

เรื่อง MCS-51 I/O - Arduino Input/Output

จัดทำโดย

นางสาวสุนันท์ เป็ดโปง
รหัสนักศึกษา B6023973
สาขาวิชา วิศวกรรมคอมพิวเตอร์

เสนอ

อาจารย์ นายวิชัย ศรีสุรักษ์
นายอำนวย ที่จันทิก
นายศพงษ์ ไชยฤกษ์
นายทองยศ ศรีเพ็ง

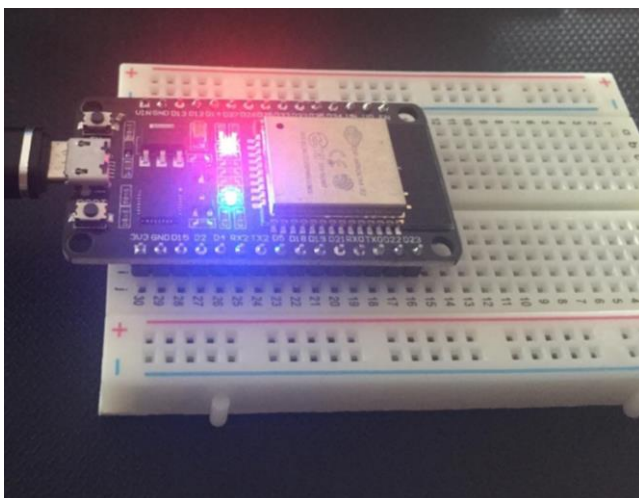
รายงานนี้เป็นส่วนหนึ่งของการเรียนวิชา ไมโครโปรเซสเซอร์
ภาคเรียนที่ 1 ปีการศึกษา 2562
มหาวิทยาลัยเทคโนโลยีสุรนารี

Part-A: การทดลอง เฉพาะข้อที่ทดลองในห้อง

การทดลองข้อ 1 ESP32

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(100);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(100);  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(100);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(100);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(100);  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(100);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```



การทดลองข้อ 1 MCS51

ORG 8000H	ORG 8000H
CLR EA	CLR EA
LOOP: CLR P3.2	LOOP: CLR P3.2
CALL Delay	CALL Delay
SETB P3.2	SETB P3.2
CALL Delay	CALL Delay
CLR P3.2	CLR P3.2
CALL Delay	CALL Delay
SETB P3.2	SETB P3.2
CALL Delay	CALL Delay
CALL Delay	CALL Delay
CALL Delay	CALL Delay
JMP LOOP	JMP LOOP
Delay: PUSH B	Delay: PUSH B
PUSH Acc	PUSH Acc
MOV A, #200	MOV A, #200
_DLY01: MOV B, #200	_DLY01: MOV B, #200
_DLY00: DJNZ B, _DLY00	_DLY00: DJNZ B, _DLY00
DJNZ Acc, _DLY01	DJNZ Acc, _DLY01
POP Acc	POP Acc
POP B	POP B
RET	RET
END	END



