

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมยานสำรวจใต้น้ำ

REMOTELY OPERATED VEHICLE (ROV) CONTROL



โดย

นาย พิชานนท์ ทิพย์ประเสริฐ
นาย ภาคย์ วาจาบัณฑิตย์

๘พ.
พ646 ๗
2549

เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

72186

12 ส.ย. 2550

b. 11x64533
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมยานสำรวจใต้น้ำ
REMOTELY OPERATED VEHICLE (ROV) CONTROL

ผู้จัดทำ นาย พิชานนท์ ทิพย์ประเสริฐ รหัสนักศึกษา 46010521
 นาย ภาคย์ วาทะบัณฑิต รหัสนักศึกษา 46010570

อาจารย์ที่ปรึกษา.....
(ผู้ช่วยศาสตราจารย์ ถาวร เบญจนาสุทธี)

การควบคุมยานสำรวจใต้น้ำ

Remotely Operated Vehicle (ROV) Control

โดย

นาย พิชานนท์ ทิพย์ประเสริฐ

นาย ภาคย์ วาจาบัณฑิตย์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ถาวร เบญจนาสุทธี

ปีการศึกษา 2549

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการออกแบบตัวควบคุมแบบพีไอดีและฟัซซี่ลอจิก เพื่อใช้ในการควบคุมทิศทางและระดับความลึกของยานสำรวจใต้น้ำได้อย่างอัตโนมัติ โดยจะเชื่อมต่อผ่านพอร์ตอนุกรมของคอมพิวเตอร์ด้วยโปรแกรมวิชวลเบสิกและโปรแกรมไมโครคอนโทรลเลอร์ คอมพิวเตอร์จะรับค่ามาจากเซนเซอร์วัดมุมและเซนเซอร์วัดความดัน แล้วส่งสัญญาณควบคุมให้มอเตอร์หมุนได้ทิศทาง 360 องศา และความลึกประมาณ 1.5 เมตรโดยผ่านแผงวงจรที่ได้ออกแบบไว้ จากผลการทดลองพบว่า การควบคุมแบบพีไอดีและแบบฟัซซี่ลอจิกสามารถควบคุมยานสำรวจใต้น้ำในทิศทางและระดับความลึกที่ต้องการได้ แต่การควบคุมแบบฟัซซี่ลอจิกจะให้ผลการทดลองที่ดีกว่า

Remotely Operated Vehicle (ROV) Control

Phichanon Thipprasoet

Pak Wachabandit

Advisor

Asst.Prof.Taworn Benjanarasuth

Academic Year 2006

ABSTRACT

This thesis presents the theory and the design concept of PID controller and fuzzy logic controller for controlling the angle and the depth of Remotely Operated Vehicle (ROV) automatically. The visual basic and micro-controller programs are employed to control the ROV via the serial port of a computer. The computer respectively receives the angle and the depth values from the compass module and the pressure transducer, and sends the control signals to the motors for 360° direction turning and for 1.5 meters diving via the designed circuits. The experimental results show that both of PID controller and fuzzy logic controller can control the angle and the depth of ROV as desired. However, the experimental results obtained from fuzzy logic controller are better than the results from PID controller.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือจาก ผศ.ถาวร เบญจนาสุทธี ที่ได้กรุณาและคอยให้คำปรึกษาที่ดีมาโดยตลอดตั้งแต่ต้น ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ รศ.ดร.จกกล งามวิวิทย์ ที่ให้คำปรึกษาในการทำวิทยานิพนธ์และเอื้อเฟื้ออุปกรณ์ต่างๆที่เป็นประโยชน์ต่อโครงการ

ขอขอบพระคุณ รศ.ดร.โยธิน เปรมปراميรัชต์ และ อาจารย์ธวัชชัย คำศรี ที่ให้คำปรึกษาในส่วนของการออกแบบโปรแกรมและวงจรอิเล็กทรอนิกส์

ขอขอบคุณรุ่นพี่ที่ได้ทำโครงการนี้ในปีที่แล้ว ได้แก่ พี่คนัย และ พี่ปริญา ที่ให้คำปรึกษาให้ส่วนต่างๆ ของโครงการมาโดยตลอด

ขอขอบคุณเพื่อนๆทุกคน ที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงการอยู่ตลอดเวลา

สุดท้ายขอขอบพระคุณคุณพ่อ คุณแม่ และครอบครัว ที่เป็นกำลังใจที่ดีเสมอตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จลงได้

ผู้จัดทำ

นายพิชานนท์ ทิพย์ประเสริฐ

นายภาคย์ วาจาบัณฑิตย์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ตัวควบคุม	3
2.1.1 ตัวควบคุมแบบดั้งเดิม	3
2.1.2 ตัวควบคุมแบบฟัซซีลอจิก	12
2.2 หลักการทำงานของยานสำรวจใต้น้ำ	20
2.3 อุปกรณ์วัดค่าต่างๆ	21
2.3.1 อุปกรณ์วัดความดัน	21
2.3.2 อุปกรณ์วัดค่ามุม	24
2.4 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกโดยผ่านพอร์ตอนุกรม	29
2.5 ไมโครคอนโทรลเลอร์	31
บทที่ 3 การสร้างและการออกแบบ	33
3.1 การออกแบบส่วนประกอบต่างๆของยานสำรวจใต้น้ำ	33
3.2 วงจรต่างๆ	34
3.3 การออกแบบระบบควบคุม	38
3.4 แผนภาพกระบวนการรับค่าจากเซนเซอร์	39
3.5 กระบวนการคำนวณและส่งสัญญาณขับเคลื่อนมอเตอร์	40
3.6 การรับค่าและคำนวณค่าพล็อตที่จ่ายออกไปด้วยโปรแกรม Visual Basic	41
3.7 การออกแบบตัวควบคุมแบบดั้งเดิม	42
3.8 การออกแบบตัวควบคุมฟัซซีลอจิก	43
บทที่ 4 ผลการทดลอง	44
4.1 ผลการควบคุมทิศทาง	46

4.1.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบดั้งเดิม	46
4.1.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีชชีลอจิก	49
4.2 ผลการควบคุมระดับความลึก	50
4.2.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบดั้งเดิม	50
4.2.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีชชีลอจิก	52
บทที่ 5 บทวิจารณ์และสรุป	
5.1 ผลสัมฤทธิ์โครงการ	54
5.2 ปัญหาที่พบและแนวทางแก้ไข	54
5.3 แนวทางในการค้นคว้าและพัฒนา	55
ภาคผนวก	56
ภาคผนวก ก. โปรแกรมที่ใช้ในโครงการ	57
ก.1 โปรแกรมที่ติดต่อกับผู้ควบคุม	57
ก.2 โปรแกรมในไมโครคอนโทรลเลอร์	75
ก.2.1 โปรแกรมสำหรับการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ ของ PIC1 และ PIC2	75
ก.2.2 โปรแกรมรับค่าและส่งข้อมูลจากอุปกรณ์วัดค่าทั้ง 2 ตัว ไปยังคอมพิวเตอร์ของ PIC3	77
ภาคผนวก ข. คู่มือประกอบการใช้งานอุปกรณ์ที่เกี่ยวข้องในโครงการ	81
ข.1 อุปกรณ์วัดค่ามุม	81
ข.2 อุปกรณ์วัดความดัน	83
ข.3 อุปกรณ์ต่างๆที่ประกอบในวงจร	84
ข.3.1 ไอซี PIC16F877A	84
ข.3.2 ไอซี CNY17-1	85
ข.3.3 ไอซี SN74LS125A	86
ข.3.4 ทรานซิสเตอร์ TIP122	87
ภาคผนวก ค. แผนผังอุปกรณ์และการเชื่อมต่อแผงวงจร	88
ค.1 วงจรแหล่งจ่ายไฟฟ้า	88
ค.2 วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์	88
ค.3 วงจรรวมชุดเซนเซอร์	89
ค.4 วงจรขยายสัญญาณ PWM	90
เอกสารอ้างอิง	91

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 การกำหนดช่วงจำกัดของเอาท์พุท	4
รูปที่ 2.2 ผลการตอบสนองของตัวควบคุมแบบพี	5
รูปที่ 2.3 บล็อกไดอะแกรมของระบบควบคุมแบบพี	5
รูปที่ 2.4 ผลการตอบสนองของตัวควบคุมแบบไอ	6
รูปที่ 2.5 บล็อกไดอะแกรมของระบบควบคุมแบบไอ	7
รูปที่ 2.6 ผลตอบสนองของระบบตัวควบคุมแบบดี	7
รูปที่ 2.7 บล็อกไดอะแกรมของระบบควบคุมแบบดี	8
รูปที่ 2.8 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ	9
รูปที่ 2.9 การตอบสนองของระบบควบคุมแบบพีไอ	9
รูปที่ 2.10 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี	10
รูปที่ 2.11 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี	11
รูปที่ 2.12 ฟิชชีเชต	12
รูปที่ 2.13 กราฟฟังก์ชันแสดงความเป็นสมาชิก	13
รูปที่ 2.14 กราฟฟังก์ชันแสดงความเป็นสมาชิกในแบบต่างๆ	14
รูปที่ 2.15 การฟิชชีฟิเคชันแบบซิงเกิลตัน	16
รูปที่ 2.16 การอินเฟอร์เรนซ์แบบฟิชชี	17
รูปที่ 2.17 การควบคุมยานสำรวจได้นำโดยใช้ฟิชชีลอจิก	18
รูปที่ 2.18 เทอมเซตของตัวแปรอินพุต	18
รูปที่ 2.19 การทำฟิชชีฟิเคชันของระบบควบคุม	19
รูปที่ 2.20 การดีฟิชชีฟิเคชันของระบบควบคุม	19
รูปที่ 2.21 อุปกรณ์วัดความดันที่ใช้ในโรงงาน	21
รูปที่ 2.22 วิธีการที่ใช้ในการทดสอบอุปกรณ์วัดความดัน	23
รูปที่ 2.23 ความสัมพันธ์ระหว่างความดันทานและแรงดันที่จ่ายให้อุปกรณ์วัดความดัน	24
รูปที่ 2.24 อุปกรณ์วัดค่ามุมที่ใช้ในโรงงาน	24
รูปที่ 2.25 ไดอะแกรมเวลาแสดงการทำงานในสถานะที่เป็นมาสเตอร์	25
รูปที่ 2.26 ไดอะแกรมเวลาแสดงการทำงานในสถานะที่เป็นสเลฟ	26
รูปที่ 2.27 การอ่านข้อมูลแบบบิตซีดีเทียบกับสัญญาณนาฬิกา	28

รูปที่ 2.28 การอ่านข้อมูลแบบไบนารีเทียบกับสัญญาณนาฬิกา	29
รูปที่ 2.29 การส่งข้อมูลแบบซิงโครนัส	29
รูปที่ 2.30 การส่งข้อมูลแบบอะซิงโครนัส	30
รูปที่ 2.31 IC MAX232	31
รูปที่ 3.1 ขานสำรวจ	33
รูปที่ 3.2 วงจรสวิตช์แบบ H-Bridge	34
รูปที่ 3.3 วงจรสวิตช์แบบ H-Bridge เมื่อ On สวิตช์ทั้ง 4 ตัว ทำให้มอเตอร์หมุนกลับทาง	34
รูปที่ 3.4 วงจรจับมอเตอร์	35
รูปที่ 3.5 วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์	35
รูปที่ 3.6 วงจรรวมชุดเซนเซอร์	36
รูปที่ 3.7 วงจรจับสัญญาณ PWM	37
รูปที่ 3.8 วงจรแหล่งจ่ายไฟฟ้า	37
รูปที่ 3.9 แผนผังแสดงการออกแบบระบบควบคุมยานสำรวจใต้น้ำ	38
รูปที่ 3.10 แผนภาพแสดงขั้นตอนการทำงานในการรับค่าจากเซนเซอร์	39
รูปที่ 3.11 แผนภาพแสดงวิธีการส่งสัญญาณและการจ่ายพัลส์ให้แก่มอเตอร์	40
รูปที่ 3.12 แผนภาพแสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์	41
รูปที่ 3.13 แสดงการเลือกช่วงของฟังก์ชันความเป็นสมาชิกของ ERROR และ $\Delta ERROR$	43
รูปที่ 4.1 การควบคุมทิศทางด้วยตัวควบคุมแบบพี $K_p = 4.5$	44
รูปที่ 4.2 การควบคุมทิศทางด้วยตัวควบคุมแบบพี $K_p = 5$	45
รูปที่ 4.3 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพี	45
รูปที่ 4.4 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอ $K_p = 5$, $K_i = 0.05$	46
รูปที่ 4.4 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอ $K_p = 5$, $K_i = 0.1$	46
รูปที่ 4.6 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอ	47
รูปที่ 4.7 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.05$, $K_d = 0.1$	47
รูปที่ 4.8 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.05$, $K_d = 0.5$	48
รูปที่ 4.9 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอดี	48
รูปที่ 4.10 การควบคุมทิศทางด้วยตัวควบคุมแบบฟัซซีลอจิก	49
รูปที่ 4.11 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบฟัซซี	49
รูปที่ 4.12 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอ $K_p = 3$, $K_i = 0.1$	50
รูปที่ 4.13 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอ $K_p = 3$, $K_i = 0.5$	50
รูปที่ 4.14 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอ	51

รูปที่ 4.15 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.1$, $K_d = 0.01$	51
รูปที่ 4.16 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.1$, $K_d = 0.01$	53
รูปที่ 4.17 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอดี	52
รูปที่ 4.18 การควบคุมทิศทางด้วยตัวควบคุมแบบพีซีลอจิก	53
รูปที่ 4.19 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีซีลอจิก	53
รูปที่ ก.1 หน้าต่างแสดงผลที่เชื่อมต่อกับผู้ควบคุม	57
รูปที่ ก.2 หน้าต่างแสดงกราฟทิศทางของยานสำรวจ	58
รูปที่ ก.3 หน้าต่างแสดงกราฟความลึกของยานสำรวจ	58
รูปที่ ข.1 ตำแหน่งขาต่างๆของอุปกรณ์วัดค่ามุม	81
รูปที่ ข.2 Vector 2X FUNCTION DESCRIPTION	82
รูปที่ ข.3 อุปกรณ์วัดความดัน	83
รูปที่ ข.4 ไอซี PIC16F877A	84
รูปที่ ข.5 ไอซี CNY17-1	85
รูปที่ ข.6 ไอซี SN74LS125A	86
รูปที่ ข.7 ทรานซิสเตอร์ TIP42C	87
รูปที่ ค.1 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรแหล่งจ่ายไฟฟ้า	88
รูปที่ ค.2 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์	89
รูปที่ ค.3 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรรวมชุดเซนเซอร์	89
รูปที่ ค.4 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรขยายสัญญาณ PWM	90

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงเมตริกซ์ FAM ของระบบควบคุม	18
ตารางที่ 3.1 ค่าเกณฑ์ในการควบคุมทิศทาง	42
ตารางที่ 3.2 ค่าเกณฑ์ในการควบคุมระดับความลึก	42
ตารางที่ 3.3 แสดงเมตริกซ์ FAM ของระบบควบคุมยานสำรวจใต้น้ำ	43



บทที่ 1

บทนำ

สืบเนื่องมาจากข้อเท็จจริงที่ว่า มนุษย์ มีความอยากรู้อยากเห็นเป็นธรรมชาติ ได้ส่งผลให้มีการสำรวจและเสาะแสวงหานานาสรรพสิ่ง ซึ่งแหล่งน้ำนี้ถือได้ว่าเป็นหนึ่งในหลายๆ แรงบันดาลใจให้มนุษย์ สืบค้น

ยานใต้น้ำ หรือ Underwater Vehicles (UVs) นั้น ได้ถือกำเนิดขึ้นมาประมาณต้นปี ค.ศ.1950 และได้รับการพัฒนาสืบทอดตลอดมาจนถึงปัจจุบัน ความจำเป็นหรือประโยชน์ที่ได้รับจากการประยุกต์ใช้ยานใต้น้ำนั้น เริ่มต้นจากการที่มนุษย์ต้องเผชิญกับความเสี่ยงอันเนื่องมาจากการดำน้ำลงไปปฏิบัติการกิจต่างๆ ได้ท้องทะเล เช่น การสำรวจค้นหาแหล่งพลังงานธรรมชาติ การสำรวจและซ่อมบำรุงแนวท่อก๊าซธรรมชาติและสายเคเบิลใยแก้วนำแสง การสำรวจทางโบราณคดี การสำรวจทางธรณีวิทยา เป็นต้น ซึ่งกรณีศึกษาเหล่านี้จะต้องปฏิบัติการภายใต้สภาวะแวดล้อมที่มีความเสี่ยงสูง จึงจำเป็นที่จะต้องใช้หุ่นยนต์หรือยานพาหนะอื่นปฏิบัติการแทนมนุษย์ การแบ่งกลุ่มของยานใต้น้ำนั้น สามารถจำแนกโดยทั่วไปได้เป็น 4 ประเภทหลักๆ คือ

- 1) Remotely Operated หรือ Remotely Piloted ซึ่งหมายถึง ยานใต้น้ำที่มีสาย (Umbilical Link หรือ Tether) เชื่อมต่อด้วยยานใต้น้ำ กับ เรือบนผิวน้ำ หรือ สถานีควบคุมและสั่งการบนบก อีกนัยหนึ่งคือ เป็นการควบคุมระยะไกล (Remotely Controlled) ผ่านสายสัญญาณนั่นเอง
- 2) Unmanned หรือ Untethered หรือ Autonomous หมายถึง ยานใต้น้ำอัจฉริยะที่มีระบบสมองกล ควบคุม สั่งการ นำร่อง และอื่นๆ ตลอดเวลาการปฏิบัติการกิจได้ท้องทะเล โดยมนุษย์ มีหน้าที่โปรแกรมภารกิจเข้าไปในสมองกลของยานใต้น้ำ หลังจากนั้นก็สามารถปล่อย (Launch) ยานใต้น้ำลงน้ำจากเรือบนผิวน้ำ จากฝั่ง หรือ จากท่าเรือ ได้เลย
- 3) Semi-Autonomous หรือ Towards Autonomous หรือ กึ่งอัจฉริยะ ซึ่งเป็นการผสมผสานเทคนิควิธีระหว่าง 1) และ 2) เข้าด้วยกัน ซึ่งถือได้ว่าเป็นพื้นฐานสำคัญประการหนึ่ง ก่อนการสร้างยานใต้น้ำ ประเภท Autonomous ให้สำเร็จลุล่วงได้
- 4) Manned Submersible หรือ เรียกขานโดยย่อว่า ManSub คือ ยานใต้น้ำที่มีมนุษย์ประจำการในตัวยานฯ นั้นๆ เพื่อ ควบคุม สั่งการ นำร่อง และอื่นๆ ตลอดเวลา อีกนัยหนึ่ง คือ เรือดำน้ำ หรือ ยานฯ สำหรับการท่องเที่ยวได้ท้องทะเล ที่เรียกว่า Underwater Safari Submersible อนึ่ง อาจมีการผสมผสาน ระบบ ควบคุม สั่งการ และ นำร่อง อัตโนมัติ ที่เรียกว่า Auto-Pilot เข้ามาในตัวยานฯ ด้วยก็ได้

ประวัติความเป็นมาของโครงการ

ในปัจจุบันนี้ความต้องการการใช้ระบบการสำรวจใต้น้ำในประเทศไทยมีสูงมาก และประเทศไทยเองยังไม่สามารถผลิตยานสำรวจใต้น้ำที่มีประสิทธิภาพสูงเพื่อใช้งานเองได้ อีกทั้งการส่งยานออกไปสำรวจใต้น้ำในแต่ละครั้งนั้นมีค่าใช้จ่ายสูงมาก และจะต้องทำโดยผู้เชี่ยวชาญเท่านั้น จึงมีความจำเป็นที่จะต้องนำเข้าเทคโนโลยีด้านนี้จากต่างประเทศ ดังนั้น จึงเป็นการดีถ้าจะมียานสำรวจใต้น้ำที่ราคาไม่สูงและควบคุมได้ง่าย ด้วยเหตุนี้จึงมีการค้นคว้าและคิดทำโครงการนี้ขึ้นมา โดยใช้ชื่อว่า “ Remotely Operated Vehicle (ROV) Control “ หรือ “ การควบคุมยานสำรวจใต้น้ำ ” ซึ่งได้มีการพัฒนาไปแล้ว 4 รุ่น โดยรุ่นแรกได้ใช้การควบคุมปริมาตรภายในเรือเพื่อควบคุมการดำขึ้นหรือลง (การใช้ถังอัดอากาศ) ส่วนรุ่นที่ 2 ได้ใช้ใบพัดแทนการควบคุมปริมาตรโดยการติดใบพัดไว้ด้านบนของเรือแล้วให้หมุนทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาเพื่อควบคุมการดำขึ้นลง รุ่นที่ 3 ได้ใช้เซนเซอร์วัดมุมในการวัดค่ามุมให้ได้ตามที่ต้องการ และในรุ่นที่ 4 ได้ออกแบบส่วนของตัวยาน แสงวงจรและระบบควบคุมแบบพีไอดี

โครงการในรุ่นที่ 5 นี้จะเป็นการพัฒนาต่อจากโครงการในรุ่นที่ 4 โดยการออกแบบโครงสร้างของยานสำรวจใหม่ รวมทั้งส่วนป้องกันตัวยานโดยให้มีความแข็งแรงทนทาน และสามารถถอดประกอบได้ทุกชิ้นส่วนเพื่อการซ่อมบำรุงได้ง่ายขึ้น และมีการออกแบบตัวควบคุมแบบพีซีลอจิกเข้ามาช่วยในการควบคุม

บทที่ 2

ทฤษฎีและหลักการ

ในบทนี้จะกล่าวถึง ทฤษฎีการควบคุมและหลักการทำงานของยานสำรวจได้นำ รวมทั้ง หลักการของอุปกรณ์วัดความดัน (Pressure Transducer) อุปกรณ์วัดค่ามุม (Compass Module) โดย จะแสดงถึงวิธีการใช้งาน การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกโดยผ่านพอร์ตอนุกรม

2.1 ตัวควบคุม

ตัวควบคุมที่ใช้ในยานสำรวจได้นำมี 2 แบบ คือ ตัวควบคุมแบบพีไอดีและตัวควบคุมแบบ ฟัซซีลอจิก

2.1.1 ตัวควบคุมแบบดั้งเดิม

ตัวควบคุมแบบดั้งเดิม (Classical Controller) เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบ ป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุตและมีเอาต์พุตเป็นอินพุตของ Corrective Element ในหัวข้อนี้จะกล่าวถึงวิธีการเลือกใช้ตัวควบคุมที่เหมาะสมกับระบบควบคุมแบบป้อนกลับและ วิธีการหาพารามิเตอร์ที่เหมาะสมสำหรับตัวควบคุม โดยที่ เอาต์พุตของตัวควบคุม คือ กฎการ ควบคุม (Control Law) โดยจะมี 3 รูปแบบ คือ

1. การควบคุมแบบพี (P: Proportional Control)
2. การควบคุมแบบไอ (I: Integral Control)
3. การควบคุมแบบดี (D: Derivative Control)

ในระบบควบคุมป้อนกลับบางประเภท มีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบ ควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่ ปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เรียกว่า การชดเชย (Compensation)

2.1.1.1 ระบบควบคุมแบบพี

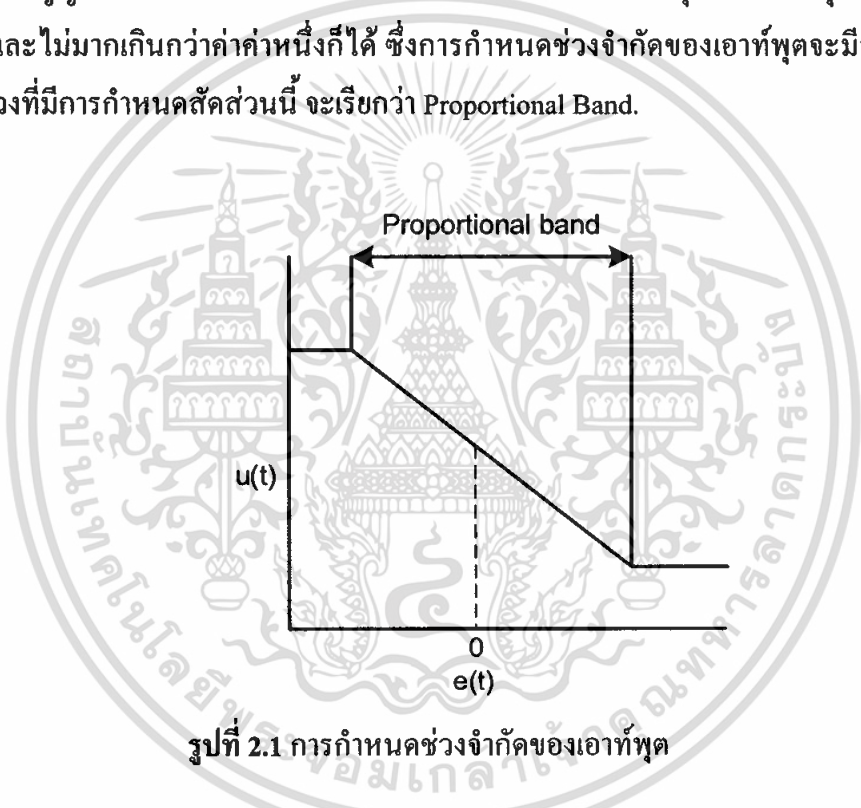
ในระบบควบคุมแบบพี(P: Proportional Control) เอาต์พุตของตัวควบคุม $u(t)$ จะเป็น สัดส่วนกับอินพุตของตัวควบคุม และถ้ากำหนดสัญญาณอินพุตให้กับตัวควบคุมเป็น ค่าความ ผิดพลาด $e(t)$ ซึ่งเป็นฟังก์ชันของเวลา จะได้

$$u(t) = K_p e(t) \quad (2.1)$$

เมื่อ K_p เป็นค่าคงที่เรียกว่า อัตราขยายแบบสัดส่วน (Proportional Gain) จะพบว่าเอาต์พุตที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วน โดยตรงกับขนาดของความผิดพลาดในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม $G_c(s)$ มีค่าเป็น

$$G_c(s) = K_p \tag{2.2}$$

ดังนั้น การควบคุมด้วยตัวควบคุมแบบนี้ก็จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จากตัวควบคุมในเวลานั้น อย่างไรก็ตาม การที่ให้อัตราขยายคงที่นั้น ในทางปฏิบัติอาจกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น หรืออาจกำหนดให้ตัวควบคุมมีค่าเอาต์พุตไม่น้อยกว่าค่าค่าหนึ่งและไม่มากเกินไปกว่าค่าค่าหนึ่งก็ได้ ซึ่งการกำหนดช่วงจำกัดของเอาต์พุตจะมีลักษณะดังรูปที่ 2.1 ช่วงที่มีการกำหนดสัดส่วนนี้ จะเรียกว่า Proportional Band.

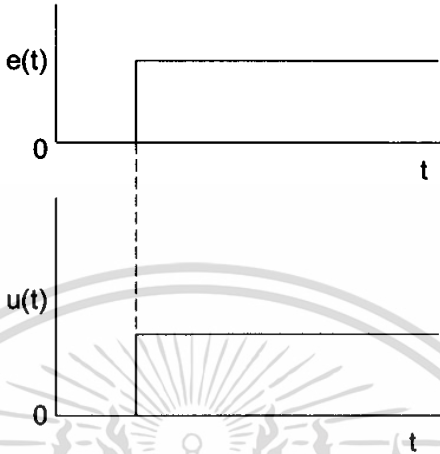


รูปที่ 2.1 การกำหนดช่วงจำกัดของเอาต์พุต

การกำหนด Proportional Band นี้ จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัดไม่ไปสู่ค่าอนันต์ทั้งทางด้านบวกและทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงที่สุดที่เป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใด ๆ เป็นร้อยละของค่าสูงสุดที่เป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของตัวควบคุม 100% ก็หมายความว่า เอาต์พุตจะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ ไปเป็นค่าสูงสุดที่เป็นไปได้ ซึ่งจะได้ว่า

$$K_p = \frac{100}{\text{Proportional Band}} \tag{2.3}$$

เนื่องจากเอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอินพุต ดังนั้น ถ้าหากอินพุตมีลักษณะเป็น สัญญาณระดับ (Step) เอาต์พุตที่ได้ก็จะมีลักษณะเป็น สัญญาณระดับเช่นกันโดยลักษณะของกราฟแสดงอินพุตและเอาต์พุตจะมีสัดส่วนที่แน่นอนค่าหนึ่ง ตามรูปที่2.2 โดยรูปนี้แสดงถึงการตอบสนองของตัวควบคุม เมื่ออินพุตอยู่ในช่วง Proportional Band

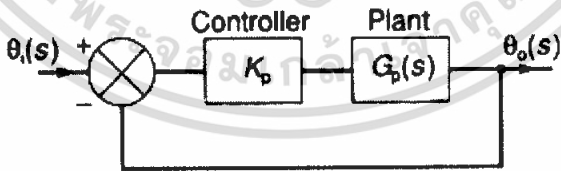


รูปที่ 2.2 ผลการตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพีนี้มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น คัน กัดก็ได้ ลักษณะของระบบที่มีตัวควบคุมแบบพี แสดงได้ในรูปที่ 2.3 และมีฟังก์ชันถ่ายโอนวงเปิดเป็น

$$G(s) = K_p G_p(s) \tag{2.4}$$

เมื่อ $G_p(s)$ เป็นฟังก์ชันถ่ายโอนของพลานต์หรือกระบวนการ



รูปที่ 2.3 บล็อกไดอะแกรมของระบบควบคุมแบบพี

ข้อเสียของระบบที่มีตัวควบคุมแบบพี คือ ถ้าระบบเป็นชนิด (Type) 0 ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบเป็นชนิด 0 เหมือนเดิม และทำให้เกิดความผิดพลาดที่สภาวะคงตัว (Steady-State Error) เนื่องจากตัวควบคุมไม่ได้ทำการเพิ่มโพล (Pole) ใหม่ให้กับระบบ เพียงแต่เปลี่ยนตำแหน่งของโพลเท่านั้น เนื่องจากระบบควบคุมแบบป้อนกลับหนึ่งหน่วย มีฟังก์ชันถ่ายโอนของระบบวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_p G_p(s)}{1 + K_p G_p(s)} \quad (2.5)$$

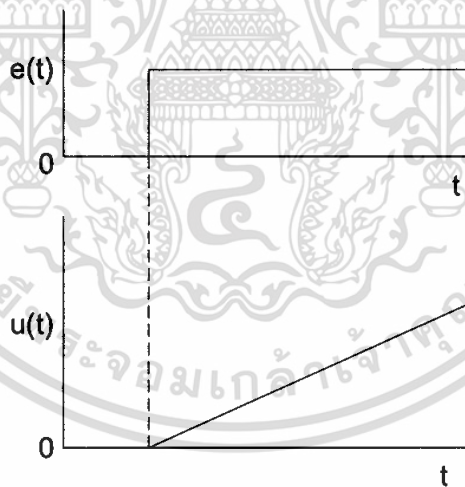
ซึ่งมีสมการคุณลักษณะเป็น $1 + K_p G_p(s) = 0$ และมีรากเปลี่ยนไปตามค่าของ K_p เท่านั้น

2.1.1.2 ระบบควบคุมแบบไอ

ในระบบควบคุมแบบไอ เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับการอินทิเกรตสัญญาณของค่าผิดพลาดเทียบกับเวลา หรือ

$$u(t) = K_i \int_0^t e(t) dt \quad (2.6)$$

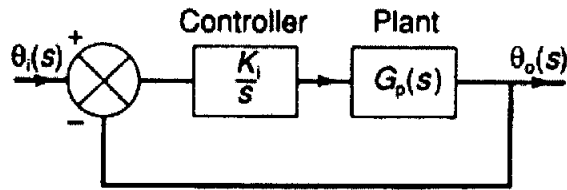
เมื่อ K_i เป็นค่าคงที่เรียกว่า อัตราขยายอินทิกรัล(Integral Gain) ซึ่งจะมีหน่วยเป็น sec^{-1} รูปที่ 2.4 แสดงลักษณะการตอบสนองของระบบควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุตแบบระดับ ค่า $u(t)$ ณ เวลาใดๆ หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุตจากเวลา 0 ถึง t คูณด้วยอัตราขยายอินทิกรัล ดังนั้น เนื่องจากเริ่มมีสัญญาณอินพุตแบบระดับ เอาท์พุทที่ออกจากตัวควบคุมมีค่ามากขึ้นเรื่อยๆ ด้วยอัตราที่คงที่ ในลักษณะรูปสัญญาณลาดเอียง (Ramp Signal)



รูปที่ 2.4 ผลการตอบสนองของตัวควบคุมแบบไอ

จากสมการที่ (2.6) จะได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบไอ เป็น

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.7)$$



รูปที่ 2.5 บล็อกไดอะแกรมของระบบควบคุมแบบไอ

ทำให้ระบบควบคุมแบบไอที่แสดงในรูปที่ 2.5 มีฟังก์ชันถ่ายโอนของระบบวงเปิดเป็น $\frac{K_i}{s} G_p(s)$ และทำให้มีฟังก์ชันถ่ายโอนวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\frac{K_i}{s} G_p(s)}{1 + \frac{K_i}{s} G_p(s)} \quad (2.8)$$

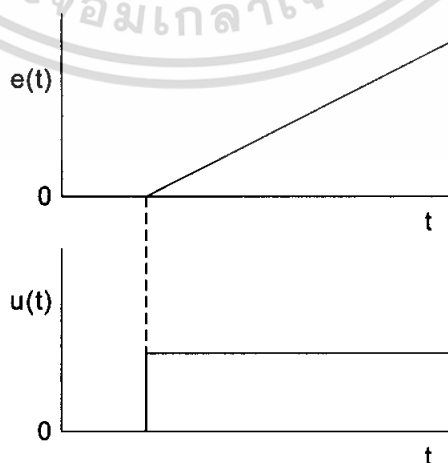
ดังนั้น จากฟังก์ชันถ่ายโอนวงเปิด จะเห็นว่าระบบควบคุมแบบไอ จะทำให้ระบบชนิด 0 เป็นชนิด 1 ซึ่งทำให้ระบบควบคุมวงปิดมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เมื่อสัญญาณอินพุตเป็นแบบระดับ

