สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

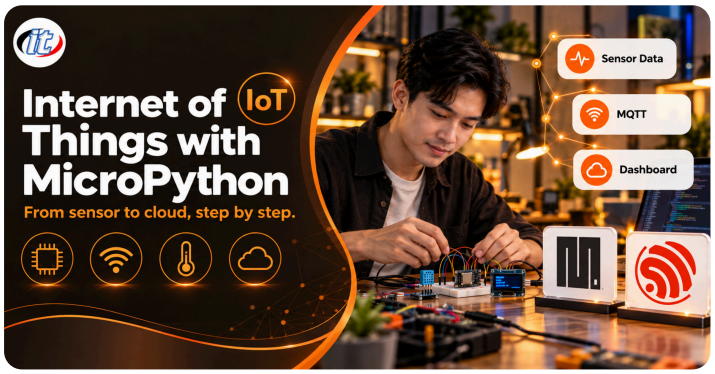
COURSE ID
PRC-35
IOT / MICROPYTHON

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

Internet of Things with MicroPython

สร้างอุปกรณ์ IoT บน ESP32 ด้วย MicroPython ตั้งแต่ต่อวงจรถึงขึ้น Cloud

- 18 ชั่วโมง
- 3 วัน
- พื้นฐาน **เริ่มจากศูนย์ได้**
- ระดับ **Beginner**



ระยะเวลา: 3 วัน
18 ชั่วโมง (3 วัน วันละ 6 ชม.) · 09:00-16:00 น.

รูปแบบ: Classroom / Hybrid
Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house

รายละเอียดหลักสูตร
www.itgenius.co.th

ESP32 เป็นไมโครคอนโทรลเลอร์ราคาประหยัดที่มี Wi-Fi และ Bluetooth ในตัว จึงเป็นบอร์ดยอดนิยมสำหรับงาน Internet of Things ตั้งแต่ระบบวัดอุณหภูมิในบ้าน ฟาร์มอัจฉริยะ ไปจนถึงงานเฝ้าระวังเครื่องจักร เมื่อใช้คู่กับ MicroPython ซึ่งเป็น Python 3 ที่ทำงานบนบอร์ดได้โดยตรง ผู้เริ่มต้นก็เขียนโปรแกรมสั่งงานฮาร์ดแวร์และทดลองคำสั่งได้ทันทีโดยไม่ต้องเรียนภาษา C ก่อน

หลักสูตรนี้เป็นฉบับปรับปรุงใหม่สำหรับ MicroPython 1.29 พาผู้เรียนตั้งแต่พื้นฐานอิเล็กทรอนิกส์ แฟลช Firmware ลงบอร์ด ESP32 เขียน MicroPython ควบคุม GPIO อ่านค่าเซนเซอร์ DHT แสดงผลบนจอ LCD และ OLED และสั่งงานอุปกรณ์ไฟฟ้าผ่านรีเลย์ จากนั้นพาอุปกรณ์ขึ้นเครือข่ายด้วย Wi-Fi ส่งข้อมูลผ่าน MQTT ไปแสดงบน Dashboard แจ้งเตือนผ่าน LINE Messaging API และบันทึกข้อมูลขึ้น Firebase แล้วปิดท้ายด้วย Workshop ระบบควบคุมพัดลมอัตโนมัติที่ใช้งานได้จริง (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Beginner)

เครื่องมือ: µP MicroPython · ESP ESP32 · MQ MQTT · Th Thonny · IoT IoT Dashboard

ราคาค่าอบรม · PUBLIC / IN-HOUSE

5,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide โค้ด MicroPython ครบทุก Lab แผนผังการต่อวงจร และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

