

ESP-SDWIFI

v1.0 (2017)

SD-Card & Wifi Module

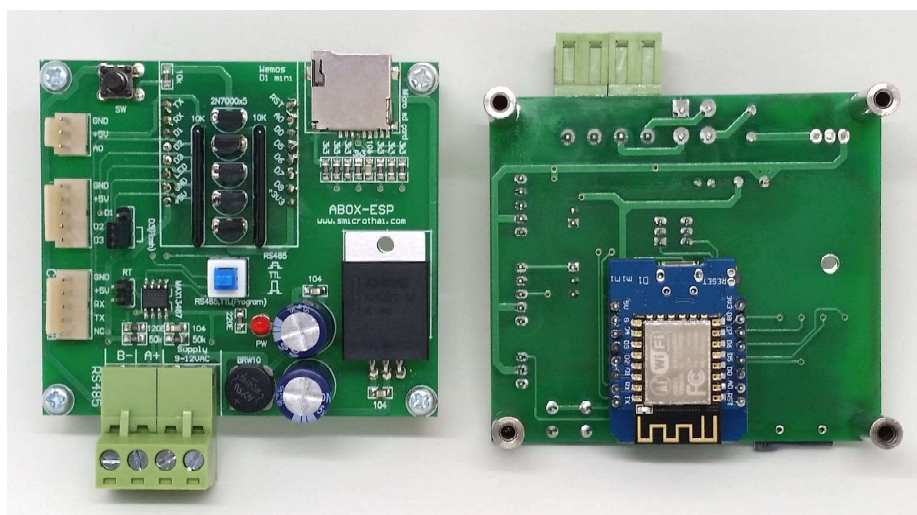


ESP-SDWIFI คือ App สำหรับสินค้า ABOX-ESP ช่วยเสริมการประยุกต์ใช้งานเกี่ยวกับ SD-Card และการสื่อสารผ่าน Wifi ได้อย่างสะดวกที่สุด สามารถรับส่งคำสั่งได้ผ่านทาง RS485 หรือ RS232-TTL รองรับการอ่านและเขียนไฟล์ (.TXT) กับ SD-Card และการสื่อสารผ่าน Wifi ที่มีรูปแบบ Protocol ยอดนิยมคือ http เหมาะกับการนำไปใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ต่างๆ เพื่อเพิ่มความสามารถทางด้าน SD-Card หรือ Wifi ที่เข้ากับยุคของ IOT (Internet of Thing) ได้เป็นอย่างดี ลดภาระการทำงานของ MCU หลัก รวมถึงลดภาระการเขียนโปรแกรมไปได้มาก

คุณสมบัติ

- MCU ยอดนิยม ESP8266 Clock 80 MHz
- ภายใต้มือถือ WeMos D1 mini และมี LED (สีฟ้า) บนโมดูลเพื่อแสดงสถานะต่าง ๆ
- มีช่องเสียบ Micro-SD-Card และมีปุ่มกด 1 ตัว พร้อม LED แสดง Power
- มีขั้ว RS485 และขั้ว RX,TX แบบ TTL (5 Pin) เลือกลงใช้งานได้ด้วยปุ่มกด (สีฟ้า)
- คุณสมบัติการสื่อสารคือ Speed=9600 Parity=None Data=8 Stop=1
- มีชุดคำสั่ง 7 คำสั่งสำหรับ SD-Card (Fat16,32) สามารถอ่าน,เขียนไฟล์ (Text-File) ได้อย่างสะดวก โดยกระทำที่โฟลเดอร์ Root (/) เท่านั้น
- มีชุดคำสั่ง 6 คำสั่งสำหรับสื่อสารผ่าน Wifi รองรับ http protocol ยอดนิยม สามารถเป็นฝั่ง Client เพื่อรับส่งข้อมูลกับ Web Server ได้ หรือจะเป็นฝั่ง Web Server เพื่อให้คอมพิวเตอร์ติดต่อมาด้วย Browser ก็ได้
- ในด้านสัญญาณ Wifi ก็ตั้งเป็น Station หรือเป็น Access Point ก็ได้
- การตั้งค่าต่าง ๆ ทำได้สะดวกด้วยการใส่เป็นข้อมูลในไฟล์ config.txt ในแผ่น SD-Card ได้เลย
- มีระบบ Auto-Send และ Auto-Echo เพื่อช่วยให้ทดลองได้สะดวกยิ่งขึ้น
- ใช้ไฟเลี้ยง 9-12 VAC,VDC จากหม้อแปลง หรือจาก Supply อื่น ๆ ได้
- ขนาดบอร์ด 75 x 80 mm (รวมส่วนยื่นของขั้วไขว้น๊อต)

