

เรือบังคับวิทยุ
RADIO-CONTROLLED BOAT

พิชานนท์	แช่ฉั่ว
Pichanon	Saechua
ภักดีพร	พลเสน
Phakdiporn	Polsen
มนสิชา	โล่กุลประกิจ
Monsicha	Lokoonprakit

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

เรือบังคับวิทยุ

RADIO-CONTROLLED BOAT

พิชานนท์	แช่ฉั่ว
Pichanon	Saechua
ภักดีพร	พลเสน
Phakdiporn	Polsen
มนสิชา	โล่กุลประกิจ
Monsicha	Lokoonprakit

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

เรือบังคับวิทยุ

RADIO-CONTROLLED BOAT

โดย

พิชานนท์ แซ่ฉั่ว

ภักดีพร พลเสน

มนลิตา โล่กุลประกิจ

อาจารย์ที่ปรึกษา

ผศ.พลผดุง ผดุงกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เรือบังคับวิทยุ

RADIO-CONTROLLED BOAT

ผู้จัดทำ นาย พิชานนท์ แซ่ฉั่ว รหัส 53011121

นางสาว ภัคดีพร พลเสน รหัส 53011201

นางสาว มนสิชา โล่กุลประกิจ รหัส 53011293

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



..... อาจารย์ที่ปรึกษา

(ผศ. พลผดุง ผดุงกุล)

หัวข้อปริญญานิพนธ์	เรือบังคับวิทยุ		
นักศึกษา	นาย พิชานนท์	แซ่ฉั่ว	รหัส 53011121
	นางสาว ภัคดีพร	พลเสน	รหัส 53011201
	นางสาว มนสิชา	โล่กุลประกิจ	รหัส 53011293
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์		
ปีการศึกษา	2556		

บทคัดย่อ

รายงานฉบับนี้อธิบายถึงการออกแบบและสร้าง เรือบังคับวิทยุ โดยเราจะนำโมดูลมาใช้ในการรับ-ส่ง สัญญาณคลื่นวิทยุขนาด 433.92 MHz จะส่งสัญญาณ Digital ผ่านโมดูลโดยใช้การประมวลผลในตัวส่งผ่าน HT12E และตัวรับประมวลผลผ่านทาง HT12D อีกทั้งนำมาใช้ในการควบคุมการเคลื่อนที่ของเรือทั้งเดินหน้า ถอยหลัง หยุด เพิ่มความเร็ว เลี้ยวซ้าย และเลี้ยวขวา ในส่วนของซอฟต์แวร์ที่นำมาใช้ในการควบคุมการทำงานของ เรือบังคับวิทยุ มี 2 โปรแกรมคือ 1.CCS PIC C Compiler ใช้เขียน microcontroller ในการควบคุมในส่วนของ DC Drive Motor และ PWM Servo Motor และ 2.Proteus นำมาใช้ในการเขียนจำลองในส่วนของฮาร์ดแวร์ สามารถนำไปใช้ประโยชน์ในการช่วยเหลือ หรือนำไปสำรวจในสถานที่เกิดอุทกภัยในที่ที่เราไม่สามารถเข้าถึงได้

Thesis Title	Radio-controlled boat		
Student	Mr.Pichanon	Saechua	ID 53011121
	Ms.Phakdiporn	Polsen	ID 53011201
	Ms.Monsicha	Lokoonprakit	ID 53011293
Degree	Bachelor of Engineering		
Program	Electronics Engineering		
Years	2012		
Thesis Advisor	Asst.Prof.Polphadung Phadungkul		

Abstract

This report describes a design and construction of radio-controlled boat. We use module to receive and transmit radio wave signal in frequency of 433.92 MHz and transmit digital signal that use to evaluate the receiver-transmitter with HT12D-HT12E for and control the direction of boat to go forward, go backwards, stop, speed, turn left and turn right. We use 2 software consist of : 1) CCS PIC C Compiler Microcontroller for controlling DC Drive Motor and PWM Servo Motor. 2) Proteus for simulated circuits. However, It has use to help or survey water flood in the area that human can't go.

กิตติกรรมประกาศ

โครงการนี้ได้จัดทำขึ้นจนประสบความสำเร็จเนื่องจากได้รับความช่วยเหลือจากหลายๆฝ่ายด้วยกัน โดยบุคคลแรกที่ขอกล่าวถึงคือ ผศ. พลผดุง ผดุงกุล อาจารย์ที่ปรึกษาของโครงการนี้ที่คอยให้ความช่วยเหลือพวกเราทุกเรื่อง และคอยให้คำแนะนำให้ความรู้พร้อมทั้งช่วยแก้ไขปัญหา พวกเราต้องขอขอบพระคุณอาจารย์เป็นอย่างสูง

บุคคลที่มีพระคุณกับเราเป็นอย่างยิ่ง คือ บิดา มารดา ที่ทำให้เรามีวันนี้ ซึ่งพวกท่านคือกำลังใจที่มีค่า ยิ่งเพราะท่านได้ให้การเลี้ยงดูที่ดีให้การอบรมสั่งสอน และคอยสนับสนุนไม่ห่างและเอาใจใส่จนทำให้เราเป็นคนดีในวันนี้ พวกเราจะขอสำนึกในพระคุณอันสูงค่านี้ตลอดไป

และสุดท้ายต้องขอบคุณเพื่อนๆ ที่คอยให้ความช่วยเหลือให้คำแนะนำที่เป็นประโยชน์พวกเราขอขอบคุณเพื่อนๆทุกคนนะ

พิชานนท์ แซ่ฉั่ว

ภักดิพร พลเสน

มนสิชา โสกุลประกิจ

สารบัญ

	<u>หน้า</u>
บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	VI
สารบัญตาราง.....	X
บทที่ 1 บทนำ.....	1
1.1 วัตถุประสงค์ของโครงงาน.....	1
1.2 ขอบเขตของโครงงาน.....	1
1.3 ผลที่คาดว่าจะได้รับ.....	1
บทที่ 2 ทฤษฎีและหลักการทำงานที่นำมาประยุกต์ใช้.....	2
2.1 มอเตอร์กระแสตรง (DC MOTOR).....	2
2.1.1 หลักการทำงานของระบบมอเตอร์ไฟฟ้ากระแสตรง (DC motor).....	3
2.1.2 การควบคุมความเร็วของมอเตอร์กระแสตรง.....	5
2.1.3 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM).....	5
2.2 การควบคุมแบบ PWM (Pulse Width Modulation).....	6
2.2.1 การทำงานของสัญญาณ PWM.....	6
2.2.2 ประโยชน์ของ PWM ในการควบคุมความเร็วมอเตอร์.....	7
2.3 Servo Motor	8
2.3.1 ความแตกต่างของเซอร์โวมอเตอร์กับมอเตอร์ชนิดอื่นๆ.....	8
2.3.2 การควบคุมทำงานของเซอร์โวมอเตอร์ ด้วย PWM.....	8

2.4	การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog, D/A).....	10
2.4.1	การมอดูเลตเชิงเลขทางแอมพลิจูด (Amplitude – Shift Keying : ASK).....	10
2.5	การส่งสัญญาณข้อมูล.....	11
2.5.1	การส่งข้อมูลแบบขนาน (Parallel Transmission).....	11
2.5.2	การส่งข้อมูลแบบอนุกรม (Serial Transmission).....	12
บทที่ 3	ทฤษฎีและหลักการทำงานที่นำมาประยุกต์ใช้.....	15
3.1	Flow Chart.....	15
3.1.1	Flow chart การทำงานของรีบบังคับวิทยุ.....	15
3.1.2	Flow chart การทำงานในส่วนของภาคส่ง.....	16
3.1.3	Flow chart การทำงานในส่วนของภาครับ.....	17
3.2	ศึกษาค้นคว้าและออกแบบในส่วนของวงจร.....	18
3.2.1	วงจรภาคส่ง.....	18
3.2.2	วงจรภาครับ.....	19
3.3	คำนวณค่าที่ใช้ในการเขียนโปรแกรม.....	20
3.3.1	คำนวณหาคาบ.....	20
3.3.2	คำนวณหาค่า%Duty Cycle.....	20
บทที่ 4	การทดลอง.....	21
4.1	ผลการทดสอบวงจรขับมอเตอร์จาก Microcontroller.....	21
4.2	ผลการทดสอบ PWM Servo Motor.....	29

