ST SMART

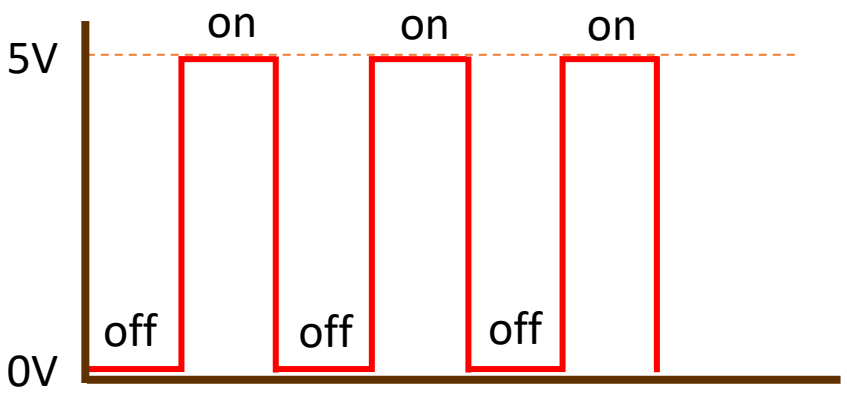
MAKE EVERYTHING SMART

SMART TECHNOLOGY

LECTURE

1

• Signal Types



Digital signal

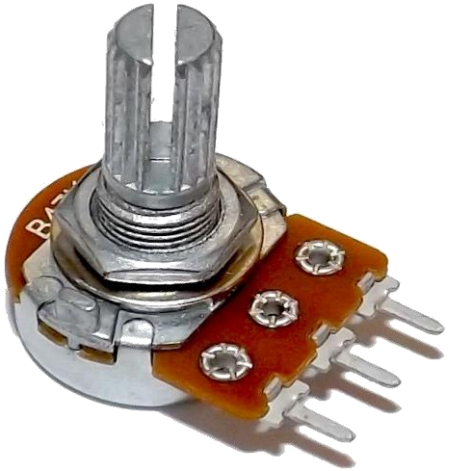
Analog signal

• Some Digital Signal app

Fire detection sensor

Movement
detection sensor

• Some Analog Signal app



Variable resistance

Light Sensor

Smoke Sensor

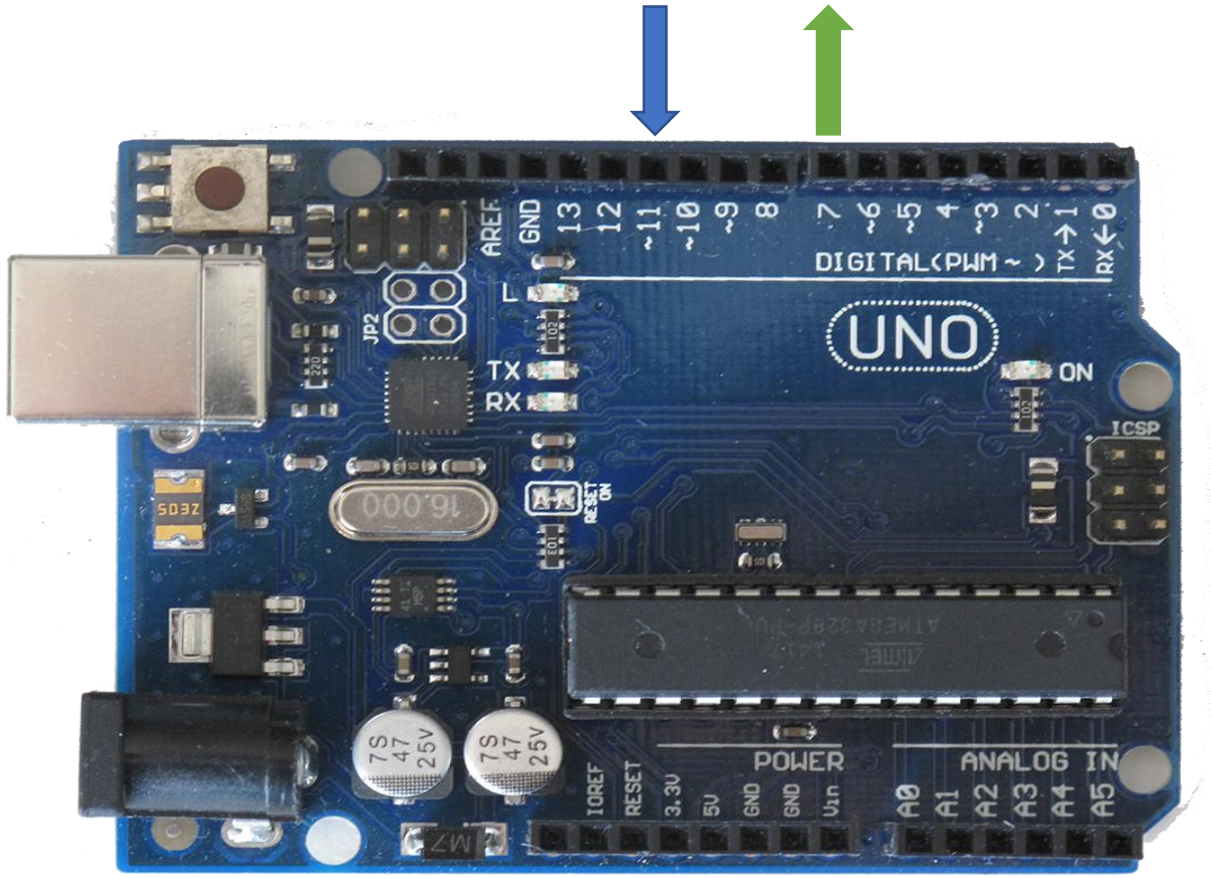
• Arduino Interfacing

1. Make Arduino OUTPUTs a digital signal on pin no. 7

```
void setup() {
  pinMode(7,OUTPUT) ;
 //OR pinMode(7,1);
}
```

2. Make Arduino receives a digital INPUT signal on pin no. 11

```
void setup() {
  pinMode(11,INPUT) ;
 //OR pinMode(7,0);
}
```



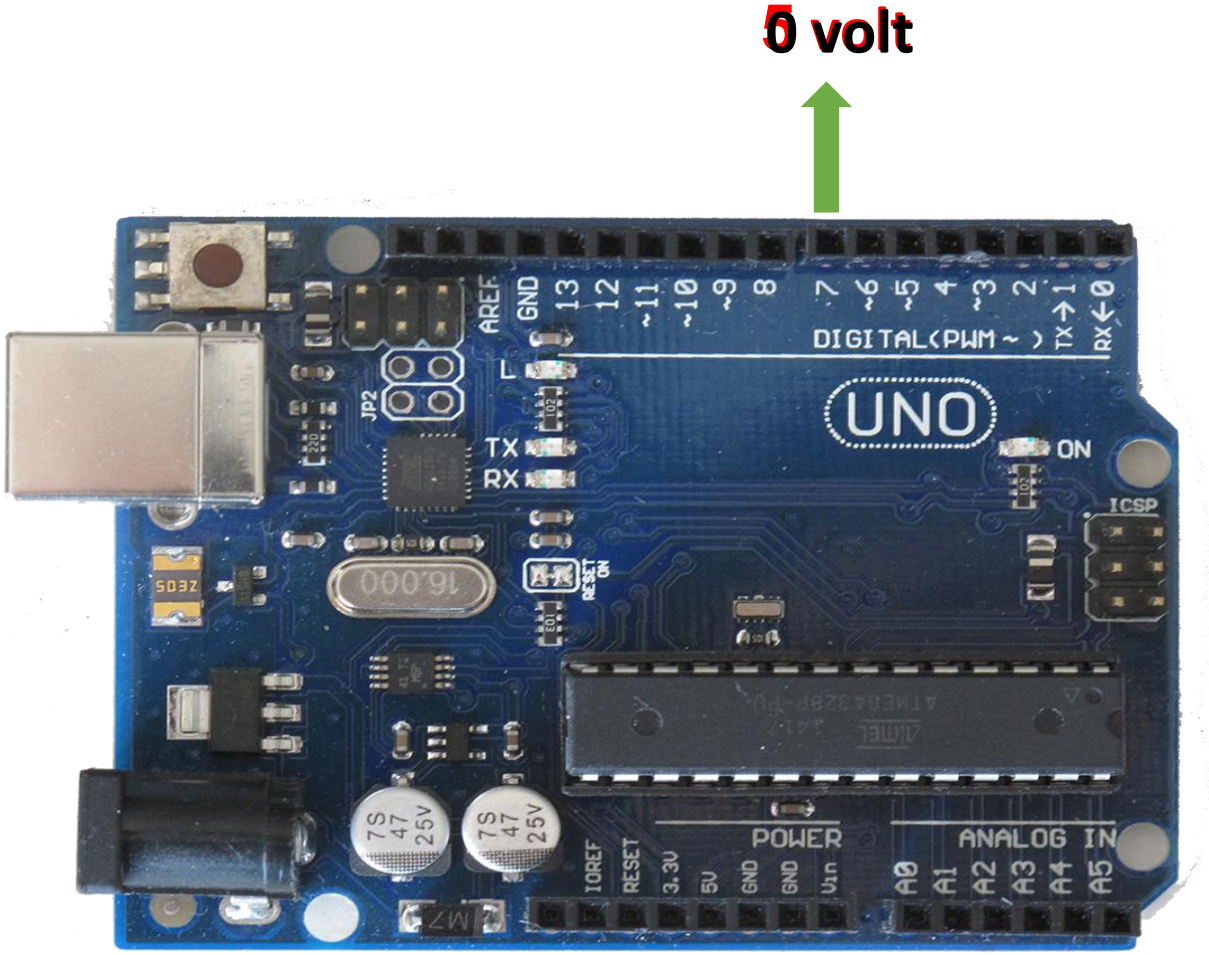
• Arduino OUTPUT signal

1. Make Arduino OUTPUTs a 5 volt signal on pin no. 4

```
void setup() {
  pinMode(7,OUTPUT);
  digitalWrite(7,HIGH);
 //digitalWrite(7,1);
}
```

1. Make Arduino OUTPUTs a 0 volt signal on pin no. 4

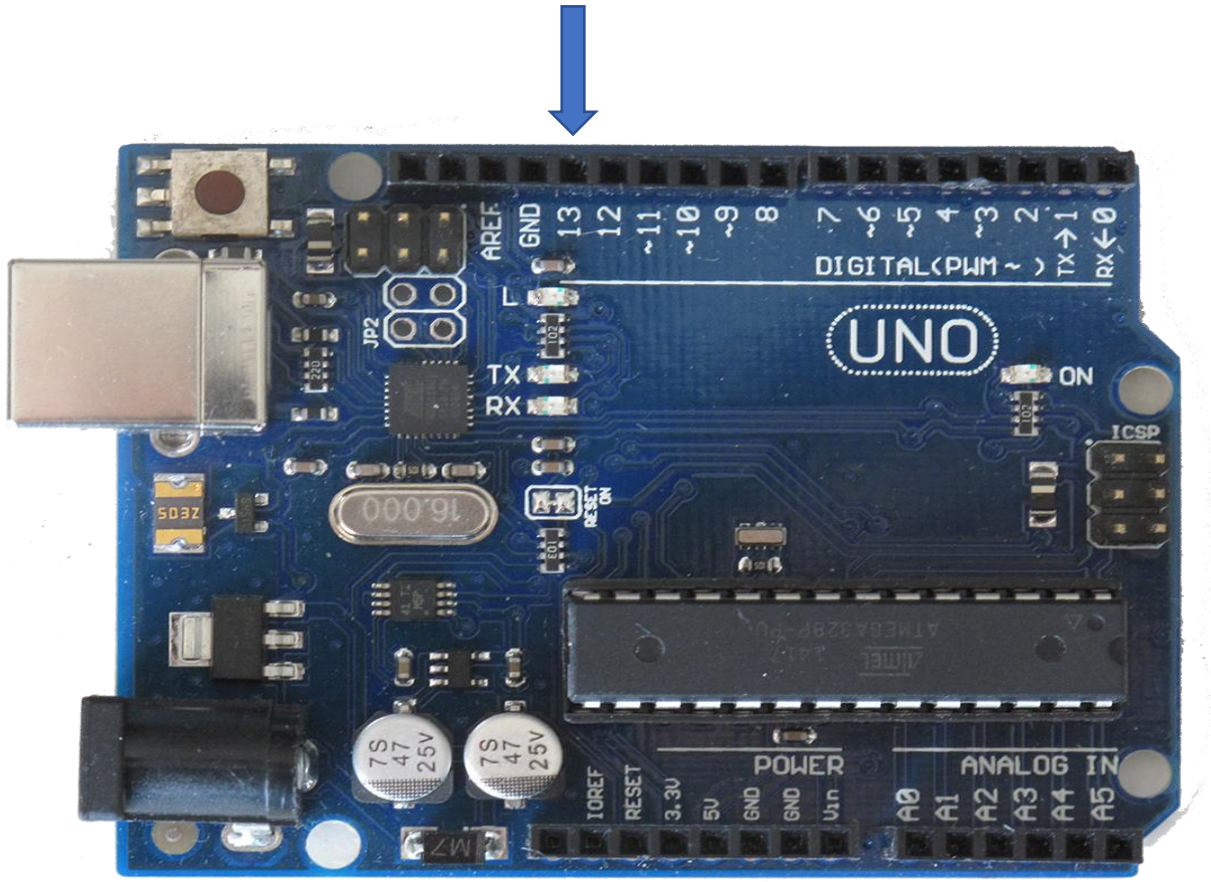
```
void setup() {
  pinMode(7,OUTPUT);
  digitalWrite(7,LOW);
 //digitalWrite(7,0);
}
```



• Arduino INPUT signal

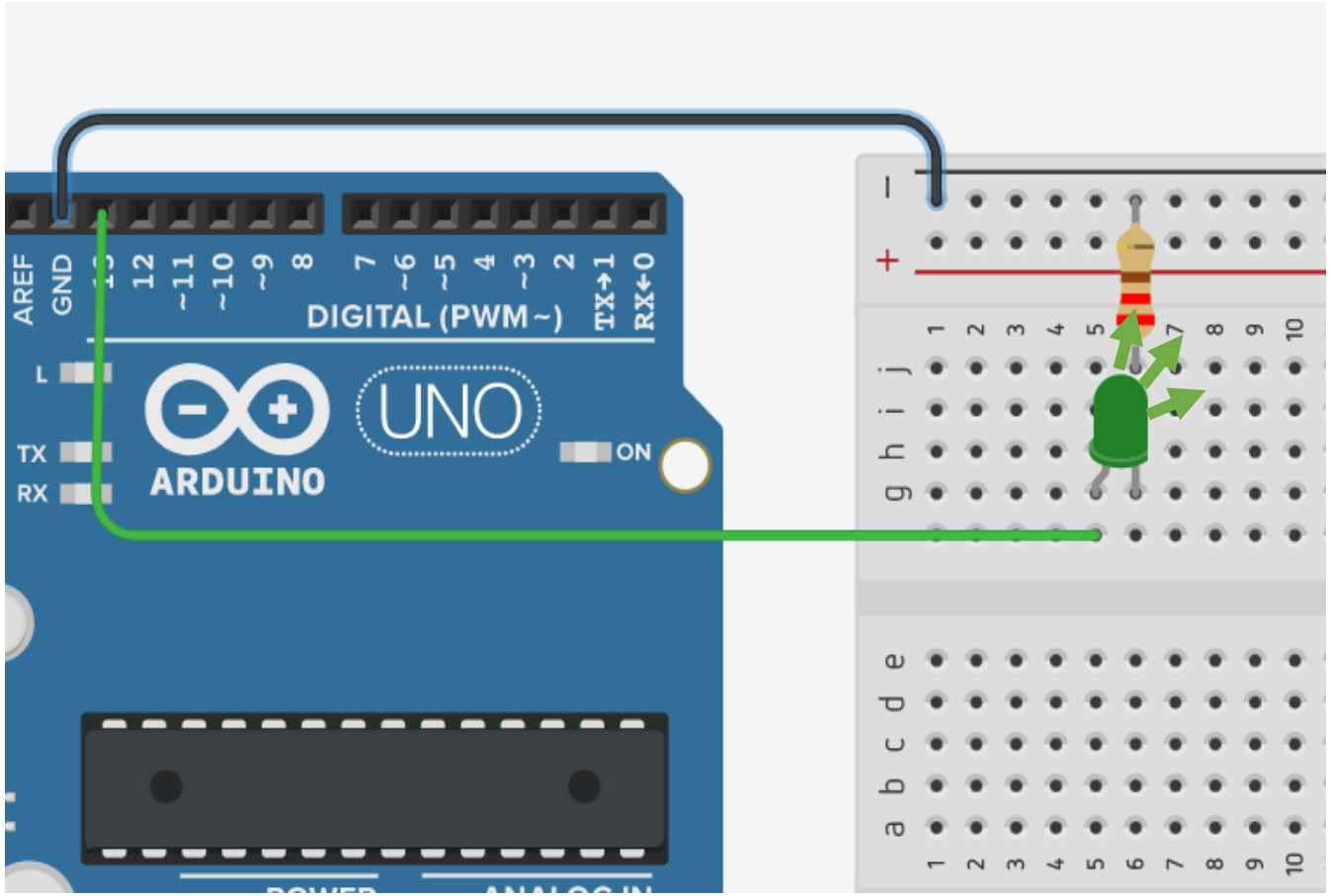
1. Make Arduino Reads an INPUT signal on pin no. 13

```
void setup() {
  pinMode(13,INPUT);
  digitalRead(13);
}
```



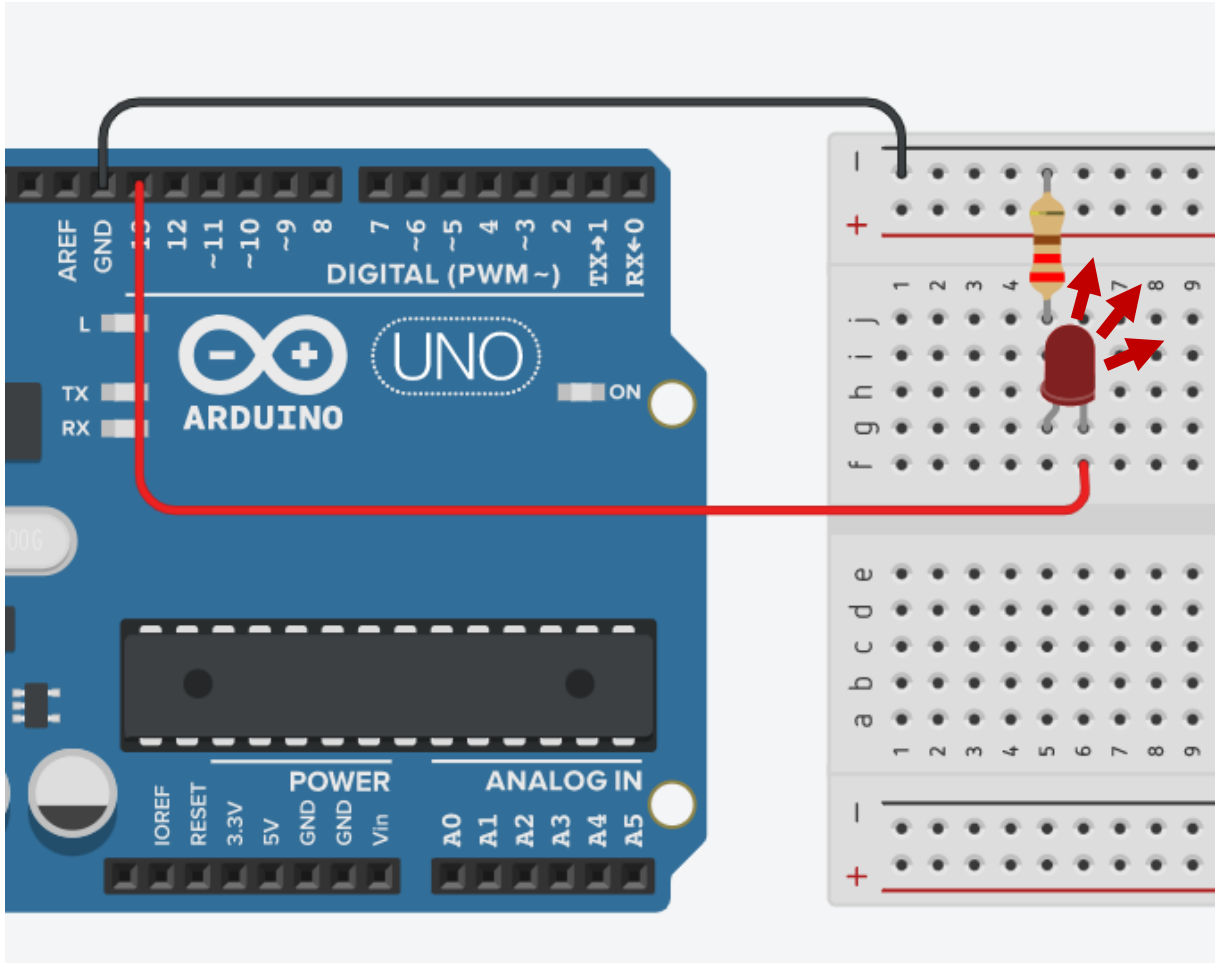
• Blink code

```
int ledPin =  13;

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

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

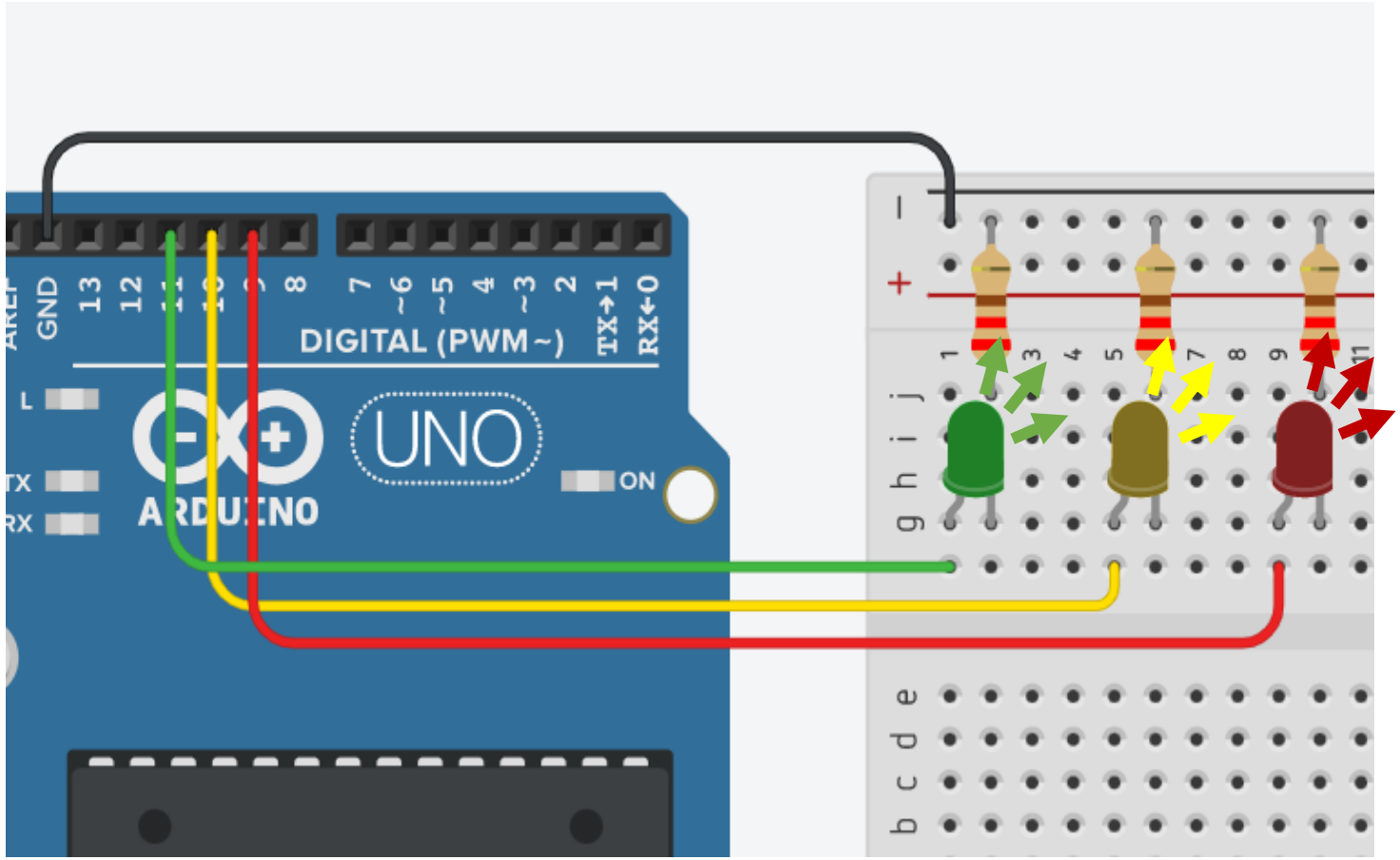


• Another way of coding

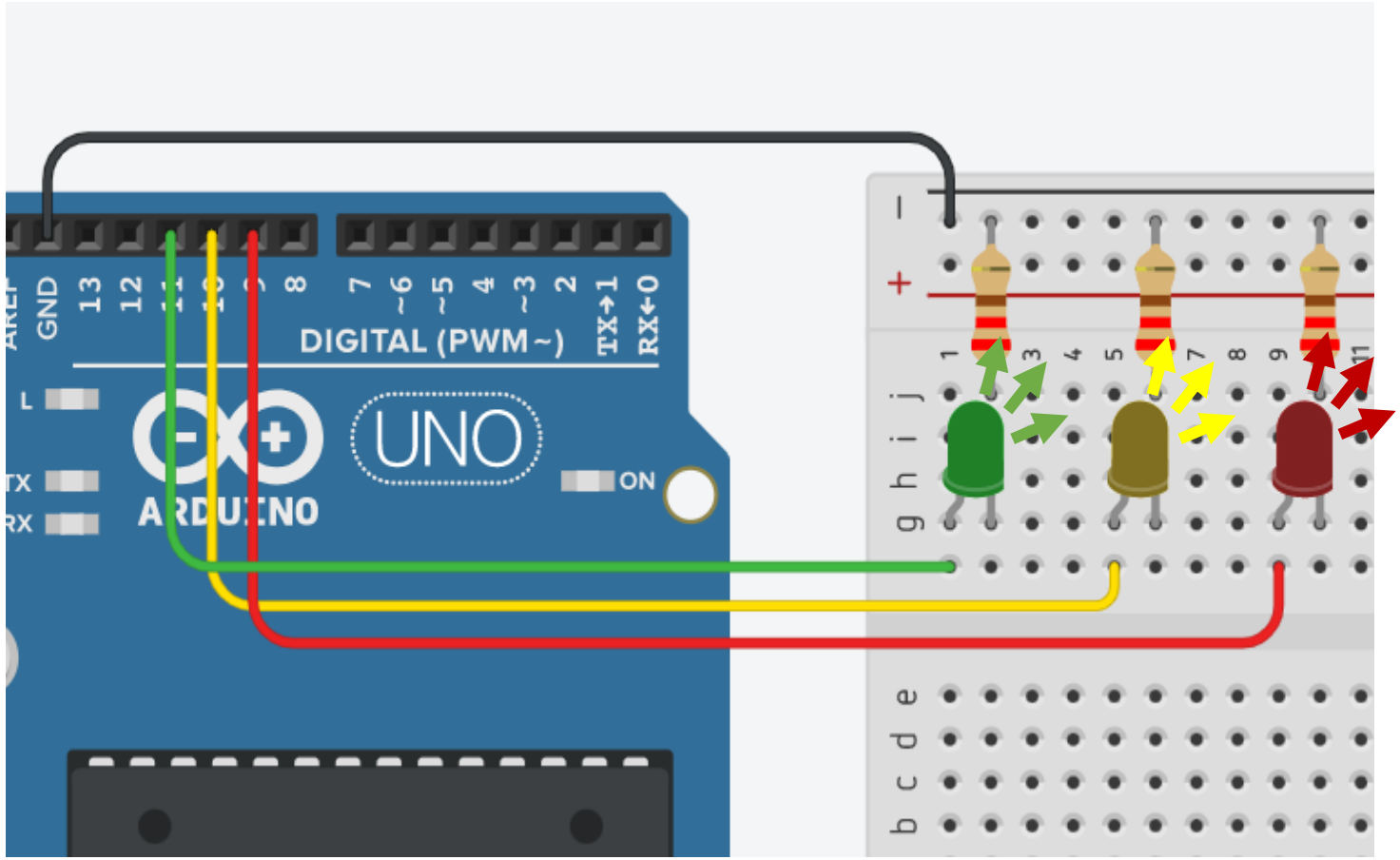
```
int ledPin =  13;
int on =  1000;
int off = 1000;
void setup()
{
  pinMode(ledPin , OUTPUT);
}

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



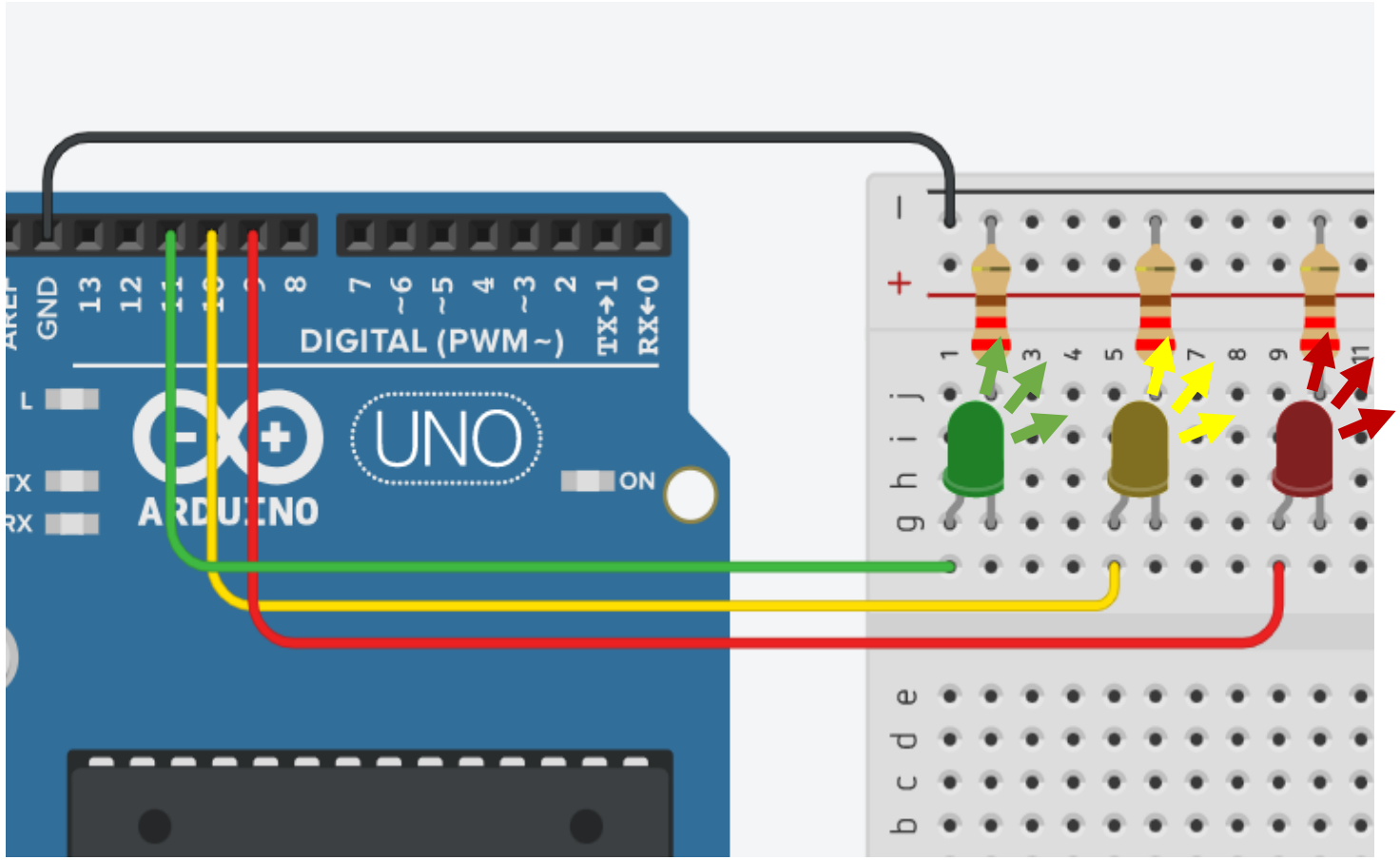
```
int red = 9;
int yellow = 10;
int green = 11;
int ON = 1000;
int OFF = 1000;

void setup()
{
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop()
{
 digitalWrite(red, HIGH);
 delay(ON);
 digitalWrite(red, LOW);
 delay(OFF);
 digitalWrite(yellow, HIGH); delay(ON); digitalWrite(yellow, LOW); delay(OFF);
 digitalWrite(green, HIGH);  delay(ON); digitalWrite(green, LOW); delay(OFF);
}
```



```ino
int red = 9;
int yellow = 10;
int green = 11;
int ON = 1000;
int OFF = 1000;

void setup()
{
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
void loop()
{
 digitalWrite(red, HIGH);
 digitalWrite(yellow,HIGH)
 digitalWrite(green, HIGH);
 delay(ON);
 digitalWrite(red, LOW);
 digitalWrite(yellow, LOW);
 digitalWrite(green, LOW);
 delay(OFF);
}
```

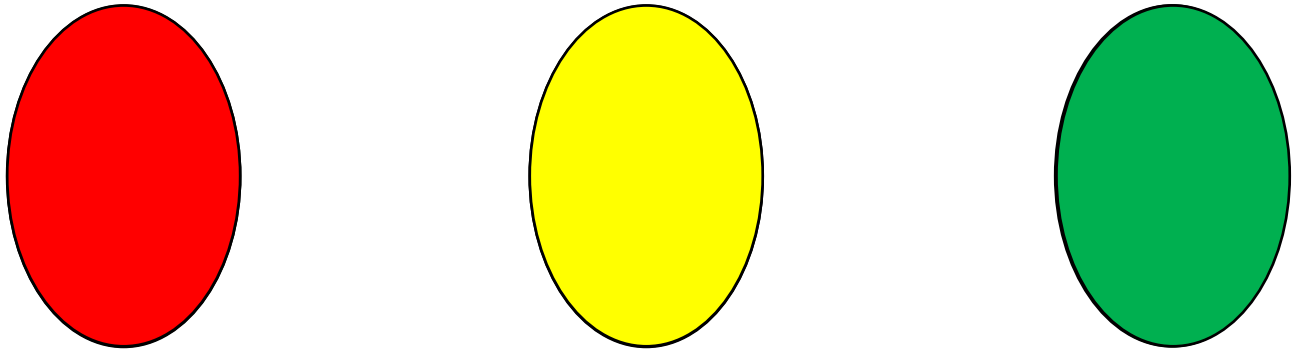


```
int red = 9;
int yellow = 10;
int green = 11;
int ON = 1000;
int OFF = 1000;

void setup()
{
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop()
{
 digitalWrite(red, HIGH);
 delay(ON);
 digitalWrite(yellow,HIGH);
 delay(ON);
 digitalWrite(green, HIGH); delay(ON);   digitalWrite(red, LOW);     delay(OFF);
 digitalWrite(yellow,LOW); delay(Off); digitalWrite(green, LOW); delay(OFF);
}
```



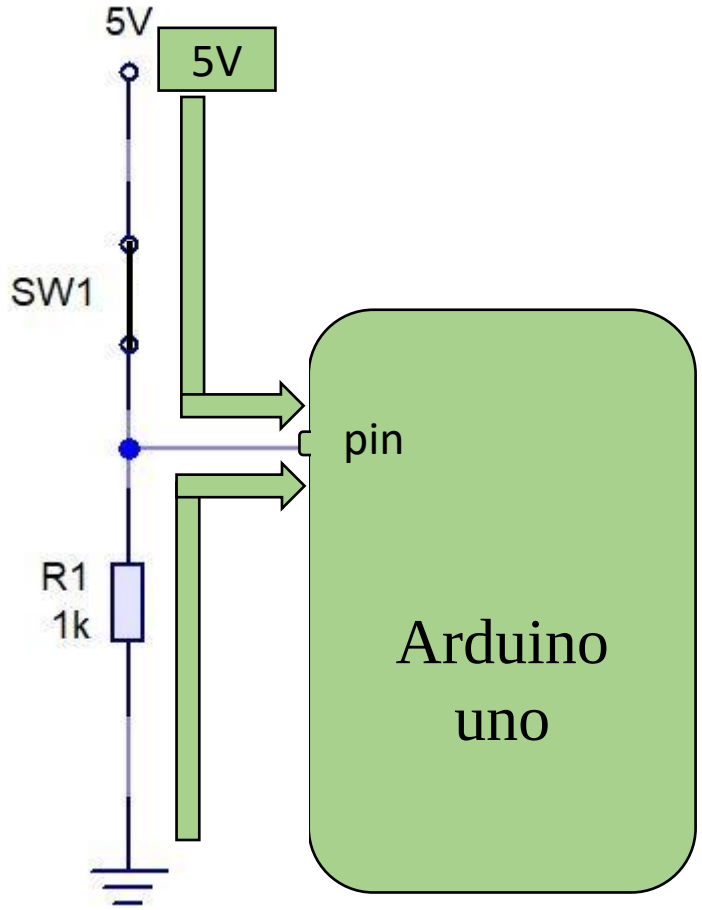
• Task

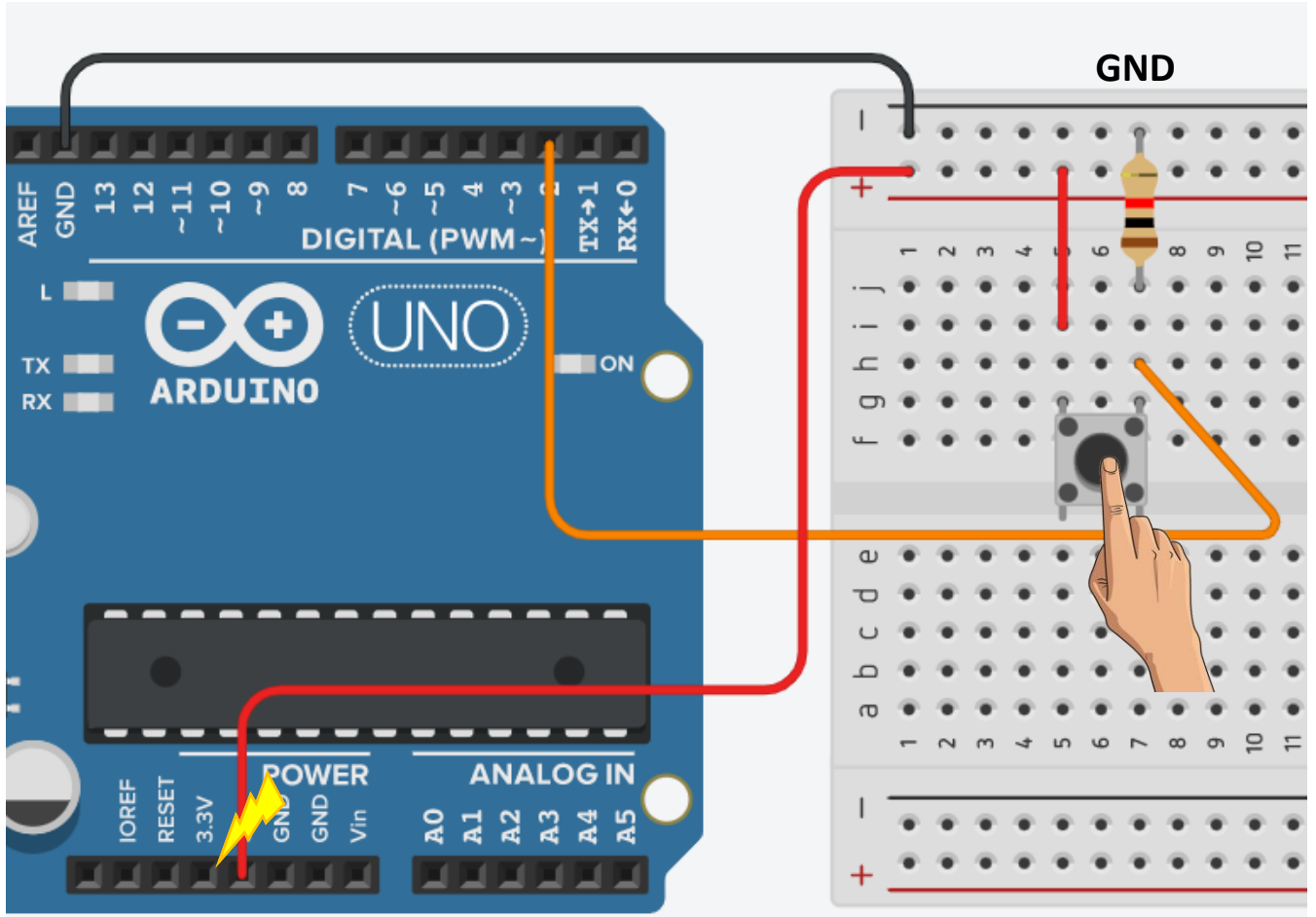
Traffic Lights



• Push Button

“Pull Down Resistor Connection”

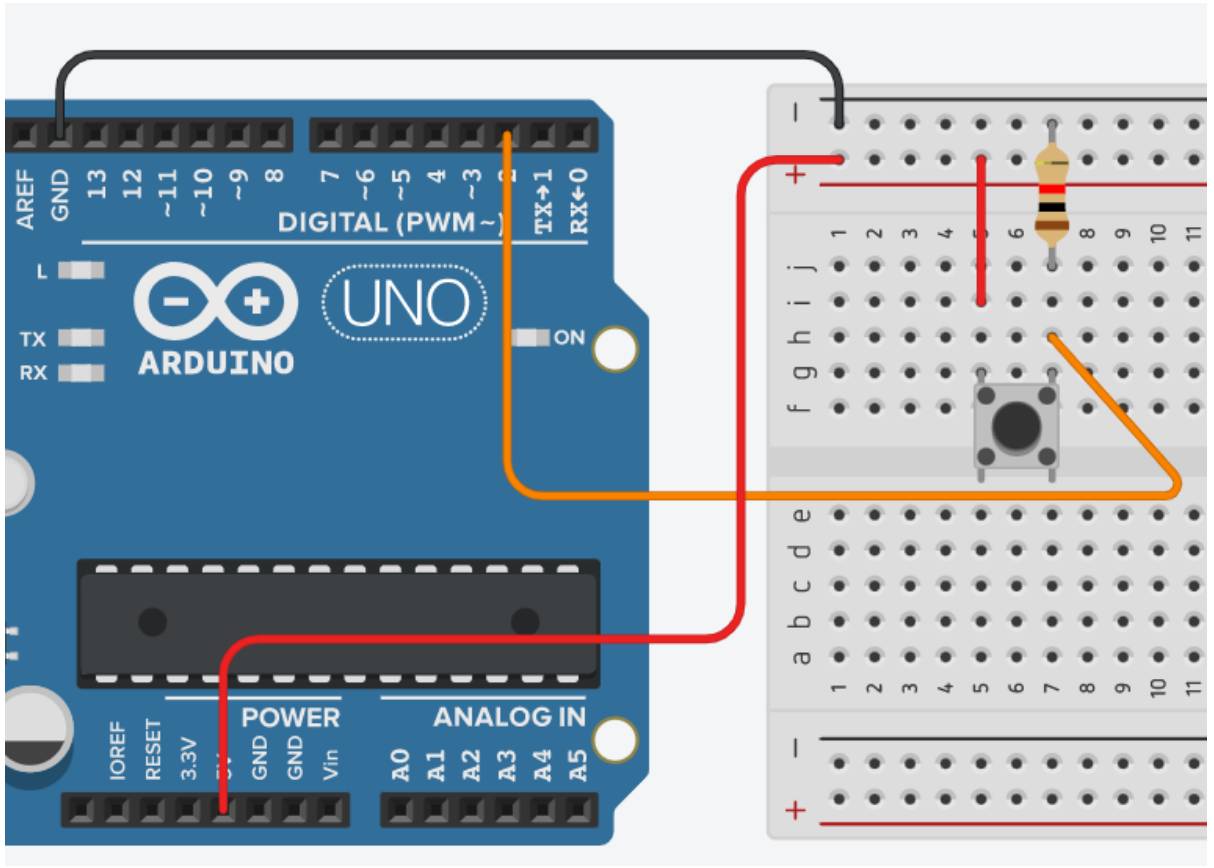




• Code

```
void setup()
{  pinMode(7, INPUT);
   Serial.begin(9600); }

void loop()
{  Serial.println(digitalRead(7)); }
```

```
bool reading;
void setup()
{  pinMode(7, INPUT);
   Serial.begin(9600);}

void loop()
{    reading = digitalRead(7);
     Serial.println(reading);  }
```