ภาพแสดงตัวเครื่อง



ชุดคำสั่งเกี่ยวกับ SD-Card

ESP-SDWIFI รับคำสั่งทาง RS485 (ขั้วเขียว) หรือ RS232-TTL (ขั้ว 5 Pin) โดยจะเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ใด ๆ คุณสมบัติคือ Speed=9600 Data=8 Stop=1 Parity=No ชุดคำสั่งที่ใช้จะเป็นแบบ Ascii Command โดยส่งคำสั่งไป แล้ว ESP-SDWIFI จะตอบกลับมาเสมอ สามารถทดลองคำสั่งได้โดยใช้โปรแกรมสื่อสารแบบ Terminal ที่ทำงานบนเครื่องคอมพิวเตอร์ เพราะมีรูปแบบเป็นตัวอักษร Ascii ทั้งหมด คำสั่งจะมีรูปแบบดังนี้...

:c\r

: คือรหัสนำหน้า (0x3A)

c คือคำสั่งเป็นตัวเลข หรือตัวอักษรตัวเดียว และอาจมีหรือไม่มีข้อมูลติดตามมาก็ได้

\r คือรหัสลงท้าย (0x0D)

ชุดคำสั่งจะสรุปได้ตามตารางต่อไปนี้ ในวงเล็บ (R) หมายถึงคำสั่งที่ส่งไป และ (T) คือข้อมูลที่ตอบกลับ โดยจะแสดงเป็นตัวอย่างสมมุติเพื่อให้เข้าใจ ตัวคำสั่งจะแสดงแบบไม่ใส่รหัสลงท้าย \r เพื่อให้ดูสบายตา ส่วนค่าภายใน [] ต่อท้ายคำสั่ง หมายถึง Option คือใส่หรือไม่ใส่ก็ได้ ซึ่งถ้าใส่จะให้ความหมายทำนอง “เขียนค่า” (write) และการไม่ใส่คือ “อ่านค่า” (read) ส่วนการตอบกลับก็จะแยกเป็น 2 กรณีด้วย คือแบบปกติ กับภายใน [] อีกที

รูปแบบคำสั่ง	การใช้งาน
(R) :Z (T) ESP-SDWIFI vx.x	Check (Show Model) แสดงรุ่นของสินค้าและหมายเลข Version ของ Firmware
(R) :1 (T) TEST.TXT 30 (T) CONFIG.TXT 120 (T) OK	List File แสดงรายชื่อไฟล์พร้อมนามสกุล (เฉพาะโฟลเดอร์ / Root) รวมทั้งแสดงจำนวน Byte ของไฟล์ด้วย โดยแสดงบรรทัดละไฟล์ และเมื่อครบแล้วจะลงท้ายด้วย OK อีกที
(R) :2TEST.TXT (T) OK,ER	Remove File ลบไฟล์ออกจาก SD-Card ถ้าเรียบร้อยก็จะส่ง OK กลับมา แต่ถ้าไม่สำเร็จ เช่นไม่มีไฟล์นี้ก็จะส่ง ER
(R) :3WTEST.TXT (T) OK,NO,ER	Open File เปิดไฟล์เพื่อจะใช้งาน โดยอักษร W หมายถึงเปิดเพื่อจะ Write คือถ้าไม่มีไฟล์นี้ก็จะสร้างให้ใหม่ และพร้อม Write ตั้งแต่ต้นเลย แต่ถ้ามีไฟล์นี้อยู่แล้ว ก็จะเปิดไว้ให้ และพร้อม Write ต่อท้ายทันที และถ้าใส่เป็นอักษร R หมายถึง Read ซึ่งต้องมีไฟล์อยู่แล้วเท่านั้น และจะพร้อมอ่านตั้งแต่ต้นไฟล์เลย ถ้าเรียบร้อยก็จะส่ง OK กลับมา แต่ถ้าไม่มีไฟล์ก็จะส่ง NO หรืออาจจะมีปัญหาอื่น ๆ ก็จะส่ง ER

