

การศึกษาและสร้างหุ่นจำลองกระบวนการแลกเปลี่ยนความร้อนสำหรับ
ห้องปฏิบัติการพลวัตกระบวนการและการควบคุม

A study and implementation of heat exchanger process model for
PDC laboratory

นายปฏิพัทธ์	ทองเที่ยง
นายภูวดล	รอดพันธ์
นายรัชชานนท์	วงศ์หงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การศึกษาและสร้างหุ่นจำลองกระบวนการแลกเปลี่ยนความร้อนสำหรับ
ห้องปฏิบัติการพลวัตกระบวนการและการควบคุม
A study and implementation of heat exchanger process model for
PDC laboratory

นายปฏิพัทธ์	ทองเที่ยง
นายภูวดล	รอดพันธ์
นายรัชชานนท์	วงศ์หงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

A study and implementation of heat exchanger process model for
PDC laboratory

PATTIPAT THONGTHIANG

PUWADON RODPUN

RATCHANON WONGHONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2014

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

หัวข้อปริญญาโท การศึกษาและสร้างหุ่นจำลองกระบวนการแลกเปลี่ยนความร้อนสำหรับ
ห้องปฏิบัติการพลวัตกระบวนการและการควบคุม
A study and implementation of heat exchanger process model for
PDC laboratory

นักศึกษาผู้จัดทำ นายปฏิพัทธ์ ทองเที่ยง รหัสนักศึกษา 54010754
นายภูวดล รอดพัน รหัสนักศึกษา 54011014
นายรัชชานนท์ วงศ์หงษ์ รหัสนักศึกษา 54011071

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2557

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
รองศาสตราจารย์วิริยะ กองรัตน์	

ภาควิชารับรองแล้ว

(ผศ.ทวีพล ชื่อสัตย์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

หัวข้อปริญญานิพนธ์ การศึกษาและสร้างหุ่นจำลองกระบวนการแลกเปลี่ยนความร้อนสำหรับ
ห้องปฏิบัติการพลวัตกระบวนการและการควบคุม
A study and implementation of heat exchanger process model for
PDC laboratory

นักศึกษาผู้จัดทำ	นายปฏิพัทธ์ ทองเที่ยง	รหัสนักศึกษา	54010754
	นายภูวดล รอดพันธ์	รหัสนักศึกษา	54011014
	นายรัชชานนท์ วงศ์หงษ์	รหัสนักศึกษา	54011071
อาจารย์ที่ปรึกษา ปีการศึกษา	รศ.วิริยะ กองรัตน์		2557

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อศึกษาการแลกเปลี่ยนความร้อนของของไหล โดยจัดทำแบบจำลองขึ้นมาซึ่งมีอุปกรณ์หลักๆ ดังนี้ Heat exchanger, Heater, Temperature sensor(LM35), Turbine flow meter, pump, Microcontroller บอร์ด Arduino Nano V3.0 และใช้โปรแกรม Microsoft Visual Studio 2010 ในการเขียนโปรแกรมโดยใช้ร่วมกับบอร์ด Arduino เพื่อประมวลผลค่าที่วัดได้จากอุปกรณ์การวัด และนำมาแสดงผลในรูปแบบกราฟ โดยจะแสดงค่าอัตราการไหลของน้ำจาก Turbine flow meter และแสดงค่าของอุณหภูมิของน้ำจาก Temperature sensor(LM35) ทั้งท่อขาเข้าและขาออกจาก Heat exchanger โดยจะแสดงผลใน Computer ผ่านโปรแกรม Microsoft Visual Studio

Thesis Title	A study and implementation of heat exchanger process model for PDC laboratory		
Authors	PATTIPAT	THONGTHIANG	
	PUWADON	RODPUN	
	RATCHANON	WONGHONG	
Thesis Advisor	Assoc Prof.Viriya Kongrat		
Year	2014		

ABSTRACT

This project is designed to study heat exchange fluid. By modeling the devices, which are as follows: Heat exchanger, Heater, Temperature sensor (LM35), Turbine flow meter, pump, Microcontroller Board Arduino Nano V3.0 and using Microsoft Visual Studio 2010 programming with Arduino board to process the measured value of the measuring device and display values in graphic form. It shows the flow of water from turbine flow meter and displays water's temperature from the temperature sensor (LM35), both inbound and outbound from Heat exchanger will display in the Computer via Microsoft Visual Studio.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเนื่องจากได้รับความกรุณาอย่างสูงจากศาสตราจารย์ วิริยะ กองรัตน์ อาจารย์ที่ปรึกษางานวิจัย ที่ได้ให้คำแนะนำปรึกษาอีกทั้งยังเอื้อเพื่ออุปกรณ์ตลอดจนปรับปรุงแก้ไข ข้อบกพร่องต่างๆ ในการทำปริญญานิพนธ์นี้ ผู้วิจัยตระหนักถึงความตั้งใจจริงของอาจารย์และขอกราบ ขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

คณะผู้จัดทำขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้เอื้อเพื่อสถานที่ในการทำวิจัย อุปกรณ์ต่างๆและช่วยให้คำแนะนำต่างๆ จนทำให้งานวิจัยชิ้นนี้สำเร็จลุล่วงด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยช่วยเหลือและเป็นกำลังใจให้ ตลอดจนขอบคุณ เพื่อนๆทุกคนที่คอยให้ความช่วยเหลือและให้คำแนะนำดีๆ จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	1
1.3 ขอบเขตของปริญญาานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 การแลกเปลี่ยนความร้อน (Heat exchanger).....	3
2.1.1 เครื่องแลกเปลี่ยนความร้อน.....	4
2.1.2 การคำนวณ Energy balance และประสิทธิภาพ ของ Heat exchanger.....	5
2.1.3 Radiator.....	5
2.2 การติดต่อสื่อสารผ่านพอร์ตอนุกรม (Serial port).....	6
2.2.1 พอร์ตอนุกรม (Serial Port).....	6
2.2.2 มาตรฐาน RS232.....	6
2.2.2.1 อุปกรณ์รับส่งข้อมูลปลายทาง.....	7
2.2.2.2 อุปกรณ์สื่อสารข้อมูล.....	7
2.2.3 การใช้งานพอร์ตอนุกรม RS232.....	8

สารบัญ(ต่อ)

หน้า

2.2.3.1 การจัดขาของคอนเน็คเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ.....	8
2.2.3.2 ระดับสัญญาณของ RS232.....	9
2.2.3.3 รูปแบบการสื่อสารแบบอนุกรม.....	10
2.3 Microcontroller Board - Arduino Nano V3.0.....	11
2.4 เครื่องมือวัดการไหลแบบกังหัน (Turbine flow meter).....	12
2.4.1 สูตรคำนวณอัตราการไหลของ Turbine flow meter.....	13
2.5 Temperature sensor (LM35).....	14
2.6 โปรแกรม Arduino.....	15
2.6.1 การตั้งค่าโปรแกรม Arduino.....	15
2.7 โปรแกรม Microsoft visual studio.....	17
2.7.1 จุดเด่นของ Visual Basic.....	17
2.7.2 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic.....	18
2.7.3 การสร้างแอปพลิเคชันด้วย Visual Basic.....	19
บทที่ 3 การออกแบบและวิธีการดำเนินงาน.....	20
3.1 อุปกรณ์.....	20
3.2 ออกแบบชุดทดลอง.....	20
3.3 จัดทำชุดทดลอง.....	21
3.4 ออกแบบโปรแกรมสำหรับ Arduino ในการรับและส่งค่าจากอุปกรณ์การวัด.....	26
3.5 การสร้างโปรแกรม Plant Monitoring Arduino จากโปรแกรม Microsoft Visual Studio....	28
บทที่ 4 ผลการทดลอง.....	32

สารบัญ(ต่อ)

หน้า

4.1 การแสดงผลของโปรแกรม Arduino.....	32
4.1.1 การแสดงผลของการวัดค่าจากอุปกรณ์การวัดผ่าน Serial Monitor ของโปรแกรม Arduino.....	32
4.1.2 การแสดงผลของการวัดค่าจากอุปกรณ์การวัดผ่านโปรแกรม PUTTY.....	33
4.2 การแสดงผลของโปรแกรม Plant Monitoring Arduino.....	33
4.2.1 การแสดงผล Temperature, Flow Rate ในรูปแบบกราฟเส้นตรงในช่วงของอุณหภูมิ 30 – 40 °C	34
4.2.2 การแสดงผลของ Energy ในรูปแบบกราฟเส้นตรง.....	38
4.2.3 การแสดงผลของ Energy lost ในรูปแบบกราฟเส้นตรง.....	41
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	43
5.1 สรุปผลการทดลอง.....	43
5.2 ข้อเสนอแนะและแนวทางพัฒนา.....	43
บรรณานุกรม.....	44
ภาคผนวก.....	45

สารบัญรูป

รูปที่	หน้า
2.1 แสดงลักษณะของการนำความร้อน การพาความร้อนและการแผ่รังสี.....	3
2.2 แสดงลักษณะการไหลที่แตกต่างของเครื่องแลกเปลี่ยนความร้อนแบบท่อสองชั้น.....	4
2.3 แสดงการไหลที่แตกต่างกันใน Cross flow heat exchanger	4
2.4 แสดงลักษณะการไหลของเครื่องแลกเปลี่ยนความร้อนแบบเซลล์และท่อ.....	5
2.5 แสดงตัวอย่างหม้อน้ำที่ติดอยู่ด้านหน้ารถมอเตอร์ไซค์.....	6
2.6 แสดงพอร์ตอนุกรม RS232 และสายที่ใช้เชื่อมต่อ.....	7
2.7 แสดงการเชื่อมต่อด้วย RS232 ระหว่าง Host computer และ Terminal device.....	8
2.8 แสดงการเชื่อมต่อระหว่าง PC และ MCU Board ด้วย RS232.....	8
2.9 แสดงขาของคอนเน็คเตอร์แบบ DB9 ตัวผู้	8
2.10 แสดงระดับสัญญาณของ RS232C และระดับสัญญาณของ TTL	9
2.11 แสดงการสื่อสารแบบซิงโครนัส.....	10
2.12 แสดงการสื่อสารแบบอะซิงโครนัส.....	11
2.13 แสดงลักษณะของ Arduino Nano V3.0.....	11
2.14 แสดงขาต่างๆของ Board Arduino nano v3.0.....	12
2.15 แสดงโครงสร้างของเครื่องมือวัดอัตราการไหลแบบเทอร์ไบน์.....	13
2.16 แสดงลักษณะภายนอกและวงจรภายในของ LM35	14
2.17 แสดงหน้าต่างโปรแกรม Arduino	15
2.18 การตั้งค่า Board.....	15
2.19 การตั้งค่า Serial Port	16
2.20 การตั้งค่า Programmer.....	16
3.1 แสดงแบบจำลองของชุดทดลอง.....	20
3.2 แสดงการติดตั้ง Heat exchanger.....	21

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.3 แสดงการติดตั้ง Heater	22
3.4 แสดงการติดตั้ง Turbine flow meter.....	22
3.5 แสดงการติดตั้ง Radiator	23
3.6 แสดงจุดที่ใส่ซิลิโคน.....	23
3.7 แสดงการติดตั้ง Temperature sensor (Lm35).....	24
3.8 แสดงการติดตั้งปั๊มของแท็งก์น้ำเย็น.....	24
3.9 แสดงการติดตั้งปั๊มของแท็งก์น้ำร้อน.....	25
3.10 แสดงการติดตั้งแผงจ่ายไฟ.....	25
3.11 แสดงการติดตั้งสวิทช์ในการสั่งเปิด – ปิดปั๊ม ฮีตเตอร์และ Radiator.....	26
3.12 แสดงส่วนของการประกาศค่าตัวแปรและกำหนดค่าของ Arduino.....	26
3.13 แสดงส่วนของโปรแกรม Void setup.....	27
3.14 แสดงส่วนของการสร้าง loop.....	27
3.15 แสดงฟังก์ชัน Void loop.....	28
3.16 แสดงส่วนของการประกาศตัวแปร.....	28
3.17 แสดงส่วนของโปรแกรมที่ใช้ในการกำหนดสเกลของกราฟ.....	29
3.18 แสดงส่วนของโปรแกรมการเชื่อมต่อ.....	29
3.19 แสดงโปรแกรมในการเรียกอ่านข้อมูลจาก Buffer.....	30
3.20 แสดงโปรแกรมในการสร้างส่วนของการตัดการเชื่อมต่อ.....	30
3.21 แสดงโปรแกรม Timer.....	31
3.22 แสดงโปรแกรมการสลับไปมาของกราฟ.....	31
4.1 แสดงค่าที่วัดได้ผ่าน Serial Monitor จากโปรแกรม Arduino.....	32
4.2 การแสดงผลของการวัดผ่านโปรแกรม PUTTY.....	33

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.3 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน33.69°C..	34
4.4 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน34.67°C..	35
4.5 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน37.11°C..	36
4.6 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน38.09°C..	37
4.7 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน40.04°C..	38
4.8 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 31.74°C	39
4.9 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 37.11°C	39
4.10 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 38.09°C	40
4.11 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 40.53°C	40
4.12 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 31.74°C	41
4.13 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 33.20°C	41
4.14 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 40.04°C	42

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

โรงงานอุตสาหกรรมมีระบบในการทำงานที่ซับซ้อน เนื่องจากมีอุปกรณ์และเครื่องจักรหลายตัวในระบบจึงอาจทำให้มีปัญหาในการควบคุมที่ไม่ทั่วถึงและไม่สามารถทราบค่าจากอุปกรณ์วัดแบบ Real time ได้ แต่ในปัจจุบันได้มีเทคโนโลยีหลายอย่างที่สามารถแสดงค่าต่างๆให้แก่ผู้ควบคุมได้ทราบค่าจากอุปกรณ์การวัดได้ทันทีคือการใช้ Arduino รับค่าจากอุปกรณ์การวัดและแสดงค่าทางจอคอมพิวเตอร์โดยใช้โปรแกรม Microsoft Visual Studio 2010

Arduino คือ ไมโครคอนโทรลเลอร์ที่สามารถใช้เป็นเครื่องมือในการรับและส่งค่าจากอุปกรณ์อิเล็กทรอนิกส์หรืออุปกรณ์การวัดต่างๆ เพื่อนำไปควบคุมการทำงานของระบบและสามารถแสดงค่าที่ได้รับจากอุปกรณ์นั้นๆได้อีกด้วย

Microsoft Visual Studio เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นใช้งานที่ใช้ได้ตั้งแต่ระดับต้น เพื่อใช้สร้างโปรแกรมง่ายๆ บน Windows หรือโปรแกรมเมอร์ระดับกลางที่จะเรียกใช้ฟังก์ชันต่างๆได้อย่างมีประสิทธิภาพ ตลอดจนโปรแกรมเมอร์ระดับมืออาชีพที่จะพัฒนาโปรแกรมในระดับสูง

ภาษา Visual Basic เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมในการนำมาใช้งานพัฒนาโปรแกรม เนื่องจากสะดวกในการหยิบเครื่องมือที่โปรแกรมได้จัดเตรียมไว้ให้สำหรับออกแบบหน้าจอและสิ่งต่างๆ สำหรับในการเขียนโปรแกรมให้เรียบร้อย

ดังนั้นในโครงการนี้จะใช้ลักษณะการทำงานดังที่กล่าวมาข้างต้นในการแสดงผลการวัดอัตราการไหลและอุณหภูมิของของไหลภายในท่อโดยใช้ Arduino เป็นตัวประมวลผล และแสดงค่าต่างๆในรูปแบบกราฟโดยซอฟต์แวร์ที่เขียนจากโปรแกรม Microsoft Visual Studio 2010

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการวัดค่าจากอุปกรณ์การวัดและวิธีการรับส่งข้อมูลจากอุปกรณ์การวัดสู่ Arduino
2. ศึกษาหลักการเรียกอ่านข้อมูลจาก Arduino ผ่านทาง serial port
3. ศึกษาการใช้งานโปรแกรม Microsoft Visual Studio 2010

1.3 ขอบเขตของปริญญาโท

1. ศึกษาการวัดอัตราการไหลด้วย Turbine flow meter
2. ศึกษาการวัดอุณหภูมิด้วย Temperature sensor(LM35)
3. ศึกษาการ รับ-ส่ง ข้อมูลของ Arduino
4. ศึกษาการเขียนโปรแกรมแสดงผลด้วยโปรแกรม Microsoft Visual Studio 2010
5. เขียนโปรแกรมเพื่อเรียกอ่านข้อมูลที่ได้รับจาก Arduino และแสดงผลในรูปแบบกราฟเส้น

1.4 ขั้นตอนการศึกษา

โครงการขั้นนี้จะเริ่มโดยศึกษาทฤษฎีการแลกเปลี่ยนความร้อนของของไหล และศึกษาหลักการการทำงานของอุปกรณ์ที่จะนำมาติดตั้งในแบบจำลองประกอบด้วย Heat exchanger, Heater, Turbine flow meter, Temperature sensor(LM35), Radiator, Pump อีกทั้งยังศึกษาวิธีการแสดงผลในรูปแบบกราฟเส้น จากนั้นจึงจัดทำแบบจำลองขึ้นมา เพื่อจำลองการแลกเปลี่ยนความร้อนโดยจะมีการติดตั้ง Heater ในการเพิ่มอุณหภูมิให้กับของไหลภายในท่อและทำการวัดซึ่งจะติดตั้ง Turbine flow meter และ Temperature sensor (LM35) ที่ท่อทั้งก่อนและหลังที่น้ำจะผ่าน Heat exchanger ในการแลกเปลี่ยนอุณหภูมิ เพื่อวัดอัตราการไหลและอุณหภูมิของน้ำ จากนั้นนำไปประมวลผลด้วย Arduino เพื่อนำไปแสดงผลในรูปแบบกราฟเส้นโดยซอฟต์แวร์ที่เขียนจากโปรแกรม Microsoft Visual Studio 2010

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. นักศึกษาสามารถเรียนรู้ เข้าใจหลักการและประยุกต์ใช้ Arduino ได้
2. นักศึกษาสามารถสร้างการติดต่อสื่อสารระหว่างคอมพิวเตอร์และ Arduino ได้
3. สามารถนำองค์ความรู้ที่ได้ไปประกอบอาชีพต่อไปในอนาคตได้