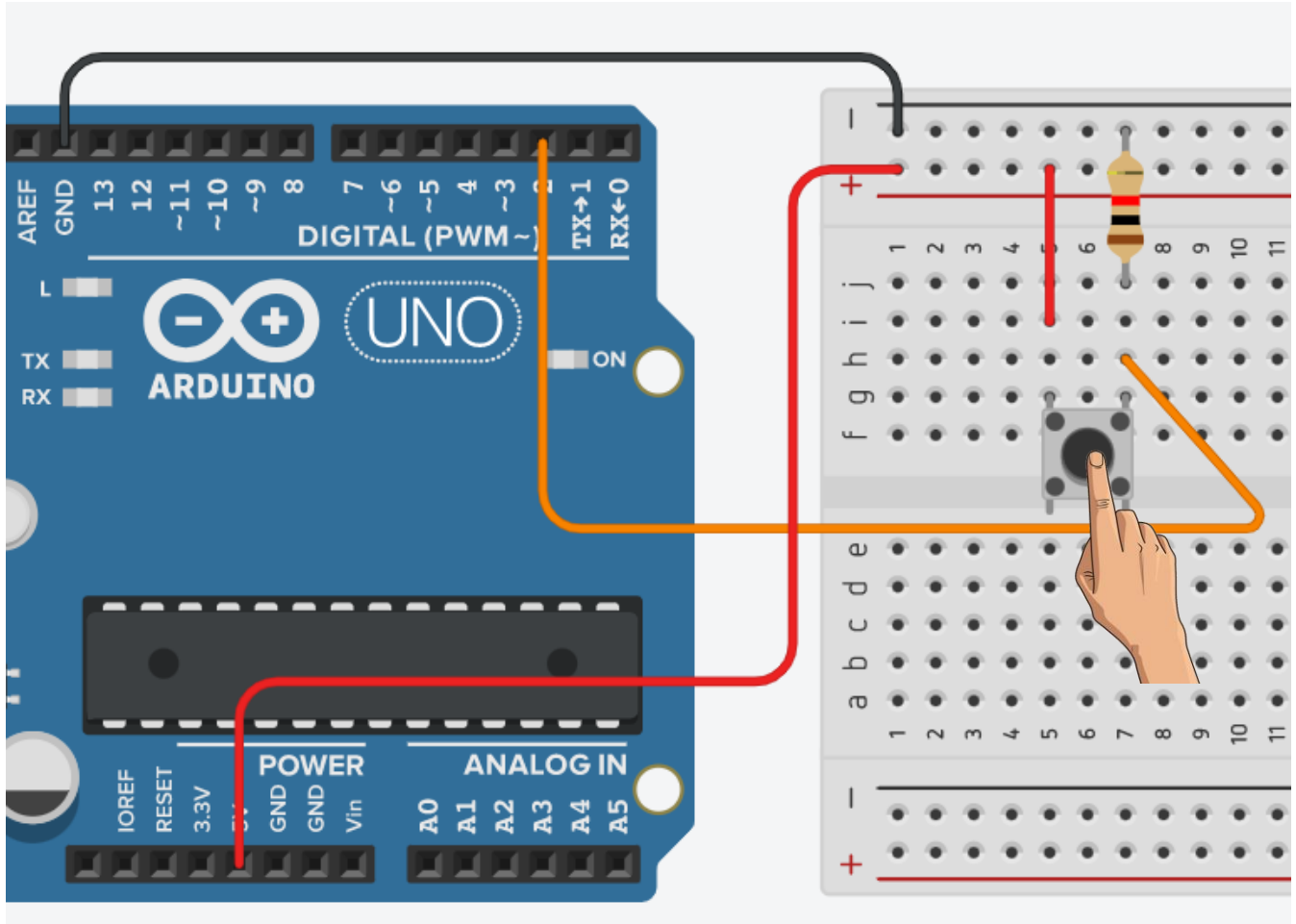


• Task

- Make a counter on the serial monitor that increases by one every time you hit the pushbutton

counter

3

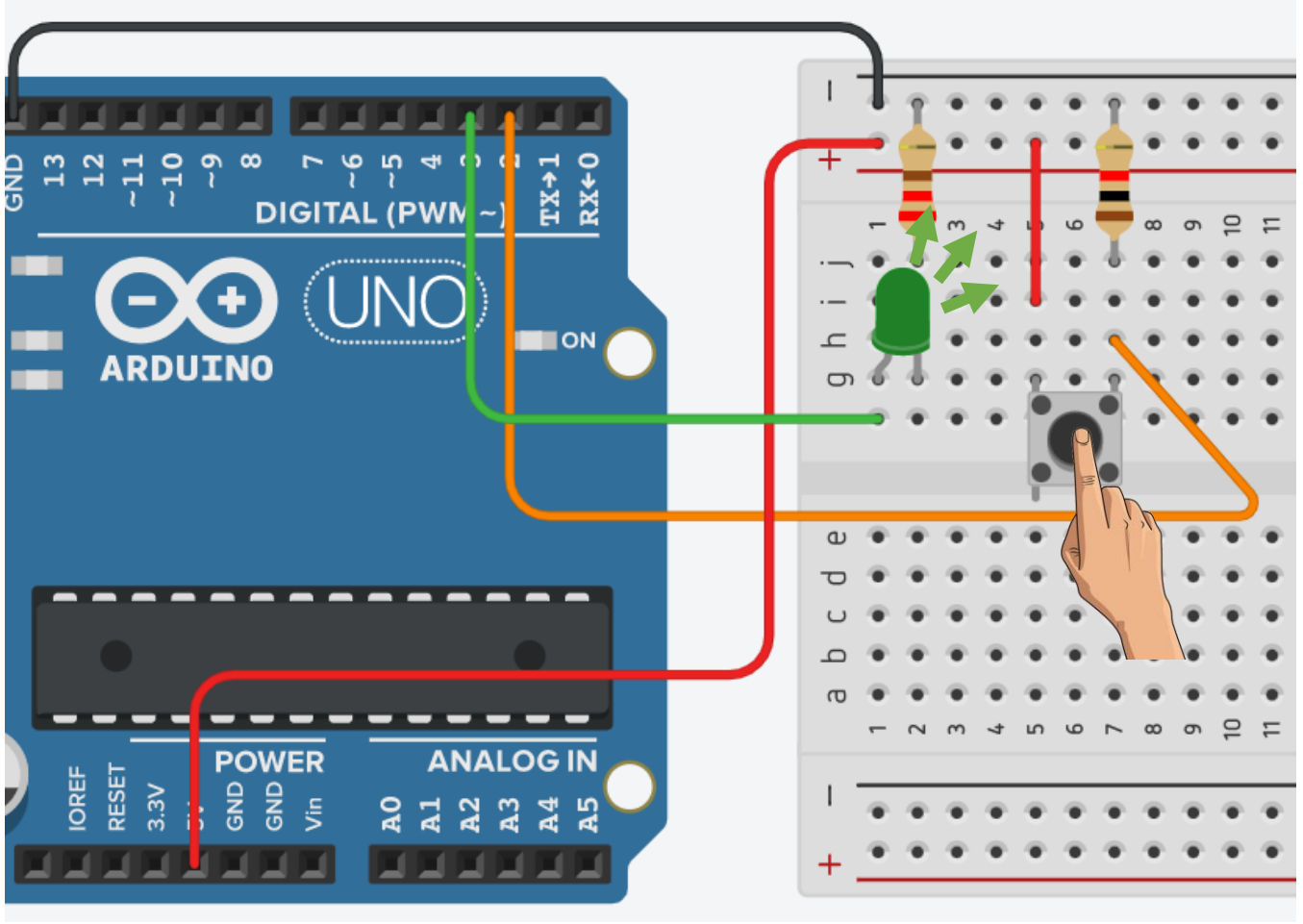


THANKS FOR COMING

LECTURE

2

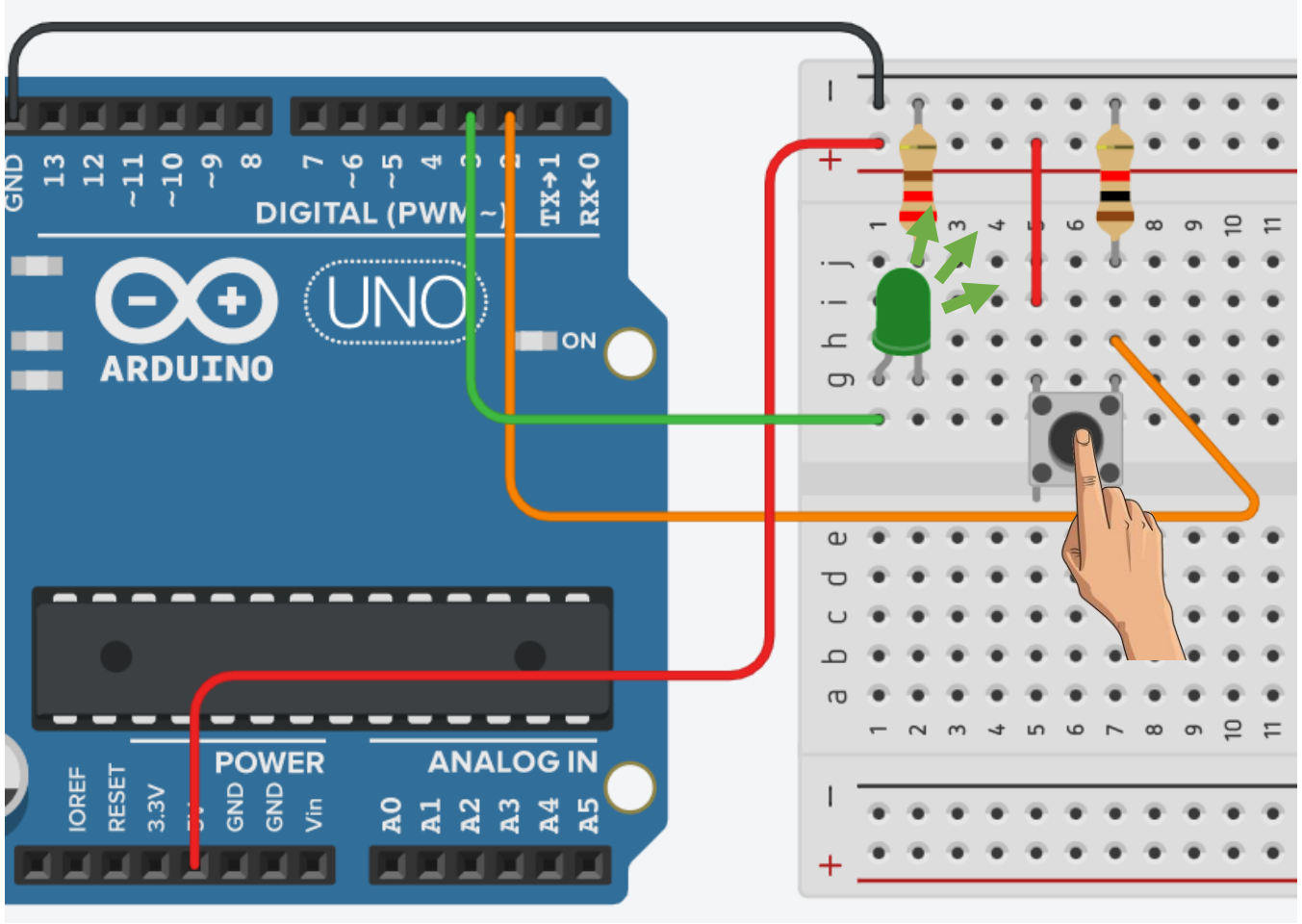
```ino
int led = 3;
int button = 2;
int Reading = 0;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(button, INPUT);
}

void loop()
{
    Reading = digitalRead(button);
    if (Reading == HIGH) { digitalWrite(led, HIGH); }
    else { digitalWrite(led, LOW); }
}
```



Reading

LOW HIGH

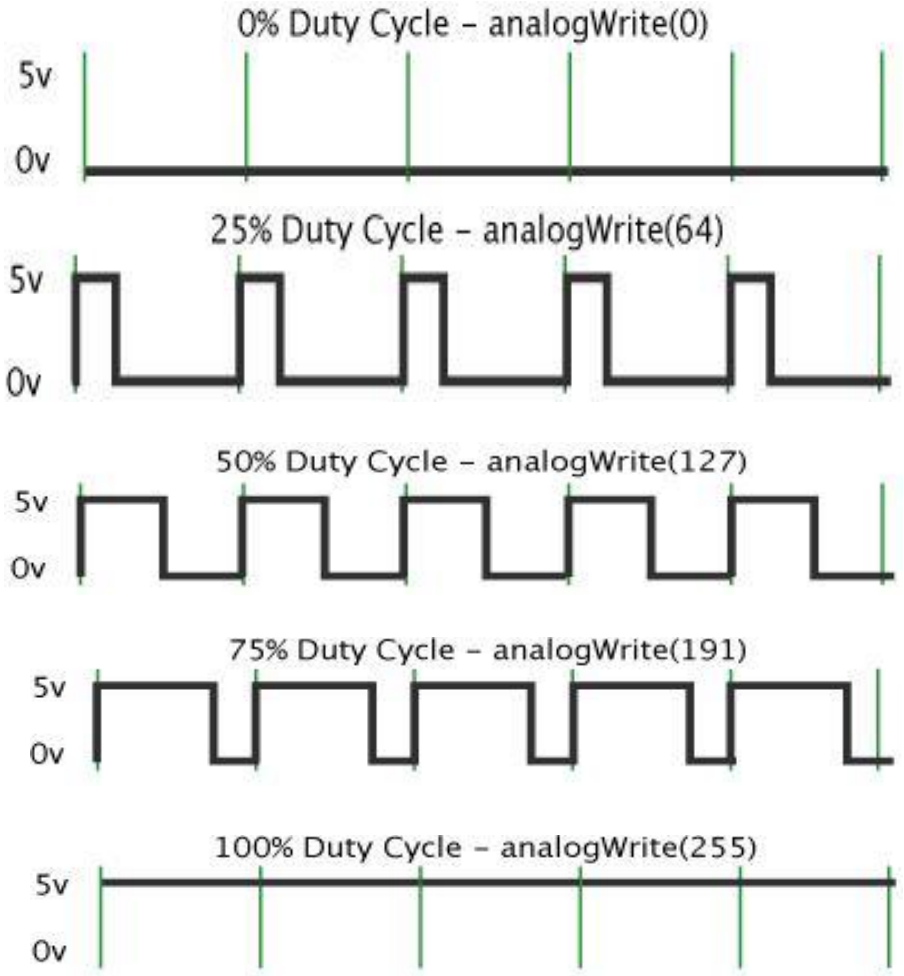
```
int led = 3;
int button = 2;
int Reading = 0;
int counter = 0;
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(button, INPUT);
}

void loop()
{
  Reading = digitalRead(button);
  if (Reading == HIGH)
  {
   counter++;
   if(counter==1)
   { digitalWrite(led, HIGH);}
   else
   { digitalWrite(led, LOW); counter=0; }
   delay(250);
  }}
```



counter

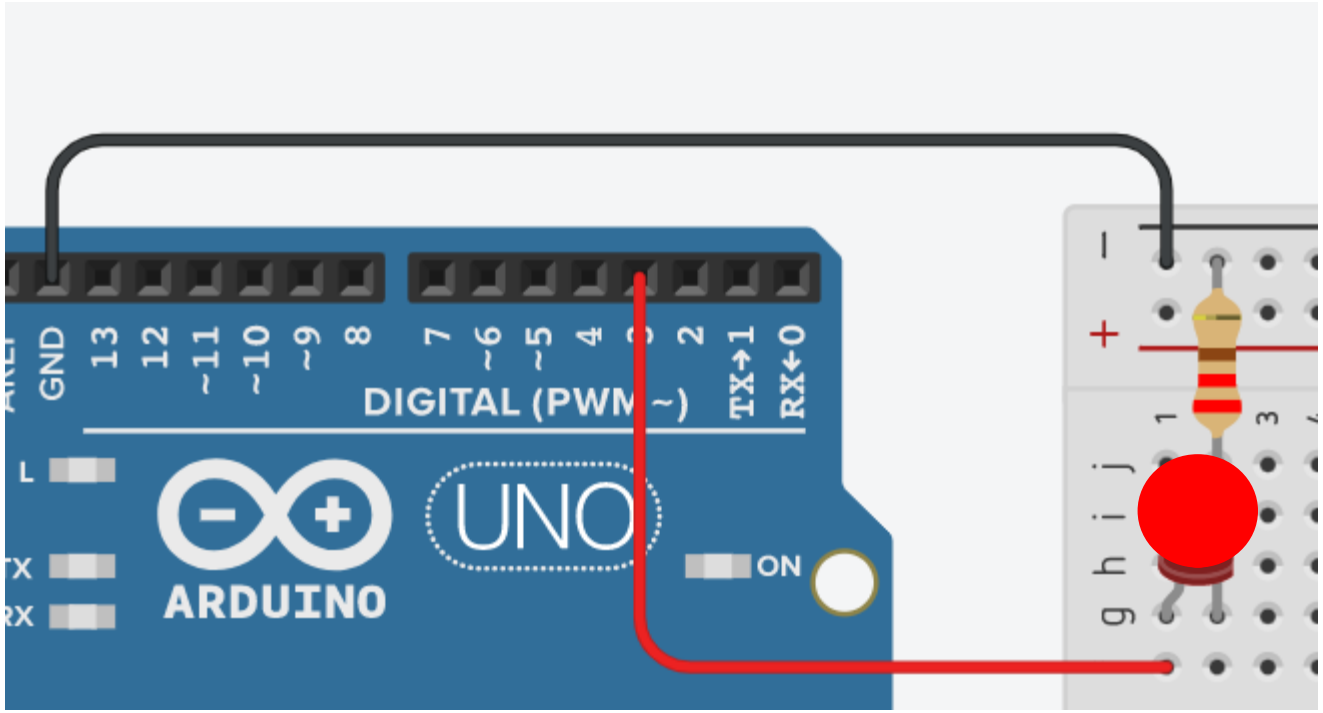
• Pulse Width Modulation PWM

- PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each.
- A call to `analogWrite()` is on a scale of 0 – 255 →MAX Value(255)
- `analogWrite(255)` requests a 100% duty cycle (always on)
- and `analogWrite(127)` is a 50% duty cycle (on half the time) for example.



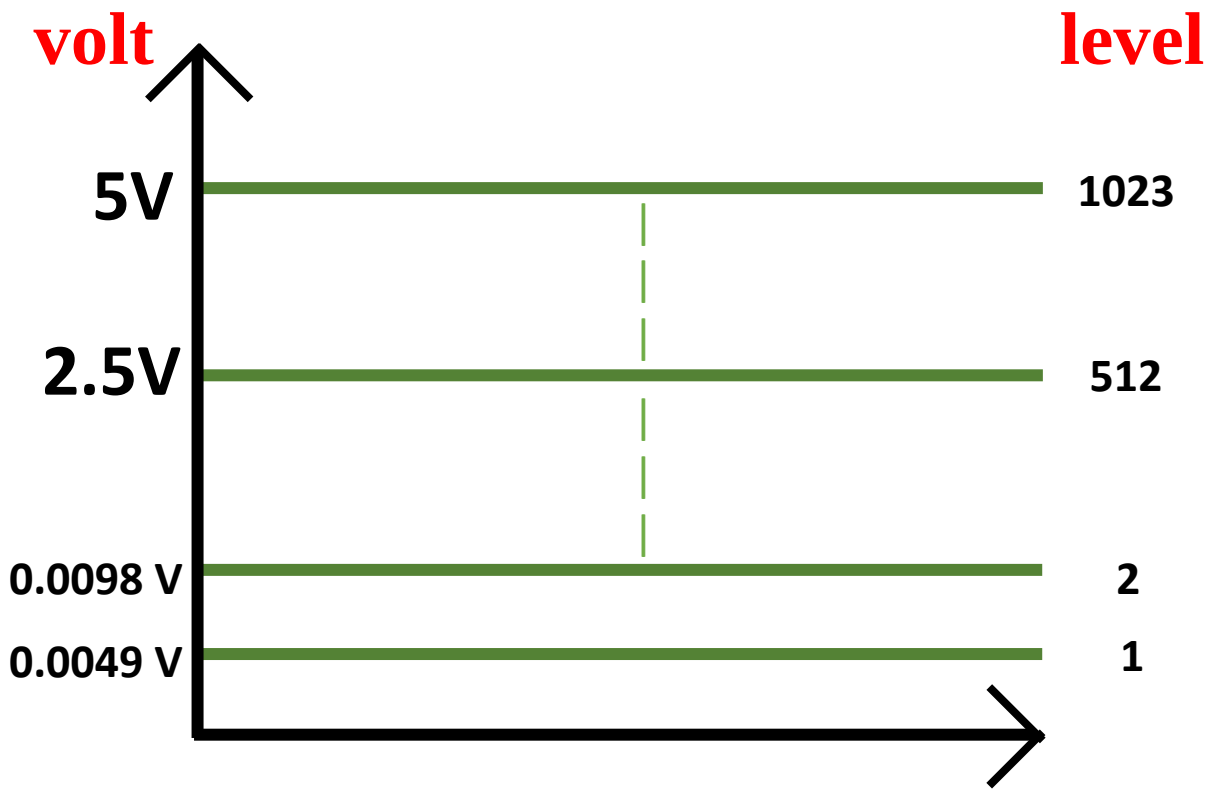
• Example

```
int led =  3;

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

void loop()
{
  analogWrite(led, 0);
  delay(1000);
  analogWrite(led, 65);
  delay(1000);
  analogWrite(led,128);
  delay(1000);
  analogWrite(led,255);
  delay(1000);
}
```



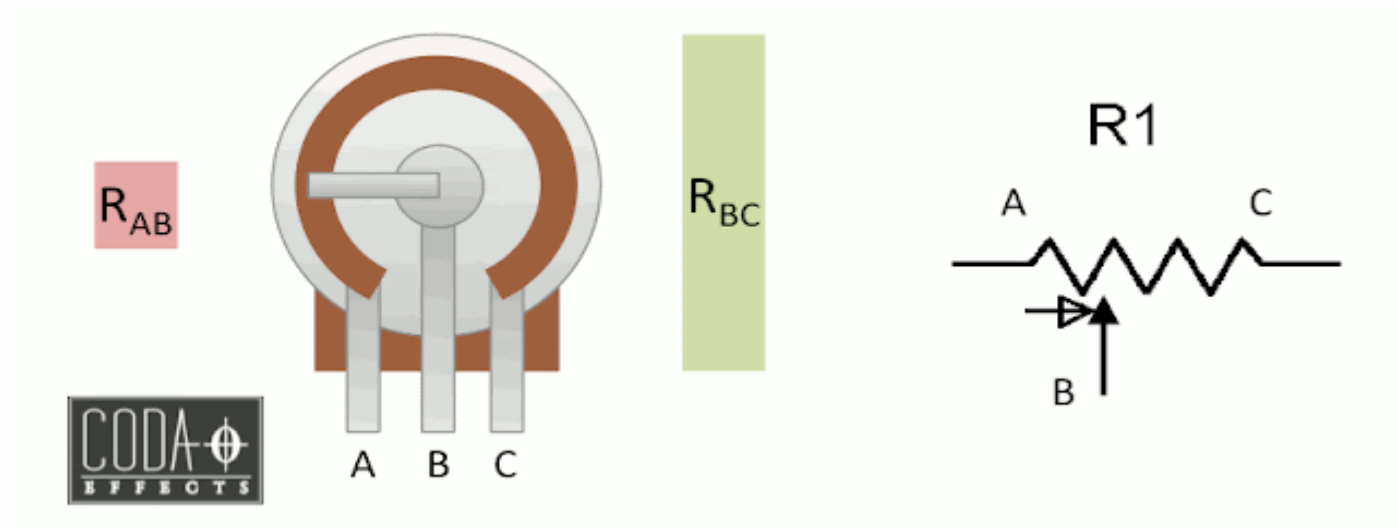
• ADC (analog to digital converter)

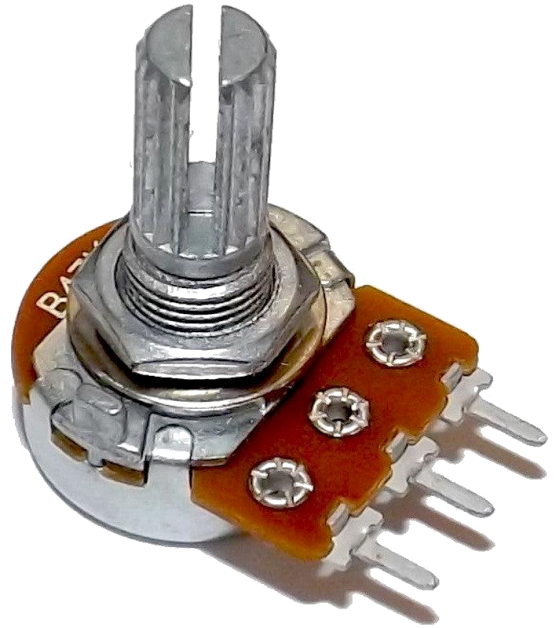
- 5V divided to 1024 level between 0 and 1023
- (4.9 mV) per unit



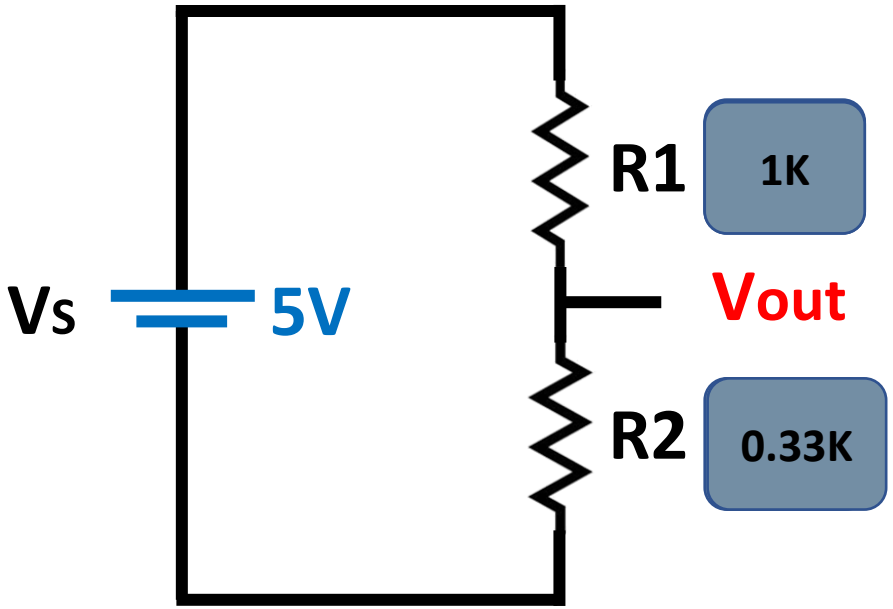
What is potentiometer

- Potentiometer or “POT” is a two variable resistance that can change its resistance value by knop





• Voltage Divider



$$V_{out} = VS \times \frac{R2}{R1+R2}$$

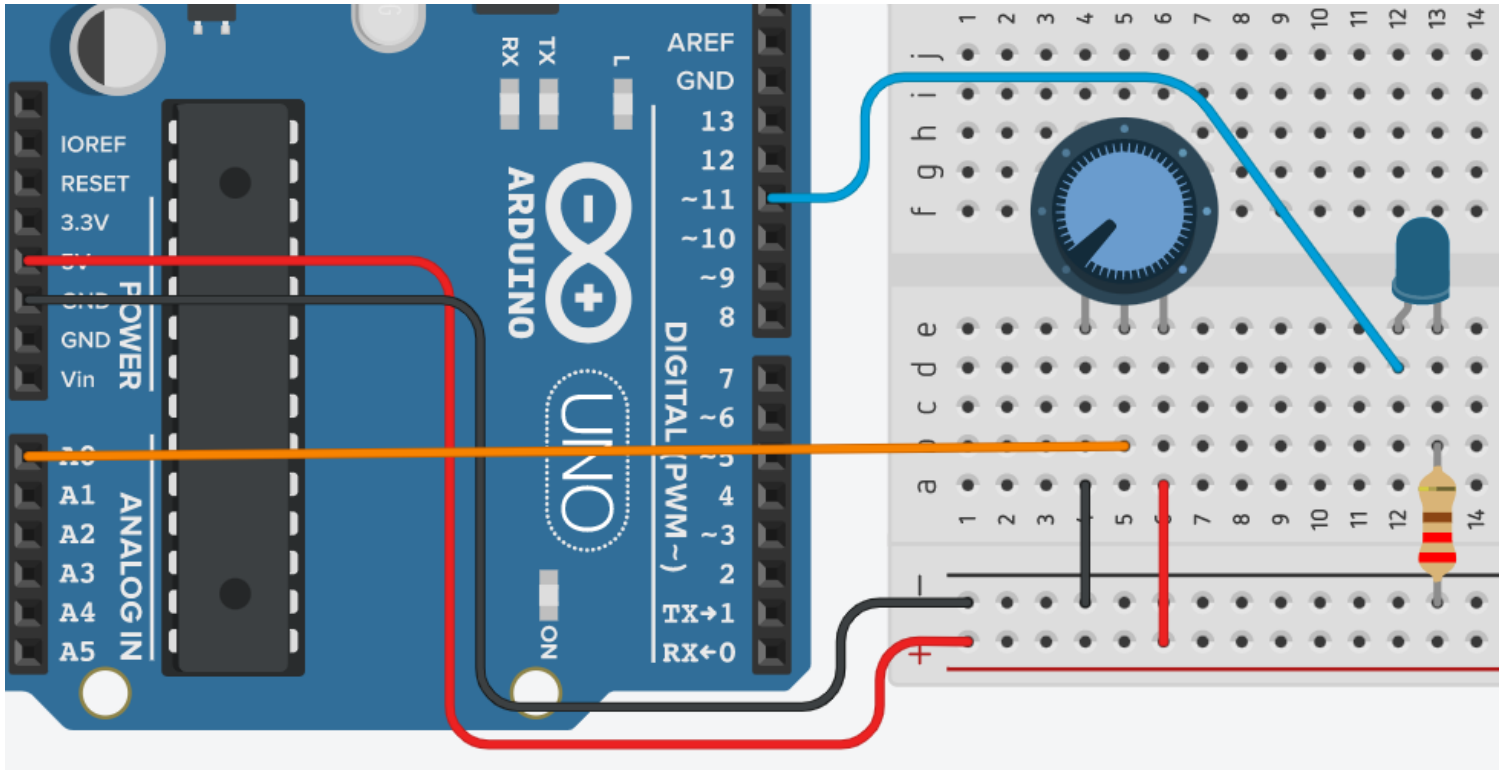
$$V_{out} = 5 \times \frac{1}{1+1} = 2.5\ V$$

$$V_{out} = 5 \times \frac{3}{1+3} = 3.75\ V$$

$$V_{out} = 5 \times \frac{0.33}{1+0.33} = 1.24\ V$$

• Flasher Control

```
int Pot = A0;
int Led = 11;
int sensorValue=0;
 void setup()
 {
  pinMode(Led, OUTPUT);
  pinMode(Pot , INPUT);
 }

 void loop()
 {
  sensorValue = analogRead(Pot);
  digitalWrite(Led, HIGH);
  delay(sensorValue);
  digitalWrite(Led, LOW);
  delay(sensorValue);
 }
```



• Brightness Control

```
int Pot = A0;
int Led = 11;
int POT_Reading = 0;

void setup()
{
  pinMode(Pot, INPUT);
  pinMode(Led, OUTPUT);
}

void loop()
{
   POT_Reading = analogRead(Pot);
   analogWrite(Led, POT_Reading);
}
```

```
analogWrite(Led,  0→255 );
```

POT_Reading = 0→255 ✓

POT_Reading = 256→1023 ✗

Over Flow

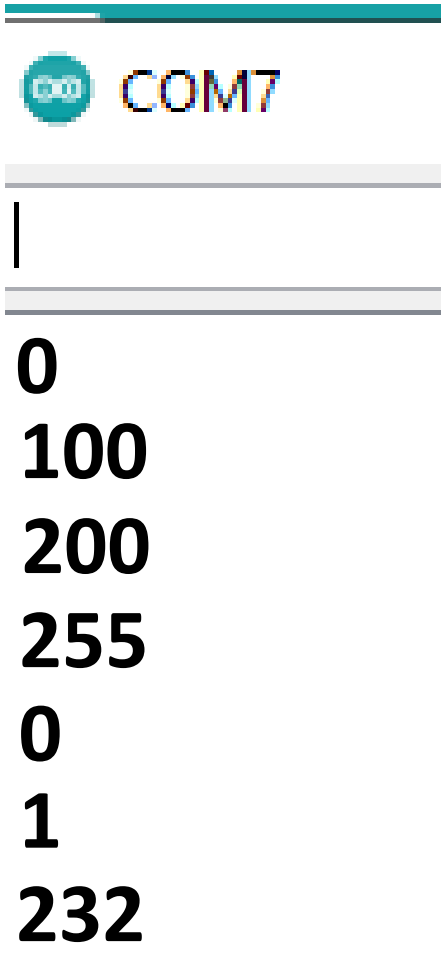
256 → 0

257 → 1

258 → 2

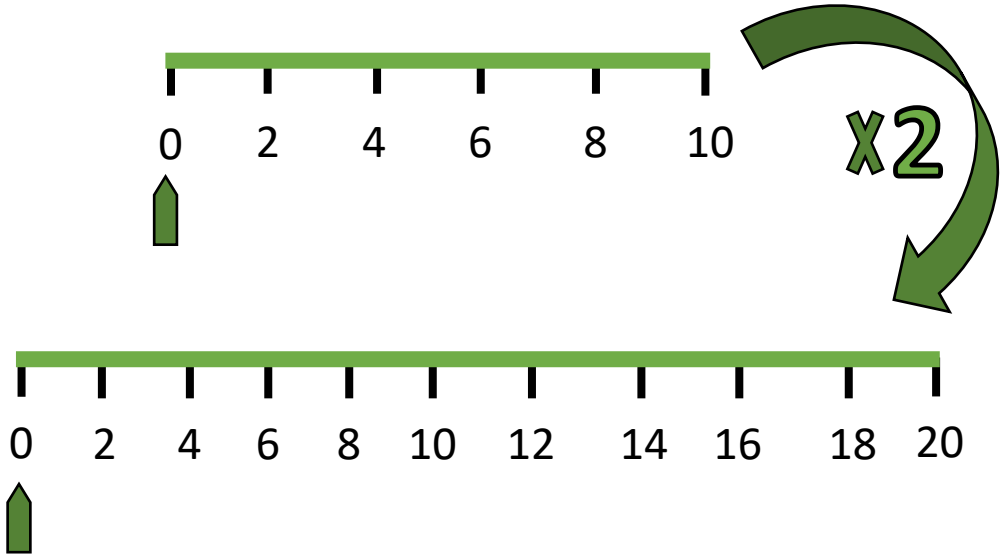
• Over Flow

```
byte x;
void setup()
 {
   Serial.begin(9600);
   x=0;    Serial.println(x);
   x=100;  Serial.println(x);
   x=200;  Serial.println(x);
   x=255;  Serial.println(x);
   x=256;  Serial.println(x);
   x=257;  Serial.println(x);
   x=1000; Serial.println(x);
 }

 void loop()
 {

 }
```



• Map Function

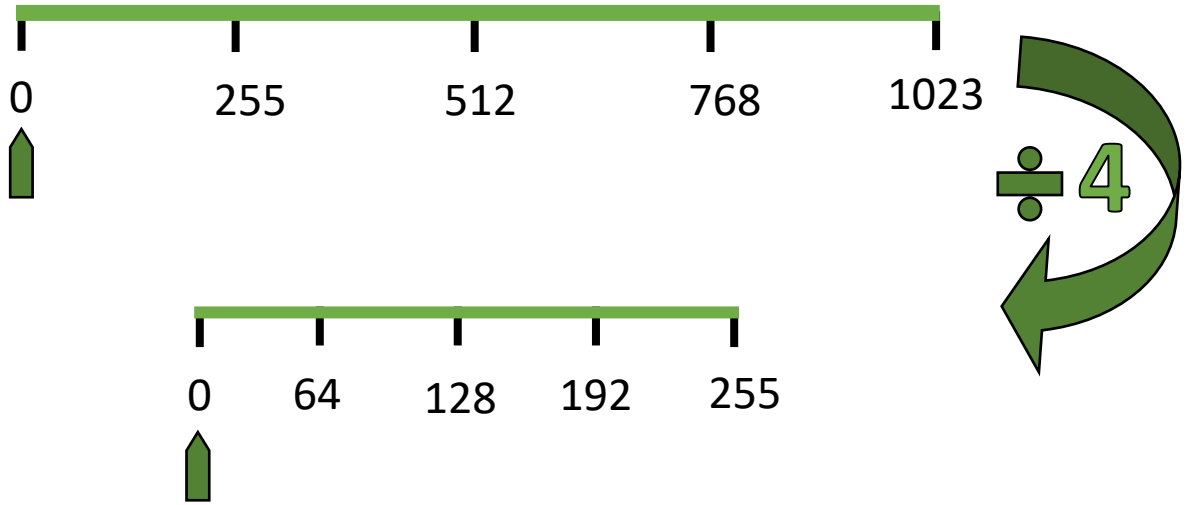
Name of range : **ST**



Map (value , fromLow, fromHigh , toLow , toHigh)

Map (ST , 0 , 10 , 0 , 20)

Name of range : **smart**

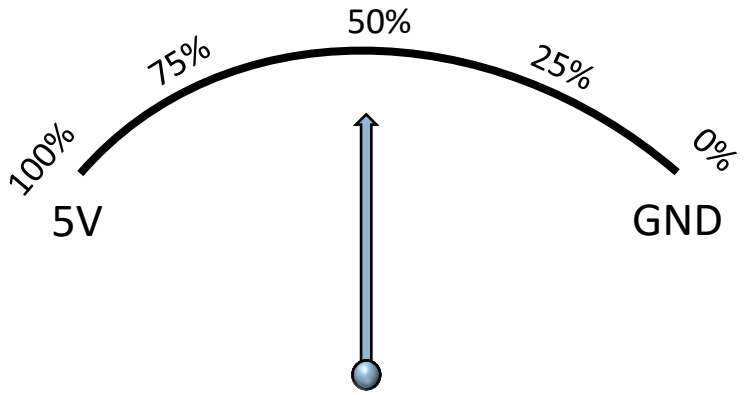


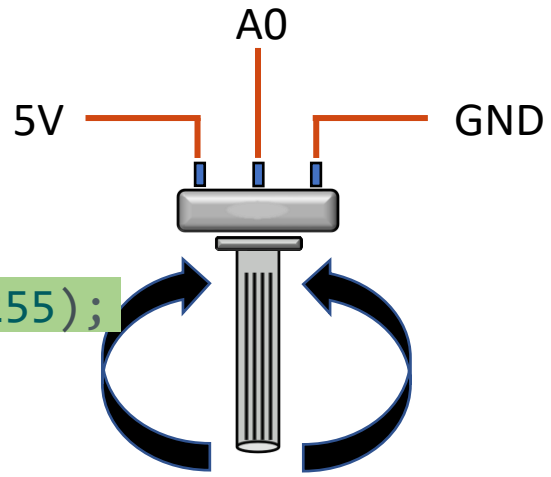
Map (value , fromLow, fromHigh , toLow , toHigh)

Map (Smart , 0 , 1023 , 0 , 255)

• Brightness Control

```
int Pot = A0;
int Led = 3;
int ledBrightness = 0;
int sensorValue   = 0;

void setup()
{
  pinMode(Pot, INPUT);
  pinMode(Led, OUTPUT);
}

void loop()
{
  sensorValue = analogRead(Pot);
  ledBrightness = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(Led, ledBrightness);
}
```



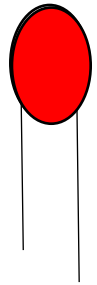


sensorValue

1023

ledBrightness

255



• Tone Function

- Generates a square wave of the specified frequency (and 50% duty cycle) on a pin
- the wave continues until a call to noTone()
- Only one tone can be generated at a time
- function will interfere with PWM output on pins 3 and 11
- It is not possible to generate tones lower than 31Hz
- the maximum frequency that can be produced is 65535 Hz
- the human hearing range is typically as high as 20 kHz

Syntax

tone(pin, frequency)

tone(pin, frequency, duration)

Parameters

pin: the Arduino pin on which to generate the tone.

frequency: the frequency of the tone in hertz. Allowed data types: unsigned int.

duration: the duration of the tone in milliseconds (optional). Allowed data types: unsigned long.

Notes and Warnings

If you want to play different pitches on multiple pins, you need to call noTone() on one pin before calling tone() on the next pin.

• Musical Notes

Constant Name	Frequency (Hz)
NOTE_B0	31
NOTE_C1	33
NOTE_CS1	35
NOTE_D1	37
NOTE_DS1	39
NOTE_E1	41
NOTE_F1	44
NOTE_FS1	46
NOTE_G1	49
NOTE_GS1	52
NOTE_A1	55
NOTE_AS1	58
NOTE_B1	62
NOTE_C2	65
NOTE_CS2	69
NOTE_D2	73
NOTE_DS2	78
NOTE_E2	82
NOTE_F2	87
NOTE_FS2	93
NOTE_G2	98
NOTE_GS2	104

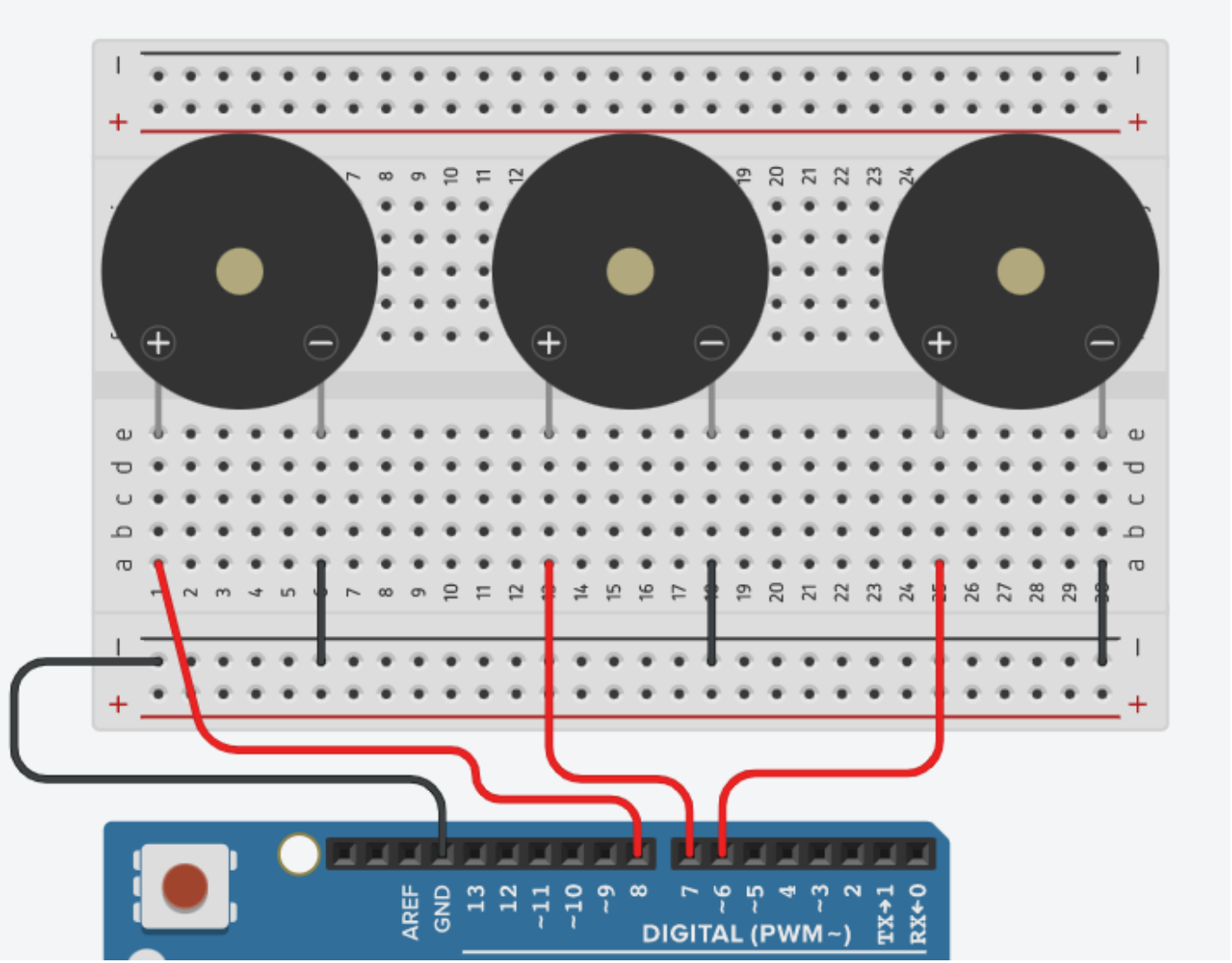
Constant Name	Frequency (Hz)
NOTE_A2	110
NOTE_AS2	117
NOTE_B2	123
NOTE_C3	131
NOTE_CS3	139
NOTE_D3	147
NOTE_DS3	156
NOTE_E3	165
NOTE_F3	175
NOTE_FS3	185
NOTE_G3	196
NOTE_GS3	208
NOTE_A3	220
NOTE_AS3	233
NOTE_B3	247
NOTE_C4	262
NOTE_CS4	277
NOTE_D4	294
NOTE_DS4	311
NOTE_E4	330
NOTE_F4	349
NOTE_FS4	370

Constant Name	Frequency (Hz)
NOTE_G4	392
NOTE_GS4	415
NOTE_A4	440
NOTE_AS4	466
NOTE_B4	494
NOTE_C5	523
NOTE_CS5	554
NOTE_D5	587
NOTE_DS5	622
NOTE_E5	659
NOTE_F5	698
NOTE_FS5	740
NOTE_G5	784
NOTE_GS5	831
NOTE_A5	880
NOTE_AS5	932
NOTE_B5	988
NOTE_C6	1047
NOTE_CS6	1109
NOTE_D6	1175
NOTE_DS6	1245
NOTE_E6	1319

Constant Name	Frequency (Hz)
NOTE_F6	1397
NOTE_FS6	1480
NOTE_G6	1568
NOTE_GS6	1661
NOTE_A6	1760
NOTE_AS6	1865
NOTE_B6	1976
NOTE_C7	2093
NOTE_CS7	2217
NOTE_D7	2349
NOTE_DS7	2489
NOTE_E7	2637
NOTE_F7	2794
NOTE_FS7	2960
NOTE_G7	3136
NOTE_GS7	3322
NOTE_A7	3520
NOTE_AS7	3729
NOTE_B7	3951
NOTE_C8	4186
NOTE_CS8	4435
NOTE_D8	4699
NOTE_DS8	4978

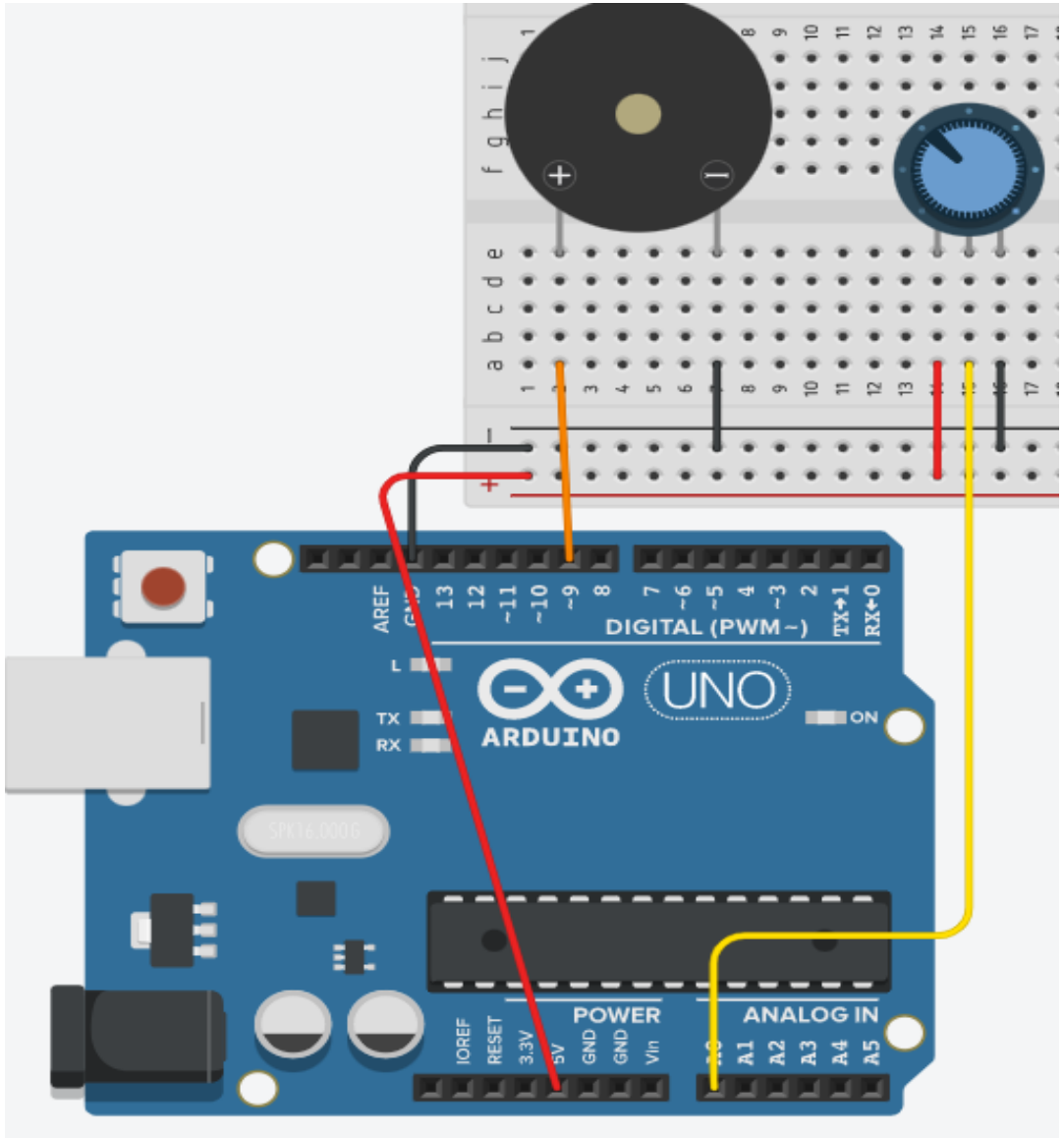
• Tone Example

```
void setup() { }

void loop(){
tone(6,440,200);
delay(200);
noTone(6);

tone(7,494,500);
delay(500);
noTone(7);

tone(8,523,300);
delay(300);
noTone(8);
}
```



• Task

- Control buzzer tones with POT



THANKS FOR COMING

LECTURE 3

• LDR (Light Dependent Resistor)

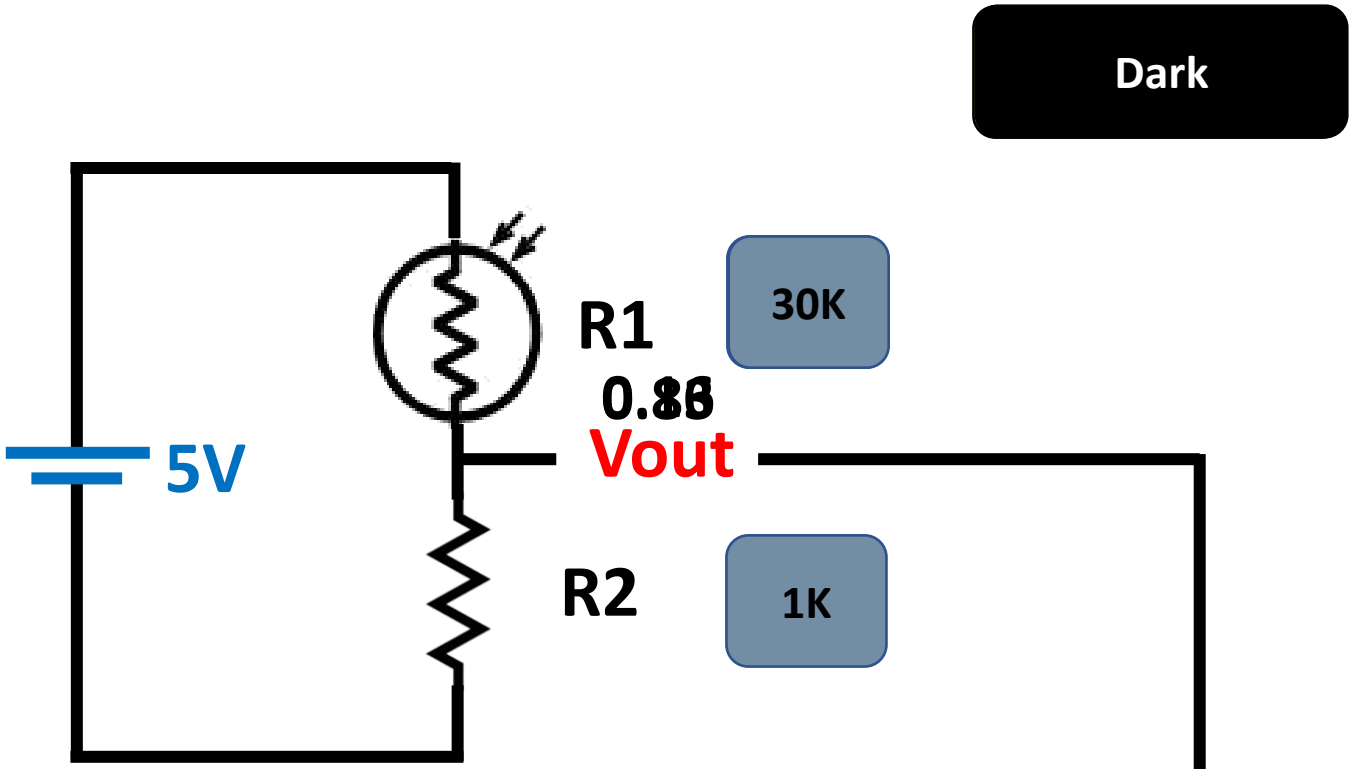
- An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

