บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การศึกษาค้นคว้าของเอกสารที่เกี่ยวข้องเพื่อให้การสร้างและหาประสิทธิภาพของเครื่องให้อาหารสัตว์เลี้ยงอัตโนมัติผู้รายงานได้ศึกษาเอกสารที่เกี่ยวข้อง โดยเสนอตามลำดับดังนี้

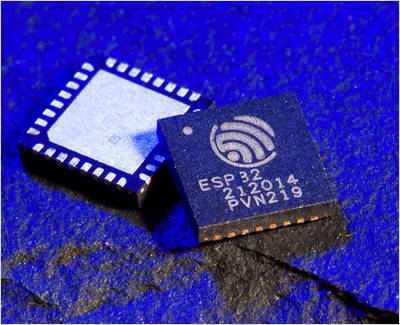
2.1 ESP32

2.2 MGR 996R Servo Motor

2.3เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04

2.4โมดูล RTC (Real Time Clock) DS3231

**2.1 ESP32**



รูปที่2.1.1 ชิปไอซีESP32 ที่มา ioxhop.com

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth 4.2 BLE ในตัว ผลิตโดยบริษัท Espressif จากประเทศจีน โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

-ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz

-มีแรมในตัว 512KB

-รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB

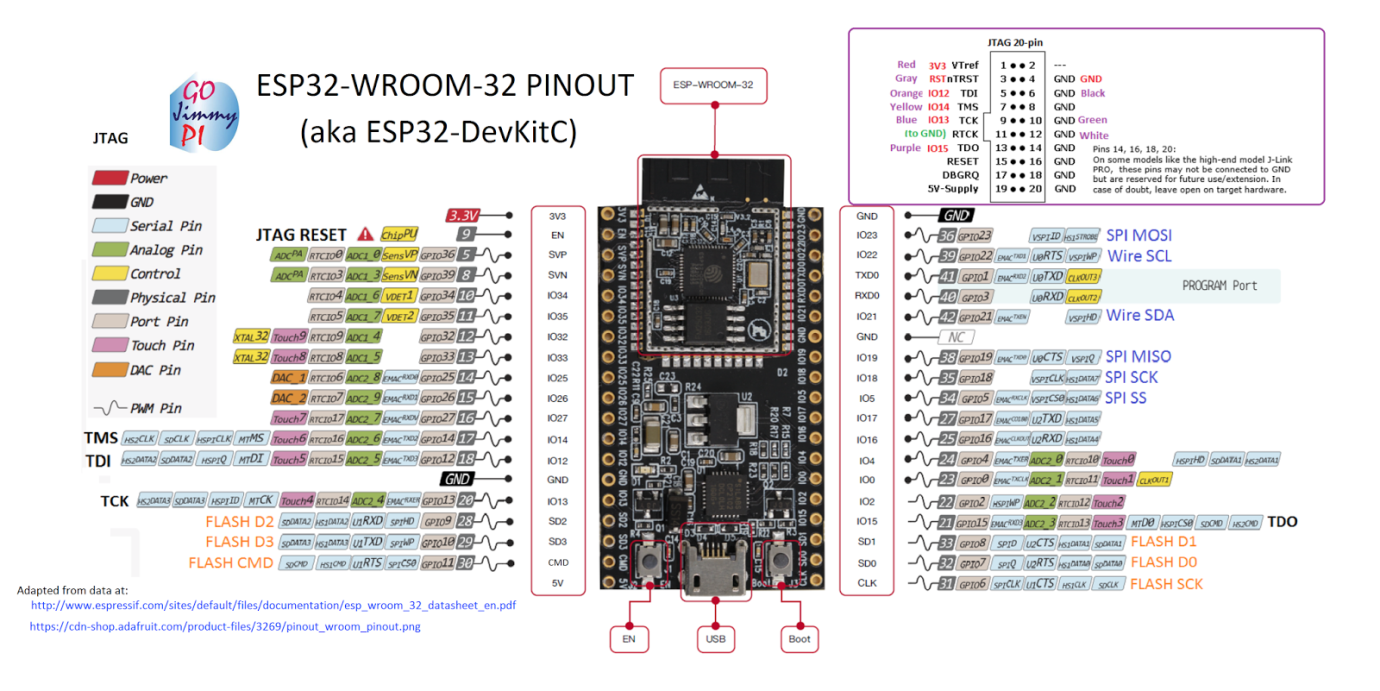
-มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ

Wi-Fi direct

-มีบลูทูธในตัวรองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE

-ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V

-ทำงานได้ที่อุณหภูมิ -40◦C ถึง 125◦C



รูปที่ 2.1.2 ตำแหน่งขา การใช้งานต่าง ๆ ที่มา allnewstep.com

ขาใช้งานต่างๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่างๆ ดังนี้

-มี GPIO จำนวน 32 ช่อง

-รองรับ UART จำนวน 3 ช่อง

-รองรับ SPI จำนวน 3 ช่อง

-รองรับ I2C จำนวน 2 ช่อง

-รองรับ ADC จำนวน 12 ช่อง

-รองรับ DAC จำนวน 2 ช่อง

-รองรับ I2S จำนวน 2 ช่อง

-รองรับ PWM / Timer ทุกช่อง

-รองรับการเชื่อมต่อกับ SD-Card

การเขียนโปรแกรม ESP32

ESP32 สามารถเขียนโปรแกรมได้ โดยใช้ Arduino IDE รูปแบบเหมือนใช้งาน Arduino ทั่ว ๆ ไป นอกจากนี้ ยังสามารถใช้เครื่องมือเขียนโปรแกรมตัวอื่นได้อีกหลายแบบ เช่น Epresssif IDE , Micropython , JavaScript , LUA และอื่น ๆ

การอัพโหลด ESP-32

กดปุ่ม boot ค้างไว้ แล้วกดปุ่ม upload รอจนขึ้นคำว่า Connection แล้วปล่อยปุ่ม boot ได้

กรณี อัพโหลด esp-32 ไม่เข้า การอัพโหลดดังนี้

-ถอดบอร์ด ESP32

-กดปุ่ม boot ค้างไว้

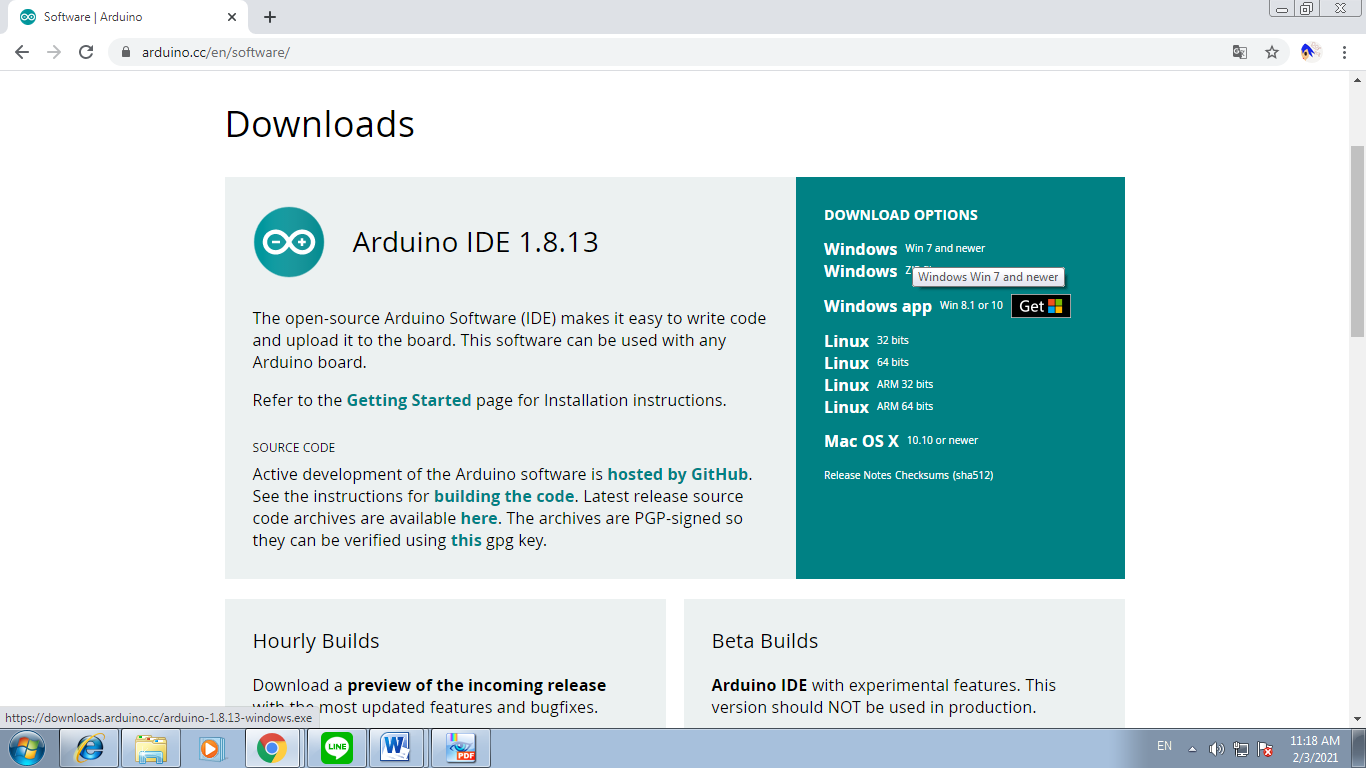
-เสียบบอร์ด ESP-32 แล้วกดปุ่ม upload ใน Arduino IDE

