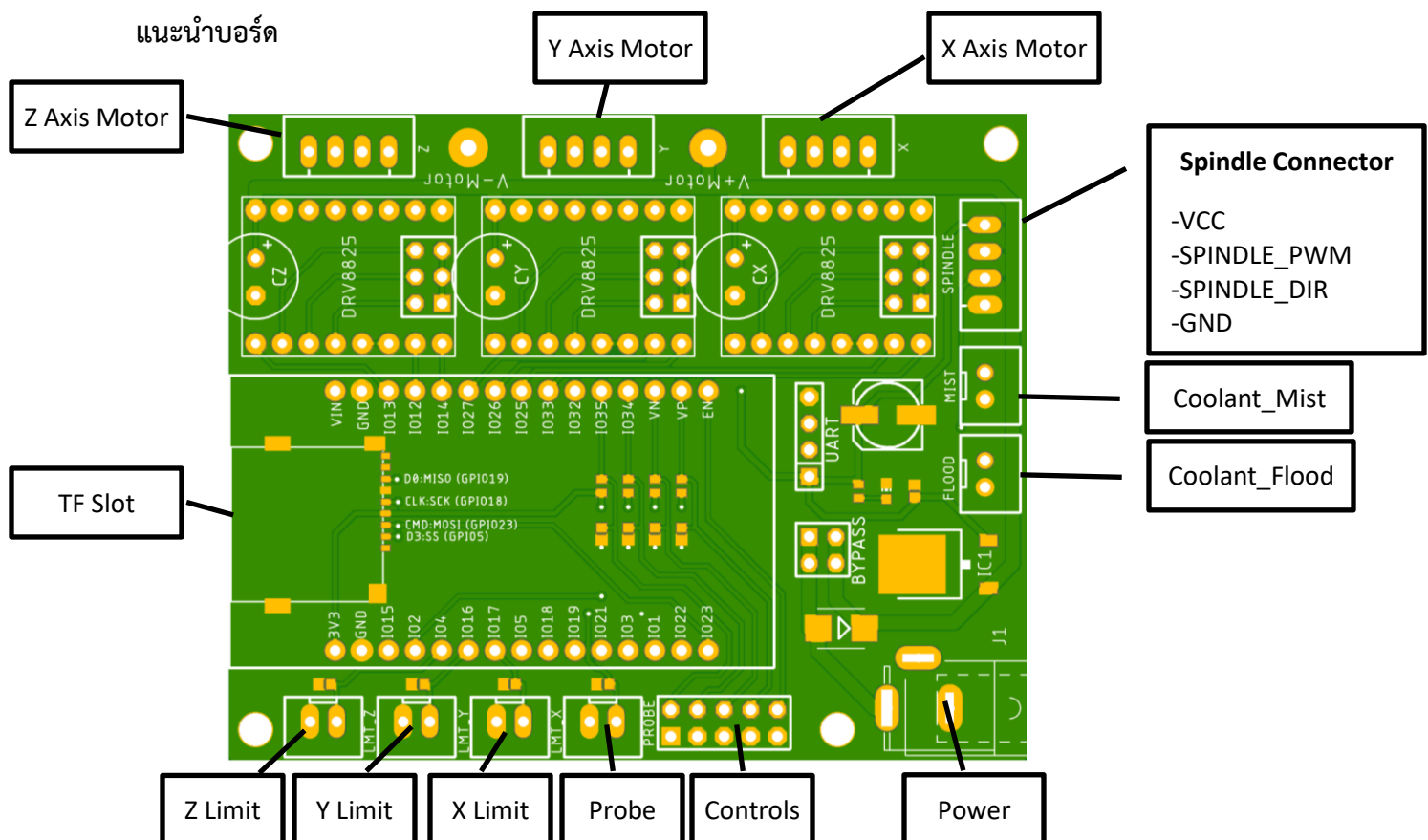


## ESP32\_CNC Shield (v.5ufc)

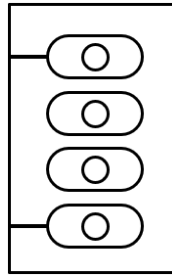
แผงวงจรสำหรับเสียบบอร์ด ESP32 Devkit V.1 สำหรับควบคุมเครื่องมินิ cnc เป็นบอร์ดขนาดเล็กใช้งานง่าย สะดวกในการต่อใช้งาน ไม่ต้องจัมป์สายระโยงระยาง

**ภาพรวม** อย่างที่ทราบกันดีว่าเครื่อง cnc นั้นคือเครื่องจักรที่ทำหน้าที่ในการกัดชิ้นงาน เพื่อให้ได้ส่วนของชิ้นงานที่มีลักษณะตามที่ได้ออกแบบไว้ในคอมพิวเตอร์ โดยการกัดเอาเนื้องานส่วนที่ไม่ต้องการออกจนเหลือเนื้องานตามที่ได้ออกแบบไว้ในที่สุด ถ้าจะให้เข้าใจง่ายๆ เครื่อง cnc ก็คือ สว่านเคลื่อนที่ได้ นั่นเอง สำหรับประเภทของเครื่อง cnc นั้นมีหลายแบบ ถ้าแบ่งประเภทตามแกนของเครื่องก็ได้แก่ 3, 4, 5 ,6..... แกน เป็นต้น



รูปด้านบนนี้คือ บอร์ดชนิดมินิ cnc แบบ 3 แกน ขนาดเท่าของจริง เป็นบอร์ดที่ถูกออกแบบมาด้วยคอนเซ็ปต์ว่าต้องมีขนาดเล็กกะทัดรัด อุปกรณ์ต่อพ่วงบนบอร์ดน้อยชิ้น ซึ่งจะมีผลดีต่อการนำไปใช้งาน โดยเฉพาะการถอดประกอบ นั้นสามารถทำได้ง่ายสะดวกรวดเร็ว ประหยัดเวลา บนบอร์ดประกอบไปด้วยคอนเน็คเตอร์ต่างๆ ได้แก่ **Axis Motor Connector (JST XH2.54 ตัวผู้ 4 ขา)** ซึ่งมีอยู่ 3 แกน ได้แก่ X,Y,Z เชื่อมต่อโดยตรงกับสเตปเปอร์มอเตอร์ขนาดเล็กที่กินกระแสไม่เกิน 3 แอมป์ เนื่องจากบอร์ดออกแบบให้ใช้บอร์ดขับคือ DRV8825 (สามารถใช้โมดูล A4988 แทนได้ เนื่องจากการจัดวางตำแหน่งขาใช้งานเหมือนกัน) ดังนั้น มอเตอร์ที่นำมาใช้จึงต้องเป็นแบบไบโพลาร์ที่มีสายต่อใช้งานจำนวน 4 เส้น ซึ่งท่านต้องทำการสลับตำแหน่งสายให้ถูกต้องหากว่ามอเตอร์นั้นไม่ได้หมุนไปในทิศทางที่ปรกติตามที่ท่านต้องการ

**Spindle Connector (JST XH2.54 ตัวผู้ 4 ขา)** ช่องต่อสำหรับสั่งงานสปินเดิล ประกอบไปด้วยขาสัญญาณต่างๆ เรียงลำดับจากบนลงล่าง ดังนี้



