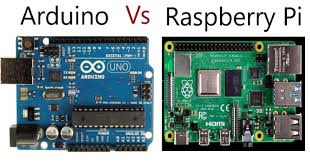
**1. Aim**

To study the Arduino and Raspberry Pi platforms and their serial communication.

**2. Theory**

* **Arduino** is an open-source microcontroller platform ideal for interactive electronics using simple programming and hardware.
* **Raspberry Pi** is a full-fledged computer that supports various programming languages like Python and can interface with hardware via GPIO.
* **Serial Communication** is used to send/receive data between Arduino and Raspberry Pi, helpful in integrating sensors and control signals.



**3. Materials Required**

* Arduino Uno
* Raspberry Pi (any model)
* USB cable (Arduino), Breadboard, Jumper wires
* LED, 220-ohm resistor, DHT11 sensor
* MicroSD with Raspbian OS
* Monitor, keyboard, mouse

**4. Procedure**

**Part 1: Arduino Setup**

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

cpp

Copy code

void setup() {

Serial.begin(9600); // Initialize serial communication

pinMode(9, OUTPUT); // Set pin 9 as output

}

void loop() {

digitalWrite(9, HIGH); // Turn the LED on

Serial.println("LED ON");

delay(1000); // Wait for a second

digitalWrite(9, LOW); // Turn the LED off

Serial.println("LED OFF");

delay(1000); // Wait for a second

}

**Part 2: Raspberry Pi Setup**

1. **Prepare Raspberry Pi:**
   * Insert the microSD card into the Raspberry Pi.
   * Connect the monitor, keyboard, and mouse.
   * Power up the Raspberry Pi and complete the initial setup.
2. **Install Python Serial Library:**
   * Open a terminal window on the Raspberry Pi.
   * Install the pyserial library with:

bash

Copy code

sudo apt-get update

sudo apt-get install python3-serial

1. **Create Python Script:**
   * Create a Python script to read the serial data from Arduino. Open a text editor and write the following script:

python

Copy code

import serial

# Open serial port

ser = serial.Serial('/dev/ttyACM0', 9600)

while True:

try:

line = ser.readline().decode('utf-8').strip()

print(line)

except KeyboardInterrupt:

break

ser.close()

1. **Run the Script:**
   * Save the Python script as read\_serial.py.
   * Execute the script by running:

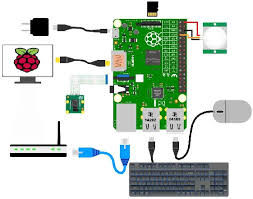
bash

Copy code

python3 read\_serial.py

1. **Connect Arduino to Raspberry Pi:**
   * Connect the Arduino to the Raspberry Pi using a USB cable.

**5. Image**

**

*Fig:-Arduino board connected to the LED Fig:- Raspberry PI Components*

**6. Working**

**Arduino Operation:**

The Arduino code continuously toggles the LED on and off every second while sending serial messages ("LED ON" and "LED OFF") to the connected serial port.

**Raspberry Pi Operation:**

The Python script running on the Raspberry Pi reads the serial messages sent by the Arduino and prints them to the terminal. This allows the Raspberry Pi to monitor the status of the LED on the Arduino.

| **S No.** | **Arduino** | **Raspberry Pi** |
| --- | --- | --- |
| 1. | In the year 2005, the classrooms of the Interactive Design Institute in Ivrea, Italy, first introduced the Arduino board. | In the year 2012, Eben Upton first introduced the Raspberry Pi device in February. |
| 2. | Control unit of the Arduino is from the Atmega family. | The control unit of Raspberry Pi is from the ARM family. |
| 3. | Arduino is based on a microcontroller. | While Raspberry Pi is based on a microprocessor. |
| 4. | It is designed to control the electrical components connected to the circuit board in a system. | While Raspberry Pi computes data and produces valuable outputs, and controls components in a system based on the outcome of its computation. |
| 5. | Arduino boards have a simple hardware and software structure. | While Raspberry Pi boards have a complex architecture of hardware and software. |
| 6. | CPU architecture: 8 bit. | CPU architecture: 64 bit. |
| 7. | It uses very little RAM, 2 kB. | While Raspberry Pi requires more RAM, 1 GB. |
| 8. | It clocks a processing speed of 16 MHz. | While Raspberry Pi clocks a processing speed of 1.4 GHz. |
| 9. | It is cheaper in cost. | While Raspberry Pi is expensive. |
| 10. | It has a higher I/O current drive strength. | While Raspberry Pi has a lower I/O current drive strength. |
| 11. | It consumes about 200 MW of power. | While it consumes about 700 MW of power. |
| 12. | Its logic level is 5V. | Its logic level is 3V. |
| 13. | It does not have internet support. | It has inbuilt Ethernet port and WiFi support. |
| 14. | It has higher current drive strength. | It has lower current drive strength. |
| 15. | Some of the applications of Arduino are traffic light countdown timer , Weighing machines , etc. | Some of the applications of Raspberry Pi are Stop motion cameras , Robot Controllers , Game Servers. |
| 16. | Operating systems are required in Arduino. | Operating System is required in Raspberry Pi. |

**7. Conclusion**

The lab exercise successfully demonstrated the basic operations of Arduino and Raspberry Pi, as well as how to establish serial communication between them. By completing this experiment, students gained practical experience in interfacing microcontrollers with single-board computers and understood the fundamentals of serial data exchange.