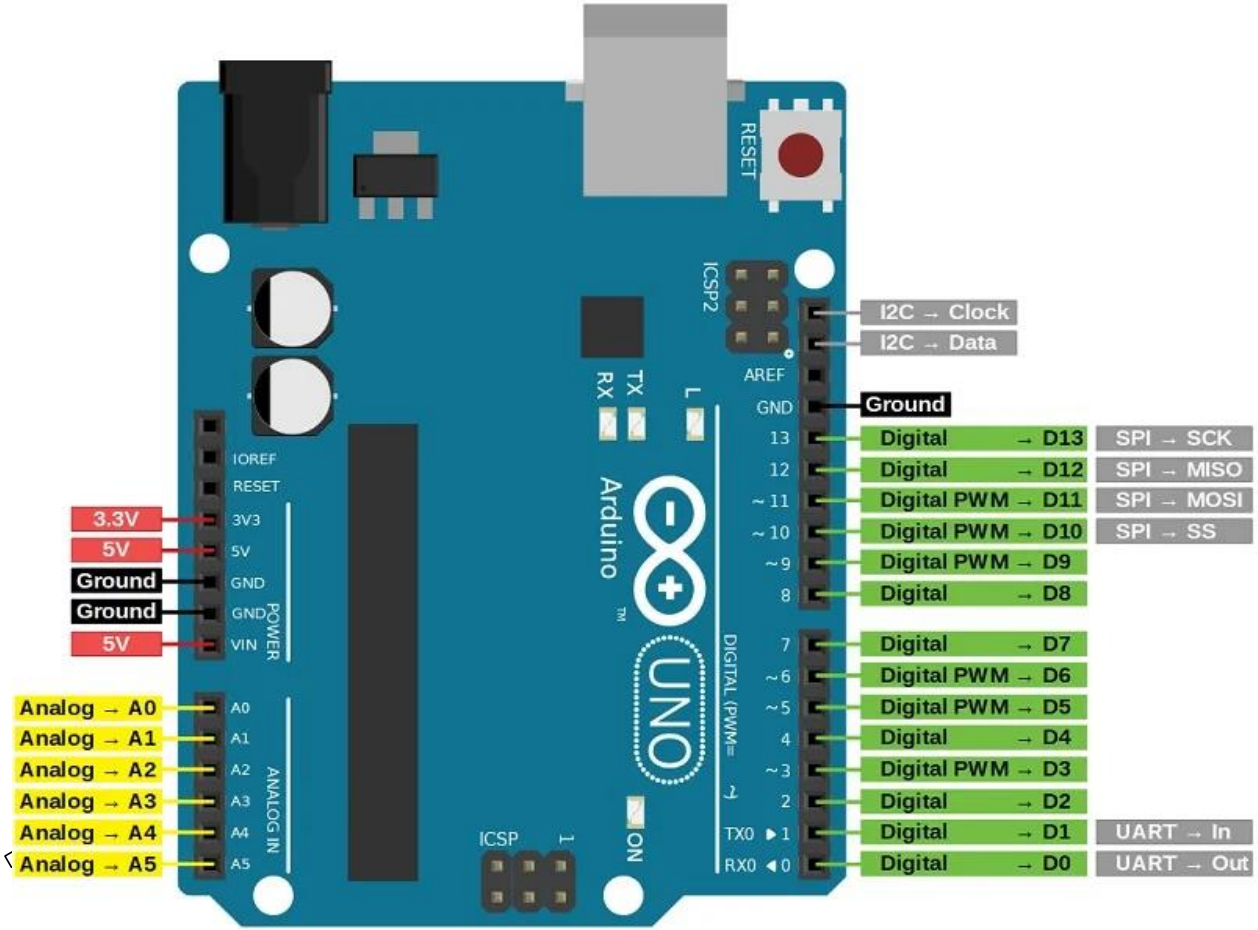
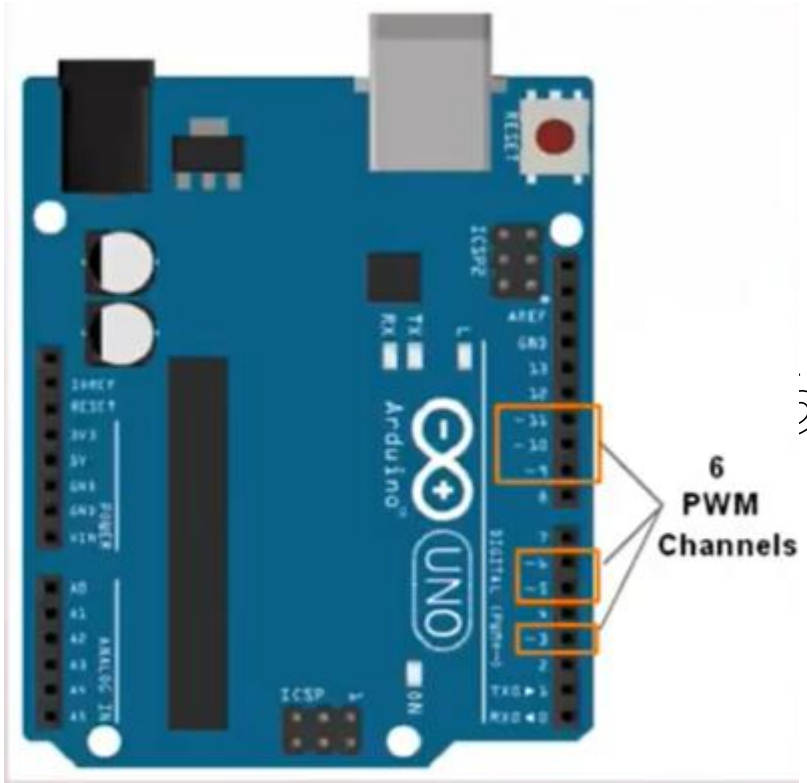
