LAMPIRAN

Kajian Pustaka

Fitur - Krisbow KW06-290

- Dua modus memberikan 2.5dB 3.5dB atau akurasi
- A dan berat C pengukuran tinggi dan rendah berkisar: Rendah (35 sampai 100dB) tinggi (65 sampai 130dB)
- Resolusi 0.1dB • Cepat / Lambat respone
- Data terus dan max. memegang
- Besar layar LCD digital ½ dengan indikasi fungsi
- Built-in kalibrasi cek (94 dB)
- Lengkap dengan baterai 9V dan tas

Spesifikasi - Krisbow KW06-290

- Basic Accuracy : ±1.4dB at 94dB
- Brand : Krisbow
- Dimension (LxWxH) (mm) : 251x63.8x40
- High Range (dB) : 65-130
- Low Range (dB) : 35-100
- Model : KW06-290
- Weight (kg) : 0.5

Hasil perhitungan dari data **Tabel 4.3**

$\text{Simpangan } (\mathbf{No.1}) =$

$$\text{Alat ukur} - \text{Modul} = 40 - 41{,}1 = 1{,}1$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{40} \times 100\% = 2{,}75$$

$\text{Simpangan } (\mathbf{No.2}) =$

$$\text{Alat ukur} - \text{Modul} = 55 - 51{,}3 = 3{,}7$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{3{,}7}{55} \times 100\% = 6{,}72$$

$\text{Simpangan } (\mathbf{No.3}) =$

$$\text{Alat ukur} - \text{Modul} = 70 - 68{,}9 = 1{,}1$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{70} \times 100\% = 1{,}57$$

Simpangan = $1{,}1 + 3{,}7 + 1{,}1 = 5{,}9$

$$= \frac{5{,}9}{3} = 1{,}9$$

Rata – rata error% = $2{,}75 + 6{,}72 + 1{,}51$ = 10,98

$$= \frac{10{,}98}{3} = 3.66$$

Simpangan = 1,9 dan Rata – rata *error%* = 3,66

Keterangan perhitungan Tabel 4.3 adalah modul alat ini dapat mengukur kebisingan di luar *baby incubator* pada *range* minimal 41,1db dan maksimal di 68,9db dengan nilai simpangan 1,9 dan rata – rara *error %* sebesar 3,66.

Tabel 4.5 Hasil perbandingan

$\text{Simpangan }(\mathbf{No.\,1}) =$

$$\text{Alat ukur} - \text{Modul} = 40 - 40{,}2 = 0{,}2$$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{0{,}2}{40} \times 100\% = 0{,}5$$

$\text{Simpangan }(\mathbf{No.\,2})$

$$\text{Alat ukur} - \text{Modul} = 55 - 53{,}0 = 2{,}0$$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{2{,}0}{55} \times 100\% = 3{,}6$$

Simpangan $(\mathbf{No. 3}) =$

$$\text{Alat ukur} - \text{Modul} = 70 - 68{,}9 = 1{,}1$$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{70} \times 100\% = 1{,}57$$

Simpangan = $0{,}20 + 2{,}00 + 1{,}10 = 3{,}3$

$$= \frac{3{,}3}{3} = 1{,}1$$

Rata – rata *error%* = $0{,}50 + 3{,}60 + 1{,}57$ = 10,48

$$= \frac{3{,}49}{3} = 1{,}00$$

Simpangan = 1,9 dan Rata – rata *error%* = 3,49

Keterangan perhitungan Tabel 4.5 adalah modul alat ini dapat mengukur kebisingan di dalam *baby incubator* pada *range* minimal 40,2db dan maksimal di 68,9db dengan nilai simpangan 1,9 dan rata – rara *error %* sebesar 3,49

