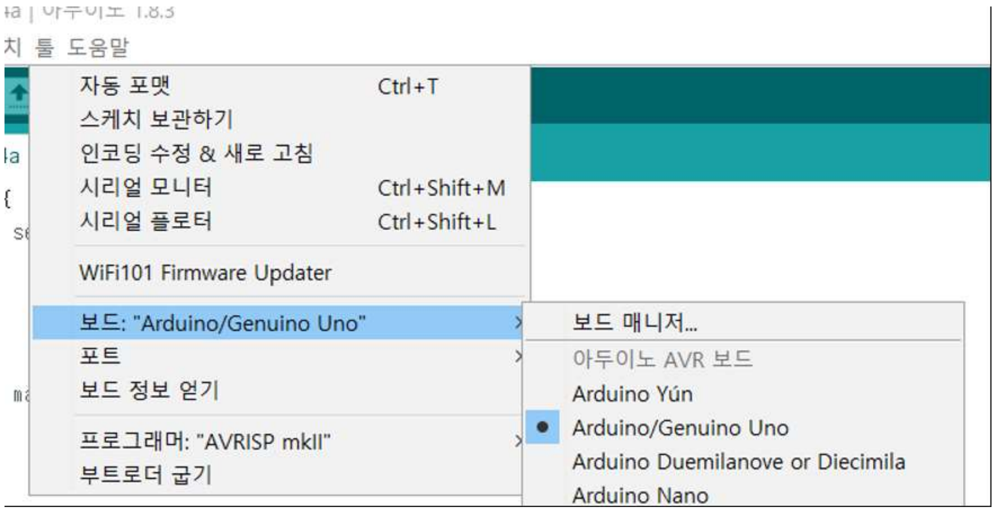
스마트시스템입문

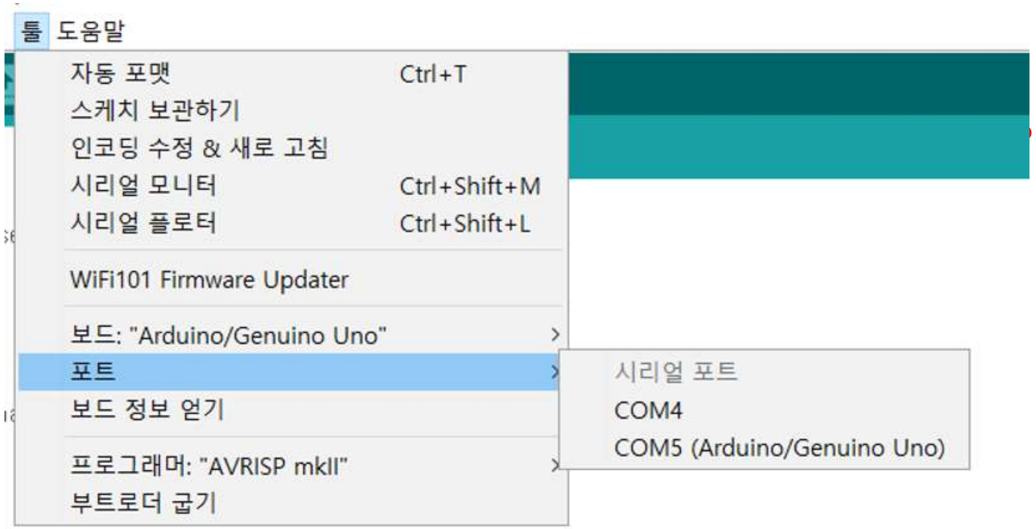
2018년 1학기

IDE programming

- Connect Arduino to PC using USB cablE

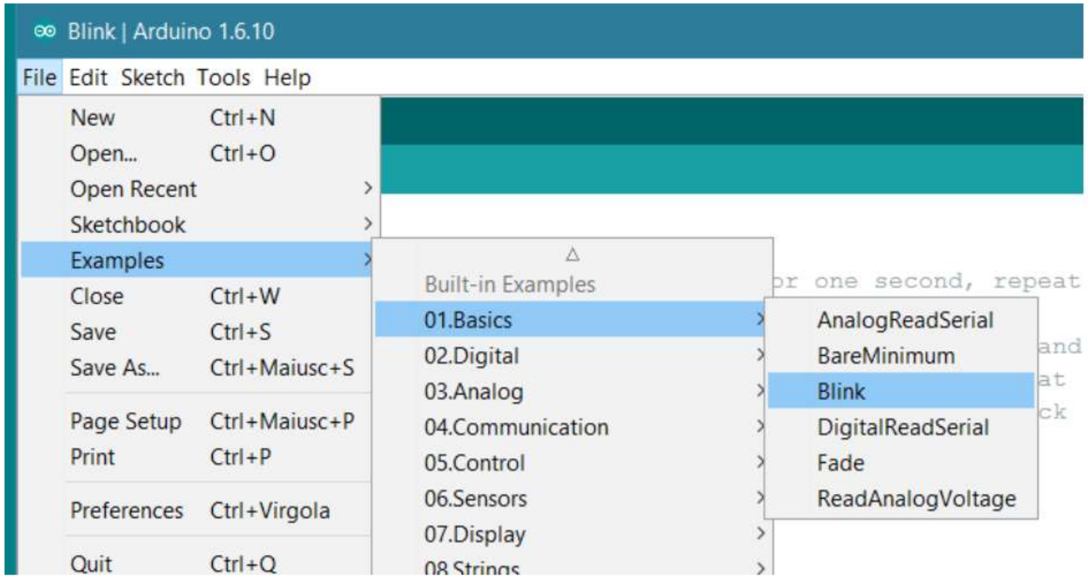


- USB 케이블을 끝까지 밀어서 연결해야함

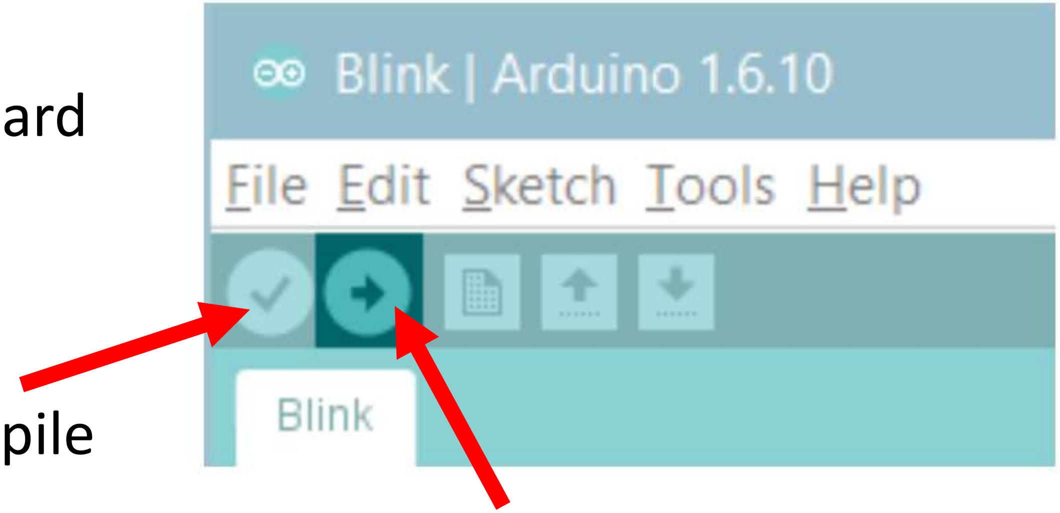


- You must choose a right COM to make connection

Blink exmaple



- Blink built-in LED
- Upload sketch to the board



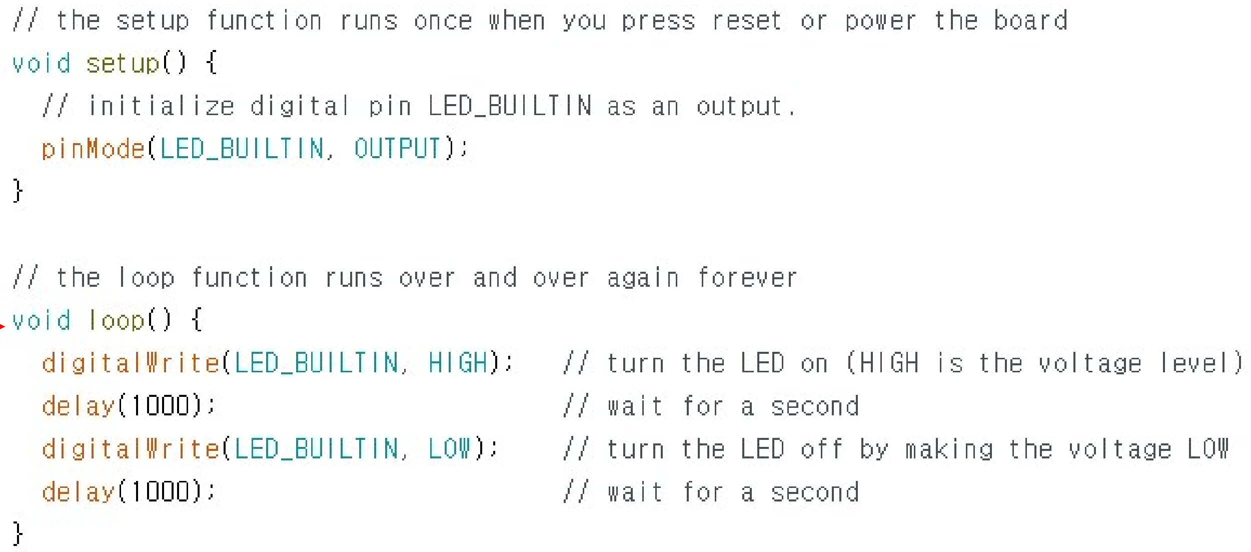
Compile

Upload

Blink exmaple

Run once at
the beginning

Repeat this
loop function



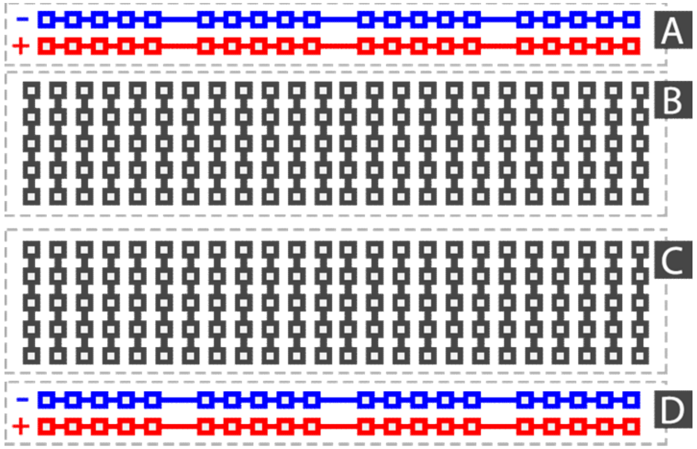
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

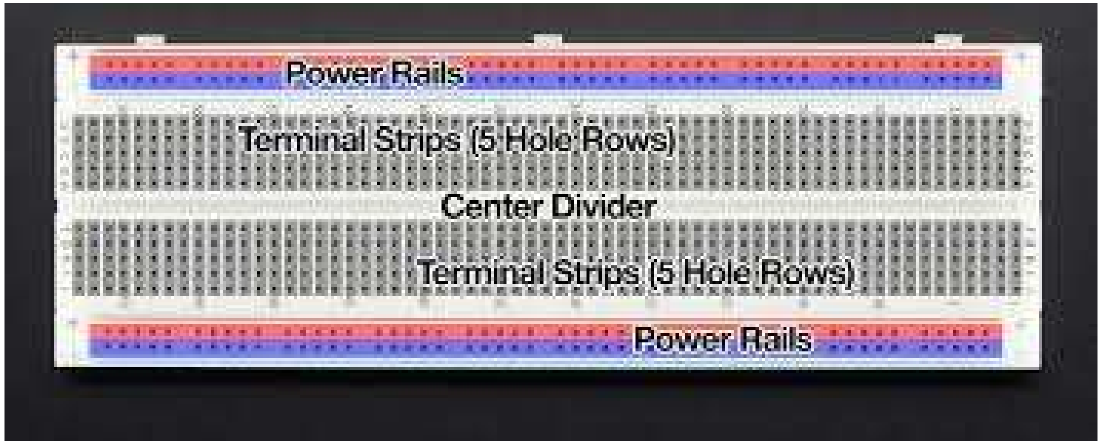
pinMode(LED_BUILTIN, OUTPUT);

digitalWrite(LED_BUILTIN, HIGH);

delay(1000);

