|  |  |  |
| --- | --- | --- |
|  | **ชุดการสอน**  **หน่วยสมรรถนะ : UOC 1** พื้นฐาน ESP32 และการเขียนโปรแกรมเบื้องต้น | **หน่วยที่1** |
| **รหัสวิชา 21901-2007 วิชา เทคโนโลยีระบบสมองกลฝังตัวและไอโอที หน่วยกิต 2** | **สัปดาห์ที่. 1-2** |
| **สมรรถนะย่อย : EOC** โปรแกรมควบคุม LED 7 หลอด ทำงานตามเงื่อนไข | **ทฤษฎี 2 ชม.**  **ปฏิบัติ 2 ชม.** |

**1. ผลลัพธ์การเรียนรู้**

ผู้เรียนสามารถเขียนโปรแกรมบน ESP32 เพื่อควบคุม LED 7 หลอดให้ทำงานตามเงื่อนไขที่กำหนดได้อย่างถูกต้อง และอธิบายหลักการทำงานของคำสั่งที่ใช้ได้

**2. อ้างอิงมาตรฐานอาชีพ**

-

**3. หน่วยสมรรถนะ : UOC 1** พื้นฐาน ESP32 และการเขียนโปรแกรมเบื้องต้น

**หน่วยสมรรถนะย่อย : EOC1.1** **EOC โปรแกรมควบคุม LED 7 หลอด ทำงานตามเงื่อนไข**

**4. เกณฑ์การประเมิน (Performance Criteria : PC)**

4.1 กำหนดค่า GPIO แต่ละขาของ LED ได้ถูกต้อง

4.2 เขียนโค้ดกำหนดเงื่อนไขการทำงานของ LED ได้ถูกต้อง (if, for, while)

4.3 ใช้โครงสร้างโปรแกรมพื้นฐานของ Arduino IDE ได้ถูกต้อง (setup(), loop())

**5. ขอบเขตการปฏิบัติ (Range: R)**

5.1 ใช้ ESP32 Development Board

5.2 LED 7 หลอดเชื่อมต่อกับ GPIO ที่กำหนด

5.3 ใช้ Arduino IDE ในการเขียนโปรแกรม

**6. จุดประสงค์เชิงพฤติกรรม**

6.1 อธิบายโครงสร้างโปรแกรมพื้นฐานใน Arduino IDE ได้

6.2 เขียนโปรแกรมควบคุม LED 7 หลอดให้ทำงานตามเงื่อนไขได้

6.3 ใช้ตัวแปรและโครงสร้างควบคุม (if, for, while) ในการเขียนโปรแกรมได้

6.4 ทดสอบและปรับแก้โค้ดให้ทำงานได้ถูกต้อง

**7. เนื้อหา**

**1. ESP 32 คืออะไร** ESP32 เป็นไมโครคอนโทรลเลอร์ที่พัฒนาโดย Espressif Systems ซึ่งมีฟีเจอร์ที่เหนือกว่าตัว Arduino ทั่วไปในหลายๆ ด้าน เช่น การเชื่อมต่อ Wi-Fi และ Bluetooth ในตัว, ความสามารถในการประมวลผลที่สูงขึ้น และขนาดที่เล็กกว่า

**2. ประเภทขาใช้งาน**

* **ขาที่เป็นอินพุตอย่างเดียว**

**ขา GPIO34, GPIO35, GPIO36, GPIO39 เป็นอินพุตอย่างเดียว ไม่มีวงจร Pull-up, Pull-down รองรับคำสั่ง analogRead(), digitalRead() เท่านั้น คำสั่งอื่น เช่น digitalWrite(), PWM ไม่รองรับ**

* **Strapping Pins**

**Strapping Pins เป็นขาเกี่ยวข้องกับการทำงานของ ESP32 ในระหว่างการบูตโปรแกรม แบ่งเป็นขาที่ใช้งานแล้วอาจทำให้ ESP32 ไม่สามารถทำงานได้ กับขาที่ใช้งานได้ปกติแต่อาจมีสัญญาณรบกวนออกมาระหว่างบูตโปรแกรม**