**VCC** คือขาสัญญาณไฟบวก 5v ได้จากไอซี Regulator เบอร์ 7805 ซึ่งจ่ายกระแสได้ 1.5A ไม่แนะนำให้ต่อนำไปใช้งาน เนื่องจาก กระแสไฟอาจไม่เพียงพอจากการจ่ายไฟเลี้ยงอุปกรณ์บนบอร์ดแล้ว

**Spindle\_PWM** เป็นขาต่อสำหรับส่งสัญญาณ PWM ไปควบคุม Spindle ต่อกับ GPIO2

**Spindle\_DIR** เป็นขาต่อทริกสลับการหมุน Spindle ควบคุมโดย M3 (CW) M4 (CCW), ต่อกับ GPIO22  
**GND** ขากราวด์

**Coolant\_Mist (JST XH2.54 ตัวผู้ 2 ขา)** คอนเน็คเตอร์สำหรับเชื่อมต่อระบบหล่อเย็นด้วยหมอกหรือลมเย็น ทั้งนี้ขึ้นอยู่กับการใช้งานว่าจะใช้สิ่งอะไร ซึ่งการควบคุมขานี้ให้เป็น High หรือ Low ทำได้โดยการสั่ง M7 เพื่อให้เป็น High และ M9 เพื่อให้เป็น Low, ขานี้ต่ออยู่กับ GPIO21

**Coolant\_Flood (JST XH2.54 ตัวผู้ 2 ขา)** คอนเน็คเตอร์สำหรับเชื่อมต่อระบบหล่อเย็นด้วยน้ำหล่อเย็น ทั้งนี้ขึ้นอยู่กับการใช้งานว่าจะใช้สิ่งอะไร ซึ่งการควบคุมพอร์ต นี้ให้เป็น High หรือ Low ทำได้โดยการใช้คำสั่ง M8 เพื่อให้เป็น High และ M9 เพื่อให้เป็น Low, ขานี้ต่ออยู่กับ GPIO15

**ข้อสังเกต** การใช้คำสั่ง M9 นั้นจะมีผลทั้ง Coolant\_Mist และ Coolant\_Flood

**X Limit, Y Limit, Z Limit Connector (JST XH2.54 ตัวผู้ 2 ขา)** พอร์ตเชื่อมต่อลิมิตสวิช สำหรับเป็นตัวเซ็นเซอร์ตำแหน่งเริ่มต้นหรือสิ้นสุดของเครื่อง สัญญาณอินพุตเป็นแบบ Active Low หมายความว่า เมื่อสายสัญญาณทั้ง 2 เส้นแตะกัน (หมายถึงสวิตช์กำลังถูกกด กล่าวคือขา Limit กำลังต่อกับ GND นั่นเอง) ซึ่ง limit ทั้ง 3 ต่ออยู่กับขาสัญญาณต่างๆ ดังนี้

X Limit ต่ออยู่กับ GPIO17

Y Limit ต่ออยู่กับ GPIO16

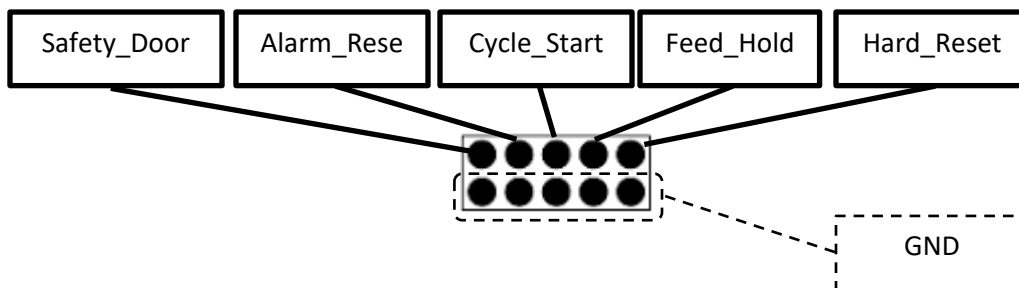
Z Limit ต่ออยู่กับ GPIO4

ถ้าไม่ได้ติดตั้ง Limit Switch ก็สามารถใช้งานเครื่องได้เหมือนปกติ เพียงแต่คุณสมบัติบางอย่างจะไม่สามารถทำงานได้ ทั้งนี้ขึ้นอยู่กับการใช้งาน GRBL



Probe Connector (JST XH2.54 ตัวผู้ 2 ขา) โพรบเชื่อมต่ออยู่กับ GPIO32

Controls (2.54 Pin Header ตัวผู้ 2x5 ขา) ช่องสำหรับเชื่อมต่อกับปุ่มควบคุมต่างๆ



**Safety\_door** ต่อกับขาสัญญาณ GPIO35 และ SMD 0402 10K External Pullup Resistor

**Alarm\_Reset** ต่อกับขาสัญญาณ GPIO34 และ SMD 0402 10K External Pullup Resistor

**Cycle\_Start** ต่อกับขาสัญญาณ GPIO36 และ SMD 0402 10K External Pullup Resistor

**Feed\_Hold** ต่อกับขาสัญญาณ GPIO39 และ SMD 0402 10K External Pullup Resistor

**Hard\_Reset** ต่อกับขาสัญญาณ EN ซึ่งเป็นขาฮาร์ดแวร์รีเซ็ตของ ESP32

DC Power Jack สำหรับต่อไฟเลี้ยง 12V โดยรูตรงกลางจะเป็นขั้ว + ดังรูปสัญลักษณ์ด้านล่างนี้ ห้ามต่อผิดขั้ว



**TF Card Slot** สิ่งที่เพิ่มเติมเข้ามาจากบอร์ด mini cnc แบบเดิมคือบอร์ดเวอร์ชันนี้สามารถรองรับการรันไฟล์ผ่าน micro sd card และสามารถสั่งงานเครื่องผ่านอินเทอร์เน็ตอื่นๆนอกจากสาย usb เช่น BlueTooth และ WiFi ซึ่งตัวเฟิร์มแวร์ที่ใช้เป็น GRBL\_ESP32 นั้นจะฝัง ESP3D ซึ่งเป็น Web Application มาด้วย ทำให้เราสามารถสั่งงานเครื่องบนเว็บเบราว์เซอร์ด้วยโทรศัพท์ ได้ด้วย

การใช้งานบอร์ด ESP32\_CNC Shield V5 (ufc) ต้องเตรียมอุปกรณ์ต่างๆ ดังนี้

- ESP32 Devkit V1 จำนวน 1 ตัว
- โมดูล DRV8825 หรือ A4988 จำนวน 3 ตัว
- วงจรขับเคลื่อนมอเตอร์ Spindle ปกติแล้วจะเป็นวงจรขับเคลื่อนมอเตอร์กระแสตรง ซึ่งหาซื้อได้ทั่วไปอยู่แล้ว
- Micro SD Card 1 ตัว
- สายสัญญาณต่างๆ พร้อมหัวต่อ JST XH2.54 ตัวเมีย 4 ตัวสำหรับต่อ มอเตอร์ และสปีนเดิล, 3 หรือ 4 ตัว สำหรับต่อ Limit Switch และ โพรบ

