# เทคโนโลยีการบำบัดน้ำเสียของบ่อเพาะเลี้ยงสัตว์น้ำ
# TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS


ยอดขุนพล นุ่มโหมด
Yodkunpol Nummode
อัครพันธ์ จันทโร
Akkharaphan Jantharo


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# เทคโนโลยีการบำบัดน้ำเสียของบ่อเพาะเลี้ยงสัตว์น้ำ

# TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS

T144360

ยอดขุนพล นุ่มโหมด
Yodkunpol Nummode
อัครพันธ์ จันทโร
Akkharaphan Jantharo

สซหมู
เลขทะเบียน 144360
รับ.เดือน.ปี 24 พ.ย. 2559

b 14820052
i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

# TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS

Yodkunpol Nummode
Akkharaphan Jantharo

THISES IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTYN OF ENGINEERING
KINGS MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

**คณะวิศวกรรมศาสตร์**
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**
**ใบรับรองปริญญานิพนธ์**

---

| | |
|---|---|
| **หัวข้อปริญญานิพนธ์** | เทคโนโลยีการบำบัดน้ำเสียของบ่อเพาะเลี้ยงสัตว์น้ำ |
| **Thesis Title** | TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS |
| **ชื่อนักศึกษา** | นายยอดขุนพล นุ่มโหมด<br>นายอัครพันธ์ จันทโร |
| **ระดับปริญญา** | วิศวกรรมศาสตรบัณฑิต |
| **สาขาวิชา** | วิศวกรรมสารสนเทศ |
| **ปริญญานิพนธ์ปีการศึกษา** | 2558 |

(.....)
ผศ.ไพศาล สิทธิโยภาสกุล
**อาจารย์ที่ปรึกษาปริญญานิพนธ์**

(.....)
อ.สรพงษ์ วชิรรัตนพรกุล
**อาจารย์ที่ปรึกษาปริญญานิพนธ์ร่วม**

**หัวข้อปริญญานิพนธ์** เทคโนโลยีบำบัดน้ำเสียในบ่อเพาะเลี้ยงสัตว์น้ำ

**Thesis Title** TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS


**ชื่อนักศึกษา** นายยอดขุนพล นุ่มโหมด รหัสนักศึกษา 55011006

นายอัครพันธ์ จันทโร รหัสนักศึกษา 55011450


**ระดับปริญญา** วิศวกรรมศาสตรบัณฑิต

**สาขาวิชา** วิศวกรรมสารสนเทศ

**ปีการศึกษา** 2558

**อาจารย์ที่ปรึกษาปริญญานิพนธ์** ผู้ช่วยศาสตราจารย์ไพศาล สิทธิโยภาสกุล

**อาจารย์ที่ปรึกษาปริญญานิพนธ์ร่วม** อาจารย์สรพงษ์ วชิรรัตนพรกุล

## บทคัดย่อ


โครงงานนี้แสดงถึงการสร้างอุปกรณ์เพื่อทำการวัดค่า pH ของน้ำและทำการแสดงผลผ่านเว็บโดยใช้ WIFI อุปกรณ์ที่ใช้สร้างผลงานชิ้นนี้ มีส่วนประกอบที่สำคัญคือไมโครคอนโทรลเลอร์ Arduino Mega 2560 (ATmega2560) pH โมดูล ESP8266 NodeMCU v.1.0 และ pH Probe หลักการทำงานคือ เมื่อเอา pH Probe ไปจุ่มน้ำ pH Probe จะส่งค่าแรงเคลื่อนไฟฟ้า ไปยัง pH Sensor Module จากนั้น pH Sensor Module จะเปลี่ยนค่าแรงเคลื่อนไฟฟ้าเป็นค่า pH แล้วส่งค่าไปที่ ESP8266 NodeMCU v.1.0 จากนั้น ESP8266 NodeMCU v.1.0 จะแสดงค่า pH ผ่านทางเว็บไซต์ และสามารถสั่งให้ Motor และ Solenoid Value ทำงานได้ผ่านทางเว็บไซต์ หลังจากสั่งงานแล้ว ESP8266 NodeMCU v.1.0 จะส่งค่าที่สั่งใน Website ไปยัง Arduino Mega 2560 เพื่อให้ Arduino Mega 2560 สั่ง ชุดขับมอเตอร์ สั่งมอเตอร์ หรือ วาล์วโซลินอย์ ทำงานตามที่ผู้ใช้งานต้องการ และยังสามารถทำงานสั่งค่าได้ด้วยตัวเองเมื่อค่า pH เกินกว่าค่าที่กำหนดไว้

| | |
|---|---|
| **Thesis Title** | TECHNOLOGY FOR WASTERWATER TREATMENT IN AQUACULTURE PONDS |
| **Student** | Mr.Yodkunpol Nummode Student ID. 55010006 |
| | Mr.Akkharaphan Jantharo Student ID. 55011450 |
| **Degree Program** | Bachelor of Engineering Information Engineering |
| **Academic Year** | 2015 |
| **Thesis Advisor** | Asst.Prof. Paisan Sithiyopasakul |
| **Thesis Co-Advisor** | Mr.Sorapong Wachirarattanapornkul |

# ABSTRACT

This project is about creating the device to measure the pH value in the water and display on web by using WIFI. In this device, the major components are Microcontroller Arduino Mega 2560 (ATmega2560), pH Sensor Module, ESP8266 NodeMCU v.1.0 and pH Probe. It has Process, When pH Probe measure water. pH Probe will send electromotive force value to pH Sensor Module to Change value electromotive force to a pH value then it will send value to the ESP8266 NodeMCU v.1.0, and then ESP8266 NodeMCU v.1.0 will show pH value by website and can order Motor and Solenoid Value work by website. After that ESP8266 NodeMCU v.1.0 will send value that command in website to the Arduino Mega 2560 to order Drive Motor to order Motor or Solenoid Value work by User need to do. And can order by themself when the pH value is exceed a predetermined value.

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยความอนุเคราะห์ช่วยเหลือจากอาจารย์ไพศาล สิทธิโยภาสกุลและอาจารย์สรพงษ์ วชิรรัตนพรกุลที่ช่วยให้คำชี้แนะและแนวคิดในการสร้างผลงานขึ้นมา

ขอบคุณพ่อแม่ ที่เลี้ยงดูและมอบกำลังให้กับพวกเรา จนทำให้พวกเราสามารถทำผลงานนี้ได้สำเร็จ

ขอบคุณเพื่อนๆ ที่คอยชี้แนะ ให้แนวคิด ให้กำลังใจ และอยู่ด้วยกันมาตลอด

ขอบคุณผู้มีพระคุณทุกๆคน ที่อาจจะไม่ได้กล่าวถึง ณ ที่แห่งนี้

สิ่งที่มีประโยชน์ สิ่งที่เกิดคุณค่า อันเกิดจากปริญญานิพนธ์นี้ คณะผู้จัดทำขอมอบให้แก่ผู้มีพระคุณทุกท่าน

ยอดขุนพล นุ่มโหมด

อัครพันธ์ จันทโร

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1
1.1 ความสำคัญและที่มาของโครงงาน.....1
1.2 วัตถุประสงค์ของการโครงงาน.....2
1.3 ขอบเขตของโครงงาน.....2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....2
1.5 อุปกรณ์ที่ใช้.....3
1.5.1 ฮาร์ดแวร์(Hardware).....3
1.5.2 ซอฟต์แวร์(Software).....3

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้.....4
2.1 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์ Arduino Mega 2560.....4
2.1.1 คุณสมบัติพื้นฐานของ Arduino Mega 2560.....5
2.1.2 คุณสมบัติพื้นฐานของ MCU.....6
2.1.3 คำอธิบายพอร์ตของ MCU.....7
2.2 ทฤษฎีเกี่ยวกับ ESP8266 NodeMCU v.1.0.....11
2.2.1 คุณสมบัติพื้นฐานของ ESP8266 NodeMCU v.1.0.....12
2.2.2 คำอธิบายพอร์ตของ ESP8266 NodeMCU v.1.0.....12
2.3 ทฤษฎีเกี่ยวกับโมดูล pH เซนเซอร์ (pH Sensor Module).....13
2.3.1 คุณสมบัติพื้นฐานของโมดูล pH เซนเซอร์.....13
2.4 ทฤษฎีเกี่ยวกับหัววัดค่า pH (pH Electrode with BNC plug GX-E1).....14
2.4.1 คุณสมบัติพื้นฐานของหัววัดค่า pH.....15

# สารบัญ(ต่อ)


หน้า

2.5 ทฤษฎีเกี่ยวกับชุดขับมอเตอร์ (L298 Dual H-Bridge Motor Driver).....16
2.5.1 คุณสมบัติพื้นฐานของชุดขับมอเตอร์.....16
2.5.2 พอร์ตของชุดขับมอเตอร์.....17
2.6 ทฤษฎีเกี่ยวกับมอเตอร์กระแสตรง (DC Motor) รุ่น CL-RS445PA-15200.....18
2.6.1 คุณสมบัติพื้นฐานของมอเตอร์กระแสตรง รุ่น CL-RS445PA-15200.....19
2.7 ทฤษฎีเกี่ยวกับ วาล์วโซลินอยด์ (Solenoid Value) รุ่น Uni-D UD-8.....20
2.7.1 คุณสมบัติพื้นฐานของวาล์วโซลินอยด์รุ่น Uni-D UD-8.....20
2.8 ทฤษฎีเกี่ยวกับทฤษฎีเกี่ยวกับอะแดปเตอร์กระแสสลับ (AC Adaptor).....21
2.9 ทฤษฎีเกี่ยวกับ WP1250 SOBO Aquarium Power Head.....22
2.9.1 คุณสมบัติพื้นฐานของ WP1250 SOBO Aquarium Power Head.....22
2.10 โปรแกรมที่เกี่ยวข้อง.....23
2.10.1 Arduino IDE.....23

บทที่ 3 การวิเคราะห์และออกแบบระบบ.....24
3.1 การออกแบบระบบ.....24
3.2 การออกแบบฮาร์ดแวร์.....25
3.2.1 ขั้นตอนในการต่ออุปกรณ์ฮาร์ดแวร์.....25
3.3 การออกแบบโปรแกรม (Program).....29
3.3.1 วิธีการใช้งาน NodeMCU ในโปรแกรม Arduino.....30
3.3.2 วิธีการใช้งาน NodeMCU ในการอ่านค่าจากโมดูล pH เซนเซอร์.....35
3.3.3 วิธีการใช้งาน NodeMCU ในการสร้าง Access Point และ Web Server.....39
3.3.4 วิธีการทำให้ Arduino และ NodeMCU สามารถติดต่อสื่อสารระหว่างกัน.....44
3.3.5 วิธีการใช้งาน Drive Motor โดยสั่งผ่าน Arduino Mega 2560 .....46

บทที่ 4 ผลการทดลอง.....59
4.1 ขั้นตอนในการต่ออุปกรณ์ฮาร์ดแวร์.....59
4.2 การทดลองการแสดงผลค่า pH ผ่านหน้าเว็บไซต์และส่งค่าผ่านทางเว็บไซต์.....63

# สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปและวิจารณ์ผลการทดลอง....72
5.1 บทสรุปโครงงาน....72
5.2 ปัญหาที่พบในระหว่างดำเนินงาน....72
5.3 แนวทางแก้ไข และพัฒนาต่อ....72

บรรณานุกรม....73

ภาคผนวก ก Poster....75
ภาคผนวก ข การติดตั้งโปรแกรม Arduino....77
ภาคผนวก ค วิธีการใช้งาน....84
ภาคผนวก ง Data Sheet....88
ภาคผนวก จ วิธีการต่ออุปกรณ์....101

# สารบัญตาราง

ตารางที่ หน้า

2.1 คุณสมบัติพื้นฐานของ Arduino Mega 2560 ..... 5
2.2 คุณสมบัติพื้นฐานของ MCU ..... 6
2.3 PIN mapping ของ MCU ..... 8
2.4 คุณสมบัติพื้นฐานของโมดูล pH เซนเซอร์ ..... 13
2.5 คุณสมบัติพื้นฐานของหัววัดค่า pH ..... 15
2.6 คุณสมบัติพื้นฐานของชุดขับมอเตอร์ ..... 16
2.7 คุณสมบัติพื้นฐานของมอเตอร์กระแสตรง รุ่น CL-RS445PA-15200 ..... 19
2.8 คุณสมบัติพื้นฐานของวาล์วโซลินอยด์ รุ่น Uni-D UD-8 ..... 20
2.9 คุณสมบัติพื้นฐานของ WP1250 SOBO Aquarium Power Head 800L/H ..... 22

# สารบัญรูป


รูปที่ หน้า

1.1 บล็อกไดอะแกรม....1

2.1 Arduino Mega 2560....4

2.2 บล็อกไดอะแกรมของ MCU : AVR ATmega2560....5

2.3 พอร์ตของ AVR ATmega 2560....7

2.4 ESP8266 NodeMCU v.1.0....11

2.5 พอร์ตของ ESP8266 NodeMCU v.1.0....12

2.6 ส่วนประกอบของโมดูล pH เซนเซอร์....13

2.7 หัววัดค่า pH....14

2.8 โครงสร้างของ combined electrode ที่ใช้สำหรับวัดค่า pH....14

2.9 ชุดขับมอเตอร์....16

2.10 พอร์ตของ L298 Dual H-Bridge Motor Drive....17

2.11 มอเตอร์กระแสตรง รุ่น CL-RS445PA-15200....18

2.12 หลักการทำงานของมอเตอร์....18

2.13 วาล์วโซลินอยด์....20

2.14 อะแดปเตอร์กระแสสลับ....21

2.15 WP1250 SOBO Aquarium Power Head....22

2.16 การเลือกบอร์ดที่ทำการเชื่อมต่อ และต้องการเขียนชุดคำสั่ง....23

3.1 โฟลว์ชาร์ตการทำงานของอุปกรณ์....24

3.2 วงจรที่ออกแบบไว้....25

3.3 การต่อวงจรของ NodeMCU เข้ากับ Arduino....27

3.4 การต่อวงจรของโมดูล pH เซนเซอร์ เข้ากับ Arduino และ NodeMCU....27

3.5 การต่อวงจรของ Arduino เข้ากับชุดขับมอเตอร์, วาล์วโซลินอยด์และมอเตอร์....28

3.6 การต่อวงจรของ Arduino และชุดขับมอเตอร์เข้ากับอะแดปเตอร์ 9 V และ 12 V....28

3.7 วงจรที่เชื่อมต่อเข้าคอมพิวเตอร์ของ Arduino Mega 2560 และ NodeMCU v.1.0....29

3.8 วงจรที่เชื่อมต่อ Arduino Mega 2560 และ NodeMCU v.1.0 เข้ากับคอมพิวเตอร์....29

3.9 หน้าต่าง Preference....30

3.10 หน้าต่างของโปรแกรม Preference....31

# สารบัญรูป (ต่อ)


รูปที่ หน้า

3.11 หน้าต่างวิธีเลือก Board Manager....32
3.12 หน้าต่าง Board Manager....32
3.13 เปลี่ยน Type เป็น Contributed แล้วกดปุ่ม Install....33
3.14 ข้อความที่ขึ้นว่า INSTALLED....33
3.15 วิธีเลือก Board เป็น NodeMCU 1.0 (ESP-12E Module....34
3.16 โปรแกรมที่อ่านค่าชุดข้อมูลอนาล็อก....35
3.17 โค้ดภาษา C ภายในโปรแกรมตัวอย่าง....35
3.18 การ Upload โค้ดใส่ NodeMCU v.1.0....36
3.19 การเปิดหน้าต่าง Serial Monitor เพื่อดูค่าอนาล็อกที่ส่งมาจากเซนเซอร์....36
3.20 หน้าต่างแสดงค่าอนาล็อกจากเซนเซอร์....37
3.21 โค้ดที่ออกแบบเพื่ออ่านค่า pH....37
3.22 หน้าต่างแสดงค่า pH....38
3.23 โปรแกรมที่ทำให้ NodeMCU กลายเป็น Access Point....39
3.24 โค้ดภาษา C ภายในโปรแกรมตัวอย่าง....40
3.25 ตำแหน่งที่เปลี่ยน SSID และ password และความเร็ว Baud Rate....41
3.26 Wi-Fi ชื่อ Gogo12345....42
3.27 ทำการเชื่อมต่อกับ Wi-Fi ชื่อ Gogo12345....42
3.28 วิธีเข้า Web Server....43
3.29 Web Server ที่ได้สร้างไว้ใน Arduino....43
3.30 Code ภาษา C ที่ใช้ทำการออกแบบระบบติดต่อสื่อสารกัน....44
3.31 หน้าต่าง Serial Monitor ที่สื่อสารระหว่างกัน....45
3.32 โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 1....46
3.33 โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 2....47
3.34 โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 3....48
3.35 วิธีส่งค่าให้ชุดขับมอเตอร์ทำงาน....48
3.36 Serial Monitor หลังจากทำการส่งค่า 1....49
3.37 วาล์วโซลินอยด์ทำงาน....49

# สารบัญรูป (ต่อ)


รูปที่ หน้า

3.38 Serial Monitor หลังจากทำการส่งค่า 2 ..... 50
3.39 วาล์วโซลินอยด์หยุดทำงาน ..... 50
3.40 Serial Monitor หลังจากทำการส่งค่า 3 ..... 51
3.41 มอเตอร์ทำงาน ..... 51
3.42 Serial Monitor หลังจากทำการส่งค่า 4 ..... 52
3.43 มอเตอร์หยุดทำงาน ..... 52
3.44 Serial Monitor หลังจากทำการส่งค่า 5 ..... 53
3.45 วาล์วโซลินอยด์และมอเตอร์ทำงาน ..... 53
3.46 Serial Monitor หลังจากทำการส่งค่า 6 ..... 54
3.47 วาล์วโซลินอยด์และมอเตอร์หยุดทำงาน ..... 54
3.48 โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 1 ..... 55
3.49 โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 2 ..... 56
3.50 โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 3 ..... 56
3.51 โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 1 ..... 57
3.52 โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 2 ..... 57
3.53 โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 3 ..... 58
3.54 โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 4 ..... 58
4.1 โมดูล pH เซนเซอร์, ชุดขับมอเตอร์, Arduino Mega 2560 และNodeMCU v.1.0 ..... 59
4.2 มอเตอร์และวาล์วโซลินอยด์ ..... 60
4.3 วิธีการต่อ Arduino Mega 2560 เข้ากับ NodeMCU v.1.0 ..... 60
4.4 วิธีการต่อโมดูล pH เซนเซอร์เข้ากับ Arduino Mega 2560 และ NodeMCU v.1.0 ..... 61
4.5 วิธีการต่อชุดขับมอเตอร์เข้ากับ Arduino Mega 2560 ..... 61
4.6 ต่อไฟจากอะแดปเตอร์ 12 V เข้ากับตัวชุดขับมอเตอร์เพื่อให้ทำงาน ..... 62
4.7 ต่อไฟจากอะแดปเตอร์ 9 V เข้ากับตัว Arduino Mega 2560 เพื่อให้ทำงาน ..... 62
4.8 เชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo ..... 63
4.9 วิธีเข้า Web Server ..... 64
4.10 หน้าต่าง Web Server ที่ได้สร้างไว้ใน NodeMCU ..... 64

# สารบัญรูป (ต่อ)


รูปที่ หน้า

4.11 ค่า pH เปลี่ยนไป....65
4.12 กดปุ่ม Motor On ที่หน้าเว็บไซต์....66
4.13 มอเตอร์ทำงาน....66
4.14 กดปุ่ม Motor Off ที่หน้าเว็บไซต์....67
4.15 มอเตอร์หยุดทำงาน....67
4.16 กดปุ่ม Solenoid On ที่หน้าเว็บไซต์....68
4.17 วาล์วโซลินอยด์ทำงาน....68
4.18 กดปุ่ม Solenoid Off ที่หน้าเว็บไซต์....69
4.19 วาล์วโซลินอยด์หยุดทำงาน....69
4.20 กดปุ่ม All On ที่หน้าเว็บไซต์....70
4.21 มอเตอร์และวาล์วโซลินอยด์ทำงาน....70
4.22 กดปุ่ม All Off ที่หน้าเว็บไซต์....71
4.23 มอเตอร์และวาล์วโซลินอยด์หยุดทำงาน....71
ข.1 เว็บไซต์ www.arduino.cc....78
ข.2 เลือก Windows Installer....79
ข.3 เลือก JUST DOWNLOAD....79
ข.4 หน้าต่างแสดงขนาดของตัวติดตั้งโปรแกรม Arduino....80
ข.5 หน้าต่างข้อตกลง....81
ข.6 หน้าต่างลงโปรแกรม Arduino (1)....81
ข.7 หน้าต่างลงโปรแกรม Arduino (2)....82
ข.8 หน้าต่างลงโปรแกรม Arduino (3)....82
ข.9 หน้าต่างลงโปรแกรม Arduino (4)....83
ข.10 ไอคอนโปรแกรม Arduino....83
ค.1 เชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo....85
ค.2 วิธีเข้า Web Server....86
ค.3 หน้าต่าง Web Server ที่ได้สร้างไว้ใน NodeMCU....87

# สารบัญรูป


รูปที่ หน้า

จ.1 โมดูล pH เซนเซอร์, ชุดขับมอเตอร์, Arduino Mega 2560 และNodeMCU v.1.0.....102

จ.2 มอเตอร์และวาล์วโซลินอยด์.....102

จ.3 วิธีการต่อ Arduino Mega 2560 เข้ากับ NodeMCU v.1.0.....103

จ.4 วิธีการต่อโมดูล pH เซนเซอร์เข้ากับ Arduino Mega 2560 และ NodeMCU v.1.0.....103

จ.5 วิธีการต่อชุดขับมอเตอร์เข้ากับ Arduino Mega 2560.....104

จ.6 ต่อไฟจากอะแดปเตอร์ 12 V เข้ากับตัวชุดขับมอเตอร์เพื่อให้ทำงาน.....104

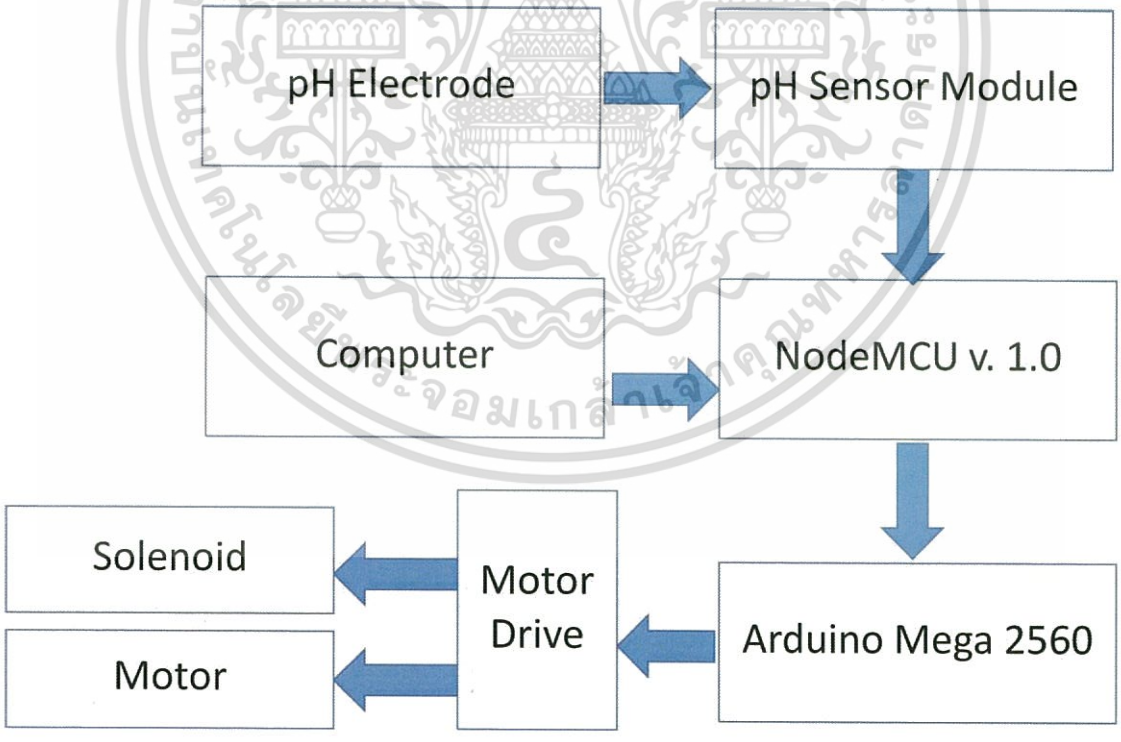
จ.7 ต่อไฟจากอะแดปเตอร์ 9 V เข้ากับตัว Arduino Mega 2560 เพื่อให้ทำงาน.....105

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มาของโครงงาน

เนื่องจากปัจจุบันมีผู้ทำอาชีพเพาะเลี้ยงสัตว์น้ำในพื้นที่ส่วนตัวหรือพื้นที่อื่นๆ เพื่อใช้ทำมาหากินเลี้ยงชีพแต่ส่วนมากจะพบปัญหา น้ำเน่าเสียซึ่งทำให้สัตว์น้ำตายเป็นจำนวนมากและสูญเสียระบบนิเวศซึ่งเกิดจากปัญหาดังกล่าว ทางคณะผู้จัดทำจึงมีแนวคิดว่าจะสร้างเทคโนโลยีการบำบัดน้ำเสียของบ่อเพาะเลี้ยงสัตว์น้ำมาช่วยแก้ปัญหาที่เกิดขึ้นนี้เพื่อให้ตรวจสอบค่าต่างๆ ของน้ำที่เป็นปัจจัยที่ทำให้เกิดน้ำเสีย และบำบัดน้ำที่เสียเหล่านั้นให้กลับมาเป็นน้ำที่สัตว์น้ำสามารถอยู่อาศัยได้ตามปกติ

หัววัดค่า pH จะทำการวัดค่าค่าแรงเคลื่อนไฟฟ้าจากน้ำแล้วส่งค่านั้นไปยังโมดูล pH เซนเซอร์เพื่อให้เปลี่ยนค่าแรงเคลื่อนไฟฟ้าเป็นค่า pH จากน้ำ แล้วนำค่า pH ได้มาประมวลผลผ่าน NodeMCU v.1.0 จากนั้นจะส่งค่า pH ไปยังคอมพิวเตอร์โดยผ่านทาง NodeMCU v.1.0 ดังรูปที่ 1.1 แล้วทำการแสดงผลค่า pH ที่วัดได้ ซึ่งคอมพิวเตอร์ยังสามารถสั่งให้เครื่องปั่นน้ำทำงานได้ ผ่านทางเว็บไซต์ที่ NodeMCU v.1.0 สร้างขึ้นมา




**รูปที่ 1.1** บล็อกไดอะแกรม

## 1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อตรวจสอบค่าความเป็นกรด-ด่าง, ค่าความต่างศักย์ไฟฟ้า, ค่าออกซิเจนที่ละลายในน้ำ, ค่าอุณหภูมิ ของบ่อเพาะเลี้ยงสัตว์น้ำ

2) เพื่อบำบัดน้ำเสียที่เกิดขึ้นในบ่อเพาะเลี้ยงสัตว์น้ำ

3) เพื่อรักษาสมดุลทางธรรมชาติของสัตว์น้ำ

4) เพื่อความรู้ที่ได้ศึกษามาประยุกต์ใช้และพัฒนาอุปกรณ์ให้มีประโยชน์และใช้งานได้สะดวกมากขึ้น

5) เพื่อศึกษาซอฟต์แวร์ และฮาร์ดแวร์ที่ใช้ในการทำโครงงาน

## 1.3 ขอบเขตของโครงงาน

1) ศึกษาการประยุกต์ใช้ Arduino ร่วมกับโมดูล pH เซนเซอร์และ NodeMCU เพื่อนำมาใช้ในการพัฒนาบ่อเพาะเลี้ยงสัตว์น้ำ

2) ออกแบบและวิเคราะห์เครื่องบำบัดน้ำเสียของบ่อเพาะเลี้ยงสัตว์น้ำ

3) เครื่องบำบัดน้ำเสียสามารถใช้ส่งค่าไปยังหน้าเว็บได้

4) ระบบมีความสามารถเพิ่มอื่นๆ เช่น สามารถแสดงค่า pH ที่ได้จากน้ำ สามารถสั่งงานเครื่องปั่นน้ำ เพื่อให้ปั่นในกรณีน้ำเสียโดยอัตโนมัติหรือกรณีผู้ใช้งานได้สั่งใช้งานเครื่องปั่นน้ำ เป็นต้น

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับประสบการณ์ ความรู้ ความเข้าใจเกี่ยวกับการพัฒนาโปรแกรมประยุกต์บนโปรแกรม

2) ศึกษาการใช้ Library Software Serial ที่อยู่ใน Arduino IDE เพื่อให้ Arduino และ NodeMCU สามารถพูดคุยสื่อสารกันได้

3) ระบบที่พัฒนาขึ้นเพื่อสามารถแสดงค่าที่ pH ที่วัดมาได้และสามารถไปใช้งานร่วมกับอย่างอื่นได้

## 1.5 อุปกรณ์ที่ต้องใช้

### 1.5.1 ฮาร์ดแวร์(Hardware)

- Arduino Mega 2560 จำนวน 1 ตัว
- ESP8266 NodeMCU v.1.0 จำนวน 1 ตัว
- โมดูล pH เซนเซอร์ จำนวน 1 ตัว
- หัววัดค่า pH (pH Electrode) จำนวน 1 ตัว
- มอเตอร์ DC(DC Motor) จำนวน 1 ตัว
- วาล์วโซลินอยด์(Solenoid Value) จำนวน 1 ตัว
- ชุดขับมอเตอร์(Motor Drive) จำนวน 1 ตัว
- อะแดปเตอร์(Adapter) 9 V จำนวน 1 ตัว
- อะแดปเตอร์ 12 V จำนวน 1 ตัว
- WP1250 SOBO Aquarium Power Head จำนวน 1 ตัว
- คอมพิวเตอร์ที่มี Wireless Adapter จำนวน 1 ตัว

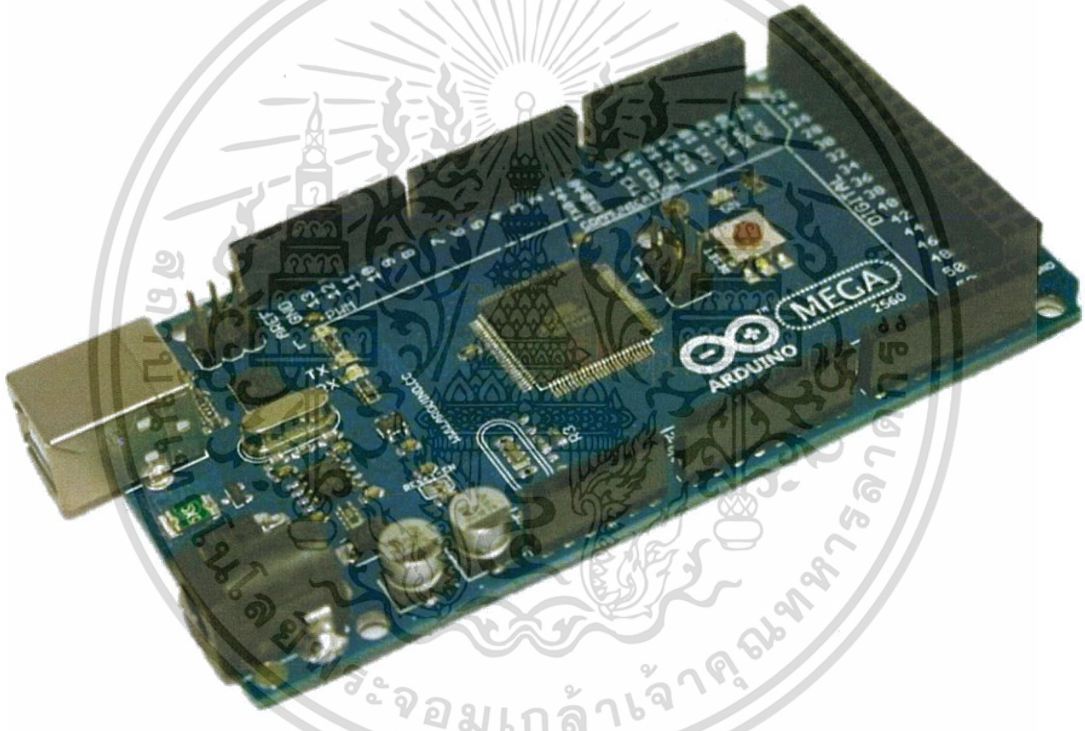
**1.5.2 ซอฟต์แวร์(Software)**

- Arduino IDE เป็นโปรแกรมสำหรับลงในบอร์ด Arduino Mega 2560 และ ESP8266 NodeMCU v.1.0 เพื่อใช้งาน

# บทที่ 2
# ทฤษฎีพื้นฐานที่ใช้

## 2.1 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์ Arduino Mega 2560

เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจากเซนเซอร์หรือควบคุมมอเตอร์เซอร์โว(Servo) หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน




**รูปที่ 2.1** Arduino Mega 2560 [1]

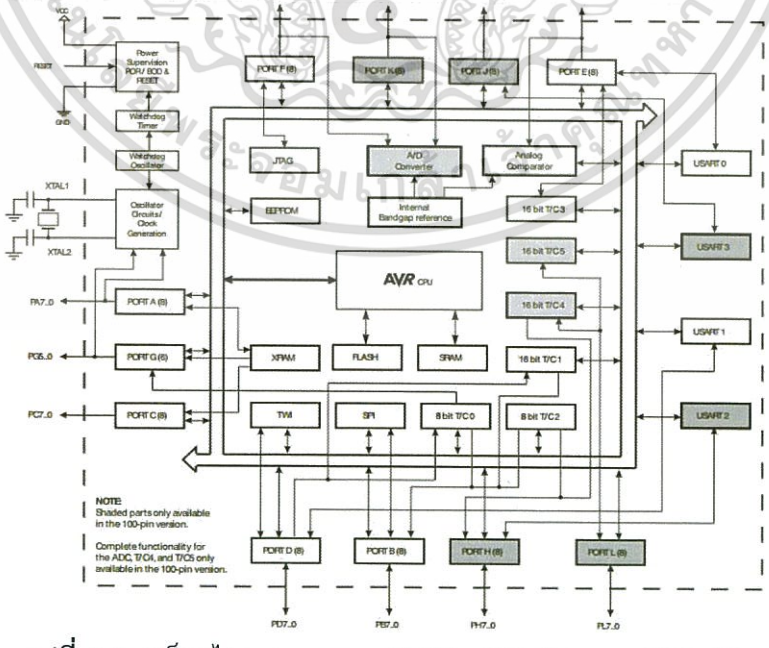
### 2.1.1 คุณสมบัติพื้นฐานของ Arduino Mega 2560

คุณสมบัติของ Arduino Mega 2560 ซึ่งจำเป็นต่อการใช้งาน โดยแสดงดังในตารางที่ 2.1

**ตารางที่ 2.1** คุณสมบัติพื้นฐานของ Arduino Mega 2560

| | |
|---|---|
| ไมโครคอนโทรลเลอร์ | ATmega 2560 |
| พลังงานไฟฟ้าที่ใช้ขณะทำงาน | 5 V |
| พลังงานไฟฟ้าที่รับเข้า (ขั้นต่ำ) | 7-12 V |
| พลังงานไฟฟ้าที่รับเข้า (แนะนำ) | 6-20 V |
| จำนวนขาของ Digital I/O | 54 (of which 15 provide PWM output) |
| จำนวนขาของ Analog Input | 16 ขา |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB (ATmega 2560) of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |
| ความยาวของอุปกรณ์ | 101.52 mm |
| ความกว้างของอุปกรณ์ | 53.3 mm |
| น้ำหนักของอุปกรณ์ | 37 g |

โดยมีไมโครคอนโทรลเลอร์ ATmega 2560 ซึ่งทำงานอยู่ใน Arduino Mega 2560 ดังรูปที่ 2.2



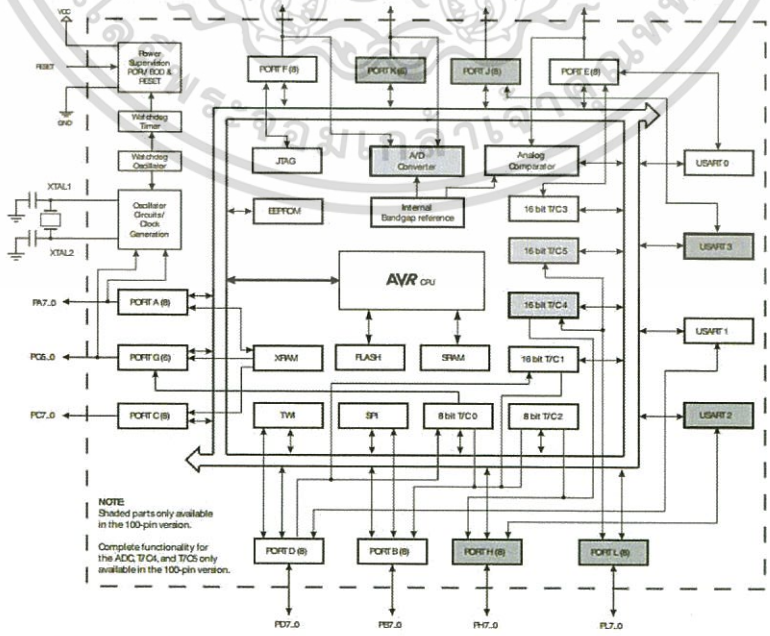

**รูปที่ 2.2** บล็อกไดอะแกรมของ MCU : AVR ATmega2560 [2]

### 2.1.2 คุณสมบัติพื้นฐานของ MCU

คุณสมบัติต่างๆ ของ MCU ซึ่งเป็นไมโครคอนโทรลเลอร์ยูนิต ที่อยู่ใน Arduino Mega 2560 มีดังตารางที่ 2.2

**ตารางที่ 2.2** คุณสมบัติพื้นฐานของ MCU

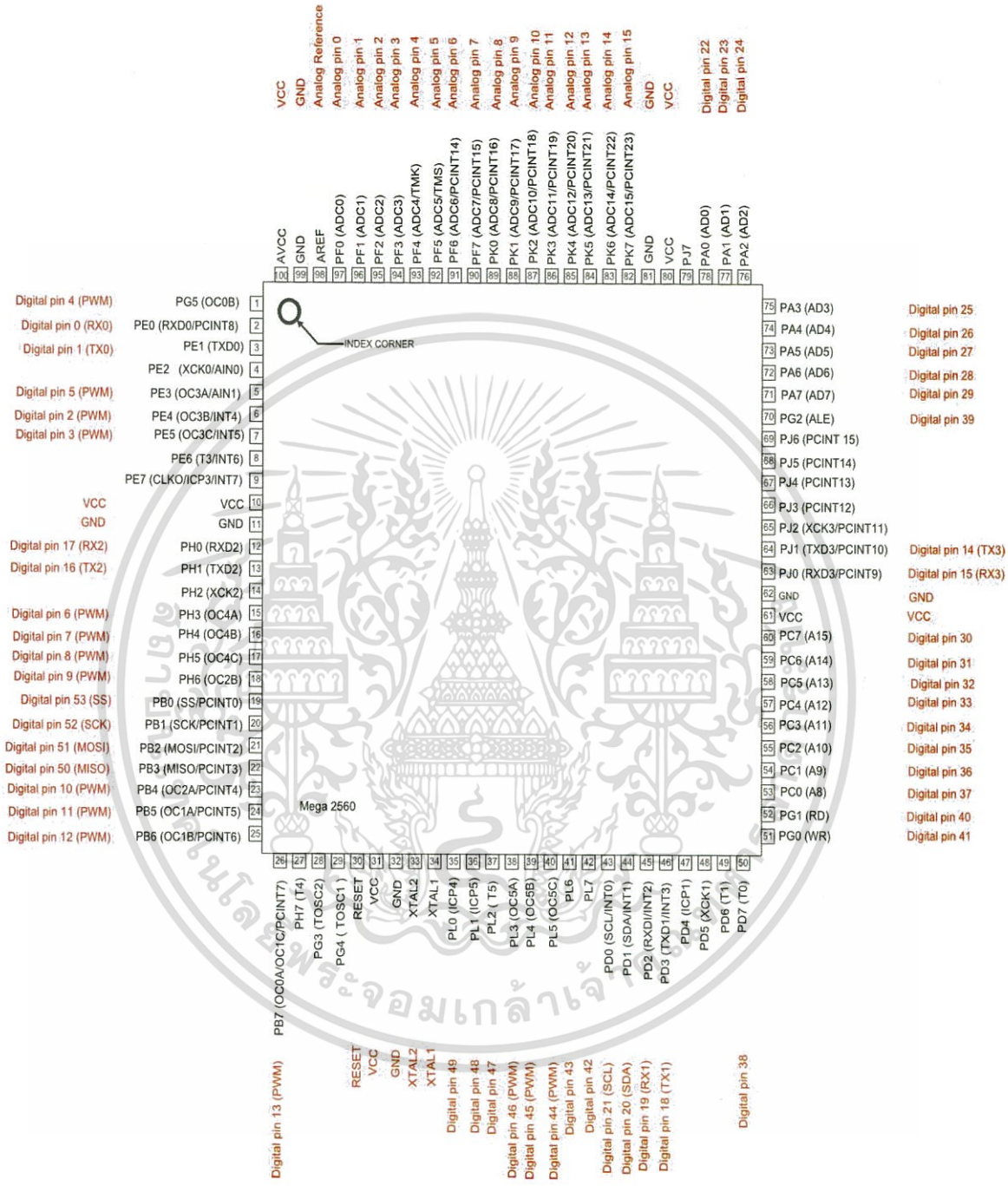
| | |
|---|---|
| Data Bus Width | 8 bit |
| Maximum Clock Frequency | 16 MHz |
| Program Memory Size | 256 kB |
| Data RAM Size | 8 kB |
| ADC Resolution | 10 bit |
| พลังงานไฟฟ้าที่ใช้ขณะทำงาน | 4.5 V to 5.5 V |
| อุณหภูมิสูงสุดระหว่างทำงาน | + 85 °C |
| อุณหภูมิต่ำสุดระหว่างทำงาน | - 40 °C |
| Data RAM Type | SRAM |
| Data ROM Size | 4 kB |
| Data ROM Type | EEPROM |
| Interface Type | 2-Wire, SPI, USART |
| จำนวนของ ADC Channels | 16 |
| จำนวนของ I/Os | 86 I/O |
| จำนวนของ Timers/Counters | 6 Timer |
| Supply Voltage - Max | 5.5 V |
| Supply Voltage - Min | 4.5 V |




