

# Introduction to Embedded Systems

Introduction to Embedded Systems  
Programming  
Lecture 1

Reference: Chapters (1..4) Arduino Cookbook, Michael Margolis, 1st Edition, 2011, O'Reilly books

# Agenda

- Introduction
- Basic Board Programming
- Programming with C/C++
- Electronics Review

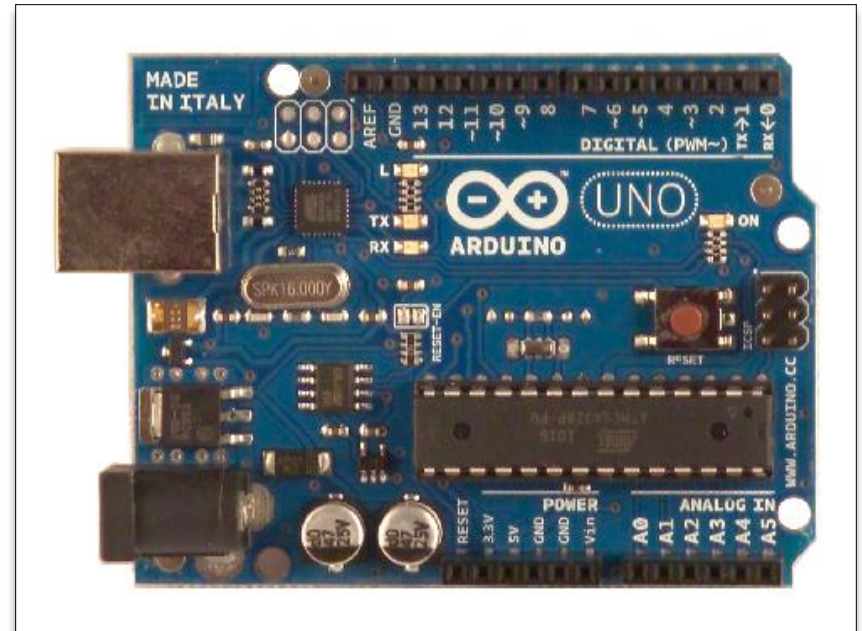
# Introduction

# Requirements

- Required Background Knowledge
  - Logic Design
  - C/C++ Programming
  - Basic Electronics
  - Basic Computer Architecture
- Required Resources
  - Arduino UNO Board with ATmega328P microcontroller
  - Arduino Development Environment
  - ISIS/Proteus Software (Version 8.X)

# Arduino UNO Board

- ATmega328P Microcontroller
- 5V (40 ma for I/O PINs)
- 14 Digital I/O (6 with PWM)
- 6 Analog IP (10 Bit ADC)
- Analog Comparator
- Flash Memory 32KB (0.5K for Arduino Boot-loader)
- SRAM 2KB
- EEPROM 1KB
- 16 MHZ
- 2 8-Bit Real Time Timers, 1 Watchdog Timer
- 1 16-Bit Counter
- Communication SPI/I2C/Serial



- USB2Serial to Interface with Computer (Programming and Communication)
- Indicators (LEDs) for Serial Communication PINs (1,2) and PIN 13
- Standalone Power
- Reset Button

# On/Off Board Led

```
test | Arduino 0022
File Edit Sketch Tools Help

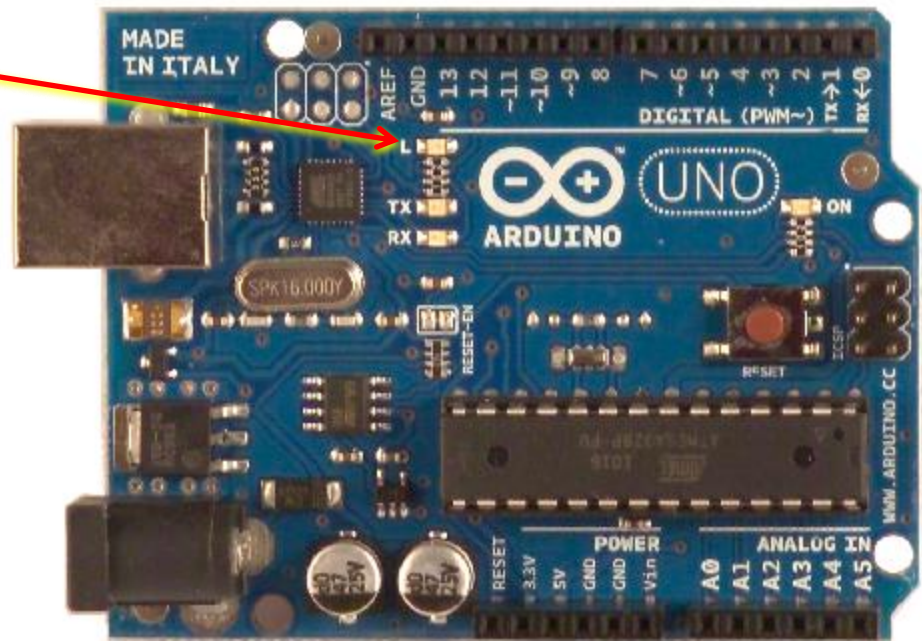
test$

#define LED 13
#define DT 500

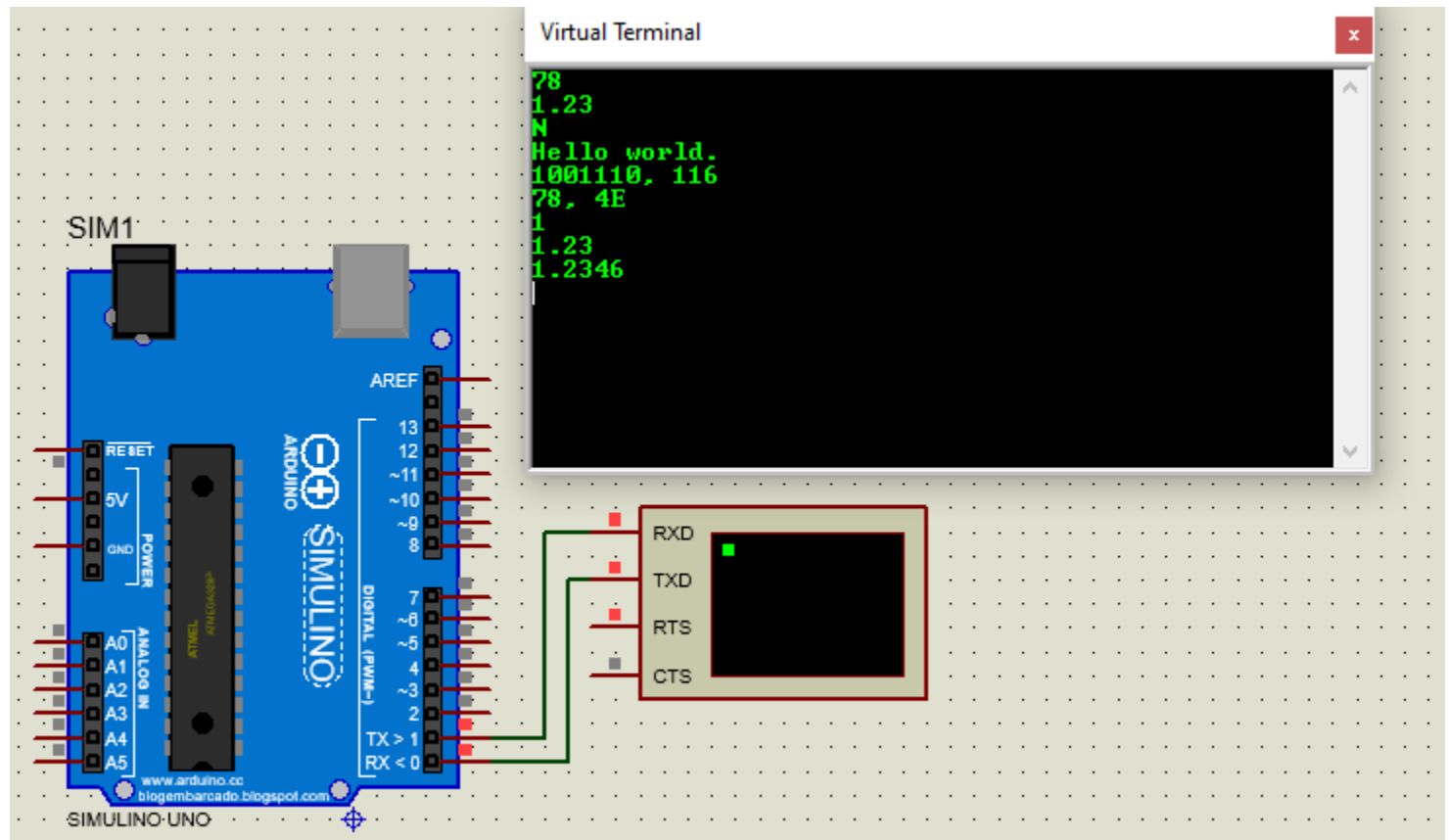
void setup()
{
    pinMode(LED, OUTPUT);
}

void loop()
{
    digitalWrite(LED, HIGH);
    delay(DT);
    digitalWrite(LED, LOW);
    delay(2*DT);
}

Done uploading.
Binary sketch size: 1588 bytes (of a 258048 byte
2
```

