

ระบบควบคุมระดับน้ำแบบไร้สาย : ส่วนอุปกรณ์วัดอัตราการไหล

WIRELESS LEVEL CONTROL SYSTEM :

FLOW TRANSMITTER PART



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมระดับน้ำแบบไร้สาย : ส่วนอุปกรณ์วัดอัตราการไหล

WIRELESS LEVEL CONTROL SYSTEM :
FLOW TRANSMITTER PART



T144406



สาขา.....
เลขทะเบียน 144406
วันเดือนปี 24 พ.ย. 2559

b. 12818896
f.

ปฏิญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS LEVEL CONTROL SYSTEM : FLOW TRANSMITTER PART



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมระดับน้ำแบบไร้สาย : ส่วนอุปกรณ์วัดอัตราการไหล
WIRELESS LEVEL CONTROL SYSTEM : FLOW TRANSMITTER PART

ผู้จัดทำ นางสาวพรรณทิพย์ ปัญญาสิทธิ์ 54010878
นางสาวปานรวี พรธิสาร 55010761
นางสาวภัทราวดี ศรีทัศน์ 55010928



.....อาจารย์ที่ปรึกษา
(ศาสตราจารย์ ดร.วันชัย ธีรสุวภา)

.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์เทพจิตร เชยโกคา)

.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า)

.....อาจารย์ที่ปรึกษา
(ดร.สิริชัย ธรรมารักษ์วัฒน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมระดับน้ำแบบไร้สาย : ส่วนอุปกรณ์วัดอัตราการไหล

โดย

นางสาวพรรณทิพย์ ปัญญาสิทธิ์	54010878
นางสาวปานรวี พรธินสาร	55010761
นางสาวภัทราวดี ศรีทัศน์	55010928

อาจารย์ที่ปรึกษา

ศาสตราจารย์ ดร.วันชัย รั้วรุจา

ผู้ช่วยศาสตราจารย์เทพจิตรี เขยโกคา

ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรเมณีล้ำค่า

ดร.สิริชัย ธรรมารักษ์วัฒน์

ปีการศึกษา 2558

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ เป็นการนำเสนอการออกแบบระบบควบคุมน้ำแบบไร้สายโดยโครงงานนี้ประกอบด้วยทั้งหมด 5 ส่วน ดังนี้ ส่วนของโครงสร้าง ส่วนของซอฟต์แวร์ที่ควบคุม PLC ส่วนของ Actuator ส่วนของ Level Transmitter และส่วนของ Flow Transmitter ซึ่งในปฏิญานิพนธ์ฉบับนี้จะนำเสนอและอธิบายในส่วนของ Flow Transmitter เท่านั้น จุดประสงค์ของการจัดทำโครงงานคือ เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์และการรับส่งข้อมูลแบบไร้สายระหว่าง Sever (PLC) กับ Client ผ่านทาง Wi-Fi Module ESP8266 เพื่อใช้แสดงค่าอัตราการไหลที่ได้จากการวัดอัตราการไหลของน้ำผ่าน Flow Sensor โดยให้ค่าที่ได้จากการทดลองให้แสดงค่าแบบ Real Time โดยทำการเขียนโค้ดคำสั่งในโปรแกรม Arduino ซึ่งแบ่งโค้ดเป็น 2 ส่วนคือ ส่วนส่งข้อมูล โดยจะอ่านค่าข้อมูลจาก Flow Sensor แล้วส่งข้อมูลผ่าน Module (Wi-Fi ESP8266) ไปให้ Server ซึ่งต่อโดยตรงกับ PLC ในด้านของส่วนรับข้อมูล โดยจะรับข้อมูลที่ส่งมาจาก Module Wi-Fi (Client) แล้วส่งข้อมูลไปให้ PLC นอกจากนี้ยังมีการเขียน Library ย่อยๆ ของ Arduino เพื่อใช้เชื่อมต่อ Wi-Fi กับ Wi-Fi ผ่านเราเตอร์ หลังจากนั้นก็นำไปทำการทดลองจริงร่วมกับส่วนอื่นๆ อีกทั้ง 4 ส่วน เพื่อดูการทำงานจริงของการควบคุมระดับน้ำของระบบที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS LEVEL CONTROL SYSTEM :

FLOW TRANSMITTER PART

By

Ms.Pantip	Panyasit	54010878
Ms.Parnravee	Pornthisarn	55010761
Ms.Phattrawadee	Sritas	55010928

Advisors

Prof. Dr. Vanchai	Riewruja
Asst. Prof. Thepjit	Cheypoca
Asst. Prof. Dr. Wandee	Petchmaneelumka
Dr. Sirichai	Thammarakwattana

Academic Year 2015

ABSTRACT

This thesis presents the design of wireless level control system that consists of five parts including hardware, software for control PLC, actuator, level transmitter and flow transmitter parts. However, this thesis will present and describe only the part of flow transmitter that our group is responsibility. The objective of this project is to study of function of microcontroller and transfer data wirelessly between Sever (PLC) to Client via Wi-Fi Module ESP8266. Measured flow rate from flow transmitter is displayed with real time. Arduino program is used to code the command that consists of two parts including sending and receiving data parts. Sending data part will get data from flow sensor and send to server connected directly to PLC via Wi-Fi mode ESP8266. Receiving data part will receive data from client via Wi-Fi mode ESP8266 and send it to PLC. In addition, the minor code is created in library of Arduino program to connect between 2 Wi-Fi Modules through the router. After that, all part of system is installed and tested the real wireless level control system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปฏิญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปด้วยดี กลุ่มผู้วิจัยขอกราบ
ขอบพระคุณบุคคลทุกคนที่มีส่วนเกี่ยวข้องส่งผลให้ปฏิญานิพนธ์ฉบับนี้เสร็จสมบูรณ์

ศ.ดร.วันชัย ธีรจุฑา กลุ่มผู้วิจัยขอกราบขอบพระคุณอย่างสูง สำหรับคำแนะนำและข้อคิดเห็น
ต่างๆ อันเป็นประโยชน์อย่างยิ่งในการทำวิจัยและความช่วยเหลือความเอาใจใส่ในทุกๆ ด้านจน
ปฏิญานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดี ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและขอขอบคุณอาจารย์เป็น
อย่างสูง

ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรธณีสัจจา กลุ่มผู้วิจัยขอกราบขอบพระคุณอย่างสูง
สำหรับความเอาใจใส่ในทุกๆ ด้านตลอดระยะเวลาที่ผ่านมา ที่ให้คำปรึกษา ข้อชี้แนะ ให้คำแนะนำ
ตลอดจนแก้ไขข้อบกพร่องในการทำปฏิญานิพนธ์ และคอยให้ความช่วยเหลือในหลายสิ่งหลายอย่าง
จนกระทั่งปฏิญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและ
ขอขอบคุณอาจารย์เป็นอย่างสูง

ผู้ช่วยศาสตราจารย์เทพจิตร เขยโสภา อาจารย์ที่ปรึกษาปฏิญานิพนธ์ กลุ่มผู้วิจัยขอกราบ
ขอบพระคุณอย่างสูงในความอนุเคราะห์จากอาจารย์ สำหรับการให้โอกาสในการศึกษาปฏิญานิพนธ์
ฉบับนี้ รวมทั้งความรู้ คำแนะนำต่างๆ จนปฏิญานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดี

ขอขอบพระคุณศาสตราจารย์ ดร.สิริชัย ธรรมารักษ์วัฒน์ ที่ให้คำแนะนำ และคอยดูแลมา
โดยตลอด

ขอบคุณครอบครัว ที่คอยช่วยเหลือห่วงใย คอยสนับสนุนเป็นแรงผลักดันที่ดีเสมอมา ทั้งด้าน
การเงิน คอยให้กำลังใจเอาใจใส่ สนับสนุนการศึกษาเพื่อรอความสำเร็จของผู้วิจัย จนสามารถฝ่าฟัน
อุปสรรคมาได้ ทำให้ปฏิญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอบคุณเพื่อนกลุ่มโครงการ Wireless Level Control System ทุกคนที่คอยช่วยเหลือ
เกื้อกูลซึ่งกันและกัน และคอยเป็นกำลังใจที่ดี ช่วยกันแก้ปัญหาพร้อมทุกข์ร่วมสุขด้วยกันเสมอมา จนทำ
ให้ปฏิญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้

ผู้จัดทำ

นางสาวพรรณทิพย์ ปัญญาสิทธิ์

นางสาวปานรวี พรธินสาร

นางสาวภัทราวดี ศรีทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการศึกษาและจัดทำโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและข้อมูลที่เกี่ยวข้อง	3
2.1 เครือข่ายไร้สาย	3
2.1.1 ความหมายของ Wi-Fi	3
2.1.2 มาตรฐานการทำงานของระบบเครือข่ายไร้สาย	3
2.2 โพรโทคอล TCP/IP	4
2.3 Wi-Fi Module ESP8266	5
2.4 Arduino Microcontroller	7
2.4.1 Arduino Mega 2560	7
2.4.2 รูปแบบการเขียนโปรแกรมบน Arduino	7
2.5 เครื่องมือวัดอัตราการไหล	8
2.6 สมการต่อเนื่อง (Continuity Equation)	9
2.7 คุณสมบัติของรูปคลื่นพัลส์	10
2.8 วงจรบัฟเฟอร์	12
2.9 สัญญาณ PWM	12
บทที่ 3 ขั้นตอนการดำเนินงาน	14
3.1 โครงสร้างของระบบ	14
3.2 การเลือกใช้อุปกรณ์	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2.1 เซนเซอร์วัดอัตราการไหลของน้ำ (Flow Sensor)	15
3.2.2 Regulator	15
3.2.3 Arduino MEGA 2560 R3	16
3.2.4 โมดูล Wi-Fi ESP8266	17
3.2.5 Adaptor	18
3.2.6 Power Supply	18
3.3 การทดลองหาค่าความคลาดเคลื่อนระหว่างค่าอัตราการไหล ที่ได้จากการเขียนโปรแกรมและการวัดอัตราการไหลจริง	19
3.3.1 อุปกรณ์การทดลอง	20
3.3.2 ขั้นตอนการทดลอง	20
3.4 การทดสอบ Wi-Fi Module ESP-8266	23
3.4.1 อุปกรณ์การทดลอง	23
3.4.2 ขั้นตอนการทดลอง	23
3.5 ทดสอบหาค่าอัตราการไหลของน้ำเปรียบเทียบกับเปอร์เซ็นต์การทำงานของปั๊ม	27
3.6 การเขียน Library ที่ใช้สำหรับ Wi-Fi Module ESP8266-01	29
3.7 การทดลองเขียนโปรแกรมส่งค่าผ่าน Wi-Fi Module ESP-8266	31
3.8 การทดลองเขียนโปรแกรมรับค่าจาก Wi-Fi Module ESP-8266	33
บทที่ 4 ผลการทดลอง	37
4.1 การทดลอง Flow Sensor เพื่อหาค่า Flow Rate ของระบบ	37
4.1.1 ผลการทดลอง	37
4.2 การทดลองเชื่อมต่อโมดูล Wi-Fi กับโปรแกรม Hercules ในคอมพิวเตอร์	38
4.3 การทดลองหาค่าแรงดันเอาต์พุตของ Flow Sensor เทียบกับเปอร์เซ็นต์ การทำงานของปั๊มน้ำ	40
4.4 ผลการทดลอง ส่ง-รับสัญญาณ (ค่าแรงดัน) เข้า Module Wi-Fi แล้วส่งเข้า PLC และโปรแกรม TIA PORTAL	42
บทที่ 5 ผลสรุปและข้อเสนอแนะ	47
5.1 สรุป	47
5.2 ปัญหาและอุปสรรคในการดำเนินโครงการนี้	48
5.3 ข้อเสนอแนะ	49
เอกสารอ้างอิง	50
ภาคผนวก ก Library สำหรับใช้งานฟังก์ชัน	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

ภาคผนวก ข โปรแกรมคำสั่งที่ใช้ในการทดสอบโปรแกรมทดสอบ
ภาคผนวก ค पोสเตอร์

หน้า
57
63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ESP8266 รุ่นที่นิยมใช้งานคือ ESP-01	6
2.2 ขาของ ESP8266 รุ่น ESP-01	6
2.3 บอร์ด Arduino Mega 2560	7
2.4 การเขียนโปรแกรมลงในบอร์ด Arduino Mega 2560	8
2.5 โครงสร้างของเครื่องมือวัดอัตราการไหลแบบเทอร์ไบน์	9
2.6 Hall Effect Sensor	9
2.7 คลื่นพัลส์ในอุดมคติ	10
2.8 คลื่นพัลส์ในทางปฏิบัติ	11
2.9 วงจรบัฟเฟอร์แบบใช้โอปแอมป์	12
2.10 กราฟการแสดงค่า PWM ในรูปของค่าดิวิตีไซเคิล	13
3.1 โครงสร้างระบบควบคุมระดับน้ำแบบไร้สายที่ออกแบบ	14
3.2 Flow Sensor	15
3.3 Regulator	16
3.4 บอร์ด Arduino MEGA 2560 R3	16
3.5 โมดูล Wi-Fi ESP8266 ESP-01	17
3.6 AC/DC Adaptor	18
3.7 Power Supply	19
3.8 การเชื่อมต่ออุปกรณ์เพื่อทดลอง	19
3.9 การต่ออุปกรณ์	20
3.10 วงจรทดสอบ Flow Sensor	21
3.11 วงจรทดสอบ Wi-Fi ESP 8266	23
3.12 การ Upload ไฟกระพริบ	24
3.13 โปรแกรม Hercules	25
3.14 การส่งข้อมูลระหว่าง Client และ Server	26
3.15 วงจร Speed Motor และ Flow Sensor	27
3.16 การใช้คำสั่ง map ค่าจากเปอร์เซ็นต์ให้เป็นแรงดัน	28
3.17 การกำหนดค่าเปอร์เซ็นต์การทำงานของปั๊ม	28
3.18 หน้าต่าง ESP_8266 Library ในส่วนของ AT Command	30
3.19 หน้าต่าง ESP_8266 Library ในส่วนของ Header File	30

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20 หน้าต่าง ESP_8266 Library ในส่วนของ Source File	31
3.21 การกำหนดรูปแบบการทำงานของ Wi-Fi Module ESP 8266 ให้เป็น Client	32
3.22 ส่วนการนับรอบการหมุนใบพัดของ Flow Sensor และส่วนแสดงค่า Flow Rate	32
3.23 ส่วนส่งค่าแรงดันของ Flow Sensor	33
3.24 การต่อวงจร Wi-Fi กับวงจร Buffer เข้า PLC	34
3.25 กำหนดรูปแบบการทำงานของ Wi-Fi Module ESP 8266 ให้เป็น Server	35
3.26 กำหนดรูปแบบการทำงานของ Wi-Fi Module ESP 8266 ให้เป็น Server	35
3.27 ส่วนส่งค่าแรงดัน โดยกำหนดให้ขา PWM ที่บอร์ด Arduino เป็นขาเอาต์พุต	36
4.1 การเชื่อมต่ออุปกรณ์เพื่อทดลอง	37
4.2 ผลจากหน้าจอ Serial Monitor และโปรแกรม Hercules	39
4.3 ระบบรวมที่ใช้ในการทดลอง	40
4.4 ส่วนของระบบดังนี้	40
4.5 ส่วนของการต่อวงจร	41
4.6 หน้าจอโปรแกรม Arduino และผลที่แสดงผ่านหน้า Serial Monitor	41
4.7 การส่งสัญญาณจาก Flow Sensor ไป PLC	42
4.8 ระบบด้านตัวส่งสัญญาณ	43
4.9 ระบบด้านตัวรับสัญญาณ	43
4.10 หน้าจอโปรแกรม Arduino และผลที่แสดงผ่านหน้า Serial Monitor (ด้านตัวส่งสัญญาณ)	44
4.11 หน้าจอโปรแกรม Arduino และผลที่แสดงผ่านหน้า Serial Monitor (ด้านตัวรับสัญญาณ)	44
4.12 หน้าจอ HMI ของ PLC	45
4.13 หน้าจอโปรแกรม TIA Portal	46

สารบัญตาราง

ตารางที่	หน้า
4.1 ค่า Flow Rate จากการวัดและจากหน้า Serial Monitor โดยจับเวลา 6 วินาที	37
4.2 ค่า Flow Rate จากการวัดและจากหน้า Serial Monitor โดยจับเวลา 1 นาที	38
4.3 ผลการทดลองหาค่าแรงดันเอาต์พุตของ Flow Sensor เทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ	42
4.4 ผลการทดลองหาค่าแรงดันที่ Module Wi-Fi ด้านตัวรับ, ค่าแรงดันที่ขา PWM ของบอร์ด Arduino และค่าแรงดันที่เข้า PLC เทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ	45



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในยุคปัจจุบันเทคโนโลยีการติดต่อสื่อสาร ได้มีการพัฒนาเจริญก้าวหน้าเป็นอย่างมาก การติดต่อสื่อสาร การส่งข้อมูลผ่านระบบเครือข่ายไร้สาย Wi-Fi เป็นที่นิยม ได้รับการยอมรับอย่างแพร่หลาย เนื่องจาก Wi-Fi ทำให้การสื่อสารง่ายดาย สะดวกต่อการส่งผ่านข้อมูล โดยไม่ต้องเคลื่อนย้ายและเปลี่ยนแปลงอุปกรณ์ ช่วยลดค่าใช้จ่ายในการวางระบบเครือข่าย ระบบเครือข่ายไม่ซับซ้อน อุปกรณ์ราคาถูกลง หาซื้อได้ง่าย ทางกลุ่มผู้จัดทำจึงได้เล็งเห็นความสำคัญของการนำเอาระบบการติดต่อเครือข่ายแบบไร้สาย การสื่อสารผ่านทาง Wi-Fi นี้มาปรับปรุงประยุกต์การใช้งานด้านอุตสาหกรรม แทนการใช้สายในการส่งข้อมูลแบบเดิม จะช่วยทำให้งานทางด้านอุตสาหกรรมมีความสะดวกมากขึ้น ง่ายต่อการดูแลรักษา อีกทั้งยังช่วยลดค่าใช้จ่ายในการติดตั้งสายที่ใช้ควบคุมเครื่องจักรต่างๆ ในโรงงานอุตสาหกรรม

ทางกลุ่มผู้จัดทำได้ทำโครงการจำลองระบบควบคุมระดับน้ำแบบไร้สาย (Wireless Level Control System) โดยการนำ Wi-Fi Module ESP8266 มาเป็นตัวกลางในการรับ-ส่งข้อมูลระหว่าง PLC (Programmable Logic Controller) และโปรแกรม TIA PORTAL โดยการทำงานของโปรแกรม TIA PORTAL จะรับค่าจากเซนเซอร์วัดความดัน (Pressure Sensor) และเซนเซอร์วัดอัตราการไหล (Flow Sensor) แล้วนำค่าที่ได้ส่งให้ PLC เพื่อสั่งการควบคุมชุดขับเคลื่อนมอเตอร์ BTS7960 ให้ควบคุมความเร็วรอบของมอเตอร์ปั้มน้ำต่อไป

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เพื่อออกแบบและสร้างแบบจำลองระบบควบคุมระดับน้ำอัตโนมัติแบบไร้สาย
2. เพื่อศึกษาการทำงานและการรับส่งข้อมูลแบบไร้สาย ผ่านทาง Wi-Fi Module ESP8266
3. เพื่อศึกษาวิธีการใช้งานและการเขียนโปรแกรม Arduino
4. เพื่อศึกษาการทำงานของระบบควบคุมระดับน้ำอัตโนมัติด้วยระบบไร้สาย

1.3 ขอบเขตของโครงการ

1. เขียนโปรแกรมเพื่อรับส่งข้อมูลผ่าน Wi-Fi Module
2. สามารถควบคุมมอเตอร์ผ่านระบบ Wi-Fi ได้
3. ออกแบบและสร้างแบบจำลองระบบควบคุมระดับน้ำอัตโนมัติแบบไร้สายใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษาและจัดทำโครงการงาน

1. ศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้องกับการทำงานของระบบควบคุมระดับน้ำอัตโนมัติ ได้แก่ หลักการทำงานของเซนเซอร์วัดระดับ, หลักการทำงานของเซนเซอร์วัดอัตราการไหล, การเขียนโปรแกรม Arduino และการรับส่งข้อมูลผ่านทาง Wi-Fi Module ESP8266
2. วางแผนการทำงาน และแบ่งหน้าที่การทำงาน
3. ทำการทดสอบ Wi-Fi Module ESP8266
4. ศึกษาวิธีการทำงานของเซนเซอร์วัดอัตราการไหลอย่างละเอียด
5. ศึกษาอุปกรณ์ที่ต้องใช้ในการทดลอง
6. ศึกษาวงจร และการเขียนโค้ดที่ใช้ในการทดลองวัดอัตราการไหลของน้ำ
7. ทดลองวัดอัตราการไหลของน้ำ และบันทึกค่าผลการทดลอง
8. สรุปผลการทำงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถควบคุมระบบควบคุมระดับน้ำแบบไร้สายได้
2. สามารถนำความรู้จากการออกแบบจำลองระบบควบคุมระดับน้ำอัตโนมัติแบบไร้สาย ไปประยุกต์ใช้ในงานอุตสาหกรรมต่อไป
3. สามารถเขียนคำสั่งควบคุมระบบผ่าน Wi-Fi Module ได้
4. สามารถเขียนโปรแกรมควบคุมการทำงานของ PLC ได้

1.6 รายละเอียดของปฏิญญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บทกับอีก 3 ภาคผนวก ประกอบด้วยรายละเอียดดังนี้

บทที่ 1 บทนำ เป็นการกล่าวถึงความเป็นมาและความสำคัญของโครงการ วัตถุประสงค์ในการทำปฏิญญานิพนธ์ ขอบเขตของโครงการ ขั้นตอนการศึกษาและจัดทำโครงการ ประโยชน์ที่คาดว่าจะได้รับและรายละเอียดของปฏิญญานิพนธ์

บทที่ 2 ทฤษฎี หลักการ และความรู้ที่เกี่ยวข้องในส่วนของเซนเซอร์วัดอัตราการไหล

บทที่ 3 ขั้นตอนการดำเนินงาน

บทที่ 4 ผลการทดลอง

บทที่ 5 บทวิจารณ์และสรุป สรุปการดำเนินการ ปัญหา และแนวทางการปรับปรุงโครงการ

ภาคผนวก ก Library สำหรับใช้งานฟังก์ชัน

ภาคผนวก ข โปรแกรมคำสั่งที่ใช้ในการทดสอบโปรแกรมทดสอบการทำงานของเซนเซอร์

ภาคผนวก ค โพสต์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและข้อมูลที่เกี่ยวข้อง

2.1 เครือข่ายไร้สาย

2.1.1 ความหมายของ Wi-Fi

Wi-Fi หรือ Wireless หมายถึง เครือข่ายไร้สาย มักใช้กับระบบเครือข่าย ไม่ว่าจะป็นในองค์กรหรือในระบบเครือข่ายอินเทอร์เน็ต

ระบบเครือข่ายไร้สาย (Wireless LAN) หมายถึงเทคโนโลยีที่ช่วยให้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง หรือกลุ่มของเครื่องคอมพิวเตอร์สามารถสื่อสารกันได้ รวมถึงการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่ายคอมพิวเตอร์ด้วยเช่นกัน โดยปราศจากการใช้สายสัญญาณในการเชื่อมต่อ แต่จะใช้คลื่นวิทยุเป็นช่องทางการสื่อสารแทน การรับส่งข้อมูลระหว่างกันจะผ่านอากาศ ทำให้ไม่ต้องเดินสายสัญญาณ และติดตั้งใช้งานได้สะดวกขึ้น

2.1.2 มาตรฐานการทำงานของระบบเครือข่ายไร้สาย

IEEE 802.11 คือ มาตรฐานการทำงานของระบบเครือข่ายไร้สายที่กำหนดขึ้นโดย Institute of Electrical and Electronics Engineers (IEEE) เป็นมาตรฐานกลาง ที่ได้นำมาปฏิบัติใช้ เพื่อที่จะทำการเชื่อมโยงอุปกรณ์เครือข่ายไร้สายเข้าด้วยกันบนระบบ ในทางปกติแล้วการเชื่อมต่อระบบเครือข่ายไร้สาย จำเป็นต้องใช้อุปกรณ์สองชิ้น นั่นคือ

1. แอคเซสพอยต์ (Access Point) คือ ตัวกลางที่ช่วยในการติดต่อระหว่าง ตัวรับส่งสัญญาณ Wireless ของผู้ใช้กับเราเตอร์ผ่านทางสายนำสัญญาณที่ทำจากทองแดงที่ได้รับการเชื่อมต่อกับระบบเครือข่าย เช่น สายแลนหรือสายโทรศัพท์ ADSL หรือผ่านทางสายใยแก้วนำแสง
2. ตัวรับส่งสัญญาณ Wireless ทำหน้าที่รับ-ส่งสัญญาณ ระหว่างตัวรับส่งแต่ละตัวด้วยกัน หรือระหว่างตัวลูกข่ายกับแอคเซสพอยต์

มาตรฐานที่อยู่ภายใต้กรอบของเทคโนโลยี IEEE 802.11

1. IEEE 802.11a : เสร็จสมบูรณ์เมื่อปี ค.ศ. 1999 โดยออกเผยแพร่ช้ากว่าของมาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุงความเร็วในการส่งข้อมูลให้วิ่งได้สูงถึง 54 Mbps บนความถี่ 5 GHz ที่ความเร็วนี้สามารถทำการแพร่ภาพและข่าวสารที่ต้องการความละเอียดสูงได้ อัตราความเร็วในการรับส่งข้อมูลสามารถปรับระดับให้ช้าลงได้ เพื่อเพิ่มระยะทางการเชื่อมต่อให้มากขึ้น แต่ทว่าข้อเสียก็คือ ความถี่ 5 GHz นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลายๆ ประเทศไม่อนุญาตให้ใช้ เพราะได้จัดสรรให้อุปกรณ์ประเภทอื่นไปแล้ว และเนื่องด้วยการที่มาตรฐานนี้ ใช้การเชื่อมต่อที่ความถี่สูงๆ ทำให้มาตรฐานนี้ มีระยะการรับส่งที่ค่อนข้างใกล้ คือ ประมาณ 35 เมตร ในโครงสร้างปิด (เช่น ในตึก ในอาคาร) และ 120 เมตรในที่โล่งแจ้ง และด้วยความที่ส่งข้อมูลด้วยความถี่สูงนี้ ทำให้การส่งข้อมูลนั้นไม่สามารถทะลุทะลวงโครงสร้างของตึกได้มากนัก จึงไม่ค่อยเป็นที่นิยมมากนัก

2. IEEE 802.11b : เสร็จสมบูรณ์เมื่อปี ค.ศ. 1999 โดยได้ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ผสมกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุ ความถี่ 2.4 GHz เนื่องจากการใช้คลื่นความถี่ที่ต่ำกว่าอุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11a ทำให้อุปกรณ์ที่ใช้มาตรฐานนี้จะมีความสามารถในการส่งคลื่นสัญญาณไปได้ไกลกว่าคือประมาณ 38 เมตรในโครงสร้างปิดและ 140 เมตรในที่โล่งแจ้ง รวมถึงสัญญาณสามารถทะลุทะลวงโครงสร้างตึกได้มากกว่าอุปกรณ์ที่รองรับกับมาตรฐาน IEEE 802.11a ด้วย

3. IEEE 802.11g : เสร็จสมบูรณ์ในปี ค.ศ. 2003 ทางคณะกรรมการ IEEE 802.11g ได้นำเอาเทคโนโลยี OFDM ของ 802.11a มาพัฒนาบนความถี่ 2.4 GHz จึงทำให้ใช้ความเร็ว 36-54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b ซึ่ง 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน มาตรฐานนี้เป็นที่ยอมรับจากผู้ใช้เป็นจำนวนมากและกำลังจะเข้ามาแทนที่ 802.11b ในอนาคตอันใกล้

4. IEEE 802.11n : เสร็จสมบูรณ์ในปี ค.ศ. 2009 ทำงานบนย่านความถี่ 2.4 และ 5 GHz โดยที่สามารถให้อัตราการส่งถ่ายข้อมูลสูงสุดถึง 300 Mbps มีความสามารถในการส่งคลื่นสัญญาณได้ระยะประมาณ 70 เมตรในโครงสร้างปิด และ 250 เมตรในที่โล่งแจ้ง เพิ่มความสามารถในการกันสัญญาณกวนจากอุปกรณ์อื่นๆ ที่ใช้ความถี่ 2.4 GHz เหมือนกัน และสามารถรองรับอุปกรณ์มาตรฐาน IEEE 802.11b และ IEEE 802.11g ได้

2.2 โพรโทคอล TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโพรโทคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โพรโทคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปได้ถึงปลายทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน คือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มี ความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล

2.3 Wi-Fi Module ESP8266

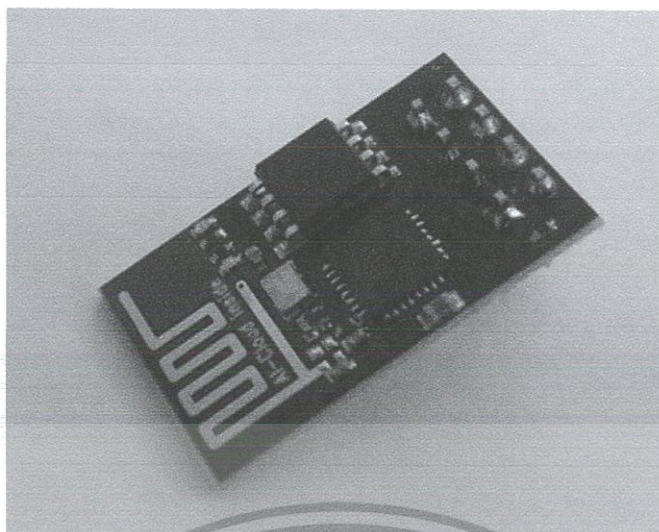
ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (Flash Memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (External Flash Memory) ในการเก็บโปรแกรมที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่า ไอซีไมโครคอนโทรลเลอร์บอร์ดอื่นๆ

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3 V - 3.6 V การนำไปใช้งานร่วมกับเซนเซอร์อื่นๆ ที่ใช้แรงดัน 5 V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200 mA ความถี่คริสตอล 40 MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ Arduino มาก

ขาของโมดูล ESP8266 แบ่งได้ดังนี้

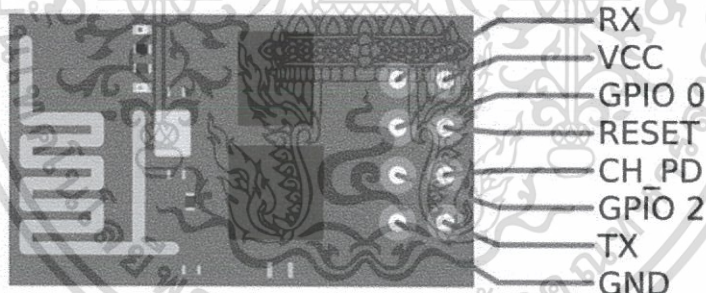
- VCC เป็นขาสำหรับจ่ายไฟเข้าเพื่อให้โมดูลทำงานได้ ซึ่งแรงดันที่ใช้งานได้คือ 3.3 V - 3.6 V
- GND
- Reset และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ ทั้ง 2 ขานี้
 - สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกัน แตกต่างตรงที่ขา Reset สามารถลอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไฟ + เท่านั้น เมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที
- GPIO เป็นขาดิจิตอลอินพุต/เอาต์พุต ทำงานที่แรงดัน 3.3 V
- GPIO15 เป็นขาที่ต้องต่อลง GND เท่านั้น เพื่อให้โมดูลทำงานได้
- GPIO0 เป็นขาสำหรับการเลือกโหมดทำงาน หากนำขานี้ลง GND จะเข้าโหมดโปรแกรม หากลอยไว้หรือนำเข้าไฟ + จะเข้าโหมดการทำงานปกติ
- ADC เป็นขาแอนาล็อกอินพุต รับแรงดันได้สูงสุดที่ 1 V ขนาด 10 บิต การนำไปใช้งานกับแรงดันที่สูงกว่าต้องใช้อุปกรณ์แบ่งแรงดันเข้าช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ESP8266 รุ่นที่นิยมใช้งานคือ ESP-01

ESP8266 รุ่นที่นิยมใช้งานคือ ESP-01 เป็นรุ่นที่เหมาะสมสำหรับการเรียนรู้ และเหมาะสำหรับนำไปใช้งานที่โปรแกรมเล็กๆ มีขาทั้งหมด 8 ขา ได้แก่ VCC CH_PD Reset Rx Tx GPIO0 GPIO2 และ GND โมดูลนี้ทำงานได้ค่อนข้างที่จะช้ามาก หากมีการเขียนโปรแกรมที่ไม่รัดกุมพอ หรือมีคำสั่งทำงานมากๆ



รูปที่ 2.2 ขาของ ESP8266 รุ่น ESP-01

โมดูล ESP8266-01 รุ่นนี้ มีขาใช้งาน 8 ขา ใช้แรงดันไฟฟ้ากระแสตรง 3.3 V มีขา GPIO 2 ขา และขาการสื่อสารแบบ UART คือ TX และ RX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Arduino Microcontroller

2.4.1 Arduino Mega 2560

Arduino Mega 2560 เป็นบอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 ใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input/Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด สามารถเขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB ได้ เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้การพัฒนาไมโครคอนโทรลเลอร์ที่ต้องการบอร์ด Arduino ที่มีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานมากขึ้น

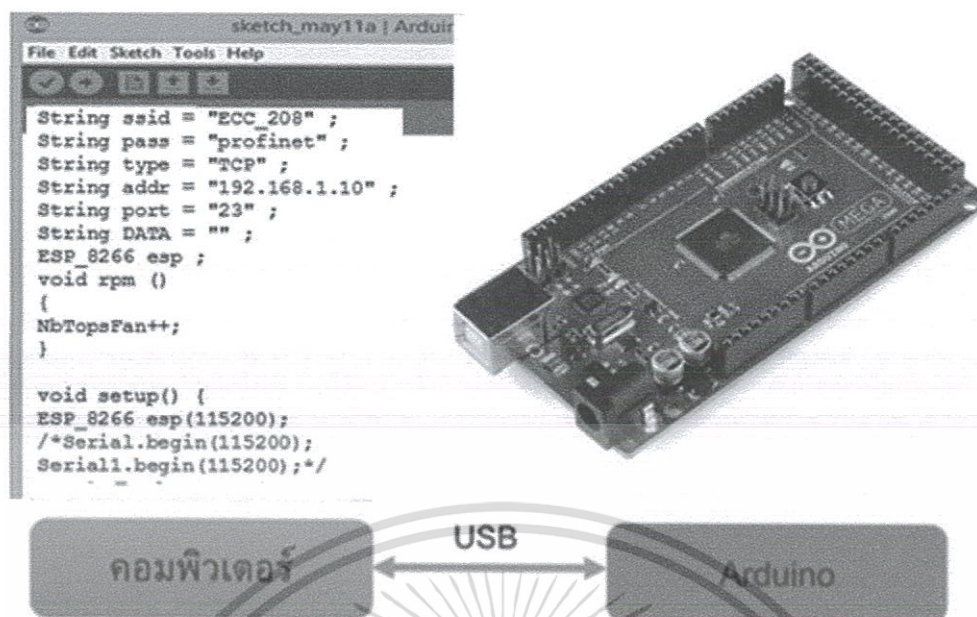


รูปที่ 2.3 บอร์ด Arduino Mega 2560

2.4.2 รูปแบบการเขียนโปรแกรมบน Arduino

โปรแกรมของ Arduino แบ่งได้เป็นสองส่วนคือ Void Setup() และ Void Loop() โดยฟังก์ชัน Setup() เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชันนี้เพียงครั้งเดียวใช้ในการกำหนดค่าเริ่มต้นของการทำงาน ส่วนฟังก์ชัน Loop() เป็นส่วนทำงาน โปรแกรมจะทำคำสั่งในฟังก์ชันนี้ต่อเนื่องตลอดเวลา โดยปกติใช้กำหนดโหมดการทำงานของขาต่างๆ กำหนดการสื่อสารแบบอนุกรม ส่วนของ Loop() เป็นโค้ดโปรแกรมที่ทำงาน เช่น อ่านค่าอินพุต ประมวลผลส่งงานเอาต์พุต โดยส่วนกำหนดค่าเริ่มต้น เช่น ตัวแปรจะต้องเขียนที่ส่วนหัวโปรแกรม ก่อนถึงตัวฟังก์ชัน นอกจากนั้นยังต้องคำนึงถึงตัวพิมพ์เล็ก-ใหญ่ ของตัวแปรและชื่อฟังก์ชันให้ถูกต้อง

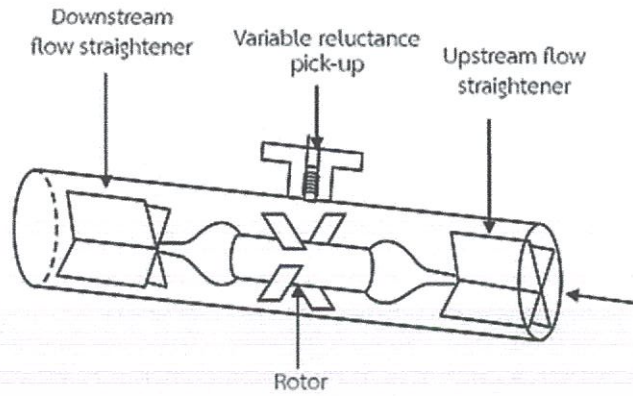
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเขียนโปรแกรมลงในบอร์ด Arduino Mega 2560

2.5 เครื่องมือวัดอัตราการไหล

เครื่องมือวัดอัตราการไหล (Flow Sensor) ที่เลือกใช้ในโครงการนี้เป็นประเภท เทอร์ไบน์ (Turbine) เป็นเครื่องมือวัดอัตราการไหลที่สามารถใช้วัดอัตราการไหลของของไหลได้โดยตรงแบบหนึ่ง โดยหลักการทำงานของเทอร์ไบน์นั้นจะทำหน้าที่แปลงค่าอัตราการไหลของของไหลให้อยู่ในรูปของสัญญาณไฟฟ้าแต่ต้องใช้ร่วมกับอุปกรณ์ตัวอื่นอีกด้วย ภายในของอุปกรณ์วัดอัตราการไหลแบบนี้จะประกอบไปด้วย ใบพัดทำจากวัสดุที่มีคุณสมบัติเป็นสารแม่เหล็กที่ติดตั้งอยู่บนโรเตอร์ ดังแสดงในรูปที่ 2.5 สำหรับการใช้นั้น เมื่อมีของไหลไหลเข้าไปในตัวเรือน และผ่านใบพัดที่ติดอยู่กับโรเตอร์ จะทำให้ใบพัดหมุน และความเร็วในการหมุนของใบพัดจะเป็นสัดส่วนโดยตรงกับอัตราการตรวจจับแม่เหล็ก ลักษณะเป็นขดลวดพันรอบแม่เหล็กถาวรติดตั้งอยู่ที่ตัวเครื่องมือวัด เช่น Hall Effect Sensor มีหลักการทำงานคือ เมื่อใบหมุนรอบแกนเหล็กจะเกิดสนามแม่เหล็กขึ้น ทำให้ Hall Effect Sensor สามารถตรวจวัดสนามแม่เหล็กที่เกิดขึ้นได้ตามความเร็วในการหมุนของใบพัด แล้วให้สัญญาณพัลส์ออกมาตามความเร็วในการหมุนของใบพัด และใช้ตัวนับแบบดิจิทัลตรวจนับจำนวนพัลส์ที่เกิดขึ้น จากนั้นทำการประมวลผลด้วยไมโครโปรเซสเซอร์แปลงหน่วยให้อยู่ในรูปของปริมาตร ซึ่งได้จากการนับจำนวนพัลส์ที่เกิดขึ้น หรืออัตราการไหล ซึ่งได้จากการนับจำนวนพัลส์ในหนึ่งช่วงเวลา หรือเรียกว่า อัตราการเกิดพัลส์



รูปที่ 2.5 โครงสร้างของเครื่องมือวัดอัตราการไหลแบบเทอร์ไบน์



รูปที่ 2.6 Hall effect sensor

เครื่องมือวัดการไหลชนิดนี้มีลักษณะความเป็นเชิงเส้น (Linearity) ที่ดีที่อัตราการไหลสูง ส่วนที่อัตราการไหลต่ำค่าการไหลที่วัดได้จะได้รับผลกระทบจากแรงต้าน เนื่องจากความเสียดทาน โดยทั่วไปเครื่องมือวัดการไหลชนิดเทอร์ไบน์มีย่านการวัด (Range) การไหลของของเหลวอยู่ในช่วง 1 ถึง 100,000 ลิตรต่อนาที และสำหรับการวัดการไหลของก๊าซมีย่านการใช้งานในช่วง 5 ถึง 100,000 ลิตรต่อนาที ไม่สามารถใช้วัดการไหลของของไหลที่มีสารแขวนลอยปะปนได้ และในการติดตั้งเครื่องมือวัดต้องพิจารณาระยะความตรงของท่อทั้งด้านหน้าก่อนเข้าตัวเครื่องมือวัด และระยะด้านหลัง เพื่อป้องกันไม่ให้ค่าความเร็วที่วัดได้คลาดเคลื่อน (Error)

