BASIC ARDUINO WORKSHOP

Topic : Development Environment & Basic Port Control

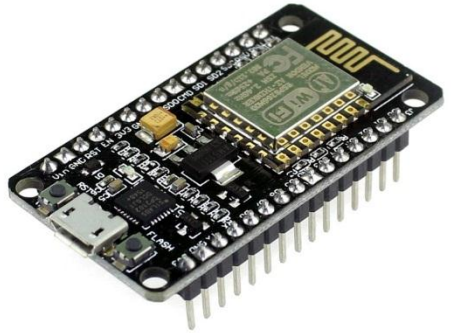
Arduino

- ✓What is Arduino:
  - ✓Open-source electronics platform based on easy-to-use hardware and software

- ✓Why Arduino:
  - ✓Inexpensive
  - ✓Cross-platform
  - ✓Simple and clear programming environment
  - ✓Open source and extensible software / hardware

(https://www.arduino.cc/en/Guide/Introduction)

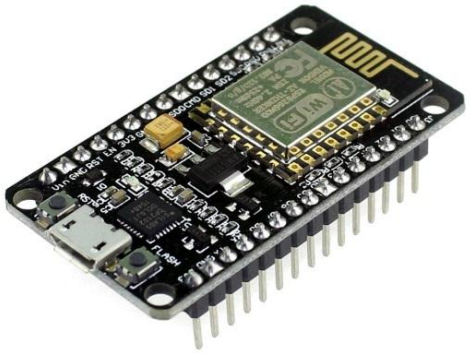
Node MCU 8266



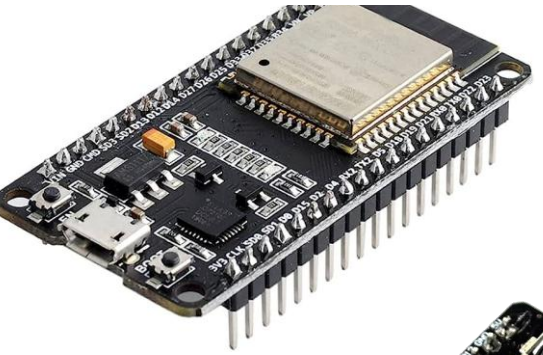
Node MCU Esp32

MCU via IoT Basic Structure

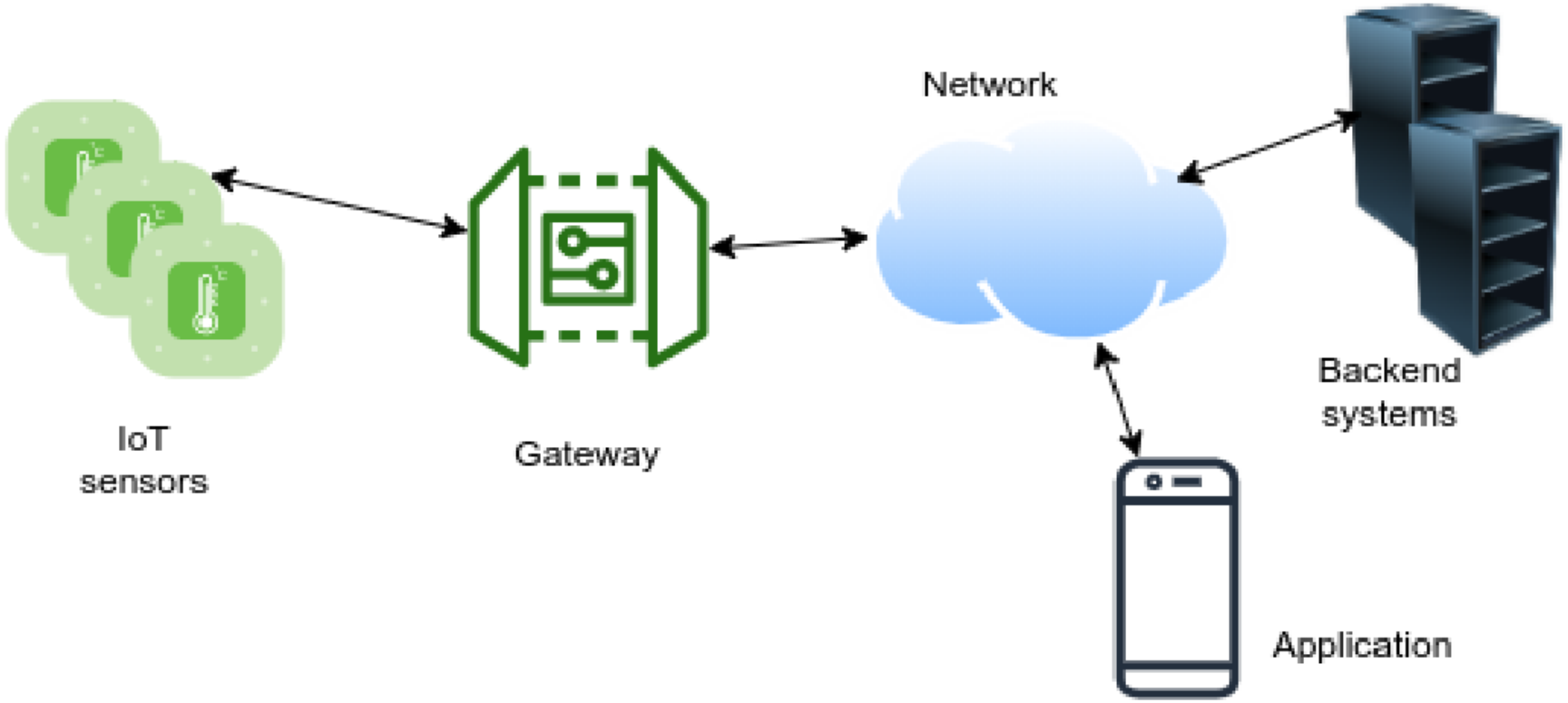
Node MCU 8266



Node MCU Esp32



Esp32 Cam



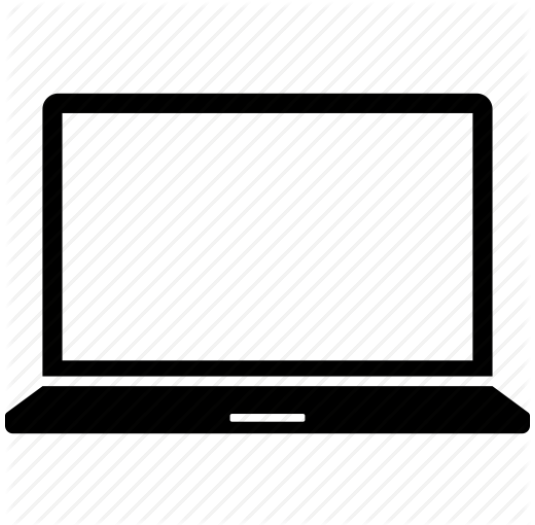
Source: Vedat Ozan Oner - Developing IoT Projects with ESP32_ Automate your home or business with inexpensive Wi-Fi devices-Packt Publishing (2021)

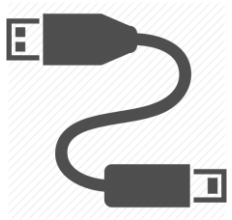
Installation of S/W

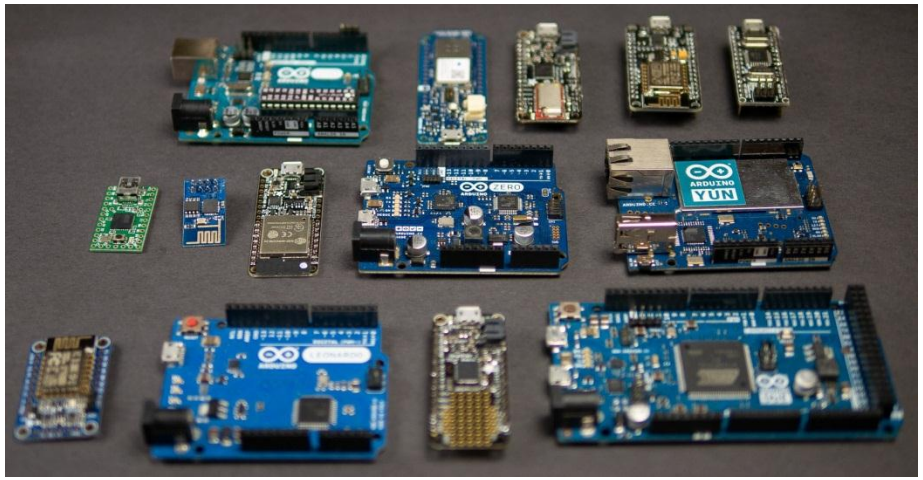
✓Download S/W from website

https://www.arduino.cc/en/Main/Donate

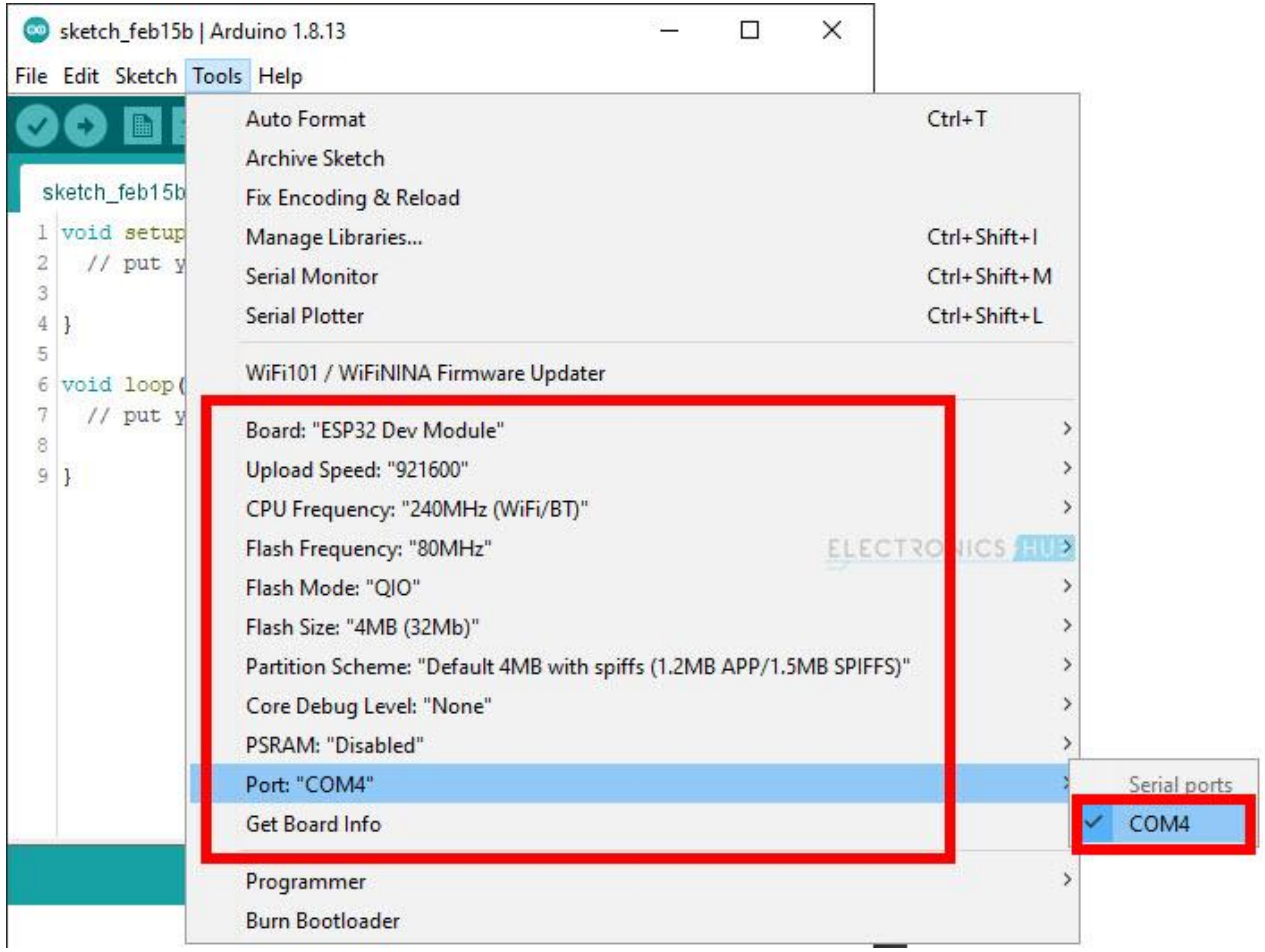
Computer **– USB -** **Controller**







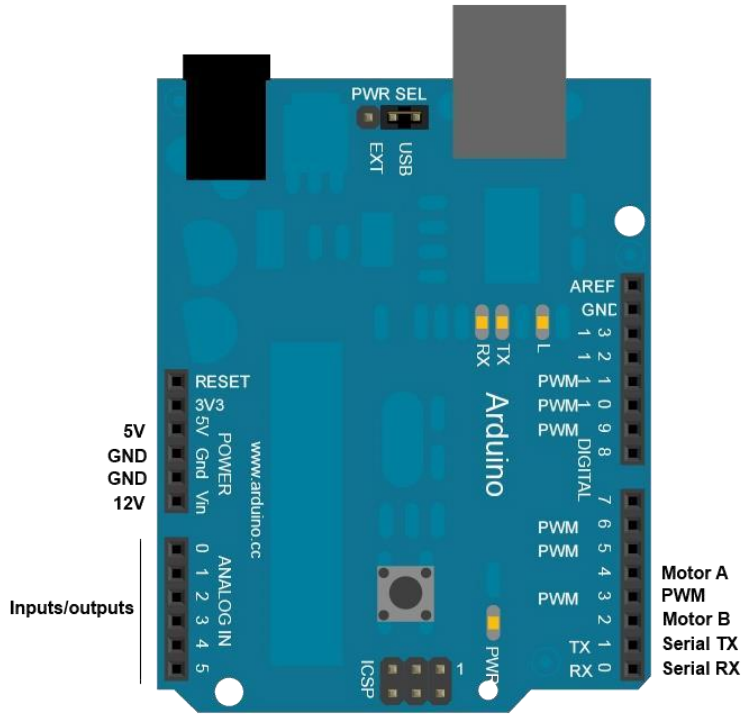
Setup development environment



1. Click on the Tool Tab
2. Board Setting
3. Select “Arduino/ Genuino Uno or ESP 32 or Node MCU 8266”

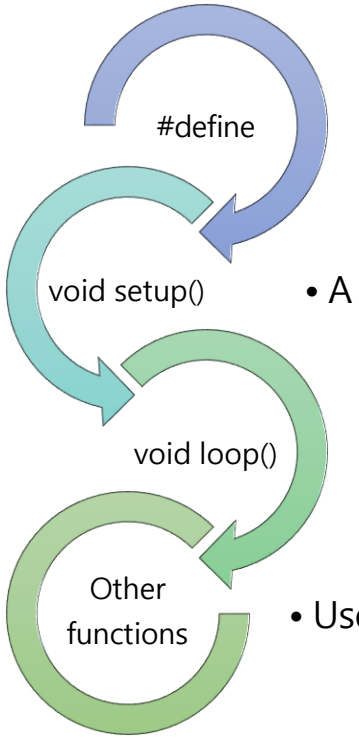
Ports in Arduino

- ✓6 Analog input ports and 13 Digital ports are equipped
  - ✓6 ADC ports
  - ✓13 Digital ports (I/O)
    - ✓6 of PWM ports (#3,5,6,9,10,11)
    - ✓2 ports for serial communication (#0,1)
    - ✓General digital I/O ports (#2,4,7,8,12,13)
  - ✓All ports can be used as general digital I/O ports



Coding Rules and Syntax

✓General form of coding in Arduino;



- Defining header files or external source codes;
- A function for Initiating operating options that runs once when Arduino turns on;
- A function that runs in loop cycles before Arduino turns off;
- User defined functions that can call from void loop() function;

Coding Rules and Syntax

- ✓Pre-defined function in Arduino
  - ✓pinMode(pin, mode)
    - ✓Configures the specified pin to behave either as an input or an output. See the description of digital pins for details on the functionality of the pins.
  - ✓digitalWrite(pin, value)
    - ✓Write a HIGH or a LOW value to a digital pin.
  - ✓digitalRead(pin)
    - ✓Reads the value from a specified digital pin, either HIGH or LOW.