Hasil perhitungan data kalibrasi dari **Tabel 4.6**

Simpangan $(\mathbf{No. 1}) = \text{Alat ukur} - \text{Modul} = 52 - 48.9 = 3{,}1$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{3{,}1}{52} \times 100\%$$

$$= 5{,}96\%$$

$\text{Simpangan } (\mathbf{No.\,2}) = \text{Alat ukur } - \text{ Modul} = 53 - 51{,}0 = 2$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{2}{53} \times 100\%$$

$$= 3{,}77\%$$

$\text{Simpangan } (\mathbf{No.\,3}) = \text{Alat ukur } - \text{ Modul} = 54 - 51{,}9 = 2{,}1$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{2{,}1}{54} \times 100\%$$

$$= 3{,}88\%$$

$\text{Simpangan } (\mathbf{No.\,4}) = \text{Alat ukur } - \text{ Modul} = 55 - 53{,}8 = 1{,}2$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{1{,}2}{55} \times 100\%$$

$$= 2{,}18\%$$

$\text{Simpangan } (\mathbf{No.\,5}) = \text{Alat ukur } - \text{ Modul} = 56 - 54{,}9 = 1{,}1$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{1{,}1}{56} \times 100\% = 1{,}9\%$$

Total simpangan = $3{,}1 + 2 + 2{,}1 + 1{,}2 + 1{,}1 = 9{,}5$

$$= \frac{9{,}5}{5} = 1{,}9$$

Total rata-rata *error* = $5{,}96 + 3{,}77 + 3{,}88 + 2{,}18 + 1{,}9 = 17{,}69$

$$\frac{17{,}69}{5} = 3{,}538$$

Rata-rata *error%* = 3,538 %dan simpangan = 1,9 modulsensor kebisingan di dalam *baby incubator*.

Foto proses pembuatan

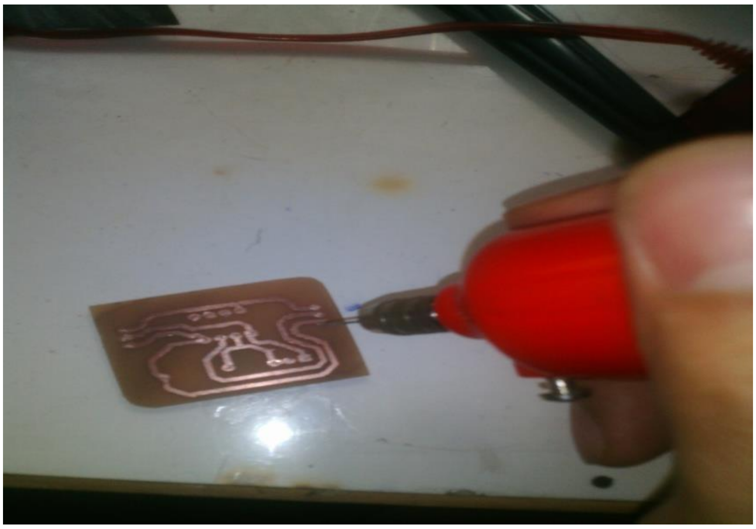
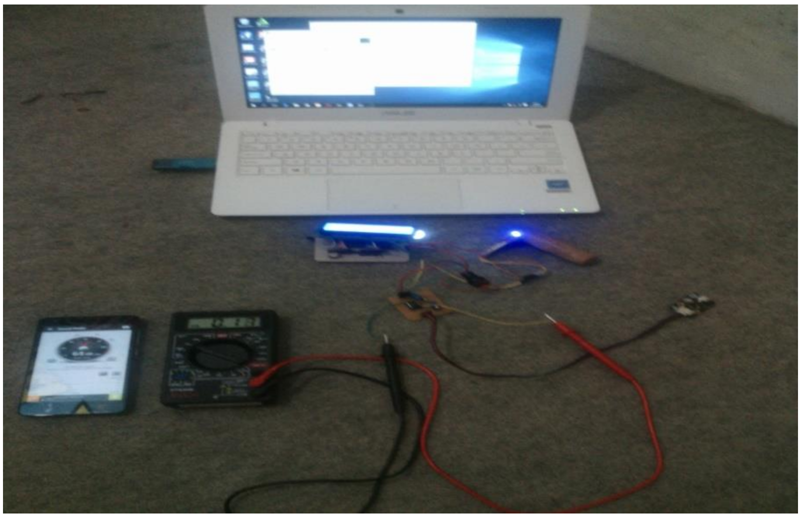