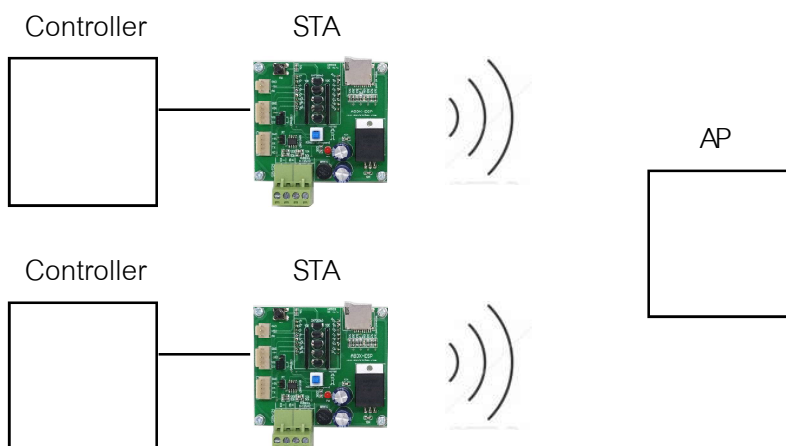
รูปแบบคำสั่ง	การใช้งาน
(R) :4 (T) LINE1,EN,ER	Read Line อ่านข้อความจากไฟล์ที่เปิดไว้ ที่ละบรรทัด LINE1 คือข้อความจากไฟล์ แต่ถ้าไม่สามารถอ่านได้ก็จะส่ง ER หรืออ่านจนครบทั้งหมดของไฟล์แล้ว ก็จะส่ง EN กลับมาให้ทราบ
(R) :5TEXT-LINE1 (T) OK,ER	Write Line เขียนข้อความลงไฟล์ที่เปิดไว้ ที่ละบรรทัด TEXT-LINE1 คือข้อความ จะเขียนลงในไฟล์ ถ้าเรียบร้อยดีก็จะส่ง OK กลับมา แต่ไม่ก็ส่ง ER
(R) :6 (T) OK,ER	Close File ปิดไฟล์ที่ใช้งาน คือเมื่อทำการ Read,Write ครบถ้วนแล้ว ก็ให้ทำการ ปิดไฟล์และพร้อมเปิดใช้งานไฟล์อื่น ๆ ได้ต่อไป ถ้าเรียบร้อยดีก็จะส่ง OK กลับมา ถ้าไม่ก็ส่ง ER
(R) :7TEST.TXT (T) TEXT-LINE1,ER (T) TEXT-LINE2,ER (T) EN	Show File สำหรับแสดงข้อความในไฟล์ทั้งหมดในคราวเดียว และจะลงท้ายด้วย EN ถ้ามีปัญหาใด ๆ ในการแสดง ก็จะส่ง ER กลับมาในแต่ละบรรทัดด้วย การแสดงไฟล์นี้จะไม่ถือว่าได้เปิดไฟล์ค้างไว้แต่อย่างใด

หมายเหตุ ... การเสียบหรือดึงออกแผ่น SD-Card ให้กระทำช่วงไม่มีไฟเลี้ยงเท่านั้น ซึ่งถ้ากระทำช่วงมีไฟเลี้ยง คำสั่งต่าง ๆ เกี่ยวกับ SD-Card จะไม่แน่นอน และส่งผลให้ได้ข้อมูลไม่ครบถ้วนสมบูรณ์

ภาพแสดงการประยุกต์ใช้งาน Wifi



- เป็น Client เพื่อรับส่งข้อมูลกับ Web Server (ทั้งภายในหรือภายนอก)
- เป็น Web Server เพื่อให้คอมพิวเตอร์ในระบบเข้าถึงด้วย Browser ได้



- เป็น Server หรือ Client เพื่อรับส่งข้อมูลระหว่าง Controller



- เป็น Web Server เพื่อการตั้งค่าต่าง ๆ ผ่านอุปกรณ์ที่มี Wifi และ Browser



- เป็น Server หรือ Client เพื่อรับส่งข้อมูลระหว่าง Controller กันเอง

หลักการใช้งานทั่วไป และไฟล์ CONFIG.TXT

- สำหรับการใช้งานเกี่ยวกับ Wifi จะต้องใส่ SD-Card ไว้เสมอ โดยจะต้องมีไฟล์ CONFIG.TXT ไว้เพื่อกำหนดคุณสมบัติต่าง ๆ ของ Wifi โดยกำหนดไว้ได้ก่อนด้วยเครื่องคอมพิวเตอร์ทั่วไป
- ข้อความในไฟล์ CONFIG.TXT จะเป็นได้ดังนี้ ...

