สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เรือดำน้ำ
(ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล)
Remotely Operated Vehicle (ROV)

T050213

โดย

นายนที อุ่นทอง

นายองอาจ ศิริเงินยวง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

วน.
ท 152 ส
น25

เลขหมู่.....
เลขทะเบียน 50213
วัน,เดือน,ปี 27 มิ.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรือดำน้ำ
(ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล)
(Remotely Operated Vehicle: ROV)

โดย

นายนที อุ่นทอง รหัส 43015306

นายองอาจ ศิริเงินยวง รหัส 43015341

อาจารย์ที่ปรึกษา

ผศ. พรสุข รติโรจน์อนันต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์การศึกษา

ภาควิชา วิศวกรรมระบบควบคุม

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล

ผู้จัดทำ

1 นาย นที อุ่นทอง

2 นาย องอาจ ศิริเงินยวง

.....อาจารย์ที่ปรึกษา

(ผศ. พรสุข รติโรจอนันต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรือดำน้ำ
(ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล)
Remotely Operated Vehicle (ROV)

โดย
นายนที อุ่นทอง
นายองอาจ ศิริเงินยวง
อาจารย์ที่ปรึกษา
ผศ. พรสุข รติโรจอนันต์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอ ทฤษฎี การออกแบบ และการสร้างเรือดำน้ำขนาดเล็กหรือที่เรียกอีกอย่างหนึ่งว่า ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะ ไกล จุดประสงค์เพื่อศึกษาทฤษฎีการลอยตัวของวัตถุในน้ำ ตลอดจนการควบคุมการลอยตัวและการจมของวัตถุ และการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ แนวความคิดในการควบคุมการลอยตัวและการจมของเรือดำน้ำ อาศัยการเปลี่ยนแปลงปริมาตรของตัวเรือ โดยการควบคุมปริมาณการไหลเข้าและไหลออกของน้ำในถังอับเฉา ส่งผลให้น้ำหนักของตัวเรือเปลี่ยนแปลงไป ทำให้สามารถควบคุมการลอยและการจมของเรือดำน้ำได้ และใช้กล้อง CCD ติดตั้งตรงด้านหน้าของตัวเรือ เพื่อเก็บภาพใต้น้ำ เรือดำน้ำที่สร้างขึ้นมาได้ผ่านการทดสอบประสิทธิภาพในการเคลื่อนที่ทั้งแนวดิ่ง แนวราบ ความเร็วในการเคลื่อนที่ และความสามารถในการแสดงทัศนะวิสัยใต้น้ำ

Abstract

This thesis presents the theory, design, and builds the small submarines that can be called the remotely operated vehicle (ROV). The objective is to learning the buoyancy theory, control the floating and diving of the object, and microcontroller application. The control method for the floating and diving of the ROV is based on the variation of the ballast tank capacity to change the weigh of the ROV. Also the CCD camera is used for

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the view finder of the ROV. All the testing and controlling procedure shows the good performance of the constructed ROV in moving direction, moving speed, and view under water.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงได้ด้วยดี เพราะได้ความเมตตาจากผู้ช่วยศาสตราจารย์ พรสุข รติโรจน์อนันต์ ที่ได้กรุณาแนะนำและคอยให้คำปรึกษา รวมทั้งจัดหาอุปกรณ์ที่เป็นประโยชน์ต่อโครงงานนี้ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคูณ อาจารย์ ถาวร เบญจนราสุทธ์ ที่เอื้อเฟื้อกล้องถ่ายภาพ และ Video capture card และให้คำปรึกษาที่เป็นประโยชน์ต่อโครงงาน

ขอขอบพระคุณ สถาบันฯ ที่ให้ความอนุเคราะห์ให้ใช้สระว่ายน้ำของสถาบันฯ ในการใช้ในการทดลอง และขออบพระคุณ อาจารย์ เกียรติวรรณ ทรงสัตย์ ที่ให้ความกรุณาเป็นธุระในการติดต่องานเรื่องขอใช้สระว่ายน้ำของสถาบันฯ เพื่อใช้ในการทดลอง และขอขอบพระคุณ คุณ ณรงค์ ป้อมศรี ผู้ดูแลสระว่ายน้ำที่ได้กรุณามาดูแลระหว่างการทดลองทุกครั้ง

ขอขอบพระคุณ ภาควิชาวิศวกรรมระบบควบคุม ที่เอื้อเฟื้อสถานที่ อุปกรณ์ และเครื่องมือทั้งหมดที่เป็นประโยชน์ต่อการทำตังถังเรือ

ขอขอบพระคุณเพื่อนๆ ทุกคน ที่คอยเป็นกำลังใจที่ดีเป็นเรี่ยวแรงย่อยที่สำคัญ รวบทั้งเอื้อเฟื้ออุปกรณ์ที่ขาดเหลือมาโดยตลอด

สุดท้ายที่ลืมมิได้คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่เคารพรักยิ่งตลอดจนญาติผู้ใหญ่ และพี่ๆ น้องๆ ที่ให้กาสนับสนุน และเป็นกำลังใจที่ดีตลอดการทำโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

กิตติกรรมประกาศ

สารบัญ

สารบัญภาพ

บทที่ 1 บทนำ 1

1.1 ประวัติความเป็นมาของโครงงาน 1

1.2 ขอบเขตการศึกษา 2

บทที่ 2 ทฤษฎีและหลักการ 3

2.1 หลักการทำงานของเรือดำน้ำ 3

2.2 ถังอับเฉา 4

2.2.1 ถังอับเฉาแบบช่องปลายเปิด (Vented ballast tank) 5

2.2.2 ถังอับเฉาแบบยืดหยุ่น (Flexible ballast tank) 5

2.2.3 ถังอับเฉาแบบแรงดันสูง(Presser ballast tank) 6

2.2.4 ถังอับเฉาแบบลูกสูบ (Piston ballast tank) 7

2.2.5 ถังอับเฉาแบบแผ่นเมมเบรน (Membrane ballast tank) 7

2.2.6 ถังอับเฉาแบบ Bellow (Bellow ballast tank) 8

2.2.7 ถังอับเฉาแบบใช้ก๊าซ (Gas operated ballast tank) 8

2.2.8 ถังอับเฉาแบบใช้คอมเพรสเซอร์ (Compressed air ballast tank) 9

2.3 สสาร 10

2.4 ความดันและความลึก 13

2.5 การลอยตัว 15

2.6 เสถียรภาพของการลอยตัว 18

2.7 ไมโครคอนโทรลเลอร์ตระกูล PIC16F87x 19

2.7.1 สถาปัตยกรรมและโครงสร้างทางฮาร์ดแวร์ของ PIC16F873 19

2.7.2 โครงสร้างของไมโครคอนโทรลเลอร์ PIC16F873 21

2.8 คุณสมบัติทางเทคนิคของ PIC16F873 21

2.9 รีจิสเตอร์กำหนดการทำงานของไมโครคอนโทรลเลอร์ PIC 23

2.10 โหมดของสัญญาณนาฬิกา 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.11 การสื่อสารแบบอนุกรม	27
2.12 การสื่อสารข้อมูลแบบอะซิงโครนัส	28
2.13 คอนเน็กเตอร์สำหรับพอร์ต RS – 232 และการเชื่อมต่อ	30
2.14 หน้าที่การทำงานแต่ละขาของพอร์ตอนุกรม RS-232	32
2.15 UART	32
2.16 วงจรภายในและรีจีสเตอร์ของพอร์อนุกรม RS-232	33
2.17 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232	40
2.18 แอดเดรสของพอร์ตอนุกรม	41
2.19 การเขียนโปรแกรมใช้งานพอร์ตอนุกรม	42
2.19.1 การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรม	42
2.19.2 การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม	43
2.20 คอนโทรล MSComm	45
บทที่ 3 การสร้างแบบออกแบบ	57
3.1 ตัวถัง	57
3.2 วงจรไมโครคอนโทรลเลอร์และส่วนควบคุม	63
3.3 Flowchart แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์	66
3.4 โปรแกรมควบคุมเรือดำน้ำ	71
บทที่ 4 ผลการทดลองและสรุป	72
การทดลองครั้งที่ 1 ทดลองการจมของตัวถัง เรือดำน้ำ	72
การทดลองครั้งที่ 2 ทดลองการสั่งงานมอเตอร์ผ่านคอมพิวเตอร์	72
การทดลองครั้งที่ 3 แก้ไขความเสียหายของ Regulator	73
การทดลองครั้งที่ 4 การทดลองการสั่งงานโซลินอยด์วาล์วผ่านทางคอมพิวเตอร์	73
การทดลองครั้งที่ 5 การทดลองจับภาพโดยใช้โปรแกรม Visual Basic	73
การทดลองครั้งที่ 6 การทดลองวัดความดันน้ำตามระดับความลึกต่างๆ กัน	74
การทดลองครั้งที่ 7 ทดลองนำเรือดำน้ำ ลงดำในสระว่ายน้ำเพื่อควบคุมทิศทางการเคลื่อนที่สังเกต ความสมดุล ของตัวเรือ	75
การทดลองที่ 8 การทดลองการลอยตัว, การจม, และการเคลื่อนที่ใต้ผิวน้ำ	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
การทดลองที่ 9 การควบคุมระดับความลึก	75
การทดลองที่ 10 การวัดความเร็ว	76
บทที่ 5 บทวิจารณ์และสรุป	77
5.1 สรุปผลการทดลอง	77
5.2 ปัญหาที่พบและแนวทางการแก้ไข	77
5.3 แนวทางในการค้นคว้าและพัฒนา	77
หนังสืออ้างอิง	78
ภาคผนวก ก	
ภาคผนวก ข	
ภาคผนวก ค	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 แสดงการลอยและการจมของเรือ 3

รูปที่ 2.2 แสดงหลักการทำงานของถังอับเฉาขณะที่กำลังจม 4

รูปที่ 2.3 แสดงหลักการทำงานของถังอับเฉาขณะที่กำลังจม 4

รูปที่ 2.4 แสดงส่วนประกอบของถังอับเฉาแบบช่องปลายเปิด (Vented ballast tank) 5

รูปที่ 2.5 แสดงส่วนประกอบของถังอับเฉาแบบยืดหยุ่น (Flexible ballast tank) 6

รูปที่ 2.6 แสดงส่วนประกอบของถังอับเฉาแบบแรงดันสูง (Presser ballast tank) 6

รูปที่ 2.7 แสดงส่วนประกอบของถังอับเฉาแบบลูกสูบ (Piston ballast tank) 7

รูปที่ 2.8 แสดงส่วนประกอบของถังอับเฉาแบบแผ่นเมมเบรน (Membrane ballast tank) 8

รูปที่ 2.9 แสดงส่วนประกอบของถังอับเฉาแบบ Bellow (Bellow ballast tank) 8

รูปที่ 2.10 แสดงส่วนประกอบของถังอับเฉาแบบใช้ก๊าซ (Gas operated ballast tank) 9

รูปที่ 2.11 แสดงส่วนประกอบของถังอับเฉาแบบใช้คอมเพรสเซอร์ (Compressed air ballast tank) 10

รูปที่ 2.12 ความดันและความลึก 13

รูปที่ 2.13 การลอยตัว 16

รูปที่ 2.14 การลอยตัว และเสถียรภาพ 18

รูปที่ 2.15 ไดอะแกรมการทำงานพื้นฐานของไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมแบบฮาร์วาร์ด 19

รูปที่ 2.16 แสดงไดอะแกรมของกระบวนการไปป์ไลน์ที่ใช้ในไมโครคอนโทรลเลอร์ PIC 20

รูปที่ 2.17 แสดงไดอะแกรมของกระบวนการไปป์ไลน์เมื่อกระทำคำสั่งการกระโดด 20

รูปที่ 2.18 แสดงโหมดการทำงานของสัญญาณนาฬิกากรณีใช้คริสตอลและเซรามิก 26

รูปที่ 2.19 แสดงโหมดการทำงานของสัญญาณนาฬิกากรณีต่อตัวต้านทานพูดดาวน์ภายนอกที่ขา RA7/OSC1/CLKIN 27

รูปที่ 2.20 แบบอย่างง่ายที่สุดของข้อมูลอนุกรม 27

รูปที่ 2.21 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 28

รูปที่ 2.22 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบDB-9 & DB-25 30

รูปที่ 2.23 แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.1 แสดงโครงสร้างของตัวเรือหลัก 59

รูปที่ 3.2 แสดงโครงสร้างของตัวเรือด้านข้าง 60

รูปที่ 3.3 แสดงโครงสร้างของตัวเรือเมื่อนำมาประกอบเข้าด้วยกัน 61

รูปที่ 3.4 แสดงชุดใบพัดที่ใช้ในการขับเคลื่อน (Propeller) 62

รูปที่ 3.5 แสดงส่วนที่ใช้ในการถ่วงน้ำหนัก ภายในประกอบด้วยแท่งเหล็กที่
สามารถเลื่อนไปมาได้ 62

รูปที่ 3.6 แสดงตัวเรือเมื่อประกอบเสร็จสมบูรณ์ 63

รูปที่ 3.7 บล็อกไดอะแกรมแสดงการทำงานของวงจรควบคุม 64

รูปที่ 3.8 วงจรควบคุม 65

รูปที่ 3.9 Flowchart แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์ 66

รูปที่ 3.10 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)67

รูปที่ 3.11 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)68

รูปที่ 3.12 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)69

รูปที่ 3.13 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)70

รูปที่ 3.14 โปรแกรมควบคุมเรือดำน้ำ 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันความต้องการที่จะทำการสำรวจพื้นที่ใต้ทะเล หรือ แม่น้ำมีปริมาณเพิ่มมากขึ้น เพื่อตอบสนองความต้องการในด้านต่าง ๆ ทั้งในปัจจุบันและอนาคต เช่น การสำรวจแหล่งพลังงานธรรมชาติ การสำรวจและซ่อมบำรุงแนวท่อก๊าซธรรมชาติและสายเคเบิลใยแก้วนำแสง การสำรวจทางโบราณคดี การสำรวจทางธรณีวิทยา เป็นต้น ซึ่งกรณีศึกษาเหล่านี้จะต้องปฏิบัติการภายใต้สภาวะแวดล้อมที่มีความเสี่ยงสูง จึงจำเป็นที่จะต้องใช้หุ่นยนต์หรือยานพาหนะอื่นปฏิบัติการแทนมนุษย์ซึ่งในปัจจุบันมีการใช้ยานพาหนะในการปฏิบัติภารกิจต่าง ๆ เหล่านี้ คือ เรือดำน้ำ หรือ ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล (Remotely Operated Vehicle: ROV) ซึ่งในรายงานฉบับนี้จะกล่าวถึง ROV เป็นหลัก

ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล (Remotely Operated Vehicle : ROV) เป็นเรือดำน้ำที่มีขนาดเล็ก ควบคุมการทำงานผ่านทางสายเคเบิล โดยผู้ควบคุมจะอยู่บนผิวน้ำ หรือพื้นดิน อุปกรณ์พื้นฐานที่จำเป็นสำหรับ ROV ได้แก่ ระบบควบคุมการเคลื่อนที่ใต้น้ำขับเคลื่อนด้วยมอเตอร์ กล้องวีดีโอขนาดเล็ก แหล่งจ่ายพลังงาน(Power Supply) ซึ่งอาจอยู่ภายในตัว ROV หรือรับพลังงานผ่านทางสายเคเบิลจากระบบควบคุมบนผิวน้ำ สัญญาณควบคุม และข้อมูลจาก Sensor ต่าง ๆ จะถูกส่งผ่านทางสายเคเบิลเช่นเดียวกัน ROV อาจมีขนาดเล็กเท่าโทรทัศน์สำหรับงานที่ไม่ยุ่งยากมากนัก ไปจนถึงมีขนาดใหญ่หลายเมตรสำหรับงานที่มีความซับซ้อนสูง ซึ่งจะต้องใช้ผู้ควบคุมที่มีความชำนาญสูง

ภารกิจหลักของ ROV คือ การให้การสนับสนุนการสำรวจและพัฒนาแหล่งน้ำมัน การวางแนวท่อหรือแนวเส้นเคเบิลใยแก้วนำแสงใต้น้ำ ที่มีความลึกและความยุ่งยากซับซ้อนมาก ๆ

1.1 ประวัติความเป็นมาของโครงงาน

ในปัจจุบันประเทศไทยมีความต้องการใช้งาน ROV เป็นจำนวนมากแต่ไม่สามารถผลิตขึ้นได้เองภายในประเทศยังจำเป็นต้องนำเข้าจากต่างประเทศ ซึ่งมีราคาสูงมาก และจากการที่ต้องนำเข้า ROV จากต่างประเทศทำให้ในบางกรณีการทำงานของ ROV อาจไม่เหมาะสมกับภูมิประเทศของประเทศไทย และบุคลากรอาจไม่มีความรู้ความเข้าใจในการใช้งานอุปกรณ์ต่าง ๆ ทำให้ไม่สามารถใช้งานอุปกรณ์ต่าง ๆ ได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางคณะผู้จัดทำได้เล็งเห็นถึงความจำเป็นดังกล่าว จึงได้ทำการศึกษาถึงความเป็นไปได้ที่จะสร้างแบบจำลอง ROV ขึ้น เพื่อศึกษาการทำงาน ตลอดจนการทดลองใช้งานจริงในสภาวะแวดล้อมที่จำลองขึ้น ซึ่งในการศึกษาครั้งนี้ทางคณะผู้จัดทำได้ใช้ไมโครคอนโทรลเลอร์ ตระกูล PIC ในการควบคุมอุปกรณ์ต่าง ๆ ภายใน ROV ผ่านทางสายเคเบิล โดยใช้คอมพิวเตอร์เป็นตัวสั่งการและแสดงผล

1.2 ขอบเขตการศึกษา

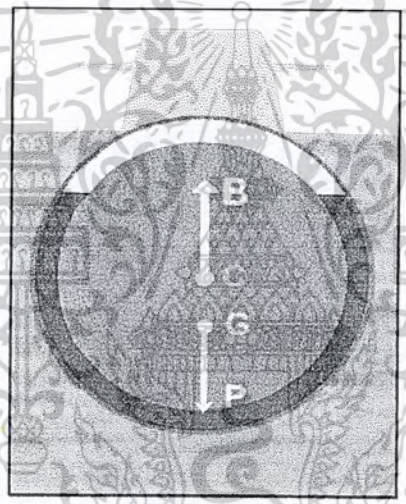
1. ศึกษาการใช้คอมพิวเตอร์ควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่านทางพอร์ตอนุกรม
2. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ ตระกูล PIC
3. ศึกษาการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก
4. ออกแบบวงจรควบคุมมอเตอร์ และ โซลินอยด์วาล์ว
5. ออกแบบและสร้าง ROV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 หลักการทำงานของเรือดำน้ำ

เรือดำน้ำ คือเรือที่สามารถลอยอยู่บนผิวน้ำหรือจมลงสู่ใต้น้ำได้ โดยเราจะอาศัยคุณสมบัติของการเปลี่ยนแปลงน้ำหนักของตัวเรือให้เพิ่มขึ้นหรือลดลง ซึ่งวิธีการดังกล่าวมีหลายแบบด้วยกัน ซึ่งในแต่ละแบบก็จะมีข้อดีข้อเสียแตกต่างกันออกไป นั้นคือเมื่อตัวเรือมีน้ำหนักเพิ่มมากขึ้นส่งผลให้ความสามารถในการลอยตัวของเรือลดลงเป็นลบ (Negative buoyancy) ส่งผลในเรือจมลง และเมื่อเราลดน้ำหนักของตัวเรือให้ลดลงจะส่งผลให้ความสามารถในการลอยตัวเพิ่มมากขึ้นเป็นบวก (positive buoyancy) ส่งผลให้เรือสามารถลอยขึ้นสู่ผิวน้ำได้



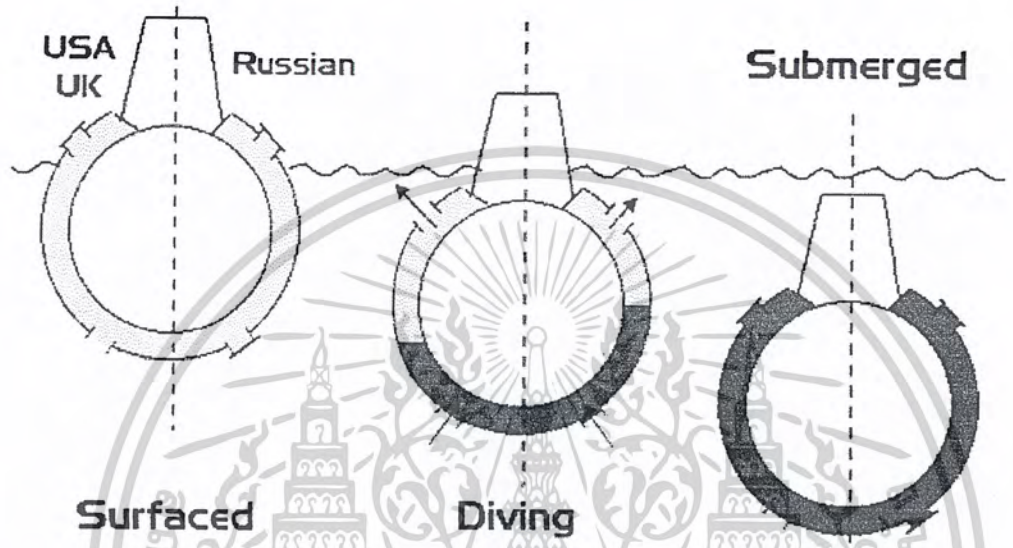
รูปที่ 2.1 แสดงการลอยและการจมของเรือ

จากรูปที่ 2.1 แรงที่กระทำต่อเรือ ณ จุด G (center of gravity) โดยแรงดึงดูดของโลก ในทิศทางตรงข้ามกันจะมีแรง B โดยแรง B เป็นแรงยกของของเหลว (buoyancy) กระทำ ณ จุด C เมื่อแรง B มีขนากเปลี่ยนแปลงไปตามระดับความลึกและสัมประสิทธิ์ของของเหลว เราสามารถเปลี่ยนแปลงแรง P ให้มีค่าเพิ่มมากขึ้นได้โดยการเพิ่มน้ำหนักให้กับตัวเรือ นั่นคือ การให้น้ำไหลเข้าไปในบางส่วนของตัวเรือ ซึ่งจะทำให้น้ำหนักของตัวเรือเพิ่มมากขึ้นส่งผลให้แรง P เพิ่มมากขึ้นด้วย ทำให้เรือสามารถจมลงสู่ใต้น้ำได้ และในทางตรงกันข้ามเมื่อเราต้องการให้เรือลอยขึ้นเราก็จะทำการลดน้ำหนักของตัวเรือโดยการสูบน้ำออกจากตัวเรือซึ่งจะทำให้น้ำหนักของตัวเรือลดลงส่งผลให้แรง P มีค่าลดลง โดยที่พื้นที่ที่เราทำการปล่อยน้ำเข้าหรือสูบน้ำออกซึ่งอยู่ภายในตัวถังเรือนี้เราเรียกว่า ถังอับเฉา (Ballast tank)

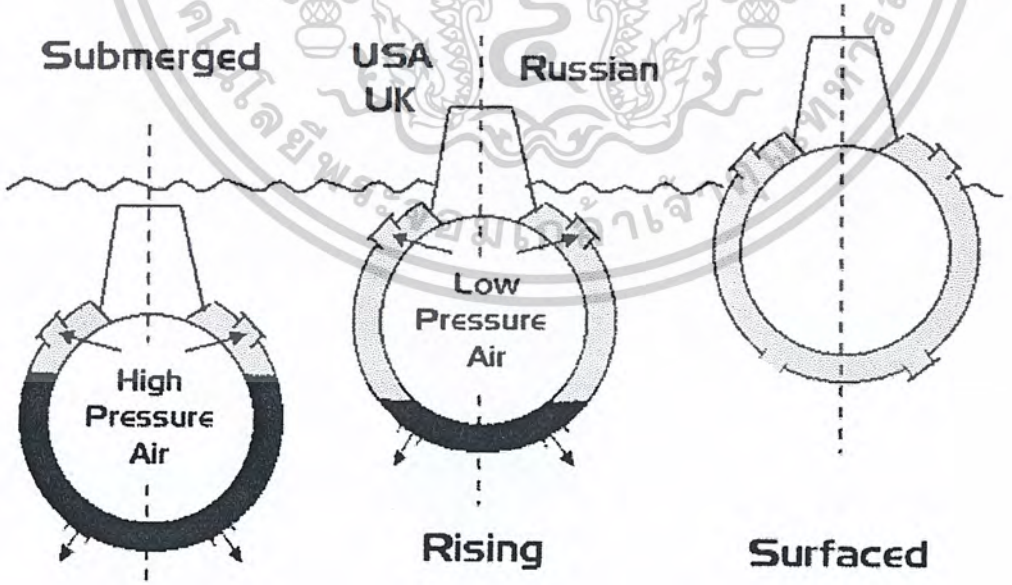
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ถังอับเฉา

ถังอับเฉาคือ พื้นที่ภายในตัวถังเรือดำน้ำหรือยานสำรวจใต้น้ำซึ่งจะถูกจัดไว้เป็นพื้นที่ว่างๆ เพื่อเป็นที่ที่เราใช้สูบน้ำเข้าและออกเพื่อเปลี่ยนแปลงน้ำหนักของตัวเรือให้ตัวเรือสามารถจมหรือลอยได้ โดยวิธีการสูบน้ำเข้าและสูบน้ำออกจากถังอับเฉามีหลายวิธีด้วยกัน แบ่งเป็นชนิดต่างๆ ของถังอับเฉา เช่น แบบลูกสูบ แบบใช้ปั๊มน้ำ แบบใช้ก๊าซ เป็นต้น



รูปที่ 2.2 แสดงหลักการทำงานของถังอับเฉาขณะที่กำลังจม

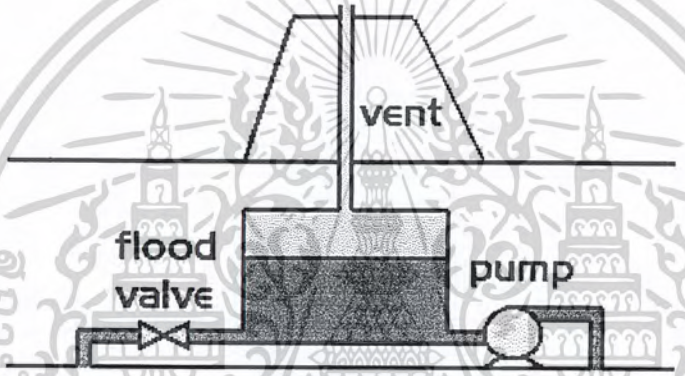


รูปที่ 2.3 แสดงหลักการทำงานของถังอับเฉาขณะที่กำลังลอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ถังอับเฉาแบบช่องปลายเปิด (Vented ballast tank)

ถังอับเฉาแบบนี้สามารถช่วยลดแรงลอยตัวของเรือ เมื่อเปิดวาล์วน้ำ (flood valve) อากาศจะไหลออกทางช่องทางออกและน้ำก็จะไหลเข้าสู่ถังอับเฉา ถ้าจะทำให้ภายในถังว่างเปล่า ทำได้โดยการปั๊มน้ำออกจากถังขณะที่อากาศถูกดูดกลับเข้าสู่ถังผ่านทางท่อ การทำงานของระบบนี้บริเวณส่วนบนสุดของท่อต้องอยู่เหนือระดับน้ำซึ่งเป็นเหตุผลที่ทำให้เรือที่ใช้ถังอับเฉาแบบนี้ไม่สามารถจมลงสู่ใต้น้ำในระดับลึกๆ ได้ แต่เรือสามารถแล่นไปตามผิวน้ำด้วยความเร็วสูง การใช้ปั๊มแบบ 2 ทิศทางเราไม่จำเป็นต้องใช้วาล์วน้ำ (flood valve) ก็ได้ และเพื่อเป็นการป้องกันไม่ให้น้ำไหลเข้าสู่ถังอับเฉาขณะที่มีการขับเคลื่อนใต้น้ำ เส้นผ่านศูนย์กลางของเส้นท่อจะต้องมีขนาดเล็ก ดังนั้นถังอับเฉาแบบนี้จึงไม่เหมาะสมที่จะนำมาใช้

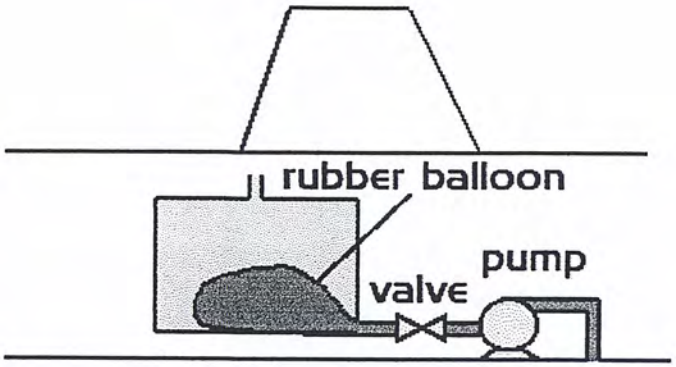


รูปที่ 2.4 แสดงส่วนประกอบของถังอับเฉาแบบช่องปลายเปิด (Vented ballast tank)

2.2.2 ถังอับเฉาแบบยืดหยุ่น (Flexible ballast tank)

ถังอับเฉาแบบนี้ประกอบด้วยถุงที่ทำด้วยยางติดตั้งตายตัวอยู่ภายในถัง ถังจะจมลงเมื่อเปิดวาล์วและปั๊มน้ำเข้าสู่ถัง ปิดวาล์วเพื่อป้องกันน้ำไหลออกขณะจม อากาศที่มีอยู่แล้วภายในถังจะไหลผ่านท่อไปยังภายในตัวเรือ (Pressure hull) ทำให้ความดันภายในตัวเรือเพิ่มขึ้น ถ้าปริมาตรของถังอับเฉาไม่มากเพียงพอกับปริมาตรของอากาศภายในตัวเรือก็ไม่เป็นปัญหาใดๆ ภายในเรือดำน้ำจะประกอบด้วยอุปกรณ์ที่ป้องกันไม่ให้ปริมาตรของอากาศมีค่าเท่ากับปริมาตรภายในตัวเรือ (hull)

