



การออกแบบและสร้างเตาความร้อนสูงแบบควบคุมและแสดงผลตามเวลาจริง

นายธรรมกรานต์ พังยะ

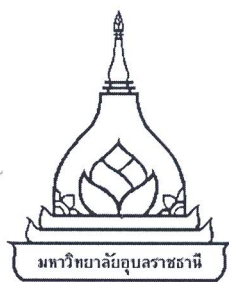
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2561
ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี



DESIGN AND CONSTRUCTION OF CONTROL AND REAL TIME
MORNITORING FOR HIGH TEMPERATURE FURNACE

TOMMAKRAN PANGYA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
MAJOR IN PHYSICS
FACULTY OF SCIENCE
UBON RATCHATHANI UNIVERSITY
ACADEMIC YEAR 2018
COPYRIGHT OF UBON RATCHATHANI UNIVERSITY



ใบรับรองวิทยานิพนธ์
มหาวิทยาลัยอุบลราชธานี
ปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์

เรื่อง การออกแบบและสร้างเตาความร้อนสูงแบบควบคุมและแสดงผลตามเวลาจริง

ผู้วิจัย นายธรรมกรานต์ พังยะ

คณะกรรมการสอบ

ผู้ช่วยศาสตราจารย์ ดร.ชีวะ ทัดนา

ประธานกรรมการ

ดร.อมร เทศสกุลวงศ์

กรรมการ

ดร.สมคิด เพ็ญขารี

กรรมการ

อาจารย์ที่ปรึกษา

(ดร.อมร เทศสกุลวงศ์)

ชรีม

(ผู้ช่วยศาสตราจารย์ ดร.ชรีดา ปุกหุด)

คณบดีคณะวิทยาศาสตร์

(รองศาสตราจารย์ ดร.อริยาภรณ์ พงษ์รัตน์)

รองอธิการบดีฝ่ายวิชาการ

ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2561

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ด้วยความกรุณาจาก ดร.อมร เทศสกุลวงศ์ ผู้เป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้วิจัยขอขอบพระคุณที่ท่านคอยช่วยเหลือปรึกษาให้คำแนะนำ และคอยแนะนำวิธีการแก้ไขข้อผิดพลาดจนวิทยานิพนธ์ฉบับนี้สำเร็จลงไปด้วยดี ขอขอบพระคุณรองศาสตราจารย์ ดร.รัตติกกร ยี่มนิรัฐวงศ์ ที่ให้คำแนะนำ แนวความคิด และข้อเสนอแนะต่างๆ จนเกิดเป็นหัวข้อวิทยานิพนธ์ฉบับนี้ขึ้นมา

ขอขอบคุณระบบลำเลียงแสงที่ 5.2 สถานีทดลองเทคนิคการดูดกลืนรังสีเอกซ์ (X-ray Absorption Spectroscopy) สถานีวิจัยแสงซินโครตรอน (องค์การมหาชน) ที่ให้การสนับสนุนและช่วยเหลือในการทำชุดอุปกรณ์อุปกรณ์ และให้คำปรึกษาปรับแก้แบบอุปกรณ์จนสำเร็จลงด้วยดี ขอขอบคุณภาควิชาฟิสิกส์ คณะวิทยาศาสตร์สำหรับการอำนวยความสะดวกในการใช้เครื่องมือ สถานที่ทำการทดลองจัดทำอุปกรณ์ และเงินสนับสนุนในการซื้อวัสดุอุปกรณ์สำหรับการทำวิจัย และท้ายที่สุดต้องขอขอบพระคุณคุณพ่อ คุณแม่ ผู้คอยให้การสนับสนุน และช่วยเหลือด้านการศึกษา

สุดท้ายนี้ หากการดำเนินการโครงการพิเศษในครั้งนี้ มีข้อบกพร่องประการใด ผู้วิจัยต้องขออภัยไว้ ณ ที่นี้ด้วย

ธรรมกรานต์ พังยะ

ผู้วิจัย

บทคัดย่อ

เรื่อง : การออกแบบและสร้างเตาความร้อนสูงแบบควบคุมและแสดงผลตามเวลาจริง
 ผู้วิจัย : ธรรมกรานต์ พังยะ
 ชื่อปริญญา : วิทยาศาสตร์มหาบัณฑิต
 สาขาวิชา : ฟิสิกส์
 อาจารย์ที่ปรึกษา: ดร.อมร เทศสกุลวงศ์
 คำสำคัญ : เตาความร้อนอุณหภูมิสูง, ระบบควบคุมอุณหภูมิแบบพีไอดี, ระบบแสดงผลตามเวลาจริง, บอร์ดออดิโน

วิทยานิพนธ์นี้นำเสนอการออกแบบและสร้างเตาความร้อนสูงและการปรับค่าตัวแปรของระบบควบคุมอุณหภูมิแบบพีไอดี ซึ่งเตาสามารถเพิ่มอุณหภูมิได้สูงถึง 1000 องศาเซลเซียสโดยการจ่ายกระแสไฟฟ้าให้กับขดลวดความร้อน ระบบควบคุมพีไอดีถูกใช้ในระบบควบคุมอุณหภูมิของเตา เนื่องจากเป็นระบบควบคุมแบบป้อนกลับ และเป็นที่ยอมรับในอุตสาหกรรมควบคุมการทดลองได้ออกแบบเตาความร้อนสูง และการป้องกันการส่งผ่านความร้อน ซึ่งเตาประกอบด้วยขดลวดความร้อน ทำจากปูนทนไฟ ภายในบรรจุด้วยขดลวดความร้อน 2000 วัตต์ ใช้บอร์ดออดิโน mega 2560 สำหรับตรวจวัดและควบคุมอุณหภูมิของเตาความร้อน โดยมีเทอร์โมคัปเปิลและไอซี MAX31850 ถูกนำมาใช้สำหรับวัดอุณหภูมิของเตา ทั้งนี้ระบบควบคุมอุณหภูมิแบบพีไอดี และระบบแสดงผลตามเวลาจริงสามารถออกแบบโดยใช้โปรแกรม LabVIEW และใช้วิธีการของ Ziegler-Nichols และ Trial and error ในการปรับค่าระบบควบคุมแบบพีไอดี

ผลการทดลองสามารถควบคุมอุณหภูมิด้วยตัวแปรควบคุมพีไอดีที่ค่า K_p เท่ากับ 250 T_i เท่ากับ 0.05 และ T_d เท่ากับ 0.20 ในการควบคุมอุณหภูมิในช่วงต่ำกว่า 500 องศาเซลเซียสสามารถควบคุมอุณหภูมิเป้าหมายให้มีค่าความผิดพลาดสูงสุดที่เกิดขึ้นที่ ± 1 องศาเซลเซียส และใช้ตัวแปรควบคุม K_p เท่ากับ 450 T_i เท่ากับ 0 และ T_d เท่ากับ 0 ในการควบคุมอุณหภูมิในช่วงสูงกว่า 500 องศาเซลเซียส โดยสามารถควบคุมอุณหภูมิเป้าหมายให้มีค่าความผิดพลาดสูงสุดที่เกิดขึ้นที่ ± 1 องศาเซลเซียส

ABSTRACT

TITLE : DESIGN AND CONSTRUCTION OF CONTROL AND REAL TIME
MONITORING FOR HIGH TEMPERATURE FURNACE

AUTHOR : TOMMAKRAN PANGYA

DEGREE : MASTER OF SCIENCE

MAJOR : PHYSICS

ADVISOR : AMORN THEDSAKHULWONG, Ph.D.

KEYWORDS : HIGH TEMPERATURE FURNACE, PID CONTROLLER, REAL-TIME
MONITORING, ARDUINO BOARD

The thesis presents design and construction high-temperature furnace and tuning the parameter of the PID temperature control system. The furnace can be operated a temperature up to 1000 Degree celsius using by current flows through a heater coil. PID control system is used as the furnace temperature control system because this is a control loop feedback and has been universally accepted in Industrial control. The experiments firstly design the high temperature furnace and heat transfer shielding. The furnace consists of a block heater made from fireproof mortar inside consist of heater coil 2000 W. An Arduino mega 2560 microcontroller board was used for measuring and control the furnace temperature. A type-K thermocouple and a MAX31850 IC was applied for a furnace temperature measurement. The PID controller and real-time monitoring can be designed by LabVIEW software. Ziegler-Nichols and Trial and error method for tuning PID controller.

The results of experiment can be controlled the Temperature with the PID parameters, K_p was 250 T_i was 0.05 and T_d was 0.20 for control the temperature less than 500-degree Celsius with the maximum error of ± 1 degree Celsius. And the PID parameter K_p was 450 T_i was 0 and T_d was 0 to control the temperature more than 500-degree Celsius with the maximum error of ± 1 degree Celsius.

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
คำอธิบายสัญลักษณ์และอักษรย่อ	ฎ
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	3
1.3 ขอบเขตงานวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	
2.1 หลักการทฤษฎี	4
บทที่ 3 วิธีดำเนินการวิจัย	
3.1 วัสดุอุปกรณ์	33
3.2 การทดลอง	34
บทที่ 4 ผลและวิเคราะห์ผลการวิจัย	
4.1 ผลทดสอบระบบควบคุมอุณหภูมิแบบพีไอดีที่อุณหภูมิต่ำกว่า 500 องศาเซลเซียส	51
4.2 ผลการทดลองหาค่าตัวแปรควบคุมอุณหภูมิในช่วงมากกว่า 500 องศาเซลเซียส	69
4.3 ผลการทดลองวัดอุณหภูมิภายนอก Block heater	71
4.4 ผลการทดลองวัดอุณหภูมิภายนอกเตาความร้อน	72
บทที่ 5 สรุปผลการวิจัย	
5.1 สรุปผลการวิจัย	74
5.2 ข้อเสนอแนะ	77
เอกสารอ้างอิง	78

สารบัญ (ต่อ)

	หน้า
ภาคผนวก	
ก เตาความร้อนที่สร้างสำเร็จเพื่อใช้ในการทดลอง	82
ข โปรแกรมวัดอุณหภูมิด้วยไอซี MAX31850 ใช้งานทาง LIFA_BASE	85
ค เอกสารงานวิจัยตีพิมพ์เผยแพร่	89
ประวัติผู้วิจัย	95

สารบัญตาราง

ตารางที่		หน้า
2.1	ค่าข้อมูลที่อ่านได้จากไอซี MAX31850	12
2.2	ค่าอัตราขยายของตัวควบคุมต่าง ๆ โดย Process reaction method	18
2.3	ค่าอัตราขยายของตัวควบคุมต่าง ๆ โดย Ultimate cycle method	18
2.4	เทอร์โมคัปเปิลชนิดต่าง ๆ ที่นิยมในการใช้งาน	20
2.5	ลักษณะของเส้น Wire แบบต่าง ๆ	26
4.1	ผลการควบคุมอุณหภูมิแบบ P Control	50
4.2	สมการคำนวณตัวแปรควบคุมด้วยวิธีการ Ziegler-Nichol	51
4.3	ผลการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียสโดยใช้ระบบควบคุมแบบสัดส่วน (Proportional control)	53
4.4	ผลการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PID Control	58
4.5	ผลการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PID Control	63

สารบัญภาพ

ภาพที่		หน้า
2.1	การถ่ายเทความร้อนของวัสดุรูปแผ่น	7
2.2	ไอซี MAX31850	9
2.3	Block diagram ของไอซี MAX 31850	9
2.4	PIN Configuration ของไอซี MAX 31850	10
2.5	การต่อวงจรใช้งานไอซี MAX 31850	10
2.6	ROM ขนาด 64 บิต	11
2.7	แสดงวิธีการหาค่า Cyclic redundancy check (CRC)	11
2.8	โพล์ชาร์ตแสดงการทำงานของ ROM	13
2.9	แผนภาพระบบควบคุมแบบเปิด (Open loop control system)	14
2.10	แผนภาพระบบควบคุมแบบเปิด (Close loop control system)	14
2.11	ส่วนประกอบของกราฟจากการควบคุมอุณหภูมิ	16
2.12	การตอบสนองของระบบด้วยวิธีการ Process Reaction Method	17
2.13	โครงสร้างเบื้องต้นของเทอร์โมคัปเปิล	20
2.14	อิฐมวลเบา หรืออิฐฉนวนทนไฟ	22
2.16	อิฐมวลหนัก หรืออิฐหนัก	22
2.16	หน้าตาการใช้งานและออกแบบ Front panel	23
2.17	ตัวอย่างเครื่องมือควบคุมบน Front panel ของ LabVIEW	24
2.18	ตัวอย่างเครื่องมือแสดงผลบน Front panel ของ LabVIEW	24
2.19	ตัวอย่างการออกแบบโปรแกรมในส่วน Block diagram ของ LabVIEW	25
2.20	การใช้งาน Terminal, Node, Function และ Wire บน Block diagram	26
2.21	การออกแบบ Icon และ Connector บน Block diagram ของ LabVIEW	27
2.22	บอร์ดไมโครคอนโทรลเลอร์ Arduino mega 2560 R3	27
2.23	ตัวอย่างการกำหนดค่าตัวแปรเริ่มต้นในส่วน Void setup() ของ Arduino	28
2.24	ตัวอย่างการเขียนโปรแกรมในส่วน Void loop() ของ Arduino	29
2.25	การติดตั้ง LabVIEW Interface For Arduino	30
2.26	หน้าตาการใช้งาน LabVIEW Interface For Arduino	31

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.1	รายละเอียดของงานวิจัย	34
3.2	การเพิ่มฟังก์ชันการติดต่อข้อมูลแบบ One-wire	35
3.3	การเพิ่มฟังก์ชันการอ่านค่าอุณหภูมิ	36
3.4	การตั้งค่าตัวแปรและกำหนดขาอินพุตของข้อมูล	37
3.5	การตั้งค่าการติดต่อสื่อสารข้อมูลกับ ROM ของไอซี MAX31850	37
3.6	การตรวจสอบชนิดของไอซีที่อ่านได้จาก ROM	38
3.7	การอ่านค่าจากหน่วยความจำของไอซี MAX31850	39
3.8	การอ่านค่าอุณหภูมิและแปลงข้อมูลให้อยู่ในรูปของเลขฐานสิบ	40
3.9	Front panel สำหรับแสดงตัวแปรควบคุมและอุณหภูมิ	41
3.10	Block diagram แสดงการออกแบบระบบควบคุมพีไอดี และการติดต่อกับ Arduino	41
3.11	Block diagram ของระบบควบคุมอุณหภูมิแบบพีไอดี ในกรณีที่ค่า setpoint น้อยกว่าหรือเท่ากับ 500 องศาเซลเซียส	44
3.12	Block diagram ของระบบควบคุมอุณหภูมิแบบพีไอดี ในกรณีที่ค่า setpoint มากกว่า 500 องศาเซลเซียส	44
3.13	การเชื่อมต่อ Arduino และ MAX31850 สำหรับวัดและควบคุมอุณหภูมิ	45
3.14	ภาพรวมของอุปกรณ์	46
3.15	ฐานเตาความร้อน	46
3.16	โครงสร้างเตาความร้อน	47
3.17	โครงสร้างส่วนป้องกันความร้อน	48
3.18	การวัดอุณหภูมิภายนอก Block heater	49
3.19	การวัดอุณหภูมิอากาศด้านข้างเตาความร้อน	50
3.20	การวัดอุณหภูมิผิวข้างของเตาความร้อน	50

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.1 ผลการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส ด้วยตัวแปรควบคุมจาก Ziegler-Nichol	52
4.2 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 150	54
4.3 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 150	54
4.4 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 200	55
4.5 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 200	55
4.6 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 250	56
4.7 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 250	56
4.8 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 300	57
4.9 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 300	57
4.10 กราฟการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.05	58
4.11 กราฟขยายการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.05	58
4.12 กราฟการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.10	59
4.13 กราฟขยายการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.10	59
4.14 กราฟการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.15	61
4.15 กราฟขยายการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.15	61
4.16 กราฟการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.20	62
4.17 กราฟขยายการปรับค่าตัวแปรควบคุมอินทิกรัลที่ T_i 0.20	62
4.18 กราฟการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.05	64
4.19 กราฟขยายการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.05	64
4.20 กราฟการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.10	65
4.21 กราฟขยายการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.10	65
4.22 กราฟการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.15	66
4.23 กราฟขยายการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.15	66
4.24 กราฟการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.20	67
4.25 กราฟขยายการปรับค่าตัวแปรควบคุมอนุพันธ์ที่ T_d 0.20	67

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4.26	ผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิต่ำกว่า 500 องศาเซลเซียส	68
4.27	การเปรียบเทียบตัวแปรควบคุม	69
4.28	ผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง	70
4.29	ผลการทดลองการวัดความร้อนภายนอก Block heater ที่จุดต่าง ๆ	71
ก.1	Block heater	83
ก.2	เตาความร้อนเมื่อเพิ่มชั้นผนังอิฐทนไฟเพื่อป้องกันความร้อน	83
ก.3	เตาความร้อนเมื่อเพิ่มชั้นสังกะสีเพื่อป้องกันการสูญเสียความร้อน	84

คำอธิบายสัญลักษณ์ และอักษรย่อ

ความหมาย	สัญลักษณ์	หน่วย SI
ค่าคงที่ระบบควบคุมพีไอดี แบบ สัดส่วน	K_p	-
ค่าคงที่ระบบควบคุมพีไอดี แบบ อินทิกรัล	T_i	-
ค่าคงที่ระบบควบคุมพีไอดี แบบ อนุพันธ์	T_d	-
อุณหภูมิเป้าหมายที่กำหนด (setpoint)	SP	°C
อุณหภูมิของระบบที่ได้จากการควบคุม (Process Variable)	PV	°C
ค่าคงที่พีไอดี แบบสัดส่วนที่ทำให้ระบบเกิด การแกว่งแบบ Oscillate	K_{cr}	-
คาบที่ได้จากกราฟการแกว่งแบบ Oscillate	P_{cr}	-
Laboratory Virtual Instrument Engineering Workbench	LabVIEW	-
LabVIEW Interface For Arduino	LIFA	-

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเตาความร้อน ได้เข้ามาเป็นหนึ่งในอุปกรณ์ที่มีประโยชน์และมีความสำคัญต่องานทางด้านการทดลองทางวิทยาศาสตร์ ยกตัวอย่างเช่นการศึกษาโครงสร้างและคุณสมบัติของสาร ในขณะที่มีการเปลี่ยนแปลงอุณหภูมิของสารตัวอย่างซึ่งอาศัยอุปกรณ์เสริมที่ติดตั้งเข้าไปเพื่อควบคุมอุณหภูมิของสารในขณะที่มีการวัดคุณสมบัติอื่นนอกจากนี้เตาความร้อนสูง ยังถูกนำไปใช้ในกระบวนการหลอมสารตัวอย่างบางชนิด ยกตัวอย่างเช่น การสร้างวัสดุเหล็กเดี่ยวที่ต้องมีกระบวนการหลอมวัสดุ และทำให้วัสดุเกิดการเย็นตัวลงอย่างช้า โดยทำการควบคุมอุณหภูมิของเตาให้มีความแตกต่างของอุณหภูมิสองด้านของเตาความร้อนเพื่อทำให้เกิดการไล่ระดับของอุณหภูมิ [1]

การออกแบบและสร้างเตาความร้อนนั้นมียุคประกอบสำคัญที่ต้องคำนึงถึงอยู่ด้วยกัน 4 องค์ประกอบ ได้แก่

- (1) วิธีการเพิ่มอุณหภูมิของเตาความร้อน
- (2) วิธีการควบคุมอุณหภูมิภายในเตาความร้อน
- (3) วัสดุที่ใช้ในการสร้างเตาความร้อน
- (4) วิธีการป้องกันความร้อนภายนอกเตาความร้อน

สำหรับองค์ประกอบแรกวิธีการเพิ่มอุณหภูมิภายในเตาความร้อนที่ใช้ในงานทดลองทางวิทยาศาสตร์ที่นิยมใช้งานกันโดยมากจะใช้หลักการจ่ายพลังงานไฟฟ้าในขดลวดความร้อนหรือ Heater ซึ่งวิธีการนี้จะทำให้เกิดการเปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานความร้อน [2] ซึ่งวิธีการนี้จะไม่ทำให้เกิดสนามแม่เหล็กมารบกวนอุปกรณ์ภายนอกด้วย

สำหรับองค์ประกอบเรื่องการควบคุมอุณหภูมิภายในเตาความร้อนจะขอนำเสนอวิธีการควบคุมอุณหภูมิด้วยระบบควบคุมแบบพีไอดี เป็นวิธีการควบคุมแบบป้อนกลับ (Feedback control) ที่นิยมใช้ในกระบวนการควบคุมของอุตสาหกรรม เนื่องจากเป็นระบบควบคุมที่ตรวจสอบค่าความผิดพลาดของระบบจากการเปรียบเทียบกับสัญญาณขาออก กับคำสั่งขาเข้าของระบบ ด้วยเหตุนี้จึงทำให้ระบบควบคุมแบบพีไอดี สามารถลดค่าความผิดพลาดของระบบให้เหลือน้อยลงไปได้ ทั้งนี้ระบบควบคุมแบบพีไอดีประกอบด้วยส่วนสำคัญสามส่วนคือ Proportional term, Integral term

และ Derivative term ในแต่ละส่วนจะมีความทำงานแตกต่างกันไป โดยที่ Proportional term จะทำหน้าที่ในการเปรียบเทียบสัญญาณขาออก และ คำสั่งขาเข้าของระบบ และทำการคำนวณสัญญาณขาออก ในส่วน Integral term จะทำการหาค่าความผิดพลาดของระบบจากพื้นที่ที่เกิดจากค่าความผิดพลาด และทำการคำนวณสัญญาณขาออกของระบบ สุดท้าย Derivative term จะทำการคำนวณความชันของระบบ และปรับค่าสัญญาณขาออกให้มีความเสถียรมากขึ้น ระบบควบคุมแบบพีไอดี จะสามารถควบคุมได้ดีมากน้อยเพียงใดนั้นจะขึ้นอยู่กับค่าตัวแปรควบคุมของระบบด้วย ดังนั้นการปรับค่าตัวแปรจึงเป็นส่วนที่สำคัญสำหรับระบบการควบคุมนี้ โดยในงานวิจัยจะนำเสนอการปรับค่าตัวแปรควบคุมแบบ Ziegler-Nichols ผสมกับวิธีการ Trial and error การปรับค่าตัวแปรควบคุมแบบพีไอดีด้วยวิธีการของ Ziegler-Nichols [3] ผสมกับวิธีการ Trial and error [3] คือการปรับค่าที่ใช้สมการควบคุมสัญญาณขาออกที่ได้จากการทดลอง และวิเคราะห์ระบบควบคุมแบบพีไอดี โดย Ziegler และ Nichol และในการปรับค่าตัวแปรในสมการควบคุมของ Ziegler-Nichols นั้นจะใช้วิธีการ Trial and error ซึ่งมีการปรับค่าตัวแปรโดยการทดลอง และวิเคราะห์ผลสามขั้นตอน ขั้นแรกทำการทดลองด้วยระบบควบคุมแบบ P control เพื่อปรับค่าตัวแปรของ P control ให้ได้ค่าที่ดีที่สุด ขั้นตอนที่สอง ทำการทดลองแบบ PI control โดยการใช้ค่าตัวแปร P control จากการทดลองก่อนหน้า และปรับค่าตัวแปร I control ให้ได้ค่าที่ดีที่สุด และขั้นตอนสุดท้ายทำการทดลองแบบ PID control โดยใช้ตัวแปร P และ I control จากการทดลองก่อนหน้านี้ และทำการทดลองเพื่อหาตัวแปร D control ที่ดีที่สุด

องค์ประกอบด้านวัสดุที่ใช้ในการสร้างเตาความร้อนจะต้องเลือกใช้วัสดุที่สามารถใช้งานในช่วงอุณหภูมิที่ต้องการโดยที่ไม่ทำให้วัสดุเกิดการแตกหักและเสียหายเนื่องจากความร้อน โดยจะต้องคำนึงถึงจุดหลอมเหลวของวัสดุ การขยายตัวทางความร้อนของวัสดุ และการทนต่อการเปลี่ยนแปลงอุณหภูมิโดยฉับพลัน เป็นต้น วัสดุที่เลือกใช้ในการสร้างผนังของเตาความร้อนในงานวิจัยนี้ใช้อลูมิเนียมและใช้ปูนทนไฟในการหล่อลือกขัดลดความร้อน เนื่องจากวัสดุทั้งสองชนิดสามารถทนต่อความร้อนช่วงที่ใช้งานได้ และสามารถทนต่อการเปลี่ยนแปลงอุณหภูมิฉับพลันที่เกิดขึ้นได้

องค์ประกอบสุดท้ายคือวิธีการป้องกันความร้อนภายนอกเตาความร้อนนั้นจะมีวิธีการที่ใช้กันอย่างแพร่หลายทั้งในการป้องกันความร้อนของเตาเผา และความร้อนของบ้าน ซึ่งวิธีการที่กล่าวมานี้คือ การใช้ฉนวนป้องกันความร้อน ซึ่งจะสามารถลดการนำความร้อนที่เกิดขึ้นในผนังเตาความร้อนได้ ทั้งนี้สามารถใช้ควบคู่กับวิธีในการลดความร้อนที่เกิดขึ้นด้วยการสร้างผนังกันความร้อนหลายชั้นเพื่อเพิ่มประสิทธิภาพในการลดความร้อนที่ส่งออกมาภายนอกเตาความร้อน

งานวิจัยมีจุดประสงค์เพื่อสร้างเตาความร้อนสูง เพื่อใช้ในงานทดลองทางด้านวิทยาศาสตร์ โดยสามารถใช้งานได้ให้อุณหภูมิสูงถึง 1000 องศาเซลเซียส โดยมีการควบคุมอุณหภูมิของเตาเผาด้วยวิธีการควบคุมแบบพีไอดี และทำการทดลองปรับค่าตัวแปรควบคุมอุณหภูมิแบบพีไอดีด้วยวิธีการ

Ziegler-Nichols ผสมกับวิธีการ Trial and error ตัวอุปกรณ์สามารถแสดงผลข้อมูลตามเวลาจริงผ่านทางระบบคอมพิวเตอร์ และสามารถลดความร้อนที่เกิดจากการถ่ายเทความร้อนภายในเตาได้โดยการสร้างผนังกันความร้อนสองชั้นและแทรกด้วย ฉนวนกันความร้อนระหว่างชั้นของผนังกันความร้อน

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาและออกแบบเตาความร้อนสูงที่สามารถนำไปใช้เป็นอุปกรณ์เสริมในงานทดลองเชิงวิทยาศาสตร์

1.2.2 เพื่อศึกษาและทดลองการปรับค่าตัวแปรควบคุมพีไอดี ด้วยเทคนิค Ziegler-Nichols ผสมกับวิธีการ Trial and error

1.3 ขอบเขตงานวิจัย

1.3.1 ชุดอุปกรณ์เตาความร้อนที่สร้างขึ้นสามารถเพิ่มอุณหภูมิและควบคุมอุณหภูมิในช่วง 50 – 1000 องศาเซลเซียส

1.3.2 เตาความร้อนใช้วิธีการให้ความร้อนจากการจ่ายแรงดันไฟฟ้าให้กับขดลวดความต้านทาน

1.3.3 เตาความร้อนใช้วิธีการควบคุมอุณหภูมิด้วยระบบควบคุมแบบพีไอดี (PID controller)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้รับอุปกรณ์เตาความร้อนสูงที่สามารถใช้ในงานทดลองทางด้านวิทยาศาสตร์

1.4.2 เข้าใจหลักการออกแบบระบบควบคุมแบบพีไอดี และสามารถปรับค่าตัวแปรควบคุมให้สามารถควบคุมอุณหภูมิให้ตรงตามความต้องการในการใช้งาน

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 หลักการทฤษฎี

2.1.1 ผลของความร้อนที่มีต่อวัสดุ [4]

เมื่อวัสดุได้รับพลังงานความร้อนในค่าหนึ่ง ๆ วัสดุจะเกิดการเปลี่ยนแปลงโครงสร้างและคุณสมบัติ ตามลักษณะต่าง ๆ ได้แก่ เกิดการขยายตัวทางความร้อนและเกิดการเปลี่ยนสถานะ เป็นต้น

2.1.1.1 การขยายตัวทางความร้อน เมื่อวัสดุได้รับความร้อนอะตอมของวัสดุจะเกิดการสั่นไหวรอบจุดหนึ่ง ๆ ในโครงสร้างของวัสดุเป็นเหตุให้เกิดระยะห่างระหว่างอะตอมขึ้นและทำให้วัสดุเกิดการขยายตัวทางความร้อนโดยส่วนมากถ้าหากเราพิจารณาวัสดุที่เป็นของแข็งวัสดุจะมีการขยายตัวในทุกทิศทางซึ่งเป็นเหตุให้รูปทรงของวัสดุมีรูปร่างเหมือนเดิมแต่ขนาดวัสดุจะมีการขยายใหญ่ขึ้นซึ่งวัสดุจะขยายตัวได้มากหรือขยายตัวได้น้อยจะขึ้นอยู่กับชนิดของแข็งนั้น ๆ โดยสามารถพิจารณาการขยายตัวของของแข็งได้เป็น 3 แบบ ได้แก่

1) การขยายตัวเชิงเส้นของวัสดุ วัสดุเมื่อได้รับความร้อนจะมีการเปลี่ยนแปลงความยาว โดยความยาวที่เพิ่มขึ้น Δl จะเป็นสัดส่วนโดยตรงกับความยาวเดิม l_0 และเป็นสัดส่วนโดยตรงกับอุณหภูมิที่เพิ่มขึ้น ΔT การขยายตัวเชิงเส้นของวัสดุแสดงได้ดังสมการที่ 2.1

$$\Delta l = \alpha l_0 \Delta T \quad (2.1)$$

เมื่อ α คือ ค่าคงที่เฉพาะของสารแต่ละชนิด เรียกว่า สัมประสิทธิ์การขยายตัวตามเส้นของวัสดุกำหนดให้ L คือ ความยาวของวัสดุที่เปลี่ยนไปหลังการขยายตัวเชิงเส้น เราจะได้สมการความสัมพันธ์ระหว่างความยาวของวัสดุที่เปลี่ยนไปต่อความยาวของวัสดุเดิมแสดงได้ดังสมการที่ 2.3

$$l = l_0(1 + \alpha \Delta T) \quad (2.2)$$

2) การขยายตัวเชิงพื้นที่ของวัสดุ วัสดุเมื่อได้รับความร้อนจะมีการเปลี่ยนแปลงพื้นที่โดยพื้นที่ที่เพิ่มขึ้นของวัสดุ A จะเป็นสัดส่วนโดยตรงกับพื้นที่เดิมของวัสดุ A_0 และเป็นสัดส่วน

โดยตรงกับอุณหภูมิที่เพิ่มขึ้น ΔT เมื่อ β คือ สัมประสิทธิ์ของการขยายตัวตามพื้นที่ของวัสดุการขยายตัวเชิงพื้นที่แสดงได้ดังสมการที่ 2.3

$$A = A_0(1 + \beta\Delta T) \quad (2.3)$$

3) การขยายตัวเชิงปริมาตร วัสดุเมื่อได้รับพลังงานความร้อนเข้ามาจะมีการเปลี่ยนแปลงปริมาตร โดยปริมาตรที่เพิ่มขึ้น V จะเป็นสัดส่วนโดยตรงกับปริมาตรเดิมของวัสดุ V_0 และเป็นสัดส่วนโดยตรงกับอุณหภูมิที่เพิ่มขึ้น ΔT การขยายตัวเชิงปริมาตรแสดงได้ดังสมการที่ 2.4

$$V = V_0(1 + \gamma\Delta T) \quad (2.4)$$

เมื่อ γ คือ สัมประสิทธิ์ของการขยายตัวตามปริมาตรของวัสดุ

ความสัมพันธ์ระหว่างสัมประสิทธิ์การขยายตัวตามเส้น สัมประสิทธิ์การขยายตัวตามพื้นที่และสัมประสิทธิ์การขยายตัวตามปริมาตรของวัสดุมีความสัมพันธ์กัน เนื่องจากพื้นที่สามารถคำนวณได้จากการคูณความยาวของด้านสองด้าน จะได้สมการแสดงความสัมพันธ์ดังสมการที่ 2.5

$$\beta = 2\alpha \quad (2.5)$$

ความสัมพันธ์ระหว่างสัมประสิทธิ์การขยายตัวตามปริมาตร และสัมประสิทธิ์การขยายตัวตามเส้น สามารถหาได้เนื่องจากการหาปริมาตรได้จากการคูณกันของความกว้าง ความยาว และความสูงซึ่งได้ได้สมการแสดงความสัมพันธ์ดังสมการที่ 2.6

$$\gamma = 3\alpha \quad (2.6)$$

ความสัมพันธ์ระหว่างสัมประสิทธิ์การขยายตัวตามปริมาตร และ สัมประสิทธิ์การขยายตัวตามพื้นที่ สามารถหาได้จากสมการแสดงความสัมพันธ์ดังสมการที่ 2.7

$$\gamma = \frac{3}{2}\beta \quad (2.7)$$

2.1.1.2 การเปลี่ยนสถานะของวัสดุเนื่องจากความร้อน วัสดุในธรรมชาติที่อยู่ในสถานะใดสถานะหนึ่งสามารถเปลี่ยนแปลงจากสถานะหนึ่งไปอีกสถานะหนึ่งได้ซึ่งเป็นการเปลี่ยนแปลงทางกายภาพเรียกว่า การเปลี่ยนสถานะของวัสดุ ซึ่งสามารถแยกออกได้เป็น 2 ประเภท

ประเภทแรกคือ การเปลี่ยนแปลงสถานะแบบดูดความร้อน ซึ่งจะเปลี่ยนสถานะของสารจาก ของแข็ง เปลี่ยนเป็นของเหลวและจากของเหลวเปลี่ยนเป็นก๊าซ ตามลำดับ ประเภทต่อมาคือ การเปลี่ยนแปลงสถานะแบบคายความร้อนเป็นการเปลี่ยนแปลงสถานะจากสถานะก๊าซเปลี่ยนเป็นของเหลว และจากของเหลวเปลี่ยนเป็นของแข็ง ตามลำดับ

การคำนวณหาปริมาณความร้อนที่ทำให้วัสดุเปลี่ยนสถานะ เมื่อมีการเปลี่ยนแปลงอุณหภูมิของวัสดุ เมื่อกำหนดให้ Q คือ ปริมาณความร้อนที่สารได้รับ m คือ มวลของวัสดุ c คือ ค่าความจุความร้อนของสาร และ ΔT คืออุณหภูมิของสารที่เปลี่ยนแปลง จะสามารถคำนวณได้จากสมการที่ 2.8

$$Q = mc\Delta T \quad (2.8)$$

2.1.2 การส่งผ่านความร้อน [5]