4.3 ผลการทดสอบวงจรขับมอเตอร์จาก Microcontroller.....	33
4.3.1 ยังไม่กดสวิตช์ (1111).....	33
4.3.2 กดสวิตช์ Go (1110).....	34
4.3.3 กดสวิตช์ Back (1101).....	35
4.3.4 กดสวิตช์ Speed (1100).....	36
4.3.5 กดสวิตช์ Stop (1011).....	37
4.3.6 กดสวิตช์ left (1010).....	38
4.3.7 กดสวิตช์ right (1001).....	39
4.3.8 กดสวิตช์ reset (0111).....	40
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง.....	41
ภาคผนวก.....	42
ผ.ก โปรแกรมควบคุม.....	42
ผ.ก.1 โปรแกรมควบคุมภาครับ.....	42
ผ.1 ข้อมูลเกี่ยวกับ PIC16F877A.....	46
ผ.2 ข้อมูลเกี่ยวกับ L293D.....	47
ผ.3 ข้อมูลเกี่ยวกับ RWS-BS Series Module.....	48
ผ.4 ข้อมูลเกี่ยวกับ TWS-BS Series Module.....	49
ผ.5 ข้อมูลเกี่ยวกับ HT12E.....	50
ผ.6 ข้อมูลเกี่ยวกับ HT12D.....	53
ผ.7 ซอฟต์แวร์ที่ใช้ในการทำงาน.....	56
ผ.7.1 โปรแกรม Proteus.....	56
ผ.7.2 โปรแกรม CCS PIC C Compiler.....	57

ผ.7.3 โปรแกรม PICkit 2.....	58
ผ.7.4 โปรแกรม Altium Designer 10.....	59
เอกสารอ้างอิง.....	60

สารบัญรูปภาพ

หน้า

รูปที่ 2.1	แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง.....	3
รูปที่ 2.2	แสดงรูปวงจรของ H-Bridge Switching.....	3
รูปที่ 2.3	การไต่ร้ฟมอเตอร์ในทิศตามเข็มนาฬิกา.....	4
รูปที่ 2.4	การไต่ร้ฟมอเตอร์ในทิศทวนเข็มนาฬิกา.....	4
รูปที่ 2.5	แสดงความกว้างของพัลส์ขนาดต่างๆและค่าดีวตี้ไซเคิลของช่วงพัลส์ที่มีความถี่คงที่.....	6
รูปที่ 2.6	แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน.....	7
รูปที่ 2.7	เซอร์โวมอเตอร์และองค์ประกอบของเซอร์โวมอเตอร์.....	8
รูปที่ 2.8	การควบคุมการทำงานของเซอร์โวมอเตอร์.....	9
รูปที่ 2.9	PWM Servo Motor.....	10
รูปที่ 2.10	การแปลงสัญญาณดิจิทัลเป็นอนาล็อกแบบ ASK.....	10
รูปที่ 2.11	Block diagram ของ Data Transmission.....	11
รูปที่ 2.12	การส่งข้อมูลแบบขนาน (Parallel Transmission).....	12
รูปที่ 2.13	การส่งข้อมูลแบบอนุกรม (Serial Transmission).....	12
รูปที่ 2.14	การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous data transmission).....	13
รูปที่ 2.15	การใช้บิตซิงค์ (Sync bit).....	14
รูปที่ 2.16	การใช้อักขระซิงค์ (Sync characters).....	14
รูปที่ 2.17	การส่งข้อมูลแบบซิงโครนัส (Synchronous data transmission).....	14
รูปที่ 3.1	Flow Chart การทำงานของเรือบังคับวิทยุ.....	15
รูปที่ 3.2	Flow Chart การทำงานในส่วนของภาคส่ง.....	16
รูปที่ 3.3	Flow Chart การทำงานในส่วนของภาครับ.....	17

รูปที่ 3.4	วงจรรภาคส่ง.....	18
รูปที่ 3.5	วงจรรภาครับ.....	19
รูปที่ 4.1	การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s).....	21
รูปที่ 4.2	การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s).....	22
รูปที่ 4.3	การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s).....	23
รูปที่ 4.4	การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s).....	24
รูปที่ 4.5	การเกิด Pulse ที่ $T_o = 1000 \mu$ (คาบ/s).....	25
รูปที่ 4.6	การเกิด Pulse ที่ $T_o = 1500 \mu$ (คาบ/s).....	26
รูปที่ 4.7	การเกิด Pulse ที่ $T_o = 2000 \mu$ (คาบ/s).....	27
รูปที่ 4.8	การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s).....	28
รูปที่ 4.9	การเกิด Pulse ที่ “ 0° ”	29
รูปที่ 4.10	การเกิด Pulse ที่ “ 90° ”	30
รูปที่ 4.11	การเกิด Pulse ที่ “ -90° ”	31
รูปที่ 4.12	สัญญาณจากขา data ตัวส่ง ขณะที่ยังไม่กดสวิตช์ (1111).....	33
รูปที่ 4.13	สัญญาณจากขา data ตัวรับ ขณะที่ยังไม่กดสวิตช์ (1111).....	33
รูปที่ 4.14	สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Go (1110).....	34
รูปที่ 4.15	สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Go (1110).....	34
รูปที่ 4.16	สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Back (1101).....	35
รูปที่ 4.17	สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Back (1101).....	35
รูปที่ 4.18	สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Speed (1100).....	36
รูปที่ 4.19	สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Speed (1100).....	36
รูปที่ 4.20	สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Stop (1011).....	37
รูปที่ 4.21	สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Stop (1011).....	37

รูปที่ 4.22 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ left (1010).....	38
รูปที่ 4.23 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Left (1010).....	38
รูปที่ 4.24 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Right (1001).....	39
รูปที่ 4.25 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Right (1001).....	39
รูปที่ 4.26 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Reset (1011).....	40
รูปที่ 4.27 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Reset (1011).....	40
รูปที่ ผ.1.1 การจัดเรียงขาของไมโครคอนโทรลเลอร์ PIC16F877A.....	46
รูปที่ ผ.2.1 โครงสร้างของ L293D.....	47
รูปที่ ผ.2.2 logic diagram ของ L293D.....	48
รูปที่ ผ.3.1 แสดงการจัดเรียงขาของ RWS-BS Module.....	49
รูปที่ ผ.4.1 แสดงการจัดเรียงขาของ TWS-BS Module.....	50
รูปที่ ผ.5.1 โครงสร้างของ HT12E.....	50
รูปที่ ผ.5.2 Block diagram ของ HT12E.....	51
รูปที่ ผ.5.3 การส่งข้อมูลของ HT12E.....	51
รูปที่ ผ.5.4 Flow chart การทำงานของ HT12E.....	52
รูปที่ ผ.5.5 แสดงรูปคลื่นของ Address/data bit จาก HT12E.....	52
รูปที่ ผ.5.6 แสดงการประยุกต์ใช้งาน HT12E.....	52
รูปที่ ผ.5.7 แสดงการเลือกใช้ R_{osc} ($f_{OSCD}(\text{decoder}) = 50 f_{OSCE}(\text{HT12E encoder})$).....	53
รูปที่ ผ.6.1 โครงสร้างของ HT12D.....	53
รูปที่ ผ.6.2 Block diagram ของ HT12D.....	54
รูปที่ ผ.6.3 แสดงการประยุกต์ใช้งาน HT12D.....	54
รูปที่ ผ.6.4 แสดงการเลือกใช้ R_{osc}	55
รูปที่ ผ.7.1 หน้าจอแสดงผลของโปรแกรม Proteus.....	56

