

การออกแบบเครื่องควบคุมอุณหภูมิสำหรับถังปฏิกริยาเคมีแบบความดันสูง
Design of Temperature controller for High Pressure Chemical Reactor



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเครื่องควบคุมอุณหภูมิสำหรับถังปฏิกริยาเคมีแบบความดันสูง
Design of Temperature controller for High Pressure Chemical Reactor



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Design of Temperature controller for High Pressure Chemical Reactor

Mr.Tanakorn Kiriyaadee

Mr.Krudsada Tharatheo

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
Academic Year 2019

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ	การออกแบบเครื่องควบคุมอุณหภูมิสำหรับถังปฏิริยาเคมีแบบความดันสูง Design of Temperature controller for High Pressure Chemical Reactor		
นักศึกษาผู้จัดทำ	นายธนากร	กิริยาดี	รหัสนักศึกษา 59010608
	นายกฤษฎา	ถาวรโท	รหัสนักศึกษา 59010038
อาจารย์ที่ปรึกษาโครงการ	รองศาสตราจารย์ ดร.อาจันต์	น่วมสำราญ	
	รองศาสตราจารย์ ดร.วิทยา	ทิพย์สุวรรณพร	
ปีการศึกษา	2562		

บทคัดย่อ

โครงการนี้มีจุดมุ่งหมายในการศึกษาเกี่ยวกับการควบคุมอุณหภูมิของถังผสมสารเคมี และการจำลองการทำงานของระบบควบคุมอุณหภูมิถังผสมสารเคมี เพื่อหาค่าของตัวแปรต่างๆ ที่จะทำให้กระบวนการควบคุมการทำงานจากระบบให้มีประสิทธิภาพมากที่สุด เนื่องจากถังผสมสารเคมีที่ทำการศึกษามีลักษณะเป็นสองถังซ้อนกัน โดยถังภายนอกจะบรรจุเหลวชนิดหนึ่งไว้ ส่วนถังภายในจะมีไว้ใส่สารเคมีที่ต้องการจะผสมกันแล้วมีฝาปิดที่มิดชิดเพื่อให้กระบวนการเป็นระบบปิด ซึ่งมีมอเตอร์หมุนถังด้านในให้ผสมสารเข้าด้วยกัน ดังนั้นเพื่อให้ง่ายต่อการวัดอุณหภูมิกระบวนการนี้จึงจำเป็นต้องติดตั้งชุดวัดอุณหภูมิต่อกับไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อกับอุปกรณ์อื่นๆ แบบไร้สายไว้กับตัวถังด้านใน แล้วส่งค่าที่วัดได้ผ่านเครือข่ายเพื่อนำมาประมวลผลและควบคุมโดยใช้การควบคุมแบบพีไอดีในอุปกรณ์ราสเบอร์รี่พาย พร้อมทั้งยังสามารถแสดงผลและควบคุมกระบวนการผ่านหน้าจอแสดงผล

Project title Design of Temperature controller for High Pressure Chemical Reactor
Authors Mr.Tanakorn Kiriyadee
Mr.Kridsada Tharatheo
Thesis Advisors Assoc.Prof.Dr. Arjin Numsomran
Assoc.Prof.Dr. Wittaya Tipsuwannaporn
Year 2019

Abstract

This project objective to study about the control of chemical mixing tanks and the simulation of the system to control the system to be more efficient. Most of the chemicals in the study are characterized as two tanks stacked together by the tank quickly. Put the chemicals that need to be mixed together and then closed completely to be a system used to mix chemicals together to make it easy. Per installation Other wireless peripherals that can be trusted with the inner body and send the measured values via the machine The results are controlled and controlled using the PID control in the Raspberry Pi device, and can be displayed and controlled via the HMI

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอขอบพระคุณ รองศาสตราจารย์อาจินต์ น่วมสำรา และ รองศาสตราจารย์วิทยา ทิพย์สุวรรณพร ที่ให้ความอนุเคราะห์รับเป็นอาจารย์ที่ปรึกษา ให้คำปรึกษา แนะนำ และ ให้การช่วยเหลือในการทำวิทยานิพนธ์ รวมถึงสอนวิธีการทำงานที่ดีทำให้ผู้วิจัยรู้จักพัฒนาตนเองในด้านการงาน ให้ดียิ่งขึ้น

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ที่ให้คำแนะนำและแนวทางในการปรับปรุงวิทยานิพนธ์ให้มีความสมบูรณ์มากขึ้น ขอขอบคุณคณะอาจารย์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่สั่งสอนวิชาความรู้ถ่ายทอดประสบการณ์ต่าง ๆ และนำความรู้เหล่านั้นมาใช้ในการทำวิจัยนี้

คณะผู้จัดทำ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญของปริภูมยานิพนธ์	1
1.2 วัตถุประสงค์ของปริภูมยานิพนธ์	1
1.3 ขอบเขตของปริภูมยานิพนธ์	2
1.4 ขั้นตอนการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 แบบจำลองทางคณิตศาสตร์	3
2.2 Curve fitting.....	5
2.2.1 Curve fitting โดยใช้เมทแลป	6
2.2.2 Curve Fitting Toolbox.....	8
2.2.3 Optimization Toolbox	10
2.3 ระบบควบคุมพีไอดี.....	11
2.3.1 ผลของสัดส่วน	12
2.3.2 ผลของปริพันธ์	12
2.3.3 ผลของอนุพันธ์	14
บทที่ 3 การประยุกต์ใช้ฮาร์ดแวร์และซอฟต์แวร์	15
3.1 ส่วนประกอบของชุดควบคุมอุณหภูมิ.....	15
3.1.1 ถังสำหรับการผสมสารเคมี	15
3.1.2 เตาให้ความร้อน Hot Plate	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3	Stirrer	15
3.1.4	Raspberry pi 4 Model B	16
3.1.5	Arduino UNO	18
3.1.6	Safety valve	20
3.1.7	Termocople type K.....	21
3.1.8	Solid State Relay Industrial, 1-Phase Analog Switching RM1E60AA25.....	22
3.1.9	RTD ชนิด PT100.....	22
3.1.10	Arduino Ethernet Shield	24
3.1.11	NodeMCU	27
3.2	HMI.....	34
บทที่ 4	จำลองระบบควบคุมอุณหภูมิสำหรับถังผสมสารเคมี	48
4.1	ระบบควบคุมอุณหภูมิ	48
4.2	ผลของการปรับค่าพีไอดีในโคดีซีดีเทียบกับแมทแลป	49
4.2.1	โครงสร้างการทำงานของตัวควบคุมพีไอดีในแมทแลป	51
4.2.2	โปรแกรมสำหรับจำลองใน CODESYS	53
4.2.3	โปรแกรมสำหรับจำลองใน MATLAB	54
4.2.4	การควบคุม แบบ P.....	54
4.2.5	การควบคุม แบบ PD.....	56
4.2.6	การควบคุม แบบ PI.....	57
4.2.7	การควบคุม แบบ PID	58
4.3	การหาโมเดลทางคณิตศาสตร์	59
4.3.1	วิธี system identification.....	60
4.3.2	วิธี Curve fitting.....	63
4.4	การจำลองการควบคุมระบบโดยใช้ตัวควบคุมพีไอดี.....	65
4.5	ทดสอบการสื่อสารแบบมอดบัสของระบบ.....	66
4.5.1	มอดบัสบนโปรแกรมโคดีซีดี.....	67
4.5.2	MODBUS on Arduino	74
4.5.3	MODBUS on NodeMCU	79
4.3	สรุป.....	85
บทที่ 5	การควบคุมอุณหภูมิในระบบจริง.....	86

5.1 การควบคุมอุณหภูมิในระบบจริง	86
5.2 อุณหภูมิของ reactor tank.....	90
5.3 สรุป.....	92
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ	93
6.1 สรุปผล	93
บรรณานุกรม.....	95
ภาคผนวก	98
ภาคผนวก ก.....	99
ภาคผนวก ข.....	100
ภาคผนวก ค.....	105



สารบัญตาราง

ตารางที่	หน้า
3. 1 เปรียบเทียบสเปกของ raspberry pi โมเดล A และ B.....	16
4. 1 ค่าพารามิเตอร์ของตัวควบคุม PID ที่ได้จากการจำลองในโปรแกรม MATLAB	66
4. 2 ผลที่ได้จากการทดสอบการสื่อสารระหว่าง Raspberry pi และ Arduino	79
4. 3 แสดงวิธีการส่งการส่งและคำสั่งModbus.....	81
4. 4 ผลที่ได้จากการทดสอบการสื่อสารระหว่าง Raspberry pi และ NodeMCU.....	84
5. 1 ตารางแสดงค่าพารามิเตอร์ในการควบคุมระบบแบบพีไอดี.....	89



สารบัญรูป

รูปที่	หน้า
2. 1 หน้าต่าง Curve Fitting.....	8
2. 2 หน้าต่างที่กำหนดค่าเอง.....	9
2. 3 การตอบสนองต่อการเปลี่ยนแปลงขั้นสำหรับตัวควบคุม PID	13
2. 4 การใช้งานอินทิกรัลเป็นการปรับอัตโนมัติ.....	14
3. 1 Stirrer.....	15
3. 2 บอร์ด raspberry pi.....	18
3. 3 Arduino UNO	19
3. 4 Safety valve.....	20
3. 5 Termocople type K.....	21
3. 6 RTD ชนิด PT100.....	23
3. 7 Arduino Ethernet Shield.....	24
3. 8 รูปแสดงขาที่ใช้สื่อสารบนบอร์ด Arduino.....	26
3. 9 NodeMCU (รูปภาพจาก https://components101.com/).....	29
3. 10 NodeMCU Pinout (รูปภาพจาก https://components101.com/).....	29
3. 11 Preferences.....	31
3. 12 หน้าต่าง Preferences	31
3. 13 Board Manager.....	32
3. 14 หน้าต่าง Board Manager	32
3. 15 เลือกบอร์ด.....	32
3. 16 เลือก port.....	33
3. 17 คอมไพล์ และอัปโหลดโปรแกรม.....	33
3. 18 การเพิ่ม Visualization.....	35
3. 19 ไดอะล็อกVisualization	35
3. 20 หน้าต่าง Visualization.....	36
3. 21 กล่องเครื่องมือ Visualization.....	37
3. 22 เมนูปรับขนาดและตำแหน่ง.....	39
3. 23 เมนูตั้งค่าสีแบบคงที่.....	40

สารบัญรูป(ต่อ)

รูปที่	หน้า
3. 24 เมนูตั้งค่าสีแบบไล่ระดับ.....	41
3. 25 หน้าต่างตั้งค่าสีแบบไล่ระดับ.....	41
3. 26 แทรก Image Pool	42
3. 27 การเพิ่มรูปไปยัง Image Pool	42
3. 28 กำหนดค่าคุณสมบัติ Image Pool.....	43
3. 29 พื้นหลัง Visualization	43
3. 30 Trace	44
3. 31 หน้าต่างเพิ่มตัวแปรและตั้งค่า Trace.....	45
3. 32 หน้าต่างตั้งค่า Display trace.....	45
3. 33 แสดงการเชื่อมต่ออุปกรณ์ต่างของระบบควบคุมอุณหภูมิ.....	47
4. 1 ระบบควบคุมอุณหภูมิสำหรับถังผสมสารเคมี	48
4. 2 การออกแบบที่ใช้สำหรับควบคุมอุณหภูมิ.....	49
4. 3 ฟังก์ชันบล็อก PID ในโปรแกรม CODESYS.....	50
4. 4 ไลบรารี (ซ้าย) และหน้าต่างตั้งค่าของฟังก์ชันบล็อก PID (ขวา) ในโปรแกรม MATLAB.....	51
4. 5 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID ในโปรแกรม MATLAB.....	52
4. 6 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID ในโปรแกรม CODESYS.....	53
4. 7 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PD ในโปรแกรม CODESYS.....	53
4. 8 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID ในโปรแกรม MATLAB.....	54
4. 9 แสดงผลของการควบคุม P เมื่อมี setpoint = 5, Kp = 1	54
4. 10 แสดงผลของการควบคุม P เมื่อมี setpoint = 5, Kp = 10.....	55
4. 11 แสดงผลของการควบคุม P เมื่อมี setpoint = 5, Kp = 90.....	55
4. 12 แสดงผลของการควบคุม P เมื่อมี setpoint = 5, Kp = 99.....	55
4. 13 แสดงผลของการควบคุม PD เมื่อมี setpoint = 5, Kp = 10, Tv = 0.1	56
4. 14 แสดงผลของการควบคุม PD เมื่อมี setpoint = 5, Kp = 10, Tv = 0.2	56
4. 15 แสดงผลของการควบคุม PD เมื่อมี setpoint = 5, Kp = 10, Tv = 0.5	56
4. 16 แสดงผลของการควบคุม PI เมื่อมี setpoint = 5, Kp = 10, Tn = 0.5.....	57
4. 17 แสดงผลของการควบคุม PI เมื่อมี setpoint = 5, Kp = 10, Tn = 1.....	57

IX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4. 18 แสดงผลของการควบคุม PI เมื่อมี setpoint = 5, Kp = 10, Tn = 10.....	57
4. 19 แสดงผลของการควบคุม PID เมื่อมี setpoint = 5, Kp = 10, Tn = 1, Tv = 0.005.....	58
4. 20 แสดงผลของการควบคุม PID เมื่อมี setpoint = 5, Kp = 10, Tn = 1, Tv = 0.02.....	58
4. 21 แสดงผลของการควบคุม PID เมื่อมี setpoint = 5, Kp = 10, Tn = 1, Tv = 0.03.....	59
4. 22 โปรแกรมสำหรับเก็บข้อมูลอุณหภูมิเมื่อเปิด heater 20%	60
4. 23 System Identification toolbox	60
4. 24 การเพิ่มข้อมูลลงใน System Identification toolbox.....	61
4. 25 หน้าต่าง Process model.....	62
4. 26 แสดงกราฟที่ได้จากการทดลองและทรานเฟอร์ฟังก์ชันจากฟังก์ชัน system identification	62
4. 27 หน้าต่าง Curve Fitting Toolbox	63
4. 28 การใส่ตัวแปรและสมการใน Curve Fitting Toolbox.....	64
4. 29 กราฟที่ได้จากการ Curve Fitting เมื่อเปิด heater 20%	64
4. 30 โปรแกรม Simulink สำหรับจำลองระบบควบคุมพีไอดีด้วย close loop control.....	65
4. 31 ผลของการปรับพีไอดีของโมเดล.....	66
4. 32 การ device ethernet ของโปรแกรม CODESYS.....	67
4. 33 Add Modbus TCP Master-Slave ของโปรแกรม CODESYS	68
4. 34 IP Address, Subnet mask และ gateway ของ raspberry pi ในโปรแกรม CODESYS	68
4. 35 IP Address, unit-ID และ Port ของ Arduino(Slave) ในโปรแกรม CODESYS.....	69
4. 36 เพิ่ม channel ในการติดต่อสื่อสารกับ Arduino.....	69
4. 37 กำหนดตัวแปรที่ใช้สำหรับรับและส่งข้อมูลกับ Arduino.....	69
4. 38 IP Address, unit-ID และ Port ของ NodeMCU (Slave) ในโปรแกรม CODESYS	70
4. 39 เพิ่ม channel ในการติดต่อสื่อสารกับ NodeMCU	70
4. 40 กำหนดตัวแปรที่ใช้สำหรับรับและส่งข้อมูลกับ NodeMCU	71
4. 41 การ include libraries.....	74
4. 42 หน้าแสดงผลในโปรแกรม Arduino IDE	78
4. 43 การ include libraries.....	80
5. 1 โปรแกรมสำหรับควบคุมระบบบน CODESYS.....	87

สารบัญรูป(ต่อ)

รูปที่	หน้า
5. 2 หน้าจอสำหรับควบคุม และแสดงผลอุณหภูมิและผลการควบคุมของระบบ.....	88
5. 3 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 70 องศาเซลเซียส	89
5. 4 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 80 องศาเซลเซียส	89
5. 5 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 90 องศาเซลเซียส	90
5. 6 กราฟแสดงอุณหภูมิในถังผสมสารเคมีควบคุมอุณหภูมิที่ 70 องศาเซลเซียส.....	91
5. 7 กราฟแสดงอุณหภูมิในถังผสมสารเคมีควบคุมอุณหภูมิที่ 80 องศาเซลเซียส.....	91
5.8 กราฟแสดงอุณหภูมิในถังผสมสารเคมีควบคุมอุณหภูมิที่ 90 องศาเซลเซียส.....	91



บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในอุตสาหกรรมสายการผลิตที่มีการผสมสารเคมี ขั้นตอนการผสมสารมีความสำคัญอย่างมากต่อสมบัติและคุณภาพของผลิตภัณฑ์ ถ้าสารเคมีที่ทำการผสมมีการกระจายตัว (distribution) หรือแตกตัว (dispersion) ได้ไม่ดี จะส่งผลโดยตรงต่อความสม่ำเสมอของคุณภาพผลิตภัณฑ์

เนื่องจากเทคโนโลยีการผสมสารให้ได้ผลิตภัณฑ์ที่มีคุณภาพคงที่ตามที่ต้องการนั้นมีปัจจัยที่ต้องพิจารณามากมาย และต้องอาศัยความรู้ความเข้าใจในการผสมสารนั้นๆ ทั้งปริมาณของสารแต่ละชนิด ลำดับขั้นตอนในการผลิต ตลอดจนลำดับการใส่สารเคมี รวมไปถึงปัจจัยต่างๆ เช่น อุณหภูมิ ความดัน และความเร็วในการกวนสาร ที่มีผลต่อการใช้เครื่องผสมสารต่างชนิดกัน(เครื่องผสมระบบปิด เครื่องผสมระบบเปิด หรือเครื่องผสมแบบต่อเนื่อง) ดังนั้นจึงต้องมีการควบคุมอุณหภูมิ ความดัน และความเร็วในการกวนสาร เพื่อให้ได้ผลิตภัณฑ์ที่มีคุณภาพตามต้องการ

เครื่องปฏิกรณ์แบบแบตช์เป็นรูปแบบการผลิตที่สำคัญในอุตสาหกรรมเคมีที่ทันสมัยสำหรับการตระหนักถึงการเปลี่ยนแปลงบ่อยครั้งโรงงานผลิตที่ต้องการผลิตภัณฑ์หลายชนิดในปริมาณตัวแปร เมื่อเปรียบเทียบกับเครื่องปฏิกรณ์ต่อเนื่องเครื่องปฏิกรณ์แบบแบตช์เสนอทางเลือกที่ยืดหยุ่นและน่าสนใจเชิงเศรษฐกิจเนื่องจากอุปกรณ์เดียวกันสามารถใช้งานได้แตกต่างกันผลิตภัณฑ์ขึ้นอยู่กับความต้องการของตลาด ตัวอย่างของกระบวนการเครื่องปฏิกรณ์แบบแบตช์กำลังใช้กันอย่างแพร่หลายผลิตภัณฑ์อาหารยาสี โพลีเมอร์และผลิตภัณฑ์พิเศษสารเคมี แนวโน้มนี้คาดว่าจะดำเนินต่อไปทศวรรษหน้าในขณะที่อุตสาหกรรมสมัยใหม่แสวงหาการผลิตสารเคมีเพิ่มมูลค่าปริมาณต่ำ

ทฤษฎีการควบคุมที่เหมาะสมได้ถูกนำมาใช้อย่างกว้างขวางในการสืบทอดนโยบายการดำเนินงานสำหรับเครื่องปฏิกรณ์แบบกะ เอกสารจำนวนมากได้รับการตีพิมพ์ที่แตกต่างกันสำรวจการออกแบบการควบคุมที่ดีที่สุด

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาและออกแบบถึงปฏิกิริยาเคมีแบบความดันสูง
2. ศึกษาและออกแบบระบบควบคุมกระบวนการ

3. ศึกษาและเลือกใช้รูปแบบการสื่อสารข้อมูลแบบไร้สายโดยใช้ raspberry pi

1.3 ขอบเขตของปริญญาโท

1. สร้างชุดควบคุมกระบวนการด้วยราสเบอร์รี่พาย
2. เขียนโปรแกรมควบคุมกระบวนการด้วยโคดีซิด
3. ปรับค่าพารามิเตอร์พีไอดีสำหรับควบคุมอุณหภูมิของตัวปฏิบัติการเคมีแบบความดันสูง
4. สร้างโมเดลคณิตศาสตร์ของระบบ โดยใช้โปรแกรมแมทแลป

1.4 ขั้นตอนการศึกษา

ปริญญาโทฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทด้วยกันดังนี้

บทที่ 1 บทนำ

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

บทที่ 3 การประยุกต์ใช้ฮาร์ดแวร์และซอฟต์แวร์

บทที่ 4 จำลองระบบควบคุมอุณหภูมิสำหรับถังผสมสารเคมี

บทที่ 5 การควบคุมอุณหภูมิในระบบจริง

บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้เครื่องมือวัดและสามารถเขียนโปรแกรมสำหรับควบคุมกระบวนการที่มีการสื่อสารกันระหว่างอุปกรณ์ผ่านทางเครือข่ายอินเทอร์เน็ต พร้อมควบคุมกระบวนการผ่านหน้าจอแสดงผล (HMI)
2. สามารถออกแบบตัวควบคุมพีไอดีที่ใช้ในกระบวนการควบคุมอุณหภูมิ
3. สามารถนำข้อมูลที่ได้จากการทดลองมาหาโมเดลทางคณิตศาสตร์ และทำการ auto-tune PID เพื่อหาค่าพารามิเตอร์ที่เหมาะสมในโปรแกรมแมทแลป
4. สามารถนำความรู้ที่ได้รับไปประยุกต์ใช้ในกระบวนการทำงานอื่นได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 แบบจำลองทางคณิตศาสตร์

โลกของคณิตศาสตร์ (Mathematical World) เป็นโลกของนามธรรมที่มีการใช้สัญลักษณ์แทนนามธรรมเหล่านั้น มีการนิยามข้อตกลงต่างๆ เพื่อใช้ในการศึกษาระบบ และพิสูจน์ผลที่ได้เป็นกฎหรือทฤษฎีเพื่อการนำไปใช้ ดังนั้นการสร้างตัวแบบเชิงคณิตศาสตร์ จึงจัดว่าเป็นการจำลองสิ่งที่อยู่ในโลกแห่งความจริง ให้เป็นตัวแบบเชิงคณิตศาสตร์ที่อยู่ในรูป และใช้กฎเกณฑ์ทางคณิตศาสตร์หาคำตอบของตัวแบบเพื่อนำผลกลับไป นามธรรมประยุกต์ใช้กับโลกแห่งความเป็นจริง ซึ่งสามารถสรุปเป็นขั้นตอนได้ 6 ขั้นตอน ดังต่อไปนี้

ขั้นที่ 1 ทำความเข้าใจปัญหา (Identify the real problem) การทำความเข้าใจปัญหาเป็นขั้นตอนแรกในการสร้างตัวแบบ ต้องวิเคราะห์ให้ทราบว่า ปัญหาคืออะไร มีอะไรบ้างที่เกี่ยวข้องกับปัญหา มีคำถามมากมายที่ต้องทำความเข้าใจ โจทย์ ปัญหาในสถานการณ์จริง เช่น ปัญหานี้ต้องการทราบอะไร มีวัตถุประสงค์และเป้าหมายอะไรจะตัดสินใจผลที่ออกมาอย่างไร แหล่งข้อมูลมาจากไหนเชื่อถือได้หรือไม่ มีคำตอบเป็นแบบเดียวหรือไม่ จำแนกปัญหาว่าเป็น แบบมีคำตอบแน่นอน (deterministic) หรือ แบบมีคำตอบไม่แน่นอน (stochastic) ต้องใช้การสร้างสถานการณ์จำลอง (simulation) หรือไม่ คำถามหรือคำตอบ ดังกล่าวมาแล้วต้อง นิยาม กำหนดขอบเขต ให้ตรงประเด็นและชัดเจน

ขั้นที่ 2 สร้างตัวแบบเชิงคณิตศาสตร์ (Formulate a mathematical model) ขั้นตอนนี้เป็นขั้นตอนการสร้างตัวแบบเชิงคณิตศาสตร์จากปัญหาที่ได้วิเคราะห์หรือทำให้ชัดเจนแล้วในขั้นที่ 1 ทดลองสร้างตัวแบบที่ซับซ้อนน้อยที่สุดก่อน เขียนแผนภาพตามความเหมาะสม เขียนรายการปัจจัยที่เกี่ยวข้อง รวบรวมข้อมูลและทดสอบเนื้อหารายละเอียดอธิบายพฤติกรรมของตัวแปร รวบรวมข้อมูลเพิ่มเติมถ้าจำเป็น แสดงตัวแปรแต่ละตัวด้วยสัญลักษณ์ที่เหมาะสมพร้อมทั้งกำหนดหน่วย กำหนดข้อสมมติที่ต้องการสร้าง เขียนความสัมพันธ์และสมการของตัวแปรในโจทย์ โดยใช้ความรู้ทางคณิตศาสตร์ เช่น การเป็นสัดส่วน ความสัมพันธ์เชิงเส้นและไม่เชิงเส้น ความสัมพันธ์จากการทดลอง หลักการ Input/output กฎการเคลื่อนที่ของนิวตัน สมการเชิงผลต่างและสมการเชิงอนุพันธ์ เมทริกซ์ ความน่าจะเป็น การกระจายเชิงสถิติ เป็นต้น

ขั้นที่ 3 หาผลลัพธ์ของตัวแบบ (Solve the mathematical model) การหาคำตอบทางคณิตศาสตร์ของตัวแบบ อาจจะใช้วิธีเกี่ยวกับพีชคณิต หรือใช้วิธีเชิงตัวเลข ใช้แคลคูลัสและกราฟ เขียน

โปรแกรมคอมพิวเตอร์ หรือใช้โปรแกรมสำเร็จรูปที่เหมาะสม หาค่าของตัวแปรที่ต้องการ อาจจะเป็นรูปแบบตารางหรือรูปภาพ

ขั้นที่ 4 แปลความหมายของผลลัพธ์ (Interpret the mathematical solution) ขั้นตอนนี้เป็นการแปลความหมาย และตรวจสอบผลลัพธ์ที่ได้จากวิธีการเชิงคณิตศาสตร์ เช่น พิจารณาค่าของตัวแปรที่หาได้ ว่ามีเครื่องหมาย และขนาดถูกต้องหรือไม่มีค่าเพิ่มหรือลดตามที่ควรจะเป็นหรือไม่พฤติกรรมความไวต่อสิ่งกระตุ้น ได้คำตอบที่ดีที่สุดตามที่คาดไว้หรือไม่ หรือต้องเปลี่ยนเงื่อนไขเริ่มต้น

ขั้นที่ 5 ตรวจสอบผลลัพธ์กับข้อมูลจริง (Compare with reality) ผลลัพธ์ที่ได้สามารถตรวจสอบกับข้อมูลจริงได้หรือไม่ คำตอบเชิงคณิตศาสตร์มีความหมายหรือไม่ การทำนายสอดคล้องกับข้อมูลจริงหรือไม่ ประเมินตัวแบบที่สร้างขึ้นว่าได้ครบตามวัตถุประสงค์หรือไม่ ตัวแบบสามารถปรับปรุงให้ดีขึ้นได้อีกหรือไม่ ผลลัพธ์ที่ได้ก่อนหน้านี้ ชี้ให้เห็นว่าต้องคำนวณหาตัวแปรจากตัวแบบที่ปรับปรุงใหม่เพื่อความแม่นยำที่ดีกว่าหรือไม่ ถ้าต้องการทำใหม่ก็ต้องกลับไปเริ่มที่ขั้นที่ 1 หรือถ้าไม่ต้องก็ให้ไปที่ขั้นที่ 6 ขั้นตอนนี้สำคัญมาก เพราะมีบ่อยครั้งที่ต้องสร้างตัวแบบหลายรอบก่อนที่จะได้ผลเป็นที่น่าพอใจ

ขั้นที่ 6 เขียนรายงาน (Write a report) การเขียนรายงานต้องทราบว่าจะเขียนเพื่อใคร ผู้อ่านต้องการทราบอะไร ต้องการรายละเอียดในรายงานมากน้อยเพียงใด จะสร้างรายงานอย่างไร จึงจะทำให้ลักษณะที่สำคัญชัดเจน และผลที่ต้องการทราบปรากฏอยู่ ขั้นตอนนี้อาจจะไม่ต้องทำ ถ้าไม่ทราบว่าจะเขียนให้ใครอ่าน

การจำแนกตัวแบบเราสามารถจำแนกตัวแบบออกได้ดังนี้

1. ตัวแบบเชิงคณิตศาสตร์ (Mathematical Model) หรือตัวแบบนามธรรม (Abstract Model) เป็นตัวแบบที่ประกอบด้วยสัญลักษณ์ มักอยู่ในรูปสมการ อสมการหรือฟังก์ชันเชิงคณิตศาสตร์
2. ตัวแบบรูปธรรม (physical Model) เป็นตัวแบบที่สามารถจับต้องได้ เช่นแบบจำลองอาคารแบบหุ่นจำลองต่างๆ
3. ตัวแบบรูปภาพ (Visual Mode) เป็นตัวแบบลักษณะรูปภาพที่สามารถมองเห็นได้เช่น กราฟ แผนที่แบบแปลน ลายแทง

การจำแนกข้อมูลทางเวลาเพื่อนำมาสร้างโมเดลทางคณิตศาสตร์สามารถจำแนกออกได้ดังนี้

1. จำแนกตามกาลเวลา ซึ่งจำแนกได้เป็นตัวแบบสถิต กับตัวแบบพลวัต (Static and Dynamic) ตัวแบบสถิต เป็นตัวแบบที่ไม่เกี่ยวข้องกับเวลา ส่วนตัวแบบพลวัตเป็นตัวแบบที่มีเวลาเข้ามาเกี่ยวข้อง คำตอบของตัวแบบขึ้นอยู่กับช่วงเวลา
2. จำแนกตามความแน่นอน ซึ่งจำแนกได้เป็นตัวแบบแน่นอน กับตัวแบบความน่าจะเป็น (Deterministic and Probabilistic) ตัวแบบแน่นอนเป็นตัวแบบที่มีข้อมูลนำเข้าที่แน่นอนซึ่งทำให้กำ

ตอบของตัวเบบมีผลที่แน่นอน ส่วนตัวเบบความน่าจะเป็น เป็นตัวเบบที่ข้อมูลนำอย่างน้อย 1 ตัว อยู่ในรูปตัวแปรสุ่มซึ่งคำตอบของตัวเบบจะได้ในเชิงการคาดคะเน

3. จำแนกตามความต่อเนื่อง ซึ่งจำแนกได้เป็น ตัวเบบต่อเนื่อง กับตัวเบบไม่ต่อเนื่อง (Contintous and Discret) ตัวเบบต่อเนื่อง เป็นตัวเบบที่มีข้อมูลนำเข้าต่อเนื่องตลอดเวลา ช่นกรเปลี่ยนแปลงของระดับน้ำ ส่วนตัวเบบไม่ต่อเนื่องเป็นตัวเบบที่มีข้อมูลนำเข้าไม่ต่อเนื่อง เช่น จำนวนลูกค้าที่มาใช้บริการของธนาคาร จะเปลี่ยนแปลงเมื่อมีลูกค้าเข้าหรือออกจากธนาคารเท่านั้น

2.2 Curve fitting

สำหรับการออกแบบอัลกอริธึมการควบคุมจำนวนมากจำเป็นต้องมีแบบจำลองกระบวนการ เมื่ออัลกอริธึมการควบคุมเป็นของสัดส่วนบวกอินทิกรัล (PI) หรือสัดส่วนบวกอินทิกรัลบวกอนุพันธ์ (PID) รูปแบบกระบวนการมักจะสันนิษฐานว่ามีลำดับแรกพร้อมโครงสร้างเวลาตาย (FOPDT) นี่เป็นเพราะการตอบสนอง (FOPDT) เป็นการประมาณที่ดีในการตอบสนองที่แท้จริงของระบบและระบบย่อยหลายอย่างที่พบในแอปพลิเคชันการควบคุมกระบวนการอุตสาหกรรม นอกจากนี้การปรับพารามิเตอร์ตัวควบคุม PID สามารถเริ่มต้นได้โดยใช้สูตรที่ระบุค่าในแง่ของพารามิเตอร์ FOPDT วิธีการปรับยอदनิยมประกอบด้วยวิธีการที่ Ziegler-Nichols [1], Cohen-Coon [2] และ Kappa-Tau [3] กระดาษนี้มุ่งเน้นไปที่การประมาณค่าพารามิเตอร์แบบจำลอง FOPDT โดยใช้ข้อมูลการตอบสนองขั้นตอนและการเดินของชีพจร

วิธีแรกในการระบุระบบอาศัยเทคนิคกราฟิกและวิธีการของ Ziegler และ Nichols [1], Oldenbourg และ Satorious, Sten และ Rake เป็นตัวอย่างของวิธีการเหล่านี้ สำหรับการตอบสนองแบบขั้นตอนเดียวการรวบรวมวิธีการที่อิงตามการคำนวณพื้นที่เฉพาะนั้นแสดงอยู่ในกระดาษของ Nishikawa และคณะ และหนังสือโดย Astrom และ Hagglund เมื่อไม่นานมานี้มีการเผยแพร่วิธีการใหม่ ๆ ในตระกูลที่เกี่ยวข้องกับการคำนวณมากขึ้น แต่ช่วยให้สามารถประมาณค่าพารามิเตอร์ของรูปแบบฟังก์ชันการถ่ายโอนรวมถึงการหน่วงเวลา พวกเขามักจะเรียกว่าวิธีการของสมการอินทิกรัล (IE) แนวคิดดั้งเดิมของ Bi et al. และ Wang and Zhang ถูกขยายเพื่ออนุญาตการระบุภายใต้เงื่อนไขเริ่มต้นชั่วคราว การพัฒนาอื่น ๆ ยังได้รับรายงาน ตัวอย่างเช่นในการอ้างอิง และมีการนำเสนอวิธีการใหม่ในการจัดการ step การทดสอบตามขั้นตอนที่ละเอียดขึ้น

ในทุกกรณีข้างต้นซอฟต์แวร์วัตถุประสงค์พิเศษได้รับการพัฒนาซึ่งโดยทั่วไปไม่สามารถใช้ได้ อย่างไรก็ตาม อย่างไรก็ตามวิธีการทางเลือกโดยใช้ซอฟต์แวร์ที่เป็นกรรมสิทธิ์ของเส้นโค้งที่เป็นกรรมสิทธิ์ได้ง่ายขึ้น มีศักยภาพที่จะให้วิธีแก้ปัญหาที่ง่ายกว่าในการกำหนดพารามิเตอร์แบบจำลอง FOPDT สำหรับผู้ที่ไม่สามารถเข้าถึงซอฟต์แวร์ผู้เชี่ยวชาญชนิดที่อธิบายไว้ใน วารสารคดีการวิจัย

เป็นเวลาหลายปีแล้วที่ความสนใจในความสัมพันธ์ระหว่างการระบุระบบและการปรับเส้นโค้ง จุดประสงค์ของบทความนี้คือเพื่อแสดงให้เห็นว่าระบบการระบุตัวตนสามารถดำเนินการในรูปแบบการ ออกกำลังกายที่กระชับโค้งได้อย่างไร หลังจากพิจารณาถึงวิธีการระบุระบบ IE ล่าสุด มีการใช้โซลูชันซอฟต์แวร์การปรับโค้งสามแบบสำหรับสภาพแวดล้อมซอฟต์แวร์ MATLAB เพื่อประเมินพารามิเตอร์ของแบบจำลอง FOPDT โดยใช้ข้อมูลที่สร้างขึ้นภายใต้การจำลอง คำแนะนำโดยละเอียดเกี่ยวกับวิธีการใช้วิธีการต่าง ๆ จะได้รับและการประเมินความถูกต้องและใช้งานง่าย จากนั้นใช้วิธีการของ IE และวิธีการปรับเส้นโค้งที่ดีที่สุดจากข้อมูลการทดลองที่ได้จากห้องปฏิบัติการและทำการเปรียบเทียบเพิ่มเติม

2.2.1 Curve fitting โดยใช้แมทแลป

นอร์ตัน อธิบายการระบุว่าเป็นกระบวนการสร้างแบบจำลองทางคณิตศาสตร์ของระบบพลวัต จากการสังเกตและความรู้ก่อนหน้า ในรายงานที่จัดทำโดยห้องแล็บ Open Openability Technology แห่งมิชิแกนเทคนิคการอธิบายเส้นโค้งเป็นการกำหนดรูปแบบทางคณิตศาสตร์ที่เหมาะสมกับข้อมูลการทดลอง นอกจากนี้แบบที่ดีที่สุดของพวกเขาถูกกำหนดเชิงปริมาณเป็นการลดความแตกต่างระหว่างข้อมูลและเส้นโค้ง ในที่สุด Ljung จัดทำ 'เครื่องมือ' ระหว่างการจำแนกระบบพลวัตเชิงเส้นและการปรับเส้นโค้งแบบคลาสสิก

ปัญหา "พอดีที่สุด" สำหรับข้อมูลการทดลองมักเกี่ยวข้องกับการแก้ปัญหากำลังสองน้อยที่สุด บ่อยครั้งที่ปัญหาเหล่านี้เป็นเชิงเส้น แต่ในกรณีของการตอบสนอง FOPDT มันไม่เชิงเส้นเพราะเส้นโค้งที่จะใช้ในกระบวนการติดตั้งไม่ได้เป็นเส้นตรงในพารามิเตอร์ ในกรณีดังกล่าวมีความจำเป็นต้องหันไปใช้วิธีการกำลังสองน้อยที่สุดแบบไม่เชิงเส้นที่เกี่ยวข้องกับการปรับปรุงซ้ำกับค่าพารามิเตอร์ ในหลักการนี้คล้ายกับวนซ้ำในวิธี IE ที่อธิบายไว้ในส่วนที่ 2 บทความที่ยอดเยียมโดย Gavin มีรายละเอียดบทความวิเคราะห์ของหนึ่งในกลยุทธ์ที่นิยมมากขึ้นเรียกว่า Levenberg-Marquardt algorithm (LMA) อย่างไรก็ตามวิธีการที่ไม่ได้เป็นการอธิบายพื้นฐานทางคณิตศาสตร์ของวิธีการต่าง ๆ เหล่านี้ แต่เพื่ออธิบายว่าโซลูชันซอฟต์แวร์ต่าง ๆ ที่ยึดตามอัลกอริทึมเหล่านี้สามารถใช้ประโยชน์ได้อย่างไร ในบทความของ Ljung การตอบสนองของระบบพลศาสตร์เชิงเส้นในเวลาต่อเนื่องอธิบายโดย:

$$y(t) = G(p)u(t) + v(t) \quad (2.1)$$

โดยที่ p คือตัวดำเนินการสร้างความแตกต่าง ในปัญหาการระบุทั่วไปจุดมุ่งหมายคือการหาผู้ดำเนินการถ่ายโอน $G(p)$ และสเปกตรัมของสารเติมแต่งเสียง $v(t)$ อย่างไรก็ตามในกรณีนี้โครงสร้าง

ของฟังก์ชันการถ่ายโอนเป็นที่รู้จักกันที่จะมีรูปแบบที่อธิบายโดยสมการ (1) ดังนั้นจึงสามารถแสดงสิ่งต่อไปนี้ได้

สำหรับฟังก์ชันขั้นตอนที่กำหนดโดย:

$$u(t) = \begin{cases} 0 & \text{for } t < 0 \\ 1 & \text{for } t \geq 0 \end{cases} \quad (2.2)$$

การตอบสนองจะได้รับจาก:

$$y(t) = Kh[1 - \exp(-(t-D)/T)] \cdot H[t-D] \quad (2.3)$$

โดยที่ $H[t-D]$ เป็นฟังก์ชันขั้นตอน Heaviside สำหรับพัลส์ที่มีความสูง h และความกว้าง W :

$$y(t) = \begin{cases} 0 & t < 0 \\ h & \text{for } 0 < t \leq W \\ 0 & t > W \end{cases} \quad (2.4)$$

$$y(t) = Kh[1 - \exp(-(t-D)/T)] \cdot H[t-D] - Kh[1 - \exp(-(t-D-W)/T)] \cdot H[t-D-W] \quad (2.5)$$

ดังนั้นจึงเป็นสมการ และ T จะเป็นตัวแทนของ 'เส้นโค้ง' เชิงทฤษฎีที่จะถูกติดตั้งกับข้อมูลการทดลองด้วย K , T และ D พารามิเตอร์ที่จะระบุ ส่วนต่อไปนี้อธิบายการใช้งานซอฟต์แวร์แอปพลิเคชันสามตัวที่พัฒนาขึ้นเพื่อใช้ภายในสภาพแวดล้อม MATLAB: กล่องเครื่องมือ Curve Fitting, กล่องเครื่องมือปรับให้เหมาะสมและกล่องเครื่องมือปรับโค้งอิสระฟรีที่เรียกว่า EzyFit ข้อมูลจะถูกนำเสนอในรูปแบบที่ใช้งานและตรงเพื่ออำนวยความสะดวกในการใช้งานเป็นแบบฝึกหัดสำหรับผู้ใช้นี้

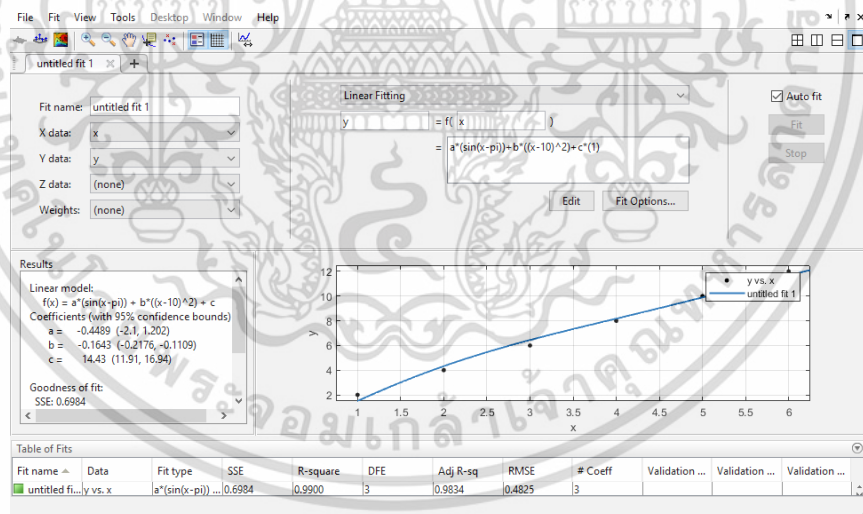
สำหรับแต่ละวิธีที่อธิบายไว้จะถือว่าข้อมูลขั้นตอนหรือการตอบสนองของซีพจร (ตามความเหมาะสม) มีอยู่ในพื้นที่ทำงานเมทแลป ในรูปแบบของเวลาและเวกเตอร์ผลลัพธ์ (โดยปกติจะเป็นสัญลักษณ์ที่กำหนด t และ y ตามลำดับข้อมูลที่ใช้ในต่อไป) ส่วนถูกสร้างขึ้นโดยใช้การจำลองซีมูลิงค์ผ่านบล็อกไปยังพื้นที่ทำงานด้วยรูปแบบบันทึกที่กำหนดเป็นอาร์เรย์

2.2.2 Curve Fitting Toolbox

กล่องเครื่องมือ Curve Fitting สามารถใช้ในสภาพแวดล้อมที่แตกต่างกันสองแบบ: ผ่านส่วนต่อประสานผู้ใช้แบบกราฟิก (GUI) (ซึ่งเป็นรูปแบบที่ตามมาด้านล่าง) หรือผ่านบรรทัดคำสั่งแมทแล็บ ส่วนนี้ให้ภาพรวมของชิ้นส่วนเหล่านั้นของกล่องเครื่องมือซึ่งจำเป็นต้องระบุฟังก์ชันการถ่ายโอน (FOPDT) โดยใช้ข้อมูลที่ปราศจากเสียงรบกวนที่สร้างขึ้นโดยรุ่น Simulink สำหรับผู้ที่ต้องการมีความเข้าใจที่สมบูรณ์ยิ่งขึ้นของคุณสมบัติทั้งหมดที่มีอยู่เมื่อใช้กล่องเครื่องมือการอ้างอิงที่ดีคือสิ่งที่ให้ไว้ในเอกสารสนับสนุนเพื่อแสดงให้เห็นถึงวิธีการขั้นพื้นฐานพิจารณาปัญหาของการกำหนดพารามิเตอร์ของแบบจำลอง (FOPDT) ที่โดดเด่นด้วยฟังก์ชันการถ่ายโอน:

$$G_p(s) = \frac{1.25 \exp(-2.15s)}{2s + 1} \quad (2.6)$$

ในการเปิดเครื่องมือปรับความโค้งให้ป้อน cftool ที่พร้อมกันในหน้าต่างคำสั่ง เครื่องมือนี้จะเปิดหน้าต่าง Curve Fitting Tool ที่แสดงในรูป



รูปที่ 2.1 หน้าต่าง Curve Fitting

ขั้นตอนต่อไปคือการคลิกที่ปุ่ม [ข้อมูล ...] ดังแสดงในรูป ซึ่งจะเปิดหน้าต่างข้อมูลคล้ายกับที่แสดงในรูปที่ 3 แต่ด้วยบานหน้าต่างแสดงตัวอย่างว่างเปล่าในขั้นต้น หน้าต่างนี้ออนุญาตให้นำเข้าข้อมูล X และ Y Data จากพื้นที่ทำงาน (เช่นข้อมูลและข้อมูล y สำหรับแอปพลิเคชันนี้) เมื่อข้อมูลถูกเลือกผ่านรายการดรอปดาวน์แล้วตัวอย่างของข้อมูลที่จะใช้สำหรับขั้นตอนการระบุจะปรากฏขึ้นดังที่แสดง หากต้องการไปยังหน้าต่างถัดไปให้คลิกที่ปุ่ม [สร้างชุดข้อมูล]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างต้นฉบับ (รูปที่ 2.1) จะมีข้อมูลตัวอย่างอยู่ในขณะนี้ ขั้นตอนต่อไปคือคลิกที่ปุ่ม [กระซับ ...] สิ่งนี้แสดงให้เห็นหน้าต่างการปรับพอดีซึ่งคล้ายกับที่แสดงในรูปที่ 4 แต่ด้วยพารามิเตอร์ที่กำหนดเองและผลลัพธ์ในตอนแรกจะว่างเปล่า การคลิกที่ปุ่ม [พอดีใหม่] อนุญาตให้ผู้ใช้ตั้งชื่อ (ชื่อพอดี) และเลือกประเภทของพอดี เนื่องจากไลบรารีเอ็กซ์โปเนนเชียลไม่มีสมการที่ต้องการผู้ใช้ต้องสร้างของตนเอง โดยใช้ทางเลือกสมการที่กำหนดเอง สิ่งนี้จะเปิดหน้าต่างสมการใหม่ที่กำหนดเองซึ่งแสดงในรูปที่ 5

ก่อนอื่นเลือกแท็บสมการทั่วไปเปลี่ยนตัวแปรอิสระเป็น t จากนั้นพิมพ์สมการที่ต้องการ ในไวยากรณ์แมทแลปการตอบสนอง (FOPDT) ถูกกำหนดดังนี้:

$$y = Kh * (1 - \exp(-(t - D)/T)) * \text{heaviside}(t - D) \quad (2.7)$$

ในสมการ (28), เฮติไซด์ ($t-D$) คือฟังก์ชันเฮวิไซด์ (ไม่น่าแปลกใจ) สิ่งนี้นำมาใช้ในกล่องเครื่องมือสัญลักษณ์แมทแลป อย่างไรก็ตามหากกล่องเครื่องมือนี้ไม่ได้ติดตั้งมันเป็นเรื่องง่ายสำหรับทุกคนที่มีทักษะพื้นฐานในการเขียนไฟล์ MATLAB m เพื่อพัฒนาเทียบเท่าผู้ใช้ที่เขียนคลิกที่ปุ่ม [ตกลง] สร้างการตอบสนองที่แสดงในรูปที่ 6. นอกจากนี้บนหน้าต่างผลลัพธ์ในหน้าต่างการติดตั้งจะแสดงค่าสำหรับ Kh , T และ D พร้อมกับคุณสมบัติที่รู้จักกันดีบางอย่างที่กำหนดคุณภาพหรือความดีของ พอดี (รูปที่ 4) ในกรณีนี้แสดงที่วันที่นั้นสมบูรณ์แบบและปราศจากเสียงรบกวนและรบกวนที่เลือกนั้นถูกต้อง ดังนั้นผลลัพธ์ที่ถูกต้องแสดงในรูปที่ 5 คือ $K = 1.25$, $D = 2.15$, $T = 2$ ในรูปที่ 5 เพชรสีน้ำเงินแสดงข้อมูลการทดลองและเส้นโค้งสีแดงได้มาจากค่าที่แสดงสำหรับ Kh , T และ D

Method: LinearLeastSquares

Coefficients	Lower	Upper
a	-Inf	Inf
b	-Inf	Inf
c	-Inf	Inf

Close

รูปที่ 2.2 หน้าต่างที่กำหนดค่าเอง

2.2.3 Optimization Toolbox

สำหรับผู้ใช้แมทแลปที่ไม่สามารถเข้าถึง Curve Fitting Toolbox แต่มีการเข้าถึง Optimization Toolbox [22] วิธีการแก้ปัญหาทางเลือกอื่นในการระบุปัญหาของระบบนั้นเป็นไปได้ที่จะใช้ประโยชน์จากอัลกอริทึม lsqcurvefit ฟังก์ชัน lsqcurvefit ช่วยแก้ปัญหากำลังสองน้อยที่สุดแบบไม่เชิงเส้นและเป็นส่วนต่อประสานที่สะดวกสำหรับปัญหาการปรับข้อมูล มันใช้อัลกอริทึม lsqnonlin ซึ่งเป็นเช่นเดียวกับที่ใช้โดย Curve Fitting Toolbox ในกรณีของฟังก์ชันการถ่ายโอน (FOPDT) จะช่วยให้ผู้ใช้สามารถหาค่าสำหรับ K, T และ L ซึ่งประกอบด้วยเวกเตอร์ซึ่งแก้ปัญหาได้:

$$\min \sum [F(a, XData) - YData]^2 \quad (2.8)$$

รับอินพุตข้อมูลเวกเตอร์ XData และเวกเตอร์ข้อมูลที่สังเกต YData ทั้งความยาว N นอกจากนี้ $F(a, XData)$ เป็นฟังก์ชันที่มีค่าเวกเตอร์ที่จับขั้นตอนหรือการตอบสนองของซีพจรดังที่แสดงก่อนหน้านี้ โปรดทราบว่าในกรณีนี้พัลส์อินพุตจะต้องเริ่มต้นในเวลาที่มากรกว่าศูนย์เช่น $t > 0$ หรืออัลกอริทึมที่อธิบายด้านล่างล้มเหลวและเกิดข้อผิดพลาดขึ้น สูตร m-ไฟล์ปกติใช้ประโยชน์จากแนวคิดฟังก์ชันที่ไม่ระบุชื่อ ฟังก์ชันที่ไม่ระบุชื่อเป็นหนึ่งที่ไม่ได้เก็บไว้ในโปรแกรม แต่ฟังก์ชันที่ยอมรับการป้อนข้อมูลและส่งกลับผลลัพธ์เช่นเดียวกับฟังก์ชันมาตรฐานทำ นอกจากนี้ยังมีคำสั่งที่เรียกใช้งานได้เพียงคำสั่งเดียว ดังที่แสดงในไฟล์ m MATLAB ด้านล่างตัวดำเนินการ @ สร้างฟังก์ชันนิรนาม

สำหรับอินพุตพัลส์เดียว, ความสูง H และความกว้าง W, ไฟล์เอ็มเค็ง FOPDT ที่เหมาะสมมีรูปแบบดังต่อไปนี้ (อัลกอริทึม 2): อัลกอริทึม 2 อัลกอริทึม 2 PULSE INPUT IDENTIFICATION โดยใช้

```
function [a, resnorm]=optlsqFOPDTPulse(W,H,Working_Data);
t=Working_Data(:,1);
y=Working_Data(:,2);
f=@(a,t) a(1)*H*(1-exp(-(t-a(3))/a(2)))...
.*heaviside(t-a(3))-a(1)*H*(1-exp(-(t-a(3)-W)/a(2)))...
.*heaviside(t-a(3)-W);
[a, resnorm] = lsqcurvefit(f, [0.5 0.5 0.5], t, y)
K=a(1); T=a(2); D=a(3);
end
```

สำหรับฟังก์ชันถ่ายโอนที่กำหนดโดยสมการ (27) ผลลัพธ์ที่ผลิตโดยอัลกอริทึม 2 สำหรับเวลาตัวอย่าง $TS = 0.1$ คือ $K = 1.2497$, $D = 2.1603$, $T = 1.999$ โดยมี $resnorm = 5.0072e-007$ ได้ผลลัพธ์ที่สมบูรณ์แบบโดยวิธี IE พร้อมการตอบสนองของพัลส์

ข้อดีข้อดี: เป็นเรื่องง่ายที่จะเรียนรู้ สามารถเรียกใช้จากส่วนต่อประสานผู้ใช้แบบ GUI ได้ • ความมั่งคั่งของข้อมูลสนับสนุนสามารถใช้งานออนไลน์ได้อย่างอิสระ

ข้อเสีย: ต้องซื้อกล่องเครื่องมือและเวอร์ชันที่ถูกต้องที่ใช้ภายในสภาพแวดล้อมแมทแลป • โครงสร้าง (FOPDT) ไม่ใช่ตัวเลือกเริ่มต้นที่ผู้ใช้สามารถเลือกได้

2.3 ระบบควบคุมพีไอดี

ตัวควบคุมพีไอดีนั้นเป็นวิธีการควบคุมที่พบได้บ่อยที่สุด ในทางปฏิบัติส่วนใหญ่ผู้ปฏิบัติงานจะขึ้นอยู่กับวิธีการควบคุมแบบพีไอดี หรือความแตกต่างเล็กน้อยของมัน คอนโทรลเลอร์จำนวนมากไม่ได้ใช้การกระทำต่อเนื่อง ตัวควบคุมพีไอดี ปรากฏในรูปแบบต่าง ๆ มากมายในฐานะที่เป็นตัวควบคุมแบบสแตนด์ออล และส่วนหนึ่งของแพ็คเกจ DDC (Direct Digital Control) หรือระบบควบคุมกระบวนการแบบกระจายตามลำดับขั้นหรือสร้างขึ้นในระบบฝังตัว วิศวกรเครื่องมือวัดและการควบคุมหลายพันคนทั่วโลกกำลังใช้คอนโทรลเลอร์ดังกล่าวในการทำงานประจำวัน อัลกอริทึมพีไอดี สามารถเข้าถึงได้จากหลาย ๆ ทิศทาง มันสามารถถูกมองว่าเป็นอุปกรณ์ที่สามารถใช้งานได้ด้วยกฎเชิงประจักษ์เพียงไม่กี่อย่าง แต่ก็สามารถวิเคราะห์ได้เช่นกัน โดยตัวควบคุมมีสมการดังนี้

$$u(t) = ke(t) + k_i \int_0^t e(t) dt + k_d \frac{de(t)}{dt} \quad (2.9)$$

โดยที่ u คือสัญญาณควบคุมและ e คือข้อผิดพลาดในการควบคุม ($e = r - y$) ค่าอ้างอิงนี้เรียกว่า เซ็ตพอยท์ สัญญาณควบคุมจึงเป็นผลรวมของสามเทอม: P-term (ซึ่งเป็นสัดส่วนกับความผิดพลาด), I-term (ซึ่งเป็นสัดส่วนกับปริพันธ์ของข้อผิดพลาด) และ D-term (ซึ่งเป็นสัดส่วนกับอนุพันธ์ของข้อผิดพลาด) พารามิเตอร์ของตัวควบคุมคืออัตราขยาย k , อินทิเกรต k_i และอนุพันธ์ของ k_d คอนโทรลเลอร์สามารถปรับพารามิเตอร์ได้เช่นกัน

$$u(t) = k(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}) \quad (2.10)$$

เมื่อ T_i เรียกว่า integral time และ T_d เรียกว่า derivative time ส่วนที่เป็นสัดส่วนทำหน้าที่เกี่ยวกับค่าปัจจุบันของข้อผิดพลาดส่วนประกอบที่เป็นตัวแทนและค่าเฉลี่ยของข้อผิดพลาดในอดีตและอนุพันธ์สามารถตีความได้ว่าเป็นการคาดการณ์ข้อผิดพลาดในอนาคตตามการประมาณเชิงเส้น (extrapolation)

2.3.1 ผลของสัดส่วน

รูปที่ 2.1 แสดงการตอบสนองของเอาต์พุตไปยังหน่วยขั้นตอนในสัญญาณคำสั่งสำหรับระบบที่มีการควบคุมแบบสัดส่วน กระบวนการนี้มีคอนโทรลเลอร์ที่มีฟังก์ชันถ่ายโอน $P(s)$ และ $C(s)$ ฟังก์ชันการถ่ายโอนจากการอ้างอิงไปยังผลลัพธ์คือ

$$G(s) = \frac{P(s)}{1+P(s)C(s)} \quad (2.11)$$

ข้อผิดพลาดของสถานะคงที่สำหรับขั้นตอนต่อหน่วยคือ $1/(1+kP(0))$ สำหรับระบบในรูปที่ 8.2 โดยได้รับ $k = 1, 2$ และ 5 ข้อผิดพลาดของสถานะคงที่คือ $0.5, 0.33$ 0.17 ข้อผิดพลาดลดลงเมื่อเพิ่มขึ้น เกน แต่ระบบก็มีความผันผวนมากขึ้นโปรดสังเกตในรูปที่ค่าเริ่มต้นของสัญญาณควบคุมเท่ากับตัวควบคุมเกนเพื่อหลีกเลี่ยงข้อผิดพลาดสถานะคงที่ตัวควบคุมสัดส่วนสามารถเปลี่ยนเป็น

$$u(t) = Ke(t) + u_b \quad (2.12)$$

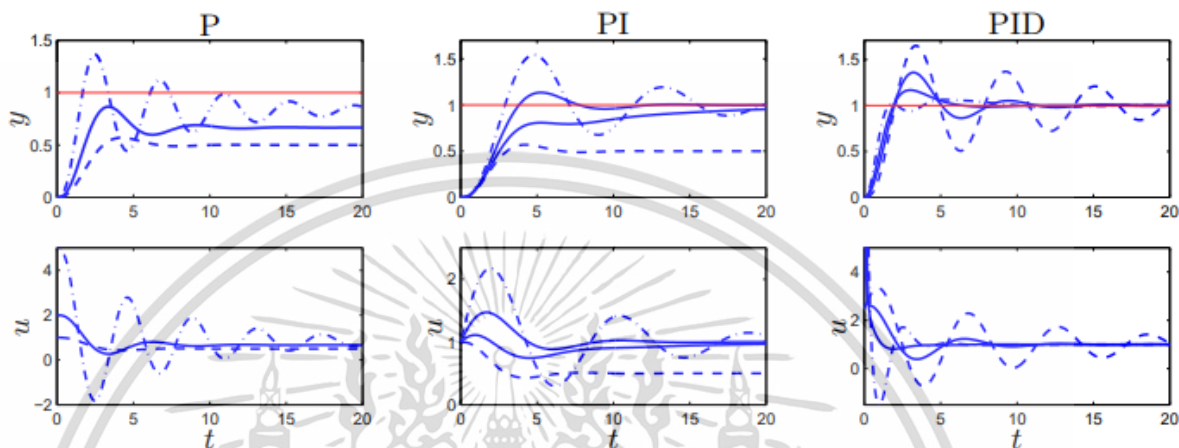
โดยที่ u_b เป็นค่ารีเซ็ตซึ่งถูกปรับเพื่อให้ค่าสถานะคงที่ที่ต้องการ

2.3.2 ผลของปริพันธ์

ผลของปริพันธ์รับประกันว่าค่าที่ส่งออกไปมีการอ้างอิงในสถานะสภาวะคงตัว สามารถแสดงได้ดังนี้ สมมติว่าระบบอยู่ในสถานะที่มีสัญญาณการควบคุมคงที่ U_0 และความผิดพลาดคงที่ $e_0 \neq 0$ มันดังมาจากสมการ

$$U_0 = ke_0 + k_i e_0 t \quad (2.13)$$

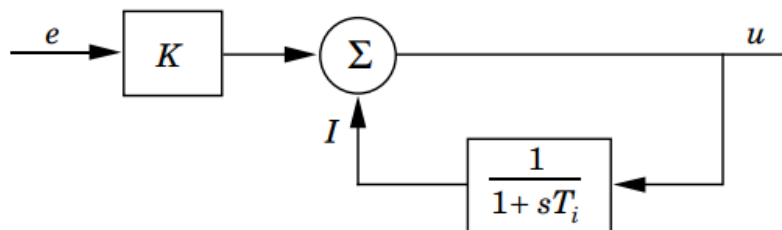
ด้านซ้ายมือเป็นค่าคงที่ แต่ทางด้านขวาเป็นฟังก์ชันของ t เรามีความขัดแย้งและ e_0 ต้องเป็นศูนย์ สังเกตว่าในการโต้แย้งนี้มีเพียงสมมติฐานเดียวเท่านั้นที่มีสถานะมั่นคง ไม่มีการพูดถึงกระบวนการที่เฉพาะเจาะจง ยกตัวอย่างเช่น ไม่เชิงเส้น



รูปที่ 2.3 การตอบสนองต่อการเปลี่ยนแปลงขั้นตอนในสัญญาณคำสั่งสำหรับตัวควบคุม P (ซ้าย), PI (กลาง) และตัวควบคุม PID (ขวา) กระบวนการนี้มีฟังก์ชันการถ่ายโอน $P(s) = 1 / (s + 1)^3$ พารามิเตอร์ของตัวควบคุมคือ $k = 1$ (ประ), 2 และ 5 (ประ-จุด) สำหรับตัวควบคุม P, $k = 1$, $k_i = 0$ (ประ), 0.2, 0.5 และ 1 (ประ-จุด) สำหรับตัวควบคุม PI และ $k = 2.5$, $k_i = 1.5$ และ $k_d = 0$ (ประ), 1, 2, 3 และ 4 (ประ-จุด) สำหรับ PID ตัวควบคุม (ที่มา : AIDAN, 2003)

อีกอาร์กิวเมนต์นี้ก็คือฟังก์ชันการถ่ายโอนของตัวควบคุมที่มีการกระทำที่สำคัญได้รับไม่มีที่สิ้นสุดที่ความถี่ศูนย์ ($C(0) = \infty$) นั้นมาจาก (2.3) ที่ $G(0) = 0$ อาร์กิวเมนต์นี้ต้องการอย่างไรก็ตามระบบเป็นแบบเชิงเส้น

การกระทำแบบอินทิกรัลยังสามารถดูได้ว่าเป็นวิธีในการสร้างเทอม u_b ในตัวควบคุมสัดส่วน (2.4) โดยอัตโนมัติ ถูกสร้างขึ้นโดยการกรองความถี่ต่ำผ่านเอาต์พุต การนำไปใช้งานนี้เรียกว่าการรีเซ็ตอัตโนมัติเป็นหนึ่งในสิ่งแรกที่สำคัญของการควบคุม ฟังก์ชันการถ่ายโอนของระบบในรูปที่ 2.2 ได้มาจากการติดตามลูป สมมติว่าเป็นสัญญาณเอ็กซ์โปเนนเชียลและติดตามพวกมัน



รูปที่ 2.4 การใช้งานอินทิกรัลเป็นการปรับอัตโนมัติ (ที่มา : AIDAN, 2003)

2.3.3 ผลของอนุพันธ์

จากรูปที่ 2.1 แสดงให้เห็นว่าการกระทำอนุพันธ์สามารถปรับปรุงเสถียรภาพของระบบวงปิด ความสัมพันธ์อินพุท - เอาท์พุทของคอนโทรลเลอร์ที่มีการกระทำตามสัดส่วนและอนุพันธ์คือ

$$u(t) = ke(t) + k_d \frac{de(t)}{dt} = k(e(t) + T_d \frac{de(t)}{dt}) \quad (2.14)$$

โดย $T_d = T_d/k$ คือเวลาอนุพันธ์ การกระทำของตัวควบคุมที่มีการกระทำตามสัดส่วนและอนุพันธ์สามารถตีความได้ราวกับว่าการควบคุมนั้นได้ทำตามสัดส่วนกับผลลัพธ์ของกระบวนการที่ทำนายไว้ซึ่งการทำนายนั้นทำโดยการคาดการณ์ข้อผิดพลาด T_d time units ในอนาคตโดยใช้แทนเจนต์กับเส้นโค้งข้อผิดพลาด แสดงให้เห็นถึงพฤติกรรมของระบบที่มีตัวควบคุม PID ระบบจะมีการแกว่งเมื่อไม่ใช้การกระทำอนุพันธ์และมันจะขึ้นมากขึ้นเมื่อได้รับอนุพันธ์เพิ่มขึ้น

บทที่ 3

การประยุกต์ใช้ฮาร์ดแวร์และซอฟต์แวร์

3.1 ส่วนประกอบของชุดควบคุมอุณหภูมิ

3.1.1 ถังสำหรับการผสมสารเคมี

สำหรับถังผสมสารเคมีจะแบ่งเป็นสองส่วน คือ

ส่วนแรก คือ ถังภายนอก ใช้สำหรับบรรจุสารที่ใช้พาความร้อนให้กับถังที่อยู่ภายในให้ทั่วถึง

ส่วนที่สอง คือ ถังภายใน ใช้สำหรับบรรจุสารที่ต้องการจะผสม ซึ่งตัวถังด้านในนี้จะมี ส่วนประกอบย่อยๆ ได้แก่ ตัวถังที่ใช้สำหรับบรรจุสาร, ตัวล็อกฝาถังกับถังเพื่อเพิ่มความแน่นในการปิดฝา ถัง และฝาสำหรับปิด ซึ่งจะติดตั้งอุปกรณ์สำหรับวัดอุณหภูมิ ความดัน Safety relief valve และ ไมโครคอนโทรลเลอร์สำหรับสื่อสารกับอุปกรณ์ภายนอก

3.1.2 เตาให้ความร้อน Hot Plate

เป็นเตาไฟฟ้ารุ่นแรกๆ มีขดลวดทำความร้อนอยู่ใต้แผ่นสีดำ ให้ความร้อนค่อนข้างช้า กินไฟมาก การปรับระดับความร้อนขึ้นลงค่อนข้างช้า เพราะความร้อนต้องผ่านตัวกลางหลายตัว ตั้งแต่ขดลวด แผ่น เซรามิกสีดำ กันหม้อ มีข้อดีคือ “ราคาถูก”

3.1.3 Stirrer

คือ เครื่องกวนสารแบบใบกวน, เครื่องกวนสารปั่นละลาย ผสมของเหลวต่างๆ ของเหลวและ ของแข็ง ผง สารละลายของสารเคมี ยา อาหาร และผลิตภัณฑ์เครื่องสำอาง



รูปที่ 3.1 Stirrer (ที่มา : กรีนเทค พลัส, 2013)

3.1.4 Raspberry pi 4 Model B

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ดราสเบอร์รี่พายรองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย คุณสมบัติทางเทคนิคของบอร์ด

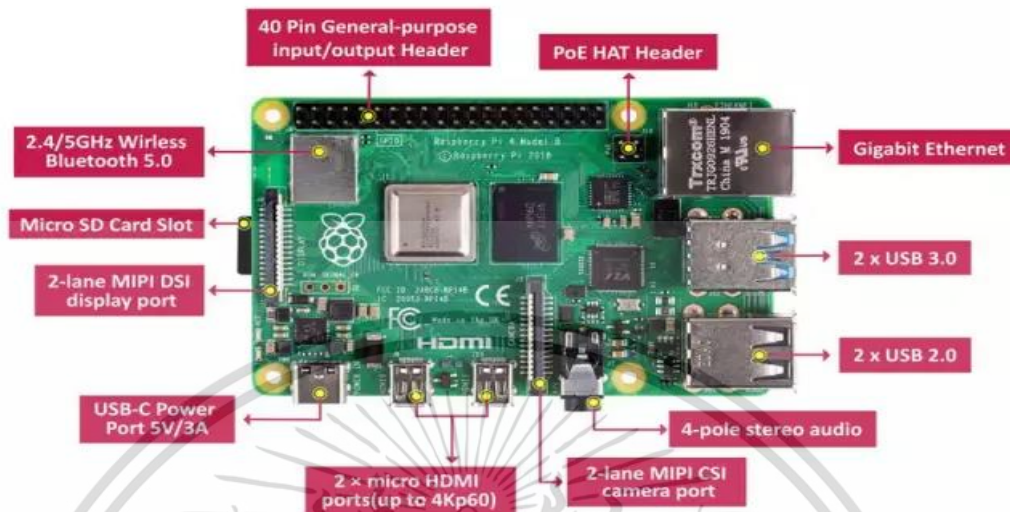
บอร์ด Raspberry Pi ปัจจุบันมีด้วยกัน 2 โมเดล คือ โมเดล A และ โมเดล B ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน รายละเอียดดังตาราง

ตารางที่ 3.1 เปรียบเทียบสเปกของราสเบอร์รี่พายโมเดล A และ B

	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1, 1080p 30 h.264/MPEG-4 AVC high-profile decoder and encoder	
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)
USB 2.0 Ports	1 (direct form BCM2835)	2 (via the build in integrated 3-port USB hub)

Video Input	A CSI input connector allows for the connection of RPF designed camera module (ออกแบบมาให้เชื่อมต่อกับ Raspberry Pi Camera Module โดยเฉพาะ)	
Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งสองแบบ คือ แบบ RCA และแบบ HDMI)	
Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I ² S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripherals	8 x GPIO, UART, I ² C Bus, SPI Bus with two chip selects, I ² S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	
Size	85.60 mm x 53. Mm (3.370 inch x 2.125 inch)	
Weight	45 g. (1.6 oz.)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 บอร์ด raspberry pi (ที่มา : Thaiware, 2019)

3.1.5 Arduino UNO

Arduino คือ โครงการที่นำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่างๆ มาใช้ร่วมกันในภาษา C ซึ่งภาษา C นี้เป็นลักษณะเฉพาะ คือมีการเขียนไลบรารีของ Arduino ขึ้นมาเพื่อให้การสั่งงานไมโครคอนโทรลเลอร์ที่แตกต่างกัน สามารถใช้งานโค้ดตัวเดียวกันได้ โดยตัวโครงการได้ออกบอร์ดทดลองมาหลายรูปแบบ เพื่อใช้งานกับ IDE ของตนเอง สาเหตุหลักที่ทำให้ Arduino เป็นนิยมมาก เป็นเพราะซอฟต์แวร์ที่ใช้งานร่วมกันสามารถโหลดได้ฟรี และตัวบอร์ดทดลองยังถูกแจกแปลน ทำให้ผู้ผลิตเงินนำไปผลิตและขายออกตลาดมาในราคาที่ถูกมากๆ

คำว่า Uno เป็นภาษาอิตาลี ซึ่งแปลว่าหนึ่ง เป็นบอร์ด Arduino รุ่นแรกๆที่ออกมา มีขนาดประมาณ 68.6x53.4mm เป็นบอร์ดมาตรฐานที่นิยมใช้งานมากที่สุด เนื่องจากเป็นขนาดที่เหมาะสมสำหรับการเริ่มต้นเรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ด Arduino รุ่นอื่นๆที่ออกแบบมาเฉพาะมากกว่า โดยบอร์ด Arduino Uno ได้มีการพัฒนาเรื่อยมา ตั้งแต่ R2 R3 และรุ่นย่อยที่เปลี่ยนชิปไอซีเป็นแบบ SMD



รูปที่ 3.3 Arduino UNO (ที่มา : Arduino.cc)

ข้อมูลจำเพาะ

ชิปไอซีไมโครคอนโทรลเลอร์	ATmega328
ใช้แรงดันไฟฟ้า	5V
รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ)	7 – 12V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6 – 20V
พอร์ต Digital I/O	14 พอร์ต (มี 6 พอร์ต PWM output)
พอร์ต Analog Input	6 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40mA
กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	50mA
พื้นที่โปรแกรมภายใน	32KB พื้นที่โปรแกรม, 500B ใช้โดย Bootloader
พื้นที่แรม	2KB
พื้นที่หน่วยความจำถาวร (EEPROM)	1KB
ความถี่คริสตัล	16MHz
ขนาด	68.6x53.4 mm
น้ำหนัก	25 กรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 Safety valve

เป็นอุปกรณ์ความปลอดภัยของระบบ มีหน้าที่หลักคือระบายแรงดันภายในระบบออก ซึ่งจะทำงานเองโดยอัตโนมัติเมื่อมีแรงดันเกินกว่าค่าที่ตั้งไว้ ซึ่ง Pressure Safety Valve จะใช้กับของเหลวที่บีบอัดได้ (Compressible Fluid) เช่น ไอน้ำ หรือ ก๊าซ ซึ่งต้องการระบายความดันอย่างรวดเร็ว สำหรับ Pressure Relief Valve จะใช้กับของเหลวที่บีบอัดไม่ได้ (Non Compressible Fluid) เช่น น้ำ หรือน้ำมัน ซึ่งจะระบายความดันอย่างช้าๆ โดยวาล์วทั้งสองชนิดมีส่วนประกอบที่คล้ายคลึงกันได้แก่

1. Valve Body: ส่วนใหญ่ผลิตจากเหล็กหล่อ หรือ วัสดุอื่นตามการใช้งาน โดยเป็นโครงสร้างที่ใช้ติดตั้งกับท่อหรือถังความดันที่จะทำการปกป้อง
2. Disc: ลักษณะเป็นแผ่นกลมที่ใช้กดปิดกั้นของเหลวไว้ โดยอาศัยแรงกดจากสปริง โดย Disc นี้จะทำหน้าที่รับแรงดันไว้ทั้งหมด
3. Stem: เป็นส่วนที่รับแรงกดจากสปริงและส่งแรงกดไปยัง Disc โดยมีหน้าที่เป็นแกนบังคับให้ส่วนที่เคลื่อนไหว เคลื่อนที่ตามแนวแกน
4. สปริง: เป็นส่วนที่ก่อให้เกิดแรงดันต้านทานความดันที่ Disc การปรับ Adjust Screw ให้สปริงยืดหดตัวแตกต่างกันส่งผลให้แรงกดที่ Disc มีค่าแตกต่างกัน จึงเป็นหลักการสำคัญที่ใช้กำหนดค่าความดันที่จะให้วาล์วทำการระบาย (Set Pressure)
5. Adjust Screw: ใช้ปรับระยะยืด หด ของสปริง จุดนี้เองที่ใช้ทำการปรับตั้งค่าความดัน



รูปที่ 3.4 Safety valve (ที่มา : Kingston Valves)

การทำงานของ Pressure Safety Valve โดยทั่วไปเพียงแค่เมื่อความดันภายในท่อหรือถังมีค่ามากกว่า Set Pressure แรงจากความดันที่กระทำต่อ Disc จะมีค่ามากกว่าแรงที่กดจากสปริง ทำให้ Disc เกิดการยกตัวขึ้น และระบายความดันส่วนเกินออกมา เมื่อความดันภายในระบบลดลงแล้ว Disc ก็จะไม่เคลื่อนที่เข้าสู่ตำแหน่งและรับความดันภายในเช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7 Thermocouple type K

เทอร์โมคัปเปิล (Thermocouple) หรือ หัววัดอุณหภูมิ ซึ่งตัวเทอร์โมคัปเปิลนี้ก็จะมียูหลากหลายชนิด หลากหลายแบบ ซึ่งก็ขึ้นอยู่กับความต้องการนำไปใช้ในลักษณะของการวัดที่แตกต่างต่างกันไป

เทอร์โมคัปเปิลมีหลาย Type ให้เลือก แล้วแต่ย่านอุณหภูมิ และลักษณะการใช้งาน โดยความแตกต่างของแต่ละ Type นี้เกิดจากการเลือกใช้คู่ของวัสดุ (Element) ของโลหะที่จะนำมาเชื่อมเข้าด้วยกันให้แตกต่างกัน เพราะโลหะแต่ละชนิดย่อมมีคุณสมบัติพิเศษเฉพาะตัวของมันอยู่แล้ว เมื่อโลหะชนิดต่าง ๆ กันมาจับคู่เชื่อมเข้าด้วยกัน จะทำให้คุณสมบัติของเทอร์โมคัปเปิล ที่ได้แตกต่างกันไป นอกจากนี้ ได้มีการทดลองผสมโลหะต่างชนิดเข้าด้วยกัน เพื่อปรับปรุงคุณสมบัติของโลหะเดิมให้ดีขึ้น หรือเพื่อใช้แทนโลหะบางชนิดที่ใช้ทำอยู่เดิม เช่น แพลทินัม เนื่องจากราคาสูง ตัวอย่างโลหะผสมที่เกิดขึ้น เช่น โครเมล (Cromel) คือโลหะผสมของ นิกเกิล 90% และ โครเมียม 10% ,อลูเมล (Alumel) คือ โลหะผสมของ นิกเกิล 95% อลูมิเนียม 2% แมงกานีส 2% และ ซิลิคอน 1% ,คอนสแตนแตน (Constantan) คือ โลหะผสมของ ทองแดง60% และ นิกเกิล40% เป็นต้น

หลักการทํางานของเทอร์โมคัปเปิล คือ อาศัยความแตกต่างของอุณหภูมิในการสร้างแรงเคลื่อนไฟฟ้าขึ้น การที่แรงเคลื่อนไฟฟ้าค่าหนึ่งจะอ้างอิงเป็นอุณหภูมิก่าหนึ่งได้ แสดงว่าความแตกต่างของอุณหภูมิที่เกิดขึ้นนั้นจะต้องอ้างอิงกับอุณหภูมิก่าคงที่ค่าหนึ่งเสมอ โดยเรียกอณหภูมิคงที่ที่ใช้อ้างอิงนี้ว่า Reference Junction และได้มีการกำหนด Reference Junction ให้เป็น 0 เพื่อให้การวัดอุณหภูมิเกิดแรงเคลื่อนไฟฟ้าที่เป็นมาตรฐานเดียวกัน และกำหนดเป็นตารางมาตรฐานแสดงค่าอุณหภูมิเทียบกับแรงเคลื่อนไฟฟ้าที่วัดได้ แต่โดยทั่วไป เทอร์โมคัปเปิลจะทำการวัดที่อุณหภูมิห้อง (เช่น 25 °C) นั่นคือไม่ได้เทียบกับ 0 °C แสดงว่าค่าแรงเคลื่อนไฟฟ้าที่ได้ยังไม่ถูกต้อง หากนำไปอ่านค่าอุณหภูมิจากตารางมาตรฐานจะผิดพลาด จึงจำเป็นต้องมีการรักษา Reference Junction เพื่อให้การวัดอุณหภูมิเทียบกับ 0 °C ตลอดเวลา



รูปที่ 3.5 Thermocouple type K (ที่มา : LeLong, 2010)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเทอร์โมคัปเปิลแบบ K เป็นเทอร์โมคัปเปิลชนิดที่นิยมใช้แพร่หลายมากที่สุด สามารถวัดอุณหภูมิได้สูงกว่าแบบ J และมีราคาถูกกว่า ทนอุณหภูมิได้ถึง 1300°C และที่อุณหภูมิต่ำถึง -250°C มีค่าความเป็นเชิงเส้นสูงที่สุดเมื่อเปรียบเทียบกับเทอร์โมคัปเปิลชนิดอื่น ให้แรงดันไฟฟ้าทางด้านเอาต์พุตสูง (ให้อัตราการเปลี่ยนแรงดันไฟฟ้าต่ออุณหภูมิดีกว่าแบบอื่นหรือมีค่าความชันใกล้เคียง 1) สามารถใช้กับงานที่มีการแผ่รังสีความร้อน (thermal radiation) ได้ ไม่เหมาะกับการใช้งานในสภาวะสุญญากาศ (ยกเว้นการใช้งานในช่วงเวลาสั้น)

3.1.8 Solid State Relay Industrial, 1-Phase Analog Switching RM1E60AA25

รีเลย์ (Relay) คือ อุปกรณ์ที่ทำหน้าที่เป็นสวิตช์ตัด-ต่อวงจร โดยใช้แม่เหล็กไฟฟ้า ซึ่งปกติถ้ามีการจ่ายไฟเข้าที่ตัวรีเลย์ (Relay) จะทำให้ขดลวดเหนี่ยวนำหน้าสัมผัสติดกัน จึงมีสถานะปิดวงจร (Closed Circuit) แต่ถ้าหากไม่มีการจ่ายไฟให้ รีเลย์ (Relay) ขดลวดเหนี่ยวนำหน้าสัมผัสไม่ติดกัน จะมีสถานะเปิดวงจร (Open Circuit)

โซลิดสเตตรีเลย์ (Solid State Relay) หรือตัวย่อ SSR คืออุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ ที่ไม่ใช้หน้าสัมผัสในการตัด-ต่อวงจร โดยใช้เทคโนโลยีของ เซมิคอนดักเตอร์ (Semiconductor) ที่ไม่มีส่วนเคลื่อนที่ จึงไม่มีเสียงในขณะเวลาตัด-ต่อของหน้าสัมผัส (Contact)

โดยทั่วไปนิยมใช้ โซลิดสเตตรีเลย์ (Solid State Relay) ในการควบคุมการทำงานของ Resistive Load เช่น ฮีตเตอร์ (Heater), หลอดไฟ หรือ Inductive Load เช่น Motor และมักจะเลือกใช้ Solid State Relay แทน Relay ทั่วไปใช้กับงานที่มีความถี่ในการตัดต่อบ่อย เพื่อลดปัญหาการสึกหรอของหน้าสัมผัส (Contact), การเกิด Arc และอายุการใช้งานของอุปกรณ์

นอกจากนี้ยังมีชนิดของ Solid State Relay ให้เลือกหลายหลากแบบ เช่น Solid State Relay สำหรับแรงดันไฟ DC (Solid State DC), Solid State Relay แบบเรจ-หรี (Phase Angle control), Solid State Relay สำหรับใช้กับแผง PCB และมีให้เลือกทั้งแบบ 1 เฟส, 2 เฟส และ 3 เฟส

สำหรับ RM1E60AA25 เป็นรีเลย์ที่ใช้ควบคุมการทำงานแบบอนาล็อก 230 VAC และมีอิมพุตควบคุมแบบอนาล็อก 4 – 20 mA, 0-10 VDC มีBlocking voltage 650 Vp โดยรีเลย์นี้จะทำงานที่ 90-230 VAC

3.1.9 RTD ชนิด PT100

RTD (Resistance Temperature Detector) คือ อุปกรณ์วัดอุณหภูมิโดยใช้หลักการเปลี่ยนแปลงค่าความต้านทานของลวดโลหะไปตามอุณหภูมิที่เปลี่ยนแปลงไปลวดโลหะที่นำมาใช้ผลิต KTD มี

หลายชนิดเช่น ทองแดง ทังสเตน นิเกิล แพรทินัม เนื่องจาก RTD แบบแพลทินัม 100 โอห์ม เป็นที่นิยมสูงสุด เพราะมีความเป็นเชิงเส้นที่ดีจึงถูกกำหนดให้เป็นแบบมาตรฐานที่ใช้ในห้องปฏิบัติการ ทั่วไป สำหรับ RTD แบบแพลทินัม 100 โอห์ม จะเปลี่ยนค่าความต้านทานโดยเฉลี่ย 0.385 โอห์ม ต่อ 1 องศาเซลเซียส การใช้งานปกติจะมีแหล่งจ่ายกระแสอยู่ที่ 1 mA ดังนั้นเมื่ออุณหภูมิเปลี่ยนไป 1 องศาเซลเซียส แรงดันจะเปลี่ยนไป 0.385 mV ซึ่งมากกว่า Thermocouple ถึง 10 เท่า ดังนั้นจึง เหมาะสำหรับวัดค่าอุณหภูมิที่ต้องการความละเอียดสูงและอุณหภูมิต่ำจนถึงติดลบ ค่าความต้านทานของลวดโลหะจะเปลี่ยนแปลงตามอุณหภูมิดังสมการ

$$dR_t / dt = \alpha R_0 \quad (3.1)$$

$$R_t = R_0(1 + \alpha T) \quad (3.2)$$

เมื่อ R_t = ค่าความต้านทานของลวดโลหะที่อุณหภูมิ t °C

R_0 = ค่าความต้านทานของลวดโลหะที่อุณหภูมิ 0 °C

α = สัมประสิทธิ์ของการเปลี่ยนแปลงค่าความต้านทานไฟฟ้าต่ออุณหภูมิ 1 °C

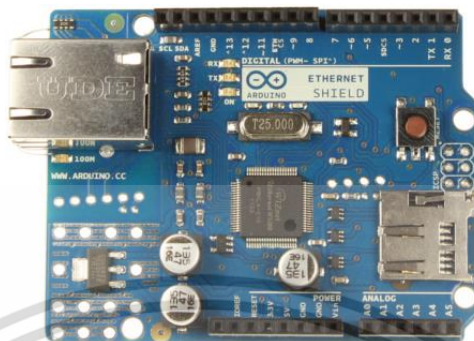
เนื่องจากตัว RTD เป็นเพียงค่าความต้านทาน จึงต้องมีวงจรจ่ายกระแสเพื่อให้เกิด เป็นแรงดันที่เปลี่ยนไป แล้วจึงนำแรงดันนี้ไปใช้งาน แต่กระแสจำนวนนี้ก็เป็นการสร้างความร้อนขึ้นในตัว RTD ด้วย ทำให้ค่าความต้านทานสูงขึ้น จึงจำเป็นที่จะต้องจำกัดไม่ให้กระแสเลี้ยง RTD นี้มีค่าสูงเกินไปในส่วนของช่วงการใช้งานสูงสุดของ RTD ชนิด PT00 นั้นจะอยู่ที่ -250 ถึง 600 องศาเซลเซียส ซึ่งช่วงการใช้งานและความไวในการวัดของ RTD แต่ละตัวจะแตกต่างกันตามแต่ละบริษัทจะกำหนด



รูปที่ 3.6 RTD ชนิด PT100 (ที่มา : IndiaMART, 2017)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.10 Arduino Ethernet Shield



รูปที่ 3.7 Arduino Ethernet Shield (ที่มา : arduino.cc)

Arduino Ethernet Shield V1 เชื่อมต่อ Arduino กับอินเทอร์เน็ตในเวลาเพียงไม่กี่นาที เพียงเสียบโมดูลนี้เข้ากับบอร์ด Arduino เชื่อมต่อกับเครือข่ายด้วยสายเคเบิล RJ45 ทุกองค์ประกอบของแพลตฟอร์มพร้อมใช้งานได้อย่างอิสระและโอเพ่นซอร์ส ซึ่งสามารถเรียนรู้วิธีการสร้างและใช้การออกแบบวงจรเอง

- ต้องใช้บอร์ด Arduino (ไม่รวม)
- แรงดันไฟฟ้า 5V (จ่ายมาจาก Arduino Board)
- Ethernet Controller: W5100 พร้อมบัฟเฟอร์ 16K ภายใน
- ความเร็วในการเชื่อมต่อ: 10/100Mb
- การเชื่อมต่อกับ Arduino บนพอร์ต SPI

Arduino Ethernet Shield ช่วยให้บอร์ด Arduino สามารถเชื่อมต่อกับอินเทอร์เน็ต ด้วยชิปอีเธอร์เน็ต Wiznet W5100 ซึ่งมอบเครือข่าย (IP) สแต็กที่มีความสามารถทั้ง TCP และ UDP รองรับการเชื่อมต่อซ็อกเก็ตพร้อมกันสูงสุดสี่ช่อง มีการเชื่อมต่อตามมาตรฐาน RJ-45 พร้อมสายหม้อแปลงในตัวและเปิดใช้งาน Power over Ethernet ใช้ไลบรารี Ethernet เพื่อเขียนภาพร่างที่เชื่อมต่อกับอินเทอร์เน็ต โดยใช้ตัว Ethernet Shield เชื่อมต่อกับบอร์ด Arduino โดยใช้หัวต่อแบบลวดพันยาวซึ่งยื่นผ่าน Shield ทำให้เลย์เอาต์ของหมุดยังคงอยู่และสามารถวาง Shield อื่น ๆ ได้

การปรับปรุงเวอร์ชันล่าสุดจะเพิ่มช่องเสียบการ์ด micro-SD ซึ่งสามารถใช้ในการจัดเก็บไฟล์ เพื่อให้บริการผ่านเครือข่าย มันเข้ากันได้กับ Arduino Duemilanove และ Mega (ใช้ไลบรารี Ethernet ที่มาใน Arduino) ไลบรารีการ์ด SD ยังไม่รวมอยู่ในมาตรฐานการจำหน่าย Arduino แต่สามารถทำงานได้ดี

การแก้ไขล่าสุดยังรวมถึงตัวควบคุมการรีเซ็ตเพื่อให้แน่ใจว่าโมดูลอีเธอร์เน็ต W5100 ถูกรีเซ็ตอย่างถูกต้องเมื่อเปิดเครื่อง การปรับปรุงเกราะก่อนหน้าไม่สามารถใช้งานร่วมกับเมกะได้และจำเป็นต้องรีเซ็ตด้วยตนเองหลังจากเปิดเครื่อง

แผงป้องกันในปัจจุบันมีโมดูล Power over Ethernet (PoE) ที่ออกแบบมาเพื่อดึงพลังงานจากสายอีเธอร์เน็ตคู่บิดเกลียวประเภท CAT 5:

- เป็นไปตามมาตรฐาน IEEE802.3af
- ระลอกคลื่นต่ำและเสียงรบกวน (100mVpp)
- ช่วงแรงดันไฟฟ้าอินพุต 36V ถึง 57V
- ป้องกันการโอเวอร์โวลตและการลัดวงจร
- เอาท์พุท 9V
- ตัวแปลง DC / DC ประสิทธิภาพสูง: typ 75% @ 50% load
- การแยก 1500V (อินพุตไปยังเอาต์พุต)

หมายเหตุ: โมดูล Power over Ethernet เป็นฮาร์ดแวร์ที่เป็นกรรมสิทธิ์ซึ่งไม่ได้ทำโดย Arduino แต่เป็นอุปกรณ์เสริมของบุคคลที่สาม

Arduino สื่อสารกับทั้ง W5100 และการ์ด SD โดยใช้บัส SPI (ผ่านส่วนหัว ICSP) คือหมุดดิจิตอล 11, 12 และ 13 บน Duemilanove และหมุด 50, 51 และ 52 บน Mega บนทั้งสองบอร์ดใช้พิน 10 เพื่อเลือก W5100 และพิน 4 สำหรับการ์ด SD หมุดเหล่านี้ไม่สามารถใช้กับ i/o ทั่วไปได้ บน Mega ฮาร์ดแวร์ SS pin, 53, ไม่ได้ใช้เพื่อเลือก W5100 หรือการ์ด SD แต่ต้องเก็บไว้เป็นเอาต์พุตหรืออินเทอร์เฟซ SPI จะไม่ทำงาน

เนื่องจาก W5100 และการ์ด SD แชนแนล SPI จึงสามารถใช้งานได้ครั้งละหนึ่งอันเท่านั้น หากใช้อุปกรณ์ต่อพ่วงทั้งสองในโปรแกรมต้องใช้ไลบรารีที่เกี่ยวข้องเพื่อจัดการ หากไม่ได้ใช้อุปกรณ์ต่อพ่วงตัวใดตัวหนึ่งในโปรแกรมจะต้องยกเลิกการเลือกอย่างชัดเจน ในการทำเช่นนี้กับการ์ด SD ให้ตั้งค่า pin 4 เป็นเอาต์พุตและเขียน high ไว้ สำหรับ W5100 ให้ตั้งค่าพินดิจิตอล 10 เป็นเอาต์พุต high

แผงป้องกันมีแจ็คอีเธอร์เน็ต RJ45 มาตรฐาน

ปุ่มรีเซ็ตบนตัวป้องกันจะรีเซ็ตทั้ง W5100 และบอร์ด Arduino

Shield มีไฟ LED ให้ข้อมูลจำนวนหนึ่ง:

PWR : ระบุว่าบอร์ดและแผงกำลังขับเคลื่อน

LINK : ระบุการมีอยู่ของลิงค์เครือข่ายและกระพริบเมื่อเกราะส่งหรือรับข้อมูล

FULLD : ระบุว่าเชื่อมต่อเครือข่ายนั้นเป็นแบบ full duplex

100M : ระบุว่ามีการเชื่อมต่อเครือข่าย 100 Mb/s (ตรงข้ามกับ 10Mb/s)

RX : กะพริบเมื่อเกราะรับข้อมูล

TX : กะพริบเมื่อเกราะส่งข้อมูล

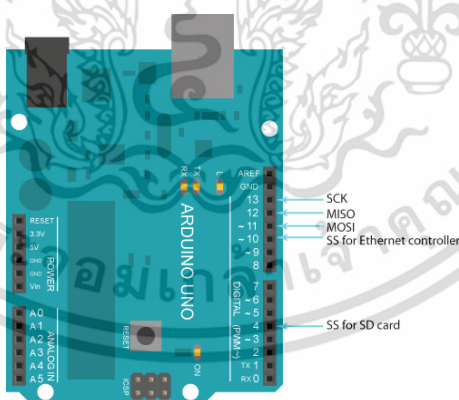
COLL : กะพริบเมื่อตรวจพบการชนกันของเครือข่าย

จัมเปอร์บัดกรีที่ทำเครื่องหมายว่า "INT" สามารถเชื่อมต่อเพื่อให้บอร์ด Arduino ได้รับการแจ้งเตือนเหตุการณ์ที่เกิดขึ้นจาก W5100 แต่การเชื่อมต่อนี้ไม่ได้รับการสนับสนุนจากไลบรารีอีเธอร์เน็ตจัมเปอร์เชื่อมต่อขา INT ของ W5100 กับขาดิจิตอล 2 ของ Arduino

Ethernet library

ไลบรารีนี้ถูกออกแบบมาเพื่อทำงานกับ Arduino Ethernet Shield, Arduino Ethernet Shield 2, Leonardo Ethernet และอุปกรณ์อื่น ๆ ที่ใช้ W5100 / W5200 / W5500 ไลบรารีนี้อนุญาตให้บอร์ด Arduino เชื่อมต่อกับอินเทอร์เน็ต บอร์ดสามารถทำหน้าที่เป็นเซิร์ฟเวอร์ที่ยอมรับการเชื่อมต่อขาเข้าหรือไคลเอนต์ที่ทำการเชื่อมต่อ ไลบรารีสนับสนุนมากถึงแปด (W5100 และบอร์ดที่มี ≤ 2 kB SRAM ถูกจำกัดที่สี่) การเชื่อมต่อพร้อมกัน (ขาเข้าขาออกหรือชุดค่าผสม)

บอร์ด Arduino สื่อสารกับตัว Shield โดยใช้บัส SPI นี่คือหมุดดิจิตอล 11, 12 และ 13 บน Uno และหมุด 50, 51 และ 52 บนเมกะ บนกระดานทั้งสองใช้พิน 10 เป็น SS บน Mega, ฮาร์ดแวร์ SS pin, 53, ไม่ได้ใช้เพื่อเลือกชิปคอนโทรลเลอร์อีเธอร์เน็ต แต่จะต้องเก็บไว้เป็นเอาท์พุทมิฉะนั้นอินเทอร์เน็ต SPI จะไม่ทำงาน



รูปที่ 3.8 รูปแสดงขาที่ใช้สื่อสารบนบอร์ด Arduino (ที่มา : arduino.cc)

ตัวอย่าง code การใช้งาน Arduino Ethernet Shield

```
#include <SPI.h>
#include <Ethernet.h>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// the media access control (ethernet hardware) address for the shield:
```

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
//the IP address for the shield:
```

```
byte ip[] = { 10, 0, 0, 177 };
```

```
void setup()
```

```
{
```

```
  Ethernet.begin(mac, ip);
```

```
}
```

```
void loop () {}
```

3.1.11 NodeMCU

NodeMCU ถูกสร้างขึ้นไม่นานหลังจาก ESP8266 ออกมา เมื่อวันที่ 30 ธันวาคม 2013, ระบบ Espressif เริ่มผลิต ESP8266 NodeMCU เริ่มเมื่อวันที่ 13 ตุลาคม 2014 เมื่อ Hong ส่งไฟล์แรกของ nodemcu-firmware ให้กับ GitHub อีกสองเดือนต่อมาโครงการขยายไปถึงแพลตฟอร์มฮาร์ดแวร์แบบเปิดเมื่อผู้พัฒนา Huang R ส่งไฟล์ gerber ของบอร์ด ESP8266 ชื่อ devkit v0.9 ต่อมาในเดือนนั้น Tuan PM ได้ทำการย้ายห้องสมุดลูกค้า MQTT จาก Contiki ไปยังแพลตฟอร์ม SoC ESP8266 และมุ่งมั่นที่จะทำโครงการ NodeMCU จากนั้น NodeMCU ก็สามารถรองรับ MQTT IoT โพรโตคอล Lua เพื่อเข้าถึงหน้า MQTT มีการอัปเดตที่สำคัญอีกอย่างในวันที่ 30 มกราคม 2558 เมื่อ Devsaurus ทำการย้าย u8glib ไปยังโครงการ NodeMCU, ทำให้ NodeMCU สามารถขับเคลื่อนจอแอลซีดี, หน้าจอ, OLED และ VGA ได้อย่างง่ายดาย ในช่วงฤดูร้อนปี 2558 ผู้สร้างดั้งเดิมได้ละทิ้งโครงการเฟิร์มแวร์และกลุ่มผู้สนับสนุนได้เข้ามารับช่วงต่อ ในช่วงฤดูร้อนปี 2559 NodeMCU ได้รวมโมดูลที่แตกต่างกันมากกว่า 40 โมดูล

NodeMCU (Node MicroController Unit) เป็นซอฟต์แวร์โอเพ่นซอร์สและสภาพแวดล้อมในการพัฒนาฮาร์ดแวร์ที่สร้างขึ้นในราคาที่ไม่แพงมาก System-on-a-Chip (SoC) ที่เรียกว่า ESP8266 ซึ่งได้รับการออกแบบและผลิตโดย Espressif Systems ประกอบด้วยองค์ประกอบที่สำคัญทั้งหมดของคอมพิวเตอร์สมัยใหม่: CPU, RAM, ระบบเครือข่าย (wifi) และแม้แต่ระบบปฏิบัติการที่ทันสมัย และ SDK ทำให้เป็นตัวเลือกที่ยอดเยี่ยมสำหรับโครงการ IoT ทุกชนิด

อย่างไรก็ตามในชิป ESP8266 นั้นยากต่อการเข้าถึงและใช้งาน ต้องบัดกรีสายไฟเอง ใช้แรงดันไฟฟ้าอะนาล็อกที่เหมาะสมไปยัง PIN และต้องเขียนโปรแกรมในคำสั่งเครื่องระดับต่ำที่สามารถตีความได้โดยฮาร์ดแวร์ชิป แม้ว่าปัญหานี้จะไม่เกิดขึ้นเมื่อใช้ ESP8266 เป็นชิปคอนโทรลเลอร์แบบฝังในอุปกรณ์อิเล็กทรอนิกส์ที่ผลิตเป็นจำนวนมากแต่เป็นปัญหาใหญ่สำหรับมือสมัครเล่น แยกเกอร์ หรือนักเรียนที่ต้องการทดลองใช้ในโครงการ IoT ของตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการ NodeMCU มีจุดประสงค์เพื่อลดความซับซ้อนของการพัฒนา ESP8266 มีองค์ประกอบที่สำคัญคือ

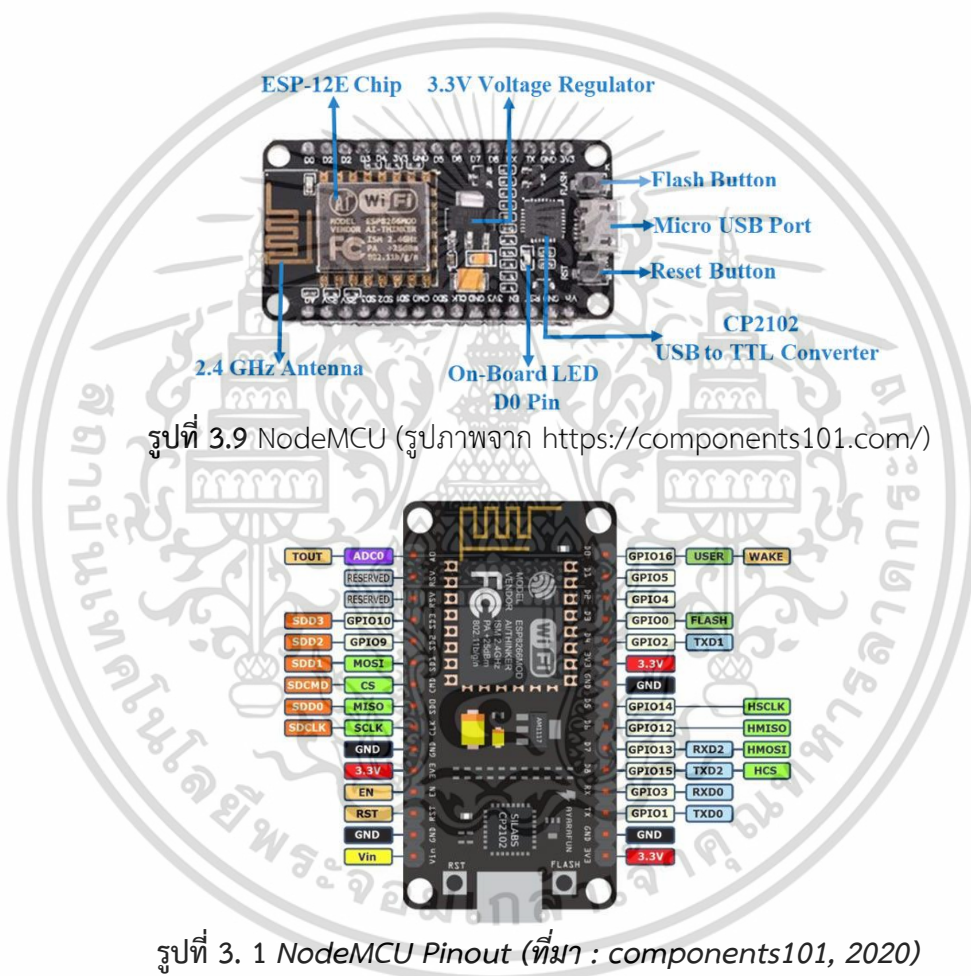
- เฟิร์มแวร์โอเพ่นซอร์ส ESP8266 ที่สร้างขึ้นจาก SDK ที่เป็นกรรมสิทธิ์ของผู้ผลิตชิป เฟิร์มแวร์จัดเตรียมการเขียนโปรแกรมอย่างง่ายตาม eLua (แบบฝังตัว Lua) ซึ่งเป็นภาษาสคริปต์ที่ใช้ง่ายและรวดเร็วสำหรับนักพัฒนา และ นั้นง่ายต่อการเรียนรู้สำหรับผู้มาใหม่
- บอร์ด DEVKIT ที่รวมชิป ESP8266 ไว้ในแผงวงจรมาตรฐาน บอร์ดนี้มีพอร์ต USB ในตัวที่ต่อสายไว้กับชิปแล้ว, ปุ่มรีเซ็ตฮาร์ดแวร์, เสาอากาศ wifi, ไฟ LED และหมุด GPIO ขนาดมาตรฐาน

NodeMCU ESP8266 ข้อมูลจำเพาะและคุณสมบัติ

- ไมโครคอนโทรลเลอร์: Tensilica RISC CPU Xtensa LX106 32 บิต
- แรงดันไฟฟ้า: 3.3V
- แรงดันไฟฟ้าอินพุต: 7-12V
- Digital I/O Pins (DIO): 16
- อนาล็อกอินพุตพิน (ADC): 1
- UART: 1
- SPIs: 1
- I2Cs: 1
- หน่วยความจำแฟลช: 4 MB
- SRAM: 64 KB
- ความเร็วสัญญาณนาฬิกา: 80 MHz
- USB-TTL ตาม CP2102 รวมอยู่ในบอร์ดเปิดใช้งาน Plug n Play
- เสาอากาศ PCB
- รองรับมาตรฐาน IEEE802.11 b/g/n
- โมดูลขนาดเล็กเพื่อให้พอดีกับโครงการ IoT ของคุณอย่างชาญฉลาด
- TCP/IP Stack
- วงจรสวิตช์ TR, สายอากาศที่มี Balun, LNA หรือวงจรขยายสัญญาณรบกวนต่ำ, วงจรขยายกำลังและแมตชิ่งเน็ตเวิร์กเพื่อการรับคลื่นวิทยุอย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรเฟสล็อกกลุ๊ปและวงจรควบคุมสัญญาณออสซิลเลเตอร์แบบดิจิทัล (DCXO) เพื่อช่วยบริหารจัดการด้านความถี่
- วงจรบริหารกำลังไฟฟ้าและวงจรควบคุมไฟเลี้ยงคงที่ เพื่อช่วยให้วงจร WiFi ใช้กำลังได้อย่างเหมาะสม โดยปกติต้องการกระแสไฟฟ้า 0.9 mA ในขณะสแตนด์บาย, 135-215 mA ขณะส่งข้อมูล, 60 mA เมื่อรับข้อมูล, 1 mA ในโหมดประหยัดพลังงาน และ 0.5 μ A ในขณะปิด
- กำลังส่ง +19.5dBm เมื่อทำงานในโหมด 802.11b



ตารางที่ 3.2 การกำหนดค่า Pinout ของ NodeMCU Development Board

พิน หมวดหมู่	ชื่อ	อธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power	Micro-USB, 3.3V, GND, Vin	Micro-USB: NodeMCU สามารถขับเคลื่อนผ่านพอร์ต USB 3.3V: ควบคุมจ่ายไฟให้กับบอร์ด 3.3V GND: Ground pins Vin: แหล่งจ่ายไฟภายนอก
Control Pins	EN, RST	หมุดและปุ่มรีเซ็ตไมโครคอนโทรลเลอร์
Analog Pin	A0	ใช้สำหรับวัดแรงดันไฟฟ้าอนาล็อกในช่วง 0-3.3V
GPIO Pins	GPIO1 to GPIO16	อินพุทเอาต์พุททั่วไป 16 จุดบนบอร์ด
SPI Pins	SD1, CMD, SD0, CLK	พินพร้อมใช้งานสำหรับการสื่อสาร SPI.
UART Pins	TXD0, RXD0, TXD2, RXD2	อินเตอร์เฟซ UART คือ UART0 (RXD0 & TXD0) และ UART1 (RXD1 & TXD1) UART1 ใช้เพื่ออัปโหลดเฟิร์มแวร์/โปรแกรม
I2C Pins		ฟังก์ชันรองรับ I2C แต่เนื่องจากฟังก์ชันการทำงานภายในของพินเหล่านี้ต้องค้นหาพินที่เป็น I2C

(ที่มา : components101, 2020)

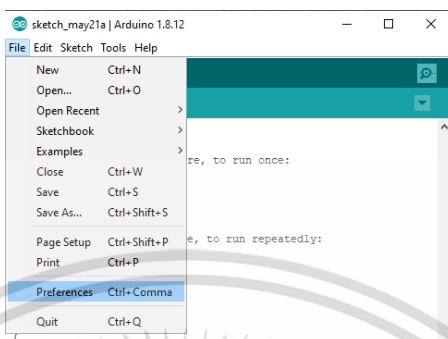
เริ่มต้นการใช้งาน

1. ติดตั้งโปรแกรม ArduinoIDE จาก

<https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>

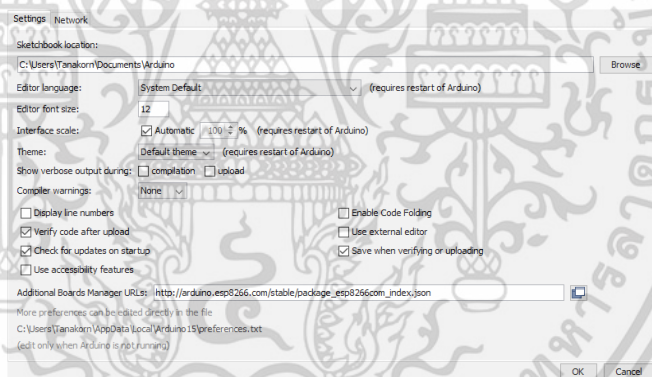
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เปิดโปรแกรม แล้วเลือก File -> Preferences เพื่อติดตั้งบอร์ด NodeMCU/ESP8266 แบบออนไลน์



รูปที่ 3.11 Preferences

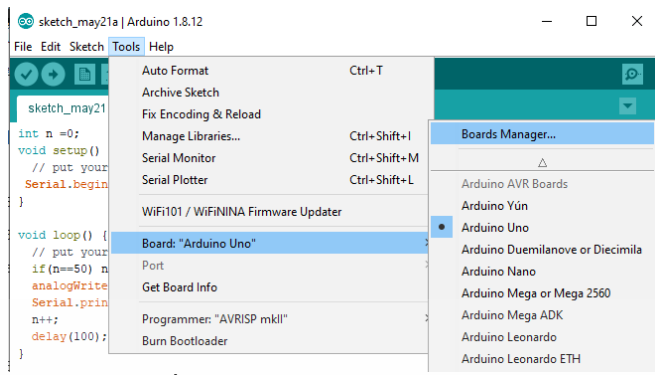
- หน้าต่าง Preferences จะปรากฏขึ้นมา เพิ่ม “http://arduino.esp8266.com/stable/package_esp8266com_index.json” ลงในช่อง Additional Boards Manager URLs ดังภาพแล้วกด Ok



รูปที่ 3.12 หน้าต่าง Preferences

- คลิกไปที่เมนู Tools -> Board -> Board Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



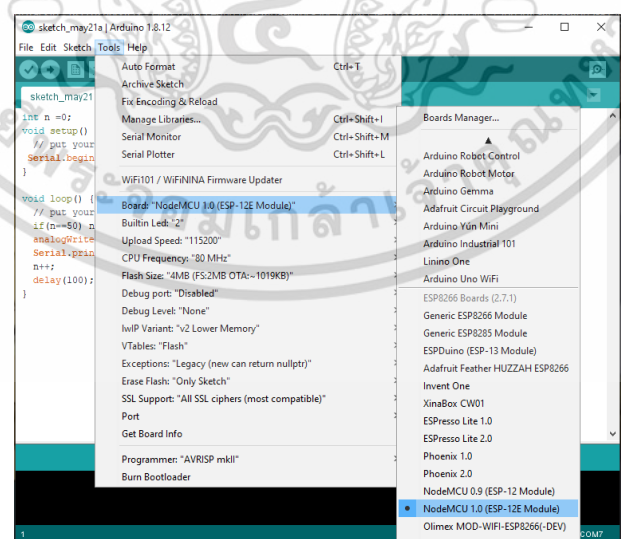
รูปที่ 3.13 Board Manager

5. พิมพ์คำว่า ESP8266 ลงในช่อง และเริ่มต้นติดตั้ง



รูปที่ 3.14 หน้าต่าง 1. Board Manager

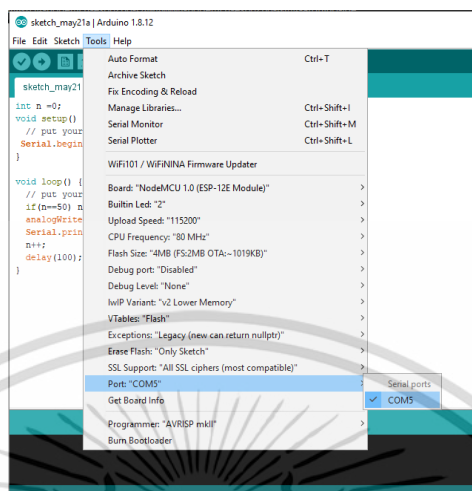
6. เลือกบอร์ด NodeMCU ESP8266 ที่จะอัปโหลด



รูปที่ 3.15 เลือกบอร์ด

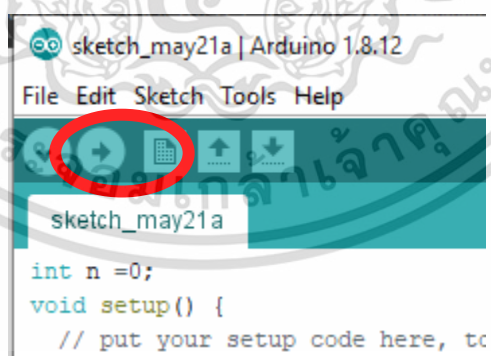
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลือก port ที่จะอัปโหลด



รูปที่ 3.16 เลือก port

8. เขียนโปรแกรม คอมไพล์ และอัปโหลดโปรแกรมลงบอร์ด



รูปที่ 3.17 คอมไพล์ และอัปโหลดโปรแกรม

การเชื่อมต่อ NodeMCU กับเครือข่าย WiFi

1. เปิดโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เขียนโค้ดดังรูป

```
#include <ESP8266WiFi.h>
const char* ssid    = "Your WiFi SSID";
const char* password = "Your Password";

void setup()
{
  Serial.begin(115200);
  Serial.println("Starting...");

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(250);
    Serial.print(".");
  }

  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop()
{
}
```

การกำหนด #include <ESP8266WiFi.h> เป็นการประกาศเรียกใช้ไลบรารี WiFi ของ ESP8266 (ESP8266WiFi.h) แล้วกำหนดค่า SSID และรหัสผ่านของเครือข่าย WiFi ที่ต้องการเชื่อมต่อ เก็บไว้ที่ตัวแปร ssid และ password ตามลำดับ

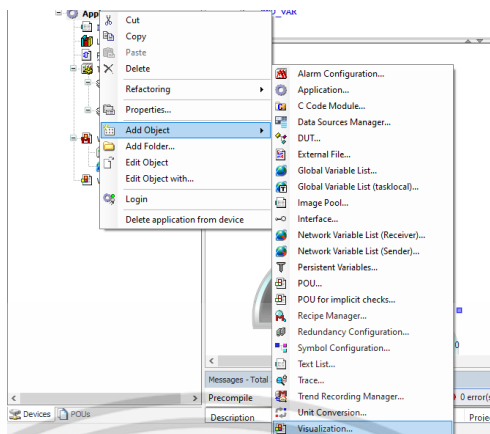
ส่วนภายในฟังก์ชัน setup() เริ่มจากตั้งค่าความเร็วเริ่มต้นของ Serial ด้วยคำสั่ง Serial.begin(ความเร็ว); ส่งพิมพ์คำว่า Serial.println("ข้อความ"); เพื่อแสดงข้อความที่ตั้งไว้ที่หน้าจอกอมพิวเตอร์ผ่าน Serial port และเริ่มทำการเชื่อมต่อกับเครือข่าย WiFi ด้วยคำสั่ง WiFi.begin จากนั้นคอยวนตรวจสอบว่าเชื่อมต่อสำเร็จหรือไม่ด้วยคำสั่ง WiFi.status หากยังไม่เชื่อมต่อให้หน่วงเวลา 250 ms โดยใช้คำสั่ง delay(เวลามิลลิวินาที); และแสดงจุดออกมาที่หน้าจอกอมพิวเตอร์ หากมีการเชื่อมต่อเรียบร้อยแล้ว ให้แสดงคำว่า WiFi connected และแสดงหมายเลข IP Address ของ NodeMCU/ESP82266 ที่ได้รับมาโดยใช้คำสั่ง WiFi.localIP หลังจากนั้นก็จะไปทำงานในโปรแกรมที่เขียนไว้ใน void loop

3.2 HMI

ในการสร้างโปรเจกใน CODESYS จะใช้ส่วนต่อประสานกับผู้ใช้ (user interface, UI) เพื่อสร้างส่วนติดต่อกันระหว่างผู้ใช้และโปรแกรมที่เหมาะสมสำหรับแอปพลิเคชัน วิธีนี้สามารถสร้างภาพเคลื่อนไหวและแสดงข้อมูล เมื่อสร้างจินตทัศน์ (Visualization) และแอปพลิเคชัน ใช้ฟังก์ชันทั่วไป เช่น การจัดการไลบรารี และซอร์สโค้ด หรือค้นหา/แทนที่ ตลอดทั้งโปรเจก

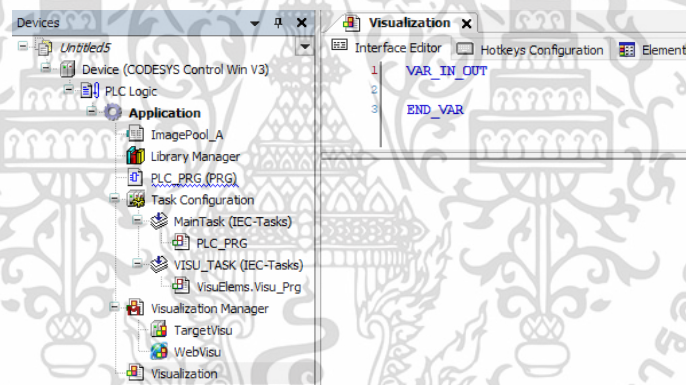
1. การเพิ่ม Visualization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 การเพิ่ม Visualization

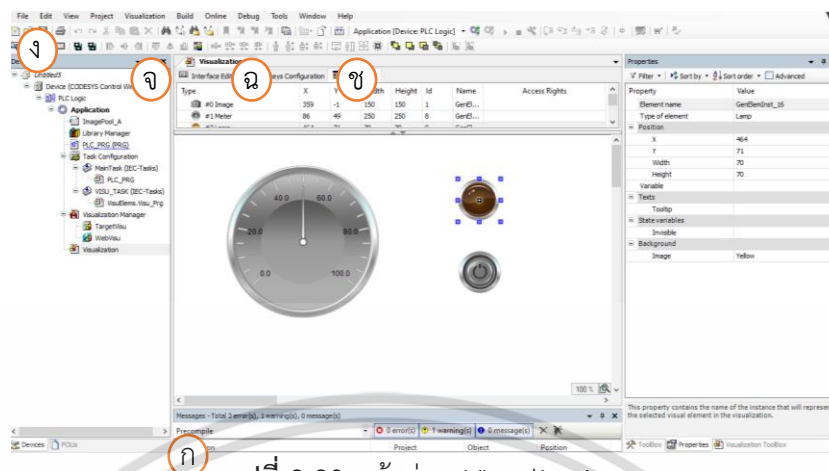
- คลิกขวาที่ Application ► Add Object ► Visualization
- ตั้งชื่อ Visualization
- เปิด Visualization โดยการดับเบิลคลิกที่ไอคอน Visualization



รูปที่ 3.19 ไอคอน Visualization

2. ออปเจ็ค และเครื่องมือของ Visualization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 หน้าต่าง Visualization

ก. ตัวแก้ไขแบบกราฟิก: สร้างภาพจากองค์ประกอบการต่างๆที่ให้ไว้ในเครื่องมือของ (ค) Tools)

ข. ดูเครื่องมือ: Elements ที่มีอยู่

ค. คุณสมบัติ: ตัวแก้ไขการกำหนดค่าสำหรับ Elements ที่เลือกในพื้นที่ตัวแก้ไข

ง. เมนู: คำสั่งสำหรับการทำงานในโปรแกรมสร้างภาพ

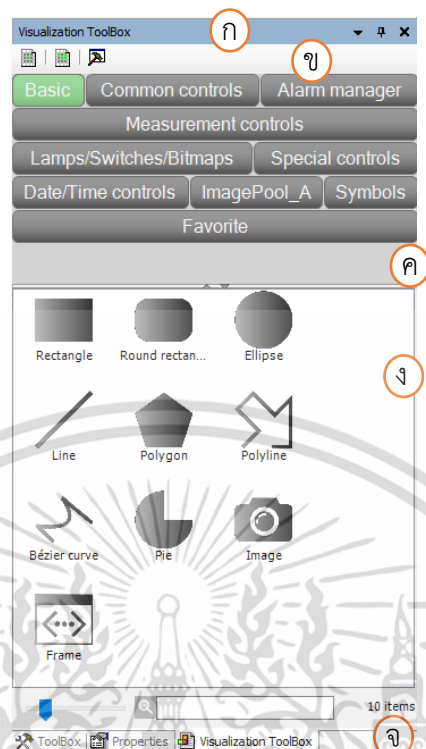
จ. Interface Editor: การประกาศตัวแปรซึ่งการอ้างอิงของการสร้างภาพสามารถกำหนดพารามิเตอร์ได้

ฉ. การกำหนดค่าปุ่มลัด: การกำหนดทางลัดสำหรับอินพุตในการสร้างภาพในโหมดออนไลน์

ช. รายการ Elements: รายการ Elements ทั้งหมดที่ใช้ในการสร้าง Visualization และเปลี่ยนตำแหน่งของในแกน z

View 'Toolbox'




มุมมองนี้จัดเตรียมองค์ประกอบที่สามารถใช้งานได้โปรแกรมแก้ไข องค์ประกอบแต่ละรายการมีตัวระบุบางอย่างเพื่อให้สามารถจัดกลุ่มได้ มีปุ่มสำหรับตัวระบุแต่ละตัว องค์ประกอบของตัวระบุที่เลือกจะแสดงด้วยรูปขนาดย่อและสามารถลากไปที่ตัวแก้ไข นอกเหนือจากตัวระบุมาตรฐานคุณยังสามารถกำหนดของคุณเอง คุณสามารถเปลี่ยนขนาดของรูปขนาดย่อด้วยตัวเลื่อนและการค้นหาข้อความแบบเต็มสำหรับชื่อองค์ประกอบเป็นไปได้



รูปที่ 3.21 กล่องเครื่องมือ Visualization

- ก. มุมมองกล่องเครื่องมือ
- ข. แถบเครื่องมือพร้อมคำสั่ง
- ค. หมวดหมู่องค์ประกอบ
- ง. การเลือก Elements สำหรับสร้าง Visualization
- จ. องค์ประกอบการควบคุม



แถบเครื่องมือพร้อมคำสั่ง

- สัญลักษณ์  สามารถเลือกได้เพียงปุ่มเดียวเท่านั้น
- สัญลักษณ์  สามารถเลือกได้หลายปุ่ม
- สัญลักษณ์  จัดการจัดการแคตตาล็อก และรายการเครื่องมือ

การเลือก Elements สำหรับสร้าง Visualization

- Slider  เพื่อเปลี่ยนขนาดของภาพย่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

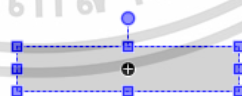
- ช่องป้อนข้อมูล  สำหรับการค้นหาข้อความแบบตามชื่อใน Elements ทั้งหมดที่มี
- จำนวนรายการ **10 items** จำนวนรายการของ Elements ที่แสดงในปัจจุบัน โดยการเลือกปุ่มหรือเนื่องจากคำค้นหาใน 

3.การออกแบบ Visualization ด้วย Elements

เครื่องมือ Visualization มีองค์ประกอบสำหรับการออกแบบส่วนต่อประสานผู้ใช้ใน “view Tools”

ลากองค์ประกอบที่ต้องการลงในหน้าต่างแก้ไขและปรับแต่งในส่วน “Properties” การออกแบบภาพ, การติดฉลาก, การแสดงข้อมูล, ปฏิกริยาต่ออินพุตของผู้ใช้, ความเป็นไปได้ในการป้อนค่า ฯลฯ สามารถกำหนดค่าคุณสมบัติแบบคงที่หรือแบบไดนามิกได้ เช่น การกำหนดค่าคงที่หรือตัวแปร แอปพลิเคชัน การกำหนดค่าแบบไดนามิกช่วยให้ภาพเคลื่อนไหวที่ดำเนินการตอนรันไทม์

3.1 การปรับตำแหน่ง ขนาด และเลเยอร์ของ Elements Visualization เป็นภาพแรสเตอร์เป็น พิกเซล ตำแหน่งพิกเซลถูกระบุในพิกัด x / y จุดเริ่มต้น $(0,0)$ ตั้งอยู่ที่มุมซ้ายบนของหน้าต่าง แกน x บวกวิ่งไปทางขวาและแกน y บวกวิ่งลงมา ตำแหน่งขององค์ประกอบบนแกน z ของ การสร้างภาพข้อมูลถูกควบคุมโดยตำแหน่งของอเล็กเมนต์ ขนาดและตำแหน่งของ Elements ถูกระบุเป็นพิกัดพิกเซลในมุมมองคุณสมบัติการตั้งค่าเหล่านี้จะแสดงแบบกราฟิก ในมุมมองโปรแกรมแก้ไขพร้อมกันเมื่อลากองค์ประกอบการสร้างภาพจากมุมมองกล่อง เครื่องมือไปยังมุมมองเครื่องมือแก้ไขจะแสดงตามที่เราเลือกไว้ดังตัวอย่างขององค์ประกอบ สีเหลี่ยมผืนผ้า





ตำแหน่งที่เป็นไปได้ขึ้นอยู่กับตารางที่กำหนด คุณสามารถเปลี่ยนการตั้งค่าตัวเลือกโคดีซีดี คำสั่ง ในเมนูพร้อมใช้งานสำหรับการจัดตำแหน่งและการจัดกลุ่ม

ตอนนี้สามารถย้ายหรือปรับขนาดองค์ประกอบได้โดยตรงในเครื่องมือแก้ไข กำหนดค่าคุณสมบัติ ตำแหน่งในตัวแก้ไขคุณสมบัติซึ่งจะเปิดขึ้นโดยอัตโนมัติสำหรับองค์ประกอบที่เลือก ตัวอย่างเช่น ในหน้า วิธีใช้สำหรับ “ปุ่ม” การเปลี่ยนแปลงจะได้รับการอัปเดตในตัวแก้ไขอื่น ๆ ด้วย

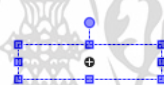
การเปลี่ยนขนาดและตำแหน่งองค์ประกอบในเครื่องมือแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

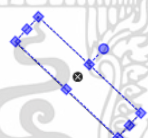
- ใช้เมาส์วางไว้ที่โฟกัสของ element จะแสดงสัญลักษณ์  และลากองค์ประกอบไปยังตำแหน่งใดก็ได้
- position ตำแหน่งขององค์ประกอบนั้นได้รับการปรับปรุงในคุณสมบัติตำแหน่ง ► X และตำแหน่ง ► Y
- โฟกัสบนที่กล่องสี่เหลี่ยม รูปร่างของตัวชี้เมาส์เป็นลูกศรคู่ที่ระบุทิศทางที่คุณสามารถลากกล่องเพื่อปรับขนาดกล่อง 
- ลากกล่องสีน้ำเงินเพื่อปรับขนาด element ตำแหน่งขององค์ประกอบนั้นได้รับการปรับปรุงในคุณสมบัติตำแหน่ง ► X และตำแหน่ง ► Y

นอกจากนี้คุณสามารถหมุนองค์ประกอบสี่เหลี่ยมผืนผ้าเส้นรูปหลายเหลี่ยมและพาย

- เลือกองค์ประกอบสำหรับการหมุนแบบคงที่ รูปจะปรากฏขึ้นพร้อมที่จับยึดจากกล่องตำแหน่งที่เคลื่อนย้ายได้



- ลากตัวชี้เมาส์ไปเหนือที่จับ เคอร์เซอร์จะปรากฏเป็นลูกศรหมุน  หมุนองค์ประกอบไปยังตำแหน่งใดก็ได้ในตำแหน่งคุณสมบัติ ► มุมมุมที่กำหนดจะแสดงเป็นองศา



แต่ละ element อยู่ในเลเยอร์ข้อมูล (แกน z) มันสามารถซ่อนได้โดยองค์ประกอบอื่น ๆ ในเบื้องหน้าและซ่อนองค์ประกอบอื่น ๆ ในพื้นหลัง ลำดับของเลเยอร์สามารถมองเห็นได้ในแท็บรายการองค์ประกอบด้านบน ลำดับขององค์ประกอบจากด้านหน้าไปด้านหลังระบุลำดับของเลเยอร์จากหลังไปข้างหน้า ใช้คำสั่งจาก “Properties” เพื่อย้ายองค์ประกอบที่เลือก

Position	
X	136
Y	61
Width	125
Height	30
Angle	43

รูปที่ 3.22 เมนูปรับขนาดและตำแหน่ง

1.2 การกำหนดสี

คุณกำหนดค่าสีของ element ทั้งแบบคงที่โดยใช้คุณสมบัติสี หรือแบบไดนามิกโดยการกำหนดตัวแปรแอพลิเคชันโดยวิธีการของคุณสมบัติตัวแปรสี ขึ้นอยู่กับองค์ประกอบการกำหนดสียังมีอยู่ในคุณสมบัติอื่น ๆ ตัวอย่างเช่น สำหรับสีแบบอักษรนี้ในคุณสมบัติข้อความของ element ที่มีข้อความ

สำหรับการกำหนดค่าสีแบบคงที่สามารถแก้ไขคุณสมบัติซึ่งมีงานสีให้เลือก สามารถระบุสีเป็นสีสไตล์ สีสไตล์ คือ ชื่อสีสำหรับคำจำกัดความสีจากสไตล์ที่ใช้ เมื่อกำหนดค่าคุณสมบัติที่สอดคล้องกันคุณจะได้รับรายการของสีสไตล์ที่พร้อมใช้งาน





นอกจากนี้สามารถกำหนดการเติมสีขององค์ประกอบเป็นการไล่ระดับสี จากนั้นสีจะเปลี่ยนเป็นเส้นตรง เรเดียน หรือตามแนวแกนจากสีเริ่มต้นเป็นสีสุดท้าย คุณกำหนดค่าการตั้งค่าการไล่ระดับสีในกล่องตัวไล่ระดับสี

การออกแบบ element ด้วยสีแบบคงที่

- แทรก element รูปร่าง
- เลือก element คลิก “Properties”



รูปที่ 3.24 เมนูตั้งค่าสีแบบคงที่

- คลิก Colors ► Normal state ► Fill color คลิกที่กล่องรายการและปุ่ม  ปรากฏขึ้น
- กำหนดสีให้กับรูปร่าง ตัวอย่างเช่นเลือก “Elementfillcolor” จากกล่องรายการ
- กำหนดระดับความโปร่งใสใน Colors ► Normal state ► Fill color ► Transparency
- เลือกรูปร่างอื่น คลิก Colors ► Normal state ► Fill color list  กล่องรายการและปุ่ม  ปรากฏขึ้น
- กำหนดค่าสีคงที่ให้กับรูปร่าง คลิก  กล่องตั้งค่าสีเปิดขึ้น
- เลือกสีมาตรฐานหรือกำหนดสีที่กำหนดเอง จากนั้นคลิก “Ok” สีถูกตั้งค่าเป็นค่าคงที่ สีจะปรากฏเป็นสีเหลี่ยมเล็กๆ ค่า RGB จะถูกระบุถัดจากค่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิก Colors ► Normal state ► Fill color ► Transparency
- ใช้แถบเลื่อนเพื่อทำให้สีเป็นแบบกึ่งโปร่งใส

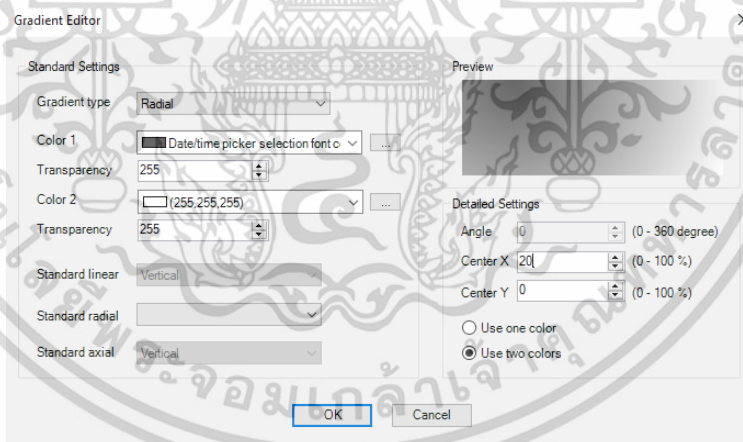
การออกแบบ element ด้วยการไล่ระดับสี

- ลาก element รูปร่างเพื่อสร้างภาพ



รูปที่ 3.24 เมนูตั้งค่าสีแบบไล่ระดับ

- เลือก Colors ► Use gradient color
- คลิก Colors ► Gradient setting กล้องตั้งค่าตัวไล่ระดับสีจะเปิดขึ้น
- กำหนดสีไล่ระดับสีสำหรับรูปร่าง ประเภทการไล่ระดับสี: เรเดียล, รัศมี, Center เดิมสีของจะเปลี่ยน



รูปที่ 3.25 หน้าต่างตั้งค่าสีแบบไล่ระดับ

3.2 การใช้ข้อความ

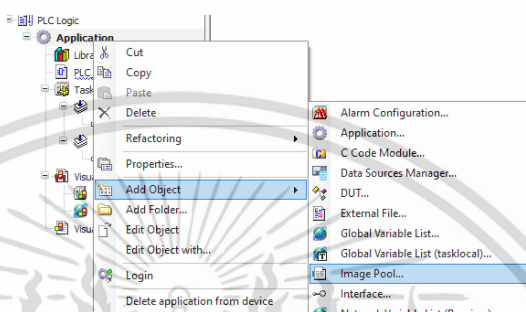
การแสดงผลข้อความใน element โดยกำหนด “string” ใน Texts ► Text นอกจากนี้ยังสามารถแสดงข้อความเป็นคำแนะนำเครื่องมือ (Texts ► Tooltip) ข้อความที่กำหนดในลักษณะนี้เป็นแบบคงที่สามารถการจัดการใน GlobalTextList ในมุมมอง POUs และสามารถเปลี่ยนแปลงได้ทั้งโดยโปรแกรมและโดยผู้ใช้ในระหว่างรันไทม์ visualization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามสามารถขยายข้อความแบบคงที่ได้ด้วยตัวยัดตำแหน่งหนึ่งตัวที่มีข้อกำหนดคุณสมบัติ การจัดรูปแบบเพื่อส่งออกเนื้อหาของตัวแปร ณ จุดนี้ ค่าปัจจุบันของตัวแปรที่คุณได้กำหนดไว้ของคุณสมบัติตัวแปรข้อความจะถูกส่งออกเมื่อรันใหม่

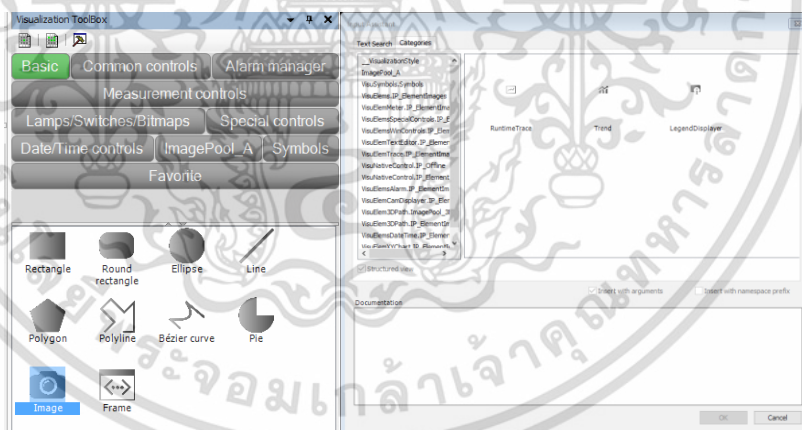
การติดป้าย element ภาพด้วยข้อความคงที่

- แทรก Image Pool ชื่อ ImagePool_A



รูปที่ 3.26 แทรก Image Pool

- เพิ่มรูปภาพไปที่ ImagePool_A
- เปิด visualization แล้วลาก element ภาพไปที่เครื่องมือแก้ไข เครื่องมือช่วยป้อนข้อมูลจะเปิดขึ้น ImagePool_A อยู่ในรายการในแท็บหมวดหมู่



รูปที่ 3.27 การเพิ่มรูปไปยัง Image Pool

- เลือกภาพหยุดแล้วคลิก “Ok”
- กำหนดค่าคุณสมบัติตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text properties	
Horizontal alignment	Centered
Vertical alignment	Centered
Text format	Default
Font	Default
Font color	Fontcolor

รูปที่ 3.29 กำหนดค่าคุณสมบัติ Image Pool

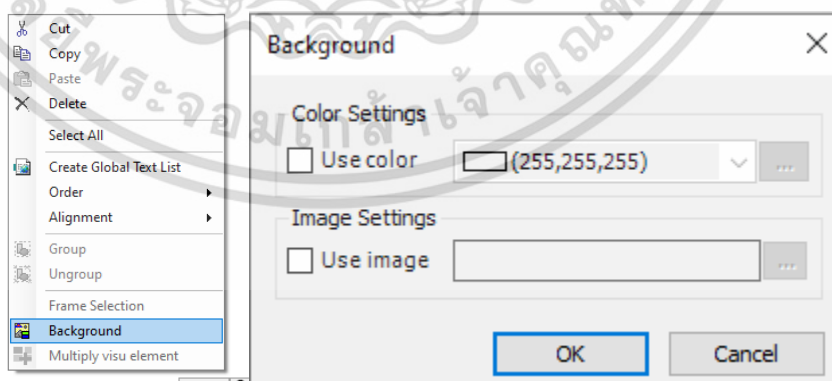
เอาต์พุตข้อความ: กำหนดค่าเอาต์พุตข้อความคงที่ และไดนามิก

เปิด visualization และเพิ่ม element ฟิลด์ข้อความ

- view มุมมองคุณสมบัติแสดงการกำหนดค่าขององค์ประกอบ
- กำหนดค่าคุณสมบัติ Texts ► Text : File name: %s ข้อความจะแสดงที่ตำแหน่ง % s
- ในแอปพลิเคชันใน POU PLC_PRG ให้ประกาศตัวแปรที่สอดคล้องกับ
`strFileName : STRING := 'File_A';`
- กำหนดค่าคุณสมบัติตัวแปรข้อความของปุ่มด้วย PLC_PRG.strFileName เนื้อหาของตัวแปร
 จะถูกส่งออกตอนที่รันใหม่

3.3 ออกแบบพื้นหลัง

สามารถออกแบบพื้นหลังของการด้วยสีหรือด้วยภาพ โดยใช้คลิกขวาบน Visualization ►
 Background .



รูปที่ 3.29 พื้นหลัง Visualization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

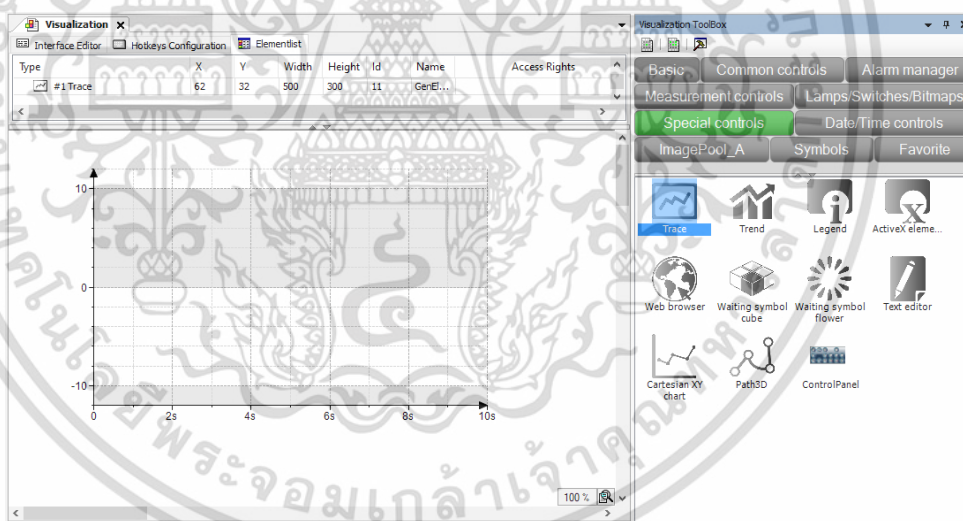
3.4 การแสดงเส้นโค้งข้อมูลด้วย “Trace”

ด้วยองค์ประกอบนี้คุณสามารถรวมกราฟการติดตามในการสร้างภาพที่ตรวจสอบและแสดงค่าตัวแปรอย่างถาวร คุณกำหนดค่ากราฟการติดตามที่แสดงในคุณสมบัติองค์ประกอบ นอกจากนี้คุณสามารถเพิ่มองค์ประกอบการควบคุมที่ควบคุมฟังก์ชันการติดตาม สิ่งนี้ทำได้ด้วยตนเองหรือโดยการใช้องค์ประกอบแทรกสำหรับการควบคุมคำสั่งติดตาม

วัตถุแสดงการติดตามข้อมูลในหนึ่งไดอะแกรมหรือมากกว่านั้น ในแอปพลิเคชันรันไทม์กราฟค่าของตัวแปรติดตามจะถูกบันทึกไว้ในคอนโทรลเลอร์ สิ่งเหล่านี้จะปรากฏในตัวแก้ไขการติดตามใน CODESYS เพื่อให้สามารถตรวจสอบกราฟค่า ความต้องการคือการตั้งค่าของการกำหนดค่าการติดตาม การถ่ายโอนการกำหนดค่าการติดตามไปยังตัวควบคุมและการเริ่มต้นของคุณสมบัติการติดตาม ข้อมูลที่บันทึกไว้จะถูกถ่ายโอนไปยังระบบการพัฒนาและแสดงเป็นไดอะแกรมตามการกำหนดค่า

การพัฒนา visualization ด้วย Trace

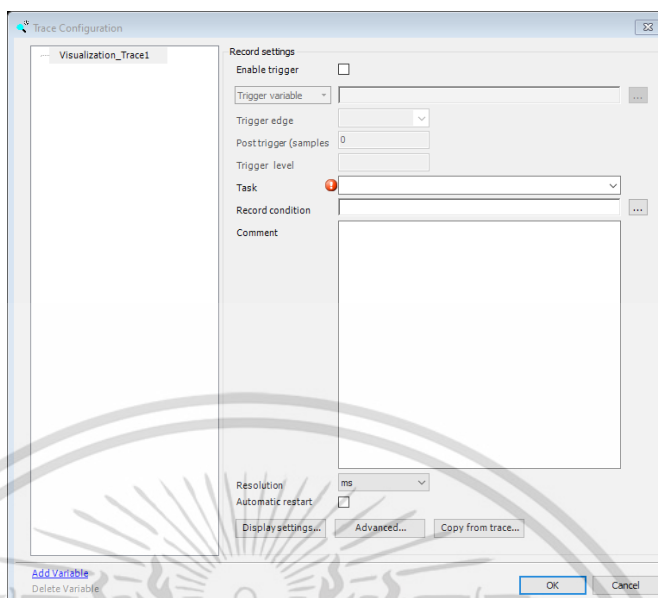
- ลาก “Trace” ที่อยู่ใน visualization toolbox ►special controls ลงในส่วนใช้งาน



รูปที่ 3.30 Trace

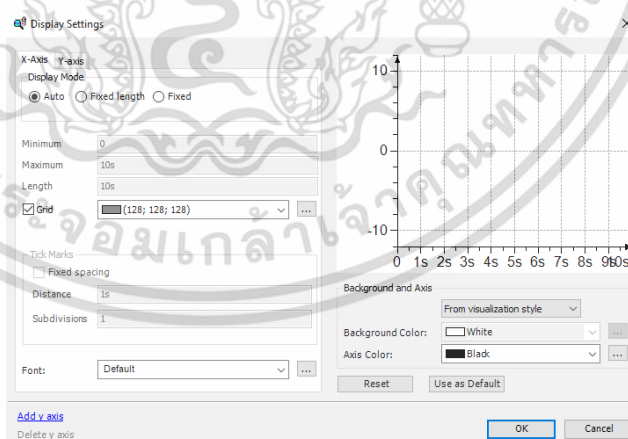
- เลือก properties ►trace ► เพื่อทำการเพิ่มตัวแปรและตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 หน้าต่างเพิ่มตัวแปรและตั้งค่า Trace

- Task ที่สำหรับการบันทึกข้อมูล
- Record condition เงื่อนไขการบันทึกที่แอปพลิเคชันบันทึกข้อมูลในโหมดรันใหม่
- Add variable เพิ่มตัวแปร
- Delete variable ลบตัวแปร
- Display Settings การกำหนดค่าสำหรับการตั้งค่าการแสดงผลของไดอะแกรมการติดตาม (สำหรับแกน x และแกน y) และแสดงตัวอย่างในไดอะแกรมการแสดงผล



รูปที่ 3.32 หน้าต่างตั้งค่า Display trace

- Advanced Trace Settings กำหนดอัตราการบันทึกของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

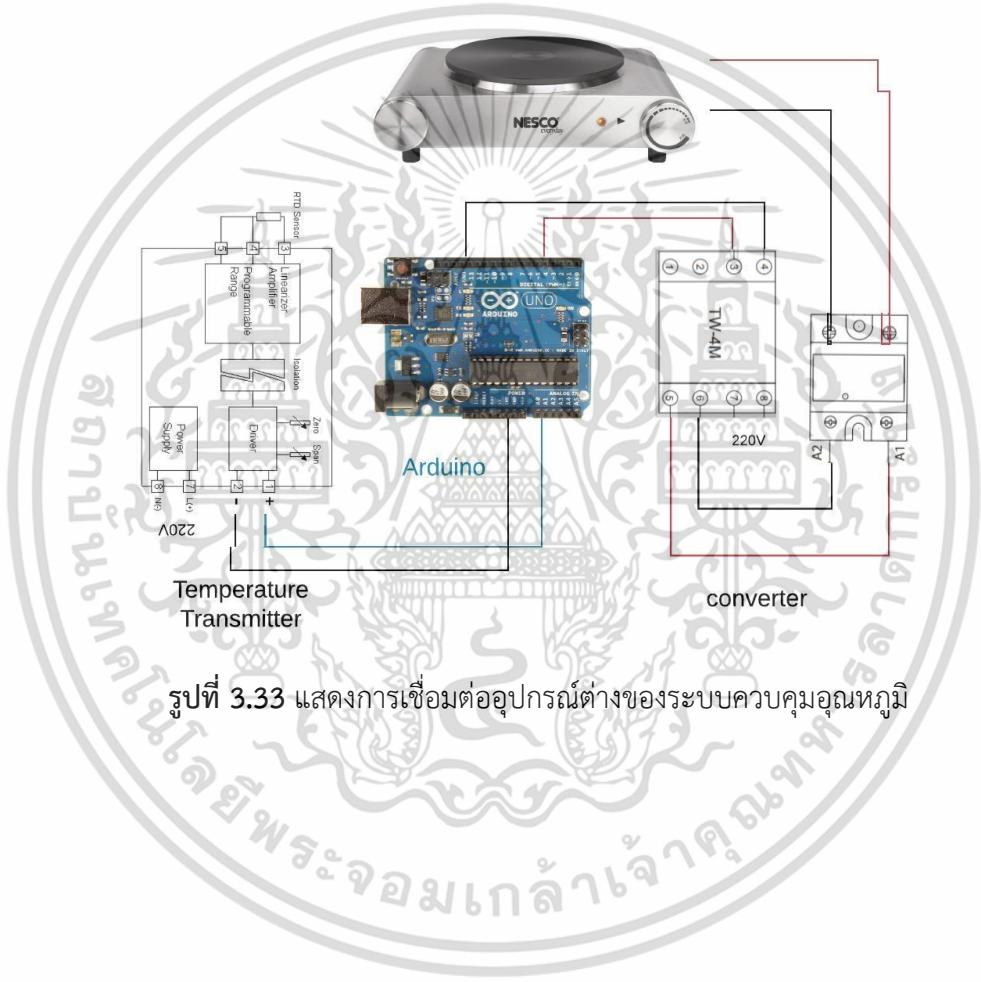
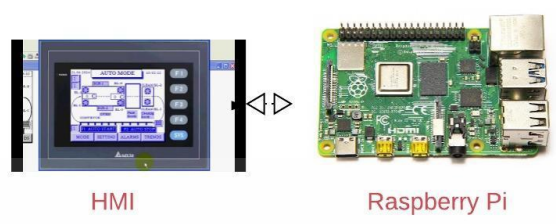
- Copy from Trace คัดลอกการตั้งค่าจาก Trace Instance ที่ได้กำหนดค่าการติดตามที่มีอยู่จากวัตถุการติดตามแล้ว สามารถคัดลอกข้อมูลการกำหนดค่าไปยัง Trace ที่สร้างขึ้น

3.5 การแสดงเส้นโค้งข้อมูลด้วย “Trend”

Trend แสดงข้อมูลที่ใช้ในฐานข้อมูลของการบันทึกแนวโน้ม ตรงกันข้ามกับองค์ประกอบติดตาม องค์ประกอบเทรนด์เหมาะสมอย่างยิ่งสำหรับการบันทึกข้อมูลระยะยาว ซึ่งจะอยู่ภายใต้ Trend Recording Manager และเปิดใช้งานการแก้ไขการกำหนดค่าการติดตาม เมื่อรันใหม่ CODESYS จะโหลดการกำหนดค่าที่มีอยู่ในองค์ประกอบของระบบรันใหม่ CmpTraceMgr สามารถกำหนดค่าแอปพลิเคชันด้วยการบันทึกแนวโน้มจำนวนเท่าใดก็ได้

การพัฒนา visualization ด้วย Trend

- สร้างอย่างน้อยหนึ่งตัวแปรใน PLC_PRG
- เพิ่ม Visualization ในโครงสร้างอุปกรณ์ด้านล่างแอปพลิเคชัน visual การสร้างภาพเปล่าปรากฏขึ้น
- เปิดกล่องเครื่องมือ Toolbox ► Special Control .
- ลากองค์ประกอบ Trend ไปที่พื้นที่ใช้งาน ไดอะล็อกการตั้งค่าจะเปิดขึ้น
- เลือก task ที่จะทำการบันทึกแนวโน้ม เช่น MainTask
- เพิ่มตัวแปรเทรนด์ด้วย Add Variable คลิกตกลงเพื่อปิดการกำหนดค่าเทรนด์
- คลิก Visualization ► Insert Elements for Controlling Trend Elements
- ตามค่าเริ่มต้นองค์ประกอบการควบคุมทั้งสามจะเปิดใช้งานในไดอะล็อก คลิกตกลงเพื่อปิดไดอะล็อก
- ตั้งค่าแอปพลิเคชันที่มี Trend เป็นใช้งานอยู่
- คอมไพล์แอปพลิเคชันด้วย F11
- คลิก ออนไลน์ ► เข้าสู่ระบบ
- เริ่มแอปพลิเคชันด้วย F5
- การสร้างภาพข้อมูลเป้าหมายปรากฏขึ้น การสร้างภาพข้อมูลประกอบด้วยแผนภาพแนวโน้มด้วยเส้นโค้งค่าของตัวแปร องค์ประกอบการควบคุมเปิดใช้งานอินพุตของผู้ใช้



รูปที่ 3.33 แสดงการเชื่อมต่ออุปกรณ์ต่างของระบบควบคุมอุณหภูมิ

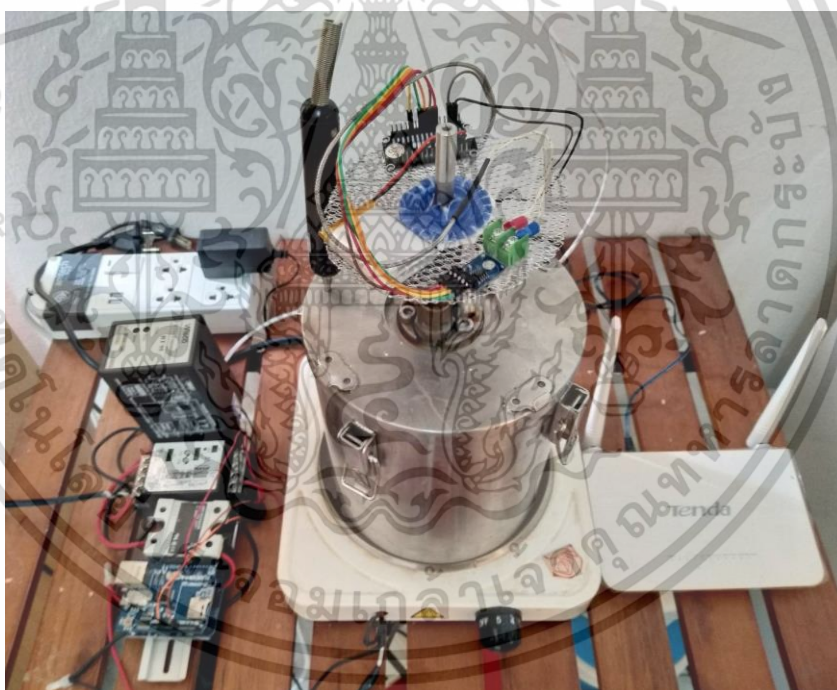
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

จำลองระบบควบคุมอุณหภูมิสำหรับถังผสมสารเคมี

บทนี้อธิบายถึงการจำลองระบบควบคุมอุณหภูมิของถังผสมสารเคมีด้วยตัวควบคุมพีไอดี เพื่อหาค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมของระบบ โดยการนำข้อมูลอินพุต คือ เปอร์เซ็นการเปิดปิดของฮีตเตอร์ และเอาต์พุต คือ อุณหภูมิของของเหลวที่เป็นตัวกลางที่ใช้กระจายความร้อนที่ใช้ในการทดลองจากการทดลองซึ่งของเหลวชนิดนี้ คือ น้ำ มาหาโมเดลทางคณิตศาสตร์ของระบบ เพื่อนำโมเดลนี้ไปใช้ในการจำลองระบบด้วยฟังก์ชัน SIMULINK ในโปรแกรม MATLAB

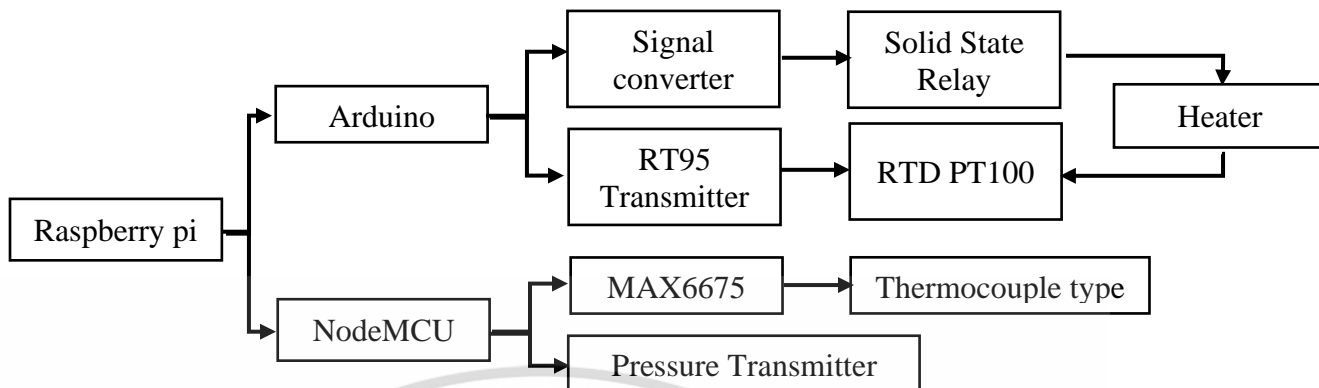
4.1 ระบบควบคุมอุณหภูมิ



รูปที่ 4.1 ระบบควบคุมอุณหภูมิสำหรับถังผสมสารเคมี

จากรูปที่ 4.1 แสดงภาพรวมของระบบสำหรับควบคุมอุณหภูมิจากองค์ประกอบฮาร์ดแวร์ในบทที่ 3 สามารถนำมาออกแบบระบบเพื่อควบคุมอุณหภูมิได้ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การออกแบบที่ใช้สำหรับควบคุมอุณหภูมิ

ระบบนี้จะแบบการทำงานออกเป็น 3 ส่วนใหญ่ๆ ได้แก่ ส่วนตัวควบคุมหลัก คือ ราบเบอร์รี่พาย จะทำการรับค่าอุณหภูมิและความดันของระบบ เพื่อใช้สำหรับควบคุมกระบวนการโดยใช้ตัวควบคุมพีไอดี ในโปรแกรมโคดีซิด แล้วส่งค่าไปยังอาดูโนเพื่อควบคุมฮีเตอร์ส่วนกลางในการรับส่งข้อมูลตัวที่ 1 คือ อาดูโนเป็นตัวกลางในการส่งค่าอุณหภูมิของของเหลวที่เป็นตัวกลางในการกระจายความร้อน ซึ่งของเหลวที่ใช้คือน้ำ และรับค่าจากตัวควบคุมเพื่อควบคุมฮีเตอร์และส่วนกลางในการรับส่งข้อมูลตัวที่ 2 คือ โหนดเอ็มซียูเป็นตัวกลางในการส่งข้อมูลความดันและอุณหภูมิภายในตัวถังสำหรับผสมสารเคมี ซึ่งระบบจะไม่ได้ควบคุมส่วนนี้โดยตรงแต่จะควบคุมความดันไม่ให้เกินค่าที่กำหนด โดยใช้ relief valve และอุณหภูมิภายในตัวถังผสมสารเคมีจะมีความใกล้เคียงกับอุณหภูมิของตัวกลางในการกระจายความร้อน

4.2 ผลของการปรับค่าพีไอดีในโคดีซิดเทียบกับแมทแลป

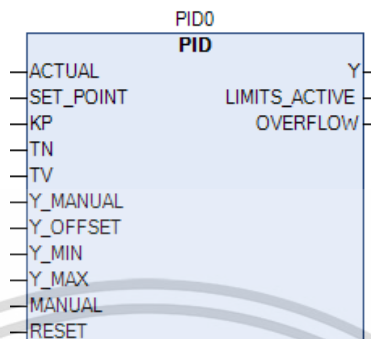
ในส่วนนี้จะทำการหาผลของพารามิเตอร์ของการควบคุมแบบ PID คือ Kp, Tn และ Tv โดยการจำลองในโปรแกรมโคดีซิดด้วยโหมด Simulation และ Simulink ในโปรแกรม MATLAB ซึ่งจะใช้ทรานเฟอร์ฟังก์ชันแบบเฟิร์สออร์เดอร์ คือ $G(s) = 1/(1s + 1)$

โครงสร้างการทำงานของตัวควบคุมพีไอดีในโคดีซิด

ไลบรารี util.library มีบล็อกฟังก์ชันตัวควบคุม PID (Proportional – Integral – Derivative) ฟังก์ชันบล็อกนี้แตกต่างจากคอนโทรลเลอร์ PD ซึ่งประกอบด้วยอินพุต TN เพิ่มเติมสำหรับการปรับแก้ไขเวลาเป็นวินาที (เช่น “ 0.5” สำหรับ 500 มิลลิวินาที)

ตัวควบคุม PID วัดเวลาที่ผ่านไประหว่างการโทรสองครั้ง อย่างไรก็ตามด้วยความแม่นยำสูงสุดเป็นมิลลิวินาที อาจนำไปสู่การทำงานที่หยابในกรณีที่มีรอบการทำงานสั้น: ตัวอย่างเช่นในกรณีของรอบ

เวลา 1 ms PID บางครั้งอาจวัด 2 มิลลิวินาทีบางครั้ง 0 มิลลิวินาที ดังนั้นหากเป็นไปได้สำหรับกรณีดังกล่าวควรใช้ PID_FIXCYCLE ซึ่งสามารถตั้งค่ารอบเวลาได้อย่างแม่นยำ



รูปที่ 4.3 ฟังก์ชันบล็อก PID ในโปรแกรม CODESYS

Y_OFFSET, Y_MIN และ Y_MAX ทำหน้าที่สำหรับการเปลี่ยนแปลงของตัวแปรที่ถูกจัดการภายในช่วงที่กำหนด สามารถใช้ MANUAL เพื่อเปลี่ยนเป็นการทำงานแบบแมนนวล; สามารถใช้ RESET เพื่อเริ่มต้นคอนโทรลเลอร์อีกครั้ง

ในการทำงานปกติ (MANUAL = RESET = LIMITS_ACTIVE = FALSE) ตัวควบคุมจะคำนวณข้อผิดพลาดของตัวควบคุม e ซึ่งแตกต่างจาก SET_POINT - ACTUAL โดยสร้างค่าที่มาตามเวลาและเก็บค่าเหล่านี้ไว้ภายใน

เอาต์พุตนั้นคือตัวแปร Y ซึ่งแตกต่างจากคอนโทรลเลอร์ PD ที่มีส่วนประกอบสำคัญเพิ่มเติมและคำนวณดังนี้:

$$Y = Kp \left[1 + \frac{1}{Tn} \int e dt + Tv \frac{de}{dt} \right] + Y_OFFSET \quad (4.1)$$

ดังนั้นนอกเหนือจาก P-part ยังมีการเปลี่ยนแปลงข้อผิดพลาดของคอนโทรลเลอร์ (D-part) ในปัจจุบันและการรวมของข้อผิดพลาดของคอนโทรลเลอร์ (I-part) ที่มีอิทธิพลต่อตัวแปรที่ถูกควบคุม คอนโทรลเลอร์พีไอดีสามารถแปลงเป็นคอนโทรลเลอร์พีไอได้อย่างง่ายดายโดยการตั้งค่า TV = 0

เนื่องจากส่วนที่สำคัญเพิ่มเติมนั้นการโอเวอร์โพล์สามารถเกิดขึ้นได้จากการกำหนดพารามิเตอร์ที่ไม่ถูกต้องของคอนโทรลเลอร์หากส่วนหนึ่งของข้อผิดพลาด e มีขนาดใหญ่มาก ดังนั้นเพื่อความปลอดภัยจึงมีการแสดงผลลิตินที่เรียกว่าโอเวอร์โพล์ ซึ่งในกรณีนี้จะมามีค่า TRUE สิ่งนี้จะเกิดขึ้นหากระบบควบคุมไม่เสถียรเนื่องจากการกำหนดพารามิเตอร์ไม่ถูกต้อง ในเวลาเดียวกันคอนโทรลเลอร์จะถูกระงับและจะเปิดใช้งานอีกครั้งโดยการเริ่มต้นใหม่เท่านั้น

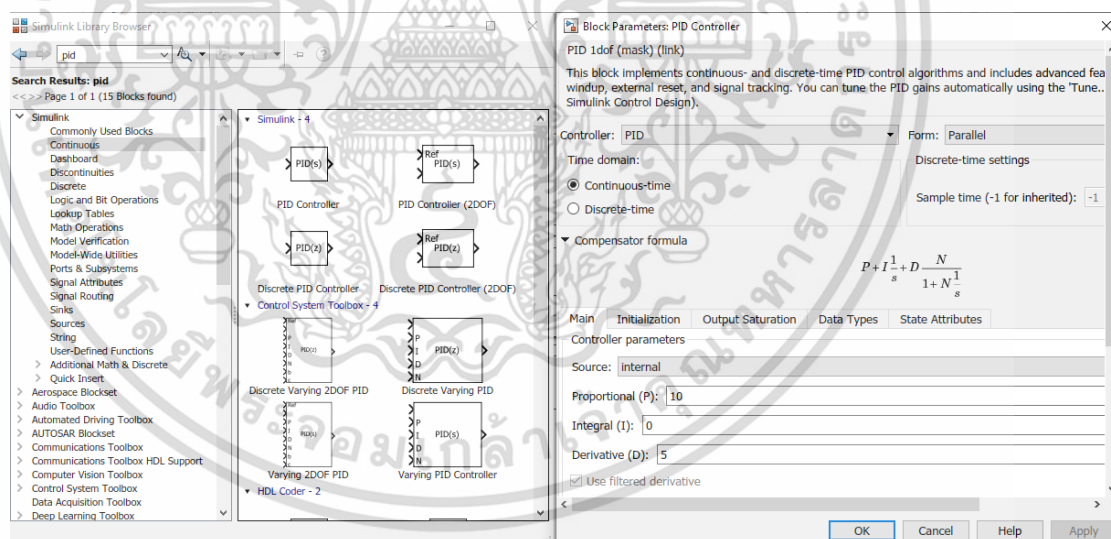
4.2.1 โครงสร้างการทำงานของตัวควบคุมพีไอดีในแมทแลป

บล็อกตัวควบคุมพีไอดีใช้ตัวควบคุม PID (PID, PI, PD, P และ I) บล็อกนั้นเหมือนกันกับบล็อกคอนโทรลเลอร์แบบไม่ต่อเนื่องที่มีพารามิเตอร์โดเมนเวลาตั้งค่าเป็นเวลาต่อเนื่อง บล็อกเอาต์พุตคือผลรวมถ่วงน้ำหนักของสัญญาณอินพุต, อินทิกรัลของสัญญาณอินพุต และอนุพันธ์ของสัญญาณอินพุต น้ำหนักคือ อัตราขยายของพารามิเตอร์ตามสัดส่วนสัดส่วนอินทิกรัล และอนุพันธ์ first-order pole ทำการกรองการกระทำของอนุพันธ์

การใช้งานของคอนโทรลเลอร์พีไอดีอย่างง่ายโดยใช้ซิมูลิงค์สามารถออกแบบได้โดยใช้สองวิธีที่กันคือ ใช้บล็อกชื่อพีไอดีในเบราวเซอร์ไลบรารี และตัวควบคุมพีไอดีที่สร้างขึ้นหรือเราสามารถออกแบบตัวควบคุมพีไอดีของเราเองโดยใช้บล็อกไดอะแกรม ผลลัพธ์ของทั้งสองวิธีอาจแตกต่างกันขึ้นอยู่กับการออกแบบตัวควบคุมพีไอดี

โครงสร้างสมการของคอนโทรลเลอร์ PID มีดังนี้

$$Y = P + \frac{I}{s} + D \frac{N}{1 + N \frac{1}{s}} \quad (4.2)$$



รูปที่ 4.4 ไลบรารี (ซ้าย) และหน้าต่างตั้งค่าของฟังก์ชันบล็อก PID (ขวา) ในโปรแกรม MATLAB

การใช้งานคอนโทรลเลอร์พีไอดีที่โปรแกรมมีมาให้ สามารถเปิดไลบรารีแล้วพิมพ์พีไอดีในช่องค้นหา จากนั้นลากบล็อกนั้นมาที่ส่วนใช้งาน และทำการดับเบิลคลิกเพื่อทำการปรับค่าต่างๆ บล็อกรองรับคอนโทรลเลอร์โครงสร้างหลายชนิด ตัวเลือกที่กำหนดค่าได้ในบล็อกประกอบด้วย:

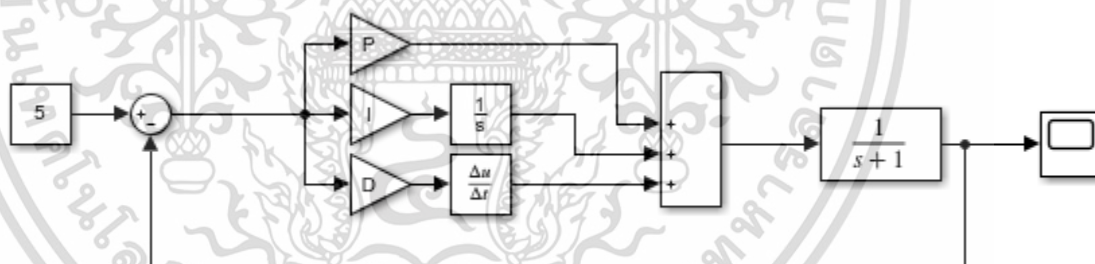
- ประเภทตัวควบคุม (PID, PI, PD, P และ I)

- รูปแบบตัวควบคุม (Parallel หรือ Ideal)
- โดเมนเวลา (ต่อเนื่องหรือไม่ต่อเนื่อง)
- เงื่อนไขเริ่มต้นและทริกเกอร์รี
- ซีต จำกัด การอิมพัลส์ของเอาต์พุตและกลไกการป้องกันการดิ่งข้อมูลในตัว
- การติดตามสัญญาณสำหรับการถ่ายโอนการควบคุมแบบไร้ขอบและการควบคุมแบบหลายปุม

ในการสร้างคอนโทรลเลอร์พีไอดีเอง ต้องออกแบบสมการคอนโทรลเลอร์พีไอดีขึ้นมา แล้วลากบล็อกที่สำคัญ ได้แก่ Gain, Integrator, Derivative, sum และ transfer function และนะบล็อกเหล่านี้ ลากเชื่อมกันตามเงื่อนไขของสมการคอนโทรลเลอร์ PID ที่ออกแบบไว้ ยกตัวอย่างเช่น สมการคอนโทรลเลอร์พีไอดีที่สร้างไว้คือ

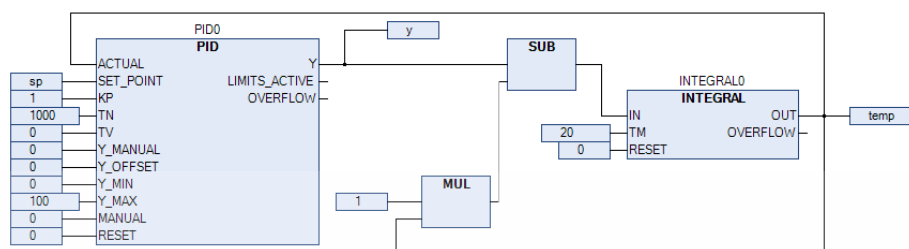
$$Y = \left[Pe + I \int e dt + D \frac{de}{dt} \right] \quad (4.3)$$

สามารถนำสมการนี้ออกแบบตัวควบคุมแบบวงรอบปิดได้คือ

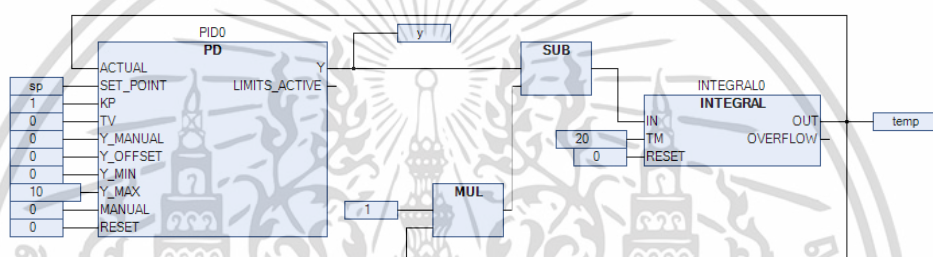


รูปที่ 4.5 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID ในโปรแกรม MATLAB

4.2.2 โปรแกรมสำหรับจำลองใน CODESYS



รูปที่ 4.6 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID ในโปรแกรม CODESYS

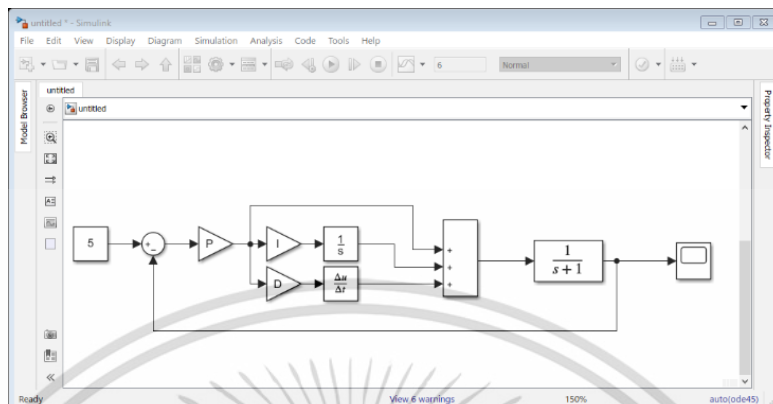


รูปที่ 4.7 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PD ในโปรแกรม CODESYS

การออกแบบโปรแกรมจำลองการควบคุมแบบ PID ของทรานเฟอร์ฟังก์ชัน $G(s) = 1/(1s + 1)$ แบ่งได้เป็นสองส่วน คือ ส่วนตัวควบคุม และทรานเฟอร์ฟังก์ชัน ในส่วนควบคุมในระบบนี้ จะใช้สองบล็อกคือ บล็อก PD ใช้สำหรับการควบคุมแบบ P และ PD และบล็อก PID ใช้สำหรับการควบคุมแบบ PI และ PID

ในส่วนของของทรานเฟอร์ฟังก์ชัน สามารถแปลงเป็นสมการดิฟเฟอเรนเชียล ได้คือ $y = u - y'$ ดังนั้นในการสร้างบล็อกจะต้องเอาเอาต์พุตที่ออกมาลบกับการ INTEGRAL ที่ได้จากการ INTEGRAL ของผลการลบก่อนหน้า และเอาต์พุตของระบบคือผลที่ได้จากการ INTEGRAL

4.2.3 โปรแกรมสำหรับจำลองใน MATLAB



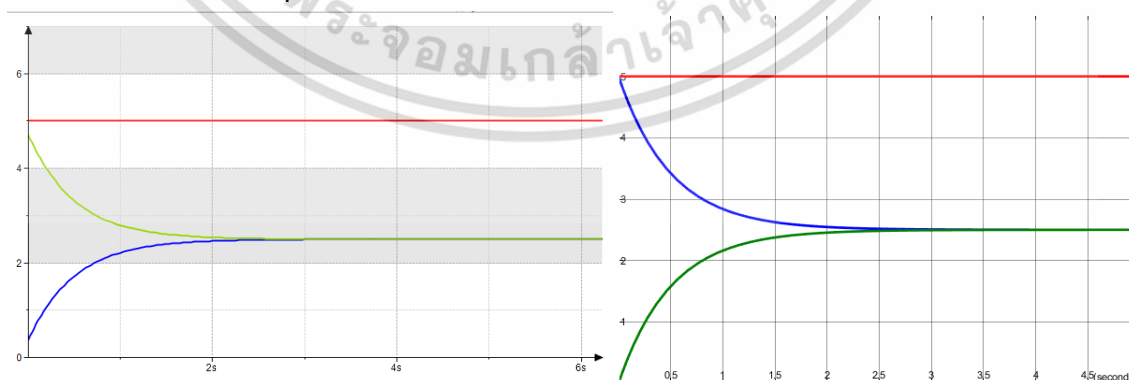
รูปที่ 4.8 โปรแกรมสำหรับการจำลองการควบคุมวงรอบปิดด้วยตัวควบคุม PID โดยใช้พารามิเตอร์ K_p , T_n และ T_v ในโปรแกรม MATLAB

การการออกแบบโปรแกรมจำลองการควบคุมแบบพีโอดีของทรานเฟอร์ฟังก์ชัน $G(s) = 1/(1s + 1)$ โดยใช้สมการคอนโทรลเลอร์พีโอดีเหมือนกับในโปรแกรม CODESYS คือ

$$Y = K_p \left[1 + \frac{1}{T_n} \int e dt + T_v \frac{de}{dt} \right] \quad (4.4)$$

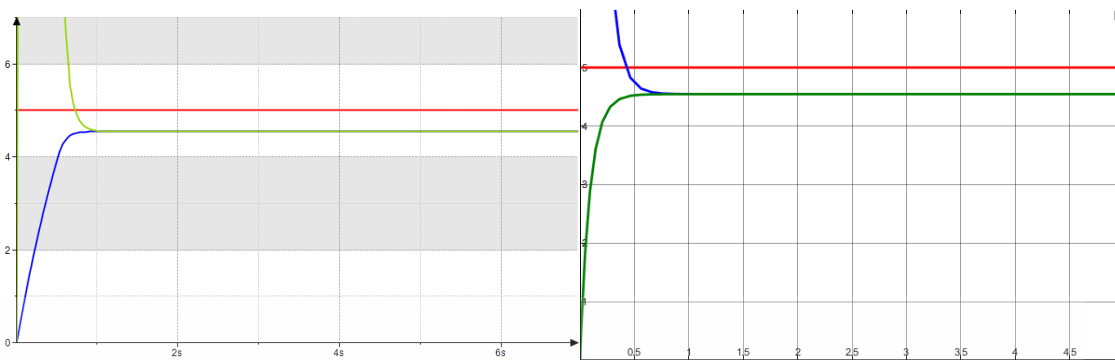
ในส่วนของตัวควบคุมพีโอดีจะนำค่า K_p คูณกับค่าเออเรอร์ แล้วนำคูณกับ $1/T_n$ แล้วทำการ Integrator และคูณกับ T_v แล้วทำการ Derivative จากนั้นนำส่วน K_p , Integrator และ Derivative มารวมกันเป็นฟังก์ชัน PID จากนั้นทำการสร้างการควบคุมแบบวงรอบปิดตั้งรูปข้างต้น

4.2.4 การควบคุม แบบ P

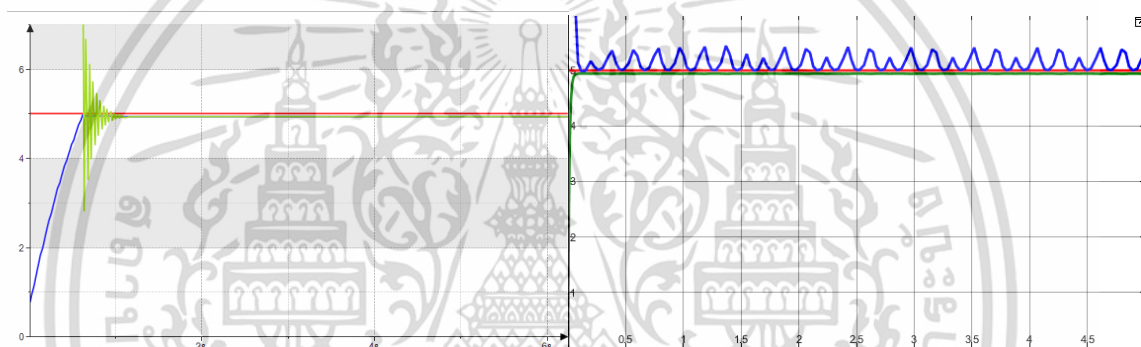


รูปที่ 4.9 รูปแสดงผลของการควบคุม P ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 1$

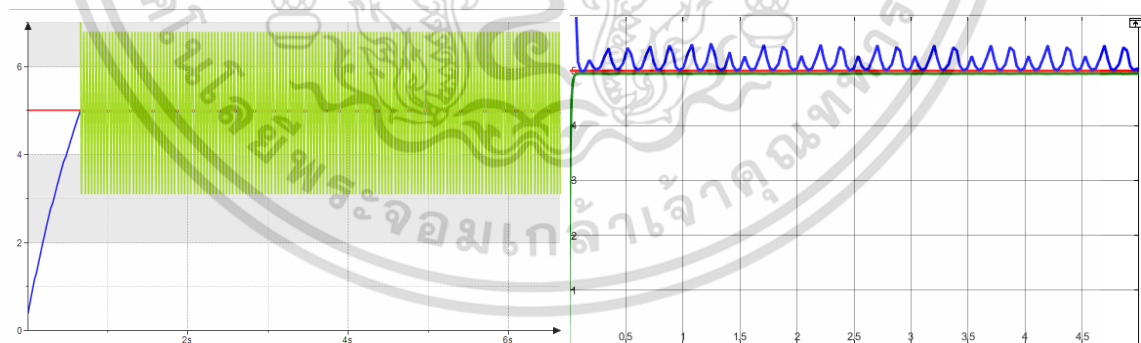
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 รูปแสดงผลของการควบคุม P ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$



รูปที่ 4.11 รูปแสดงผลของการควบคุม P ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 90$

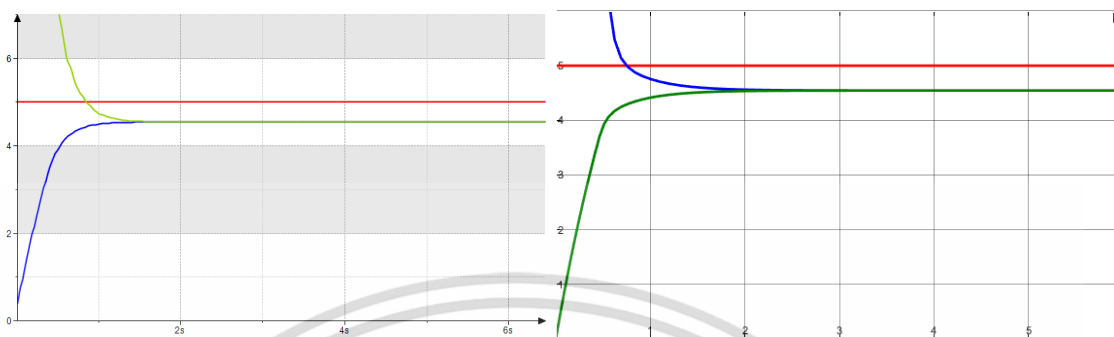


รูปที่ 4.12 รูปแสดงผลของการควบคุม P ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 99$

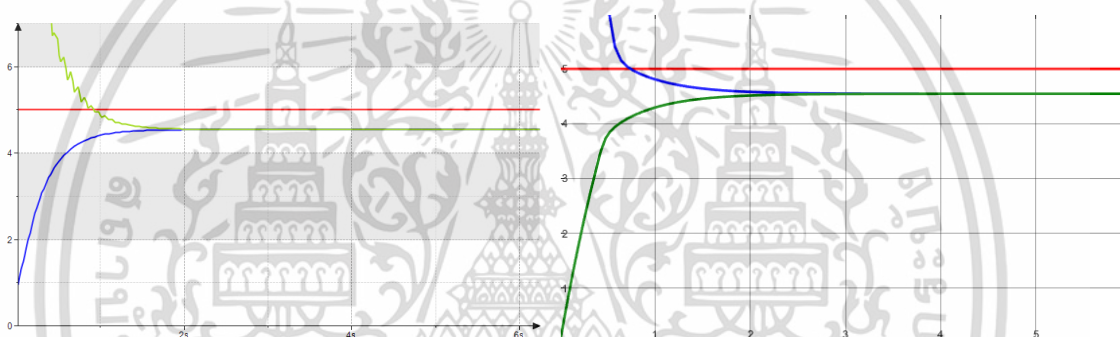
จากรูปข้างต้นแสดงผลของการควบคุมแบบ P จะเห็นได้ว่า ถ้าหากค่า P มีค่าน้อยจะทำให้ระบบเกิดการออสซิลเลชันที่มากและระบบช้า หากเพิ่มค่า P ขึ้นมาจะทำให้ระบบเร็วขึ้นและช่วยลดค่าออสซิลเลชัน และหากเพิ่มค่า P มากเกินไปจะทำให้ระบบเกิดการแกว่ง และไม่เสถียรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

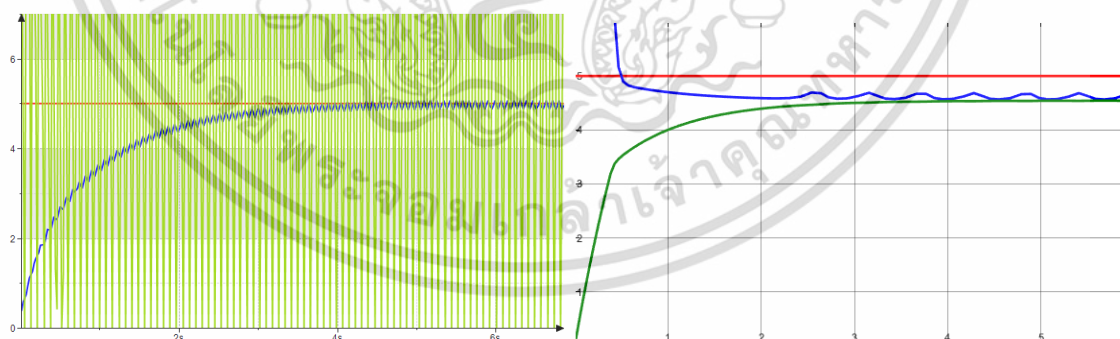
4.2.5 การควบคุม แบบ PD



รูปที่ 4.13 รูปแสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_v = 0.1$



รูปที่ 4.14 รูปแสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_v = 0.2$

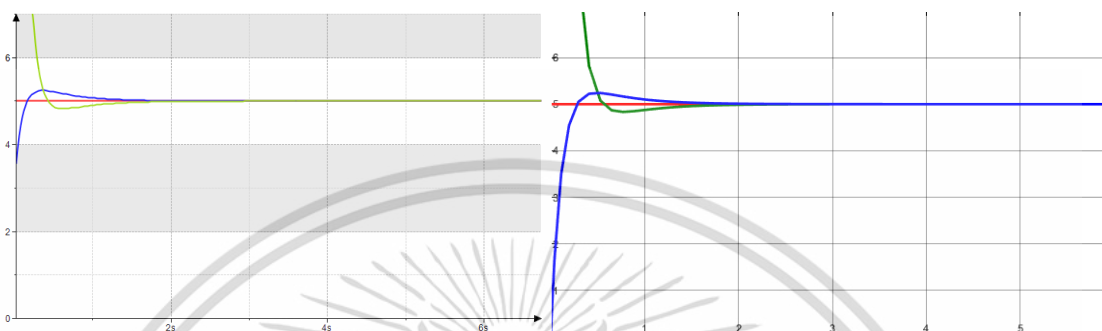


รูปที่ 4.15 รูปแสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_v = 0.5$

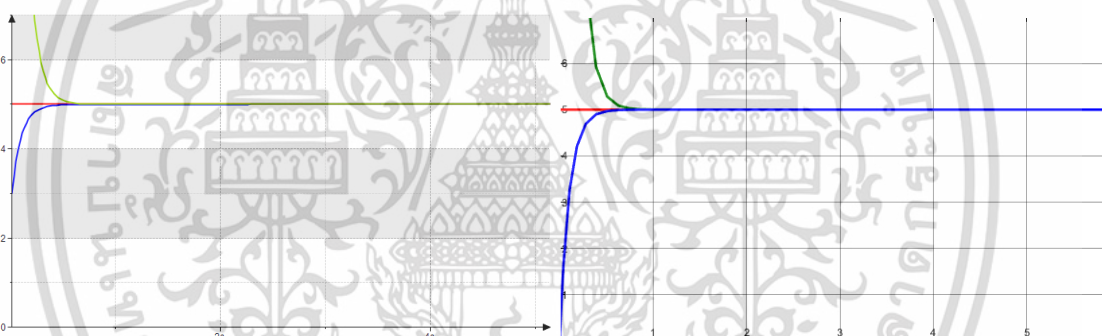
ในการควบคุมแบบ PD การควบคุมแบบ PI ในเทอมของ D จะใช้ค่าพารามิเตอร์ T_v ซึ่งมีค่าเท่ากับ K_d/K_p การเพิ่มค่า T_v จนถึงจุดจุดหนึ่งจะไม่มีมีการเปลี่ยนแปลง แต่เมื่อถึงจุดหนึ่งการเพิ่มค่า T_v จะทำให้ระบบช้าลง ในการเพิ่มค่า T_v มากขึ้นอีกจะทำให้ระบบเกิดการแกว่งและไม่เสถียร สำหรับ

โปรแกรม CODESYS หากเพิ่มมากเกินไปนั่นคือพารามิเตอร์ไม่เหมาะสมจะทำให้เกิด OVERFLOW ขึ้นให้
ทำค่าแกว่งระหว่างจุดต่ำสุดและสูงสุดของเอาต์พุต

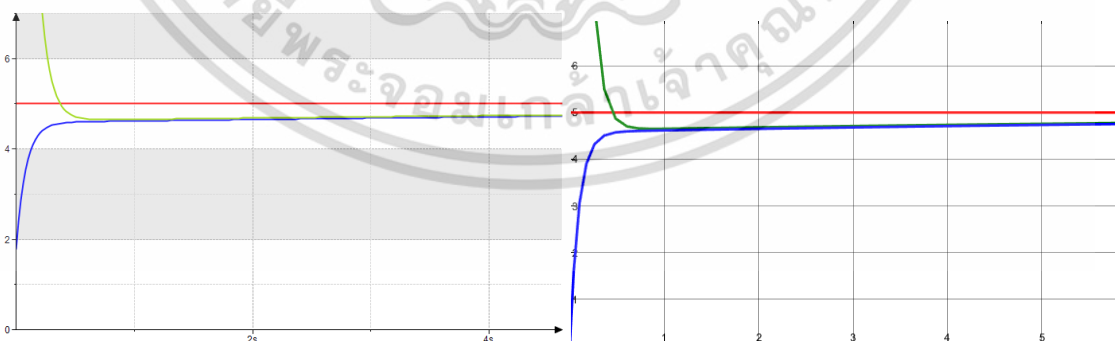
4.2.6 การควบคุม แบบ PI



รูปที่ 4.16 รูปแสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี
setpoint = 5, $K_p = 10$, $T_n = 0.5$



รูปที่ 4.17 แสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี
setpoint = 5, $K_p = 10$, $T_n = 1$

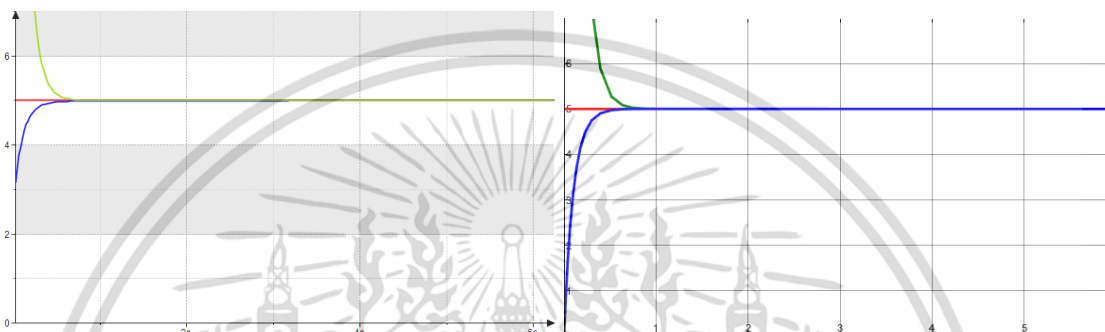


รูปที่ 4.18 แสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5,
 $K_p = 10$, $T_n = 10$

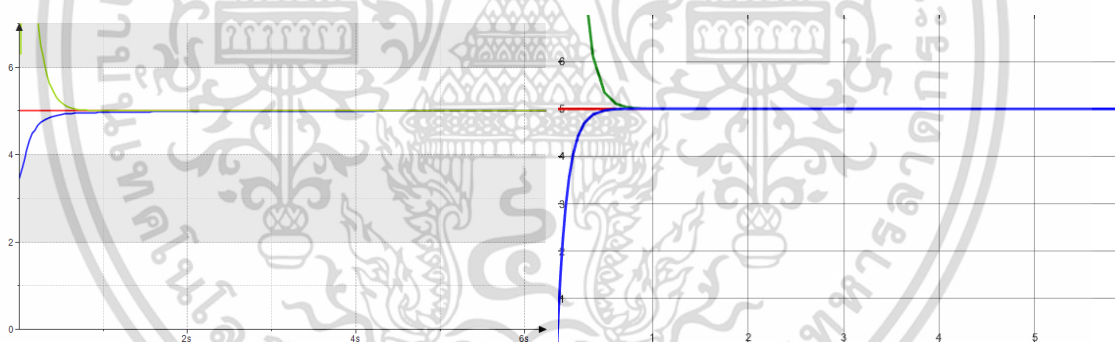
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบ PI ในเทอมของ I จะใช้ค่าพารามิเตอร์ T_n ซึ่งมีค่าเท่ากับ K_p/K_i ในการใช้ตัวควบคุมนี้ไม่สามารถปรับค่า T_n ให้เท่ากับ 0 ได้และจะต้องปรับค่า T_n ให้สูง เนื่องจากจะทำให้ในเทอมของ I มีค่าเข้าใกล้ 0 การลดค่า T_n ลงจะทำให้ระบบมีค่าออฟเซตที่ลดลงตาม และหากลดค่า T_n มากเกินไปจะทำให้เกิดโอเวอร์ชูต

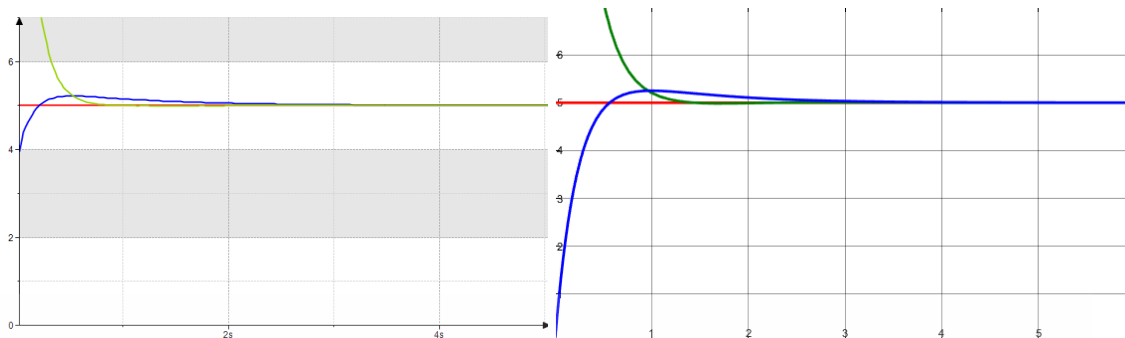
4.2.7 การควบคุม แบบ PID



รูปที่ 4.19 แสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_n = 1$, $T_v = 0.005$



รูปที่ 4.20 แสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_n = 1$, $T_v = 0.02$



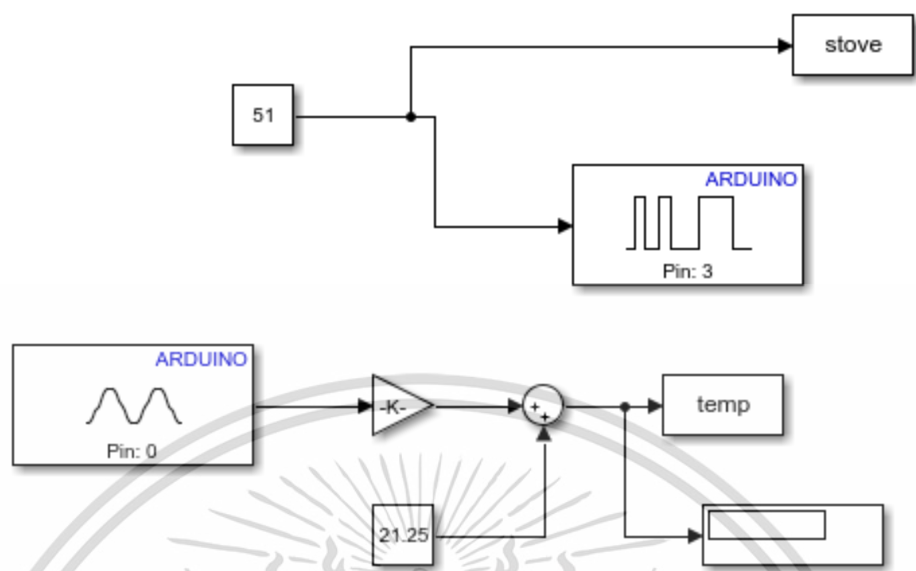
รูปที่ 4.21 แสดงผลของการควบคุม PD ของโปรแกรม CODESYS (ซ้าย) และ MATLAB (ขวา) เมื่อมี setpoint = 5, $K_p = 10$, $T_n = 1$, $T_v = 0.03$

การควบคุมแบบพีไอดีจะใช้พารามิเตอร์จากการควบคุมแบบ PI มาใช้ใน และทำการปรับค่าในเทอม D การเพิ่มค่า T_v จนถึงจุดจุดหนึ่งจะไม่มีมีการเปลี่ยนแปลง แต่เมื่อถึงจุดหนึ่งการเพิ่มค่า T_v จะทำให้ระบบช้าลง ในการเพิ่มค่า T_v มากขึ้นอีกจะทำให้ระบบเกิดโอเวอร์ชูดขึ้น

การควบคุมระบบโดยใช้ตัวควบคุมพีไอดีของฟังก์ชันเฟิร์สอเดอร์การปรับค่าในเทอม P ทำให้เร็วขึ้นและช่วยลดค่าออฟเซตลง และหากเพิ่มค่า P มากเกินไปจะทำให้ระบบเกิดการแกว่ง และไม่เสถียรได้ การเพิ่มค่าในเทอม I หรือลดค่า T_n จะทำให้ระบบลดการออฟเซตลง และการปรับค่าในเทอม D ไม่เหมาะกับระบบเฟิร์สอเดอร์ เนื่องจากทำให้ระบบช้าลงแล้วยังให้เกิดการโอเวอร์ชูดและระบบเกิดการไม่เสถียร

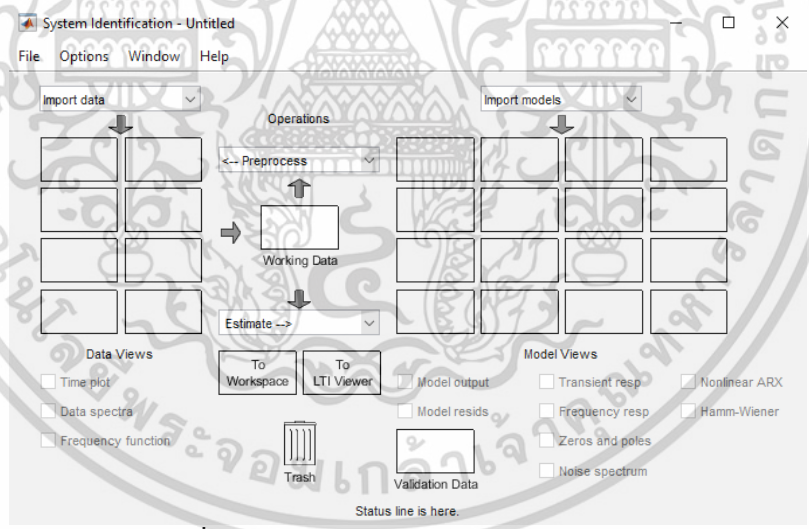
4.3 การหาโมเดลทางคณิตศาสตร์

การทดลองหาธรรมชาติของระบบโดยการป้อนอินพุตโดยเปิดฮีตเตอร์ 20 เเปอร์เซ็นต์ เนื่องจากการเปิดฮีตเตอร์ 20 เเปอร์เซ็นต์ไม่ทำให้น้ำถึงจุดเดือดทำให้หาโมเดลทางคณิตศาสตร์ได้ง่าย และเก็บค่าอุณหภูมิของของเหลวที่เป็นตัวกระจายความร้อนเป็นเวลา 3 ชั่วโมง หรือ 10800 วินาที แล้วนำข้อมูลมาใส่ในฟังก์ชัน system identification ในโปรแกรม MATLAB ดังรูปที่ 4.3



รูปที่ 4.22 โปรแกรมสำหรับเก็บข้อมูลอุณหภูมิเมื่อเปิด heater 20%

4.3.1 วิธี system identification

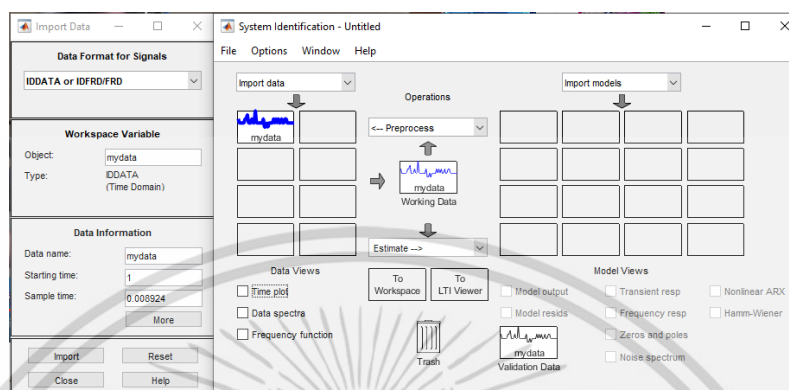


รูปที่ 4.23 System Identification toolbox

System Identification toolbox ในโปรแกรม MATLAB เป็นเครื่องมือที่ใช้ในการทำนายเอกลักษณ์ หรือโมเดลทางคณิตศาสตร์ของระบบทั้งในโดเมนเวลาและโดเมนความถี่ โดยใช้ข้อมูลความสัมพันธ์ของอินพุตและเอาต์พุต ซึ่งมีวิธีหาสมการได้หลากหลายรูปแบบ เช่น transfer function models, state space models, process models polynomial models เป็นต้น และสามารถนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดลที่ได้จากการประมาณการมาเทียบกับข้อมูลจริงเพื่อหาความแตกต่างและเลือกโมเดลที่เหมาะสมกับข้อมูลมากที่สุด



รูปที่ 4.24 การเพิ่มข้อมูลลงใน System Identification toolbox

ในการหาโมเดลทางคณิตศาสตร์ของระบบจะเริ่มจากการนำเอาข้อมูลที่ได้จากการทดลอง แล้วตั้งชื่อข้อมูล เวลาเริ่มต้น และเวลาของตัวอย่าง ขึ้นต่อมาเป็นการเลือกรูปแบบโมเดลที่จะสร้างที่ได้กล่าวไว้ในข้างต้น ในการทดลองนี้จะทำการหาโมเดลทางคณิตศาสตร์ด้วยวิธี process model เนื่องจากวิธีนี้สามารถให้โปรแกรมประมาณค่าพารามิเตอร์และยังสามารถปรับค่าให้ใกล้เคียงกับข้อมูลมากขึ้นได้ด้วยตัวเอง จะเห็นได้ว่าข้อมูลที่บันทึกมาเป็นกราฟที่มีลักษณะเป็นแบบ first order และมีดีเลย์ ซึ่งมีรูปแบบสมการ

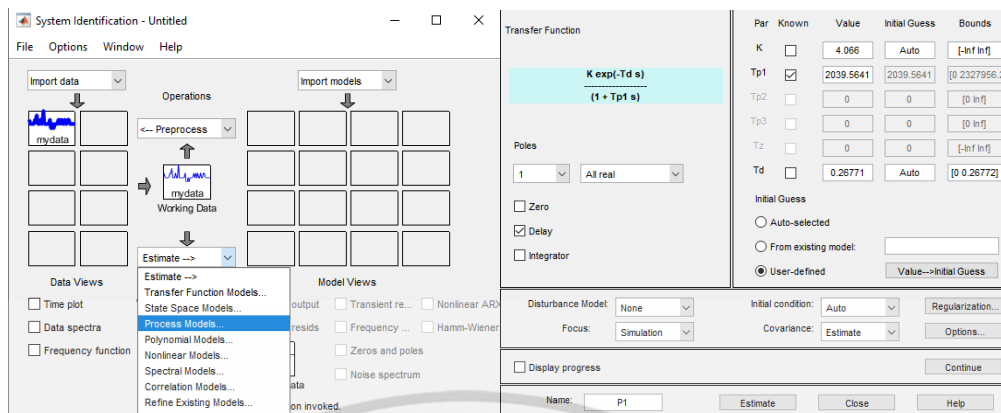
$$G = \frac{K \exp(-T_d s)}{\tau_p s + 1} \quad (4.6)$$

เมื่อ K คือ ค่าอัตราขยาย

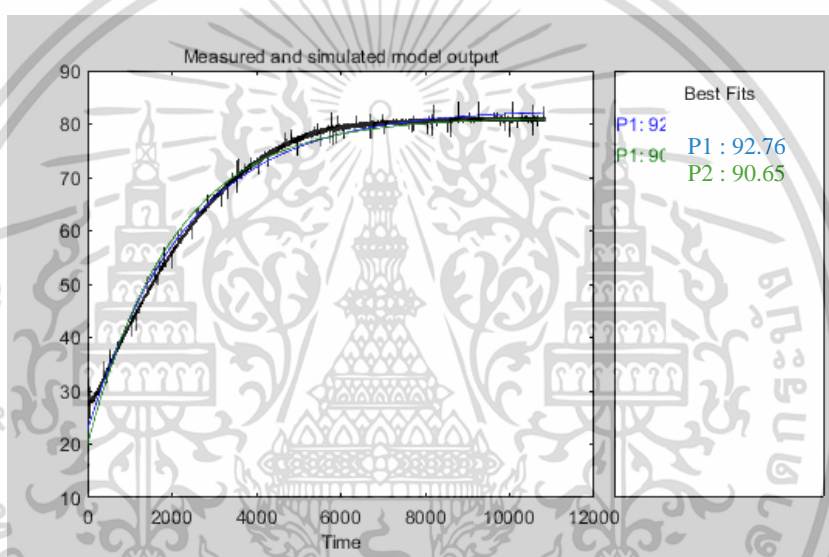
T_d คือ ค่าดีเลย์ของระบบ

τ_p คือ time constant หรือเวลาที่ทำให้ระบบมีค่า 63% ของแอมพลิจูด

เมื่อทำการปรับแต่งรูปแบบโมเดล และประมาณการหาโมเดลทางคณิตศาสตร์ของระบบแล้วนำกราฟโมเดลทางคณิตศาสตร์ของระบบมาเปรียบเทียบกับข้อมูลจริง ถ้ายังไม่พึงพอใจโมเดลทางคณิตศาสตร์ของระบบที่ทำการประมาณการได้นั้น สามารถทำการประมาณการใหม่โดยแก้ไขพารามิเตอร์ตามต้องการและนำกราฟโมเดลทางคณิตศาสตร์ใหม่มาเปรียบเทียบกับของเดิมและข้อมูลจริงจนกว่าจะหาโมเดลทางคณิตศาสตร์ที่เหมาะสมกับระบบได้



รูปที่ 4.25 หน้าต่าง Process model



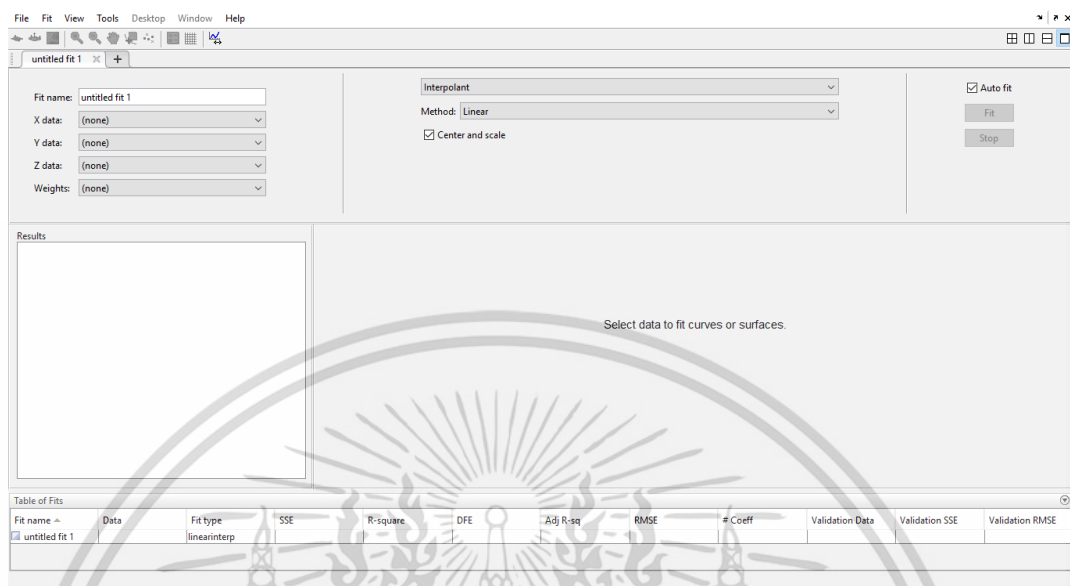
รูปที่ 4.26 แสดงการเปรียบเทียบระหว่างกราฟที่ได้จากการทดลองและทรานเฟอร์ฟังก์ชันจากฟังก์ชัน system identification เมื่อเปิด heater 20%

ในการประมาณโมเดลทางคณิตศาสตร์ของระบบในครั้งแรก P1 ทำการประมาณโดยอัตโนมัติได้ ค่า $K_p = 4.1382$, $T_{p1} = 2339.6$ และ $T_d = 0.26772$ ซึ่งมีความใกล้เคียงกับข้อมูล 92.76% ดังเส้นกราฟสีน้ำเงิน จะเห็นได้ว่าในส่วปลายมีแนวโน้มที่ลู่ออกจากข้อมูลค่อนข้างมาก ดังนั้นจึงได้ทำการปรับค่าตัวแปรในโมเดลใหม่เป็น $K_p = 4.066$, $T_{p1} = 2039.6$ และ $T_d = 0.26771$ ซึ่งมีความใกล้เคียงกับข้อมูล 90.65% ดังเส้นกราฟสีเขียว ถึงแม้ว่าจะมีความแตกต่างจากข้อมูลมากกว่า แต่กราฟมีการลู่เข้าสู่ค่าที่ใกล้เคียงกับข้อมูลมากกว่า ดังนั้นจึงเลือกโมเดลนี้ซึ่งมีรูปแบบสมการคือ

$$G(s) = \frac{4.066 \exp(-0.26771 s)}{2039.6s + 1} \quad (4.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 วิธี Curve fitting



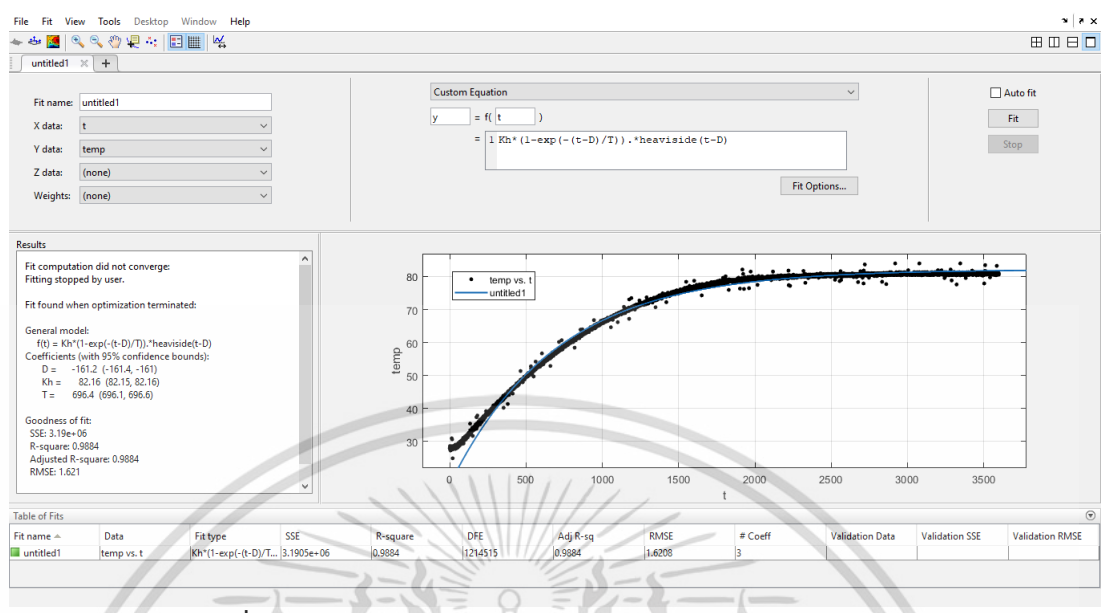
รูปที่ 4.27 หน้าต่าง Curve Fitting Toolbox

Curve Fitting Toolbox ในโปรแกรม MATLAB สามารถใช้ได้ทั้งสองแบบ คือ ผ่านยูสเซอร์กรรฟิกรอินเทอร์เฟส (GUI) หรือผ่านทางบรรทัดคำสั่ง MATLAB โดยกล่องเครื่องมือนี้จะใช้ในการหาสมการหรือโมเดลทางคณิตศาสตร์ของระบบ ซึ่งสามารถกำหนดเองหรือใช้สมการพื้นฐานที่กล่องเครื่องมือมีให้ก็ได้

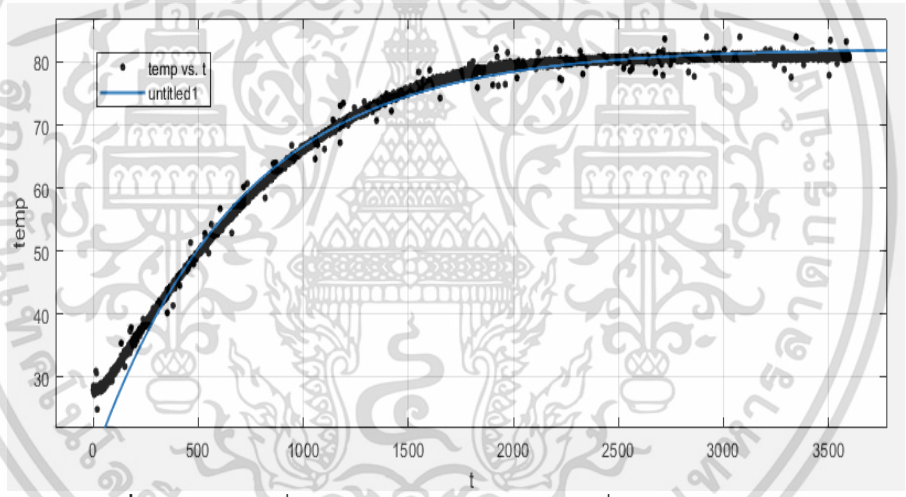
ระบบนี้เป็นรูปแบบเฟิร์สอเดอร์สมการที่ใช้ คือ

$$Y(x) = Kh * (1 - \exp(-(x - D)/T)) * \text{heaviside}(x - D) \quad (4.8)$$

ในการใช้งานทำการตั้งค่าเวลาที่แกน X และอุณหภูมิที่แกน Y แล้วเลือกสมการที่ต้องการเป็นแบบ Custom Equation และทำการใส่สมการข้างต้นลงไปและกด FIT เพื่อหาตัวแปรของสมการ



รูปที่ 4.28 การใส่ตัวแปรและสมการใน Curve Fitting Toolbox



รูปที่ 4.29 กราฟที่ได้จากการ Curve Fitting เมื่อเปิด heater 20%

จากรูปที่ 4.29 เป็นผลลัพธ์ที่ได้จาก Curve Fitting ของข้อมูลได้ค่าตัวแปรต่างๆ คือ $D = -161.2$, $Kh = 82.16$ และ $T = 696.4$ มีความใกล้เคียงกับข้อมูล 95% ดังนั้นสมการของระบบนี้ที่มีอินพุตของระบบเท่ากับ 20% คือ

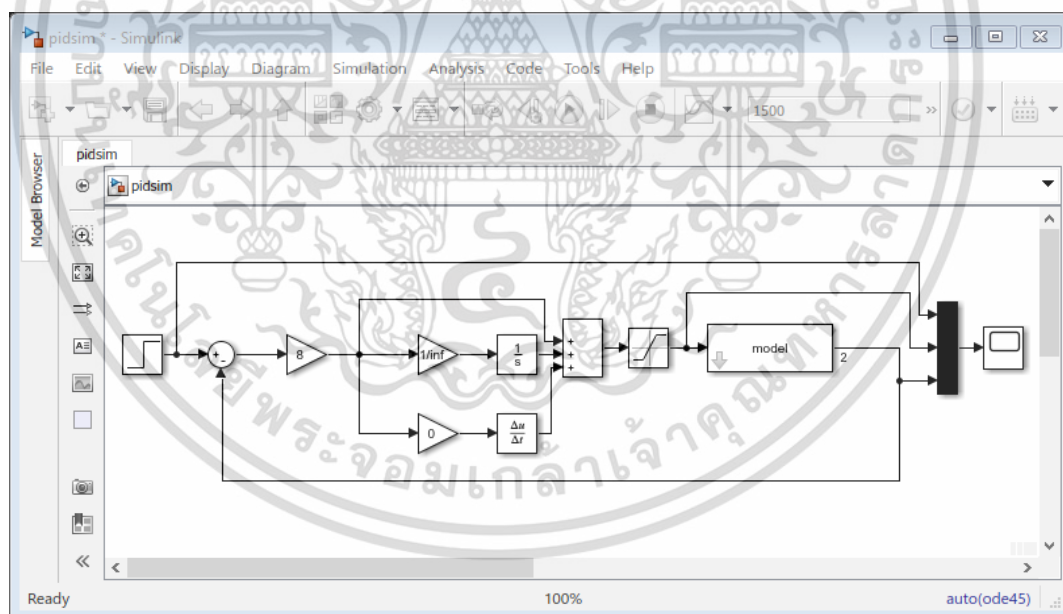
$$Y(x) = 82.16 * (1 - \exp(-(x + 161.2)/696.4)) \cdot \text{heaviside}(x + 161.2) \quad (4.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการที่ได้จากการหาโมเดลทางคณิตศาสตร์ของระบบด้วย Curve Fitting toolbox นั้นมีความแตกต่างจาก Identification toolbox จากหัวข้อข้างต้น คือ วิธีนี้สามารถหาโมเดลทางคณิตศาสตร์ได้เฉพาะอินพุตเดียว ไม่สามารถใช้ได้กับอินพุตอื่น ถ้าหากต้องการประมาณผลลัพธ์ที่ได้จากอินพุตอื่นต้องเก็บข้อมูล และหาโมเดลทางคณิตศาสตร์ของระบบใหม่

4.4 การจำลองการควบคุมระบบโดยใช้ตัวควบคุมพีไอดี

จากสมการที่ 4.7 ซึ่งเป็นสมการที่เหมาะสมสำหรับระบบนี้จากจากหัวข้อ 4.3 ได้นำมาออกแบบการจำลองระบบด้วยตัวควบคุมพีไอดีด้วย close loop control ตามแผนภาพบล็อก (block diagram) บนซิมูลิงค์ของโปรแกรม MATLAB ดังรูปที่ 4.26 โดยกำหนดค่าเป้าหมาย (setpoint) ในการควบคุมอยู่ที่ 70 80 และ 90 องศาเซลเซียส และค่าเปอร์เซ็นต์การพุ่งเกิน (percent overshoot) น้อยกว่า 1% และเวลาเข้าสู่สมดุลน้อยกว่า 20 นาที โดยตั้งพารามิเตอร์ Tn ไว้ที่อินฟินิตี้ แล้วทำการปรับค่าตัวพารามิเตอร์ควบคุม P ถ้าหากยังมีการออฟเซตให้ปรับลดค่า Tn ลง และปรับ Tv ให้เหมาะสมกับระบบ

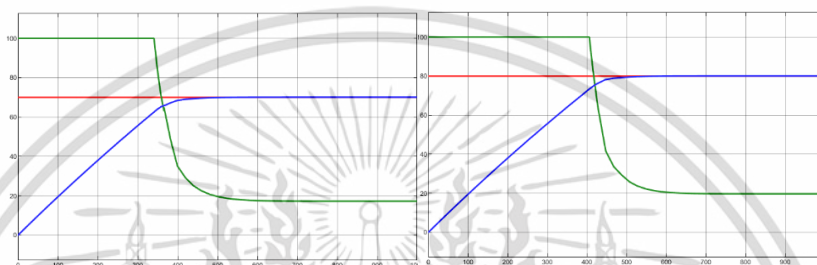


รูปที่ 4.30 โปรแกรม Simulink สำหรับจำลองระบบควบคุมพีไอดีด้วย close loop control สำหรับควบคุมอุณหภูมิของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าพารามิเตอร์ของตัวควบคุม PID ที่ได้จากการจำลองในโปรแกรม MATLAB

setpoint	Kp	Tn	Tv	%overshoot
70	11	8000	0	0
80	12	10000	0	0
90	13	12000	0	0



(ก)

(ข)

(ค)

รูปที่ 4.31 ผลของการปรับพีไอดีตามตาราง ก.เมื่ออุณหภูมิเป้าหมายคือ 70 องศาเซลเซียส ข.เมื่ออุณหภูมิเป้าหมายคือ 80 องศาเซลเซียส ค.เมื่ออุณหภูมิเป้าหมายคือ 90 องศาเซลเซียส

จากรูปที่ 4.27 และตารางที่ 4.1 เป็นการหาค่าพีไอดีที่สามารถทำให้โมเดลทางคณิตศาสตร์ของระบบเข้าใกล้อุณหภูมิเป้าหมายเร็วที่สุด และมีค่าเปอร์เซ็นต์การพุ่งเกินเท่ากับ 0 ผลที่ได้ คือ อุณหภูมิเป้าหมายมีค่า 70 เซลเซียส ค่า $K_p = 11$ และ $T_n = 8000$ อุณหภูมิเป้าหมายมีค่า 80 เซลเซียส ค่า $K_p = 12$ และ $T_n = 10000$ และอุณหภูมิเป้าหมายมีค่า 90 เซลเซียส ค่า $K_p = 13$ และ $T_n = 13000$ และยังคงอยู่ในเงื่อนไขค่าเปอร์เซ็นต์การพุ่งเกิน (percent overshoot) น้อยกว่า 1% และเวลาเข้าสู่สมดุลน้อยกว่า 20 นาที (1200 วินาที)

4.5 ทดสอบการสื่อสารแบบมอดบัสของระบบ

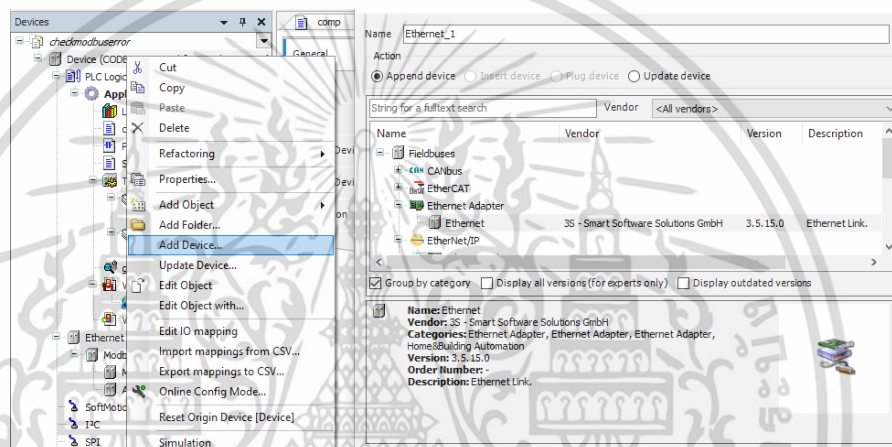
ในส่วนนี้เป็นการทดสอบความสามารถในการรับส่งข้อมูลระหว่าง Master และ Slave ผ่านการสื่อสารแบบมอดบัสโดย Master คือ Raspberry pi และ Slave คืออาตุโนและโหนดเอ็มซียูสำหรับอา

ดูโน้จะทำกรรับและส่งข้อมูลกับรสบอร์รี่พาย ส่วนหนดเอ็มซียูจะทำกรส่งข้อมูลให้รสบอร์รี่พายเพียงอย่างเดย ในกรส่งข้อมูลจะทำกรส่งข้อมูลวรอบละ 100 มลลวินาที เป็นเวลา 10 นาที

4.5.1 มอดบ์สนโปรแกรมโคตซีต

ตั้งค่า สำหรับการมอดบ์สโดยใช้โคตซีตเป็น Master

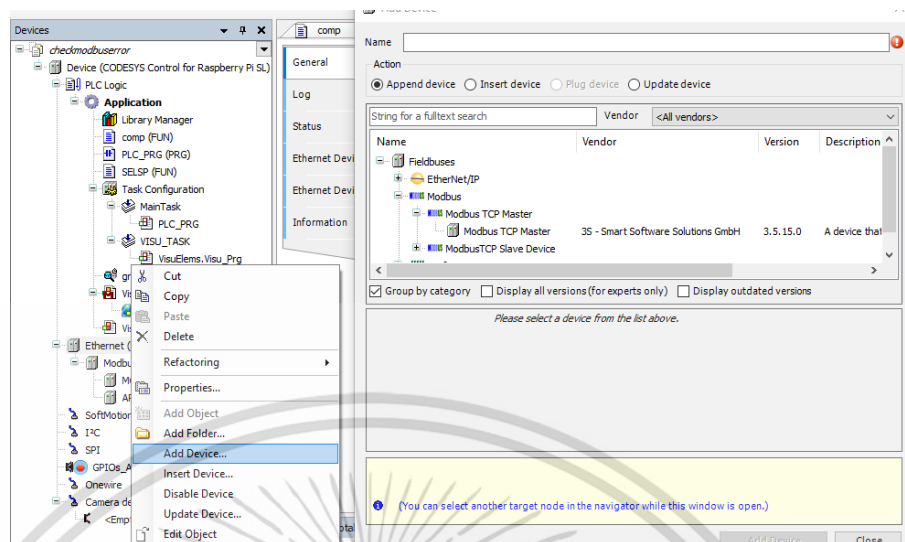
1. เพิ่ม device ethernet โดยคลิกขวาที่ Application ➤ Add Device ➤ Ethernet Adepter ➤ Ethernet



รูปที่ 4.32 การ device ethernet ของโปรแกรม CODESYS

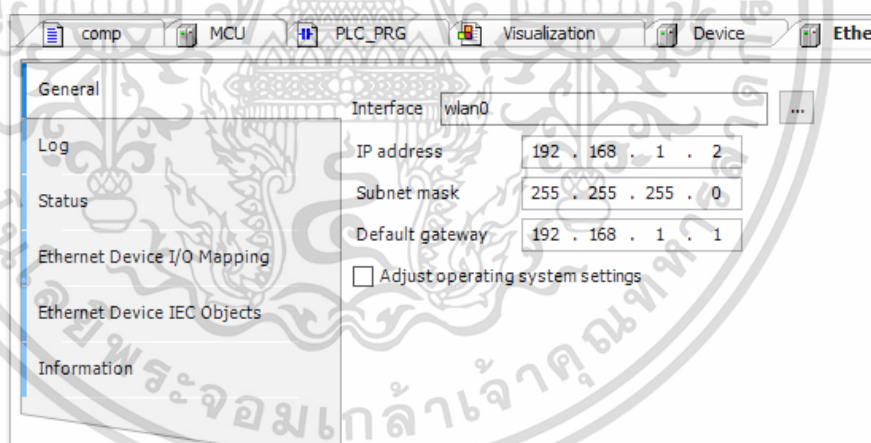
2. เพิ่ม device master-modbus และ slave-modbus
โดยคลิกขวาที่ device Ethernet ➤ Add Device ➤ Modbus TCP Master
โดยคลิกขวาที่ device Ethernet ➤ Add Device ➤ Modbus TCP Slave แล้วตั้งชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อกรศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านกรค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้



รูปที่ 4.33 Add Modbus TCP Master-Slave ของโปรแกรม CODESYS

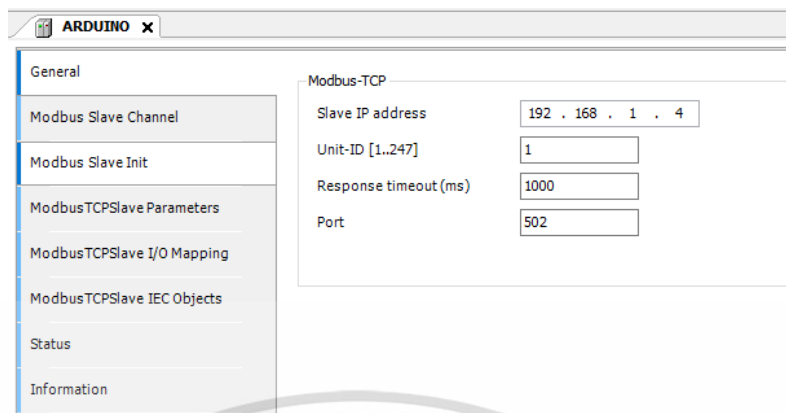
3. เปิดหน้าต่าง Ethernet ขึ้นมา ทำการตั้งค่า IP Address, Subnet mask และ gateway ของ raspberry pi (Master)



รูปที่ 4.34 IP Address, Subnet mask และ gateway ของ raspberry pi (Master) ในโปรแกรม CODESYS

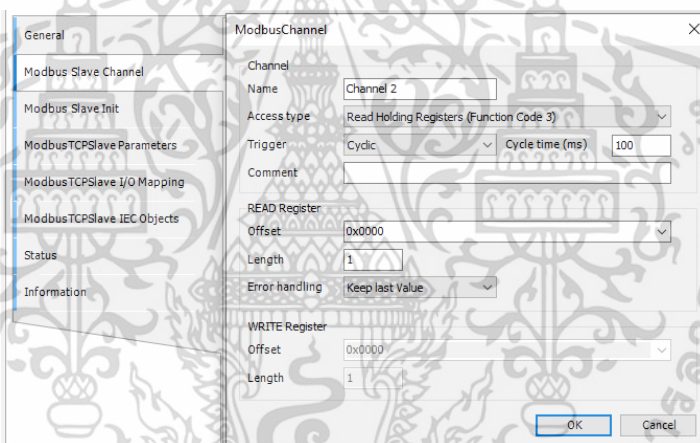
4. เปิดหน้าต่าง slave ที่ใช้เชื่อมต่อกับ Arduino ขึ้นมา ทำการตั้งค่า IP Address, unit-ID และ Port ของ Arduino (slave)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



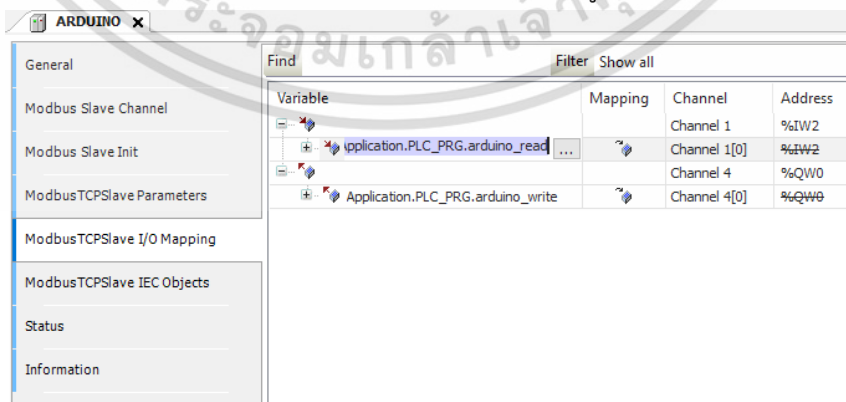
รูปที่ 4.35 IP Address, unit-ID และ Port ของ Arduino(Slave) ในโปรแกรม CODESYS

5. เพิ่ม channel ในการติดต่อสื่อสาร โดยจะตั้งวิธีการในการติดต่อสื่อสาร (access type) และ offset ในตรงกับตัว Arduino เพื่อให้สามารถสื่อสารกันได้



รูปที่ 4.36 เพิ่ม channel ในการติดต่อสื่อสารกับ Arduino

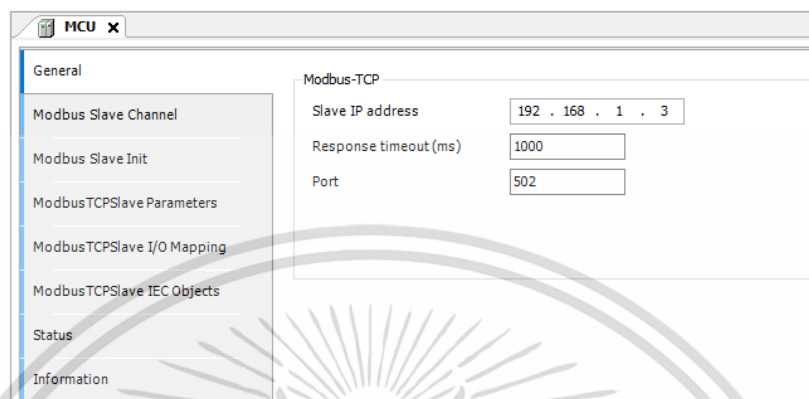
6. กำหนดตัวแปรสำหรับแต่ละ channel ที่ใช้สำหรับรับและส่งข้อมูล



รูปที่ 4.37 กำหนดตัวแปรที่ใช้สำหรับรับและส่งข้อมูลกับ Arduino

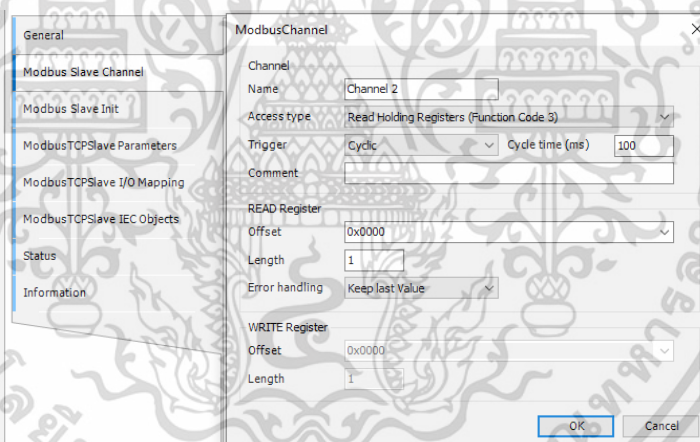
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เปิดหน้าต่าง slave ที่ใช้เชื่อมต่อกับ NodeMCU ขึ้นมา ทำการตั้งค่า IP Address, unit-ID และ Port ของ NodeMCU (slave)



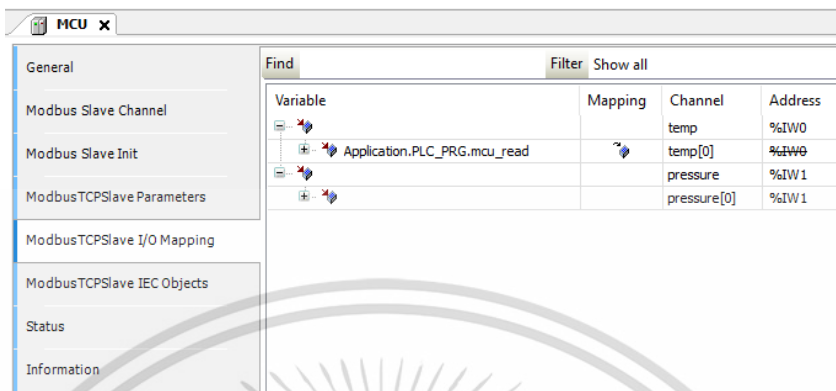
รูปที่ 4.38 IP Address, unit-ID และ Port ของ NodeMCU (Slave) ในโปรแกรม CODESYS

8. เพิ่ม channel ในการติดต่อสื่อสาร โดยจะตั้งวิธีในการติดต่อสื่อสาร (access type) และ offset ในตรงกับตัว NodeMCU เพื่อให้สามารถสื่อสารกันได้



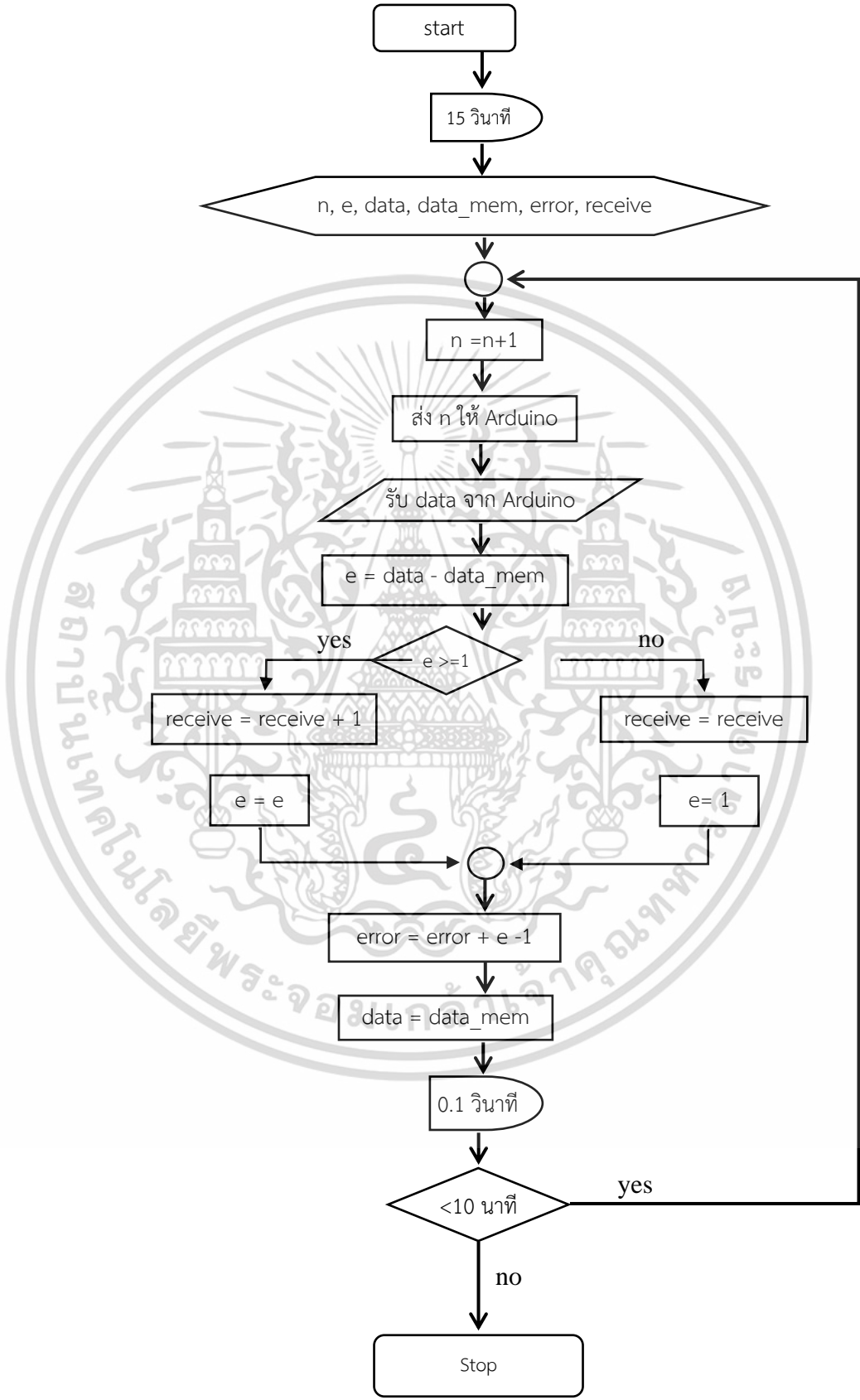
รูปที่ 4.39 เพิ่ม channel ในการติดต่อสื่อสารกับ NodeMCU

9. กำหนดตัวแปรสำหรับแต่ละ channel ที่ใช้สำหรับรับและส่งข้อมูล



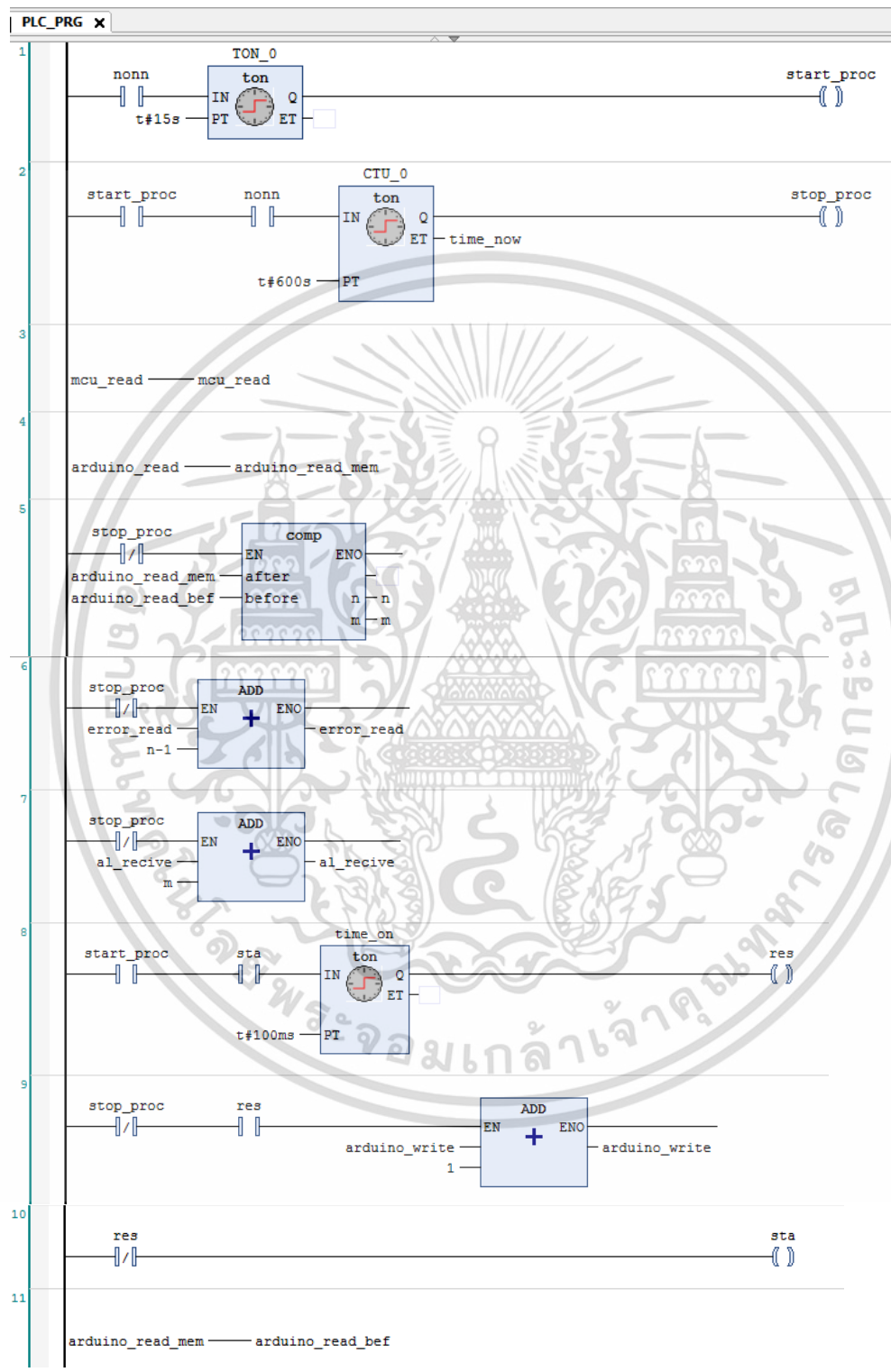
รูปที่ 4.40 กำหนดตัวแปรที่ใช้สำหรับรับและส่งข้อมูลกับ NodeMCU

MODBUS flowchart บนโปรแกรม CODESYS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับทดสอบการติดต่อสื่อสารข้อมูลแบบ MODBUS ใน CODESYS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมบล็อกที่ 1 จะถูกตั้งไว้เพื่อหน่วงเวลาเริ่ม 10 วินาที เพื่อเผื่อเวลาสำหรับการเชื่อมต่อ อุปกรณ์บล็อกที่ 2 จะใช้เป็นตัวตั้งเวลาให้หยุดการทำงาน โดยตั้งเวลาไว้ 10 นาที บล็อกที่ 3 และ 4 เป็นบล็อกสำหรับเก็บค่าที่รับมาจากโหนดเอ็มซียูและโหนดเอ็มซียูบล็อกที่ 5 คือเงื่อนไข โดย n คือ ผลต่างของข้อมูลที่ได้รับมากับข้อมูลก่อนหน้าโดยที่ค่าจะไม่มีทางต่ำกว่า 1 ส่วน m คือ มีค่าผลต่างของข้อมูลที่ได้รับมากับข้อมูลก่อนหน้า บล็อกที่ 6 นำค่า n ไปหา error ในการรับข้อมูล error จะมีค่าเท่ากับ error +n-1 นั่นคือ error จะเพิ่มขึ้นตามผลต่างของข้อมูลที่ได้รับมากับข้อมูลก่อนหน้ามีค่ามากกว่า 1 บล็อกที่ 7 ใช้หาจำนวนข้อมูลที่ได้รับมาได้ เมื่อ m มากกว่าหรือเท่ากับ 1 แสดงว่าได้รับข้อมูลแล้ว บล็อกที่ 8, 9 และ 10 ใช้สร้างข้อมูลที่เพิ่มขึ้นทีละ 1 ทุกๆ 100 มิลลิวินาทีและเก็บข้อมูลไปในตัวแปรที่ใช้ส่งข้อมูลผ่าน MODBUS และบล็อกที่ 11 ใช้ในการเก็บข้อมูลที่ได้รับมาเพื่อใช้ในรอบต่อไป

4.5.2 MODBUS on Arduino

MODBUS TCP/IP LIBRARY สำหรับ ARDUINO

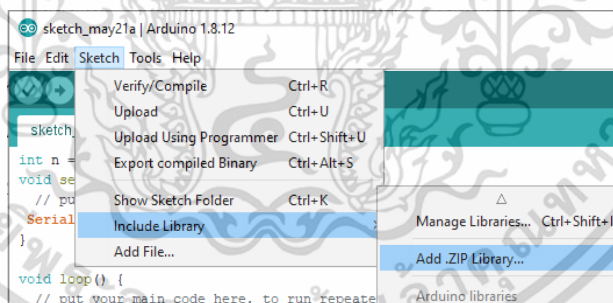
การติดตั้ง

- ดาวน์โหลด Modbus TCP/IP libraries จากเว็บไซต์

<https://github.com/luizcantoni/mudbus>

- เปิดโปรแกรม ArduinoIDE และคลิก Sketch ➤ include libraries ➤ Add.ZIP libraries

แล้วเลือกไฟล์ที่ดาวน์โหลดมา



รูปที่ 4.41 การ include libraries

การใช้งาน

ค่าเริ่มต้นโปรโตคอลการสื่อสารไลบรารี Modbus นี้จะถูกตั้งค่าพอร์ตไว้ที่ 502 (มาตรฐาน Modbus) สามารถเปลี่ยนพอร์ตนี้เพื่อเปลี่ยนค่าของ MB_PORT ในไฟล์ Mudbus.h

ไลบรารี Mudbus สามารถใช้ฟังก์ชันมาตรฐานของ Modbus ต่อไป

- Read Coils
- 3 Read Registers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5 Write Coil, 6 Write Register
- 15 Write multiple coils
- 16 Write multiple registers

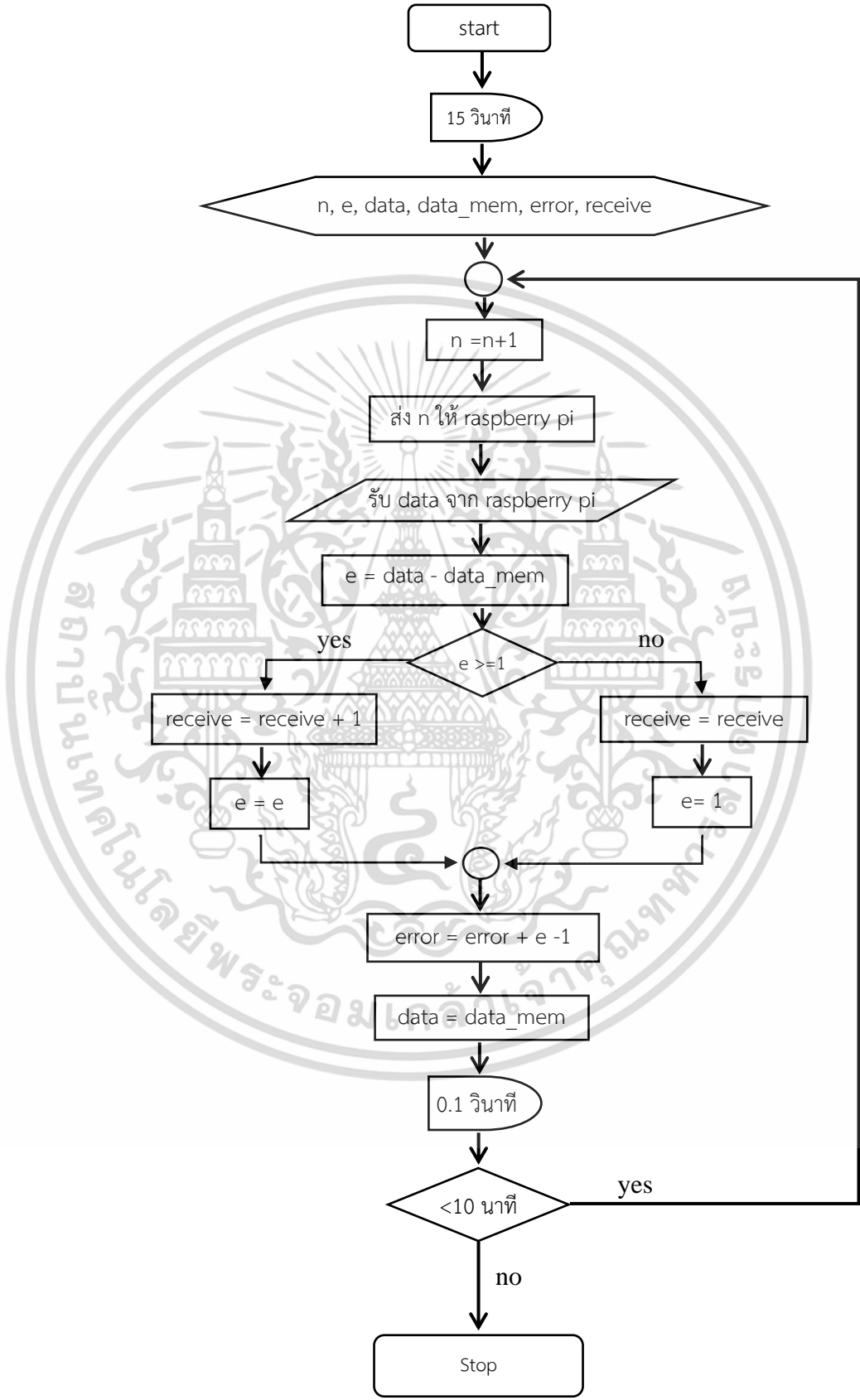
มีข้อมูลสองประเภทที่สามารถทำการส่งได้ คือ อนุาล็อก (Registers)และคิจิตอล (Coil) ค่าอนุาล็อกเป็นจำนวนเต็มและค่าคิจิตอลเป็นบูลีน ดังนั้นโนโปรโตคอล Modbus นี้คุณสามารถใช้ฟังก์ชันหรือการประกาศนี้:

- Mb.Run (): เป็นฟังก์ชันที่ใช้ในการอัปเดตสัญญาณ Modbus
- Mb.R [0 ถึง 125]: ใช้เพื่ออ่านหรือเขียนค่าจำนวนเต็มบนโปรโตคอล Modbus
- Mb.C [0 ถึง 128]: ใช้เพื่ออ่านหรือเขียนค่าบูลีนโปรโตคอล Modbus



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODBUS flowchart บนโปรแกรม Arduino



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับทดสอบการติดต่อสื่อสารข้อมูลแบบ MODBUS ใน Arduino

```

#include <SPI.h>
#include <Ethernet.h>
#include "Mudbus.h"
#define SENSOR_PIN A0
word temp,read0= 0,readmem = 0,write0= 0,writemem = 0,n,error,al_receive,m;
word con;
Mudbus Mb;
//Function codes 1(read coils), 3(read registers), 5(write coil), 6(write register)
//signed int Mb.R[0 to 125] and bool Mb.C[0 to 128] MB_N_R MB_N_C
//Port 502 (defined in Mudbus.h) MB_PORT

void setup()
{
  uint8_t mac[]      = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[]       = { 192, 168, 1, 4 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[]  = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet);
  //Avoid pins 4,10,11,12,13 when using ethernet shield

  delay(5000);
  Serial.begin(9600);
  pinMode(7, INPUT);
  pinMode(8, OUTPUT);
  digitalWrite(13,HIGH);
}
void loop()
{
  Mb.Run();
  read0 = Mb.R[3]; // รับค่าจาก raspberry pi
  n = read0-readmem;
  m = read0-readmem;
  if(n<1) n = 1;
  error = error+n-1;
  if(m>=1) al_receive = al_receive+1;
  if(read0>=1) write0 = write0+1;
  Mb.R[1] = write0; //ส่งค่าให้ raspberry pi
  Serial.print("read = "); //แสดงผล
  Serial.print(" : ");
  Serial.print(read0);
  Serial.print(" Amount receive = ");
  Serial.print(" : ");
  Serial.print(al_receive);
  Serial.print(" error receive = ");
  Serial.print(" : ");
  Serial.println(error);
  readmem = read0;
  delay(100);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้จะทำการเพิ่มไลบรารีสำหรับการเชื่อมต่ออินเทอร์เน็ตด้วยอินเทอร์เน็ตชิลด์บนอาดูโนและไลบรารีสำหรับ MODBUS และประกาศตัวแปรที่ใช้งาน จากนั้นทำการเซต IP Address, mac IP, gateway และ subnet ใน void setup

ในส่วนของวอยด์ลูปจะใช้คำสั่ง Mb.Run เพื่อทำเป็นสัญญาณในการอัปเดตการ MODBUS แล้วทำการรับค่าที่ส่งมาจากراسเบอร์รี่พายที่ตั้งไว้คือ offset = 3 จากนั้นทำการสร้างเงื่อนไข โดย n คือผลต่างของข้อมูลที่รับมากับข้อมูลก่อนหน้าโดยที่ค่าจะไม่มีทางต่ำกว่า 1 เมื่อนำค่า n ไปหา error ในการรับข้อมูล error จะมีค่าเท่ากับ error +n-1 นั่นคือ error จะเพิ่มขึ้นตามผลต่างของข้อมูลที่รับมากับข้อมูลก่อนหน้ามีค่ามากกว่า 1 ส่วน m คือ มีค่าผลต่างของข้อมูลที่รับมากับข้อมูลก่อนหน้าใช้หาจำนวนข้อมูลที่รับมาได้ เมื่อ m มากกว่าหรือเท่ากับ 1 แสดงว่าได้รับข้อมูลแล้ว ในการส่งข้อมูลแต่ละรอบจะถูกหน่วงเวลาไว้ 100 มิลลิวินาที

```

18:04:33.760 -> read = : 1188 Amount receive = : 527 error receive = : 661
18:04:33.863 -> read = : 1188 Amount receive = : 527 error receive = : 661
18:04:33.966 -> read = : 1188 Amount receive = : 527 error receive = : 661
18:04:34.069 -> read = : 1192 Amount receive = : 528 error receive = : 664
18:04:34.174 -> read = : 1192 Amount receive = : 528 error receive = : 664
18:04:34.279 -> read = : 1193 Amount receive = : 529 error receive = : 664
18:04:34.384 -> read = : 1193 Amount receive = : 529 error receive = : 664
18:04:34.488 -> read = : 1195 Amount receive = : 530 error receive = : 665
18:04:34.593 -> read = : 1195 Amount receive = : 530 error receive = : 665
18:04:34.697 -> read = : 1197 Amount receive = : 531 error receive = : 666
18:04:34.801 -> read = : 1197 Amount receive = : 531 error receive = : 666
18:04:34.906 -> read = : 1199 Amount receive = : 532 error receive = : 667
18:04:35.010 -> read = : 1199 Amount receive = : 532 error receive = : 667
18:04:35.115 -> read = : 1201 Amount receive = : 533 error receive = : 668
  
```

รูปที่ 4. 42 หน้าแสดงผลในโปรแกรม Arduino IDE

ตารางที่ 4.2 ผลที่ได้จากการทดสอบการสื่อสารระหว่าง Master (Raspberry pi) และ Slave (Arduino) เป็นเวลา 10 นาที โดยทั้ง Raspberry pi และ Arduino มีหน้าที่ทำการรับและส่งทั้งคู่

ครั้งที่	Master (Raspberry pi)			Slave (Arduino)		
	ที่รับได้	error	รวมทั้งหมด	ที่รับได้	error	รวมทั้งหมด
1	2885	2955	5840	2749	2806	5555
2	2679	3161	5840	2623	2932	5555
3	2714	3126	5840	2710	2845	5555
4	2862	2978	5840	2864	2691	5555
5	2622	3218	5840	2625	2930	5555
6	2526	3314	5840	2503	3052	5555
7	2507	3333	5840	2425	3130	5555
8	2446	3394	5840	2451	3104	5555
9	2500	3340	5840	2505	3050	5555
10	2626	3214	5840	2629	2926	5555
เฉลี่ย	2636.7	3203.3		2608.4	2946.6	
เปอร์เซ็นต์	45.14	54.85		46.95	53.04	

จากตารางที่ 4.2 จะเห็นได้ว่าราสเบอร์รี่พายมีความสามารถในการรับข้อมูลจากอาดัวโน่เฉลี่ยอยู่ที่ 45.14% จากการรับค่าทั้งหมด 5840 ค่า นั่นคือ ใน 10 นาที Arduino สามารถส่งข้อมูลได้ 5840 ครั้ง และราสเบอร์รี่พาย สามารถรับค่าได้ประมาณ 2636 ครั้ง หรือ 4.4 ครั้งต่อวินาทีและอาดัวโน่มีความสามารถในการรับข้อมูลจากราสเบอร์รี่พาย เฉลี่ยอยู่ที่ 46.95% จากการรับค่าทั้งหมด 5555 ค่า นั่นคือ ใน 10 นาทีราสเบอร์รี่พาย สามารถส่งข้อมูลได้ 5555 ครั้ง และอาดัวโน่สามารถรับค่าได้ประมาณ 2608 ครั้ง หรือ 4.5 ครั้งต่อวินาที

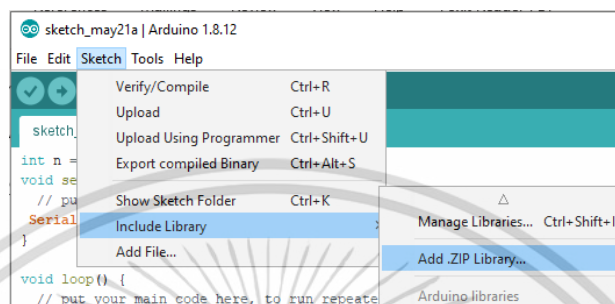
4.5.3 MODBUS on NodeMCU

มอดบัสไอพีไลบารีสำหรับโหนดเอ็มซียู

การติดตั้ง

- ดาวน์โหลด ModbusRTU and ModbusIP Master-Slave Library for ESP8266/ESP32 v3.0 จากเว็บไซต์ <https://github.com/emelianov/modbus-esp8266>

- เปิดโปรแกรม ArduinoIDE และคลิก Sketch ➤ include libraries ➤ Add.ZIP libraries แล้วเลือกไฟล์ที่ดาวน์โหลดมา



รูปที่ 4. 43 การ include libraries

การใช้งาน

ไลบรารีนี้รองรับอุปกรณ์ ESP8266/ESP32 ให้สื่อสารผ่านโปรโตคอล Modbus Modbus เป็นโปรโตคอลต้นแบบที่ใช้ในระบบอัตโนมัติทางอุตสาหกรรมและสามารถใช้ในพื้นที่อื่นๆ เช่นระบบอัตโนมัติที่บ้าน

Modbus โดยทั่วไปมีรูปแบบการเชื่อมต่อโดยใช้อุปกรณ์ RS-232 หรือ RS-485 เรียกว่า Modbus Serial และ TCP/IP ผ่าน Ethernet หรือ WiFi เรียกว่า Modbus IP

ในเวอร์ชันปัจจุบันไลบรารีอนุญาตให้ ESP8266/ESP32 ทำงานเป็นมาสเตอร์และ/หรือสเลฟรองรับ Modbus IP ผ่านเครือข่ายไร้สาย และ Modbus RTU ผ่านพอร์ตอนุกรม

คุณสมบัติ

แพลตฟอร์มที่รองรับ

- ESP8266
- ESP32

ดำเนินการเป็น

- Slave
- master

สนับสนุน

- Modbus IP (TCP)
- Modbus RTU (RS-485)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับฟังก์ชัน Modbus0x01 - Read Coils

- 0x02 - Read Input Status (Read Discrete Inputs)
- 0x03 - Read Holding Registers
- 0x04 - Read Input Registers
- 0x05 - Write Single Coil
- 0x06 - Write Single Register
- 0x0F - Write Multiple Coils
- 0x10 - Write Multiple Registers

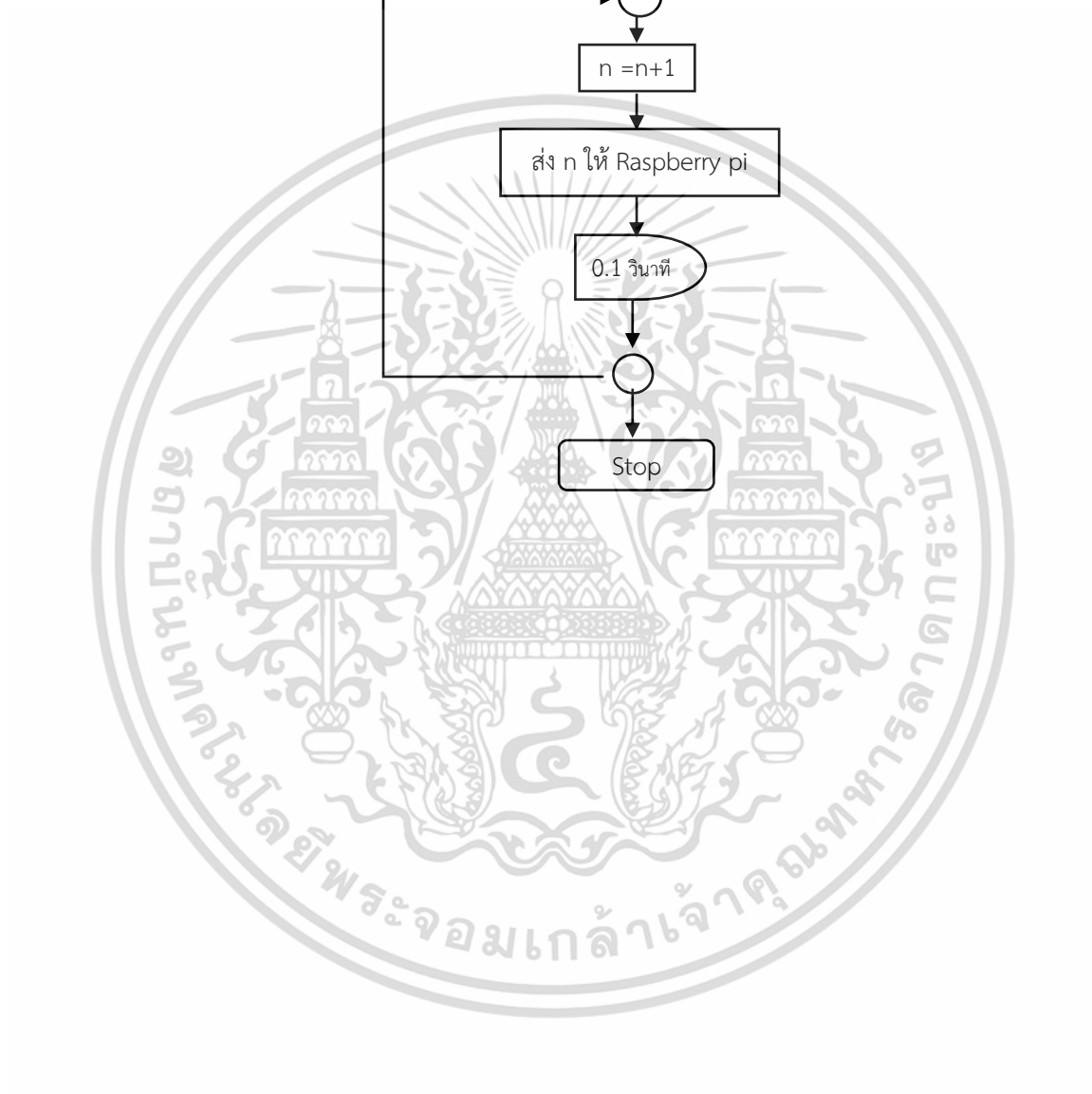
การตอบกลับ

- Master connect (ModbusIP)
- Master/Slave disconnect (ModbusIP)
- Read specific Register
- Write specific Register
- Slave transaction finish

ตารางที่ 4.3 แสดงวิธีการส่งการส่งและคำสั่ง Modbus

Register	Use as	Access	Library methods
Coil	Digital Output	Read/Write	addCoil(), Coil()
Holding Register	Analog Output	Read/Write	addHreg(), Hreg()
Input Status	Digital Input	Read Only	addIsts(), Ists()
Input Register	Analog Input	Read Only	addIreg(), Ireg()

MODBUS flowchart บนโปรแกรม NodeMCU



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับทดสอบการติดต่อสื่อสารข้อมูลแบบมอดบัสในโหนดเอ็มซียู

```

#ifdef ESP8266
  #include <ESP8266WiFi.h>
#else //ESP32
  #include <WiFi.h>
#endif
#include <ModbusIP_ESP8266.h>
const int SENSOR_IREG = 1;
ModbusIP mb;
long ts;
int n=0;
void setup() {
  #ifdef ESP8266
    Serial.begin(74880);
  #else
    Serial.begin(115200);
  #endif
  WiFi.begin("your_ssid", "your_password");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  mb.slave(); //Start Modbus IP
  // Add SENSOR_IREG register - Use addIreg() for analog Inputs
  mb.addIreg(SENSOR_IREG);
  ts = millis();
}

void loop() {
  //Call once inside loop() - all magic here
  mb.task();

  mb.Ireg(SENSOR_IREG, n);
  n++;
  delay(10);
}

```

โปรแกรมนี้อาจจะทำการเพิ่มไลบรารีสำหรับการเชื่อมต่ออินเทอร์เน็ตบนโหนดเอ็มซียู และไลบรารีสำหรับมอดบัส และประกาศตัวแปรที่ใช้งาน จากนั้นทำการเซ็ทรหัสไวด์ไฟ และช่องออฟเซ็ทที่ใช้ติดต่อสื่อสาร ใน void setup

ในส่วนของวอยด์ลูป จะใช้คำสั่ง Mb.task เพื่อทำเป็นสัญญาณในการอัปเดตการมอดบัส แล้วทำการส่งข้อมูลให้ราสเบอร์รี่พาย ที่ตั้งไว้คือ offset = 1 ในการส่งข้อมูลจะส่งจำนวนเต็มเพิ่มขึ้น 1 ในแต่ละรอบ และแต่ละรอบจะถูกหน่วงเวลาไว้ 100 มิลลิวินาที

ตารางที่ 4.4 ผลที่ได้จากการทดสอบการสื่อสารระหว่าง Master (Raspberry pi) และ Slave (NodeMCU) เป็นเวลา 10 นาที โดยทั้ง Raspberry pi มีหน้าที่ทำการรับข้อมูล และ NodeMCU มีหน้าที่ทำการส่งข้อมูล

ครั้งที่	NodeMCU to Master (Raspberry pi)		
	ที่ได้รับ	error	รวมทั้งหมด
1	2535	2645	5180
2	2546	2634	5180
3	2587	2593	5180
4	2549	2631	5180
5	2589	2591	5180
6	2506	2674	5180
7	2519	2661	5180
8	2448	2732	5180
9	2495	2685	5180
10	2473	2707	5180
เฉลี่ย	2524.7	2655.3	
เปอร์เซ็นต์	48.74	51.26	

จากตารางที่ 4.3 จะเห็นได้ว่า Raspberry pi มีความสามารถในการรับข้อมูลจาก NodeMCU เฉลี่ยอยู่ที่ 48.74% จากการรับค่าทั้งหมด 5180 ค่า นั่นคือ ใน 10 นาที Arduino สามารถส่งข้อมูลได้ 5180 ครั้ง และ Raspberry pi สามารถรับค่าได้ประมาณ 2524 ครั้ง หรือ 4.21 ครั้งต่อวินาที

4.3 สรุป

การจำลองระบบควบคุมโดยใช้ตัวควบคุมพีไอดีของฟังก์ชันเฟิร์สอเดอร์ในโปรแกรมโคดีซีดและแมทแลบจะมีบางช่วงที่มีความใกล้เคียงกันและบางช่วงที่แตกต่างกัน แต่ผลของการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีมีแนวโน้มของผลลัพธ์ที่เหมือนหรือใกล้เคียงกัน

การสร้างโมเดลทางคณิตศาสตร์ของระบบด้วยแอปพลิเคชัน system identification ของโปรแกรม MATLAB ด้วยวิธี process model เนื่องจากวิธีนี้สามารถปรับจูนพารามิเตอร์ด้วยตัวเองเพื่อให้ระบบมีความใกล้เคียงกับข้อมูลมากยิ่งขึ้น แล้วนำโมเดลทางคณิตศาสตร์ที่ได้มาควบคุมแบบ close loop control เพื่อหาค่าพีไอดีที่เหมาะสมของระบบและง่ายต่อการสร้างระบบควบคุม ซึ่งพารามิเตอร์ที่ได้นั้นไม่ทำให้ระบบเกิดโอเวอร์ชูตและยังเข้าสู่อุณหภูมิตarget ภายในช่วงเวลาที่กำหนด



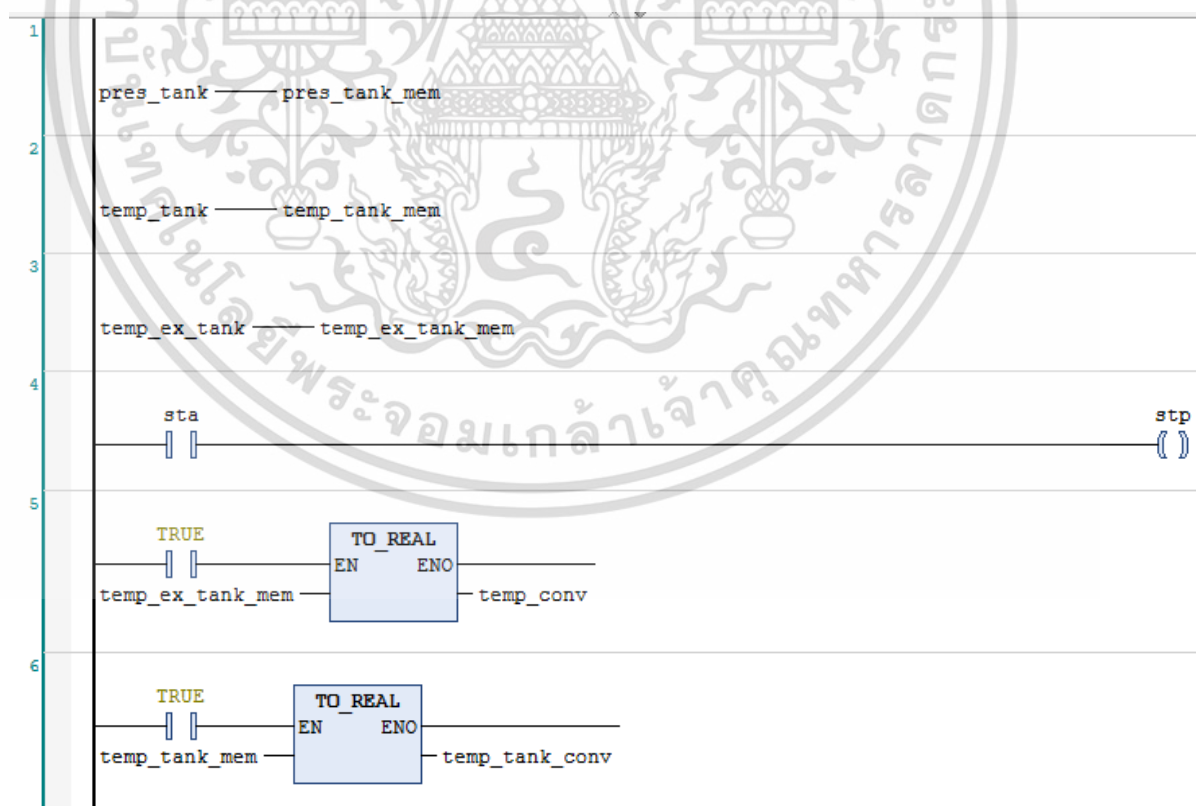
บทที่ 5

การควบคุมอุณหภูมิในระบบจริง

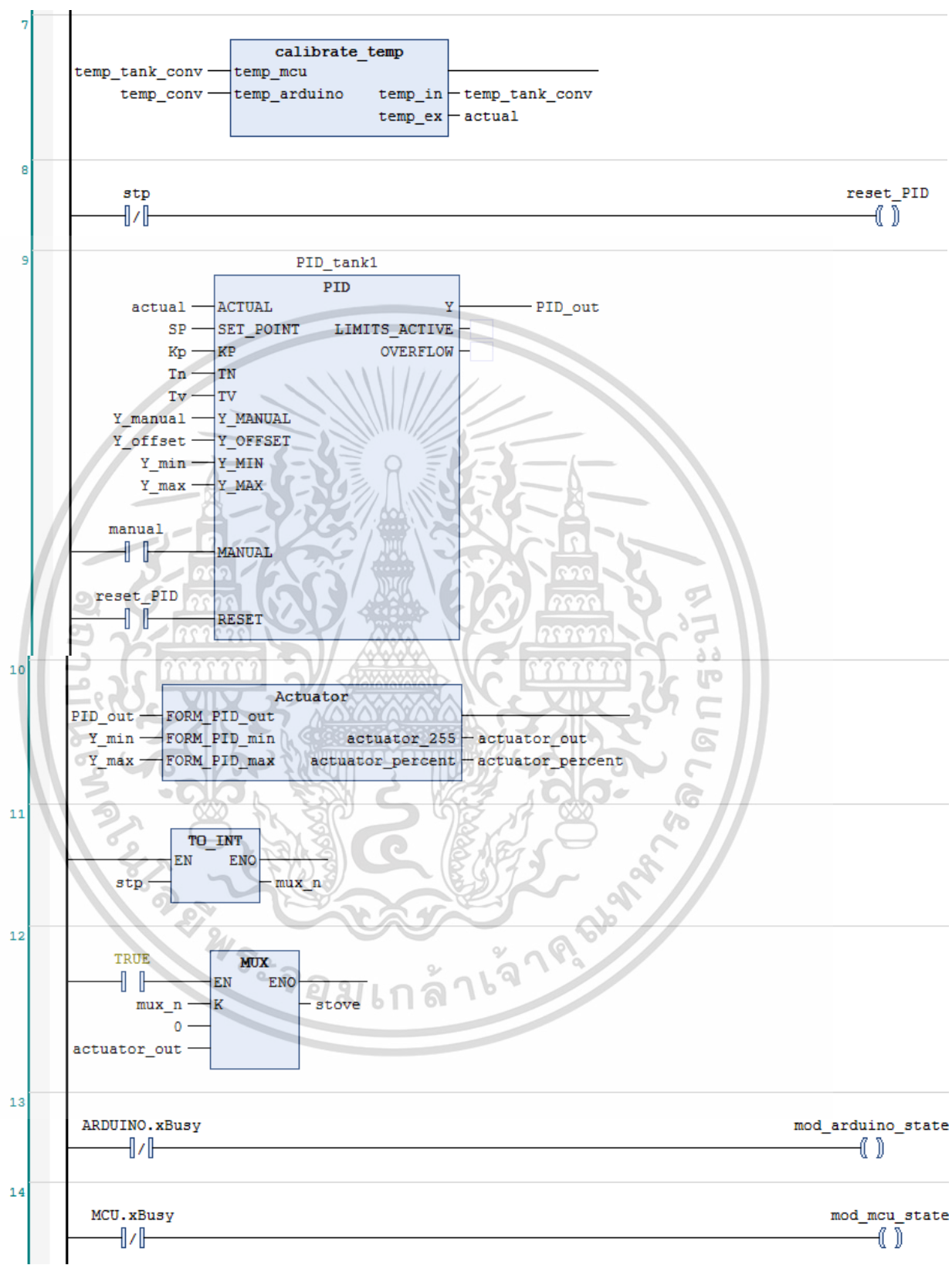
บทนี้เป็นการนำเสนอการควบคุมอุณหภูมิของระบบจริงซึ่งเป็นระบบเดียวกันกับระบบที่ใช้ในการทดลองในบทที่ 4 โดยใช้ตัวควบคุมพีไอดีด้วยโปรแกรมที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์พร้อมเปรียบเทียบค่าพารามิเตอร์ในการควบคุมและกราฟกับตัวควบคุมที่ได้จากการจำลองระบบที่ได้ทำการนำเสนอไว้ในบทที่ 4

5.1 การควบคุมอุณหภูมิในระบบจริง

ในหัวข้อนี้จะทำการควบคุมอุณหภูมิของระบบโดยใช้ราสเบอร์รี่พาย เป็นตัวควบคุมหลักเปรียบเสมือน PLC ในอุตสาหกรรม และใช้โปรแกรม CODESYS V3.5 SP15 ในการพัฒนาซอฟต์แวร์เพื่อควบคุม โดยใช้ภาษา Function Block Diagram และ structured text ในการพัฒนาซอฟต์แวร์ ซึ่งมีกระบวนการดังหัวข้อที่ 4.1



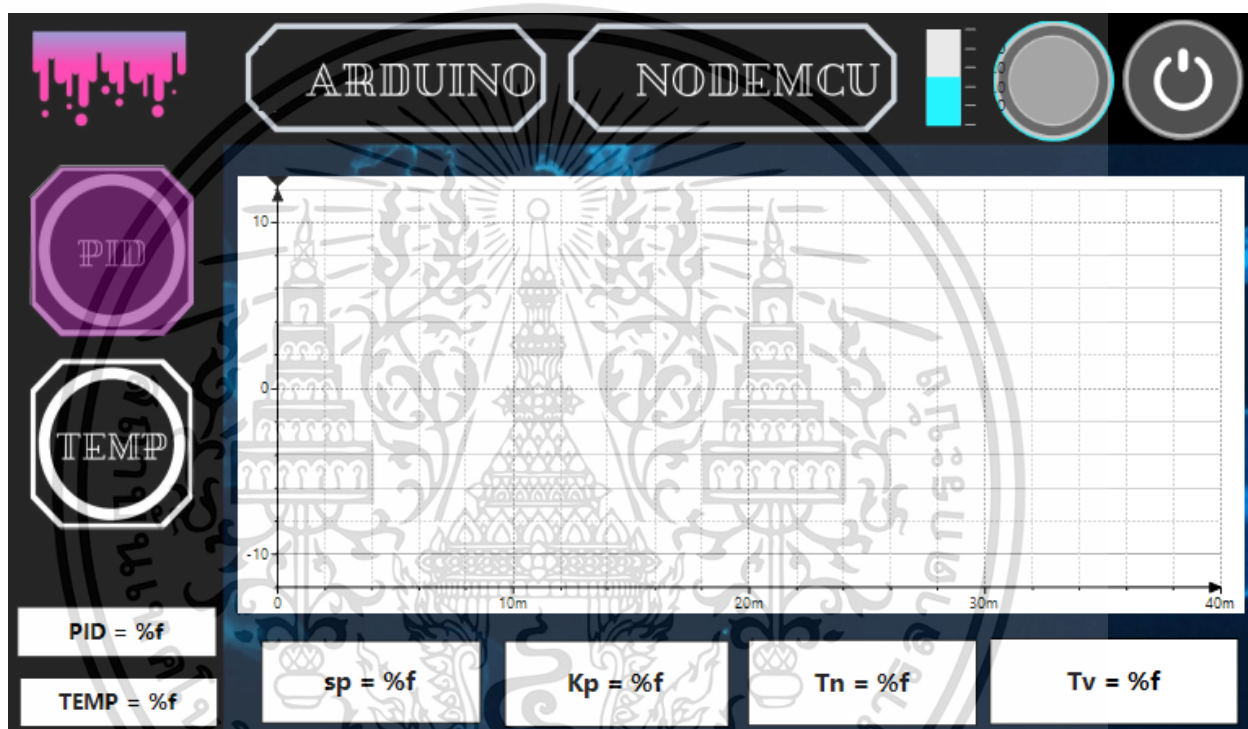
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 โปรแกรมสำหรับควบคุมระบบบน CODESYS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของโปรแกรมบล็อกที่ 1-3 จะเป็นการรับค่าความดันและอุณหภูมิของถังผสมสารเคมี และค่าอุณหภูมิของของเหลวที่ใช้ในการกระจายความร้อนจากโหนดเอ็มซียูและอาควโนตามลำดับ บล็อกที่ 4,8 มีไว้สำหรับเริ่มและหยุดกระบวนการ บล็อกที่ 5-7 ทำหน้าที่แปลงข้อมูลทางด้านจำนวนและชนิดให้ถูกต้องตรงตามที่จะใช้งาน บล็อกที่ 9 คือพีไอดีบล็อกที่ใช้สำหรับควบคุมอุณหภูมิ บล็อกที่ 10-12 ทำหน้าที่แปลงข้อมูลเพื่อที่จะแสดงผลและส่งข้อมูลไปเพื่อควบคุมเตาให้ความร้อน และบล็อกที่ 13-14 ทำหน้าที่แสดงสถานะการเชื่อมต่อของอาควโนและโหนดเอ็มซียูตามลำดับ

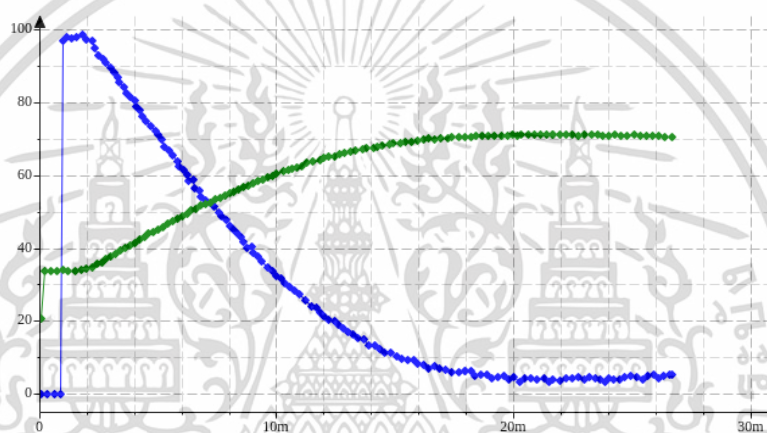


รูปที่ 5.1 หน้าจอสำหรับควบคุม และแสดงผลอุณหภูมิและผลการควบคุมของระบบ

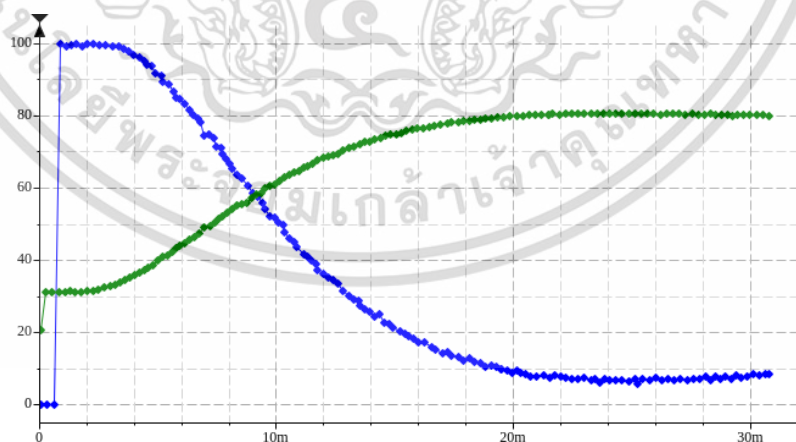
รูปที่ 5.2 แสดงหน้าจอแสดงผล HMI ที่สามารถควบคุมระบบและปรับค่าพารามิเตอร์สำหรับควบคุมระบบทั้ง ค่า K_p , T_n , T_v และค่าอุณหภูมิเป้าหมาย พร้อมทั้งยังสามารถแสดงค่าเปอร์เซ็นต์การเปิดของเตา (ค่าที่ออกมาจาก PID) และอุณหภูมิในปัจจุบันในรูปแบบตัวเลขและแบบกราฟ และยังสามารถดูสถานะการเชื่อมต่อของอาควโนและโหนดเอ็มซียูได้อีกด้วย ในการควบคุมกระบวนการได้ทดลองหาค่าพีไอดีของระบบที่เหมาะสมซึ่งได้ผลตามตารางที่ 5.1 และมีผลลัพธ์แสดงดังรูปที่ 5.3-5.5

ตารางที่ 5.1 ตารางแสดงค่าพารามิเตอร์ในการควบคุมระบบแบบพีไอดี

setpoint	Kp	Tn	Tv	Setting time(min)
70	6.5	6000	0	17
80	6.9	1260	0	19
90	7.5	1050	0	20

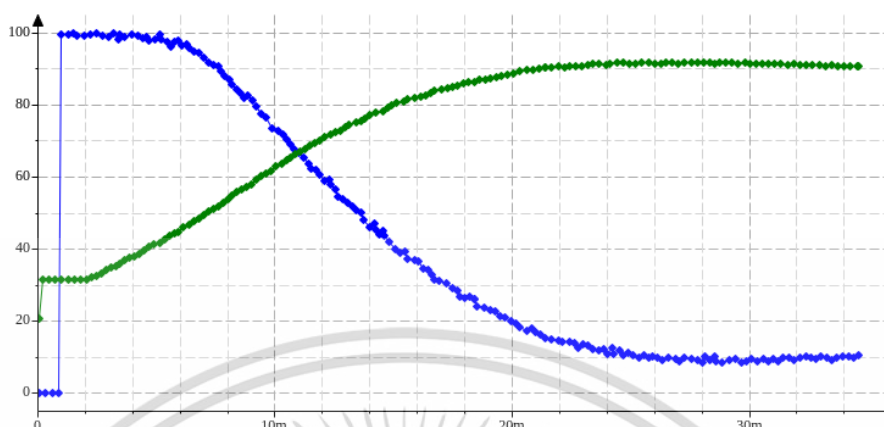


รูปที่ 5.3 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 70 องศาเซลเซียส



รูปที่ 5.4 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 80 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

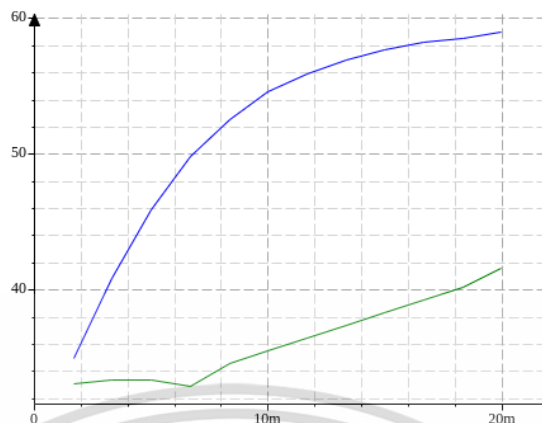


รูปที่ 5.5 กราฟแสดงอุณหภูมิและค่าพีไอดี เมื่อค่าอุณหภูมิเป้าหมาย คือ 90 องศาเซลเซียส

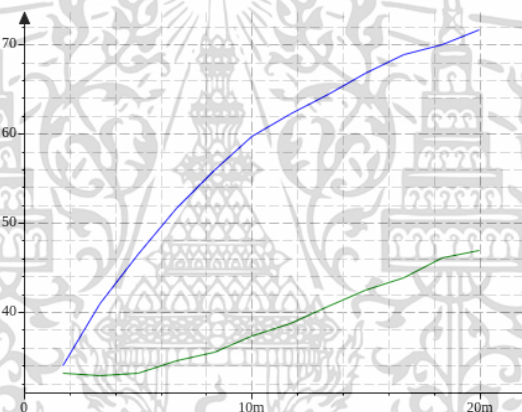
จากรูปที่ 5.3 - 5.5 และตารางที่ 5.1 แสดงให้เห็นว่าระบบควบคุมนี้มีค่าพารามิเตอร์แต่ละอุณหภูมิเป้าหมายที่แตกต่างกัน โดยค่า K_p นั้นจะช่วยให้ระบบไวขึ้นแต่ถ้ามากเกินไปจะทำให้เกิดโอเวอร์ชูตได้ ดังนั้นค่า K_p จึงมากขึ้นเมื่อมีอุณหภูมิเป้าหมายมากขึ้น ถ้าหากใช้ K_p ของอุณหภูมิเป้าหมายที่ต่ำกว่าอุณหภูมิเป้าหมายที่สูงจะทำให้ระบบช้าลง และหากใช้ K_p ของอุณหภูมิเป้าหมายที่สูงกับอุณหภูมิเป้าหมายที่ต่ำจะทำให้ระบบอาจเกิดการโอเวอร์ชูตได้ ในส่วนของค่า T_n การปรับค่าให้น้อยลงนั้นจะช่วยให้ระบบลดเกิดการออฟเซต แต่ถ้ามีค่าน้อยเกินไปจะทำให้ระบบหน่วงและอาจเกิดโอเวอร์ชูตได้ สุดท้ายคือค่า T_v เนื่องจากระบบเป็นแบบ first order การปรับค่านี้ในระบบในบางช่วงจะไม่มีผลกับระบบ แต่ถ้าหากใส่มากเกินไปจะทำให้ระบบเกิดการหน่วงและช้าลง

5.2 อุณหภูมิของ reactor tank

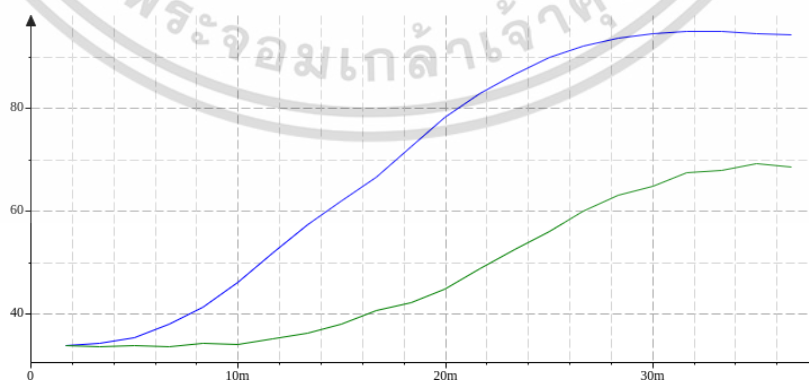
เมื่อให้ความร้อนแก่ถังระบบปิดทำให้อะตอมของสสารและอากาศที่อยู่ในถังผสม ได้รับพลังงานและทำให้โมเลกุลเกิดการขยายตัว เป็นผลให้ความดันในระบบเพิ่มสูงขึ้นจากสูตร $PV = nRT$ และนอกจากนี้สารเคมีที่ทำการผสมกันในถัง ยังทำให้คุณสมบัติต่างๆของสารเคมีเปลี่ยนแปลงไปด้วย ดังนั้นเราจึงต้องทำการติดตามผลของอุณหภูมิในถังนี้ด้วย ในหัวข้อนี้จะทำการติดตามสถานะอุณหภูมิของระบบเมื่อไม่มีการผสมสารเคมีเพื่อสังเกตพฤติกรรมของอุณหภูมิในถังผสม ในขณะที่ทำการควบคุมระบบ มีผลที่ได้ดังรูปที่ 5.6-5.8



รูปที่ 5.6 กราฟแสดงอุณหภูมิในถังผสมสารเคมี(เส้นสีเขียว) และอุณหภูมิของน้ำ(เส้นสีน้ำเงิน) เมื่อทดลองควบคุมอุณหภูมิที่ 70 องศาเซลเซียส



รูปที่ 5.7 กราฟแสดงอุณหภูมิในถังผสมสารเคมี(เส้นสีเขียว) และอุณหภูมิของน้ำ(เส้นสีน้ำเงิน) เมื่อทดลองควบคุมอุณหภูมิที่ 80 องศาเซลเซียส



รูปที่ 5.8 กราฟแสดงอุณหภูมิในถังผสมสารเคมี(เส้นสีเขียว) และอุณหภูมิของน้ำ(เส้นสีน้ำเงิน) เมื่อควบคุมอุณหภูมิที่ 90 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.6-5.8 จะเห็นได้ว่าอุณหภูมิในถังผสมสารเคมีนั้นมีอัตราการเปลี่ยนแปลงและมีแอมพลิจูดที่น้อยกว่าอุณหภูมิของน้ำที่เป็นตัวกระจายความร้อน เนื่องจากเซนเซอร์ที่ใช้สำหรับวัดอุณหภูมิถูกติดตั้งไว้บริเวณฝาของถังผสมสารทำให้ทำให้อุณหภูมิจากภายนอกมีการรบกวน และยังเป็นบริเวณที่ระบบมีการถ่ายเทความร้อนสู่ภายนอกสูง

5.3 สรุป

ในบทที่ 5 เป็นการหาพารามิเตอร์ในการควบคุมระบบแบบพีไอดีที่เหมาะสม ซึ่งพารามิเตอร์สำหรับควบคุมเหล่านี้ไม่มีความสัมพันธ์กับกันทำให้ยากต่อการคาดเดาพารามิเตอร์มีใช้กับอุณหภูมิเป้าหมายอื่นๆ ดังนั้นระบบนี้จึงต้องใช้พารามิเตอร์ในการควบคุมที่เฉพาะในแต่ละอุณหภูมิเป้าหมาย ในระหว่างควบคุมอุณหภูมิที่วัดได้มีการแกว่งไปมาเนื่องจากความไม่แน่นอนของการวัด ซึ่งเป็นสาเหตุที่ทำให้ค่าที่ออกมาจากยบล็อกพีไอดีมีการแกว่งไปมาทำให้ค่าที่ใช้ในการควบคุมเตาแกว่งไปมา ทำให้การการกำหนดเกณฑ์ขยายของตัวควบคุมหรือ K_p ถ้ามีค่ามากเกินไปจะทำให้เกิดการแกว่งของค่าที่ออกมาจากบล็อกพีไอดีมากเกินไป

บทที่ 6

สรุปผลการทดลองและข้อเสนอแนะ

6.1 สรุปผล

วิทยานิพนธ์นี้มีจุดประสงค์เพื่อออกแบบระบบควบคุมกระบวนการ เพื่อควบคุมอุณหภูมิของถังผสมสารเคมีระบบปิด โดยเริ่มจากการออกแบบและประกอบเครื่องมือที่ใช้ควบคุม แล้วหาโมเดลทางคณิตศาสตร์และจำลองระบบเพื่อหาพารามิเตอร์ในการควบคุมระบบแบบพีโอดี และนำความสัมพันธ์ของพารามิเตอร์ของระบบมาประยุกต์ใช้ในระบบจริง

ในการหาโมเดลทางคณิตศาสตร์ของระบบจะเก็บข้อมูลจากการเปิดเตาให้ความร้อนและวัดค่าอุณหภูมิ เนื่องจากของเหลวที่ใช้กระจายความร้อนของระบบคือน้ำ ซึ่งมีจุดเดือดประมาณ 100 องศาเซลเซียส และอุณหภูมิของน้ำจะไม่มีทางเกินจุดเดือดของตัวเองดังนั้นจึงเก็บข้อมูลข้อมูลจากการเปิดเตาให้ความร้อนเพียง 20% ซึ่งมีอุณหภูมิสูงสุดอยู่ที่ 81 องศาเซลเซียส โมเดลทางคณิตศาสตร์ที่ได้จากระบบคือ $G = \frac{4.091 \exp(-100s)}{2108.6s+1}$ มีความแตกต่างจากข้อมูล 80.06%

ในการจำลองระบบโดยใช้โมเดลทางคณิตศาสตร์ที่ได้ด้วย close loop control เพื่อหาพารามิเตอร์สำหรับควบคุมระบบ ผลที่ได้คือ ค่า K_p มีค่าเท่ากับ 1.98 และ K_i มีความสัมพันธ์กันแบบเป็นเชิงเส้นมีสมการคือ $K_i = (-6 \times 10^{-6})T + 0.00121$ เมื่อ T คืออุณหภูมิเป้าหมาย และใช้เวลาทำให้ค่าเข้าสู่อุณหภูมิเป้าหมายได้ไม่เกิด 20 นาทีตามกำหนด

การหาพารามิเตอร์ในระบบจริงนั้นจะต้องกำหนดค่า T_n ไว้นสูงที่สุดเท่าที่จะทำได้ เนื่องจากพารามิเตอร์ในระบบจริงที่มีพารามิเตอร์แตกต่างจากระบบจำลองคือ ค่า $T_n = K_p/K_i$ และ $T_v = K_d/K_p$ ดังนั้นถ้าหากตัวแปร T_n มีค่าเป็น 0 จะทำให้ค่า K_i เป็นอินฟินิตีมีผลให้บล็อกพีโอดีไม่ทำงาน ในการกำหนดค่า K_p ในแต่ละอุณหภูมิเป้าหมายของระบบ โดยจะเพิ่มค่าขึ้นจนถึงจุดหนึ่งเพื่อให้ระบบไม่เกิดโอเวอร์ชูดผลที่ได้ คือ $K_p = 6.5$ ที่อุณหภูมิเป้าหมายที่ 70 องศาเซลเซียส, $K_p = 6.9$ ที่อุณหภูมิเป้าหมายที่ 80 องศาเซลเซียส และ $K_p = 7.5$ ที่อุณหภูมิเป้าหมายที่ 90 องศาเซลเซียส เมื่อได้ค่า K_p แล้วจึงทำการลดค่า T_n เพื่อลดการเกิดออฟเซตและทำให้ระบบเข้าสู่อุณหภูมิเป้าหมายได้อย่างถูกต้องหรือใกล้เคียงที่สุดผลที่ได้ คือคือ $T_n = 6000$ ที่อุณหภูมิเป้าหมายที่ 70 องศาเซลเซียส, $T_n = 1260$ ที่อุณหภูมิเป้าหมายที่ 80 องศาเซลเซียส และ $T_n = 1050$ ที่อุณหภูมิเป้าหมายที่ 90 องศาเซลเซียส โดยค่าพารามิเตอร์ของแต่ละอุณหภูมิเป้าหมายนั้น มีความแตกต่างกันและยากที่จะคาดเดาที่จะให้พารามิเตอร์เหล่านี้ทำให้ระบบเข้าสู่อุณหภูมิเป้าหมายได้อย่างพอดีและไม่เกิดโอเวอร์ชูด ดังนั้นค่าพารามิเตอร์ของแต่ละอุณหภูมิเป้าหมายจะถูกกำหนดไว้เพื่อให้ระบบเข้าสู่อุณหภูมิเป้าหมายได้อย่างพอดีและไม่เกิดโอเวอร์ชูด

จากการศึกษาพบว่าโมเดลทางคณิตศาสตร์นี้ยังเหมาะสมไม่เพียงพอที่จะกำหนดพารามิเตอร์ในการควบคุมของระบบจำลองและระบบจริงมีความใกล้เคียงกัน เนื่องจากระบบจริงยังมีฟังก์ชันพีเอ็ดที่แตกต่างกันจากระบบจำลองทำให้พารามิเตอร์และผลลัพธ์มีความแตกต่างกัน นอกจากนี้ในระบบจริงนั้นยังมีสิ่งรบกวนอยู่มากทั้งอุณหภูมิห้องที่แต่ละช่วงเวลามีค่าไม่เหมือนกันหรืออาจมีผลมาจากอุปกรณ์ ทำให้มีผลกระทบต่อการศึกษาหาค่าพารามิเตอร์และระยะเวลาเข้าสู่อุณหภูมิเป้าหมาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Alvarez-Ramirez, J. (2005). *Robust temperature control for batch chemical reactors*. Mexico.
- [2] Andresarmento. (2015, 11 11). *Modbus Library for Arduino*. Retrieved from <https://github.com/andresarmento/modbus-arduino>
- [3] Arduino. (n.d.). *Arduino Ethernet Shield V1*. Retrieved from <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>
- [4] Bars, R. (2019). *Control Engineering: MATLAB Exercises*. The Mathworks.
- [5] chrisbeardy. (2016). *Programming Raspberry Pi With CODESYS*. Retrieved from <https://www.instructables.com/id/Programming-Raspberry-Pi-With-CODESYS/>
- [6] Components101. (2020, 5 22). *NodeMCU ESP8266*. Retrieved from <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>
- [7] Desborough, L. R. (2000). *Regulatory Control Survey, Honeywell*, . unpublished manuscript.
- [8] Electronics, B. (2014). *CODESYS Introduction. Automation Industry news that puts your finger on the pulse.*
- [9] Emelianov. (2015, 10 21). *Modbus Library for ESP8266*. Retrieved from <https://github.com/andresarmento/modbus-esp8266>
- [10] Hahn, B. D. (2019). *Essential MATLAB for Engineers and Scientists, 7th edition*. The MathWorks, Inc.
- [11] Halfacree, G. (2019). *Raspberry Pi Beginner's Guide*. Cambridge,; Raspberry Pi Trading Ltd.
- [12] Hanssen, D. H. (2015). *Programmable*. Norway: Institute of Engineering and Safety, University of Tromsø.
- [13] Johnson, M. A. (2005). *PID Control New Identification and Design Methods*. London: Springer-Verlag London Limited.

- [14] Kevin Burn and Chris Cox. (n.d.). *Curve fitting software for first order plus dead time*. United Kingdom: Department of Computing, Engineering and Technology, University of Sunderland,.
- [15] MathWorks, T. (2020). *MATLAB Programming Fundamentals*. The MathWorks, Inc.
- [16] Modicon. (1996). *Modbus Protocol Reference Guide*. Massachusetts: Industrial Automation Systems.
- [17] O'DWYER, A. (2003). *Handbook of PI and PID Controller Tuning Rules*. London: Imperial College Press.
- [18] Parker. (2018, 10 31). *SAMPLE CONTINUOUS FUNCTION CHART CFC PROGRAM: TIME BASED COUNTER*. Retrieved from <https://www.printfriendly.com/p/g/kYSDWy>
- [19] Peter. (2015, 6 21). *Structured Text Tutorial to Expand Your PLC Programming Skills*. Retrieved from <https://www.plcademy.com/structured-text-tutorial/>
- [20] Peter. (2018, 3 13). *Function Block Diagram (FBD) Programming Tutorial*. Retrieved from <https://www.plcademy.com/function-block-diagram-programming/>
- [21] *Principle of heat transfer*. (n.d.). Retrieved from http://www2.dede.go.th/bhrd/old/Download/file_handbook/Pre_Heat/pre_heat_4.pdf
- [22] Sharma, K. (2011). *Closed Loop Control*. Retrieved from <https://www.sciencedirect.com/topics/engineering/closed-loop-control>
- [23] Teschl, G. (2012). *Ordinary Differential Equations and Dynamical Systems*. Providence: American Mathematical Society.
- [24] Thaieasyelec. (2016, oct 7). *บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt*. Retrieved from <https://www.thaieasyelec.com/article-wiki/embedded-electronics-application/raspberry-pi-programming-with-qt-ch1.html>
- [25] yan, C. (2562, กันยายน 2). *ระบบควบคุมพีไอดี*. Retrieved from <https://th.wikipedia.org/wiki/%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A>

%E0%B8%84%E0%B8%A7%E0%B8%9A%E0%B8%84%E0%B8%B8%E0%B8%A1%E
 0%B8%9E%E0%B8%B5%E0%B9%84%E0%B8%AD%E0%B8%94%E0%B8%B5

[26] นาคะบุตร, ร. (2546). *ตัวแบบเชิงคณิตศาสตร์*. นครปฐม.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมอาตุโนสำหรับควบคุมเตาให้ความร้อน และรับค่ามาจากราสเบอร์รี่พาย
ผ่านการมอดบัส

```

#include <SPI.h>
#include <Ethernet.h>
#include "Mudbus.h"
#include <Smoothed.h> // Include the library
#define SENSOR_PIN A0
int temp,read0;
word con;
Mudbus Mb;
Smoothed <float> mySensor;
//Function codes 1(read coils), 3(read registers), 5(write coil), 6(write register)
//signed int Mb.R[0 to 125] and bool Mb.C[0 to 128] MB_N_R MB_N_C
//Port 502 (defined in Mudbus.h) MB_PORT

void setup()
{
  uint8_t mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[] = { 192, 168, 1, 4 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[] = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet);
  //Avoid pins 4,10,11,12,13 when using ethernet shield

  delay(5000);
  Serial.begin(9600);
  mySensor.begin(SMOOTHED_AVERAGE, 5);
  pinMode(7, INPUT);
  pinMode(8, OUTPUT);
  digitalWrite(13,HIGH);
}

void loop()
{
  Mb.Run();
  read0 = analogRead(A0);
  mySensor.add(read0);
  temp = mySensor.get();
  Mb.R[1] = temp; // ส่งค่าให้ raspberry pi
  con = Mb.R[3]; //อ่านค่าจาก raspberry pi
  analogWrite(3, con);
  float lastValueStoredAvg = mySensor.getLast();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปรแกรม NodeMCU ส่งค่าอุณหภูมิภายในถึงผสมสารเคมี ให้ raspberry pi
ผ่านการ MODBUS

```

/*
 * More Tutorials:
 * Website http://trialcommand.com
 * In English: http://en.trialcommand.com
 * En Español: http://en.trialcommand.com
 */
#include <ESP8266WiFi.h>
#include <SPI.h>

#define MAX6675_CS 15
#define MAX6675_SO 12
#define MAX6675_SCK 14

const char* ssid = "Tenda_8DA328";
const char* password = "12345678";
int ModbusTCP_port = 502;
word temp, pressure;

////////// Required for Modbus TCP / IP /// Requerido para Modbus TCP/IP //////////
#define maxInputRegister 20
#define maxHoldingRegister 20

#define MB_FC_NONE 0
#define MB_FC_READ_REGISTERS 3 //implemented
#define MB_FC_WRITE_REGISTER 6 //implemented
#define MB_FC_WRITE_MULTIPLE_REGISTERS 16 //implemented
//
// MODBUS Error Codes
//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MB_FC_NONE 0
#define MB_FC_READ_REGISTERS 3 //implemented
#define MB_FC_WRITE_REGISTER 6 //implemented
#define MB_FC_WRITE_MULTIPLE_REGISTERS 16 //implemented
//
// MODBUS Error Codes
//
#define MB_EC_NONE 0
#define MB_EC_ILLEGAL_FUNCTION 1
#define MB_EC_ILLEGAL_DATA_ADDRESS 2
#define MB_EC_ILLEGAL_DATA_VALUE 3
#define MB_EC_SLAVE_DEVICE_FAILURE 4
//
// MODBUS MBAP offsets
//
#define MB_TCP_TID 0
#define MB_TCP_PID 2
#define MB_TCP_LEN 4
#define MB_TCP_UID 6
#define MB_TCP_FUNC 7
#define MB_TCP_REGISTER_START 8
#define MB_TCP_REGISTER_NUMBER 10

byte ByteArray[260];
unsigned int MBHoldingRegister[maxHoldingRegister];

////////////////////////////////////

WiFiServer MBServer (ModbusTCP_port);

void setup() {

  pinMode(14, OUTPUT);

  Serial.begin(9600);
  delay(100);
  WiFi.begin(ssid, password);
  delay(100);
  Serial.println(".");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  MBServer.begin();
  Serial.println("Connected ");
  Serial.print("ESP8266 Slave Modbus TCP/IP ");
  Serial.print(WiFi.localIP());
  Serial.print(":");
  Serial.println(String(ModbusTCP_port));
  Serial.println("Modbus TCP/IP Online");
  digitalWrite(16, HIGH);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void loop() {

// Check if a client has connected // Modbus TCP/IP
WiFiClient client = MBServer.available();
if (!client) {
return;
}

boolean flagClientConnected = 0;
byte byteFN = MB_FC_NONE;
int Start;
int WordDataLength;
int ByteDataLength;
int MessageLength;

// Modbus TCP/IP
while (client.connected()) {

if(client.available())
{
flagClientConnected = 1;
int i = 0;
while(client.available())
{
ByteArray[i] = client.read();
i++;
}

client.flush();

temp = readThermocouple();
pressure = analogRead(A0);

MBHoldingRegister[0] = temp;
MBHoldingRegister[1] = pressure;

Serial.print("temp : ");
//Serial.print('c');
Serial.print(temp);
Serial.print("pressure : ");

Serial.println(pressure);
delay(100);
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//// routine Modbus TCP
byteFN = ByteArray[MB_TCP_FUNC];
Start = word(ByteArray[MB_TCP_REGISTER_START], ByteArray[MB_TCP_REGISTER_START+1]);
WordDataLength = word(ByteArray[MB_TCP_REGISTER_NUMBER], ByteArray[MB_TCP_REGISTER_NUMBER+1]);
}

// Handle request

switch(byteFN) {
case MB_FC_NONE:
break;

case MB_FC_READ_REGISTERS: // 03 Read Holding Registers
ByteDataLength = WordDataLength * 2;
ByteArray[5] = ByteDataLength + 3; //Number of bytes after this one.
ByteArray[8] = ByteDataLength; //Number of bytes after this one (or number of bytes of data).
for(int i = 0; i < WordDataLength; i++)
{
ByteArray[ 9 + i * 2] = highByte(MB HoldingRegister[Start + i]);
ByteArray[10 + i * 2] = lowByte(MB HoldingRegister[Start + i]);
}
MessageLength = ByteDataLength + 9;
client.write((const uint8_t *)ByteArray, MessageLength);

byteFN = MB_FC_NONE;
break;

case MB_FC_WRITE_REGISTER: // 06 Write Holding Register
MB HoldingRegister[Start] = word(ByteArray[MB_TCP_REGISTER_NUMBER], ByteArray[MB_TCP_REGISTER_NUMBER+1]);
ByteArray[5] = 6; //Number of bytes after this one.
MessageLength = 12;
client.write((const uint8_t *)ByteArray, MessageLength);
byteFN = MB_FC_NONE;
break;

case MB_FC_WRITE_MULTIPLE_REGISTERS: //16 Write Holding Registers
ByteDataLength = WordDataLength * 2;
ByteArray[5] = ByteDataLength + 3; //Number of bytes after this one.
for(int i = 0; i < WordDataLength; i++)
{
MB HoldingRegister[Start + i] = word(ByteArray[ 13 + i * 2], ByteArray[14 + i * 2]);
}
MessageLength = 12;
client.write((const uint8_t *)ByteArray, MessageLength);
byteFN = MB_FC_NONE;

break;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double readThermocouple() {

    uint16_t v;
    pinMode(MAX6675_CS, OUTPUT);
    pinMode(MAX6675_SO, INPUT);
    pinMode(MAX6675_SCK, OUTPUT);

    digitalWrite(MAX6675_CS, LOW);
    delay(1);

    v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
    v <<= 8;
    v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

    digitalWrite(MAX6675_CS, HIGH);
    if (v & 0x4)
    {
        // Bit 2 indicates if the thermocouple is disconnected
        return NAN;
    }

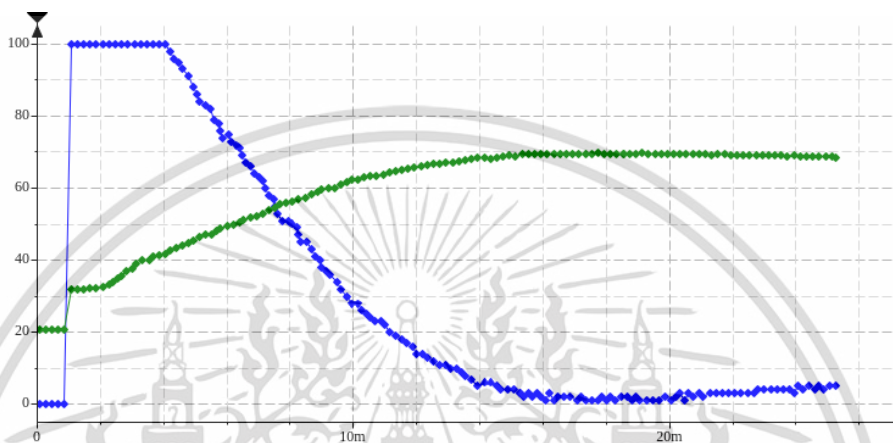
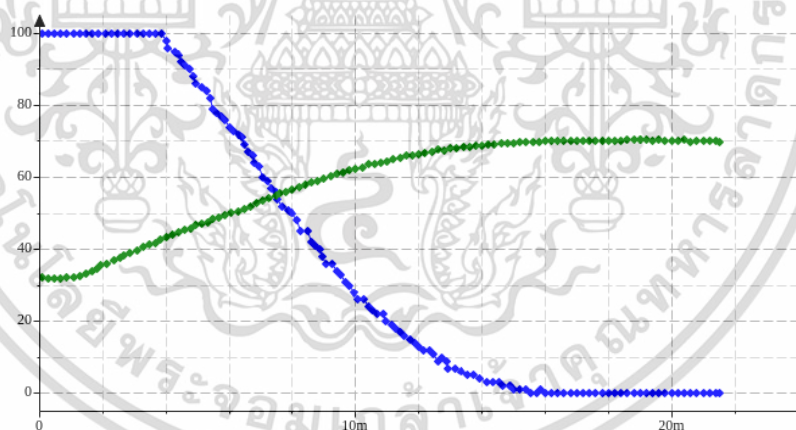
    // The lower three bits (0,1,2) are discarded status bits
    v >>= 3;
    return v*1;
}

```

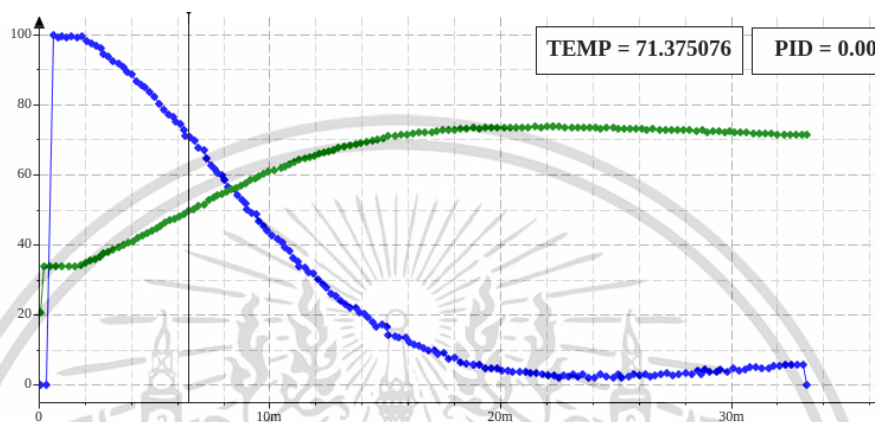
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

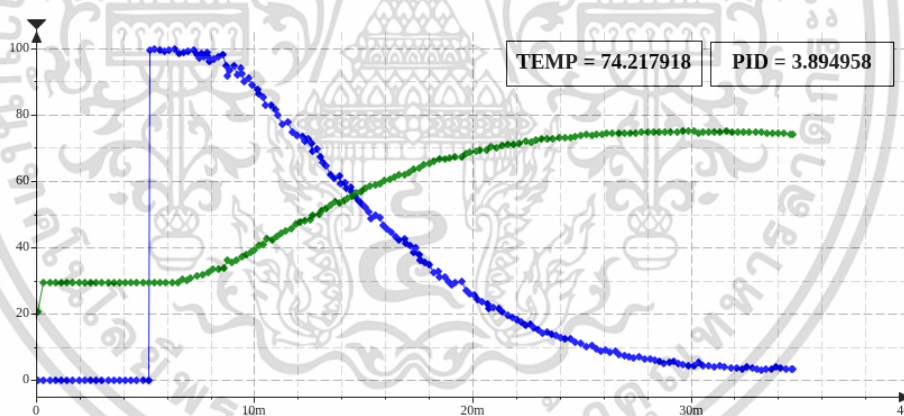
ค.1 ผลการทดลองการควบคุมแบบ PD ที่อุณหภูมิ 70 องศาเซลเซียส

รูปที่ ค.1 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 3.6$ และ $T_v = 0$ รูปที่ ค.2 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 3.7$ และ $T_v = 0$

ค.2 ผลการทดลองการควบคุมแบบ PID ที่อุณหภูมิ 70 องศาเซลเซียส

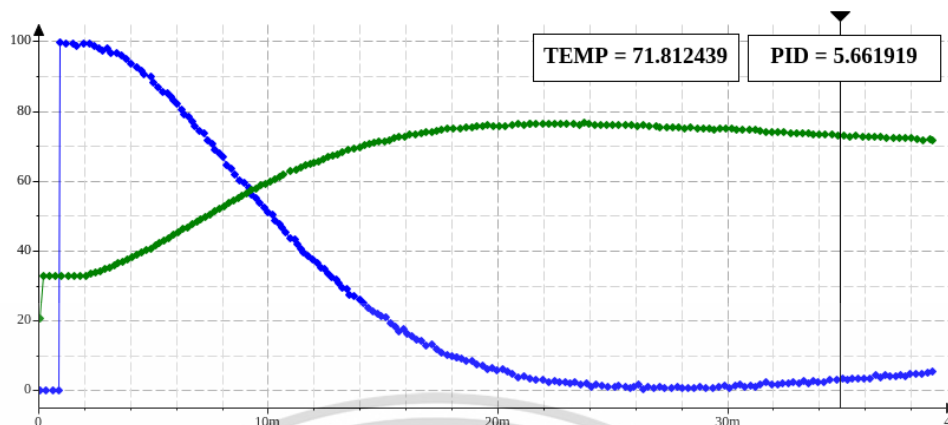


รูปที่ ค.3 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 8$, $T_n = 1255$ และ $T_v = 0$

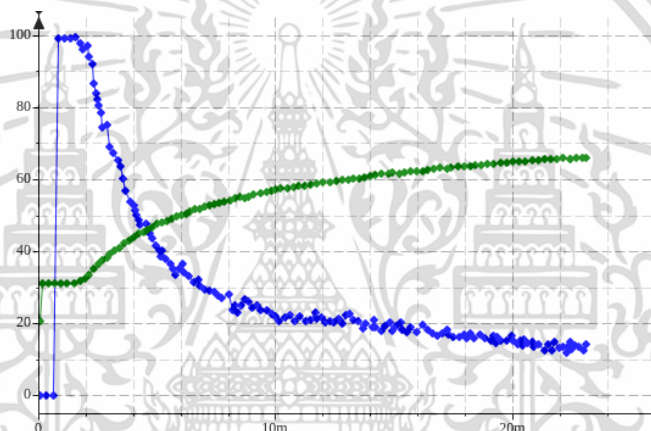


รูปที่ ค.4 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 7$, $T_n = 1255$ และ $T_v = 0$

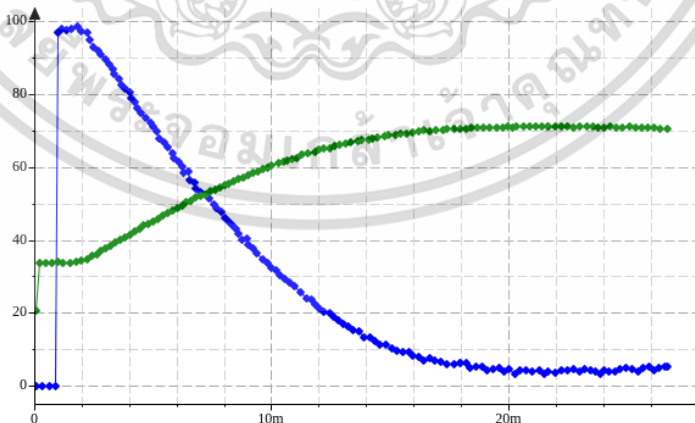
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.5 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 7$, $T_n = 1255$ และ $T_v = 0.000001$

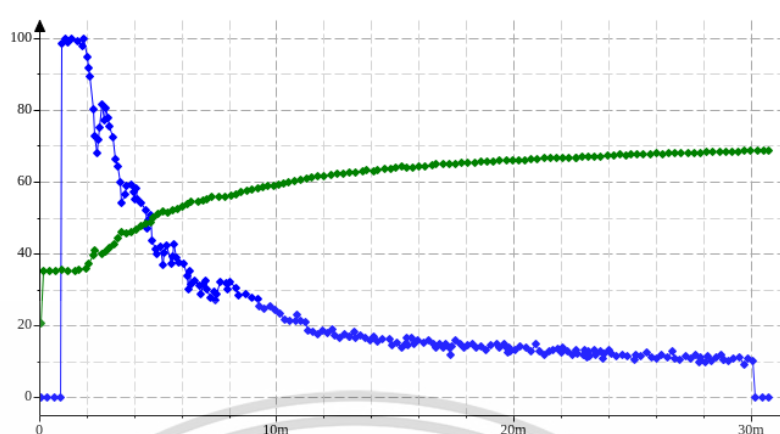


รูปที่ ค.6 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 7$, $T_n = 750$ และ $T_v = 0$

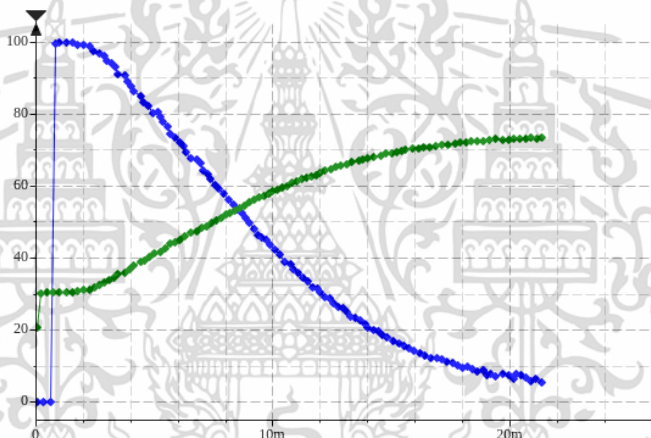


รูปที่ ค.7 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 6.9$, $T_n = 5000$ และ $T_v = 0$

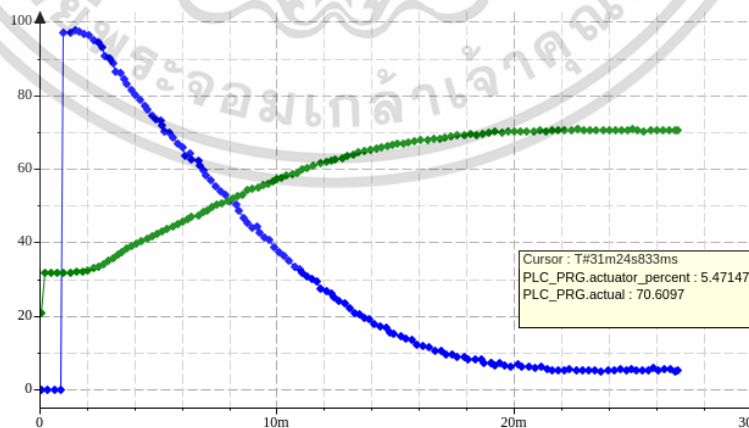
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.8 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 6.5$, $T_n = 750$ และ $T_v = 0$



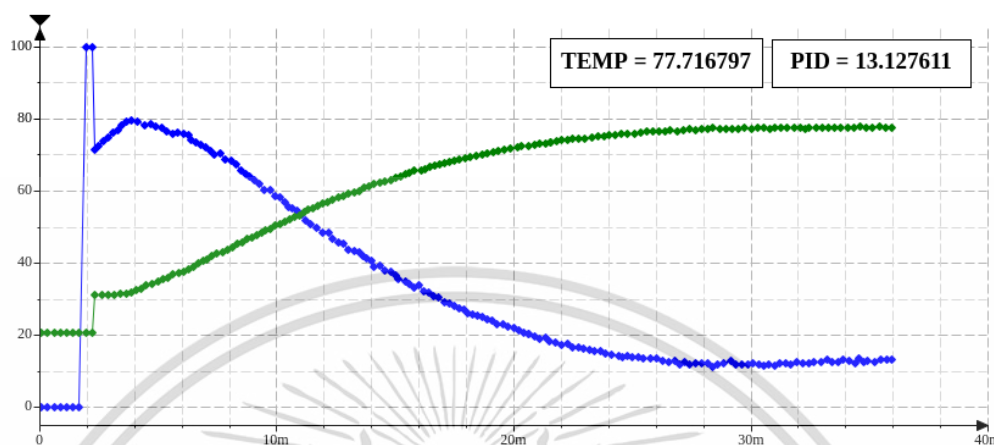
รูปที่ ค.9 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 6.5$, $T_n = 2250$ และ $T_v = 0$



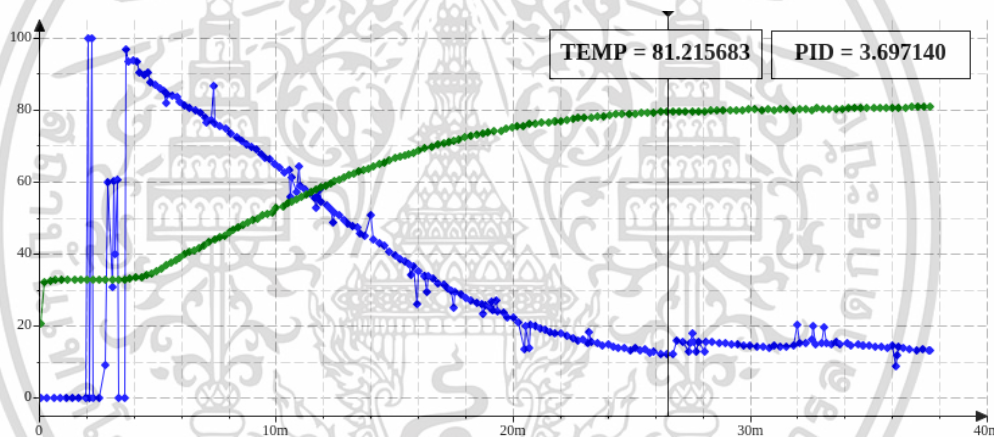
รูปที่ ค.10 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 70, $K_p = 6.5$, $T_n = 6000$ และ $T_v = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

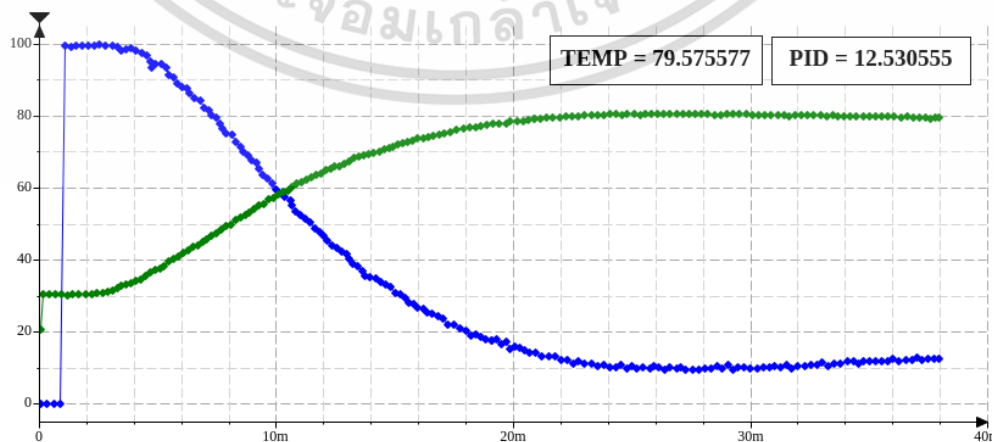
ค.3 ผลการทดลองการควบคุมแบบ PID ที่อุณหภูมิ 80 องศาเซลเซียส



รูปที่ ค.11 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1300$ และ $T_v = 0$

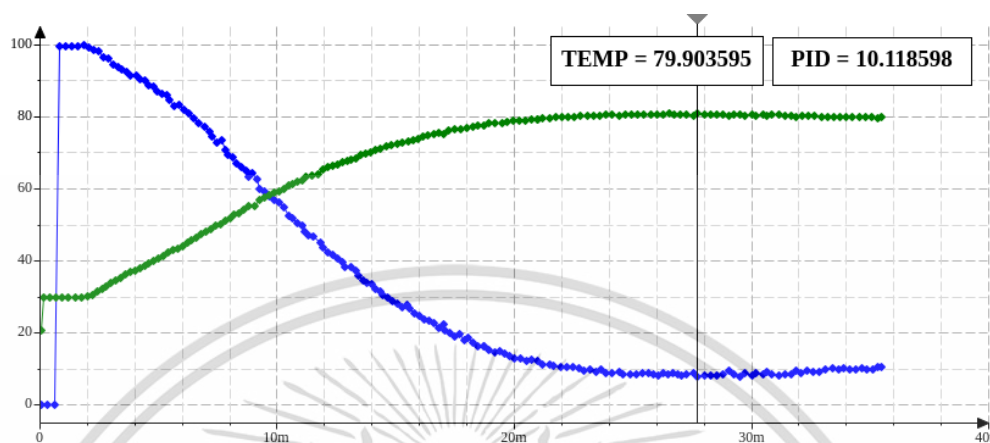


รูปที่ ค.12 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1200$ และ $T_v = 0$

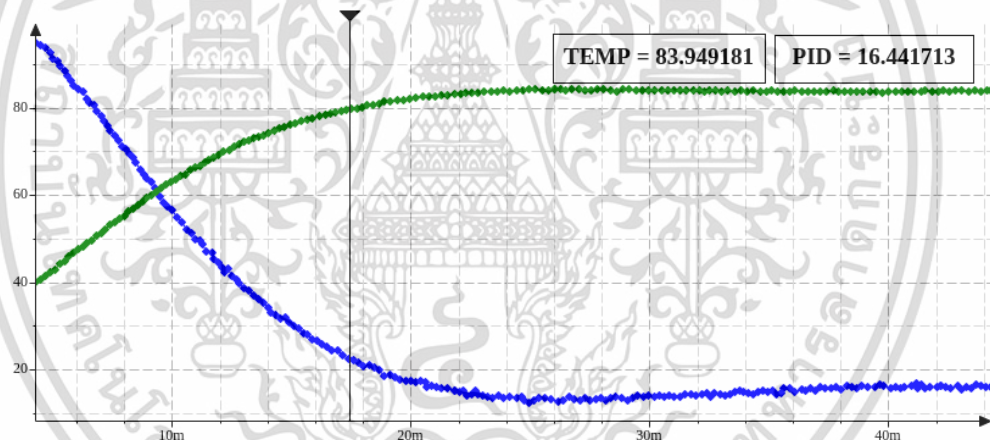


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.13 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1250$ และ $T_v = 0$

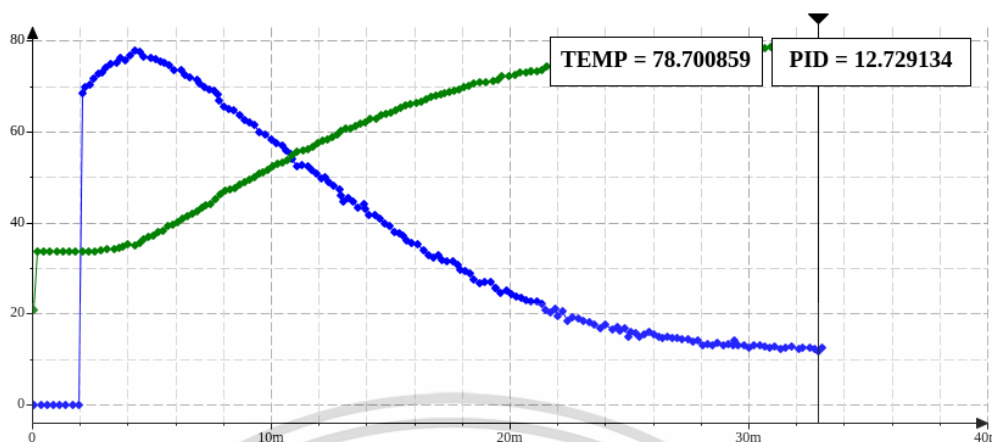


รูปที่ ค.14 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1270$ และ $T_v = 0$

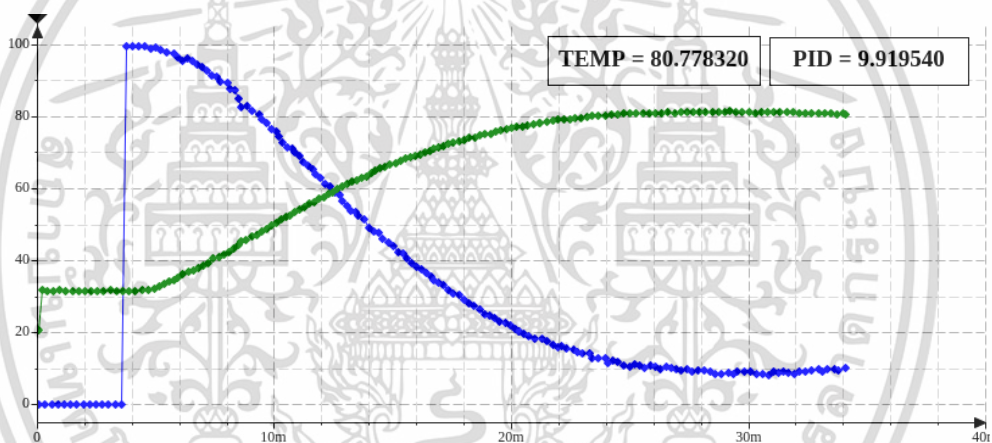


รูปที่ ค.15 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 85, $K_p = 7$, $T_n = 1255$ และ $T_v = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



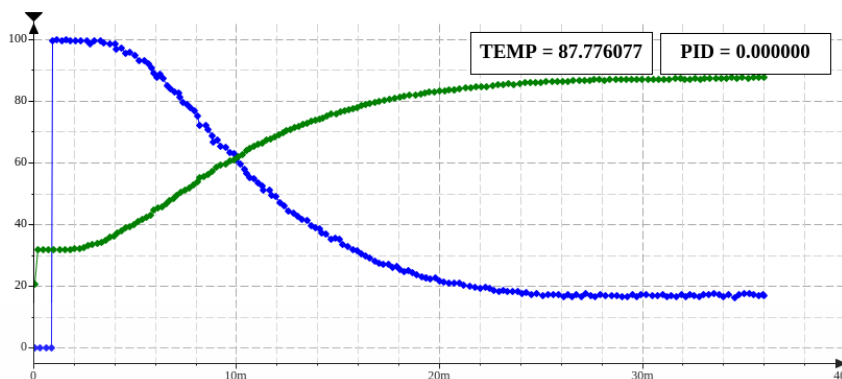
รูปที่ ค.16 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1255$ และ $T_v = 0.001$



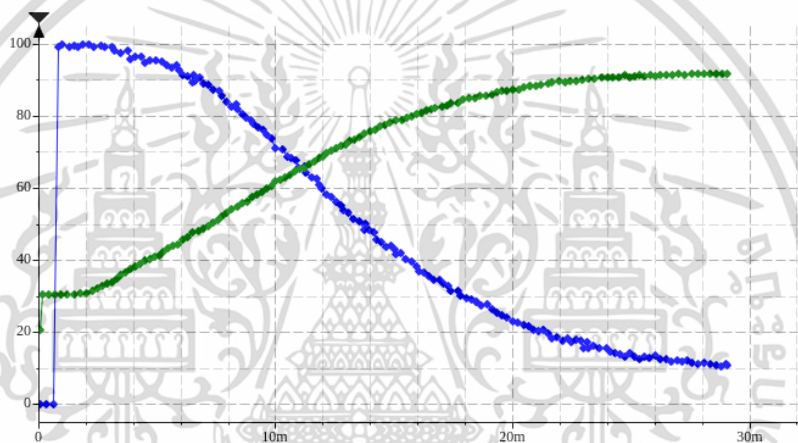
รูปที่ ค.17 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 80, $K_p = 7$, $T_n = 1255$ และ $T_v = 0.000001$

ค.3 ผลการทดลองการควบคุมแบบ PID ที่อุณหภูมิ 90 องศาเซลเซียส

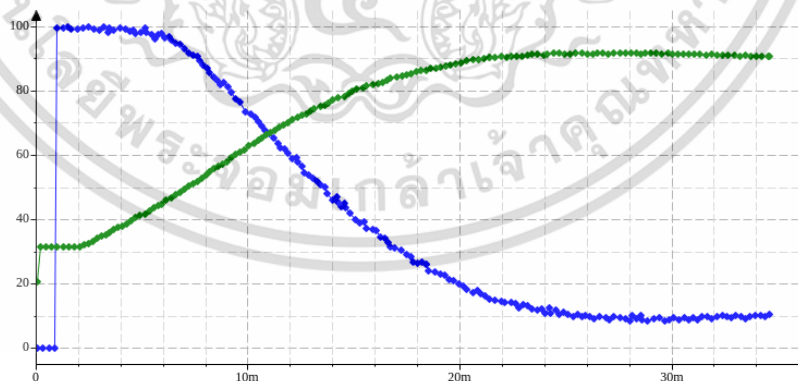
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.18 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 90, $K_p = 7$, $T_n = 1255$ และ $T_v = 0$



รูปที่ ค.18 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมายคือ 90, $K_p = 7.5$, $T_n = 1000$ และ $T_v = 0$



รูปที่ ค.19 การควบคุมในระบบจริงเมื่อ อุณหภูมิเป้าหมาย คือ 90, $K_p = 7.5$, $T_n = 900$ และ $T_v = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้