2.6 สมการต่อเนื่อง (Continuity Equation)

อัตราการไหล (Flow Rate หรือ Discharge) คือ ปริมาณของไหลที่ไหลผ่านพื้นที่หน้าตัดใดๆ ที่กำหนด ต่อหนึ่งหน่วยเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการไหลเชิงปริมาตร (Volume Flow Rate) : Q มีสมการทั่วไป ดังนี้

$$Q = Av \quad (2.1)$$

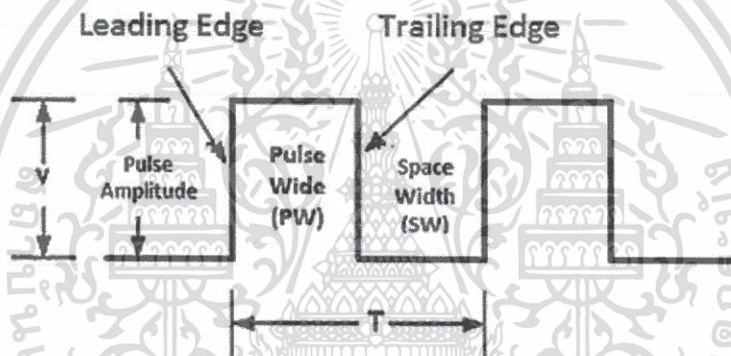
เมื่อ A คือ พื้นที่หน้าตัดการไหล

v คือ ความเร็วเฉลี่ยของการไหล

หน่วยของอัตราการไหลในระบบ SI คือ $\frac{m^3}{s}$ และในระบบอังกฤษ คือ $\frac{ft^3}{s}$

2.7 คุณลักษณะของรูปคลื่นพัลส์

รูปคลื่นพัลส์ในอุดมคติ (Ideal Pulse Waveform) คือ รูปคลื่นที่พิจารณาโดยไม่คิดค่าช่วงเวลาศูนย์เสี้ยวที่เกิดขึ้นในขณะที่พัลส์มีการเปลี่ยนแปลงจากระดับต่ำไปสูง และจากสูงไปต่ำดังแสดงในรูปที่ 2.7



รูปที่ 2.7 คลื่นพัลส์ในอุดมคติ

จากรูปสัญญาณพัลส์เมื่อพิจารณาจะพบว่า

1. ขนาดของพัลส์ (Pulse Amplitude) คือ ค่าที่วัดจากระดับแรงดันศูนย์โวลต์ไปถึงค่าสูงสุด (Peak Value)
2. ขอบหน้าของพัลส์ (Leading Edge หรือ Rising Edge หรือ Positive-Going Edge) คือ ค่าของสัญญาณที่เปลี่ยนค่าจากระดับแรงดันต่ำไปสู่ระดับแรงดันสูงสุด พิจารณาที่เวลา $t = 0$
3. ขอบหลังของพัลส์ (Trailing Edge หรือ Falling Edge หรือ Negative-Going Edge) คือ ค่าของสัญญาณที่เปลี่ยนค่าจากระดับแรงดันสูงสุดไปสู่ระดับต่ำสุดของพัลส์
4. คาบเวลาของพัลส์ (Time Period) ใช้ตัวย่อ T คือ ระยะเวลาที่วัดจากขอบหน้าของพัลส์ที่เวลา $t = 0$ จนถึงขอบหน้าของพัลส์รูปคลื่นต่อไป
5. ความถี่ของพัลส์ (Pulse Repetition Frequency หรือ PRF) คือ ค่าส่วนกลับของเวลา หรือ $PRF = PPS$ (Pulse Per Sec)

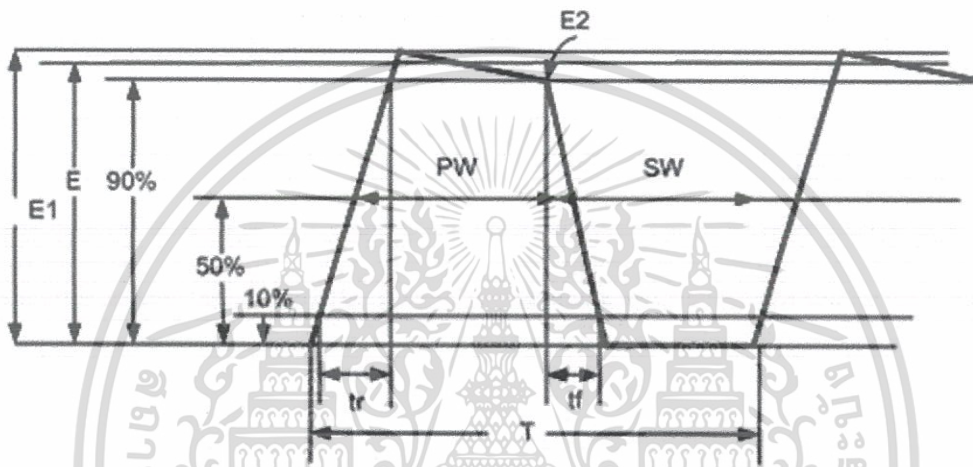
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ความกว้างของพัลส์ (Pulse Width หรือ PW) คือ ช่วงเวลาที่วัดจากขอบหน้าของพัลส์จนถึงขอบหลังของสัญญาณพัลส์

7. ช่องว่างระหว่างพัลส์ (Space Width หรือ SW) คือ ช่วงเวลาที่วัดจากขอบหลังของพัลส์รูปคลื่นแรกจนถึงขอบหน้าพัลส์รูปคลื่นถัดไป

8. อัตราส่วนของพัลส์ (M/S Ratio) คือ อัตราส่วนระหว่างความกว้างของพัลส์ PW กับช่องว่างระหว่างพัลส์ (PW/SW)

9. วัฏจักรหน้าที่ (Duty Cycle) คือ ร้อยละของอัตราส่วนระหว่างความกว้างของพัลส์ (PW) กับคาบเวลาของพัลส์ (T)



รูปที่ 2.8 คลื่นพัลส์ในทางปฏิบัติ

รูปคลื่นพัลส์ในทางปฏิบัติจะแตกต่างกับทฤษฎีดังนี้

1. เวลาไต่ขึ้น (Rise Time) คือ ระยะเวลาของสัญญาณพัลส์ขอบหน้าเปลี่ยนแปลงจากขนาดของสัญญาณที่ 10% ไปจนถึง 90% ของขนาดสูงสุด

2. เวลาดตก (Fall Time) คือ ระยะเวลาของสัญญาณพัลส์ขอบหลังตกจากระดับสัญญาณที่ 90% ลงไปถึง 10% ของขนาดสูงสุด

3. ความกว้างของพัลส์ (PW) ต้องวัดมาจากค่าเฉลี่ยของความกว้างทั้งหมด คือ วัดที่ 50% ของขนาดสูงสุดของแรงดันเฉลี่ย และค่าช่องว่างของพัลส์ (SW) ก็เช่นเดียวกัน ดังนั้นจะได้

$$T = PW + SW$$

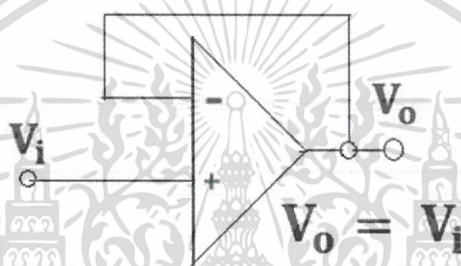
4. ค่าขนาดแรงดันเฉลี่ยของพัลส์ (Average Pulse Amplitude) คือ ค่าเฉลี่ยจากค่าสูงสุดของขอบหน้าและขอบหลัง

5. ความลาดเอียง (Tilt) คือ ความลาดเอียงบริเวณระดับสัญญาณสูงสุดขอบหน้าถึงขอบหลังของพัลส์

6. ค่าเฉลี่ยของรูปคลื่นพัลส์ (V_{av}) คือ การนำเอาพื้นที่ที่เกิดพัลส์ ในช่วงความกว้างของพัลส์ (PW) ทั้งด้านบวกและด้านลบของพัลส์มารวมกัน และหารด้วยคาบเวลาของพัลส์

2.8 วงจรบัฟเฟอร์

วงจรบัฟเฟอร์ (Buffer) หรือวงจรตามแรงดัน (Voltage Follower) เป็นวงจรที่ให้อัตราการขยายเท่ากับ 1 ใช้เป็นตัวเชื่อมระหว่างวงจรสองวงจรเข้าด้วยกัน วงจรบัฟเฟอร์นี้จะมีค่าความต้านทานอินพุตสูง เมื่อนำไปต่อเป็นโหลดวงจรอื่น ก็ไม่ทำให้ให้แรงดันเอาต์พุตของวงจรอื่นนั้นลดลง และเมื่อนำไปขับวงจรอื่นก็สามารถจ่ายกระแสไปให้วงจรอื่นได้ โดยไม่ทำให้ระดับแรงดันเอาต์พุตของมันเองลดลง

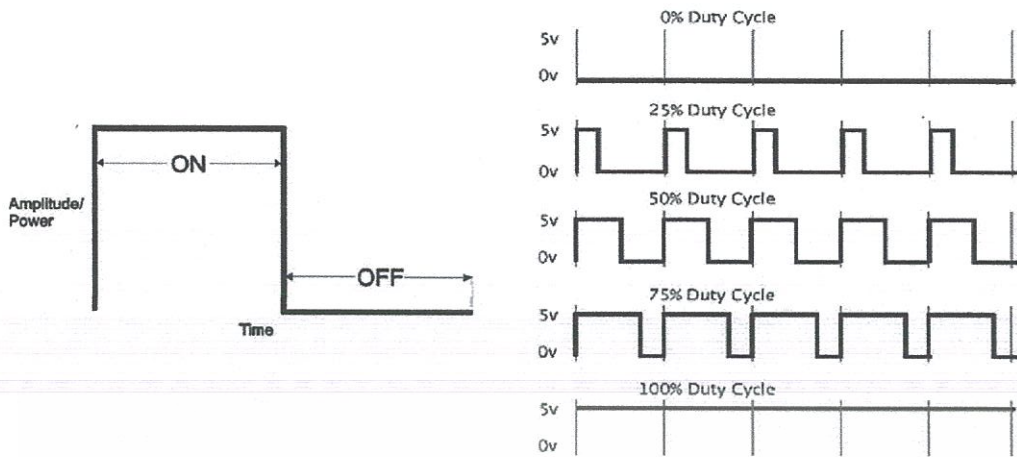


รูปที่ 2.9 วงจรบัฟเฟอร์แบบใช้โอปแอมป์

2.9 สัญญาณ PWM

PWM ย่อมาจาก Pulse Width Modulation คือ การมอดูเลชันทางความกว้างของพัลส์ PWM เป็นการปรับเปลี่ยนที่สัดส่วน และความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์ จะไม่มีการเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดิวตี้ไซเคิล (Duty Cycle) นั้นเอง ซึ่งค่าของดิวตี้ไซเคิลคือ ช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 50% ก็หมายถึงใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูปที่ 2.10 และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดิวตี้ไซเคิลสามารถจะหาได้จากค่าความสัมพันธ์ดังนี้

$$\text{ค่าดิวตี้ไซเคิล} = (\text{ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ}) \times 100 \quad (2.2)$$



รูปที่ 2.10 กราฟแสดงค่า PWM ในรูปของค่าตัวเลข



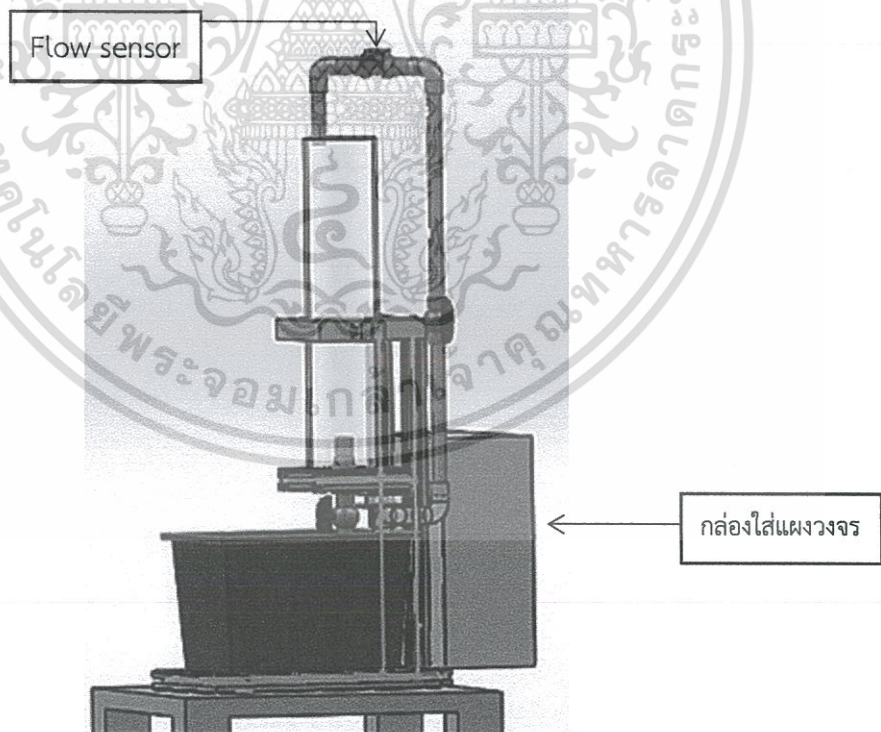
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนการดำเนินงาน

3.1 โครงสร้างของระบบ

ในโครงการระบบควบคุมระดับน้ำอัตโนมัติแบบไร้สายนี้ (Wireless Level Control System) เริ่มต้นจากการศึกษาข้อมูลต่างๆ ที่เกี่ยวข้อง และได้แบ่งงานออกเป็นกลุ่มต่างๆ คือ Hardware, Software, Level Transmitter, Flow Transmitter, Actuator ในเบื้องต้นส่วน Hardware จะทำการออกแบบโครงสร้าง และจัดวางตำแหน่งของอุปกรณ์เซนเซอร์ต่างๆ ของระบบ ด้วยโปรแกรม SolidWorks และใช้โปรแกรม MakerWare ในการขึ้นรูป 3 มิติ ของชิ้นงานบางส่วน โดยในส่วนของ Flow Sensor จึงขอกกล่าวถึง Flow Sensor ซึ่งได้ติดตั้งแผงวงจรในกล่องด้านหลัง ดังรูปที่ 3.1 โดยเลือกใช้ Flow Sensor ขนาด 1/2" ให้ตรงกับขนาดท่อ และได้ทำการติดตั้ง Flow Sensor ในแนวนอน เพื่อป้องกันการไหลกลับของน้ำเนื่องจากแรงโน้มถ่วงของโลก ในกรณีที่มีการสั่งการทำงานมอเตอร์ที่ความเร็วต่ำๆ ซึ่งอาจทำให้ Flow Sensor เสียหายได้ เพราะ Flow Sensor ที่เลือกใช้สามารถปล่อยให้น้ำไหลผ่านได้ทางเดียว การออกแบบการติดตั้ง Flow Sensor ในแนวนอน จึงเหมาะสมที่สุดในการทดลอง



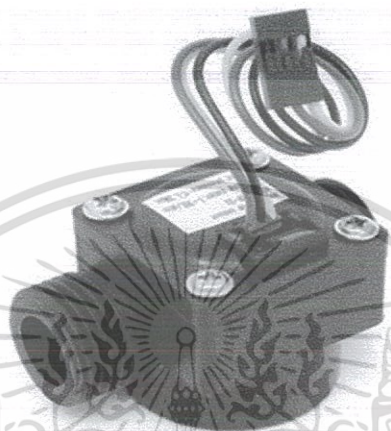
รูปที่ 3.1 โครงสร้างระบบควบคุมระดับน้ำแบบไร้สายที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเลือกใช้อุปกรณ์

3.2.1 เซนเซอร์วัดอัตราการไหลของน้ำ (Flow Sensor)

โครงการนี้ได้เลือกใช้ เซนเซอร์วัดอัตราการไหลของน้ำ (Electronic Flow Sensor Electronic Flow Meter) 1-30 ลิตรต่อนาที 1/2" เพราะสามารถต่อเข้าได้กับท่อขนาด 1/2"



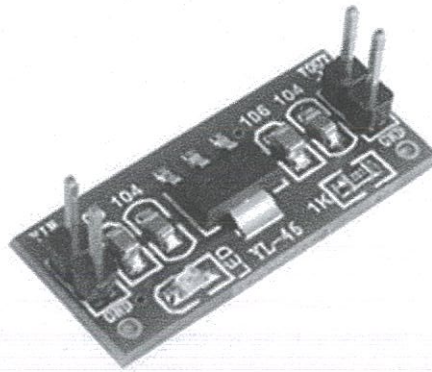
รูปที่ 3.2 Flow Sensor

คุณสมบัติของ Flow Sensor

- The Level of The Test Pulse Frequency (Hz) = $[8.1Q - 3] \pm 10$
(Q = Flow Rate L/min.)
- Minimum Rated Working Voltage : DC 3.5 , 5-24 V
- Maximum Current : 15 mA DC 5V
- Voltage Range : DC 5-18 V
- Operating Temp : ≤ 80 °C
- Allow Compression : Water Pressure 1.75 Mpa Below

3.2.2 Regulator

โครงการนี้ได้เลือกใช้ โมดูลแปลงไฟ 4.5 - 7 V ไปเป็น 3.3 V กระแสสูงสุด 800 mA พร้อม LED แสดงผล



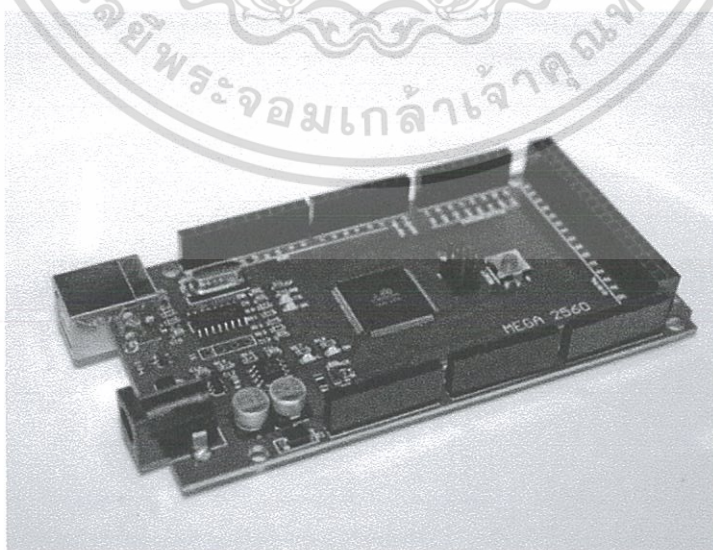
รูปที่ 3.3 Regulator

คุณสมบัติของ Regulator

- Input : 4.5 V dc - 7 V (1 V Input Voltage Must be Higher Than to The Output Voltage Above.)
- Output : 3.3 V, 800 mA (Load Current Should Not Exceed 800 mA)
- Size : 2.5 cm x 1.1 cm
- With Power Indicator Light (Red)

3.2.3 Arduino MEGA 2560 R3

โครงการนี้ได้เลือกใช้ Arduino MEGA 2560 R3 เพราะสามารถใช้ได้กับทุก Windows ได้แก่ Windows xp, Windows 7, Windows 8, Windows 10



รูปที่ 3.4 บอร์ด Arduino MEGA 2560 R3

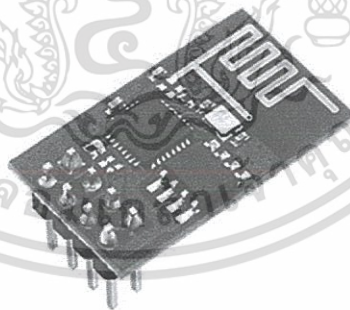
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ Arduino MEGA 2560 R3

- Operating Voltage : 5 V
- Input Voltage (Recommended) : 7-12 V
- Input Voltage (Limits) : 6-20 V
- Digital I/O Pins : 54 of which 14 provide PWM Output, 4 UART TTL
- Analog Input Pin : 16
- DC Current per I/O Pin : 40 mA
- DC Current for 3.3 V Pin : 50 mA
- Flash Memory : 256 KB of which 8 KB used by Bootloader
- SRAM : 8 KB
- EEPROM : 4 KB
- Clock Speed : 16 MHz
- Microcontroller : ATmega2560

3.2.4 โมดูล Wi-Fi ESP8266

โครงการนี้ได้เลือกใช้ Wi-Fi ESP8266 ESP-01 เพราะเป็นโมดูล Wi-Fi ขนาดเล็ก ใช้พลังงานต่ำ รองรับการใช้งานได้หลากหลายรูปแบบทั้ง Client, Access Point และ Client+AP โมดูล Wi-Fi ESP8266 ใช้การเชื่อมต่อแบบ Serial UART ที่ไฟ 3.3 V ใช้คำสั่งควบคุมการทำงานแบบ AT Command ทำให้เขียนโปรแกรมเชื่อมต่อได้สะดวก