รูปที่ ผ.7.2 หน้าจอแสดงผลของโปรแกรม CCS PIC C Compiler.....	57
รูปที่ ผ.7.3 หน้าจอแสดงผลของโปรแกรม PICkit 2.....	58
รูปที่ ผ.7.4 หน้าจอแสดงผลของโปรแกรม Altium Designer 10.....	59

สารบัญตาราง

	<u>หน้า</u>
ตารางที่ 4.1.1 SW_GO (1110).....	21
ตารางที่ 4.1.2 SW_BACK (1101).....	22
ตารางที่ 4.1.3 SW_STOP (1110).....	23
ตารางที่ 4.1.4 SW_SPEED (1011).....	24-28
ตารางที่ 4.2.1 เมื่อยังไม่ได้กดสวิตช์ หรือ กด SW_RESET (0111).....	29
ตารางที่ 4.2.2 SW_LEFT (1010).....	30
ตารางที่ 4.2.3 SW_RIGHT (1001).....	31
ตารางที่ 4.2.3 ขณะสวิตช์ยังไม่ถูกกด (ไม่มี output).....	32

บทที่ 1

บทนำ

เรือบังคับวิทยุ คือ ของเล่นอิเล็กทรอนิกส์ชนิดหนึ่ง ที่สร้างความบันเทิงให้กับผู้เล่นและผู้ดู โดยเราจะออกแบบให้เรือบังคับวิทยุรับ-ส่งสัญญาณในย่านความถี่ 433.92 MHz ขนาด 1 ช่องสัญญาณ ผ่านโมดูลที่มีการมอดูเลตแบบ ASK ทั้งส่วนของภาครับและภาคส่ง และควบคุมการเดินหน้า ถอยหลัง หยุด เพิ่มความเร็ว เลี้ยวซ้าย และเลี้ยวขวา จากการเขียนโปรแกรม CCS PIC C Compiler เพื่อควบคุมการทำงานของ microcontroller

1.1 วัตถุประสงค์ของโครงการ

โครงการวิศวกรรมเรื่องเรือบังคับวิทยุ จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อสร้างและออกแบบเรือบังคับวิทยุที่สามารถนำไปใช้งานได้จริง

1.2 ขอบเขตของโครงการ

- 1.2.1 สามารถออกแบบและสร้างเรือบังคับวิทยุได้
- 1.2.2 สามารถส่งสัญญาณในย่านความถี่ 433.92 MHz ได้
- 1.2.3 สามารถเขียนโปรแกรมควบคุมการทำงานของตัวขับเคลื่อนได้
- 1.2.4 สามารถเขียนโปรแกรมควบคุมการทำงานของ PWM Servo Motor

1.3 ผลที่คาดว่าจะได้รับ

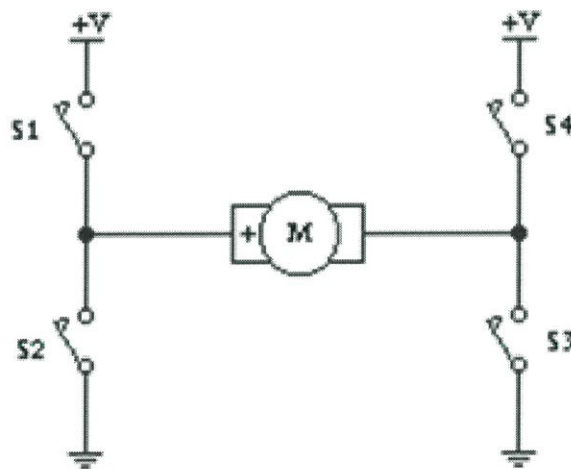
- 1.3.1 สามารถนำไปประยุกต์ใช้งานให้เกิดประโยชน์แก่สังคมได้
- 1.3.2 สามารถนำไปใช้งานได้จริง

รูปที่ 2.1 แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง

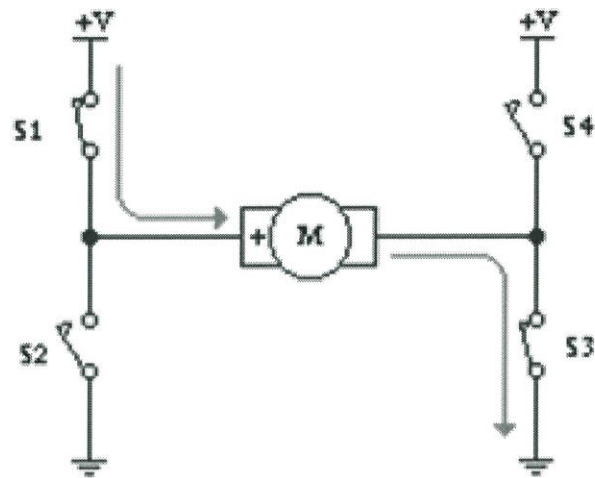
จากในรูปทางเดินของฟลักซ์แม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเฟอร์ไรต์ 2 ชั้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถังได้พอดี เพื่อที่จะให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ได้ ดังนั้น ความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กด้วย ซึ่งส่งผลให้ฟลักซ์แม่เหล็กวิ่งไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับขั้วโรเตอร์ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวร จึงเกิดเป็นแรงบิดเพื่อที่จะหมุนขั้วโรเตอร์ ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่านไปยังขั้วโรเตอร์ โดยผ่านแปรงถ่าน ซึ่งจะสัมผัสกับแหวนตัวนำในขั้วโรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวดนั่นเอง

2.1.1 หลักการทำงานของระบบมอเตอร์ไฟฟ้ากระแสตรง (DC motor)

การควบคุมทิศทางการหมุนของ DC motor ในการควบคุมทิศทางการหมุน DC motor นั้น ผู้จัดทำสามารถควบคุมได้โดยการควบคุมการจ่ายไฟให้กับ DC motor ในที่นี้จะพูดถึงการทำงานของ H-Bridge Switching เป็นวงจรที่ควบคุมการจ่ายไฟให้กับ DC motor ด้วยการ switching เพื่อเปลี่ยนทิศทางของกระแสที่ไปเลี้ยง DC motor ซึ่งจะมีผลให้ DC motor สามารถหมุนกลับทางได้ซึ่งวงจรจะประกอบไปด้วย switch 4 ตัวที่ต่อดังรูปด้านล่าง

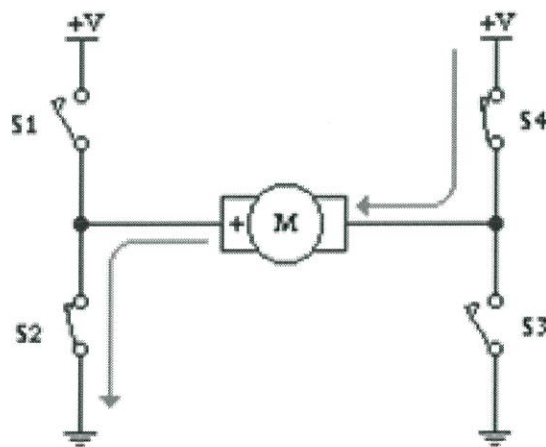


รูปที่ 2.2 แสดงรูปวงจรของ H-Bridge Switching



รูปที่ 2.3 การไต่รฟมอเตอร์ในทิศตามเข็มนาฬิกา

จากรูปที่ 2.3 จะเห็นว่ายังไม่มีกระแสไฟฟ้ให้กั DC motor จะยังไม่มีกระแสหมุน แต่เมื่อมีการจ่ายไฟดังรูปด้านล่าง DC motor จะหมุนในลักษณะตามเข็มนาฬิกา ดังรูปที่ 2.6



รูปที่ 2.4 การไต่รฟมอเตอร์ในทิศทวนเข็มนาฬิกา

แต่เมื่อใดก็ตามที่มีการ switch ดังรูปที่ 2.4 ด้านล่างจะเกิดการป้อนกระแสกลับทางให้กั DC motor ทำให้ DC motor หมุนในทิศทวนเข็มนาฬิกา