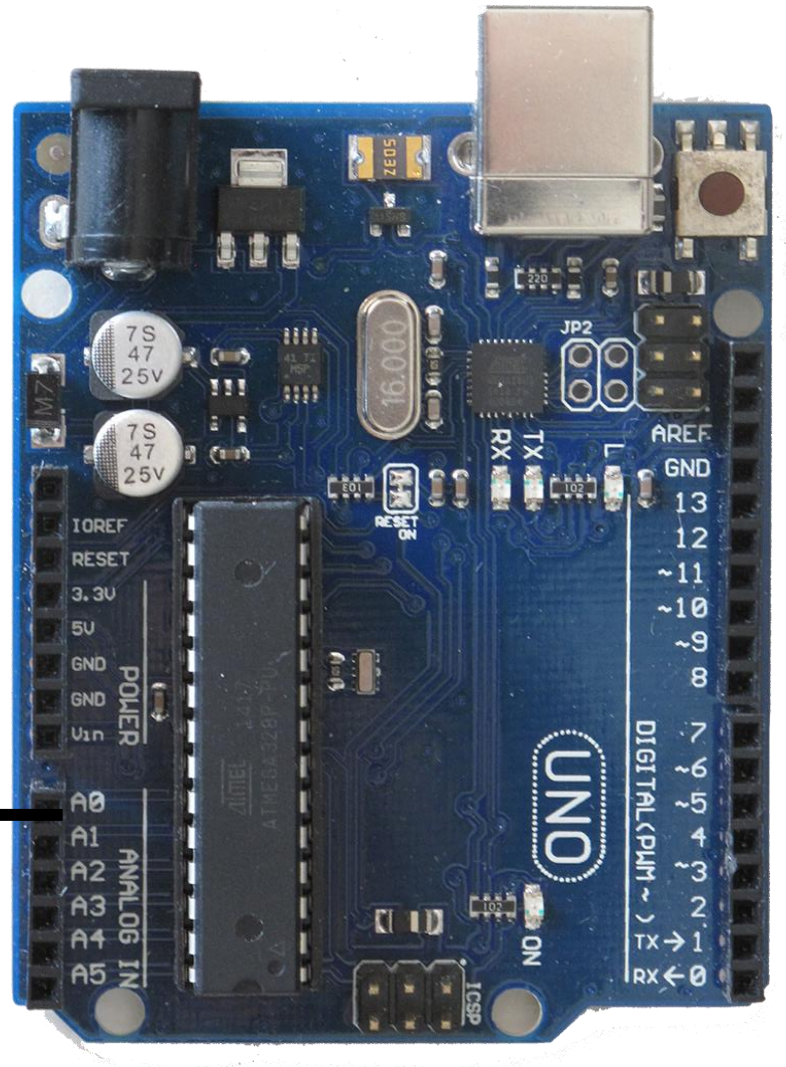
- The most common type of LDR has a resistance that falls with an increase in the light intensity falling upon the device

- The resistance of an LDR may typically have the following resistances:

- Daylight = 5000Ω
- Dark = 20MΩ

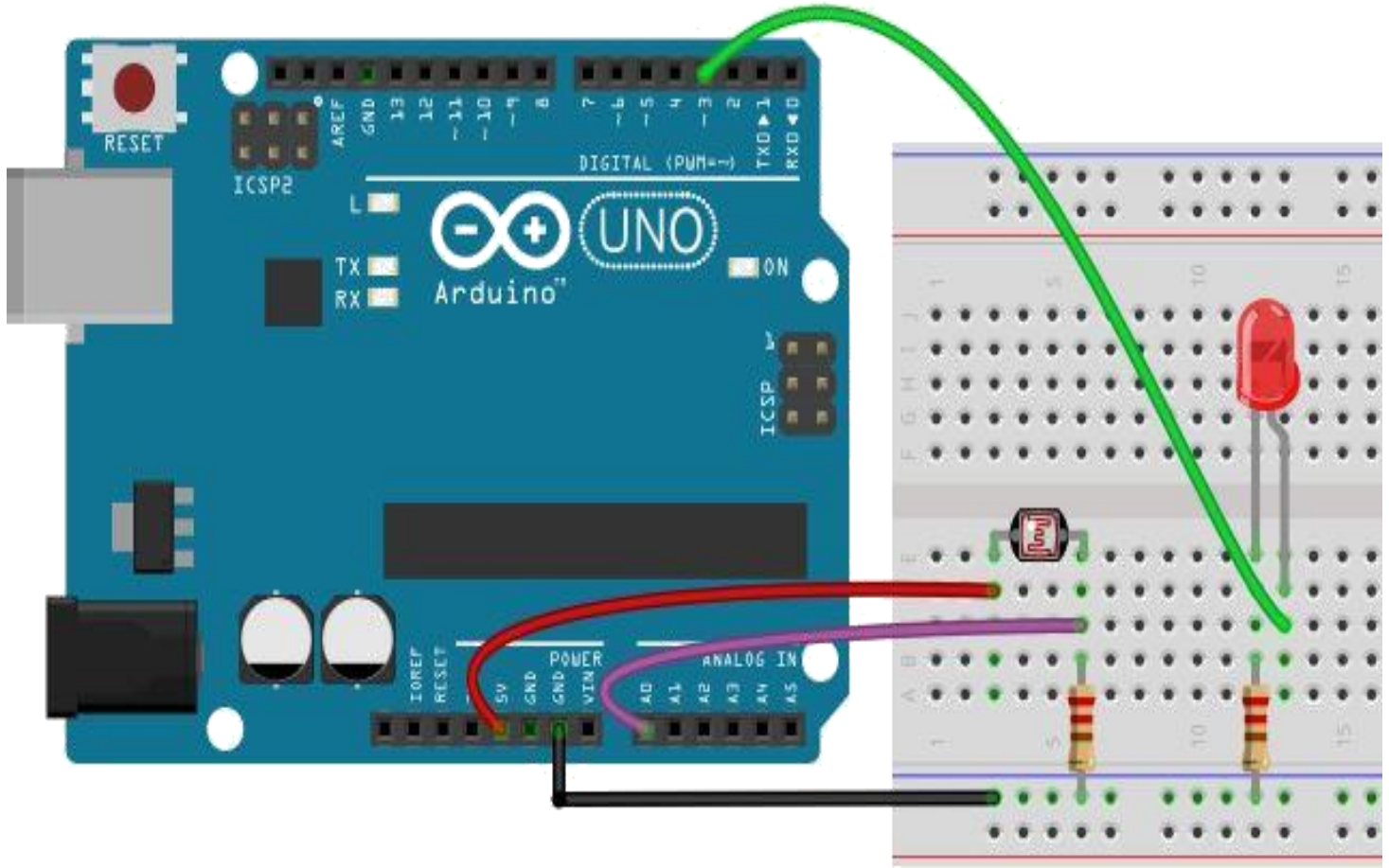
• Basic principle





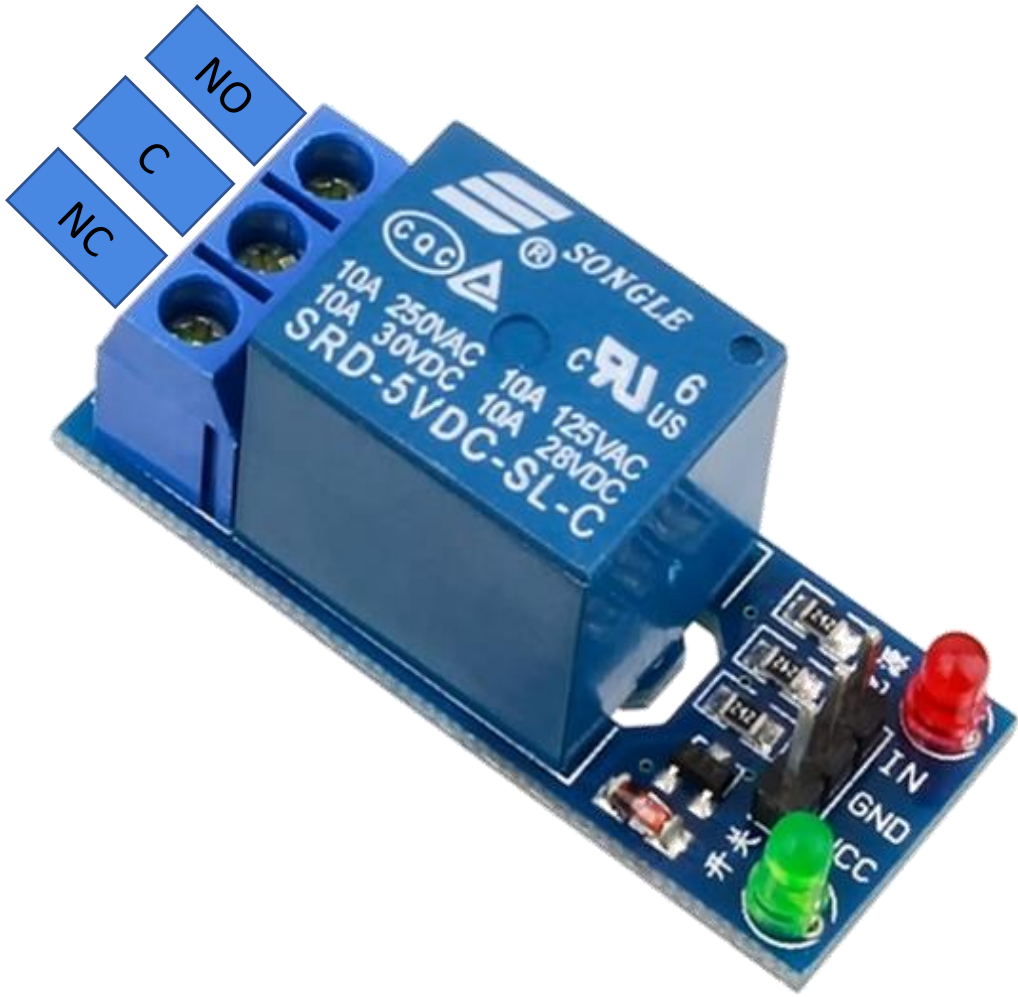
• Code

```
#define ldr A0
#define led 3
int threshold = 40;
int level;
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}
void loop() {
  level = analogRead(ldr);
    Serial.println(level);
  if (level < threshold) {
      digitalWrite(led, 1); }
   else {
    digitalWrite(led, 0);}
}
```



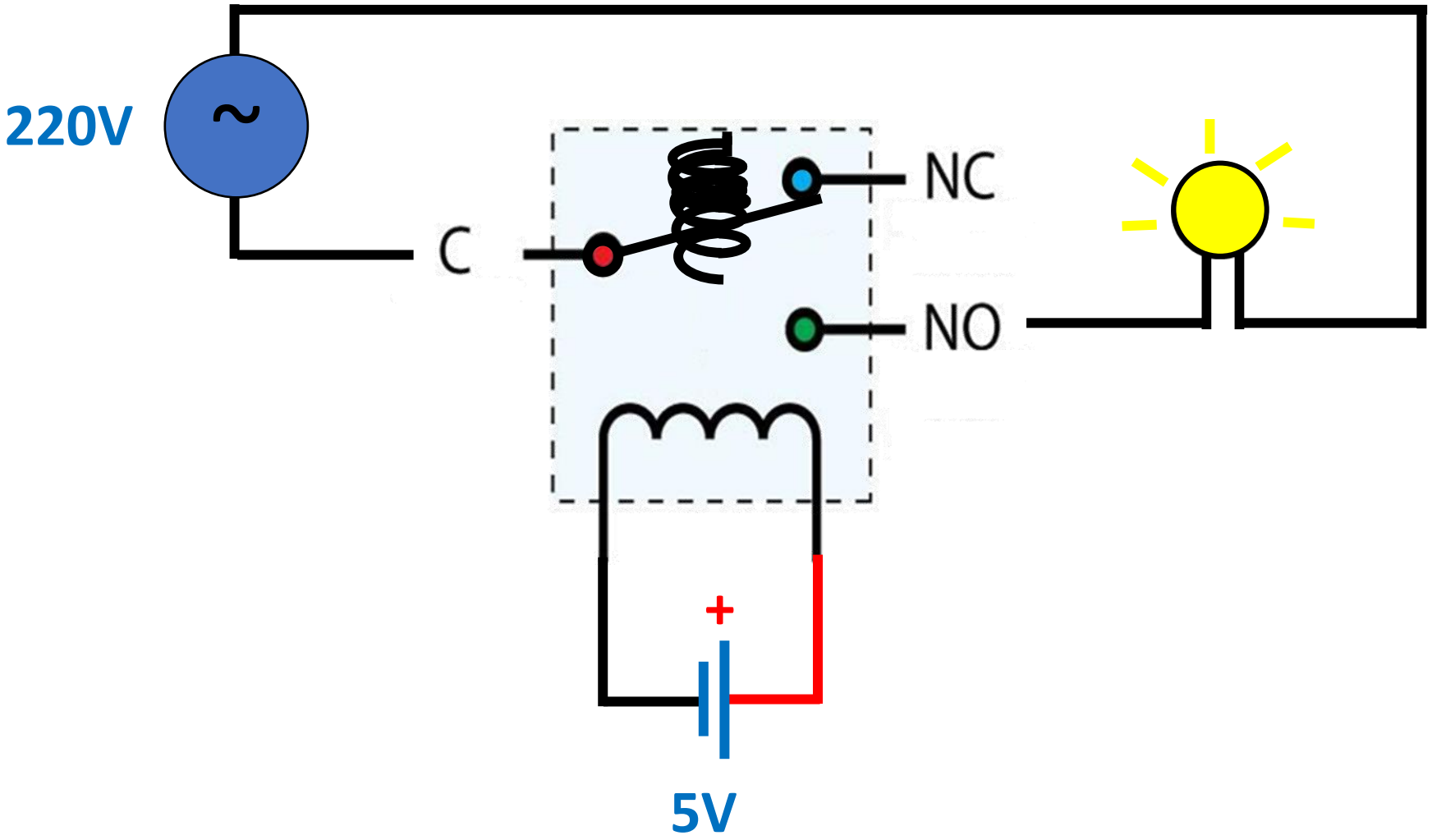
• Relay

An electrical relay is an electromagnetically operated electrical switch - an electromechanical switch. A relatively small current is used to create a magnetic field in a coil within a magnetic core and this is used to operate a switch that can control a much larger current.



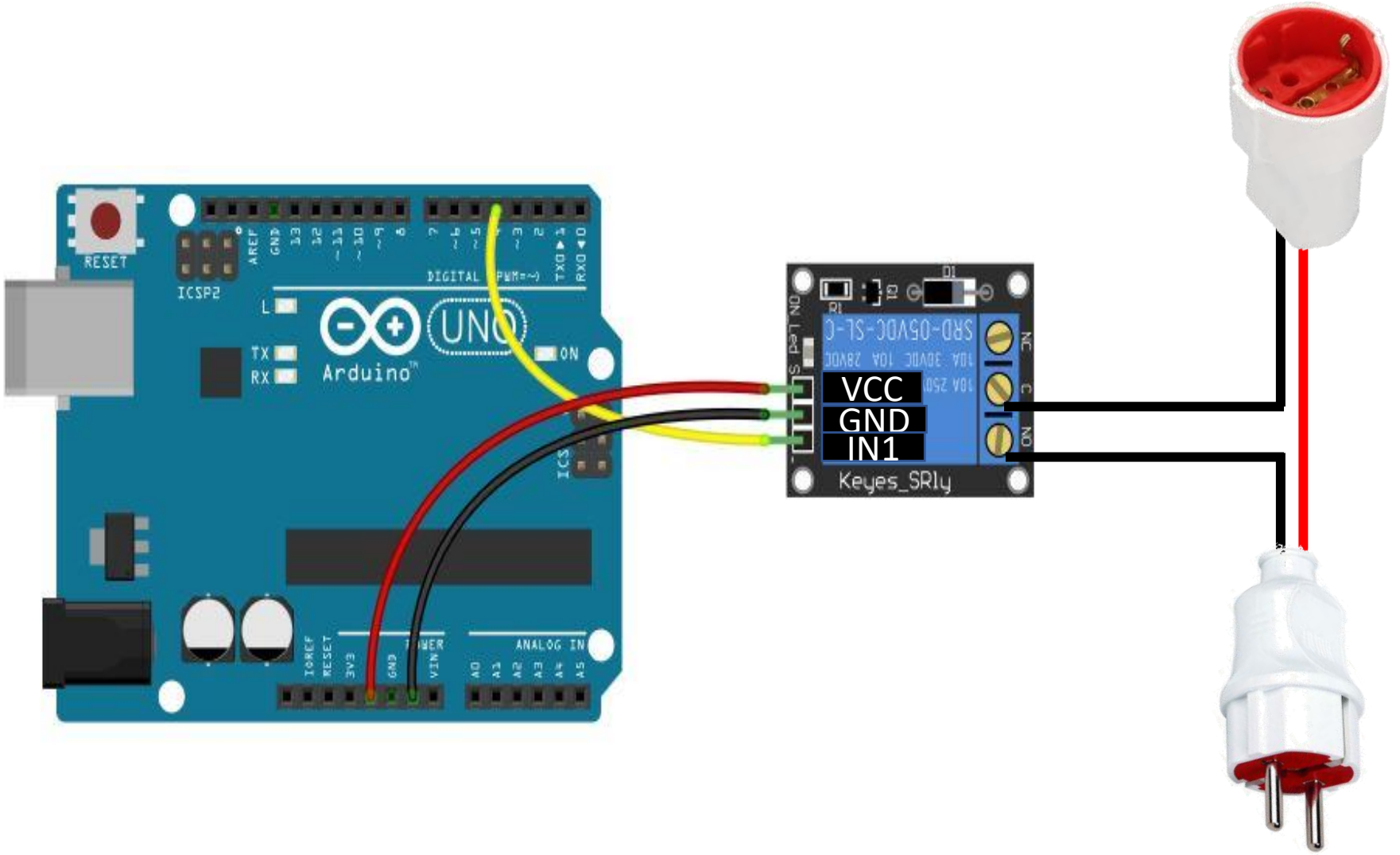
• Basic principle

• Basic principle

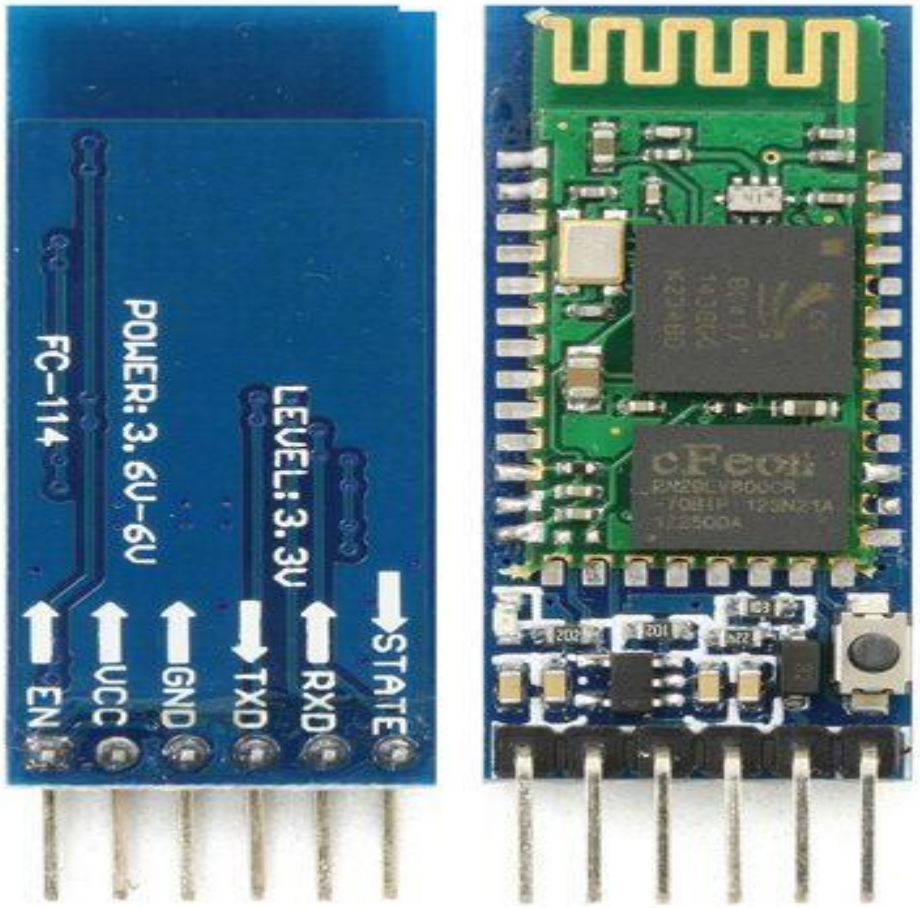


• Code of Relay

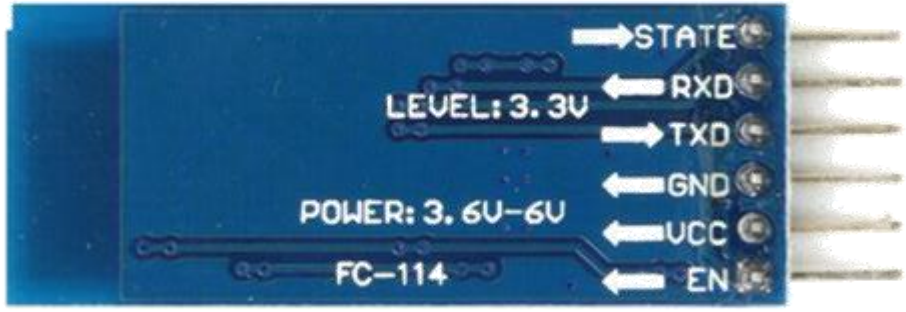
```
#define on 0
#define off 1
int relay=4 ;
void setup()
{
pinMode(relay,OUTPUT);
 }
 void loop()
{
digitalWrite(relay,on);
delay(5000);
digitalWrite(relay,off);
delay(5000);
}
```



• Bluetooth (HC-05)



• Pin Configuration



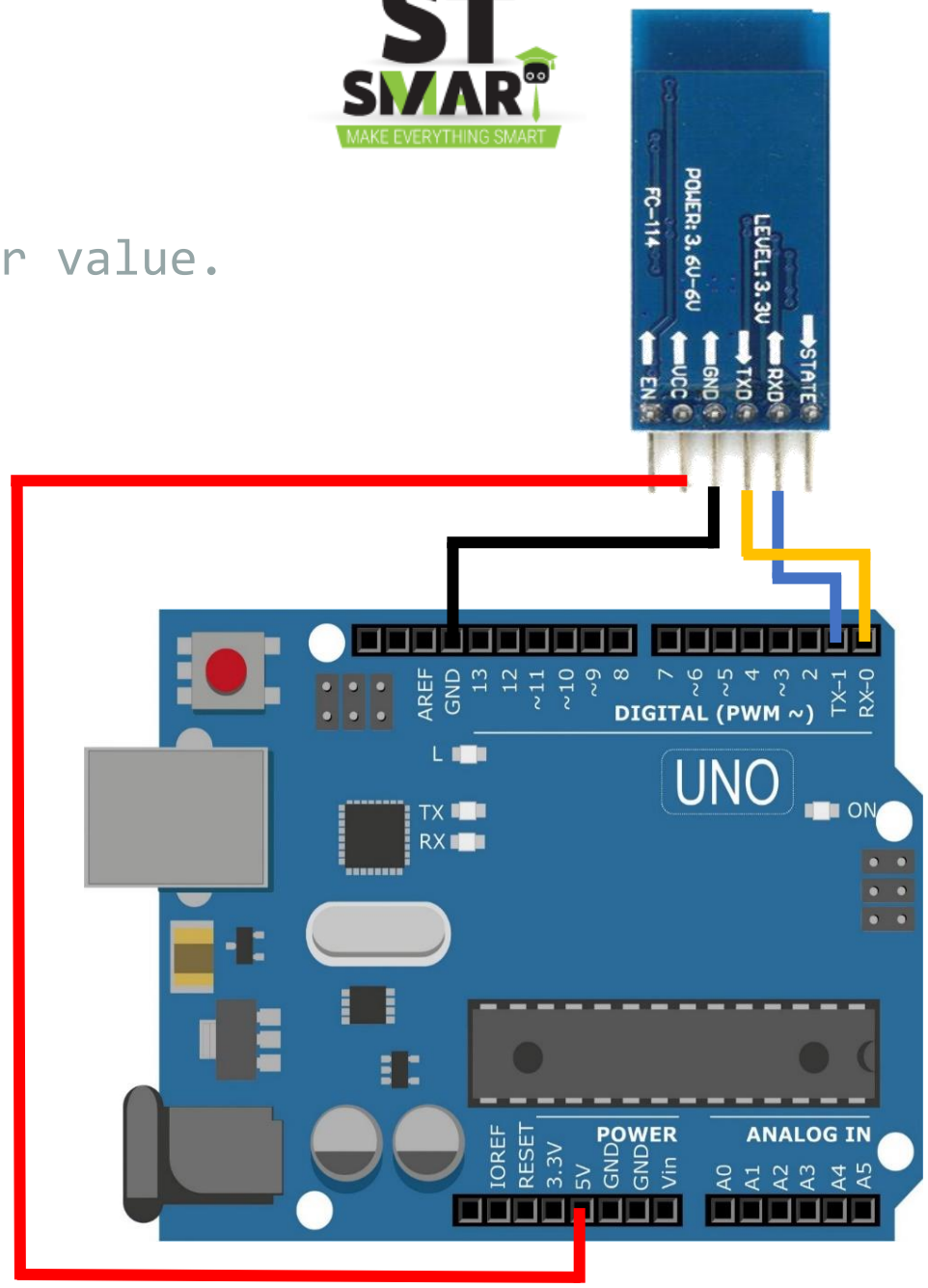
Pin number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

HC-05 Default Settings

- Default Bluetooth Name: “HC-05”
- Default Password: 1234 or 0000
- Default Communication: Slave
- Default Mode: Data Mode
- Data Mode Baud Rate: 9600
- Command Mode Baud Rate: 38400
- This Bluetooth module covers 9 meters (30ft)
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA

• Example

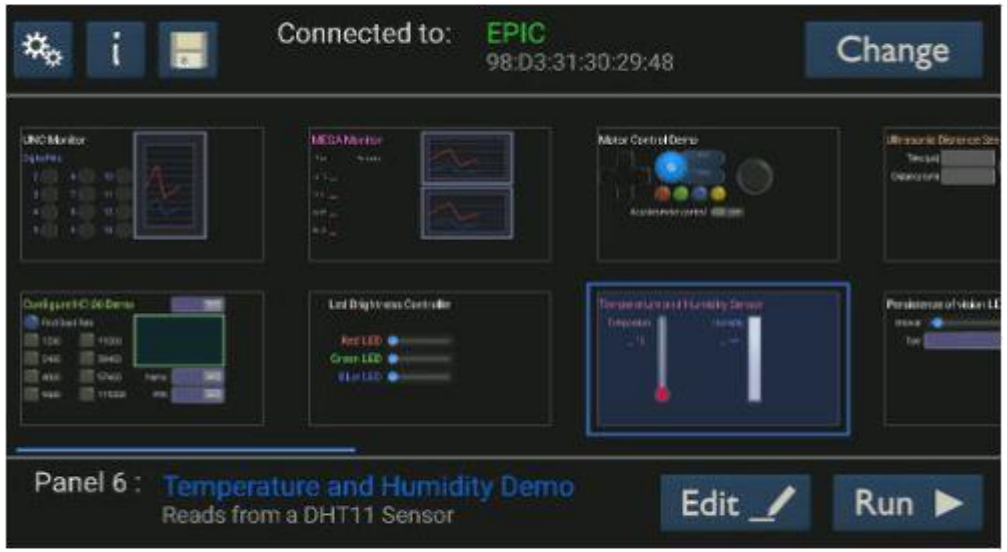
```
const int red = 13;
char reading;  // data type used to store a character value.
void setup( )
{
 pinMode(red, OUTPUT);
 Serial.begin(9600);
}
void loop( ) {
  if(Serial.available()>0) {
     reading=Serial.read();
     switch(reading){
        case 'F': digitalWrite(red,1);
                          break;
        case 'S': digitalWrite(red,0);
                          break;
       }
   }
}
```

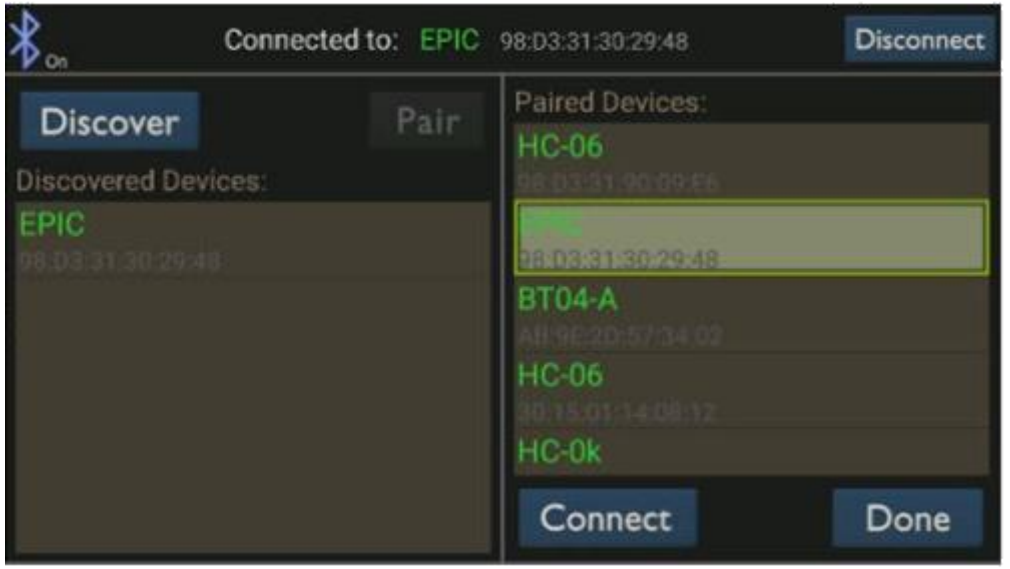


• Android App.

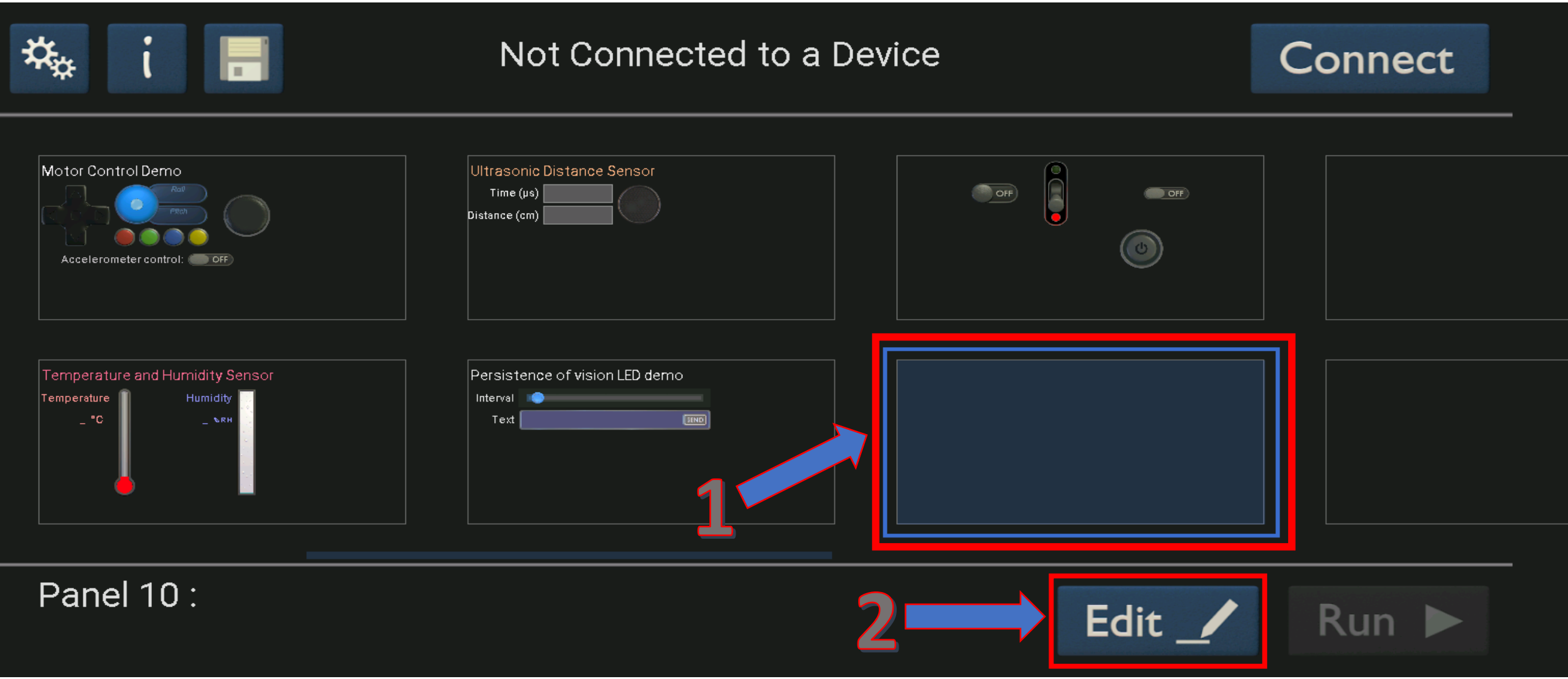

Bluetooth Electronics
Control your electronic project with your Android device!
Keuwlsoft


GET IT ON
Google Play

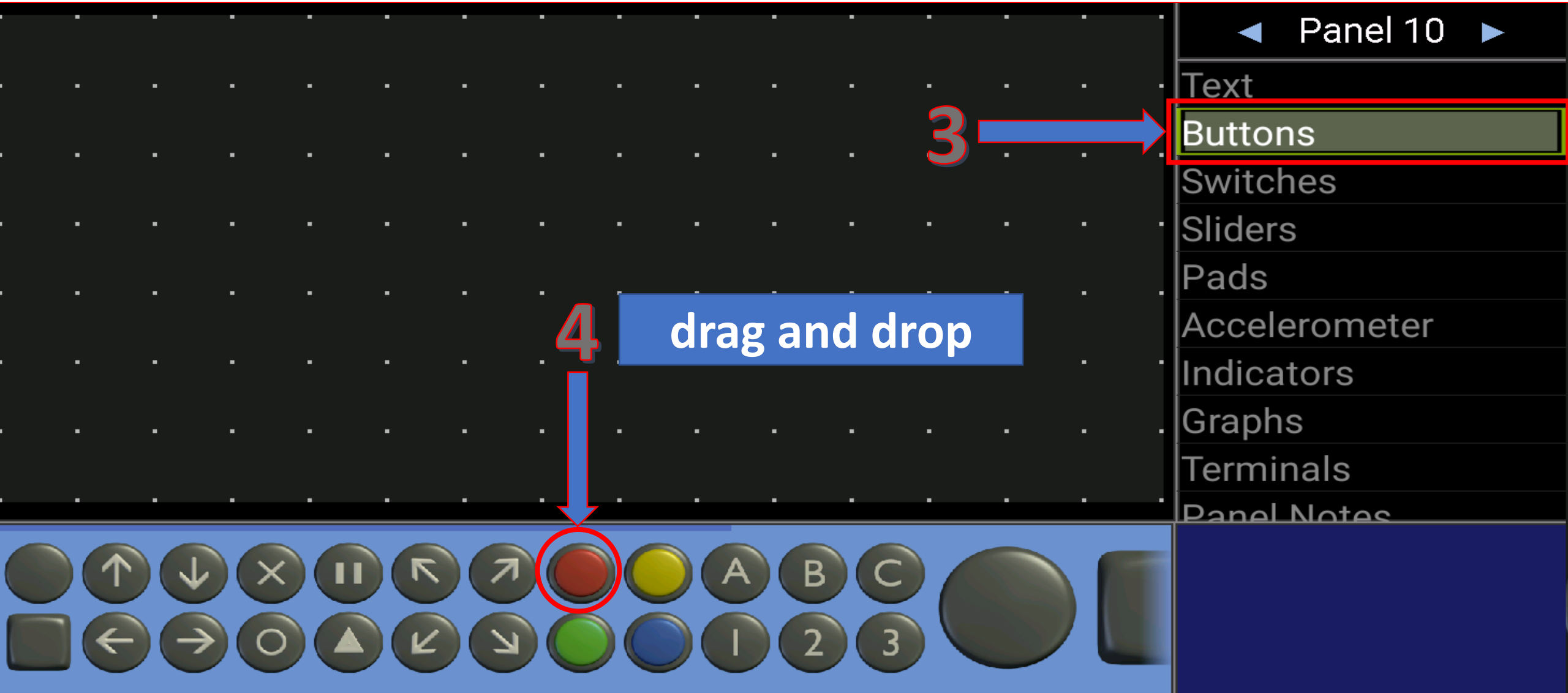

Connected to: EPIC
98:D3:31:30:29:48
Change
Panel 6 : Temperature and Humidity Demo
Reads from a DHT11 Sensor
Edit
Run


On
Connected to: EPIC 98:D3:31:30:29:48
Disconnect
Discover
Pair
Discovered Devices:
EPIC
Paired Devices:
HC-06
BT04-A
HC-06
HC-0k
Connect
Done

• Setting Up

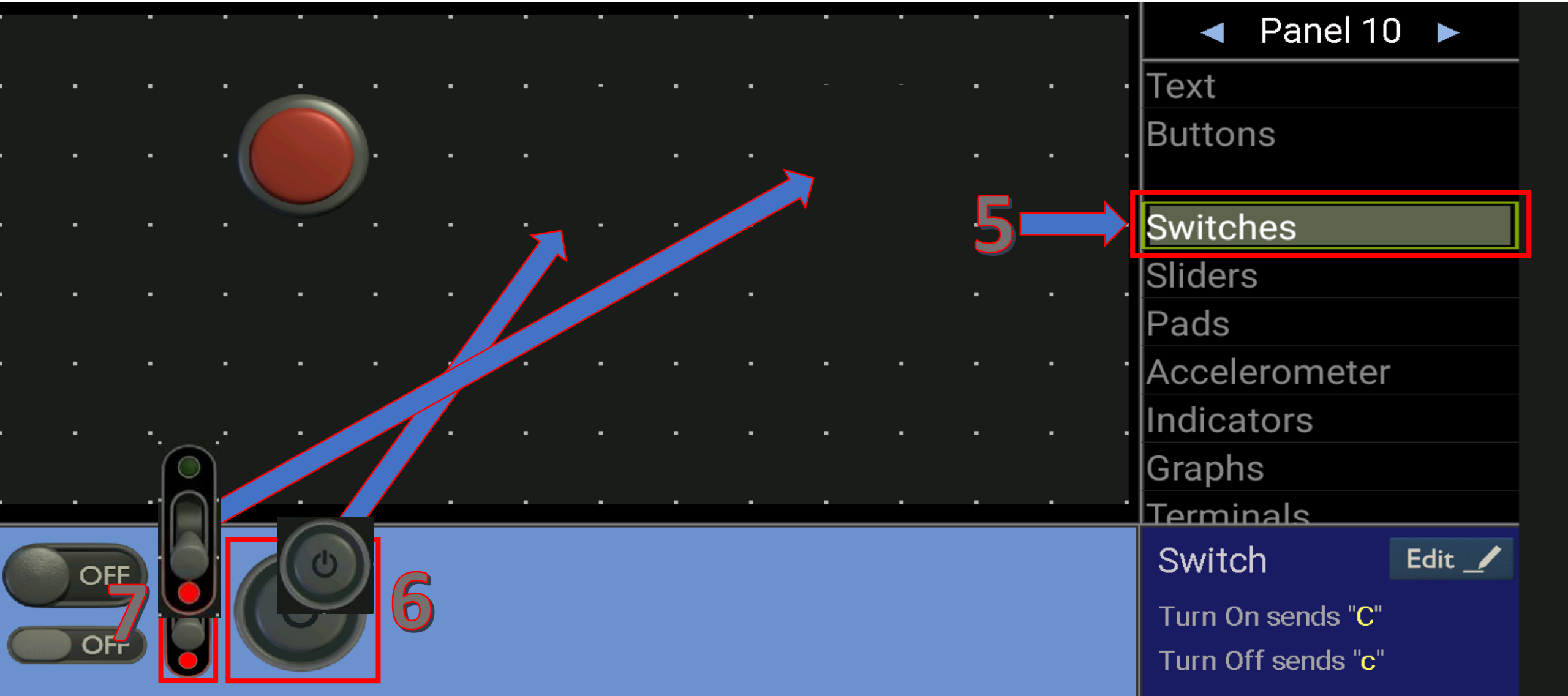

Not Connected to a Device
Connect
Motor Control Demo
Roll
Pitch
Accelerometer control:
OFF
Ultrasonic Distance Sensor
Time (µs)
Distance (cm)
OFF
OFF
Temperature and Humidity Sensor
Temperature
_ °C
Humidity
_ %RH
Persistence of vision LED demo
Interval
Text
SEND
1
Panel 10 :
2
Edit
Run

• Setting Up

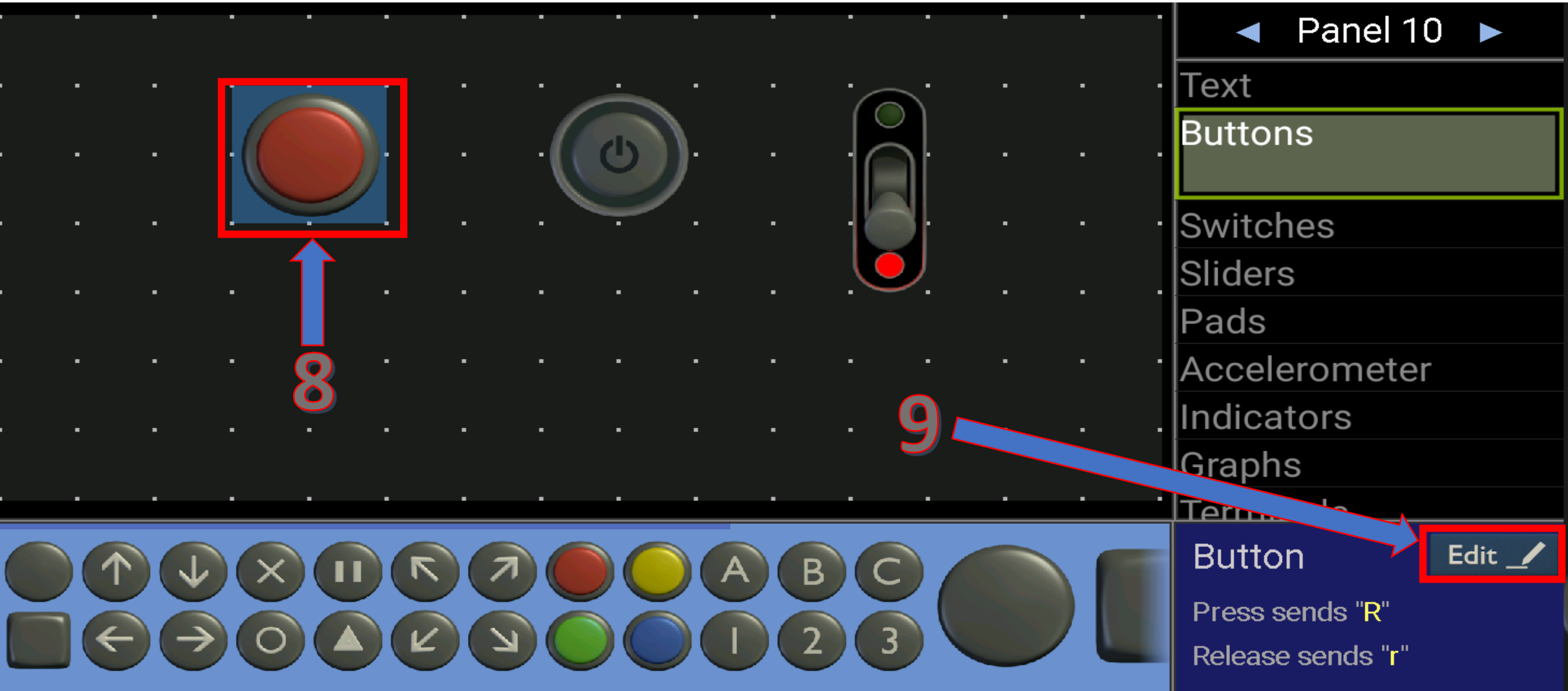

Panel 10
Text
Buttons
Switches
Sliders
Pads
Accelerometer
Indicators
Graphs
Terminals
3
4
drag and drop
A
B
C
1
2
3

- Setting Up

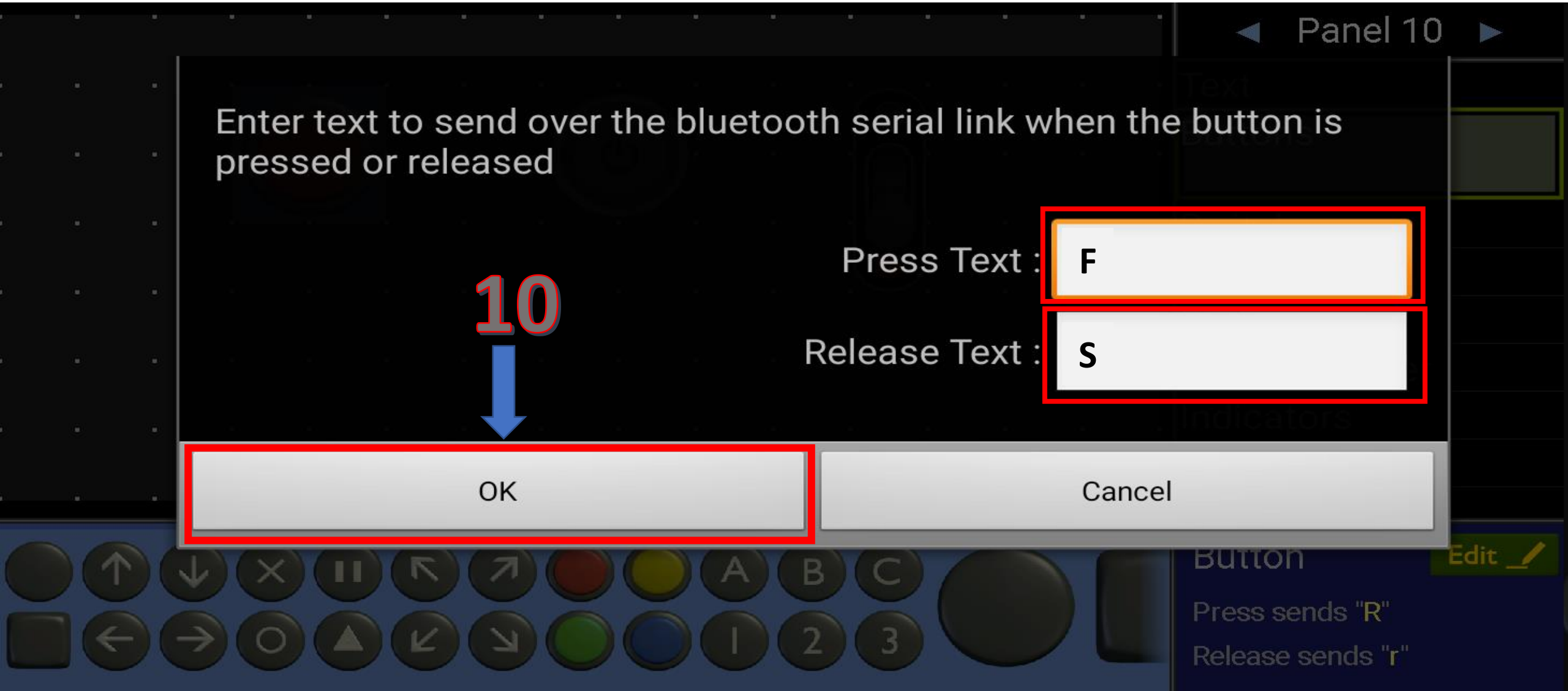
Bluetooth Electronic



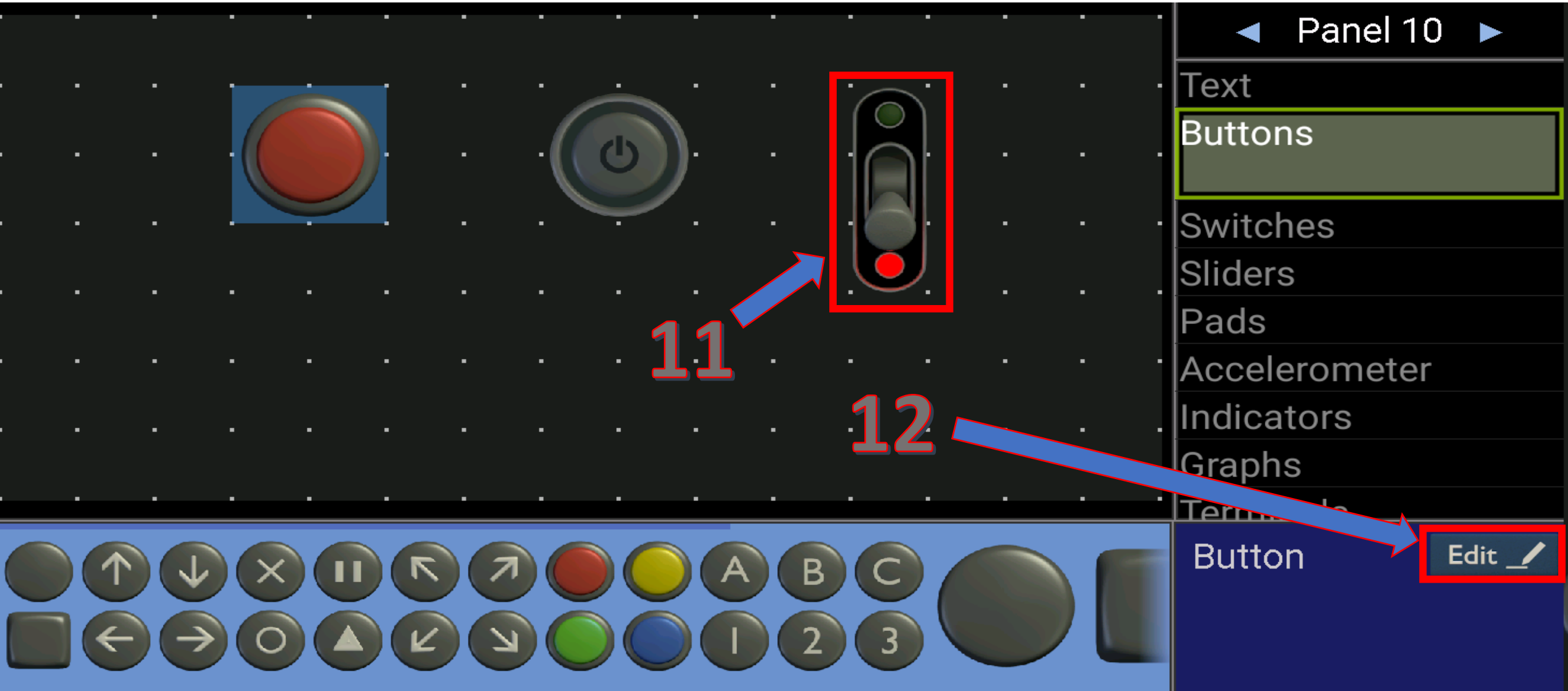
• Setting Up


Panel 10
Text
Buttons
Switches
Sliders
Pads
Accelerometer
Indicators
Graphs
8
9
A
B
C
1
2
3
Button
Edit
Press sends "R"
Release sends "r"