เพิ่มเพื่อน LINE

02-570-8449 · 088-807-9770 · @itgenius · www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- อธิบายสถาปัตยกรรม IoT และเลือกใช้ HTTP หรือ MQTT ให้เหมาะกับงานได้
- ต่อวงจรอิเล็กทรอนิกส์พื้นฐานบน Breadboard ได้อย่างถูกต้องและปลอดภัย
- แฟลช MicroPython ลงบอร์ด ESP32 และเขียนโปรแกรมผ่าน Thonny และ REPL ได้
- ควบคุม GPIO, PWM และอ่านค่า Analog จากอุปกรณ์ต่าง ๆ ได้
- อ่านค่าเซนเซอร์ DHT และแสดงผลบนจอ LCD หรือ OLED ผ่าน I2C ได้
- สั่งงานเครื่องใช้ไฟฟ้าผ่านโมดูลรีเลย์ได้อย่างปลอดภัย
- เชื่อมต่อ Wi-Fi ส่งข้อมูลผ่าน MQTT และแสดงผลบน Dashboard ได้
- ส่งการแจ้งเตือนผ่าน LINE และบันทึกข้อมูลขึ้น Firebase ได้

02 กลุ่มเป้าหมาย Target Audience

- ผู้ที่ต้องการเริ่มต้นเรียนรู้การพัฒนา IoT โดยไม่มีพื้นฐานมาก่อน
- นักเรียน นักศึกษา และครูที่ต้องการทำโครงงานหรือสอนด้าน IoT
- ผู้ที่เขียน Python มาบ้างและอยากต่อยอดมาสั่งงานฮาร์ดแวร์
- ช่างเทคนิค เจ้าของกิจการ หรือ Maker ที่อยากสร้างระบบอัตโนมัติขนาดเล็กเอง

03 จุดเด่นของหลักสูตร Highlights

- **อัปเดตเป็น MicroPython 1.29** ESP32 รุ่นปัจจุบัน, Thonny และ mpremote
- **เริ่มจากศูนย์ได้จริง** ปูพื้นอิเล็กทรอนิกส์และ Python ให้ก่อน
- **เพิ่ม MQTT และ Dashboard** ส่งข้อมูลแบบ IoT จริงและดูผลบน Node-RED
- **แจ้งเตือนและเก็บข้อมูล** LINE Messaging API และ Firebase
- **ลงมือต่อวงจรทุก Lab** จบด้วยระบบพัดลมอัตโนมัติที่ใช้งานได้

04 ความรู้พื้นฐาน Prerequisites

- ไม่จำเป็นต้องมีพื้นฐานอิเล็กทรอนิกส์หรือเขียนโปรแกรมมาก่อน หากเคยเขียน Python มาบ้างจะเรียนได้เร็วขึ้น
- ใช้คอมพิวเตอร์ติดตั้งโปรแกรมและจัดการไฟล์ได้
- โน้ตบุ๊ก Windows หรือ macOS ที่มีพอร์ต USB และสิทธิ์ติดตั้งโปรแกรมและไดรเวอร์
- มีบัญชี LINE และบัญชี Google สำหรับใช้ LINE Developers และ Firebase แบบฟรี

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook มีพอร์ต USB และสาย USB ที่รับส่งข้อมูลได้ เชื่อมต่ออินเทอร์เน็ต
บอร์ดและ Firmware	บอร์ด ESP32 DevKit พร้อม MicroPython 1.29
อุปกรณ์ทดลอง	Breadboard, LED, ปุ่มกด, DHT11 หรือ DHT22, จอ LCD I2C หรือ OLED และโมดูลรีเลย์
ซอฟต์แวร์	Thonny IDE, esptool, mpremote, Node-RED และ MQTT Broker แบบ Mosquitto
บริการออนไลน์	LINE Messaging API และ Firebase Realtime Database บนแผนใช้ฟรี

หมายเหตุ: ปรับปรุงใหม่ปี 2026 จากหลักสูตรเดิม: อัปเดตเป็น MicroPython 1.29 ใช้ Thonny และ mpremote แทนขั้นตอนแบบเก่า เพิ่มการต่อ Wi-Fi และเรียก Web API, MQTT, Dashboard ด้วย Node-RED, จอ OLED, Interrupt, Timer, asyncio และการทำให้อุปกรณ์ทำงานเสถียร ส่วนการแจ้งเตือนใช้ LINE Messaging API แทน LINE Notify ที่ปิดบริการไปแล้ว ชุดอุปกรณ์ทดลองเป็นไปตามรายการที่สถาบันแจ้งก่อนวันอบรม บริการ LINE และ Firebase ใช้บัญชีของผู้เรียนเองบนแผนใช้ฟรี ซึ่งมีโควตาจำกัดและอาจเปลี่ยนเงื่อนไขได้ Lab ที่เกี่ยวกับไฟบ้านสาธิตด้วยอุปกรณ์แรงดันต่ำเพื่อความปลอดภัย