รูปที่ 3.5 โมดูล Wi-Fi ESP8266 ESP-01

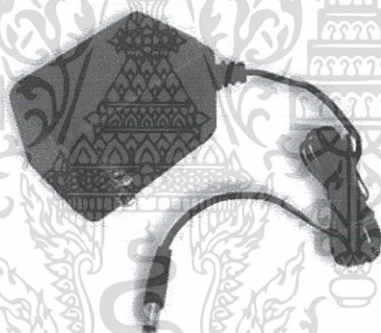
คุณสมบัติของ Wi-Fi ESP8266 ESP-01

- 802.11 b/g/n
- Wi-Fi Direct (P2P), Soft-AP
- Integrated TCP/IP Protocol Stack

- Integrated TR Switch, Balun, LNA, Power Amplifier and Matching Network
- Integrated PLLs, Regulators, DCXO and Power Management Units
- +19.5dBm Output Power in 802.11b Mode
- Power Down Leakage Current of < 10 uA
- Integrated Low Power 32-bit CPU Could be used as Application Processor
- SDIO 1.1/2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU Aggregation & 0.4 ms Guard Interval
- Wake up and Transmit packets in < 2 ms
- Standby Power Consumption of < 1.0 mW (DTIM3)

3.2.5 Adaptor

อุปกรณ์จ่ายไฟขนาดเล็ก นำมาใช้ในโครงการนี้เพื่อจ่ายแรงดันขนาด 12 V ให้แก่ Board Arduino MEGA 2560



รูปที่ 3.6 AC/DC Adaptor

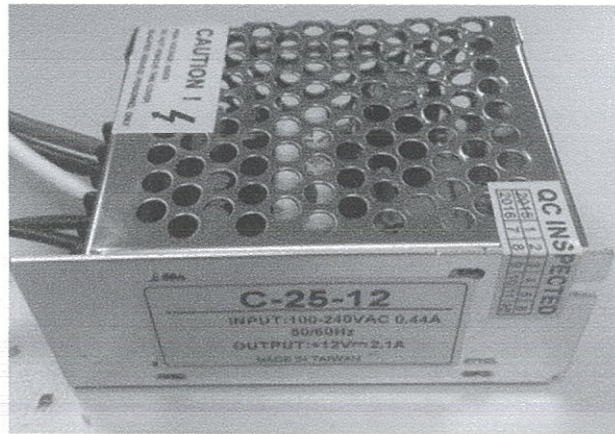
คุณสมบัติของ Adaptor

- Model : YX-4116C 12 A
- Input : AC 220 V 50 Hz
- Output : DC 12 V 500 mA

3.2.6 Power Supply

แหล่งจ่ายไฟตรง ซึ่งสามารถเปลี่ยนแรงดันไฟจากไฟสลับโวลต์สูง ให้เป็นแรงดันไฟตรงค่าโวลต์ต่ำ เพื่อใช้ในงานอิเล็กทรอนิกส์ได้ ซึ่งการนำมาใช้ในโครงการนี้เพื่อจ่ายไฟ 12 V ให้กับไอซี LM358 ในวงจร Buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



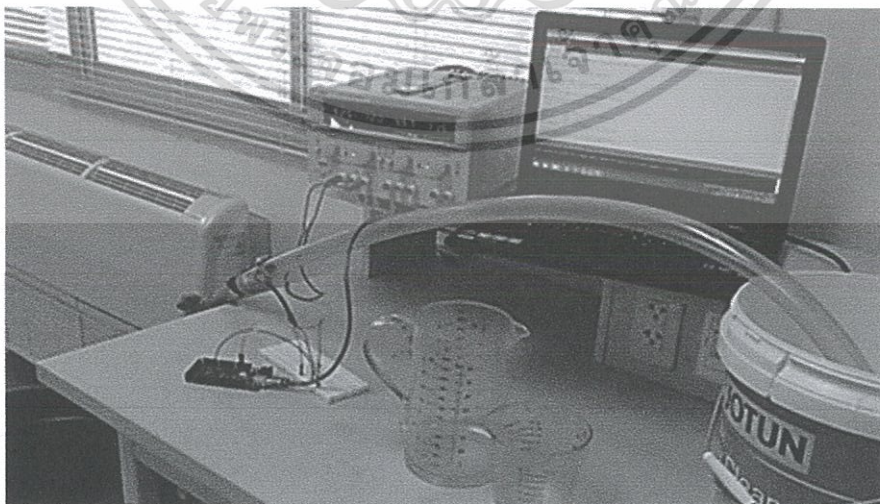
รูปที่ 3.7 Power Supply

คุณสมบัติของ Power Supply

- Input : 100-240 V
- Output : 12 V 2.1 A
- ขนาด 58 x 85 x 38 มิลลิเมตร

3.3 การทดลองหาค่าความคลาดเคลื่อนระหว่างค่าอัตราการไหลที่ได้จากการเขียนโปรแกรมและการวัดอัตราการไหลจริง

หลังจากที่ศึกษาข้อมูลต่างๆ ที่เกี่ยวข้องกับ Flow Sensor เพื่อใช้วัดอัตราการไหลของน้ำในท่อ เลือกใช้ Flow Sensor แบบ Turbine Flow Meter ซึ่งมีอุปกรณ์และขั้นตอนในการทดลองดังนี้



รูปที่ 3.8 การเชื่อมต่ออุปกรณ์เพื่อทดลอง

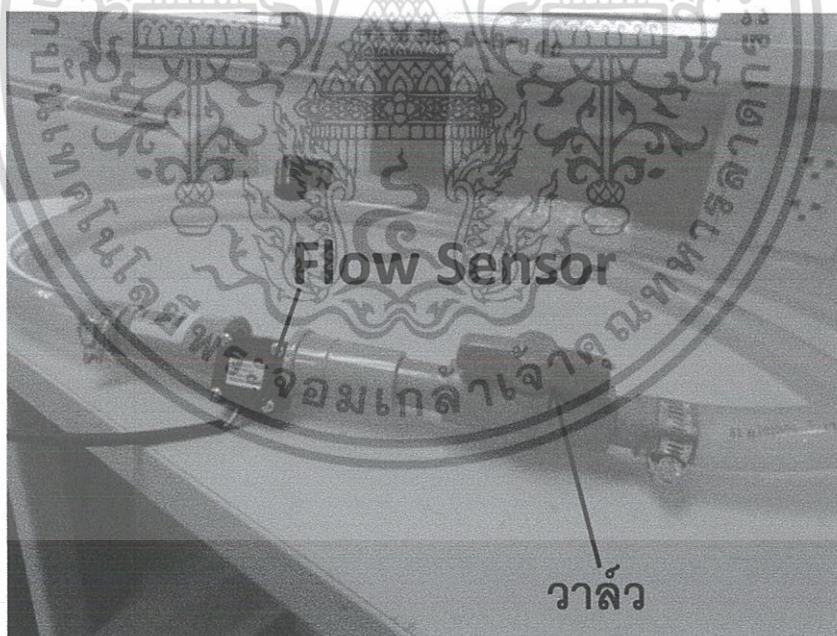
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 อุปกรณ์การทดลอง

1. ถังน้ำ 2 ใบ
2. ปั๊มน้ำ DC 12 V 70 L/min
3. สายยาง ขนาด 1 นิ้ว
4. Supply 12 V
5. วาล์ว ขนาดท่อ ½ นิ้ว
6. Turbine Flow Meter รุ่น FS200A G1/2 นิ้ว, Flow Range: 0.5-30 L/min
7. ถ้วยตวง
8. Arduino Mega 2560
9. ตัวต้านทานขนาด 10 k Ω
10. โปรแกรม Arduino

3.3.2 ขั้นตอนการทดลอง

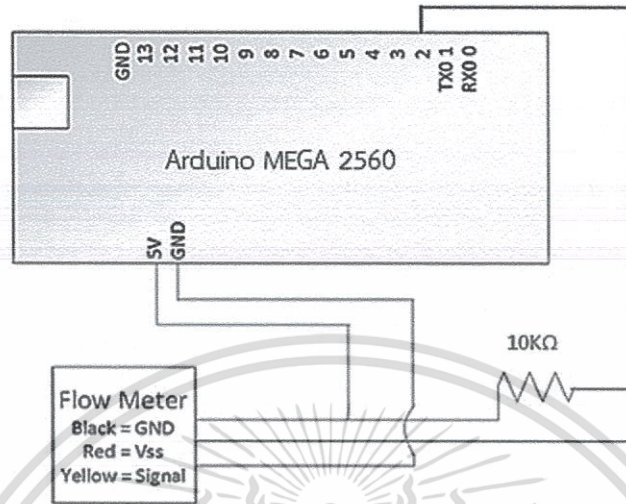
1. ต่อ Flow Sensor, วาล์วและสายยางดังรูปที่ 3.9



รูปที่ 3.9 การต่ออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ต่อดวงจรตามรูปที่ 3.10



รูปที่ 3.10 วงจรทดสอบ Flow Sensor

3. เขียน Code Arduino เพื่อคำนวณค่า Flow Rate ออกมาเป็น L/min และเทียบเป็นเปอร์เซ็นต์

คำสั่งที่ใช้ในการทดลอง

```
volatile int NbTopsFan; //measuring the rising edges of the signal

int Calc;
int hallsensor = 2; //The pin location of the sensor
void rpm () //This is the function that the interrupt calls
{
  NbTopsFan++; //This function measures the rising and falling edge of the
  //hall effect sensors signal
}
// The setup() method runs once, when the sketch starts
void setup() //
{
  pinMode(hallsensor, INPUT); //initializes digital pin 2 as an input
  Serial.begin(9600); //This is the setup function where the serial port is
  //initialised,
  attachInterrupt(0, rpm, RISING); //and the interrupt is attached
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop ()
{
  int outputValue;
  NbTopsFan = 0; //Set NbTops to 0 ready for calculations
  sei(); //Enables interrupts
  delay (1000); //Wait 1 second
  cli(); //Disable interrupts
  Calc = (NbTopsFan/ 7.5); //(Pulse frequency) / 7.5Q, = flow rate
  outputValue = map(Calc, 0, 13, 0, 100);
  Serial.print("flowrate = " );
  Serial.print(Calc, DEC); //Prints the number calculated above
  Serial.print (" L/min\r\n"); //Prints "L/min" and returns a new line
  Serial.print("\t%output = ");
  Serial.println(outputValue);
}

```

4. ต่อบอร์ด Arduino กับคอมพิวเตอร์แล้ว Upload Code ลงบอร์ด Arduino
5. ต่อสายทางด้านที่ติดกับวาล์วเข้ากับปั้มน้ำ
6. ใส่น้ำในถัง 1 ใบแล้วจุ่มปั้มน้ำลงในถัง โดยสายยางอีกด้านให้ใส่ไว้ในถังใบเดียวกัน
7. ต่อ Supply 12 V กับปั้มน้ำ
8. ไล่อากาศในสายยางโดยเปิดวาล์วให้น้ำไหลจนเต็มพื้นที่สายยาง
9. นำสายยางด้านที่ไม่ได้ต่อบั้มน้ำไปใส่ในถังเปล่าอีก 1 ใบ
10. เปิดปั้มน้ำแล้วจึงเปิดวาล์วพร้อมๆ กับจับเวลา 6 วินาที
11. ใช้ถ้วยตวงวัดปริมาณน้ำในถังที่ผ่าน Flow Sensor แล้วคูณด้วย 10 เพื่อหาค่า

Flow Rate ในหน่วย L/min แล้วบันทึกผล

12. อ่านค่า Flow Rate ใน Serial Monitor แล้วบันทึกผล
13. ทำซ้ำตั้งแต่ข้อ 10-12 จำนวน 10 ครั้ง แล้วหาค่าผิดพลาด (Error) โดย

$$\text{Flow Rate} = \frac{\sum(\text{ค่าเฉลี่ย} - \text{ค่าจริง})}{\text{จำนวนครั้งที่ทดลอง}} \quad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ค่าเฉลี่ยหา = $\frac{\sum \text{ค่าจริง}}{\text{จำนวนครั้งที่ทดลอง}}$ (3.2)

14. ทำซ้ำตั้งแต่ข้อ 10-13 แต่เปลี่ยนเวลาในการวัดเป็น 1 นาที (หา Flow Rate จากปริมาณน้ำ โดยปริมาณน้ำที่วัดได้เทียบกับเวลาคือค่า Flow Rate)

3.4 การทดสอบ Wi-Fi Module ESP-8266

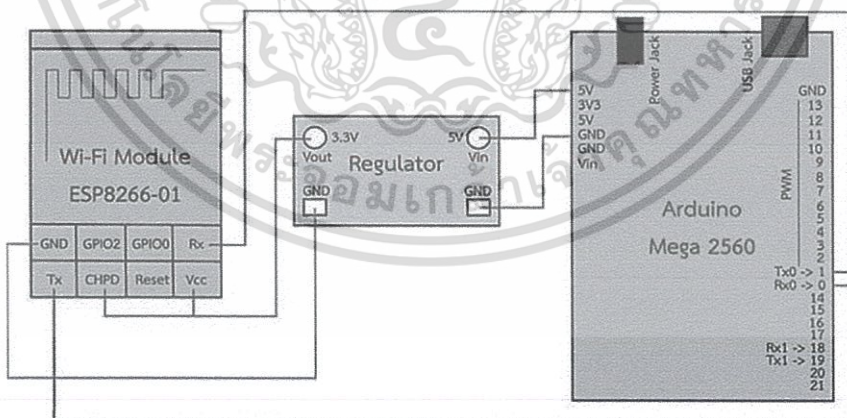
ใช้คำสั่ง AT Command ในการตรวจสอบว่าตัว Wi-Fi Module สามารถรับ-ส่งข้อมูลได้จริง โดยกำหนดให้ Wi-Fi ESP8266 เป็น Client และโปรแกรม Hercules เป็น TCP Server หลังจากนั้นต้องวงจรทดสอบ ESP8266 ระหว่างบอร์ด Arduino MEGA 2560, Wi-Fi Module, Regulator และคอมพิวเตอร์ ดังรูปที่ 3.11

3.4.1 อุปกรณ์การทดลอง

1. ESP8266 ESP-04 Wi-Fi Module
2. บอร์ด Arduino Mega 2560
3. โปรแกรม Arduino

3.4.2 ขั้นตอนการทดลอง

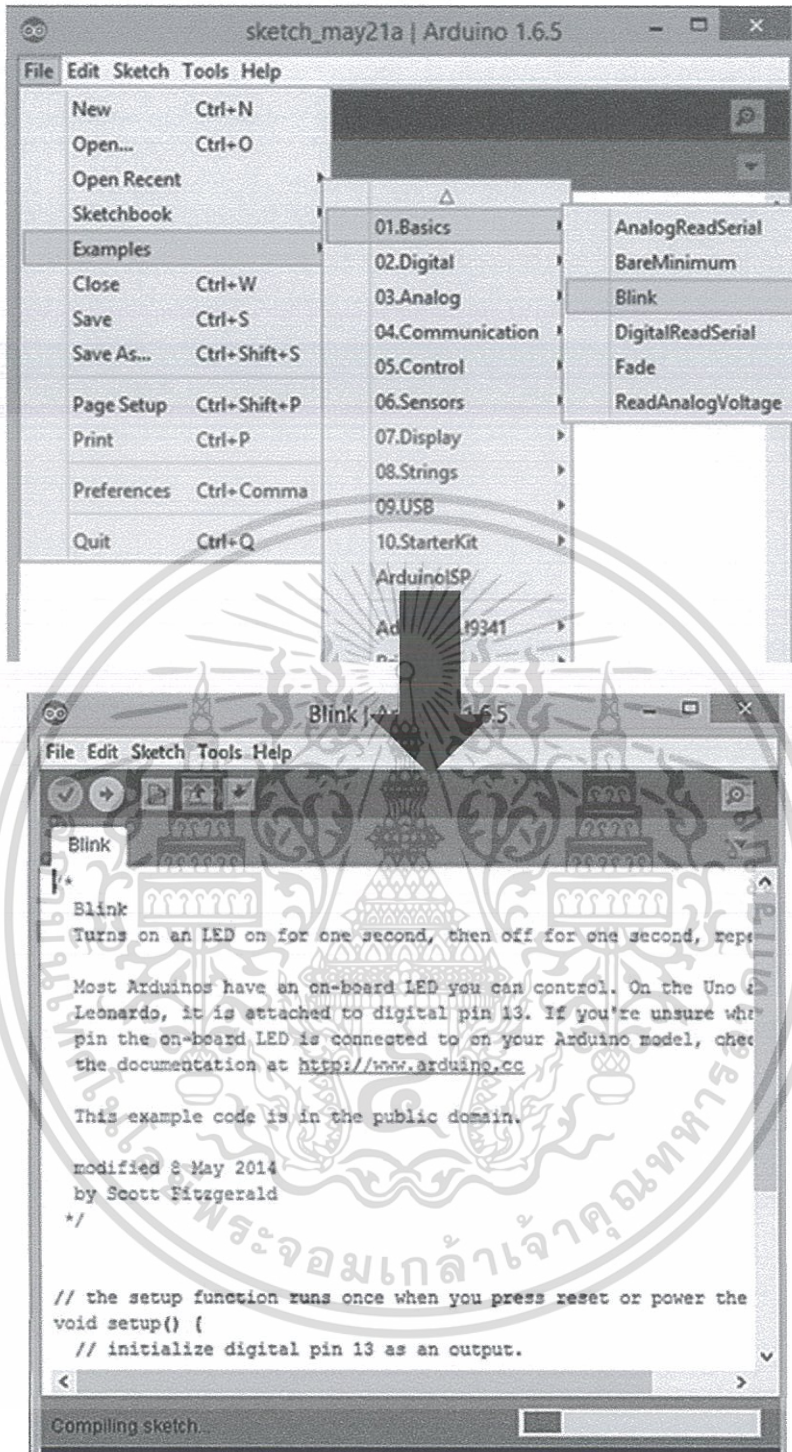
1. ต้องวงจรตามรูปที่ 3.11



รูปที่ 3.11 วงจรทดสอบ Wi-Fi ESP8266

2. Upload โปรแกรมไฟกระพริบเข้าไปในบอร์ด เพื่อให้แน่ใจว่าตัวไอซีไม่ได้ใช้ช่องสัญญาณ Serial Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การ Upload ไฟกระพริบ

3. เปิด Serial Monitor ขึ้นมา ตั้งค่า Baud Rate 115200 และปรับช่องในรูปให้เป็น Both NL&CR

4. พิมพ์คำสั่ง AT แล้วกด Enter ก็จะได้ผลลัพธ์จาก ESP8266 ตอบกลับมาว่า OK เป็นอันว่าเชื่อมต่อสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. พิมพ์คำสั่ง AT+GMR ซึ่งเป็นคำสั่งในการเช็คเวอร์ชัน กด Enter จะแสดงเวอร์ชัน Firmware ของ Wi-Fi

6. ทำการเซตโหมด Wi-Fi Module ESP8266 ซึ่งสามารถตั้งโหมดได้ 3 โหมด โดยพิมพ์ AT+CWMODE=<โหมด> : 1 = โหมด Sta

2 = โหมด AP

3 = โหมด Sta และ AP

ปกติจะใช้โหมด Sta สำหรับ Router และใช้ AP สำหรับ Devices

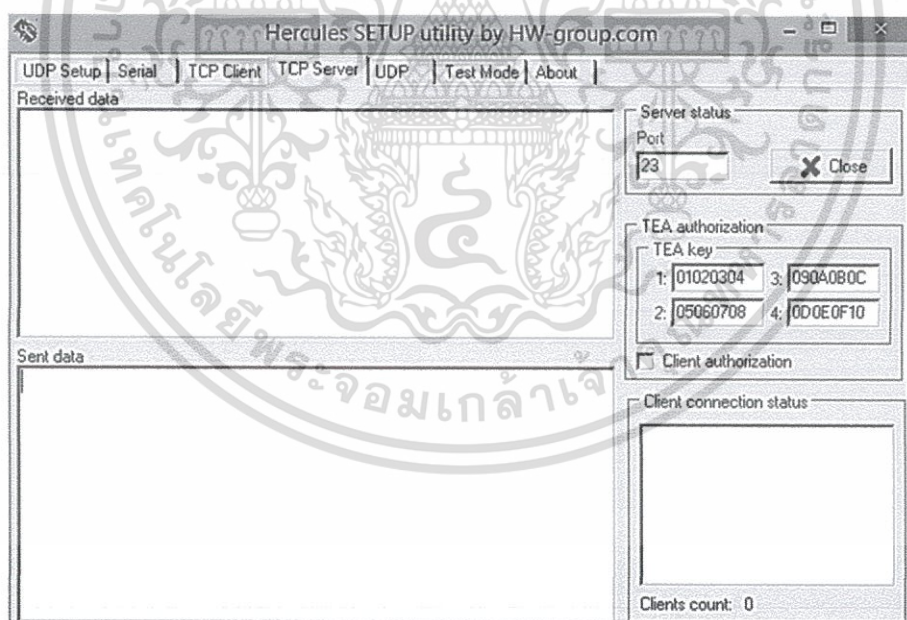
7. ค้นหาเครือข่าย Wi-Fi ที่ใช้ได้ โดยพิมพ์ AT+CWLAP เมื่อทำการกด Enter ก็ จะเห็นรายชื่อ W-iFi ที่อยู่รอบๆ ตัวปรากฏขึ้นมา

8. ทำการเชื่อมต่อระหว่าง Wi-Fi ESP8266 กับ Wi-Fi Router เข้าด้วยกันโดยพิมพ์ AT+CWJAP="ssid","pwd" เช่น ในกรณีที่เชื่อมต่อกับ ECC_208 จะต้องใส่รหัส Profinet แล้วพิมพ์ AT+CWJAP="ECC_208" , "Profinet" ก็จะขึ้นว่า OK

9. ตรวจสอบผลการเชื่อมต่อโดยพิมพ์ AT+CWJAP?