บทที่ 2

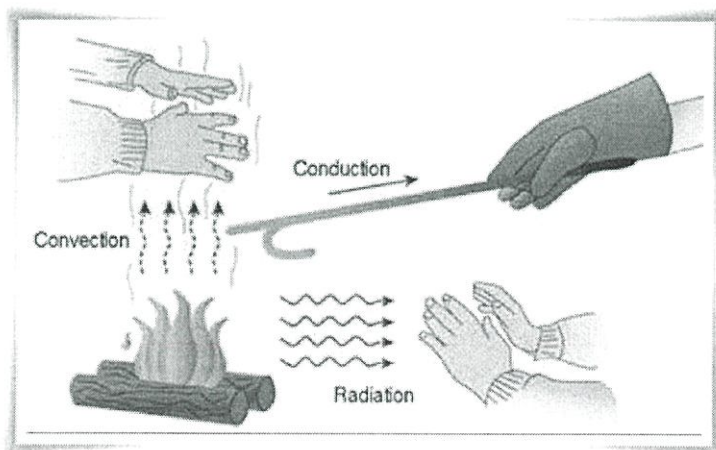
ทฤษฎีและหลักการ

2.1 การแลกเปลี่ยนความร้อน (Heat exchanger)

ความร้อน คือ พลังงานรูปหนึ่งที่มีอยู่ในสารทั้งหลาย โดยการที่สารหนึ่งจะถ่ายเทความร้อนให้กับอีกสารหนึ่งจะต้องมีความแตกต่างของอุณหภูมิระหว่างสารทั้งสอง เมื่อมีอุณหภูมิต่างกันจะมีการถ่ายเทความร้อนจากสารที่มีอุณหภูมิสูงกว่าไปสู่สารที่มีอุณหภูมิต่ำกว่า โดยการถ่ายเทความร้อนจะดำเนินไปอย่างต่อเนื่องจนสารทั้งสองที่อุณหภูมิเท่ากัน

การถ่ายเทความร้อน คือ การถ่ายเทพลังงานความร้อนระหว่าง 2 บริเวณที่มีอุณหภูมิแตกต่างกัน โดยแบ่งออกเป็น 3 ประเภท ดังนี้

1. การนำความร้อน (Heat Conduction) เป็นการถ่ายเทความร้อนภายในวัตถุที่เป็นของแข็งภายในวัตถุหนึ่งๆ หรือระหว่างวัตถุสองชิ้นที่สัมผัสกัน โดยมีทิศทางของการเคลื่อนที่ของพลังงานความร้อนจากบริเวณที่มีอุณหภูมิสูงไปยังบริเวณที่มีอุณหภูมิต่ำกว่า โดยที่ตัวกลางไม่มีการเคลื่อนที่ซึ่งวัตถุแต่ละชนิดจะมีประสิทธิภาพการนำความร้อนแตกต่างกัน โดยโลหะจะนำความร้อนได้ดีที่สุด
2. การพาความร้อน (Heat Convection) เป็นการถ่ายเทความร้อนในสภาวะก๊าซหรือของเหลว โดยเมื่อได้รับความร้อนจากแหล่งกำเนิดจะมีอุณหภูมิสูงและลอยตัวขึ้นตัวกลางจึงไหลเข้าไปแทนที่ เช่น เมื่อนำกาต้มน้ำไปวางบนเตา ความร้อนจากไฟจะค่อยๆเคลื่อนที่จากน้ำด้านล่างขึ้นมายังบริเวณผิวน้ำด้านบน
3. การแผ่รังสี (Radiation) เป็นการถ่ายเทความร้อนโดยไม่อาศัยตัวกลาง เช่น พลังงานความร้อนจากดวงอาทิตย์เดินทางผ่านสุญญากาศมายังโลก



รูปที่ 2.1 แสดงลักษณะของการนำความร้อน การพาความร้อนและการแผ่รังสี

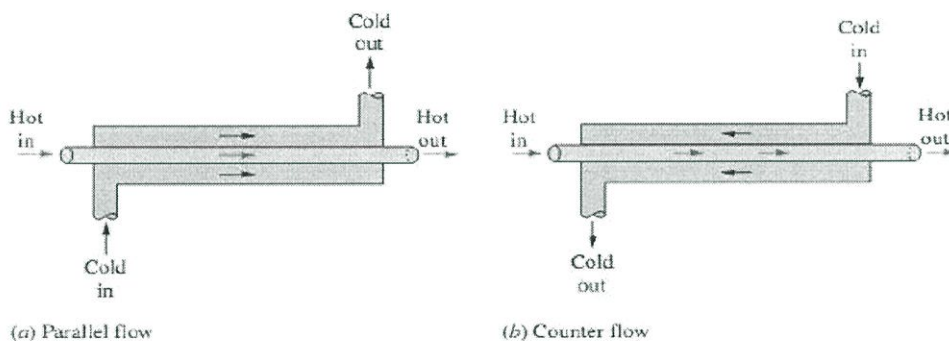
2.1.1 เครื่องแลกเปลี่ยนความร้อน (Heat Exchanger)

เครื่องแลกเปลี่ยนความร้อน คือ เครื่องมือที่ใช้สำหรับการถ่ายเทความร้อนจากของไหลชนิดหนึ่งไปยังของไหลอีกชนิดหนึ่ง โดยที่ของไหลไม่จำเป็นต้องผสมกัน เครื่องแลกเปลี่ยนความร้อนเป็นเครื่องมือที่สำคัญมากอย่างหนึ่ง เพราะเป็นเครื่องมือที่ใช้ในระบบต่างๆ ทางวิศวกรรมอย่างกว้างขวาง วิศวกรควรมีความรู้ที่จะคำนวณการออกแบบเครื่องแลกเปลี่ยนความร้อนได้

ในปัจจุบัน กระบวนการอุตสาหกรรมที่เกี่ยวข้องกับพลังงานความร้อน ส่วนใหญ่มีเครื่องแลกเปลี่ยนความร้อนเป็นองค์ประกอบ เช่น อุตสาหกรรมน้ำมัน เครื่องแลกเปลี่ยนความร้อนจะถูกใช้ในการเพิ่มอุณหภูมิของน้ำมันดิบ สำหรับเปลี่ยนสถานะของไอน้ำที่ออกมาจากหม้อต้มให้เป็นของเหลว และสำหรับการลดอุณหภูมิของน้ำมันหรือก๊าซ ในทำนองเดียวกันกับในอุตสาหกรรมปุ๋ย อุตสาหกรรมเส้นใยสังเคราะห์และอุตสาหกรรมอื่นๆ ก็ใช้เครื่องแลกเปลี่ยนความร้อนสำหรับเพิ่มอุณหภูมิ ลดอุณหภูมิหรือหมุนเวียนความร้อนจากของไหลกลับมาใช้ใหม่

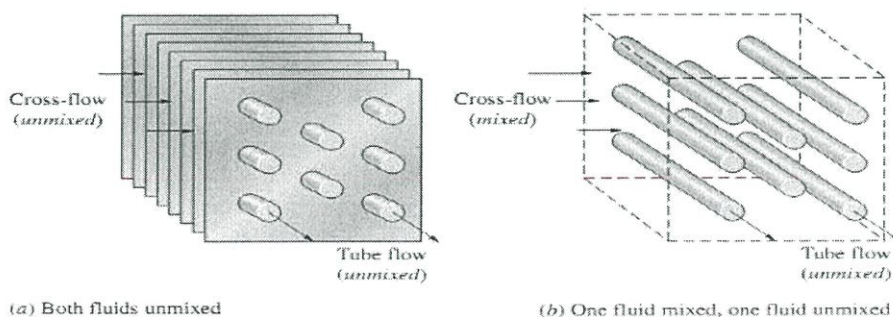
ชนิดของเครื่องแลกเปลี่ยนความร้อนแบ่งตามทิศทางการไหลของของไหลได้ 3 แบบ คือ

1. เครื่องแลกเปลี่ยนความร้อนแบบท่อสองชั้น (Concentric tube) เครื่องแลกเปลี่ยนความร้อนชนิดนี้อาจอยู่ในลักษณะที่ท่อสองท่อสวมเข้าด้วยกันดังรูป ส่วนการไหลอาจไหลสวนทางกัน เรียกว่า Counter flow หรือไหลขนานกัน เรียกว่า Parallel flow



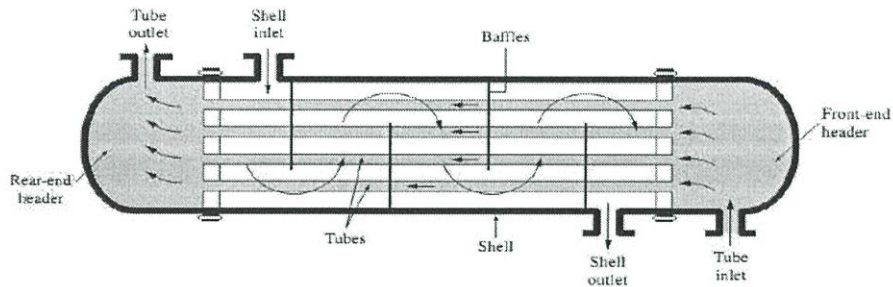
รูปที่ 2.2 แสดงลักษณะการไหลที่แตกต่างของเครื่องแลกเปลี่ยนความร้อนแบบท่อสองชั้น

2. เครื่องแลกเปลี่ยนความร้อนแบบที่ของไหลมีทิศทางตั้งฉากกัน (Cross flow) ในเครื่องแลกเปลี่ยนความร้อนแบบนี้ของไหลจะไหลในทิศทางตั้งฉากกันดังรูป



รูปที่ 2.3 แสดงการไหลที่แตกต่างกันใน Cross flow heat exchanger

3. เครื่องแลกเปลี่ยนความร้อนแบบเชลล์และท่อ (Shell and tube) เครื่องแลกเปลี่ยนความร้อนชนิดนี้ของไหลอย่างหนึ่งจะอยู่ในเชลล์ และอีกอย่างหนึ่งจะอยู่ในท่อ สำหรับการไหลนั้นจะอยู่ในลักษณะไหลสวนทาง หรือไหลขนานกันก็ได้



รูปที่ 2.4 แสดงลักษณะการไหลของเครื่องแลกเปลี่ยนความร้อนแบบเชลล์และท่อ

2.1.2 การคำนวณ Energy balance และประสิทธิภาพของ Heat exchanger

$$\text{พลังงานความร้อนสูญเสียจากด้าน Hot side} = -Q = m_H \times C_{pH} \times (T_{iH} - T_{oH}) \dots (1)$$

$$\text{พลังงานความร้อนที่ได้รับด้าน Cold side} = Q = m_C \times C_{pC} \times (T_{oC} - T_{iC}) \dots (2)$$

เปรียบเทียบสมการ (1) และ (2),

$$m_H \times C_{pH} \times (T_{iH} - T_{oH}) = m_C \times C_{pC} \times (T_{oC} - T_{iC}) \dots (3)$$

m_H และ m_C คือ อัตราการไหลเชิงมวลของ hot fluid และ cold fluid ในหน่วย kg/hr

C_{pH} และ C_{pC} คือ ความจุความร้อนของ hot fluid และ cold fluid ในหน่วย Joules/kg

T_{oC} และ T_{iC} คือ อุณหภูมิขาออกและขาเข้าของ Heat exchanger ฝั่ง cold side ในหน่วย °C

T_{oH} และ T_{iH} คือ อุณหภูมิขาออกและขาเข้าของ Heat exchanger ฝั่ง hot side ในหน่วย °C

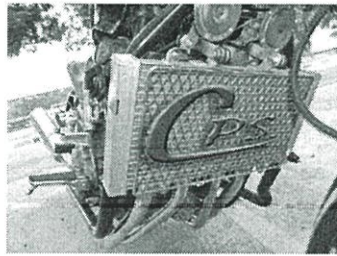
ประสิทธิภาพของ heat exchanger หาจากสมการ $Q_2/Q_1 \times 100$

2.1.3 Radiator

หม้อน้ำ(Radiator) คือตัวแลกเปลี่ยนความร้อน ใช้ในการระบายพลังงานความร้อนจากที่หนึ่งไปสู่อีกที่หนึ่ง เพื่อช่วยในการทำให้ระบบเย็นลงหรือร้อนขึ้น แล้วแต่ระบบนั้นๆจะใช้งาน หลักๆแล้วหม้อน้ำจะใช้กับงานลักษณะ ดังนี้

1. อุตสาหกรรมรถยนต์ เช่น หม้อน้ำด้านหน้ารถยนต์ รถบรรทุก รถประจำทาง รถมอเตอร์ไซด์

2. ตัวอาคาร เช่น การใช้งานในส่วนของเครื่องปั่นไฟ ซึ่งแล้วแต่ตัวขนาดอาคารว่าเล็กหรือใหญ่และจะใช้น้ำของเครื่องปั่นไฟที่แตกต่างกันไป หรือ ในกรณีที่ต่างประเทศ หม้อน้ำจะใช้ในการทำความร้อนให้กับตัวอาคารในช่วงฤดูหนาว
3. ระบบอิเล็กทรอนิกส์ เช่น หม้อน้ำในระบบระบายความร้อนของน้ำจากหัวเครื่องเชื่อม หรือการระบายความร้อนในระบบอิเล็กทรอนิกส์ที่มีการใช้การประมวลผลมากๆ ซึ่งจะทำให้เกิดความร้อนมาก และการใช้แค่อากาศนั้นไม่เพียงพอสำหรับการทำความเย็นให้อยู่ในระดับที่ปลอดภัยกับตัวระบบจึงต้องนำหม้อน้ำและปั้มน้ำเข้ามาช่วยในการระบายความร้อนของระบบ



รูปที่ 2.5 แสดงตัวอย่างหม้อน้ำที่ติดอยู่ด้านหน้ารถมอเตอร์ไซด์

หม้อน้ำนั้นเป็นส่วนหนึ่งของการระบายความร้อน หรือทำความร้อนในสภาพแวดล้อมที่มันอยู่ ในที่นี้หมายถึงการใช้หม้อน้ำในการช่วยทำให้สภาพแวดล้อมนั้นๆ ร้อนขึ้น หรือการทำให้ของเหลวหรือน้ำยาทำความเย็นที่อยู่ในตัวมันเย็นลง เพื่อจะทำการระบายความร้อนให้กับเครื่องยนต์ ตามชื่อมันแล้วในภาษาอังกฤษนั้นหม้อน้ำ หรือ radiator ไม่ได้ทำงานตรงกับชื่อมันคือไม่ได้ใช้หลักการแผ่รังสี (Radiation) แต่มันเป็นการแลกเปลี่ยนความร้อนด้วยการสัมผัส (convection) ในที่นี้อาจจะหมายถึงการที่หม้อน้ำสัมผัสน้ำภายในและระบายความร้อนออกมาสัมผัสกับอากาศภายนอก ซึ่งก็คือการแลกเปลี่ยนความร้อนชนิดหนึ่งเหมือนกัน

ระบบการทำความร้อนหรือระบายความร้อนด้วยหม้อน้ำนั้นถูกคิดค้นโดย Franz San Galli ซึ่งเป็นนักธุรกิจชาวรัสเซียที่เกิดในโปแลนด์อาศัยอยู่ที่เซนต์ปีเตอส์เบิร์ก ในช่วงปี 1855 และ 1857

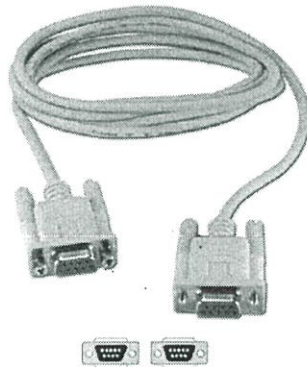
2.2 การติดต่อสื่อสารผ่านพอร์ตอนุกรม (Serial port)

2.2.1 พอร์ตอนุกรม (Serial Port)

เป็นพอร์ตสำหรับต่อกับอุปกรณ์อินพุตและเอาต์พุต โดยส่วนใหญ่เราจะใช้สำหรับต่อกับเมาส์ในกรณีที่คอมพิวเตอร์เครื่องนั้นไม่มีพอร์ต PS/2 หรือเป็นเคสแบบ AT นอกจากนั้นเรายังใช้สำหรับเป็นช่องทางการติดต่อโมเด็มด้วย ในคอมพิวเตอร์หนึ่งเครื่องจะมีพอร์ตอนุกรมให้อยู่สองพอร์ต เรียกว่าพอร์ตคอม 1 และพอร์ตคอม 2 นอกจากนั้นอาจจะมีฮาร์ดแวร์บางตัว พอร์ตอนุกรมจะมีหัวเข็ม 9 เข็ม หรือ 25 เข็ม (พอร์ตนี้จะเป็นตัวผู้เพราะมีเข็มยื่นออกมา) พอร์ตนี้จะต่อกับอุปกรณ์ต่าง ๆ เช่น เมาส์ โมเด็ม สแกนเนอร์ เป็นต้น สามารถต่อความยาวได้ถึง 6 เมตร และราคาสายก็ไม่แพงนัก

2.2.2 มาตรฐาน RS232

- เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรมที่นิยมใช้มากที่สุด
- กำหนดโดย EIA (Electronics Industry Association) หรือสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกาตั้งแต่ปี 1969
- เริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น
- RS ย่อมาจาก Recommend Standard
- 232 เป็นเลขบ่งบอกมาตรฐาน



รูปที่ 2.6 แสดงพอร์ตอนุกรม RS232 และสายที่ใช้เชื่อมต่อ

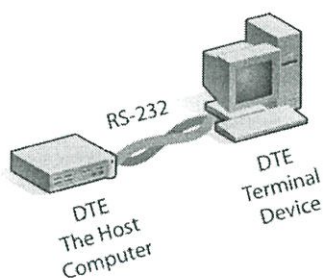
จุดประสงค์ของมาตรฐานนี้คือเพื่อบรรยายคุณลักษณะของการเชื่อมต่อ อุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE) กับ อุปกรณ์สื่อสารข้อมูล (Data Communication Equipment: DCE) ซึ่งจะขึ้นอยู่กับผู้ผลิต

2.2.2.1 อุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE)

- เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (data source) หรือ ตัวรับข้อมูล (data link) หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้
- ประกอบด้วย function unit คือ control logic, buffer/store, input/output device และ เครื่องคอมพิวเตอร์

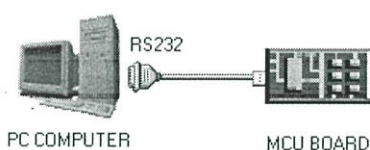
2.2.2.2 อุปกรณ์สื่อสารข้อมูล (Data Communication Equipment: DCE)

- DTE จะแทนแหล่งกำเนิดข้อมูลแหล่งแรกและอุปกรณ์ที่เป็นแหล่งรับข้อมูลแหล่งสุดท้าย อาทิ เครื่องพิมพ์ หรือ จอภาพ
- อุปกรณ์ที่รับข้อมูลได้เพียงอย่างเดียวจะเป็น DTE เพราะเป็นอุปกรณ์ที่รับข้อมูลเป็นตัวสุดท้าย คีย์บอร์ดเป็นทั้งตัวรับและตัวกำเนิดข้อมูล
- DCE เป็นอุปกรณ์ที่ทำให้การสื่อสารข้อมูลระหว่างแหล่งกำเนิดกับตัวรับข้อมูลที่ปลายทาง ทำให้สะดวกขึ้น ตัวอย่างหนึ่งของ DCE ก็คือโมเด็ม