2.1.1.3 ระบบควบคุมแบบดี

ระบบควบคุมแบบดี นี้เอาท์พุตจะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงความผิดพลาดเทียบกับเวลา นั่นคือ

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.9)$$

เมื่อ K_d คือ Derivative Gain และมีหน่วยเป็นวินาที



รูปที่ 2.6 ผลตอบสนองของตัวควบคุมแบบดี

รูปที่ 2.6 แสดงผลตอบสนองของตัวควบคุมแบบดี เมื่อสัญญาณอินพุตเป็นสัญญาณแบบความชัน จากสมการที่ (2.9) จะเห็นว่าตัวควบคุมแบบดีจะให้สัญญาณออกจากตัวควบคุมมีค่ามากเมื่อค่าอินพุตความผิดพลาดมีแนวโน้มเพิ่มในอัตราที่สูงก่อนที่จะเกิดความผิดพลาดขึ้นมากจริงๆ

อย่างไรก็ตาม หากความผิดพลาดมีค่าคงที่ ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบดีนี้ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่ หรือเปลี่ยนแปลงอย่างช้าๆ ดังนั้น การควบคุมแบบนี้จึงไม่เป็นที่นิยมใช้เพียงลำพัง แต่มักใช้ควบคู่รวมกับการควบคุมแบบอื่น ฟังก์ชันถ่ายโอนของตัวควบคุมแบบดี คือ

$$G_c(s) = \frac{U(s)}{E(s)} = K_d s \quad (2.10)$$

ดังนั้น ระบบควบคุมแบบดีจึงมีฟังก์ชันถ่ายโอนของระบบวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \quad (2.11)$$



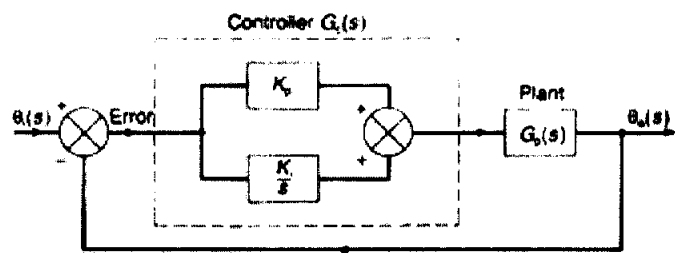
รูปที่ 2.7 บล็อกไดอะแกรมของระบบควบคุมแบบดี

ถ้าหากระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดีจะลดเทอม s ในพหุนามส่วนของฟังก์ชันถ่ายโอนวงเปิดลง ซึ่งจะปลดชนิดของระบบลง 1 อย่างไรก็ตามดังที่ได้กล่าวก่อนหน้านี้แล้วว่า การควบคุมแบบดีนี้จะไม่ใช่เพียงลำพังแต่จะใช้รวมกับการควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดี ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

ในทางปฏิบัติการนำการควบคุมดีไปใช้งานนั้นค่อนข้างจะลำบาก ดังนั้น ในทางปฏิบัติโดยทั่วไปจะเป็นการประมาณการควบคุมแบบดีโดยใช้ตัวชดเชยแบบมุนนำ (Lead Compensator)

2.1.1.4 ระบบควบคุมแบบพีไอ

การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการนำการควบคุมแบบพีรวมกับแบบไอ (PI: Proportional plus Integral Control) ซึ่งลักษณะของระบบควบคุมเป็นตามรูปที่ 2.8

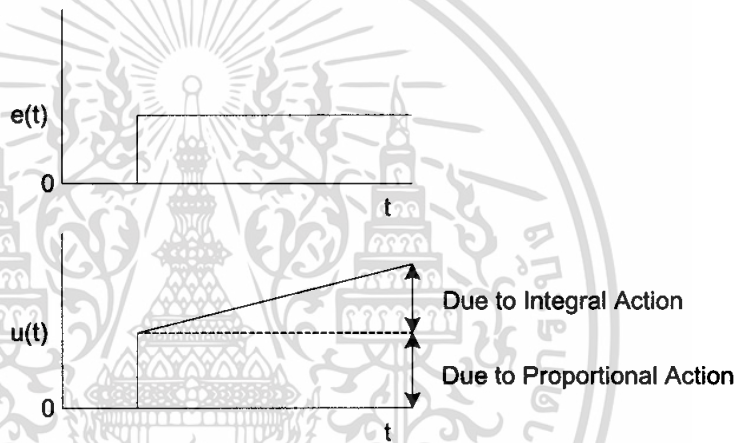


รูปที่ 2.8 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ

และจะมีเอาต์พุตของตัวควบคุมเป็น

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt$$

(2.12)



รูปที่ 2.9 การตอบสนองของระบบควบคุมแบบพีไอ

เอาต์พุตของตัวควบคุมแบบพีไอเมื่อได้รับอินพุตเป็นแบบระดับแสดงได้ดังรูปที่ 2.9 และจะมีฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอ เป็น

$$\begin{aligned} G(s) &= \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \\ &= K_p \frac{(s + K_i/K_p)}{s} \\ &= \frac{K_p [s + (1/\tau_i)]}{s} \end{aligned}$$

(2.13)

โดยที่ค่าเวลาคงตัวอินทิกรัล (Integral Time Constant) คือ $\tau_i = \frac{K_p}{K_i}$ และจะทำให้ฟังก์ชันถ่ายโอนของระบบวงเปิดเป็น

$$G(s) = G_c(s) G_p(s)$$

$$= \frac{K_p[s + (1/\tau_i)G_p(s)]}{s} \quad (2.14)$$

และจะได้ฟังก์ชันถ่ายโอนวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_p[s + (1/\tau_i)G_p(s)]}{s + K_p[s + (1/\tau_i)G_p(s)]} \quad (2.15)$$

จะเห็นว่ามีซีโรที่ $s = -1/\tau_i$ และมีโพลที่ $s = 0$ ซึ่งเป็นการเพิ่มโพลให้กับฟังก์ชันถ่ายโอนวงเปิดของระบบ จึงเป็นการเพิ่มชนิดของระบบขึ้นไป 1 ซึ่งทำให้ระบบควบคุมพีไอนี้ไม่มีความผิดพลาดที่สภาวะคงตัวสำหรับอินพุตแบบระดับ

2.1.1.5 ระบบควบคุมแบบพีดี

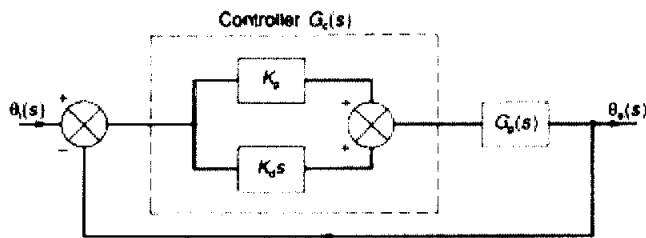
การควบคุมแบบพีดีรวมกับการควบคุมแบบพี (PD: Proportional plus Derivative Control) ดังที่แสดงในรูปที่ 2.10 จะมีฟังก์ชันถ่ายโอนระบบวงเปิดเป็น

$$\begin{aligned} G(s) &= (K_p + K_d s)G_p(s) \\ &= K_p(1 + \tau_d s)G_p(s) \end{aligned} \quad (2.16)$$

และจะได้ฟังก์ชันถ่ายโอนวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_p(1 + \tau_d s)G_p(s)}{1 + K_p(1 + \tau_d s)G_p(s)} \quad (2.17)$$

เมื่อ $\tau_d = \frac{K_p}{K_d}$ คือ ค่าเวลาคงตัวอนุพันธ์ (Derivative Time Constant) ซึ่งในการควบคุมแบบนี้ จะมีซีโรเพิ่มขึ้นที่ $s = -\tau_d$ และไม่มีการเปลี่ยนแปลงชนิดของระบบ ซึ่งทำให้ค่าความผิดพลาดที่สภาวะคงตัวยังคงมีอยู่



รูปที่ 2.10 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี

2.1.1.6 ระบบควบคุมแบบพีไอดี

การควบคุมโดยใช้แบบพืรวมกับแบบโอและรวมกับแบบดี (PID: Proportional plus Integral plus Derivative Control) หรือ ที่เรียก การควบคุมแบบ 3 เทอม (Tree-Term Control) ระบบจะมีลักษณะตามรูปที่ 2.11 ทำให้เอาท์พุตของตัวควบคุมเมื่อรับอินพุตเป็นความผิดพลาด $e(t)$ ดังนี้

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.18)$$

เพราะฉะนั้น ฟังก์ชันถ่ายโอนของตัวควบคุม จะเป็น

$$\begin{aligned} G_c(s) &= \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \\ &= K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \end{aligned} \quad (2.19)$$

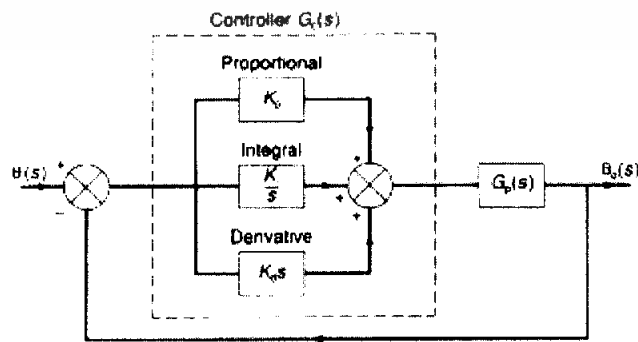
และฟังก์ชันถ่ายโอนของระบบวงเปิดของระบบควบคุมแบบพีไอดี คือ

$$\begin{aligned} G(s) &= G_c(s) G_p(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) G_p(s) \\ &= \frac{K_p (\tau_i s + 1 + \tau_i \tau_d s^2) G_p(s)}{\tau_i s} \end{aligned} \quad (2.20)$$

ดังนั้น จะได้ฟังก์ชันถ่ายโอนวงปิดเป็น

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_p (1 + \tau_i s + \tau_i \tau_d s^2) G_p(s)}{\tau_i s + K_p (1 + \tau_i s + \tau_i \tau_d s^2) G_p(s)} \quad (2.21)$$

เนื่องจาก การควบคุมแบบพีไอดี จะเพิ่มจำนวนขีโรให้กับระบบเท่ากับ 2 ตัวและเพิ่มจำนวนโพล 1 ตัว และทำให้ชนิดของระบบเพิ่มขึ้น 1 ดังนั้น ความผิดพลาดที่สภาวะคงตัวจะมีค่าเป็นศูนย์



รูปที่ 2.11 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี

2.1.2 ตัวควบคุมแบบฟัซซีลอจิก

ฟัซซีลอจิก (Fuzzy Logic) เป็นทฤษฎีทางคณิตศาสตร์ที่สามารถนำมาออกแบบตัวควบคุมฟัซซี ดังนั้น เนื้อหาในบทนี้จะกล่าวถึงทฤษฎีของฟัซซีลอจิก จากนั้นก็กล่าวถึงบล็อกไดอะแกรมโดยใช้ฟัซซีลอจิกในการควบคุม

2.1.2.1 ทฤษฎีของฟัซซีลอจิก

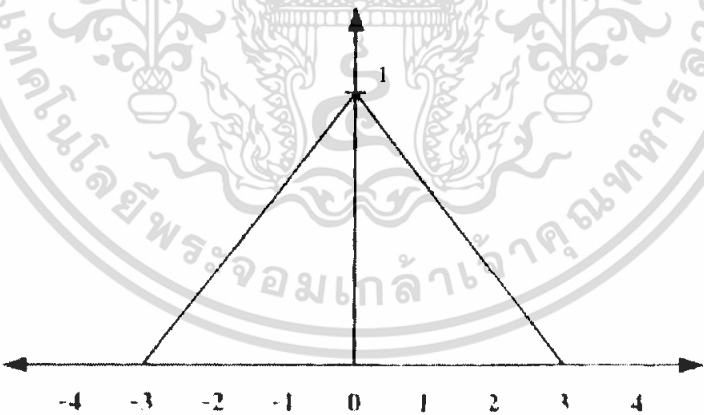
ทฤษฎีของฟัซซีลอจิกมี 5 ส่วน ดังนี้

1. ฟัซซีเซต

ฟัซซีเซตเป็นเซตที่แสดงถึงความสัมพันธ์ของสมาชิกภายในกลุ่มแต่ละตัวกับค่าจำกัดความของเซตนั้นๆ โดยความสัมพันธ์นี้จะถูกแสดงในลักษณะของระดับความเป็นสมาชิกที่มีค่าอยู่ในช่วง $[0,1]$ แทนที่จะแสดงว่าสมาชิกตัวใดเป็นสมาชิกหรือไม่เป็นสมาชิกของเซตนั้นอย่างเซตธรรมดา ซึ่งมีค่าเป็น $\{0, 1\}$ ถ้าเรากำหนดให้ U เป็นเซตเอกภพสัมพัทธ์ และฟัซซีเซต A มีสมาชิกของเซตเป็น $x_1, x_2, x_3, \dots, x_n$ โดยที่ x_i ($i = 1, 2, 3, \dots, n$) เป็นค่าฟัซซีเซต ดังนั้นฟัซซีเซต A สามารถแสดงในรูปของความสัมพันธ์ฟัซซีเซตได้โดย $\mu_A(x_i)$ ความสัมพันธ์เช่นนี้ในทางทฤษฎีของฟัซซีจะเรียกว่า ฟังก์ชันความเป็นสมาชิก (Membership Function) และสามารถเขียนเป็นสัญลักษณ์ทางคณิตศาสตร์ได้ดังนี้

$$A = \{(x, \mu_A(x)) | x \in U\} \tag{2.22}$$

โดยที่ $\mu_A(x_i)$ ก็คือฟังก์ชันความเป็นสมาชิกของ A



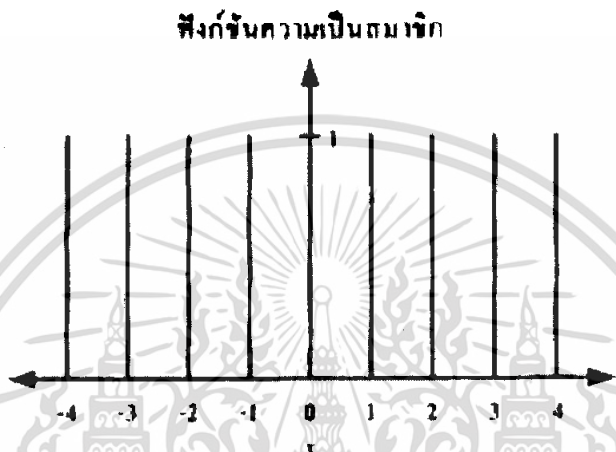
รูปที่ 2.12 ฟัซซีเซต

จากรูปที่ 2.12 แสดงให้เห็นถึงฟัซซีเซต โดยกำหนดให้เอกภพสัมพัทธ์ U มีค่าต่างๆดังนี้ $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ โดยที่เซต A มีสมาชิก 6 ตัวคือ $x_i = \{-3, -2, -1, 0, 1, 2, 3\}$ ซึ่งจะพบว่าสมาชิกของเซต A แต่ละตัวมีระดับการเป็นสมาชิกค่านั้นๆเป็นคู่ลำดับกันไปทำให้พิจารณาได้ว่าฟัซซีเซตจะไม่สามารถแยกแยะการเป็นสมาชิกแต่ละค่าได้ แต่สมาชิกจะมีระดับการเป็นสมาชิกอยู่ในช่วง 0 ถึง 1

2. ฟังก์ชันความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิกจะถูกกำหนดขึ้นตามคำจำกัดความของเทอมเซตที่มีความยืดหยุ่น และตรงตามความเป็นจริง โดยการกำหนดฟังก์ชันความเป็นสมาชิกนั้นจะต้องสัมพันธ์กับระบบที่ทำการควบคุม ซึ่งรูปแบบฟังก์ชันความเป็นสมาชิกมีอยู่หลายแบบด้วยกัน ดังรายละเอียดต่อไปนี้

ฟังก์ชันความเป็นสมาชิกแบบซิงเกิลตัน (Singleton Membership Function) จะมีฟังก์ชันความเป็นสมาชิกเพียงตัวเดียว และมีค่าฟังก์ชันความเป็นสมาชิกเท่ากับ 1



รูปที่ 2.13 กราฟฟังก์ชันแสดงความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิกแบบลูกสามเหลี่ยม (Triangular Membership Function) ถูกกำหนดโดย 3 พารามิเตอร์ {a,b,c} แสดงดังนี้

$$\text{triangle}(x;a,b,c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \tag{2.23}$$

โดยที่ $a < b < c$ พารามิเตอร์ {a,b,c} เป็นตัวตัดสินใจฟังก์ชันความเป็นสมาชิกจากค่า x

ฟังก์ชันความเป็นสมาชิกแบบสี่เหลี่ยมคางหมู (Trapezoidal Membership Function) ถูกกำหนดโดย 4 พารามิเตอร์ {a,b,c,d} แสดงดังนี้

$$\text{trapezoid}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (2.24)$$

โดยที่ $a < b < c < d$ พารามิเตอร์ $\{a, b, c, d\}$ ตัวตัดสินค่าฟังก์ชันความเป็นสมาชิกจากค่า x

ฟังก์ชันความเป็นสมาชิกแบบเกาส์เซียน (Gaussian Membership Function) ถูกกำหนดโดย 2 พารามิเตอร์ $\{c, \sigma\}$ แสดงดังนี้

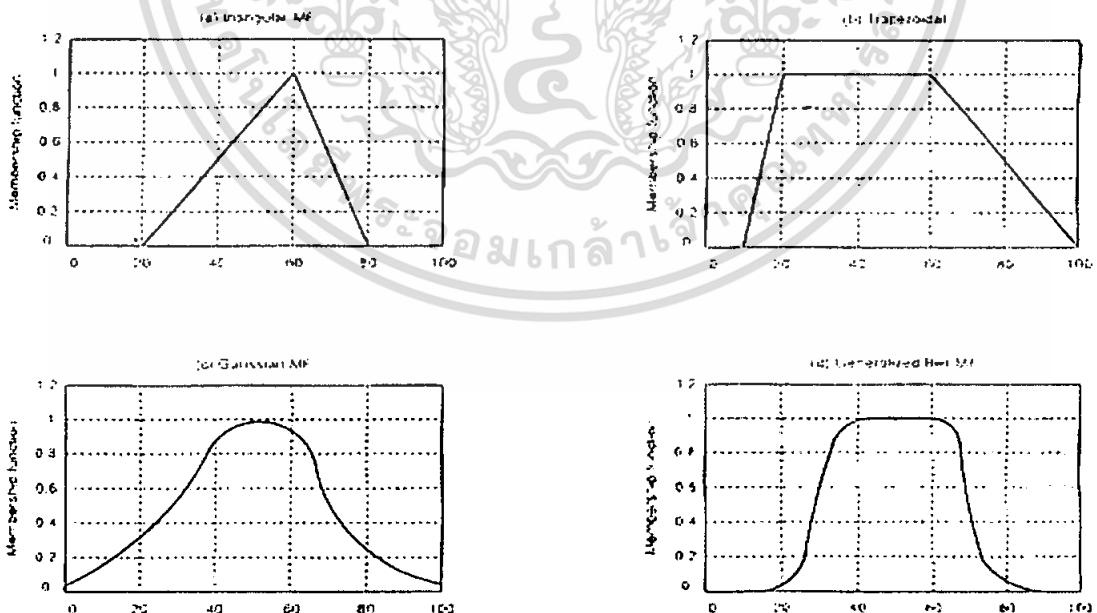
$$\text{gaussian}(x; c; \sigma) = \exp^{-0.5\left(\frac{x-c}{\sigma}\right)^2} \quad (2.25)$$

โดยที่ c เป็นค่ากลางและ σ เป็นความกว้างของฟังก์ชันความเป็นสมาชิกแบบเกาส์เซียน ตามลำดับ

ฟังก์ชันความเป็นสมาชิกแบบระฆัง (Bell Membership Function) ถูกกำหนดโดย 3 พารามิเตอร์ $\{a, b, c\}$ แสดงดังนี้

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.26)$$

โดยที่ b ต้องเป็นบวก



รูปที่ 2.14 กราฟฟังก์ชันแสดงความเป็นสมาชิกในแบบต่างๆ (a) สามเหลี่ยม ($x; 20, 60, 80$);

(b) สี่เหลี่ยมคางหมู ($x; 10, 20, 60, 95$); (c) เกาส์เซียน ($x; 50, 20$); (d) ระฆัง ($x; 20, 40, 50$)

3. ตัวดำเนินการฟัซซีเซต

ตัวดำเนินการของฟัซซีเซตส่วนใหญ่จะมีตัวดำเนินการคล้ายกับเซตธรรมดา เช่น อินเตอร์เซกชัน ยูเนียน คอมพลิเมนต์ เป็นต้น ซึ่งตัวดำเนินการเหล่านี้จะใช้ในการอนุมานฟัซซีเซตเพื่อประโยชน์ในการตัดสินใจโดยประมาณค่าของตัวแปร ในที่นี้จะเสนอตัวดำเนินการของฟัซซีเซตบางส่วนเฉพาะที่ใช้ในแบบจำลองการตัดสินใจของระบบควบคุมโดยใช้ฟัซซีเซตเท่านั้น โดยกำหนดให้ U เป็นเซตเอกภพสัมพัทธ์ A, B เป็นเทอมเซตใดๆที่อยู่ใน U และตัวดำเนินการพื้นฐานต่างๆสามารถกำหนดได้ดังนี้

1. เซตย่อย

A เป็นเซตย่อยของ B ($A \subseteq B$) ก็ต่อเมื่อ

$$\mu_A(x_i) \leq \mu_B(x_i); \forall x_i \in U \quad (2.27)$$

2. ยูเนียน

ยูเนียนระหว่าง A และ B ($A \cup B$) ก็ต่อเมื่อ

$$\mu_{A \cup B}(x_i) = \text{Max}(\mu_A(x_i), \mu_B(x_i)); \forall x_i \in U \quad (2.28)$$

3. อินเตอร์เซกชัน

อินเตอร์เซกชันระหว่าง A และ B ($A \cap B$) ก็ต่อเมื่อ

$$\mu_{A \cap B}(x_i) = \text{Min}(\mu_A(x_i), \mu_B(x_i)); \forall x_i \in U \quad (2.29)$$

4. คอมพลิเมนต์

A เป็นคอมพลิเมนต์ของ B ($A = B'$) ก็ต่อเมื่อ

$$\mu_A(x_i) = \mu_{B'}(x_i) = 1 - \mu_B(x_i); \forall x_i \in U \quad (2.30)$$

5. A ยกกำลังด้วยค่า α (A^α) กำหนดได้โดย

$$\mu_{A^\alpha}(x_i) = (\mu_A(x_i))^\alpha; \forall x_i \in U \quad (2.31)$$

6. Concentration ของเซต A กำหนดได้โดย

$$\text{CON}(A) \Rightarrow \mu_{\text{CON}(A)}(x_i) = (\mu_A(x_i))^2; \forall x_i \in U \quad (2.32)$$

7. Dilation ของเซต A กำหนดได้โดย

$$\text{DIL}(A) \Rightarrow \mu_{\text{DIL}}(x_i) = (\mu_A(x_i))^{0.5}; \forall x_i \in U \quad (2.33)$$

4. ส่วนขยายของฟัซซีเซต

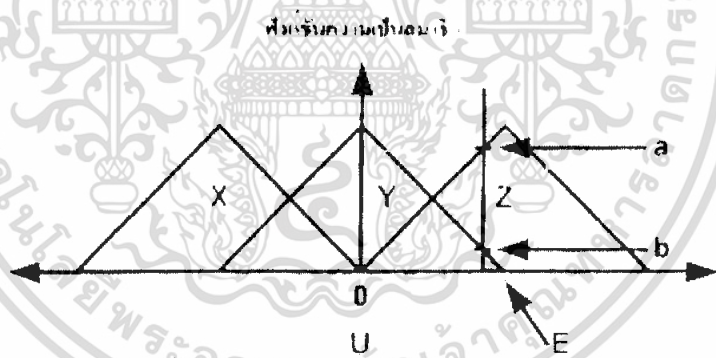
ส่วนขยาย (Hedge) ของฟัซซีเซตจะมีคุณสมบัติคล้ายกับคำคุณศัพท์ในประโยคที่ใช้พูดคุยติดต่อกันซึ่งจะขยายใจความของประโยคให้มีความหมายที่ชัดเจนยิ่งขึ้นหรือทำให้ความหมายเปลี่ยนไป สำหรับ Hedge ของฟัซซีเซตนั้นจะขยายขอบเขตของตัวแปรฟัซซีเซต เพื่อให้การประมาณค่าของตัวแปรมีความถูกต้องมากยิ่งขึ้น เช่น ตัวดำเนินการ Concentration, Dilation เป็นต้น

5. ตัวควบคุมฟัซซี

จากหัวข้อที่กล่าวมาทั้งหมดในเรื่องของฟัซซีเซต จะถูกนำมาใช้ในการออกแบบตัวควบคุมฟัซซีโดยโครงสร้างของตัวควบคุมฟัซซีประกอบไปด้วย 4 หน่วยหลัก ดังนี้

1. การฟัซซีฟิเคชัน

ฟัซซีฟิเคชัน (Fuzzification) คือ กระบวนการแปลค่าของตัวแปรอินพุตไปเป็นค่าระดับความเป็นสมาชิกของแต่ละเทอมเซต โดยใช้ฟังก์ชันความเป็นสมาชิกที่ได้นำเสนอไปแล้ว ซึ่งการฟัซซีฟิเคชันสามารถทำได้หลายวิธี ยกตัวอย่างเช่น การฟัซซีฟิเคชันโดยใช้ฟัซซีซิงเกิลตัน ฟัซซีนัมเบอร์ หรือไฮบริดนัมเบอร์ ในปริภูมิพหุนัยจะกล่าวเฉพาะในส่วนของการฟัซซีฟิเคชันโดยใช้ฟัซซีซิงเกิลตันเท่านั้น



รูปที่ 2.15 การฟัซซีฟิเคชันแบบซิงเกิลตัน

จากรูปที่ 2.15 จะเห็นว่า กลุ่มเซตทางอินพุตประกอบด้วยเซต X, Y และ Z ซึ่งเป็นฟัซซีเซตแบบฟัซซีนัมเบอร์ และเซตที่ถูกนำมาอินเตอร์เซกชันคือเซต E ที่เป็นเซตแบบซิงเกิลตัน ซึ่งการอินเตอร์เซกชันดังรูป จะทำให้เกิดจุดซ้อนทับกันสองจุด คือ a และ b ซึ่งเป็นจุดที่บอกค่าระดับความเป็นสมาชิกที่จะเข้ามาเป็นตัวแทนทางอินพุต

2. ฐานกฎการควบคุมฟัซซี

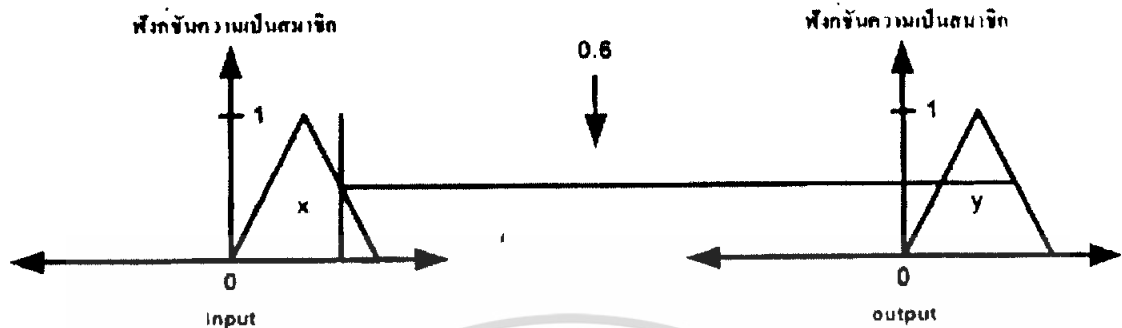
ฐานกฎการควบคุมฟัซซี (Fuzzy Rule Base) นั้นเป็นการควบคุมโดยใช้เงื่อนไข เช่น

IF X is hot THEN Y is zero

ซึ่ง X is hot เป็นเงื่อนไขฝั่งอินพุต และ Y is zero เป็นส่วนของเอาต์พุต

3. การอินเฟอเรนซ์แบบฟัซซี

การอินเฟอเรนซ์(Inference Unit) คือ กระบวนการในการหาข้อสรุปจากหลักฐาน โดยจะหาค่าเอาต์พุตของระบบฟัซซี ด้วยการหาระดับความเป็นสมาชิกแล้วจึงคำนวณค่าเอาต์พุตออกมา



รูปที่ 2.16 การอินเฟอเรนซ์แบบฟัซซี

4. การดีฟัซซีฟิเคชัน

การดีฟัซซีฟิเคชัน (Defuzzification) คือ กระบวนการหาเอาต์พุตเพียงค่าเดียวที่เหมาะสมที่สุดเพื่อเป็นตัวแทนของฟัซซีเอาต์พุต ในวิทยานิพนธ์นี้จะกล่าวถึงเพียงแบบเดียว คือ การหาค่าเฉลี่ยของน้ำหนัก (Weighted Average Method) ซึ่งหาได้จากสมการ

$$v_0 = \frac{\sum \mu_B(v) \cdot v}{\sum \mu_B(v)} \quad (2.34)$$

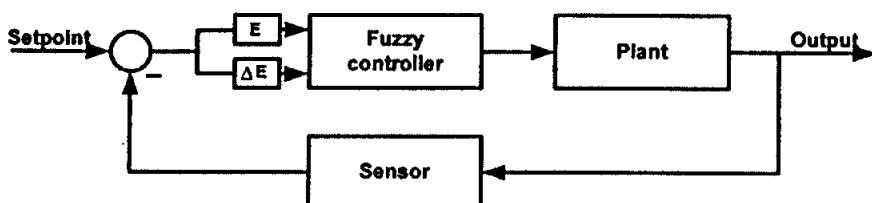
โดยที่ v_0 คือ ค่าที่ได้รับจากการดีฟัซซีฟิเคชัน

μ_B คือ ระดับความเป็นสมาชิก

v คือ สมาชิกของเอกภพสัมพัทธ์ทางเอาต์พุต

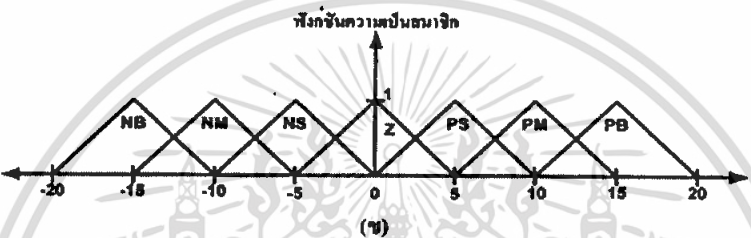
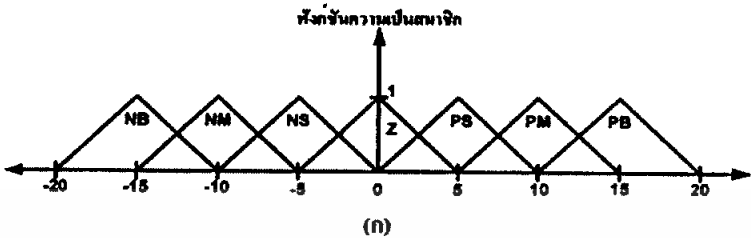
2.1.2.2 การควบคุมยานสำรวจใต้น้ำโดยใช้ฟัซซีลอจิก

การควบคุมยานสำรวจใต้น้ำที่นำเสนอในปริญญานิพนธ์นี้จะใช้หลักการของตัวควบคุมฟัซซีลอจิกซึ่งมีบล็อกไดอะแกรมดังรูปที่ 2.17 ซึ่งอินพุต



รูปที่ 2.17 การควบคุมยานสำรวจใต้น้ำโดยใช้ฟัซซีลอจิก

อินพุตของแบบจำลองฟัซซีลอจิกมีสองตัวแปรคือ E : Error และ ΔE : Δ Error ส่วนของเอาต์พุตมีตัวแปรหนึ่งตัว คือ ความเร็วในการหมุนของมอเตอร์ กำหนดให้ตัวแปรของอินพุตแต่ละตัวมีเทอมเซตอย่างละเจ็ดเทอม คือ NB, NM, NS, Z, PS, PM, PB ในรูปที่ 2.18 แสดงเทอมเซตของตัวแปรทั้งสองอินพุต



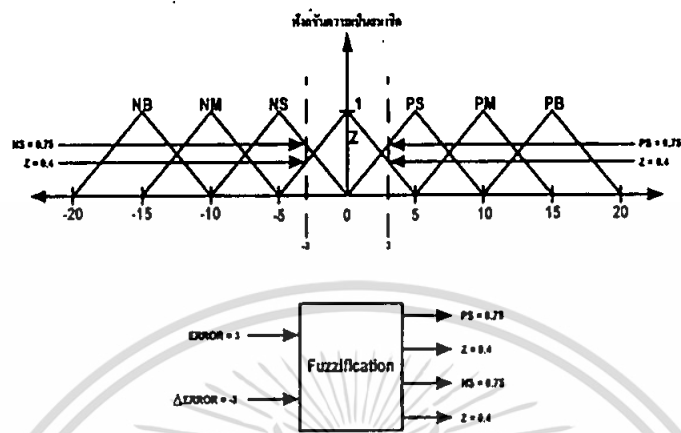
รูปที่ 2.18 เทอมเซตของตัวแปรอินพุต (ก) ERROR (ข) Δ ERROR