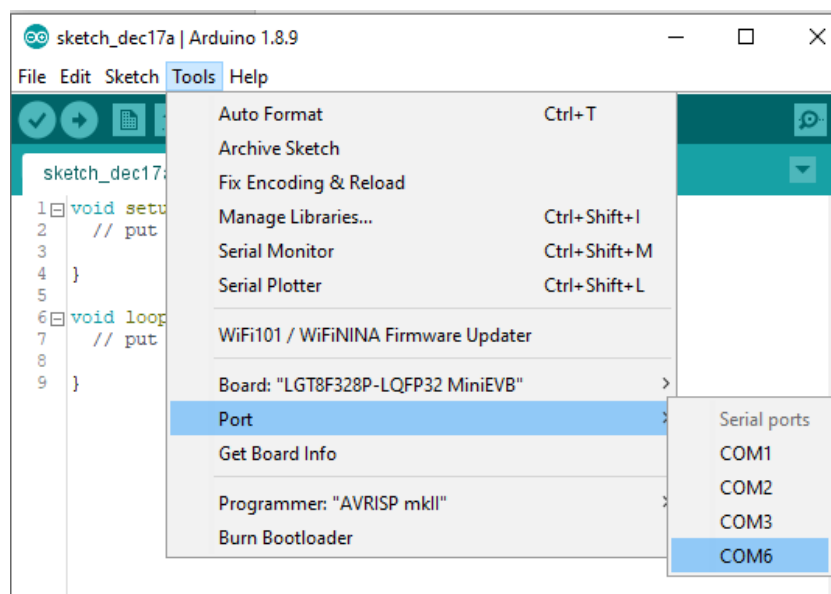
-โปรแกรม Arduino IDE

-ชุดพัฒนา ESP32 บน Arduino IDE (Arduino-ESP32) สามารถดาวน์โหลดได้จาก Github:  
<https://github.com/espressif/arduino-esp32> หรือสามารถติดตั้งผ่าน Boards Manager บน Arduino IDE ได้เช่นกัน โดยใช้ลิงค์บอร์ดตามนี้ [https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

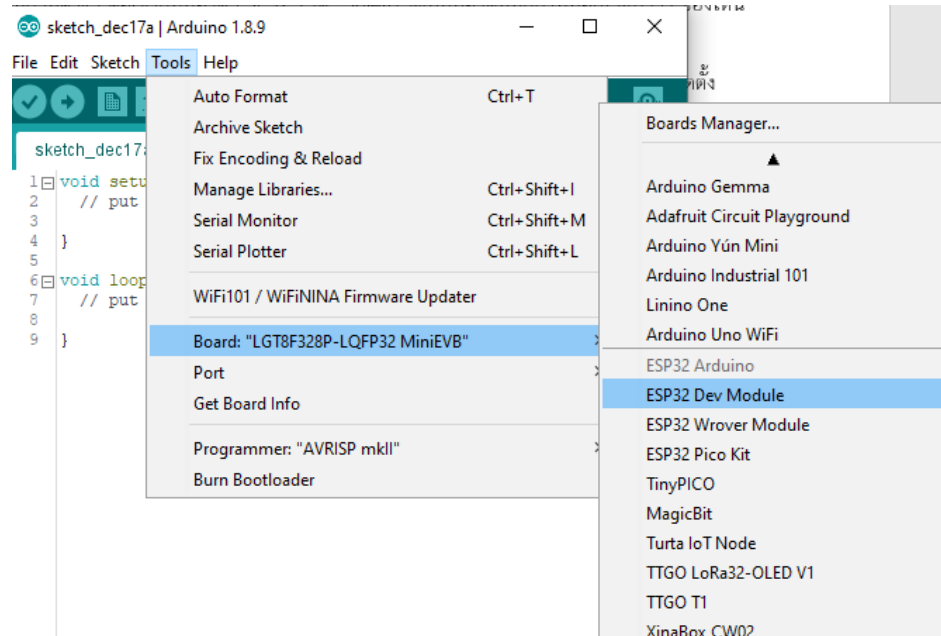
ซึ่งการติดตั้ง ESP32\_SDK for Arduino IDE สามารถศึกษาได้จาก  
<https://www.youtube.com/watch?v=mBaS3YnqDaU>

### ขั้นตอนการใช้งาน

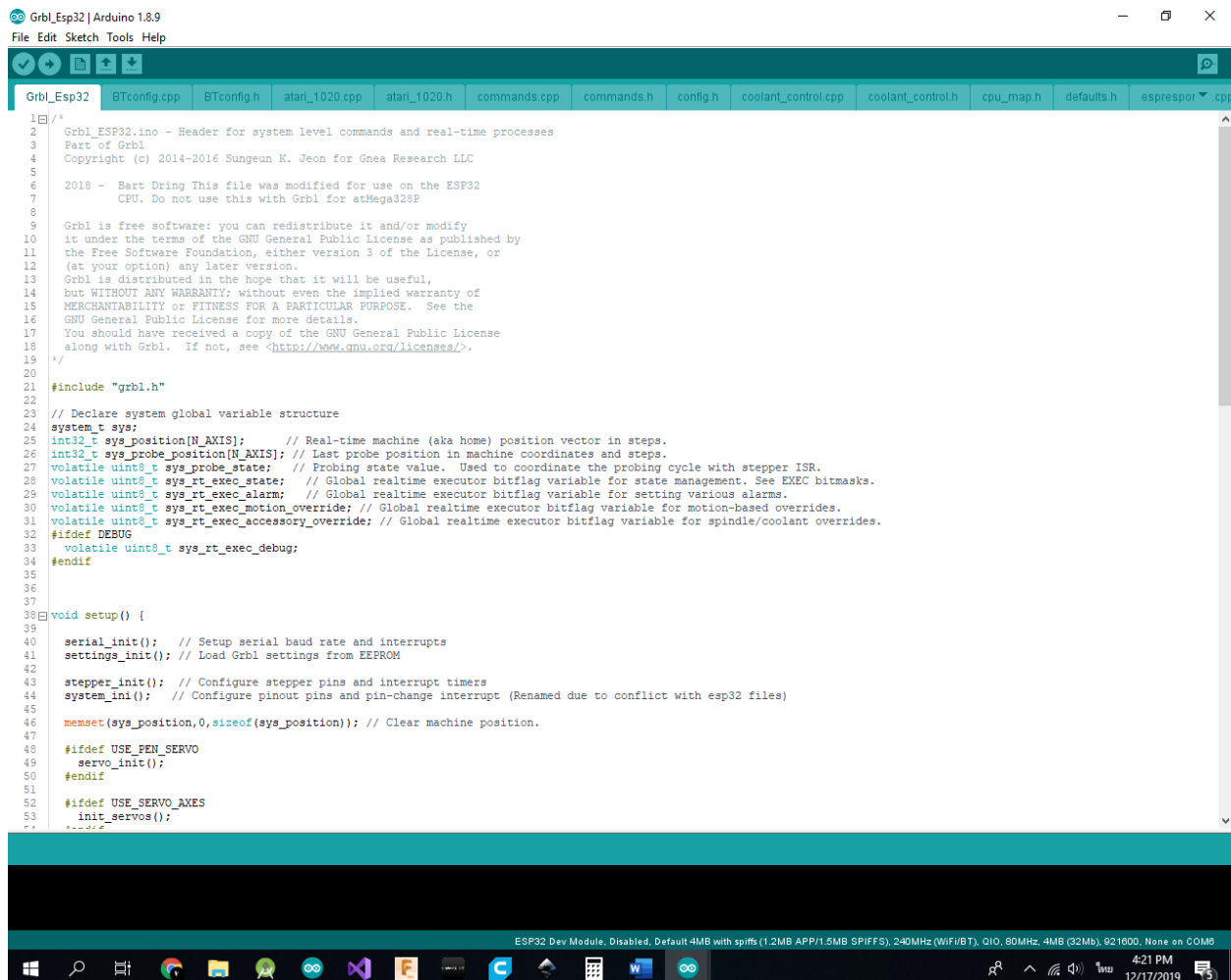
1. นำชิ้นส่วนต่างๆ ต่อเข้ากับบอร์ด ESP32\_CNC Shield ซึ่งบนบอร์ดนั้นสกรีนบอกไว้แล้วว่า ช่องไหนต่อกับอะไร
2. เสียบสายไมโคร usb เข้ากับบอร์ดและคอมพิวเตอร์ จากนั้นทำการเปิด Arduino IDE ที่ได้ติดตั้งบอร์ด ESP32 เรียบร้อยแล้วขึ้นมา
3. ทำการเลือกพอร์ตที่บอร์ดเชื่อมต่ออยู่