การส่งผ่านความร้อนในเตาความร้อนมี 3 วิธี ได้แก่ การนำความร้อน การพาความร้อน และการแผ่รังสีความร้อน

2.1.2.1 การนำความร้อน (Heat conduction) เมื่ออุณหภูมิของจุดสองจุดบนเนื้อวัสดุมีค่าต่างกัน พลังงานความร้อนจะเคลื่อนที่จากจุดที่มีอุณหภูมิสูงไปยังจุดที่มีอุณหภูมิต่ำ เรียกการส่งผ่านความร้อนในลักษณะนี้ว่าการนำความร้อน โดยปริมาณความร้อนที่ส่งผ่านจากจุดที่ร้อนกว่าสู่จุดที่เย็นเมื่อเวลาผ่านไปจะแปรผันตรงกับพื้นที่หน้าตัดของวัสดุ ความแตกต่างของอุณหภูมิทั้งสองด้านของวัสดุและแปรผกผันกับความหนาของวัสดุซึ่งสามารถแสดงความสัมพันธ์ของการนำความร้อนในรูปของสมการกฎของฟูเรียร์ (Fourier's Law)

$$\frac{\Delta Q}{\Delta t} = -kA \frac{\Delta T}{\Delta x} \quad (2.9)$$

ΔQ คือ ปริมาณความร้อน

A คือ พื้นที่หน้าตัดของวัสดุ

k คือ สภาพนำความร้อน (Thermal conductivity)

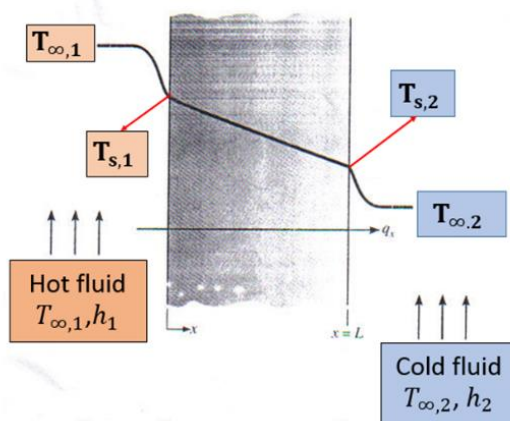
วัสดุรูปแผ่นที่มีอุณหภูมิทั้งสองด้านไม่เท่ากัน ในที่นี้จะพิจารณาถึงผลต่างของอุณหภูมิตามความหนาของวัสดุรูปแผ่น ($T_2 - T_1$) ที่ความหนาของวัสดุ (l) และสภาพการนำความร้อน (KA) จะสามารถหาค่าการนำความร้อนของวัสดุรูปแผ่นได้จากสมการที่ 2.10

$$Q = -KA \frac{(T_2 - T_1)}{l} \quad (2.10)$$

เมื่อพิจารณาตามตัวแปรความต้านทานความร้อนของวัสดุ จะสามารถแสดงความสัมพันธ์ตามสมการที่ 2.11

$$Q = \frac{(T_2 - T_1)}{R_c} \quad (2.11)$$

โดยที่ R_c คือ สภาพการต้านทานความร้อน มีค่าเท่ากับ $\frac{l}{KA}$



ภาพที่ 2.1 การถ่ายเทความร้อนของวัสดุรูปแผ่น

เมื่อนำวัสดุการนำความร้อนที่มีสัมประสิทธิ์การนำความร้อนที่แตกต่างกันมาประกอบเป็นแผ่นหลายชั้น และมีการกระจายของอุณหภูมิที่พื้นผิวอย่างสม่ำเสมอ การนำความร้อนของวัสดุเมื่อคิดจากสภาพการต้านทานความร้อน จะสามารถคำนวณได้จากสมการ 2.12

$$Q = \frac{(T_1 - T_{n+1})}{R_c} \quad (2.12)$$

2.1.2.2 การพาความร้อน การส่งผ่านความร้อนโดยอาศัยโมเลกุลของตัวกลางที่ได้รับความร้อนเป็นตัวเคลื่อนที่พาความร้อนไป เรียกการส่งผ่านความร้อนในลักษณะนี้ว่า การพาความร้อน โดยตัวกลางในการพาความร้อนที่ดีมักจะเป็นของเหลวและก๊าซ เนื่องจากโมเลกุลของของแข็งเคลื่อนไหวได้ยาก ในขณะที่โมเลกุลของของเหลว และโมเลกุลของก๊าซจะมีการเคลื่อนที่ได้ง่าย

ทำให้เกิดการหมุนเวียนของความร้อนเกิดขึ้น อัตราการส่งผ่านความร้อนโดยการพาความร้อนสามารถคำนวณได้จากสมการที่ 2.13

$$\frac{\Delta Q}{\Delta t} = hA(T_2 - T_1) = k(T_2 - T_1) \quad (2.13)$$

$\frac{\Delta Q}{\Delta t}$ คือ อัตราการส่งผ่านความร้อนโดยการพาความร้อน

h คือ สัมประสิทธิ์การพาความร้อนของตัวกลาง

A คือ พื้นที่ผิวตัวกลางที่พิจารณา

$(T_2 - T_1)$ คือ ผลต่างของอุณหภูมิที่พิจารณาทั้งสองจุด

k คือ สัมประสิทธิ์ของการเยื้องตัว

2.1.2.3 การแผ่รังสีความร้อน การแผ่รังสีความร้อนคือการส่งผ่านความร้อนจากผิวของวัสดุด้วยพลังงานความร้อนในรูปของคลื่นแม่เหล็กไฟฟ้าโดยอัตราการแผ่รังสีของวัตถุใด ๆ จะแปรผกผันกับพื้นที่ผิวของวัสดุ (A) และกำลังสี่ของอุณหภูมิสัมบูรณ์ของวัสดุนั้น ตามกฎของสเตฟาน (Stefan's Law) ซึ่งเขียนเป็นสมการได้ดังสมการที่ 2.14

$$R = \frac{\Delta Q}{\Delta T} = \sigma A e T^4 \quad (2.14)$$

R คือ อัตราการแผ่รังสีของวัตถุ

e คือ สภาพการส่งรังสีของผิววัตถุ

σ คือ ค่าคงที่ของสเตฟาน

T คือ อุณหภูมิของวัตถุ

2.1.3 ไอซี MAX31850 [6]

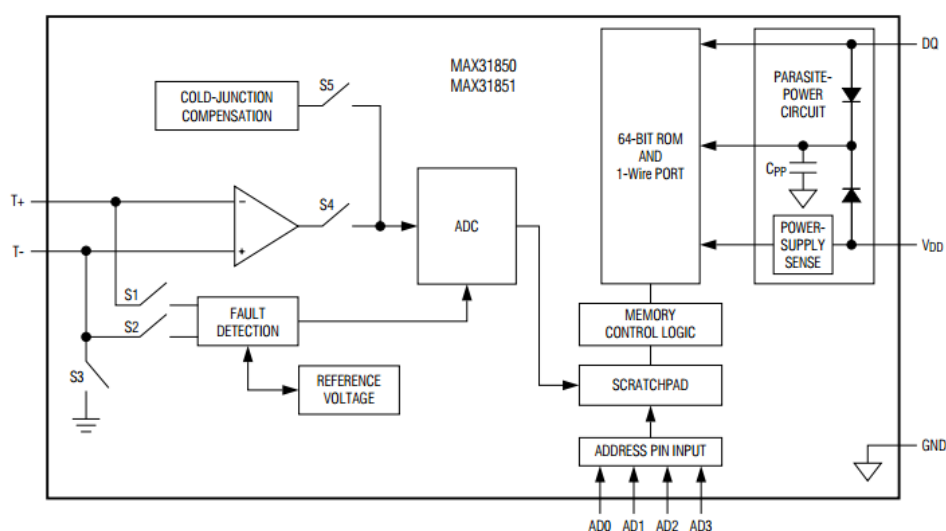
ไอซี MAX31850 เป็นไอซีที่ใช้งานในการวัดค่าอุณหภูมิแบบ Digital thermometer ซึ่งมีการติดต่อสื่อสารแบบ 1-wire ทำให้สามารถต่อพ่วงอุปกรณ์ได้ในบัสเส้นเดียว ทำให้ง่ายต่อการใช้งานในกรณีที่ต้องการวัดอุณหภูมิหลายจุด แสดงภาพไอซี MAX31850 ได้ดังภาพที่ 2.2



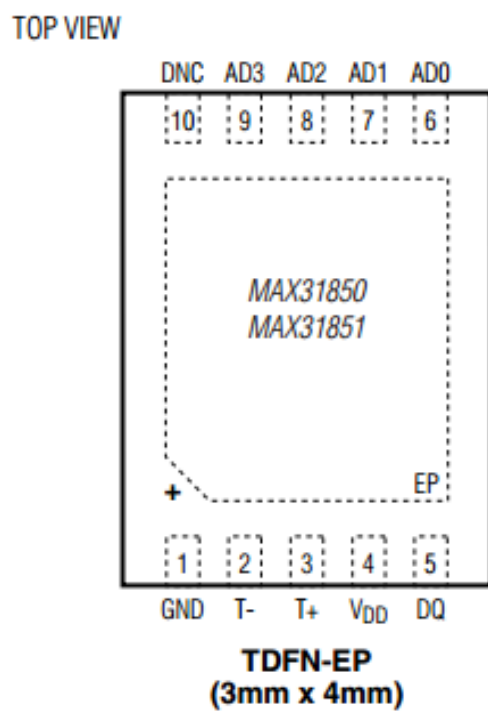
ภาพที่ 2.2 ไอซี MAX31850

2.1.3.1 คุณสมบัติโดยทั่วไปของไอซี MAX31850 [6]

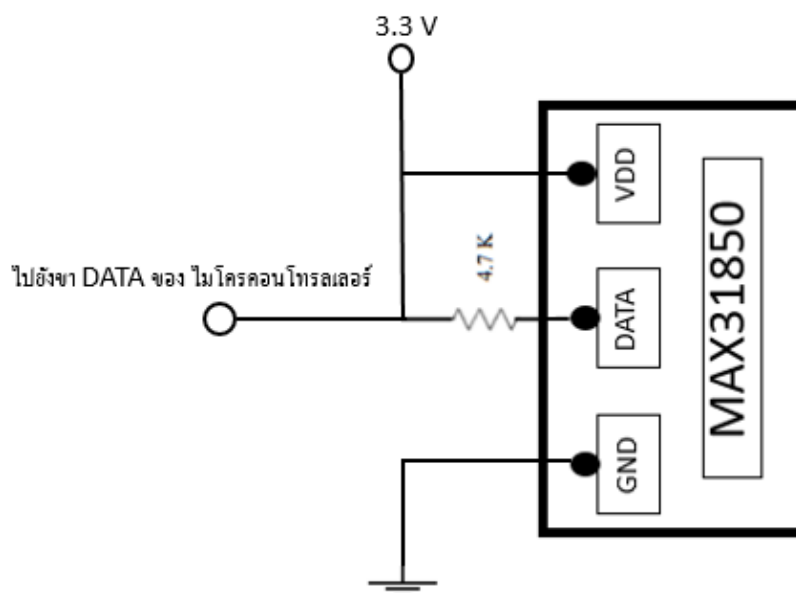
- 1) MAX31850 เป็นดิจิทัลเทอร์โมมิเตอร์ที่ใช้การอินเตอร์เฟสแบบ One-Wire โดยสามารถใช้งานได้สองแบบคือแบบ Normal mode (ใช้งานสามขา) และแบบ Parasite power mode ใช้แรงดัน 3.0 – 3.7 V
- 2) การใช้งานจะมีสายสัญญาณเพียง 2 เส้น คือ สาย GND และสายสัญญาณ DATA โดยสาย DATA จะจัดการเกี่ยวกับสัญญาณข้อมูลและสัญญาณนาฬิกาและจะเป็นแบบ Open Drain ดังนั้นในการใช้งานขา DATA จะต่อเข้ากับตัวต้านทาน 4.7 K Ω
- 3) ความละเอียด 0.25 องศาเซลเซียส และย่านการวัดอุณหภูมิ -270 ถึง 1768 องศาเซลเซียส



ภาพที่ 2.3 Block diagram ของไอซี MAX31850 [6]

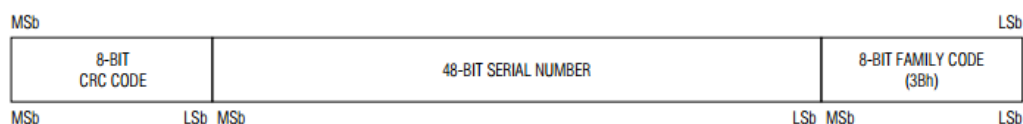


ภาพที่ 2.4 PIN Configuration ของไอซี MAX31850 [6]



ภาพที่ 2.5 การต่อวงจรใช้งานไอซี MAX31850 [6]

2.1.3.2 ROM ขนาด 64 บิต [6]



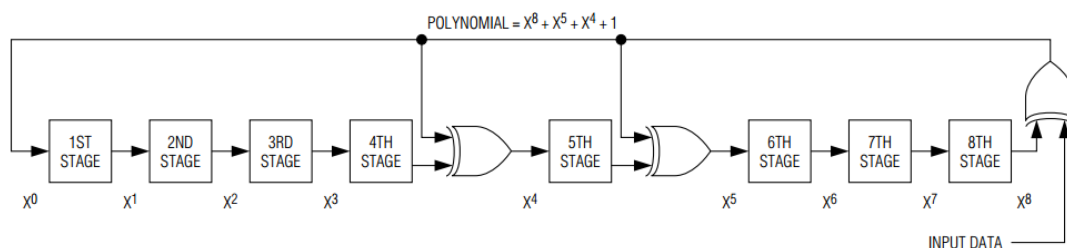
ภาพที่ 2.6 ROM ขนาด 64 บิต [6]

ไอซี Max31850 จะมีรหัสประจำตัวแต่ละไอซีเก็บไว้ใน ROM 64 บิต โดยที่ 8 บิตแรกจะเป็นรหัสของตระกูลย่อย (FAMILY CODE) รหัสตระกูลของ Max31850 คือ 3Bh ส่วนต่อมา 48 บิต จะเป็นรหัสประจำตัวของไอซีแต่ละตัว (SERIAL NUMBER) ซึ่งไอซีแต่ละตัวจะมีรหัสไม่เหมือนกัน ทำให้สามารถต่อไอซีหลายตัวเข้ากับบัส One-Wire ในชุดเดียวกันได้ จะทำให้ทราบว่าข้อมูลอุณหภูมินี้มาจากไอซีตัวใด เมื่อมีการอ่านค่าอุณหภูมิที่มาจากหลายไอซีในบัสชุดเดียว และ 8 บิตสุดท้ายคือ Cyclic redundancy check (CRC)

2.1.3.3 Cyclic redundancy check (CRC) [6] เป็นค่าที่ไอซีสร้างขึ้นมาเพื่อตรวจสอบข้อผิดพลาดของข้อมูลที่อ่านได้จาก ROM ด้วยการเปรียบเทียบค่า CRC ที่คำนวณมาจาก 56 บิตแรกของ ROM กับค่า CRC ที่คำนวณมาจากข้อมูลที่อ่านออกมาได้ โดยเมื่อค่า CRC ทั้งสองมีค่าตรงกัน แสดงว่าข้อมูลที่อ่านออกมามีความถูกต้อง การคำนวณค่า CRC ด้วย Polynomial function ดังนี้

$$CRC = X^8 + X^5 + X^4 + 1 \quad (2.15)$$

BUS master สามารถคำนวณค่า CRC ใหม่ และเปรียบเทียบกับค่า CRC จากอุปกรณ์ โดยการ XOR และเลื่อนข้อมูลด้วยวงจรดังภาพที่ 2.5 โดยวงจรประกอบด้วย Shift register และ XOR gates และเริ่มต้นเลื่อนบิตจากบิตแรก (1ST STAGE) ไปยังบิต 0 (8TH STAGE)



ภาพที่ 2.7 วิธีการหาค่า Cyclic redundancy check (CRC) [6]

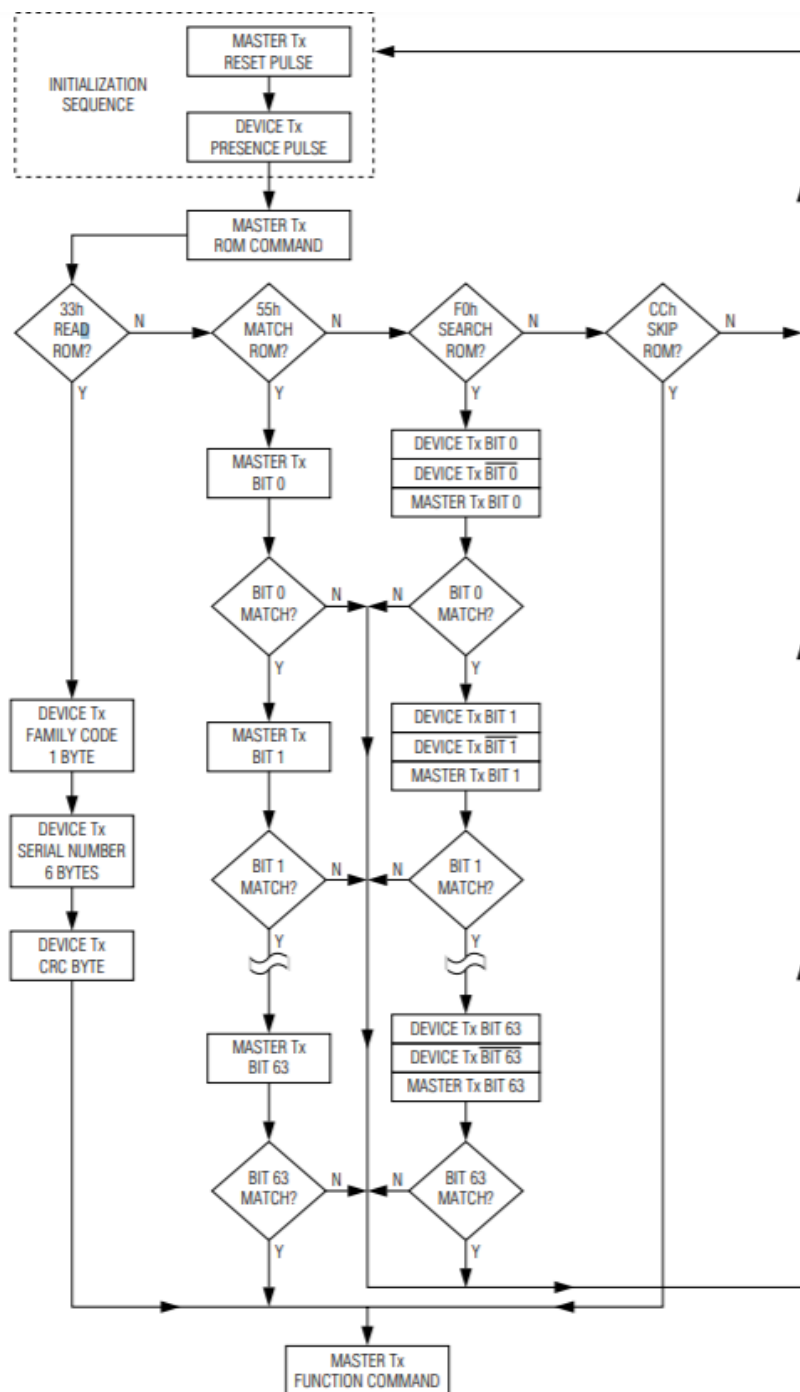
ค่าที่อ่านได้จากไอซี MAX31850 แบ่งเป็นสองค่าคือ อุณหภูมิจาก Thermo couple จาก ไบต์ 0 และ 1 และอุณหภูมิเปรียบเทียบภายใน ที่ไบต์ 2 และ 3 เป็นข้อมูลเลขฐาน 2 ดังแสดงในตาราง โดยเลข 1 ซ้ายมือสุดบอกถึงอุณหภูมิมีค่าเป็นลบ

2.1.3.4 การอ่านค่าอุณหภูมิที่ได้จาก MAX31850 [6] ข้อมูลที่อ่านเข้ามาจากไอซี MAX31850 เป็นเลขฐานสอง โดยข้อมูลที่รับมาจะถูกนำมาคำนวณในฟังก์ชันการวัดค่าอุณหภูมิของโปรแกรม Arduino ก่อนจะส่งข้อมูลอุณหภูมิไปแสดงผลยังโปรแกรม LabVIEW ค่าข้อมูลที่อ่านได้จากไอซี MAX31850 แสดงข้อมูลดังตารางที่ 2.1

ตารางที่ 2.1 ค่าข้อมูลที่อ่านได้จากไอซี MAX31850

Temperature (°c)	Digital Output
1600	0110 0100 0000 00
1000	0011 1110 1000 00
100.75	0000 0110 0100 11
25.00	0000 0001 1001 00
0.00	0000 0000 0000 00
-0.25	1111 1111 1111 11
-1.00	1111 1111 1111 00
-250.00	1111 0000 0110 00

2.1.3.5 การทำงานเกี่ยวกับ ROM [6] เริ่มต้นการทำงานเกี่ยวกับ ROM โดยมาสเตอร์ส่งสัญญาณพัลส์ เพื่อรีเซ็ต MAX31850 และสามารถเรียกใช้งาน ROM ได้โดยการส่งการไปที่ MAX31850 และมาสเตอร์จะรอรับข้อมูลที่ MAX31850 ส่งออกมา ตัวอย่างการใช้งานเกี่ยวกับ ROM เช่น เมื่อต้องการอ่านรหัสตระกูล Serial number และ CRC ของไอซี สามารถทำได้โดยการส่งคำสั่ง 33h ไปที่ MAX31850 หลังจากนั้นมาสเตอร์จะรอรับข้อมูลที่ MAX31850 ส่งออกมา โพล์ชาร์ตแสดงการทำงานเกี่ยวกับ ROM และ หน่วยความจำของไอซี MAX31850 แสดงดังภาพที่ 2.8



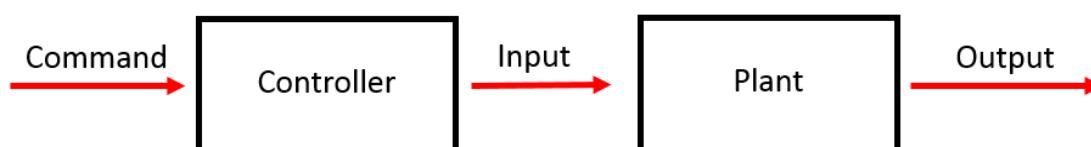
ภาพที่ 2.8 โฟลว์ชาร์ตแสดงการทำงานของ ROM [6]

2.1.3 ระบบควบคุม [7]

ระบบควบคุม คือกระบวนการที่ใช้ในการควบคุมระบบพลวัต (*dynamic system*) หรือ กระบวนการ (*process*) ให้ได้การตอบสนอง (*response*) ตามความต้องการของผู้ใช้งาน

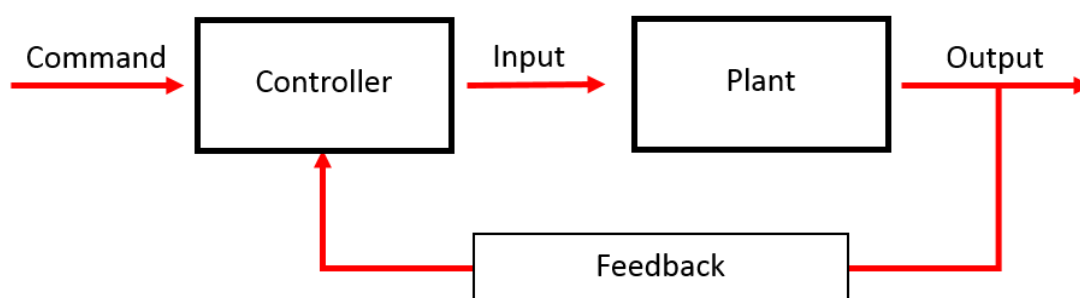
เช่นการควบคุมอุณหภูมิของเตาเผา จะมีกระบวนการที่ใช้ในการควบคุมอุณหภูมิเพื่อปรับและรักษาความร้อนให้อยู่ในช่วงที่ผู้ใช้งานต้องการ ระบบควบคุมที่ใช้ในการควบคุมกระบวนการมีอยู่หลายชนิดซึ่งสามารถเลือกใช้ได้ตามความเหมาะสมกับการทำงาน สามารถแบ่งชนิดของระบบควบคุมหลักได้ 2 ชนิด ได้แก่ ระบบควบคุมแบบเปิด (*Open loop control system*) และระบบควบคุมแบบปิด (*Close loop control system*) มีรายละเอียดดังหัวข้อย่อยต่อไปนี้

2.1.4.1 ระบบควบคุมแบบเปิด (*Open loop control system*) เป็นระบบควบคุมที่ไม่นำเอาเอาต์พุตของสัญญาณมาทำการเปรียบเทียบกับความผิดพลาดกับสัญญาณอินพุตของระบบ ทำให้ง่ายต่อการสร้างระบบควบคุม แต่มีข้อเสียที่ระบบควบคุมนี้จะเกิดความผิดพลาดได้ง่ายมาก ดังแสดงในภาพที่ 2.9



ภาพที่ 2.9 แผนภาพระบบควบคุมแบบเปิด (*Open loop control system*)

2.1.4.2 ระบบควบคุมแบบปิด (*Close loop control system*) เป็นระบบควบคุมแบบป้อนกลับ ที่นำเอาเอาต์พุตของสัญญาณมาเปรียบเทียบกับค่าอินพุตของระบบเพื่อหาความแตกต่างที่เกิดขึ้น การควบคุมวิธีนี้จะทำให้สัญญาณเอาต์พุตมีค่าใกล้เคียงกับความต้องการใช้งาน แสดงแผนภาพระบบควบคุมแบบปิดดังภาพที่ 2.10



ภาพที่ 2.10 แผนภาพระบบควบคุมแบบปิด (*Close loop control system*)

2.1.5 ระบบควบคุมแบบพีไอดี [7]

การควบคุมแบบพีไอดี (Proportional Integral Derivative Controller) เป็นระบบควบคุมแบบปิด (Close loop control system) ชนิดหนึ่งที่มีหลักการทำงานคือการเปรียบเทียบสัญญาณอินพุตและเอาต์พุตและทำการคำนวณด้วยสมการในฟังก์ชันควบคุมเพื่อลดความผิดพลาดให้เหลือน้อยที่สุด ก่อนจะส่งสัญญาณเพื่อไปควบคุมระบบภายนอก ระบบควบคุมแบบพีไอดีประกอบไปด้วยเทคนิคการควบคุม 3 เทอมเข้ามาประกอบกัน ซึ่งประกอบด้วย ระบบควบคุมแบบสัดส่วน (Proportional term) ระบบควบคุมอินทิกรัล (Integral term) และระบบควบคุมแบบอนุพันธ์ (Derivative term) ทำให้ในระบบควบคุมแบบพีไอดี มีตัวแปรที่ใช้ในการควบคุม 3 ตัวแปร คือค่าอัตราขยายแบบสัดส่วน K_p ค่าคงที่ของ Integral K_i และ ค่าคงที่ของ Derivative K_d และสามารถหาสัญญาณควบคุมได้จากสมการที่ 2.16

$$Y(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} e(t) \quad (2.16)$$

โดยที่ค่า K_p , K_i และ K_d เป็นค่าคงที่ของ Proportional Integral และ Derivative ตามลำดับโดยค่าคงที่ดังกล่าวมีค่าขึ้นอยู่กับระบบควบคุม

2.1.5.1 Proportional term Integral term Derivative term ของระบบควบคุมแบบพีไอดี มีผลต่อเอาต์พุตของสัญญาณ ดังนี้

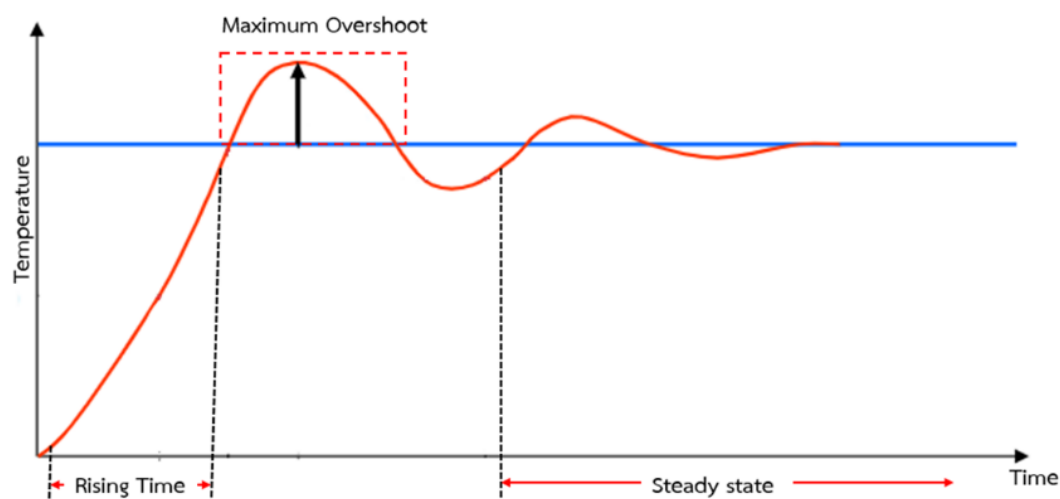
1) Proportional term เป็นส่วนที่เพิ่มความเร็วในการตอบสนองของระบบ ทำให้สามารถลดข้อผิดพลาดของระบบเมื่อทำการควบคุมในช่วงสภาวะอยู่ตัวของระบบ เหมาะสำหรับใช้งานในกระบวนการที่ต้องการผลตอบสนองที่รวดเร็ว และสามารถยอมรับค่าความคลาดเคลื่อนที่เกิดขึ้นได้ในระดับหนึ่ง

2) Integral term เป็นส่วนที่ถูกเพิ่มเข้ามาเพื่อลดค่าความผิดพลาดในสภาวะอยู่ตัวของระบบให้น้อยลงจนหมดไป แต่จะมีผลทำให้ค่าการตอบสนองของระบบลดลง ทำให้ช่วงเวลาการแกว่งของระบบยาวนานขึ้น และทำให้เสถียรภาพของระบบลดลงด้วย

3) Derivative term เป็นส่วนที่ถูกเพิ่มเข้ามาในระบบเพื่อทำให้ระบบมีเสถียรภาพมากขึ้น และเพิ่มค่าอัตราหน่วงให้แก่ระบบ ซึ่งจะช่วยลดค่าความคลาดเคลื่อนที่เกิดขึ้นจากการใช้งานส่วนควบคุมของทั้ง Proportional term และ Integral term ผลที่เกิดขึ้นคือระบบจะคำนวณแนวโน้มของความคลาดเคลื่อนช่วยลดค่าความผิดพลาดของระบบที่เกิดขึ้น และทำให้ระบบมีผลการตอบสนองที่เร็วขึ้น

2.1.5.2 ส่วนประกอบของกราฟจากการควบคุมอุณหภูมิ ในการพิจารณารูปภาพการควบคุมอุณหภูมิ ประการแรกนั้นต้องทำความรู้จักกับส่วนประกอบที่ได้จากกราฟ

ระบบควบคุมอุณหภูมิเสียก่อน เพื่อให้ง่ายต่อการเข้าใจการวิเคราะห์ข้อมูลอุณหภูมิในส่วนต่อไป โดยส่วนประกอบต่าง ๆ ที่ควรรู้จักนั้นจะแสดงตามภาพที่ 2.11



ภาพที่ 2.11 ส่วนประกอบของกราฟจากการควบคุมอุณหภูมิ

จากภาพที่ 2.11 ส่วนประกอบของกราฟจากการควบคุมอุณหภูมิ ตัวแปรที่ต้องทำความเข้าใจก่อนจะเข้าสู่หัวข้อถัดไปมีดังต่อไปนี้

- 1) Setpoint คือ ค่าอุณหภูมิเป้าหมายที่ต้องการควบคุม
- 2) Process Variable คือ ค่าอุณหภูมิที่ได้จากระบบควบคุม
- 3) Rising time คือ ช่วงเวลาที่ระบบควบคุมอุณหภูมิใช้ไปในการเพิ่มอุณหภูมิเป้าหมายจนถึงค่าอุณหภูมิที่ต้องการ (setpoint)
- 4) Maximum overshoot คือ ค่าอุณหภูมิในคลื่นลูกแรกที่ได้จากการควบคุมอุณหภูมิซึ่งปกติแล้วอุณหภูมิที่ถูกควบคุมในคลื่นลูกแรกนั้นมักจะมีผลผิดพลาดที่เกิดจากการควบคุมเกิดขึ้นในลักษณะเป็นภูเขาสูงขึ้นไปดังแสดงตามภาพที่ 2.11
- 5) Steady state คือ สภาวะอยู่ตัวของระบบ หรือ จุดที่ระบบเริ่มเข้าสู่สภาวะสมดุลของระบบ
- 6) Steady state error คือ ค่าความผิดพลาดจากการควบคุมอุณหภูมิของระบบที่อยู่ในช่วงสภาวะอยู่ตัวของระบบ (Steady state)

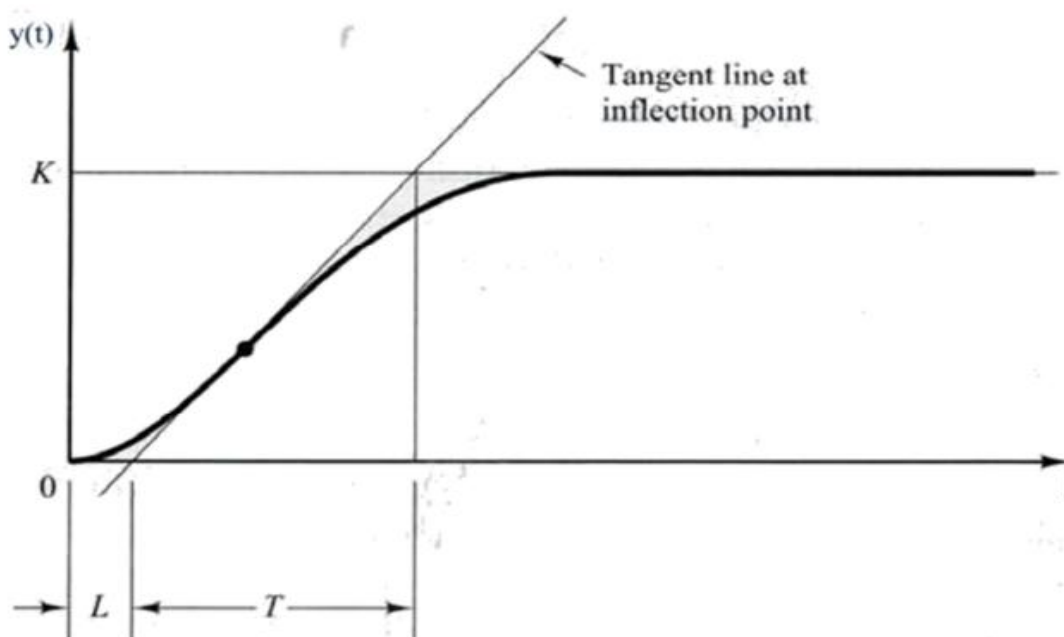
2.1.6 การออกแบบการปรับค่าตัวควบคุมแบบพีไอดี [7]