06 ภาพรวมการเรียน 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	เข้าใจสถาปัตยกรรม IoT, พื้นฐานอิเล็กทรอนิกส์และความปลอดภัย, แฟลช MicroPython 1.29 ลงบอร์ด ESP32, เขียนโปรแกรมผ่าน Thonny และ REPL และควบคุม GPIO ทั้ง LED ปุ่มกด Interrupt และ Timer
Day 2 Section 6-10	อ่านค่า Analog และใช้ PWM, วัดอุณหภูมิความชื้นด้วย DHT11 และ DHT22, แสดงผลบนจอ LCD และ OLED ผ่าน I2C, สั่งงานอุปกรณ์ด้วยรีเลย์อย่างปลอดภัย และเชื่อม Wi-Fi เรียก Web API
Day 3 Section 11-15	ส่งและรับข้อมูลด้วย MQTT, สร้าง Dashboard ด้วย Node-RED, แจ้งเตือนผ่าน LINE Messaging API และบันทึกข้อมูลขึ้น Firebase, ทำให้อุปกรณ์ทำงานเสถียร และปิดท้ายด้วยระบบพัดลมอัตโนมัติ

DAY 1

พื้นฐาน IoT, ESP32 และ MicroPython

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 บทนำและสถาปัตยกรรม IoT

- ความหมายของ IoT และตัวอย่างการใช้งานในบ้าน เกษตร และโรงงาน
- องค์ประกอบของระบบ: อุปกรณ์ เครือข่าย Broker หรือ Cloud และแอปแสดงผล
- รู้จักโปรโตคอลที่ใช้บ่อย: HTTP, MQTT และ Bluetooth Low Energy
- ความปลอดภัยพื้นฐานของอุปกรณ์ IoT: รหัสผ่าน Token และการอัปเดต Firmware

2 Lab: พื้นฐานอิเล็กทรอนิกส์สำหรับ IoT

- แรงดัน กระแส ความต้านทาน และกฎของโอห์มแบบใช้งานจริง
- อ่านค่าตัวต้านทาน และใช้ Breadboard กับสาย Jumper
- รู้จัก LED, ตัวเก็บประจุ, ไดโอด, ทรานซิสเตอร์ และรีเลย์
- ระดับสัญญาณ 3.3V ของ ESP32 และข้อควรระวังไม่ให้บอร์ดเสียหาย
- Lab: ใช้มัลติมิเตอร์วัดแรงดันและตรวจวงจรก่อนจ่ายไฟ

3 Lab: ติดตั้ง MicroPython บน ESP32

- รู้จักตระกูล ESP32 รุ่นที่พบบ่อย เช่น ESP32, S3, C3 และ C6 และเลือกให้เหมาะกับงาน
- อ่าน Pinout และขาที่ควรหลีกเลี่ยงในการต่ออุปกรณ์
- ติดตั้งไดรเวอร์ USB และแฟลช MicroPython 1.29 ด้วย Thonny หรือ esptool
- ใช้ REPL ทดลองคำสั่งสด ๆ และอัปโหลดไฟล์ด้วย Thonny และ mpremote
- หน้าที่ของไฟล์ boot.py และ main.py บนบอร์ด

4 Lab: พื้นฐานการเขียนโปรแกรม MicroPython

- MicroPython ต่างจาก Python บนคอมพิวเตอร์อย่างไร
- ตัวแปร เงื่อนไข ลูป และฟังก์ชัน
- List, Dictionary และการจัดการข้อผิดพลาดด้วย try และ except
- แยกโค้ดเป็นโมดูล และติดตั้งไลบรารีเพิ่มด้วย mip
- Lab: เขียนโปรแกรมสั้น ๆ ทดสอบผ่าน REPL และไฟล์บนบอร์ด

