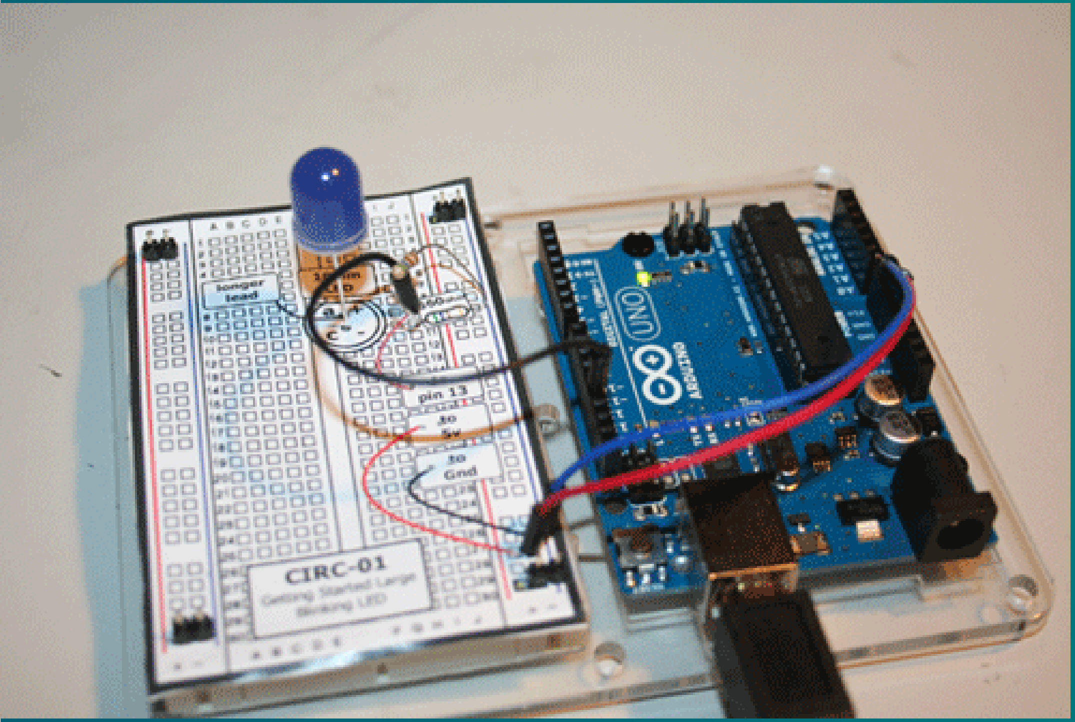
Internet Of Things



Analog Digital Input Output

Define & declare it clearly!

Purwadhika
Startup and Coding School

Electronic Signals

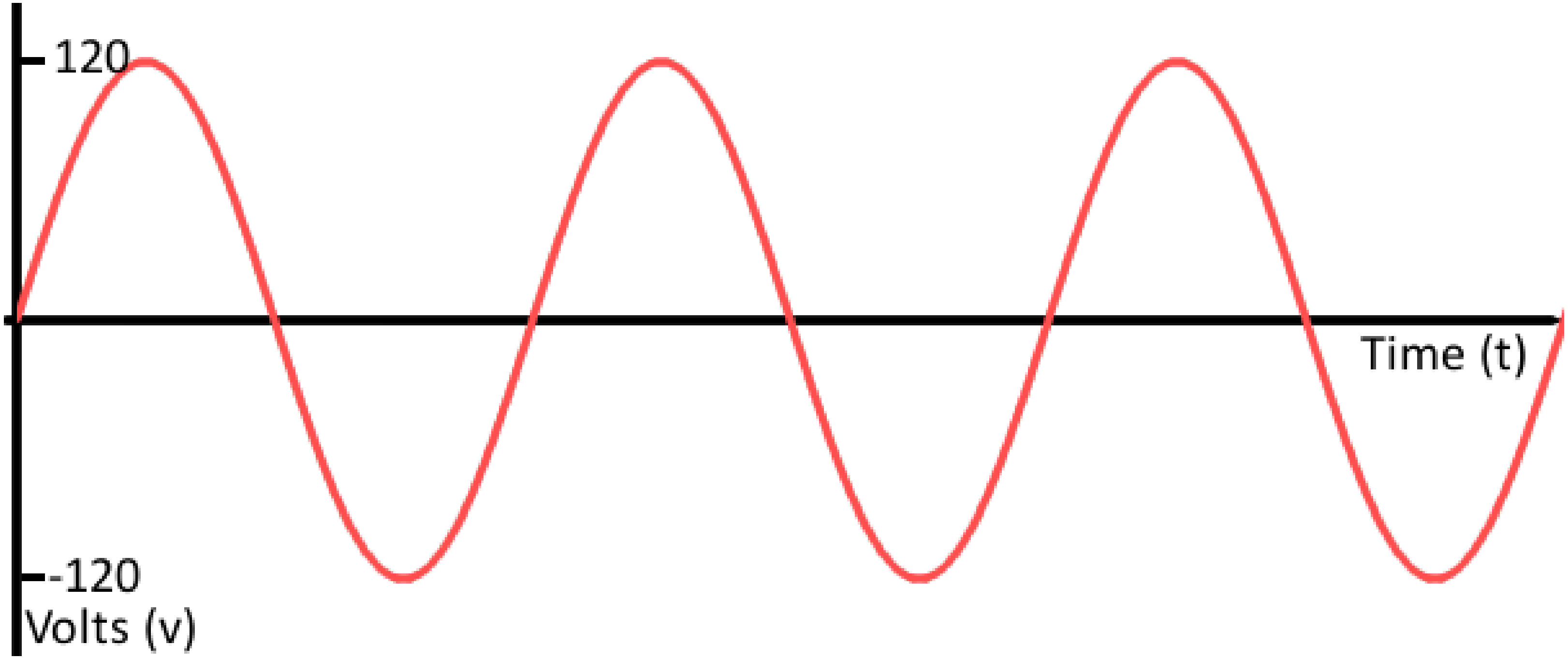
- Signals are time-varying "quantities" which convey some sort of information. In electrical engineering the quantity that's time-varying is usually voltage (orcurrent). So when we talk about signals, just think of them as a voltage that's changing over time.

- Signals are passed between devices in order to send and receive data information. Usually the signals are transmitted through wires, but they could also pass through the air wirelessly.

- A signal varies over time, so it's helpful to plot it on a graph where time is plotted on x-axis, and voltage on y-axis. Looking at a graph of a signal is the easiest way to identify if it's analog or digital.

Analog Signal

A time-versus-voltage graph of an **analog signal** should be **smooth and continuous**.

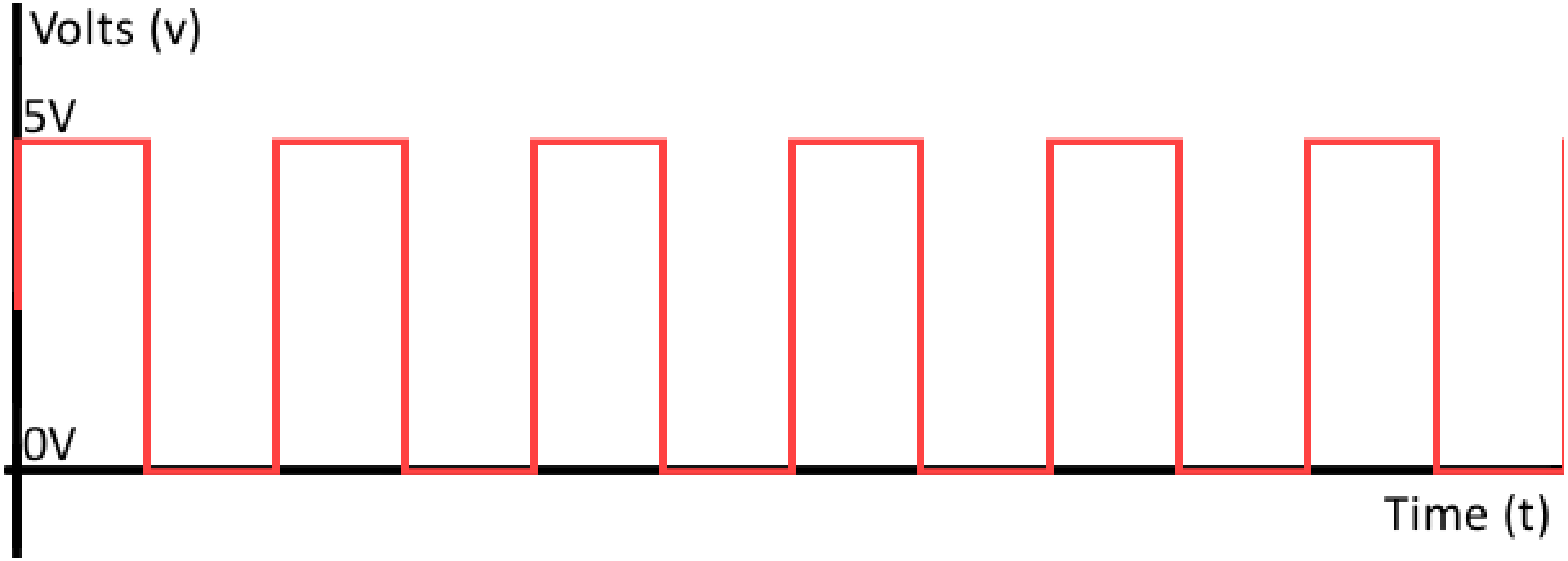


Example: RCA video & mic audio transmissions

Purwadhika
Startup and Coding School

Digital Signal

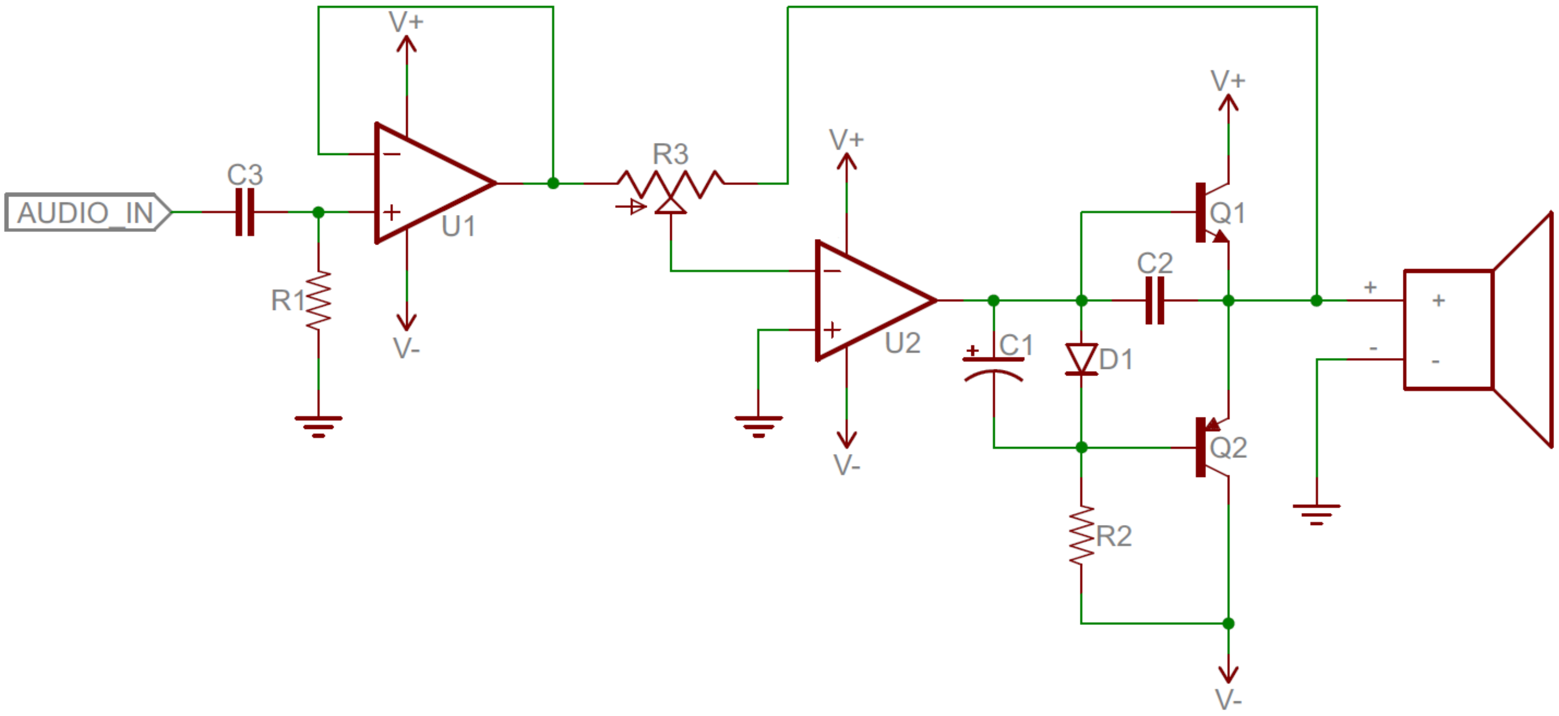
Timing graphs of digital signals are stepping, square, and discrete. Usually called as square waves.



