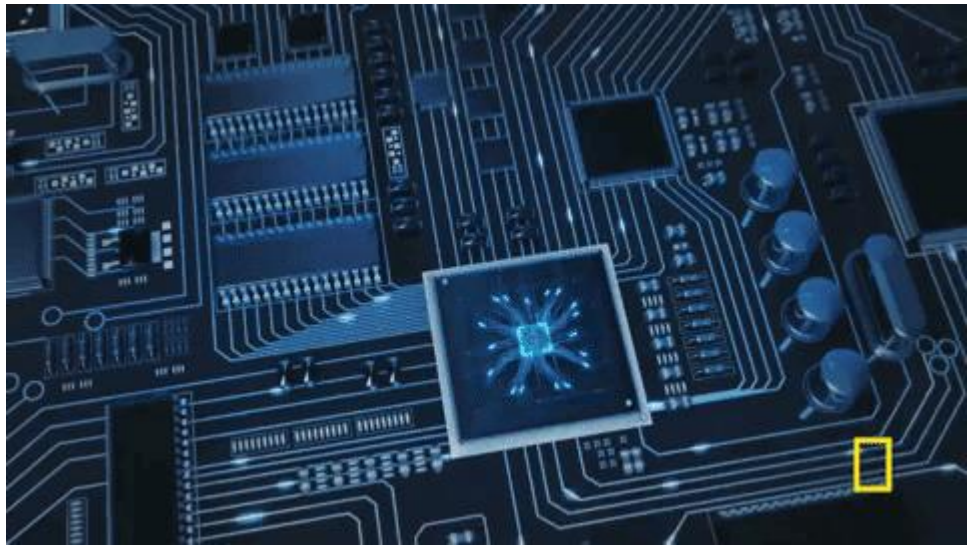
ATME College of Engineering

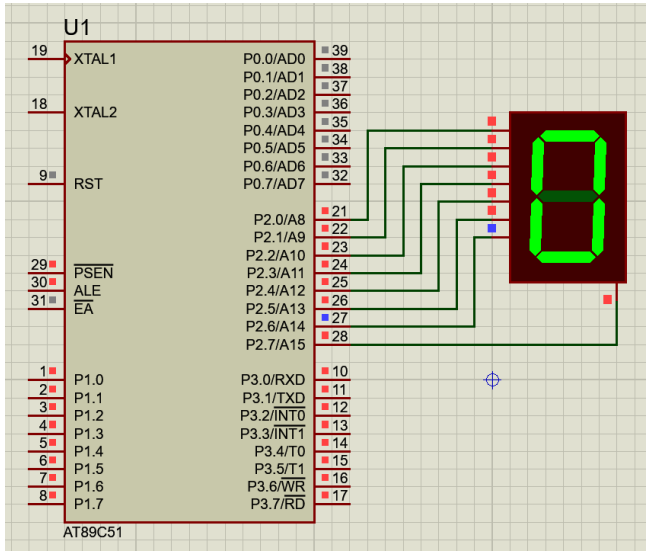
Department of EEE
Emitting Elite Energy

BEEL456D
ARDUINO & RASPBERRY PI BASED PROJECTS

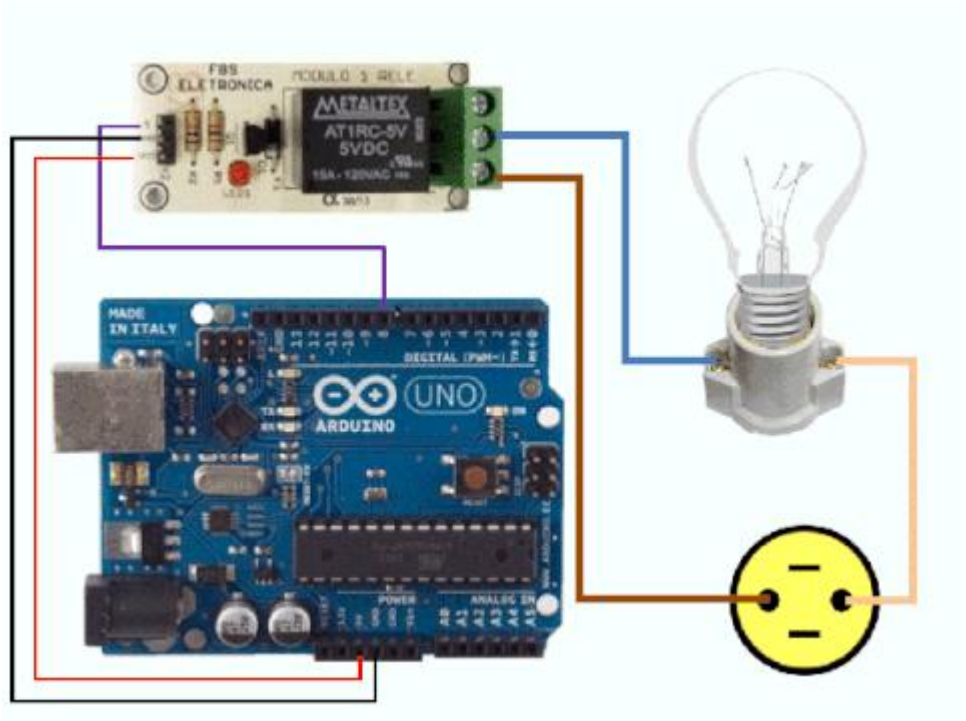


Presented by,
Mr.Shreeshayana R
Assistant Professor
Electrical and Electronics Engineering
ATME College of Engineering, Mysuru

SESSION-1


U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51

SESSION OUTLINE



1.1: Arduino Fundamentals

1.1.1Arduino Platform

1.1.2 Arduino Board

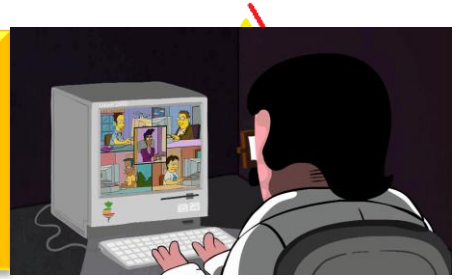
1.1.3 Arduino UNO Board:

1.2.What is used for?

