

Serial Communication between Arduino Uno and PC

Experiment 1.1: Sending Text from Arduino to PC

Aim:

To send a simple text message from the Arduino Uno to the PC and display it on the Serial Monitor.

Hardware Requirements:

Arduino Uno
USB cable
PC with Arduino IDE installed

Theory:

Serial communication is the process of transmitting data one bit at a time over a communication channel. The Arduino Uno communicates with the PC using the UART protocol via a USB connection. The `Serial.begin()` function initializes the serial communication, and `Serial.print()` or `Serial.println()` sends data.

Procedure:

1. Connect the Arduino Uno to the PC using the USB cable.
2. Open the Arduino IDE.
3. Write the following code to send a text message to the Serial Monitor.
4. Upload the code to the Arduino Uno.
5. Open the Serial Monitor (Ctrl+Shift+M) in the Arduino IDE.
6. Observe the message being displayed.

Program:

```
void setup() {  
    Serial.begin(9600); // Initialize serial communication at 9600 baud  
    Serial.println("Hello, PC! This is Arduino.");  
}  
  
void loop() {  
    // No repeated actions needed for this experiment  
}
```

Result:

The Serial Monitor displays the message: *"Hello, PC! This is Arduino."*

Experiment 1.2: Sending Sensor Data to PC

Aim:

To send analog sensor data (e.g., potentiometer readings) from the Arduino Uno to the PC.

Hardware Requirements:

Arduino Uno USB
cable Potentiometer
Connecting wires

Theory:

Analog sensors provide variable voltage signals based on physical conditions. The Arduino reads these signals using its ADC (Analog-to-Digital Converter) and sends the readings to the PC via serial communication.

Procedure:

1. Connect the potentiometer to the Arduino Uno:
 - Connect the middle pin of the potentiometer to A0.
 - Connect one outer pin to 5V and the other to GND.
2. Connect the Arduino Uno to the PC using the USB cable.
3. Write and upload the following code to the Arduino Uno.
4. Open the Serial Monitor in the Arduino IDE.
5. Rotate the potentiometer and observe the values.

Program:

```
void setup() {  
    Serial.begin(9600); // Initialize serial communication  
}  
  
void loop() {  
    int sensorValue = analogRead(A0); // Read the potentiometer value  
    Serial.print("Sensor Value: ");  
    Serial.println(sensorValue); // Send the value to the Serial Monitor  
    delay(500); // Wait for 500 milliseconds  
}
```

Result:

The Serial Monitor displays the potentiometer readings as you rotate it.

Experiment 1.3: Sending Data from PC to Arduino

Aim:

To send data from the PC to the Arduino Uno using the Serial Monitor and control an LED based on the input.

Hardware Requirements:

- Arduino Uno
- USB cable
- LED
- 220-ohm resistor
- Breadboard
- Connecting wires

Theory:

Serial communication is bidirectional, allowing data transfer from the PC to the Arduino. Input data sent via the Serial Monitor can be processed by the Arduino to perform specific tasks, such as controlling an LED.

Procedure:

1. Connect the LED to the Arduino Uno:
 - Connect the positive leg of the LED to pin 13 through a 220-ohm resistor.
 - Connect the negative leg to GND.
2. Connect the Arduino Uno to the PC.
3. Write and upload the following code.
4. Open the Serial Monitor in the Arduino IDE.
5. Enter ON or OFF in the Serial Monitor and observe the LED behavior.

Program:

```
void setup() {  
  Serial.begin(9600); // Initialize serial communication  
  pinMode(13, OUTPUT); // Set pin 13 as output  
}  
  
void loop() {  
  if (Serial.available() > 0) { // Check if data is available  
    String command = Serial.readString(); // Read the input as a string if  
    (command == "ON") {  
      digitalWrite(13, HIGH); // Turn on the LED  
      Serial.println("LED is ON");  
    } else if (command == "OFF") { digitalWrite(13,  
      LOW); // Turn off the LED Serial.println("LED  
      is OFF");  
    } else {  
      Serial.println("Invalid Command");  
    }  
  }  
}
```

Result: The LED turns ON or OFF based on the input from the Serial monitor.

Experiment 1.4: Bi-Directional Data Communication

Aim:

To establish bi-directional communication where the Arduino sends a query to the PC and processes the response.

Hardware Requirements:

Arduino Uno
USB cable
PC with Arduino IDE

Theory:

Bi-directional communication involves both sending and receiving data. The Arduino can request input from the user, process it, and respond accordingly.

Procedure:

1. Connect the Arduino Uno to the PC using the USB cable.
2. Write and upload the following code.
3. Open the Serial Monitor in the Arduino IDE.
4. Observe the Arduino sending a query and process your response.

Program:

```
void setup() {  
    Serial.begin(9600); // Initialize serial communication  
    Serial.println("Enter your name:"); // Prompt the user  
}  
  
void loop() {  
    if (Serial.available() > 0) { // Check if data is available  
        String  
        name = Serial.readString(); // Read the user's input  
        Serial.print("Hello, ");  
        Serial.print(name);  
        Serial.println("! Welcome to Arduino.");  
        delay(5000); // Delay before the next prompt  
        Serial.println("Enter your name:");  
    }  
}
```

Result:

The Serial Monitor displays a personalized greeting based on the user's input.