Coding Rules and Syntax

- ✓Pre-defined function in Arduino
  - ✓delay(ms)
    - ✓Pauses the program for the amount of time (in miliseconds) specified as parameter. (There are 1000 milliseconds in a second.)
  - ✓delayMicroseconds(us)
    - ✓Pauses the program for the amount of time (in microseconds) specified as parameter. There are a thousand microseconds in a millisecond, and a million microseconds in a second.
  - ❖Please refer more pre-defined functions on Arduino.cc website.

Coding Rules and Syntax

✓Loop in Arduino coding

while (condition) {}

```
while (var < 200)
{
Var ++;
}
```

For (initialization; condition; increment) {}

```
For (int i=0; I <= 255; i++)
{
analogWrite(PWMpin, i);
delay(10;
}
```

Coding Rules and Syntax

✓If statement in Arduino coding

if (conditional) {}

```
if (someVariable > 50)
{
  // do something here
}
```

switch (variable)

```
switch (var)
{
  case 1:
    //do something when var equals 1
    break;
  case 2:
    //do something when var equals 2
    break;
  default:
    // if nothing else matches, do the default
    // default is optional
Break;
}
```

Coding Rules and Syntax

✓User defined function

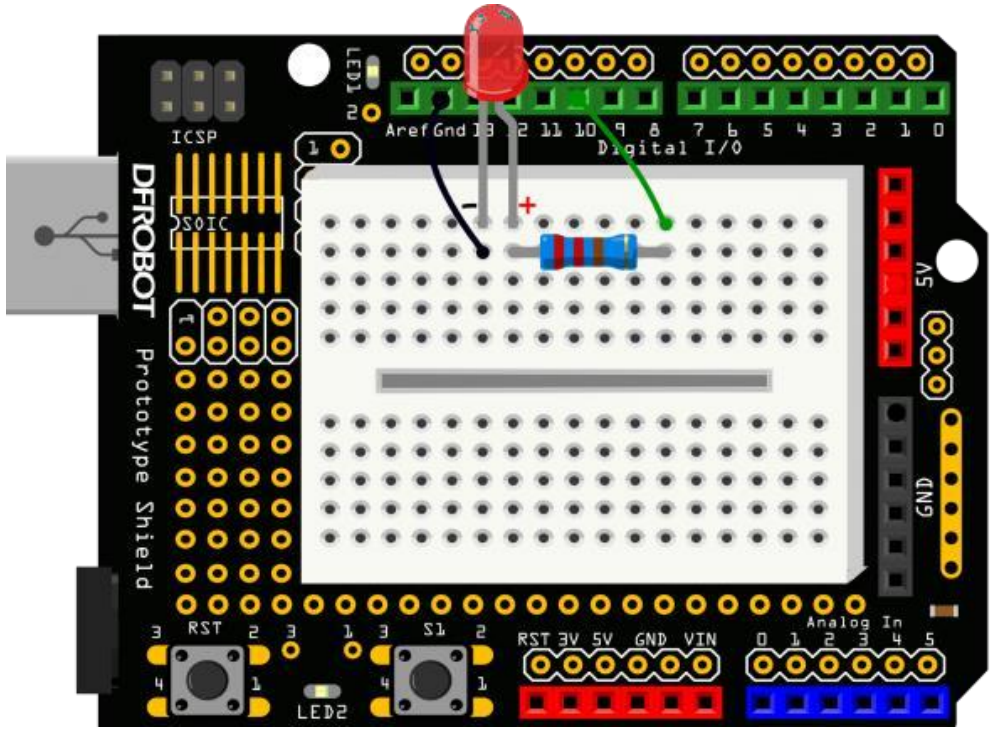
Void func_name_1(val_1, val_n)

```
{
int val_1;
float val_n;
double val_m;

if val_1 > val_n {
Serial.write(val_m);}
}
```

int, float, double, … func_name_n(int val_1, float val_n)

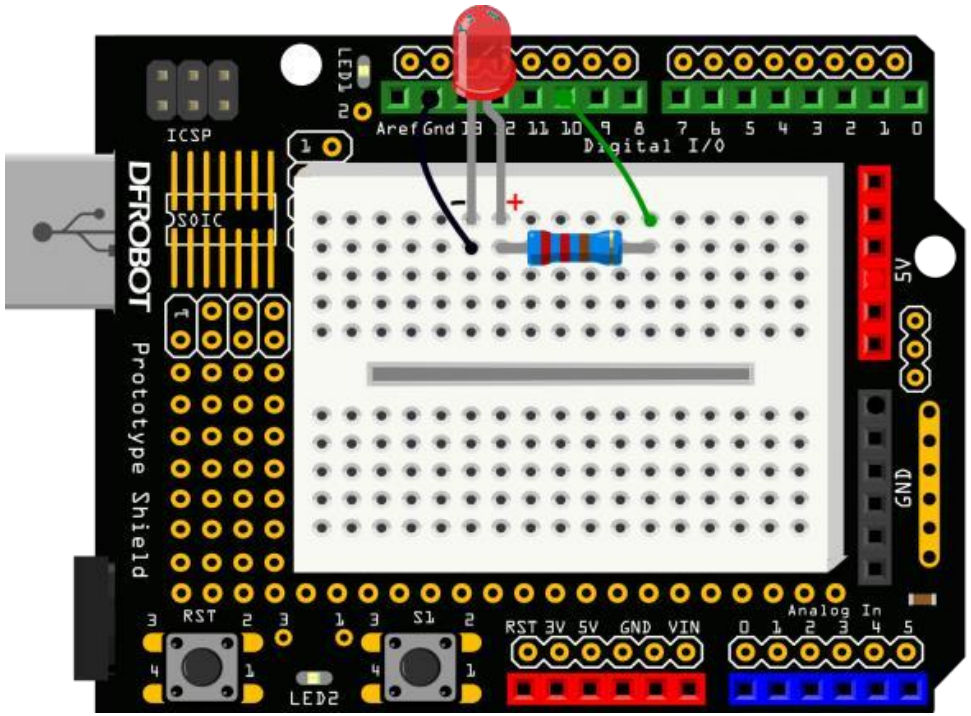
```
{
int val_1;
float val_n;
double val_m;
Int, float, double val_k;

val_k = (int, float, double)val_1 + (int, float, double)val_n +val_m;
Return val_k;
}
```

Practice: a) Single LED Dimming



```
int ledPin = 10;
void setup() {

      pinMode(ledPin, OUTPUT);
}
void loop() {
      digitalWrite(ledPin,HIGH);
      delay(1000);
      digitalWrite(ledPin,LOW);
      delay(1000);
}
```

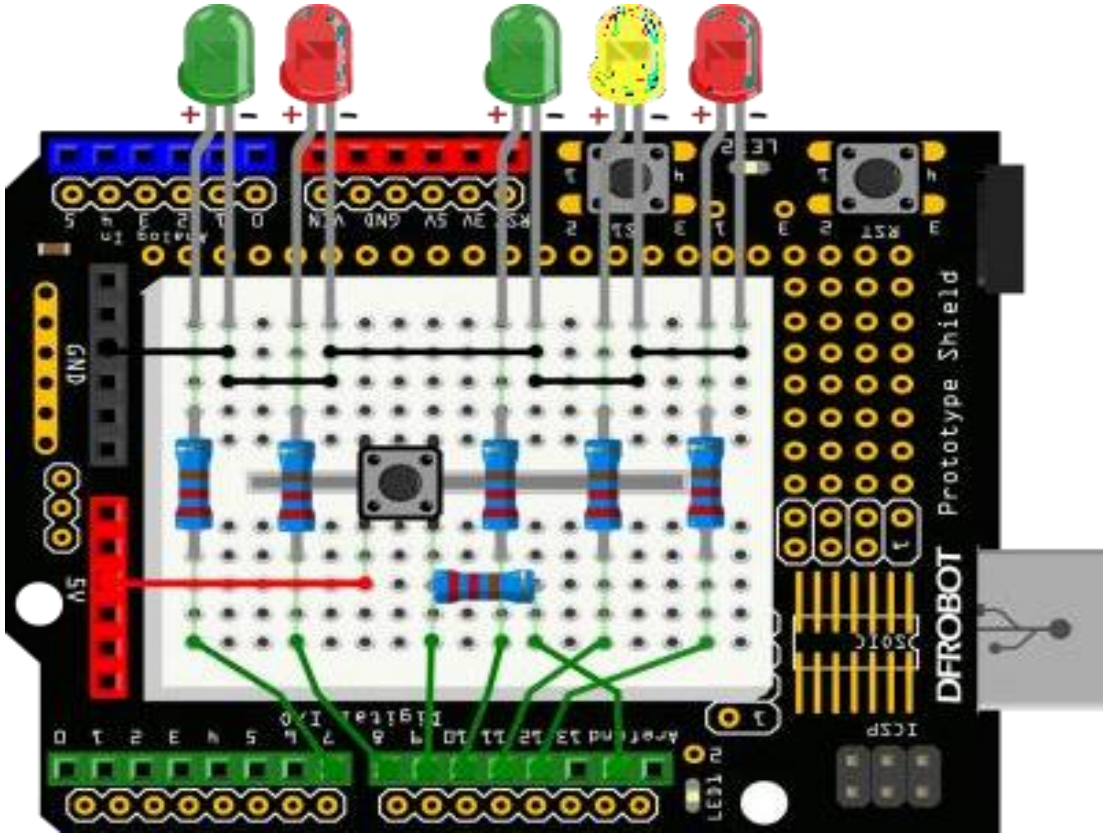
Practice: b-1) SOS Beacon



```
 ledPin = 10;
void setup() {
        pinMode(ledPin, OUTPUT);
}
void loop() {

      // S(...) three dot
       for(int x=0;x<3;x++){
        digitalWrite(ledPin,HIGH);
        delay(150);
        digitalWrite(ledPin,LOW);
        delay(100);
        }
```

```
            delay(100);
      // O(---) three dash
       for(int x=0;x<3;x++){
        digitalWrite(ledPin,HIGH);
        delay(400);
        digitalWrite(ledPin,LOW);
        delay(100);
        }

        delay(100);

      //S(...) three dot
       for(int x=0;x<3;x++){
        digitalWrite(ledPin,HIGH);
        delay(150);
        digitalWrite(ledPin,LOW);
        delay(100);
        }

        delay(5000);
}
```

Practice: b-2) Traffic Light



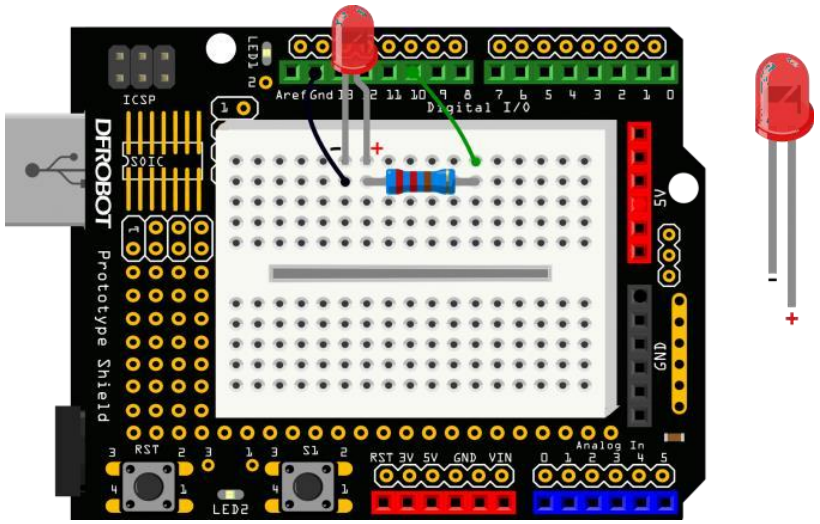
Practice: b-2) Traffic Light

```
int carRed = 12; //assign the car lights
int carYellow = 11;
int carGreen = 10;
int button = 9; //button pin
int pedRed = 8; //assign the pedestrian lights
int pedGreen = 7;
int crossTime =5000; //time for pedestrian to cross
unsigned long changeTime; //time since button pressed

void setup() {
    pinMode(carRed, OUTPUT);
    pinMode(carYellow, OUTPUT);
    pinMode(carGreen, OUTPUT);
    pinMode(pedRed, OUTPUT);
        pinMode(pedGreen, OUTPUT);
        pinMode(button, INPUT);
        digitalWrite(carGreen, HIGH); //turn on the green
lights
    digitalWrite(pedRed, HIGH);
}
```

```
    void loop() {
    int state = digitalRead(button);
        //check if button is pressed and it is over 5 seconds
since last button press
        if(state == HIGH && (millis() - changeTime)> 5000){
            //call the function to change the lights
            changeLights();
        }
}

void changeLights() {
    digitalWrite(carGreen, LOW); //green off
    digitalWrite(carYellow, HIGH); //yellow on
    delay(2000); //wait 2 seconds

    digitalWrite(carYellow, LOW); //yellow off
    digitalWrite(carRed, HIGH); //red on
        delay(1000); //wait 1 second till its safe

    digitalWrite(pedRed, LOW); //ped red off
    digitalWrite(pedGreen, HIGH); //ped green on

    delay(crossTime); //wait for preset time period
```

```
  //flash the ped green
        for (int x=0; x<10; x++) {
            digitalWrite(pedGreen, HIGH);
        delay(250);
        digitalWrite(pedGreen, LOW);
        delay(250);
         }

        digitalWrite(pedRed, HIGH); //turn ped red on
    delay(500);

    digitalWrite(carRed, LOW); //red off
    digitalWrite(carYellow, HIGH); //yellow on
    delay(1000);
    digitalWrite(carYellow, LOW); //yellow off
        digitalWrite(carGreen, HIGH);

    changeTime = millis(); //record the time since last
change of lights
    //then return to the main program loop
}
```