การออกแบบการปรับค่าตัวควบคุมแบบพีไอดีคือ กระบวนการทดลอง คำนวณ เพื่อหาค่าพารามิเตอร์ที่ดีที่สุดของตัวควบคุมแบบพีไอดี ซึ่งวิธีที่ใช้ในการหาค่าพารามิเตอร์ดังกล่าว ที่นิยมใช้กันมีอยู่ 3 วิธี คือ

2.1.6.1 วิธีการของ Ziegler – Nichols Ziegler ได้อาศัยการทดลองและวิเคราะห์เปรียบเทียบ เพื่อหาวิธีการปรับค่าพารามิเตอร์ของระบบควบคุมแบบพีไอดี โดยมีวิธีในการปรับค่าพารามิเตอร์ของ Ziegler – Nichols มี 2 วิธี คือ Process Reaction Method) และวิธีการ Ultimate Cycle Method สมการที่ใช้ในการคำนวณสำหรับวิธีการของ Ziegler – Nichols แสดงดังสมการที่ 2.17

$$G_c(s) = K_p(1 + \frac{1}{T_i} + T_d) \quad (2.17)$$

วิธีการ Process Reaction Method เป็นการปรับค่าพารามิเตอร์ แบบ Open loop โดยเริ่มทำการทดลองแล้วทำการวัดค่าตัวแปรต่างๆ จากการตอบสนองของกระบวนการ จากนั้นนำกราฟที่ได้จากกระบวนการมาเลือกเส้นสัมผัสที่มีความชันสูงสุดแล้วทำการคำนวณค่าพารามิเตอร์ตามตารางที่ 2.2



ภาพที่ 2.12 การตอบสนองของระบบด้วยวิธีการ Process Reaction [7]

ตารางที่ 2.2 ค่าอัตราการขยายของตัวควบคุมต่าง ๆ โดย Process Reaction Method

ชนิดตัวควบคุม	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	2L	0.5L

วิธีการ Ultimate Cycle Method เป็นวิธีการปรับค่าตัวควบคุมพารามิเตอร์ที่ทำให้ระบบมีการแกว่งแบบลดลงด้วยอัตรา 25% ซึ่งสามารถแสดงได้ด้วยสมการ $A_2 = \frac{1}{4}A_1$ เมื่อ A_1 และ A_2 แทน โอเวอร์ชูตแรก และ โอเวอร์ชูตที่สอง ของระบบ ตามลำดับ ขั้นตอนการปรับค่าพารามิเตอร์ ตัวควบคุมด้วยวิธี Ultimate Cycle เริ่มจากการทดลองตัวควบคุมแบบ Proportional control โดยการปรับค่าพารามิเตอร์ K_p จนกระทั่งระบบเกิดการแกว่งแบบ Oscillate จากนั้นบันทึกค่าพารามิเตอร์นี้ให้อยู่ในตัวแปร K_{cr} และหาคาบการแกว่งของสัญญาณ P_{cr} จากนั้นทำการคำนวณค่าพารามิเตอร์ตามตารางที่ 2.3

ตารางที่ 2.3 ค่าอัตราการขยายของตัวควบคุมต่าง ๆ โดย Ultimate Cycle Method

ชนิดของตัวควบคุม	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$1/6P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

2.1.6.2 วิธีการลองผิดลองถูก (Trial and error tuning) ในกรณีที่การปรับค่าพารามิเตอร์ดังวิธีการข้างต้นนั้นทำให้ระบบเกิดการตอบสนองที่ไม่เป็นดังที่ต้องการใช้งานหรือมีคุณสมบัติบางอย่างที่แตกต่างไปจากกระบวนการที่ต้องการควบคุม จึงได้มีวิธีการลองผิดลองถูกเข้ามาช่วยเพิ่มประสิทธิภาพของการควบคุมโดยทำการสุ่มค่าพารามิเตอร์ต่าง ๆ โดยอาศัยประสบการณ์และความชำนาญจากการทดลองหลายครั้ง

2.1.7 ตัวกำเนิดความร้อน

การเพิ่มอุณหภูมิของเตาความร้อนโดยการใช้งานที่บรรยากาศปกติทั่วไปมี 3 ลักษณะ คือ ตัวกำเนิดความร้อนจากลวดนิโครม ตัวกำเนิดความร้อนจากซิลิกอนคาร์ไบด์ และตัวกำเนิดความร้อนจากโมลิบดีนัมไดซัลไฟด์

2.1.7.1 ตัวกำเนิดความร้อนจากลวดนิโครม ตัวกำเนิดความร้อนจากลวดนิโครมมีลักษณะเป็นเส้นลวด สามารถเลือกใช้งานทั้งแบบเส้นแบน และแบบเส้นกลมตามความต้องการใช้งานมีขนาด และเส้นผ่านศูนย์กลางต่าง ๆ กัน มีการใช้งานในช่วงอุณหภูมิไม่เกิน 1,400 องศาเซลเซียส ในการเลือกใช้งานสามารถทำได้โดยการคำนวณจากค่าความต้านทานและความยาวของเส้นลวดตามสมการที่ 2.18

$$R = \rho \frac{l}{A} \quad (2.18)$$

เมื่อ R คือ ความต้านทานของเส้นลวด ρ คือ ความต้านทานของวัสดุ
 l คือ ความยาวของเส้นลวด และ A คือ พื้นที่หน้าตัดของเส้นลวด

2.1.7.2 ตัวกำเนิดความร้อนจากซิลิกอนคาร์ไบด์ ตัวกำเนิดความร้อนที่สร้างจากซิลิกอนคาร์ไบด์มีลักษณะเป็นแท่ง สามารถเลือกใช้งานได้ที่แรงดันไฟฟ้าตั้งแต่ 40 โวลต์ขึ้นไป และมีช่วงอุณหภูมิที่ใช้งานไม่เกิน 1,500 องศาเซลเซียส ทั้งนี้ต้องเลือกใช้งานให้เหมาะสมกับอุณหภูมิและพลังงานที่ใช้

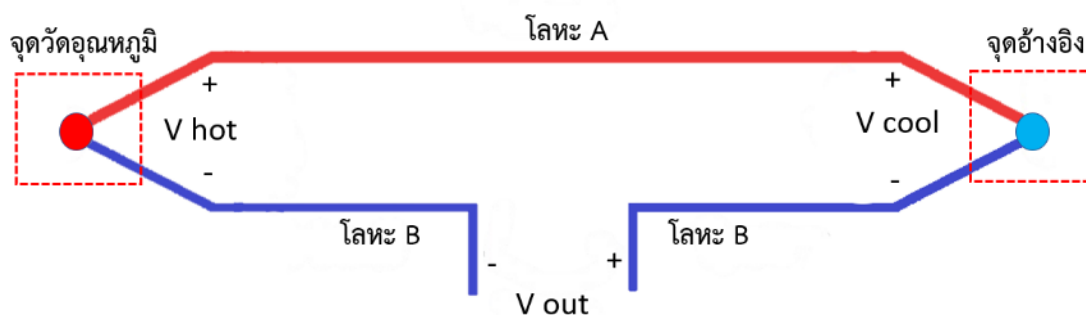
2.1.7.3 ตัวกำเนิดความร้อนจากโมลิบดีนัมไดซัลไฟด์ ตัวกำเนิดความร้อนที่สร้างจากโมลิบดีนัมไดซัลไฟด์มีลักษณะให้เลือกหลากหลายรูปทรง โดยทั่วไปแล้วนิยมใช้ในลักษณะของรูปตัวยู เนื่องจากสามารถถอดใช้งานได้ง่ายมีอุณหภูมิการใช้งานที่บรรยากาศปกติถึง 1,800 องศาเซลเซียส ในการใช้งานในช่วงอุณหภูมิต่าง ๆ จะเลือกจากเส้นผ่านศูนย์กลางของเส้นลวด

2.1.8 เทอร์โมคัปเปิล (Thermocouple) [8]

เทอร์โมคัปเปิลคืออุปกรณ์วัดอุณหภูมิที่อาศัยหลักการเปลี่ยนแปลงพลังงานความร้อนเป็นพลังงานไฟฟ้า ทำมาจากโลหะที่ต่างกันสองชนิดนำมาเชื่อมต่อกันที่ปลายของโลหะด้านหนึ่งเรียกว่า จุดวัดอุณหภูมิ ส่วนปลายอีกด้านหนึ่งเปิดไว้เรียกว่า จุดอ้างอิง เมื่ออุณหภูมิที่จุดวัดอุณหภูมิ และจุดอ้างอิงมีอุณหภูมิต่างกันจะทำให้เกิดการนำกระแสที่เทอร์โมคัปเปิลเกิดขึ้น

2.1.8.1 โครงสร้างเบื้องต้นของเทอร์โมคัปเปิล เมื่อเชื่อมโลหะสองชนิดเข้าด้วยกัน จะทำให้เกิดรอยต่อระหว่างโลหะสองแท่งและเมื่อจุดทั้งสองมีอุณหภูมิแตกต่างกันจะทำให้เกิดกระแสไฟฟ้าไหลภายในวงจร โดยรอยต่อของทางฝั่งร้อนจะเป็นขั้วบวก และรอยต่อเย็นจะเป็นขั้วลบ

ขนาดของแรงเคลื่อนไฟฟ้าที่เกิดขึ้นภายในเทอร์โมคัปเปิลจะขึ้นอยู่กับโลหะทั้งสองชนิดที่นำมาต่อกัน แสดงโครงสร้างเบื้องต้นของเทอร์โมคัปเปิลดังภาพที่ 2.13



ภาพที่ 2.13 โครงสร้างเบื้องต้นของเทอร์โมคัปเปิล

ตารางที่ 2.4 เทอร์โมคัปเปิลชนิดต่าง ๆ ที่นิยมในการใช้งาน

ชนิดของเทอร์โมคัปเปิล	โลหะที่ใช้	ช่วงการวัดอุณหภูมิ (°C)
J	เหล็กทองแดง - นิกเกิล	0 – 800
K	นิกเกิล - โครเมียม และ นิกเกิล - อลูมิเนียม	0 – 1100
T	ทองแดง และ ทองแดง - นิกเกิล	-200 – 400
E	นิกเกิล - โครเมียม และ ทองแดง - นิกเกิล	0 – 800
N	นิกเกิล - โครเมียม - ซิลิกอน และ นิกเกิล - ซิลิกอน	0 – 1250

2.1.8.2 คุณสมบัติของเทอร์โมคัปเปิล

- 1) ความไว ความไวในการใช้งานเทอร์โมคัปเปิลจะมีค่าขึ้นอยู่กับการใช้งาน วงจรปรับสภาพสัญญาณและชนิดของเทอร์โมคัปเปิล
- 2) ย่านการใช้งาน ย่านการใช้งานของเทอร์โมคัปเปิลแต่ละตัวจะขึ้นอยู่กับชนิดของเทอร์โมคัปเปิล ซึ่งแรงเคลื่อนไฟฟ้าที่ออกมาจากเทอร์โมคัปเปิลจะถูกอ้างอิงตามค่ามาตรฐานของชนิดเทอร์โมคัปเปิลนั้น ๆ
- 3) เวลาการตอบสนอง ความไวในการตอบสนองของเทอร์โมคัปเปิลจะขึ้นอยู่กับขนาดของสายสัญญาณและขึ้นอยู่กับวัสดุที่นำมาทำเป็นท่อป้องกันของเทอร์โมคัปเปิล
- 4) การปรับสภาพสัญญาณ ในกรณีที่สัญญาณจากเทอร์โมคัปเปิลมีค่าน้อยมาก จำเป็นต้องใช้วงจรขยายสัญญาณเข้ามาเพื่อขยายสัญญาณของข้อมูล

2.1.9 อิฐทนไฟ

อิฐทนไฟเป็นวัสดุที่ทนต่อการขึ้นเพื่อใช้ในการก่อสร้างผนังที่ต้องการใช้งานในสภาพที่มีความร้อนสูงโดยที่ตัวอิฐนั้นไม่เสียหายหรือเกิดการเสื่อมสภาพขึ้นเช่นการแตกร้าวของอิฐและการขยายตัวของอิฐ อิฐทนไฟโดยทั่วไปทำจากดินและมีส่วนผสมของอลูมินาเพื่อให้มีค่าการนำความร้อนต่ำ ความจุความร้อนต่ำใช้เป็นฉนวนกันความร้อนได้ดีมีการสูญเสียความร้อนน้อย

2.1.9.1 ชนิดของอิฐทนไฟ เมื่อพิจารณาตามจำนวนของอลูมินาที่ผสมอยู่ในอิฐทนไฟ จะสามารถแบ่งชนิดของอิฐทนไฟได้ออกเป็น 2 ชนิด คือ อิฐทนไฟมวลเบาหรือเรียกว่า อิฐฉนวนทนไฟ และ อิฐทนไฟมวลหนักหรือเรียกว่า อิฐหนัก ซึ่งอิฐทั้งสองมีลักษณะการนำไปใช้งานที่แตกต่างกันดังนี้

- 1) อิฐทนไฟมวลเบา หรือ อิฐฉนวนทนไฟ ผลิตจากดินที่มีส่วนผสมของอลูมินา ตั้งแต่ 35% - 60% มีลักษณะที่มีค่าการนำความร้อนต่ำ ความจุความร้อนต่ำ และมีความเป็นฉนวนที่ดี เนื่องจากมีการสูญเสียความร้อนน้อย สามารถใช้งานตั้งแต่ช่วงอุณหภูมิ 900 – 1000 องศาเซลเซียส อิฐชนิดนี้เป็นอิฐที่มีรูปพรุนในตัวทำให้มีความเปราะและแตกง่าย



ภาพที่ 2.14 อิฐมวลเบาหรืออิฐฉนวนทนไฟ

2) อิฐมวลหนัก หรืออิฐหนัก ผลิตจากดินที่มีส่วนผสมของอลูมินาตั้งแต่ 30% - 43% อิฐชนิดนี้มีความแข็งแรงสูงสามารถทนต่อการแตกร้าวเนื่องจากการเปลี่ยนแปลงของอุณหภูมิฉับพลันได้และทนต่อการหดตัวในช่วงอุณหภูมิสูงเหมาะในการนำไปใช้งานกับเตาที่ต้องการความแข็งแรง



ภาพที่ 2.15 อิฐมวลหนัก หรืออิฐหนัก

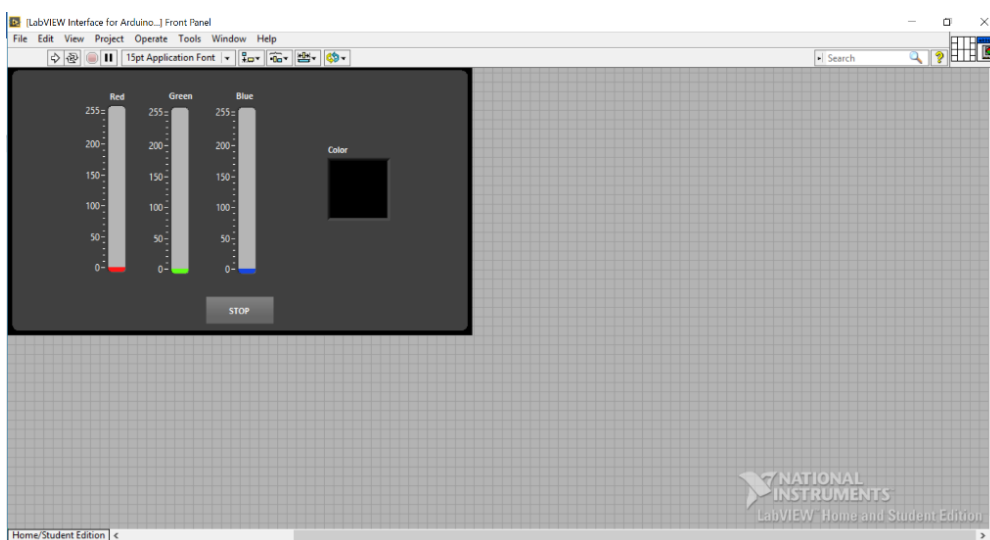
2.1.10 โปรแกรม LabVIEW [15]

Laboratory Virtual Instrument Engineering Workbench หรือโปรแกรม LabVIEW เป็นโปรแกรมทางคอมพิวเตอร์ที่ใช้ในการออกแบบระบบเพื่อนำมาใช้งานในการทดลองด้านการวัดปริมาณทางวิทยาศาสตร์ และสามารถแสดงผลการทดลองตามเวลาจริงบนหน้าจอคอมพิวเตอร์ได้ โปรแกรม LabVIEW มีการใช้งานโปรแกรมแกรมด้วยภาษารูปภาพทั้งหมด ทำให้สะดวกในการเขียนโปรแกรมมากขึ้น และจะช่วยลดเวลาในการใช้งานการเขียนโปรแกรมได้ โดยตัวโปรแกรมมีส่วนประกอบหลักในการใช้งาน 3 ส่วน ดังนี้

- (1) Front panel
- (2) Block Diagram
- (3) Icon และ Connector

ทั้ง 3 ส่วนนี้มีความสำคัญและหน้าที่ในการทำงานแตกต่างกัน ซึ่งมีรายละเอียดการใช้งานของแต่ละส่วนทั้ง 3 ส่วนดังจะแสดงต่อไปนี้

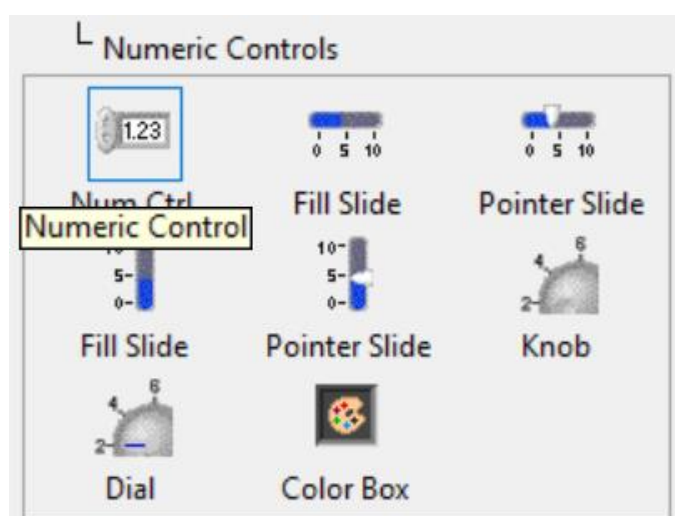
2.1.10.1 Front panel เป็นส่วนที่ใช้ป้อนค่าตัวแปร ควบคุม และแสดงผลของการทดลอง ซึ่งจะมีการเชื่อมโยงข้อมูลระหว่างส่วนของ Front panel และส่วนของ Block Diagram โดยทั่วไปแล้วหน้าจอแสดงผลในส่วนนี้จะประกอบไปด้วย เครื่องมือ หรือ อุปกรณ์ที่ใช้งานในการวัดและควบคุมทั่วไป แสดงหน้าต่างการใช้งานส่วน Front panel ได้ดังภาพที่ (2.16)



ภาพที่ 2.16 หน้าต่างการใช้งานและออกแบบ Front panel

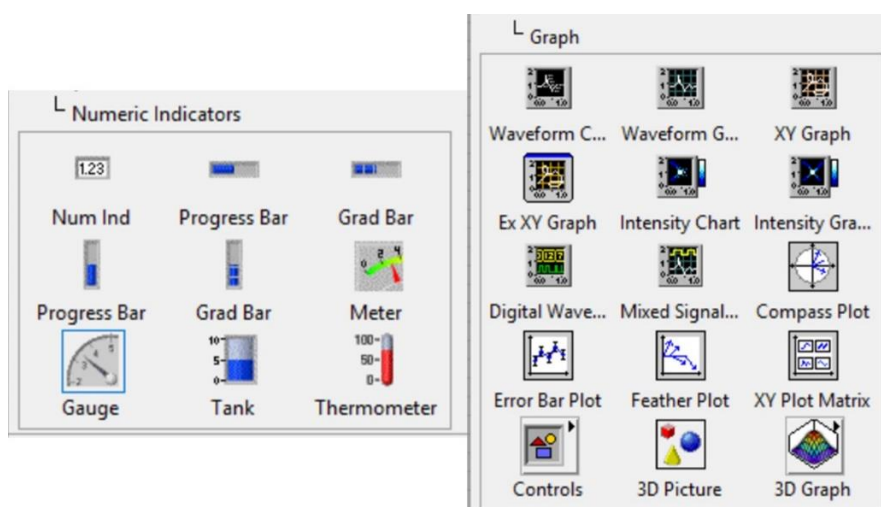
เครื่องมือและอุปกรณ์ในส่วน Front panel โดยทั่วไปแล้วจะถูกแบ่งออกมาได้ 2 แบบ คือ ตัวควบคุม (Control) และตัวแสดงผล (Indicator) มีรายละเอียดในการใช้งานในส่วนประกอบต่าง ๆ ดังนี้

1) ตัวควบคุม (Controls) มีหน้าที่เป็นตัวควบคุมค่าตัวแปรต่าง ๆ ซึ่งจะถูกป้อนข้อมูลเข้ามาโดยผู้ใช้งาน สามารถใช้งานได้ทั้งการ ปรับค่าข้อมูลแบบตัวเลข การปรับค่าข้อมูลแบบแท่งเลื่อน และสามารถเรียกใช้งานปุ่มควบคุมการ เปิด-ปิด การทำงานของฟังก์ชันต่าง ๆ ได้แสดงส่วนของการควบคุมได้ดังภาพที่ 2.17



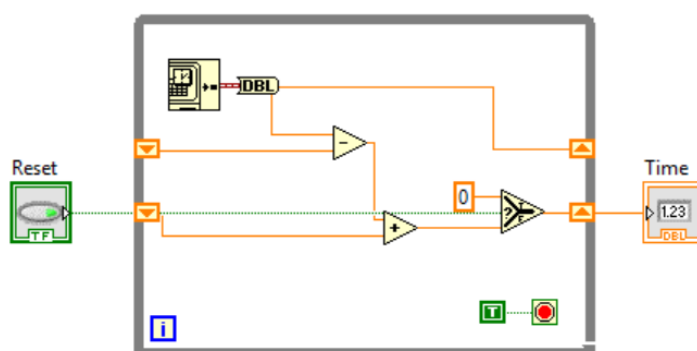
ภาพที่ 2.17 ตัวอย่างเครื่องมือควบคุมบน Front panel ของ LabVIEW

2) ตัวแสดงผล (Indicators) มีหน้าที่ในการแสดงผลที่ได้จากการทดลองจากฟังก์ชันในส่วน Block Diagram โดยสามารถแสดงผลในรูปของหน้าจอตัวเลขดิจิทัล กราฟ เข็มชี้ เป็นต้น ตัวแสดงผลในส่วน Front panel แสดงในภาพที่ 2.18



ภาพที่ 2.18 ตัวอย่างเครื่องมือแสดงผลบน Front Panel ของ LabVIEW

2.1.10.2 Block Diagram เป็นส่วนที่ใช้กำหนดการทำงานของโปรแกรม ซึ่งใช้ภาษารูปภาพในการพัฒนาในส่วนนี้จะประกอบไปด้วย ฟังก์ชัน ค่าคงที่ และ บล็อกการทำงานของโปรแกรมต่าง ๆ ให้เลือกใช้งาน สำหรับการเชื่อมต่อข้อมูลในแต่ละบล็อกจะใช้งาน Wire ซึ่งต้องใช้สายที่เหมาะสมเพื่อกำหนดชนิดข้อมูลที่ใช้ในการส่ง และรับในแต่ละบล็อก แสดงตัวอย่างการใช้งานการต่อ Block Diagram ดังภาพที่ 2.19



ภาพที่ 2.19 ตัวอย่างการออกแบบโปรแกรมในส่วน Block Diagram ของ LabVIEW

การใช้งาน Block Diagram จะมีส่วนที่ใช้ในการกำหนดการไหลของข้อมูลของระบบ และส่วนของการประมวลผลหลักอยู่ 4 ส่วน ดังนี้

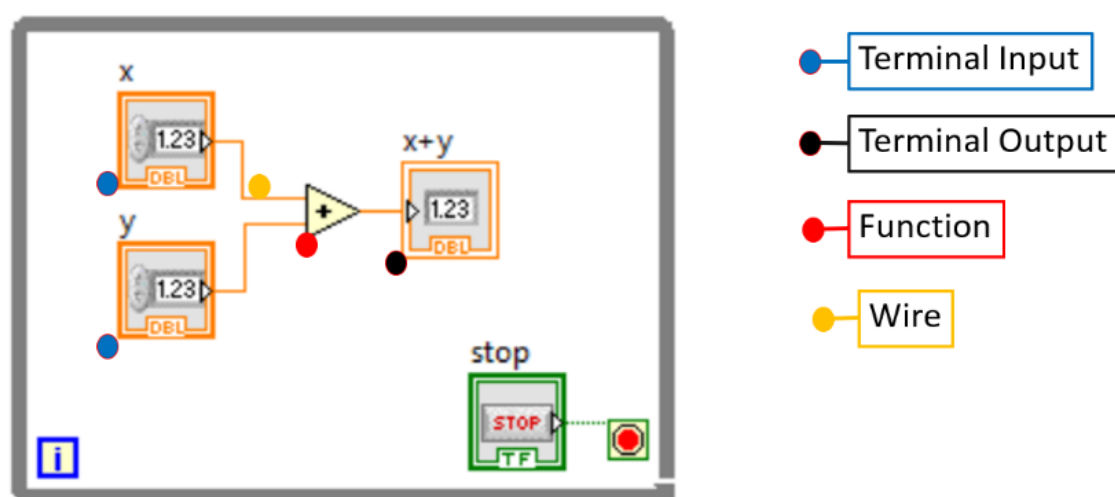
1) Terminal ในขณะที่สร้างบล็อกที่ใช้ในการ Control และ Indicator บนส่วนของ Front panel จะทำให้เกิดส่วนของบล็อกที่เป็นตัวที่ใช้ในการ Control และ Indicator เกิดขึ้นที่ Block Diagram ด้วยเช่นกัน ซึ่งบล็อกดังกล่าวจะทำหน้าที่เป็นเหมือนสถานีปลายทาง

ของข้อมูล และจะถูกนำไปใช้ในการกำหนดการทำงานของโปรแกรมต่อไปในส่วนต่าง ๆ ภายใน Block Diagram ต่อไป แสดงดังภาพที่ 2.19

2) Node เป็นจุดต่อบนบล็อกฟังก์ชันต่าง ๆ ทำงานเป็นจุดขาเข้าของข้อมูล (Input) หรือเป็นจุดขาออกของข้อมูล (Output) แสดงดังภาพที่ (2.19)

3) Function คือตัวดำเนินการต่าง ๆ ในโปรแกรม โดยจะเป็นฟังก์ชันที่ถูกสร้างมาสำเร็จและพร้อมใช้งานแล้ว ยกตัวอย่างเช่น ฟังก์ชันของการคำนวณ Sin, Cos, Tan, Log เป็นต้น แสดงส่วนของฟังก์ชันดังภาพที่ 2.19

4) Wire เป็นส่วนที่เชื่อมต่อข้อมูลระหว่าง Terminal หรือ Node ที่อยู่ในบล็อกต่าง ๆ เข้าด้วยกัน ลักษณะของสาย Wire จะมีสีที่แตกต่างกันตามชนิดของข้อมูลที่ใช้ในการส่ง แสดงได้ดังตารางที่ 2.5

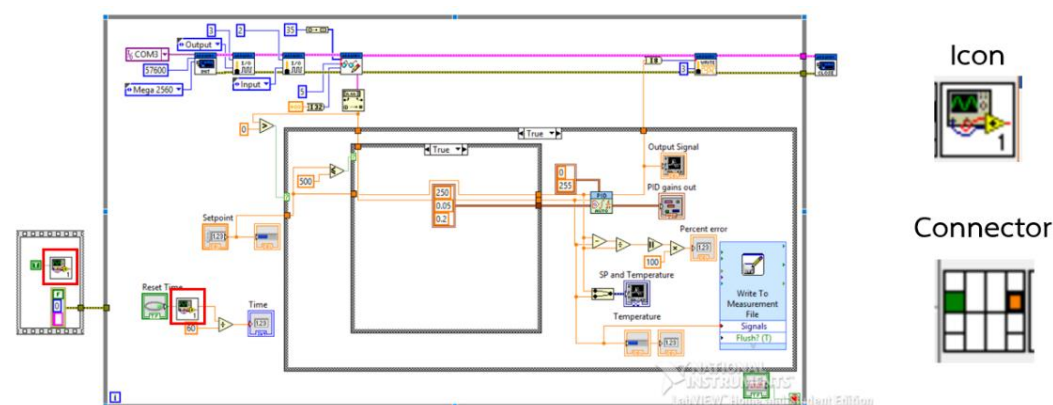


ภาพที่ 2.20 การใช้ Terminal Node Function และ Wire บน Block Diagram ของ LabVIEW

ตารางที่ 2.5 ลักษณะของเส้น Wire แบบต่าง ๆ

Type	Scalars	1-D Array	2-D Array
เลขทศนิยม	—————	—————	=====
เลขจำนวนเต็ม	—————	—————	=====
Boolean		~~~~~	~~~~~
ตัวอักษร	~~~~~	~~~~~	~~~~~

2.1.10.3 Icon และ Connector เป็นโปรแกรมย่อยที่ใช้ในการทำงานในลักษณะต่าง ๆ ที่ถูกสร้างขึ้นแล้วบันทึกให้เป็นบล็อก เมื่อใช้งานสามารถเรียกใช้งานเป็นบล็อกเล็ก ๆ ที่เชื่อมโยงกับตัวโปรแกรมภายใน โดยตัวบล็อกโปรแกรมที่ถูกสร้างขึ้นมาจะถูกเรียกว่าเป็น Sub VI ลักษณะทั่วไปของ Icon และ Connector แสดงดังภาพที่ 2.21



ภาพที่ 2.21 การออกแบบ Icon และ Connector บน Block Diagram ของ LabVIEW

2.1.11 ไมโครคอนโทรลเลอร์ Arduino [9]

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์แบบสำเร็จรูปในยุคปัจจุบัน ซึ่งถูกพัฒนาจากไมโครคอนโทรลเลอร์ตระกูล AVR ซึ่งมีข้อดีในการใช้งานคือเป็นบอร์ดไมโครคอนโทรลเลอร์แบบ Open Source ทำให้สามารถนำไปพัฒนาต่อได้หลากหลาย ทั้งนี้ตัวบอร์ดมีลักษณะภาษาเป็นภาษา (C/C++) ที่สามารถใช้งานได้ง่าย มีฮาร์ดแวร์และซอฟต์แวร์ที่ยืดหยุ่น สามารถประยุกต์ใช้งานได้ในระดับสูง ตัว Arduino มีส่วนประกอบหลักในการใช้งานที่สำคัญคือ ส่วนฮาร์ดแวร์และส่วนซอฟต์แวร์

2.1.11.1 ฮาร์ดแวร์ เป็นบอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ประกอบไปด้วยไมโครคอนโทรลเลอร์เป็นส่วนหลัก และมีแผงวงจรที่สามารถนำไปประยุกต์ใช้งานได้เพิ่มเข้ามาเป็นตัวเสริม การใช้งานสามารถติดต่อสื่อสารกับอุปกรณ์ภายนอกได้ด้วยขาและพอร์ตที่มีอยู่ในตัวบอร์ด เพื่อเพิ่มความหลากหลายในการพัฒนาและใช้งาน



ภาพที่ 2.22 บอร์ดบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3

2.1.11.2 ซอฟต์แวร์ ในการเขียนโปรแกรมสำหรับบอร์ด Arduino จะใช้ภาษา C/C++ โดยสามารถเรียกใช้งานฟังก์ชันและไลบรารีที่ทางผู้พัฒนา Arduino ได้เตรียมไว้สำเร็จแล้ว ทำให้มีความสะดวกในการใช้งาน สามารถแบ่งส่วนของการเขียนโปรแกรมได้เป็น 2 ส่วนหลักคือ

- 1) โครงสร้างของโปรแกรม ตัวแปรและค่าคงที่
- 2) ฟังก์ชัน

โครงสร้างของโปรแกรมใน Arduino มีแบ่งออกได้สองส่วนคือ Void setup() และ Void loop() ในส่วนของ setup() เป็นส่วนที่เมื่อเริ่มทำงานโปรแกรมจะทำตามคำสั่งที่อยู่บน setup() เพียงครั้งเดียว จึงใช้ในการกำหนดค่าตัวแปรเริ่มต้นในการทำงานของขาต่างๆ และเรียกใช้งานไลบรารี ตัวอย่างการใช้งานส่วน setup() แสดงดังภาพที่ 2.23

```
void setup() {
  // Setup serial port.
  Serial.begin(115200);
  Serial.println("Circuit Playground Analog Sensor Demos!");

  // Setup Circuit Playground library.
  CircuitPlayground.begin();
}
```

ภาพที่ 2.23 ตัวอย่างการกำหนดค่าตัวแปรเริ่มต้นในส่วน Void setup() ของ Arduino

เมื่อกำหนดตัวแปรในส่วนของ setup() แล้ว ในส่วนต่อมาคือฟังก์ชัน loop() ซึ่งมีการทำงานตามฟังก์ชันวนต่อเนื่องตลอดเวลา ภายในฟังก์ชันจะมีการเขียนโปรแกรมเพื่อรับค่าจากพอร์ตต่าง ๆ ที่เรียกใช้งาน เมื่อรับค่าเสร็จแล้วจะทำการเขียนโปรแกรมเพื่อคำนวณและประมวลผลก่อนจะส่ง

เอาต์พุตออกตามขาที่กำหนด เพื่อควบคุมการทำงานของบอร์ด แสดงตัวอย่างการเขียนโปรแกรมในส่วน Void loop() ดังภาพที่ 2.24

```
void loop() {
  uint16_t value = analogRead(ANALOG_INPUT);
  Serial.println(value, DEC);

  if (value < VALUE_MIN)      value = VALUE_MIN;
  else if (value > VALUE_MAX) value = VALUE_MAX;
  int red  = map(value, VALUE_MIN, VALUE_MAX, COLOR_RED_MIN , COLOR_RED_MAX);
  int green = map(value, VALUE_MIN, VALUE_MAX, COLOR_GREEN_MIN, COLOR_GREEN_MAX);
  int blue  = map(value, VALUE_MIN, VALUE_MAX, COLOR_BLUE_MIN , COLOR_BLUE_MAX);

  red  = CircuitPlayground.gamma8(red);
  green = CircuitPlayground.gamma8(green);
  blue  = CircuitPlayground.gamma8(blue);

  CircuitPlayground.clearPixels();
  CircuitPlayground.setPixelColor(4, red, green, blue);
  CircuitPlayground.setPixelColor(5, red, green, blue);

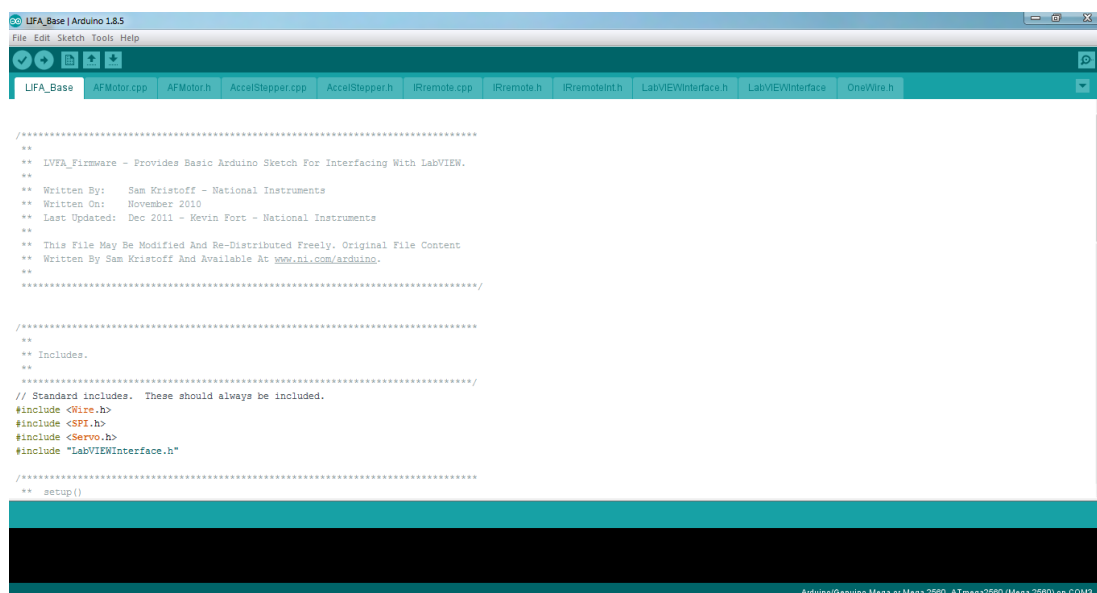
  int frequency = map(value, VALUE_MIN, VALUE_MAX, TONE_FREQ_MIN, TONE_FREQ_MAX);

  if (CircuitPlayground.slideSwitch()) {
    CircuitPlayground.playTone(frequency, 100);
  }
  delay(100);
}
```

ภาพที่ 2.24 ตัวอย่างการเขียนโปรแกรมในส่วนของ Void loop() ของ Arduino

2.1.12 LabVIEW Interface For Arduino

เป็นโปรแกรมเสริมให้ LabVIEW มีการใช้งานที่หลากหลายและสะดวกมากขึ้น เมื่อสามารถต่อใช้งานร่วมกับตัวโปรแกรม Arduino ได้ ซึ่งจะทำให้การพัฒนาโปรแกรมถูกแบ่งออกเป็นสองส่วน ส่วนแรกจะเป็นการพัฒนาโปรแกรมจาก LabVIEW ที่เรียกว่า LabVIEW Interface For Arduino และส่วนที่สองจะเป็นการเขียนฟังก์ชันการทำงานของโปรแกรมจาก Arduino ด้วยภาษา wiring ซึ่งถูกพัฒนาเป็นไลบรารีสำเร็จ เรียกว่า LIFA Base.