รูปที่ 2.7 แสดงการเชื่อมต่อด้วย RS232 ระหว่าง Host computer และ Terminal device

2.2.3 การใช้งานพอร์ตอนุกรม RS232



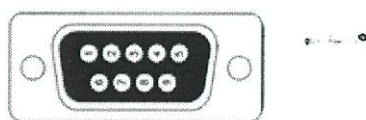
รูปที่ 2.8 แสดงการเชื่อมต่อระหว่าง PC และ MCU Board ด้วย RS232

การสื่อสารแบบอนุกรมนับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์และจอภาพของ PC เป็นอินพุตและเอาต์พุตในการติดต่อ หรือควบคุมไมโครคอนโทรลเลอร์ด้วยสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้น คือ สายส่งสัญญาณ TX, สายรับสัญญาณ RX, สาย GND

โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณขึ้นอยู่กับชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน

- พอร์ตอนุกรมของ PC จะเป็นคอนเน็คเตอร์แบบ DB9 ตัวผู้ (Male)
- พอร์ตอนุกรมของอุปกรณ์ภายนอกจะเป็นคอนเน็คเตอร์แบบ DB9 ตัวเมีย (Female)

2.2.3.1 การจัดขาของคอนเน็คเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ



รูปที่ 2.9 แสดงขาของคอนเน็คเตอร์แบบ DB9 ตัวผู้

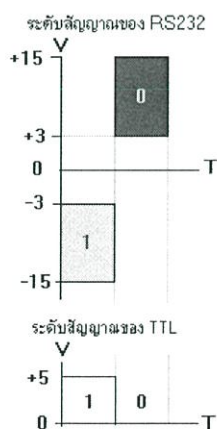
Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RD)	Input
3	Transmitted Data (TD)	Output

4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request to Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

การทำงานของขาสัญญาณ

TX	เป็นขาที่ใช้ส่งข้อมูล
RX	เป็นขาที่ใช้รับข้อมูล
DTR	แสดงสถานะพอร์ตว่าเปิดใช้งาน
DSR	ตรวจสอบว่าพอร์ตที่ติดต่อดำเนินการเปิดอยู่หรือไม่ <ul style="list-style-type: none"> เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าการติดต่อดำเนินการ ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่
RTS	แสดงสถานะพอร์ตที่ต้องการส่งข้อมูล
CTS	ตรวจสอบว่าพอร์ตที่ติดต่อดำเนินการส่งข้อมูลหรือไม่ <ul style="list-style-type: none"> เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่
GND	ขา ground

2.2.3.2 ระดับสัญญาณของ RS232



รูปที่ 2.10 แสดงระดับสัญญาณของ RS232C และระดับสัญญาณของ TTL

- สัญญาณรบกวนที่เกิดขึ้นในสายนำสัญญาณส่วนมากจะมีแรงดันเป็นบวกเมื่อเทียบกับ ground เพื่อป้องกันสัญญาณรบกวนนี้จึงออกแบบแรงดันของลอจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -15 ส่วนแรงดันของลอจิก "0" อยู่ในช่วง +3V ถึง +15V และเหตุที่ระดับสัญญาณของ RS232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น
- ดังนั้นจึงจำเป็นต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

อัตราการส่งข้อมูล (Baud rate)

คือความเร็วของการรับ-ส่งข้อมูลเป็นจำนวนบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 , 9,600 ,14,400 ,19,200, 38,400 ,56,000 เป็นต้น

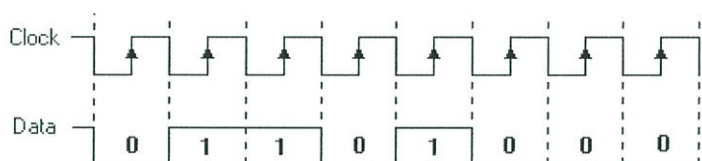
การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

2.2.3.3 รูปแบบการสื่อสารแบบอนุกรม

มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

1. การสื่อสารแบบซิงโครนัส (Synchronous)

การรับส่งข้อมูลจะมีสัญญาณนาฬิกาซึ่งเป็นตัวกำหนดจังหวะเวลาการส่งข้อมูลร่วมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูลตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด

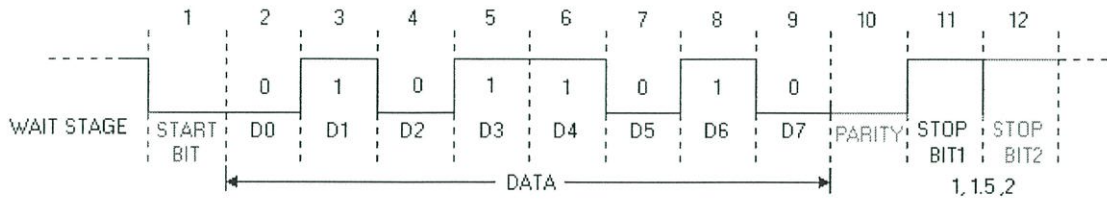


รูปที่ 2.11 แสดงการสื่อสารแบบซิงโครนัส

2. การสื่อสารแบบอะซิงโครนัส (Asynchronous)

เป็นการรับส่งข้อมูลโดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้ให้ตัวส่งและตัวรับมีอัตราส่งข้อมูลที่เท่ากัน รูปแบบข้อมูลแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูล (Data) มีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต



รูปที่ 2.12 แสดงการสื่อสารแบบอะซิงโครนัส

- เมื่อไม่มีการส่งข้อมูล ขา data จะมีสถานะเป็นลอจิก "1" หรือสถานะหยุดรอ (Waiting stage)
- เมื่อเริ่มต้นส่งข้อมูลจะให้ขา data เป็นลอจิก "0" เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit)
- จากนั้นก็จะเริ่มต้นส่งข้อมูลโดยส่งบิตต่ำไปก่อน (LSB)
- แล้วตามด้วยพริตต์บิต (จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับการติดตั้งค่าของทั้งสองฝ่าย)
- สุดท้ายตามด้วยลอจิก "1" อย่างน้อย 1 บิต (มีขนาด 1, 1.5, หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

การรับและส่งข้อมูลแบบอนุกรมยังแบ่งออกเป็นลักษณะการใช้งานได้ 3 แบบคือ

1. แบบซิมเพลกซ์ (Simplex) เป็นการส่งหรือรับข้อมูลแบบทิศทางเดียวเท่านั้น
2. แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลแบบสลับกัน คือ เมื่อด้านหนึ่งส่งอีกด้านหนึ่งเป็นฝ่ายรับสลับกัน ไม่สามารถรับ-ส่งในเวลาเดียวกันได้
3. แบบฟูลดูเพลกซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

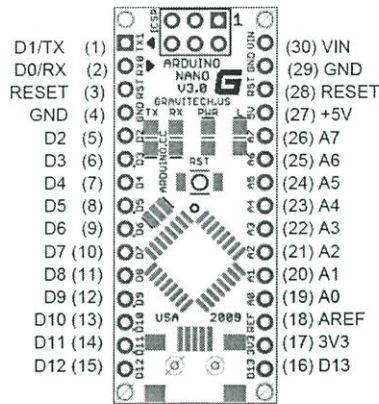
2.3 Microcontroller Board - Arduino Nano V3.0

Arduino คือ ไมโครคอนโทรลเลอร์ที่สามารถใช้เป็นเครื่องมือในการควบคุมอุปกรณ์อิเล็กทรอนิกส์ต่างๆ (I/O – device) เพื่อใช้งานตามที่เราต้องการ สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยการเสียบสาย USB เชื่อมต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ โดยใช้ไฟเลี้ยงจากสาย USB (+5V)

ทั้งนี้ Arduino ยังจัดได้ว่าเป็นรูปแบบการพัฒนาประเภท Open Source สามารถเรียกใช้หรือเพิ่ม Library ต่างๆเพื่อสะดวกใช้งานตามจุดประสงค์ที่เราต้องการจึงทำให้ Arduino เป็นที่นิยมใช้งานมากในปัจจุบัน Arduino มีคุณสมบัติสามารถต่อกับอุปกรณ์อิเล็กทรอนิกส์ เช่น ตัววัดอุณหภูมิ มอเตอร์เซอร์โว รีเลย์ ไดโอดเปล่งแสง (LED) Analog to Digital Converter และอื่นๆอีกมากมายและโปรแกรมใช้งาน Arduino สามารถใช้งานได้หลายแพลตฟอร์มทั้ง Window Linux และ MAC OSX



รูปที่ 2.13 แสดงลักษณะของ Arduino Nano V3.0



รูปที่ 2.14 แสดงขาต่างๆของ Board Arduino nano v3.0

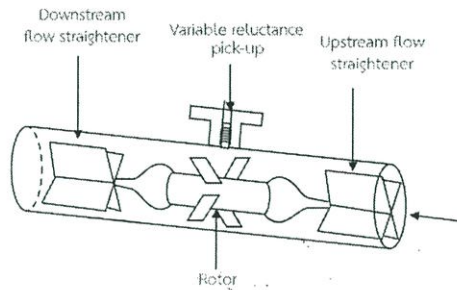
Pin1	= D1/TX
Port D1	= Digital input/output
Port TX	= ทำหน้าที่ส่งข้อมูลออกไปภายนอก
Pin 2	= D0/RX
Port D0	= Digital input/output
Port RX	= ทำหน้าที่รับข้อมูลจากภายนอก
Pin 3/28	= Port RST คือ Reset ทำหน้าที่ Reset (active low)
Pin 4/29	= Port GND คือ Supply ground
Pin 5-16	= Port D2-D13 คือ Digital input/output
Pin 17	= Port 3V3 คือ output จ่ายไฟ +3.3v
Pin 18	= Port input Analog to Digital Converter Reference
Pin 19-26	= Port A0-A7 คือ Analog input channel
Pin 27	= Port 5V คือ +5V output (from on-board regulator) หรือ +5V input
Pin 30	= Port VIN คือ Supply voltage

2.4 เครื่องมือวัดการไหลแบบกังหัน (Turbine flow meter)

เป็นเครื่องมือวัดการไหล (flow measurement) เชิงกลที่สามารถวัดการไหลได้ทั้งของเหลวและก๊าซ พัฒนาตัดแปลงมาจากเครื่องมือวัดความเร็วลมชนิดแอนนิโมมิเตอร์แบบรูปถ้วย ซึ่งความเร็วรอบในการหมุนของแอนนิโมมิเตอร์ได้รับการสอบเทียบ (calibration) กับความเร็วลม หากทราบความเร็วรอบการหมุนก็สามารถทราบความเร็วลมได้ เครื่องมือวัดการไหลแบบกังหันได้ออกแบบโดยใช้ชุดกังหัน (turbine) ติดตั้งภายในท่อที่ของไหลไหลผ่าน (ดังรูปที่ 2.16) ทำให้วงล้อใบพัดหมุน โดยความเร็วรอบที่ใบพัดหมุนแปรผันตรงกับความเร็วที่ของไหลไหลผ่าน โดยทั่วไปความเร็วที่ใบพัดหมุนสามารถวัดได้ด้วยแรงดันไฟฟ้าเหนี่ยวนำที่ได้จากcoil ตรวจจับแม่เหล็ก (magnetic pick-up coil) มีลักษณะเป็นขดลวดพันรอบแม่เหล็กถาวรติดตั้งอยู่ที่ตัวเครื่องมือวัด โดยใบพัดต้องทำจากวัสดุที่มีสมบัติเป็นสารแม่เหล็ก เมื่อใบพัดแต่ละใบเคลื่อนที่ตัดผ่าน coil

ตรวจจับแม่เหล็กจะเกิดค่าแรงดันไฟฟ้าทางด้านเอาต์พุตออกมา มีลักษณะเป็นพัลส์ (pulse) ตามความเร็วในการหมุนของใบพัด และใช้ตัวนับแบบดิจิตอลตรวจนับจำนวนพัลส์ที่เกิดขึ้น จากนั้นทำการประมวลผลด้วยไมโครโปรเซสเซอร์แปลงหน่วยให้อยู่ในรูปของปริมาตร ซึ่งได้จากการนับจำนวนพัลส์ที่เกิดขึ้นหรืออัตราการไหลซึ่งได้จากการนับจำนวนพัลส์ในหนึ่งช่วงเวลาหรือเรียกว่า อัตราการเกิดพัลส์

นอกจากนี้การวัดความเร็วรอบของใบพัดยังสามารถทำได้โดยการติดตั้งเซนเซอร์แสง (photo sensor/optical sensor) ที่ ตัวเครื่องมือวัด โดยลักษณะของสัญญาณเอาต์พุตที่ได้เป็นพัลส์เช่นเดียวกับเอาต์พุตที่ได้จาก การวัดแรงดันไฟฟ้าเหนี่ยวนำ หรือเอาต์พุตจากอุปกรณ์เข้ารหัสเชิงมุม (rotary encoder)



รูปที่ 2.15 แสดงโครงสร้างของเครื่องมือวัดอัตราการไหลแบบเทอร์ไบน์

เครื่องมือวัดการไหลชนิดนี้มีลักษณะความเป็นเชิงเส้น (linearity) ที่ดีที่อัตราการไหลสูง ส่วนที่อัตราการไหลต่ำ ค่าการไหลที่วัดได้จะได้รับผลกระทบจากแรงต้านเนื่อง จากความเสียดทาน โดยทั่วไปเครื่องมือวัดการไหลชนิดเทอร์ไบน์มีย่านการวัด (range) การไหลของของเหลวอยู่ในช่วง 1 ถึง 100,000 ลิตรต่ออนาที และสำหรับการวัดการไหลของก๊าซมีย่านการใช้งานในช่วง 5 ถึง 100,000 ลิตรต่ออนาที ไม่สามารถใช้วัดการไหลของของไหลที่มีสารแขวนลอยปะปนได้ และในการติดตั้งเครื่องมือวัดต้องพิจารณาระยะความตรงของท่อทั้งด้านหน้าก่อน เข้าตัวเครื่องมือวัดและระยะด้านหลัง เพื่อป้องกันไม่ให้ค่าความเร็วที่วัดได้คลาดเคลื่อน (error)

2.4.1 สูตรคำนวณอัตราการไหลของ Turbine flow meter

$$K_{F,V} = \frac{f_{Hz}}{q_v} \quad \text{หน่วย} \quad \frac{\text{Pulse}}{\text{Unit Volume}}$$

เมื่อ K คือ K - factor หรือความสัมพันธ์ระหว่างความถี่และอัตราการไหลที่ได้จาก Turbine

f_{Hz} คือ ความถี่ที่ได้จาก Turbine

q_v คือ อัตราการไหลที่ได้จาก Turbine

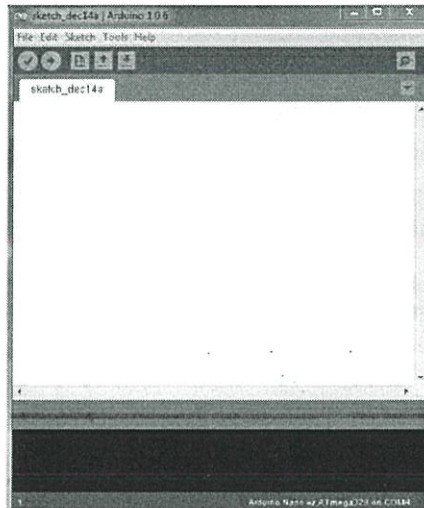
$$\text{เนื่องจาก } f_{Hz} = \frac{\text{Pulse}}{\text{sec}}$$

$$\text{ถ้า } q_v \text{ มีหน่วยเป็น } \frac{\text{l}}{\text{sec}}, \frac{\text{gallon}}{\text{sec}}, \frac{\text{m}^3}{\text{sec}}$$

$$K_{F,V} = \frac{\text{Pulse}}{\text{l}}, \frac{\text{Pulse}}{\text{gallon}}, \frac{\text{Pulse}}{\text{m}^3}$$

2.6 โปรแกรม Arduino

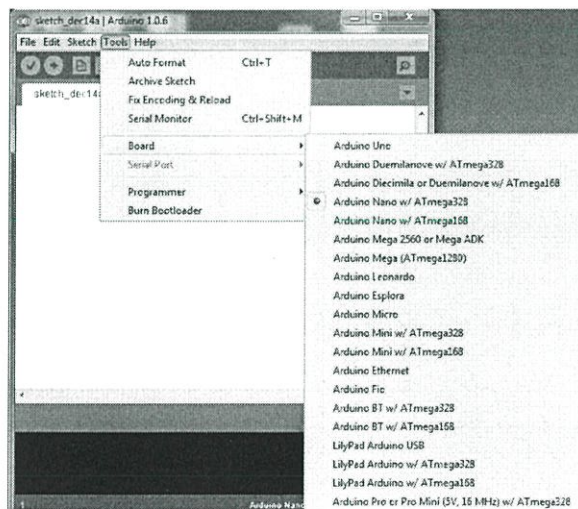
เป็นโปรแกรม Arduino เป็นโปรแกรมที่ทำให้คอมพิวเตอร์สามารถติดต่อสื่อสารกับ Board Arduino ได้และใช้ในการเขียนโปรแกรมจากคอมพิวเตอร์ไปยัง Board Arduino



รูปที่ 2.17 แสดงหน้าต่างโปรแกรม Arduino

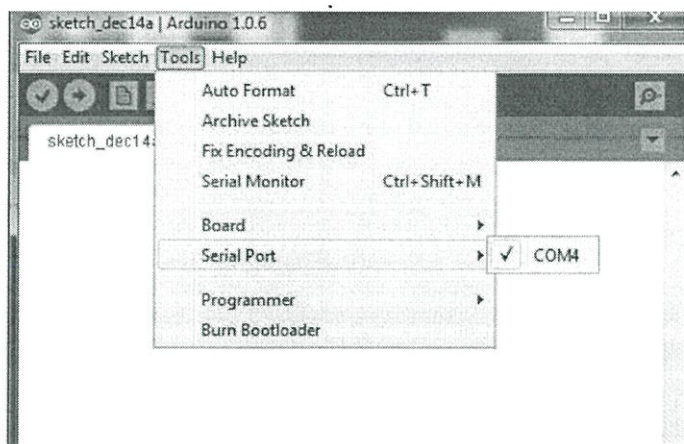
2.6.1 การตั้งค่าโปรแกรม Arduino

1. ตั้งค่า Board เพื่อเลือกรุ่นของ Arduino ตามที่ใช้งานโดยการ เลือกที่ Tools >> Board >> Arduino Nano V3.0/ATmega328



