

## การติดตั้ง esp32 tools เพื่อใช้งานกับโปรแกรม Arduino IDE

ในกรณีที่ยังไม่ได้ติดตั้งใช้งานโปรแกรม Arduino IDE ไว้เลย ผู้ใช้จะต้องเริ่มต้นทำการติดตั้งโปรแกรมใหม่ทั้งหมด ซึ่งโปรแกรม Arduino IDE สามารถทำการติดตั้งใช้งานได้กับระบบปฏิบัติการที่หลากหลาย ซึ่งวิธีการอาจมีความแตกต่างกันแต่ในที่นี้จะขอแนะนำขั้นตอนการติดตั้งกับระบบปฏิบัติการของ Windows เท่านั้น สำหรับระบบปฏิบัติการอื่นผู้ใช้สามารถเข้าไปศึกษาขั้นตอนวิธีการได้ในหัวข้อ Installation Instructions ซึ่งจะมี Link ของหัวข้อสำหรับวิธีการติดตั้งในแต่ละระบบปฏิบัติการให้เลือกในที่นี่ให้ไปที่หัวข้อ Development Status -> Installation Instructions ให้เลือก Instructions for Windows

ซึ่งวิธีการและขั้นตอนการติดตั้งโปรแกรม esp32 tools เพื่อใช้งานกับ Arduino IDE แบบต่างๆ ในหลายๆระบบปฏิบัติการ รวมทั้งของ Windows จะอยู่ที่

<https://github.com/espressif/arduino-esp32>

ซึ่งผู้ใช้สามารถอ้างอิงได้จากหัวข้อ Development Status -> Installation Instructions ดังรูป

### Development Status

Most of the framework is implemented. Most noticeable is the missing analogWrite. While analogWrite is on it's way, there are a few other options that you can use:

- 16 channels LEDC which is PWM
- 8 channels SigmaDelta which uses SigmaDelta modulation
- 2 channels DAC which gives real analog output

### Installation Instructions

- Using Arduino IDE
  - [Instructions for Windows](#)
  - [Instructions for Mac](#)
  - [Instructions for Debian/Ubuntu Linux](#)
  - [Instructions for Fedora](#)
  - [Instructions for openSUSE](#)
- Using PlatformIO
- Building with make
- Using as ESP-IDF component

#### Decoding exceptions

You can use [EspExceptionDecoder](#) to get meaningful call trace.

ในกรณีที่ยังไม่เคยติดตั้งโปรแกรม Arduino IDE ไว้ก่อนเลย ผู้ใช้ต้องทำการ download และติดตั้งโปรแกรม Arduino IDE จาก url ชื่อ <https://www.arduino.cc> ให้เรียบร้อย ซึ่งผู้ใช้สามารถสั่ง Run โปรแกรม Install และ ให้โปรแกรม Install ของ Arduino IDE ทำการติดตั้ง tools ต่างๆให้เองโดยอัตโนมัติตามขั้นตอนมาตรฐานของการ Install โปรแกรม

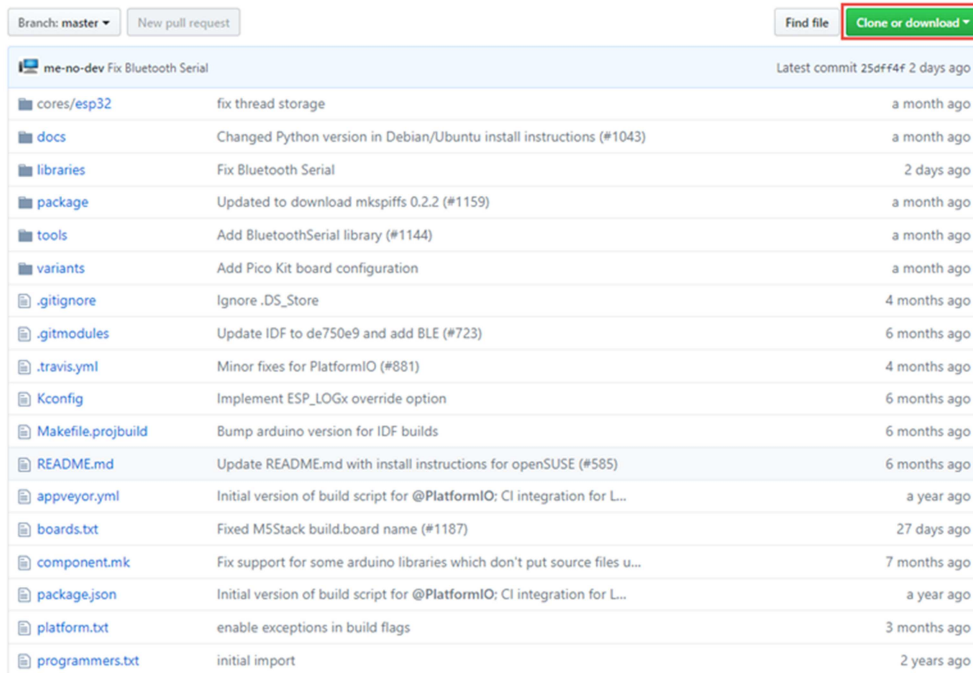
## Download the Arduino IDE



หลังจากทำการดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE เสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการติดตั้ง esp32 tools ลงใน Arduino IDE เพื่อใช้งานร่วมกัน ซึ่งในขั้นตอนนี้ ผู้พัฒนา esp32 tools สำหรับใช้งานกับ Arduino IDE ยังไม่ได้จัดทำไฟล์สำหรับ Install ให้ทำการติดตั้งโปรแกรมให้ เหมือนกรณีของ Arduino IDE ซึ่งในการติดตั้ง esp32 tools นั้นผู้ใช้จะต้องทำการคัดลอก esp32 tools แบบ Clone ทั้งไฟล์และโฟลเดอร์ที่ผู้พัฒนาจัดทำไว้มาติดตั้งในเครื่องคอมพิวเตอร์ของผู้ใช้เอง ซึ่งสามารถทำได้ 2 แบบ โดยแบบที่ 1 จะใช้การดาวน์โหลด esp32 tools แบบ Zip ไฟล์มาพักไว้ที่เครื่องคอมพิวเตอร์ก่อนแล้วจึงสั่ง copy ไฟล์และโฟลเดอร์ไปติดตั้งไว้ภายใต้โปรแกรม Arduino IDE ทั้งหมดเอง ส่วนในแบบที่ 2 จะใช้โปรแกรม Git ในการสั่ง Clone ไฟล์และโฟลเดอร์ต่างๆของ esp32 tools มาติดตั้งในเครื่องคอมพิวเตอร์เองแบบออนไลน์

1. การติดตั้ง esp32 tools โดยการดาวน์โหลดและ copy ติดตั้งเอง โดยมีขั้นตอนดังนี้

1.1 ไปที่ URL ชื่อ <https://github.com/espressif/arduino-esp32> แล้วเลือก Clone or download