ภาพที่ 2.26 หน้าต่างการใช้งาน LabVIEW Interface for Arduino

หลังจากนั้นจะสามารถใช้งาน LabVIEW Interface For Arduino ได้ ทั้งนี้ผู้ใช้งานสามารถแก้ไขโค้ดโปรแกรมในการเพิ่มฟังก์ชันการใช้งานในส่วนของ LIFA_BASE ได้ โดยส่วนหลักๆที่จะถูกเรียกใช้งานแก้ไขโปรแกรมจะอยู่ใน Tap ของ LabVIEWInterface บนไลบรารี LIFA_BASE ของตัว Arduino

2.2 งานวิจัยที่เกี่ยวข้อง

การสร้างผนังกันความร้อนเป็นวิธีการใช้เพื่อป้องกันการส่งผ่านพลังงานความร้อนจากเตาความร้อน เพื่อลดการถ่ายเทพลังงานความร้อนที่จากภายในเตาความร้อนที่จะกระทบกับอุปกรณ์ภายนอกเตาความร้อน ในงานวิจัยได้ศึกษาวิธีการป้องกันการส่งผ่านความร้อนโดยการสร้างผนังกันความร้อนซึ่งมีงานวิจัยที่ศึกษาดังนี้

J. Roux et al. (1994) [10] ได้ออกแบบและสร้างเตาเผาอุณหภูมิสูง 1400 องศาเซลเซียส พบว่าในอุปกรณ์ได้ออกแบบผนังป้องกันความร้อนทำจากแผ่นอลูมิเนียมเพื่อใช้ป้องกันการส่งความร้อนจากภายในเตาออกมาภายนอกเตาทั้งหมด 5 ชั้น ผนังชั้นแรกสามารถลดอุณหภูมิจาก 1400 องศาเซลเซียสเหลืออุณหภูมิ 800 องศาเซลเซียส ผนังชั้นที่สองสามารถลดอุณหภูมิลงเหลือ 300 องศาเซลเซียส และวัดอุณหภูมิจากผนังชั้นสุดท้ายในระยะ 1 Cm ได้ อุณหภูมิ 30 องศาเซลเซียส

โปรแกรม LabVIEW หรือ Laboratory Virtual Instrument Engineering Workbench เป็นโปรแกรมที่สร้างเครื่องมือวัด และควบคุมชนิดหนึ่งในห้องปฏิบัติการทางวิศวกรรม ซึ่งสามารถใช้งานได้ง่าย เนื่องจากเป็นโปรแกรมประเภท GUI (Graphic User Interface) ที่มีลักษณะภาษาในการ

ใช้งานเป็นภาษารูปภาพ หรือที่เรียกว่า G (Graphical Language) และสามารถแสดงผลข้อมูลการวัดและควบคุมผ่านทางหน้าจอ Front panel ได้ งานวิจัยในก่อนหน้าที่นำโปรแกรมนี้มาใช้งานทางด้านการวัดและควบคุมอุณหภูมิ ดังนี้

Sankal Kumer et al. (2016) [11] ได้ออกแบบระบบควบคุมพีไอดี และปรับค่าตัวแปรควบคุมเพื่อควบคุมอุณหภูมิในช่วง 300 – 400 องศาเซลเซียส และแสดงผลค่าตัวแปรควบคุมและอุณหภูมิเป้าหมายผ่านหน้าจอด้วยโปรแกรม LabVIEW ผลการปรับค่าตัวแปรควบคุมแบบพีไอดีด้วยโปรแกรม LabVIEW ผลการปรับค่าตัวแปรควบคุมแบบพีไอดีด้วยระบบ Auto tune จาก LabVIEW พบว่าระบบมีค่าOvershoot ของสัญญาณเกิดขึ้นในช่วงระยะเวลาแรกของการควบคุมก่อนที่อุณหภูมิจะเริ่มเข้าใกล้ค่า setpoint

H. Muhammad Asraf et al. (2016) [12] ได้ศึกษาการใช้งานระบบควบคุมแบบพีไอดีด้วยโปรแกรม LabVIEW ร่วมกับ Arduino โดยทำการควบคุมอุณหภูมิในช่วง 100 องศาเซลเซียส พบว่าระบบสามารถควบคุมอุณหภูมิได้โดยมีค่าOvershoot เกิดขึ้นในช่วงระยะเวลาแรกของสัญญาณเหมือนเช่นเดียวกับการใช้งาน และแนะนำการใช้งาน LabVIEW interface for Arduino ว่าสามารถใช้งานได้ง่าย

การปรับค่าตัวแปรระบบควบคุมแบบพีไอดี ด้วยวิธีการของ Ziegler-Nichols ในการปรับค่าพารามิเตอร์ ในการทดลองควบคุมอุณหภูมิพบว่าอุณหภูมิที่ควบคุมเกิดการแกว่งรอบจุด setpoint และมีค่า Overshoot ที่มากเกินไป ซึ่งวิธีลดค่าความคลาดเคลื่อนจากการควบคุมด้วยการปรับค่าตัวแปรการควบคุมแบบ Ziegler-Nichols ร่วมกับการปรับค่าแบบ Trial and error ซึ่งมีตัวอย่างของงานวิจัยที่นำวิธีนี้มาใช้งาน ดังนี้

Y.V. pavan Kumar et al. (2013) [13] ทำการทดลองปรับค่าตัวแปรควบคุมแบบพีไอดีสำหรับควบคุมเตาความร้อนด้วยวิธีการของ Ziegler-Nichols ผลที่ได้จากการทดลองพบว่าระบบมีค่า Overshoot ที่สูง และสัญญาณควบคุมมีการสั่นรอบจุด Setpoint ซึ่งแก้ไขได้ด้วยการปรับค่าตัวแปรควบคุมแบบ Ziegler-Nichols ผสมกับวิธีการ Trial and error ผลที่ได้พบว่าค่า Overshoot ของสัญญาณลดน้อยลงกว่าเดิมระบบมีความผิดพลาดน้อย

บทที่ 3

วิธีดำเนินการวิจัย

3.1 วัสดุอุปกรณ์

3.1.1 อุปกรณ์ที่ใช้ในการทดลอง

3.1.1.1 เครื่องควบคุมกำลังไฟ SSR ขนาด 40 AMP. 220V. พร้อมฮีตซิงค์ จากบริษัทรวมวัสดุทนไฟ แอนด์ เซอร์วิส ราคา 885 บาท

3.1.1.2 สายไฟทนความร้อน จากบริษัท รวมวัสดุทนไฟ แอนด์ เซอร์วิส ราคา 145 บาท

3.1.1.3 ปลอกสายทนความร้อน 500 °C จากบริษัท รวมวัสดุทนไฟ แอนด์ เซอร์วิส ราคา 260 บาท

3.1.1.4 Arduino Mega 2560 จากบริษัท วินัส ซัพพลาย จำกัด ราคา 1,250 บาท

3.1.1.5 Thermocouple Type K(-200 to 1000°C) จากบริษัท เลกะ คอร์ปอเรชั่น จำกัด ราคา 2,100 บาท

3.1.1.6 IC MAX31850

3.1.1.7 ฉนวนใยหิน Rockwool ขนาด 600x500x25 mm. จากบริษัท ท็อป อินซูเลชั่น แอนด์ เทรตติ้ง จำกัด ราคา 1,033 บาท

3.1.1.8 ท่อแกรไฟต์ ขนาด 65/50x150 mm. จากบริษัท บีแอนด์บี อินเตอร์โปรดักส์ ราคา 4,250 บาท

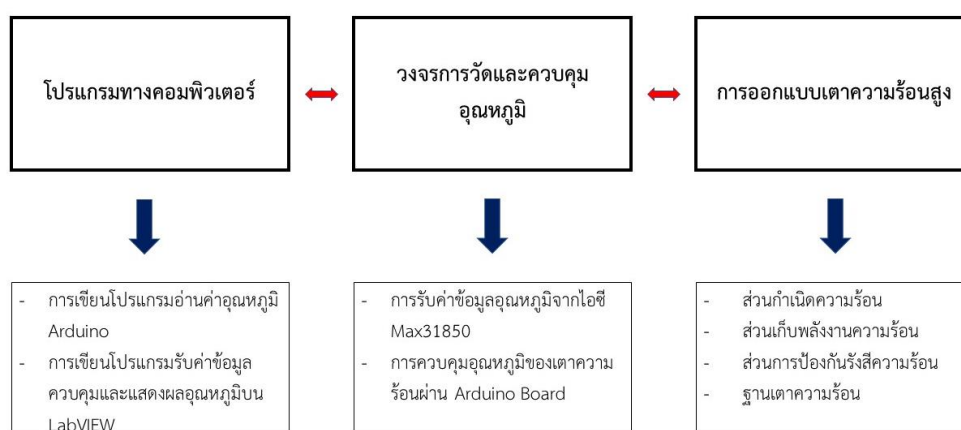
3.1.1.9 Block heater ขนาด 100x100x70 mm. จำนวนสองชิ้น จากบริษัท เอ.บี.เอส.ฮีต แอนด์เพอร์เนซ จำกัด ราคารวม 2 ชิ้น 9,000 บาท

3.1.1.10 ขดลวดความร้อน 110 โวลต์ 1000 วัตต์ จำนวนสองเส้น จากบริษัท เอ.บี.เอส.ฮีต แอนด์เพอร์เนซ จำกัด ราคารวมสองชิ้น 4,000 บาท

3.1.1.11 อิฐทนไฟ C2 ขนาด 70x70x120 mm จำนวน 20 ก้อน จากบริษัทเซล่าวัลซัพพลาย จำกัด ก้อนละ 72 บาท รวมเป็นเงิน 1,440 บาท

3.2 การทดลอง

การออกแบบเตาความร้อนอุณหภูมิสูงมีขั้นตอนในการดำเนินงานวิจัย ออกแบบ และพัฒนาอุปกรณ์ โดยขั้นตอนการทดลองมีส่วนที่สำคัญ 3 ส่วนคือ ส่วนโปรแกรมทางคอมพิวเตอร์ ส่วนวงจรการวัดและควบคุมอุณหภูมิ และส่วนออกแบบเตาความร้อน แสดงรายละเอียดของส่วนต่าง ๆ ในการดำเนินงานวิจัย ดังภาพที่ 3.1



ภาพที่ 3.1 รายละเอียดของงานวิจัย

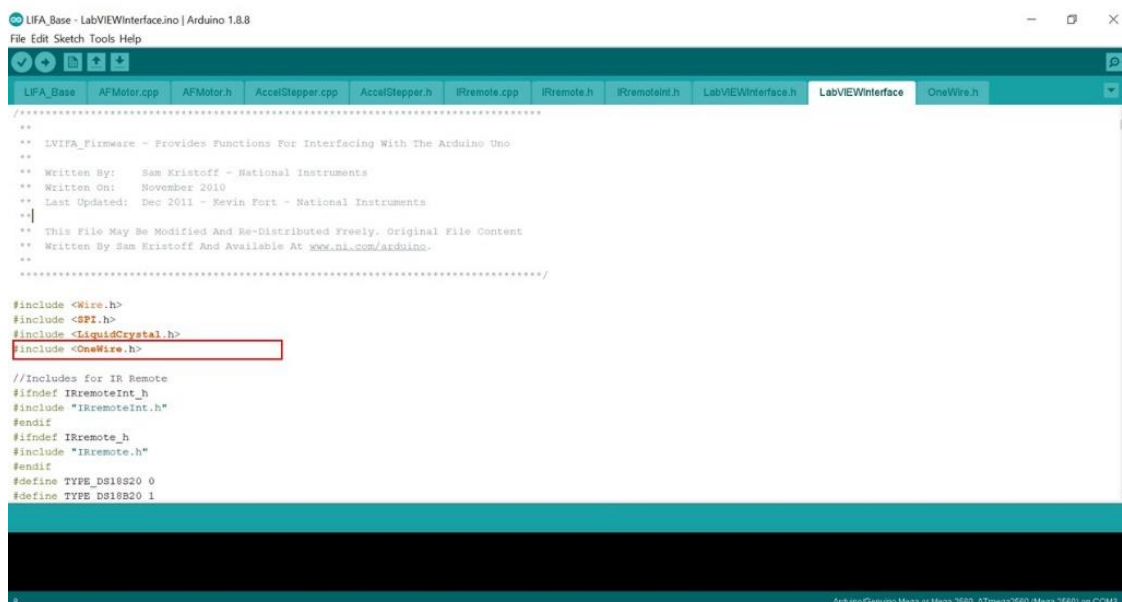
จากภาพที่ 3.1 รายละเอียดของงานวิจัยจะถูกแบ่งย่อยออกเป็น 3 ส่วนหลักๆ ในส่วนแรกจะเป็นส่วนการพัฒนาโปรแกรมทางคอมพิวเตอร์จะครอบคลุมตั้งแต่การเขียนโปรแกรมในส่วนของ Arduino ซึ่งมีการเพิ่มเติมโปรแกรมบน LIFA_BASE และการเขียนโปรแกรมวัดอุณหภูมิจากไอซี MAX31850 และมีการออกแบบโปรแกรมการควบคุมและแสดงผลอุณหภูมิด้วยโปรแกรม LabVIEW ส่วนที่สองเป็นส่วนการออกแบบวงจรที่ใช้ในการวัดและควบคุมเตาความร้อนและส่วนสุดท้ายจะเป็นการออกแบบและสร้างเตาความร้อนที่ใช้ในงานวิจัย

3.2.1 โปรแกรมทางคอมพิวเตอร์

การเขียนโปรแกรมควบคุมและแสดงผลอุณหภูมิในงานวิจัยนี้ใช้โปรแกรม LabVIEW ร่วมกับ Arduino Mega 2560 สามารถใช้งานได้โดยการติดตั้ง LabVIEW Interface for Arduino ซึ่งเป็นวิธีการทำให้บอร์ด Arduino ติดต่อกับ LabVIEW เพื่อทำการแสดงผลตัวแปรการควบคุมอุณหภูมิ การใช้งานดังกล่าวจำเป็นต้องทำการติดตั้ง LabVIEW Interface For Arduino ผ่านทางโปรแกรม VI Packaged Manager

3.2.1.1 การเขียนโปรแกรมควบคุมอุณหภูมิ Arduino เมื่อทำการติดตั้ง LabVIEW Interface For Arduino ลงบอร์ด Arduino เสร็จแล้ว เมื่อเปิดโปรแกรม Arduino ขึ้นมาจะได้

หน้าต่างการใช้งานแสดงดังภาพที่ 3.3 ซึ่งจะต้องทำการเพิ่มคำสั่งการใช้งานเข้าไปเพื่อให้สามารถติดต่อข้อมูลอุณหภูมิกับ LabVIEW ได้ แสดงการเพิ่มคำสั่งการใช้งานตามภาพต่อไปนี้



```

LIFA_Base - LabVIEWInterface.ino | Arduino 1.8.8
File Edit Sketch Tools Help

LIFA_Base AFMotor.cpp AFMotor.h AccelStepper.cpp AccelStepper.h IRemote.cpp IRemote.h IRemoteInt.h LabVIEWInterface.h LabVIEWInterface OneWire.h

/*
** LVIFA_Firmware - Provides Functions For Interfacing With The Arduino Uno
**
** Written By: Sam Kristoff - National Instruments
** Written On: November 2010
** Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
** This File May Be Modified And Re-Distributed Freely. Original File Content
** Written By Sam Kristoff And Available At www.ni.com/arduino.
**
** =====*/

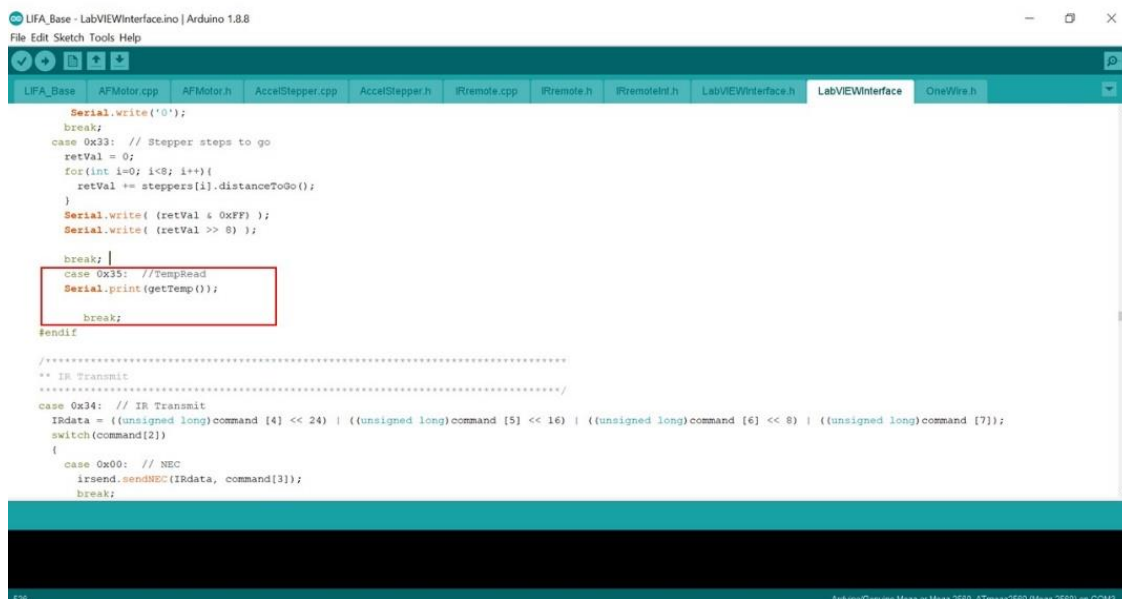
#include <Wire.h>
#include <SPI.h>
#include <LiquidCrystal.h>
#include <OneWire.h>

//Includes for IR Remote
#ifndef IRemoteInt_h
#include "IRremoteInt.h"
#endif
#ifndef IRemote_h
#include "IRremote.h"
#endif
#define TYPE_DS18S20 0
#define TYPE_DS18B20 1

```

ภาพที่ 3.2 การเพิ่มฟังก์ชันการติดต่อข้อมูลแบบ One-wire

จากภาพที่ 3.2 เนื่องจากในไลบรารี LIFA_BASE ที่ถูกพัฒนาใช้งานกับโปรแกรม LabVIEW ในตัว Arduino นั้นยังไม่สามารถใช้งานฟังก์ชันการติดต่อสื่อสารข้อมูลแบบ One-Wire ได้ ในการเริ่มต้นการใช้งานในขั้นตอนแรกจำเป็นต้องเพิ่มการติดต่อสื่อสารแบบ One-wire ในโปรแกรม โดยการ #Include <OneWire.h> เข้าไปในส่วนต้นโปรแกรมใน Tap LabVIEWInterface ที่เป็นส่วนในการตั้งค่าการใช้งาน



ภาพที่ 3.3 การเพิ่มฟังก์ชันการอ่านค่าข้อมูลอุณหภูมิ

จากภาพที่ 3.3 แสดงการเพิ่มฟังก์ชันการอ่านค่าข้อมูลอุณหภูมิ ฟังก์ชันที่สามารถเรียกใช้งานใน LIFA_BASE จะถูกเก็บไว้ในคำสั่ง Case ซึ่งสามารถเรียกใช้งานบนตัวโปรแกรม LabVIEW ทั้งนี้เนื่องจากผู้วิจัยต้องการเพิ่มฟังก์ชันการวัดอุณหภูมิที่จะรับค่าข้อมูลของอุณหภูมิจากการเขียนโค้ดวัดอุณหภูมิจากไอซี MAX31850 ที่จะแสดงในส่วนถัดไป โดยเมื่อเก็บข้อมูลอุณหภูมิไว้ในส่วนนี้แล้ว ข้อมูลอุณหภูมิที่ถูกเก็บไว้จะถูกเรียกนำไปแสดงผล และใช้ในการคำนวณในฟังก์ชัน PID ที่อยู่บน LabVIEW ดังนั้นผู้วิจัยจึงได้เพิ่มฟังก์ชันการอ่านค่าอุณหภูมิเพิ่มเข้ามาใน Case 35 และใช้แทนในตัวแปร get Temp () ดังภาพ อุณหภูมิที่อยู่ในฟังก์ชัน get Temp() นี้เป็นค่าอุณหภูมิที่ได้จากการเขียนโค้ดในการอ่านค่าจากไอซี Max 31850

ในส่วนถัดมาจะเป็นการแสดงการเพิ่มฟังก์ชันโปรแกรมการอ่านค่าข้อมูลอุณหภูมิจาก IC MAX 31850 โดยค่าอุณหภูมิที่อ่านได้จะถูกส่งไปยัง ฟังก์ชัน get Temp() ใน Case35 สามารถแสดงคำสั่งในการวัดอุณหภูมิดังแสดงดังต่อไปนี้

```
#include <OneWire.h>

// OneWire DS18S20, DS18B20, DS1822 Temperature Example
//
// http://www.pjrc.com/teensy/td\_libs\_OneWire.html
//
// The DallasTemperature library can do all this work for you!
// https://github.com/milesburton/Arduino-Temperature-Control-Library

OneWire ds(10); // on pin 10 (a 4.7K resistor is necessary)

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

ภาพที่ 3.4 การตั้งค่าตัวแปรและกำหนดขาอินพุตข้อมูล

จากภาพที่ 3.4 แสดงการตั้งค่าตัวแปรและการเรียกใช้งานการติดต่อข้อมูลแบบ One-Wire เพื่อที่จะสามารถเขียนโปรแกรมรับข้อมูลอุณหภูมิจากไอซี Max 31850 ในส่วนของการพัฒนาที่โปรแกรม Arduino ได้ หลังจากนั้นทำการกำหนดขาที่ใช้ติดต่อสื่อสารกับไอซี MAX31850 ที่ขา 10 ของ Arduino board ทั้งนี้ข้อมูลอุณหภูมิที่รับค่ามาจากไอซี Max 31850 จะสามารถส่งข้อมูลมาแสดงได้ที่หน้าต่าง Serial monitor โดยใช้ความเร็วในการรับส่งข้อมูล 9600 Mbps.

```
/* *****
** Functions get Temp
***** */
float getTemp(){
  //returns the temperature from one Max31850 in DEG Celsius
  byte i;
  byte present = 0;
  byte temptype;
  byte data[12];
  byte addr[8];
  float celsius, fahrenheit;
  if ( !ds.search(addr) ) {
    // Serial.println();
    // Serial.println("No more addresses.");
    // Serial.println();
    ds.reset_search();
    delay(250);
    return;
  }
  // Serial.print("ROM =");
  for( i = 0; i < 8; i++) {
    // Serial.write(' ');
    // Serial.print(addr[i], HEX);
  }

  if (OneWire::crc8(addr, 7) != addr[7]) {
```

ภาพที่ 3.5 แสดงการตั้งค่าการติดต่อสื่อสารกับ ROM ของไอซี MAX31850

จากภาพที่ 3.5 ในส่วนหัวหลักของตัวโปรแกรมเป็นการกำหนดค่าฟังก์ชัน get Temp() เพื่อแสดงว่าข้อมูลอุณหภูมิที่ประมวลผลได้ในฟังก์ชันนี้จะถูกส่งต่อไปยัง Case 35 ที่ใช้ชื่อฟังก์ชันว่า get Temp() ต่อมาจะเป็นการเรียกใช้งานคำสั่งการอ่านค่าจาก ROM ของไอซี MAX31850 โดยตัวArduino จะส่งพัลส์ไปที่ ROM หลังจากนั้นจะรอรับข้อมูลที่ส่งมาจากตัว ROM ข้อมูลที่ได้จากการอ่านค่าดังกล่าวนี้จะแสดง รหัสตระกูล, Serial number และรหัส CRC กลับมาให้แสดงผล โดยข้อมูลชุดดังกล่าวจะถูกนำไปตรวจสอบในกรณีที่มีการต่อพ่วงไอซี MAX31850 หลายตัวเข้าด้วยกัน ดังนั้นจะทำให้ทราบได้ว่าค่าอุณหภูมิที่ได้รับมานั้นมาจากไอซีตัวใด ทั้งนี้ผู้วิจัยได้เลือกที่จะไม่แสดงผลดังกล่าวเนื่องจากในงานวิจัยนี้ผู้วิจัยมีการใช้งานไอซีเพียงตัวเดียวเท่านั้น ดังนั้นจึงทำการตัดส่วนของโปรแกรมนี้ออกไป โดยจะแสดงเพียงแค่อุณหภูมิที่วัดได้จากไอซีเท่านั้น

```
switch (addr[0]) {
  case 0x10:
    // Serial.println("  Chip = DS18S20"); // or old DS1820
    temptype = TYPE_DS18S20;
    break;
  case 0x28:
    // Serial.println("  Chip = DS18B20");
    temptype = TYPE_DS18B20;
    break;
  case 0x22:
    // Serial.println("  Chip = DS1822");
    temptype = TYPE_DS18S22;
    break;
    // ADDED SUPPORT FOR MAX31850!
  case 0x3B:
    // Serial.println("  Chip = MAX31850");
    temptype = TYPE_MAX31850;
    break;
  default:
    // Serial.println("Device is not a DS18x20 family device.");
    return;
}
```

ภาพที่ 3.6 การตรวจสอบชนิดของไอซีที่อ่านได้จาก ROM

จากภาพที่ 3.6 เป็นการตรวจสอบว่าข้อมูลที่ได้จาก ROM ที่เป็นค่า ค่า รหัสตระกูล Serial number และรหัส CRC ที่ได้รับมานั้นตรงกับรหัสของไอซีตัวใดในกรณีที่มีการใช้งานต่อพ่วงไอซีหลายตัวจะสามารถรู้ได้ว่าอุณหภูมิที่วัดมาแต่ละชุดนั้นได้มาจากไอซีตัวใด และเมื่อทำการตรวจสอบแล้วระบบจะทำการแสดงผลของไอซีที่อ่านค่าออกมาโดยข้อมูลที่แสดงออกมาในส่วนนี้จะเป็นการแสดงผลของอุณหภูมิ พร้อมกับบอกรหัสตระกูล Serial number และ CRC ของไอซี

ดังกล่าว ออกมาเป็นชุด ๆ ของไอซีในแต่ละจุดเรียงลงมาทั้งนี้ผู้วิจัยได้เลือกที่จะไม่แสดงผลข้อมูลดังกล่าวด้วยสาเหตุเดิมคือ ทางผู้วิจัยมีการใช้งานไอซีเพียงตัวเดียว คือการใช้งานเพื่อวัดอุณหภูมิภายในเตาความร้อนเท่านั้นจึงมีความต้องการแสดงผลข้อมูลอุณหภูมิเพียงอย่างเดียวเท่านั้น

```

ds.select(addr);
ds.write(0x44, 1);          // start conversion, with parasite power on at the end

delay(1000);                // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE);              // Read Scratchpad

// Serial.print(" Data = ");
// Serial.print(present, HEX);
// Serial.print(" ");
for ( i = 0; i < 9; i++) {    // we need 9 bytes
    data[i] = ds.read();
    // Serial.print(data[i], HEX);
    // Serial.print(" ");
}
//Serial.print(" CRC=");
// Serial.print(OneWire::crc8(data, 8), HEX);
// Serial.println();

// Serial.print(" Address = 0x"); Serial.println(data[4] & 0xF, HEX);

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits

```

ภาพที่ 3.7 การอ่านค่าจากหน่วยความจำของไอซี MAX31850

จากภาพที่ 3.7 เป็นค่าอ่านค่าจากหน่วยความจำของไอซีที่ต่อพ่วงกับบอร์ด Arduino ซึ่งได้ใช้งานไอซี MAX31850 โดยทำการต่อวัดอุณหภูมิเพียงตัวเดียวเท่านั้นค่าที่อ่านข้อมูลที่อ่านค่าออกมาจากหน่วยความจำของไอซี MAX31850 นี้จะเป็นข้อมูลที่เกิดจากการเก็บค่าอุณหภูมิที่วัดได้จากไอซี โดยในส่วนนี้ผู้วิจัยได้ทำการตัดโปรแกรมในส่วนของการแสดงผลข้อมูลไอซี ที่ใช้บอกถึงข้อมูลจำเพาะของไอซีแต่ละตัวในกรณีที่มีการต่อพ่วงหลายตัวด้วยเหตุผลเดิม ทั้งนี้ค่าอุณหภูมิที่อ่านได้จากไอซีจะถูกนำไปแปลงให้สามารถอ่านค่าได้ในรูปแบบของเลขฐานสิบ เพื่อให้ง่ายต่อการนำไปต่อยอดการใช้งานในส่วนถัดไปที่อยู่ในตัวโปรแกรม LabVIEW

```

// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (temptype == TYPE_DS18S20) {
    raw = raw << 3; // 9 bit resolution default
    if (data[7] == 0x10) {
        // "count remain" gives full 12 bit resolution
        raw = (raw & 0xFFF0) + 12 - data[6];
    }
} else if (temptype == TYPE_MAX31850) {
    //Serial.println(raw, HEX);
    if (raw & 0x01) {
        Serial.println("***FAULT!***");
        return;
    }
} else {
    byte cfg = (data[4] & 0x60);
    // at lower res, the low bits are undefined, so let's zero them
    if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
    else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
    else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
    //// default is 12 bit resolution, 750 ms conversion time
}
celsius = (float)raw / 16.0;
float TemperatureSum = celsius;