**รูปที่ 2.2** บล็อกไดอะแกรมของ MCU : AVR ATmega2560 [2]

### 2.1.3 พอร์ตของ MCU

ใน MCU จะมีพอร์ตดังรูปที่ 2.3 และ PIN mapping table ดังตารางที่ 2.3




**รูปที่ 2.3** พอร์ตของ AVR ATmega 2560 [2]

**ตารางที่ 2.3** PIN mapping ของ MCU [3]

| Pin Number | Pin Name | Mapped Pin Name |
|---|---|---|
| 1 | PG5 ( OC0B ) | Digital pin 4 (PWM) |
| 2 | PE0 ( RXD0/PCINT8 ) | Digital pin 0 (RX0) |
| 3 | PE1 ( TXD0 ) | Digital pin 1 (TX0) |
| 4 | PE2 ( XCK0/AIN0 ) | |
| 5 | PE3 ( OC3A/AIN1 ) | Digital pin 5 (PWM) |
| 6 | PE4 ( OC3B/INT4 ) | Digital pin 2 (PWM) |
| 7 | PE5 ( OC3C/INT5 ) | Digital pin 3 (PWM) |
| 8 | PE6 ( T3/INT6 ) | |
| 9 | PE7 ( CLKO/ICP3/INT7 ) | |
| 10 | VCC | VCC |
| 11 | GND | GND |
| 12 | PH0 ( RXD2 ) | Digital pin 17 (RX2) |
| 13 | PH1 ( TXD2 ) | Digital pin 16 (TX2) |
| 14 | PH2 ( XCK2 ) | |
| 15 | PH3 ( OC4A ) | Digital pin 6 (PWM) |
| 16 | PH4 ( OC4B ) | Digital pin 7 (PWM) |
| 17 | PH5 ( OC4C ) | Digital pin 8 (PWM) |
| 18 | PH6 ( OC2B ) | Digital pin 9 (PWM) |
| 19 | PB0 ( SS/PCINT0 ) | Digital pin 53 (SS) |
| 20 | PB1 ( SCK/PCINT1 ) | Digital pin 52 (SCK) |
| 21 | PB2 ( MOSI/PCINT2 ) | Digital pin 51 (MOSI) |
| 22 | PB3 ( MISO/PCINT3 ) | Digital pin 50 (MISO) |
| 23 | PB4 ( OC2A/PCINT4 ) | Digital pin 10 (PWM) |
| 24 | PB5 ( OC1A/PCINT5 ) | Digital pin 11 (PWM) |
| 25 | PB6 ( OC1B/PCINT6 ) | Digital pin 12 (PWM) |
| 26 | PB7 ( OC0A/OC1C/PCINT7 ) | Digital pin 13 (PWM) |
| 27 | PH7 ( T4 ) | |
| 28 | PG3 ( TOSC2 ) | |
| 29 | PG4 ( TOSC1 ) | |
| 30 | RESET | RESET |
| 31 | VCC | VCC |
| 32 | GND | GND |
| 33 | XTAL2 | XTAL2 |
| 34 | XTAL1 | XTAL1 |
| 35 | PL0 ( ICP4 ) | Digital pin 49 |

**ตารางที่ 2.3** PIN mapping ของ MCU (ต่อ)

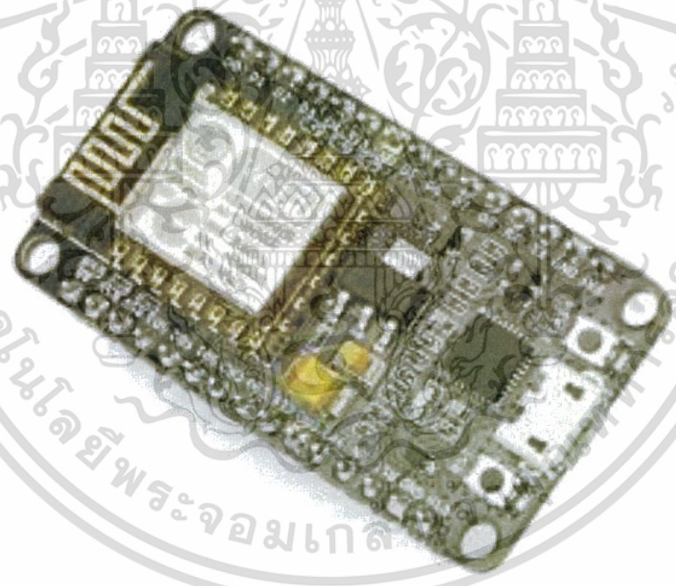
| Pin Number | Pin Name | Mapped Pin Name |
|---|---|---|
| 36 | PL1 ( ICP5 ) | Digital pin 48 |
| 37 | PL2 ( T5 ) | Digital pin 47 |
| 38 | PL3 ( OC5A ) | Digital pin 46 (PWM) |
| 39 | PL4 ( OC5B ) | Digital pin 45 (PWM) |
| 40 | PL5 ( OC5C ) | Digital pin 44 (PWM) |
| 41 | PL6 | Digital pin 43 |
| 42 | PL7 | Digital pin 42 |
| 43 | PD0 ( SCL/INT0 ) | Digital pin 21 (SCL) |
| 44 | PD1 ( SDA/INT1 ) | Digital pin 20 (SDA) |
| 45 | PD2 ( RXDI/INT2 ) | Digital pin 19 (RX1) |
| 46 | PD3 ( TXD1/INT3 ) | Digital pin 18 (TX1) |
| 47 | PD4 ( ICP1 ) | |
| 48 | PD5 ( XCK1 ) | |
| 49 | PD6 ( T1 ) | |
| 50 | PD7 ( T0 ) | Digital pin 38 |
| 51 | PG0 ( WR ) | Digital pin 41 |
| 52 | PG1 ( RD ) | Digital pin 40 |
| 53 | PC0 ( A8 ) | Digital pin 37 |
| 54 | PC1 ( A9 ) | Digital pin 36 |
| 55 | PC2 ( A10 ) | Digital pin 35 |
| 56 | PC3 ( A11 ) | Digital pin 34 |
| 57 | PC4 ( A12 ) | Digital pin 33 |
| 58 | PC5 ( A13 ) | Digital pin 32 |
| 59 | PC6 ( A14 ) | Digital pin 31 |
| 60 | PC7 ( A15 ) | Digital pin 30 |
| 61 | VCC | VCC |
| 62 | GND | GND |
| 63 | PJ0 ( RXD3/PCINT9 ) | Digital pin 15 (RX3) |
| 64 | PJ1 ( TXD3/PCINT10 ) | Digital pin 14 (TX3) |
| 65 | PJ2 ( XCK3/PCINT11 ) | |
| 66 | PJ3 ( PCINT12 ) | |
| 67 | PJ4 ( PCINT13 ) | |
| 68 | PJ5 ( PCINT14 ) | |
| 69 | PJ6 ( PCINT 15 ) | |
| 70 | PG2 ( ALE ) | Digital pin 39 |

**ตารางที่ 2.3** PIN mapping ของ MCU (ต่อ)

| Pin Number | Pin Name | Mapped Pin Name |
|---|---|---|
| 71 | PA7 ( AD7 ) | Digital pin 29 |
| 72 | PA6 ( AD6 ) | Digital pin 28 |
| 73 | PA5 ( AD5 ) | Digital pin 27 |
| 74 | PA4 ( AD4 ) | Digital pin 26 |
| 75 | PA3 ( AD3 ) | Digital pin 25 |
| 76 | PA2 ( AD2 ) | Digital pin 24 |
| 77 | PA1 ( AD1 ) | Digital pin 23 |
| 78 | PA0 ( AD0 ) | Digital pin 22 |
| 79 | PJ7 | |
| 80 | VCC | VCC |
| 81 | GND | GND |
| 82 | PK7 ( ADC15/PCINT23 ) | Analog pin 15 |
| 83 | PK6 ( ADC14/PCINT22 ) | Analog pin 14 |
| 84 | PK5 ( ADC13/PCINT21 ) | Analog pin 13 |
| 85 | PK4 ( ADC12/PCINT20 ) | Analog pin 12 |
| 86 | PK3 ( ADC11/PCINT19 ) | Analog pin 11 |
| 87 | PK2 ( ADC10/PCINT18 ) | Analog pin 10 |
| 88 | PK1 ( ADC9/PCINT17 ) | Analog pin 9 |
| 89 | PK0 ( ADC8/PCINT16 ) | Analog pin 8 |
| 90 | PF7 ( ADC7 ) | Analog pin 7 |
| 91 | PF6 ( ADC6 ) | Analog pin 6 |
| 92 | PF5 ( ADC5/TMS ) | Analog pin 5 |
| 93 | PF4 ( ADC4/TMK ) | Analog pin 4 |
| 94 | PF3 ( ADC3 ) | Analog pin 3 |
| 95 | PF2 ( ADC2 ) | Analog pin 2 |
| 96 | PF1 ( ADC1 ) | Analog pin 1 |
| 97 | PF0 ( ADC0 ) | Analog pin 0 |
| 98 | AREF | Analog Reference |
| 99 | GND | GND |
| 100 | AVCC | VCC |

## 2.2 ทฤษฎีเกี่ยวกับ ESP8266 NodeMCU v.1.0

NodeMCU v.1.0 เป็นแพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น open source สามารถเขียนโปรแกรมด้วยภาษา Lau ได้ ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล Wi-Fi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเตอร์เน็ตนั่นเอง ตัวโมดูล ESP8266 นั้นมีอยู่ด้วยกันหลายรุ่นมาก ตั้งแต่เวอร์ชันแรกที่เป็น ESP-01 ไล่ไปเรื่อยๆจนปัจจุบันมีถึง ESP-12 แล้ว และที่ฝังอยู่ใน NodeMCU v.1.0 นั้นก็เป็น ESP-12E NodeMCU นั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output build in มาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ และเมื่อไม่นานมานี้ก็มีนักพัฒนาที่สามารถทำให้ Arduino IDE ใช้งานร่วมกับ NodeMCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้เราสามารถใช้งานมันได้หลากหลายมากยิ่งขึ้น NodeMCU ตัวนี้สามารถทำอะไรได้หลายอย่างมาก โดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่าน Wi-Fi และอื่นๆ อีกมากมาย ดังรูปที่ 2.4



**รูปที่ 2.4** ESP8266 NodeMCU v.1.0 [4]

### 2.2.1 คุณสมบัติพื้นฐานของ ESP8266 NodeMCU v.1.0

โดย ESP8266 NodeMCU v.1.0 จะมีคุณสมบัติดังนี้

-ใช้โมดูล ESP8266-12E ที่ภายในมีไมโครคอนโทรลเลอร์ 32 บิต หน่วยความจำแบบแฟลชความจุ 4 MB และโมดูล Wi-Fi ในตัว

-มีชิป CP2102 สำหรับแปลงสัญญาณพอร์ต USB เป็น UART เพื่อเชื่อมต่อคอมพิวเตอร์สำหรับโปรแกรม Firmware

- ใช้ไฟเลี้ยงภายนอก 5 V มีวงจรควบคุมแรงดันไฟเลี้ยงสำหรับอุปกรณ์ 3.3 V กระแสไฟฟ้าสูงสุด 800 mA

- มีขาพอร์ต SPI สำหรับติดต่อกับ SD การ์ด

- มีสวิตช์ RESET และ Flash สำหรับโปรแกรมเฟิร์มแวร์ใหม่

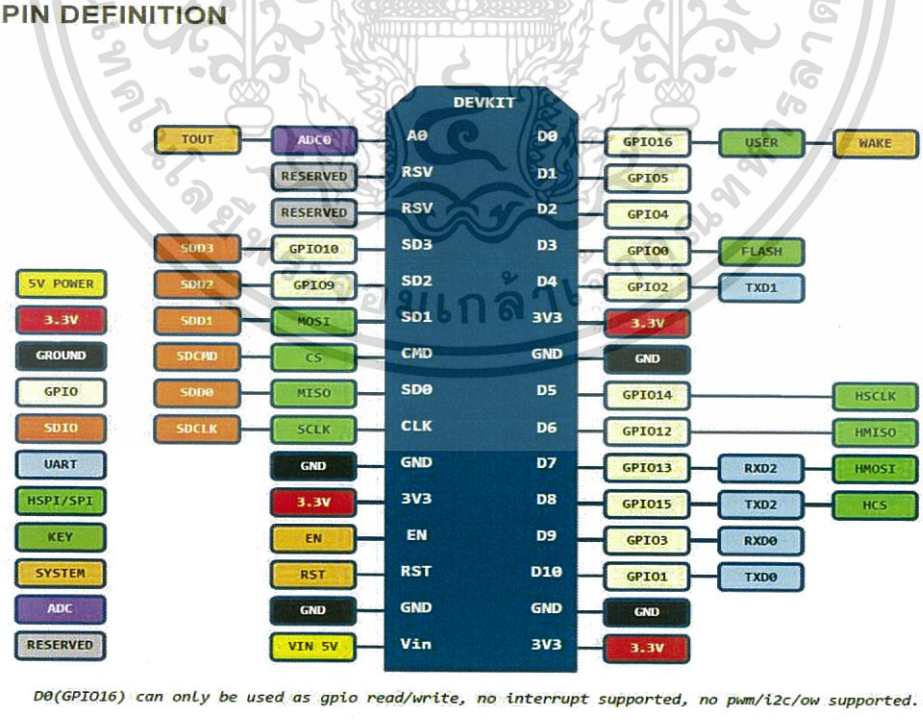
- มีอินพุตเอาต์พุตดิจิตอล (ลอจิก 3.3 V) รวม 16 ขา

- มีอินพุตอนาล็อก 1 ช่อง รับแรงดันไฟตรง 0 ถึง 1 V เข้าสู่วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ความละเอียด 10 bit

- เสียบลงบนเบรดบอร์ดเพื่อทำการทดลองได้ทันที หรือนำไปติดตั้งบนแผงวงจรประยุกต์ที่ออกแบบขึ้นเองได้สะดวก

### 2.2.2 คำอธิบายพอร์ตของ ESP8266 NodeMCU v.1.0

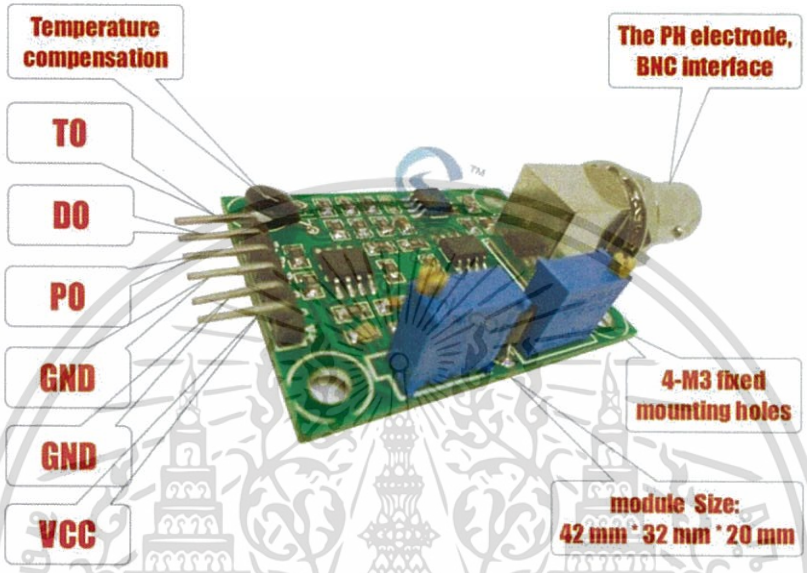
ESP8266 NodeMCU v.1.0 นั้นจะมีพอร์ตดังรูปที่ 2.5 จะมีพอร์ตต่างๆให้ใช้งานเหมือนกับ Arduino ทั่วๆไป คือ Analog และ Digital




**รูปที่ 2.5** พอร์ตของ ESP8266 NodeMCU v.1.0

## 2.3 ทฤษฎีเกี่ยวกับโมดูล pH เซนเซอร์ (pH Sensor Module)

โมดูล pH เซนเซอร์เป็นตัวแปลงค่าแรงเคลื่อนไฟฟ้าจาก หัววัดค่า pHที่ต่ออยู่ให้กลายเป็นค่า pH และจะส่งไปยังอุปกรณ์ที่เชื่อมต่อกับโมดูล pH เซนเซอร์. โมดูล pH เซนเซอร์จะมีส่วนประกอบดังรูปที่ 2.6




**รูปที่ 2.6** ส่วนประกอบของโมดูล pH เซนเซอร์

### 2.3.1 คุณสมบัติพื้นฐานของโมดูล pH เซนเซอร์

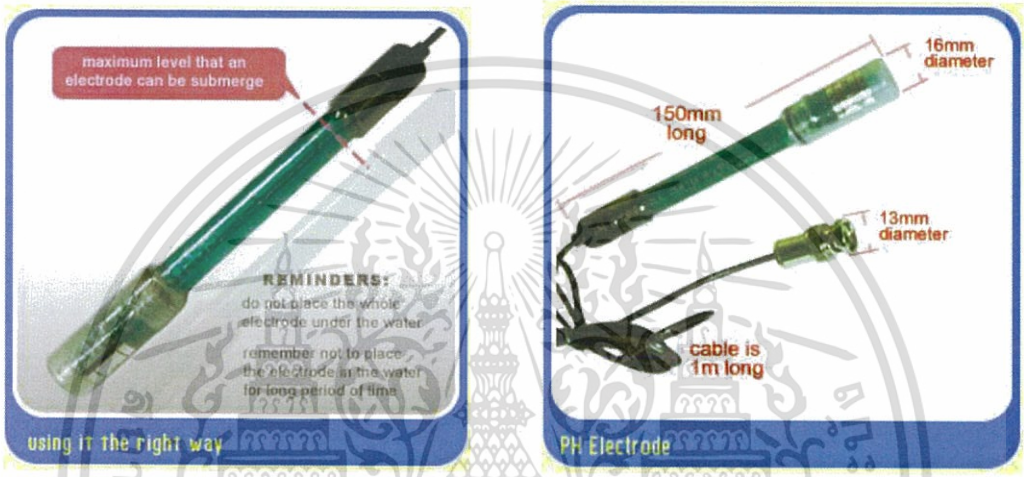
คุณสมบัติของโมดูล pH เซนเซอร์จะแสดงดังในตารางที่ 2.4

**ตารางที่ 2.4** คุณสมบัติพื้นฐานของโมดูล pH เซนเซอร์

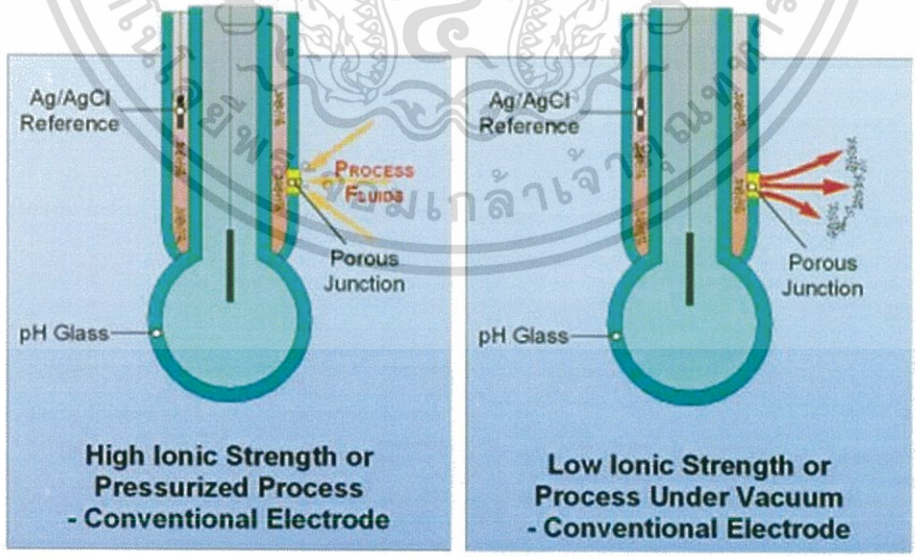
| | |
|---|---|
| ขนาดของแรงดันไฟฟ้า | 5 + 0.2 V (AC · DC) |
| ปริมาณกระแสไฟฟ้าที่ไหลผ่าน | 5-10 mA |
| ช่วงความเข้มข้นของการตรวจสอบวัด | 0-14 pH |
| ช่วงอุณหภูมิของการตรวจวัด | 0-80 °C |
| เวลาตอบสนอง | <5 S |
| เวลาเสถียรภาพ | <60 S |
| การใช้พลังงานส่วนประกอบ | <0.5 W |
| อุณหภูมิระหว่างการทำงาน | -10 ~ 50 °C (อุณหภูมิปกติ 20 °C) |
| ความชื้นระหว่างการทำงาน | 95% RH (ความชื้นปกติ 65% RH) |
| อายุการใช้งาน | 3 ปี |
| ขนาด | 42 mm × 32 mm × 20 mm |
| น้ำหนัก | 25 g |

## 2.4 ทฤษฎีเกี่ยวกับหัววัดค่า pH (pH Electrode with BNC plug GX-E1)

โครงสร้างทั่วไปของหัววัดค่า pH ประกอบด้วยขั้วไฟฟ้าอ้างอิง (reference electrode)กับขั้วไฟฟ้าที่ใช้วัดความต่างศักย์เทียบกับขั้วไฟฟ้าอ้างอิง (indicator electrode หรือ working electrode) โดยอุปกรณ์จะเป็นดังรูปที่ 2.7 การออกแบบในยุคแรกนั้นจะแยกขั้วไฟฟ้าทั้งสองออกเป็นขั้วไฟฟ้าสองชิ้นซึ่งไม่ค่อยจะพบเห็นกันแล้วในปัจจุบัน รูปแบบที่ใช้กันในปัจจุบันเป็นแบบที่เรียกว่า combined electrode กล่าวคือเป็นชนิดที่รวมขั้วไฟฟ้าสองขั้วไว้ใน probe ตัวเดียวกัน ดังแสดงในรูปที่ 2.8




**รูปที่ 2.7** หัววัดค่า pH




**รูปที่ 2.8** โครงสร้างของ combined electrode ที่ใช้สำหรับวัดค่า pH

### 2.4.1 คุณสมบัติพื้นฐานของหัววัดค่า pH

คุณสมบัติของหัววัดค่า pH จะแสดงดังในตารางที่ 2.5

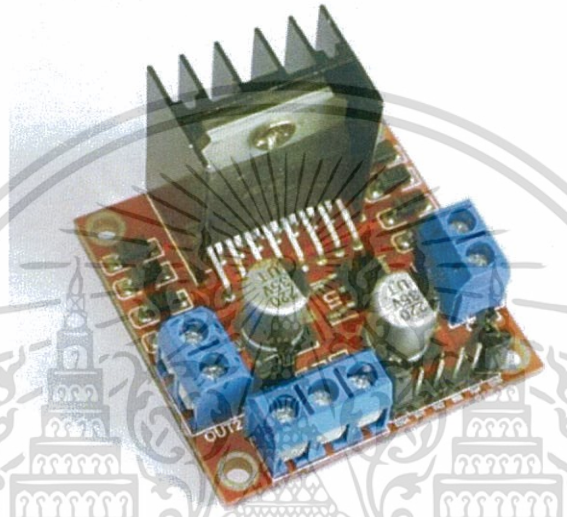
**ตารางที่ 2.5** คุณสมบัติพื้นฐานของหัววัดค่า pH

| | |
|---|---|
| ค่าที่วัดได้ | 0.00~14.00 pH |
| Resolution | 0.01 pH |
| ความแม่นยำ | 0.01 pH |
| อุณหภูมิระหว่างเครื่องทำงาน | 0 °C ~ 50 °C |
| Automatic Temperature Compensation (ATC) | 0 °C ~ 50 °C |
| ความยาวของสาย | 1 m |
| Connection | BNC socket |

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## 2.5 ทฤษฎีเกี่ยวกับชุดขับมอเตอร์ (L298 Dual H-Bridge Motor Driver)

L298 Dual H-Bridge Motor Driver(ชุดขับมอเตอร์) สามารถขับมอเตอร์ได้ 2 ชุด และสามารถขับกระแสได้สูงสุด 4 A ใช้แบบบริดจ์ 2 ช่อง ภายในบอร์ดได้ต่อไดโอดเพื่อกันไฟที่จะเข้ามารบกวนระบบเรียบร้อยแล้ว และยังมีแหล่งจ่ายไฟ 5 V ในตัว ในกรณีท่านจะนำไปต่อแบบแยกกราวด์ไม่ต้องไปหาแหล่งจ่ายเพิ่ม และยังสามารถควบคุมความเร็วมอเตอร์ได้ ใช้ไฟ 5 V สามารถรับไฟเข้า 7-35 V ได้ มีวงจรเรกูเลตในตัว ขับกระแสสูงสุดได้ 2 A



**รูปที่ 2.9** ชุดขับมอเตอร์

### 2.5.1 คุณสมบัติพื้นฐานของชุดขับมอเตอร์

คุณสมบัติของชุดขับมอเตอร์จะแสดงดังในตารางที่ 2.6

**ตารางที่ 2.6** คุณสมบัติพื้นฐานของชุดขับมอเตอร์

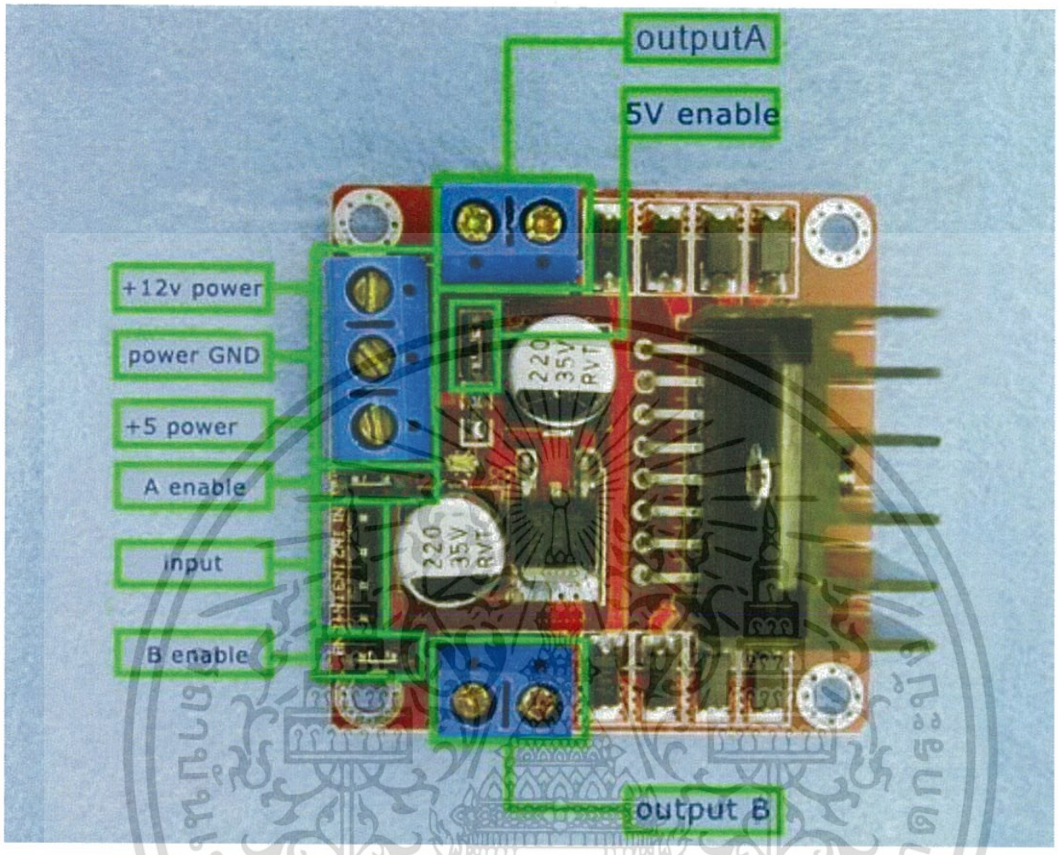
| | |
|---|---|
| พลังงานไฟฟ้าที่ใช้ขณะทำงาน | +5 V ~ +46 V |
| Driver peak current | 2 A |
| Logic power output Vss | +5 ~ +7 V (internal supply +5 V) |
| Logic current | 0~36 mA |
| Controlling level | Low -0.3 V ~ 1.5 V, high: 2.3 V ~ Vss |
| Enable signal | Low -0.3 V ~ 1.5 V, high: 2.3V ~ Vss |
| Max drive power | 25 W (Temperature 75 °C) |
| อุณหภูมิขณะทำงาน | -25 °C ~ +130 °C |
| ขนาดของอุปกรณ์ | 60 mm*54 mm |
| น้ำหนักของอุปกรณ์ | ~48 g |

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



### 2.5.2 พอร์ตของชุดขับมอเตอร์

พอร์ตต่างๆของชุดขับมอเตอร์จะแสดงดังรูปที่ 2.10




**รูปที่ 2.10** พอร์ตของ L298 Dual H-Bridge Motor Driver

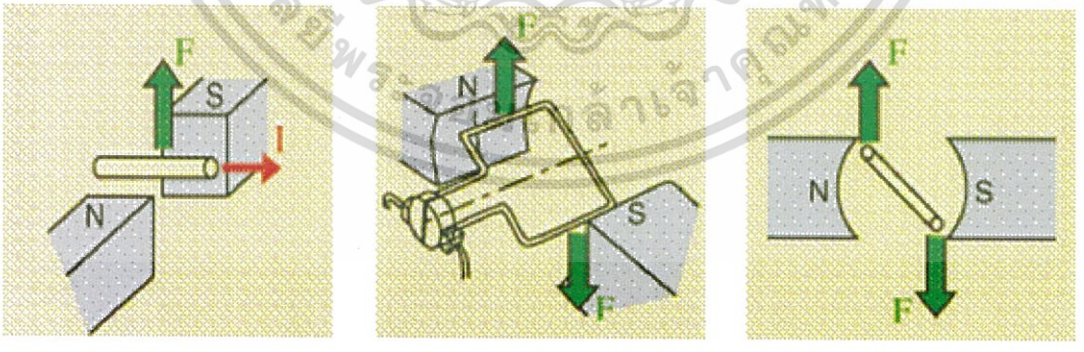
144360

## 2.6 ทฤษฎีเกี่ยวกับมอเตอร์กระแสตรง (DC Motor) รุ่น CL-RS445PA-15200

มอเตอร์ไฟฟ้าเป็นอุปกรณ์ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังกลมอเตอร์ที่ใช้งานในปัจจุบัน แต่ละชนิดก็จะมีคุณสมบัติที่แตกต่างออกไปต้องการความเร็ว รอบหรือกำลังงานที่แตกต่างกัน ในที่นี้ใช้มอเตอร์กระแสตรง รุ่น CL-RS445PA-15200 เพื่อใช้งานร่วมกับใบพัดเพื่อปั่นน้ำ โดยจะเป็นดังรูปที่ 2.11

**รูปที่ 2.11** มอเตอร์กระแสตรง รุ่น CL-RS445PA-15200

มอเตอร์ไฟฟ้ากระแสตรงประกอบด้วย แม่เหล็กถาวร 2 ขั้ววางอยู่ระหว่างขดลวดตัวนำ ขดลวดตำนำจะได้รับแรงดันไฟตรงป้อนให้ในการทำงาน ทำให้เกิดอำนาจแม่เหล็ก 2 ชุด มีขั้วแม่เหล็กเหมือนกันวางใกล้กัน เกิดแรงผลักดันทำให้ขดลวดตัวนำหมุนเคลื่อนที่ได้ การทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง แสดงดังรูปที่ 2.12




**รูปที่ 2.12** หลักการทำงานของมอเตอร์ [4]

### 2.6.1 คุณสมบัติพื้นฐานของมอเตอร์กระแสตรง รุ่น CL-RS445PA-15200

คุณสมบัติของมอเตอร์กระแสตรง รุ่น CL-RS445PA-15200 จะแสดงดังในตารางที่ 2.7

**ตารางที่ 2.7** คุณสมบัติพื้นฐานของมอเตอร์กระแสตรง รุ่น CL-RS445PA-15200

| | |
|---|---|
| Model Number | CL-RS445PA-15200 |
| Type | Micro Motor |
| Torque | 148 g.cm |
| ความเร็วรอบต่อนาที | 7600 rpm/min |
| Continuous Current | 0.067 A |
| Output Power | 9.77 W |
| พลังงานไฟฟ้า | 42 V |

## 2.7 ทฤษฎีเกี่ยวกับวาล์วโซลินอยด์ (Solenoid Value) รุ่น Uni-D UD-8

วาล์วโซลินอยด์ประกอบด้วยแม่เหล็กไฟฟ้าสำหรับทำหน้าที่ปิดเปิดวาล์วเมื่อเปิดและปิดสวิทช์ เมื่อกระแสไหลผ่านขดลวดแม่เหล็กไฟฟ้า สนามแม่เหล็กที่เกิดขึ้นจะดูดเดือยวาล์วเพื่อเปิดวาล์ว และเมื่อปิดสวิทช์ตัดกระแสไฟฟ้าเดือยวาล์วจะกลับไปสู่ตำแหน่งเดิมโดยน้ำหนักของตัวเองเพื่อปิดวาล์ว โดยจะเป็นดังรูปที่ 2.13

**รูปที่ 2.13** วาล์วโซลินอยด์

### 2.7.1 คุณสมบัติพื้นฐานของวาล์วโซลินอยด์ รุ่น Uni-D UD-8

คุณสมบัติของวาล์วโซลินอยด์ รุ่น Uni-D UD-8 จะแสดงดังในตารางที่ 2.8

**ตารางที่ 2.8** คุณสมบัติพื้นฐานของวาล์วโซลินอยด์ รุ่น Uni-D UD-8

| | |
|---|---|
| Model Number | UD-8 |
| ขนาดของท่อ | 1/4" |
| CV | 0.58 |
| Orifice | 4 mm |
| อุณหภูมิของของเหลวที่สามารถไหลผ่านได้ | -5 °C ~ +80 °C |
| Water Operation Perssure kgf/cm$^3$ | 0 ~ 3 bar |
| Air Operation Perssure kgf/cm$^3$ | 0 ~ 3 bar |
| Light oil Perssure kgf/cm$^3$ | 0 ~ 3 bar |
| Gas Perssure kgf/cm$^3$ | 0 ~ 3 bar |
| Vacuum | 0 ~ $10^{-3}$ Torr |
| ขนาดของอุปกรณ์ (L x W x H) | 41 mm x66 mm x21.5 mm |
| น้ำหนักของอุปกรณ์ | 310 g |

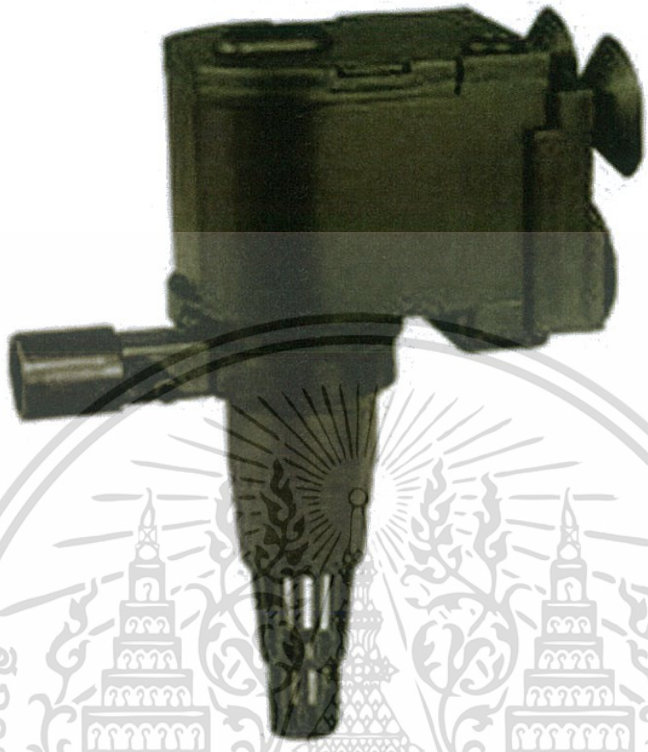
## 2.8 ทฤษฎีเกี่ยวกับอะแดปเตอร์กระแสสลับ (AC Adaptor)

Adaptor หรือ อะแดปเตอร์ คือหม้อแปลงไฟฟ้าจากไฟฟ้ากระแสสลับ (AC) ที่ใช้ทั่วๆ ไปตามบ้าน ที่มีความต่างศักย์ 220 V ให้เป็นไฟฟ้ากระแสตรง (DC) ที่มีความต่างศักย์ต่ำลง เพื่อให้สามารถจ่ายกระแสไฟฟ้ากับเครื่องใช้ไฟฟ้าได้ ซึ่งในที่นี้เราได้ใช้อะแดปเตอร์ 9 V จ่ายไฟให้ Arduino Mega 2560 และอะแดปเตอร์ 12 V จ่ายไฟให้ชุดขับมอเตอร์ทำงานซึ่งชุดขับมอเตอร์จะส่งไฟเข้ามอเตอร์กระแสตรงและวาล์วโซลินอยด์อีกที โดยจะเป็นดังรูปที่ 2.14

**รูปที่ 2.14** อะแดปเตอร์กระแสสลับ

## 2.9 ทฤษฎีเกี่ยวกับ WP1250 SOBO Aquarium Power Head

เป็นตัวปั้มลมเพื่อลำเอียงน้ำขึ้น โดยจะเป็นดังรูปที่ 2.15



**รูปที่ 2.15** WP1250 SOBO Aquarium Power Head

### 2.9.1 คุณสมบัติพื้นฐานของ WP1250 SOBO Aquarium Power Head 800L/H

คุณสมบัติของ WP1250 SOBO Aquarium Power Head 800L/H จะแสดงดังในตารางที่ 2.9

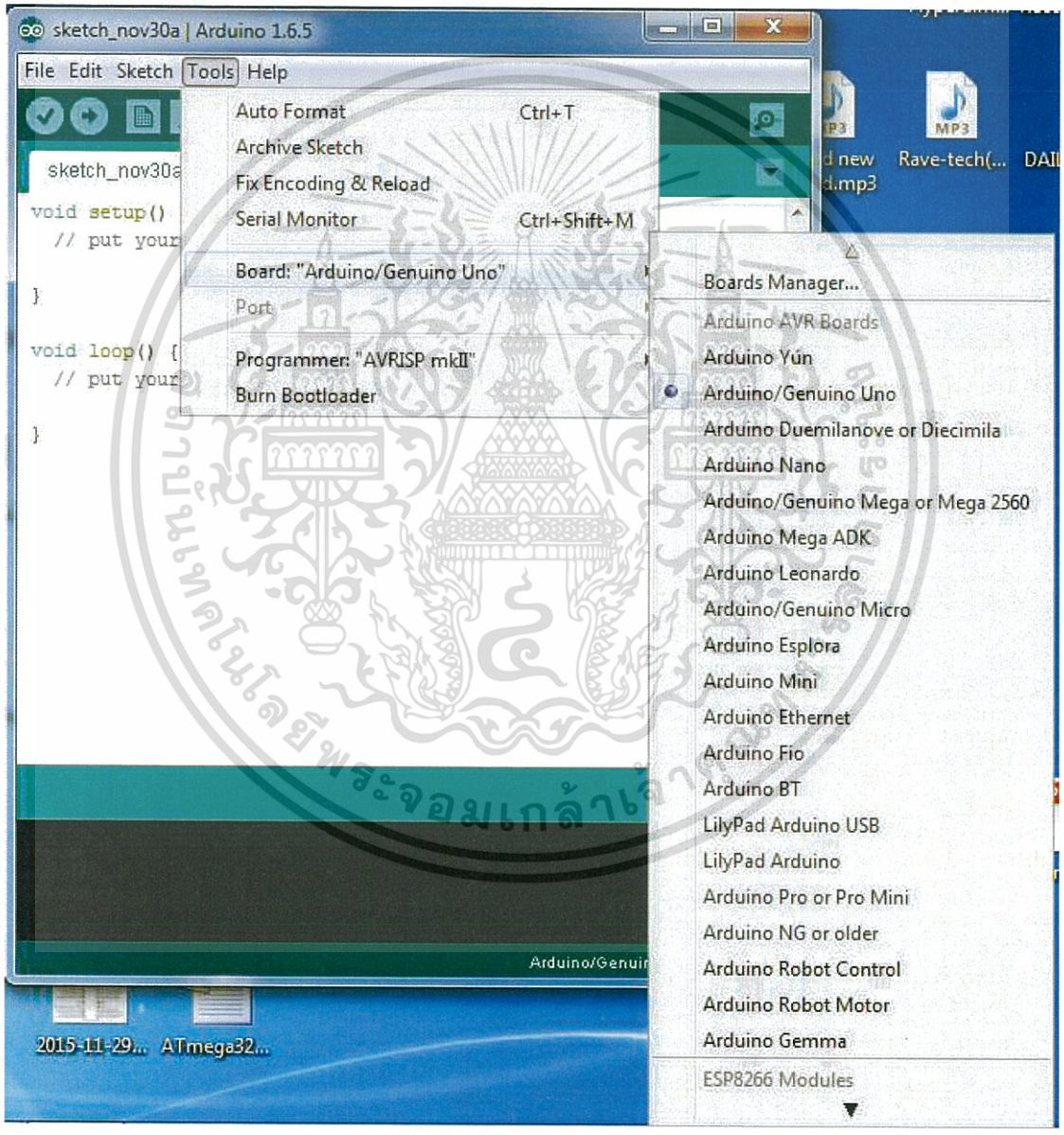
**ตารางที่ 2.9** คุณสมบัติพื้นฐานของ WP1250 SOBO Aquarium Power Head 800L/H

| | |
|---|---|
| พลังงานไฟฟ้าที่ใช้ | 220V - 240 V & 50 Hz 1/4" |
| กำลังไฟฟ้าที่ใช้ | 10 W |
| จำนวนน้ำที่ไหลออกได้ | 800L/H |
| ความสูงของน้ำที่ไหลออกได้ | 0.8 M |
| Water Flow Adjuster | N/A |
| ขนาดของอุปกรณ์ (L x W x H) | 10.5 cm x 5 cm x 13 cm |
| ขนาดของตู้น้ำที่ใช้ได้ | Up to 260 L |

