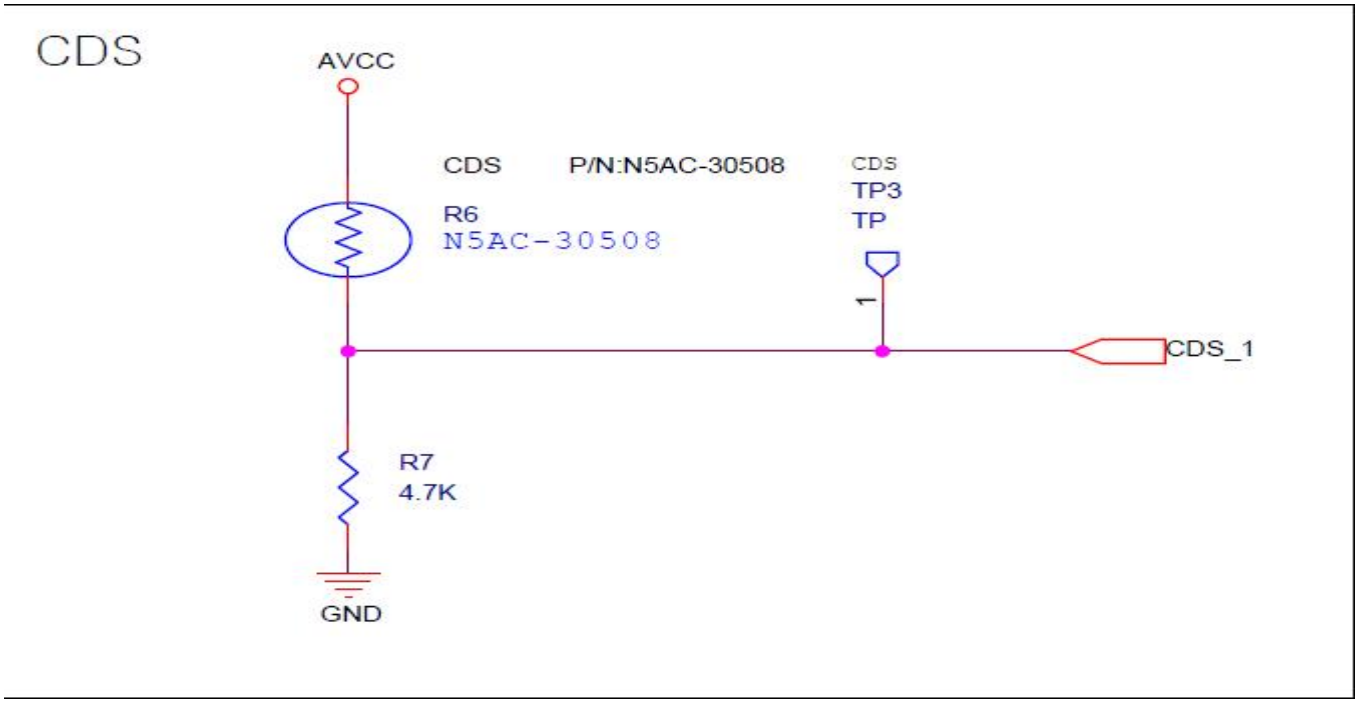
로봇학실험3 과제보고서

2014741022 서덕현

- **CDS 측정**



1. CDS에 나오는 전압값(CDS_1)을 구할려면 회로이론에서 배웠던 이론을 쓰면 **CDS_1 = R6/(R6+R7) x AVCC**이므로 R6, 즉 CDS의 저항은 **R6 = (R7 X AVCC)/CDS_1 – R7** 이다.

Parameter	Symbol	Min	Max	Units
Light Resistance at 10 Lux	RL	30	50	kΩ
Gamma Value at 10~100 Lux		0.8(Typ)		
Dark Resistance (10 sec. after shut off 10 Lux)	RD	5		MΩ
Peak Spectral Response		550	650	nm

2. 이 표는 CDS의 데이터시트다. 이 표를 간략하게 해석하자면 첫 번째 칸은 10Lux의 빛의 밝기일 때 최소 30kΩ에서 최대 50kΩ의 저항을 가진다는 뜻이다. 두 번째 칸은 10~100Lux사이의 조도지수는 항상 0.8이라는 것이다.