ในส่วนของเทอมเซตเอาต์พุตได้ออกแบบเป็นแบบฟัซซีซิงเกิลตัน และเมื่อนำแต่ละเทอมเซตของแต่ละตัวแปรมาเขียนเพื่อกำหนดกฎการควบคุมทั้งหมด 49 กฎ โดยเขียนเป็นเมตริกซ์ที่แสดงความสัมพันธ์ร่วมกันของตัวแปรได้ดังรูปที่ 2.19 ซึ่งเมตริกซ์จะถูกเรียกว่า Fuzzy Associative (FAM) โดยจะมีมิติเท่ากับจำนวนตัวแปรของอินพุต ซึ่งในตัวอย่างนี้มีสองตัวแปร ดังนั้น FAM จะมีเมตริกซ์เป็นสองมิติ ขนาด $M \times N$ เมื่อ M แทนจำนวนเทอมเซตของตัวแปร ERROR และ N เป็นจำนวนเทอมเซตของตัวแปร Δ ERROR และสมาชิกของเมตริกซ์นี้คือค่าของตัวแปรเอาต์พุต ซึ่งเป็นเมตริกซ์แบบซิงเกิลตัน ซึ่งที่มาของค่าต่างๆในตารางที่ 2.1 เป็นการกำหนดขึ้นมาเอง เพื่อใช้ในการยกตัวอย่าง

ตารางที่ 2.1 แสดงเมตริกซ์ FAM ของระบบควบคุม

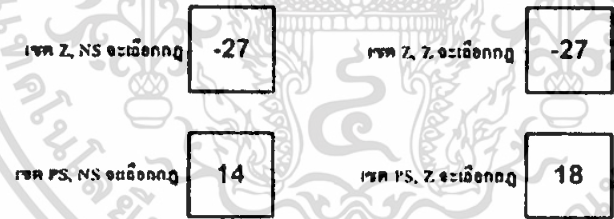
		Δ ERROR						
		NB	NM	NS	Z	PS	PM	PB
ERROR	NB	-50	-50	-50	-40	-50	-50	-50
	NM	-45	-45	-45	-35	-45	-45	-45
	NS	-40	-40	-40	-40	-40	-41	-41
	Z	-36	-31	-27	-27	-27	-28	7
	PS	10	10	14	18	20	31	31
	PM	34	34	34	34	34	34	34
	PB	40	40	40	50	50	50	50

สมมติค่า ERROR ที่วัดได้ที่เวลา t เท่ากับ 3 จากกฎฟัซซี่ ชุดของ PS และ Z จะถูกนำมาพิจารณา ซึ่งได้ค่าเป็น 0.75 และ 0.4 ในทำนองเดียวกัน ค่า $\Delta ERROR$ ที่เวลา t จะมีค่าเป็น -3 ดังนั้น จะได้ชุดของ NS และ Z มีค่าเท่ากับ 0.75 และ 0.4 ซึ่งกระบวนการนี้เรียกว่ากระบวนการฟัซซี่ฟิเคชัน โดยแสดงให้เห็นดังรูปที่ 2.19



รูปที่ 2.19 การทำฟัซซี่ฟิเคชันของระบบควบคุม

หน่วยอินเฟอร์เรนซ์จะรับค่าจากหน่วยฟัซซี่ฟิเคชันแล้วนำไปเลือกฐานกฎการควบคุมฟัซซี่ที่ได้จากประสบการณ์ของผู้เชี่ยวชาญซึ่งในที่นี้คือเมตริกซ์ FAM ซึ่งจะได้เอาที่พุดส่งต่อไปยังหน่วยดีฟัซซี่ฟิเคชันต่อไป ชุดของ ERROR และชุดของ $\Delta ERROR$ ที่ได้จากหน่วยฟัซซี่ฟิเคชันดังแสดงในรูปที่ 2.19 จะถูกนำไปเลือกฐานกฎการควบคุมฟัซซี่ได้ 4 เงื่อนไข ดังรูปที่ 2.20



Set(ERROR, ΔERROR)	Output	Membership Function
Z, NS	-27	0.4
Z, Z	-27	0.4
PS, NS	14	0.75
PS, Z	18	0.4

รูปที่ 2.20 การดีฟัซซี่ฟิเคชันของระบบควบคุม

ซึ่งสามารถคำนวณค่าเอาต์พุตได้ดังนี้

$$\text{Output} = \frac{(-27 \times 0.4 \times 2) + (14 \times 0.75) + (18 \times 0.4)}{(0.4 + 0.4 + 0.75 + 0.4)} = -2$$

การกำหนดหน่วยฟัซซีฟิเคชันในส่วนของการหาค่าเอาต์พุตของตัวควบคุมแบบฟัซซีหรือการแปลงตัวแปรแบบฟัซซีมาเป็นตัวแปรของกระบวนการในหน่วยดีฟัซซีฟิเคชันนี้ จะใช้วิธีการของการหาค่าเฉลี่ยของน้ำหนัก โดยจะนำค่าเซตของเอาต์พุตที่ได้จากหน่วยอินเฟอร์เรนซ์มาคำนวณ เช่น Z, NS ได้ค่าเอาต์พุตเท่ากับ -27 และมีระดับการเป็นสมาชิก 2 ค่าคือ 0.4 และ 0.75 แต่การดีฟัซซีฟิเคชันด้วยวิธีนี้จะใช้หลักการของการกระทำแบบ Min Operator ก็คือการเลือกค่าระดับการเป็นสมาชิกที่ต่ำกว่า ดังนั้น ค่าระดับการเป็นสมาชิกของเงื่อนไขอินพุตนี้ คือ 0.4 ในทำนองเดียวกันค่าเงื่อนไขอินพุตที่เหลืออีก 3 เงื่อนไขก็จะใช้หลักการเดียวกันกับ รูปที่ 2.14 แสดงค่าเอาต์พุตที่ต้องใช้ในการคำนวณของกระบวนการดีฟัซซีฟิเคชัน

การควบคุมด้วยฟัซซีลอจิก เป็นวิธีควบคุมสมัยใหม่อีกวิธีหนึ่งที่อาศัยรูปแบบการแก้ปัญหาหรือการให้เหตุผลของมนุษย์เป็นแนวทางในการวิเคราะห์และควบคุมระบบโดยใช้การดำเนินการเชิงตรรกะแบบฟัซซีเป็นเครื่องมือ ซึ่งตัวควบคุมแบบฟัซซีนีมีลักษณะเป็นตัวประมาณค่าตัวแปรที่ต้องการควบคุม โดยไม่จำเป็นต้องใช้แบบจำลองทางคณิตศาสตร์ของระบบ ทำให้สามารถใช้กับระบบที่ไม่เป็นเชิงเส้นหรือระบบที่ยากแก่การหาแบบจำลองทางคณิตศาสตร์ อีกทั้งยังสามารถใช้ได้กับระบบที่มีหลายอินพุตหลายเอาต์พุต แนวทางของทฤษฎีระบบควบคุมแบบฟัซซีนี สามารถแก้ปัญหาที่เคยเกิดขึ้นกับระบบควบคุมแบบดั้งเดิมได้ นอกจากนี้ ฟัซซีลอจิกไม่ต้องใช้คณิตศาสตร์ที่ซับซ้อน ทำให้สามารถพัฒนาระบบให้ทำงานแบบทันทีได้โดยไม่ยากนัก ในปัจจุบันนี้ มีผู้ค้นคว้าวิจัยเกี่ยวกับระบบ Fuzzy Rule-Base Expert System ระบบ Neuro-Fuzzy ซึ่งเป็นการผสมผสานกันระหว่างฟัซซีกับโครงข่ายประสาท (Neuro Network) รวมทั้งระบบ Adaptive Fuzzy System ที่สามารถปรับการควบคุมตามพฤติกรรมของระบบได้ดียิ่งขึ้นอีกด้วย

2.2 หลักการทำงานของยานสำรวจใต้น้ำ

ยานสำรวจใต้น้ำ คือยานสำรวจที่สามารถลอยอยู่บนผิวน้ำหรือดำลงสู่ใต้น้ำได้ โดยอาศัยคุณสมบัติของการเปลี่ยนแปลงน้ำหนักหรือแรงกดที่กระทำกับตัวยานสำรวจ

สำหรับโครงการนี้ ตัวยานสำรวจจะลอยอยู่ที่ระดับผิวน้ำโดยมีมอเตอร์ 2 ตัวที่ถูกติดตั้งอยู่ตรงกลางของตัวยานสำรวจใต้น้ำ จะทำหน้าที่ในควบคุมการเคลื่อนที่ในแนวดิ่ง และมอเตอร์อีก 2 ตัวติดตั้งทางด้านท้ายของตัวยานสำรวจใต้น้ำจะทำหน้าที่ควบคุมการเคลื่อนที่ในแนวระนาบ ถ้าต้องการจะให้ตัวยานสำรวจดำลงสู่ใต้น้ำก็สั่งให้มอเตอร์ 2 ตัวที่อยู่ตรงกลางของยานสำรวจใต้น้ำทำงานเพื่อให้ใบพัดสร้างแรงกดแก่ตัวยานสำรวจ ยานสำรวจก็จะดำลงสู่ใต้น้ำได้ การที่จะดำลง

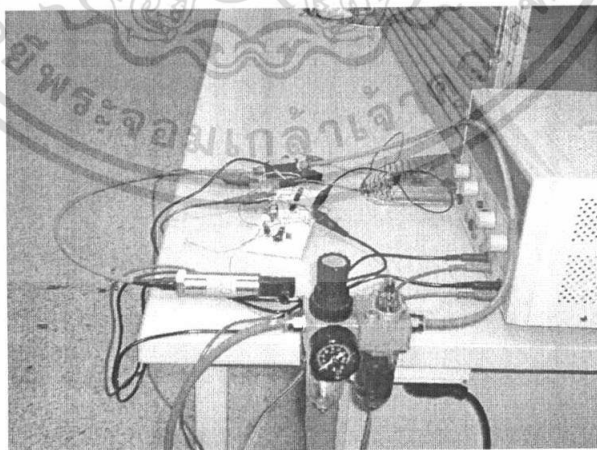
ไปได้ผิวน้ำให้ลึกเท่าใดนั้น ก็ขึ้นอยู่กับระดับความเร็วของการหมุนของมอเตอร์ และเมื่อต้องการจะให้ตัวยานสำรวจลอยขึ้นสู่ผิวน้ำก็ทำได้โดยการสั่งให้มอเตอร์หยุดทำงาน แรงกดที่ตัวยานสำรวจก็จะหมดไป ส่วนการเคลื่อนที่ในแนวระนาบนั้น ก็จะใช้แรงขับเคลื่อนจากมอเตอร์ 2 ตัวที่อยู่ทางด้านท้ายของยานสำรวจ เมื่อต้องการจะให้ยานสำรวจเคลื่อนที่ไปข้างหน้าตรงๆก็สั่งให้มอเตอร์ทั้ง 2 ตัวทำงานพร้อมกัน ถ้าสั่งให้หมุนในทิศทางทวนเข็มนาฬิกาใบพัดที่ติดอยู่กับมอเตอร์ก็จะสร้างแรงดัน ทำให้ตัวยานสำรวจเคลื่อนที่ไป ถ้าต้องการจะให้ยานสำรวจเลี้ยวซ้ายหรือเลี้ยวขวาก็สั่งให้มอเตอร์ทั้ง 2 ตัวทำงานในทิศทางตรงข้ามกัน

2.3 อุปกรณ์ตรวจจับต่างๆ

ในการควบคุมยานสำรวจใต้น้ำ มีความจำเป็นอย่างยิ่งที่จะต้องมีการตรวจวัดตำแหน่งที่อยู่ของยานเทียบกับจุดอ้างอิง โดยเฉพาะระดับความลึกและทิศทาง ดังนั้นในหัวข้อนี้จะกล่าวถึงหลักการทำงานของอุปกรณ์วัดความดันและอุปกรณ์วัดค่ามุม รวมทั้งการนำมาใช้ในการวัดระดับความลึกและทิศทางของยานสำรวจใต้น้ำ

2.3.1 อุปกรณ์วัดความดัน

อุปกรณ์วัดความดัน เป็นอุปกรณ์ที่ใช้ตรวจวัดความดันของของไหล (ก๊าซหรือน้ำ) โดยจะแปลงค่าความดันที่ได้ออกมาในรูปของกระแสไฟฟ้าหรือแรงดันไฟฟ้า สำหรับในโครงการนี้ ตัวอุปกรณ์วัดความดันจะถูกนำมาประยุกต์ใช้ในการวัดระดับความลึกที่ยานสำรวจดำลงไป ทำให้ทราบสถานะของยานสำรวจที่อยู่ลึกจากผิวน้ำลงไปได้อย่างถูกต้อง แล้วจึงนำเอาระดับความลึกที่ได้ไปควบคุมการเคลื่อนที่ของยานสำรวจใต้น้ำต่อไป



รูปที่ 2.21 อุปกรณ์วัดความดันที่ใช้ในโครงการ

2.3.1.1 การควบคุมระดับความลึก

การควบคุมระดับความลึกเป็นส่วนสำคัญส่วนหนึ่งของการควบคุมยานสำรวจใต้น้ำ เนื่องจากเป็นส่วนที่จะทำให้ยานสำรวจใต้น้ำสามารถเคลื่อนที่ขึ้นหรือลงได้ตามตำแหน่งความลึกที่

ผู้ใช้งานต้องการ ในการที่จะควบคุมระดับความลึกแบบอัตโนมัติได้อย่างถูกต้องนั้น จำเป็นที่จะต้องมีการตรวจสอบระดับความลึก แต่เนื่องจากการตรวจสอบความลึกจากผิวน้ำนั้นทำได้ยาก และไม่มีอุปกรณ์ตรวจสอบที่ออกแบบมาสำหรับงานนี้โดยตรง

ดังนั้น จึงได้นำอุปกรณ์วัดความดันมาประยุกต์ใช้ โดยนำค่าความดันที่ได้มาแปรเป็นค่าของระดับความลึก โดยอาศัยความรู้เรื่องเกี่ยวกับความดันในของเหลวซึ่งในที่นี้ก็คือ น้ำ ดังนี้

“ ความดันในของเหลว ณ จุด ๆ หนึ่ง = ความดันบรรยากาศ + ความดันของของเหลวที่กดทับ ”

$$P_{ab} = P_{atm} + P_g \quad (2.47)$$

โดย P_{ab} คือ ความดันสมบูรณ์ (Absolute pressure) (psi)

P_{atm} คือ ความดันบรรยากาศ (Atmosphere pressure) (psi)

P_g คือ ความดันเกจ (Gage pressure) (psi)

โดยความดันของของเหลวที่กดทับ (ในที่นี้ก็คือ น้ำ) หรือความดันเกจนี้ จะแปรผันตามระดับความลึก ดังสมการ

$$P_g = \rho_{water} gh \quad (2.48)$$

เมื่อ ρ คือ ความหนาแน่นของน้ำ (kg/m^3)

g คือ ความเร่งเนื่องจากแรงโน้มถ่วงของโลก (m/s^2)

h คือ ระดับความลึก (m)

จากสมการ (2.47) และ (2.48) จะได้ว่า

$$P_g = P_{atm} + \rho_{water} gh \quad (2.49)$$

ซึ่งค่าความดันบรรยากาศเป็นค่าคงที่มีค่าเท่ากับ 14.7 psi ค่าความหนาแน่นของน้ำก็เป็นค่าคงที่เช่นกัน โดยจะมีค่าเท่ากับ 10^3 kg/m^3 ส่วนระดับความลึกก็คือความลึกที่วัดจากผิวน้ำลงไป

ในโครงการนี้ได้ใช้อุปกรณ์วัดความดัน รุ่น PT150 ผลิตโดยบริษัท Winters Instruments และสามารถวัดความดันได้ในช่วง 0 - 150 psig ความสามารถในการวัดระดับความลึกของอุปกรณ์วัดความดันตัวนี้อยู่ในช่วง ประมาณ 0 - 100 เมตร จากผิวน้ำ

เนื่องจากสถานที่ในการทดลองมีระดับความลึกของน้ำประมาณ 0 - 5 เมตร ดังนั้น จึงหาความดันสมบูรณ์ได้ดังต่อไปนี้

ที่ความลึก $h = 0 \text{ m}$ จะได้

$$P_{ab} = 14.7 + ((10^3 \times 9.81 \times 0) \div (6.985 \times 10^3)) = 14.7 \text{ psi}$$

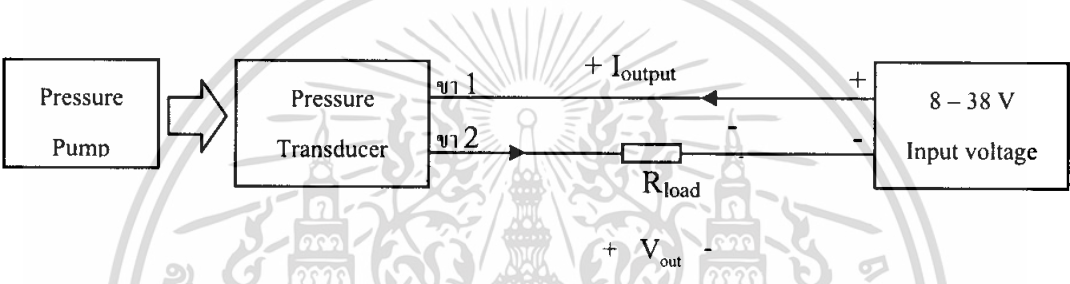
ที่ความลึก $h = 5\text{ m}$ จะได้

$$P_{ab} = 14.7 + ((10^3 \times 9.81 \times 5) \div (6.985 \times 10^3)) = 21.72 \approx 25\text{ psi}$$

ซึ่งจะเห็นว่าช่วงที่เหมาะสมของความดันสมบูรณ์ P_{ab} ที่จะใช้งานคือ $0 - 25\text{ psi}$

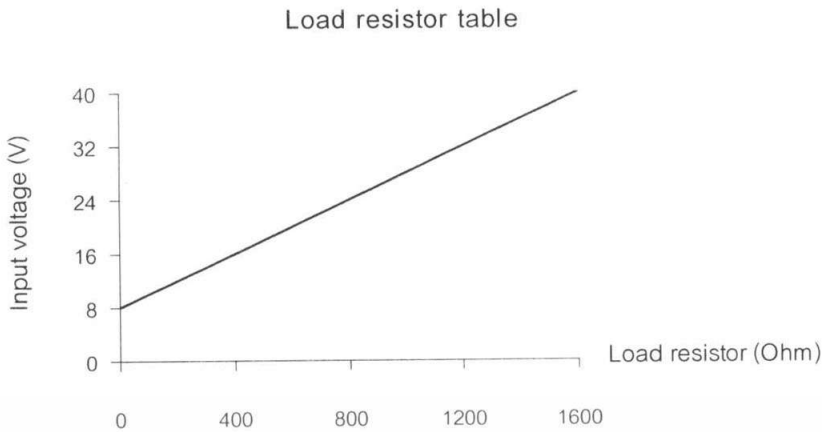
2.3.1.2การประยุกต์ใช้งานอุปกรณ์วัดความดัน

ในการประยุกต์ใช้อุปกรณ์วัดความดันเพื่อวัดระดับความลึกนั้นจะต้องเริ่มต้นจากการหาความสัมพันธ์ระหว่างค่าความดันและค่ากระแสไฟฟ้าหรือแรงดันไฟฟ้าที่ตัวอุปกรณ์วัดจ่ายออกมา ซึ่งจะต้องทดลองซ้ำหลายๆครั้ง แล้วนำข้อมูลที่ได้มาหาความสัมพันธ์ระหว่างค่าความดันและค่ากระแสไฟฟ้าที่ได้หรือค่าแรงดันไฟฟ้าแล้วแต่การนำไปใช้งาน โดยใช้วิธีการทดสอบดังรูป



รูปที่ 2.22 วิธีการที่ใช้ในการทดสอบอุปกรณ์วัดความดัน

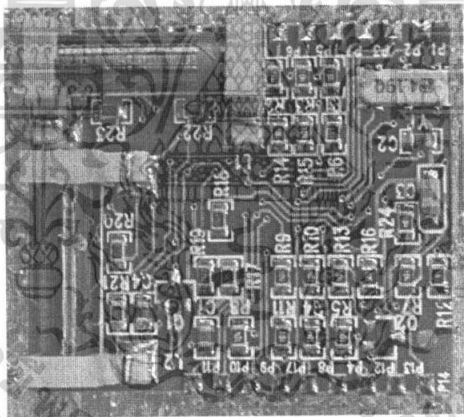
ค่ากระแสไฟฟ้าที่ได้จากการทดสอบดังรูปที่ 2.22 จะมีค่าอยู่ระหว่าง $4 - 20\text{ mA}$ แต่ถ้าต้องการนำไปใช้ในรูปแบบของแรงดันไฟฟ้า ก็สามารถนำค่าแรงดันไฟฟ้า V_{out} ที่ตกคร่อม R_{load} ไปใช้งานได้ ซึ่ง V_{out} จะมีค่าอยู่ในช่วงหนึ่ง ขึ้นอยู่กับค่าของ R_{load} ที่นำมาใช้งาน เช่น ถ้าใช้ R_{load} ที่มีค่า $200\ \Omega$, I_{output} มีค่าคงที่คืออยู่ในช่วง $4 - 20\text{ mA}$ ดังนั้นจะได้ค่าแรงดันไฟฟ้า V_{out} ที่ตกคร่อม R_{load} อยู่ในช่วง $0.8 - 4\text{ V}$ อย่างไรก็ตามค่า R_{load} ที่จะนำมาใช้จะต้องสัมพันธ์กับแหล่งจ่ายแรงดันไฟตรง (Input Voltage) ที่จ่ายให้แก่ตัวอุปกรณ์วัดความดันโดยมีความสัมพันธ์กันตามตารางความต้านทานภาระ (Load Resistor Table) ในรูปที่ 2.23



รูปที่ 2.23 ความสัมพันธ์ระหว่างความต้านทานและแรงดันที่จ่ายให้อุปกรณ์วัดความดัน

จะเห็นได้ว่าย่านที่ใช้งานได้อยู่ในช่วง 0 – 25 psi แต่ในช่วงนี้จะมีการเปลี่ยนแปลงของสัญญาณน้อยมาก ดังนั้น จึงต้องสร้างวงจรขยายย่านเพื่อขยายย่านให้กว้างขึ้นให้เหมาะสมกับการใช้งาน

2.3.2 อุปกรณ์วัดค่ามุม (Compass Module)



รูปที่ 2.24 อุปกรณ์วัดค่ามุมที่ใช้ในโครงการ

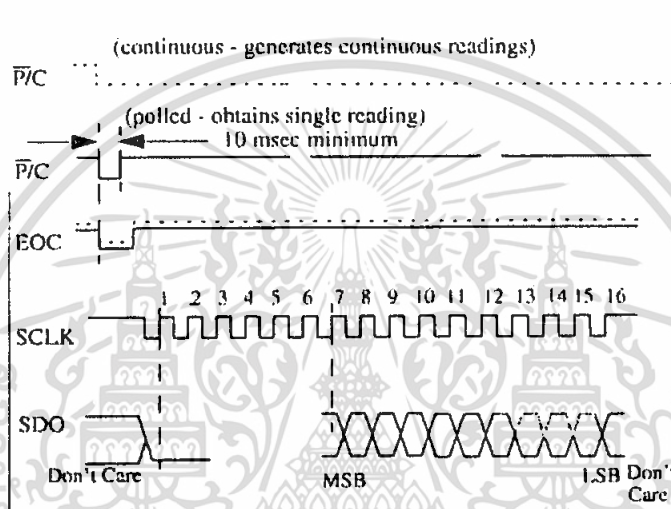
อุปกรณ์วัดค่ามุม เป็นอุปกรณ์ที่ใช้สำหรับวัดมุมหรือทิศทาง โดยสร้างสนามแม่เหล็กขึ้นแล้วเปรียบเทียบกับความเข้มของสนามแม่เหล็กกับสนามแม่เหล็กโลก ซึ่งค่าที่ได้จะเป็นค่ามุมที่วัดเทียบกับทิศเหนือ ตำแหน่งที่ทิศเหนือพอดีจะมีค่าเป็น 0 องศาและ ณ ตำแหน่งทิศเหนือหรือ 0 องศา เมื่อหมุนทวนเข็มนาฬิกาจะให้ค่าตั้งแต่ 0 องศา ถึง 359 องศา อุปกรณ์วัดค่ามุมที่นำมาใช้ในโครงการนี้เป็นรุ่น Vector 2X (V2X) ซึ่งผลิตโดย Precision Navigation, Inc.'s

สถานะการทำงานของอุปกรณ์วัดค่ามุมชนิดนี้ มี 2 สถานะหลัก ๆ คือ

- การทำงานในสถานะที่เป็นมาสเตอร์ (Master Mode)
- การทำงานสถานะที่เป็นสเลฟ (Slave Mode)

2.3.2.1 การทำงานในสถานะที่เป็นมาสเตอร์

ในสถานะนี้จะใช้อุปกรณ์วัดมุมเป็นอุปกรณ์หลักในการทำงาน การประยุกต์ใช้กับงานบางประเภทสำหรับการทำงานในสถานะนี้ไม่จำเป็นจะต้องใช้ตัวคอนโทรลเลอร์ในการควบคุมเพราะตัวอุปกรณ์วัดค่ามุมสามารถสร้างสัญญาณนาฬิกาออกมาได้ เมื่อตัวอุปกรณ์วัดค่ามุมคำนวณค่าเรียบร้อยแล้ว จะให้ค่าเอาต์พุตออกมา 2 สัญญาณ คือ สัญญาณของมุมที่วัดได้และสัญญาณนาฬิกาจำนวน 16 ลูก สำหรับในการติดต่อกับตัวอุปกรณ์วัดค่ามุมเพื่อขออ่านข้อมูลนั้น สามารถทำได้ทั้ง 2 แบบ คือการขออ่านค่าตลอดเวลา และการขออ่านค่าเพียงครั้งเดียว ตามไคอะแกรมเวลา (Timing diagram) ดังนี้



รูปที่ 2.25 ไคอะแกรมเวลาแสดงการทำงานในสถานะที่เป็นมาสเตอร์

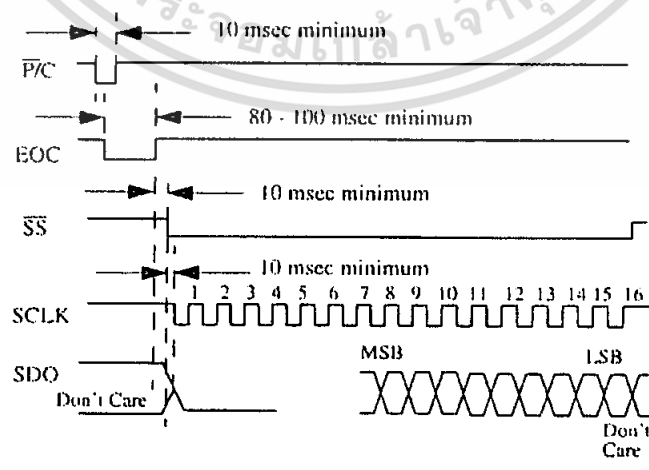
การตั้งค่าสำหรับการทำงานในสถานะที่เป็นมาสเตอร์

1. ต่อขา $\overline{M} / S$ เข้ากับ Low ส่วนขา $\overline{RAW}$ ต่อไว้ที่ High หรือจะไม่ต่อก็ได้
2. ต่อขา $\overline{SS}$ เข้ากับ High หรือจะไม่ต่อก็ได้
3. ขา $\overline{BCD} / \text{Bin}$ เป็นขาสำหรับเลือกรูปแบบของเอาต์พุต ต่อขานี้เข้ากับ High ถ้าต้องการให้เอาต์พุตเป็น Binary หรือต่อกับ Low ถ้าต้องการให้เป็น BCD
4. ขา $\overline{RES}$ เป็นขาที่ใช้สำหรับเลือกความละเอียด ต่อขานี้เข้ากับ High ถ้าต้องการความละเอียดสูง หรือต่อเข้ากับ Low ถ้าต้องการความละเอียดต่ำ
5. สำหรับรูปแบบของการติดต่อนั้นเลือกโดยการจ่ายพัลส์เข้าที่ขา $\overline{P} / C$ คือ ถ้าต้องการอ่านค่าเพียงครั้งเดียวให้จ่ายพัลส์ Low อย่างน้อย 10 msec เข้าที่ขานี้ (ตามที่แสดงในไคอะแกรมเวลาในรูป) แต่ถ้าต้องการอ่านค่าตลอดเวลาอย่างต่อเนื่องก็ให้คงสถานะ Low ไว้ตลอดเวลา
6. ในการคำนวณค่าเอาต์พุตนั้น จะเป็นดังนี้

- เมื่อขา $\bar{P}/C$ เป็น Low แล้ว EOC จะเปลี่ยนสถานะเป็น Low ด้วย โดยขณะนั้นตัวอุปกรณ์วัดค่ามุมจะคำนวณค่า
 - หลังจากที่ตัวอุปกรณ์วัดค่ามุมทำการคำนวณเสร็จเรียบร้อยแล้ว สถานะของ EOC จะเปลี่ยนเป็น High ทำให้ที่ขา SCLK ให้ค่าเอาต์พุตเป็นสัญญาณนาฬิกาจำนวน 16 ลูกที่มีความถี่ 4 KHz และที่ขา SDO จะให้ค่าเอาต์พุตเป็นค่ามุมที่วัดได้ตามรูปแบบของข้อมูลที่เลือกไว้แล้ว
7. สำหรับการติดต่อขออ่านข้อมูลจากตัวอุปกรณ์วัดค่ามุมแบบขออ่านเพียงครั้งเดียวนั้น หลังจากที่ตัวอุปกรณ์คำนวณเสร็จสิ้นแล้วก็จะเซตตัวเองเข้าสู่ Sleep mode จนกว่าจะมีการติดต่อขออ่านค่าครั้งต่อไป ส่วนการติดต่อแบบขออ่านค่าอย่างต่อเนื่องนั้น เมื่อเสร็จสิ้นกระบวนการตัวอุปกรณ์จะไม่เซตตัวเองเข้าสู่ Sleep mode แต่จะวนลูป (loop) ทำงานในข้อ 6 เพื่อคำนวณค่ามุมเมื่อมีการเปลี่ยนแปลงทิศทางของตัวอุปกรณ์ต่อไปตามไคอะแกรมเวลาในรูป ทำให้สามารถอ่านค่าได้ตลอดเวลา

2.3.2.2 การทำงานในสถานะที่เป็นสเลฟ (Slave mode)

การทำงานในสถานะนี้จะแตกต่างกับการทำงานในสถานะที่เป็นมาสเตอร์ตรงที่ตัวอุปกรณ์วัดค่ามุมจะไม่สร้างสัญญาณนาฬิกาเป็นเอาต์พุตออกมา แต่ในทางกลับกัน จะรับสัญญาณนาฬิกาจากภายนอกเข้ามา เพื่อเป็นตัวกำหนดการคลิก (Clock) ข้อมูลที่คำนวณได้ออกไป โดยสัญญาณนาฬิกาที่ส่งเข้าไปจะต้องมีจำนวน 16 ลูกสมบูรณ์และมีความถี่ไม่เกิน 1 MHz นั่นคือ ความถี่สูงสุดของการคลิกข้อมูลออกจะไม่เกิน 1 MHz ด้วย และการติดต่อขออ่านข้อมูลในการทำงานในสถานะที่เป็นสเลฟ สามารถติดต่อขออ่านข้อมูลได้เพียงแบบเดียวเท่านั้น คือ แบบการขออ่านค่าเพียงครั้งเดียวตามไคอะแกรมเวลา ดังนี้



รูปที่ 2.26 ไคอะแกรมเวลาแสดงการทำงานในสถานะที่เป็นสเลฟ

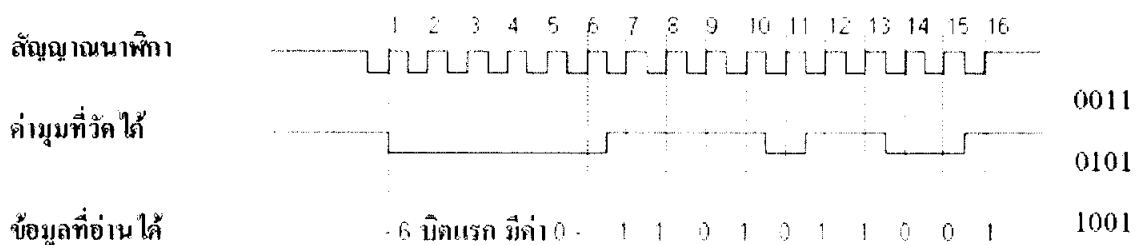
การตั้งค่าสำหรับการทำงานในสถานะที่เป็นสเลฟ