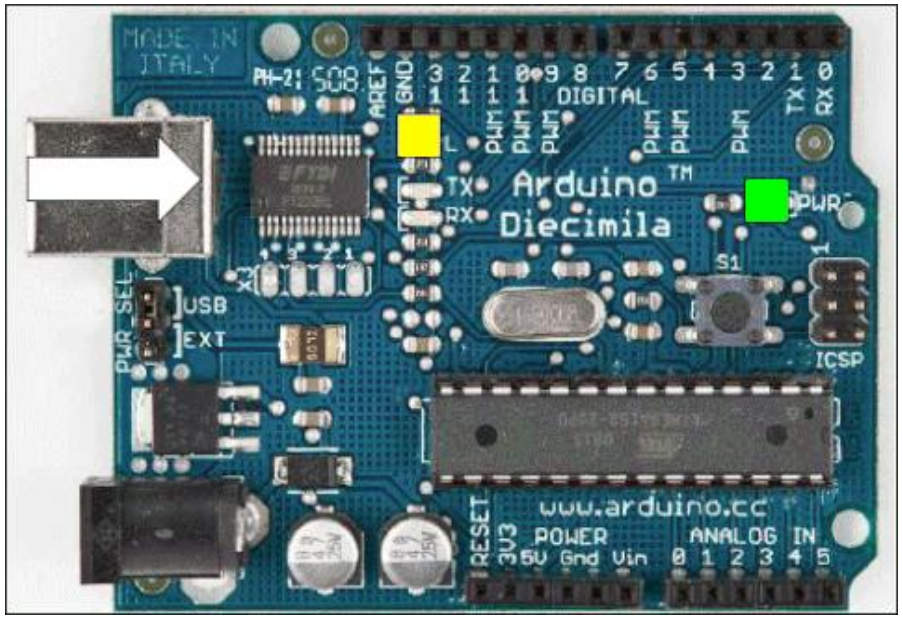
1.2.1 What can it do?

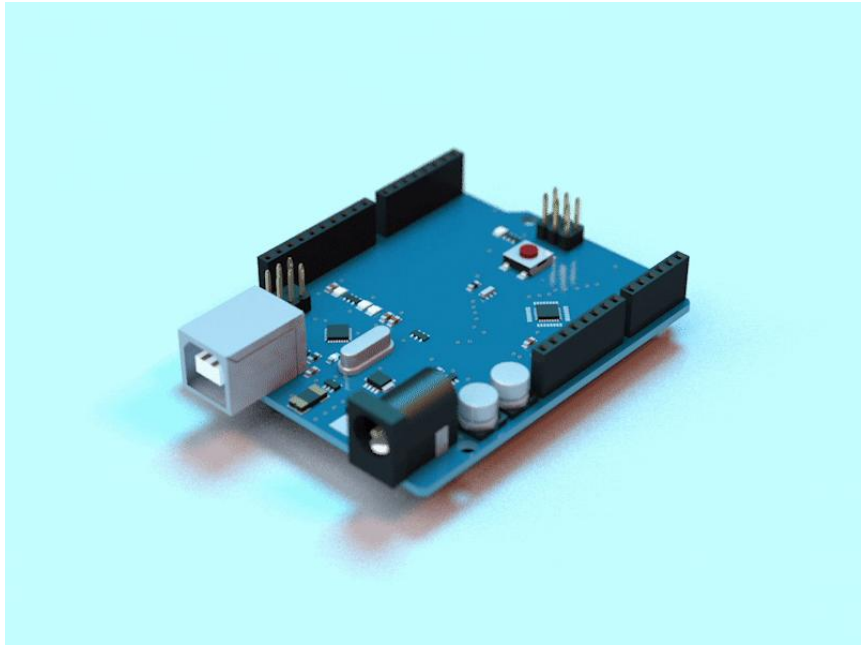
1.3. Why Arduino?

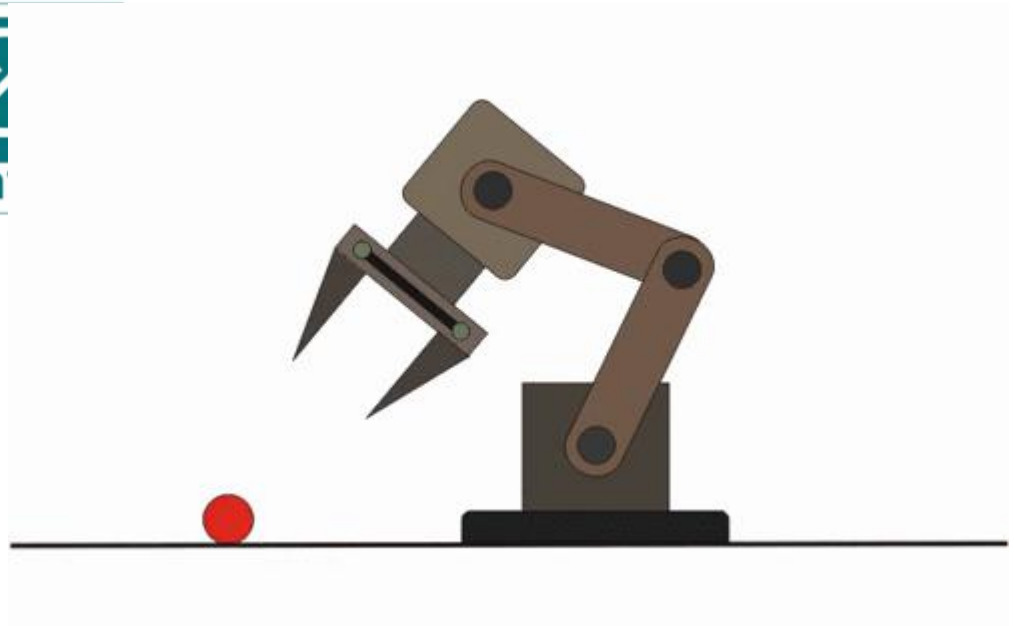
ARDUINO



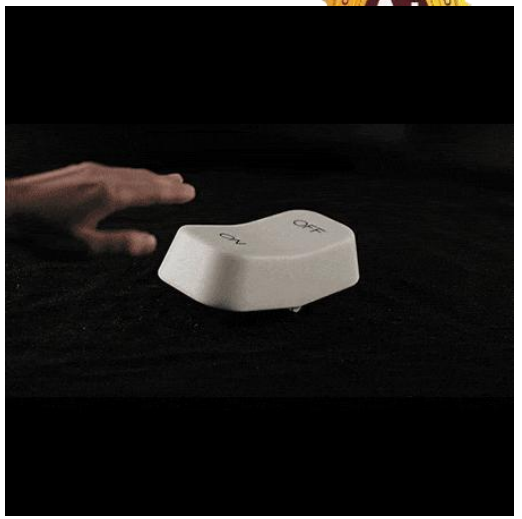
Arduino is an open-source prototyping platform in electronics based on **easy-to-use hardware** and **software**.