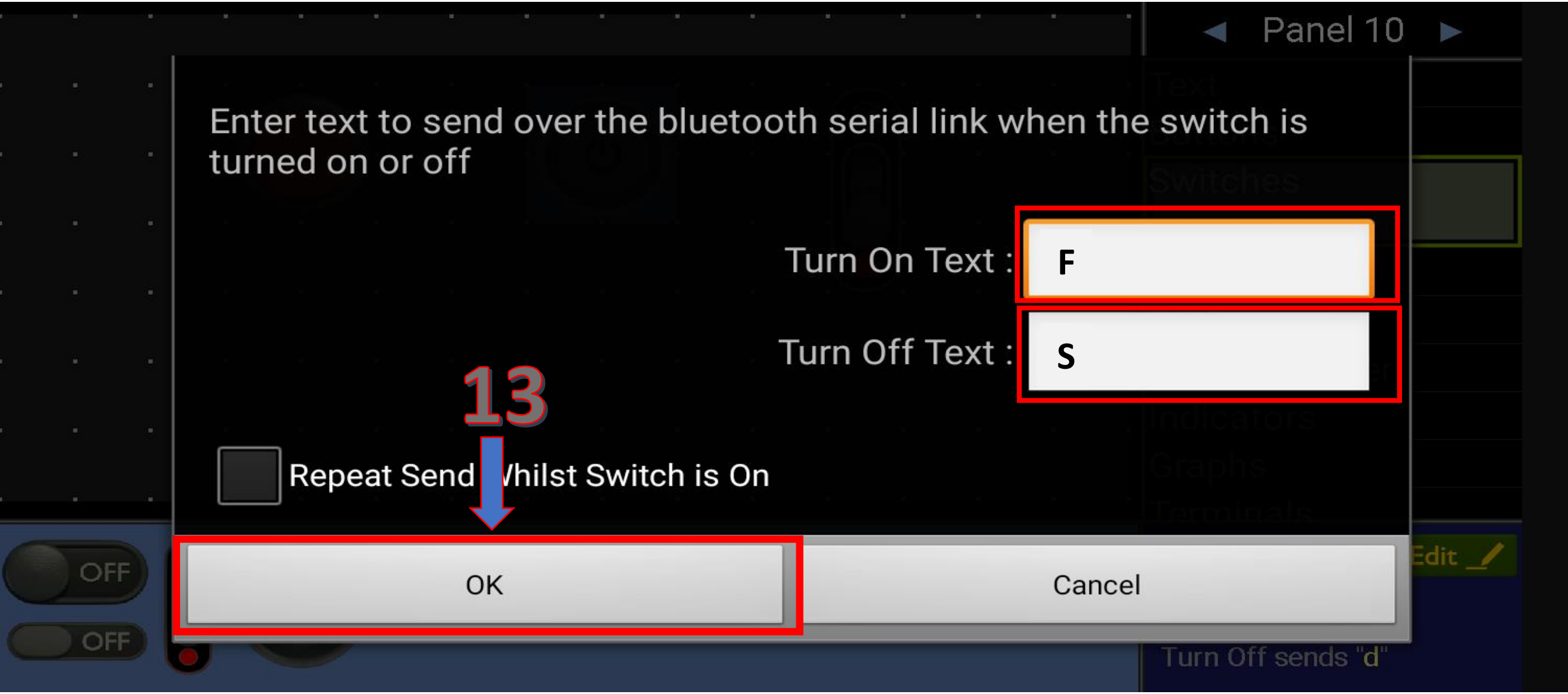
• Setting Up


Panel 10
Enter text to send over the bluetooth serial link when the button is pressed or released
Press Text :
F
Release Text :
S
10
OK
Cancel
Button
Edit
Press sends "R"
Release sends "r"

• Setting Up

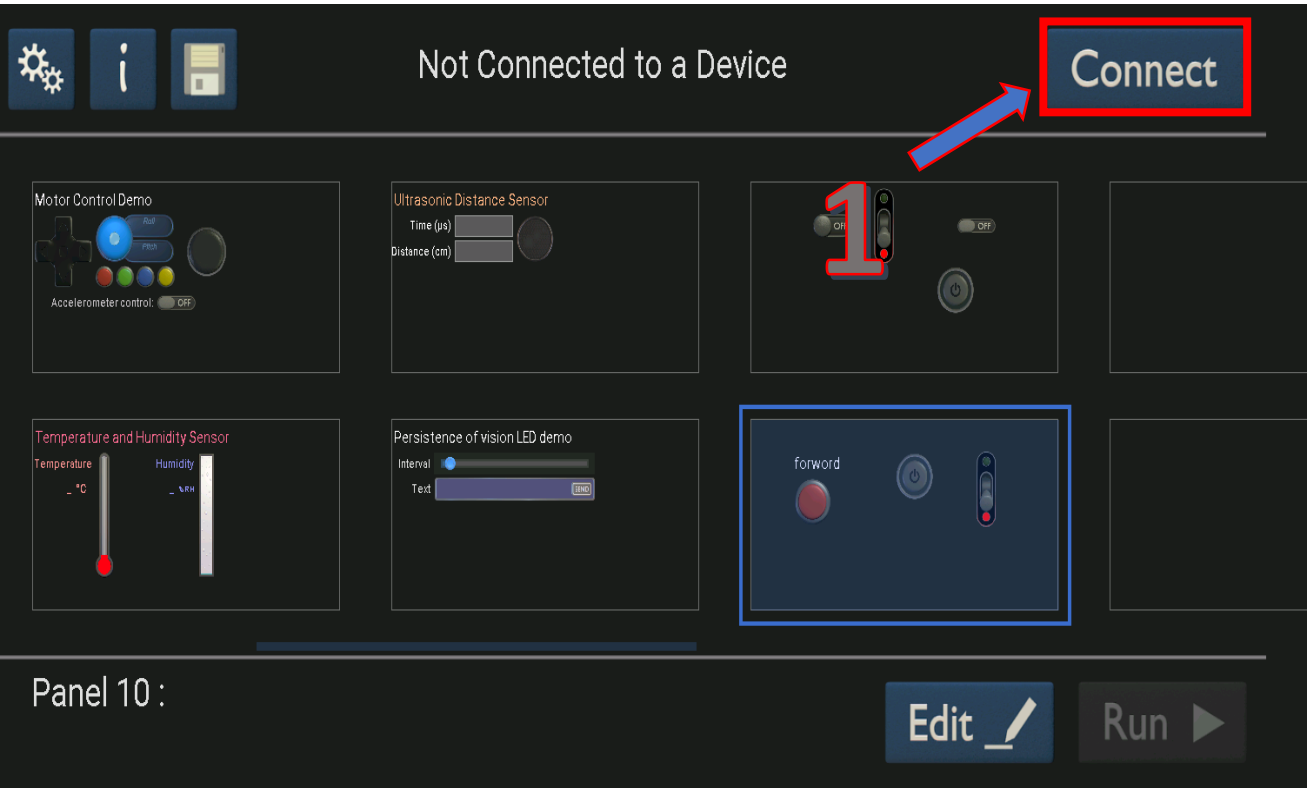

Panel 10
Text
Buttons
Switches
Sliders
Pads
Accelerometer
Indicators
Graphs
11
12
A
B
C
1
2
3
Button
Edit

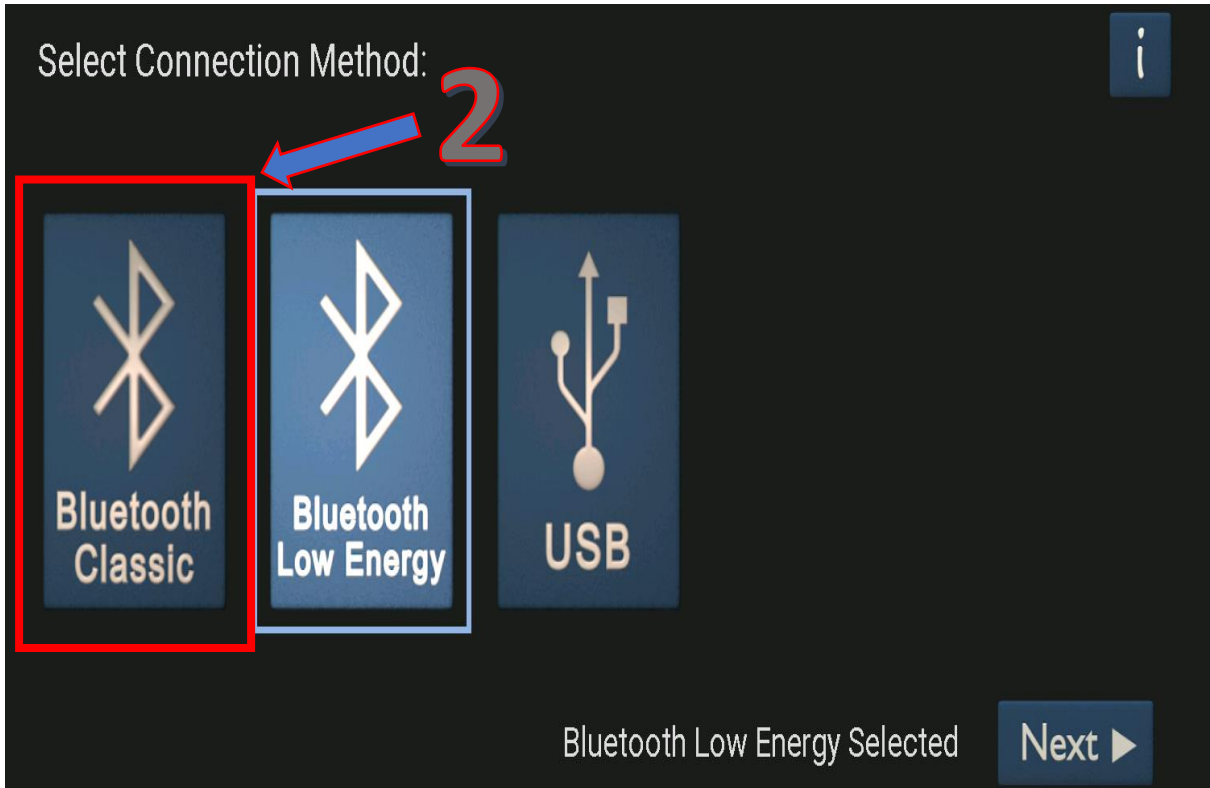
• Setting Up


Panel 10
Enter text to send over the bluetooth serial link when the switch is turned on or off
Turn On Text :
F
Turn Off Text :
S
13
Repeat Send Whilst Switch is On
OK
Cancel
OFF
OFF
Edit
Turn Off sends "d"

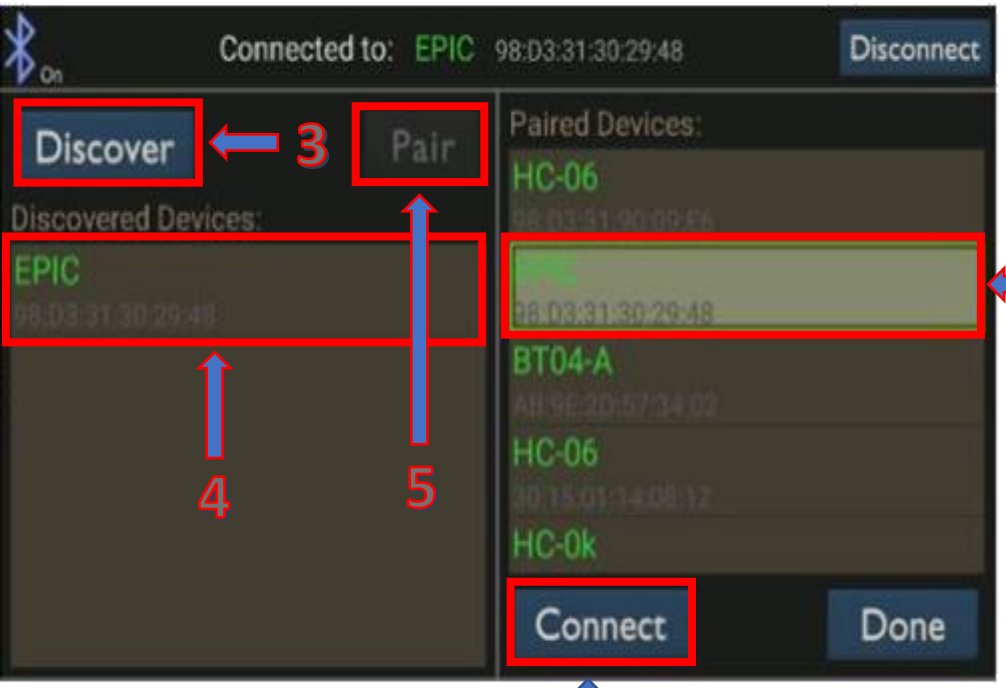
• Setting Up

return to background

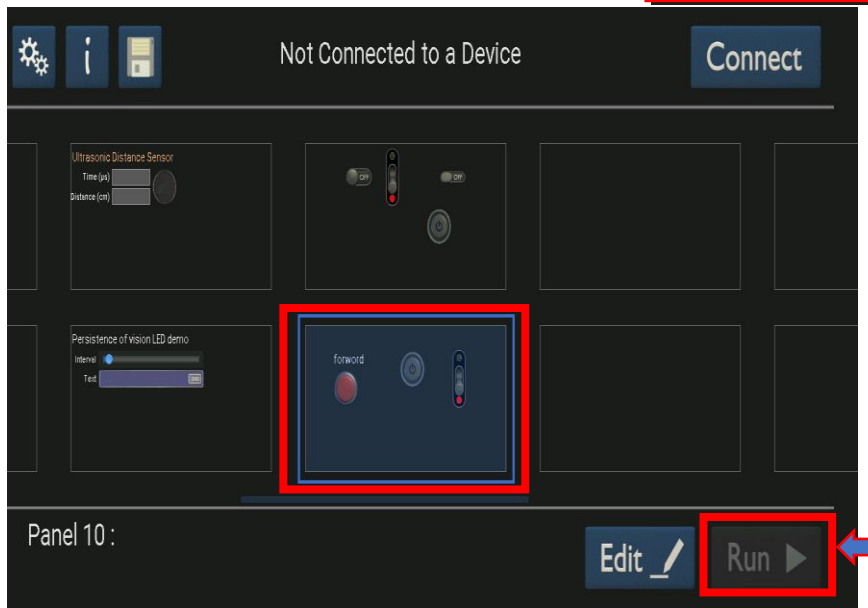


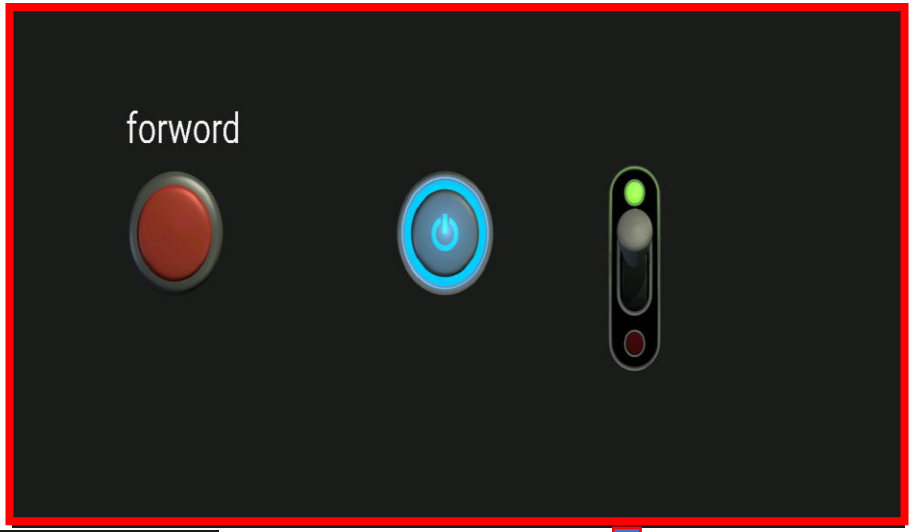


• Setting Up



password
0000 -1234

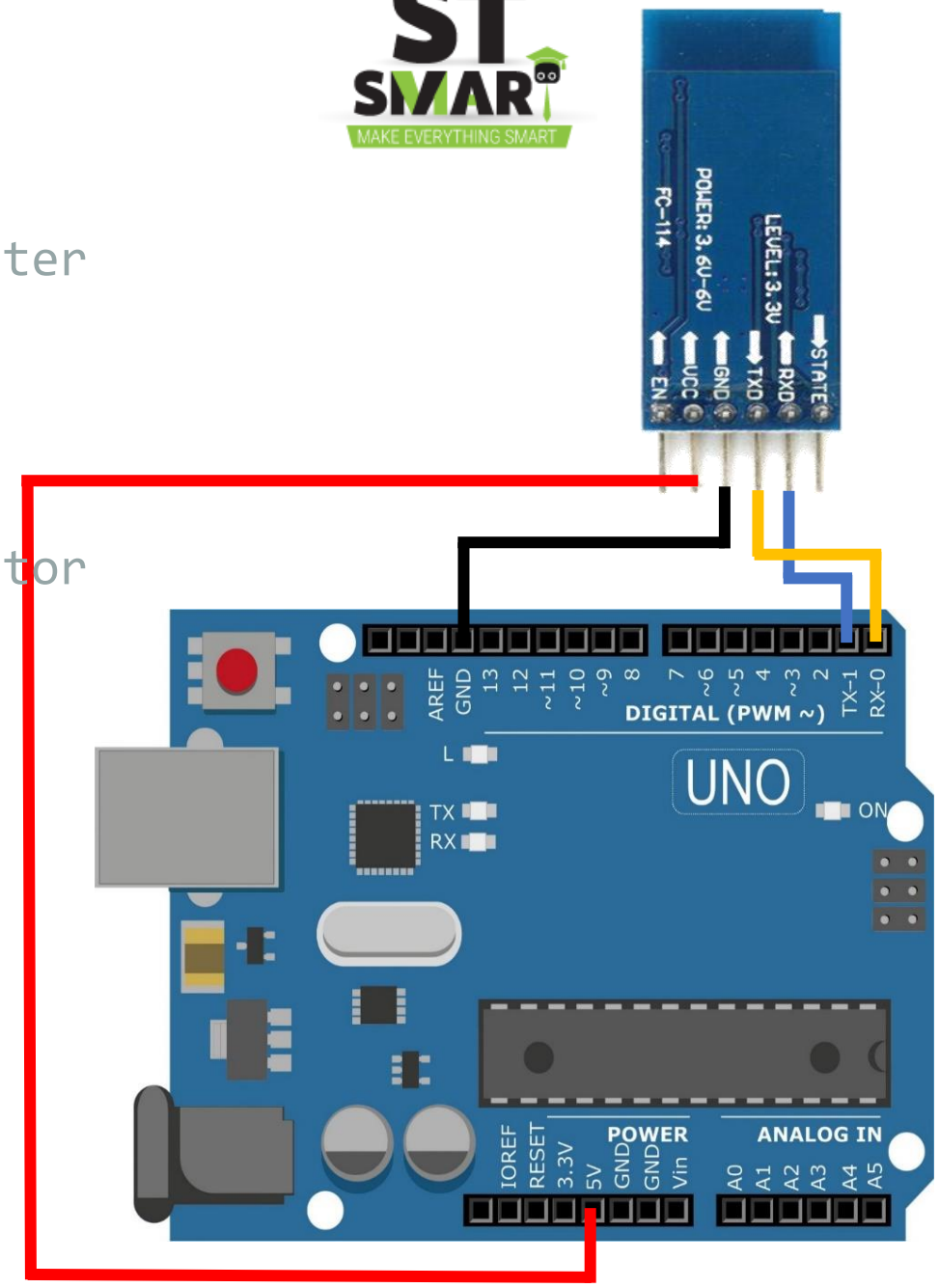




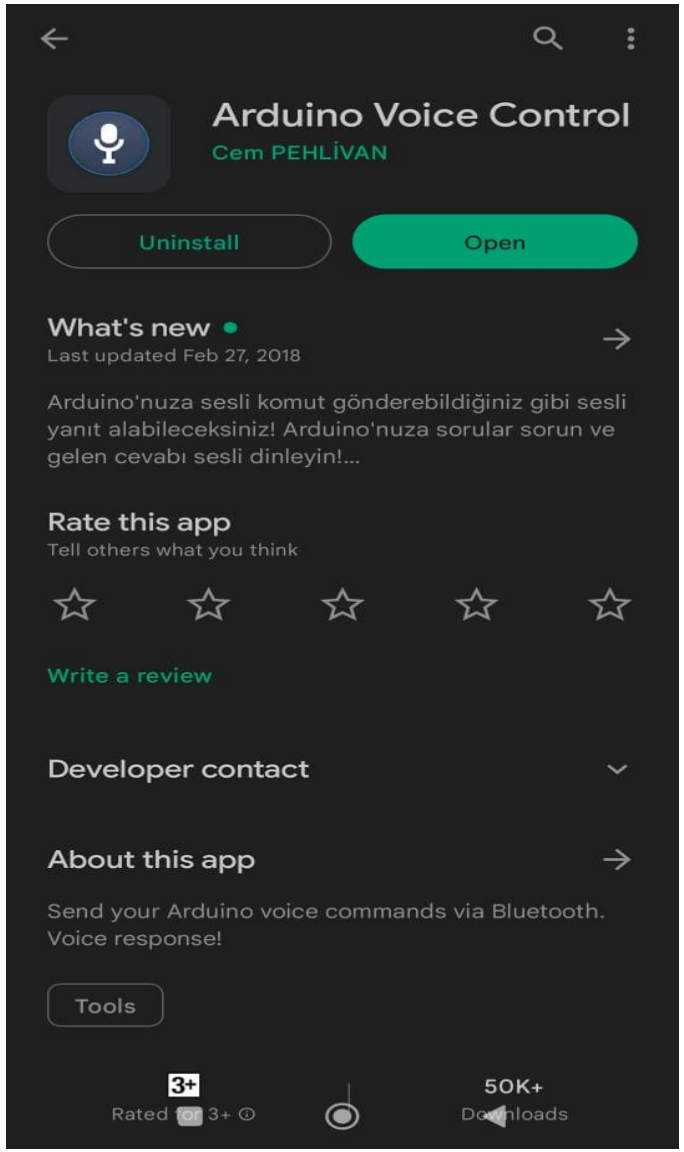
final control panel

• Example

```
const int red = 13;
String reading;  // data type used to store a character
value.
void setup( )
{
 pinMode(red, OUTPUT);
 Serial.begin(9600);  // Adjust speed of serial monitor
}
void loop( ) {
  if(Serial.available()>0) {
     reading=Serial.readString();

if (reading=="turn on"){ digitalWrite(red,1);}
else if (reading=="turn off"){
               digitalWrite(red,0); }
}}
```



• Arduino Voice Control



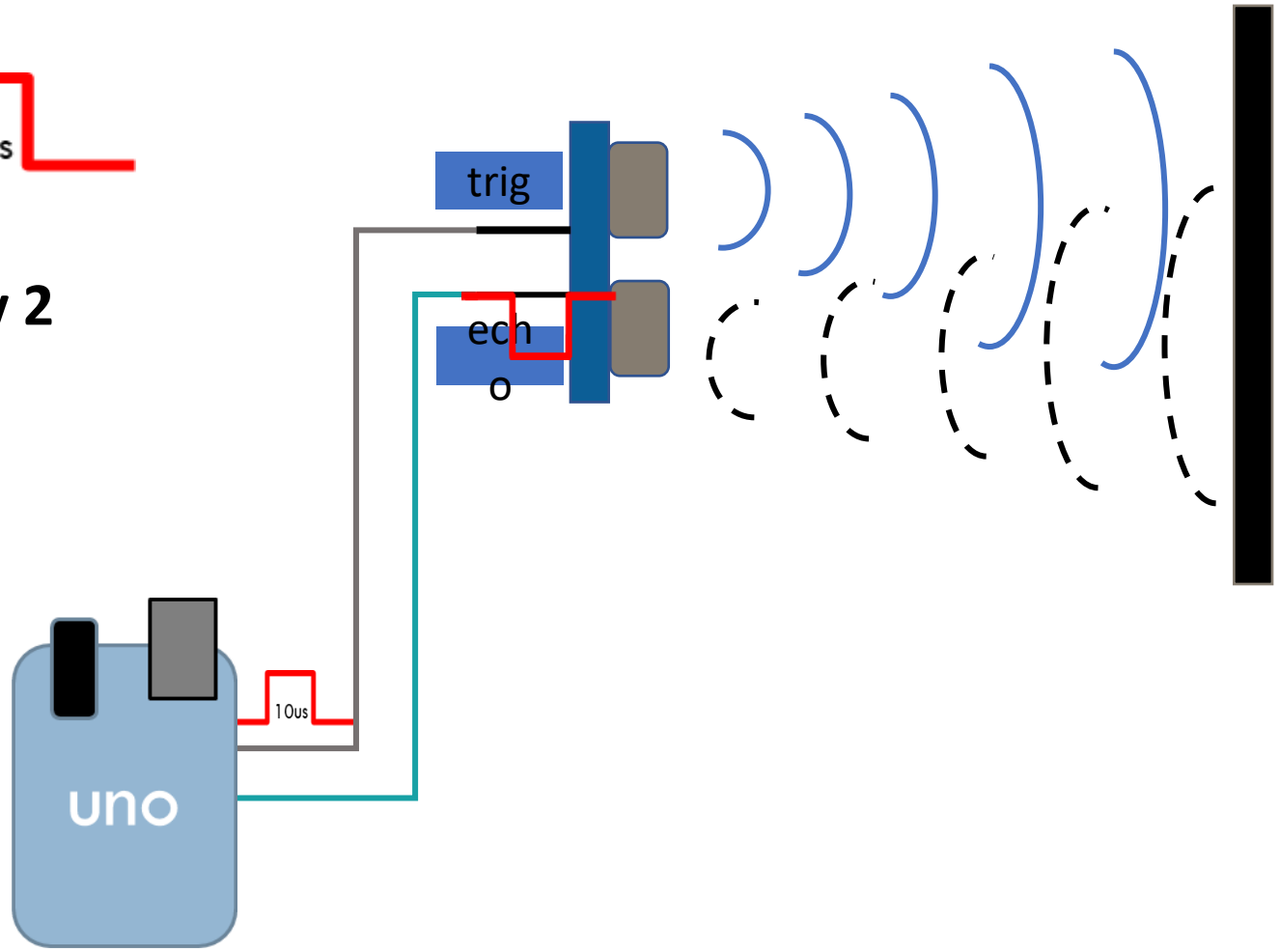
• UltraSonic Sensor

- The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do
- Power Supply :+5V DC
- Ranging Distance : 2cm – 400 cm
- Trigger Input Pulse width: 10uS
- waves are sound waves

• Basic Principle

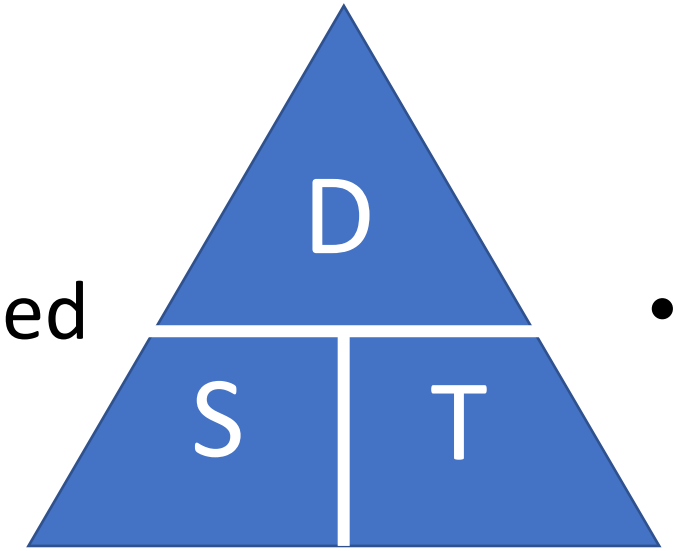
- signal HIGH → 10usec

- We need to divide the travel time by 2
- speed sound 343 m/s
- Arduino calculate the time
- after receiving wave echo send low signal



• Basic Principle

- Time = Distance / Speed

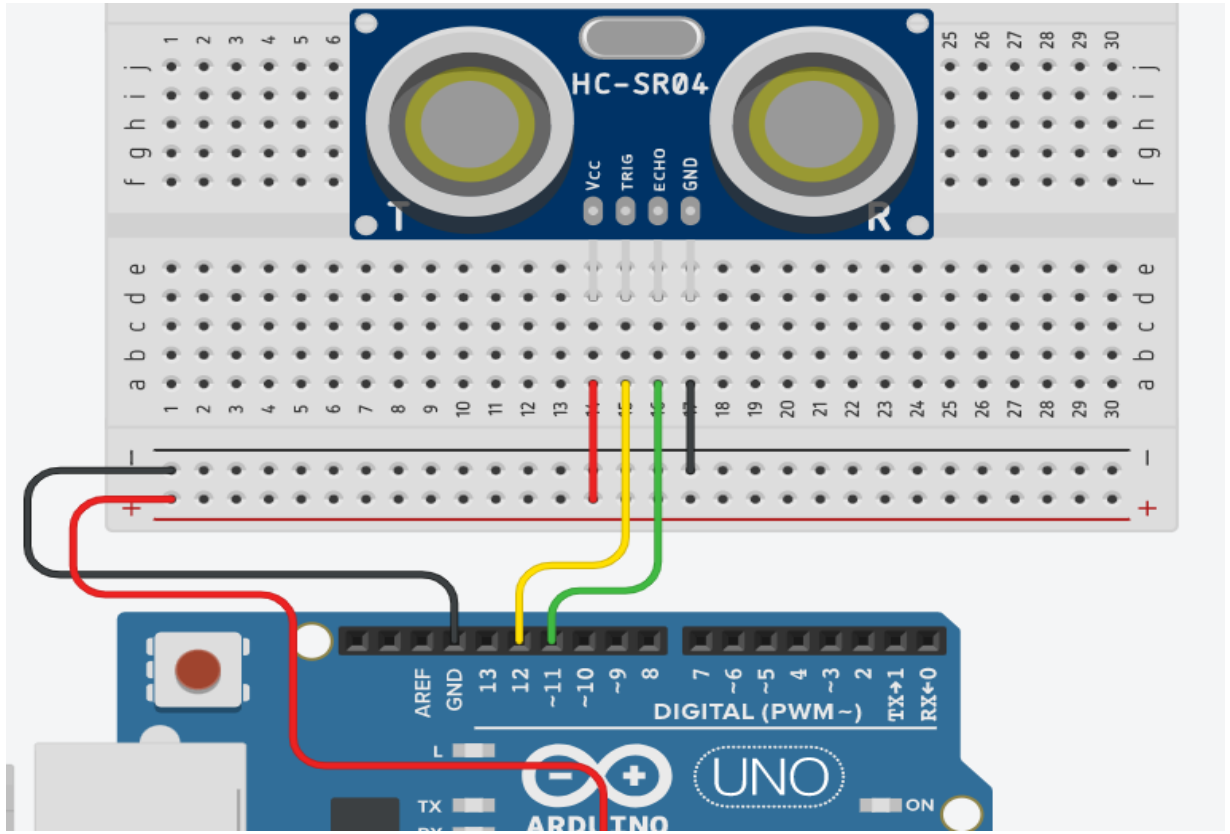


- Speed = Distance / Time

- Distance = Speed * Time

• Code

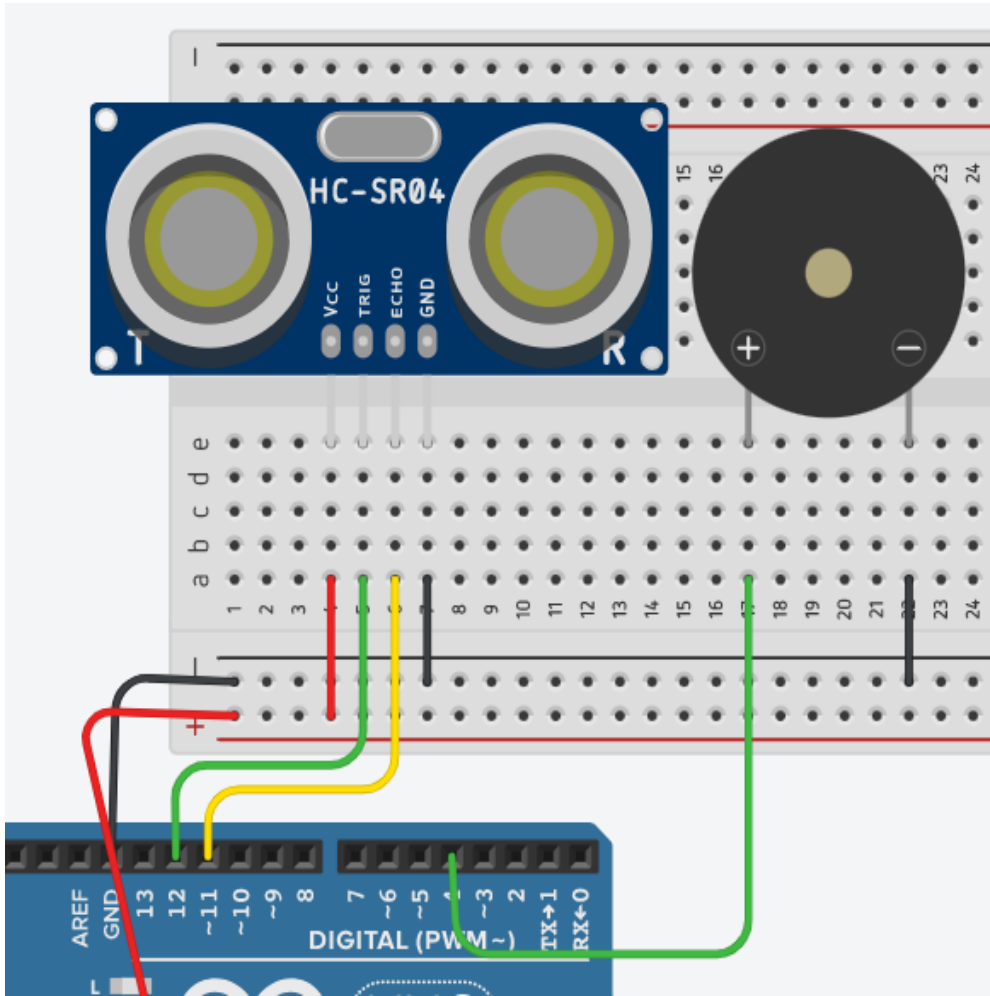
```
#define trigPin 12
#define echoPin 11
 long duration, distance;
void setup() {
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) * 0.0343;        // 343 * (100/1000000) = 343/10000
Serial.println(distance);
delay(5); // wait till next scan
}
```



• Example

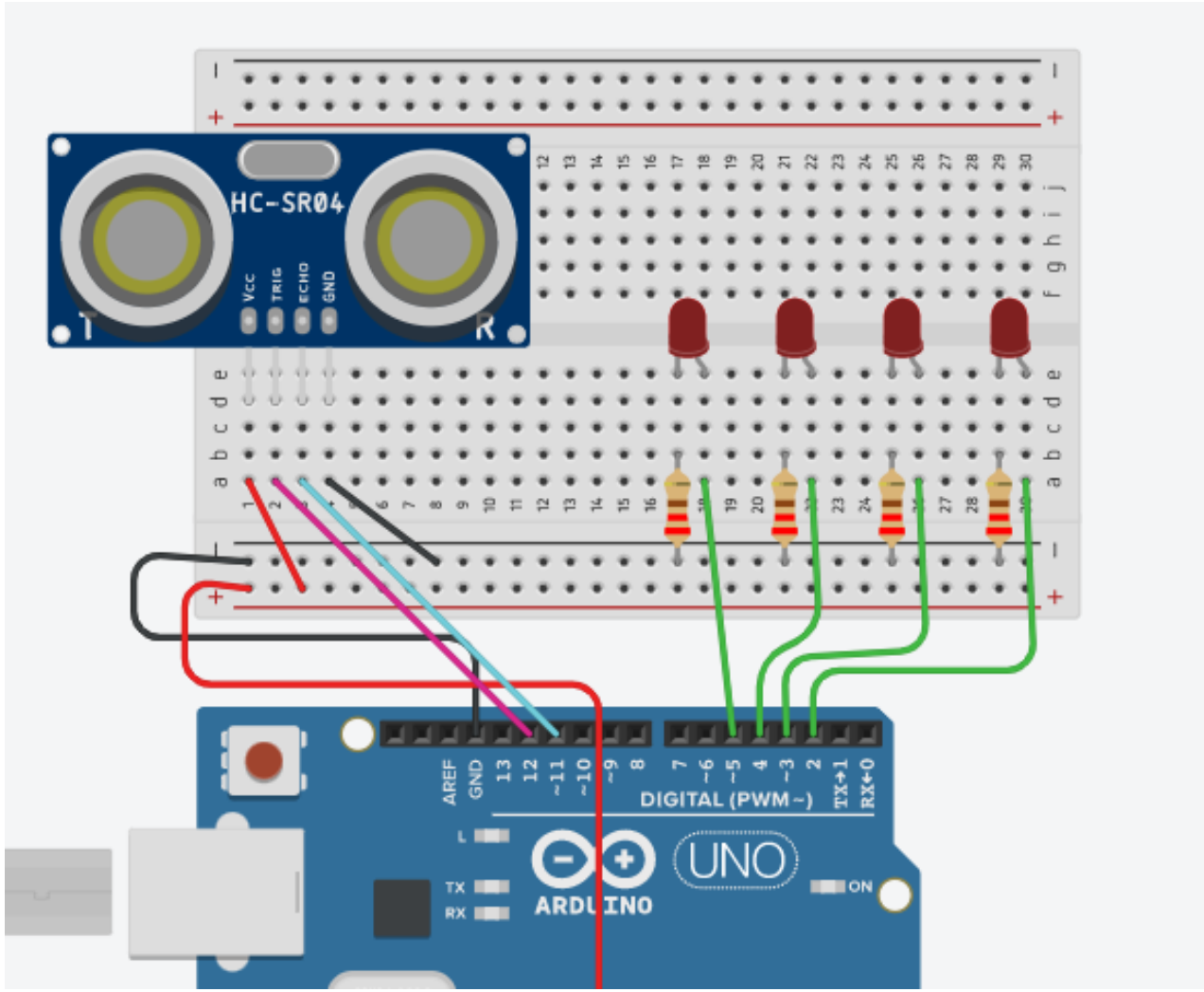
```
#define trigPin 12
#define echoPin 11
long duration, distance;
void setup() {
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(4, OUTPUT);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) * 0.0343;
Serial.println(distance);
delay(5); // wait till next scan
```

```
if(distance<=20){
  digitalWrite(4,1);
  delay(distance*30);
  digitalWrite(4,0);
  delay(distance*30);
}}
```



• Task

- Use 4 led’s to indicate distance

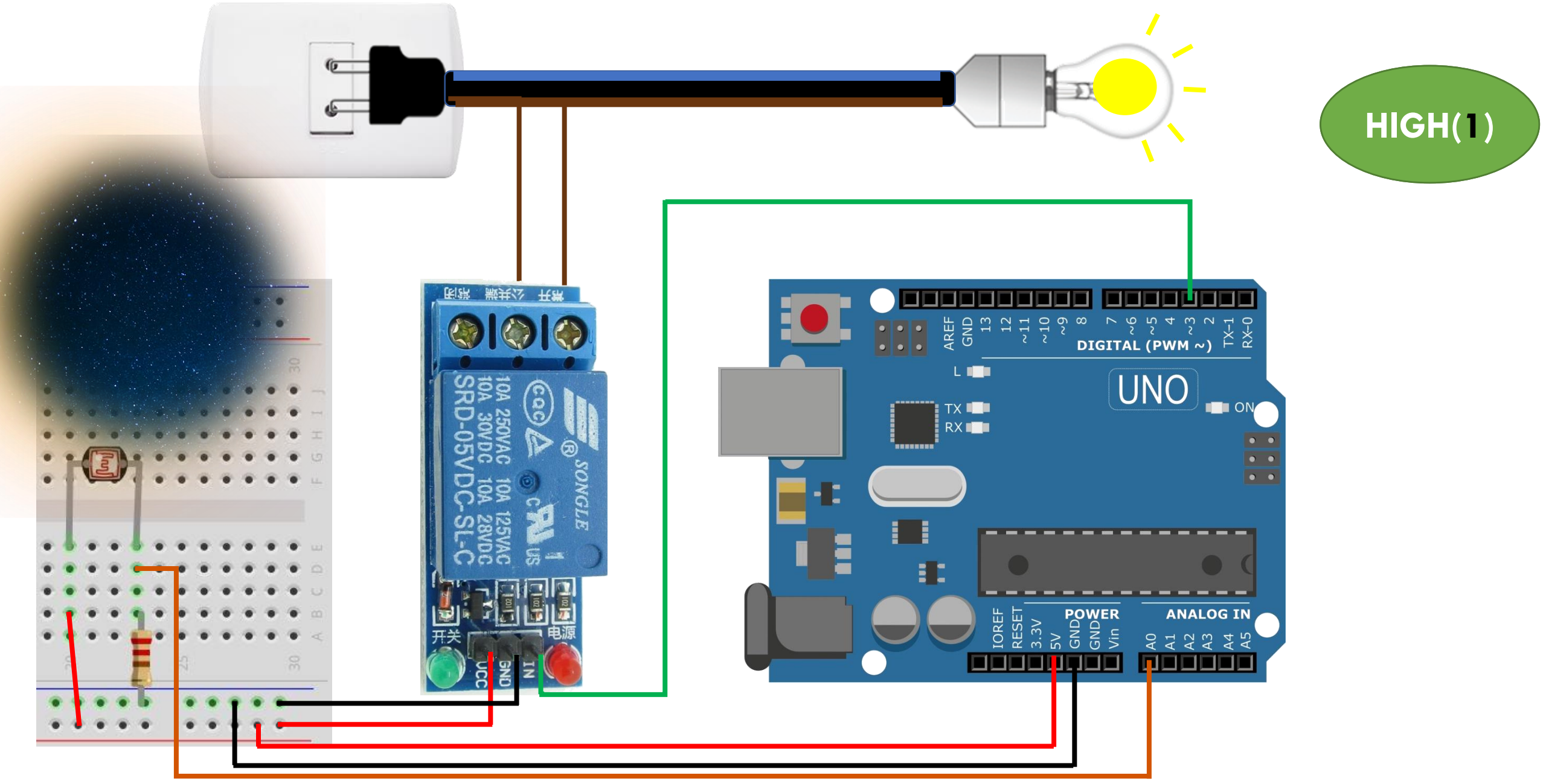


THANKS FOR COMING

LECTURE 4

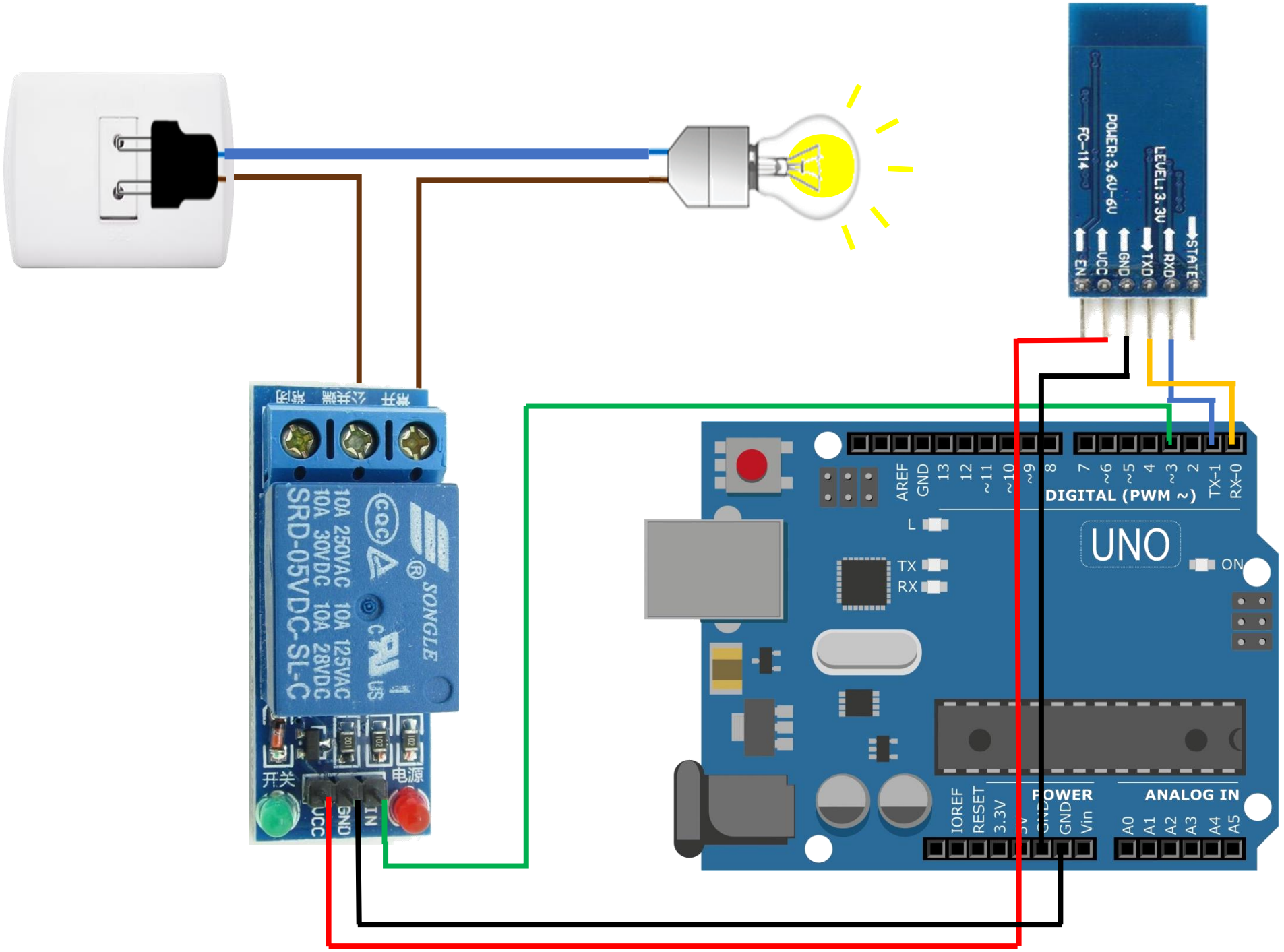
(Practical)

Automatic Night Light circuit (Wiring)



• Automatic Night Light circuit (Code)

```
#define ldr A0
#define led 3
int threshold = 40;
int level;
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}
void loop() {
  level = analogRead(ldr);
   Serial.println(level);
   if (level < threshold) { digitalWrite(led, 1); }
    else { digitalWrite(led, 0);}
}
```