3. 조도지수를 구하는 공식을 알면 조도센서의 값을 알 수 있다.

• 광도전 셀의 전압–전류 관계

$$I = \alpha V^{\beta} L^{\gamma}$$

- I : 전류[A], V : 전압[V], L : 조도[Lx]
- α : 상수
- β : 전압지수, 인가전압 수~ 수십[v]에서 거의 1
- γ : 조도지수, 조도 및 센서 종류에 따라 다름. 0.5~1의 범위. 저조도에서 크고 고조도로 갈수록 작아져 거의 0.5가 된다.

4. 사실상 저 공식은 **I = α x V x L^r**이다.

• 조도 지수

- 조도 L_a, L_b에서의 저항을 R_a, R_b라 하면 - 양변에 log를 취하고 조도지수에 대해 정리하면

$$R_a = \frac{V}{I} = \alpha^{-1} {L_a}^{-\gamma}$$

$$R_b = \frac{V}{I} = \alpha^{-1} {L_b}^{-\gamma}$$

$$\gamma = log(\frac{R_a}{R_b}) / log(\frac{L_b}{L_a})$$

$$\therefore \gamma = \frac{log(R_a) - log(R_b)}{log(L_b) - log(L_a)}$$

5. 앞에서 구했던 **I = α x V x L^r** 를 저항(R)를 구하는 공식인 R = V/I를 이용하면 **R = α^-1 x L^-r** 이다. 그럼 여기서 조도 값인 La과 Lb 가 있다고 가정하면 La의 저항(Ra)은 **Ra = α^-1 x La^-r** , Lb의 저항(Rb)은 **Rb = α^-1 x Lb^-r** 이다. 두 값을 로그로 취해주면 **log(Ra) = - log(α) – rlog(La), log(Rb) = -log(α) – rlog(Lb)**이다. 여기서 **log(α) = -log(Rb) – rlog(Lb) = - log(Ra) – rlog(La)**로 도출할 수 있다. 그러면 **rlog(Lb/La) = log(Ra/Rb)**이므로 조도지수인 r은 **r = (log(Ra) – log(Rb))/(log(Lb) – log(La))** 이다. 이걸 정리한게 위의 사진이다.

6. La를 현재 조도센서값(x)으로 잡고 Lb를 10Lux 조도값으로 잡으면 Ra = R6 =Rcds 즉 CDS의 저항값이고 Rb=R7를 10Lux일 때의 저항의 평 균값인 (30+50)/2 = 40kΩ으로 잡아서 취급한다. 그럼 아래 식처럼 정리가 된다.

$$\gamma = \frac{log(R_{cds}\ [\Omega]) - log(40\ [k\Omega])}{log(10\ [Lux]) - log(x\ [Lux])}$$

7. 조도지수는 앞에서 0.8로 취급하기로 했으니 이제 구해야 할 변수는 현재 조도센서의 값인 x다. 앞에서 구한 조도지수 공식을 정리하면

$$log(x\ [Lux]) = 1 - \frac{log(R_{cds}\ [\Omega]) - log(40\ [k\Omega])}{\gamma}$$

이 식이 나오는데 여기서 로그값을 빼면 **x = 10^{1-(log(Rcds)-log(40k))/r}** 즉 아래 식처럼 나온다.