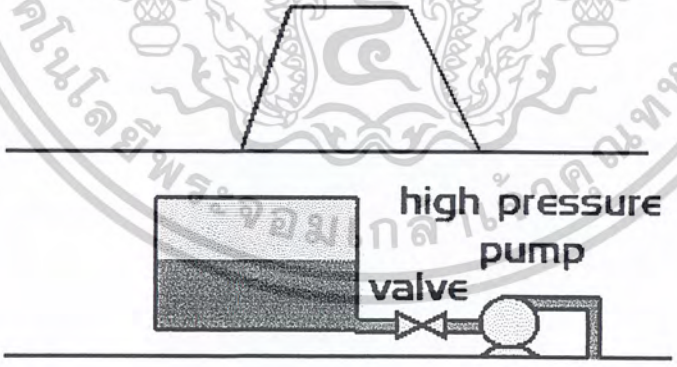
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงส่วนประกอบของถังอับเฉาแบบยืดหยุ่น (Flexible ballast tank)

2.2.3 ถังอับเฉาแบบแรงดันสูง(Pressure ballast tank)

ถังอับเฉาแบบนี้จะถูกปิดตาย ค่าความดันจะมีค่าเพิ่มขึ้นอย่างมีนัยสำคัญ (5 Bar หรือมากกว่า) เมื่อเราต้องการให้เรือจมเราก็จะปั๊มน้ำเข้าสู่ถังอับเฉาด้วยปั๊มน้ำความดันสูง เพราะว่าอากาศภายในถังอับเฉาไม่สามารถเล็ดลอดออกสู่ภายนอกได้ และเมื่อต้องการให้เรือลอยขึ้นเราก็จะปั๊มน้ำออก ความดันที่เพิ่มขึ้นภายในถังอับเฉาจะมีผลทำให้เราไม่สามารถปั๊มน้ำเข้าสู่ถังอับเฉาจนเต็มได้เนื่องจากอากาศภายในถังอับเฉาต้องการที่อยู่ ตัวอย่างเช่น เมื่อความดันสูงสุดภายในถังอับเฉาเท่ากับ 5 bar เราสามารถปั๊มน้ำเข้าสู่ถังอับเฉาได้เพียง 80 เปอร์เซ็นต์ของปริมาตรทั้งหมดภายในถัง

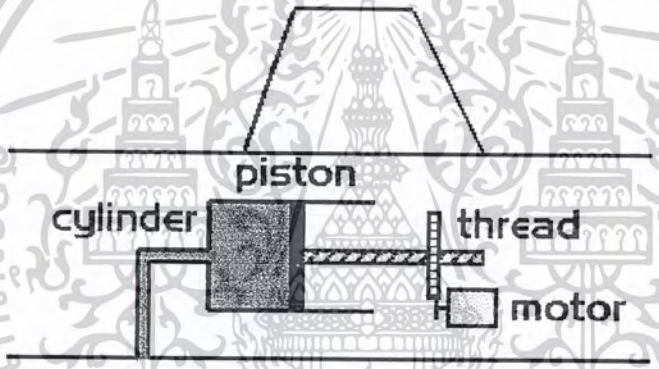


รูปที่ 2.6 แสดงส่วนประกอบของถังอับเฉาแบบแรงดันสูง(Presser ballast tank)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 ถังอับเฉาแบบลูกสูบ (Piston ballast tank)

ถังอับเฉาแบบนี้ประกอบด้วยลูกสูบทรงกระบอกที่สามารถเคลื่อนที่ได้คล้ายกระบอกฉีดยาขนาดใหญ่ ลูกสูบเคลื่อนที่ได้ด้วยแกนเฟืองและมอเตอร์เล็กๆ ปลายเปิดของทรงกระบอกจะต่อโดยตรงกับน้ำที่อยู่รอบๆ ถังอับเฉาแบบนี้จะไม่มีอากาศอยู่ภายในถังอับเฉาเหมือนถังอับเฉาแบบอื่น ความดันภายในตัวเรือจะเพิ่มมากขึ้นถ้าลูกสูบของถังอับเฉาสูบน้ำเข้าจนเต็ม ถ้ามีการตรวจวัดตำแหน่งของกระบอกสูบ เช่น ตรวจวัดอุปกรณ์วัดระดับความดันไฟฟ้าที่ต่ออยู่กับแกนเฟืองทำให้ค่าการลอยตัวของเรือสามารถปรับให้มีความเหมาะสมและมีความแม่นยำสูงได้ ลูกสูบของถังอับเฉาแบบนี้มีความยาวมาก ดังนั้นถังอับเฉาแบบนี้จึงเหมาะที่จะอยู่ในแนวนอน แต่ในระหว่างที่มีการสูบน้ำเข้า-ออกภายในถังอับเฉาจะส่งผลกระทบต่อศูนย์ถ่วงของเรือซึ่งอุปสรรคนี้สามารถแก้ไขได้โดยการใช้ถังอับเฉาแบบลูกสูบ 2 ตัวโดยติดตั้งไว้บริเวณด้านหน้าและด้านหลังของตัวเรือเพื่อลดปัญหาที่เกิดผลกระทบต่อศูนย์ถ่วงของตัวเรือ

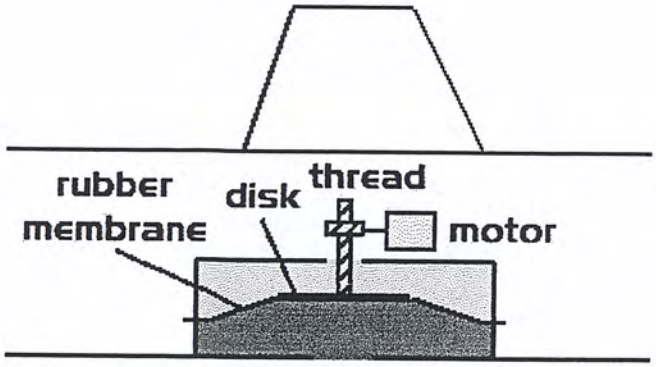


รูปที่ 2.7 แสดงส่วนประกอบของถังอับเฉาแบบลูกสูบ (Piston ballast tank)

2.2.5 ถังอับเฉาแบบแผ่นเมมเบรน (Membrane ballast tank)

ถังอับเฉาแบบนี้เป็นถังอับเฉาแบบง่ายๆ ประกอบด้วยแผ่นดิสก์ที่สามารถเคลื่อนที่ขึ้น-ลงด้วยแกนเฟืองที่ติดกับมอเตอร์คล้ายกับถังอับเฉาแบบลูกสูบ แผ่นดิสก์นี้จะต่ออยู่กับลูกสูบทรงกระบอกและแผ่นยางบางๆ ที่ยืดหยุ่นได้ เมื่อแผ่นดิสก์เลื่อนขึ้นทำให้น้ำสามารถไหลเข้าสู่ตัวเรือได้ ลักษณะที่ดีของถังอับเฉาแบบนี้คือ น้ำจะถูกกักเก็บภายในถังได้ง่าย แผ่นยางที่เชื่อมต่อกับแผ่นดิสก์และถังอับเฉาจะต้องมีความยาวที่เหมาะสมและไม่เกิดการรั่ว ข้อเสียของถังอับเฉาแบบนี้คือ ระยะลูกสูบมีความยาวไม่มากดังนั้นค่าการลอยตัวของเรือจึงเปลี่ยนแปลงได้ไม่มากนัก

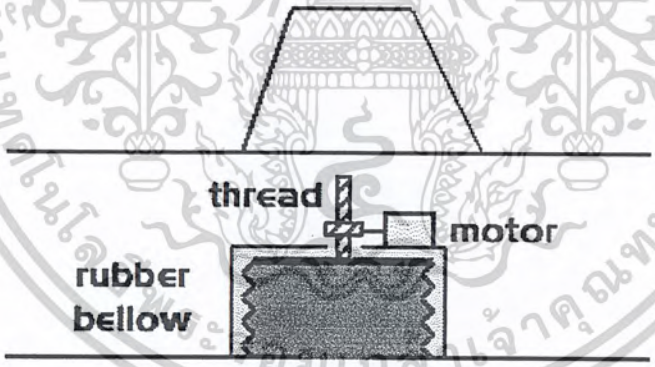
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงส่วนประกอบของถังอับเฉาแบบแผ่นเมมเบรน (Membrane ballast tank)

2.2.6 ถังอับเฉาแบบ Bellow (Bellow ballast tank)

ถังอับเฉาแบบนี้เป็นอีกรูปแบบหนึ่งของถังอับเฉาแบบเมมเบรนโดยแทนที่จะใช้แผ่นยางบางเราจะใช้แผ่นยางแบบ Bellow แทน ซึ่งมีลักษณะคล้ายกับสปริงที่มีพื้นที่ว่างภายในและสามารถยืดหดได้โดยการยืดหดของยางแบบ bellow นี้จะทำให้น้ำไหลเข้า-ออกได้ ข้อดีของถังอับเฉาแบบนี้คือ ระยะในการเคลื่อนที่มีค่าเพิ่มมากขึ้นจึงสามารถสูบน้ำเข้าสู่ภายในตัวเรือได้มากขึ้น แผ่นยาง bellow มีเส้นผ่านศูนย์กลางประมาณ 5-10 เซนติเมตรหรือมากกว่า



รูปที่ 2.9 แสดงส่วนประกอบของถังอับเฉาแบบ Bellow (Bellow ballast tank)

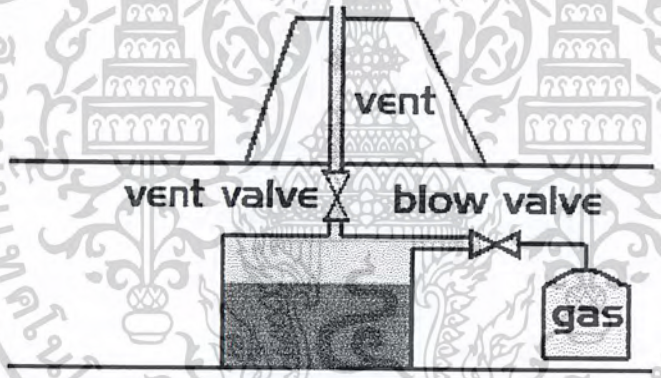
2.2.7 ถังอับเฉาแบบใช้ก๊าซ (Gas operated ballast tank)

ในระบบนี้ประกอบด้วยกระบอกแรงดันอัดก๊าซ ถังอับเฉาและวาล์วจำนวน 2 ตัว ระบบนี้จะคล้ายคลึงกับระบบการถ่วงน้ำหนักของเรือดำน้ำมาก โดยที่น้ำจะสามารถเข้าสู่ถังอับเฉาได้เมื่อวาล์วตัวบนถูกเปิดน้ำจะไหลเข้าสู่ถังผ่านช่องเปิดทางด้านล่าง ถ้าต้องการให้ปริมาตรของน้ำในถังอับเฉาลดลงวาล์วตัวบนจะต้องถูกปิดและน้ำในถังสามารถไหลออกได้เมื่อมีแรงดันของก๊าซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไปในถังอับเฉาโดยการเปิดวาล์วตัวล่าง รูปแบบนี้เรือที่มีการเป่าก๊าซเข้าสู่ถังอับเฉาได้นั้น ปริมาณก๊าซที่สะสมไว้จะต้องเพียงพอ มีการเลือกใช้ก๊าซคาร์บอนไดออกไซด์ที่บรรจุในถังรูปทรงกระบอก เพราะถังและก๊าซชนิดนี้มีราคาถูกสามารถหาได้ทั่วไป ถ้าใช้ก๊าซคาร์บอนไดออกไซด์ วาล์วจะถูกปรับลงให้ความดันมีค่าประมาณ 2-3 Bar ที่นำมาใช้ส่วนใหญ่จะอยู่ในรูปก๊าซเหลวซึ่งสามารถเก็บไว้ได้ในปริมาณมาก ง่ายต่อการเติมลงถัง ข้อดีของการใช้ก๊าซเหลวคือทำให้แรงดันภายในมีค่าประมาณ 3-4 bar ดังนั้นจึงไม่จำเป็นต้องใช้อุปกรณ์ช่วยในการลดแรงดันวาล์วที่ใช้ในถังอับเฉา เป็นวาล์วมาตรฐานที่นิยมในห้องทดลอง ซึ่งสามารถป้องกันการสูญเสียพลังงานได้ดี วาล์วที่ปิดง่ายจะถูกนำมาใช้ การใช้คาร์บอนไดออกไซด์กับวาล์วลดแรงดันหรือ ก๊าซเหลวนั้น จะทำให้แรงดันที่ค้างอยู่ในถังมีประมาณ 5 bars

วาล์วที่เปิดให้อากาศออกจากถังอับเฉา ทำให้แรงดันมีความแตกต่างกันระหว่างอากาศในถังและอากาศที่ล้อมรอบตัวถัง ดังนั้นจึงต้องเปิดวาล์วให้อากาศไหลเข้าอย่างเต็มที่ เพราะเมื่อวาล์วถูกปิดแล้ว แรงดันที่แตกต่างกันเพียงเล็กน้อยก็สามารถจมเรือลงได้ ระลึกไว้ว่าวาล์วที่เปิดต้องน้อยกว่า 1 มิลลิเมตร และไม่ทำให้น้ำไหลเข้าเร็วเกินไป ซึ่งทำได้ไม่ยากนัก

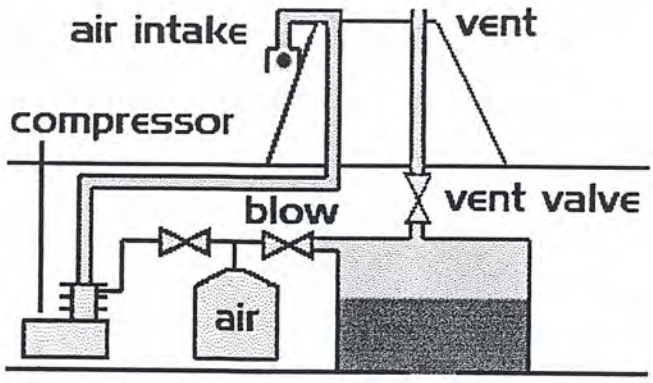


รูปที่ 2.10 แสดงส่วนประกอบของถังอับเฉาแบบใช้ก๊าซ (Gas operated ballast tank)

2.2.8 ถังอับเฉาแบบใช้คอมเพรสเซอร์ (Compressed air ballast tank)

ถังอับเฉาแบบนี้ใช้ Compressor air เหมือนกับถังอับเฉาที่ใช้ในเรือดำน้ำจริง ระบบถังอับเฉาแบบนี้คล้ายกับระบบถังอับเฉาแบบก๊าซ คอมเพรสเซอร์สามารถเพิ่มความดันอากาศได้ประมาณ 6-8 bar เป็นความดันที่สูงมากพอที่จะทำให้เกิดความเสียหายได้จึงต้องใช้ความระมัดระวังสูง ที่ความดันต่ำๆ ก๊าซสามารถถูกเก็บไว้ได้ดี โดยทั่วไปเรือจะมีแผงควบคุม air compressor เพื่อทำการเติมอากาศตลอดเวลาที่เรือขับเคลื่อนอยู่บนผิวน้ำและเมื่อเรือจมลงเราจะต้องไม่ให้น้ำซึมเข้าสู่คอมเพรสเซอร์โดยการปิดวาล์วทางเข้าของอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงส่วนประกอบของถังอับเฉาแบบใช้คอมเพรสเซอร์ (Compressed air ballast tank)

2.3 สสาร

ในหัวข้อนี้เราจะกล่าวถึงสมบัติของสสาร โดยจะเน้นถึงสมบัติที่จะเป็นประโยชน์แก่การพิจารณางานโครงงานชิ้นนี้ ความดันและความหนาแน่นเป็นปริมาณที่เรากล่าวถึงมากที่สุด จึงควรทำความเข้าใจให้ถ่องแท้ สิ่งสำคัญของสาระในบทนี้คือ การพิจารณาสมบัติของสสารในสภาวะนิ่งและสภาวะเคลื่อนที่

สสาร(Matter) เป็นสิ่งหนึ่งที่ถือได้ว่าเป็นรูปหนึ่งของพลังงาน (energy) โดยมีมวล (mass) และต้องการที่อยู่ในปริภูมิ (space) โดยทั่วไปเรามักกล่าวกันว่าสสารมี 3 สถานะคือ ของแข็ง (solid) ของเหลว (liquid) และก๊าซ (gas) โดยเรียกของเหลวและก๊าซรวมกันว่าของไหล (fluid)

อนึ่ง เราพบว่าสารบางชนิดจะมีโอกาสเป็นไปได้ทั้ง 3 สถานะ กล่าวคือ ของแข็ง ของเหลว หรือ ก๊าซ ภายใต้อุณหภูมิและค่าความดันที่เหมาะสม และจะปรากฏว่า ของแข็งส่วนมากเมื่อหลอมละลายเป็นของเหลว ก็จะมีความหนาแน่นลดลง และเมื่อของเหลวเปลี่ยนสถานะเป็นก๊าซก็ยิ่งจะมีความหนาแน่นลดลงมากด้วย แสดงว่า ระยะระหว่างโมเลกุลเพิ่มขึ้นเมื่อของแข็งเปลี่ยนสถานะเป็นของเหลวและระยะดังกล่าวจะเพิ่มมากขึ้นเมื่อของเหลวเปลี่ยนสภาพเป็นกลาง สำหรับความหนาแน่นจำนวนโมเลกุล (molecular number-density) ของสารต่างๆ ก็จะมีความสอดคล้องกับสถานะของสสารดังกล่าวนั้น เช่น ทองคำ น้ำและอากาศ มีความหนาแน่นจำนวนโมเลกุลประมาณ 0.6×10^{29} , 3.3×10^{28} และ 2.7×10^{26} โมเลกุลต่อลูกบาศก์เมตรตามลำดับ

ความหนาแน่นและความดัน

โดยทั่วไปเรากล่าวถึงความหนาแน่น (Density " ρ ") ของสารว่าคือ อัตราส่วนระหว่างมวล (mass "m") กับปริมาตร (volume "v") ของสารนั้น

กล่าวคือ $$\rho = \frac{m}{v} \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหนาแน่นของสารจะมีหน่วยในระบบเอสไอเป็นกิโลกรัมต่อลูกบาศก์เมตร อย่างไรก็ตามเพื่อความถูกต้องอย่างสมบูรณ์เราน่าจะเรียกความหนาแน่นของสารนี้ว่า ความหนาแน่นมวลเชิงปริมาตร (volume mass density) ทั้งนี้เพราะยังปรากฏว่ามีความหนาแน่นมวลในลักษณะอื่นๆ อีก เช่น ความหนาแน่นเชิงผิว (surface mass density "σ") เป็นกิโลกรัมต่อตารางเมตร และความหนาแน่นเชิงเส้น (linear mass density "μ") เป็นกิโลกรัมต่อเมตรอีก อย่างไรก็ตามในกรณีนี้เราจะกล่าวเพียงแต่ว่า ความหนาแน่น นั่นคือเราจะหมายถึงความหนาแน่นมวลเชิงปริมาตรเสมอ ในกรณีที่ต้องการเปรียบเทียบค่าความหนาแน่นของสารกับค่าความหนาแน่นของน้ำ ณ อุณหภูมิ 4 องศาเซลเซียสซึ่งเป็นขณะที่น้ำมีความหนาแน่นสูงสุด เท่ากับ 10^3 กิโลกรัมต่อลูกบาศก์เมตร เราจะเรียกคำดังกล่าวนี้ว่า ความถ่วงจำเพาะ (specific gravity "s") ของสารนั้น ตัวอย่างเช่น อะลูมิเนียมและปรอทมีความหนาแน่นเท่ากับ 2.7×10^3 กิโลกรัมต่อลูกบาศก์เมตร และ 13.6×10^3 กิโลกรัมต่อลูกบาศก์เมตร ดังนั้น จึงมีค่าความถ่วงจำเพาะเป็น 2.7 และ 13.6 ตามลำดับ โดยที่ค่าความถ่วงจำเพาะของสารต่างๆ จะทำให้เกิดมโนภาพเกี่ยวกับสภาวะการลอยหรือจมของสารต่างๆ เหล่านั้นในน้ำได้เป็นอย่างดี ซึ่งโดยทั่วไปของแข็งจะมีความหนาแน่นมากกว่าของเหลวและของเหลวจะมีความหนาแน่นมากกว่าก๊าซ

สาร	ρ (กิโลกรัมต่อลูกบาศก์เมตร)
น้ำแข็ง	0.92×10^3
อะลูมิเนียม	2.70×10^3
เหล็ก	7.86×10^3
ทองแดง	8.92×10^3
เงิน	10.05×10^3
ทองคำ	19.3×10^3
น้ำ	1.0×10^3
ปรอท	13.6×10^3
อากาศ	1.29
ออกซิเจน	1.43

ตารางที่ 2.1 แสดงค่าความหนาแน่นของสารตัวอย่าง

นอกจากความหนาแน่นตามที่กล่าวมาแล้ว ความดัน (Pressure "p") ก็เป็นปริมาณหลักที่สำคัญอีกอย่างหนึ่งของสสาร โดยหมายถึงอัตราส่วนระหว่างแรงดัน (force of pressure "F") และพื้นที่ตั้งฉาก (normal area "A") กับแรงดันนั้น ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$p = \frac{F}{A} \tag{2.2}$$

สังเกตได้ว่าแรงดัน (F) เป็นเวกเตอร์ ความดัน (P) จึงต้องเป็นเวกเตอร์ที่มีทิศทางเดียวกันจะต่างกันเฉพาะขนาดเท่านั้น ด้วยเหตุนี้เพื่อความถูกต้องจึงควรกล่าวถึงความดันในลักษณะเวกเตอร์ ถ้าต้องการบอกแต่เพียงขนาดที่สื่อความหมายเพียงเชิงสเกลาร์ก็ควรกล่าวว่า "ขนาดความดัน" ถ้าคุ้นเคยกับการกล่าวถึงคำว่า "ความดัน" ที่มีความหมายเชิงสเกลาร์ ก็อาจใช้เวกเตอร์ความดัน (vector of pressure) เพื่อแสดงความหมายเชิงเวกเตอร์

เนื่องจากในระบบเอสไอ แรงมีหน่วยเป็นนิวตัน และพื้นที่มีหน่วยเป็นตารางเมตร ความดันของเราจึงมีหน่วยเป็นนิวตันต่อตารางเมตร นอกจากนี้ก็ยังมีหน่วยเป็นอย่างอื่นอีก เช่น พาสคัล (Pascal "Pa") บาร์ (bar) และบรรยากาศ (atmosphere) โดยมีความสัมพันธ์กับหน่วยนิวตันต่อตารางเมตรดังนี้

1 พาสคัล	เท่ากับ	1 นิวตันต่อตารางเมตร
1 บาร์	เท่ากับ	10^5 นิวตันต่อตารางเมตร
1 บรรยากาศ	เท่ากับ	1.013×10^5 นิวตันต่อตารางเมตร

เมื่อวัตถุอยู่ในของไหลหรือจมอยู่ในของเหลว ของเหลวจะออกแรงกดตั้งฉากกับผิววัตถุ เราจึงสามารถใช้สมการ (2.2) เพื่อหาค่าความดันที่ของเหลวกระทำต่อวัตถุได้โดยตรง อย่างไรก็ตามยังมีความสำคัญประการหนึ่งที่ต้องระวังคือ ค่าความดันในของเหลวหรือในของไหล ณ ตำแหน่งต่างๆ กันไม่จำเป็นต้องเท่ากัน การกำหนดค่าความดัน ณ ตำแหน่งหนึ่งในของเหลวจึงต้องพิจารณาโดยใช้กฎเกณฑ์ทางคณิตศาสตร์ประกอบ กล่าวคือใช้ขีดจำกัดมีค่าเข้าสู่ศูนย์ของชิ้นประกอบผิว (surface element) ของพื้นที่ ΔA ที่มีแรงขนาด ΔF กระทำตั้งฉากทั่วพื้นที่ของชิ้นประกอบผิวนั้น จะได้

$$p = \lim_{\Delta A \to 0} \frac{\Delta F}{\Delta A} = \frac{dF}{dA} \tag{2.3}$$

จากการศึกษาต่อไปเราพบว่าค่าความดันในของเหลวจะเพิ่มขึ้นตามความลึก

2.4 ความดันและความลึก

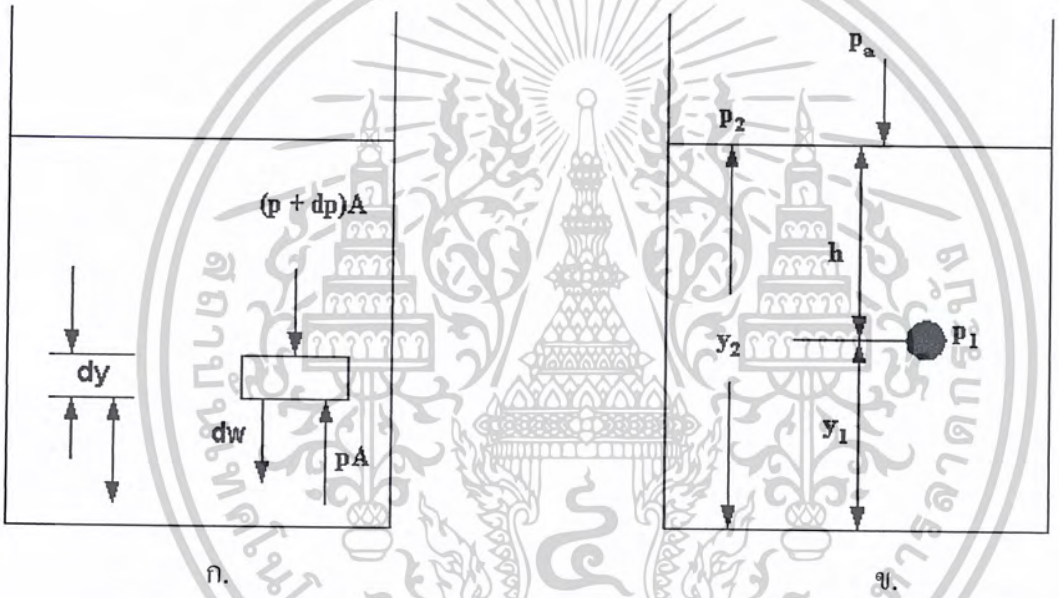
โลกมีบรรยากาศห่อหุ้ม บรรยากาศจึงห่อหุ้มตัวเราอยู่ตลอดเวลา ในขณะเดียวกันอากาศก็มีแรงกดหรือมีความดันต่อเราอยู่ตลอดเวลา แต่เนื่องจากความเคยชิน เราจึงไม่รู้สึกต่อความดันของอากาศที่กดเรา ตรงกันข้ามถ้าความดันที่บรรยากาศกระทำต่อเราเปลี่ยนแปลง เช่น เมื่อเราอยู่ ณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานที่สูงจากระดับน้ำทะเลมากๆ หรือเมื่อเราอยู่ในสภาวะที่เปลี่ยนความสูงโดยฉับพลัน เราย่อมรับรู้ต่อการเปลี่ยนแปลงนั้น เราเรียกค่าความดันของบรรยากาศซึ่งมีผลต่อทุกสิ่งบนโลก ไม่เพียงแต่ตัวเราเท่านั้น ว่า ค่าความดันบรรยากาศ (Atmospheric pressure) โดยใช้สัญลักษณ์เป็น p_a

ในหัวข้อนี้ เราจะพิจารณาค่าความดันของของเหลว หรือค่าความดันเกจ (Gauge pressure "p_g") ตำแหน่งหนึ่ง ที่สมพันธ์กับความลึก (depth "h") ของของเหลวที่ตำแหน่งนั้น ตลอดจนค่าความดันสมบูรณ์ (absolute pressure "p") ของของเหลว ซึ่งคือ ผลรวมระหว่างค่าความดันบรรยากาศและค่าความดันเกจของของเหลว ณ ตำแหน่งความลึก (h) นั้น
กล่าวคือ

$$p = p_a + p_g \qquad (2.4)$$



รูปที่ 2.12 ความดันและความลึก

สมมติพิจารนาชิ้นประกอบของเหลวชิ้นหนึ่ง ซึ่งมีพื้นที่หน้าตัด A สูง dy อยู่สูงจากระดับเทียบ y มีแรงกระทำขึ้นที่ผิวล่าง และแรงกระทำลงที่ผิวบนเป็น pA กับ $(p+dp)A$ ตามลำดับ

ให้ $dV = A\,(dy)$ แทนปริมาตรของชิ้นประกอบของเหลวชิ้นนี้

ρ แทนความหนาแน่นของของเหลว

g แทนค่าความโน้มถ่วงของโลก

และ $dw = \rho\, g\,(dV) = \rho\, g\, A(dy)$ แทนค่าน้ำหนักของชิ้นประกอบของเหลวดังกล่าวดังแสดงในรูปที่ 2.12 ก

เนื่องจากชิ้นประกอบของเหลวอยู่ในสภาวะสมดุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ $pA-(p+dp)A-dw=0$

หรือ $pA-(p+dp)A-\rho g A(dy)=0$

ดังนั้น $\frac{dp}{dy} = -\rho g h$ (2.5)

จากสมการ ข้างต้น เนื่องจากทั้ง ρ และ g เป็นบวก ถ้า dy เป็นบวก dp จะเป็นลบ แสดงว่า ค่าความดันในของเหลวจะลดลง ขณะที่ตำแหน่งในของเหลวจะสูงขึ้น

รูปที่ 2.12 ข สมมติตำแหน่งที่อยู่สูงจากระดับเทียบ y_1 หรืออยู่ลึก h ของเหลวมีค่าความดัน (p) คือ p_1 ส่วนตำแหน่งที่ผิวหน้าของเหลวพอดี ซึ่งอยู่สูงจากระดับเทียบ y_2 มีค่าความดัน p_2 ซึ่งก็คือความดันบรรยากาศ (p_a)
พิจารนาการหาปริพันธ์ของสมการดังนี้

$$\int_{p_1}^{p_2} dp = -\rho g \int_{y_1}^{y_2} dy$$

จะได้ $p_2 - p_1 = -\rho g (y_2 - y_1) = -\rho g h$

หรือ $p = p_a + \rho g h$ (2.6)

กล่าวคือ ค่าความดันสมบูรณ์ (p) ของของเหลวในที่เปิด ณ ตำแหน่งลึก h เท่ากับผลรวมของค่าความดันบรรยากาศ (p_a) และค่าความดันเนื่องจากของเหลวลึก h นั้น