-เมื่อขึ้นคำว่า Connecting ปล่อยปุ่ม boot ได้ แล้วรอจนอัพโหลดสำเร็จ

ขั้นตอนการปฏิบัติ

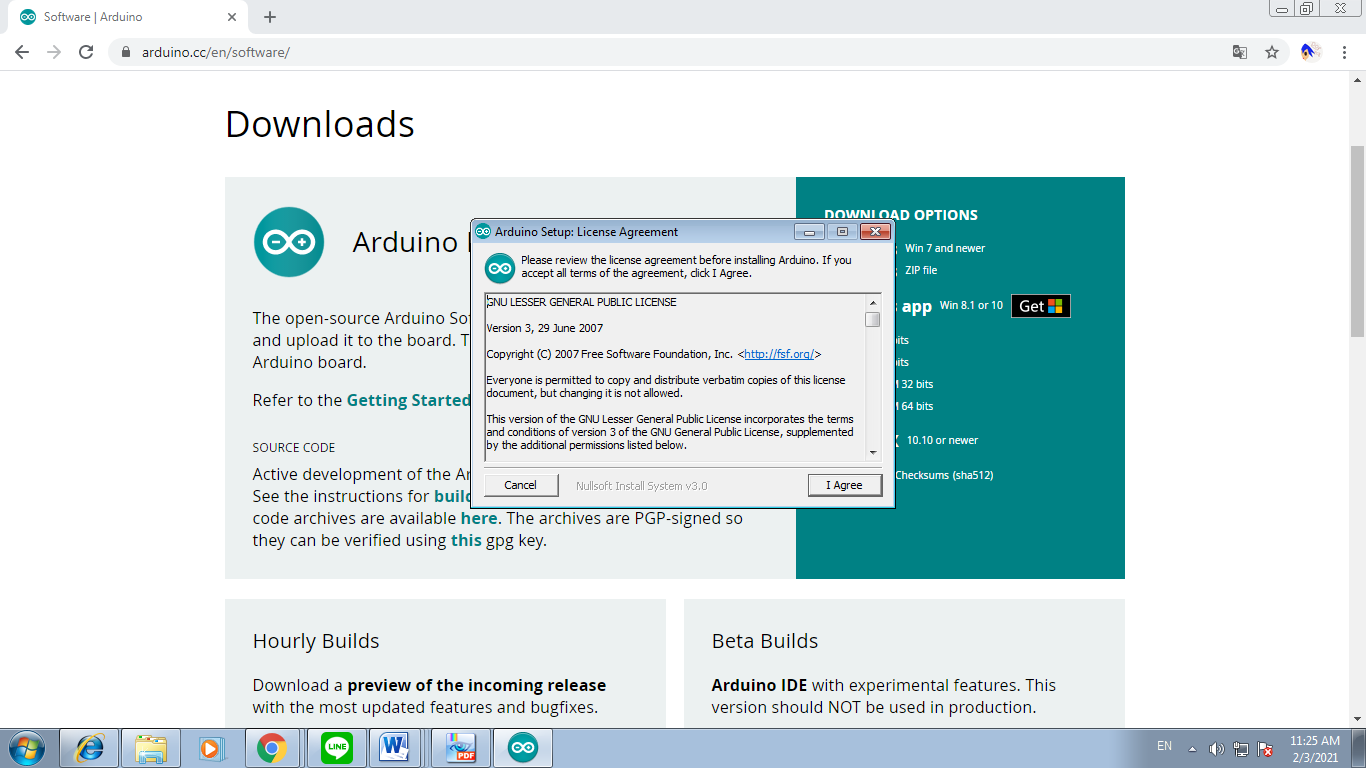
1.ดาวน์โหลด ArduinoIDE จาก https://www.arduino.cc/en/software/

2.เลือกระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่ท่านจะใช้ในการติดตั้งโปรแกรม ArduinoIDE