2.1.2 การควบคุมความเร็วของมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

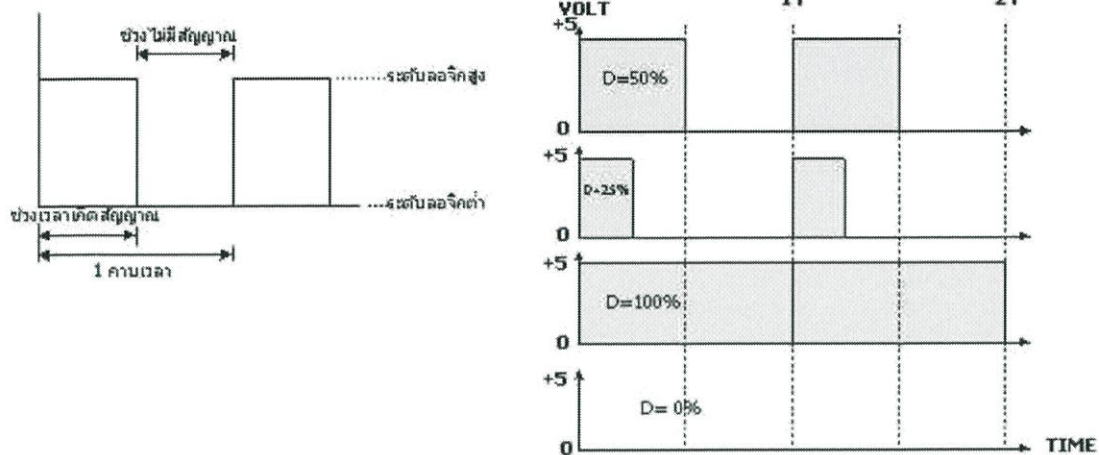
2.1.3 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM)

การทำให้มอเตอร์ไฟฟ้ากระแสตรงหมุนสามารถทำได้โดยจ่ายไฟฟ้ากระแสตรงให้กับมอเตอร์ซึ่งจะทำให้มอเตอร์หมุนที่ความเร็วสูงสุดภายใต้สภาวะที่มอเตอร์รับภาระอยู่ในขณะนั้น ถ้าต้องการให้มอเตอร์หมุนกลับทิศทางก็ทำได้โดยการกลับขั้วของแหล่งจ่ายกระแสไฟฟ้าที่จ่ายให้กับมอเตอร์

ในกรณีที่ต้องการควบคุมความเร็วรอบของมอเตอร์สามารถทำได้โดยการเพิ่มหรือลดขนาดของแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ Pulse Width Modulator (PWM) เป็นวิธีที่นิยมมากในการเพิ่มหรือลดขนาดของแรงดันไฟฟ้าที่ใช้ในการควบคุมความเร็วของมอเตอร์โดยที่ขนาดของแรงดันไฟฟ้าเฉลี่ยจะเปลี่ยนไปตามความกว้างของพัลส์

การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนที่สัดส่วน และความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มี การเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดิวตี้ไซเคิล (duty cycle) นั้นเอง ซึ่งค่าของดิวตี้ไซเคิล คือ ช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดิวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึงใน 1 วัฏสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูป 6.27 และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดิวตี้ไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ดังนี้

$$\text{ค่าดิวตี้ไซเคิล} = \left(\frac{\text{ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ}}{\text{สัญญาณ}} \right) \times 100\%$$

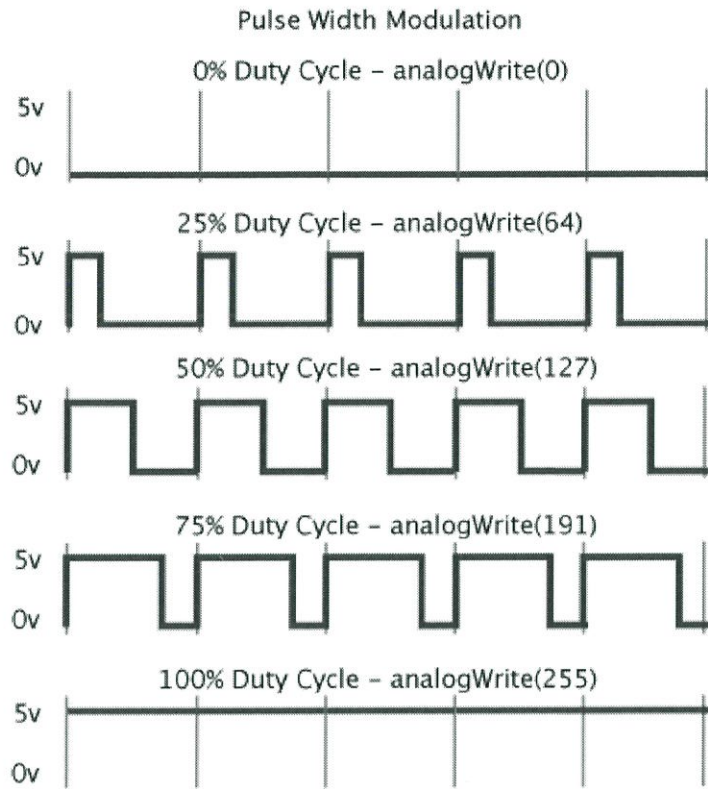


รูปที่ 2.5 แสดงความกว้างของพัลส์ขนาดต่างๆ และค่าดิวตีไซเคิล ของช่วงพัลส์ที่มีความถี่คงที่

2.2 การควบคุมแบบ PWM (Pulse Width Modulation)

Pulse width modulation (PWM) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม

2.2.1 การทำงานของสัญญาณ PWM



รูปที่ 2.6 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่าง ๆ กัน

จากรูปที่ 2.6 แสดงสัญญาณ PWM ที่แตกต่างกัน 3 สัญญาณ โดย 2.9 (a) แสดงสัญญาณ PWM ที่ 10% duty cycle คือ สัญญาณช่วง high จะเป็น 10% ของคาบสัญญาณ และ สัญญาณช่วง low เป็น 90% ของคาบสัญญาณ ซึ่งจะได้ค่าแรงดันเฉลี่ยของแรงดันเป็น 10% ของแรงดันสูงสุดโดย 2.9 (b) แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณช่วง high จะเป็น 50% ของคาบสัญญาณ และ สัญญาณช่วง low เป็น 50% ของคาบสัญญาณ ซึ่งจะได้ค่าแรงดันเฉลี่ยของแรงดันเป็น 50% ของแรงดันสูงสุดโดย 2.9 (c) แสดงสัญญาณ PWM ที่ 90% duty cycle คือ สัญญาณช่วง high จะเป็น 90% ของคาบสัญญาณ และ สัญญาณช่วง low เป็น 10% ของคาบสัญญาณ ซึ่งจะได้ค่าแรงดันเฉลี่ยของแรงดันเป็น 90% ของแรงดันสูงสุดยกตัวอย่างเช่น ถ้า Power Supply มีแรงดัน 9V และใช้ duty cycle เป็น 10% จะได้แรงดันเอาต์พุต 0.9V