อนึ่ง เมื่อเปรียบเทียบสมการ (2.6) กับความจริงตามสมการ (2.4)แสดงว่าค่าความดันเนื่องจากของเหลวลึก h ก็คือความดันเกจ (p_a) ของของเหลวซึ่งมีขนาดน้ำหนัก w นั้นเอง

กล่าวคือ $p_a = \rho g h$ (2.7)

ข้อควรสังเกต

1. $p_g = \rho g h$ มีหน่วยเป็น นิวตันต่อ(เมตร)^{2}ตรงตามหน่วยของความดันในระบบเอสไอ
2. ณ ที่หนึ่งซึ่ง g มีค่าคงตัว จะได้ $p_a \propto \rho h$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สำหรับของเหลวชนิดหนึ่งซึ่งที่ค่า ρ คงตัว อยู่ในบริเวณหนึ่งที่มีค่า g คงตัว จะได้ $p_a \propto h$ กล่าวคือ ค่าความดันเกจเพิ่มขึ้นตามความลึกของของเหลว หรือกล่าวโดยกลับกันได้ว่า ณ ระดับความลึกเดียวกันในของเหลวชนิดหนึ่ง ค่าความดันเกจ (p_a) ตลอดจนค่าความดันสมบูรณ์ (p) ของของเหลว ย่อมเป็นค่าคงตัว

4. หน่วยของ $\frac{pg}{\rho g}$ คือ เมตร ซึ่งเป็นหน่วยของระยะทางที่สอดคล้องกับหน่วยของความลึกของของเหลว h

ด้วยเหตุนี้ ในบางกรณีโดยเฉพาะเมื่อต้องการบอกค่าความดับบรรยากาศ จะนิยมบอกเป็นค่าความลึกหรือส่วนสูงของของเหลวแทน กล่าวคือ $h = \frac{pg}{\rho g}$ นั้นเอง เช่น ความดันบรรยากาศปกติมีค่าเท่ากับ 0.76 เมตรของปรอท โดยมีความหมายว่า บรรยากาศปกติจะมีความดันที่มีค่าเท่ากับค่าความดันเนื่องจากปรอทสูง (h) เท่ากับ 0.76 เมตร

อนึ่ง ความดันเป็นปริมาณทางฟิสิกส์ที่มีสมบัติพิเศษอย่างหนึ่ง โดย แบลส์ พาสคัส (Blaise Pascal "พ.ศ. 2166-2205) เป็นผู้พบความจริงนี้ และรู้จักกันในนามกฎของพาสคัส (pascal's law) ดังนี้

"ถ้าเพิ่มความดันเกจให้แก่ของไหลในภาชนะใบหนึ่ง จะมีความดันส่งต่อกันในของไหลโดยทำให้ทุกจุดในของไหล และผนังของภาชนะใบนั้น มีค่าความดันเพิ่มขึ้นเท่ากับค่าความดันที่ให้นั้น"

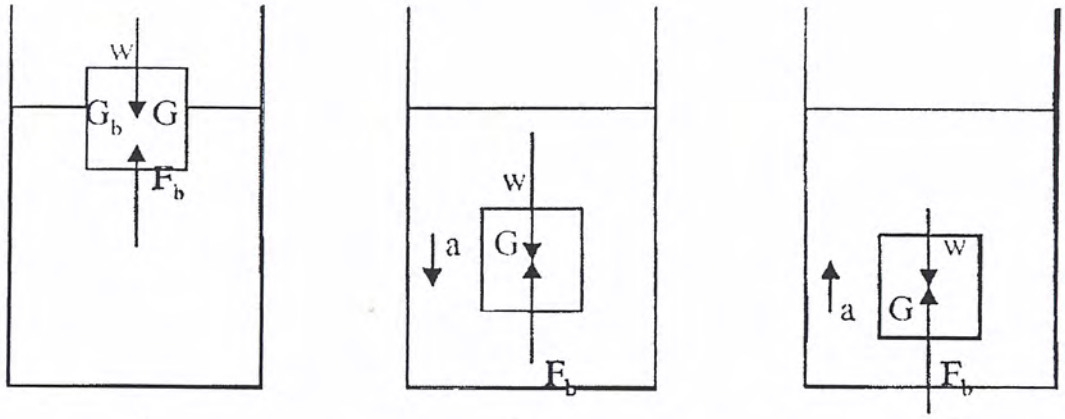
2.5 การลอยตัว

ขณะที่วัตถุก้อนหนึ่งอยู่ในของเหลว จะเห็นว่า มีทั้งแบบที่ มีส่วนลอยและส่วนที่จม ลอยปริ่มในของเหลว หรือจมในของเหลว ขึ้นอยู่กับคุณสมบัติของวัตถุที่สัมพันธ์กับสมบัติของของเหลว เราเรียกลักษณะที่วัตถุปรากฏอยู่ในของเหลวดังกล่าวเหล่านี้ว่า การ ลอยตัว (Buoyancy)

สมมติว่า วัตถุก้อนหนึ่งซึ่งมีความหนาแน่น ρ_0 มีปริมาตร V_0 มวล m และมีน้ำหนัก w มีการลอยตัวในของเหลวชนิดหนึ่ง ซึ่งมีความหนาแน่น ρ_f และมีแรงลอยตัว (buoyant force) ขนาด Fb ซึ่งคือแรงที่ของเหลวพยุงวัตถุ ในขณะที่ปริมาตรของวัตถุที่จมอยู่ในของเหลวเท่ากับ V และอยู่ในบริเวณที่มีค่าความโน้มถ่วงของโลกคือ g

อนึ่ง พึงสังเกตว่า เวคเตอร์เนื่องจากน้ำหนักของวัตถุ (w) กระทำ ณ จุดศูนย์ (center of gravity) ของก้อนวัตถุ ส่วนเวคเตอร์ของแรงลอยตัว (Fb) กระทำ ณ จุดศูนย์ถ่วงของวัตถุเฉพาะส่วนที่จมอยู่ในของเหลว และเรามีนิยามของแรงลอยตัวดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก. ข. ค.

รูปที่ 2.13 การลอยตัว

แรงลอยตัวมีขนาดเท่ากับขนาดน้ำหนักของของเหลว ซึ่งมีปริมาตรเท่ากับส่วนที่วัตถุจมในของเหลวนั้น ณ ที่นี้ จะพิจารณาตามกรณีต่างๆ ดังนี้

2.5.1 เมื่อวัตถุจมในของเหลว

ในสภาวะดังกล่าวนี้ แสดงว่า แรงภายนอกสุทธิที่กระทำกับวัตถุเท่ากับศูนย์ หรือกล่าวได้ว่า แรงลอยตัว (Fb) ของของเหลวย่อมมีขนาดเท่ากับขนาดของน้ำหนักวัตถุ (w) แต่มีทิศทางตรงกันข้ามในแนวเดียวกัน

กล่าวคือ $Fb = -w$ หรือ $Fb = w$ (2.8)

โดยที่ $w = mg = \rho_0 V_0 g$ (2.9)

จากรูป 2.13 ก วัตถุก้อนนั้นอยู่สมดุลในของเหลว โดยมีส่วนที่จมอยู่ในของเหลว (V) น้อยกว่าปริมาตรของวัตถุ (V_0) กล่าวคือ $V < V_0$

ตามกรณีนี้ $Fb = \rho_f V g = \rho_0 V_0 g$ (2.10)

หรือ $$\frac{\rho o}{\rho f} = \frac{V}{Vo}$$ (2.11)

เนื่องจาก $V < V_0$ จะได้ $\rho_0 < \rho_f$ แสดงว่า วัตถุก้อนหนึ่งซึ่งมีค่าความหนาแน่น (ρ_0) น้อยกว่าค่าความหนาแน่นของของเหลว (ρ_f) ก็จะจมอยู่ในของเหลวไม่ครบทั้งก้อน อีกนัยหนึ่ง ถ้า $\rho_0 = \rho_f$ จะได้ $V = V_0$ ผลคือ วัตถุจมในของเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.5.2 เมื่อวัตถุกำลังเคลื่อนที่ในของเหลว

การเคลื่อนที่ของวัตถุในของเหลวตามที่กล่าวถึงนี้ เป็นการเคลื่อนที่ในแนวดิ่งเท่านั้น หรือกล่าวอีกนัยหนึ่งได้ว่า เป็นการที่วัตถุกำลังจมลงในของเหลว ดังรูป 2.13 ข หรือกำลังลอยตัวขึ้นในของเหลวดังรูป 2.13 ค โดยลักษณะดังกล่าวนี้ วัตถุย่อมไม่อยู่ในสภาวะสมดุล หรือ กำลังเคลื่อนที่ด้วยความเร่งมีขนาดเป็น a นั่นเอง ด้วยเหตุนี้ แสดงว่า $Fb \neq w$ โดยที่ทั้ง Fb และ w กระทำ ณ จุดศูนย์ถ่วงของวัตถุ

จากรูป 2.13 ข $w > Fb$

จะได้

$$w - Fb = ma \tag{2.12}$$

$$\rho_0 V_0 g - \rho_f V_0 g = \rho_0 V_0 a \tag{2.13}$$

หรือ $$a = \left(\frac{\rho_0 - \rho_f}{\rho_0}\right) g$$

$$a = \left(1 - \frac{\rho_f}{\rho_0}\right) g \tag{2.14}$$

ผลที่ได้ตามสมการ นี้ แสดงว่า $\rho_f < \rho_0$ ซึ่งก็สอดคล้องกับประสบการณ์ทางธรรมชาติของกรณีที่วัตถุจมลงในของเหลว เช่น ก้อนหินที่จมลงในน้ำ ก้องหินย่อมมีความหนาแน่นมากกว่าค่าความหนาแน่นของน้ำ

โดยกลับกัน จากรูป 2.13 ค แสดงว่า $\rho_f > \rho_0$ แต่จะมีพฤติกรรมตามรูป 2.13 ค ที่สมมติได้ ต้องคิดว่า เรากดวัตถุก้อนหนึ่งให้จลงในของเหลวก่อน แล้วปล่อย วัตถุก้อนนี้จึงกำลังเคลื่อนที่ลอยขึ้นในของเหลวดังกล่าวนั้น

ตามกรณีนี้ $Fb > w$

จะได้

$$Fb - w = ma \tag{2.15}$$

$$\rho_f V_0 g - \rho_0 V_0 g = \rho_0 V_0 a \tag{2.16}$$

หรือ $$a = \left(\frac{\rho_f}{\rho_0} - 1\right) g \tag{2.17}$$

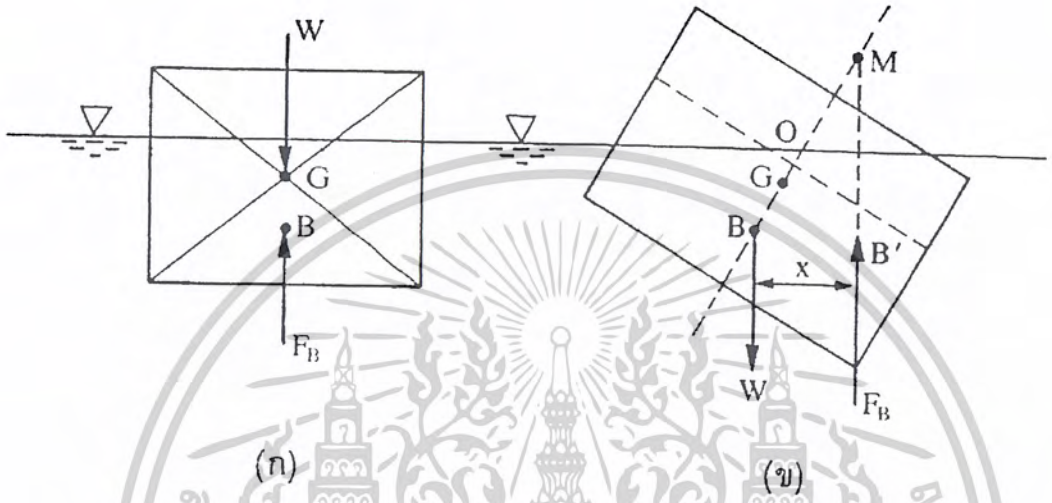
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50213

โดยที่

$\rho_f > \rho_0$ สอดคล้องกับที่กล่าวมาแล้วเป็นอย่างดี

2.6 เสถียรภาพของการลอยตัว



รูปที่ 2.14 การลอยตัว และเสถียรภาพ

ในรูป 2.14 (ก) G เป็นจุดศูนย์กลางมวลของวัตถุ B เป็นจุดศูนย์กลางของการลอยตัว (จุดเซนทรอยด์ของปริมาตรส่วนที่จม) ถ้ามีการโคลงตัวรอบแกนในแนวระดับที่ผ่านจุด O ในรูป 2.14 (ข) ทำให้แนวแรง F_B เลื่อนไปจากเดิมเป็นระยะ x ทำให้เกิดโมเมนต์แรงคู่ควบขนาด Wx และพบว่า แนวของ F_B นั้นตัดแนวแกนเดิมของวัตถุที่จุด M ซึ่งเรียกว่าจุดศูนย์กลางเบต้า ตำแหน่งจุด G และ M ใช้กำหนดเสถียรภาพของการลอยตัวดังนี้

1 สมดุลเสถียร เมื่อจุด M อยู่เหนือจุด G

2 สมดุลสะเทิน เมื่อจุด M ซ้อนทับจุด G

3 สมดุลไม่เสถียร เมื่อจุด M อยู่ต่ำกว่าจุด G

ระยะ GM หาได้จาก $GM = BM \pm GB$

ดังนั้น

$$GM = \frac{I}{V_{DIVE}} \pm GB \qquad (2.18)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

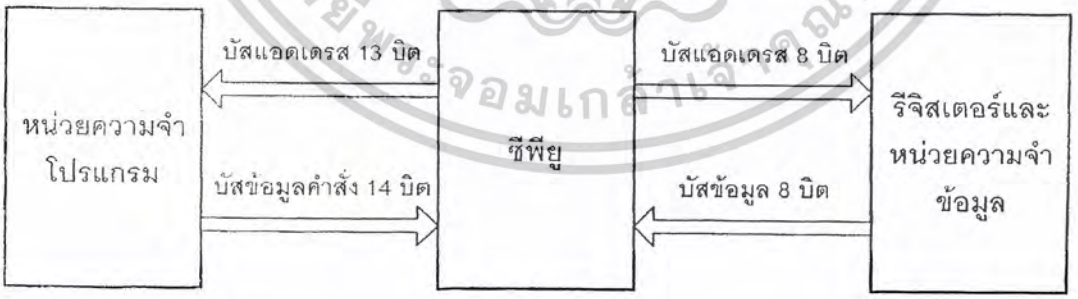
โดยที่ I คือ โมเมนต์ความเฉื่อยของพื้นที่ระนาบตัดของวัตถุขนานกับผิวของเหลวรอบแกนแนวระดับที่ผ่านจุด O ถ้าจุด G อยู่เหนือจุด B ใช้เครื่องหมายลบ ถ้า G อยู่ต่ำกว่า B ใช้เครื่องหมายบวก V_{DIVE} คือปริมาตรของวัตถุที่จมอยู่ใต้น้ำ

2.7 ไมโครคอนโทรลเลอร์ตระกูล PIC16F87x

2.7.1 สถาปัตยกรรมและโครงสร้างทางฮาร์ดแวร์ของ PIC16F87x

ไมโครคอนโทรลเลอร์ตระกูล PIC เป็นไมโครคอนโทรลเลอร์ที่มีสถาปัตยกรรมแบบฮาร์วาร์ด (Harvard architecture) กล่าวคือ มีการจัดแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกัน มีบัสสำหรับติดต่อแยกกัน ดังแสดงในรูปที่ 2.15 จะเห็นได้ว่าซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสของแอดเดรส 13 บิต และบัสของข้อมูลหน่วยความจำโปรแกรม 14 บิต ในขณะที่บัสติดต่อหน่วยความจำข้อมูลและรีจิสเตอร์ภายในเป็นแบบ 8 บิต

นอกจากการจัดสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งทำงานของไมโครคอนโทรลเลอร์ PIC ยังใช้กระบวนการที่เรียกว่า ไปป์ไลน์ (pipeline) ทำให้สามารถเฟตช์คำสั่งถัดไป ในขณะที่กำลังเอ็กซิคิวต์คำสั่งในปัจจุบัน ส่งผลให้ความเร็วในการทำงานของไมโครคอนโทรลเลอร์เพิ่มมากขึ้น นั่นจึงเป็นที่มาของความสามารถในการกระทำคำสั่ง 1 คำสั่งภายในสัญญาณนาฬิกา 1 ลูก (กระบวนการเฟตช์ (fetch) เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลคำสั่งนั้นให้เป็นรหัสเลขฐานสิบหกเพื่อให้ซีพียูเข้าใจ ส่วนกระบวนการเอ็กซิคิวต์ (execute) เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ตามที่คำสั่งนั้นๆ คำหนด) สำหรับกระบวนการไปป์ไลน์แสดงดังในรูปที่ 2.16

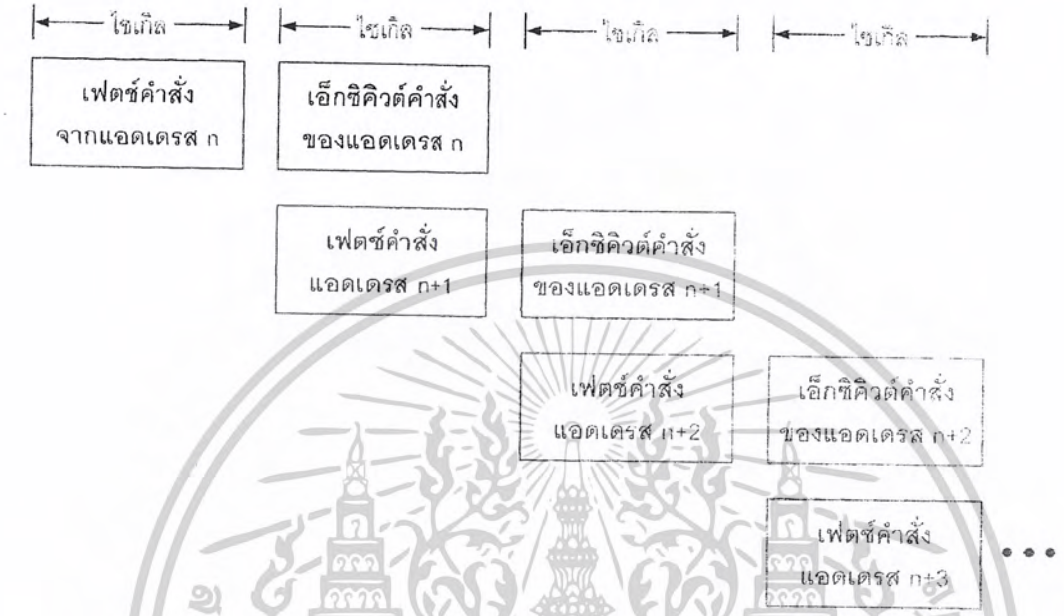


รูปที่ 2.15 ไดอะแกรมการทำงานพื้นฐานของไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมแบบฮาร์-วาร์ด

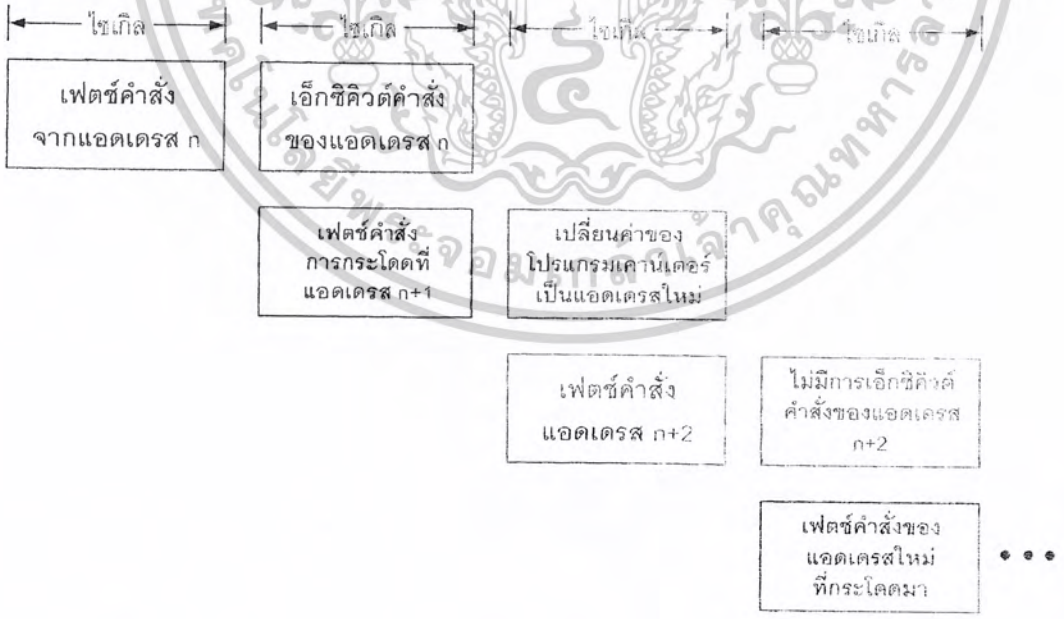
เมื่อเริ่มต้นกระทำคำสั่งที่ 1 ซีพียูจะเฟตช์คำสั่งจากหน่วยความจำโปรแกรมที่แอดเดรส n จากนั้นทำการเอ็กซิคิวต์ในไซเกิลต่อมา และที่ไซเกิลของการเอ็กซิคิวต์คำสั่งที่แอดเดรส n นั้น ซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะเริ่มต้นเฟตช์คำสั่งจากแอดเดรส n+1 ทันที เมื่อเอ็กซิคิวต์คำสั่งที่แอดเดรส n เรียบร้อย ซีพียูก็จะสามารถเอ็กซิคิวต์คำสั่งที่แอดเดรส n+1 ต่อเนื่องกันไปได้ในทันที และในทำนองเดียวกัน ขณะที่กำลังเอ็กซิคิวต์คำสั่งแอดเดรส n+1 ซีพียูก็จะดำเนินการเฟตช์คำสั่งที่แอดเดรส n+2 ต่อไป



รูปที่ 2.16 แสดงไดอะแกรมของกระบวนการไปป์ไลน์ที่ใช้ในไมโครคอนโทรลเลอร์ PIC



รูปที่ 2.17 แสดงไดอะแกรมของกระบวนการไปป์ไลน์เมื่อกระทำคำสั่งการกระโดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าคำสั่งที่กระทำนั้นเป็นคำสั่งการกระโดด จะมีขั้นตอนที่เพิ่มขึ้นมา ดังแสดงในรูปที่ 2.17 เมื่อทำการเอ็กซิคิวต์คำสั่งที่แอดเดรส n ซีพียูก็จะทำการเฟตช์คำสั่งที่แอดเดรส n+1 ปรากฏว่าคำสั่งที่แอดเดรส n+1 นั้นเป็นคำสั่งการกระโดด ดังนั้นในไซเกิลถัดไปจึงยังไม่เกิดการเอ็กซิคิวต์ในทันที แต่จะเป็นการเปลี่ยนค่าของโปรแกรมเคาน์เตอร์ (PC : Program Counter) ซึ่งเป็นรีจิสเตอร์ที่ใช้ในการระบุแอดเดรสถัดไปที่ซีพียูจะไปทำงาน ทำให้เกิดการกระโดดไปยังแอดเดรสใหม่ในระหว่างที่เกิดการกระโดดไปยังแอดเดรสใหม่นั้น ซีพียูจะทำการเฟตช์คำสั่งที่แอดเดรส n+2 ต่อไปตามขั้นตอนปกติแต่ทว่าเมื่อการกระโดดสิ้นสุดลง แอดเดรสของการทำงานเปลี่ยนไป จึงไม่เกิดการเอ็กซิคิวต์คำสั่งที่แอดเดรส n+2 แต่จะเกิดการเฟตช์คำสั่งที่แอดเดรสใหม่ที่กระโดดมาแทน จากนั้นจึงเข้าสู่กระบวนการทำงานตามปกติต่อไป ดังนั้นคำสั่งการกระโดด จึงต้องใช้ขนาดของหน่วยความจำมากกว่า 1 ไบต์ ส่งผลให้ต้องใช้สัญญาณนาฬิกาในการกระทำคำสั่งในกลุ่มนี้ 2 ลูก ซึ่งแตกต่างจากคำสั่งอื่นๆ ของไมโครคอนโทรลเลอร์ PIC ที่ใช้สัญญาณนาฬิกาเพียง 1 ลูก

2.7.2 โครงสร้างของไมโครคอนโทรลเลอร์ PIC16F873

ไมโครคอนโทรลเลอร์ PIC16F873 จะมีส่วนที่เพิ่มเติมเข้ามา ได้แก่ วงจรบราวเอาต์รีเซต (brown-out reset) สำหรับสร้างสัญญาณรีเซตซีพียูเมื่อไฟเลี้ยงลดต่ำลงเกินกว่าที่กำหนด, วงจรโปรแกรมข้อมูลด้วยแรงดันต่ำ (low-voltage programming), ไทเมอร์ที่มีมากถึง 3 ตัว, โมดูลแรงดันเปรียบเทียบ (reference voltage module), โมดูลเปรียบเทียบแรงดันอะนาลอก 2 ชุด (analog comparator), วงจรสื่อสารอนุกรม (USART : Universal Synchronous Asynchronous Receiver Transmitter) และโมดูลตรวจจับสัญญาณ-เปรียบเทียบข้อมูล-วงจรสร้างสัญญาณมอดูเลชั่นทางความกว้างของพัลส์หรือ PWM (CCP : Capture Compare Pulse-width modulation module)

นอกจากนั้นขนาดของหน่วยความจำทั้งส่วนของโปรแกรม, ข้อมูล, รีจิสเตอร์ และหน่วยความจำอีอีพรอมในไมโครคอนโทรลเลอร์ PIC16F873 รายละเอียดของการทำงานในแต่ละขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F87x การทำงานและการใช้งานขาสัญญาณทั้งหมดจะได้กล่าวต่อไป สำหรับในรายงานเล่มนี้จะขออ้างอิงกับ PIC16F873 เป็นหลัก เพราะมีขนาดหน่วยความจำมากกว่า การอธิบายด้วย PIC16F873 จะครอบคลุมในอนุกรม PIC16F87x ทั้งหมด

2.8 คุณสมบัติทางเทคนิคของ PIC16F873

- ซีพียูเป็นแบบ RISC (Reduce Instruction-Set Computer) มีคำสั่งใช้งานเพียง 35 คำสั่ง
- ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20MHz (สูงสุด)
- ขนาดหน่วยความจำโปรแกรม 4 กิโลเวิร์ด สำหรับ PIC16F873 และ 8 กิโลเวิร์ดสำหรับ PIC16F876
- หน่วยความจำแรมข้อมูล 192 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูลอีอีพรอม 128 ไบต์
- ตอบสนองแหล่งกำเนิดอินเตอร์รัปต์ได้ 13 แหล่ง
- มีสแต็ก 8 ระดับ
- มีวงจรเพาเวอร์ออนรีเซต (POR), เพาเวอร์อัปไทเมอร์ (PWRT) และออสซิลเลเตอร์สตาร์ตอัปไทเมอร์ (OST)
- มีวอตช์ด็อกไทเมอร์ (WDT) ที่มีวงจรออสซิลเลเตอร์ในตัว ทำให้มีความน่าเชื่อถือในการทำงานสูง
- เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล และเลือกระดับการป้องกันได้
- เลือกใช้วงจรกำเนิดสัญญาณนาฬิกาได้ 6 โหมดหลัก
  1. โหมด EC ใช้สัญญาณนาฬิกาจากภายนอก
  2. โหมด ER ใช้ตัวต้านทานภายนอก
  3. โหมด INTRC ใช้วงจร RC ภายในไมโครคอนโทรลเลอร์มี 2 ความถี่ให้เลือก
  4. โหมด LP ใช้คริสตอลพลังงานต่ำ ความถี่สูงไม่เกิน 200kHz
  5. โหมด XT ใช้คริสตอลความถี่ตั้งแต่ 100 kHz สูงสุดไม่เกิน 4 MHz
  6. โหมด HS ใช้คริสตอลความถี่สูง สูงสุดไม่เกิน 20 MHz (ต้องใช้กับรุ่นที่รองรับความถี่ 20 MHz ด้วย)
- สามารถโปรแกรมโดยใช้แรงดัน +5 V ได้
- สามารถโปรแกรมในวงจรได้
- ไฟเลี้ยง +3 ถึง +5.5V
- กระแสซิงก์และซอร์สของพอร์ต 25mA
- ขาพอร์ตปกติ 22 บิต เมื่อทำงานในโหมด INTRC และกำหนดให้ MCLR เป็นพอร์ตอินพุต
- ไทเมอร์ 3 ตัว (ไทเมอร์ 0, ไทเมอร์1 และไทเมอร์2)
- มีโมดูล CCP (Capture/Compare/PWM) 2 ชุด
- มีโมดูลเปรียบเทียบแรงดันอะนาลอก 5 อินพุต
- มีโมดูลสร้างแรงดันอ้างอิง
- มีโมดูลสื่อสารข้อมูลอนุกรม USART
- มีวงจรตรวจจับระดับแรงดันไฟเลี้ยงหรือบราวเอาต์ดีเท็กชั่น (Brown-out detection) เพื่อสร้างสัญญาณรีเซตซีพียูหรือเรียกว่า บราวเอาต์รีเซต (Brown-out reset : BOR)
- การใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด

  น้อยกว่า 2 mA ที่ +5V และสัญญาณนาฬิกา 4MHz, 15μA ที่ +3V และสัญญาณนาฬิกา 32 kHz น้อยกว่า 1μA ในโหมดประหยัดพลังงานหรือสแตนด์บายที่ไฟเลี้ยง +3V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 รีจิสเตอร์กำหนดการทำงานของไมโครคอนโทรลเลอร์ PIC