10. พิมพ์ AT+CIPMUX=0 เพื่อเปิดโหมดการเชื่อมต่อแบบจุดเดียว

11. เปิดโปรแกรม Hercules ไปที่ TCP Server แล้วเปิด Port



รูปที่ 3.13 โปรแกรม Hercules

13. พิมพ์ AT+CIPSTART = "TCP", "192.168.1.35", 23 ลงใน AT Command ไปยัง Server ตามเลข IP เครื่องและ Port ที่เลือกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. พิมพ์ AT+CIPSEND = 5 คำสั่ง ส่ง Data จำนวนความยาว 5 Byte หลังจากกดคำสั่ง จะมีเครื่องหมาย ">" ให้พิมพ์ข้อมูลที่จะส่ง เช่น "ABCDE" แล้วข้อมูลจะถูกส่งไปยังโปรแกรม Hercules

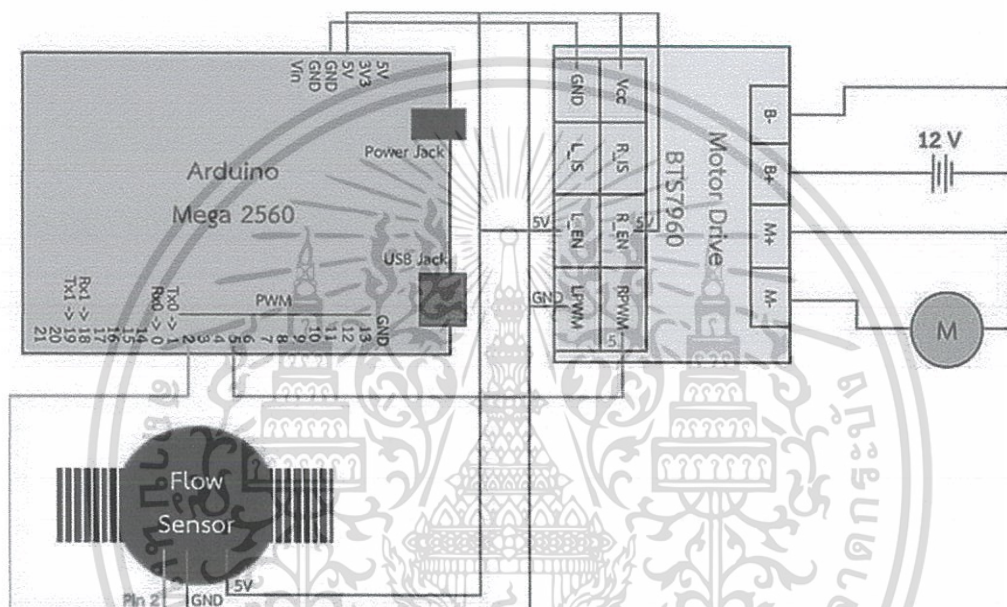


รูปที่ 3.14 การส่งข้อมูลระหว่าง Client และ Serve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

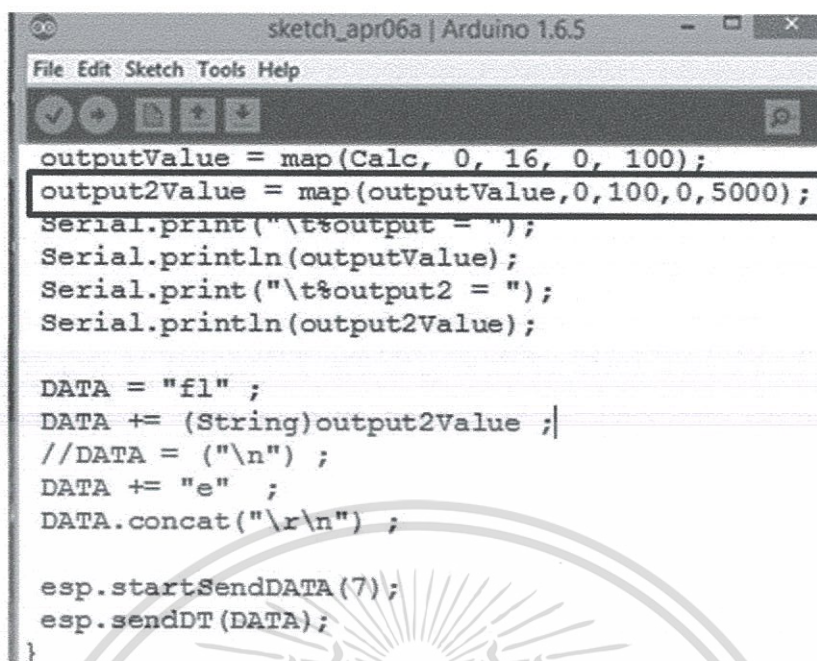
3.5 ทดสอบหาค่าอัตราการไหลของน้ำเปรียบเทียบกับเปอร์เซ็นต์การทำงานของปั๊ม

1. ต่อบางจร ดังรูปที่ 3.15
2. ใช้คำสั่ง map เทียบค่าจากเปอร์เซ็นต์ Flow Rate ให้เป็นแรงดัน
3. ทำการเพิ่มคำสั่งปรับ Speed Motor ลงไปในโปรแกรมที่ใช้ในการทดลอง ในหัวข้อที่ 3.3
4. กำหนดค่า Speed Motor เริ่มตั้งแต่ 45% ซึ่งเป็นค่าที่ Motor สามารถปั๊มน้ำขึ้นไปถึง Flow Sensor ได้ จากนั้นเปลี่ยนค่าไปเรื่อยๆ ทีละ 5% จนกระทั่งถึง 100%
5. บันทึกค่าแรงดัน และเปอร์เซ็นต์ที่อ่านได้ในแต่ละช่วงจากโปรแกรม Arduino



รูปที่ 3.15 วงจร Speed Motor และ Flow Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

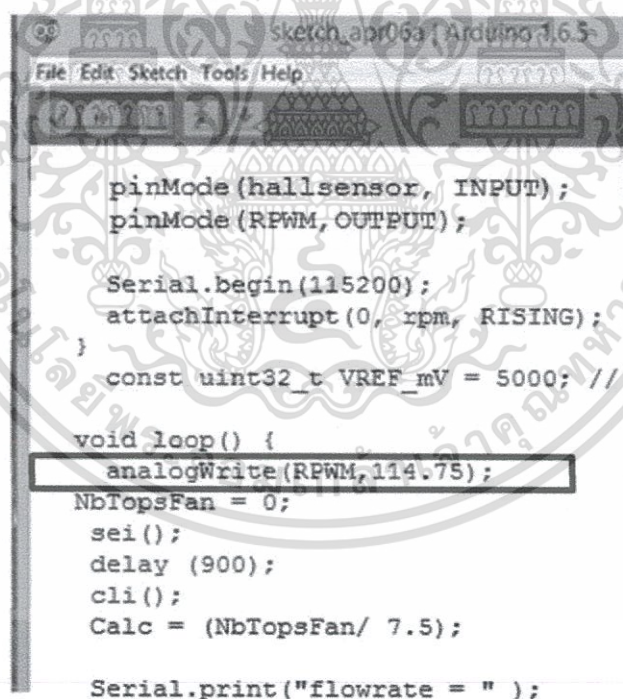
outputValue = map(Calc, 0, 16, 0, 100);
output2Value = map(outputValue, 0, 100, 0, 5000);
Serial.print("\toutput = ");
Serial.println(outputValue);
Serial.print("\toutput2 = ");
Serial.println(output2Value);

DATA = "f1" ;
DATA += (String)output2Value ;
//DATA = ("\n") ;
DATA += "e" ;
DATA.concat("\r\n") ;

esp.startSendDATA(7);
esp.sendDT(DATA);
}

```

รูปที่ 3.16 การใช้คำสั่ง map ค่าจากเปอร์เซ็นต์ให้เป็นแรงดัน



```

pinMode(hallsensor, INPUT);
pinMode(RPWM, OUTPUT);

Serial.begin(115200);
attachInterrupt(0, rpm, RISING);
}
const uint32_t VREF_mV = 5000; //

void loop() {
  analogWrite(RPWM, 114.75);
  NbTopsFan = 0;
  sei();
  delay(900);
  cli();
  Calc = (NbTopsFan/ 7.5);

  Serial.print("flowrate = " );
}

```

รูปที่ 3.17 การกำหนดค่าเปอร์เซ็นต์การทำงานของปั๊ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การเขียน Library ที่ใช้สำหรับ Wi-Fi Module ESP8266-01

ESP_8266 Library ถูกเขียนเพื่อใช้งาน Wi-Fi Module ESP8226 โดยใช้ Software Serial สามารถใช้งานได้กับ Arduino Mega 2560 R3 และต้องใช้กับ ESP8226 ที่มี Firmware Version 0.9.2.2 เป็นต้นไป เนื่องจากเป็น Version ที่มีการพัฒนาให้สามารถปรับค่า Baud Rate ให้เหมาะสมกับ Software Serial ของ Arduino ได้ โดย ESP_8266 Library ที่เขียนขึ้นมีฟังก์ชันต่างๆ ดังนี้

- ESP_8266 esp(long baud) ใช้ในการกำหนดความเร็วในการรับ/ส่งข้อมูลระหว่าง Arduino และ ESP8266
 - esp.restartESP() ใช้ในการ Restart ตัวโมดูล
 - esp.firmwareESP() ใช้สำหรับตรวจสอบ Firmware ของโมดูล ESP8226
 - esp.listAP() ใช้สำหรับสั่งให้โมดูล ESP8226 แสดงรายชื่อ Access Point ที่สแกนได้
 - esp.connectAP (String SSID, String PASS) ใช้สำหรับเชื่อมต่อ ESP8266 เข้ากับ Access Point โดยกำหนดชื่อของ SSID และ Password
 - esp.disconnectAP() ใช้สำหรับหยุดเชื่อมต่อ ESP8266 เข้ากับ Access Point
 - esp.esplP() ใช้สำหรับแสดงค่า IP ของโมดูล ESP8266
 - esp.setmode(String Mode) ใช้สำหรับเลือกโหมดการทำงานให้กับ โมดูล ESP8226 โดยสามารถกำหนดได้ 3 โหมดด้วยกันคือ STA, AP, BOTH
 - esp.connectTCP(String TCP) ใช้สำหรับตั้งค่าการเชื่อมต่อ TCP/UDP โมดูล ESP8226 โดยสามารถกำหนดได้ 2 โหมดด้วยกันคือ Single และ Multiple
 - esp.startClient(String TYPE, String ADDR, String PORT) ใช้สำหรับกำหนดค่าเริ่มต้นการใช้งาน Client Mode ในกรณีที่กำหนด esp.connectTCP(String TCP) เป็น Single
 - esp.startServer(String STATUS, String PORT) ใช้สำหรับกำหนดค่าเริ่มต้นการใช้งาน Server Mode
 - esp.disconnectSV() ใช้สำหรับหยุดการเชื่อมต่อในกรณี TCP และ UDP
 - esp.sendDT(String DATA) ใช้สำหรับส่ง String ออกไปทาง TCP และ UDP ในกรณีที่กำหนด
 - esp.connectTCP(String TCP) เป็นการสื่อสารแบบ SINGLE โดยจะใช้งานร่วมกับ esp.startSendDATA (int length) ซึ่งจะสามารถกำหนดความยาวของข้อมูลที่จะส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ESP_8266_AT_CMD_SET - Notepad
File Edit Format View Help
#ifndef ESP_8266_AT_CMD_SET_H
#define ESP_8266_AT_CMD_SET_H

#define WORKING          "AT"
#define RESTART          "AT+RST"
#define FIRMWARE_VER    "AT+GMR"
#define CHECK_IP        "AT+CIFSR"
#define LIST_ACCESS_POINT "AT+CWLAP"
#define JOINT_ACCESS_POINT "AT+CWJAP=\""
#define QUIT_ACCESS_POINT "AT+CWQAP"
#define GET_IP_ADD      "AT+CIFSR"
#define SET_PARAM_AP    "AT+CWSAP="
#define WIFI_MODE       "AT+CWMODE="
#define STA              1
#define AP               2
#define BOTH             3
#define TCP_CONNECTIONS "AT+CIPMUX="
#define SINGLE           0
#define MULTIPLE         1
#define SET_UP_TCP       "AT+CIPSTART="
#define SEND_DATA        "AT+CIPSEND="
#define SET_AS_SERVER    "AT+CIPSERVER="
#define CLOSE            0
#define OPEN             1

#endif

```

รูปที่ 3.18 หน้าต่าง ESP_8266 Library ในส่วนของ AT Command

```

File Edit Format View Help
#ifndef ESP_8266_h
#define ESP_8266_h
#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

#include "ESP_8266_AT_CMD_SET.h"

class ESP_8266
{
public :
    ESP_8266();
    ESP_8266(long baud);
    ~ESP_8266();
    void connectAP(String SSID,String PASS);
    void disconnectAP(String SSID,String PASS);
    void checkIP();
    void setmode(String MODE);
    void connectTCP(String TCP);
    void connectSV (String TYPE,String ADDR,String PORT);
    void sendDT(String DATA);
    void startSendDATA(int length);
    void startServer(String STATUS,String PORT);
};
#endif

```

รูปที่ 3.19 หน้าต่าง ESP_8266 Library ในส่วนของ Header File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "ESP_8266.h"
ESP_8266::ESP_8266() {}
ESP_8266::ESP_8266(long baud)
{
    Serial1.begin(baud);
    Serial.begin(baud);
}
ESP_8266::~~ESP_8266()
{}

void ESP_8266::connectAP(String SSID,String PASS)
{
    Serial.println("Connecting to access point .....");
    String CMD = JOINT_ACCESS_POINT ;
    CMD += SSID ;
    CMD += "\",\"" ;
    CMD += PASS ;
    CMD += "\"\r\n" ;
    Serial1.print(CMD);
    delay(5000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
    Serial.flush();
    /*char response[3] = Serial1.read();
    if(response == "OK")
    {
        return true;
    }
}

```

รูปที่ 3.20 หน้าต่าง ESP_8266 Library ในส่วนของ Source File

3.7 การทดลองเขียนโปรแกรมส่งค่าผ่าน Wi-Fi Module ESP8266

1. ต่อกองจรตามรูปที่ 3.15
2. โปรแกรมจะแบ่งออกเป็น 3 ส่วน ดังนี้

2.1 ส่วนการกำหนดรูปแบบการทำงานของ Wi-Fi Module ESP8266 กำหนดเป็น Client โดยนำ Function ที่เขียนใน ESP8266 Library มาใช้ ซึ่งจะกำหนด IP Address และ Port ให้ตรงกับ Wi-Fi Module ESP8266 ที่เป็น Server โดย Wi-Fi Module ESP8266 ทั้งสองฝั่งต้องเชื่อมต่อ Access Point เดียวกัน จึงจะสามารถเชื่อมต่อกันได้

```

sketch_may14a | A
File Edit Sketch Tools Help
sketch_may14a$
String ssid = "ECC_208" ;
String pass = "profinet" ;
String type = "TCP" ;
String addr = "192.168.1.10" ;
String port = "23" ;
String DATA = "" ;
ESP_8266 esp ;
void rpm ()
{
  NbTopsFan++;
}

void setup() {
  ESP_8266 esp(115200);
  /*Serial.begin(115200);
  Serial1.begin(115200);*/
}

sketch_may14a | A
File Edit Sketch Tools Help
sketch_may14a$
void setup()
{
  ESP_8266 esp(115200);
  esp.connectAP(ssid,pass);
  esp.checkIP();
  esp.setmode("1");
  esp.connectTCP("1");
  esp.connectSV("TCP",addr,"23");
  esp.startServer (Status,port);
  //pinMode (A5,OUTPUT);
  // pinMode (RPWM, OUTPUT);
  pinMode (PWM, OUTPUT);
}

void loop() {

```

รูปที่ 3.21 การกำหนดรูปแบบการทำงานของ Wi-Fi Module ESP8266 ให้เป็น Client

2.2 ส่วนการนับรอบการหมุนใบพัดของ Flow Sensor เพื่อนำค่าที่ได้มาคำนวณหา
ค่า Flow Rate ส่วนแสดงค่า Flow Rate และค่าที่ map เป็นแรงดันออกมาทาง Serial Monitor

```

pinMode(hallsensor, INPUT);
pinMode (RPWM, OUTPUT);

Serial.begin(115200);
attachInterrupt(0, rpm, RISING);
}
const uint32_t VREF_mV = 5000;

void loop() {
  analogWrite (RPWM, 114.75);
  NbTopsFan = 0;
  sei();
  delay (900);
  cli();
  Calc = (NbTopsFan/ 7.5);

  Serial.print("flowrate = ");
  Serial.print(Calc, DEC);
  Serial.print (" L/min\r\n");
  outputValue = map(Calc, 0, 16, 0, 100);
  output2Value = map(outputValue, 0, 100, 0, 5000);
  Serial.print("\toutput = ");
  Serial.println(outputValue);
  Serial.print("\toutput2 = ");
  Serial.println(output2Value);
}

```

รูปที่ 3.22 ส่วนการนับรอบการหมุนใบพัดของ Flow Sensor และส่วนแสดงค่า Flow Rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนส่งค่าแรงดันของ Flow Sensor

```

sketch_apr02a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_apr02a$
Serial.print("flowrate = " );
Serial.print(Calc, DEC); //Prints the number calc
Serial.print (" L/min\r\n"); //Prints "L/min" and
outputValue = map(Calc, 0, 16, 0, 100);
output2Value = map(outputValue,0,100,0,5000);
Serial.print("\t%output = ");
Serial.println(outputValue);
Serial.print("\t%output2 = ");
Serial.println(output2Value);