Example: HDMI video & MIDI audio transmissions

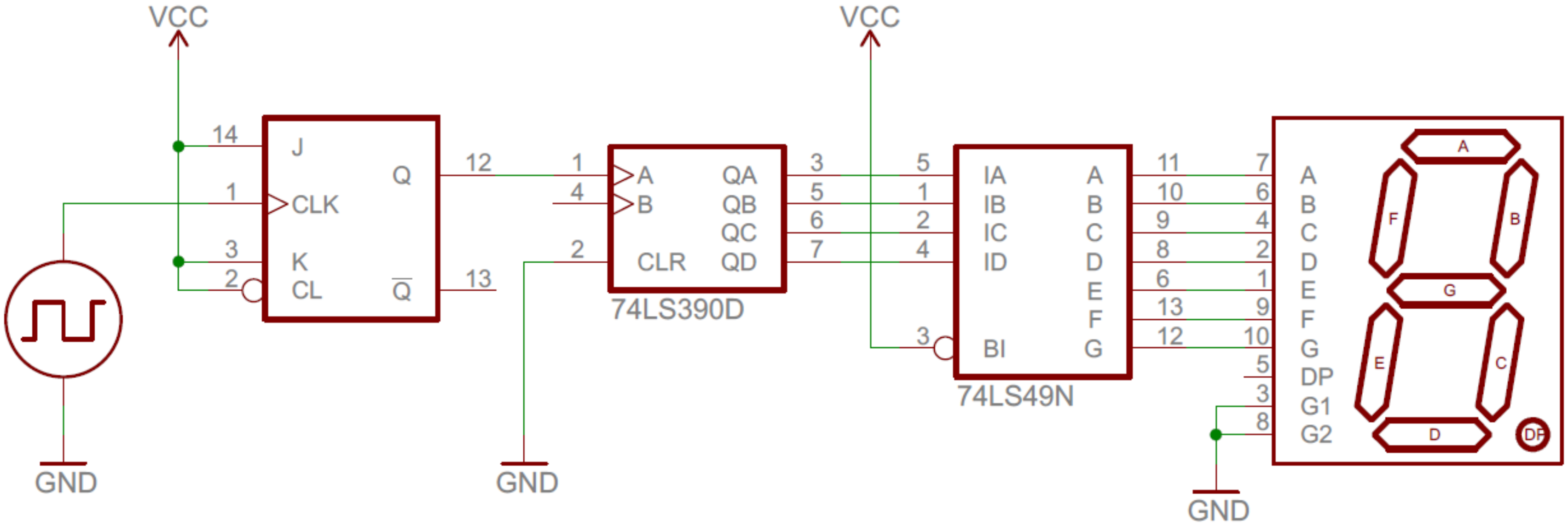
Analog Circuit

- Most of the fundamental electronic components: resistors, capacitors, inductors, diodes, transistors, and operational amplifiers, are all inherently analog. Analog circuits are much more susceptible to noise (small, undesired variations in voltage). Small changes in the voltage level of an analog signal may produce significant errors when being processed.



Digital Circuit

- Digital circuits operate using digital/discrete signals. These circuits are usually made of a combination of transistors and logic gates and, at higher levels, microcontrollers or other computing chips. Most processors, whether they're big beefy processors in your computer, or tiny little microcontrollers, operate in the digital realm.



Combined Circuit

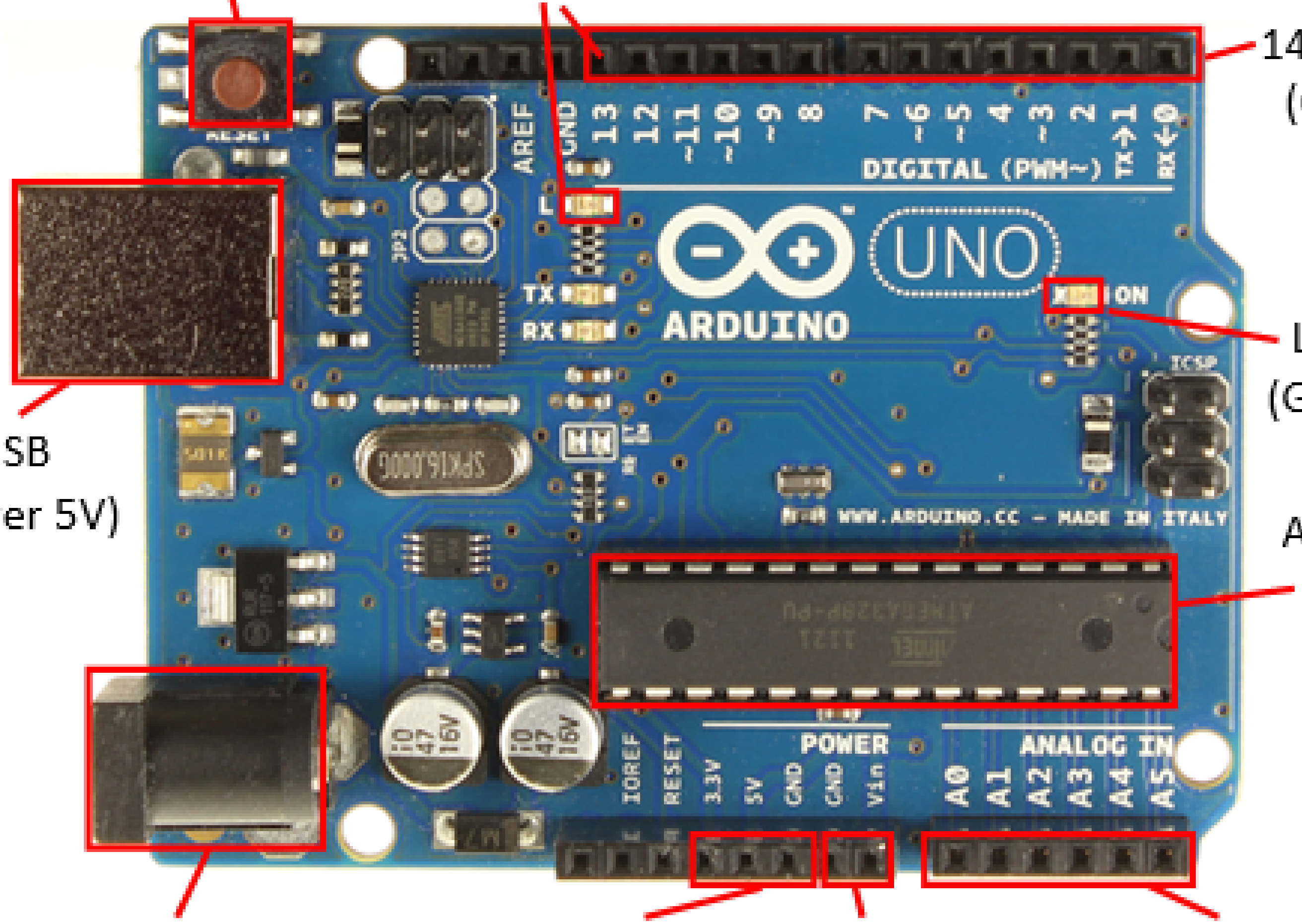
- It's not rare to see a mixture of analog and digital components in a circuit. Although microcontrollers are usually digital beasts, they often have internal circuitry which enables them to interface with analog circuitry.

- An analog to digital converter (ADC) allows a microcontroller to connect to an analog sensor (like photocells or temperature sensors), to read in an analog voltage.

- The less common digital to analog converter (DAC) allows a microcontroller to produce analog voltages, which is handy when it needs to make sound.

- Pulse Width Modulation (PWM) is a trick microcontrollers can use to make a digital signal appear to be analog.

Digital Analog I/O Uno



Reset Button

LED – Load & Pin 13

14x Digital IN/OUT
(6x PWM~ OUT)
(5V, 40mA)

USB
(Power 5V)

LED - Power ON
(Green or Orange)

Atmel ATmega328P
Microcontroller
(8-bit, 16 MHz,
32 KB Flash,
1 KB EEPROM,
2 KB SRAM)

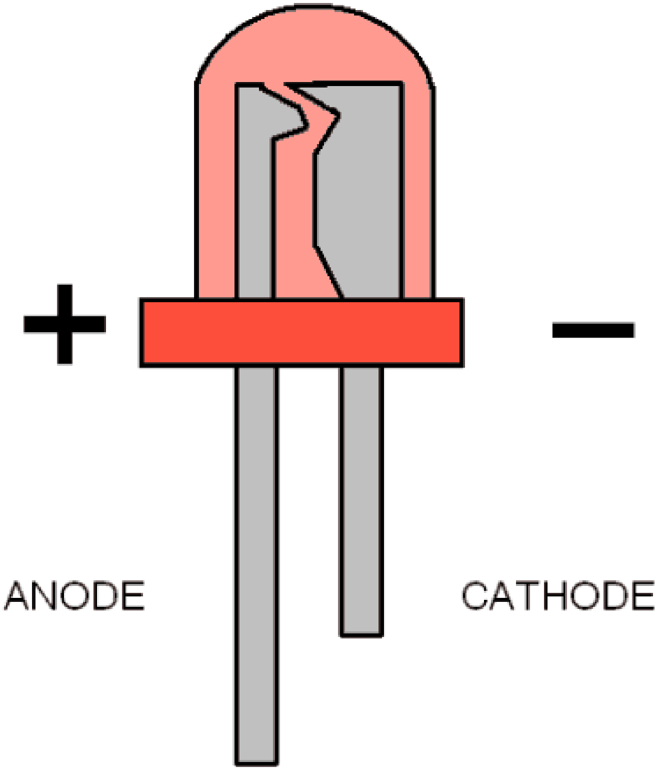
DC Power Jack
(AC-to-DC adapter)
(7-12V)

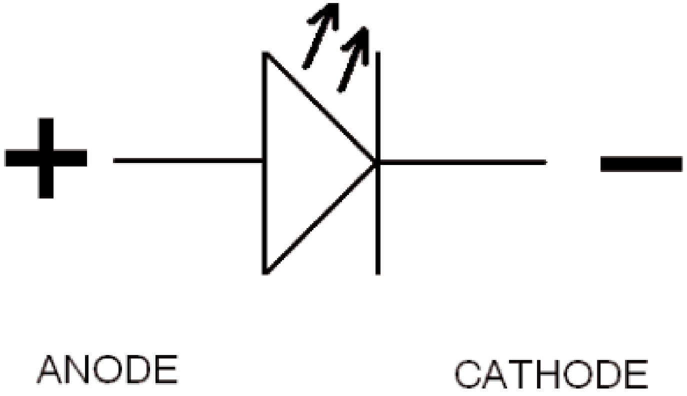
Power OUT
(5V, 3.3V)