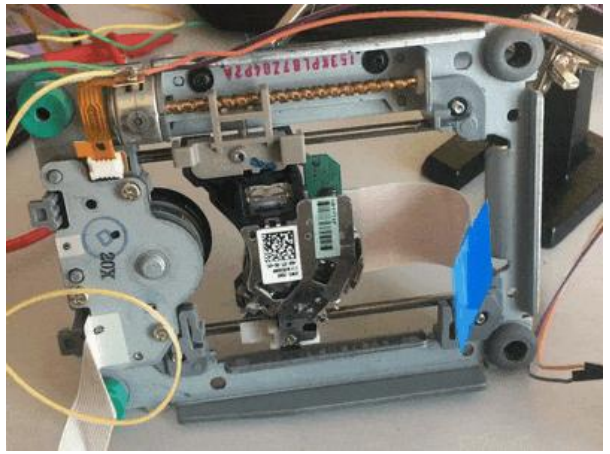


Robotic Applications

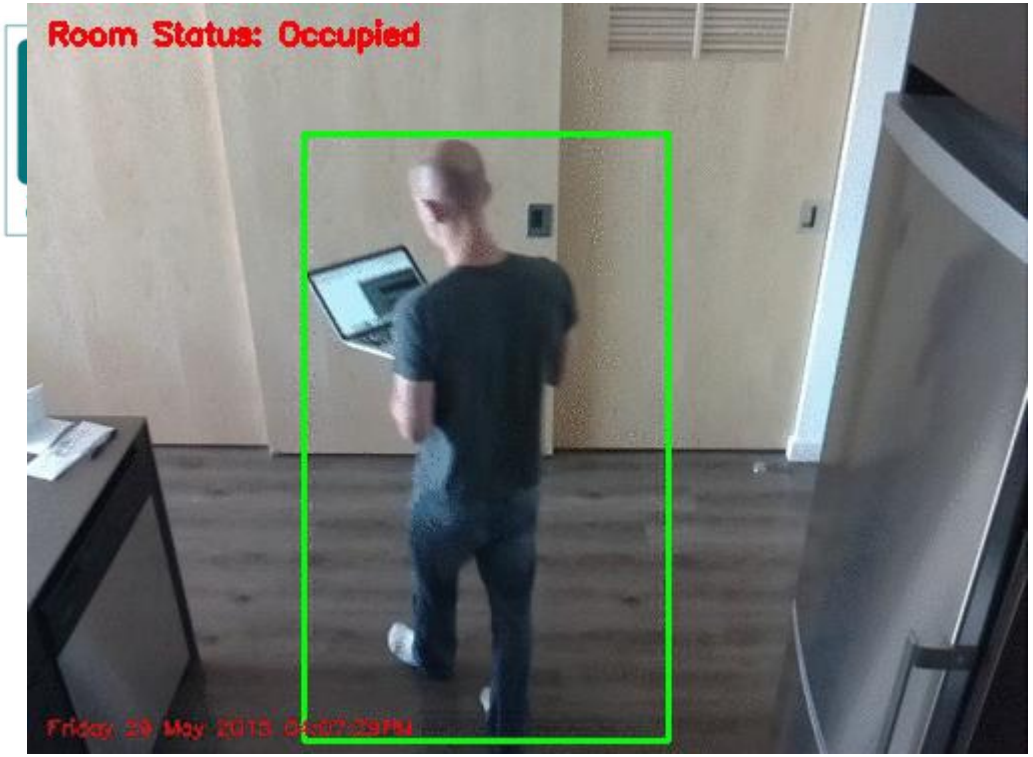


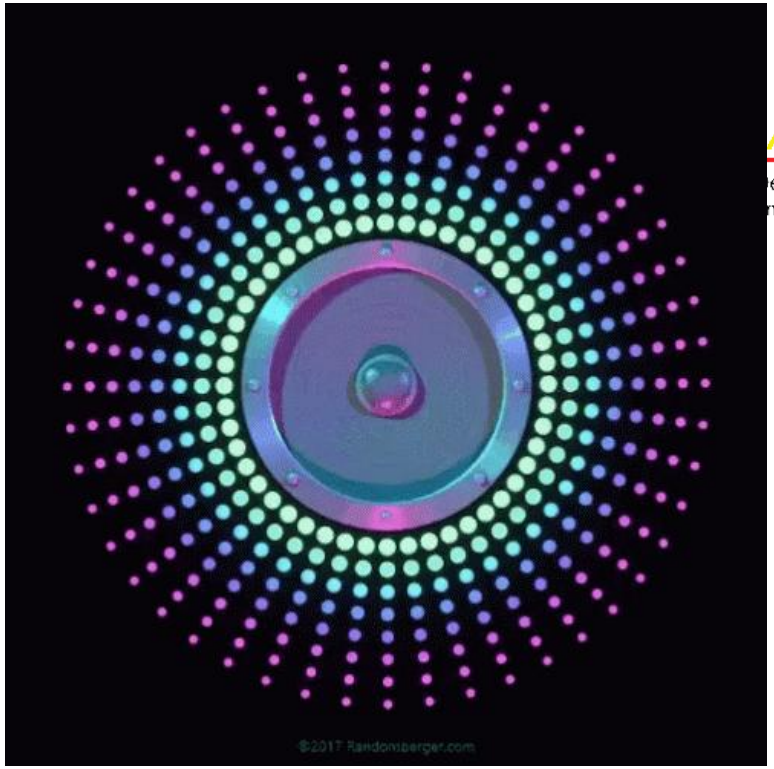
Light on a sensor

switching on an LED

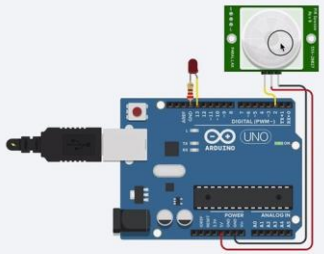


Printers





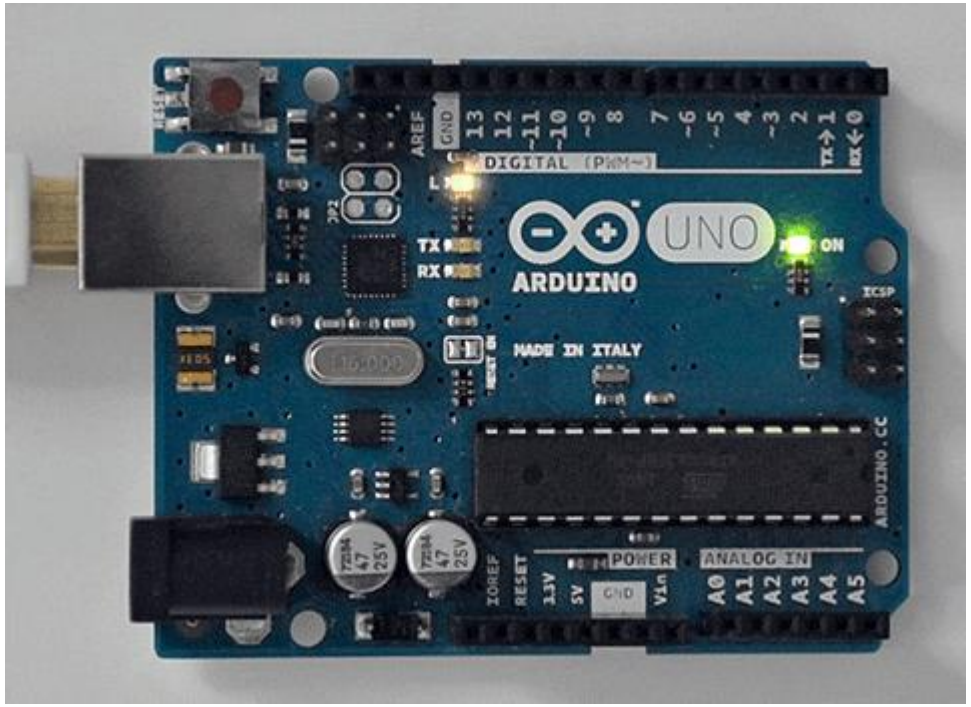
Motion sensing



Playing songs through a speaker

1.1: Arduino Fundamentals

Arduino is a physical Input/ Output board(I/O) with a programmable Integrated Circuit(IC)



1.1.1Arduino Platform

- The language to program is C based which eliminate the difficulty to program in **Assembly language** that mean the Arduino software convert the C code to the respective **assembly code of the Arduino and upload it.**

It includes a

- ❖ Code editor
- ❖ Compiler
- ❖ Up-loader

It also includes:

Code libraries for using peripherals
Such as serial ports and various types of displays

