UART Communication Protocol

```
#include "HardwareSerial.h"
HardwareSerial mySerial(1);
void setup() {
 mySerial.begin(115200, SERIAL_8N1, 16, 17);   Serial.begin(115200);  }
void loop() {
 mySerial.println("Hello from ESP32 #1");   delay(1000);
 if (mySerial.available()) {
  String data = mySerial.readStringUntil('\n');
  Serial.print("Received from ESP32 #2: ");   Serial.println(data);}    }
```

I2C Communication Protocol

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
lcd.init();     lcd.begin(16,2);     lcd.backlight();    }
lcd.setCursor(0, 0);     lcd.print("Hello, World!");
lcd.setCursor(0, 1);     lcd.print("I2C with ESP32");    }
void loop(){}_____________________________________________________
#include <Wire.h>
void setup() {
Wire.begin();          Serial.begin(115200);     }
void loop() {
Wire.beginTransmission(0x08);
const char* msg = "Hello from Master";
Wire.write((const uint8_t*)msg, strlen(msg) + 1);
Wire.endTransmission();      delay(1000);
Wire.requestFrom(0x08, 20);
while (Wire.available())
char c = Wire.read();    Serial.print(c);      }
Serial.println();    delay(1000);|      }
```

SPI Communication Protocol

```
#include <SPI.h>
#define CS1 PIN 5          #define CS2_PIN 4
void setup() {
Serial.begin(115200);      SPI.begin();
pinMode(CS1_PIN, OUTPUT);      pinMode(CS2_PIN, OUTPUT);
digitalWrite(CS1_PIN, HIGH);      digitalWrite(CS2_PIN, HIGH);      }
void loop() {
digitalWrite(CS1_PIN, LOW);
const char* msg1 = "Hello Slave 1";
for (size_t i = 0; i < strlen(msg1); i++) {
SPI.transfer(msg1[i]);     }
digitalWrite(CS1_PIN, HIGH);     delay(1000);
digitalWrite(CS2_PIN, LOW);
const char* msg2 = "Hello Slave 2";
for (size_t i = 0; i < strlen(msg2); i++) {
SPI.transfer(msg2[i]);     }
digitalWrite(CS2_PIN, HIGH);     delay(1000);      }
```

LDR Photoresistor Sensor Module

```
int sensorPin = 34;     int ledPin = 25;     int threshold = 2000;
void setup() {
pinMode(ledPin, OUTPUT);      Serial.begin(115200);      }
void loop() {
int sensorValue = analogRead(sensorPin);
Serial.println(sensorValue);     }
if (sensorValue < threshold) {
digitalWrite(ledPin, HIGH);
else {
digitalWrite(ledPin, LOW);     }      delay(100);     }
```

DHT11 Temperature and Humidity Sensor module

```
#include "DHT.h"
#define DHTPIN 15          #define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
Serial.begin(115200);     dht.begin();      }
void loop() {
float h = dht.readHumidity();      float t = dht.readTemperature();
if (isnan(h) || isnan(t)) {
Serial.println("Failed to read from DHT sensor!");
return;      }
Serial.print("Humidity: ");    Serial.print(h);      Serial.print(" %\t");
Serial.print("Temperature: ");     Serial.print(t);     Serial.println(" *C");
delay(2000);    }
```

Ultrasonic Sensor Module HC-SR04P

```
const int trigPin = 5;          const int echoPin = 18;
void setup() {
Serial.begin (115200);
pinMode(trigPin, OUTPUT);          pinMode (echoPin, INPUT);     }
void loop() {
long duration;          int distance;
digitalWrite(trigPin, LOW);     delayMicroseconds (2);
digitalWrite(trigPin, HIGH);     delayMicroseconds (10);
digitalWrite(trigPin, LOW);
```

```
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");     Serial.print(distance);     Serial.println(" cm");
delay (1000);     }
```

LDR + DHT11 + Ultrasonic