## 2.10 โปรแกรมที่เกี่ยวข้อง

### 2.10.1 Arduino IDE

Arduino IDE คือโปรแกรมที่เอาไว้ใช้ในการเชื่อมต่อกับบอร์ด Arduino ในการใช้งานในการเขียนชุดคำสั่งในการใช้บอร์ด Arduino ในคำสั่งต่างๆ ซึ่งใช้งานได้ทั้งบนระบบปฏิบัติการ Window (XP, Vista, 7, 8, 10) ทั้ง 32 และ 64 บิต Mac OS X และ Linux และเป็นอิสระจากการทำงานของ OS (Operating System) ทุกชนิด ดังรูปที่ 2.16

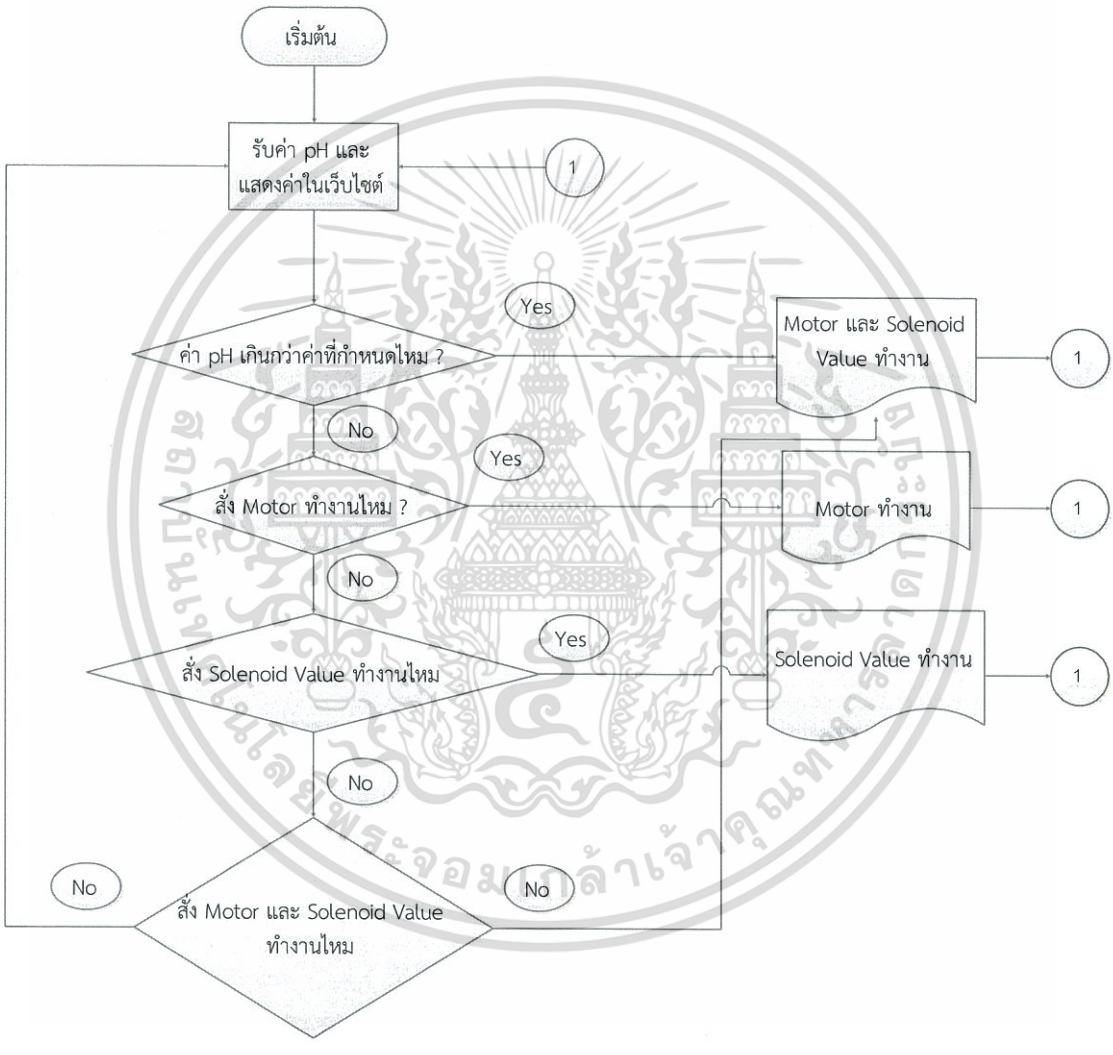



**รูปที่ 2.16** การเลือกบอร์ดที่ทำการเชื่อมต่อ และต้องการเขียนชุดคำสั่ง

# บทที่ 3
# การวิเคราะห์และออกแบบระบบ

## 3.1 การออกแบบระบบ

การทำงานของอุปกรณ์นั้นจะเป็นตามรูปที่ 3.1

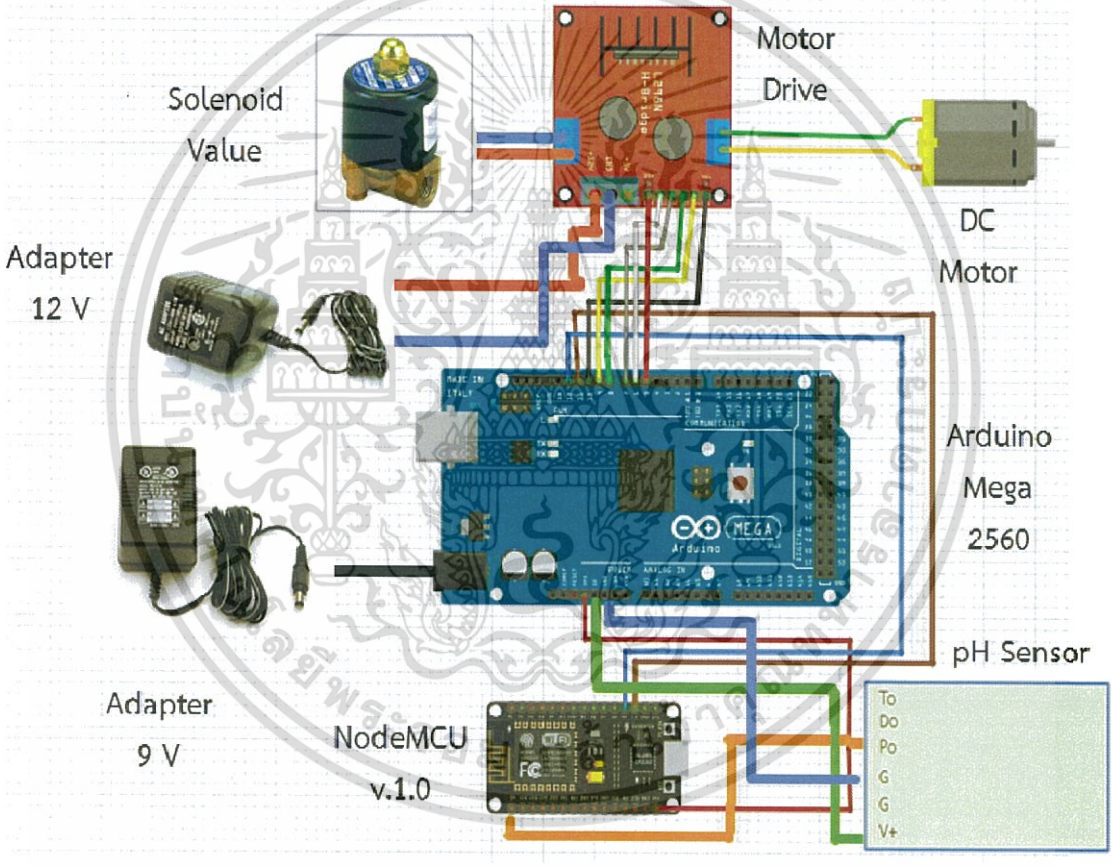



**รูปที่ 3.1** โฟลว์ชาร์ตการทำงานของอุปกรณ์

## 3.2 การออกแบบฮาร์ดแวร์ (Hardware)

การออกแบบในส่วนของฮาร์ดแวร์ จะใช้หัววัดค่า pH ที่เชื่อมต่อโมดูล pH เซนเซอร์ ซึ่งทำหน้าที่เป็นตัวแปลงค่าแรงเคลื่อนไฟฟ้าของหัววัดค่า pH เพื่อแปลงเป็นค่า pH แล้วส่งค่าไปยังไมโครคอนโทรลเลอร์ Arduino Mega 2560 โดยที่โมดูล pH เซนเซอร์สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ Arduino Mega 2560 และใช้ ไมโครคอนโทรลเลอร์ ESP8266 NodeMCU v.1.0 ในการส่งข้อมูลไปที่คอมพิวเตอร์

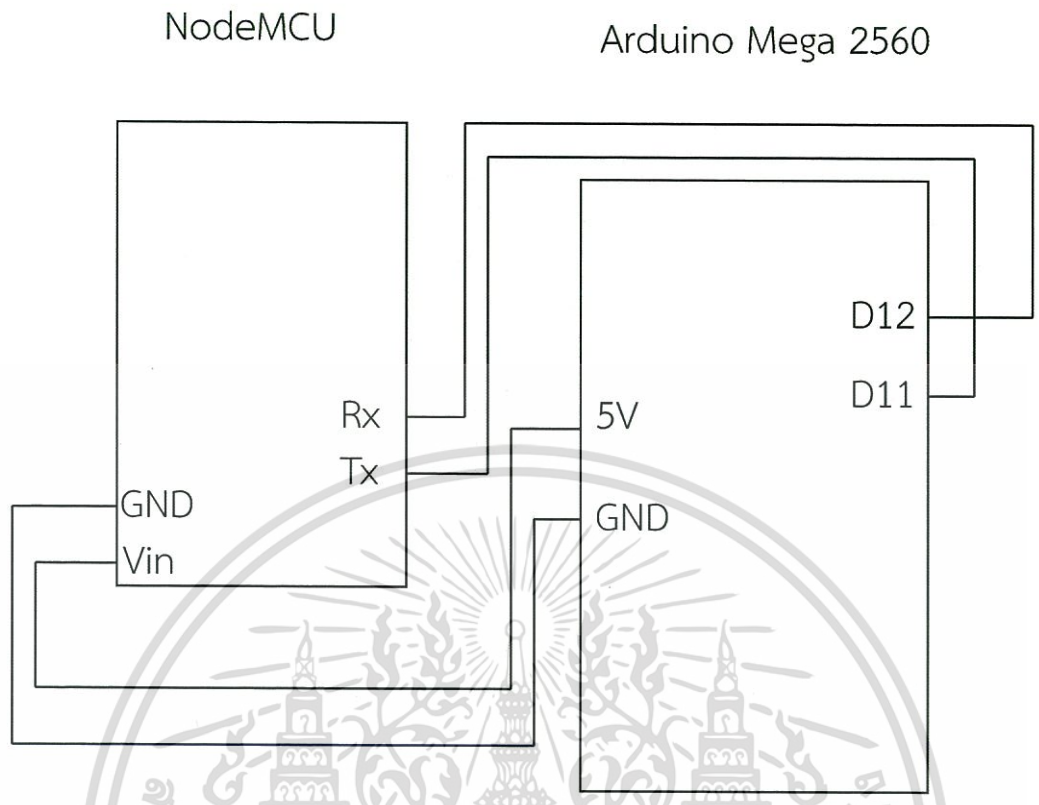
### 3.2.1 ขั้นตอนในการต่ออุปกรณ์ฮาร์ดแวร์



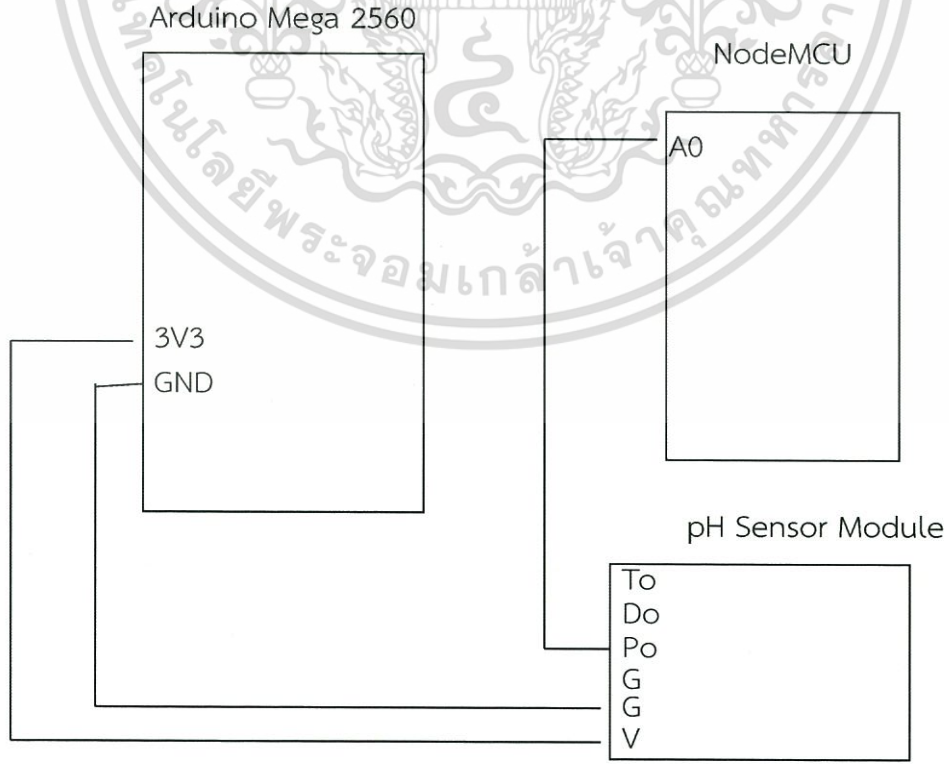

**รูปที่ 3.2** วงจรที่ออกแบบไว้

**อธิบายวงจร**

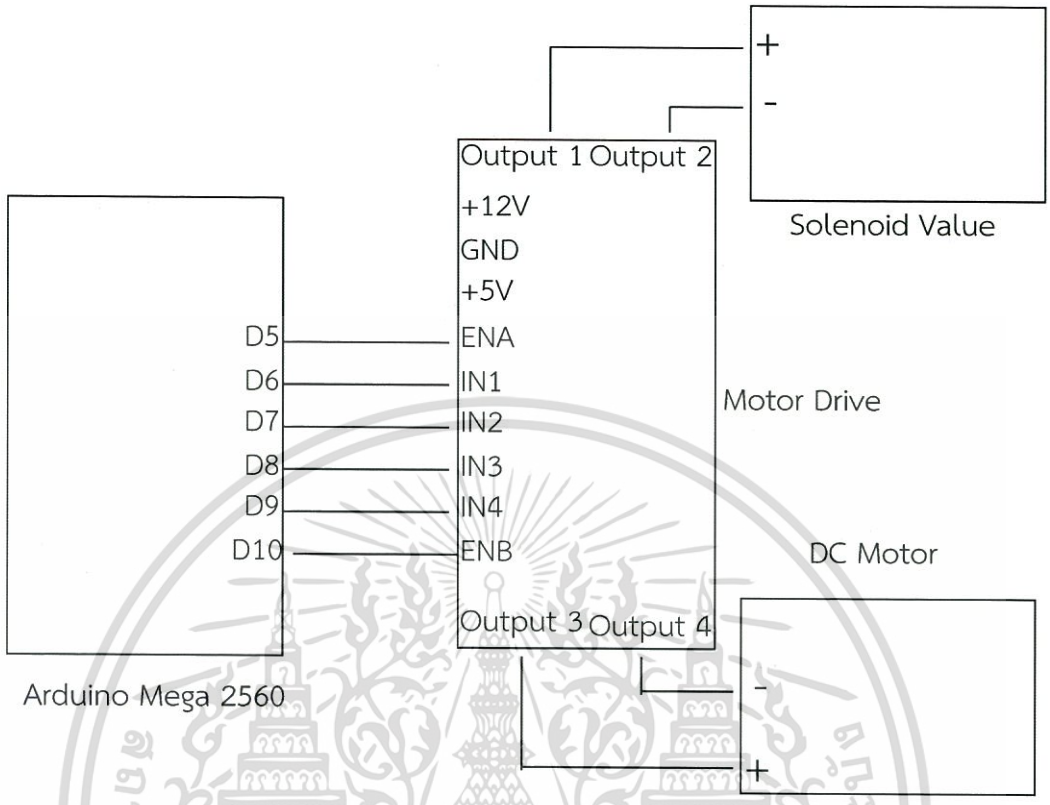
1. ต่อขา 3V3 จาก Arduino Mega 2560 R3 ไปที่ขา V+ ของโมดูล pH เซนเซอร์
2. ต่อขา GND (Ground) จาก Arduino Mega 2560 R3 ไปที่ขา G ของโมดูล pH เซนเซอร์
3. ต่อขา A0 จาก ESP8266 NodeMCU v.1.0 ไปที่ขา Po ของโมดูล pH เซนเซอร์
4. ต่อหัววัดค่า pH เข้ากับโมดูล pH เซนเซอร์
5. ต่อขา 5V จาก Arduino Mega 2560 R3 ไปที่ขา Vin ของ ESP8266 NodeMCU v.1.0
6. ต่อขา GND (Ground) จาก Arduino Mega 2560 R3 ไปที่ขา GND (Ground) ของ ESP8266 NodeMCU v.1.0
7. ต่อขา D11 จาก Arduino Mega 2560 ไปที่ขา Tx ของ ESP8266 NodeMCU v.1.0
8. ต่อขา D12 จาก Arduino Mega 2560 ไปที่ขา Rx ของ ESP8266 NodeMCU v.1.0
9. ต่ออะแดปเตอร์ 9 V เข้ากับ Power Jack ของ Arduino Mega 2560
10. ต่ออะแดปเตอร์ 12 V ขาบวกไปที่ +12 V ของชุดขับมอเตอร์
11. ต่ออะแดปเตอร์ 12 V ขาลบไปที่ GND ของชุดขับมอเตอร์
12. ต่อขา D5 จาก Arduino Mega 2560 ไปที่ขา ENA ของชุดขับมอเตอร์
13. ต่อขา D6 จาก Arduino Mega 2560 ไปที่ขา IN1 ของชุดขับมอเตอร์
14. ต่อขา D7 จาก Arduino Mega 2560 ไปที่ขา IN2 ของชุดขับมอเตอร์
15. ต่อขา D8 จาก Arduino Mega 2560 ไปที่ขา IN3 ของชุดขับมอเตอร์
16. ต่อขา D9 จาก Arduino Mega 2560 ไปที่ขา IN4 ของชุดขับมอเตอร์
17. ต่อขา D10 จาก Arduino Mega 2560 ไปที่ขา ENB ของชุดขับมอเตอร์
18. ต่อวาล์วโซลินอยด์เข้ากับชุดขับมอเตอร์(Motor Drive) โดยที่ขา + ของวาล์วโซลินอยด์ต่อกับ OUTPUT 1 ของชุดขับมอเตอร์และ ขา - ของวาล์วโซลินอยด์ต่อกับ OUTPUT 2 ของชุดขับมอเตอร์
19. ต่อมอเตอร์ (Motor) เข้ากับชุดขับมอเตอร์โดยที่ขา + ของมอเตอร์ต่อกับ OUTPUT 3 ของ ชุดขับมอเตอร์และ ขา - ของมอเตอร์กับ OUTPUT 4 ของชุดขับมอเตอร์



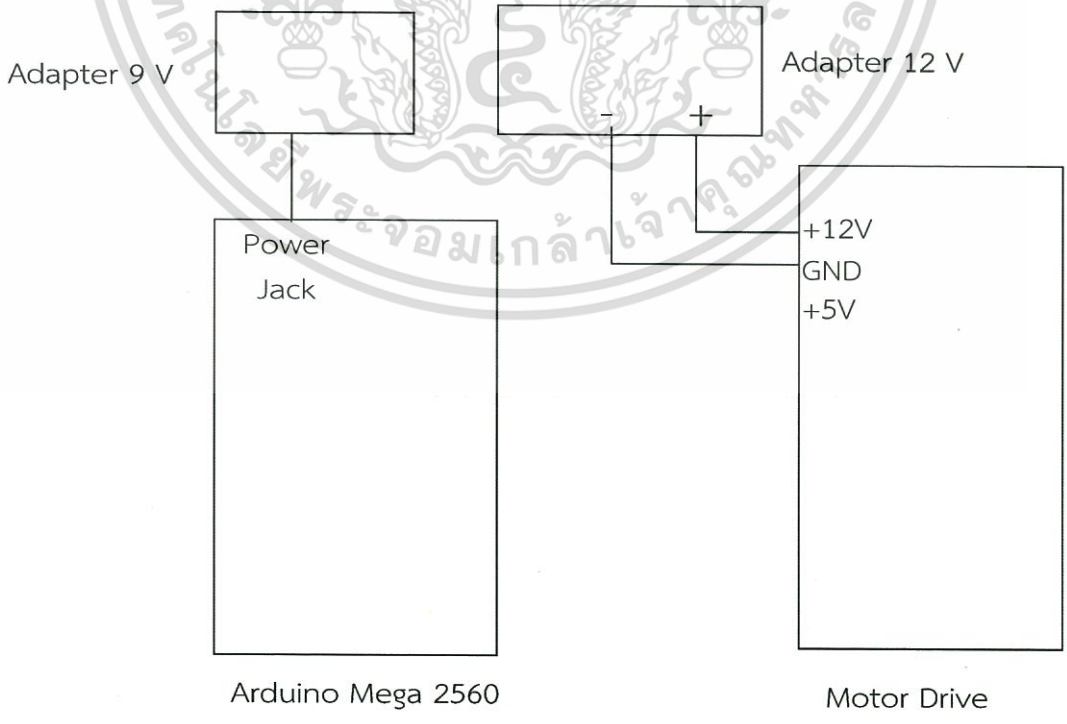

**รูปที่ 3.3** การต่อวงจรของ NodeMCU เข้ากับ Arduino




**รูปที่ 3.4** การต่อวงจรของโมดูล pH เซนเซอร์เข้ากับ Arduino และ NodeMCU




**รูปที่ 3.5** การต่อวงจรของ Arduino เข้ากับชุดขับมอเตอร์, วาล์วโซลินอยด์และมอเตอร์กระแสตรง

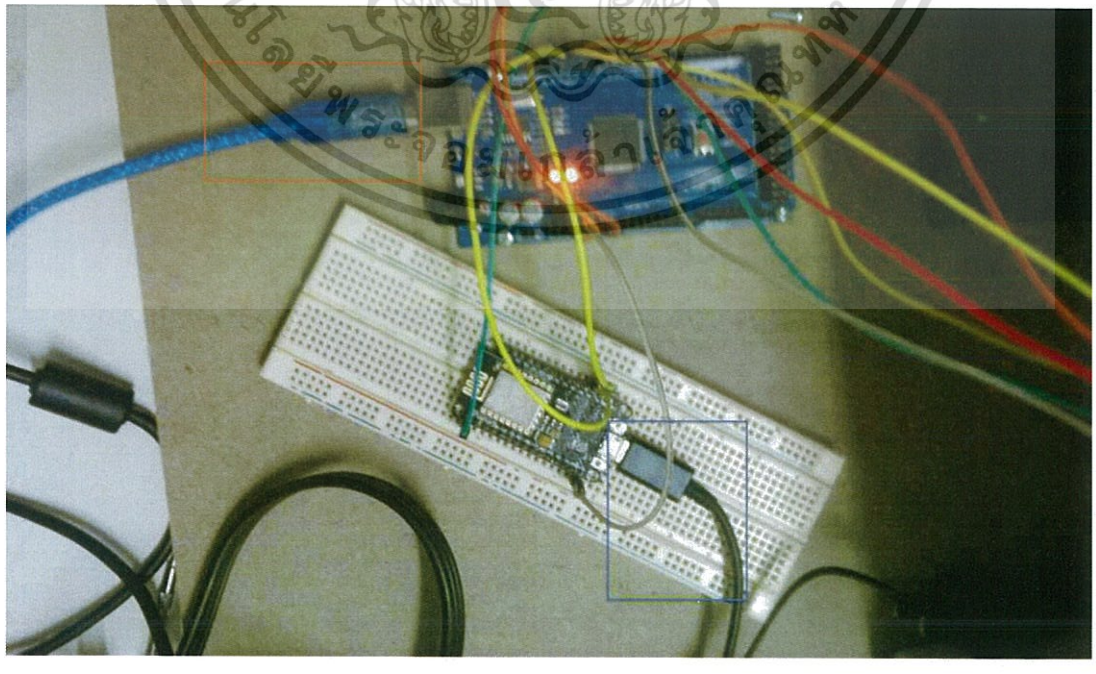



**รูปที่ 3.6** การต่อวงจรของ Arduino และชุดขับมอเตอร์เข้ากับอะแดปเตอร์ 9 V และ 12 V

## 3.3 การออกแบบโปรแกรม (Program)

การออกแบบโปรแกรมในการวัดค่า pH จะทำการออกแบบโปรแกรมด้วยโปรแกรม Arduino IDE (Intergrated Development Environment) โดยการเชื่อมต่อ USB กับตัว Arduino Mega 2560 และ NodeMCU v.1.0 ดังรูปที่ 3.7 และ 3.8

**รูปที่ 3.7** วงจรที่เชื่อมต่อเข้าคอมพิวเตอร์ของ Arduino Mega 2560 และ NodeMCU v.1.0

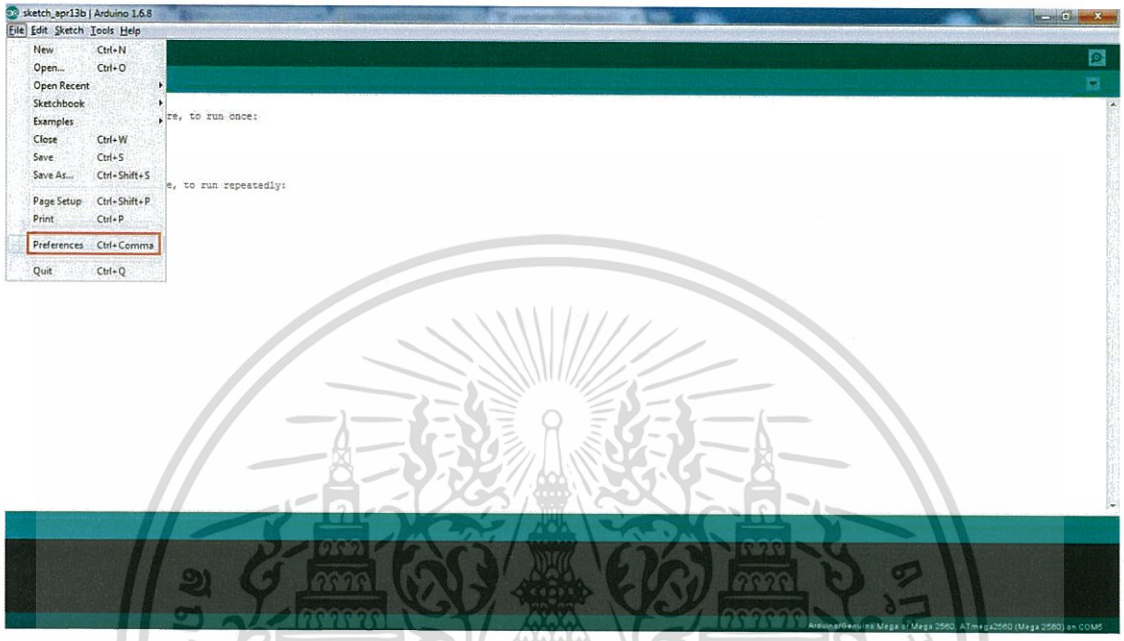


**รูปที่ 3.8** วงจรที่เชื่อมต่อ Arduino Mega 2560 และ NodeMCU v.1.0 เข้ากับคอมพิวเตอร์

### 3.3.1 วิธีการใช้งาน NodeMCU ในโปรแกรม Arduino IDE

โดยปกติ NodeMCU ไม่สามารถใช้งานในโปรแกรม Arduino IDE ได้โดยวิธีติดตั้งให้ทำดังนี้

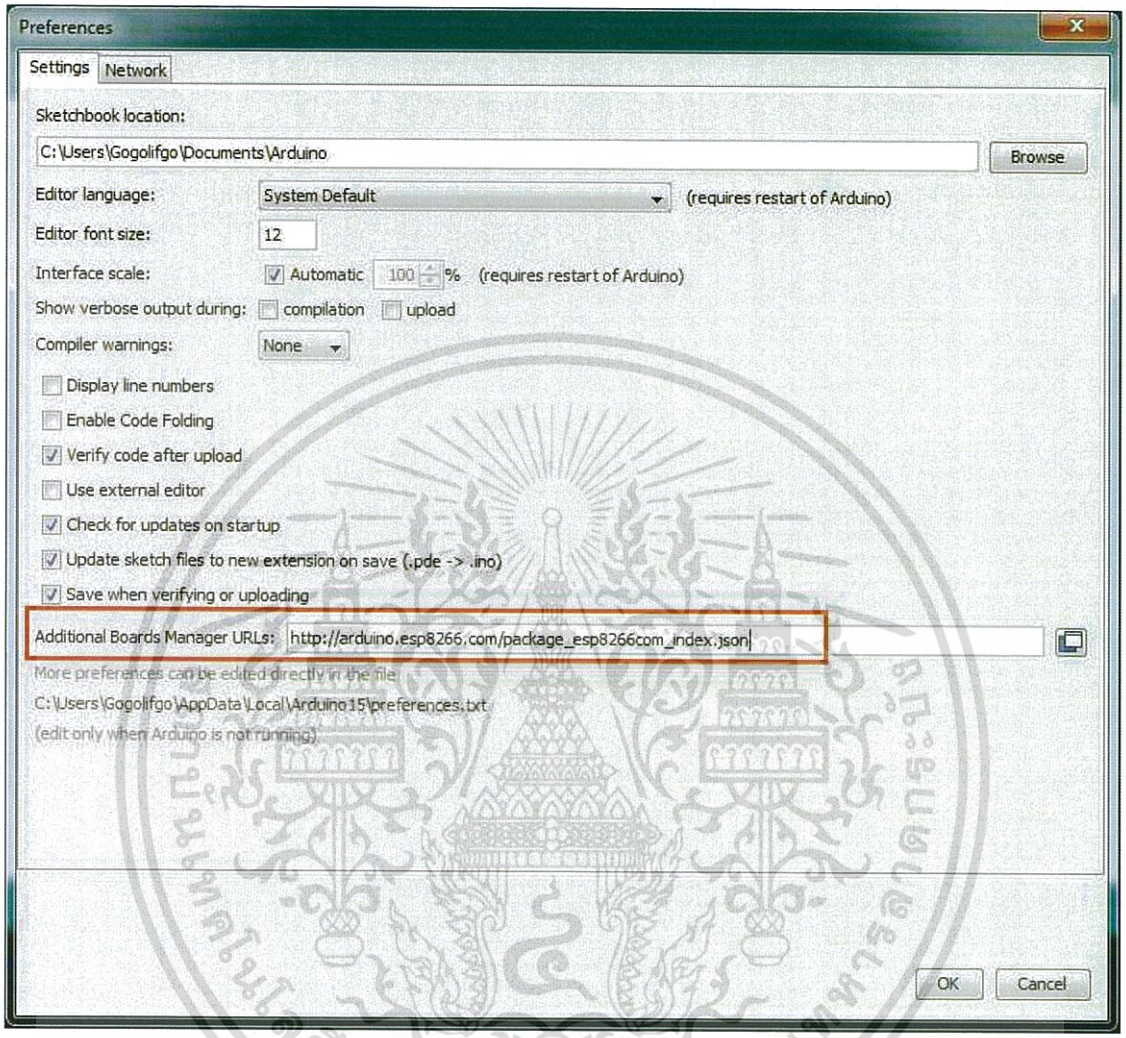
ให้ไปที่ File>Preference ดังรูปที่ 3.9




**รูปที่ 3.9** หน้าต่าง Preference

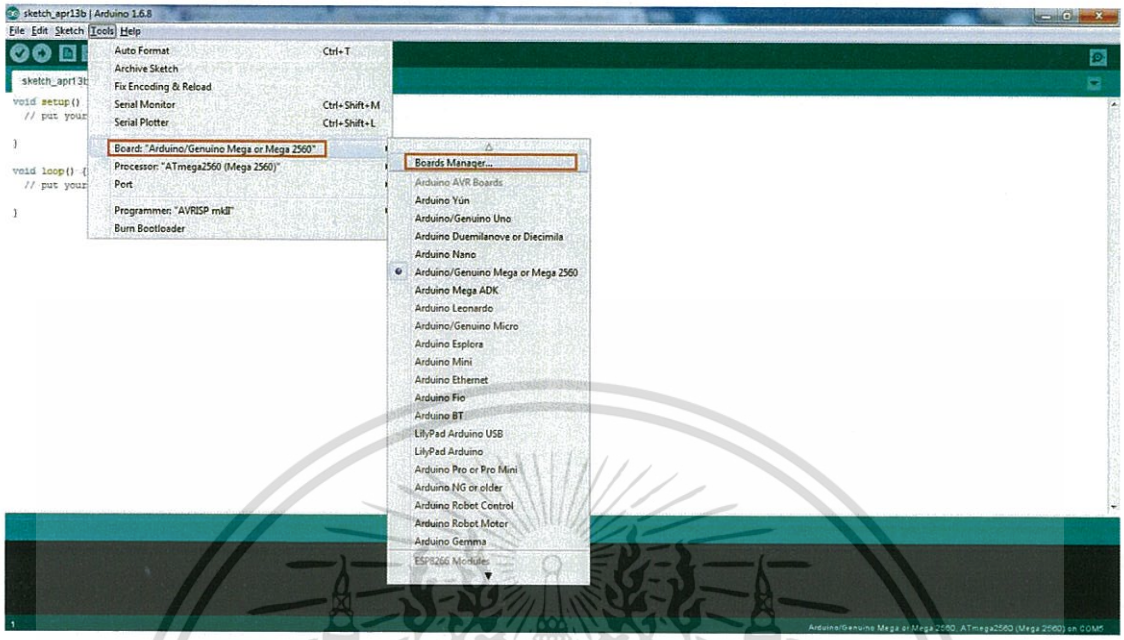
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

จะได้กล่องเมนูตามรูปที่ 3.10 ตรงช่อง Additional Boards Manager URLs: ให้ใส่ URL เป็น http://arduino.esp8266.com/package_esp8266com_index.json แล้วกด OK



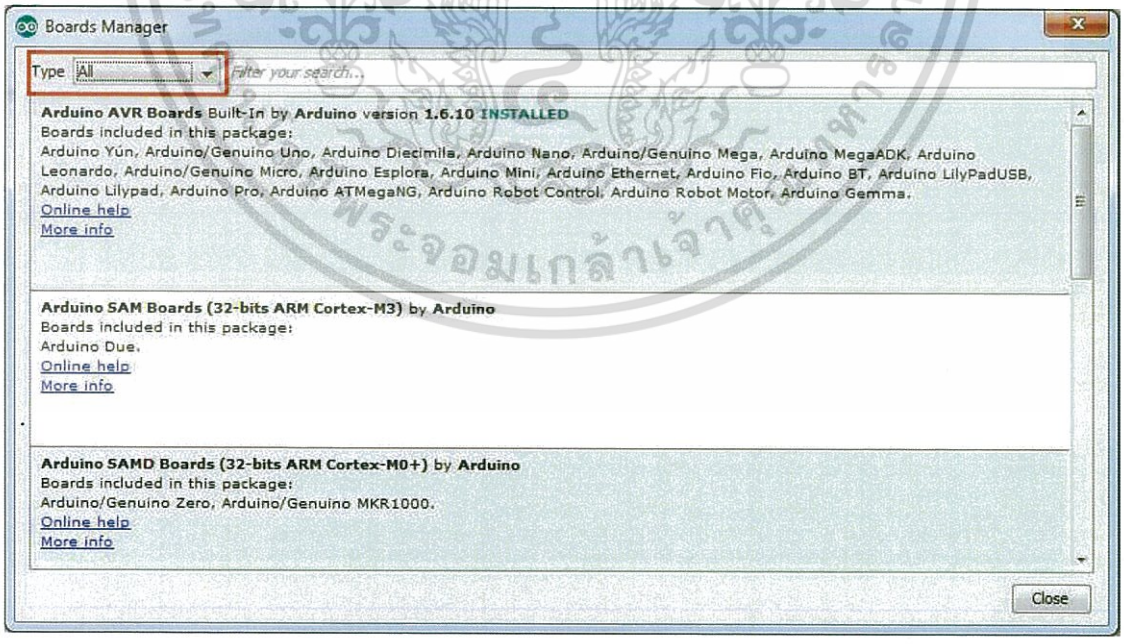

**รูปที่ 3.10** หน้าต่างของโปรแกรม Preference

จากนั้นให้ไปที่ Tools>Board>Board Manager ดังรูปที่ 3.11

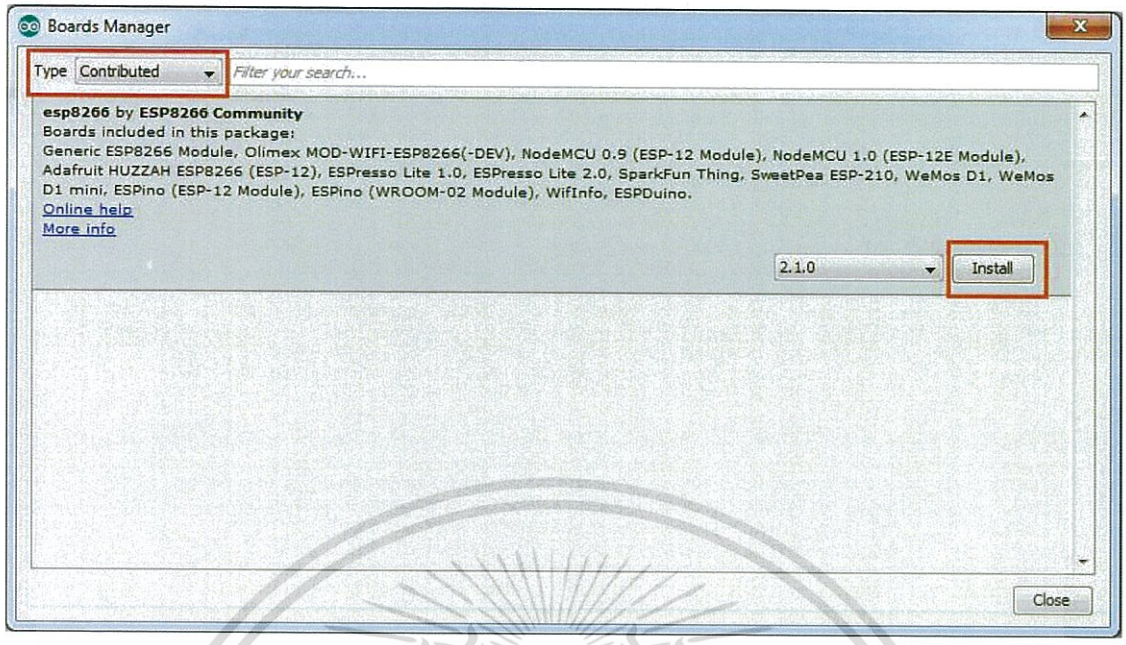



**รูปที่ 3.11** หน้าต่างวิธีเลือก Board Manager

จากนั้นจะได้กล่องเมนูของ Board Mangaer ดังรูปที่ 3.12 จากนั้นตรง Type ให้เปลี่ยน All เป็น Contributed ดังรูปที่ 3.13 หลังจากนั้นกดปุ่ม Install ที่อยู่แถบด้านขวา