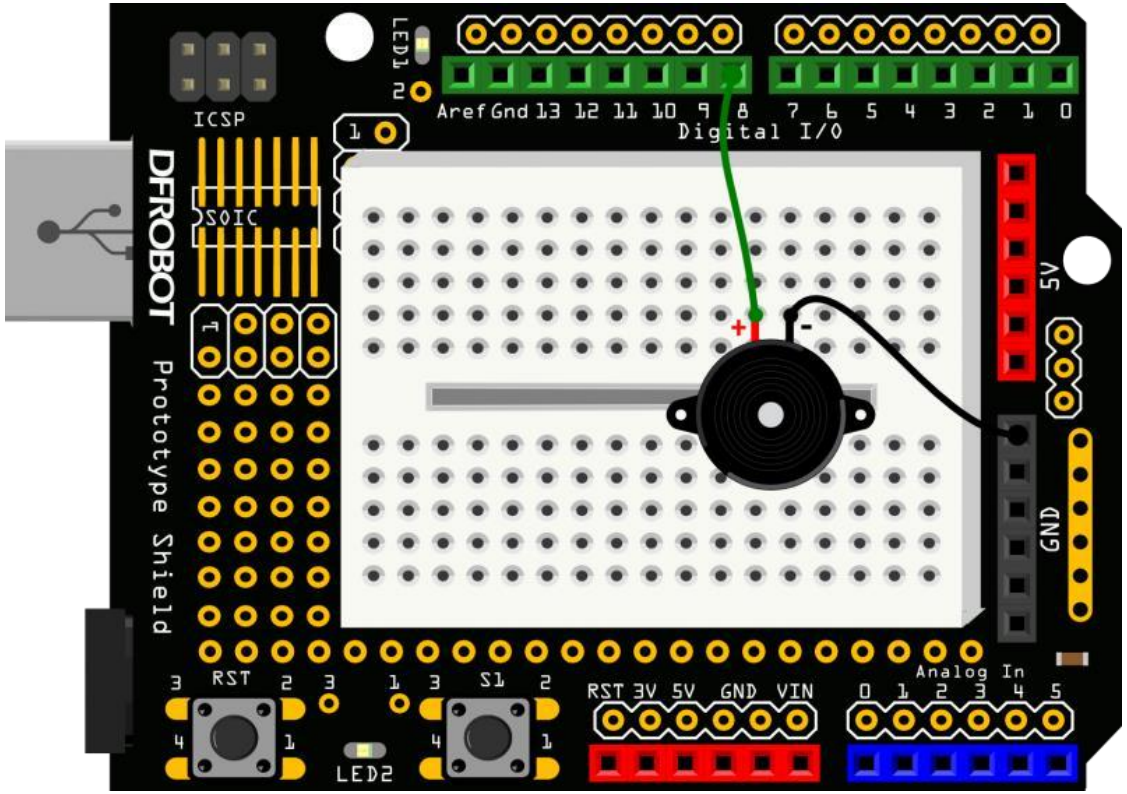
Practice: b-3) Fading Light




```
int ledPin = 10;       // the pin that the LED is attached to

void setup() {
      // declare pin 9 to be an output:
      pinMode(ledPin,OUTPUT);
      // initialize serial communication at 9600 bits per
second:
      Serial.begin(9600);
}

void loop(){

      fadeOn(1000,5);
      fadeOff(1000,5);

}
```

```
void fadeOn(unsigned int time,int increament){

     //change the brightness by FOR statement
    for (byte value = 0 ; value < 255;
value+=increament){
                // print out the value:
            Serial.println(value);
                // set the brightness of pin 10:
        analogWrite(ledPin, value);
        delay(time/(255/5));
        }
}

void fadeOff(unsigned int time,int decreament){
       //change the brightness by FOR statement
    for (byte value = 255; value >0; value-
=decreament){
            Serial.println(value);
        analogWrite(ledPin, value);
        delay(time/(255/5));
    }
}
```

Practice: b-4). Alarm



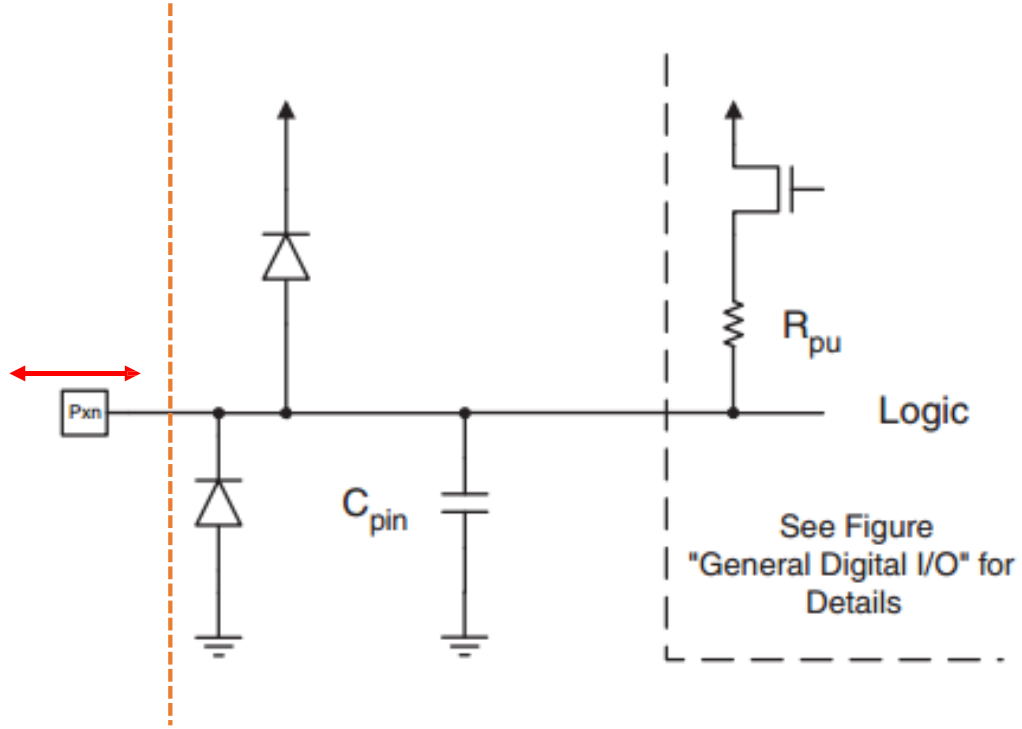

```
float sinVal;
int toneVal;

void setup(){
    pinMode(8, OUTPUT);
}

void loop(){
    for(int x=0; x<180; x++){
        // convert degrees to radians then obtain value
        sinVal = (sin(x*(3.1412/180)));
        // generate a frequency from the sin value
        toneVal = 2000+(int(sinVal*1000));
        tone(8, toneVal);
        delay(2);
    }
}
```

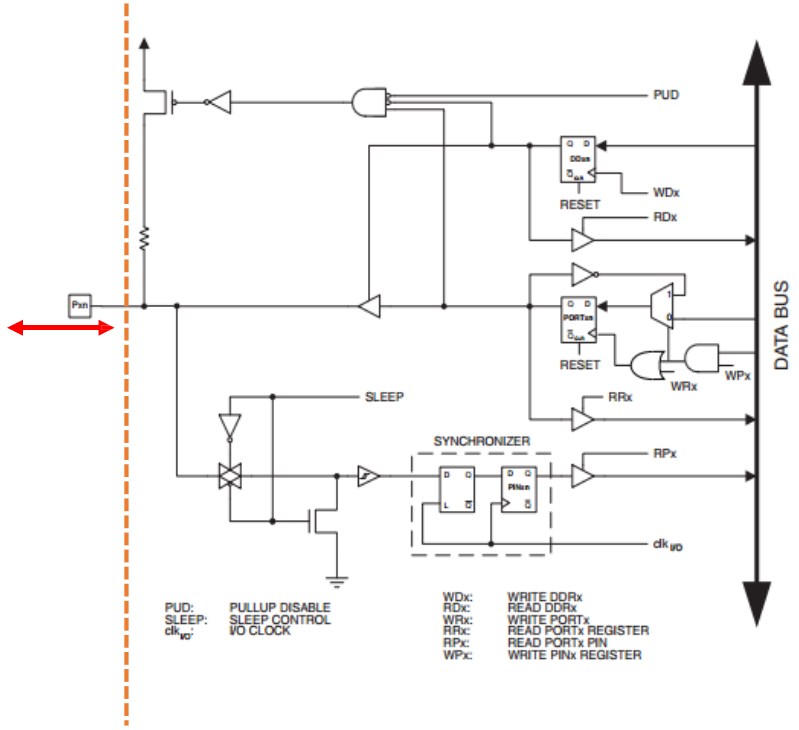
Appendix

- ✓Basic data type in C programming [https://en.wikipedia.org/wiki/C_data_types]
  - ✓Char
    - ✓1 byte for a character (8bits)
  - ✓Int
    - ✓Integer (32 bit)
  - ✓Float
    - ✓Single-precision floating point number (real-number)
  - ✓Double
    - ✓Double-precision floating point number (real-number)

Appendix

✓Principle of Port I/O





References

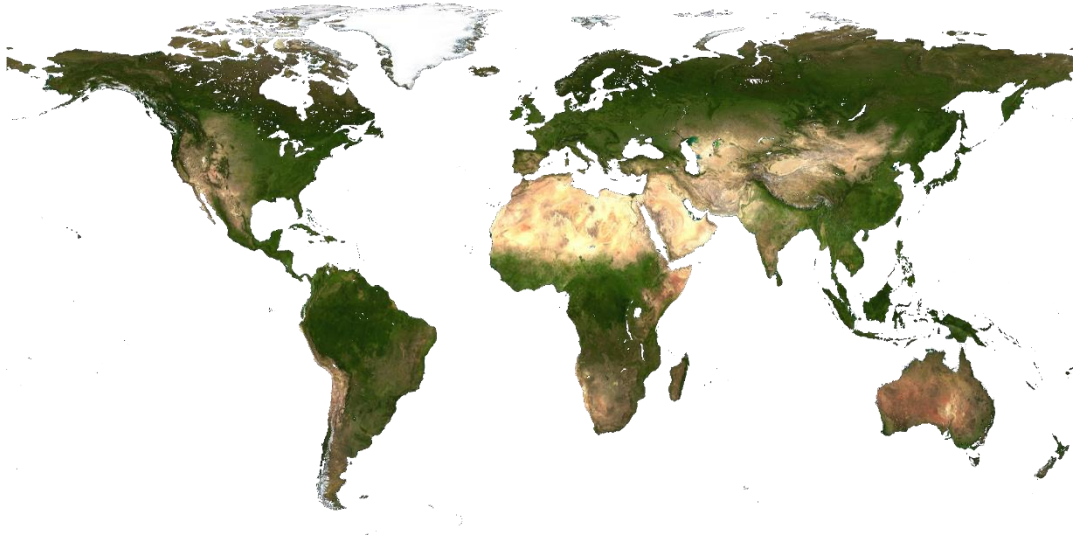
- ✓https://www.arduino.cc/
- ✓Chapter 14. I/O-Ports, Atmega328P datasheet
  - ✓http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
- ✓Transistor as a switch
  - ✓https://electrosome.com/transistor-as-a-switch/
- ✓https://www.arduino.cc/en/Reference/HomePage
- ✓https://www.arduino.cc/
- ✓http://www.dfrobot.com/wiki/index.php/DFRduino_Beginner_Kit_For_Arduino_V3_SKU:DFR0100

ARDUINO WORKSHOP Lv. 2

Topic : Advanced Port control and
ADC (Analog to Digital Converter)

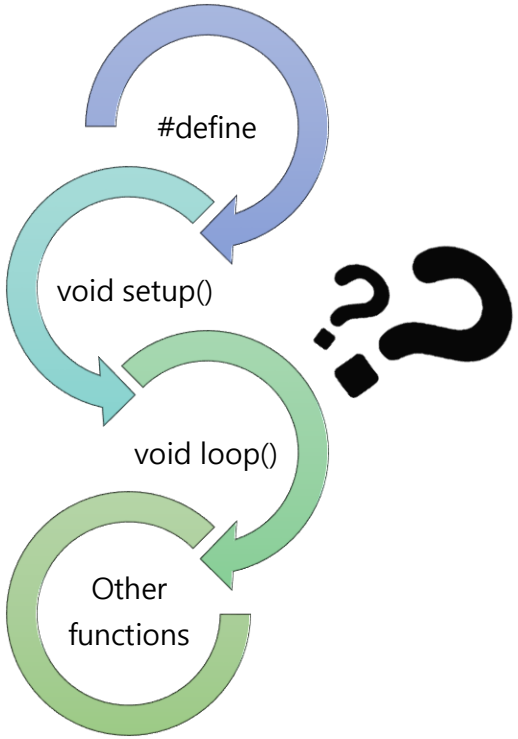
Timer and ADC

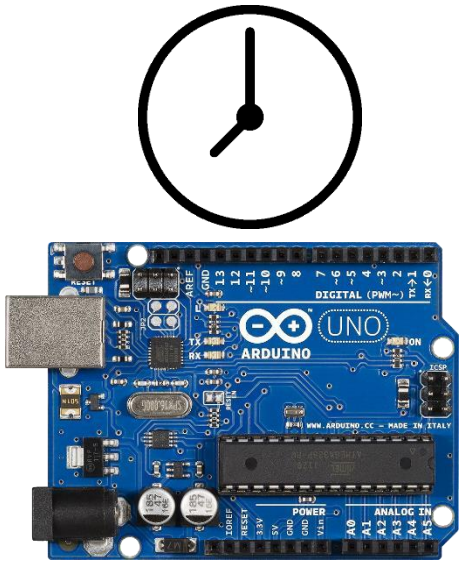
- ✓Motivation
  - ✓Nature is not digitalized but analog and change in real-time.