2.2.2 ประโยชน์ของ PWM ในการควบคุมความเร็วมอเตอร์

PWM สามารถนำไปประยุกต์ใช้ในการควบคุมความเร็วของมอเตอร์ได้ดังนี้

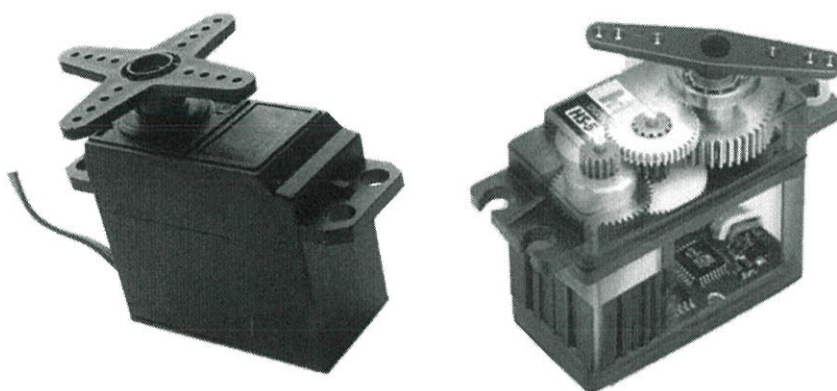
- PWM ง่ายในการอินเทอร์เฟสกับไมโครคอนโทรลเลอร์ และ ใช้เพียงแค่เอาต์พุตสัญญาณเดียวในการควบคุมความเร็ว
- PWM มีประสิทธิภาพ คือ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF

(FULL ON and FULL OFF)

- PWM สามารถทำให้ความเร็วของมอเตอร์เปลี่ยนในขณะที่แรงบิดยังเท่าเดิมได้ เพราะ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF (FULL ON and FULL OFF)

2.3 Servo Motor

Servo motor คือมอเตอร์ที่นำมาใช้ในระบบการควบคุมทางความเร็วและตำแหน่งร่วมกัน โดยระบบservoที่ดีจะต้องตอบสนองต่อความเร็วและการเข้าถึงตำแหน่งหรือการเคลื่อนที่ไปที่ระยะเป้าหมายอย่างเหมาะสม ระบบservo จะมีการป้อนกลับของความเร็วและตำแหน่งกลับมาที่ส่วนควบคุมหรือdrive ขึ้นอยู่กับการออกแบบ



รูปที่ 2.7 เซอร์โวมอเตอร์และองค์ประกอบของเซอร์โวมอเตอร์

2.3.1 ความแตกต่างของเซอร์โวมอเตอร์กับมอเตอร์ชนิดอื่นๆ

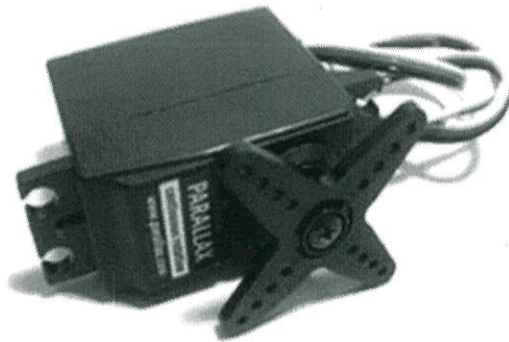
เซอร์โวมอเตอร์ได้ถูกพัฒนาขึ้นเพื่อประยุกต์ใช้กับงานที่มีการควบคุมตำแหน่ง (Position Control) เป็นหลัก ซึ่งต้องการคุณสมบัติการตอบสนองด้านไดนามิกส์สูง (high dynamics response) ต้องการความละเอียดและความแม่นยำสูงในการควบคุมสูง

ในทางปฏิบัติเพื่อให้ได้มาซึ่งคุณสมบัติดังกล่าว ผู้ผลิตจึงได้พยายามออกแบบให้โรเตอร์มีเส้นผ่าศูนย์กลางและรัศมีเล็กๆ รวมถึงการทำให้โรเตอร์เป็นแม่เหล็กถาวร เพื่อเป็นการลดแรงเฉื่อยของโรเตอร์ (Rotor moment of inertia) และชดเชยแรงบิดหรือทอร์กที่หายไปเนื่องจากโรเตอร์มีขนาดเล็กลง ด้วยการเพิ่มความยาวโรเตอร์ ดังนั้นรูปร่างโครงสร้างทางกายของเซอร์โวมอเตอร์ ที่เราพบเห็นทั่วไปจึงมีลักษณะผอมบางและมีความยาวมากกว่ามอเตอร์ที่ใช้งานทั่วไป นอกจากนั้นยังมีแรงบิดสูงกว่ามอเตอร์ทั่วไปอีกด้วย (หากพิจารณาที่พิกัดกำลังเท่าๆกัน) ซึ่งสามารถเปรียบเทียบได้คล้ายกับคนที่รูปร่าง “สูงผอมบาง แต่แรงดี” เป็นต้น

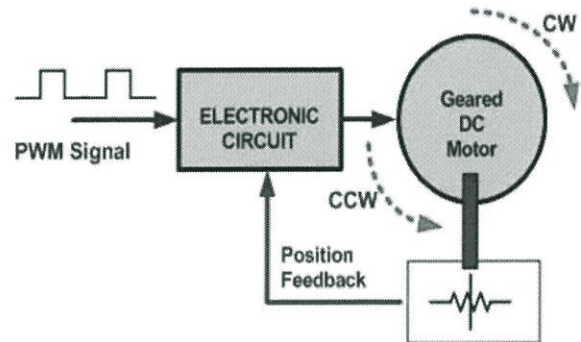
2.3.2 การควบคุมทำงานของเซอร์โวมอเตอร์ ด้วย PWM

การต่อใช้งานเซอร์โวมอเตอร์ ในการควบคุมเซอร์โวมอเตอร์นั้นทำได้โดยอาศัยความกว้างของพัลส์ที่ทำการป้อนให้เซอร์โวมอเตอร์ โดยสัญญาณพัลส์นี้จะเป็นสัญญาณ TTL จะมีแรงดัน 5VDC และ แรงดัน 0 VDC

<http://www.ermicro.com/blog>



Typical Servo Motor



Servo Motor Block Diagram

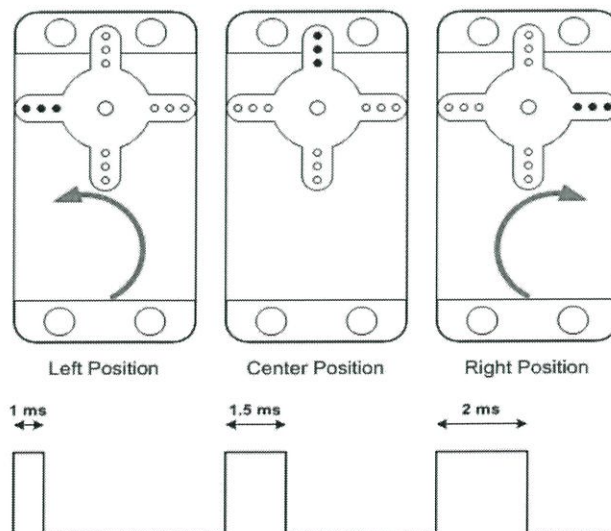
The Servo Motor

รูปที่ 2.8 การควบคุมการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์หมุนไปทางขวา หรือ ตามเข็มนาฬิกา จะต้องสร้างสัญญาณพัลส์ ความกว้างขนาด 1 ms และมีช่วงระยะห่างระหว่างพัลส์ 20 ms

เซอร์โวมอเตอร์หมุนไปทางซ้าย หรือ ทวนเข็มนาฬิกา จะต้องสร้างสัญญาณพัลส์ ขนาด 2ms และมีช่วงระยะห่างระหว่างพัลส์ 20

เซอร์โวมอเตอร์อยู่ในตำแหน่งกึ่งกลาง จะต้องสร้างสัญญาณพัลส์ ขนาด 1.5 ms และมีช่วงระยะห่างระหว่างพัลส์ 20



