**ใบงานที่ 1 การใช้งานพอร์ด GPIO แบบดิจิทัล (ไฟกระพริบ)**

**กล่าวนำ**

ใบงานนี้เป็นใบงานแรกที่นักศึกษาจะได้ฝึกใช้งานบอร์ดทดลองและโปรแกรม ดังนั้นเพื่อเป็นการประหยัดเวลาในการเรียนรู้ ใบงานนี้จะอธิบายวิธีการใช้งานและบอร์ดไปพร้อมกับการทดลอง

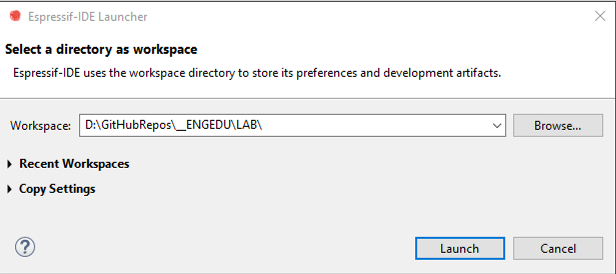
**ขั้นตอนการทดลอง**

1. เปิดโปรแกรม Espressif IDE โดยการดับเบิ้ลคลิกที่ไอคอน โปรแกรมจะแสดง splash screen ดังภาพ



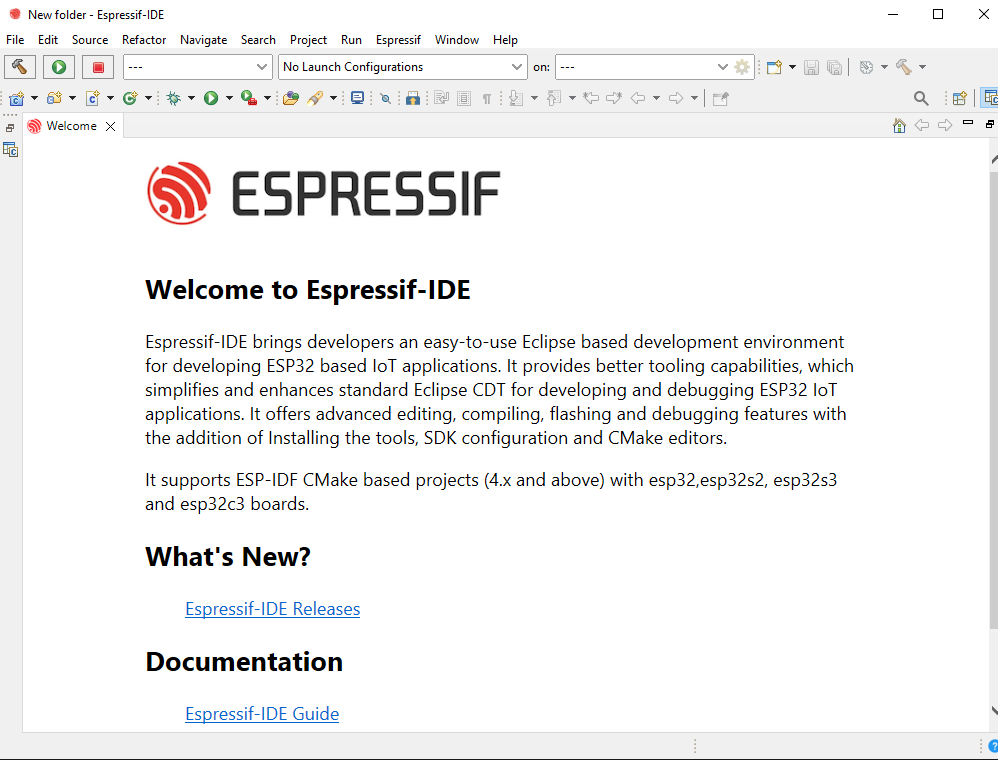
**รูปที่ 2.1** Splash screen ของ ESP-IDF

1. โปรแกรม Espressif IDE จะถามที่ตั้งของ Project (เรียกว่า Workspace) ให่ browse ไปยังที่จะสร้าง workspace อาจจะใช้ folder ที่มีอยู่แล้วหรือสร้างใหม่ก็ได้ ซึ่งต้องไม่มี spacebar ในชื่อของ folder (ถ้ามี spacebar จะไม่สามารถสร้าง project ได้)