หลังจากทำการดาวน์โหลดเสร็จจะได้ไฟล์แบบ Zip File ชื่อ “arduino-esp32-master.zip” ซึ่งจะประกอบไปด้วยไฟล์และโฟลเดอร์ต่างๆดังรูป

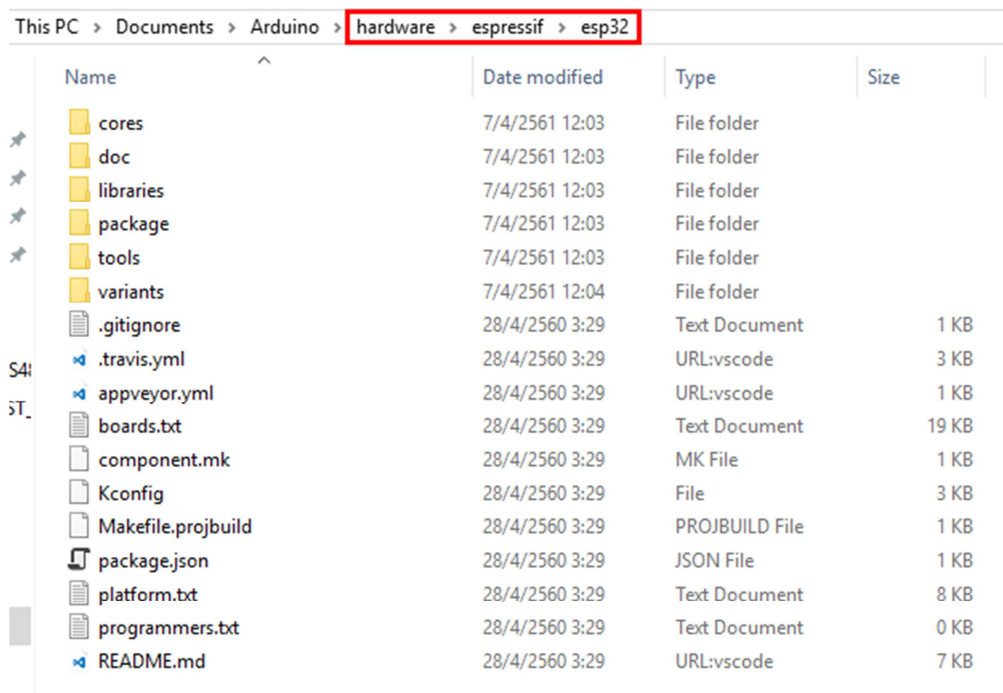
Name	Date modified	Type	Size
cores	3/5/2560 10:17	File folder	
doc	3/5/2560 10:17	File folder	
libraries	3/5/2560 10:17	File folder	
package	3/5/2560 10:17	File folder	
tools	3/5/2560 10:17	File folder	
variants	3/5/2560 10:17	File folder	
.gitignore	28/4/2560 3:29	Text Document	1 KB
.travis.yml	28/4/2560 3:29	URL:vscode	3 KB
appveyor.yml	28/4/2560 3:29	URL:vscode	1 KB
boards.txt	28/4/2560 3:29	Text Document	19 KB
component.mk	28/4/2560 3:29	MK File	1 KB
Kconfig	28/4/2560 3:29	File	3 KB
Makefile.projbuild	28/4/2560 3:29	PROJBUILD File	1 KB
package.json	28/4/2560 3:29	JSON File	1 KB
platform.txt	28/4/2560 3:29	Text Document	8 KB
programmers.txt	28/4/2560 3:29	Text Document	0 KB
README.md	28/4/2560 3:29	URL:vscode	7 KB

1.2 ในการทำงานของโปรแกรม esp32 tools ภายใต้ระบบปฏิบัติการของ Windows โปรแกรมจะถูกกำหนดให้ค้นหาและเรียกใช้ tools ต่างๆจากโฟลเดอร์ที่มีโครงสร้างดังนี้ คือ

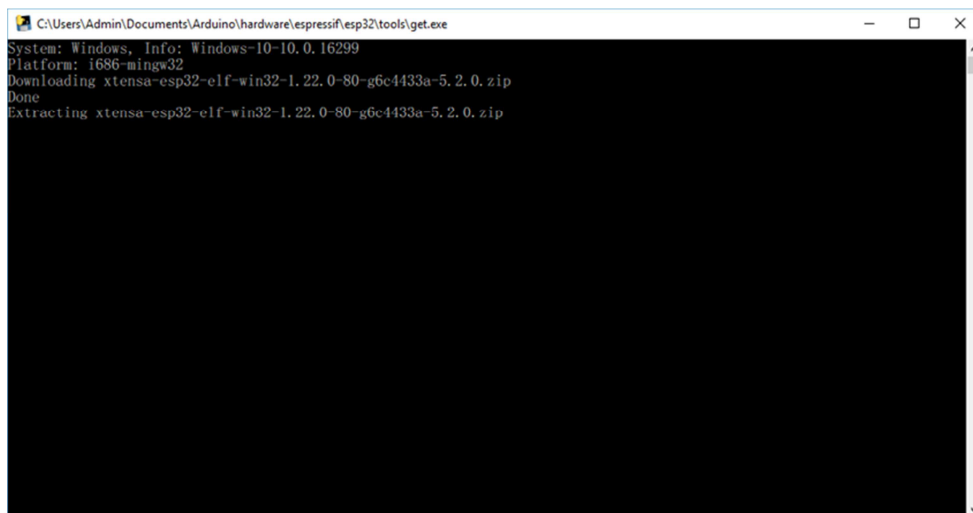
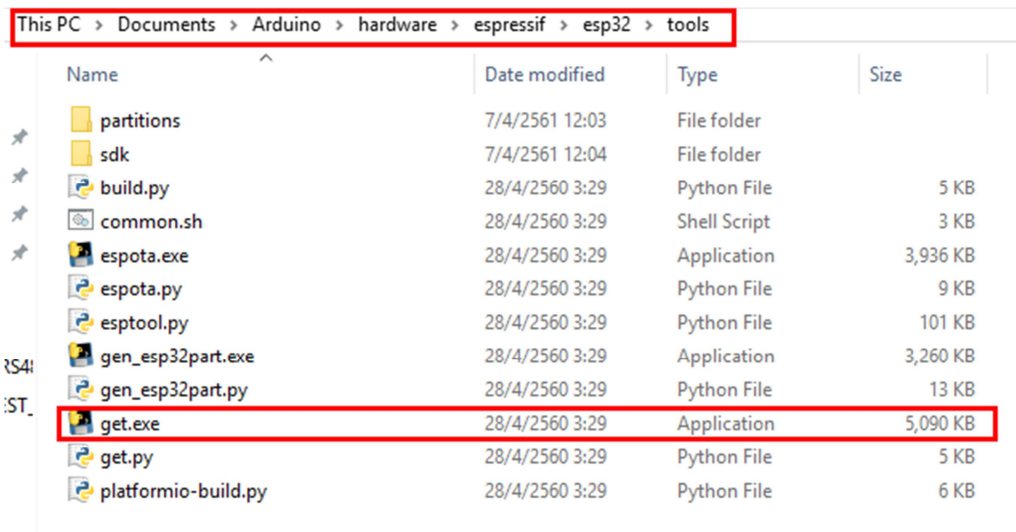
“C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32”