1. ต่อขา $\overline{M}/S$ เข้ากับ High ส่วนขา $\overline{RAW}$ ต่อไว้กับ High หรือไม่ต่อก็ได้
2. ขา $\overline{BCD}$ / Bin เป็นขาสำหรับเลือกรูปแบบของเอาต์พุต ต่อขานี้เข้ากับ High ถ้าต้องการให้เอาต์พุตเป็น Binary หรือต่อกับ Low ถ้าต้องการให้เป็น BCD
3. ขา $\overline{RES}$ เป็นขาที่ใช้สำหรับเลือกความละเอียด ต่อขานี้เข้ากับ High ถ้าต้องการความละเอียดสูง หรือต่อเข้ากับ Low ถ้าต้องการความละเอียดต่ำ
4. จ่ายพัลส์ Low เข้าที่ขา $\overline{P}/C$ ให้ความยาวพัลส์ไม่ต่ำกว่า 10 msec เพื่อขอติดต่อกับอุปกรณ์เพื่ออ่านค่า สำหรับความยาวพัลส์นั้นจะไม่จำกัดค่ามากที่สุดว่าจะต้องยาวเท่าใด แต่มีข้อแม้เพียงว่าขา $\overline{P}/C$ จะต้องมิสสถานะกลับเป็น High ก่อนที่จะให้ขา $\overline{SS}$ มีสถานะเป็น Low เพื่อคล็อกข้อมูลออก ส่วนขา $\overline{SS}$ นั้นจะเริ่มจ่ายพัลส์ Low ได้หลังจากที่ $\overline{EOC}$ มีสถานะกลับเป็น High แล้วอย่างน้อยที่สุด 10 msec
5. อุปกรณ์วัดมุมจะทำการคำนวณในขณะที่ขา $\overline{EOC}$ มีสถานะเป็น Low และเมื่อขา $\overline{EOC}$ มีสถานะกลับเป็น High ข้อมูลที่ได้จากการคำนวณก็พร้อมที่จะเป็นค่าเอาต์พุตโดย
 - หลังจากที่ทำให้ขา $\overline{P}/C$ เป็น Low แล้ว $\overline{EOC}$ จะมีสถานะเป็น Low ตามด้วย
 - หลังจากที่ได้วัดอุปกรณ์ทำการคำนวณค่าเพื่อเป็นเอาต์พุตเสร็จแล้ว $\overline{EOC}$ ก็จะมีสถานะกลับเป็น High แต่มีข้อจำกัดว่า $\overline{P}/C$ จะต้องมิสสถานะกลับเป็น High ก่อนที่ $\overline{EOC}$ จะมีสถานะกลับเป็น High
6. หลังจาก that $\overline{EOC}$ มีสถานะกลับเป็น High แล้ว ต้องรอน้อย 10 msec จึงจะทำให้ $\overline{SS}$ เป็น Low ได้ แต่ $\overline{SS}$ จะต้องเป็น Low ขณะที่มีการคล็อกข้อมูลออกไป
7. ที่ SCLK จะต้องได้รับสัญญาณนาฬิกา 16 ลูกอย่างสมบูรณ์ และสัญญาณนั้นจะต้องมีความถี่ไม่เกิน 1 MHz ด้วย โดยกระบวนการช่วงนี้คือ
 - ก่อนที่จะทำให้ SCLK เป็น Low ที่ขา $\overline{SS}$ จะต้องถูกทำให้เป็น Low ก่อนช่วงเวลาหนึ่ง ซึ่งความยาวของช่วงเวลาจะขึ้นอยู่กับความซับซ้อนของระบบ
 - ในการอ่านข้อมูลนั้น จะอ่านข้อมูลบิตแรกจาก SDO ได้เมื่อสัญญาณนาฬิกาเปลี่ยนจาก Low เป็น High (ในช่วงขอบขาขึ้น)
 - และเช่นกันคือจะต้องจ่ายสัญญาณนาฬิกา 6 ลูกสมบูรณ์ให้แก่ SCLK เพื่อที่จะอ่านข้อมูลใน 6 บิตถัดไป
 - เมื่ออ่านข้อมูลได้ครบ 6 บิตแรกแล้วจะต้องหยุดรอเวลาช่วงหนึ่งแล้วจึงจ่ายสัญญาณนาฬิกาอีก 8 ลูกสมบูรณ์เป็นอย่างน้อย เพื่ออ่านข้อมูลอีก 8 บิตที่เหลือต่อไป ซึ่งการอ่านข้อมูลนั้นก็เหมือนๆกับใน 6 บิตแรกคือ จะอ่านที่ขอบขาขึ้นของสัญญาณนาฬิกา ซึ่งเมื่ออ่านข้อมูลตัวสุดท้ายเสร็จแล้ว SCLK ก็จะเปลี่ยนสถานะเป็น High และคงค่าไว้ตลอดจนกว่าจะมีการอ่านข้อมูลครั้งต่อไป

8. สำหรับข้อมูลในบิตสุดท้ายจะถูกคัดลอกออกไปภายในเวลา 20 μsec นับตั้งแต่ขอบขาขึ้นของสัญญาณนาฬิกาสุดท้าย สำหรับ 15 บิตก่อนหน้านี้สามารถอ่านค่าได้เลยไม่ต้องรอเวลาเหมือนในลูกสุดท้าย แต่อาจจะมีการคีย์บ้างในระหว่างบิตต่อบิต
9. หลังจากที่ข้อมูลถูกคัดลอกออกไปแล้ว และทำให้ $\overline{SS}$ เป็น High อีกครั้งหนึ่ง แล้วตัวอุปกรณ์วัดค่ามุมจะเซตตัวเองให้กลับไปสู่ Sleep mode
10. สำหรับขา $\overline{P}/C$ และขา $\overline{SS}$ สามารถต่อไว้ด้วยกันซึ่งการทำงานก็จะเป็นไปตามที่ได้กล่าวมาแล้วข้างต้นทำให้สามารถลดความยุ่งยากในการควบคุมได้ แต่ถ้าเป็นในกรณีที่ไม่ได้ต่อขาทั้งสองไว้ด้วยกันแล้ว เมื่อมีการจ่ายพัลส์ Low ให้แก่ $\overline{P}/C$ เพื่อแสดงการขอติดต่อกับอุปกรณ์วัดค่ามุมแล้ว $\overline{EOC}$ จะเป็น Low ซึ่งความยาวของพัลส์ Low ที่จ่ายให้แก่ $\overline{P}/C$ นั้น จะต้องเป็น Low อยู่จนกระทั่ง $\overline{EOC}$ เป็น Low แล้ว จึงจะให้ $\overline{P}/C$ กลับเป็น High ได้ และเมื่อ $\overline{EOC}$ เป็น Low แล้ว ก็จะคงค่า Low นี้อยู่ 80 - 100 msec จากนั้น $\overline{EOC}$ จะกลับเป็น High อีกครั้ง หลังจากนั้นให้ทิ้งช่วงเวลาประมาณ 10 msec แล้วจึงทำให้ขา $\overline{SS}$ เป็น Low จนกระทั่งคัดลอกข้อมูลบิตสุดท้ายออกไปเรียบร้อยแล้ว จึงทำให้ขา $\overline{SS}$ กลับเป็น High เพื่อรอการขอติดต่อกครั้งต่อไป

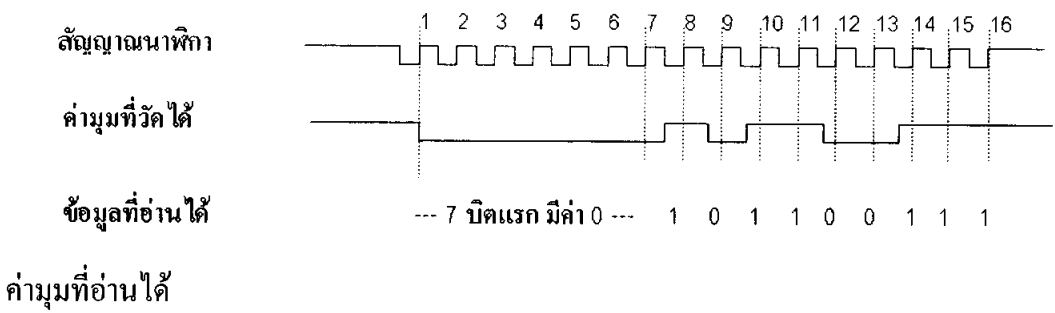
2.3.2.3 รูปแบบของข้อมูล

สำหรับรูปแบบของข้อมูลที่ออกมานั้นสามารถเลือกรูปแบบได้ คือ เป็นได้ทั้งแบบ บีซีดี (BCD: Binary Code Decimal) และ แบบไบนารีหรือเลขฐานสอง (Binary) ในการอ่านข้อมูลนั้นจะต้องอ่านเทียบกับสัญญาณนาฬิกาที่ตัวอุปกรณ์วัดค่ามุมจ่ายออกมาจากการทำงานในสถานะที่เป็นมาสเตอร์ หรือสัญญาณนาฬิกาที่จ่ายให้แก่อุปกรณ์วัดค่ามุมจากการทำงานในสถานะที่เป็นสเลฟ โดยจะอ่านทุก ๆ ขอบขาลงของสัญญาณนาฬิกา สำหรับแบบบีซีดี ข้อมูลใน 6 บิตแรกจะมีค่าเป็น 0 และจะเริ่มอ่านข้อมูลที่เป็นค่ามุมตั้งแต่บิตที่ 7 ไปแล้ว ส่วนแบบไบนารี ข้อมูลใน 7 บิตแรกจะมีค่าเป็น 0 และจะเริ่มอ่านข้อมูลที่เป็นค่ามุมตั้งแต่บิตที่ 8 เป็นต้นไป รูปที่ 2.27 แสดงตัวอย่างการอ่านข้อมูลแบบบีซีดีเทียบกับสัญญาณนาฬิกา ณ ค่ามุม 359 องศา และรูปที่ 2.28 แสดงตัวอย่างการอ่านข้อมูลแบบไบนารีเทียบกับสัญญาณนาฬิกา ณ ค่ามุม 359 องศา



$$\text{ค่ามุมที่อ่านได้} = (3^\circ \times 100) + (5^\circ \times 10) + (9^\circ \times 1) = 359^\circ$$

รูปที่ 2.27 การอ่านข้อมูลแบบบีซีดีเทียบกับสัญญาณนาฬิกา



รูปที่ 2.28 การอ่านข้อมูลแบบไบนารีเทียบกับสัญญาณนาฬิกา

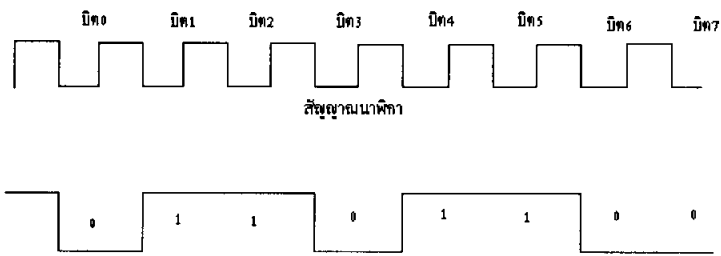
จากรูปที่ 2.27 ค่ามุมที่วัดได้ตัวแรกนั้นจะต้องอ่านเทียบกับลูกที่ 7 ของสัญญาณนาฬิกา เนื่องจากการเลือกรูปแบบของข้อมูล ดังนั้นในบิตแรกของข้อมูลที่เป็นค่ามุมจึงมีค่าเป็น 1 และเมื่ออ่านค่ามุมเทียบกับสัญญาณนาฬิกาในบิตต่อ ๆ ไปก็จะได้ข้อมูลทั้งหมดเป็น “1111011001” ซึ่งสามารถแปลค่าได้เป็นมุม 359 องศา ส่วนในรูปที่ 2.28 นั้น เมื่อเลือกรูปแบบของข้อมูลเป็นแบบไบนารีแล้ว ข้อมูลที่จะอ่านในบิตแรกจะต้องอ่านเทียบกับสัญญาณนาฬิกาที่ 8 ซึ่งจากรูปจะมีค่าเป็น 1 และเมื่ออ่านค่ามุมเทียบกับสัญญาณนาฬิกาในบิตต่อ ๆ ไปก็จะได้ข้อมูลเป็น “101100111” ซึ่งสามารถแปลค่าได้เป็นมุม 359 องศา

2.4 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกโดยผ่านพอร์ตอนุกรม

การสื่อสารแบบอนุกรมนั้นแบ่งออกได้เป็น 2 แบบ คือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารแบบอะซิงโครนัส

2.4.1 การสื่อสารข้อมูลแบบซิงโครนัส

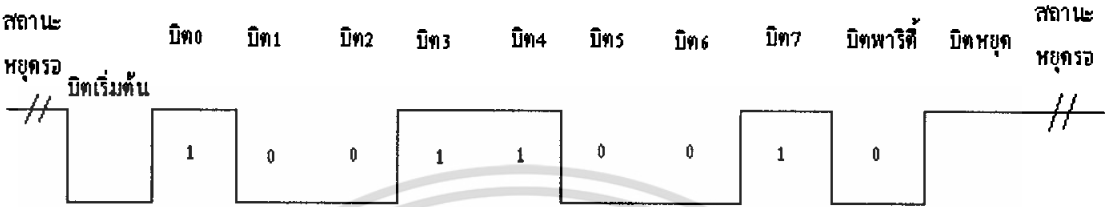
การสื่อสารข้อมูลแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณ ด้วย การติดต่อกันแบบนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูล และกราวด์ ข้อดีของการส่งแบบนี้ สามารถควบคุมให้เครื่องส่งและเครื่องรับทำงานได้พร้อมกัน อย่างถูกต้อง และทางเครื่องรับสามารถแปลรหัสข่าวสารได้ถูกต้องตรงตามด้านส่ง การส่งสัญญาณแบบนี้มีใช้ทั่วไปในเครื่องรับส่งที่มีความเร็วสูง แสดงดังรูปที่ 2.29



รูปที่ 2.29 การส่งข้อมูลแบบซิงโครนัส

2.4.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อม แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งค่ารับและค่าส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกานี้ว่า อัตราการถ่ายข้อมูล หรือ บอกระต (Board Rate) มีหน่วยเป็น บิตต่อวินาที (bit per second: bps) แสดงดังรูปที่ 2.30

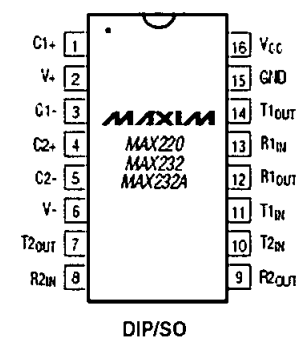


รูปที่ 2.30 การส่งข้อมูลแบบอะซิงโครนัส

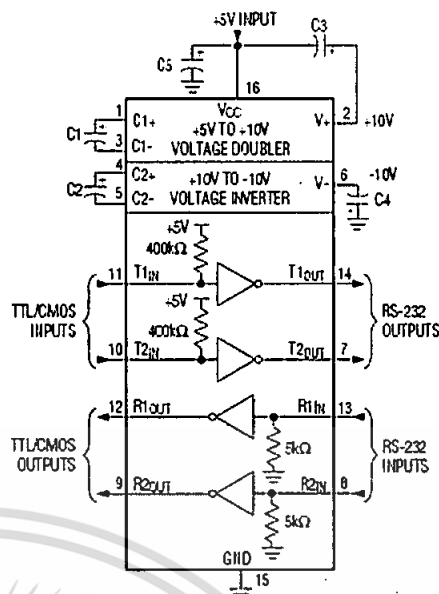
ในโครงงานนี้จะติดต่อกับระบบคอมพิวเตอร์ โดยผ่านพอร์ตอนุกรมแบบ RS232 ในที่นี้จึงขอกล่าวการสื่อสารข้อมูลแบบอะซิงโครนัส เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการสื่อสารข้อมูลอนุกรมอะซิงโครนัส 2 ทิศทาง โดยได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ DTE ต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

ขาสัญญาณเอาต์พุตของพอร์ตอนุกรม ได้แก่ ขา DTR, RTS และ TxD โดยรีจิสเตอร์ที่ทำหน้าที่ควบคุมขาเหล่านี้คือ รีจิสเตอร์ควบคุมโมเด็ม (MCR) โดยมีแอดเดรสอยู่ถึรีจิสเตอร์หลักของพอร์ตอนุกรม 4 ตำแหน่ง รีจิสเตอร์ควบคุมการทำงานของขา DTR จะอยู่ที่ตำแหน่งบิต 0 ส่วนขา RTS จะอยู่ที่ตำแหน่งบิต 1 ส่วนขา TxD เป็นขาปกติในการส่งข้อมูล ดังนั้น จึงมีแอดเดรสอยู่ถึแอดเดรสรีจิสเตอร์หลักระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS232 ได้ระบุช่วงระดับแรงดัน สำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก “0” จะมีระดับสัญญาณ +3V ถึง +15V ส่วนลอจิก “1” จะมีระดับสัญญาณ -3V ถึง -15V ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำขาเอาต์พุตใดๆต่อเข้ากับลอจิกเกตใช้งานได้โดยตรง จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน โดยปกติจะใช้ไอซีจำพวก RS232 ซึ่งจะทำหน้าที่แปลงระดับแรงดันของ RS232 ให้อยู่ในระดับที่ที่แอล โดยลอจิก “0” ซึ่งเดิมมีระดับสัญญาณ +3V ถึง +15V จะถูกแปลงเป็น 0V ส่วนลอจิก “1” ซึ่งมีระดับสัญญาณ -3V ถึง -15V จะเปลี่ยนเป็น +5V ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิทัลอื่นที่ใช้ระดับแรงดันที่ที่แอลได้แสดงดังรูปที่ 2.31

TOP VIEW



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



รูปที่ 2.31 ไอซี MAX232

2.5 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปแบบสัญญาณดิจิทัลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิทัลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้

ภายในไมโครคอนโทรลเลอร์จะมีหน่วยประมวลผล หน่วยความจำ Port อยู่ในชิพเพียงตัวเดียวซึ่งอาจจะเรียกได้ว่าเป็นคอมพิวเตอร์ชิพเดี่ยว ไมโครคอนโทรลเลอร์เป็นไมโครโปรเซสเซอร์ชนิดหนึ่ง เช่นเดียวกับหน่วยประมวลผลกลาง (CPU: Central Processing Unit) ที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนาแยกออกมาภายหลังเพื่อนำไปใช้ในวงจรทางด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องวงจรภายนอกต่าง ๆ เพิ่มเติมเช่นเดียวกับไมโครโปรเซสเซอร์ ก็จะรวมวงจรที่จำเป็น เช่น หน่วยความจำ ส่วนอินพุต/เอาต์พุต บางส่วนเข้าไปในตัว ไอซีเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปเพื่อให้มีความสามารถและเหมาะสมกับการใช้งานควบคุม เช่น วงจรตั้งเวลา วงจรการสื่อสารอนุกรม วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล เป็นต้น สรุปคือ

$$\text{Microcontroller} = \text{Microprocessor} + \text{Memory} + \text{I/O}$$

ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานอย่างกว้างขวาง โดยมักจะเป็นการนำไปใช้ฝังในระบบของอุปกรณ์อื่น ๆ เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในรถยนต์ เต้าอบ ไมโครเวฟ เครื่องปรับอากาศ เครื่องซักผ้าอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้งานควบคุมหลายประการ เช่น

- ชิพไอซีและระบบที่ได้มีขนาดเล็ก
- ระบบควบคุมที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์
- วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
- มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย
- ช่วยลดระยะเวลาในการพัฒนาระบบได้

ไมโครคอนโทรลเลอร์มีหลายยี่ห้อ หลายตระกูล และหลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์จะมีโครงสร้างภายในและความสามารถในการทำงานที่แตกต่างกัน ทำให้สามารถเลือกใช้กับงานได้อย่างเหมาะสม

คุณสมบัติทั่วไปของ IC 16F877

- 35 Instruction คำสั่ง
- ปฏิบัติคำสั่งทั่วไปใน Cycle เดียว และ 2 Cycle ในคำสั่งการกระโดด
- ทำงานที่ความถี่สูงสุด 20 MHz (16F877-20/P)
- หน่วยความจำโปรแกรม 8 K(14 Bit Words)
- หน่วยความจำข้อมูล (RAM) 368 Byte
- EEPROM 256 Byte
- สามารถตอบสนองอินเตอร์รัพท์ 14 แหล่ง
- STACK 8 ระดับ
- สามารถเลือกโหมดของสัญญาณนาฬิกาได้
- กระแสซิงค์ และ ซอร์สของพอร์ต คือ 25 mA
- Timer/Counter 3 ตัว Timer0, Timer1 และ Timer2
- โมดูล Capture/Compare/PWM จำนวน 2 ชุด
- Analog to Digital Converter 10Bit 8 Channel
- พอร์ต I/O จำนวน 5 พอร์ต A, B, C, D และ E รวมทั้งสิ้น 33 บิต

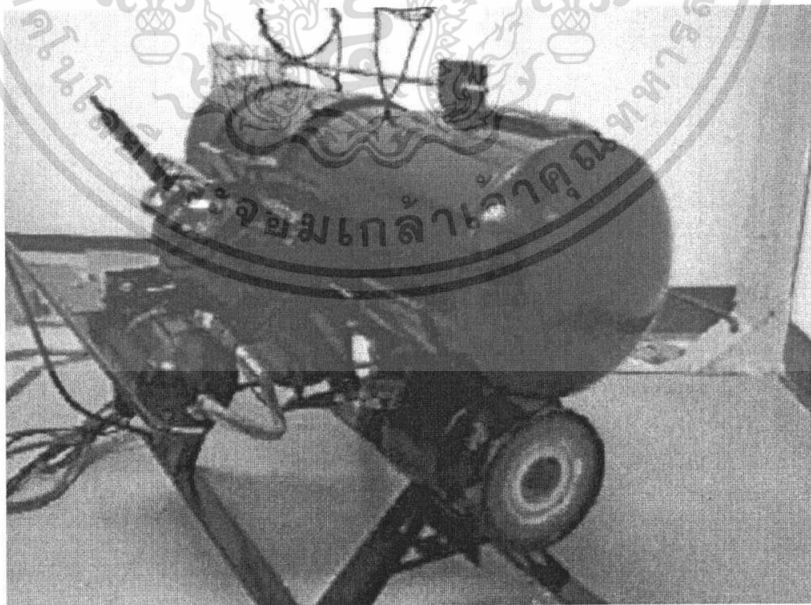
บทที่ 3

การสร้างและออกแบบ

ในบทนี้จะกล่าวถึง การออกแบบยานสำรวจและวงจรต่างๆที่ใช้ในการควบคุมยานสำรวจได้นำ หลักการทำงานของวงจรและการเชื่อมต่อวงจรที่ใช้ในยานสำรวจ

3.1 การออกแบบส่วนประกอบต่างๆของยานสำรวจได้นำ

ยานสำรวจที่ใช้ในโครงการจะถูกออกแบบโดยผู้ที่ทำยานสำรวจในรุ่นที่ 4 จะมีลักษณะเป็นโครงสร้างทรงกระบอก ด้วยยานสำรวจทำด้วยท่อ PVC ขนาดเส้นผ่านศูนย์กลาง 6 นิ้ว โดยใช้ฝาครอบปิดหัวท้าย ภายในท่อมีพื้นที่สำหรับใส่อุปกรณ์และวงจรควบคุมต่างๆได้ง่ายและสะดวก สาเหตุที่เลือกใช้ท่อ PVC เนื่องจากท่อ PVC สามารถดัดขึ้นรูปได้ตามต้องการโดยใช้ความร้อน ดัดต่อได้ง่าย มีความคงทนแข็งแรงและหาซื้อได้ง่ายตามท้องตลาด และในรุ่นที่ 5 นี้ได้ออกแบบส่วนป้องกันตัวเรือซึ่งทำจากเหล็ก เนื่องจากมีความแข็งแรง ทนทานต่อการกระแทกทำให้สามารถป้องกันส่วนของใบพัดที่จะกระแทกกับพื้นได้ แต่มีข้อเสียคือน้ำหนักมากจึงต้องใช้หุ่นช่วยพยุงตัวเรือให้ลอยและได้มีการพันสีที่ตัวเรือเพื่อการป้องกันสัญญาณรบกวนจากมอเตอร์และยังทำให้ดูสวยงามอีกด้วย



รูปที่ 3.1 ยานสำรวจ

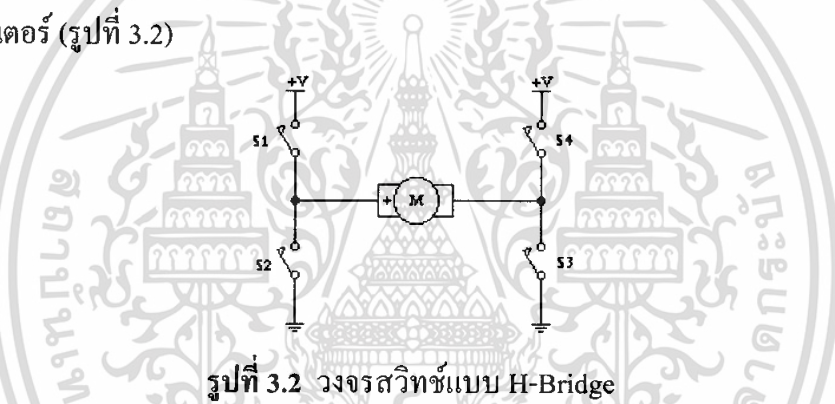
3.2 วงจรต่างๆ

ในส่วนของวงจรที่ติดตั้งอยู่บนยานสำรวจนั้นถูกออกแบบโดยผู้ที่ทำยานสำรวจรุ่นก่อน และในรุ่นนี้จะทำการพัฒนาส่วนของ PCB ซึ่งจะออกแบบให้มีความแน่นหนาทนทานมากขึ้นและยังสามารถป้องกันสัญญาณรบกวนจากมอเตอร์ได้ด้วยซึ่งจะประกอบด้วยวงจรขับเคลื่อนมอเตอร์ วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ วงจรรวมชุดเซนเซอร์ วงจรขับสัญญาณ PWM และวงจรแหล่งจ่ายไฟฟ้า

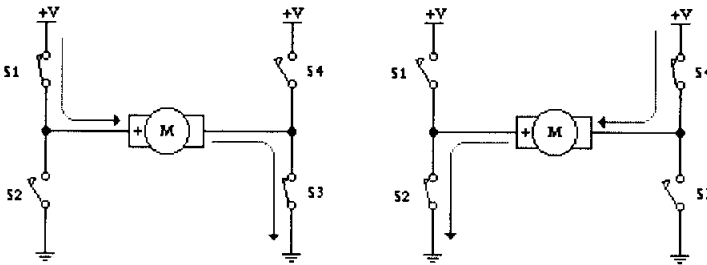
3.2.1 วงจรขับเคลื่อนมอเตอร์

ในการควบคุมทิศทางการหมุนของมอเตอร์ดีซี ในโครงงานนี้อาศัยหลักการทำงานของ วงจร H-Bridge Switching เป็นตัวควบคุมทิศทางการหมุนของมอเตอร์ซึ่งจะประกอบด้วย สวิตช์ 4 ตัว ก็คือ S1, S2, S3 และ S4 โดยใช้มอเตอร์ดีซีเป็นภาระของวงจรดังรูปที่ 3.2

ในสถานะเริ่มต้น สวิตช์ทุกตัว Off อยู่ ก็จะไม่มีการเกิดขึ้นทั้งสิ้น เพราะไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์ (รูปที่ 3.2)

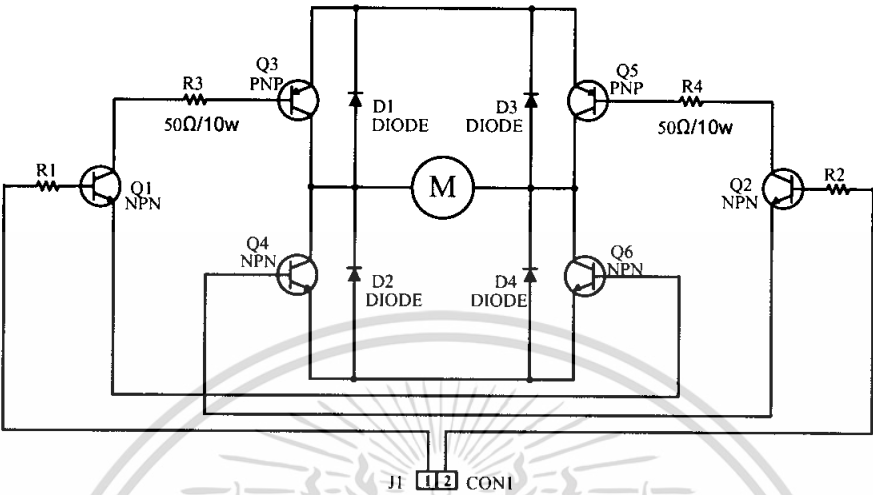


เมื่อสวิตช์ S1 และ S3 ทำงานพร้อมกัน จะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้าไหลผ่านมอเตอร์จากขั้วบวกของมอเตอร์ไปยังขั้วลบของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกานั้นขึ้นอยู่กับลักษณะของการพันขดลวดภายในมอเตอร์ ในทางกลับกัน ถ้าหากสวิตช์ S2 และ S4 ทำงานพร้อมกัน ก็จะเป็นการเชื่อมวงจร และทำให้เกิดกระแสไฟฟ้าไหลผ่านมอเตอร์จากขั้วลบของมอเตอร์ไปยังขั้วบวกของมอเตอร์ ทำให้มอเตอร์สามารถหมุนได้และเป็นการหมุนในทิศทางหมุนกลับกันทิศทางแรก ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 วงจรสวิตช์แบบ H-Bridge เมื่อ On สวิตช์ทั้ง 4 ตัว ทำให้มอเตอร์หมุนกลับทาง

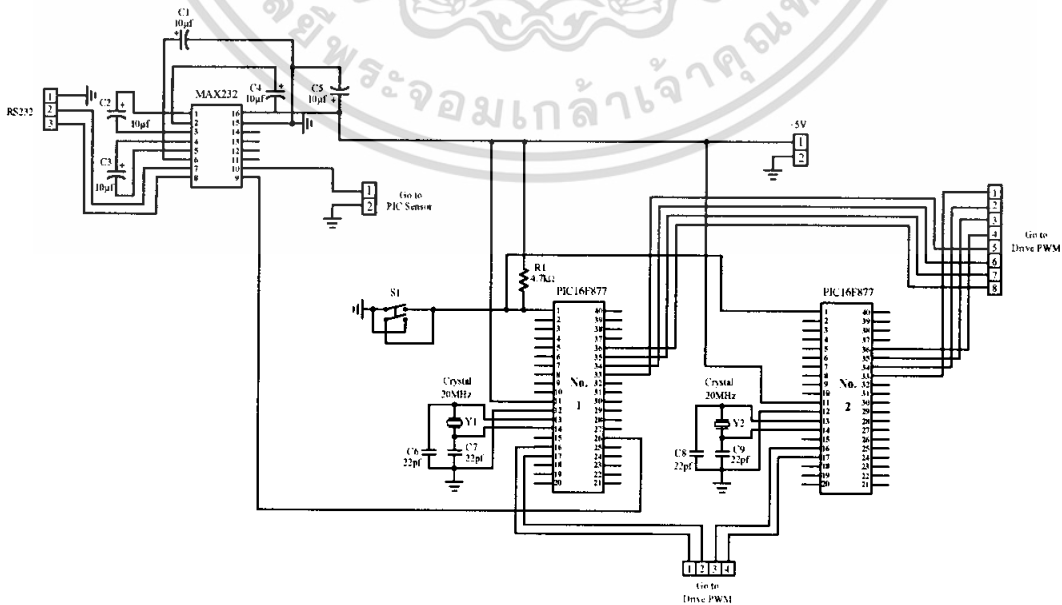
สรุปว่า วงจรนี้จะใช้สวิทซ์ 4 ตัว เพื่อบังคับทิศทางการไหลของกระแสไฟฟ้าที่ไหลผ่านมอเตอร์ เพื่อควบคุมให้มอเตอร์หมุนตามทิศทางที่ต้องการ โดยผลัดกันทำงานของสวิทซ์ทีละ 2 ตัว พร้อมกันนั่นเอง



รูปที่ 3.4 วงจรขับมอเตอร์

3.2.2 วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

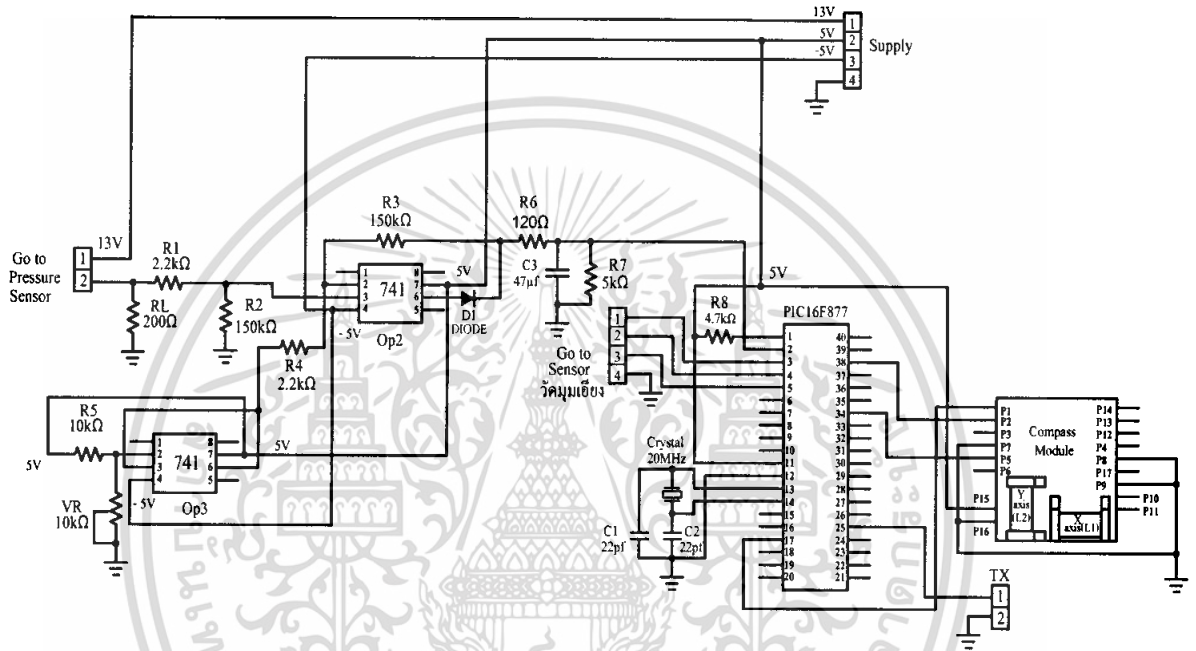
หลักการทำงานของวงจรเริ่มต้นจาก PIC1 และ PIC2 ทำหน้าที่เป็นตัวรับข้อมูลการควบคุมยานสำรวจจากคอมพิวเตอร์ และทำหน้าที่ส่งสัญญาณ PWM ไปยังวงจรขับสัญญาณ PWM เพื่อควบคุมทิศทางของการหมุนและความเร็วของมอเตอร์ต่อไปคอมพิวเตอร์จะรับข้อมูลจากวงจรรวมชุดเซนเซอร์และส่งสัญญาณมาที่ไมโครคอนโทรลเลอร์ทั้ง 2 ตัว โดยผ่านไอซี MAX 232 ซึ่งจะทำหน้าที่เชื่อมต่อข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