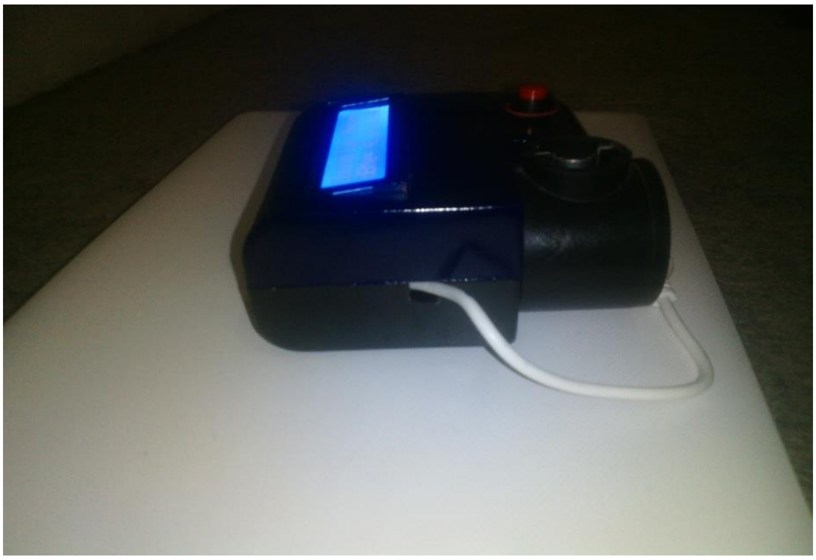
Foto proses pengambilan data

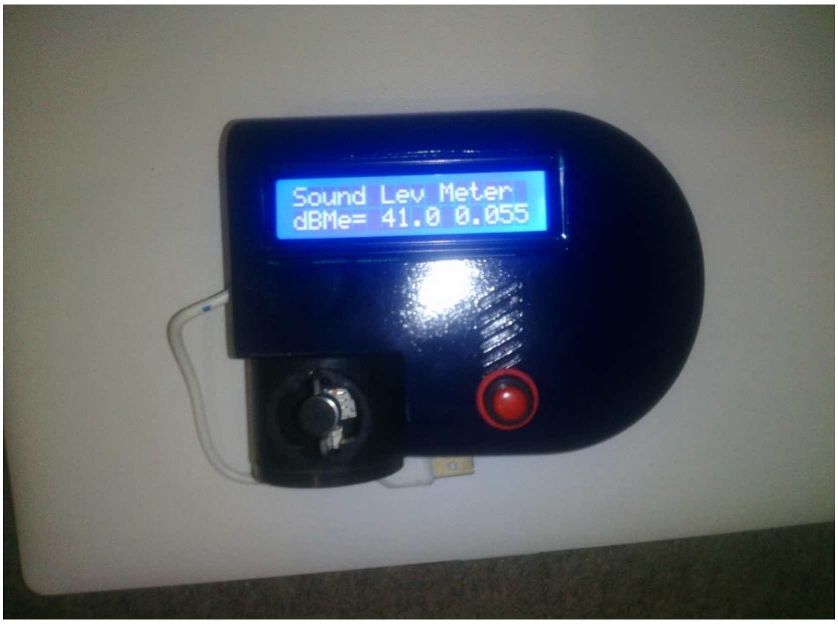
Di luar *baby incubator*



Di dalam baby incubator

Foto alat tampak dari berbagai macam sisi





Sound Lev Meter
dBMe= 41.0 0.055

LISTING PROGRAM

```
/*****************************************************
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    : 23/07/2016
Author  : user
Company :
Comments:


Chip type               : ATmega8
Program type            : Application
AVR Core Clock frequency: 12,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
```