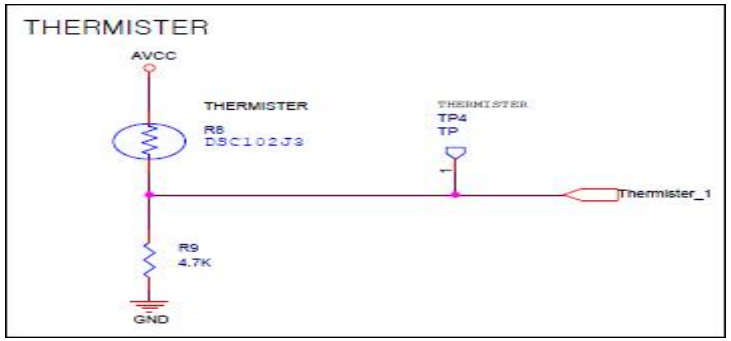
$$\therefore x\ [Lux] = 10^{1-\frac{log(R_{cds}\ [\Omega]) - log(40\ [k\Omega])}{\gamma}}$$

8. 이제 소스에 넣어야 공식들을 정리하자면

1) Rcds = (R7 X AVCC)/CDS_1 – R7 (R7 = 4.7k, AVCC = 5V, CDS_1 =

Vout) 2) Lux = 10^{1-(log(Rcds)-log(40k))/r}

- **Thermister**

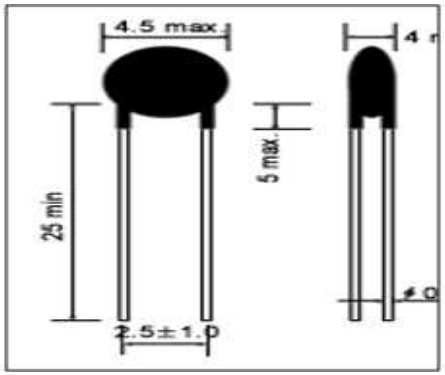


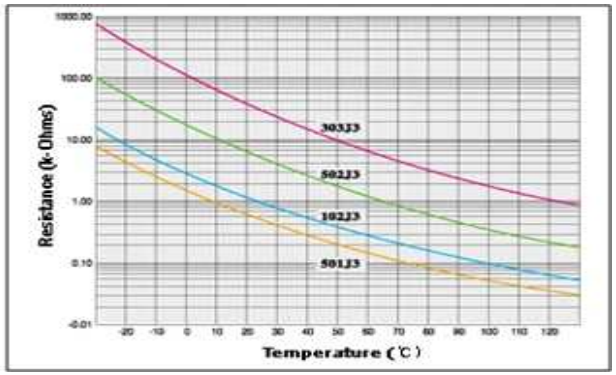
1. Thermister의 전압값은 CDS에서 말한 회로이론에서 배웠던 공식을 써서 **Thermister_1 = R8/(R8+R9) x AVCC**이다. 그럼 Thermister의 저항값 은 R8은 **R8 = (AVCC/Thermister_1) x R9 –R9** 이다.

- **DSC102J3**

  → 옴의 온도계수(NTC) 저항체

  - $T_0 = 25 + 273.15\ [℃]$
  - $R_0 = 1000\ [\Omega]$, at $T = T_0$
  - $\beta = 3620$, at $T = T_0$





2. Thermister는 절대온도를 표현하기 때문에 상온 온도에서 273.15도에서 더해준다.
3. 위의 사진은 상온 온도 25도에서 저항값과 베타 값을 나타내는 것이다.

$$R_{th} = R_0 e^{\beta\left(\frac{1}{T}-\frac{1}{T_0}\right)}$$

4. Thermistor의 저항값인 Rth을 상온온도 25도에서 저항값과 베타값을 표현한 공식이다. 이 공식을 로그(ln)를 곱해주면 아래식처럼 표현된다.

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta}\ln\left(\frac{R_{th}}{R_0}\right)$$

5. T는 현재 온도값이고 T0는 25도고 베타는 앞에서 언급한 3620이고 Rth는 Thermistor의 저항이고 R0는 25도에서의 저항인 1000Ω이다.
6. T를 구하는 방법은 위의 식을 역수로 취하면 된다. 아래 식처럼 표현이 된다.

$$T\ [K] = {}^{1}\Big/{\frac{1}{T_0} + \frac{1}{\beta}\ln\left(\frac{R_{th}}{R_0}\right)}$$

7. 현재 T는 절대온도인 273.15로 더해져 있는 T라 절대온도를 빼서 우리가 쓰는 온도 수치로 바꿔준다. Tc = T - 273.15,즉 아래식처럼 표현된다.