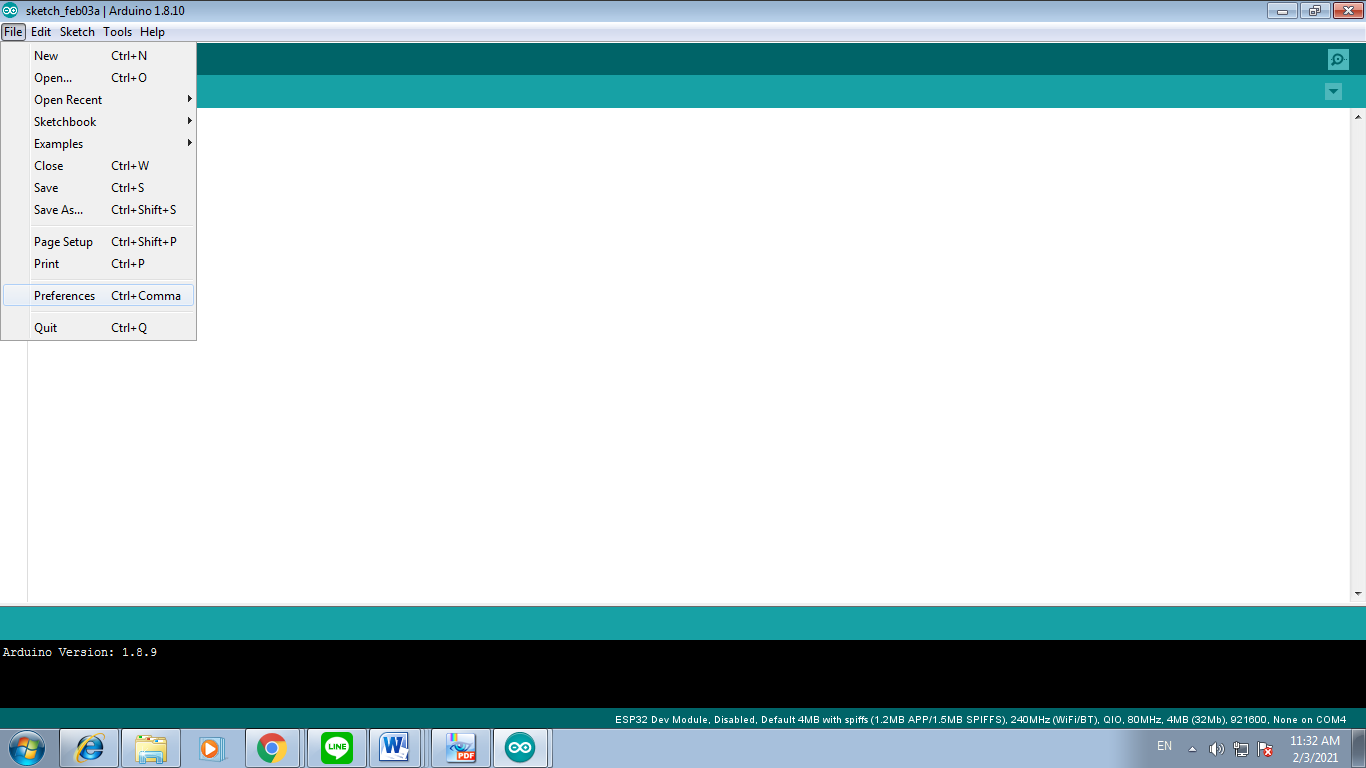
รูปที่ 2.1.3

3.ทำการติดตั้งโปรแกรม



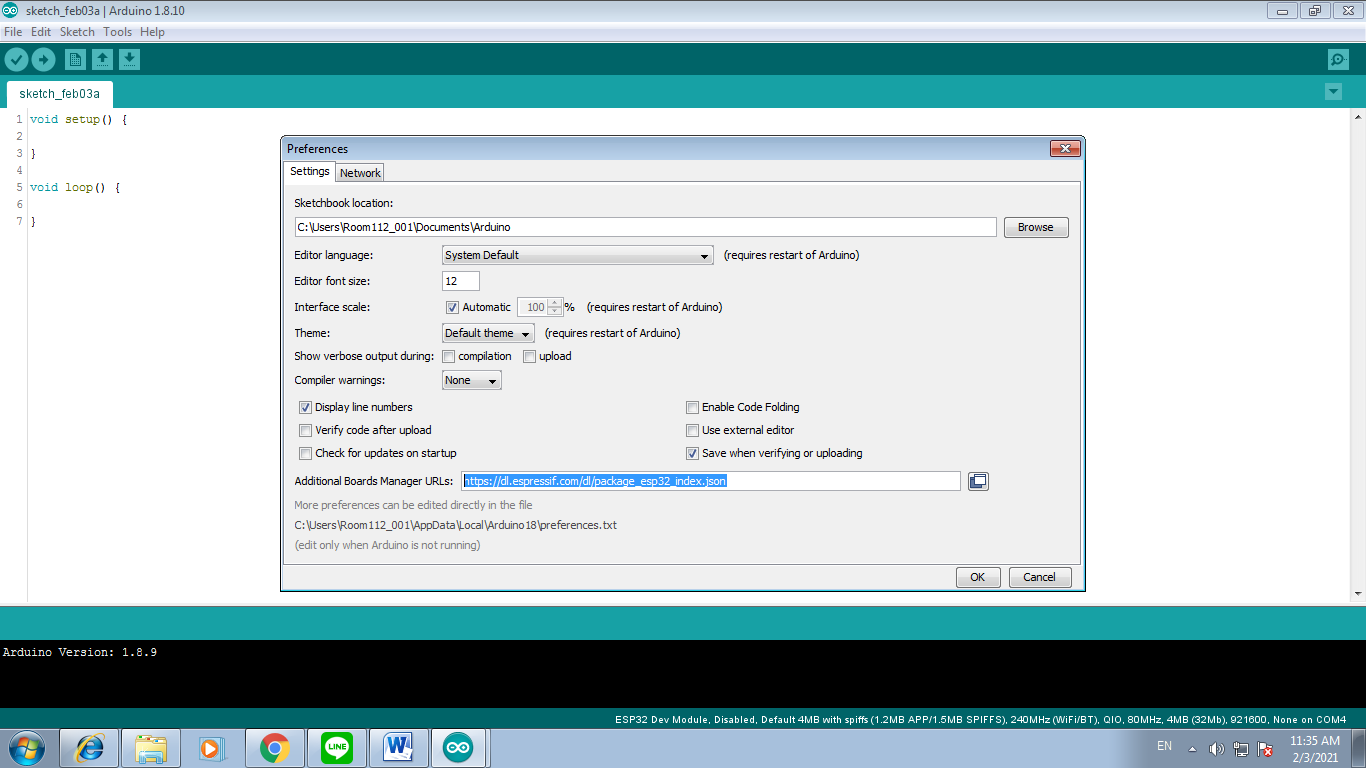
รูปที่ 2.1.4

4.เมื่อเปิดโปรแกรมขึ้นมาแล้ว ให้ไปที่เมนู File -> Preferences เพื่อติดตั้งบอร์ด NodeMCU/ESP32 แบบออนไลน์



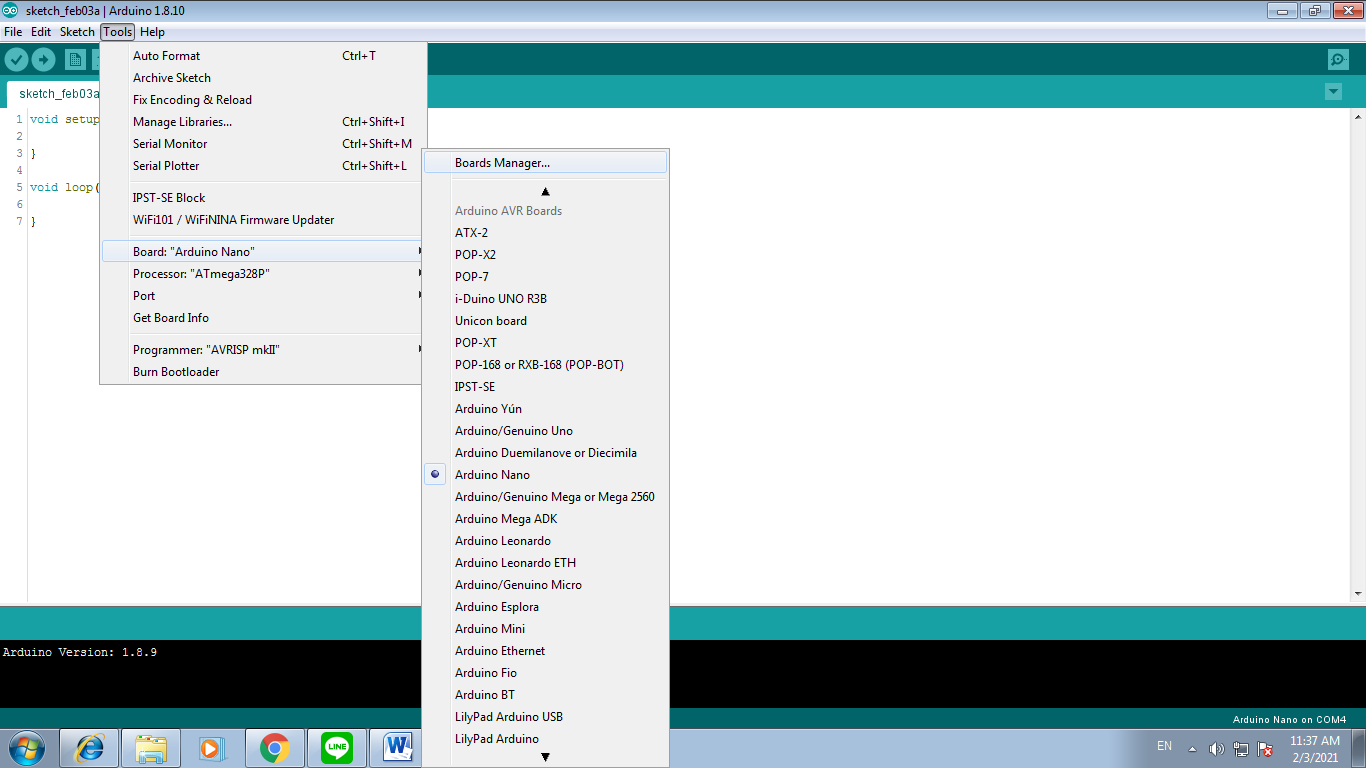
รูปที่ 2.1.5

6.เพิ่ม https://dl.espressif.com/dl/package\_esp32\_index.json ลงในช่อง Additional Boards Manager URLs