## 4. ทำการเลือกบอร์ด เป็น ESP32 Dev Module ดังรูป



5. เปิด Sketch Project ชื่อ Grbl\_Esp32 ขึ้นมา โดยสามารถดาวน์โหลดซอร์สโค้ดโปรเจกต์ดังกล่าวได้จาก [https://github.com/bdring/Grbl\\_Esp32](https://github.com/bdring/Grbl_Esp32)



6. ทำการตรวจสอบไฟล์ “cpu\_map.h” ว่าขาสัญญาณต่างๆ ถูกต้องตามบอร์ดหรือไม่ ถ้าไม่ก็เปลี่ยนให้ตรงกับการเดินลายวงจรของบอร์ด สำหรับ Schematic ของบอร์ดดูได้ในท้ายบทความนี้สำหรับส่วนที่ต้องแก้ไข แก้ไขตำแหน่งขาสัญญาณต่างๆ ตามด้านล่าง

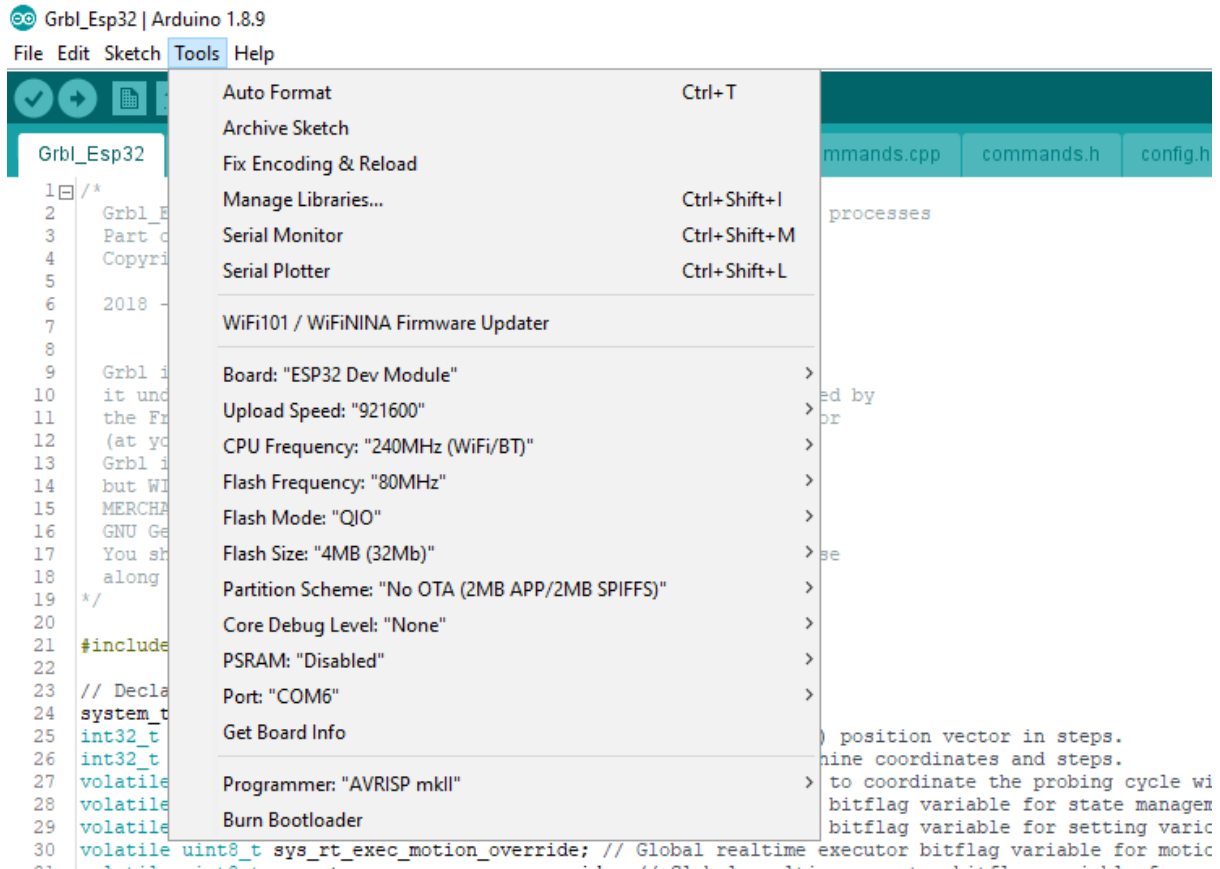
GrbI\_Esp32 - cpu\_map.h | Arduino 1.8.9  
File Edit Sketch Tools Help

```

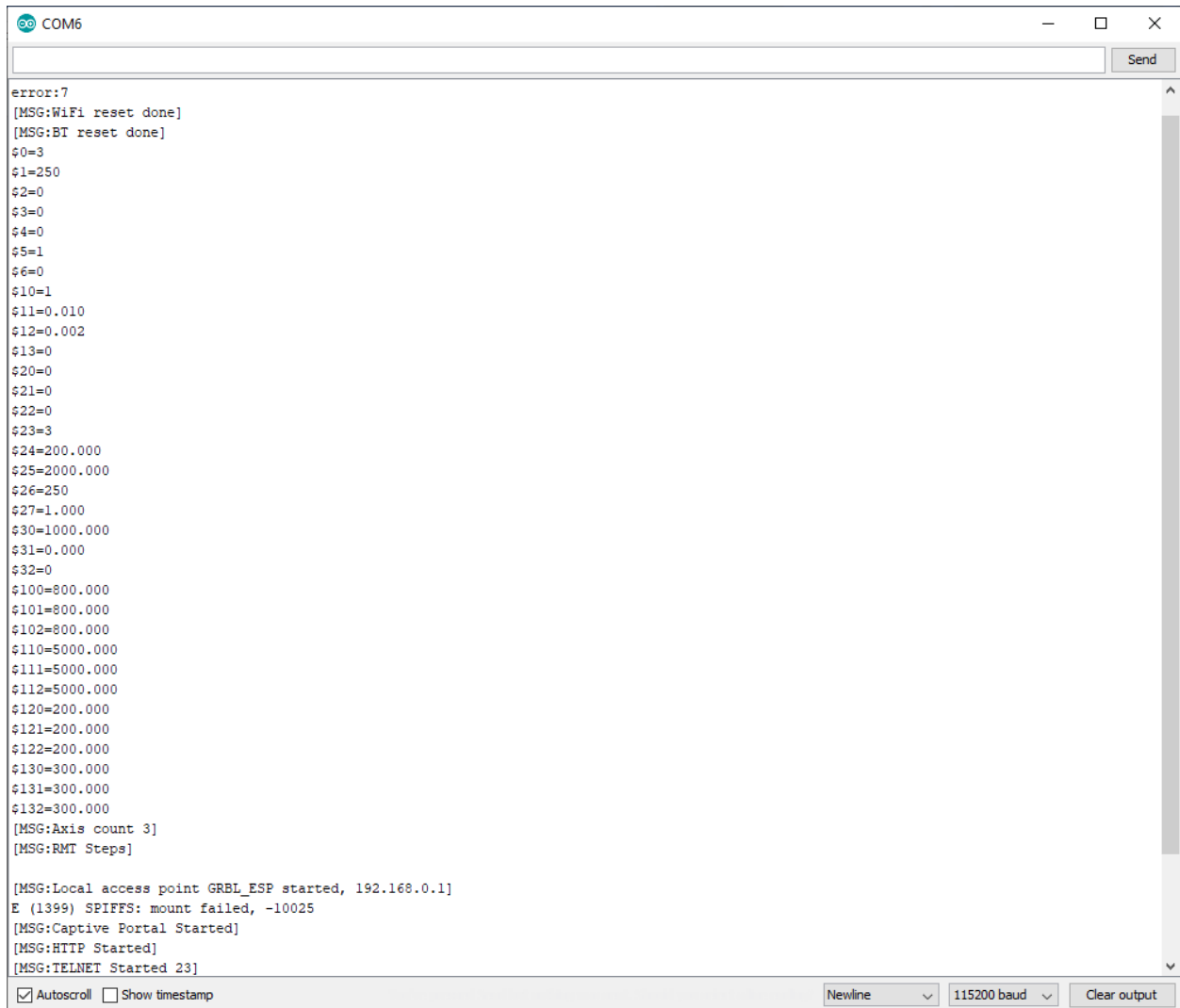
127
128 #define USE_RMT_STEPS
129
130 // It is OK to comment out any step and direction pins. This
131 // won't affect operation except that there will be no output
132 // form the pins. Grbl will virtually move the axis. This could
133 // be handy if you are using a servo, etc. for another axis.
134 #if (defined CPU_MAP_V5)
135 #define CPU_MAP_NAME "CPU_MAP_ESP32_V5_Unofficial"
136 #define X_DIRECTION_PIN GPIO_NUM_33
137 #define Y_STEP_PIN GPIO_NUM_27
138 #define Y_DIRECTION_PIN GPIO_NUM_26
139 #define COOLANT_FLOOD_PIN GPIO_NUM_15
140 #define SPINDLE_PWM_PIN GPIO_NUM_2
141 #define SPINDLE_DIR_PIN GPIO_NUM_22 //for this used please undefine SPINDLE_ENABLE_PIN
142 #define X_LIMIT_PIN GPIO_NUM_17
143 #define Z_LIMIT_PIN GPIO_NUM_4
144 #elif (defined CPU_MAP_V4)
145 #define CPU_MAP_NAME "CPU_MAP_ESP32_V4"
146 #define X_DIRECTION_PIN GPIO_NUM_14
147 #define Y_STEP_PIN GPIO_NUM_26
148 #define Y_DIRECTION_PIN GPIO_NUM_15
149 #define COOLANT_FLOOD_PIN GPIO_NUM_25
150 #define SPINDLE_PWM_PIN GPIO_NUM_2
151 #define SPINDLE_DIR_PIN GPIO_NUM_22
152 #define X_LIMIT_PIN GPIO_NUM_17
153 #define Z_LIMIT_PIN GPIO_NUM_16
154 #elif (defined CPU_MAP_V3p5)
155 #define CPU_MAP_NAME "CPU_MAP_ESP32_V3.5"
156 #define X_DIRECTION_PIN GPIO_NUM_26
157 #define Y_STEP_PIN GPIO_NUM_14