Power IN
(9V battery)

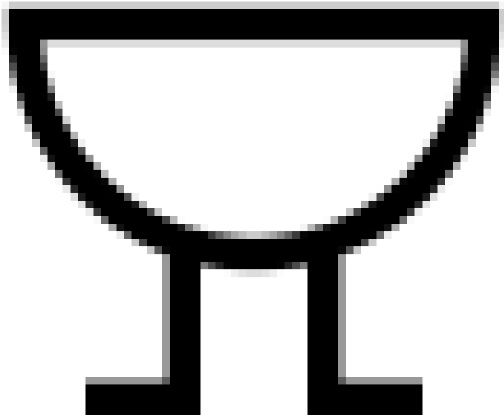
6x Analog IN
(0-5V 10-bit ADC)

LED (Light Emitting Diode)

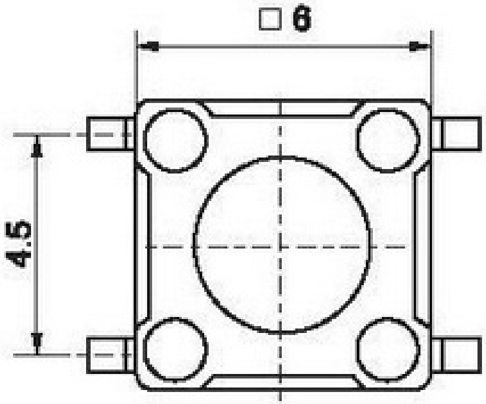


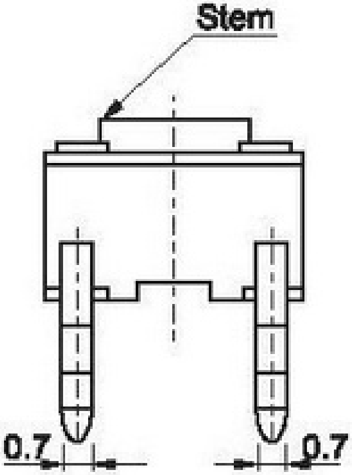


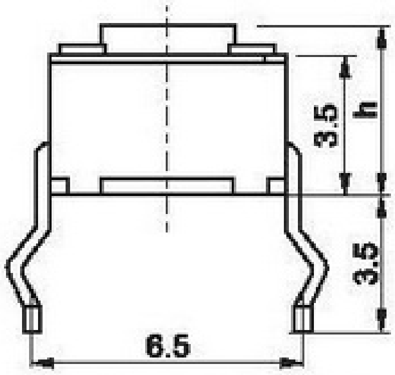
Buzzer



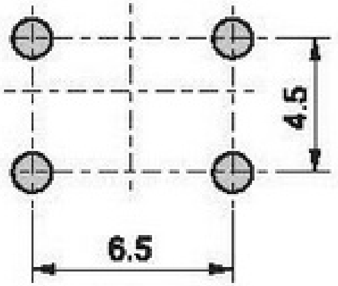
Push Button



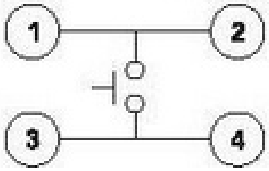




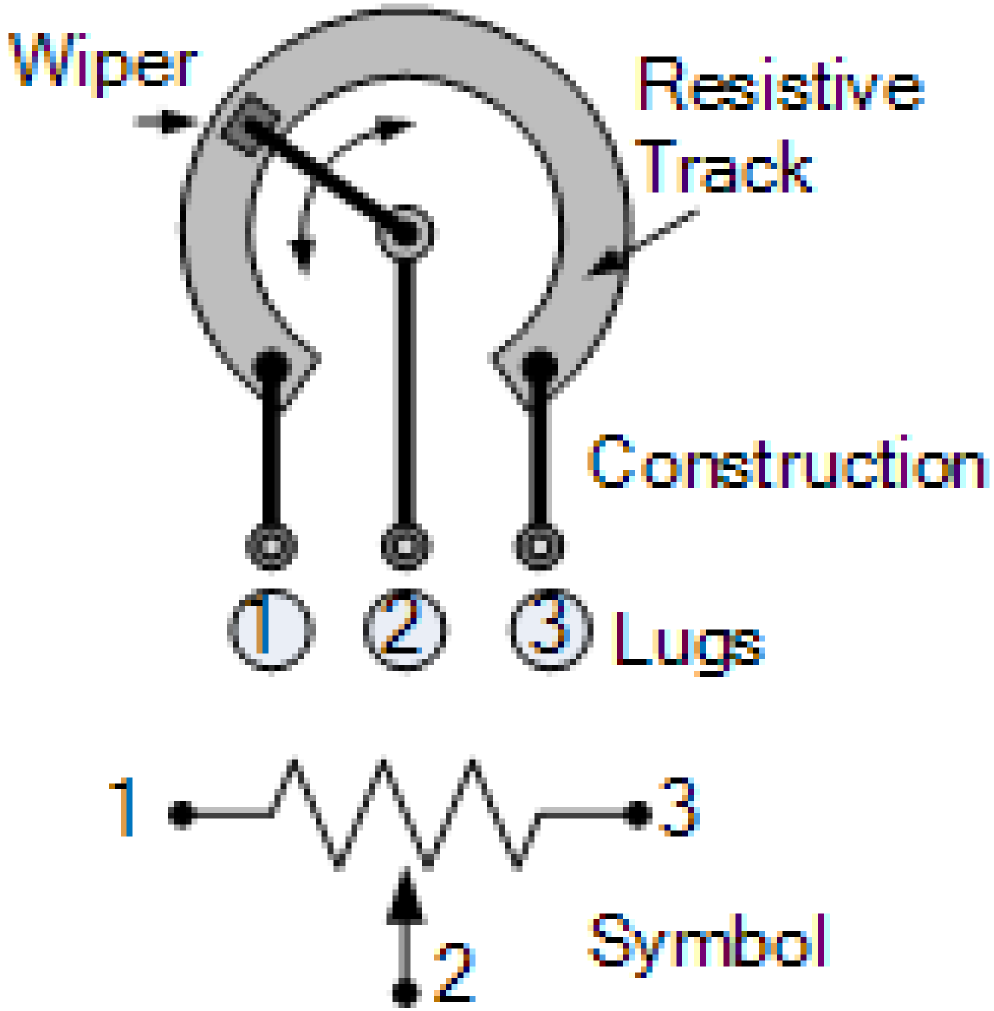
P.C.B.Land Pattern



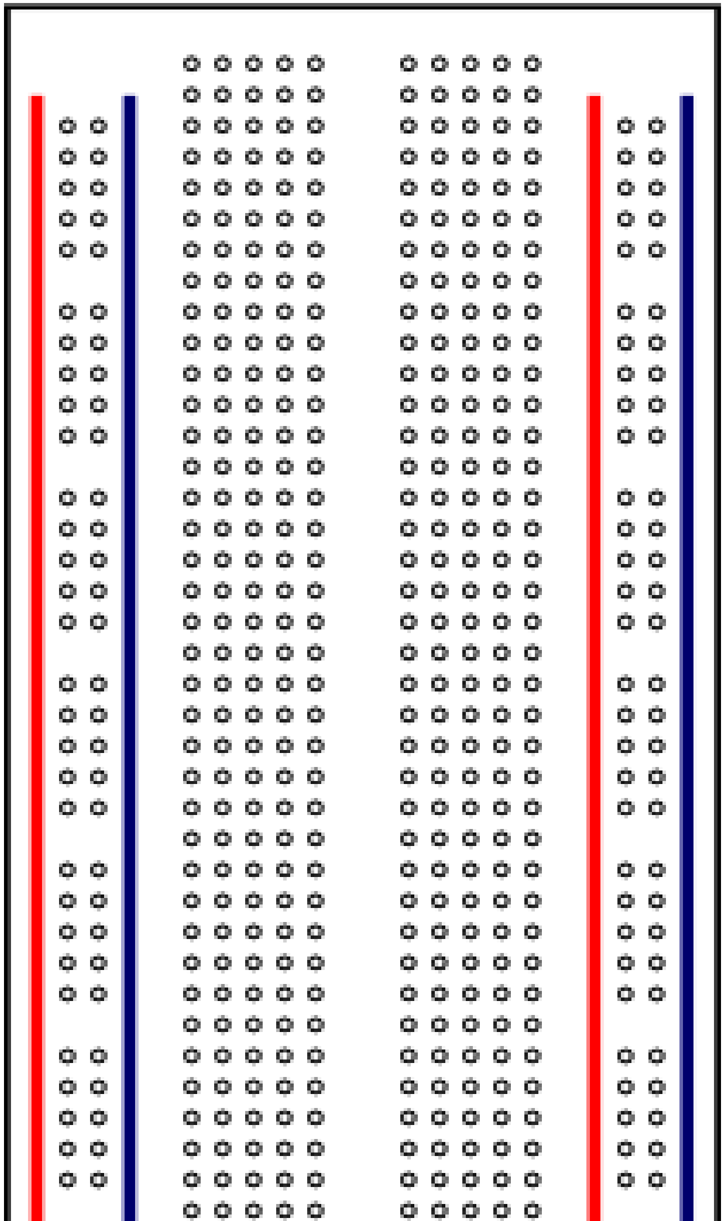
Circuit Diagram

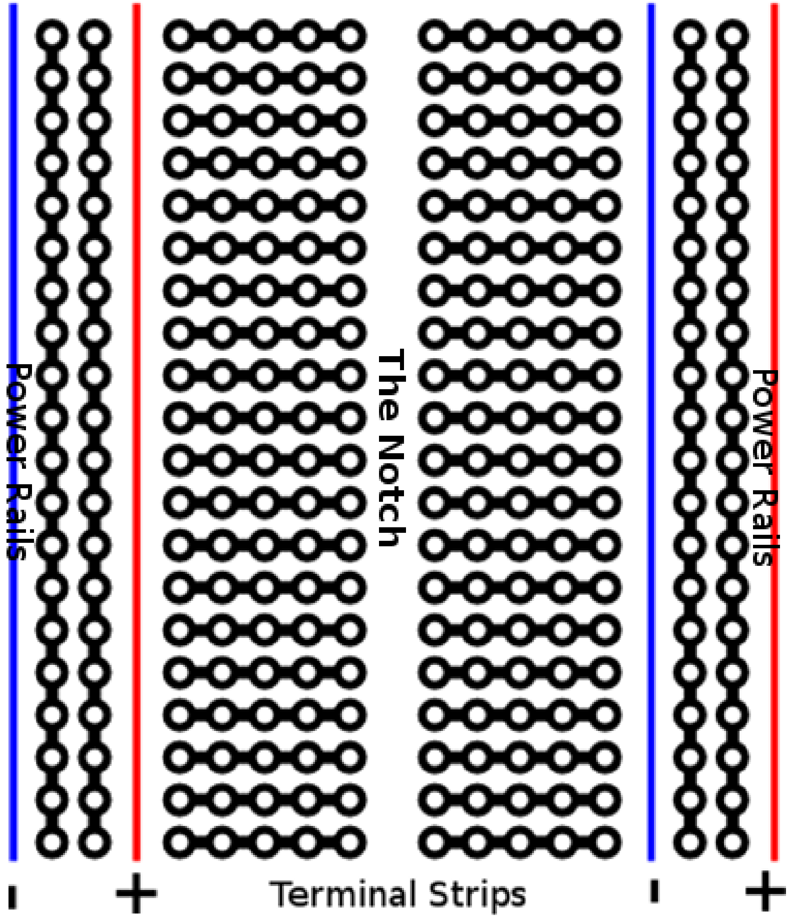


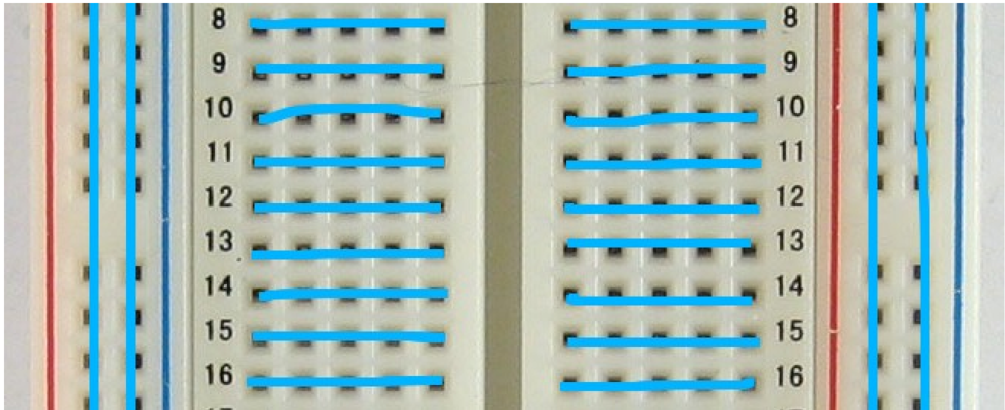
Potentiometer



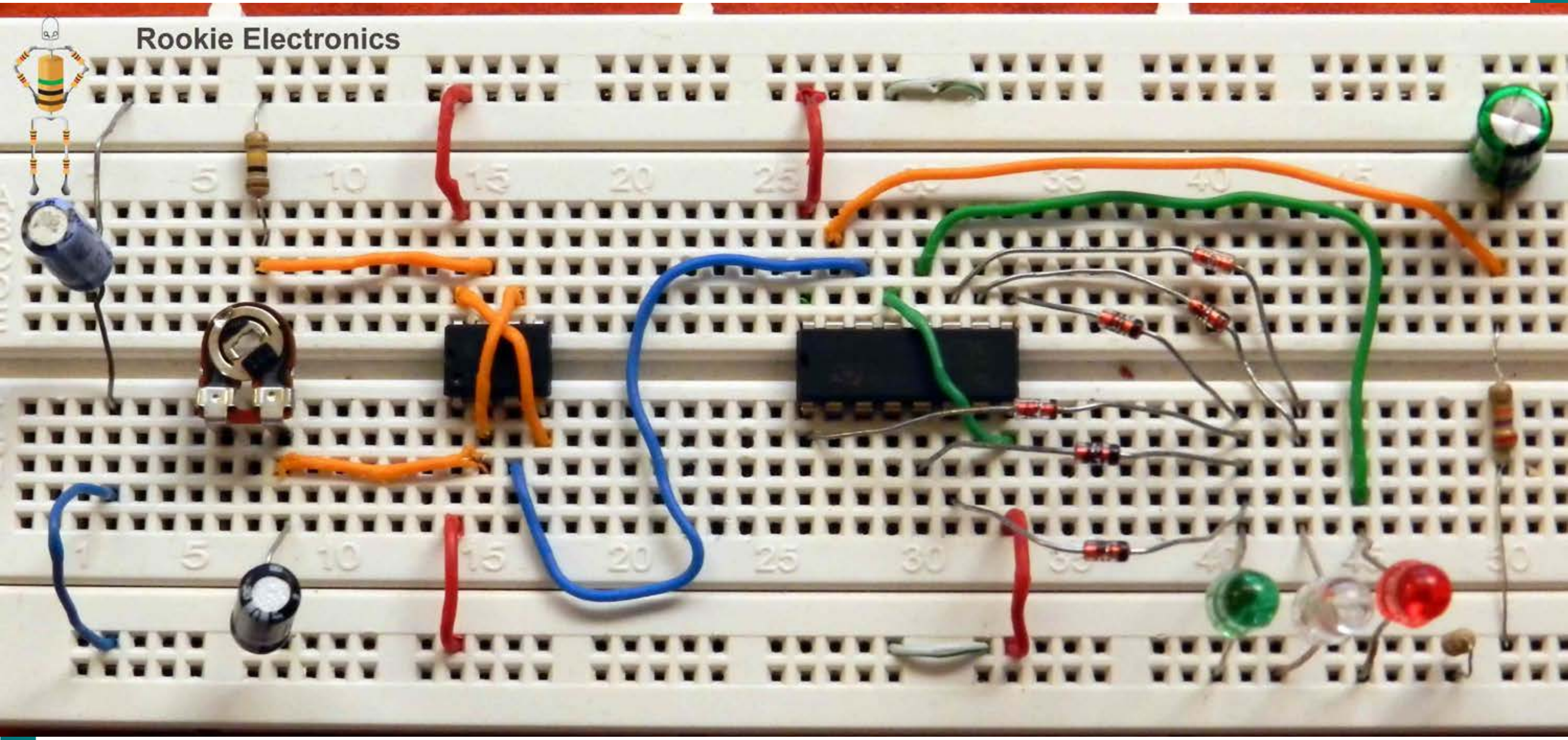