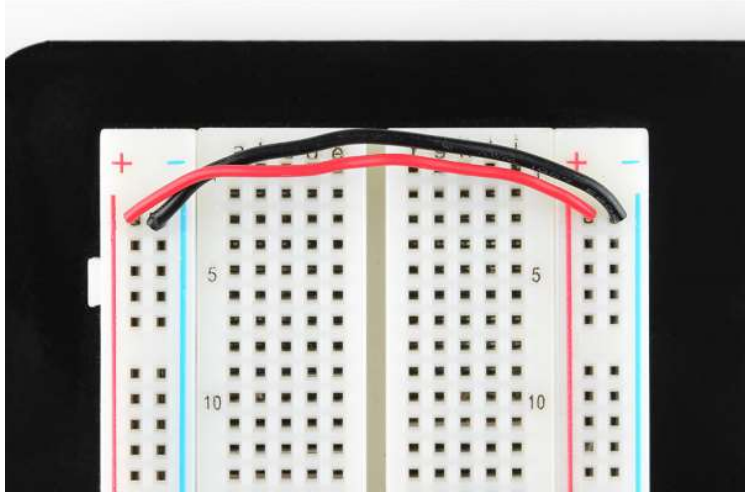
Breadboard

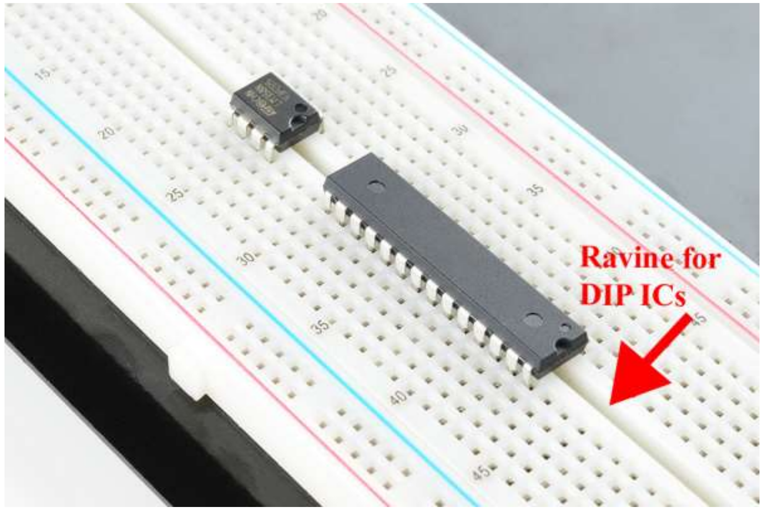




https://www.tweaking4all.com/wp-content/uploads/2013/12/basic_breadboard_layout.png

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQVr2BtZ8enjA-ZTOF_HIvY0yTD5LTdsiOpa7-6rPuPEAoXy3pSLA





https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard

Analog vs Digital

- 아날로그 신호 vs 디지털 신호

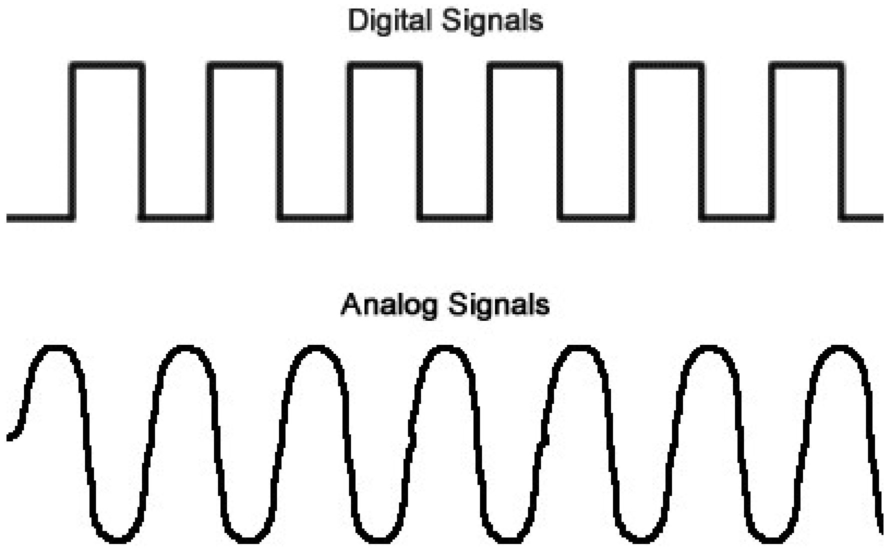
- 신호, signal
  - Time-varying “quantities” which convey some sort of information. In electrical engineering the quantity that’s time-varying is usually voltage.

https://learn.sparkfun.com/tutorials/analog-vs-digital

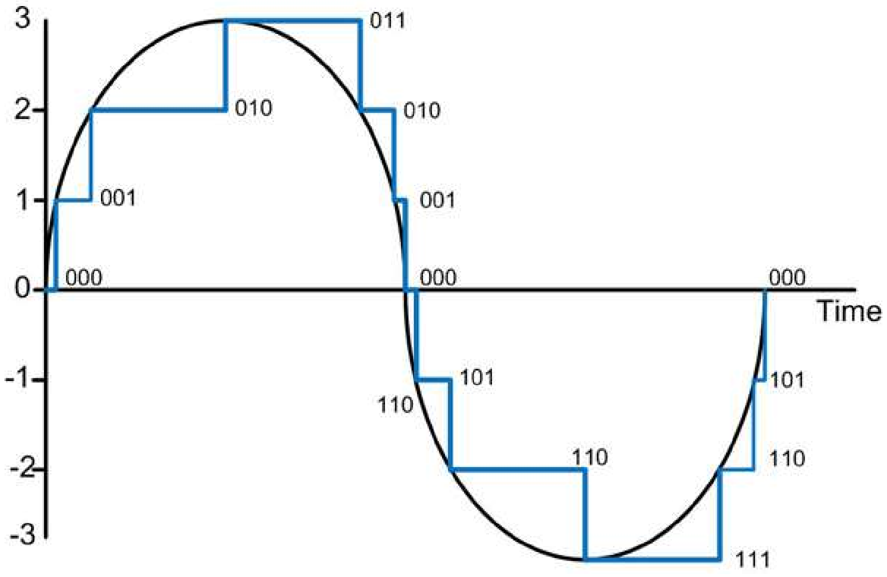
- 차이점: 변화량이 가질 수 있는 값의 범위

Analog vs Digital

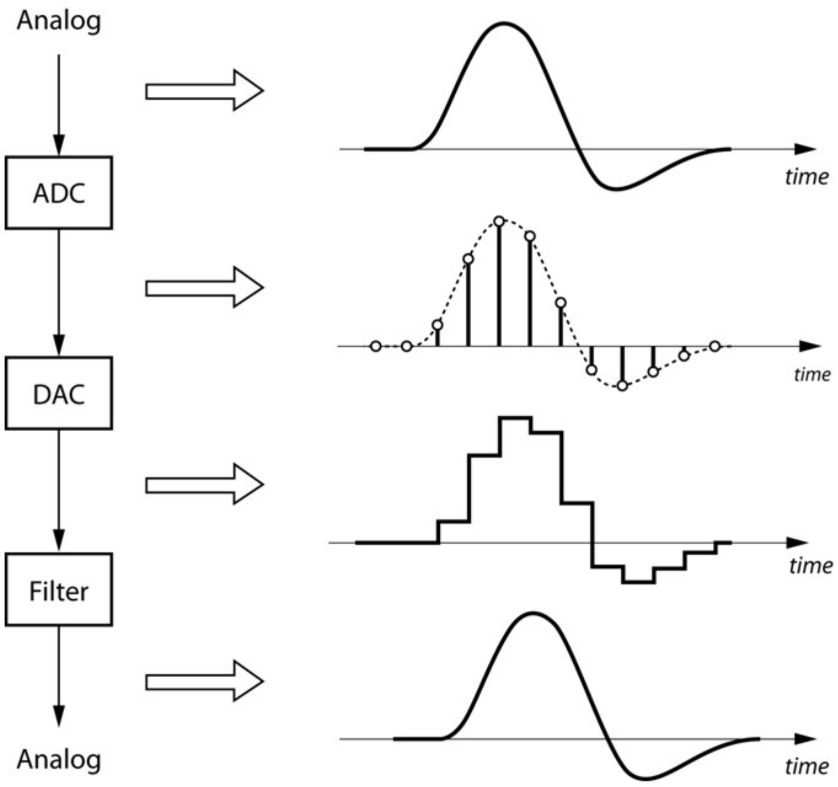
- Digital signal: only 0 and 1 exist
  - On / off button, switch
- Analog signal: continuous value
  - Analog sensors: light sensor



ADC (analog to digital converter)



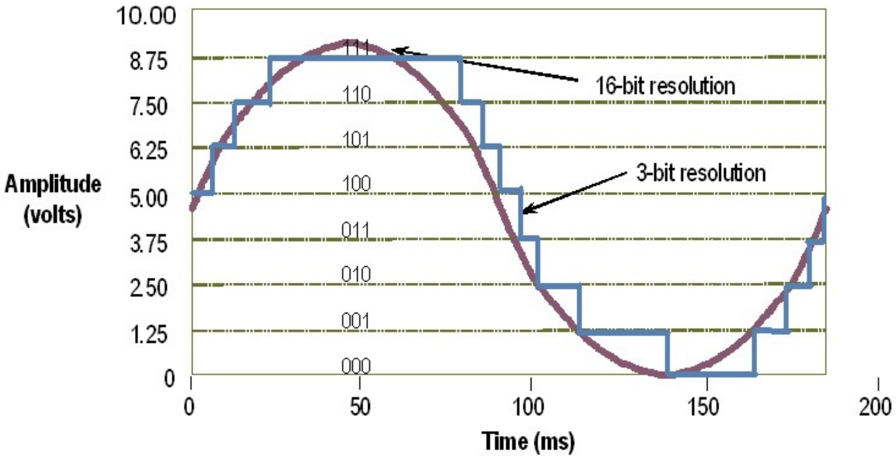
http://screaminfx.com/images/tech-images/what-is-analog-verse-digital-explanation.jpg

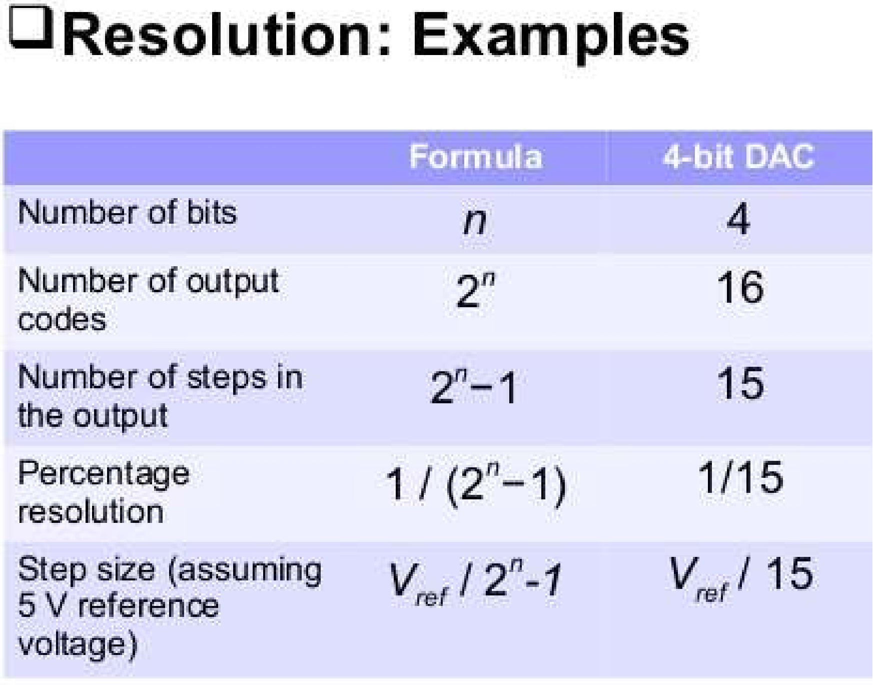


https://www.nutaq.com/sites/default/files/images/blog-images/Process%20of%20digitizing%20and%20converting%20a%20signal%20with%20an%20infinite%20precision%20ADC-DAC_0.png

ADC (analog to digital converter)

What is “resolution”? For example, 10-bit resolution.





❑Resolution: Examples

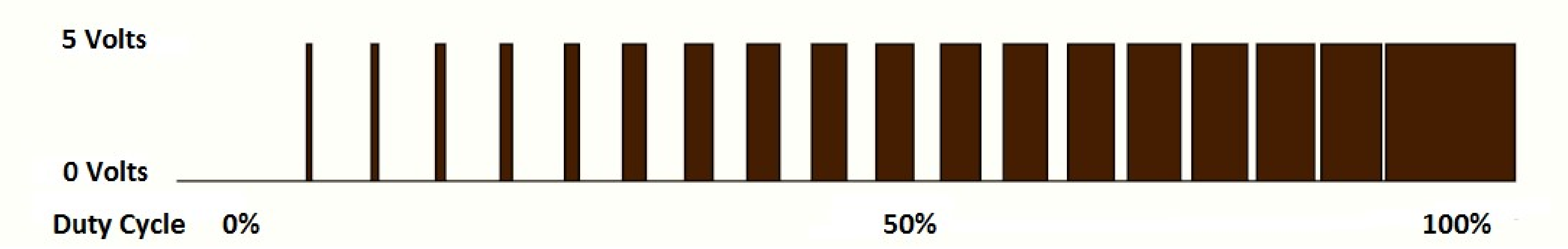
	Formula	4-bit DAC
Number of bits	n	4
Number of output codes	2^n	16
Number of steps in the output	2^n-1	15
Percentage resolution	$1 / (2^n-1)$	1/15
Step size (assuming 5 V reference voltage)	$V_{ref} / 2^n$-1	V_{ref} / 15

http://www.globalspec.com/learnmore/data_acquisition_signal_conditioning/signal_converting/digital_to_analog_converters

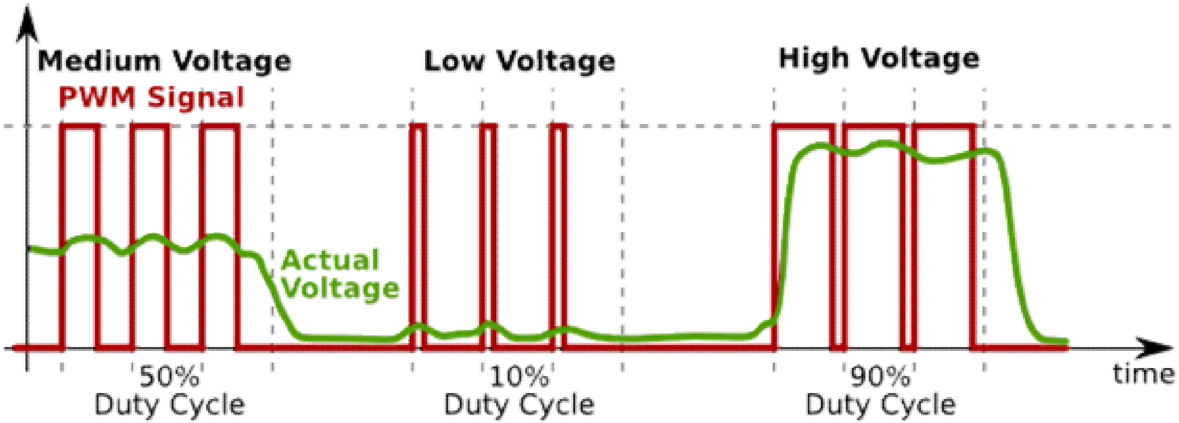
https://image.slidesharecdn.com/group1-141018102550-conversion-gate01/95/dacdigital-to-analog-converter-11-638.jpg?cb=1486790152