158 #define Y_DIRECTION_PIN GPIO_NUM_25
159 //define COOLANT_FLOOD_PIN GPIO_NUM_16
160 #define SPINDLE_PWM_PIN GPIO_NUM_17
161 #define X_LIMIT_PIN GPIO_NUM_2
162 #define Z_LIMIT_PIN GPIO_NUM_15
163 #endif
164
165 #if (defined CPU_MAP_V5)
166 #define X_STEP_PIN GPIO_NUM_25
167 #define Z_STEP_PIN GPIO_NUM_12
168 #define Z_DIRECTION_PIN GPIO_NUM_14
169 #define Y_LIMIT_PIN GPIO_NUM_16
170 #else
171 #define X_STEP_PIN GPIO_NUM_12
172 #define Z_STEP_PIN GPIO_NUM_27
173 #define Z_DIRECTION_PIN GPIO_NUM_33
174 #define Y_LIMIT_PIN GPIO_NUM_4
175 #endif
176
177 //define Y_STEP_PIN (see hardware shield)
178
179 #define X_DIRECTION_PIN GPIO_NUM_33
180 #define Y_STEP_PIN GPIO_NUM_27
181 #define Y_DIRECTION_PIN GPIO_NUM_26
182 #define COOLANT_FLOOD_PIN GPIO_NUM_15
183 #define SPINDLE_PWM_PIN GPIO_NUM_2
184 #define SPINDLE_DIR_PIN GPIO_NUM_22 //for this used please undefine SPINDLE_ENABLE_PIN
185 #define X_LIMIT_PIN GPIO_NUM_17
186 #define Z_LIMIT_PIN GPIO_NUM_4
187 #define X_STEP_PIN GPIO_NUM_25
188 #define Z_STEP_PIN GPIO_NUM_12
189 #define Z_DIRECTION_PIN GPIO_NUM_14
190 #define Y_LIMIT_PIN GPIO_NUM_16
191 #define STEPPERS_DISABLE_PIN GPIO_NUM_13
192 #define COOLANT_MIST_PIN GPIO_NUM_21
193 #define PROBE_PIN GPIO_NUM_32
194
195 #define CONTROL_SAFETY_DOOR_PIN GPIO_NUM_35 // needs external pullup
196 #define CONTROL_RESET_PIN GPIO_NUM_34 // needs external pullup
197 #define CONTROL_FEED_HOLD_PIN GPIO_NUM_36 //vp needs external pullup
198 #define CONTROL_CYCLE_START_PIN GPIO_NUM_39 //vn needs external pullup

```

7. ทำการเลือก Partition Scheme เป็น “No OTA (2MB APP/2MB SPIFFS)” โดยคลิกที่เมนู Tools>Partition Scheme:> No OTA (2MB APP/2MB SPIFFS) ตรงนี้จะทำให้บอร์ดมีหน่วยความจำแฟลชฟรี 2MB หลังจากอัปโหลดโปรแกรมลงบอร์ดแล้ว ทำให้สามารถใส่ไฟล์ G-code ขนาดเล็ก เพื่อสั่งรันเครื่องได้ โดยเฉพาะกรณีที่ไม่ใช้ TF Card เสียบอยู่บนบอร์ด เป็นการใช้องค์หน่วยความจำ SPIFFS ที่เหลืออยู่ให้เกิดประโยชน์สูงสุด