โดยชื่อ [NamePC] ของแต่ละเครื่องจะมีความแตกต่างกันขึ้นอยู่กับการกำหนดในขั้นตอนของการติดตั้ง Windows ซึ่งส่วนนี้ผู้ใช้ต้องปรับแก้ให้ตรงตามความเป็นจริงของเครื่องคอมพิวเตอร์ที่ใช้อยู่ ในขั้นตอนนี้ให้ดูว่ามีโฟลเดอร์ C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32 อยู่ครบถ้วนแล้วหรือยัง ซึ่งในกรณีที่ยังไม่มี folder ย่อยของ espressif อยู่ภายใต้ระบบไฟล์และโฟลเดอร์ เช่น อาจมีเพียง C:\Users\[NamePC]\Documents\Arduino\hardware ก็ให้ทำการสร้าง folder เพิ่มขึ้นมาเป็นลำดับขั้นให้ครบองค์ประกอบตามที่กำหนด คือ “\espressif\esp32” ขึ้นมาเตรียมไว้ให้ครบตามองค์ประกอบที่กำหนดไว้ คือ “C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32”

1.3 ถ้าทำการ Extract ไฟล์ arduino-esp32-master เรียบร้อยแล้วให้ทำการ Copy ข้อมูลทั้งหมดใน folder ไปไว้ใน “C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32” ดังรูป



1.4 ภายใต้โฟลเดอร์ “C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32\tools” ให้สั่ง Run โปรแกรม ชื่อ get.exe เพื่อสั่งให้โปรแกรม get ทำการดาวน์โหลด tools ต่างๆของ esp32 มาติดตั้งในเครื่องให้เรียบร้อย ซึ่งในขั้นตอนนี้จะปรากฏ console ของ command line ขึ้นมาแสดงการทำงานของโปรแกรมให้เห็น ให้รอจนการทำงานเสร็จสมบูรณ์หน้าต่าง console ของ command line จะปิดไปเองโดยอัตโนมัติเป็นอันเสร็จสิ้นขั้นตอนการติดตั้ง esp32 tools ให้กับ Arduino IDE

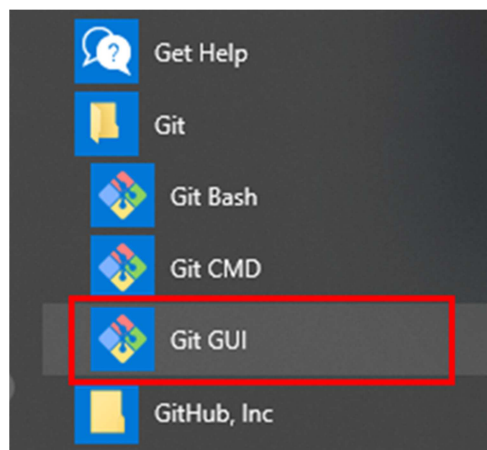


2. ทำการติดตั้ง esp32 tools โดยใช้โปรแกรม Git ทำการ clone ไฟล์และโฟลเดอร์มาติดตั้งที่เครื่อง

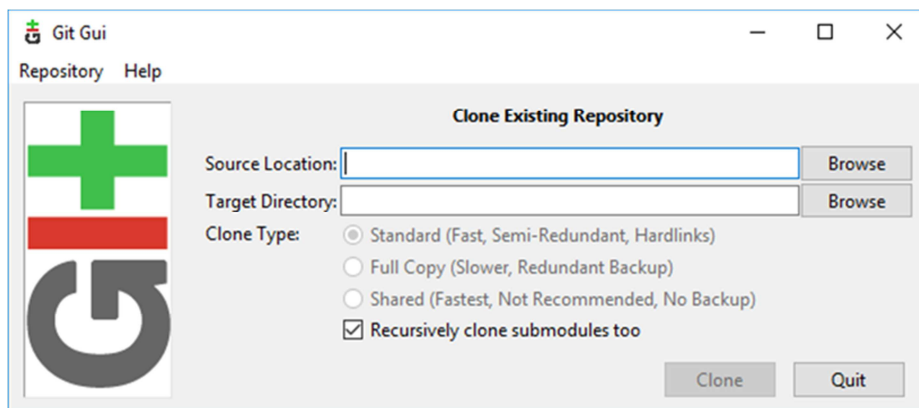
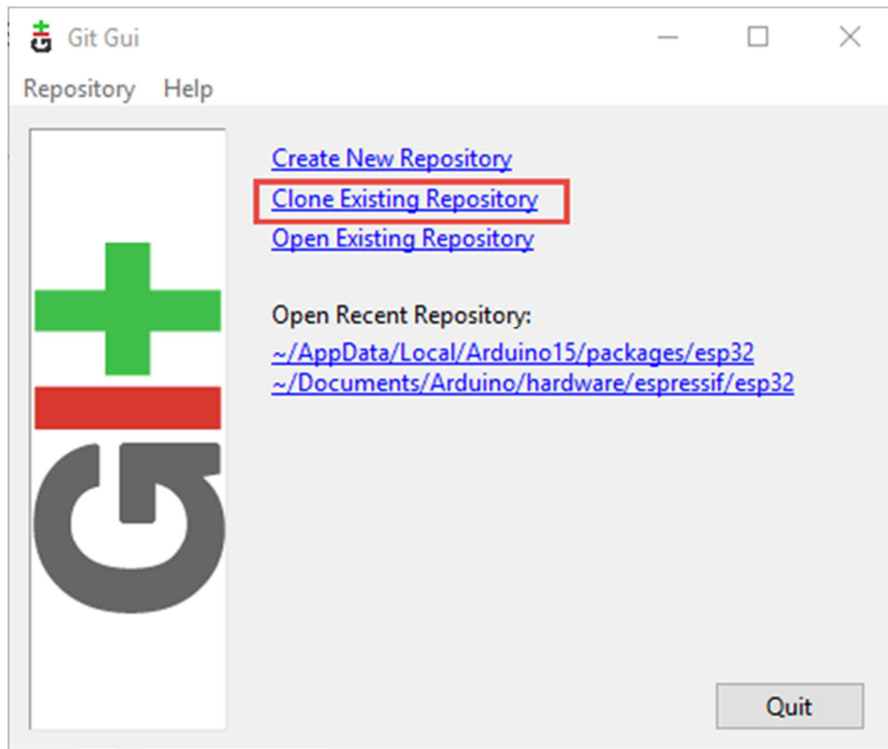
2.1 ทำการ download และติดตั้งโปรแกรม Git จาก <https://git-scm.com/download/win>