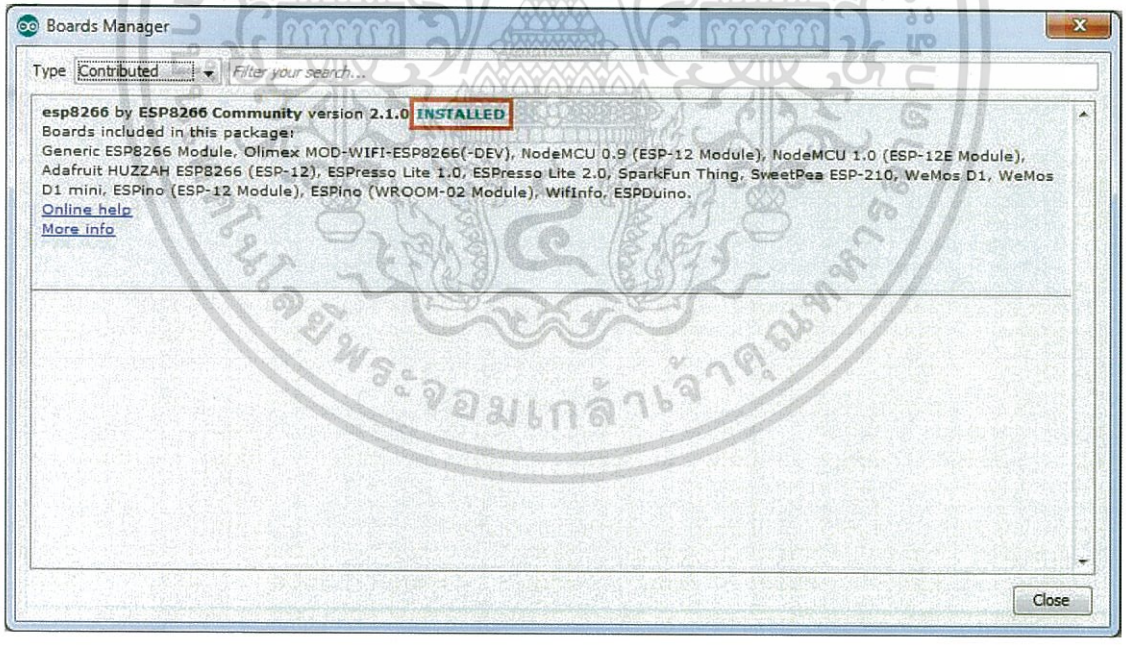



**รูปที่ 3.12** หน้าต่าง Board Manager



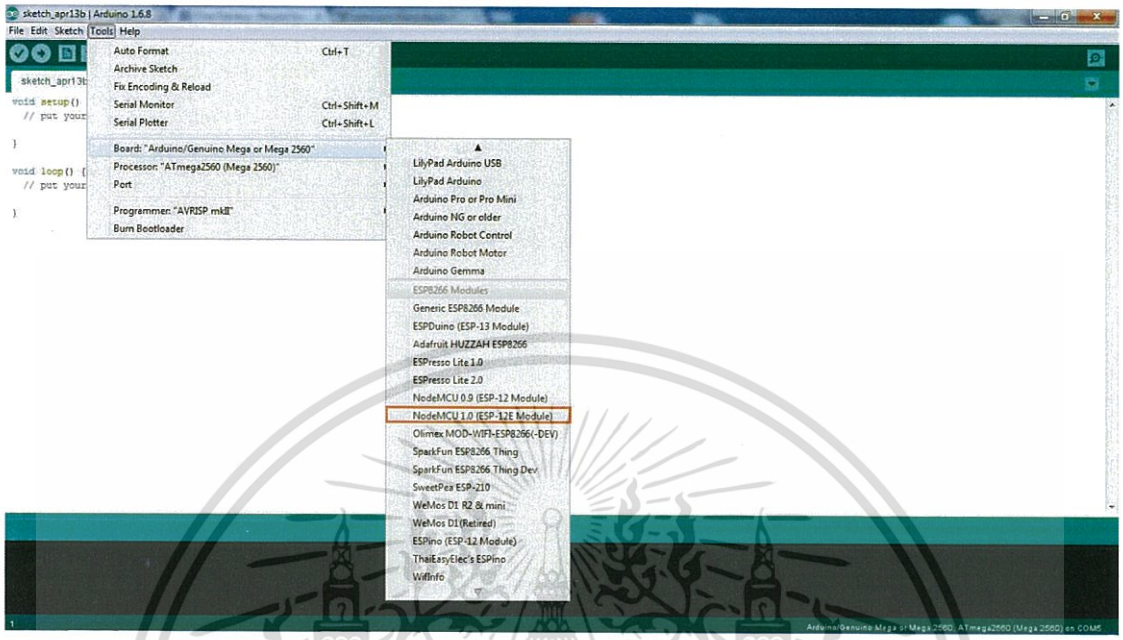

**รูปที่ 3.13** เปลี่ยน Type เป็น Contributed แล้วกดปุ่ม Install

หลังจาก Install แล้วจะเป็นดังรูปที่ 3.14 โดยจะมีตัวอักษรสีฟ้าที่เขียนว่า INSTALLED




**รูปที่ 3.14** ข้อความที่ขึ้นว่า INSTALLED

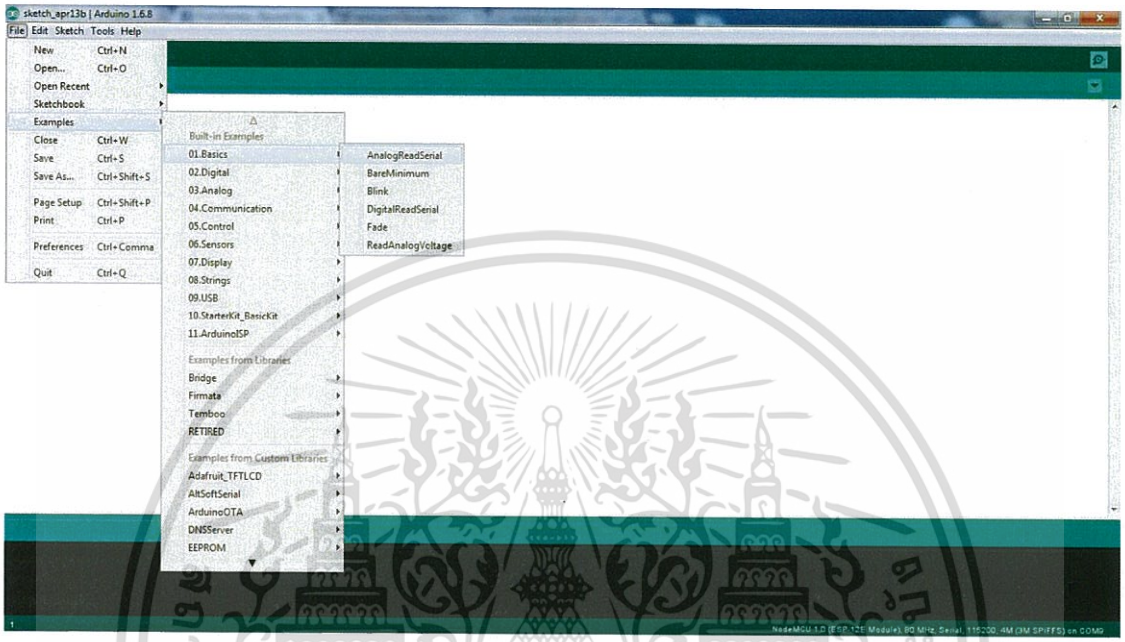
จากนั้นให้เปลี่ยน Board เป็น NodeMCU v.1.0 ได้โดยไปที่ Tools>Board>NodeMCU 1.0 (ESP-12E Module) จะสามารถ Upload Program เข้า NodeMCU ได้ ดังรูปที่ 3.15




**รูปที่ 3.15** วิธีเลือก Board เป็น NodeMCU 1.0 (ESP-12E Module)

### 3.3.2 วิธีการใช้งาน NodeMCU ในการอ่านค่าจากโมดูล pH เซนเซอร์

โดยใช้โปรแกรมตัวอย่างทดลองตัวอย่าง ซึ่งเป็นโปรแกรมที่ใช้สำหรับอ่านค่าชุดข้อมูลอนาล็อก (AnalogReadSerial) สามารถนำโค้ดมาจาก File>Examples>Basic>AnalogReadSerial ดังรูปที่ 3.16



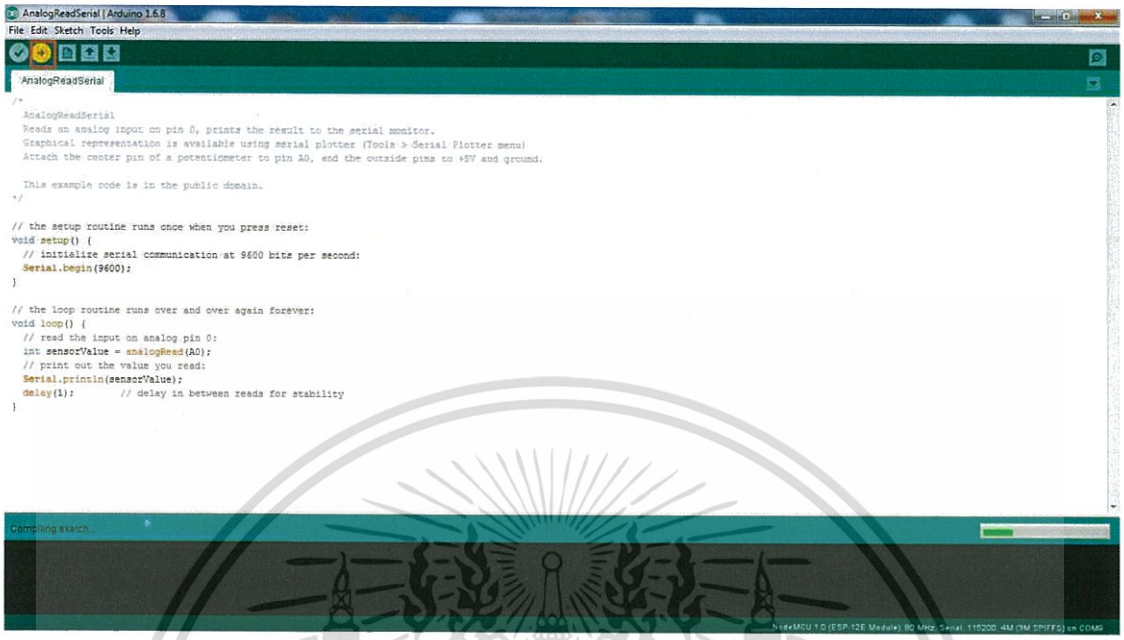

**รูปที่ 3.16** โปรแกรมที่อ่านค่าชุดข้อมูลอนาล็อก

และเมื่อกดแล้วจะปรากฏโค้ด (Code) ภาษา C ดังรูปที่ 3.17

```
/*
  AnalogReadSerial
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Graphical representation is available using serial plotter (Tools > Serial Plotter menu)
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

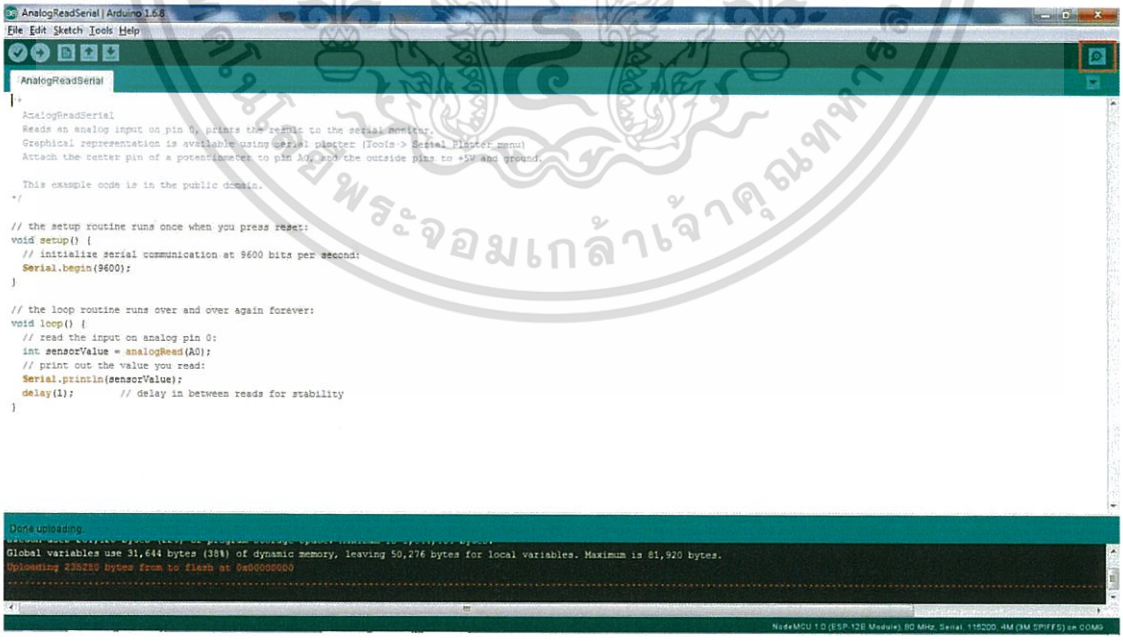
**รูปที่ 3.17** โค้ดภาษา C ภายในโปรแกรมตัวอย่าง

เมื่อได้โค้ด กด Upload ลง NodeMCU ดังรูปที่ 3.18

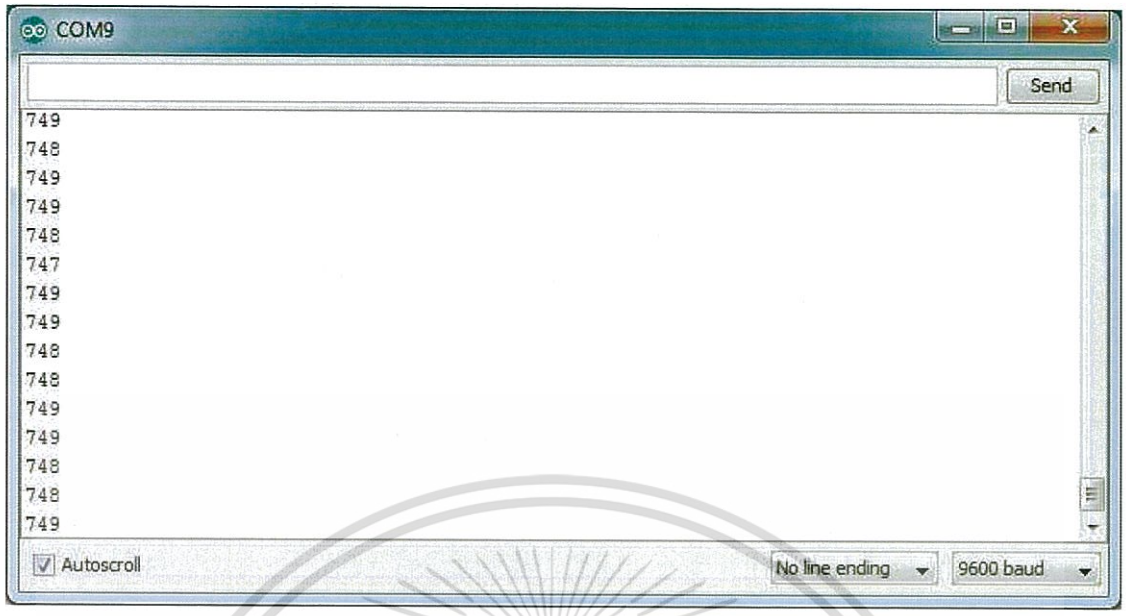



**รูปที่ 3.18** การ Upload โค้ดใส่ NodeMCU v.1.0

หลังจากนั้นกด Serial Monitor ดังแสดงในรูปที่ 3.19 เพื่อทำการดูค่าอนาล็อกที่ตัวเซนเซอร์ส่งมา จะขึ้นเป็นหน้าต่างดังแสดงในรูปที่ 3.20




**รูปที่ 3.19** การเปิดหน้าต่าง Serial Monitor เพื่อดูค่าอนาล็อกที่ส่งมาจากเซนเซอร์

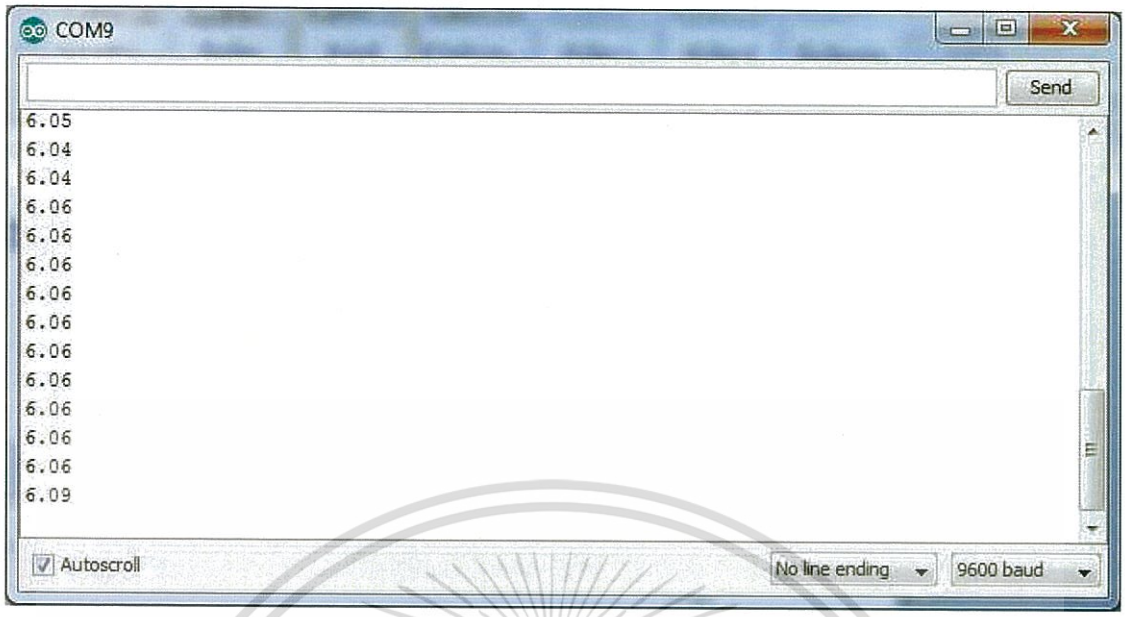



**รูปที่ 3.20** หน้าต่างแสดงค่าอนาล็อกจากเซนเซอร์

เมื่อทำการอ่านค่าจากเซนเซอร์ได้แล้ว จึงเริ่มออกแบบโปรแกรมเพื่อส่งค่าที่อ่านได้ไปทำการคำนวณให้อยู่ในรูปค่า pH และส่งค่าสู่คอมพิวเตอร์ดังรูปที่ 3.21 หลังจากนั้นกด Serial Monitor เพื่อทำการดูค่าอนาล็อกที่ตัวเซนเซอร์ส่งมา จะขึ้นเป็นหน้าต่างดังแสดงในรูปที่ 3.22

```
const byte pHpin = A0;
float Po;

void setup()
{
    Serial.begin(9600);
}

void loop()
{

    Po = (analogRead(pHpin) - 300 ) / 73.07;   // Read and reverse the analogue input value from the pH sensor then scale 0-14.
    Serial.println(Po, 2);                     // Print the result in the serial monitor.
    delay(2000);                               // Take 1 reading per second.
}
```

**รูปที่ 3.21** โค้ดที่ออกแบบเพื่ออ่านค่า pH

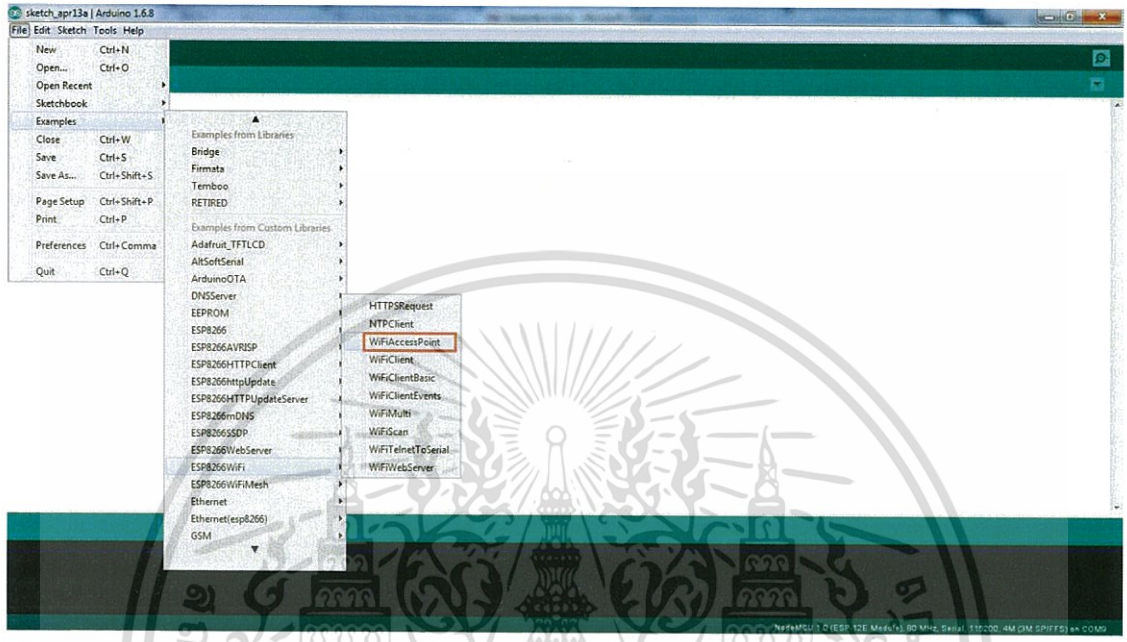



**รูปที่ 3.22** หน้าต่างแสดงค่า pH

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

### 3.3.3 วิธีการใช้งาน NodeMCU ในการสร้าง Access Point และ Web Server

โดยใช้โปรแกรมตัวอย่างทดลองตัวอย่าง ซึ่งเป็นโปรแกรมที่ใช้สำหรับการทำให้ NodeMCU เป็น Access Point และสร้าง Web Server ใน Access point สามารถนำโค้ดมาจาก File>Examples>ESP8266WiFi>WiFiAccessPoint ดังรูปที่ 3.23




**รูปที่ 3.23** โปรแกรมที่ทำให้ NodeMCU กลายเป็น Access Point

และเมื่อกดแล้วจะปรากฏโค้ดภาษา C ดังรูปที่ 3.24

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const char *ssid = "ESPap";
const char *password = "thereisnospoon";

ESP8266WebServer server(80);
void handleRoot() {
  server.send(200, "text/html", "<h1>You are connected</h1>");
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  /* You can remove the password parameter if you want the AP to be open. */
  WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
}
```

รูปที่ 3.24 โค้ดภาษา C ภายในโปรแกรมตัวอย่าง

สามารถทำการเปลี่ยน SSID และ password ได้ดังกรอบสีแดงดังรูปที่ 3.25 และให้เปลี่ยน Baud Rate ใน Serial Monitor ให้เท่ากับกรอบสีแดงอันล่าง ในที่นี้ใช้ SSID = Gogo12345 และ password = gogo54321 จากนั้นทำการอัพโหลดโค้ดลงใน NodeMCU

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
const char *ssid = "Gogo12345";
const char *password = "gogo54321";

ESP8266WebServer server(80);
void handleRoot() {
  server.send(200, "text/html", "<h1>You are connected</h1>");
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  /* You can remove the password parameter if you want the AP to be open. */
  WiFi.softAP(ssid, password);

  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
}
```

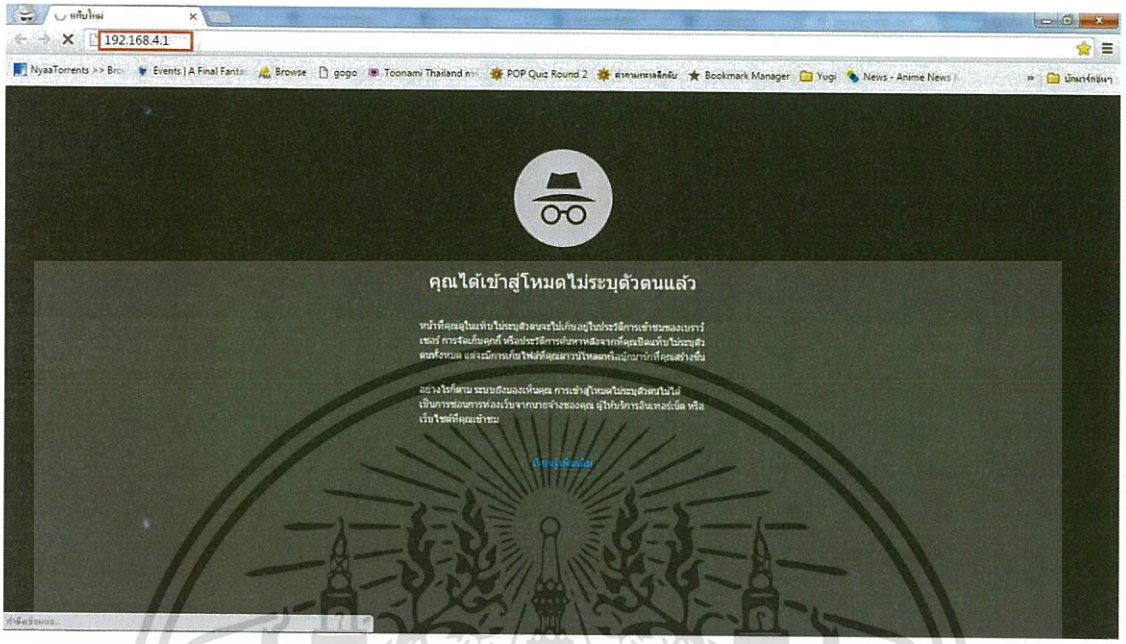
**รูปที่ 3.25** ตำแหน่งที่เปลี่ยน SSID และ password และความเร็ว Baud Rate

หลังจากอัพโหลดโค้ดแล้ว ให้เปิดที่ Wi-Fi Adapter ขึ้นมา จะสังเกตเห็นว่ามี Wi-Fi ชื่อ Gogo12345 ตามที่ตั้งในโค้ดของ NodeMCU ดังรูปที่ 3.26 จากนั้นให้ทำการเชื่อมต่อกับ Wi-Fi อันนั้นดังรูปที่ 3.27

**รูปที่ 3.26** Wi-Fi ชื่อ Gogo12345

**รูปที่ 3.27** ทำการเชื่อมต่อกับ Wi-Fi ชื่อ Gogo12345

จากนั้นให้เปิด Web Browser (ในที่นี้ใช้โปรแกรม Google Chrome) ตรงช่อง URL ให้พิมพ์ 192.168.4.1 เพื่อเข้า Web Server ใน NodeMCU ดังรูปที่ 3.28




**รูปที่ 3.28** วิธีเข้า Web Server

หลังจากที่เข้าได้แล้วจะแสดงหน้าต่างดังรูปที่ 3.29

**รูปที่ 3.29** Web Server ที่ได้สร้างไว้ใน Arduino

### 3.3.4 วิธีการทำให้ Arduino และ NodeMCU สามารถติดต่อสื่อสารระหว่างกัน

โดยดังรูปที่ จะเห็นว่ามี 2 หน้าต่างโดยหน้าต่างด้านซ้ายจะเป็นหน้าต่างของ Arduino Mega 2560 จะทำหน้าที่รับข้อมูล และหน้าต่างด้านขวาจะเป็นหน้าต่างของ NodeMCU จะทำหน้าที่ส่งข้อมูล โดยในที่นี้จะให้ส่งตัว A จาก NodeMCU ไปที่ Arduino Mega 2560 ให้เขียนโค้ดดังรูปที่ 3.30 จากนั้นทำการอัพโค้ดลงในแต่ละบอร์ด

รับ

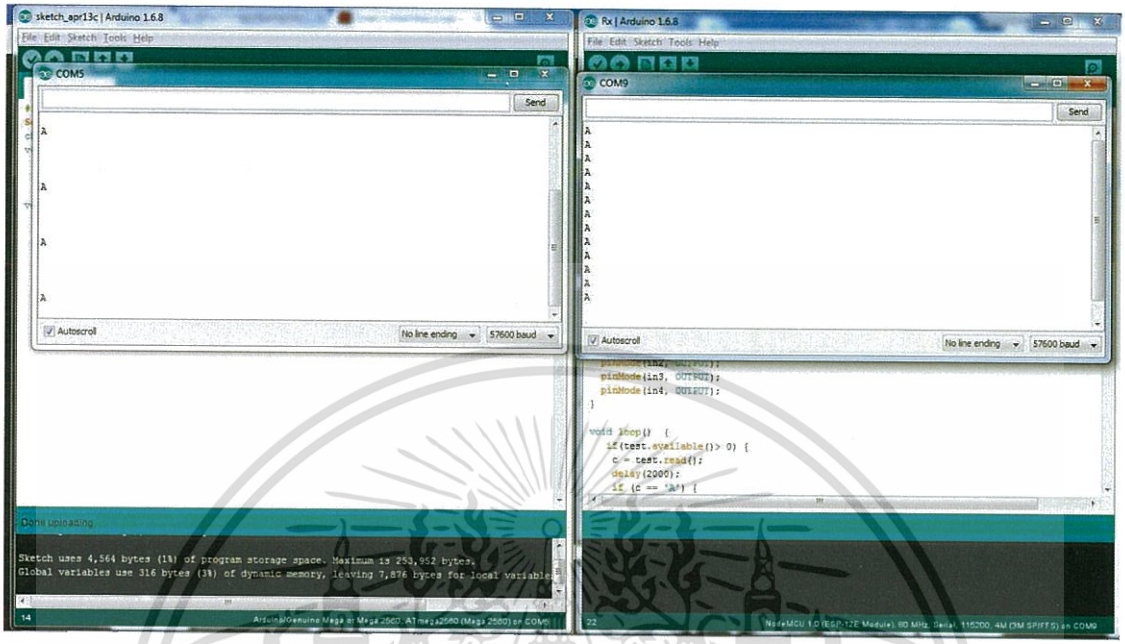
```
#include <SoftwareSerial.h>
SoftwareSerial project(11, 12); // RX, TX
char a ;
void setup() {
  Serial.begin(57600);
  project.begin(57600);
}
void loop() {
  if (project.available()>0) {
    a = project.read();
    Serial.println(a);
    delay(2000);
  }
}
```

ส่ง

```
char a = 'A';
void setup() {
  Serial.begin(57600);
}
void loop() {
  Serial.println(a);
  delay(10000);
}
```

**รูปที่ 3.30** Code ภาษา C ที่ใช้ทำการออกแบบระบบติดต่อสื่อสารกัน

หลังจากนั้นเปิดหน้าต่าง Serial Monitor ขึ้นมาจะเห็นว่าจะด้าน NodeMCU จะส่งค่าตัวแปร A ไปยัง Arduino Mega 2560 ได้ ดังรูปที่ 3.31




**รูปที่ 3.31** หน้าต่าง Serial Monitor ที่สื่อสารระหว่างกัน

### 3.3.5 วิธีการใช้งาน Drive Motor โดยสั่งผ่าน Arduino Mega 2560

ให้ทำการสร้างโค้ดดังรูปที่ 3.32, 3.33 และ 3.34 หลังจากนั้นให้ทำการอัพโหลดโค้ดไปยัง Arduino Mega 2560