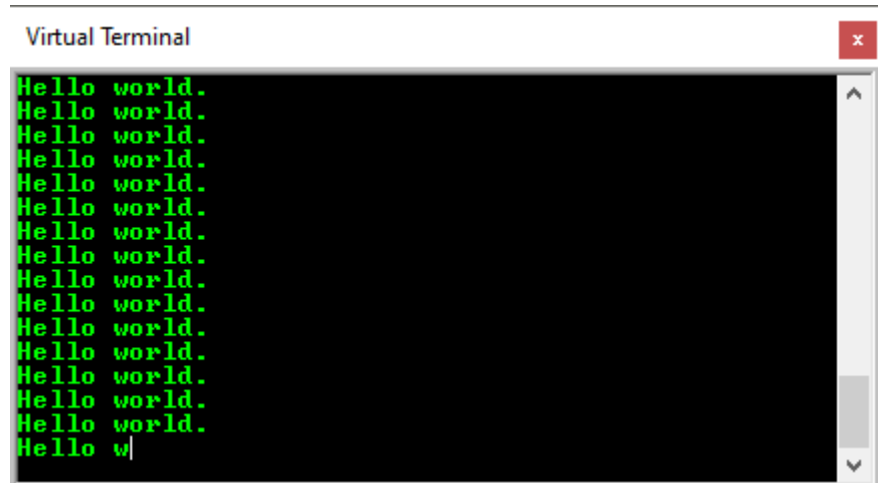


# Serial Communication



# Serial Communication

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Hello world.");  
}
```





# Serial Communication

```
int incomingByte = 0; // for incoming serial data

void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();

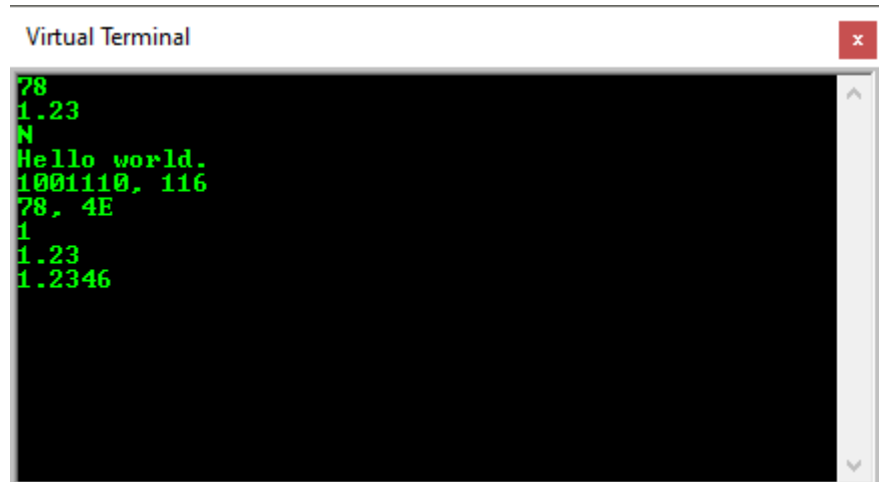
    // say what you got:
    Serial.print("I received: ");
    Serial.println(incomingByte, DEC);
  }
}
```

Virtual Terminal

```
I received: 49
I received: 50
I received: 51
I received: 13
I received: 52
```

# Serial Communication

```
void setup() {  
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps  
}  
  
void loop() {  
  if(Serial.available())  
  {  
    Serial.read();  
    Serial.println(78);  
    Serial.println(1.23456);  
    Serial.println('N');  
    Serial.println("Hello world.");  
    Serial.print(78, BIN);  
    Serial.print(", ");  
    Serial.print(78, OCT);  
    Serial.println("");  
    Serial.print(78, DEC);  
    Serial.print(", ");  
    Serial.print(78, HEX);  
    Serial.println("");  
    Serial.println(1.23456, 0);  
    Serial.println(1.23456, 2);  
    Serial.println(1.23456, 4);  
  }  
}
```



The screenshot shows a 'Virtual Terminal' window with a black background and green text. The output corresponds to the code in the loop function, displaying the value 78 in various formats and the floating-point number 1.23456 with different precision settings.

```
78  
1.23  
N  
Hello world.  
1001110, 116  
78, 4E  
1  
1.23  
1.2346
```

# Programming with C/C++

# Supported Data Types

- `int`
  - 2 byte signed integer value, Range: -32768 to 32767
- `unsigned int`
  - 2 byte unsigned integer value, Range: 0 to 65535
- `long`
  - 4 byte signed integer value, Range: -2147483648 to 2147483647
- `unsigned long`
  - 4 byte unsigned integer value, Range: 0 to 4294967295
- `float/double`
  - 4 byte real value, Range: -3.4028235E+38 to 3.4028235E+38
  - Resolution: 3.4028235E-38

# Supported Data Types

- **boolean**
  - 1 byte integer value. false (0) or true (1)
- **char**
  - 1 byte signed integer value, Range: -128 to 127
- **byte**
  - 1 byte signed integer value, Range: 0 to 255
- **string**
  - C++ class represents array of chars
- **void**
  - used by function to indicate no value is returned
  - used by pointers to indicate unknown data type

# Using Float Data Type

```
float value = 1.1;
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    value = value - 0.1;
    if( value == 0)
        Serial.println("The value is exactly zero");
    else if(fabs(value) < .0001)
        Serial.println("The value is close enough to zero");
    else
        Serial.println(value);
    delay(100);
}
```

# Using Arrays

```
int inputPins[] = {2,3,4,5};
int ledPins[] = {10,11,12,13};
void setup()
{
    for(int index = 0; index < 4; index++)
    {
        pinMode(ledPins[index], OUTPUT);
        pinMode(inputPins[index], INPUT);
    }
}
void loop()
{
    for(int index = 0; index < 4; index++)
    {
        int val = digitalRead(inputPins[index]);
        if (val == LOW)
        {
            digitalWrite(ledPins[index], HIGH);
        }
        else
        {
            digitalWrite(ledPins[index], LOW);
        }
    }
}
```

# Using Strings

```
String text1 = "Welcome to ";
String text2 = " Arduino";
String text;
void setup()
{
    Serial.begin(9600);

    text = text1 + text2 + " board";
    Serial.println(text);
    Serial.print("Length : ");
    Serial.println(text.length());

    for(int i=0;i<text.length();i++)
    {
        Serial.print(text[i]); Serial.print(" ");
    }
    Serial.println("");

    text = text.toUpperCase();
    Serial.println(text);