```
#include "DHT.h"
const int trigPin = 5;  // Ultrasonic Sensor
const int echoPin = 18;  // Ultrasonic Sensor
#define DHTPIN 15     #define DHTTYPE DHT11 // DHT Sensor
DHT dht(DHTPIN, DHTTYPE);     // DHT Sensor
const int sensorPin = 34;   const int ledPin = 25; // LDR
const int threshold = 2000;    // LDR
void setup() {
 Serial.begin(115200);
 pinMode(trigPin, OUTPUT);     pinMode(echoPin, INPUT); // Ultrasonic
 pinMode(ledPin, OUTPUT); // LDR Setup
 dht.begin(); // DHT11 Setup     }
void loop() {
 long duration;     int distance_mm;     // Ultrasonic
 digitalWrite(trigPin, LOW);      delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);      delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 distance_mm = duration * 0.34 / 2; // Convert cm. ---> mm.
 Serial.print("Distance: ");     Serial.print(distance_mm);
 Serial.println(" mm"); // Ultrasonic
 float h = dht.readHumidity(); // DHT
 float tC = dht.readTemperature(); // Revice ---> Celsius
 float tF = tC * 9.0 / 5.0 + 32.0; // Convert Celsius ---> Fahrenheit
 if (isnan(h) || isnan(tC)) {
  Serial.println("Failed to read from DHT sensor!");
 } else { //DHT
  Serial.print("Humidity: "); Serial.print(h);  Serial.print(" %\t");
  Serial.print("Temperature: "); Serial.print(tF);  Serial.println(" °F");   }
 int sensorValue = analogRead(sensorPin); // LDR
 Serial.print("Sensor Value: ");     Serial.println(sensorValue);
 if (sensorValue < threshold) {
  digitalWrite(ledPin, HIGH);
 } else {
  digitalWrite(ledPin, LOW);     // LDR     }      delay(2000);     }
```

Light Module

```
const int redPin = 5;   const int greenPin = 18;  const int bluePin = 19;
void setup() {
pinMode(redPin, OUTPUT);      pinMode(greenPin, OUTPUT);
pinMode(bluePin, OUTPUT);  }
void loop() {
digitalWrite(redPin, HIGH);     delay(1000);      digitalWrite(redPin, LOW);
digitalWrite(greenPin, HIGH);  delay(1000);   digitalWrite(greenPin, LOW);
digitalWrite(bluePin, HIGH);  delay(1000);    digitalWrite(bluePin, LOW); }
```

Servo Motor SG90

```
#include <ESP32Servo.h>
Servo myservo;
int servoPin = 5;          int angle = 0;
void setup() {
myservo.attach(servoPin);     }
void loop() {
for (angle = 0; angle <= 90; angle += 5) {
myservo.write(angle);          delay(100);     }
for (angle = 90; angle >= 0; angle -= 5) {
myservo.write(angle);          delay(100);     }     }
```

Stepper Motor

```
#include <Stepper.h>
const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21);
void setup() {
myStepper.setSpeed(10);      Serial.begin(115200);      }
void loop() {
Serial.println("Clockwise");
myStepper. step(stepsPerRevolution);      delay(1000);
Serial.println("Counterclockwise");
myStepper. step(-stepsPerRevolution);     delay(1000);     }
```

5V Relay 1-Channel Active High

```
const int relayPin = 5;
void setup() {
pinMode(relayPin, OUTPUT);          digitalWrite(relayPin, LOW);      }
void loop() {
digitalWrite(relayPin, HIGH);     delay(2000);
digitalWrite(relayPin, LOW);     delay(2000);      }
```

LED + Stepper motor+ Relay

```
#include <Stepper.h>
const int relayPin = 23;
const int stepsPerRevolution = 2048
const int redPin=25;   const int yellowPin=26;   const int greenPin=27;
Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21);
```