PWM

- Control duty cycle → control average voltage

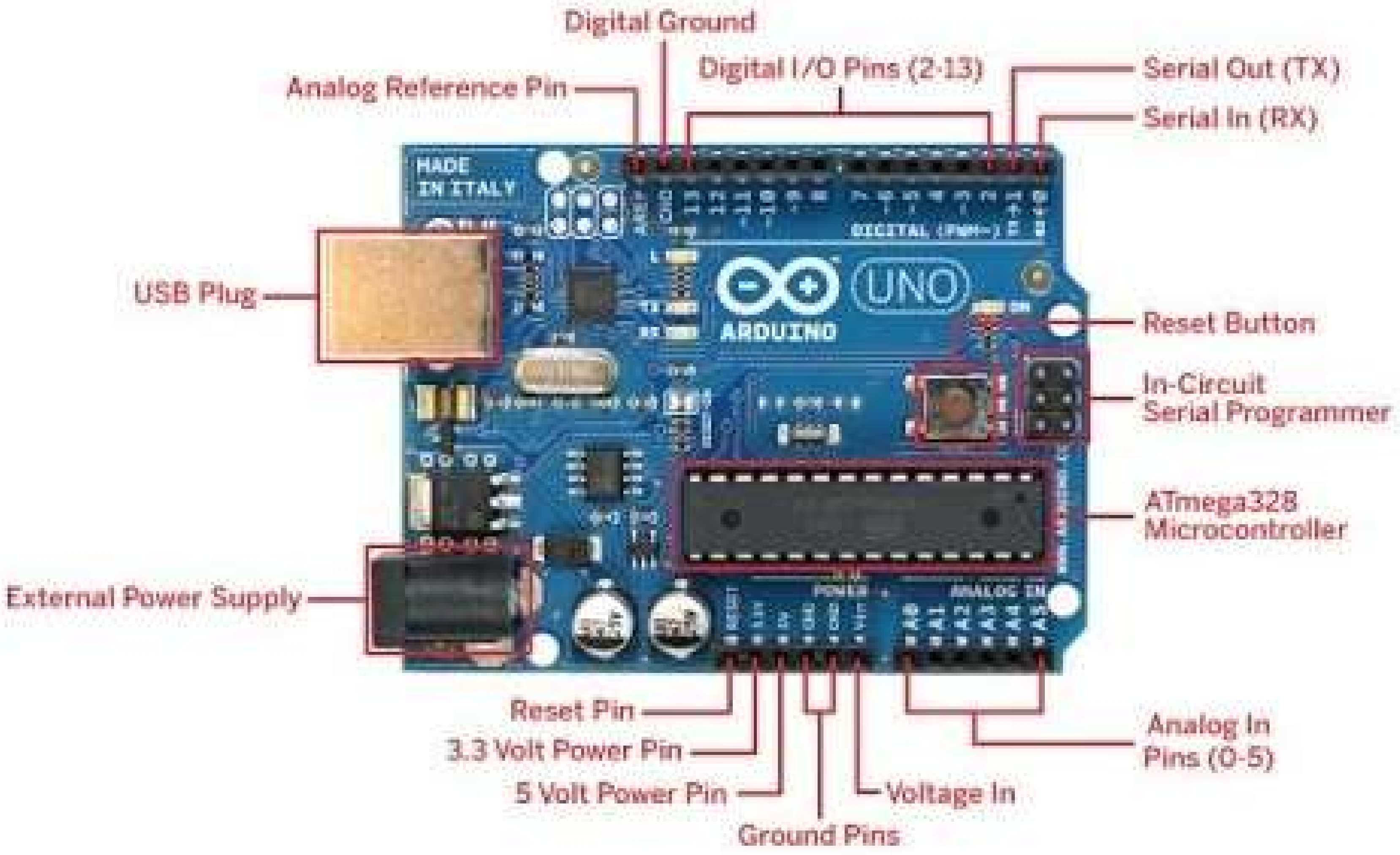


https://electrosome.com/pwm-pulse-width-modulation/



https://www.wayneandlayne.com/projects/video-game-shield/design/

Arduino Uno

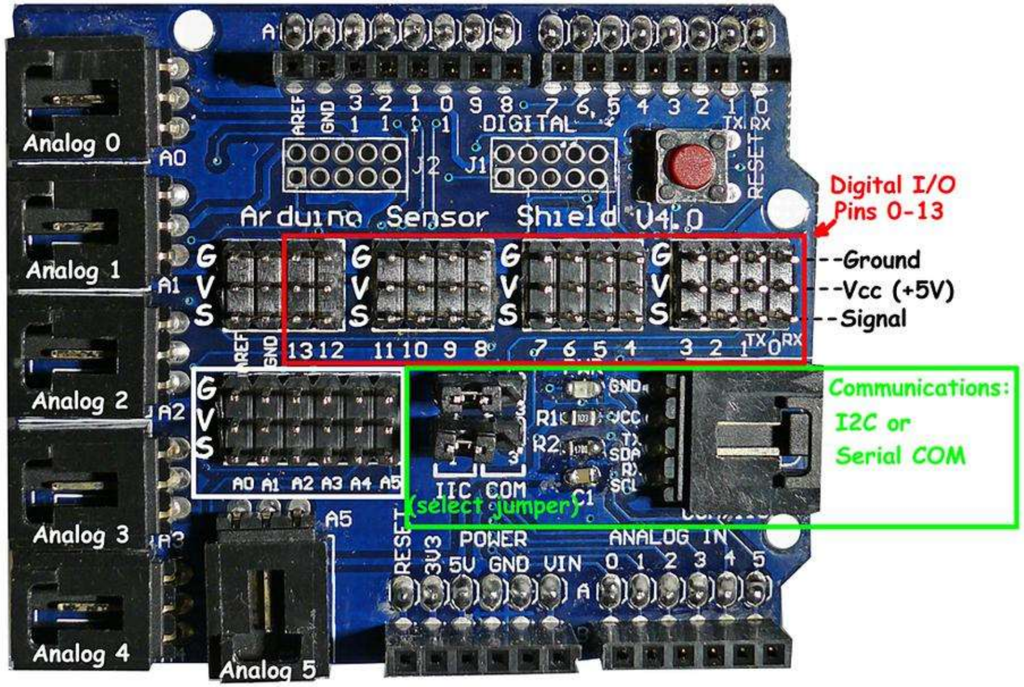


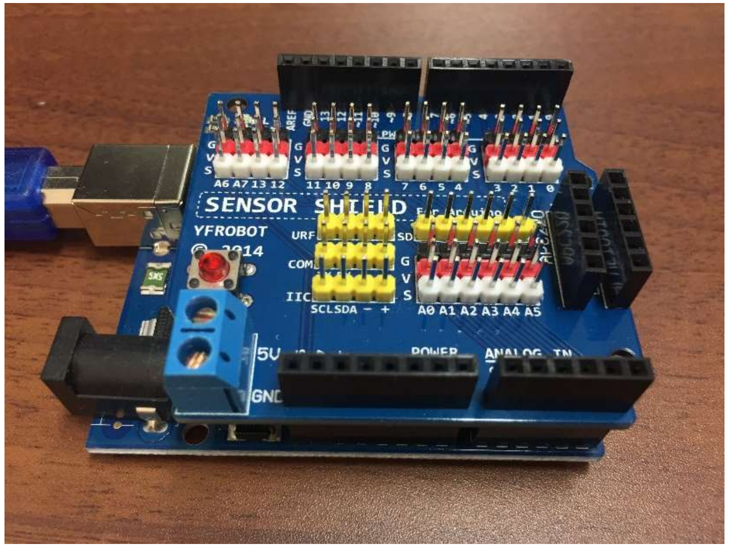
https://i.ytimg.com/vi/4sopoQKVNIM/hqdefault.jpg

Sensor shield (extension board)

https://arduino-info.wikispaces.com/SensorShield

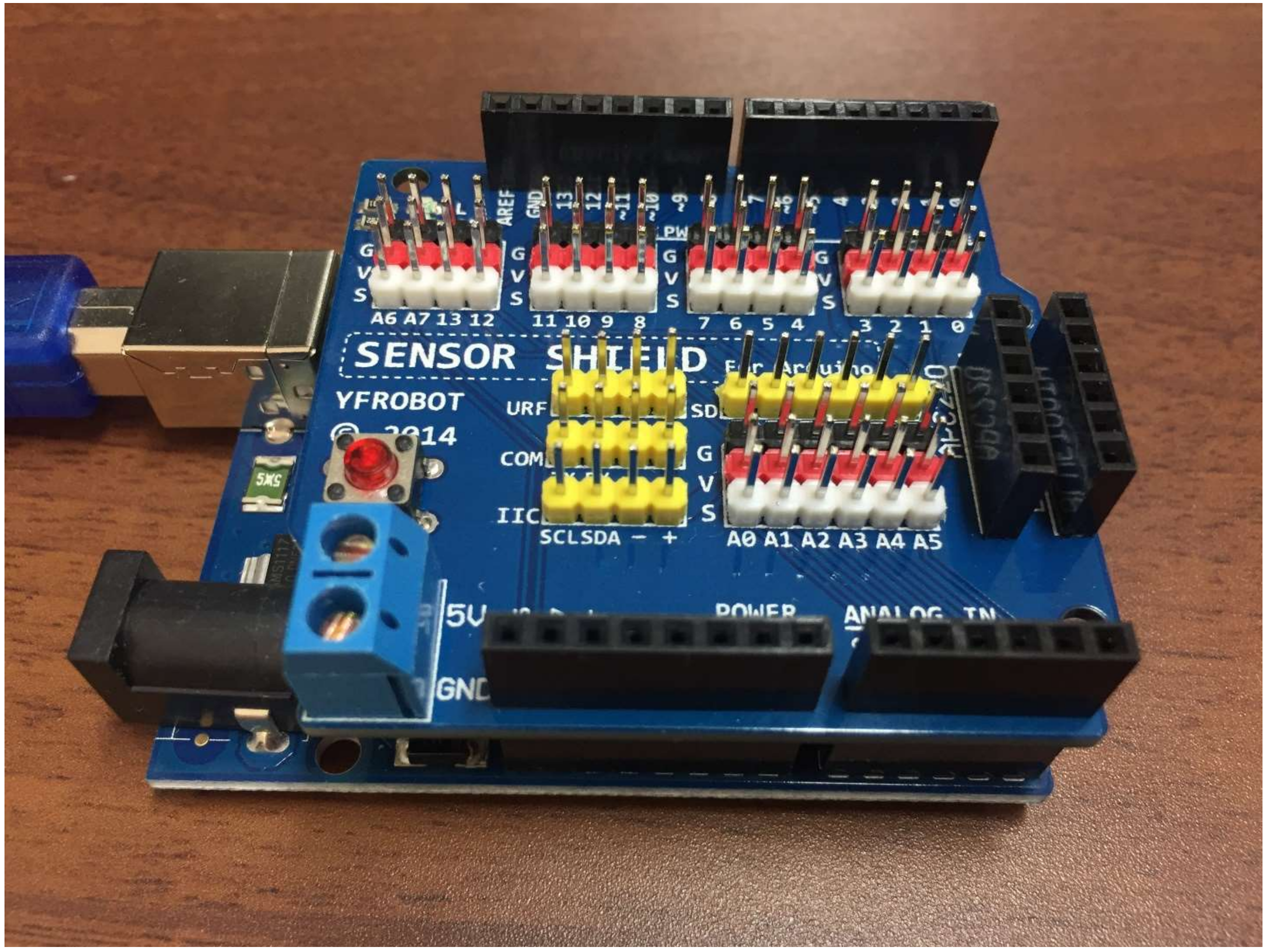
SENSOR SHIELD:
An easy way to connect many INPUT DEVICES and OUTPUT DEVICES to Arduino. Not just sensors!