    text = " hello ";
    text = text.trim();
    Serial.println(text);
}

void loop() {}
```



# Electronics Review

Using ISIS Proteus Software

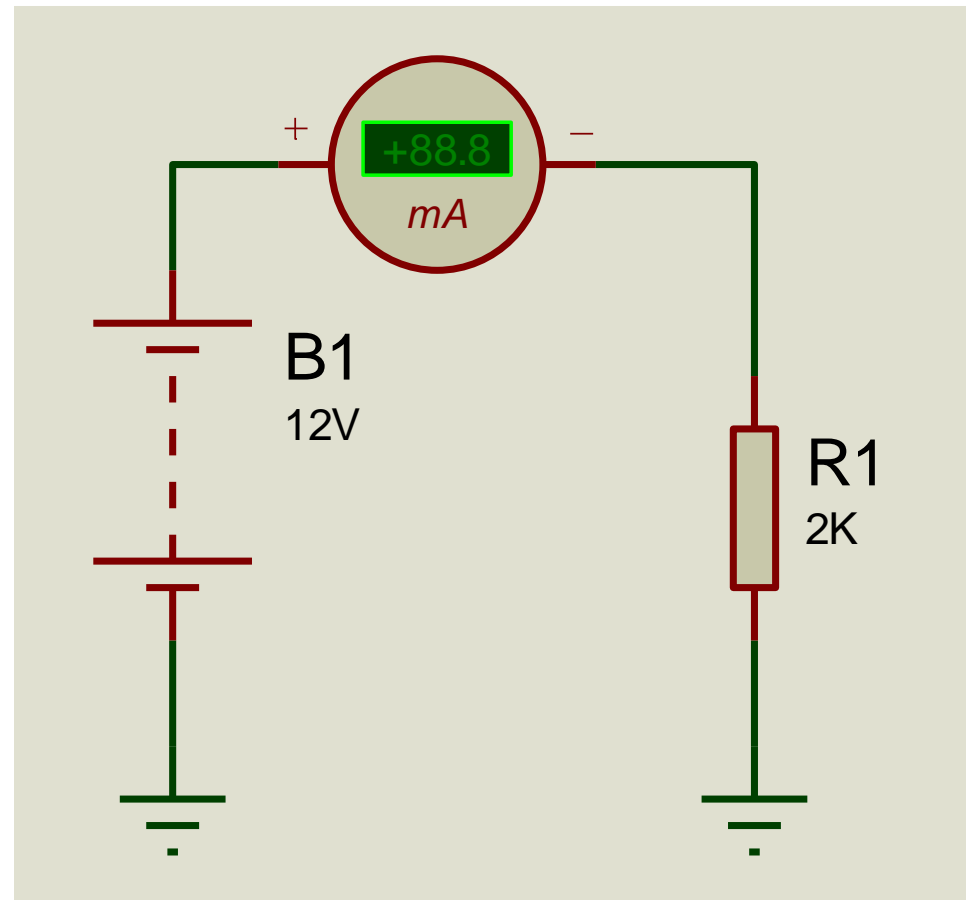
# Electronics Review

- Resistance
- Capacitor
- Diodes
- Transistors