**ขาที่ไม่ควรนำมาใช้งานเลย ประกอบด้วยขา GPIO0, GPIO2 ซึ่งเกี่ยวข้องกับการบูตโปรแกรม**

**ขาที่ใช้งานได้ปกติแต่อาจมีสัญญาณรบกวนออกมาระหว่างบูตโปรแกรม ประกอบด้วยขา GPIO12, GPIO15, GPIO5 ไม่ควรนำไปใช้ต่อกับอุปกรณ์ I2C เพราะอาจทำให้เกิดการทำงานผิดปกติของอุปกรณ์ I2C ได้**

* **ขาที่ต่อชิป Flash และ PSRAM**

**ขาที่ต่อชิป Flash เป็นขาที่ห้ามนำมาใช้เด็ดขาดเพราะกระทบกับการอ่านโปรแกรมที่เคยอัพโหลดไว้ หรือทำให้อัพโหลดโปรแกรมไม่เข้า ประกอบด้วยขา GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8**

**ขาที่ต่อชิป PSRAM เป็นขาที่เมื่อนำมาใช้จะไม่สามารถใช้งาน PSRAM ได้ ซึ่ง PSRAM จะมีในบอร์ดที่ระบุอย่างชัดเจนว่ามี PSRAM หากบอร์ดไม่มี PSRAM อยู่แล้วก็ไม่จำเป็นต้องสนใจ แต่หากบอร์ดมี PSRAM ต้องไม่นำขา GPIO16, GPIO17 มาใช้งาน หากนำมาใช้ต้องแน่ใจว่าโปรแกรมที่เขียนไม่เรียกใช้ PSRAM**

* **ขาอัพโหลดโปรแกรม**

**นอกจากขา Strapping Pins ที่เกี่ยวข้องกับการอัพโหลดโปรแกรมและการบูตโปรแกรม ยังมีขา RX, TX ที่ใช้ในการอัพโหลดโปรแกรม หากนำขาดังกล่าวไปใช้ จะทำให้อัพโหลดโปรแกรมไม่ได้ (แต่โปรแกรมที่เคยอัพโหลดไว้ยังทำงานได้ปกติ)**

* **ขา Digital Input**

**ขาที่ใช้อ่านค่าดิจิทัลได้ (ใช้คำสั่ง digitalRead() ได้) คือทุกขาที่ ไม่ใช่ Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา Digital Output**

**ขาที่เขียนค่าดิจิทัลได้ (ใช้คำสั่ง digitalWrite() ได้) คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา Analog Input**

**ขาที่อ่านค่าแอนะล็อกได้ (ใช้คำสั่ง analogRead() ได้) อ่านค่าได้ 0 ถึง 3.3V แบ่งขาออกเป็น 2 ชุด คือ ADC1 และ ADC2 โดย ADC1 สามารถใช้งานได้เลย ไม่มีเงื่อนไขอะไร ส่วน ADC2 จะใช้งานได้เมื่อปิดใช้ WiFi เท่านั้น**

**ADC1 ใช้งานได้เลย ประกอบด้วยขา GPIO36, GPIO39, GPIO34, GPIO35, GPIO32, GPIO33**

**ADC2 ต้องปิด WiFi จึงจะใช้งานได้ ประกอบด้วยขา GPIO25, GPIO26, GPIO27, GPIO14, GPIO12, GPIO13, GPIO15, GPIO4**

* **ขา PWM**

**ขาที่ใช้ PWM ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา I2C**

**I2C แบ่งเป็น 2 ชุด คือ I2C0 และ I2C1 โดยปกติใช้เฉพาะ I2C0 ซึ่ง I2C0 ค่าเริ่มต้นกำหนดให้อยู่ที่ขา SDA = GPIO21, SCL = GPIO22 สามารถเปลี่ยนเป็นขาอื่นได้โดยกำหนดในคำสั่ง Wire.begin() ตัวอย่างการย้ายขา SCL ไปที่ GPIO4 และ SDA ไป GPIO5 มีดังนี้**