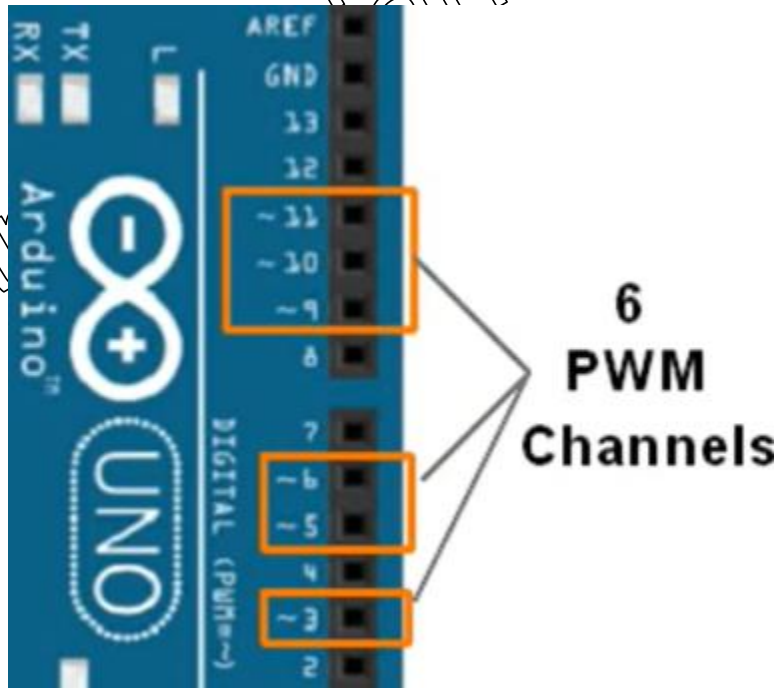
LED_Brightness_Control