mode=STA,AP	STA=Station AP=Access-Point
name=smicro	คือชื่อ Wifi ที่จะใช้งาน หรือที่เรียกว่า SSID นั้นเอง
pass=12345678	คือ password สำหรับชื่อ SSID ที่ใช้งาน
ipad=192.168.1.50	กำหนด Fix-IP ของตัวเอง
gate=192.168.1.1	กำหนด Gateway IP ของระบบที่เชื่อมต่อ
subn=255.255.255.0	กำหนด Subnetmask ของระบบ
host=www.smicrothai.com	กำหนด Host ที่จะเชื่อมต่อด้วยในฐานะ Client

- สำหรับการเชื่อมต่อสัญญาณ Wifi นั้น ESP-SDWIFI สามารถทำตัวเป็น STA เพื่อไปเชื่อมกับ AP ที่มีอยู่ได้ หรือจะทำตัวเป็น AP เพื่อให้อุปกรณ์อื่น ๆ มาเชื่อมต่อด้วยก็ได้ โดยต้องเลือกใช้งานได้อย่างใดอย่างหนึ่ง
- การกำหนด mode name และ pass จะต้องทำควบคู่กันเสมอ ถ้าเป็น STA ชื่อ name,pass ก็ต้องตรงกับที่แต่ละสถานที่กำหนดไว้จริง ส่วนกรณีเป็น AP ชื่อ name,pass ก็คือชื่อ SSID ที่ต้องการให้อุปกรณ์มาเชื่อมต่อนั่นเอง ทั้งนี้ pass จะต้องมียาวน้อย 8 ตัวเสมอ ถ้าต่ำกว่านั้น เครื่องจะถือว่าไม่กำหนดไว้ คือสามารถเข้าถึงได้แบบไม่มี password นั้นเอง
- การกำหนด ipad gate subn จะต้องทำให้ครบทั้ง 3 บรรทัดเสมอ จึงจะมีผลจริง แต่ถ้าไม่ครบ หรือไม่กำหนดเลย ก็จะเป็นการใช้ระบบ Auto (หรือที่เรียกว่า DHCP) คือถ้าเป็น STA ก็จะได้รับค่าทั้ง 3 จากระบบที่เชื่อมต่อ แต่ถ้าเป็น AP ก็จะใช้ค่า Default ดังนี้ ipad=192.168.4.1 gate=192.168.4.1 subn=255.255.255.0 และตัวอุปกรณ์ที่มาเชื่อมต่อ ถ้าไม่กำหนด Fix-IP ไว้ ตัว AP ก็จะจ่าย ip ให้ตั้งแต่ 192.168.4.2 เป็นต้นไป
- เมื่อจ่ายไฟเข้าบอร์ด เครื่องจะกระพริบไฟสีฟ้า (บนโมดูล WeMos) 2 ครั้งให้รับทราบ และจะพยายามเชื่อมต่อ Wifi ให้ทันที โดยถ้าเป็นโหมด STA จะพยายามเชื่อมใน 30 วินาที ถ้าเชื่อมไม่ได้ก็จะถือว่าเป็นสถานะ OFF คือ LED ดับ แต่ถ้าเชื่อมได้ LED จะสว่าง ส่วนถ้าเป็นโหมด AP ตัว LED ก็สว่างและรออุปกรณ์มาเชื่อมต่ออีกที