Breadboard




Power Rails
The Notch
Power Rails
-
+
Terminal Strips
-
+


8
9
10
11
12
13
14
15
16
8
9
10
11
12
13
14
15
16

Prototyping with Breadboard

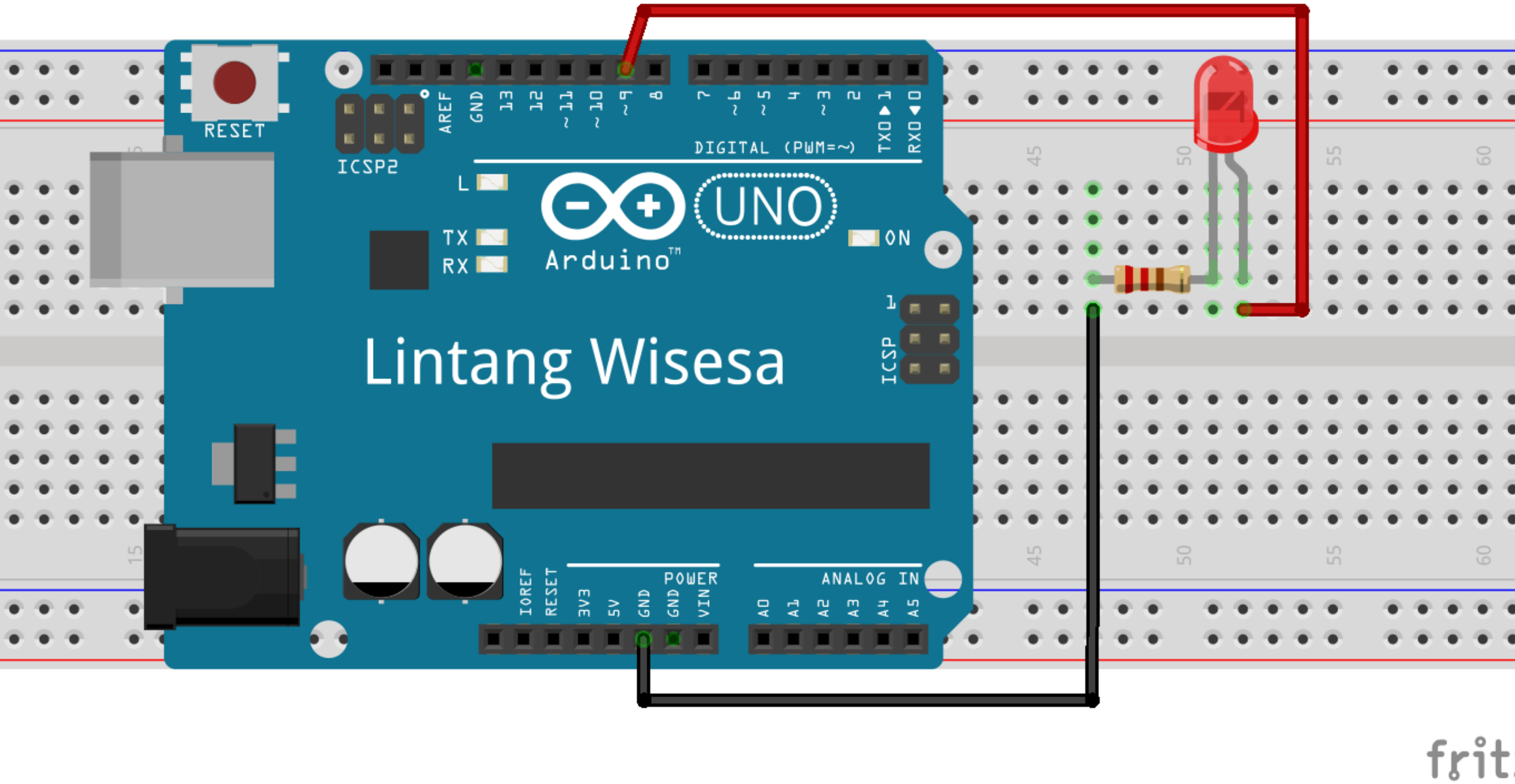


Purwadhika
Startup and Coding School

Analog Digital I/O

```
1.digitalWrite(Dpin,HIGH/LOW);

2.digitalRead(Dpin);

3.analogWrite(PWMpin,0-255)

4.analogRead(Apin);
```

Blink LED



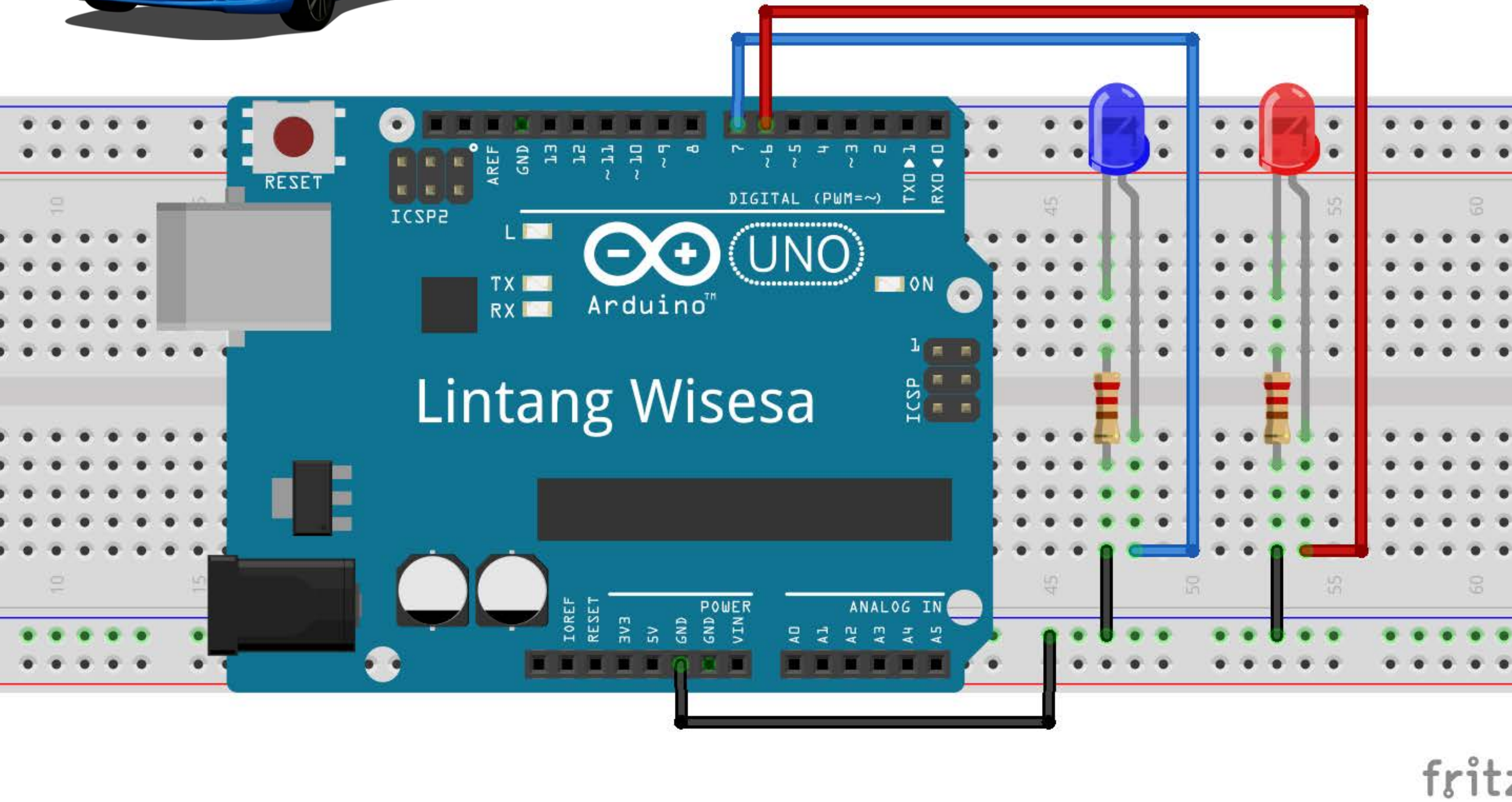
Purwadhika
Startup and Coding School

Blink LED

```
void setup() {
  pinMode(9, OUTPUT);
}

void loop() {
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(1000);
}
```