5 Lab: ควบคุม GPIO ของ ESP32

- ใช้ machine.Pin เป็น Output เพื่อเปิดปิด LED
- อ่านปุ่มกดเป็น Input พร้อม Pull-up และแก้ปัญหาปุ่มเด้ง (Debounce)
- ตอบสนองเหตุการณ์ทันทีด้วย Interrupt (IRQ)
- ทำงานตามรอบเวลาด้วย Timer แทนการใช้ sleep ยาว ๆ
- Lab: ไฟจราจรจำลองที่กดปุ่มเพื่อข้ามถนนได้

DAY 2

เซนเซอร์ จอแสดงผล รีเลย์ และ Wi-Fi

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: สัญญาณ Analog และ PWM

- อ่านค่า Analog ด้วย machine.ADC จาก Potentiometer และ LDR
- ตั้งช่วงการวัดด้วย Attenuation และอ่านค่าเป็นแรงดันด้วย read_uv
- สร้างสัญญาณ PWM ด้วย duty_u16 เพื่อหรี่ไฟ LED และสร้างเสียงจาก Buzzer
- Lab: ปรับความสว่าง LED ตามค่าแสงรอบตัว

7 Lab: วัดอุณหภูมิและความชื้นด้วย DHT11 และ DHT22

- เซนเซอร์ DHT11 กับ DHT22 ต่างกันอย่างไร และเลือกใช้เมื่อไร
- ต่อเซนเซอร์กับ ESP32 และอ่านค่าด้วยโมดูล dht
- เว้นจังหวะการอ่าน และจัดการกรณีอ่านค่าไม่สำเร็จ
- Lab: บันทึกค่าอุณหภูมิและความชื้นลงไฟล์บนบอร์ด

8 Lab: จอแสดงผล LCD และ OLED ผ่าน I2C

- หลักการของบัส I2C และการค้นหาอุปกรณ์ด้วย i2c.scan()
- ต่อจอ LCD 16×2 แบบ I2C และแสดงข้อความหลายบรรทัด
- ต่อจอ OLED SSD1306 แสดงตัวเลข กราฟิกง่าย ๆ และไอคอน
- Lab: แสดงอุณหภูมิและความชื้นจาก DHT บนจอแบบเรียลไทม์

9 Lab: ควบคุมอุปกรณ์ไฟฟ้าผ่านรีเลย์

- รู้จักโมดูลรีเลย์แบบ Single Channel และขา COM, NO, NC
- รีเลย์แบบ Active Low และ Active High ต่างกันอย่างไรในโค้ด
- ความปลอดภัยเมื่อทำงานกับไฟบ้าน และทางเลือกอย่าง SSR
- สั่งเปิดปิดตามเงื่อนไขพร้อม Hysteresis ไม่ให้รีเลย์ตัดต่อถี่เกินไป
- Lab: เปิดพัดลมจำลองเมื่ออุณหภูมิเกินค่าที่ตั้ง

10 Lab: เชื่อมต่อ Wi-Fi และเรียก Web API

- เชื่อมต่อ Wi-Fi ด้วย network.WLAN และเชื่อมต่อใหม่อัตโนมัติเมื่อหลุด
- แยกรหัส Wi-Fi และ Token ไว้ในไฟล์ตั้งค่า ไม่เขียนลงโค้ดหลัก
- ตั้งเวลาบอร์ดจากอินเทอร์เน็ตด้วย NTP
- เรียก Web API ด้วยโมดูล requests และอ่านผลแบบ JSON
- Lab: ดึงข้อมูลสภาพอากาศจาก Web API มาแสดงบนจอ

DAY 3

เชื่อมต่อ Cloud และโปรเจกต์ IoT จริง

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: ส่งและรับข้อมูลด้วย MQTT

- แนวคิด Broker, Topic, Publish, Subscribe และ QoS
- ติดตั้ง Mosquitto เป็น Broker และทดสอบด้วยเครื่องมือฝั่งคอมพิวเตอร์
- ติดตั้ง umqtt.simple ด้วย mip และส่งค่าเซนเซอร์เป็น JSON
- Subscribe คำสั่งจากภายนอกเพื่อเปิดปิดรีเลย์
- ตั้งชื่อ Topic ให้เป็นระบบ และใช้ Username กับ Password ป้องกัน Broker