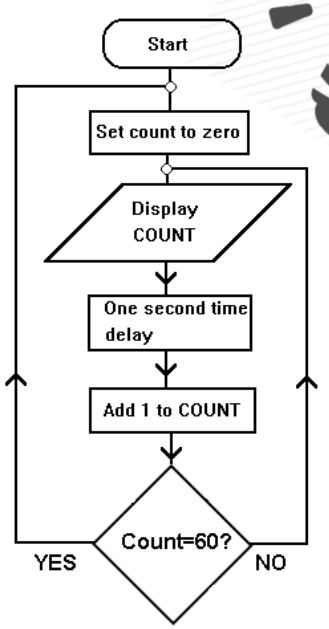
Timer and ADC

- ✓Timer
  - ✓Calculating time for precision real-time control.





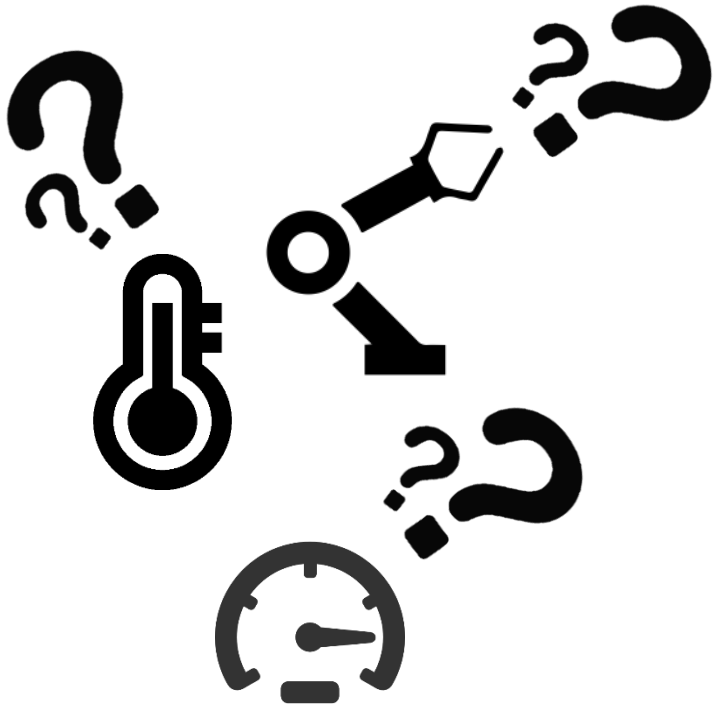
Internal clock in microprocessor



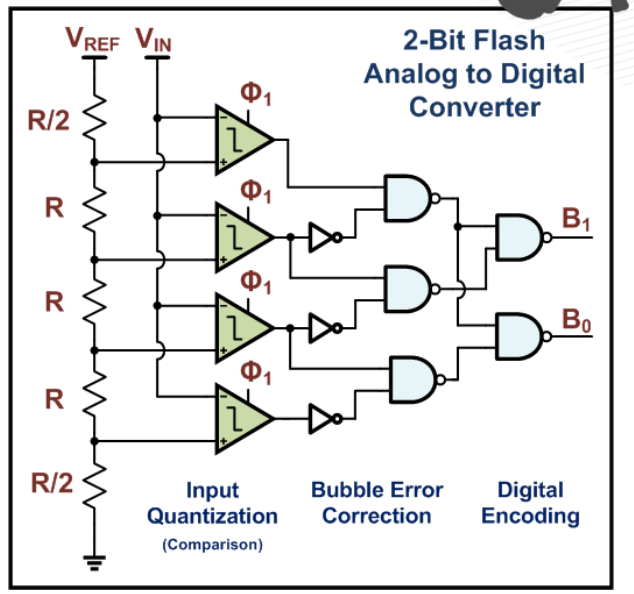
1 MINUTE TIMER

Timer and ADC

- ✓ADC
  - ✓Measuring various phenomenon from nature.

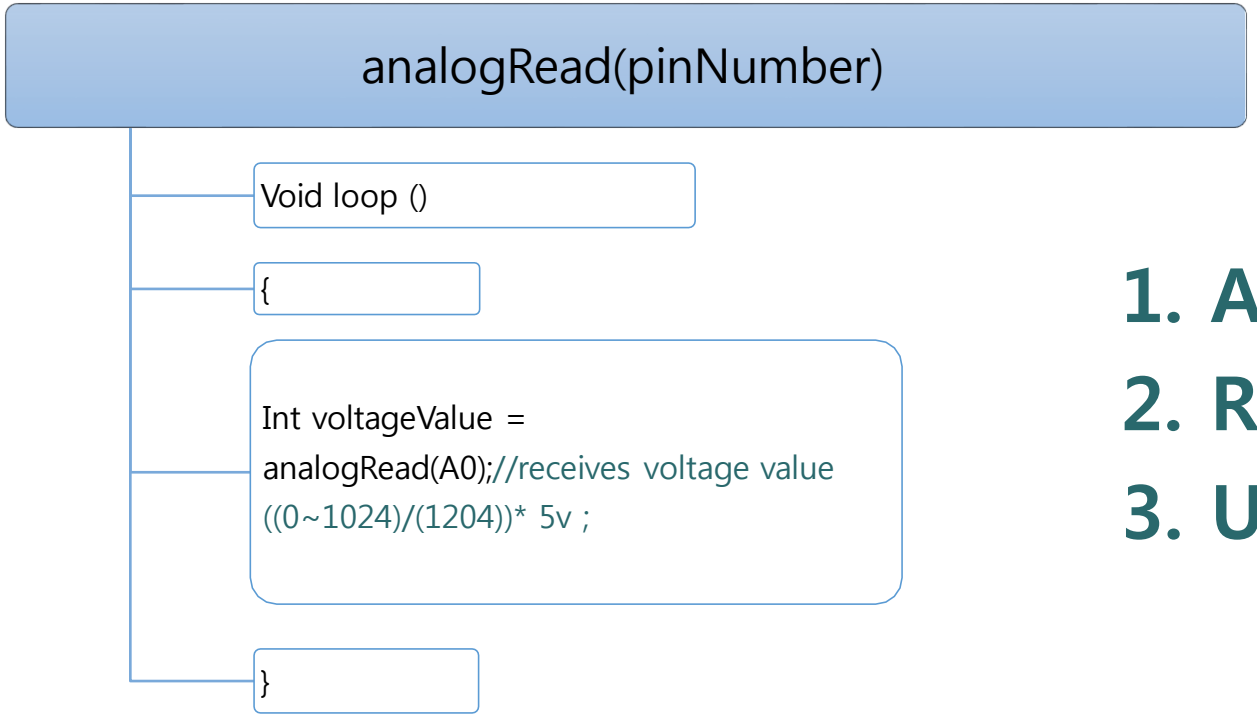


Sensors
(Transducer)



ADC

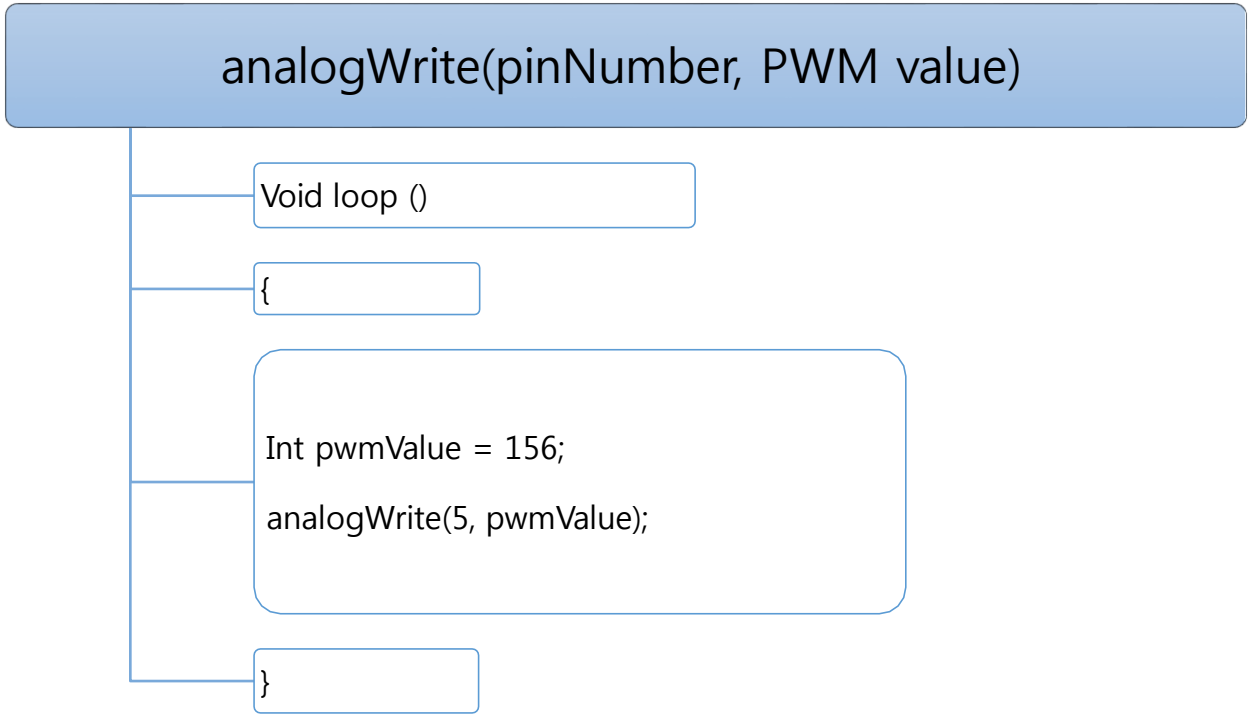
✓Using the analogRead function



1. Analog Pin #: A0~A5
2. Receives Voltage Value(0~1023)
3. Used with a variety of sensors

ADC

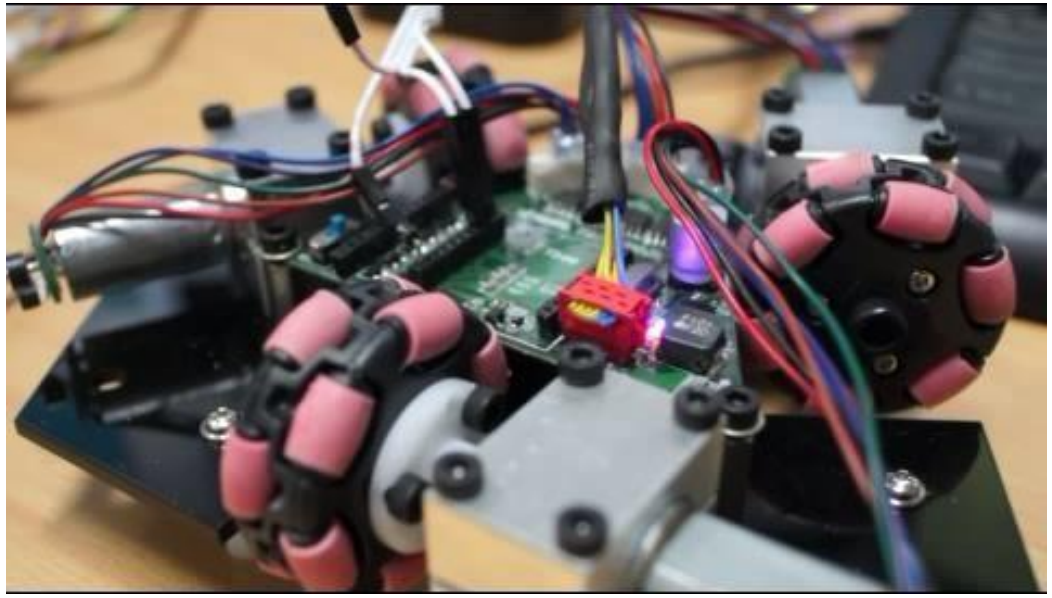
✓Using the analogWrite function

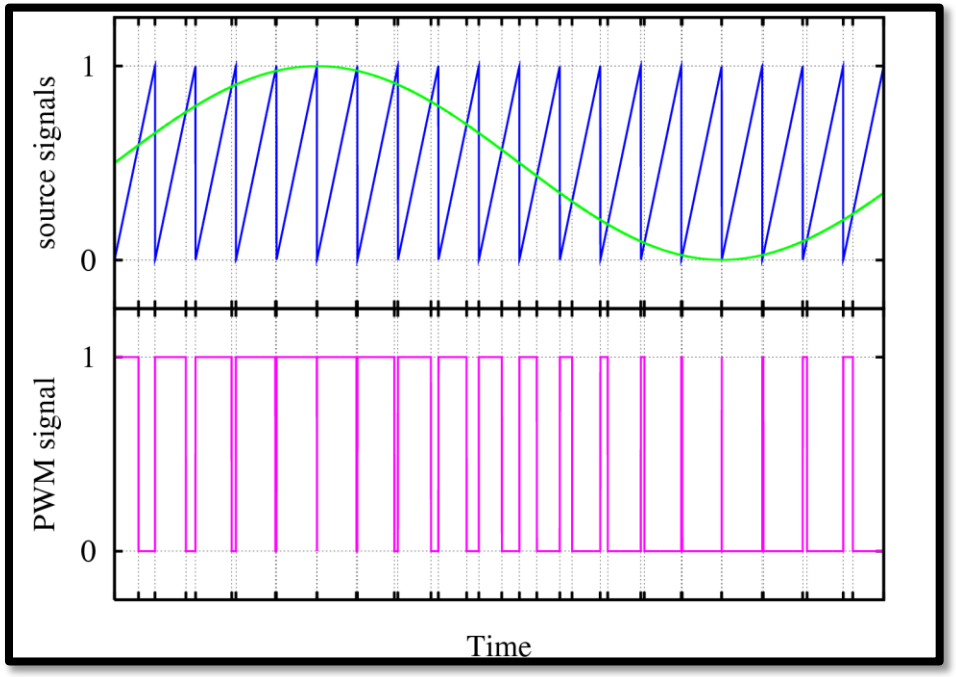


1. PWM Pin #: 3,5,6,9,10,11
2. PWM Value: 0~255
3. Average Voltage Control

Application

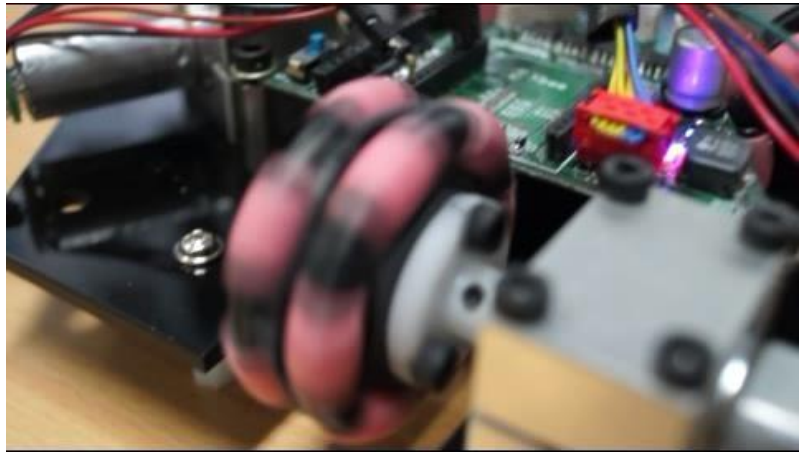
✓Pulse width modulation (Timer)