- Operational Amplifier

# Resistance

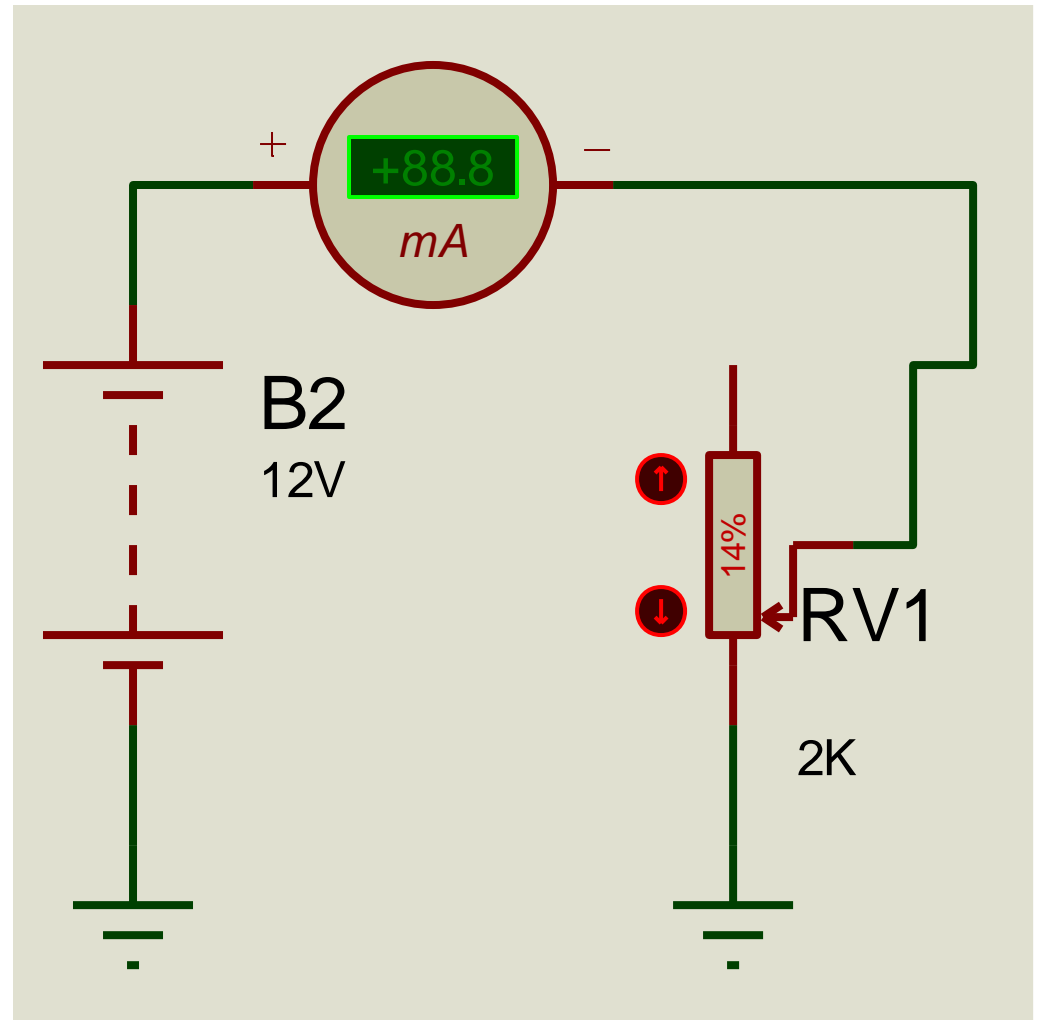
# Volt to Current Relation

- $I = V/R$

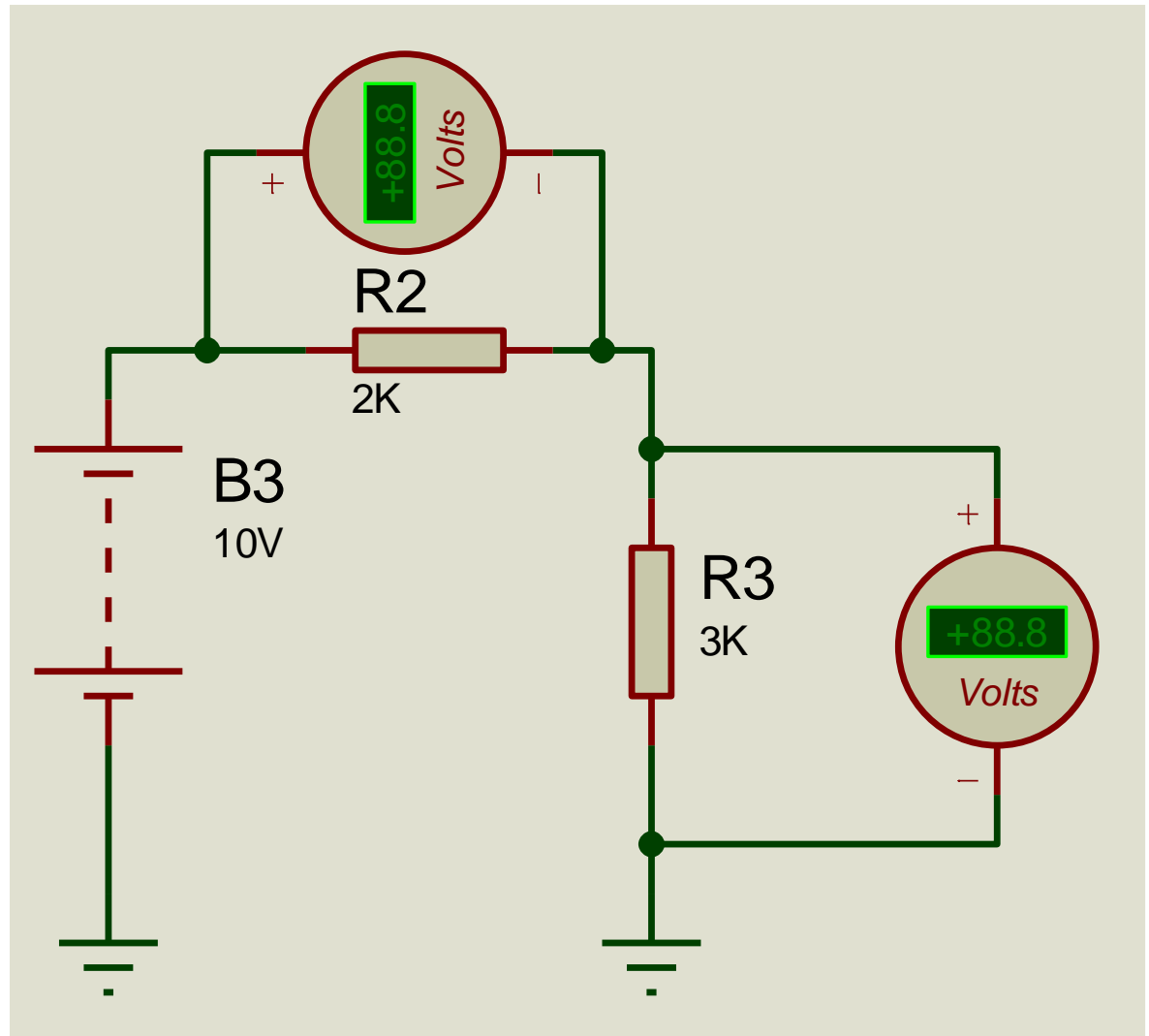


# Volt to Current Relation

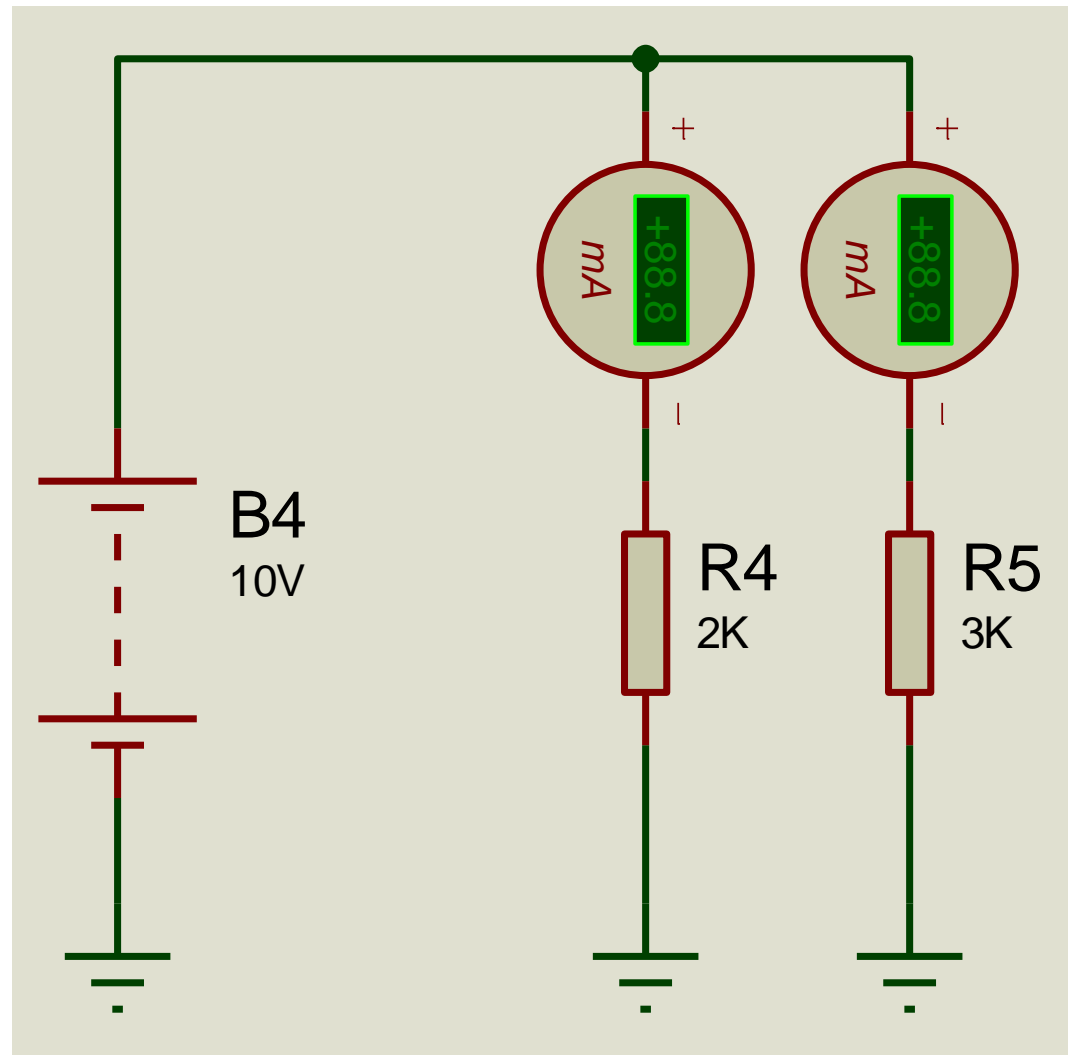
- Current Increase when Resistance Decrease