* **Wire.begin(5, 4); // SCL, SDA**

**ขา SCL, SDA ที่ย้ายไปใช้ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา SPI**

**SPI แบ่งเป็น 2 ชุด คือ HSPI และ VSPI โดยปกติใช้เฉพาะ VSPI ซึ่ง VSPI ค่าเริ่มต้นกำหนดให้อยู่ที่ขา CS = GPIO5, SCK = GPIO18, MISO = GPIO19, MOSI = GPIO23 สามารถเปลี่ยนเป็นขาอื่นได้โดยกำหนดในคำสั่ง SPI.begin() ตัวอย่างการย้ายขา VSPI ไปที่ SCK = GPIO25, MISO = GPIO26, MOSI = GPIO27 และ CS = GPIO15 มีดังนี้**

**SPI.begin(25, 26, 27, 15); // SCK, MISO, MOSI, CS**

**ขา SCK, MISO, MOSI, CS ที่ย้ายไปใช้ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา UART**

**UART แบ่งเป็น 3 ชุด คือ Serial0, Serial1 และ Serial2 โดย Serial0 ใช้อัพโหลดโปรแกรมและสื่อสารกับคอมพิวเตอร์ อยู่ที่ขา TX, RX ไม่ควรนำมาใช้ต่ออุปกรณ์อื่น ส่วน Serial1 และ Serial2 ใช้งานได้อิสระ**

**Serial1 ค่าเริ่มต้นกำหนดให้อยู่ที่ขา RX = GPIO26, TX = GPIO27 เปลี่ยนขาได้โดยใช้คำสั่ง Serial1.setPins()**

**Serial2 ค่าเริ่มต้นกำหนดให้อยู่ที่ขา RX = GPIO4, TX = GPIO25 เปลี่ยนขาได้โดยใช้คำสั่ง Serial2.setPins()**

**ขา TX, RX ที่ย้ายไปใช้ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา CAN**

**CAN bus ขา TX, RX ไม่ได้ต่อกับขาใด ๆ เป็นค่าเริ่มต้น ก่อนเริ่มใช้งาน CAN bus ต้องกำหนดขา TX, RX ลงในโค้ดโปรแกรมก่อน โดยขา TX, RX ที่ใช้ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

* **ขา I2S**

**I2S แบ่งเป็น 2 ชุด คือ I2S0 และ I2S1 โดยขา DIN, DOUT, BCLK, WS ไม่ได้ต่อกับขาใด ๆ เป็นค่าเริ่มต้น ก่อนเริ่มใช้งาน I2S ต้องกำหนดขาทั้งหมดลงในโค้ดโปรแกรมก่อน โดยขาที่ใช้ได้ คือทุกขาที่ ไม่ใช่ ขาที่เป็นอินพุตอย่างเดียว, Strapping Pins, ขาที่ต่อชิป Flash และ PSRAM, ขาอัพโหลดโปรแกรม โดยสรุปคือขา GPIO0, GPIO2, GPIO9, GPIO10, GPIO11, GPIO6, GPIO7, GPIO8, RX, TX, GPIO34, GPIO35, GPIO36, GPIO39 ใช้ไม่ได้ ที่เหลือใช้ได้หมด**