```
int enA = 5;
int in1 = 6;
int in2 = 7;
int enB = 10;
int in3 = 8;
int in4 = 9;
void setup() {
  Serial.begin(9600);
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
if (Serial.available() > 0) {
int inByte = Serial.read();
int speed; // Local variable
switch (inByte) {
      case '1':
          analogWrite(enA, 255);
          digitalWrite(in1, HIGH);
          digitalWrite(in2, LOW);
          Serial.println("Motor 1 On");
          Serial.println("   ");
      break;
      case '2':
          analogWrite(enA, 255);
```

รูปที่ 3.32 โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 1

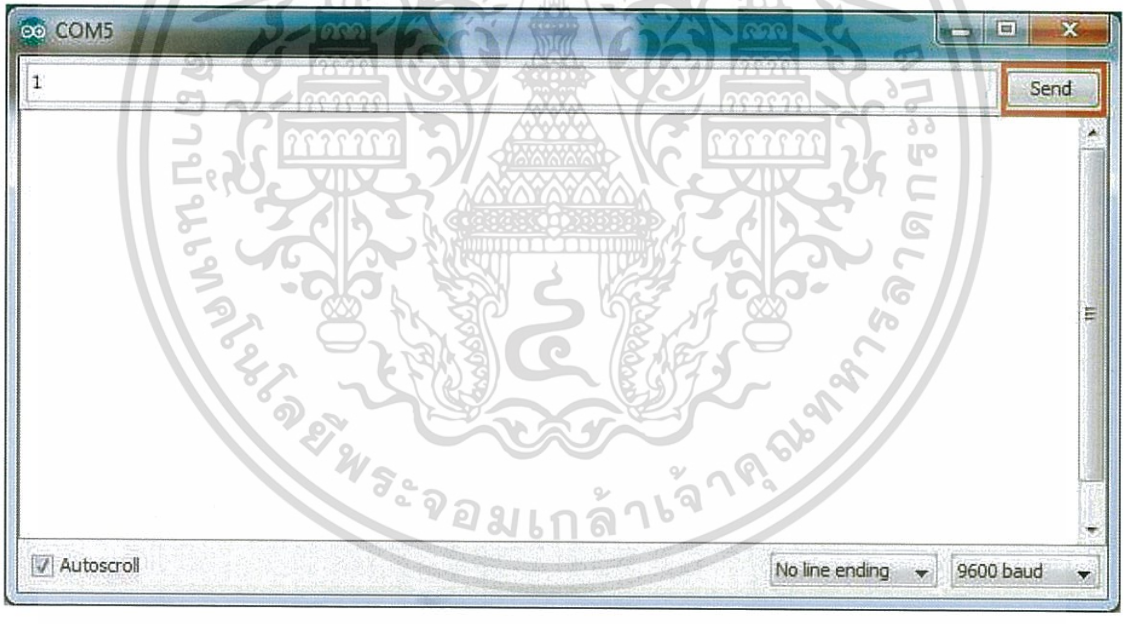
```
case '2':
    analogWrite(enA, 255);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    Serial.println("Motor 1 Off");
    Serial.println("   ");
break;
case '3':
    analogWrite(enB, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    Serial.println("Motor 2 On");
    Serial.println("   ");
break;
case '4':
    analogWrite(enB, 255);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    Serial.println("Motor 2 Off");
    Serial.println("   ");
break;
case '5':
    analogWrite(enA, 255);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW );
    Serial.println("All On");
    Serial.println("   ");
break;
```

**รูปที่ 3.33** โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 2

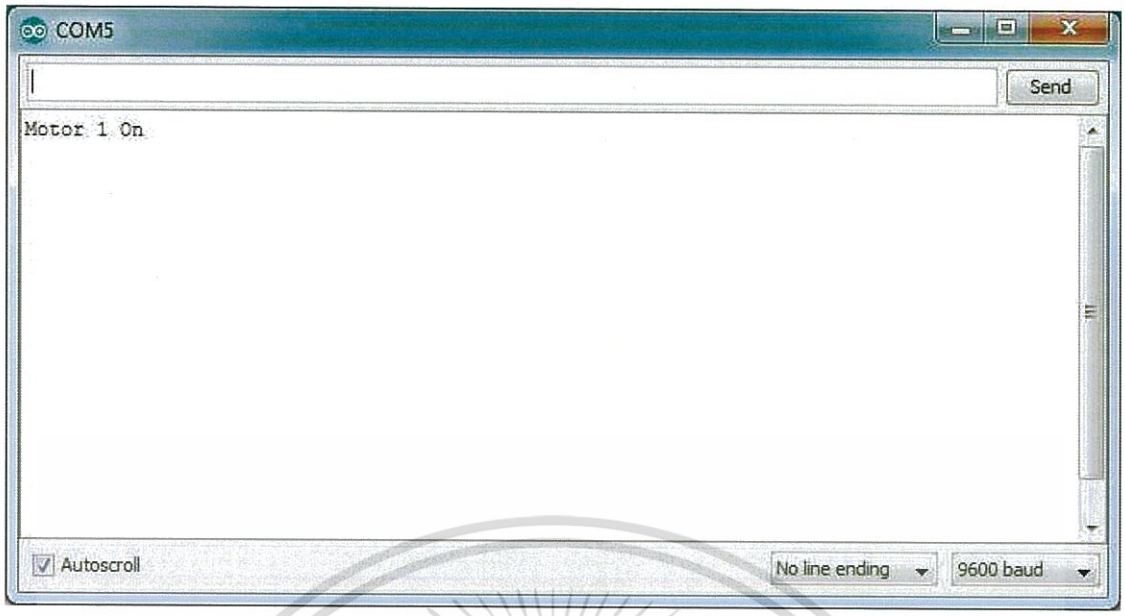
```
        case '6': // Motor 2 Reverse
            digitalWrite(in1, LOW);
            digitalWrite(in2, LOW);
            digitalWrite(in3, LOW);
            digitalWrite(in4, LOW);
            Serial.println("All Off");
            Serial.println("    ");
          break;
        default:
          for (int thisPin = 2; thisPin < 11; thisPin++)
          {
            digitalWrite(thisPin, LOW);
          }
      }
    }
}
```

**รูปที่ 3.34** โค้ดภาษา C ในการสั่งค่าไปยังชุดขับมอเตอร์ส่วนที่ 3

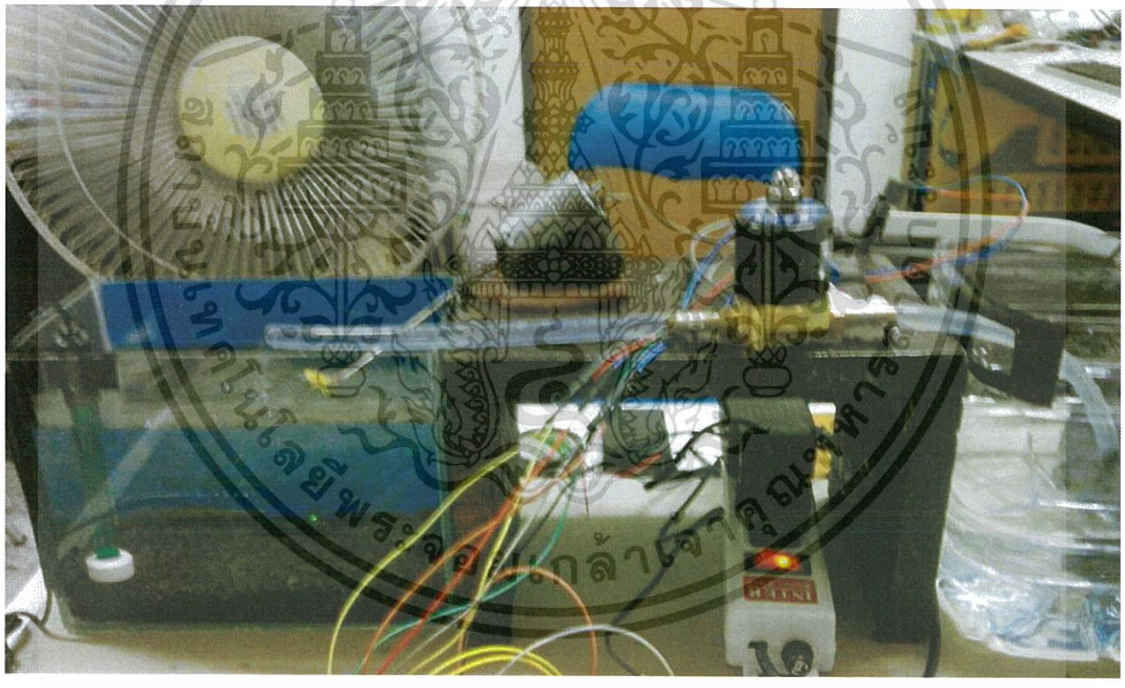
จากนั้นให้ทำการเปิด Serial Monitor ขึ้นมา แล้วให้พิมพ์เลข 1 แล้วกด Send ดังรูปที่ 3.35 จะได้ดังรูปที่ 3.36 จากนั้นวาล์วโซลินอยด์ก็จะทำงานดังรูปที่ 3.37




**รูปที่ 3.35** วิธีส่งค่าให้ชุดขับมอเตอร์ทำงาน



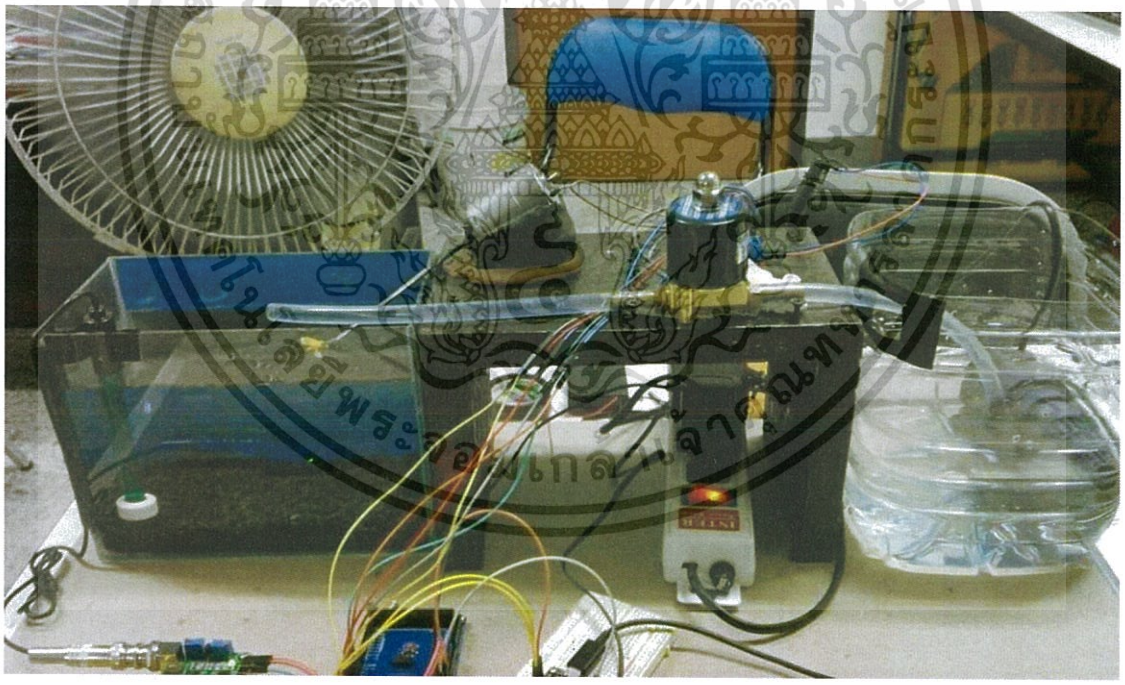

**รูปที่ 3.36** Serial Monitor หลังจากทำการส่งค่า 1



**รูปที่ 3.37** วาล์วโซลินอยด์ทำงาน

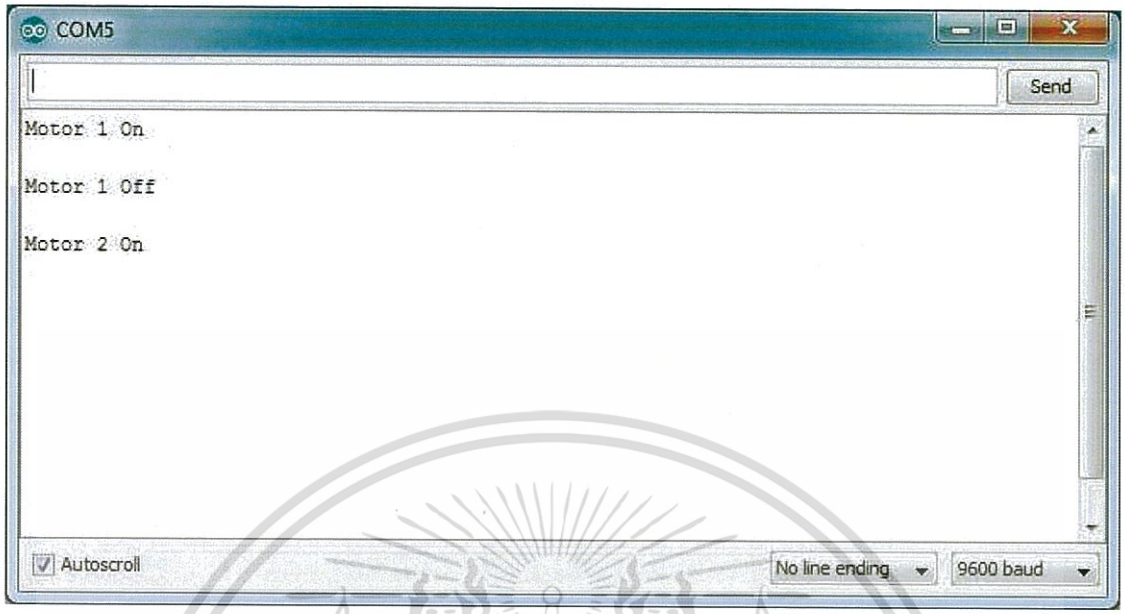
ให้พิมพ์เลข 2 แล้วกด Send จะได้ดังรูปที่ 3.38 จากนั้นวาล์วโซลินอยด์ก็จะหยุดทำงานดังรูปที่ 3.39

**รูปที่ 3.38** Serial Monitor หลังจากทำการส่งค่า 2

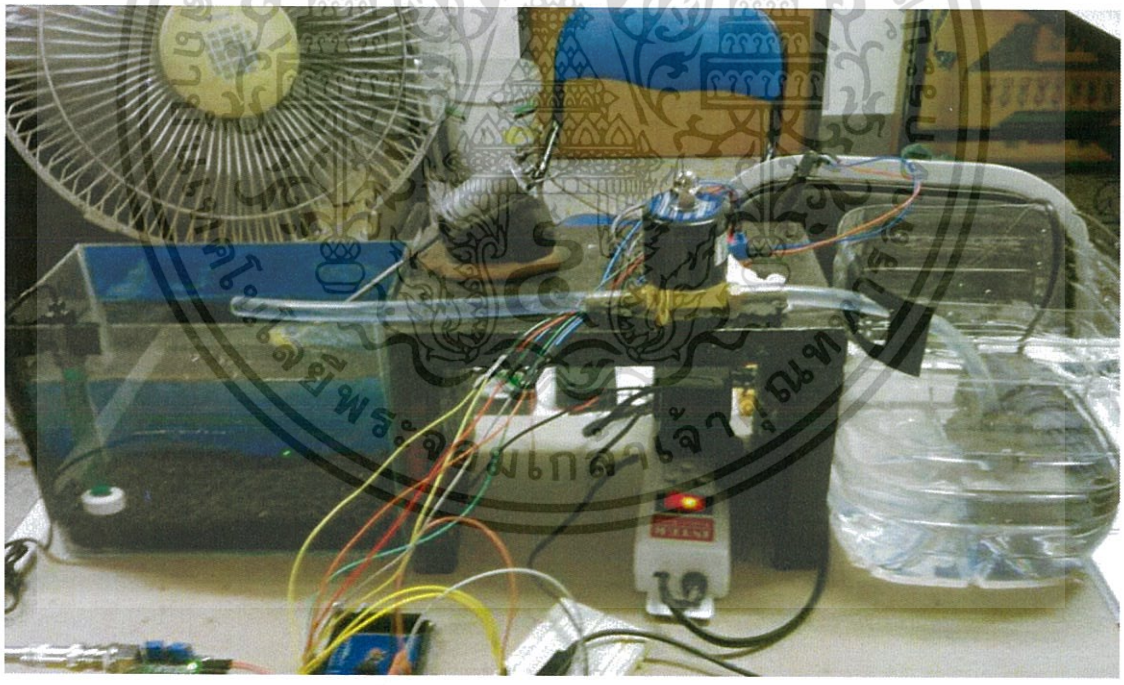


**รูปที่ 3.39** วาล์วโซลินอยด์หยุดทำงาน

ให้พิมพ์เลข 3 แล้วกด Send จะได้ดังรูปที่ 3.40 จากนั้นมอเตอร์ก็จะทำงานดังรูปที่ 3.41



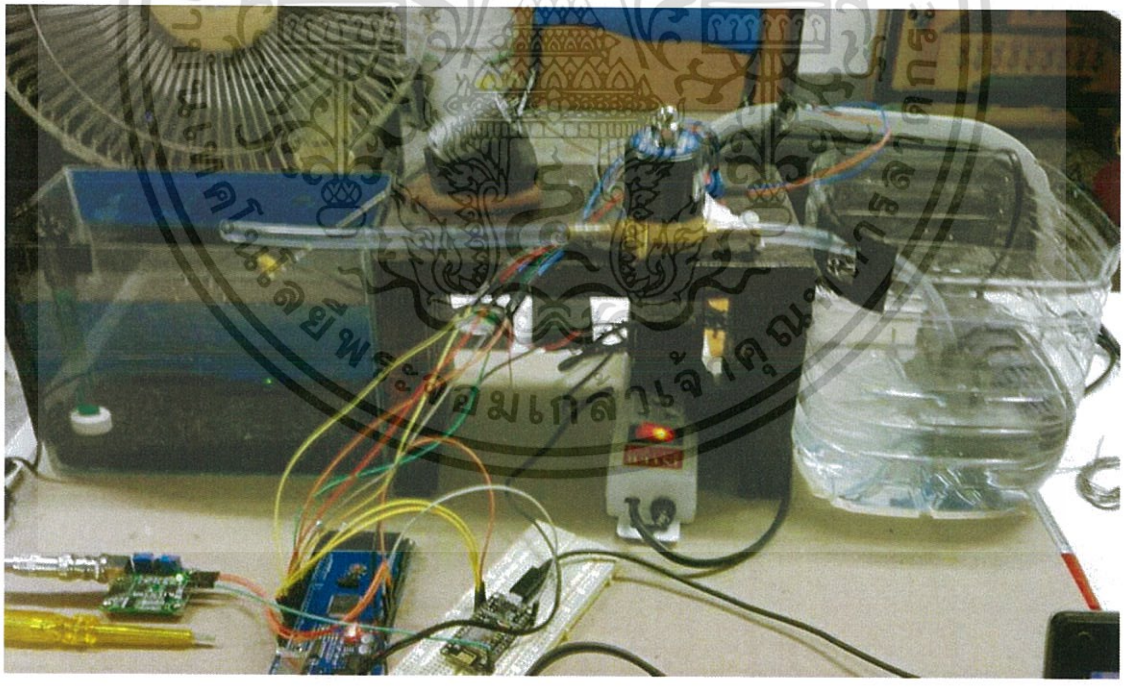

**รูปที่ 3.40** Serial Monitor หลังจากทำการส่งค่า 3



**รูปที่ 3.41** มอเตอร์ทำงาน

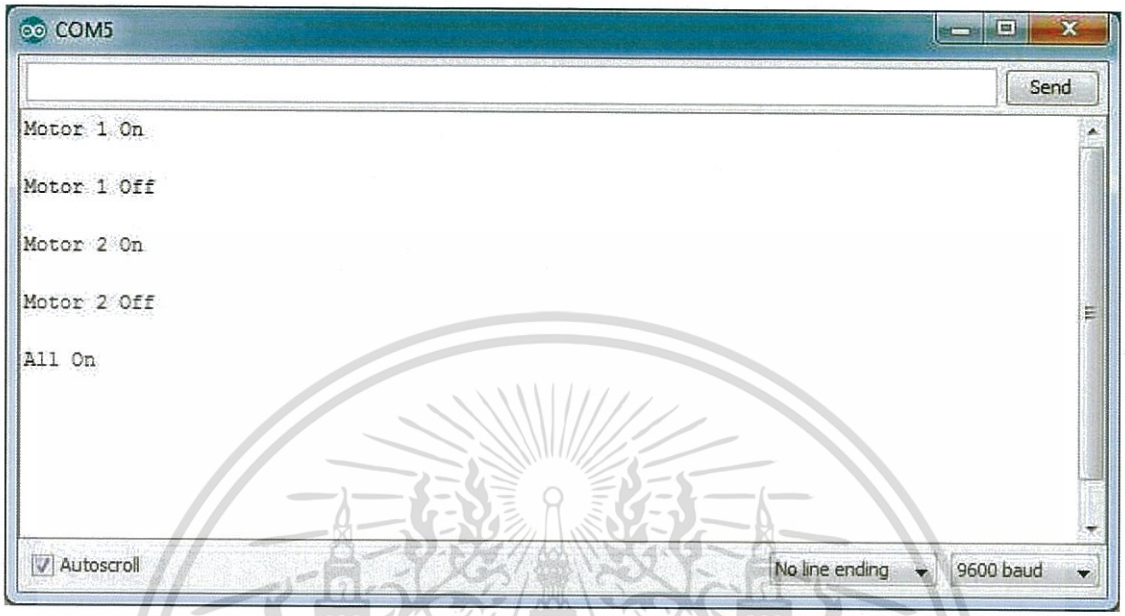
ให้พิมพ์เลข 4 แล้วกด Send จะได้ดังรูปที่ 3.42 จากนั้น มอเตอร์ก็จะหยุดทำงานดังรูปที่ 3.43

**รูปที่ 3.42** Serial Monitor หลังจากทำการส่งค่า 4

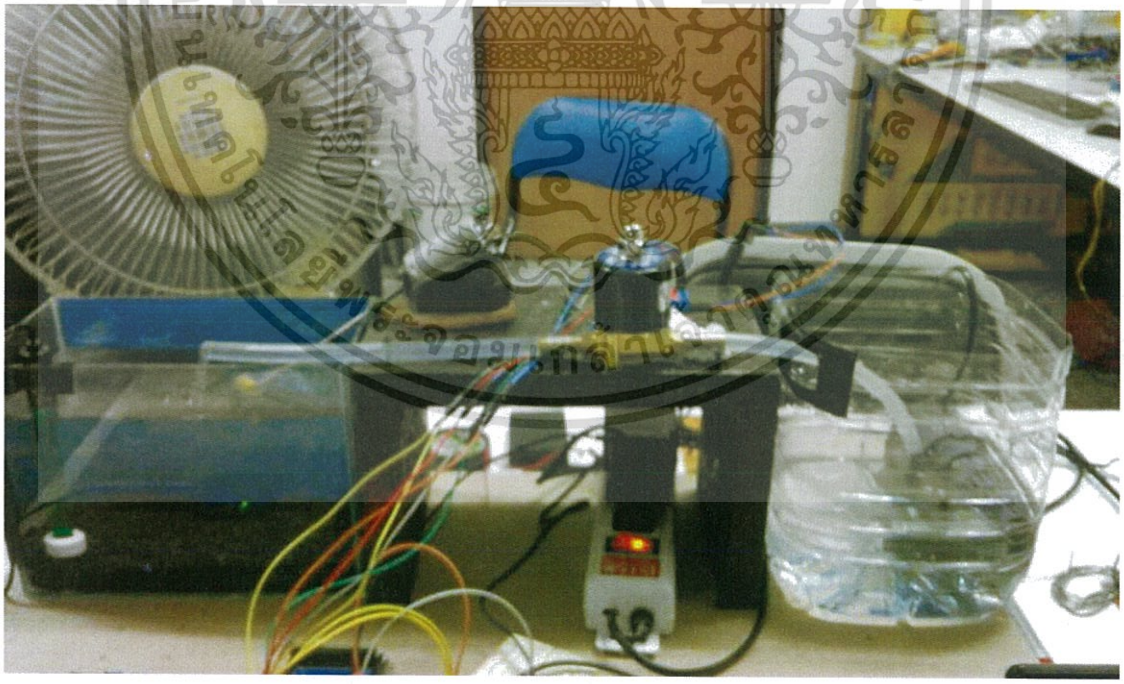


**รูปที่ 3.43** มอเตอร์หยุดทำงาน

ให้พิมพ์เลข 5 แล้วกด Send จะได้ดังรูปที่ 3.44 จากนั้นวาล์วโซลินอยด์และมอเตอร์ก็จะทำงานดังรูปที่ 3.45



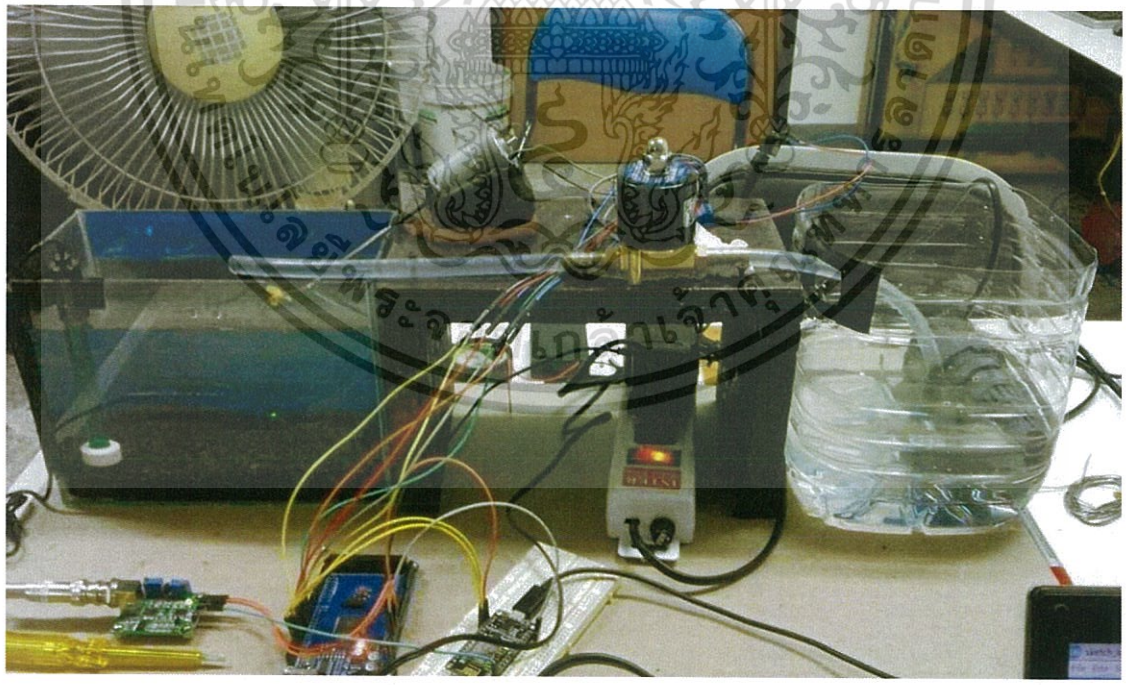

**รูปที่ 3.44** Serial Monitor หลังจากทำการส่งค่า 5



**รูปที่ 3.45** วาล์วโซลินอยด์และมอเตอร์ทำงาน

ให้พิมพ์เลข 6 แล้วกด Send จะได้ดังรูปที่ 3.46 จากนั้นวาล์วโซลินอยด์และมอเตอร์ก็จะหยุดทำงานดังรูปที่ 3.47

**รูปที่ 3.46** Serial Monitor หลังจากทำการส่งค่า 6



**รูปที่ 3.47** วาล์วโซลินอยด์และมอเตอร์หยุดทำงาน

นำโค้ดทุกอันที่ได้ในบทที่ 3.3.2, 3.3.3, 3.3.4 และ 3.3.5 มาเขียนเข้าด้วยกัน จากนั้นนำโค้ดที่เขียนได้อัพโหลดโค้ดไปยัง Arduino Mega 2560 ดังรูปที่ 3.48, 3.49 และ 3.50 และอัพโค้ดไปยัง NodeMCU ดังรูปที่ 3.51, 3.52, 3.53 และ 3.54 โดย SSID และ password ที่ตั้งไว้ใน NodeMCU คือ SSID คือ gogolifgo และ password คือ gogo35241

```
#include <SoftwareSerial.h>
SoftwareSerial test(11, 12); // RX, TX
int enB = 10;
int in3 = 8;
int in4 = 9;
// motor two
int enA = 5;
int in2 = 7;
int in1 = 6;
char c ;

void setup()  {
  Serial.begin(57600);
  test.begin(57600);
  // set all the motor control pins to outputs
 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
}

void loop()  {
  if(test.available()> 0) {
   c = test.read();
   delay(2000);
   if (c == 'A') {
      analogWrite(enB, 255);;
      digitalWrite(in3, HIGH);
```

**รูปที่ 3.48** โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 1

```
void loop()  {
   if(test.available()> 0) {
    c = test.read();
    delay(2000);
    if (c == 'A') {
         analogWrite(enB, 255);;
          digitalWrite(in3, HIGH);
          digitalWrite(in4, LOW);
          Serial.println("Motor On"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
    if (c == 'B') {
          analogWrite(enB, 255);;
          digitalWrite(in3, LOW);
          digitalWrite(in4, LOW);
          Serial.println("Motor Off"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
     if (c == 'C') {
          analogWrite(enA, 255);
          digitalWrite(in1, HIGH);
          digitalWrite(in2, LOW);
          Serial.println("Solenoid On"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
     if (c == 'D') {
          analogWrite(enA, 255);
          digitalWrite(in1, LOW);
          digitalWrite(in2, LOW);
          Serial.println("Solenoid Off"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
     if (c == 'E') {
          analogWrite(enA, 255);
```

**รูปที่ 3.49** โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 2

```
     if (c == 'E') {
          analogWrite(enA, 255);
          analogWrite(enB, 255);
          digitalWrite(in1, HIGH);
          digitalWrite(in2, LOW);
          digitalWrite(in3, HIGH);
          digitalWrite(in4, LOW );
          Serial.println("All On"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
     if (c == 'F') {
          analogWrite(enA, 255);
          analogWrite(enB, 255);
          digitalWrite(in1, LOW);
          digitalWrite(in2, LOW);
          digitalWrite(in3, LOW);
          digitalWrite(in4, LOW);
          Serial.println("All Off"); // Prints out â€œMotor 1 Forwardâ€  on the serial monitor
    }
}
}
```

**รูปที่ 3.50** โค้ดภาษา C ที่อัพโหลดลงใน Arduino Mega 2560 ส่วนที่ 3

```
#include <ESP8266WiFi.h>
const char* ssid = "gogolifgo";
const char* password = "gogo35241";
String readStrings = "";
char b = 'F';
const byte pHpin = A0;
float Po;
WiFiServer server(80);

void setup() {
  Serial.begin(57600);
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  server.begin();
}

void loop() {
Po = (analogRead(pHpin) - 300 ) / 73.07;
delay(3000);
  WiFiClient client = server.available();
  if(client){
   while(client.connected()){
      if(client.available()){
       char c = client.read();
      if(readStrings.length() < 100){
        readStrings += c;
      }
       if(c=='\n'){
          client.print("HTTP/1.1 200 OK\r\n");
          client.print("Content-Type: text/html\r\n\r\n");
```

**รูปที่ 3.51** โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 1

```
if(c=='\n'){
    client.print("HTTP/1.1 200 OK\r\n");
    client.print("Content-Type: text/html\r\n\r\n");
    client.print("<!DOCTYPE HTML>\r\n");
    client.print("<HTML>\r\n");
    client.print("<HEAD>\r\n");
    client.print("<meta http-equiv='refresh' content='10'/>\r\n");
    client.print("<meta name='apple-mobile-web-app-capable' content='yes' />\r\n");
    client.print("<meta name='apple-mobile-web-app-status-bar-style' content='black-translucent' />\r\n");
    client.print("<link rel='stylesheet' type='text/css' href='http://randomnerdtutorials.com/ethernetcss.css' />\r\n");
    client.print("<TITLE>NodeMCU Project</TITLE>\r\n");
    client.print("</HEAD>\r\n");
    client.print("<BODY>\r\n");//ในส่วนของ body คือสิ่งที่จะแสดงไปยังหน้า web
    client.print("<H1>Project ITE</H1>\r\n"); //สร้างหัวข้อความที่มีขนาด 1
    client.print("<hr />\r\n");//ขีดเส้นใต้
    client.print("<br />\r\n");  //ขึ้นบรรทัดใหม่
    client.print("<H2>Use Node MCU 1.0</H2>\r\n"); //สร้างหัวข้อความที่มีขนาด 2
    client.print("<br />\r\n");  //ขึ้นบรรทัดใหม่
    client.print("<H2>PH = "); client.print(Po); client.print("</H2>\r\n");
    client.print("<br />\r\n");  //ขึ้นบรรทัดใหม่
    client.print("<a href=\"/?button1on\"><font color = \"green\">Motor On</font></a>\r\n");
    client.print("<a href=\"/?button1off\"><font color = \"red\">Motor Off</font></a><br />\r\n");
    client.print("<br />\r\n");
    client.print("<a href=\"/?button2on\"><font color = \"green\">Solenoid On</font></a>\r\n");
    client.print("<a href=\"/?button2off\"><font color = \"red\">Solenoid Off</font></a><br />\r\n");
    client.print("<br />\r\n");
    client.print("<a href=\"/?button3on\"><font color = \"green\">All On</font></a>\r\n");
    client.print("<a href=\"/?button3off\"><font color = \"red\">All Off</font></a><br />\r\n");
    client.print("<br />\r\n");
    client.print("</BODY>\r\n");
```

**รูปที่ 3.52** โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 2

```
client.print("</HTML>\n");// ปิด tag

delay(1);

client.stop(); // เมื่อส่งไปแล้วก็ให้ client stop เพื่อหยุดการทำงาน

if(readStrings.indexOf("?button1on") > 0){
  b = 'A';

}
if(readStrings.indexOf("?button1off") >0){
  b = 'B';

}
if(readStrings.indexOf("?button2on") >0){
  b = 'C';

}
if(readStrings.indexOf("?button2off")>0){
  b = 'D';

}
if(readStrings.indexOf("?button3on")>0){
  b = 'E';

}
if(readStrings.indexOf("?button3off")>0){
  b = 'F';
}
```

**รูปที่ 3.53** โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 3

