkes RI. 1989. *Hematologi*. Pusdiknes Depkes RI: Jakarta.
- Drosos, GI; Ververidis, A; Mavropoulos, R; Vastardis, G; Tsioros, KI; Kazakos, K (September 2013). "*The silicone ring tourniquet in orthopaedic operations of the extremities*". *Surg Technol Int*. **23**: 251–7. Diakses pada tanggal 26 September 2016 pukul 19:57 WIB.
- Frances. K. Widman. 1989. *Tinjauan Klinis Atas Hasil Laboratorium*. Jakarta. Diakses pada tanggal 26 September 2016 pukul 19:57 WIB.
- Gavriely, N (May 2010). "*Surgical Tourniquets in Orthopaedics*". *J Bone Joint Surg Am*. **92A** (5): 1318–1322. Diakses pada tanggal 26 September 2016 pukul 12:56 WIB.
- McEwen, James A (June 2009). "*Tourniquet Overview*". [tourniquets.org](http://tourniquets.org). Retrieved.(2009-06-10). Diakses pada tanggal 15 September 2016 pukul 12:56 WIB.

# LAMPIRAN



/\*\*\*\*\*

\*\*\*\*\*

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 :

Version :

Date : 22/05/2017

Author : Abdul Hakim

Company : Gabdul Elek

Comments:

Chip type : ATmega8

Program type : Application

AVR Core Clock frequency: 1,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

```

*****
*****/

#include <mega8.h>

#include <delay.h>

// Alphanumeric LCD functions
#include <alcd.h>
#include <stdlib.h>
#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input
    voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
}

```

```
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here
int tekanan=100;
unsigned char temp [5];
int p,press;
long int buff_press[50],jumlah;
float press1,press2,press3;
char temp1 [10];

void read_press()
{
for (p=0;p<50;p++)
{
buff_press[p]=read_adc(0);
}
jumlah=0;
for (p=0;p<50;p++)
{
jumlah=jumlah+buff_press[p];
}
}
```

```

press1=(float)(jumlah/50);
press2= (float)(press1 * 5)/1023;// ubah ke tegangan
press3=          (float          )(press2          -
0.04*4.96)/(0.009*4.98)+(2.5*1*0.009*4.98); //transfer
function
press= (float)(press3*7.5-7);
}

void settekanan()
{
    lcd_clear();
    lcd_gotoxy(1,0);
    lcd_putsf("Atur Tekanan");
    itoa(tekanan,temp);
    lcd_gotoxy(4,1);
    lcd_puts(temp);
    delay_ms(100);
}

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

```

```
// Input/Output Ports initialization

// 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=0x01;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State6=P State5=P State4=P State3=P State2=P
State1=P State0=P
PORTC=0x7F;
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
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// 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=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: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

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

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

// 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 Clock frequency: 125,000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;

// 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 - PORTD Bit 0
// RD - PORTD Bit 1
// EN - PORTD Bit 2
// D4 - PORTD Bit 4
// D5 - PORTD Bit 5
// D6 - PORTD Bit 6
// D7 - PORTD Bit 7
// Characters/line: 16
lcd_init(16);

lcd_gotoxy (2,0);
lcd_putsf ("TOURNIQUET");
delay_ms (1000) ;
lcd_clear ();
```

```
lcd_gotoxy (0,0);
lcd_putsf ("Abdul Hakim M");
delay_ms(1000);
lcd_gotoxy (0,1);
lcd_putsf ("P27838014035");

while (PINC.4==1)
{
    settekanan();
    PORTB.0=0;
    read_press();
    itoa (press,temp1);
    lcd_gotoxy(0,1);
    lcd_puts(temp1);
    ftoa (press2,1,temp1);
    lcd_gotoxy(13,1);
    lcd_puts(temp1);
    if (PINC.2==0) // Push Button UP
    {
        tekanan = tekanan +10;
        delay_ms (400);
        if (tekanan > 500)
```

```
{
    tekanan = 100;
}
}
if (PINC.3==0)// Push Button Down
{
    tekanan = tekanan - 10;
    delay_ms (400);
    if (tekanan <100)
    {
        tekanan = 500;
    }
}
if (PINC.1==0)
{
    lcd_clear();
    PORTB.0=1;
    lcd_putsf("Inflate:");
    read_press();
    itoa (press,temp1);
    lcd_gotoxy(3,1);
    lcd_puts(temp1);
    lcd_gotoxy(7,1);
```

```
lcd_putsf("mmHg");
ftoa (press2,1,temp1);
lcd_gotoxy(13,1);
lcd_puts(temp1);
}
delay_ms(500);
}
while (1)

{
lcd_clear();
lcd_gotoxy(3,0);
lcd_putsf("Tekanan:");
read_press();
itoa (press,temp1);
lcd_gotoxy(3,1);
lcd_puts(temp1);
lcd_gotoxy(7,1);
lcd_putsf("mmHg");
ftoa (press2,1,temp1);
lcd_gotoxy(13,1);
lcd_puts(temp1);
if (PINC.1==0)
```

```

{
PORTB.0=1;
lcd_clear();
lcd_gotoxy(3,0);
lcd_putsf("Inflate:");
read_press();
itoa (press,temp1);
lcd_gotoxy(3,1);
lcd_puts(temp1);
lcd_gotoxy(7,1);
lcd_putsf("mmHg");
}
else if (press >= tekanan )
{
if (PINC.1==0)//INFLATE
{
PORTB.0=1;
lcd_clear();
lcd_gotoxy(3,0);
lcd_putsf("Inflate:");
read_press();
itoa (press,temp1);
lcd_gotoxy(3,1);

```

```
    lcd_puts(temp1);  
    lcd_gotoxy(7,1);  
    lcd_putsf("mmHg");  
    }  
    else  
    {  
    PORTB.0=0;  
    }  
    }  
    else  
    {  
    PORTB.0=1;  
    }  
    delay_ms(100);  
    }  
}
```