

Experiment 10: Develop a program to setup a UART protocol and pass a string through the protocol.

Pin Connections:

Component	ESP32 Pin	Arduino UNO Pin
LED	GPIO 2	D13

Code:

```
#define LED_PIN 2 // GPIO2 for ESP32, use 13 for Arduino Uno
String inputString = ""; // To store incoming serial data
bool stringComplete = false;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600);
  Serial.println("UART Command Processor Initialized.");
  inputString.reserve(200); // Optional: reserve memory for
  speed
}

void loop() {
  // When a complete command is received
  if (stringComplete) {
    processCommand(inputString);
    inputString = "";
    stringComplete = false;
  }
}

// Serial Event triggered by new data
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char) Serial.read();
    inputString += inChar;
    // Check for end of line
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```

```
}  
void processCommand(String command) {  
    command.trim(); // Remove newline or extra whitespace  
    if (command.equalsIgnoreCase("LED ON")) {  
        digitalWrite(LED_PIN, HIGH);  
        Serial.println("LED is turned ON");  
    } else if (command.equalsIgnoreCase("LED OFF")) {  
        digitalWrite(LED_PIN, LOW);  
        Serial.println("LED is turned OFF");  
    } else if (command.equalsIgnoreCase("STATUS")) {  
        int state = digitalRead(LED_PIN);  
        if (state == HIGH) {  
            Serial.println("LED STATUS: ON");  
        } else {  
            Serial.println("LED STATUS: OFF");  
        }  
    } else {  
        Serial.print("Unknown Command: ");  
        Serial.println(command);  
    }  
}
```