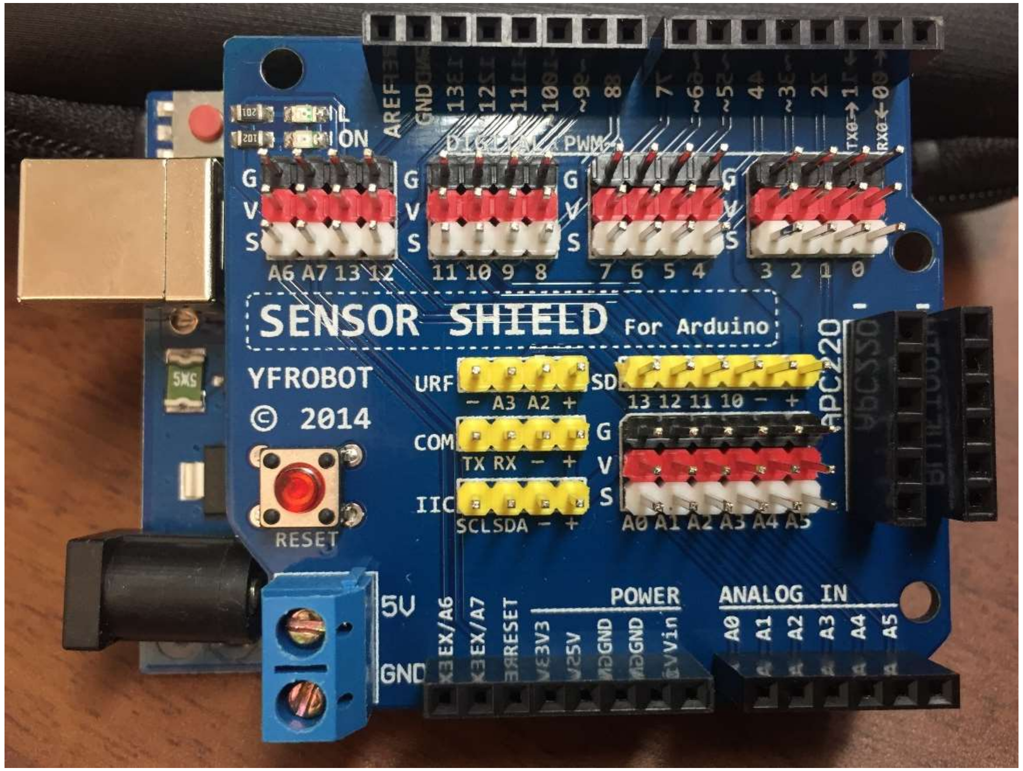


Sensor shield (extension board)

Mount the extension board on the top of the main board



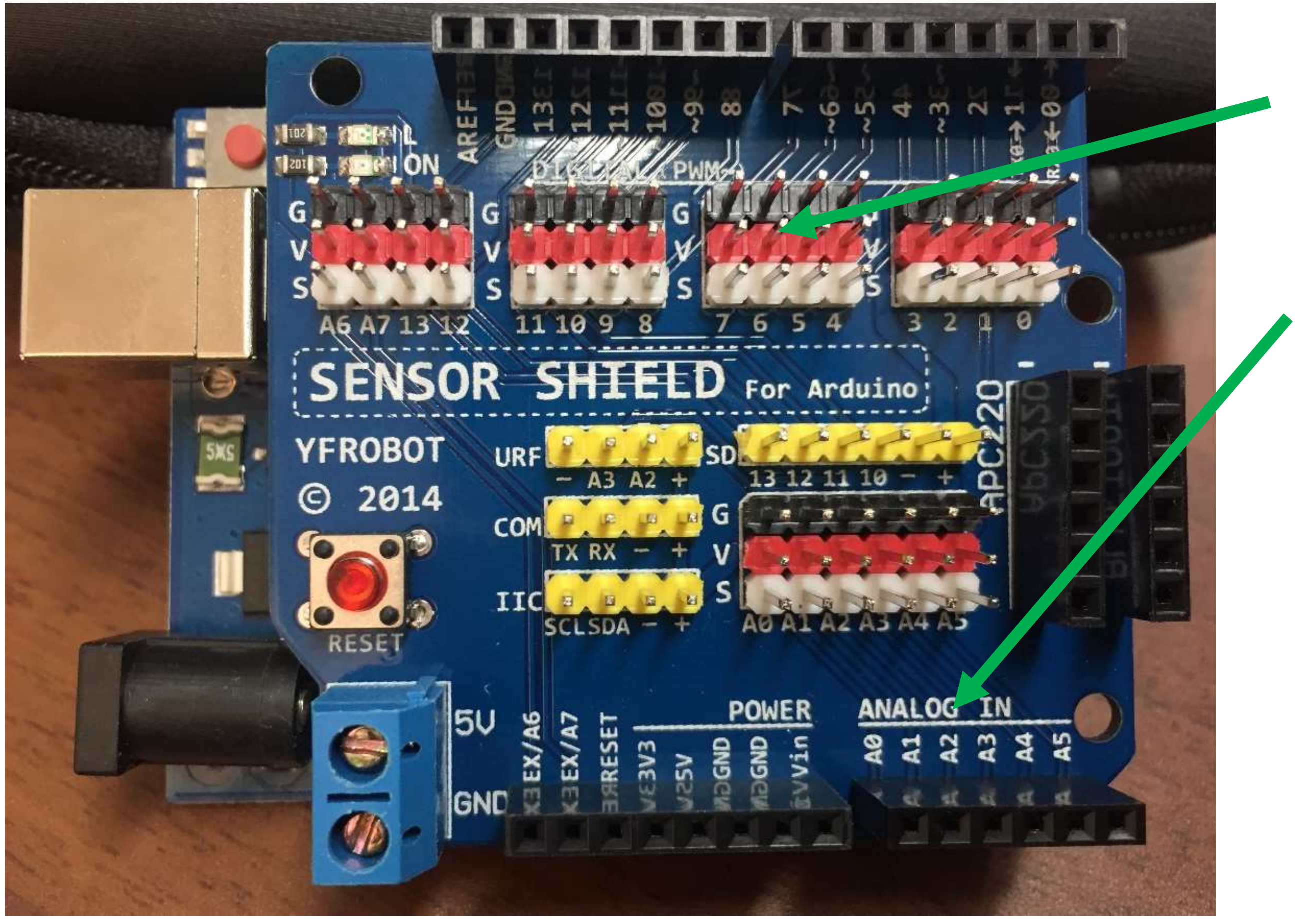
Sensor shield (extension board)



- Input/ouput ports are now connected to the main board
- G/V/S – black, red, white (check the sensor and board)
- Analog ports
- Digital ports

PWM from Arduino

- Can Arduino generate analog signal?



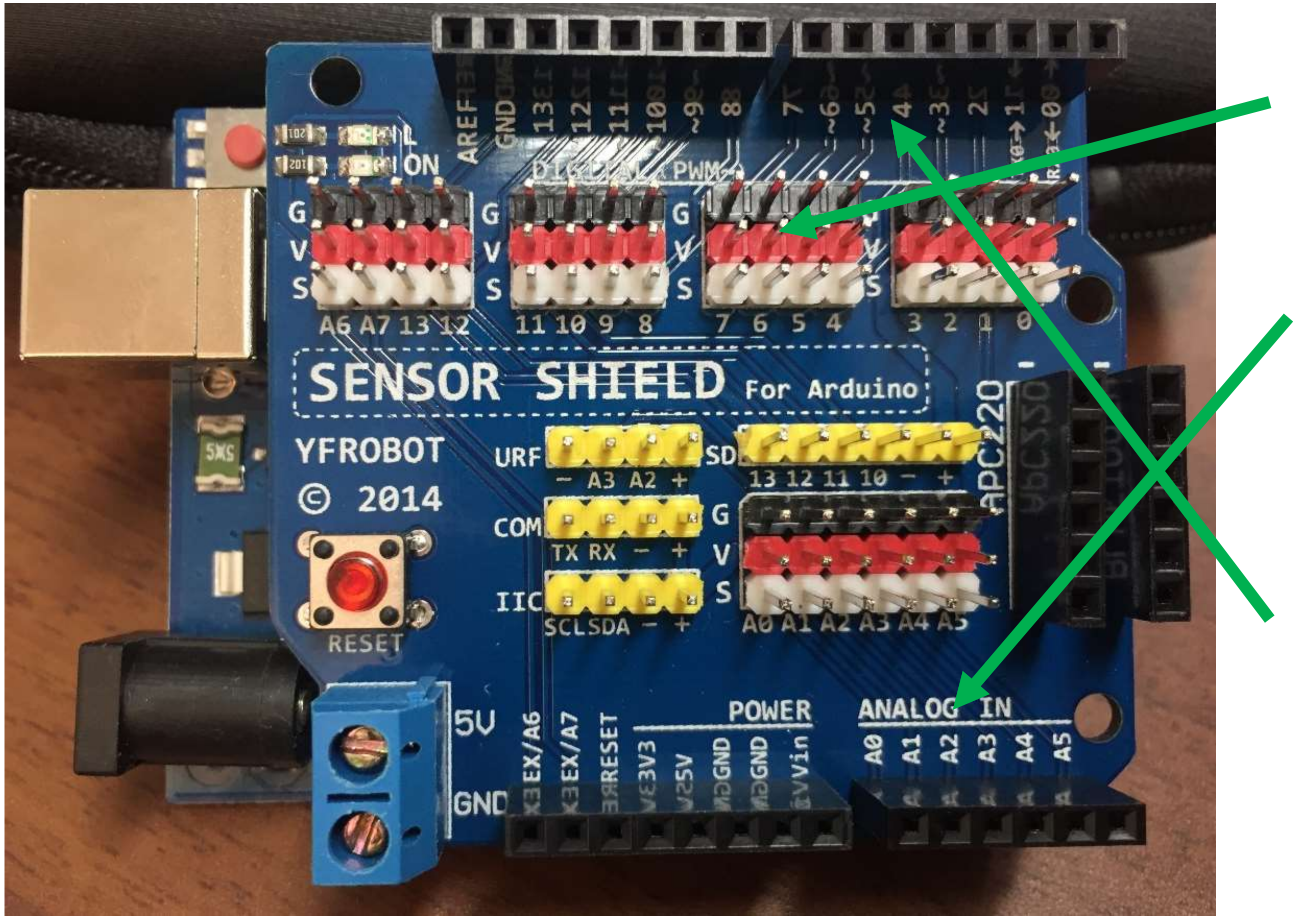
Digital in/out

Analog in

Where's analog out?

PWM from Arduino

- Can Arduino generate analog signal?



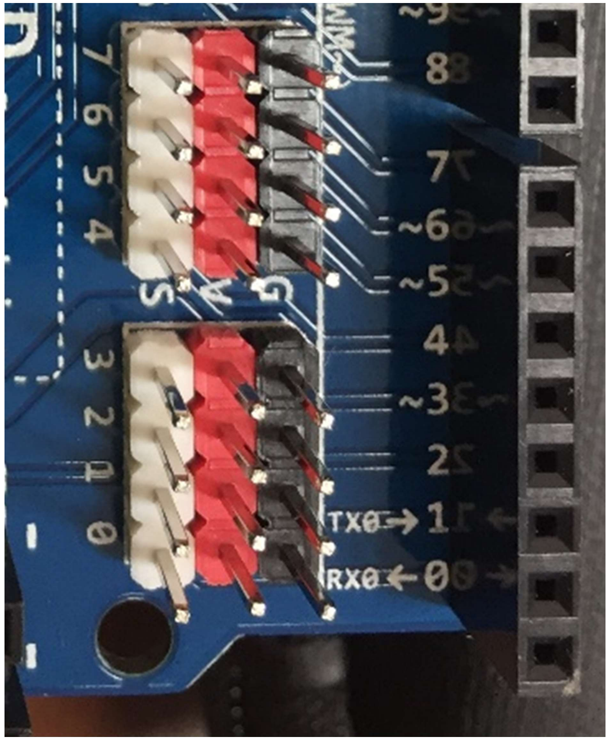
Digital in/out

Analog in

Where's analog out?
~ 11, ~10, ~9, ~6, ~5, ~3 (PWM)

PWM and analogWrite()

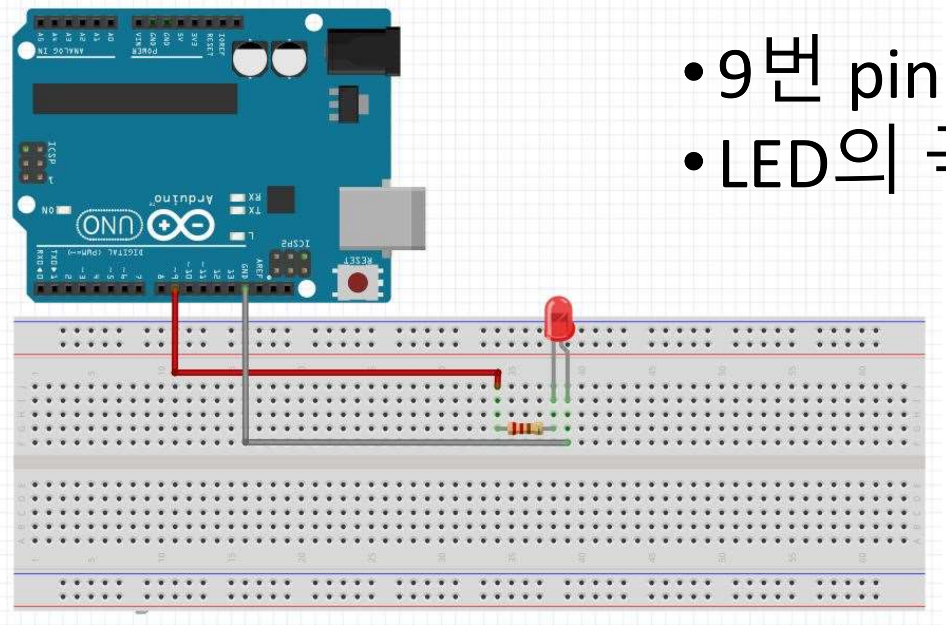
- Arduino board has analog output pins
- There is no “true” analog signal output in Arduino
- Arduino will create pseudo-analog signal using PWM
- PWM will be covered in detail later



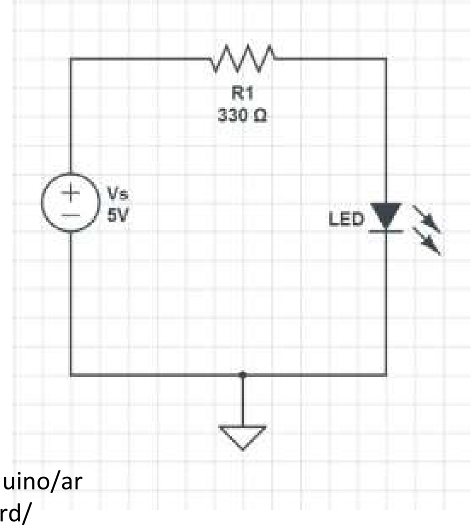
analogWrite()