| **GPIO** | **Digital Input** | **Digital Output** | **Analog Input** | **PWM** | **I2C / SPI / UART / CAN / I2S** |
| --- | --- | --- | --- | --- | --- |
| **IO36** |  |  |  |  |  |
| **IO39** |  |  |  |  |  |
| **IO34** |  |  |  |  |  |
| **IO35** |  |  |  |  |  |
| **IO32** |  |  |  |  |  |
| **IO33** |  |  |  |  |  |
| **IO25** |  |  |  |  |  |
| **IO26** |  |  |  |  |  |
| **IO27** |  |  |  |  |  |
| **IO14** |  |  |  |  |  |
| **IO12** |  |  |  |  |  |
| **IO13** |  |  |  |  |  |
| **IO9** |  |  |  |  |  |
| **IO10** |  |  |  |  |  |
| **IO11** |  |  |  |  |  |
| **IO6** |  |  |  |  |  |
| **IO7** |  |  |  |  |  |
| **IO8** |  |  |  |  |  |
| **IO15** |  |  |  |  |  |
| **IO2** |  |  |  |  |  |
| **IO0** |  |  |  |  |  |
| **IO4** |  |  |  |  |  |
| **IO16** |  |  |  |  |  |
| **IO17** |  |  |  |  |  |
| **IO5** |  |  |  |  |  |
| **IO18** |  |  |  |  |  |
| **IO19** |  |  |  |  |  |
| **IO21** |  |  |  |  |  |
| **RX** |  |  |  |  |  |
| **TX** |  |  |  |  |  |
| **IO22** |  |  |  |  |  |
| **IO23** |  |  |  |  |  |

**3. ติดตั้ง Arduino IDE**

* ดาวน์โหลด Arduino IDE จาก Arduino website
* ติดตั้งโปรแกรมโดยทำตามขั้นตอนใน installe

**4. ติดตั้ง ESP32 Board**

* เปิด Arduino IDE -> ไปที่ "File" -> "Preferences
* เพิ่ม URL: https://dl.espressif.com/dl/package\_esp32\_index.json ใน "Additional Boards Manager URLs
* ไปที่ "Tools" -> "Board" -> "Board Manager"
* ค้นหา "ESP32" และติดตั้งแพ็คเกจ

**5. สิ่งที่ควรรู้ก่อนเริ่ม**

**5.1 โครงสร้างโปรแกรมพื้นฐาน**

**// ฟังก์ชันเริ่มต้น ทำงานครั้งเดียวตอนเปิดบอร์ดหรือกด Reset**

**void setup() {**

**// ใช้สำหรับกำหนดค่าเริ่มต้น เช่น ตั้งค่า pin, เริ่ม Serial**

**}**

**// ฟังก์ชันวนซ้ำ ทำงานไปเรื่อย ๆ**

**void loop() {**

**// ใช้สำหรับคำสั่งที่ทำซ้ำ เช่น อ่านค่าเซนเซอร์ ควบคุม LED**

**}**

**5.2 ตัวแปร (Variables)**

**int ledPin = 2; // เก็บค่าเลขขา GPIO**

**int counter = 0; // เก็บตัวเลขจำนวนเต็ม**

**float voltage = 3.3; // เก็บตัวเลขทศนิยม**

**char letter = 'A'; // เก็บตัวอักษร**

**5.3 ตัวแปรแบบ Global (ใช้งานได้ทั้งโปรแกรม)**

**int globalVar = 10; // global variable**

**void setup() {**

**Serial.begin(115200);**

**}**

**void loop() {**

**Serial.println(globalVar); // ใช้ได้ทุกที่**

**}**

**5.4 ตัวแปรแบบ Local (ใช้งานเฉพาะในฟังก์ชัน)**

**void loop() {**

**int localVar = 5; // ใช้ได้เฉพาะใน loop()**

**Serial.println(localVar);**

**}**

**5.5 คำสั่งควบคุมขา GPIO**

**pinMode(pin, mode); // กำหนดโหมดของขา**

**digitalWrite(pin, value); // เขียนค่า (HIGH / LOW)**

**digitalRead(pin); // อ่านค่าดิจิทัล (0 / 1)**

**analogRead(pin); // อ่านค่าอนาล็อก (0–4095 บน ESP32)**

**analogWrite(pin, value); // เขียนค่า PWM (0–255)**

**5.6 ฟังก์ชัน (Functions)**

**// สร้างฟังก์ชัน**

**void blinkLED(int pin, int timeDelay) {**

**digitalWrite(pin, HIGH);**

**delay(timeDelay);**

**digitalWrite(pin, LOW);**

**delay(timeDelay);**

**}**

**void loop() {**

**blinkLED(led, 500); // เรียกใช้ฟังก์ชัน**

**}**

**8. คู่มือ / Data**

**9. เอกสารอ้างอิง**