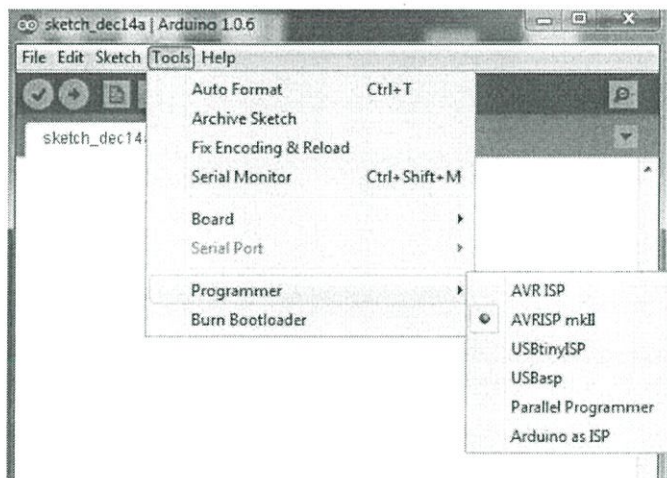
รูปที่ 2.18 การตั้งค่า Board

2. การตั้งค่า Serial Port เพื่อให้ Arduino สามารถสื่อสารกับ Computer ได้ โดยการเลือกที่ Tools >> Serial Port >> COM4



รูปที่ 2.19 การตั้งค่า Serial Port

3. การตั้งค่า Programmer เพื่อกำหนดวิธีการสื่อสารระหว่าง Arduino และ Computer โดยการเข้าไปที่ Tools >> Programmer >> AVRISP mkII



รูปที่ 2.20 การตั้งค่า Programmer

4. ลงโปรแกรมและทำการ Compile โปรแกรม โดยการกดที่เครื่องเครื่องมือ  เพื่อ Compile

5. ทำการ Upload โปรแกรมไปยัง Arduino Nano V3.0 โดยการกดที่เครื่องมือ 

2.7 โปรแกรม Microsoft visual studio

Microsoft Visual Studio คือซอฟต์แวร์ประเภท IDE (Integrated Development Environment) พัฒนาขึ้นโดยไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชันและเว็บเซอร์วิส ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ ทัชคเอดพีซี Smartphone และเว็บเบราวเซอร์ ในปัจจุบันช่วยให้ผู้พัฒนาโปรแกรมสามารถเขียนโปรแกรมด้วยความสะดวกสบายขึ้น สามารถแก้ไขข้อผิดพลาดในการเขียนโปรแกรมได้ง่ายและรวดเร็ว Visual Studio นั้นจะประกอบไปด้วยภาษาที่ใช้พัฒนาหลัก ๆ 3 ภาษา คือ Visual Basic.Net, Visual C#.Net และ Visual C++.Net ซึ่งล่าสุดได้มีการเพิ่มภาษาเข้ามาอีกหนึ่งภาษาซึ่งอยู่ในช่วงพัฒนาคือ Visual J#.Net

Visual Basic เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมในการนำมาใช้งานพัฒนาโปรแกรมบน ระบบ Windows เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize นั่นก็คือจะสะดวกในการหยิบเครื่องมือเครื่องมือที่โปรแกรมได้จัดเตรียมไว้ให้สำหรับออกแบบหน้าจอและสิ่งต่างๆ สำหรับในการเขียนโปรแกรมให้เรียบร้อย ซึ่งแตกต่างจากสมัยก่อนเวลาจะออกแบบหน้าจอก็ยังคงต้องมานั่งเขียน Source Code ให้ลำบาก

Visual Basic เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นใช้งานที่ใช้ได้ตั้งแต่ระดับต้น เพื่อใช้สร้างโปรแกรมง่าย ๆ บน Windows หรือโปรแกรมเมอร์ระดับกลางที่จะเรียกใช้ฟังก์ชันต่างๆได้อย่างมีประสิทธิภาพ ตลอดจนโปรแกรมเมอร์ระดับมืออาชีพ ที่จะพัฒนาโปรแกรมในระดับสูงโดยการใช้ Object Linking and Embedding (OLE) และ Application Programming Interface (API) ของระบบ windows มาประกอบการเขียนโปรแกรม

2.7.1 จุดเด่นของ Visual Basic

1. สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว

Visual Basic ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอปพลิเคชันให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAC ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ซากๆ ออกไปเหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริงๆ แล้วเขียนโปรแกรมจัดการปัญหานั้นๆ ส่วนเรื่องอื่นๆ เหลือให้ Visual Basic จัดการ

2. การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้ามีโอกาสเขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่า ภาษา Basic ที่ใช้ใน Visual Basic นั้น เข้าใจง่ายคือใกล้เคียงกับภาษาที่เราใช้งานปกติ สื่อความหมายและเข้าใจง่ายกว่าภาษาโปรแกรมอื่นๆ ทำให้ผู้เริ่มต้นเขียนโปรแกรมทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

3. รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic ยังมีเครื่องมือที่ช่วยในการเขียนโปรแกรมเป็นสิ่งที่ไม่ยุ่งยากเพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ้งยากมีการตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก

นอกจากจะมีการเครื่องมือช่วยในการเขียนโปรแกรมแล้วยังมีเครื่องมือที่ใช้ในการทดสอบแก้ไข โปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online help) ไว้ อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยข้อใจได้

เครื่องมือทั้งหมดที่กล่าวมาถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกลย่อว่า IDE ซึ่งย่อ มาจาก Integral Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปเรื่อยๆ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้เพื่อประหยัดพื้นที่ฮาร์ดดิสก์ก็ได้เช่นกัน

2.7.2 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic

เมื่อเรามองเห็นว่า Visual Basic จะช่วยให้เราสร้างแอปพลิเคชันบน Windows ได้ง่ายและรวดเร็ว ยังมีรูปแบบที่ Visual Basic สามารถสร้างขึ้นมาก็อีกคือ

1. พัฒนาแอปพลิเคชันกับ Active X Control

เทคโนโลยีที่มีชื่อว่า Active X เป็นตัวอยู่เบื้องหลังความสำเร็จของ Visual Basic ซึ่งช่วยลดงานที่ว้า ซ้อนของการเขียนโปรแกรมลงไปมาก

ตัวอย่างเช่นการเขียนโปรแกรมเพื่อรับค่าข้อมูลของผู้ใช้ แต่ก่อนเราต้องเขียนโปรแกรมเพื่อวาด หน้าจอ เขียนโปรแกรมวาดรูปปุ่มและช่องรับข้อความรวมทั้งการเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ผู้ใช้ ป้อนเข้ามา แต่ Active X จะทำให้เราสนใจเฉพาะการจัดการกับข้อมูลที่ผู้ใช้ป้อนเข้ามาเท่านั้น ส่วนที่เหลือจะ จัดการให้เอง โดยช่องรับข้อความและปุ่มต่างๆจะใช้ Active X Control จัดการ

นอกจากจะลดความซับซ้อนลงแล้วการใช้ Active X Control ในการเขียนโปรแกรมยังช่วยให้ โปรแกรมที่เราเขียนกับโปรแกรมที่คนอื่นเขียนนั้น ตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การบำรุงรักษาง่ายดายเพราะ ใครๆก็เข้าใจมาตรฐานของ Active X Control นี้ทำให้ไม่ต้องกังวลใจว่าโปรแกรมที่เขียนขึ้นจะมีเฉพาะคน เขียนเท่านั้นที่เข้าใจ

2. สร้างแอปพลิเคชันที่ใช้งานกับฐานข้อมูล

เป็นแอปพลิเคชันที่มีการใช้งานมากที่สุดเพราะระบบต่างๆ เช่น ร้านค้า คลังสินค้า ระบบบัญชี ระบบ บริหารงานบุคคลหรือแม้แต่ E-Commerce ทั้งหมดต่างก็ต้องมีส่วนที่ติดต่อกับฐานข้อมูล

Visual Basic ได้ช่วยในการสร้างแอปพลิเคชันกับฐานข้อมูลเป็นเรื่องที่ทำได้โดยง่ายดาย เพราะ มี เครื่องมือต่างๆ ที่คอยให้ความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูล ซึ่งไม่จำกัดด้วยว่า จะเห็นข้อมูลแบบใดทั้งฐานข้อมูลส่วนบุคคล ฐานข้อมูลผ่านเครือข่ายหรือฐานข้อมูลผ่านอินเทอร์เน็ต ตาม ความสามารถที่หลากหลายนี้ Visual Basic จึงเป็นตัวเลือกอันดับต้นๆ ของการสร้างแอปพลิเคชันที่เกี่ยวกับ งานฐานข้อมูลในโลกธุรกิจ

3. สร้างแอปพลิเคชันแบบใหม่กับอินเทอร์เน็ต

อินเทอร์เน็ตนับว่ามีความสำคัญกับชีวิตของคนเราที่ใช้ไต่เพิ่มขึ้นทุกวัน ซึ่ง Visual Basic เปิดโอกาส ให้เราสามารถสร้างแอปพลิเคชันเพื่อรองรับการทำงานร่วมกับอินเทอร์เน็ตได้ด้วย การใช้ความรู้เดิมที่เรามีจาก การสร้างแอปพลิเคชันปกติ รวมทั้งเครื่องมือเสริมการทำงานต่างๆมากมาย

จุดเด่นของ Visual Basic อีกข้อหนึ่งคือเปิดโอกาสให้เรานำแอปพลิเคชันปกติที่เดิมทำงานกับคอมพิวเตอร์ส่วนบุคคลสามารถดัดแปลงใช้งานกับอินเทอร์เน็ตได้อย่างไม่ยากเย็นนัก ทำให้ไม่ต้องเปลี่ยนแปลงรูปแบบงานเดิม

2.7.3 การสร้างแอปพลิเคชันด้วย Visual Basic

1. ออกแบบแอปพลิเคชัน

ต้องทราบให้แน่ชัดก่อนว่าแอปพลิเคชันที่เราจะทำการสร้างนั้น จะนำไปใช้ประโยชน์อะไรต้องมีความสำคัญอะไรบ้าง และต้องมีรูปร่างหน้าตาเป็นอย่างไร ซึ่งจำเป็นอย่างยิ่งที่ต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจนโดยอาจจะร่างง่าย ๆ ในกระดาษ

2. ตกแต่งหน้าต่างแอปพลิเคชัน

เป็นการตกแต่งรูปร่างของแอปพลิเคชันที่ได้ออกแบบไว้แล้วพร้อมๆ กับการคำนวณ ค่า Property ต่างๆ ให้กับ คอนโทรลแต่ละตัวในแอปพลิเคชัน

3. เขียนโค้ดกำกับแอปพลิเคชัน

หลังจากตกแต่งหน้าต่างเสร็จแล้วขั้นตอนต่อไปคือการเขียนโค้ดหรือการเขียนโปรแกรมเพื่อควบคุมการทำงานต่างๆ

4. ทดสอบการทำงานของแอปพลิเคชัน

เมื่อเขียนโค้ดเสร็จแล้วก็ถึงเวลาที่จะทดสอบการทำงานของแอปพลิเคชันที่เราสร้างขึ้น

5. บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบแน่ใจว่าแอปพลิเคชันที่สร้างขึ้นนั้นทำงานได้ถูกต้อง เราจึงบันทึกเก็บไว้ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆ ได้ภายหลัง

6. สร้างไฟล์ .EXE (Make)

เมื่อเราสร้างแอปพลิเคชันเสร็จแล้ว หากเราต้องการนำแอปพลิเคชันที่สร้างขึ้นให้เรียกใช้งานได้เองโดยไม่ต้องเรียกผ่าน Visual Basic หรือต้องการนำไปใช้งานในเครื่องคอมพิวเตอร์เครื่องอื่นๆ เราสามารถทำได้โดยสร้างไฟล์เอ็กซ์คิวต์ (ไฟล์ที่มีนามสกุลเป็น .EXE)

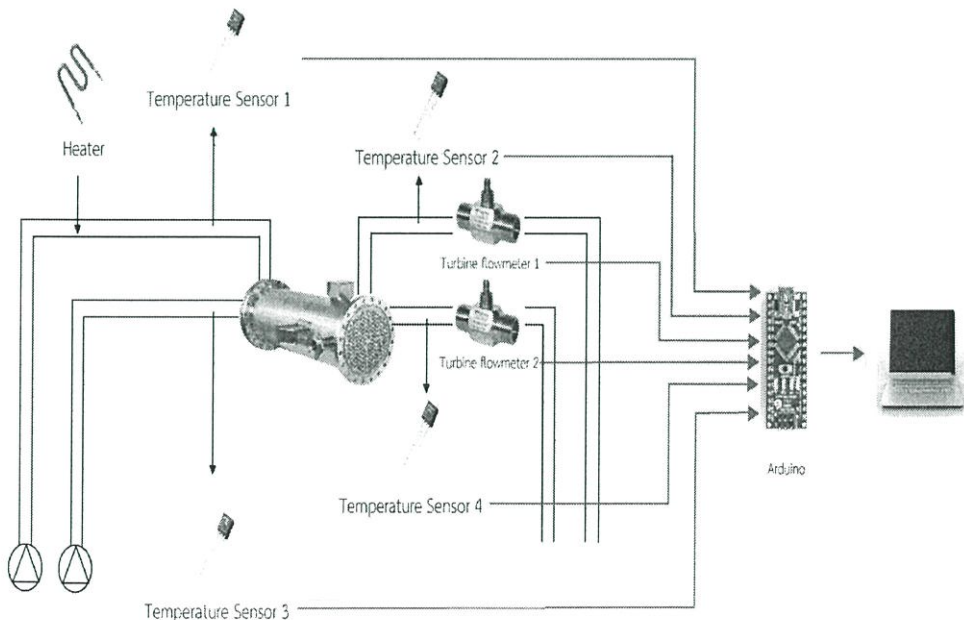
บทที่ 3

การออกแบบและวิธีการดำเนินงาน

3.1 อุปกรณ์

1. เครื่องแลกเปลี่ยนความร้อน
2. ฮีตเตอร์
3. เครื่องมือวัดการไหลแบบกักกัน FS300A (G 3/4inch)
4. เซนเซอร์วัดอุณหภูมิ (LM35)
5. ปั๊มน้ำ
6. Radiator
7. Board Arduino nano v3.0
8. คอมพิวเตอร์
9. แท็งก์น้ำ
10. ท่อทองแดง
11. ฉนวนกันความร้อน

3.2 ออกแบบชุดทดลอง



รูปที่ 3.1 แสดงแบบจำลองของชุดทดลอง

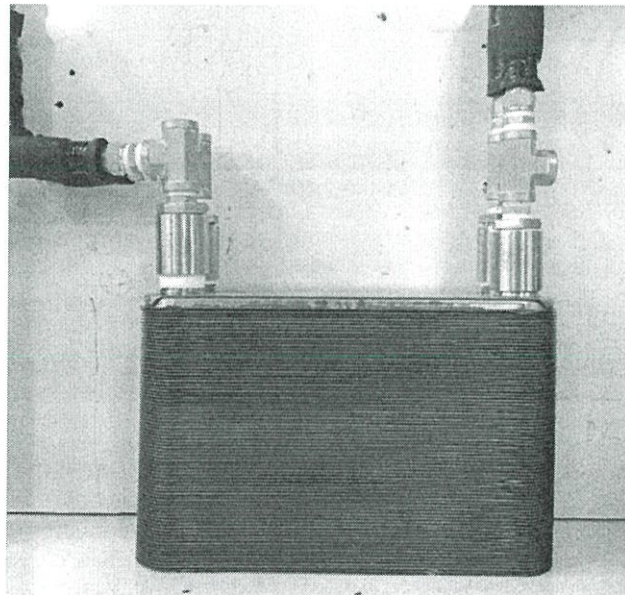
ส่วนประกอบและหน้าที่ของอุปกรณ์ในชุดทดลอง

1. Pump ทำหน้าที่สูบน้ำขึ้นมาจากแทงก์น้ำ
2. Heater ทำหน้าที่ในการเพิ่มอุณหภูมิแก่ของไหลภายในท่อ
3. Heat exchanger ทำหน้าที่ในการแลกเปลี่ยนความร้อนระหว่างของไหล
4. Radiator ทำหน้าที่ในการระบายความร้อนให้กับของไหล
5. Temperature sensor (Lm35) ทำหน้าที่ในการวัดอุณหภูมิที่ด้านขาเข้าและขาออกของท่อ
6. Turbine flow meter ทำหน้าที่ในการวัดอัตราการไหลของของไหลภายในท่อ
7. Board Arduino nano v3.0 ทำหน้าที่ในการรับค่าที่วัดได้จากอุปกรณ์การวัดและนำไปประมวลผลเพื่อแสดงค่าที่วัดได้ผ่านจอคอมพิวเตอร์ในรูปแบบกราฟด้วยซอฟต์แวร์ที่เขียนโดย Microsoft visual studio

3.3 จัดทำชุดทดลอง

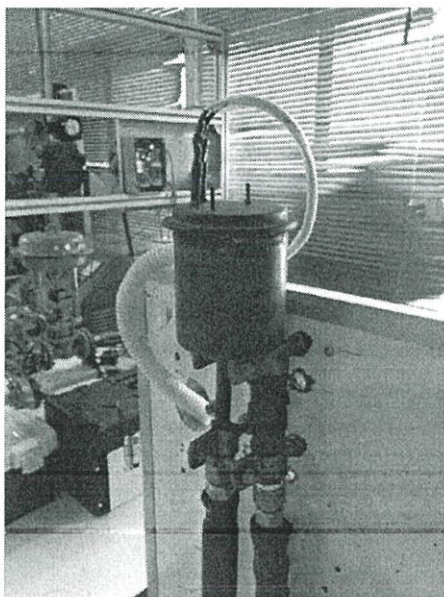
มีขั้นตอนการจัดทำดังนี้

1. ติดตั้ง Heat exchanger ดังรูป



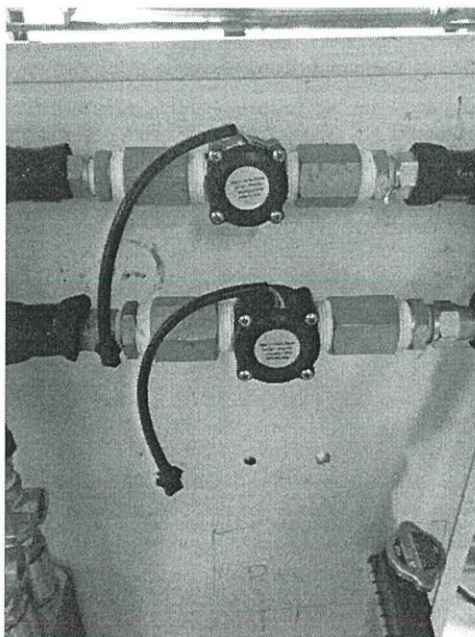
รูปที่ 3.2 แสดงการติดตั้ง Heat exchanger