2.2 หลังจากทำการติดตั้งโปรแกรม Git เสร็จเรียบร้อยแล้ว สั่ง Run Git GUI จากเมนู



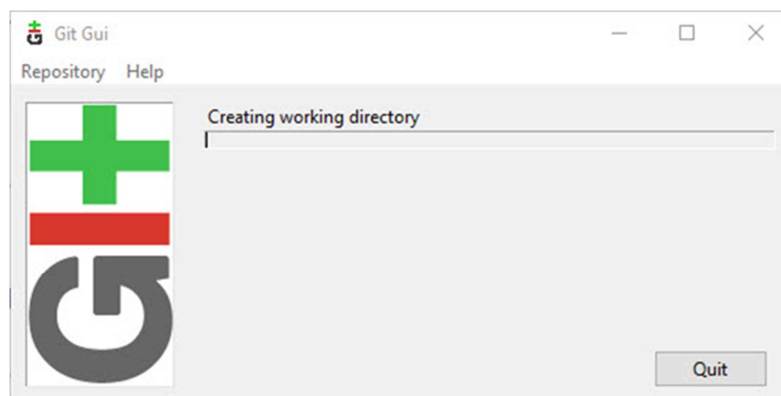
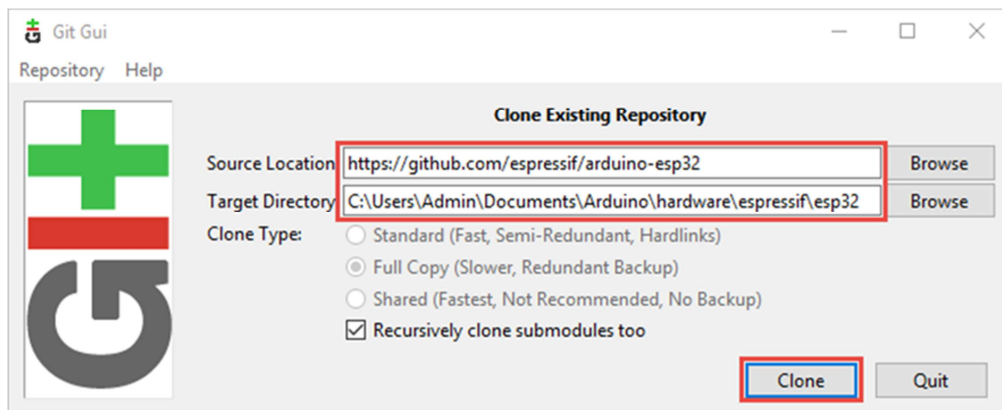
2.3 เลือก Clone Existing Repository



## 2.4 ทำการกำหนด Source Location และ Target Directory ให้กับโปรแกรม Git โดย

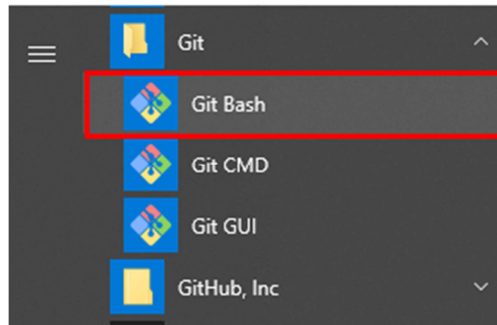
- กำหนด Source Location:
  - <https://github.com/espressif/arduino-esp32.git>
- กำหนด Target Directory:
  - “C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32”

โดยชื่อ [NamePC] ของแต่ละเครื่องจะมีความแตกต่างกันขึ้นอยู่กับที่กำหนดในขั้นตอนของการติดตั้ง Windows ซึ่งส่วนนี้ผู้ใช้ต้องปรับแก้ให้ตรงตามความเป็นจริงของเครื่องคอมพิวเตอร์ที่ใช้อยู่ เช่น “C:\Users\Admin\Documents\Arduino\hardware\espressif\esp32” เมื่อทำการกำหนดค่า Source Location และ Target Directory เรียบร้อยแล้วให้เลือกที่ Clone และรอจนการทำงานเสร็จสิ้นดังรูป



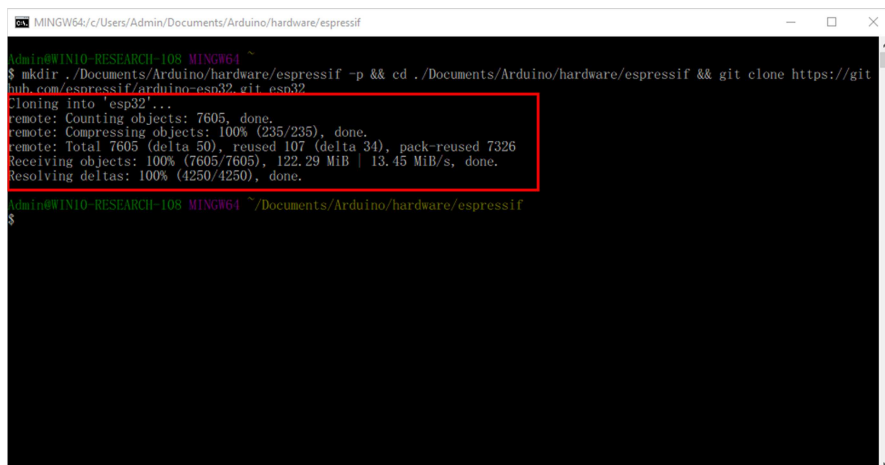
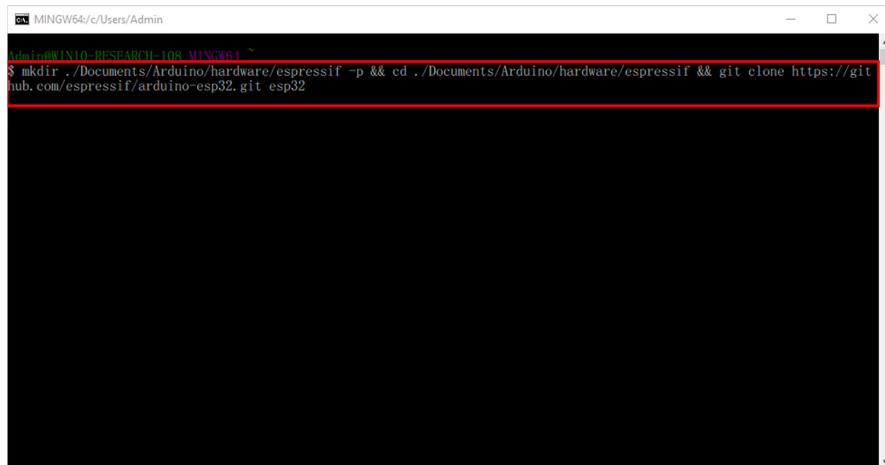