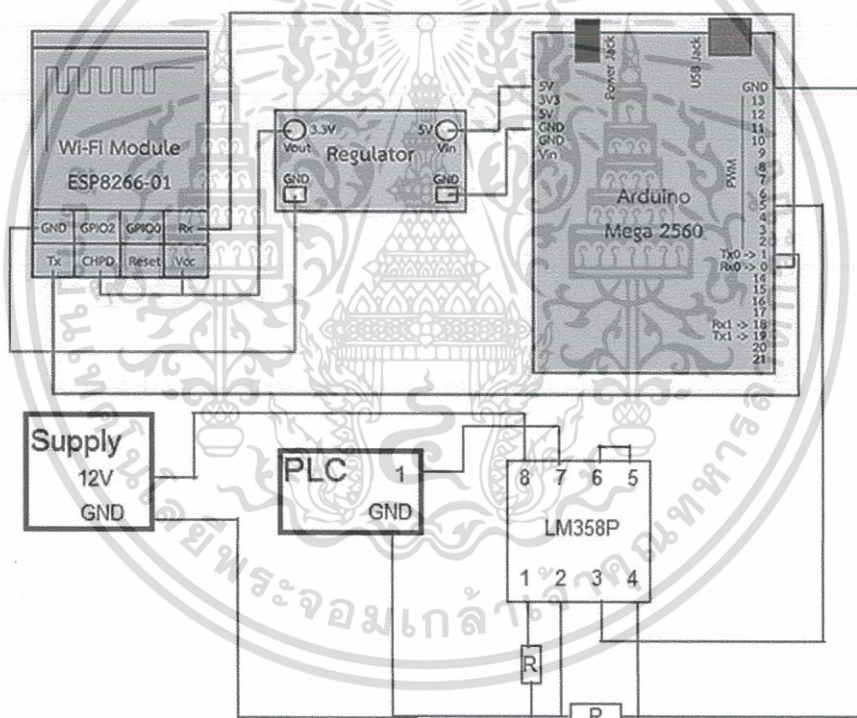
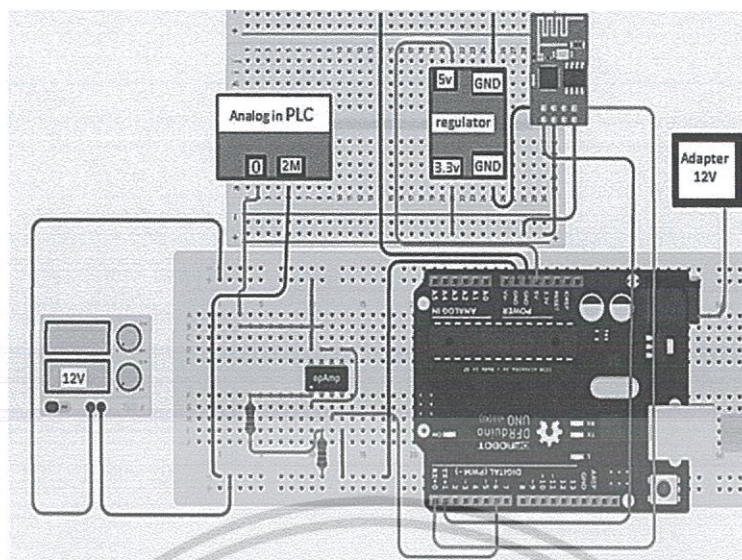
DATA = "f1" ;
DATA += (String)output2Value ;
//DATA = ("\n") ;
DATA += "e" ;
DATA.concat("\r\n") ;
esp.startSendDATA(7);
esp.sendDT(DATA);
}

```

รูปที่ 3.23 ส่วนส่งค่าแรงดันของ Flow Sensor

3.8 การทดลองเขียนโปรแกรมรับค่าจาก Wi-Fi Module ESP8266

1. ต่อบอร์ด Wi-Fi กับวงจร Buffer เข้า PLC โดยวงจร Buffer ต่อเพื่อขยายแรงดันจากบอร์ด Arduino จากช่วง 0-5 V ให้เป็น 0-10 V เพื่อให้ PLC อ่านค่าได้โดยเกิดความคลาดเคลื่อนน้อย ดังรูปที่ 3.24

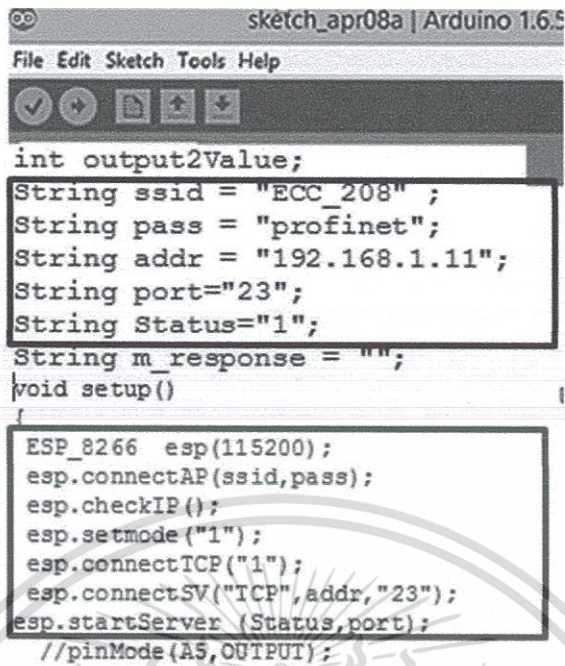


รูปที่ 3.24 การต่อวงจร Wi-Fi กับวงจร Buffer เข้า PLC

2. โปรแกรมจะแบ่งออกเป็น 3 ส่วน ดังนี้

2.1 นำ Function ที่เขียนใน ESP8266 Library มาใช้ กำหนดรูปแบบการทำงานของ Wi-Fi Module ESP8266 ให้เป็น Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



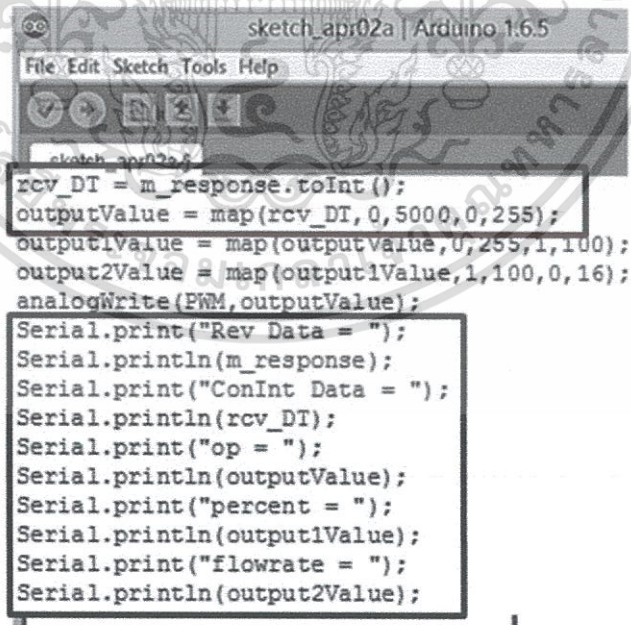
```

sketch_apr08a | Arduino 1.6.5
File Edit Sketch Tools Help
int output2Value;
String ssid = "ECC_208" ;
String pass = "profinet";
String addr = "192.168.1.11";
String port="23";
String Status="1";
String m_response = "";
void setup()
{
  ESP_8266 esp(115200);
  esp.connectAP(ssid,pass);
  esp.checkIP();
  esp.setmode("1");
  esp.connectTCP("1");
  esp.connectSV("TCP",addr,"23");
  esp.startServer (Status,port);
  //pinMode (A5, OUTPUT);

```

รูปที่ 3.25 กำหนดรูปแบบการทำงานของ Wi-Fi Module ESP8266 ให้เป็น Server

2.2 ส่วนรับค่าและแสดงค่าแรงดัน ของ Flow Sensor โดยนำค่าที่ได้มาเก็บไว้ในตัวแปร m_response จากนั้นทำการแปลงค่าจากแรงดันเป็น 0-255 โดยใช้คำสั่ง map ซึ่งเป็นช่วงของ PWM ที่สามารถส่งเข้า PLC ได้



```

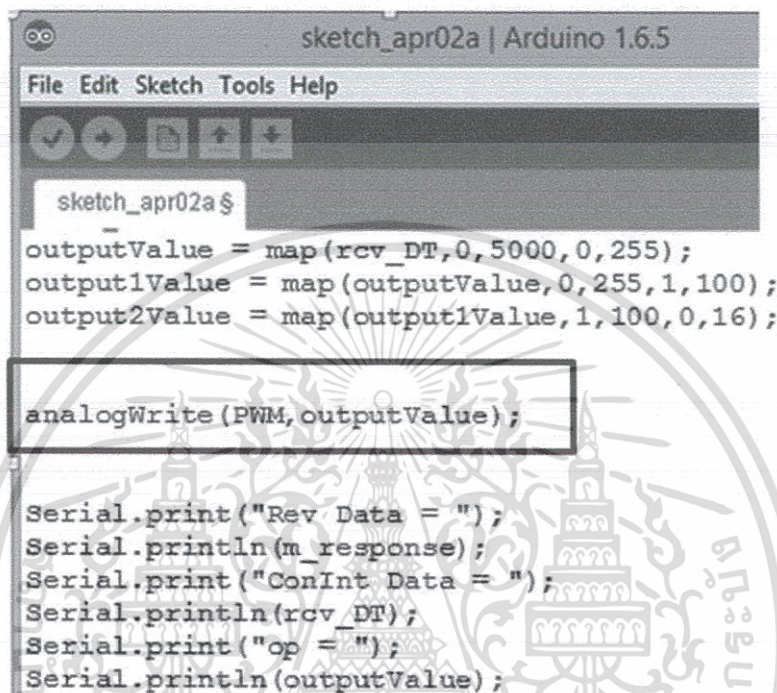
sketch_apr02a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_apr02a
rcv_DT = m_response.toInt();
outputValue = map(rcv_DT, 0, 5000, 0, 255);
output1Value = map(outputValue, 0, 255, 1, 100);
output2Value = map(output1Value, 1, 100, 0, 16);
analogWrite (PWM, outputValue);
Serial.print("Rev Data = ");
Serial.println(m_response);
Serial.print("ConInt Data = ");
Serial.println(rcv_DT);
Serial.print("op = ");
Serial.println(outputValue);
Serial.print("percent = ");
Serial.println(output1Value);
Serial.print("flowrate = ");
Serial.println(output2Value);

```

รูปที่ 3.26 กำหนดรูปแบบการทำงานของ Wi-Fi Module ESP8266 ให้เป็น Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนส่งค่าโวลต์ โดยทำการกำหนดให้ขา PWM ที่บอร์ด Arduino เป็นขาเอาต์พุต โดยสัญญาณ Digital ที่อยู่ในขา PWM จะถูกแปลงเป็นสัญญาณ Analog หลังจากนั้นค่าแรงดันจะถูกส่งเข้า Buffer เพื่อขยายแรงดันเข้า PLC และเมื่อวัดค่าแรงดันที่ PLC จะพบว่าเท่ากับค่าแรงดันที่หน้า Serial Monitor พอดี



```

sketch_apr02a | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_apr02a $
outputValue = map(rcv_DT,0,5000,0,255);
output1Value = map(outputValue,0,255,1,100);
output2Value = map(output1Value,1,100,0,16);

analogWrite(PWM, outputValue);

Serial.print("Rev Data = ");
Serial.println(m_response);
Serial.print("ConInt Data = ");
Serial.println(rcv_DT);
Serial.print("op = ");
Serial.println(outputValue);

```

รูปที่ 3.27 ส่วนส่งค่าแรงดัน โดยกำหนดให้ขา PWM ที่บอร์ด Arduino เป็นขาเอาต์พุต

บทที่ 4

ผลการทดลอง

4.1 การทดลอง Flow Sensor เพื่อหาค่า Flow Rate ของระบบ

ทดลองหาค่าความคลาดเคลื่อนระหว่างค่าอัตราการไหลที่ได้จากการเขียนโปรแกรมเปรียบเทียบกับการวัดอัตราการไหลจริง



รูปที่ 4.1 การเชื่อมต่ออุปกรณ์เพื่อทดลอง

4.1.1 ผลการทดลอง

ตารางที่ 4.1 ค่า Flow Rate จากการวัดและจากหน้า Serial Monitor โดยจับเวลา 6 วินาที

ครั้งที่	ปริมาณน้ำที่วัดได้ (L)	เวลา (s)	Flow Rate (วัด) L/min	ค่าผิดพลาดจากการวัด	Flow Rate (monitor) L/min	ค่าผิดพลาดจาก monitor
1	1.265	6	12.65	0.47	12-13	1.45
2	1.25	6	12.5	0.32	11	0.55
3	1.275	6	12.75	0.57	12	0.45
4	1.1	6	11	1.18	11	0.55
5	1.14	6	11.4	0.78	11	0.55
6	1.195	6	11.95	0.23	11-12	0.55
7	1.2	6	12	0.18	11-12	0.45
8	1.275	6	12.75	0.57	11-12	0.05
9	1.13	6	11.3	0.88	11-12	0.55
10	1.15	6	13.5	1.32	12-13	0.45
เฉลี่ย			12.18	0.65	11.55	0.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

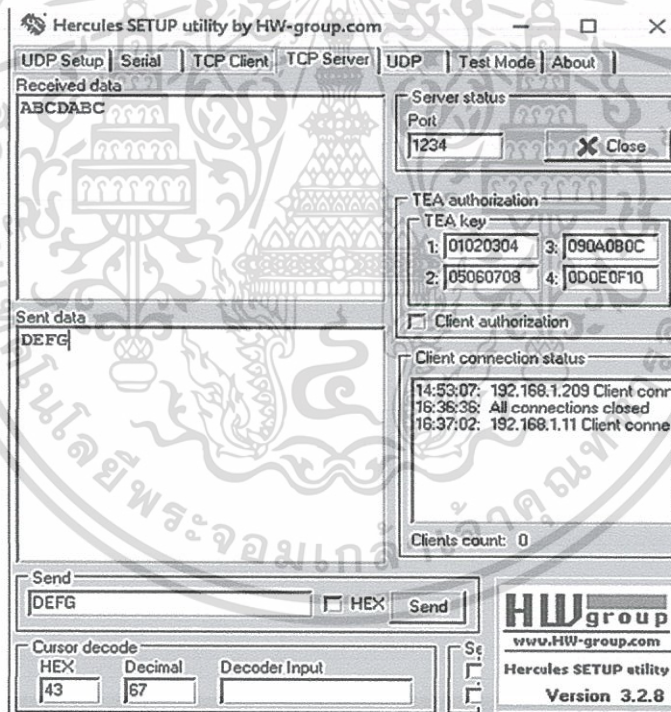
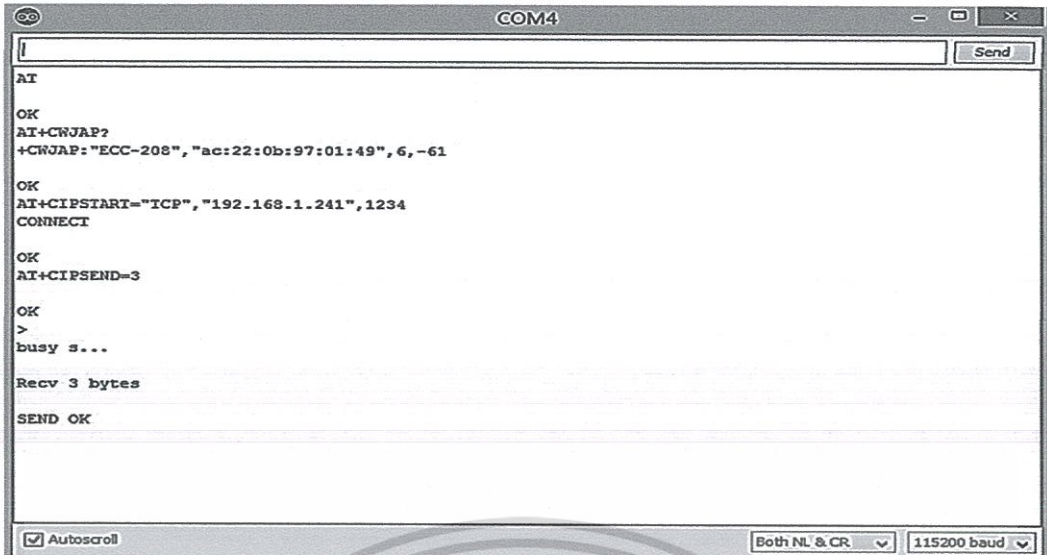
ตารางที่ 4.2 ค่า Flow Rate จากการวัดและจากหน้า Serial Monitor โดยจับเวลา 1 นาที

ครั้งที่	ปริมาณน้ำที่วัดได้ (L)	เวลา (min)	Flow Rate (วัด) L/min	ค่าผิดพลาดจากการวัด	Flow Rate (monitor) L/min	ค่าผิดพลาดจาก monitor
1	11.445	1	11.445	0.069	11-12	0
2	11.455	1	11.455	0.079	12	0
3	11.54	1	11.54	0.164	12-13	0
4	11.355	1	11.355	0.021	12	0
5	11.2	1	11.2	0.176	12	0
6	11.51	1	11.51	0.134	12	0
7	11.25	1	11.25	0.126	12	0
8	11.285	1	11.285	0.091	12	0
9	11.3	1	11.3	0.076	12	0
10	11.42	1	11.42	0.044	12	0
เฉลี่ย			11.376	0.098	12	0

จากการทดลองหาค่า Flow Rate ด้วย Flow Sensor และโปรแกรม Arduino พบว่าค่าที่ได้ในแต่ละครั้งมีความใกล้เคียงกัน และค่าผิดพลาดที่ได้มีค่าน้อย (ค่าผิดพลาดส่วนใหญ่ที่เกิดขึ้น เกิดจากความผิดพลาดในการวัดปริมาณน้ำและการจับเวลา) แต่อย่างไรก็ตาม เมื่อส่งค่า Flow Rate ไปให้ PLC จะต้องส่งเป็นค่าเปอร์เซ็นต์ Flow Rate และบอกช่วงค่าผิดพลาดเพื่อความสะดวกในการเขียนโปรแกรมควบคุมต่อไป

4.2. การทดลองเชื่อมต่อโมดูล Wi-Fi กับโปรแกรม Hercules ในคอมพิวเตอร์

จากการทดลองโมดูล WiFi พบว่าเมื่อเชื่อมต่อคอมพิวเตอร์และตัวโมดูล Wi-Fi กับเราเตอร์ เราเตอร์จะส่งค่า IP Address ให้คอมพิวเตอร์และโมดูล WiFi ESP 8266 หลังจากนั้นเมื่อพิมพ์ AT+CIPSTART = "TCP", "IP Address ของคอมพิวเตอร์", port ของโปรแกรม Hercules เช่น AT+CIPSTART = "TCP", "192.168.1.244", 6000 ที่หน้าต่าง Arduino Serial Monitor เพื่อเชื่อมต่อคอมพิวเตอร์กับตัว Wi-Fi โดยตรง (ในโครงการนี้กำหนดให้ตัวโมดูล Wi-Fi เป็น Client และคอมพิวเตอร์เป็นตัว Server) แล้วจึงตรวจสอบการเชื่อมต่อโดยส่งข้อความไป-กลับระหว่างโปรแกรม Hercules (โปรแกรมจำลองเป็น Server) และ Serial Monitor



รูปที่ 4.2 ผลจากหน้าจอ Serial Monitor และ โปรแกรม Hercules

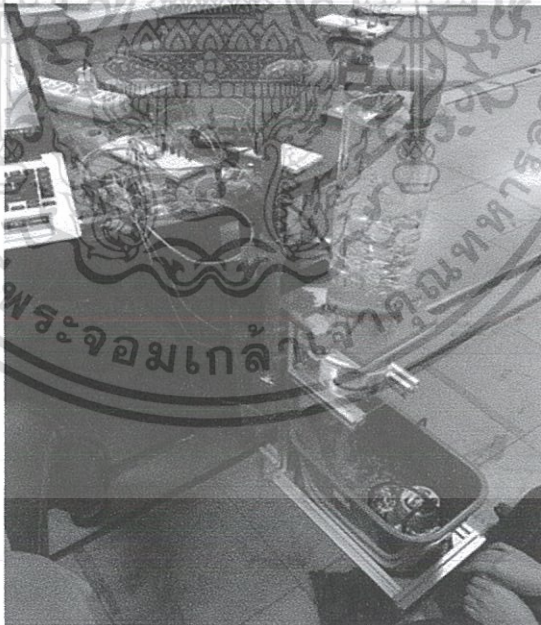
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองหาค่าแรงดันเอาต์พุตของ Flow Sensor เทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ

ทำการทดลองต่อ Flow Sensor เข้ากับระบบจริงโดยให้ผลลัพธ์ออกมาเป็นค่า Flow Rate (L/min), ค่าเปอร์เซ็นต์ Flow Rate และค่าแรงดันเอาต์พุตจาก Flow Sensor ในหน่วย mV โดยให้น้ำไหลผ่าน Flow Sensor ด้วยความเร็วที่แตกต่างกัน เพื่อเทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ

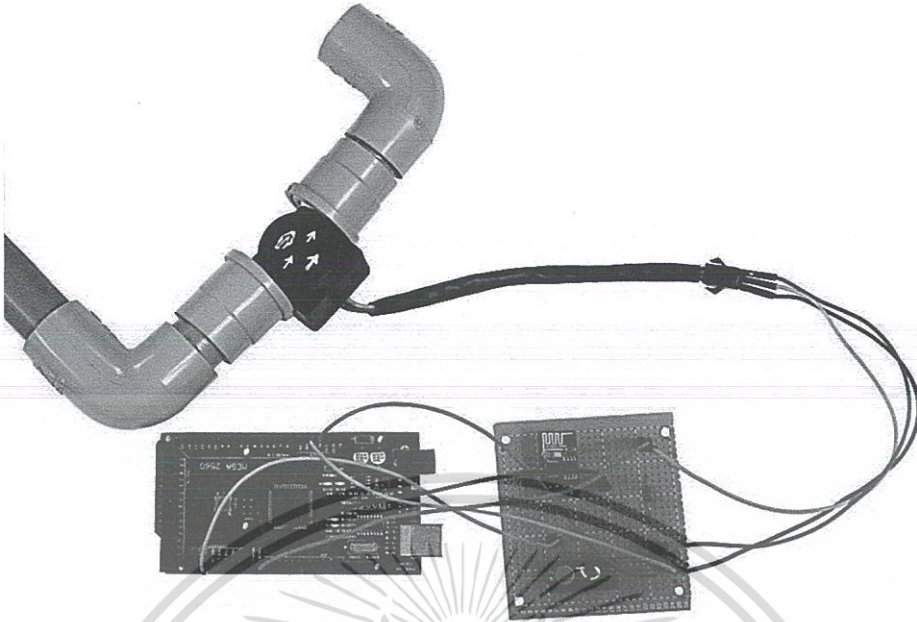


รูปที่ 4.3 ระบบรวมที่ใช้ในการทดลอง



รูปที่ 4.4 ส่วนของระบบถังน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ส่วนของการต่อวงจร