3.2.3 วงจรรวมชุดเซนเซอร์

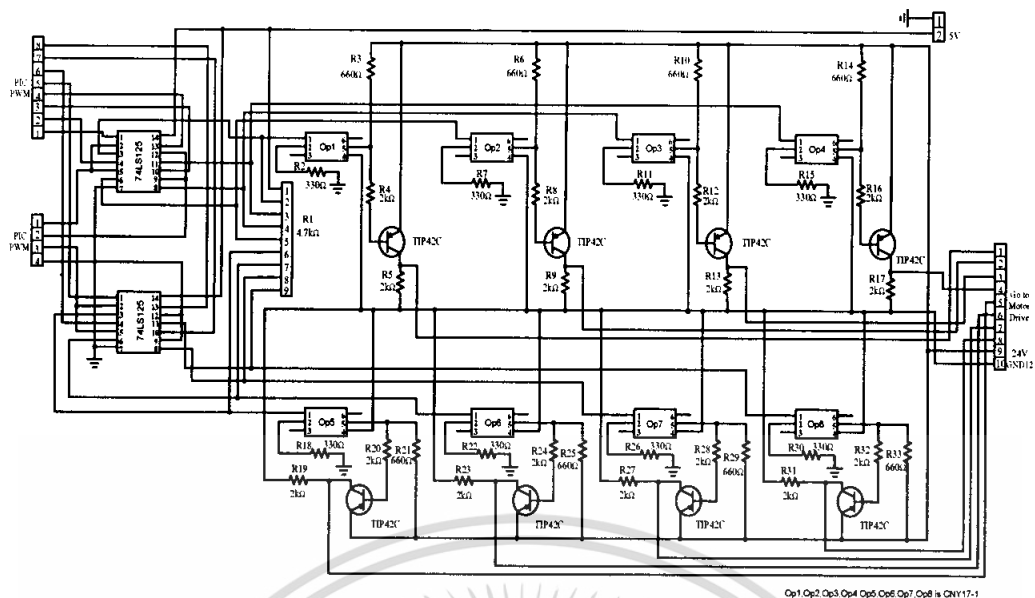
จากรูปที่ 3.6 จะประกอบด้วยเซนเซอร์สองชุด ได้แก่ เซนเซอร์วัดค่ามุม และเซนเซอร์วัดความดัน วงจรนี้จะทำหน้าที่อ่านค่าจากเซนเซอร์ทั้ง 2 ตัวแล้วส่งค่าไปยังวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์เพื่อแสดงค่าที่เซนเซอร์ทั้ง 2 ตัวส่งมา ซึ่งค่าที่เซนเซอร์วัดมุมส่งมาจะเป็นค่าปัจจุบันที่ส่วนหัว (Heading) ของเรือหันไป และเซนเซอร์วัดความดันก็จะแสดงค่าความลึกปัจจุบันที่ยานกำลังสำรวจอยู่เทียบกับจุดอ้างอิง ซึ่งในที่นี้คือระดับผิวน้ำ



รูปที่ 3.6 วงจรรวมชุดเซนเซอร์

3.2.4 วงจรขับสัญญาณ PWM

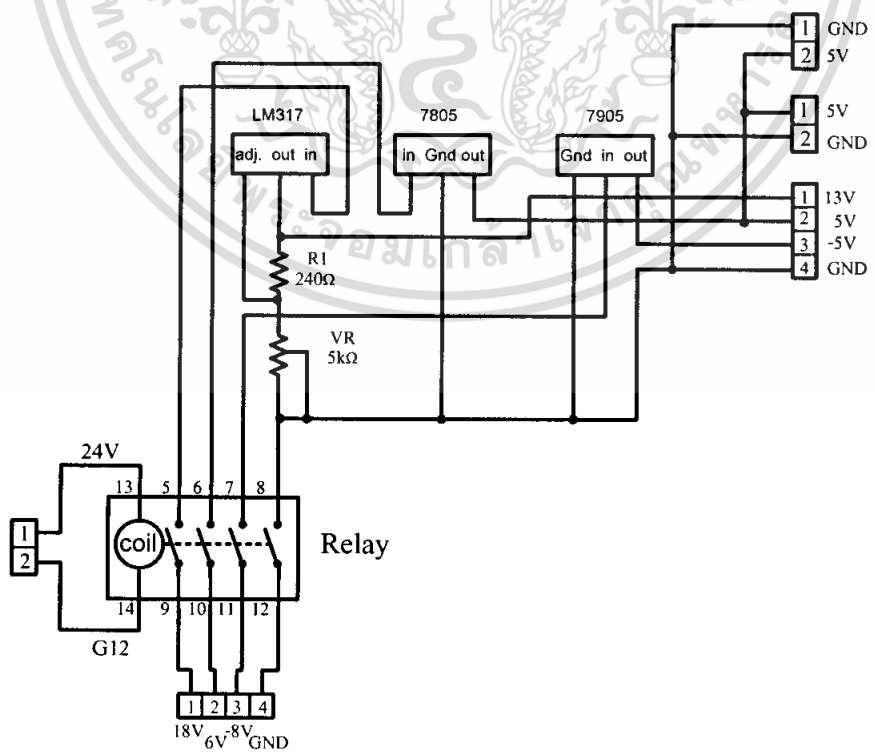
วงจรในรูปที่ 3.7 ทำหน้าที่ขยายสัญญาณที่รับมาจากวงจรรูปที่ 3.5 และทำหน้าที่แยกกราวด์ ระหว่างไมโครคอนโทรลเลอร์กับแหล่งจ่ายไฟให้กับมอเตอร์ เพื่อป้องกันสัญญาณรบกวนจากมอเตอร์ สัญญาณที่ขยายนั้นจะถูกส่งไปยังวงจรขับมอเตอร์เพื่อเป็นสัญญาณไปควบคุมการหมุนและความเร็วของมอเตอร์ ในวงจรนี้จะมีการแยกกราวด์ออกเป็น 2 กราวด์คือกราวด์ของไฟ 24 โวลต์ และกราวด์สัญญาณ(Signal Ground) เพื่อเป็นการป้องกันสัญญาณรบกวนอีกทางหนึ่ง



รูปที่ 3.7 วงจรขับสัญญาณ PWM

3.2.5 วงจรแหล่งจ่ายไฟฟ้า

ภายในยานสำรวจได้นำจะมีแผงวงจรที่ใช้จ่ายพลังงานเลี้ยงวงจรภายในยานสำรวจทั้งหมด และจะใช้รีเลย์เป็นตัวตัดต่อพลังงานเพื่อให้ประหยัดพลังงาน ซึ่งขดลวดของรีเลย์จะต่อเข้ากับแหล่งจ่ายพลังงานภายนอก และแหล่งจ่ายพลังงานภายนอกนี้จะจ่ายพลังงานให้กับมอเตอร์ด้วย แหล่งจ่ายพลังงานภายในตัวเรือดำนํ้าจะให้แรงดันไฟฟ้าขนาด 13 โวลต์ เพื่อจ่ายให้กับเซนเซอร์วัดความดัน แรงดันไฟฟ้าขนาด +5 โวลต์ และ -5 โวลต์ เพื่อไปเลี้ยงอุปกรณ์ต่างๆในแผงวงจร

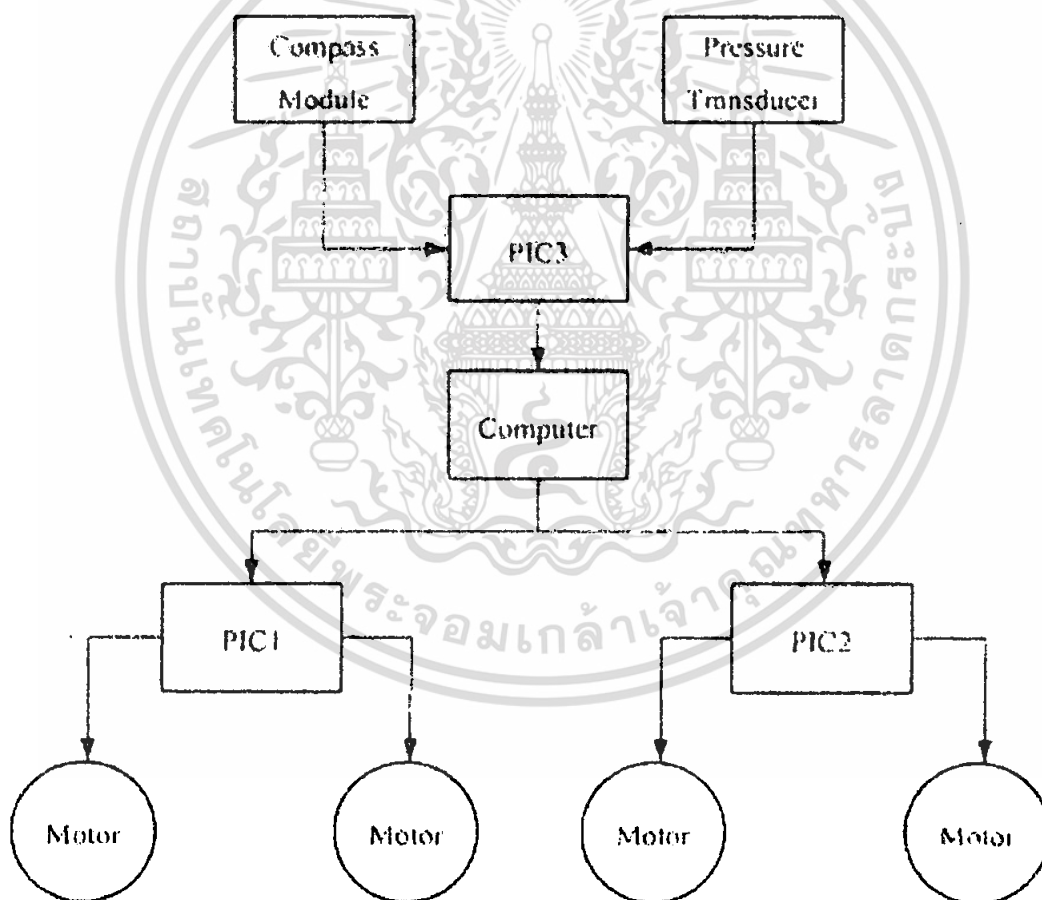


รูปที่ 3.8 วงจรแหล่งจ่ายไฟฟ้า

3.3 การออกแบบระบบควบคุม

ในการออกแบบระบบควบคุมเรือดำน้ำ จะทำโดยใช้คอนโทรลเลอร์ทั้งหมด 3 ตัว ดังแสดงในรูปที่ 3.9 โดย PIC3 จะรับข้อมูลจากเซนเซอร์ทั้ง 2 ชุด คือค่ามุมและค่าความลึกแล้วจึงนำข้อมูลส่งให้กับคอมพิวเตอร์ เพื่อประมวลผลโดยอาศัยตัวควบคุมแบบพีไอดี หรือแบบฟuzzyลอจิกและสร้างสัญญาณควบคุมออกมา แล้วส่งต่อไปให้กับ PIC1 และ PIC2 เพื่อใช้ในการสร้างสัญญาณพัลส์จ่ายให้มอเตอร์ต่อไป

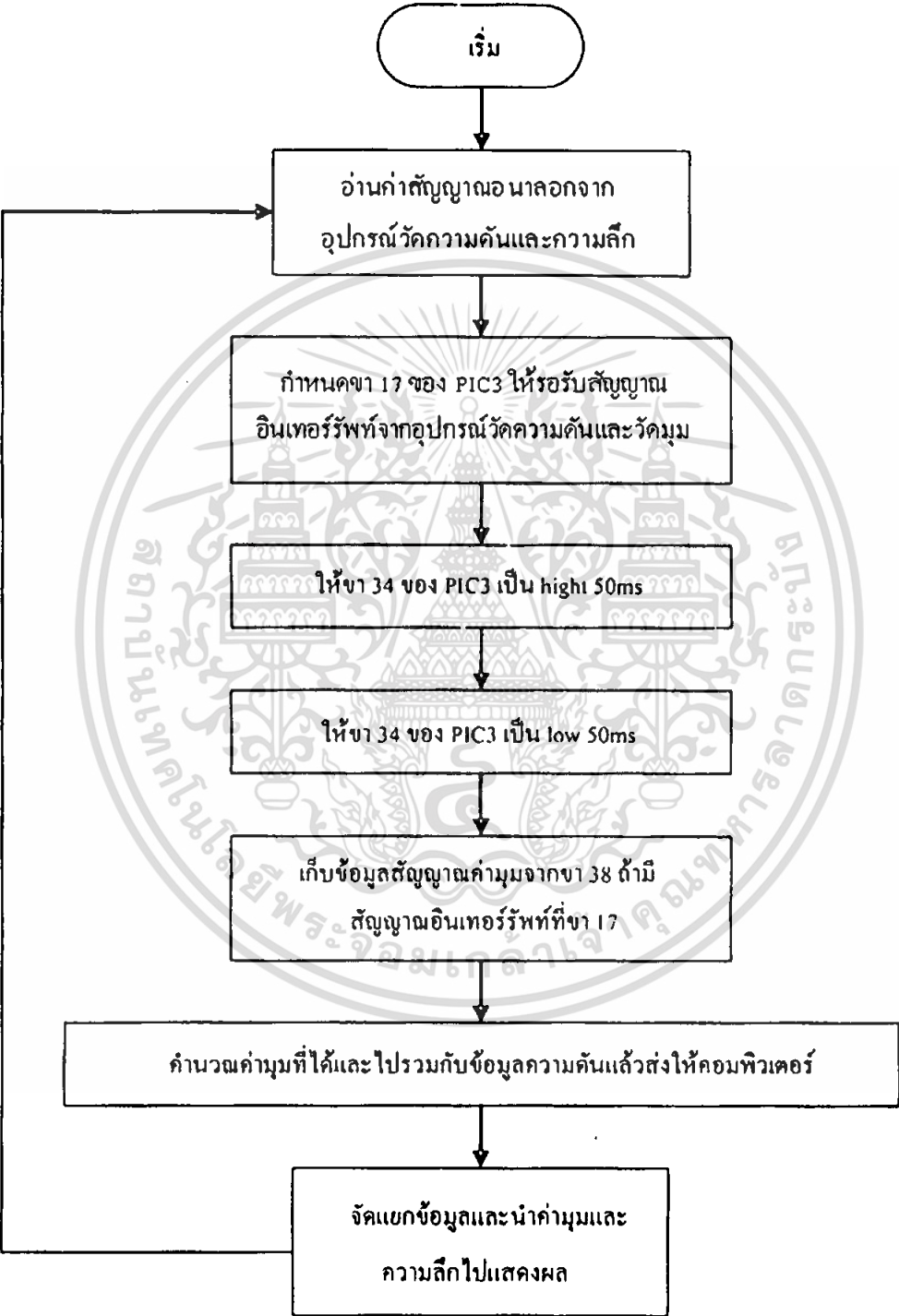
เนื่องจากคอนโทรลเลอร์ PIC หนึ่งตัว สามารถจ่ายสัญญาณพัลส์ได้เพียง 2 ช่องเท่านั้น ดังนั้น จึงจำเป็นต้องใช้ตัวคอนโทรลเลอร์ 2 ตัว เพื่อจ่ายสัญญาณพัลส์ 4 ช่องให้กับมอเตอร์ทั้ง 4 ตัว ซึ่งต้องใช้สัญญาณ PWM ทั้งหมด 4 ช่อง โดยในส่วนของโปรแกรมที่ออกแบบให้กับตัวคอนโทรลเลอร์ทั้ง 2 ตัวได้แสดงรายละเอียดไว้แล้วในภาคผนวก



รูปที่ 3.9 แผนผังแสดงการออกแบบระบบควบคุมยานสำรวจใต้น้ำ

3.4 แผนภาพกระบวนการรับค่าจากเซนเซอร์

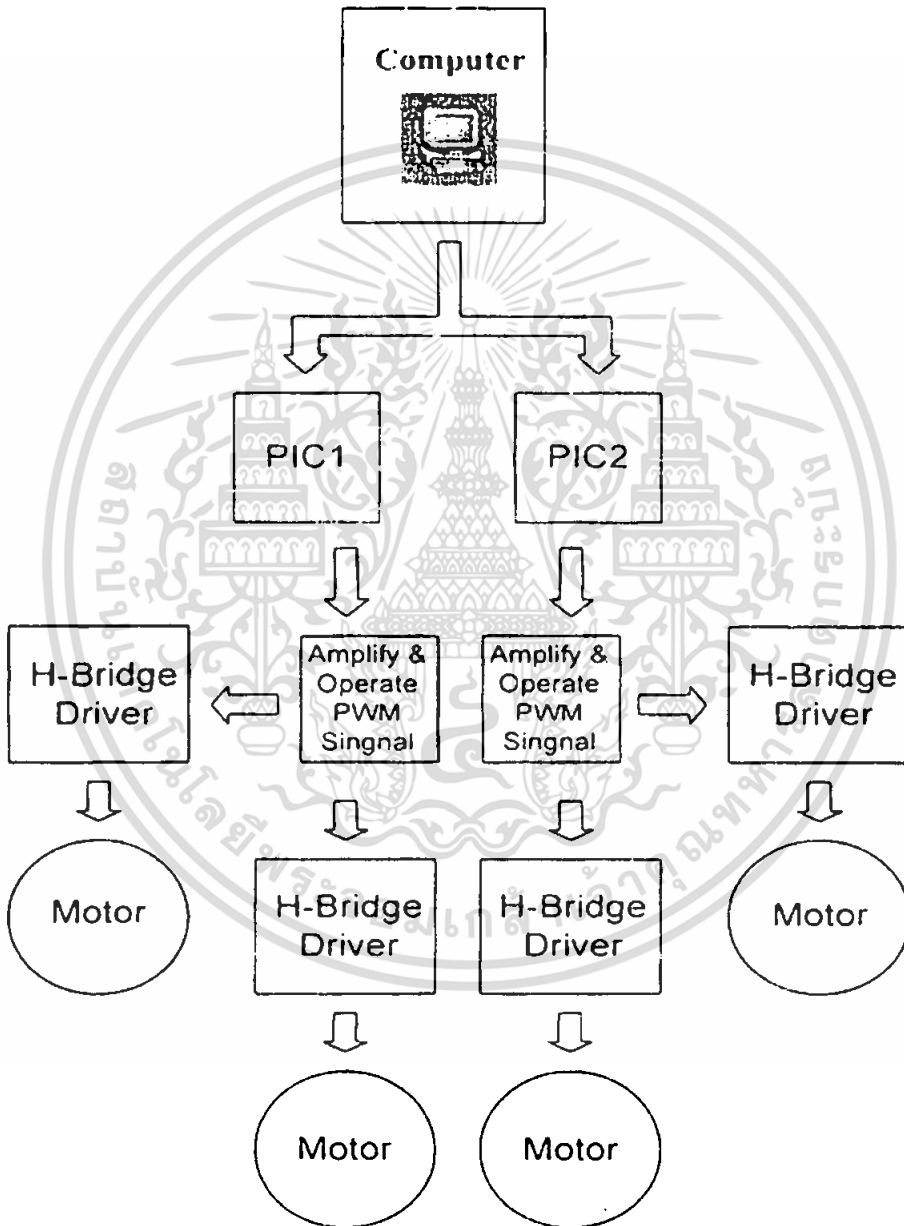
จากแผนผังดังรูปที่ 3.6 จะสามารถออกแบบโปรแกรมการทำงานของ PIC3 ได้ดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพแสดงขั้นตอนการทำงานในการรับค่าจากเซนเซอร์

3.5 กระบวนการคำนวณและส่งสัญญาณขับเคลื่อนมอเตอร์

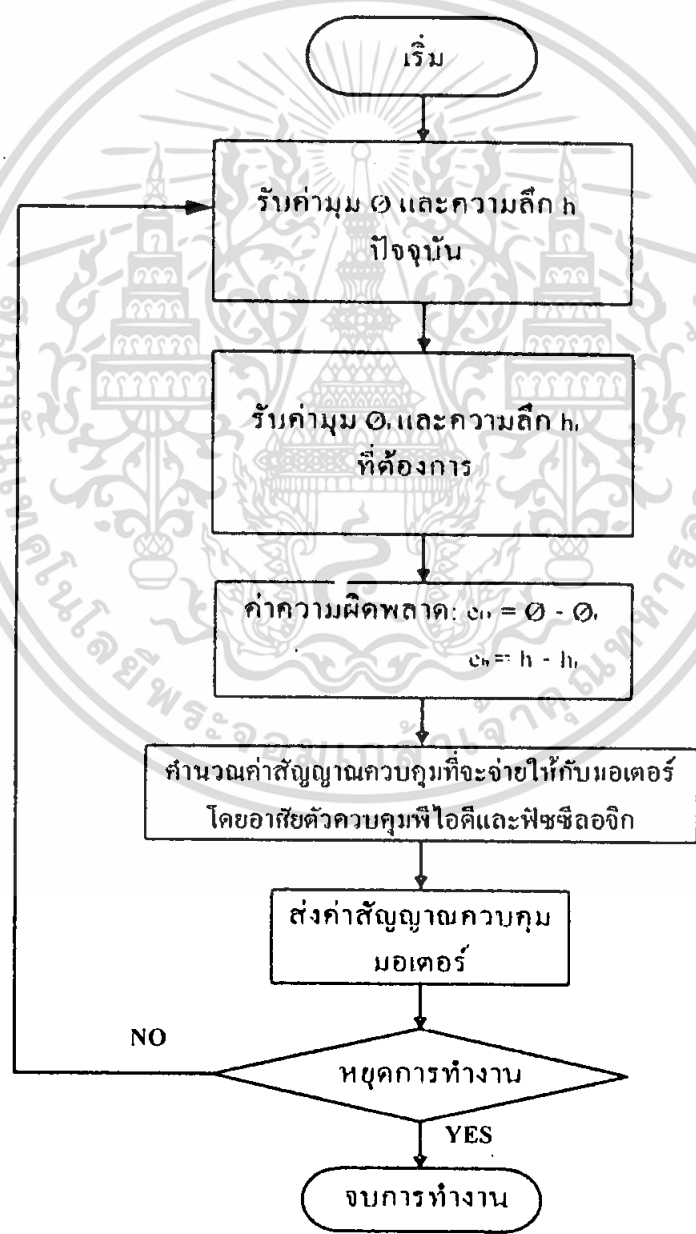
ในส่วนของการคำนวณนั้น จะใช้โปรแกรม Visual Basic คำนวณ นั่นคือ จะต้องมีการส่งสัญญาณเชื่อมต่อกันระหว่างคอมพิวเตอร์และคอนโทรลเลอร์ PIC1 และ PIC2 หลังจากนั้น คอนโทรลเลอร์ทั้ง 2 ตัวก็จะส่งงานไปที่วงจรควบคุมมอเตอร์เพื่อจ่ายพัลส์ให้มอเตอร์ทำงานตามต้องการ ดังแสดงในรูปที่ 3.11



รูปที่ 3.11 แผนภาพแสดงวิธีการส่งสัญญาณและการจ่ายพัลส์ให้แก่มอเตอร์

3.6 การรับค่าและคำนวณค่าพัลส์ที่จ่ายออกไปด้วยโปรแกรม Visual Basic

รูปที่ 3.12 แสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์ เริ่มต้นจากการรับค่ามุมกับระดับความลึกที่ต้องการ และค่ามุมกับระดับความลึกปัจจุบันซึ่งรับมาจาก PIC3 ที่รับค่าจากอุปกรณ์วัดดังกล่าวในหัวข้อ 3.2.3 จากนั้นจึงนำมาเปรียบเทียบเพื่อคำนวณค่าความผิดพลาด และนำมาหาสัญญาณควบคุมมอเตอร์โดยอาศัยตัวควบคุมแบบ PID หรือแบบฟัซซี่ลอจิก โดยใช้คาบเวลาของการสุ่มสัญญาณ (Sampling Time) ซึ่งได้จากการทดลองวัดเท่ากับ 150 ms จากนั้นนำค่าที่คำนวณได้ส่งไปยังตัวคอนโทรลเลอร์ PIC1 และ PIC2 ซึ่งคอนโทรลเลอร์ทั้ง 2 ตัวจะส่งสัญญาณ PWM ไปยังวงจรขยายสัญญาณ PWM ในรูปที่ 3.7 และส่งสัญญาณ PWM ไปยังวงจรขับมอเตอร์



รูปที่ 3.12 แผนภาพแสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์

3.7 การออกแบบตัวควบคุมแบบดั้งเดิม

ในโครงการนี้ได้ใช้ตัวควบคุมดั้งเดิมแบบพี แบบพีไอและแบบพีไอดีในการควบคุมทิศทาง สำหรับการควบคุมระดับความลึกของยานสำรวจได้ใช้ตัวควบคุมแบบพีไอและพีไอดีในการควบคุม การออกแบบเพื่อหาค่าเกณฑ์ของตัวควบคุมแต่ละแบบให้เหมาะสมกับยานสำรวจได้นี้จะใช้วิธีลองผิดลองถูกจากการทดลองควบคุมยานสำรวจได้นี้และเลือกค่าเกณฑ์ของตัวควบคุมแต่ละชนิดจำนวน 2 ค่าเพื่อไว้ใช้ในการทดสอบประสิทธิภาพของตัวควบคุมต่อไป ตารางที่ 3.1 และ 3.2 เป็นตารางสรุปผลของการหาค่าเกณฑ์ของตัวควบคุมแบบดั้งเดิมที่ใช้ในการควบคุมทิศทางและความลึกตามลำดับ ต่อไปนี้

ตารางที่ 3.1 ค่าเกณฑ์ในการควบคุมทิศทาง

ตัวควบคุม	K_p	K_i	K_d
P	4.5	-	-
	5		
PI	5	0.05	-
	5	0.1	
PID	5	0.05	0.1
	5	0.05	0.5

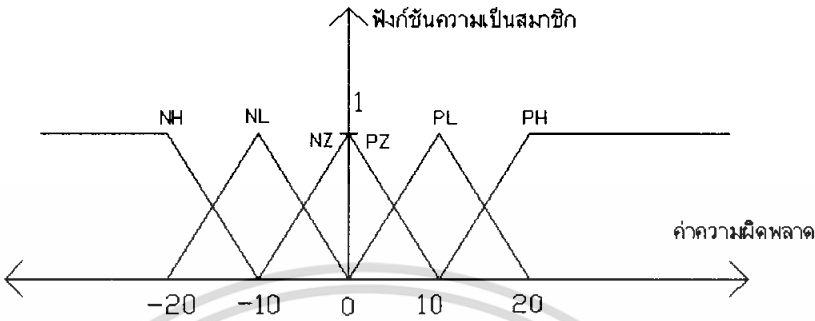
ตารางที่ 3.2 ค่าเกณฑ์ในการควบคุมระดับความลึก

ตัวควบคุม	K_p	K_i	K_d
PI	5	0.1	-
	5	0.5	
PID	5	0.1	0.05
	5	0.1	0.1

หลังจากที่ออกแบบตัวควบคุมดั้งเดิมแต่ละชนิดเป็นที่เรียบร้อยแล้ว ก็จะนำตัวควบคุมเหล่านั้นไปควบคุมยานสำรวจ ซึ่งจะนำเสนอผลการทดลองในบทที่ 4 โดยละเอียดต่อไป

3.8 การออกแบบตัวควบคุมฟัซซี่ลอจิก

ในการออกแบบตัวควบคุมแบบฟัซซี่ลอจิก ได้เลือกช่วงของฟังก์ชันความเป็นสมาชิกที่ดีที่สุดในการควบคุมทิศทางและความลึกของยานสำรวจได้นำได้ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการเลือกช่วงของฟังก์ชันความเป็นสมาชิกของ ERROR และ ΔERROR

และเขียนออกมาเป็นกฎการควบคุมฟัซซี่ได้ทั้งหมด 25 กฎ โดยสามารถเขียนเป็นเมตริกซ์ Fuzzy Associative (FAM) แสดงความสัมพันธ์ร่วมกันของตัวแปร $E:ERROR$ และ $\Delta E:\Delta ERROR$ ได้ดังตารางที่ 3.3

ตารางที่ 3.3 แสดงเมตริกซ์ FAM ของระบบควบคุมยานสำรวจได้นำ

$e \backslash \Delta e$	NH	NL	Z	PL	PH
NH	-255	-255	-255	-255	-255
NL	-127.5	-127.5	-127.5	-127.5	-127.5
Z	-63	-63	-50	63	63
PL	127.5	127.5	127.5	127.5	127.5
PH	255	255	255	255	255

สำหรับผลการทดลองที่ใช้ตัวควบคุมแบบฟัซซี่ควบคุมยานสำรวจได้นำจะได้นำจะแสดงผลการทดลองโดยละเอียดในบทต่อไป

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงผลการทดลองของการควบคุมยานสำรวจได้นำ ที่ใช้ตัวควบคุมแบบดั้งเดิมและตัวควบคุมแบบฟัซซี่ลอจิก

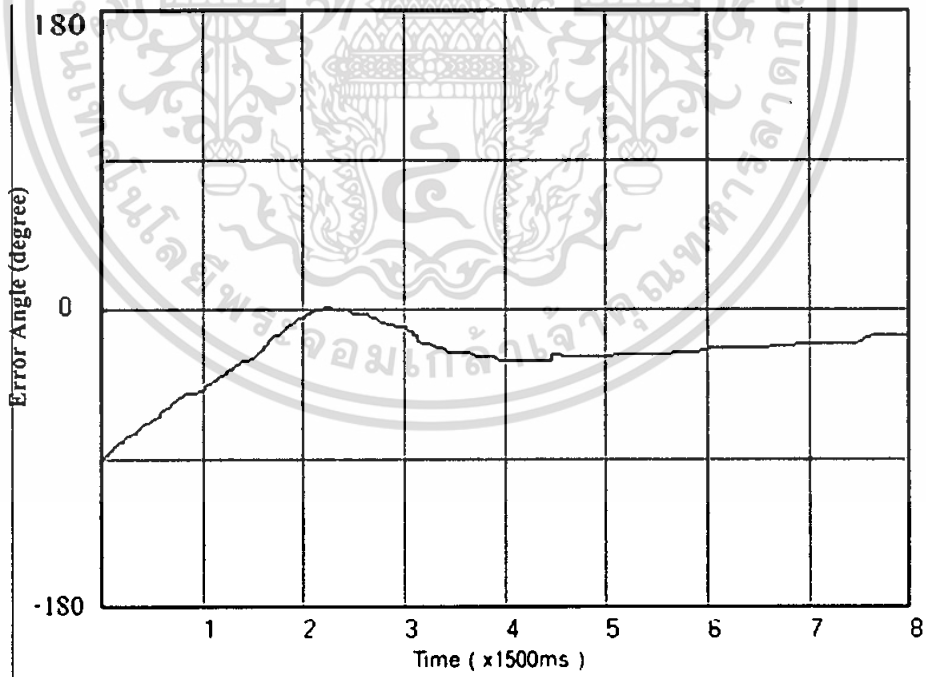
4.1 ผลการควบคุมทิศทาง

ในหัวข้อนี้ จะแสดงผลการทดลองในการควบคุมทิศทางของยานสำรวจได้นำ โดยใช้ตัวควบคุมแบบดั้งเดิมและตัวควบคุมแบบฟัซซี่ลอจิก

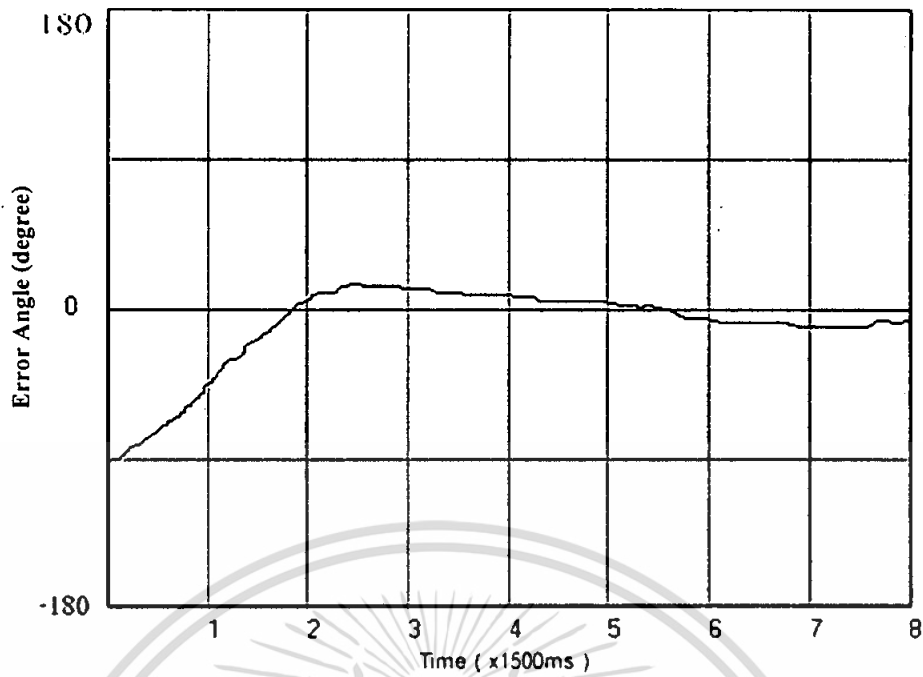
4.1.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบดั้งเดิม

ในการทดลองนี้ เป็นการทดลองเพื่อควบคุมทิศทางของยานสำรวจโดยใช้ตัวควบคุมแบบพี ตัวควบคุมแบบพีไอและตัวควบคุมแบบพีไอดี ด้วยค่าเกณฑ์ที่แสดงในตารางที่ 3.1

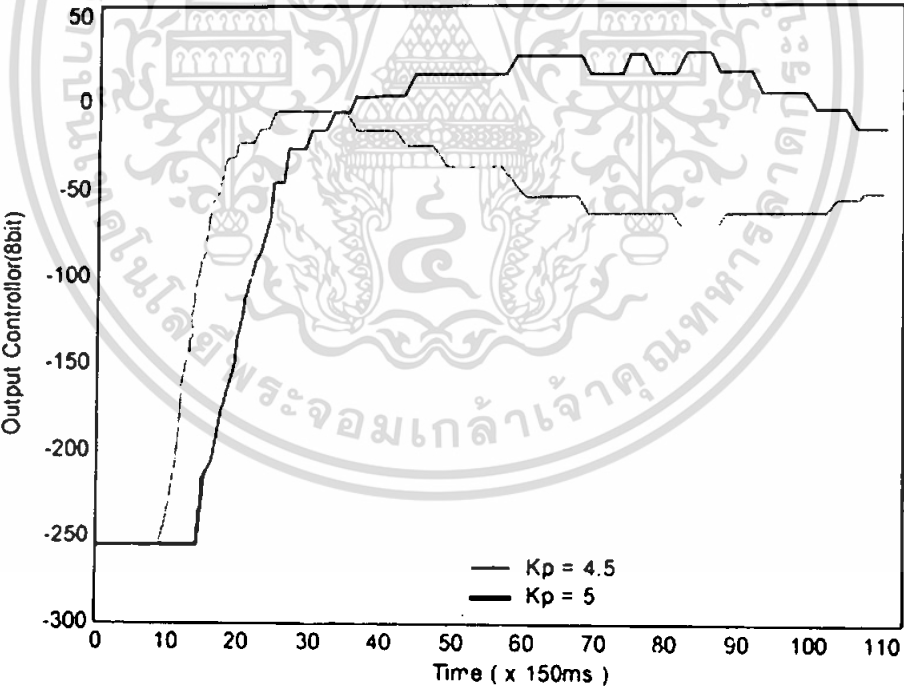
4.1.1.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพี



รูปที่ 4.1 การควบคุมทิศทางด้วยตัวควบคุมแบบพี $K_p = 4.5$



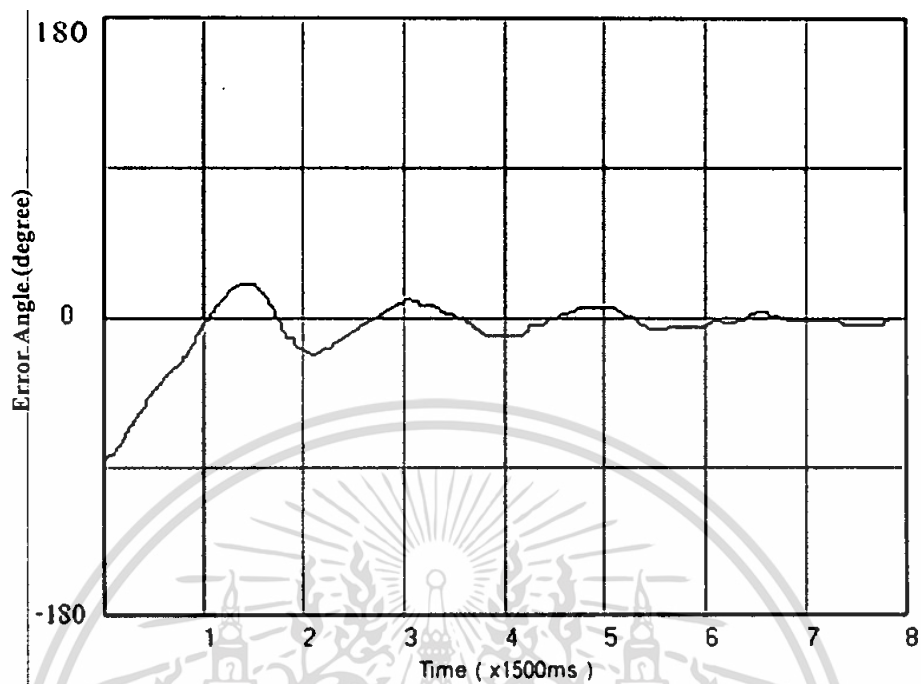
รูปที่ 4.2 การควบคุมทิศทางด้วยตัวควบคุมแบบพี $K_p = 5$



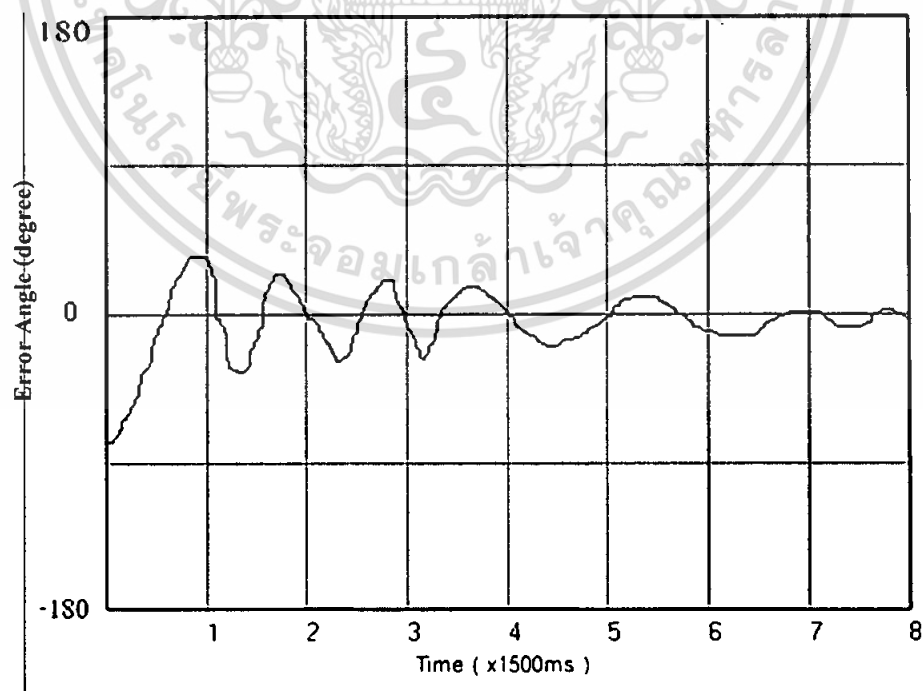
รูปที่ 4.3 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพี

จากการทดลองปรับเปลี่ยนค่าเกน K_p นั้น จะพบว่า ถ้าปรับค่าอัตราขยายตัวนี้ให้มีค่ามากเท่าไร จะทำให้ผลตอบสนองเร็วขึ้นเท่านั้น แต่ในขณะเดียวกัน ถ้าปรับค่าเกนตัวนี้เพิ่มขึ้นมากๆ ก็จะทำให้ผลตอบสนองเกิดการแกว่ง

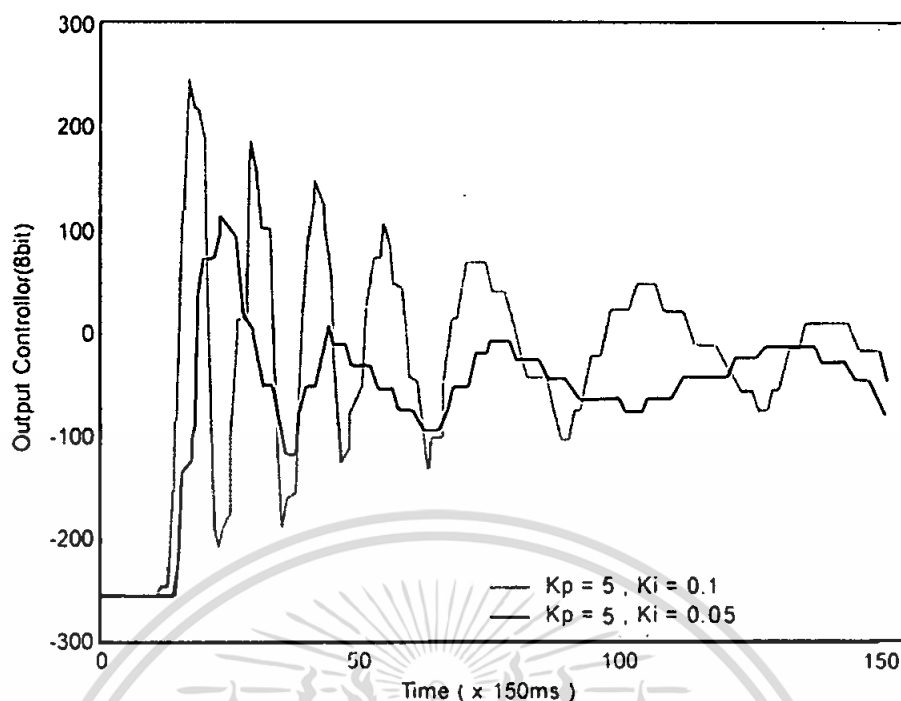
4.1.1.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีไอ



รูปที่ 4.4 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอ $K_p = 5$, $K_i = 0.05$



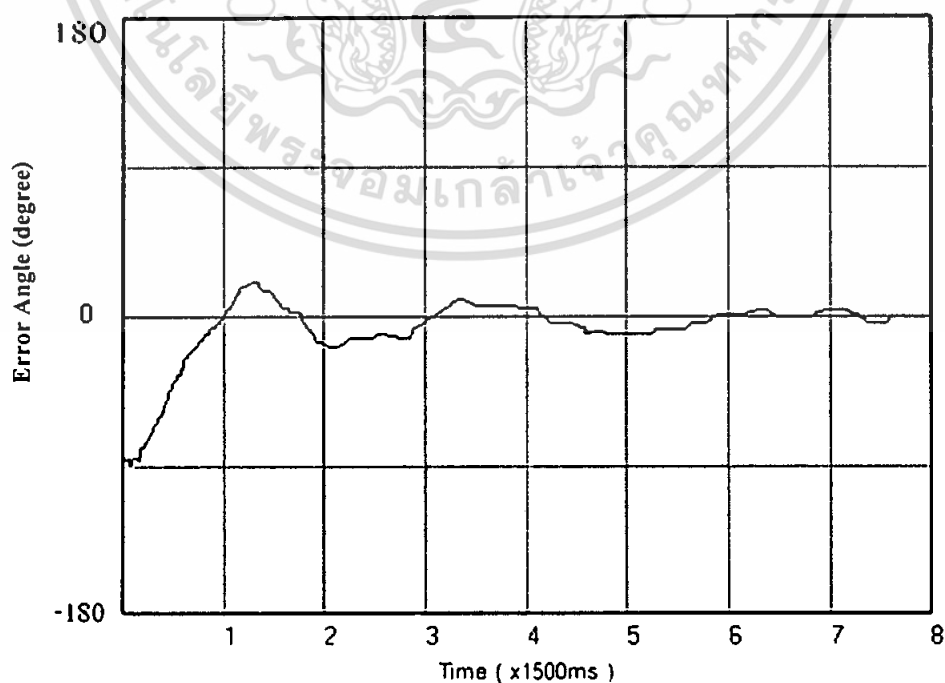
รูปที่ 4.5 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอ $K_p = 5$, $K_i = 0.1$



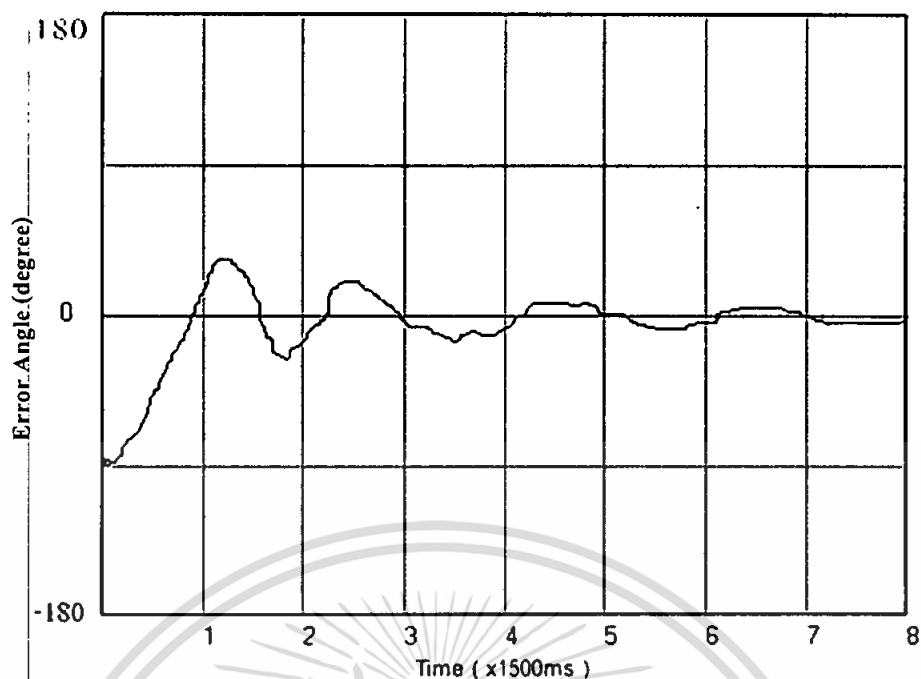
รูปที่ 4.6 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอ

เมื่อปรับเปลี่ยนเฉพาะค่า K_i ของตัวควบคุมแบบพีไอจะพบว่าค่าความผิดพลาดที่สภาวะคงตัวจะเป็นศูนย์ แต่ถ้า K_i มากจะมีผลตอบสนองเร็วโดยมีค่าพุ่งเกินและมีการแกว่ง

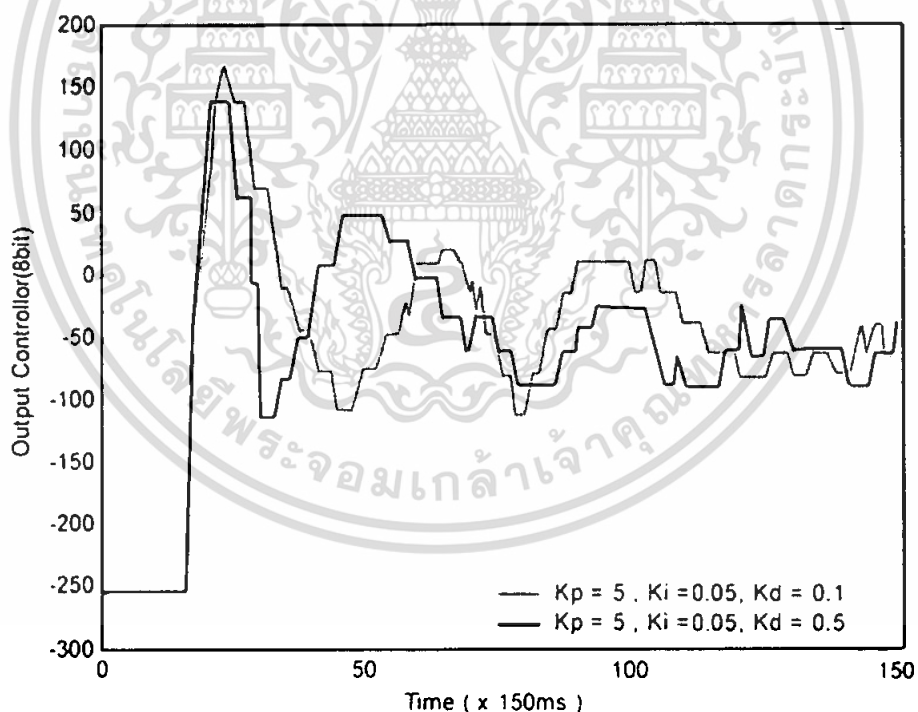
4.1.1.3 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีไอดี



รูปที่ 4.7 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.05$, $K_d = 0.1$



รูปที่ 4.8 การควบคุมทิศทางด้วยตัวควบคุมแบบพีไอดี $K_p = 5$, $K_i = 0.05$, $K_d = 0.5$

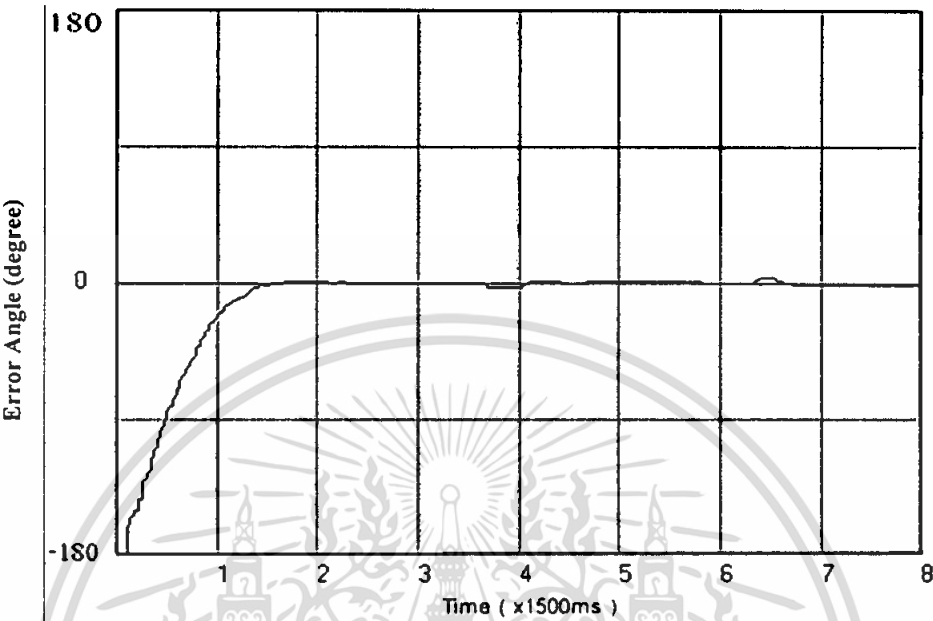


รูปที่ 4.9 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอดี

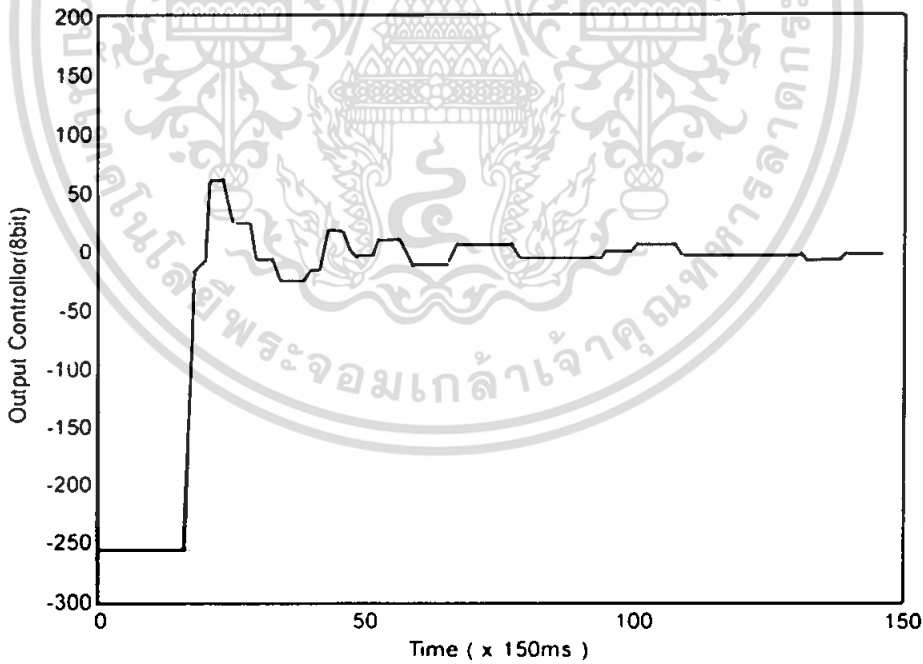
เมื่อทำการทดลองโดยใช้ตัวควบคุมแบบดั้งเดิมไป ทำให้ได้เป็นตัวควบคุมแบบพีไอดี จะพบว่า ค่าพ่วงเกินได้ลดลงและมีผลตอบสนองของระบบเร็วขึ้น จากการทดลองการควบคุมทิศทาง การเคลื่อนที่พบว่า สามารถใช้ตัวควบคุมแบบพีไอหรือพีไอดีควบคุมยานสำรวจได้น่านี้ได้ดีกว่าตัวควบคุมแบบพี

4.1.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบฟัซซี่ลอจิก

ในการทดลองนี้ เป็นการทดลองเพื่อควบคุมทิศทางของยานสำรวจโดยเปลี่ยนมาใช้ตัวควบคุมแบบฟัซซี่ลอจิกที่ได้ออกแบบไว้ตามหัวข้อที่ 3.8 ผลการควบคุมแสดงได้ดังรูปที่ 4.10



รูปที่ 4.10 การควบคุมทิศทางด้วยตัวควบคุมแบบฟัซซี่ลอจิก



รูปที่ 4.11 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบฟัซซี่

จากรูป จะเห็นว่าตัวควบคุมแบบฟัซซี่ลอจิกสามารถควบคุมทิศทางของยานสำรวจเข้าสู่เป้าหมายได้เร็วโดยไม่มีค่าพุ่งเกิน(Overshoot) และค่าความผิดพลาดที่สภาวะคงตัว สำหรับรูปที่ 4.11 เป็นรูปของสัญญาณควบคุมที่ได้จากตัวควบคุมแบบฟัซซี่ลอจิก

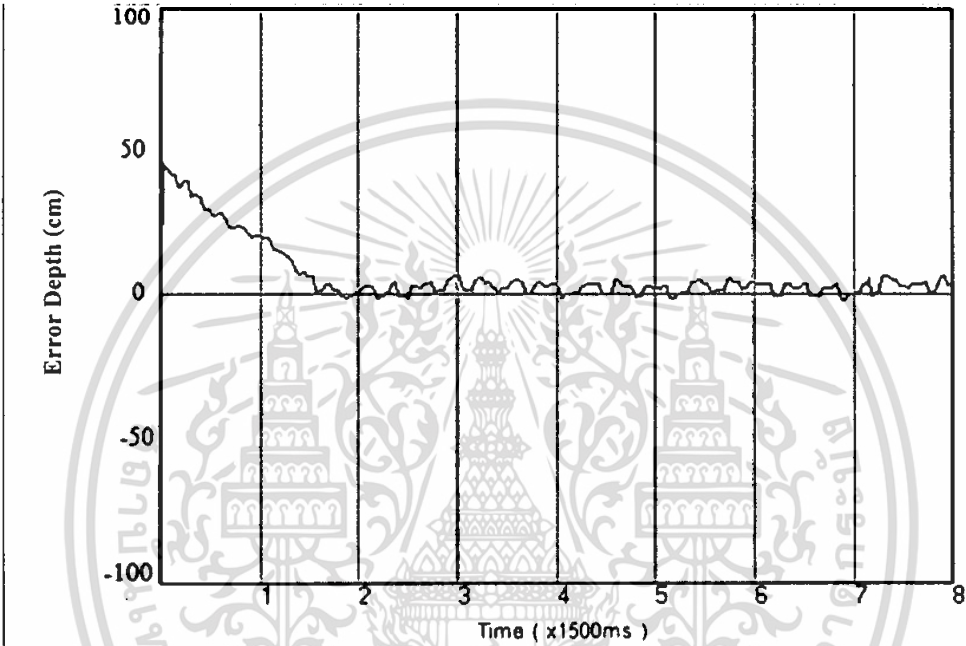
4.2 ผลการควบคุมระดับความลึก

ในหัวข้อนี้ จะแสดงผลการทดลองในการควบคุมระดับความลึกของยานสำรวจได้นำโดยใช้ตัวควบคุมแบบดั้งเดิมและตัวควบคุมแบบฟัซซีลอจิก

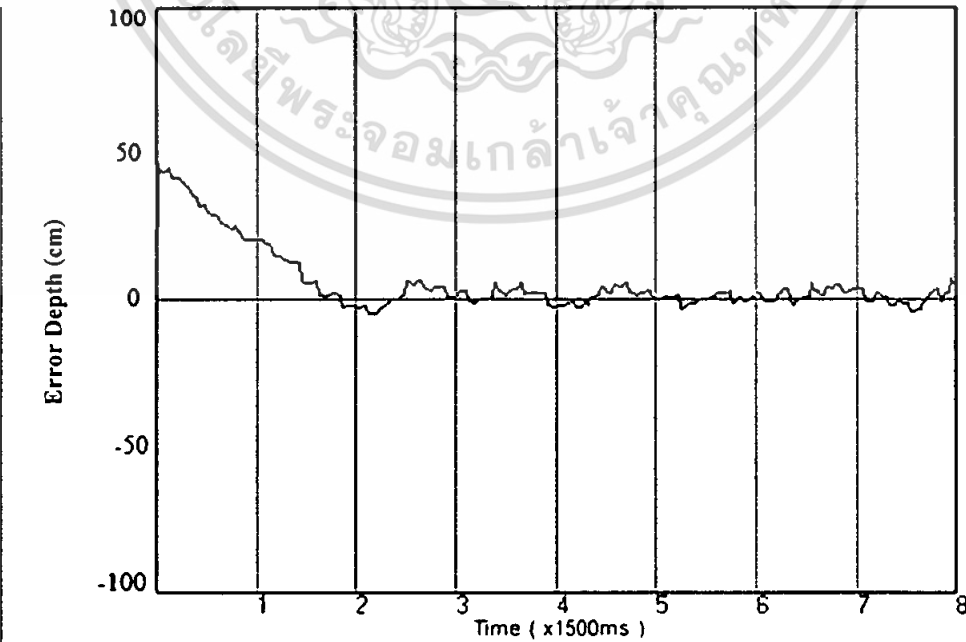
4.2.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบดั้งเดิม

ในการทดลองนี้ เป็นการทดลองเพื่อควบคุมระดับความลึกของยานสำรวจโดยใช้ตัวควบคุมแบบพีไอและตัวควบคุมแบบพีไอดี ด้วยค่าเกณฑ์ที่แสดงในตารางที่ 3.2

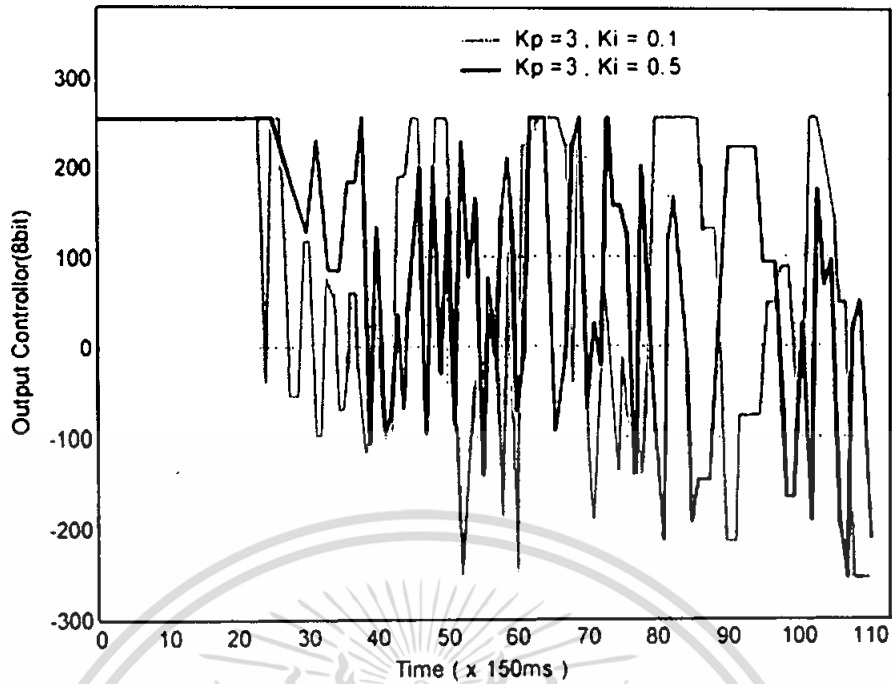
4.2.1.1 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีไอ



รูปที่ 4.12 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอ $K_p = 3$, $K_i = 0.1$



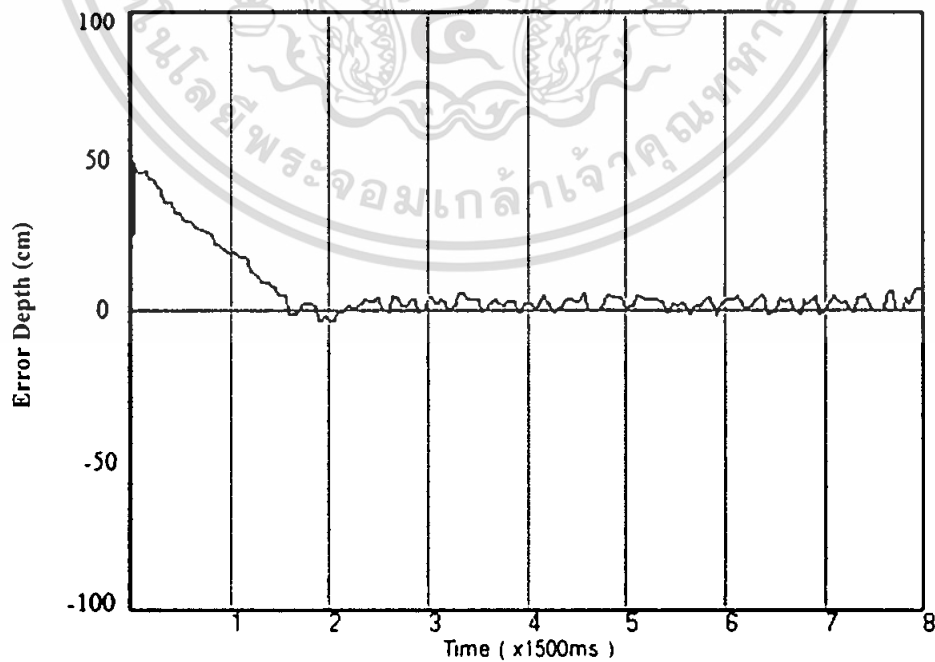
รูปที่ 4.13 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอ $K_p = 3$, $K_i = 0.5$



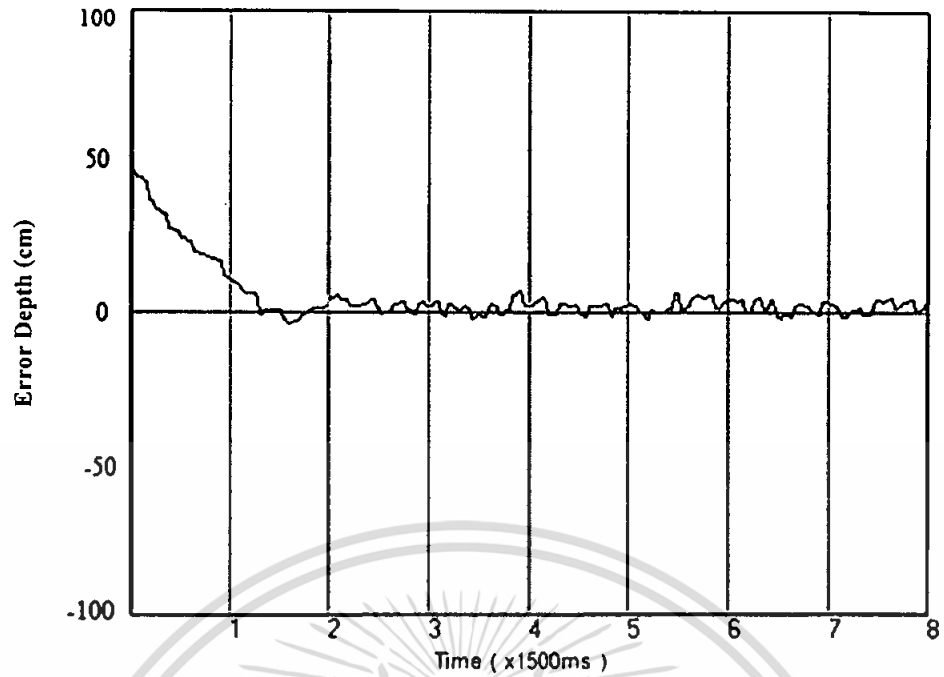
รูปที่ 4.14 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอ

เมื่อใช้ตัวควบคุมพีไอควบคุมระดับความลึก 1.5 เมตรของยานสำรวจได้นำพบว่า สามารถควบคุมได้โดยผลตอบสนองจะมีการแกว่งอยู่ที่ความลึก 1.5 เมตร แต่ถ้าเพิ่มค่าของเกน K_i ให้มากขึ้น พบว่าผลตอบสนองเร็วขึ้นแต่มีค่าพุ่งเกินเกิดขึ้น

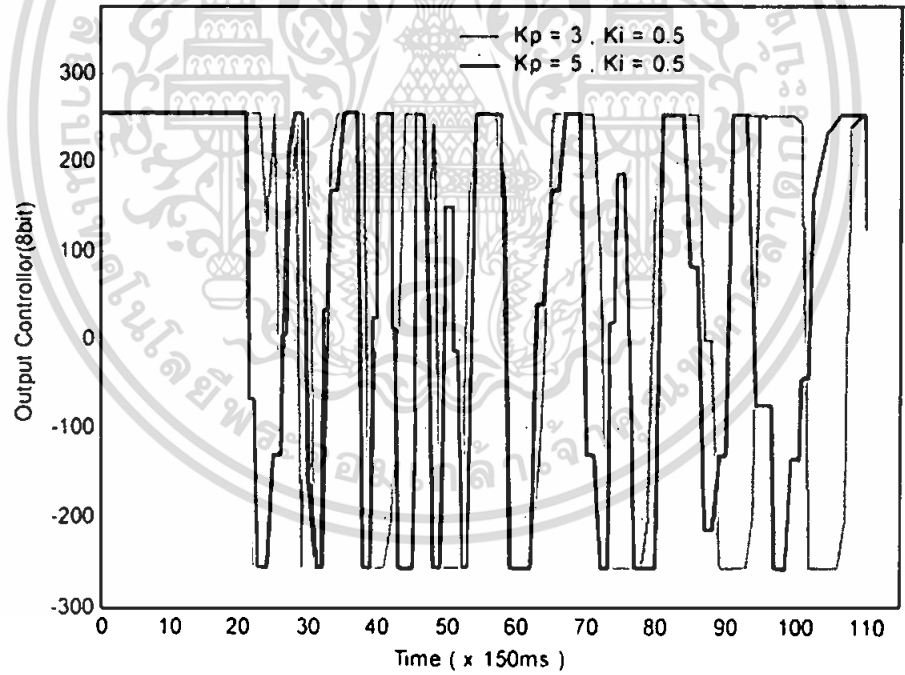
4.2.1.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบพีไอดี



รูปที่ 4.15 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอดี $K_p = 3, K_i = 0.1, K_d = 0.01$



รูปที่ 4.16 การควบคุมการดำน้ำด้วยตัวควบคุมแบบพีไอดี $K_p = 3$, $K_i = 0.1$, $K_d = 0.05$