12 Lab: สร้าง Dashboard ด้วย Node-RED

- ติดตั้ง Node-RED และรับข้อมูลจาก MQTT Broker
- แสดงค่าด้วย Gauge และกราฟย้อนหลังบน Dashboard
- เพิ่มปุ่มและสวิตช์บน Dashboard เพื่อสั่งงานอุปกรณ์
- Lab: Dashboard ติดตามอุณหภูมิและควบคุมพัดลมจากมือถือ

13 Lab: แจ้งเตือนผ่าน LINE และบันทึกข้อมูลบน Firebase

- สร้าง LINE Official Account และ Messaging API Channel ใน LINE Developers Console
- ส่ง Push Message จาก ESP32 ผ่าน HTTPS ด้วย Channel Access Token
- สร้าง Firebase Realtime Database และกำหนด Security Rules เบื้องต้น
- บันทึกและอ่านข้อมูลผ่าน REST API ของ Firebase
- ป้องกันการแจ้งเตือนซ้ำถี่เกินไปด้วยเงื่อนไขและช่วงเวลาพัก

14 ทำให้อุปกรณ์ IoT ทำงานเสถียร

- ทำหลายงานพร้อมกันด้วย asyncio เช่น อ่านเซนเซอร์และรอคำสั่งไปพร้อมกัน
- ใช้ Watchdog Timer รีสตาร์ตบอร์ดเมื่อโปรแกรมค้าง
- ประหยัดพลังงานด้วย Deep Sleep สำหรับอุปกรณ์ที่ใช้แบตเตอรี่
- จัดการเมื่อ Wi-Fi หรือ Broker หลุด และแนวทางไล่แก้ปัญหาที่พบบ่อย

15 Workshop: ระบบควบคุมพัดลมอัตโนมัติ

- ออกแบบระบบและต่อวงจร DHT, จอแสดงผล และรีเลย์ครบชุด
- เขียนโปรแกรมควบคุมพัดลมตามอุณหภูมิและความชื้นพร้อมโหมดสั่งงานเอง
- ส่งข้อมูลขึ้น Dashboard ผ่าน MQTT และบันทึกขึ้น Firebase
- แจ้งเตือนผ่าน LINE เมื่อค่าเกินเกณฑ์ที่ตั้งไว้
- ทดสอบ ปรับปรุง และนำเสนอผลงาน

07 เรียนจบแล้วได้อะไร Outcomes

- **ต่อวงจรพื้นฐานได้** Breadboard, LED, ปุ่ม และรีเลย์
- **เขียน MicroPython บน ESP32 ได้** Thonny, REPL และ mpremote
- **ใช้เซนเซอร์และจอได้** DHT, ADC, LCD และ OLED
- **พาอุปกรณ์ขึ้นเครือข่ายได้** Wi-Fi, Web API และ MQTT
- **สร้าง Dashboard และแจ้งเตือนได้** Node-RED, LINE และ Firebase
- **สร้างโปรเจกต์ IoT ของตัวเองได้** ระบบพัดลมอัตโนมัติที่ใช้งานได้

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- โค้ด MicroPython ครบทุก Lab และ Workshop พร้อมแผนผังการต่อวงจร
- ไฟล์ Flow ของ Node-RED Dashboard สำหรับนำไปปรับใช้ต่อ
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและนักพัฒนาที่สร้างโปรเจกต์ IoT ด้วย ESP32 และ MicroPython ในงานจริง จุดเด่นคือการสอนแบบลงมือต่อวงจรไปพร้อมกันทีละขั้น อธิบายทั้งฝั่งฮาร์ดแวร์และโค้ดให้เห็นว่าเชื่อมกันอย่างไร และพาไล่แก้ปัญหาที่ผู้เริ่มต้นเจอบ่อย เพื่อให้ผู้เรียนกลับไปสร้างระบบ IoT ของตนเองได้อย่างมั่นใจ

ราคาค่าอบรม 5,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

5,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449 · 088-807-9770

@itgenius · contact@itgenius.co.th

เพิ่มเพื่อน LINE