```

sketch_may12a | Arduino
File Edit Sketch Tools Help
sketch_may12a
// The setup() method runs once, when the sketch starts
void setup() //
{
  pinMode(hallsensor, INPUT); //initializes digital pin 2 as an input
  Serial.begin(115200); //This is the setup function where the serial port is
//initialized,
  attachInterrupt(0, rpm, RISING); //and the interrupt is attached
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop ()
{
  int outputValue;
  NbTopsFan = 0; //Set NbTops to 0 ready for calculations
  sei(); //Enables interrupts
  delay(1000); //Wait 1 second
  cli(); //Disable interrupts
  Calc = (NbTopsFan/ 7.5); // (pulse frequency) / 7.5q. = flow rate
  outputValue = map(Calc, 0, 13, 0, 100);
  Serial.print("flowrate = ");
  Serial.print(Calc, DEC); //Prints the number calculated above
  Serial.print(" L/min\r\n"); //Prints "L/min" and returns a new line
  Serial.print("\t\toutput = ");
  Serial.println(outputValue);
}

Serial Monitor
sketch uses 5,052 bytes (14% of program storage space. Maximum is 263,952 bytes.
Global variables use 256 bytes (3% of dynamic memory, leaving 7,936 bytes for local va
Autoscroll Both NL & CR 115200 baud

```

รูปที่ 4.6 หน้าจอโปรแกรม Arduino และผลที่แสดงผ่านหน้า Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

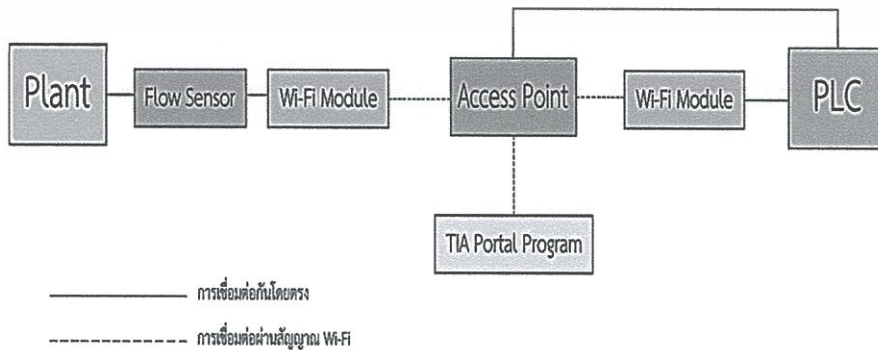
ผลการทดลองหาค่าแรงดันเอาต์พุตของ Flow Sensor เทียบกับเปอร์เซ็นต์การทำงานของปั๊มน้ำ

ตารางที่ 4.3 ผลการทดลองหาค่าแรงดันเอาต์พุตของ Flow Sensor เทียบกับเปอร์เซ็นต์การทำงานของปั๊มน้ำ

เปอร์เซ็นต์การทำงานของ Motor ปั๊มน้ำ (%)	ค่า Flow Rate (L/min)	ค่าแรงดันเอาต์พุตของ Flow Sensor (mV)
45	3	900
50	5	1550
55	6	1850
60	8	2500
65	9	2800
70	10	3100
75	11	3400
80	12	3750
85	13	4050
90	14	4350
95	15	4650
100	16	5000

4.4 ผลการทดลอง ส่ง-รับสัญญาณ (ค่าแรงดัน) เข้า Module Wi-Fi แล้วส่งเข้า PLC และโปรแกรม TIA PORTAL

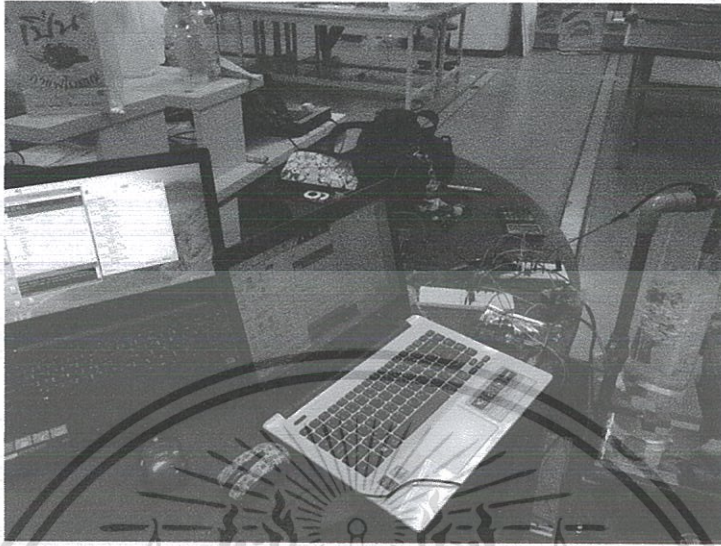
ทำการทดลองส่งค่าแรงดันที่ได้จาก Flow Sensor โดยการส่งค่าแรงดันนั้นจะต้องส่งผ่าน Module Wi-Fi 2 ตัว ซึ่งตัวแรกจะต่อกับบอร์ด Arduino ที่ติดกับ Flow Sensor ส่วนอีกตัวจะต่อกับบอร์ด Arduino ที่ติดกับ PLC และเป็นการส่ง-รับค่าแบบเรียลไทม์ หลังจากนั้นบอร์ด Arduino ที่ติดกับ PLC จึงจะทำการส่งค่าแรงดันที่ได้รับไปที่ PLC โดยผ่านขา PWM ของบอร์ด Arduino เข้าขา Analog in ของ PLC และสุดท้ายโปรแกรม TIA PORTAL จะนำค่าแรงดันที่ PLC ไปแสดงผลและใช้งานต่อไป ดังรูปที่ 4.7



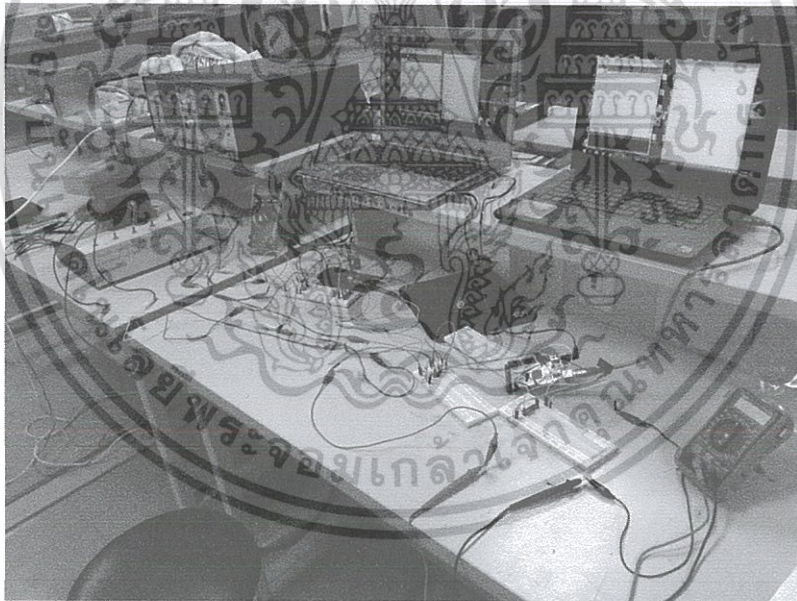
รูปที่ 4.7 การส่งสัญญาณจาก Flow Sensor ไป PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง ส่ง-รับ สัญญาณระหว่าง Module Wi-Fi กับ Module Wi-Fi แล้วส่งเข้า PLC และโปรแกรม TIA PORTAL



รูปที่ 4.8 ระบบด้านตัวส่งสัญญาณ

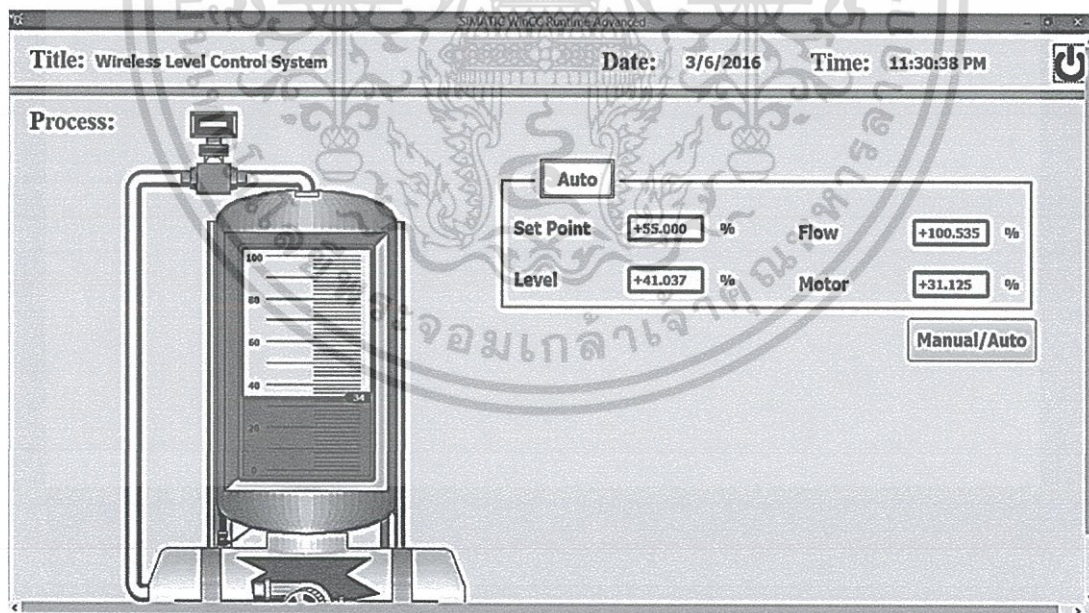


รูปที่ 4.9 ระบบด้านตัวรับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

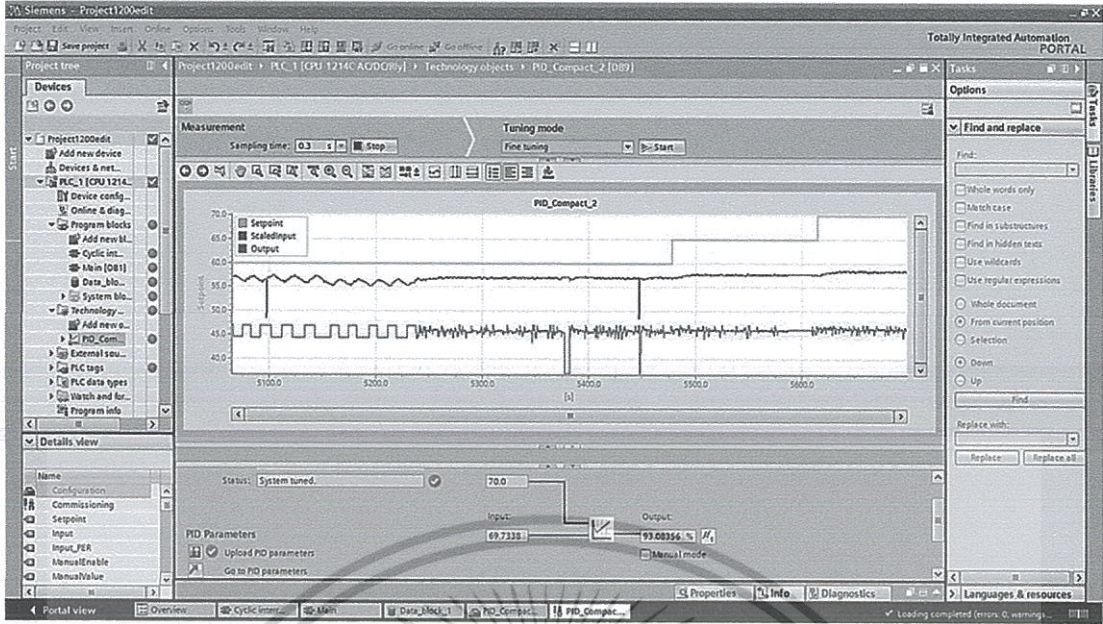
ผลการทดลองหาค่า Volt Output ที่ Module Wi-Fi ฝั่งรับ, ค่า Volt ที่ขา PWM ของบอร์ด Arduino และค่าแรงดันที่เข้า PLC เทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ ตารางที่ 4.4 ผลการทดลองหาค่าแรงดันที่ Module Wi-Fi ด้านตัวรับ, ค่าแรงดันที่ขา PWM ของบอร์ด Arduino และค่าแรงดันที่เข้า PLC เทียบกับเปอร์เซ็นต์การทำงานของปั้มน้ำ

เปอร์เซ็นต์การทำงานของ Motor ปั้มน้ำ (%)	ค่าแรงดันของ Flow Sensor (mV)	ค่าแรงดันที่ขา PWM (V)	ค่าแรงดันที่เข้า PLC (V)
45	1250	1.24	2.630
50	1850	1.85	3.836
55	2150	2.15	4.419
60	2500	2.50	5.120
65	2800	2.79	5.702
70	3100	3.11	6.32
75	3400	3.40	6.91
80	3750	3.76	7.60
85	4050	4.05	8.16
90	4350	4.35	8.70
95	4650	4.66	9.29
100	5000	5.02	10.00



รูปที่ 4.12 หน้าจอ HMI ของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 หน้าจอโปรแกรม TIA Portal

จากการทดลองพบว่าค่าแรงดันที่ได้จาก Flow Sensor (แสดงที่หน้าจอ Serial Monitor ของ Arduino ด้าน Client หรือด้านส่งสัญญาณ) ซึ่งมีค่าเท่ากับค่าแรงดันที่อ่านได้จากหน้าจอ Serial Monitor ของ Arduino ด้าน Server หรือด้านรับสัญญาณ, หน้า HMI ของ PLC และที่หน้าจอโปรแกรม TIA Portal ซึ่งเป็นไปตาม Code ที่ได้เขียนไว้ในโปรแกรม Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลสรุปและข้อเสนอแนะ

จากการดำเนินงานในส่วนของ Flow Transmitter ตามที่ได้กล่าวมาแล้วในบทข้างต้นพบว่า ระหว่างการทดลองและดำเนินการในขั้นตอนต่างๆ ได้พบเจอกับอุปสรรคและปัญหาเป็นระยะๆ จึงต้องอาศัยความรู้ ทักษะและการค้นคว้าจากหลายๆ ส่วนมาประกอบกัน โดยส่วนใหญ่ ใช้การค้นคว้า ข้อมูลจากอินเทอร์เน็ต ซึ่งเป็นวิธีที่ได้สะดวก รวดเร็วและหลากหลาย นอกจากนี้ยังมีคำแนะนำจาก อาจารย์ที่ปรึกษา ซึ่งอาจารย์ท่านได้ให้ความกรุณาในข้อมูลต่างๆ ที่จำเป็น คำแนะนำของรุ่นพี่ที่มีความรู้ รวมถึงคำแนะนำจากเพื่อนซึ่งเคยทำการทดลองประเภทนี้มาก่อน เพื่อเป็นแนวทางในการ แก้ไขปัญหาและอุปสรรคต่างๆ ที่พบเจอ

ในการทดลองส่วนของ Flow Transmitter นั้นมักจะพบปัญหาที่ตัว Wi-Fi Module เนื่องจากเป็นโมดูลที่มีราคาต่ำจึงไม่มีความเสถียรทำให้เกิดปัญหาในการเชื่อมต่อ หรือรับส่งค่าอยู่ เสมอๆ นอกจากนี้ยังคงพบปัญหาในส่วนรายละเอียดอื่นๆ มากมาย ตั้งแต่ปัญหาที่เกิดขึ้นทั้งก่อนการ ดำเนินงาน ระหว่างการดำเนินงาน และหลังการดำเนินงาน ซึ่งในส่วนของบทที่ 5 จะเป็นการกล่าวถึง ปัญหาต่างๆ การแก้ปัญหาต่างๆ รวมไปถึงข้อเสนอแนะที่เกิดขึ้นจากการทำโครงงานชิ้นนี้

5.1 สรุป

ในส่วนของ Flow Sensor นั้น เป็นการทดสอบว่าตัว Sensor สามารถทำงานได้ตามปกติ และสามารถใช้งานร่วมกับระบบจริงของโครงงานนี้ได้หรือไม่ โดยเป็นการทดสอบดูค่าแรงดันที่วัดได้จาก Flow Sensor แล้วแสดงผ่านหน้าจอ Serial Monitor ของโปรแกรม Arduino ตามค่า Speed Motor ที่สั่งในแต่ละครั้ง นำมาเปรียบเทียบกับค่าที่วัดได้จากมิเตอร์ พบว่าในทุกค่าที่นำมา เปรียบเทียบกันนั้น เป็นไปตามที่ได้คาดการณ์และคำนวณไว้ อีกทั้งยังมีแนวโน้มไปในทางเดียวกัน แต่ อาจมีความคลาดเคลื่อนบ้างเล็กน้อยเท่านั้น ทำให้สรุปได้ว่า Flow Sensor ที่เลือกใช้มีคุณสมบัติตรงกับผลการคำนวณทางทฤษฎี และสามารถใช้งานในระบบนี้ได้จริง

ในส่วนของการเขียนโปรแกรมส่งข้อมูลจะกำหนดให้ Wi-Fi Module เป็น Client โดยเริ่ม จากการเขียน Library ESP8266 เพื่อใช้สำหรับ Wi-Fi Module จากนั้นทำการเขียนโปรแกรมใน Arduino โดยนำเอา Function จาก Library ESP8266 ที่เขียนไว้มาใช้งาน โดยในช่วงแรกจะทำการ ทดสอบส่งค่าข้อมูลแรงดันเอาต์พุตที่ได้จาก Flow Sensor ไปให้โปรแกรม Hercules ซึ่งจำลองเป็น Server เพื่อรับค่าจาก Wi-Fi Client ซึ่งพบว่าตัว Wi-Fi สามารถส่งค่าออกไปให้กับ Server ได้ และ หลังจากเขียนโปรแกรมรับค่าให้กับ Wi-Fi Server เรียบร้อยแล้ว ได้ทำการเขียนโปรแกรมส่งค่าของ Wi-Fi Client เพิ่มเติมในบางส่วน เนื่องจากเปลี่ยนตัว Server จากโปรแกรม Hercules มาเป็น Wi-Fi เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Module จากนั้นทำการทดสอบส่งข้อมูลให้กับ Wi-Fi Server ซึ่งจากการทดสอบพบว่า Wi-Fi Module สามารถรับ – ส่งข้อมูลได้ตามที่ต้องการ

ในส่วนของการเขียนโปรแกรมเพื่อรับข้อมูลนั้นจะกำหนดให้ Wi-Fi Module เป็น Server เพื่อรับค่า ทำการเขียนโปรแกรม Arduino โดยนำ Function ใน Library ESP8266 ที่ได้เขียนไว้มาใช้งานเช่นเดียวกับการเขียนโปรแกรมเพื่อส่งข้อมูล แต่กำหนดให้เป็น Wi-Fi Server เพื่อรับค่าจาก Wi-Fi Client ซึ่งพบว่า Wi-Fi สามารถรับค่าที่ส่งมาได้ตามที่เขียนโปรแกรมไว้ จากนั้นจึงเขียนคำสั่งเพิ่มเติมกำหนดให้ขา PWM ที่บอร์ด Arduino เป็นขาเอาต์พุตส่งข้อมูลแรงดันที่รับจาก Wi-Fi Client ออกไปให้กับ PLC แล้วทำการวัดค่าแรงดันที่ขา PWM และที่ PLC เปรียบเทียบกับค่าที่แสดงใน Serial Monitor เพื่อตรวจสอบให้ค่าที่ได้ออกมานั้นมีค่าตรงกัน และโปรแกรมกับระบบที่ได้ดำเนินการมาสามารถใช้งานได้จริง ซึ่งพบว่าค่าแรงดันเอาต์พุตที่ได้เป็นไปตามที่ได้คำนวณไว้ และโปรแกรมสามารถรับค่าจาก Wi-Fi Client แล้วส่งค่าเอาต์พุตให้กับ PLC ได้ตามที่เขียนโปรแกรมไว้

5.2 ปัญหาและอุปสรรคในการดำเนินโครงการนี้

1. เนื่องจาก ขาดทักษะ ความรู้ ความชำนาญและประสบการณ์ที่จำเป็นต่อการทำโครงการนี้ ในหลายๆ ส่วนงาน ตั้งแต่การเลือกใช้อุปกรณ์ การต่อวงจร รวมไปถึงการเขียนโปรแกรมเพื่อควบคุม ส่วนต่างๆ จึงทำให้การทำงานมีปัญหาและล่าช้ากว่าที่ควรจะเป็น
2. อุปกรณ์ที่ใช้เกิดความเสียหายและไม่พร้อมใช้งาน เนื่องจากอุปกรณ์ที่ใช้ในโครงการนี้เป็นแบบประหยัด มีราคาถูก จึงทำให้อุปกรณ์มีความไม่เสถียร เปราะบาง และมีความเสี่ยงที่จะเสียหาย สูง
3. การเขียนโปรแกรมทั้งการส่งข้อมูล การรับข้อมูล และการเขียน Library ESP_8266 เกิดปัญหาในรายละเอียด และการใช้คำสั่งต่างๆ ในโปรแกรม Arduino อันเนื่องมาจากผู้จัดทำขาด ความรู้ความเข้าใจในการเขียนโปรแกรมเพื่อการใช้งานประเภทนี้
4. ในตอนส่งข้อมูลพบว่า การเชื่อมต่อ Wi-Fi Module ESP8266 เกิดปัญหาอยู่บ่อยครั้ง ทั้ง การเชื่อมต่อระหว่าง Wi-Fi Client กับ Wi-Fi Server เชื่อมต่อกันไม่พบ และหลุดการเชื่อมต่อ นอกจากนี้ ในตอนรับข้อมูลยังพบว่ามีการส่งค่าอื่นนอกเหนือจากที่ต้องการส่งมาด้วย จึงต้องเขียน คำสั่งเพิ่มเติมเพื่อตัดเอาค่าที่ไม่ต้องการออก
5. การต่อและบัดกรีวงจรมีความผิดพลาด อันเนื่องมาจากความไม่ชำนาญ ส่งผลให้วงจรที่ได้ ไม่เสถียร อุปกรณ์เสียหาย และการผลทดลองเกิดค่าความผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

1. ควรเตรียมอุปกรณ์สำรองไว้ใช้งาน เนื่องจากอุปกรณ์ที่ใช้ในโครงงานนี้มีต้นทุนไม่สูงนัก ทำให้อุปกรณ์ไม่เสถียร เพราะบาง และมีความเสี่ยงที่จะเสียหายสูง ซึ่งจะส่งผลกับการทำงานของระบบได้
2. ควรศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับสิ่งที่ต้องทำให้ครอบคลุมและชัดเจน โดยเฉพาะพื้นฐานการเขียนโปรแกรม Arduino นอกจากนี้ควรมีการวางแผนงานให้เป็นระบบระบบ เพื่อให้โครงงานเสร็จสิ้นตามกำหนดการที่ตั้งไว้
3. โครงงานนี้สามารถนำไปประยุกต์ใช้กับระบบอุตสาหกรรมต่างๆ ได้ แต่ควรใช้ Wi-Fi Module ที่มีเสถียรภาพมากกว่านี้ เพื่อป้องกันไม่ไห้ระบบเสียหาย ซึ่งโครงงานนี้จะช่วยให้เกิดความสะดวกรสบายในการดำเนินงาน และลดต้นทุนการผลิตได้เป็นอย่างดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Reading Water Flow Rate from a flow meter. เข้าถึงได้จาก: <http://forum.arduino.cc/index.php/topic,8548.0.html> (วันที่สืบค้นข้อมูล: 25 กันยายน 2558)
- [2] บทความ Arduino. เข้าถึงได้จาก: <http://www.arduinoall.com/article> (วันที่สืบค้นข้อมูล: 24 กันยายน 2558)
- [3] Water Flow Sensor. เข้าถึงได้จาก: http://www.seeedstudio.com/wiki/G1%2F4_Water_Flow_sensor (วันที่สืบค้นข้อมูล: 25 กันยายน 2558)
- [4] Hall Effect Sensor. เข้าถึงได้จาก: http://mcu56.learninginventions.org/page_id=258 (วันที่สืบค้นข้อมูล: 28 กันยายน 2558)
- [5] Regulator 4.5V-7V to 3.3V. เข้าถึงได้จาก: <http://www.arduinoall.com/product/12/4-5v-7v-to-3-3v-ams1117-3-3v-power-supply-module-ams-1117> (วันที่สืบค้นข้อมูล: 10 กันยายน 2558)
- [6] วิธีการใช้งาน Arduino ส่ง Data ผ่าน TCP ส่งผลไปยัง Server. เข้าถึงได้จาก: <http://www.thaieasyelec.com/article-wiki/review-product-article/wifi-data-logging-using-arduino.html> (วันที่สืบค้นข้อมูล: 10 ตุลาคม 2558)
- [7] LED Power supply switching 12V 5A 60W. เข้าถึงได้จาก: <http://www.thianthong.com/product/transformer-power-supply/led-power-supply-switching-12v-5a-60w> (วันที่สืบค้นข้อมูล: 20 มกราคม 2559)
- [8] How to use a pressure sensor with a Microcontroller. เข้าถึงได้จาก: <http://langster1980.blogspot.com/2014/11/how-to-use-pressure-sensor-with.html> (วันที่สืบค้นข้อมูล: 28 ตุลาคม 2558)
- [9] PLC Analog Input Module With Arduino. เข้าถึงได้จาก: <http://programplc.blogspot.com/2014/06/plc-analog-input-module-with-arduino.html> (วันที่สืบค้นข้อมูล: 20 ธันวาคม 2558)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Library สำหรับใช้งานฟังก์ชัน

ESP_8266 Library ส่วน AT Command