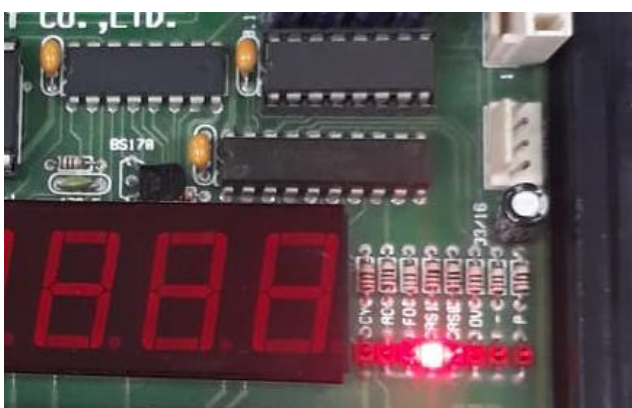
การทดลองข้อ 2 ESP32

<pre> ORG 8000H ; 1 MOV DPTR,#0E000H LOOP: MOV A,#000H CALL Send_Port MOV A,#0FFH CALL Send_Port JMP LOOP Send_Port: MOVX @DPTR,A MOV P1,A Delay: PUSH B PUSH Acc MOV A,#10 _DLY00: MOV B,#9 _DLY01: DJNZ B,_DLY01 DJNZ Acc,_DLY00 POP Acc POP B RET END </pre>	<pre> ORG 8000H ; 1 MOV DPTR,#0E000H LOOP: MOV A,#000H CALL Send_Port MOV A,#0FFH CALL Send_Port JMP LOOP Send_Port: MOVX @DPTR,A MOV P1,A Delay: PUSH B PUSH Acc MOV A,#10 _DLY00: MOV B,#9 _DLY01: DJNZ B,_DLY01 DJNZ Acc,_DLY00 POP Acc POP B RET END </pre>
--	--

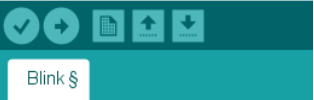
การทดลองข้อ 3 ESP32

<pre> ORG 8000H LOOP: CALL LED_R2L JMP LOOP LED_R2L: MOV A,#10000000B CALL SendData MOV A,#01000000B CALL SendData MOV A,#00100000B CALL SendData MOV A,#00010000B CALL SendData MOV A,#00001000B CALL SendData MOV A,#00000100B CALL SendData MOV A,#00000010B CALL SendData MOV A,#00000001B CALL SendData RET SendData: MOV DPTR,#0E001H MOVX @DPTR,A MOV A,#6 MOV DPTR,#0E002H MOVX @DPTR,A Delay: CLR A MOV B,A _DLY00: DJNZ Acc,_DLY00 DJNZ B,_DLY00 RET END </pre>	<pre> ORG 8000H LOOP: CALL LED_R2L JMP LOOP LED_R2L: MOV A,#10000000B CALL SendData MOV A,#01000000B CALL SendData MOV A,#00100000B CALL SendData MOV A,#00010000B CALL SendData MOV A,#00001000B CALL SendData MOV A,#00000100B CALL SendData MOV A,#00000010B CALL SendData MOV A,#00000001B CALL SendData RET SendData: MOV DPTR,#0E001H MOVX @DPTR,A MOV A,#6 MOV DPTR,#0E002H MOVX @DPTR,A </pre>
---	---

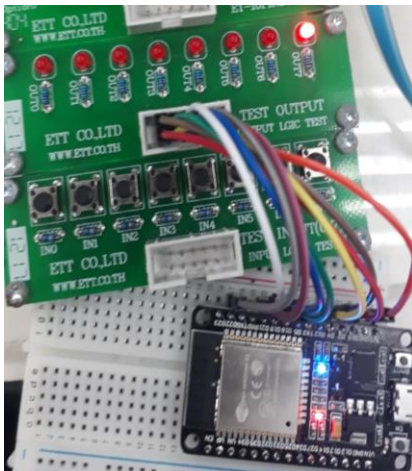
	<pre>Delay: CLR A MOV B,A _DLY00: DJNZ Acc,_DLY00 DJNZ B,_DLY00 RET END</pre>
--	---