$$\therefore T\,[℃] = T\,[K] - 273.5\,[℃]$$

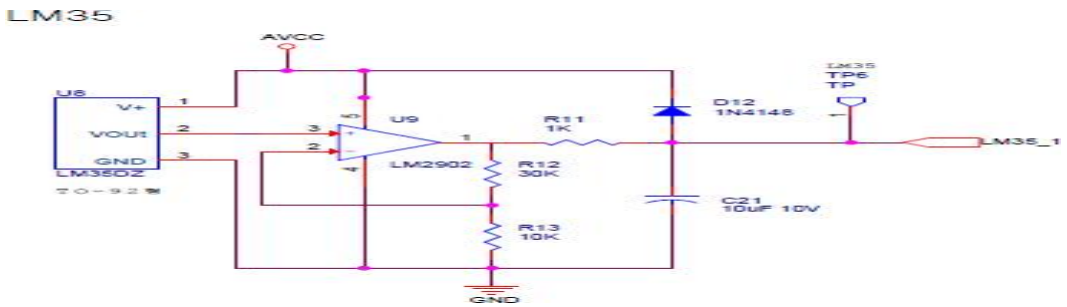
8. 소스에 넣어야 할 공식들은

1) R8 = (AVCC/Thermister_1) x R9 –R9 (R8=Rth, AVCC = 5V, Thermister_1 = Vout, R9 = 4.7kΩ) 2) 1/T = 1/T0 + 1/β x (ln(Rth/R0))

3) T[K] = 1/T

4) T[C] = T[K] – 273.15

- **LM35**



1. LM35의 전압인 LM35_1를 VLm35로 취급해서 Vout와의 관계는 **VLm35 = Vout/4** 이다.
2. 온도인 T와 VLm35와의 관계는 **T = 100 x VLm35 = 100 x Vout/4 = 25 x Vout** 이다.
3. 위의 두 식을 소스에 넣어야 할 공식이다

- **소스설명(레지스터 설정은 저번 과제랑 똑같아서 안 넣었습니다)**

```
//////////////////// Define ////////////////////////////
#define      sbi(PORTX,BitX) PORTX|=(1<<BitX)   //비트 set    명령 정의
#define      cbi(PORTX,BitX) PORTX&=~(1<<BitX)   //비트 clear 명령 정의
#define      K 1000.0      //단위 킬로
#define      bit10 1023.0  // 2의 10승
#define      Vcc 5.0   //5V
#define      Tk  273.15 //절대온도
```

1. 10^3를 소스에 일일이 붙이기 귀찮으므로 10^3를 줄이는 단위는 k를 #define으로 정의해준다.

2. ADC의 값은 0에서 2^10 – 1한 값이 1023까지이므로 이 최댓값을 쓸 일이 센서값을 전압을 구할려면 ADC/1023 x Vcc(5V)로 구해야하므로 애 도 일일이 치기 귀찮고 어떤 의미인지 알려주기 위해 bit10로 써서 #define으로 정의해준다.

3. 5V도 그냥 5만 쓰면 무슨 의미인지를 모를 수 있으니 Vcc로 정의해서 5V라는 것을 알려준다.