ในไมโครคอนโทรลเลอร์ PIC ทุกเบอร์จะมีพารามิเตอร์อยู่ตัวหนึ่งที่สำคัญและจำเป็นอย่างมากที่ผู้ใช้งานต้องทราบและทำความเข้าใจ พารามิเตอร์ตัวนี้คือ Configuration word ซึ่งเป็นรีจิสเตอร์ที่บรรจุข้อมูลสำหรับกำหนดการทำงานทั้งหมดของไมโครคอนโทรลเลอร์ ไม่ว่าจะเป็นการเลือกป้องกันข้อมูล การเลือกให้สามารถโปรแกรมหน่วยความจำด้วยแรงดันต่ำ เลือกการรีเซตอัตโนมัติเมื่อไฟเลี้ยงลดต่ำถึงค่าที่กำหนด ควบคุมการทำงานของวอตช์ด็อกไทเมอร์หรือกระทั่งการเลือกชนิดของวงจรกำเนิดสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์โดยการกำหนดข้อมูลสำหรับรีจิสเตอร์ตัวนี้นั้นสามารถกระทำได้ 2 ทางคือ ด้วยคำสั่ง_CONFIG (_ เป็นเครื่องหมาย under scroll หรือขีดล่าง 2 ตัวติดกัน) ในส่วนต้นของโปรแกรมภาษาแอสเซมบลี แล้วแอสเซมเบลอร์ด้วยโปรแกรม MPASM ซึ่งบรรจุอยู่ในชุดของโปรแกรม MPLAB อันเป็นซอฟต์แวร์สำหรับพัฒนาไมโครคอนโทรลเลอร์ PIC ของ Microchip ผู้ผลิตไมโครคอนโทรลเลอร์ PIC นั่นเอง ทางที่สองคือกำหนดที่ซอฟต์แวร์ที่ใช้ในการโปรแกรมหน่วยความจำของไมโครคอนโทรลเลอร์ การกำหนดสามารถกระทำในทางใดทางหนึ่งหรือทั้งสองทางก็ได้ แต่ถ้าการกำหนดทั้งสองทางแตกต่างกัน การกำหนดที่ซอฟต์แวร์ของเครื่องโปรแกรมจะมีนัยสำคัญสูงกว่า

รีจิสเตอร์ Configuration มีขนาด 14 บิต เท่ากับขนาดของเวิร์ดในหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ PIC16F873 หรือเบอร์อื่นๆ ในอนุกรม PIC16xxxx เว้น PIC16C5x และ PIC16C50x ซึ่งมีขนาด 12 บิต ตำแหน่งของ configuration word อยู่ที่แอดเดรส 0x2007 ดังมีรายละเอียดของการกำหนดข้อมูลแต่ละบิตดังนี้

บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8	บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CP1	CP0	CP1	CP0	-	CPD	LVP	BODEN	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0

CP1, CP0 (Code protection bits- บิต 13-10) : บิตเลือกพื้นที่สำหรับป้องกันการอ่านข้อมูลจากหน่วยความจำโปรแกรม โดยข้อมูลในบิต 13 กับ 11 (บิต CP1) ต้องเหมือนกัน และข้อมูลในบิต 12 กับ 10 (บิต CP0) ต้องเหมือนกัน

ในกรณีหน่วยความจำโปรแกรมมีขนาด 4 กิโลเวิร์ด(PIC16F873)

"00" - เลือกป้องกันหน่วยความจำในแอดเดรส 0x0000-0xFFF (ป้องกันทั้งหมด)

"01" - เลือกป้องกันหน่วยความจำในแอดเดรส 0x800-0xFFF (ป้องกัน 3/4 ส่วนของหน่วยความจำในแอดเดรสด้านสูงทั้งหมด)

"10" – เลือกป้องกันหน่วยความจำในแอดเดรส 0xF00-0xFFF (ป้องกันหน่วยความจำในแอดเดรสด้านสูงครึ่งหนึ่ง)

"11" – เลือกไม่ป้องกันหน่วยความจำโปรแกรมหรือ Code protection off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีหน่วยความจำโปรแกรมมีขนาด 8 กิโลเวิร์ด (PIC16F876)

"00" – เลือกป้องกันหน่วยความจำในแอดเดรส 0x000-01FFF (ป้องกันทั้งหมด)

"01" – เลือกป้องกันหน่วยความจำในแอดเดรส 01000-01FFF (ป้องกันหน่วยความจำในแอดเดรสด้านสูงครึ่งหนึ่ง)

"10" – เลือกป้องกันหน่วยความจำในแอดเดรส 01F00-01FFF (ป้องกันหน่วยความจำในแอดเดรสด้านสูงครึ่งหนึ่ง)

"11" – เลือกไม่ป้องกันหน่วยความจำโปรแกรมหรือ Code protection off

CPD (Data code protection bit – บิต 8) : บิตเลือก การป้องกันการอ่านหน่วยความจำข้อมูล อีอีพรอม

"0" – เลือกป้องกันการอ่านข้อมูลจากหน่วยความจำข้อมูลอีอีพรอม

"1" – เลือกไม่ป้องกันการอ่านข้อมูลจากหน่วยความจำข้อมูลอีอีพรอม

LVP (Low voltage programming Enable-บิต 7): บิตเลือกการโปรแกรมหน่วยความจำด้วยแรงดันต่ำ

"0" – ไม่เลือกความสามารถนี้ ทำให้ขา RA4/PGM ทำงานเป็นขาพอร์ตดิจิตอลและใช้ขา MCLR รับแรงดันสูงสำหรับการโปรแกรมหน่วยความจำแทน

"1" – เลือกความสามารถนี้ ทำให้ขา RA4/PGM เป็นขาอินพุตรับแรงดันต่ำ (+5V) ในการโปรแกรมหน่วยความจำ

BODEN (Brown-out detect enable bit – บิต 6) : บิตเลือกการตรวจจับแรงดันไฟเลี้ยงเพื่อทำให้เกิดการรีเซต โดยอัตโนมัติหรือเรียกว่า บราวเอาต์รีเซต (brown-out reset)

"0" – ดิสเอเบิลความสามารถนี้

"1" – เอ็นเอเบิลความสามารถนี้

MCLRE (RA5/MCLR pin function select –บิต 5): บิตเลือกการทำงานของขา RA5/MCLR

"0" – เลือกให้ขา RA5/MCLR ทำงานเป็นขาพอร์ตอินพุตดิจิตอล

"1" – เลือกให้ขา RA5/MCLR ทำงานเป็นขารีเซตหลักหรือ MCLR

PWRTE (Power-up timer enable bit – บิต 3) : บิตเลือกการทำงานของเพาเวอร์อัปไทเมอร์

"0" – ดิสเอเบิลความสามารถนี้

"1" – เอ็นเอเบิลความสามารถนี้

WDTE (Watchdog timer enable bit – บิต 2) : บิตเลือกการทำงานของวอตช์ด็อกไทเมอร์

"0" – ดิสเอเบิลความสามารถนี้หรือ Watchdog timer off

"1" – เอ็นเอเบิลความสามารถนี้หรือ Watchdog timer on

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FOSC2:FOSC0 (Oscillator selection bit – บิต 4, บิต 1 และบิต 0) : บิตเลือกโหมดของวงจรกำเนิดสัญญาณนาฬิกาหลักของไมโครคอนโทรลเลอร์ PIC

"000" – โหมด LP (คริสตอลพลังงานต่ำ) ต่อคริสตอลเข้าที่ขา RA6/OSC2/CLKOUT และ RA7/OSC1/CLKIN

"001" – โหมด XT (คริสตอลหรือเรโซเนเตอร์) ต่อคริสตอลเข้าที่ขา RA6/OSC2/CLKOUT และ RA7/OSC1/CLKIN

"010" – โหมด HS (คริสตอลหรือเรโซเนเตอร์ความเร็วสูง) โดยต่อคริสตอลเข้าที่ขา RA6/OSC2/CLKOUT และ RA7/OSC1/CLKIN

"011" – โหมด EC (External clock: สัญญาณนาฬิกาจากภายนอก) โดยป้อนสัญญาณนาฬิกาเข้ามาที่ขา RA7/OSC1/CLKIN ส่วนขา RA6/OSC2/CLKOUT กำหนดให้เป็นขาพอร์ตอินพุตเอาต์พุต

"100" – โหมด INTRC (วงจร RC ภายในไมโครคอนโทรลเลอร์) และกำหนดให้ขา RA6/OSC2/CLKOUT และ RA7/OSC1/CLKIN เป็นขาพอร์ตอินพุตเอาต์พุตทำให้สามารถใช้งานพอร์ต A ได้เพิ่มเติมอีก 2 บิตคือ RA6 และ RA7

"101" – โหมด INTRC (วงจร RC ภายในไมโครคอนโทรลเลอร์) และกำหนดให้ขา RA6/OSC2/CLKOUT เป็นขาเอาต์พุตสัญญาณนาฬิกา CLKOUT ส่วนขา RA7/OSC1/CLKIN เป็นขาพอร์ตอินพุตเอาต์พุต ทำให้ใช้งานพอร์ต A เพิ่มอีก 1 บิตคือ RA7

"110" – โหมด ER (วงจรตัวต้านทานภายนอก) กำหนดให้ขา RA6/OSC2/CLKOUT เป็นขาพอร์ตอินพุตเอาต์พุต ส่วนขา RA7/OSC1/CLKIN ใช้ต่อตัวต้านทานเพื่อกำหนดค่าความถี่ของสัญญาณนาฬิกาหลัก ทำให้สามารถใช้งานพอร์ต A ได้เพิ่มเติมอีก 1 บิตคือ RA6

"111" – โหมด ER (วงจรตัวต้านทานภายนอก) กำหนดให้ขา RA6/OSC2/CLKOUT เป็นขาเอาต์พุตสัญญาณนาฬิกา CLKOUT ส่วนขา RA7/OSC1/CLKIN ใช้ต่อตัวต้านทานเพื่อกำหนดค่าความถี่ของสัญญาณนาฬิกาหลัก

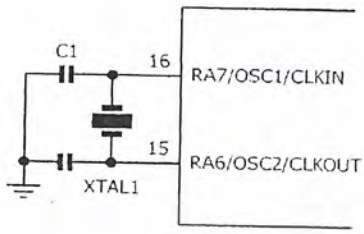
2.10 โหมดของสัญญาณนาฬิกา

PIC 16F873 สามารถเลือก โหมดของสัญญาณนาฬิกาได้มากถึง 6 โหมดหลัก 8 โหมดย่อย โดยกำหนดที่บิต FOSC0-FOSC2 ในรีจิสเตอร์ configuration ซึ่งในการทำงานจะต้องเลือกโหมดใดโหมดหนึ่ง สามารถสรุปเพื่อแบ่งได้เป็น 3 กลุ่มคือ

2.10.1 กลุ่มของคริสตอลหรือเซรามิกเรโซเนเตอร์ ซึ่งมีด้วยกัน 3 โหมดคือ LX , XT และ HS โดยมีการต่อคริสตอลและตัวเก็บประจุเพิ่มเติมอีก 2 ตัวในกรณีใช้กับคริสตอล แต่ถ้าใช้กับเซรา

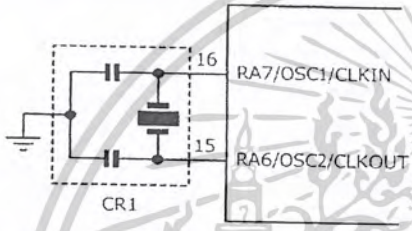
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มิกเรโซเนเตอร์แบบที่มีตัวเก็บประจุอยู่ภายใน สามารถตัดตัวเก็บประจุออกไปได้อีก 2 ตัว ดังในรูปที่ 2.18



โหมด	ความถี่	ค่าของตัวเก็บประจุ
LP	32kHz	68-100pF
	200kHz	15-30pF
XT	2MHz	15-30pF
	4MHz	15-30pF
HS	8MHz	15-30pF
	10MHz	15-30pF
	20MHz	15-30pF

(ก) กรณีใช้คริสตอล



(ข) กรณีใช้เซรามิกเรโซเนเตอร์แบบมีตัวเก็บประจุในตัว

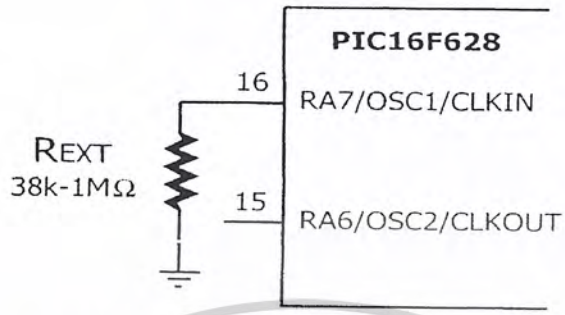
รูปที่ 2.18 แสดงโหมดการทำงานของสัญญาณนาฬิกากรณีใช้คริสตอลและเซรามิก

2.10.2 โหมด INTRC มีอีก 2 โหมดให้ใช้งาน แตกต่างกันที่ การทำงานของเขา RA6/OSC2/CLKOUT โดยในโหมด INTRC (100) จะกำหนดให้เป็นขาพอร์ตอินพุตเอาต์พุต ส่วนในโหมด INTRC (101) จะกำหนดให้ขา RA6 เป็นขาเอาต์พุตของสัญญาณนาฬิกา CLKOUT ซึ่งมีความถี่เท่ากับ 1/4 เท่าของความถี่สัญญาณนาฬิกาหลัด สำหรับความถี่ของสัญญาณนาฬิกาหลักในโหมดนี้เลือกได้ 2 ค่าคือ 4MHz และ 37MHz โดยกำหนดที่บิต OSCF (บิต 3) ในรีจิสเตอร์ PCON

2.10.3 กลุ่มสัญญาณนาฬิกาจากภายนอก มีด้วยกันคือ ER (2 โหมด) และ EC โดยในโหมด ER (external resistor) กำหนดความถี่ได้ 2 ค่าคือ 4 MHz และ 37 MHz เช่นเดียวกับโหมด INTRC และต้องต่อตัวต้านทานพูดดาวน์ภายนอกที่ขา RA7/OSC1/CLKIN ดังในรูปที่ 2.19โดยตัวต้านทานนี้จะเป็นตัวกำหนดความถี่ของสัญญาณนาฬิกาเมื่อเลือกโหมด 4 MHz หากค่าความต้านทานเปลี่ยนไปเป็นความถี่ก็จะเปลี่ยนแปลงตาม โดยอยู่ในช่วง 37kHz ถึง 4 MHz แต่ถ้าเลือกโหมด 37 kHz การเปลี่ยนแปลงค่าความต้านทานภายนอกจะไม่มีผลต่อความถี่ ค่าความต้านทานที่เหมาะสมอยู่ในช่วง 38 kΩ-1MΩ ในโหมด ER นี้ยังสามารถเลือกได้อีก 2 โหมด ซึ่งแตกต่างกันที่การกำหนดหน้าที่ของขา RA6/OSC2/CLKOUT โดยในโหมด ER (110) กำหนดให้ขา RA6 ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

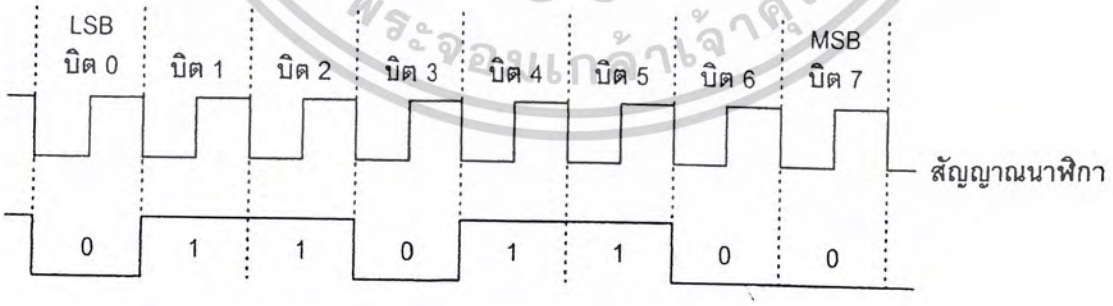
เป็นขาพอร์ต ส่วนในโหมด ER (111) กำหนดให้ขา RA6 ทำงานเป็นขาเอาต์พุตของสัญญาณนาฬิกาจากภายนอกผ่านทางขา RA7/OSC1/CLKIN ในขณะที่ขา RA6/OSC2/CLKOUT จะทำงานเป็นขาพอร์ต



รูปที่ 2.19 แสดงโหมดการทำงานของสัญญาณนาฬิกากรณีต่อตัวต้านทานพูดดาวน์ภายนอกที่ขา RA7/OSC1/CLKIN

2.11 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูลและกราวด์ รูปที่ 2.20 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.20 แบบอย่างง่ายที่สุดของข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

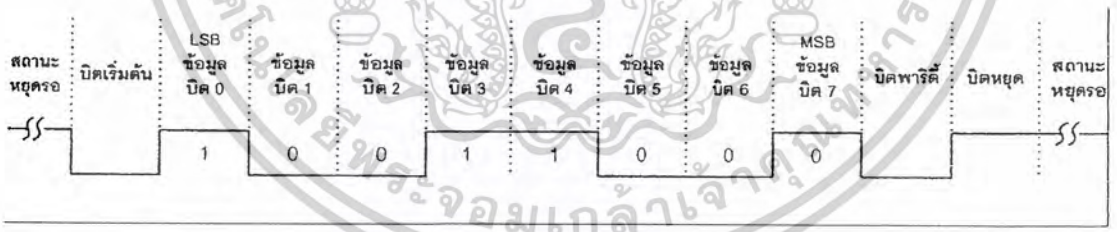
2.12 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสาย โดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second: bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต

รูปที่ 2.21 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วยระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 2.21 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd) , แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" มีจำนวนรวมกันเป็นเลขคี่

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

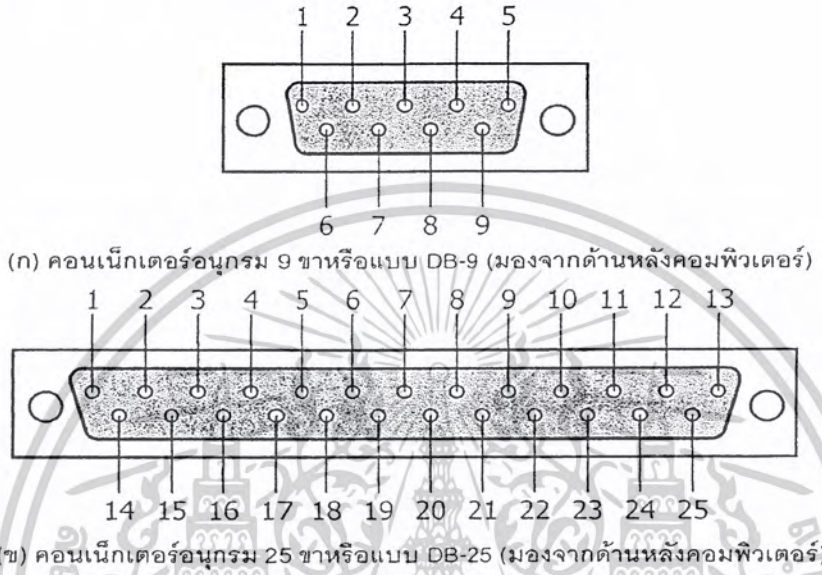
ตารางที่ 2.2 แสดงบิตพาริตี้ของข้อมูล

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ UART จะทำการตรวจทั้งบิตพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้เป็น NONE นั้นทั้งภาครับและภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 คอนเน็กเตอร์สำหรับพอร์ต RS – 232 และการเชื่อมต่อ

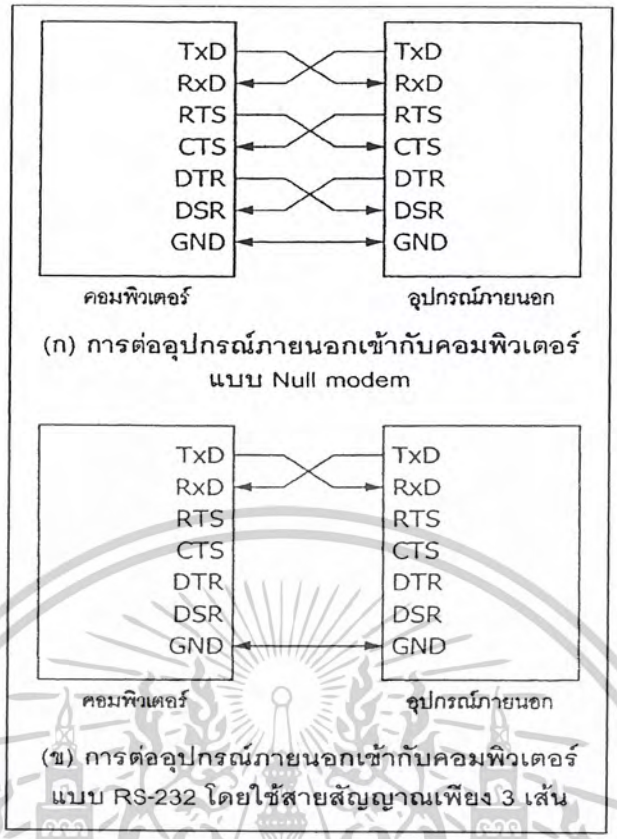
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB- 25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 2.22



รูปที่ 2.22 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบDB-9 & DB-25

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังรูปที่ 2.23 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 2.23 (ก) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบส่วนในรูปที่ 2.23 (ข) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้นโดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Cattier Detect : DBD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Trasmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Sinnal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

ตารางที่ 2.3 แสดงตำแหน่งขาของคอนเน็คเตอร์พอร์ตอนุกรมมาตรฐาน RS-232 แบบ DB9 และ DB25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 หน้าที่การทำงานแต่ละขาของพอร์ดอนุกรม RS-232

- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
- Receive Data : RD หรือ RXD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์
- Transmitted Data : TD หรือ TXD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติต่อด้วย โดย ขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- Signal Ground : GND ขากราวด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RST ก็คือ ขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.15 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitted ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่หลักของ UART คือหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART สามารถรับส่งข้อมูลๆได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์ นั้นสามารถรับและส่งได้ในคราวเดียวกัน

2.15.1 ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหลับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 5706 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุก ๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจีสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจีสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจีสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงานสามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UARTเบอร์ใหม่ ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.16 วงจรภายในและรีจีสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COMI, COM2, COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานภายในของพอร์ตอนุกรม ประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD, RI, DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

2.16.1 รีจิสเตอร์ตำแหน่ง 00H: รีจิสเตอร์บัฟเฟอร์

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาและข้อมูลที่จะส่งออกไป โดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูลที่ต้องการจะส่งจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูลและข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อข้อมูลที่รับเข้ามาเรียบร้อยและแปลงเป็นแบบขนานแล้ว ข้อมูลจะถูกส่งมายังรีจิสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจิสเตอร์นี้ออกไปรีจิสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับการรับข้อมูลในไบต์ต่อไป

2.16.2 รีจิสเตอร์ตำแหน่ง 01H: รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์

เป็นรีจิสเตอร์สำหรับการอีนาเบิลการอินเตอร์รัปต์ ซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเตอร์รัปต์ขึ้นมา ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์นี้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	SINP	ERBK	TBE	RxRD

บิต 4-7 บิตเหล่านี้ไม่ถูกใช้งาน กำหนดให้เท่ากับ “0”

SINP อีนาเบิลการอินเตอร์รัปต์เนื่องจากเกิดการเปลี่ยนสถานะที่ขาอินพุท (CTS , DSR , DCDหรือขา RI

“1” อีนาเบิลการอินเตอร์รัปต์

“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือดิสเอเบิล

ERBK อีนาเบิลการอินเตอร์รัปต์เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจากพาริตี้, โอเวอร์รัน, เฟรมข้อมูล หรือการเบรกข้อมูล

“1” อีนาเบิลการอินเตอร์รัปต์

“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือดิสเอเบิล

TBE อีนาเบิลการอินเตอร์รัปต์เมื่อรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

“1” อีนาเบิลการอินเตอร์รัปต์

“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือดิสเอเบิล

RxED อีนาเบิลการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ได้รับข้อมูลเรียบร้อย

“1” อีนาเบิลการอินเตอร์รัปต์

“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือดิสเอเบิล

2.16.3 รีจิสเตอร์ตำแหน่ง 02H: รีจิสเตอร์แสดงโหมดและสถานะการอินเตอร์รัปต์

มีรายละเอียดของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	ID1	ID0	PND

บิต 3-7 ไม่ได้ใช้งาน อ่านค่าได้เท่ากับ “0”

ID1, ID0 ใช้งานร่วมกันเพื่อแจ้งสาเหตุของการเกิดอินเตอร์รัปต์

“00” เกิดการอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตขึ้น การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 4

“01” เกิดการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่างขึ้น การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"10" เกิดการอินเตอร์รัปต์เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 2

"11" เกิดการอินเตอร์รัปต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรก (break: เกิดการหยุดถ่ายทอดข้อมูลกะทันหัน) การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 1 หรือมีนัยสำคัญสูงสุด

PND ใช้แสดงสถานะของการเกิดอินเตอร์รัปต์

"1" แสดงว่าไม่มีการอินเตอร์รัปต์

"0" แสดงว่ามีการอินเตอร์รัปต์เกิดขึ้น

เมื่อมีการสร้างสัญญาณอินเตอร์รัปต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดอินเตอร์รัปต์ครั้งต่อไป โดยสามารถทำได้ดังนี้คือ

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตจะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดการอินเตอร์รัปต์เนื่องจากบัฟเฟอร์ส่งข้อมูลว่าง จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ส่งข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเก็บข้อมูลลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อย จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการเบรก จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านค่ารีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม

2.16.4 รีจิสเตอร์ตำแหน่ง 03H: รีจิสเตอร์กำหนดรูปแบบของข้อมูล

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0

DLAB ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัฟเฟอร์ (00H)

"1" เป็นการเข้าสู่โหมดการหารค่าบอดเรต

"0" เป็นการเข้าถึงรีจิสเตอร์บัฟเฟอร์ (รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับอีนาเบิลการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 01H)

เมื่อบิต DLAB เป็น "1" รีจิสเตอร์บัฟเฟอร์ (00H) และรีจิสเตอร์อีนาเบิลการอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์รัปต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนดค่าบอดเรต โดยรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหาค่าบอดเรตสามารถเขียนเป็นสมการได้ดังนี้

บอดเรต = 115200/ค่าตัวหาร 16 บิต

ค่าตัวเลข 115200 มาจากความถี่ของคริสตอลในวงจร UART ภายในเครื่องคอมพิวเตอร์โดยคริสตอลที่ใช้มีความถี่ 1.8432 MHz วงจรภายใน UART จะทำการหารค่าความถี่นี้ด้วย 16 ทำให้ได้ค่าความถี่ 115200 Hz ออกมา

ค่าตัวหาร 16 บิต = ข้อมูลในรีจิสเตอร์ 00H + (256 x ข้อมูลในรีจิสเตอร์ 01H)

สมมติว่าต้องการค่าบอดเรตเท่ากับ 9600 ค่าตัวหารที่ใช้จะต้องมีค่าเท่ากับ 12 ซึ่งค่านี้จะต้องถูกโหลดลงในรีจิสเตอร์ 00H และโหลดค่า 0 ลงไปในรีจิสเตอรี 01H ค่าตัวหารที่ทำให้เกิดค่าบอดเรตสูงสุดที่ 115200 บิตต่อวินาที คือค่า 0001 นั่นคือรีจิสเตอร์ 00H มีค่าเท่ากับ 1 และรีจิสเตอร์ 01H มีค่าเท่ากับ 0

BRK ใช้ควบคุมการหยุดถ่ายทอดข้อมูล

"1" สามารถหยุดหรือเบรกได้

"0" ไม่มีการหยุดหรือเบรกได้

PAR2, PAR1, PAR0 ใช้เพื่อกำหนดบิตพาริตี้

"000" ไม่ใช้พาริตี้

"001" กำหนดพาริตี้คี่

"011" กำหนดพาริตี้คู่

"101" มาร์ก (mark)

"111" ช่องว่าง (space)

STOP ใช้กำหนดจำนวนบิตปิดท้าย

"1" มีบิตปิดท้าย 2 บิต

"0" มีบิตปิดท้าย 1 บิต

DAB1, DAB0 ใช้ร่วมกันในการกำหนดจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด

"00" จำนวนบิตข้อมูลเท่ากับ 5 บิต

"01" จำนวนบิตข้อมูลเท่ากับ 6 บิต

"10" จำนวนบิตข้อมูลเท่ากับ 7 บิต

"11" จำนวนบิตข้อมูลเท่ากับ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.5 รีจิสเตอร์ตำแหน่ง 04H: รีจิสเตอร์ควบคุมโมเด็ม

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

บิต 5-7 ไม่มีการใช้งาน อ่านค่าได้เท่ากับ "0"

LOOP "1" อีนาเบิลการส่งค่ากลับ

"0" ดิสเอเบิล

OUT1, OUT2 "1" อีนาเบิลการใช้งานภายใน

"0" ดิสเอเบิล

RTS ใช้ควบคุมการทำงานของขา RTS (Ready To Send)

"1" อีนาเบิล

"0" ดิสเอเบิล

DTR ใช้ควบคุมการทำงานของขา DTR (Data Terminal Ready)

"1" อีนาเบิล

"0" ดิสเอเบิล

2.16.6 รีจิสเตอร์ตำแหน่ง 05H: รีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลอนุกรมของ UART

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อแสดงสาเหตุของการเกิดอินเตอร์รัปต์ มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	TXE	TBE	BREK	FRME	PARE	OVFE	RxRD

TXE (Transmitter Empty) "1" แสดงว่ารีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" แสดงว่ายังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

TBE (Transmitter Buffer Empty) "1" รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BREK (Break) “1” UART ตรวจพบการเบรก

“0” ไม่มีการเบรก

FRME (Frame Error) “1” UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล

“0” ไม่พบความผิดพลาดด้านเฟรมข้อมูล

PARE (Parity Error) “1” UART ตรวจพบความผิดพลาดทางพาริตี้

“0” ไม่พบความผิดพลาดทางพาริตี้

OVRE (Overrun Error) “1” UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน

“0” ไม่พบความผิดพลาดแบบโอเวอร์รัน

RxRD (Received Data Ready) “1” มีการรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์