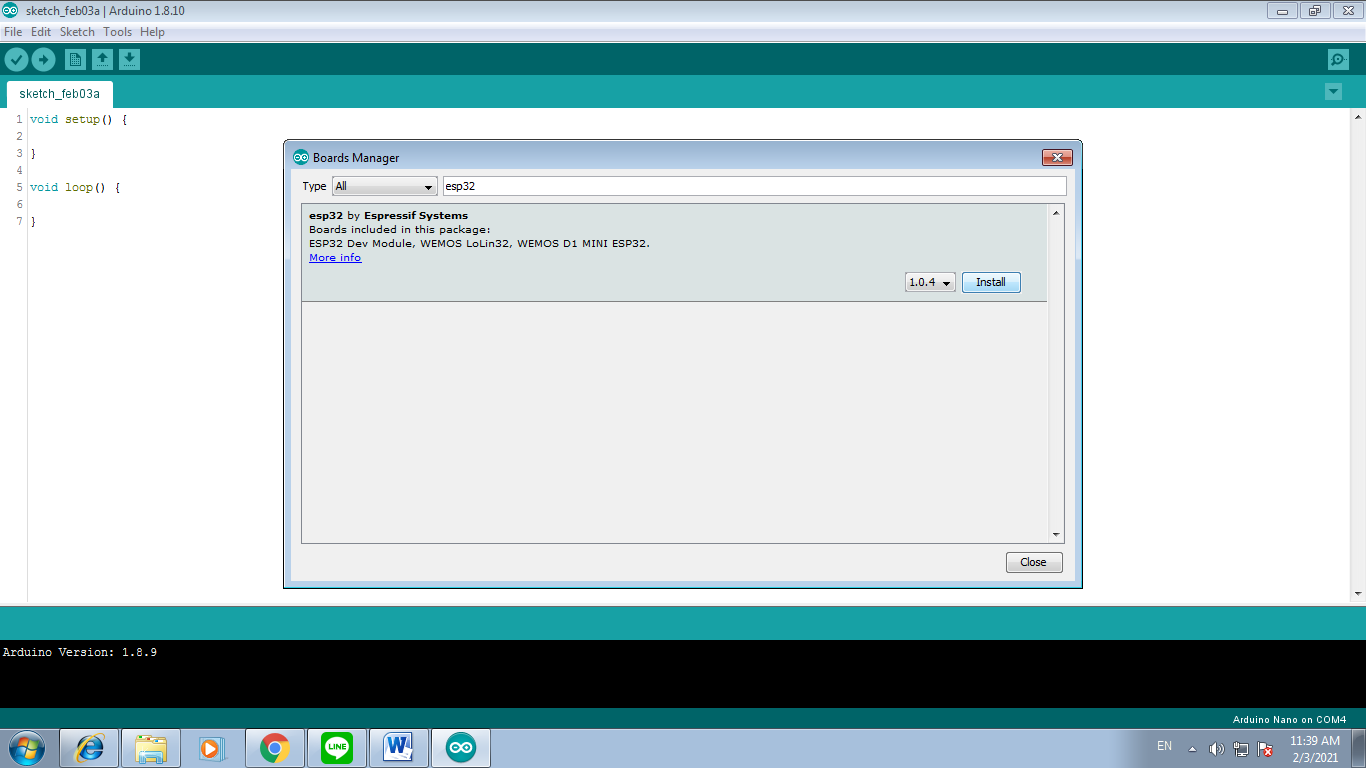
รูปที่ 2.1.6

7.คลิกไปที่เมนู Tools -> Board -> Board Manager



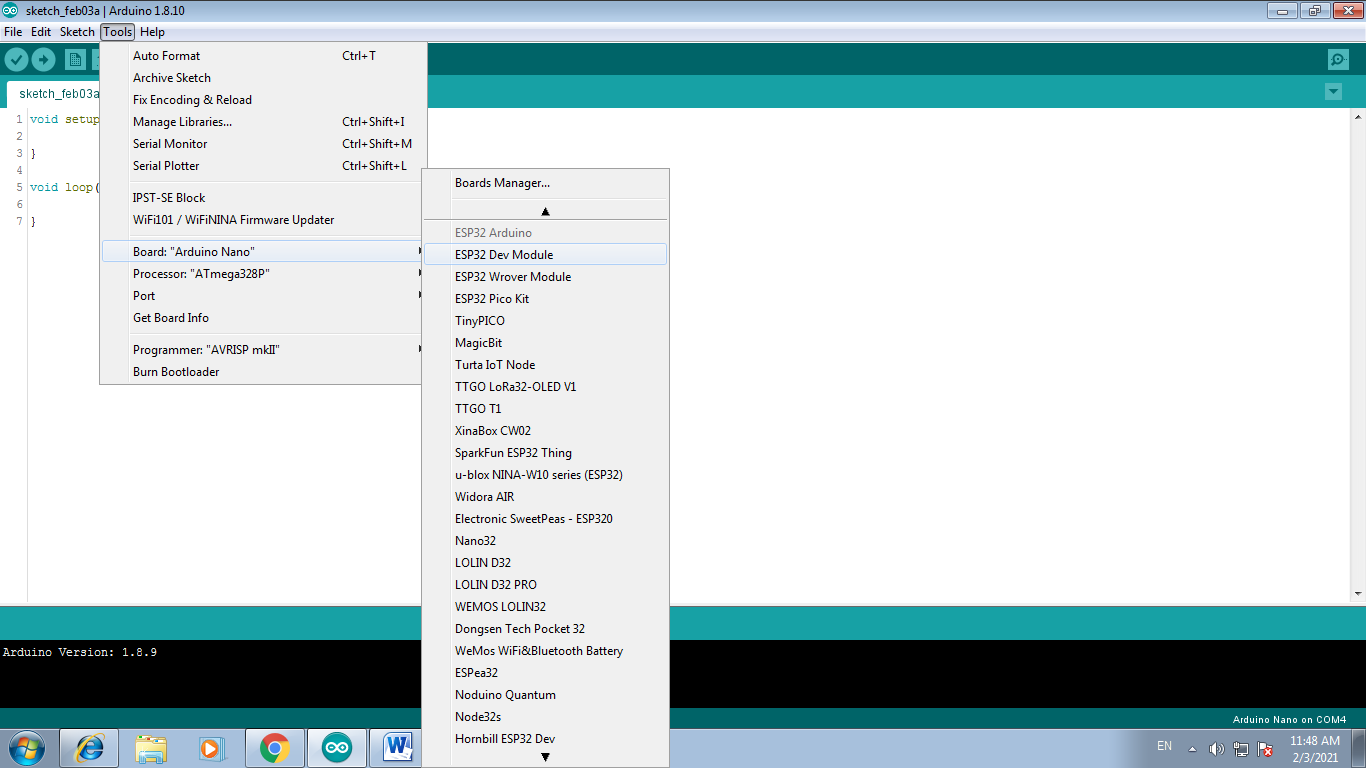
รูปที่ 2.1.7

8.พิมพ์คำว่า ESP32 ลงในช่อง และเริ่มต้นติดตั้งดังภาพ



รูปที่ 2.1.8

10.เมื่อติดตั้งเสร็จสิ้น ในหน้าต่าง Board ก็จะปรากฏประเภทของบอร์ด ESP32 ขึ้นมา ให้เลือกใช้งานได้แล้ว



รูปที่ 2.1.9

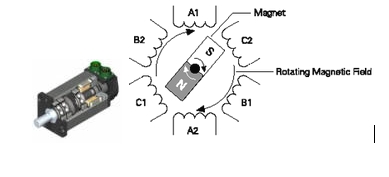
**2.2 MGR 996R Servo Motor**

เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของมัน (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) เป็นอุปกรณ์ที่สามารถควบคุมเครื่องจักรกล หรือระบบการทํางานนั้นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตําแหน่ง (Position), ระยะทางในการเคลื่อนที่(หมุน) (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมในลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยําสูง

**หลักการทำงานของเซอร์โวมอเตอร์**

การทำงานของเซอร์โวมอเตอร์ชนิดนี้จะคล้ายกับการทำงานของซิงโครนัสมอเตอร์ 3 เฟส กล่าวคือเมื่อมีการควบคุมให้คอนโทรลเลอร์จ่ายกระแสไฟฟ้าเข้าไปยังขดลวดที่สเตเตอร์ แกนเหล็กของสเตเตอร์จะกลายเป็นแม่เหล็กไฟฟ้า และหมุนเคลื่อนที่ด้วยความเร็วที่แปรผันตามความถี่ ซึ่งเรียกว่า ความเร็วซิงโครนัส (synchronous speed) หรือความเร็วสนามแม่เหล็กหมุน และจะดูดให้โรเตอร์ซึ่งเป็นแม่เหล็กถาวรหมุนเคลื่อนที่ตาม

จากลักษณะโครงสร้างของโรเตอร์และหลักการทำงานที่เหมือนกับซิงโครนัสมอเตอร์ซึ่งเป็นมอเตอร์แบบเอซี แต่ไม่มีแปรงถ่าน (Brushless) ไม่มีซี่คอมมิวเตอรเตอร์ จึงทำให้มอเตอร์ชนิดนี้มีชื่อเรียกขานแตกต่างกันออกไป เช่น เรียกทับศัพท์ว่า Permanent Magnet Synchronous Motor(PMSM) ซึ่งหมายถึงซิงโครนัสมอเตอร์ที่ไม่มีแปรงถ่าน บ้างก็เรียกว่าเอซีเซอร์โวมอเตอร์ (AC Servo motor) หรือบ้างก็เรียกสั้นๆย่อๆว่า AC Brushless หรือ Brushless Motor เป็นต้น



รูปที่ 2.2.1 โครงสร้างของโรเตอร์ ที่มา advance-electronic.com

**Servo Tower Pro MG996R**

Servo Tower Pro MG996R เป็นรุ่นที่อัพเกรดจากรุ่น Tower Pro MG995 เฟืองเป็นแบบโลหะ หมุน 0-180องศา ให้แรงบิดสูงถึง 15 kg