```
      readStrings = ""; //เมื่อจบโปรแกรมก็ให้ตัวแปร readStrings เคลียร์ค่าทั้งหมด เพื่อรอรับค่าใหม่
    }//if(c == '\n')
   }//if(client.available())
  }//while(client.connected())
 }//if(client)
 if(Po >= 8){
   b = 'E';
 }
 Serial.println(b);
 delay(10000);
}//void loop()
```

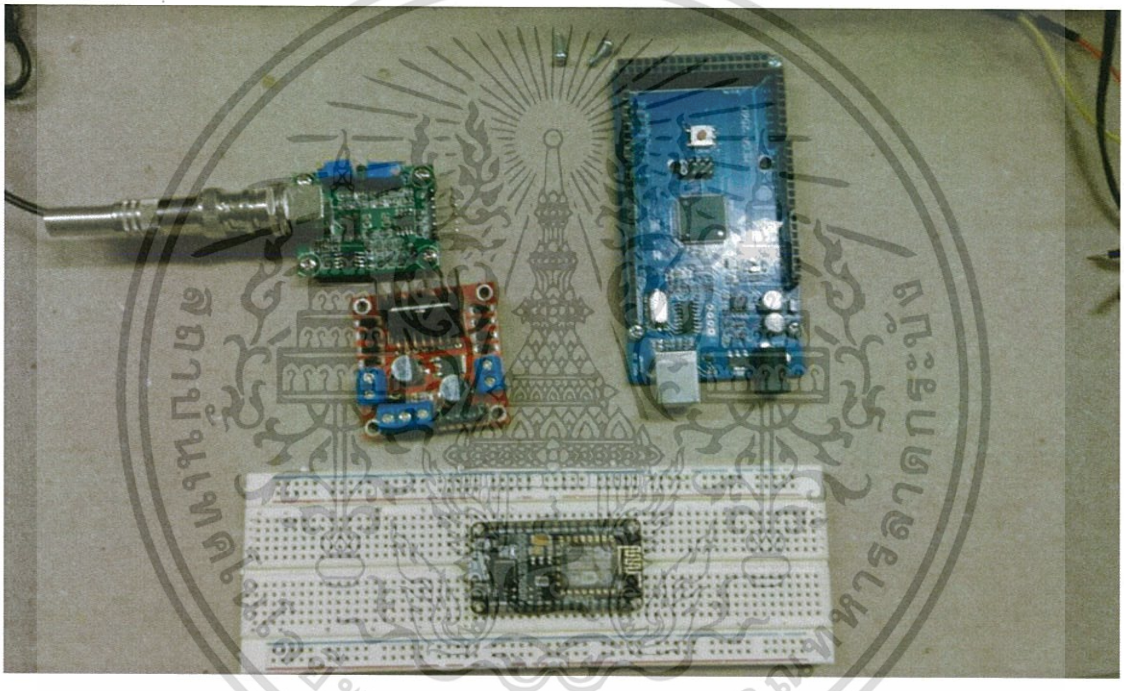
**รูปที่ 3.54** โค้ดภาษา C ที่อัพโหลดลงใน NodeMCU ส่วนที่ 4

# บทที่ 4
# ผลการทดลอง

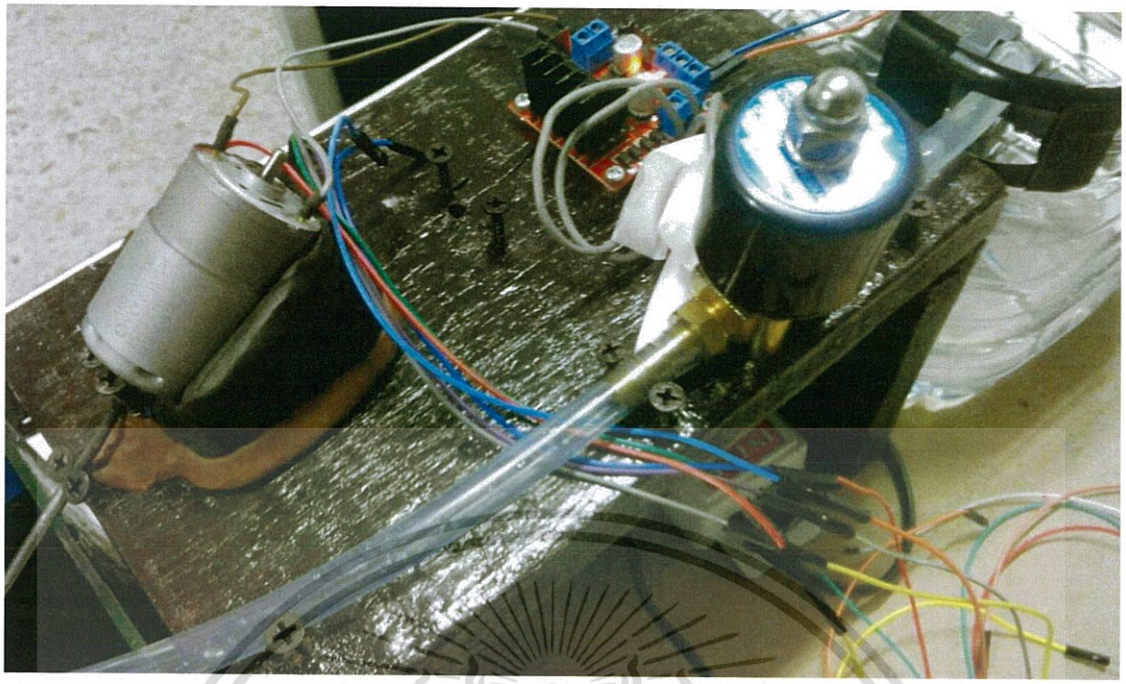
ในหัวข้อนี้จะกล่าวถึงรายละเอียดผลการทำงานของชิ้นงาน และผลการทำงานของโปรแกรม โดยจะแบ่งการผลทดลองเป็น 2 ส่วนคือ

1. ขั้นตอนในการต่ออุปกรณ์ฮาร์ดแวร์
2. การทดลองการแสดงผลค่า pH ผ่านหน้าเว็บไซต์และส่งค่าผ่านทางเว็บไซต์

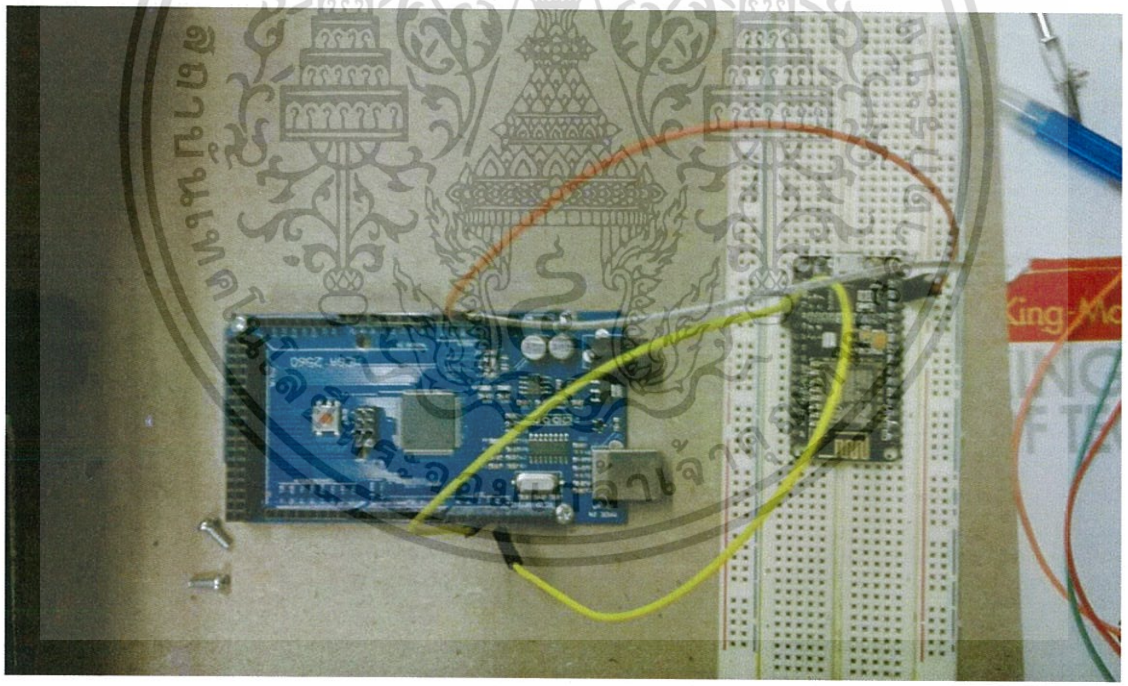
## 4.1 ขั้นตอนในการต่ออุปกรณ์ฮาร์ดแวร์



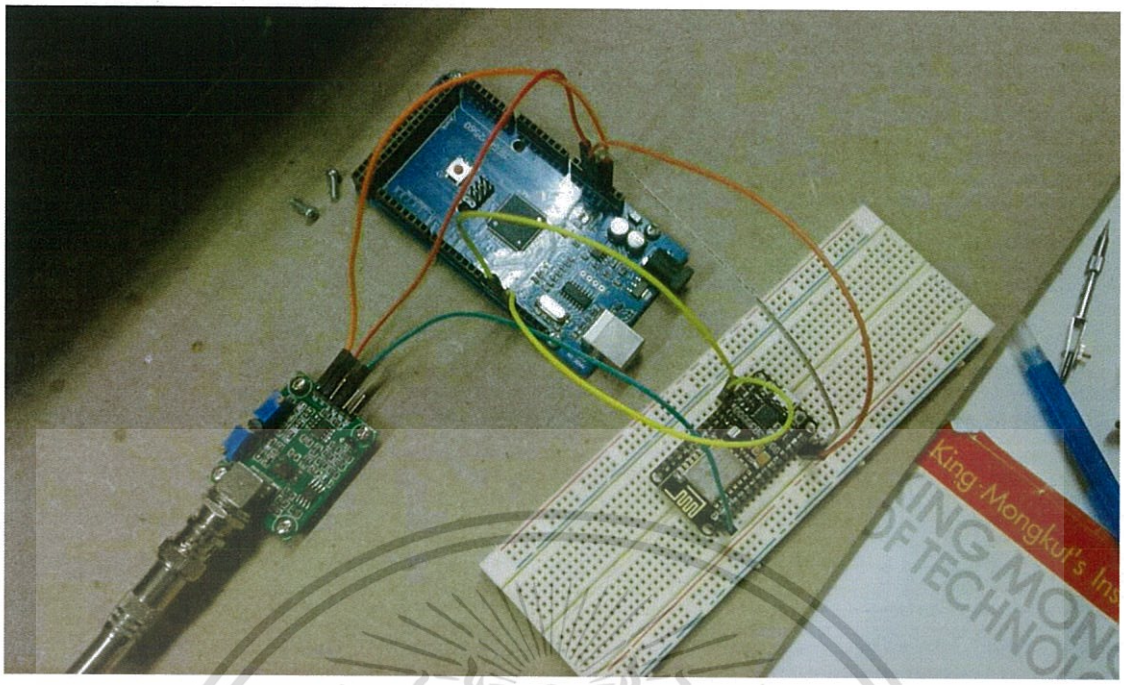
**รูปที่ 4.1** โมดูล pH เซนเซอร์, ชุดขับมอเตอร์, Arduino Mega 2560 และNodeMCU v.1.0



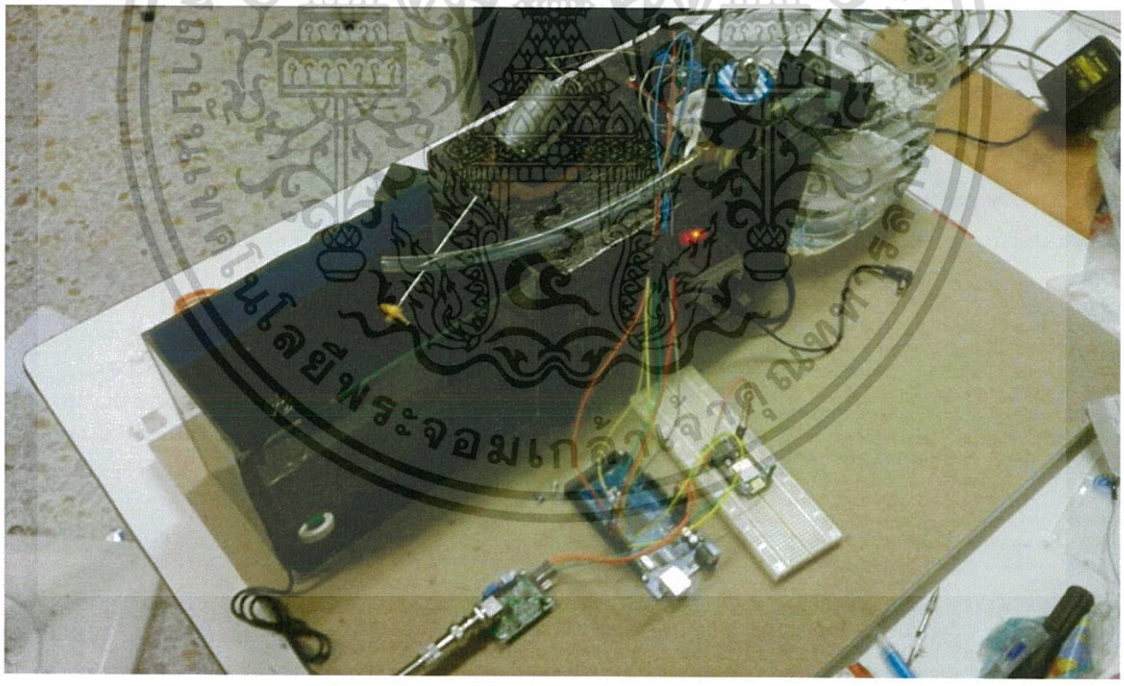
**รูปที่ 4.2** มอเตอร์และวาล์วโซลินอยด์



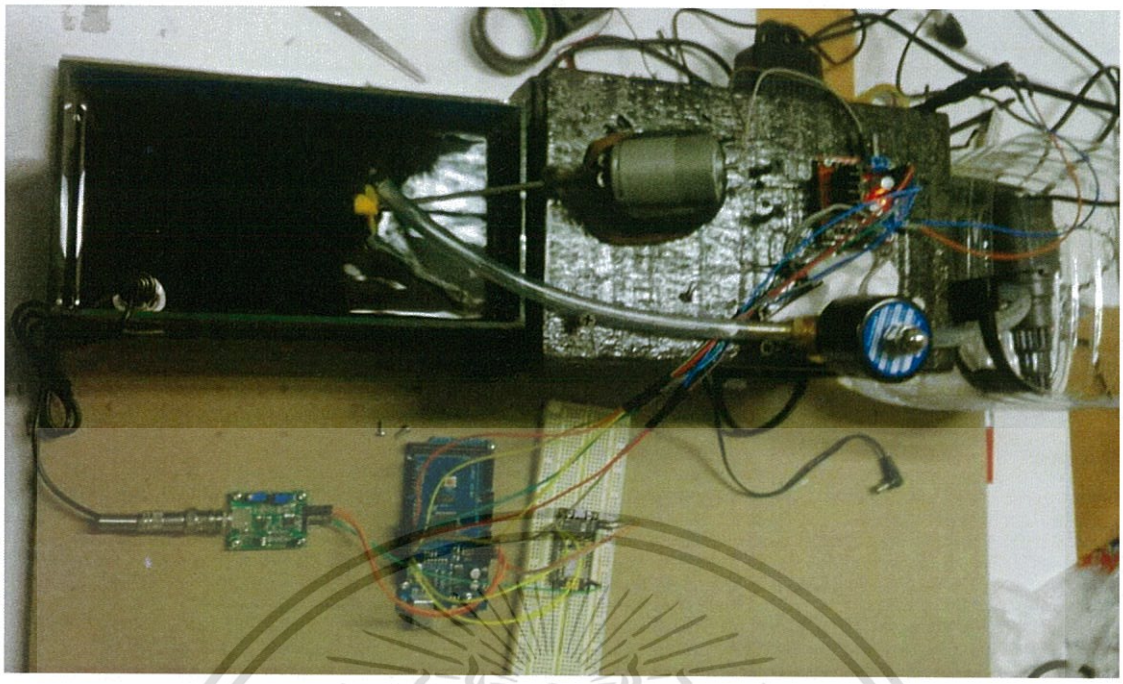
**รูปที่ 4.3** วิธีการต่อ Arduino Mega 2560 เข้ากับ NodeMCU v.1.0



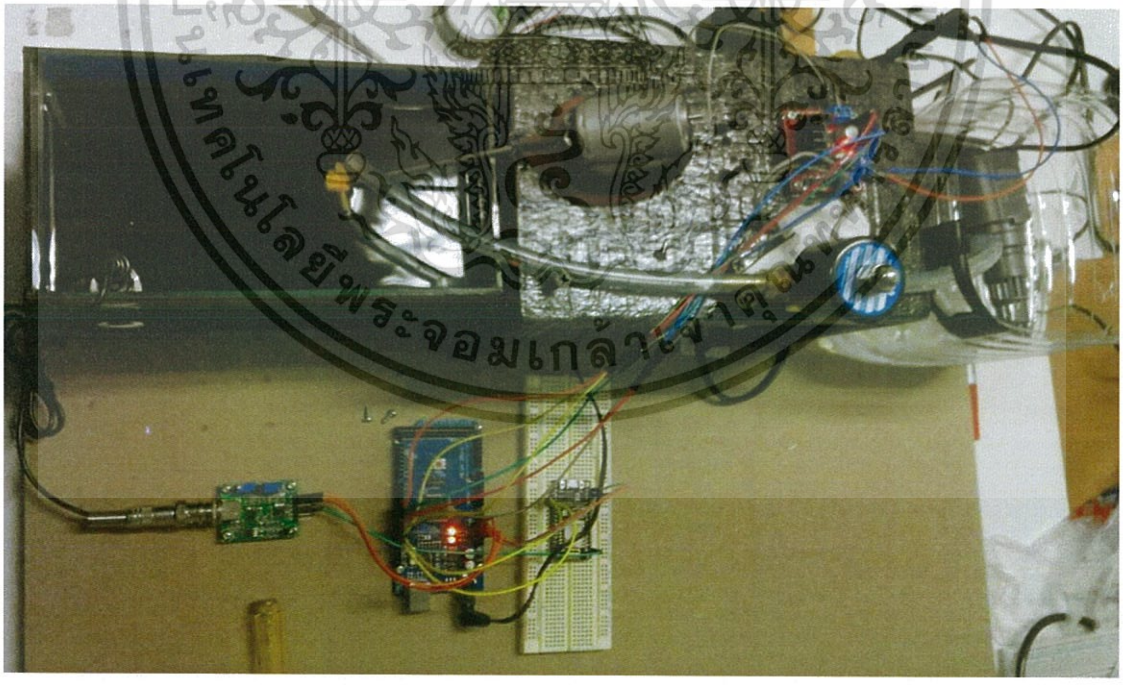
**รูปที่ 4.4** วิธีการต่อโมดูล pH เซนเซอร์เข้ากับ Arduino Mega 2560 และ NodeMCU v.1.0



**รูปที่ 4.5** วิธีการต่อชุดขับมอเตอร์เข้ากับ Arduino Mega 2560



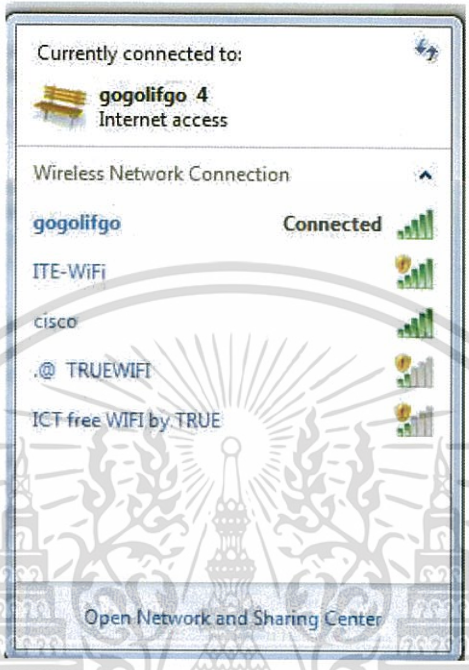
**รูปที่ 4.6** ต่อไฟจากอะแดปเตอร์ 12 V เข้ากับตัวชุดขับมอเตอร์เพื่อให้ทำงาน



**รูปที่ 4.7** ต่อไฟจากอะแดปเตอร์ 9 V เข้ากับตัว Arduino Mega 2560 เพื่อให้ทำงาน

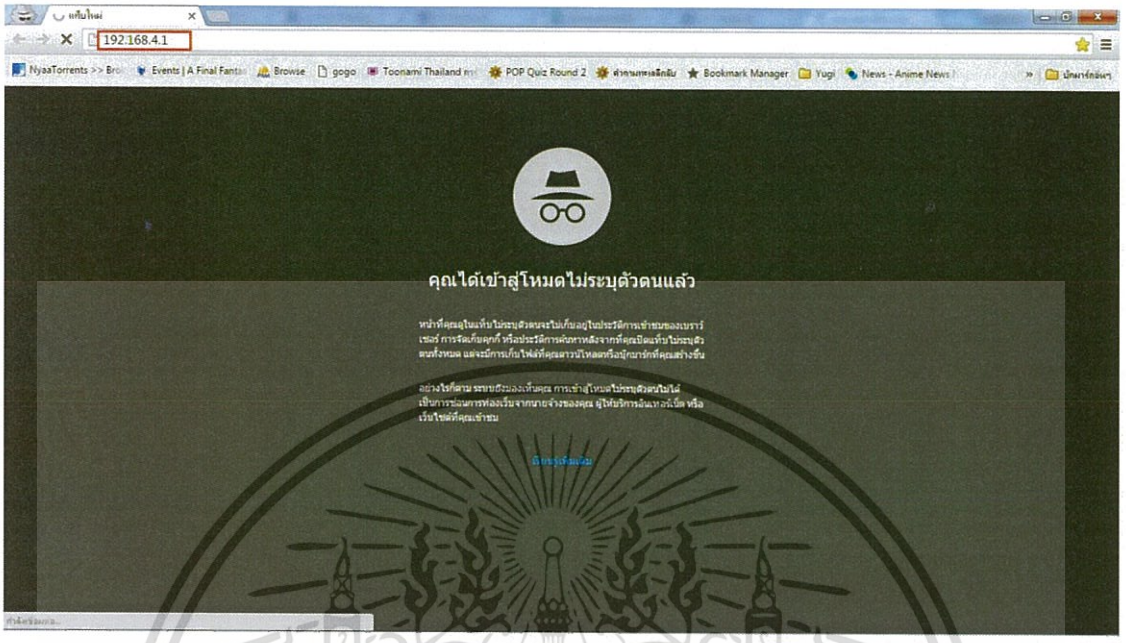
## 4.2 การทดลองการแสดงผลค่า pH ผ่านหน้าเว็บไซต์และส่งค่าผ่านทางเว็บไซต์

หลังจากได้ Upload โค้ดที่ได้จากบทที่ 3 ลงใน Arduino Mega 2560 และ NodeMCU v.1.0 แล้วจากนั้นให้ทำการเชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo ที่สร้างขึ้นมาจาก NodeMCU v.1.0 ดังรูปที่ 4.8




**รูปที่** 4.8 เชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo

จากนั้นให้เปิด Web Browser ตรงช่อง URL ให้พิมพ์ 192.168.4.1 เพื่อเข้า Web Server ใน NodeMCU ดังรูปที่ 4.9

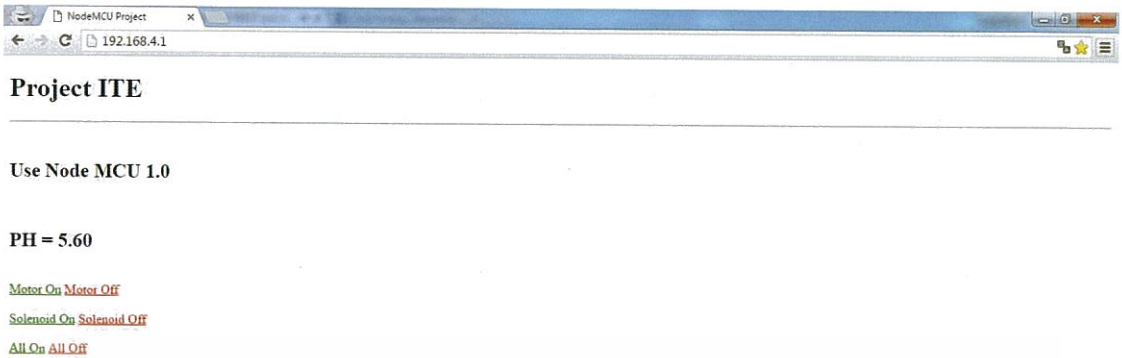



**รูปที่ 4.9** วิธีเข้า Web Server

หลังจากที่เข้าได้แล้วจะแสดงหน้าต่างดังรูปที่ 4.10

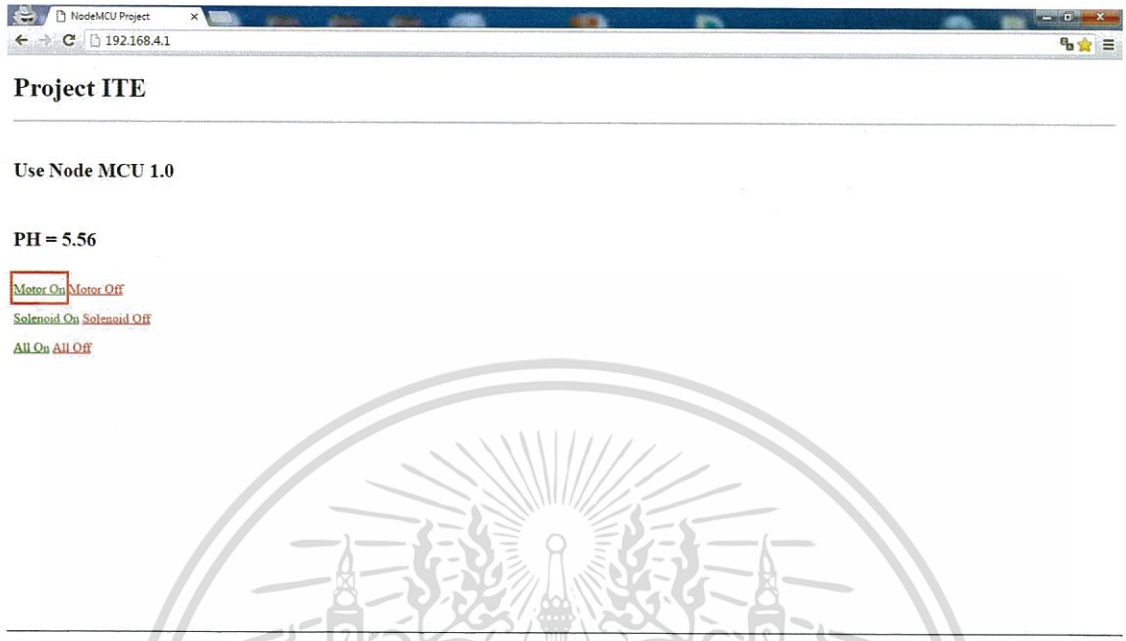
**รูปที่ 4.10** หน้าต่าง Web Server ที่ได้สร้างไว้ใน NodeMCU

โดยที่ตรงค่า pH จะเปลี่ยนทุกครั้งหลังจากกด Refresh หน้าเว็บไซต์ ดังรูปที่ 4.11

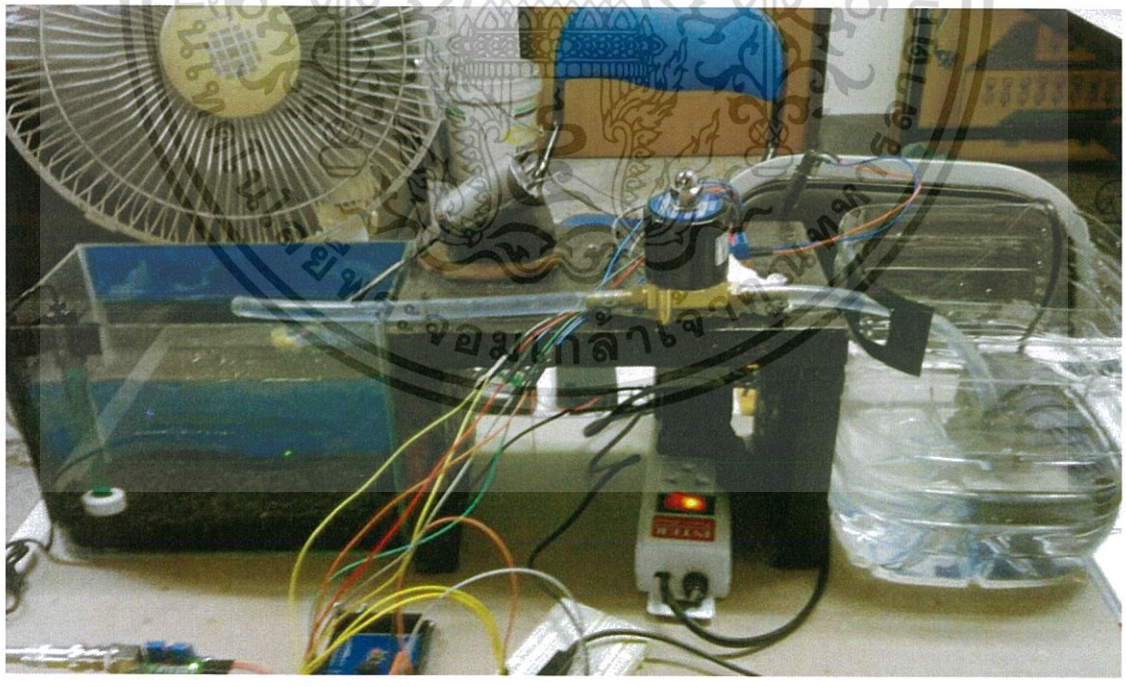



**รูปที่ 4.11** ค่า pH เปลี่ยนไป

หลังจากนั้นสามารถสั่งให้เครื่องต่างๆ ทำงานได้โดยจะสั่งให้มอเตอร์ทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม Motor On ดังรูปที่ 4.12 ซึ่งจะทำให้มอเตอร์ทำงานดังรูปที่ 4.13

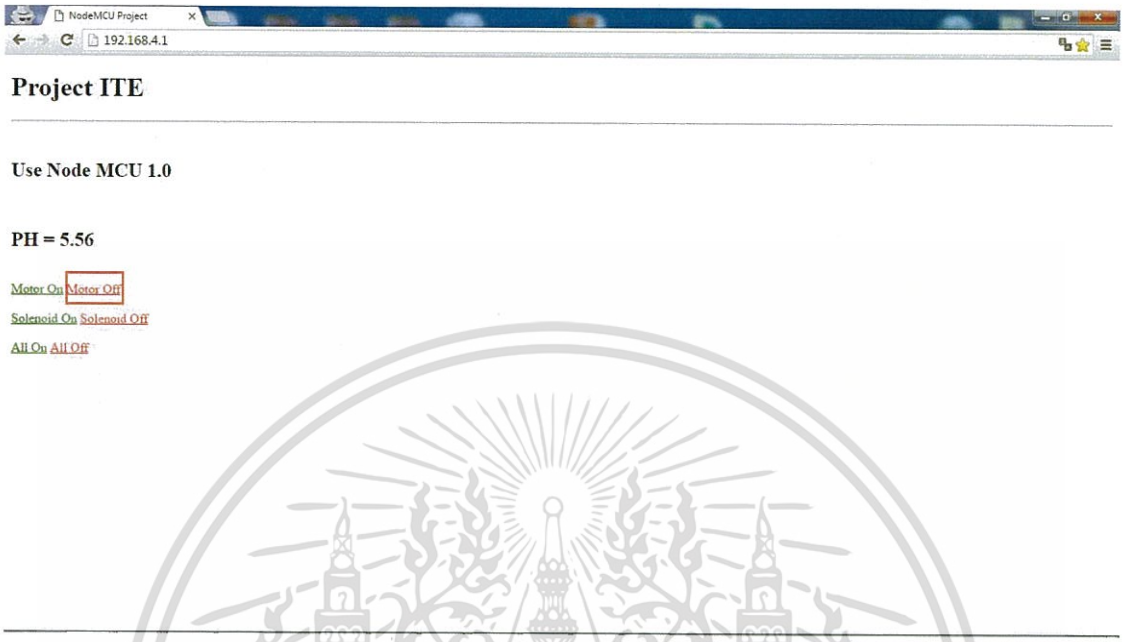



**รูปที่ 4.12** กดปุ่ม Motor On ที่หน้าเว็บไซต์

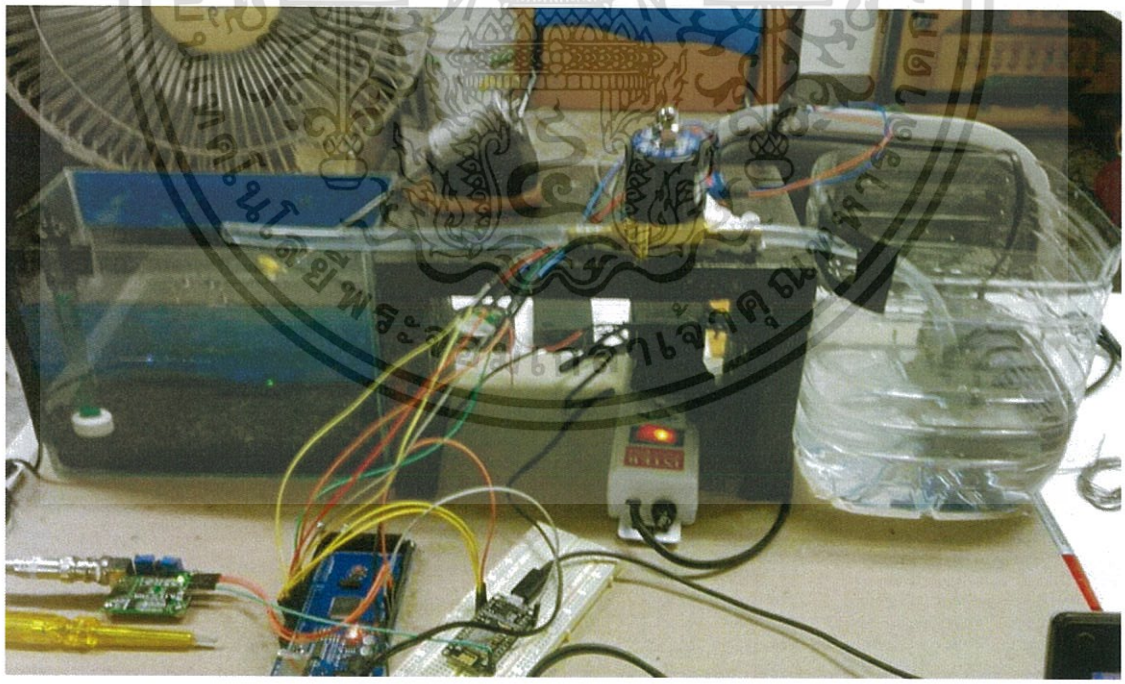


**รูปที่ 4.13** มอเตอร์ทำงาน

ต่อไปจะสั่งให้มอเตอร์หยุดทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม Motor Off ดังรูปที่ 4.14 ซึ่งจะทำให้มอเตอร์หยุดทำงานดังรูปที่ 4.15

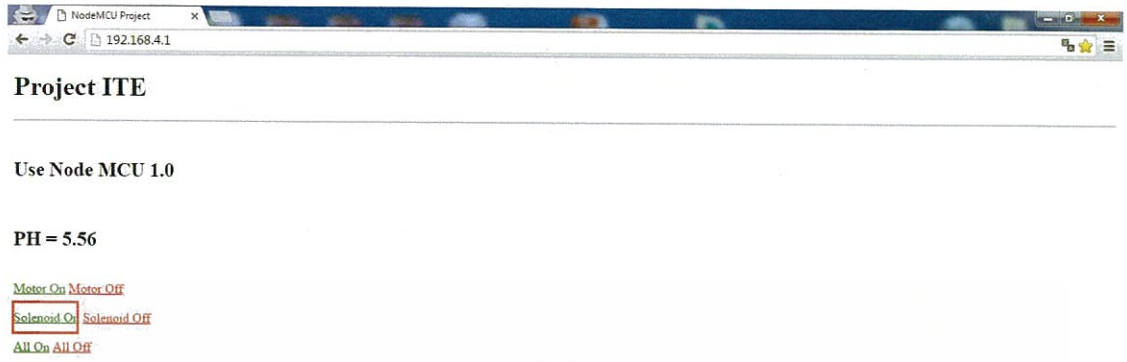



**รูปที่ 4.14** กดปุ่ม Motor Off ที่หน้าเว็บไซต์

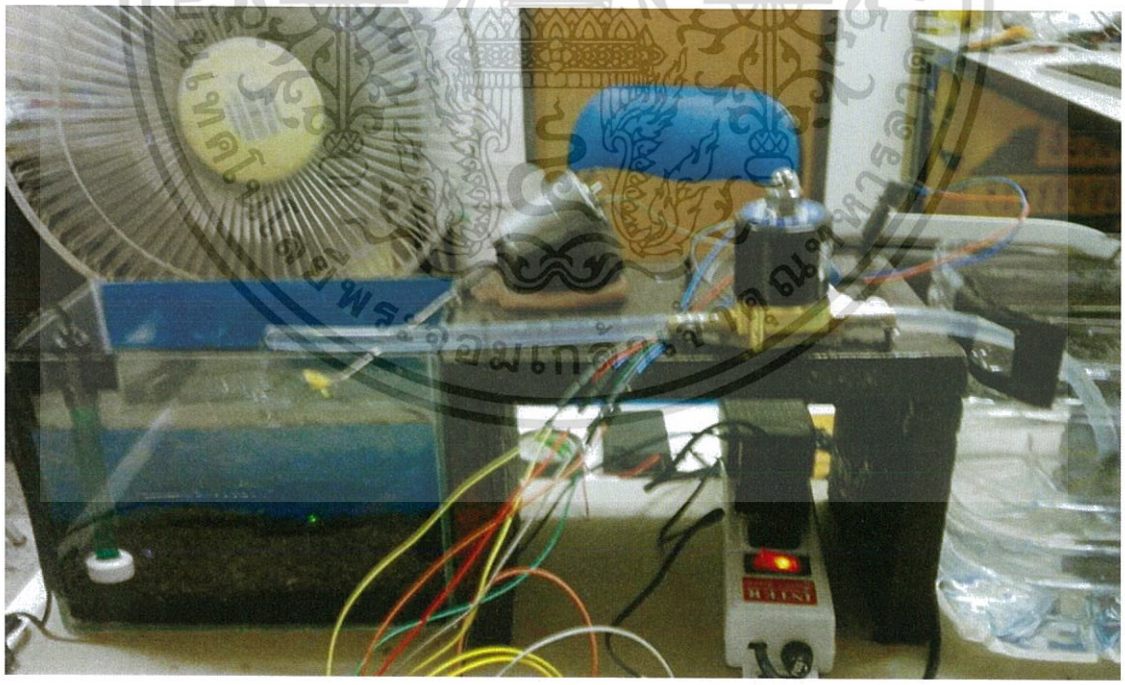


**รูปที่ 4.15** มอเตอร์หยุดทำงาน

ต่อไปจะสั่งให้วาล์วโซลินอยด์ทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม Solenoid On ดังรูปที่ 4.16 ซึ่งจะทำให้วาล์วโซลินอยด์ทำงานดังรูปที่ 4.17

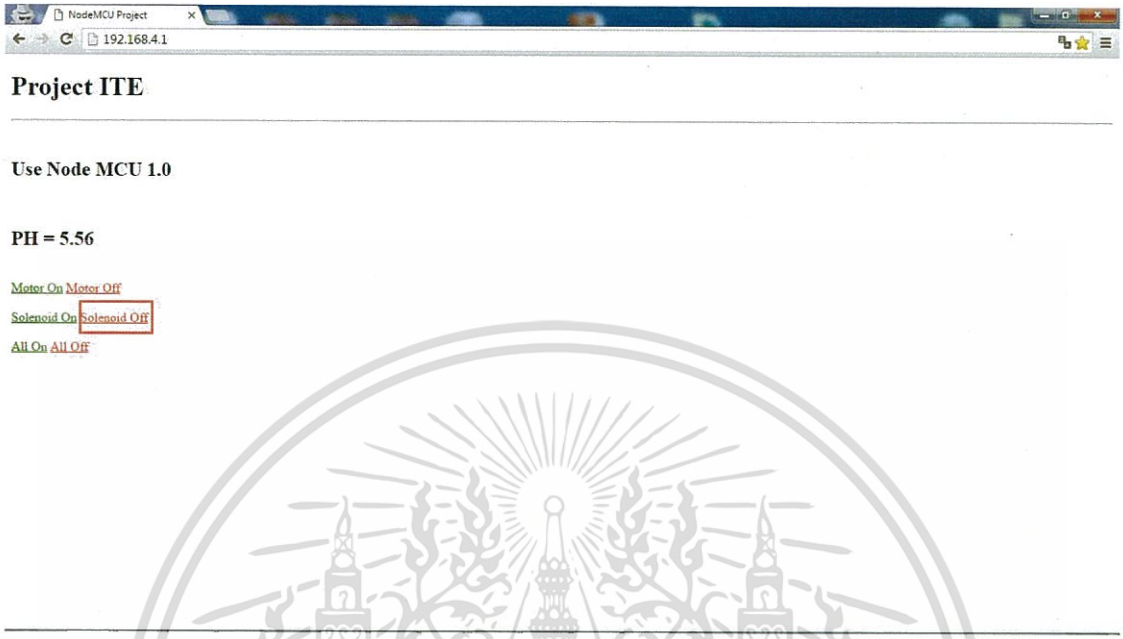



**รูปที่ 4.16** กดปุ่ม Solenoid On ที่หน้าเว็บไซต์

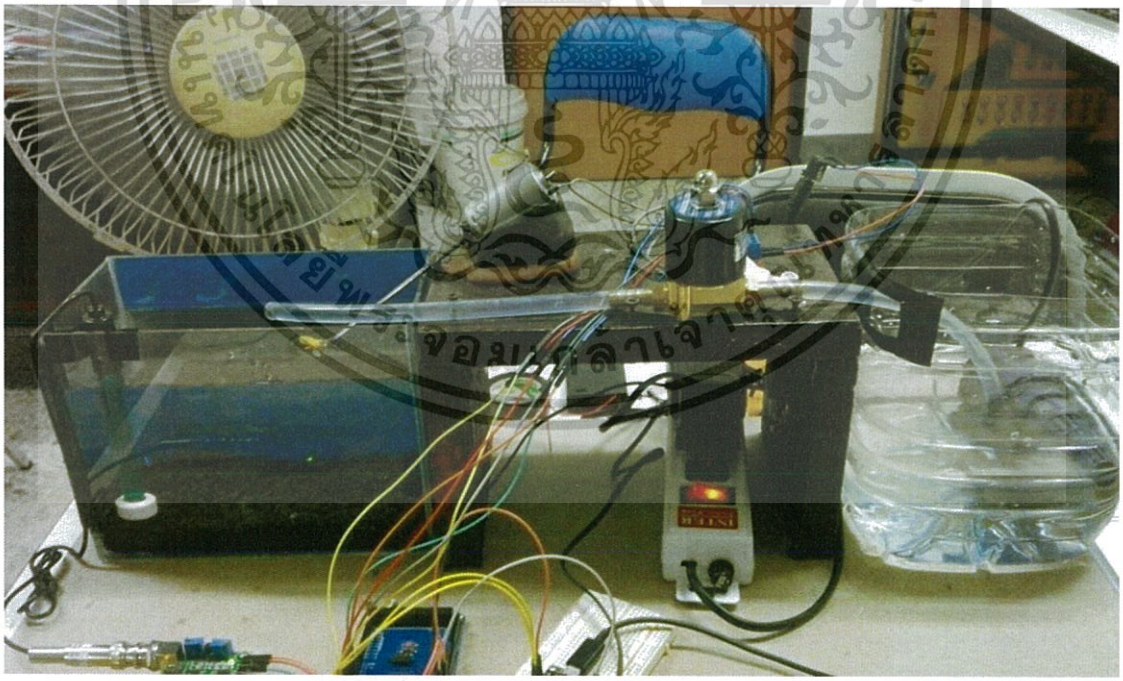


**รูปที่ 4.17** วาล์วโซลินอยด์ทำงาน

ต่อไปจะสั่งให้วาล์วโซลินอยด์หยุดทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม Solenoid Off ดังรูปที่ 4.18 ซึ่งจะทำให้วาล์วโซลินอยด์หยุดทำงานดังรูปที่ 4.19

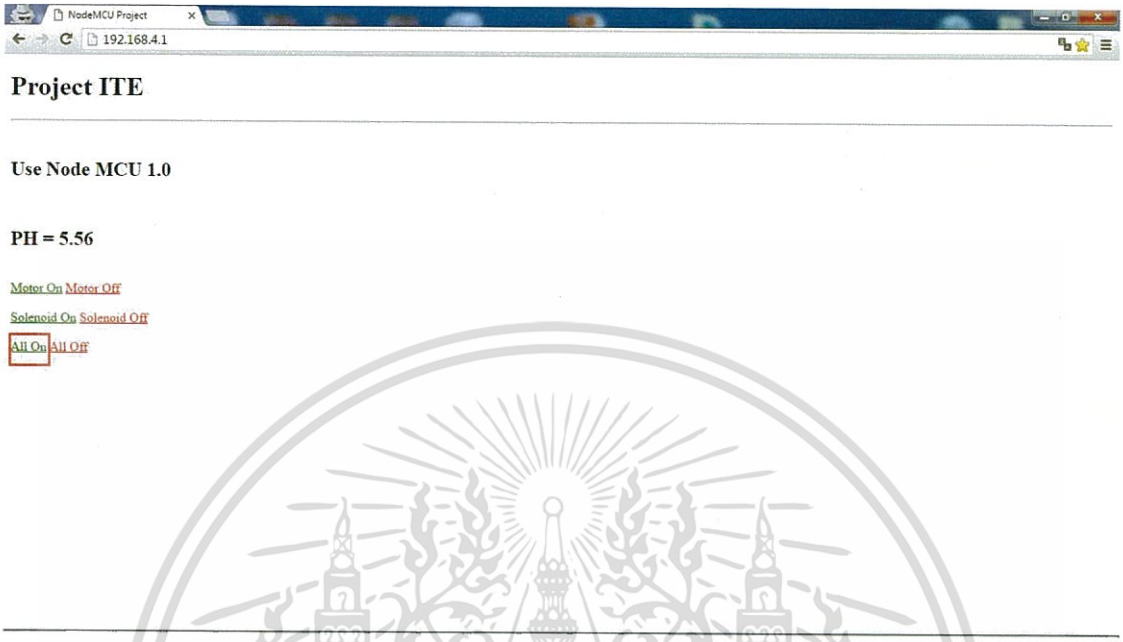



**รูปที่ 4.18** กดปุ่ม Solenoid Off ที่หน้าเว็บไซต์

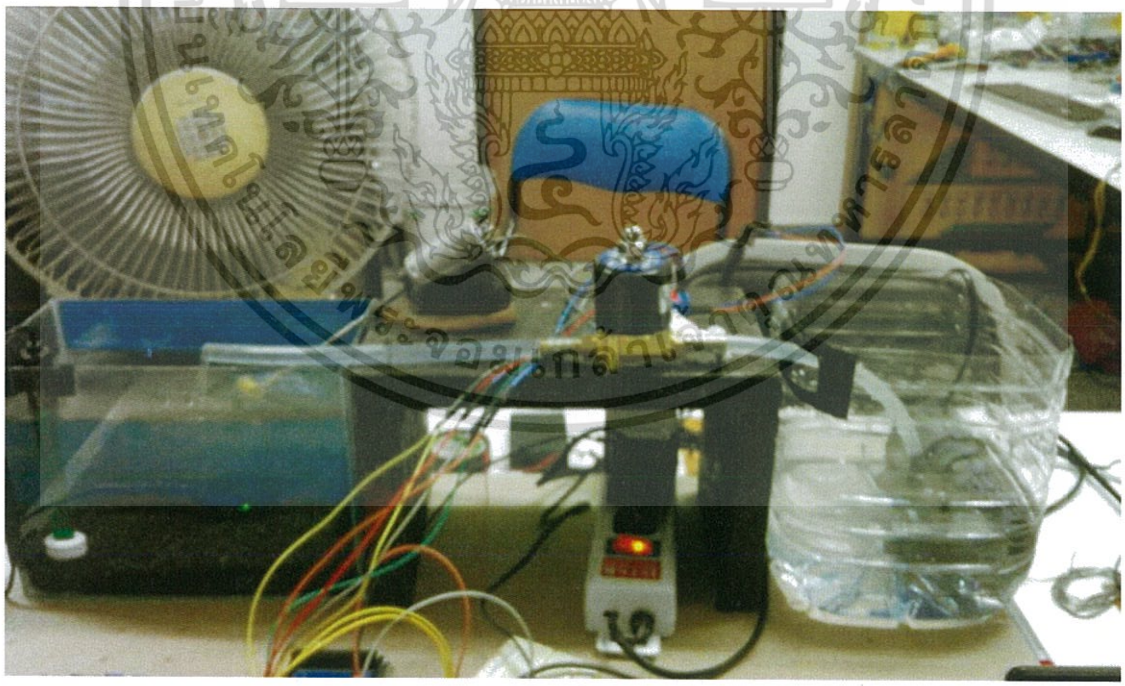


**รูปที่ 4.19** วาล์วโซลินอยด์หยุดทำงาน

ต่อไปจะสั่งให้มอเตอร์และวาล์วโซลินอยด์ทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม All On ดังรูปที่ 4.20 ซึ่งจะทำให้มอเตอร์และวาล์วโซลินอยด์ทำงานดังรูปที่ 4.21

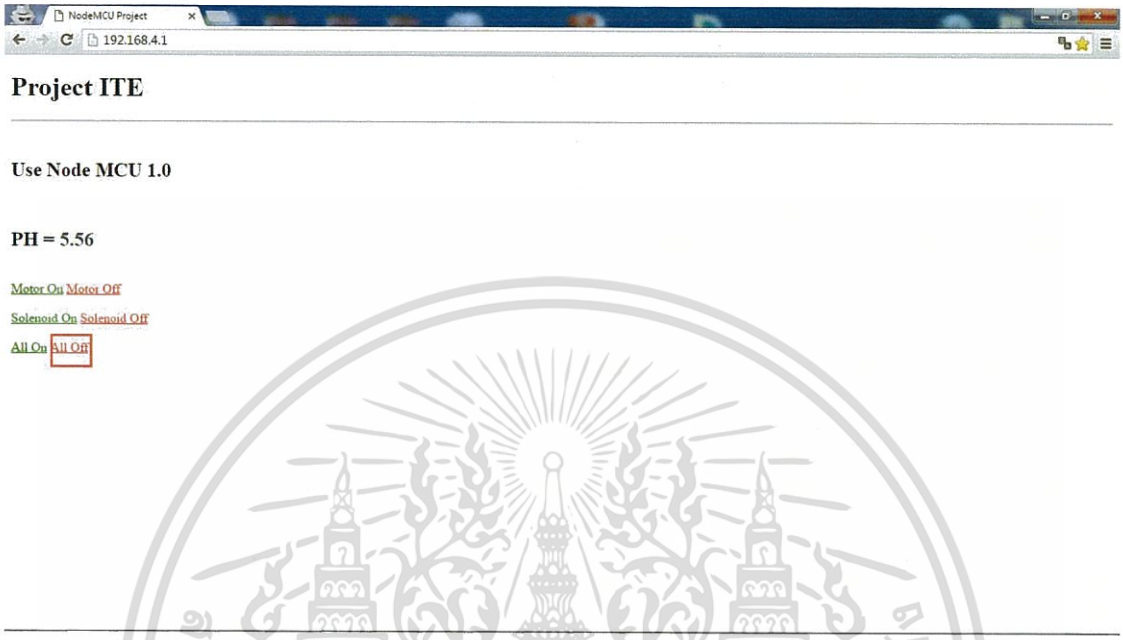



**รูปที่ 4.20** กดปุ่ม All On ที่หน้าเว็บไซต์

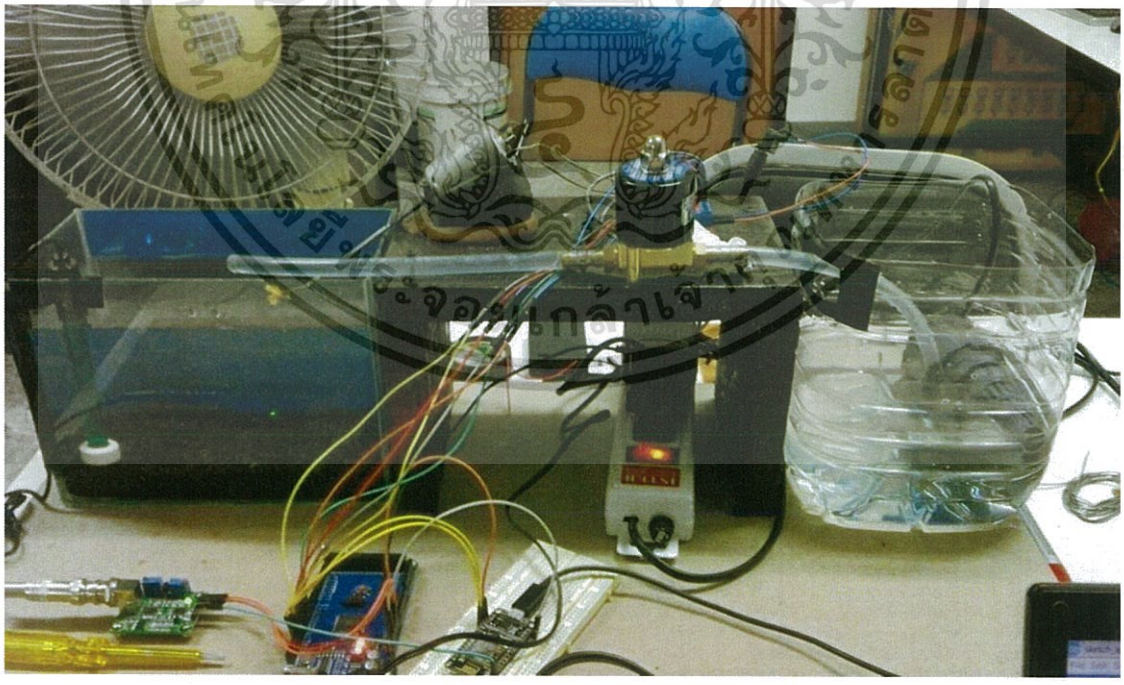


**รูปที่ 4.21** มอเตอร์และวาล์วโซลินอยด์ทำงาน

ต่อไปจะสั่งให้มอเตอร์และวาล์วโซลินอยด์หยุดทำงานได้โดยกดปุ่มตรงหน้าเว็บไซต์ซึ่งในที่นี้จะกดปุ่ม All Off ดังรูปที่ 4.22 ซึ่งจะทำให้มอเตอร์และวาล์วโซลินอยด์หยุดทำงานดังรูปที่ 4.23




**รูปที่ 4.22** กดปุ่ม All Off ที่หน้าเว็บไซต์



**รูปที่ 4.23** มอเตอร์และวาล์วโซลินอยด์หยุดทำงาน

# บทที่ 5
# สรุปและวิจารณ์ผลการทดลอง

## 5.1 บทสรุปโครงงาน

โครงงานนี้ นำเสนอในเรื่องการออกแบบ และคิดค้นอุปกรณ์เพื่อใช้ในการวัดค่า pH ในน้ำ โดยวัดจากโมดูล pH เซนเซอร์และแสดงค่าผ่านทางหน้าเว็บไซต์ที่ NodeMCU สร้างขึ้นมา หลังจากนั้นสามารถสั่งให้มอเตอร์และวาล์วโซลินอยด์ทำงานและหยุดการทำงานได้จากหน้าเว็บไซต์ซึ่งตัวอุปกรณีมีระบบสั่งงานด้วยตัวเองถ้าค่า pH เกินกว่าค่าที่กำหนดได้ แต่อย่างไรก็ตามความเสถียร์ของ NodeMCU ในการทำให้ตัวเองเป็น Wi-Fi Access Point มีน้อยมากเนื่องจากหลังเราทำการเชื่อมต่อกับ Access Point ที่ NodeMCU สร้างแล้วบ้างช่วงจะมีโอกาสที่จะหลุดการเชื่อมต่อ และตัวโมดูล pH เซนเซอร์บ้างที่จะวัดค่าเพี้ยนจากปกติเป็นบางครั้ง

## 5.2 ปัญหาที่พบในระหว่างดำเนินงาน

1. ความเสถียร์ของ NodeMCU ในการทำตัวเป็น Wi-Fi Access Point มีน้อยเพราะการเชื่อมต่อกับ Access Point ที่ NodeMCU สร้างแล้วบางช่วงจะมีโอกาสที่จะหลุดการเชื่อมต่อ
2. ส่วนประกอบในอุปกรณ์บางชิ้นส่วน ยังไม่มีประสิทธิภาพ และตอบสนองตามความต้องการของผู้จัดทำ
3. มีข้อจำกัดด้านขีดความสามารถของอุปกรณ์ที่ไม่สามารถให้ค่าได้เที่ยงตรงพอเท่าที่ควรและมีขนาดใหญ่กว่าที่ได้ออกแบบไว้ อันเนื่องมาจากต้นทุนของการผลิต
4. อุปกรณีมีขนาดใหญ่ และมีชิ้นส่วนที่ไม่จำเพาะเจาะจงสำหรับการใช้งานทางด้านนี้

## 5.3 แนวทางแก้ไข และพัฒนาต่อ

1. ปรับปรุงการแสดงผล และวัดค่าให้มีความเที่ยงตรงแม่นยำขึ้น
2. พัฒนาและลดขนาดของอุปกรณ์ให้มีขนาดเล็กลง
3. ต้องมีทุนมากกว่านี้ เพื่อสามารถจัดหาอุปกรณ์ที่มีขีดความสามารถสูงขึ้นได้

# ภาคผนวก

# ภาคผนวก ก

Poster

# ภาคผนวก ก

## Poster

CE-072

# TECHNOLOGY FOR WASTEWATER TREATMENT IN AQUACLUTURE PONE

**Yodkunpol Nummode and Akkharaphan Jantharo**
**Advisor: Asst.Prof. Paisan Sithiyopasakul**
**Co-Advisor: Sorapong Wachirarattanapornkul**
**Department of Information Engineering**
E-mail: Yodkunpolfirst@gmail.com, Au_Gol@hotmail.com

## Abstract

This project is about creating the device to measure the pH value in the water and display on web by using WIFI to send data . In this device, the major components are Microcontroller Arduino Mega 2560 (ATmega2560), pH Sensor Module, ESP8266 NodeMCU v.1.0 and pH Probe . It has Process , When pH Probe measure water , pH Probe will send electromotive force value to pH Sensor Module to Change value electromotive force to a pH value then it will send value to the ESP8266 NodeMCU v.1.0 and. then ESP8266 NodeMCU v.1.0 will show pH value by website and can order Motor and Solenoid Value work by website. After that ESP8266 NodeMCU v.1.0 will send value that command in website to the Arduino Mega 2560 to order Drive Motor to order Motor or Solenoid Value work by User need to do . And Can order by themself when the pH value is exceed a predetermined value .