การทดลองข้อ 3 MCS51

 <pre> int LED1 = 22 ; int LED2 = 21 ; int LED3 =19 ; int LED4 = 18 ; int LED5 = 5 ; int LED6 = 4 ; int LED7 = 2 ; int LED8 = 15 ; void setup() { pinMode (LED1,OUTPUT); pinMode (LED2,OUTPUT); pinMode (LED3,OUTPUT); pinMode (LED4,OUTPUT); pinMode (LED5,OUTPUT); pinMode (LED6,OUTPUT); pinMode (LED7,OUTPUT); pinMode (LED8,OUTPUT); } void loop() { digitalWrite (LED8,LOW); delay (300) ; digitalWrite (LED8,HIGH) ; digitalWrite (LED7,LOW); delay (300) ; digitalWrite (LED7,HIGH) ; digitalWrite (LED6,LOW); delay (300) ; digitalWrite (LED6,HIGH) ; digitalWrite (LED5,LOW); delay (300) ; digitalWrite (LED5,HIGH) ; digitalWrite (LED4,LOW); delay (300) ; digitalWrite (LED4,HIGH) ; digitalWrite (LED3,LOW); delay (300) ; digitalWrite (LED3,HIGH) ; digitalWrite (LED2,LOW); delay (300) ; digitalWrite (LED2,HIGH) ; digitalWrite (LED1,LOW); delay (300) ; digitalWrite (LED1,HIGH) ; } </pre>	<pre> int LED1 = 22 ; int LED2 = 21 ; int LED3 =19 ; int LED4 = 18 ; int LED5 = 5 ; int LED6 = 4 ; int LED7 = 2 ; int LED8 = 15 ; void setup() { pinMode (LED1,OUTPUT); pinMode (LED2,OUTPUT); pinMode (LED3,OUTPUT); pinMode (LED4,OUTPUT); pinMode (LED5,OUTPUT); pinMode (LED6,OUTPUT); pinMode (LED7,OUTPUT); pinMode (LED8,OUTPUT); } void loop() { digitalWrite (LED8,LOW); delay (300) ; digitalWrite (LED8,HIGH) ; digitalWrite (LED7,LOW); </pre>
---	--

	<pre> delay (300) ; digitalWrite (LED7,HIGH) ; digitalWrite (LED6,LOW); delay (300) ; digitalWrite (LED6,HIGH) ; digitalWrite (LED5,LOW); delay (300) ; digitalWrite (LED5,HIGH) ; digitalWrite (LED4,LOW); delay (300) ; digitalWrite (LED4,HIGH) ; digitalWrite (LED3,LOW); delay (300) ; digitalWrite (LED3,HIGH) ; digitalWrite (LED2,LOW); delay (300) ; digitalWrite (LED2,HIGH) ; digitalWrite (LED1,LOW); delay (300) ; digitalWrite (LED1,HIGH) ; } </pre>
--	---



การทดลองข้อ 4 ESP32

```

const byte SEVEN_SEG[7] = {2, 5, 18, 19,21,22,23};
byte nCount = 0;
const byte DIGIT_7SEG[] = {
  B00111111, //0
  B00000110, //1
  B01011011, //2
  B01001111, //3
  B01100110, //4
  B01101101, //5
  B01111101, //6
  B00000111, //7
  B01111111, //8
  B01101111, //9
  B11110111,
  B01111100,
  B10111001,
  B11011110,
  B11111001,
  B01110001
};

void setup() {
  for (int i = 0; i < 8; i++) {
    pinMode( SEVEN_SEG[i], OUTPUT );
    digitalWrite( SEVEN_SEG[i], HIGH );
  }
}

void loop() {
  displayDigit( nCount );
  nCount = (nCount + 1) % 16;
  delay(1000);
}

void displayDigit( byte value ) {
  if ( 0 <= value && value < 16 ) {
    byte value = DIGIT_7SEG[ nCount ];
    for (int i = 0; i < 8; i++) {
      digitalWrite( SEVEN_SEG[i], !(value & 1) );
      value >>= 1;
    }
  }
}

```

```

const byte SEVEN_SEG[7] = {2, 5, 18,
19,21,22,23};
byte nCount = 0;
const byte DIGIT_7SEG[] = {
  B00111111, //0
  B00000110, //1
  B01011011, //2
  B01001111, //3
  B01100110, //4
  B01101101, //5
  B01111101, //6
  B00000111, //7
  B01111111, //8
  B01101111, //9
  B11110111,
  B01111100,
  B10111001,
  B11011110,
  B11111001,
  B01110001

```

};

void setup() {

for (int i = 0; i < 8; i++) {

pinMode(SEVEN_SEG[i], OUTPUT);

digitalWrite(SEVEN_SEG[i], HIGH);