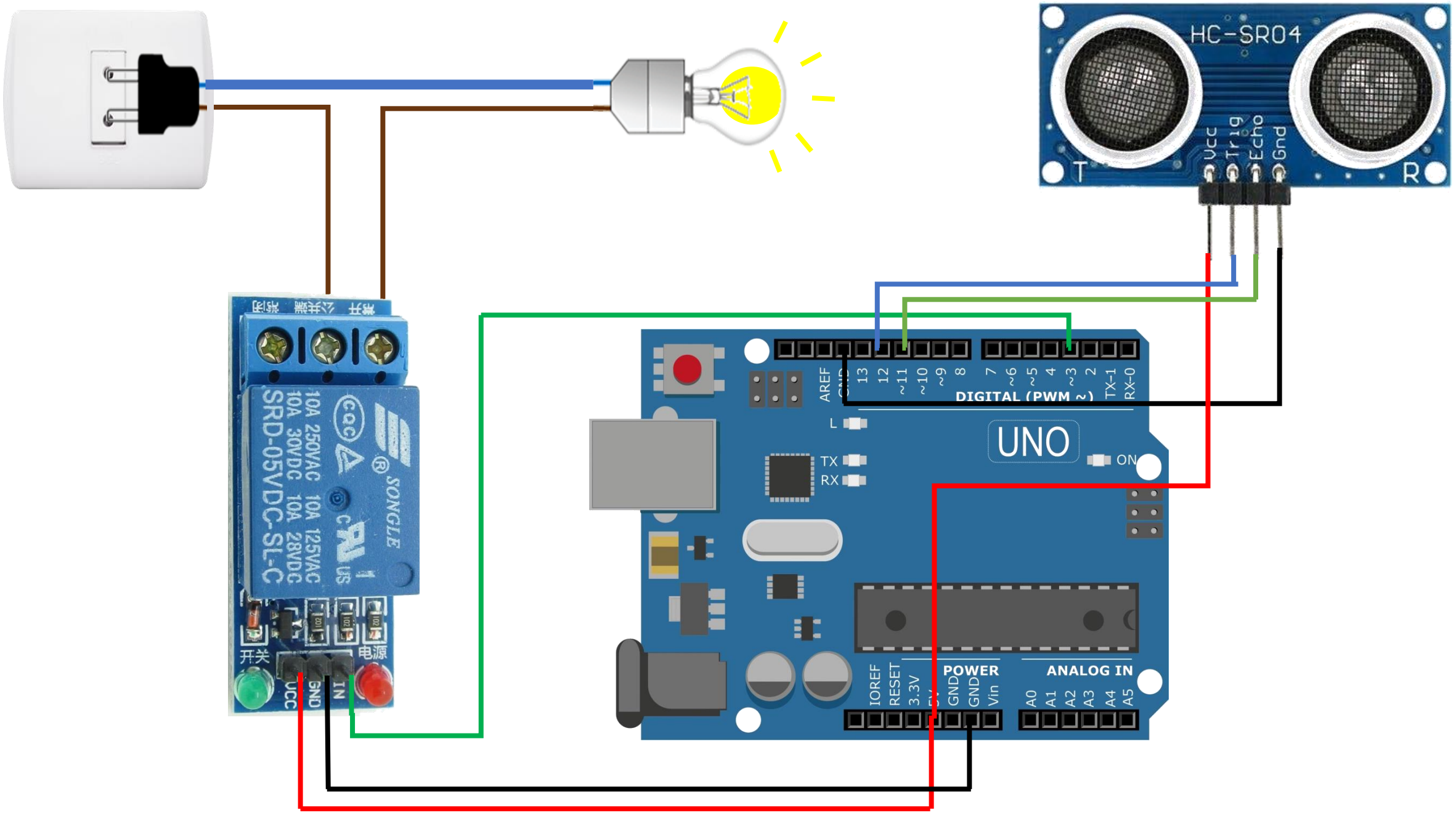

SRD-05VDC-SL-C
10A 250VAC 10A 125VAC
10A 30VDC 10A 28VDC
SONGLE
开关
电源
VCC
GND
IN
FC-114
POWER: 3.6V-6V
LEVEL: 3.3V
EN
VCC
GND
TXD
RXD
STATE
UNO
DIGITAL (PWM ~)
ANALOG IN
POWER
AREF
GND
L
TX
RX
ON
IOREF
RESET
3.3V
Vin
A0
A1
A2
A3
A4
A5
TX→1
RX←0

• Light Control with Smartphone

```
const int red = 13;
char reading;  // data type used to store a character value.
void setup( )
{
 pinMode(red, OUTPUT);
 Serial.begin(9600);
}
void loop( ) {
  if(Serial.available()>0) {
     reading=Serial.read();
     switch(reading){
        case 'F': digitalWrite(red,1);
                  break;
        case 'S': digitalWrite(red,0);
                  break;
       }
   }
}
```

• Light Control With Voice

```
const int red = 13;
String reading;  // data type used to store a character value.
void setup( )
{
 pinMode(red, OUTPUT);
 Serial.begin(9600);  // Adjust speed of serial monitor
}
void loop( ) {
  if(Serial.available()>0) {
     reading=Serial.readString();

    if (reading=="turn on"){ digitalWrite(red,1);}
     else if (reading=="turn off"){   digitalWrite(red,0); }

}}
```


HC-SR04
T
R
Vcc
Trig
Echo
Gnd
SRD-05VDC-SL-C
10A 250VAC 10A 125VAC
10A 30VDC 10A 28VDC
SONGLE
开关
电源
VCC
GND
IN
UNO
ON
L
TX
RX
DIGITAL (PWM ~)
POWER
ANALOG IN
IOREF
RESET
3.3V
GND
Vin
A0
A1
A2
A3
A4
A5

• Lighting Control With Motion

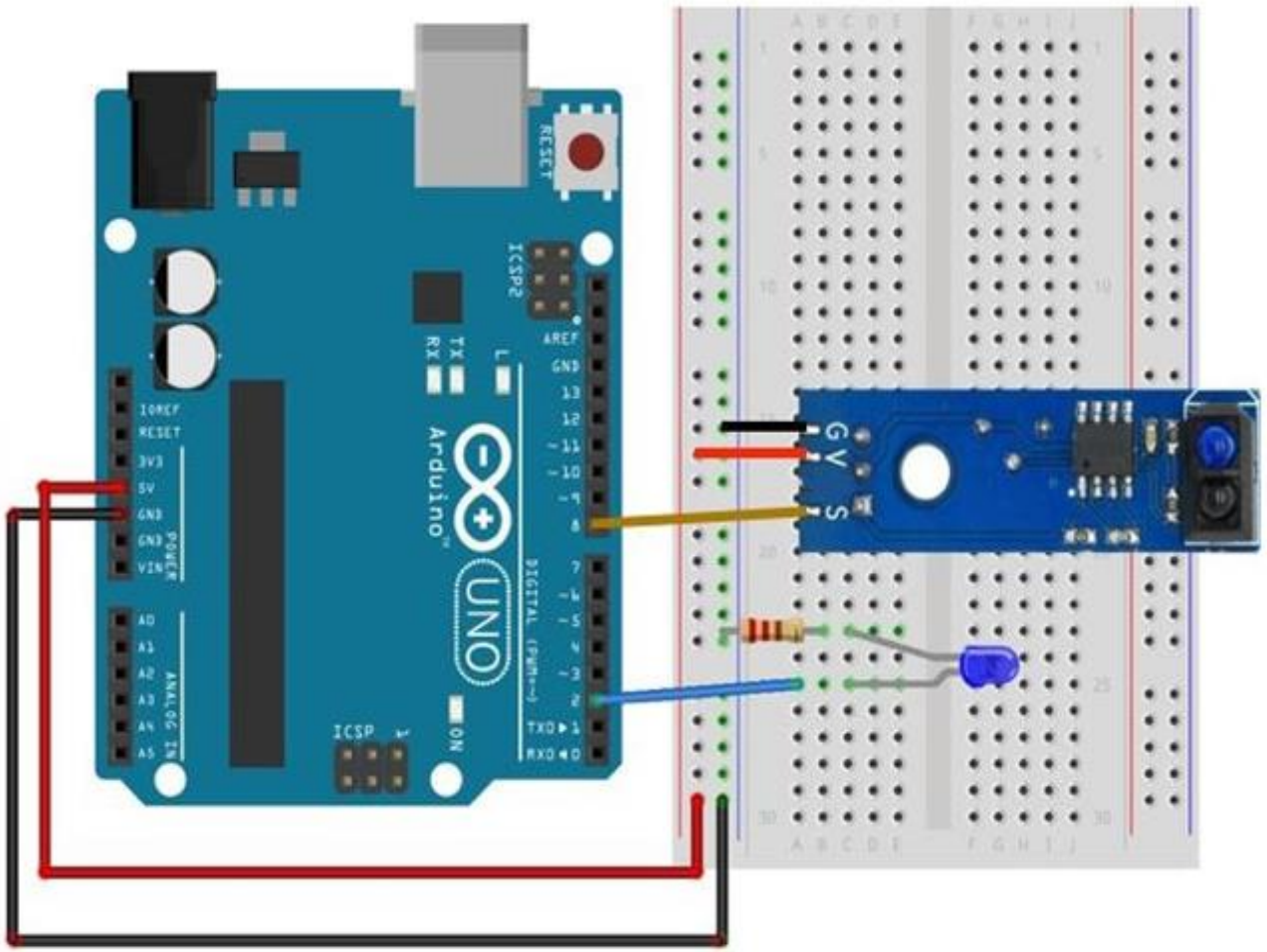
```
#define trigPin 12
#define echoPin 11
long duration, distance;
void setup() {
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(3, OUTPUT);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) * 0.0343;
Serial.println(distance);
delay(5); // wait till next scan
```

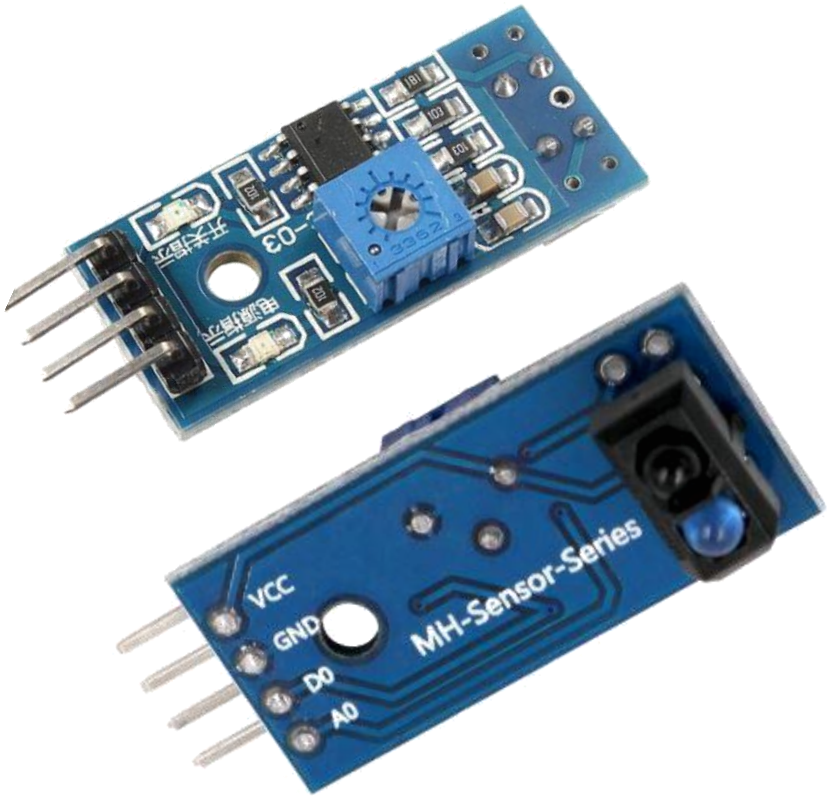
```
if(distance<=20){
  digitalWrite(3,1);
  delay(4000);
  digitalWrite(3,0);

}}
```

THANKS FOR COMING

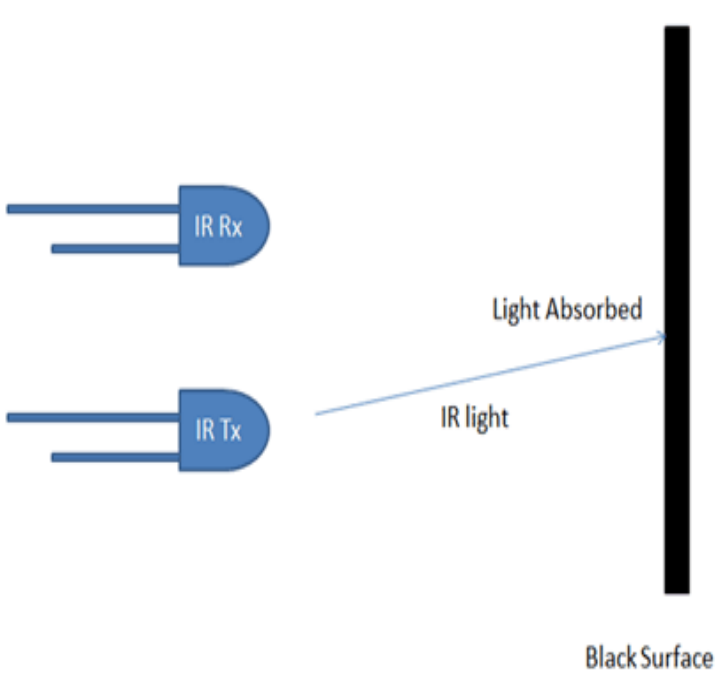
LECTURE 5

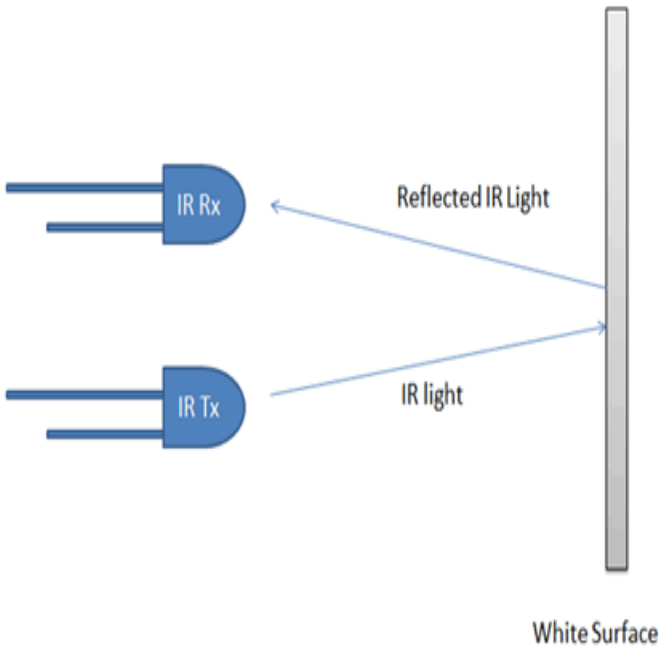
• Line Tracker Sensor (TCRT5000)





• Basic Principle





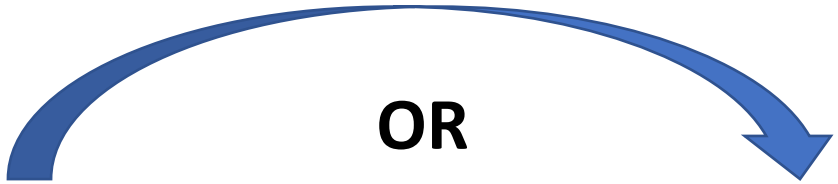
reflection distance: 1mm to 25mm

Reading
1

Reading
0

• Code

```
int lineTracker = 8;
int led = 2;
int state = 0;
void setup()
{
 pinMode(lineTracker,INPUT);
 pinMode(led,OUTPUT);
}
void loop()
{
  state=digitalRead(lineTracker);
  digitalWrite(led,state);
}
```




```
void setup()
{
 pinMode(8,INPUT);
 pinMode(2,OUTPUT);
}
void loop()
{
  digitalWrite(led,digitalRead(8));
}
```

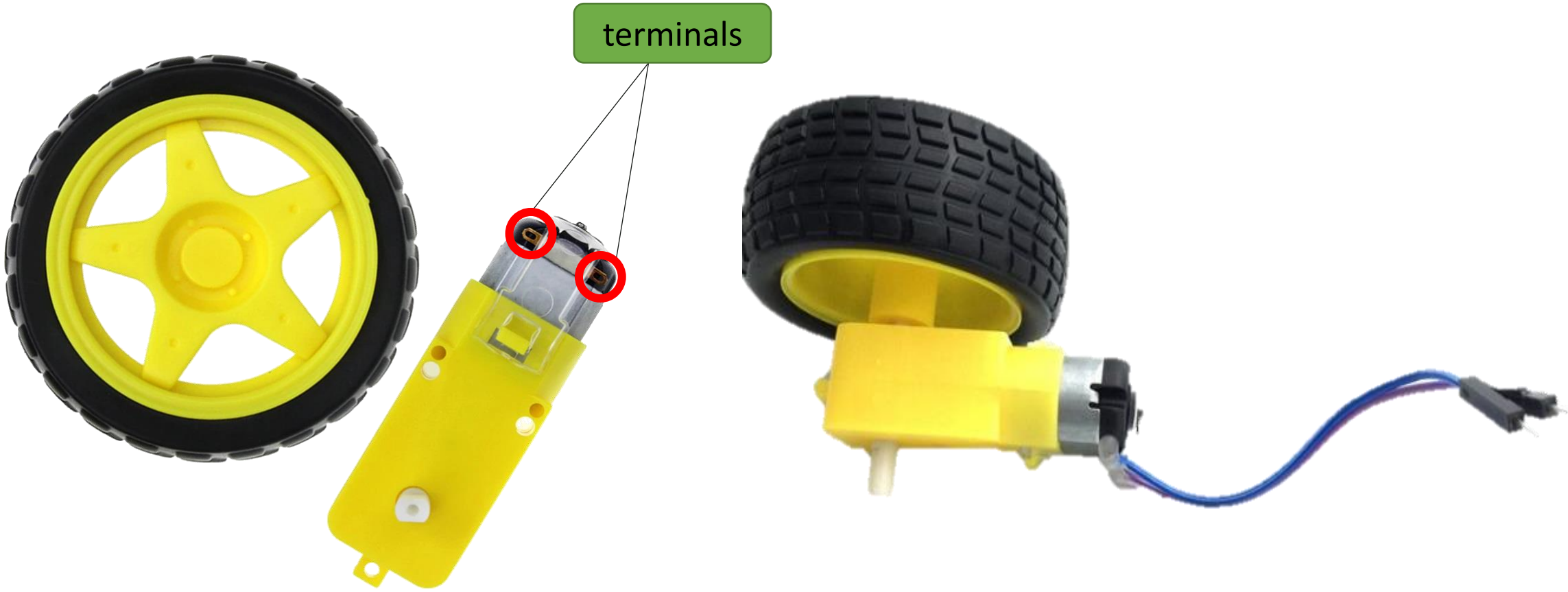
• DC Motor

DC Motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy

• DC Motor Specification

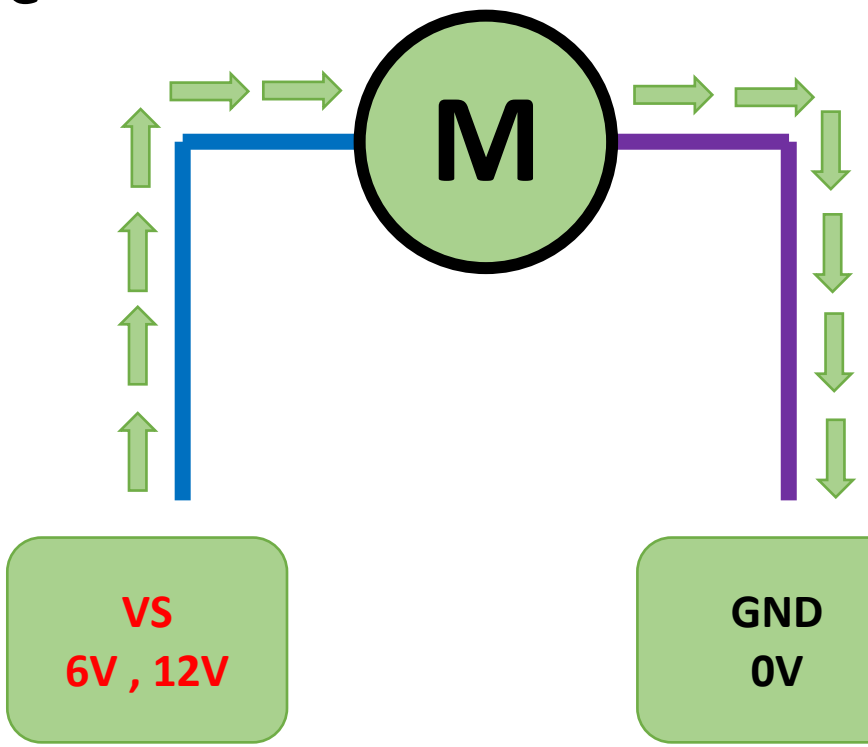
Operating Voltage	3V-12VDC
Maximum Torque	800g/cm max
Gear Ratio	1:48
Load Current	70mA (250mA max. @ 3V)
Weight	29g

• DC Motor



• Basic Principle

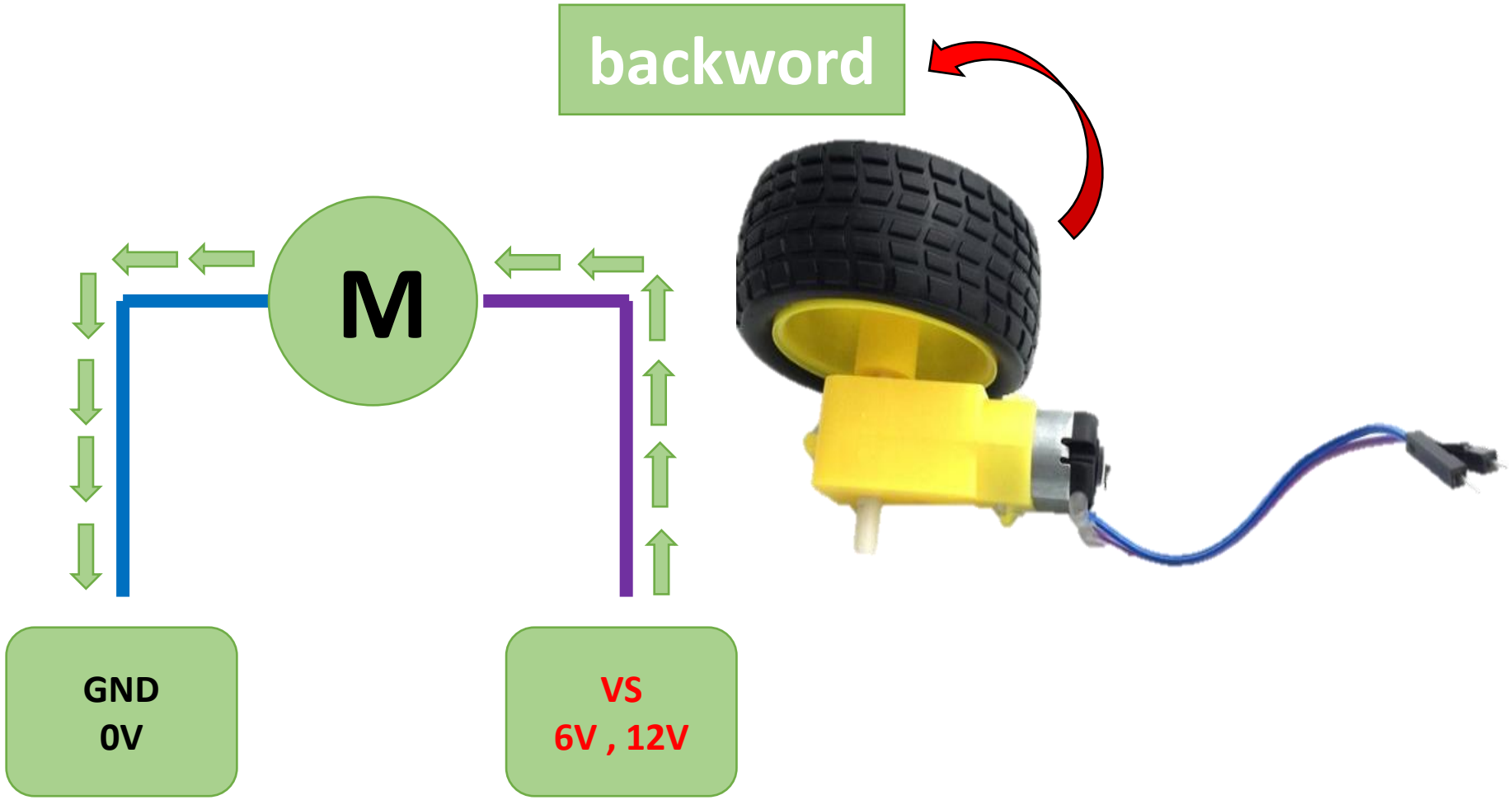
blue → positive
Purple → negative



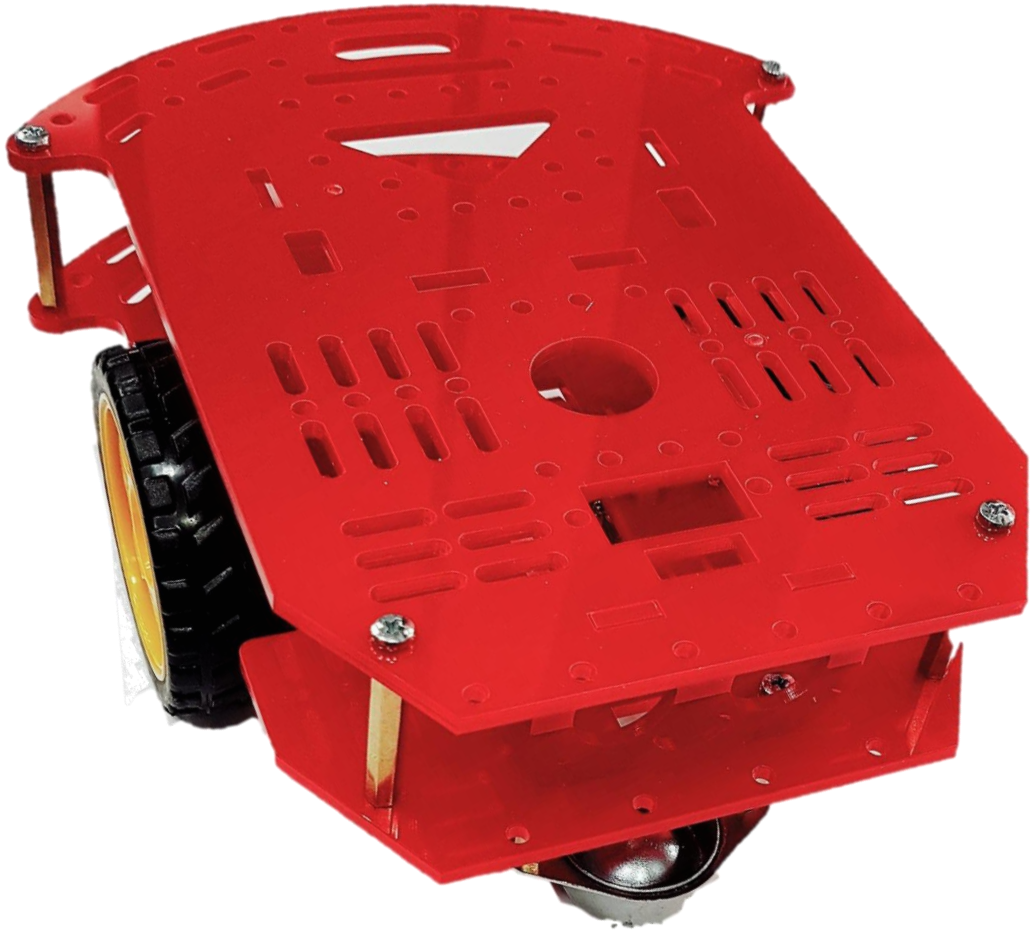
forword

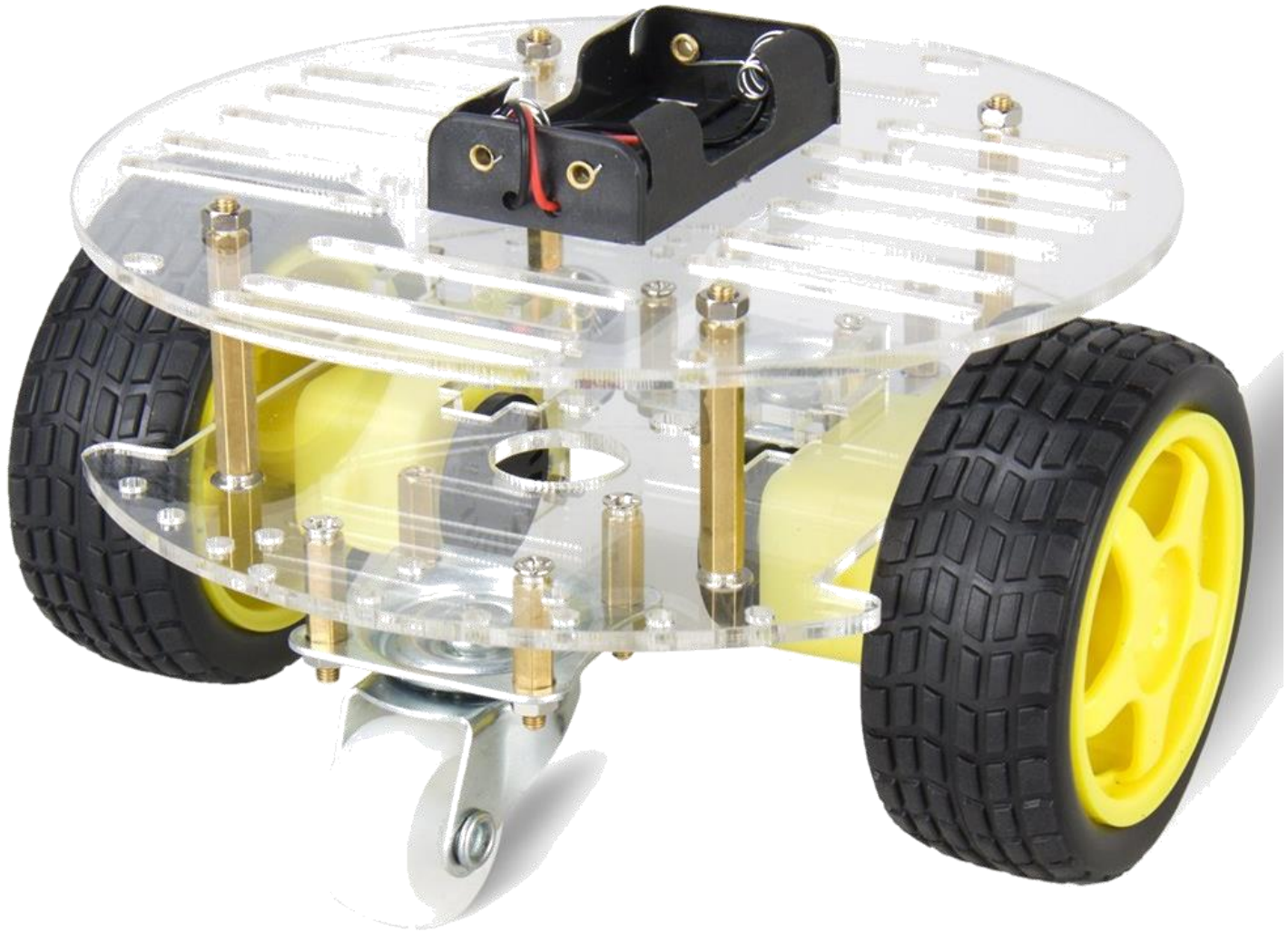
• Basic Principle

blue → positive
Purple → negative

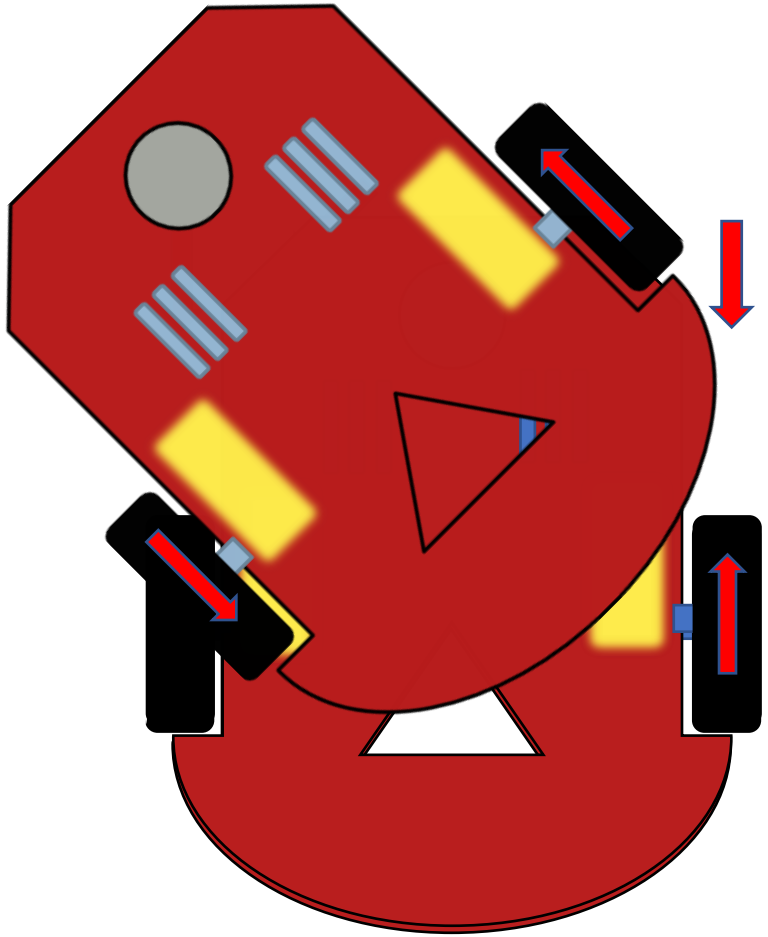


• Mobile Robot





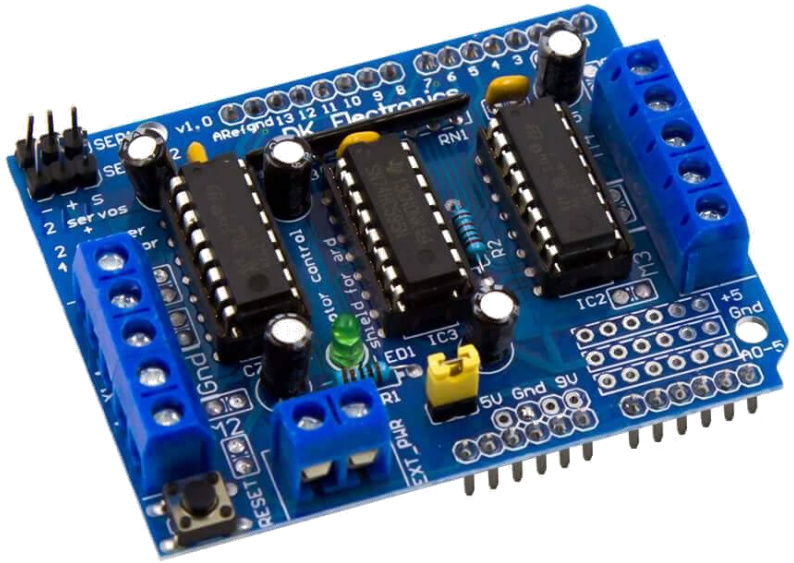
• Differential steering

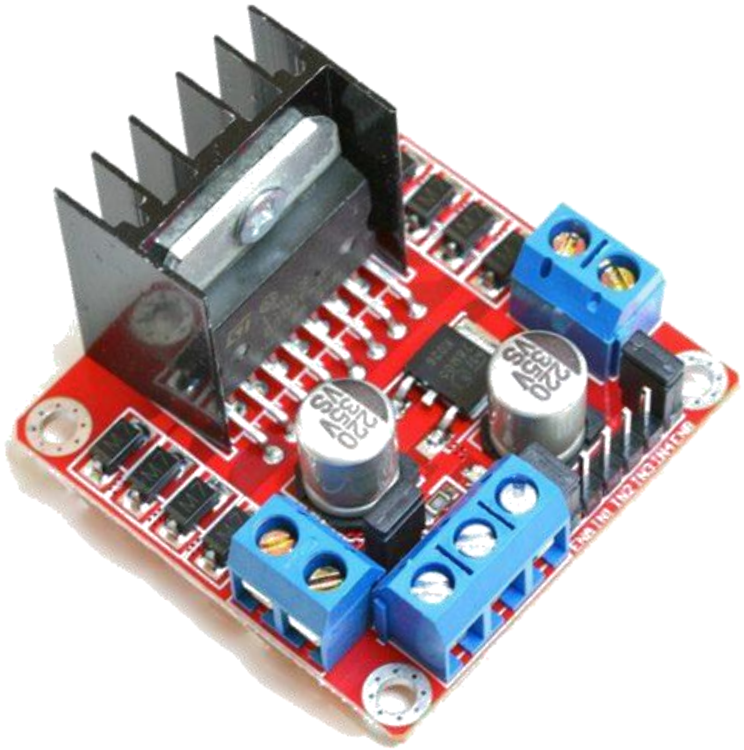


Left

• Motor Driver

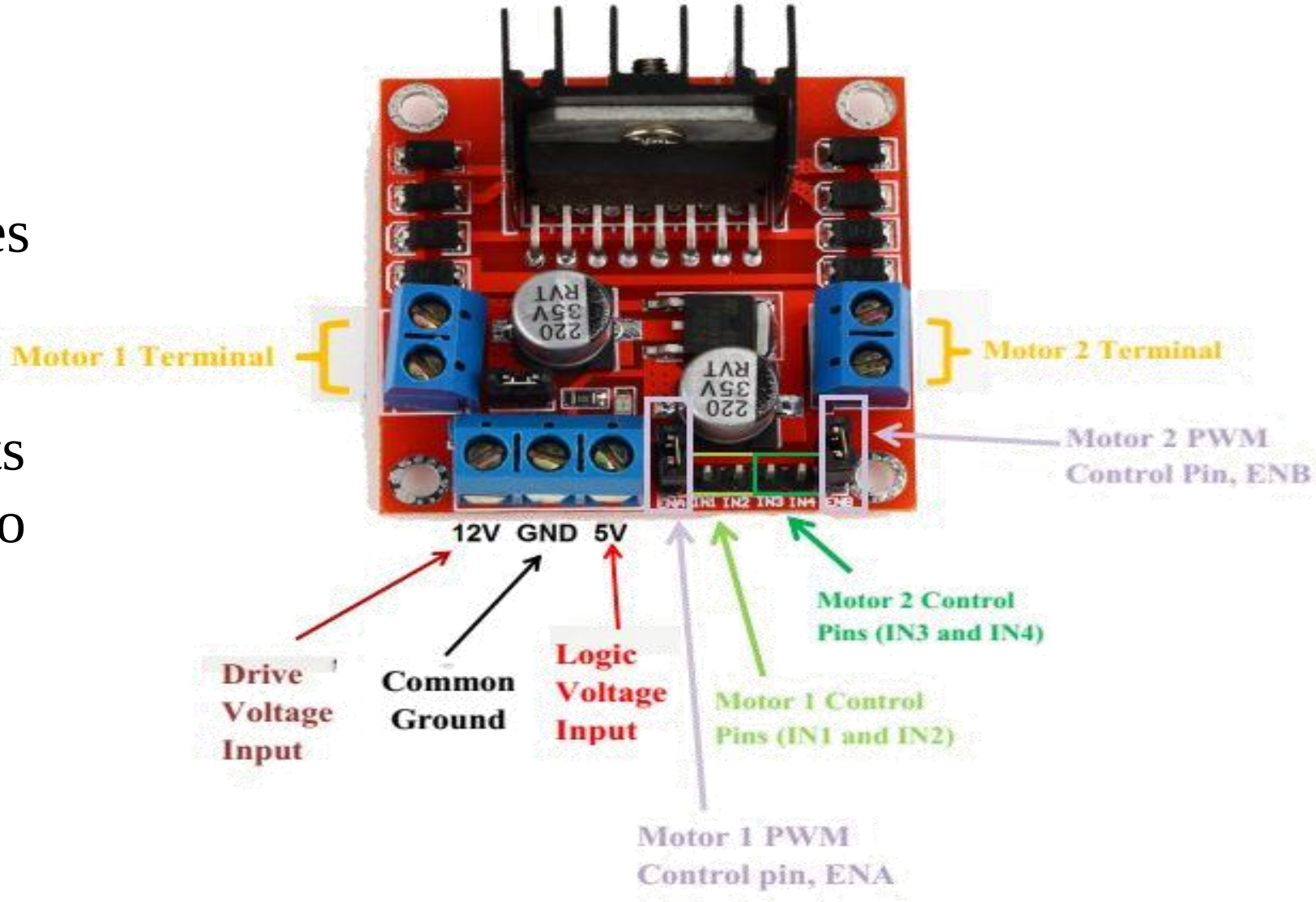
- What is motor driver and Why do we need it ?





• Motor Driver (l298N)

- **Motor driver** receives signals from the microprocessor and eventually, it transmits the converted signal to the motors.



• Direction Control

we will agree:

car is forword when

blue cable → HIGH signal

purpule cable→LOW signal

left motor

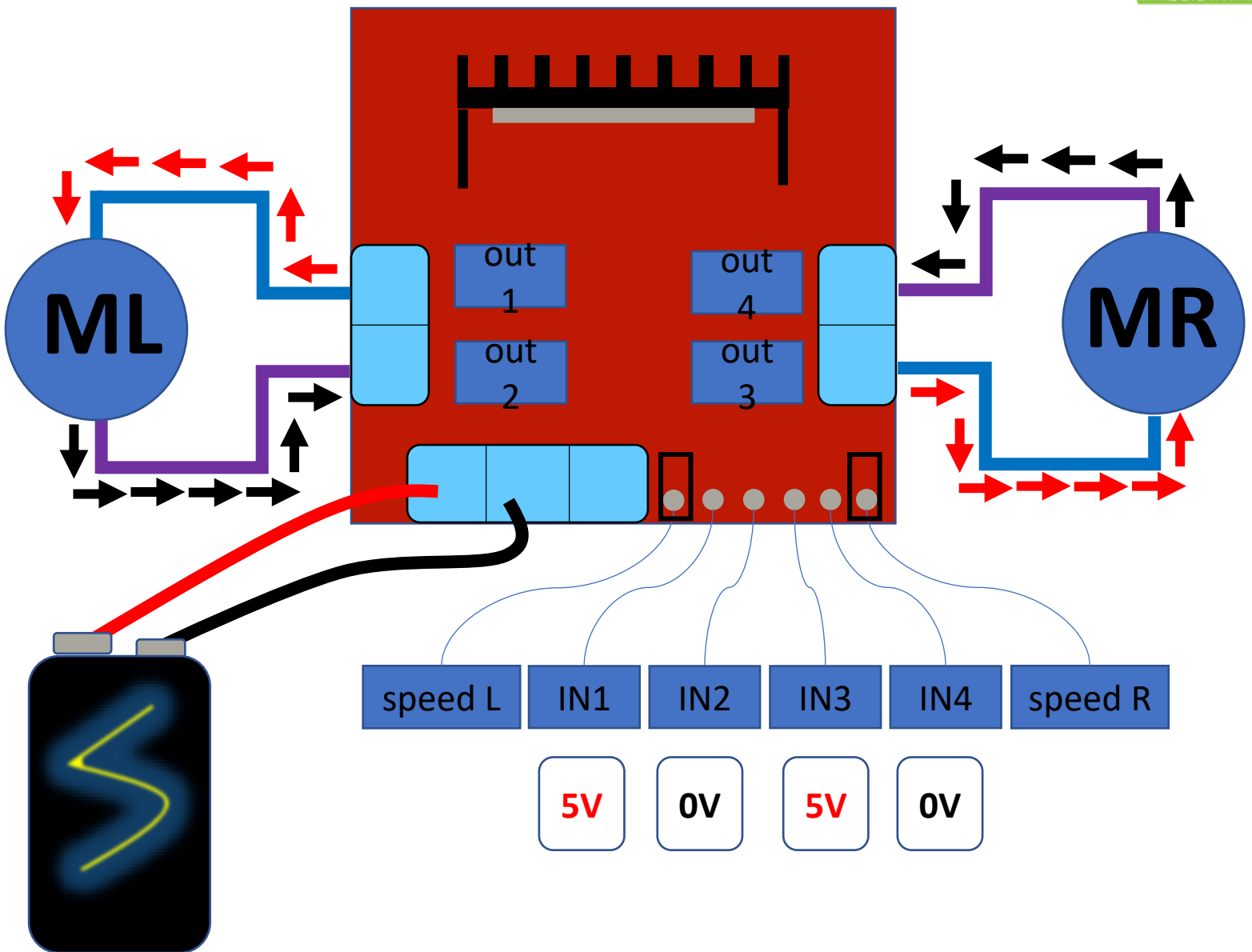
blue cable →out1

purpule cable→out2

right motor

blue cable→out3

purpule cable→out4



• Direction Control

we will agree:

car is forword when

blue cable → HIGH signal

purpule cable→LOW signal

left motor

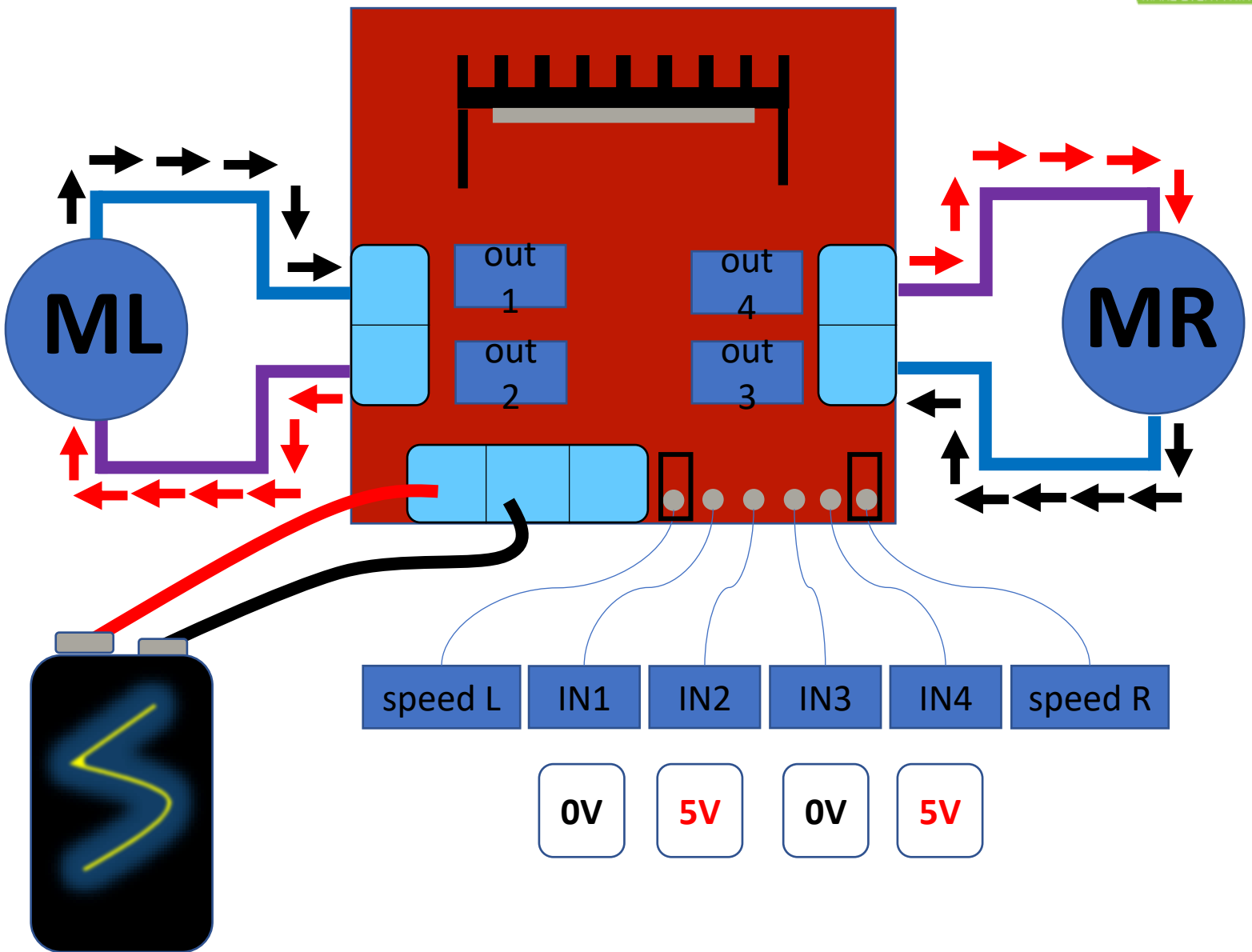
blue cable →out1

purpule cable→out2

right motor

blue cable→out3

purpule cable→out4



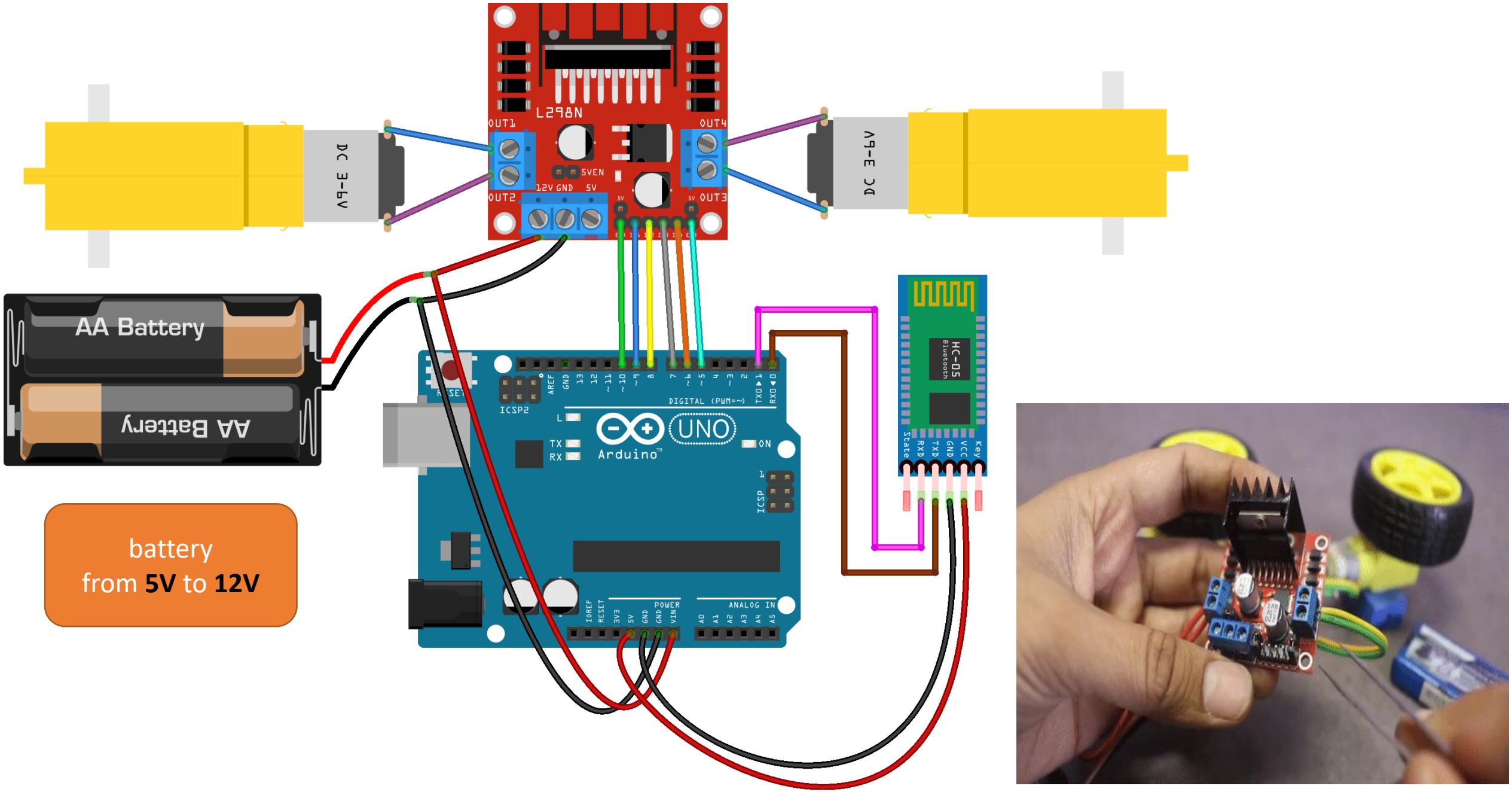
• Car Direction Test

```
#define speedL 10
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
void setup()
{
  Serial.begin (9600);
  for(int i=5 ; i<=10 ; i++)
  {
    pinMode(i, OUTPUT);
  }
}
```

```
void forword()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
void backword()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
```

```
void left()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,150);
}
void right()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,0);
}
```

```
void stopp() {
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,0);  }
void loop() {

    forword();    delay(2000);
    backword();   delay(2000);
    right();      delay(2000);
    left();       delay(2000);
    stopp();      delay(2000);

}
```