8. ทำการอัปโหลดโปรแกรมลงบน ESP32 Devkit V1 ในขั้นตอนนี้ ข้อควรระวังคือ บอร์ด ESP32 Devkit V1 นั้นโดยปกติ เราจะต้องทำการกดปุ่ม Boot ค้างไว้จนกว่าโปรแกรมจะเข้าสู่ขั้นตอนการอัปโหลด เนื่องจากปัญหาทางเทคนิคของตัวบอร์ดเอง
9. หลังจากทำการอัปโหลดเสร็จสิ้น บอร์ด ESP32\_CNC ก็พร้อมรัน Grbl แล้ว ตรวจสอบได้โดยการเปิด Serial Monitor จะพบหน้าจอคล้ายด้านล่าง



```
error:7
[MSG:WiFi reset done]
[MSG:BT reset done]
$0=3
$1=250
$2=0
$3=0
$4=0
$5=1
$6=0
$10=1
$11=0.010
$12=0.002
$13=0
$20=0
$21=0
$22=0
$23=3
$24=200.000
$25=2000.000
$26=250
$27=1.000
$30=1000.000
$31=0.000
$32=0
$100=800.000
$101=800.000
$102=800.000
$110=5000.000
$111=5000.000
$112=5000.000
$120=200.000
$121=200.000
$122=200.000
$130=300.000
$131=300.000
$132=300.000
[MSG:Axis count 3]
[MSG:RMT Steps]

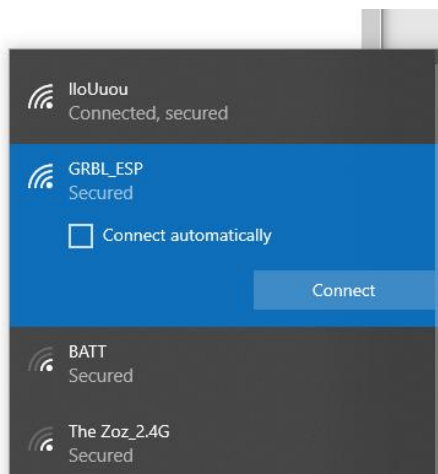
[MSG:Local access point GRBL_ESP started, 192.168.0.1]
E (1399) SPIFFS: mount failed, -10025
[MSG:Captive Portal Started]
[MSG:HTTP Started]
[MSG:TELNET Started 23]
```

เห็นข้อความ error ไม่ต้องตกใจ เริ่มต้นทำการตั้งค่าบอร์ดก่อนโดยเชื่อมต่อ WiFi ที่บอร์ดปล่อยออกมา ชื่อ GRBL\_ESP ซึ่งเป็นชื่อ WiFi SSID เริ่มต้น และรหัสผ่านเป็น 12345678 ซึ่งค่าเหล่านี้ถูกต้องตั้งมาในส่วนของซอร์สโค้ดในไฟล์ที่ชื่อ wificonfig.h ท่านสามารถเปลี่ยนเป็นอย่างอื่นได้

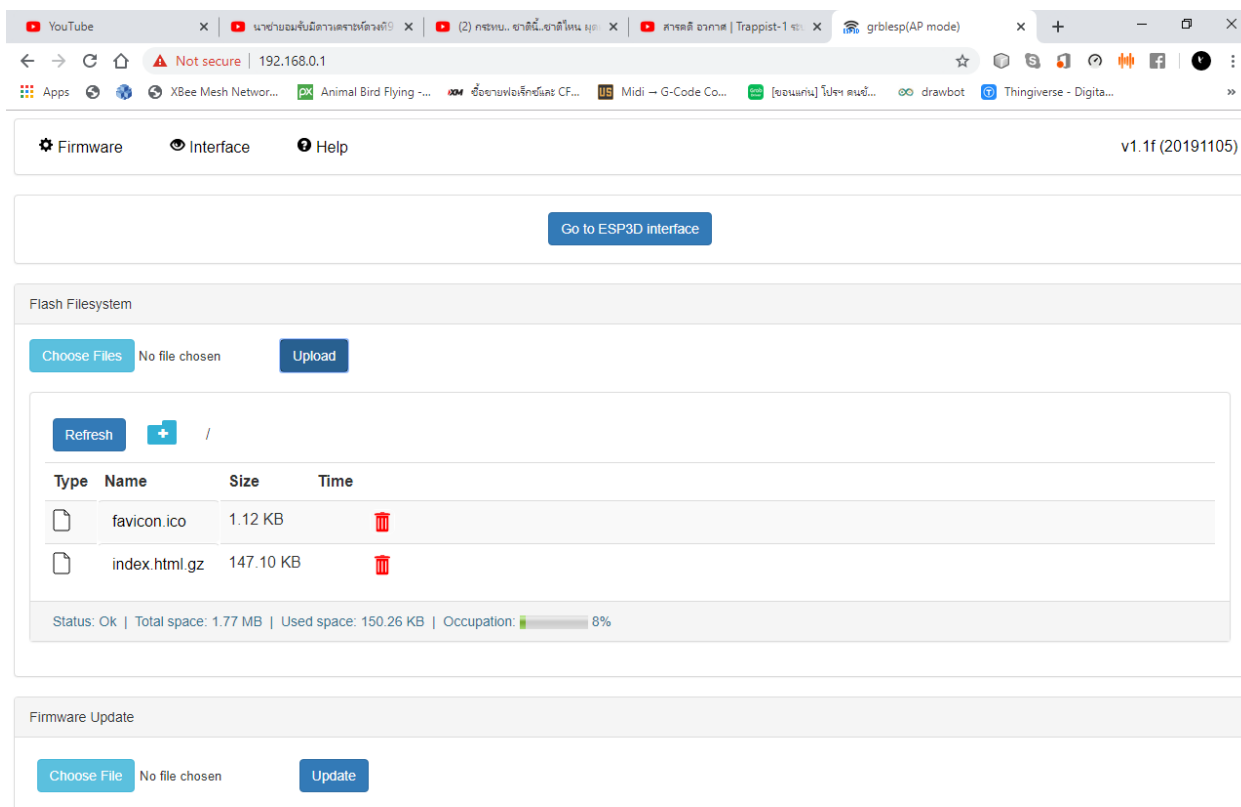
```
#define DEFAULT_AP_SSID "GRBL_ESP"
```

```
#define DEFAULT_AP_PWD "12345678"
```