- Syntax
  - analogWrite(pin, value)
- Parameters
  - pin: the pin to write to.
  - value: the duty cycle: between 0 (always off) and 255 (always on)

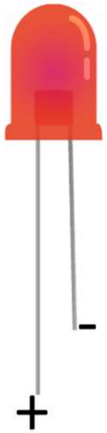
Breadboard



- 9번 pin에 LED 연결
- LED의 극성에 주의

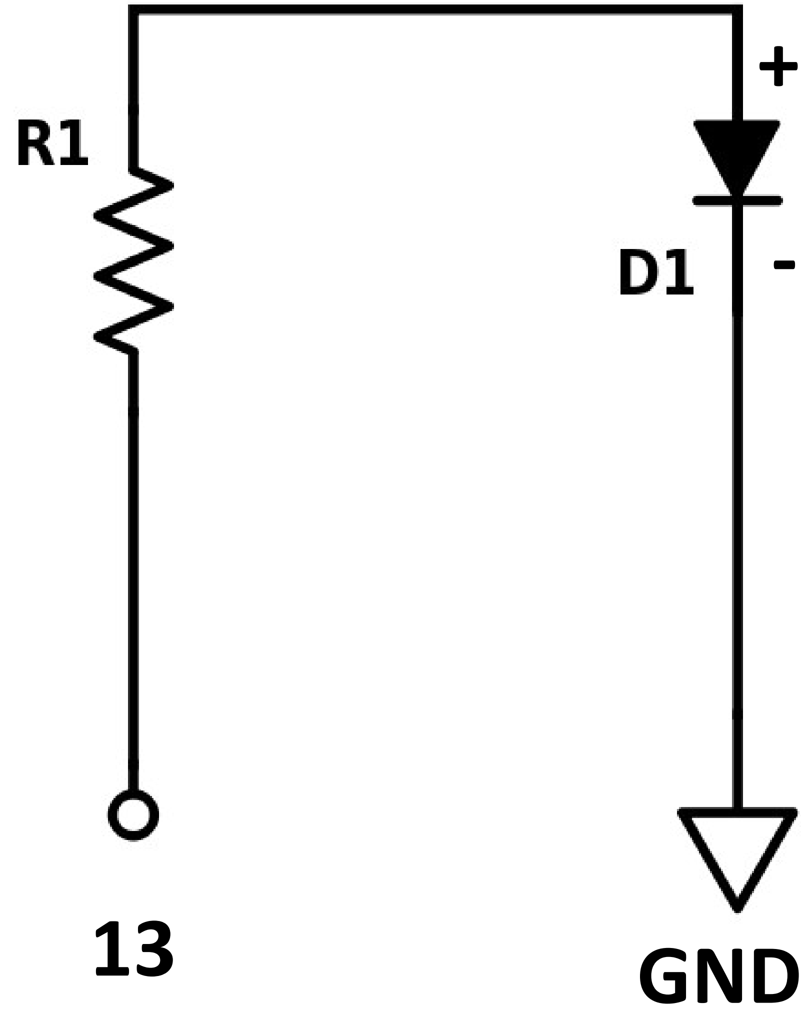


http://www.toptechboy.com/arduino/arduino-lesson-2-using-a-breadboard/



http://www.toptechboy.com/arduino/arduino-lesson-2-using-a-breadboard/

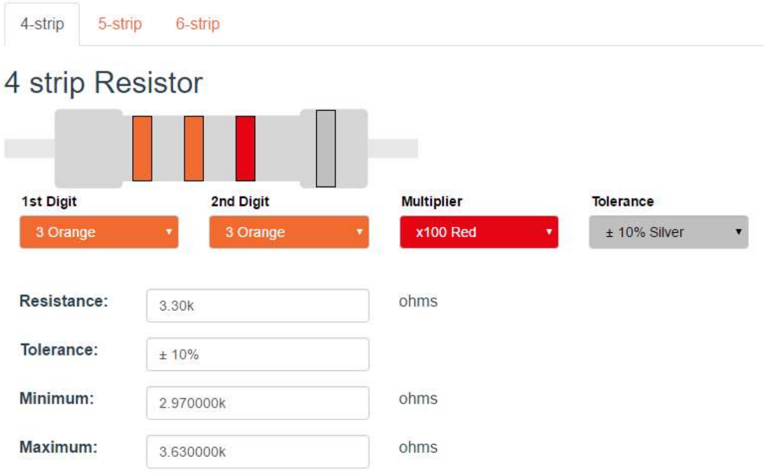
Breadboard



```
int LED = 13;

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

void loop() {
digitalWrite(LED,HIGH);
delay(500);
digitalWrite(LED,LOW);
delay(500);
}
```

Breadboard: why do we use a resistor?

- You have to use a resistor to reduce the current flowing though the LED
- LED is limited by its maximum current capacity
- How to read a 4 strip resistor



4 strip Resistor

1st Digit: 3 Orange

2nd Digit: 3 Orange

Multiplier: x100 Red

Tolerance: ± 10% Silver

Resistance: 3.30k ohms

Tolerance: ± 10%

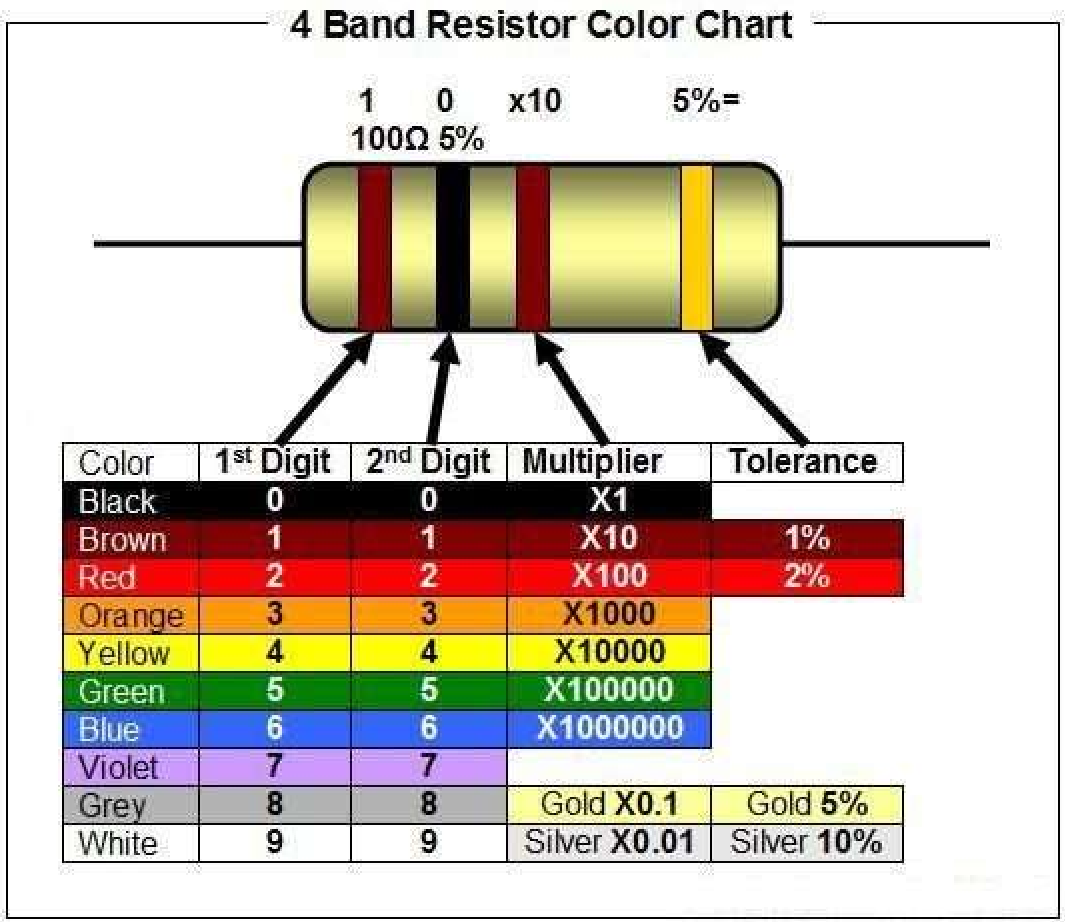
Minimum: 2.970000k ohms

Maximum: 3.630000k ohms

https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/

Breadboard: why do we use a resistor?

- How to read a 4 strip resistor



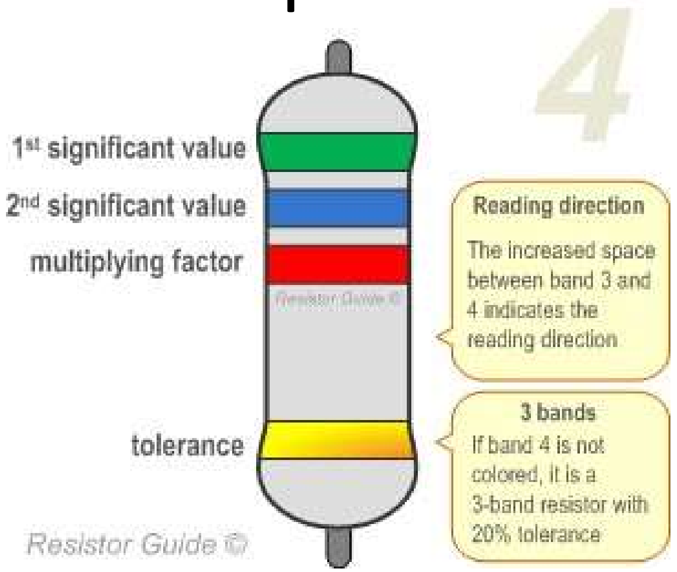
4 Band Resistor Color Chart

1 0 x10 5%=
100Ω 5%

Color	1st Digit	2nd Digit	Multiplier	Tolerance
Black	0	0	X1	
Brown	1	1	X10	1%
Red	2	2	X100	2%
Orange	3	3	X1000	
Yellow	4	4	X10000	
Green	5	5	X100000	
Blue	6	6	X1000000	
Violet	7	7		
Grey	8	8	Gold X0.1	Gold 5%
White	9	9	Silver X0.01	Silver 10%

https://i.pinimg.com/564x/ee/41/de/ee41def7253002c7a6c00315d28459e5--color-charts-amateur-radio.jpg

Example



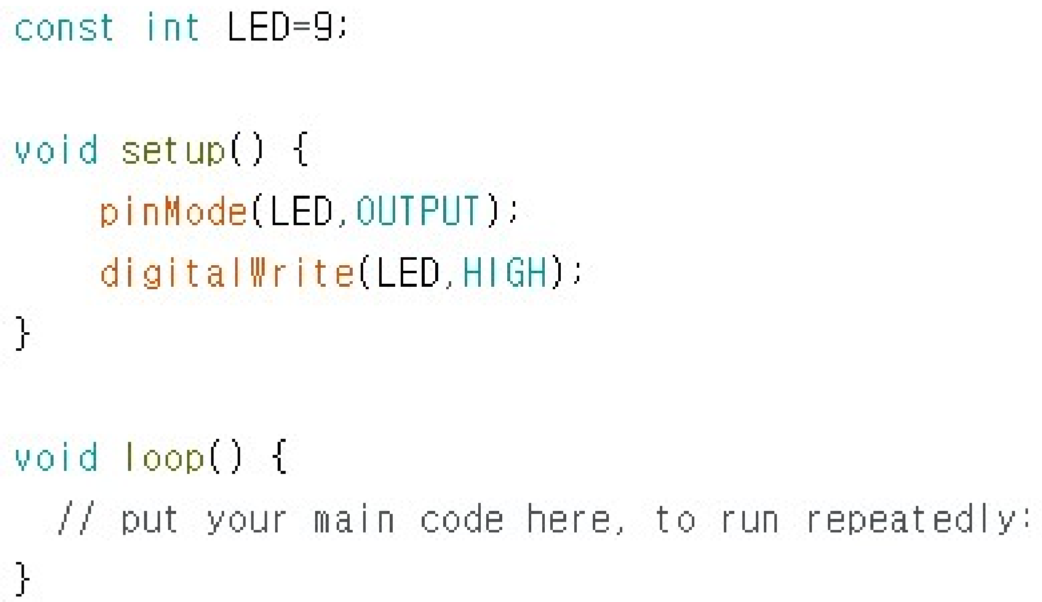
Reading direction
The increased space between band 3 and 4 indicates the reading direction