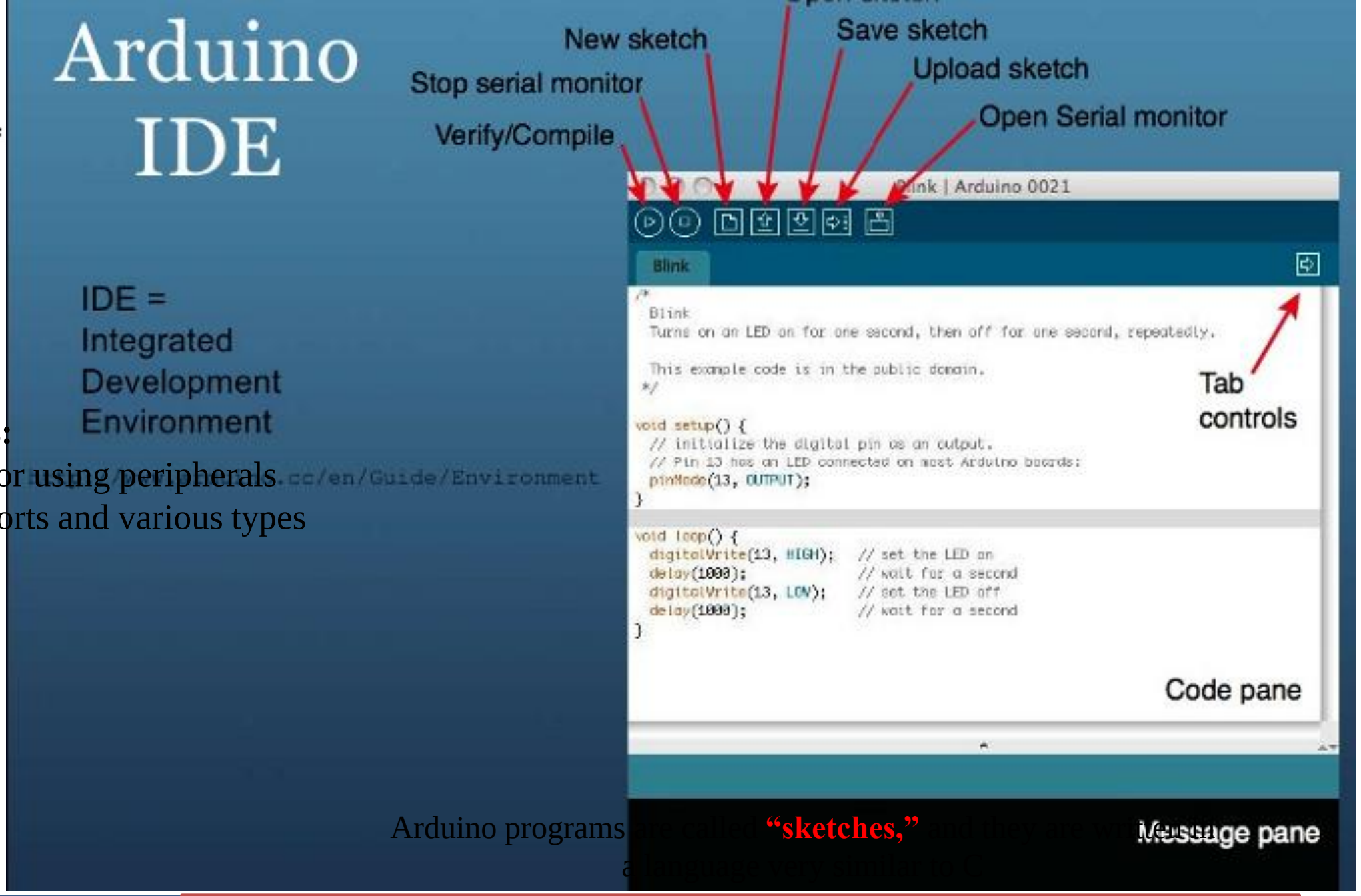
Arduino programs are called **"sketches,"** and they are written in a language very similar to C

It includes a

- ❖ Code editor
- ❖ Compiler
- ❖ Up-loader

It also includes:

Code libraries for using peripherals.
Such as serial ports and various types of displays



Arduino programs are called **"sketches,"** and they are written in a language very similar to C.

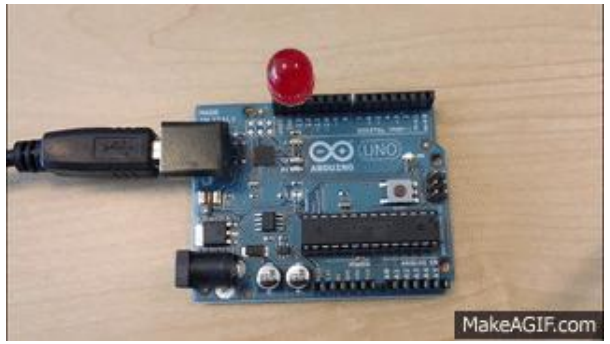
1.1.2 Arduino Board

- The Arduino Board itself is a **blue circuit board**, the size of a **credit card** (but they also have models in other sizes). It has two rows of connectors (the ‘headers’), a power connector and a USB connector.
- The brain of the board is an **Atmel microcontroller.** It’s like a really small, very low power ‘computer’. (It only has 32KB of storage, 2KB of RAM, and the 8-bit processor runs at only 16MHz).