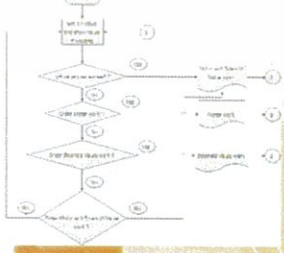
## Introduction

In the Present time . A Person who earns his or her living raise aquatic animal in private areas or other areas but He or She will found water pollution problems . Water pollution will do many animal in water die and loss ecology from this problem . We have idea to create technology for wastewater in Aquaculture Ponds to help solve water pollution problems by determine the value of water as factor that cause wastewater and wastewater treatment make water that aquatic animals can live normally.

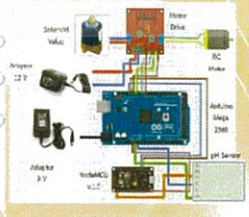
## Methodology

HardWare
- Arduino Mega 2560
- ESP8266 NodeMCU v. 1.0
- pH Sensor Module
- pH Electrode with BNC plug GX-E1
- DC Motor
- Solenoid
- Motor Drive
- Adapter 9 V and 12 V
- WP1250 SOBO Aquarium Power Head

Software
- Arduino IDE



Flowchart System



System Overview

## Results

Model

Front Web Page

Motor On | Solenoid On | All On



## Conclusion

This project is presented in terms of design and invented a device used to measure the pH in the water. As measured by pH Sensor Module and displayed pH value via the website that NodeMCU made and can order Motor and Solenoid Value to work and stop via website that NodeMCU was made. The Device can order Motor and Solenoid Value work by themself when the pH value over the exceed . However NodeMCU made itseft as WiFi Access Point is Unstable because when we connected to the Access Point that the NodeMCU made. Sometime it have a chance to disconnected from the NodeMCU Access Point .

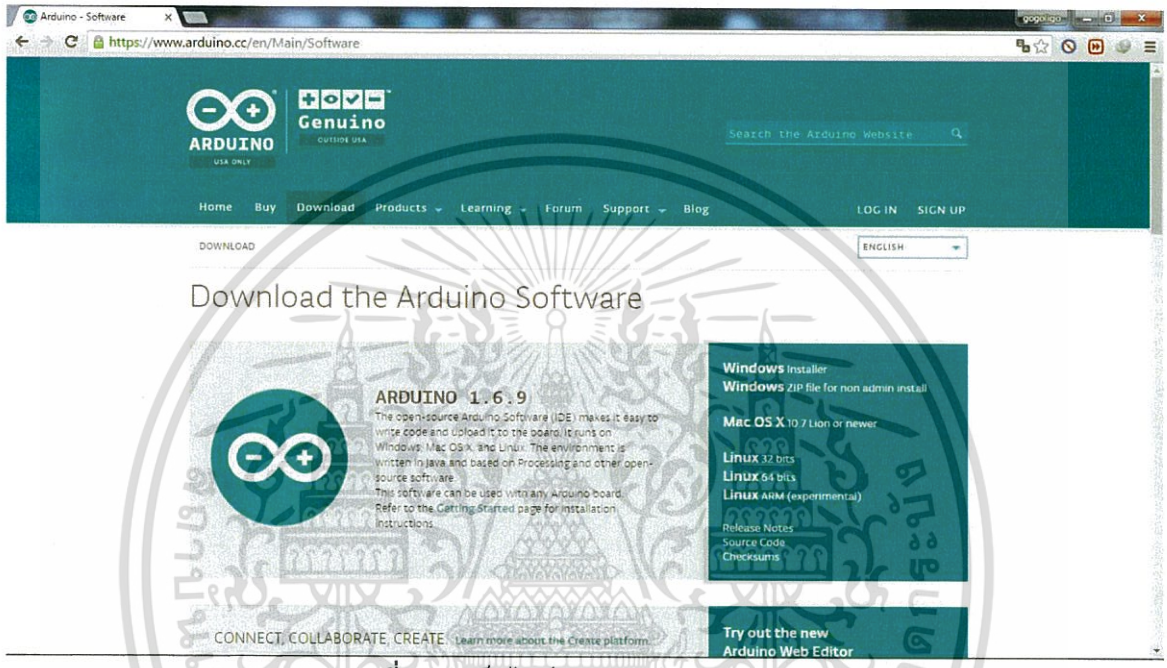
## References

[1] cnxsoft. 2557. NodeMCU is both a Breadboard-Friendly ESP8266 Wi-Fi Board and a LUA based Firmware. [Online]. Available : http://www.cnx-software.com/2015/04/18/nodemcu-is-both-a-breadboard-friendly-esp8266-wi-fi-board-and-a-lua-based-firmware
[2] MO Memoir. 2552. pH Probe MO Memoir. [Online]. Available : http://tamagozzilla.blogspot.com/2009/08/mo-memoir-ednesday-1-july- 2552-ph-probe.html
[3] Arduino. Arduino - SoftwareSerial. [Online]. Available : https://www.arduino.cc/en/Reference/softwareSerial

# ภาคผนวก ข
# การติดตั้งโปรแกรม Arduino

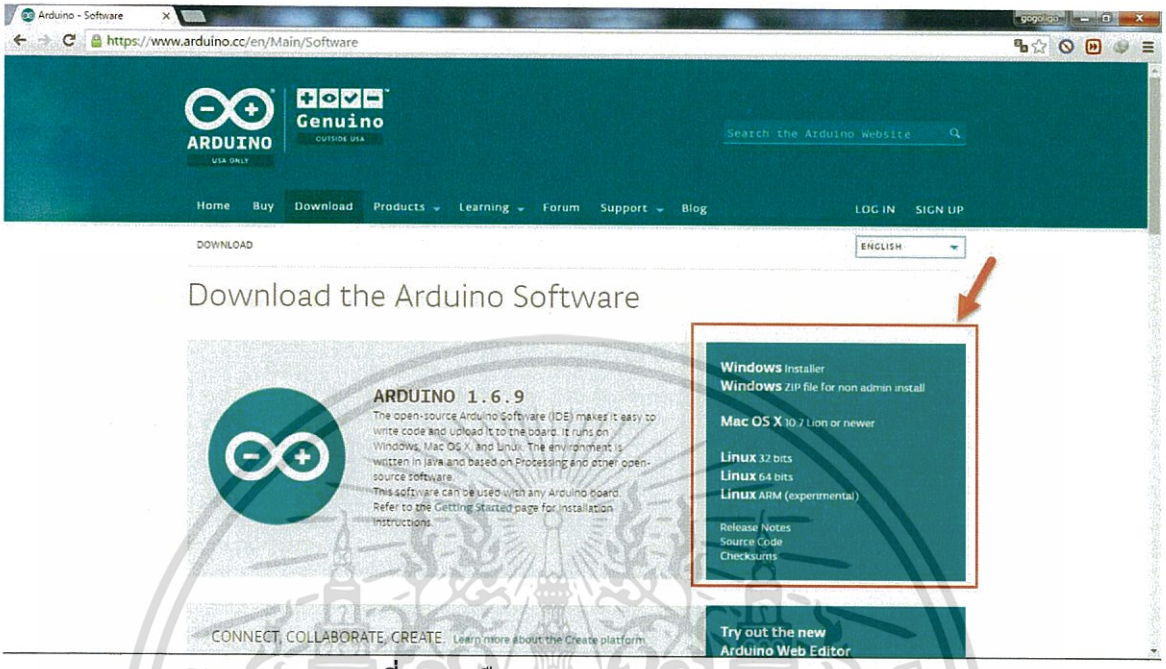
# ภาคผนวก ข
# การติดตั้งโปรแกรม Arduino

1. ทำการดาวน์โหลดตัวติดตั้งโปรแกรม Arduino โดยเข้าไปที่ URL นี้จาก www.arduino.cc/en/Main/Software ดังรูปที่ ข.1



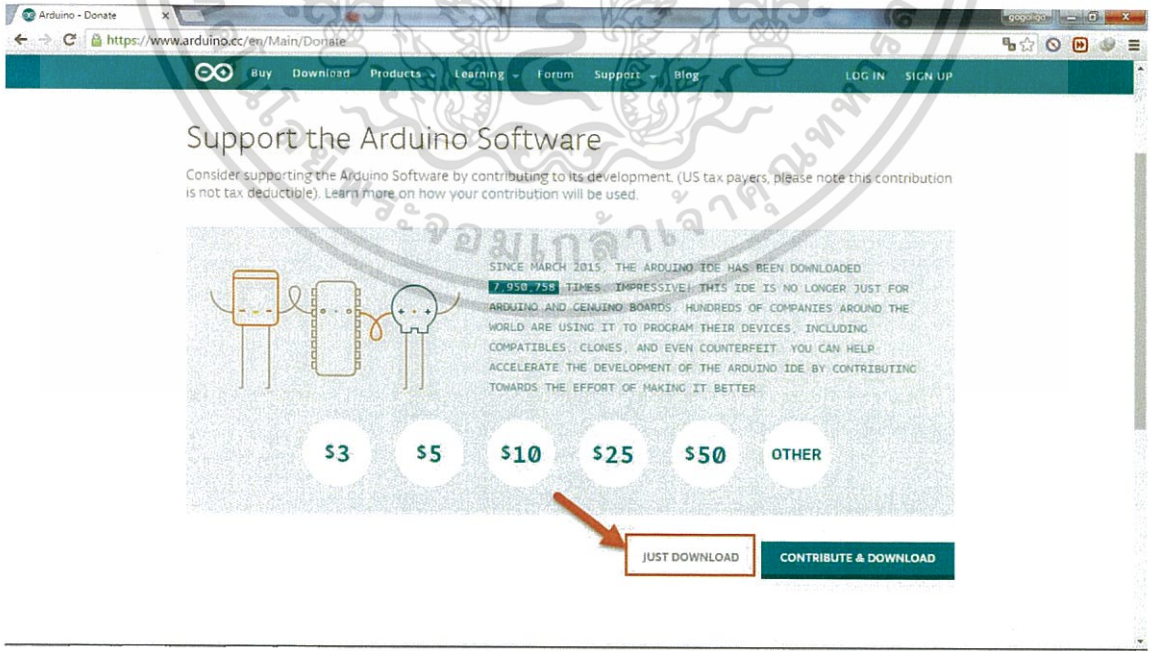

**รูปที่ ข.1** เว็บไซต์ www.arduino.cc

2. ให้ทำการเลือกโปรแกรมตาม OS(Operating System) ที่ใช้อยู่ในที่นี้ได้ใช้ Window 7 จึงได้เลือก Windows Installer



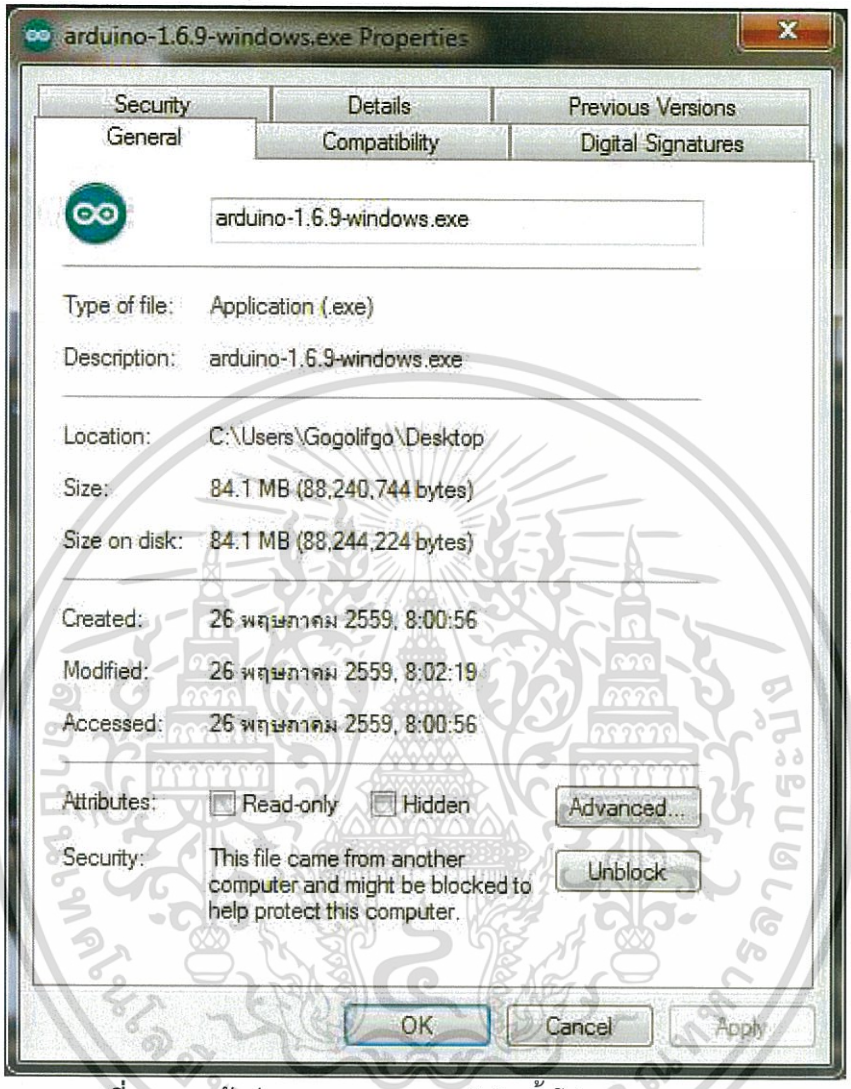

**รูปที่ ข.2** เลือก Windows Installer

3. หลังคลิกเข้าไปแล้วจะได้หน้าต่างดังรูป ข.3 ให้ทำการเลือกแถบ JUST DOWNLOAD ดังรูปที่ ข.3



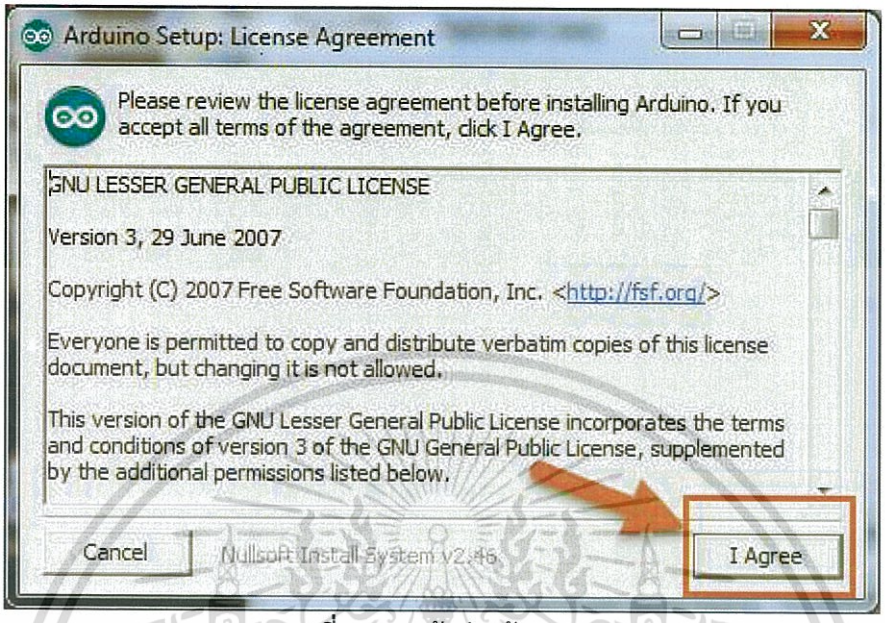

**รูปที่ ข.3** เลือก JUST DOWNLOAD

4. หลังจากนั้นจะได้โปรแกรมติดตั้งโปรแกรม Arduino มาโดยมีขนาดดังรูปที่ ข.4 (สามารถเปลี่ยนแปลงได้ตาม version)



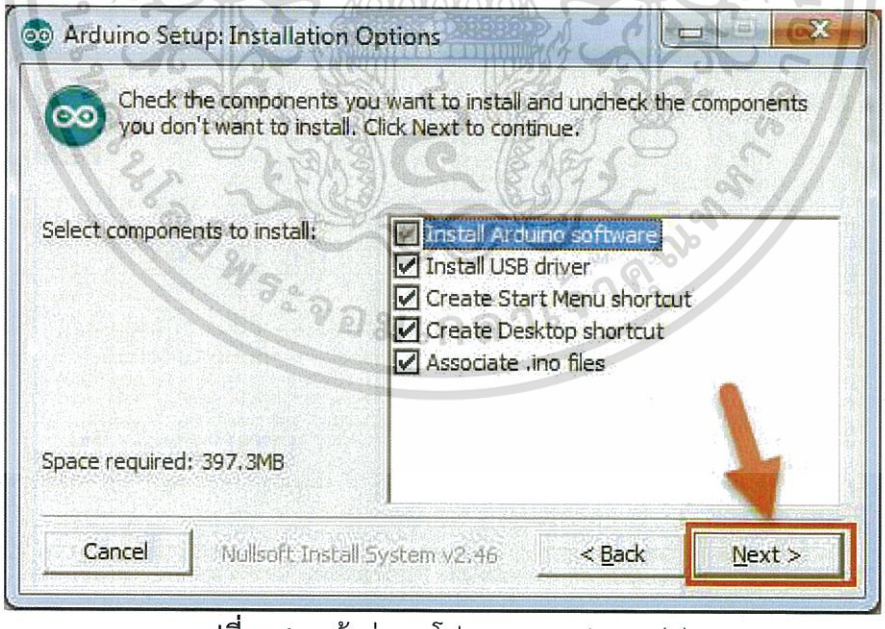

**รูปที่ ข.4** หน้าต่างแสดงขนาดของตัวติดตั้งโปรแกรม Arduino

5. double-click ไฟล์ arduino-1.6.9-windows.exe เพื่อทำการติดตั้งจะได้หน้าต่างดังรูปที่ ข.5 ซึ่งเป็นหน้าต่างข้อตกลงของโปรแกรมให้กดปุ่ม I Agree



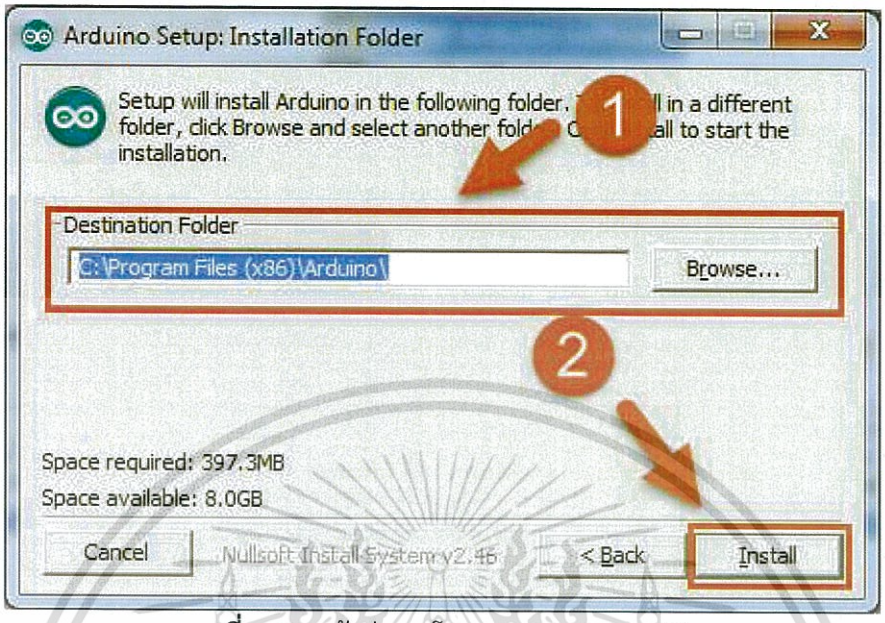

**รูปที่ ข.5** หน้าต่างข้อตกลง

6. ทำการเลือกทุกหัวข้อแล้วกดปุ่ม Next ดังรูปที่ ข.6



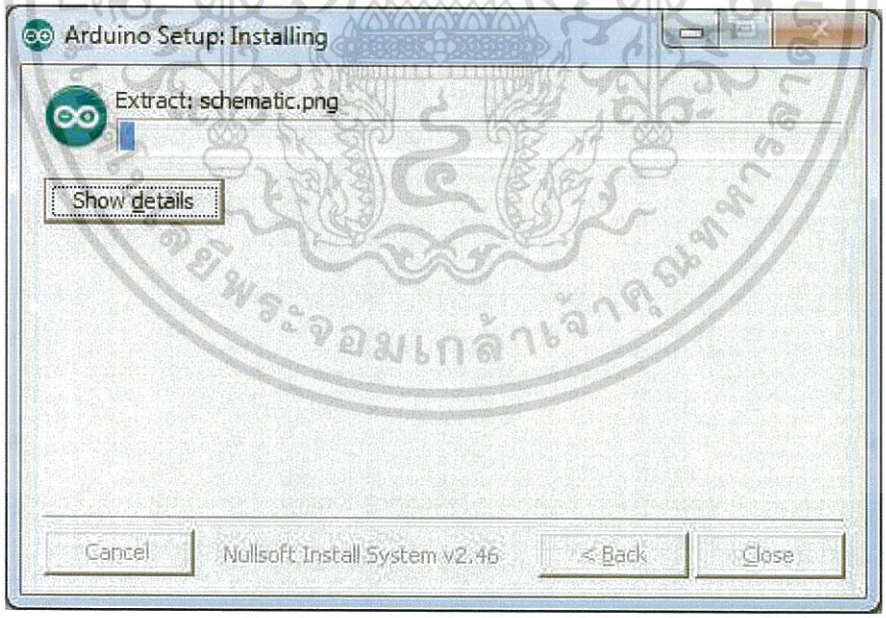

**รูปที่ ข.6** หน้าต่างลงโปรแกรม Arduino (1)

7. เลือกพื้นที่ที่ต้องการจัดเก็บโปรแกรม Arduino เสร็จแล้วกดปุ่ม Install ดังรูปที่ ข.7



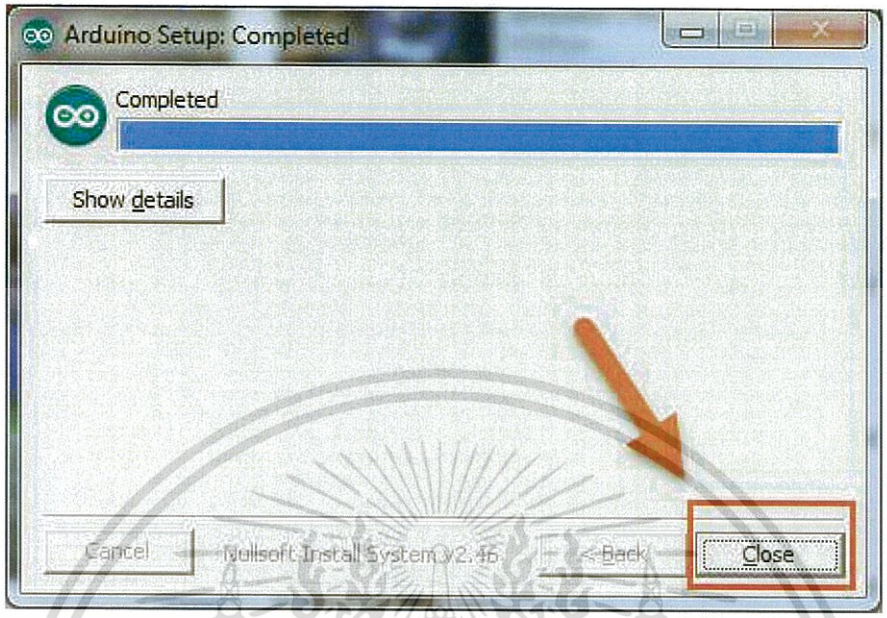

**รูปที่ ข.7** หน้าต่างลงโปรแกรม Arduino (2)

8. โปรแกรมติดตั้งจะทำการลงโปรแกรม Arduino ดังรูปที่ ข.8




**รูปที่ ข.8** หน้าต่างลงโปรแกรม Arduino (3)

9. เมื่อลงโปรแกรมเสร็จเรียบร้อย กดปุ่ม Close ดังรูปที่ ข.9




**รูปที่ ข.9** หน้าต่างลงโปรแกรม Arduino (4)

10. สามารถเข้าโปรแกรม Arduino ได้ผ่านหน้า Desktop โดย double-click icon ดังรูปที่ ข.10

**รูปที่ ข.10** ไอคอนโปรแกรม Arduino

# ภาคผนวก ค
# วิธีใช้งาน

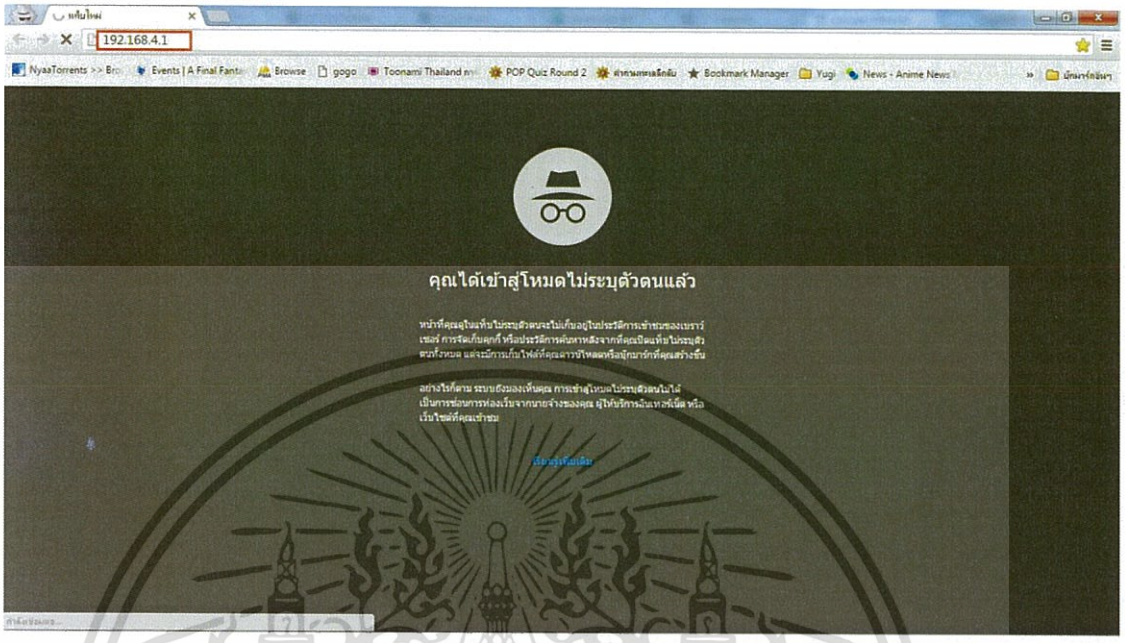
# ภาคผนวก ค
# วิธีใช้งาน

**การสมัครใช้บริการเว็บแอพพลิเคชันเพื่อเก็บค่าข้อมูลจากเซ็นเซอร์**

1. เชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo ที่สร้างขึ้นมาจาก NodeMCU v.1.0 ดังรูปที่ ค.1

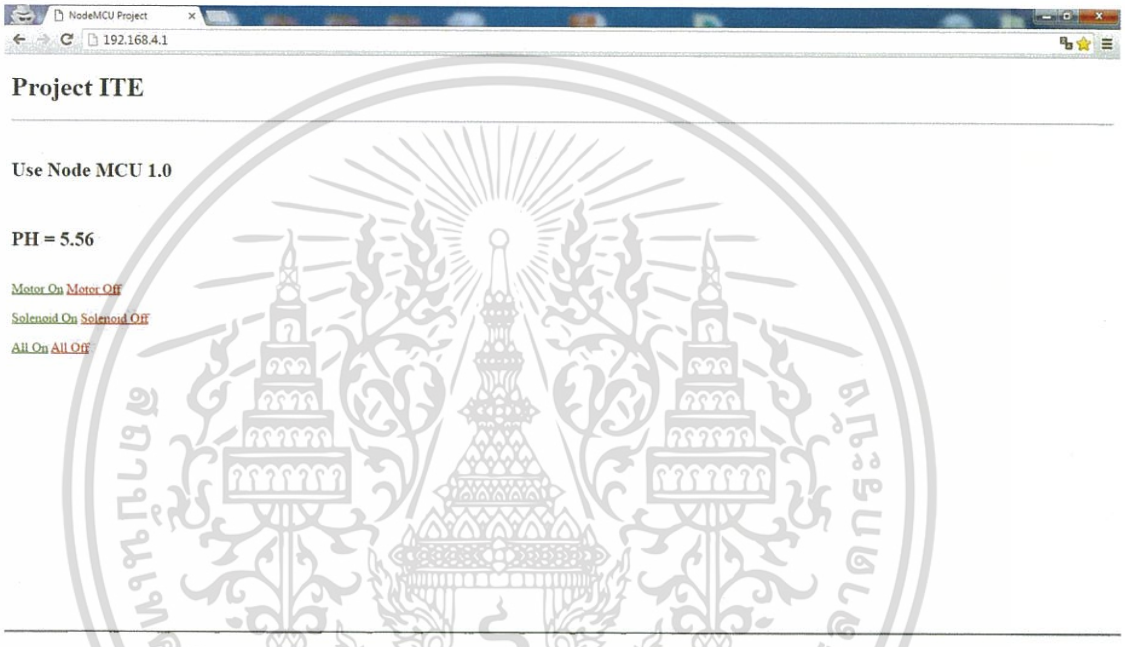
**รูปที่ ค.1** เชื่อมต่อกับ Wi-Fi ชื่อ gogolifgo

2. เปิด Web Browser ขึ้นมาและตรงช่อง URL ให้พิมพ์ 192.168.4.1 เพื่อเข้า Web Server ใน NodeMCU




**รูปที่ ค.2** วิธีเข้า Web Server

3. จะได้หน้าต่างดังรูปที่ ค.3 โดยสามารถดูค่า pH ของน้ำได้จาก PH = .... และสามารถสั่งให้มอเตอร์และวาล์วโซลินอยด์ทำงานหรือหยุดทำงานได้จากปุ่มต่างๆ โดยแบ่งได้เป็นดังนี้
   - Motor On = สั่งให้มอเตอร์ทำงาน
   - Motor Off = สั่งให้มอเตอร์หยุดทำงาน
   - Solenoid On = สั่งให้วาล์วโซลินอยด์ทำงาน
   - Solenoid Off = สั่งให้วาล์วโซลินอยด์หยุดทำงาน
   - All On = สั่งให้มอเตอร์และวาล์วโซลินอยด์ทำงาน
   - All Off = สั่งให้มอเตอร์และวาล์วโซลินอยด์หยุดทำงาน



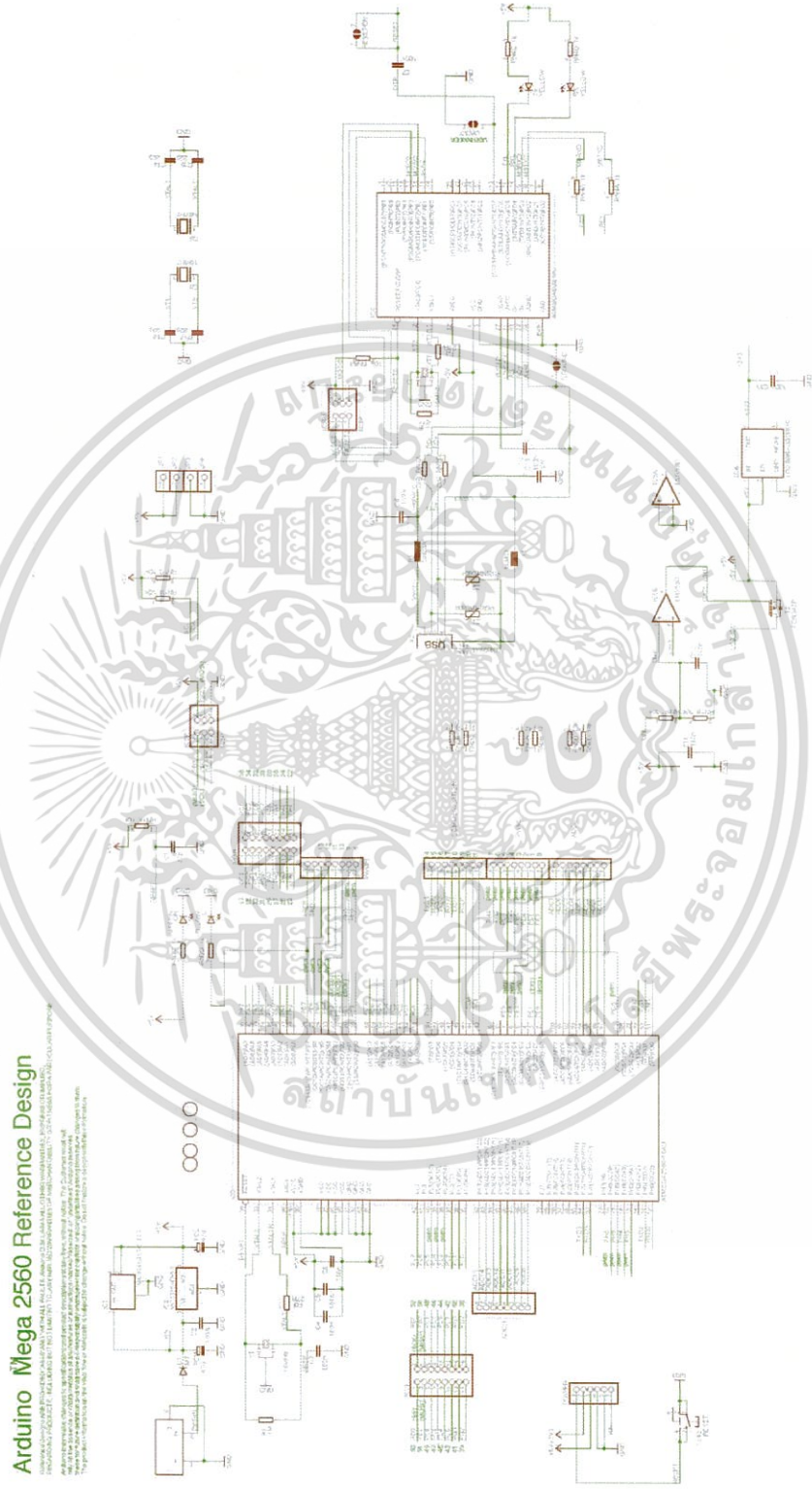

**รูปที่ ค.3** หน้าต่าง Web Server ที่ได้สร้างไว้ใน NodeMCU

# ภาคผนวก ง

## Data Sheet

# ภาคผนวก ง

## Data Sheet



Atmel

# Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V

## 8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash

**DATASHEET**

## Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Fully Green
- Temperature Range:
  - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
  - Active Mode: 1MHz, 1.8V: 500µA
  - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
  - ATmega640V/ATmega1280V/ATmega1281V:
    - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega2560V/ATmega2561V:
    - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega640/ATmega1280/ATmega1281:
    - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
  - ATmega2560/ATmega2561:
    - 0 - 16MHz @ 4.5V - 5.5V

# 1. Pin Configurations

**Figure 1-1.** TQFP-pinout ATmega640/1280/2560

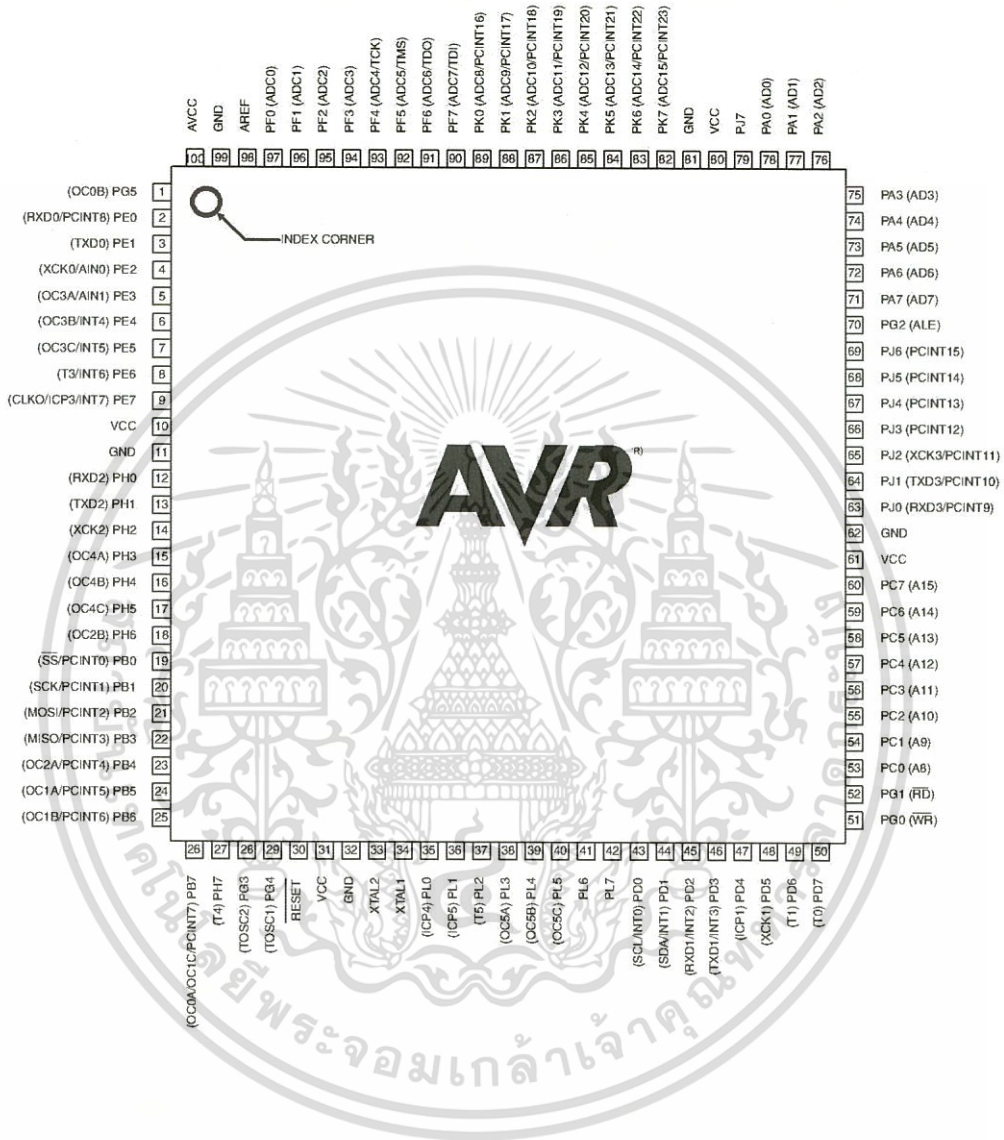


**Figure 1-2.** CBGA-pinout ATmega640/1280/2560

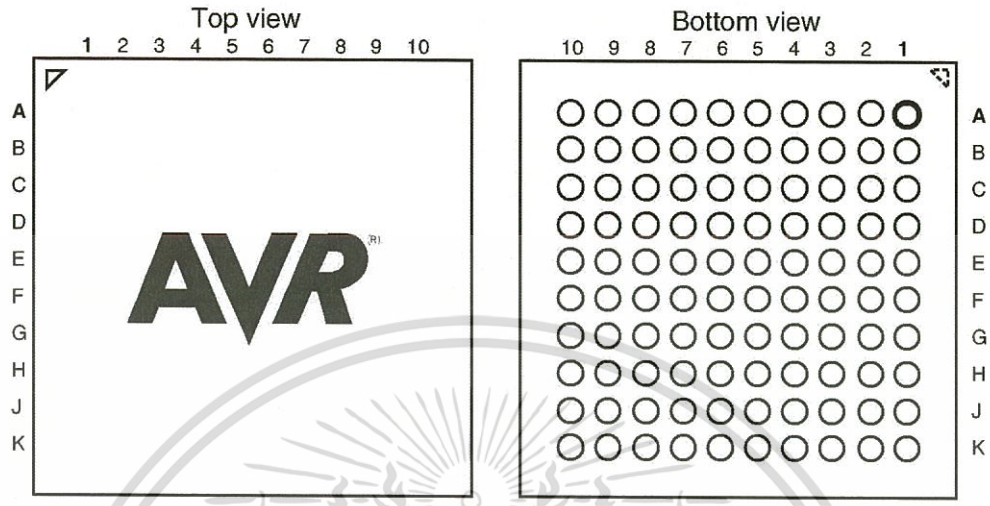
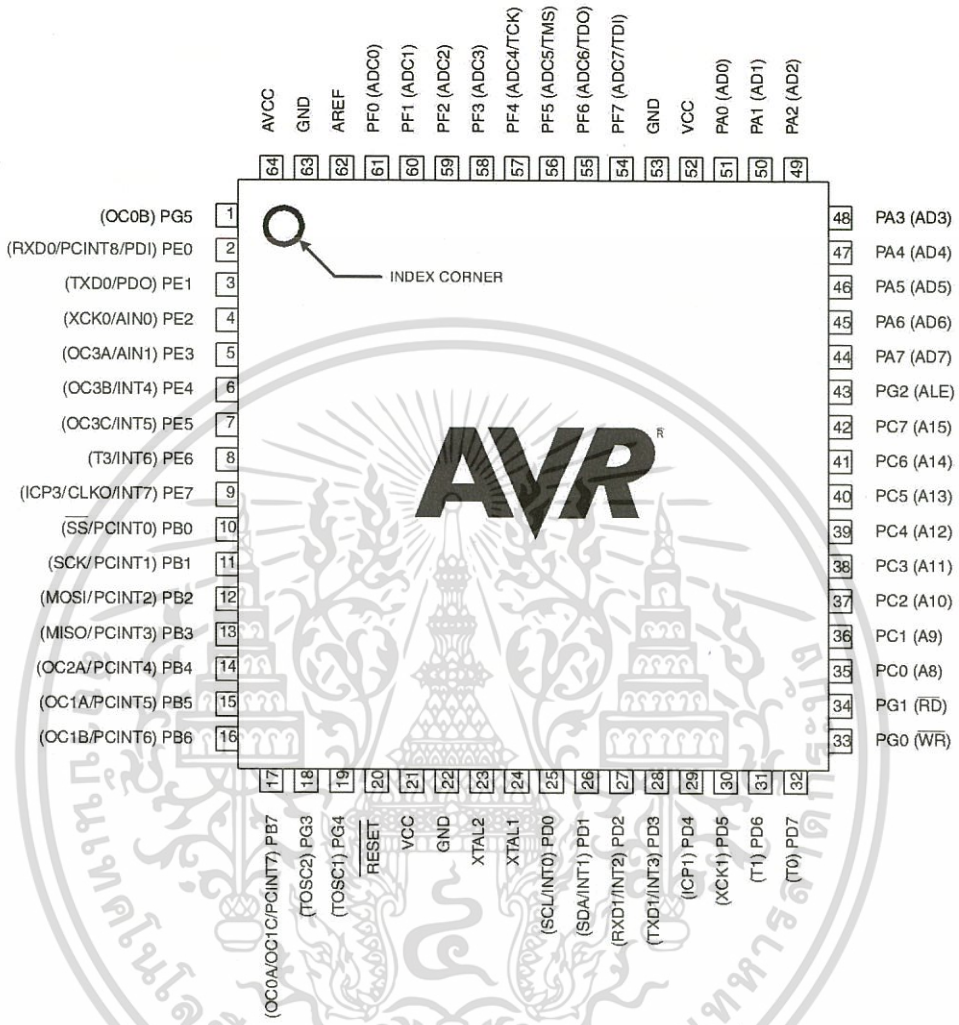



**Table 1-1.** CBGA-pinout ATmega640/1280/2560

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | GND | AREF | PF0 | PF2 | PF5 | PK0 | PK3 | PK6 | GND | VCC |
| B | AVCC | PG5 | PF1 | PF3 | PF6 | PK1 | PK4 | PK7 | PA0 | PA2 |
| C | PE2 | PE0 | PE1 | PF4 | PF7 | PK2 | PK5 | PJ7 | PA1 | PA3 |
| D | PE3 | PE4 | PE5 | PE6 | PH2 | PA4 | PA5 | PA6 | PA7 | PG2 |
| E | PE7 | PH0 | PH1 | PH3 | PH5 | PJ6 | PJ5 | PJ4 | PJ3 | PJ2 |
| F | VCC | PH4 | PH6 | PB0 | PL4 | PD1 | PJ1 | PJ0 | PC7 | GND |
| G | GND | PB1 | PB2 | PB5 | PL2 | PD0 | PD5 | PC5 | PC6 | VCC |
| H | PB3 | PB4 | RESET | PL1 | PL3 | PL7 | PD4 | PC4 | PC3 | PC2 |
| J | PH7 | PG3 | PB6 | PL0 | XTAL2 | PL6 | PD3 | PC1 | PC0 | PG1 |
| K | PB7 | PG4 | VCC | GND | XTAL1 | PL5 | PD2 | PD6 | PD7 | PG0 |

Note: The functions for each pin is the same as for the 100 pin packages shown in Figure 1-1 on page 2.

**Figure 1-3.** Pinout ATmega1281/2561



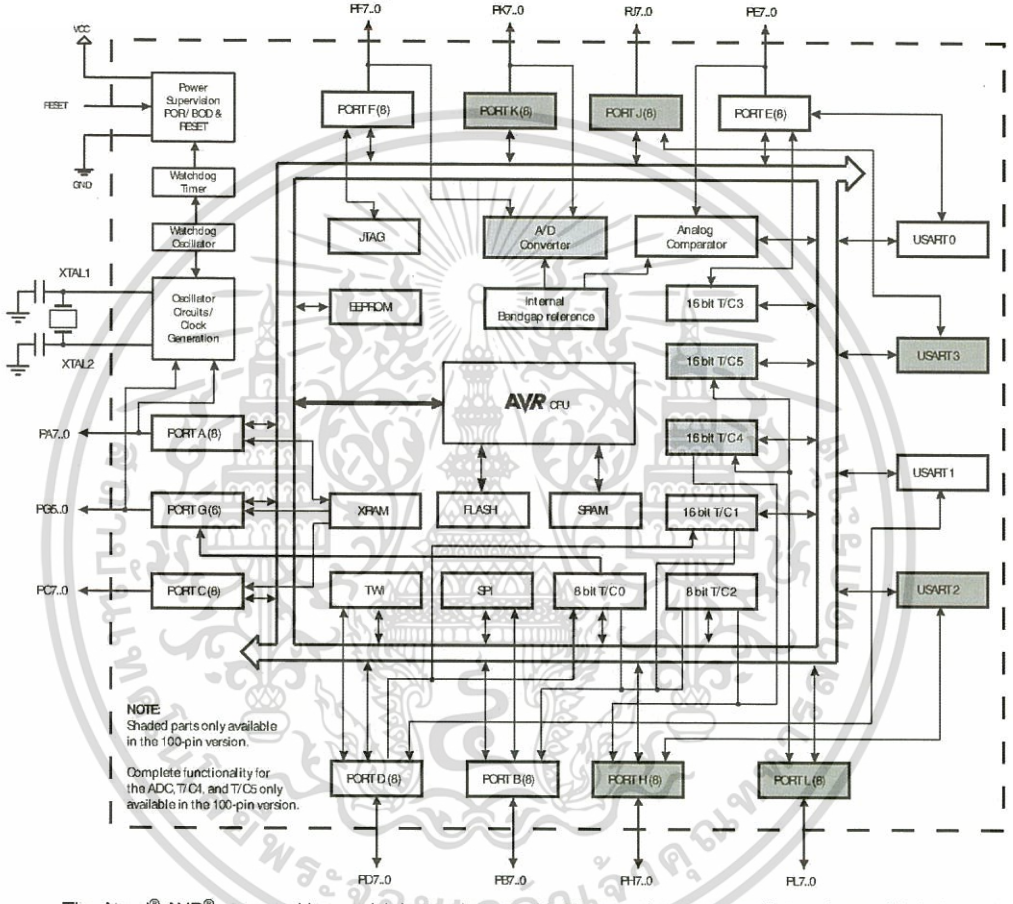