3 bands
If band 4 is not colored, it is a 3-band resistor with 20% tolerance

5600 Ω

http://www.resistorguide.com/resistor-color-code/

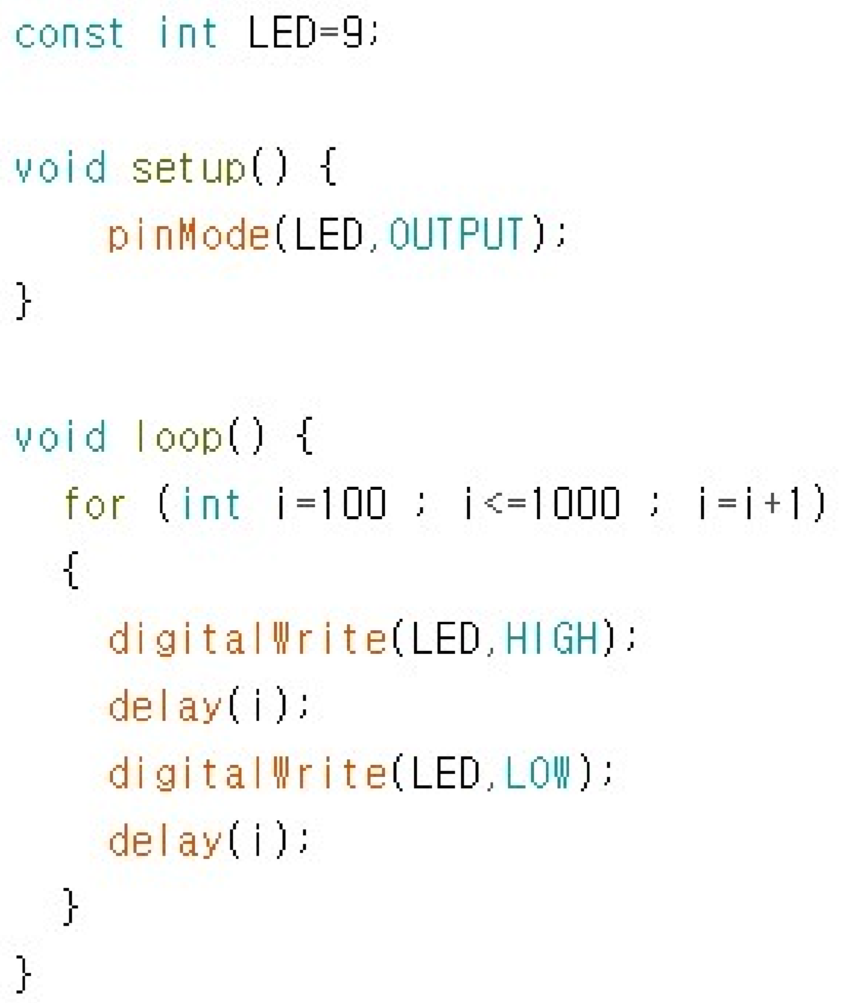
digitalWrite()



```
const int LED=9;

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

void loop() {
  // put your main code here, to run repeatedly:
}
```

- A variable type must be declared.
- General variables can be changed.
- “const” : constant variable → no change
- pinMode: OUTPUT or INPUT
- digitalWrite: HIGH = 5V, LOW = 0V

digitalWrite()

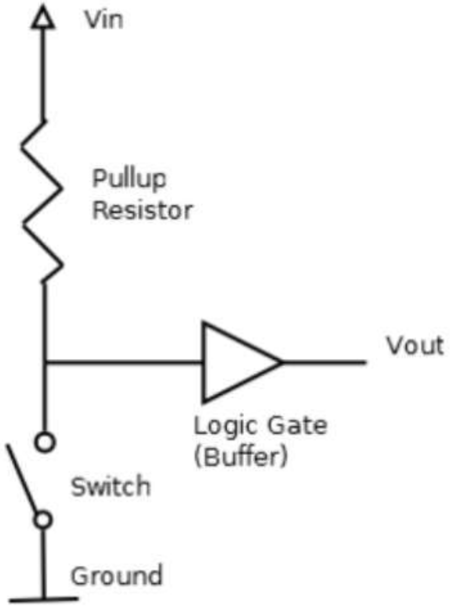


```
const int LED=9;

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

void loop() {
  for (int i=100 ; i<=1000 ; i=i+1)
  {
    digitalWrite(LED,HIGH);
    delay(i);
    digitalWrite(LED,LOW);
    delay(i);
  }
}
```

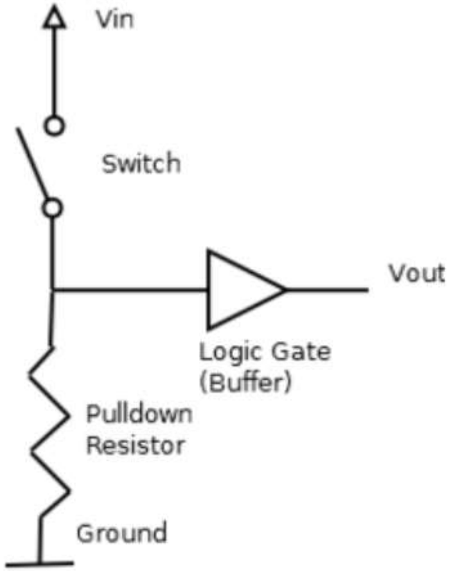
- Delay between blinking is now changing

digitalRead()

- digitalRead() : read HIGH or LOW
- Example: pushbutton, switch
- Noise by unstable button → **floating**
- Use pull-up or pull-down resistor

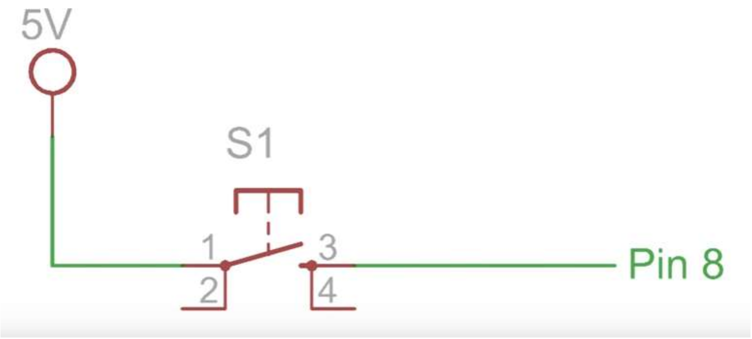


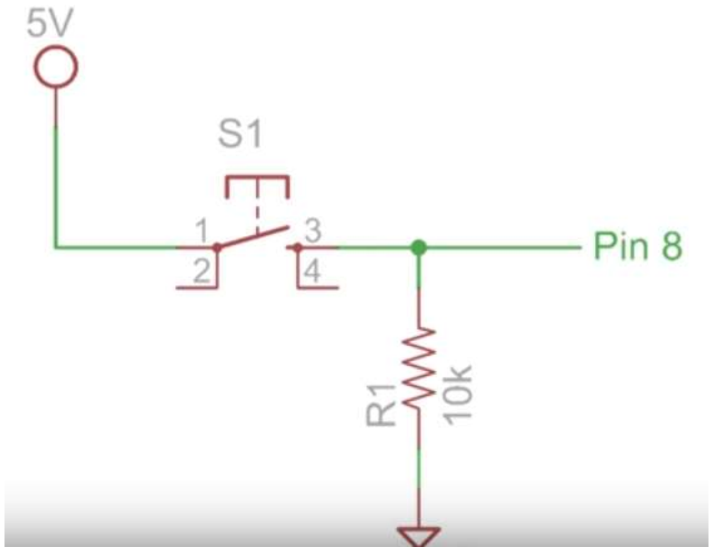
Pull Up Resistor



Pull Down Resistor

Floating & Pull-down registor



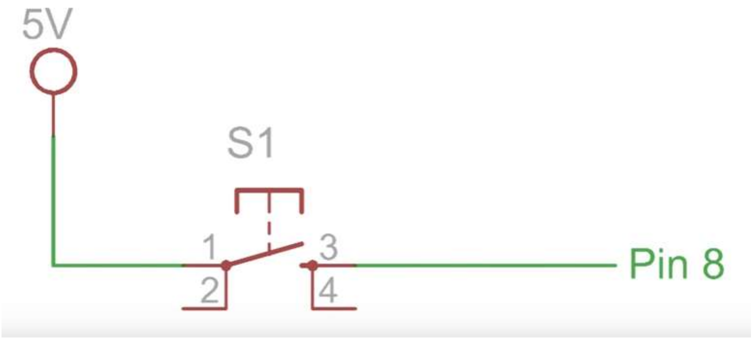


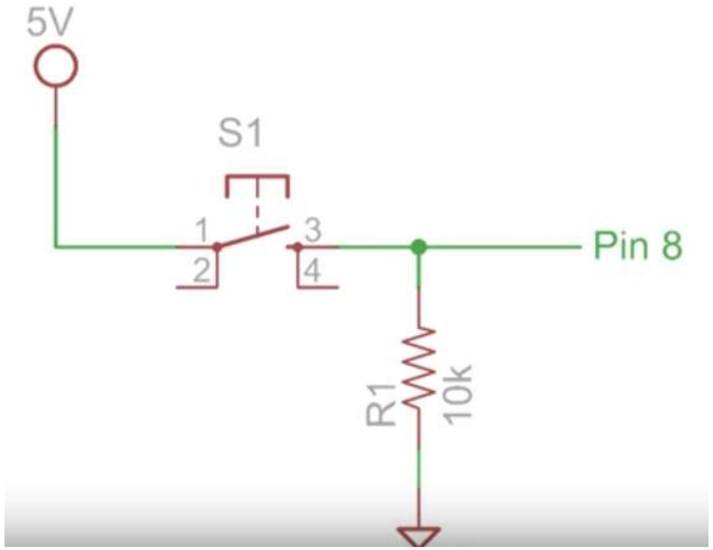
- 스위치가 연결된 상태에서는 핀8에 5V가 전달
- 스위치가 열린 상태에서는 어떻게 되는가?

- 스위치가 열린 상태에서 그라운드와 연결

https://www.youtube.com/watch?v=abWCy_aOSwY&t=770s

Floating & Pull-down registor

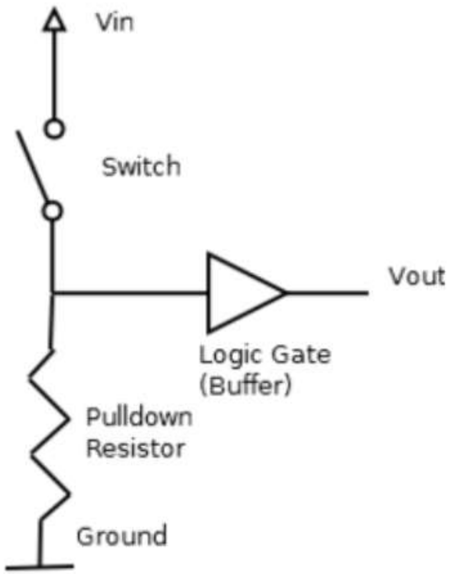




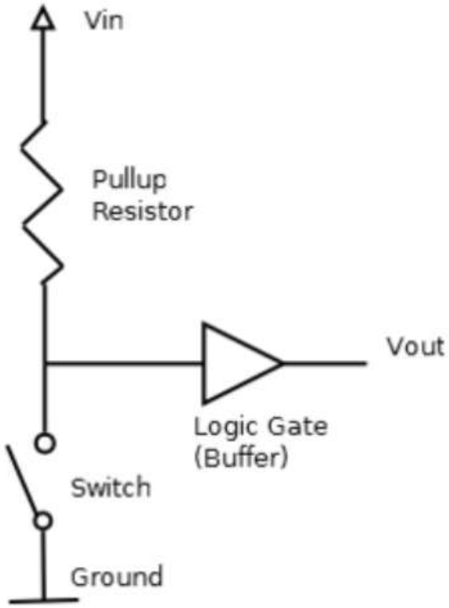
https://www.youtube.com/watch?v=abWCy_aOSwY&t=770s

Button/switch	ON	OFF
Without pull down	5 V (HIGH)	Floating
With pull down	5 V (HIGH)	0 V (LOW)

digitalRead()