L298N
OUT1
OUT2
OUT3
OUT4
5VEN
12V GND 5V
DC 3-6V
DC 3-6V
AA Battery
AA Battery
HC-05
Bluetooth
State
RXD
TXD
GND
VCC
Key
UNO
Arduino
ICSP2
ICSP
DIGITAL (PWM=~)
POWER
ANALOG IN
battery
from 5V to 12V

• Controlled robot by Bluetooth

```
#define speedL 10
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
char Reading;
void setup()
{
 Serial.begin (9600);
 for(int i=5 ; i<=10 ; i++)
 {
  pinMode(i, OUTPUT);
 }
}
```

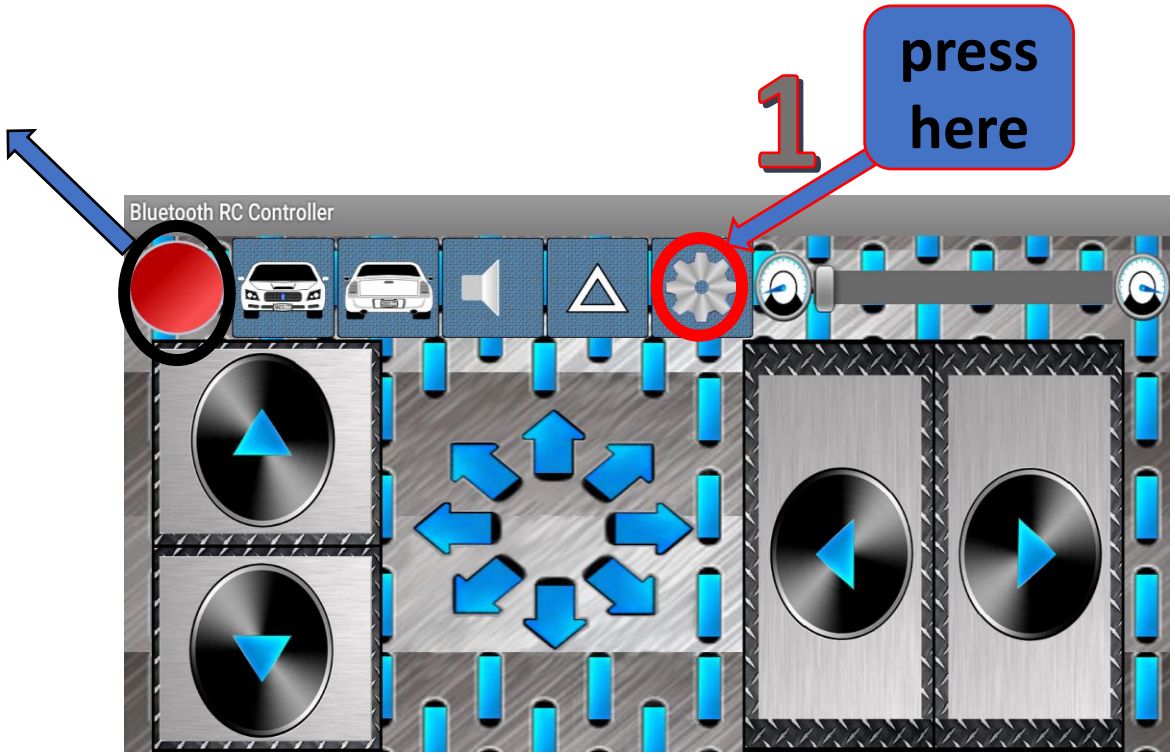
```
void forword()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
void backword()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
```

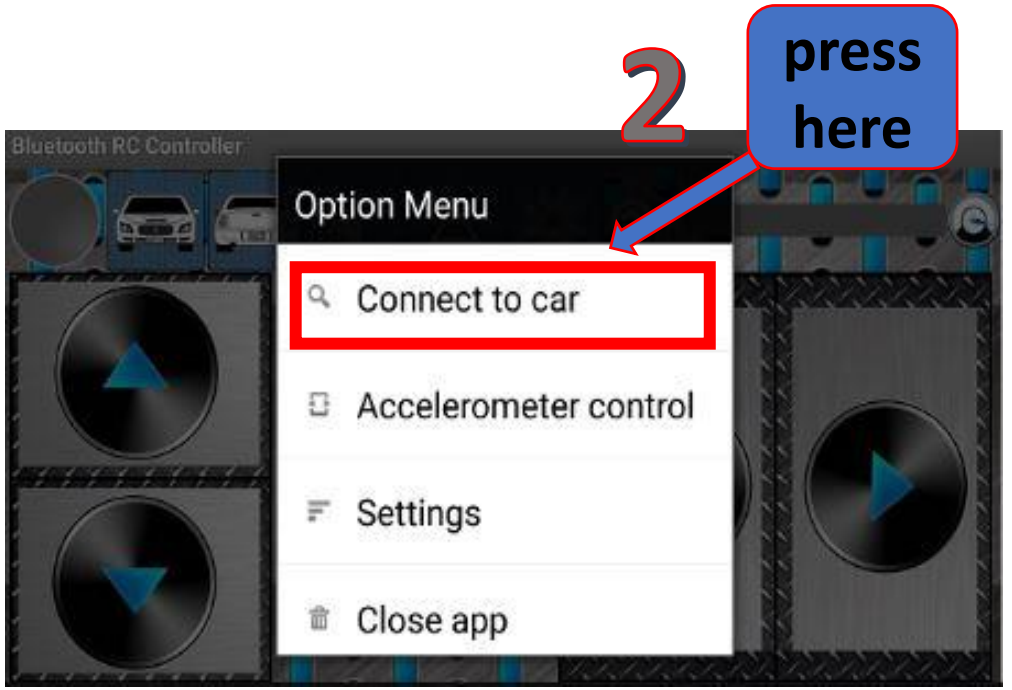
```
void left()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,150);
}
void right()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,0);
}
```

```
void stopp() {
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,0);  }
void loop() {
 if(Serial.available()>0){
  Reading=Serial.read();
   switch(Reading){
      case 'F': forword(); break;
      case 'B': backword(); break;
      case 'R': right(); break;
      case 'L' : left(); break;
      case 'S' :stopp(); break;
    }
  }
}
```

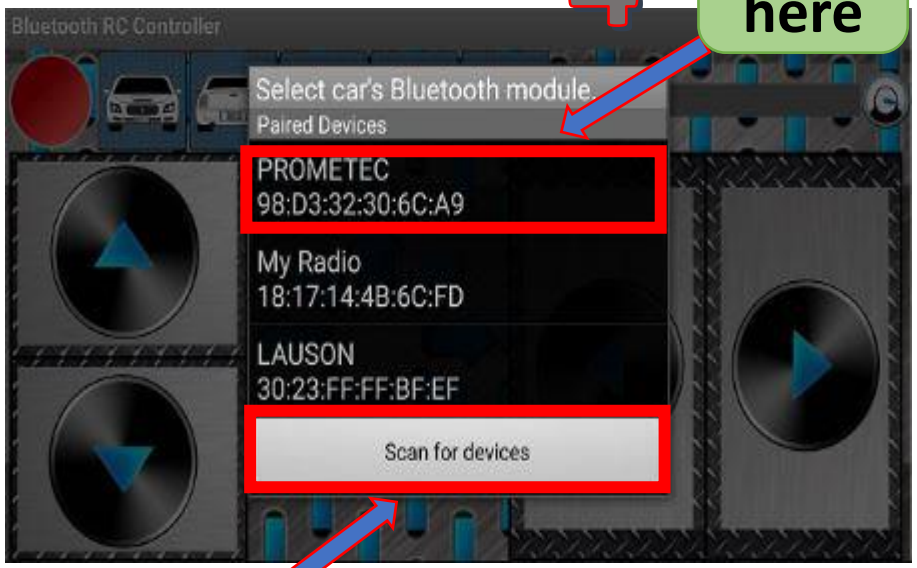
• Bluetooth RC App.

not connect with any bluetooth





• Bluetooth RC App.



4

press here

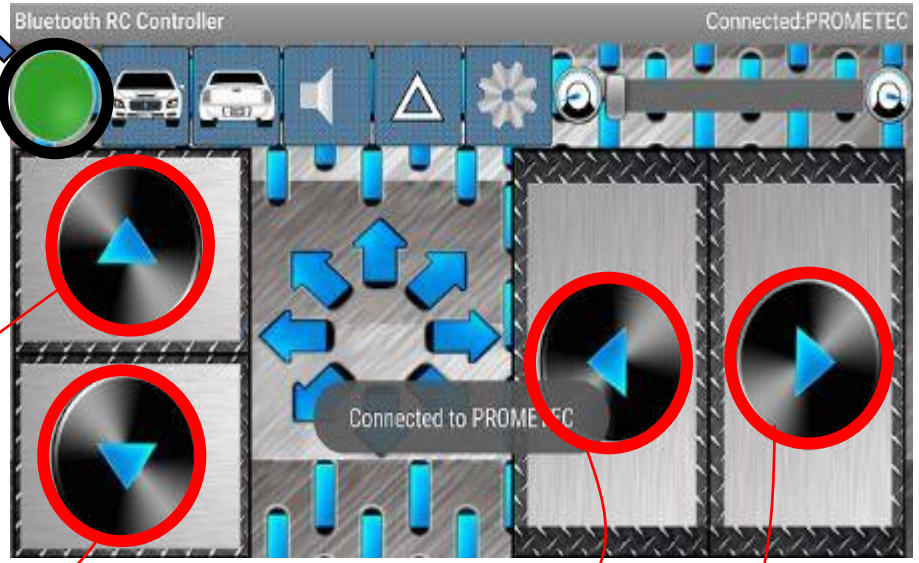
3

press here

Bluetooth is connected

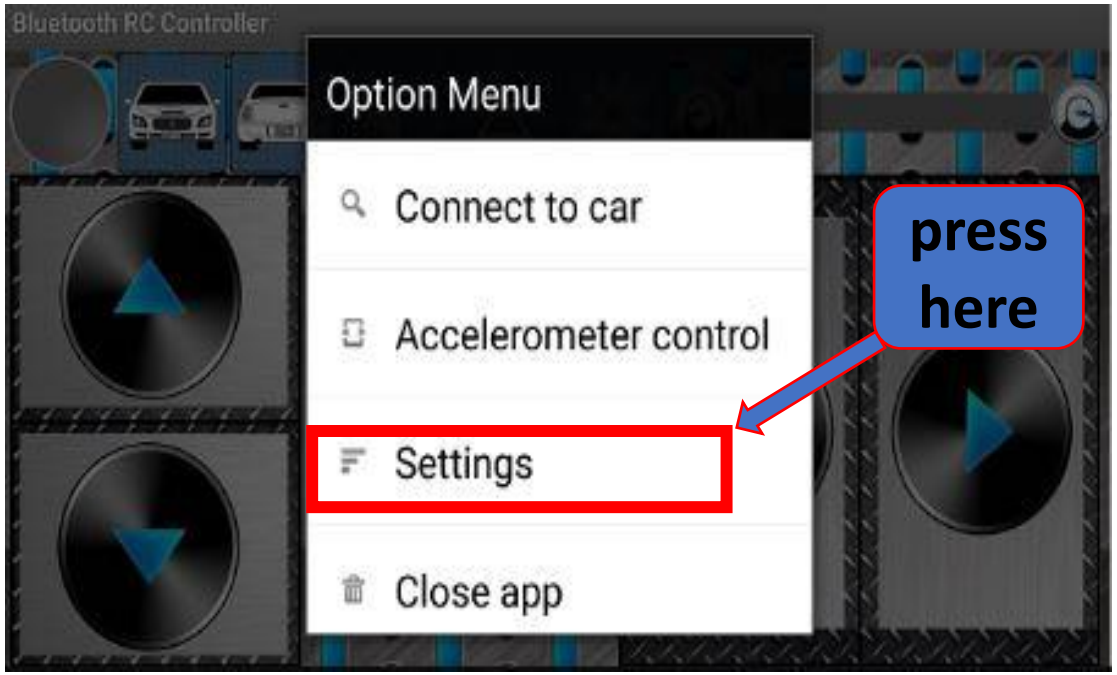
5

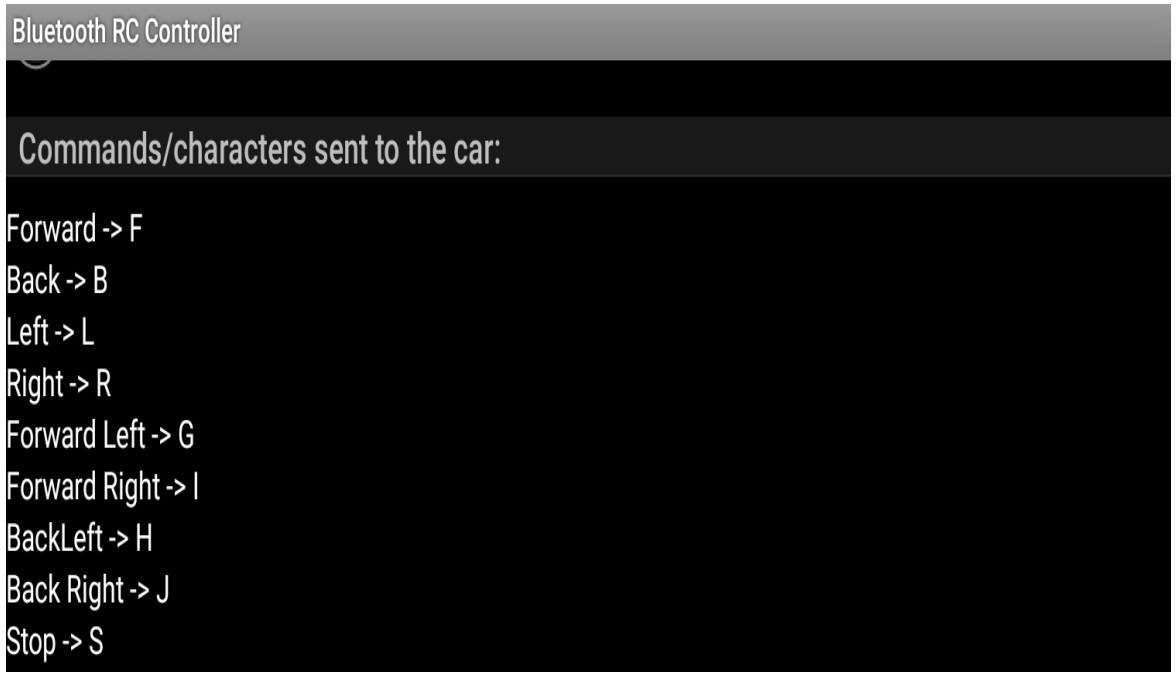
password
0000 -1234



control with these buttons

• Bluetooth RC App.



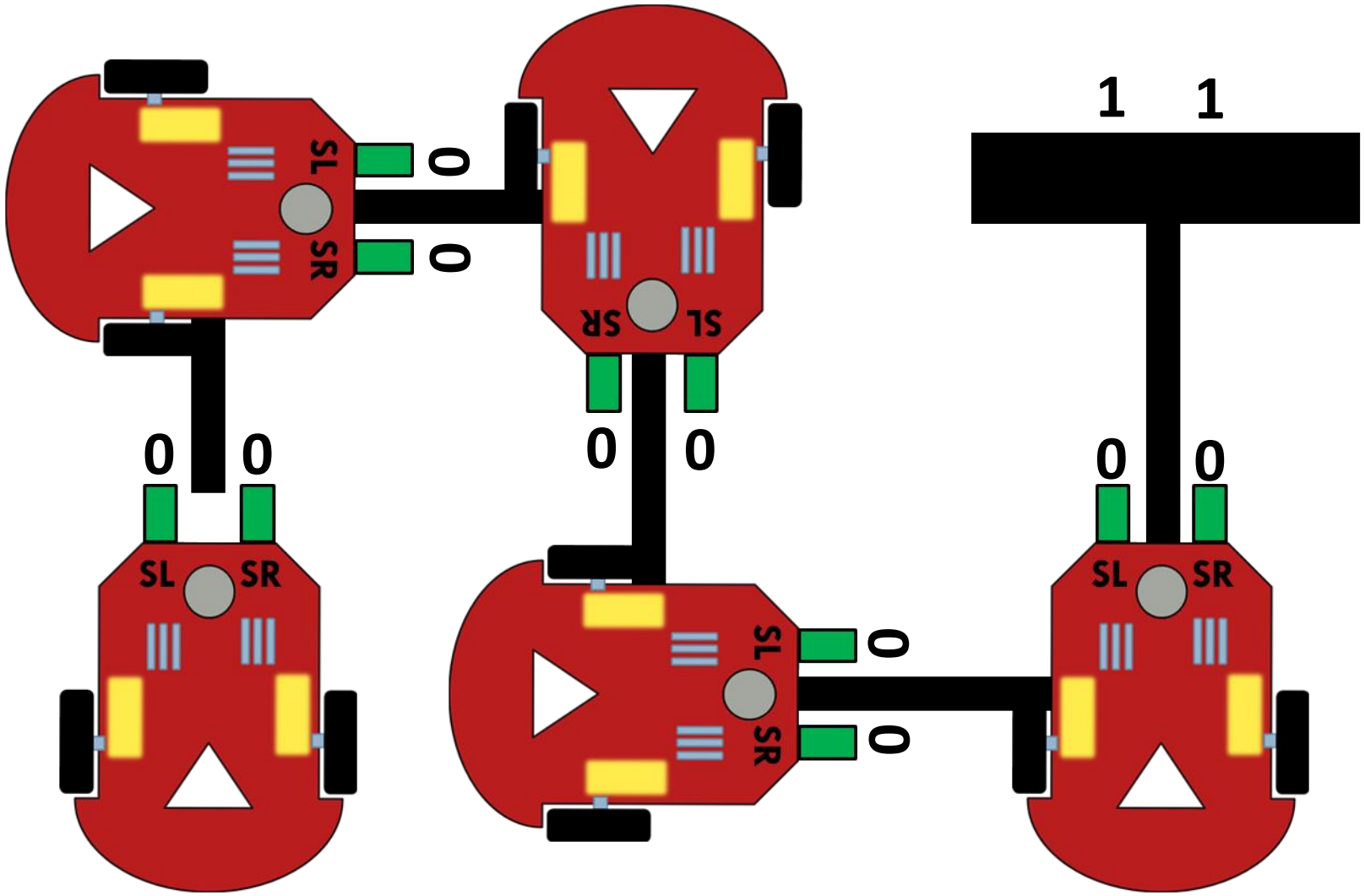


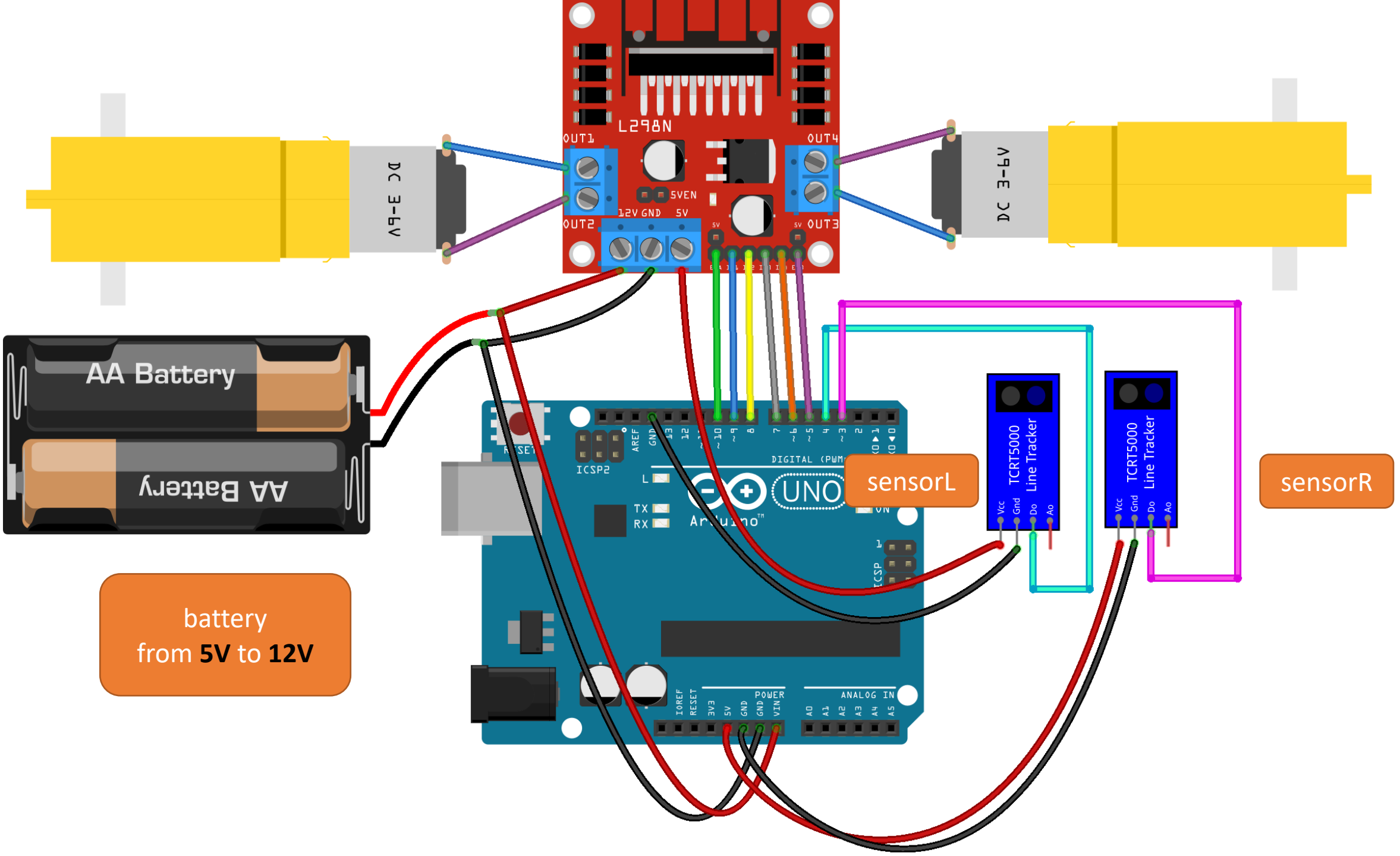
• Line Follower Robot

SL	SR	Direction
		Forword
		Right
		Left
		Stop

$$2^{N} = 2^{2} = 4$$

N : number of sensor




L298N
OUT1
OUT2
OUT3
OUT4
5VEN
12V GND 5V
DC 3-6V
DC 3-6V
AA Battery
AA Battery
UNO
Arduino
sensorL
sensorR
TCRT5000
Line Tracker
Vcc
Gnd
Do
Ao
battery
from 5V to 12V

• Line Follower Code

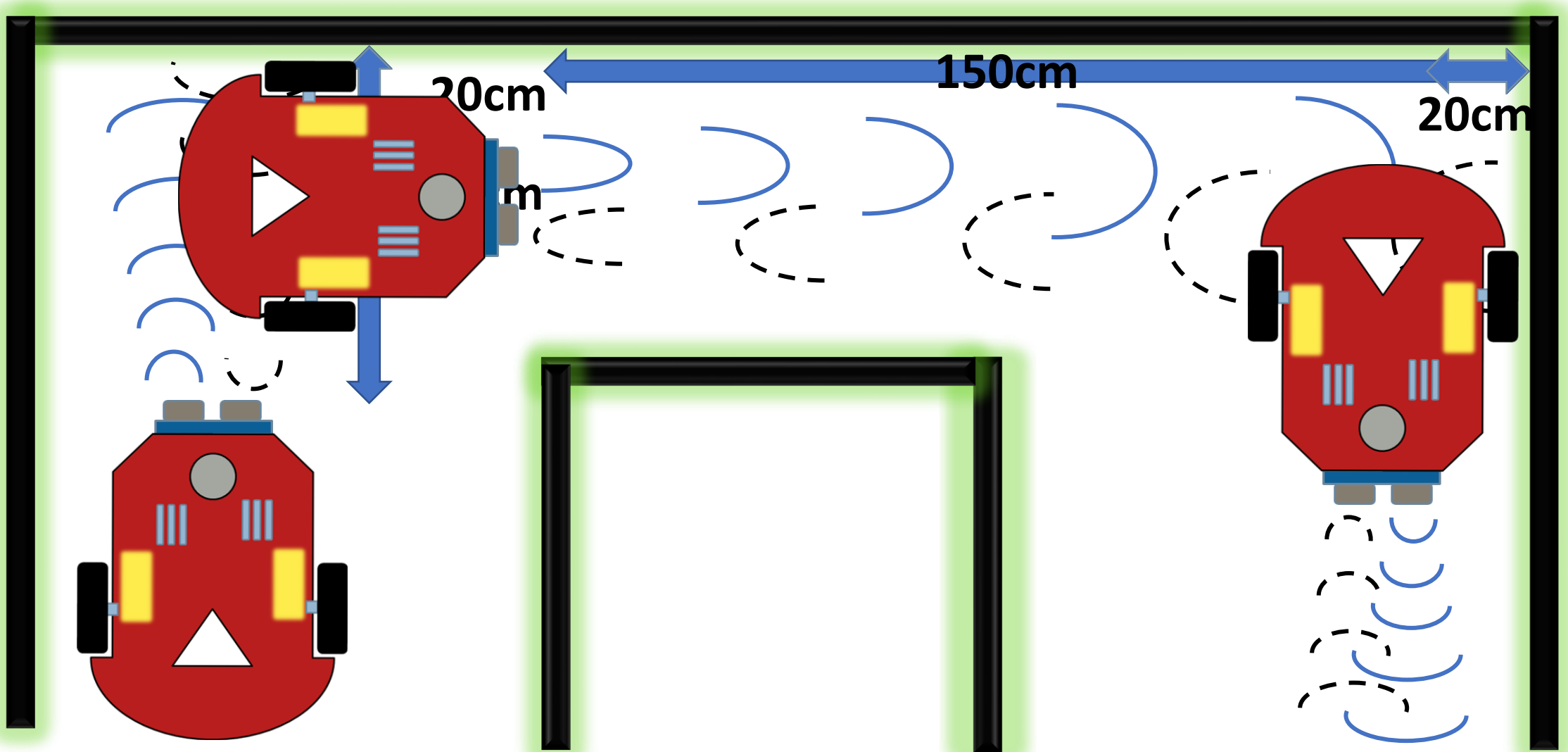
```
#define speedL 10
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
#define sensorL 4
#define sensorR 3
int sl=0;
int sr=0;
void setup() {
for(int i=5;i<=10;i++)
 {
   pinMode(i, OUTPUT);
 }
pinMode(sensorR, INPUT);
pinMode(sensorL, INPUT);
}
```

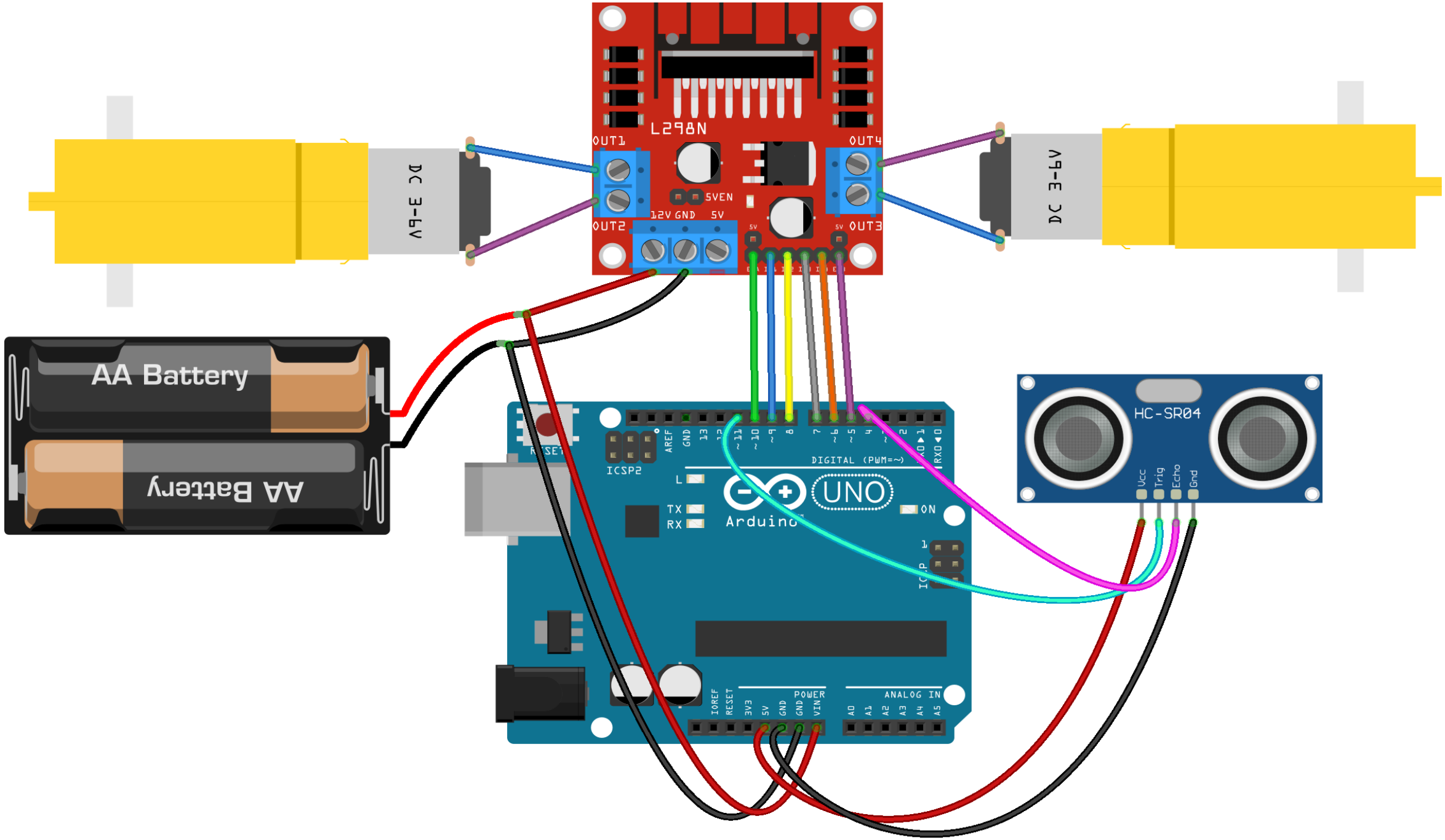
```
void forword()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
void backword()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
```

```
void left()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,150);
}
void right()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,0);
}
```

```
void stopp(){
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,0);
}
void loop(){
sl=digitalRead(sensorL);
sr=digitalRead(sensorR);
if (sl==0&&sr==0)
forword();
else if (sl==0&&sr==1)
right();
else if (sl==1&&sr==0)
left();
else if (sl==1&&sr==1)
stopp();  }
```

• Obstacle avoiding robot




L298N
OUT1
OUT2
OUT3
OUT4
5VEN
12V GND 5V
DC 3-6V
DC 3-6V
AA Battery
AA Battery
HC-SR04
Vcc
Trig
Echo
Gnd
UNO
Arduino
DIGITAL (PWM=~)
POWER
ANALOG IN
ICSP2
TX
RX
ON

• Obstacle Avoiding Code

```
#define speedL 10
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
#define trig 3
#define echo 4
long duration,distance;
void setup() {
for(int i=5 ; i<=11 ; i++)
 {
   pinMode(i, OUTPUT);
 }
pinMode(echo, INPUT);
}
```

```
void Ultrasonic(){
digitalWrite(trig, LOW);
delayMicroseconds(2);
digitalWrite(trig, HIGH);
delayMicroseconds(10);
digitalWrite(trig, LOW);
duration = pulseIn(echo, HIGH);
distance = (duration/2) * 0.0343;
}
void forword()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
```

```
void backword()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
analogWrite(speedL,150);
analogWrite(speedR,150);
}
void left()
{
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(speedL,0);
analogWrite(speedR,150);
}
```

```
void right()
{
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(speedL,150);
analogWrite(speedR,0);
}
void stopp(){
digitalWrite(speedL, LOW);
digitalWrite(speedR, LOW);
}
void loop(){
Ultrasonic();
if(distance<20){
   stopp(); delay(250);
   backword();delay(500);
   right(); delay(1000);
  }
else{ forword(); }
}
```

• Task

- Prepare the codes for experimentation in the next lecture

THANKS FOR COMING

LECTURE

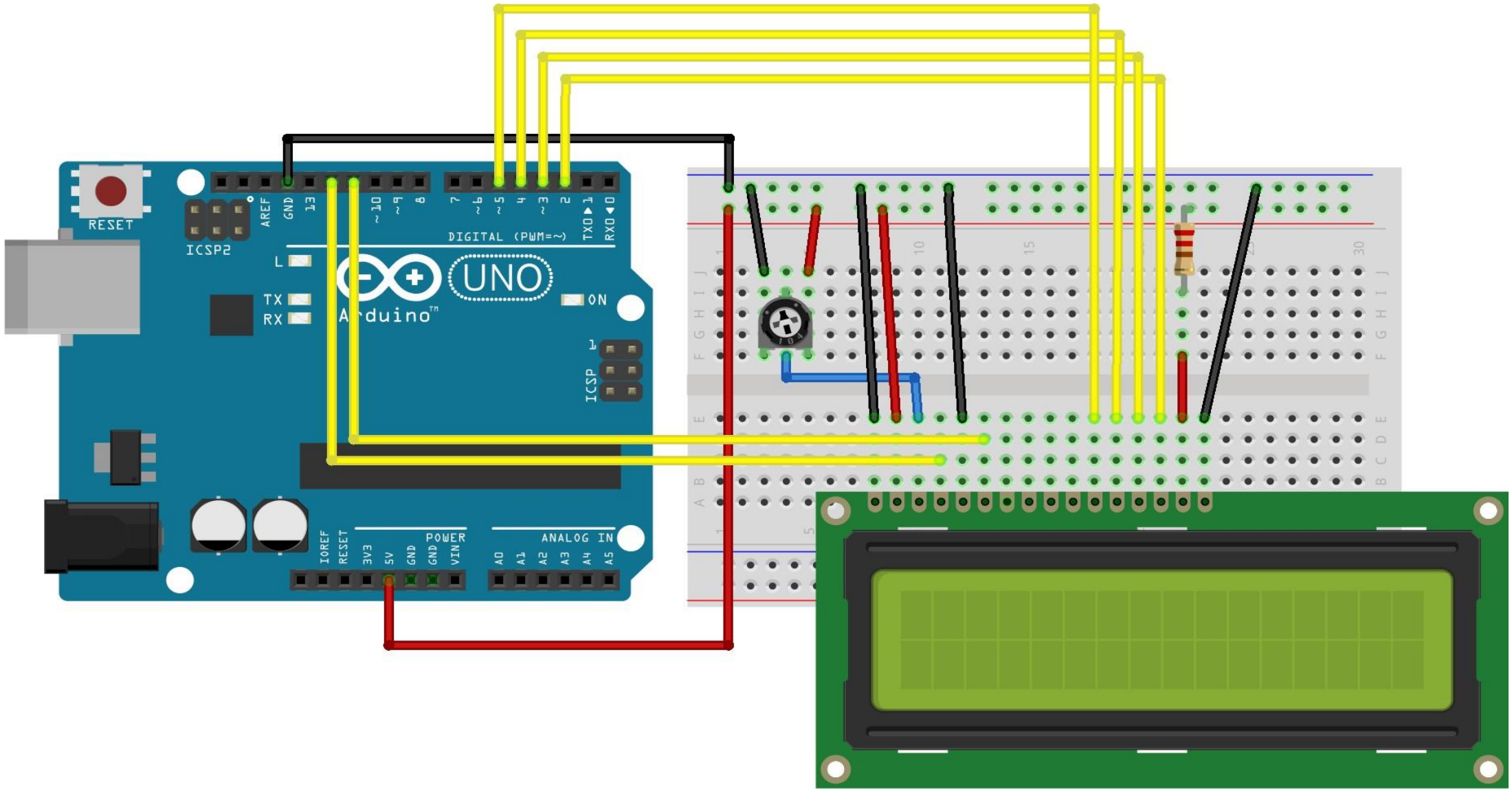
6

(Practical)

THANKS FOR COMING

LECTURE 7

• LCD with Arduino (Wiring)



• Code of LCD display “without I^2C ”

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
lcd.begin(16, 2);
lcd.print(“ST SMART");
lcd.setCursor(7,1);
lcd.print(“2020");
}
void loop() {
  for(int i=0;i<15;i++){
   lcd.scrollDisplayRight();
   delay(200);
   }
   for(int i=0;i<15;i++){
   lcd.scrollDisplayLeft();
   delay(200);
   }}
```

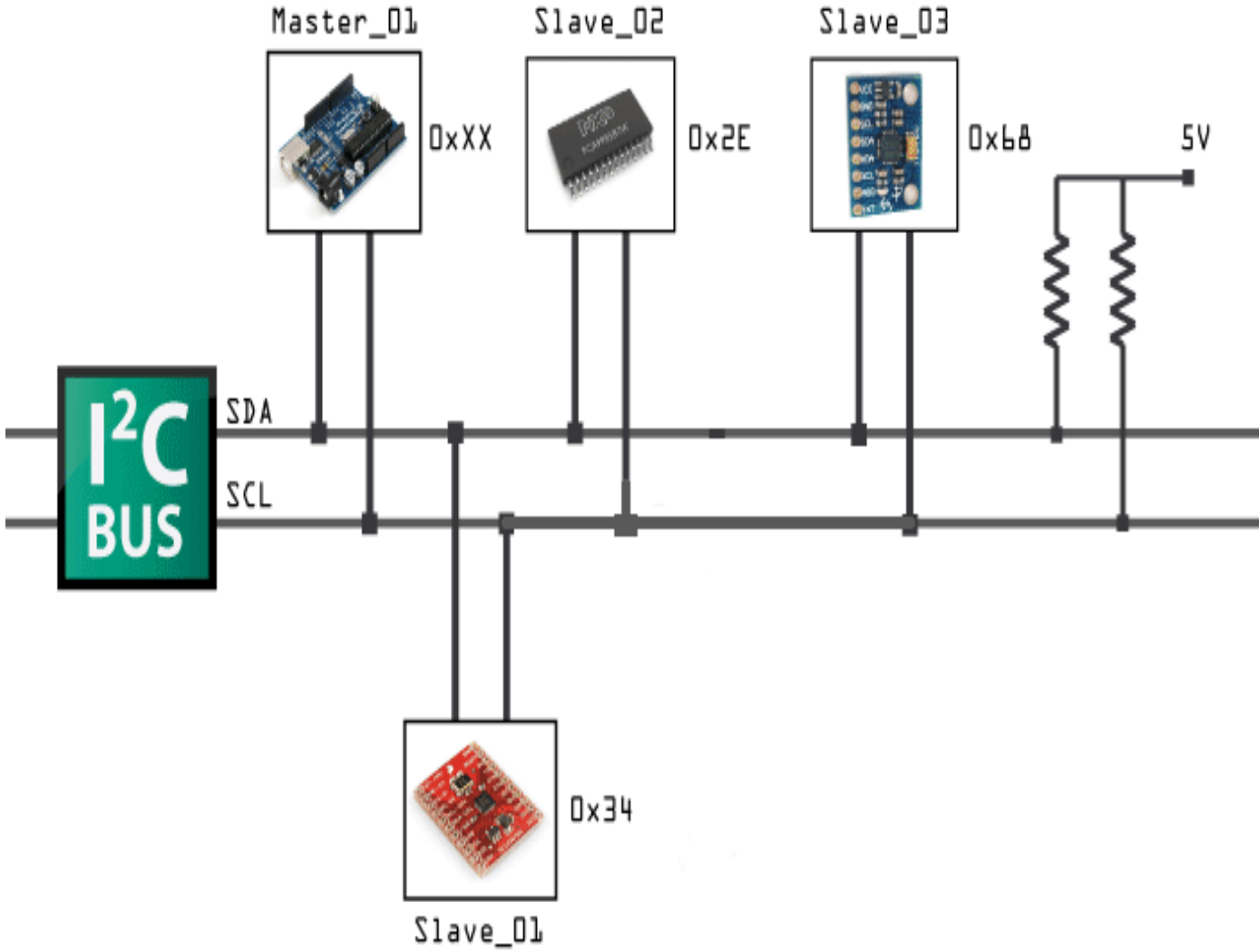
• I²C protocol

- The two wires, or lines are called Serial Clock (or SCL) and Serial Data (or SDA). The SCL line is the clock signal which synchronize the data transfer between the devices on the I2C bus and it's generated by the master device. The other line is the SDA line which carries the data.

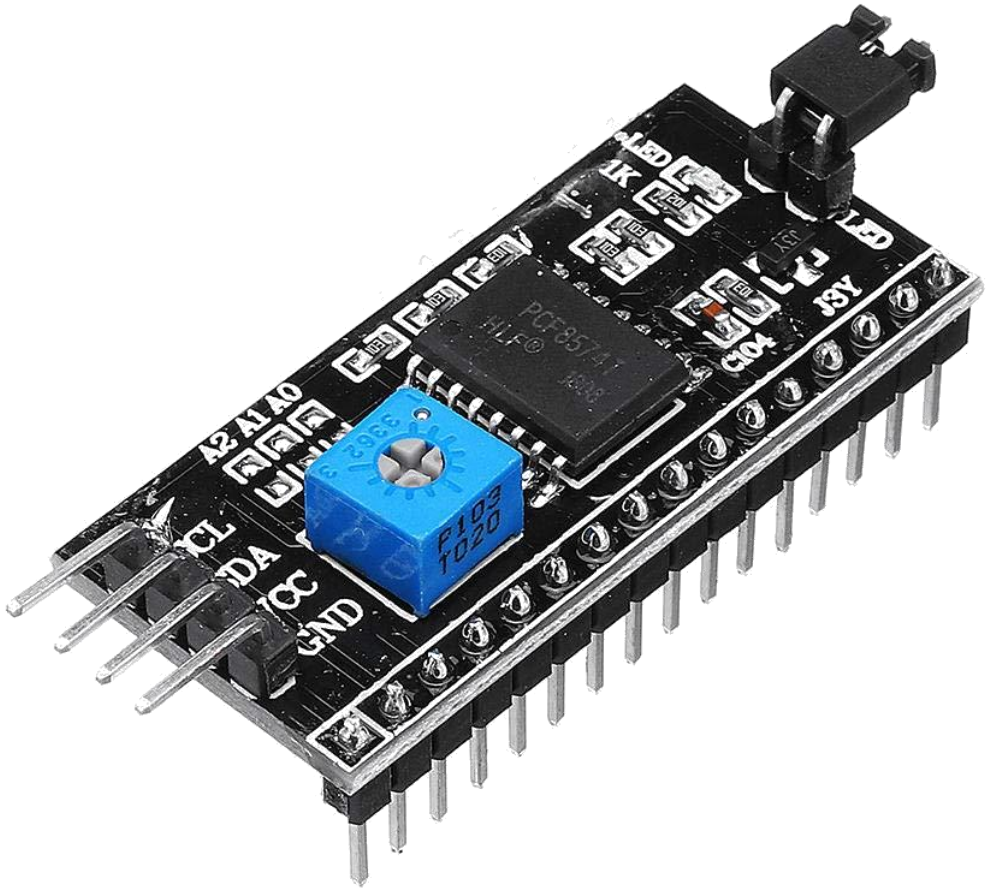
- The two lines are "open-drain" which means that pull up resistors needs to be attached to them so that the lines are high because the devices on the I2C bus are active low. Commonly used values for the resistors are from 2K for higher speeds at about 400 kbps, to 10K for lower speed at about 100 kbps.

- The speed (standard mode: 100 kbit/s,
  full speed:400 kbit/s,
  fast mode: 1 mbit/s,
  high speed: 3,2 Mbit/s)

- Wire Library on Arduino programming language

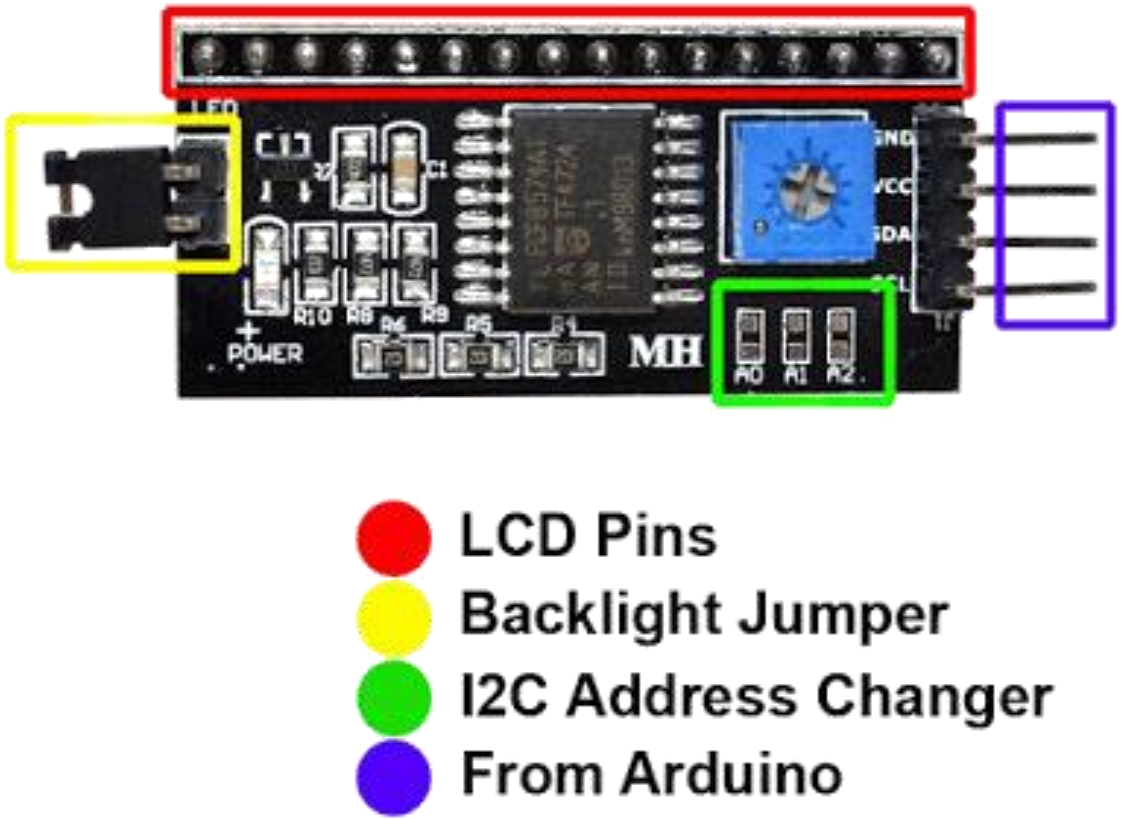


• Why I^2C Module



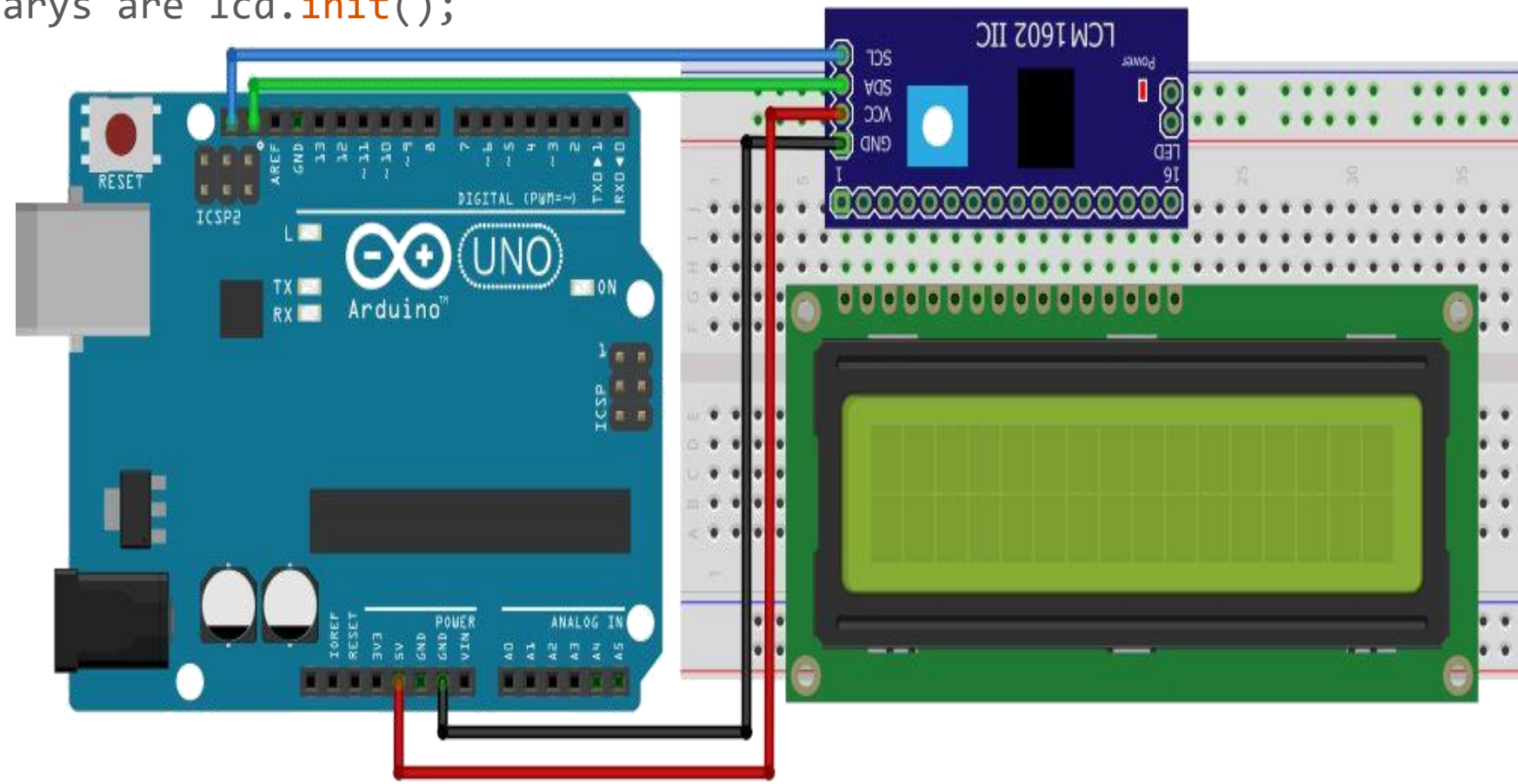
• Address of I^2C Module

A0	A1	A2	Address
Open	Open	Open	0X27
Jumper	Open	Open	0X26
Open	Jumper	Open	0X25
Jumper	Jumper	Open	0X24
Open	Open	Jumper	0X23
Jumper	Open	Jumper	0X22
Open	Jumper	Jumper	0X21
Jumper	Jumper	Jumper	0X20



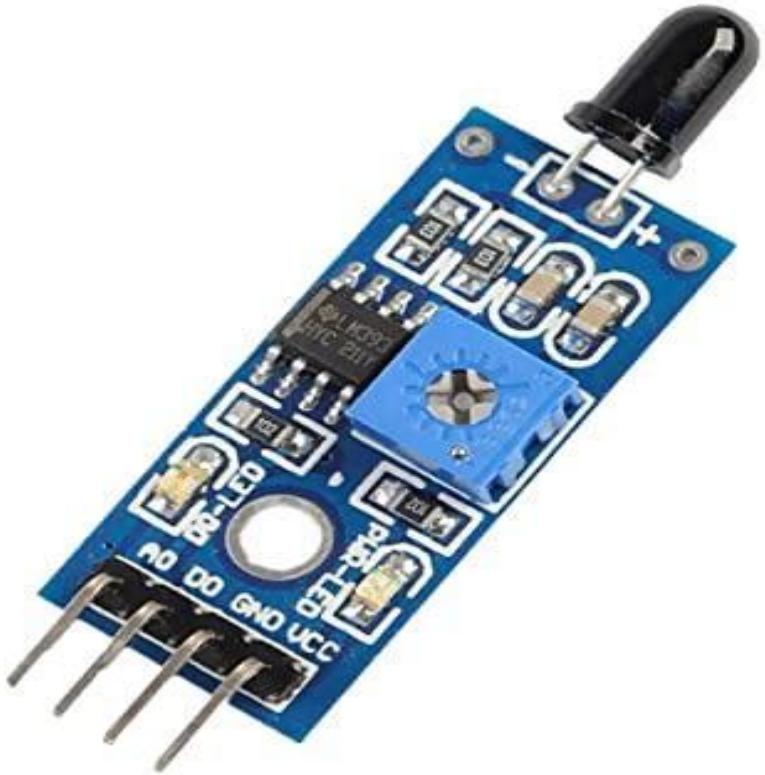
• Code of LCD display with I2C

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
lcd.begin(); // or some librarys are lcd.init();
lcd.backlight();
lcd.print("ST SMART");
lcd.setCursor(7,1);
lcd.print("2023");
}
void loop() {
  for(int i=0;i<15;i++){
   lcd.scrollDisplayRight();
   delay(200);   }

   for(int i=0;i<15;i++){
   lcd.scrollDisplayLeft();
   delay(200);   }
}
```