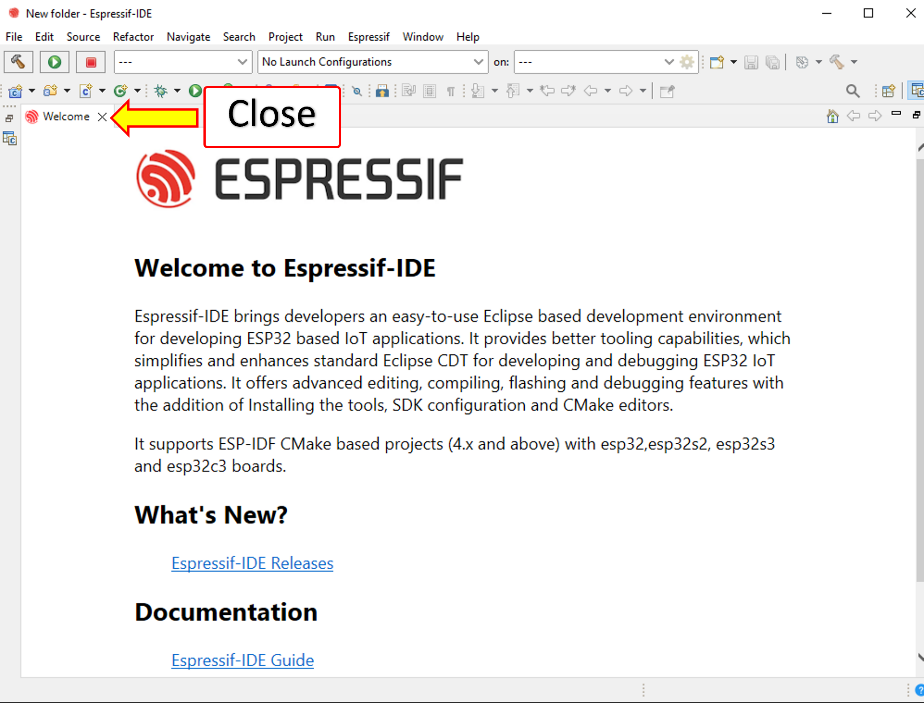
**รูปที่ 2.2** กล่องข้อความโต้ตอบสำหรับเลือก workspace directory

1. โปรแกรม Espressif IDE จะแสดงหน้าจอ Welcome



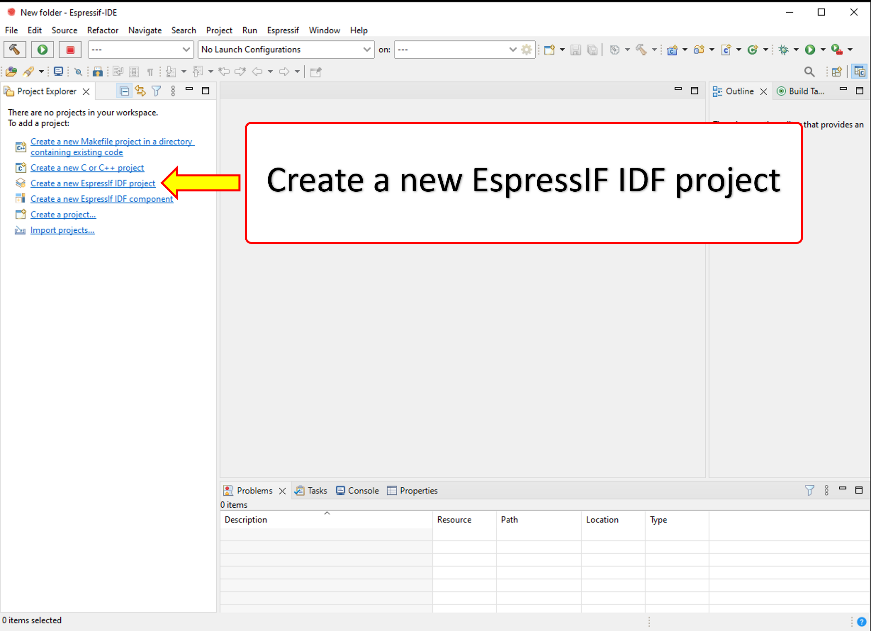
**รูปที่ 2.3** หน้าจอ Welcome to Espressif-IDE

1. ให้ปิดหน้าต่าง Welcome แล้วโปรแกรม Espressif IDE จะแสดง Project Explorer ขึ้นมา