- ติดตั้ง Heater และทำการต่อท่อและสายไฟกับ Heater ดังรูป



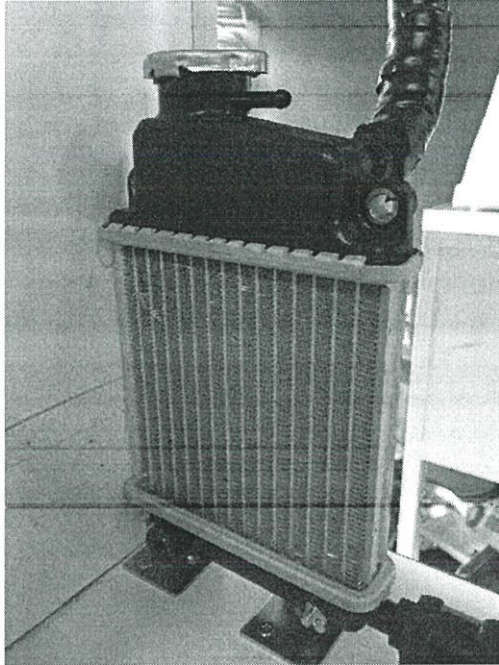
รูปที่ 3.3 แสดงการติดตั้ง Heater

- ติดตั้ง Turbine flow meter และทำการต่อท่อเข้ากับ Heat exchanger ดังรูป



รูปที่ 3.4 แสดงการติดตั้ง Turbine flow meter

- ติดตั้ง Radiator ต่อจาก Turbine flow meter (FM2) ทางด้านขาออกดังรูป



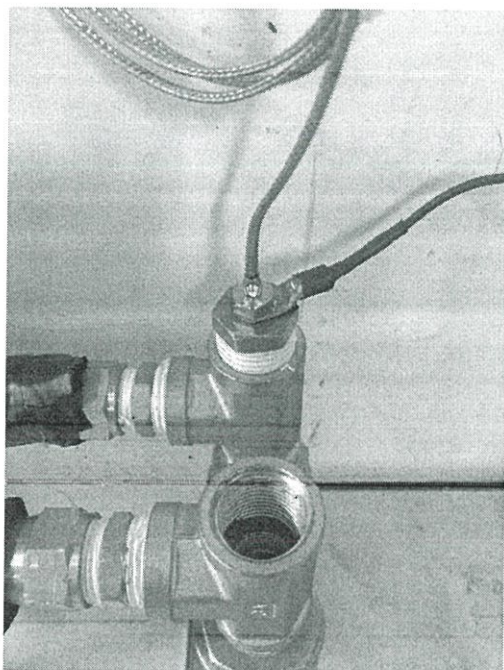
รูปที่ 3.5 แสดงการติดตั้ง Radiator

- ทำการใส่ซิลิโคนเพื่อป้องกันไม่ให้น้ำเข้าสู่ Temperature sensor (Lm35) ดังรูป



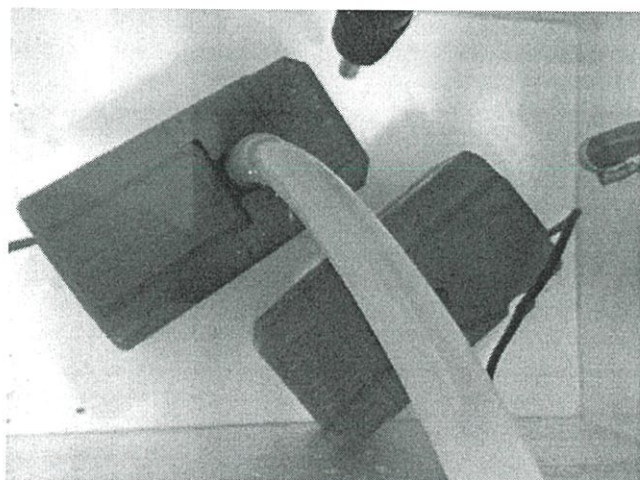
รูปที่ 3.6 แสดงจุดที่ใส่ซิลิโคน

6. ทำการติดตั้ง Temperature sensor (Lm35) เข้ากับ Heat exchanger ทางด้านขาเข้าและขาออก ดังรูป



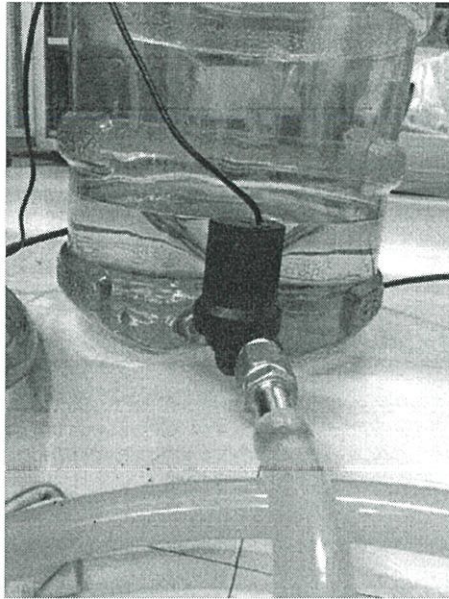
รูปที่ 3.7 แสดงการติดตั้ง Temperature sensor (Lm35)

7. ทำการติดตั้งปั๊มของแท็งก์น้ำเย็นดังรูป



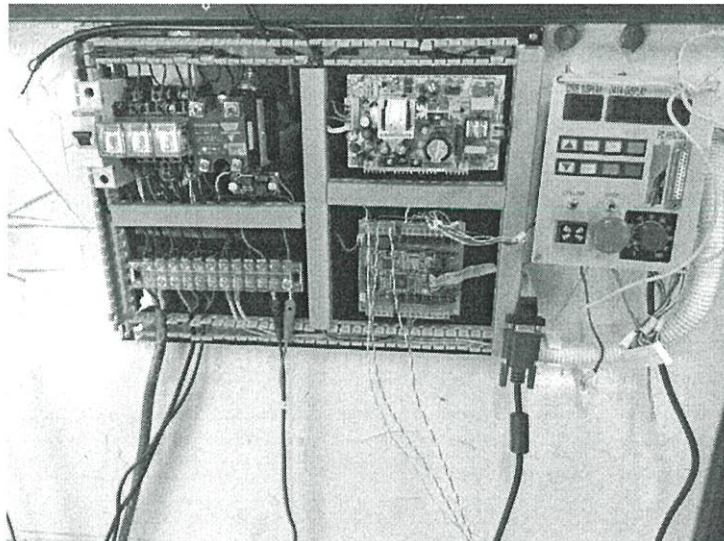
รูปที่ 3.8 แสดงการติดตั้งปั๊มของแท็งก์น้ำเย็น

8. ทำการติดตั้งปั้มของแท็งก์น้ำร้อนดังรูป



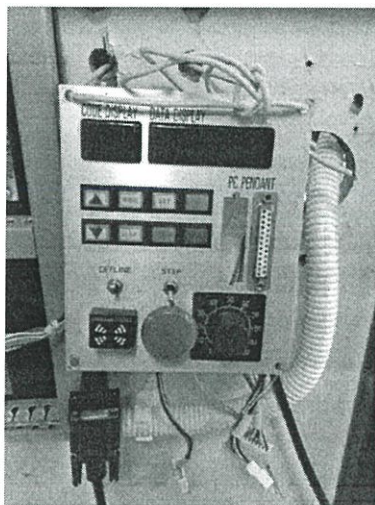
รูปที่ 3.9 แสดงการติดตั้งปั้มของแท็งก์น้ำร้อน

9. ทำการติดตั้งแผงจ่ายไฟที่ด้านหลังของชุดทดลอง



รูปที่ 3.10 แสดงการติดตั้งแผงจ่ายไฟ

10. ทำการติดตั้งสวิทซ์ในการสั่งเปิด – ปิดปั้ม ฮีตเตอร์และ Radiator



รูปที่ 3.11 แสดงการติดตั้งสวิทซ์ในการสั่งเปิด – ปิดปั้ม ฮีตเตอร์และ Radiator

3.4 การออกแบบโปรแกรมสำหรับ Arduino ในการรับและส่งค่าจากอุปกรณ์การวัด

1. การประกาศค่าตัวแปรและกำหนดขาของ Arduino ในการเป็นขา Input และ Output ดังรูป

```
volatile int spfan1;
volatile int spfan2;
int fpin1=2;
int fpin2=3;
float data[10];
```

รูปที่ 3.12 แสดงส่วนของการประกาศค่าตัวแปรและกำหนดขาของ Arduino

2. กำหนดค่าของ Void loop โดยกำหนดให้

- fpin1 และ fpin2 เป็น Input
- เมื่อ rpm1 ได้รับสัญญาณขอบขาขึ้นจะมีค่าเพิ่มขึ้นทีละ1
- เมื่อ rpm 2 ได้รับสัญญาณขอบขาขึ้นจะมีค่าเพิ่มขึ้นทีละ1
- กำหนดค่า baud rate เป็น 9600 bps

```
void setup()
{
  pinMode(fpin1, INPUT);
  pinMode(fpin2, INPUT);
  attachInterrupt(0, rpm1, RISING);
  attachInterrupt(1, rpm2, RISING);
  Serial.begin(9600);
}
```

รูปที่ 3.13 แสดงส่วนของโปรแกรม Void setup

3. สร้าง loop เพื่อให้ฟังก์ชัน attachInterrupt เรียกใช้งาน

```
void rpm1 ()
{
  spfan1++;
}
void rpm2 ()
{
  spfan2++;
}
```

รูปที่ 3.14 แสดงส่วนของการสร้าง loop

4. หลังจากสร้างฟังก์ชัน Void setup และทำการตั้งค่าเริ่มต้นและตั้งชื่อต่างๆ เพื่อวัดและคำนวณค่าของ อุณหภูมิ อัตราการไหล และพลังงานความร้อน และส่งข้อมูลผ่าน serial port

```
void loop()
{
    data[0]=analogRead(A0)*0.48828125;
    data[1]=analogRead(A1)*0.48828125;
    data[2]=analogRead(A2)*0.48828125;
    data[3]=analogRead(A3)*0.48828125;
    spfan1=0;
    spfan2=0;
    sei();
    delay (1000);
    cli();
    data[4] = (spfan1 / 330);
    data[5] = (spfan2 / 330);
    data[6] = 4187*data[4]*abs(data[1]-data[0]);
    data[7] = 4187*data[5]*abs(data[3]-data[2]);
    Serial.print("a");
    Serial.print(data[0]);
    Serial.print(",");
    Serial.print(data[1]);
    Serial.print(",");
    Serial.print(data[2]);
    Serial.print(",");
    Serial.print(data[3]);
    Serial.print(",");
    Serial.print(data[4]);
    Serial.print(",");
    Serial.print(data[5]);
    Serial.print(",");
    Serial.print(data[6]);
    Serial.print(",");
    Serial.print(data[7]);
    Serial.print("a");
}
```

รูปที่ 3.15 แสดงฟังก์ชัน Void loop

3.5 การสร้างโปรแกรม Plant Monitoring Arduino จากโปรแกรม Microsoft Visual Studio

1. การประกาศตัวแปรและสร้าง sub

```
Public Class Form1
    Dim x As Integer = 0
    Dim arr1(120), arr2(120), arr3(120), arr4(120), arr5(120), arr6(120), arr7(120), arr8(120) As Integer
    Private comBuffer As Byte()
    Private Delegate Sub UpdateFormDelegate()
    Private UpdateFormDelegate1 As UpdateFormDelegate
    Public Delegate Sub StringsSubPointer(ByVal Buffer As String)
    Dim WithEvents sp As New SerialPort
    Private Sub GetSerialPortNames()
        For Each sport As String In My.Computer.Ports.SerialPortNames
            cmbPort.Items.Add(sport)
        Next
    End Sub
```

รูปที่ 3.16 แสดงส่วนของการประกาศตัวแปร

2. เป็นส่วนของการกำหนดค่า Baud Rate และ ตั้งค่าสเกลของกราฟ

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim BaudRates() As String = {"600", "1200", "2400", "4800", "9600", "14400", "19200", "28800", "38400", "57600", "115200"}
    sp.Parity = IO.Ports.Parity.None
    sp.DataBits = 8
    sp.StopBits = IO.Ports.StopBits.One
    sp.ReadTimeout = 20000
    sp.Handshake = Handshake.None
    btnG1.Enabled = False
    Chart3.Hide()
    Chart1.ChartAreas("ChartArea1").AxisX.Minimum = 0
    Chart1.ChartAreas("ChartArea1").AxisX.Maximum = 120
    Chart1.ChartAreas("ChartArea1").AxisX.Interval = 10
    Chart1.ChartAreas("ChartArea1").AxisY.Minimum = 0
    Chart1.ChartAreas("ChartArea1").AxisY.Maximum = 100
    Chart1.ChartAreas("ChartArea1").AxisY.Interval = 10
    Chart2.ChartAreas("ChartArea1").AxisX.Minimum = 0
    Chart2.ChartAreas("ChartArea1").AxisX.Maximum = 120
    Chart2.ChartAreas("ChartArea1").AxisX.Interval = 10
    Chart2.ChartAreas("ChartArea1").AxisY.Minimum = 0
    Chart2.ChartAreas("ChartArea1").AxisY.Maximum = 15
    Chart2.ChartAreas("ChartArea1").AxisY.Interval = 1.5
    Chart3.ChartAreas("ChartArea1").AxisX.Minimum = 0
    Chart3.ChartAreas("ChartArea1").AxisX.Maximum = 120
    Chart3.ChartAreas("ChartArea1").AxisX.Interval = 10
    Chart3.ChartAreas("ChartArea1").AxisY.Minimum = 0
    Chart3.ChartAreas("ChartArea1").AxisY.Maximum = 8000
    Chart3.ChartAreas("ChartArea1").AxisY.Interval = 1000

    cmbBaud.Items.AddRange(BaudRates)
    cmbBaud.SelectedIndex = 8
    Try
        GetSerialPortNames()
        cmbPort.SelectedIndex = 0
    Catch
        MsgBox("No ports connected.")
    End Try

```

รูปที่ 3.17 แสดงส่วนของโปรแกรมที่ใช้ในการกำหนดสเกลของกราฟ

3. การสร้างการเชื่อมต่อ(Connect) โดยเมื่อเชื่อมต่อแล้วจะใช้ค่า COM และ Baud Rate มาใช้ และอ่านข้อมูลจาก Serial port

```

Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnConnect.Click
    Try
        sp.BaudRate = cmbBaud.SelectedItem.ToString
        sp.PortName = cmbPort.SelectedItem.ToString
        sp.Open()
        If sp.IsOpen Then
            Timer1.Interval = 1000
            Timer1.Start()
            btnConnect.Enabled = False
            btnDisconnect.Enabled = True
            cmbPort.Enabled = False
            cmbBaud.Enabled = False
        Else : Timer1.Stop()

        End If
    Catch
        btnDisconnect.Enabled = True
        sp.Close()
    End Try
End Sub
Private Sub Receiver(ByVal sender As Object, _
    ByVal e As SerialDataReceivedEventArgs) Handles sp.DataReceived
    Me.BeginInvoke(New StringSubPointer(AddressOf Display), sp.ReadExisting)
End Sub

```

รูปที่ 3.18 แสดงส่วนของโปรแกรมการเชื่อมต่อ

4.เป็นส่วนในการเรียกอ่านข้อมูลจาก Buffer และแยกเก็บไว้ใน Array

```

Private Sub Display(ByVal Buffer As String)
    Dim strArr() As String
    Dim dataArr() As String
    Dim e1, e2, f1, f2 As String
    For i As Integer = 1 To Buffer.Length - 1

        Next

    strArr = Buffer.Split({"a"}, StringSplitOptions.RemoveEmptyEntries)
    If strArr.Length > 0 Then
        dataArr = strArr(strArr.Length - 1).Split({","}, StringSplitOptions.RemoveEmptyEntries)
        If dataArr.Length = 8 Then
            f1 = dataArr(5) / 60
            f2 = dataArr(6) / 60
            f1 = FormatNumber(f1, 2)
            f2 = FormatNumber(f2, 2)
            dataArr(5) = dataArr(5) / 3600
            dataArr(5) = FormatNumber(dataArr(5), 2)
            dataArr(6) = dataArr(6) / 3600
            dataArr(6) = FormatNumber(dataArr(6), 2)
            e1 = 4187 * dataArr(5) * Math.Abs(dataArr(2) - dataArr(1))
            e2 = 4187 * dataArr(6) * Math.Abs(dataArr(4) - dataArr(3))
            e1 = FormatNumber(e1, 2)
            e2 = FormatNumber(e2, 2)

            Label15.Text = dataArr(1)
            Label16.Text = dataArr(2)
            Label17.Text = dataArr(3)
            Label18.Text = dataArr(4)
            Label19.Text = f1
            Label20.Text = f2
            Label21.Text = e1
            Label22.Text = e2

        End If
    End If
End Sub

```

รูปที่ 3.19 แสดงโปรแกรมในการเรียกอ่านข้อมูลจาก Buffer

5.การสร้างส่วนของการตัดการเชื่อมต่อ (disconnect) และสร้างเงื่อนไขของการปิดโปรแกรม

```

Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisconnect.Click
    Try
        sp.Close()
        Timer1.Stop()
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
        cmbPort.Enabled = True
        cmbBaud.Enabled = True
        Exit Sub
    Catch
        MessageBox.Show("Some kind of problem.")
    End Try
End Sub

Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    If sp.IsOpen() Then
        MessageBox.Show("Disconnect before closing")
        e.Cancel = True
    End If
End Sub

```

รูปที่ 3.20 แสดงโปรแกรมในการสร้างส่วนของการตัดการเชื่อมต่อ

6. การสร้าง Timer เพื่อใช้ในการวาดกราฟ

```

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    For i As Integer = 0 To 119
        arr1(i) = arr1(i + 1)
        arr2(i) = arr2(i + 1)
        arr3(i) = arr3(i + 1)
        arr4(i) = arr4(i + 1)
        arr5(i) = arr5(i + 1)
        arr6(i) = arr6(i + 1)
        arr7(i) = arr7(i + 1)
        arr8(i) = arr8(i + 1)
    Next
    arr1(120) = Label15.Text
    arr2(120) = Label16.Text
    arr3(120) = Label17.Text
    arr4(120) = Label18.Text
    arr5(120) = Label19.Text
    arr6(120) = Label20.Text
    arr7(120) = Label21.Text
    arr8(120) = Label22.Text
    Chart1.Series("temp1").Points.Clear()
    Chart1.Series("temp2").Points.Clear()
    Chart1.Series("temp3").Points.Clear()
    Chart1.Series("temp4").Points.Clear()
    Chart2.Series("flow1").Points.Clear()
    Chart2.Series("flow2").Points.Clear()
    Chart3.Series("Energy1").Points.Clear()
    Chart3.Series("Energy2").Points.Clear()
    For i As Integer = 0 To 120
        Chart1.Series("temp1").Points.AddXY(i, arr1(i))
        Chart1.Series("temp2").Points.AddXY(i, arr2(i))
        Chart1.Series("temp3").Points.AddXY(i, arr3(i))
        Chart1.Series("temp4").Points.AddXY(i, arr4(i))
        Chart2.Series("flow1").Points.AddXY(i, arr5(i))
        Chart2.Series("flow2").Points.AddXY(i, arr6(i))
        Chart3.Series("Energy1").Points.AddXY(i, arr7(i))
        Chart3.Series("Energy2").Points.AddXY(i, arr8(i))
    Next
End Sub

```

รูปที่ 3.21 แสดงโปรแกรม Timer

7. การสร้างปุ่มสำหรับการสลับไปมาเพื่อกราฟระหว่าง อุณหภูมิ อัตราการไหล กับ พลังงาน