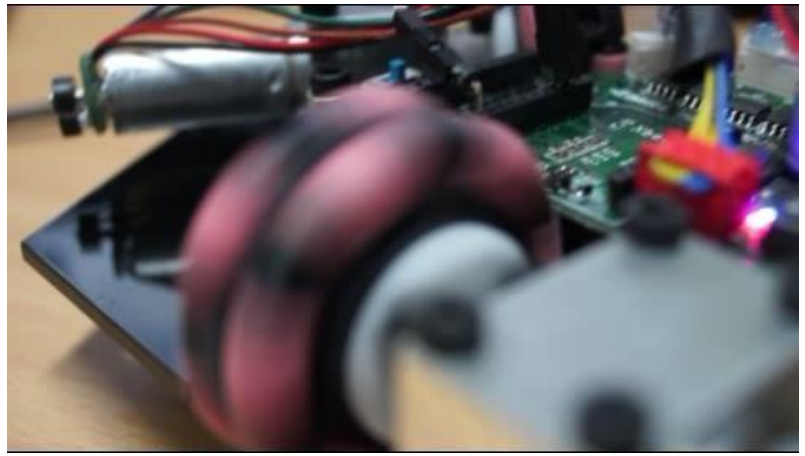


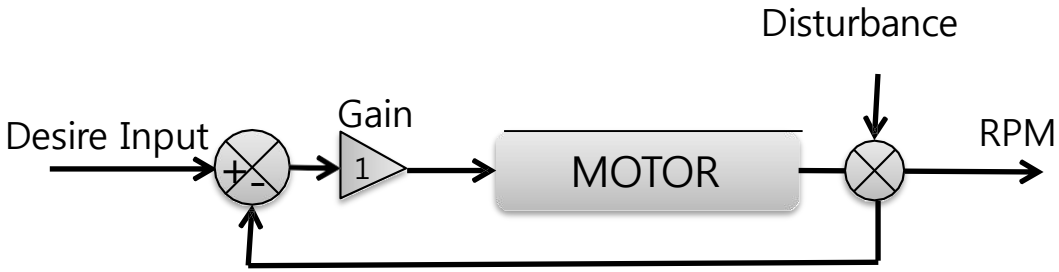


Application

✓Motor control (Feedback control)

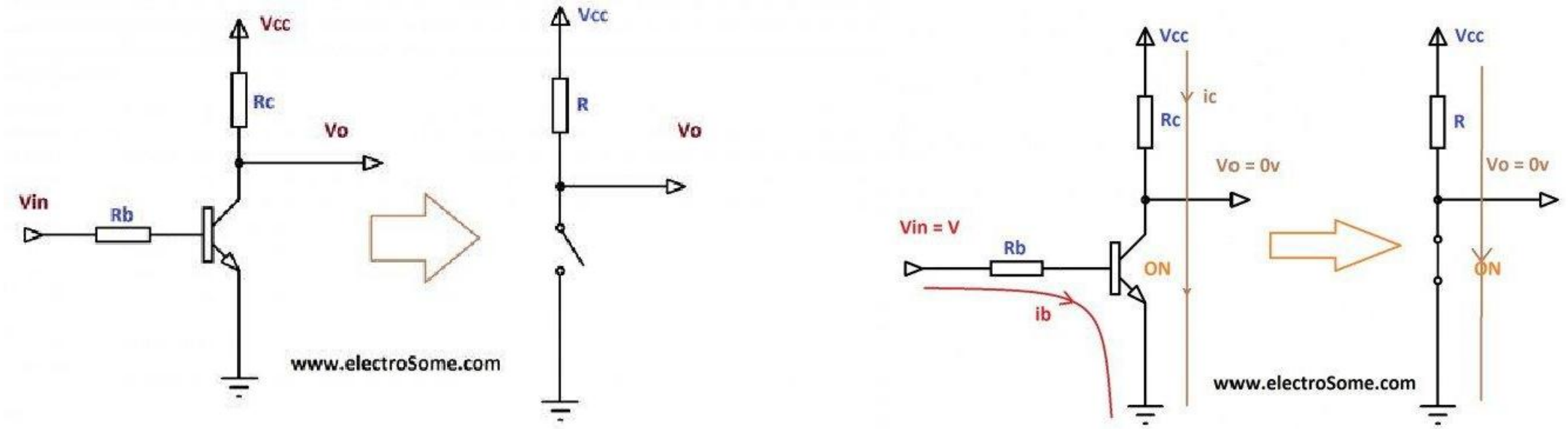




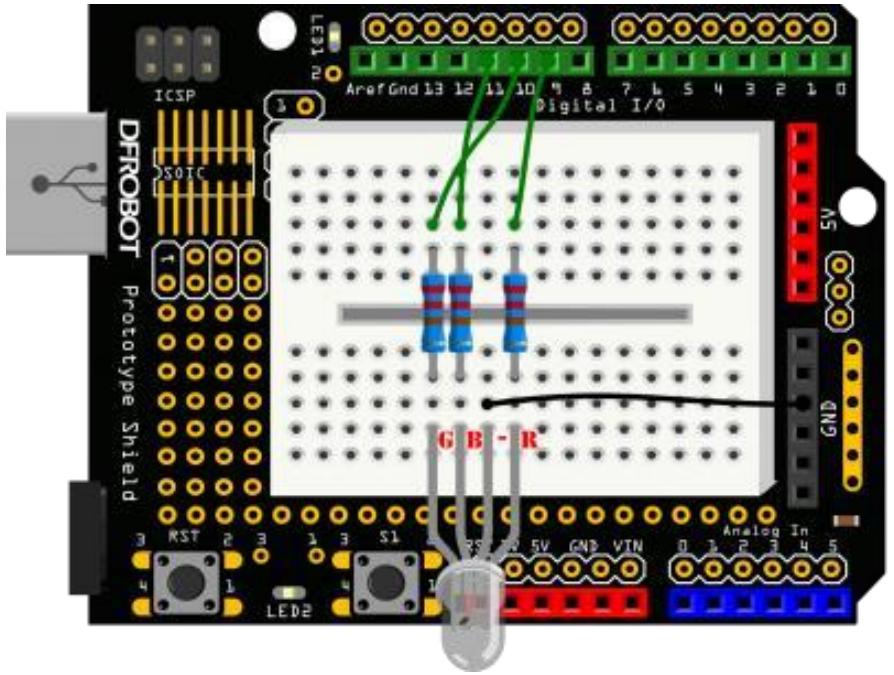


Driving circuit for higher current

✓Transistor as switch

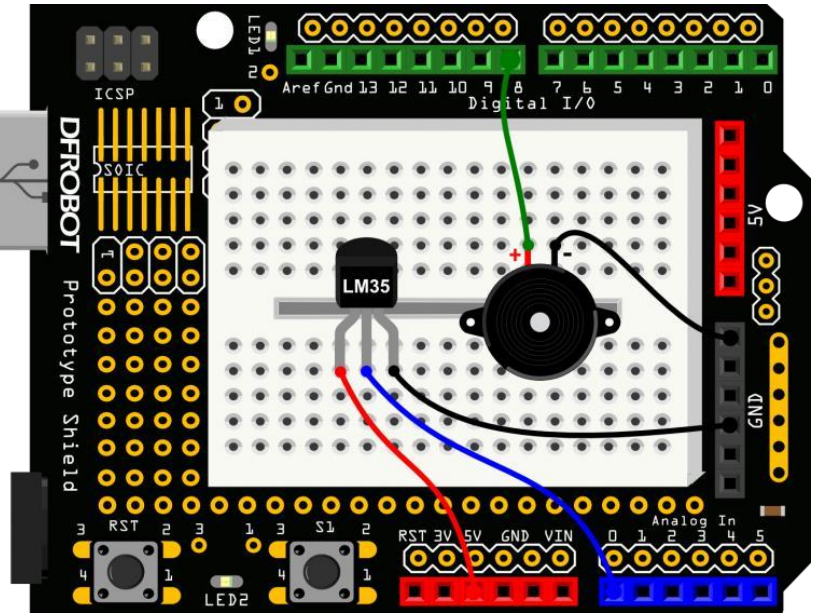


Practice: a) RGB LED Control



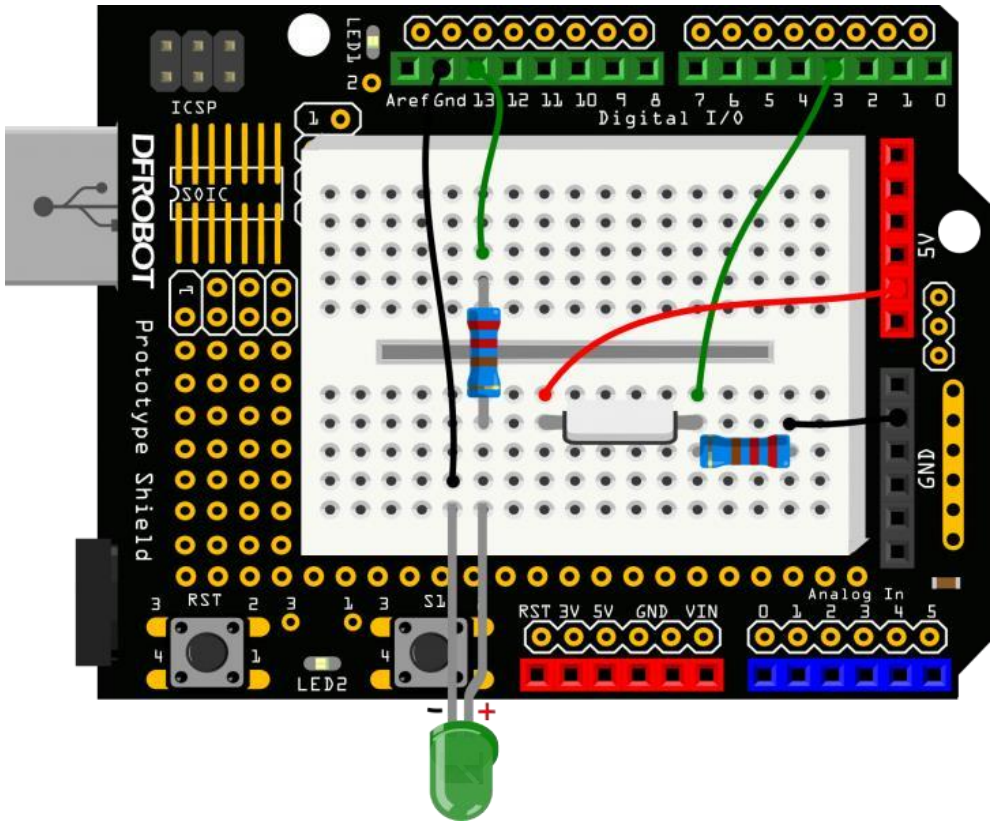

```
tnt redPin = 9;    // the pin that the red LED is attached to
int greenPin = 10; // the pin that the green LED is attached to
int bluePin = 11;  // the pin that the blue LED is attached to
 void setup(){
      pinMode(redPin, OUTPUT);
      pinMode(greenPin, OUTPUT);
      pinMode(bluePin, OUTPUT);

}
void loop(){
    // call the function to change the colors of LED randomly.
    colorRGB(random(0,255),random(0,255),random(0,255));
//R:0-255 G:0-255 B:0-255
    delay(1000);
}

void colorRGB(int red, int green, int blue){
      analogWrite(redPin,constrain(red,0,255));
      analogWrite(greenPin,constrain(green,0,255));
      analogWrite(bluePin,constrain(blue,0,255));
}
```

Practice: b-1) Temperature Alarm



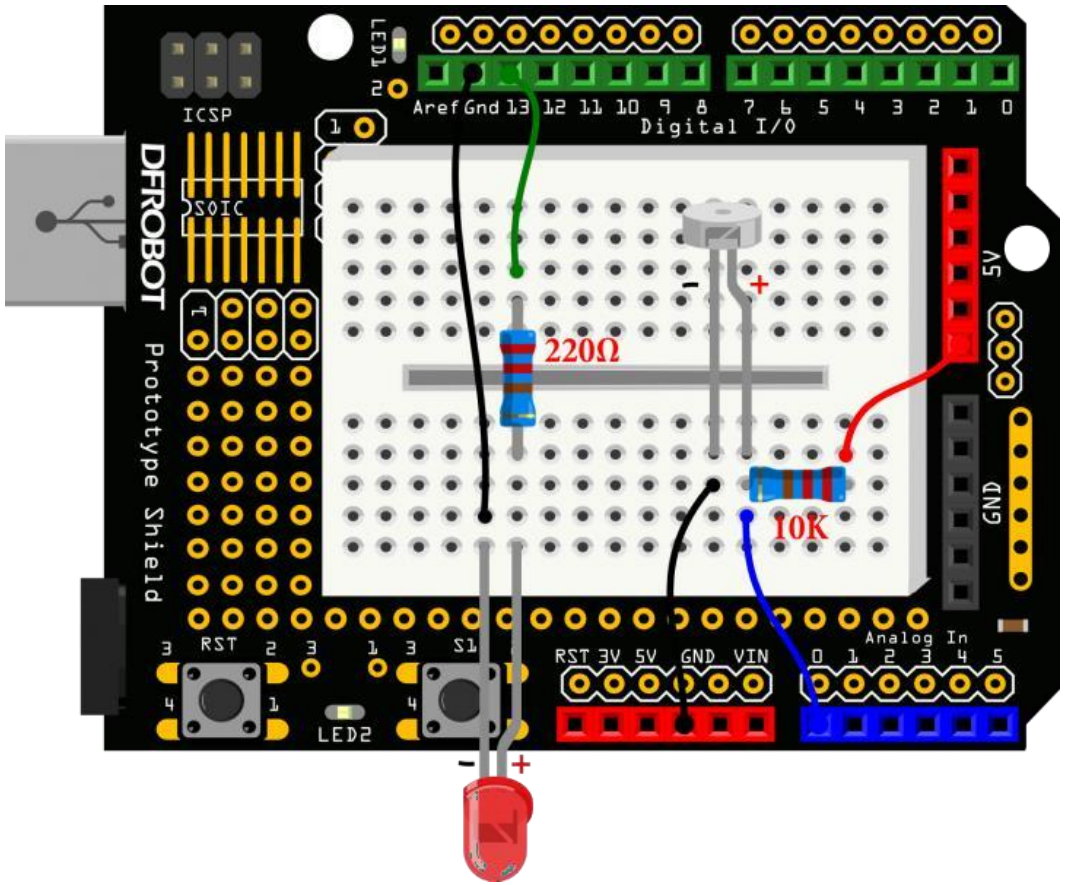

```
float sinVal;
int toneVal;
unsigned long tepTimer ;

void setup(){
    pinMode(8, OUTPUT);
    Serial.begin(9600);
}

void loop(){
    int val;
    double data;
    val=analogRead(0);
    data = (double) val * (5/10.24);  // convert the voltage to temperture

    if(data>27){        // If the temperture is over 27 degree, buzzer will alarm.
          for(int x=0; x<180; x++){
            sinVal = (sin(x*(3.1412/180)));
            toneVal = 2000+(int(sinVal*1000));
            tone(8, toneVal);
            delay(2);
      }
} else {     // If the temperturn is below 27 degree, buzzer will not alarm
          noTone(8);
    }

    if(millis() - tepTimer > 500){      // output the temperture value per 500ms
          tepTimer = millis();
          Serial.print("temperature: ");
          Serial.print(data);
          Serial.println("C");
       }
}
```

Practice: b-2) Detecting Vibration




```
int SensorLED = 13;        //LED PIN
int SensorINPUT = 3;       //Connect the sensor to digital
Pin 3 which is Interrupts 1
unsigned char state = 0;
 void setup() {
  pinMode(SensorLED, OUTPUT);
  pinMode(SensorINPUT, INPUT);

  // Trigger the blink function when the falling edge is
detected
  attachInterrupt(1, blink, RISING);
 }

void loop(){
      if(state!=0){
        state = 0;
        digitalWrite(SensorLED,HIGH);
        delay(500);
      }
      else
        digitalWrite(SensorLED,LOW);
}
 void blink(){              //Interrupts function
    state++;
}
```

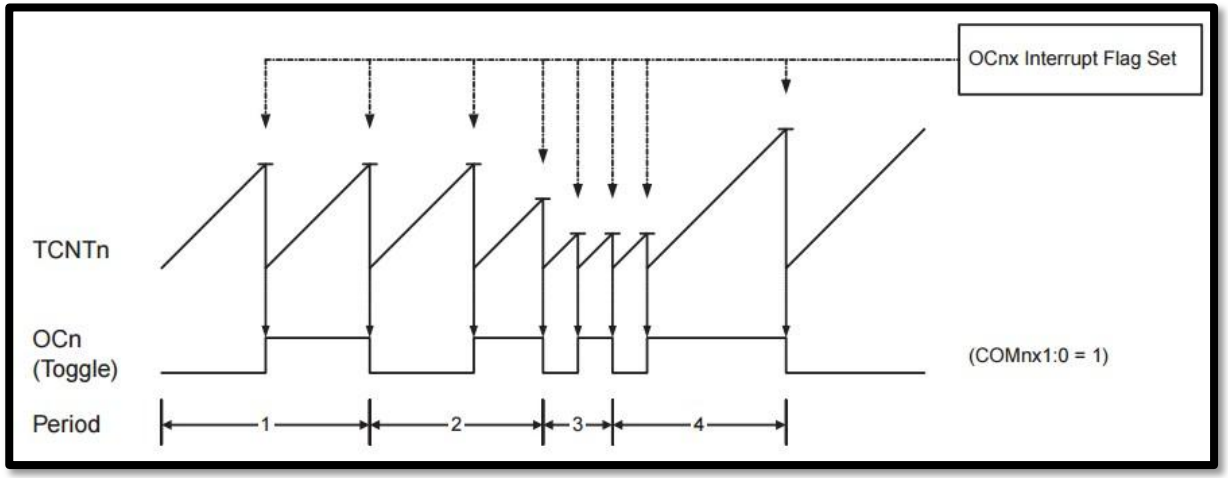
Practice: b-3) Auto Light

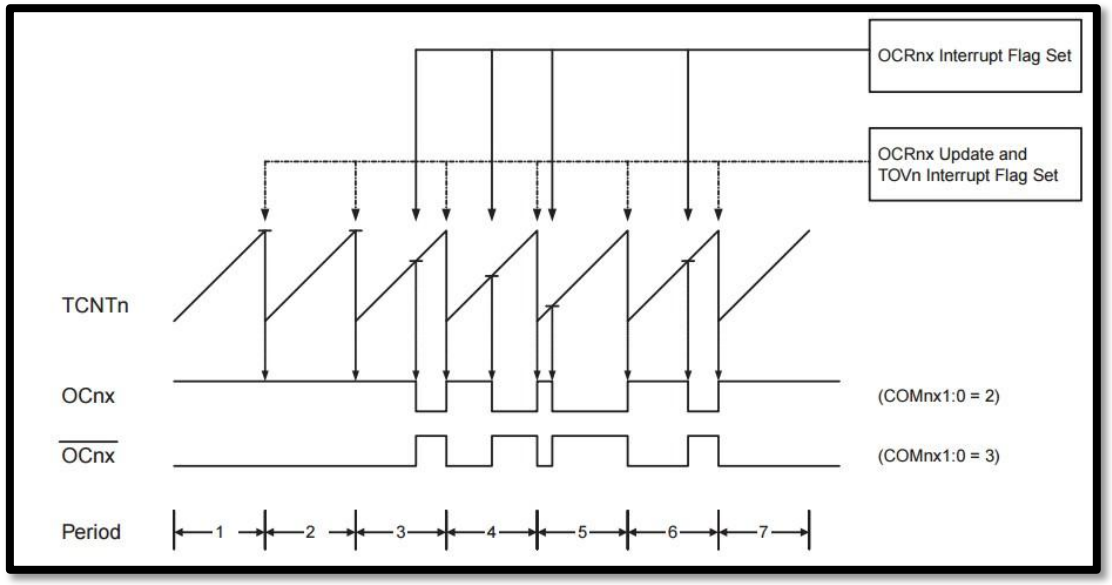



```
int LED = 13;                    //Led pin
int val = 0;

void setup(){
    pinMode(LED,OUTPUT);
    Serial.begin(9600);
}

void loop(){
    val = analogRead(0);         // read voltage value
    Serial.println(val);
    if(val<1000){                // if the value is less than
1000 , LED turns off
        digitalWrite(LED,LOW);
    }else{                       // if the value is more than
1000 , LED turns on
        digitalWrite(LED,HIGH);
    }
    delay(10);
}
```

Appendix: Timer and ADC

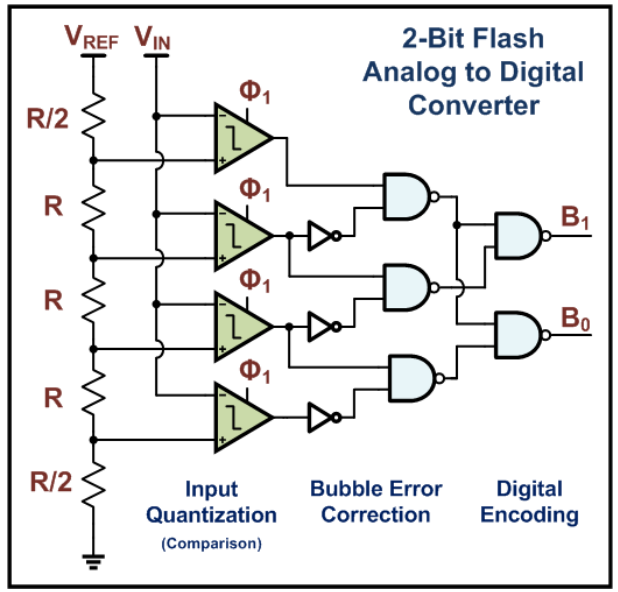
✓Working principle of Timer

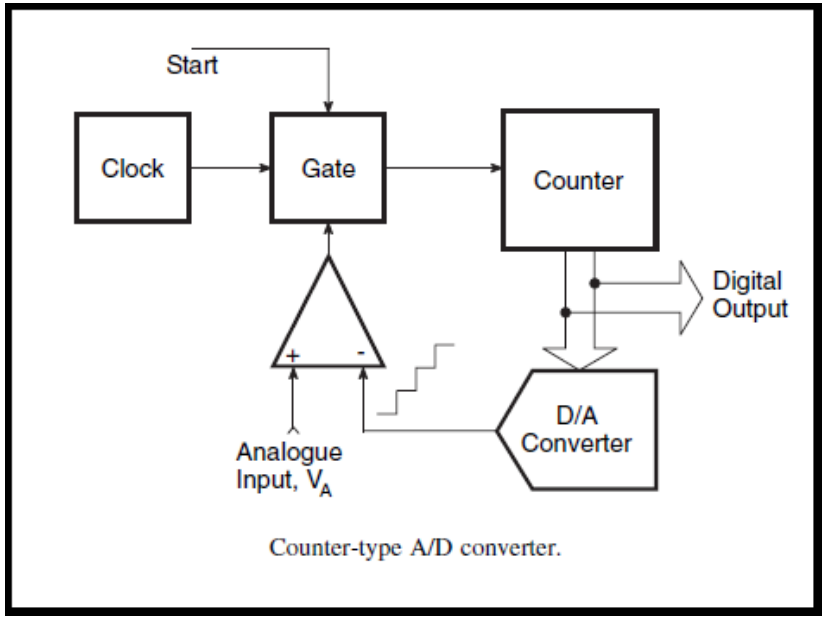




Appendix: Timer and ADC

✓Working principle of ADC

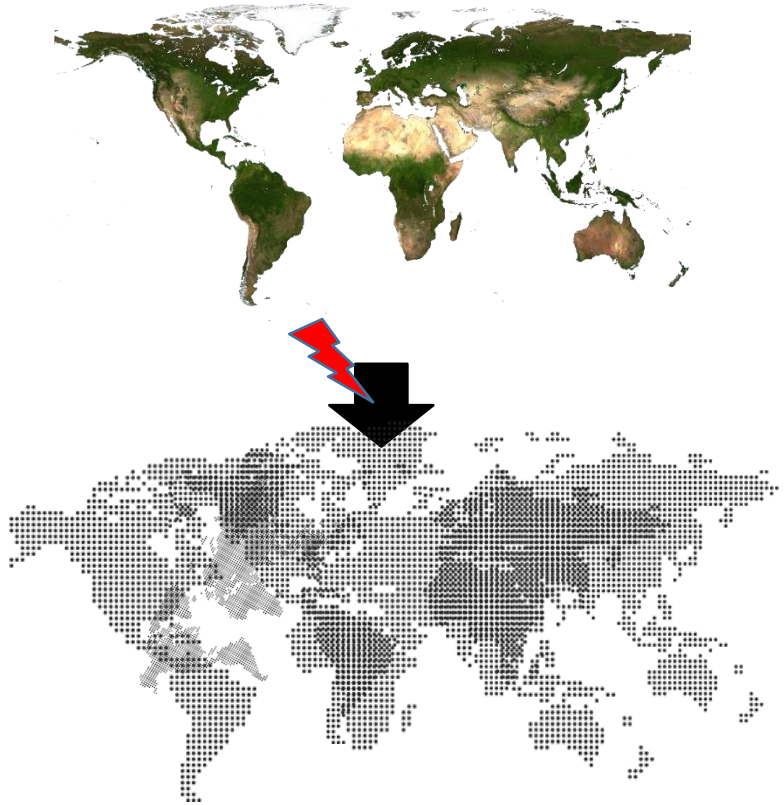


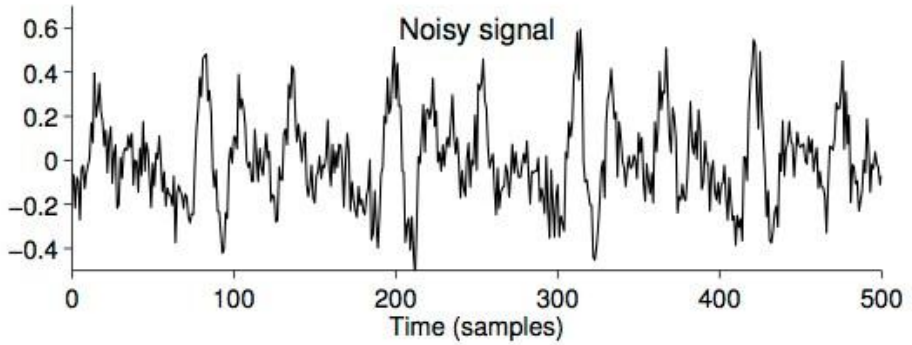


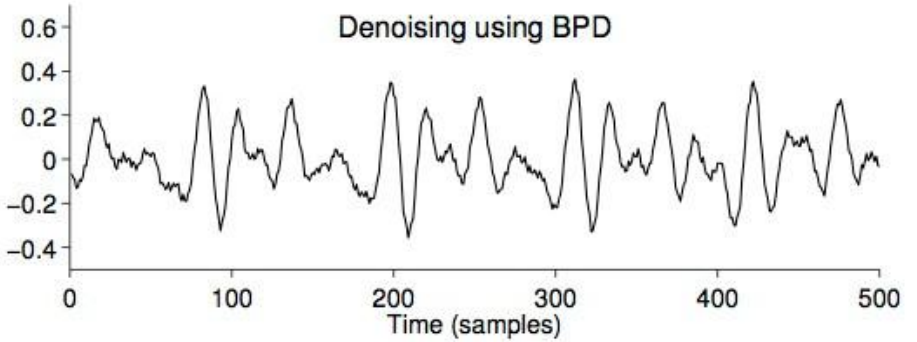
Counter-type A/D converter.