- ในการรับคำสั่งใด ๆ จาก RS485 (RS232-TTL) หรือเมื่อมีการสื่อสารใด ๆ จากด้าน Wifi ตัว LED สีฟ้าจะกระพริบให้รับทราบด้วยเสมอ
- เมื่อเชื่อมต่อผ่าน Wifi ได้แล้ว รายละเอียดการรับส่งข้อมูลในรูปแบบ http protocol จะแยกเป็น 2 รูปแบบคือ
 - Server Process ... คือรอให้มีการติดต่อมาจาก Client แล้วจึงทำการตอบกลับไป
 - Client Process ... คือส่งข้อมูลไปติดต่อกับ Server (หรือ Host) แล้วรอรับการตอบกลับด้วย
 รายละเอียดดูหัวข้อต่อไป
- การเชื่อมต่อสู่โลก Internet หรือ Intranet ขบวนการรับส่งแบบ http protocol จะมีปริมาณข้อมูลในระดับหนึ่ง และมีการรอ Delay ตามความเหมาะสมด้วย รวมทั้งอาจจะเชื่อมต่อด้วย Client จากหลาย ๆ จุดเป็น สิบ,ร้อย,พัน โดยสรุปคือเมื่อสื่อสารไปแล้ว 1 ครั้ง ระยะเวลาไม่ควรน้อยกว่า 30 วินาที จึงจะสื่อสารครั้งต่อไปได้

แนวทางการใช้งาน Server Process

คือการสื่อสารที่ต้องรอให้มีการติดต่อมาจาก Client ก่อน แล้วจึงทำการตอบกลับไป ขอทำความเข้าใจจากตัวอย่าง สมมุติว่าได้เชื่อมต่อ Wifi แล้วในโหมด STA และมี IP เป็น 192.168.1.5 ถ้าเรียกจากคอมด้วย Browser โดยเรียกดังนี้ `http://192.168.1.5` ข้อมูลจะมาถึงตัว EM-SDWIFI ดังนี้

```
GET / HTTP/1.1
Host: 192.168.1.5
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: th,en-US;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

ทั้งนี้ตัว EM-SDWIFI จะดักข้อความไว้ แล้วจะส่งข้อมูลเท่าที่จำเป็นออกทาง RS485 (RS232-TTL) ดังนี้ (ถ้าอยากเห็นข้อมูลทั้งหมด ก็สามารถตั้งด้วยคำสั่ง :M1 ได้)

```
#GET / HTTP/1.1
```

คือเป็นการส่งข้อความออกมาทันที โดยที่ไม่ได้ส่งคำสั่งอะไรไปถามก่อน เครื่องหมาย # จะเพิ่มเติมเข้าไป เพื่อให้ผู้ใช้รับทราบว่า เป็นข้อมูลในบริบทของ Wifi ทั้งนี้ตัว Controller เมื่อได้รับข้อมูลแล้ว ก็สามารถตอบกลับไปได้ทันทีด้วยการส่งข้อความต่าง ๆ หลายบรรทัด เช่น