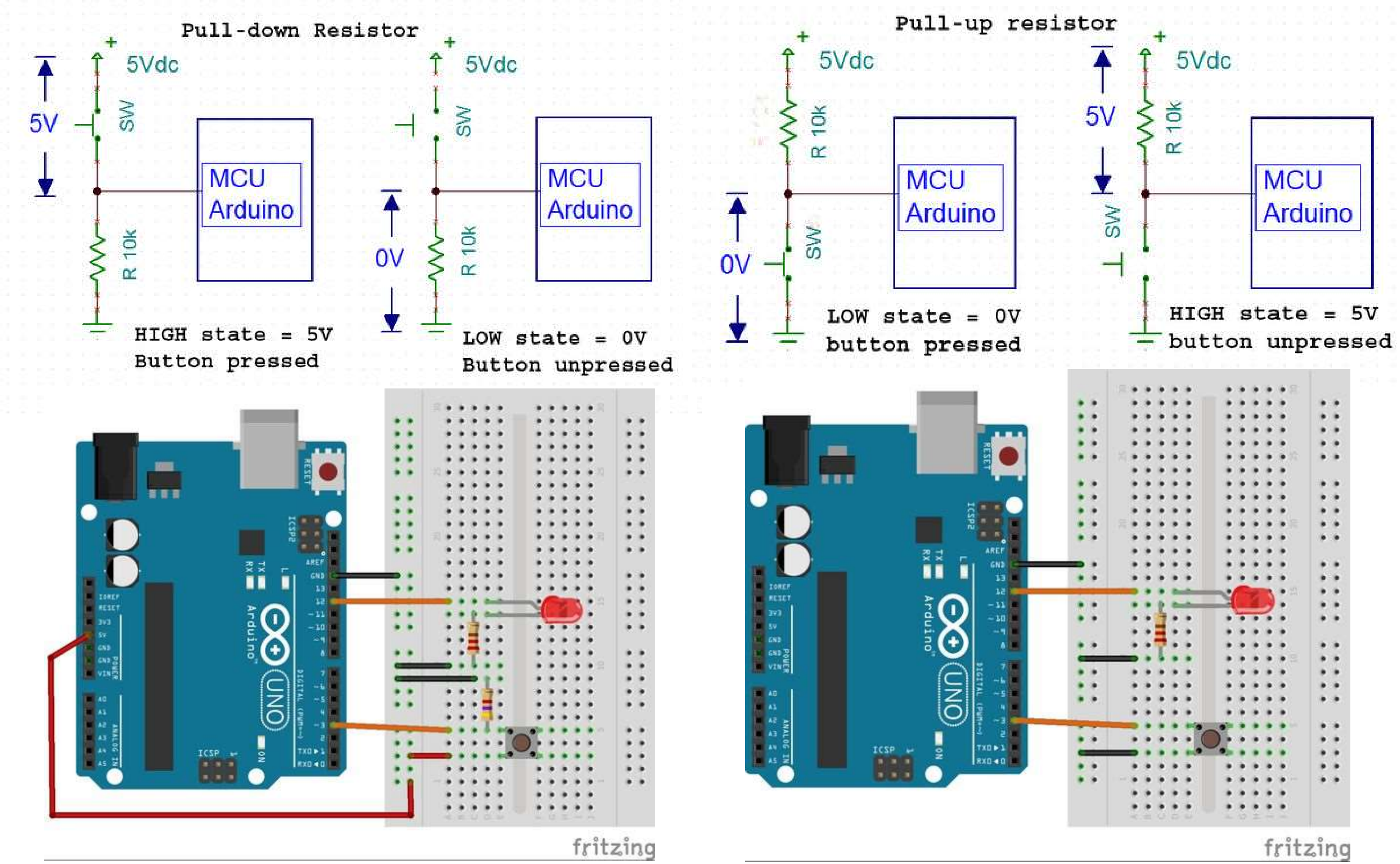
Pull Down Resistor



Pull Up Resistor

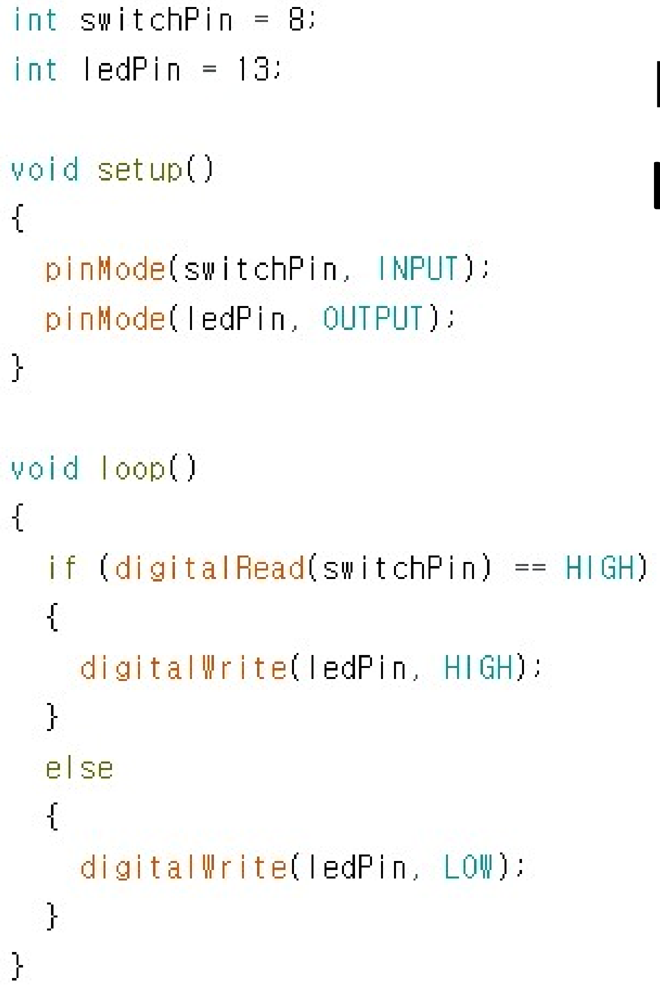
- Pull-up 의 경우
  - On = 0V
  - Off = 5V

Floating & Pull-down registor



http://www.instructables.com/id/Understanding-the-Pull-up-Resistor-With-Arduino/

Control LED with button / switch



```
int switchPin = 8;
int ledPin = 13;

void setup()
{
  pinMode(switchPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  if (digitalRead(switchPin) == HIGH)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

버튼을 찾아 연결 → pin 13 LED를 제어

Bouncing issue will be covered later

https://www.jeremyblum.com/2011/01/10/arduino-tutorial-2-now-with-more-blinky-things/

Control LED with button / switch

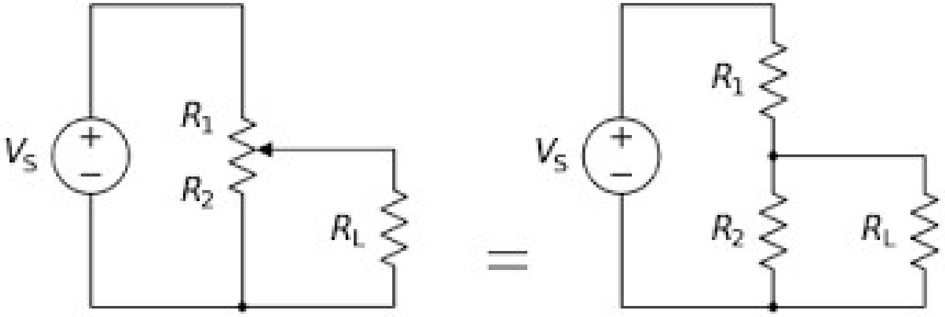
digitalRead : 외부의 디지털 신호를 읽는다.
- 예: 버튼, 스위치

digitalWrite: 디지털 신호를 출력한다.
- 예: LED

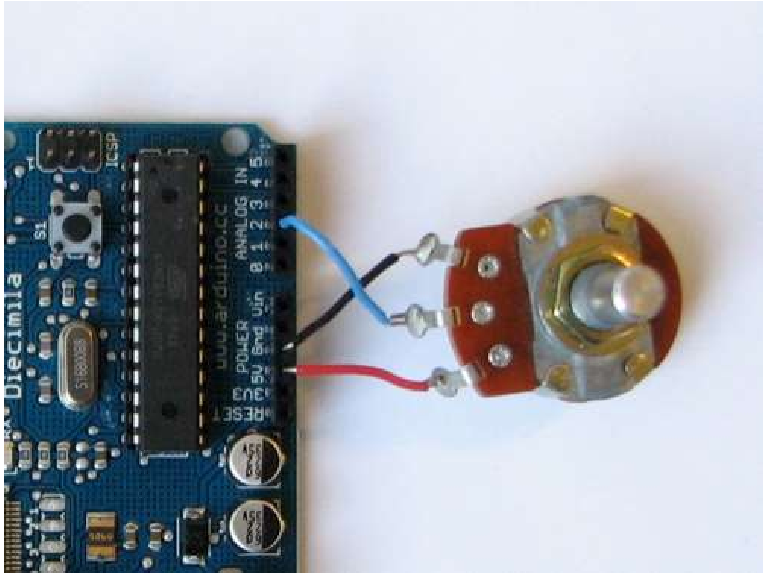
analogRead: 아날로그 신호를 읽는다.

analogWrite: 아날로그 신호를 출력한다.

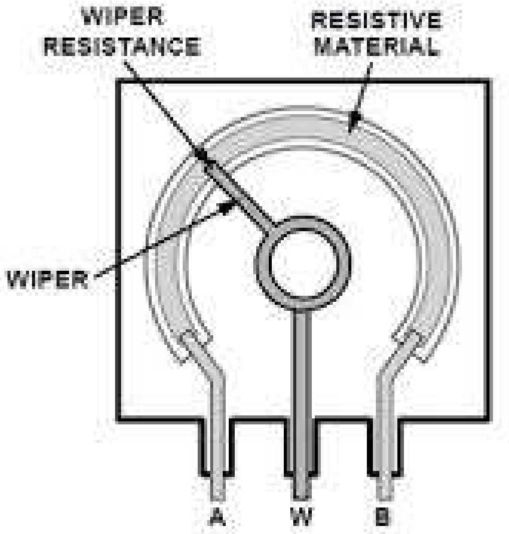
analogRead: Voltage divider & potentiometer



Wikipedia

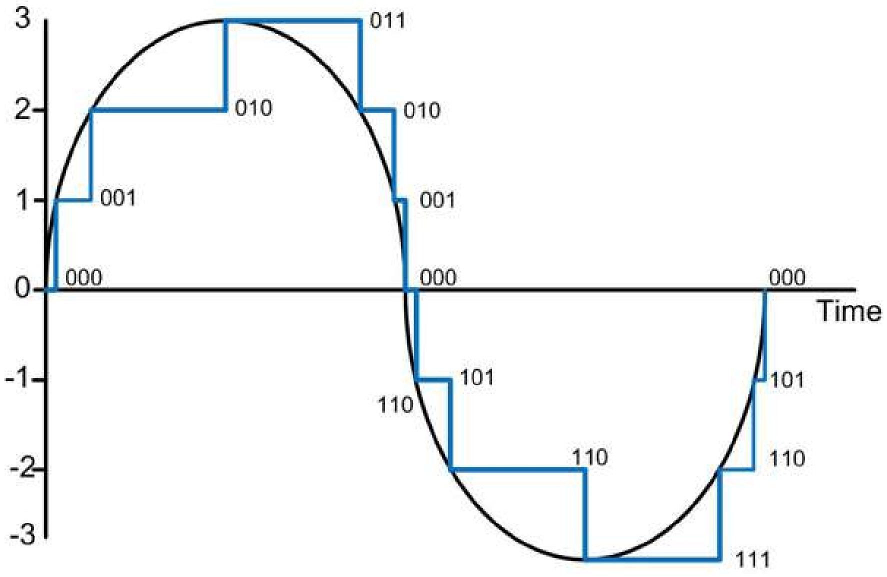


https://www.arduino.cc/en/tutorial/potentiometer

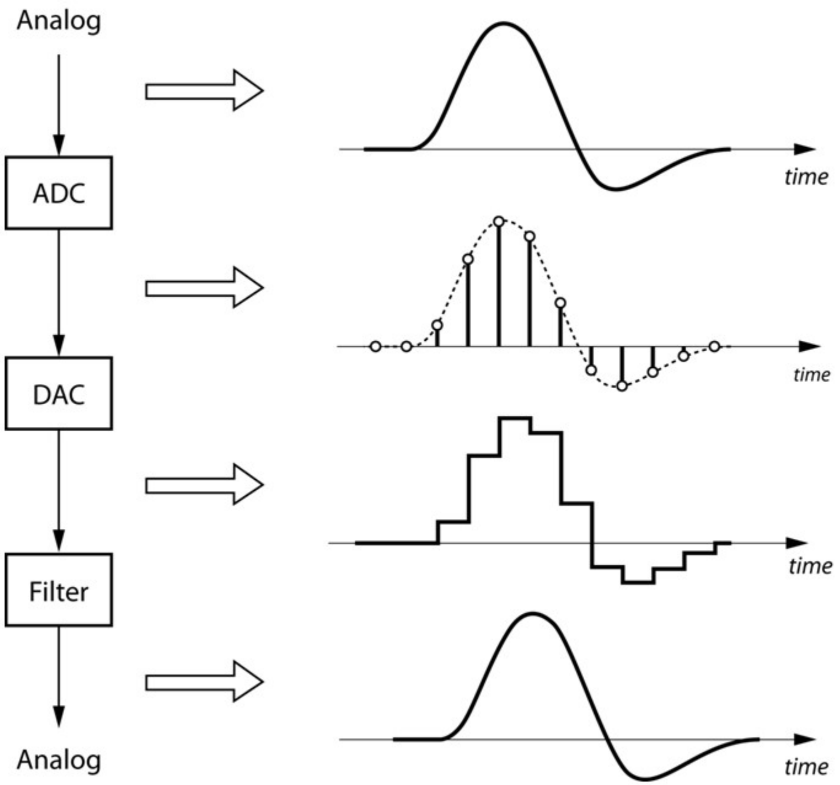


https://www.quora.com/What-is-potentiometer

ADC (analog to digital converter)



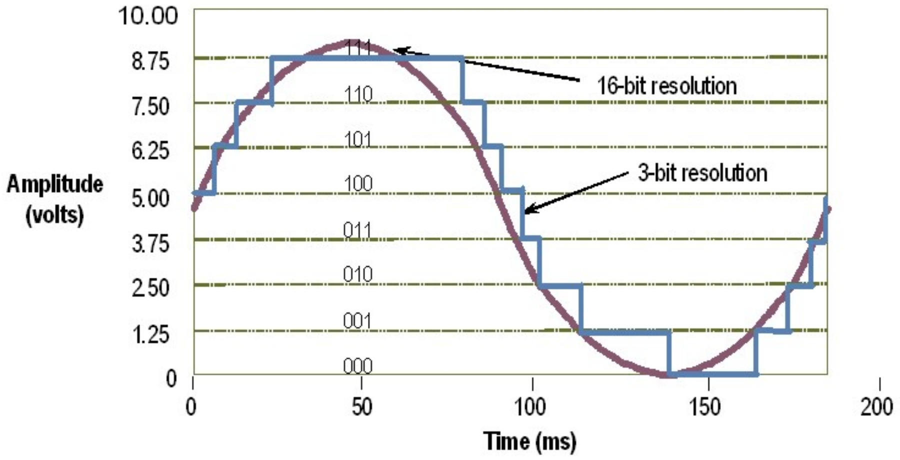
http://screaminfx.com/images/tech-images/what-is-analog-verse-digital-explanation.jpg



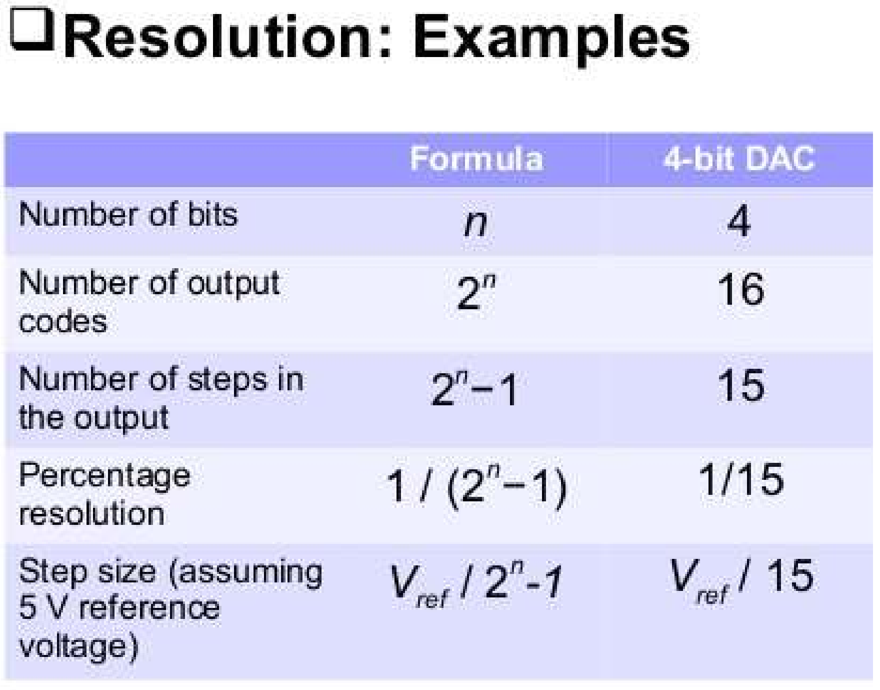
https://www.nutaq.com/sites/default/files/images/blog-images/Process%20of%20digitizing%20and%20converting%20a%20signal%20with%20an%20infinite%20precision%20ADC-DAC_0.png

ADC (analog to digital converter)

What is “resolution”? For example, 10-bit resolution.



http://www.globalspec.com/learnmore/data_acquisition_signal_conditioning/signal_converting/digital_to_analog_converters

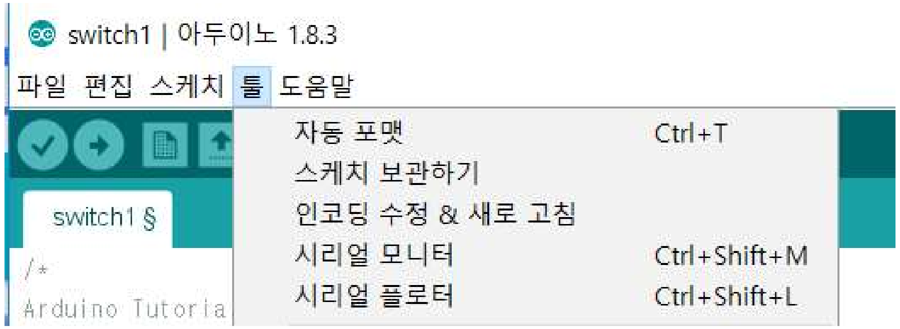


❑Resolution: Examples

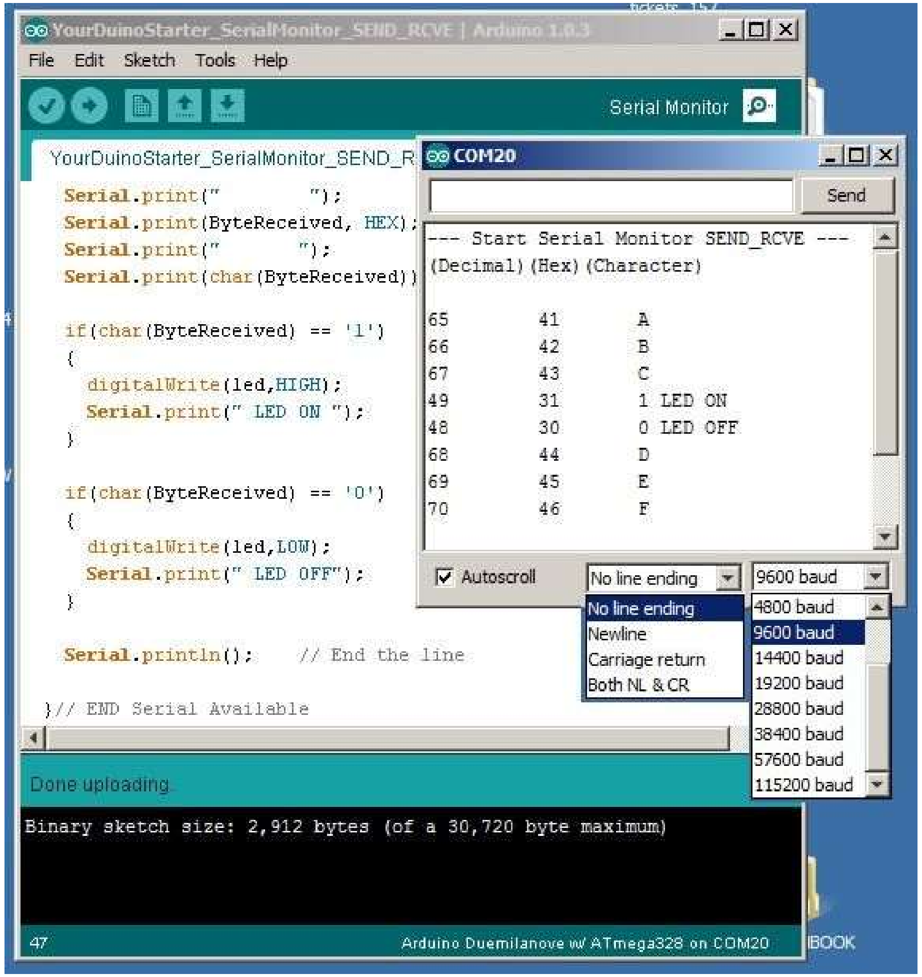
	Formula	4-bit DAC
Number of bits	n	4
Number of output codes	2^n	16
Number of steps in the output	2^n-1	15
Percentage resolution	$1 / (2^n-1)$	1/15
Step size (assuming 5 V reference voltage)	$V_{ref} / 2^n$-1	V_{ref} / 15

https://image.slidesharecdn.com/group1-141018102550-conversion-gate01/95/dacdigital-to-analog-converter-11-638.jpg?cb=1486790152

Serial monitor