“0” ไม่มีข้อมูล

2.16.7 รีจิสเตอร์ตำแหน่ง 06H: รีจิสเตอร์แสดงสถานะของโมเด็ม

ใช้เพื่อกำหนดสถานะสัญญาณอินพุต ของพอร์ตอนุกรม RS-232 ซึ่งได้แก่ สัญญาณ DCD, DSR, CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบอเนกประสงค์ ดังมีรายละเอียดหน้าที่ของแต่ละบิตต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

DCD ใช้แสดงสภาวะของขา DCD

“1” แสดงว่าที่ขา DCD เป็นลอจิก “1”

“0” แสดงว่าที่ขา DCD เป็นลอจิก “0”

RI ใช้แสดงสภาวะของขา RI

“1” แสดงว่าที่ขา RI เป็นลอจิก “1”

“0” แสดงว่าที่ขา RI เป็นลอจิก “0”

DSR ใช้แสดงสภาวะของขา DSR

“1” แสดงว่าที่ขา DSR เป็นลอจิก “1”

“0” แสดงว่าที่ขา DSR เป็นลอจิก “0”

DCTS (Delta Clear to Send) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

“1” แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

“0” แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DDSR (Delta Data Set Ready) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DSR

"1" แสดงว่าบิต DSR (Data Set Ready) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

"1" แสดงว่าบิต RI (Ringing Indicator) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDCD (Delta Data Carrier Detect) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DDCD

"1" แสดงว่าบิต CTS (Clear to Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DCTS (Delta Clear to Send) ใช้แสดงสภาวะของขา CTS

"1" แสดงว่าที่ขา CTS เป็นลอจิก "1"

"0" แสดงว่าที่ขา CTS เป็นลอจิก "0"

2.16.8 รีจิสเตอร์ตำแหน่ง 07H: รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว

ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใดๆ ต่อการใช้งาน UART

2.17 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และDTR) และสัญญาณแสดงสถานะอินพุต (CTS , DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์ และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.18 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1: 3F8H

COM2: 2F8H

COM3: 3E8H

COM4: 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆ มีรายละเอียดดังนี้

COM2 = 0000:00402H – 0000:0403H

COM3 = 0000:0404H – 0000:0405H

COM4 = 0000:0406H – 0000:0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000: 0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 2.4

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

ตารางที่ 2.4 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19 การเขียนโปรแกรมใช้งานพอร์ตอนุกรม

2.19.1 การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรม การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรมสามารถทำได้หลายวิธีคือ

วิธีที่ 1 โดยการใช้โปรแกรม Debug ไปดูค่าตำแหน่งแอดเดรสที่ตำแหน่ง 0000:0400H โดยใช้พิมพ์คำสั่งที่ดอสพร้อมพ์

ค่าที่เห็นมีตำแหน่งแอดเดรสไล่เรียงกันตั้งแต่หน่วยความจำที่ 0000: 0400H-0000:0407H ตัวอย่างเช่น ที่ตำแหน่งหน่วยความจำ 0000:0400H-0000:0401H แสดงตัวเลข F8 03 ซึ่งหมายความว่าแอดเดรสของพอร์ต COM1 คือ 03F8H นั้นเอง สำหรับจำนวนของพอร์ตอนุกรมที่ระบุอยู่ที่หน่วยความจำตำแหน่ง 0000:0411H มีค่าเท่ากับ D8H ซึ่งเมื่อแปลงเป็นเลขฐานสองจะได้ค่าเป็น 11011000 บิตที่ 1-3 มีค่าเท่ากับ 100 หมายความว่ามีจำนวนพอร์ตอนุกรมทั้งหมด 4 พอร์ต

วิธีที่ 2 สามารถดูได้จากวินโดวส์ 95 โดยไปที่ Control panel เรียก system => device manager => ports (com & LTP) จากนั้นเลือกพอร์ตอนุกรมที่ต้องการดูค่าแล้วเลือก properties => resources ซึ่งหน้าต่างนี้จะแสดงทั้งตำแหน่งแอดเดรสของพอร์ตอนุกรมนั้นๆรวมถึงตำแหน่งของอินเตอร์รัปต์ที่ใช้ด้วย

วิธีที่ 2 การเขียนโปรแกรมเพื่ออ่านค่า โดยสามารถใช้โปรแกรมภาษาใดๆ ก็ได้เพื่ออ่านค่า แต่ในตัวอย่างนี้จะใช้ QBASIC ในการแสดงค่าแอดเดรสของพอร์ตอนุกรม โดยใช้คำสั่ง PEEK ซึ่งเป็นคำสั่งที่อ่านค่าจากหน่วยความจำ นอกจากนี้ยังอ่านค่าจำนวนของพอร์ตอนุกรมออกมาโดยใช้คำสั่งเดียวกัน แต่จะใช้คำสั่ง AND เข้ามาเพื่อเลือกเอาเฉพาะบิตที่ต้องการอ่านค่าเท่านั้น จากนั้นก็ทำการเลื่อนบิตไปทางขวา 1 บิต โดยใช้วิธีการหารด้วย 2 เพื่อให้บิตที่ต้องการไปอยู่ด้านขวามือสุด และแสดงค่าจำนวนพอร์ตที่แท้จริงออกมา คำสั่ง HEX $ ช่วยในการแสดงผลตำแหน่งแอดเดรสของพอร์ตอนุกรมที่ออกมาเป็นค่าเลขฐานสิบหก ส่วนการคูณค่าด้วย 100H นั้นก็เพื่อที่จะเลื่อนบิตไปด้านหน้า 1 ไบต์ทำให้เมื่อเวลานำมาบวกแล้วจะได้ค่าเป็น 2 ไบต์พอดีแสดงได้ดังโปรแกรมตัวอย่างต่อไปนี้

```
DEF SEG = 0
CLS
PRINT "Address of COM1: ", HEX$((PEEK (&H401)* &H100) +PEEK (&H400))
PRINT "Address of COM2: ", HEX$((PEEK (&H403)* &H100) +PEEK (&H402))
PRINT "Address of COM3: ", HEX$((PEEK (&H405)* &H100) +PEEK (&H404))
PRINT "Address of COM4: ", HEX$((PEEK (&H407)* &H100) +PEEK (&H406))
PRINT "Number of RS-232 Port in this computer: ", (PEEK (&H411) AND &HE)/2
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.2 การกำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม

ก่อนการใช้งานพอร์ตอนุกรมนั้นจะต้องมีการกำหนดค่าเริ่มต้นให้กับตัวมันก่อน ซึ่งก็คือการกำหนดจำนวนบิตข้อมูลที่ต้องการส่ง , จำนวนบิตปิดท้าย , ชนิดของพาริตี้ที่ใช้ และบอดเรต

การกำหนดสามารถทำได้หลายวิธีคือ

วิธีที่ 1 เป็นการกำหนดจากดอสพร้อมพ์ โดยใช้คำสั่ง MODE ซึ่งมีวิธีการใช้งานดังนี้

MODE COMm: baud = b, parity = p , data = d , stop = s , retry = r

หรือ MODE COMm: b, p, d, s, r

ตัวอย่าง MODE COM1: 9600, n, 8, 1 จะเป็นการกำหนดให้พอร์ตอนุกรม COM1 มีบอดเรตเท่ากับ 9600 บิตต่อวินาที ไม่มีการตรวจสอบพาริตี้ รับส่งข้อมูลแบบ 8 บิต และมีบิตปิดท้าย 1 บิต

วิธีที่ 2 เป็นการกำหนดโดยใช้อินเตอร์รัปต์ของดอสตำแหน่งที่ 14H ซึ่งการใช้งานจะต้องกำหนดค่าต่างๆ ลงในรีจิสเตอร์ด้วย โดยจะต้องกำหนดให้รีจิสเตอร์ AH มีค่าเท่ากับ 0 รีจิสเตอร์ DX เก็บค่าของพอร์ตอนุกรมที่ต้องการกำหนดค่าเริ่มต้น โดย

DX = 0 จะกำหนดให้กับพอร์ตอนุกรม COM1

DX = 1 จะกำหนดให้กับพอร์ตอนุกรม COM2

DX = 2 จะกำหนดให้กับพอร์ตอนุกรม COM3

DX = 3 จะกำหนดให้กับพอร์ตอนุกรม COM4

รีจิสเตอร์ AL ซึ่งมีขนาด 8 บิต ใช้เก็บค่าเริ่มต้นต่างๆ มีรายละเอียดดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BD2	BD1	BD0	PAR1	PAR0	STOP	AD1	AD0

BD2, BD1, BD0 ใช้สำหรับกำหนดค่าบอดเรต

"111" บอดเรตเท่ากับ 9600 บิตต่อวินาที

"110" บอดเรตเท่ากับ 4800 บิตต่อวินาที

"101" บอดเรตเท่ากับ 1200 บิตต่อวินาที

"011" บอดเรตเท่ากับ 600 บิตต่อวินาที

"010" บอดเรตเท่ากับ 300 บิตต่อวินาที

"001" บอดเรตเท่ากับ 150 บิตต่อวินาที

"000" บอดเรตเท่ากับ 110 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมย่อยเทอร์โบปาสคาลสำหรับการกำหนดค่าให้กับพอร์ตอนุกรมโดย กำหนดค่าเริ่มต้นให้กับพอร์ตอนุกรม COM1 มีอัตราบอดเรตเท่ากับ 9600 ไม่มีการตรวจพาริตี้ บิตข้อมูล 8 บิตและบิตปิดท้าย 1 บิต เขียนได้ดังนี้

```
Procedue initial;
Var Regis: registers;
Begin
        With Regis do begin
                Ah: = 0;
                Al: = $0e3;
                Dx: = 0;
                Intr ($14, Regis);
        End;
End;
```

PAR1, PAR0 ใช้กำหนดพาริตี้โดย

"0" หรือ "10" ไม่มีการตรวจสอบพาริตี้

"01" พาริตี้คี่

"11" พาริตี้คู่

STOP ใช้กำหนดจำนวนของบิตปิดท้าย

"1" มีบิตปิดท้ายเท่ากับ 2 บิต

"0" มีบิตปิดท้ายเท่ากับ 1 บิต

DA1, DA0 ใช้กำหนดความยาวของข้อมูลโดย

"10" ความยาวข้อมูลเท่ากับ 7 บิต

"11" ความยาวข้อมูลเท่ากับ 8 บิต

2.19.3 การรับและส่งข้อมูลแบบอนุกรม

การรับและส่งข้อมูลแบบอนุกรมมีหลายวิธีในการรับส่งข้อมูลแบบอนุกรมผ่านพอร์ตอนุกรม RS –232 เช่น ใช้คำสั่งพิมพ์ออกทางเครื่องพิมพ์ เรียกอินเตอร์รัปต์ของไบออสหรือของดอส การเขียนหรืออ่านไปยังแอดเดรสของพอร์ตโดยตรง วิธีสุดท้ายเป็นวิธีที่มีความยืดหยุ่นในการใช้งานที่สุด ยกตัวอย่าง ถ้าต้องการส่งข้อมูลไปยังพอร์ตอนุกรม COM1 สามารถเขียนข้อมูลโดยตรงไปที่รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (แอดเดรส 3F8H) โดยใช้คำสั่งภาษา QBASIC ง่ายได้ดังนี้

```
Out &H3F8, X
```

ค่า X ในที่นี้หมายถึงข้อมูลที่ต้องการส่ง มีขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการอ่านข้อมูลจากพอร์ตอนุกรมจะเป็นการอ่านข้อมูลมาจากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (แอดเดรส 3F8H เช่นเดียวกัน) ซึ่งสามารถเขียนโปรแกรมง่ายๆได้ดังนี้

Y = INP (&H3F8)

ค่า Y ในที่นี้คือค่าที่อ่านได้จากรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลโดยมีขนาด 8 บิต

สำหรับการเขียนโปรแกรมด้วย TURBO PASCAL ก็สามารถใช้คำสั่ง

Port [$3F8] = X สำหรับการเขียนข้อมูลไปยังพอร์ตอนุกรมและ

Y = port [$3F8] สำหรับการอ่านข้อมูลจากพอร์ตอนุกรม

แต่เมื่อใช้คำสั่งนี้ในขณะที่โปรแกรมทำงานผ่านระบบปฏิบัติการวินโดวส์ จะไม่สามารถใช้งานได้เนื่องจากระบบปฏิบัติการวินโดวส์ ได้เข้าฝังตัวพอร์ตอนุกรมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการแล้ว ดังนั้นการเรียกใช้งานจึงจำเป็นต้องเรียกผ่านเครื่องมือที่ติดต่อผ่านระบบปฏิบัติการ เช่น การใช้คอนโทรล MSComm32.OCX ของโปรแกรม Visual BASIC

2.20 คอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมีคัสตอมคอนโทรลสำหรับสื่อสารอนุกรมผ่านพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกไว้ 2 ทางเลือกเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event- driven communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา data carrier detect (DCD) หรือขา request to send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชั่นต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต ต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ control panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (property) มากมาย แต่สามารถทำความเข้าใจได้ดังนี้คือ

2.20.1 CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

Object. Commport [= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ ProtOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

2.20.2 Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด,พาริตี้,จำนวนของบิตข้อมูล,จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

Object. Setting [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบ string มีรูปแบบเป็น “BBBB, P, S” โดย BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตี้, D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น "9600, N, 8, 1"

ค่าบอดเรตมาตรฐานที่ใช้กับMSComm มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

1200 บิตต่อวินาที

2400 บิตต่อวินาที

9600 บิตต่อวินาที (ค่าปกติ)

14400 บิตต่อวินาที

19200 บิตต่อวินาที

28800 บิตต่อวินาที

38400 บิตต่อวินาที (สงวน)

56000 บิตต่อวินาที (สงวน)

128000 บิตต่อวินาที (สงวน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

256000 บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่ (EVEN)
M	ลอจิก "1" (MARK)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก "0" (Space)

ตารางที่ 2.5 แสดงค่ามาตรฐานในการกำหนดค่าพาริตี้

ค่าที่ใช้ในการกำหนดจำนวนบิตมี ค่า 5 ค่าคือ 4,5,6,7 และ8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ) 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้บิต จำนวนบิตข้อมูล 8 บิต และบิต ปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSComm1. Settings = "9600, N, 8,"

สาเหตุที่ค่าที่กำหนดจะต้องอยู่ในเครื่องหมายคำพูด "" เนื่องจาก ค่าที่กำหนดนี้อยู่ในรูปตัวแปร string

2.20.3 PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

Object. PortOpen [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น false หลังจากปิดพอร์ต แต่ถ้าเซตเป็นหลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1. Settings = "9600, n, 8, 1"
MSComm1. CommPort = 1
MSComm1. PortOpen = True
```

2.20.4 Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

Object. Input

คุณสมบัติ InputMode เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private sub Commandl_Click ()
Dim Instring as String
    MSComm1. InputLen = 0                 'Retrieve all available data.
    If MSComm. InBufferCount Then         'Check for data.
    InString = MSComm1. Input             'Read data.
    End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.20.5 InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object. InBufferCount [= value]

คำสั่ง Inbuffercount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

2.20.6 InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

Object. InBufferSize [= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1024 ไบต์

การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่ามากขึ้น

2.20.7 InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object.InputLen [= value]

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้ คำสั่ง Input ของ MSComm อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่าง (“”) กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ InBufferCount โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```
Private Commandl_Click ()
Dim CommData as String
MSComm1.InputLen = 10
CommData = MSComm1.Input          'Specify a 10 Charcter block of data.
End Sub                           'Read data.
```

2.20.8 InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

Object. InputMode [= value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบต์จากพอร์ตอนุกรมและเก็บข้อมูลไว้ในตัวแปรแบบอาเรย์ ชนิดข้อมูลเป็นแบบไบต์

```
Private Sub Command1_Click ()
Dim Buffer as Variant
Dim Arr () as Byte
MSComm1.CommPort = 1
MSComm1.PortOpen = True
MSComm1.InputMode = comInputModeBinary          'Set InputMode to read binary data
Do Until MSComm1.InBufferCount < 10             'Wait until 10 bytes are in the Input buffer
    DoEvents
Loop
Buffer = MSComm1.Input                          'Store binary data in buffer
Arr = Buffer                                    'Assign to byte array for processing
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.20.9 Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ของข้อมูล

รูปแบบการใช้งานคำสั่ง

Object. Output [= value]

ค่า Value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่จะต้องกำหนดชนิดของข้อมูลเป็นแบบ variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่าง โปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ด ไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output

```
Private Sub Form_KeyPress (KeyAscii As Integer)
    Dim Buffer as Variant
    MSComm1.CommPort = 1                    'Use COM1
    MSComm1.PortOpen = True                 'Open port
    Buffer = Chr$(KeyAscii)
    MSComm1.Output = Buffer                 'Send data
End Sub
```

2.20.10 OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

Object. OutBufferCount [= value]

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ "0" ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

2.20.11 OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

Object.OutBufferSize [= value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

2.20.12 ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี้

รูปแบบการใช้งานคำสั่ง

Object.ParityReplace [= value]

บิตพาริตี้ เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี้ คอนโทรล MSCOM จะทำการบวกบิตทุกบิตที่มีค่าลอจิก "1" ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก "1" เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตี้ไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่า เริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตี้ผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง ("") จะเป็นการยกเลิกการใช้งาน ParityReplace และไม่มีการป้อนข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งสามารถใช้ค่าใดๆก็ได้ที่เป็นโค้ด ANIS มีค่าอยู่ระหว่าง 0-255

2.20.13 DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งานคำสั่ง

Object.DTREnable [= value]

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น "0" หรือ "1" โดย

True หมายถึง ให้ขา DTR มีลอจิก "1"

False หมายถึง ให้ขา DTR มีลอจิก "0" (เป็นค่าปกติ)

เมื่อขา DTR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะลอจิก "1" เมื่อทำการเปิดพอร์ตและจะมีสถานะเป็น "0" เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ที่ขา DTR จะมีสถานะลอจิก "0" ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานกับโมเด็ม การทำให้ขา DTR เป็นลอจิก “0” จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

2.20.14 RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Request to send (RTS) โดยขา RTS จะเป็นสัญญาณที่ส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งานคำสั่ง

Object.RTSEnable [= value]

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก “0” หรือ “1” ให้ขา RTS โดย

True หมายถึง ให้ขา RTS มีลอจิก “1”

False หมายถึง ให้ขา RTS มีลอจิก “0” (เป็นค่าปกติ)

เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก “1” เมื่อเปิดพอร์ตและมีสถานะลอจิด “0” เมื่อปิดพอร์ต

2.20.15 EOFEnable

เป็นการกำหนดให้ MSComm รอสัญญาณแสดงส่วนท้ายสุดของไฟล์ (End of file: EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

รูปแบบการใช้งานคำสั่ง

Object. EOFEnable [= value]

โดย Value เป็นค่าสถานะ True หรือ False เพื่ออีนาเบิลหรือดิสเอเบิลการทำงานของเหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EOF โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงานด้วย EOF

False หมายถึง เหตุการณ์ OnComm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF (เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มีการตรวจสอบสัญลักษณ์ EOF

2.20.16 CTSHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send (CTS) ได้ว่ามีสถานะลอจิก “0” หรือ “1” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CTSHolding เป็น True ขา CTS จะมีสถานะลอจิกเป็น “1” ถ้าค่า CTSHolding เป็น False ขา CTS จะมีสถานะลอจิกเป็น “0”

รูปแบบการใช้งานคำสั่ง

Object.CTSHolding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อขา CTS เป็นลอจิก “0” (CTSHolding = False) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

2.20.17 CDHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect (DCD) ได้ว่ามาสถานะลอจิกเป็น “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CDHHolding เป็น True ขา DCD จะมีสถานะลอจิกเป็น “1” ถ้าค่า CDHHolding เป็น False ขา DCD จะมีสถานะลอจิกเป็น “0”

รูปแบบการใช้งานคำสั่ง

Object.CDHolding

เมื่อขา DCD เป็นลอจิก “1” (CDHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCDTO (Carrier DetectTimeout Error) และกระตุ้นให้เกิดเหตุการณ์ OnComm

2.20.18 DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR ได้ว่ามีสถานะลอจิก “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า DSRHolding เป็น True ขา DSR จะมีสถานะลอจิกเป็น “1” ถ้าค่า DSRHolding เป็น False ขา DSR จะมีสถานะลอจิกเป็น “0”

รูปแบบการใช้งาน

Object.DSRHolding

เมื่อขา DSR เป็นลอจิก “1” (DSRHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

2.20.19 Handshaking

กำหนดคุณสมบัติและคืนค่ารูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งาน

Object.Handshaking [= value]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบด้วยกันคือ

1. ComNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)

2. ComXOnXOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ComRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request to Send/Clear to Send)

4. ComRTSXOnXOff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request to Send และ XON/XOFF

คุณสมบัติ Handshaking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายใน ระหว่างที่ข้อมูลถูกส่งยังบัฟเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัฟเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัฟเฟอร์ภาครับโปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ Handshaking ช่วยให้ผู้ใช้งานแน่ใจได้ว่าข้อมูลที่ได้รับมานั้นไม่มีการสูญหาย เมื่อบัฟเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือ โอเวอร์โฟลว (overflow) โดยใช้วิธีการตรวจสอบความพร้อมของบัฟเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

2.20.20 Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งานคำสั่ง

Object. Break [= value]

โดย Value 1เป็นค่าบูลีน ถ้า Value = True หมายถึง การส่งสัญญาณ Break ออกไป ถ้า Value = False หมายถึงการเคลียร์สัญญาณ Break

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

ตัวอย่าง เป็นวิธีการส่งสัญญาณ Break ออกไปเป็นช่วงเวลาสั้นๆ ที่ 1/10 ของวินาที

```
MSComm.Break = True                 'Set the Break condition.
Duration! = Timer + 1               'Set duration to 1/10 second
Do Until Timer > Duration!          'Wait for the duration to pass.
Dummy = DoEvents()
Loop
MSComm.Break = False                'Clear the Break condition.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ OnComm

เหตุการณ์ OnComm จะถูกสร้างขึ้นมาเมื่อค่าของคุณสมบัติ CommEvent มีการเปลี่ยนแปลงเพื่อแสดงผลการเปลี่ยนแปลงเหล่านั้นแบบทันทีทันใดหรือแสดงข้อผิดพลาดที่เกิดขึ้น

ตัวอย่างโปรแกรมย่อย OnComm เพื่อนำเหตุการณ์ CommEvent มาแสดง

```
Private Sub MSComm_OnComm ()
    Select Case MSComm1.CommEvent
    'Handle each event or error by placing
    'Code below each case statement
    'Errors
      Case commEventBreak          'A Break was received.
      Case commEventCDTO           'CD (RLSD) Timeout.
      Case commEventCTSTO          'CTS timeout.
      Case commEventDSRTO          'DSR timeout.
      Case commEventFrame          'Framing Error
      Case commEventOverrun        'Data Lost .
      Case commEventRxOver         'Receive buffer overflow .
      Case commEventRxParity       'Parity Error .
Case comEventTxFull                'Transmit buffer full .
    'Events
      Case commEvCD                'Change in the CD line .
      Case commEvCTS               'Change in the CTS line .
      Case commEvDSR               'Change in the CSR line .
      Case commEvRing              'Change in the Ring Indicator .
      Case commEvReceive           'Received Rthreshold # of chars .
      Case commEvSend              'Sthreshold number in the ' transmit
                                    buffer .
Case comEvEof                      'An EOF charater was found in the
                                    input stream .
    End Select
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและออกแบบ

3.1 ตัวถังเรือ

3.1.1 การออกแบบ

ตัวถังเรือที่ทำการออกแบบจะมีลักษณะเป็นรูปทรงกระบอก โดยประกอบด้วย 2 ส่วน คือ ส่วนกลางและส่วนด้านข้าง ส่วนด้านข้างเป็นส่วนที่ใช้ติดตั้งมอเตอร์และใบพัดขับเคลื่อน 2 ชุด เพื่อควบคุมทิศทางการเคลื่อนที่ในแนวระนาบ ส่วนการเคลื่อนที่ในแนวดิ่งใช้หลักการของระบบถังอับเฉา ในส่วนด้านหน้าของตัวเรือออกแบบเป็นโดมทำด้วยแผ่นอาครีลิคที่มีลักษณะโค้งมนและโปร่งแสงเพื่อที่จะให้กล้องวิดีโอสามารถเก็บภาพได้อย่างชัดเจน ตัวถังที่มีความโค้งมลทำให้ลดแรงเสียดทานจากน้ำขณะที่ตัวเรือเคลื่อนที่อยู่ในน้ำ

วัสดุที่ใช้ทำตัวเรือเป็นโลหะอาร์กอน ซึ่งคำนึงถึงว่าตัวเรือคำน้ำนั้นจะต้องสัมผัสกับน้ำอยู่ตลอดเวลา หากใช้ตัวถังเป็นเหล็กแล้วอาจทำให้เกิดสนิมขึ้นได้ง่าย ตัวเรือสามารถถอดประกอบออกได้เป็น 3 ส่วน เพื่อความสะดวกในการติดตั้งชุดควบคุม แผงวงจร และชุดเฟืองของใบพัดเรือ ส่วนที่ถอดประกอบได้ของตัวเรือจะทำการรองด้วยแผ่นยางแข็ง แล้วยึดน็อตให้แน่นเพื่อเป็นการป้องกันไม่ให้น้ำสามารถซึมหรือไหลเข้าสู่ภายในตัวเรือได้ ส่วนแกนของใบพัดจะใช้แหวนยางซิลน์โดยอยู่ระหว่างด้านในและด้านนอกของตัวเรือเพื่อที่จะป้องกันน้ำไหลเข้ามาตามแกนของใบพัดเรือขณะที่ใบพัดหมุน

การออกแบบโครงสร้างของตัวเรือแสดงดังรูปที่ 3.1 ถึง รูปที่ 3.3 เมื่อนำแบบดังกล่าวมาสร้างเป็นตัวเรือ ได้มีการแก้ไขเปลี่ยนแปลงเล็กน้อย เพื่อลดค่าใช้จ่ายด้านวัสดุอุปกรณ์ที่สามารถหาได้ง่ายและราคาถูก

3.1.2 ปริมาตรภายในตัวเรือ

ตัวเรือหลัก	17,671	เซนติเมตร^3
ตัวเรือด้านข้าง 2 ด้าน	5,969	เซนติเมตร^3
โดมด้านหน้าตัวเรือ	4,090	เซนติเมตร^3
กรวยด้านท้ายตัวเรือ	261	เซนติเมตร^3
รวม	27,991	เซนติเมตร^3

ในการออกแบบได้ออกแบบให้ถังอับเฉามีปริมาตร 4,800 เซนติเมตร^3 ดังนั้นปริมาตรภายในตัวเรือจะลดลงเหลือ 23,191 เซนติเมตร^3 (เมื่อปล่อยน้ำเข้าถังอับเฉาจนเต็ม)

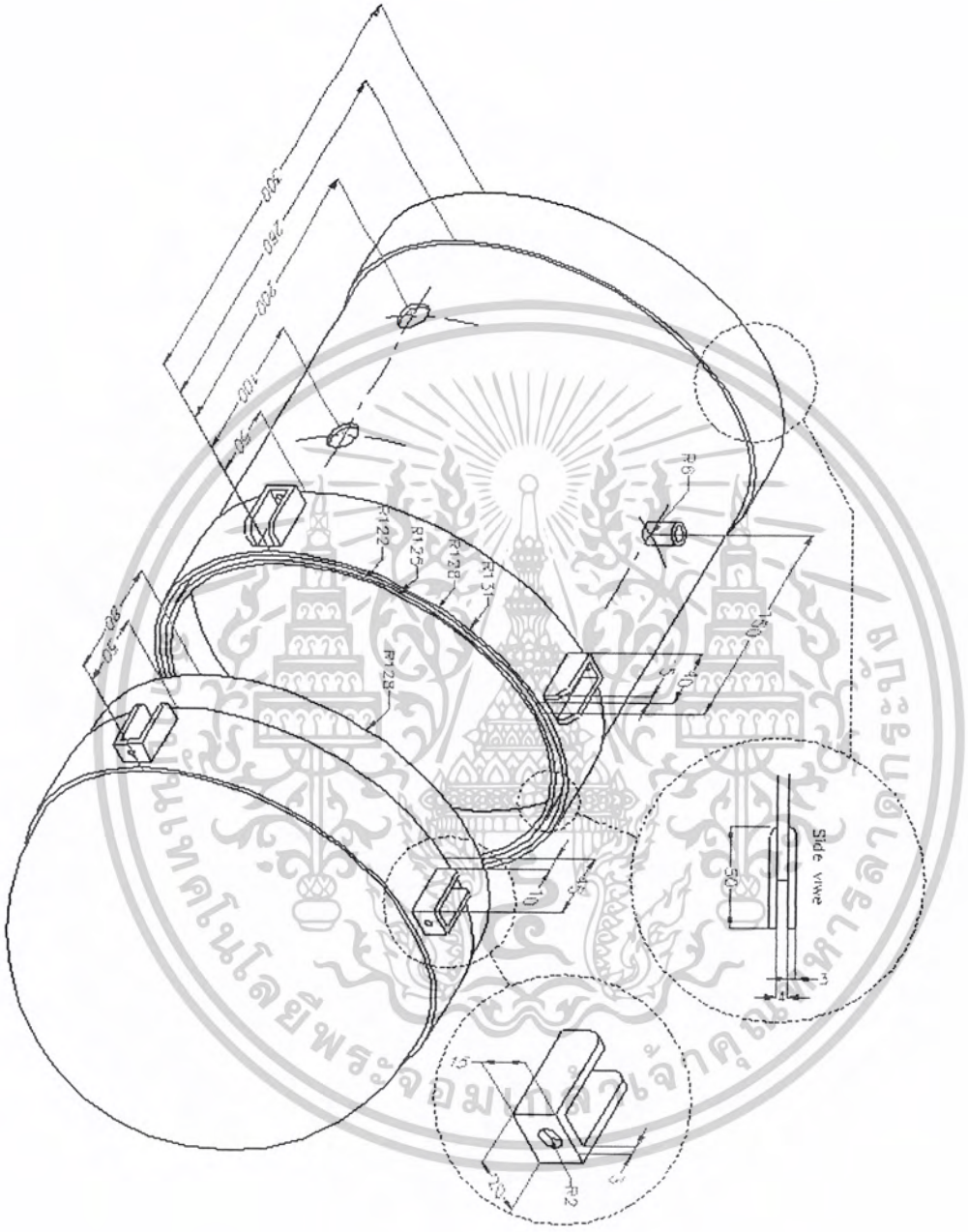
เนื่องจากน้ำบริสุทธิ์ 1 ลิตรมีปริมาตรเท่ากับ 1,000 เซนติเมตร^3 และมีน้ำหนัก 1 กิโลกรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้ำหนักของตัวเรือรวมอุปกรณ์และแบตเตอรี่ เท่ากับ 17.8 กิโลกรัม ดังนั้นต้องใช้น้ำหนักถ่วงอย่างน้อยที่สุด 23.191-17.8 = 5.391 กิโลกรัม ในการทำโครงงานชิ้นนี้เลือกใช้น้ำหนักถ่วง 5.7 กิโลกรัม