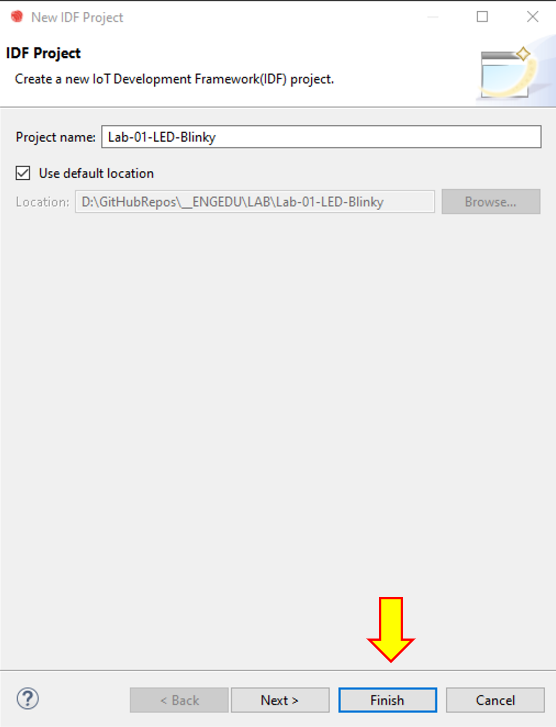
**รูปที่ 2.4** การปิดหน้าจอ Welcome to Espressif-IDE

1. สร้างโปรเจคใหม่ โดยการคลิกที่ Create a new Espressif-IDF project



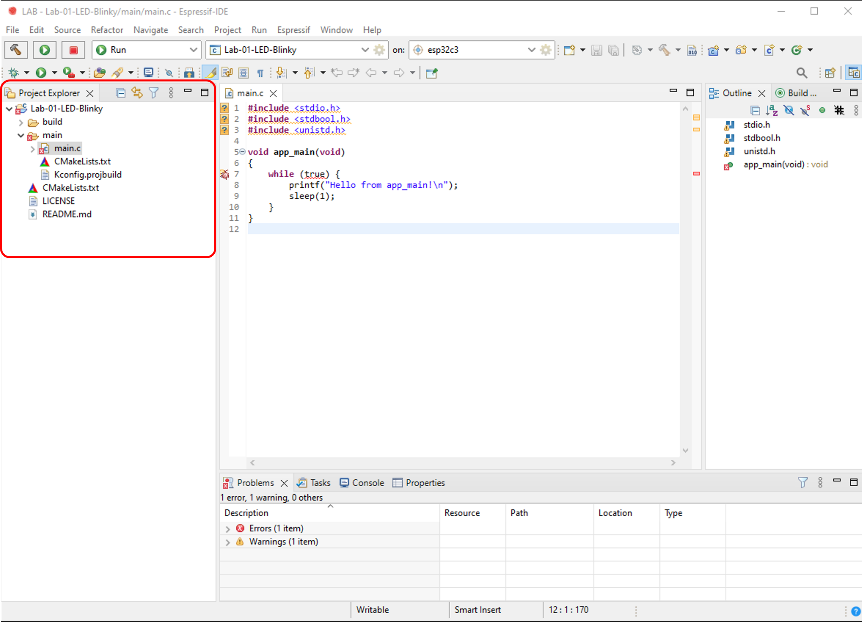
**รูปที่ 2.5** การสร้าง Espressif project

1. สร้างโปรเจค IDF โดยการกำหนด Project name ระวังอย่าให้ชื่อ project มี spacebar ถ้าต้องการแยกคำให้ใช้เครื่องหมายขีด (-) หรือ underscore (\_) เช่น Lab-01-LED-Blinky หรือ Lab-01-LED-Blinky เสร็จแล้วกด Finish (ถ้ากดปุ่ม Next จะสามารถเลือกโปรแกรมจากตัวอย่างได้)



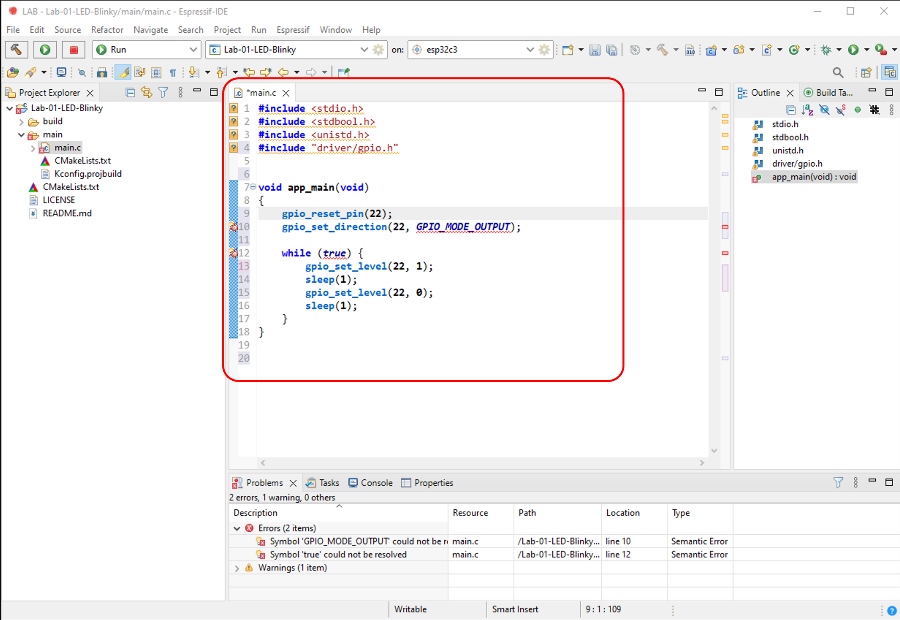
**รูปที่ 2.6** การกำหนดชื่อ project และกด Finish เพื่อเริ่มสร้าง project