Police
Police

Strobo


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
1
ICSP
Lintang Wisesa
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
10
15
45
50
55
60
fritz

Strobo

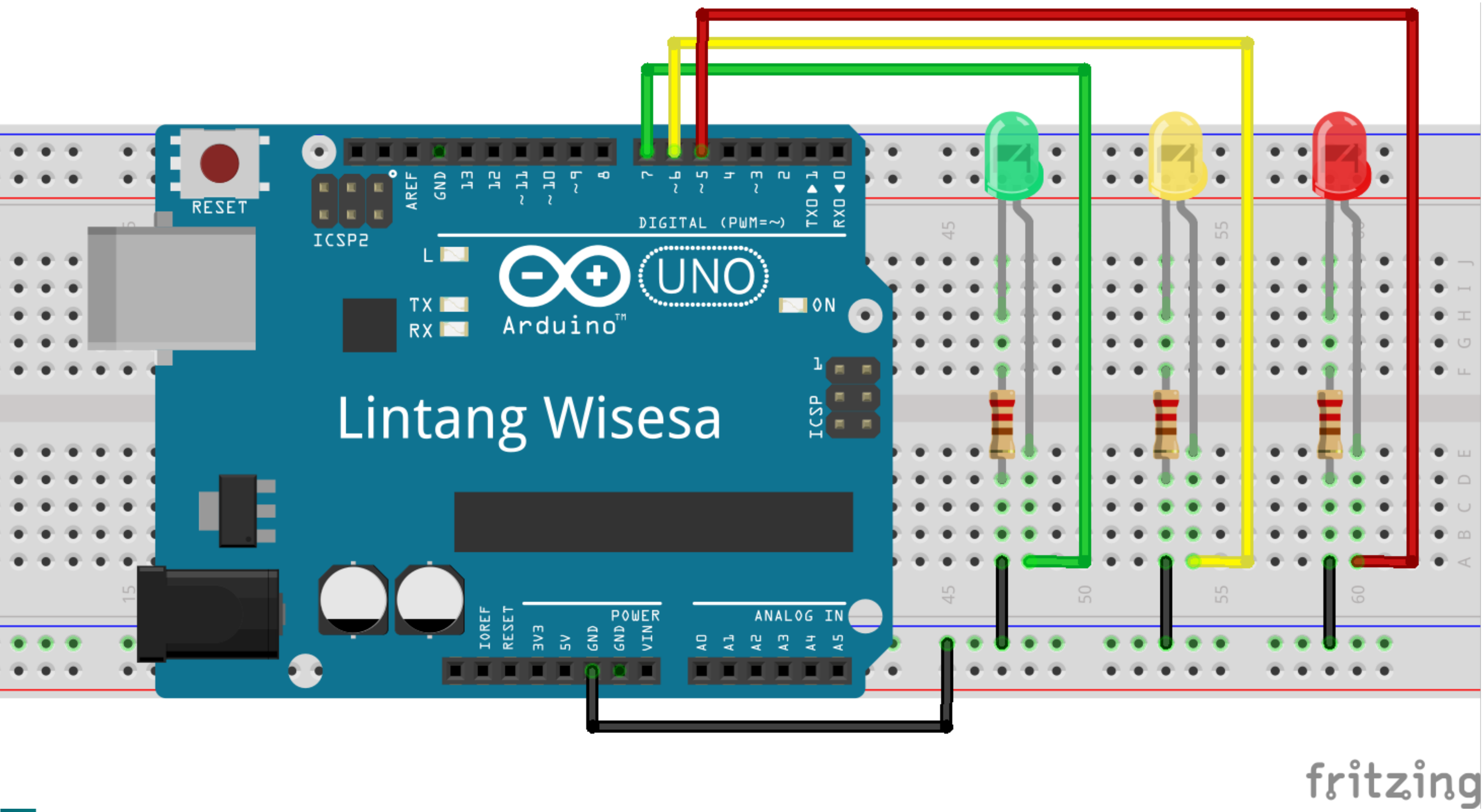
```
void setup() {
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);}

void loop() {
  digitalWrite(6, HIGH); delay(300);
  digitalWrite(6, LOW); delay(300);
  digitalWrite(6, HIGH); delay(300);
  digitalWrite(6, LOW); delay(300);
  digitalWrite(7, HIGH); delay(300);
  digitalWrite(7, LOW); delay(300);
  digitalWrite(7, HIGH); delay(300);
  digitalWrite(7, LOW); delay(300);}
```

Try It!

Build a simple Traffic Light Miniature With 3 LEDs!

Purwadhika
Startup and Coding School

Traffic Light Miniature

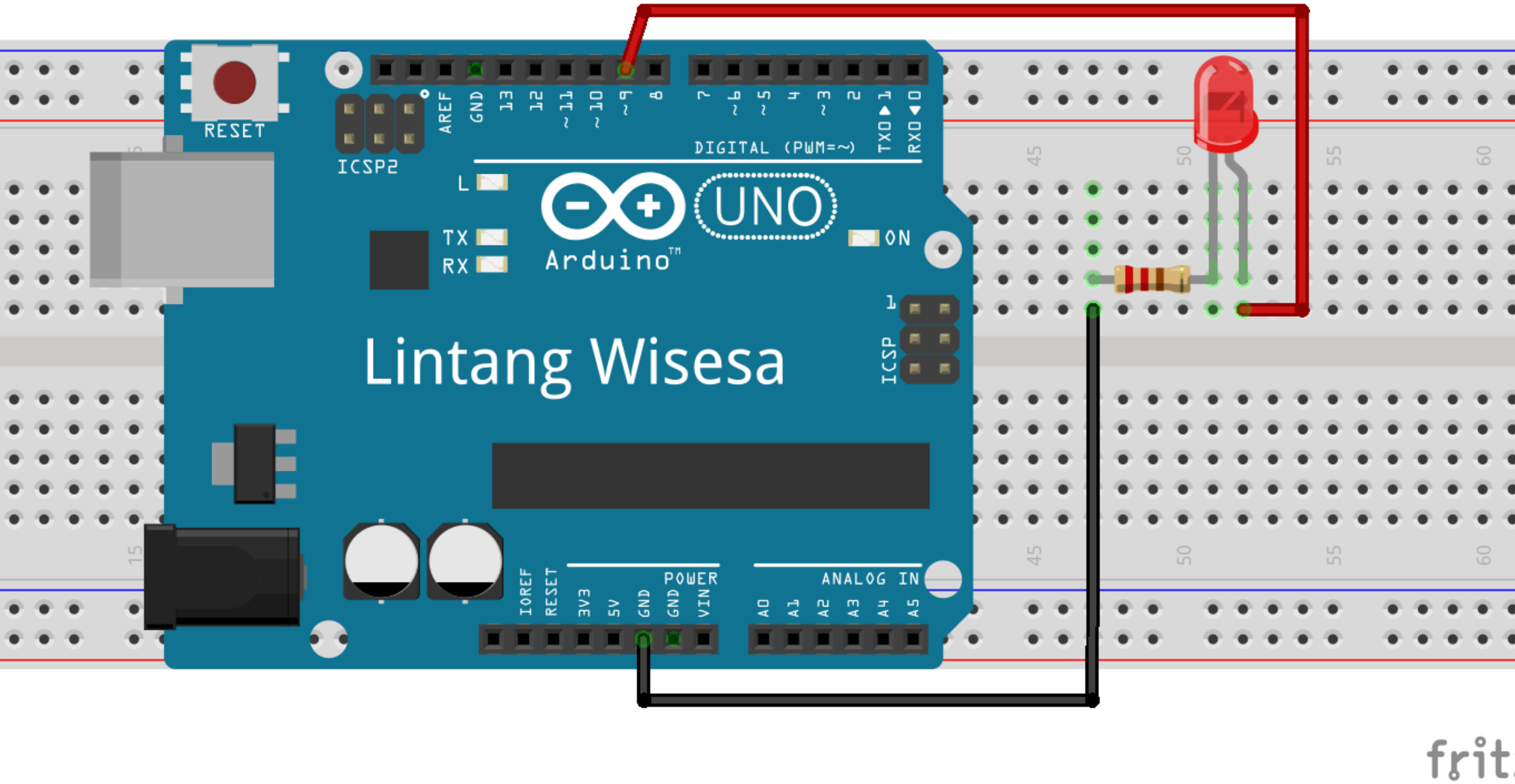


Purwadhika
Startup and Coding School

Traffic Light Miniature

```
void setup() {
pinMode(5, OUTPUT); pinMode(6, OUTPUT);
pinMode(7, OUTPUT);}

void loop() {
digitalWrite(5, HIGH); digitalWrite(6, LOW);
digitalWrite(7, LOW);
  delay(3000);
digitalWrite(5, LOW); digitalWrite(6, HIGH);
digitalWrite(7, LOW);
  delay(3000);
digitalWrite(5, LOW); digitalWrite(6, LOW);
digitalWrite(7, HIGH);
  delay(3000);}
```