Note: The large center pad underneath the QFN/MLF package is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

# 2. Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## 2.1 Block Diagram

**Figure 2-1.** Block Diagram

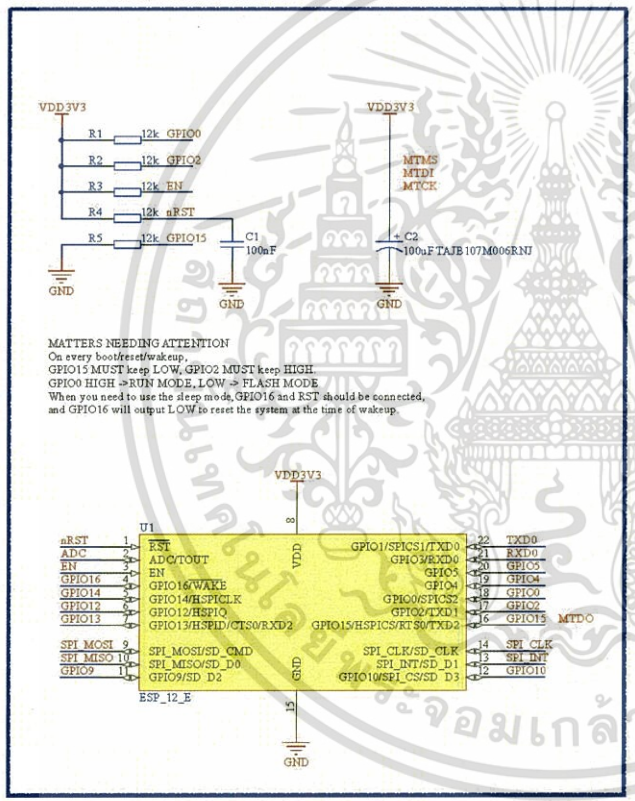


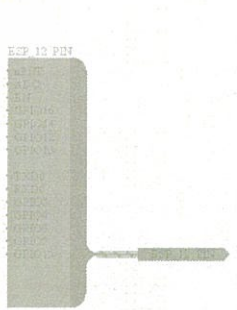

The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

# NODE MCU DEVKIT V1.0

| VER | DATE |
|---|---|
| 1.0 | 25/01/2015 |
| ORGANIZATION | |
| NODE MCU TEAM | |
| WEBSITE | |
| WWW.NODEMCU.COM | |

# ESP-12 CORE




| VER | DATE |
|---|---|
| 1.0 | 20/11/2014 |
| ORGANIZATION | |
| NODE MCU TEAM | |
| WEBSITE | |
| WWW.NODEMCU.COM | |

# L298

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

Multiwatt15

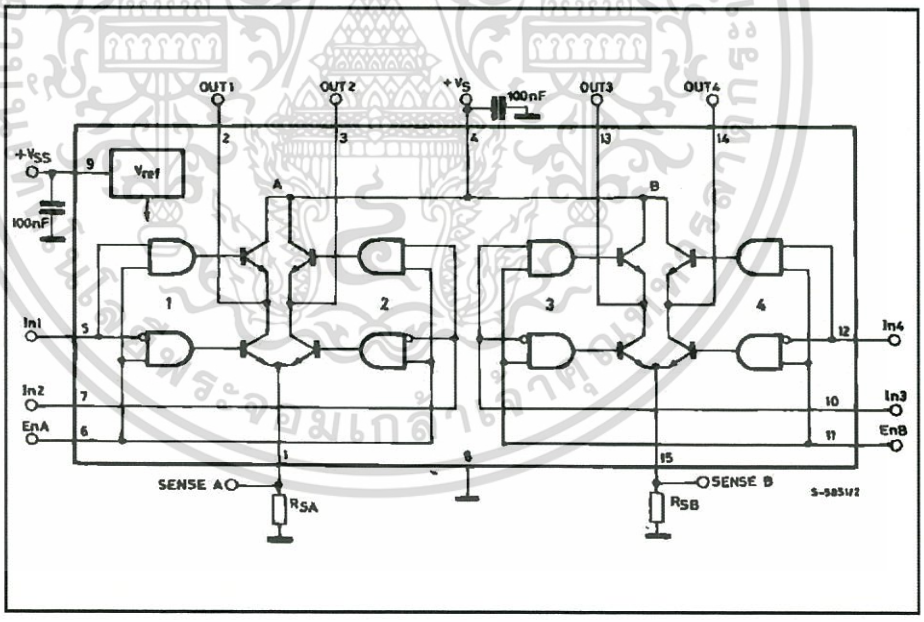
PowerSO20

**ORDERING NUMBERS** : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

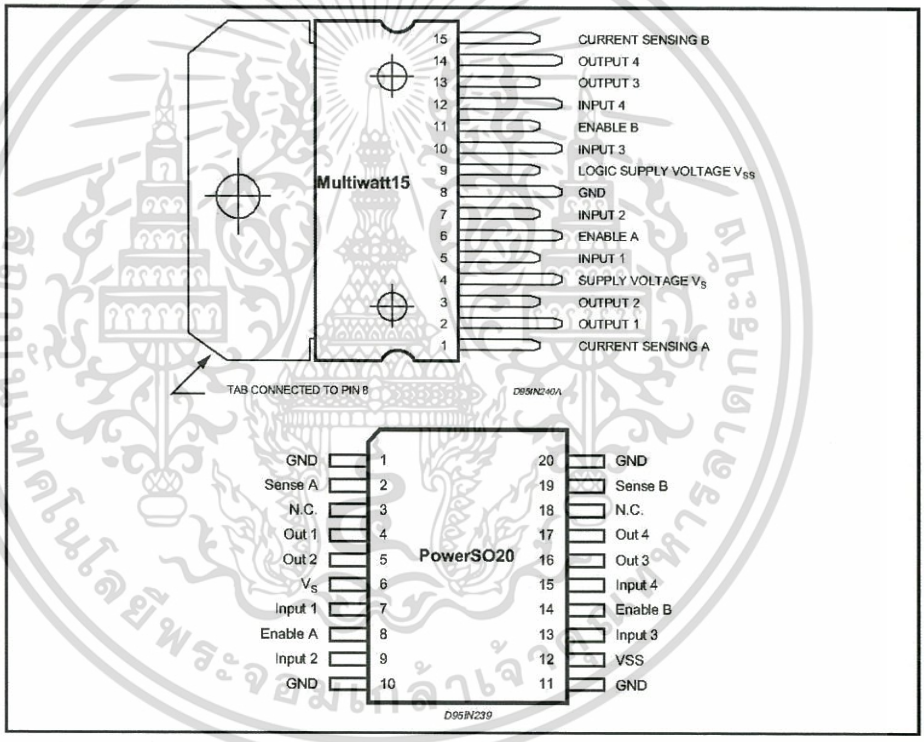
### BLOCK DIAGRAM



## ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Value | Unit |
|---|---|---|---|
| $V_S$ | Power Supply | 50 | V |
| $V_{SS}$ | Logic Supply Voltage | 7 | V |
| $V_I$,$V_{en}$ | Input and Enable Voltage | –0.3 to 7 | V |
| $I_O$ | Peak Output Current (each Channel)<br>– Non Repetitive (t = 100µs)<br>–Repetitive (80% on –20% off; $t_{on}$ = 10ms)<br>–DC Operation | <br>3<br>2.5<br>2 | <br>A<br>A<br>A |
| $V_{sens}$ | Sensing Voltage | –1 to 2.3 | V |
| $P_{tot}$ | Total Power Dissipation ($T_{case}$ = 75°C) | 25 | W |
| $T_{op}$ | Junction Operating Temperature | –25 to 130 | °C |
| $T_{stg}$, $T_j$ | Storage and Junction Temperature | –40 to 150 | °C |

## PIN CONNECTIONS (top view)




## THERMAL DATA

| Symbol | Parameter | | PowerSO20 | Multiwatt15 | Unit |
|---|---|---|---|---|---|
| $R_{th\ j\text{-}case}$ | Thermal Resistance Junction-case | Max. | – | 3 | °C/W |
| $R_{th\ j\text{-}amb}$ | Thermal Resistance Junction-ambient | Max. | 13 (*) | 35 | °C/W |

(*) Mounted on aluminum substrate

**PIN FUNCTIONS** (refer to the block diagram)

| MW.15 | PowerSO | Name | Function |
|---|---|---|---|
| 1;15 | 2;19 | Sense A; Sense B | Between this pin and ground is connected the sense resistor to control the current of the load. |
| 2;3 | 4;5 | Out 1; Out 2 | Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1. |
| 4 | 6 | $V_S$ | Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground. |
| 5;7 | 7;9 | Input 1; Input 2 | TTL Compatible Inputs of the Bridge A. |
| 6;11 | 8;14 | Enable A; Enable B | TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B). |
| 8 | 1,10,11,20 | GND | Ground. |
| 9 | 12 | VSS | Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground. |
| 10; 12 | 13;15 | Input 3; Input 4 | TTL Compatible Inputs of the Bridge B. |
| 13; 14 | 16;17 | Out 3; Out 4 | Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15. |
| – | 3;18 | N.C. | Not Connected |

**ELECTRICAL CHARACTERISTICS** ($V_S$ = 42V; $V_{SS}$ = 5V, $T_j$ = 25°C; unless otherwise specified)

| Symbol | Parameter | Test Conditions | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $V_S$ | Supply Voltage (pin 4) | Operative Condition | | $V_{IH}$ +2.5 | | 46 | V |
| $V_{SS}$ | Logic Supply Voltage (pin 9) | | | 4.5 | 5 | 7 | V |
| $I_S$ | Quiescent Supply Current (pin 4) | $V_{en}$ = H; $I_L$ = 0 | $V_i$ = L<br>$V_i$ = H | | 13<br>50 | 22<br>70 | mA<br>mA |
| | | $V_{en}$ = L | $V_i$ = X | | | 4 | mA |
| $I_{SS}$ | Quiescent Current from $V_{SS}$ (pin 9) | $V_{en}$ = H; $I_L$ = 0 | $V_i$ = L<br>$V_i$ = H | | 24<br>7 | 36<br>12 | mA<br>mA |
| | | $V_{en}$ = L | $V_i$ = X | | | 6 | mA |
| $V_{iL}$ | Input Low Voltage (pins 5, 7, 10, 12) | | | -0,3 | | 1.5 | V |
| $V_{iH}$ | Input High Voltage (pins 5, 7, 10, 12) | | | 2.3 | | VSS | V |
| $I_{iL}$ | Low Voltage Input Current (pins 5, 7, 10, 12) | $V_i$ = L | | | | -10 | µA |
| $I_{iH}$ | High Voltage Input Current (pins 5, 7, 10, 12) | $V_i$ = H ≤ $V_{SS}$ -0.6V | | | 30 | 100 | µA |
| $V_{en}$ = L | Enable Low Voltage (pins 6, 11) | | | -0.3 | | 1.5 | V |
| $V_{en}$ = H | Enable High Voltage (pins 6, 11) | | | 2.3 | | $V_{SS}$ | V |
| $I_{en}$ = L | Low Voltage Enable Current (pins 6, 11) | $V_{en}$ = L | | | | -10 | µA |
| $I_{en}$ = H | High Voltage Enable Current (pins 6, 11) | $V_{en}$ = H ≤ $V_{SS}$ -0.6V | | | 30 | 100 | µA |
| $V_{CEsat\ (H)}$ | Source Saturation Voltage | $I_L$ = 1A<br>$I_L$ = 2A | | 0.95 | 1.35<br>2 | 1.7<br>2.7 | V<br>V |
| $V_{CEsat\ (L)}$ | Sink Saturation Voltage | $I_L$ = 1A (5)<br>$I_L$ = 2A (5) | | 0.85 | 1.2<br>1.7 | 1.6<br>2.3 | V<br>V |
| $V_{CEsat}$ | Total Drop | $I_L$ = 1A (5)<br>$I_L$ = 2A (5) | | 1.80 | | 3.2<br>4.9 | V<br>V |
| $V_{sens}$ | Sensing Voltage (pins 1, 15) | | | -1 (1) | | 2 | V |

# CL-RS445

**OUTPUT : 3.0W～55W (APPROX)** Carbon-brush motors 碳刷电机

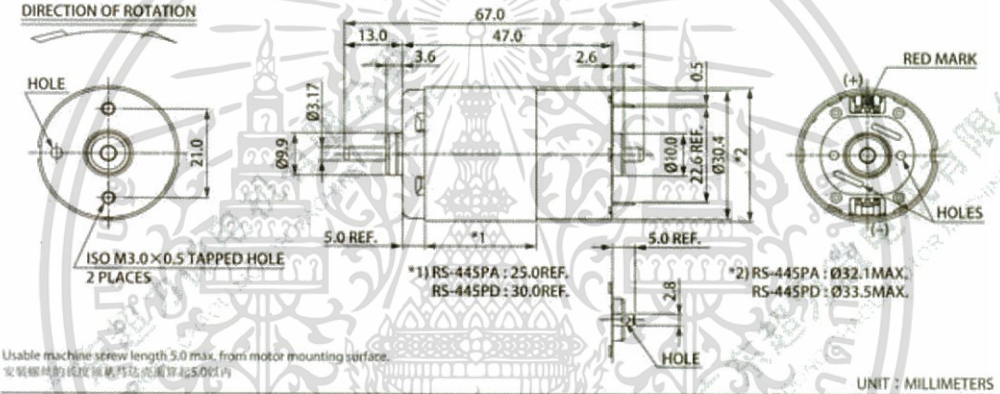
Typical Applications **Precision Instruments :** Printer

主要用途 **精密机器 :** 打印机

WEIGHT : 127g (APPROX)

| MODEL | VOLTAGE | | NO LOAD | | AT MAXIMUM EFFICIENCY | | | | | STALL | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OPERATING RANGE | NOMINAL | SPEED | CURRENT | SPEED | CURRENT | TORQUE | | OUTPUT | TORQUE | | CURRENT |
| | | | r/min | A | r/min | A | mN・m | g・cm | W | mN・m | g・cm | A |
| CL-RS445PA-14233 | 12～42 | (!) 42V CONSTANT | 6500 | 0.060 | 5410 | 0.30 | 13.7 | 140 | 7.78 | 81.8 | 834 | 1.47 |
| CL-RS445PA-15200 | 12～42 | (!) 42V CONSTANT | 7600 | 0.067 | 6420 | 0.36 | 14.6 | 148 | 9.77 | 93.7 | 955 | 1.98 |
| CL-RS445PD-16175 | 12～42 | (!) 42V CONSTANT | 7400 | 0.055 | 6410 | 0.35 | 15.3 | 156 | 10.3 | 114 | 1162 | 2.28 |

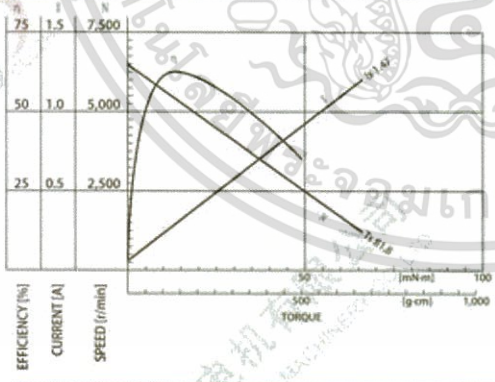
(!) This operating voltage indicates the average or r.m.s. in case of pulse-width modulation (PWM) power supply.
Besides, as withstand voltage, 1800V for one second can not be guaranteed.
(!) 这个标示使用电压是PWM执行控制时的有效电压。另外，我们无法提供耐电压1800V 1sec的保证。敬请见谅。



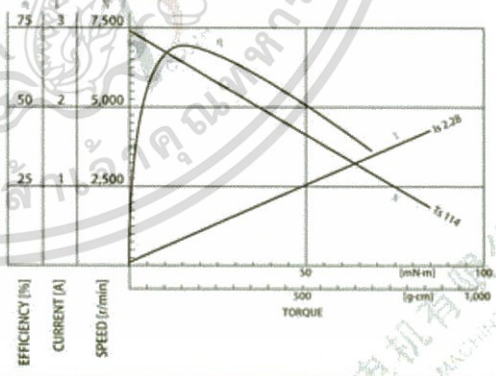

Usable machine screw length 5.0 max. from motor mounting surface.
安装螺丝的长度最大允许距离算起5.0以内

UNIT : MILLIMETERS

**CL-RS445PA-14233** 42.0V

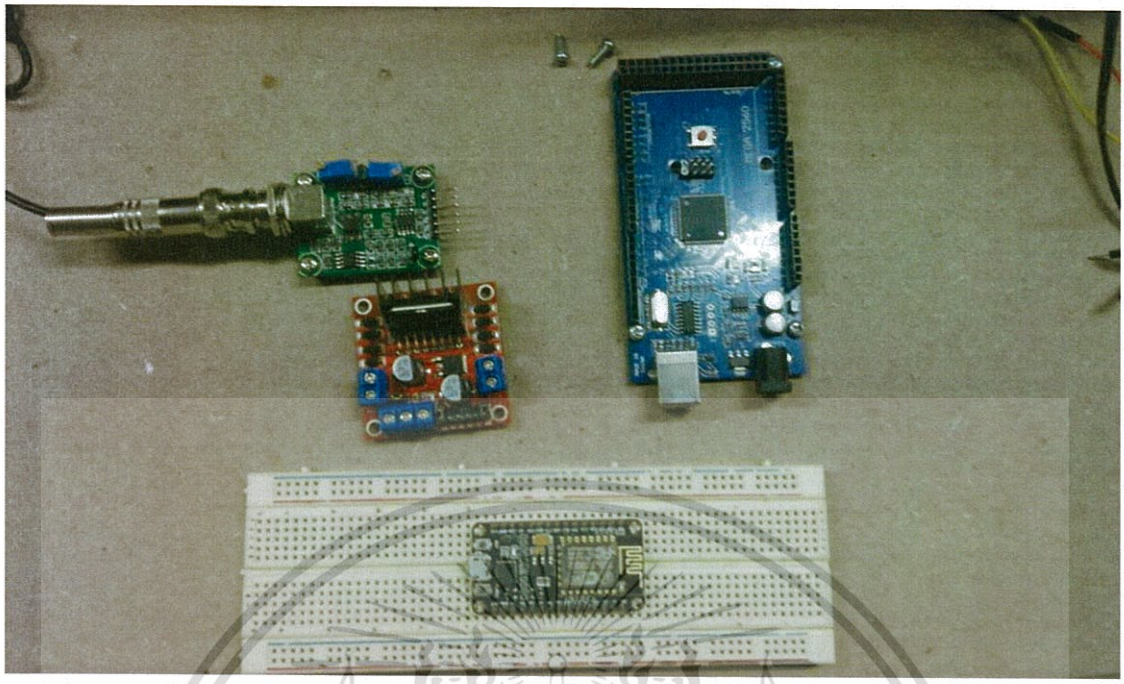



**CL-RS445PD-16175** 42.0V

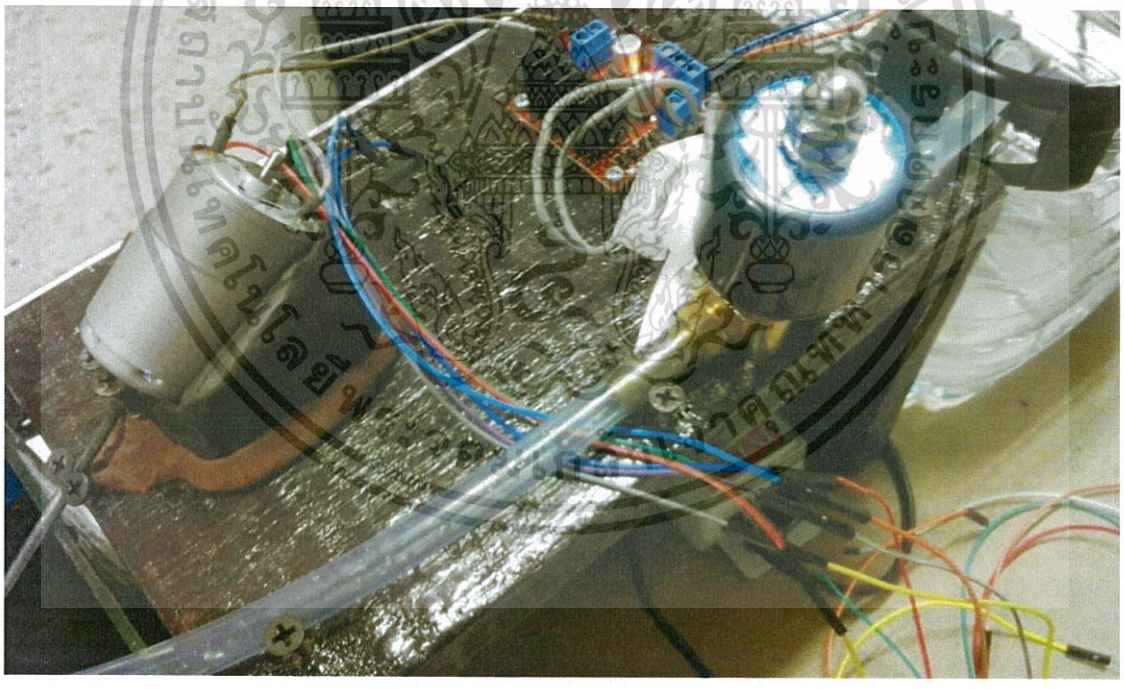



Guangdong Chaoli motors Co.,Ltd

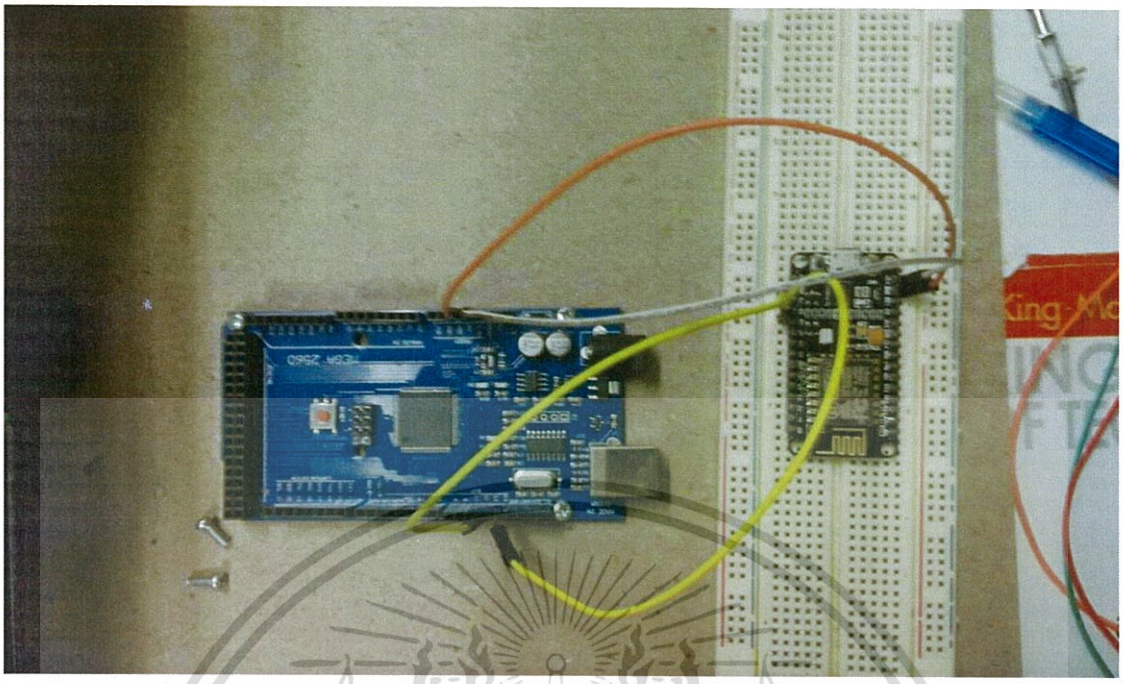
# ภาคผนวก จ
# วิธีการต่ออุปกรณ์



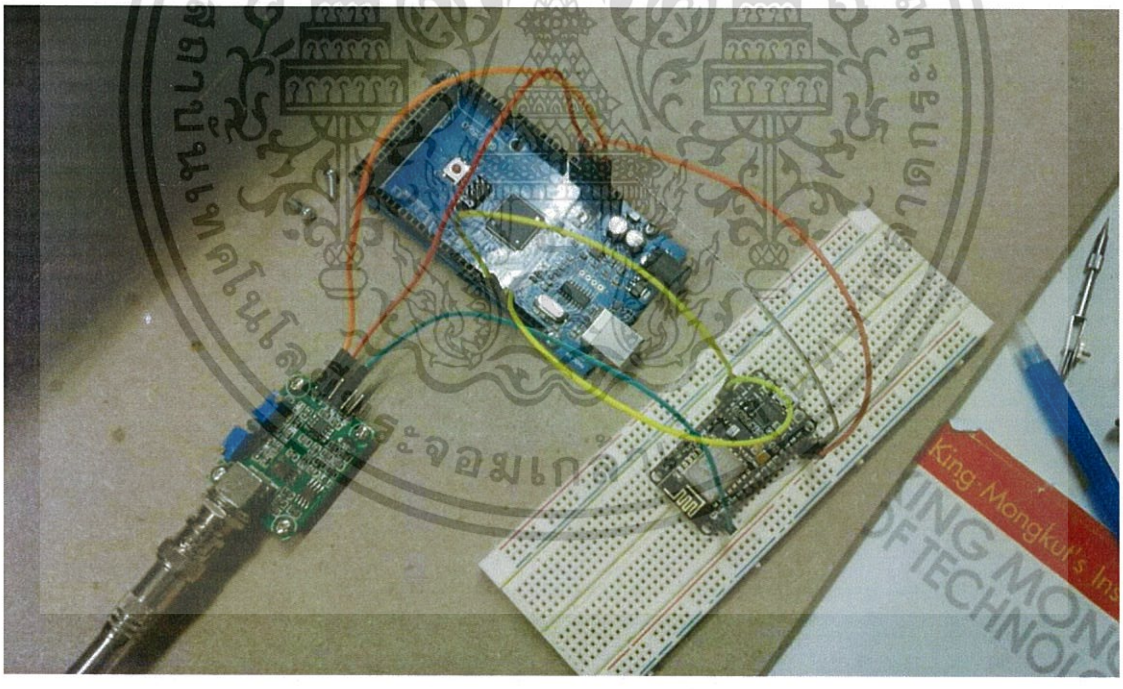
**รูปที่ จ.1** โมดูล pH เซนเซอร์, ชุดขับมอเตอร์, Arduino Mega 2560 และ NodeMCU v.1.0



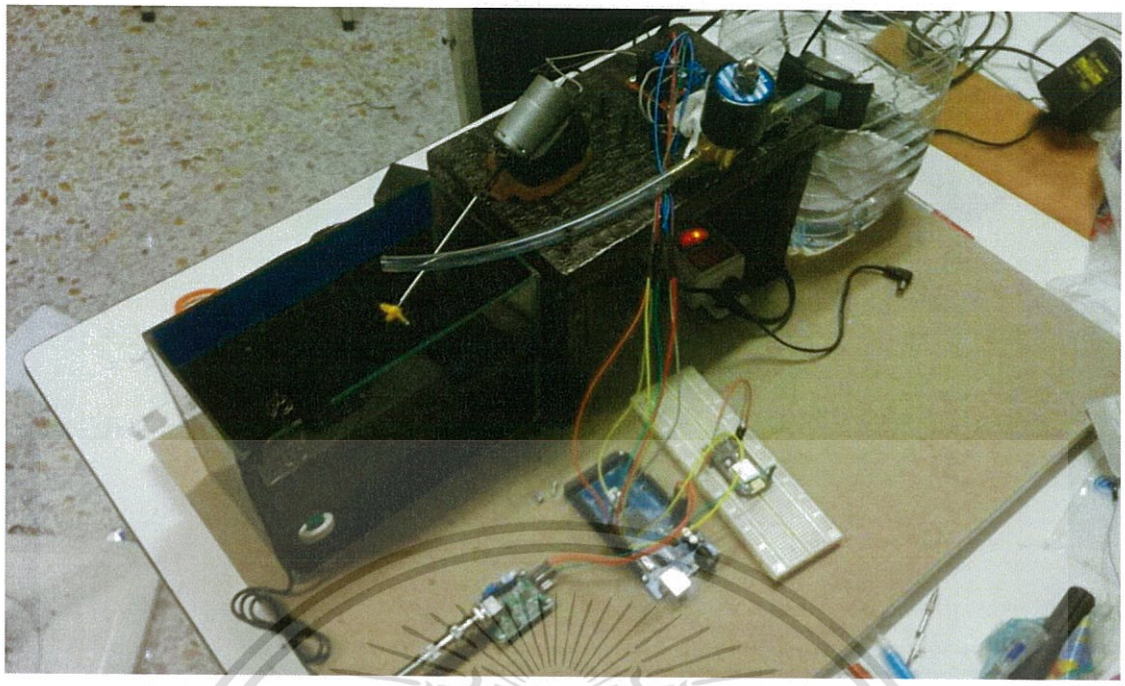
**รูปที่ จ.2** มอเตอร์และวาล์วโซลินอยด์



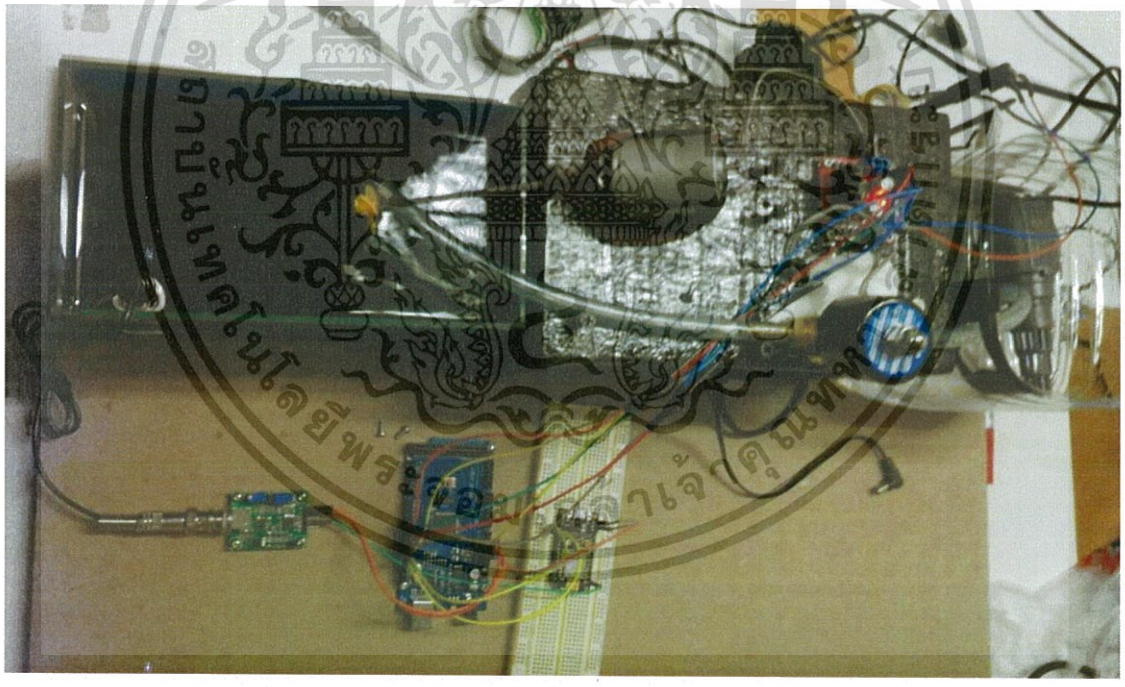
**รูปที่ จ.3** วิธีการต่อ Arduino Mega 2560 เข้ากับ NodeMCU v.1.0



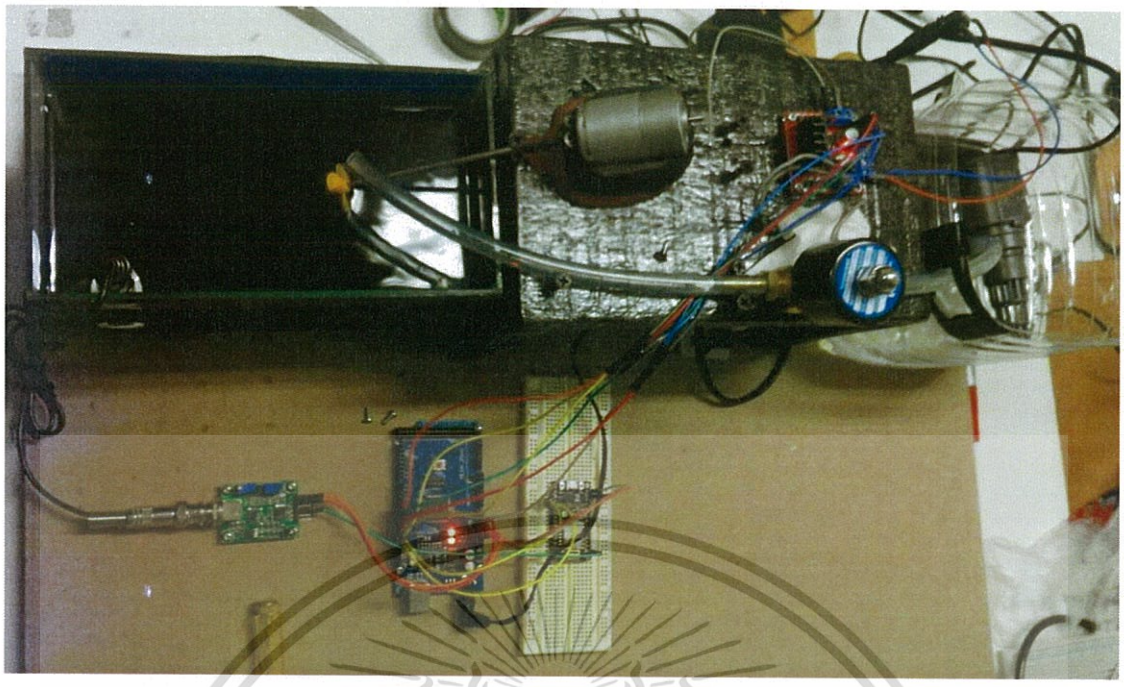
**รูปที่ จ.4** วิธีการต่อโมดูล pH เซนเซอร์เข้ากับ Arduino Mega 2560 และ NodeMCU v.1.0



**รูปที่ จ.5** วิธีการต่อชุดขับมอเตอร์เข้ากับ Arduino Mega 2560



**รูปที่ จ.6** ต่อไฟจากอะแดปเตอร์ 12 V เข้ากับตัวชุดขับมอเตอร์เพื่อให้ทำงาน



**รูปที่ จ.7** ต่อไฟจากอะแดปเตอร์ 9 V เข้ากับตัว Arduino Mega 2560 เพื่อให้ทำงาน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง