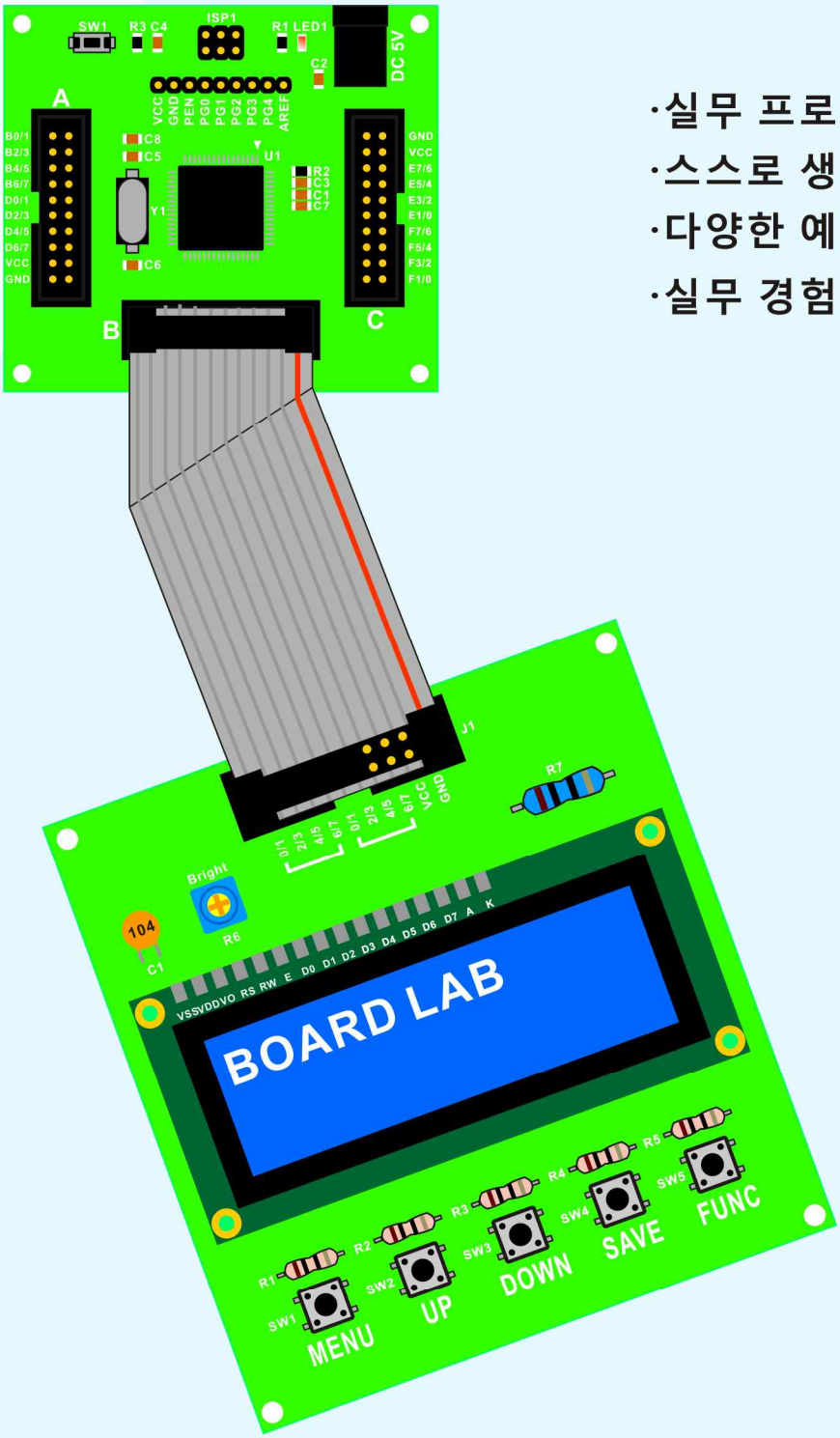
선배가 취업나와 보니까 꼭 필요하더라

전자분야 취준생을 위한 실무 프로젝트



·실무 프로젝트를 활용한 교육
·스스로 생각하는힘 형성
·다양한 예제를 통한 실력향상
·실무 경험 꿀팁 전달

www.BoardFree.kr

저자:김명수.이소리 공저 / 감수 : 김용필

차례

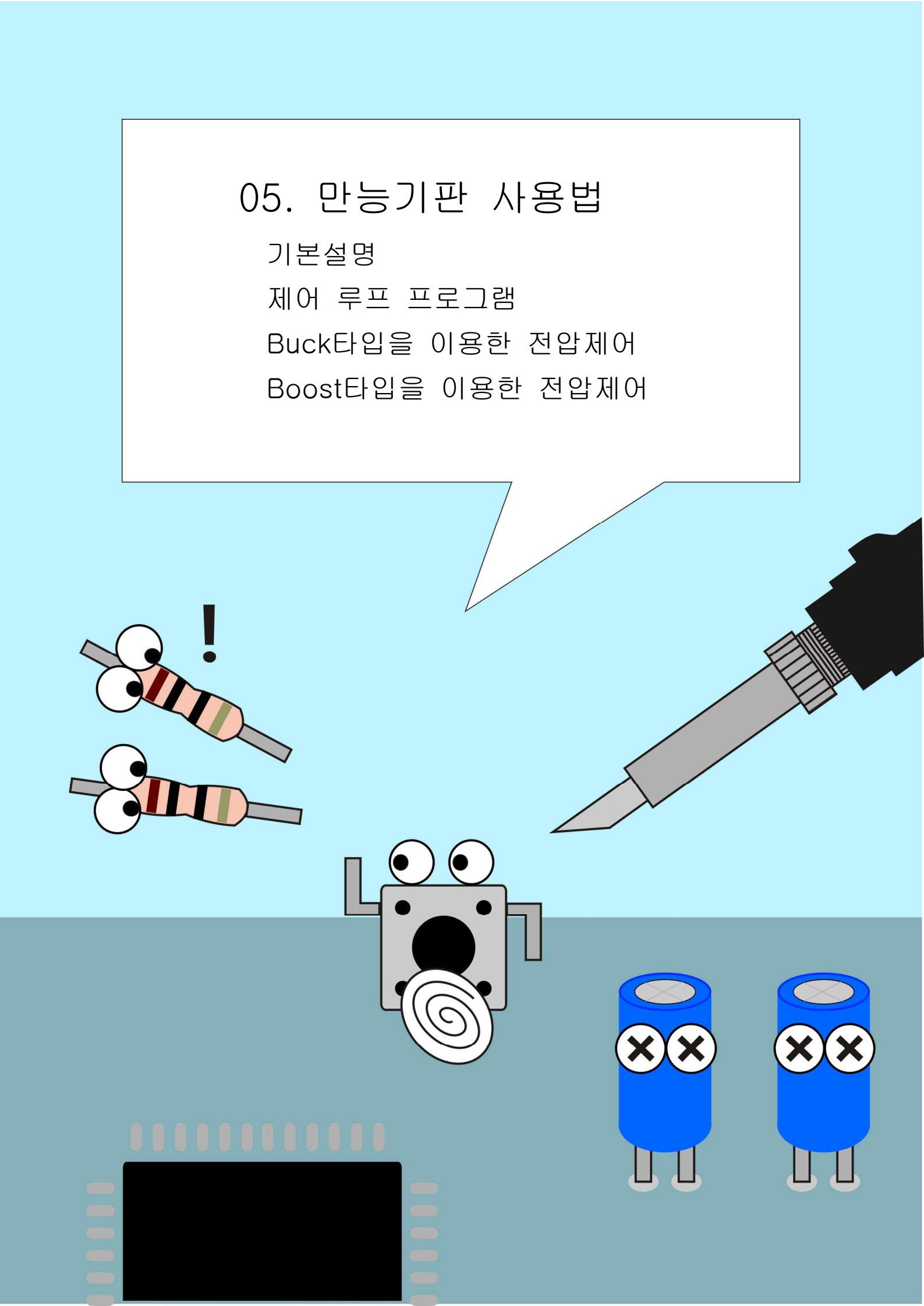
05. 만능기판 사용법

기본설명

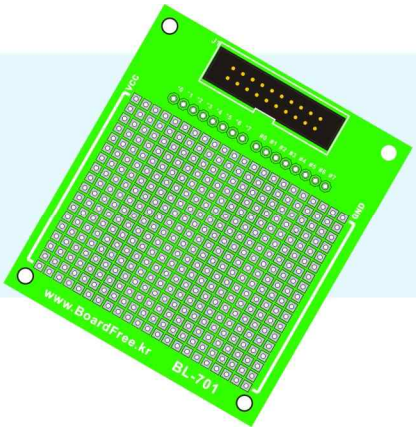
제어 루프 프로그램

Buck타입을 이용한 전압제어

Boost타입을 이용한 전압제어



1 기본설명



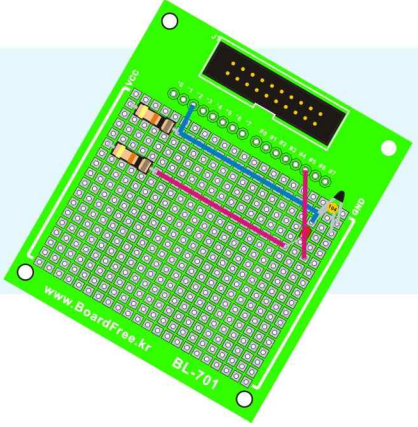
1.1 이 보드는 뭐죠?

이번 단원부터 주로 사용할 보드는 조그마한 만능기판이야. 조금 특이하게 생겼지? 패드 모양도 동그랗지 않고 네모 모양이고. CPU 보드를 포함한 배웠던 다른 보드들 하나를 만들기 위해 들어가는 비용은 생각보다 비싸. 또, 이미 만들어져있어서 마음대로 수정하기가 힘들지. 그래서 이 보드를 만들게 되었어. 패드가 네모 모양인 이유는 DIP타입 뿐만 아니라 SMD를 사용할 때도 있기 때문이고.

1.2 이번 단원에서 뭘 배운다는 거죠?

이번 단원은 너희의 프로그램적인 능력을 향상하기 위해 만들어졌어. 또, 매번 LCD, FND, LED 제어만 하니까. 그 외에 뭔가를 해볼 수 있는 게 필요하다고 생각했거든. 직접적인 프로젝트라고 할 수는 없지만. 하나하나가 실무에서 많이 사용될 수밖에 없는 내용으로 구성했어. 아마 나중에 너희가 엔지니어로서 무언가를 하게 되었을 때 도움이 많이 될 거야. 정해져 있는 답은 없으니까 열심히 해봐.

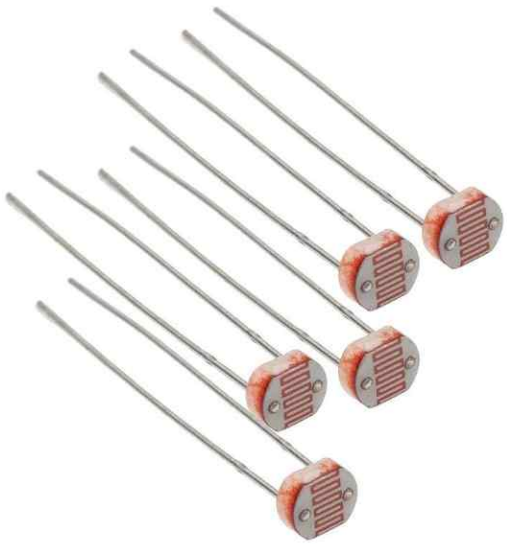
2 제어 루프 프로그램



2.1 제어 루프 프로그램?

말이 너무 어렵게 다가오지? 근데 이번 단원에서 할 설명을 읽으면 뭐가 이렇게 쉽지 라는 느낌이 들 거야. 이번 단원에서 해야 할 일은 CDS에 따른 LED 밝기를 조절할 거야. 그 전에 먼저 CDS가 뭔지 설명해줄게.

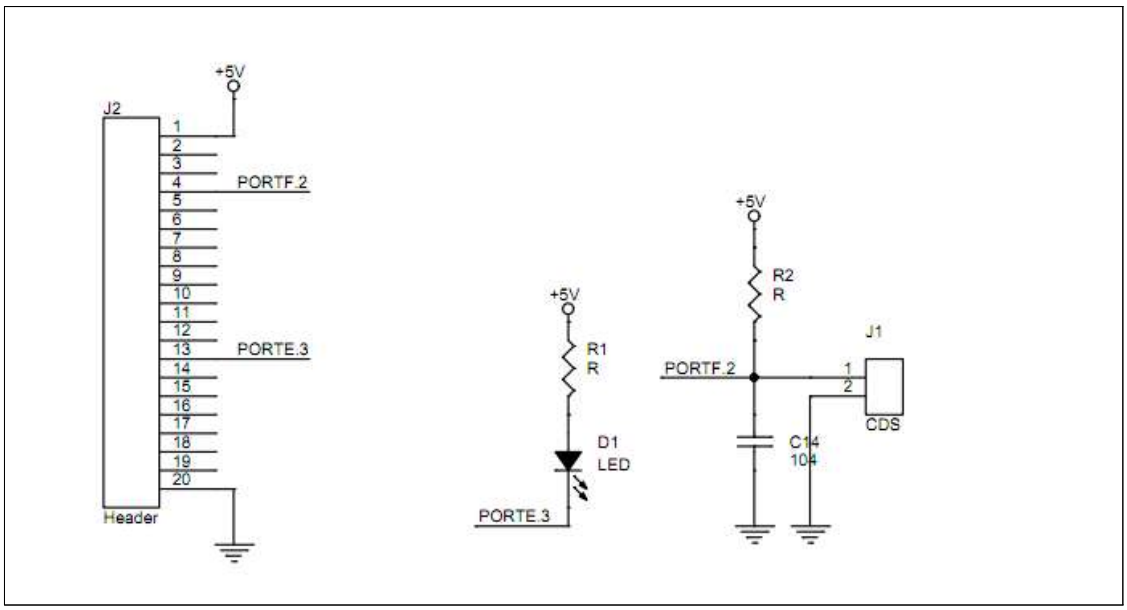
2.2 CDS란?



너희 조도 센서라고 들어봤어? 쉽게 말하자면 주변 밝기를 측정할 수 있는 센서야. 예를 들면 어두워지면 자동으로 켜지는 가로등, 밝기에 따라 변하는 핸드폰 화면 등이 있어. 근데 우리가 사용하려는 이 조도 센서는 황하카드뮴(CDS)을 소자로 사용했어. 그래서 CDS 센서라고 불리는 거야. 그럼 왜 하필 많은 조도 센서 중에 CDS를 쓰느냐. 이유는 간단해 CDS는 구조가 단순해서 싸고 보편적으로 사용되고 있거든.

2.3 회로도

자, 이제 이론적으로는 이해 됐지? 그럼 이제 프로그램을 하기 전에 만능기판으로 필요한 회로를 구성해보자. 나는 아래와 같이 구성했어.

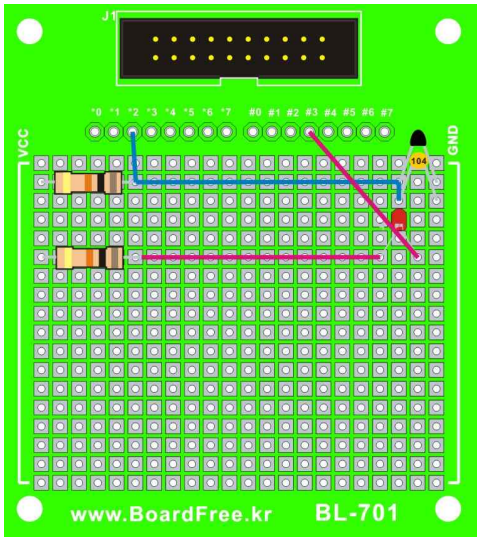


전구는 밝기를 PWM으로 조절할 거기 때문에 PORTE.3로 연결했고, CDS는 ADC값으로 읽기 위해 PORTF.2로 연결했어. 물론 꼭 이렇게 할 필요 없이 너희가 원하는 PWM과 ADC포트로 연결하면 되.

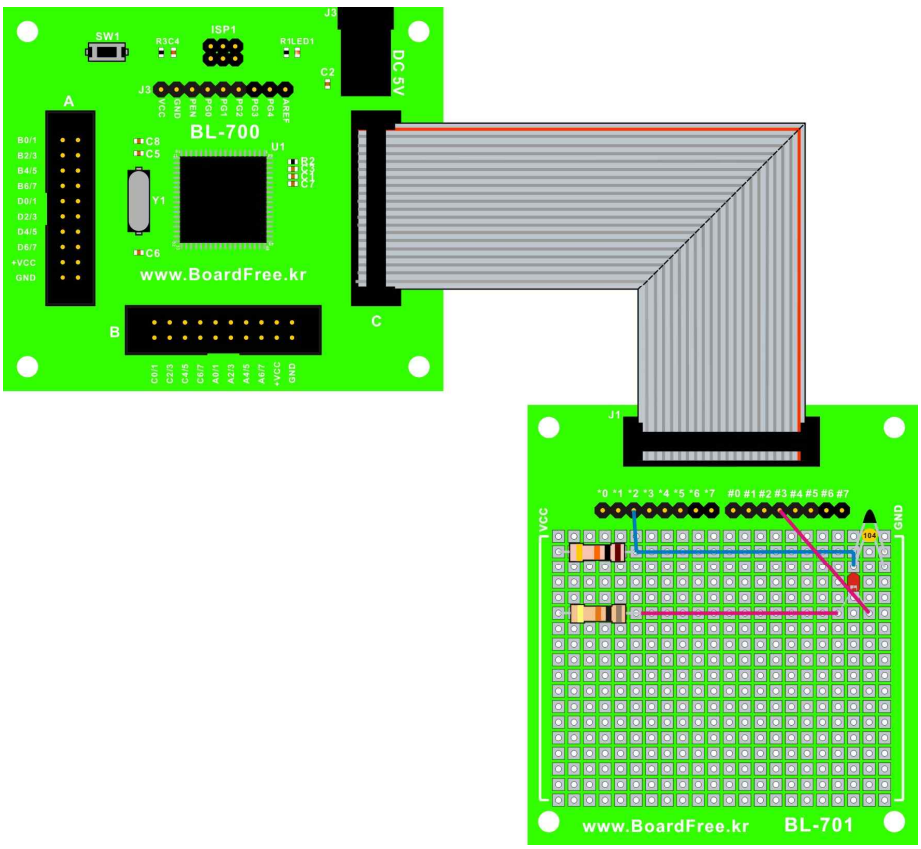
그리고 전구 저항값은 너희가 적당히 달면 되고, CDS 저항값은 너희가 가지고 있는 CDS의 데이터 시트를 잘 읽어보고, 전구의 밝기에 따른 ADC값의 변화를 봐가면서 저항을 달면 돼.

근데 여기서 가장 중요한 것은 CDS와 전구와의 거리, 전구의 밝기 정도야. 그러니까 되도록 가깝게 구성하고 전구의 밝기도 밝은 거로 해야지 ADC값의 변화를 확인하기 쉬울 거야.

위에 회로도로 나는 만능기판을 이렇게 구성했어.



보드 간의 연결은 아래처럼 했어.



2.4 동작해보기

CDS를 이용해서 LED 밝기 제어를 하기 전에 먼저 기본적으로 CDS값을 읽어보고 LED를 PWM값을 바꿔서 밝기를 조절해보자.

STEP1. PWM을 이용한 LED 밝기 조절

PWM은 전에 배웠지? 그때와 똑같아. PWM으로 LED의 밝기를 조절을 하는 거지. PWM 값을 바꾸면 LED의 밝기가 바뀌는 걸 볼 수 있을 거야.

```
#include <mega128.h>
#include <delay.h>

#define UDRE 5
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define ADC_VREF_TYPE 0x40
#define PWM OCR3AL
#define CDS_DATA_MSB_2bit adc_data>>8
#define CDS_DATA_LSB_8bit adc_data

void init();
unsigned int adc_data=0;

void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
```

```
}

void main(void)
{
    init();
    while (1)
    {
        PWM=0x50; //전구에 보낼 PWM
    }
}

void init()
{
    PORTE=0x00;
    DDRE=0x08;

    TCCR3A=0x81;
    TCCR3B=0x0b;
    OCR3AH=0x00;
    OCR3AL=0xef;

    TIMSK=0x00;

    UCSR1A=0x00;
    UCSR1B=0x08;
    UCSR1C=0x06;
    UBRR1H=0x00;
    UBRR1L=0x08;

    ACSR=0x80;
    SFIOR=0x00;

    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
}
```

STEP2. ADC로 CDS값 읽어서 터미널 창에 띄우기

통신 보드 했던 것도 기억나지? 그때랑 똑같아 ADC로 센서 값을 읽고 그 값을 그대로 터미널에 띄우면 되는 거야.

그 전에 통신 보드를 연결하려는데 전에 했던 자리에 이미 만능기판 보드가 연결되어있지? ATmega128은 통신 포트가 또 있어. 그러니까 그에 해당하는 포트가 있는 박스헤더에 연결하면 되!

```
//설정은 1번과 똑같음

void main(void)
{
    init();
    while (1)
    {
        adc_data=read_adc(2); //ADC2로 들어온 data값을 adc_data에 저장
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력
        delay_ms(100); // 통신으로 값을 보내줄 시간
    }
}
```

STEP3. 어떻게 하면 CDS로 읽어온 값을 일정하게 할 수 있을까?

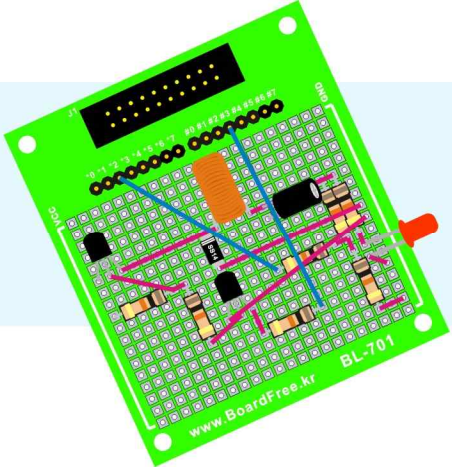
이젠 위에 두 개를 합쳐서 CDS로 측정된 밝기가 항상 일정하게 유지되도록 전구의 밝기를 조절하게 할 거야. 주변이 밝으면, 전구를 어둡게 하고, 주변이 어두우면 전구를 밝게 해서 항상 CDS로 읽어온 밝기를 똑같이 유지 되도록 하는 거야. 이번엔 너희들이 해봐!

```
#include <mega128.h>
#include <delay.h>

#define UDRE 5
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define ADC_VREF_TYPE 0x40
#define PWM OCR3AL
#define CDS_DATA_MSB_2bit adc_data>>8
#define CDS_DATA_LSB_8bit adc_data
```

```
void init();
unsigned int adc_data=0;

void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void main(void)
{
    init();
    while (1)
    {
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력

        //adc_data=read_adc(2); // CDS값
        //PWM=0x50; //전구에 보낼 PWM

        //CDS값을 일정하게 하려면 PWM값을 어떻게 줘야할까??
        //이제 너희가 코드를 작성해봐!

    }
}

void init()
{
    PORTE=0x00;
```

```
    DDRE=0x08;

    TCCR3A=0x81;
    TCCR3B=0x0b;
    OCR3AH=0x00;
    OCR3AL=0xef;

    TIMSK=0x00;

    UCSR1A=0x00;
    UCSR1B=0x08;
    UCSR1C=0x06;
    UBRR1H=0x00;
    UBRR1L=0x08;

    ACSR=0x80;
    SFIOR=0x00;

    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
}
```

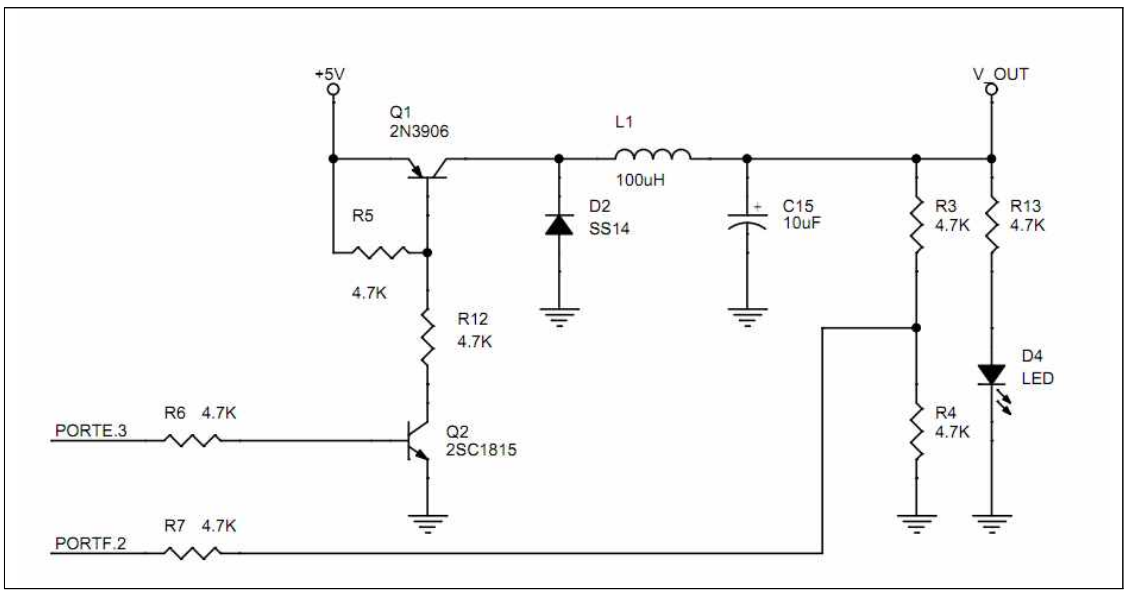
3 BUCK 타입을 이용한 전압제어



3.1 BUCK이 뭔데요??

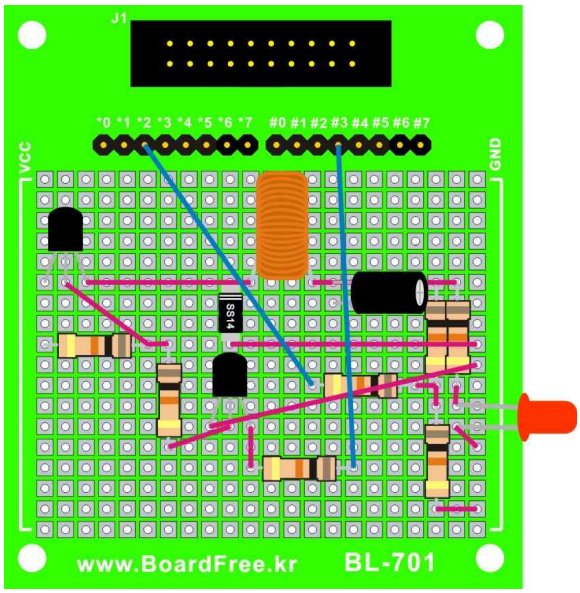
BUCK이란 쉽게 말해 +5V를 입력했다고 했을 때, +5V 이하의 원하는 전압으로 만들어주는 변환해주는 방식을 의미해. BUCK Converter는 DC/DC 컨버터의 대표적인 방식으로써 칩으로 만들어져서 많이 사용되고 있어.

3.2 회로도



위에 회로도는 BUCK Converter의 기본 회로야. 추가된 회로는 전압을 ADC로 읽어온 점과 PWM으로 TR을 ON/OFF시켜서 출력 전압을 조절할 수 있게 만든 점, 부하로 LED를 넣은 점이야. ADC로 가는 분압 저항 값은 읽은 데이터 값을 보고 원하는 데로 수정해도 좋아!

아래 그림은 내가 위에 회로를 바탕으로 만든 만능기판이야!



3.3 동작해보기

이번 단원에서는 BUCK 방식을 사용해서 출력 전압을 흔들리지 않게 조절을 해볼 거야.

STEP1. PWM과 ADC값을 사용해서 전압 확인해보기

아래 코드는 CDS에서 사용했던 코드와 같아. 아래와 같이하면 터미널 창을 통해 전압의 변화를 확인할 수 있어.

```
#include <mega128.h>
#include <delay.h>

#define UDRE 5
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define ADC_VREF_TYPE 0x40
#define PWM OCR3AL
#define CDS_DATA_MSB_2bit adc_data>>8
#define CDS_DATA_LSB_8bit adc_data

void init();
```

```
unsigned int adc_data=0;

void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void main(void)
{
    init();
    while (1)
    {
        adc_data=read_adc(2); //전압값
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력
        PWM=0x10;
    }
}

void init()
{
    PORTE=0x00;
    DDRE=0x08;

    TCCR3A=0x81;
    TCCR3B=0x0b;
    OCR3AH=0x00;
    OCR3AL=0xef;
```

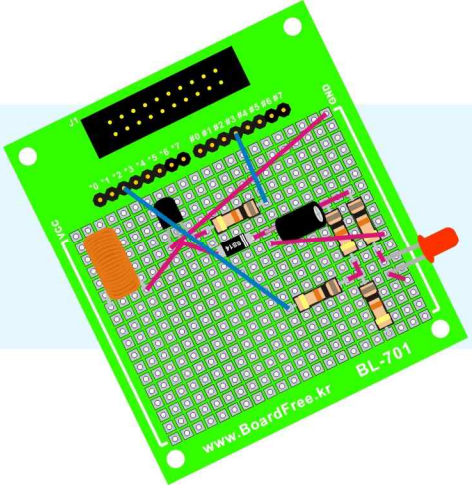
```
    TIMSK=0x00;

    UCSR1A=0x00;
    UCSR1B=0x08;
    UCSR1C=0x06;
    UBRR1H=0x00;
    UBRR1L=0x08;

    ACSR=0x80;
    SFIOR=0x00;

    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
}
```

STEP2. 어떻게 하면 전압을 일정하게 맞출 수 있을까?

아래 코드는 위에 설정과 똑같은 상태야. 이 설정(PWM설정 등)을 바꿔도 좋고 회로의 저항값, 캐패시터 값 등을 바꿔도 좋아. ADC로 읽히는 전압 값을 일정하게 해봐!

```
void main(void)
{
    init();
    while (1)
    {
        adc_data=read_adc(2); //전압값
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력

        //PWM=0x10; //TR에 보낼 PWM

        //전압값을 일정하게 하려면 어떻게 해야할까??
        //이제 너희가 코드를 작성해봐!
```