	<pre>} } void loop() { displayDigit(nCount); nCount = (nCount + 1) % 16; delay(1000); } void displayDigit(byte value) { if (0 <= value && value < 16) { byte value = DIGIT_7SEG[nCount]; for (int i = 0; i < 8; i++) { digitalWrite(SEVEN_SEG[i], !(value & 1)); value >>= 1; } } }</pre>
--	---



การทดลองข้อ 4 MCS51

1	ORG 8000H	ORG 8000H
2		
3	MAIN: MOV DPTR,#0E060H	MAIN: MOV DPTR,#0E060H
4	MOV A,#03FH	MOV A,#03FH
5	MOVX @DPTR,A	MOVX @DPTR,A
6	CALL DELAY	CALL DELAY
7	CALL DELAY	CALL DELAY
8		
9	MOV A,#06H	
10	MOVX @DPTR,A	
11	CALL DELAY	
12	CALL DELAY	
13		
14	MOV A,#5BH	
15	MOVX @DPTR,A	
16	CALL DELAY	
17	CALL DELAY	
18		
19	MOV A,#4FH	
20	MOVX @DPTR,A	
21	CALL DELAY	
22	CALL DELAY	
23		
24	MOV A,#66H	
25	MOVX @DPTR,A	
26	CALL DELAY	
27	CALL DELAY	
28		
29	MOV A,#6DH	
30	MOVX @DPTR,A	
31	CALL DELAY	
32	CALL DELAY	
33		
34	MOV A,#7DH	
35	MOVX @DPTR,A	
36	CALL DELAY	
37	CALL DELAY	
38		
39	MOV A,#07H	
40	MOVX @DPTR,A	
41	CALL DELAY	
42	CALL DELAY	
43		
44	MOV A,#7FH	
45	MOVX @DPTR,A	
46	CALL DELAY	
47	CALL DELAY	
48		
49	MOV A,#6FH	
50	MOVX @DPTR,A	
51	CALL DELAY	
52	CALL DELAY	
53		
54	MOV A,#77H	
55	MOVX @DPTR,A	
56	CALL DELAY	
57	CALL DELAY	
58		
59	MOV A,#7CH	
60	MOVX @DPTR,A	
61	CALL DELAY	
62	CALL DELAY	
63		
64	MOV A,#39H	
65	MOVX @DPTR,A	
66	CALL DELAY	
67	CALL DELAY	
68		
69	MOV A,#5EH	
70	MOVX @DPTR,A	
71	CALL DELAY	
72	CALL DELAY	
73		
74	MOV A,#79H	
75	MOVX @DPTR,A	
76	CALL DELAY	
77	CALL DELAY	
78		
79	MOV A,#71H	
80	MOVX @DPTR,A	
81	CALL DELAY	
82	CALL DELAY	
83		
84	JMP MAIN	
85		
86	DELAY: MOV R1,#250	
87		
88	_DLY01: MOV R2,#250	
89	_DLY00: DJNZ R2,_DLY00	
90	DJNZ R1,_DLY01	
91	RET	
		MOV A,#06H
		MOVX @DPTR,A
		CALL DELAY
		CALL DELAY
		MOV A,#5BH
		MOVX @DPTR,A
		CALL DELAY
		CALL DELAY
		MOV A,#4FH
		MOVX @DPTR,A
		CALL DELAY
		CALL DELAY
		MOV A,#66H
		MOVX @DPTR,A
		CALL DELAY
		CALL DELAY

	<div>MOV A,#6DH</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div> <div>MOV A,#7DH</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div> <div>MOV A,#07H</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div> <div>MOV A,#7FH</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div> <div>MOV A,#6FH</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div> <div>MOV A,#77H</div> <div>MOVX @DPTR,A</div> <div>CALL DELAY</div> <div>CALL DELAY</div>
--	---

	<pre> MOV A,#7CH MOVX @DPTR,A CALL DELAY CALL DELAY MOV A,#39H MOVX @DPTR,A CALL DELAY CALL DELAY MOV A,#5EH MOVX @DPTR,A CALL DELAY CALL DELAY MOV A,#79H MOVX @DPTR,A CALL DELAY CALL DELAY MOV A,#71H MOVX @DPTR,A CALL DELAY CALL DELAY JMP MAIN DELAY: MOV R1,#250 _DLY01: MOV R2,#250 </pre>
--	---