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

#include <mega8.h>
#include <delay.h>
#include <stdlib.h>
float data,teg,db;
unsigned int adc;
unsigned char temp[10],i=0;
// Alphanumeric LCD functions
#include <alcd.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
void konversi_data()
{
 if(teg>=0.15&&teg<0.18)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.18&&teg<0.21)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.21&&teg<0.24)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.24&&teg<0.28)
 {
  db=(25*teg)+37;
 }
 else if(teg>=0.28&&teg<0.31)
 {
  db=(33.33*teg)+34.66;
 }
```

```
}

else if(teg>=0.31&&teg<0.35)
{
 db=(25*teg)+37.25;
}
else if(teg>=0.35&&teg<0.38)
{
 db=(33.33*teg)+34.33;
}
else if(teg>=0.38&&teg<0.41)
{
 db=(33.3*teg)+34.33;
}
else if(teg>=0.41&&teg<0.56)
{
 db=(6.666*teg)+45.26;
}
else if(teg>=0.56&&teg<0.6)
{
 db=(25*teg)+25;
}
else if(teg>=0.6&&teg<0.71)
{
 db=(9.090*teg)+44.54;
}
else if(teg>=0.71&&teg<0.88)
```

```
{
 db=(5.882*teg)+46.82;
}
else if(teg>=0.88&&teg<0.94)
{
 db=(16.66*teg)+37.33;
}
else if(teg>=0.94&&teg<1.28)
{
 db=(2.941*teg)+50.23;
}
else if(teg>=1.28&&teg<1.34)
{
 db=(16.66*teg)+32.66;
}
else if(teg>=1.34&&teg<1.5)
{
 db=(6.25*teg)+46.62;
}
else if(teg>=1.5&&teg<1.6)
{
 db=(10*teg)+41;
}
else if(teg>=1.6&&teg<1.66)
{
```

```
{
 db=(16.66*teg)+30.33;
}
else if(teg>=1.66&&teg<1.94)
{
 db=(3.571*teg)+52.07;
}
else if(teg>=1.94&&teg<2.47)
{
 db=(3.030*teg)+53.12;
}
 else if(teg>=2.47&&teg<2.67)
{
 db=(5*teg)+48.65;
}
else if(teg>=2.67&&teg<2.86)
{
 db=(5.263*teg)+47.94;
}
 else if(teg>=2.86&&teg<3.06)
{
 db=(5*teg)+48.7;
 }
 else if(teg>=3.06&&teg<3.26)
{
db=(7*teg)+48.7;
```

```
 }
 else if(teg>=3.26&&teg<3.40)
{
 db=(7.142*teg)+41.71;
 }
 else if(teg>=3.40&&teg<3.60)
{
 db=(5*teg)+49;
 }
 else if(teg>=3.60&&teg<3.69)
{
 db=(11.11*teg)+27;
 }
 else if(teg>=3.69&&teg<3.78)
{
 db=(11.11*teg)+27;
 }
 else if(teg>=3.78&&teg<3.94)
{
 db=(6,25*teg)+45.37;
 }
}
void main(void)
{
// Declare your local variables here
```

```
// Input/Output Ports initialization

// Port B initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out

// State7=1 State6=1 State5=1 State4=1 State3=1 State2=1 State1=1 State0=1

PORTB=0x00;

DDRB=0xFF;


// Port C initialization

// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// 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

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: 750,000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// 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 - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
```

```
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 16
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Sound Lev Meter");
lcd_gotoxy(0,1);
lcd_putsf("dBMe=");
while (1)
    {
     for(i=0;i<100;i++)
     {
        adc=read_adc(0);
        data=(float)adc*5/1024;
        teg=teg+data;
     }
     teg=teg/100;
     konversi_data();
     ftoa(db,1,temp);
     lcd_gotoxy(6,1);
     lcd_puts(temp);
     ftoa(teg,3,temp);
     lcd_gotoxy(11,1);
     lcd_puts(temp);
     delay_ms(200);
```

