

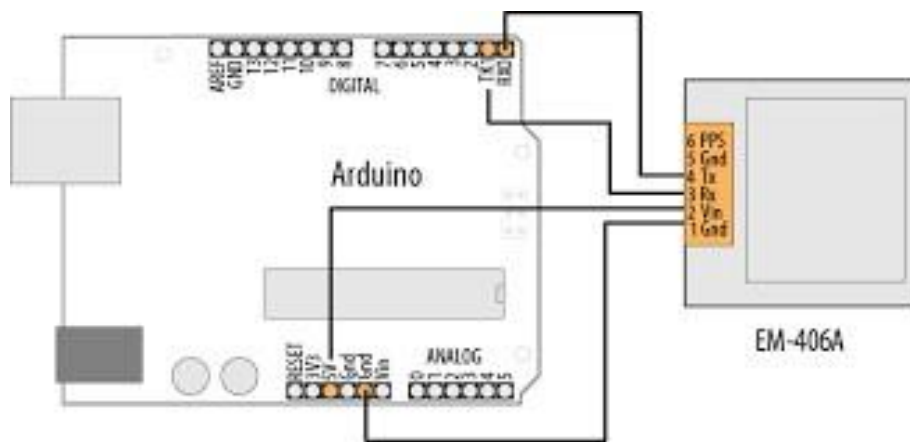
Ex. No: 4	Simulation of serial communication and data exchange in Arduino IDE.
09-08-2024	

Simulate serial communication and data exchange in Arduino IDE.

Aim: To create a serial communication system where an Arduino receives commands from the Serial Monitor to control the state of an LED. The system will respond to commands to turn the LED on or off and will provide feedback to the user through the Serial Monitor.

Algorithm:

1. Initialize Serial Communication: Start serial communication at a specified baud rate.
2. Initialize LED Pin: Set the LED pin as an output.
3. Loop:
 - Check if data is available in the serial buffer.
 - Read the incoming command from the Serial Monitor.
 - Process the command:
 - If the command is "LED ON", turn the LED on and send a confirmation message.
 - If the command is "LED OFF", turn the LED off and send a confirmation message.
 - If the command is unknown, send an error message.
4. Repeat: Continue looping to handle new commands.



Pseudocode:

BEGIN

Initialize serial communication at 9600 baud rate

Set LED pin as OUTPUT

WHILE true

IF data is available in serial buffer THEN

Read the incoming command from serial

IF command is "LED ON" THEN

Turn LED ON

Send "LED is now ON" to serial monitor

ELSE IF command is "LED OFF" THEN Turn

LED OFF

Send "LED is now OFF" to serial monitor ELSE

Send "Unknown command" to serial monitor

END IF

END WHILE

END

Source Code:

```
const int ledPin = 13; // Define the pin for the LED
```

```
void setup() {
```

```
  Serial.begin(9600); // Initialize serial communication at 9600 baud pinMode(ledPin,
```

```
  OUTPUT); // Set LED pin as output
```

```
}
```

```
void loop() {
```

```
  if (Serial.available() > 0) {
```

```
    String command = Serial.readStringUntil('\n'); // Read the incoming command
```

```
    command.trim(); // Remove any leading or trailing whitespace
```

```
    if (command == "LED ON") {
```

```
      digitalWrite(ledPin, HIGH); // Turn the LED on
```

```
      Serial.println("LED is now ON");
```

```
    } else if (command == "LED OFF") {
```

```
      digitalWrite(ledPin, LOW); // Turn the LED off
```

```
      Serial.println("LED is now OFF");
```

```
    } else {
```

```
      Serial.println("Unknown command");
```

```
    }
```

```
  }
```

```
}
```

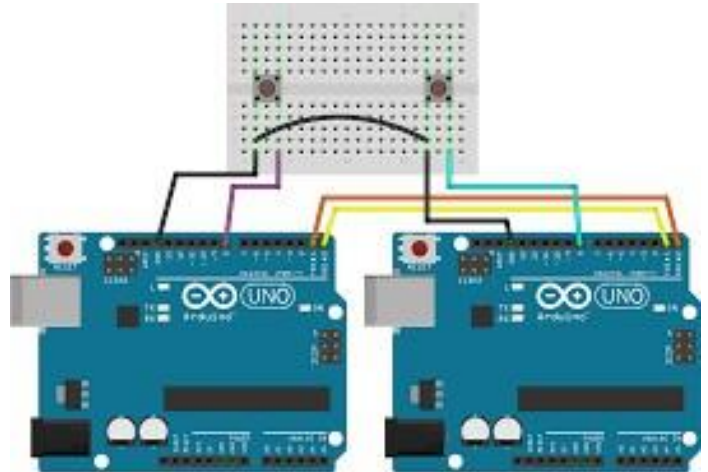
Sample Input/Output

Sample Input:

1. Type `LED ON` in the Serial Monitor and press `Send`.
2. Type `LED OFF` in the Serial Monitor and press `Send`.
3. Type `TEST` in the Serial Monitor and press `Send`.

Sample Output:

1. After sending `LED ON`:
 - Arduino response: `LED is now ON`
 - LED on pin 13 will turn on.
2. After sending `LED OFF`:
 - Arduino response: `LED is now OFF`
 - LED on pin 13 will turn off.
3. After sending `TEST`:
 - Arduino response: `Unknown command`



Documentation Notes:

- Hardware Setup: Connect an LED to pin 13 and ground. Ensure that the LED is correctly oriented with the positive leg connected to the pin and the negative leg to the ground.
- Serial Monitor Configuration: Set the baud rate of the Serial Monitor to 9600 to match the `Serial.begin(9600)` setting in the code.
- Testing: Validate that commands sent from the Serial Monitor control the LED as expected and that feedback is correctly displayed.

RESULT: Hence, we have successfully implemented and completed simulating serial communication and data exchange in Arduino IDE.