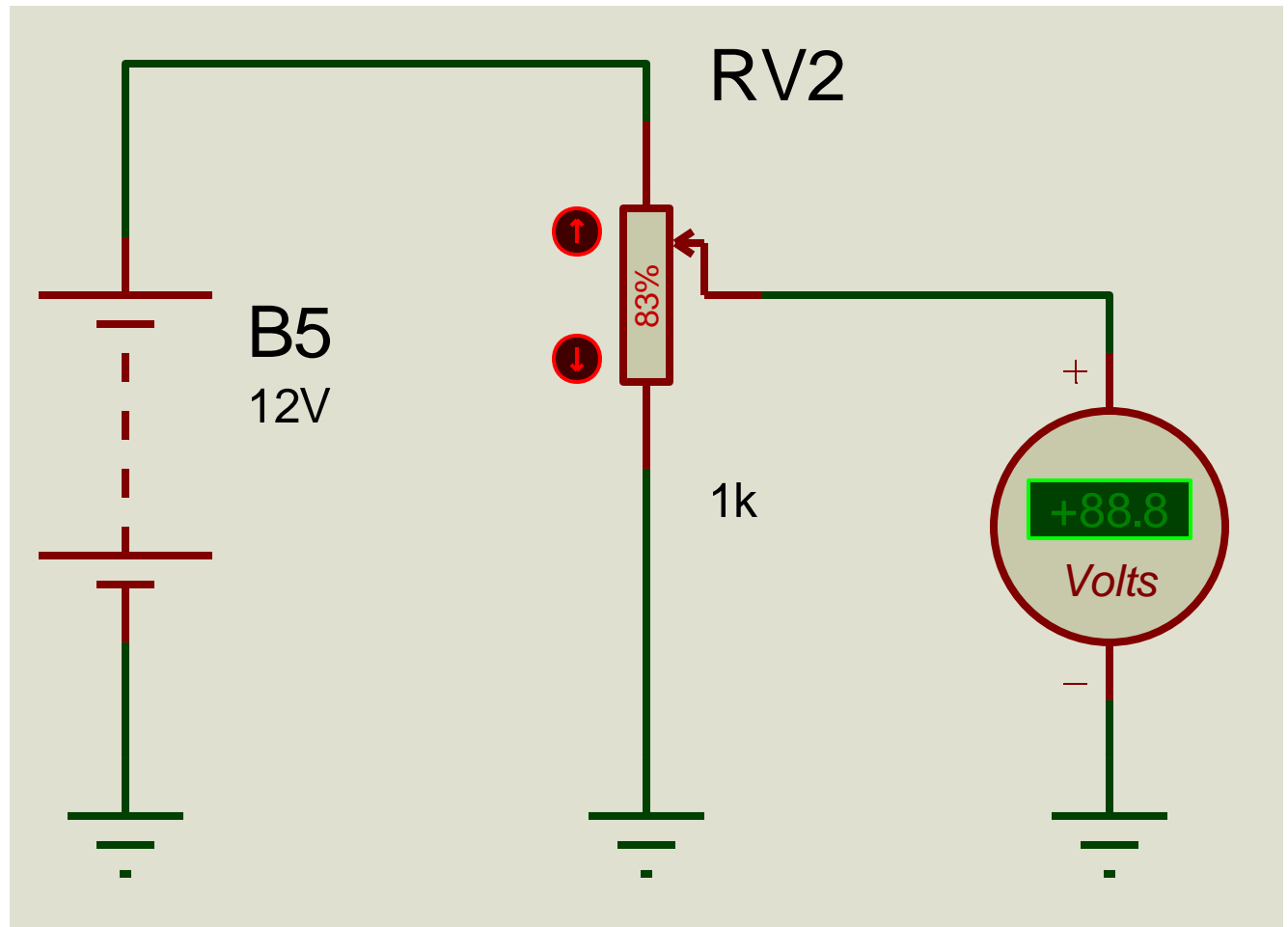
# Series Resistors



# Parallel Resistors



# Produce Variable Voltage using Potentiometer (Variable Resistance)

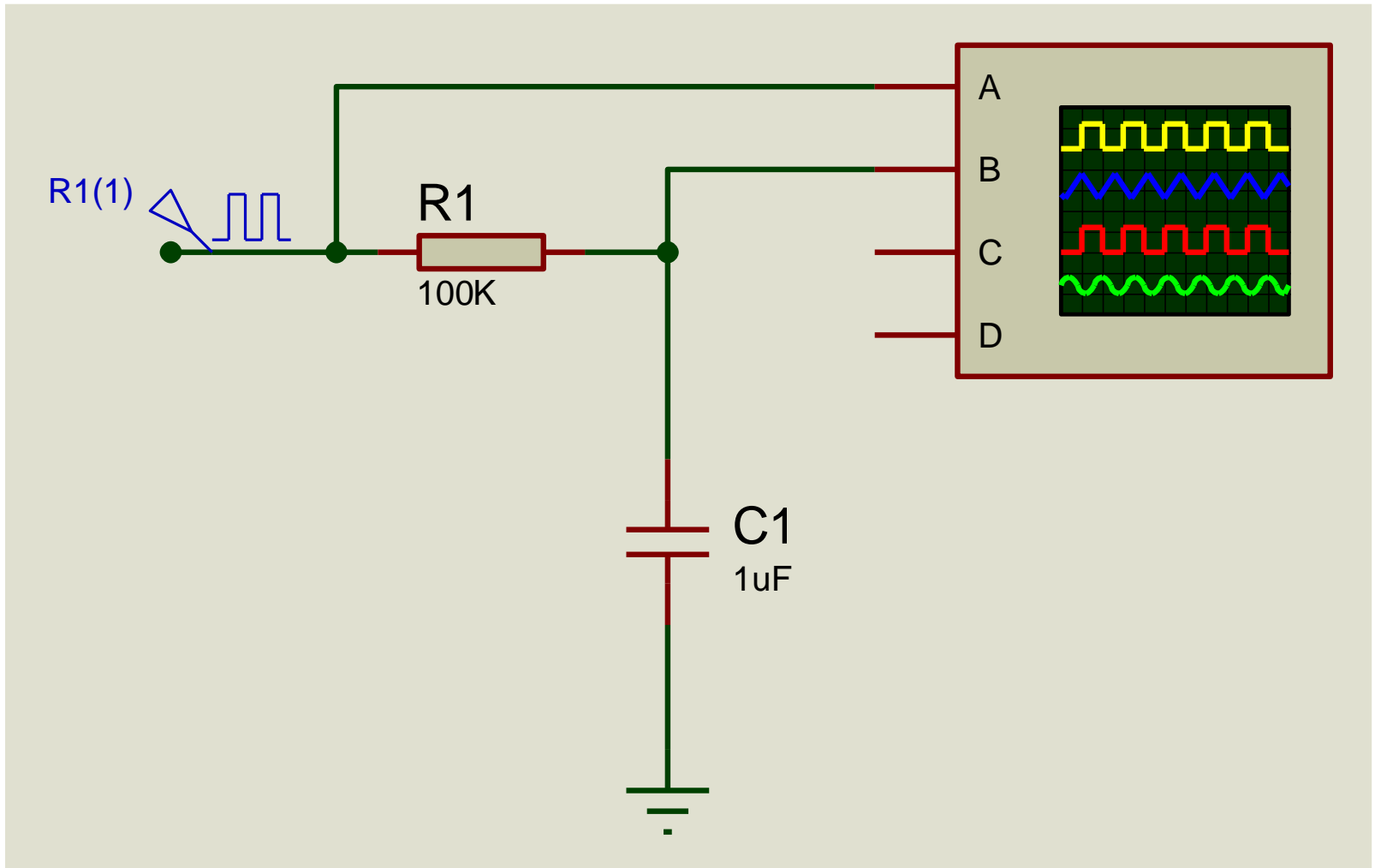




# Capacitors

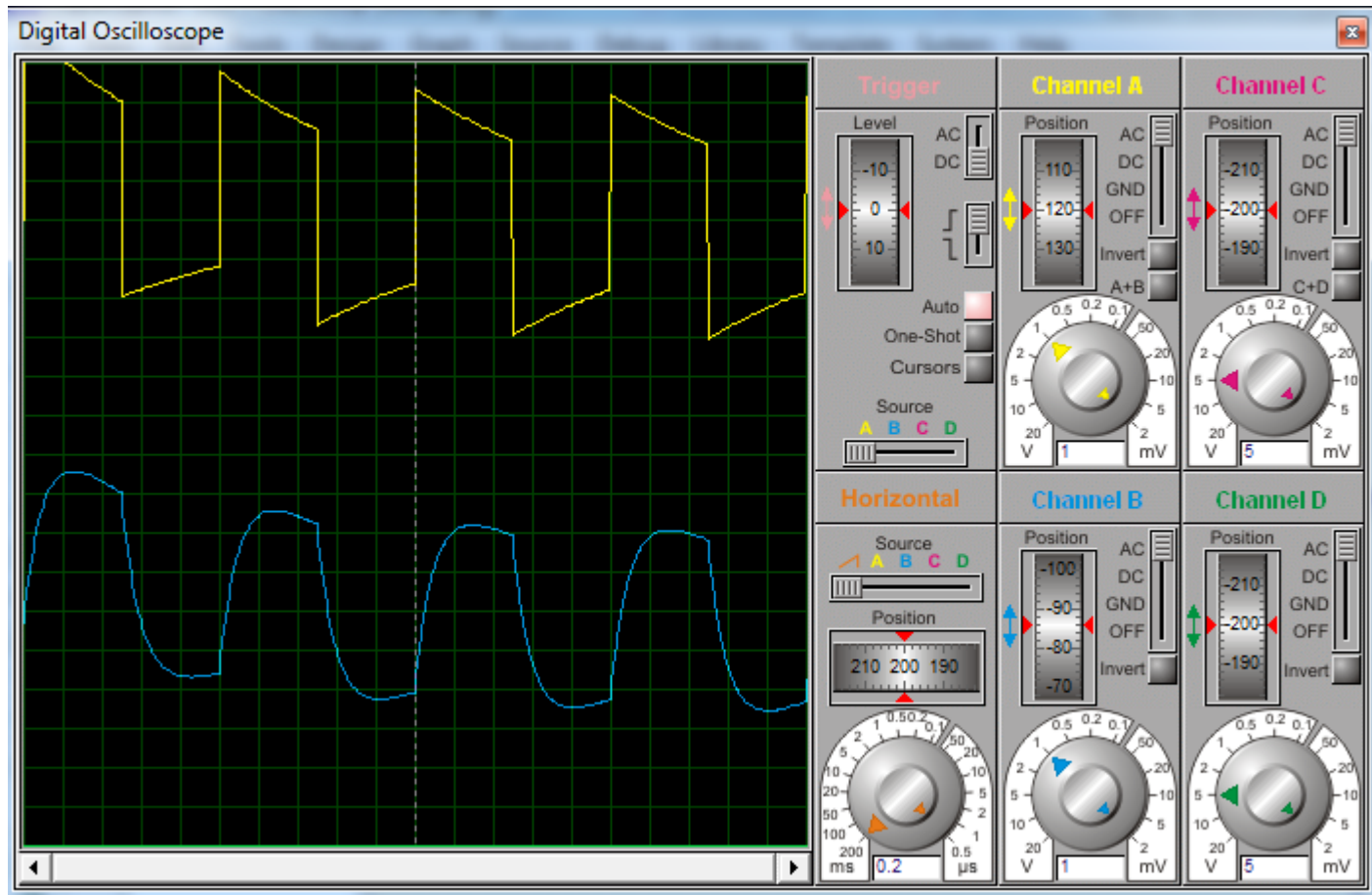