1.2 Types of Arduino Board

Different Types Of Arduino Boards

- Arduino Uno (R3)
- Arduino Nano.
- Arduino Micro.
- Arduino Due.
- LilyPad Arduino Board.
- Arduino Bluetooth.
- Arduino Diecimila.
- RedBoard Arduino Board.



1.1.3 Arduino UNO Board:

Department of EEE
Emitting Elite Energy

- **Arduino Uno** is a microcontroller board based on the ATmega328P (datasheet).
- It has 14 digital input/output pins (of which 6 can be used as PWM (The pins marked with the (~) symbol can simulate analog outputs), 6 analog inputs(A0-A5), a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.
- It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Digital Input/Output Pins.
Connect signal pins of sensors, lights, etc here.

Plug in the USB cable here to connect to your computer.

Connect to a 9V battery here if you have an adapter

Provides 5V power to a circuit

Either of these will ground the circuit

Analog pins, for sensors that measure a continuum.

Quiz

What are the different types of Arduino Boards?

Options

1. Arduino UNO
2. Arduino Nano
3. Arduino Mini
4. All the options

1.2.What is used for?

1. What is used for?
2. Physical Computing projects/ research
3. Rapid Prototyping
4. Interactive experiments
5. When you think to control hardware with little knowledge and easy installation.

1.2.1 What can it do?

Sensors(to sense the environment)

→ *Push buttons, touch pads.*

→*Motion detection*

→ *Photoresistors (sensing light levels)*

→ *Thermistors (measuring temperature)*

→ *Ultrasound (proximity range finder)*

Actuators (to perform action)