รูปที่ 2.2.2  [Servo Tower Pro MG996R ที่มา commandronestore.com](https://www.google.com/url?sa=i&url=https%3A%2F%2Fcommandronestore.com%2Fproducts%2Fbc201.php&psig=AOvVaw2GV9uGS1uB4UPIt0be_GUL&ust=1613012452200000&source=images&cd=vfe&ved=2ahUKEwji3o_jqd7uAhWVGXIKHcNFD7YQr4kDegUIARCHAg" \o "Tower Pro MG996R - Servo Motor - ขาย MG996R" \t "_blank)

Specification

-Use for: RC Model H309

-Dimension : 40 x 19 x 43mm

-Weight : 55g

-Operating Speed : 0.17sec / 60 degrees (4.8V no load)

-Operating Speed : 0.13sec / 60 degrees (6.0V no load)

-Stall Torque : 13 kg-cm (180.5 oz-in) at 4.8V

-Stall Torque : 15 kg-cm (208.3 oz-in) at 6V

-Operation Voltage : 4.8 - 7.2Volts

-Gear Type: All Metal Gears

-Connector Wire: 11.81" (300mm)

-Color: Black

-Compliant with most standard receiver connector: Futaba, Hitec, Sanwa, GWS etc...

-Great for truck, Boat, Racing Car, Helicopter and Airplane.

-Power Supply: Through External Adapter.

-Stable and Shock Proof

-Double Ball Bearing

**วิธีการต่อและใช้งาน Servo Tower Pro MG996R**

1.เชื่อมต่ออุปกรณ์

ESP32 -> MG996R Servo หมุนแบบ 0-180 องศา

-ขา D13 ->สายสีเหลือง

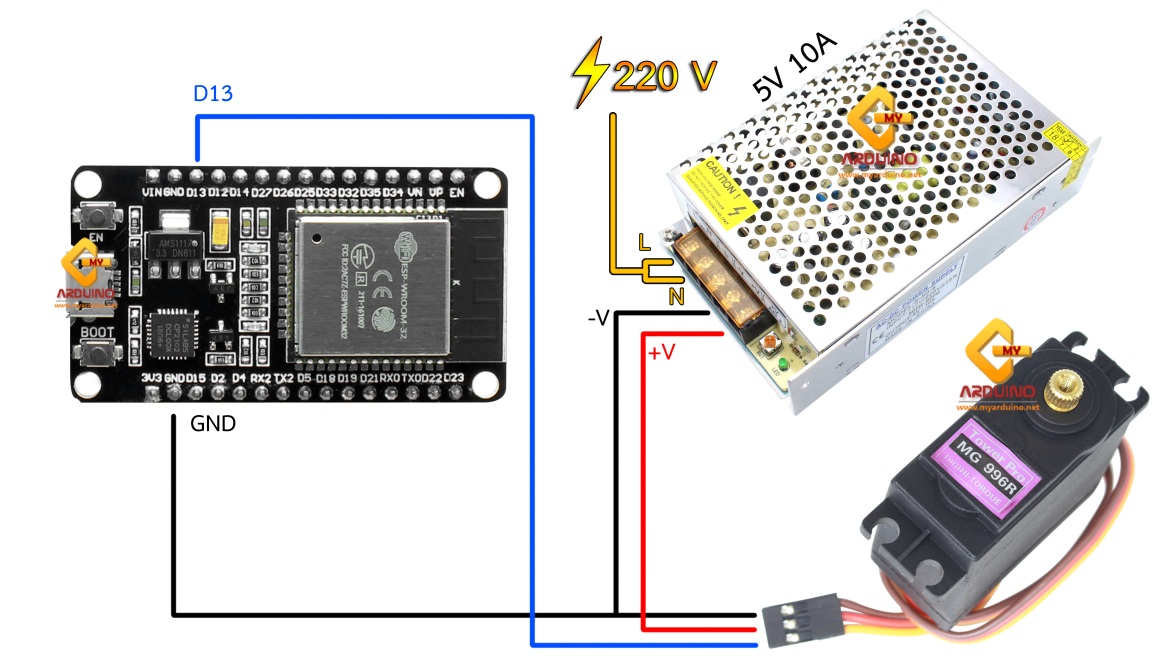
ESP32 -> Switching 5V 10A

-GND -> ( -V )

MG996RServo หมุนแบบ 0-180 องศา -> Switching 5V 10A

-สายสีแดง -> ( +V )

-สายสีน้ำตาล -> ( -V )