4. Tk는 절대온도를 정의한 것이다. Thermistor는 절대 온도로 표현되기 때문에 그걸 빼준 값을 나타내기 위해 정의한 것이다.

5. 여기서 숫자들 뒤에 .0(점오) 즉 소수점을 나타낸 이유는 뒤에서 변수 선언할 때 실수형으로 선언했기 때문이다. 소수점을 안 하면 실수형 변 수가 잘못하면 정수형 변수로 형변환이 될 수 있기 때문이다.

```
////////// Local Var ////////////////////////
//unsigned int a_ADC_data[4]={0,};
    float adc_V[2]={0,};// 센서값 전압
    float Rth; //Thermistor 저항
    float T1,T2; //Thermistor 변수
    float Tlm35; //LM35 변수
    float Rcds; //CDS의 저항
    float Lux; //CDS 빛의 밝기
    float x; // 10의 제곱 부분 변수

////////// CDS /////////////////
Rcds = (4.7*K*5.0)/adc_V[0]-4.7*K;
x= 1-(log10(Rcds)-log10(40.0*K))/0.8;
Lux = pow(10,x);

/////// Uart Send ////////////////////////
      m_send_period++;
      m_send_period%=50;

      if(m_send_period==0)//500ms
     {
      mcu_init.Uart_Tempnum(Tlm35*10);   mcu_init.Uart_Putch(',');//데이터구분
      mcu_init.Uart_Tempnum(T2*10);  mcu_init.Uart_Putch(',');//데이터구분
      mcu_init.Uart_num(Lux);
      mcu_init.Uart_Putch(0x0D);//ENTER

     }
```

1. 앞에서 언급한 조도센서의 저항인 Rcds 와 현 Lux값을 실수형으로 선언한다. 실수형으로 선언한 이유는 저항이 정확하게 정수로 안 나오고 실수로 나오기 때문에 선언했고 저항 변수의 실수형을 형변환 안 시킬려고 Lux값도 실수형으로 선언했다.

    ㄴ `float Rcds; float Lux;`

2.센서의 전압을 알려면 앞에서 언급한 **ADC/1023 x Vcc(5V)**로 나타낸다

    ㄴ `for(int i=0;i<3;i++)`
    `adc_V[i]=(mcu_init.ADC_Read(i+1)/bit10)*Vcc;`
    ㄴ `adc_V[0]`는 CDS의 전압, `adc_V[1]`는 LM35의 전압, `adc_V[2]`는 Thermistor의 전압을 나타낸다.

1. Rcds 값은 앞에서 언급한 것처럼 **Rcds = (R7 X AVCC)/CDS_1 – R7 (R7 = 4.7k, AVCC = 5V, CDS_1 = Vout)**로 풀어준다.

    ㄴ `Rcds = (4.7*K*Vcc)/adc_V[0]-4.7*K;`

2. `float x;`를 선언한 이유는 앞에 식인 **1-(log(Rcds)-log(40k))/r**를 표현하기 복잡하므로 간단하게 나타내기 위해 x를 선언했다.

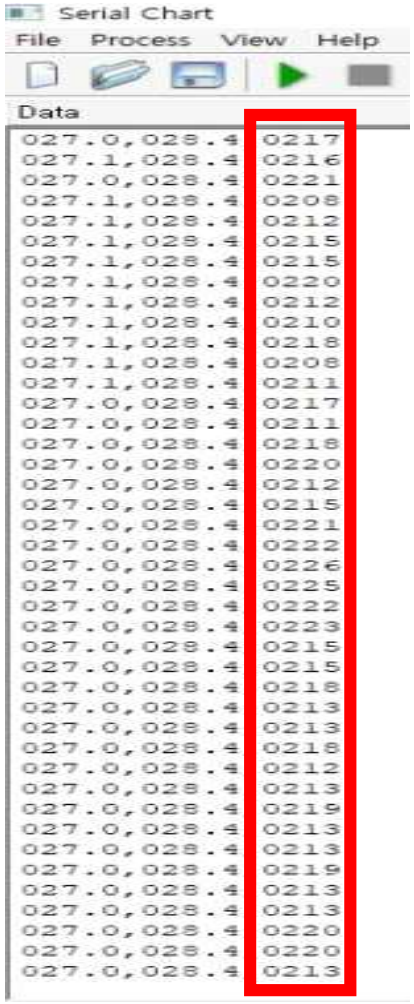
    ㄴ `x= 1-(log10(Rcds)-log10(40.0*K))/0.8;`