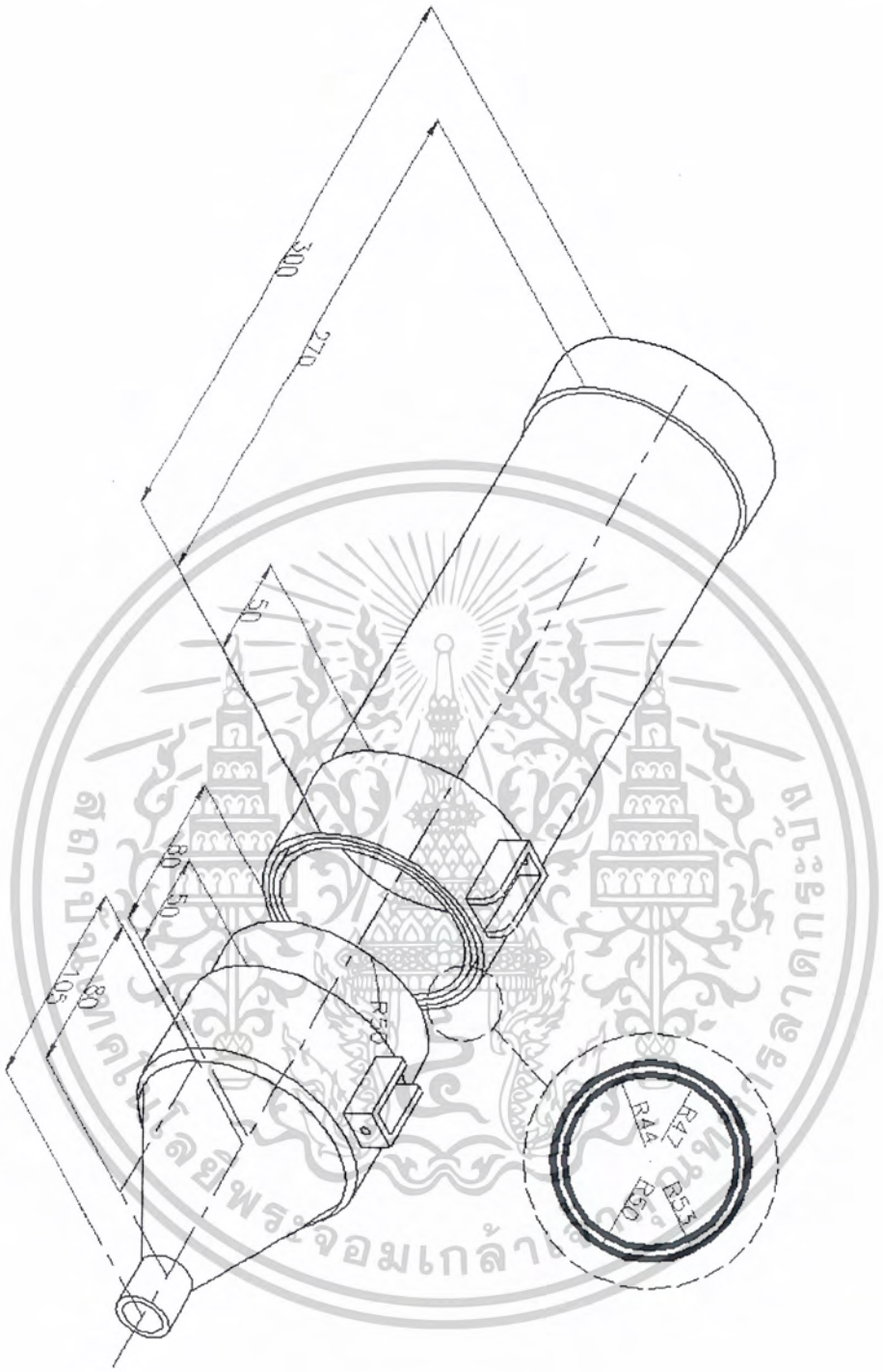
รวมน้ำหนักทั้งหมด เท่ากับ 23.5 กิโลกรัม (น้ำหนักอาจเปลี่ยนแปลงตามปริมาตรของก๊าซที่เหลืออยู่ในกระป๋องเก็บก๊าซ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงโครงสร้างของตัวเรือหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

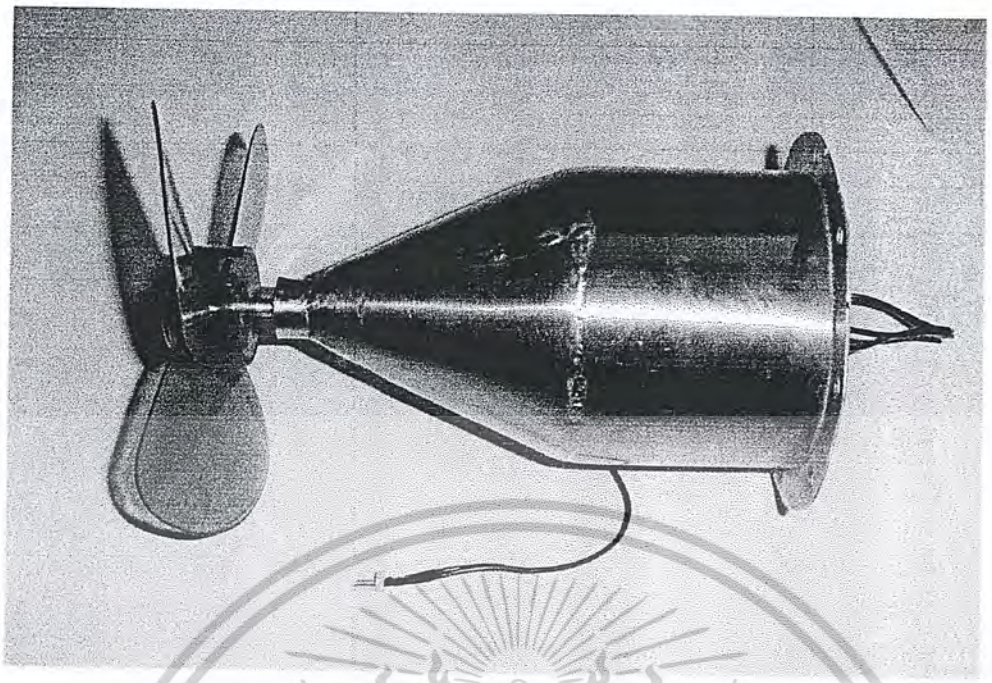


รูปที่ 3.2 แสดงโครงสร้างของตัวเรือด้านข้าง

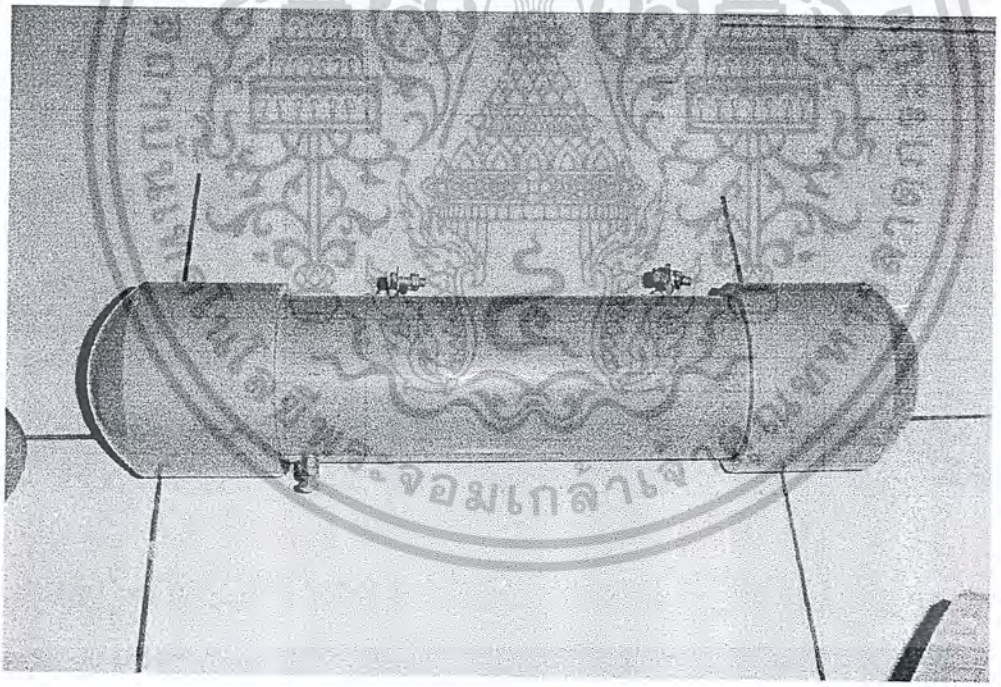
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงโครงสร้างของตัวเรือเมื่อนำมาประกอบเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

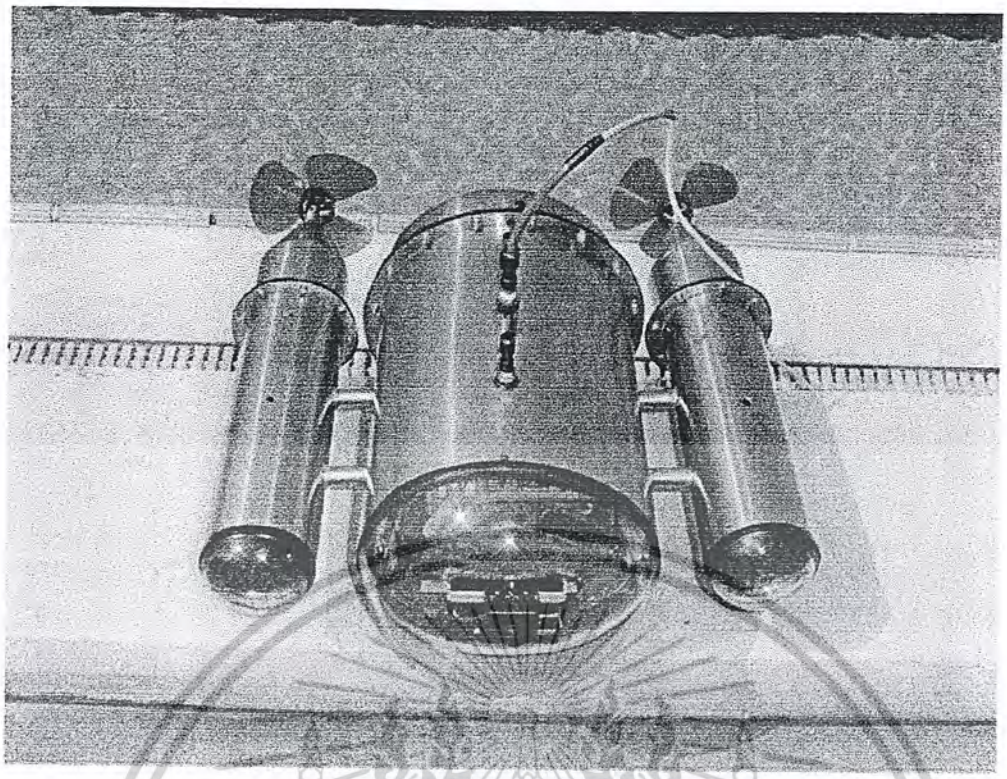


รูปที่ 3.4 แสดงชุดใบพัดที่ใช้ในการขับเคลื่อน (Propeller)



รูปที่ 3.5 แสดงส่วนที่ใช้ในการถ่วงน้ำหนัก ภายในประกอบด้วยแท่งเหล็กที่สามารถเลื่อนไปมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงตัวเรือเมื่อประกอบเสร็จสมบูรณ์

3.2 วงจรไมโครคอนโทรลเลอร์และส่วนควบคุม

วงจรไมโครคอนโทรลเลอร์และส่วนควบคุมแสดงดังรูปที่ 3.8 การทำงานของวงจรแบ่งออกเป็น 6 ส่วนดังนี้

1. ส่วนไมโครคอนโทรลเลอร์ เลือกใช้ PIC16873 เป็นไมโครคอนโทรลเลอร์ที่มีอินพุต/เอาต์พุตพอร์ตมากถึง 22 พอร์ต มีอีอีพรอม 4 กิโลไบต์ เหมาะสำหรับการพัฒนาโครงสร้างต่อไปในอนาคต

2. ส่วนควบคุมการทำงานของมอเตอร์ ใช้ TA2729P เนื่องจากสามารถควบคุมการทำงานของมอเตอร์โดยใช้การสั่งงานในระบบแรงดันไฟแบบ TTL จึงสามารถสั่งการทำงานโดยตรงได้จากไมโครคอนโทรลเลอร์

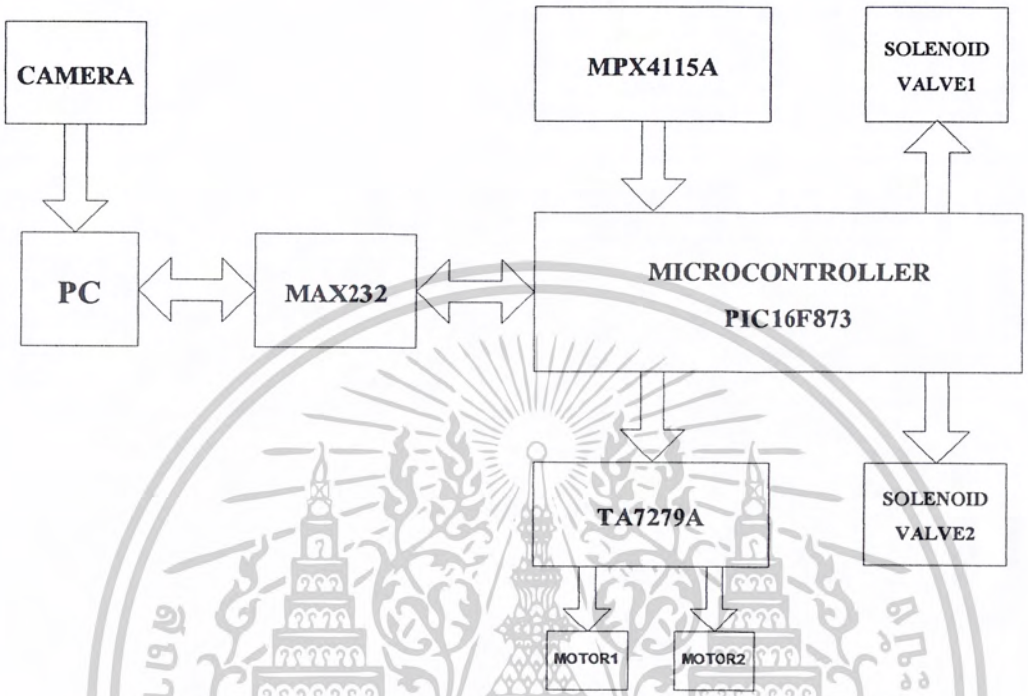
3. ส่วนควบคุมการทำงานของโซลินอย์วาล์ว ใช้ทรานซิสเตอร์เบอร์ BD137 สามารถทนกระแสได้สูงสุด 1.5 A ทนแรงดันได้สูงสุด 80 V ในการออกแบบจะต้องมีไดโอดต่อคร่อมขดลวดของโซลินอย์ดวาล์ว เพื่อป้องกันการเสียหายกับทรานซิสเตอร์อันเนื่องมาจาก Back EMF

4. ส่วน Interface คอมพิวเตอร์ ใช้ IC MAX232CPE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

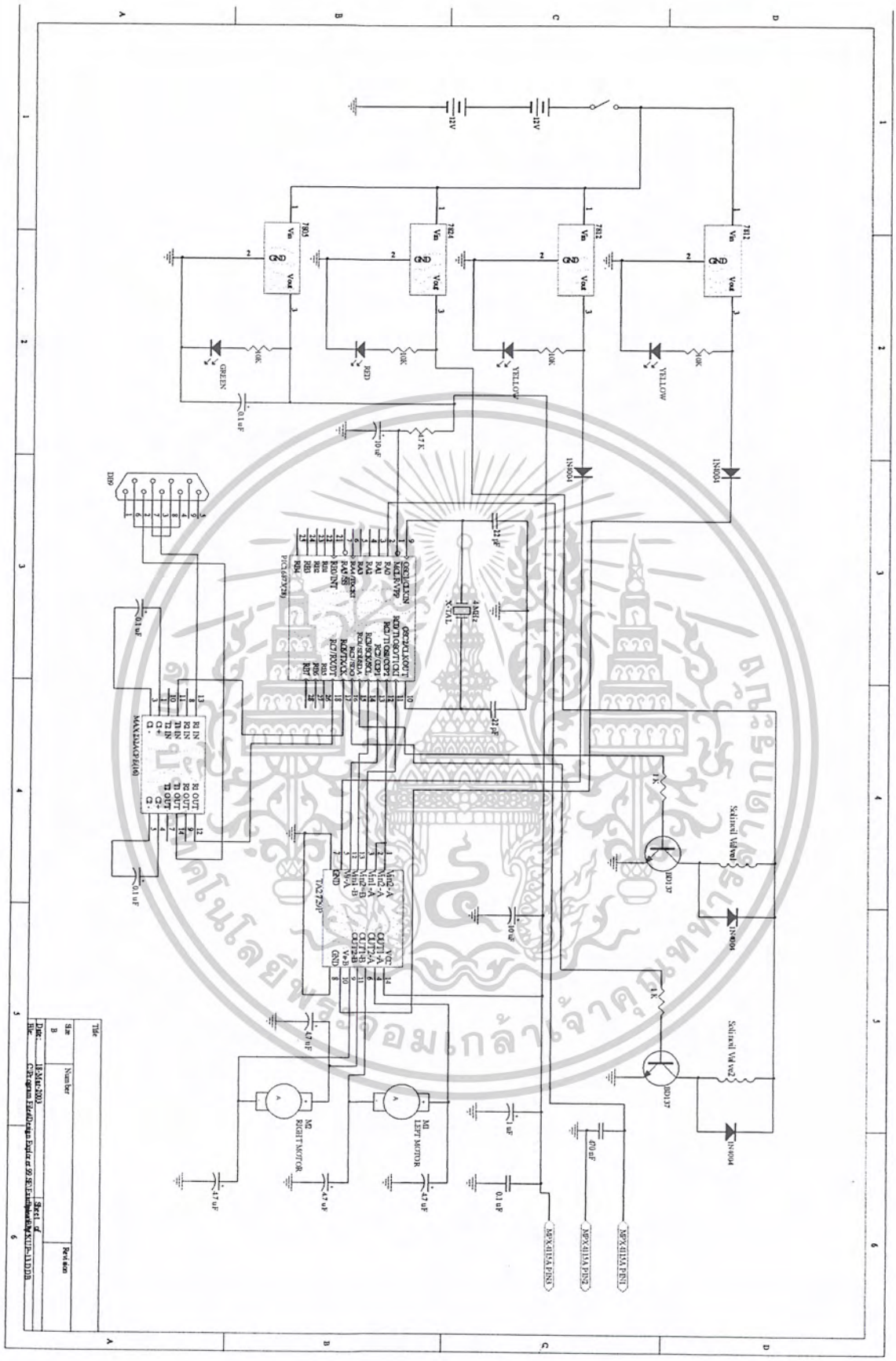
5. ส่วนรักษาระดับแรงดันให้คงที่ ใช้ IC Regulators เบอร์ 7805, 7812, 7824 แต่ละตัวสามารถจ่ายกระแสสูงสุดได้ 1A

6. ส่วนวัดความดันของน้ำเพื่อนำไปแปลงเป็นระดับความลึก ใช้ MPX4115A



รูปที่ 3.7 บล็อกไดอะแกรมแสดงการทำงานของวงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

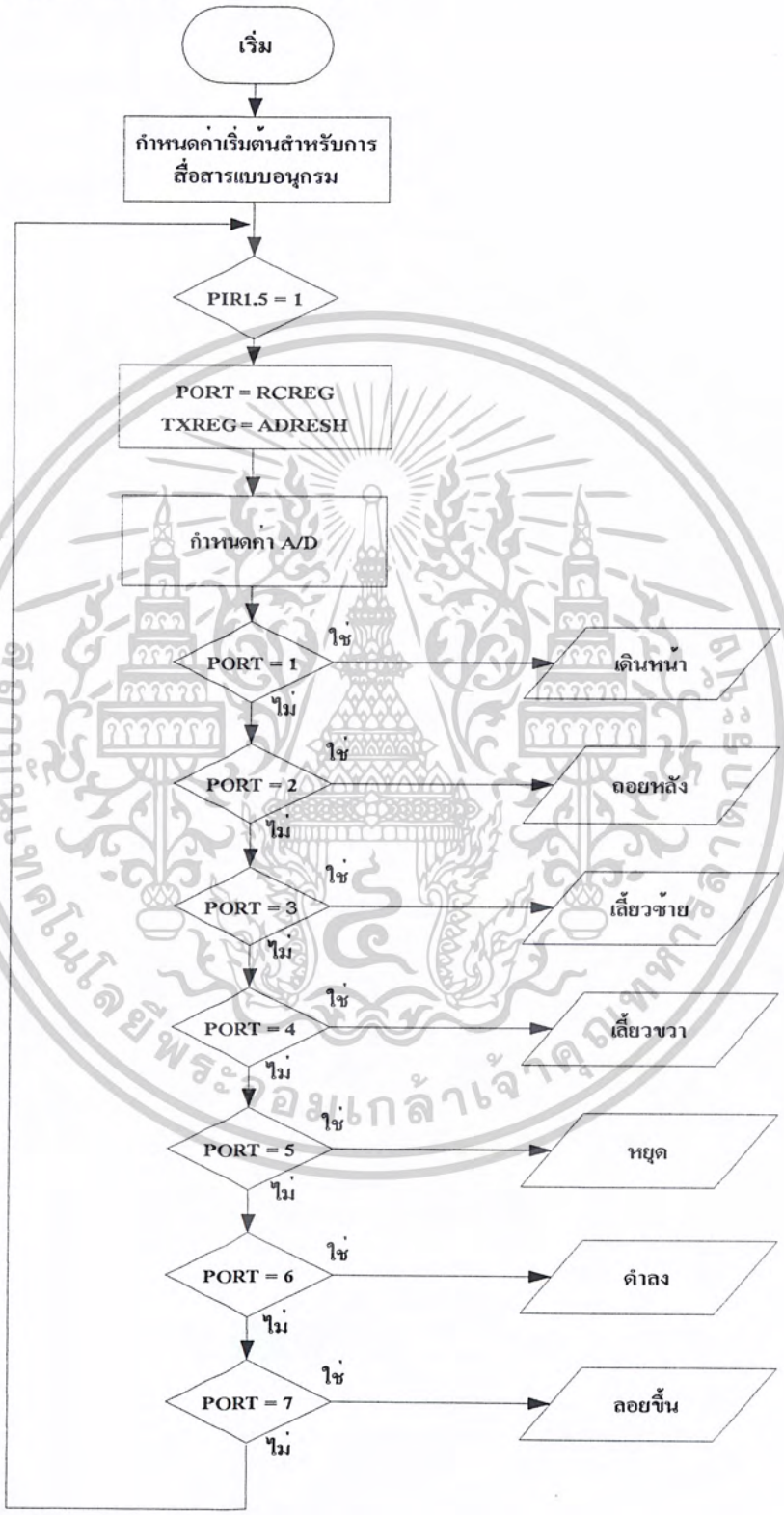


รูปที่ 3.8 วงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Flowchart แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์

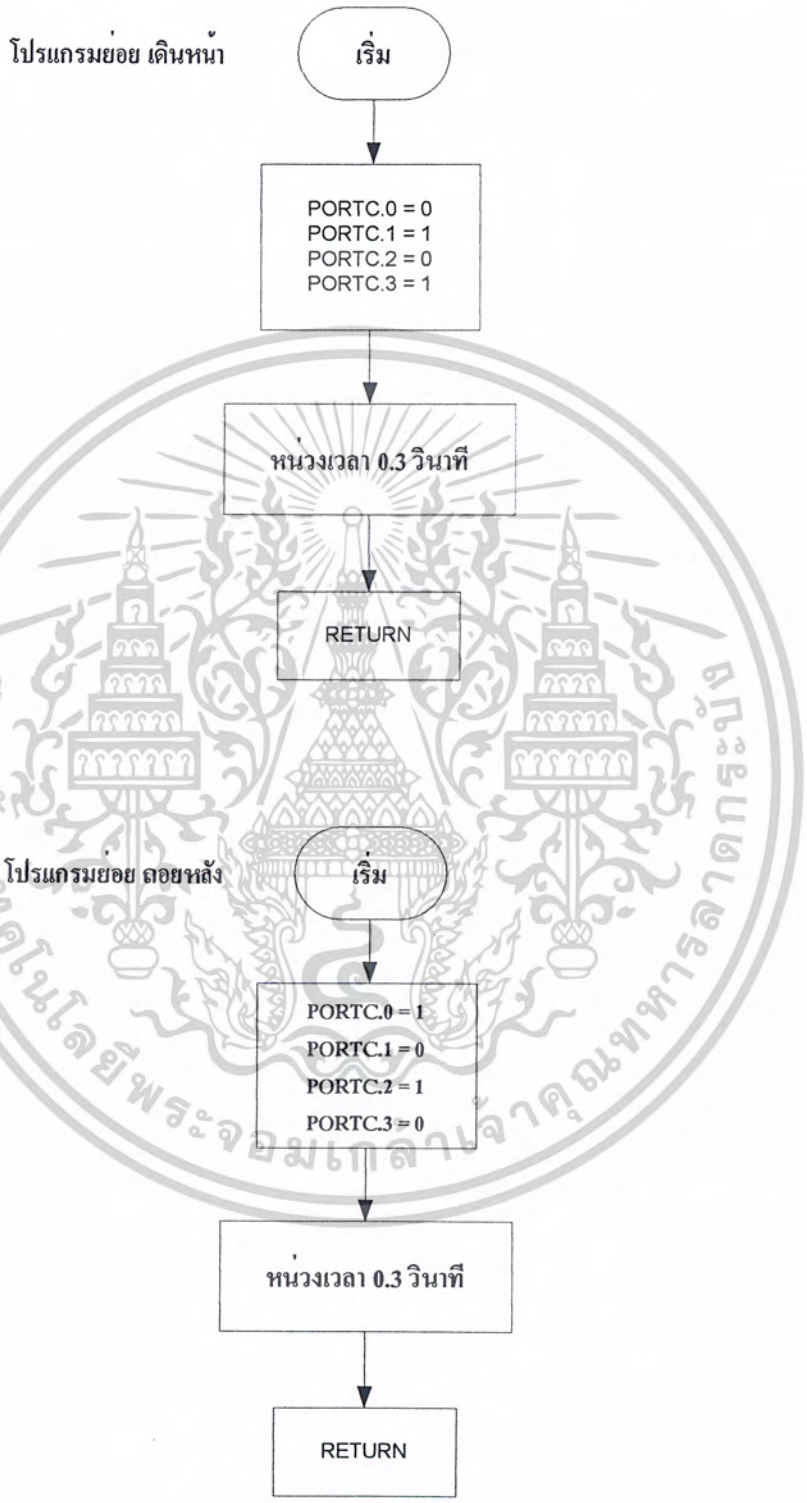
ในส่วนของโปรแกรมควบคุมการทำงานของไมโครคอนโทรเลอร์แสดงในรูปที่ 3.8 รายละเอียดของโปรแกรมจัดแสดงไว้ในภาคผนวก ข