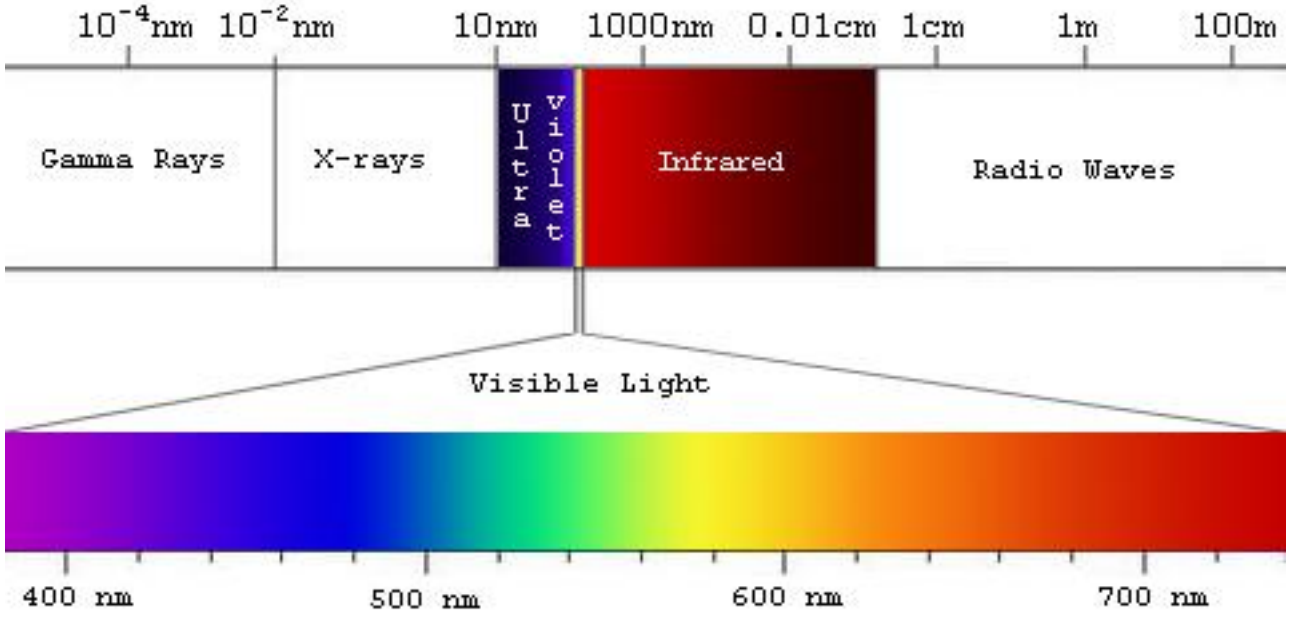


• Flame sensor

- Detects a flame or a light source of a wavelength in the range of 760nm-1100 nm.
- Detection range: up to 80 cm.
- Operating temp: -25° - 85°
- Adjustable detection range.
- Detection angle about 60 degrees, it is sensitive to the flame spectrum.
- Operating voltage 3.3V-5V.
- Digital and Analog Output.



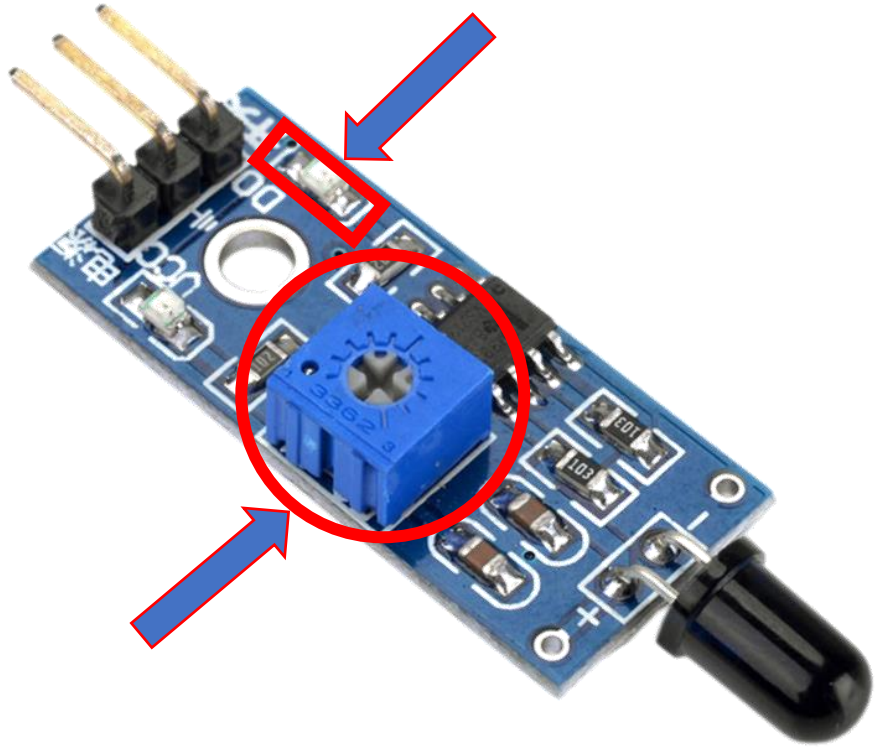
• Basic principle



principle that the infrared ray is sensitive to flame. It uses the photo diode to detect flame and converts the brightness of flame to the variable level signal.

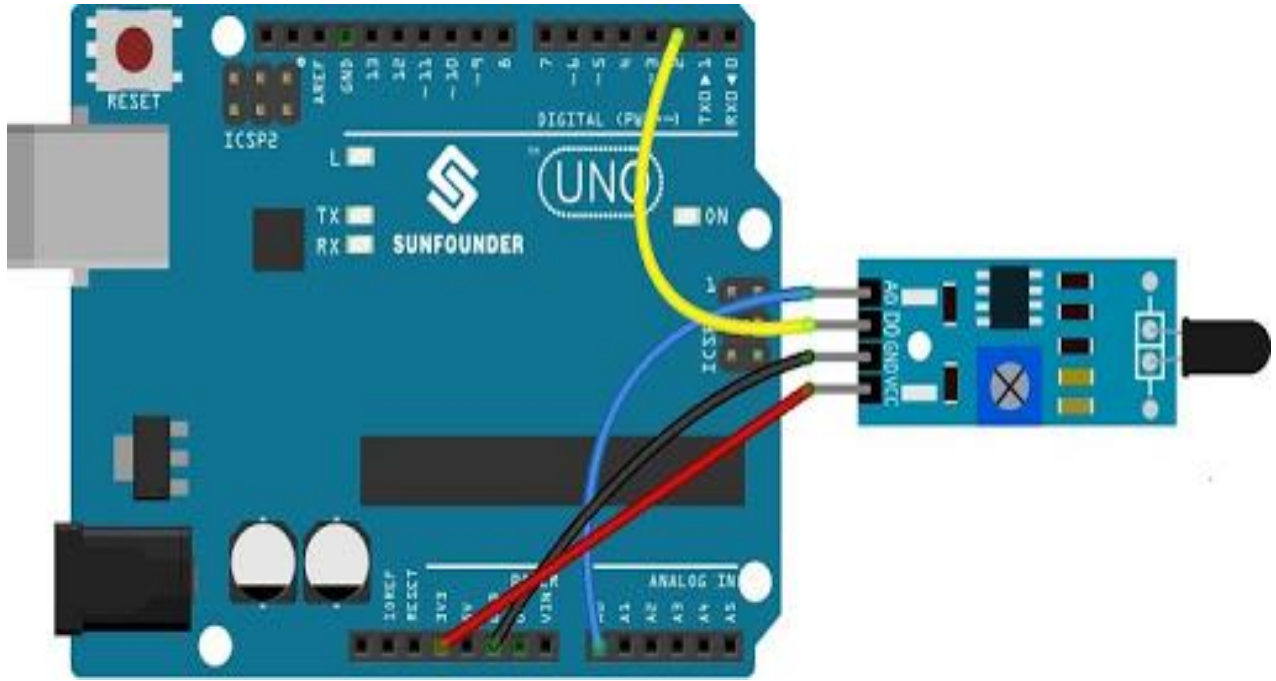
• Calibrate Flame Sensor Module

Expose the module to flame or strong light and turn the knob of the potentiometer gently till the D0 indicator light is on



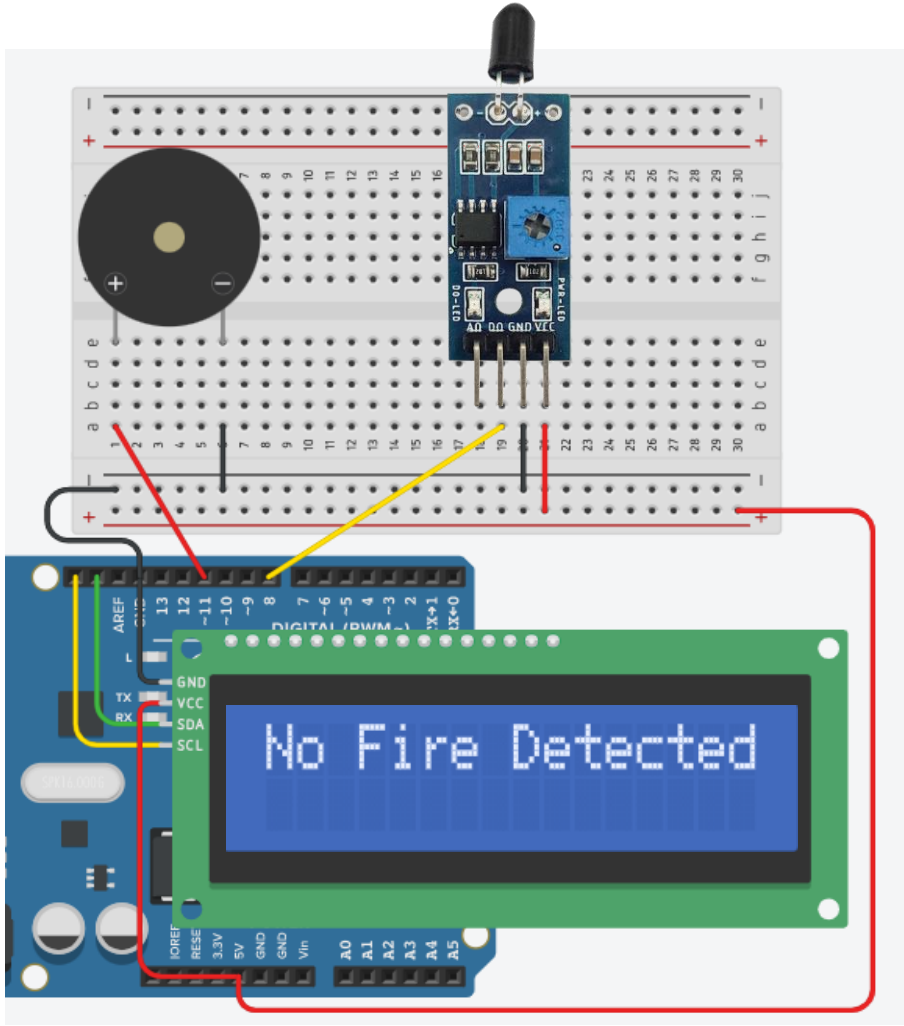
• Code

```
#define flame 3
#define buzzer 5
int Val = 0;
void setup()
{ Serial.begin(9600);
  pinMode(flame , INPUT);
  pinMode(buzzer,OUTPUT);
}
void loop()
{ Val = digitalRead(flame);
  Serial.print("value of flame:");
  Serial.println(Val);
if (Val == LOW) {  digitalWrite(buzzer, HIGH); }
  else {digitalWrite(buzzer, LOW);}
}
```



• Flame With LCD & Buzzer

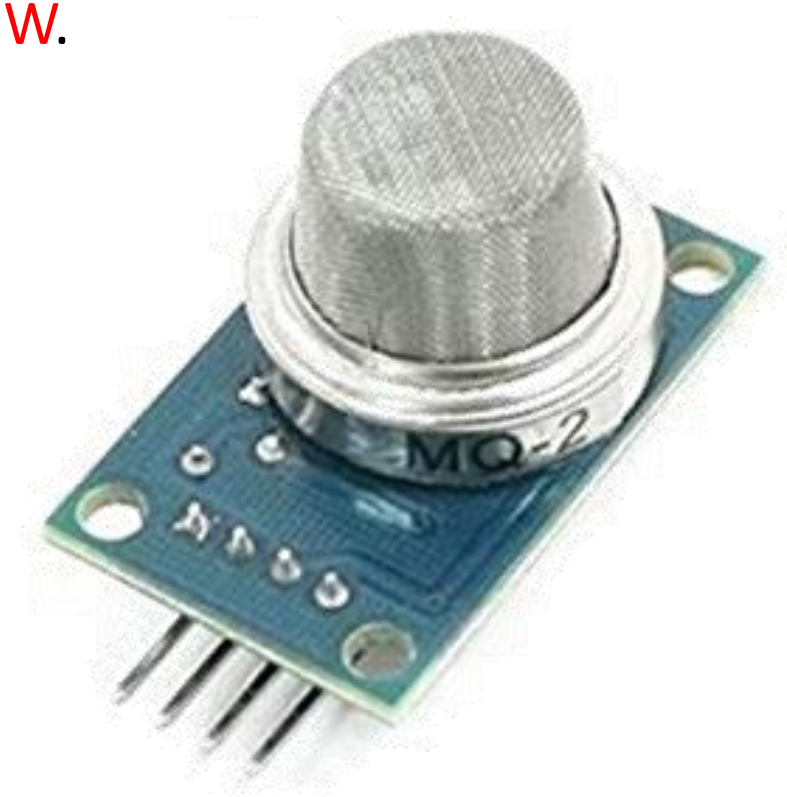
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{  lcd.init();
   lcd.backlight();
  pinMode(8,INPUT);
  pinMode(11,OUTPUT);
}
void loop()
{  if (digitalRead(8)==0){
       lcd.setCursor(0,0);
       lcd.print("Fire Detected");
       digitalWrite(11,1);
       delay(100);
       digitalWrite(11,0);
       delay(100);}
  else {
    lcd.setCursor(0,0);
    lcd.print("NO Fire Detected");
    digitalWrite(11,0);}
}
```



• Smoke sensor

- MQ2 Gas sensor works on 5V DC and draws around 800mW.
- It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Butane, Methane and Carbon Monoxide concentrations anywhere from 200 to 10000ppm

Parts-per-million (abbreviated ppm) is the ratio of one gas to another. For example, 1,000ppm of CO means that if you could count a million gas molecules, 1,000 of them would be of carbon monoxide and 999,000 molecules would be some other gases.



- preheat time : NOT less than 48 hours

• Basic Principle

How do MOS type gas sensors detect gas?

Basic principle

When tin dioxide (semiconductor particles) is heated in air at high temperature, oxygen is adsorbed on the surface. In clean air, donor electrons in tin dioxide are attracted toward oxygen which is adsorbed on the surface of the sensing material. This prevents electric current flow.

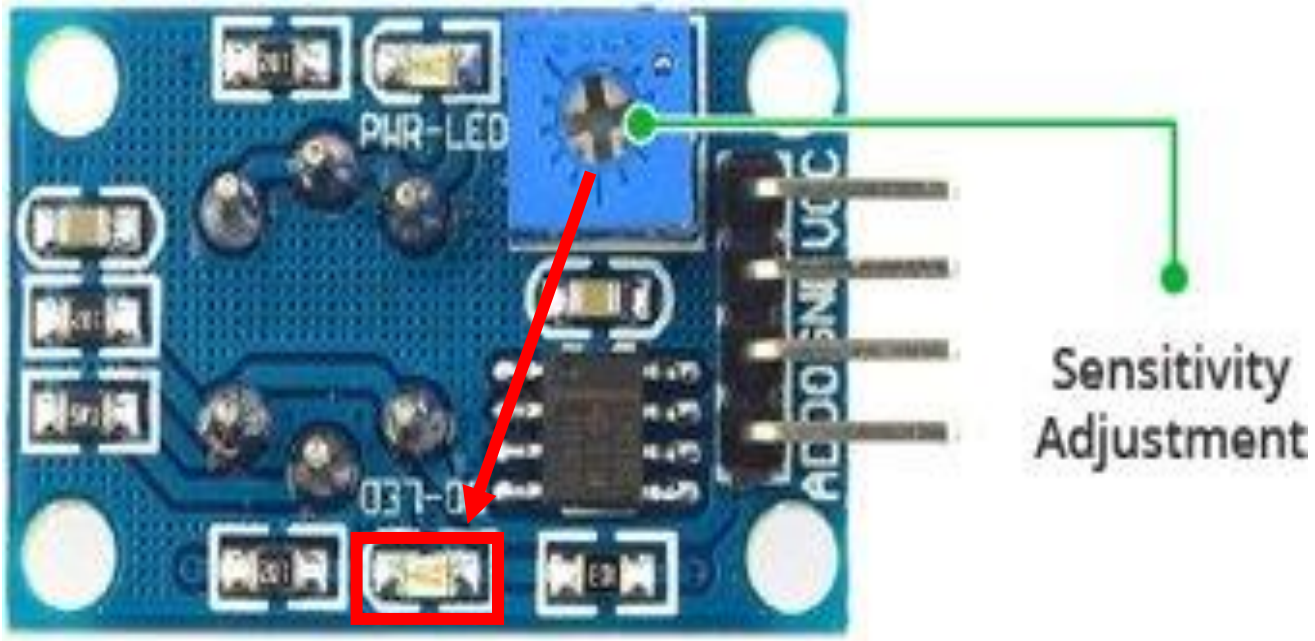
In the presence of reducing gases, the surface density of adsorbed oxygen decreases as it reacts with the reducing gases. Electrons are then released into the tin dioxide, allowing current to flow freely through the sensor.

• Basic principle

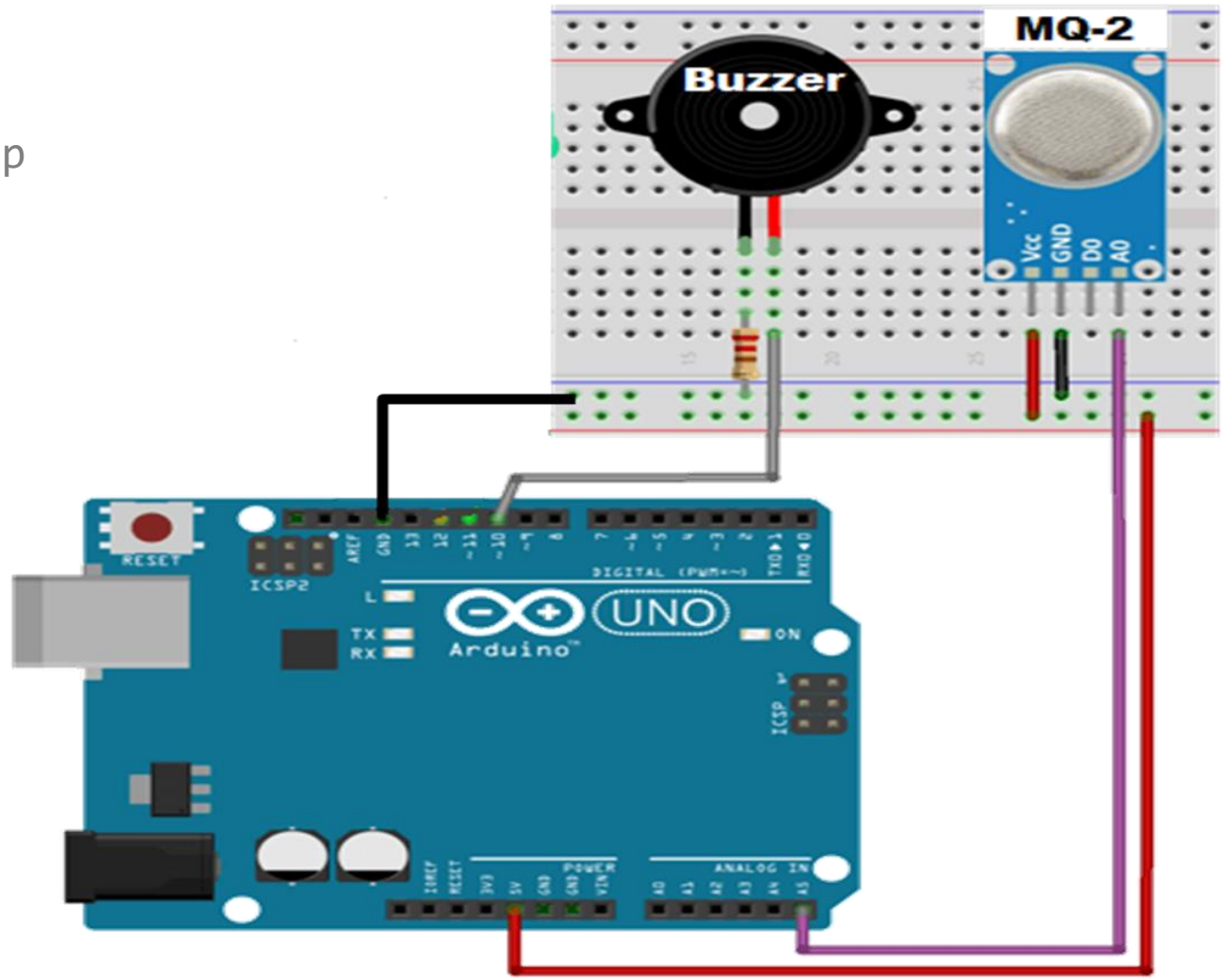
- The analog output voltage provided by the sensor changes in proportional to the concentration of smoke/gas. The greater the gas concentration, the higher is the output voltage; while lesser gas concentration results in low output voltage. The following animation illustrates the relationship between gas concentration and output voltage.

• Calibrate MQ2 Sensor

- To calibrate the gas sensor you can hold the gas sensor near smoke/gas you want to detect and keep turning the potentiometer until the Red LED on the module starts glowing. Turn the screw clockwise to increase sensitivity or anticlockwise to decrease sensitivity.

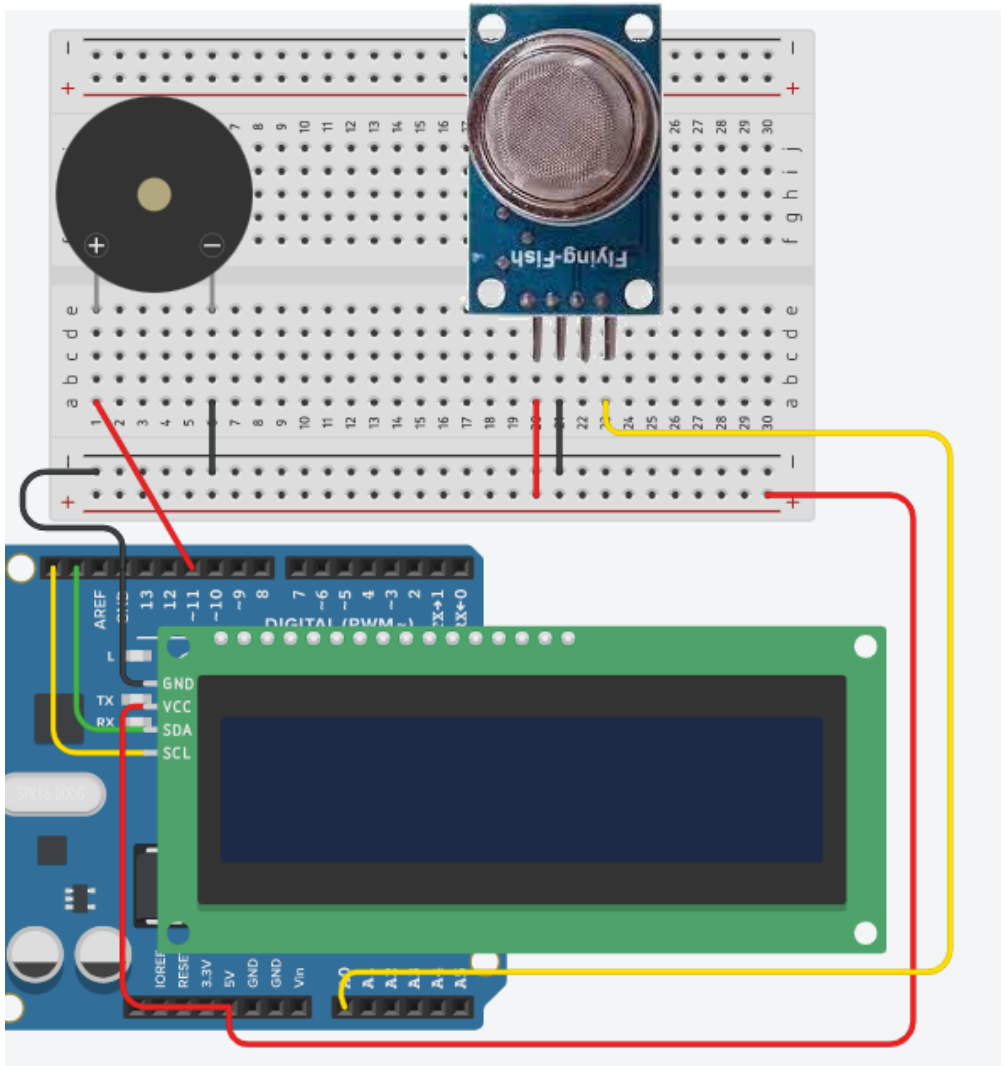


```
#define MQ2pin A5
#define buzzer 10
int sensorValue;
void setup()
{
  Serial.begin(9600);
  pinMode(buzzer,OUTPUT);
  Serial.println("Gas sensor warming up!");
  delay(20 000); // allow the MQ-2 to warm up
}

void loop()
{
  sensorValue = analogRead(MQ2pin);
  Serial.print("Sensor Value: ");
  Serial.println(sensorValue);
  if(sensorValue > 300){
    Serial.print(" | Smoke detected!");
   digitalWrite(buzzer,HIGH);
  }
  else {  digitalWrite(buzzer,LOW); }
  delay(2000); // wait 2s for next reading
}
```



• Smoke sensor with LCD & Buzzer

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  lcd.begin();
  lcd.backlight();
  lcd.print("MQ-2 worming"); delay(20 000);
}

void loop()
{
      lcd.setCursor(0,0);
      lcd.print("Smoke intensty");
      lcd.setCursor(0,1);
      lcd.print(analogRead(A0));
}
```



• DHT11

- DHT11 digital temperature & humidity sensor module

vcc : pin supplies power for the sensor. 5V supply is recommended, although the supply voltage ranges from 3.3V to 5.5V. In case of 5V power supply, you can keep the sensor as long as 20 meters. However, with 3.3V supply voltage, cable length shall not be greater than 1 meter. Otherwise, the line voltage drop will lead to errors in measurement.

OUT : pin is used to communication between the sensor and the Arduino.

GND : should be connected to the ground of Arduino.

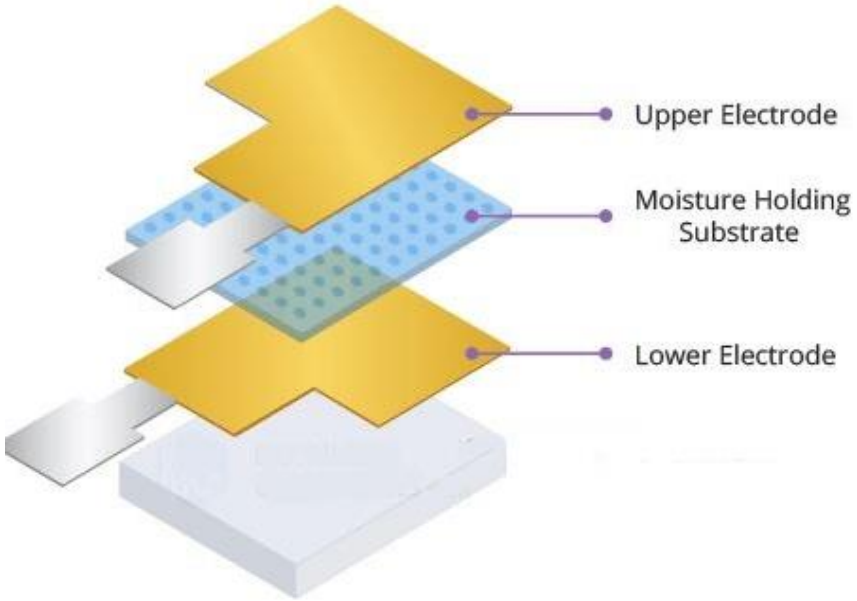
• Basic Principle

- Humidity sensing component has two electrodes with moisture holding substrate sandwiched between them.

- The ions are released by the substrate as water vapor is absorbed by it, which in turn increases the conductivity between the electrodes.

- The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

- DHT11 also contains a NTC/Thermistor to measure temperature. A thermistor is a thermal resistor whose resistance changes drastically with temperature. The term “NTC” means “Negative Temperature Coefficient”, which means that the resistance decreases with increase of the temperature.



• DHT 11

Temperature Range	0°C to 50°C ±2.0°C
Humidity Range	20% to 80% ±5%
Operating Voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby)
Sampling period	1 Second

Vcc OUT GND

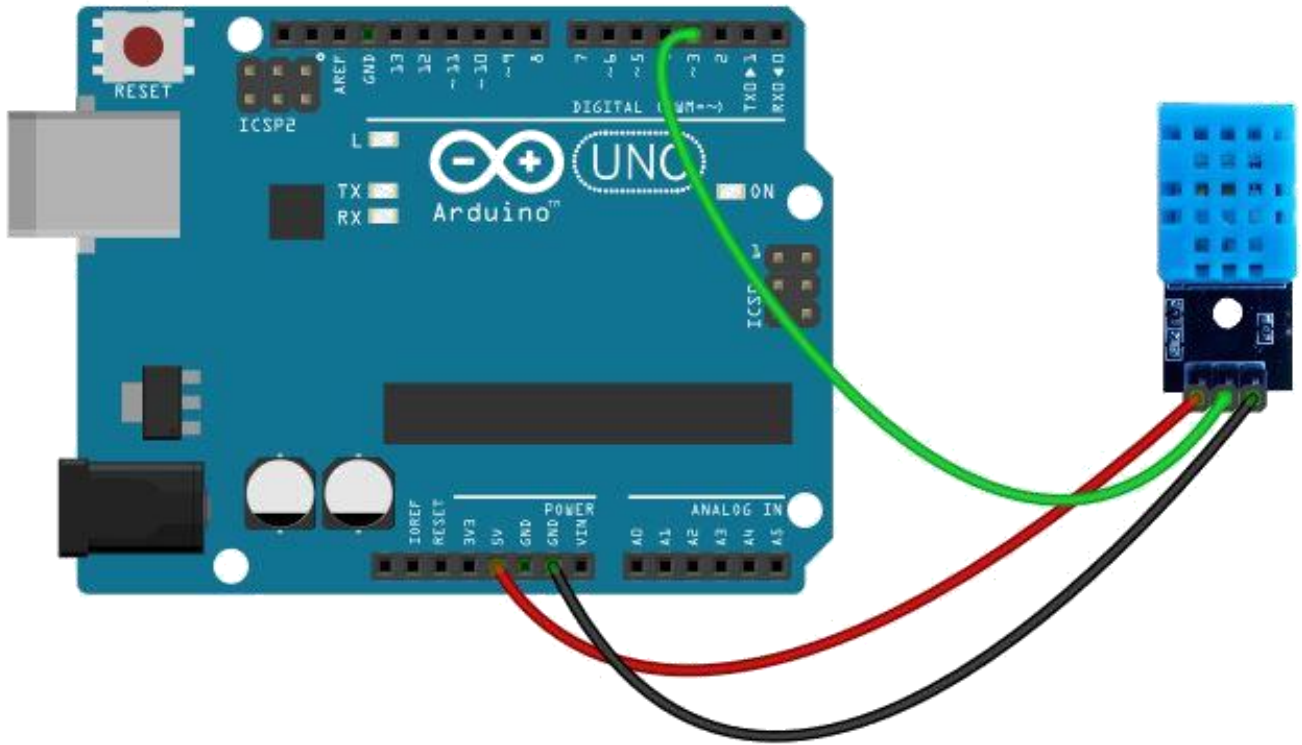
• DHT 11

```
#include "DHT.h"
#define DHTPIN 3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print("%   Temperature: ");
  Serial.print(t);
  Serial.println("°C ");
}
```

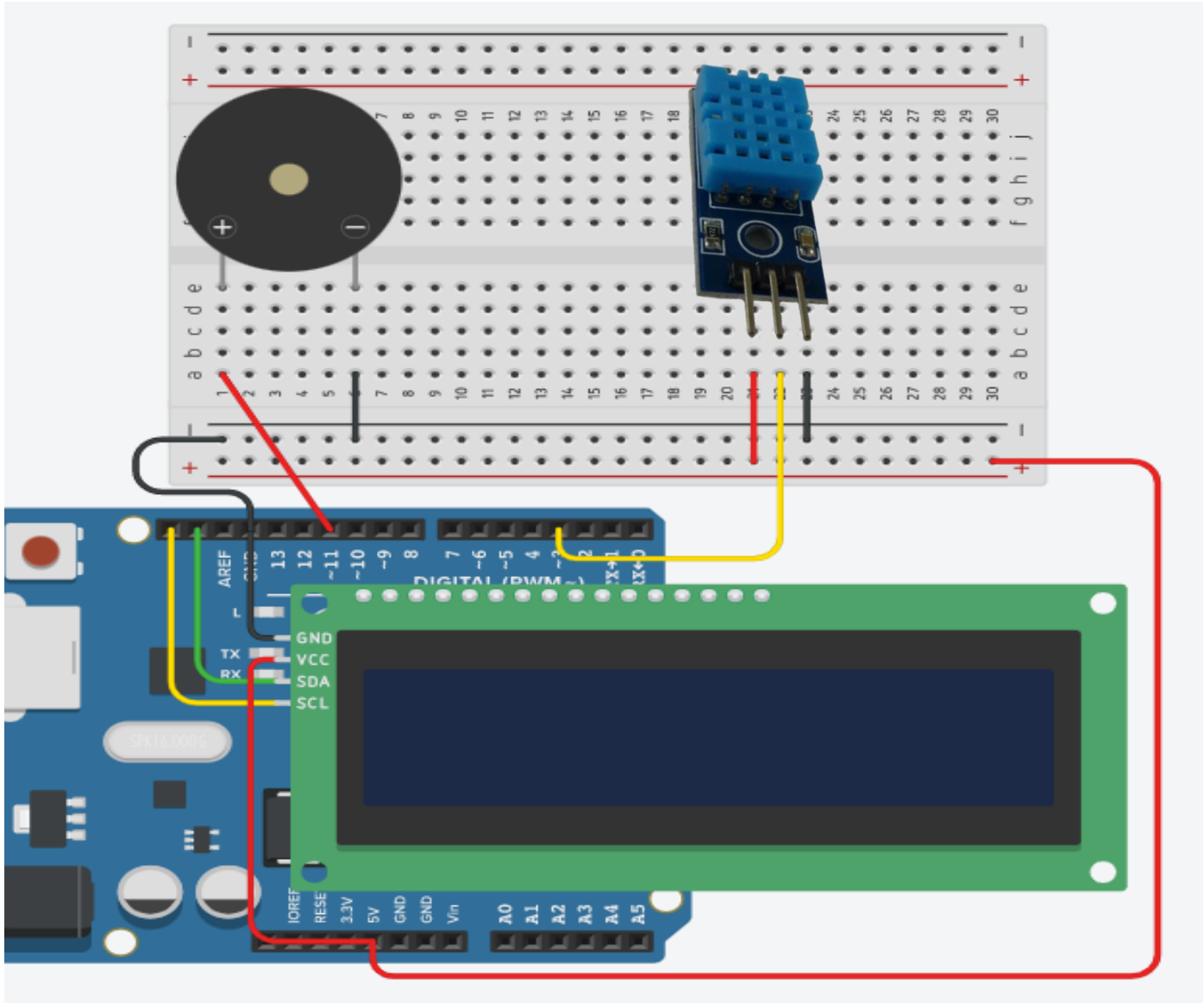
REQUIRES the following Arduino libraries:

1-DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

2-Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit_Sensor



• DHT With LCD & Buzzer



• DHT With LCD & Buzzer

```
#define DHTPIN 3
#define DHTTYPE DHT11
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  lcd.begin();
   lcd.backlight();
    dht.begin();
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
    lcd.setCursor(0,0);
    lcd.print("Temp=");
    lcd.setCursor(5,0);
    lcd.print(t);

    lcd.setCursor(1,0);
    lcd.print("Humidity=");
    lcd.setCursor(9,1);
    lcd.print(h);
}
```

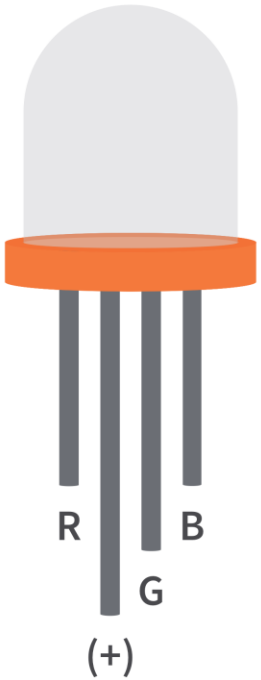
• Task

- Write a code that’s combine (DHT11 + Smoke + Flame) with LCD

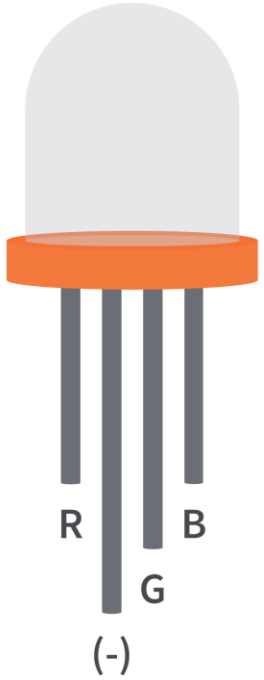
THANKS FOR COMING

LECTURE 8

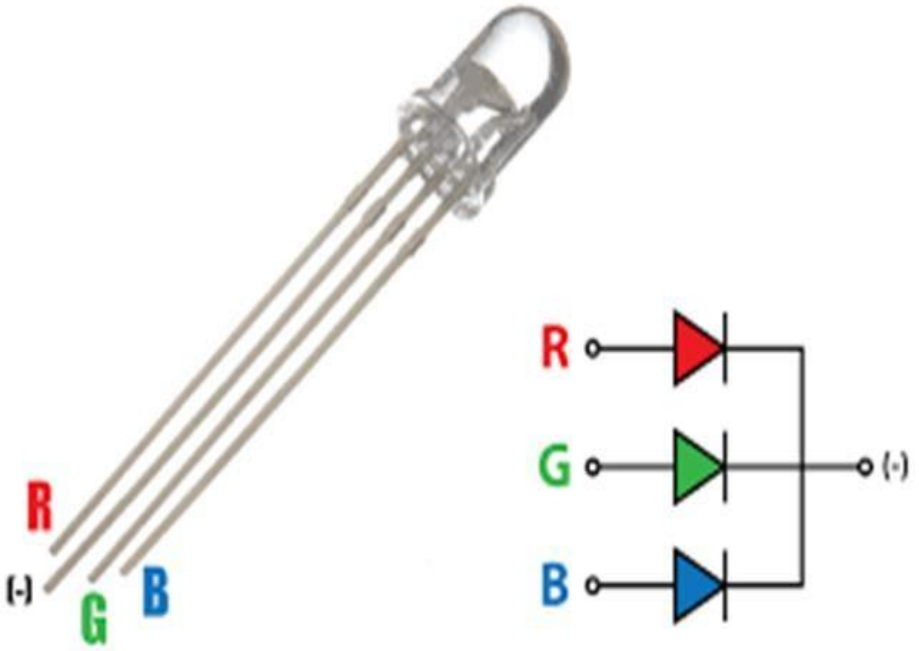
• RGB Led



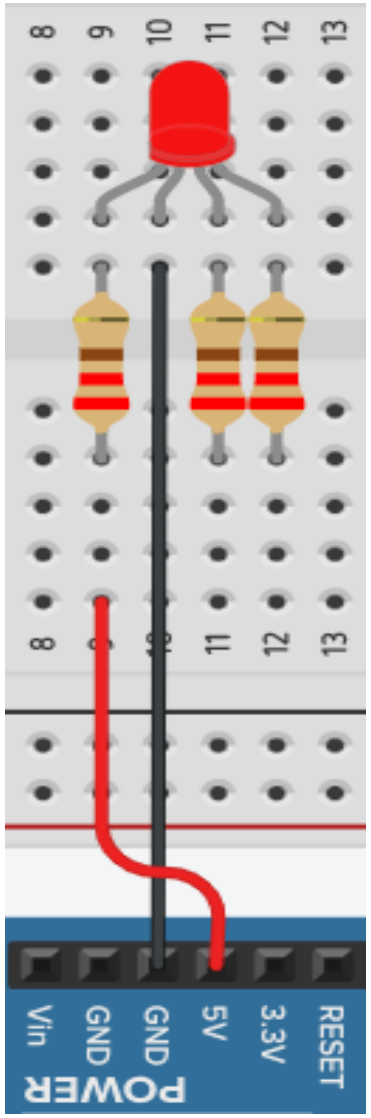
Common Anode

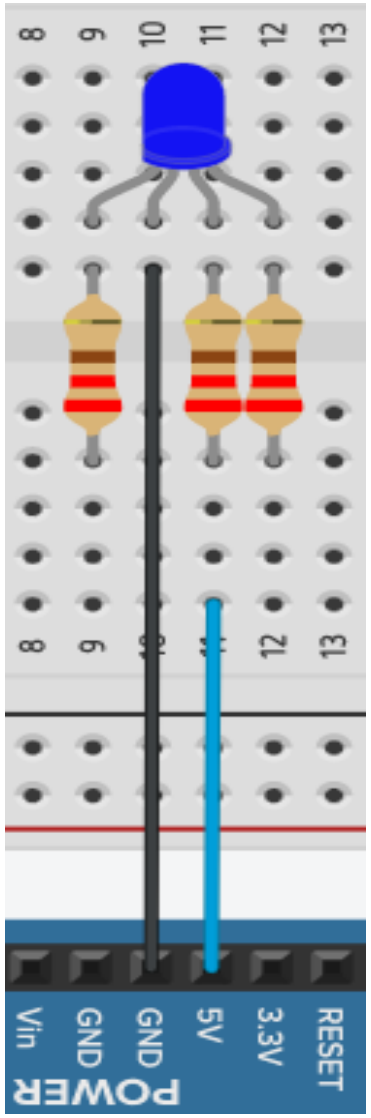


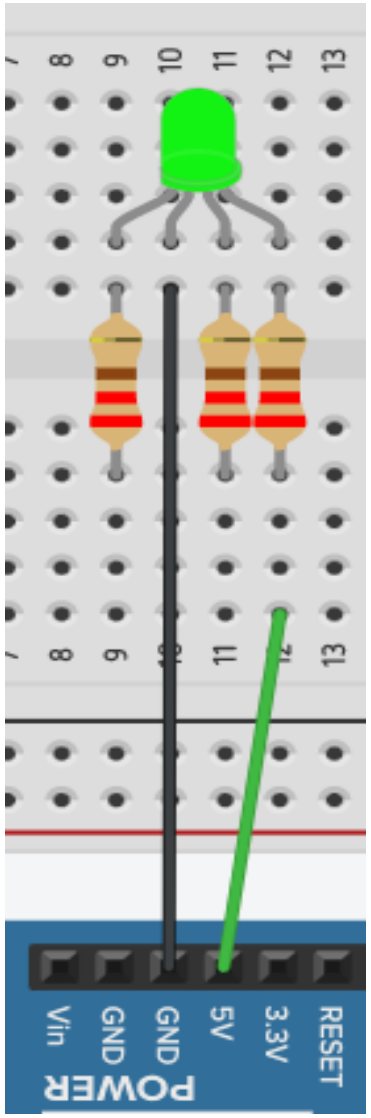
Common Cathode



• Cathode RGB Led Testing







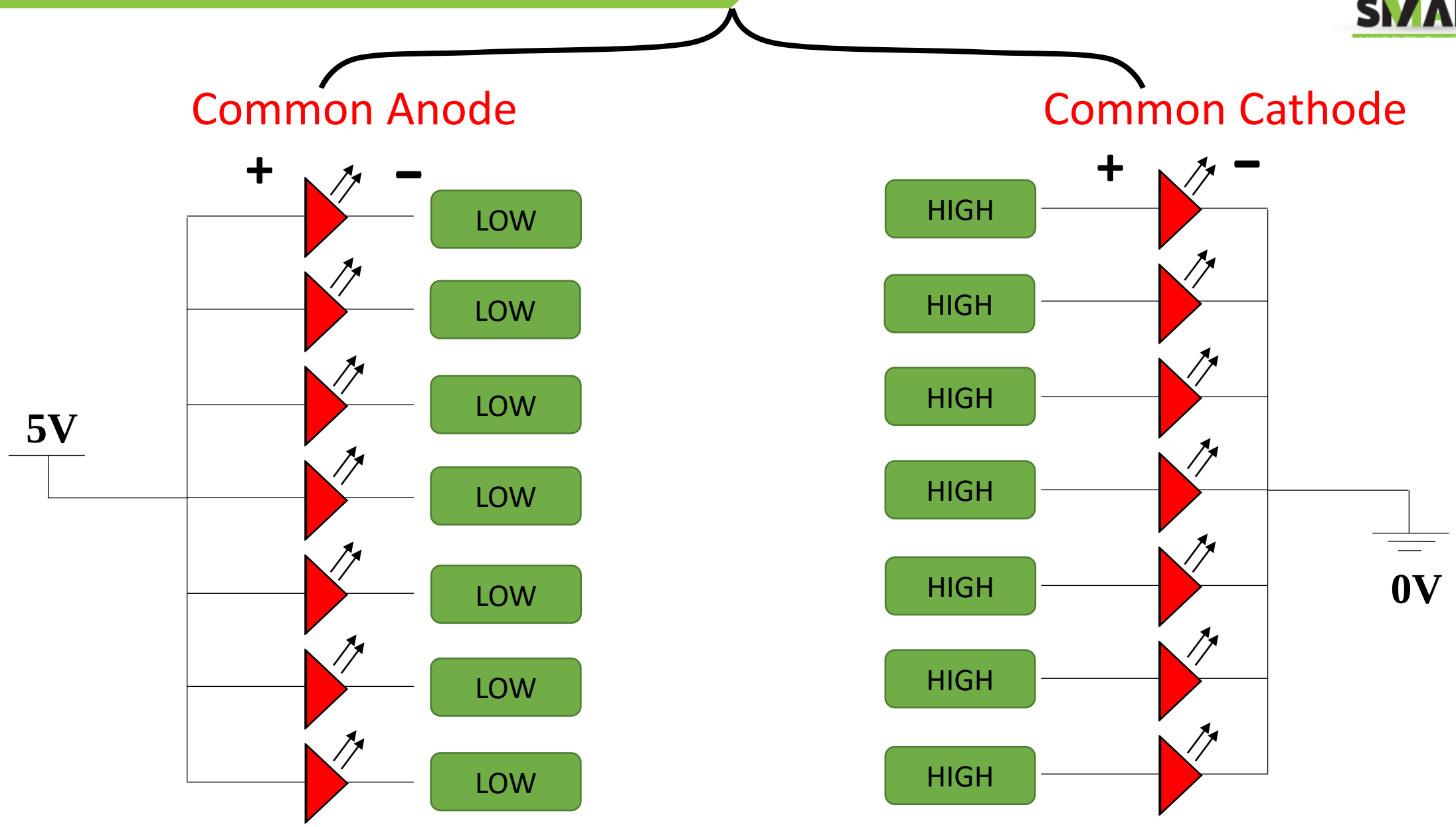
Q: How to test an anode RGB led ??