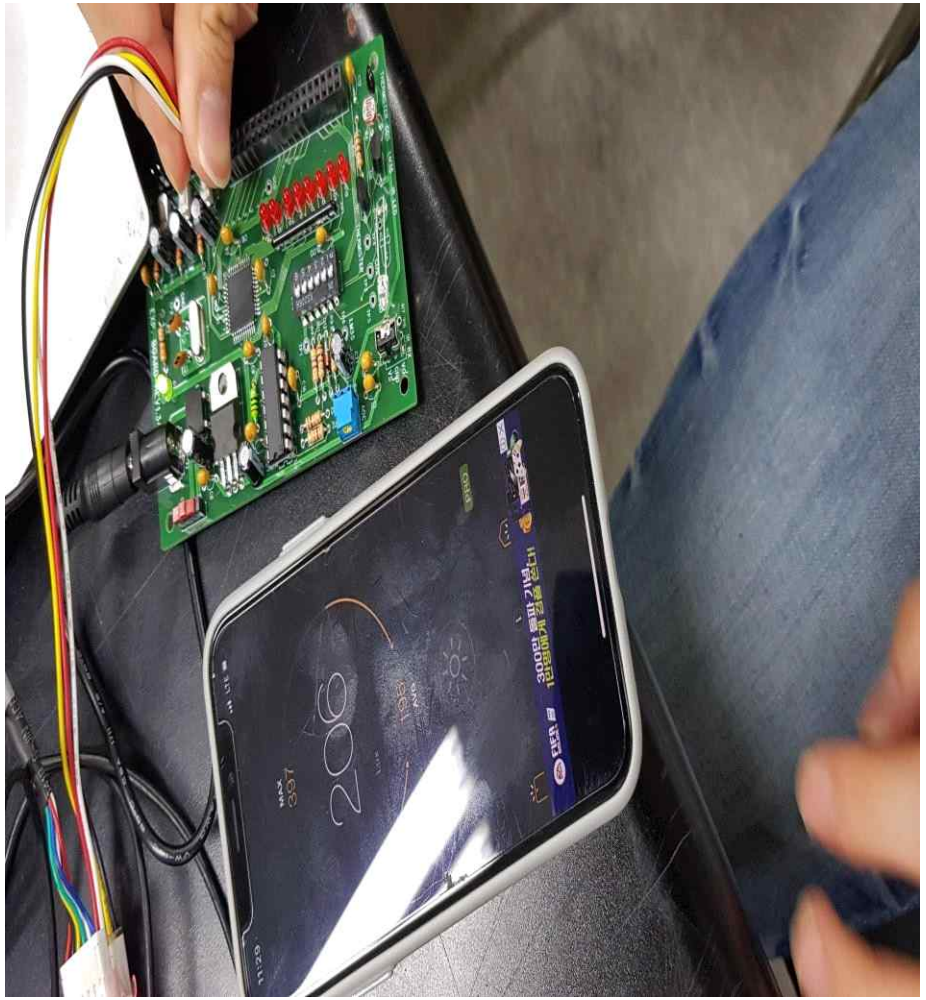
```
#include <math.h>
```

3. **Lux = 10^{1-(log(Rcds)-log(40k))/r}** 의 값을 10의 제곱으로 나타낼려면 먼저 헤더파일 중 math.h를 #include 쪽에 선언해줘야한다. 그리고 pow(10,x)로 쓰면 앞에 공식으로 취급된다.

    ㄴ `Lux = pow(10,x);`

4. 지난 주 과제에서 값을 보내주는 함수인 Uart_num를 Lux에 넣어줘서 시리얼통신으로 값을 띄운다.





