



ตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต

โดย
อดิศักดิ์ แซ่สาริกิจ

สนับสนุนงบประมาณโดย
มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์
งบประมาณ ปี 2561

Position Controller of Kinetics Rain via Intranet

By
Adisak Khaengsarigid

Granted by
Rajamangala University of Technology Rattanakosin
Fiscal year 2018

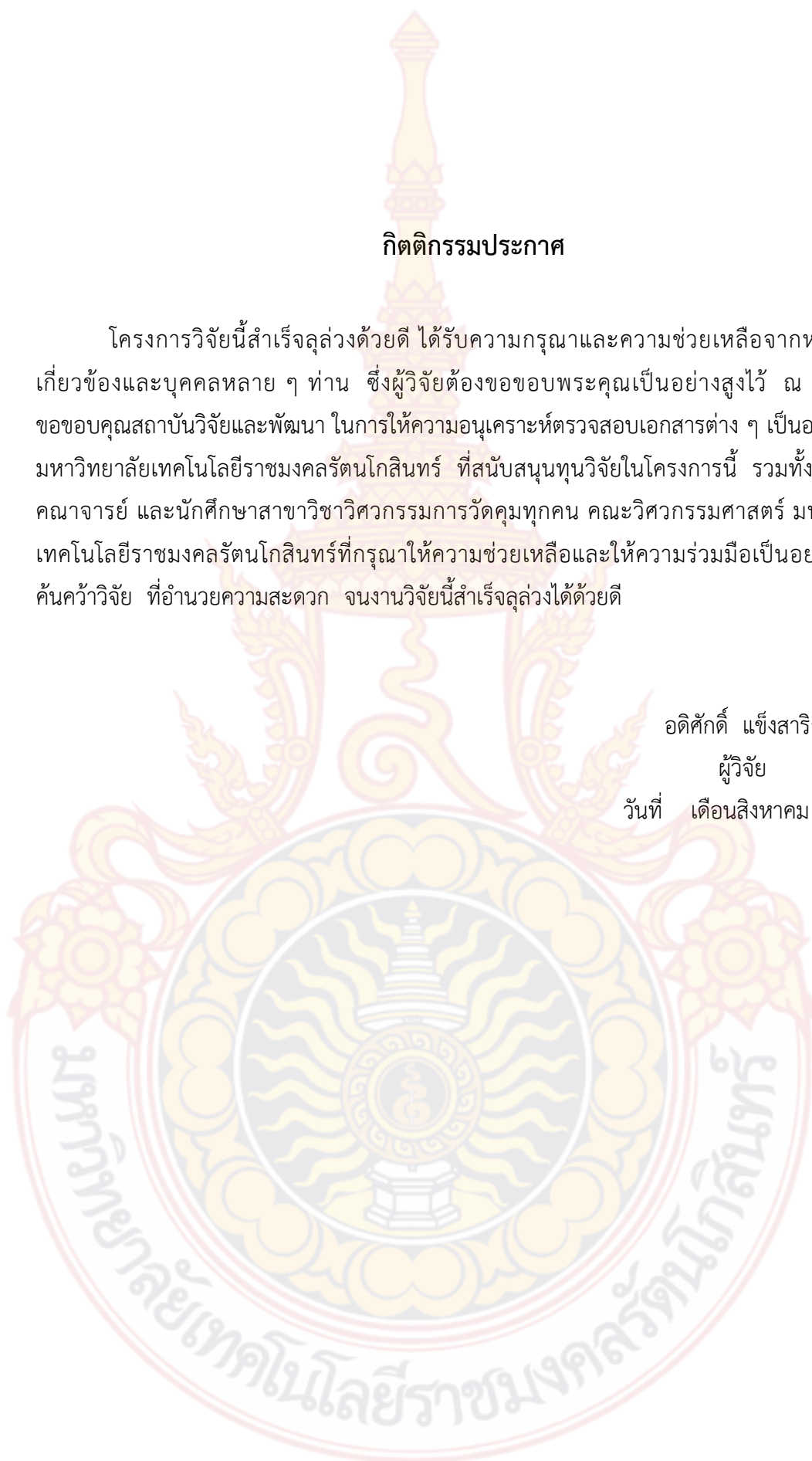
กิตติกรรมประกาศ

โครงการวิจัยนี้สำเร็จลุล่วงด้วยดี ได้รับความกรุณาและความช่วยเหลือจากหน่วยงานที่เกี่ยวข้องและบุคคลหลาย ๆ ท่าน ซึ่งผู้วิจัยต้องขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ ผู้วิจัยขอขอบคุณสถาบันวิจัยและพัฒนา ในการให้ความอนุเคราะห์ตรวจสอบเอกสารต่าง ๆ เป็นอย่างดี และมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ที่สนับสนุนทุนวิจัยในโครงการนี้ รวมทั้งขอขอบคุณคณาจารย์ และนักศึกษาด้านวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรีที่กรุณาให้ความช่วยเหลือและให้ความร่วมมือเป็นอย่างดีในการค้นคว้าวิจัย ที่อำนวยความสะดวก งานวิจัยนี้สำเร็จลุล่วงได้ด้วยดี

อดิศักดิ์ แข็งสาริกิจ

ผู้วิจัย

วันที่ เดือนสิงหาคม ปี 2562



บทคัดย่อ

รหัสโครงการ : INN-10

ชื่อโครงการ : ตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต

ชื่อนักวิจัย : อติศักดิ์ แข็งสาริกิจ

โครงการวิจัยนำเสนอตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต การออกแบบกำหนดให้บอร์ดติดต่อสื่อสารอินเทอร์เน็ตทำหน้าที่รับรูปแบบการเลือกจากเว็บ และบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega2560 ทำหน้าที่สร้างเส้นทางเดิน และเป็นตัวควบคุม PID

การทดลองกระทำการควบคุมตำแหน่ง Kinetics Rain 3 รูปแบบ คือ 1. รูปคลื่น Sine wave 2. รูปเรียงระฆังคว่ำ และ 3. รูปเรียงระฆังหงาย โดยการเลือกรูปแบบจากเว็บ จากนั้นบอร์ดติดต่อสื่อสารอินเทอร์เน็ต รับคำสั่งรูปแบบและส่งต่อให้บอร์ดไมโครคอนโทรลเลอร์ ซึ่งบอร์ดไมโครคอนโทรลเลอร์ทำหน้าที่สร้างเส้นทางเดิน ตัวควบคุม PID และส่งสัญญาณให้กับวงจรขับเคลื่อนมอเตอร์ ส่วนสัญญาณป้อนกลับใช้โพเทนซิโอมิเตอร์ ผลการทดลองพบว่าตัวควบคุมสามารถควบคุมได้ทั้ง 3 รูปแบบได้ โครงการนี้เป็นแนวทางให้เกิดประโยชน์สำหรับการเรียนการสอนในสาขาวิชาวิศวกรรมการวัดคุมสามารถนำมาเป็นชุดปฏิบัติการของสาขาวิศวกรรมการวัดคุมได้

คำสำคัญ : “พีไอดี” “ตัวควบคุม” “ไมโครคอนโทรลเลอร์” “Kinetics Rain” “เส้นทางเดิน”

E-mail address : adisak.kha@rmutr.ac.th

ระยะเวลาโครงการ : ตุลาคม 2560 – กันยายน 2561

Abstract

Code of project : INN-10

Project name : Position Controller of Kinetics Rain via Intranet

Researchers name : Mr. Adisak Khaengsarigid

The research presents the position controller of kinetics rain via Intranet. The design requires the internet communication board to accept the selection form of kinetics rain from the web. The Arduino Mega 2560 microcontroller board was defined to trajectory generator and PID controller.

The experiments perform to position control for three Kinetics Rain form follow as: Sine wave pattern, the supine bell and the bell inverted. By choosing a pattern from the web. The internet communication board accept the pattern and send the pattern to the microcontroller board. Which the microcontroller board performs to generate the path to control, the PID controller and sending the signal to the motor driver. The feedback signal uses a potentiometer. The results of the experiment showed that the controller can control all 3 forms.

Keywords : “PID” “Controller” “Microcontroller” “Kinetics Rain” “Trajectory”

E-mail address : adisak.kha@rmutr.ac.th

Project Duration : October 2017 – September 2018

สารบัญ

หน้า

กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	จ
สารบัญตาราง	ช
สารบัญภาพ	ญ
บทที่ 1	
บทนำ.....	1
ความสำคัญและที่มาของปัญหา.....	1
วัตถุประสงค์.....	1
ขอบเขตการวิจัย.....	2
ผลประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2	
ทฤษฎีที่เกี่ยวข้อง.....	3
โครงสร้างพื้นฐาน Kinetics Rain	3
การออกแบบเส้นทางเดิน	5
ระบบควบคุมพื้นฐาน.....	6
ระบบควบคุมแบบ PID.....	7
การแปลงลาปลาซ (Laplace Transform)	14
อุปกรณ์ตรวจวัด (Sensor and Transducer).....	18
มอเตอร์ไฟฟ้ากระแสตรง (DC Motor)	19
ไมโครคอนโทรลเลอร์.....	23
บอร์ดติดต่อสื่อสารกับอินเทอร์เน็ต ESP8266	26
บทที่ 3	
วิธีดำเนินการวิจัย.....	28
ขั้นตอนการวิจัย.....	28
ขั้นตอนการดำเนินงานวิจัย	28

สารบัญ (ต่อ)

บทที่ 4	ผลการทดสอบและการวิเคราะห์ข้อมูล.....	35
	วิธีการทดลอง.....	36
	บันทึกผลการทดลอง.....	36
	วิจารณ์ผลการทดลอง.....	37
บทที่ 5	สรุปผลของการวิจัย.....	38
	แนวทางการพัฒนาโครงการวิจัย.....	38
ภาคผนวก	39
บรรณานุกรม	49
ประวัติผู้วิจัย	50



สารบัญตาราง

ตารางที่		หน้า
1	ค่าพารามิเตอร์ของการควบคุมแต่ละชนิดซึ่งหาได้จากวิธี Process Reaction Curve	13
2	ค่าพารามิเตอร์ของการควบคุมแต่ละชนิดซึ่งหาได้จากวิธี Ultimate Method	14
3	ตารางการแปลงลาปลาซ	15
4	พอร์ตที่สามารถใช้งานอินเทอร์รัพท์	25

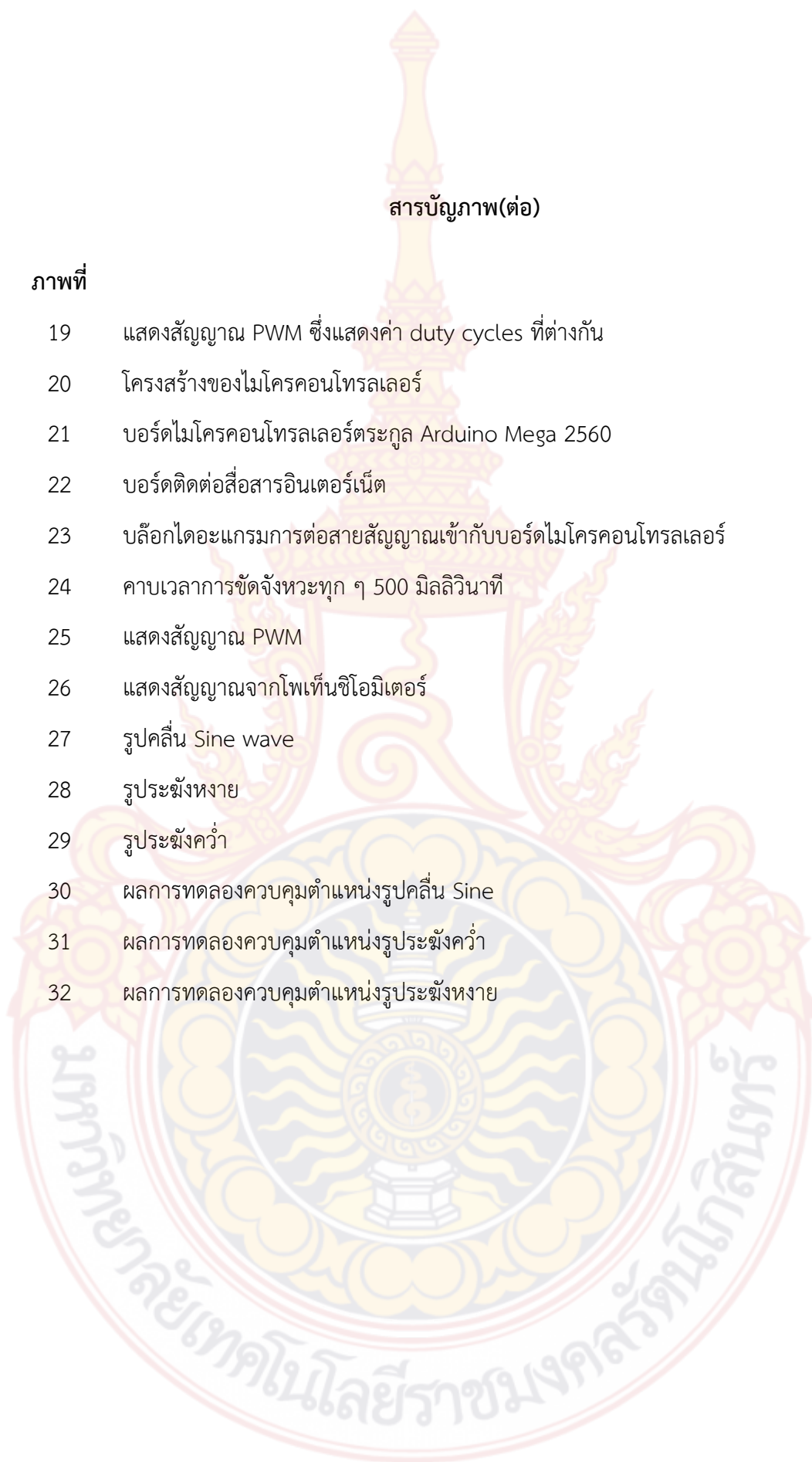


สารบัญภาพ

ภาพที่		หน้า
1	Kinetics Rain	1
2	แสดงการเคลื่อนที่ Kinetics Rain	3
3	โครงสร้างทางกลของ Kinetics Rain	4
4	โครงสร้างทางไฟฟ้าของ Kinetics Rain	4
5	แสดงเส้นทางเดินการเคลื่อนที่	5
6	ระบบควบคุมวงเปิด	7
7	ระบบควบคุมวงปิด	7
8	แผนภาพบล็อกของการควบคุมแบบพีไอดี	8
9	กราฟ PV ต่อเวลา, K_p กำหนดเป็น 3 ค่า K_i และ K_d คงที่	9
10	กราฟ PV ต่อเวลา, K_i กำหนดเป็น 3 ค่า K_p และ K_d คงที่	10
11	กราฟ PV ต่อเวลา, K_d กำหนดเป็น 3 ค่า K_p และ K_i คงที่	11
12	ผลตอบสนองของกระบวนการที่มีอัตราเสื่อมเท่ากับ $\frac{1}{4}$ (25%)	12
13	การหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method	14
14	โครงสร้างภายในโพเทนชิโอมิเตอร์	19
15	แสดงการเลื่อน หรือการเปลี่ยนแปลงค่าความต้านทาน	19
16	ความสัมพันธ์ระหว่างแรงบิดกับความเร็วในทางอุดมคติ	21
17	ความสัมพันธ์ระหว่างแรงบิดกับความเร็วในทางปฏิบัติ	21
18	ความสัมพันธ์ระหว่างแรงดันกับความเร็วในการควบคุมด้วยแรงดัน	22

สารบัญภาพ(ต่อ)

ภาพที่		หน้า
19	แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างกัน	22
20	โครงสร้างของไมโครคอนโทรลเลอร์	23
21	บอร์ดไมโครคอนโทรลเลอร์ตระกูล Arduino Mega 2560	24
22	บอร์ดติดต่อสื่อสารอินเทอร์เน็ต	26
23	บล็อกไดอะแกรมการต่อสายสัญญาณเข้ากับบอร์ดไมโครคอนโทรลเลอร์	32
24	คาบเวลาการขัดจังหวะทุก ๆ 500 มิลลิวินาที	33
25	แสดงสัญญาณ PWM	33
26	แสดงสัญญาณจากโพเทนชิโอมิเตอร์	34
27	รูปคลื่น Sine wave	35
28	รูประฆังหาง	35
29	รูประฆังคว่ำ	36
30	ผลการทดลองควบคุมตำแหน่งรูปคลื่น Sine	36
31	ผลการทดลองควบคุมตำแหน่งรูประฆังคว่ำ	37
32	ผลการทดลองควบคุมตำแหน่งรูประฆังหาง	37



บทที่ 1

บทนำ

1. ความสำคัญและที่มาของปัญหา

ปัจจุบันการประยุกต์ใช้งานระบบเครือข่ายอินเทอร์เน็ตหรือ อินเทอร์เน็ตเข้ามามีบทบาทในการดำเนินชีวิตประจำวันมากขึ้น เช่นการสืบค้นข้อมูล ซื้อขายสินค้าออนไลน์ นำเสนอสินค้าออนไลน์ แนะนำสถานที่ท่องเที่ยว การเงิน การกีฬา ไปรษณีย์อิเล็กทรอนิกส์ หรืออีเมล (email) สนทนา (chat) อ่านหรือแสดงความคิดเห็นในเว็บบอร์ด การติดตามข่าวสาร การดาวโหลด เกม เพลง ไฟล์ข้อมูล ฯลฯ การเล่นเกมคอมพิวเตอร์ออนไลน์ การเรียนรู้ออนไลน์ (e-learning) การประชุมทางไกลผ่านอินเทอร์เน็ต (video conference) โทรศัพท์ผ่านอินเทอร์เน็ต (VoIP) การอัปโหลดข้อมูล หรืออื่น ๆ

โดยงานวิจัยนี้เป็นการประยุกต์ใช้งานเครือข่ายอินเทอร์เน็ตให้สามารถใช้งานด้านการควบคุมตำแหน่งและแสดงผลข้อมูลทางเทคนิคของ Kinetics Rain ผ่านเครือข่ายอินเทอร์เน็ต[1-2] หมายความว่าสามารถควบคุมตำแหน่งของ Kinetics Rain ได้หลายรูปแบบ แสดงค่าสถานะ และแสดงผลข้อมูลทางเทคนิคบนเครือข่ายอินเทอร์เน็ต ซึ่งเป็นแนวทางในการพัฒนาเข้าสู่อุตสาหกรรมยุคที่ 4 และหรือประเทศไทย 4.0 คือการทำให้ Kinetics Rain เป็นอุปกรณ์อุตสาหกรรมยุคที่ 4[1-3] ที่มีการจัดการบนเครือข่ายอินเทอร์เน็ตผ่านโปรแกรมบราวเซอร์ และบอร์ดไมโครคอนโทรลเลอร์ [4-5] ทำงานร่วมกัน



ภาพที่ 1 Kinetics Rain

2. วัตถุประสงค์ของการวิจัย

- 2.1 ศึกษาการควบคุมตำแหน่งของมอเตอร์ ในการทำ Kinetics Rain
- 2.2 ศึกษาการรับส่งข้อมูลชุดควบคุมให้ตัวควบคุมตำแหน่งผ่านเครือข่ายอินเทอร์เน็ต
- 2.3 ศึกษาการเขียนโปรแกรมติดต่อเว็บเบราว์เซอร์ เพื่อให้ Kinetics Rain แสดงผล

3. ขอบเขตของโครงการวิจัย

- 3.1 ชุด Kinetics Rain 10 ช่อง
- 3.2 สามารถควบคุมตำแหน่งมอเตอร์ของชุด Kinetics ได้
- 3.3 สามารถรับส่งค่าตำแหน่งมอเตอร์ของชุด Kinetics Rain ผ่านเครือข่ายอินเทอร์เน็ต

4. ผลประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

- 4.1 ทราบถึงการควบคุมตำแหน่งของมอเตอร์ ในการทำ Kinetics Rain
- 4.2 ทราบถึงการรับส่งข้อมูลชุดควบคุมให้ตัวควบคุมตำแหน่งผ่านเครือข่ายอินเทอร์เน็ต
- 4.3 ทราบถึงการเขียนโปรแกรมติดต่อเว็บเบราว์เซอร์ เพื่อให้ Kinetics Rain แสดงผล



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีเบื้องต้นที่เกี่ยวข้องในการออกแบบโครงการวิจัยตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต โดยส่วนใหญ่เป็นเนื้อหาการออกแบบระบบควบคุมตำแหน่งการเคลื่อนที่ของ Kinetics Rain ดังต่อไปนี้

1. โครงสร้างพื้นฐาน
2. การออกแบบทางเดินการเคลื่อนที่
3. ระบบควบคุมพื้นฐาน
4. ระบบควบคุมแบบ PID
5. การแปลงลาปลาซ
6. อุปกรณ์ตรวจวัด (Sensor and Transducer)
7. มอเตอร์ไฟฟ้ากระแสตรง
8. บอร์ดไมโครคอนโทรลเลอร์
9. บอร์ดติดต่อสื่อสารอินเทอร์เน็ต

1. โครงสร้างพื้นฐาน

นิยาม Kinetics Rain คือการควบคุมการเคลื่อนที่ทั้งตำแหน่ง และความเร็วของอุปกรณ์เลียนแบบเม็ดฝน โดยแต่ละเม็ดฝนจะขึ้นลงตามโปรแกรมที่ตั้งไว้ ทั้งนี้ขึ้นอยู่กับจำนวนช่องของการควบคุมดังภาพที่ 2 ฉะนั้นในการควบคุมการเคลื่อนที่ทั้งตำแหน่ง และความเร็ว จะต้องมีความราบเรียบในการเคลื่อนที่ขึ้น และลง เพื่อให้เกิดความสวยงาม



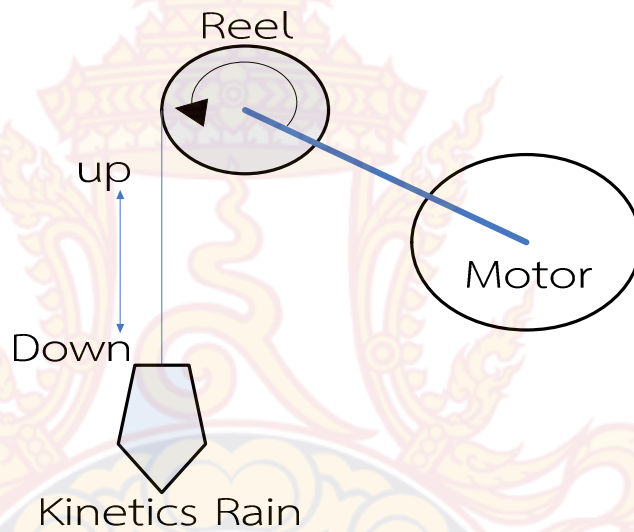
ภาพที่ 2 แสดงการเคลื่อนที่ Kinetics Rain

โครงสร้างพื้นฐานของ Kinetics Rain

Kinetics Rain ประกอบด้วย 2 ส่วนคือ

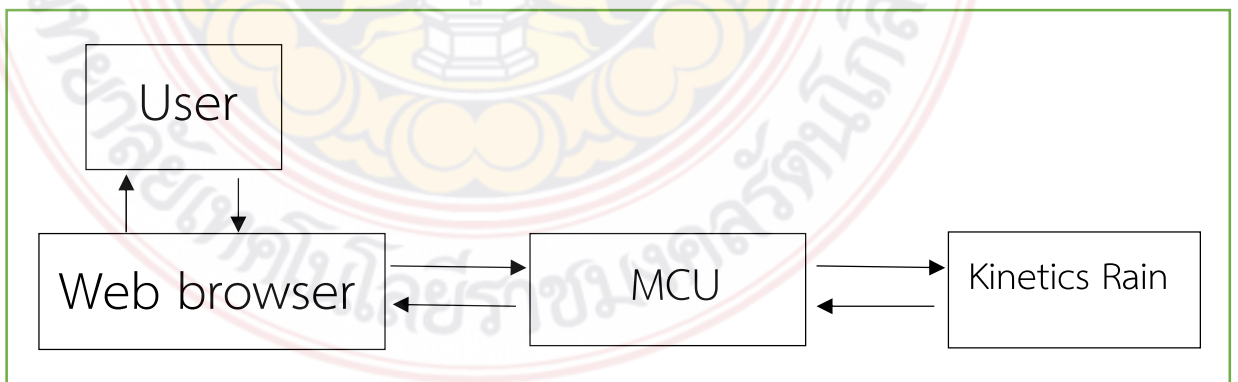
1. โครงสร้างเชิงกล
2. โครงสร้างทางไฟฟ้า

โครงสร้างเชิงกล จากภาพที่ 3 ประกอบด้วย มอเตอร์ไฟฟ้า รอก สายพาน และอุปกรณ์เลียนแบบเม็ดฝน โดยที่แกนของมอเตอร์จะถูกยึดติดกับรอก ที่มีสายพานผูกยึดกับอุปกรณ์เลียนแบบเม็ดฝน หลักการทำงานคือมอเตอร์จะหมุนซ้าย และขวา เพื่อให้อุปกรณ์เลียนแบบเม็ดฝนเคลื่อนที่ขึ้นและลงได้



ภาพที่ 3 โครงสร้างทางกลของ Kinetics Rain

โครงสร้างทางไฟฟ้า ประกอบด้วยบอร์ดไมโครคอนโทรลเลอร์[3] บอร์ดติดต่อสื่อสารกับอินเทอร์เน็ต และวงจรขับมอเตอร์ โดยไมโครคอนโทรลเลอร์ทำหน้าที่คำนวณเส้นทางเดินสำหรับควบคุมมอเตอร์ไฟฟ้า รับสัญญาณป้อนกลับจากโพเทนชิโอมิเตอร์ และตัวควบคุม PID สำหรับควบคุมตำแหน่ง ซึ่งโครงสร้างส่วนใหญ่แสดงดังภาพที่ 4



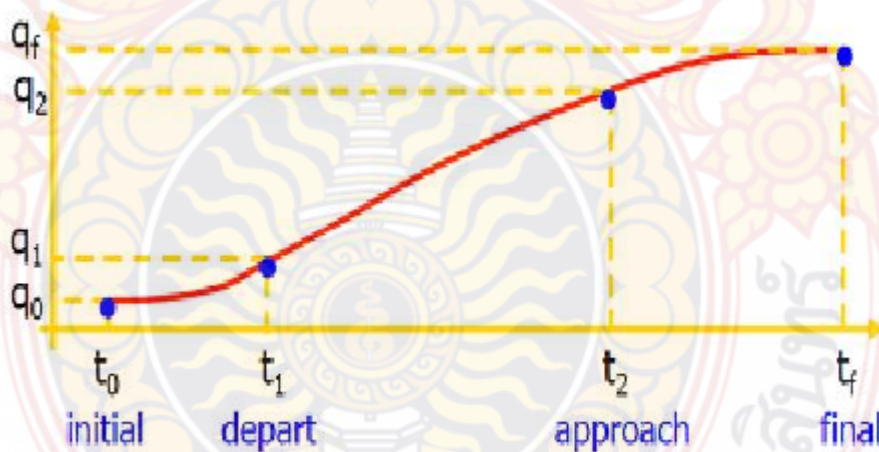
ภาพที่ 4 โครงสร้างทางไฟฟ้าของ Kinetics Rain

2. การออกแบบเส้นทางเดิน

การออกแบบเส้นทางเดิน (Trajectory Generator หรือ Command Generator) คือการออกแบบค่าเป้าหมายสำหรับการเคลื่อนที่ของมอเตอร์ให้มีความเป็นไปตามที่ต้องการ โดยการนำพารามิเตอร์ที่มีผลกระทบทต่อการเคลื่อนที่เช่น เวลาเริ่มต้น เวลาปลายทาง ความเร็วเริ่มต้น ค่าตำแหน่งเริ่มต้น ความเร็วปลายทาง และค่าตำแหน่งปลายทาง เริ่มต้นการออกแบบจะใช้พารามิเตอร์ต่าง ๆ และใช้เทคนิคของ Auto Regressive หรือ Trajectory Generator[11,12] เป็นส่วนประกอบสำคัญในกระบวนการควบคุม นั่นคือกระบวนการสร้างคำสั่งการควบคุม หรือค่าเป้าหมาย ซึ่งค่าเป้าหมายจะมีการเปลี่ยนแปลง เมื่อตัวควบคุมส่งสัญญาณควบคุมให้กระบวนการทำงาน และถึงจุดเป้าหมายเดิมแล้ว ค่าเป้าหมายใหม่จะเปลี่ยนแปลงค่าเป็นค่าเป้าหมายใหม่ โดยการป้อนให้กับตัวควบคุม(Controller) และสั่งให้อุปกรณ์สุดท้ายเกิดการเคลื่อนที่ตามเส้นทางเดินของการเคลื่อนที่ประกอบด้วยค่าตำแหน่ง ความเร็ว ความเร่งที่ต้องการ

โดยทั่วไปแล้ว การออกแบบเส้นทางเดินมีกระบวนการดังต่อไปนี้

1. **Path planning** คือการวางแผนเกี่ยวกับเส้นทางเดินการเคลื่อนที่ที่ต้องการควบคุม ให้เป็นไปตามที่ต้องการ เช่นเส้นตรง(linear) วงกลม (circle) หรือส่วนของวงกลม(arc) และเส้นโค้ง(curve) เป็นต้น
2. **Trajectory Planning** คือเกี่ยวกับเส้นทางเดินการเคลื่อนที่ที่ต้องการควบคุม ให้เป็นไปตามที่ต้องการ ที่ขึ้นอยู่กับเวลา เพื่อให้การควบคุมเป็นไปอย่างต่อเนื่อง และราบเรียบ



ภาพที่ 5 แสดงเส้นทางเดินการเคลื่อนที่

พิจารณาในภาพที่ 5 เส้นทางเดินการเคลื่อนที่เป็นกราฟ พบว่าหลักการจะคล้ายกับการทำ curve fitting ของกราฟทั้งหมดเข้าด้วยกันให้เป็นฟังก์ชันเดียว ซึ่งทางคณิตศาสตร์เรียกว่าการทำ Interpolation [12] เนื่องจากทราบค่าพารามิเตอร์เพียงค่าเริ่มต้นและค่าสุดท้ายเท่านั้น นั่นคือต้องมีวิธีการที่จะสร้าง profile การควบคุมระหว่างค่าทั้งสองนั้น ฟังก์ชันที่แทน profile การควบคุมจะใช้ฟังก์ชันพหุนาม (polynomial) อย่างเช่น

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

โดยที่

$\theta(t)$ คือค่าเส้นทางเดินของกระบวนการ

a_0, a_1, a_2 และ a_3 คือตัวแปรที่ต้องการหาค่า

t คือค่าเวลาที่เปลี่ยนแปลงตามการเคลื่อนที่

โดย $\theta(t)$ ค่าเส้นทางเดินจะมีค่าเริ่มต้น และค่าเป้าหมายที่ต้องการ ซึ่งค่าเริ่มต้นส่วนใหญ่จะถูกกำหนดให้เท่ากับ $\theta(0)$ และค่าเป้าหมายจะเท่ากับ $\theta(t_f)$ จากนั้นกำหนดให้ t_f เท่ากับเวลาที่เสร็จสิ้นกระบวนการ

จากเงื่อนไขดังกล่าวสามารถเขียนสมการได้ 2 สมการคือ

$$\theta(0) = a_0$$

และ

$$\theta(t_f) = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3$$

จากสมการทั้งสองสามารถคำนวณหาค่าตัวแปรต่าง ๆ ได้ ซึ่งนั่นก็คือค่าตัวแปรกำหนดเส้นทางเดินของกระบวนการได้ โดยที่ $a_0, \theta(t_f)$ และ t_f คือพารามิเตอร์ที่ทราบค่า (Known Value)

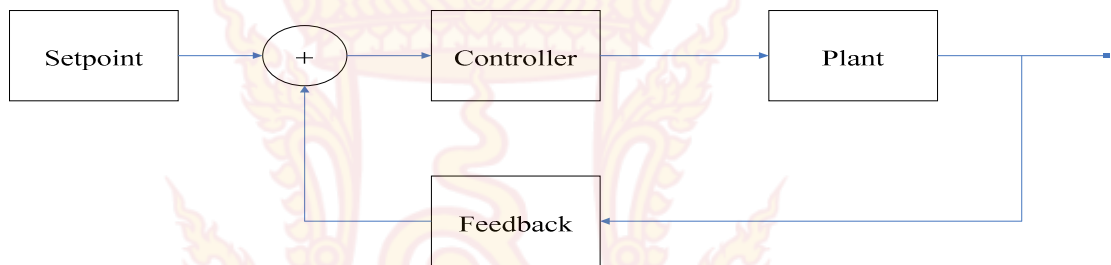
3. ระบบควบคุมพื้นฐาน

ระบบควบคุม (Control System) หมายถึง การควบคุมระบบพลศาสตร์ให้มีค่าเอาต์พุตที่ต้องการ โดยการป้อนค่าอินพุตที่เหมาะสมให้กับระบบ เช่น ระบบควบคุมอุณหภูมิห้องของเครื่องปรับอากาศ หรือแม้แต่ลูกลอยในถังน้ำ ที่ทำหน้าที่เปิด-ปิดน้ำอัตโนมัติ การควบคุมการขับเคลื่อนยานพาหนะเช่นรถยนต์ก็เป็นการควบคุมชนิดหนึ่ง โดยผู้ขับขี่เป็นผู้ควบคุมทิศทางและความเร็ว ซึ่งระบบควบคุมประเภทที่ต้องมีคนเข้ามาเกี่ยวข้องนี้ถือว่าเป็น ระบบควบคุมไม่อัตโนมัติ (manual control) แต่ทฤษฎีระบบควบคุมจะครอบคลุมเฉพาะการวิเคราะห์และออกแบบระบบควบคุมอัตโนมัติ(automatic control)เท่านั้น ระบบควบคุมยังอาจแบ่งออกได้เป็นระบบควบคุมวงเปิด (open-loop control) คือระบบควบคุมที่ไม่ได้ใช้สัญญาณจากเอาต์พุต มาป้อนกลับมายังอินพุตเพื่อเปรียบเทียบ ส่วนระบบควบคุมวงปิด(closed-loop control) หรือ ระบบป้อนกลับ (feedback control) นั้นจะนำค่าที่วัดได้จากเอาต์พุตมาคำนวณค่าการควบคุม นอกจากนี้ยังอาจแบ่งได้ตามคุณลักษณะของระบบเช่น เป็นเชิงเส้น(linear) ไม่เป็นเชิงเส้น(nonlinear) ,แปรเปลี่ยนตามเวลา

(time-varying) ไม่เปลี่ยนแปลงตามเวลา(time-invariant) และเวลาต่อเนื่อง(Continuous time) เวลาไม่ต่อเนื่อง(Discontinuous time)



ภาพที่ 6 ระบบควบคุมวงเปิด



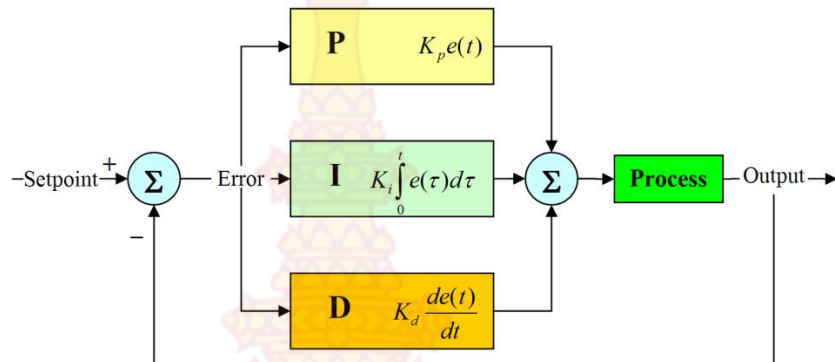
ภาพที่ 7 ระบบควบคุมวงปิด

4. ระบบควบคุมแบบ PID

การควบคุมแบบ PID คือการควบคุมแบบลูปปิด โดยมีการนำเอาสัญญาณทางเอาพุทกลับมาป้อนกลับให้กับด้านอินพุทของตัวควบคุม และนำสัญญาณดังกล่าวมาเปรียบเทียบกับค่าเป้าหมาย (Setpoint) ส่งให้เกิดค่าผิดพลาด (Error) จากนั้นจะส่งต่อให้กับตัวควบคุม PID เพื่อคำนวณสัญญาณควบคุม นำไปควบคุม Plant โดยที่ PID นั้นย่อมาจาก

1. "P"มาจากคำว่า Proportional control action (P - Action)
2. "I"มาจากคำว่า...Integral control action (I-Action)
3. "D"มาจากคำว่า....Derivative control action (D-Action)

ระบบควบคุมแบบสัดส่วน ปริพันธ์ และอนุพันธ์(PID controller) เป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ วิธีคำนวณของ PID ขึ้นอยู่กับสามตัวแปรคือสัดส่วน ปริพันธ์ และอนุพันธ์ ค่าสัดส่วนกำหนดจากผลของความผิดพลาดในปัจจุบันค่าปริพันธ์ กำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งผ่านพ้นไป และค่าอนุพันธ์กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการ



ภาพที่ 8 แผนภาพบล็อกของการควบคุมแบบพีไอดี

โดยการปรับค่าคงที่ใน PID ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะสมกับที่กระบวนการต้องการได้ การตอบสนองของตัวควบคุมจะอยู่ในรูปของการไหวตัวของตัวควบคุมจนถึงค่าความผิดพลาด ค่าโอเวอร์ชูต (Overshoots) และ ค่าแกว่งของระบบ (Oscillation) วิธี PID ไม่รับประกันได้ว่าจะเป็นระบบควบคุมที่เหมาะสมที่สุดหรือสามารถทำให้กระบวนการมีความเสถียรแน่นอนการประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งถึงสองรูปแบบ ขึ้นอยู่กับกระบวนการเป็นสำคัญ พีไอดีบางครั้งจะถูกเรียกว่าการควบคุมแบบ PI, PD, P หรือ I ขึ้นอยู่กับว่าใช้รูปแบบใดบ้างการควบคุมแบบ PID ได้ชื่อตามการรวมกันของเทอมของตัวแปรทั้งสามดังสมการที่ 1

$$MV(t) = P_{out} + I_{out} + D_{out} \quad (1)$$

โดยที่

P_{out} , I_{out} และ D_{out} เป็นผลของสัญญาณขาออกจากระบบควบคุม PID จากแต่ละเทอมซึ่งนิยามตามรายละเอียดดังนี้

4.1 เทอมของสัดส่วน หรือ P (Proportional control action)

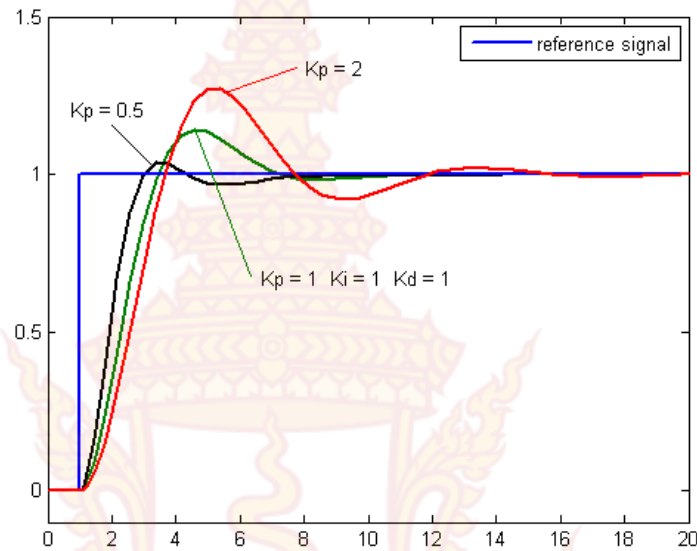
เทอมของสัดส่วน หรือ P (อัตราขยาย) จะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่ K_p หรือที่เรียกว่าอัตราขยายสัดส่วนเทอมของสัดส่วนจะเป็นไปดังสมการที่ 2

$$P_{out} = K_p e(t) \quad (2)$$

โดยที่

P_{out} คือ สัญญาณขาออกของเทอมสัดส่วน

K_p คือ อัตราขยายสัดส่วน, ตัวแปรปรับค่าได้
 e คือ ค่าความผิดพลาด (SP - PV)
 t คือ เวลา



ภาพที่ 9 กราฟ PV ต่อเวลา, K_p กำหนดเป็น 3 ค่า K_i และ K_d คงที่

ผลอัตราขยายสัดส่วนที่สูงค่าความผิดพลาดก็จะเปลี่ยนแปลงมากเช่นกัน แต่ถ้าสูงเกินไประบบจะไม่เสถียรได้ ในทางตรงกันข้าม ผลอัตราขยายสัดส่วนที่ต่ำ ระบบควบคุมจะมีผลตอบสนองต่อกระบวนการน้อยตามไปด้วยดังภาพที่ 9

4.2 เทอมของปริพันธ์ หรือ I (Integral control action)

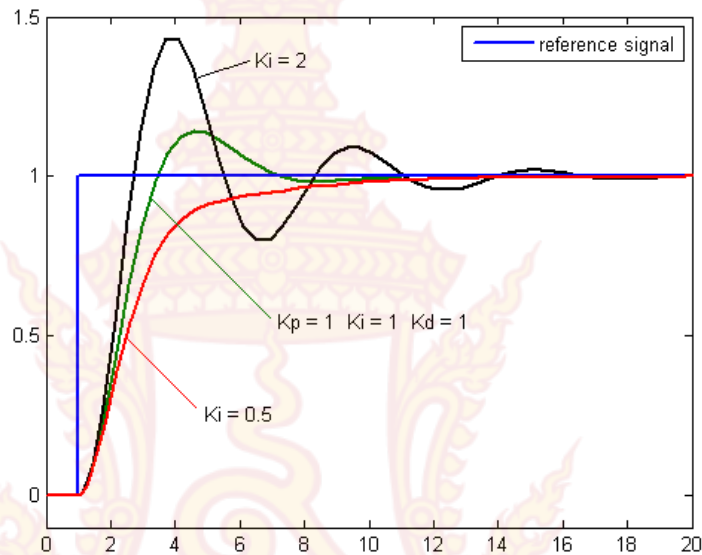
ผลจากเทอมปริพันธ์ หรือ (Reset) เป็นสัดส่วนของขนาดความผิดพลาดและระยะเวลาของความผิดพลาด ผลรวมของความผิดพลาดในทุกช่วงเวลา (ปริพันธ์ของความผิดพลาด) จะให้ออฟเซตสะสมที่ควรจะเป็นในก่อนหน้า ความผิดพลาดสะสมจะถูกคูณโดยอัตราขยายปริพันธ์ ขนาดของผลของเทอมปริพันธ์กำหนดโดยอัตราขยายปริพันธ์ K_i เทอมปริพันธ์จะเป็นไปดังสมการที่ 3

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (3)$$

โดยที่

I_{out} คือ สัญญาณขาออกของเทอมปริพันธ์
 K_i คือ อัตราขยายปริพันธ์, ตัวแปรปรับค่าได้
 e คือ ค่าความผิดพลาด (SP - PV)
 t คือ เวลา
 τ คือ ตัวแปรปริพันธ์หุ่น

เทอมปริพันธ์เมื่อรวมกับเทอมสัดส่วนจะเร่งกระบวนการให้เข้าสู่จุดที่ต้องการและขจัดความผิดพลาดที่เหลืออยู่ที่เกิดจากการใช้เพียงเทอมสัดส่วน แต่อย่างไรก็ตาม เทอมปริพันธ์เป็นการตอบสนองต่อความผิดพลาดสะสมในอดีต จึงสามารถทำให้เกิดโอเวอร์ชูตได้ (ข้ามจุดที่ต้องการและเกิดการหันเหไปทางทิศทางอื่น) ดังภาพที่ 10



ภาพที่ 10 กราฟ PV ต่อเวลา, K_i กำหนดเป็น 3 ค่า K_p และ K_d คงที่

2.4.3 เทอมของอนุพันธ์ หรือ D (Derivative control action)

เทอมของอนุพันธ์ หรือ D อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการนั้น คำนวณหาจากความชันของความผิดพลาดทุกๆเวลา คือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา และคูณด้วยอัตราขยายอนุพันธ์ K_d ขนาดของผลของเทอมอนุพันธ์ (อัตรา) ขึ้นกับ อัตราขยายอนุพันธ์ K_d เทอมอนุพันธ์เป็นไปดังสมการที่ 4

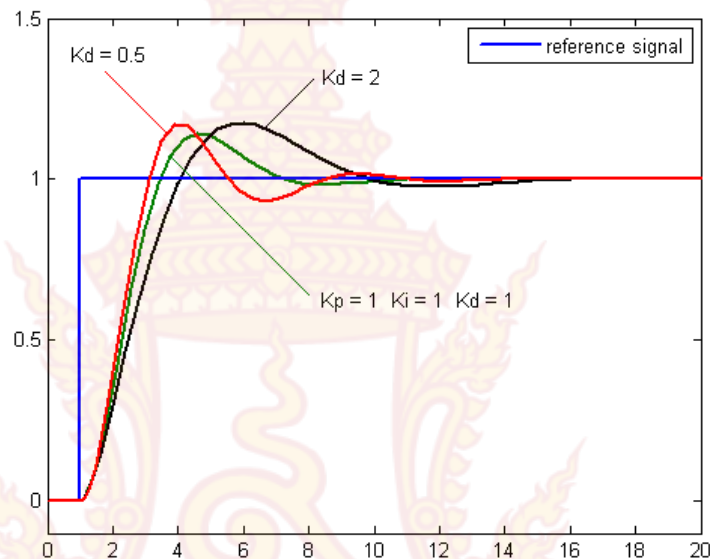
$$D_{out} = K_d \frac{d}{dt} e(t) \quad (4)$$

โดยที่

- D_{out} คือ สัญญาณขาออกของเทอมอนุพันธ์
- K_d คือ อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้
- e คือ ค่าความผิดพลาด (SP - PV)
- t คือ เวลา

เทอมอนุพันธ์จะชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุมและด้วยผลนี้จะช่วยให้ระบบควบคุมเข้าสู่จุดที่ต้องการ ดังนั้นเทอมอนุพันธ์จะใช้ในการลดขนาดของโอเวอร์ชูต

ที่เกิดจาเทอมปริพันธ์และทำให้เสถียรภาพของการรวมกันของระบบควบคุมดีขึ้น แต่อย่างไรก็ตามอนุพันธ์ของสัญญาณรบกวนที่ถูกขยายในระบบควบคุมจะไวมากต่อการรบกวนในเทอมของความผิดพลาดและสามารถทำให้กระบวนการไม่เสถียรได้ถ้าสัญญาณรบกวนและอัตราขยายอนุพันธ์มีขนาดใหญ่เพียงพอดังภาพที่ 11



ภาพที่ 11 กราฟ PV ต่อเวลา, K_d กำหนดเป็น 3 ค่า K_p และ K_i คงที่

พารามิเตอร์เทอมสัดส่วน ปริพันธ์และอนุพันธ์ จะนำมารวมกันเป็นสัญญาณขาออกของการควบคุมแบบ PID ถูกกำหนดให้เป็น $u(t)$ เท่ากับสัญญาณขาออกดังสมการที่ 5 เป็นสมการสุดท้ายของวิธี PID

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (5)$$

ซึ่งสมการที่ (5) จะถูกนำไปพัฒนาเป็นโปรแกรมควบคุม PID ต่อไป

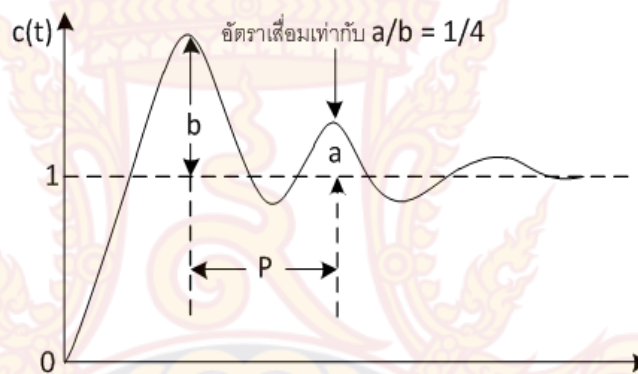
การปรับแต่งตัวควบคุมพีไอดีแบบZiegler และ Nichols

การปรับแต่งระบบควบคุมโดยใช้วิธี Ziegler-Nichols นี้ ถูกนำเสนอขึ้นเป็นครั้งแรกโดย J.G. Ziegler และ N. B. Nichols ในการประชุมของสมาคม American Society of Mechanical Engineering ณ มลรัฐนิวยอร์ก เมื่อเดือนธันวาคม ปี ค.ศ. 1941 นับจากนั้นเป็นต้นมา เทคนิคดังกล่าวซึ่งรู้จักกันในชื่อ “Ziegler-Nichols One-quarter Wave Decay” ก็ได้รับความนิยมและถูกนำมาใช้กันอย่างแพร่หลายจนถึงปัจจุบัน บล็อกไดอะแกรมแสดงการทำงานของระบบควบคุมแบบพีไอดีสามารถแสดงดังภาพที่ 8 โดยการหาค่าพารามิเตอร์ของกระบวนการ ซึ่งก็คือค่า K_p , T_i และ T_d จะขึ้นอยู่กับ

คุณลักษณะของผลตอบสนองชั่วขณะ (Transient Response) ของกระบวนการที่ถูกควบคุม ซึ่งมี 2 วิธี ดังนี้

1. การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Process Reaction Curve
2. การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method

โดยทั้งสองวิธีนั้นต่างก็มีจุดมุ่งหมายเดียวกัน คือพยายามที่จะปรับแต่งเพื่อให้ผลตอบสนองของกระบวนการต่อสัญญาณอินพุตแบบสัญญาณระดับหนึ่งหน่วย มีอัตราส่วนระหว่างค่าพุ่งเกินสูงสุด (Maximum Overshoot) ครั้งแรก (b) ต่อครั้งที่สอง (a) เท่ากับ 4 ต่อ 1 หรือมีอัตราเสื่อม (One-Quarter Ratio) เท่ากับ 25% ดังแสดงในภาพที่ 12



ภาพที่ 12 ผลตอบสนองของกระบวนการที่มีอัตราเสื่อมเท่ากับ 1/4(25%)

2.5.1 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Process Reaction Curve

วิธีการคำนวณหาค่าพารามิเตอร์ของกระบวนการการควบคุมแบบพีไอดีด้วยวิธีแบบ Process Reaction Curve นี้จะอาศัยการพิจารณาจากผลตอบสนองของกระบวนการเมื่อป้อนสัญญาณอินพุตที่เป็นสัญญาณระดับหนึ่งหน่วยโดยในกรณีนี้กระบวนการหรือระบบต้องไม่มีตำแหน่งโพลอยู่ที่จุดกำเนิด หรือต้องไม่มีตำแหน่งโพลเชิงซ้อน (Dominant Complex Conjugate Poles) และไม่ได้ต่อร่วมกับอุปกรณ์ควบคุมกระบวนการ ดังนั้นผลตอบสนองของกระบวนการในเชิงเวลาจะอยู่ในลักษณะตัว S ดังภาพที่ 13 ซึ่งวิธีการนี้จะไม่สามารถใช้ได้กับระบบหรือกระบวนการที่ไม่มีผลตอบสนองในลักษณะตัว S เช่นกัน

วิธีการนี้จะแยกการพิจารณาค่าคงที่ได้ 2 ค่า คือ ค่าเวลาหน่วง (Delay Time, L) และค่าคงที่เวลา (Time Constant, τ) ซึ่งหาได้โดยการลากเส้นตรงให้สัมผัสกับจุดที่มีความชันสูงสุด และจะได้เส้นตรงเส้นนี้ตัดกับแกนเวลาและแกนเอาต์พุต $C(t)$ ที่จุด $C(t) = K$ ดังนั้นเมื่ออาศัยการประมาณฟังก์ชันด้วยระบบอันดับหนึ่งที่มีการหน่วงเวลาจึงสามารถหาฟังก์ชันถ่ายโอน (Transfer Function) ได้ ดังสมการที่ 8

$$\frac{C(s)}{M(s)} = \frac{K e^{-sL}}{\tau s + 1} \quad (8)$$

จากวิธี Process Reaction Curve สามารถคำนวณหาค่าพารามิเตอร์ของการควบคุมแต่ละชนิดที่ให้ผลตอบสนองของกระบวนการที่เหมาะสม โดยสรุปได้ดังตารางที่ 2-1 และสามารถสรุปข้อดีข้อเสียของวิธีการดังกล่าวได้ดังต่อไปนี้

ตารางที่ 1 ค่าพารามิเตอร์ของการควบคุมแต่ละชนิดซึ่งหาได้จากวิธี Process Reaction Curve

ชนิดของการควบคุม	K_p	T_i	T_d
แบบพี	T / KL	-	-
แบบพีไอ	$0.9T / KL$	$3.3L$	-
แบบพีไอดี	$1.2T / KL$	$2L$	$0.5L$

1. ข้อดีของการใช้วิธี Process Reaction Curve

- สามารถทำการทดสอบเพียงครั้งเดียว
- สามารถคำนวณหาค่าพารามิเตอร์ของกระบวนการได้ง่าย
- ไม่จำเป็นต้องใช้วิธีการลองผิดลองถูก

2. ข้อเสียของการใช้วิธี Process Reaction Curve

- เนื่องจากการทดสอบกระบวนการกระทำภายใต้สถานะลูปเปิด (Open-loop) ดังนั้นหากมีการเปลี่ยนแปลงค่าโหลดในขณะที่ทำการทดสอบ ก็จะทำให้ผลการทดสอบที่ได้มีความคลาดเคลื่อนเกิดขึ้น

- หากเครื่องมือตรวจวัดที่ใช้ในการทดสอบมีสัญญาณรบกวนหรือกราฟผลตอบสนองที่ได้จากการทดสอบมีสเกลขนาดเล็กไม่เหมาะสมก็จะทำให้การหาค่าความชันของกราฟเส้นตรงทำได้ยาก ซึ่งเป็นสาเหตุหนึ่งที่ทำให้เกิดค่าความผิดพลาดขึ้นเช่นกัน

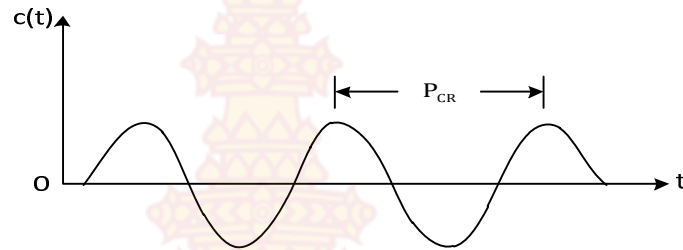
- วิธีการนี้มีความไวต่อความผิดพลาดจากการสอบเทียบของอุปกรณ์ควบคุม

- การใช้ค่าพารามิเตอร์ในตารางที่ 2-1 มีแนวโน้มที่จะทำให้ผลตอบสนองของกระบวนการเกิดการแกว่งได้

2.5.2 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method

วิธีการนี้จะเป็นการคำนวณหาค่าพารามิเตอร์ของกระบวนการโดยอาศัยผลตอบสนองของกระบวนการซึ่งถูกควบคุมแบบพีต่อสัญญาณอินพุตที่เป็นสัญญาณระดับหนึ่งหน่วย โดยทำการปรับค่าอัตราขยายการควบคุมแบบพี หรือ K_p ไปจนกระทั่งผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ดังแสดงในภาพที่ 14 ดังนั้นวิธีการนี้จึงไม่สามารถใช้ได้กับกระบวนการที่มีผลตอบสนองที่ไม่เกิดการแกว่งอย่างต่อเนื่อง ซึ่งค่าพารามิเตอร์โดยทั่วไปของชนิดการควบคุมแต่ละแบบที่คำนวณได้ สามารถนำมาสรุปได้ดังตารางที่ 2-2 เมื่อ K_{CR} คือ อัตราขยายที่ทำให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Critical Gain) และ P_{CR} คือ คาบเวลาของการแกว่งอย่างต่อเนื่อง (Oscillation Period) แต่เนื่องจากต้องปรับแต่งค่าพารามิเตอร์ของกระบวนการให้มีการทำงาน

ใกล้เคียงกับความไม่เสถียรและยังใช้เวลาค่อนข้างนาน โดยเฉพาะกับกระบวนการที่มีอัตราห้วงหลายๆ เช่นกระบวนการควบคุมอุณหภูมิตั้งนั้นด้วยเหตุดังกล่าว จึงทำให้วิธีการ Ultimate Method นี้ไม่เป็นที่นิยมใช้กันมากนัก



ภาพที่ 13 การหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method
ตารางที่ 2 ค่าพารามิเตอร์ของการควบคุมแต่ละชนิดซึ่งหาได้จากวิธี Ultimate Method

ชนิดของการควบคุม	K_p	T_i	T_d
แบบพี	$0.5 K_{CR}$	-	-
แบบพีไอ	$0.45 K_{CR}$	$P_{CR} / 1.2$	-
แบบพีดี	$0.6 K_{CR}$	$0.125 P_{CR}$	-
แบบพีไอดี	$0.6 K_{CR}$	$0.5 P_{CR}$	$0.125 P_{CR}$

จากวิธีการคำนวณหาค่าพารามิเตอร์ของกระบวนการควบคุมแบบพีไอดี โดยใช้วิธีของ Ziegler-Nichols ทั้งสองวิธีที่ได้กล่าวถึงข้างต้นแล้วนั้น ค่าพารามิเตอร์ที่ได้ไม่สามารถนำไปใช้ได้ทันที ทั้งนี้เนื่องจากค่าพารามิเตอร์ดังกล่าวเป็นค่าที่ได้จากประมาณให้ใกล้เคียงเท่านั้น รวมทั้งผลตอบสนองของกระบวนการยังมีค่าพุ่งเกินสูงสุดถึง 25% อีกด้วย ดังนั้นการนำไปใช้งานในทางปฏิบัติ จึงควรทำการปรับละเอียดค่าพารามิเตอร์ต่างๆ เหล่านี้อีกครั้ง

5. การแปลงลาปลาซ(Laplace Transform)

การแปลงลาปลาซทรานส์ฟอร์ม (Laplace transform) เป็นการแปลงจากฟังก์ชันเชิงเวลาไปเป็นอีกฟังก์ชันหนึ่งเชิง เอสโดเมน (S-domain)

เป็นการแปลงฟังก์ชันใด ๆ ในโดเมนเวลาให้เป็นฟังก์ชันในโดเมนความถี่เชิงซ้อนหรือโดเมน

$$F(s) = L\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt$$

โดยที่

L คือสัญลักษณ์ในการแปลงลาปลาซ

$f(t)$ ฟังก์ชันใดๆที่แปรตามเวลา ค่าการแปลงลาปลาซ

$F(s)$ ฟังก์ชันที่แปลงเป็นลาปลาซในโดเมนความถี่เชิงซ้อน

S คือตัวแปรเชิงซ้อนที่มีค่าเป็น $s = \sigma + j\omega$

σ คือส่วนจริงของตัวแปรเชิงซ้อน

ω คือส่วนจินตภาพของตัวแปรเชิงซ้อน

คุณสมบัติการแปลงลาปลาซ

1. การคูณหรือหารด้วยค่าคงที่

$$L[f(t)] = F(s) \implies L[af(t)] = aF(s)$$

2. การหารด้วยค่าคงที่

$$L[f(t)] = F(s) \implies L\left[\frac{f(t)}{b}\right] = \frac{F(s)}{b}$$

3. การบวกและการลบ

$$L[f(t)] = F(s) \implies L[f_1(t) \pm f_2(t)] = F_1(s) \pm F_2(s)$$

4. การอนุพันธ์ฟังก์ชันที่แปรตามเวลา

$$L[f(t)] = F(s) \implies L\left[\frac{df(t)}{dt}\right] = sF(s) - f(0)$$

$$L[f(t)] = F(s) \implies L\left[\frac{df^2(t)}{dt^2}\right] = s^2F(s) - sf(0) - \frac{df(0)}{dt}$$

5. การอินทิเกรตฟังก์ชันที่แปรตามเวลา

$$L[f(t)] = F(s) \implies L\left[\int f(t)dt\right] = \frac{F(s)}{s} - \frac{\int f(0)dt}{s}$$

ตารางที่ 3 ตารางการแปลงลาปลาซ

ลำดับที่	F(s)	f(t) = L ⁻¹ {F(s)}
1	$\frac{1}{s}$	1
2	$\frac{1}{s^2}$	t
3	$\frac{(n-1)!}{s^n}$	t ⁿ⁻¹ ; n = 1, 2, 3, ...
4	$\frac{1}{2}\sqrt{\pi}s^{-3/2}$	t ^{1/2}
5	$\sqrt{\pi}s^{-1/2}$	t ^{-1/2}
6	$\frac{(1 \cdot 3) \cdot (2n-1)\sqrt{\pi}}{2^n} s^{-(n+1)/2}$	t ^{(n-1)/2} ; n = 1, 2, 3, ...
7	$\frac{1}{s-a}$	e ^{at}
8	$\frac{a}{s^2+a^2}$	sin(at)
9	$\frac{s}{s^2+a^2}$	cos(at)
10	$\frac{a}{s^2-a^2}$	sinh(at)

11	$\frac{s}{s^2 - a^2}$	$\cosh(at)$
12	$\frac{2as}{(s^2 + a^2)^2}$	$t \sin(at)$
13	$\frac{s^2 - a^2}{(s^2 + a^2)^2}$	$t \cos(at)$
14	$\frac{(n-1)!}{(s-a)^n}$	$t^{n-1} e^{at}$
15	$\frac{a}{(s-b)^2 + a^2}$	$e^{bt} \sin(at)$
16	$\frac{s-b}{(s-b)^2 + a^2}$	$e^{bt} \cos(at)$
17	$\frac{2a^3}{(s^2 + a^2)^2}$	$\sin(at) - at \cos(at)$
18	$\frac{1}{1+as}$	$\frac{1}{a} e^{-t/a}$
19	$\frac{1}{s-a}$	$\frac{1}{a} (e^{at} - 1)$
20	$\frac{1}{s+1+as}$	$1 - e^{-t/a}$
21	$\frac{1}{(1+as)^2}$	$\frac{1}{a^2} t e^{-t/a}$
22	$\frac{1}{(s-a)(s-b)}$	$\frac{e^{at} - e^{bt}}{a-b}$
23	$\frac{1}{(1+as)(1+bs)}$	$\frac{e^{-t/a} - e^{-t/b}}{a-b}$
24	$\frac{s}{(s-a)^2}$	$(1+at)e^{at}$
25	$\frac{s}{(1+as)^2}$	$\frac{1}{a^3} (a-t)e^{-t/a}$
26	$\frac{s}{(s-a)(s-b)}$	$\frac{ae^{at} - be^{bt}}{a-b}$; เมื่อ $a \neq b$
27	$\frac{s}{(1+as)(1+bs)}$	$\frac{ae^{-t/b} - be^{-t/a}}{a(a-b)}$
28	$\frac{1}{s^2(s-a)}$	$\frac{1}{a^2} (e^{at} - at - 1)$
29	$\frac{2a^2}{s^2 + 4a^2}$	$\sin^2(at)$
30	$\frac{2a^2}{s^2 - 4a^2}$	$\sinh^2(at)$

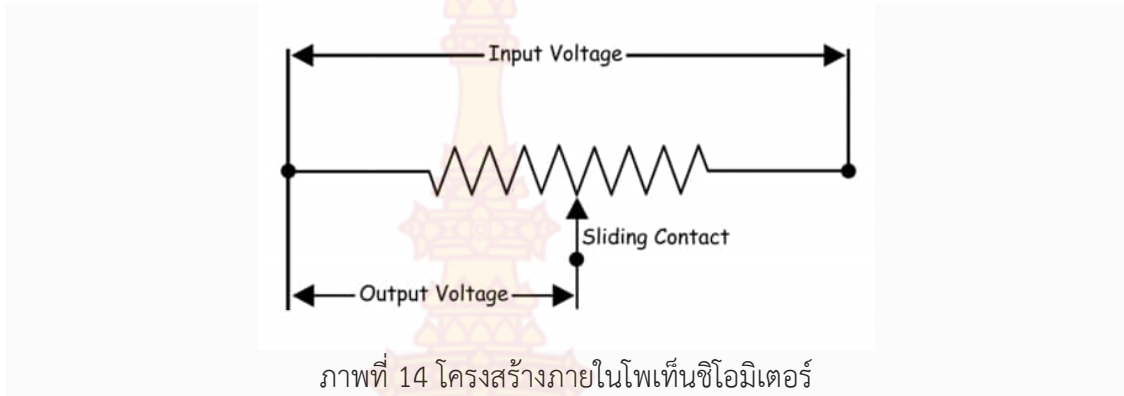
31	$\frac{a^3}{s^4 + a^4}$	$\frac{1}{\sqrt{2}} \left(\cosh \frac{at}{\sqrt{2}} \sin \frac{at}{\sqrt{2}} - \sinh \frac{at}{\sqrt{2}} \cos \frac{at}{\sqrt{2}} \right)$
32	$\frac{a^2 s}{s^4 + a^4}$	$\sin \frac{at}{\sqrt{2}} \sinh \frac{at}{\sqrt{2}}$
33	$\frac{as^2}{s^4 + a^4}$	$\frac{1}{\sqrt{2}} \left(\cos \frac{at}{\sqrt{2}} \sinh \frac{at}{\sqrt{2}} + \sin \frac{at}{\sqrt{2}} \cosh \frac{at}{\sqrt{2}} \right)$
34	$\frac{s^3}{s^4 + a^4}$	$\cos \frac{at}{\sqrt{2}} \cosh \frac{at}{\sqrt{2}}$
35	$\frac{a^3}{s^4 - a^4}$	$\frac{1}{2} (\sinh(at) - \sin(at))$
36	$\frac{a^2 s}{s^4 - a^4}$	$\frac{1}{2} (\cosh(at) - \cos(at))$
37	$\frac{as^2}{s^4 - a^4}$	$\frac{1}{2} (\sinh(at) + \sin(at))$
38	$\frac{s^3}{s^4 - a^4}$	$\frac{1}{2} (\cosh(at) + \cos(at))$
39	$\frac{2a^2 s}{s^4 + 4a^4}$	$\sin(at) \sinh(at)$
40	$\frac{s^2 - 2a^2}{s^4 + 4a^4}$	$\cos(at) \sinh(at)$
41	$\frac{s^2 + 2a^2}{s^4 + 4a^4}$	$\sin(at) \cosh(at)$
42	$\frac{s^3}{s^4 + 4a^4}$	$\cos(at) \cosh(at)$
43	$\frac{as^2}{(s^2 + a^2)^2}$	$\frac{1}{2} (\sin(at) + at \cos(at))$
44	$\frac{s^3}{(s^2 + a^2)^2}$	$\cos(at) - \frac{at}{2} \sin(at)$
45	$\frac{a^3}{(s^2 - a^2)^2}$	$\frac{1}{2} (at \cosh(at) - \sinh(at))$
46	$\frac{as}{(s^2 - a^2)^2}$	$\frac{t}{2} \sinh(at)$
47	$\frac{as^2}{(s^2 - a^2)^2}$	$\frac{t}{2} (\sinh(at) + at \cosh(at))$
48	$\frac{s^3}{(s^2 - a^2)^2}$	$\cosh(at) + \frac{at}{2} \sinh(at)$

6. อุปกรณ์ตรวจวัด (Sensor and Transducer)

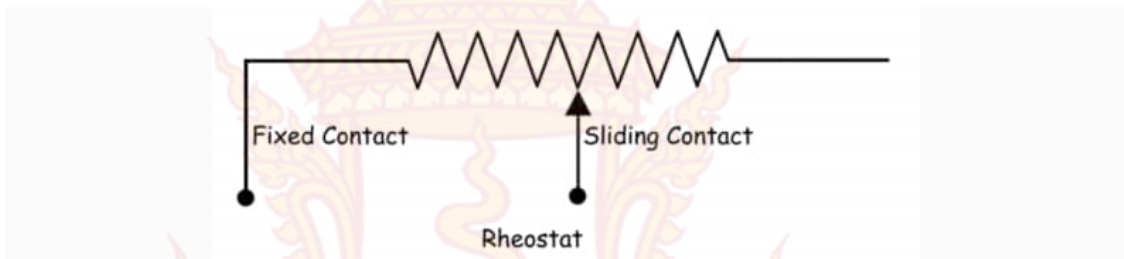
อุปกรณ์ตรวจวัดคือตัวตรวจจับสัญญาณ (Sensor) ตัวทรานสดิวเซอร์ (Transducer) หรืออุปกรณ์แปลง (Transmitter) หรือเครื่องวัดสัญญาณอื่น ๆ ในกระบวนการเพื่อนำสัญญาณที่ได้ไปใช้เป็นตัวแปรในการควบคุม โดยสัญญาณขาออกของอุปกรณ์ตรวจวัดตามทั่วไปจะเป็นสัญญาณมาตรฐานทางอุตสาหกรรม เช่นสัญญาณไฟฟ้ากระแสตรง 4 ถึง 20 มิลลิแอมป์ สัญญาณแรงดันไฟฟ้ากระแสตรง 0 ถึง 5 โวลต์ อุปกรณ์ตรวจวัดจะเป็นตัวตรวจวัดปริมาณทางฟิสิกส์เชิงกล และเชิงไฟฟ้าของตัวแปรกระบวนการ ส่วนอุปกรณ์ส่งสัญญาณจะทำหน้าที่เปลี่ยนปรากฏการณ์เหล่านี้เป็นสัญญาณที่สามารถส่งออกมา ซึ่งสัญญาณที่ได้นั้นจะมีความสัมพันธ์กับตัวแปรในกระบวนการ อุปกรณ์วัดและอุปกรณ์ส่งสัญญาณจะมีเทอมสำคัญประกอบกัน 3 เทอมคือ พิกัดของเครื่องมือวัด ถูกกำหนดโดยค่าสูงสุดและค่าต่ำสุดของตัวแปรกระบวนการที่ทำการตรวจวัด

การควบคุมกระบวนการต่าง ๆ ทางอุตสาหกรรมของอุปกรณ์ตรวจวัด และส่งสัญญาณแบบอัตโนมัติ นั้น เป็นการรักษาระดับปริมาณในการทำงานทางฟิสิกส์ ซึ่งรูปแบบการควบคุมจะใช้ตัวควบคุมที่ทำงานด้วยระบบอิเล็กทรอนิกส์เป็นเครื่องควบคุมกระบวนการแทนมนุษย์ ดังนั้น ในการทำงานจึงจำเป็นต้องมีเครื่องมือหรืออุปกรณ์ตรวจวัดที่ทำหน้าที่ในการตรวจจับการเปลี่ยนแปลงของกระบวนการต่าง ๆ ที่ควบคุมโดยสัญญาณที่วัดได้จะนำไปใช้ในการตัดสินใจสร้างลักษณะควบคุมไปที่เหมาะสมต่อไป อุปกรณ์เหล่านี้อาจจะได้แก่ ตัวตรวจจับสัญญาณ ทรานสดิวเซอร์ หรืออุปกรณ์วัดสัญญาณอื่น ๆ ที่ทำหน้าที่ตรวจวัดการเปลี่ยนแปลงของกระบวนการ หรือตัวแปรกระบวนการและส่งสัญญาณที่วัดได้ให้กับตัวควบคุมเพื่อใช้เป็นเกณฑ์เลือกสัญญาณควบคุมที่เหมาะสม ตัวตรวจจับสัญญาณในระบบอัตโนมัติใด ๆ ตัวตรวจจับสัญญาณเป็นตัวเชื่อมอันดับแรกในสายงานของการควบคุมเพื่อตรวจสอบว่าอะไรจะเกิดขึ้นเหมือนกับการตรวจสอบไม่ให้เกิดการเปลี่ยนไปจากข้อกำหนด เครื่องมือวัด (Measuring Instrument) คือการตรวจจับสัญญาณ (Sensor) และตัวแสดงค่า (Indicator) การวัดสามารถกระทำได้ 2 วิธีคือ วิธีแรกคุณสมบัติทางฟิสิกส์บางประการของตัวตรวจจับสัญญาณสามารถใช้ประโยชน์ได้โดยตรงหรือเปลี่ยนแปลงโดยให้เปรียบเทียบกับสิ่งที่เรารู้แล้วแต่ปรับค่าให้มีคุณรูปที่แท้จริงเหมือนกัน ในวิธีที่ 2 นี้กรรมวิธีของการวัดเกี่ยวกับการกำหนดค่าที่แน่นอนให้เท่ากันระหว่างปริมาณค่า 2 ค่าที่คล้ายกันตัวตรวจจับสัญญาณต้องมีพื้นฐานที่สำคัญคือต้องรู้ว่าผลอะไรเกิดขึ้นเมื่อ ตัวตรวจจับสัญญาณกำลังวัดปริมาณค่าอันหนึ่งในกรณีที่ดีที่สุด ตัวตรวจจับสัญญาณไม่ควรจะเปลี่ยนสถานะที่ดำรงอยู่ก่อนที่จะทำการวัด ในทางปฏิบัติจะกระทำได้ยากมากตัวตรวจจับสัญญาณบางชนิดผลตอบสนองต่อคุณสมบัติอื่นเสียมากกว่าสิ่งพื้นฐานที่ต้องการ เช่นโวลต์มิเตอร์จะมีผลตอบสนอง (Response) ต่อค่าเฉลี่ย (Average) ของสัญญาณกระแสสลับหรือค่าพิก (Peak) ดังนั้นมีความสำคัญในการเลือกตรวจจับสัญญาณจะต้องมั่นใจว่ามันจะตอบสนองต่อคุณสมบัติที่เราต้องการเท่านั้นความเร็วของผลตอบสนองซึ่งตรวจจับสัญญาณจะต้องตอบสนองอย่างรวดเร็วต่อการเปลี่ยนสถานะของกระบวนการ

ในการควบคุมการเคลื่อนที่ของ Kinetics Rain นี้ ใช้โพเทนซีโอมิเตอร์ทำหน้าที่เป็นอุปกรณ์ตรวจวัดสัญญาณการเคลื่อนที่ดังภาพที่ 15 โดยทำการเปลี่ยนตำแหน่งของหน้าสัมผัสแบบเลื่อนบนตัวต้านทาน เพื่อใช้แบ่งแรงดันไฟฟ้าผันแปรจากตัวต้านทาน ซึ่งแรงดันไฟฟ้าขาออก คือแรงดันไฟฟ้าตกระหว่างหน้าสัมผัสคงที่และส่วนที่เลื่อนดังภาพที่ 16



ภาพที่ 14 โครงสร้างภายในโพเทนชิโอมิเตอร์



ภาพที่ 15 แสดงการเลื่อน หรือการเปลี่ยนแปลงค่าความต้านทาน

7. มอเตอร์ไฟฟ้ากระแสตรง หรือดีซีมอเตอร์

มอเตอร์ไฟฟ้ากระแสตรง หลักการทำงานคือขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบ ๆ แกนขั้วแม่เหล็กขดลวด ทำหน้าที่รับกระแสจากภายนอก เพื่อสร้างแรงเส้นแม่เหล็กให้เกิดขึ้นและเส้นแรงแม่เหล็กนี้จะเกิดการหักล้าง และเสริมกันกับสนามแม่เหล็กของอาร์เมเจอร์ทำให้เกิดแรงบิดขึ้น ประกอบด้วยตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ ตัวหมุนนี้ทำให้เกิดกำลังงานแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกันคือ

- แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์เมเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานี้จะวางอยู่บนแบร็ง เพื่อบังคับให้หมุนอยู่ในแนวหนึ่งไม่มีการสั่นสะเทือนได้

- แกนเหล็กอาร์เมเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาจฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์เมเจอร์ซึ่งสร้างแรงบิด (Torque)

- คอมมิวเตเตอร์ (Commutation) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์จะมีร่องสำหรับใส่ปลายสาย ของขดลวดอาร์เมเจอร์ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลลาเป็นรูปทรงกระบอกมีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยังขดลวดอาร์เมเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็กดังกล่าวเรียกว่าปฏิกิริยามอเตอร์ (Motor action)

- ขดลวดอาร์เมเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอต (Slot) ของแกน

อาร์เมเจอร์ขนาดของขดลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ

หลักการการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

หลักการการทำงานของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะแปร่งผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์เมเจอร์ สร้างสนามแม่เหล็กขึ้นและกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามเหล็ก (Field Coil) สร้างขั้วเหนือ – ใต้ ขึ้นจะเกิดสนามแม่เหล็ก 2 สนามในขณะเดียวกัน ตามคุณสมบัติเส้นแรงแม่เหล็กจะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวกันจะเสริมแรงกันทำให้เกิดแรงบิดในขดลวดอาร์เมเจอร์ซึ่งวางแกนเพลลาและแกนเพลลาเหล่านี้สวมกับตลับลูกปืนของมอเตอร์ทำให้อาร์เมเจอร์นี้หมุนได้ ขณะที่ตัวอาร์เมเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุนที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกริยาต่อกันทำให้ขดลวดอาร์เมเจอร์หรือโรเตอร์หมุนไปนั้นเป็นไปตามกฎซ้ายของเฟลมมิง (Fleming left hand rule)

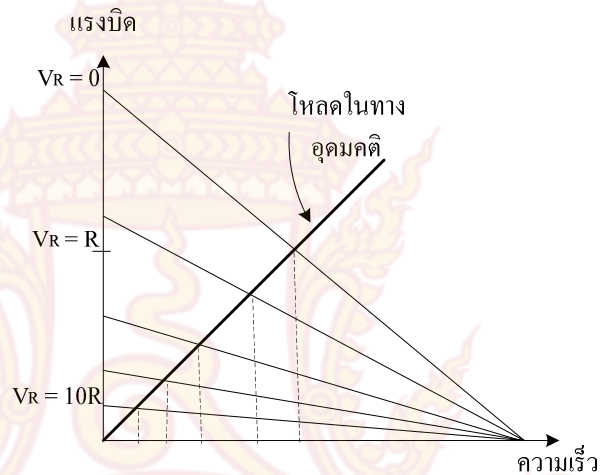
ชนิดของมอเตอร์ไฟฟ้ากระแสตรง

1. มอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม คือมอเตอร์ที่ต่อขดลวดสนามแม่เหล็กอนุกรมกับอาร์เมเจอร์ของมอเตอร์ชนิดนี้ว่า ซีรีส์ฟิลด์ (Series Field) มีคุณลักษณะที่ดีคือให้แรงบิดสูงนิยมใช้เป็นต้นกำลังของรถไฟฟ้ารถยกของเครนไฟฟ้า ความเร็วรอบของมอเตอร์อนุกรมเมื่อไม่มีโหลดความเร็วจะสูงมากแต่ถ้ามีโหลดมาต่อความเร็ว ก็จะลดลงตามโหลด โหลดมากหรือทำงานหนักความเร็วลดลง แต่ขดลวด ของมอเตอร์ ไม่เป็นอันตราย มอเตอร์กระแสตรงแบบอนุกรม ใช้งานหนักได้ดีเมื่อใช้งานหนัก กระแสจะมากความเร็วรอบ จะลดลงเมื่อไม่มีโหลดมาต่อความเร็วจะสูงมากอาจเกิดอันตรายได้ดังนั้นเมื่อเริ่มสตาร์ทมอเตอร์แบบอนุกรมจึงต้องมีโหลดมาต่ออยู่เสมอ
2. มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน หรือเรียกว่าชันท่มอเตอร์ (Shunt Motor) มอเตอร์แบบขนานนี้ขดลวดสนาม แม่เหล็กจะต่อ (Field Coil) จะต่อขนานกับขดลวด ชุดอาร์เมเจอร์ มอเตอร์แบบขนานนี้มีคุณลักษณะมีความเร็วคงที่ แรงบิดเริ่มหมุนต่ำแต่ความเร็วรอบคงที่ ชันท่มอเตอร์ส่วนมากเหมาะกับงานดังนี้พัดลมเพราะพัดลมต้องการความเร็วคงที่ และต้องการเปลี่ยนความเร็วได้ง่าย
3. มอเตอร์ไฟฟ้ากระแสตรงแบบผสม (Compound Motor) หรือเรียกว่าคอมเปาวด์มอเตอร์ มอเตอร์ไฟฟ้ากระแสตรงแบบผสมนี้ จะนำคุณลักษณะที่ดีของมอเตอร์ไฟฟ้ากระแสตรง แบบขนานและแบบอนุกรมมารวมกัน มอเตอร์แบบผสม มีคุณลักษณะพิเศษคือมีแรงบิดสูง (High starting torque) แต่ความเร็วรอบคงที่ ตั้งแต่ยังไม่มีการโหลดจนกระทั่งมีโหลดเต็มที่

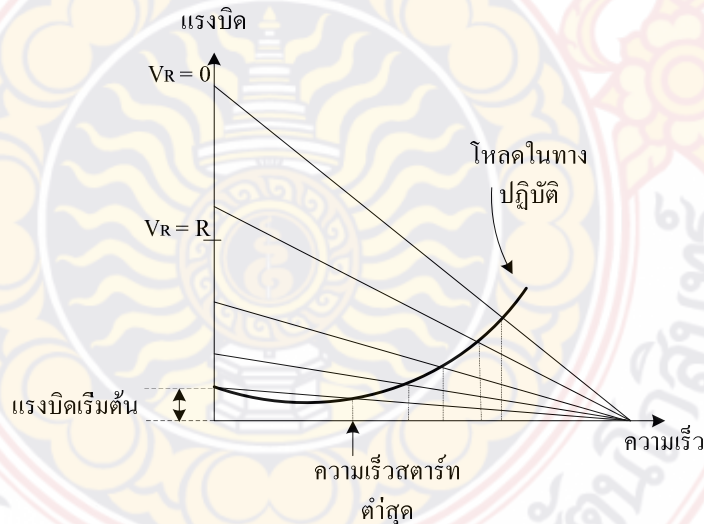
หลักการควบคุมความเร็วมอเตอร์

ในการควบคุมความเร็วของมอเตอร์นั้นเราสามารถเลือกใช้วิธีการในการควบคุมได้หลายแบบ ซึ่งขึ้นอยู่กับลักษณะของงานที่จะนำมอเตอร์ไปใช้ ว่าต้องการคุณสมบัติใด มากน้อยเพียงใด ซึ่งวงจรควบคุมความเร็วมอเตอร์ในแต่ละแบบก็จะมีข้อดี ข้อเสียต่างกันไป โดยในการควบคุมความเร็วมอเตอร์ขั้นพื้นฐานแล้วจะประกอบวิธีการต่าง ๆ ดังนี้

1. การควบคุมด้วยตัวต้านทานปรับค่าได้



ภาพที่ 16 ความสัมพันธ์ระหว่างแรงบิดกับความเร็วในทางอุดมคติ



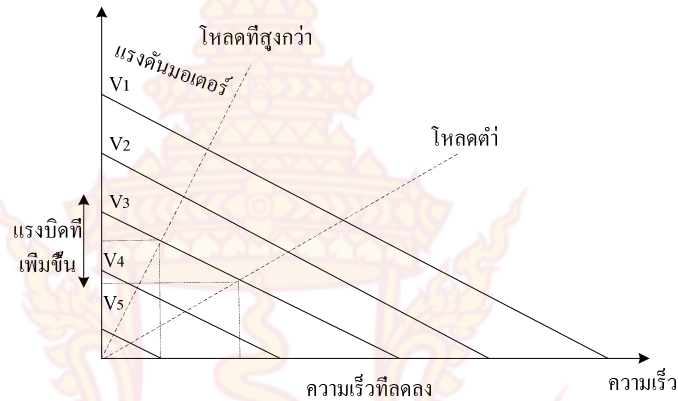
ภาพที่ 17 ความสัมพันธ์ระหว่างแรงบิดกับความเร็วในทางปฏิบัติ

เป็นรูปแบบพื้นฐานที่สุดของการควบคุมมอเตอร์คือ ใช้ตัวต้านทานปรับค่าได้อนุกรมกับมอเตอร์ โดยตัวต้านทานที่ปรับค่าได้จะเป็นตัวกำหนดความเร็วในการหมุนของมอเตอร์ การบังคับแบบนี้มีประสิทธิภาพน้อยเพราะกำลังสูญเสียไปในตัวความต้านทาน มักนิยมใช้กับมอเตอร์ตัวเล็ก ๆ การบังคับแบบนี้ให้

คุณสมบัติการสตาร์ทดี (ให้แรงบิดสูงที่ความเร็วต่ำ) แต่จะให้ความเร็วสูงมากเมื่อมอเตอร์อยู่ในภาวะที่มีโหลดน้อย ๆ ดังนั้นการบังคับแบบนี้มีประโยชน์เฉพาะภาวะที่แรงต้านคงที่

2. การควบคุมด้วยวิธีเปลี่ยนค่าแรงดัน

วิธีการนี้ดีกว่าวิธีการแรกแต่จะซับซ้อนกว่าต้องใช้อุปกรณ์อิเล็กทรอนิกส์ที่มีอัตรา ขยายกำลังสูง และมอเตอร์จะถูกป้อนด้วยแรงดันที่เปลี่ยนแปลงค่าได้ จากแหล่งจ่ายที่มีอิมพีแดนซ์ต่ำ

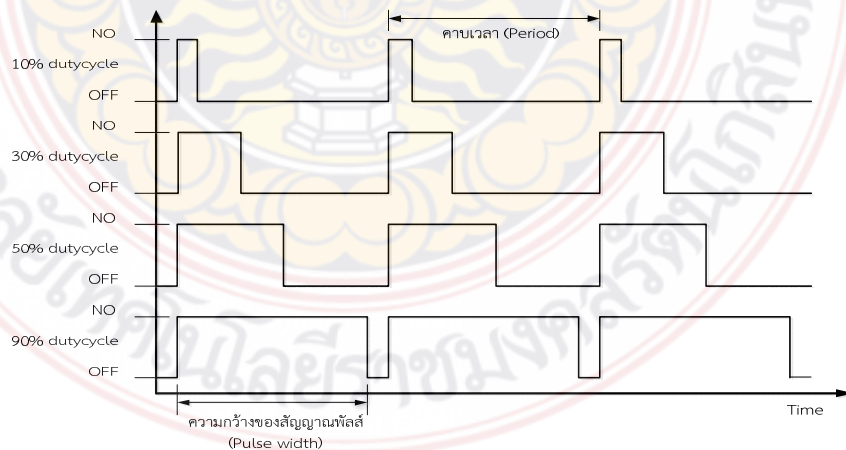


ภาพที่ 18 ความสัมพันธ์ระหว่างแรงดันกับความเร็วในการควบคุมด้วยแรงดัน

ข้อดีของการควบคุมวิธีนี้คือ ถ้าความเร็วลดลงจากผลของแรงบิด แรงดันที่ป้อนให้กับมอเตอร์ จะเพิ่มขึ้นเพื่อรักษาระดับความเร็ว ส่วนข้อเสียจากการควบคุมวิธีนี้คือ เมื่อมอเตอร์มีความเร็วต่ำ แรงดันที่ป้อนให้กับมอเตอร์จะมีค่าต่ำเช่นกัน

3. การควบคุมแบบ PWM (Pulse Width Modulation)

PWM คือ การควบคุมแบบ Digital ซึ่งจะไม่เหมือนกับ วิธีแบบ Analog ซึ่งวิธีแบบ Analog ก็เปรียบเหมือนวิธีข้างต้น คือ การให้พลังงานมากหรือน้อย ตามการเปลี่ยนแปลงของระดับ Voltage เมื่อป้อน Voltage ให้มาก พลังงานไฟฟ้าที่จ่ายให้แก่อุปกรณ์ไฟฟ้าก็จะมากถ้าอุปกรณ์ไฟฟ้าเป็น หลอดไฟ



ภาพที่ 19 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างกัน

หลอดไฟก็จะสว่างมากขึ้นถ้าเป็นมอเตอร์ก็จะหมุนเร็วขึ้น แต่ PWM จะจ่ายสัญญาณเป็นแบบ Digital คือเป็นสัญญาณที่มีความถี่คงที่ แต่มีจังหวะที่เป็น no และ off อยู่ในช่วงของความถี่นั้น ๆ โดยจะให้ค่าแรงดัน Output ที่คงที่ซึ่งการควบคุม PWM ก็คือการควบคุมความกว้างของช่วงจังหวะ on และ off นั้นเอง และในปัจจุบันก็มีการนำหลักการควบคุมแบบนี้ไปประยุกต์ใช้งานมากขึ้นเนื่องจากอุปกรณ์ต่าง ๆ และไมโครคอนโทรลเลอร์สามารถที่จะจ่ายสัญญาณแบบนี้ได้ และง่ายต่อการนำไปควบคุม

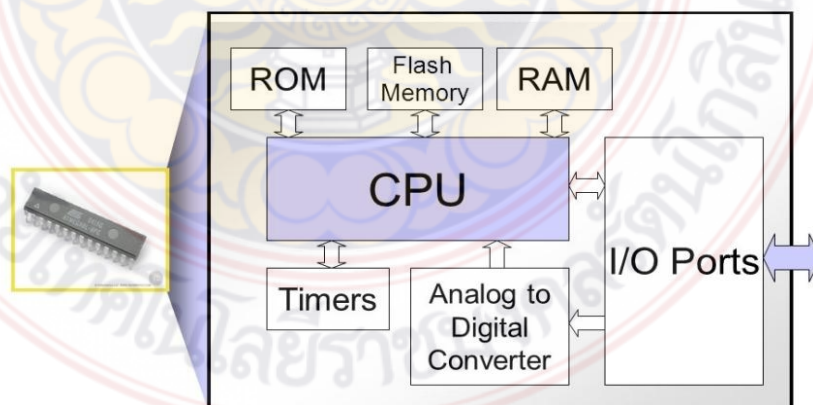
ในการควบคุมความเร็วมอเตอร์ด้วย PWM ก็คือการจ่ายสัญญาณที่มีแรงดันคงที่ ความถี่คงที่ แต่ความกว้างของ Pulse เปลี่ยนแปลงได้ ถ้า Pulse แคบความเร็วจะต่ำ แต่ถ้า Pulse กว้างความเร็วจะสูง โดย PWM ถูกเลือกใช้ในการควบคุมความเร็วของมอเตอร์เนื่องจากง่ายในการอินเทอร์เฟสกับไมโครคอนโทรลเลอร์ และใช้เพียงแค่เอาท์พุทสัญญาณเดียวในการควบคุมความเร็ว รวมไปถึง PWM มีประสิทธิภาพมากกว่า เพราะ Power Supply สามารถจ่ายกำลังได้เต็มที่ทั้ง On, Off ทำให้ได้ค่าทอร์ค และความเร็วสูงสุดของมอเตอร์

8. ไมโครคอนโทรลเลอร์

โครงสร้างของไมโครคอนโทรลเลอร์

โครงสร้างของไมโครคอนโทรลเลอร์ (Microcontroller) สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit)
2. หน่วยความจำซึ่งประกอบด้วย RAM และ EEPROM / EPROM / PROM / ROM (EEPROM Erasable Electrically Read-Only Memory)
3. หน่วยรับและแสดงผลข้อมูล (Input/Output) ซึ่งมีพอร์ตขยายแบบขนาน (Parallel) และแบบอนุกรม (Serial)
4. ตัวนับเวลา (Timer)
5. หน่วยควบคุมการอินเทอร์รัพท์ (Interrupt Controller)
6. ส่วนประกอบอย่างอื่นอีกเพื่อเพิ่มความสามารถขึ้นอยู่กับไมโครคอนโทรลเลอร์แต่ละเบอร์ เช่น A/D, PWM (Pulse Width Modulation) เป็นต้น



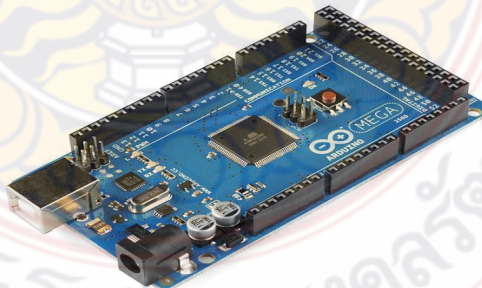
ภาพที่ 20 โครงสร้างของไมโครคอนโทรลเลอร์

บอร์ดไมโครคอนโทรลเลอร์ตระกูล Arduino Mega 2560

บอร์ดไมโครคอนโทรลเลอร์ตระกูลนี้ คือ การนำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่างๆ มาใช้ร่วมกันในภาษาซี ซึ่งภาษาซีเป็นลักษณะเฉพาะ คือมีการเขียนไลบาร์ของอาดุยโน ขึ้นมาเพื่อให้การสั่งงานไมโครคอนโทรลเลอร์ที่แตกต่างกันสามารถใช้งานโค้ดตัวเดียวกันได้โดยตัวปริณญาณิพนธ์ได้ออกบอร์ดทดลองมาหลายๆ รูปแบบเพื่อใช้งานกับ IDE (Integrated Development Environment) ของตนเองจากที่ได้กล่าวไปแล้วอาดุยโนนั้นได้ใช้ชิป AVR (Automatic Voltage Regulator) สาเหตุมาจากไมโครคอนโทรลเลอร์ของตระกูล AVR นั้นมีความทันสมัยในชิป ในบางตัวสามารถเชื่อมต่อผ่าน USB ได้โดยตรง สามารถใช้กับคอมพิวเตอร์ได้เป็นอย่างดีและในไมโครคอนโทรลเลอร์ตระกูล AVR ยังมีส่วนของโปรแกรมพิเศษที่เรียกว่า Bootloader อยู่ในระดับต่ำกว่าโปรแกรมปกติ ซึ่งจะเป็นส่วนโปรแกรมที่จะถูกเรียกขึ้นมาก่อนการเรียกโปรแกรมปกติ ทำให้สามารถเขียนสั่งให้ทำงานใดๆ ก็ได้ก่อนการเรียกโปรแกรมปกติทำให้อาดุยโนนั้นอาศัยส่วนโปรแกรมพิเศษนี้ในการทำให้ชิป สามารถโปรแกรมผ่านพอร์ตอนุกรมชนิด UART (Universal Asynchronous Receiver Transmitter) ได้จึงทำให้การเขียนโปรแกรมลงไปบนชิปใช้เพียง USB to UART ก็เพียงพอต่อการใช้งาน แต่การโปรแกรมด้วยการใช้โปรโตคอล UART ก็มีข้อเสียตรงที่ต้องใช้เวลาในการบูตเข้าโปรแกรมประมาณ 1-2 วินาที

1. ข้อมูลบอร์ดไมโครคอนโทรลเลอร์ตระกูล Arduino Mega 2560

ไมโครคอนโทรลเลอร์	ATmega2560
แรงดันไฟฟ้า	5 โวลต์
แรงดันไฟฟ้า(แนะนำ)	7-12 โวลต์
แรงดันไฟฟ้า(จำกัด)	6-20 โวลต์
ขาอินพุต เอาต์พุต ดิจิตอล	54
ขาแอนะล็อกอินพุต	16
SRAM	8 กิโลไบต์
EEPROM	4 กิโลไบต์
ความเร็วสัญญาณนาฬิกา	16 เมกะเฮิร์ตซ์



ภาพที่ 21 บอร์ดไมโครคอนโทรลเลอร์ตระกูล Arduino Mega 2560

การขัดจังหวะการทำงานโปรแกรม (Interrupt)

การขัดจังหวะการทำงานโปรแกรมคือการขัดจังหวะการทำงานของโปรแกรมปกติ เมื่อเกิดเหตุการณ์บางอย่างขึ้น ทำให้ซีพียูไปทำงานที่กำหนดไว้เมื่อเกิดอินเทอร์รัพท์ หากเปรียบเทียบการเกิดอินเทอร์รัพท์กับชีวิตประจำวันละก็ มันก็จะเหมือนกับตอนที่เรากำลังดูทีวีอยู่ แต่มีคนโทรเข้ามาในโทรศัพท์มือถือ ทำให้เราต้องหยุดดูทีวีแล้วไปรับโทรศัพท์ เมื่อคุยโทรศัพท์เสร็จแล้วจึงกลับมาดูทีวีตามเดิม

1. ชนิดของอินเทอร์รัพท์ แบ่งตามชนิดของการเกิดได้ดังนี้

- อินเทอร์รัพท์จากภายนอก เช่น การเปลี่ยนสถานะลอจิกของพอร์ตใดพอร์ตหนึ่ง
- อินเทอร์รัพท์จากภายใน เช่น อินเทอร์รัพท์ที่เกิดจากไทม์เมอร์

2. การควบคุมอินเทอร์รัพท์

คือการควบคุมว่าจะให้ซีพียูตอบสนองต่ออินเทอร์รัพท์หรือไม่ แบ่งได้ดังนี้

- Disable Interrupt คือการควบคุมให้ซีพียูไม่ตอบสนองกับอินเทอร์รัพท์ เมื่อเกิดการอินเทอร์รัพท์ขึ้นซีพียูจะปล่อยผ่านอินเทอร์รัพท์นั้น
- Enable Interrupt คือการควบคุมให้ซีพียูตอบสนองต่ออินเทอร์รัพท์ไปตามปกติ การควบคุมอินเทอร์รัพท์จะใช้ในกรณีที่ต้องการให้ซีพียูกระทำคำสั่งที่ไม่สามารถหยุดการทำงานได้ เช่น การนับเวลา หากมีการอินเทอร์รัพท์เกิดขึ้นจะทำให้การนับเวลาคลาดเคลื่อนได้

3. การใช้งานอินเทอร์รัพท์ในโปรแกรมอาดูยโน้

การใช้งานอินเทอร์รัพท์จากภายนอกใช้เพียงแค่การสร้างฟังก์ชันรอร์บ แล้วจึงใช้คำสั่งที่กำหนดว่าจะให้เกิดอินเทอร์รัพท์เมื่อไร แต่หากเป็นการอินเทอร์รัพท์จากภายในจะค่อนข้างยุ่งยากมากๆ ดังนั้นในบทความนี้จึงจะกล่าวถึงการใช้อินเทอร์รัพท์จากภายนอกเท่านั้น

4. พอร์ตที่สามารถใช้งานอินเทอร์รัพท์

ในบอร์ดอาดูยโน้จะมีพอร์ตที่สามารถใช้อินเทอร์รัพท์ได้แบบจำกัด โดยพอร์ตที่สามารถใช้อินเทอร์รัพท์ได้ในบอร์ดอาดูยโน้แต่ละรุ่นก็จะแตกต่างกัน ซึ่งสามารถดูได้ตามตารางด้านล่างนี้

ตารางที่ 4 พอร์ตที่สามารถใช้งานอินเทอร์รัพท์

Board	int. 0	Int. 1	Int. 2	Int. 3	Int. 4	Int. 5
Uno, Ethernet	2	3				
Mega 2560	2	3	21	20	19	18
Leonardo	3	2	0	1	7	
Due	(see below)					

5. คำสั่งกำหนดใช้อินเทอร์รัพท์

คำสั่งกำหนด และสร้างอินเทอร์รัพท์ โดยมีรูปแบบการใช้งานดังนี้

-interrupt คือ ลำดับขาอินเทอร์รัพท์ตามที่ได้ดูไปในตารางที่2-2 ใน Arduino Due สามารถใส่หมายเลขขาลงไปได้เลย

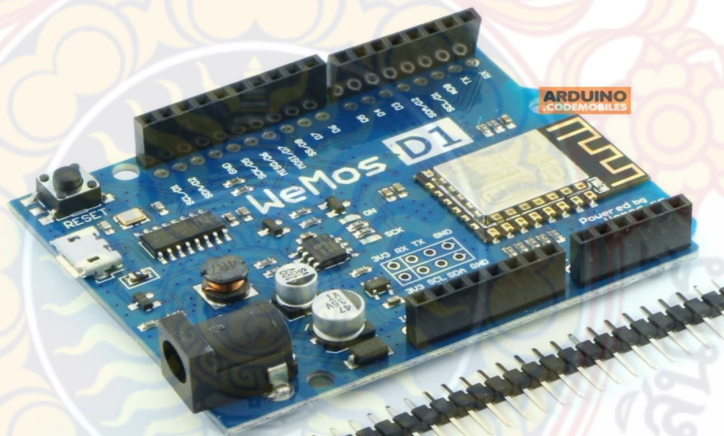
-ISR คือ ชื่อฟังก์ชันที่จะไปถูกเรียกขึ้นมาเมื่อเกิดอินเทอร์รัพท์

รูปแบบที่จะให้เกิดอินเทอร์รัพท์ มีทั้งหมด 4 รูปแบบดังนี้

- LOW จะเกิดอินเทอร์รัพท์ต่อเมื่อพอร์ตที่กำหนดไว้มีสถานะเป็น LOW
- CHANGE จะเกิดอินเทอร์รัพท์เมื่อพอร์ตที่กำหนดไว้มีการเปลี่ยนสถานะ เช่น จากสถานะ HIGH เป็น LOW หรือจาก LOW เป็น HIGH
- RISING จะเกิดอินเทอร์รัพท์เมื่อพอร์ตที่กำหนดไว้มีการเปลี่ยนสถานะจาก LOW เป็น HIGH
- FALLING จะเกิดอินเทอร์รัพท์เมื่อพอร์ตที่กำหนดไว้มีการเปลี่ยนสถานะจาก HIGH เป็น LOW
- HIGH จะเกิดอินเทอร์รัพท์ต่อเมื่อพอร์ตที่กำหนดไว้มีสถานะเป็น HIGH

9. บอร์ดติดต่อสื่อสารอินเทอร์เน็ต

บอร์ดติดต่อสื่อสารอินเทอร์เน็ต หรือ บอร์ด ESP8266 คือบอร์ดติดต่อสื่อสารกับ WI-FI แบบ Serial สามารถเขียนโปรแกรมลงในชิพ โดยใช้ Arduino IDE ได้ ทำให้การเขียนโปรแกรมและใช้งานเป็นเรื่องง่าย คล้ายกับการใช้ Arduino แน่นนอนว่าสามารถติดต่ออุปกรณ์อื่น ๆ เช่น เซอร์ ต่าง ๆ แบบสไตล์ Arduino ซึ่ง ESP8266 เป็นชิพของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.0-3.6V ทำงานใช้กระแสโดยเฉลี่ย 80mA รองรับคำสั่ง deep sleep ในการประหยัดพลังงาน ใช้กระแสน้อยกว่า 10 ไมโครแอมป์ สามารถ wake up กลับมาส่งข้อมูลใช้เวลา น้อยกว่า 2 มิลลิวินาที ภายในมี Low power MCU 32bit ทำให้เขียนโปรแกรมสั่งงานได้ มีวงจร analog digital converter ทำให้สามารถอ่านค่าจาก analog ได้ความละเอียด 10 bit ทำงานได้ที่ อุณหภูมิ -40 ถึง 125 องศาเซลเซียส

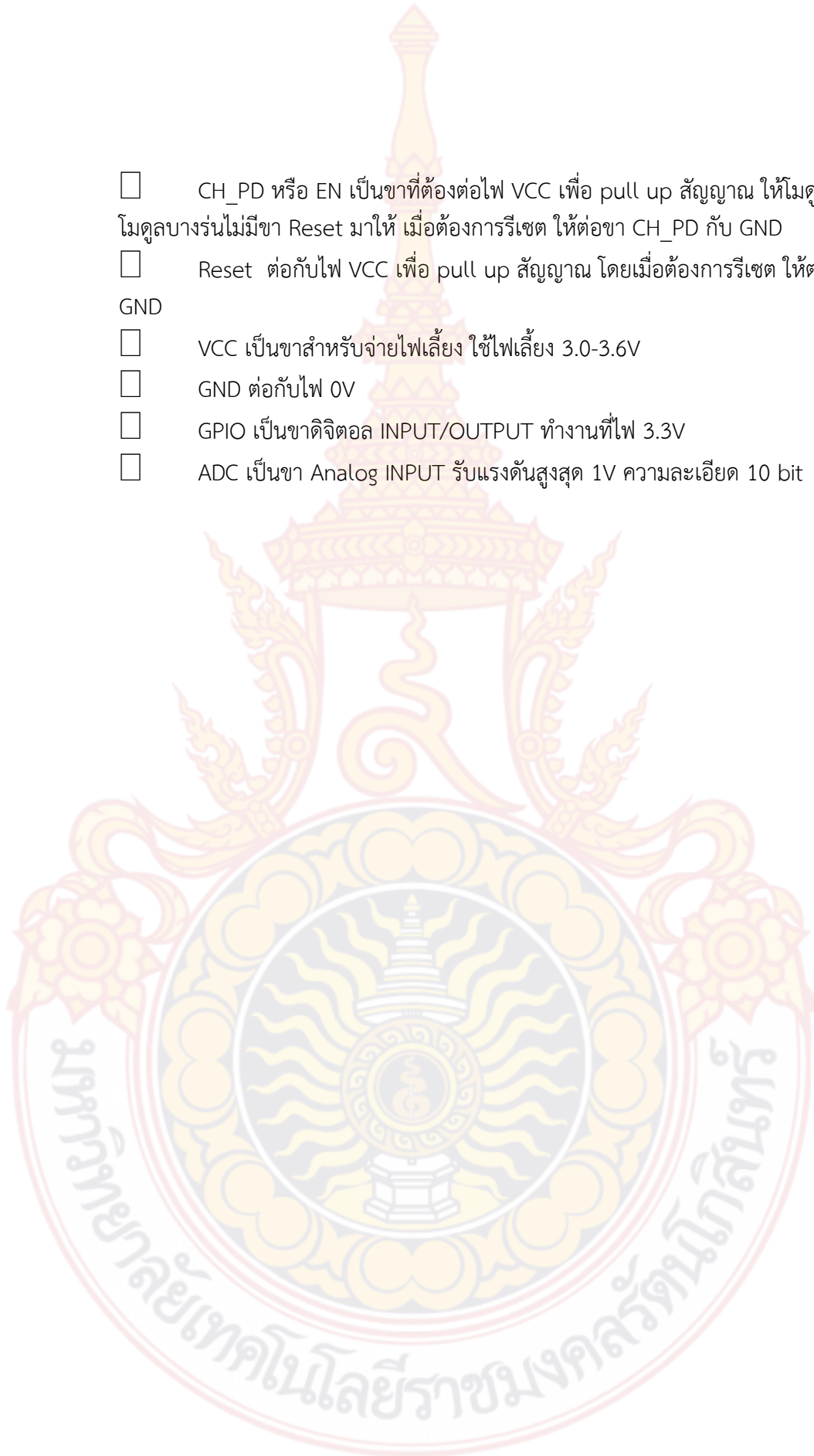


ภาพที่ 22 บอร์ดติดต่อสื่อสารอินเทอร์เน็ต

โมดูล ESP8266 โครงสร้างและขาที่ใช้งานก็จะมีลักษณะคล้ายกันคือ

- GPIO0 เป็นขาสำหรับเลือกโหมด โดยเมื่อต่อกับ GND จะเข้าโหมดโปรแกรม เมื่อต้องการให้ทำงานปกติก็ไม่ต้องต่อ
- GPIO15 เป็นขาที่ต้องต่อลง GND เพื่อให้โมดูลทำงาน

- CH_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ pull up สัญญาณ ให้โมดูลทำงาน โมดูลบางรุ่นไม่มีขา Reset มาให้ เมื่อต้องการรีเซ็ต ให้ต่อขา CH_PD กับ GND
- Reset ต่อกับไฟ VCC เพื่อ pull up สัญญาณ โดยเมื่อต้องการรีเซ็ต ให้ต่อกับไฟ GND
- VCC เป็นขาสำหรับจ่ายไฟเลี้ยง ใช้ไฟเลี้ยง 3.0-3.6V
- GND ต่อกับไฟ 0V
- GPIO เป็นขาคิจิตอล INPUT/OUTPUT ทำงานที่ไฟ 3.3V
- ADC เป็นขา Analog INPUT รับแรงดันสูงสุด 1V ความละเอียด 10 bit



บทที่ 3

วิธีดำเนินการวิจัย

3.1 ขั้นตอนการวิจัย

ในการดำเนินการวิจัยได้ทำการศึกษาข้อมูลเบื้องต้นของอุปกรณ์ต่าง ๆ ที่จะนำมาใช้ในการออกแบบโครงงานนี้ โดยได้ทำการออกแบบเครื่องมือและอุปกรณ์ที่ใช้ดังนี้

เครื่องมือและอุปกรณ์ที่ใช้

1. มอเตอร์กระแสตรง 10 ตัว
2. โฟโตนชิโอมิเตอร์ 10 ตัว
3. แบบจำลองเม็ดฝน 10 ตัว
4. ลูกรอก 10 ตัว
5. บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 1 บอร์ด
6. บอร์ดติดต่อสื่อสารอินเทอร์เน็ต
7. แหล่งจ่ายไฟ

3.2 ขั้นตอนการดำเนินงานวิจัย

1. ศึกษาและค้นคว้าข้อมูลการติดต่อเครือข่ายบนไมโครคอนโทรลเลอร์
2. ออกแบบและสร้างวงจรเชื่อมต่ออุปกรณ์ภายนอกบนไมโครคอนโทรลเลอร์
3. ออกแบบและเขียนโปรแกรมตัวควบคุมตำแหน่ง
4. ทดสอบส่วนต่างๆ ของโปรแกรมทั้งหมด

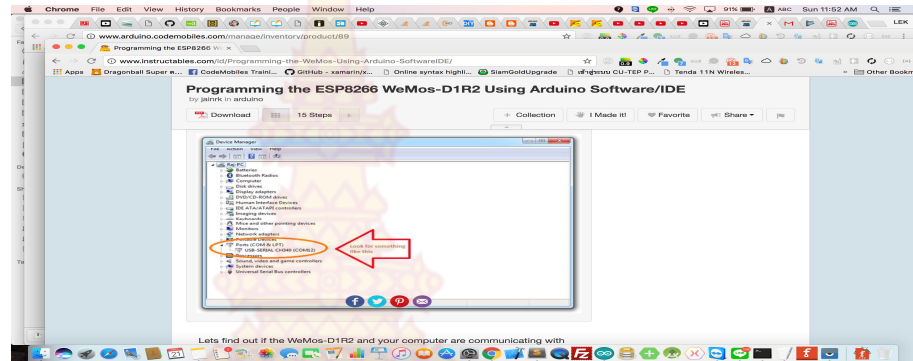
1. ศึกษาและค้นคว้าข้อมูลการติดต่อเครือข่ายบนไมโครคอนโทรลเลอร์

การใช้งานบอร์ด We Mos D1 ESP8266 และการทำเป็น Webserver ผ่าน Wi-Fi บอร์ด WeMos-D1 R2 คือบอร์ด Arduino UNO R3 กับ ชิพ ESP8266 WIFI มารวมกัน หรือที่เรียกกันคือ Arudino UNO with Buitin WiFi การอัปโหลดโปรแกรมลงบนบอร์ด ใช้ Arudino IDE และติดตั้ง USB Driver

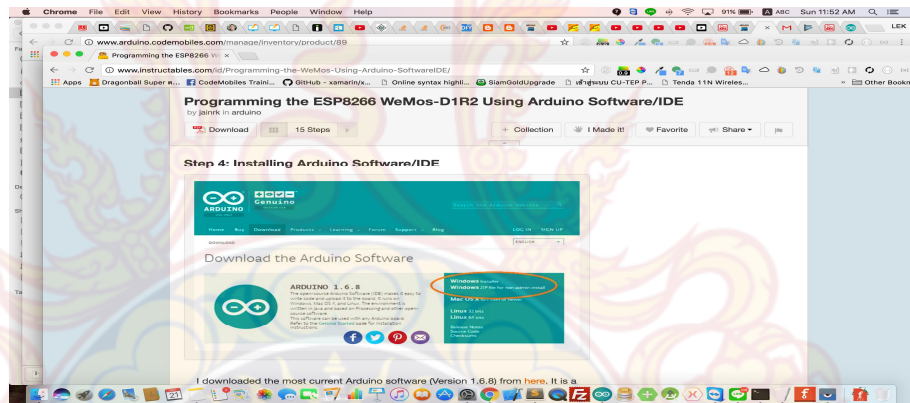
การติดตั้ง USB Driver ดังนี้

1. ติดตั้ง USB Driver
2. ติดตั้ง Arduino IDE
3. เพิ่ม Additional Board Manager URLs ที่รองรับ ESP8266
4. เพิ่ม board esp8266 ใน board manager
5. ทดสอบ Example LED Blink
6. ทดสอบ Example Wi-Fi Web server เชื่อมต่อ Wi-Fi

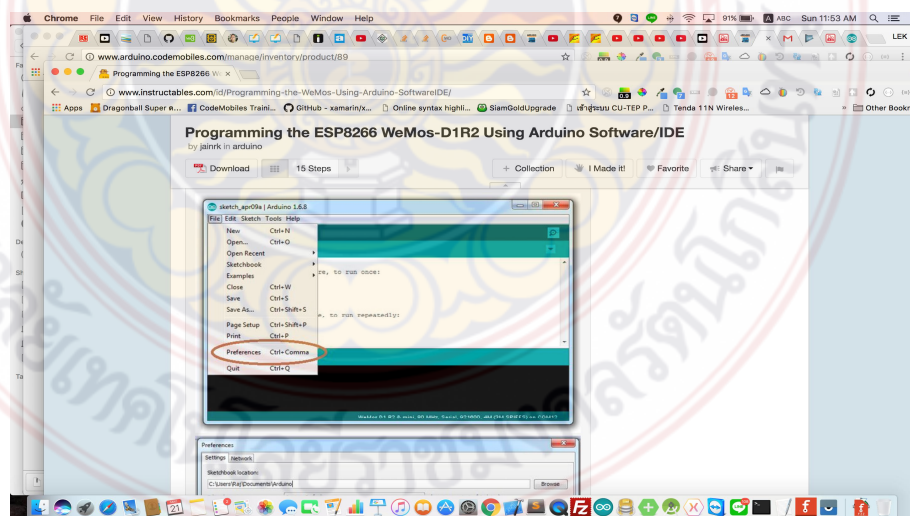
1. ติดตั้ง USB Driver

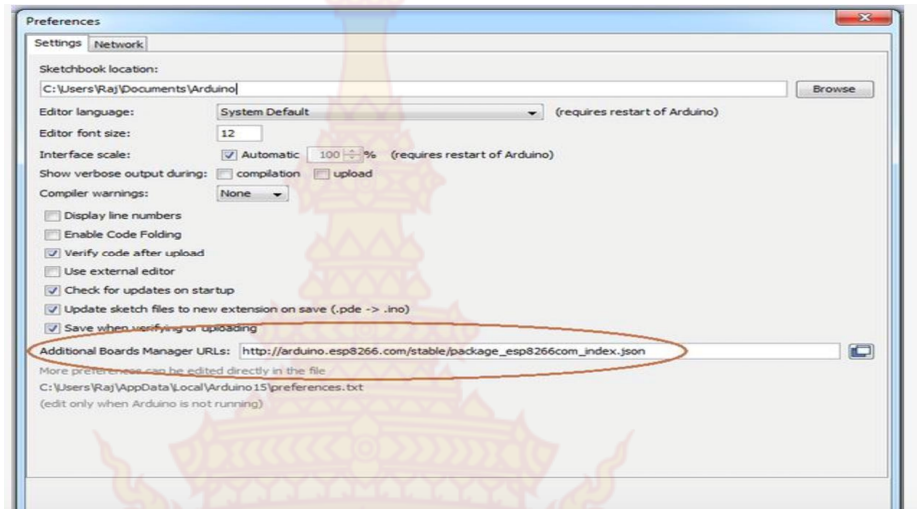


2. ติดตั้ง Arduino IDE

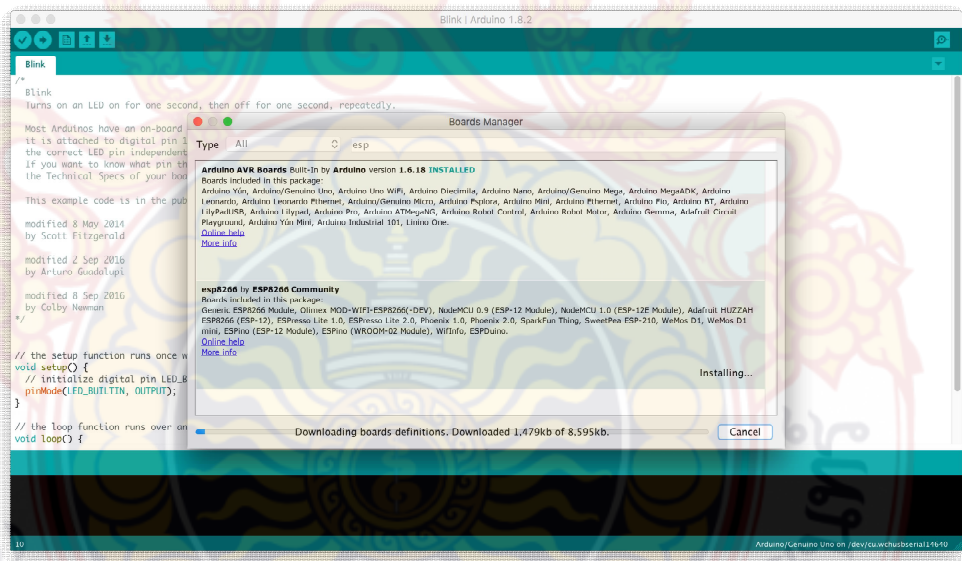


3. เพิ่ม Addition Board Manager URLs ที่รองรับ ESP8266

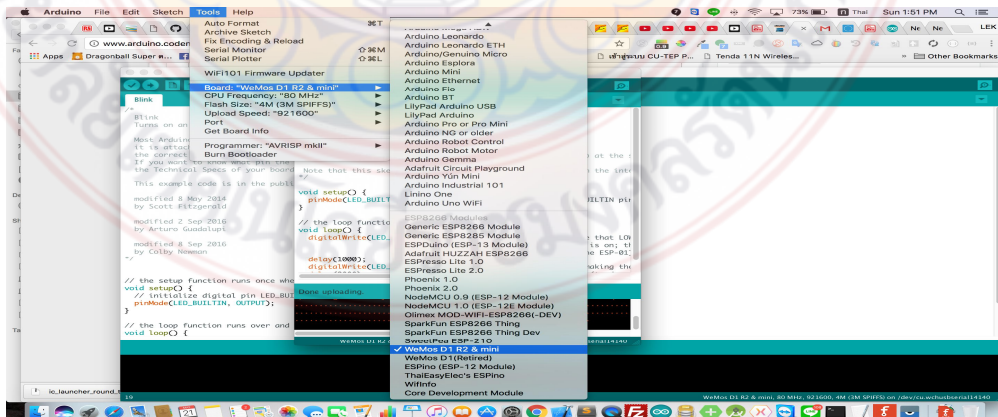




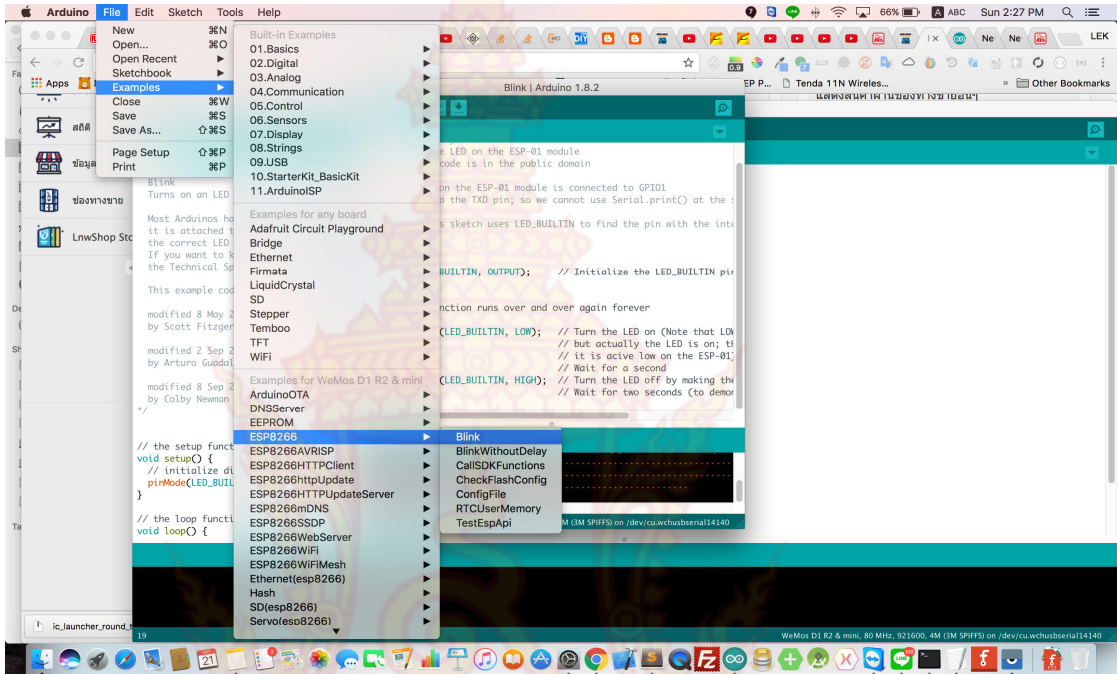
4. เพิ่ม Board ESP8266 ด้วย Board Manager โดยไปที่เมนู Tool->BoardManager และ พิมพ์ค้นหา "esp8266 of ESP8266 Community"



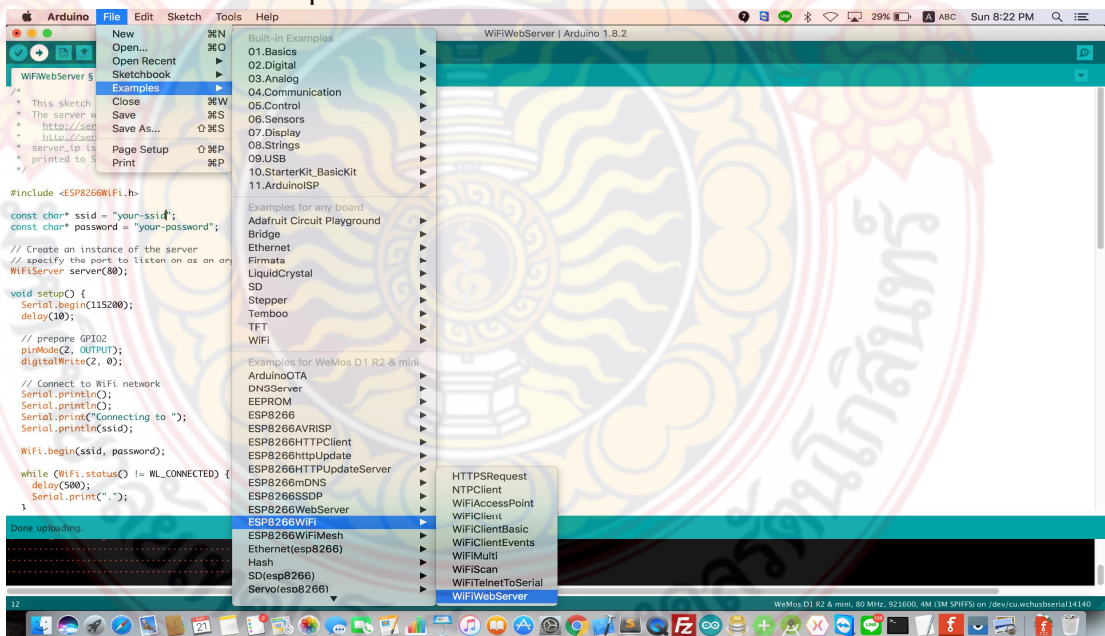
5. ทดสอบโปรแกรม LED Blink



เลือกโปรแกรมตัวอย่าง Blink เพื่อทำการทดสอบบอร์ด จากนั้นทำการอัปเดตโปรแกรม สังเกตเห็นบอร์ดมันมีไฟกระพริบเกิดขึ้น

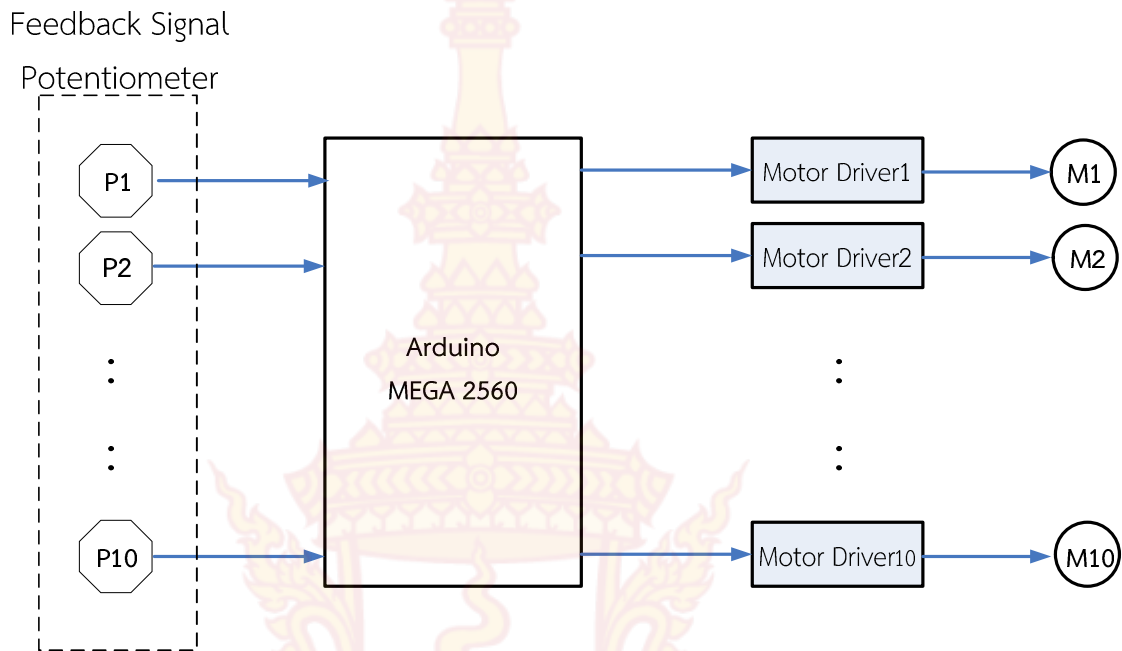


6 . ทดสอบ Example Wi-Fi Web server เพื่อต่อเชื่อมต่อ Wi-Fi



2. การออกแบบและสร้างวงจรเชื่อมต่ออุปกรณ์ภายนอกบนไมโครคอนโทรลเลอร์

ในการออกแบบสร้างวงจรเชื่อมต่ออุปกรณ์ภายนอกนั้น มี 2 ส่วน คือส่วนอ่านสัญญาณจากอุปกรณ์ตรวจวัดสัญญาณป้อนกลับ ส่วนที่สองคือส่วนส่งสัญญาณควบคุม (PWM) วงจรขับเคลื่อนมอเตอร์



ภาพที่ 23 บล็อกไดอะแกรมการต่อสายสัญญาณเข้ากับบอร์ดไมโครคอนโทรลเลอร์

3. ออกแบบและเขียนโปรแกรมตัวควบคุมตำแหน่ง

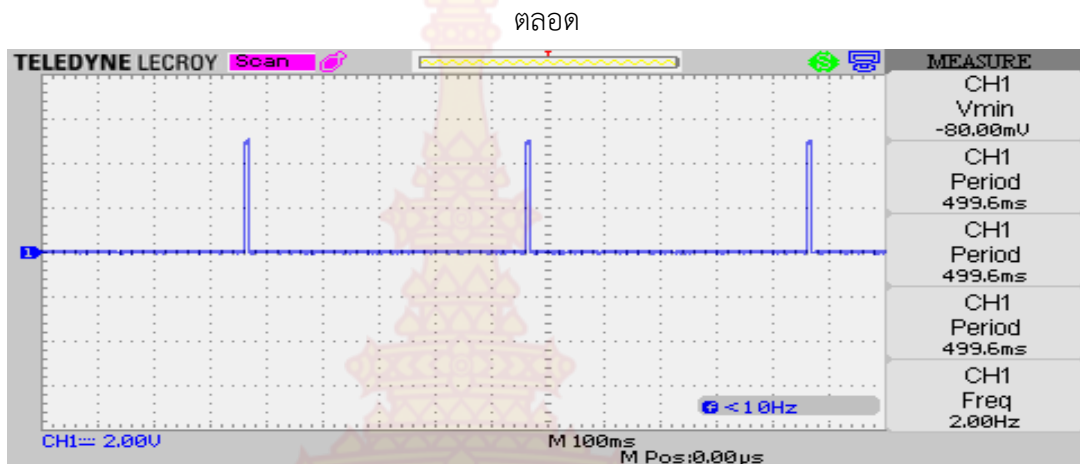
การออกแบบและเขียนโปรแกรมสามารถแบ่งออกเป็น 3 ส่วนคือส่วนติดต่อสื่อสารกับอินเทอร์เน็ต ส่วนโปรแกรมการขัดจังหวะ สำหรับเขียนตัวควบคุม PID และส่วนประมวลผล PID สำหรับตัวควบคุม PID โดยจะรับค่าจากอุปกรณ์วัดสัญญาณนำมาเปรียบเทียบกับค่าเป้าหมาย จะทำให้ได้ค่าผิดพลาด จากนั้นนำค่าผิดพลาดไปคำนวณเป็นสัญญาณควบคุมส่งให้อุปกรณ์ภายนอกต่อไป

1. ส่วนติดต่อสื่อสารกับอินเทอร์เน็ต

คือส่วนที่ทำหน้าที่ติดต่อระหว่างผู้ใช้กับตัวควบคุม เพื่อเปลี่ยนแปลงรูปแบบของ Kinetics Rain รายละเอียดอยู่ในภาคผนวก ข

2. ส่วนโปรแกรมการขัดจังหวะ

คือการขัดจังหวะการทำงานของโปรแกรมปกติ ในโครงงานวิจัยนี้ได้ออกแบบคาบเวลาการสุ่มในการขัดจังหวะ 500 มิลลิวินาที ทั้งนี้เพื่อให้โปรแกรมย่อยส่วนของโปรแกรม PID ประมวลผลสัญญาณส่งให้วงจรถับมอเตอร์ โดยการขัดจังหวะจะทำงานทุก ๆ คาบเวลาเท่ากับ 500 มิลลิวินาที



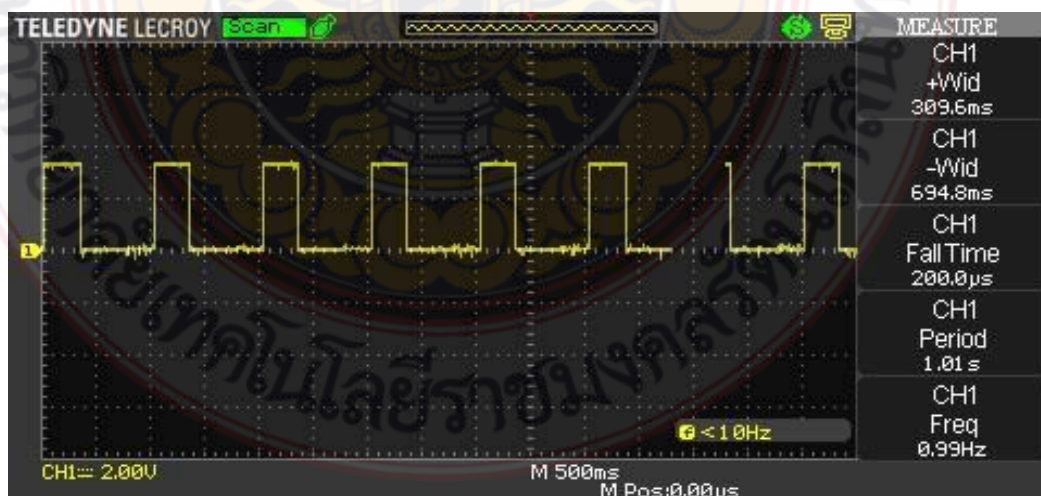
ภาพที่ 24 คาบเวลาการขั้ดจ้งหะทุก ๆ 500 มิลลิวินาที

3. ส่วนประมวลผล PID

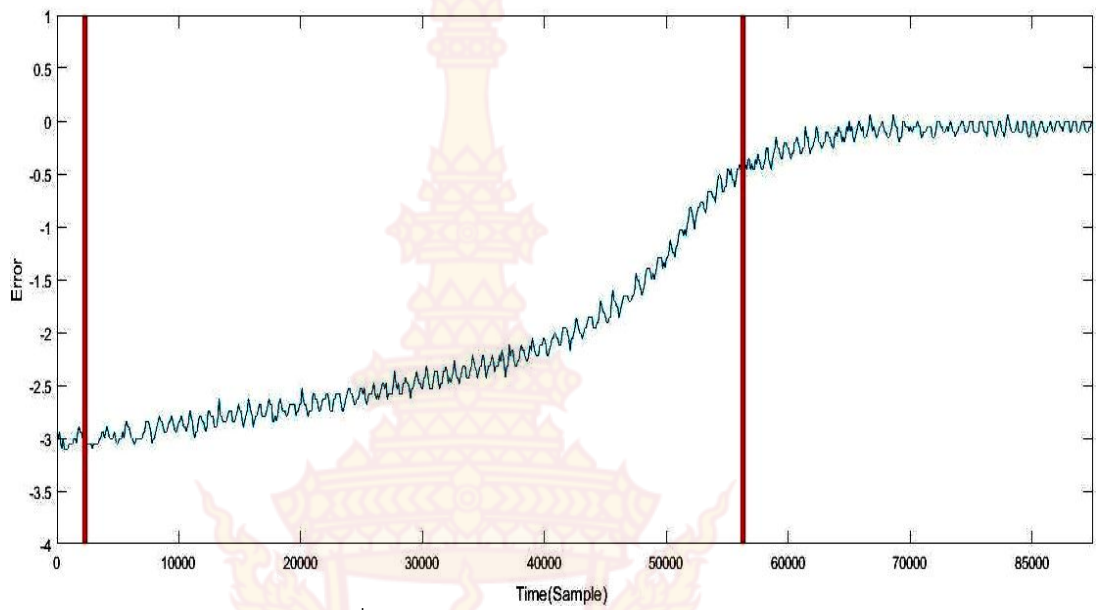
ส่วนประมวลผล PID ขึ้นอยู่กับพารามิเตอร์ 3 พารามิเตอร์คือ Kp Ki และKd บนพื้นฐานของการประมวลผลทั้งสามพารามิเตอร์จะคูณกับค่าผิดพลาด ซึ่งได้มาจากความแตกต่างของตัวแปรในกระบวนการกับค่าเป้าหมาย ตัวควบคุมที่มีพารามิเตอร์ทั้งสามจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุด รายละเอียดของโปรแกรมอยู่ในภาคผนวก ค

4. ทดสอบส่วนต่าง ๆ ของโปรแกรมทั้งหมด

การทดสอบโปรแกรมย่อยต่าง ๆ สำหรับโปรแกรมที่เขียนมานั้น จะทดสอบควบคู่กับเครื่องมือวัดสัญญาณแสดงผลด้วยภาพ ที่เรียกว่า ออสซิลโลสโคป ซึ่งการทดสอบจะแบ่งออกเป็น 2 ส่วนคือส่วนส่งสัญญาณ PWM ดังภาพที่ 25 และ ส่วนการอ่านค่าสัญญาณป้อนกลับ โดยในการอ่านค่าจากโพเทนชิโอมิเตอร์ ดังภาพที่ 26



ภาพที่ 25 แสดงสัญญาณ PWM



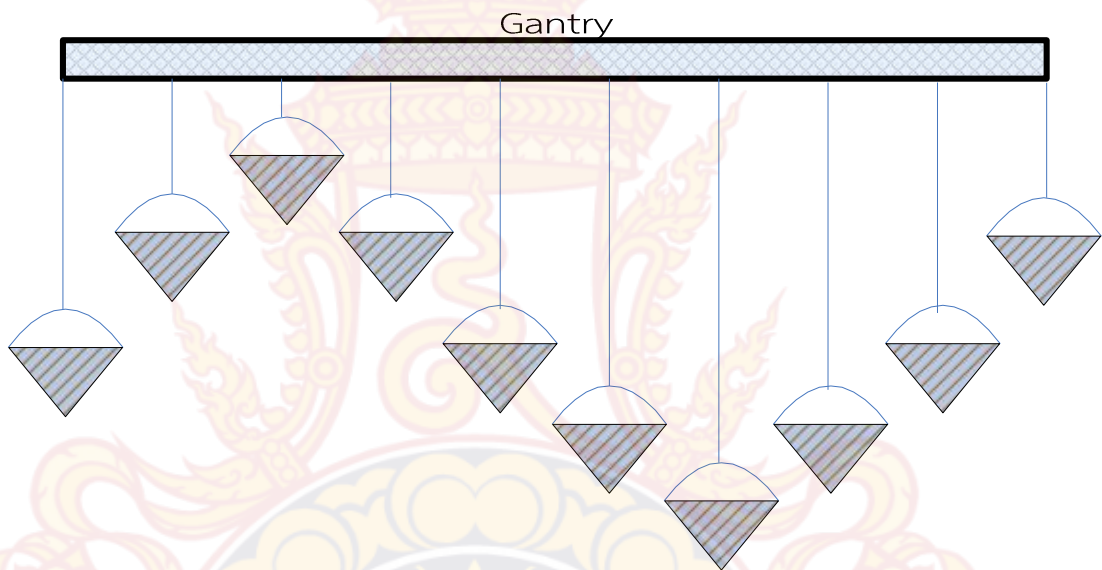
ภาพที่ 26 แสดงสัญญาณจากโพเทนชิโอเมเตอร์



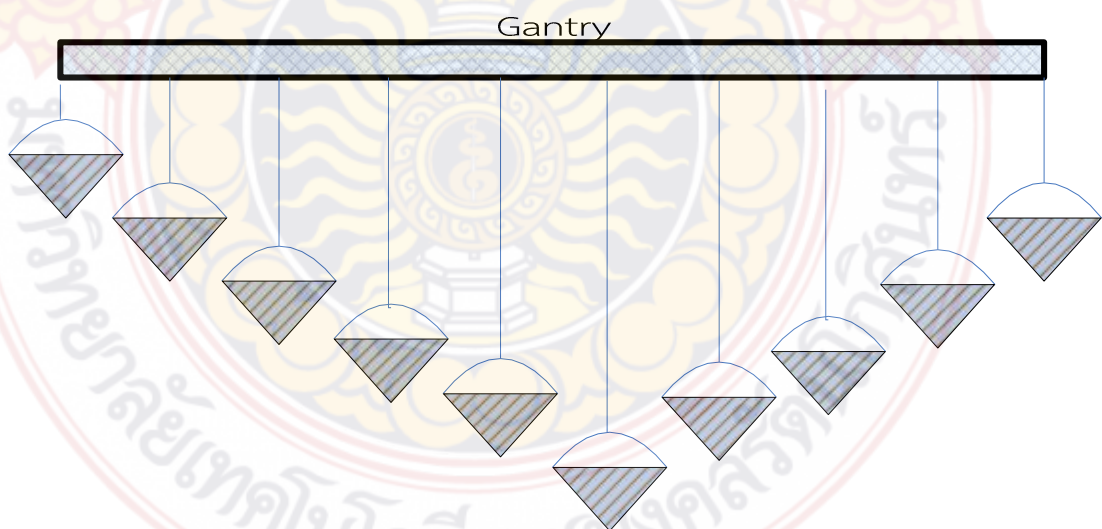
บทที่ 4

ผลการทดสอบและการวิเคราะห์ข้อมูล

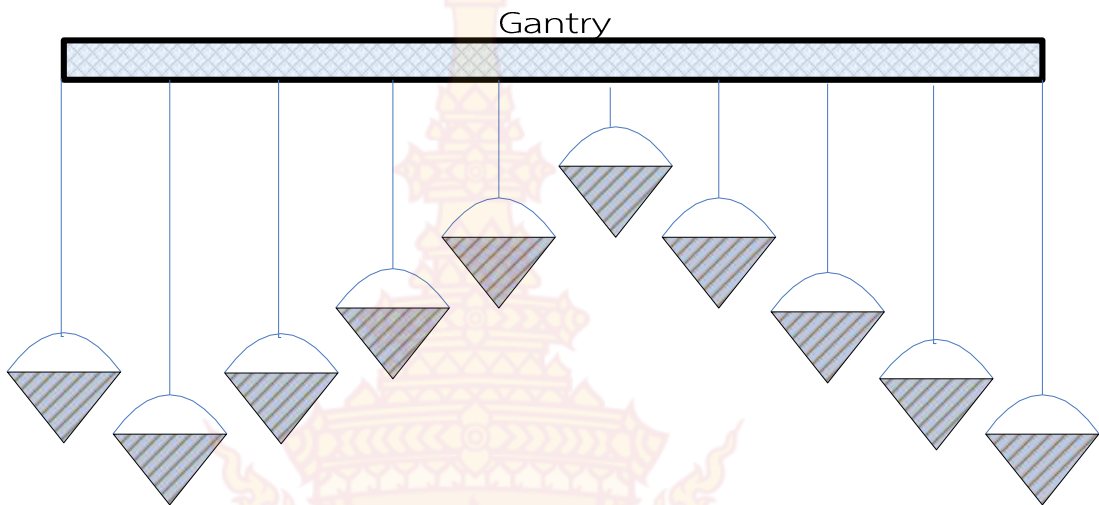
จากการศึกษาทฤษฎีที่เกี่ยวข้องกับตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต เพื่อให้สามารถนำไปใช้งานได้เป็นอย่างดี ตามวัตถุประสงค์และขอบเขตของโครงการ จึงได้ทำการทดลองจัดรูปแบบในการควบคุม 3 รูปแบบคือ 1. รูปคลื่น Sine wave 2. รูปเรียงระฆังคว่ำ และ 3. รูปเรียงระฆังหงาย โดยทำการทดลองควบคุมค่าตำแหน่งดังภาพที่ 27 28 และ 29 ตามลำดับ



. ภาพที่ 27 รูปคลื่น Sine wave



ภาพที่ 28 รูประฆังหงาย



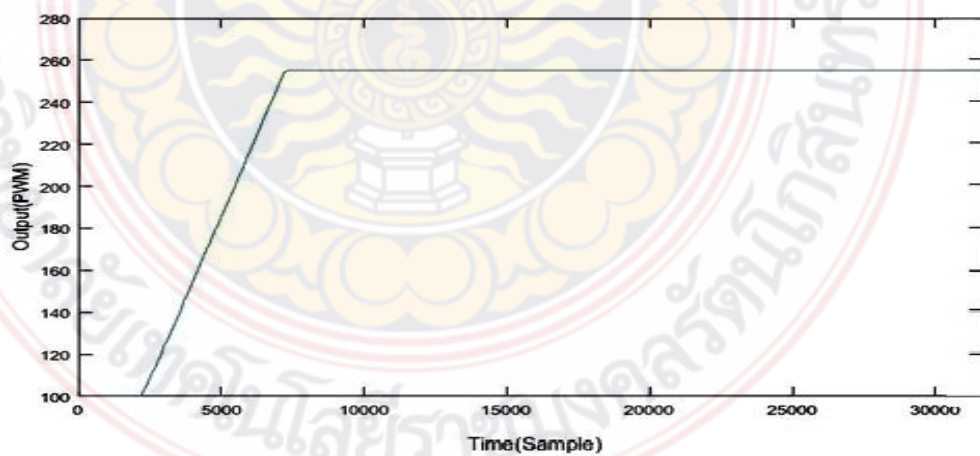
ภาพที่ 29 รูประฆังคว่ำ

4.1. วิธีการทดลอง

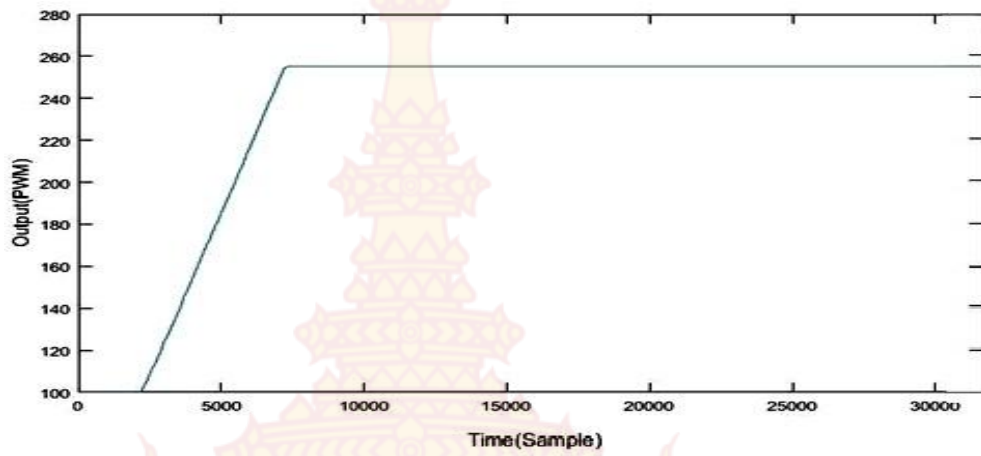
โดยการตั้งค่ารูปแบบไว้ทั้ง 3 รูปแบบ จากนั้นแปลงรูปแบบทั้ง 3 รูปแบบเป็นค่าตำแหน่งเส้นทางเดินของแต่ละเม็ดฝนจำลอง

4.2. บันทึกผลการทดลอง

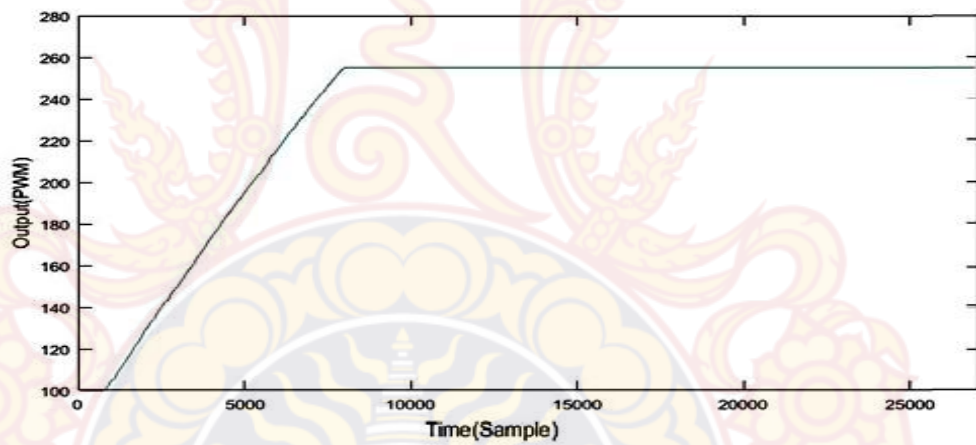
ผลการทดลองจะเปรียบเทียบเป็นกราฟแสดงการตอบสนองของรูปแบบการควบคุมทั้ง 3 แบบ คือ 1. รูปคลื่น Sine wave 2. รูปเรียงระฆังคว่ำ และ 3. รูปเรียงระฆังหงาย ดังภาพที่ 29 30 และ 31 ตามลำดับ



ภาพที่ 30 ผลการทดลองควบคุมตำแหน่งรูปคลื่น Sine



ภาพที่ 31 ผลการทดลองควบคุมตำแหน่งรูปประฆังคว่ำ



ภาพที่ 32 ผลการทดลองควบคุมตำแหน่งรูปประฆังหงาย

4.3 วิจัยณ์ผลการทดลอง

จากการทดลองการทำงานของตัวควบคุม Kinetics Rain โดยใช้ตัวควบคุม PID ทำงานร่วมกับโพเทนซีโอมิเตอร์ และการออกแบบเส้นทางเดินนั้น จากผลการทดลองพบว่าตัวควบคุม Kinetics Rain สามารถควบคุมค่าตำแหน่งได้

บทที่ 5

สรุปผลของการวิจัย

จากผลการวิจัยครั้งนี้เรื่องตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต สรุปได้ดังนี้

5.1 สรุปผลการวิจัย

จากการทดลองการทำงานของตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ต ในการทดลองนี้ได้ทำการทดลองเปลี่ยนรูปแบบของตำแหน่ง Kinetics Rain มี 3 รูปแบบคือ 1. รูปคลื่น Sine wave 2. รูปเรียงระฆังคว่ำ และ 3. รูปเรียงระฆังหงาย การควบคุมตำแหน่ง Kinetics Rain ตัวควบคุมอ่านค่าแรงดันจากโพเทนชิโอมิเตอร์ ซึ่งถูกกำหนดให้เป็นอุปกรณ์ส่งสัญญาณป้อนกลับ และส่งสัญญาณเข้าสู่ชุดแปลงสัญญาณอนาล็อกเป็นดิจิทัลของบอร์ดไมโครคอนโทรลเลอร์ เพื่อนำค่าดังกล่าวเปรียบเทียบกับค่าเป้าหมายที่เป็นค่าตำแหน่งมาจากฟังก์ชันเส้นทางเดิน จากนั้นเกิดค่าความผิดพลาด และตัวควบคุม PID จะประมวลผลสัญญาณควบคุม เพื่อส่งให้กับวงจรถับมอเตอร์ ผลการทดลองพบว่าตัวควบคุม PID บนบอร์ดไมโครคอนโทรลเลอร์ สามารถควบคุมค่าตำแหน่งของ Kinetics Rain ให้มีรูปแบบทั้ง 3 รูปแบบตามที่ต้องการได้

5.2 แนวทางในการพัฒนาโครงการวิจัย

การพัฒนาตัวควบคุมตำแหน่ง Kinetics rain ผ่านอินทราเน็ตสามารถนำไปพัฒนาต่อ มี 2 ส่วนด้วยกัน คือการจัดรูปแบบให้มีมากกว่า 3 รูปแบบ โดยการสร้างภาพบนกราฟฟิก จากนั้นแปลงภาพกราฟฟิกเป็นค่าตำแหน่ง และการพัฒนาสามารถนำโครงสร้างเชิงกลให้มีความสวยงาม



ภาคผนวก

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

ภาคผนวก ก
โปรแกรมการทำงานไมโครคอนโทรลเลอร์
สำหรับติดต่อสื่อสารอินเทอร์เน็ต



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = ".....";
const char* password = ".....";

ESP8266WebServer server(80);

const int led = 13;

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "Hello from esp8266!");
  digitalWrite(led, 0);
}

void handleNotFound(){
  digitalWrite(led, 1);
  String message = "File Not Found\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}
```



```
void setup(void){
  pinMode(led, OUTPUT);
  digitalWrite(led, 0);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.println("");

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("esp8266")) {
    Serial.println("MDNS responder started");
  }

  server.on("/", handleRoot);

  server.on("/inline", [](){
    server.send(200, "text/plain", "this works as well");
  });

  server.onNotFound(handleNotFound);

  server.begin();
  Serial.println("HTTP server started");
}
```

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





```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <Keypad.h>
#include <TimerOne.h>

int ph_pin = A0;
unsigned long time;
float Setpoint = 7.00; // pH Default
float process = 0; // process value
float Error = 0;
// PID Gain
float Kp = 5.0;
float Ki = 1.0;
float Kd = 1.0;
float dt = 0.1;
// PID Output Value
float P = 0;
float I = 0;
float D = 0;
float PIDSUM = 0.00;
float Int = 0.00;
float Diff = 0.00;
// Map PID Value to PWM
float math = 0.00;

int IN1 = 22;
int IN2 = 24;
int IN3 = 26;
int IN4 = 28;
int ENA = 6;
int ENB = 7;
int aaa = 4;
int point;
int key1;
int drive;
String str_pad = "";
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','.', 'D'}
};
byte rowPins[ROWS] = {36, 38, 40, 42};
byte colPins[COLS] = {44, 46, 48, 50};
Keypad keypad = Keypad(
    makeKeymap(keys), rowPins, colPins,
    ROWS, COLS );

```



```

Serial.begin(9600);
lcd.begin();
lcd.backlight();

Timer1.initialize(100000); //
Sampling time is 100 Millisec.

Timer1.attachInterrupt(pid);
}

void loop() {
int measure = analogRead(ph_pin);

float pHvalue =
map(measure,285,401,400,1000);

double voltage = 5 / 1024.0 *
measure;

process = (pHvalue/100);

KEYpad();
Serial.print(" pH = ");
Serial.println(process);
delay(100);
}

void pid()
{
Error = Setpoint - process;
P = Kp * Error;
I = Ki * (Error * dt + Int);
D = Kd * ((Error - Diff) / dt);

PIDSUM = P + I + D;
Int += Error;
Diff = Error;
PIDSUM = PIDSUM*PIDSUM;
PIDSUM = sqrt(PIDSUM);
math = constrain(PIDSUM,100,255);
}

void KEYpad(){
char key = keypad.getKey();
if (key != NO_KEY)
{
switch (key){
case 'A':
if (key == 'A') {
key1++;
}
break;
case 'B':
if (key == 'B') {
point++;
}
break;
case 'C':
str_pad = "";
lcd.setCursor(9, 1);
}
}
}

```

```

    lcd.print("    ");
    break;
case 'D':
    Setpoint = str_pad.toFloat();
    str_pad = "";
    lcd.setCursor(9, 1);
    lcd.print("    ");
    break;
case '*':
    drive++;
    break;
default:
    str_pad += key;
    lcd.print(key);
}
}

switch(key1) // Press 'A' Start motor
{
    case 0 : analogWrite(aaa,0);
    break;
    case 1 : analogWrite(aaa,255);
    break;
    case 2 : key1 = 0;
    break;
}

switch(point){ // Press 'B' Change to
page
    case 0 :
        lcd.setCursor(0,0);
        lcd.print("pH =    ");
        lcd.setCursor(5,0);
        lcd.print(process,2);
        delay(300);
        break;
    case 1 :
        lcd.setCursor(0, 0);
        lcd.print("Setup pH = ");
        lcd.print(Setpoint);
        lcd.setCursor(0,1);
        lcd.print("New pH = ");
        lcd.print(str_pad);
        lcd.setCursor(9, 1);
        break;
    case 2 :
        lcd.clear();
        point = 0;
        break;
}

switch(drive){
    case 0:

// Hybride Controller

```

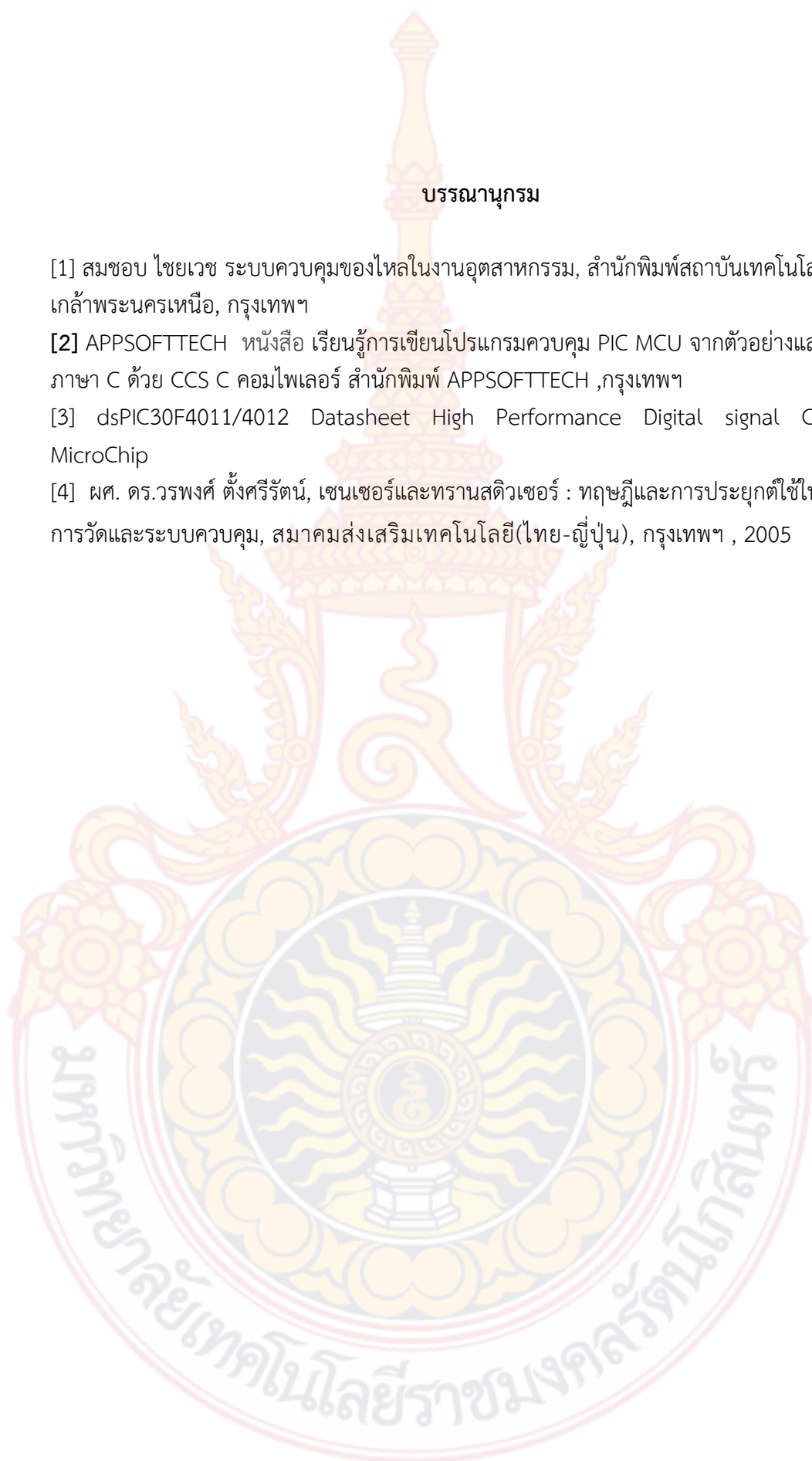
```

    if(Error < -0.1) // pump acid
    {
        analogWrite(ENA,math);
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,HIGH);
        digitalWrite(IN2,LOW);
    }
    else{
        analogWrite(ENA,0);
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,HIGH);
    }
}
if(Error > 0.1) // pump base
{
    analogWrite(ENB,math);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,HIGH);
}
else{
    analogWrite(ENB,0);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,HIGH);
}
break;
}
case 1:
    if(Error < -0.1) // pump acid

```

บรรณานุกรม

- [1] สมชอบ ไชยเวช ระบบควบคุมของไหลในงานอุตสาหกรรม, สำนักพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพฯ
- [2] APPSOFTTECH หนังสือ เรียนรู้การเขียนโปรแกรมควบคุม PIC MCU จากตัวอย่างและโค้ด ภาษา C ด้วย CCS C คอมไพเลอร์ สำนักพิมพ์ APPSOFTTECH ,กรุงเทพฯ
- [3] dsPIC30F4011/4012 Datasheet High Performance Digital signal Controllers MicroChip
- [4] ผศ. ดร.วรพงศ์ ตั้งศรีรัตน์, เซนเซอร์และทรานสดิวเซอร์ : ทฤษฎีและการประยุกต์ใช้ในระบบการวัดและระบบควบคุม, สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), กรุงเทพฯ , 2005



ประวัติผู้วิจัย



หัวหน้าโครงการ/ผู้วิจัยหลัก

1. ชื่อ - นามสกุล (ภาษาไทย) นายอดิศักดิ์ แข็งสาริกิจ
(ภาษาอังกฤษ) Mr. Adisak Khaengsarigid
2. ตำแหน่งปัจจุบัน : อาจารย์
3. หน่วยงานที่อยู่ที่สามารถติดต่อได้สะดวก พร้อมหมายเลขโทรศัพท์
โทรสาร และ E-mail
มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ วิทยาเขตศาลายา
เลขที่ 96 หมู่ 3 ถนนพุทธมณฑลสาย 5
ตำบลศาลายา อำเภอพุทธมณฑล จังหวัดนครปฐม 73170
โทรศัพท์ 0-2441-6000 ต่อ 2690 มือถือ 08-67897181
E-mail : adisak.kha@rmutr.ac.th
4. ประวัติการศึกษา
ปริญญาโท : วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมระบบควบคุม
จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปริญญาตรี : วิศวกรรมศาสตรบัณฑิต สาขาเทคโนโลยีโทรคมนาคม
จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
5. สาขาวิชาการที่มีความชำนาญพิเศษ
 - Control System
 - Model , Identification
 - Automata Machine
 - CNC Machine
 - Speed Motor Control
 - Industrial Sensor
 - Inverter Drive
 - Power Electronics circuit