	<<<= Control 1 - Adjust Hue(Color) =>>>	<<<= Control 2 - Adjust FROM Colour to White =>>>																	
		1			2			3			4			5			6		
		R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
Red	23	255	0	0	255	100	100	255	150	150	255	200	200	255	230	230	255	255	255
	22	255	100	0	255	110	20	255	120	100	255	165	155	255	230	210	255	255	255
Orange	21	255	150	0	255	140	20	255	180	40	255	220	120	255	240	190	255	255	255
	20	255	200	0	255	210	20	255	230	100	255	240	150	255	255	200	255	255	255
Yellow	19	255	255	0	255	255	20	255	255	80	255	255	150	255	255	210	255	255	255
	18	200	255	0	200	255	20	220	255	80	230	255	150	235	255	180	255	255	255
Chartreuse	17	150	255	0	150	255	20	200	255	40	220	255	150	230	255	170	255	255	255
	16	100	255	0	110	255	20	150	255	40	200	255	150	230	255	160	255	255	255
Green	15	0	255	0	100	255	100	150	255	150	200	255	200	225	255	225	255	255	255
	14	0	255	100	20	255	150	100	255	180	180	255	210	235	255	220	255	255	255
Aquamarine	13	0	255	150	20	255	160	100	255	200	160	255	210	200	255	220	255	255	255
	12	0	255	200	20	255	200	40	255	220	140	255	230	200	255	255	255	255	255
Cyan	11	0	255	255	20	255	255	40	255	255	150	255	255	180	255	255	255	255	255
	10	0	200	255	20	210	255	40	230	255	60	240	255	170	255	255	255	255	255
Azure	9	0	150	255	20	180	255	40	200	255	60	230	255	150	255	255	255	255	255
	8	0	100	255	20	150	255	40	180	255	60	200	255	140	255	255	255	255	255
Blue	7	0	0	255	20	100	255	40	150	255	80	180	255	160	220	255	255	255	255
	6	100	0	255	120	20	255	140	100	255	150	150	255	200	200	255	255	255	255
Violet	5	150	0	255	150	50	255	180	100	255	220	140	255	230	180	255	255	255	255
	4	200	0	255	200	20	255	220	60	255	240	90	255	255	160	255	255	255	255
Magenta	3	255	0	255	255	60	255	255	100	240	255	150	250	255	180	255	255	255	255
	2	255	0	200	255	20	220	255	40	230	255	140	240	255	170	255	255	255	255
Rose	1	255	0	150	255	20	200	255	40	200	255	100	200	255	160	240	255	255	255
	0	255	0	100	255	20	120	255	80	140	255	100	160	255	140	180	255	255	255

• Automatic Color Change

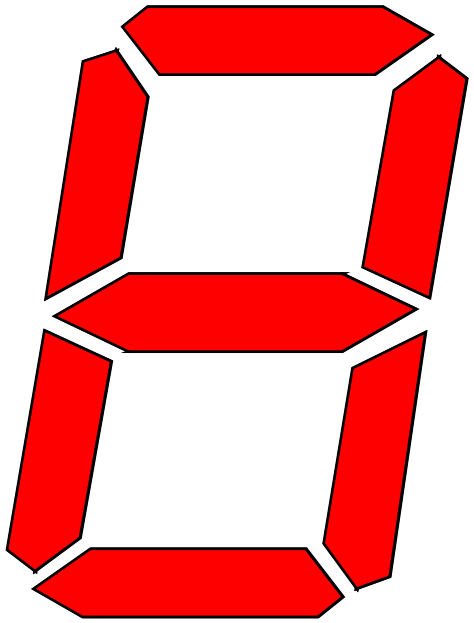
```
#define ledR 3
#define ledG 5
#define ledB 6

void setup()
{
 pinMode(ledR, OUTPUT);
 pinMode(ledG, OUTPUT);
 pinMode(ledB, OUTPUT);
}
void loop()
{
   for (byte R=0, G=100, B=255 ; R<=255 || G<=255 || B>=0 ; R++, G++, B-- ){

     analogWrite(ledR,R) ;
     analogWrite(ledG,G) ;
     analogWrite(ledB,B) ;
     delay(30);
   }}
```

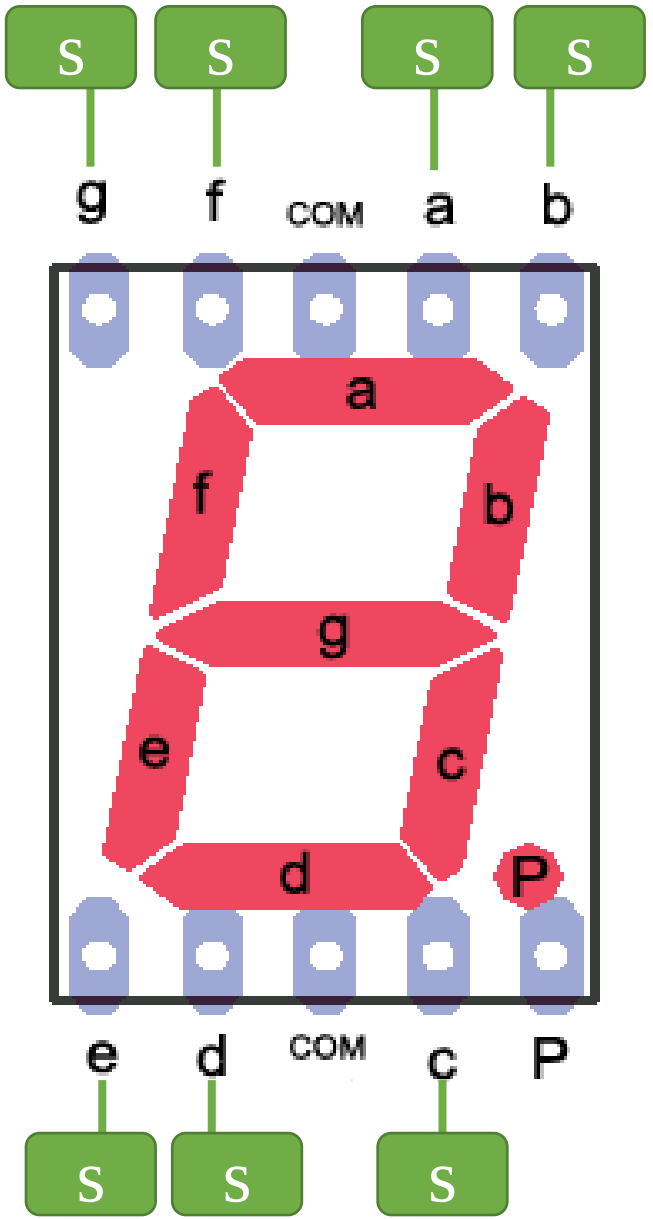
• 7-Segment Types



• 7-Segment Pins

8

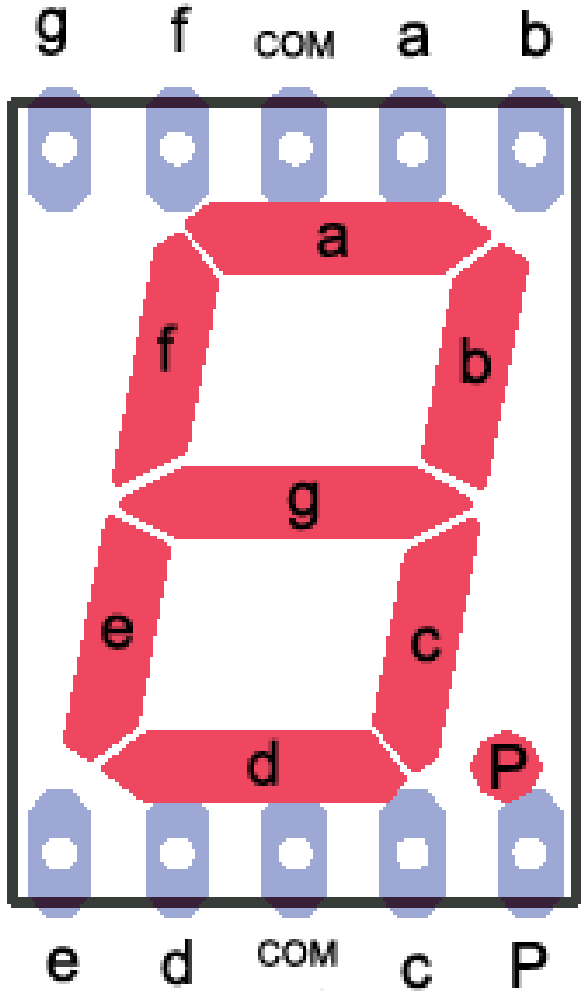




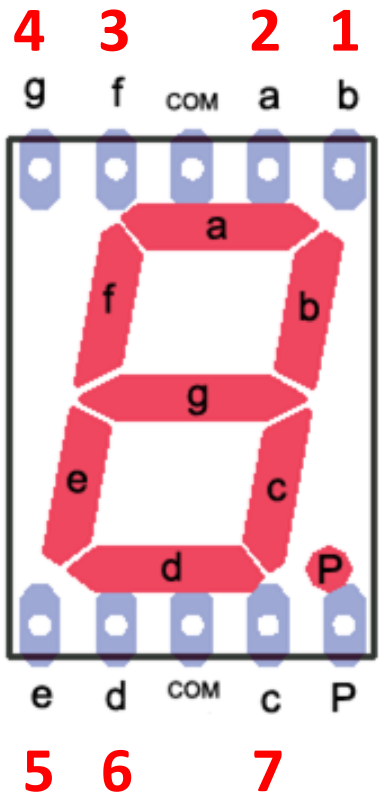
• Truth Table of 7Segment

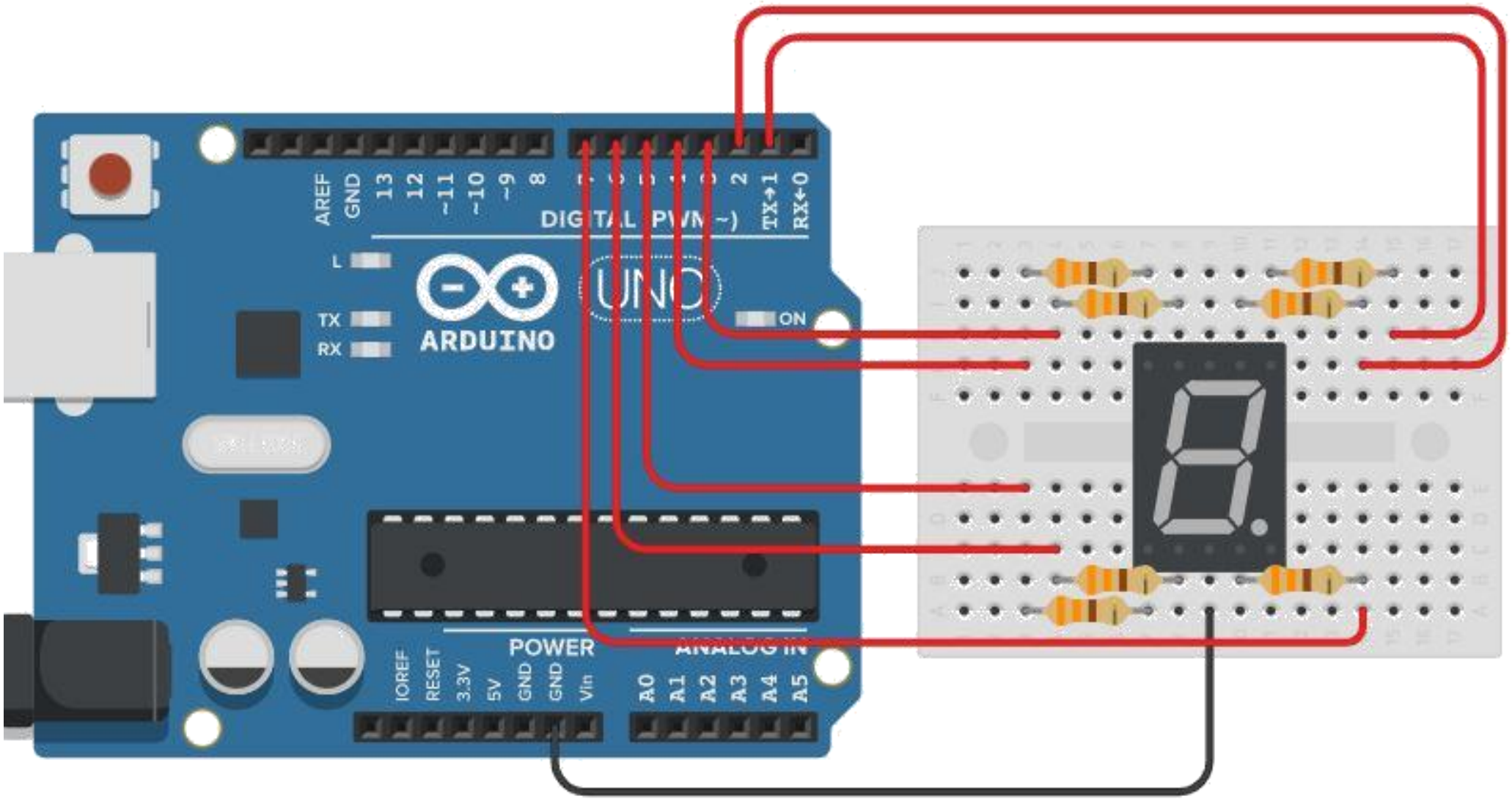
Segments (✓ = ON)							Display
a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0
	✓	✓					1
✓	✓		✓	✓		✓	2
✓	✓	✓	✓			✓	3
	✓	✓			✓	✓	4
✓		✓	✓		✓	✓	5
✓		✓	✓	✓	✓	✓	6
✓	✓	✓					7

Segments (✓ = ON)							Display
a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓	✓	8
✓	✓	✓			✓	✓	9
✓	✓	✓		✓	✓	✓	A
		✓	✓	✓	✓	✓	b
✓			✓	✓	✓		C
	✓	✓	✓	✓		✓	d
✓			✓	✓	✓	✓	E
✓				✓	✓	✓	F



• Connection


4 3 2 1
g f COM a b
a
f b
g
e c
d P
e d COM c P
5 6 7


AREF GND 13 12 ~11 ~10 ~9 8
DIGITAL PWM~)
TX→1 RX←0
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF RESET 3.3V 5V GND GND Vin
A0 A1 A2 A3 A4 A5

• Code

```
int b=1; int a=2; int f=3; int g=4;
int e=5; int d=6; int c=7;
void setup(){
 pinMode(a,OUTPUT);
 pinMode(b,OUTPUT);
 pinMode(c,OUTPUT);
 pinMode(d,OUTPUT);
 pinMode(e,OUTPUT);
 pinMode(f,OUTPUT);
 pinMode(g,OUTPUT);
}
void zero(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,0);}
```

```
void one(){
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,0);
}
void two(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,0);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,0);
digitalWrite(g,1);
}
```

```
void three(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,1);
}
void four(){
digitalWrite(a,0);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,1);
digitalWrite(g,1);
}
```

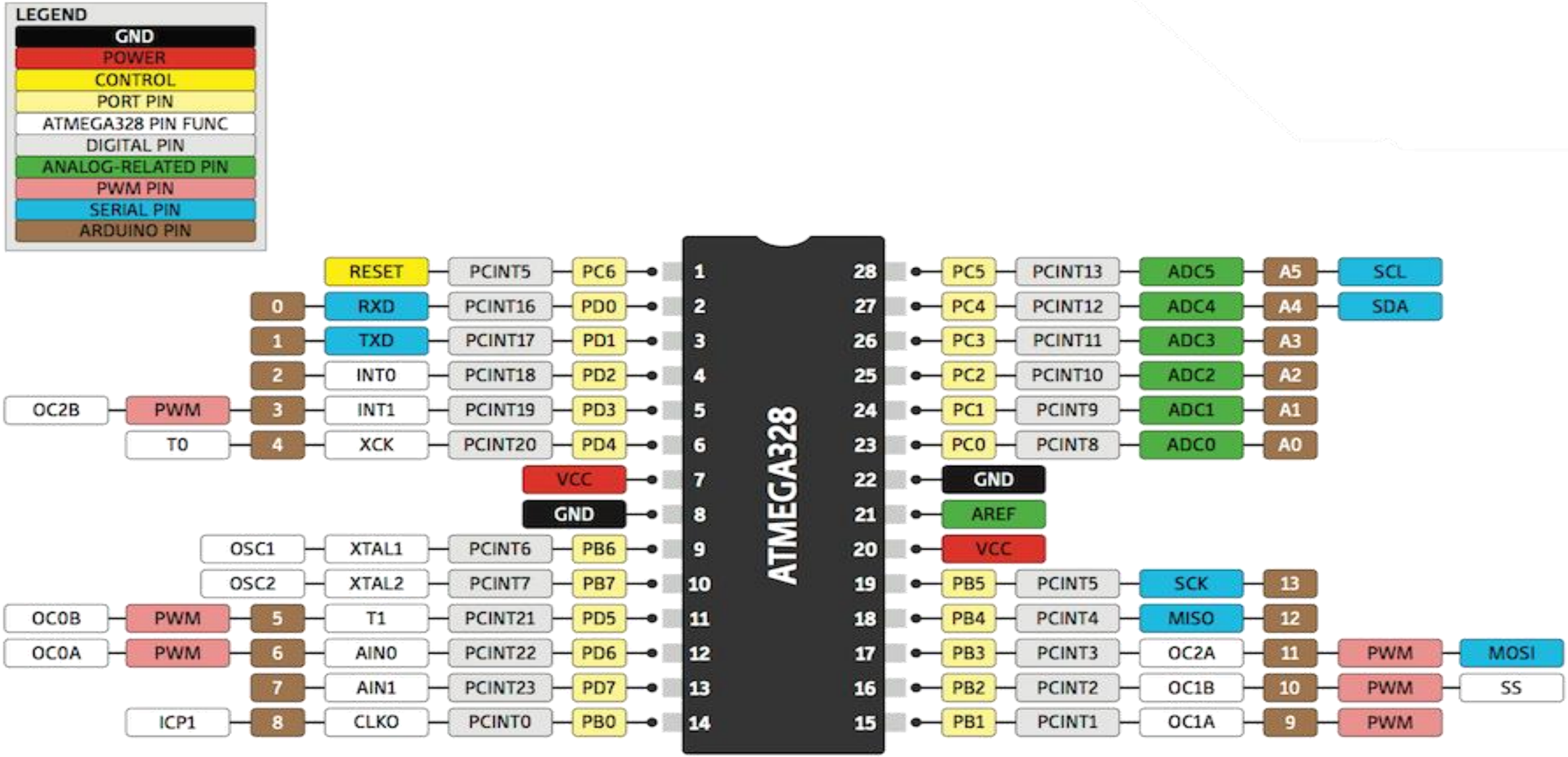
• Code

```
void five(){
digitalWrite(a,1);
digitalWrite(b,0);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,0);
digitalWrite(f,1);
digitalWrite(g,1);
}
void six(){
digitalWrite(a,1);
digitalWrite(b,0);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,1);
}
```

```
void seven(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,0);
}
void eight(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,1);
}
```

```
void nine(){
digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,0);
digitalWrite(f,1);
digitalWrite(g,1);}
void loop(){
   zero();  delay(1000);
   one();   delay(1000);
   two();   delay(1000);
   three(); delay(1000);
   four();  delay(1000);
   five();  delay(1000);
   six();    delay(1000);
  seven();  delay(1000);
   eight(); delay(1000);
   nine();  delay(1000); }
```

• ATMEGA328P Pinout

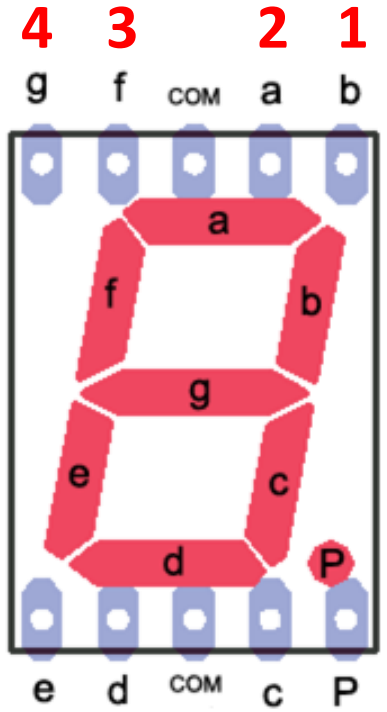


• 7-Segment in 2 Line code

Truth table

Segments (✓ = ON)							Display	Segments (✓ = ON)							Display
a	b	c	d	e	f	g		a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0	✓	✓	✓	✓	✓	✓	✓	8
	✓	✓					1	✓	✓	✓			✓	✓	9
✓	✓		✓	✓		✓	2	✓	✓	✓		✓	✓	✓	A
✓	✓	✓	✓			✓	3			✓	✓	✓	✓	✓	b
	✓	✓			✓	✓	4	✓			✓	✓	✓		C
✓		✓	✓		✓	✓	5		✓	✓	✓	✓		✓	d
✓		✓	✓	✓	✓	✓	6	✓			✓	✓	✓	✓	E
✓	✓	✓					7	✓				✓	✓	✓	F

connection



7	6	5	4	3	2	1	0
c	d	e	g	f	a	b	
1	1	1	1	1	1	1	1

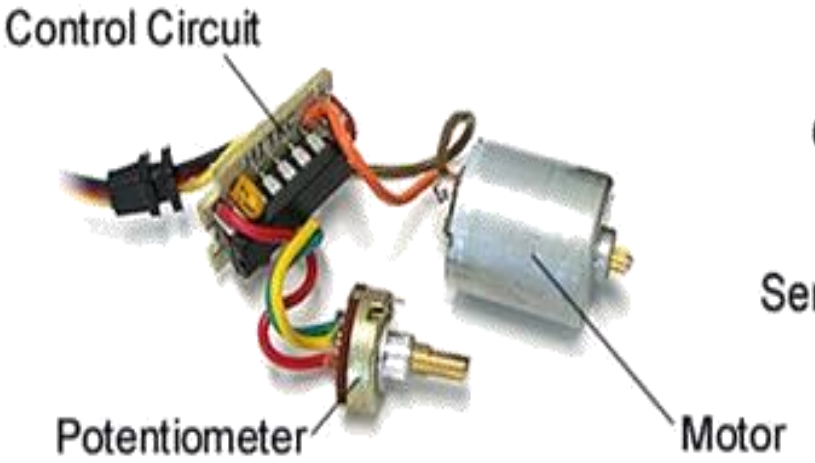
DDRD=0b11111111;

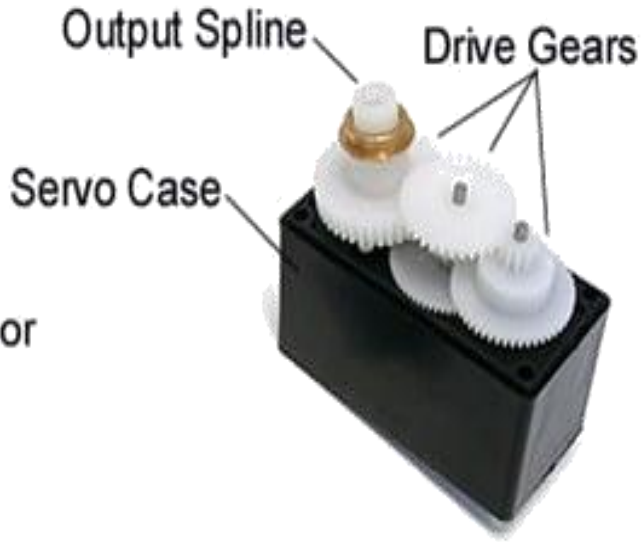
	7	6	5	4	3	2	1	0
	c	d	e	g	f	a	b	
2	0	1	1	1	0	1	1	0

PORTD=0b01110110;

• What is Servo Motor

The **servo motor** is most commonly used for high technology devices in the industrial application like automation technology. It is a self contained electrical device, that rotate parts of a machine with high efficiency and great precision. The output shaft of this motor can be moved to a particular angle. Servo motors are mainly used in home electronics, toys, cars, airplanes, etc.This article discusses about what is a servo motor, servo motor working, servo motor types and its applications





• Servo Motor Types

1) DC Servo Motor

2) AC Servo Motor

3) Positional Rotation Servo Motor

4) Continuous Rotation Servo Motor

• Positional Vs Continuous Servo

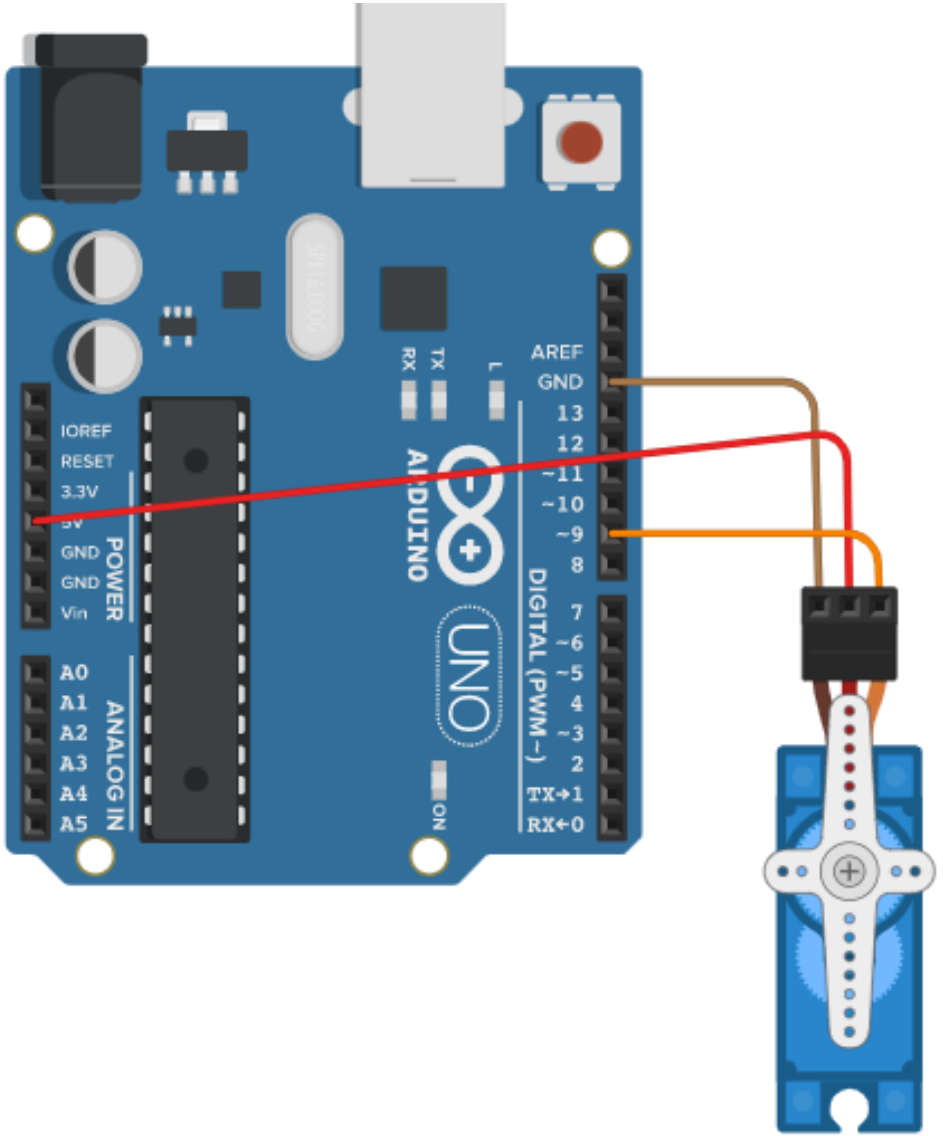
- **Positional Rotation Servo Motor**

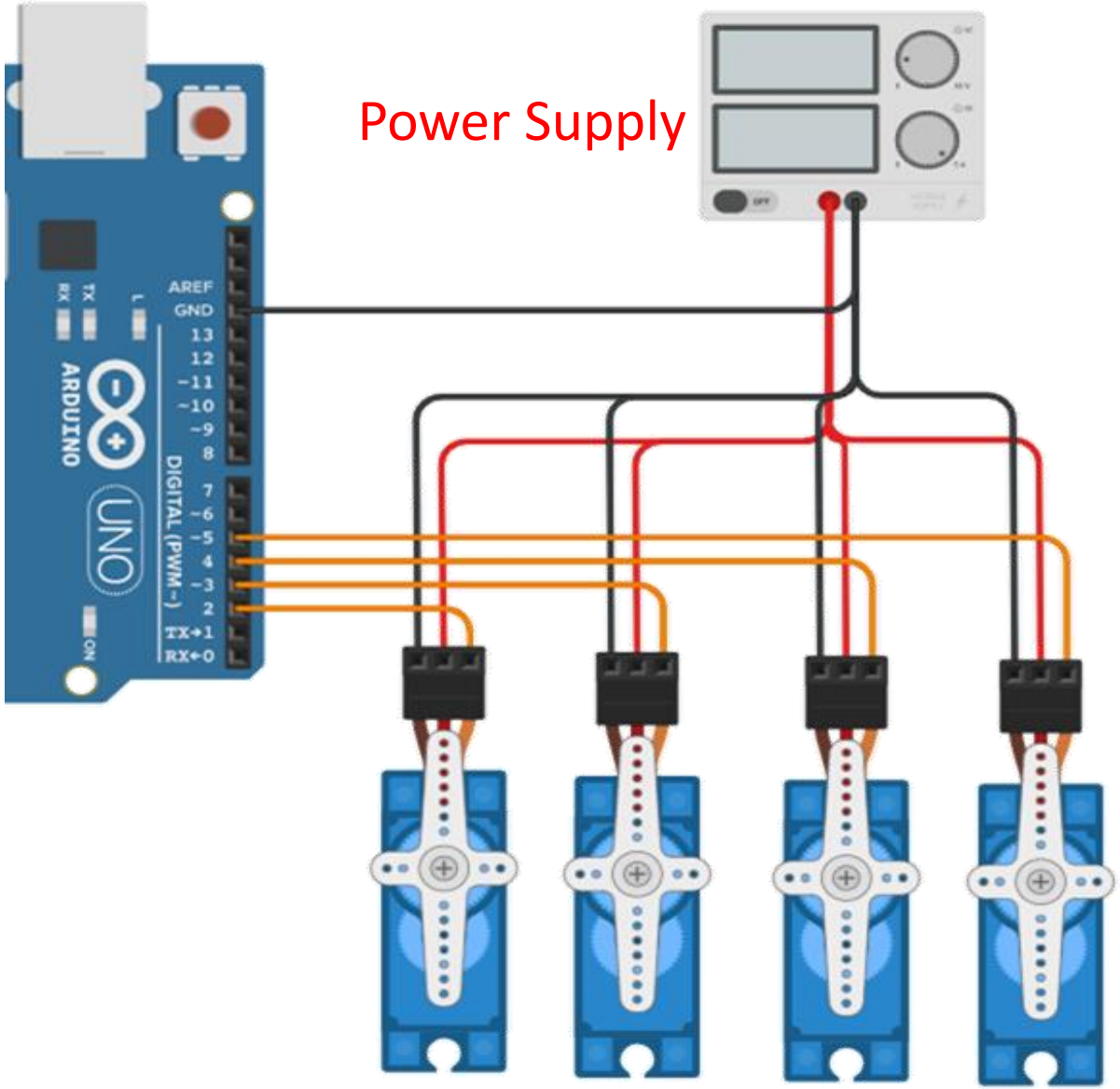
Positional rotation servo motor is a most common type of servo motor. The shaft's o/p rotates in about 180o. It includes physical stops located in the gear mechanism to stop turning outside these limits to guard the rotation sensor. These common servos involve in radio controlled water, radio controlled cars, aircraft, robots, toys and many other applications.

- **Continuous Rotation Servo Motor**

Continuous rotation servo motor is quite related to the common positional rotation servo motor, but it can go in any direction indefinitely. The control signal, rather than set the static position of the servo, is understood as the speed and direction of rotation. The range of potential commands sources the servo to rotate clockwise or anticlockwise as preferred, at changing speed, depending on the command signal. This type of motor is used in a radar dish if you are riding one on a robot or you can use one as a drive motor on a mobile robot.

• Servo Motor Connection





• Sweep code

```
#include <Servo.h>
Servo myservo;
int pos = 0;
void setup() {
  myservo.attach(9);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    // in steps of 1 degree
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}
```

• Knop Code

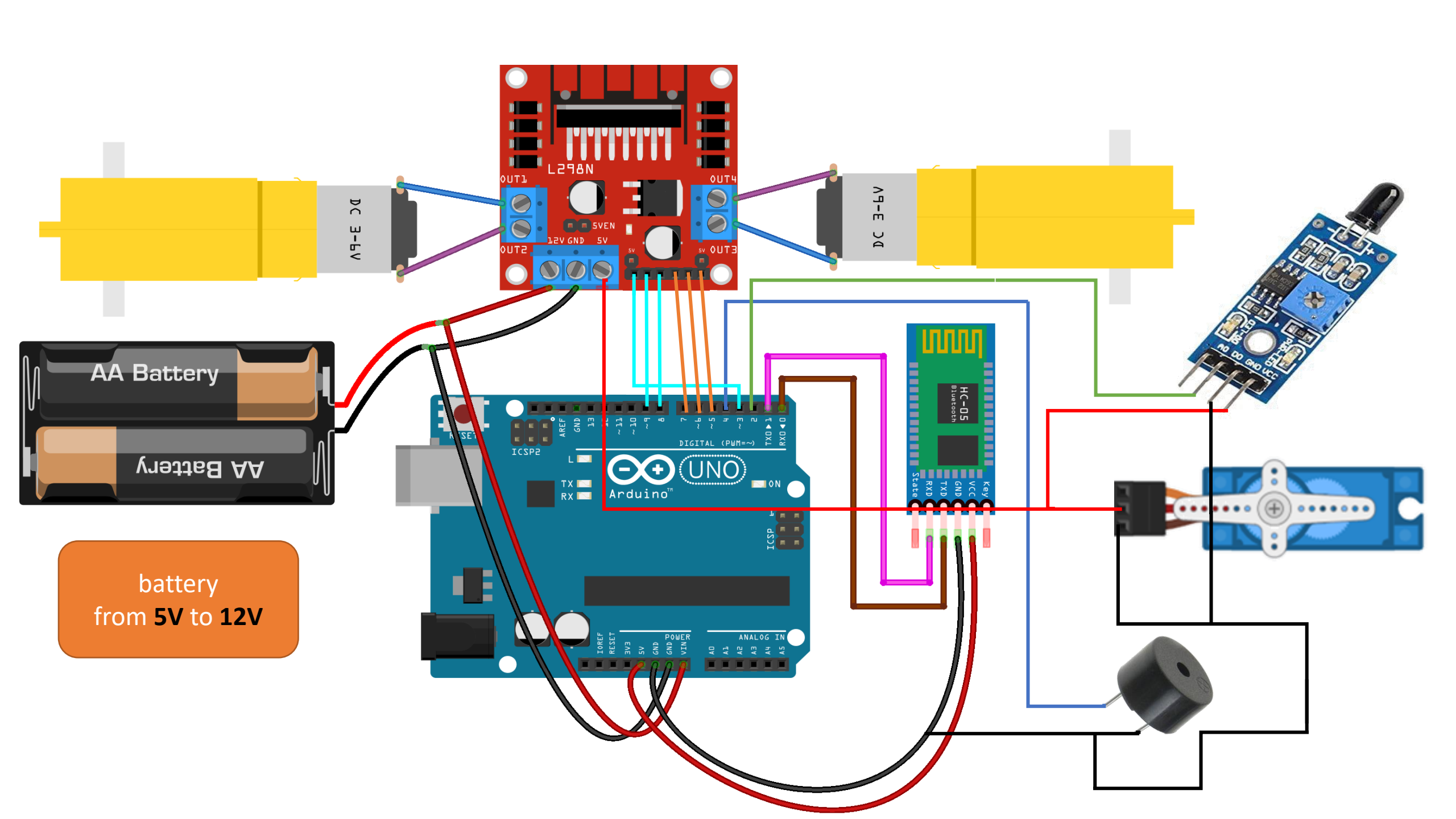
```
#include <Servo.h>
Servo myservo;
int potpin = A0;
int val;

void setup() {
  myservo.attach(9);
}

void loop() {
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 180);
  myservo.write(val);
  delay(15);
}
```

• Task

- Write a code that’s Show the same number you send from your computer on the 7-segment

THANKS FOR COMING

LECTURE
9
(Practical)


L298N
OUT1
OUT2
OUT3
OUT4
5VEN
12V GND 5V
DC 3-6V
DC 3-6V
AA Battery
AA Battery
UNO
Arduino
DIGITAL (PWM=~)
POWER
ANALOG IN
ICSP2
ICSP
HC-05
Bluetooth
State
RXD
TXD
GND
VCC
Key
AO DO GND VCC
battery
from 5V to 12V

```
#include <Servo.h>
#define speedL 3
#define IN1 9
#define IN2 8
#define IN3 7
#define IN4 6
#define speedR 5
#define flame 2
#define buzzer 4
char Reading;
int pos , flame_detected=0 ;
Servo myservo;

void setup() {
  Serial.begin(9600);
  myservo.attach(11);
  myservo.write(90);
  for (int i = 3 ; i <= 9 ; i++)
  {
    pinMode(i, OUTPUT);
  }
    pinMode(flame, INPUT);
}
```

```
void forword()
{ digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 150);
  analogWrite(speedR, 150);
}
void backword()
{ digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(speedL, 150);
  analogWrite(speedR, 150);
}
void left()
{ digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 0);
  analogWrite(speedR, 150);
}
```

```
void right()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(speedL, 150);
  analogWrite(speedR, 0);
}

void stopp() {
digitalWrite(speedL, LOW);
digitalWrite(speedR, LOW);
}
void loop() {
  if (Serial.available() > 0) {
  Reading = Serial.read();
  switch (Reading) {
   case 'F' : forword(); break;
   case 'B' : backword();break;
   case 'R' : right(); break;
   case 'L' : left(); break;
   case 'S' : stopp(); break;
```

```
case 'Q' : for (pos = 90; pos <= 180; pos += 1) {
                        if(digitalRead(flame)==0){flame_detected++;}
                        myservo.write(pos);
                        delay(15); }

                for (pos = 180; pos >= 0; pos -= 1) {
                        if(digitalRead(flame)==0){flame_detected++;}
                        myservo.write(pos);
                        delay(15);}

                        myservo.write(90);

                        if(flame_detected > 0){
                          for(int i=0 ; i<=10 ; i++){
                        digitalWrite(buzzer,1);
                        delay(100);
                        digitalWrite(buzzer,0);
                        delay(100);
                        }
                        flame_detected=0;
                        }

                        }}}
```

THANKS FOR COMING

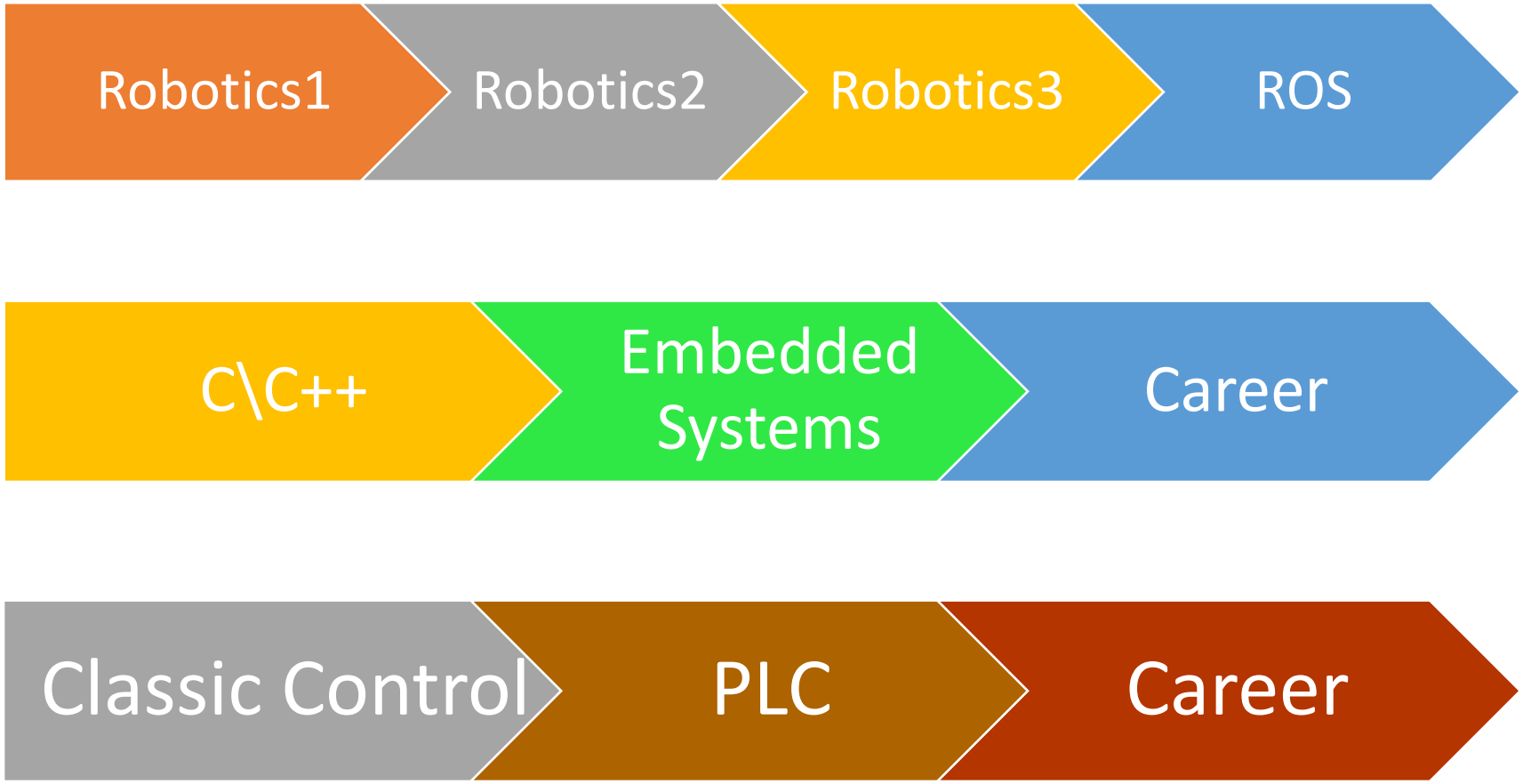
LECTURE

10

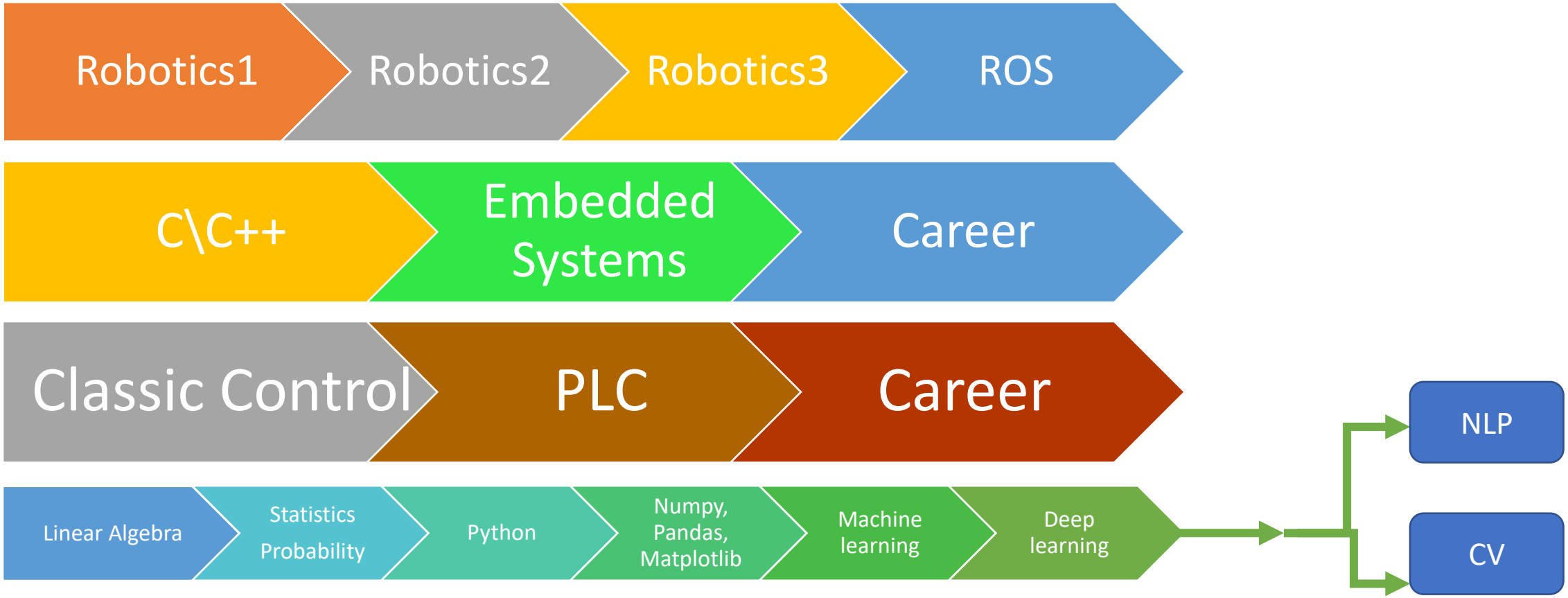
Oral Exam

Graduation Project

• What To Do Next



• What To Do Next



THANKS FOR COMING