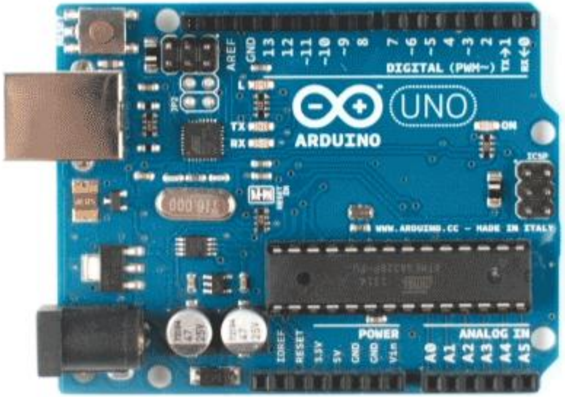
ELECTRONICS CLUB
DHAPAKHEL, LALITPUR

Kantipur Engineering College
Kathmandu 1998

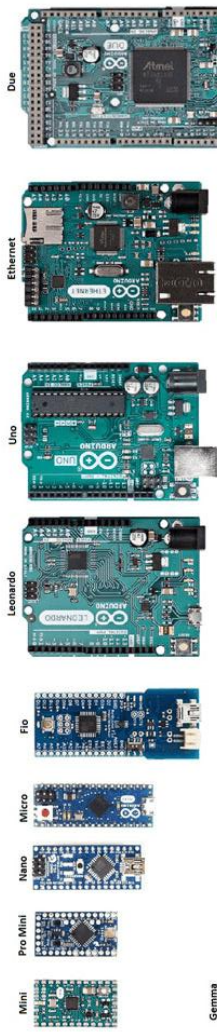
Arduino Workshop

Introduction to Microcontroller Programming and Sensor Interfacing



Objectives

- ❑Understand Arduino basics
- ❑Learn about sensors and actuators
- ❑Get hands-on experience with Arduino IDE and components
- ❑Connection of sensors
- ❑Learn about PWM
- ❑Explore communication protocols (I2C, SPI, UART)
- ❑About Interrupts



What Is An Arduino?

❑ Open-source electronics platform for interactive projects.

❑ Combines hardware (microcontroller) and software (IDE).

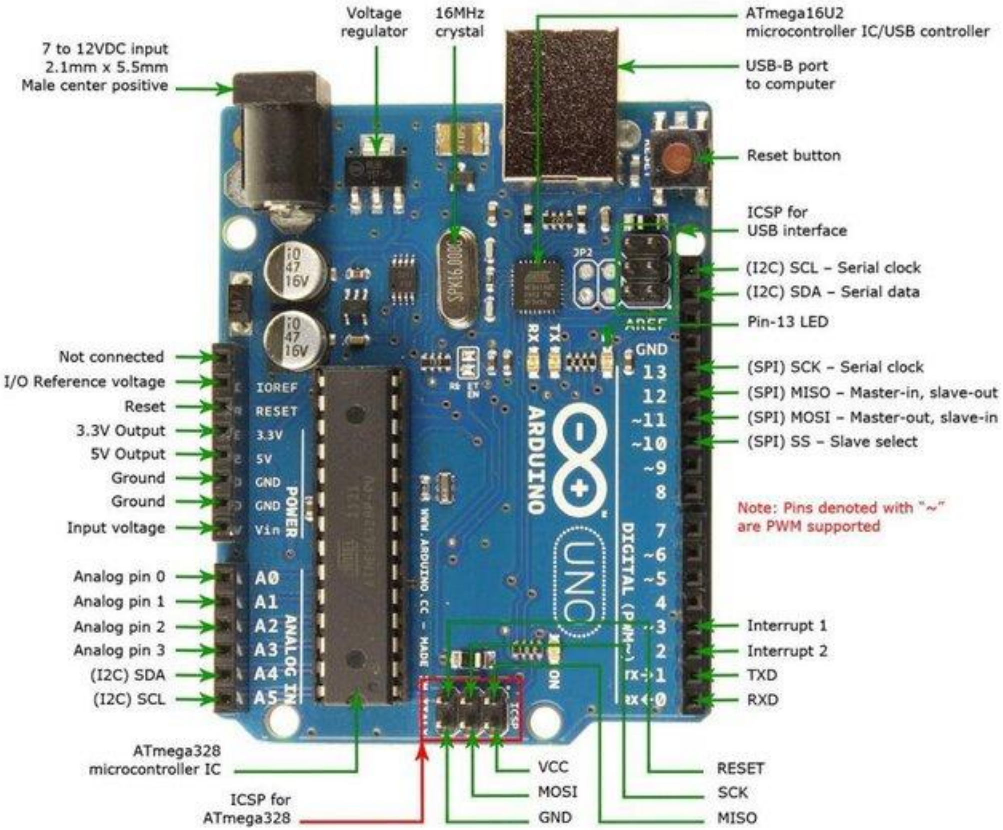
❑ Arduino Uno uses ATmega328P (16 MHz, 32 KB Flash, 2 KB SRAM).