```

Private Sub btnG1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnG1.Click
    Chart1.Show()
    Chart2.Show()
    Chart3.Hide()
    btnG1.Enabled = False
    btnG2.Enabled = True
End Sub

Private Sub btnG2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnG2.Click
    btnG1.Enabled = True
    btnG2.Enabled = False
    Chart3.Show()
    Chart1.Hide()
    Chart2.Hide()
End Sub
End Class

```

รูปที่ 3.22 แสดงโปรแกรมการสลับไปมาของกราฟ

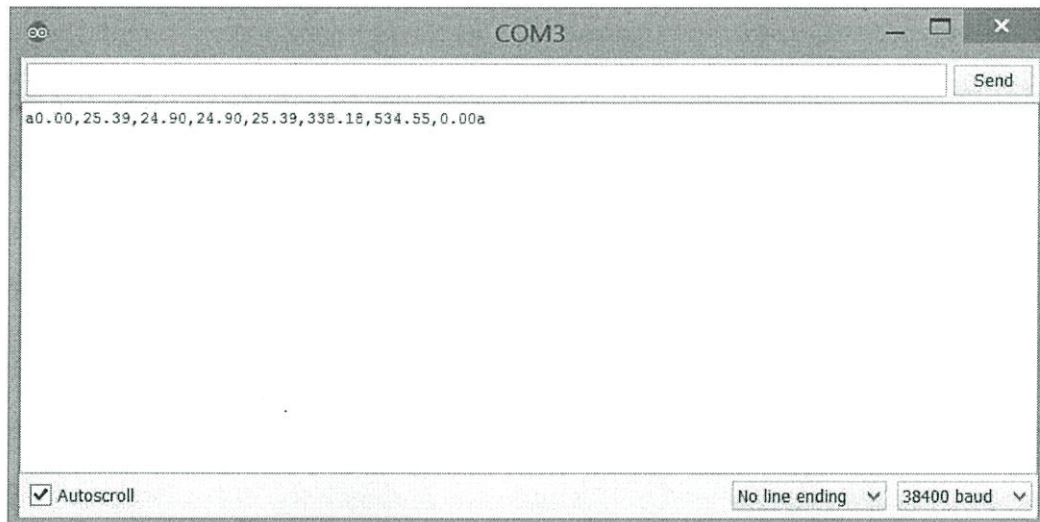
บทที่ 4

ผลการทดลอง

4.1 การแสดงผลของโปรแกรม Arduino

4.1.1 การแสดงผลของการวัดค่าจากอุปกรณ์การวัดผ่าน Serial Monitor ของโปรแกรม Arduino

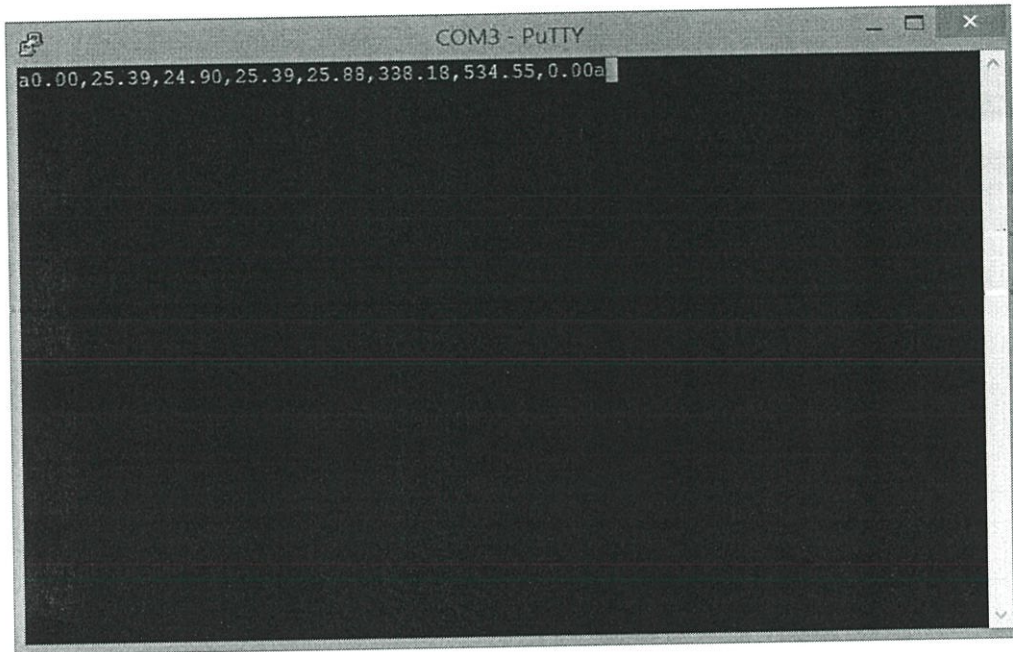
เป็นการแสดงผลของอุณหภูมิจาก Temperature sensor(Lm35)ทั้งหมด 4 ตัว, Turbine flow meter 2 ตัว และ Energy ที่ท่อด้านขาออกและด้านขาเข้า ตามลำดับ โดยจะเป็นการแสดงผลผ่าน Serial Monitor ภายในโปรแกรม Arduino โดยตั้งค่า Baud Rate เป็น 38400 bpm



รูปที่ 4.1 แสดงค่าที่วัดได้ผ่าน Serial Monitor จากโปรแกรม Arduino

4.1.2 การแสดงผลของการวัดค่าจากอุปกรณ์การวัดผ่านโปรแกรม PUTTY

เป็นการแสดงผลของการวัด เช่นเดียวกันกับการแสดงผลผ่าน Serial Monitor ของโปรแกรม Arduino แต่ในที่นี้จะใช้โปรแกรม PUTTY แทน โดยจะแสดงผลการวัดอุณหภูมิ, อัตราการไหล และพลังงานเรียงลำดับ และตั้งค่า Baud Rate เช่นเดียวกันกับโปรแกรม Arduino



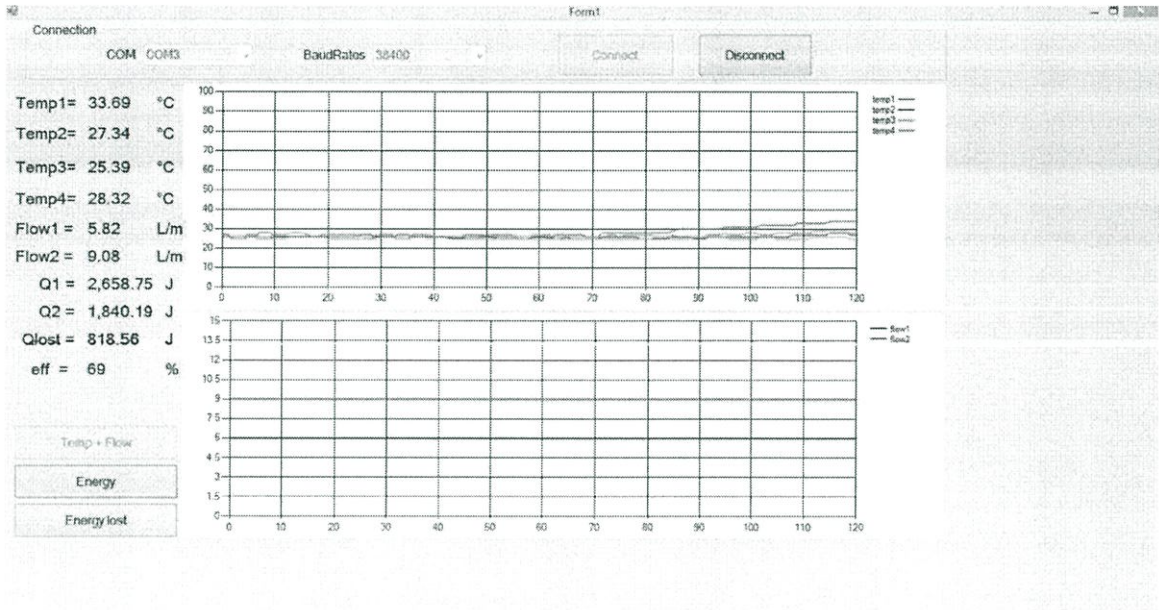
รูปที่ 4.2 การแสดงผลของการวัดผ่านโปรแกรม PUTTY

4.2 การแสดงผลของโปรแกรม Plant Monitoring Arduino

โปรแกรม Plant Monitoring Arduino เป็นโปรแกรมที่เขียนจากซอฟต์แวร์ Microsoft Visual Studio โดยจะแสดงผลของอุณหภูมิ, อัตราการไหล พลังงาน พลังงานสูญเสีย และประสิทธิภาพ ที่วัดจาก Temperature sensor(Lm35) และ Turbine flow meter ตามลำดับ โดยจะแสดงผลในรูปแบบตัวเลขและรูปแบบของกราฟ

4.2.1 การแสดงผล Temperature, Flow Rate ในรูปแบบกราฟเส้นตรงในช่วงของอุณหภูมิ 30 – 40 °C

ในส่วนนี้จะเป็นการแสดงผล Temperature, Flow Rate, Energy, Energy lost และ Efficiency ในรูปแบบของตัวเลขและแสดงผลของ Temperature และ Flow Rate ในรูปแบบของกราฟเส้นตรง ดังรูป



รูปที่ 4.3 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 33.69°C

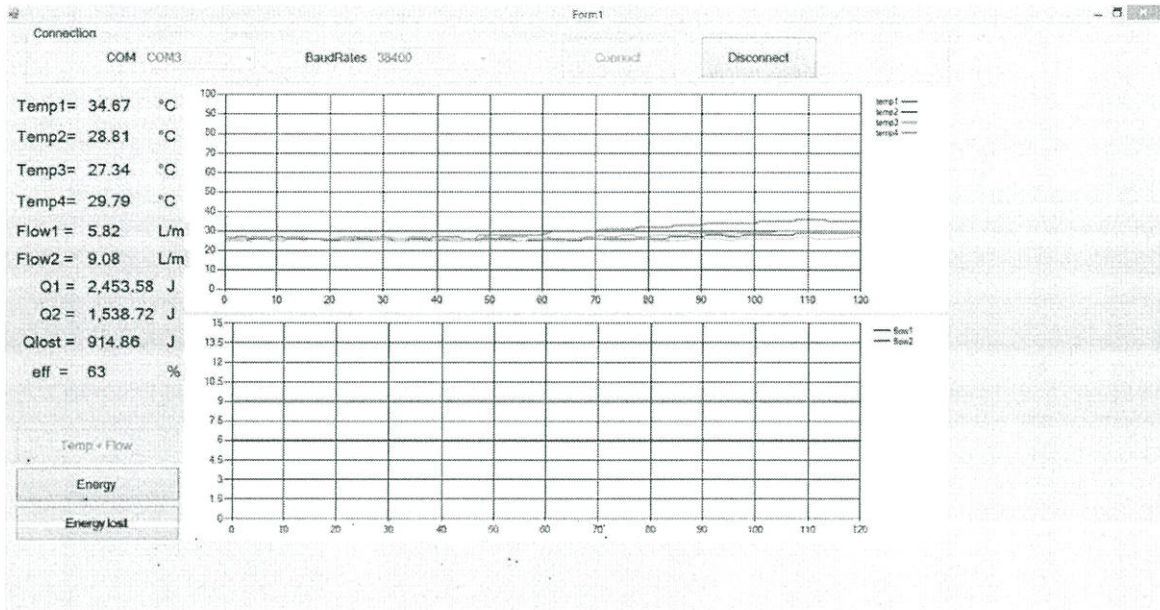
จากรูปที่ 4.3 การคำนวณหา Q_1 และ Q_2 หาได้จากสมการ $Q = m \times C \times (T_o - T_i)$

$$\text{แทนค่า } Q_1 = \frac{5.82}{60} \times 4.187 \times (27.34 - 33.69) \text{ จะได้ค่า } Q_1 = 2658.75 \text{ J}$$

$$\text{แทนค่า } Q_2 = \frac{9.08}{60} \times 4.187 \times (28.32 - 25.39) \text{ จะได้ค่า } Q_2 = 1840.19 \text{ J}$$

$$\text{หา } Q_{\text{lost}} = Q_1 - Q_2 = 2658.75 - 1840.19 \text{ จะได้ค่า } Q_{\text{lost}} = 818.56 \text{ J}$$

$$\text{หา Efficiency} = \frac{Q_2}{Q_1} \times 100\% = \frac{1840.19}{2658.75} \times 100\% \text{ จะได้ค่าประสิทธิภาพ} = 69\%$$



รูปที่ 4.4 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 34.67°C

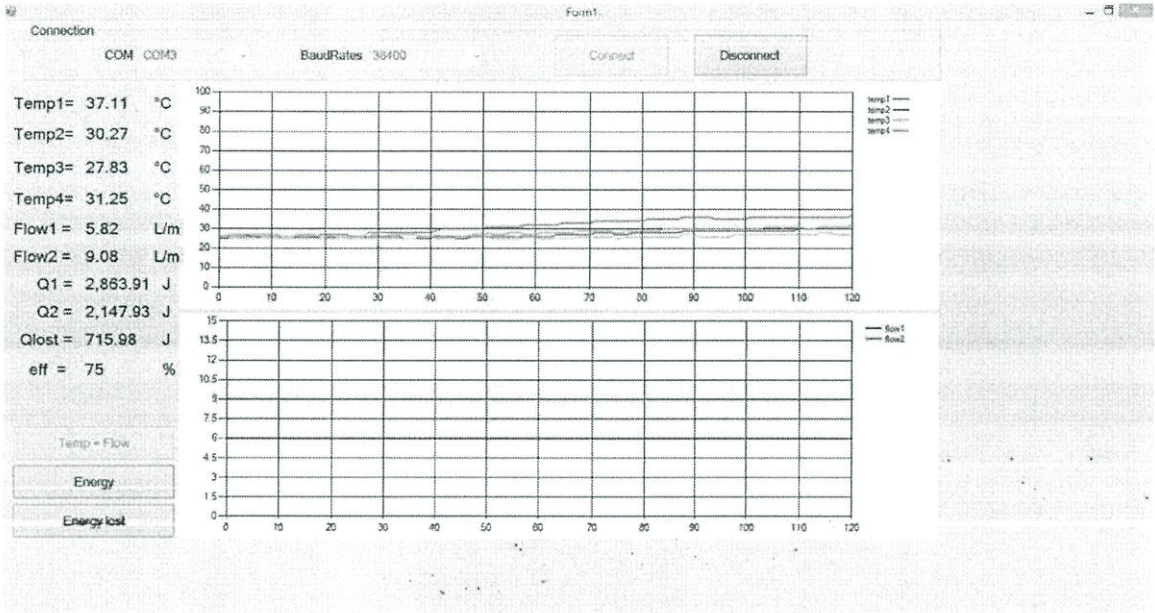
จากรูปที่ 4.4 คำนวณหา Q1 และ Q2 หาได้จากสมการ $Q = m \times C \times (T_o - T_i)$

$$\text{แทนค่า } Q_1 = \frac{5.82}{60} \times 4.187 \times (28.81 - 34.67) \text{ จะได้ค่า } Q_1 = 2453.58 \text{ J}$$

$$\text{แทนค่า } Q_2 = \frac{9.08}{60} \times 4.187 \times (29.79 - 27.34) \text{ จะได้ค่า } Q_2 = 1538.72 \text{ J}$$

$$\text{หา } Q_{\text{lost}} = Q_1 - Q_2 = 2453.58 - 1538.72 \text{ จะได้ค่า } Q_{\text{lost}} = 914.86 \text{ J}$$

$$\text{หา Efficiency} = \frac{Q_2}{Q_1} \times 100\% = \frac{1538.72}{2453.58} \times 100\% \text{ จะได้ค่าประสิทธิภาพ} = 63\%$$



รูปที่ 4.5 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 37.11°C

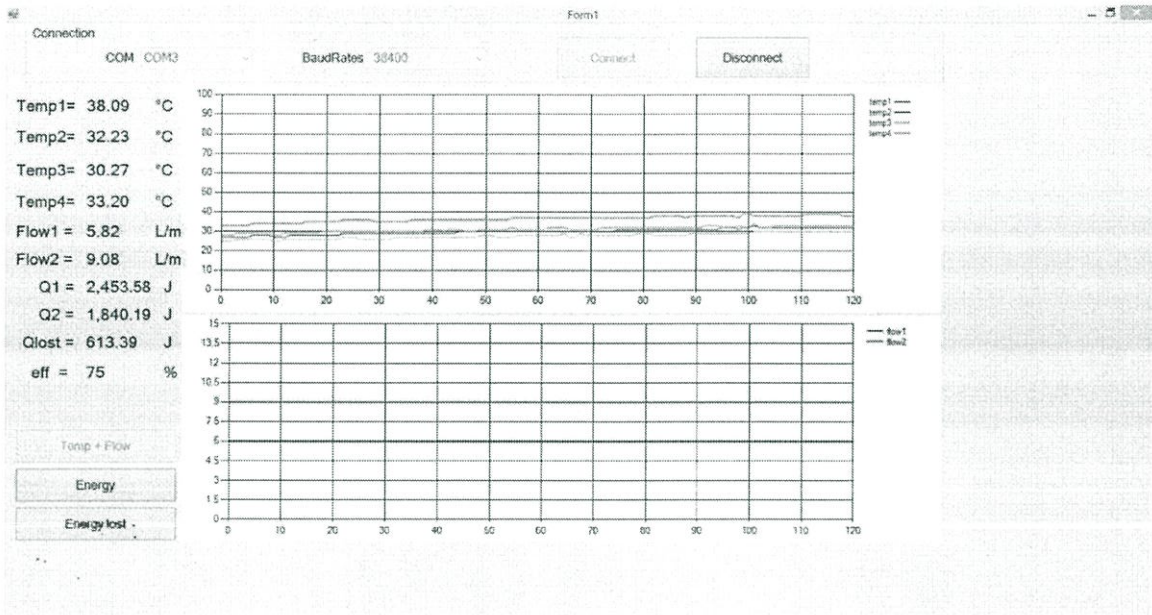
จากรูปที่ 4.5 คำนวนหา Q_1 และ Q_2 หาได้จากสมการ $Q = m \times C \times (T_o - T_i)$

$$\text{แทนค่า } Q_1 = \frac{5.82}{60} \times 4.187 \times (30.27 - 37.11) \text{ จะได้ค่า } Q_1 = 2863.91 \text{ J}$$

$$\text{แทนค่า } Q_2 = \frac{9.08}{60} \times 4.187 \times (31.25 - 27.83) \text{ จะได้ค่า } Q_2 = 2147.93 \text{ J}$$

$$\text{หา } Q_{\text{lost}} = Q_1 - Q_2 = 2863.91 - 2147.93 \text{ จะได้ค่า } Q_{\text{lost}} = 715.98 \text{ J}$$

$$\text{หา Efficiency} = \frac{Q_2}{Q_1} \times 100\% = \frac{2147.93}{2863.91} \times 100\% \text{ จะได้ค่าประสิทธิภาพ} = 75\%$$



รูปที่ 4.6 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 38.09°C

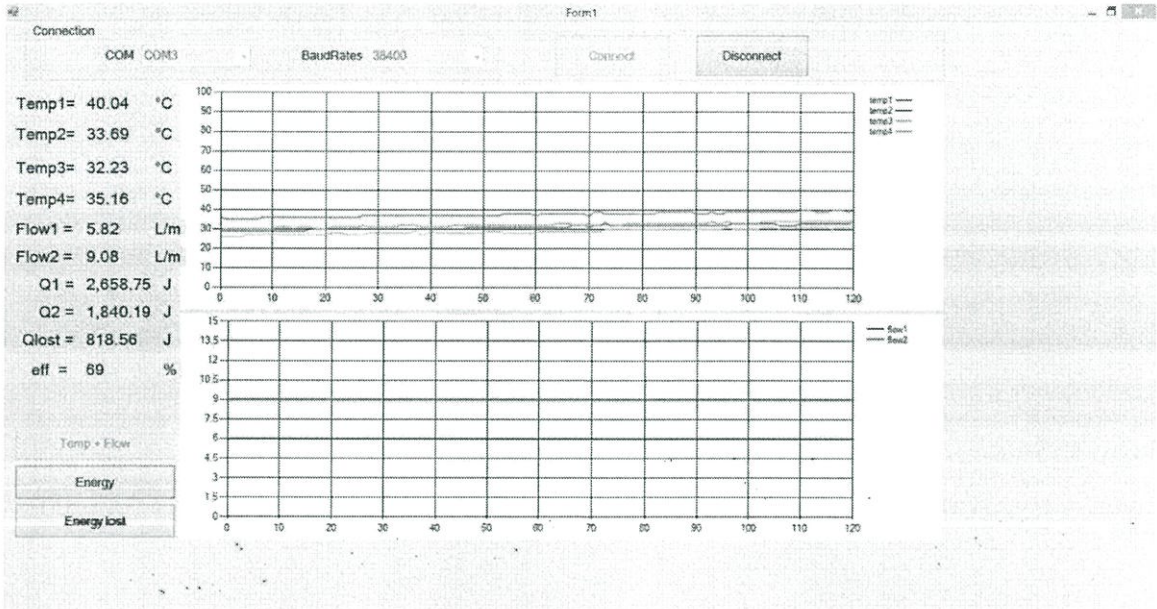
จากรูปที่ 4.6 คำนวณหา Q_1 และ Q_2 หาได้จากสมการ $Q = m \times C \times (T_o - T_i)$

$$\text{แทนค่า } Q_1 = \frac{5.82}{60} \times 4.187 \times (32.23 - 38.09) \text{ จะได้ค่า } Q_1 = 2453.58 \text{ J}$$

$$\text{แทนค่า } Q_2 = \frac{9.08}{60} \times 4.187 \times (33.20 - 30.27) \text{ จะได้ค่า } Q_2 = 1840.19 \text{ J}$$

$$\text{หา } Q_{\text{lost}} = Q_1 - Q_2 = 2453.58 - 1840.19 \text{ จะได้ค่า } Q_{\text{lost}} = 613.39 \text{ J}$$

$$\text{หา Efficiency} = \frac{Q_2}{Q_1} \times 100\% = \frac{2453.58}{1840.19} \times 100\% \text{ จะได้ค่าประสิทธิภาพ} = 75\%$$



รูปที่ 4.7 แสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino ที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 40.04°C