with ARDUINO


RESET
ICSP2
I2C → Clock
I2C → Data
AREF
GND
Ground
13
Digital → D13
SPI → SCK
12
Digital → D12
SPI → MISO
~ 11
Digital PWM → D11
SPI → MOSI
~ 10
Digital PWM → D10
SPI → SS
~ 9
Digital PWM → D9
8
Digital → D8
7
Digital → D7
~ 6
Digital PWM → D6
~ 5
Digital PWM → D5
4
Digital → D4
~ 3
Digital PWM → D3
2
Digital → D2
TX0 ▶ 1
Digital → D1
UART → In
RX0 ◀ 0
Digital → D0
UART → Out
DIGITAL (PWM=~)
TX
RX
L
ON
ICSP
1
Arduino
UNO
IOREF
RESET
3.3V
3V3
5V
5V
Ground
GND
Ground
GND
5V
VIN
POWER
Analog → A0
A0
Analog → A1
A1
Analog → A2
A2
Analog → A3
A3
Analog → A4
A4
Analog → A5
A5
ANALOG IN

Arduino PWM Pins





Apparatus

1. PC with Arduino IDE
2. Arduino UNO Board
3. USB cable
4. LED
5. Bread board
6. 10kΩ Potentiometer
7. Jumper wires
8. 220 ohm resistor

Learning Objectives

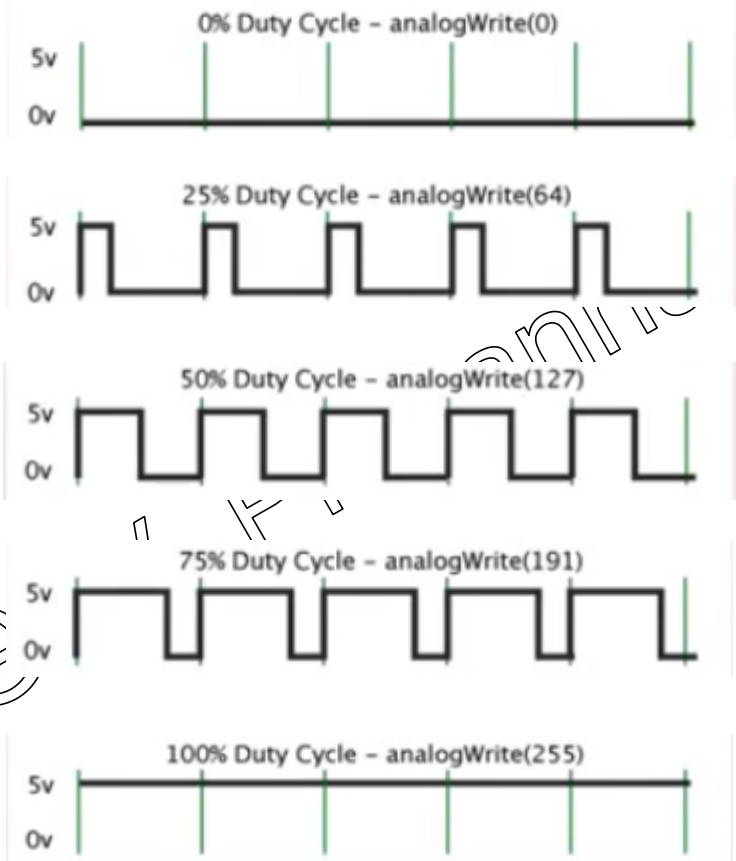
- PWM i.e. Pulse Width Modulation
- PWM Duty Cycle
- PWM Frequency
- L293D Motor Driver IC

Pulse Width Modulation

- PWM is a technique by which the width of the pulse is varied
- It is done while keeping the frequency of wave constant

- PWM signal consists of two main properties that define its behaviour
- They are duty cycle and frequency

