Study of Arduino and Raspberry Pi

# 1. Aim

To study Arduino, Raspberry Pi, and how they communicate using serial communication.

# 2. Theory

## Arduino

Arduino is an open-source electronics board. It is easy to use and is mainly used for small electronic projects. It has a microcontroller that can control devices like LEDs, motors, and sensors using a simple programming language.

## Raspberry Pi

Raspberry Pi is a small computer that can run an operating system (like Linux). It has GPIO pins to connect and control hardware like sensors. It supports many programming languages, especially Python.

## Serial Communication

Serial communication is a way for devices to send and receive data one bit at a time. Arduino and Raspberry Pi can be connected through a serial port to share data (like sensor readings or control signals).

# 3. Materials Required

• Arduino Uno board

• Raspberry Pi (any model)

• USB cable for Arduino

• Breadboard

• Jumper wires

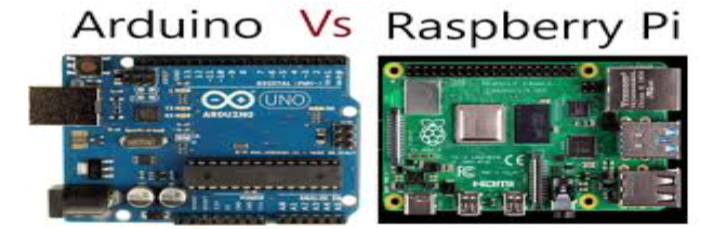
• LED

• 220-ohm resistor

• DHT11 Temperature and Humidity Sensor

• MicroSD card with Raspbian OS (for Raspberry Pi)

• Monitor, keyboard, and mouse (for Raspberry Pi)



# 4. Procedure

## Part 1: Arduino Setup

1. Connect the LED:  
 - Place the LED on the breadboard.  
 - Connect the long leg (anode) to digital pin 9 of Arduino using a 220-ohm resistor.  
 - Connect the short leg (cathode) to GND on Arduino.  
  
2. Upload Code to Arduino:  
 - Open Arduino IDE on your computer.  
 - Write and upload the following code:

void setup() {  
 Serial.begin(9600); // Start serial communication  
 pinMode(9, OUTPUT); // Set pin 9 as output  
}  
void loop() {  
 digitalWrite(9, HIGH); // Turn LED ON  
 Serial.println("LED ON");  
 delay(1000); // Wait 1 second  
 digitalWrite(9, LOW); // Turn LED OFF  
 Serial.println("LED OFF");  
 delay(1000); // Wait 1 second  
}

## Part 2: Raspberry Pi Setup

1. Prepare Raspberry Pi:  
 - Insert microSD card with Raspbian OS.  
 - Connect monitor, keyboard, and mouse.  
 - Power it ON and finish setup.  
  
2. Install Python Serial Library:  
 Open terminal and type:  
 sudo apt-get update  
 sudo apt-get install python3-serial  
  
3. Create Python Script:  
 Create a file called read\_serial.py and add:

import serial  
ser = serial.Serial('/dev/ttyACM0', 9600)  
while True:  
 try:  
 line = ser.readline().decode('utf-8').strip()  
 print(line)  
 except KeyboardInterrupt:  
 break  
ser.close()

4. Run Script:  
 python3 read\_serial.py  
  
5. Connect Arduino to Raspberry Pi:  
 Use a USB cable to connect Arduino.

# 5. Working

• Arduino: Turns LED ON and OFF every second and sends "LED ON/OFF" messages through serial communication.  
• Raspberry Pi: Reads these messages from Arduino and displays them on the terminal.

# 6. Difference between Arduino & Raspberry Pi

|  |  |  |
| --- | --- | --- |
| S No. | Arduino | Raspberry Pi |
| 1 | Started in 2005 in Italy | Started in 2012 by Eben Upton |
| 2 | Uses Atmega microcontroller | Uses ARM processor |
| 3 | Works like a microcontroller | Works like a small computer |
| 4 | Controls electronics directly | Computes and controls based on OS |
| 5 | Simple hardware & software | Complex hardware & software |
| 6 | 8-bit CPU | 64-bit CPU |
| 7 | Very low RAM (2 KB) | High RAM (1 GB) |
| 8 | 16 MHz speed | 1.4 GHz speed |
| 9 | Cheaper | More expensive |
| 10 | Higher I/O current | Lower I/O current |
| 11 | Low power use (200 mW) | High power use (700 mW) |
| 12 | 5V logic | 3V logic |
| 13 | No internet support | Has Ethernet and WiFi |
| 14 | Used for simple projects | Used for advanced projects |
| 15 | Examples: Traffic light, weighing machine | Examples: Robots, game servers |
| 16 | No OS needed | OS required |

# 7. Conclusion

In this lab, we learned how to use Arduino and Raspberry Pi and how they can communicate using serial communication. We controlled an LED with Arduino and read its status on Raspberry Pi.