```
void setup() {
 myStepper.setSpeed(12);     Serial.begin(115200);
 pinMode(relayPin, OUTPUT);  digitalWrite(relayPin, LOW); // รีเลย์ปิด
 pinMode(redPin, OUTPUT);   pinMode(yellowPin, OUTPUT);
 pinMode(greenPin, OUTPUT);      }
void loop() {
 digitalWrite(relayPin, HIGH);      // เปิดรีเลย์
 digitalWrite(redPin, HIGH);        // เปิดไฟแดง  delay(1000);
 myStepper.step(-stepsPerRevolution);   // หมุนทวนเข็ม delay(3000);
 digitalWrite(redPin, LOW);         // ปิดไฟแดง
 digitalWrite(relayPin, LOW);       // ปิดรีเลย์ (ไม่จ่ายไฟ)
 digitalWrite(yellowPin, HIGH);     // เปิดไฟเหลือง
 delay(1000);   // ไฟเหลือติด 1 วินาที delay(3000);     // ค้าง 3 วิ
 digitalWrite(yellowPin, LOW);      // ปิดไฟเหลือง
 digitalWrite(relayPin, HIGH);      // เปิดรีเลย์
 digitalWrite(greenPin, HIGH);      // เปิดไฟเขียว   delay(1000);
 myStepper.step(stepsPerRevolution);   // หมุนตามเข็ม  delay(3000);
 digitalWrite(greenPin, LOW);       // ปิดไฟเขียว    }
```

Button Switch

```
const int buttonPin = 18;     const int redLedPin = 5;
const int yellowLedPin = 19;     const int greenLedPin = 21;
int buttonState = 0;          int ledState = 0;
void setup() {
pinMode(buttonPin, INPUT_PULLUP);   pinMode(redLedPin, OUTPUT);
pinMode(yellowLedPin, OUTPUT);     pinMode(greenLedPin, OUTPUT);
digitalWrite(redLedPin, LOW);          digitalWrite(yellowLedPin, LOW);
digitalWrite(greenLedPin, LOW);        Serial.begin(115200);  }
void loop() {
buttonState = digitalRead(buttonPin);
if (buttonState == LOW) {
ledState = !ledState;
if (ledState) {
digitalWrite(redLedPin, HIGH);      digitalWrite(yellowLedPin, LOW);
digitalWrite(greenLedPin, LOW);     Serial.println("Red LED ON");
} else {
digitalWrite(redLedPin, LOW);       digitalWrite(yellowLedPin, HIGH);
digitalWrite(greenLedPin, LOW);     Serial.println("Yellow LED ON");   }
delay(300);     }
delay(5000);          digitalWrite(redLedPin, LOW);
digitalWrite(yellowLedPin, LOW);      digitalWrite(greenLedPin, HIGH);
Serial.println("Green LED ON");          delay(300);   }
```

Potentiometer VR 10 Kohm

```
const int potPin = 34;      const int ledPin = 5;
void setup() {
pinMode(ledPin, OUTPUT);       Serial.begin(115200)     }
void loop() {
int potValue = analogRead(potPin);
int pwmValue = map(potValue, 0, 4095, 0, 255);
analogWrite(ledPin, pwmValue);
Serial.print("Potentiometer Value: ");       Serial.print(potValue);
Serial.print(" - PWM Value: ");          Serial.println(pwmValue);
delay(100);     }
```

Rocker Switch

```
const int switchPin = 18;      const int ledPin = 5;
void setup() {
pinMode(switchPin, INPUT_PULLUP);      pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, LOW);     Serial.begin(115200);      }
void loop() {
int switchState = digitalRead(switchPin);
if (switchState == LOW) {
digitalWrite(ledPin, HIGH);        Serial.println("LED ON");
} else {
digitalWrite(ledPin, LOW);     Serial.println("LED OFF");     delay(100);    }
```

Stepper motor + Potentiometer

```
#include <Stepper.h>
const int stepsPerRevolution = 200;  // 1 รอบ = 200 steps
const int potPin = 34;               // ขาโพเทน
Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21);  // IN1-IN4
void setup() {
 Serial.begin(115200);    myStepper.setSpeed(60);  // รอบต่อนาที      }
void loop() {
 int potValue = analogRead(potPin);  // อ่านค่าโพเทน (0–4095)
 // แปลงค่าเป็นจำนวนรอบที่ต้องการหมุน (0–20 รอบ)
 int rotations = map(potValue, 0, 4095, 0, 20);
 // คำนวณจำนวนสเต็ปที่จะหมุน
 int stepsToMove = rotations * stepsPerRevolution;
 Serial.print("Potentiometer: ");       Serial.print(potValue);
 Serial.print(" ⟶ Rotations: ");       Serial.print(rotations);
 Serial.print(" ⟶ Steps: ");          Serial.println(stepsToMove);
 // หมุนตามจำนวนรอบไปข้างหน้า
 myStepper.step(stepsToMove);     delay(1000);
 // หมุนกลับมาจุดเดิม
 myStepper.step(-stepsToMove);     delay(1000);     }
```

Stepper motor + switch