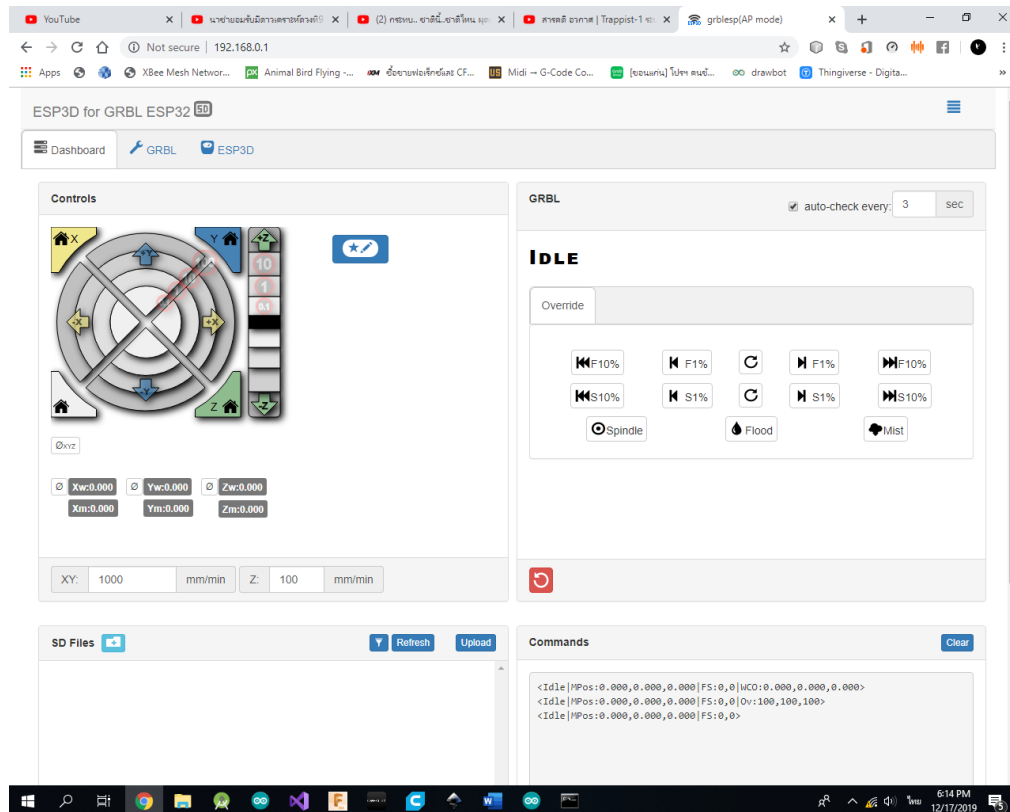


10. เมื่อเชื่อมต่อ WiFi ดังกล่าวแล้ว ทำการเปิดเว็บเบราว์เซอร์ขึ้นมาแล้วเข้าไปที่ <http://192.168.0.1> จะพบหน้าจอด้านล่าง จากนั้นทำเลือกเลือกและอัปโหลดไฟล์หน้าเว็บ ESP3D โดยไฟล์นั้นถูกบีบอัดอยู่ในชื่อ index.html.gz



จากตัวอย่าง ผู้เขียนได้ทำการอัปโหลดจำนวน 2 ไฟล์ ได้แก่ index.html.gz และ favicon.ico เมื่ออัปโหลดเสร็จเรียบร้อยแล้ว ทำการคลิกที่ปุ่ม **Go to ESP3D interface** จะพบหน้าจอด้านล่าง

เขียนโดย: นักประดิษฐ์สแตนดาร์ดอะไลน์



และเมื่อกลับไปดูที่ Serial Monitor ของ Arduino IDE และลองกดปุ่ม รีเซ็ต หรือปุ่ม EN หนึ่งครั้ง  
ปรากฏว่าข้อความ error ที่เห็นในตอนแรกนั้น ไม่ขึ้นมาอีกแล้ว

```

COM6
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

[MSG:Axis count 3]
[MSG:RMT Steps]

[MSG:Local access point GRBL_ESP started, 192.168.0.1]
[MSG:Captive Portal Started]
[MSG:HTTP Started]
[MSG:TELNET Started 23]
[MSG:Using cpu_map...CPU_MAP_ESP32_V5_Unofficial]

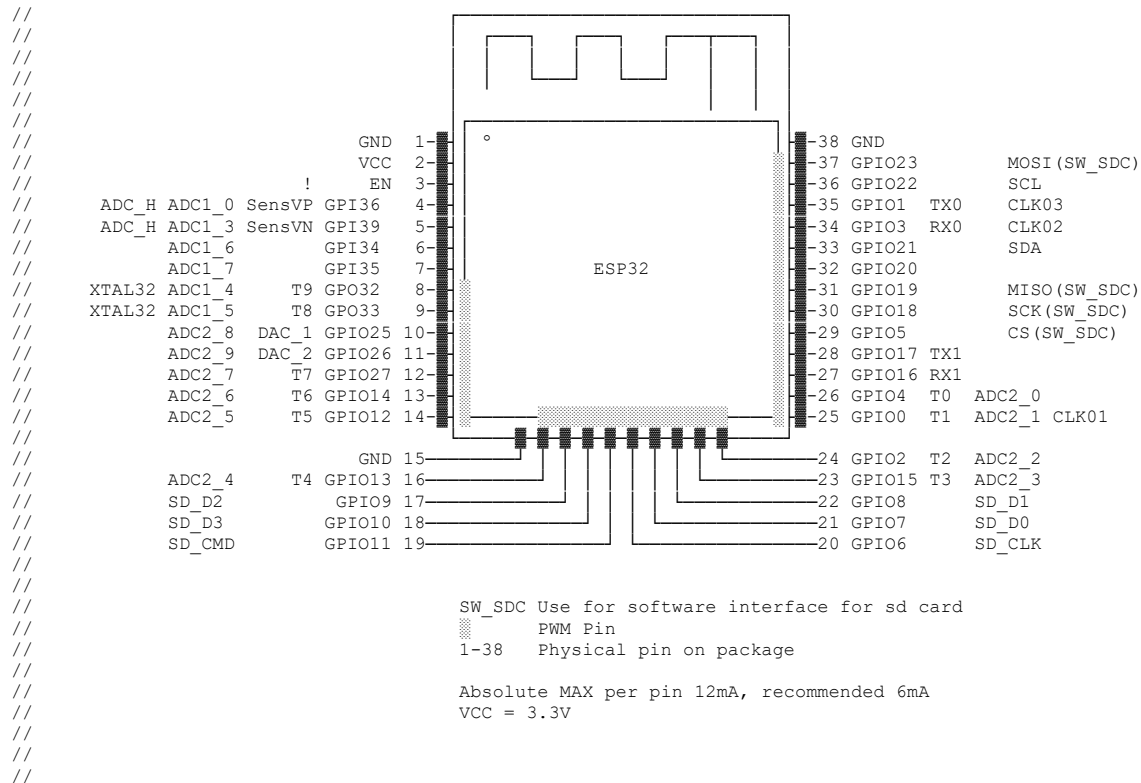
Grbl 1.1f ['$' for help]
  
```

การตั้งค่าการใช้งาน ไม่ว่าจะเป็นในส่วนของ ESP3D interface หรือ GRBL Setting ซึ่งบทความนี้เป็นบทความที่เขียนขึ้นมาเพื่อสนับสนุนการใช้งาน บอร์ด ESP32\_CNC shield ofc เท่านั้น สำหรับ

การใช้งานในขั้นตอนอื่นๆ นั้นท่านสามารถศึกษาเพิ่มเติมได้จากอินเทอร์เน็ต ซึ่งมีอยู่ทั่วไป หรือ ท่านสามารถติดต่อสอบถามเพิ่มเติมได้ทางเพจ