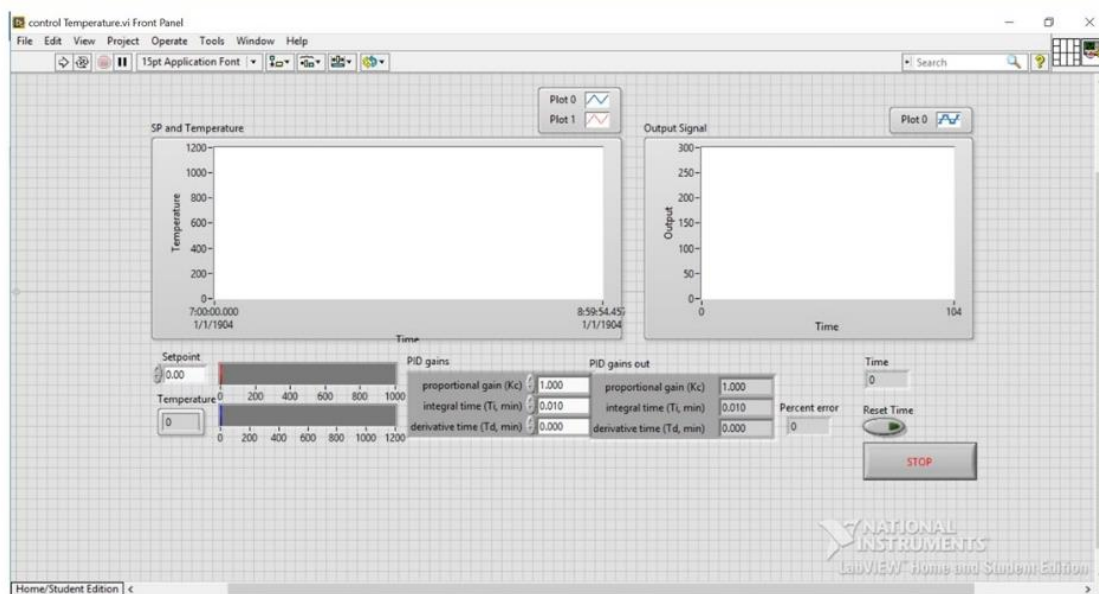
return TemperatureSum;
}

```

ภาพที่ 3.8 การอ่านค่าอุณหภูมิและแปลงข้อมูลเป็นให้อยู่ในรูปของเลขฐานสิบ

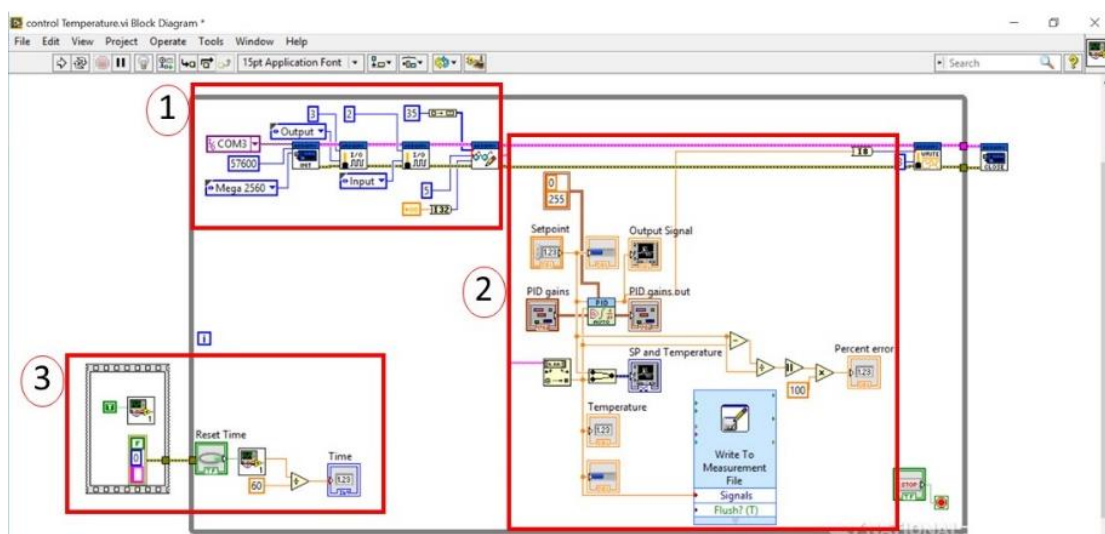
จากภาพที่ 3.8 เมื่อรับค่าอุณหภูมิจากไอซี MAX31850 แล้วต่อไปจะเป็นการแปลงข้อมูลอุณหภูมิให้อยู่ในรูปของเลขฐานสิบ จะทำให้ได้ค่าอุณหภูมิที่ต้องการออกมา เสร็จแล้วจะทำการส่งอุณหภูมิที่วัดได้กลับไปฟังก์ชัน get Temp() ต่อไป

3.2.1.2 การเขียนเขียนโปรแกรมรับส่งข้อมูลและแสดงผลจาก LabVIEW การออกแบบหน้าต่างแสดงผลและระบบควบคุม การออกแบบระบบควบคุมแบบพีไอดี โดยการใช้การติดต่อสื่อสารกับบอร์ด Arduino และ หน้าต่างแสดงผลตัวแปรและอุณหภูมิด้วยโปรแกรม LabVIEW มีส่วนสำคัญ 2 ส่วน คือ ส่วน Front panel เป็นส่วนที่ใช้สำหรับการออกแบบหน้าต่างการแสดงผลตัวแปรควบคุม และ อุณหภูมิของระบบเตา และส่วนที่ 2 คือ Block diagram. เป็นส่วนที่ใช้สำหรับการออกแบบคำสั่งการติดต่อสื่อสารกับ Arduino Board. และ ใช้ออกแบบระบบควบคุมแบบพีไอดี ซึ่งแสดงหน้าต่าง Front panel และ Block diagram ดังภาพที่ 3.9 และแสดงดังภาพที่ 3.10



ภาพที่ 3.9 Front panel สำหรับแสดงตัวแปรควบคุม และ อุณหภูมิ

จากภาพที่ 3.9 แสดงผังของ Front panel ที่ออกแบบบนโปรแกรม LabVIEW ใช้เพื่อส่งข้อมูลเพื่อตั้งค่าอุณหภูมิเป้าหมายในการทดลองและค่าตัวแปรพีไอดี ในการใช้ควบคุมอุณหภูมิกับโปรแกรม Arduino และในหน้าต่าง Front panel ยังใช้ในการแสดงผลของอุณหภูมิภายในเตาความร้อนและสัญญาณขาออกของระบบที่ใช้ในการควบคุมอุณหภูมิ ซึ่งสามารถแสดงเป็นกราฟออกมาตามเวลาจริงที่ใช้ในการทดลองแต่ละครั้ง



ภาพที่ 3.10 Block diagram แสดงการออกแบบระบบควบคุมพีไอดี และการติดต่อกับ Arduino

ภาพที่ 3.10 แสดง Block diagram ที่ออกแบบบนโปรแกรม LabVIEW ซึ่งในส่วนนี้เป็นส่วนที่ใช้ในการออกแบบคำสั่งในการทำงานของระบบไม่ว่าจะเป็นการรับข้อมูลอุณหภูมิที่ได้จาก Arduino และส่วนที่เป็นการออกแบบโปรแกรมควบคุมอุณหภูมิ โดยผลของการทดลองที่ได้จากส่วนนี้จะถูกส่งไปแสดงผลข้อมูลที่หน้าต่าง Front panel ประกอบไปด้วยสามส่วนหลัก

ส่วนที่ 1 เป็นส่วนที่ใช้รับค่าข้อมูลอุณหภูมิจากโปรแกรม Arduino โดยทำการป้อนข้อมูลของบอร์ด Arduino ที่ใช้เชื่อมต่อภายนอก ทั้งนี้ผู้วิจัยได้ใช้งาน Arduino Mega2560 ติดต่อสื่อสารที่ช่อง COM3 และส่งข้อมูลผ่าน serial monitor ความเร็วในการรับส่งข้อมูล 9600 Mbps. ทั้งนี้ในส่วนนี้ ยังได้ทำการกำหนดค่าที่ใช้ในการวัดสัญญาณควบคุมที่ส่งออกไปที่ขา 2 และขาที่ใช้ส่งสัญญาณควบคุมที่ขา 3 ของ Arduino จากนั้นรับข้อมูลอุณหภูมิมาจาก Case 35 ของโปรแกรม Arduino แสดงการออกแบบโปรแกรมการกำหนดค่าของบอร์ด Arduino ของส่วนที่ 1 ดังภาพที่ 3.10

ส่วนที่ 2 เป็นส่วนโปรแกรมควบคุมอุณหภูมิแบบพีไอดี โดยได้ทำการกำหนดค่าตัวแปรพื้นฐานในระบบควบคุมแบบพีไอดี เช่น ส่วนของการกำหนดค่าตัวแปรควบคุมพีไอดี (PID GAIN) ช่วงสัญญาณที่ใช้ในการควบคุมระบบ การป้อนค่า setpoint variable การป้อนค่า process variable หลังจากนั้นก็จะเป็นส่วนที่ใช้ในการแสดงผลของสัญญาณควบคุมที่ออกจากระบบ (Output signal) การตรวจสอบตัวแปรควบคุมพีไอดีในระบบควบคุม (PID GAIN OUT) จากนั้นจะเป็นส่วนที่ใช้ในการแสดงผลข้อมูลอุณหภูมิของค่า setpoint variable และ process variable และทำการคำนวณเปอร์เซ็นต์ความผิดพลาดของการควบคุมที่เกิดขึ้นที่ percent error จากนั้นผลการทดลองควบคุมอุณหภูมิที่ได้จะถูกบันทึกไว้ในไฟล์ Excel ในส่วนของ Write to measurement File แสดงการออกแบบโปรแกรมการควบคุมอุณหภูมิในส่วนที่ 2 ดังภาพที่ 3.10

ส่วนที่ 3 คือ ส่วนแสดงข้อมูลเวลาในการทดลองควบคุมอุณหภูมิ ซึ่งทำการเรียกใช้งาน SubVI ที่ได้ทำการออกแบบโปรแกรมเอาไว้แล้วตัวข้อมูลเวลานี้จะถูกแสดงผลที่หน้าจอ Front panel แสดงการออกแบบโปรแกรมการแสดงผลเวลาในการทดลองในส่วนที่ 3 ดังภาพที่ 3.10

3.2.1.3 การปรับค่าพารามิเตอร์แบบพีไอดี ในงานวิจัยนี้ใช้วิธีการปรับค่าตัวแปรของพีไอดีด้วยวิธีการของ Ziegler – Nichols โดยการใช้การทดลองควบคุมอุณหภูมิ และเปรียบเทียบตัวแปรด้วยกระบวนการ Ultimate Cycle Method และคำนวณค่าตัวแปรที่ได้ โดยมีความสัมพันธ์ของตัวแปรต่าง ๆ ดังนี้

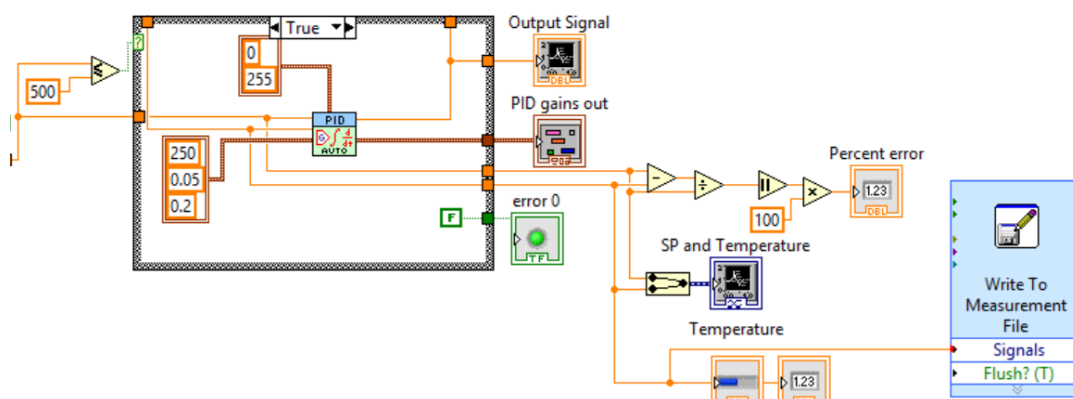
$$K_p = 0.6K_{cr} \quad T_i = 0.5P_{cr} \quad T_d = 0.125P_{cr} \quad (3.1)$$

เมื่อ K_{cr} P_{cr} คือ ค่าพารามิเตอร์ K_p ที่ทำให้ระบบเกิดการแกว่งแบบ Oscillate และคาบการแกว่งของสัญญาณ P_{cr} ตามลำดับ หลังจากนั้นจะปรับปรุงค่าตัวแปรเพิ่มเติมด้วยวิธีการลองผิดลองถูก

3.2.1.4 การทดลองปรับค่าตัวแปรควบคุมด้วยวิธีการลองผิดลองถูก การปรับค่าตัวแปรควบคุมด้วยวิธีการลองผิดลองถูกในงานวิจัยนี้ จะใช้วิธีการปรับค่าตัวแปรควบคุมโดยทำการสุมค่าตัวแปรพีไอดี โดยจะยึดการสุมค่าตัวแปรพีไอดีรอบตัวแปรที่ได้จากการทดลองด้วยวิธีการของ Ziegler-Nichol หลังจากนั้นทำการสังเกตสัญญาณควบคุมที่ได้จากการทดลองเพื่อหาสัญญาณควบคุมที่ดีที่สุดออกมา ซึ่งมีขั้นตอนในการทดลองดังนี้

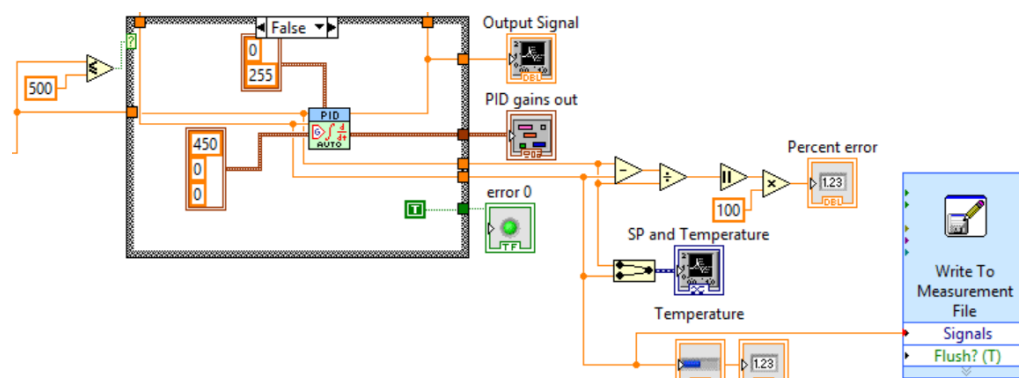
- 1) ปรับค่าอุปกรณ์ควบคุมให้อยู่ในสภาวะสมดุล
- 2) เปลี่ยนแปลงค่าพารามิเตอร์ควบคุม และสังเกตการณ์เปลี่ยนแปลง
- 3) ทำการทดลองแบบ P Control โดยการปรับค่าตัวแปร T_i และ T_d ให้เป็น 0 จากนั้นปรับอัตราการขยายการควบคุม K_p และสังเกตสัญญาณควบคุมที่ได้จากการปรับค่าให้มีผลการตอบสนองของระบบที่เหมาะสม
- 4) ทำการทดลองแบบ PI control โดยปรับค่าตัวแปร T_d เป็น 0 และใช้ค่าพารามิเตอร์ K_p จากการทดลองข้างต้น และปรับค่าในช่วงเวลาการเริ่มต้น T_i ให้มีผลการตอบสนองของระบบที่เหมาะสม
- 5) ทำการทดลองแบบ PID Control โดยใช้ค่าพารามิเตอร์ K_p และ T_i จากการทดลองข้างต้นและปรับค่าช่วงเวลา T_d เพื่อหาค่าการตอบสนองของระบบที่เหมาะสม

3.2.1.4 การออกแบบและแก้ไข Block diagram สำหรับควบคุมอุณหภูมิของ LabVIEW หัวข้อนี้นำเสนอผลการปรับแก้ Block diagram ของโปรแกรม LabVIEW ในส่วนของการควบคุมอุณหภูมินั้นได้สร้างตัวเลือกการทำงานของระบบควบคุมอุณหภูมิออกเป็น 2 ส่วน คือตัวแปรที่ใช้สำหรับควบคุมอุณหภูมิในช่วงอุณหภูมิน้อยกว่า หรือเท่ากับ 500 องศาเซลเซียส และกรณีต่อมาเป็นการออกแบบระบบควบคุมสำหรับควบคุมอุณหภูมิในช่วงที่อุณหภูมิ 500 – 1000 องศาเซลเซียส โดยได้สร้างฟังก์ชันใน Block diagram โดยกำหนดเมื่อค่า Setpoint มีค่าน้อยกว่าหรือเท่ากับ 500 องศาเซลเซียส หากเป็นจริงระบบจะใช้ค่าตัวแปรพีไอดี ดังนี้ K_p 250 T_i 0.05 และ T_d 0.20 แสดงดังภาพที่ 4.21 ถ้าหากไม่เป็นจริงตามกรณีดังกล่าวระบบจะเลือกใช้ค่าตัวแปรพีไอดี ดังนี้ $K_p = 450$ $T_i = 0$ และ $T_d = 0$ ดังแสดงในภาพที่ 3.11



ภาพที่ 3.11 Block diagram ของระบบควบคุมอุณหภูมิแบบพีไอดี ในกรณีที่ค่า Setpoint น้อยกว่าหรือเท่ากับ 500 องศาเซลเซียส

จากภาพที่ 3.11 ทำการใส่เงื่อนไขในการทำงานของระบบควบคุมแบบพีไอดีที่จะใช้ในการเลือกตัวแปรควบคุมที่อยู่ใน loop ของฟังก์ชัน โดยกำหนดเงื่อนไขในการเลือกใช้งานตัวแปรควบคุมไว้ว่า ถ้าค่า setpoint ที่ป้อนให้กับระบบควบคุมแบบพีไอดีมีค่าน้อยกว่าหรือเท่ากับ 500 องศาเซลเซียส ทั้งนี้ระบบจะแสดงผลการตอบรับเป็นเงื่อนไข True เมื่อมีการกำหนด setpoint น้อยกว่าหรือเท่ากับ 500 องศาเซลเซียสจริง และจะแสดงผล False เมื่อมีการกำหนดค่า setpoint มากกว่า 500 องศาเซลเซียส ซึ่งไม่ตรงตามเงื่อนไขในการใช้งาน โดยในภาพที่ 3.9 นี้แสดงการทำงานของเงื่อนไข เมื่อมีการป้อนค่า setpoint ให้กับระบบควบคุมอุณหภูมิแบบพีไอดี น้อยกว่าหรือเท่ากับ 500 องศาเซลเซียสจริง จะทำการใช้งานตัวแปรควบคุมพีไอดีตาม loop True ซึ่งได้กำหนดตัวแปรควบคุมแบบพีไอดีดังนี้ $K_p = 250$ $T_i = 0.05$ และ $T_d = 0.200$ โดยมีช่วงของสัญญาณการควบคุมอุณหภูมิอยู่ที่ 0- 255 เสร็จแล้วตัวแปรควบคุมจะถูกส่งเข้าไปในฟังก์ชันควบคุมแบบพีไอดีและส่งสัญญาณควบคุมไปที่ขา 3 ของ Arduino board

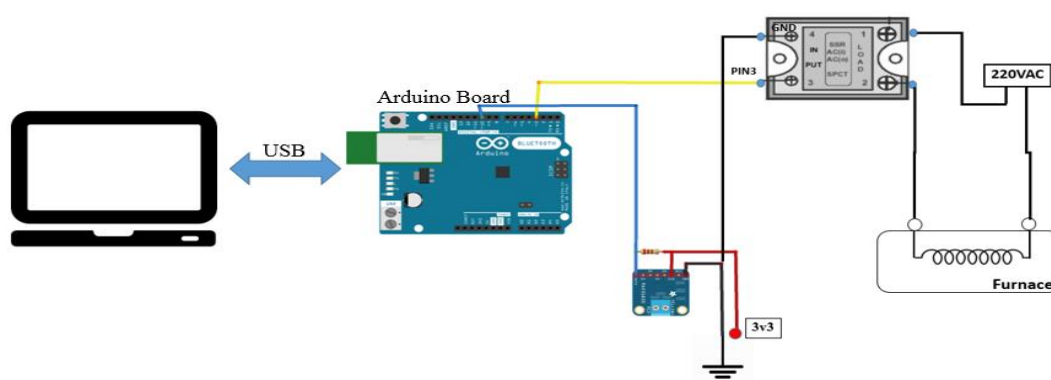


ภาพที่ 3.12 Block diagram ของระบบควบคุมอุณหภูมิแบบพีไอดี ในกรณีที่ค่า Setpoint มากกว่า 500 องศาเซลเซียส

จากภาพที่ 3.12 แสดงการทำงานของเงื่อนไข เมื่อมีการป้อนค่า setpoint ให้กับระบบควบคุมอุณหภูมิแบบพีไอดีมากกว่า 500 องศาเซลเซียสจริง ระบบจะทำการใช้งานตัวแปรควบคุมพีไอดีตาม loop False ซึ่งได้กำหนดตัวแปรควบคุมแบบพีไอดีที่ใช้ดังนี้ $K_p = 450$ $T_i = 0$ และ $T_d = 0$ โดยมีช่วงของสัญญาณการควบคุมอุณหภูมิอยู่ที่ 0- 255 เสร็จแล้วตัวแปรควบคุมจะถูกส่งเข้าไปในฟังก์ชันควบคุมแบบพีไอดี และส่งสัญญาณควบคุมไปที่ขา 3 ของ Arduino board

3.2.2 วงจรวัดและควบคุมอุณหภูมิ

งานวิจัยได้ใช้ Arduino Mega 2560 Board ซึ่งทำหน้าที่อ่านค่ารับ-ส่งอุณหภูมิจาก IC MAX31850 และควบคุมการทำงานของขดลวดความร้อนผ่านทาง Solid state relay การเชื่อมต่ออุปกรณ์ และวงจรวัดและควบคุมอุณหภูมิแสดงในวงจรภาพที่ 3.13

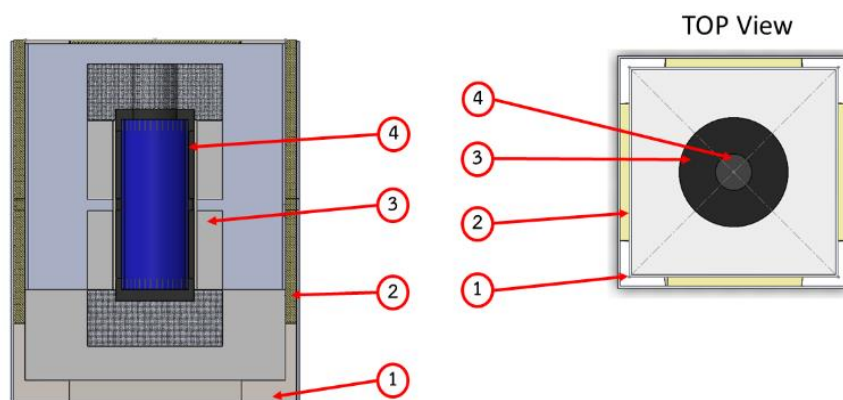


ภาพที่ 3.13 แสดงการเชื่อมต่อ Arduino และ MAX31850 สำหรับวัดและควบคุมอุณหภูมิ

3.2.3 การออกแบบเตาความร้อน

เตาความร้อนที่ออกแบบและใช้ในงานวิจัยนี้ได้ออกแบบมาเพื่อให้มีลักษณะที่เหมาะสมในการใช้ในงานทดลองทางวิทยาศาสตร์ ทั้งนี้เตาความร้อนสามารถเพิ่มอุณหภูมิภายในเตาได้ถึง 1000 องศาเซลเซียส และมีทางเดินของสัญญาณข้อมูลเช่น ข้อมูลรังสีเอ็กซ์ที่ได้จากการวัดทางวิทยาศาสตร์ ทั้งนี้ได้ออกแบบให้มีช่องดังกล่าวทั้งหมด 3 ช่อง โดยสมมติให้เป็นช่องทางเดินของรังสีเอ็กซ์โดยจะประกอบไปด้วย เส้นทางเดินของรังสีเอ็กซ์ที่เดินทางตกกระทบบสารตัวอย่าง และเส้นทางเดินของรังสีเอ็กซ์ที่ทะลุผ่านสารตัวอย่าง สุดท้ายคือ เส้นทางเดินของรังสีเอ็กซ์จากการเกิดฟลูออเรสเซนซ์ ซึ่งเตาความร้อนที่ออกแบบในงานวิจัยมีองค์ประกอบหลักด้วยกัน 4 ส่วน ดังนี้

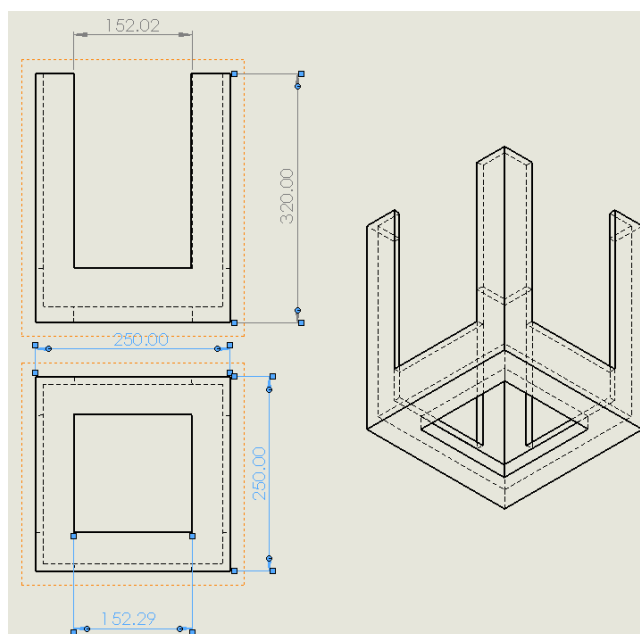
- (1) ฐานเตาความร้อน เป็นส่วนที่ใช้ยึดฉนวนกันความร้อน
- (2) Block heater เป็นส่วนเพิ่มความร้อนภายในเตาความร้อน
- (3) ส่วนกักเก็บพลังงานความร้อน
- (4) ส่วนป้องกันรังสีความร้อน



ภาพที่ 3.14 ภาพรวมของอุปกรณ์

จากภาพที่ 3.14 แสดงภาพรวมของอุปกรณ์ ออกแบบให้มีส่วนประกอบหลักทั้งหมด 4 ส่วน ดังรูป โดยส่วนที่ 1 คือ ฐานเตาความร้อน ทำจากเหล็กกล่อง เป็นส่วนที่ใช้ยึดผนังฉนวนกันความร้อนกับเตาความร้อน ส่วนที่ 2 คือ ผนังฉนวนกันความร้อน เป็นส่วนป้องกันรังสีความร้อนภายในเตา ถัดมา ส่วนที่ 3 คือ Block heater เป็นส่วนกำเนิดความร้อนภายในเตาความร้อน และส่วนที่ 4 คือ ท่อแกรไฟต์ทำหน้าที่ในการกักเก็บพลังงานความร้อนภายในเตา

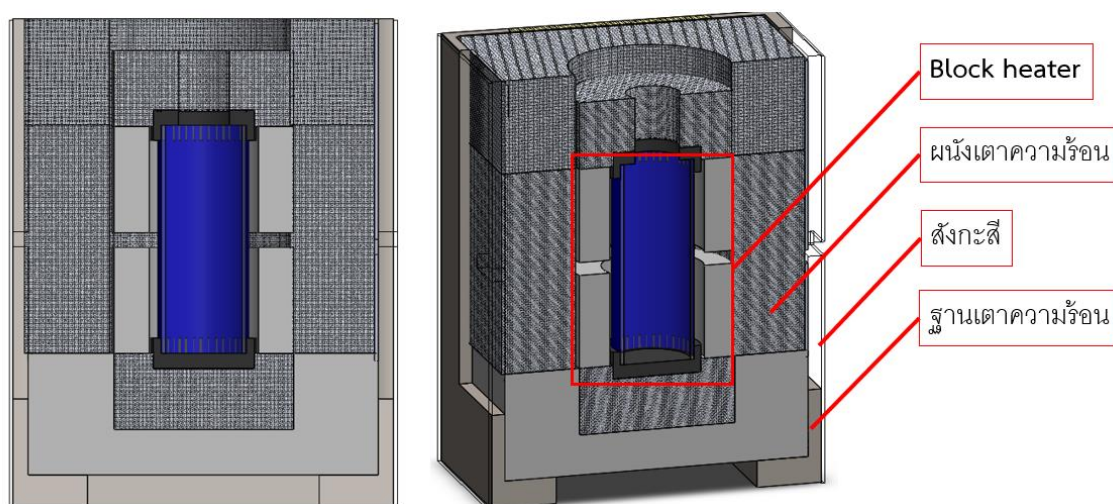
3.2.3.1 ฐานเตาความร้อน ฐานเตาความร้อนทำจากเหล็กมีความหนา 5 มิลลิเมตร เชื่อมเป็นโครงของอุปกรณ์ใช้เป็นที่วางเตาความร้อนโดยออกแบบให้มีขนาดความกว้าง 250 มิลลิเมตร ยาว 250 มิลลิเมตร และมีความสูง 300 มิลลิเมตร ซึ่งมีลักษณะตามภาพที่ 3.15



ภาพที่ 3.15 ฐานเตาความร้อน

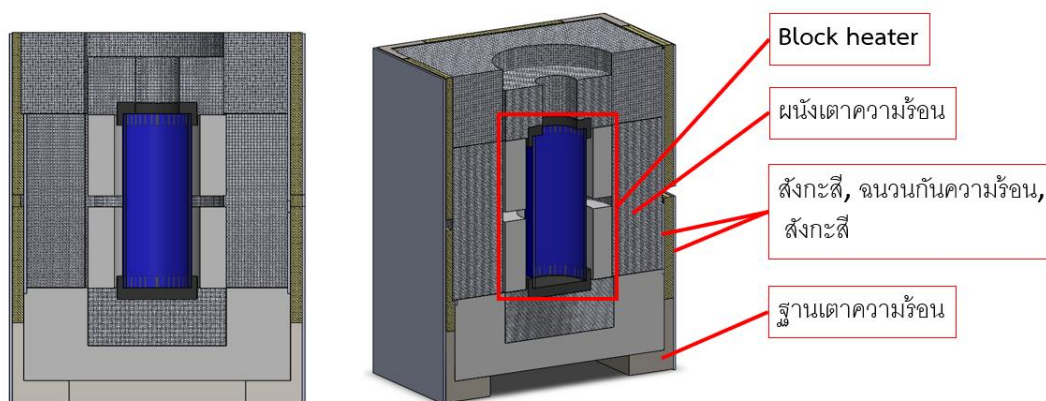
3.2.3.2 Block heater สร้างจากอิฐทนไฟร้อยด้วย Heater ความต้านทาน 12 โอห์ม กำลัง 1000 W ความต่างศักย์ 110V เตาความร้อนประกอบด้วย Block heater สองชิ้นประกบกันเป็นส่วนบนและส่วนล่างของเตาความร้อนโดยเว้นช่องว่างเพื่อให้รังสีเอ็กซ์สามารถทะลุผ่านไปได้ ขนาด 10 มิลลิเมตร ต่อมา Heater ทั้งสองเส้นจาก Block heater ถูกต่ออนุกรมกัน ดังนั้นจะได้ขดลวดความต้านทานรวม 24 โอห์ม กำลัง 2000 W ความต่างศักย์ 220 V ถัดมาตรงกลาง Block heater มีลักษณะกลวงไว้สำหรับส่วนกักเก็บพลังงานความร้อนของเตาความร้อน

3.2.3.3 ส่วนกักเก็บพลังงานความร้อน ส่วนกักเก็บพลังงานความร้อนจะใช้วิธีการลดการสูญเสียความร้อนภายในระบบด้วยวิธีการสร้างผนังของเตาความร้อนเพื่อป้องกันการสูญเสียความร้อนภายในออกสู่ภายนอกทั้งนี้ได้ใช้สังกะสีเข้ามาเป็นตัวช่องให้เกิดความมิดชิดของผนังเตาความร้อนไม่ให้เกิดช่องว่างขึ้นด้วย โดยความร้อนที่เกิดขึ้นภายในเตาความร้อนจะถูกกักเก็บไว้ภายในเนื่องจากชั้นผนังเตาความร้อนที่ประกอบด้วยสังกะสีที่ออกแบบให้เกิดความมิดชิด หลังจากนั้นทำการปิดเตาด้านบนด้วยอิฐทนไฟที่เจาะรูเพื่อเป็นส่วนที่ใช้ในการใส่สารตัวอย่างที่ต้องการศึกษา แสดงส่วนของผนังป้องกันการสูญเสียความร้อนดังภาพที่ 3.16



ภาพที่ 3.16 โครงสร้างเตาความร้อน

3.2.3.4 ส่วนป้องกันความร้อนจากการส่งผ่านความร้อน ด้วยวิธีการสร้างผนังเพื่อลดการพาความร้อน และการแผ่รังสีความร้อนจากภายในเตา โดยผนังกันความร้อน สร้างจากแผ่นสังกะสีจำนวนสองชั้น และเพิ่มใยหินในระหว่างชั้นผนังกันความร้อนเพื่อช่วยดูดซับความร้อนที่ส่งผ่านออกมา ซึ่งผนังเตาความร้อนนี้จะครอบคลุมทั้ง 4 ด้านของเตาความร้อน แสดงโครงสร้างส่วนป้องกันความร้อนในภาพที่ 3.17



ภาพที่ 3.17 โครงสร้างส่วนป้องกันความร้อน

3.2.4 วิธีการทดลองระบบควบคุมที่อุณหภูมิต่าง ๆ

การทดลองนี้เป็นการทดลองเพื่อตรวจสอบระบบควบคุมอุณหภูมิแบบพีไอดีที่ได้รับค่าตัวแปรจากวิธีการ Ziegler-Nichol ผสมกับวิธีการลองผิดลองถูก ในส่วนนี้จะทำการแบ่งช่วงการทดลองออกเป็นสองช่วง เนื่องจากในแต่ละช่วงนั้นใช้ตัวแปรในการควบคุมที่ต่างกัน โดยจะแบ่งเป็นการทดลองเพื่อควบคุมอุณหภูมิในช่วงอุณหภูมิต่ำ และการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง

3.2.4.1 การทดลองควบคุมอุณหภูมิของเตาความร้อนที่อุณหภูมิต่ำ ทำการทดลองควบคุมอุณหภูมิของเตาความร้อนที่อุณหภูมิต่ำ โดยทดลองในช่วงอุณหภูมิ 200 300 400 และ 500 องศาเซลเซียส ในการทดลองจะทำการปรับค่าตัวแปรควบคุมแบบพีไอดี และหาค่าตัวแปรที่เหมาะสมตามวิธีการของ Ziegler – Nichol ผสมกับวิธีการ Trial and error เมื่อได้ค่าตัวแปรที่เหมาะสมแล้ว จะทำการทดลองควบคุมอีกครั้งเพื่อตรวจสอบการทำงานของระบบควบคุม และสรุปผลการทดลองว่าระบบสามารถควบคุมอุณหภูมิได้ดีมากน้อยอย่างไร

3.2.4.2 วิธีการทดลองควบคุมอุณหภูมิกับเตาอุณหภูมิสูง ทำการทดลองควบคุมอุณหภูมิของเตาอุณหภูมิสูงที่ประกอบขึ้นในช่วงอุณหภูมิ 600 800 และ 1000 องศาเซลเซียส ในการทดลองจะทำการปรับค่าตัวแปรควบคุมแบบพีไอดีและหาค่าตัวแปรควบคุมที่เหมาะสมกับเตาความร้อนตามวิธีการของ Ziegler – Nichol ผสมกับวิธีการ Trial and error เมื่อได้ค่าตัวแปรที่เหมาะสมแล้ว จะทำการทดลองควบคุมอีกครั้งเพื่อตรวจสอบการทำงานของระบบควบคุม และหาค่าความผิดพลาดของระบบว่ามี error เท่าไร

3.2.5 วิธีการทดลองวัดอุณหภูมิที่ส่งออกมาภายนอกเตาความร้อน

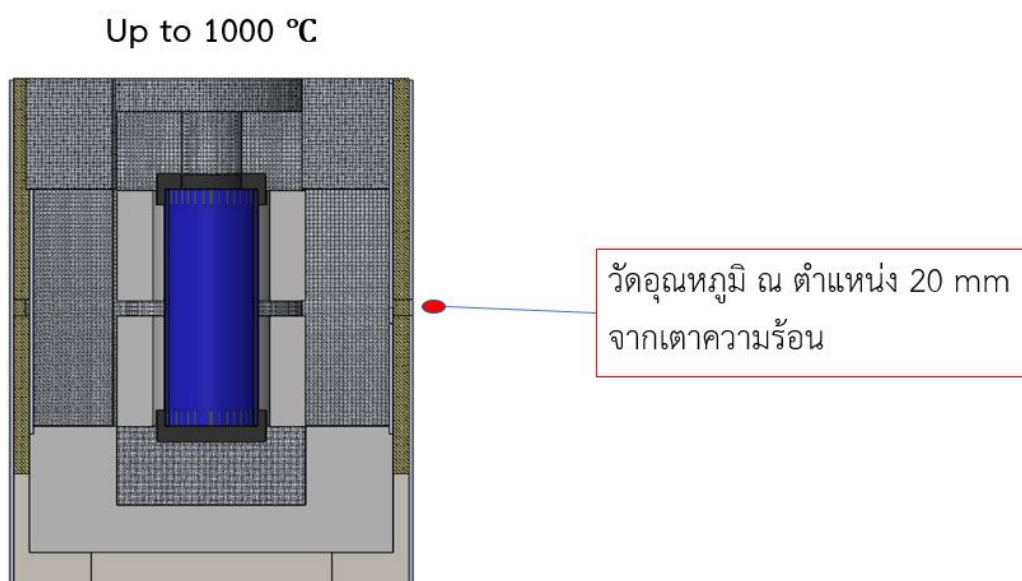
การทดลองนี้เป็นการทดลองเพื่อวัดความร้อนที่ส่งออกมาภายนอกเตาความร้อน ซึ่งจะทำให้การวัดทั้งความร้อนที่ออกมาจาก Block heater และความร้อนที่เกิดขึ้นภายนอกเตาความร้อนทั้งจากการนำความร้อน และการแผ่รังสีความร้อน ดังแสดงในหัวข้อย่อยต่อไปนี้

3.2.5.1 การวัดอุณหภูมิที่ส่งผ่านออกมาจาก Block heater ในการทดลองนี้เป็นการทดลองเพื่อทดสอบการป้องกันความร้อนของโครงสร้าง Block heater ทั้งนี้ยังสามารถบอกได้อีกว่าตัว Block heater ของเราสามารถเก็บความร้อนไว้ภายในเตาความร้อนได้ดีหรือไม่ โดยทำการทดลองควบคุมอุณหภูมิของเตาที่ 1000 องศาเซลเซียส และวัดอุณหภูมิที่ส่งออกมาภายนอกเตาความร้อนที่ระยะ 2 4 6 8 10 12 14 16 18 และ 20 เซนติเมตร เพื่อตรวจสอบประสิทธิภาพการทำงานในส่วนการป้องกันรังสีความร้อนภายในเตา แสดงดังภาพที่ 3.18



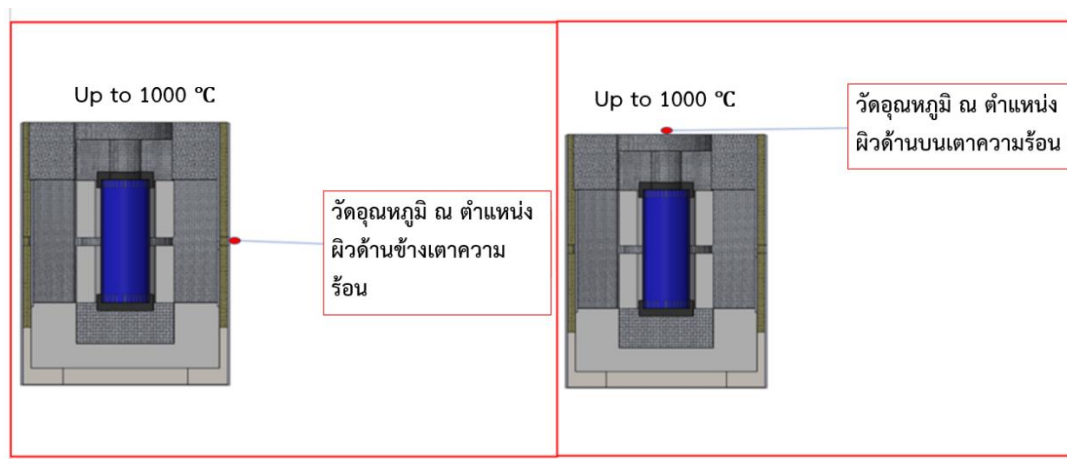
ภาพที่ 3.18 การวัดอุณหภูมิภายนอก Block heater

3.2.5.2 ทำการทดลองวัดอุณหภูมิที่ส่งผ่านออกมาจากเตาความร้อนเนื่องจากความร้อนของบรรยากาศด้านข้างเตาความร้อน การทดลองนี้จะทำการวัดอุณหภูมิภายนอกเตาที่เป็นอุณหภูมิของบรรยากาศ ทั้งนี้เพื่อตรวจสอบว่าความร้อนภายในเตานั้นส่งออกมาภายนอกเตามากน้อยแค่ไหน ซึ่งจะส่งผลต่อการเลือกใช้งานหัววัดคุณสมบัติของสารที่และอุปกรณ์อื่นๆที่จะนำมาใช้ในการทดลองในงานวิจัยทางวิทยาศาสตร์ โดยในส่วนนี้จะทำการทดลองควบคุมอุณหภูมิของเตาที่ 1000 องศาเซลเซียส เพื่อให้เป็นค่าอุณหภูมิสูงสุดที่จะมีการใช้งานเตาความร้อน หลังจากนั้นวัดอุณหภูมิที่ส่งออกมาภายนอกเตาความร้อน ที่ระยะ 2 เซนติเมตร โดยแสดงการจัดวางเครื่องมือการทดลองดังภาพที่ 3.19



ภาพที่ 3.19 การวัดอุณหภูมิอากาศด้านข้างเตาความร้อน