1. โปรแกรมพัฒนา (IDE) จะพามาที่หน้าโปรเจค ใน Project Explorer ให้คลิกที่เครื่องหมาย > ที่หน้าชื่อโปรเจคและโฟลเดอร์ย่อยในนั้น จนกระทั่งเจอไฟล์ main.c ให้ดับเบิ้ลคลิกเพื่อเปิดไฟล์ main.c จะเห็นโปรแกรมเบื้องต้นที่ IDE สร้างให้มา



**รูปที่ 2.7**

1. แก้ไขไฟล์ main.c ให่มีหน้าตาดังภาพด้านล่างนี้



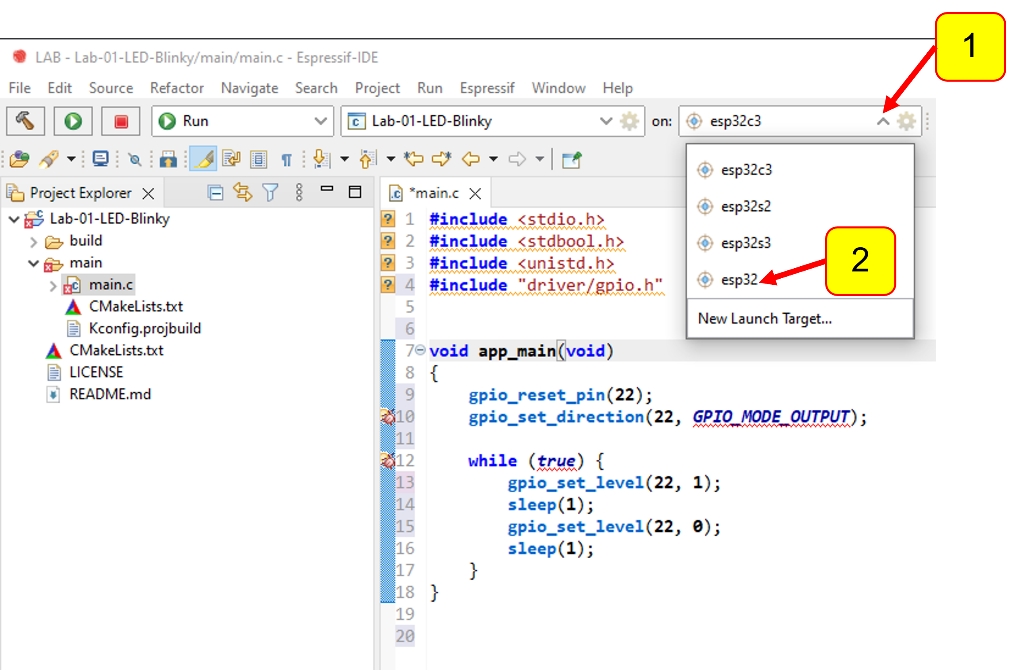
**รูปที่ 2.8**

|  |
| --- |
| Source code ของไฟล์ main.c ในตัวอย่างโปรแกรมไฟกระพริบ |
| #include <stdio.h>  #include <stdbool.h>  #include <unistd.h>  #include "driver/gpio.h" // เพื่อการใช้งาน digital output (GPIO)  void app\_main(void)  {  gpio\_reset\_pin(22); // รีเซ็ตสถานะของขา 22  gpio\_set\_direction(22, GPIO\_MODE\_OUTPUT);// กำหนดให้ขา 22 เป็น digital output  while (true) // while (true) = วนรอบไม่มีที่สิ้นสุด  {  gpio\_set\_level(22, 1); // สั่งให้ LED ติด  sleep(1); // หน่วงเวลา 1 วินาที  gpio\_set\_level(22, 0); // สั่งให้ LED ดับ  sleep(1); // หน่วงเวลา 1 วินาที  }  } |