```
    }
}
```

Hasil perhitungan dari data **Tabel 4.3**

$\text{Simpangan } (\textbf{No. 1}) =$

$$\text{Alat ukur} - \text{Modul} = 40 - 41{,}1 = 1{,}1$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{40} \times 100\% = 2{,}75$$

$\text{Simpangan } (\textbf{No. 2}) =$

$$\text{Alat ukur} - \text{Modul} = 55 - 51{,}3 = 3{,}7$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{3{,}7}{55} \times 100\% = 6{,}72$$

$\text{Simpangan } (\textbf{No. 3}) =$

$$\text{Alat ukur} - \text{Modul} = 70 - 68{,}9 = 1{,}1$$

$$Error\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{70} \times 100\% = 1{,}57$$

Simpangan = $1,1 + 3,7 + 1,1 = 5,9$

$$= \frac{5,9}{3} = 1,9$$

Rata – rata error% = $2,75 + 6,72 + 1,51$ = 10,98

$$= \frac{10,98}{3} = 3.66$$

Simpangan = 1,9 dan Rata – rata *error%* = 3,66

Keterangan perhitungan Tabel 4.3 adalah modul alat ini dapat mengukur kebisingan di luar *baby incubator* pada *range* minimal 41,1db dan maksimal di 68,9db dengan nilai simpangan 1,9 dan rata – rara *error %* sebesar 3,66.

Tabel 4.5 Hasil perbandingan

Simpangan $(\mathbf{No.1}) =$

$$\text{Alat ukur} - \text{Modul} = 40 - 40,2 = 0,2$$

$$\mathit{Error}\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\bar{\text{X}})} \times 100\% =$$

$$\frac{0,2}{40} \times 100\% = 0,5$$