```
<html>
Echo from ESP-SDWIFI (Server) <br>
</html>
^Z
```

ข้อความเหล่านี้ก็คือภาษา HTML นั่นเอง หรือจะมองเป็น Text ทั่วไปก็ได้ การส่งข้อความทาง RS485 ผ่าน ESP-SDWIFI ผ่าน Wifi สู่โลก network นี้ ขอให้มึระยะเวลาห่างระหว่างบรรทัดอย่างน้อย 200mS และจบท้ายสุดด้วย ^Z (รหัส 0x1A) เพื่อให้ทราบว่าจบข้อความแล้ว อย่างไรก็ตาม ตัว ESP-SDWIFI จะตรวจสอบภายใน 1 วินาทีด้วย คือถ้าข้อความหยุดไป ก็จะถูกถือว่าจบข้อความด้วย แล้วจะแสดงดังนี้

#CLOSE

ก็คือจบการสื่อสาร 1 ครั้งอย่างสมบูรณ์ แล้วข้อความที่ตอบกลับไป ก็จะไปปรากฏบนหน้าเพจของ Browser นั้นเอง ... ทั้งหมดนี้คือขบวนการสื่อสารแบบ http protocol กรณีที่จะส่งข้อมูลใด ๆ จากทาง Browser ก็สามารถใส่ข้อความต่อท้ายได้ เช่น http://192.168.1.5/aabbccdd โดย ESP-SDWIFI ก็จะส่งต่อให้ดังนี้

#GET /aabbccdd HTTP/1.1

นอกจากนี้ยังมีอีกลักษณะหนึ่งของ http ด้วยการส่งแบบ POST เช่นการส่งจากฟอร์มของ html ด้วยการกดปุ่มในหน้า web ตัว ESP-SDWIFI ก็รองรับรูปแบบ POST ด้วย โดยที่ส่งแล้วจะส่งต่อข้อความมาให้ 2 บรรทัด เช่น

#POST / HTTP/1.1

#data=smicro&select=enter

ซึ่งก็สามารถส่งค่าตัวแปรต่าง ๆ ตามมาได้เช่นกัน แต่จะแยกเป็นอีกบรรทัดหนึ่ง ด้วย Server Process ทั้งหมดนี้ จะสามารถประยุกต์ใช้ได้หลากหลายแบบ แล้วแต่ว่าเป็นการเชื่อมต่ออะไร กับอะไร ถ้าเรียกมาจากคอม (Browser) ส่วนของ Controller ก็ทำตัวเป็น Web Server ได้ แต่ถ้าเรียกมาจาก Controller ก็สามารถมองทุกอย่างเป็น Text ธรรมดาโดยไม่ต้องสนใจภาษา HTML ก็ได้ เพียงแต่แปลงไปกับรูปแบบ http protocol นั้นเอง

แนวทางการใช้งาน Client Process

คือการสื่อสารที่จะส่งข้อมูลไปติดต่อกับ Server (หรือ Host) แล้วรอการตอบกลับมา ขอทำความเข้าใจจากตัวอย่าง สมมุติว่าได้เชื่อมต่อ Wifi แล้วในโหมด STA ทั้งนี้ได้ตั้ง Host ด้วยคำสั่ง :Hwww.smicrothai.com ไว้แล้ว (หรือจะตั้งด้วย host=www.smicrothai.com ในไฟล์ CONFIG.TXT ก็ได้) จากนั้นลองเรียกคำสั่งดังนี้

:S/_link/test.php?a=2

#Data From Server = 10

OK

รูปแบบที่ส่งไปคือแบบ GET ของ http protocol นั้นเอง โดยคำสั่ง :S จะไปทำการเพิ่มเติมข้อความอื่น ๆ ให้ข้อความ :S/_link/test.php?a=2 คือส่วนของข้อมูล หรือในแง่ Web Server ทั่วไปก็หมายถึงไพล์โคเรียย่อยไปด้วย ตามตัวอย่างนี้ Server จะนำตัวแปร a ที่มีค่า 2 ไปคูณด้วย 5 แล้วส่งกลับมาเป็นผลลัพธ์ 10 นั้นเอง ผู้ใช้สามารถทดลองเรียกเหมือนกับตัวอย่างนี้ได้จริง ๆ แต่กรณีถ้าส่งไปยัง Controller ด้วยกันเอง ก็สามารถกำหนดเป็นข้อความใด ๆ ได้อย่างอิสระ

ข้อความที่นำหน้าด้วย # ก็คือข้อความที่มาจาก Server นั้นเอง จริง ๆ ก็คือ Web Server แต่เขียนด้วยภาษา PHP เพื่อส่งเป็น text เท่านั้น (คือถ้าส่งกลับมาเป็น HTML ก็ได้ แต่ไม่เป็นประโยชน์อะไร เพราะตัว Controller ไม่ได้เอามาแสดงผลแบบหน้า web อยู่แล้ว) จากนั้นก็จบท้ายด้วย OK อีกที ขบวนการแบบ Client Process ทั้งหมดก็มีเพียงเท่านั้น แต่จริง ๆ การสื่อสารแบบ http จะมีข้อความหลายบรรทัดอยู่ แต่ขอละไว้ไม่อธิบาย ณ ตรงนี้ อีกประการหนึ่ง ค่า Delay ที่กำหนดจากคำสั่ง :D จะมีผลต่อขบวนการนี้ด้วย คือเป็นระยะเวลาที่ ESP-SDWIFI จะรอจนแน่ใจว่าไม่มีข้อความมาจาก Server แล้ว ซึ่งมีค่า Default เป็น 1000mS (1 วินาที) ทั้งหมดผู้ใช้อาจจะปรับให้เหมาะสมกับการใช้งานจริงได้ เช่น ถ้าเป็นการติดต่อกับ Server ภายในสถานที่ (Intranet) อาจจะต้องให้เร็วขึ้นเป็น 500mS ก็พอ หรือถ้าเป็น Server ต่างประเทศที่การสื่อสารอาจมีข้อจำกัด ก็อาจตั้งเป็น 2000mS (2 วินาที) เพื่อไว้ด้วย

ระบบ Auto-Echo และ Auto-Send

ESP-SDWIFI ได้เพิ่มเติมความสะดวกให้อีก คือแทนที่จะต้องจำลองขบวนการแบบ Server Process หรือ Client Process ผ่านคำสั่งทาง RS485 ซึ่งมีความยุ่งยากพอสมควร ผู้ใช้สามารถใช้ Auto-Echo หรือ Auto-Send เพื่อการทดสอบการทำงานได้อย่างง่ายดายด้วย รายละเอียดดังนี้

Auto-Echo คือใช้กับ Server Process (ดูหัวข้อ Server Process ด้วย) คือเมื่อเรียกจาก Browser ดังนี้
`http://192.168.1.5/ABOX-ESP-TEST` ตัว ESP-SDWIFI จะไม่ส่งข้อความต่อไปยัง RS485 เหมือนขบวนการ Server Process ปกติ แต่จะแจ้งทาง RS485 ด้วยข้อความดังนี้