1. กำหนดบอร์ดพัฒนาที่ใช้ (ขั้นตอนนี้้จะทำเพียงครั้งแรก ที่เริ่ม workspace ใหม่หรือเปลี่ยนบอร์ดที่ใช้ในการทดลอง)

(1) คลิกที่ dropdown ของ combobox on:

(2) เลือกบอร์ดเป้าหมาย (ในชุดทดลองนี้เป็น esp32)

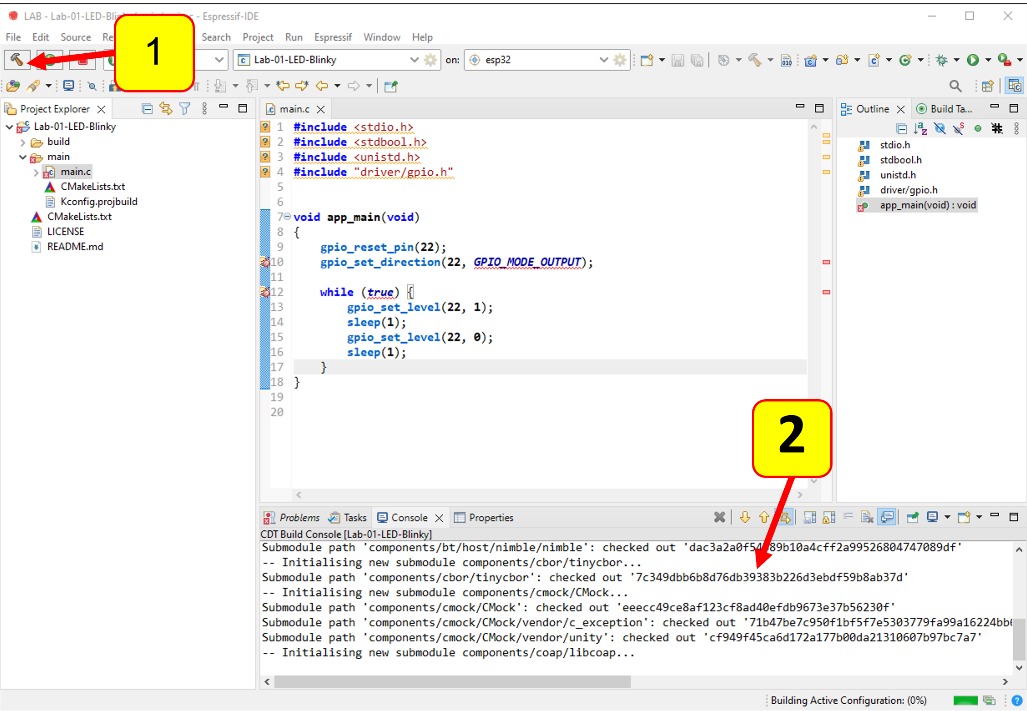


**รูปที่ 2.8**

1. Build โปรเจค ถ้ามีข้อผิดพลาดให้แก้ไขให้เรียบร้อย

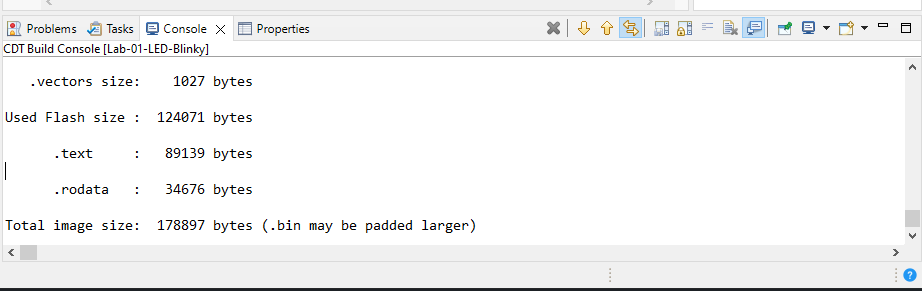
(1) คลิกที่ปุ่ม Build (ดังตำแหน่งในรูปด้านล่าง)

(2) สังเกตุรายงานจาก IDE



**รูปที่ 2.8**

1. หากไม่มีสิ่งใดผิดพลาด (ทั้งการติดตั้งและ source code) โปรแกรม IDE จะสร้างโปรเจกต์จนเสร็จเรียบร้อยและรายงานขนาดสุดท้ายของ binary ที่จะเขียนลงใน microcontroller