```
#ifndef ESP_8266_AT_CMD_SET_H
#define ESP_8266_AT_CMD_SET_H

#define WORKING "AT"
#define RESTART "AT+RST"
#define FIRMWARE_VER "AT+GMR"
#define CHECK_IP "AT+CIFSR"
#define LIST_ACCESS_POINT "AT+CWLAP"
#define JOINT_ACCESS_POINT "AT+CWJAP=Y"
#define QUIT_ACCESS_POINT "AT+CWOAP"
#define GET_IP_ADD "AT+CIFSR"
#define SET_PARAM_AP "AT+CWSAP="
#define WIFI_MODE "AT+CWMODE="
    #define STA 1
    #define AP 2
    #define BOTH 3
#define TCP_CONNECTIONS "AT+CIPMUX="
    #define SINGLE 0
    #define MULTIPLE 1
#define SET_UP_TCP "AT+CIPSTART="
#define SEND_DATA "AT+CIPSEND="
#define SET_AS_SERVER "AT+CIPSERVER="
    #define CLOSE 0
    #define OPEN 1

#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP_8266 Library ส่วน Header File

```

#ifndef ESP_8266_h
#define ESP_8266_h
#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

#include "ESP_8266_AT_CMD_SET.h"
class ESP_8266
{
public :
    ESP_8266();
    ESP_8266(long baud);
    ~ESP_8266();
    void connectAP(String SSID,String PASS);
    void disconnectAP(String SSID,String PASS);
    void checkIP();
    void setmode(String MODE);
    void connectTCP(String TCP);
    void connectSV (String TYPE,String ADDR,String PORT);
    void sendDT(String DATA);
    void startSendDATA(int length);
    void startServer(String STATUS,String PORT);

};
#endif

```

ESP_8266 Library ส่วน Source File

```

#include "ESP_8266.h"
ESP_8266::ESP_8266(){}
ESP_8266::ESP_8266(long baud)
{
    Serial1.begin(baud);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.begin(baud);
}
ESP_8266::~ESP_8266()
{}
void ESP_8266::connectAP(String SSID,String PASS)
{
    Serial.println("Connecting to access point .....");
    String CMD = JOINT_ACCESS_POINT ;
    CMD += SSID ;
    CMD += "\",\"";
    CMD += PASS ;
    CMD += "\r\n";
    Serial1.print(CMD);
    delay(5000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
    Serial.flush();
    /*char response[3] = Serial1.read();
    if(response == "OK")
    {
        return true;
    }
    else return false ;*/
}
void ESP_8266::checkIP()
{
    Serial.println("ESP IP is.....");
    String CMD = CHECK_IP;
    CMD += "\r\n";
    Serial1.print(CMD);
    delay(7000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.flush();
}

void ESP_8266::setmode(String MODE)
{
    Serial.println("Connecting to WIFI setmode.....");
    String CMD = WIFI_MODE + MODE;
    CMD += "\r\n" ;
    Serial1.print(CMD);
    delay(5000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
    Serial.flush();
}

void ESP_8266::connectTCP(String TCP)
{
    Serial.println("Connecting to TCP ....");
    String CMD = TCP_CONNECTIONS + TCP ;
    CMD += "\r\n" ;
    Serial1.print(CMD);
    delay(6000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
    Serial.flush();
}

void ESP_8266::connectSV(String TYPE,String ADDR,String PORT)
{
    String CMD = SET_UP_TCP + String("\n") + TYPE + String(",") + String("\n") + ADDR +
String(",") + PORT ;
    CMD += "\r\n";
    Serial1.print(CMD);
    delay(5000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    Serial1.flush();
    Serial.flush();
}
void ESP_8266::sendDT(String DATA)
{
    Serial1.flush();
    Serial.flush();
    Serial1.println(DATA);
}
void ESP_8266::startSendDATA(int length)
{
    Serial1.flush();
    Serial.flush();
    Serial.println("Send to SERVER.....");

    String CMD = SEND_DATA ;
    CMD.concat(length);
    CMD += "\r\n";

    Serial1.print(CMD);
    delay(30000);
}
void ESP_8266::startServer(String STATUS,String PORT)
{
    Serial.println("Setting as server.....");
    String CMD = SET_AS_SERVER + STATUS+String(",")+ PORT;
    CMD += "\r\n";
    Serial1.print (CMD);
    delay(5000);
    while (Serial1.available()) {
        Serial.write(Serial1.read());
    }
    Serial1.flush();
    Serial.flush();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมคำสั่งที่ใช้ในการทดสอบ

โปรแกรมทดสอบการทำงานของเซนเซอร์

```
// http://themakersworkbench.com http://thebestcasescenario.com http://seedstudio.com
volatile int NbTopsFan; //measuring the rising edges of the signal
int Calc;
int hallsensor = 2; //The pin location of the sensor
void rpm () //This is the function that the interrupt calls
{
  NbTopsFan++; //This function measures the rising and falling edge of the
//hall effect sensors signal
}
// The setup() method runs once, when the sketch starts
void setup() //
{
  pinMode(hallsensor, INPUT); //initializes digital pin 2 as an input
  Serial.begin(9600); //This is the setup function where the serial port is
//initialised,
  attachInterrupt(0, rpm, RISING); //and the interrupt is attached
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop ()
{
  int outputValue;
  NbTopsFan = 0; //Set NbTops to 0 ready for calculations
  sei(); //Enables interrupts
  delay (1000); //Wait 1 second
  cli(); //Disable interrupts
```

```

Calc = (NbTopsFan/ 7.5); //(Pulse frequency) / 7.5Q, = flow rate
//in L/min
outputValue = map(Calc, 0, 13, 0, 100);
Serial.print("flowrate = ");
Serial.print (Calc, DEC); //Prints the number calculated above
Serial.print (" L/min\n\n"); //Prints "L/min" and returns a new line
Serial.print("\t\t%output = ");
Serial.println(outputValue);
Serial.print ("PULSE = ");
Serial.print (NbTopsFan);
Serial.print ("\n\n");
}

```

โปรแกรมส่งค่าให้กับ Wi-Fi Module

```

#include <ESP_8266.h>
volatile int NbTopsFan;
int Calc;
float RPWM = 5 ;
int hallsensor = 2;
int outputValue;
int output2Value;
String ssid = "ECC_208" ;
String pass = "profinet" ;
String type = "TCP" ;
String addr = "192.168.1.68" ;
String port = "23" ;
String DATA = "" ;
ESP_8266 esp ;
void rpm ()
{
NbTopsFan++;
}
void setup() {
ESP_8266 esp(115200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*Serial.begin(115200);
Serial1.begin(115200);*/

//Serial.println(DATA.length());
esp.connectAP(ssid,pass);
esp.checkIP();
esp.setmode("1");
esp.connectTCP("0");
esp.connectSV(type,addr,port);

pinMode(hallsensor, INPUT);
pinMode(RPWM,OUTPUT);

Serial.begin(115200);
attachInterrupt(0, rpm, RISING);
}
const uint32_t VREF_mV = 5000; // Voltage reference in mV

void loop() {
  analogWrite(RPWM,114.75);
  NbTopsFan = 0;
  sei();
  delay (900);
  cli();
  Calc = (NbTopsFan/ 7.5);

  Serial.print("flowrate = ");
  Serial.print(Calc, DEC); //Prints the number calculated above
  Serial.print (" L/min\n\n"); //Prints "L/min" and returns a new line
  outputValue = map(Calc, 0, 16, 0, 100);
  output2Value = map(outputValue,0,100,0,5000);
  Serial.print("\t%output = ");
  Serial.println(outputValue);
  Serial.print("\t%output2 = ");
  Serial.println(output2Value);

  DATA = "f" ;
  DATA += (String)Calc ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//DATA = ("\n");
DATA += "e" ;
DATA.concat("\r\n");

esp.startSendDATA(7);
esp.sendDT(DATA);
}
```

โปรแกรมรับค่าให้กับ Wi-Fi Module

```
#include <ESP_8266.h>
#include <SoftwareSerial.h>
#define DEBUG true
//int RPWM = 5 ;
int PWM = 2;
//int analog = A5;
//int motorSpeedPWM = 0 ;
//int motorSpeedDUT;
int rcv_DT;
int last;
int outputValue;
int output1Value;
int output2Value;
String ssid = "ECC_208" ;
String pass = "profinet";
String addr = "192.168.1.11";
String port="23";
String Status="1";
String m_response = "";

void setup()
{
  ESP_8266 esp(115200);
  esp.connectAP(ssid,pass);
  esp.checkIP();
  esp.setmode("1");
  esp.connectTCP("1");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//esp.connectSV("TCP",addr,"23");
esp.startServer (Status,port);
//pinMode(A5,OUTPUT);
// pinMode(RPWM,OUTPUT);
pinMode(PWM,OUTPUT);

}

void loop() {

// analogWrite(RPWM,127);
m_response = "";
while(Serial1.available())
{
char c = Serial1.read();
// Serial.print(c);
//Serial.print(c);
//int c = Serial1.read();
//Serial.print(c);
//Serial.println(rcv_DT);
//last = map(rcv_DT, 0, 5000, 0, 255);
m_response += c;
//Serial.print(last);
}
int f_index = m_response.indexOf('f') + 2;
int e_index = m_response.indexOf('e');
m_response = m_response.substring(f_index,e_index);
rcv_DT = m_response.toInt();
outputValue = map(rcv_DT,0,16,0,255);
output1Value = map(rcv_DT,0,16,1,100);
output2Value = map(output1Value,1,100,0,16);
analogWrite(PWM,outputValue);

Serial.print("Rev Data = ");
Serial.println(m_response);
Serial.print("Flow_Rate = ");
Serial.println(rcv_DT);



```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print("Out_PWM = ");  
Serial.println(outputValue);  
Serial.print("Percent_FR = ");  
Serial.println(output1Value);  
//Serial.print("flowrate = ");  
//Serial.println(output2Value);  
Serial1.flush();  
Serial.flush() ;  
  
delay(2000);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wireless Level Control System

IC-047

Ms. Waratchaya Boonchui, Ms. Pataraporn Singhaphet, Ms. Thanisa Musikasopon, Ms. Narisara Leatjantuk, Ms. Chanida Sriprab, Ms. Jitraphorn Chusaeng, Mr. Chanathip Chokkanapitak, Mr. Worrathun Thamsuntree, Ms. Pannravee Porntthisarn, Ms. Phattrawadee Sritas, Ms. Pantip Panyasit, Ms. Ramrada Khunnaphap, Ms. Siwan Sasipaisith and Mr. Sarat Pongprasert

Advisor: Prof. Dr. Vanchai Riewruja, Asst. Prof. Thepjit Cheypoca, Asst. Prof. Dr. Wande Petchmaneelumka and Dr. Sirichai Thammarakwattana

Department of Instrumentation and Control Engineering, Faculty of Engineering

E-mail: 55011089@kmitd.ac.th, 55011046@kmitd.ac.th

Abstract

This project presents a design and implementation of a small plant for study and simulation of closed-loop control systems in the industrial work. Level control system is used for study in this project. Moreover, controlling with wireless is employed in the system. The project divides into two parts: wireless plant networks and controlling method.

First, the plant is built as a water level controlling plant. Transducers used in the model plant consist of two sensors and one motor functioned as flow and level transmitters and actuator. Arduino, a widely used microcontroller in engineering works, is employed to provide establish a communication between transducers and controller over Wi-Fi. As mentioned earlier, control system based on wireless connectivity can reduce signaling cable requirement, which is helpful in a big plant using wired system.

Second, PLC is used as a controller to control the level of water in the tank to the point where users need. There are various types of controller, however PID controller is selected as the plant control method. A PID controller is a control loop feedback mechanism commonly used in industrial control systems. A PID controller continuously calculates an error value as the difference between a desired setpoint and a measured process variable then attempts to minimize the error over time by adjustment of a control variable. TIA Portal V13 is used to program a control logic and PID controller to make the water reach to the setpoint of the wireless level control system spontaneously.

Results

The experimental result shows that the devices in the wireless system can communicate with each other over Wi-Fi by sending the data from pressure sensor to PLC then using TIA Portal to tune PID parameters while the setpoint of the system is 60%. The PID parameters are set as follow.

Proportional gain	=	3.506227E-1
Integral action time	=	14.78867 s
Derivative action time	=	3.986802 s
Derivative delay coefficient	=	0.1
Proportional action weighting	=	5.480434E-1
Derivative action weighting	=	0.0
Sampling time of PID algorithm	=	3.999916E-1 s

Afterward, PLC will send a data to control motor speed and pump the water and control it to the setpoint level. When the setpoint is changed to a difference value, the system still able to keep the stability at the request values.

Introduction

At present, most industrial factory prefer using PLC to control plant from remote location reducing the number of wiring used for connecting relays. Nevertheless, it still need some wiring to connect between some elements, such as sensors and actuators. SIEMENS has developed a client module to eliminate the problem by connect via wireless network instead of regular wiring. Due to the fact that client module has high price, so we use Wi-Fi module ESP8266-01 to control water level in the wireless level control system as it has lower cost. Wi-Fi module is used as a medium of a communication between PLC and components. The model plant is designed using SolidWorks and controlled with PLC SIEMENS model S7-1200 via Wi-Fi module ESP8266-01.

Conclusion

We simulated an inexpensive water level control system to replace the commonly used in the industrial one and be able to control it to reach the setpoint. By using TIA Portal to program a PLC and using Arduino to connect PLC with the devices over Wi-Fi module, the system is stable and able to be controlled as user desire. Which can be applied to a bigger plant in a control field of work.

Objectives

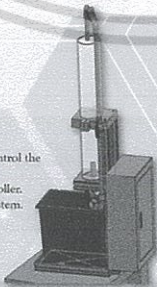
1. Study of the principle of microcontroller and the communication via Wi-Fi module.
2. Study of the usage of Arduino and TIA Portal v13.
3. Design a wireless control system.

References

- [1] Deitel, Harvey, M. and Deitel, Paul. (1997). C++ How to Program, 2nd Edition. Hall.
- [2] MaysPrentice arm43. "Protocol." [Online]. Available : <https://sites.google.com/site/maysarm43/phi-ngan-khsng-cha>
- [3] Electrocomponents "อุปกรณ์ PLC Siemens model S7-1214C AC/DC/Relay (6ES7 214-1BG31-0XB0)." [Online]. Available : <http://docs.europec.electrocomponents.com/webdocs/1129/0900766b81129785.pdf>
- [4] Paratrasnet. "อุปกรณ์ PLC Siemens model S7-1214C AC/DC/Relay (6ES7 214-1BG31-0XB0)." [Online]. Available : <http://www.paratrasnet.co/pdf/automatizari-industriale/S7-1200.pdf>
- [5] Sahkonumerot. "trans analog datasheet." [Online]. Available : <https://www.sahkonumerot.fi/2020/07/4/doc/rechusainitodoc/>
- [6] Arduino.cc. "simon code simon's Flow sensor." [Online]. Available : <http://forum.arduino.cc/index.php/topic/all.html>
- [7] Learninginventions. "Hall Effect Sensor." [Online]. Available : http://mcu4learning.inventions.org/?page_id=288

Methodology

1. Build a plant for study.
2. Study about sensor and actuator.
3. Design a circuit board for sensor and motor.
4. Create a library for Wi-Fi ESP8266-01.
5. Program the Arduino boards to send/ receive data and control the motor.
6. Make ladder program to control the plant using PID controller.
7. Make an interface for users for monitor and control the system.
8. Assemble all component of the plant together.
9. Tune PID controller and control the system via Wi-Fi.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้