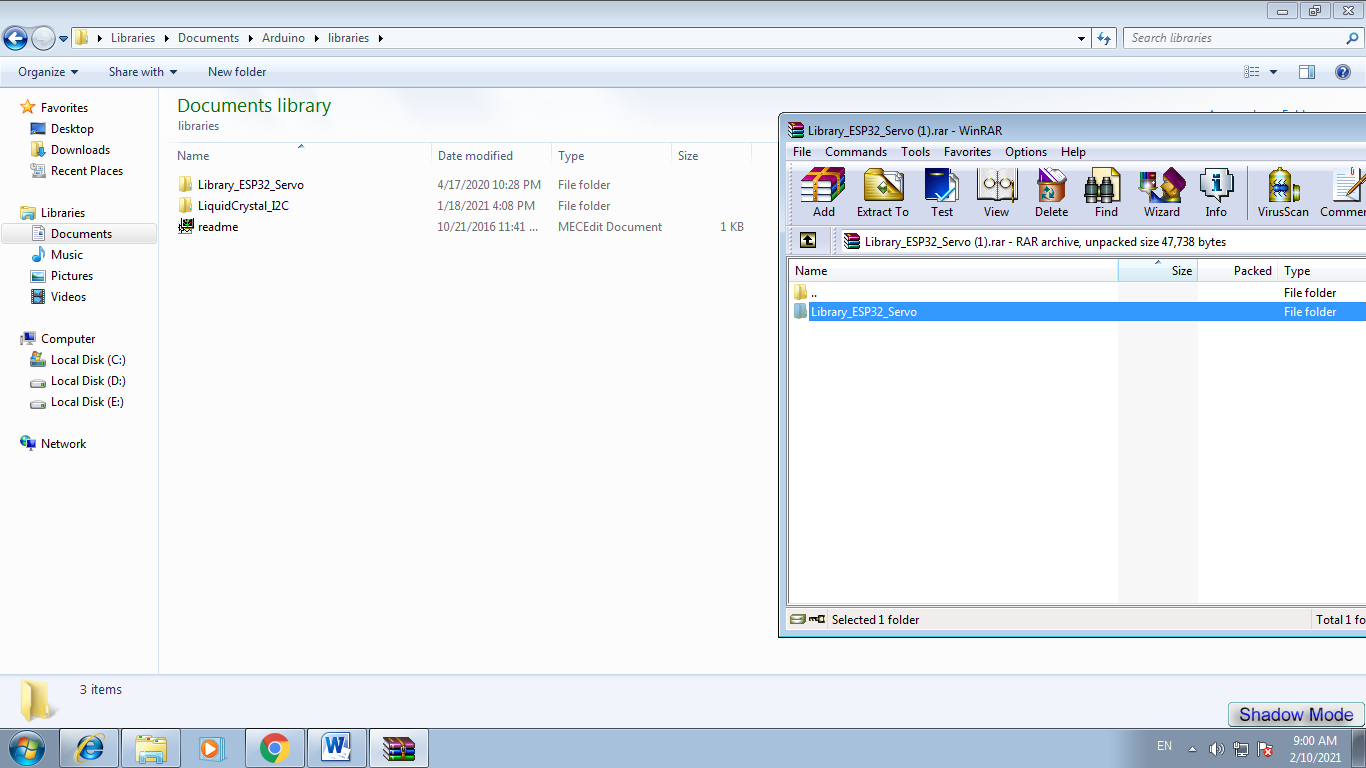


รูปที่ 2.2.3 การต่ออุปกรณ์ ที่มาmyarduino.net

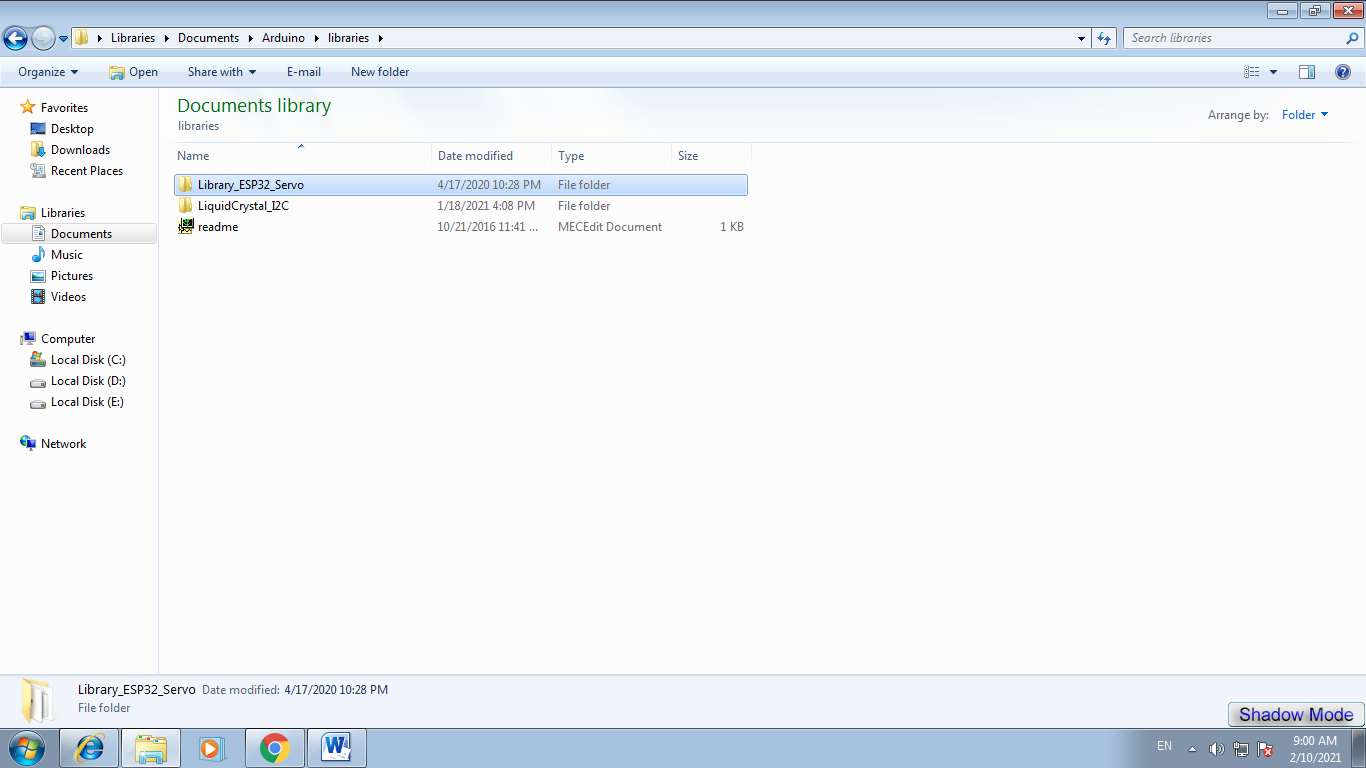
2.ติดตั้งLibrary

โหลด Library จากลิ้งก์นี้ <http://www.mediafire.com/file/56agj48q66a9e85/Library_ESP32_Servo.rar/file>

-ติดตั้งโดยนำไฟล์ไปไว้ใน C:\(ชื่อเครื่อง)\.....\Documents\Arduino\libraries

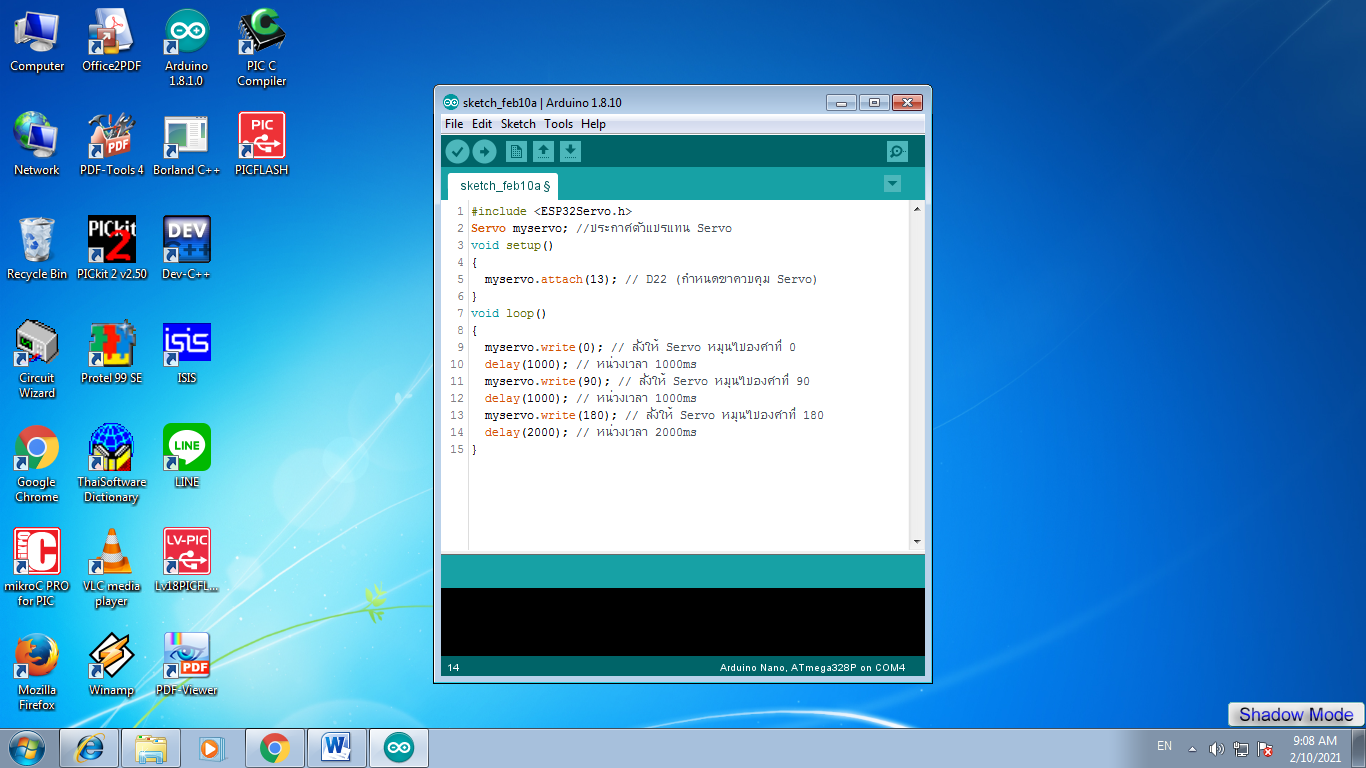


รูปที่ 2.2.4



รูปที่ 2.2.5

สามารถใช้งาน Arduino IDE สำหรับเขียนโปรแกรมServoได้แล้ว



รูปที่ 2.2.6

**2.3เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04**

โมดูลอัลตร้าโซนิคนี้เป็นอุปกรณ์ใช้วัดระยะทางโดยไม่ต้องมีการสัมผัสกับตำแหน่งที่ต้องการวัด วัดได้ตั้งแต่ 2 cm ถึง 400 cm โดยส่งสัญญาณอัลตร้าโซนิคความถี่ 40 kHz ไปที่วัตถุที่ต้องการวัดและรับสัญญาณที่สะท้อนกลับมา พร้อมทั้งจับเวลาเพื่อนำมาใช้ในการคำนวณระยะทาง



รูปที่ 2.3.1 เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04 ที่มา ktshopth.com

Specifications:

Working voltage: DC 5 V

Static current: 3 mA

Working temperature: 0 ~ + 70

Output way: GPIO

Induction Angle: Less than 15

Detection range:2 cm to 4 m

Detecting precision: 0.3 cm + 1%

Sensor size: Approx. 45 x 20 x 1.6mm

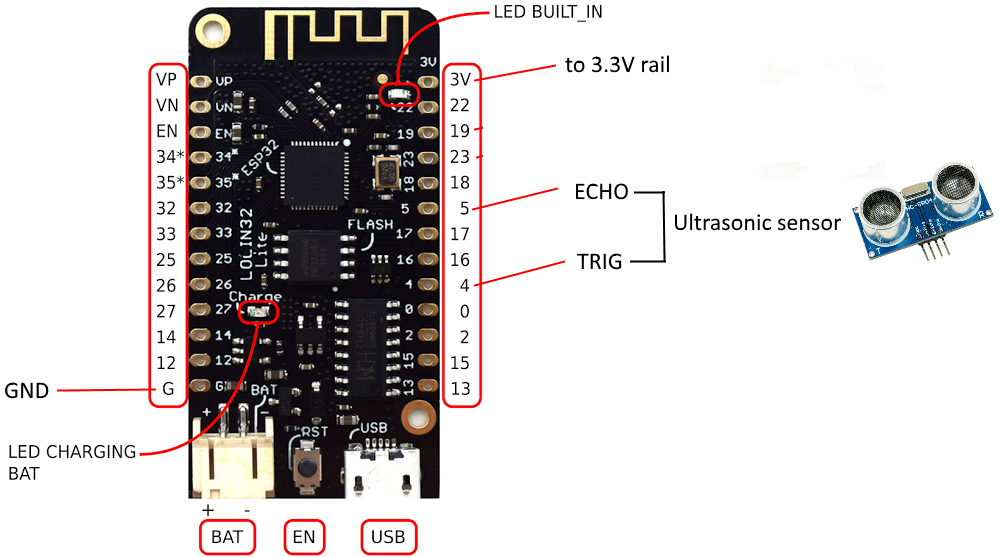
การต่อเพื่อใช้งานทำได้โดยต่อ PIN ดังนี้

PIN ที่ 1 -- 5V Supply

PIN ที่ 2 -- Trigger Pulse Input

PIN ที่ 3 -- Echo Pulse Output

PIN ที่ 4 -- GND



รูปที่ 2.3.2 การต่อใช้งาน ที่มา https://home.et.utwente.nl/

**2.4 โมดูล RTC (Real Time Clock) DS3231**

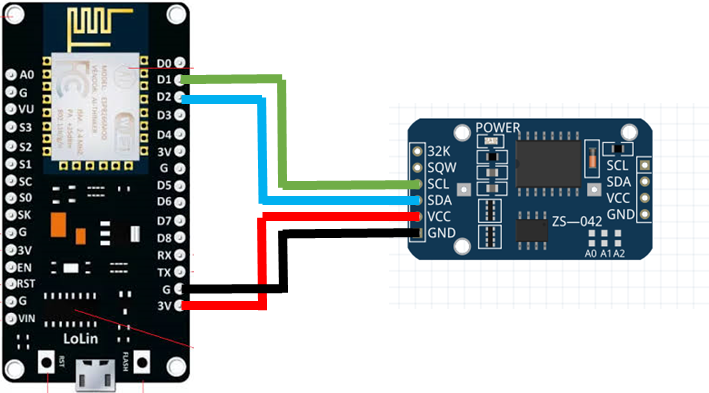


รูปที่ 2.4.1 โมดูล RTC (Real Time Clock) DS3231 ที่มา allnewstep.com

ในการบันทึกข้อมูลลง SD card หรือส่งข้อมูลออกไปยังผู้ใช้งาน หรือการตั้งเวลาเปิดปิดอุปกรณ์ไฟฟ้าตามช่วงเวลา เราต้องใช้เวลาที่ตรงกับเวลาจริงเพื่อให้เกิดความต่อเนื่องในการทำงานอย่างถูกต้อง ซึ่งโดยปกติแล้วไมโครคอนโทรลเลอร์ ไม่สามารถดึงข้อมูลเวลาจริงจากอินเตอร์เน็ตอย่างเช่นที่คอมพิวเตอร์หรืออุปกรณ์ IoT ทำได้ เพราะเวลาที่ได้จากบนตัวบอร์ดเช่น micros(); จะนับเวลาจากเริ่มเปิดเครื่อง ทำให้เวลาปิดเปิดเครื่องใหม่เวลาที่ตั้งไว้ก็จะหายไปด้วย

จึงมีการพัฒนาโมดูลสำหรับเก็บเวลาอย่าง DS3231 ที่จะเสมือนมีนาฬิกาไว้ติดบนมือของ MCU เพื่อคอยบอกเวลา โดยตัว DS3231 มีแบตเตอรี่ติดกับตัวโมดูลซึ่งใช้งานได้เป็นปี

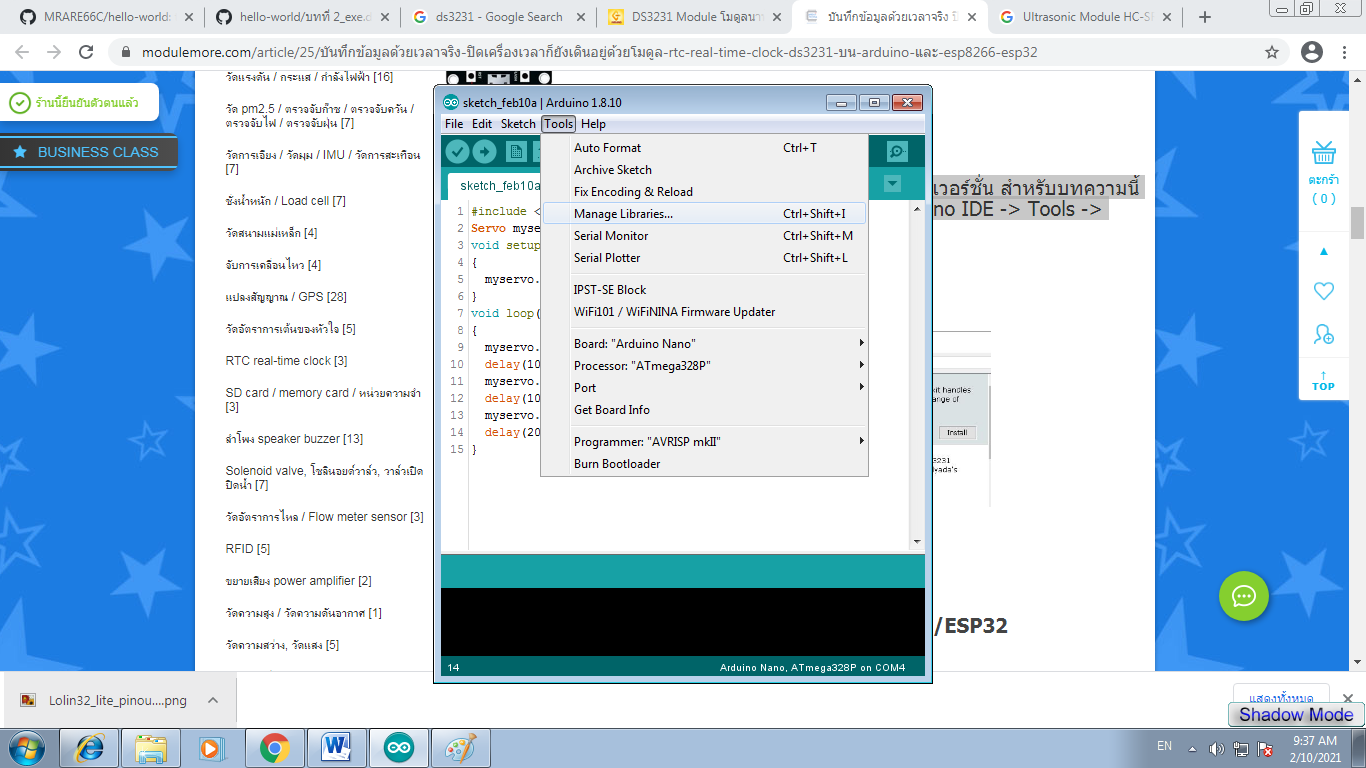
การต่อใช้งาน RTC DS3231 กับ ESP8266 (Nodemcu v3)



รูปที่ 2.4.2 การต่อใช้งาน RTC DS3231 ที่มา modulemore.com

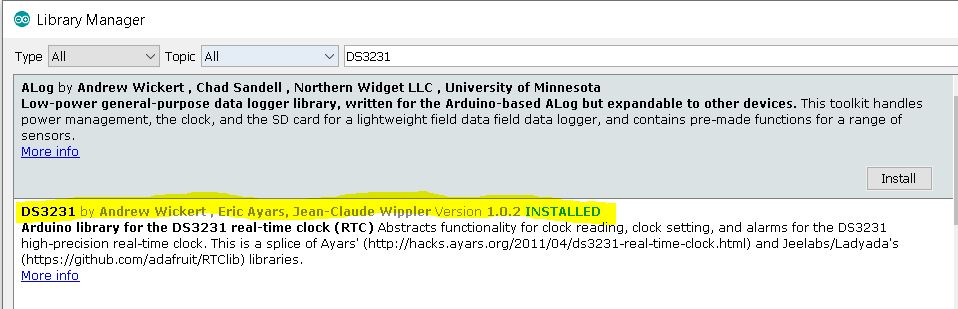
วิธีการลง Library DS3231 ด้วย Arduino IDE