```
/*
Arduino Tutoria
```



```
  Serial.print("        ");
  Serial.print(ByteReceived, HEX);
  Serial.print("       ");
  Serial.print(char(ByteReceived))

  if(char(ByteReceived) == '1')
  {
    digitalWrite(led,HIGH);
    Serial.print(" LED ON ");
  }

  if(char(ByteReceived) == '0')
  {
    digitalWrite(led,LOW);
    Serial.print(" LED OFF");
  }

  Serial.println();    // End the line

}// END Serial Available
```

```
--- Start Serial Monitor SEND_RCVE ---
(Decimal)(Hex)(Character)

65        41        A
66        42        B
67        43        C
49        31        1 LED ON
48        30        0 LED OFF
68        44        D
69        45        E
70        46        F
```

☑ Autoscroll — No line ending — 9600 baud

Done uploading.

```
Binary sketch size: 2,912 bytes (of a 30,720 byte maximum)
```

47 — Arduino Duemilanove w/ ATmega328 on COM20

- Baud rate : 9600
- Speed of communication

디버깅할 때 유용하게 사용됨

https://cdn.instructables.com/FR2/1V4Y/HU7GE879/FR2 1V4YHU7GE879.MEDIUM.jpg

Serial monitor

```
int sensePin = 0 ;

void setup() {

   Serial.begin(9600);

}

void loop() {

  Serial.println(analogRead(sensePin));
  delay(500);

}
```

https://www.youtube.com/watch?v=abWCy_aOSwY&t=770s

Serial Communication: functions

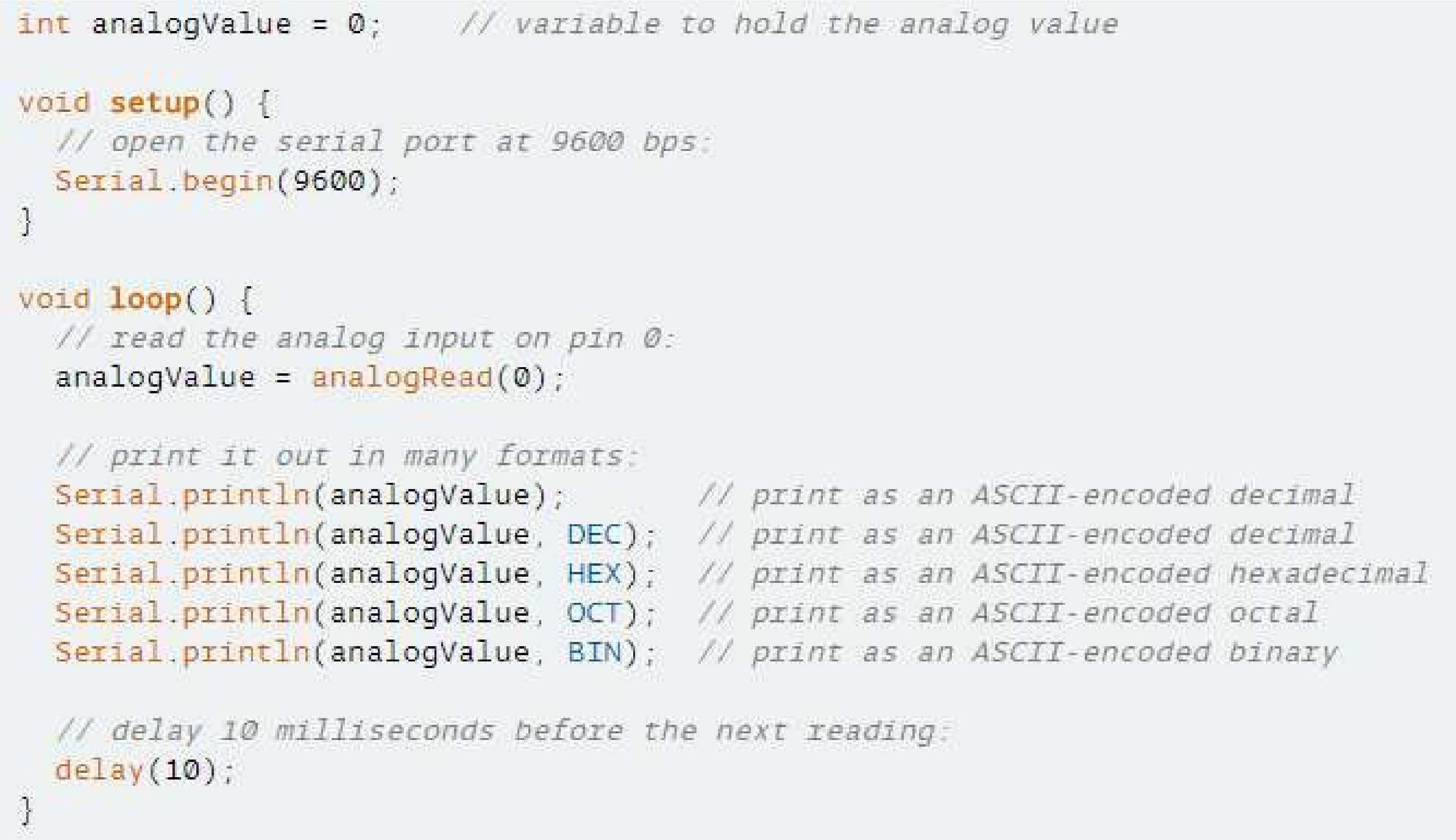
- Serial.begin (baud rate) : define the baud rate
- Serial.print(“text”)
- Serial.println(“text”) : new line

- \n : new line
- \t : tab
- \\n : print \n
- \\ t : print \t

- Exmpale:
  - Serial.println(“\nOut1\tOut2\tOut3”);

Serial Communication

- Choose the data type of printed value
- DEC, HEX, OCT, BIN
- Default is DEC



```
int analogValue = 0;    // variable to hold the analog value

void setup() {
  // open the serial port at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog input on pin 0:
  analogValue = analogRead(0);

  // print it out in many formats:
  Serial.println(analogValue);       // print as an ASCII-encoded decimal
  Serial.println(analogValue, DEC);  // print as an ASCII-encoded decimal
  Serial.println(analogValue, HEX);  // print as an ASCII-encoded hexadecimal
  Serial.println(analogValue, OCT);  // print as an ASCII-encoded octal
  Serial.println(analogValue, BIN);  // print as an ASCII-encoded binary

  // delay 10 milliseconds before the next reading:
  delay(10);
}
```

https://www.arduino.cc/en/Serial/Println

Read potentiometer

```
int InputPin = 0;

void setup() {
pinMode(InputPin, INPUT);
Serial.begin(9600);
}

void loop() {

 Serial.println(analogRead(InputPin));

}
```

- Potentiometer를 조절 해 serial monitor에 나오는 숫자가 어떻게 변화하는지 관찰
- 10-bit resolution → 1024 steps (0 ~ 1023)

Buzzer

```
int buzzer = 7;

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

void loop() {

 tone(buzzer, 5000);
 delay(100);
 noTone(buzzer);
 delay(500);

}
```

소리의 주파수

Buzzer with potentiometer

Aim: turn on the buzzer when the potentiometer value is greater than 500

```
int InputPin = 0;
const int buzzer = 7;

void setup() {
pinMode(InputPin, INPUT);
pinMode(buzzer, OUTPUT);
Serial.begin(9600);
}

void loop() {

 Serial.println(analogRead(InputPin));
 if(analogRead(InputPin)>500) tone(buzzer, 5000);
 else noTone(buzzer);

}
```

Assignment

다음과 같은 동작을 하는 코드를 제작

- 버튼을 누를 때마다 버저의 on / off toggling
    - push the button → turn on the buzzer → push the button → turn off the buzzer
- 버튼을 빵판이 아닌 쉴드에 직접 연결
- Potentiometer에서 읽는 아날로그 숫자를 버저의 소리주파수로 사용한다. 즉 knob을 돌리면 소리의 높낮이가 달라진다.

Hint : Jeremy Blum's tutorial #2, Youtube

- **제출 요령**
    - 실행 가능한 파일 및 폴더만 제출
    - 버튼의 연결 핀 = 5
    - Potentiometer의 연결 핀 = A0
    - 버저의 연결 핀 = 8
    - 제출 기한: 화요일 수업 전까지 (4/10)