รูปที่ 3.9 Flowchart แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

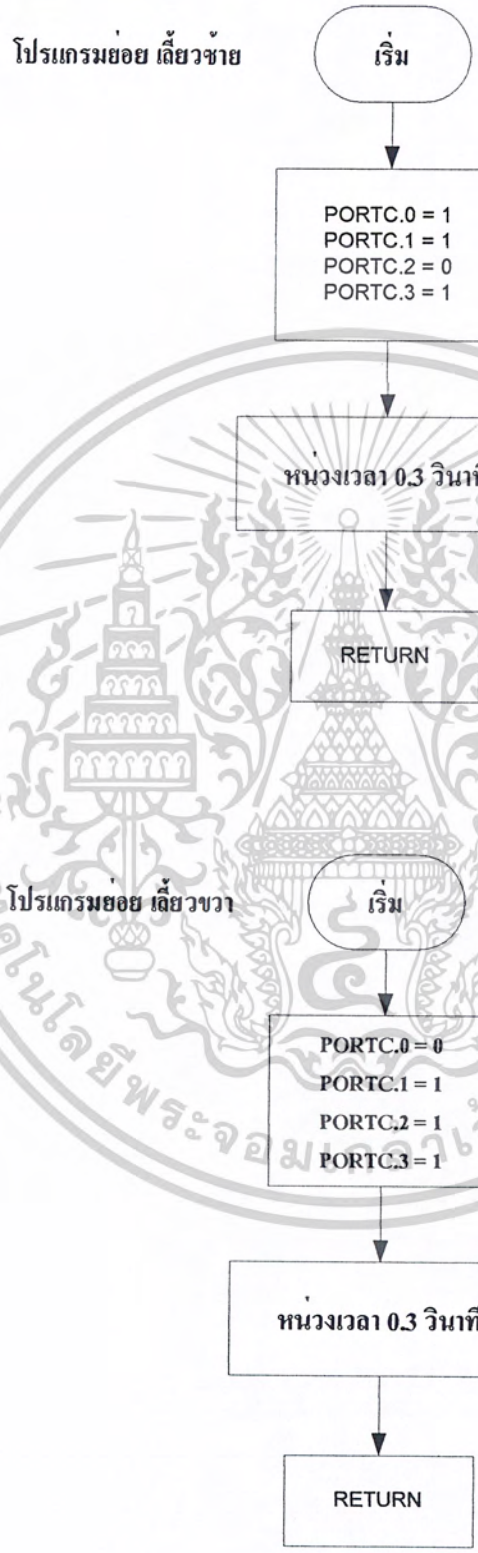
Flowchart แสดงการทำงานของโปรแกรมย่อย



รูปที่ 3.10 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

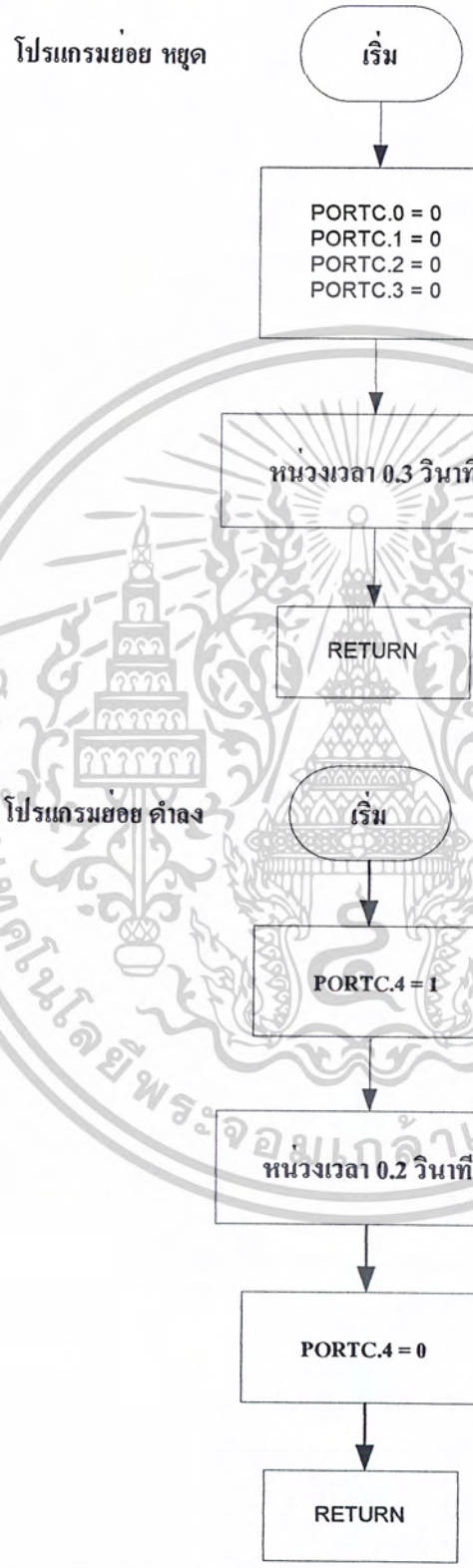
Flowchart แสดงการทำงานของโปรแกรมย่อย



รูปที่ 3.11 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

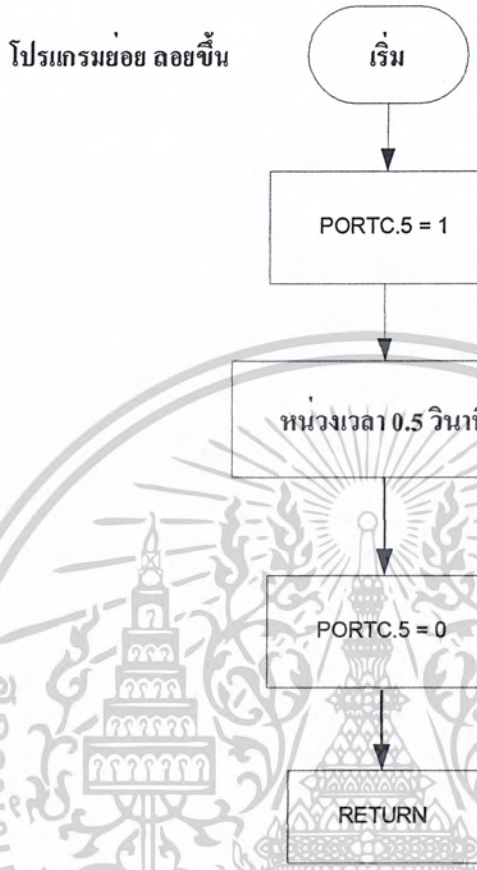
Flowchart แสดงการทำงานของโปรแกรมย่อย



รูปที่ 3.12 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงการทำงานของโปรแกรมย่อย



รูปที่ 3.13 แสดงการทำงานของโปรแกรมควบคุมไมโครคอนโทรเลอร์(โปรแกรมย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โปรแกรมควบคุมเรือดำน้ำ

โปรแกรมควบคุมเรือดำน้ำใช้โปรแกรม Visual Basic โดยมีการติดต่อสื่อสารผ่านพอร์ตอนุกรม
RS-232 ตัวโปรแกรมสามารถแสดงภาพเวลาจริง (Real-time) ขณะที่เรือกำลังเคลื่อนที่อยู่บนผิวน้ำและใต้ผิวน้ำ โดยใช้เครื่องมือ CCDcontrol.ocx ของโปรแกรม Visual Basicให้จับภาพ (Capture) จากกล้องวีดีโอที่ติดตั้งอยู่ในตัวเรือดำน้ำ ไปแสดงผลบนโปรแกรม Visual Basic โดยการแปลงสัญญาณภาพจากอะนาลอกไปเป็นระบบดิจิตอลจะกระทำโดย Video Capture Card นอกจากนี้ตัวไมโครคอนโทรลเลอร์ที่อยู่ในเรือดำน้ำยังส่งข้อมูลที่ได้จาก ไอซีแปลงแรงดัน ไปแสดงผลเป็นระดับความลึก โดยข้อมูลที่ได้รับจากไมโครคอนโทรเลอร์จะถูกนำมาเปรียบเทียบกับค่าที่ได้จากการทดลอง แล้วจึงนำไปแสดงผลเป็นระดับความลึก โดยมีหน่วยเป็นเซนติเมตร รายละเอียดของโปรแกรมสามารถดูได้จากภาคผนวก ก

รูปที่ 3.14 โปรแกรมควบคุมเรือดำน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและสรุป

การทดลองครั้งที่ 1 ทดลองการจมของตัวถัง เรือดำน้ำ

ในการทดลองครั้งนี้ ต้องการทดลองเพียงแค่ว่าจะต้องใช้น้ำหนักเท่าไรเพื่อทำเรือดำน้ำลอยหรือจม และเกิดรอยรั่วภายในตัวถัง เรือดำน้ำ หรือไม่

ผลการทดลอง

จากการทดลองการที่จะทำให้เรือดำน้ำ จมนั้นจะต้องใช้น้ำหนักเพิ่มเข้าไปในตัวถังเรือดำน้ำ โดยน้ำหนักรวมทั้งหมดของตัวเรือประมาณ 27 กิโลกรัม และเกิดการรั่วเล็กน้อยระหว่างรอยต่อของแนวผนึกตัวถัง และได้ทำการแก้ไขแล้วโดยการเปลี่ยนแผ่นยางที่นำมากั้นระหว่างแนวผนึกตัวถัง

การทดลองครั้งที่ 2 ทดลองการสั่งงานมอเตอร์ผ่านคอมพิวเตอร์

ในการทดลองครั้งที่ 2 นี้ใช้โปรแกรม Visual Basic 6 เขียนโปรแกรมในการสั่งให้มอเตอร์ทำงาน โดยจะส่งข้อมูลผ่านทางพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์เป็นตัวสั่งการทำงานของมอเตอร์อีกต่อหนึ่ง

ผลการทดลอง

จากการทดลอง ได้มีการกำหนดรหัสของข้อมูล (เลขฐานสิบ) ที่ส่งออกจากพอร์ตอนุกรมของคอมพิวเตอร์โดย

0 คือ ให้มอเตอร์หมุนตามเข็มนาฬิกาทั้งสองตัว (การเดินหน้า)

1 คือ ให้มอเตอร์ด้านซ้ายหมุนตามเข็มนาฬิกา และมอเตอร์ด้านขวาหยุดการาทำงาน (การเลี้ยวขวา)

2 คือ ให้มอเตอร์ด้านซ้ายหยุดการทำงาน และมอเตอร์ด้านขวาหมุนตามเข็มนาฬิกา (การเลี้ยวซ้าย)

3 คือ ให้มอเตอร์หมุนทวนเข็มนาฬิกาทั้งสองตัว (การถอยหลัง)

4 คือ ให้มอเตอร์หยุดการทำงานทั้งสองตัว

ซึ่งเมื่อทดลองสั่งงานดูนั้น การทำงานของมอเตอร์ได้เป็นไปตามที่ได้กำหนดไว้ข้างต้นอย่างถูกต้อง แต่เมื่อทำการสั่งการเดินหน้าและถอยหลังสลับกันไปอย่างรวดเร็วติดต่อกันหลาย ๆ ครั้ง ทำให้ Regulator เสียหาย

การทดลองครั้งที่ 3 แก้ไขความเสียหายของ Regulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 3 แก้ไขความเสียหายของ Regulator

จากการทดลองครั้งที่ 2 เมื่อสั่งงานให้มอเตอร์ทำการเดินหน้าและถอยหลังอย่างรวดเร็วติดต่อกันหลาย ๆ ครั้ง ทำให้เกิด Back EMF จะส่งผลให้ Regulator เกิดความเสียหาย จึงได้มีการแก้ไขโดย การต่อไดโอดเพิ่มเข้าที่ขา VSS ของ IC ขับมอเตอร์ (TA2729P) โดยเป็นการต่อแบบอนุกรม จาก Output ของ Regulator ผ่านไดโอดซึ่งต่อแบบ Forward Bias แล้วจึงต่อเข้ากับขา VSS ของ IC ที่ขับมอเตอร์

ผลการทดลอง

จากการทดลอง หลังจากที่ได้ทำการแก้ไขแล้วนั้น จึงทำให้สามารถสั่งการเดินหน้าและถอยหลังอย่างรวดเร็วได้โดยไม่เกิดความเสียหายกับ Regulator

การทดลองครั้งที่ 4 การทดลองการสั่งงานโซลินอยด์วาล์วผ่านทางคอมพิวเตอร์

ในการทดลองครั้งที่ 4 นี้ใช้โปรแกรม Visual Basic 6 เขียนโปรแกรมในการสั่งให้โซลินอยด์วาล์วทำงาน โดยส่งข้อมูลผ่านทางพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์เป็นตัวสั่งการทำงานของโซลินอยด์วาล์วอีกต่อหนึ่ง

ผลการทดลอง

จากการทดลอง ได้มีการกำหนดรหัสของข้อมูล (เลขฐานสิบ) ที่ส่งออกจากพอร์ตอนุกรมของคอมพิวเตอร์โดย

5 คือ ให้โซลินอยด์วาล์ว A อยู่ในสถานะเปิด

6 คือ ให้โซลินอยด์วาล์ว B อยู่ในสถานะเปิด

ซึ่งเมื่อทดลองสั่งงานนั้น การทำงานของโซลินอยด์วาล์วได้เป็นไปตามที่ได้กำหนดไว้ข้างต้นอย่างถูกต้อง

การทดลองครั้งที่ 5 การทดลองจับภาพโดยใช้โปรแกรม Visual Basic

ในการทดลองครั้งนี้ได้ทำการเขียนโปรแกรม Visual Basic โดยใช้เครื่องมือ CCDControl.ocx เพื่อนำภาพที่ได้จาก Video Capture Card ไปแสดงผลบนโปรแกรมที่ได้เขียนไว้

ผลการทดลอง

จากการทดลอง โปรแกรมได้แสดงภาพออกหน้าจอแสดงผลเป็นภาพ ขาว-ดำ โดยภาพที่ได้มีความมืดเล็กน้อย จึงได้ทำการปรับแต่งความเข้มของแสงจากโปรแกรม Window Movie Maker จนได้ภาพที่มีความสว่าง และคมชัดในระดับที่สามารถมองเห็นภาพได้อย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 6 การทดลองวัดความดันน้ำตามระดับความลึกต่างๆ กัน

ในการทดลองครั้งนี้ ได้ต่อวงจรไอซี MPX4115A แล้วนำไปใส่ในภาชะนะที่ปิดสนิท ยกเว้นบริเวณส่วนรับแรงดันของไอซีซึ่งจะสัมผัสกับน้ำโดยตรง จากนั้นต่อแรงดันเอาต์พุต เข้ากับขา PORTA.0 ของไมโครคอนโทรเลอร์ ไมโครคอนโทรเลอร์จะรับแรงที่ได้นี้ไปทำการแปลงเป็นค่าในระบบดิจิตอล (A/D) แล้วส่งค่าที่ได้เป็นเลขฐาน 16 ผ่าน พอร์ต USART ของไมโครคอนโทรลเลอร์ไปยัง พอร์ตอนุกรมของคอมพิวเตอร์ คอมพิวเตอร์จะรับข้อมูลที่ได้ไปแปลงเป็นเลขฐาน 10 และแสดงผลบนโปรแกรม Visual Basic เมื่อนำไอซี MPX4115A ลงไปในน้ำที่ระดับความความลึกต่างๆ กัน ค่าที่แสดงผลในโปรแกรมจากการเปลี่ยนไปตามระดับความลึก ได้ตารางนี้

ผลการทดลอง

ค่าที่อ่านได้	ระดับความลึก (ซม.)
210	0
211	10
214	18
216	26
217	32
219	40
220	48
222	54
224	62
226	70
227	78
228	86
230	94
232	102
234	110
238	126
239	135

จากผลการทดลองที่ได้นำค่าเหล่านี้ไปเขียนโปรแกรม เพื่อแสดงค่าระดับความลึกของเรือดำน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 7 ทดลองนำเรือดำน้ำ ลงดำในสระว่ายน้ำเพื่อควบคุมทิศทางการเคลื่อนที่สังเกตความสมดุล ของตัวเรือ

ในการทดลองครั้งนี้ได้สั่งการให้เรือดำน้ำ เคลื่อนที่ไปในทิศทางต่างๆ บนผิวน้ำ โดยสั่งการด้วยโปรแกรม Visual Basic

ผลการทดลอง

จากการทดลองสามารถควบคุมทิศทางการเคลื่อนที่ของเรือดำน้ำ ได้ตามคำสั่ง ในการสั่งการให้เดินหน้า ตัวเรือจะมีการเคลื่อนที่ไปทางด้านซ้ายเล็กน้อย เนื่องจากความเร็วรอบของใบพัดเรือมีความเร็วไม่เท่ากัน และได้ทำการแก้ไขโดยการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์สั่งการ ON มอเตอร์ด้านขวา โดยให้เวลา ON น้อยลง ทำให้สามารถควบคุมเรือไปในทิศทางตรงได้อย่างถูกต้อง

นอกจากนี้จากการสังเกตพบว่าตัวเรือได้เอียงไปด้านหลังเล็กน้อย จึงได้ทำการเลื่อนตัวถ่วงน้ำหนักไปทางด้านหน้า เพื่อปรับให้เรือลอยตัวอย่างสมดุล

การทดลองที่ 8 การทดลองการลอยตัว, การจม, และการเคลื่อนที่ใต้ผิวน้ำ

ในการทดลองครั้งนี้ได้สั่งการให้เรือดำลงใต้ผิวน้ำ สั่งการให้เรือเคลื่อนที่ใต้น้ำตามทิศทางที่กำหนดนอกจากนี้ยังแสดงภาพ ณ เวลาจริง และระดับความลึกของเรือดำน้ำ ด้วย

ผลการทดลอง

จากการทดลองสามารถสั่งการให้เรือดำน้ำ จมลง และเคลื่อนที่ใต้ผิวน้ำตามทิศทางที่กำหนดอีกทั้งยังสามารถแสดงภาพ ณ เวลาจริง และค่าระดับความลึกได้อีกด้วย ทางคณะผู้ทำการทดลองได้มองภาพที่แสดงที่ตัวโปรแกรมควบคุม แล้วทำการควบคุมการเคลื่อนที่ของเรือดำน้ำ ให้เคลื่อนที่ไปในแนวเส้นตรงของสระว่ายน้ำซึ่งปรากฏบนหน้าจอแสดงผล ซึ่งได้ผลตามต้องการ

การทดลองที่ 9 การควบคุมระดับความลึก

ในการทดลองที่ 9 นี้ เป็นการควบคุมระดับความลึกของเรือดำน้ำ ให้เป็นไปตามระดับความลึกที่ต้องการ

ผลการทดลอง

จากการทดลองพบว่า การควบคุมให้ตัวเรือลอยขึ้นสู่ผิวน้ำและดำลงสู่ก้นสระ สามารถควบคุมได้ง่าย แต่การควบคุมให้เรือลอยอยู่ในระดับใดระดับหนึ่งระหว่างผิวน้ำและก้นสระสามารถทำได้อย่างช้าๆ คือจะต้อง ปรับระดับน้ำในถังอับเฉาให้เปลี่ยนแปลงอย่างช้าๆ ตัวเรือก็จะค่อยๆ เปลี่ยนระดับความลึกไปสู่ระดับที่ต้องการ หากมีการปรับระดับของน้ำในถังอับเฉาให้เปลี่ยนแปลงไปอย่างรวดเร็ว ตัวเรือก็จะเปลี่ยนระดับความลึกไปอย่างรวดเร็ว และไม่สามารถควบคุมระดับความลึกของตัวเรือให้หยุดอยู่ ณ ระดับความลึกที่ต้องการได้ ลักษณะเช่นนี้เปรียบได้กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมที่ผลตอบสนองมีการพุ่งเกิน (Overshoot) สูงจนไม่สามารถควบคุมระบบได้ หรือระบบไม่มีเสถียรภาพนั่นเอง

การทดลองที่ 10 การวัดความเร็ว

ในการทดลองนี้เป็นการวัดความเร็วบนผิวน้ำ, ความเร็วใต้น้ำ, ความเร็วในการดำและการลอยขึ้นของเรือดำน้ำ

ผลการทดลอง

จากการทดลองสามารถบันทึกผลการทดลองได้ดังนี้

ความเร็วบนผิวน้ำ	0.12	เมตร/วินาที
ความเร็วใต้น้ำ	0.06	เมตร/วินาที
ความเร็วในการดำลง	0.122	เมตร/วินาที
ความเร็วในการลอยตัวขึ้น	0.168	เมตร/วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากผลการทดลองในบทที่ 4 สามารถสรุปได้ดังนี้

1. การเปลี่ยนแปลงระดับความลึกอย่างรวดเร็ว จะทำให้ไม่สามารถควบคุมระดับความลึกได้

2. จากข้อที่ 1 แสดงให้เห็นว่าเรือดำน้ำใน โครงงานนี้เป็นระบบมีเสถียรภาพเมื่อเปลี่ยนแปลงค่าเป้าหมาย (Set point) อย่างช้าๆ แต่จะเป็นไม่มีเสถียรภาพเมื่อมีการเปลี่ยนแปลง Set point อย่างทันทีทันใด

5.2 ปัญหาที่พบและแนวทางการแก้ไข

การปฏิบัติการเคลื่อนที่ใต้น้ำที่มีระดับความลึกมาก เป็นเวลานานจะส่งผลให้มีน้ำรั่วซึมเข้ามาสะสมในตัวเรือมากขึ้น โดยเฉพาะบริเวณเพลาของใบพัดเรือ เนื่องจากส่วนนี้จะมีการเคลื่อนที่อยู่ตลอดเวลา ประกอบกับแรงดันของจะน้ำจะเพิ่มมากขึ้นเมื่อระดับความลึกเพิ่มขึ้น แนวทางการแก้ไขโดยเปลี่ยนระบบป้องกันน้ำซึม (Water Shield) ให้มีประสิทธิภาพสูงขึ้น เช่น ระบบป้องกันน้ำซึมที่ใช้ในปั๊มน้ำแรงดันสูง (O-Ring)

5.3 แนวทางในการค้นคว้าและพัฒนา

โครงงานที่จัดทำขึ้นนี้ ในส่วนของการควบคุมระดับความลึกเป็นแบบสถิต (Static Diving) คืออาศัยการเปลี่ยนแปลงระดับความลึก จากการปรับปริมาตรของตัวเรือให้เปลี่ยนแปลงไป การปรับปรุงให้เรือดำน้ำสามารถปรับระดับความลึกได้อย่างรวดเร็วยิ่งขึ้นนั้น ทำได้โดยการเพิ่มชุดควบคุมการปรับระดับความลึกแบบเคลื่อนที่ (Dynamic Diving) โดยใช้ชุดใบพัดควบคุมระดับในแนวดิ่ง การเพิ่มชุดใบพัดควบคุมระดับในแนวดิ่งนี้ จะต้องใช้ชุดใบพัดอย่างน้อย 2 ชุดเพื่อความสมดุลและง่ายในการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] สมพงษ์ ใจดี , " ฟิสิกส์ มหาลัย 2 " ลิขสิทธิ์ของสำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย ถนน พญาไทย กรุงเทพ 10330 พิมพ์ครั้งที่ 1: โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย , 2541

[2] ผู้ช่วยศาสตราจารย์ ศรีธน วรศักดิ์โยธิน , " ฟิสิกส์ 1 " ภาควิชาฟิสิกส์ คณะวิทยาสตร์ มหาวิทยาลัยขอนแก่น พิมพ์ครั้งที่ 1: หจก. โรงพิมพ์คลังนานาวิทยา , 2542

[3] มนตรี พิรุณเกษตร , " ฟิสิกส์ ระดับมหาวิทยาลัย " ภาควิชาเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ : บริษัท ซีเอ็ดยูเคชัน จำกัด (มหาชน) 800/43-45 ซอย ตระกูลสุข ถนน อโศกดินแดง เขต ดินแดง กรุงเทพ 10400

[4] ณัฏฐพล วงศ์สุนทรชัย , ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F628 " : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด 3133/53 ซอย สุขุมวิท 101/2 ถนน สุขุมวิท แขวงบางนา เขตบางนา กรุงเทพ 10260

[5] กฤษดา ใจเย็น , อรรถพล บุญยะโภคา , ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการ เชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม " : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด ชั้น 3 อาคารจูราสสิกเฮ้าส์ 2188/199 ซ. รามวิลล่า ถ. รามคำแหง แขวงหัวหมาก เขตบางกะปิ กรุงเทพ 10240

http://www.rov.org/index.html

http://www.heiszwolf.com/subs/tech/tech01.html#Contents

http://www.microchip.com/1010/index.htm

http://www.thaiio.com/

http://perso.club-internet.fr/pyerok/Ballast_EV.htm#HP

http://www.submarine.freehosting.net/

www.motorola.com/brdata/PDFDB/docs/MPX4115A.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมแสดงผลและควบคุมการทำงานของเรือดำน้ำ (Visual Basic)

```
Private Sub Form_Load()
MSComm1.CommPort = 2
MSComm1.Settings = "300,n,8,1"
MSComm1.PortOpen = True
If MSComm1.InBufferCount Then
   Text2.Text = Asc(MSComm1.Input)
   End If
  End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
   Case vbKeyW
    a = 1
    MSComm1.Output = Chr(a)
   Case vbKeyS
    b = 2
    MSComm1.Output = Chr(b)
   Case vbKeyA
    c = 3
    MSComm1.Output = Chr(c)
   Case vbKeyD
    d = 4
    MSComm1.Output = Chr(d)
   Case vbKeyE
    e = 5
    MSComm1.Output = Chr(e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Case vbKeyP
    f = 6
    MSComm1.Output = Chr(f)
   Case vbKeyL
    g = 7
    MSComm1.Output = Chr(g)
    End Select
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
   Case vbKeyP
    a = 8
    MSComm1.Output = Chr(a)
   Case vbKeyL
    b = 8
    MSComm1.Output = Chr(b)
    End Select
End Sub

Private Sub Forward_Click(Index As Integer)
a = 1
MSComm1.Output = Chr(a)
End Sub

Private Sub Revers_Click(Index As Integer)
b = 2
MSComm1.Output = Chr(b)
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Left_Click(Index As Integer)
c = 3
MSComm1.Output = Chr(c)
End Sub

Private Sub Right_Click(Index As Integer)
d = 4
MSComm1.Output = Chr(d)
End Sub

Private Sub break_Click()
e = 5
MSComm1.Output = Chr(e)
End Sub

Private Sub Flote_Click()
f = 6
MSComm1.Output = Chr(f)
End Sub

Private Sub Dive_Click()
g = 7
MSComm1.Output = Chr(g)
End Sub

Private Sub Stop_Click()
h = 8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1.Output = Chr(h)
End Sub

Private Sub Timer1_Timer()
If MSComm1.InBufferCount Then
  a = Asc(MSComm1.Input)
  x = a - 209
  y = 4 * x
  Text2.Text = y
End If
End Sub
```

ค่าคงที่คุณสมบัติของคอนโทรล MSComm Control

ค่าคงที่	ค่า	รายละเอียด
comNone	0	ไม่ใช้การตรวจสอบแฮนด์เชก
comXonXoff	1	ใช้การตรวจสอบแฮนด์เชกแบบ XOn/XOff
comRTS	2	ใช้การตรวจสอบแฮนด์เชกผ่านทางขา RTS และ CTS
comRTSXOnXOff	3	กำหนดการตรวจสอบแฮนด์เชกทั้งแบบ RTS,CTS และ XOn/Xoff

ตารางที่ 1 แสดงค่าคงที่สำหรับคุณสมบัติ Handshake

ค่าคงที่	ค่า	รายละเอียด
comEvSend	1	ส่งค่าเหตุการณ์ (send event)
comEvReceive	2	รับค่าเหตุการณ์ (receive event)
comEvCTS	3	มีการเปลี่ยนแปลงที่ขา CTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

comEvDSR	4	มีการเปลี่ยนแปลงที่ขา DSR
comEvCD	5	มีการเปลี่ยนแปลงที่ขา DCD
comEvRing	6	ตรวจจับสัญญาณกระดิ่งของโทรศัพท์
comEvEOF	7	ตรวจพบตำแหน่งท้ายสุดของไฟล์ (End of file)

ตารางที่ 2 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm

ค่าคงที่	ค่า	รายละเอียด
comInputModeText	0	ข้อมูลที่รับมีคุณสมบัติเป็นข้อความ (ค่าปกติ)
comInputModeBinary	1	ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี่

ตารางที่ 3 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode

ค่าคงที่	ค่า	รายละเอียด
comEventBreak	1001	ได้รับสัญญาณ Break
comEventCTSTO	1002	ขา CTS เกิดไทม์เอาต์
comEventDSRTO	1003	ขา DSR เกิดไทม์เอาต์
comEventFrame	1004	เกิดข้อผิดพลาดที่เฟรมข้อมูล (Framing error)
comEventOverrun	1006	พอร์ตอนุกรมเกิดโอเวอร์รัน (Port overrun)
comEventCDTO	1007	ขา DCD เกิดไทม์เอาต์
comEventRxOver	1008	บัฟเฟอร์รับข้อมูลเกิดโอเวอร์โฟลว
comEventRxParity	1009	เกิดข้อผิดพลาดที่พาริตี้ (Parity error)
comEventTxFull	1010	บัฟเฟอร์ส่งข้อมูลเต็ม

ตารางที่ 4 แสดงค่าคงที่สำหรับคุณสมบัติ Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์

จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังขาสถานะและขาควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายดายมาก โดยใช้คำสั่งเหล่านี้

DTREnable	สำหรับสั่งให้ขา DTR มีลอจิก “0” หรือ “1”
RTSEnable	สำหรับสั่งให้ขา RTS มีลอจิก “0” หรือ “1”
CTSHolding	สำหรับอ่านค่าสถานะจากขา CTS ว่ามีลอจิก “0” หรือ “1”
CDHolding	สำหรับอ่านค่าสถานะจากขา DCD ว่ามีลอจิก “0” หรือ “1”
DSRHolding	สำหรับอ่านค่าสถานะจากขา DSR ว่ามีลอจิก “0” หรือ “1”
Break	สำหรับการสั่งให้ขา Txd มีลอจิก “0” หรือ “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมควบคุมการทำงานไมโครคอนโทรเลอร์

ชุดคำสั่ง (Instruction) ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ตระกูล PIC มีชุดคำสั่งทั้งหมด 35 คำสั่ง ในการทำโครงงานนี้ใช้โปรแกรม PicBasic Pro Compiler ทำการแปลงคำสั่งที่อยู่ในภาษา Basic ซึ่งง่ายในการทำความเข้าและการเขียนโปรแกรม ไปอยู่รูปแบบคำสั่งที่ไมโครคอนโทรเลอร์ตระกูล PIC สามารถนำไปปฏิบัติได้ โดยมีรายละเอียดของโปรแกรมดังนี้

โปรแกรมภาษาเบสิก

```
PORT   VAR   BYTE
A:    TRISC=%10111111
      SPBRG=207
      RCSTA=%10010000
      TXSTA=%00100000

B:    If PIR1.5 = 1 then goto C
      goto B
C:    PORT=RCREG

Include "modedefs.bas"          ' Include serial modes
B1    var   byte


TXREG = ADRESH
SO    con   0                   ' Define serial output pin
ADCON1 = 0                      ' Set PortA 0-3 to analog inputs
ADCON0 = $41                    ' Set A/D to Fosc/8, Channel 0, On
Pause 1                         ' Wait for channel to setup
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ADCON0 = $45            ' Start conversion A/D
Pause 1                 ' Wait for conversion


If PORT = 1 then gosub Forward
If PORT = 2 then gosub Backward
If PORT = 3 then gosub Left
If PORT = 4 then gosub Right
If PORT = 5 then gosub Break
If PORT = 6 then gosub Dive
If PORT = 7 then gosub Float
IF PORT = 8 then gosub SDIVE
goto C

Forward:    high  portc.1
            high   portc.3
            low    portc.0
            low    portc.2
            pause 100
            return
Backward:   high  portc.0
            high   portc.2
            low    portc.1
            low    portc.3
            pause 00
            return
Left:       high   portc.0
            high   portc.1
            low    portc.2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            high   portc.3
            pause 100
            return
Right:      high   portc.1
            low    portc.0
            high   portc.2
            high   portc.3
            pause 100
            return
Break:      low    portc.0
            low    portc.1
            low    portc.2
            low    portc.3
            pause  10
            return
Dive:       high   portc.4
            pause  10
            return
Float:      high   portc.5
            pause  10
            return
SDIVE:      low    portc.4
            low    portc.5
            pause  10
            return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ทำการ Compiler ด้วยโปรแกรม Picbasic Pro Compiler