# Low Pass RC Filter

## Remove Sharp Edges and Noise



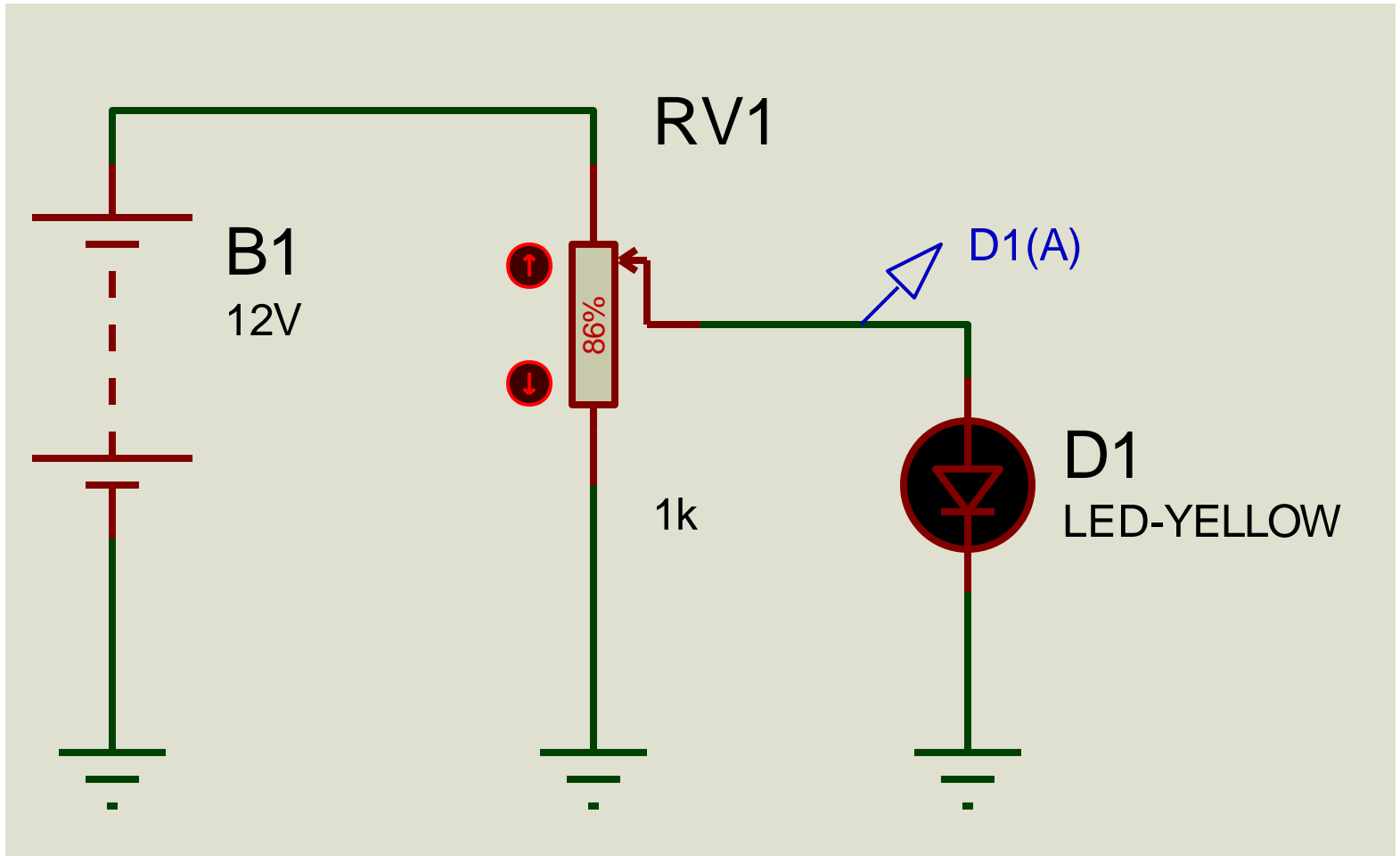
# Low Pass RC Filter

## Remove Sharp Edges and Noise

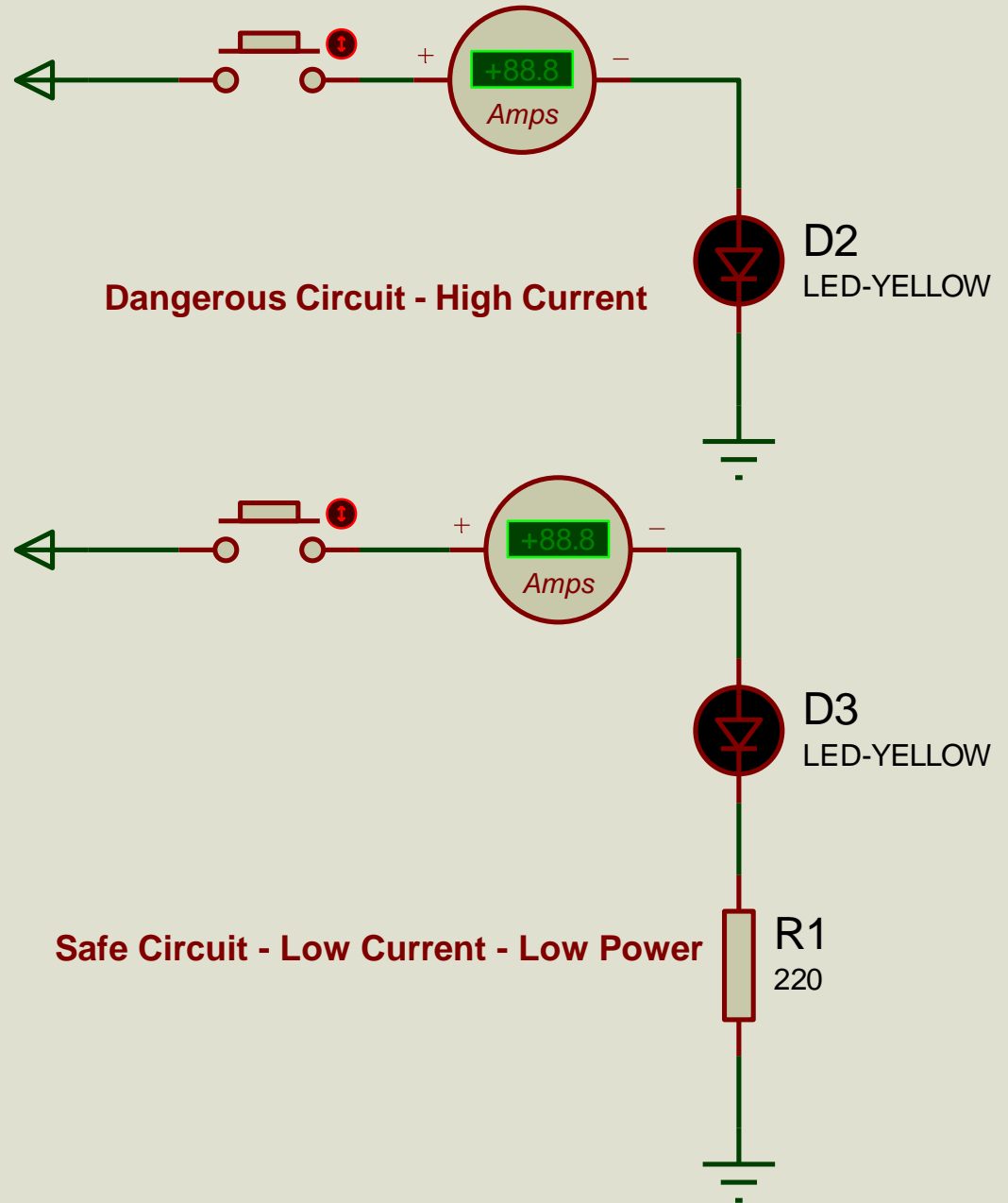


# Diodes

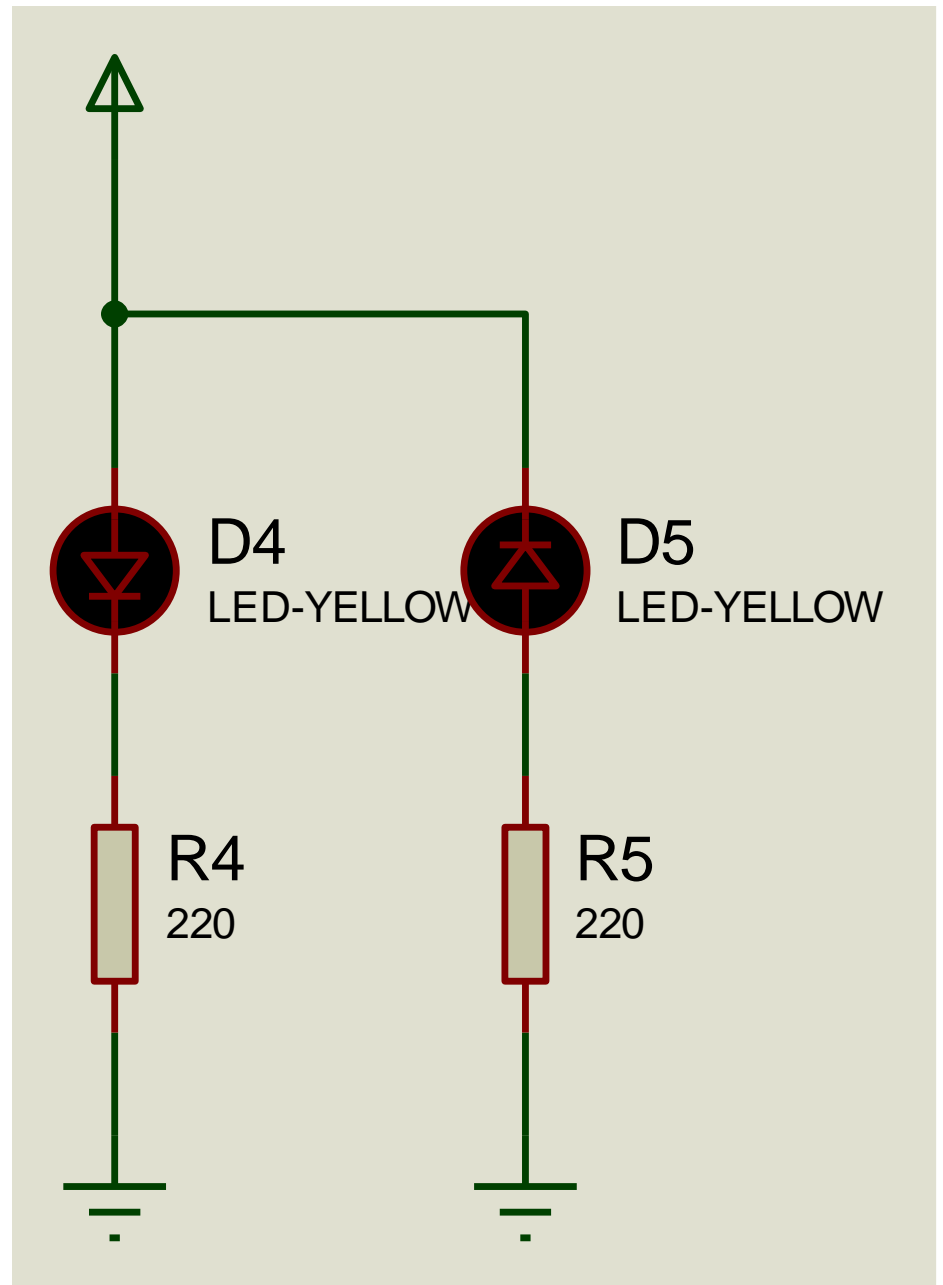