PWM on LED

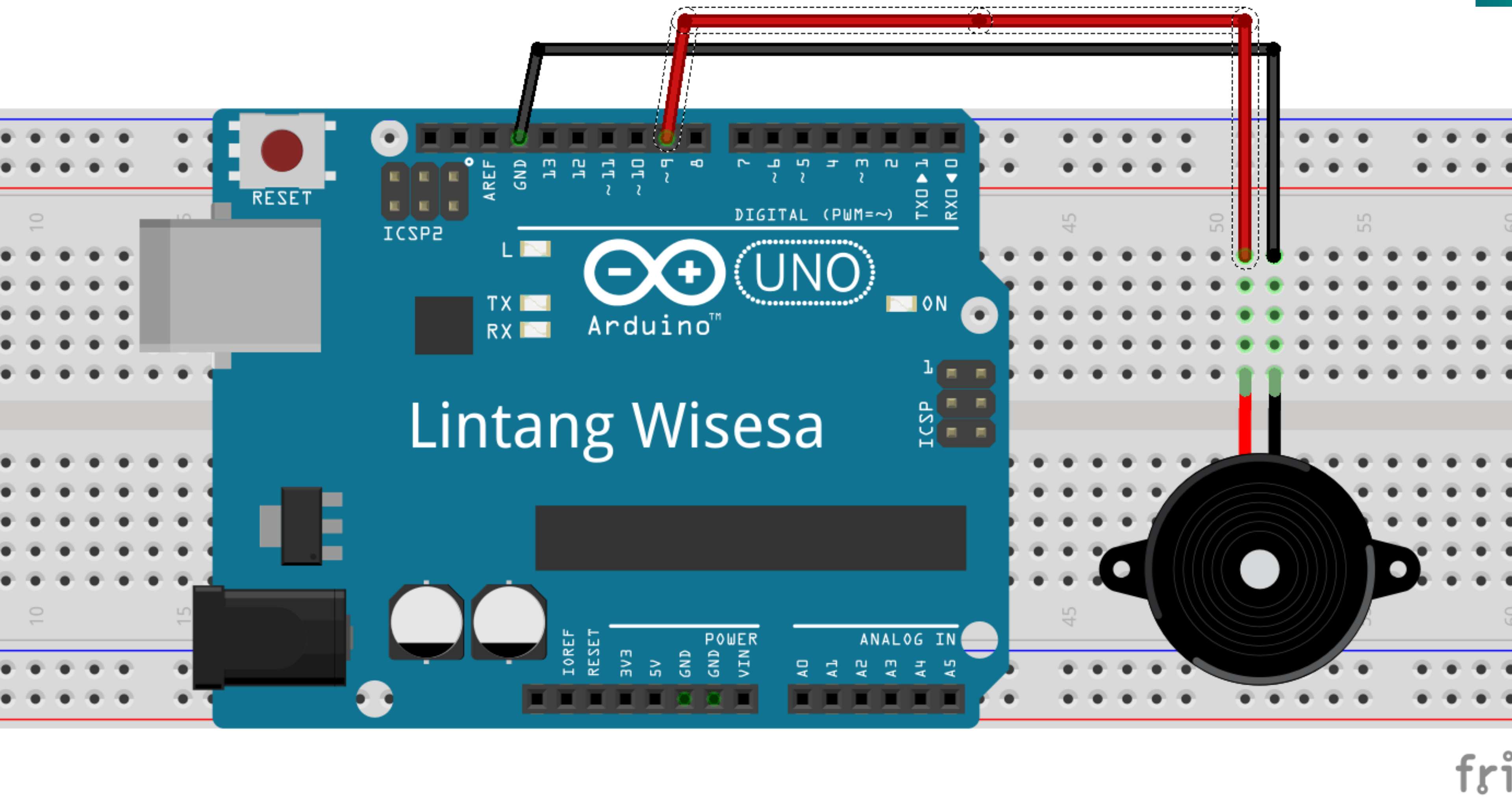


Purwadhika
Startup and Coding School

PWM on LED

```
void setup() {
  pinMode(9, OUTPUT);}

void loop() {
 analogWrite(9, 0); delay(300);
 analogWrite(9, 65); delay(300);
 analogWrite(9, 130); delay(300);
 analogWrite(9, 195); delay(300);
 analogWrite(9, 255); delay(300);
 analogWrite(9, 195); delay(300);
 analogWrite(9, 130); delay(300);
 analogWrite(9, 65); delay(300);
}
```

Blink Buzzer

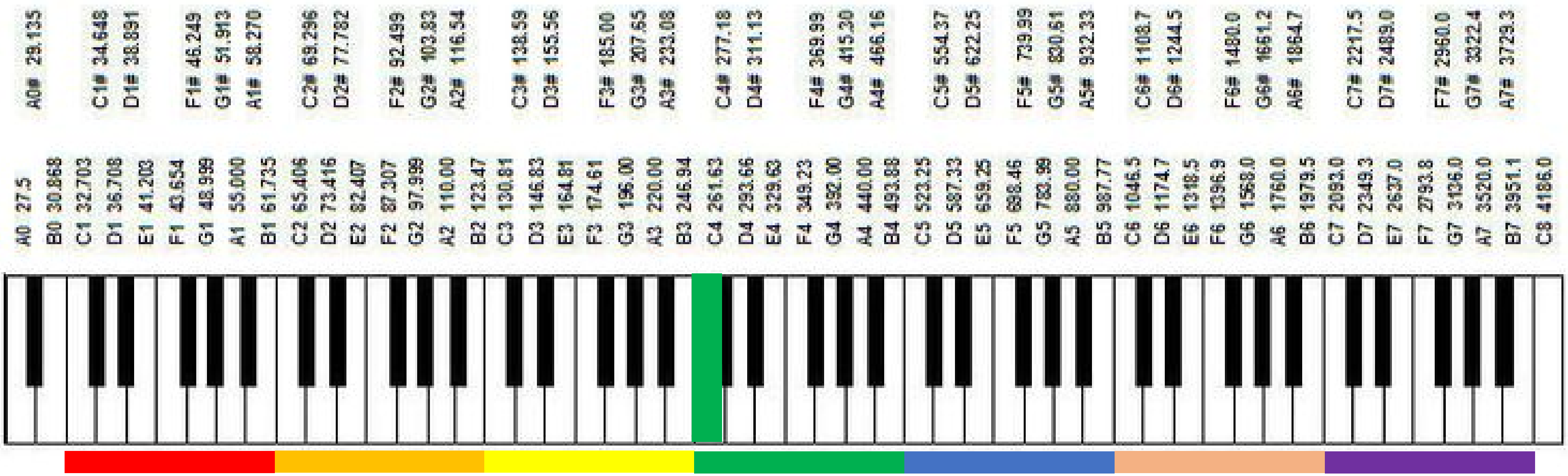


Purwadhika
Startup and Coding School

Blink Buzzer

```
void setup() {
  pinMode(9, OUTPUT);
}

void loop() {
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(1000);
}
```

Buzzer Tones



- `noTone(pin);`
- `tone(pin, freq);`

Purwadhika
Startup and Coding School

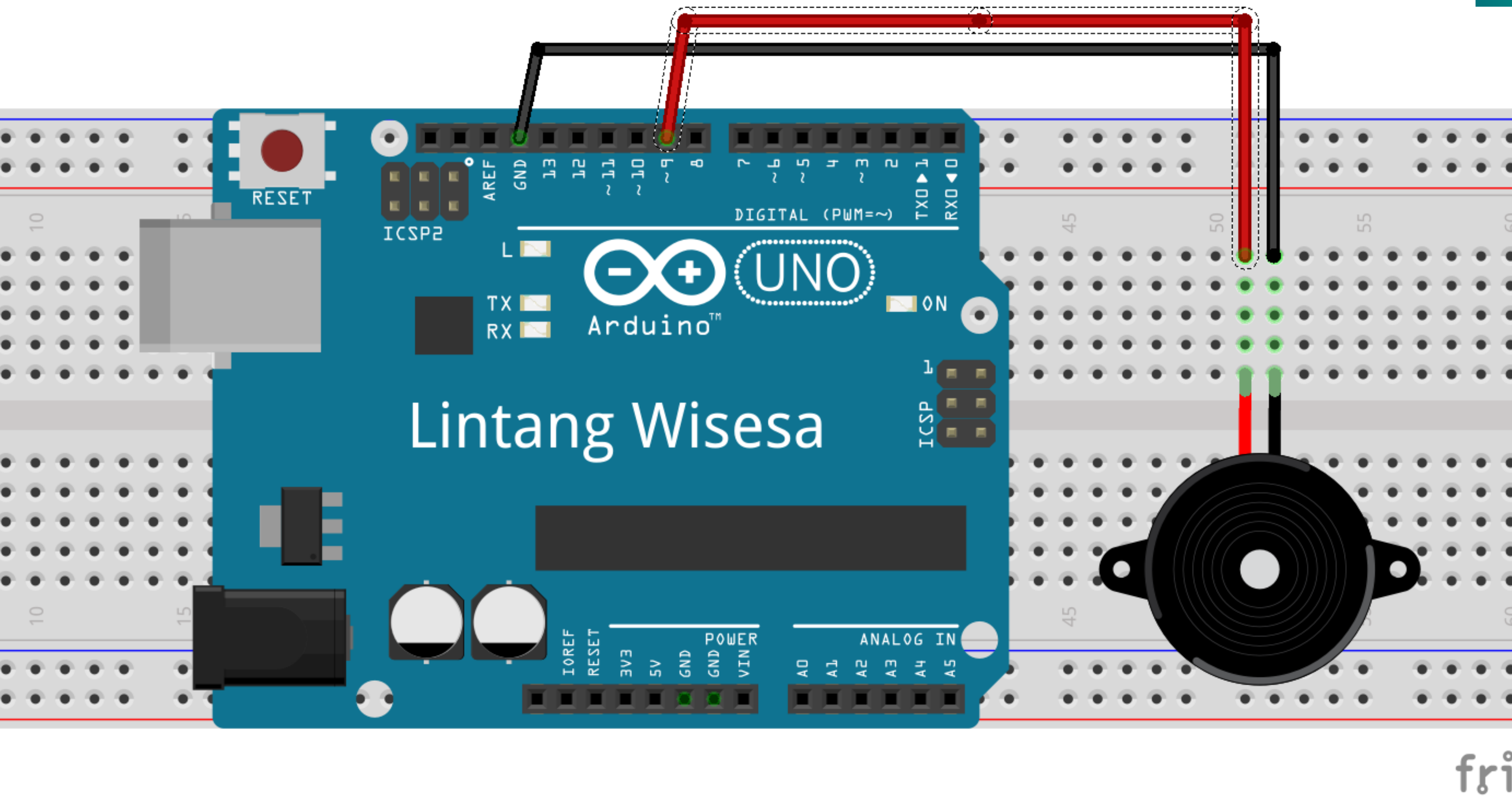
Buzzer Freq Tones

```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
```

```
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
```

```
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
```

Buzzer Tones



Buzzer Tones

```
void setup() {}

void loop() {
 tone(9, 262); delay(1000);
 tone(9, 294); delay(1000);
 tone(9, 330); delay(1000);
 tone(9, 349); delay(1000);
 tone(9, 392); delay(1000);
 tone(9, 440); delay(1000);
 tone(9, 494); delay(1000);
 tone(9, 523); delay(1000);}
```