2.5 ให้สั่งดาวน์โหลดโปรแกรม tools ต่างๆของ esp32 มาติดตั้งในเครื่อง โดยใช้โปรแกรม Git Bash

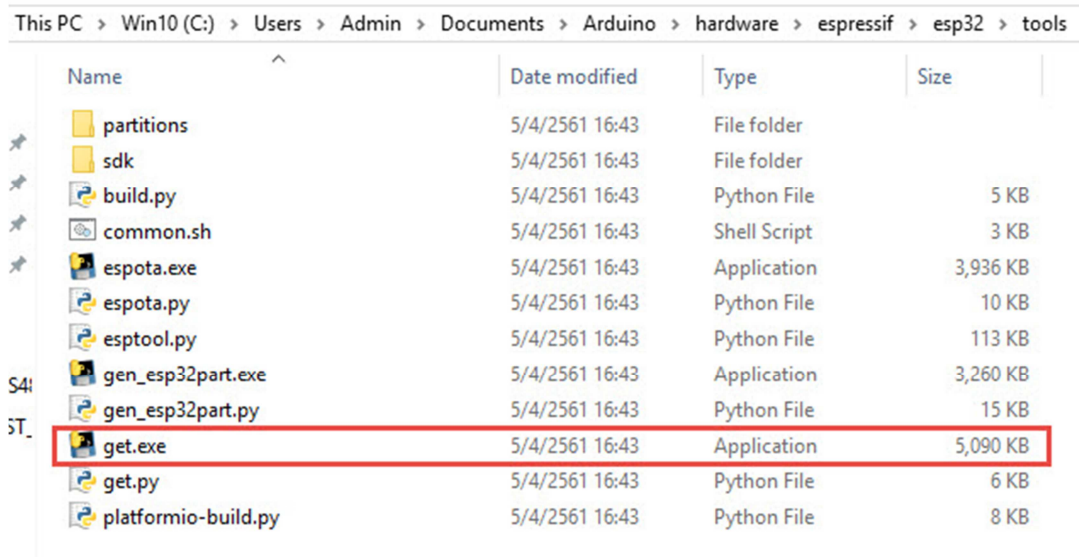


ให้พิมพ์คำสั่งเพื่อสั่งให้โปรแกรม Git ทำการ Clone ชุดโปรแกรมของ esp32 tools มาไว้ในเครื่องคอมพิวเตอร์ของเราภายใต้โปรแกรม Arduino IDE ดังตัวอย่าง

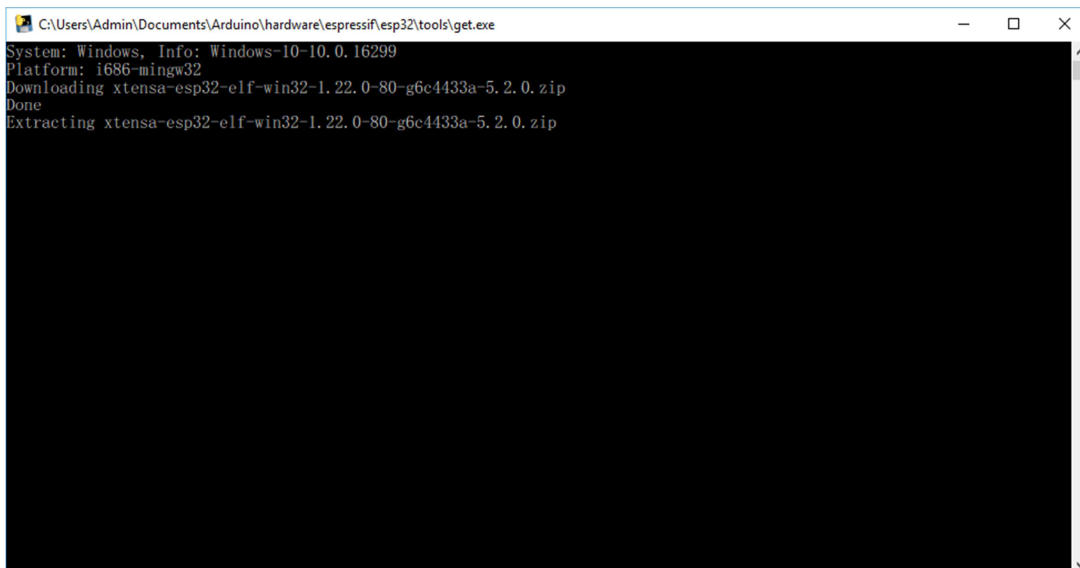
```
"mkdir ./Documents/Arduino/hardware/espessif -p && cd ./Documents/Arduino/hardware/espessif && git clone https://github.com/espessif/arduino-esp32.git esp32"
```



ภายใต้โฟลเดอร์ “C:\Users\[NamePC]\Documents\Arduino\hardware\espressif\esp32\tools” ให้สั่ง Run โปรแกรม ชื่อ get.exe เพื่อสั่งให้โปรแกรม get ทำการดาวน์โหลด tools ต่างๆของ esp32 มาติดตั้งในเครื่องให้เรียบร้อย ซึ่งในขั้นตอนนี้จะปรากฏ console ของ command line ขึ้นมาแสดงการทำงานของโปรแกรมให้เห็น ให้รอจนการทำงานเสร็จสมบูรณ์หน้าต่าง console ของ command line จะปิดไปเองโดยอัตโนมัติเป็นอันเสร็จสิ้นขั้นตอนการติดตั้ง esp32 tools ให้กับ Arduino IDE



Name	Date modified	Type	Size
partitions	5/4/2561 16:43	File folder	
sdk	5/4/2561 16:43	File folder	
build.py	5/4/2561 16:43	Python File	5 KB
common.sh	5/4/2561 16:43	Shell Script	3 KB
espota.exe	5/4/2561 16:43	Application	3,936 KB
espota.py	5/4/2561 16:43	Python File	10 KB
esptool.py	5/4/2561 16:43	Python File	113 KB
gen_esp32part.exe	5/4/2561 16:43	Application	3,260 KB
gen_esp32part.py	5/4/2561 16:43	Python File	15 KB
get.exe	5/4/2561 16:43	Application	5,090 KB
get.py	5/4/2561 16:43	Python File	6 KB
platformio-build.py	5/4/2561 16:43	Python File	8 KB