# Switch Diode On/Off



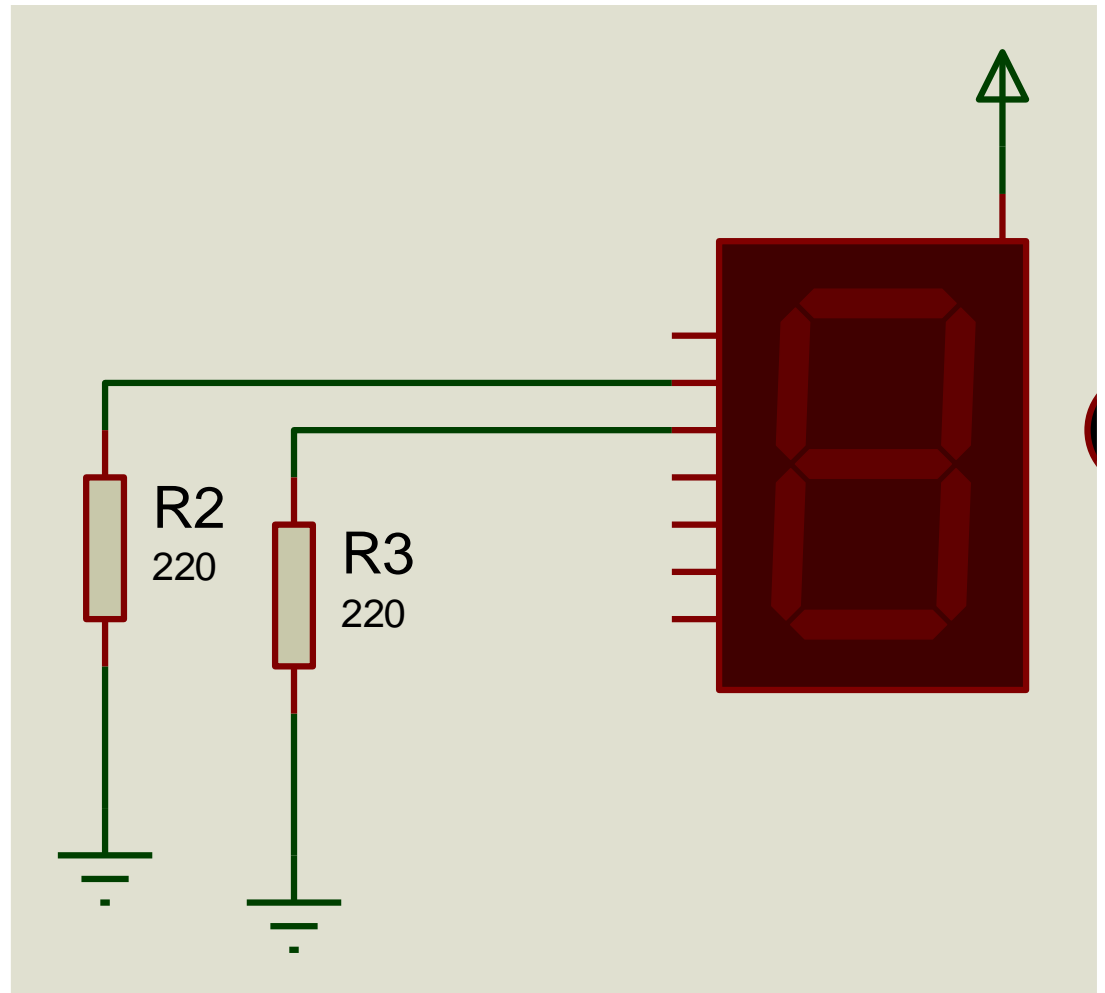
# Protecting Diode Circuits using Series Resistance



# Forward and Reverse Diode Connection



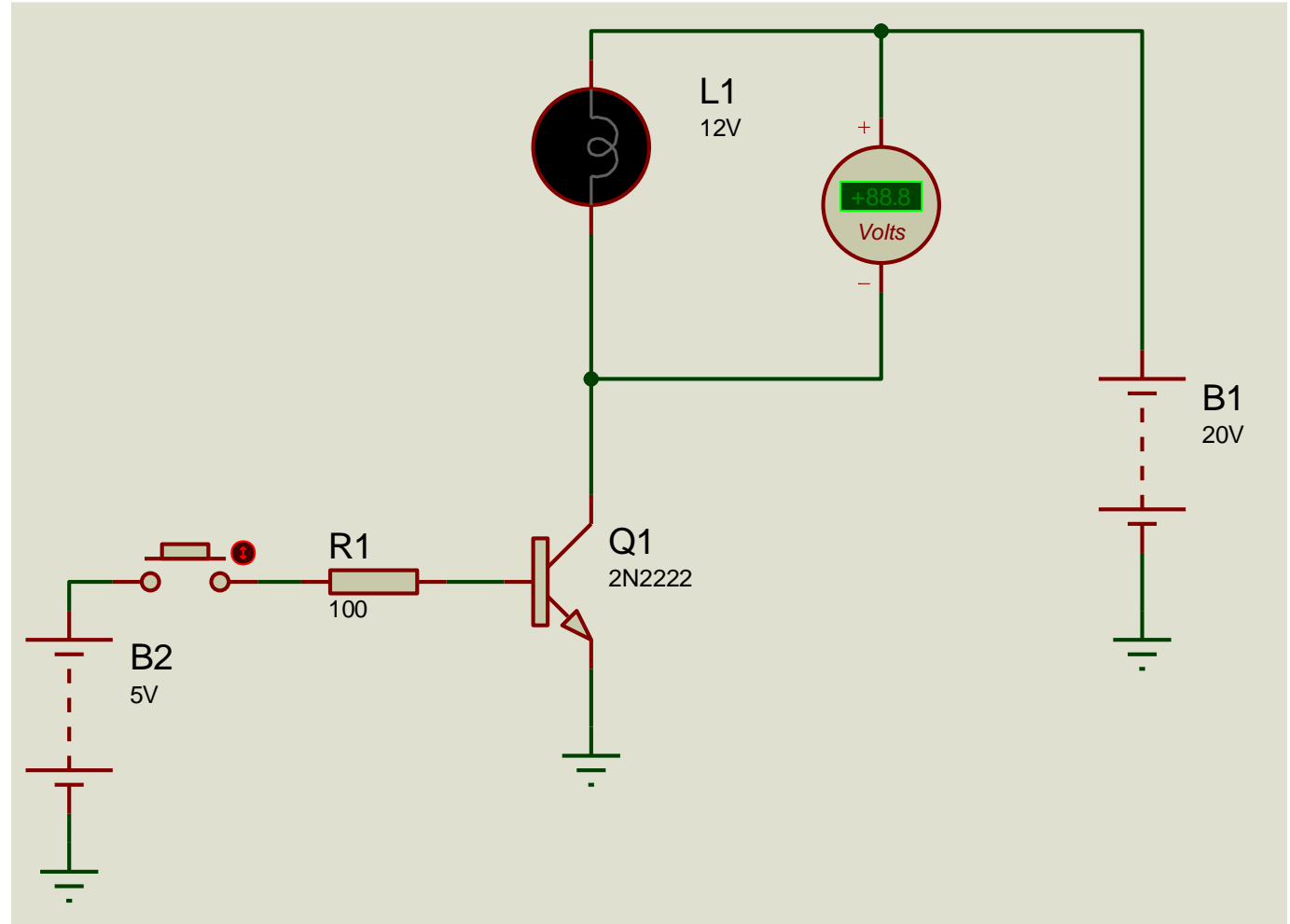
# 7-Segment Display (7 Diodes)





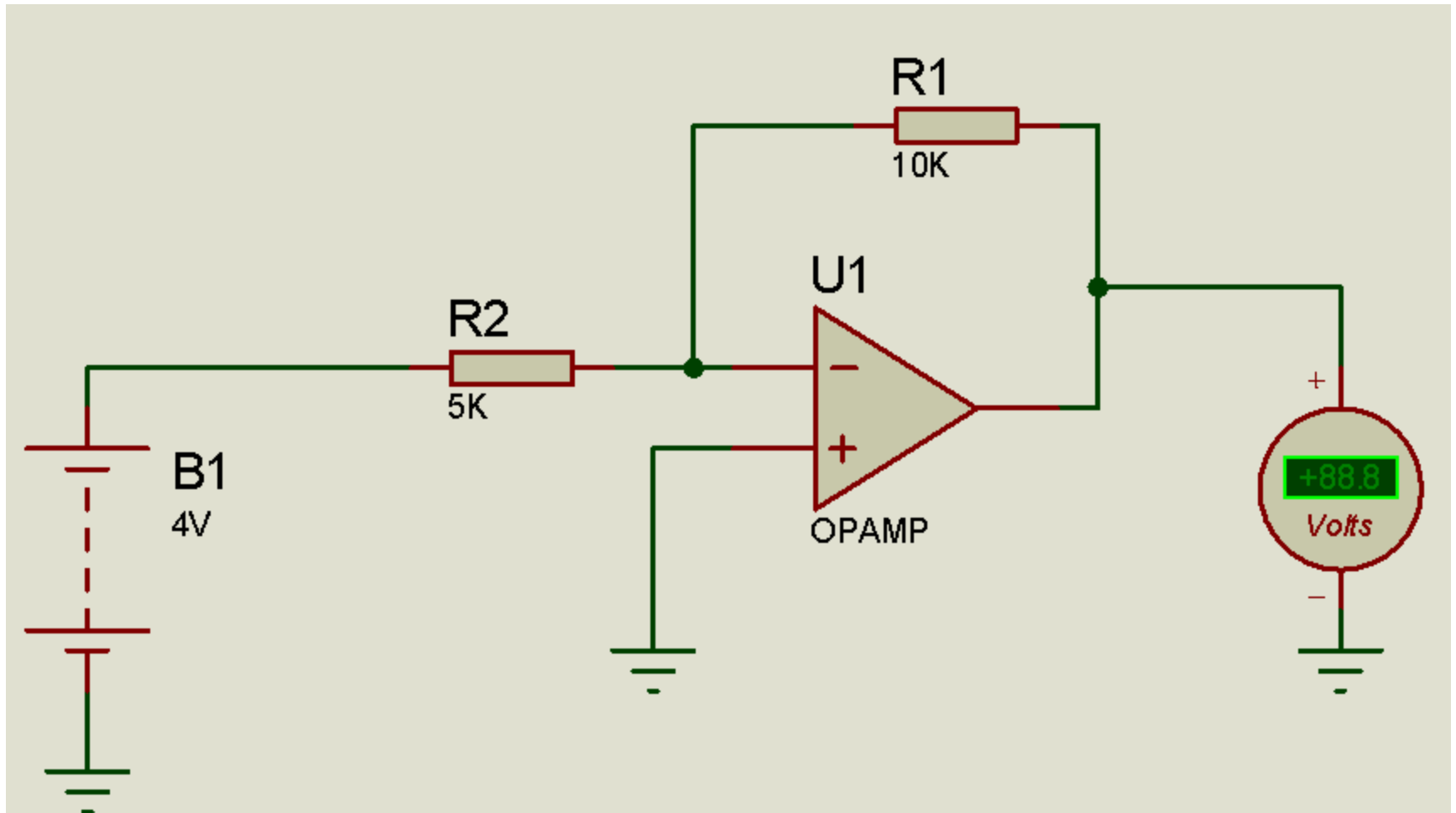