ในการใช้ DS3231 เราต้องการ Library ซึ่งมีผู้พัฒนาอยู่หลายเวอร์ชั่น ซึ่งสามารถโหลดได้จาก Arduino IDE -> Tools -> Manage Libraries



รูปที่ 2.4.3

จากนั้นพิมพ์ DS3231 เลือกติดตั้ง Library ดังรูป



รูปที่ 2.4.4

การเซ็ตเวลาให้โมดูล DS3231 บน Arduino/ESP8266/ESP32

ให้เราทำการเขียนโค้ดใส่ใน sketch ของเรา

/\*\*

\* www.arduinona.com

\* ตัวอย่างการเซ็ตค่าเวลา ให้ DS3231

\* ใช้คู่กับ Library DS3231 จาก https://github.com/PilleStat/PilleStat/tree/master/arduino%20code/libraries/DS3231

\* วิธีการต่อ

\* Arduino UNO -> DS3231

\* 5V -> VCC

\* GND -> GND

\* SDA -> A4

\* SCL -> A5

\*/

#include <DS3231.h>

#include <Wire.h>

DS3231 Clock;

byte Year;

byte Month;

byte Date;

byte DoW;

byte Hour;

byte Minute;

byte Second;

/\*\*

\* ฟังก์ชันนี้ใช้รับค่าจาก Serial monitor เพื่อตั้งเวลา โดยมีรูปแบบคือ

\* YYMMDDwHHMMSS และ x ลงท้าย เช่น

\* (2001315105300x คือ ปี 20 เดือน 01 วัน 31 วันศุกร์(5) 10 นาฬิกา 53 นาที 00 วินาที)

\*/

void GetDateStuff(byte& Year, byte& Month, byte& Day, byte& DoW,

byte& Hour, byte& Minute, byte& Second) {

/\*\*

\* Call this if you notice something coming in on

\* the serial port. The stuff coming in should be in

\* the order YYMMDDwHHMMSS, with an 'x' at the end.

\*/

boolean GotString = false;

char InChar;

byte Temp1, Temp2;

char InString[20];

byte j=0;

while (!GotString) {

if (Serial.available()) {

InChar = Serial.read();

InString[j] = InChar;

j += 1;

if (InChar == 'x') {

GotString = true;

}

}

}

Serial.println(InString);

// Read Year first

Temp1 = (byte)InString[0] -48;

Temp2 = (byte)InString[1] -48;

Year = Temp1\*10 + Temp2;

// now month

Temp1 = (byte)InString[2] -48;

Temp2 = (byte)InString[3] -48;

Month = Temp1\*10 + Temp2;

// now date

Temp1 = (byte)InString[4] -48;

Temp2 = (byte)InString[5] -48;

Day = Temp1\*10 + Temp2;

// now Day of Week

DoW = (byte)InString[6] - 48;

// now Hour

Temp1 = (byte)InString[7] -48;

Temp2 = (byte)InString[8] -48;

Hour = Temp1\*10 + Temp2;

// now Minute

Temp1 = (byte)InString[9] -48;

Temp2 = (byte)InString[10] -48;

Minute = Temp1\*10 + Temp2;

// now Second

Temp1 = (byte)InString[11] -48;

Temp2 = (byte)InString[12] -48;

Second = Temp1\*10 + Temp2;

}

void setup() {

/\*\*

\* เริ่มการสื่อสารกับ Serial monitor

\*/

Serial.begin(115200);

/\*\*

\* เริ่มการสื่อสารแบบ I2C

\*/

Wire.begin();

}

void loop() {

/\*\*

\* เช็คว่ามีค่าเข้ามาทาง Serial monitor หรือไม่

\* ถ้าใช่ให้นำค่านั้นไปคิดในฟังก์ชัน GetDateStuff

\*/

if (Serial.available()) {

GetDateStuff(Year, Month, Date, DoW, Hour, Minute, Second);

Clock.setClockMode(false); // ตั้งเวลาแบบ 24h

//setClockMode(true); // ตั้งเวลาแบบ 12h

Clock.setYear(Year);

Clock.setMonth(Month);

Clock.setDate(Date);

Clock.setDoW(DoW);

Clock.setHour(Hour);

Clock.setMinute(Minute);

Clock.setSecond(Second);

/\*\*

\* ทดสอบฟังก์ชันเซ็ตนาฬิกาปลุก โดยตั้งเวลานาฬิกาปลุก A1 ให้ล่วงหน้าค่าที่ตั้งไว้ 1 นาที และ A2 ล่วงหน้าไว้ 2 นาที

\* หากมีการปลุกเมื่อไรให้แจ้งออกมาทาง external interrupt

\*/

Clock.setA1Time(DoW, Hour, Minute+1, Second, 0x0, true,

false, false);

// set A2 to two minutes past, on current day of month.

Clock.setA2Time(Date, Hour, Minute+2, 0x0, false, false,

false);

// Turn on both alarms, with external interrupt

Clock.turnOnAlarm(1);

Clock.turnOnAlarm(2);

}

/\*\*

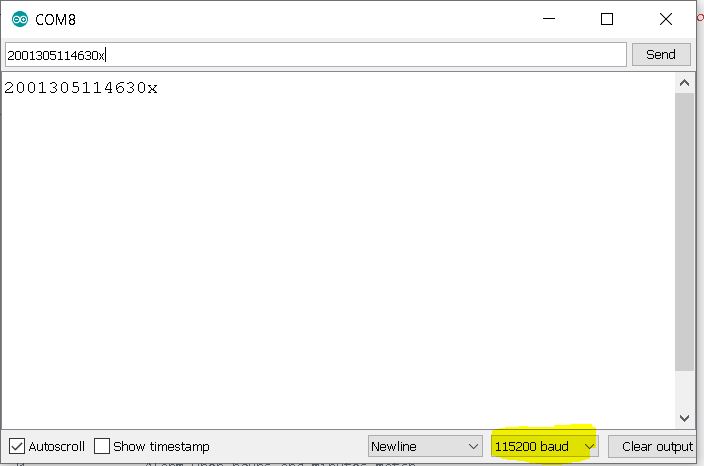
\* อัพเดททุก 1 วินาที

\*/

delay(1000);

}

ให้อัพโหลดและเปิด Serial monitor ขึ้นมาจาก Tools->Serial monitor จากนั้นเปลี่ยน baudrate ให้เป็น 115200 ดังรูป



รูปที่ 2.4.5

จากนั้นพิมพ์ ปี เดือน วัน วันที่ในสัปดาห์(Day of week) ชั่วโมง นาที วินาที ตามด้วยตัว x ปิดท้าย เช่น

ปี 2020 ใส่ 20

เดือน 1 ใส่ 01

วันที่ 30 ใส่ 30

วันที่ในสัปดาห์ (วันศุกร์ =5) ใส่ 5

ชั่วโมง 11

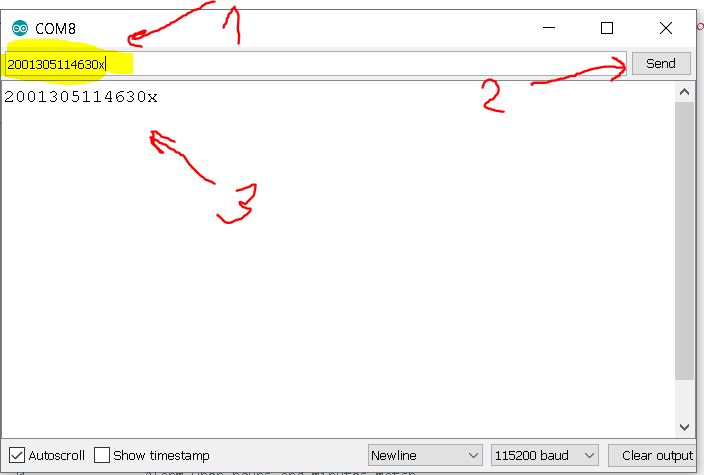
นาที 46

วินาที 30

จึงรวมกันได้เป็น

2001305114630x เป็นต้น

ให้เราใส่เลขข้างต้นในตำแหน่งที่ 1 ตามรูปด้านล่าง แล้วกด send เพื่อเซ็ตเวลา ควรจะมีหมายเลข 3 ตอบกลับมาดังรูป แสดงให้ทราบว่าเวลาได้ถูกเซ็ตแล้ว



รูปที่ 2.4.6