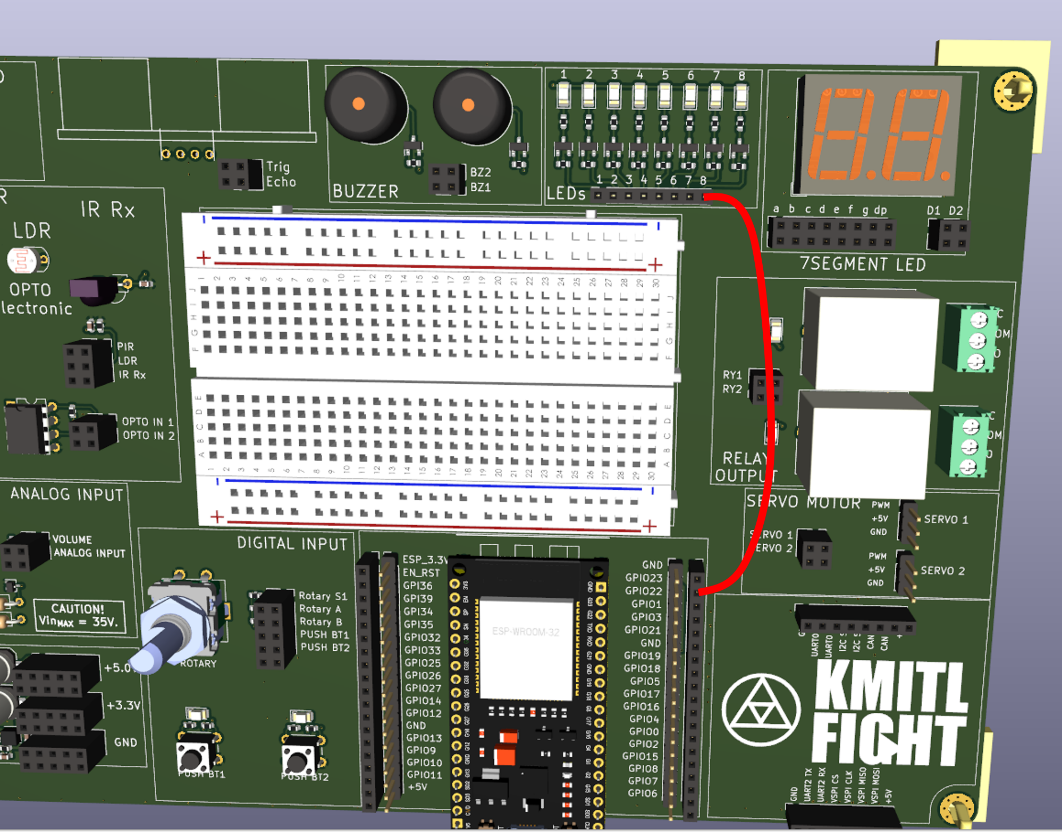


**รูปที่ 2.8**

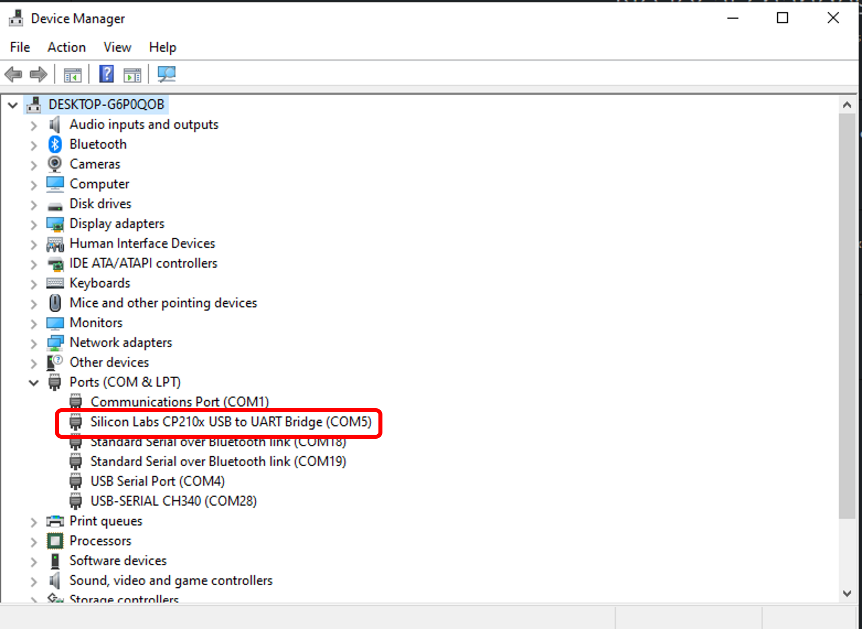
|  |
| --- |
| **ตัวอย่างรายงาน** |
| Used static DRAM: 11352 bytes ( 169384 remain, 6.3% used)  .data size: 9064 bytes  .bss size: 2288 bytes  Used static IRAM: 45762 bytes ( 85310 remain, 34.9% used)  .text size: 44735 bytes  .vectors size: 1027 bytes  Used Flash size : 124071 bytes  .text : 89139 bytes  .rodata : 34676 bytes  Total image size: 178897 bytes (.bin may be padded larger)  C:\Espressif\python\_env\idf4.4\_py3.8\_env\Scripts\python.exe C:/Espressif/frameworks/esp-idf-v4.4.2/\components\partition\_table\gen\_esp32part.py D:\GitHubRepos\\_\_ENGEDU\LAB\Lab-01-LED-Blinky\build\partition\_table\partition-table.bin  Total time taken to build the project: 163,765 ms |