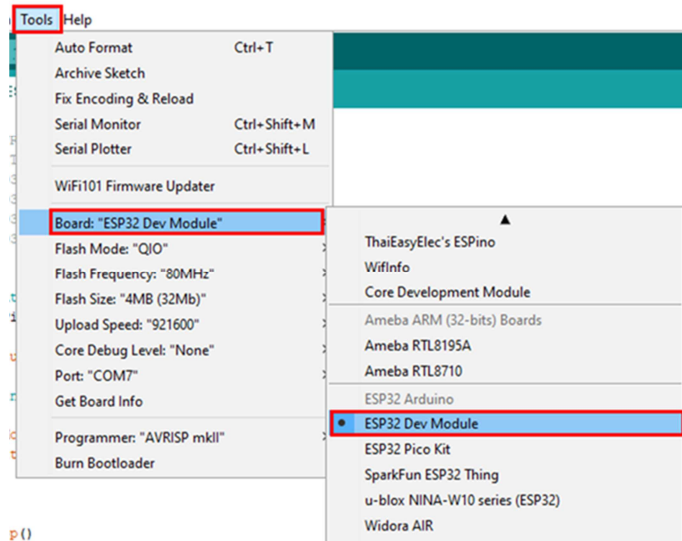


```
C:\Users\Admin\Documents\Arduino\hardware\espressif\esp32\tools\get.exe
System: Windows, Info: Windows-10-10.0.16299
Platform: i686-mingw32
Downloading xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip
Done
Extracting xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip
```

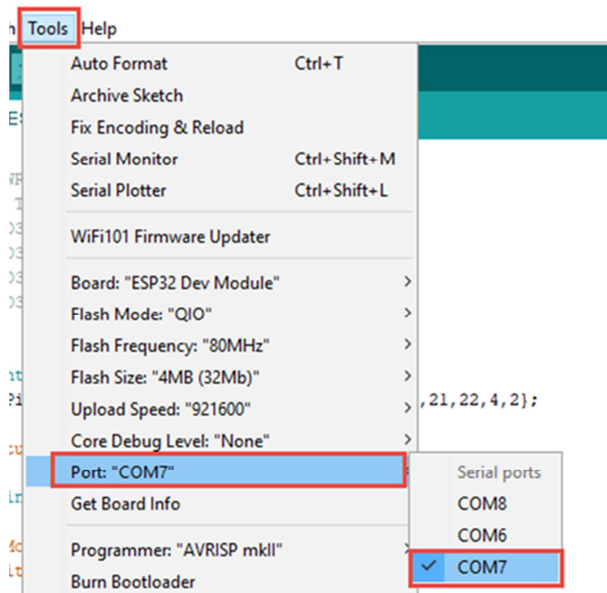
## การใช้งาน esp32 กับ Arduino IDE

1. สั่ง Run Arduino IDE แล้วเลือกที่เมนู Tools -> Board: ซึ่งจะมีเมนูรุ่นของบอร์ดในตระกูล ESP32 ให้เลือกกำหนดในการใช้งาน ในที่นี้ให้เลือก ESP32 Dev Module และที่เมนู Port ให้เลือกหมายเลขพอร์ตเป็น COM Port ของบอร์ดที่ได้จากการจัดสรรของ Windows Driver ดังตัวอย่าง

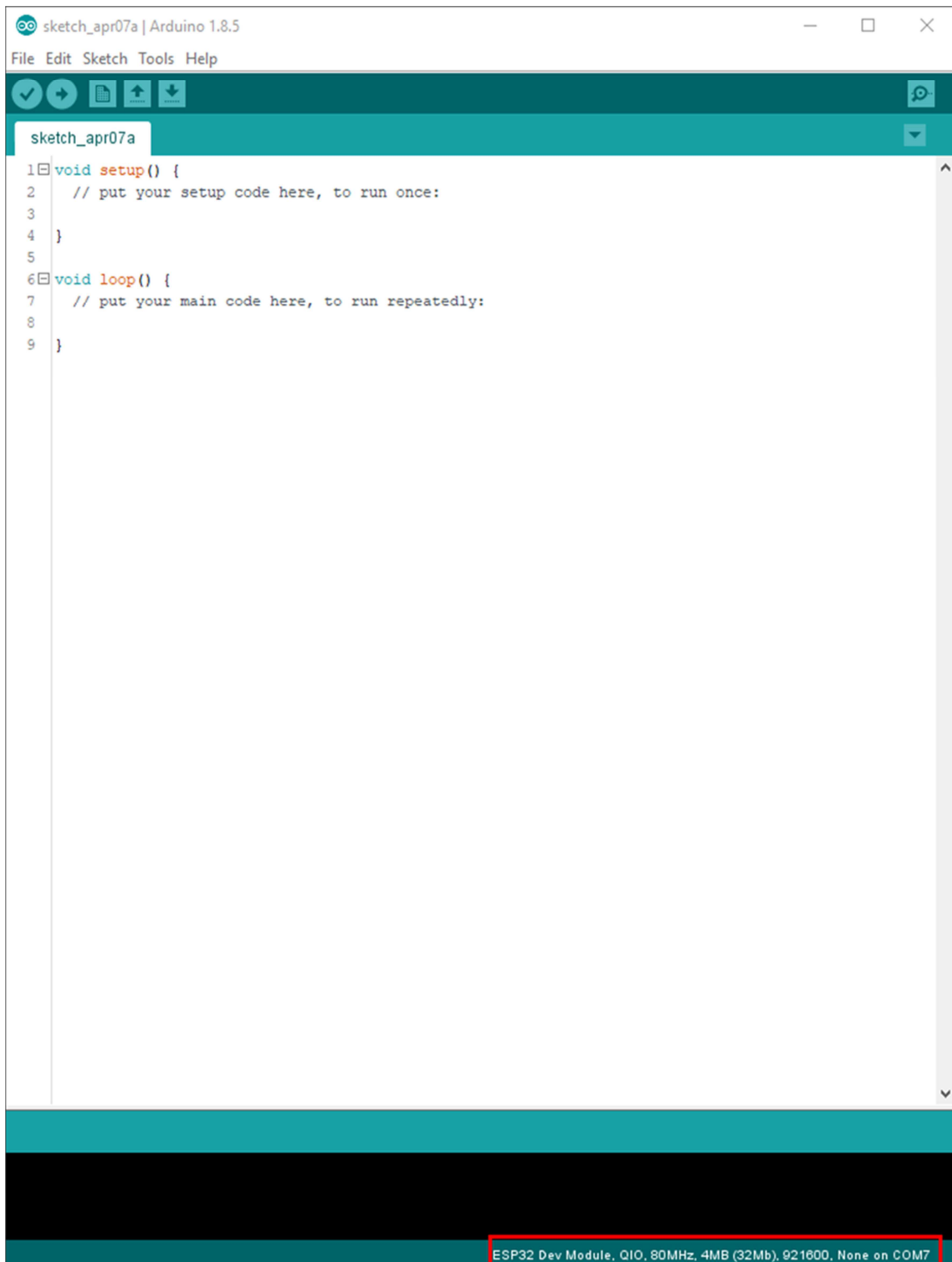
### Tools -> Board: ESP32 Dev Module



### Tools -> Port:หมายเลข COM Port ของบอร์ด



2. ทำการเลือกที่เมนู File New ซึ่งจะได้หน้าต่างกระดาษสำหรับเริ่มต้นเขียนโปรแกรมมาให้ ให้ทำการพิมพ์โค้ดโปรแกรมสำหรับทดสอบการทำงานของบอร์ดแล้วสั่งบันทึกไว้ดังตัวอย่าง



```
sketch_apr07a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_apr07a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

ESP32 Dev Module, QIO, 80MHz, 4MB (32Mb), 921600, None on COM7