3.2.5.3 การวัดอุณหภูมิที่ส่งผ่านออกมาจากเตาความร้อนที่บริเวณผิวของเตาความร้อน เพื่อแสดงค่าอุณหภูมิพื้นผิวเตาเพื่อให้สามารถเลือกใช้งานอุปกรณ์ที่จะนำมาเชื่อมต่อภายนอกเตาที่จะมาเชื่อมกับผนังเตาความร้อนให้ตรงตามช่วงอุณหภูมิที่เกิดขึ้นบนผนังเตาความร้อน โดยทำการทดลองควบคุมอุณหภูมิของเตาที่ 1000 องศาเซลเซียส เพื่อแสดงถึงอุณหภูมิสูงสุดภายในเตาที่จะมีการเรียกใช้งานเตาความร้อน จากนั้นวัดอุณหภูมิที่ส่งออกมาภายนอกเตาความร้อนที่บริเวณผนังเตาความร้อน ทั้งนี้จะแบ่งจุดในการวัดอุณหภูมิที่ผิวเตาความร้อนออกเป็น 2 บริเวณ คือการทดลองวัดอุณหภูมิที่ผิวด้านข้างเตาความร้อนและการทดลองวัดอุณหภูมิที่ผิวด้านบนเตาความร้อน แสดงการจัดวางอุปกรณ์ในการทดลองดังภาพที่ 3.20



ภาพที่ 3.20 การวัดอุณหภูมิผิวข้างของเตาความร้อน

บทที่ 4

ผลและวิเคราะห์ผลการวิจัย

4.1 ผลทดสอบระบบควบคุมอุณหภูมิแบบพีไอดีที่อุณหภูมิต่ำกว่า 500 องศาเซลเซียส

การทดลองเพื่อทดสอบระบบควบคุมแบบพีไอดี เป็นการทดลองเพื่อตรวจสอบโปรแกรมควบคุมอุณหภูมิ และทดลองหาค่าตัวแปรพีไอดี ที่เหมาะสมในการควบคุมอุณหภูมิในช่วงอุณหภูมิที่น้อยกว่า 500 องศาเซลเซียส ในการทดลองเพื่อหาตัวแปรควบคุมอุณหภูมิจะใช้สองวิธีคือ วิธีการของ Ziegler-Nichol และอีกหนึ่งวิธีคือ Trial and error และเลือกใช้งานตัวแปรควบคุมแบบพีไอดีที่ให้ผลลัพธ์ที่ดีที่สุด หลังจากนั้นจะนำเสนอผลการควบคุมอุณหภูมิของเตาความร้อนที่อุณหภูมิ 200 300 400 500 องศาเซลเซียส

4.1.1 ผลการทดลองหาค่าตัวแปรควบคุมด้วยวิธีการ Ziegler-Nichol

วิธีการ Ziegler - Nichol เป็นวิธีที่ใช้การทดลองและใช้สูตรคำนวณตัวแปรควบคุมพีไอดี แสดงผลการทดลองด้วยวิธีการของ Ziegler-Nichol ด้วยวิธี Ultimate Cycle method

4.1.1.1 ผลการทดลองปรับค่าตัวแปรควบคุมแบบ P Control ในส่วนแรกของวิธีการ Ultimate Cycle method คือการควบคุมอุณหภูมิแบบ P control และสังเกตผลการควบคุมอุณหภูมิที่เกิดขึ้น แสดงผลการทดลองดังตารางที่ 4.1

ตารางที่ 4.1 ผลการควบคุมอุณหภูมิแบบ P control

Parameter	Settling value	Steady state error (°C)	Overshoot (°C)
$K_p = 150$	อุณหภูมิเกิดการแกว่ง ใต้จุด Setpoint	± 3	2
$K_p = 200$	อุณหภูมิเกิดการแกว่ง รอบจุด Setpoint	± 3	2
$K_p = 250$	อุณหภูมิเกิดการแกว่ง รอบจุด Setpoint	± 2	2
$K_p = 300$	อุณหภูมิเกิดการแกว่ง เหนือจุด Setpoint	± 4	3

4.1.1.2 การคำนวณตัวแปรพีไอดีตามวิธีการของ Ziegler-Nichol จากการทดลองแบบ P control ตามวิธีการทดลองที่ 4.1.1.1 จะสามารถวิเคราะห์ผลและหาค่าตัวแปร K_{cr} เท่ากับ 250 และเมื่อทำการคำนวณหาค่าที่ได้จากการทดลองควบคุมอุณหภูมิจะสามารถหาค่าได้เท่ากับ 1 นั่นคือตัวแปร P_{cr} เท่ากับ 1 หลังจากนั้นทำการคำนวณด้วยวิธีการ Ziegler-Nichol เพื่อหาค่าตัวแปรแบบพีไอดีตามวิธีการต่อไปนี้

ตารางที่ 4.2 สมการคำนวณตัวแปรควบคุมด้วยวิธีการ Ziegler-Nichol

ชนิดของตัวควบคุม	ตัวแปร K_p	ตัวแปร T_i	ตัวแปร T_d
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

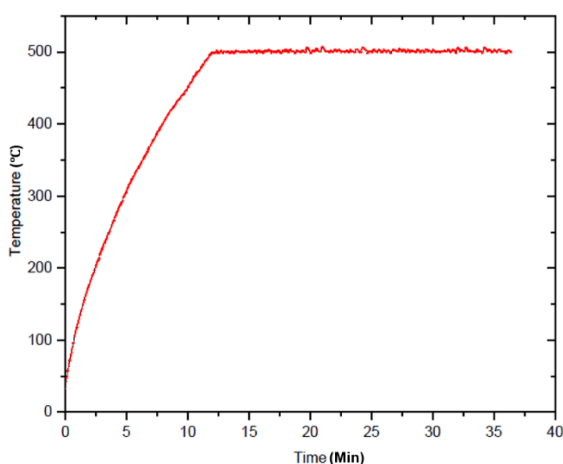
จะสามารถคำนวณหาตัวแปรพีไอดีได้ดังสมการ

$$K_p = 0.6K_{cr} = 0.6(250) = 150$$

$$T_i = 0.5P_{cr} = 0.5(1) = 0.5$$

$$T_d = 0.125P_{cr} = 0.125(1) = 0.125$$

4.1.2.3 การทดลองควบคุมอุณหภูมิด้วยตัวแปรจากวิธีการ Ziegler-Nichol ในส่วนต่อไปจะเป็นทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส หลังจากนั้นจะสังเกตผลการทดลองที่ได้จากการทดลอง



ภาพที่ 4.1 ผลการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส ด้วยตัวแปรควบคุมจาก Ziegler-Nichol

4.1.2 ผลการทดลองหาค่าตัวแปรควบคุมด้วยวิธีการ Trial and error

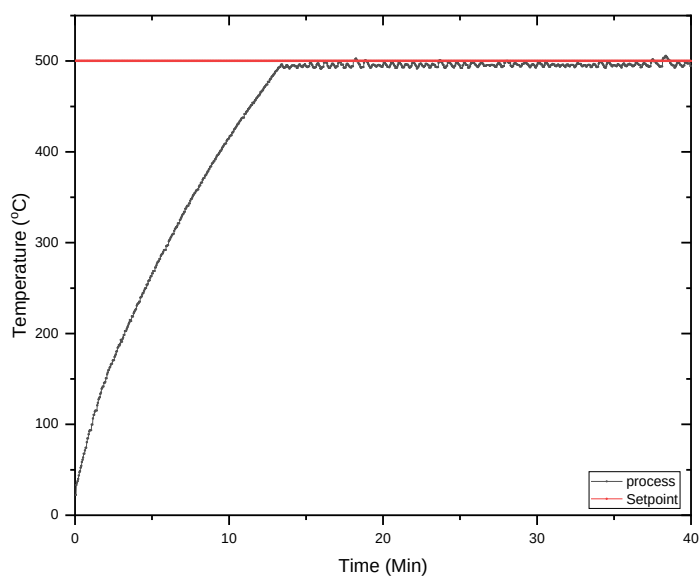
การทดลองด้วยวิธีการ Trial and error นี้ จะแบ่งการทดลองออกเป็น 3 ส่วน เพื่อหาค่าตัวแปรควบคุมแบบสัดส่วน ตัวแปรควบคุมอินทิกรัลและตัวแปรควบคุมอนุพันธ์ ซึ่งมีผลการทดลองดังแสดงตามหัวข้อย่อยดังต่อไปนี้

4.1.2.1 การทดลองเพื่อปรับค่าตัวแปรควบคุมสัดส่วน ในขั้นตอนนี้ทำการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบสัดส่วน (Proportional control) โดยทำการปรับค่าตัวแปรควบคุมแบบสัดส่วน สามารถแสดงผลการทดลองดังตารางที่ (4.1)

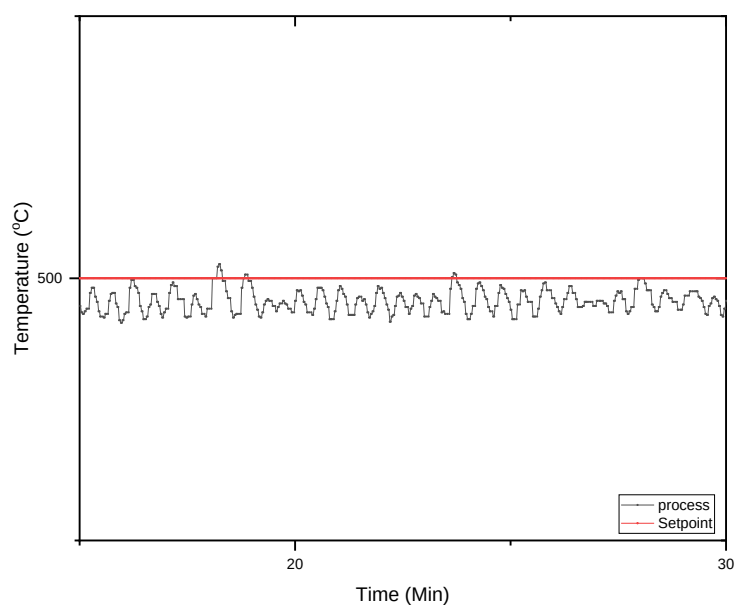
ตารางที่ 4.3 ผลการทดลองการควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบสัดส่วน (Proportional control)

Parameter	Settling value	Steady state error (°C)	Overshoot (°C)
$K_p = 150$	อุณหภูมิเกิดการแกว่งใต้จุด Setpoint	± 3	-
$K_p = 200$	อุณหภูมิเกิดการแกว่งรอบจุด Setpoint	± 3	-
$K_p = 250$	อุณหภูมิเกิดการแกว่งรอบจุด Setpoint	± 2	2
$K_p = 300$	อุณหภูมิเกิดการแกว่งเหนือจุด Setpoint	± 4	3

จากการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบสัดส่วน และทำการปรับค่าตัวแปรแบบสัดส่วนที่ค่าต่าง ๆ ดังตาราง จะได้ค่าคงที่แบบสัดส่วนที่เหมาะสมเท่ากับ 250 เนื่องจากระบบมีการแกว่งแบบ Oscillates รอบอุณหภูมิเป้าหมายด้วยค่าอุณหภูมิ ± 2 องศาเซลเซียสซึ่งเป็นค่า Steady state error ที่น้อยที่สุด แต่ผลการควบคุมแบบสัดส่วนที่ค่าคงที่ 250 นี้ ยังมีค่า Overshoot อยู่ที่ 2 องศาเซลเซียส ซึ่งจะสามารถแก้ได้ด้วยตัวแปรควบคุมของการทดลองปรับค่าตัวแปรควบคุมแบบอินทิกรัลและตัวแปรควบคุมอนุพันธ์ สามารถแสดงกราฟการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียสโดยใช้การควบคุมแบบสัดส่วนได้ดังภาพต่อไปนี้

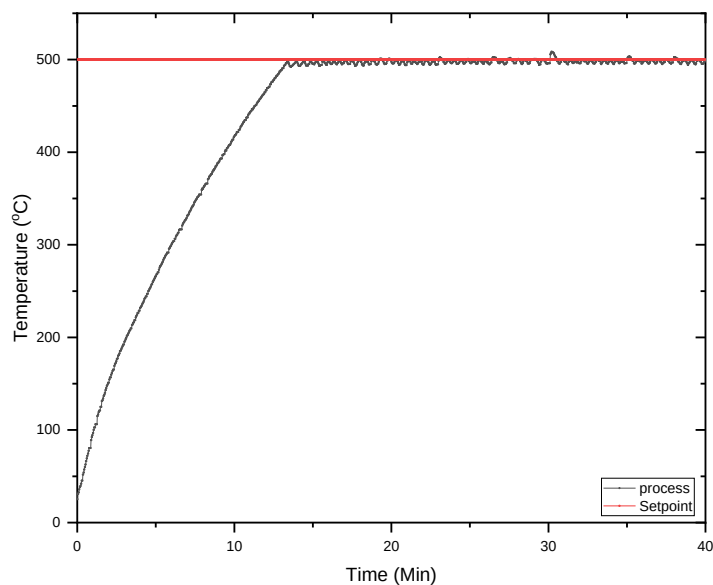


ภาพที่ 4.2 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 150

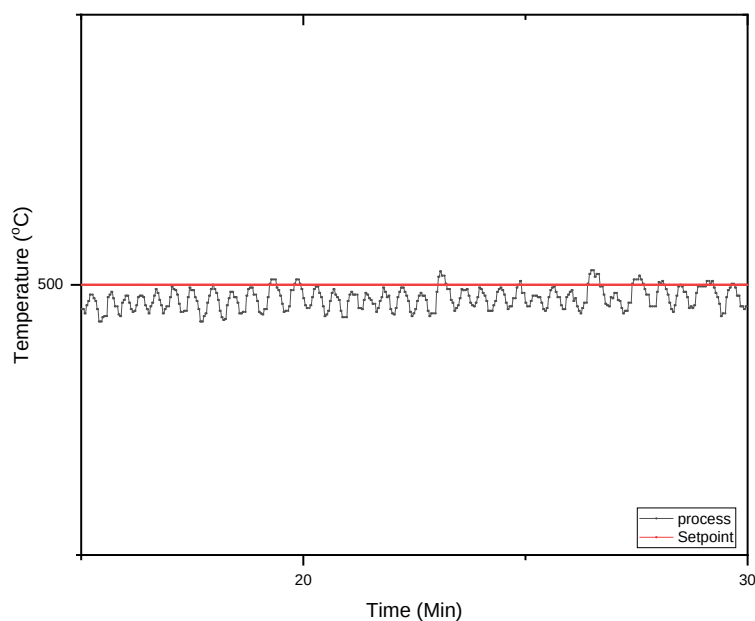


ภาพที่ 4.3 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 150

จากภาพที่ 4.3 พบว่าระบบที่ได้จากการควบคุมอุณหภูมิด้วยตัวแปร K_p 150 มีลักษณะการแกว่งของสัญญาณอยู่ที่จุด setpoint โดยมีบางช่วงของสัญญาณที่มีค่าความผิดพลาดเลยขึ้นมาเหนือจุด setpoint บ้าง ตามภาพที่ 4.2 ซึ่งไม่ตรงตามความต้องการของเราที่ต้องการการแกว่งอยู่ระหว่างจุด setpoint

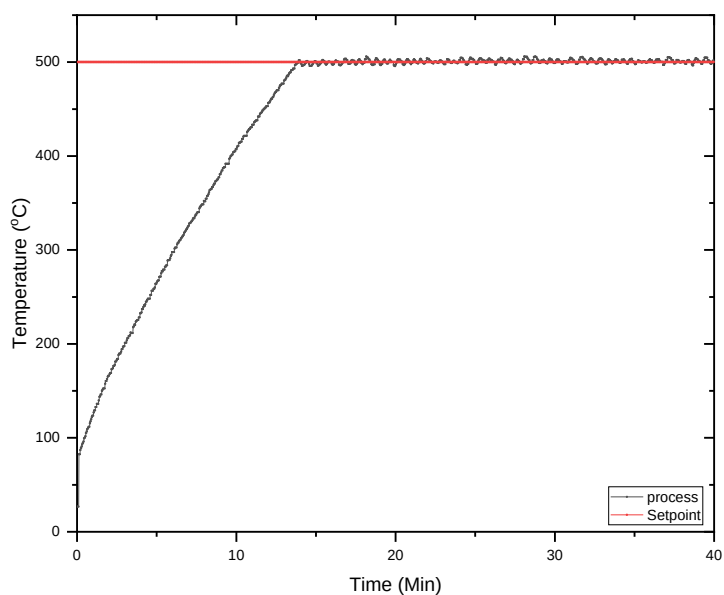


ภาพที่ 4.4 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 200

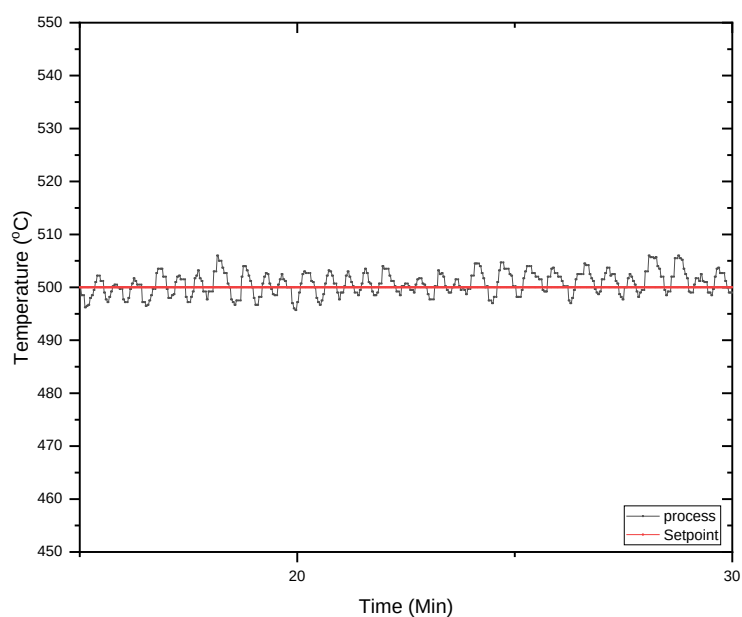


ภาพที่ 4.5 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 200

จากภาพที่ 4.5 ระบบที่เกิดจากการควบคุมด้วยค่า K_p 200 มีการแกว่งของสัญญาณควบคุม อยู่ได้จุด setpoint เล็กน้อย และเช่นเคยคือ มีค่าความผิดพลาดของสัญญาณในช่วงที่มีการแกว่ง ขึ้นมาเหนือจุด setpoint บ้างเล็กน้อย

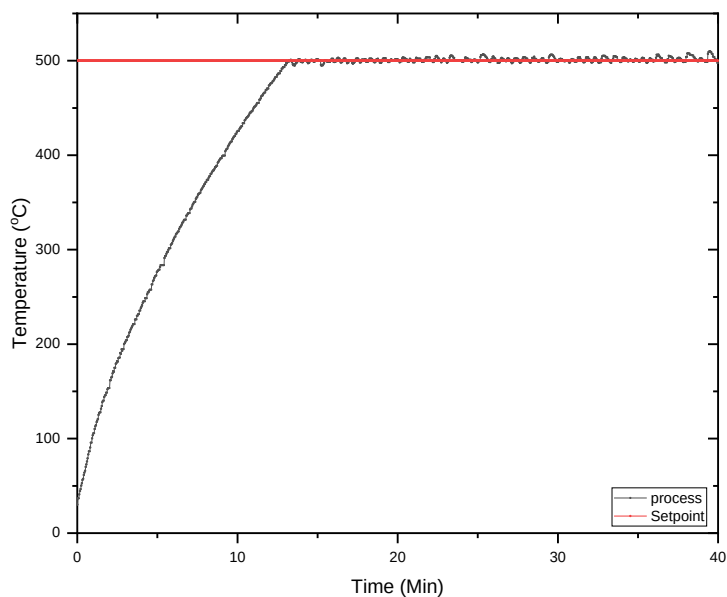


ภาพที่ 4.6 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 250

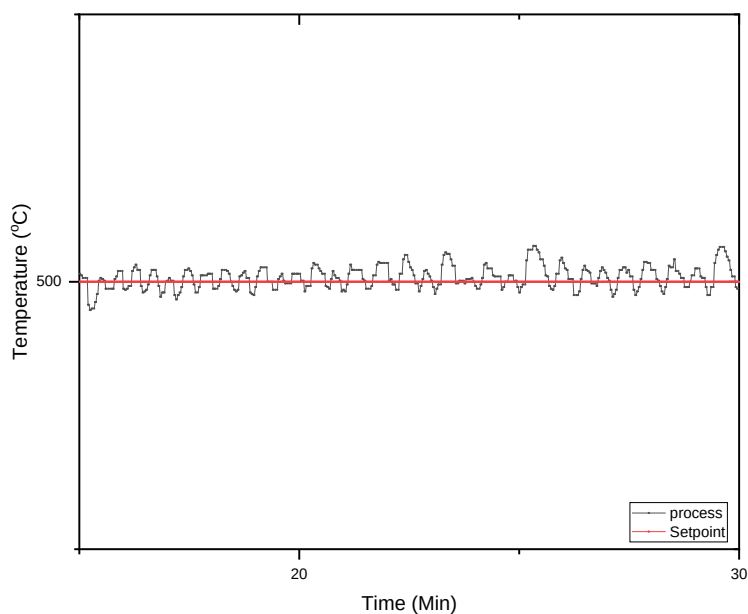


ภาพที่ 4.7 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 250

จากภาพที่ 4.7 ระบบที่เกิดจากการควบคุมด้วยตัวแปร K_p 250 มีการแกว่งของสัญญาณรอบจุด setpoint โดยมีความผิดพลาดของระบบบางช่วงที่มีค่าเกินถึง 3 องศาเซลเซียส แต่โดยเฉลี่ยแล้วระบบมีการแกว่งรอบจุด setpoint ที่ ± 2 องศาเซลเซียส ทั้งนี้ระบบสามารถควบคุมอุณหภูมิให้มีค่า Overshoot เกิดขึ้นเท่ากับ 2 องศาเซลเซียส



ภาพที่ 4.8 กราฟการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 300



ภาพที่ 4.9 กราฟขยายการปรับค่าตัวแปรควบคุมแบบสัดส่วนที่ K_p 300

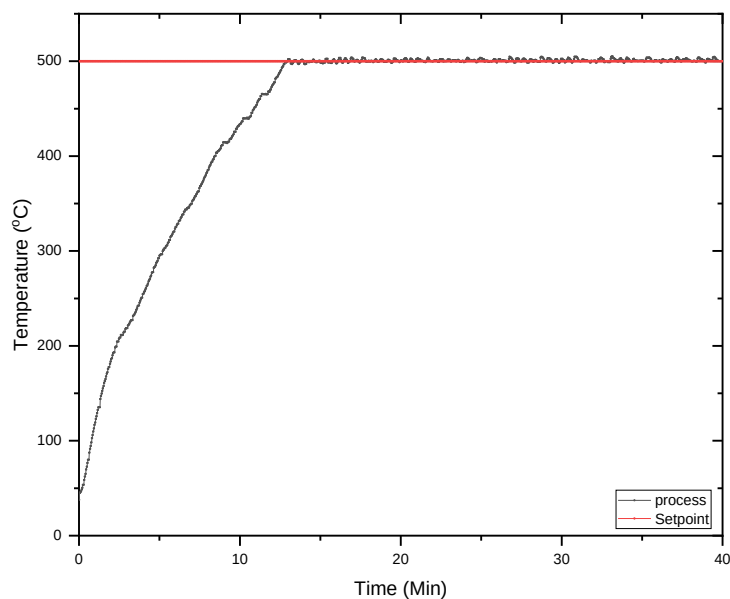
จากภาพที่ 4.9 ระบบที่ได้จากการควบคุมอุณหภูมิด้วยตัวแปร K_p 300 พบว่ามีการแกว่งของสัญญาณเหนือจุด setpoint ทั้งนี้ระบบควบคุมสามารถควบคุมอุณหภูมิโดยมีค่าความผิดพลาดเหนือจุด setpoint สูงสุดอยู่ที่ 3 องศาเซลเซียส และมีค่า Overshoot เกิดขึ้นที่ 3 องศาเซลเซียส เช่นเดียวกัน

4.1.2.2 การทดลองเพื่อปรับค่าตัวแปรควบคุมอินทิกรัล ในขั้นตอนนี้ทำการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PI control โดยมีการกำหนดค่าตัวแปรพีไอดีเพียง 2 ตัวแปร ได้แก่ตัวแปรควบคุมสัดส่วน และตัวแปรควบคุมอินทิกรัล โดยในการทดลองนี้จะทำการปรับค่าตัวแปรควบคุมแบบอินทิกรัล เพื่อหาสัญญาณควบคุมของระบบที่ดีที่สุด โดยใช้ค่าตัวแปรแบบสัดส่วนตามการทดลองปรับค่าแบบสัดส่วนข้างต้น ส่วนตัวแปรควบคุมอนุพันธ์นั้นจะถูกกำหนดให้มีค่าเท่ากับ 0 แสดงผลการทดลองเพื่อปรับค่าตัวแปรควบคุมอินทิกรัล ดังตารางที่ 4.2

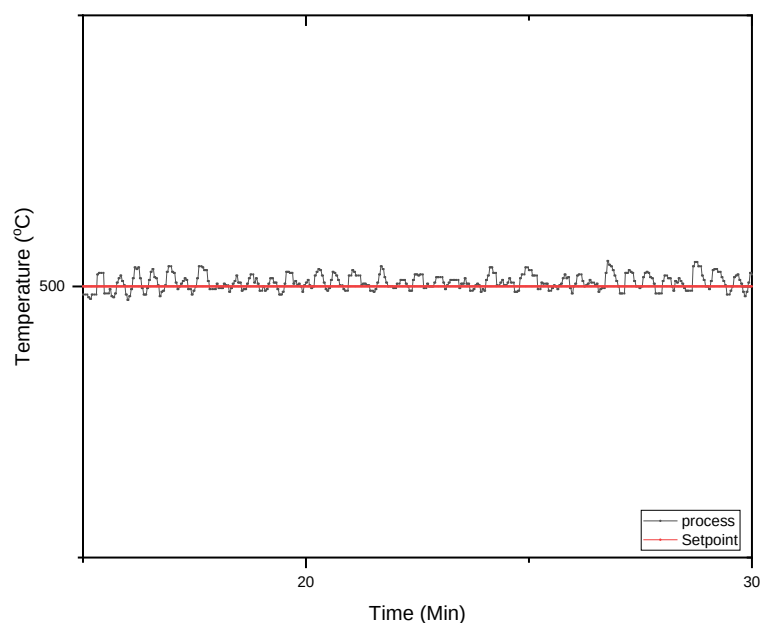
ตารางที่ 4.4 ผลการทดลองการควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PI control

Parameter	Rising time(นาทื)	Steady state error (°c)	Overshoot (°c)
$T_i = 0.05$	13	± 1	2
$T_i = 0.10$	11	± 2	2
$T_i = 0.15$	10	± 2	3
$T_i = 0.20$	11	± 3	3

จากการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบPI control และทำการปรับค่าตัวแปรอินทิกรัลที่ค่าต่าง ๆ ดังตาราง จะได้ค่าคงที่แบบอินทิกรัลที่เหมาะสมเท่ากับ 0.05 เนื่องจากระบบมีค่าความผิดพลาดของระบบอยู่ที่ ± 1 องศาเซลเซียส ซึ่งเป็นค่าที่น้อยที่สุด ทั้งนี้ระบบควบคุมยังคงมีค่า Overshoot เท่ากับ 2 องศาเซลเซียส และเนื่องจากตัวแปรอินทิกรัลมีผลต่อการตอบสนองของระบบทำให้มีค่า Rising time อยู่ที่ 13 นาที่ ดังนั้นจะทำการแก้ไขโดยการเพิ่มตัวแปรควบคุมอนุพันธ์เสริมเข้ามาในการทดลองต่อไป แสดงกราฟการปรับค่าตัวแปรควบคุมของอินทิกรัลที่อุณหภูมิ 500 องศาเซลเซียสได้ดังภาพต่อไปนี้

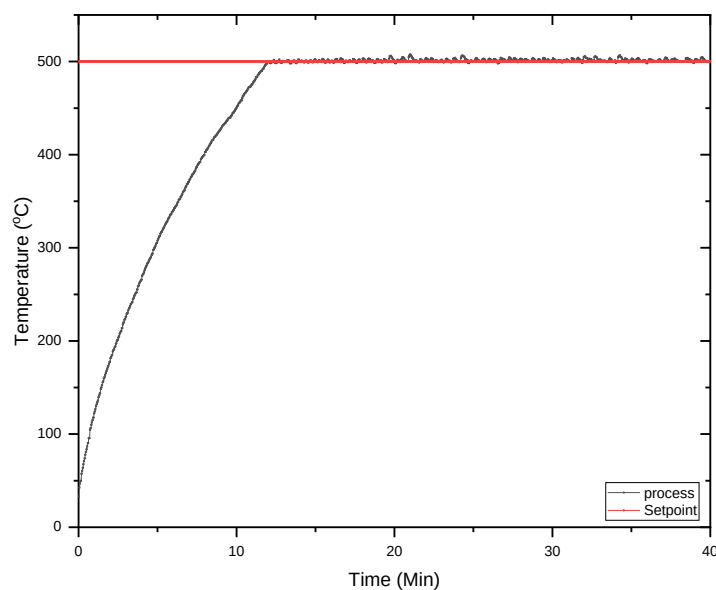


ภาพที่ 4.10 กราฟการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.05

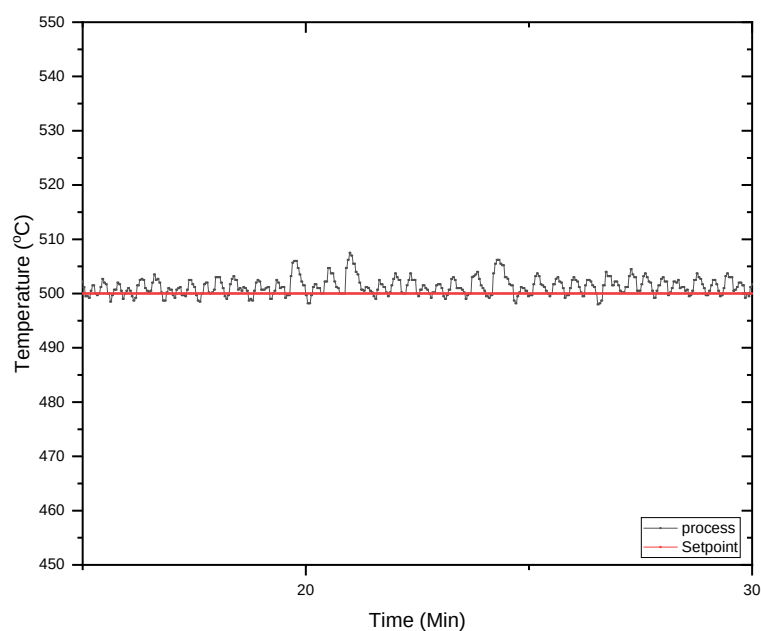


ภาพที่ 4.11 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.05

จากภาพที่ 4.11 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_i 0.05 พบว่าระบบสามารถควบคุมอุณหภูมิให้มีค่าความผิดพลาดน้อยที่สุดคือ 1 องศาเซลเซียส ทั้งนี้ระบบควบคุมสามารถควบคุมอุณหภูมิให้มีค่า Rising time อยู่ที่ 13 นาทีและยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 2 องศาเซลเซียส

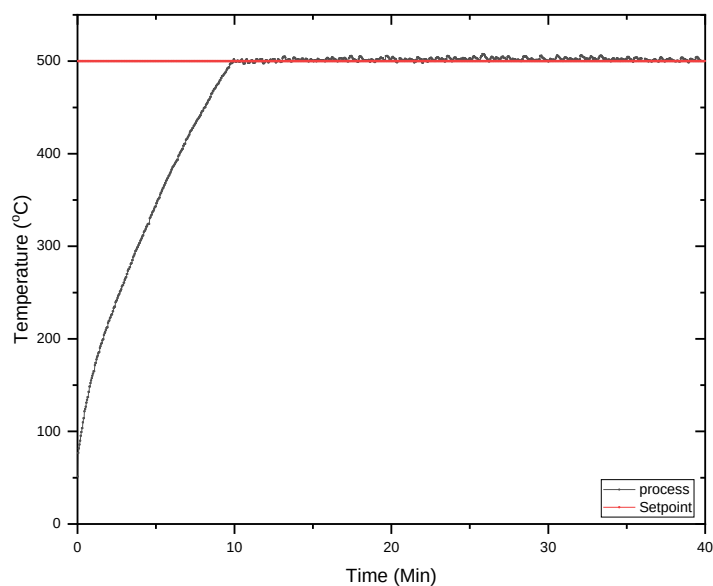


ภาพที่ 4.12 กราฟการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.10

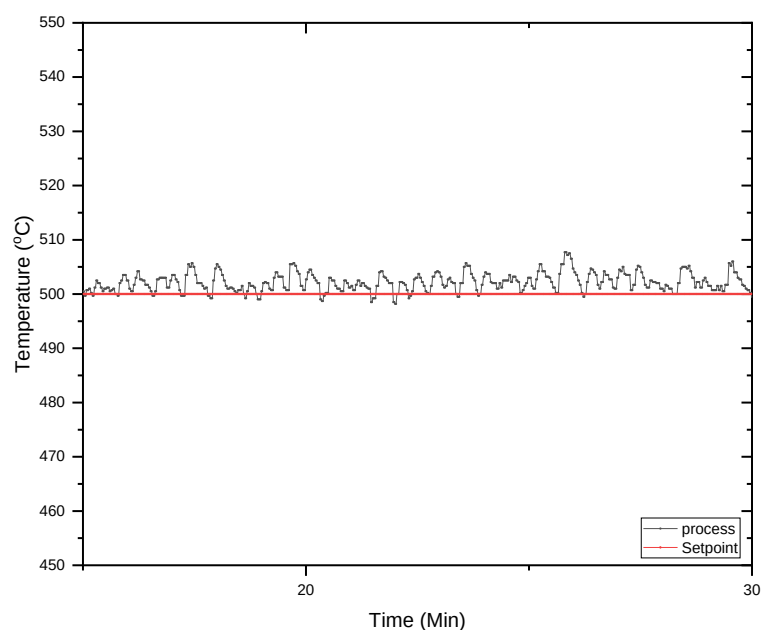


ภาพที่ 4.13 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.10

จากภาพที่ 4.13 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_i 0.10 พบว่าระบบควบคุมอุณหภูมิสามารถควบคุมให้มีค่าความผิดพลาดน้อยที่สุดคือ 2 องศาเซลเซียส ทั้งนี้ระบบควบคุมสามารถควบคุมอุณหภูมิให้มีค่า Rising time อยู่ที่ 11 นาที และยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 2 องศาเซลเซียส

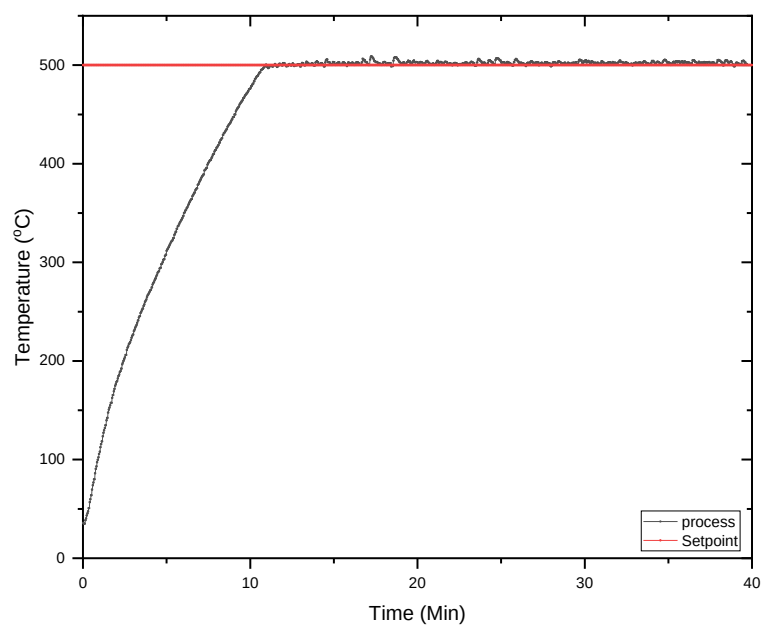


ภาพที่ 4.14 กราฟการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.15

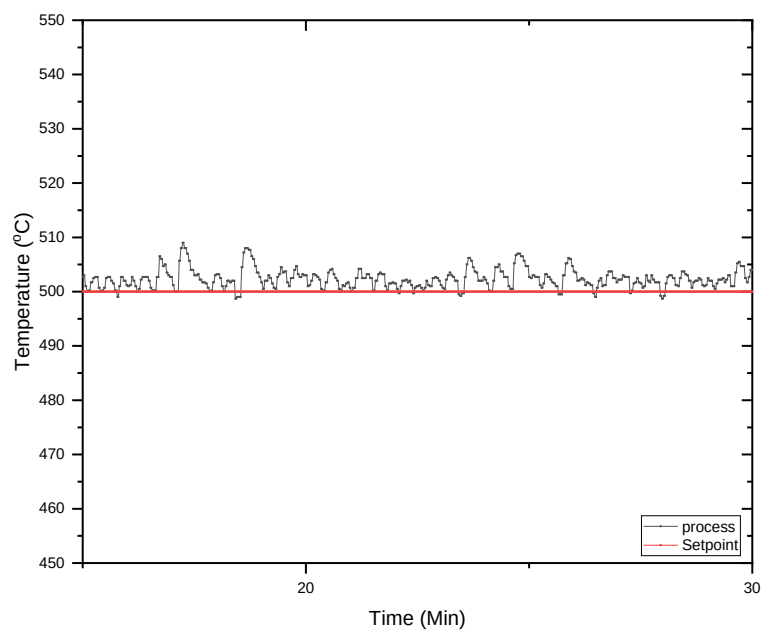


ภาพที่ 4.15 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.15

จากภาพที่ 4.15 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_i 0.15 พบว่าระบบควบคุมสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดน้อยที่สุดคือ 2 องศาเซลเซียส ทั้งนี้ระบบควบคุมสามารถควบคุมอุณหภูมิให้มีค่า Rising time อยู่ที่ 10 นาที และยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 3 องศาเซลเซียส



ภาพที่ 4.16 กราฟการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.20



ภาพที่ 4.17 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอินทิกรัลที่ T_i 0.20

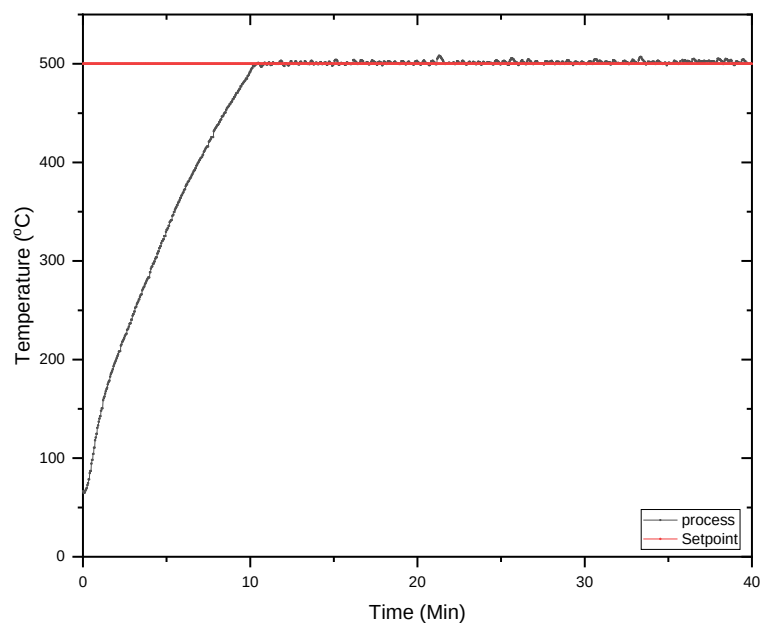
จากภาพที่ 4.17 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_i 0.20 พบว่าระบบมีค่าความผิดพลาดน้อยที่สุดคือ 3 องศาเซลเซียส ทั้งนี้ระบบได้มีค่า Rising time อยู่ที่ 11 นาที และยังคงค่า Overshoot เอาไว้อยู่ที่ 3 องศาเซลเซียส

4.2.1.3 การทดลองเพื่อปรับค่าตัวแปรควบคุมแบบอนุพันธ์ ในขั้นตอนนี้ทำการทดลองควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PID control จากนั้นทำการปรับตัวแปรควบคุมแบบอนุพันธ์ โดยใช้ค่าตัวแปรแบบสัดส่วนตามการทดลองปรับค่าแบบสัดส่วน และใช้ตัวแปรแบบอินทิกรัลจากการทดลองแบบอินทิกรัล จากนั้นปรับค่าตัวแปรอนุพันธ์ที่ค่าต่างๆ แสดงผลการทดลองเพื่อปรับค่าตัวแปรควบคุมอนุพันธ์ดังตารางที่ 4.3

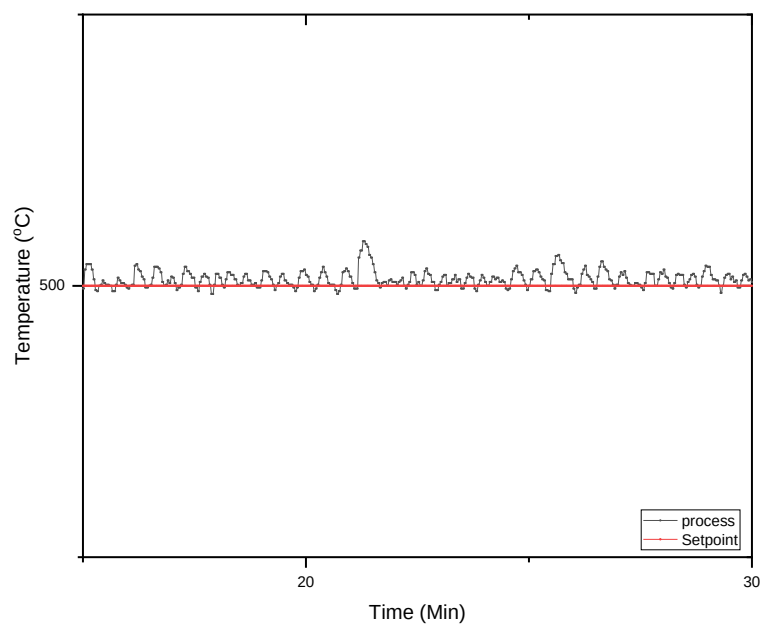
ตารางที่ 4.5 ผลการทดลองการควบคุมอุณหภูมิที่ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PID control

Parameter	Rising time (นาทีก)	Steady state error(°C)	Overshoot (°C)
$T_d = 0.05$	10	± 2	2
$T_d = 0.10$	11	± 2	3
$T_d = 0.15$	14	± 1	3
$T_d = 0.20$	12	± 2	1

จากการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส โดยใช้ระบบควบคุมแบบ PID control และทำการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ค่าต่าง ๆ ดังตาราง จะได้ค่าตัวแปรควบคุมแบบอนุพันธ์ ที่เหมาะสมเท่ากับ 0.20 ทั้งนี้ระบบควบคุมยังสามารถควบคุมอุณหภูมิของเตาความร้อนโดยมีค่าความผิดพลาดของระบบอยู่ที่ ± 1 องศาเซลเซียส ซึ่งเป็นค่าความผิดพลาดที่น้อยที่สุดเมื่อเปรียบเทียบกับค่าคงที่อนุพันธ์ค่าอื่น ๆ และสามารถลดค่า Overshoot ของระบบให้เหลือ 1 องศาเซลเซียสดังจะเห็นว่าเป็นค่าที่น้อยที่สุดจากตัวแปรอนุพันธ์ทั้งหมดเช่นเดียวกัน ทั้งนี้ระบบมีค่า Rising time อยู่ที่ 22 นาทีก โดยสามารถแสดงกราฟจากการทดลองเพื่อหาค่าตัวแปรอนุพันธ์ที่อุณหภูมิ 500 องศาเซลเซียสของค่าตัวแปรอนุพันธ์ต่าง ๆ ได้ดังภาพต่อไปนี้

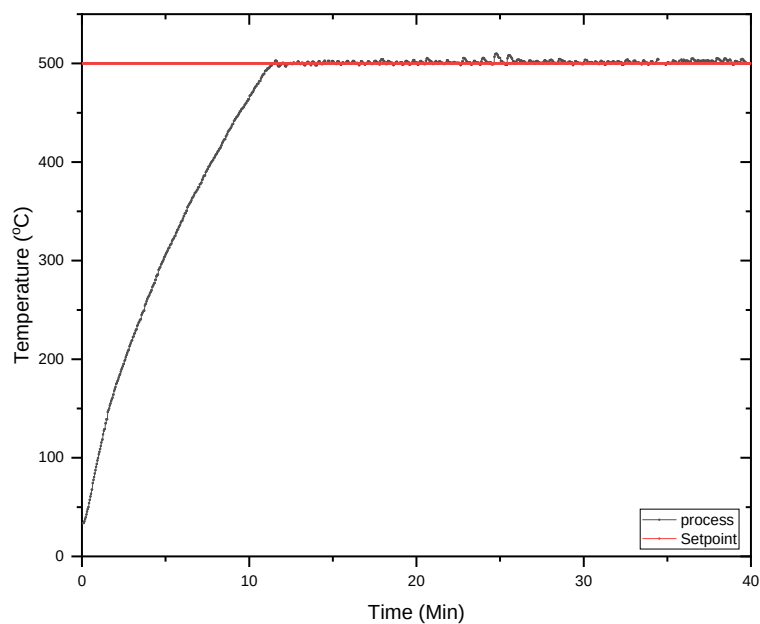


ภาพที่ 4.18 กราฟการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.05

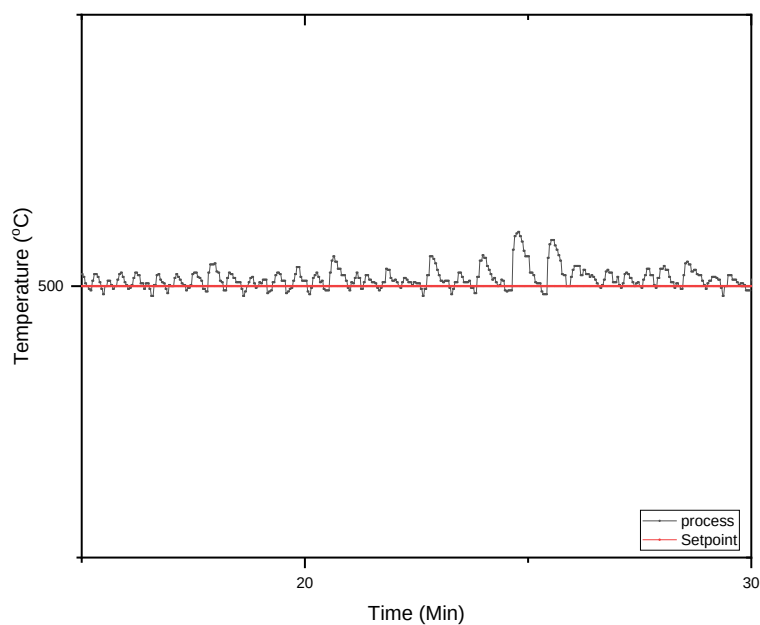


ภาพที่ 4.19 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.05

จากภาพที่ 4.19 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_d 0.05 พบว่าระบบมีค่าความผิดพลาดน้อยที่สุดคือ 1 องศาเซลเซียส ทั้งนี้ระบบควบคุมได้มีค่า Rising time อยู่ที่ 10 นาที และยังคงค่า Overshoot ของอุณหภูมิอยู่ที่ 2 องศาเซลเซียส

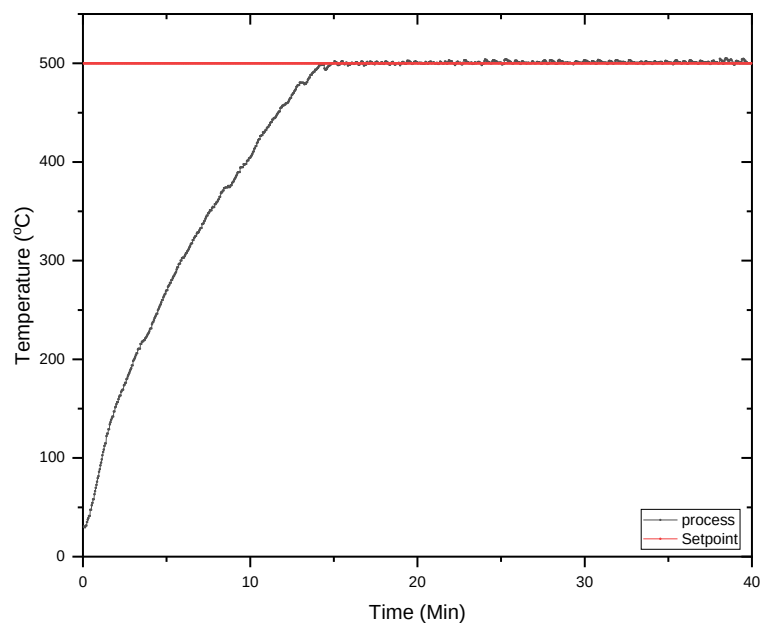


ภาพที่ 4.20 กราฟการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.10

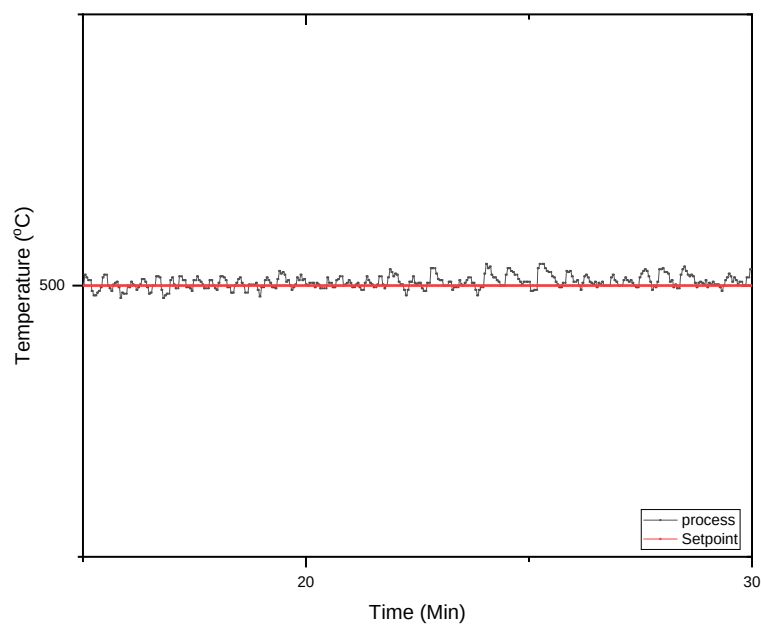


ภาพที่ 4.21 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.10

จากภาพที่ 4.20 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_d 0.10 พบว่าระบบสามารถควบคุมอุณหภูมิให้มีค่าความผิดพลาดน้อยที่สุดคือ 1 องศาเซลเซียส ทั้งนี้ระบบได้มีค่า Rising time อยู่ที่ 11 นาที และยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 3 องศาเซลเซียส

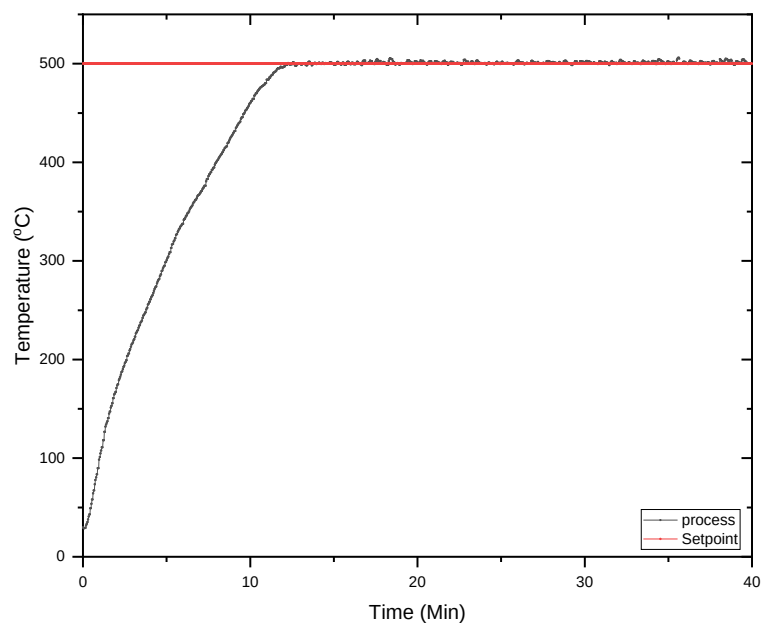


ภาพที่ 4.22 กราฟการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.15

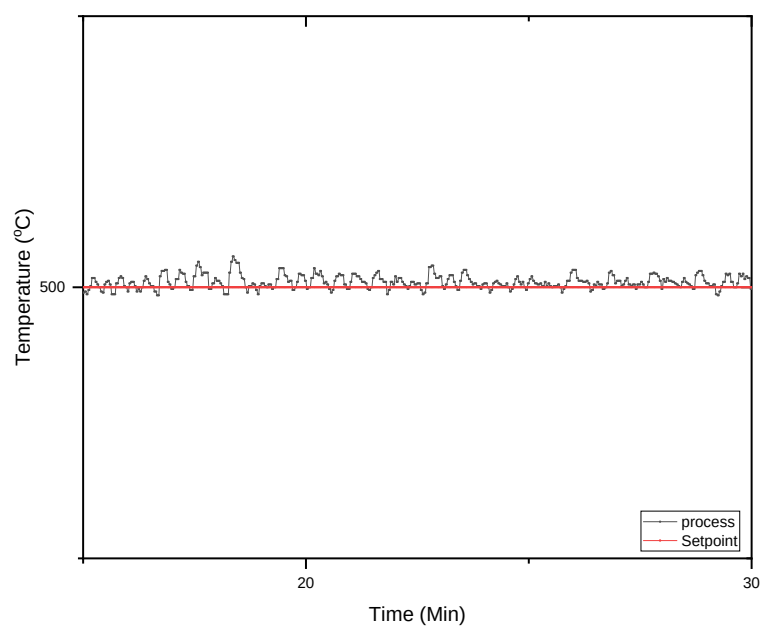


ภาพที่ 4.23 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.15

จากภาพที่ 4.23 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_d 0.15 พบว่าระบบสามารถควบคุมอุณหภูมิให้มีค่าความผิดพลาดน้อยที่สุดคือ 1 องศาเซลเซียส ทั้งนี้ระบบได้มีค่า Rising time อยู่ที่ 14 นาที และยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 3 องศาเซลเซียส



ภาพที่ 4.24 กราฟการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.20

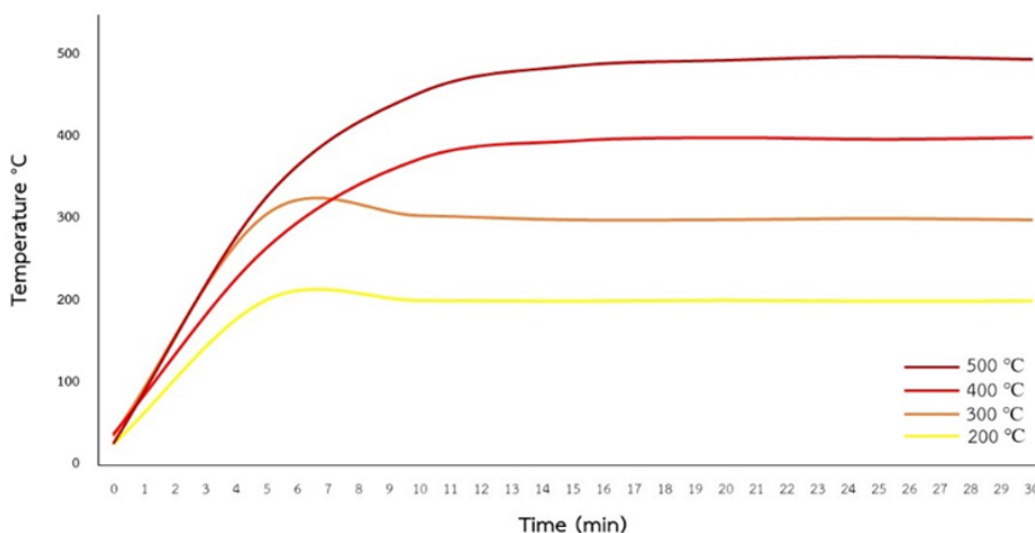


ภาพที่ 4.25 กราฟขยายการปรับค่าตัวแปรควบคุมแบบอนุพันธ์ที่ T_d 0.20

จากภาพที่ 4.25 แสดงกราฟการควบคุมอุณหภูมิที่ได้จากการปรับค่าตัวแปร T_d 0.20 พบว่าระบบสามารถควบคุมอุณหภูมิให้มีค่าความผิดพลาดน้อยที่สุดคือ 1 องศาเซลเซียส ทั้งนี้ระบบได้มีค่า Rising time อยู่ที่ 12 นาที และยังคงเกิดค่า Overshoot ของอุณหภูมิอยู่ที่ 1 องศาเซลเซียส

4.1.3 ผลการทดลองควบคุมอุณหภูมิในช่วงต่ำกว่า 500 องศาเซลเซียส

ผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิต่ำกว่า 500 องศาเซลเซียส ในอุณหภูมิ 200 300 400 และ 500 องศาเซลเซียส สามารถแสดงผลการทดลองควบคุมอุณหภูมิในช่วงดังกล่าว ดังภาพที่ 4.26



ภาพที่ 4.26 ผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิต่ำกว่า 500 องศาเซลเซียส

จากภาพที่ 4.26 แสดงผลการทดลองควบคุมอุณหภูมิในช่วงต่ำกว่า 500 องศาเซลเซียส พบว่าระบบควบคุมสามารถควบคุมอุณหภูมิโดยมีค่าความคลาดเคลื่อนดังต่อไปนี้

4.1.3.1 ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 200 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 10 นาที โดยมีค่า Overshoot 10 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 15 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 3 องศาเซลเซียส

4.1.3.2 ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 300 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 15 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

4.1.3.3 ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 400 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 20 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

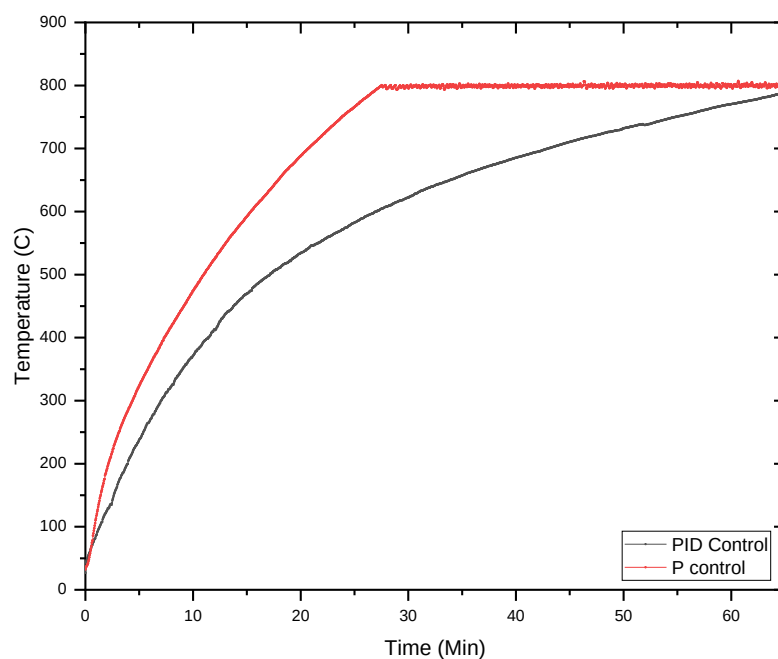
4.1.3.4 ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_d = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 25 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

4.2 ผลการทดลองหาตัวแปรควบคุมอุณหภูมิในช่วงอุณหภูมิมากกว่า 500 องศาเซลเซียส

การทดลองเพื่อทดสอบระบบควบคุมแบบพีไอดี เป็นการทดลองเพื่อตรวจสอบโปรแกรมควบคุมอุณหภูมิ และทดลองหาค่าตัวแปรพีไอดีที่เหมาะสมในการควบคุมอุณหภูมิในช่วงมากกว่า 500 องศาเซลเซียส ในการทดลองเพื่อหาตัวแปรควบคุมอุณหภูมิใช้วิธีการ Trial and error หลังจากนั้นจะนำเสนอผลการควบคุมอุณหภูมิที่ใช้ตัวแปรควบคุมจากวิธีการ Trial and error ที่อุณหภูมิ 600 700 800 900 และ 1000 องศาเซลเซียส

4.2.1 การเปรียบเทียบตัวแปรระบบควบคุม

ในส่วนนี้จะทำการเปรียบเทียบระบบควบคุมอุณหภูมิที่ใช้งานตัวแปรควบคุมอุณหภูมิที่ใช้งานค่า K_p T_i T_d ทั้งสามตัวแปร และระบบที่ทำการทดลองโดยใช้งานตัวแปรควบคุมที่ใช้งานค่า K_p เพียงตัวแปรเดียว ทั้งนี้ได้ทำการทดลองในช่วงอุณหภูมิ 1000 องศาเซลเซียส ซึ่งเป็นอุณหภูมิสูงสุดของการใช้งานเตาความร้อน

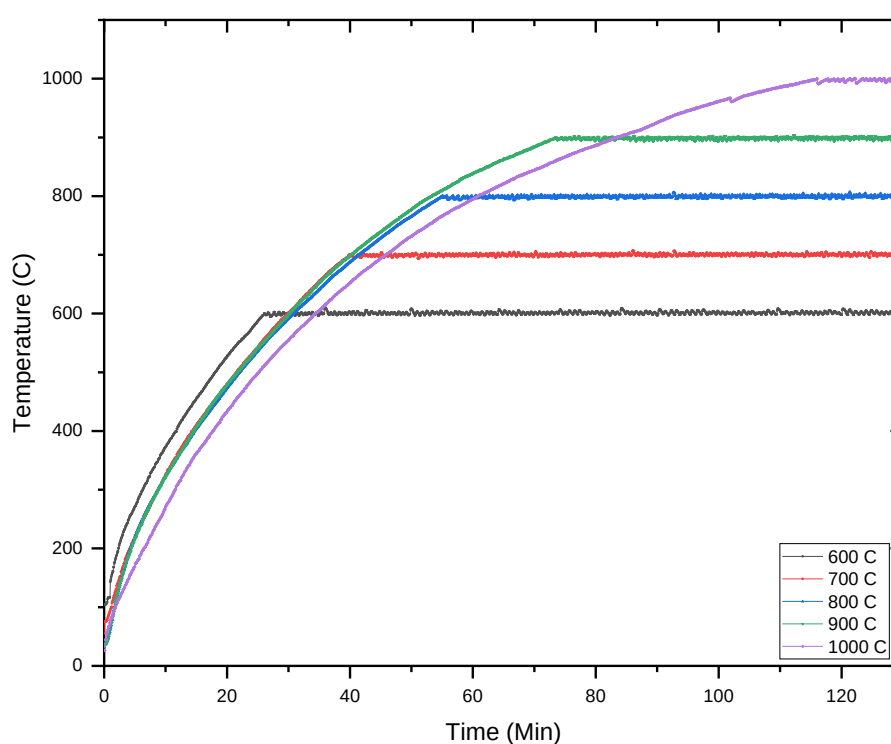


ภาพที่ 4.27 การเปรียบเทียบตัวแปรควบคุมอุณหภูมิ

จากการเปรียบเทียบการใช้งานตัวแปรควบคุมในช่วงอุณหภูมิสูงพบว่า ในกรณีที่ทำการทดลองโดยใช้ตัวแปรพีเอ็ดี ครบทั้งสามตัวแปรคือ K_p T_i T_d ระบบเข้าใช้เวลาในการเพิ่มอุณหภูมิเข้าสู่จุด setpoint มากเกินไป เนื่องจากตัวแปร T_i และ T_d จะคอยชะลอการเพิ่มของอุณหภูมิทำให้เข้าช่วง setpoint นาน และในกรณีที่ทำการทดลองโดยใช้งานตัวแปรควบคุม K_p เพียงค่าเดียว โดยปรับค่าตัวแปร T_i และ T_d เป็น 0 พบว่าระบบเข้าสู่ช่วง setpoint เร็วกว่าในกรณีแรก

4.2.2 การทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง

ผลการทดลองควบคุมอุณหภูมิที่ 1000 องศาเซลเซียส โดยใช้ค่าตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_d = 0$ ในการทำการทดลอง เนื่องจากที่ตัวแปร K_p 450 อุณหภูมิมีการเข้าสู่ setpoint เร็วที่สุด โดยที่สามารถควบคุมอุณหภูมิได้โดยมีความคลาดเคลื่อนน้อย โดยที่อุณหภูมิสูงขึ้น เวลาที่ใช้ในการเข้าสู่สภาวะสมดุลจะมากขึ้น สามารถแสดงผลการทดลองได้ดังภาพที่ 4.28



ภาพที่ 4.28 ผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง

จากภาพที่ 4.28 แสดงกราฟผลการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง ซึ่งในแต่ละช่วงจะใช้เวลาในการเพิ่มอุณหภูมิที่แตกต่างกัน ค่าความผิดพลาดของระบบควบคุมที่เกิดขึ้นโดยสามารถสรุปผลการทดลองในช่วงนี้ได้ดังนี้

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 600 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 25 นาที โดยมีค่า Overshoot ของอุณหภูมิที่ 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 700 องศาเซลเซียส จากการทดลองใช้ตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 40 นาที โดยมีค่า Overshoot ของอุณหภูมิที่ 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

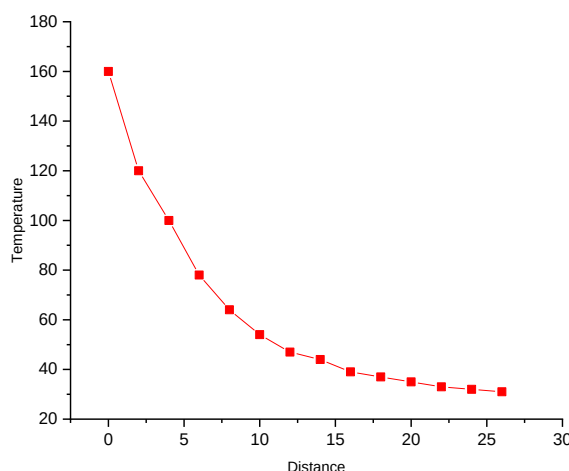
ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 800 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 55 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 900 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 70 นาที โดยมีค่า Overshoot ของอุณหภูมิที่ 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 1000 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 450$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 120 นาที โดยมีค่า Overshoot ของอุณหภูมิที่ 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมีค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

4.3 ผลการทดลองวัดอุณหภูมิภายนอก Block heater

จากการทดลองควบคุมอุณหภูมิที่ 1000 องศาเซลเซียส ผู้วิจัยได้ทำการทดลองวัดอุณหภูมิภายนอกของ Block heater เพื่อทดสอบการป้องกันความร้อนที่ออกมาจาก Block heater สู่อากาศภายนอกเตาความร้อน ผลการทดลองวัดความร้อนภายนอก Block heater ที่จุดต่าง ๆ แสดงผลดังภาพที่ 4.29



ภาพที่ 4.29 ผลการทดลองการวัดความร้อนภายนอก Block heater ที่จุดต่าง ๆ

4.4 ผลการทดลองวัดความร้อนภายนอกเตาความร้อน

จากการทดลองควบคุมอุณหภูมิที่ 1000 องศาเซลเซียส ผู้วิจัยได้ทำการทดลองวัดอุณหภูมิภายนอกของเตาความร้อนซึ่งมีการทดลอง 2 ส่วน ในส่วนแรกเป็นการวัดอุณหภูมิที่พื้นผิวบริเวณผิวข้างของเตาความร้อนและส่วนต่อมาเป็นการทดลองวัดอุณหภูมิจากความร้อนที่ส่งผ่านมาทางอากาศบริเวณด้านข้างเตาความร้อนมีผลการทดลองดังนี้

4.4.1 การทดลองวัดอุณหภูมิที่พื้นผิวของเตาความร้อน

ในการวัดอุณหภูมิภายนอกเตาความร้อนในหัวข้อนี้เป็นการวัดอุณหภูมิที่บริเวณผิวของวัสดุที่สัมผัสความร้อน โดยแบ่งจุดในการวัดอุณหภูมิออกเป็น 2 จุด คือ การวัดอุณหภูมิพื้นผิวของเตาความร้อนบริเวณผิวข้างของเตาความร้อน และการวัดอุณหภูมิพื้นผิวของเตาความร้อนบริเวณผิวด้านบนของเตาความร้อน

4.4.1.1 การวัดอุณหภูมิพื้นผิวของเตาความร้อนบริเวณผิวข้างของเตาความร้อน จากการทดลองควบคุมอุณหภูมิภายในเตาความร้อนที่ 1000 องศาเซลเซียส พบว่ามีอุณหภูมิที่พื้นผิวของเตาความร้อนบริเวณผิวด้านข้างของเตาความร้อนอยู่ที่ 55 องศาเซลเซียส

4.4.1.2 การวัดอุณหภูมิพื้นผิวของเตาความร้อนบริเวณผิวด้านบนของเตาความร้อน จากการควบคุมอุณหภูมิภายในเตาความร้อนที่ 1000 องศาเซลเซียส พบว่าอุณหภูมิที่ส่งออกมายังพื้นผิวของเตาความร้อนบริเวณส่วนด้านบนเตาความร้อนอยู่ที่ 75 องศาเซลเซียส

4.4.2 การทดลองวัดอุณหภูมิที่ส่งผ่านมาทางอากาศ

ในการวัดอุณหภูมิภายนอกเตาความร้อนในหัวข้อนี้จะแสดงผลการทดลองที่ได้จากการวัดอุณหภูมิที่ส่งผ่านความร้อนมาทางอากาศ โดยแบ่งจุดในการวัดอุณหภูมิออกเป็น 2 จุดวัด ในส่วนแรกคือการวัดอุณหภูมิจากการพาความร้อนบริเวณด้านข้างเตาความร้อน และการวัดอุณหภูมิจากการพาความร้อนบริเวณช่องทางเดิน ของรังสีเอ็กซ์

4.4.2.1 การวัดอุณหภูมิที่ส่งผ่านทางอากาศบริเวณด้านข้างเตาความร้อน จากการทดลองควบคุมอุณหภูมิภายในเตาความร้อนที่ 1000 องศาเซลเซียส สามารถวัดอุณหภูมิที่ส่งผ่านทางอากาศบริเวณผิวข้างของเตาความร้อนได้ 30 องศาเซลเซียส

4.4.2.2 การวัดอุณหภูมิที่ส่งผ่านทางอากาศบริเวณช่องทางเดินของรังสีเอ็กซ์ จากการทดลองควบคุมอุณหภูมิภายในเตาความร้อนที่ 1000 องศาเซลเซียส สามารถวัดอุณหภูมิที่ส่งผ่านทางอากาศช่องทางเดินของรังสีเอ็กซ์ได้ 40 องศาเซลเซียส

บทที่ 5

สรุปผลการวิจัย

5.1 สรุปผลการวิจัย

จากการทดลองและออกแบบสร้างเตาความร้อนในส่วนของการกำเนิดความร้อนในระบบ การออกแบบส่วนป้องกันการสูญเสียพลังงานความร้อนภายในเตา การทดลองเพื่อหาตัวแปรควบคุม อุณหภูมิแบบพีไอดี และผลของการทดลองควบคุมอุณหภูมิในระบบ สามารถสรุปผลการทดลองได้ดังนี้

ในส่วนแรกคือ การออกแบบในส่วนของการกำเนิดความร้อนภายในระบบ ในงานวิจัย ได้ออกแบบส่วนกำเนิดความร้อนภายในเตาความร้อนโดยใช้การจ่ายพลังงานไฟฟ้า ให้กับขดลวดความร้อน จากการทดลองได้ใช้ขดลวด 2 เส้นต่ออนุกรมกัน แต่ละเส้นมีขนาดความต้านทาน 12 โอห์ม และมีกำลังไฟฟ้า 1000 วัตต์ เมื่อต่ออนุกรมกันแล้วจะได้ค่าความต้านทานของขดลวดความร้อนรวม 24 โอห์ม กำลังไฟฟ้า 2000 วัตต์ ซึ่งจากการทดลองควบคุมอุณหภูมิโดยมีแกรไฟต์เป็นตัวกักเก็บพลังงานความร้อนภายในระบบ สามารถเพิ่มอุณหภูมิขึ้นไปได้ถึง 1000 องศาเซลเซียส และมีแนวโน้มว่าสามารถเพิ่มขึ้นไปอีกถึง 1500 องศาเซลเซียส

ในงานวิจัยได้ออกแบบและสร้าง Block heater ซึ่งทำจากการหล่อปูนทนไฟ โดยทำการวาง Block heater สองชั้น ซึ่งแต่ละชั้นมีลักษณะเป็นทรงกระบอกกลวง มีขนาดเส้นผ่านศูนย์กลาง 120 มิลลิเมตร สูง 70 มิลลิเมตร และเมื่อวางซ้อนกันสองชั้นโดยเว้นระยะห่างกัน 10 มิลลิเมตร จะได้ Block heater สูง 150 มิลลิเมตร

จากการทดลองควบคุมอุณหภูมิในหัวข้อก่อนหน้านี้ทำให้ทราบว่าเมื่อเพิ่มอุณหภูมิของแกรไฟต์ในบรรยากาศ จะทำให้แกรไฟต์เกิดการทำปฏิกิริยากับอากาศและมีการกร่อนเกิดขึ้น ดังนั้นจึงได้ตัดสินใจไม่ใช้แกรไฟต์ในการกักเก็บพลังงานความร้อนของระบบและทำการทดลองเพื่อหาวิธีการป้องกันการสูญเสียความร้อนภายในระบบต่อไป

จากการทดลองออกแบบและสร้างระบบป้องกันการสูญเสียความร้อนภายในระบบ ผู้ทำการวิจัย ได้ใช้วิธีการสร้างผนังขึ้นมาทั้งหมด 4 ชั้น โดยในชั้นในสุดคือ ท่อสแตนเลส ซึ่งถูกวางอยู่ในตำแหน่งถัดจาก Block heater ซึ่งถูกปิดด้านบนด้วยอิฐทนไฟอีกรอบ ถัดมาเป็นชั้นของผนังอิฐทนไฟ มีขนาดกว้าง 230 มิลลิเมตร ยาว 230 มิลลิเมตร และสูง 300 มิลลิเมตร ในชั้นถัดมาเป็นชั้นระหว่างสังกะสีและฉนวนใยแก้ว โดยจุดประสงค์ในชั้นนี้คือการป้องกันการสูญเสียความร้อนภายในระบบสู่ภายนอกและช่วยป้องกันความร้อนที่แผ่ออกมาจากภายในที่จะส่งผลกระทบต่ออุปกรณ์ภายนอก

จากการทดลองควบคุมอุณหภูมิเพื่อทดสอบระบบกักเก็บพลังงานความร้อนพบว่าเตาความร้อนสามารถเพิ่มอุณหภูมิได้ถึงเป้าหมายที่ 1000 องศาเซลเซียส ตรงตามความต้องการในการใช้งาน และผลการทดลองต่อไปจะเป็นผลการทดลองปรับค่าตัวแปรควบคุมอุณหภูมิ และแสดงผลการควบคุมอุณหภูมิ

ในการทดลองปรับค่าตัวแปรควบคุมแบบพีไอดีด้วยวิธีการของ Ziegler – Nichol ผสมกับวิธีการ Trial and error ได้แบ่งตัวแปรในการควบคุมอุณหภูมิออกเป็นสองช่วง ได้แก่ ตัวแปรในการควบคุมอุณหภูมิในช่วงต่ำกว่า 500 องศาเซลเซียส จากการทดลองสามารถหาค่าตัวแปรควบคุมพีไอดี ดังนี้ $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ เมื่อทำการทดลองควบคุมอุณหภูมิแล้วพบว่าในช่วงอุณหภูมิ 100 และ 200 องศาเซลเซียสระบบเกิดความไม่เสถียร เนื่องจากตัวแปรควบคุม T_i และ T_D ดังนั้นผู้วิจัยจึงใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0$ $T_D = 0$ สำหรับควบคุมอุณหภูมิในช่วง 100 และ 200 องศาเซลเซียส และตัวแปรในการควบคุมอุณหภูมิในช่วงอุณหภูมิสูง 1000 องศาเซลเซียส ทดลองโดยใช้ค่าตัวแปรพีไอดี $K_p = 450$ $T_i = 0$ $T_D = 0$

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 100 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 5 นาที โดยมีค่า Overshoot 10 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 30 นาที พบว่าสามารถควบคุมอุณหภูมิมี่ค่าความผิดพลาดของระบบอยู่ที่ ± 3 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 200 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0$ $T_D = 0$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 10 นาที โดยมีค่า Overshoot 10 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 15 นาที พบว่าสามารถควบคุมอุณหภูมิมี่ค่าความผิดพลาดของระบบอยู่ที่ ± 3 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 300 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 15 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมี่ค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 400 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 20 นาที โดยมีค่า Overshoot 2 องศาเซลเซียส และเมื่อระบบเข้าสู่สมดุลโดยใช้เวลา 2 นาที พบว่าสามารถควบคุมอุณหภูมิมี่ค่าความผิดพลาดของระบบอยู่ที่ ± 2 องศาเซลเซียส

ผลการทดลองอุณหภูมิในช่วงอุณหภูมิ 500 องศาเซลเซียส จากการใช้ตัวแปรควบคุม $K_p = 250$ $T_i = 0.05$ $T_D = 0.20$ พบว่าระบบใช้เวลาเพิ่มอุณหภูมิถึงเป้าหมายในเวลา 25 นาที

5.2 ข้อเสนอแนะ

ข้อเสนอแนะที่ 1 จากการทดลองควบคุมอุณหภูมิ พบว่าในช่วงแรกของการควบคุมระบบจะมีการแกว่งของอุณหภูมิเล็กน้อย ต้องรอเวลาในการควบคุมสักระยะให้ระบบเข้าสู่สภาวะสมดุลแล้ว อุณหภูมิจะมีระดับคงที่ โดยมีการแกว่งของอุณหภูมิเล็กน้อยประมาณ 2 องศาเซลเซียส ที่อุณหภูมิ 1000 องศาเซลเซียส

ข้อเสนอแนะที่ 2 ในการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิระหว่าง 100 และ 200 องศาเซลเซียส ระบบควบคุมจะมีความเสถียรน้อย ซึ่งใช้เวลาในการเข้าสู่สภาวะสมดุลนาน และในค่าอุณหภูมิที่มีค่าความผิดพลาดมากเกินไปสามารถเลือกปรับแก้ตัวแปรควบคุมอุณหภูมิได้ โดยใช้วิธีการ Trial and error เพื่อหาค่าตัวแปรควบคุมที่เหมาะสมในการใช้งานได้ที่หน้าจอคอมพิวเตอร์

ข้อเสนอแนะที่ 3 ในการทดลองควบคุมอุณหภูมิในช่วงอุณหภูมิสูง เมื่อทำการทดลองเสร็จแล้ว ควรพักเตาไว้สักระยะเพื่อให้อุณหภูมิลดลงก่อนทำการเปิดเตา เพื่อป้องกันการเกิดอันตรายจากการสัมผัสความร้อนในอุปกรณ์ภายในเตาความร้อน

เอกสารอ้างอิง

เอกสารอ้างอิง

- [1] ประชา เหล่าอวยพร. **การพัฒนาระบบควบคุมอุณหภูมิในเตาปลูกผลึกโดยใช้แบบจำลองทางคณิตศาสตร์**. วิทยานิพนธ์ปริญญาวิศวกรรมมหาบัณฑิต: มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2554.
- [2] ธนากร เกียรติขวัญบุตร และสุชาติ จันทรมณี. **การออกแบบและสร้างเตาอุณหภูมิสูงแบบท่อ**. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต: มหาวิทยาลัยสงขลานครินทร์, 2551.
- [3] National Instrument. (2011). “PID Theory Explained”, **White Papers**.
<http://www.ni.com/white-paper/3782/en/#toc2>. February 15, 2016.
- [4] เชิดศักดิ์ บุตรจอมชัย. **สมบัติเชิงวัสดุ**. กรุงเทพฯ: แดเน็กซ์ อินเตอร์คอร์ปอเรชั่น, 2558.
- [5] กรมพัฒนาพลังงานทดแทน และอนุรักษ์พลังงาน, องค์การร่วมมือระหว่างประเทศแห่งประเทศไทย และสถาบันวิจัยและให้คำปรึกษาแห่งมหาวิทยาลัยธรรมศาสตร์. **ร่างตำราฝึกอบรมผู้รับผิดชอบด้านพลังงาน(ผขพ.) ด้านความร้อน**. กรุงเทพฯ: กรมพัฒนาและอนุรักษ์พลังงาน, 2558.
- [6] Maxim integrated. (2015) “Cold-Junction Compensation 1-WIRE Thermocouple – to – Digital Converters”, **Products 1-WIRE**.
<https://www.maximintegrated.com>. February 16, 2016.
- [7] วรพงศ์ ตั้งศรีรัตน์. **การวัดและควบคุมกระบวนการ**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยีไทย – ญี่ปุ่น, 2550.
- [8] มหาวิทยาลัยศิลปากร **คู่มือปฏิบัติการฟิสิกส์1**. นครปฐม: คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร, 2528.
- [9] กนต์ธร ขำนิประศาสน์ **การวัดเชิงกลด้วย LabVIEW**. นครราชสีมา: มหาวิทยาลัยเทคโนโลยีสุรนารี, 2560.
- [10] ทันพงษ์ ภูริรักษ์. “โครงสร้างโปรแกรมของ Arduino”, **เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์เบื้องต้น**. สระบุรี: วิทยาลัยเทคนิคสระบุรี, 2559.
- [11] S Kurmar, J B hattacharyya and M Bhattacharyya. “Design of a PID auto tuning controller using LabVIEW”, **International Research Journal of Engineering and Technology**. 3(12): 341-345; December 16, 2016.
- [12] H Muhammad Asraf and KA. Nur Dalila. “Virtual Laboratory in LabVIEW for control and supervision applications with Arduino”, **International journal of Applied Engineering Research**. 13(23): 53-57; February 18, 2018.

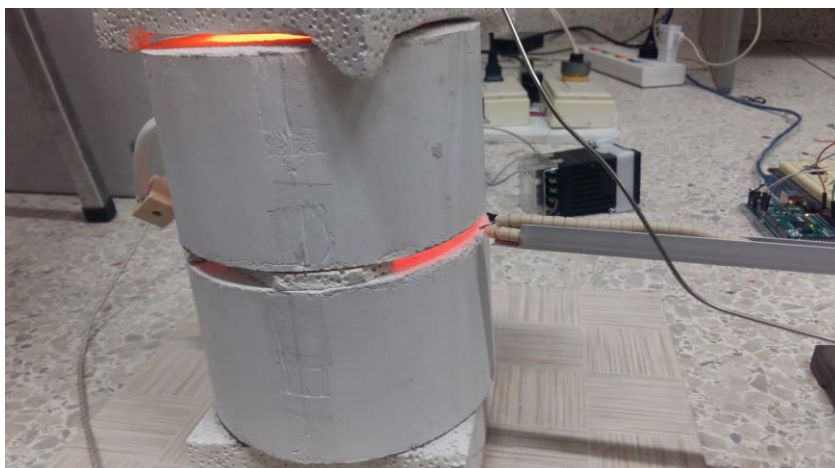
เอกสารอ้างอิง (ต่อ)

- [13] แม้น อมรสิทธิ์, สมชัย อัครทิวา และธรรมนุญ อุตมมัน. **วัสดุวิศวกรรม**. กรุงเทพฯ: แมดกรอ – ฮิล อินเทอร์เน็ตชั้นแนล เอ็นเตอร์ไพรส์ แอลแอลซี, 2551.

ภาคผนวก

ภาคผนวก ก

เตาความร้อนที่สร้างสำเร็จเพื่อใช้ในการทดลอง



ภาพที่ ก.1 Block heater



ภาพที่ ก.2 เตาความร้อนเมื่อเพิ่มชั้นฉนวนอิฐทนไฟเพื่อป้องกันความร้อน



ภาพที่ ก.3 เตาความร้อนเมื่อเพิ่มชั้นสังกะสีเพื่อป้องกันการสูญเสียความร้อน

ภาคผนวก ข

โปรแกรมการวัดอุณหภูมิด้วยไอซี MAX31850 ใช้งานทาง LIFA_BASE

โปรแกรมการวัดอุณหภูมิด้วยไอซี MAX31850 ใช้งานทาง LIFA_BASE

```
float getTemp()    //returns the temperature from one Max31850 in DEG Celsius
{
    byte i;
    byte present = 0;
    byte temptype;
    byte data[12];
    byte addr[8];
    float celsius, fahrenheit;
    if ( !ds.search(addr)) {
        // Serial.println();
        // Serial.println("No more addresses.");
        // Serial.println();
        ds.reset_search();
        delay(250);
        return;
    }
    // Serial.print("ROM =");
    for( i = 0; i < 8; i++) {
        // Serial.write(' ');
        // Serial.print(addr[i], HEX);
    }
    if (OneWire::crc8(addr, 7) != addr[7]) {
        // Serial.println("CRC is not valid!");
        return;
    }
    // Serial.println();
    // the first ROM byte indicates which chip
    switch (addr[0]) {
        case 0x10:
            // Serial.println(" Chip = DS18S20"); // or old DS1820
            temptype = TYPE_DS18S20;
            break;
        case 0x28:
```

```

    // Serial.println(" Chip = DS18B20");
    temptype = TYPE_DS18B20;
    break;
case 0x22:
    // Serial.println(" Chip = DS1822");
    temptype = TYPE_DS18S22;
    break;
    // ADDED SUPPORT FOR MAX31850!
case 0x3B:
    // Serial.println(" Chip = MAX31850");
    temptype = TYPE_MAX31850;
    break;
default:
    // Serial.println("Device is not a DS18x20 family device.");
    return;
}
ds.reset();
ds.select(addr);
ds.write(0x44, 1);    // start conversion, with parasite power on at the end
delay(1000);    // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.
present = ds.reset();
ds.select(addr);
ds.write(0xBE);    // Read Scratchpad
// Serial.print(" Data = ");
// Serial.print(present, HEX);
// Serial.print(" ");
for ( i = 0; i < 9; i++) {    // we need 9 bytes
    data[i] = ds.read();
    // Serial.print(data[i], HEX);
    // Serial.print(" ");
}
//Serial.print(" CRC=");
// Serial.print(OneWire::crc8(data, 8), HEX);
// Serial.println();
// Serial.print(" Address = 0x"); Serial.println(data[4] & 0xF, HEX);

```

```

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (temptype == TYPE_DS18S20) {
    raw = raw << 3; // 9 bit resolution default
    if (data[7] == 0x10) {
        // "count remain" gives full 12 bit resolution
        raw = (raw & 0xFFF0) + 12 - data[6];
    }
} else if (temptype == TYPE_MAX31850) {
    //Serial.println(raw, HEX);
    if (raw & 0x01) {
        Serial.println("***FAULT***");
        return;
    }
} else {
    byte cfg = (data[4] & 0x60);
    // at lower res, the low bits are undefined, so let's zero them
    if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
    else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
    else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
    //// default is 12 bit resolution, 750 ms conversion time
}
celsius = (float)raw / 16.0;
float TemperatureSum = celsius;
return TemperatureSum;
}

```

ภาคผนวก ค
เอกสารงานวิจัยตีพิมพ์เผยแพร่

PAPER • OPEN ACCESS

Development of the PID controller and real-time monitoring system for a low-temperature furnace

To cite this article: T Pangya et al 2018 *J. Phys.: Conf. Ser.* **1144** 012160

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices
to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of
every title for free.

Development of the PID controller and real-time monitoring system for a low-temperature furnace

T Pangya¹, A Thedsakulwong^{1*} and C Ekwongsa²

¹Department of Physics, Faculty of science, Ubon Ratchathani University, Warinchamrap, Ubon Ratchathani, 34190, Thailand

²School of Physics, Institute of science, Suranaree University of Technology, Mung, Nakhon Ratchasima, 30000, Thailand

*Email: amorn.t@ubu.ac.th

Abstract. The PID controller and real-time monitoring system for a low-temperature furnace was developed. The system has two part, the PID controller, and the real-time monitoring part. An Arduino mega2560 microcontroller board was used for measuring and control the furnace temperature. A type-K thermocouple and a MAX31850 IC was applied for a furnace temperature measurement. The microcontroller board and a MAX31850 were connected via the One-wire bus for convert the temperature values and sent to a personal computer. The PID parameters can be varied by a user in the program, which developed by LabVIEW Software on a computer. The laboratory made furnace was established for testing the controller and monitoring system. The results have shown that the temperature with the range of 25-500 degree Celsius can be controlled. By the trial and error method with the PID parameters, k_p was 250, T_i was 0.05 and T_d was 0.20, the target temperature can be controlled with the maximum error of 1 degree Celsius.

1. Introduction

The temperature controller is a necessity in research on the thermal properties of substances. There are many types of research that have been conducted previously using temperature controller system. Such as the effect of heating rate on structural and functional properties of wheat and potato starch-water system in food science technology [1]. The investigate the effect of each temperature-dependent material property on the transient temperature, residual stress and distortion of welding process [2] and examples of research that use high-temperature control. Such as The high-temperature ablation properties of NANO zirconia reinforced fused silica ceramic [3]. In materials process technology. All of above was require the high resolution of the temperature controller and monitoring system.

The Proportional Integral Derivative (PID) Controller is the most common method for the temperature controls system designing. The heat treatment furnace in the industrial, which used Intelligent Fuzzy Logic and PID controllers to design of temperature controller [4]. Recently, the LabVIEW program applied to designs PID control system [5] and the PID temperature autotuning control designing can accomplish by using LabVIEW [6].

This research has development of the PID temperature control system and real-time temperature monitoring Using LabVIEW interface for Arduino. The temperature measured with sensor MAX31850 Communication with the master microcontroller over the 1-Wire bus. Read temperature data from a sensor with Arduino MEGA2560. Then we get temperature data from Arduino and real-time temperature monitoring with the LabVIEW software. In this research design of PID temperature control by Ziegler Nichols and trial error method.

2. Experimental

2.1 Hardware design.

The system composes of the temperature measurement and the Arduino microcontroller board. The furnace temperature was measured by the thermocouple amplifier with 1-wire breakout Board-MAX31850K. The Arduino board read the temperature data and send to display on a computer via USB port. The target temperature maintains by on/off a solid state relay (SSR) which controls the period with the Arduino. The diagram of the system shows in Figure 1.

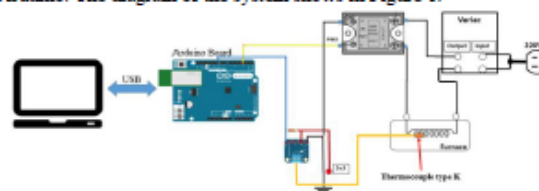


Figure1. The diagram of the PID controller and real-time monitoring system.

2.2 Software design.

The PID parameters setting can be varies and real time temperature monitoring can be shows on the front panel of the developed program, respectively. The front panel of the temperature control system designed shows in Figure2.



Figure 2. The front panel of the temperature controller

The block diagram of PID control program shows in Figure 3. The program read the furnace temperature from the Arduino and real time display in the front panel. The period for on/off the SSR accumulates from the PID parameters and send to the Arduino for on/off the SSR. The target temperature and the various PID parameters are the input parameter of the program. The Ziegler Nichols function from equation (1) used to calculate the output of signal and the trial and error used to find optimizes parameter of this system. Where K_p , T_i and T_d is Parameter of proportional, integral and Derivative respectively

$$U = K_p \left(1 + \frac{1}{T_i} + T_d \right) \quad (1)$$

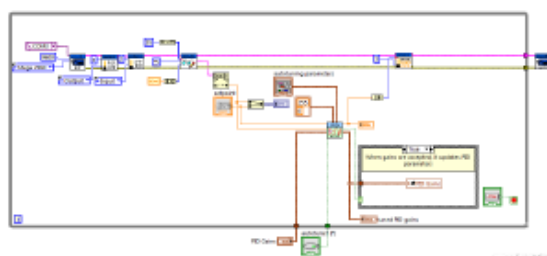


Figure3. Block diagram of PID control from LabVIEW

3. Result and discussion

The Ziegler Nichols function used to generates the output signal for controller. The PID parameter was varied by the trial and error method.

3.1 The K_p parameter optimization

Table 1. Shows the temperature settling value, the steady state error, and the overshoot where T_i was 0, T_d was 0 and the parameter K_p was varied the K_p has the effect on the settling and steady state error. The optimize K_p was 250.

Table1. List of comparisons of response parameter of all design K_p at $T_i = 0$, $T_d = 0$			
Parameter	Settling value	Steady state error (°C)	Overshoot
$K_p = 50$	This temperature is below the setpoint at 3°C	± 6	No overshoots.
$K_p = 100$	This temperature is below the setpoint at 3°C	± 6	No overshoots.
$K_p = 150$	This temperature is below the setpoint at 2°C	± 5	No overshoots.
$K_p = 200$	This temperature little below the setpoint at 1°C	± 3	No overshoots.
$K_p = 250$	This temperature is oscillates around the setpoint at ± 2 °C	± 2	Overshoot is absorb

3.2 The T_i parameter optimization

Table 2. Show the rising time, the steady state error, and the overshoot temperature where the K_p was 250, the T_d was 0, and the T_i was varied. The results shown that the T_i has the effect of rising time and the oscillates of temperature. The optimize T_i value was 0.05.

Table2. List of comparisons of response parameter of all design T_i at $K_p = 250$, $T_d = 0$.			
Parameter	Rising time (s)	Steady state error(°C)	Overshoot (°C)
$T_i = 0.05$	120	± 1	5
$T_i = 0.10$	155	± 2	5
$T_i = 0.15$	165	± 3	5
$T_i = 0.20$	185	± 4	5
$T_i = 0.30$	210	± 6	7

3.3 The T_d parameter optimization.

Table 3. Show the rising time, the steady state error, and the overshoot temperature where the K_p was 250, the T_i was 0.05, and the T_d was varied. The results shown that the least T_d has minimum rising time and the minimum oscillates was 1 °C with the T_d value was 0.20

Table3. List of comparisons of response parameter of all design Td at $Kp=250$, $Ti=0.05$

Parameter	Rising time (s)	Steady state error (°C)	Overshoot (°C)
$Td = 0.05$	120	± 3	4
$Td = 0.10$	180	± 2	3
$Td = 0.15$	195	± 2	3
$Td = 0.20$	180	± 1	2
$Td = 0.25$	180	± 2	4

The experiment shown that the optimize of the Kp parameter was 250, the Ti parameter was 0.05, and the Td parameter was 20.

3.4 Test the temperature control.

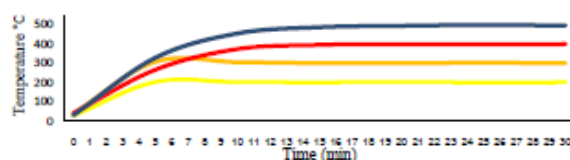


Figure 4. The various target temperature control with the optimized PID parameter.

The results shown that all of target temperature can be controlled with the maximum error of 1°C. The rising time of control was 270, 290, 720, and 1200 second with the target temperature 200°C, 300°C, 400°C, 500°C respectively.

4. Conclusion

In research, to developed and designed the PID temperature control which can be monitoring the temperature on the computer display. From experiments to find parameters of PID show that the best of Kp cause to the system response better and the output is a small steady-state error. The best Ti value cause to decrease the response of the system and the best Td cause to decrease overshoot of the system and increase the response of the system. By trial and error method with the PID parameters, Kp was 250, Ti was 0.05 and Td was 0.20. The experimental results of control temperature can be controlled to a stable temperature. The maximum error is 1 °C and rising time of temperature 200°C, 300 °C, 400°C, and 500°C was 270,290,720, and 1200 second respectively.

References

- [1] Mulumbaa P, Doranb L, Danthineb S, Bleckerb C and Mera F 2018 *Food science and Technology* 88 196-202
- [2] Zhu XK and Chao YJ 2002 *Computer & Structures* 80 967-976
- [3] Wan W, Luo J, Feng Y, Ouyang Y, Chen D and Qiu T 2018 *Ceramic International* 44 7273-7275
- [4] Srinivasulu Raju S, Chaitanya Kumar R, Ravi K, Viswanath M 2013 *International J. Advanced Research in Computer Science* 4 12-17
- [5] Muhammad Asraf H and Nur Dalila KA 2016 *J. Telecommunication, Electronic and Computer Engineering* 9 53-57
- [6] Kumar S, Bhattacharyya J and Bhattacharyya M 2016 *International Research Journal of Engineering and Technology* 3 341-345

ประวัติผู้วิจัย

ชื่อ	นายธรรมกรานต์ พังยะ
ที่อยู่	เลขที่ 10 หมู่ 11 ตำบลพุ่มแก อำเภอนาแก จังหวัดนครพนม 48130
E-mail	Thammakarn.pa.55@ubu.ac.th
ประวัติการศึกษา	พ.ศ. 2549-2552 มัธยมศึกษาตอนต้น โรงเรียนธาตุพนม ตำบลธาตุพนม อำเภธาตุพนม จังหวัดนครพนม 48110 พ.ศ. 2552-2555 มัธยมศึกษาตอนปลาย โรงเรียนนาแกสามัคคีวิทยา ตำบล บ้านแก่ง อำเภอนาแก จังหวัดนครพนม 48130 พ.ศ. 2555-2559 ปริญญาตรี มหาวิทยาลัยอุบลราชธานี วิทยาศาสตร์บัณฑิต สาขาวิชาฟิสิกส์
ประวัติการวิจัย	- Development of the PID controller and real-time monitoring system for a low-temperature furnace