# Transistor

# Control High Voltage using Small Voltage

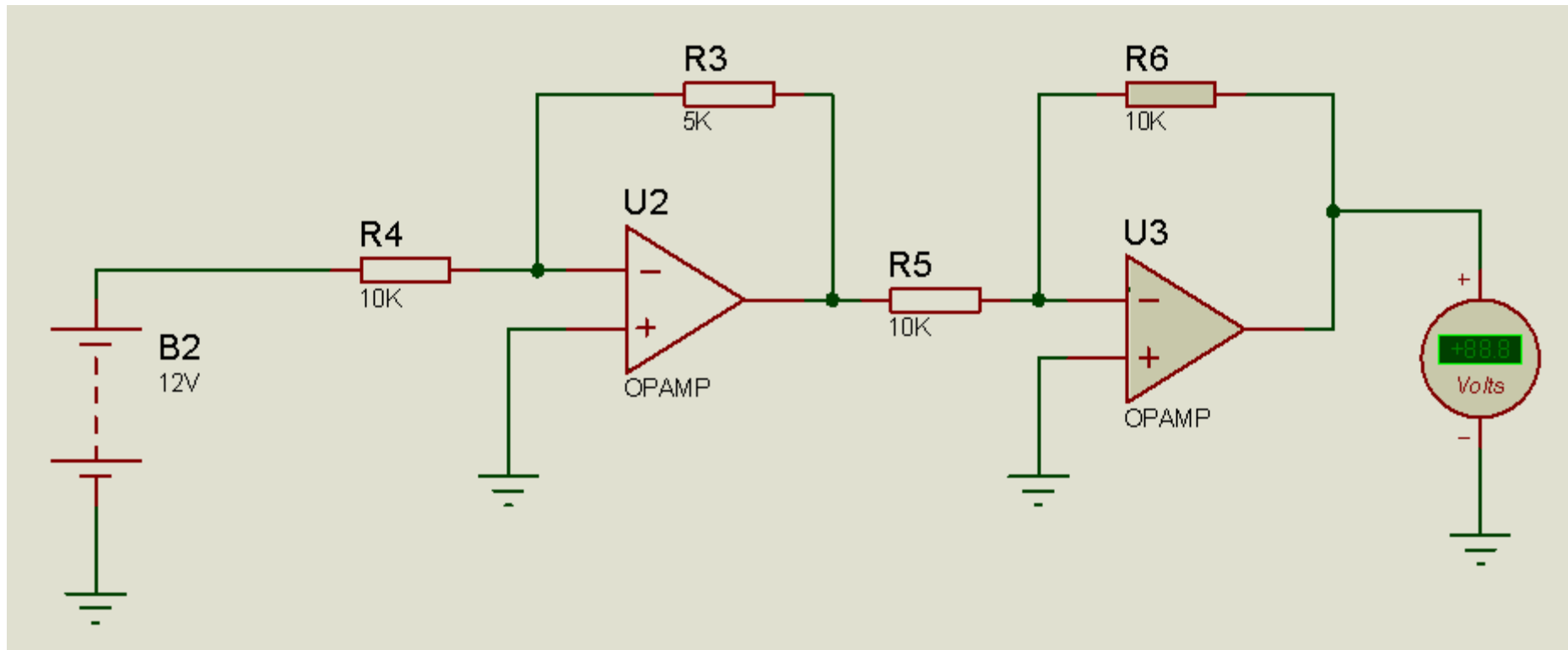


# Operational Amplifier

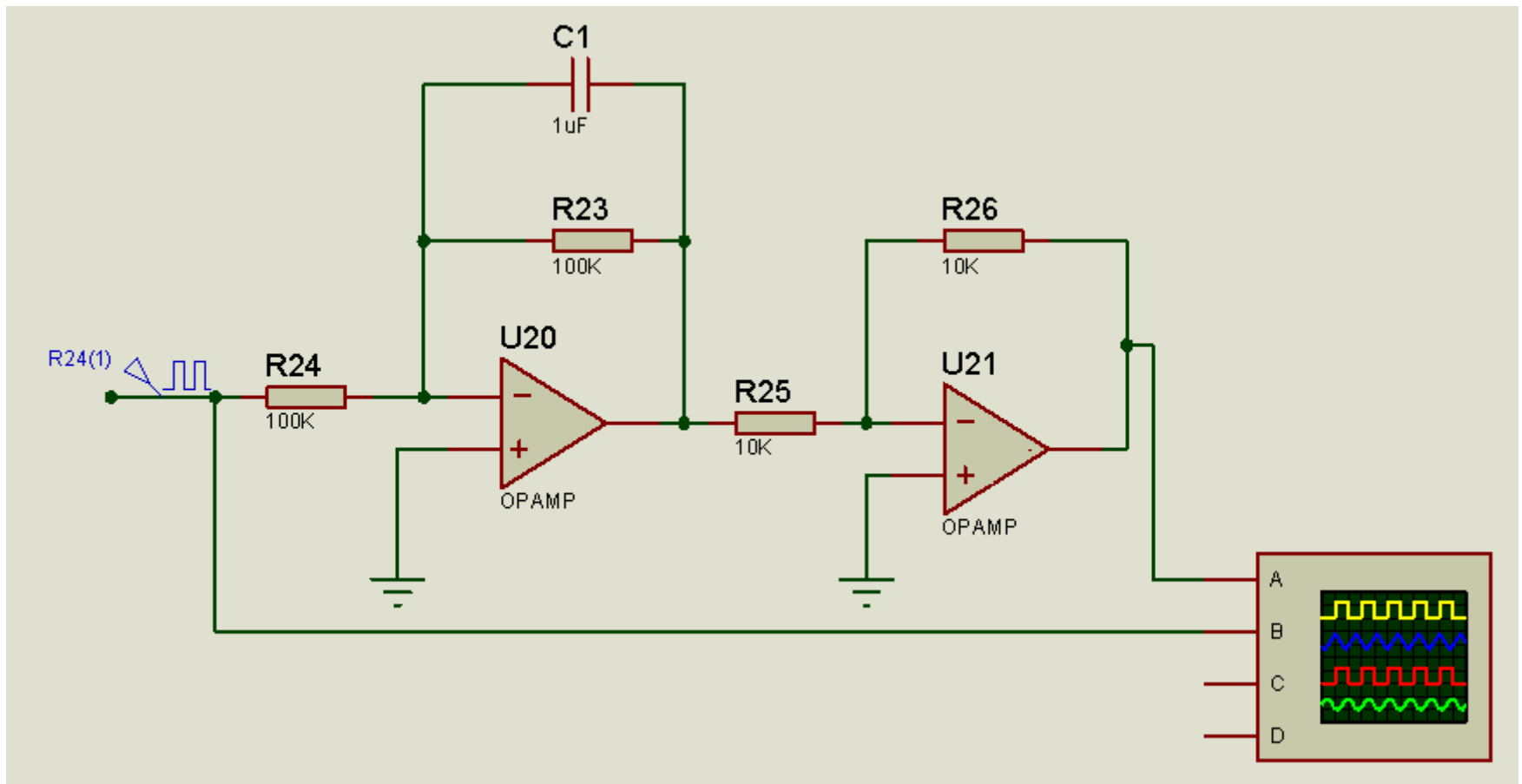
# Inverting Amplifier



# Amplification Only



# Low Pass Filter (Noise Removal)



# Protecting Output Voltage