5. 시리얼 통신에서 빨간 색 박스 안에 있는 값들이 조도 값들이다. 핸드폰에서 측정한 조도값과 좀 차이가 있지만 이건 부품 및 개발자가 한 수치변환이 우리랑 다를 수 있으므로 작은 차이가 있을 수 있다.

```
        /////////// Local Var ////////////////////////
        //unsigned int a_ADC_data[4]={0,};
            float adc_V[2]={0,};// 센서값 전압
            float Rth; //Thermistor 저항
            float T1,T2; //Thermistor 변수
            float Tlm35; //LM35 변수
            float Rcds; //CDS의 저항
            float Lux; //CDS 빛의 밝기
            float x; // 10의 제곱 부분 변수
        /////////// ADC Read ////////////////////////////
            for(int i=0;i<3;i++)
               adc_V[i]=(mcu_init.ADC_Read(i+1)/bit10)*Vcc;

       /////////// Thermistor ////////////////////
        Rth = (Vcc/adc_V[2])*(4.7*K)-4.7*K;
        T1 = 1/(25+Tk)+log(Rth/K)/3620 ;
        T2 = 1/T1 -Tk;

        /////////// LM35 ////////////////////
        Tlm35 = 100/4*adc_V[1];

void MCU_Init::Uart_Tempnum(int data)
 {
     Uart_Putch((data/1000)+48);
     Uart_Putch((data%1000)/100+48);
     Uart_Putch((data%100)/10+48);
     Uart_Putch('.');
     Uart_Putch((data%10)+48);
 }
////// Uart Send ////////////////////////////
     m_send_period++;
     m_send_period%=50;

     if(m_send_period==0)//500ms
    {
     mcu_init.Uart_Tempnum(Tlm35*10);   mcu_init.Uart_Putch(',');//데이터구분
     mcu_init.Uart_Tempnum(T2*10);  mcu_init.Uart_Putch(',');//데이터구분
     mcu_init.Uart_num(Lux);
     mcu_init.Uart_Putch(0x0D);//ENTER

    }
```

6. 온도를 소수점도 표현하기 위해 실수형으로 선언했다.

7. 앞에서 구한 Thermistor의 공식을 소스에 넣는 부분이다.

1) Rth = (5V/Vout) x 4.7k –4.7k

ㄴ `Rth = (Vcc/adc_V[2])*(4.7*K)-4.7*K;` 로 구현

2) 1/T = 1/T0 + 1/β x (ln(Rth/R0))

3) T[K] = 1/T

ㄴ 2번 3번 식을 `T1 = 1/(25+Tk)+log(Rth/K)/3620;` 로 구현

4) T[C] = T[K] – 273.15

ㄴ `T2 = 1/T1 -Tk;` 로 구현

8. 앞에서 구한

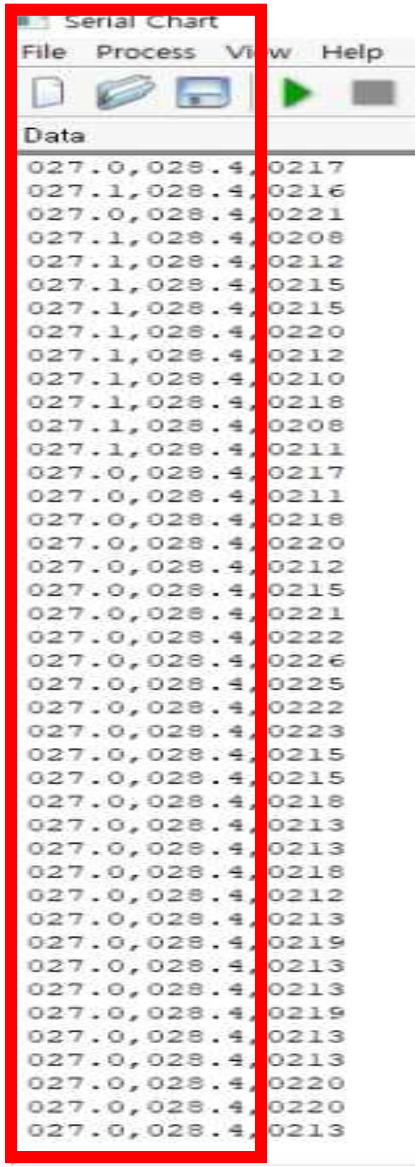
LM35 의 공식을 소스에 넣는 부분이다

1)VLm35 = Vout/4

2)T = 25 x Vout

ㄴ 1,2번 식을 `Tlm35 =100/4*adc_V[1];`

9. 값을 보낼 때 곱하기 10을 해주었는데 해준 이유가 현재 우리는 값을 보낼 때 값을 나눈 뒤 아스키코드에서 0에 해당되는 48을 더해줘서 보냈는데 소수점을 나타내는 . 를 데이터로 안 보내주었고 그 소수점 아래 숫자를 인식을 못하기 때문에 곱하기 10을 한 뒤 Uart.Tempnum 이라는 함수를 만들어서 1의 자리 앞에 `Uart_Putch('.');` 를 만들어서 소수점을 표현해주는 값을 보내주고 1의자리를 나누면 시리얼 통신에서는 마치 소수점을 표현한 것처럼 나온다.