Duty Cycle



Duty Cycle Formula

$$\text{duty cycle} = \frac{t_{ON}}{t_{ON} + t_{OFF}}$$

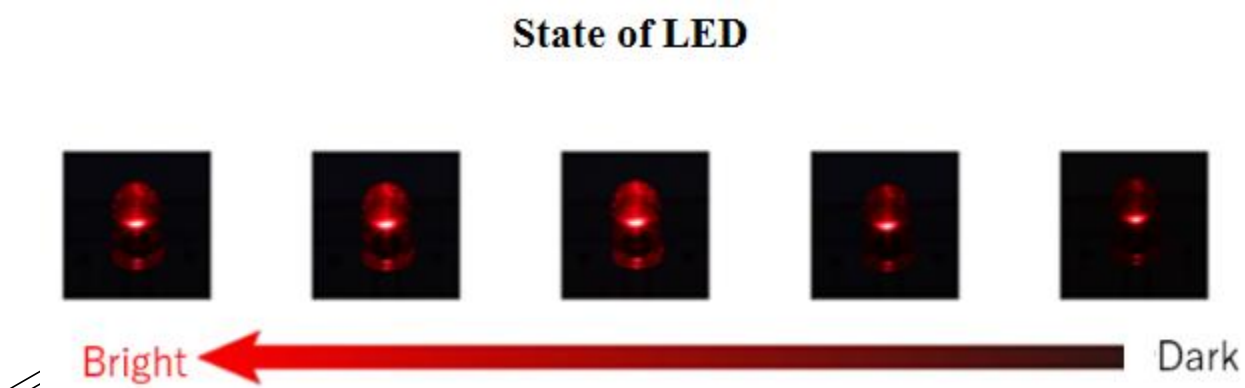
t_{ON} = ON time

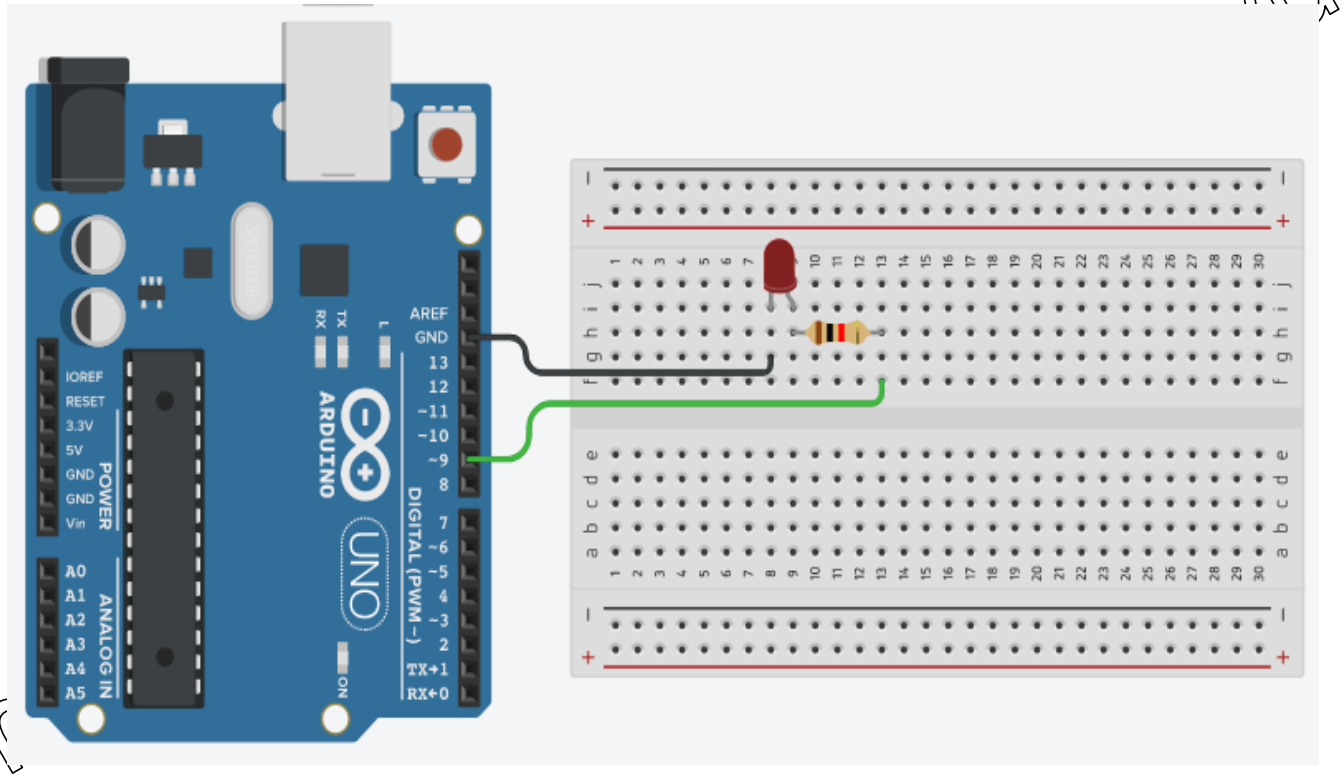
t_{OFF} = OFF time

$t_{ON} + t_{OFF}$ = Time period

Example - Duty Cycle

- We will perform one simple experiment by varying duty cycle
- This will control brightness of LED