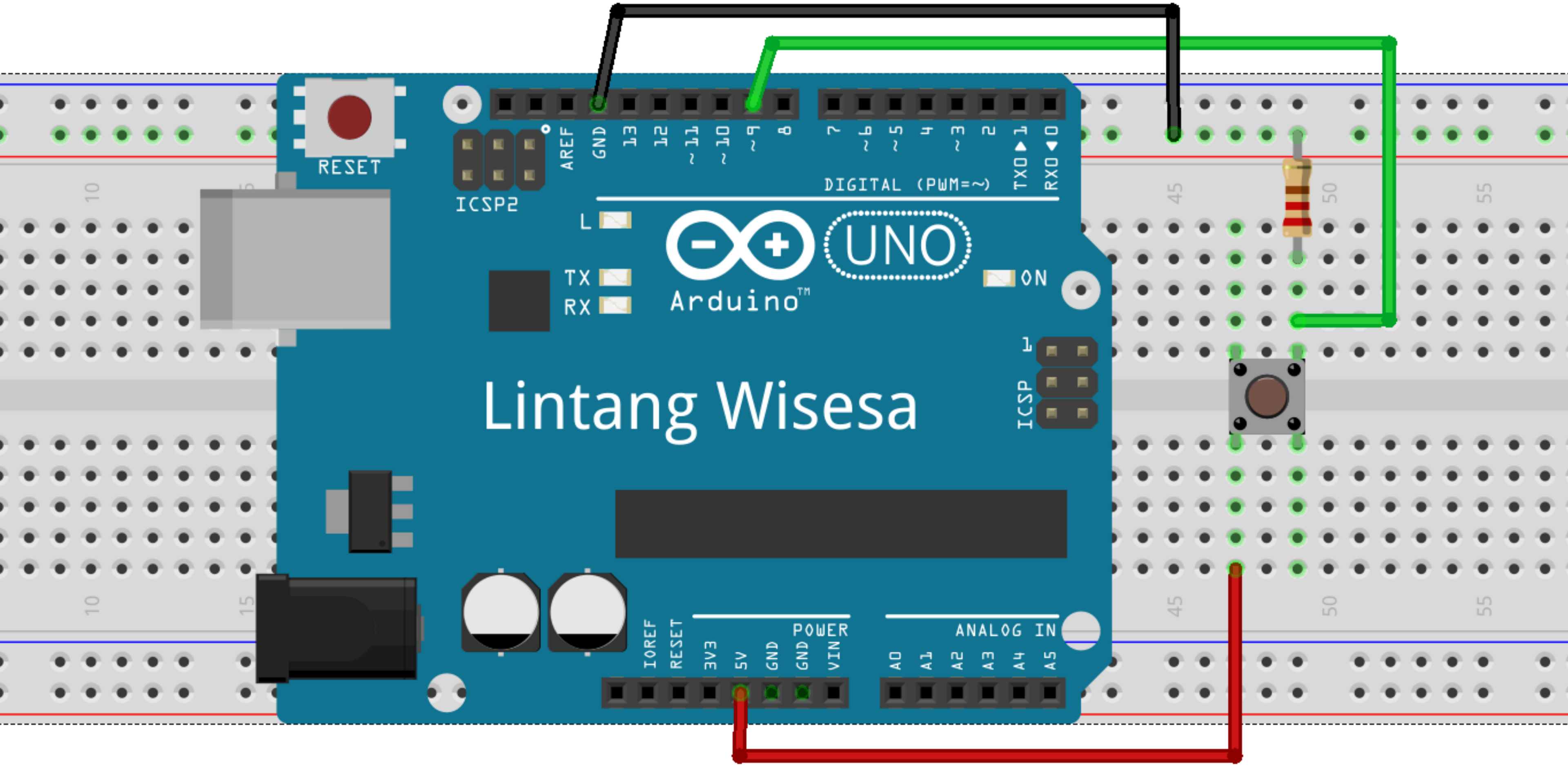
Buzzer Tones

```
int melody[] = {262, 294, 330, 349, 392, 440, 494,
523};

int noteDurations[] = {4, 4, 4, 4, 4, 4, 4, 4};

void setup() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {
  int noteDuration = 1000 / noteDurations[thisNote];
    tone(9, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(9);
  }}

void loop() {}
```

Try It!

Build a simple music tones with a buzzer!

Read Button



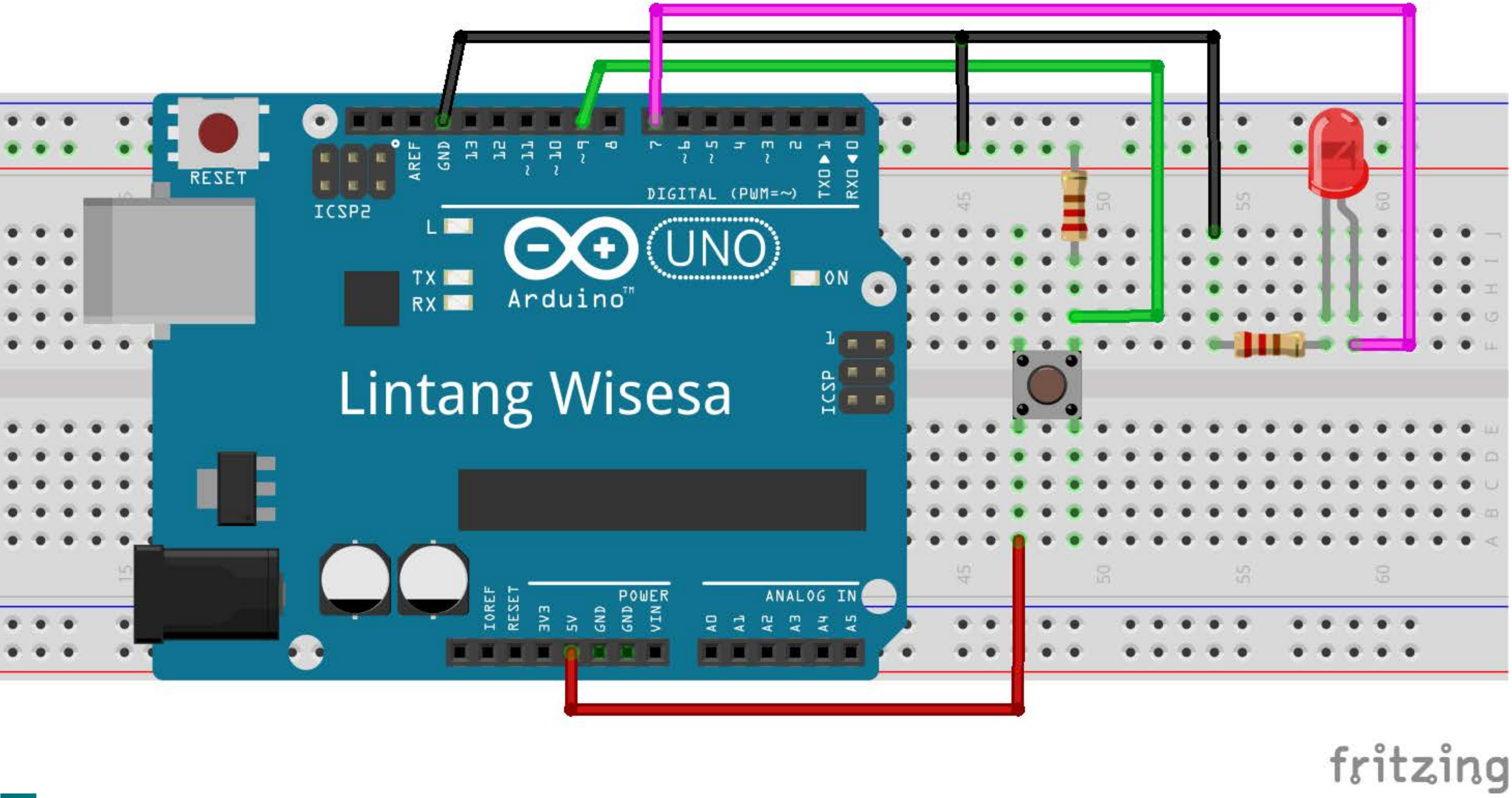
Purwadhika
Startup and Coding School

Read Button

```
void setup() {
  Serial.begin(9600);
  pinMode(9,INPUT);
}

void loop() {
  Serial.println(digitalRead(9));
  delay(100);
}

//Read on Serial Monitor & Serial Plotter
```

Purwadhika
Startup and Coding School

Control LED with a button

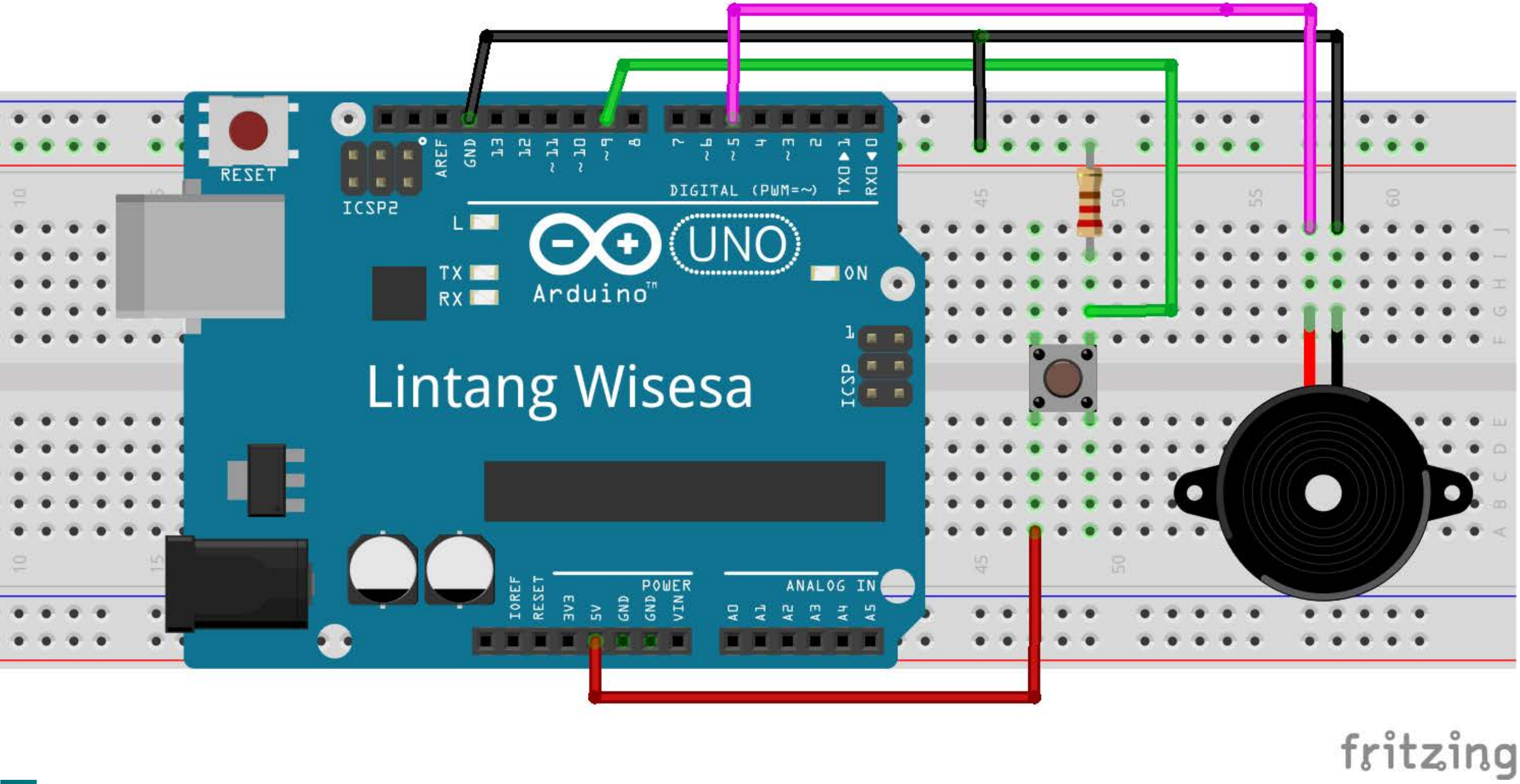


Purwadhika
Startup and Coding School

Control LED with a button

```
void setup() {
  pinMode(7,OUTPUT);
  pinMode(9,INPUT);
}

void loop() {
  int tombol = digitalRead(9);
  if(tombol==1){digitalWrite(7,HIGH);}
  else{digitalWrite(7,LOW);}
}
```