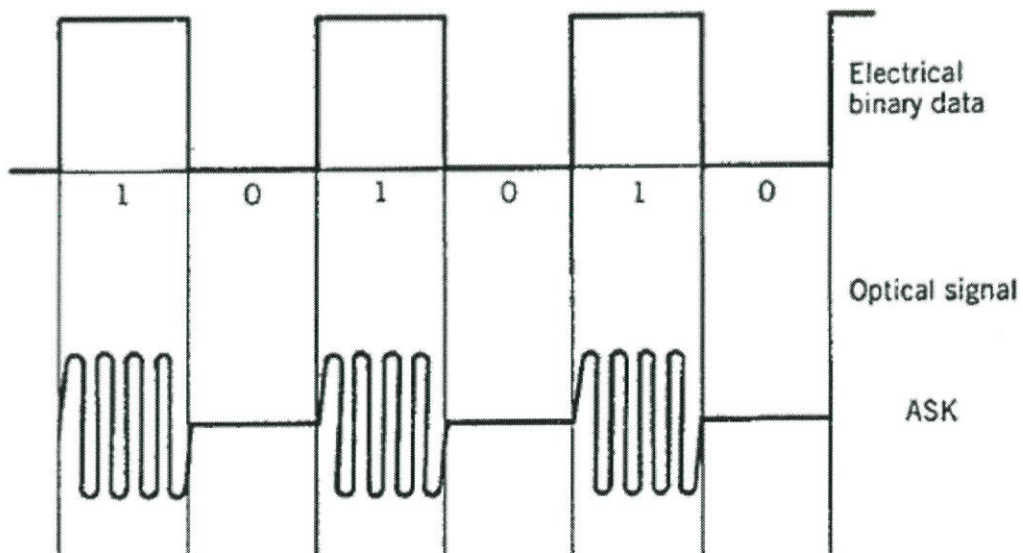
รูปที่ 2.9 PWM Servo Motor

2.4 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog, D/A)

ในปัจจุบันการส่งสัญญาณข้อมูลดิจิทัลโดยผ่านช่องทางการสื่อสารแบบอนาล็อกที่เราคุ้นเคยกันได้แก่ การส่งข้อมูลคอมพิวเตอร์ผ่านทางเครือข่ายโทรศัพท์สาธารณะ เครือข่ายโทรศัพท์ถูกออกแบบมาเพื่อทำให้การสลับสวิทช์และส่งสัญญาณอนาล็อกซึ่งเป็นย่านความถี่เสียง หรือประมาณ 300-3,400 Hz อุปกรณ์ที่ทำหน้าที่แปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อกย่านความถี่เสียงเราเรียกว่า โมเด็ม (MODulator – DEModulator : MODEM) ในตอนนี้เราจะกล่าวเฉพาะการมอดูเลตแบบ ASK เท่านั้น

2.4.1 การมอดูเลตเชิงเลขทางแอมพลิจูด (Amplitude – Shift Keying : ASK)

การมอดูเลตวิธีนี้ ความถี่ของคลื่นพาหะ (Carrier Wave) ซึ่งทำหน้าที่นำสัญญาณแบบอนาล็อกผ่านตัวกลางสื่อสารนั้นจะคงที่ ลักษณะของสัญญาณที่มอดูเลตนั้น เมื่อค่าของบิตของสัญญาณข้อมูลดิจิทัลมีค่าเป็น “1” ขนาดของคลื่นพาหะจะสูงกว่าตอนปกติ และเมื่อบิตมีค่าเป็น “0” ขนาดของคลื่นพาหะจะตกลงกว่าปกติ การมอดูเลตแบบ ASK มักจะไม่ค่อยได้รับความนิยม เพราะจะถูกรบกวนจากสัญญาณอื่นได้ง่าย

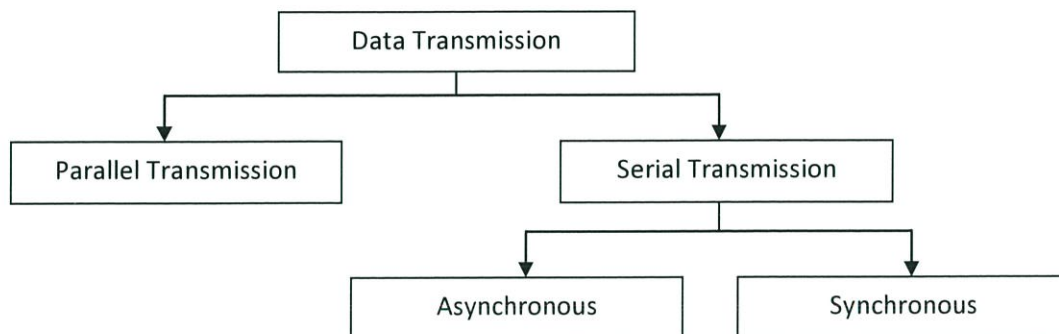


รูปที่ 2.10 การแปลงสัญญาณดิจิทัลเป็นอนาล็อกแบบ ASK

2.5 การส่งสัญญาณข้อมูล

การส่งสัญญาณข้อมูลแบ่งออกได้เป็น 2 ประเภท คือ

1. การส่งแบบอนุกรม (Serial Transmission)
2. การส่งแบบขนาน (Parallel Transmission)



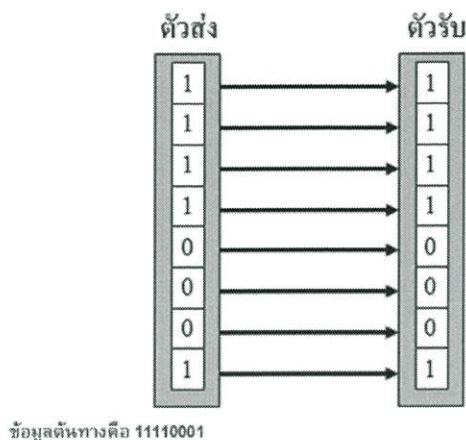
รูปที่ 2.11 Block diagram ของ Data Transmission

2.5.1 การส่งข้อมูลแบบขนาน (Parallel Transmission)

การส่งข้อมูลแบบขนานนั้นจะทำการส่งข้อมูลที่หลายๆบิต เช่น ส่ง 11110001 ทั้ง 8 บิต ออกไปพร้อมกันโดยผ่านสายส่งข้อมูลที่มี 8 เส้น ตัวอย่างการใช้งานที่เห็นชัดของการส่งข้อมูลแบบขนาน เช่น การต่อเครื่องพิมพ์เข้ากับเครื่องคอมพิวเตอร์ ซึ่งปกติจะใช้สายยาว 5 เมตร ถึง 10 เมตรเท่านั้น

ข้อดี คือสามารถส่งข้อมูลได้รวดเร็ว เพราะส่งครั้งละ 8 บิต

ข้อเสีย คือใช้ส่งแต่เฉพาะใกล้ๆเท่านั้น ราคาแพง



ข้อมูลต้นทางคือ 11110001

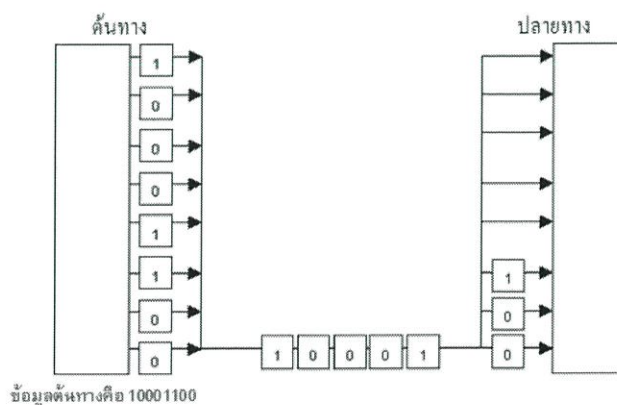
รูปที่ 2.12 การส่งข้อมูลแบบขนาน (Parallel Transmission)

2.5.2 การส่งข้อมูลแบบอนุกรม (Serial Transmission)

การส่งข้อมูลแบบอนุกรม ข้อมูลจะถูกส่งออกไปทีละบิต ต่อเนื่องกันไป เช่น ถ้าข้อมูลคือ 10001100 เลข 0 ทางขวามือสุดเป็นบิตที่ 1 เรียงลำดับไปจนครบ 8 บิต โดยการส่งนั้นจะใช้สายส่งเส้นเดียวเท่านั้น ตัวอย่างการส่งข้อมูลแบบอนุกรม เช่น การต่อเทอร์มินัลเข้ากับคอมพิวเตอร์แม่ที่อยู่ห่างกันสัก 100 เมตร ซึ่งทำให้ประหยัดสาย