LED_Brightness_Control bY Dr. GVP

Program

– – –

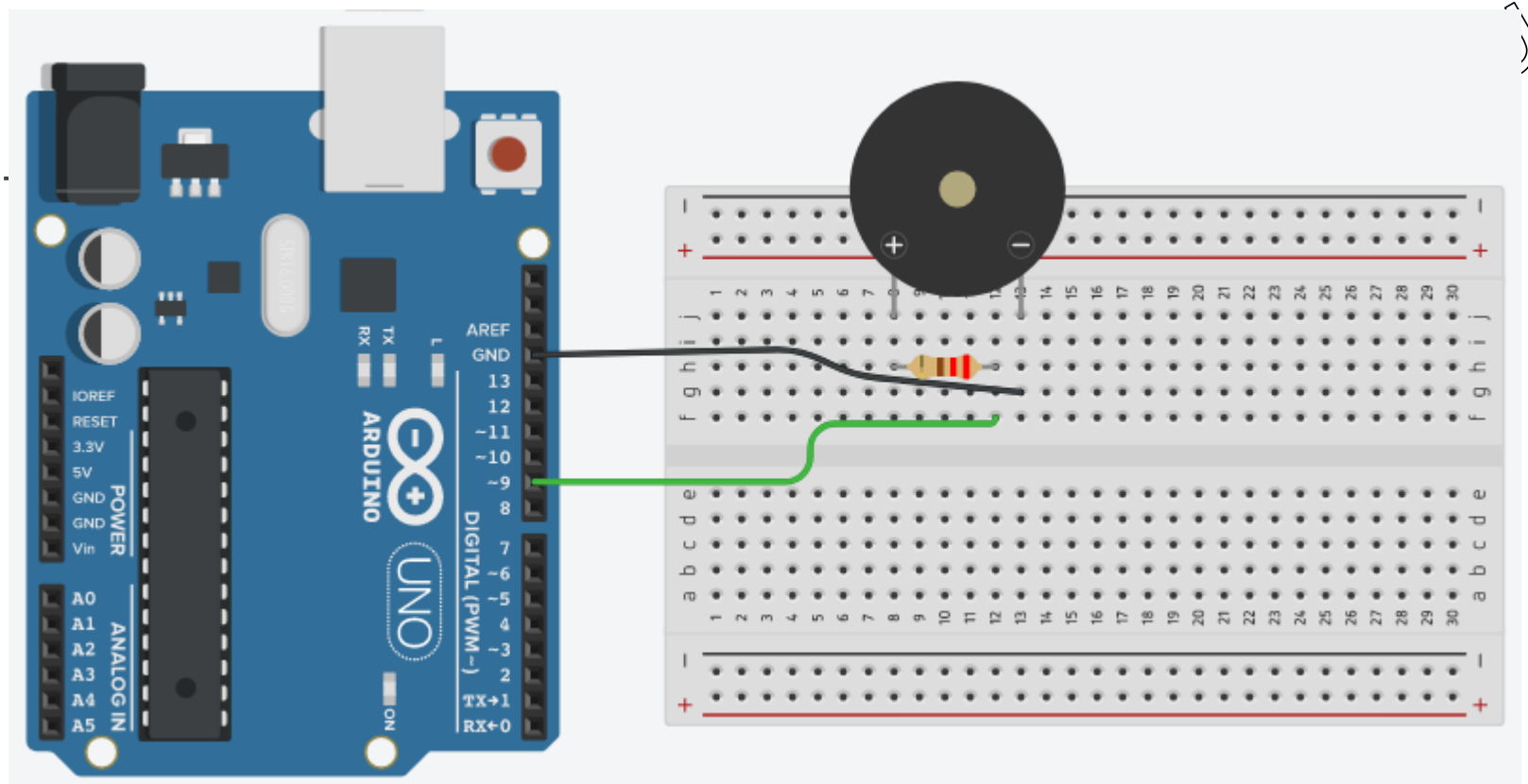
```
// Brightness_LED_CONTROL
int dutyCycleValue=1;
void setup()
{
  pinMode(9, OUTPUT); //here, 9 is one of PWM pins
}

void loop()
{
 if(dutyCycleValue<=255)
 {
  analogWrite(9, dutyCycleValue);
  delay(1000); // Wait for 1000 millisecond(s)
  dutyCycleValue++;
 }
}
```

Assignment

Repeat same instead of LED use Buzzer(piezo)

Then Buzzer sound will vary from minimum to maximum



LED_Brightness_Control bY Dr. GVP