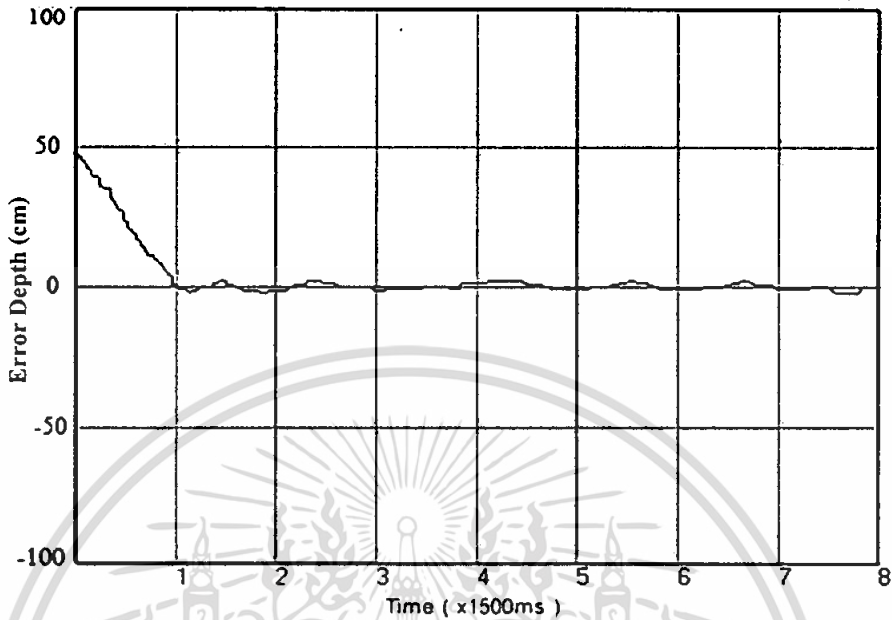


รูปที่ 4.17 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอดี

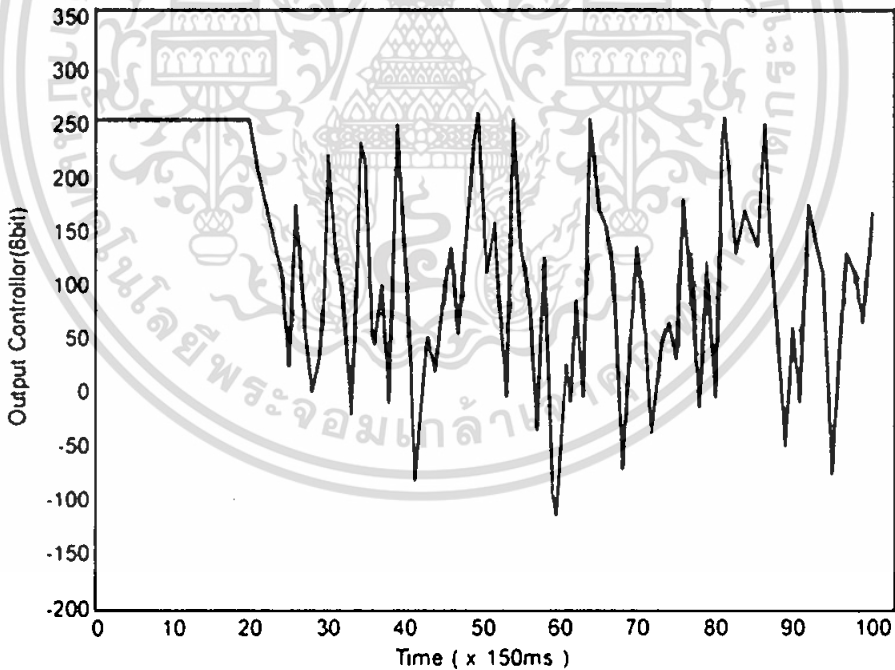
เมื่อทดลองควบคุมความลึกของยานสำรวจได้นำมาใช้ตัวควบคุมแบบพีไอดี พบว่า ผลตอบสนองจะเร็วกว่าและมีค่าพุ่งเกินน้อยกว่าการใช้ตัวควบคุมแบบพีไอ โดยผลตอบสนองจะ แกว่งเล็กน้อยที่สถานะคงตัวทั้งตัวควบคุมแบบพีไอและพีไอดี แต่สัญญาณควบคุมที่ใช้ตัวควบคุม แบบพีไอดีจะเกิดการอ้อมตัว

4.2.2 ผลการทดลองเมื่อใช้ตัวควบคุมแบบฟัซซีลอจิก

ในการทดลองนี้ เป็นการทดลองเพื่อควบคุมระดับความลึกของยานสำรวจโดยเปลี่ยนมาใช้ตัวควบคุมแบบฟัซซีลอจิกที่ได้ออกแบบไว้ตามหัวข้อที่ 3.8 ผลการควบคุมแสดงได้ดังรูปที่ 4.18



รูปที่ 4.18 การควบคุมการดำน้ำด้วยตัวควบคุมแบบฟัซซีลอจิก



รูปที่ 4.19 สัญญาณอินพุตของมอเตอร์เมื่อใช้ตัวควบคุมแบบฟัซซีลอจิก

จากผลการทดลองที่แสดงในรูปที่ 4.18 จะเห็นว่า ตัวควบคุมแบบฟัซซีลอจิกสามารถควบคุมระดับความลึกของยานสำรวจเข้าสู่เป้าหมายได้เร็วกว่าตัวควบคุมแบบพีไอและพีไอดีโดยไม่มีค่าพุ่งเกิน และไม่มีค่าความผิดพลาดที่สภาวะคงตัว สำหรับรูปที่ 4.19 เป็นรูปของสัญญาณควบคุมที่ได้จากตัวควบคุมแบบฟัซซีลอจิก

บทที่ 5

บทวิจารณ์และสรุป

5.1 ผลสัมฤทธิ์โครงการ

ยานสำรวจได้นำในรุ่นนี้เป็นรุ่นที่ 5 ซึ่งในปริญญานิพนธ์ฉบับนี้ได้ปรับปรุงและพัฒนาต่อจากยานสำรวจได้นำรุ่นที่ 4 โดยใช้ตัวควบคุมแบบดั้งเดิมและตัวควบคุมแบบฟัซซีลอจิกเพื่อควบคุมทิศทางและระดับความลึกได้อย่างอัตโนมัติ โดยเชื่อมต่อผ่านพอร์ตอนุกรมของคอมพิวเตอร์ด้วยโปรแกรมวิชวลเบสิกและโปรแกรมไมโครคอนโทรลเลอร์ ผลการทดลองสรุปได้ดังนี้

1. ผลการทดลองควบคุมทิศทางของยานสำรวจได้นำพบว่า ตัวควบคุมแบบฟัซซีลอจิกให้ผลตอบสนองตามที่ต้องการได้ โดยเลือกเลือกฟังก์ชันความเป็นสมาชิกให้เหมาะสมกับยานสำรวจได้นำ จึงทำให้สามารถควบคุมให้ยานสำรวจได้นำไปสู่ทิศทางที่กำหนดไว้และระบบมีเสถียรภาพในกรณีที่ใช้ตัวควบคุมแบบพี ไอและพี ไอดีควบคุมยานสำรวจพบว่า ก็จะให้ผลตอบสนองตามที่ต้องการได้เช่นเดียวกับตัวควบคุมแบบฟัซซีลอจิก แต่ตัวควบคุมแบบพีจะมีค่าความผิดพลาดที่สถานะคงตัว

2. ผลการทดลองควบคุมระดับความลึกของยานสำรวจได้นำพบว่า ตัวควบคุมแบบฟัซซีลอจิกซึ่งใช้ฟังก์ชันความเป็นสมาชิกเหมือนกับการควบคุมทิศทาง สามารถควบคุมยานสำรวจได้นำให้ดำสู่ระดับความลึกตามที่กำหนดไว้ได้เร็วกว่าตัวควบคุมแบบพี ไอและพี ไอดีโดยไม่มีค่าพุ่งเกินและไม่มีค่าความผิดพลาดที่สถานะคงตัว ในกรณีที่ใช้ตัวควบคุมแบบพี ไอและพี ไอดีควบคุมยานสำรวจได้นำ พบว่า ผลตอบสนองที่ใช้ตัวควบคุมแบบพี ไอดีจะเร็วกว่าและมีค่าพุ่งเกินน้อยกว่าการใช้ตัวควบคุมแบบพี ไอ แต่ผลตอบสนองจะแกว่งเล็กน้อยที่สถานะคงตัวทั้งตัวควบคุมแบบพี ไอและพี ไอดี

5.2 ปัญหาที่พบและแนวทางแก้ไข

ปัญหาที่พบในการศึกษาการควบคุมยานสำรวจได้นำคือ ปัญหาการวางตำแหน่งของอุปกรณ์วัดค่ามุม ถ้าวางอุปกรณ์วัดค่ามุมให้ใกล้กับมอเตอร์จะทำให้การวัดค่ามุมอาจผิดพลาดได้ในบางครั้ง เพราะมอเตอร์จะส่งสนามแม่เหล็กไฟฟ้าไปรบกวนการทำงานของอุปกรณ์วัดค่ามุม

แนวทางในการแก้ไขในการวัดค่ามุมให้ถูกต้องคือ จะต้องวางตำแหน่งของอุปกรณ์วัดค่ามุมให้อยู่ห่างจากมอเตอร์ให้มากที่สุดเท่าที่จะสามารถทำได้ เพื่อหลีกเลี่ยงการรบกวนจากสนามแม่เหล็กไฟฟ้าของมอเตอร์

5.3 ข้อเสนอแนะอื่นๆ

1. ศึกษาและหาวิธีการเพื่อป้องกันสัญญาณรบกวนที่เกิดขึ้นจากมอเตอร์ รวมทั้งสัญญาณรบกวนที่เกิดขึ้นภายในแผงวงจรที่มีไมโครคอนโทรลเลอร์เป็นส่วนประกอบ
2. ศึกษาและออกแบบซอฟต์แวร์ที่ใช้ในการควบคุมยานสำรวจได้นำให้มีประสิทธิภาพดีขึ้น โดยเปลี่ยนการส่งสัญญาณจากที่ส่งเป็นตัวเลขฐาน 2 ให้เป็นการส่งโดยใช้ตัวอักษรเป็นตัวเชื่อมต่อระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์



เอกสารอ้างอิง

- [1] กิตติ ภัคดีวัฒนกุล และ จำลอง ทรูตสาหะ. **Visual Basic 6 (ฉบับโปรแกรมเมอร์)**. หจก. ไทยเจริญการพิมพ์ 64-68 หมู่ 5 ถนนวงแหวนรอบนอก แขวงบางบอน เขตบางบอน กรุงเทพฯ 10150
- [2] ประจัน พลังสันติกุล. **เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC**. บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด กรุงเทพฯ, พิมพ์ครั้งที่ 1/2547
- [3] อภิชาติ ภู่วัฒน. **เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic**. บริษัท Infopress Developer Book. นนทบุรี. 2546
- [4] นายคณัย นุ่มนัม. **ว่าที่ร้อยตรีปริญญา ปัญญาไว “การควบคุมยานสำรวจใต้น้ำ.”** วิทยานิพนธ์ ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2548
- [5] ข้อมูลจาก <http://www.alldatasheet.com>

ภาคผนวก



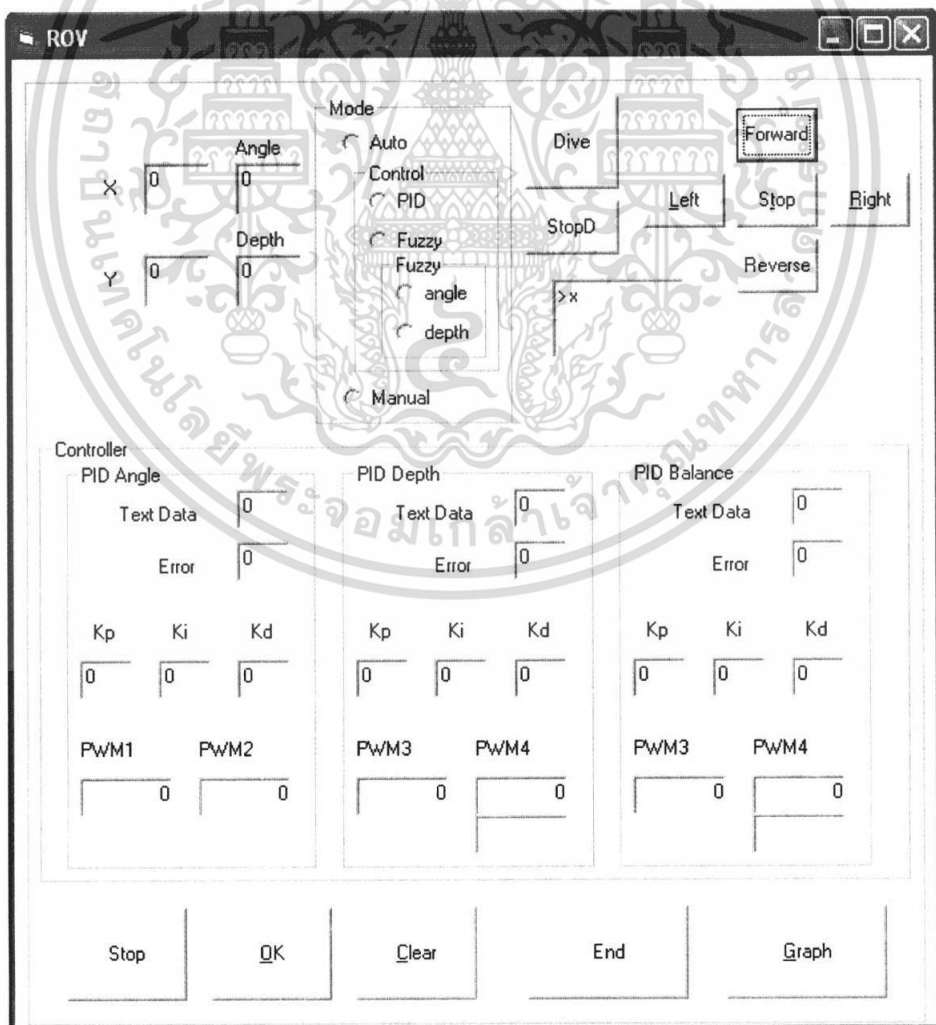
ภาคผนวก ก.

โปรแกรมที่ใช้ในโครงการ

ในบทนี้จะอธิบายถึงส่วนของโปรแกรมต่างๆที่ใช้ในโครงการนี้ ซึ่งประกอบด้วย 2 ส่วนหลัก คือ ส่วนโปรแกรมที่ใช้ติดต่อกับผู้ควบคุมบนคอมพิวเตอร์ส่วนบุคคล และ ส่วนโปรแกรมที่อยู่ในไมโครคอนโทรเลอร์ที่ใช้ควบคุมยานสำรวจ

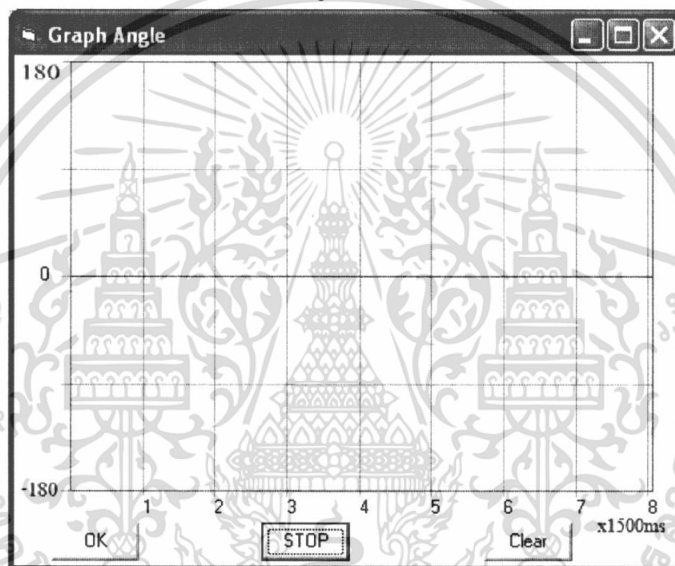
ก.1 โปรแกรมที่ติดต่อกับผู้ควบคุม

โปรแกรมที่ติดต่อกับผู้ควบคุมบนคอมพิวเตอร์ส่วนบุคคล ซึ่งในโครงการนี้จะใช้โปรแกรม Visual Basic ซึ่งจะประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ส่วนหน้าต่างที่เชื่อมต่อกับผู้ใช้งาน(User Interface) และ ตัวโปรแกรมที่เขียนขึ้นเพื่อควบคุมหน้าต่างเชื่อมต่อกับผู้ใช้งาน(Code)

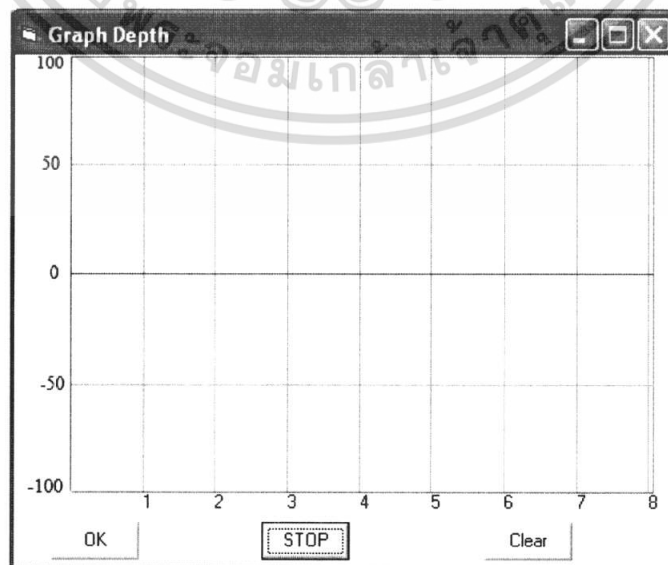


รูปที่ ก.1 หน้าต่างแสดงผลที่เชื่อมต่อกับผู้ควบคุม

ส่วนที่ใช้เชื่อมต่อระหว่างผู้ใช้งานและตัวยาน จะถูกเปิดขึ้นเมื่อเริ่มใช้งานดังแสดงในรูป ก.1 เริ่มแรกโปรแกรมจะแสดงค่ามุมและความลึกปัจจุบันในช่อง Angle และ Depth ตามลำดับ ในส่วนของการทำงานสามารถเลือกโหมดการทำงาน และโหมดตัวควบคุมได้ ส่วนโหมดการทำงานเราสามารถเลือกได้ 2 โหมดคือ Manual และ Auto โดยในโหมด Manual นั้นเราสามารถควบคุมทิศทางได้โดยการกดปุ่ม Forward, Reverse, Left หรือ Right และเมื่อต้องการหยุดการเคลื่อนที่ให้กดปุ่ม Stop และในโหมด Auto นั้น เราสามารถเลือกตัวควบคุมได้ 2 แบบคือ แบบ PID และ FUZZY LOGIC ซึ่งสามารถกำหนดค่ามุมและระดับความลึกที่ต้องการได้ในช่อง Text Data และในส่วนของการควบคุมแบบ PID เราสามารถระบุใส่ค่าอัตราขยายของตัวควบคุม PID ในการใช้งานได้ โปรแกรมสามารถแสดงกราฟได้โดยกดที่ปุ่ม Graph ซึ่งจะมีกราฟแสดงทิศทางดังรูปที่ ก.2 และ กราฟแสดงความลึกดังรูปที่ ก.3



รูปที่ ก.2 หน้าต่างแสดงกราฟทิศทางของยานสำรวจ



รูปที่ ก.3 หน้าต่างแสดงกราฟความลึกของยานสำรวจ

ในส่วนของโค้ด ที่สำคัญของ โปรแกรมจะเป็นส่วนที่เขียนขึ้นเพื่อใช้เป็นตัวควบคุมงานสำรวจมีดังนี้

```
pom1_s = Comm1.Input
```

```
Text1.Text = pom1_s
```

```
t = Text1.Text
```

//นำข้อมูลที่ส่งมาให้ทั้งหมดเก็บไว้ในตัวแปร t

```
c = Len(t)
```

```
searchPattern$ = "a"
```

```
pos1 = InStr(t, searchPattern$)
```

```
searchPattern$ = "b"
```

```
pos2 = InStr(t, searchPattern$)
```

```
searchPattern$ = "c"
```

```
pos3 = InStr(t, searchPattern$)
```

```
searchPattern$ = "d"
```

```
pos4 = InStr(t, searchPattern$)
```

```
searchPattern$ = "e"
```

```
pos5 = InStr(t, searchPattern$)
```

```
If pos1 > 0 And pos2 > 0 Then
```

```
    x1 = pos2 - pos1
```

```
    If x1 > 0 Then
```

```
        a = Mid$(t, pos1 + 1, x1 - 1)
```

```
        Text4 = a          'output data a _Angle
```

```
        sensor_angle = a
```

```
    End If
```

```
End If
```

```
*****
```

```
If pos2 > 0 And pos3 > 0 Then
```

```
    x2 = pos3 - pos2
```

```
    If x2 > 0 Then
```

```
        b = Mid$(t, pos2 + 1, x2 - 1)
```

```
        Text5 = b          'output data b
```

```
    End If
```

```
End If
```

```
*****
```

```

If pos4 > 0 And pos3 > 0 Then
    X3 = pos4 - pos3
    If X3 > 0 Then
        c = Mid$(t, pos3 + 1, X3 - 1)
        Text2 = c          'output data c
    End If
End If

'*****

If pos5 > 0 And pos4 > 0 Then
    X4 = pos5 - pos4
    If X4 > 0 Then
        d = Mid$(t, pos4 + 1, X4 - 1)
        Text3 = d          'output data d
    End If
End If

'*****

sensor_angle = Val(Text4.Text)
If (Text6.Text) <> 0 Then
    If Text6.Text = "" Then
        Text6.Text = 0
    End If
    If Text6.Text > 359 Then
        Text6.Text = 359
    End If
    If Text6.Text < -359 Then
        Text6.Text = -359
    End If
End If

data_angle = Val(Text6.Text)
Ki1_text = Val(Text_ki1.Text)
Kd1_text = Val(Text_kd1.Text)
Ka1_text = Val(Text_ka1.Text)

```

```

'caculate error
    error_angle = data_angle - sensor_angle
    If error_angle > 0 And error_angle < 180 Then
        error_angle2 = error_angle * -1
    '    Text16.Text = error_angle2
    End If
    If error_angle < 0 And error_angle > -180 Then
        error_angle2 = error_angle * -1
    End If
    If error_angle >= 180 And error_angle <= 360 Then
        error_angle2 = ((359 - data_angle) + sensor_angle)
    End If
    If error_angle <= -180 And error_angle >= -360 Then
        error_angle2 = ((359 - sensor_angle) + data_angle) * -1
    End If
    If error_angle = 0 Then
        error_angle2 = 0
    End If
    Text7.Text = error_angle2
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
' Auto or manual
    If Not (angle_auto = 0) Then ' auto
        If Controller = 1 Then
            'caculate Ka_M1
            Ka1_M1 = Ka1_text * error_angle2
            'caculate KI_M1
            Ki1_M1 = Ki1_M1_old + (Ki1_text * (Ts / 2 * (error_angle2 + eM1_old)))
            Ki1_M1_old = Ki1_M1
            'caculate Kd_M1
            Kd1_M1 = ((error_angle2 - eM1_old) / Ts) * Kd1_text
            'caculate PID_M1
            PID_M1 = ((Ka1_M1) + Ki1_M1 + Kd1_M1) + go_speed1

```

```

If PID_M1 <> 0 Then
    If PID_M1 > 255 Then
        PID_M1 = 255
    End If
    If PID_M1 < -255 Then
        PID_M1 = -255
    End If
End If

Text11.Text = PID_M1
motor1 = PID_M1
'xxxxxxxxxxxxxxxxxxxxxxxxxxxx
'caculate Ka_M2
Ka1_M2 = Ka1_text * error_angle2
'caculate Ki_M2
Ki1_M2 = Ki1_M2_old + (Ki1_text * (Ts / 2 * (error_angle2 + eM1_old)))
Ki1_M2_old = Ki1_M2
'caculate Kd_M2
Kd1_M2 = ((error_angle2 - eM1_old) / Ts) * Kd1_text
'caculate PID_M2
PID_M2 = ((Ka1_M2) + Ki1_M2 + Kd1_M2) + go_speed2
If PID_M2 <> 0 Then
    If PID_M2 > 255 Then
        PID_M2 = 255
    End If
    If PID_M2 < -255 Then
        PID_M2 = -255
    End If
End If

Text8.Text = PID_M2
'motor2 = PID_M2
eM1_old = error_angle2

```

```

If PID_M1 > 0 Then
    control_M1 = 253
Else
    control_M1 = 254
    PID_M1 = -1 * PID_M1
End If
If PID_M2 > 0 Then
    control_M2 = 251
Else
    control_M2 = 247
    PID_M2 = -1 * PID_M2
End If
If Controller = 1 Then
    control1 = control_M1 And control_M2
    control2 = 255
End If
If angle_auto = 255 Then
    If control1 = 246 Then
        sending = "S"
    ElseIf control1 = 249 Then
        sending = "W"
    End If
End If
PID_M1 = PID_M1 + 50
If PID_M1 > 255 Then
    PID_M1 = 255
End If
tmp1 = PID_M1
If Len(tmp1) = 1 Then
    tmp1 = "00" & tmp1
End If
If Len(tmp1) = 2 Then

```

```

    tmp1 = "0" & tmp1
End If

If Len(tmp1) = 3 Then
    tmp1 = "" & tmp1
End If

PID_M2 = PID_M2 + 50
If PID_M2 > 255 Then
    PID_M2 = 255
End If

tmp2 = PID_M2
If Len(tmp2) = 1 Then
    tmp2 = "00" & tmp2
End If
If Len(tmp2) = 2 Then
    tmp2 = "0" & tmp2
End If
If Len(tmp2) = 3 Then
    tmp2 = "" & tmp2
End If
DataCD = "c" + tmp1 + tmp2 + sending + "0" + "d"
' Data = c0000d
Comm1.Output = DataCD
Comm1.Output = DataAB
Text16.Text = ">" & DataCD & "-" & DataAB
' caculate PID depth

*****

sensor_depth = Val(Text5.Text)
data_Depth = Val(Text12.Text)
Ki2_text = Val(Text_ki2.Text)
Kd2_text = Val(Text_kd2.Text)
Ka2_text = Val(Text_ka2.Text)
'caculate error depth

```

```

error_depth = (data_Depth - sensor_depth)

Text13.Text = error_depth

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

'caculate Ka2_M1

Ka2_M1 = Ka2_text * error_depth

'caculate KI2_M1

Ki2_M1 = Ki2_M1_old + (Ki2_text * (Ts / 2 * (error_depth + eM2_old)))

Ki2_M1_old = Ki2_M1

'caculate Kd2_M1

Kd2_M1 = ((error_depth - eM2_old) / Ts) * Kd2_text

'caculate PID2_M1

PID2_M1 = ((Ka2_M1) + Ki2_M1 + Kd2_M1)

If PID2_M1 <> 0 Then
    If PID2_M1 > 255 Then
        PID2_M1 = 255
    End If
    If PID2_M1 < -255 Then
        PID2_M1 = -255
    End If
End If

Text17.Text = PID2_M1

motor3 = PID2_M1

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

'caculate Ka2_M2

Ka2_M2 = Ka2_text * error_depth

'caculate KI2_M2

Ki2_M2 = Ki2_M2_old + (Ki2_text * (Ts / 2 * (error_depth + eM2_old)))

Ki2_M2_old = Ki2_M2

'caculate Kd2_M1

Kd2_M2 = ((error_depth - eM2_old) / Ts) * Kd2_text

'caculate PID2_M2

PID2_M2 = ((Ka2_M2) + Ki2_M2 + Kd2_M2)

```

```

If PID2_M2 <> 0 Then
    If PID2_M2 > 255 Then
        PID2_M2 = 255
    End If
    If PID2_M2 < -255 Then
        PID2_M2 = -255
    End If
End If

Text9.Text = PID2_M2

'motor4 = PID2_M2
eM2_old = error_depth
'caculate PID Balance *****
sensor_Balance = Val(Text2.Text)
data_Balance = Val(Text21.Text)
Ki3_text = Val(Text_ki3.Text)
Kd3_text = Val(Text_kd3.Text)
Ka3_text = Val(Text_ka3.Text)
'caculate error Balance
error_Balance = (data_Balance - sensor_Balance)
Text22.Text = error_Balance
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
'caculate Ka2_M1
Ka3_M1 = Ka3_text * -error_Balance
'caculate KI3_M1
Ki3_M1 = Ki3_M1_old + (Ki3_text * (Ts / 2 * (-error_Balance + eM3_old)))
Ki3_M1_old = Ki3_M1
'caculate Kd3_M1
Kd3_M1 = ((-error_Balance - eM3_old) / Ts) * Kd3_text
'caculate PID3_M1
PID3_M1 = ((Ka3_M1) + Ki3_M1 + Kd3_M1)
If PID3_M1 <> 0 Then
    If PID3_M1 > 255 Then

```

```

PID3_M1 = 255
End If
If PID3_M1 < -255 Then
    PID3_M1 = -255
End If
End If
Text23.Text = PID3_M1
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
'caculate Ka3_M2
Ka3_M2 = Ka3_text * error_Balance
'caculate KI3_M2
Ki3_M2 = Ki3_M2_old + (Ki3_text * (Ts / 2 * (error_Balance + eM3_old)))
Ki3_M2_old = Ki3_M2
'caculate Kd3_M2
Kd3_M2 = ((error_Balance - eM3_old) / Ts) * Kd3_text
'caculate PID3_M2
PID3_M2 = ((Ka3_M2) + Ki3_M2 + Kd3_M2)
If PID3_M2 <> 0 Then
    If PID3_M2 > 255 Then
        PID3_M2 = 255
    End If
    If PID3_M2 < -255 Then
        PID3_M2 = -255
    End If
End If
Text10.Text = PID3_M2
eM3_old = error_Balance
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
PWM_M1 = PID2_M1 + PID3_M1
PWM_M2 = PID2_M2 + PID3_M2
Text14.Text = PWM_M1
Text15.Text = PWM_M2

```

If PWM_M1 > 0 Then

 control_M1 = 253

End If

If PWM_M1 < 0 Then

 control_M1 = 254

 PWM_M1 = -1 * PWM_M1

End If

If PWM_M2 > 0 Then

 control_M2 = 251

End If

If PWM_M2 < 0 Then

 control_M2 = 247

 PWM_M2 = -1 * PWM_M2

End If

control1 = control_M1 And control_M2

control2 = 255

If stop1 = 0 Then

 If control1 = 246 Then

 sending = "A"

 ElseIf control1 = 249 Then

 sending = "D"

 End If

'Dim tmp1, tmp2 As String

End If

If PWM_M1 > 255 Then

 PWM_M1 = 255

End If

tmp1 = PWM_M1

If Len(tmp1) = 1 Then

 tmp1 = "00" & tmp1

End If

If Len(tmp1) = 2 Then

```

    tmp1 = "0" & tmp1
End If
If Len(tmp1) = 3 Then
    tmp1 = "" & tmp1
End If
    If PWM_M2 > 255 Then
        PWM_M2 = 255
    End If
    tmp2 = PWM_M2
    If Len(tmp2) = 1 Then
        tmp2 = "00" & tmp2
    End If
    If Len(tmp2) = 2 Then
        tmp2 = "0" & tmp2
    End If
    If Len(tmp2) = 3 Then
        tmp2 = "" & tmp2
    End If
    DataCD = "x" + tmp1 + tmp2 + sending + "0" + "y"
    Comm1.Output = DataCD
    Comm1.Output = DataAB
    Text16.Text = ">" & DataAB & " - " & DataCD
' end if pid
Else
    ' fuzzy
    'calculate Fuzzy angle
    If DEF = 255 And stop1 = 1 Then
        error_depth = 0
    End If
    If error_angle2 >= 0 And error_angle2 <= 10 Then
        yPZ = (1 / 10) * (error_angle2 - 10) * -1
        yPL = (1 / 10) * error_angle2

```

PWM_M3 = (yPZ * go_speed3) + (yPL * 127.5)

PWM_M4 = (yPZ * go_speed3) + (yPL * 127.5)

End If

If error_angle2 > 10 And error_angle2 <= 20 Then

yPL = (1 / 10) * (error_angle2 - 20) * -1

yPH = (1 / 10) * (error_angle2 - 10)

PWM_M3 = (yPL * 127.5) + (yPH * 255)

PWM_M4 = (yPL * 127.5) + (yPH * 255)

End If

If error_angle2 > 20 Then

yPH = 1

PWM_M3 = 255

PWM_M4 = 255

End If

If error_angle2 < 0 And error_angle2 >= -10 Then

yNZ = (1 / 10) * (error_angle2 + 10)

yNL = (1 / 10) * error_angle2 * -1

PWM_M3 = ((yNZ * 50) + (yNL * 127.5)) * -1

PWM_M4 = ((yNZ * 50) + (yNL * 127.5)) * -1

End If

If error_angle2 < -10 And error_angle2 >= -20 Then

yNL = (1 / 10) * (error_angle2 + 20) * -1

yNH = (1 / 10) * (error_angle2 + 10)

PWM_M3 = ((yNL * 127.5) + (yNH * 255)) * -1

PWM_M4 = ((yNL * 127.5) + (yNH * 255)) * -1

End If

If error_angle2 < -20 Then

yNH = 1

PWM_M3 = -255

PWM_M4 = -255

End If

```

Text11.Text = PWM_M3

Text8.Text = PWM_M4

If PWM_M3 > 0 Then
    control_M3 = 253          '11111101
End If

If PWM_M3 < 0 Then
    control_M1 = 254          '11111110
    PWM_M3 = -1 * PWM_M3
End If

If PWM_M4 > 0 Then
    control_M4 = 251          '11111011
End If

If PWM_M4 < 0 Then
    control_M2 = 247          '11110111
    PWM_M4 = -1 * PWM_M4
End If

control1 = control_M1 And control_M2
control2 = 255

If angle_auto = 255 Then
    If control1 = 246 Then
        sending = "S"
    ElseIf control1 = 249 Then
        sending = "W"
    End If
End If

'-----

'Dim tmp1, tmp2 As String
PWM_M3 = Round(PWM_M3)
PWM_M3 = PWM_M3 + 10

If PWM_M3 > 255 Then
    PWM_M3 = 255
End If

tmp1 = PWM_M3

```

```

If Len(tmp1) = 1 Then
    tmp1 = "00" & tmp1
End If

If Len(tmp1) = 2 Then
    tmp1 = "0" & tmp1
End If

If Len(tmp1) = 3 Then
    tmp1 = "" & tmp1
End If

PWM_M4 = Round(PWM_M4)
PWM_M4 = PWM_M4 + 10
If PWM_M4 > 255 Then
    PWM_M4 = 255
End If
tmp2 = PWM_M4
If Len(tmp2) = 1 Then
    tmp2 = "00" & tmp2
End If
If Len(tmp2) = 2 Then
    tmp2 = "0" & tmp2
End If
If Len(tmp2) = 3 Then
    tmp2 = "" & tmp2
End If

DataCD = "c" + tmp1 + tmp2 + sending + "0" + "d"

Comm1.Output = DataCD

Comm1.Output = DataAB

Text16.Text = ">" & DataCD & " - " & DataAB

End If

' End If

'*****