10. 빨간 박스 안이 온도센서 수치화한 값들이다. 첫번 째가 LM35고 두번 째는 Thermistor다. 이론대로라면 두 개의 온도센서의 값이 똑같이 나와야겠지만 부품마다 차이가 있어 작은 차이가 있을 수 있다.

- 전체소스

```
//////////////////// Include ////////////////////////////
#include <avr/io.h>
#include <avr/interrupt.h>
#include <math.h>
#include "MCU_Init.h"

//////////////////// Define /////////////////////////////
#define      sbi(PORTX,BitX) PORTX|=(1<<BitX)   //비트 set    명령 정의
#define      cbi(PORTX,BitX) PORTX&=~(1<<BitX)   //비트 clear 명령 정의
#define      K 1000.0      //단위 킬로
#define      bit10 1023.0  // 2의 10승
#define      Vcc 5.0    //5V
#define      Tk  273.15 //절대온도

//////////////////// Global Var /////////////////////////////
MCU_Init mcu_init;
static int m_send_period=0;
 ////////////////// Interrupt ////////////////////////////
SIGNAL(TIMER2_OVF_vect)
{
    ////////// Timer Start ////////////////////////////
    sbi(PORTD,PORTD7);
    cli();

    ////////// Local Var ////////////////////////
    //unsigned int a_ADC_data[4]={0,};
        float adc_V[2]={0,};// 센서값 전압
        float Rth; //Thermistor 저항
        float T1,T2; //Thermistor 변수
        float Tlm36; //LM36 변수
        float Rcds; //CDS의 저항
        float Lux; //CDS 빛의 밝기
        float x; // 10의 제곱 보조 변수
    ////////// ADC Read ////////////////////////////
        for(int i=0;i<3;i++)
           adc_V[i]=(mcu_init.ADC_Read(i+1)/bit10)*Vcc;

   ////////// Thermistor //////////////////
    Rth = (Vcc/adc_V[2])*(4.7*K)-4.7*K;
    T1 = 1/(25+Tk)+log(Rth/K)/3620 ;
    T2 = 1/T1 -Tk;

    ////////// LM36 //////////////////
    Tlm36 = 100/4*adc_V[1];

    ////////// CDS //////////////////
    Rcds = (4.7*K*5.0)/adc_V[0]-4.7*K;
    x= 1-(log10(Rcds)-log10(40.0*K))/0.8;
    Lux = pow(10,x);

    /////// Uart Send ////////////////////////////
           m_send_period++;
           m_send_period%=50;

           if(m_send_period==0)//500ms
          {
           mcu_init.Uart_Tempnum(Tlm36*10);   mcu_init.Uart_Putch(',');//데이터구분
           mcu_init.Uart_Tempnum(T2*10);  mcu_init.Uart_Putch(',');//데이터구분
           mcu_init.Uart_num(Lux);
           mcu_init.Uart_Putch(0x0D);//ENTER

          }

    //// Control Period Set ////////////////////////////
           TCNT2=100;
    //// Timer End ////////////////////////////
           cbi(PORTD,PORTD7);
           sei();
}
```

```
//////////////////// Main ////////////////////////////
int main(void)
{

//////////////////// Register Init ////////////////////////////
    DDRC=0xff;
    PORTC=0xff;
    mcu_init.Init_DDR();
    mcu_init.Init_Timer2();
    mcu_init.Init_TIMSK();
    mcu_init.Init_UART();
    mcu_init.Init_ADC();


//////////////////// Interrupt Enable ////////////////////////////
    sei();

    while(1)
    {

    }
}
```