Appendix: Basic of signal processing design

✓Motivation



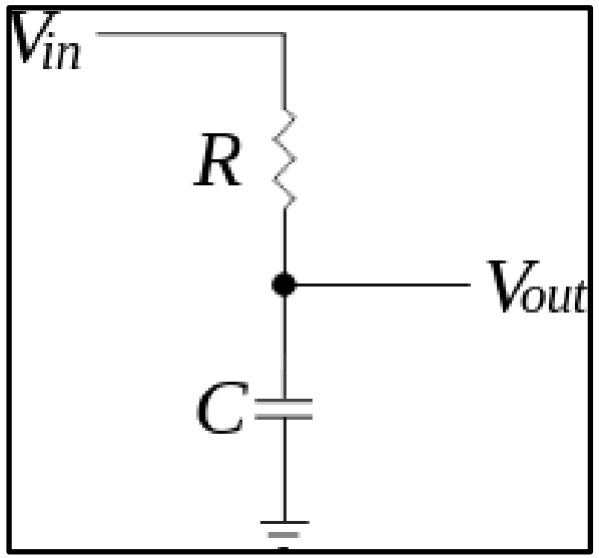


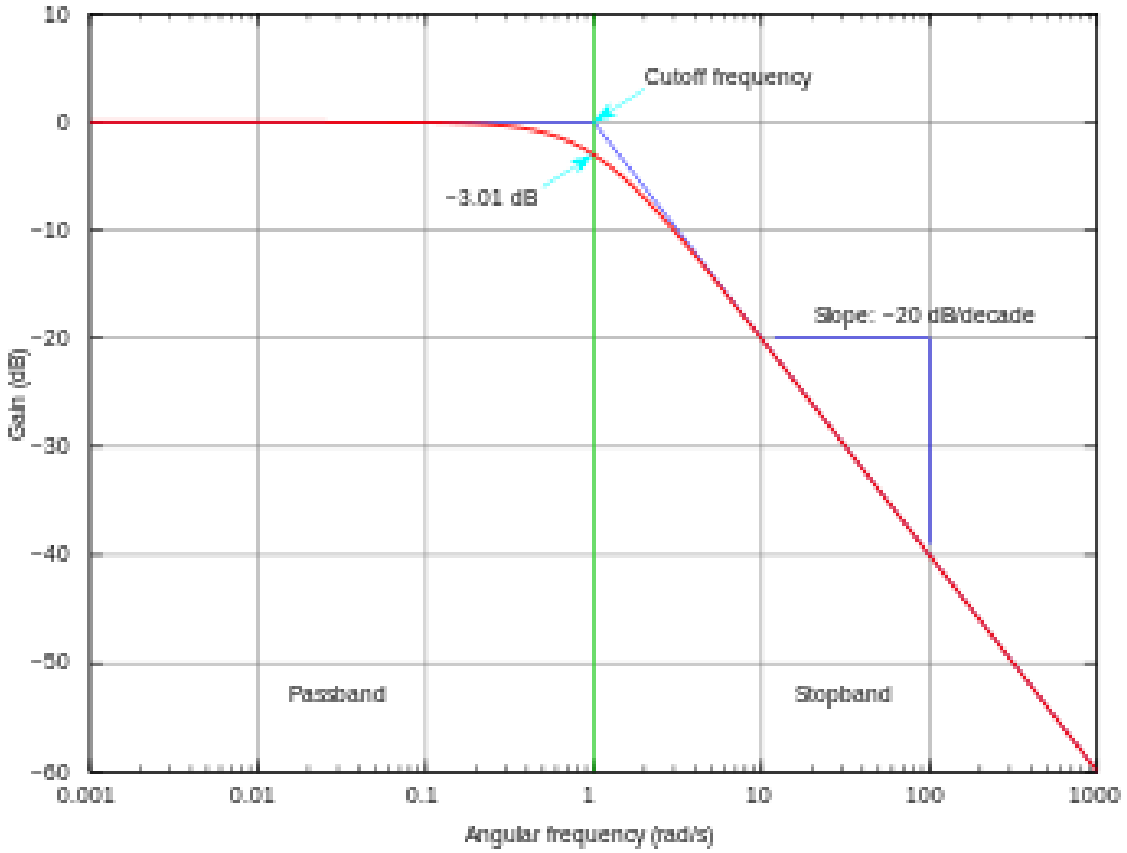


Appendix: Simple filter design

- ✓Analog filter (low-pass filter)
  - ✓$f_c = \frac{1}{2\pi c} = \frac{1}{2\pi R}$

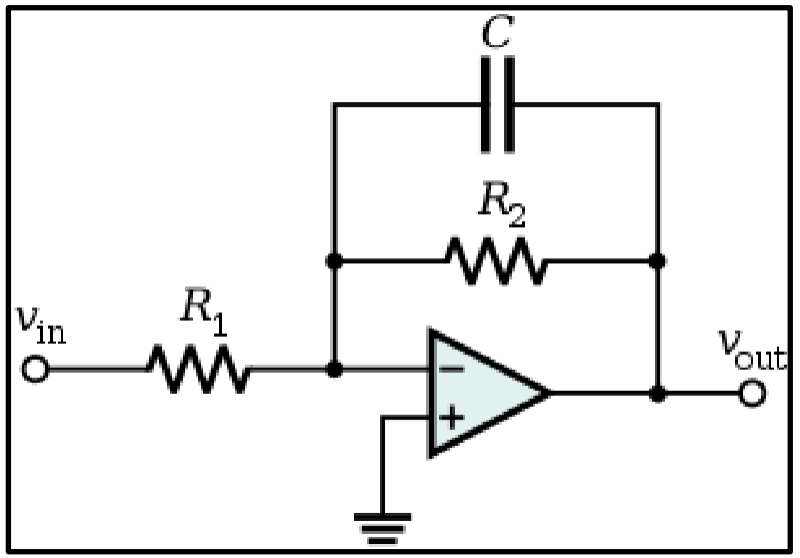
C





Appendix: Simple filter design

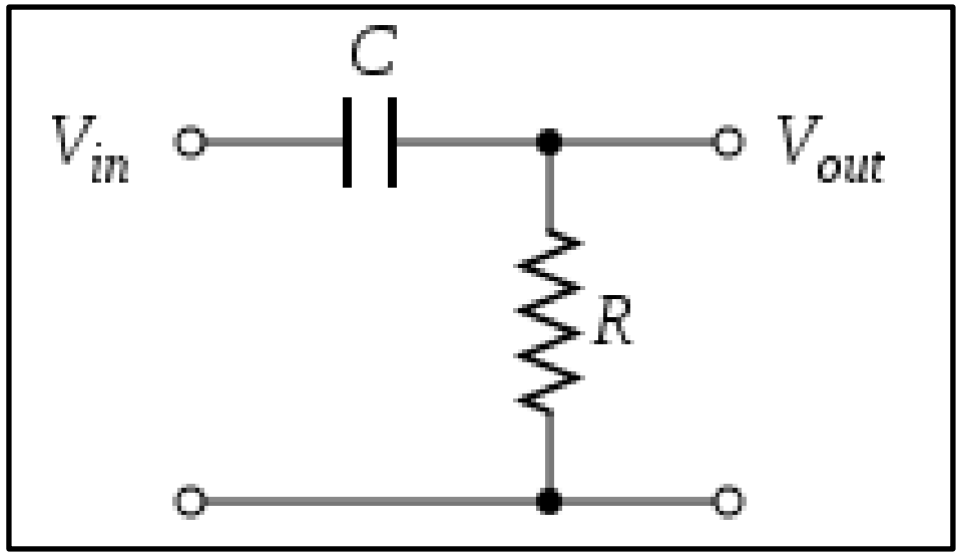
- ✓Analog filter (low-pass filter)
  - ✓$f_c = \frac{1}{2\pi c} = \frac{1}{2\pi R_2 C}, with\ gain\ in\ passband\ is\ -\frac{R_2}{R_1}$



Appendix: Simple filter design

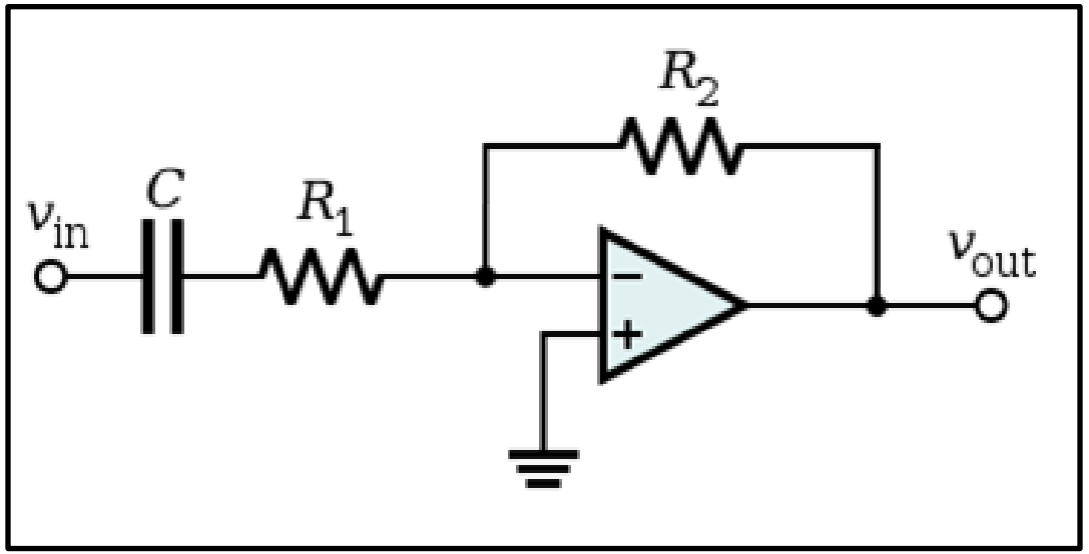
- ✓Analog filter (high-pass filter)
  - ✓$f_c = \frac{1}{2\pi c} = \frac{1}{2\pi R}$

C



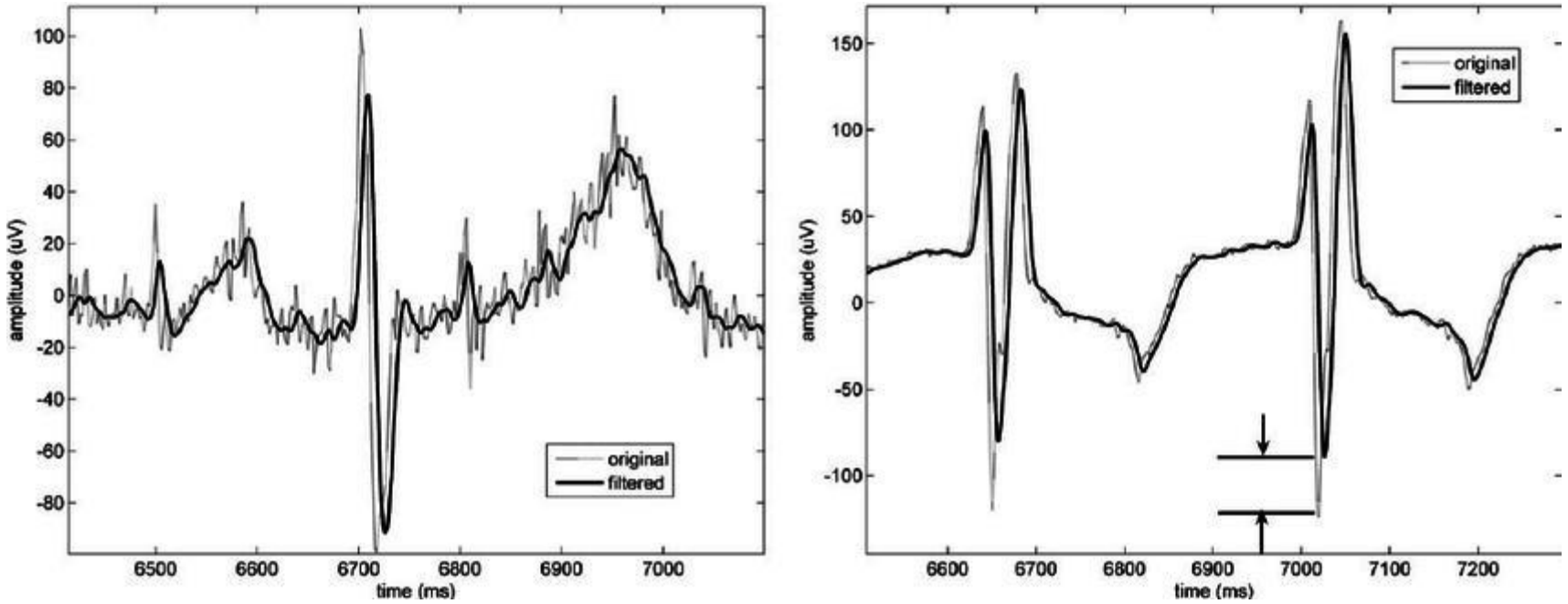
Appendix: Simple filter design

- ✓Analog filter (high-pass filter)
  - ✓$f_c = \frac{1}{2\pi c} = \frac{1}{2\pi R_1 C}, with\ gain\ in\ passband\ is\ -\frac{R_2}{R_1}$



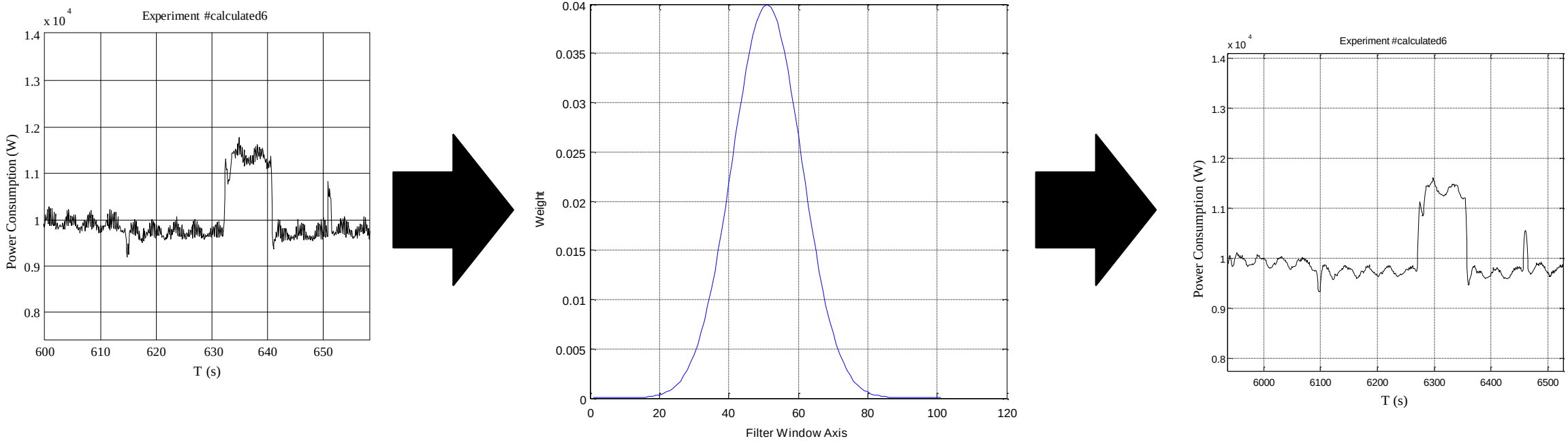
Appendix: Simple filter design

✓Effect of the filter (low-pass filter)