```

```

'calculate Fuzzy depth
If DEF = 0 And stop1 = 0 Then
    error_angle2 = 0
End If
If error_depth >= 0 And error_depth < 20 Then
    yPZ = (1 / 20) * (error_depth) - 10 * -1
    yPL = (1 / 20) * error_depth
    PWM_M1 = (yPZ * go_speed4) + (yPL * 127.5)
    PWM_M2 = (yPZ * go_speed4) + (yPL * 127.5)
End If
If error_depth >= 20 And error_depth < 40 Then
    yPL = (1 / 20) * (error_depth - 20) * -1
    yPH = (1 / 20) * (error_depth - 10)
    PWM_M1 = (yPL * 127.5) + (yPH * 255)
    PWM_M2 = (yPL * 127.5) + (yPH * 255)
End If
If error_depth > 40 Then
    yPH = 1
    PWM_M1 = 255
    PWM_M2 = 255
End If
If error_depth <= 0 And error_depth >= -20 Then
    yNZ = (1 / 20) * (error_depth + 20)
    yNL = (1 / 20) * error_depth * -1
    PWM_M1 = (yNZ * go_speed4) * -1 + (yNL * 127.5) * -1
    PWM_M2 = (yNZ * go_speed4) * -1 + (yNL * 127.5) * -1
End If
If error_depth < -20 And error_depth >= -40 Then
    yNL = (1 / 20) * (error_depth - 20)
    yNH = (1 / 20) * (error_depth + 20) * -1
    PWM_M1 = (yNL * 127.5) * -1 + (yNH * 255) * -1
    PWM_M2 = (yNL * 127.5) * -1 + (yNH * 255) * -1

```

```

End If

If error_depth < -40 Then

    yNH = 1

    PWM_M1 = -255

    PWM_M2 = -255

End If

Text17.Text = PWM_M1

Text9.Text = PWM_M2

If PWM_M1 > 0 Then

    control_M1 = 253          '11111101

End If

If PWM_M1 < 0 Then

    control_M1 = 254          '11111110

    PWM_M1 = -1 * PWM_M1

End If

If PWM_M2 > 0 Then

    control_M2 = 251

End If

If PWM_M2 < 0 Then

    control_M2 = 247

    PWM_M2 = -1 * PWM_M2

End If

control1 = control_M1 And control_M2

control2 = 0

If stop1 = 0 Then

    DataAB = "x" + "000" + "000" + "X" + "0" + "y"

    Comm1.Output = DataCD

    Comm1.Output = DataAB

    Text16.Text = ">" & DataCD & " - " & DataAB

End If

End If ' end if

End If

```

ก.2 โปรแกรมในไมโครคอนโทรลเลอร์

ในโครงการนี้จะใช้ไมโครคอนโทรลเลอร์ตระกูล PIC โดยเลือกใช้เบอร์ PIC 16F877 ซึ่งนำมาใช้งานในโครงการนี้จำนวน 3 ตัว โดยที่ไมโครคอนโทรลเลอร์ PIC 1 และ PIC 2 ทั้ง 2 ตัวจะทำหน้าที่รับข้อมูลจากคอมพิวเตอร์และส่งสัญญาณไปขับมอเตอร์ ซึ่งรายละเอียดของวงจรได้กล่าวถึงแล้วในหัวข้อที่ 3.2.2 ส่วนของไมโครคอนโทรลเลอร์ PIC 3 ทำหน้าที่รับค่าจากอุปกรณ์วัดค่ามุมและอุปกรณ์วัดความดันแล้วส่งข้อมูลไปยังคอมพิวเตอร์ ซึ่งรายละเอียดของวงจรได้กล่าวถึงในหัวข้อ 3.2.3

การเขียนโปรแกรมไมโครคอนโทรลเลอร์ในโครงการนี้ใช้โปรแกรมภาษาซีในการสั่งงานไมโครคอนโทรลเลอร์ทั้ง 3 ตัว ซึ่งโปรแกรมหาดังนี้

ก.2.1 โปรแกรมสำหรับการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ของ PIC1 และ PIC2

```
#include <16f877a.h>

#define TxD    PIN_C6
#define RxD    PIN_C7
#define CLOCK_SP 20000000
#fuses hs
#fuses NOLVP, NOWDT
#fuses NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=TxD, rcv=RxD)
#bit    cren=0x18.4

int i;
int8 k,pwm[6],control,control1,control2;
int16 pwmz,pwmz;
char strz,data[10];
char pwm1,pwm2;
void main(void)
{
    for(k=0; k < 10; k++)
    {
        data[k] = 0;
    }
    while(true)
    {
```

```

while(true)
{
    cren=0;
    delay_us(10);
    cren=1;
    strz=getch();
    if (strz =='c')break;
}
for(k=1; k < 10; k++)
{
    data[0]='c';
    data[k]=getch();
}
if ((data[0]=='c')&&(data[9]=='d'))
{
    //pwm[1] = data[1] ;
    //pwm[2] = data[2] ;
    pwm[1] = data[1] * 100 + data[2] * 10 + data[3] * 1;
    pwm[2] = data[4] * 100 + data[5] * 10 + data[6] * 1;
    pwm[3]= data[7] ;
    pwm[4]= data[8] ;
    control1 = pwm[3];
    control2 = pwm[4];
    if(control1 == 'W')
    {
        output_b(0xFA);
    }
    if(control1 == 'S')
    {
        output_b(0xF5);
    }
    if(control1 == 'A')

```

```

{
    output_b(0xF6);
}
if(control1 == 'D')
{
    output_b(0xF9);
}
if(control1 == 'X')
{
    output_b(0xFF);
}
setup_ccp1(ccp_pwm);
setup_ccp2(ccp_pwm);
setup_timer_2(t2_div_by_16,150,1);
set_pwm1_duty(pwm[1]);
set_pwm2_duty(pwm[2]);
for(k=0; k<10; k++)
{
    data[k] = 0;
}
}
}
}

```

ก.2.2 โปรแกรมรับค่าและส่งข้อมูลจากอุปกรณ์วัดค่าทั้ง 2 ตัวไปยังคอมพิวเตอร์ของ PIC3

```

#include <16f877A.h>

#define TxD    PIN_C6
#define RxD    PIN_C7
#define CLOCK_SP 20000000
#define fuses HS
#define fuses NOLVP, NOWDT
#define fuses NOPROTECT

```

```

#device ADC=10

#use delay(clock=CLOCK_SP)

#use rs232(baud=9600, xmit=TxD, rcv=RxD)

#use fast_io(A)

#use fast_io(b)

#define Vbe 0.0048875855327468230694037145650049

int data[16];

int16 degree[16];

int d,i,f;

int16 data_real;

int16 data_degree;

#int_ccp1 //interrupt compass-module

void capture_re(){
d=1;
}

void main(void){
int16 value1,value2,value3; //start pressure
float volt;
setup_port_a(ALL_ANALOG);
setup_adc(ADC_CLOCK_INTERNAL);

do {
set_adc_channel(0);
delay_us(20);
value1=Read_ADC();

set_adc_channel(1);
delay_us(20);
value2=Read_ADC();

set_adc_channel(2);
delay_us(20);
value3=Read_ADC();

set_tris_b(0x3d);

d=0;

```

```

i=0;
data_degree=0;
data_real=0;
output_high(pin_b1);           //pin-P/C
output_low(pin_b1);
delay_ms(15);                   //0.05*delay .015 sec
output_high(pin_b1);
setup_ccp1(ccp_capture_re);
enable_interrupts(int_ccp1);
enable_interrupts(global);

while(true){
  if(d==1){
    data[i]=input(pin_b5);
    i++;
    d=0;
  }
  if(i>=16)
    break;
  }
  degree[6]=data[6]*200;         //data[6]=msb
  degree[7]=data[7]*100;
  degree[8]=data[8]*80;
  degree[9]=data[9]*40;
  degree[10]=data[10]*20;
  degree[11]=data[11]*10;
  degree[12]=data[12]*8;
  degree[13]=data[13]*4;
  degree[14]=data[14]*2;
  degree[15]=data[15]*1;        //data[15]=lsb
  for (f=6;f<=15;f++){
    data_degree=data_degree+degree[f];

```

```

data_real=data_degree;
}
for (f=6;f<=16;f++){
degree[f]=0;
data[f]=0;
}

//output_high (pin_d1);

printf("a%d ", data_real);           // Send Data of Angle
printf("b%d",value1);                //Send Data of depth Pin a0,Pin NO. 2
printf("c%d", "");
printf("e");
//output_low (pin_d1);
d=0;
i=0;
f=0;
data_degree=0;
data_real=0;                          //2*delay.015sec
delay_ms(15);
} while(TRUE);
}

```

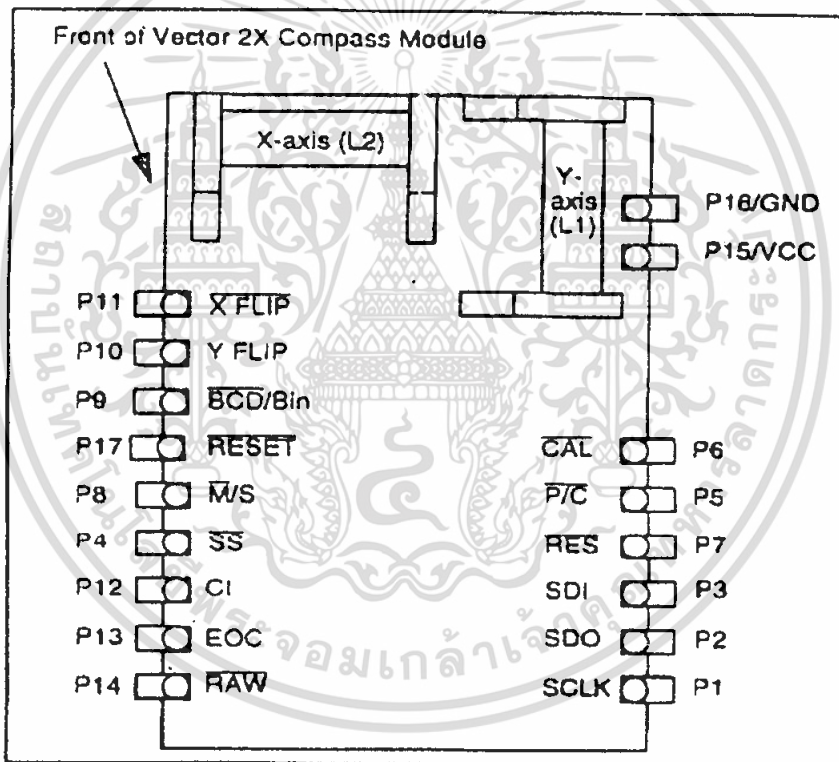
ภาคผนวก ข.

คู่มือประกอบการใช้งานอุปกรณ์ที่เกี่ยวข้องในโครงการ

ในส่วนของภาคผนวก-ข. จะอธิบายในรายละเอียดสำคัญของอุปกรณ์วัดค่ามุม อุปกรณ์วัดความดันและอุปกรณ์ที่ประกอบอยู่ภายในส่วนต่างๆของวงจรที่ได้สร้างขึ้นเพื่อควบคุมยานสำรวจได้นำ

ข.1 อุปกรณ์วัดค่ามุม

อุปกรณ์วัดค่ามุมนี้จะประกอบอยู่ในส่วนของวงจรรวมชุดเซนเซอร์ ซึ่งรูปของวงจรแสดงในรูปที่ 3.9 อุปกรณ์วัดค่ามุมที่นำมาใช้ในโครงการนี้เป็นรุ่น Vector 2X (V2X) ซึ่งผลิตโดยบริษัท Precision Navigation, Inc.



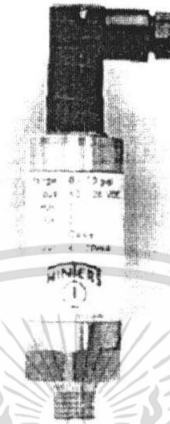
รูปที่ ข.1 ตำแหน่งขาต่างๆของอุปกรณ์วัดค่ามุม

Pin	Name	Description	Input or Output
P1	SCLK	Serial Clock Input: Data valid on rising clock edge%or *%or	Input/Output
P2	SDO	Serial Data Output	Output
P3	SDI	Serial Data Input	No Connect
P4	SS	Slave Select: Active Low	Input
P5	P/C	Poll / Continuous: Active when P/C = Low	Input
P6	CAL	Calibrate Select: Active when CAL = Low	Input
P7	RES	Resolution: Low Resolution when RES = Low	Input
P8	M/S	Master/Slave select: Master Mode when M/S = Low	Input
P9	BCD/Bin	BCD/Binary select: BCD when BCD/Bin = Low	Input
P10	Y FLIP	Flip Y-axis direction. Low = Normal Direction	Input
P11	X FLIP	Flip X-axis direction. High = Normal Direction	Input
P12	CI	Calibrate indicator signal: Active High	Output
P13	EOC	End of Conversion signal: Conversion complete on rising edge of EOC	Output
P14	RAW	Raw data mode select pin: Active Low	Input
P15	VCC	5 volt power input	
P16	GND	Power supply return	
P17	RESET	Reset pin: Active Low	Input

รูปที่ ๖.2 Vector 2X FUNCTION DESCRIPTION

ข.2 อุปกรณ์วัดความดัน

อุปกรณ์วัดความดันนี้จะประกอบอยู่ในส่วนของวงจรรวมชุดเซนเซอร์ ซึ่งรูปของวงจรแสดงในรูปที่ 3.9 ซึ่งในโครงการนี้ได้ใช้อุปกรณ์วัดความดัน รุ่น PT150 ผลิตโดยบริษัท Winters Instruments



รูปที่ ข.3 อุปกรณ์วัดความดัน

Specification:

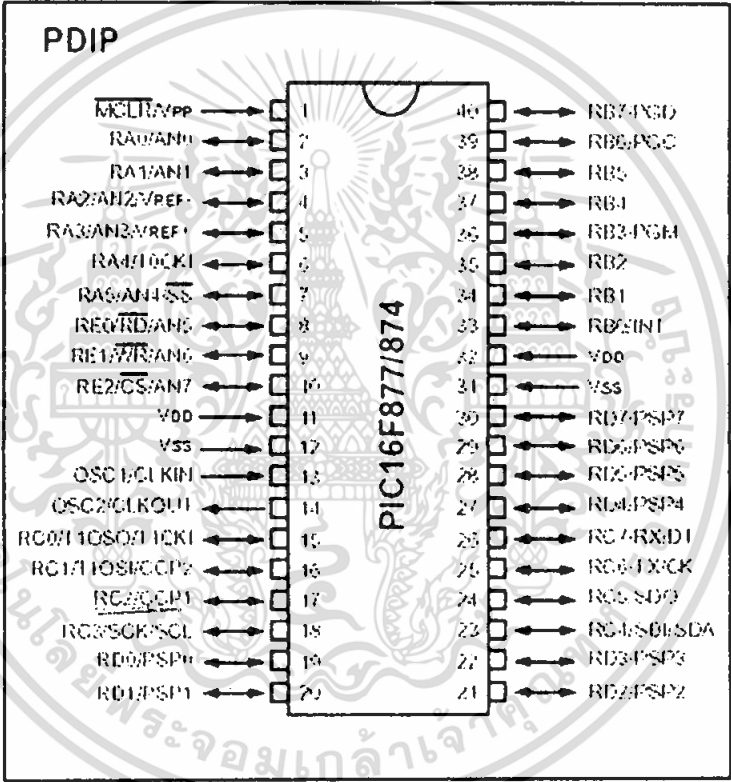
- Electrical Output Signal: 4-20mA, 2 wire (0.5-4.5Vdc, 0.5-5.5Vdc, 1-5Vdc 3 wire) available
- Electrical Connection: Mini DIN 43650 style (4pin) connection includes mate (other options available)
- Excitation Voltage: 8-38 Vdc Housing: 316 stainless steel
- Wetted Parts/Connection: 316 stainless steel, (1/4" npt male)
- Proof Pressure: 2X full scale
- Burst Pressure: 3X full scale
- Normal Operating Temp. Range: -30F to 200F (-35C to 100C)
- Compensated Temperature performance: 0F to 185F (-18C to 85C) ambient Temp. Effect on Zero: Less than +/-1.5% FSO per 100F (37.7C)
- Response Time: Less than and equal to 5 milliseconds
- Accuracy: $\pm 0.25\%$ FSO (+/-0.1% available)
- Long Term stability: Less than +/-0.25 FS per annum Weight: 10 oz. (283g)

ข.3 อุปกรณ์ต่างๆที่ประกอบในวงจร

เป็นอุปกรณ์ที่ใช้ในวงจรควบคุมยานสำรวจได้นำซึ่งจะประกอบอยู่ในส่วนต่างๆของวงจรเป็นไอซีและทรานซิสเตอร์

ข.3.1 ไอซี PIC16F877A

ไอซี PIC16F877A นี้จะประกอบอยู่ในส่วนของวงจรรวมชุดเซนเซอร์เป็นตัวควบคุมการรับส่งข้อมูลให้กับคอมพิวเตอร์และเป็นตัวควบคุมการทำงานของมอเตอร์ ซึ่งรูปของวงจรแสดงในรูปที่ 3.8 และรูปที่ 3.9 โดยขาต่างๆของ PIC16F877A แสดงไว้ดังรูปที่ ข.4



รูปที่ ข.4 ไอซี PIC16F877A

ข.3.2 ไอซี CNY17-1

เป็นไอซีออปโตไดโอดใช้สำหรับแยกกราวด์ระหว่างแรงดันไฟฟ้า 5V กับแรงดันไฟฟ้า 12V ออกจากกันเพื่อป้องกันสัญญาณรบกวนซึ่งจะประกอบอยู่ในส่วนของวงจรขยายสัญญาณ PWM ดังรูปของวงจรที่แสดงในรูปที่ 3.10

DESCRIPTION

The CNY17 series consists of a Gallium Arsenide IRED coupled with an NPN phototransistor.

FEATURES

- CNY17-1/2/3 are also available in white package by specifying -M suffix (e.g. CNY17-2-M).
- UL recognized (File # E90700).
- VDE recognized.
- 102497 for white package. -Add option V for white package (e.g., CNY17-2V-M).
- File #102497. -Add option 3(M) for black package (e.g., CNY17-2 3(M)).
- File #94798.
- Current transfer ratio in select groups.
- High BV_{CEO}—70V minimum.

APPLICATIONS

- Power supply regulators.
- Microprocessor inputs.
- Industrial controls.
- Digital logic inputs.
- Appliance sensor systems.

CNY17-1	CNY17-3
CNY17-2	CNY17-4

SCHEMATIC

WHITE PACKAGE (-M SUFFIX)

BLACK PACKAGE (NO -M SUFFIX)

Parameters	Symbol	Device	Value	Units
TOTAL DEVICE				
Storage Temperature	T _{stg}	All	-55 to +150	°C
Operating Temperature	T _{op}	All	-55 to +100	°C
Lead Solder Temperature	T _{sol}	All	260 for 10 sec.	°C
Total Device Power Dissipation @ 25°C (LED chip detection)	P _D	.M	250	mW
Derate Linearly From 25°C		non-.M	260	
		.M	2.64	mW/°C
		non-.M	3.50	
EMITTER				
Continuous Forward Current	I _F	.M	60	mA
		non-.M	90	
Reverse Voltage	V _R	All	8	V
Forward Current - Peak (1 us pulse, 300 pps)	I _{imp}	.M	1.5	A
		non-.M	3.0	
LED Power Dissipation 25°C Ambient				
Derate Linearly From 25°C	P _{LED}	.M	120	mW
		non-.M	135	
		.M	1.41	mW/°C
		non-.M	1.6	
		.M	150	mW
DETECTOR				
Detector Power Dissipation @ 25°C	P _D	.M	150	mW
		non-.M	200	
		.M	1.75	mW/°C
		non-.M	2.67	

รูปที่ ข.5 ไอซี CNY17-1

ข.3.3 ไอซี SN74LS125A

เป็นไอซีบัฟเฟอร์ใช้สำหรับตัดต่อสัญญาณการทำงานของสัญญาณ PWM และเป็นตัวตัดต่อการควบคุมการหมุนของมอเตอร์ซึ่งจะประกอบอยู่ในส่วนของวงจรขยายสัญญาณ PWM ดังรูปของวงจรที่แสดงในรูปที่ 3.10

SN74LS125A, SN74LS126A

Quad 3-State Buffers

LS125A

LS126A

TRUTH TABLES

INPUTS		OUTPUT
E	D	
L	-	L
L	H	H
H	X	(Z)

INPUTS		OUTPUT
E	O	
H	-	L
H	H	H
L	X	(Z)

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
(Z) = High Impedance (Hi-Z)

ON Semiconductor™
<http://onsemi.com>

LOW POWER SCHOTTKY

PLASTIC DIP SUFFIX CASE 648

SOIC D SUFFIX CASE 751A

SOEIAJ M SUFFIX CASE 965

ORDERING INFORMATION

Device	Package	Shipping
SN74LS125AN	14 Pin DIP	2000 Units/Box
SN74LS125AD	SOIC-14	55 Units/Rail
SN74LS125ADR2	SOIC-14	2500 Tape & Reel
SN74LS125AM	SOEIAJ-14	See Note 1
SN74LS125AMEL	SOEIAJ-14	See Note 1
SN74LS125AJ	14 Pin DIP	2000 Units/Box
SN74LS126AD	SOIC-14	55 Units/Rail
SN74LS126ADR2	SOIC-14	2500 Tape & Reel
SN74LS126AM	SOEIAJ-14	See Note 1
SN74LS126AMEL	SOEIAJ-14	See Note 1

* For ordering information on the FIAJ version of the SOIC package, please contact your local ON Semiconductor representative.

GUARANTEED OPERATING RANGES

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Supply Voltage	4.75	5.0	5.25	V
Ta	Operating Ambient Temperature Range	0	25	70	°C
I _{OH}	Output Current - High			2.4	mA
I _{OL}	Output Current - Low			24	mA

รูปที่ ข.6 ไอซี SN74LS125A

ข.3.4 ทรานซิสเตอร์ TIP122

เป็นทรานซิสเตอร์ทำหน้าที่ขยายกระแสให้สูงขึ้นซึ่งจะประกอบอยู่ในส่วนของ วงจรขยายสัญญาณ PWM ดังรูปของวงจรที่แสดงในรูปที่ 3.10

FAIRCHILD
SEMICONDUCTOR

TIP120/121/122

Medium Power Linear Switching Applications

Complementary to TIP125/126/127



TO-220

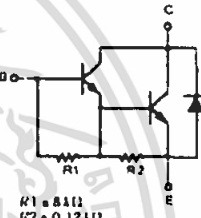
1 Base 2 Collector 3 Emitter

NPN Epitaxial Darlington Transistor

Absolute Maximum Ratings $T_C=25^{\circ}\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CE0}	Collector-Base Voltage	TIP120	60
		TIP121	80
		TIP122	100
V_{CE0}	Collector-Emitter Voltage	TIP120	60
		TIP121	80
		TIP122	100
V_{EB0}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	5	A
I_{CP}	Collector Current (Pulse)	8	A
I_B	Base Current (DC)	120	mA
P_C	Collector Dissipation ($T_J=25^{\circ}\text{C}$)	2	W
		65	W
T_J	Junction Temperature	150	$^{\circ}\text{C}$
T_{STG}	Storage Temperature	-65 - 150	$^{\circ}\text{C}$

Equivalent Circuit



$R1 = 81\Omega$
 $R2 = 0.171\Omega$

Electrical Characteristics $T_C=25^{\circ}\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
$V_{CE0(sat)}$	Collector-Emitter Sustaining Voltage	$I_C = 100\text{mA}$, $I_B = 0$	60		V
I_{CE0}	Collector Cut-off Current	TIP120		0.5	mA
		TIP121		0.5	mA
		TIP122		0.5	mA
I_{CBO}	Collector Cut-off Current	TIP120		0.2	mA
		TIP121		0.2	mA
		TIP122		0.2	mA
I_{EBO}	Emitter Cut-off Current	$V_{BE} = 5\text{V}$, $I_C = 0$		2	mA
h_{FE}	DC Current Gain	$V_{CE} = 3\text{V}$, $I_C = 0.5\text{A}$ $V_{CE} = 3\text{V}$, $I_C = 3\text{A}$	1000		
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = 3\text{A}$, $I_B = 12\text{mA}$ $I_C = 5\text{A}$, $I_B = 20\text{mA}$		2.0	V
$V_{BE(on)}$	Base-Emitter ON Voltage	$V_{CE} = 3\text{V}$, $I_C = 3\text{A}$		2.5	V
C_{ob}	Output Capacitance	$V_{CE} = 10\text{V}$, $I_C = 0.1$, $f = 0.1\text{MHz}$		200	pF

* Pulse Test, PW=100µs, Duty cycle ≤2%

รูปที่ ข.7 ทรานซิสเตอร์ TIP42C

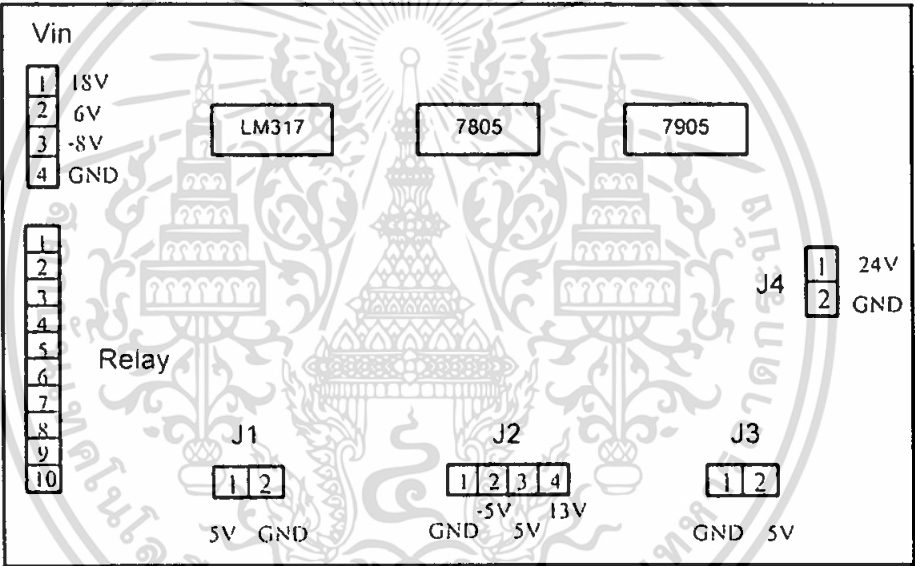
ภาคผนวก ค.

แผนผังอุปกรณ์และการเชื่อมต่อแผงวงจร

ในส่วนของภาคผนวก ค. จะประกอบด้วยแผนผังการวางอุปกรณ์แต่ละวงจรดังที่ได้กล่าวมาในบทที่ 3 และจุดเชื่อมต่อระหว่างแผงวงจรแต่ละแผงวงจร

ค.1 วงจรแหล่งจ่ายไฟฟ้า

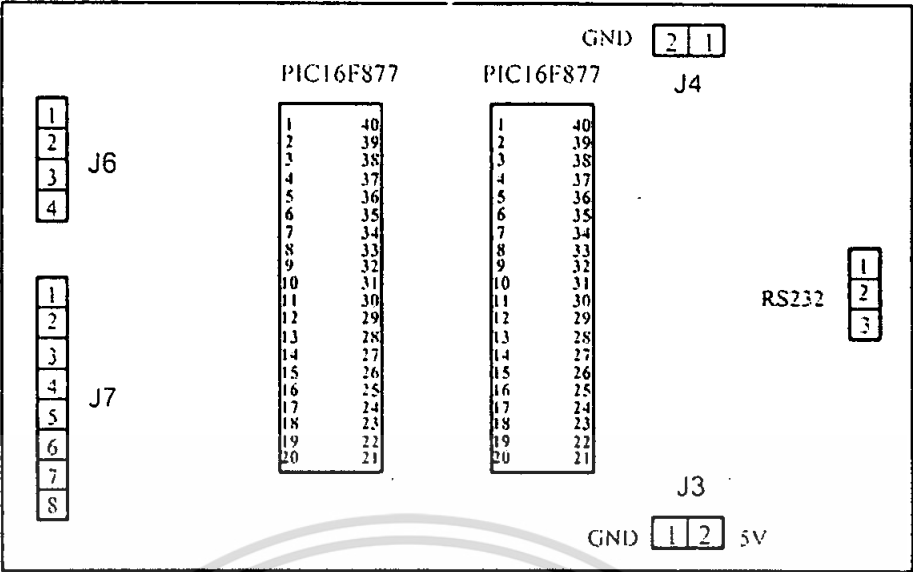
ในวงจรแหล่งจ่ายไฟฟ้านี้ จะประกอบด้วยจุดเชื่อมต่อทั้งหมด 6 จุด คือ J1, J2, J3, J4, Vin และ Relay ซึ่งรูปร่างวงจรแหล่งจ่ายไฟฟ้านี้จะแสดงอยู่ในรูปที่ 3.11



รูปที่ ค.1 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรแหล่งจ่ายไฟฟ้า

ค.2 วงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

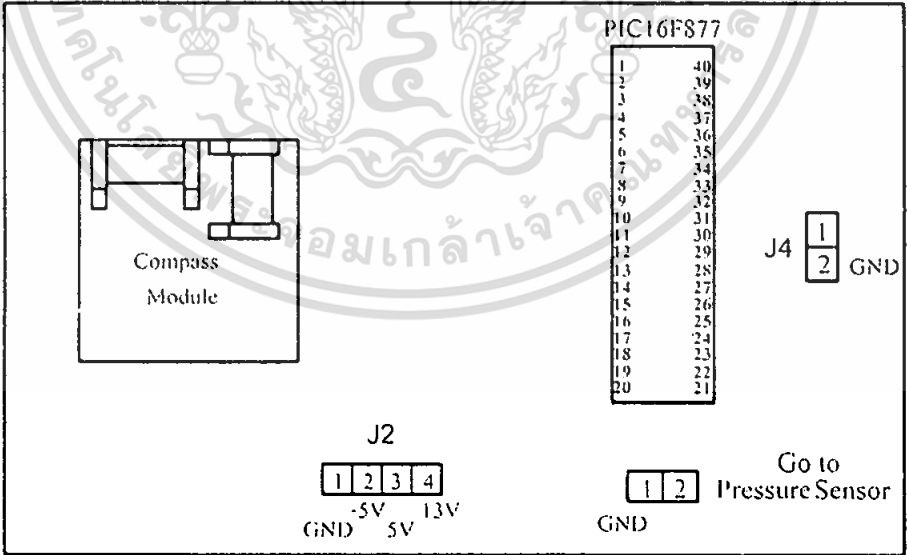
ในวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์นี้ ประกอบด้วยจุดเชื่อมต่อทั้งหมด 5 จุดคือ J3, J4, J6, J7 และ RS232 จุดเชื่อมต่อ RS232 จะนำไปเชื่อมต่อกับคอมพิวเตอร์ ซึ่งรูปร่างวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์นี้จะแสดงอยู่ในรูปที่ 3.8



รูปที่ ค.2 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

ค.3 วงจรรวมชุดเซนเซอร์

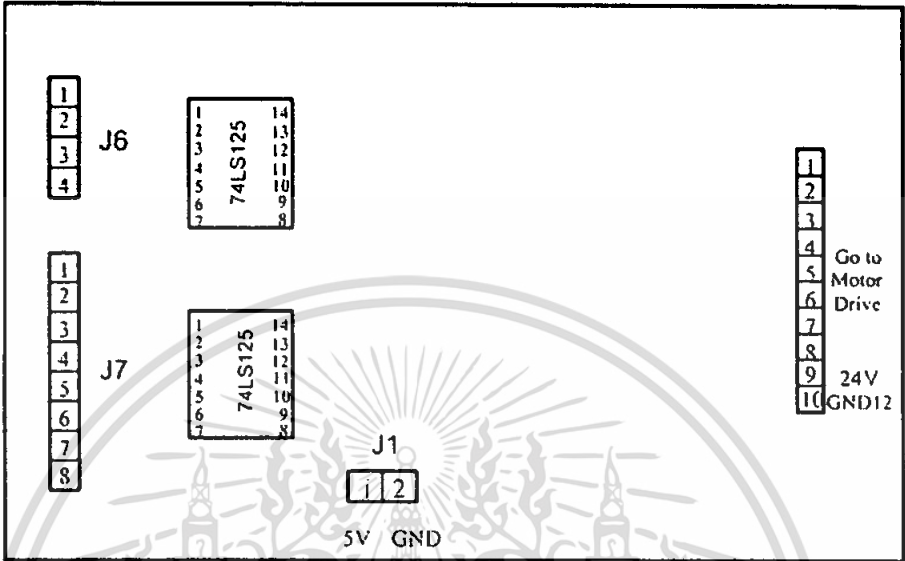
ในวงจรรวมชุดเซนเซอร์นี้ จะประกอบด้วยจุดเชื่อมต่อทั้งหมด 3 จุดคือ J2, J4 และจุดเชื่อมต่อกับอุปกรณ์วัดความดัน ซึ่งรูปวงจรรวมชุดเซนเซอร์นี้จะแสดงอยู่ในรูปที่ 3.9



รูปที่ ค.3 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรรวมชุดเซนเซอร์

ค.4 วงจรขยายสัญญาณ PWM

ในวงจรขยายสัญญาณ PWM นี้ จะประกอบด้วยจุดเชื่อมต่อทั้งหมด 4 จุดคือ J1, J6, J7n และจุดเชื่อมต่อกับวงจรขับเคลื่อนมอเตอร์ ซึ่งวงจรขยายสัญญาณ PWM นี้จะแสดงอยู่ในรูปที่ 3.10



รูปที่ ค.4 แผนผังอุปกรณ์และจุดเชื่อมต่อของวงจรขยายสัญญาณ PWM