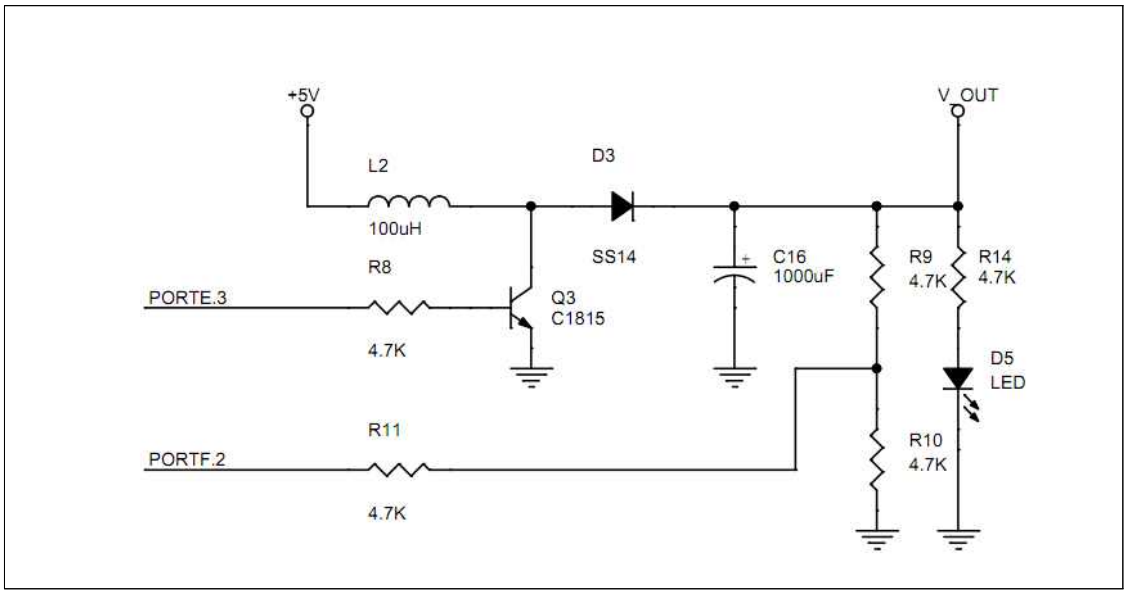
4 BOOST 타입을 이용한 전압제어



4.1 BOOST가 뭔데요??

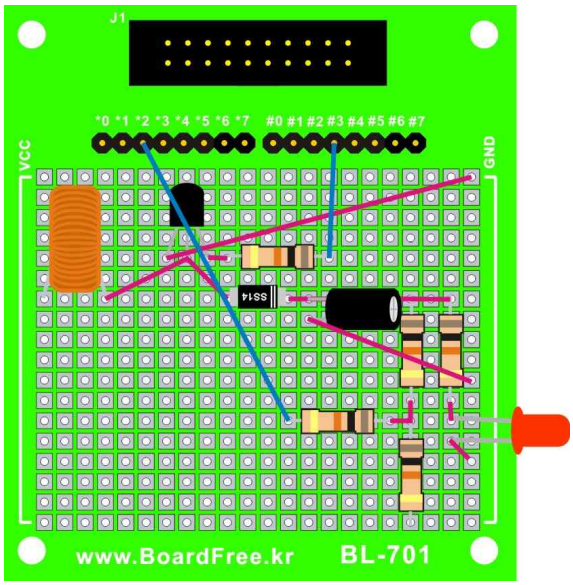
BOOST란 쉽게 말해 +5V를 입력했다고 했을 때, +5V 이상의 원하는 전압으로 만들어주는 변환해주는 방식을 의미해. BOOST Converter는 BUCK Converter 외에 DC/DC 컨버터의 또 하나의 대표적인 방식으로써 칩으로 만들어져서 많이 사용되고 있어.

4.2 회로도



위에 회로도는 Boost Converter의 기본 회로야. 추가된 회로는 전압을 ADC로 읽어온 점과 PWM으로 TR을 ON/OFF시켜서 출력 전압을 조절할 수 있게 만든 점, 부하로 LED를 넣은 점이야. ADC로 가는 분압 저항 값은 읽은 데이터 값을 보고 원하는 데로 수정해도 좋아!

아래 그림은 내가 위에 회로를 바탕으로 만든 만능기판이야!



4.3 동작해보기

이번 단원에서는 Boost 방식을 사용해서 출력 전압을 흔들리지 않고 일정하도록 조절을 해볼거야.

STEP1. PWM과 ADC값을 사용해서 전압 확인해보기

아래 코드는 이전에 사용했던 코드들과 같아. 아래와 같이하면 터미널 창을 통해 전압의 변화를 확인할 수 있어.

```
#include <mega128.h>
#include <delay.h>

#define UDRE 5
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define ADC_VREF_TYPE 0x40
#define PWM OCR3AL
#define CDS_DATA_MSB_2bit adc_data>>8
#define CDS_DATA_LSB_8bit adc_data

void init();
unsigned int adc_data=0;

void putchar1(char c)
```

```
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void main(void)
{
    init();
    while (1)
    {
        adc_data=read_adc(2); //전압값
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력
        PWM=0x50;
    }
}

void init()
{
    PORTE=0x00;
    DDRE=0x08;

    TCCR3A=0x81;
    TCCR3B=0x0b;
    OCR3AH=0x00;
    OCR3AL=0xef;

    TIMSK=0x00;

    UCSR1A=0x00;
```

```
    UCSR1B=0x08;
    UCSR1C=0x06;
    UBRR1H=0x00;
    UBRR1L=0x08;

    ACSR=0x80;
    SFIOR=0x00;

    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
}
```

STEP2. 어떻게 하면 전압을 일정하게 맞출 수 있을까?

아래 코드는 위에 설정과 똑같은 상태야. 이 설정(PWM설정 등)을 바꿔도 좋고 회로의 저항값, 캐패시터 값 등을 바꿔도 좋아. ADC로 읽히는 전압 값을 일정하게 해봐!

```
void main(void)
{
    init();
    while (1)
    {
        adc_data=read_adc(2); //전압값
        putchar1(CDS_DATA_MSB_2bit); // [9],[8] ADC data 출력
        putchar1(CDS_DATA_LSB_8bit); // [7]~[0] ADC data 출력

        //PWM=0x10; //TR에 보낼 PWM

        //전압값을 일정하게 하려면 어떻게 해야할까??
        //이제 너희가 코드를 작성해봐!


    }
}
```