	<pre>_DLY00: DJNZ R2,_DLY00 DJNZ R1,_DLY01 RET</pre>
--	--



```

ORG      8000H
        jmp      0100H
ORG      1000H
        MOV      SP, #2FH
        MOV      P1, #00001111B
LOOP:    JB       P1.0, JOB0
        JNB      P1.0, JOB_DEF
        JMP      LOOP

JOB0:    MOV      P3, #00000000B
        JMP      LOOP

JOB_DEF MOV      P3, 01000000B
        JMP      LOOP

        END

```

[illegible]

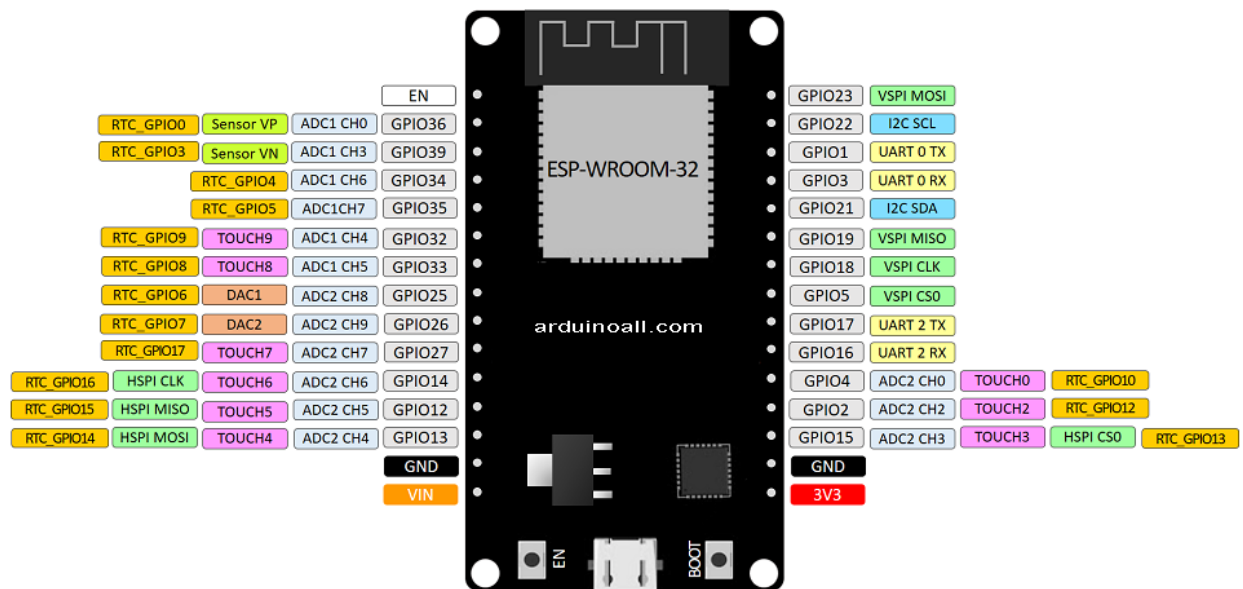
การทดลองข้อ 4

1	ORG	8000H	ORG	8000H		
2	MOV	P1,#00001111B	MOV	P1,#00001111B		
3	MOV	P3,#00000000B	MOV	P3,#00000000B		
4						
5	LOOP:	MOV	A,P1	LOOP: MOV	A,P1	
6		XRL	A,#00001110B		XRL	A,#00001110B
7		JZ	JOB		JZ	JOB
8		SETB	P3.2		SETB	P3.2
9		JMP	LOOP		JMP	LOOP
10						
11	JOB:	CLR	P3.2	JOB: CLR	P3.2	
12		RET			RET	
13						
14		END			END	