```
#ABOX-ESP-TEST (GET หรือ POST) Auto-Echo
```

และจะจัดการตอบกลับไปให้เอง เป็นเหมือน Web Server ง่าย ๆ ในตัว โดยจะไปปรากฏที่ Browser ดังนี้

```
ABOX-ESP-TEST (192.168.1.5)
```

```
Count=0001
```

```
3B-Port [111] [ปุ่ม Enter]
```

ซึ่งผู้ใช้อีกก็สามารถรับทราบได้ว่าการสื่อสารสมบูรณ์หรือยัง รวมทั้งยังทดลองเปลี่ยนแปลง I/O ที่อยู่บนหัว 3B-Port (5Pin) ได้จากหน้า Web ด้วย ค่าตัวแปร Count ก็จะเป็นตัวเลขไปเรื่อย ๆ คือจำนวนครั้งของการสื่อสารนั่นเอง นอกจากนี้ ยังเรียกเป็นคำว่า /ABOX-ESP-CHK ได้ด้วย ซึ่งจะให้ผลในทำนองเดียวกัน แต่แทนที่จะเป็นหน้า Web Page ก็จะเป็นข้อความ Text ง่าย ๆ ซึ่งกรณีนี้ที่ทดลองจากด้าน Controller ก็จะทำให้สะดวกขึ้น ข้อความจะตอบกลับมาดังนี้

```
ABOX-ESP-CHK Count=0001 (192.168.1.5)
```

สำหรับ Auto-Send คือใช้กับ Client Process (ดูหัวข้อ Client Process ด้วย) คือแทนที่จะต้องส่งคำสั่งผ่านทาง RS485 ผู้ใช้ก็เพียงแค่กดปุ่ม (สีดำ) บนบอร์ด ตัว ESP-SDWIFI ก็จะจำลองสถานะ Client Process ไปยังเป้าหมายให้ทันที และสามารถดูผลจาก LED ที่กระพริบด้วย ... ตามตัวอย่างที่ผ่านมา กำหนดให้มี ESP-SDWIFI อีกตัวหนึ่ง เชื่อมต่อโดยมี IP เป็น 192.168.1.6 และทำการตั้ง Host เป็น 192.168.1.5 คืออีกตัวหนึ่ง จากนั้นให้กดปุ่มที่เครื่อง 192.168.1.6 จะมีผลดังนี้ (แสดงข้อความทาง RS485)

```
(ที่เครื่อง 192.168.1.6) AUTO-SEND
```

```
#ABOX-ESP-CHK Count=0001 (192.168.1.5)
```

```
OK
```

LED สีฟ้าจะดับชั่วคราว จากนั้น ...

(1) กระพริบ 2 ครั้ง แล้วสว่าง หมายถึงสื่อสารได้สมบูรณ์

(2) กลับมาสว่าง หมายถึงไม่สามารถสื่อสารได้ (ทาง RS485 แสดง ER)