## **การ Upload Firmware สู่บอร์ดทดลอง**

1. จากตัวอย่างโปรแกรม จะเห็นว่าเราใช้ GPIO หมายเลข 22 ดังนั้น ที่บนบอร์ด ให้ต่อสาย jump จากขาที่เขียน GPIO22 ไปยัง LED ดวงหนึ่งที่อยู่บนบอร์ด (สมมติเป็น LED หมายเลข 8)



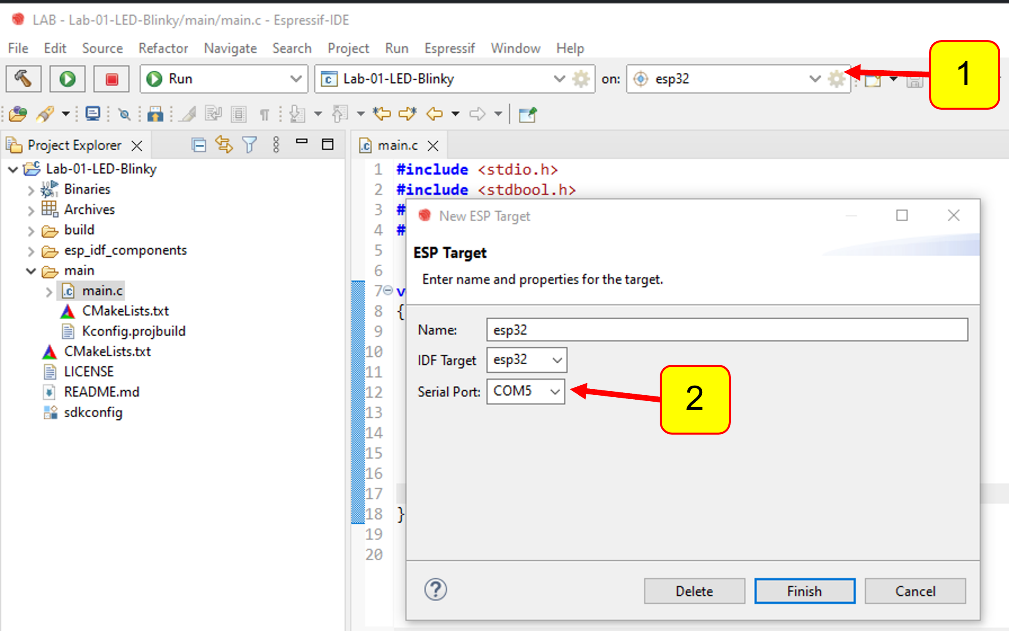
1. ให้ต่อบอร์ดทดลองเข้ากับ USB ของคอมพิวเตอร์
2. ให้เข้าไปที่ device manager เพื่อดูว่าหมายเลขของ communication port ที่คอมพิวเตอร์จัดให้บอร์ดทดลองของเราเป็นเลขอะไร ซึ่งบอร์ด ESP32 ที่ใช้ในการทดลองนี้ใช้ชิป USB to Serial รุ่น Silicon Labs CP210x USB to UART Bridge ให้จำชื่อ Communication port ไว้ ซึ่งในตัวอย่างนี้คือ **COM5**



1. กลับมาที่โปรแกรม ESP32 IDE ให้เลือก Comm port โดยคลิกที่รูปเฟืองใน combobox on:

(1) คลิกที่รูปเฟืองใน combobox on:

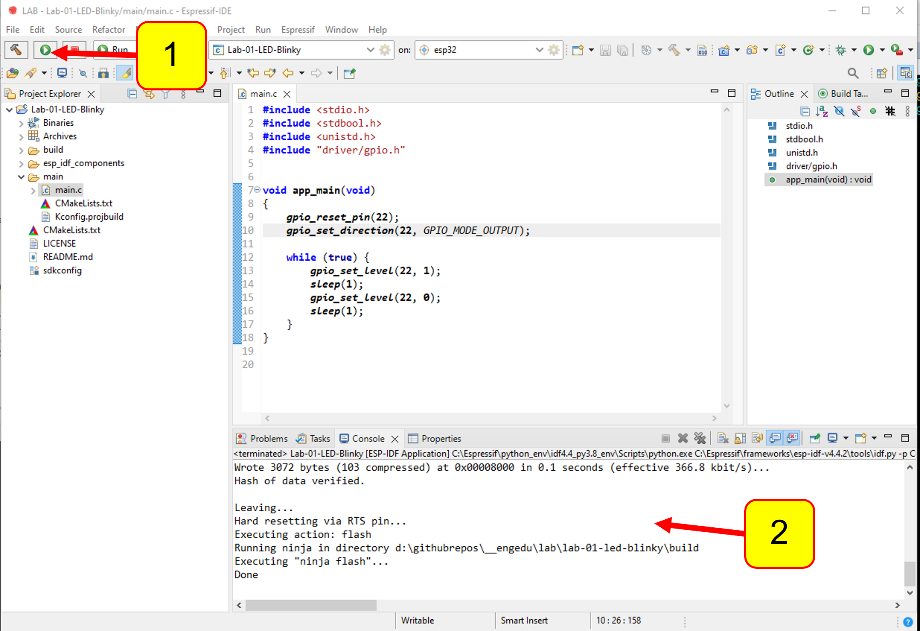
(2) เลือก Communication port ที่ต่อกับบอร์ดทดลอง แล้วกด Finish



1. Upload firmware เข้าสู่บอร์ดทดลอง

(1) คลิกปุ่ม Launch in 'RUN' mode

(2) สังเกตุการรายงานผลจาก IDE



ถ้าโปรแกรมได้สำเร็จ จะมีข้อความดังตัวอย่างต่อไปนี้

Connecting.....

Chip is ESP32-D0WD-V3 (revision 3)

Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None

Crystal is 40MHz

MAC: 94:b5:55:f3:98:00

Uploading stub...

Running stub...

Stub running...

Changing baud rate to 460800

Changed.

Configuring flash size...

Flash will be erased from 0x00001000 to 0x00007fff...

Flash will be erased from 0x00010000 to 0x0003bfff...

Flash will be erased from 0x00008000 to 0x00008fff...

Compressed 25392 bytes to 15888...

Writing at 0x00001000... (100 %)

Wrote 25392 bytes (15888 compressed) at 0x00001000 in 0.8 seconds (effective 264.9 kbit/s)...

Hash of data verified.

Compressed 179024 bytes to 93086...

Writing at 0x00010000... (16 %)

Writing at 0x0001bc95... (33 %)

Writing at 0x000214bd... (50 %)

Writing at 0x0002717c... (66 %)

Writing at 0x0002f7b2... (83 %)

Writing at 0x0003794b... (100 %)

Wrote 179024 bytes (93086 compressed) at 0x00010000 in 2.6 seconds (effective 558.1 kbit/s)...

Hash of data verified.

Compressed 3072 bytes to 103...

Writing at 0x00008000... (100 %)

Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 366.8 kbit/s)...

Hash of data verified.

Leaving...

Hard resetting via RTS pin...

Executing action: flash

Running ninja in directory d:\githubrepos\\_\_engedu\lab\lab-01-led-blinky\build

Executing "ninja flash"...

Done

1. สังเกตบนบอร์ด จะมีไฟกระพริบเป็นจังหวะติด-ดับสลับกันทุก 1 วินาที

* ถ้ามีสิ่งผิดปกติ เช่นไม่สามารถ upload ได้ให้ตรวจสอบการเชื่อมต่อและหลายเลข Communication port ให้ถูกต้อง
* ถ้ามีข้อความ Connecting............................. และรายงานว่าไม่สามารถเชื่อมต่อได้ ให้ทำการกดปุ่ม BOOT ที่บนบอร์ดไว้จนกว่าโปรแกรมได้
* สามารถสังเกตว่าบอร์ดสามารถโปรแกรมได้ จากบรรทัดต่อไปนี้

Writing at 0x00010000... (16 %)

Writing at 0x0001bc95... (33 %)

Writing ...