ข้อดี คือ สามารถส่งได้ระยะทางไกลมากกว่า Parallel Transmission

ข้อเสีย คือ ความเร็วในการรับส่งข้อมูลมีจำกัด ต้องคำนึงถึงรายละเอียดในการรับส่งข้อมูล



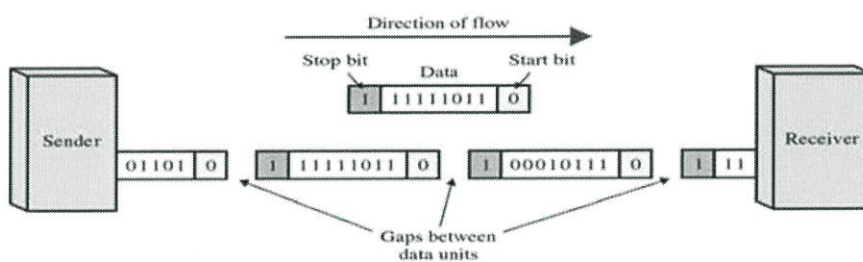
ข้อมูลต้นทางคือ 10001100

รูปที่ 2.13 การส่งข้อมูลแบบอนุกรม (Serial Transmission)

การส่งข้อมูลแบบอนุกรมแบ่งออกได้เป็น 2 ประเภทได้แก่

1. การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Data Transmission)

- มักจะใช้กับเทอร์มินัลธรรมดา (dumb terminal) ไว้สำหรับรับข้อมูลจากคอมพิวเตอร์แม่และแสดงผลที่จอ โดยไม่สามารถเปลี่ยนแปลงข้อมูลได้
- การส่งข้อมูลแบบนี้มักจะมีอัตราในการรับส่งข้อมูลที่แน่นอนมีหน่วยเป็นบิตต่อวินาที (bit per second)
- เมื่ออุปกรณ์อะซิงโครนัสจะส่งข้อมูล 1 ไบต์ ก็จะส่งบิตเริ่มต้น (start bit) ก่อน ซึ่งมักจะเป็น “0” และตามด้วยข้อมูลทั้ง 8 บิตใน 1 ไบต์ แล้วจึงจะส่งบิตหยุด (stop bit) ซึ่งมักจะเป็น “1” บิตทั้งหมดนี้ จะรวมกันเป็น 10 บิต ในการส่งข้อมูลเรียงตามลำดับดังนี้ 1 บิต เริ่มต้น 8 บิตข้อมูล (data bit) และ 1 บิตหยุด
- กระบวนการเหล่านี้ จะห่างกัน 1 วินาที ที่จะส่งข้อมูลชุดต่อไป ซึ่งก็หมายความว่าเมื่อคอมพิวเตอร์แม่ได้รับบิตเริ่มต้น ก็คาดหวังว่าจะได้รับอีก 9 บิตภายในเวลา 1 วินาที
- ในระบบนี้จะเกี่ยวข้องกับเวลาที่เมื่อไรบิตต่อไปจะมาถึง ถ้าไม่ตรงตามที่กำหนดไว้ การส่งข้อมูลก็จะล้มเหลว

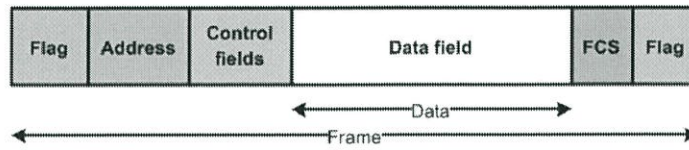


รูปที่ 2.14 การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous data transmission)

2. การส่งข้อมูลแบบซิงโครนัส (Synchronous Data Transmission)

จะไม่ใช้บิตเริ่มต้นและบิตหยุด จะไม่มีการหยุดชั่วคราวระหว่างอักขระ จะใช้วิธีให้จังหวะเวลาทั้งสองทางที่ติดต่อกัน มีอยู่ 2 วิธีคือ 1. ใช้บิตซิงค์ (sync bit) 2. ใช้อักขระซิงค์ (sync character)

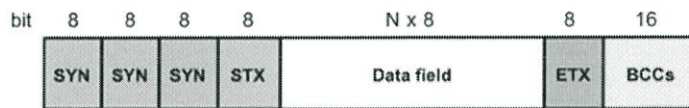
■ การใช้บิตซิงค์ (sync bit)



- **Flag** : ใช้บ่งบอกจุดเริ่มและจุดสิ้นสุดของ frame
- **Control** : ใช้ควบคุมการส่งข้อมูล
- **Address** : ของฝั่งผู้ส่งและผู้รับ
- **FCS** : ใช้ตรวจสอบความถูกต้องของ Frame ข้อมูล

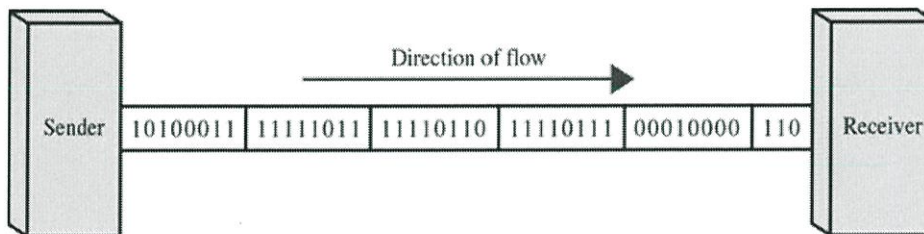
รูปที่ 2.15 การใช้บิตซิงค์ (sync bit)

■ การใช้อักขระซิงค์ (Syn Characters)



- SYN** = Synchronous
- ETX** = End of Text
- STX** = Start of Text
- BCCs** = Block Check Characters

รูปที่ 2.16 การใช้อักขระซิงค์ (sync characters)



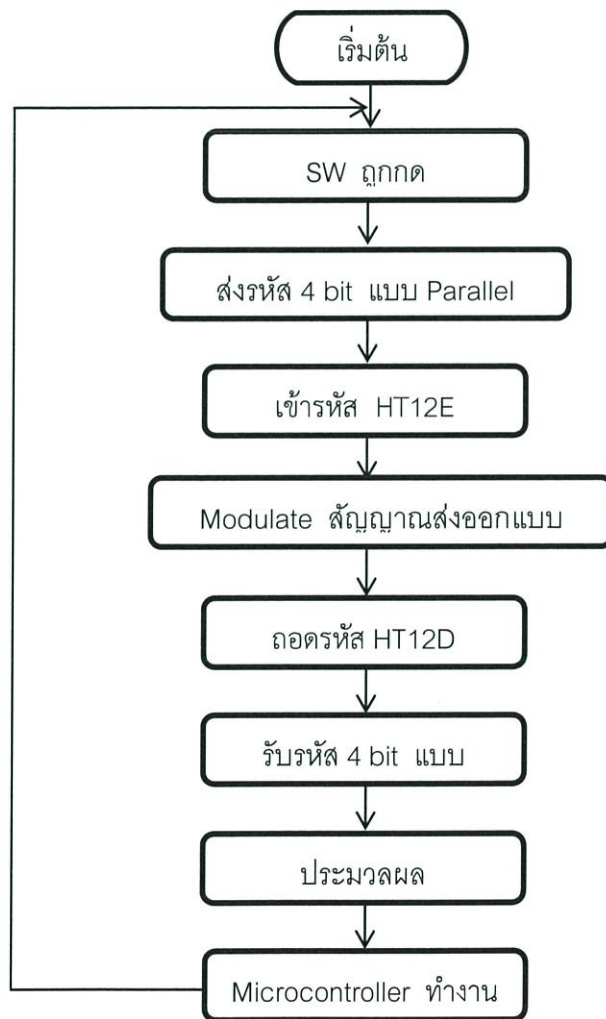
รูปที่ 2.17 การส่งข้อมูลแบบซิงโครนัส (Synchronous data transmission)