```
(ที่เครื่อง 192.168.1.5) #ABOX-ESP-CHK (GET) Auto-Echo
```

LED สีฟ้าจะดับชั่วคราว จากนั้นก็สว่าง

ด้วยระบบเหล่านี้ ผู้ใช้จะทดสอบพื้นฐานต่าง ๆ ในการเชื่อมต่อได้สะดวกยิ่งขึ้น และดูผลจาก LED ก็รับทราบได้ ยิ่งกว่านั้น ถ้าต้องการทดสอบเสถียรภาพและความต่อเนื่องในการทำงาน ยังสามารถตั้งเวลาด้วยคำสั่ง :T ได้ด้วย คือตั้งเวลาเป็นวินาที เพื่อให้เสมือนมีการกดปุ่มได้เอง โดยอาจจะลองตั้งที่ 010 วินาทีก็ได้ และถ้าต้องการหยุดก็ให้ตั้งเป็น 000 ตามเดิม

ชุดคำสั่งเกี่ยวกับ Wifi

รูปแบบคำสั่ง	การใช้งาน
<p>(R) :W</p> <p>(T) STA-CONNECTED smicro 12345678 192.168.4.2 192.168.4.1 255.255.255.0 </p> <p>(T) AP-STARTED smicro 12345678 192.168.4.1 </p> <p>(T) OFF </p>	<p>Wifi Status</p> <p>แสดงสถานะของ Wifi ตามที่เป็นจริง (แสดงตัวอย่างข้อความ 3 กรณี)</p> <p>คือใช้งานโหมด STA ตามด้วยข้อมูล name,pass,ipad,gate และ subn</p> <p>คือใช้งานโหมด AP ตามด้วยข้อมูล name,pass และ ipad (โหมด AP จะไม่แสดง gate และ subn)</p> <p>คือไม่ใช้งาน Wifi (คือไม่กำหนดอะไรในไฟล์ CONFIG.TXT หรือกำหนดแต่ไม่สามารถ Connect ได้)</p>
<p>(R) :H[192.168.4.1]</p> <p>(T) 192.168.4.1[OK]</p>	<p>Host Read/Write</p> <p>กำหนดชื่อ Host ที่จะเชื่อมต่อด้วยคำสั่ง :S อื่นๆ ตามตัวอย่างจะกำหนดเป็นหมายเลข IP ก็ได้ หรือจะเป็นชื่อเช่น www.smicrothai.com ก็ได้ การกำหนด Host อาจกระทำตั้งแต่ไฟล์ CONFIG.TXT ก็ได้ และสามารถเปลี่ยนด้วยคำสั่ง :H อื่นๆ ได้ คำสั่งนี้ถ้าใส่ชื่อต่อท้ายก็จะเป็นการกำหนดค่า (Write) และถ้าใส่เฉพาะ :H ก็คือดูค่า (Read)</p>
<p>(R) :S/_link/test.php?a=2</p> <p>(T) #Data From Server = 10</p> <p>(T) OK</p>	<p>Send to Host</p> <p>คือการส่งข้อมูลไปยัง Host (Server) ในรูปแบบ http protocol และรอทาง Host ตอบกลับมา ภายในค่า Delay ที่กำหนดจากคำสั่ง :D (Default=1000mS) จากนั้นจบท้ายด้วย OK ถ้ากรณีเชื่อมต่อกับ Host ไม่ได้ก็จะแสดง ER ให้ทราบ หรือถ้าไม่ได้กำหนดชื่อ Host ไว้ก็จะแสดง NH (No-Host) (ดูหัวข้อ Client Process)</p>
<p>(R) :M[0,1]</p> <p>(T) :0,1[OK]</p>	<p>Monitor ON/OFF</p> <p>สำหรับกำหนดว่าให้แสดงข้อมูลทั้งหมดที่รับมาจาก Wifi หรือไม่ โดยปกติจะมีค่าเป็น 0 คือไม่แสดง แต่ถ้ากำหนดเป็น 1 คือให้แสดงข้อมูลทั้งหมด</p>
<p>(R) :D[1000]</p> <p>(T) 1000[OK]</p>	<p>Delay Read/Write (mS)</p> <p>กำหนดค่า Delay ที่เกี่ยวข้องกับคำสั่ง :S อื่นๆ (ดูหัวข้อ Client Process)</p>
<p>(R) :T[000]</p> <p>(T) 000[OK]</p>	<p>Test Loop Read/Write (Sec)</p> <p>สำหรับกำหนดค่าวินาที เพื่อเสมือนให้กดปุ่ม Test แบบอัตโนมัติได้เอง (ดูหัวข้อระบบ Auto-Echo และ Auto-Send)</p>