Part-C: ข้อมูลเพิ่มเติม

ข้อที่ 1. PinOut

ESP32 DEVKIT V1 – DOIT version with 30 GPIOs



ตำแหน่งขาที่ใช้ต่อของบอร์ด ESP 32 ใช้ในการต่อวงจรที่ระบุเอาไว้ต้อง

ข้อที่2 Minimum connection

การใช้งานบลูทูธบน ESP32

ตัว ESP32 นั้นใช้งานบลูทูธเวอร์ชัน 4.2 ซึ่งเป็นเวอร์ชันที่รองรับ BLE อย่างเต็มรูปแบบ สำหรับ BLE จะค่อนข้างยืดหยุ่น และใช้งานง่ายในระดับหนึ่ง คือเมื่อมีการเชื่อมต่อแล้ว จะต้องมีการกำหนดบริการที่จะขอข้อมูล หรือส่งข้อมูล ซึ่งจะเรียกว่า Service ซึ่งแต่ละ Service ก็จะมีสิ่งที่เรียกว่า Characteristic เป็นตัวลูกที่เราจะเข้าไปรับ - ส่งข้อมูลอีกที

ใช้งาน BLE บน ESP32 ผ่านไลบรารี easyBLE

- ให้ติดตั้งไลบรารี easyBLE ก่อน

- จากนั้นให้เปิดโปรแกรม Arduino เลือกไปที่เมนู File > Examples > easyBLE > basic

- จากนั้นเลือกบอร์ด เลือกพอร์ตให้ถูกต้อง แล้วอัปโหลดโปรแกรมลงบอร์ด ESP32

- หลังจากอัปโหลดแล้ว ให้เปิด Serial Monitor มารอไว้ก่อน (ปรับ Band เป็น 115200)

โทรศัพท์มือถือแอนดรอยด์ เปิดบลูทูธ แล้วเข้าไปดาวน์โหลดแอปฯ BLE Scanner จาก Google Play

จากนั้นเปิดแอปฯ ขึ้นมา จะพบว่ามีอุปกรณ์ BLE Basic แสดงขึ้นมาแล้ว ให้กดปุ่ม Connect

จากนั้นให้กดที่ปุ่มลูกศร จะสังเกตเห็นว่าแอปฯ จะแสดงค่า UUID แบบเต็มของ Service

และ Characteristic มาให้ ซึ่งตามจริงแล้วเมื่อนำไปใช้สร้างแอปฯ จะต้องระบุ UUID แบบเต็ม แต่เนื่องจากไลบรารี easyBLE ต้องการให้พิมพ์โค้ดในโปรแกรม Arduino ให้สั้น จึงไม่ให้กรอก UUID แบบเต็ม แต่ให้

กรอก UUID แบบสั้นแทน ซึ่งค่าที่กรอกก็นำไปใช้เป็นส่วนหนึ่งของ UUID เช่นเดียวกัน

ให้ทดลองกดปุ่มกลมรูปตัว R จะเป็นการทดลองส่งคำสั่ง Read จะสังเกตว่าหลังกด จะแสดงคำว่า OK ! ในส่วนของ Value