```
#include <Stepper.h>
const int switchPin = 22;       const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 21, 18, 19, 5);
void setup() {
 pinMode(switchPin, INPUT_PULLUP);    myStepper.setSpeed(20);
 Serial.begin(115200);          }
void loop() {
 int switchState = digitalRead(switchPin);
 if (switchState == LOW) {
  myStepper.step(stepsPerRevolution);
  Serial.println("Rotating Clockwise");
 } else {
  myStepper.step(-stepsPerRevolution);
  Serial.println("Rotating Counter-Clockwise");  }     }
```

พัดลม + เปิดปิดไฟ

```
#include "DHT.h"
#include <Stepper.h>
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const int relayPin = 23; // ควบคุมไฟเลี้ยงพัดลม
const int stepsPerRevolution = 2048; // ค่า Step ของมอเตอร์
Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21);
float X = 30.0; // อุณหภูมิฐาน (°C)
void setup() {
 Serial.begin(115200);          dht.begin();
 myStepper.setSpeed(12);      pinMode(relayPin, OUTPUT);
 digitalWrite(relayPin, LOW); // เริ่มต้นปิดรีเลย์          }
void loop() {
 float temp = dht.readTemperature();
 if (isnan(temp)) {
  Serial.println("อ่านค่า DHT11 ไม่สำเร็จ");     delay(2000);
  return;     }
 Serial.print("Temp: ");     Serial.println(temp);
 if (temp > X + 1) {
  // พัดลมพัดอากาศเข้า (หมุนทวนเข็ม)
  digitalWrite(relayPin, HIGH);
  myStepper.step(-stepsPerRevolution);
  Serial.println("พัดลมหมุนทวนเข็ม ⟶ อากาศเข้า");       }
 else if (temp < X) {
  // พัดลมพัดอากาศออก (หมุนตามเข็ม)
  digitalWrite(relayPin, HIGH);
  myStepper.step(stepsPerRevolution);
  Serial.println("พัดลมหมุนตามเข็ม ⟶ อากาศออก");     }
 else {
  // หยุด
  digitalWrite(relayPin, LOW);     Serial.println("หยุดพัดลม");
  delay(2000);     }      }
```

```
const int ledPin = 2;     // ขา LED
const int ldrPin = 34;    // ขา LDR ต่อกับ ADC
int lightValue = 0;
int thresholdValue = 2000; // ค่ากำหนดความมืด (ปรับได้)
void setup() {
 Serial.begin(115200);     pinMode(ledPin, OUTPUT);          }
void loop() {// อ่านค่าจาก LDR
 lightValue = analogRead(ldrPin);
 if (Serial.available() > 0) { // ตรวจ Serial Command
  String cmd = Serial.readStringUntil('\n');
  cmd.trim(); // ลบช่องว่าง
  if (cmd.equalsIgnoreCase("ON")) {
   digitalWrite(ledPin, HIGH);
   Serial.println("สั่งเปิดไฟจาก Serial");          delay(500);
   return; // ข้ามการตรวจ LDR ในรอบนี้          }
  else if (cmd.equalsIgnoreCase("OFF")) {
   digitalWrite(ledPin, LOW);
   Serial.println("สั่งปิดไฟจาก Serial");          delay(500);
   return;     }          }
 if (lightValue < thresholdValue) { // ตรวจค่าแสงจาก LDR
  digitalWrite(ledPin, HIGH);     Serial.println("มืด ⟶ เปิดไฟ");     }
 else {
  digitalWrite(ledPin, LOW);     Serial.println("สว่าง ⟶ ปิดไฟ");     }
 delay(500);     }
```