❑ Features 14 digital and 6 analog I/O pins.

❑ Supported by a large global community.

❑ Ideal for beginners, students, and DIY projects.

Arduino Board Overview

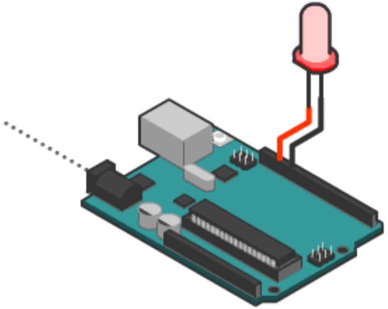


Arduino IDE

- Free, open-source software to write, compile, and upload code.
- Supports C/C++ with built-in libraries.
- Code follows a simple structure:
  - setup() – runs once to initialize.
  - loop() – runs continuously after setup.
- Serial Monitor allows real-time data display and debugging.
- Works on Windows, macOS, and Linux.

Upload your first sketch: "Blink"

- ❑ Open Arduino IDE.
- ❑ Click on File > Examples > 01.Basics > Blink.



Common Arduino Functions

- ❑ **pinMode(pin, mode)** – Sets pin as INPUT, OUTPUT, or INPUT_PULLUP.
- ❑ **digitalWrite(pin, value)** – Writes HIGH or LOW to a digital pin.
- ❑ **analogWrite(pin, value)** – Sends PWM signal (0–255) to pin.
- ❑ **Serial.begin(baud)** – Starts serial communication.
- ❑ **delay(ms)** – Pauses program for given milliseconds.
- ❑ **digitalRead(pin)** – Reads value (HIGH or LOW) from a digital pin.
- ❑ **analogRead(pin)** – Reads analog voltage (0–1023) from analog pin.
- ❑ **Serial.print() / Serial.println()** – Sends data to Serial Monitor.

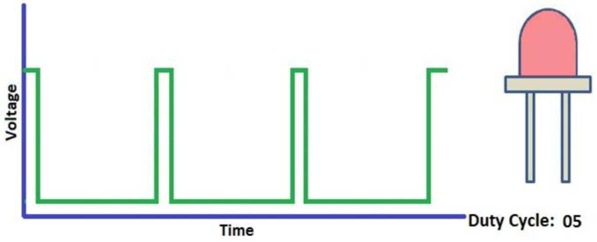
PWM(Pulse Width Modulation)