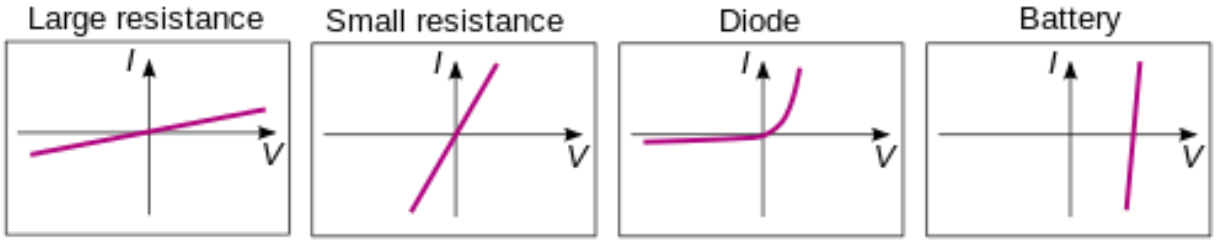
Appendix: Digital filter design

✓Gaussian digital filter

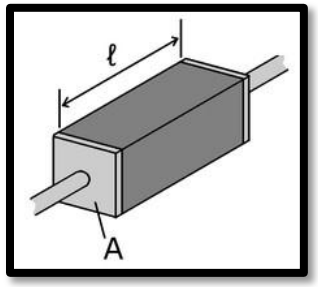


Appendix

- ✓Resistor, Inductor and Capacitor
  - ✓I a V (Ohn's law)

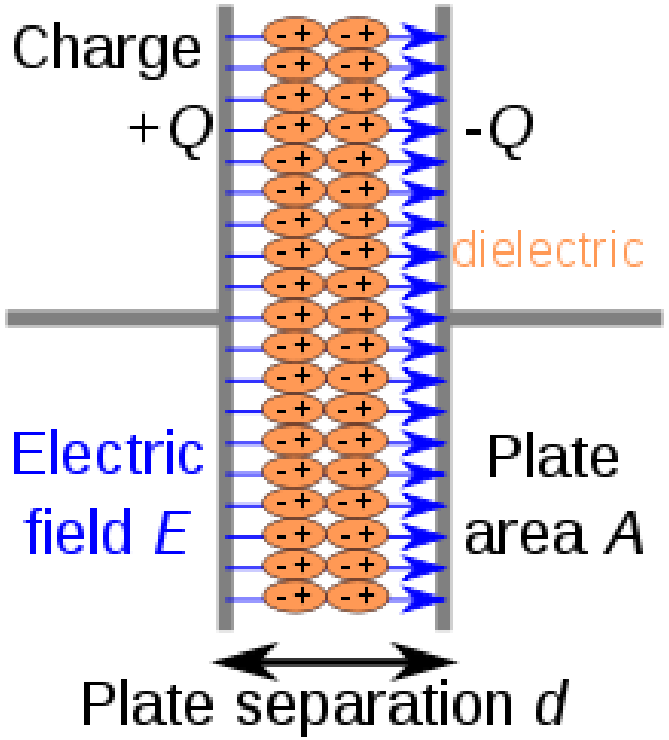


  - ✓$R = \rho \frac{l}{A}, where\ \rho\ is\ specific\ electrical\ resistance$



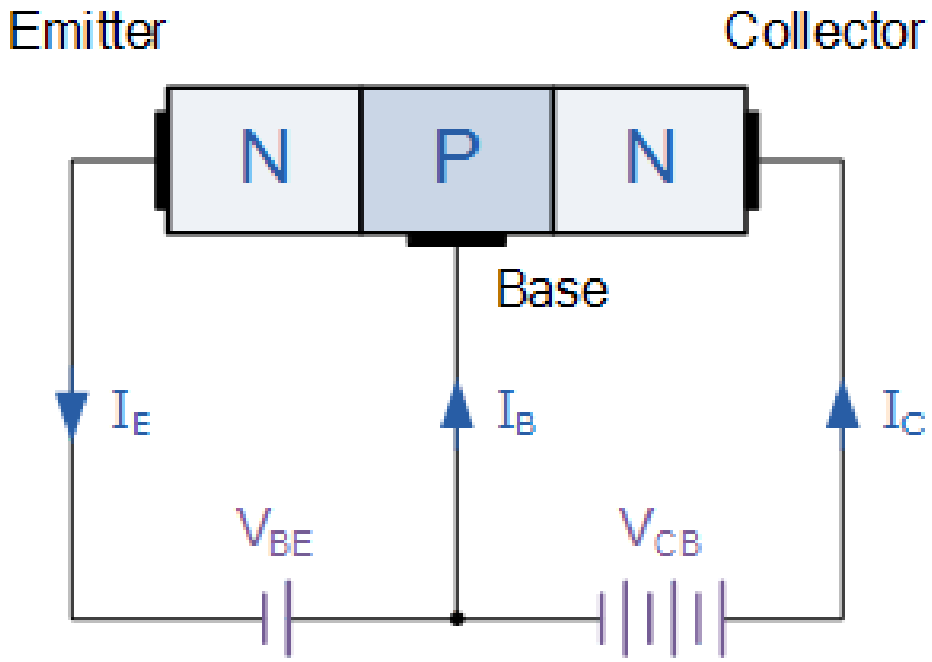
Appendix

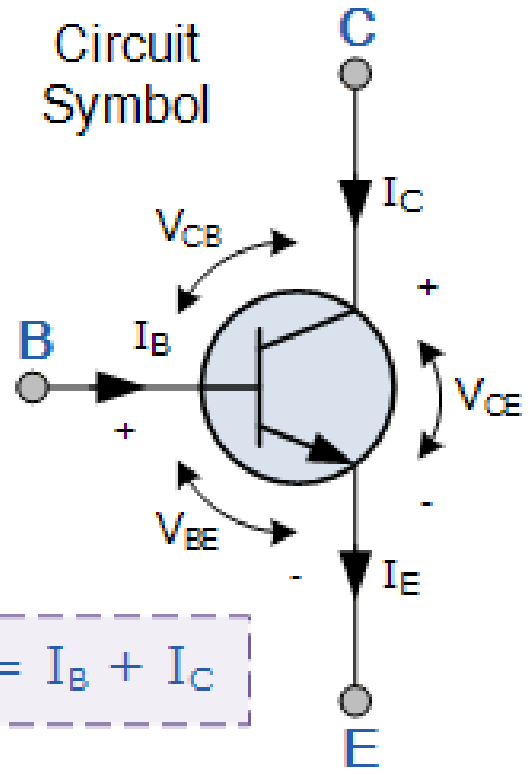
- ✓Resistor, Capacitor
  - ✓$C = \frac{Q}{V}$
  - ✓$V(t) = \frac{Q(t)}{C} = \frac{1}{C}\int_{t_0}^{t} I(r)dr + V(t_0)$



Appendix

✓Transistor [http://www.electronics-tutorials.ws/transistor/tran_2.html]

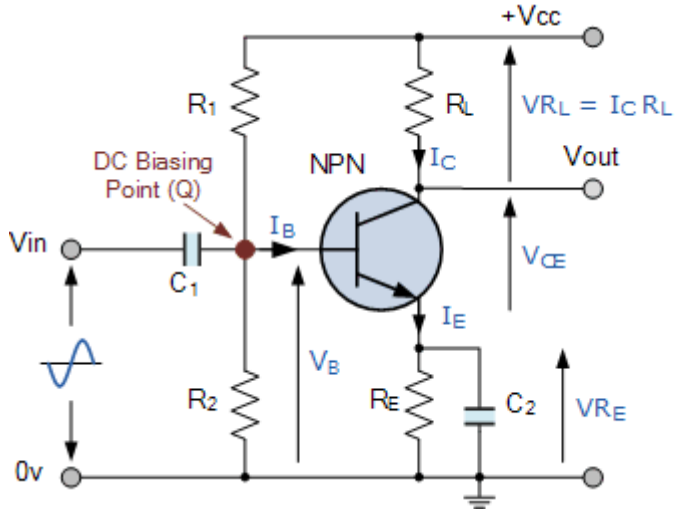


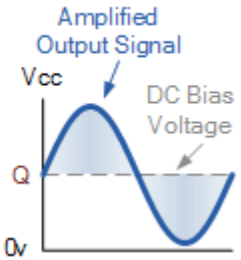


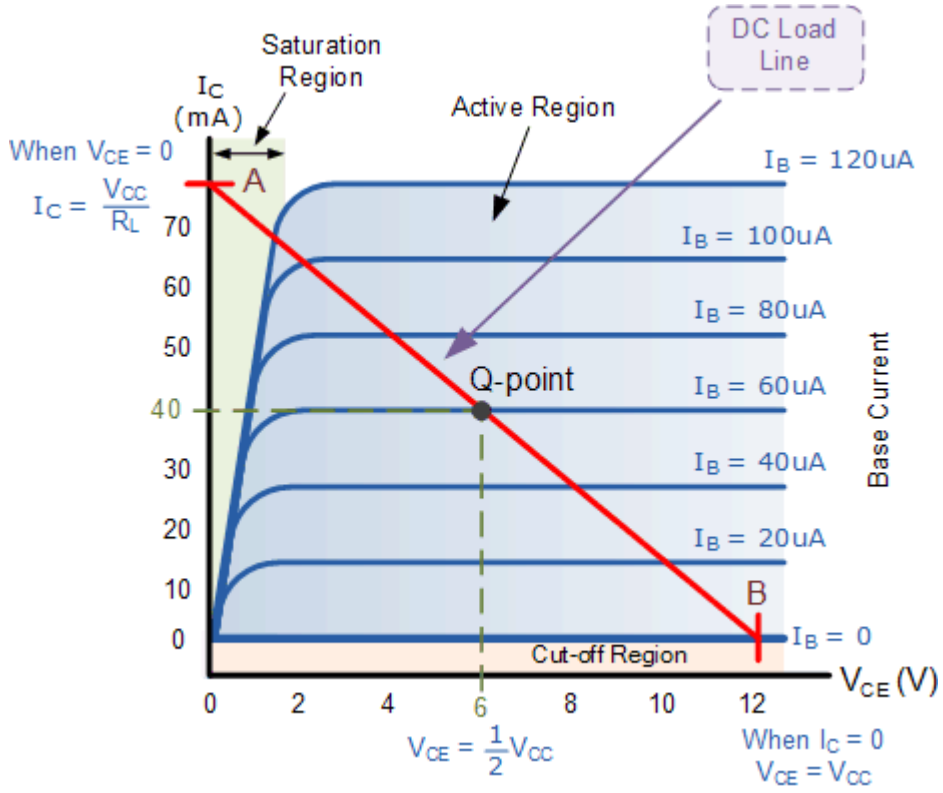
$I_E = I_B + I_C$

Appendix

✓Transistor [http://www.electronics-tutorials.ws/transistor/tran_2.html]

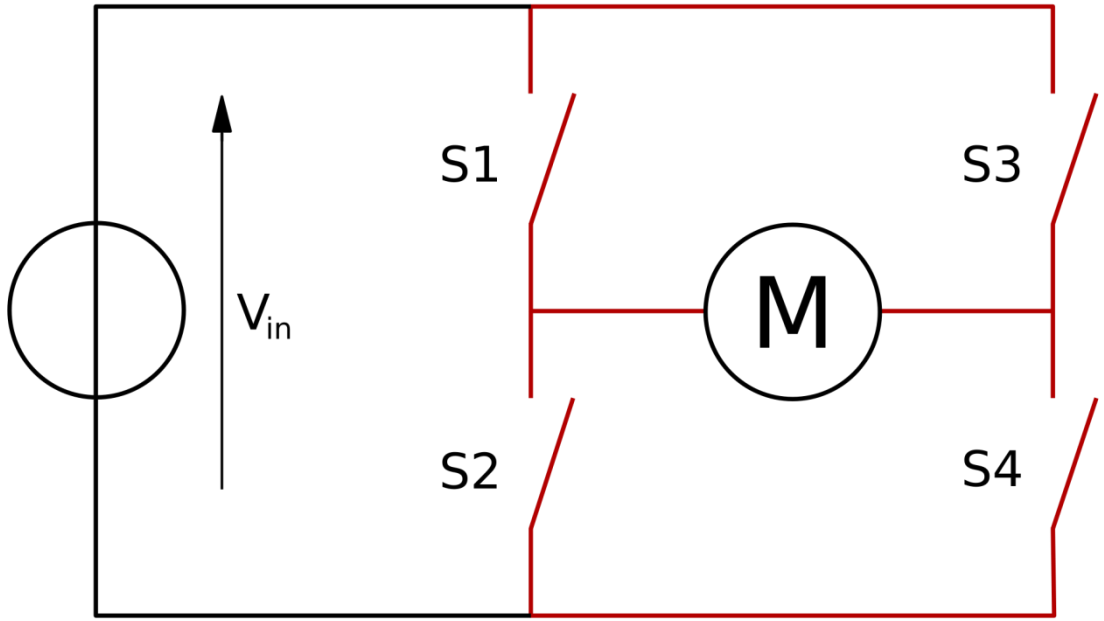






Appendix

✓Current direction control (H-bridge)



References

- **Innovative Design and Integrated Manufacturing Lab., Seoul National University**
- https://www.arduino.cc/
- http://electronics.stackexchange.com/questions/22830/principal-of-analog-to-digital-converteradc-and-digital-to-analog-converterda
- http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/adc.html
- https://en.wikipedia.org/wiki/Low-pass_filter
- https://www.researchgate.net/
- http://www.dfrobot.com/wiki/index.php/DFRduino_Beginner_Kit_For_Arduino_V3_SKU:DFR0100