จากรูปที่ 4.7 คำนวนหา Q1 และ Q2 หาได้จากสมการ $Q = m \times C \times (T_o - T_i)$

$$\text{แทนค่า } Q_1 = \frac{5.82}{60} \times 4.187 \times (33.69 - 40.04) \text{ จะได้ค่า } Q_1 = 2658.75 \text{ J}$$

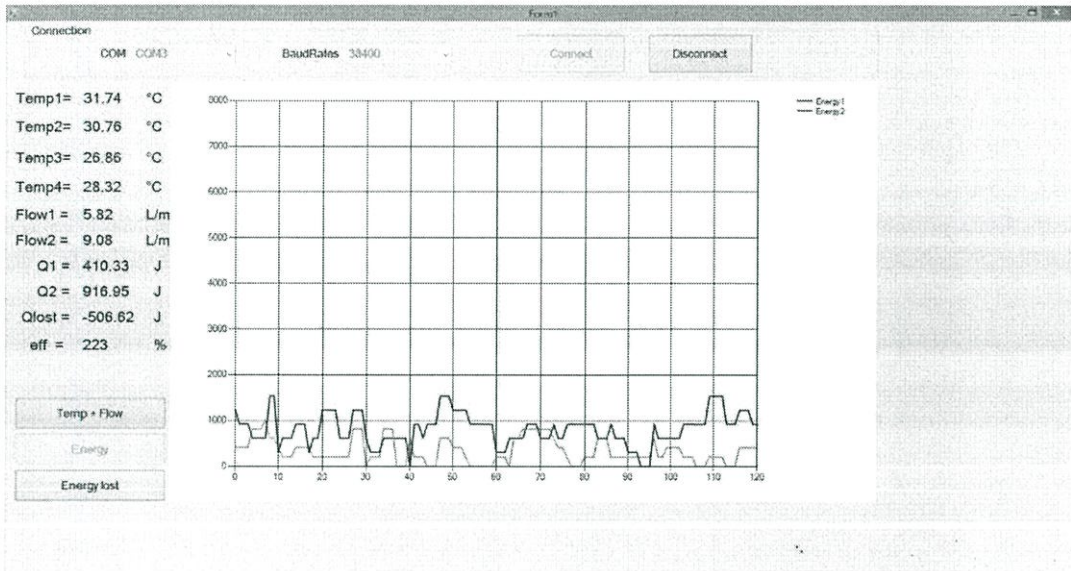
$$\text{แทนค่า } Q_2 = \frac{9.08}{60} \times 4.187 \times (35.16 - 32.23) \text{ จะได้ค่า } Q_2 = 1840.19 \text{ J}$$

$$\text{หา } Q_{\text{lost}} = Q_1 - Q_2 = 2658.75 - 1840.19 \text{ จะได้ค่า } Q_{\text{lost}} = 818.56 \text{ J}$$

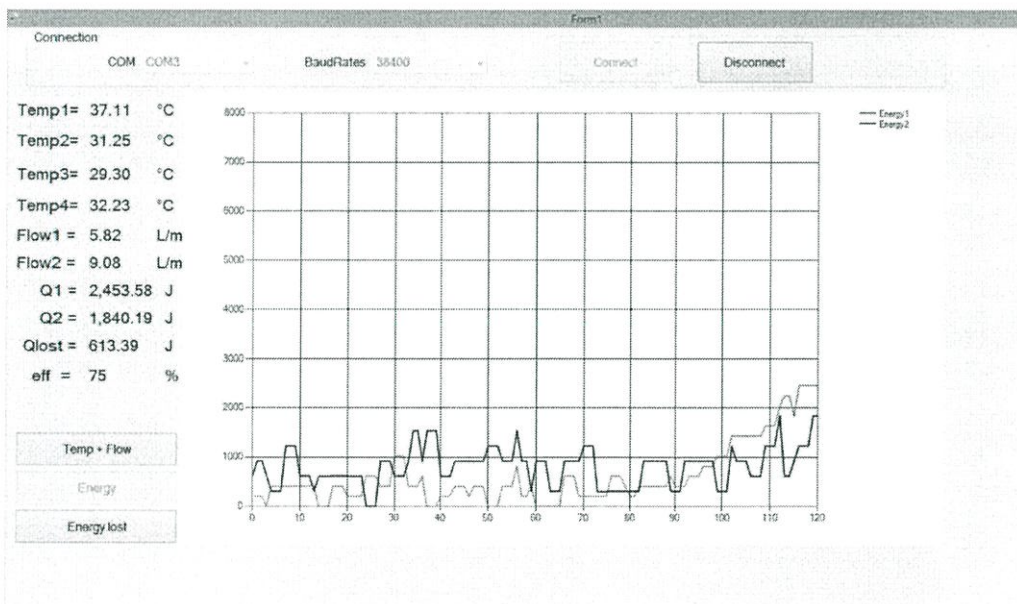
$$\text{หา Efficiency} = \frac{Q_2}{Q_1} \times 100\% = \frac{2658.75}{1840.19} \times 100\% \text{ จะได้ค่าประสิทธิภาพ} = 69\%$$

4.2.2 การแสดงผลของ Energy ในรูปแบบกราฟเส้นตรง

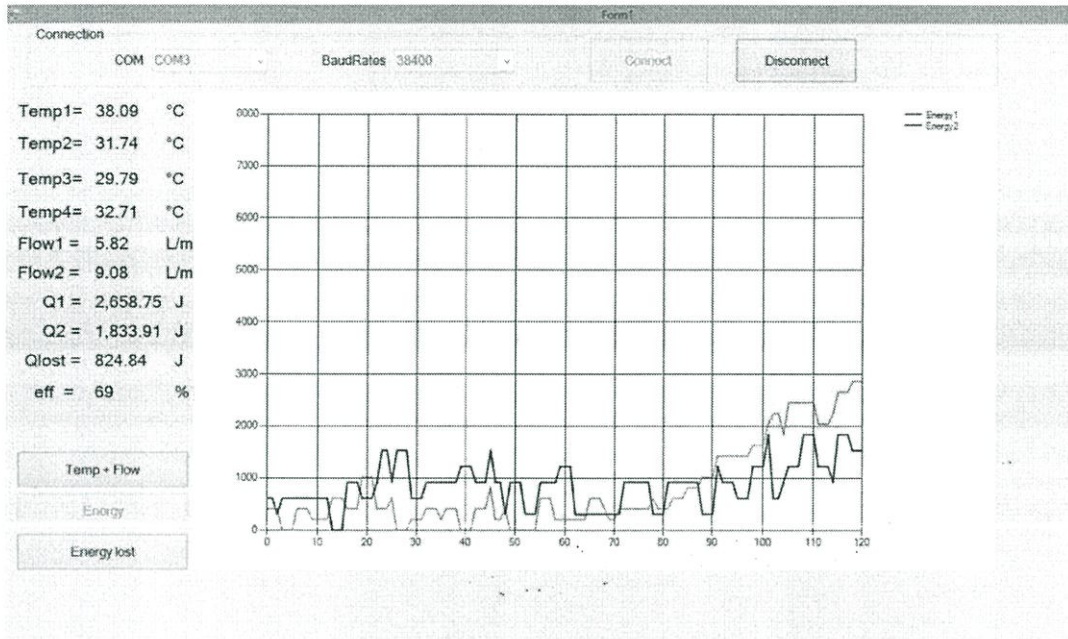
ในส่วนนี้จะเป็นการแสดงผล Temperature, Flow Rate, Energy, Energy lost และ Efficiency ในรูปแบบของตัวเลขและแสดงผลของ Energy ในรูปแบบของกราฟเส้นตรง ดังรูป



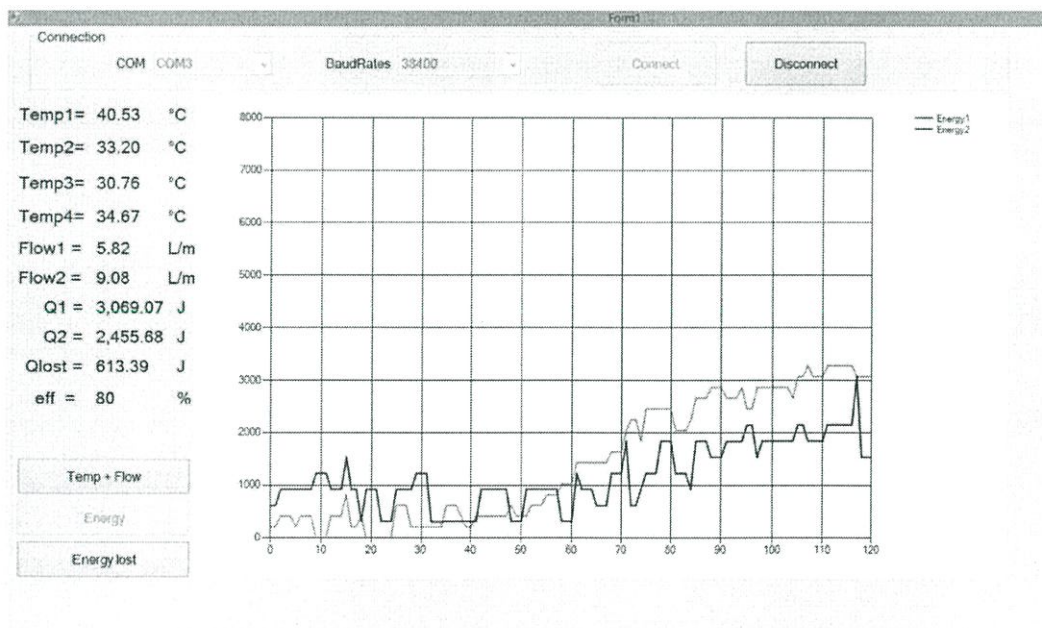
รูปที่ 4.8 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 31.74°C



รูปที่ 4.9 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 37.11°C



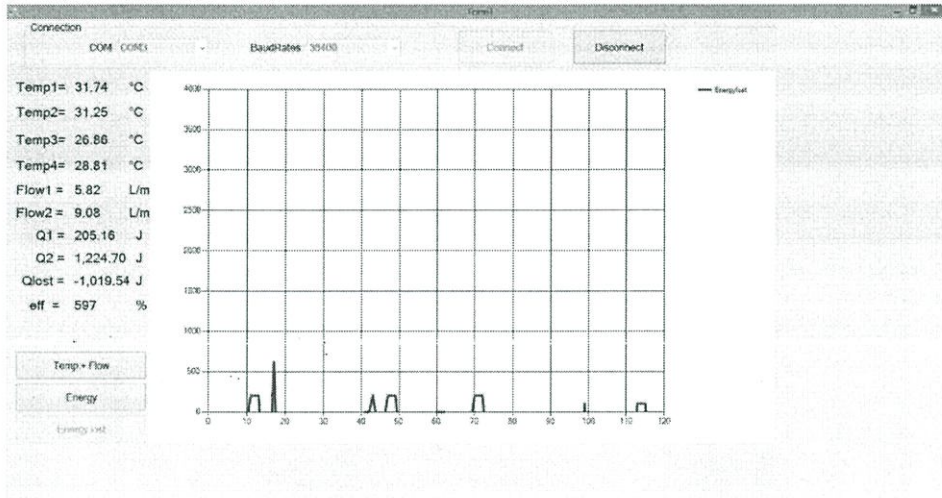
รูปที่ 4.10 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 38.09°C



รูปที่ 4.11 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า Energy ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 40.53°C

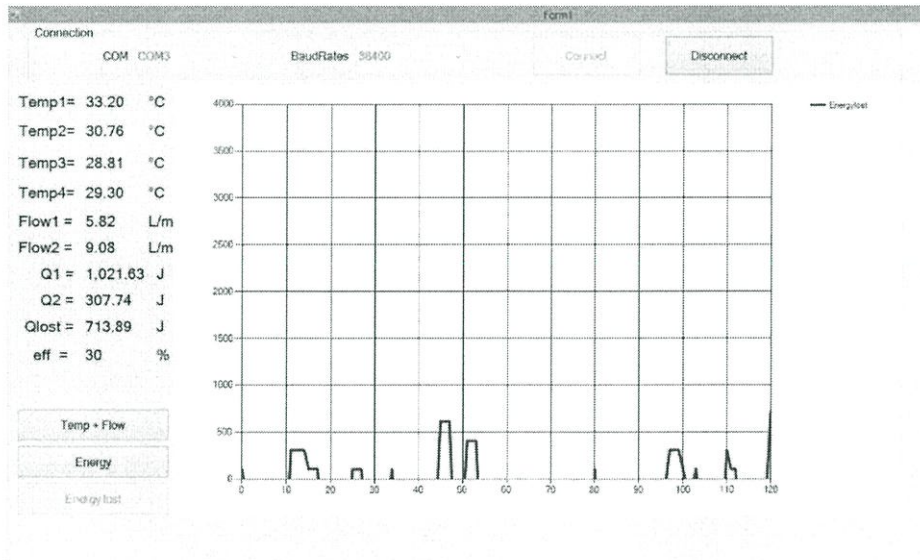
4.2.3 การแสดงผลของ Energy lost ในรูปแบบกราฟเส้นตรง

ในส่วนนี้จะเป็นการแสดงผล Temperature, Flow Rate, Energy, Energy lost และ Efficiency ในรูปแบบของตัวเลขและแสดงผลของ Energy lost ในรูปแบบของกราฟเส้นตรง ดังรูป



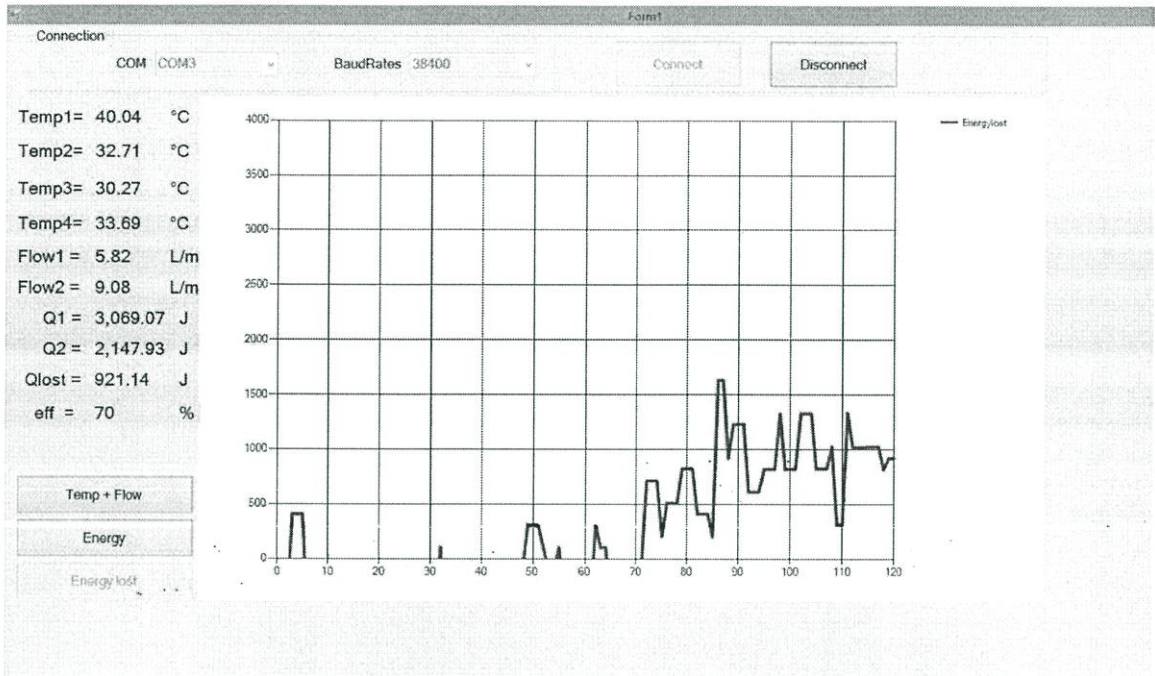
รูปที่ 4.12 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า

Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 31.74°C



รูปที่ 4.13 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า

Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 33.20°C



รูปที่ 4.14 แสดงผลแสดงผลของการวัดผ่านโปรแกรม Plant Monitoring Arduino โดยจะแสดงค่า

Energy lost ในรูปแบบของกราฟเส้นตรงที่อุณหภูมิขาเข้าที่ท่อน้ำร้อน 40.04°C

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการศึกษาการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ Arduino กับคอมพิวเตอร์ และหลักการแลกเปลี่ยนความร้อนของของไหล ซึ่งในโครงการนี้ได้นำมาประยุกต์โดยจัดทำชุดทดลองการแลกเปลี่ยนความร้อนของน้ำขึ้นและวัดค่าอัตราการไหลและอุณหภูมิของน้ำด้วยไมโครคอนโทรลเลอร์ Arduino แล้วนำค่าไปแสดงผลผ่านจอคอมพิวเตอร์ด้วยซอฟต์แวร์ที่เขียนโดย Microsoft Visual Studio พบว่าการจะใช้ไมโครคอนโทรลเลอร์ Arduino วัดและแสดงผลค่าอัตราการไหลและอุณหภูมิของน้ำนั้นจะต้องมี Protocol ที่กำหนดขึ้นเพื่อใช้ในการสื่อสารกันผ่านพอร์ตอนุกรมระหว่างอุปกรณ์วัดและไมโครคอนโทรลเลอร์ Arduino

5.2 ข้อเสนอแนะและแนวทางการพัฒนา

เนื่องจากการแสดงค่าอัตราการไหลและอุณหภูมิของน้ำผ่านทางหน้าจอคอมพิวเตอร์โดยใช้ซอฟต์แวร์ที่เขียนโดย Microsoft Visual basic นั้นเป็นเพียงการแสดงผลเพียงอย่างเดียวไม่สามารถใช้ในการควบคุมการทำงานของอุปกรณ์ต่างๆได้ เช่น ควบคุมกระบวนการทำงานของ Heater เพื่อใช้ในการควบคุมอุณหภูมิ เป็นต้น ดังนั้นหากต้องการควบคุมการทำงานของอุปกรณ์จึงควรศึกษาการส่งข้อมูลผ่านทางพอร์ตอนุกรมเพิ่มเติม หรืออาจพัฒนาหน้าจอแสดงผลให้สามารถติดต่อสื่อสารกับผู้ใช้ได้โดยนำเทคโนโลยี HMI (Human Machine Interface) มาใช้

บรรณานุกรม

- [1] กิตติ ภัคดีวัฒนะกุล. Visual basic 6 ฉบับโปรแกรมเมอร์, กรุงเทพมหานคร: เคทีพี คอมพ์ แอนด์คอนซัลท์, 2545
- [2] 2557.Arduino. “What is Arduino?”. แหล่งที่มา: <http://arduino.cc/en/Guide/Introduction>
14 ธันวาคม 2557
- [3] ผศ.ดร.นวกัทธา หนูนาท. 2557. Food Network Solution.2555 “Turbine flow meter”.
แหล่งที่มา: <http://www.foodnetworksolution.com/wiki/word/7253/turbine-flow-meter>
14 ธันวาคม 2557
- [4] ญัฐกานต์ กันยาประสานกิจ. 2558. เครื่องแลกเปลี่ยนความร้อน (Heat Exchanger). แหล่งที่มา:
http://www.npcse.co.th/npc_date/npc_previews.asp?id_head=11&id_sub=36&id=582.
1 เมษายน 2558
- [5] 2558. บทที่7 เครื่องแลกเปลี่ยนความร้อน (Heat exchangers).
แหล่งที่มา: www.completesengineering.com/style/article_heat%20exchanger1.pdf
1 เมษายน 2558
- [6] 2558. การใช้งานพอร์ตอนุกรม RS232. แหล่งที่มา: <http://www.thaimicrotron.com/CCS-628/Reference/RS232.htm>. 25 มีนาคม 2558
- [7] 2558. การติดต่อสื่อสารผ่านพอร์ตอนุกรม. แหล่งที่มา: panudech.rmutl.ac.th/OPN/Slide03.pdf
25 มีนาคม 2558
- [8] 2558. MCP9701 ไอซีวัดอุณหภูมิให้ผลการทำงานเป็นแรงดันแบบเชิงเส้น บทความฉบับที่ 21.
แหล่งที่มา: <http://www.tpemagazine.com/2009/2013-07-03-11-21-48/89--21/427-mcp9701-.html>. 5 เมษายน 2558
- [9] ธวัช สุธีรังกูร. 2558. หม้อน้ำคืออะไร มีไว้ทำอะไร. แหล่งที่มา:
<http://www.cpsradiator.com/2012/04/blog-post.html>. 27 มีนาคม 2558
- [10] สุรวุฒ อึ้งประเสริฐ. 2558. เรื่องตอนที่ 1 เริ่มต้นรู้จัก Visual Studio.Net และ Visual Basic.Net.
แหล่งที่มา : <http://www.jobpub.com/articles/showarticle.asp?id=6>. 5 เมษายน 2558

ภาคผนวก ก

โปรแกรมที่ใช้กับ Arduino และ Microsoft visual studio

1.การประกาศตัวแปรเริ่มต้นสำหรับโปรแกรม Arduino