❑PWM: Technique to create analog-like output using digital pins

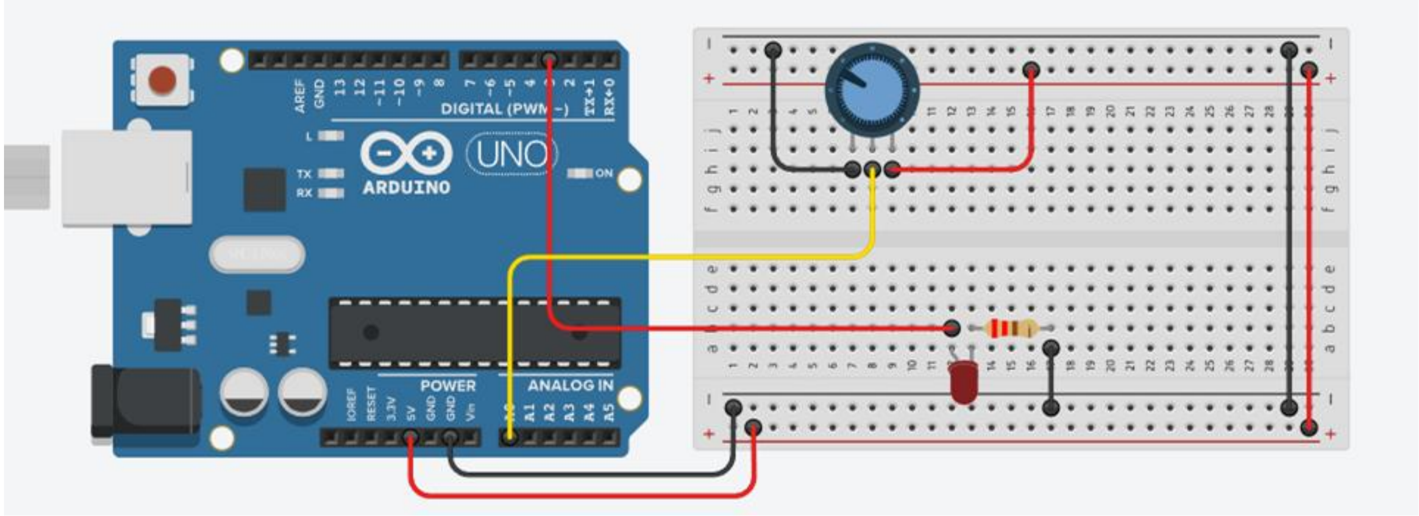
❑Uses analogWrite(pin, value) where value = 0–255

❑Common on pins 3, 5, 6, 9, 10, 11 (marked ~), based on duty cycle.

❑Example: LED dimming or motor speed control



Led dimmer with PWM



❑ For code github.

What are Sensors?

❑Devices that detect physical input (temperature, light, humidity, etc.) and convert it to electrical signal

❑**Analog Sensors**: Provide a range of values (e.g., TMP36, LDR)

❑**Digital Sensors**: Provide HIGH/LOW output (e.g., DHT11, IR switch)

❑Application examples: automation, health, weather monitoring

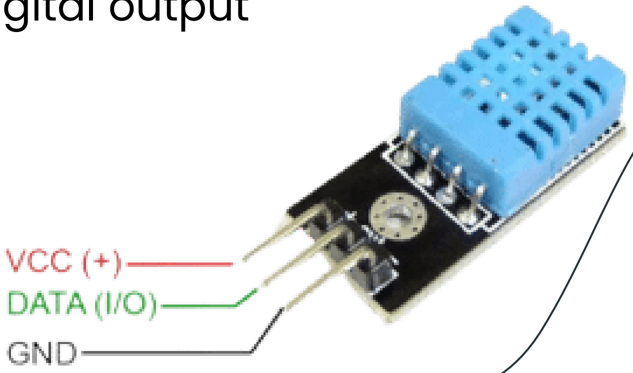
Interfacing DHT11 Sensor

- ❑ **DHT11**: Measures temperature & humidity, digital output
- ❑ **Wiring**:

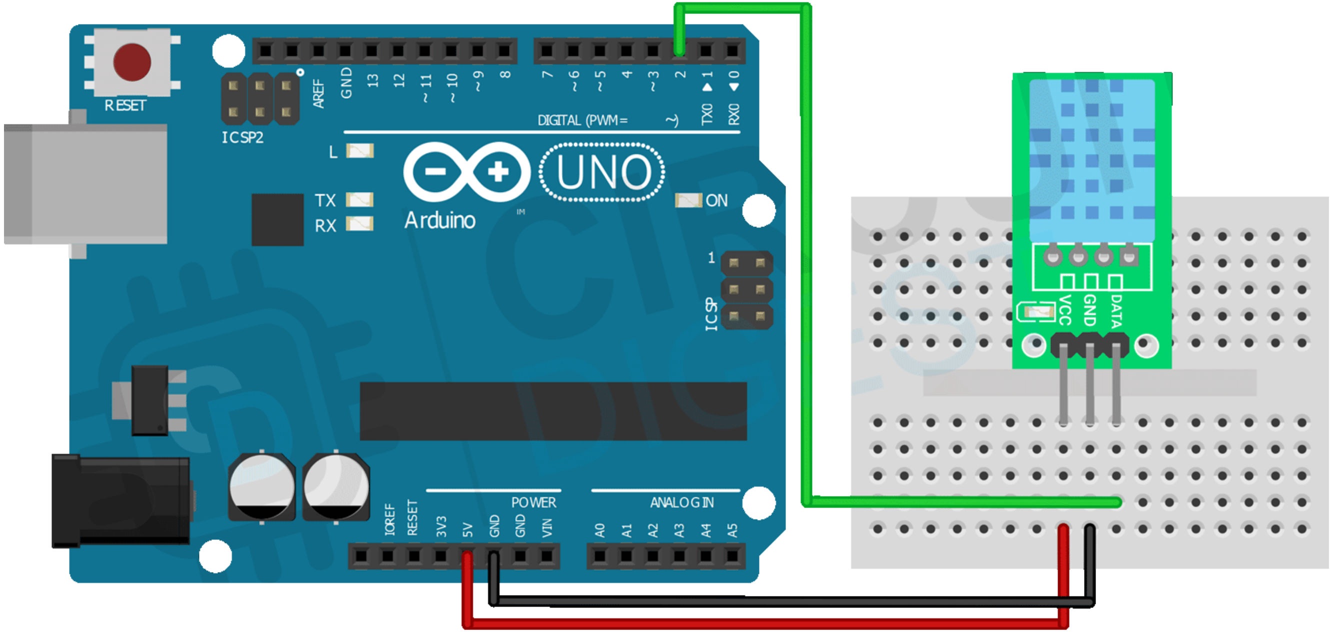
  VCC → 5V

  GND → GND

  Data → Digital pin (e.g., 2)
- ❑ Install library **DHT11 by Dhruba Saha**



Task 3: DHT11 Interfacing



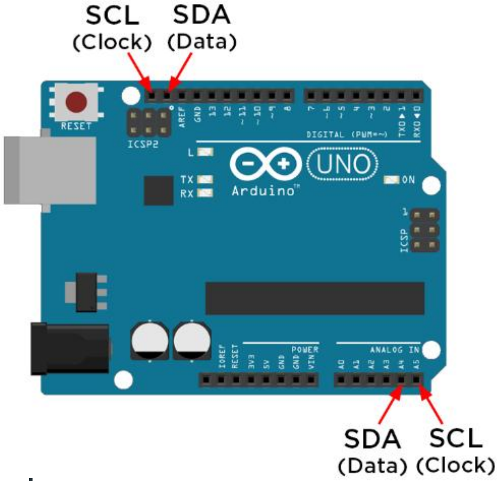
- ☐ For code github.

Communication Protocols

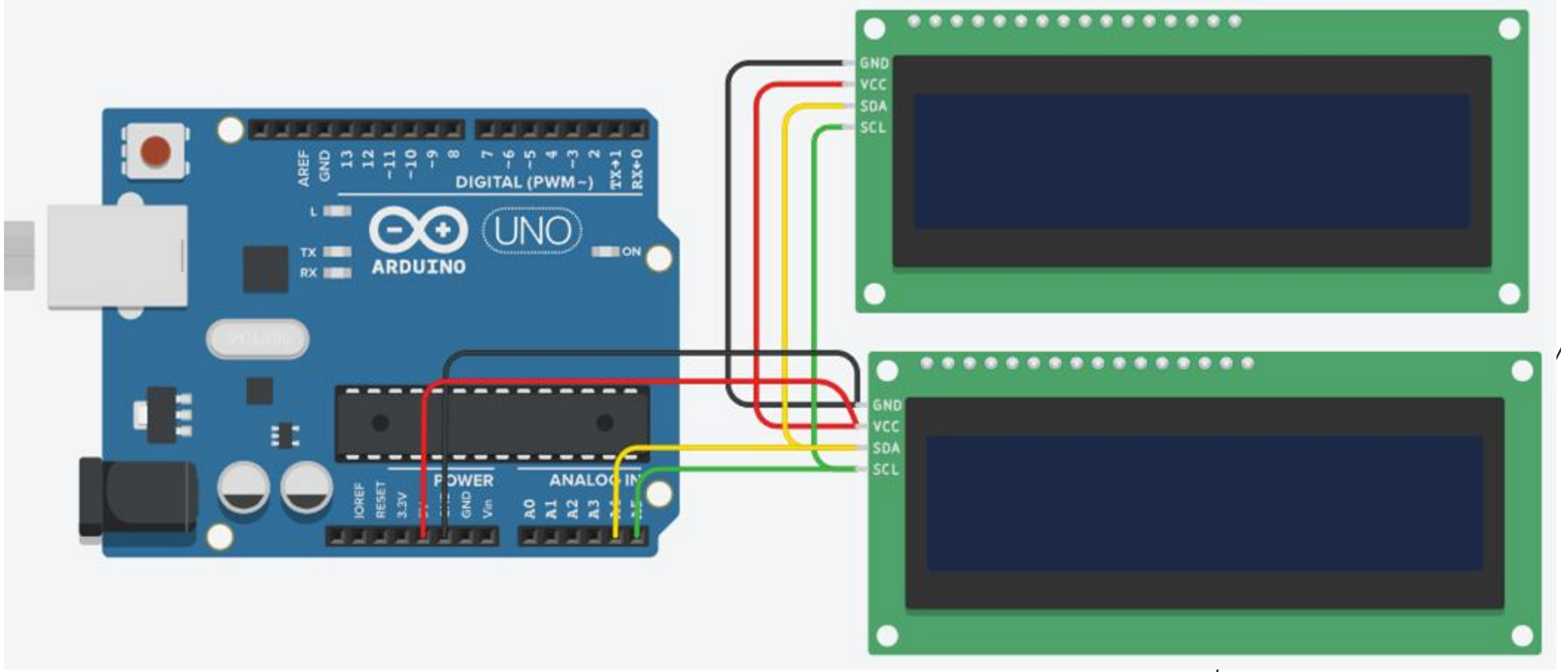
- **I2C**: 2-wire, used for sensors like TC74, OLED displays
- **SPI**: Faster, used for memory cards, some sensors
- **UART (Serial)**: For serial communication via USB or between boards

I2C Protocol

- ❑ **Full Form:** Inter-Integrated Circuit (I²C).
- ❑ **Type:** Synchronous, serial communication protocol.
- ❑ **Wires Used:** Only 2 lines – **SDA** (data) and **SCL** (clock).
- ❑ **Master-Slave Architecture:** One master controls one or multiple slaves.
- ❑ **Addressing:** Each slave has a unique 7-bit address.
- ❑ **Speed Modes:**
  - ➢ **Standard Mode:** 100 kbps
  - ➢ **Fast Mode:** 400 kbps
  - ➢ **High-Speed Mode:** 3.4 Mbps
- ❑ **Data Transfer:** Data sent in bytes (8 bits) with an ACK/NACK signal.
- ❑ **Advantages:** Simple wiring, supports multiple devices.
- ❑ **Disadvantages:** Low Bandwidth



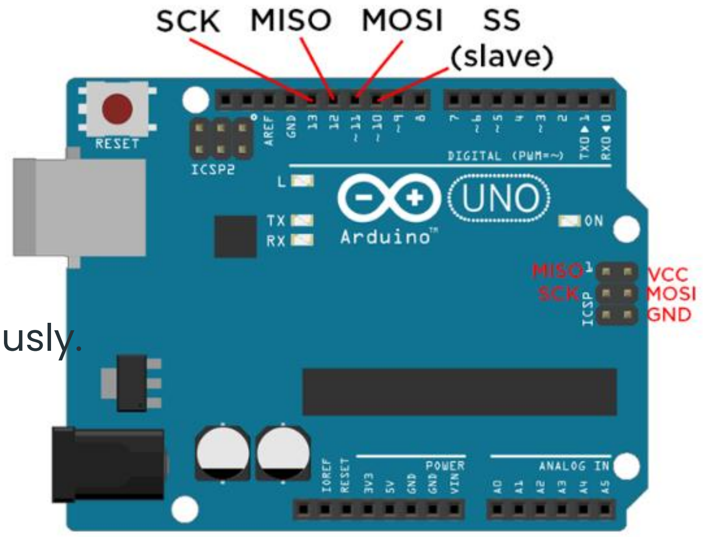
I2C Protocol



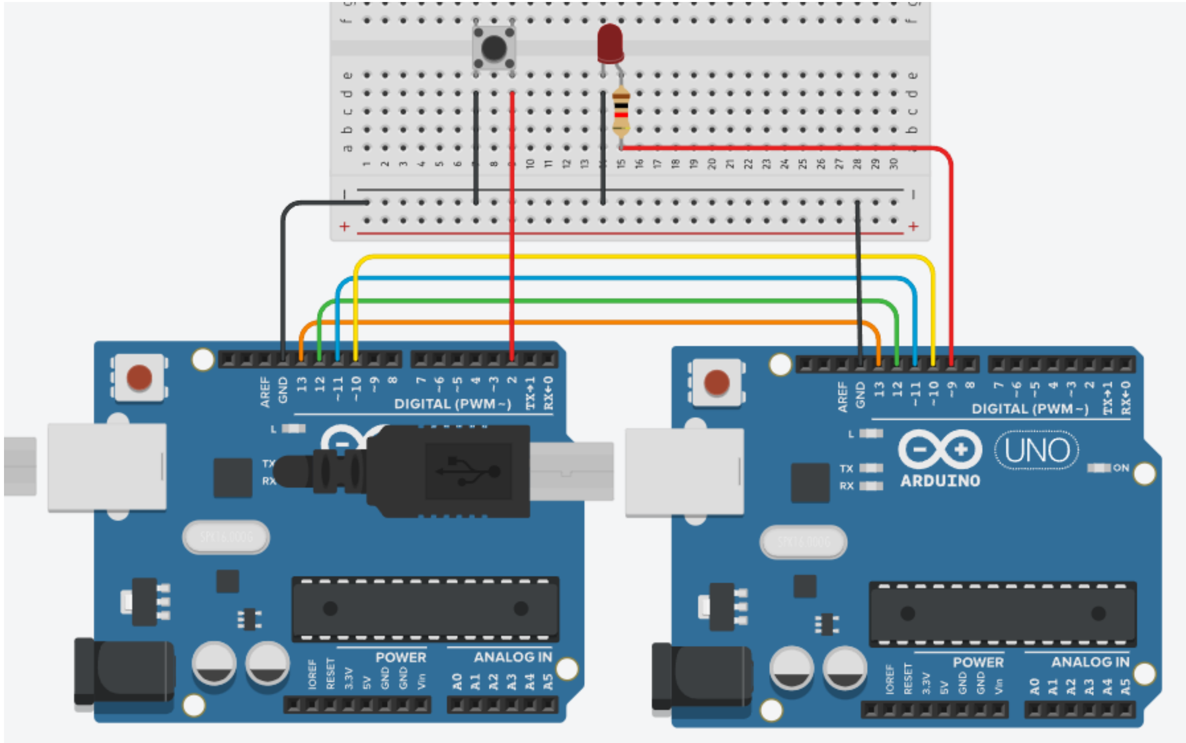
❑ For code github.

SPI Protocol

- **Full Form:** Serial Peripheral Interface (SPI).
- **Type:** Synchronous, serial communication protocol.
- **Wires Used:**
  - **MOSI:** Master Out Slave In
  - **MISO:** Master In Slave Out
  - **SCK:** Clock
  - **SS/CS:** Slave Select/Chip Select
- **Architecture:** One master can control multiple slaves.
- **Full-Duplex:** Data can be sent and received simultaneously.
- **Speed:** Faster than I²C (up to tens of Mbps).
- **No Addressing:** Each slave is selected using a separate SS/CS pin.
- **Data Frame:** 8-bit or more, configurable clock polarity (CPOL) and phase (CPHA).
- **Advantages:** High speed, full-duplex, simple protocol.



SPI Protocol

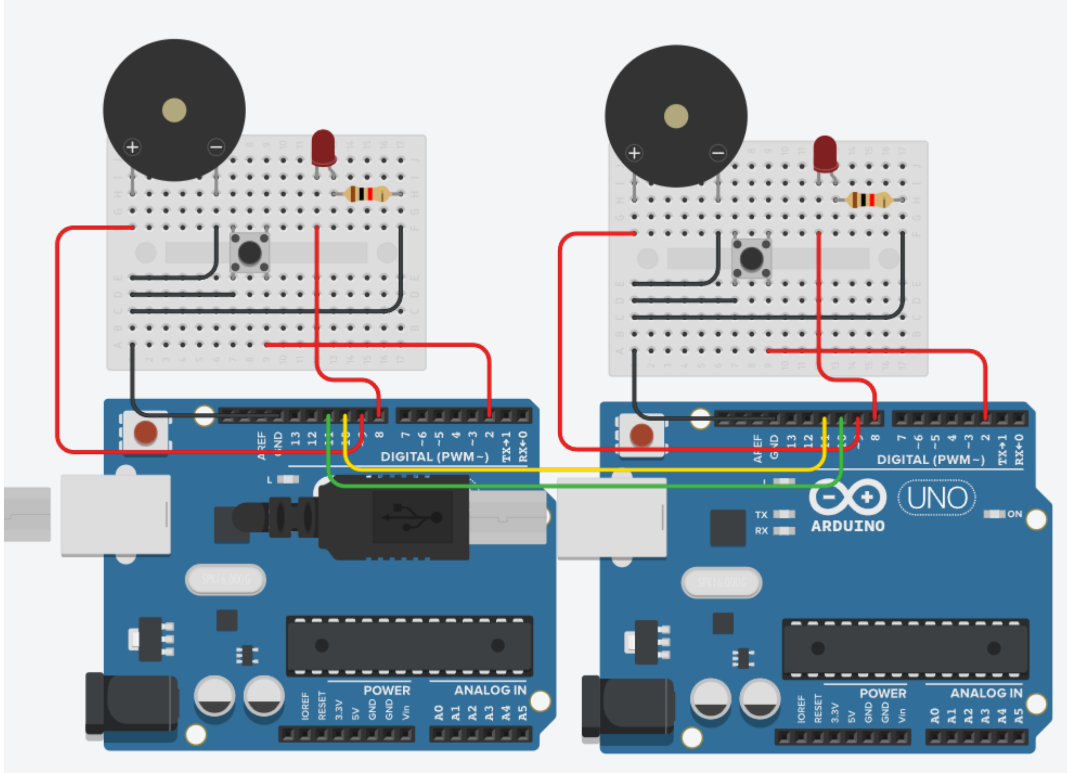


□ For code github.

UART Protocol

- **Full Form:** Universal Asynchronous Receiver/Transmitter.
- **Type:** Asynchronous, serial communication protocol.
- **Wires Used:** Only TX (transmit) and RX (receive).
- **No Clock Line:** Uses start and stop bits for synchronization.
- **Communication:** Point-to-point (only two devices).
- **Data Frame:** Start bit, data bits (5–9), optional parity bit, stop bits.
- **Speed (Baud Rate):** Common values – 9600, 115200 bps, etc.
- **Half-Duplex:** Data is sent one way at a time per line pair.
- **Advantages:** Simple, low-cost, widely supported.
- **Disadvantages:**
    - Limited to two devices only.
    - Slower compared to SPI and I^{2}C.

UART Protocol



- ❑ For code github.

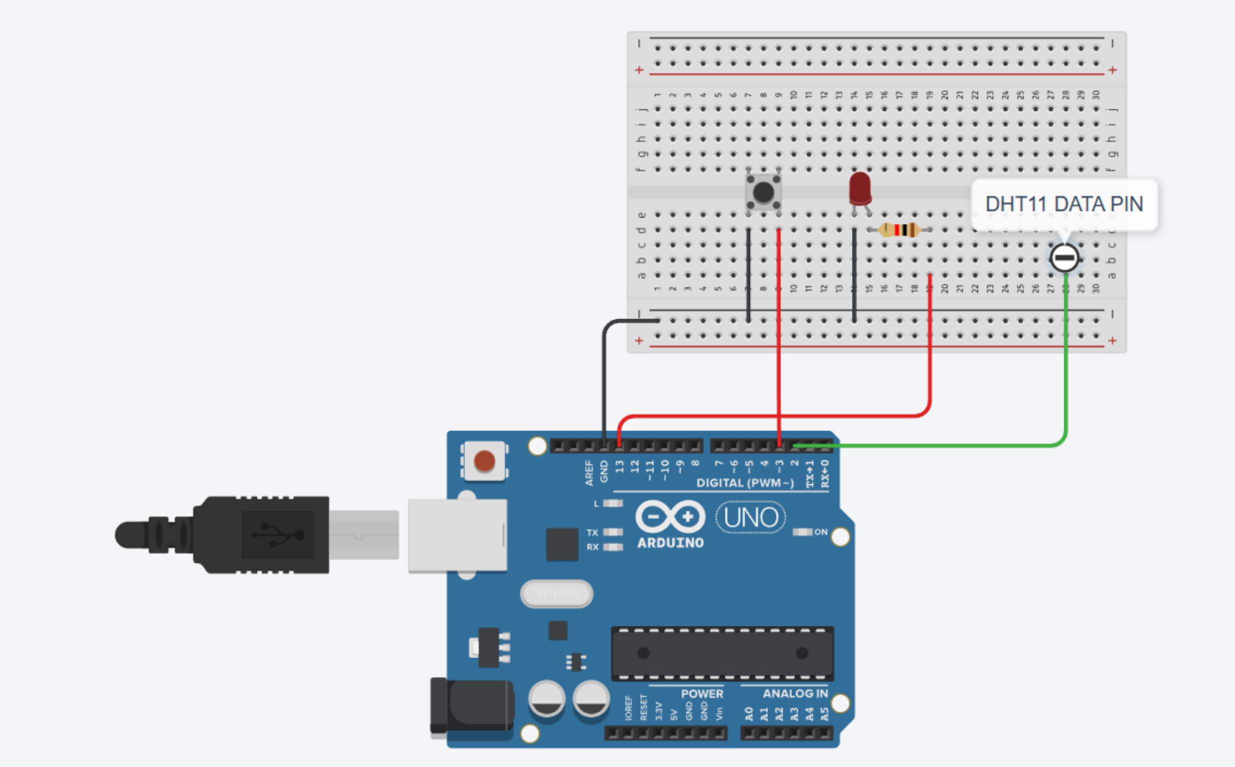
Interrupts in Arduino

❑ **Polling**: Constantly checks for a condition (inefficient)

❑ **Interrupts**: Respond immediately to events (e.g., button press)

❑ **Types:** External (attachInterrupt), Timer (millis(), ISR)

Interrupts in Arduino



- For code github.

THE END