```
; PicBasic Pro Compiler 2.30, (c) 1998, 2000 microEngineering Labs, Inc. All
Rights Reserved.
PM_USED                 EQU     1

        INCLUDE         "16f873.INC"


; Define statements.
#define                 CODE_SIZE               4

RAM_START                       EQU     0020h
RAM_END                         EQU     00FFh
RAM_BANKS                       EQU     0002h
BANK0_START                     EQU     0020h
BANK0_END                       EQU     007Fh
BANK1_START                     EQU     00A0h
BANK1_END                       EQU     00FFh
EEPROM_START                    EQU     2100h
EEPROM_END                      EQU     217Fh

R0              EQU     RAM_START + 000h
R1              EQU     RAM_START + 002h
R2              EQU     RAM_START + 004h
R3              EQU     RAM_START + 006h
R4              EQU     RAM_START + 008h
R5              EQU     RAM_START + 00Ah
R6              EQU     RAM_START + 00Ch
R7              EQU     RAM_START + 00Eh
R8              EQU     RAM_START + 010h
T1              EQU     RAM_START + 012h
FLAGS                   EQU     RAM_START + 014h
GOP                     EQU     RAM_START + 015h
RM1                     EQU     RAM_START + 016h
RM2                     EQU     RAM_START + 017h
RR1                     EQU     RAM_START + 018h
RR2                     EQU     RAM_START + 019h
_B1                     EQU     RAM_START + 01Ah
_PORT                           EQU     RAM_START + 01Bh
_PORTL                          EQU     PORTB
_PORTH                          EQU     PORTC
_TRISL                          EQU     TRISB
_TRISH                          EQU     TRISC
#define _PIR1_5         PIR1, 005h
#define _PORTC_1                PORTC, 001h
#define _PORTC_3                PORTC, 003h
#define _PORTC_0                PORTC, 000h
#define _PORTC_2                PORTC, 002h
#define _PORTC_4                PORTC, 004h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define _PORTC_5          PORTC, 005h

; Constants.
_T2400                  EQU     00000h
_T1200                  EQU     00001h
_T9600                  EQU     00002h
_T300                   EQU     00003h
_N2400                  EQU     00004h
_N1200                  EQU     00005h
_N9600                  EQU     00006h
_N300                   EQU     00007h
_OT2400                 EQU     00008h
_OT1200                 EQU     00009h
_OT9600                 EQU     0000Ah
_OT300                  EQU     0000Bh
_ON2400                 EQU     0000Ch
_ON1200                 EQU     0000Dh
_ON9600                 EQU     0000Eh
_ON300                  EQU     0000Fh
_MSBPRE                 EQU     00000h
_LSBPRE                 EQU     00001h
_MSBPOST                EQU     00002h
_LSBPOST                EQU     00003h
_LSBFIRST               EQU     00000h
_MSBFIRST               EQU     00001h
_CLS                    EQU     00000h
_HOME                   EQU     00001h
_BELL                   EQU     00007h
_BKSP                   EQU     00008h
_TAB                    EQU     00009h
_CR                     EQU     0000Dh
_unitOn                 EQU     00012h
_unitOff                EQU     0001Ah
_unitsOff               EQU     0001Ch
_lightsOn               EQU     00014h
_lightsOff              EQU     00010h
_dim                    EQU     0001Eh
_bright                 EQU     00016h
_SO                     EQU     00000h
        INCLUDE     "M.MAC"
        INCLUDE     "PBPPIC14LIB"


        LABEL?L     _A
        MOVE?CB     0BFh, TRISC
        MOVE?CB     0CFh, SPBRG
        MOVE?CB     090h, RCSTA
        MOVE?CB     020h, TXSTA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LABEL?L     _B
CMPNE?TCL   _PIR1_5, 001h, L00001
GOTO?L      _C
LABEL?L     L00001
GOTO?L      _B

LABEL?L     _C
MOVE?BB     RCREG, _PORT
MOVE?BB     ADRESH, TXREG
MOVE?CB     000h, ADCON1
MOVE?CB     041h, ADCON0
PAUSE?C     001h
MOVE?CB     045h, ADCON0
PAUSE?C     001h
CMPNE?BCL   _PORT, 001h, L00003
GOSUB?L     _Forward
LABEL?L     L00003
CMPNE?BCL   _PORT, 002h, L00005
GOSUB?L     _Backward
LABEL?L     L00005
CMPNE?BCL   _PORT, 003h, L00007
GOSUB?L     _Left
LABEL?L     L00007
CMPNE?BCL   _PORT, 004h, L00009
GOSUB?L     _Right
LABEL?L     L00009
CMPNE?BCL   _PORT, 005h, L00011
GOSUB?L     _Break
LABEL?L     L00011
CMPNE?BCL   _PORT, 006h, L00013
GOSUB?L     _Dive
LABEL?L     L00013
CMPNE?BCL   _PORT, 007h, L00015
GOSUB?L     _Float
LABEL?L     L00015
CMPNE?BCL   _PORT, 008h, L00017
GOSUB?L     _SDIVE
LABEL?L     L00017
GOTO?L      _C

LABEL?L     _Forward
HIGH?T      _PORTC_1
HIGH?T      _PORTC_3
LOW?T _PORTC_0
LOW?T _PORTC_2
PAUSE?C     064h
RETURN?

LABEL?L     _Backward
HIGH?T      _PORTC_0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HIGH?T      _PORTC_2
LOW?T _PORTC_1
LOW?T _PORTC_3
PAUSE?C     000h
RETURN?

LABEL?L     _Left
HIGH?T      _PORTC_0
HIGH?T      _PORTC_1
LOW?T _PORTC_2
HIGH?T      _PORTC_3
PAUSE?C     064h
RETURN?

LABEL?L     _Right
HIGH?T      _PORTC_1
LOW?T _PORTC_0
HIGH?T      _PORTC_2
HIGH?T      _PORTC_3
PAUSE?C     064h
RETURN?

LABEL?L     _Break
LOW?T _PORTC_0
LOW?T _PORTC_1
LOW?T _PORTC_2
LOW?T _PORTC_3
PAUSE?C     00Ah
RETURN?

LABEL?L     _Dive
HIGH?T      _PORTC_4
PAUSE?C     00Ah
RETURN?

LABEL?L     _Float
HIGH?T      _PORTC_5
PAUSE?C     00Ah
RETURN?

LABEL?L     _SDIVE
LOW?T _PORTC_4
LOW?T _PORTC_5
PAUSE?C     00Ah
RETURN?

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Data sheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Integrated Silicon Pressure Sensor for Manifold Absolute Pressure, Altimeter or Barometer Applications On-Chip Signal Conditioned, Temperature Compensated and Calibrated

MPX4115A
MPXA4115A
SERIES

INTEGRATED PRESSURE SENSOR
15 to 115 kPa (2.2 to 16.7 psi)
0.2 to 4.8 Volts Output

Motorola's MPX4115A/MPXA4115A series sensor integrates on–chip, bipolar op amp circuitry and thin film resistor networks to provide a high output signal and temperature compensation. The small form factor and high reliability of on–chip integration make the Motorola pressure sensor a logical and economical choice for the system designer.

The MPX4115A/MPXA4115A series piezoresistive transducer is a state–of–the–art, monolithic, signal conditioned, silicon pressure sensor. This sensor combines advanced micromachining techniques, thin film metallization, and bipolar semiconductor processing to provide an accurate, high level analog output signal that is proportional to applied pressure.

Figure 1 shows a block diagram of the internal circuitry integrated on a pressure sensor chip.

Features

- 1.5% Maximum Error over 0° to 85°C
- Ideally suited for Microprocessor or Microcontroller–Based Systems
- Temperature Compensated from –40° to +125°C
- Durable Epoxy Unibody Element or Thermoplastic (PPS) Surface Mount Package

Application Examples

- Aviation Altimeters
- Industrial Controls
- Engine Control
- Weather Stations and Weather Reporting Devices

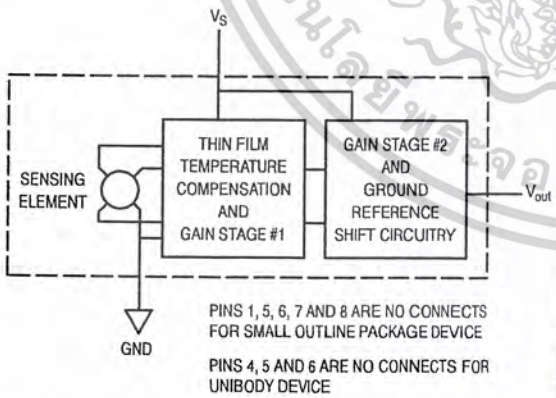
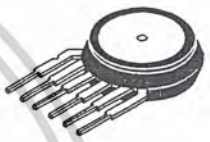
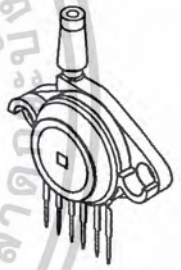


Figure 1. Fully Integrated Pressure Sensor Schematic

UNIBODY PACKAGE



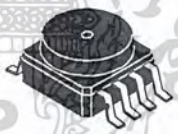
MPX4115A
CASE 867



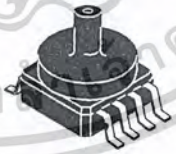
MPX4115AP
CASE 867B

MPX4115AS
CASE 867E

SMALL OUTLINE PACKAGE



MPXA4115A6U
CASE 482



MPXA4115AC6U
CASE 482A

PIN NUMBER			
1	N/C	5	N/C
2	V_S	6	N/C
3	Gnd	7	N/C
4	V_{out}	8	N/C

NOTE: Pins 1, 5, 6, 7, and 8 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the lead.

PIN NUMBER			
1	V_{out}	4	N/C
2	Gnd	5	N/C
3	V_S	6	N/C

NOTE: Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the lead.

REV 4

© Motorola, Inc. 2001

MOTOROLA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIMUM RATINGS(NOTE)

Parametrics	Symbol	Value	Units
Maximum Pressure (P1 > P2)	P_{max}	400	kPa
Storage Temperature	T_{stg}	–40° to +125°	°C
Operating Temperature	T_A	–40° to +125°	°C

NOTE: Exposure beyond the specified limits may cause permanent damage or degradation to the device.

OPERATING CHARACTERISTICS (V_S = 5.1 Vdc, T_A = 25°C unless otherwise noted, P1 > P2. Decoupling circuit shown in Figure 3 required to meet Electrical Specifications.)

Characteristic		Symbol	Min	Typ	Max	Unit
Pressure Range		P_{OP}	15	—	115	kPa
Supply Voltage(1)		V_S	4.85	5.1	5.35	Vdc
Supply Current		I_o	—	7.0	10	mAdc
Minimum Pressure Offset(2) @ V_S = 5.1 Volts	(0 to 85°C)	V_{off}	0.135	0.204	0.273	Vdc
Full Scale Output(3) @ V_S = 5.1 Volts	(0 to 85°C)	V_{FSO}	4.725	4.794	4.863	Vdc
Full Scale Span(4) @ V_S = 5.1 Volts	(0 to 85°C)	V_{FSS}	4.521	4.590	4.659	Vdc
Accuracy(5)	(0 to 85°C)	—	—	—	±1.5	$\%V_{FSS}$
Sensitivity		V/P	—	45.9	—	mV/kPa
Response Time(6)		t_R	—	1.0	—	ms
Output Source Current at Full Scale Output		I_{o+}	—	0.1	—	mAdc
Warm–Up Time(7)		—	—	20	—	ms
Offset Stability(8)		—	—	±0.5	—	$\%V_{FSS}$

NOTES:

1. Device is ratiometric within this specified excitation range.
2. Offset (V_{off}) is defined as the output voltage at the minimum rated pressure.
3. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.
4. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
5. Accuracy is the deviation in actual output from nominal output over the entire pressure range and temperature range as a percent of span at 25°C due to all sources of error including the following:
   - Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
   - Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
   - Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from minimum or maximum rated pressure at 25°C.
   - TcSpan: Output deviation over the temperature range of 0° to 85°C, relative to 25°C.
   - TcOffset: Output deviation with minimum pressure applied, over the temperature range of 0° to 85°C, relative to 25°C.
6. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.
7. Warm–up Time is defined as the time required for the product to meet the specified output voltage after the pressure has been stabilized.
8. Offset Stability is the product's output deviation when subjected to 1000 cycles of Pulsed Pressure, Temperature Cycling with Bias Test.

MECHANICAL CHARACTERISTICS

Characteristics	Typ	Unit
Weight, Basic Element (Case 867)	4.0	grams
Weight, Small Outline Package (Case 482)	1.5	grams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

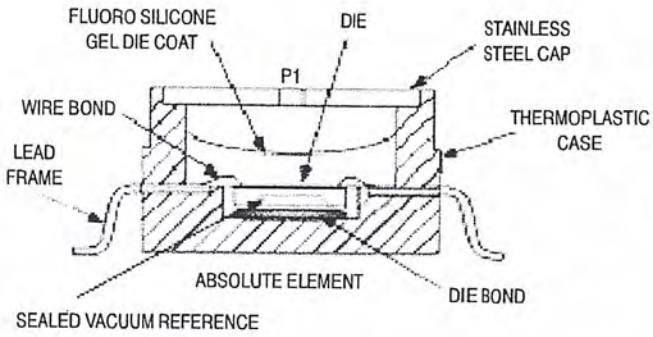


Figure 2. Cross Sectional Diagram SOP (not to scale)

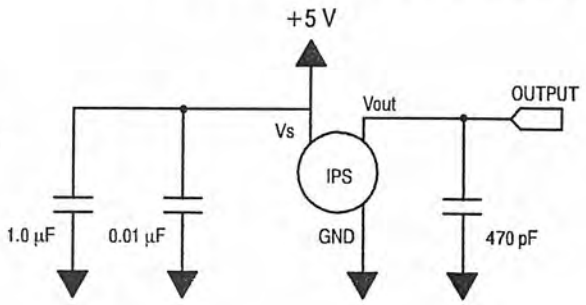


Figure 3. Recommended power supply decoupling and output filtering.
For additional output filtering, please refer to Application Note AN1646.

Figure 2 illustrates the absolute sensing chip in the basic chip carrier (Case 482).

Figure 3 shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

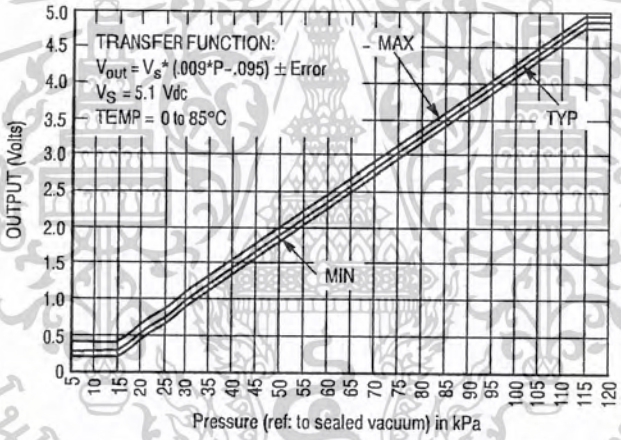


Figure 4. Output versus Absolute Pressure

Figure 4 shows the sensor output signal relative to pressure input. Typical minimum and maximum output curves are shown for operation over 0 to 85°C temperature range. The output will saturate outside of the rated pressure range.

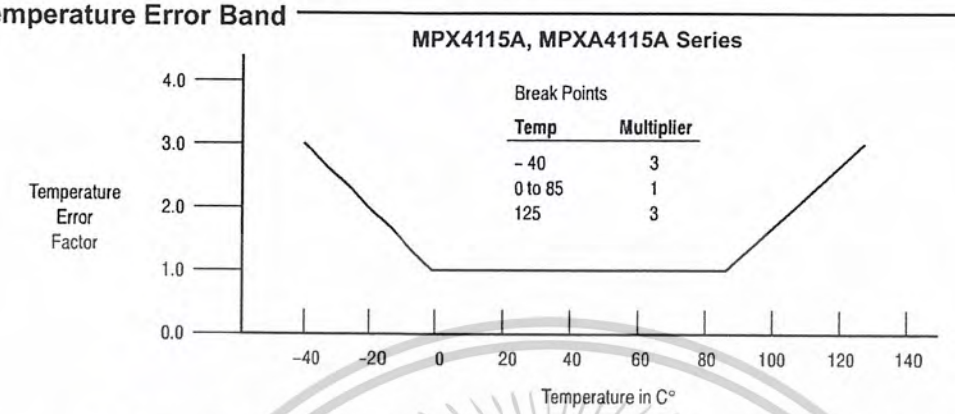
A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the silicon diaphragm. The MPX4115A/MPXA4115A series pressure sensor operating characteristics, internal reliability and qualification tests are based on use of dry air as the pressure media. Media other than dry air may have adverse effects on sensor performance and long–term reliability. Contact the factory for information regarding media compatibility in your application.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transfer Function (MPX4115A, MPXA4115A)

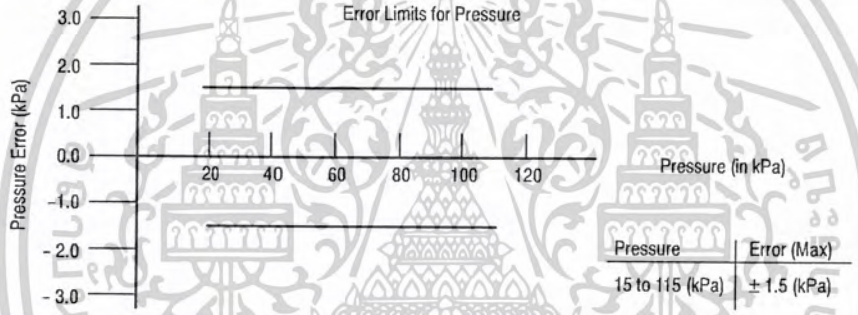
Nominal Transfer Value: $V_{out} = V_S \times (0.009 \times P - 0.095)$
$\pm$ (Pressure Error x Temp. Factor x 0.009 x V_S)
$V_S = 5.1 \pm 0.25$ Vdc

Temperature Error Band



NOTE: The Temperature Multiplier is a linear response from 0°C to –40°C and from 85°C to 125°C

Pressure Error Band



ORDERING INFORMATION — UNIBODY PACKAGE

Device Type	Options	Case No.	MPX Series Order No.	Marking
Basic Element	Absolute, Element Only	867	MPX4115A	MPX4115A
Ported Elements	Absolute, Ported	867B	MPX4115AP	MPX4115AP
	Absolute, Stove Pipe Port	867E	MPX4115AS	MPX4115A

ORDERING INFORMATION — SMALL OUTLINE PACKAGE

Device Type	Options	Case No.	MPX Series Order No.	Packing Options	Marking
Basic Element	Absolute, Element Only	482	MPXA4115A6U	Rails	MPXA4115A
	Absolute, Element Only	482	MPXA4115A6T1	Tape and Reel	MPXA4115A
Ported Element	Absolute, Axial Port	482A	MPXA4115AC6U	Rails	MPXA4115A
	Absolute, Axial Port	482A	MPXA4115AC6T1	Tape and Reel	MPXA4115A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INFORMATION FOR USING THE SMALL OUTLINE PACKAGE (CASE 482)

MINIMUM RECOMMENDED FOOTPRINT FOR SURFACE MOUNTED APPLICATIONS

Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package. With the correct fottprint, the packages will self–align when subjected to a solder reflow process. It is always recommended to design boards with a solder mask layer to avoid bridging and shorting between solder pads.

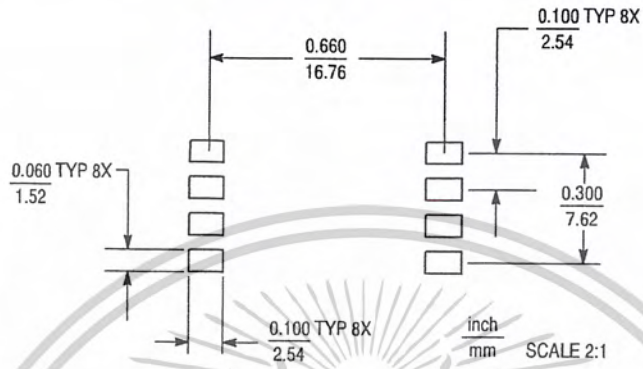
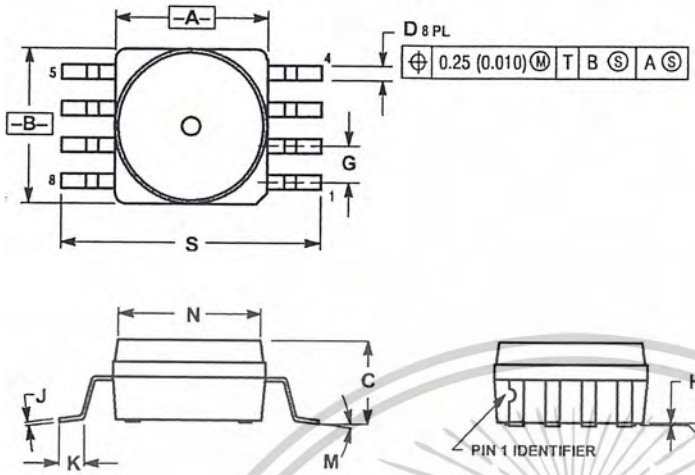


Figure 5. SOP Footprint (Case 482)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

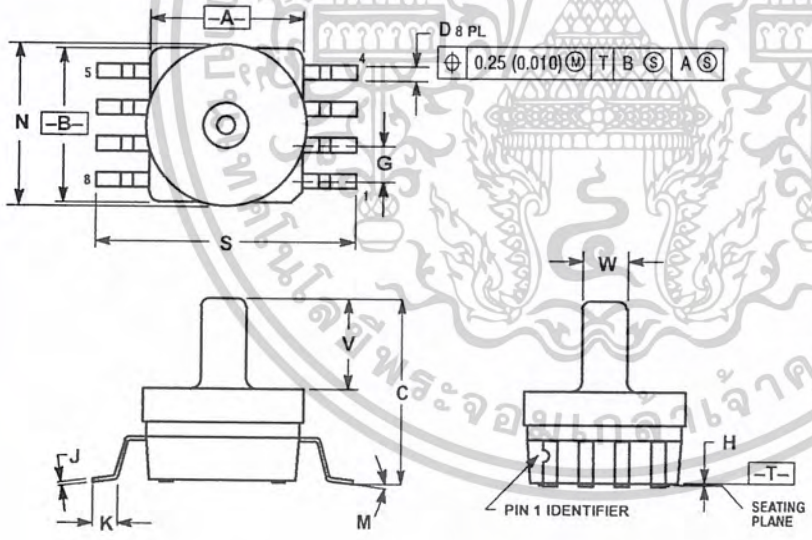
SMALL OUTLINE PACKAGE DIMENSIONS



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006).
5. ALL VERTICAL SURFACES 5° TYPICAL DRAFT.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.415	0.425	10.54	10.79
B	0.415	0.425	10.54	10.79
C	0.212	0.230	5.38	5.84
D	0.038	0.042	0.96	1.07
G	0.100 BSC		2.54 BSC	
H	0.002	0.010	0.05	0.25
J	0.009	0.011	0.23	0.28
K	0.061	0.071	1.55	1.80
M	0°	7°	0°	7°
N	0.405	0.415	10.29	10.54
S	0.709	0.725	18.01	18.41

CASE 482–01
ISSUE O



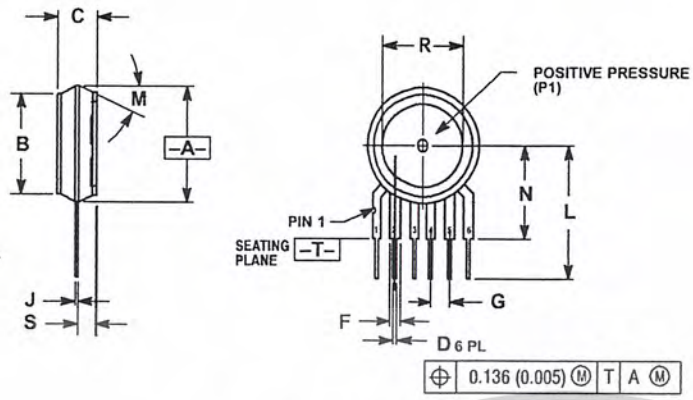
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006).
5. ALL VERTICAL SURFACES 5° TYPICAL DRAFT.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.415	0.425	10.54	10.79
B	0.415	0.425	10.54	10.79
C	0.500	0.520	12.70	13.21
D	0.038	0.042	0.96	1.07
G	0.100 BSC		2.54 BSC	
H	0.002	0.010	0.05	0.25
J	0.009	0.011	0.23	0.28
K	0.061	0.071	1.55	1.80
M	0°	7°	0°	7°
N	0.444	0.448	11.28	11.38
S	0.709	0.725	18.01	18.41
V	0.245	0.255	6.22	6.48
W	0.115	0.125	2.92	3.17

CASE 482A–01
ISSUE A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIBODY PACKAGE DIMENSIONS



NOTES:
9. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
10. CONTROLLING DIMENSION: INCH.
11. DIMENSION -A- IS INCLUSIVE OF THE MOLD STOP RING. MOLD STOP RING NOT TO EXCEED 16.00 (0.630).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.595	0.630	15.11	16.00
B	0.514	0.534	13.06	13.56
C	0.200	0.220	5.08	5.59
D	0.027	0.033	0.68	0.84
F	0.048	0.064	1.22	1.63
G	0.100 BSC		2.54 BSC	
J	0.014	0.016	0.36	0.40
L	0.695	0.725	17.65	18.42
M	30° NOM		30° NOM	
N	0.475	0.495	12.07	12.57
R	0.430	0.450	10.92	11.43
S	0.090	0.105	2.29	2.66

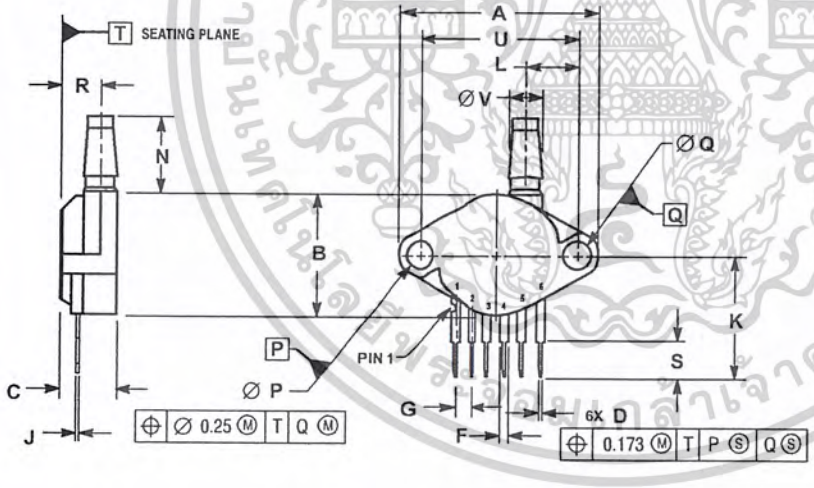
STYLE 1:
PIN 1. VOUT
2. GROUND
3. VCC
4. V1
5. V2
6. VEX

STYLE 2:
PIN 1. OPEN
2. GROUND
3. -VOUT
4. VSUPPLY
5. +VOUT
6. OPEN

STYLE 3:
PIN 1. OPEN
2. GROUND
3. +VOUT
4. +VSUPPLY
5. -VOUT
6. OPEN

CASE 867–08
ISSUE N

BASIC ELEMENT



NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.

	MILLIMETERS	
DIM	MIN	MAX
A	29.08	29.85
B	17.4	18.16
C	7.75	8.26
D	0.68	0.84
F	1.22	1.63
G	2.54 BSC	
J	0.36	0.41
K	17.65	18.42
L	7.37	7.62
N	10.67	11.18
P	3.89	4.04
Q	3.89	4.04
R	5.84	6.35
S	5.59	6.1
U	23.11 BSC	
V	4.62	4.93

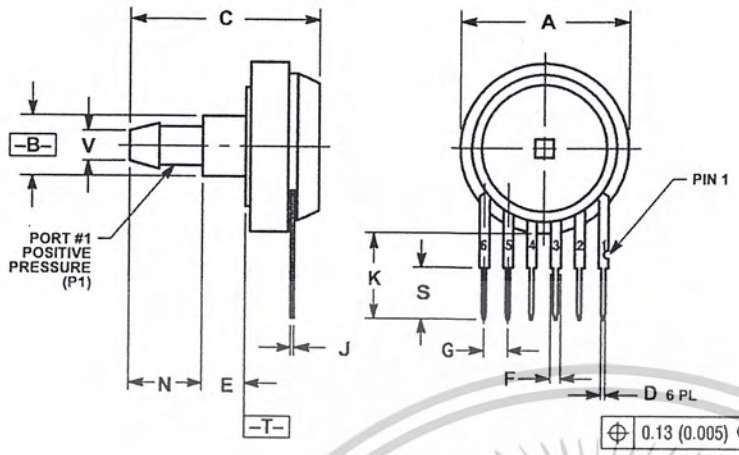
STYLE 1:
PIN 1. V_{OUT}
2. GROUND
3. V_{CC}
4. V1
5. V2
6. V_{EX}

CASE 867B–04
ISSUE F

PRESSURE SIDE PORTED (AP, GP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIBODY PACKAGE DIMENSIONS—CONTINUED



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.690	0.720	17.53	18.28
B	0.245	0.255	6.22	6.48
C	0.780	0.820	19.81	20.82
D	0.027	0.033	0.69	0.84
E	0.178	0.186	4.52	4.72
F	0.048	0.064	1.22	1.63
G	0.100 BSC		2.54 BSC	
J	0.014	0.016	0.36	0.41
K	0.345	0.375	8.76	9.53
N	0.300	0.310	7.62	7.87
S	0.220	0.240	5.59	6.10
V	0.182	0.194	4.62	4.93

STYLE 1:
PIN 1. V_{OUT}
2. GROUND
3. V_{CC}
4. V1
5. V2
6. V_{EX}

CASE 867E–03
ISSUE D

PRESSURE SIDE PORTED (AS, GS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217. 1–303–675–2140 or 1–800–441–2447

Technical Information Center: 1–800–521–6274

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3–20–1, Minami–Azabu. Minato–ku, Tokyo 106–8573 Japan. 81–3–3440–3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2, Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852–26668334

HOME PAGE: http://www.motorola.com/semiconductors/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้