```
volatile int spfan1;  
volatile int spfan2;  
int fpin1=2;  
int fpin2=3;  
float data[10];
```

2.การสร้างฟังก์ชันสำหรับการวัดอุณหภูมิ, อัตราการไหล และ พลังงาน สำหรับ Arduino

```
void rpm1 ()  
{  
    spfan1++;  
}  
void rpm2 ()  
{  
    spfan2++;  
}  
void setup()  
{pinMode(fpin1, INPUT);  
    pinMode(fpin2, INPUT);  
    attachInterrupt(0, rpm1, RISING);  
    attachInterrupt(1, rpm2, RISING);
```

```
Serial.begin(9600);}

void loop()
{
    data[0]=analogRead(A0)*0.48828125;
    data[1]=analogRead(A1)*0.48828125;
    data[2]=analogRead(A2)*0.48828125;
    data[3]=analogRead(A3)*0.48828125;

    spfan1=0;
    spfan2=0;

    sei();

    delay (1000);

    cli();

    data[4] = (spfan1 / 330);
    data[5] = (spfan2 / 330);

    data[6] = 4187*data[4]*abs(data[1]-data[0]);
    data[7] = 4187*data[5]*abs(data[3]-data[2]);

    Serial.print("a");

    Serial.print(data[0]);

    Serial.print(",");

    Serial.print(data[1]);

    Serial.print(",");

    Serial.print(data[2]);

    Serial.print(",");
```

```

        Serial.print(data[3]);

        Serial.print(",");

        Serial.print(data[4]);

        Serial.print(",");

        Serial.print(data[5]);

        Serial.print(",");

        Serial.print(data[6]);

        Serial.print(",");

        Serial.print(data[7]);

        Serial.print("a");

    }

```

3. การสร้างโปรแกรม Plant Monitoring Arduino จากโปรแกรม Microsoft Visual Studio

```
Imports System.IO.Ports
```

```
Imports System
```

```
Public Class Form1
```

```
    Dim x As Integer = 0
```

```
    Dim arr1(120), arr2(120), arr3(120), arr4(120), arr5(120), arr6(120), arr7(120), arr8(120) As Integer
```

```
    Private comBuffer As Byte()
```

```
    Private Delegate Sub UpdateFormDelegate()
```

```
    Private UpdateFormDelegate1 As UpdateFormDelegate
```

```
    Public Delegate Sub StringSubPointer(ByVal Buffer As String)
```

```
    Dim WithEvents sp As New SerialPort
```

```
    Private Sub GetSerialPortNames()
```

```
For Each sport As String In My.Computer.Ports.SerialPortNames
```

```
    cmbPort.Items.Add(sport)
```

```
Next
```

```
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles MyBase.Load
```

```
    Dim BaudRates() As String = {"600", "1200", "2400", "4800", "9600", "14400", "19200", "28800",  
    "38400", "57600", "115200"}
```

```
    sp.Parity = IO.Ports.Parity.None
```

```
    sp.DataBits = 8
```

```
    sp.StopBits = IO.Ports.StopBits.One
```

```
    sp.ReadTimeout = 20000
```

```
    sp.Handshake = Handshake.None
```

```
    btnG1.Enabled = False
```

```
    Chart3.Hide()
```

```
    Chart1.ChartAreas("ChartArea1").AxisX.Minimum = 0
```

```
    Chart1.ChartAreas("ChartArea1").AxisX.Maximum = 120
```

```
    Chart1.ChartAreas("ChartArea1").AxisX.Interval = 10
```

```
    Chart1.ChartAreas("ChartArea1").AxisY.Minimum = 0
```

```
    Chart1.ChartAreas("ChartArea1").AxisY.Maximum = 100
```

```
    Chart1.ChartAreas("ChartArea1").AxisY.Interval = 10
```

```
    Chart2.ChartAreas("ChartArea1").AxisX.Minimum = 0
```

```
Chart2.ChartAreas("ChartArea1").AxisX.Maximum = 120
```

```
Chart2.ChartAreas("ChartArea1").AxisX.Interval = 10
```

```
Chart2.ChartAreas("ChartArea1").AxisY.Minimum = 0
```

```
Chart2.ChartAreas("ChartArea1").AxisY.Maximum = 15
```

```
Chart2.ChartAreas("ChartArea1").AxisY.Interval = 1.5
```

```
Chart3.ChartAreas("ChartArea1").AxisX.Minimum = 0
```

```
Chart3.ChartAreas("ChartArea1").AxisX.Maximum = 120
```

```
Chart3.ChartAreas("ChartArea1").AxisX.Interval = 10
```

```
Chart3.ChartAreas("ChartArea1").AxisY.Minimum = 0
```

```
Chart3.ChartAreas("ChartArea1").AxisY.Maximum = 8000
```

```
Chart3.ChartAreas("ChartArea1").AxisY.Interval = 1000
```

```
cmbBaud.Items.AddRange(BaudRates)
```

```
cmbBaud.SelectedIndex = 8
```

```
Try
```

```
    GetSerialPortNames()
```

```
    cmbPort.SelectedIndex = 0
```

```
Catch
```

```
    MsgBox("No ports connected.")
```

```
End Try
```

```
For i As Integer = 0 To 120
```

```
    arr1(i) = 0
```

```
    arr2(i) = 0
```

```
arr3(i) = 0
```

```
arr4(i) = 0
```

```
arr5(i) = 0
```

```
arr6(i) = 0
```

```
arr7(i) = 0
```

```
arr8(i) = 0
```

```
Chart1.Series("temp1").Points.AddXY(i, arr1(i))
```

```
Chart1.Series("temp2").Points.AddXY(i, arr2(i))
```

```
Chart1.Series("temp3").Points.AddXY(i, arr3(i))
```

```
Chart1.Series("temp4").Points.AddXY(i, arr4(i))
```

```
Chart2.Series("flow1").Points.AddXY(i, arr5(i))
```

```
Chart2.Series("flow2").Points.AddXY(i, arr6(i))
```

```
Chart3.Series("Energy1").Points.AddXY(i, arr7(i))
```

```
Chart3.Series("Energy2").Points.AddXY(i, arr8(i))
```

```
Next
```

```
End Sub
```

```
Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnConnect.Click
```

```
Try
```

```
sp.BaudRate = cmbBaud.SelectedItem.ToString
```

```
sp.PortName = cmbPort.SelectedItem.ToString
```

```
sp.Open()
```

```
If sp.IsOpen Then
```

```
    Timer1.Interval = 1000
```

```
    Timer1.Start()
```

```
    btnConnect.Enabled = False
```

```
    btnDisconnect.Enabled = True
```

```
    cmbPort.Enabled = False
```

```
    cmbBaud.Enabled = False
```

```
Else : Timer1.Stop()
```

```
End If
```

```
Catch
```

```
    btnDisconnect.Enabled = True
```

```
    sp.Close()
```

```
End Try
```

```
End Sub
```

```
Private Sub Receiver(ByVal sender As Object, _
```

```
    ByVal e As SerialDataReceivedEventArgs) Handles sp.DataReceived
```

```
    Me.BeginInvoke(New StringSubPointer(AddressOf Display), sp.ReadExisting)
```

```
End Sub
```

```
Private Sub Display(ByVal Buffer As String)
```

```
    Dim strArr() As String
```

```
Dim dataArr() As String
```

```
Dim e1, e2, f1, f2 As String
```

```
For i As Integer = 1 To Buffer.Length - 1
```

```
Next
```

```
strArr = Buffer.Split({"a"}, StringSplitOptions.RemoveEmptyEntries)
```

```
If strArr.Length > 0 Then
```

```
    dataArr = strArr(strArr.Length - 1).Split({","}, StringSplitOptions.RemoveEmptyEntries)
```

```
    If dataArr.Length = 8 Then
```

```
        f1 = dataArr(5) / 60
```

```
        f2 = dataArr(6) / 60
```

```
        f1 = FormatNumber(f1, 2)
```

```
        f2 = FormatNumber(f2, 2)
```

```
        dataArr(5) = dataArr(5) / 3600
```

```
        dataArr(5) = FormatNumber(dataArr(5), 2)
```

```
        dataArr(6) = dataArr(6) / 3600
```

```
        dataArr(6) = FormatNumber(dataArr(6), 2)
```

```
        e1 = 4187 * dataArr(5) * Math.Abs(dataArr(2) - dataArr(1))
```

```
        e2 = 4187 * dataArr(6) * Math.Abs(dataArr(4) - dataArr(3))
```

```
        e1 = FormatNumber(e1, 2)
```

```
        e2 = FormatNumber(e2, 2)
```

```
Label15.Text = dataArr(1)
```

```
Label16.Text = dataArr(2)
```

```
Label17.Text = dataArr(3)
```

```
Label18.Text = dataArr(4)
```

```
Label19.Text = f1
```

```
Label20.Text = f2
```

```
Label21.Text = e1
```

```
Label22.Text = e2
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnDisconnect.Click
```

```
Try
```

```
sp.Close()
```

```
Timer1.Stop()
```

```
btnConnect.Enabled = True
```

```
btnDisconnect.Enabled = False
```

```
cmbPort.Enabled = True
```

```
cmbBaud.Enabled = True
```

```
Exit Sub
```

```
Catch
```

```
    MessageBox.Show("Some kind of problem.")
```

```
End Try
```

```
End Sub
```

```
Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As  
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
```

```
    If sp.IsOpen() Then
```

```
        MessageBox.Show("Disconnect before closing")
```

```
        e.Cancel = True
```

```
    End If
```

```
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Timer1.Tick
```

```
    For i As Integer = 0 To 119
```

```
        arr1(i) = arr1(i + 1)
```

```
        arr2(i) = arr2(i + 1)
```

```
        arr3(i) = arr3(i + 1)
```

```
        arr4(i) = arr4(i + 1)
```

```
        arr5(i) = arr5(i + 1)
```

```
        arr6(i) = arr6(i + 1)
```

```
        arr7(i) = arr7(i + 1)
```

```
        arr8(i) = arr8(i + 1)
```

```
    Next
```

```
arr1(120) = Label15.Text
```

```
arr2(120) = Label16.Text
```

```
arr3(120) = Label17.Text
```

```
arr4(120) = Label18.Text
```

```
arr5(120) = Label19.Text
```

```
arr6(120) = Label20.Text
```

```
arr7(120) = Label21.Text
```

```
arr8(120) = Label22.Text
```

```
Chart1.Series("temp1").Points.Clear()
```

```
Chart1.Series("temp2").Points.Clear()
```

```
Chart1.Series("temp3").Points.Clear()
```

```
Chart1.Series("temp4").Points.Clear()
```

```
Chart2.Series("flow1").Points.Clear()
```

```
Chart2.Series("flow2").Points.Clear()
```

```
Chart3.Series("Energy1").Points.Clear()
```

```
Chart3.Series("Energy2").Points.Clear()
```

```
For i As Integer = 0 To 120
```

```
    Chart1.Series("temp1").Points.AddXY(i, arr1(i))
```

```
    Chart1.Series("temp2").Points.AddXY(i, arr2(i))
```

```
    Chart1.Series("temp3").Points.AddXY(i, arr3(i))
```

```
    Chart1.Series("temp4").Points.AddXY(i, arr4(i))
```

```
    Chart2.Series("flow1").Points.AddXY(i, arr5(i))
```

```
    Chart2.Series("flow2").Points.AddXY(i, arr6(i))
```

```
Chart3.Series("Energy1").Points.AddXY(i, arr7(i))
```

```
Chart3.Series("Energy2").Points.AddXY(i, arr8(i))
```

```
Next
```

```
End Sub
```

```
Private Sub btnG1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles btnG1.Click
```

```
Chart1.Show()
```

```
Chart2.Show()
```

```
Chart3.Hide()
```

```
btnG1.Enabled = False
```

```
btnG2.Enabled = True
```

```
End Sub
```

```
Private Sub btnG2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles btnG2.Click
```

```
btnG1.Enabled = True
```

```
btnG2.Enabled = False
```

```
Chart3.Show()
```

```
Chart1.Hide()
```

```
Chart2.Hide()
```

```
End Sub
```

```
End Class
```