Simpangan $(\mathbf{No.2})$

$$\text{Alat ukur} - \text{Modul} = 55 - 53,0 = 2,0$$

$$\mathit{Error}\ \% = \frac{\text{Simpangan}}{\text{Alat ukur}(\bar{\text{X}})} \times 100\% =$$

$$\frac{2,0}{55} \times 100\% = 3,6$$

$\text{Simpangan}\ (\mathbf{No.3}) =$

$$\text{Alat ukur} - \text{Modul} = 70 - 68{,}9 = 1{,}1$$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\% =$$

$$\frac{1{,}1}{70} \times 100\% = 1{,}57$$

Simpangan = $0{,}20 + 2{,}00 + 1{,}10 = 3{,}3$

$$= \frac{3{,}3}{3} = 1{,}1$$

Rata – rata *error%* = $0{,}50 + 3{,}60 + 1{,}57$ = 10,48

$$= \frac{3{,}49}{3} = 1{,}00$$

Simpangan = 1,9 dan Rata – rata *error%* = 3,49

Keterangan perhitungan Tabel 4.5 adalah modul alat ini dapat mengukur kebisingan di dalam *baby incubator* pada *range* minimal 40,2db dan maksimal di 68,9db dengan nilai simpangan 1,9 dan rata – rara *error %* sebesar 3,49

Hasil perhitungan data kalibrasi dari **Tabel 4.6**

$$\text{Simpangan}\ (\mathbf{No.1}) = \text{Alat ukur} - \text{Modul} = 52 - 48.9 = 3{,}1$$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{3{,}1}{52} \times 100\%$$

$$= 5{,}96\%$$

$\text{Simpangan } (\mathbf{No.\,2}) = \text{Alat ukur } - \text{ Modul} = 53 - 51{,}0 = 2$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{2}{53} \times 100\%$$

$= 3{,}77\%$

$\text{Simpangan } (\mathbf{No.\,3}) = \text{Alat ukur } - \text{ Modul} = 54 - 51{,}9 = 2{,}1$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{2{,}1}{54} \times 100\%$$

$= 3{,}88\%$

$\text{Simpangan } (\mathbf{No.\,4}) = \text{Alat ukur } - \text{ Modul} = 55 - 53{,}8 = 1{,}2$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{1{,}2}{55} \times 100\%$$

$= 2{,}18\%$

$\text{Simpangan } (\mathbf{No.\,5}) = \text{Alat ukur } - \text{ Modul} = 56 - 54{,}9 = 1{,}1$

$$Error\ \% \quad = \frac{\text{Simpangan}}{\text{Alat ukur}(\overline{\text{X}})} \times 100\%$$

$$= \frac{1{,}1}{56} \times 100\% = 1{,}9\%$$

Total simpangan = $3{,}1 + 2 + 2{,}1 + 1{,}2 + 1{,}1 = 9{,}5$

$$= \frac{9{,}5}{5} = 1{,}9$$

Total rata-rata *error* = $5{,}96 + 3{,}77 + 3{,}88 + 2{,}18 + 1{,}9 = 17{,}69$

$$\frac{17{,}69}{5} = 3{,}538$$

Rata-rata *error%* = 3,538 %dan simpangan = 1,9 modulsensor kebisingan di dalam *baby incubator*.

Foto proses pembuatan

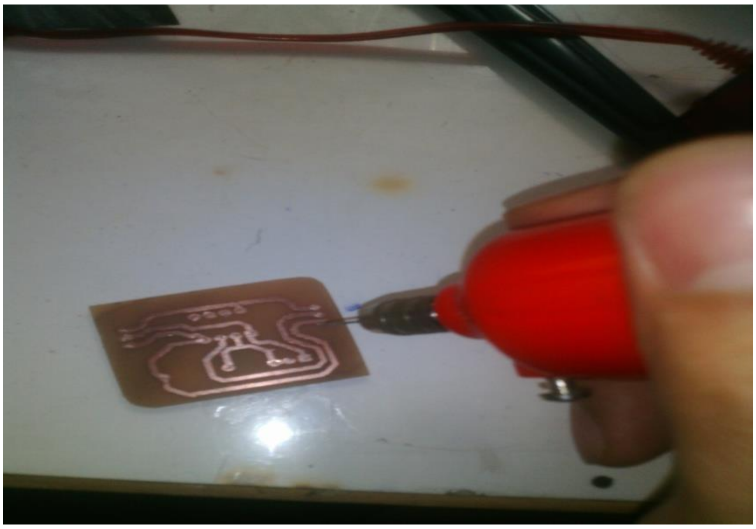
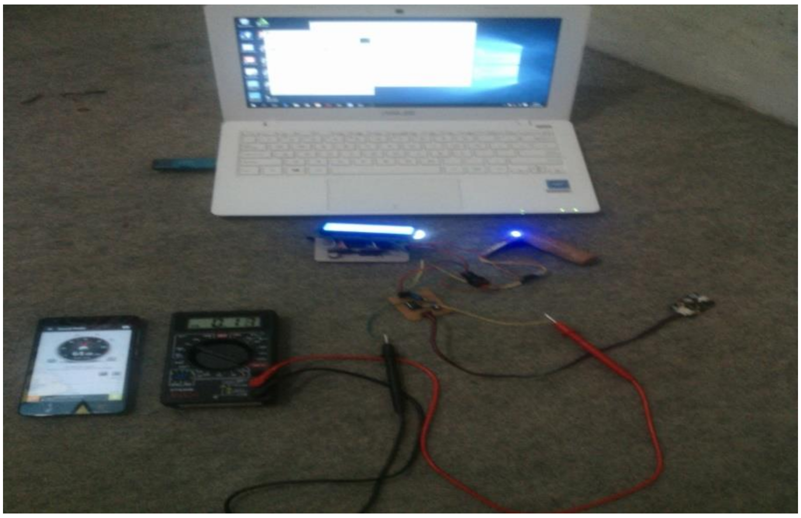


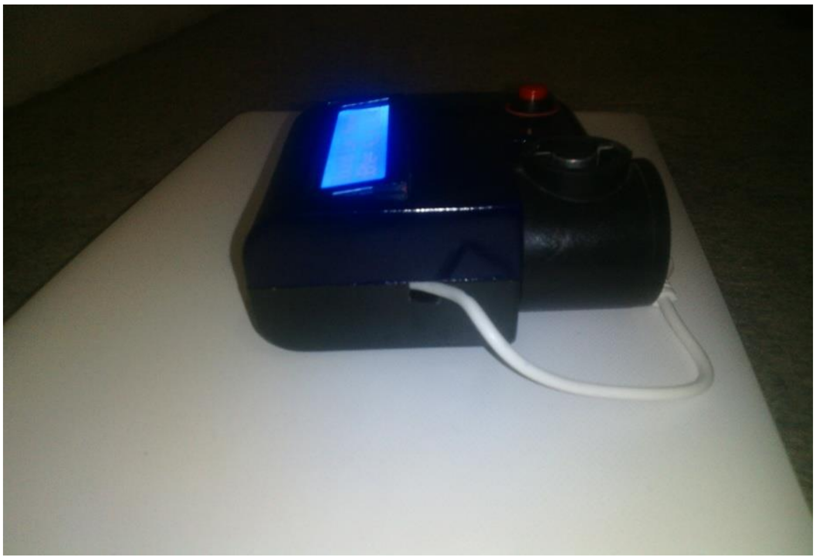
Foto proses pengambilan data

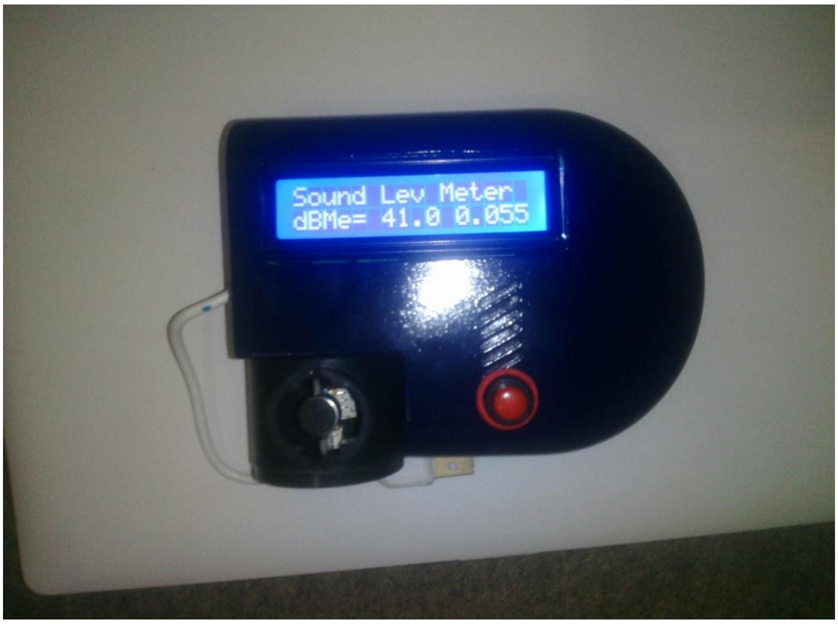
Di luar *baby incubator*



Di dalam baby incubator

Foto alat tampak dari berbagai macam sisi




Sound Lev Meter
dBMe= 41.0 0.055

LISTING PROGRAM

```
/*****************************************************
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    : 23/07/2016
Author  : user
Company :
Comments:


Chip type               : ATmega8
Program type            : Application
AVR Core Clock frequency: 12,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
```

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

#include <mega8.h>
#include <delay.h>
#include <stdlib.h>
float data,teg,db;
unsigned int adc;
unsigned char temp[10],i=0;
// Alphanumeric LCD functions
#include <alcd.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
void konversi_data()
{
 if(teg>=0.15&&teg<0.18)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.18&&teg<0.21)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.21&&teg<0.24)
 {
  db=(33.33*teg)+35;
 }
 else if(teg>=0.24&&teg<0.28)
 {
  db=(25*teg)+37;
 }
 else if(teg>=0.28&&teg<0.31)
 {
  db=(33.33*teg)+34.66;
 }
```

```
else if(teg>=0.31&&teg<0.35)
{
 db=(25*teg)+37.25;
}
else if(teg>=0.35&&teg<0.38)
{
 db=(33.33*teg)+34.33;
}
else if(teg>=0.38&&teg<0.41)
{
 db=(33.3*teg)+34.33;
}
else if(teg>=0.41&&teg<0.56)
{
 db=(6.666*teg)+45.26;
}
else if(teg>=0.56&&teg<0.6)
{
 db=(25*teg)+25;
}
else if(teg>=0.6&&teg<0.71)
{
 db=(9.090*teg)+44.54;
}
else if(teg>=0.71&&teg<0.88)
```

```
{
 db=(5.882*teg)+46.82;
}
else if(teg>=0.88&&teg<0.94)
{
 db=(16.66*teg)+37.33;
}
else if(teg>=0.94&&teg<1.28)
{
 db=(2.941*teg)+50.23;
}
else if(teg>=1.28&&teg<1.34)
{
 db=(16.66*teg)+32.66;
}
else if(teg>=1.34&&teg<1.5)
{
 db=(6.25*teg)+46.62;
}
else if(teg>=1.5&&teg<1.6)
{
 db=(10*teg)+41;
}
else if(teg>=1.6&&teg<1.66)
{
```

```
{
 db=(16.66*teg)+30.33;
}
else if(teg>=1.66&&teg<1.94)
{
 db=(3.571*teg)+52.07;
}
else if(teg>=1.94&&teg<2.47)
{
 db=(3.030*teg)+53.12;
}
 else if(teg>=2.47&&teg<2.67)
{
 db=(5*teg)+48.65;
}
else if(teg>=2.67&&teg<2.86)
{
 db=(5.263*teg)+47.94;
}
 else if(teg>=2.86&&teg<3.06)
{
 db=(5*teg)+48.7;
 }
 else if(teg>=3.06&&teg<3.26)
{
db=(7*teg)+48.7;
```

```
 }
 else if(teg>=3.26&&teg<3.40)
{
 db=(7.142*teg)+41.71;
 }
 else if(teg>=3.40&&teg<3.60)
{
 db=(5*teg)+49;
 }
 else if(teg>=3.60&&teg<3.69)
{
 db=(11.11*teg)+27;
 }
 else if(teg>=3.69&&teg<3.78)
{
 db=(11.11*teg)+27;
 }
 else if(teg>=3.78&&teg<3.94)
{
 db=(6,25*teg)+45.37;
 }
}
void main(void)
{
// Declare your local variables here
```

```
// Input/Output Ports initialization
// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=1 State6=1 State5=1 State4=1 State3=1 State2=1 State1=1 State0=1
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// 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
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: 750,000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// 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 - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
```

```
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 16
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Sound Lev Meter");
lcd_gotoxy(0,1);
lcd_putsf("dBMe=");
while (1)
    {
     for(i=0;i<100;i++)
     {
        adc=read_adc(0);
        data=(float)adc*5/1024;
        teg=teg+data;
     }
     teg=teg/100;
     konversi_data();
     ftoa(db,1,temp);
     lcd_gotoxy(6,1);
     lcd_puts(temp);
     ftoa(teg,3,temp);
     lcd_gotoxy(11,1);
     lcd_puts(temp);
     delay_ms(200);
```

```
    }
}
```