→ *On/Off lights, LED's*

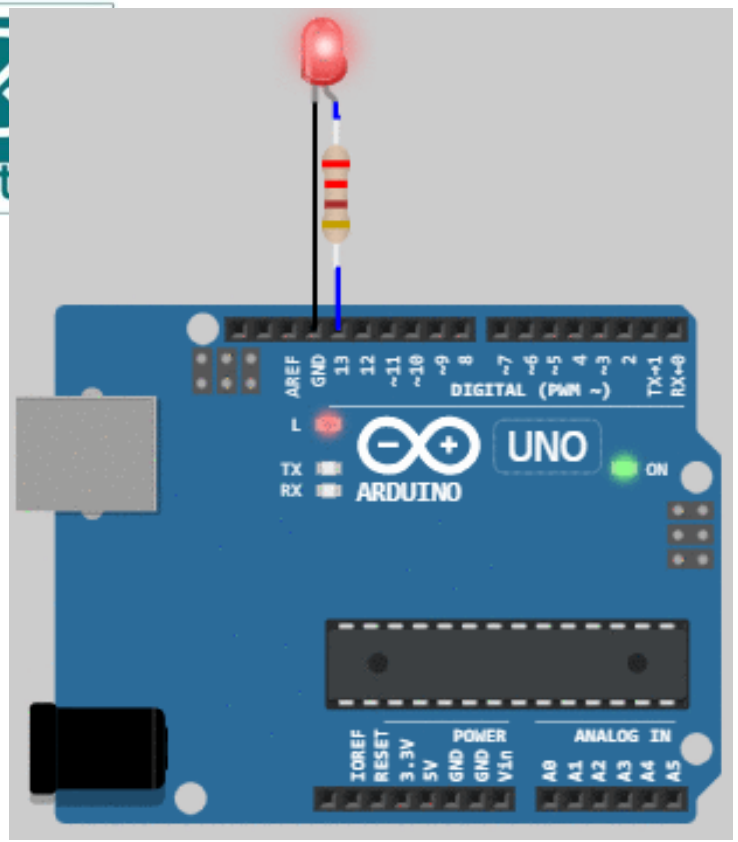
→ *Motors*

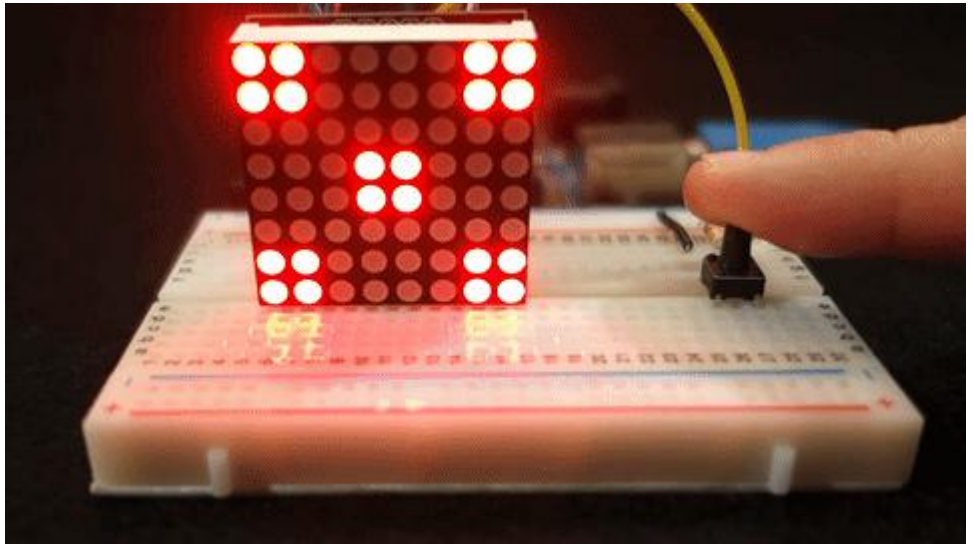
→ *Speakers*

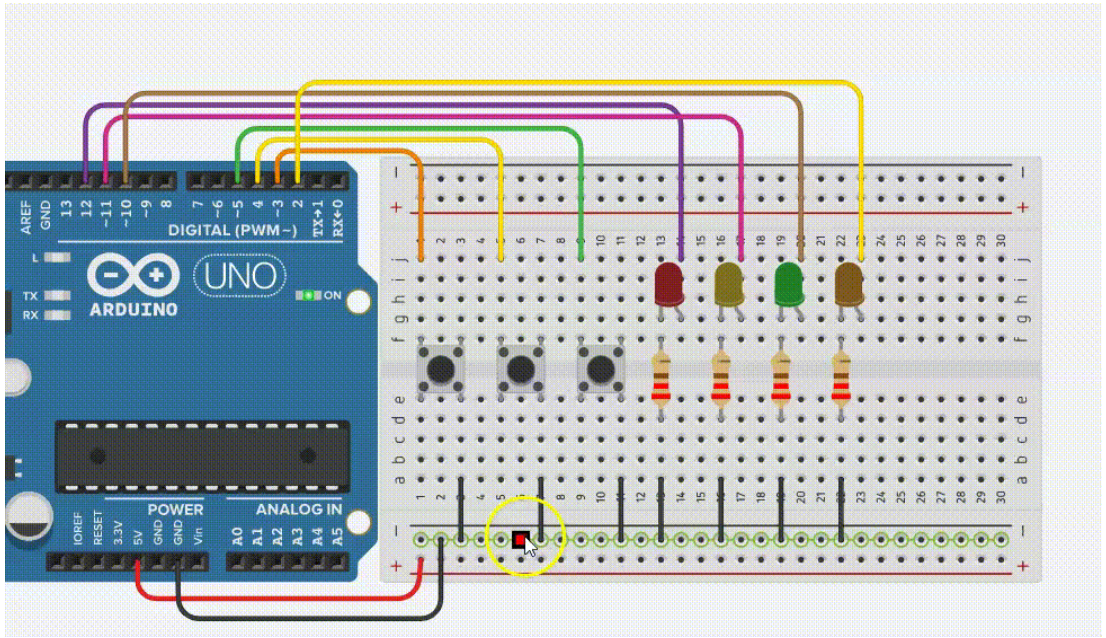
→ *Dispays(LCD)*

1.3 Why Arduino?

1. It is Open Source, both in terms of **Hardware and software**.
2. It is cheap, the board itself and hardware components used in it.
3. It can communicate with a computer through serial Connection over USB.
4. It can be powered from **USB or standalone DC power (5v)**
5. It can work with both Digital and Analog electronic Signals. Sensors and Actuators.
6. You can make cool stuff! Automation vehicles, robot to take care of you and Automate your house and more :-)





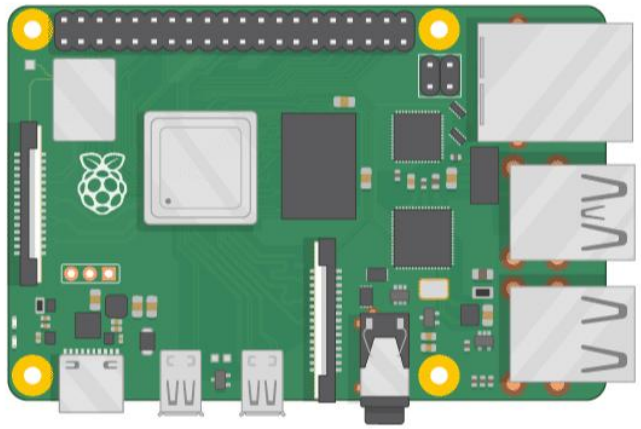


Department of EEE
Emitting Elite Energy

Raspberry Pi

The Raspberry Pi features a set of GPIO (General Purpose Input/Output) pins that allow it to interact with various external devices, such as sensors, motors, LEDs, and other components.

Raspberry Pi

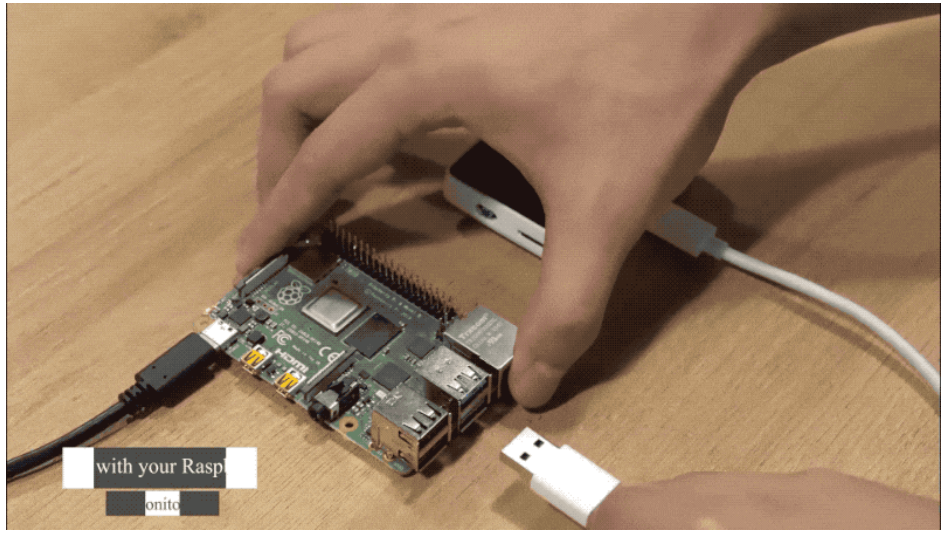


Pinout Overview

The Raspberry Pi pinout typically includes:

- **GPIO Pins (General Purpose Input/Output)**: These can be configured as inputs or outputs for controlling external devices or receiving signals.
- **Power Pins**: These provide 3.3V, 5V, and ground (GND) connections.
- **Ground Pins**: Used to complete circuits by providing a common ground reference.
- **Special Function Pins**: Some pins can perform specific functions like I2C, SPI, UART, PWM (Pulse Width Modulation), and more.

Raspberry Pi



Pinout Overview

The Raspberry Pi pinout typically includes:

- **GPIO Pins (General Purpose Input/Output)**: These can be configured as inputs or outputs for controlling external devices or receiving signals.
- **Power Pins**: These provide 3.3V, 5V, and ground (GND) connections.
- **Ground Pins**: Used to complete circuits by providing a common ground reference.
- **Special Function Pins**: Some pins can perform specific functions like I2C, SPI, UART, PWM (Pulse Width Modulation), and more.

Raspberry Pi

Power Pins:

- **Pin 1**: 3.3V Power
- **Pin 2**: 5V Power
- **Pin 4**: 5V Power
- **Pin 17**: 3.3V Power

Ground Pins:

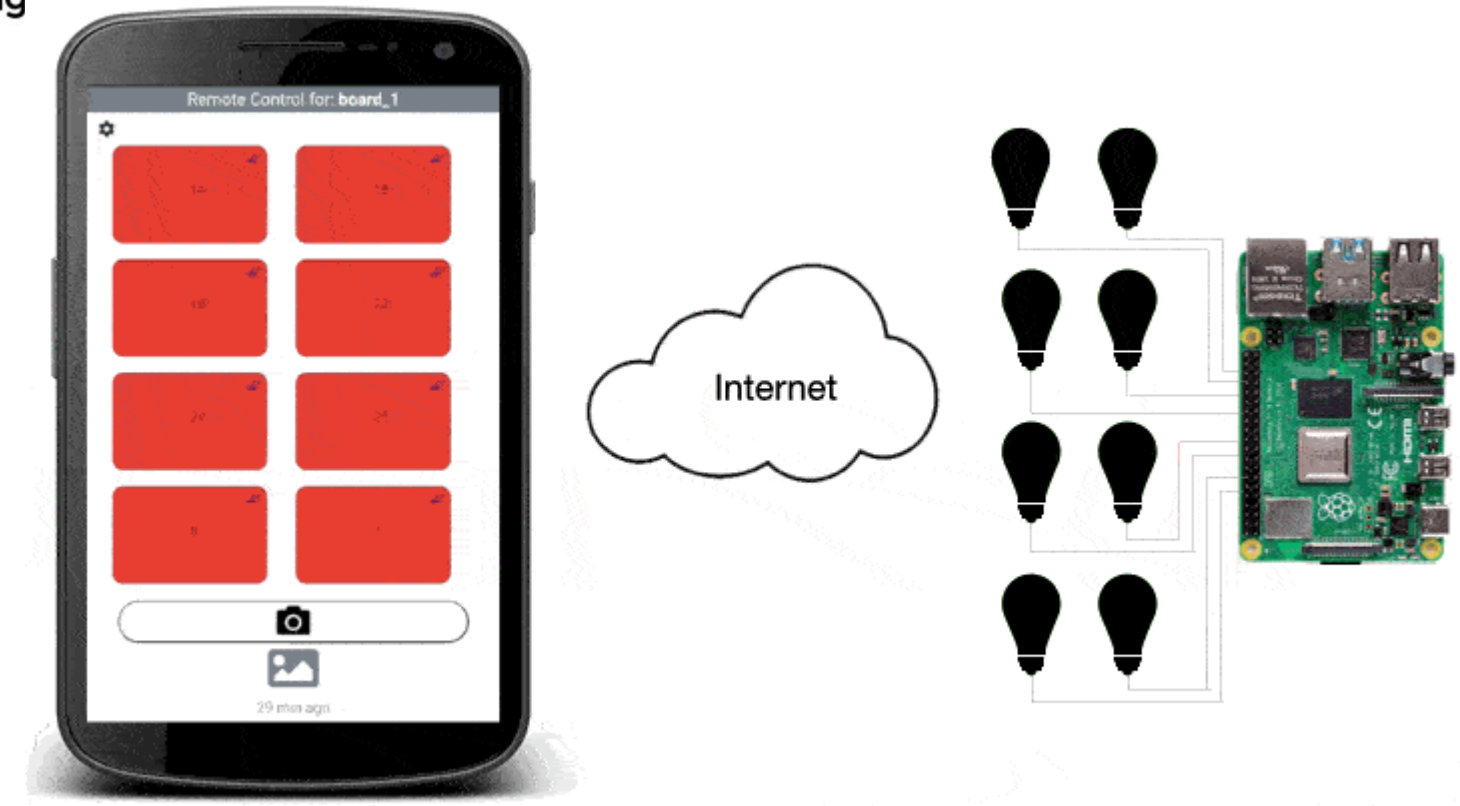
- **Pin 6**, **Pin 9**, **Pin 14**, **Pin 20**, **Pin 25**, **Pin 30**, **Pin 34**, **Pin 39**

Key GPIO Pins:

- **GPIO 2 (SDA)** and **GPIO 3 (SCL)**: I2C data and clock pins.
- **GPIO 14 (TXD)** and **GPIO 15 (RXD)**: UART serial communication (TX and RX).
- **GPIO 10 (MOSI)**, **GPIO 9 (MISO)**, **GPIO 11 (SCK)**: SPI data pins for communication with SPI-compatible devices.
- **GPIO 4, 17, 27, 22, 23, 24, 25, 5, 6, 12, 13, 19, 16, 26, 20, 21**: General-purpose digital input/output pins.
- **GPIO 18, 12, 13, 32, 33**: PWM pins for controlling motor speed or LED brightness.

Special Features:

- **PWM (Pulse Width Modulation)**: Allows you to simulate analog output by adjusting the duty cycle of a digital signal. This is useful for controlling motor speed or dimming lights.
- **I2C (Inter-Integrated Circuit)**: A protocol for communication between the Raspberry Pi and multiple peripherals like sensors and displays using just two wires.
- **SPI (Serial Peripheral Interface)**: A high-speed communication protocol for transferring data between devices like sensors, displays, and SD cards.
- **UART (Universal Asynchronous Receiver-Transmitter)**: Serial communication that allows the Raspberry Pi to communicate with other devices like GPS receivers or Bluetooth modules.


Remote Control for: board_1
Internet

Feature	Arduino UNO	Raspberry Pi
Processor	ATmega328P microcontroller	ARM-based CPU (e.g., Raspberry Pi 4)
Clock Speed	16 MHz	Up to 1.5 GHz (varies by model)
RAM	2 KB	2 GB to 8 GB (depends on the model)
Storage	32 KB Flash Memory	MicroSD card storage (with OS)
GPIO Pins	14 Digital, 6 Analog	40 Digital pins (more flexible)
Power Supply	5V DC/ external or via USB	USB-C or Micro-USB (depending on model)
Operating System	None (bare-metal programming)	Full OS (e.g., Raspberry Pi OS)
Programming Language	Arduino IDE (C/C++)	Python, C/C++, Java, JavaScript, etc.
Connectivity	Requires external modules (Wi-Fi, BT)	Built-in Wi-Fi, Bluetooth, Ethernet (depending on the model)
Primary Use	Simple control and automation tasks	Complex tasks, computing, networking
Power Consumption	Low power consumption	Higher power consumption
Cost	~$20	~$35 and up (depends on the model)
Real-time Performance	Excellent for real-time tasks	Not suited for real-time performance
Expansion	Shields (e.g., motor, Wi-Fi, Bluetooth)	USB ports, GPIO pins, HATs, cameras, displays
I/O Features	Limited I/O pins and functions	More advanced I/O options (SPI, I2C, UART, PWM)
Community Support	Very large and beginner-friendly	Large, active community with more complex resources
Usage Complexity	Easy to use and beginner-friendly	Slightly more complex, suitable for advanced projects

Experiment No. 1

- To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to 'turn ON' LED for 1 sec after every 2 seconds.
- To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to 'turn ON' LED when push button is pressed or at sensor detection.

Objectives:

- To implement the basic principles of interfacing an LED and buzzer with Arduino/Raspberry Pi and write a C program to turn ON the LED for 1 second after every 2 seconds.
- To implement the concept of interfacing a push button and digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a C program to turn ON the LED when the push button is pressed or sensor detection occurs.