Simple Bell

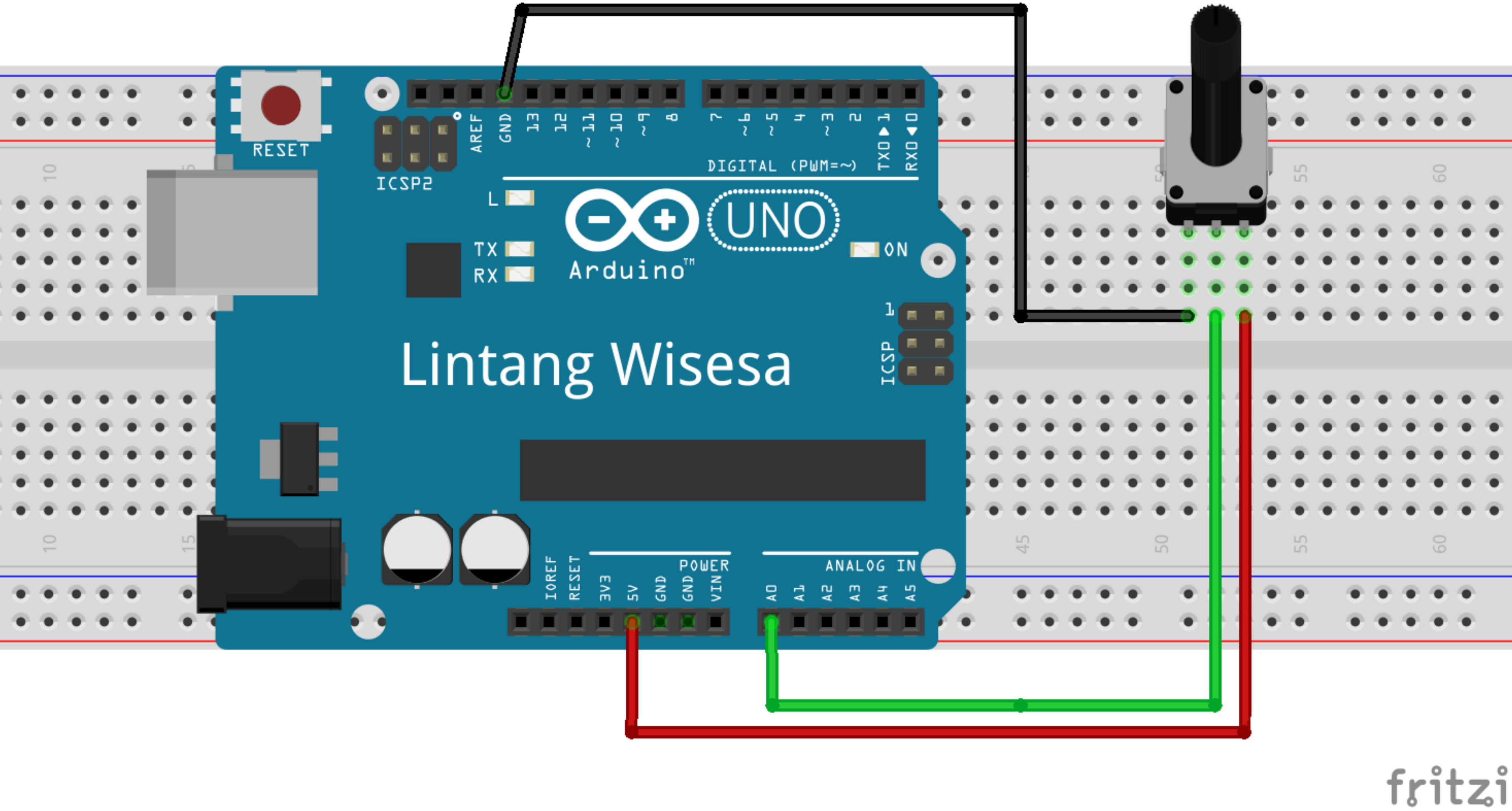


Purwadhika
Startup and Coding School

Simple Bell

```
void setup() {
  pinMode(5,OUTPUT);
  pinMode(9,INPUT);
}

void loop() {
  int tombol = digitalRead(9);
  if(tombol==1){tone(5, 440);}
  else{noTone(5);}
}
```

Read Potentiometer

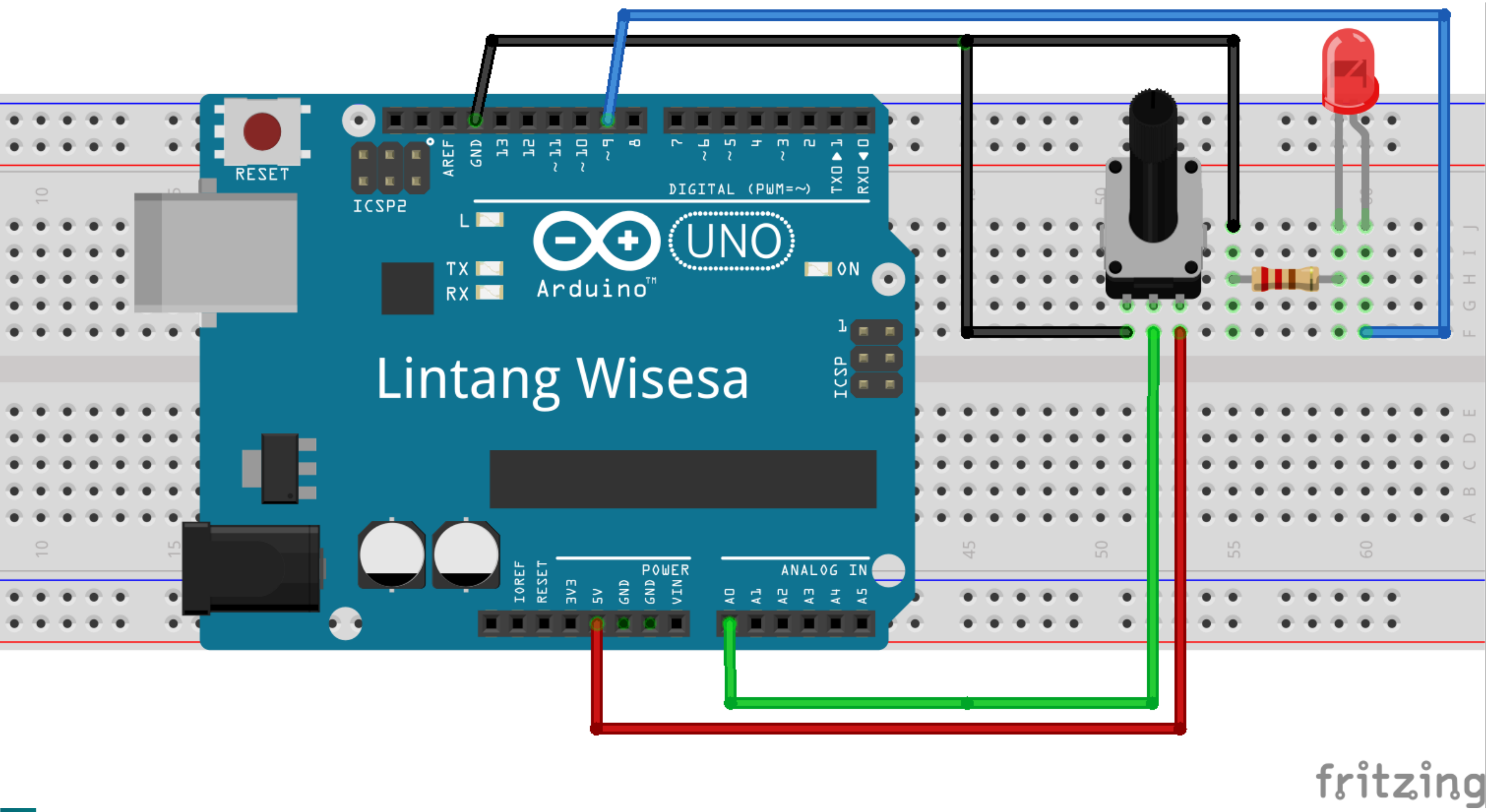


Purwadhika
Startup and Coding School

Read Potentiometer

```
void setup() {
  Serial.begin(9600);}

void loop() {
  int pot = analogRead(A0);
  Serial.println(pot);
  delay(100);}


//Read on Serial Monitor & Serial Plotter
```

Brightness Control

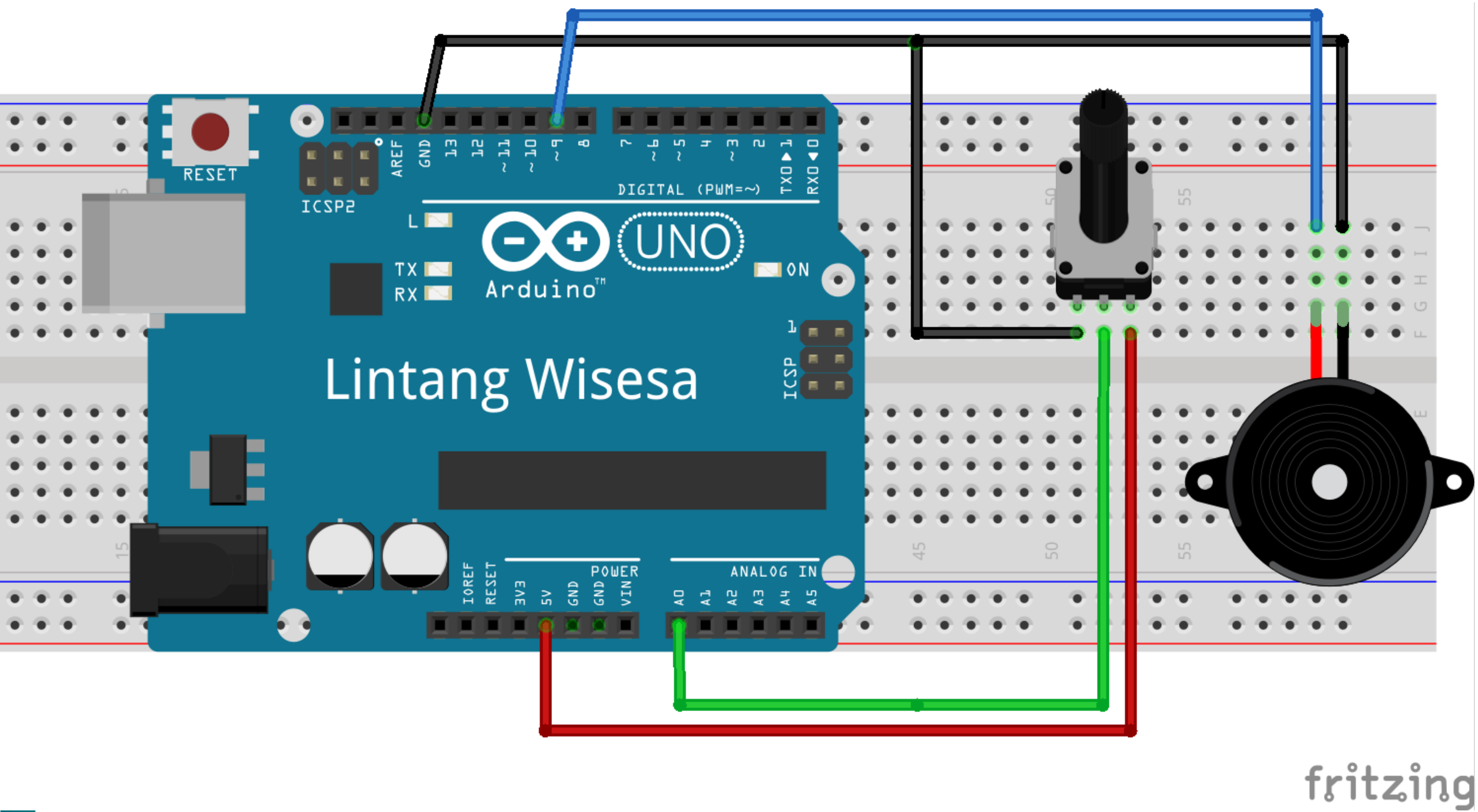


Purwadhika
Startup and Coding School

Brightness Control

```
void setup() {
  pinMode(9, OUTPUT);}

void loop() {
  int pot = analogRead(A0);
  analogWrite(9, pot/4);
  delay(100);}
```

Freq Control



Purwadhika
Startup and Coding School

Freq Control

```
void setup() {
  pinMode(9, OUTPUT);}

void loop() {
  int pot = analogRead(A0);
  if(pot < 45){noTone(9);}
  if(pot > 45){tone(9, pot * 4);}
  delay(100);}
```

Purwadhika
Startup and Coding School