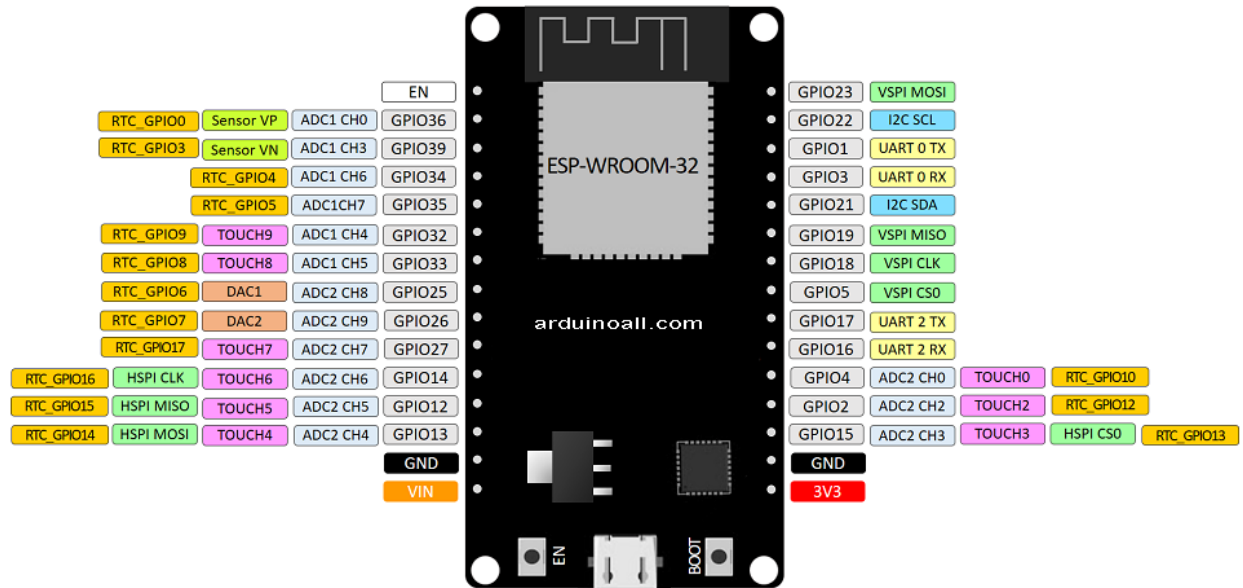
เมื่อกลับมาที่ Serial Monitor จะพบข้อความแสดงขึ้นแจ้งรับคำสั่ง Read เข้ามา ซึ่งจะระบุค่าของ Service UUID และ Characteristic UUID มาให้ด้วย ทำให้ง่ายต่อการนำไปใช้งาน โดยสามารถนำค่าของ Service UUID และ Characteristic UUID ไป if แล้วส่งข้อมูลกลับไป

จะเห็นได้ว่า สามารถรับ - ส่งข้อมูลผ่านบลูทูธด้วย ESP32 ได้แล้ว

ข้อที่ 3 Pin.Funtion

ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



Pin Category	Pin Name	Details
Power	Micro-USB, 3.3V, 5V, GND	<p>Micro-USB: ESP32 can be powered through USB port</p> <p>5V: Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board.</p> <p>3.3V: Regulated 3.3V can be supplied to this pin to power the board.</p> <p>GND: Ground pins.</p>
Enable	En	The pin and the button resets the microcontroller.
Analog Pins	ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC
DAC pins	DAC1 and DAC2	Used for Digital to analog Conversion
Input/Output Pins	GPIO0 to GPIO39	Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	T0 to T9	These 10 pins can be used a touch pins normally used for capacitive pads
RTC GPIO pins	RTCIO0 to RTCIO17	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.
Serial	Rx, Tx	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be use to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM though software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.

HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	AREF	To provide reference voltage for input voltage.

ข้อที่4 Power on Reset

บอร์ดจะทำการ Power on Reset ในแต่ละบิตจนกว่าจะทำการปิดหรือไม่ไฟเข้าใช้

- ESP-WROOM-32 จาก Espressif ซึ่งเป็น WiFi/BLE SoC (System On Chip)
- Breadboard Friendly มีขนาดกว้าง 0.9" วางบน breadboard จะเหลือข้างละ 1 ช่อง
- ใช้ USB2Serial ตระกูล FTDI ชิปเพื่อการโหลดโปรแกรมแบบอัตโนมัติ ความเร็วสูงสุดถึง 921000
- JST 2mm Connector สำหรับเสียบแบตเตอรี่
- มีวงจรชาร์จ Lithium Ion และ Lithium Polymer (1 cell) พร้อมทั้งไฟแสดงสถานะ
- มีวงจร PTC Fuse ตัดกระแสไฟเกินที่ 500mA
- 3.3V 600mA On-board Voltage Regulator
- Push Button Switch ที่ขา IO0 และ EN (Reset)
- Crystal 32.768KHz เพื่อใช้เลี้ยงวงจร RTC

เหมาะสำหรับงาน พัฒนาด้านแบบ อุปกรณ์รูปแบบ Portable และ Wearable