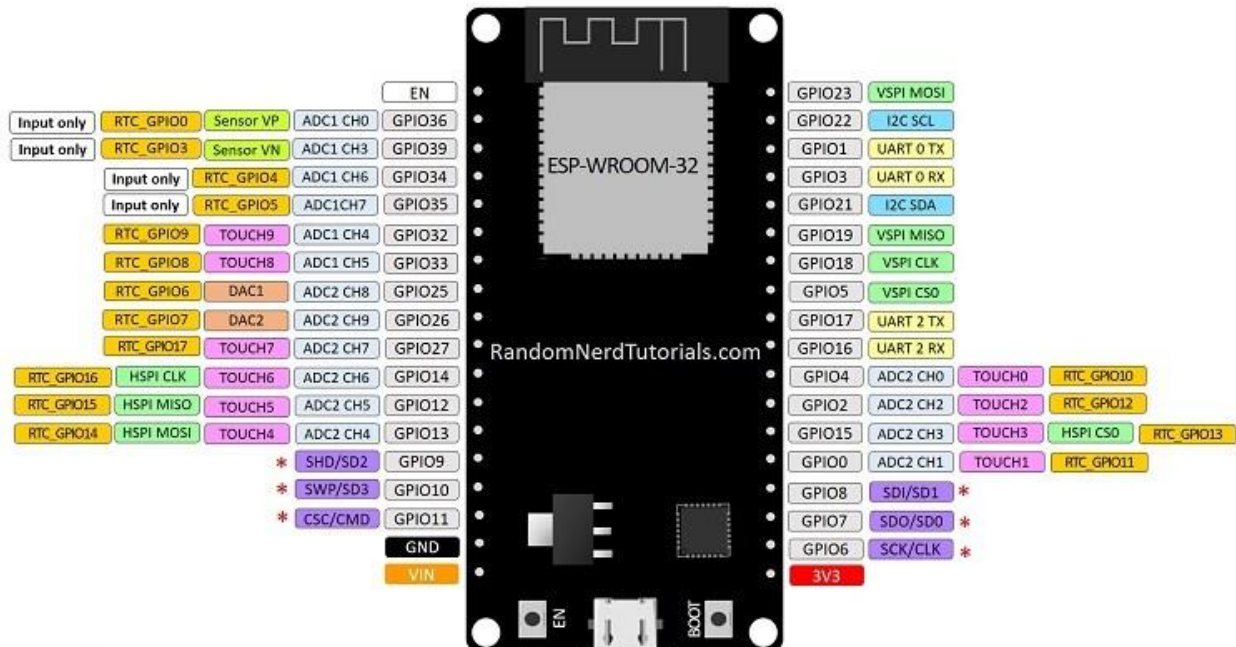
<https://www.facebook.com/arduinoprojectkits>

ทางผู้เขียนและทีมงานยินดีให้คำแนะนำ สำหรับบทความนี้ก็มีเพียงเท่านี้ ขอกล่าวคำว่า “สวัสดีครับ”



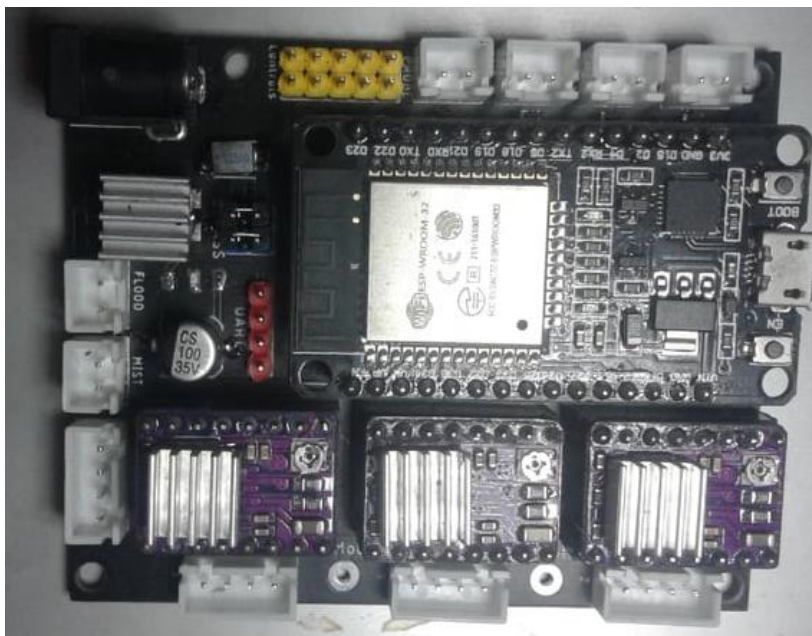
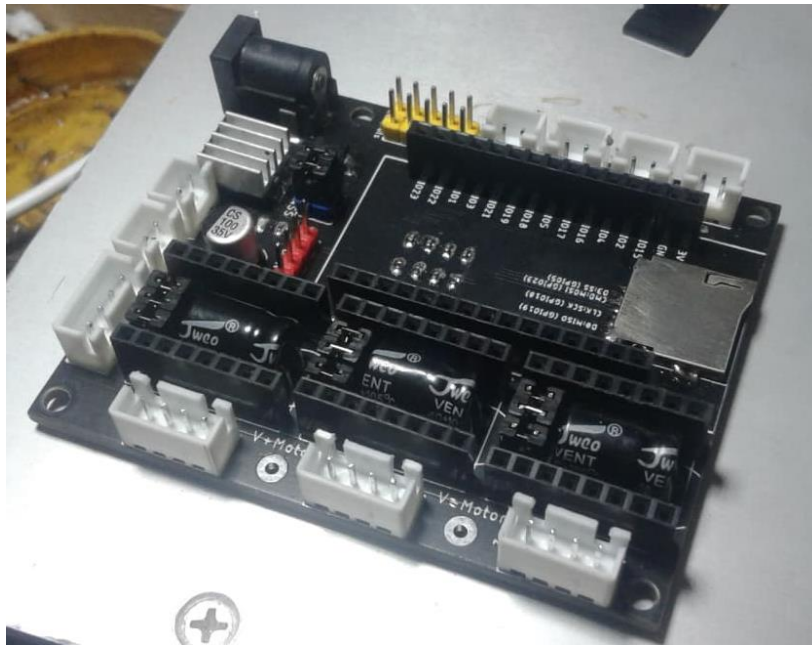
รูป แสดงขาสัญญาณของโมดูล ESP32 และบอร์ด ESP32 Devkit V1

## ESP32 DEVKIT V1 – DOIT version with 36 GPIOs



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.





## ลิงค์ที่เกี่ยวข้อง

เว็บไซต์อย่างเป็นทางการของผู้พัฒนา Arduino

<https://www.arduino.cc>

Github page ของไมโครคอนโทรลเลอร์ ESP32 สำหรับ arduino

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

เพจของทีมพัฒนาบอร์ดตามบทความนี้ (เพจคนไทย)

<https://www.facebook.com/ArduinoProjectkits>

เพจซอร์สโค้ดเฟิร์มแวร์ GRPB\_ESP32

[https://github.com/bdring/Grbl\\_Esp32](https://github.com/bdring/Grbl_Esp32)

[https://github.com/cchian/Grbl\\_Esp32](https://github.com/cchian/Grbl_Esp32) (ปรับแต่งให้เข้ากับบอร์ดแล้ว)

เพจซอร์สโค้ดเว็บอินเทอร์เฟซสำหรับสั่งงานเครื่อง mini cnc grbl

<https://github.com/luc-github/ESP3D-WEBUI>

<https://github.com/cchian/ESP3D-WEBUI> (ปรับแต่งแล้ว เพิ่มภาษาไทยเข้าไปแล้ว)

ช่องของนักประดิษฐ์ รับข่าวสารความรู้ใหม่ๆ จาก นักประดิษฐ์สแตนดาร์ดอะโตน อย่าลืมกดติดตามและกระดิ่งกันนะครับ

<https://www.youtube.com/cchian1>

<https://www.facebook.com/wichian.toso>

<https://web.facebook.com/kissada.po>

ลิงค์บอร์ด ESP32 สำหรับติดตั้งบน Arduino IDE

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)