Components Required:

- Arduino UNO
- LED/ Buzzer.
- Resistor (220Ω)
- Connecting cable or USB cable.
- Breadboard.
- Jumper wires

Pin Connections:

- Arduino Digital Pin (e.g., pin 13) -> One end of 220Ω Resistor -> Anode (+, longer leg) of LED/Positive (+) of Buzzer
- Cathode (-, shorter leg) of LED/Negative (-) of Buzzer -> GND Pin (Arduino)

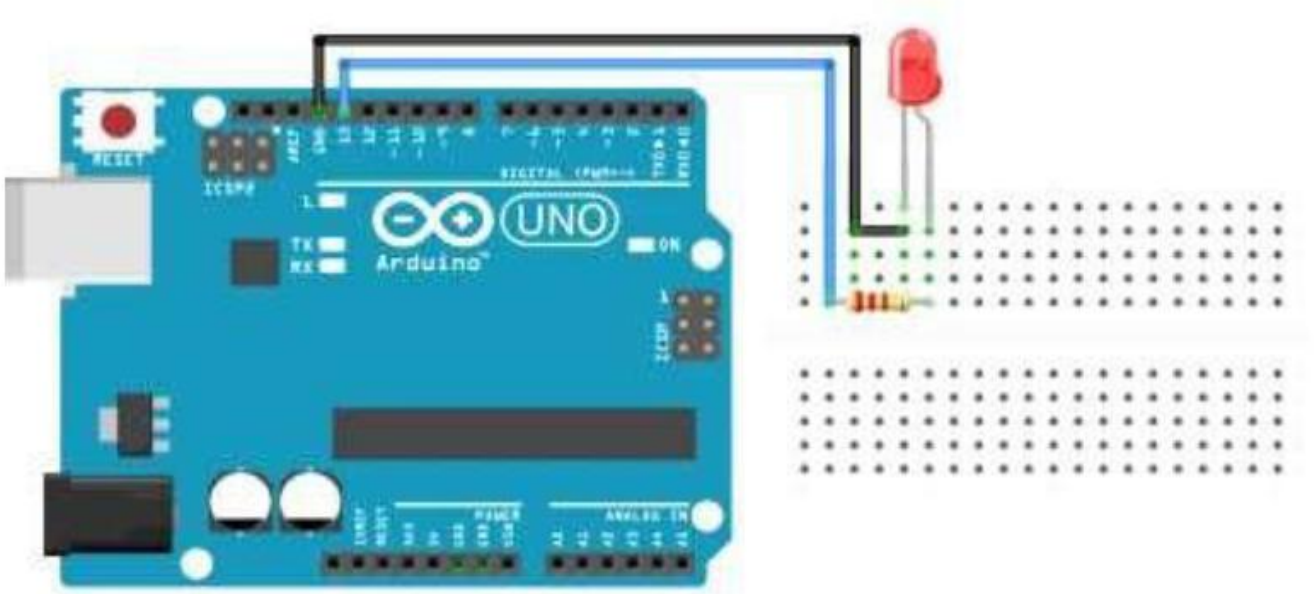


Figure-1.1 b): LED interfacing with Arduino

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

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

ii) To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to 'turn ON' LED when push button is pressed or at sensor detection.

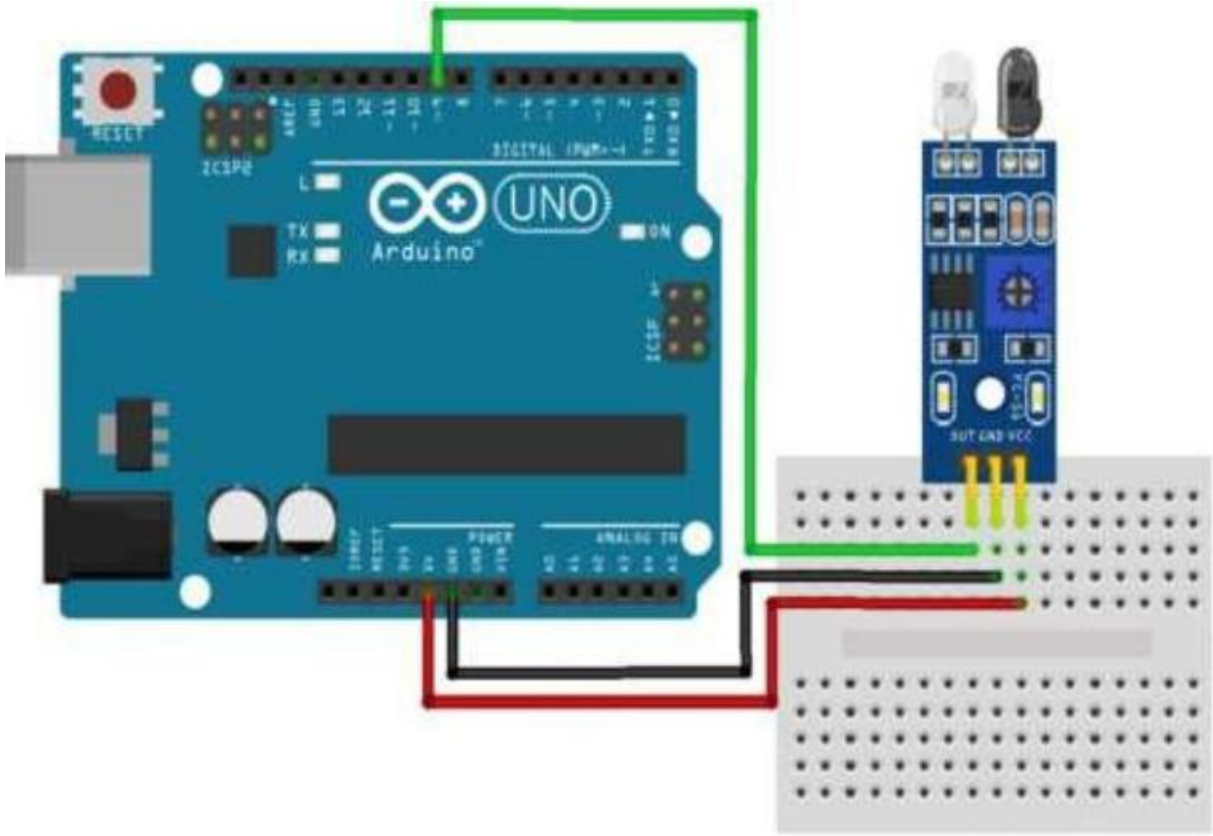
Objective:

- To understand and implement the concept of interfacing a push button and digital sensor (IR/LDR) with Arduino/Raspberry Pi
- Write a C program to turn ON the LED when the push button is pressed or sensor detection occurs

CASE : Using Infrared Sensor

Components Required:

- Arduino UNO.
- Infrared Sensor(IR)
- LED.
- Resistor (220Ω)
- Connecting cable or USB cable.
- Breadboard.
- Jumper wires.



Pin Connections:

- Arduino 5V -> IR Sensor VCC
- Arduino GND -> IR Sensor GND
- Arduino Digital Pin (e.g., pin 9) -> IR Sensor OUT
- Arduino Digital Pin (e.g., pin 13) -> One end of 220Ω Resistor -> Anode (+, longer leg) of LED
- Cathode (-, shorter leg) of LED -> GND Pin (Arduino)

Thank You