```
const int ledCount = 16;

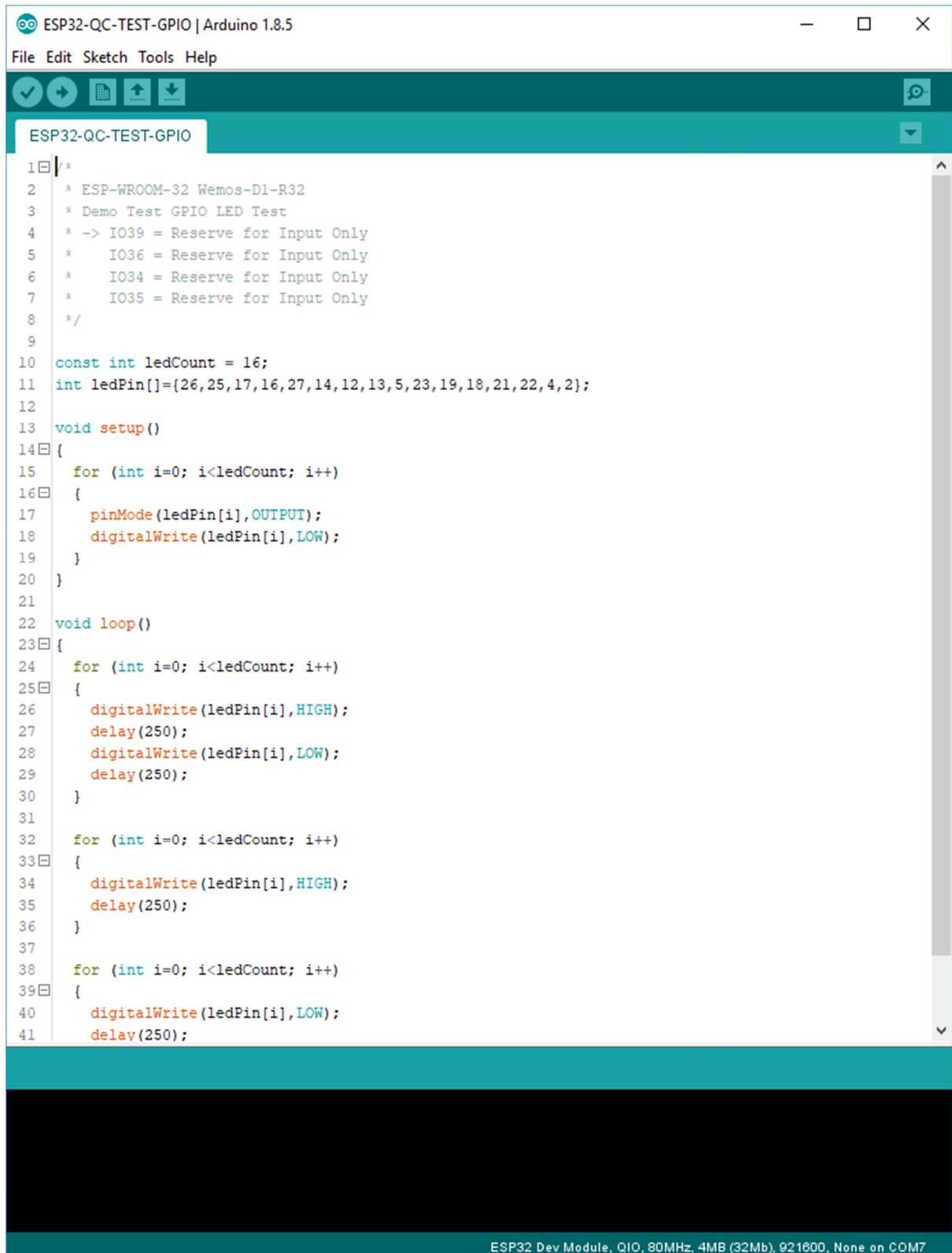
int ledPin[]={26,25,17,16,27,14,12,13,5,23,19,18,21,22,4,2};

void setup()
{
  for (int i=0; i<ledCount; i++)
  {
    pinMode(ledPin[i],OUTPUT);
    digitalWrite(ledPin[i],LOW);
  }
}

void loop()
{
  for (int i=0; i<ledCount; i++)
  {
    digitalWrite(ledPin[i],HIGH);
    delay(250);
    digitalWrite(ledPin[i],LOW);
    delay(250);
  }

  for (int i=0; i<ledCount; i++)
  {
    digitalWrite(ledPin[i],HIGH);
    delay(250);
  }

  for (int i=0; i<ledCount; i++)
  {
    digitalWrite(ledPin[i],LOW);
    delay(250);
  }
}
```



```
ESP32-QC-TEST-GPIO | Arduino 1.8.5
File Edit Sketch Tools Help
ESP32-QC-TEST-GPIO
1 | *
2 | * ESP-WROOM-32 Wemos-D1-R32
3 | * Demo Test GPIO LED Test
4 | * -> IO39 = Reserve for Input Only
5 | *   IO36 = Reserve for Input Only
6 | *   IO34 = Reserve for Input Only
7 | *   IO35 = Reserve for Input Only
8 | */
9 |
10 | const int ledCount = 16;
11 | int ledPin[]={26,25,17,16,27,14,12,13,5,23,19,18,21,22,4,2};
12 |
13 | void setup()
14 | {
15 |   for (int i=0; i<ledCount; i++)
16 |   {
17 |     pinMode(ledPin[i],OUTPUT);
18 |     digitalWrite(ledPin[i],LOW);
19 |   }
20 | }
21 |
22 | void loop()
23 | {
24 |   for (int i=0; i<ledCount; i++)
25 |   {
26 |     digitalWrite(ledPin[i],HIGH);
27 |     delay(250);
28 |     digitalWrite(ledPin[i],LOW);
29 |     delay(250);
30 |   }
31 |
32 |   for (int i=0; i<ledCount; i++)
33 |   {
34 |     digitalWrite(ledPin[i],HIGH);
35 |     delay(250);
36 |   }
37 |
38 |   for (int i=0; i<ledCount; i++)
39 |   {
40 |     digitalWrite(ledPin[i],LOW);
41 |     delay(250);

```

ESP32 Dev Module, Q10, 80MHz, 4MB (32Mb), 921600, None on COM7

### 3. สลับบันทึกและทดสอบ Compile

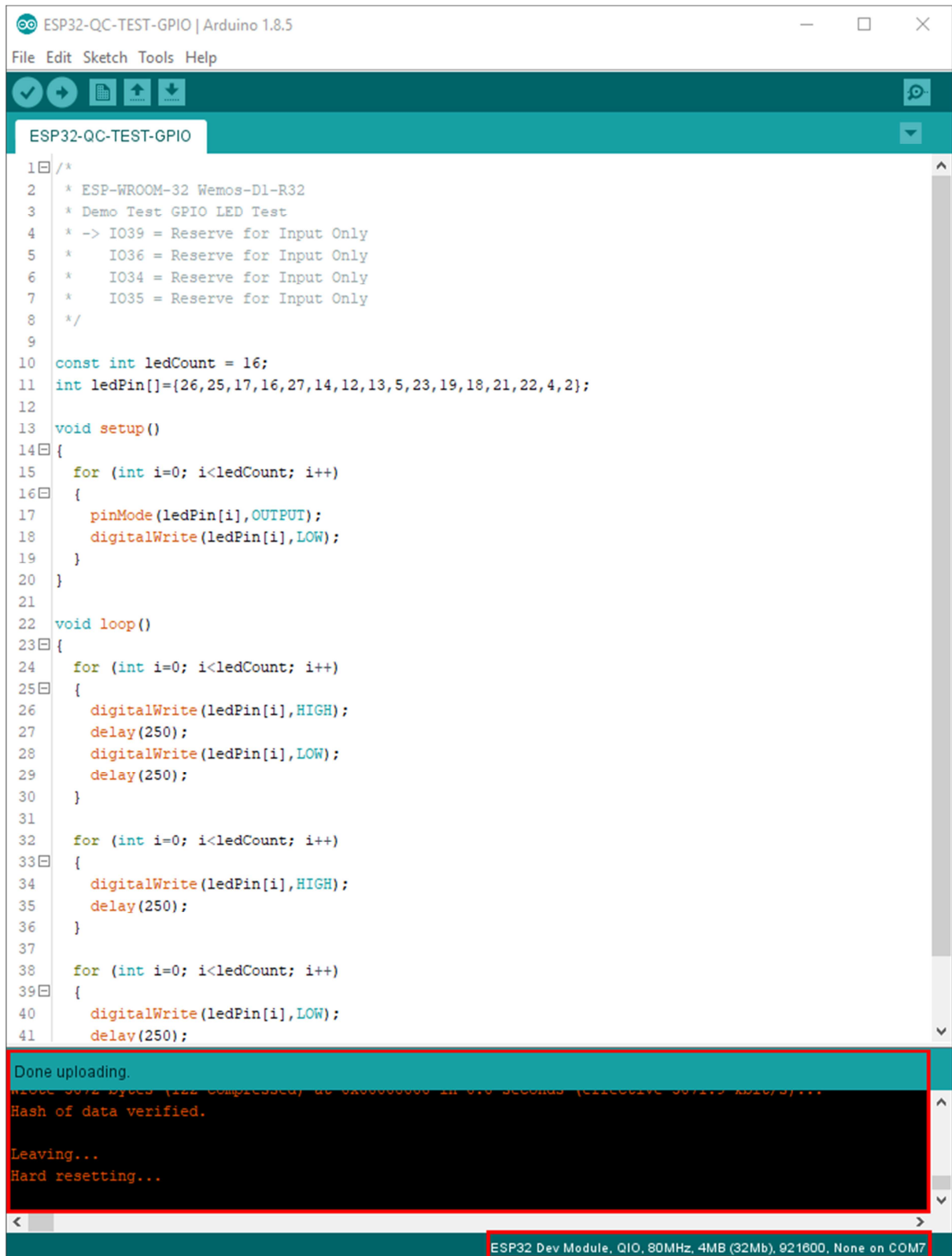
```
1 /*
2  * ESP-WROOM-32 Wemos-D1-R32
3  * Demo Test GPIO LED Test
4  * -> IO39 = Reserve for Input Only
5  *   IO36 = Reserve for Input Only
6  *   IO34 = Reserve for Input Only
7  *   IO35 = Reserve for Input Only
8  */
9
10 const int ledCount = 16;
11 int ledPin[]={26,25,17,16,27,14,12,13,5,23,19,18,21,22,4,2};
12
13 void setup()
14 {
15   for (int i=0; i<ledCount; i++)
16   {
17     pinMode(ledPin[i],OUTPUT);
18     digitalWrite(ledPin[i],LOW);
19   }
20 }
21
22 void loop()
23 {
24   for (int i=0; i<ledCount; i++)
25   {
26     digitalWrite(ledPin[i],HIGH);
27     delay(250);
28     digitalWrite(ledPin[i],LOW);
29     delay(250);
30   }
31
32   for (int i=0; i<ledCount; i++)
33   {
34     digitalWrite(ledPin[i],HIGH);
35     delay(250);
36   }
37
38   for (int i=0; i<ledCount; i++)
39   {
40     digitalWrite(ledPin[i],LOW);
41     delay(250);
42   }
43 }
```

Done compiling.  
http://www.espressos.com/downloads/package\_manifest\_index.json file signature verification failed. file is  
esptool.py v2.1

Sketch uses 159205 bytes (12%) of program storage space. Maximum is 1310720 bytes.  
Global variables use 11032 bytes (3%) of dynamic memory, leaving 283880 bytes for local variables. Maxi

ESP32 Dev Module, Q10, 80MHz, 4MB (32Mb), 921600, None on COM7

4. ทดสอบสั่ง Upload และดูการทำงานของบอร์ด



```
ESP32-QC-TEST-GPIO | Arduino 1.8.5
File Edit Sketch Tools Help
ESP32-QC-TEST-GPIO
1 /*
2  * ESP-WROOM-32 Wemos-D1-R32
3  * Demo Test GPIO LED Test
4  * -> IO39 = Reserve for Input Only
5  *   IO36 = Reserve for Input Only
6  *   IO34 = Reserve for Input Only
7  *   IO35 = Reserve for Input Only
8  */
9
10 const int ledCount = 16;
11 int ledPin[]={26,25,17,16,27,14,12,13,5,23,19,18,21,22,4,2};
12
13 void setup()
14 {
15   for (int i=0; i<ledCount; i++)
16   {
17     pinMode(ledPin[i],OUTPUT);
18     digitalWrite(ledPin[i],LOW);
19   }
20 }
21
22 void loop()
23 {
24   for (int i=0; i<ledCount; i++)
25   {
26     digitalWrite(ledPin[i],HIGH);
27     delay(250);
28     digitalWrite(ledPin[i],LOW);
29     delay(250);
30   }
31
32   for (int i=0; i<ledCount; i++)
33   {
34     digitalWrite(ledPin[i],HIGH);
35     delay(250);
36   }
37
38   for (int i=0; i<ledCount; i++)
39   {
40     digitalWrite(ledPin[i],LOW);
41     delay(250);
42   }
43 }
```

Done uploading.  
11000 3072 Bytes (122 compressed) at 0x00000000 in 0.10 seconds (effective 3071.9 KB/s)...

Hash of data verified.

Leaving...

Hard resetting...

ESP32 Dev Module, Q10, 80MHz, 4MB (32Mb), 921600, None on COM7