บทที่ 3

ขั้นตอนการดำเนินงานและการออกแบบ

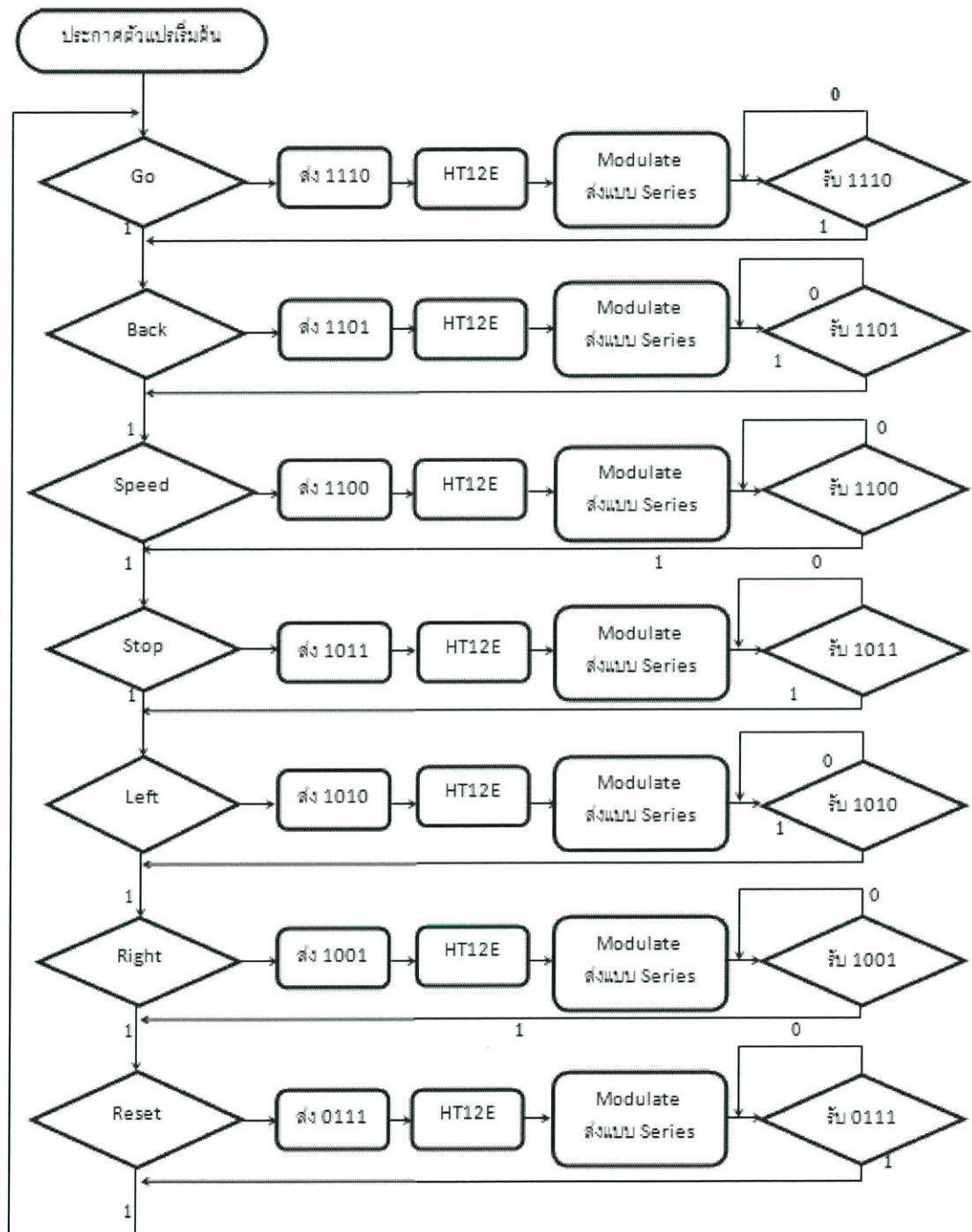
3.1 Flow Chart

Flow Chart การทำงานของเรือบังคับวิทยุ



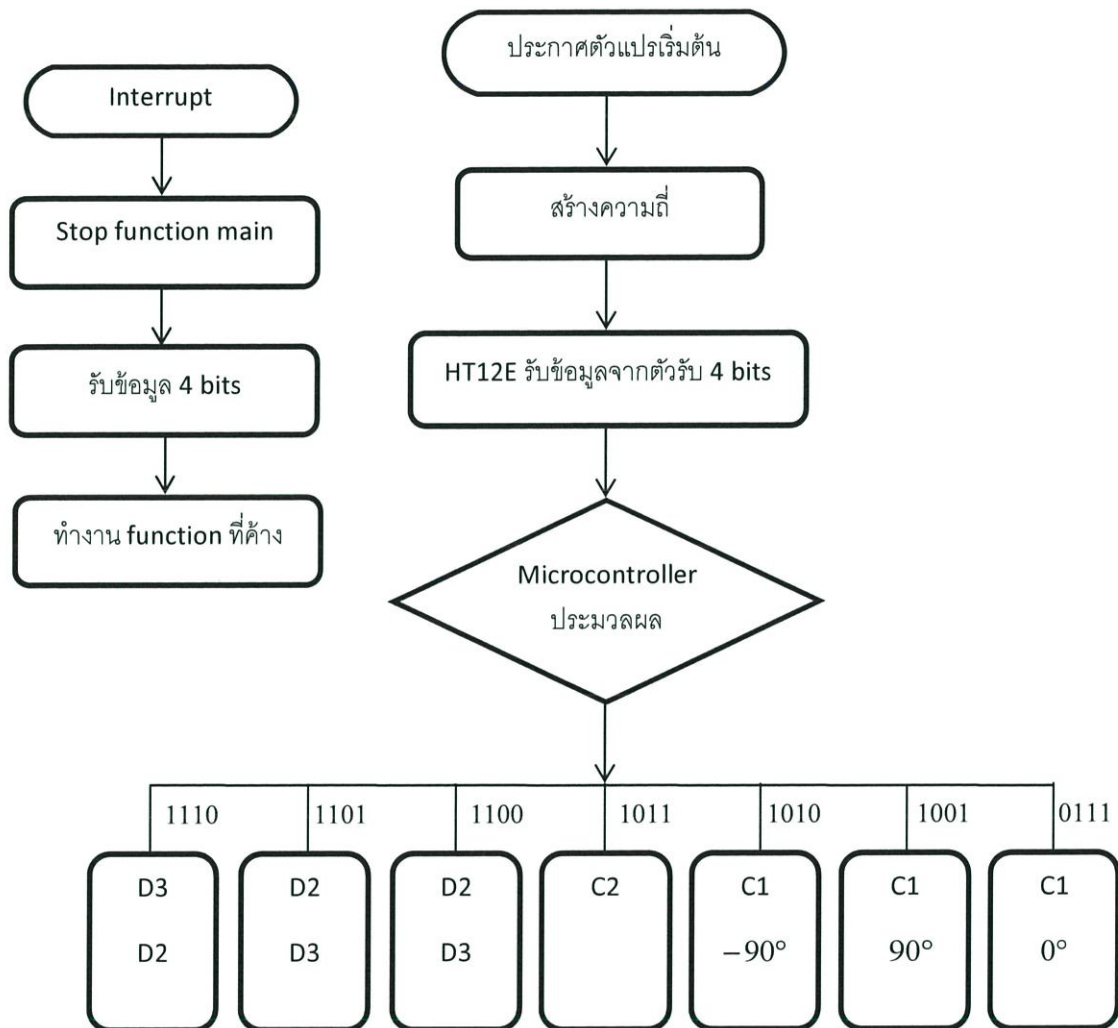
รูปที่ 3.1 Flow Chart การทำงานของเรือบังคับวิทยุ

Flow Chart การทำงานในส่วนของภาคส่ง



รูปที่ 3.2 Flow Chart การทำงานในส่วนของภาคส่ง

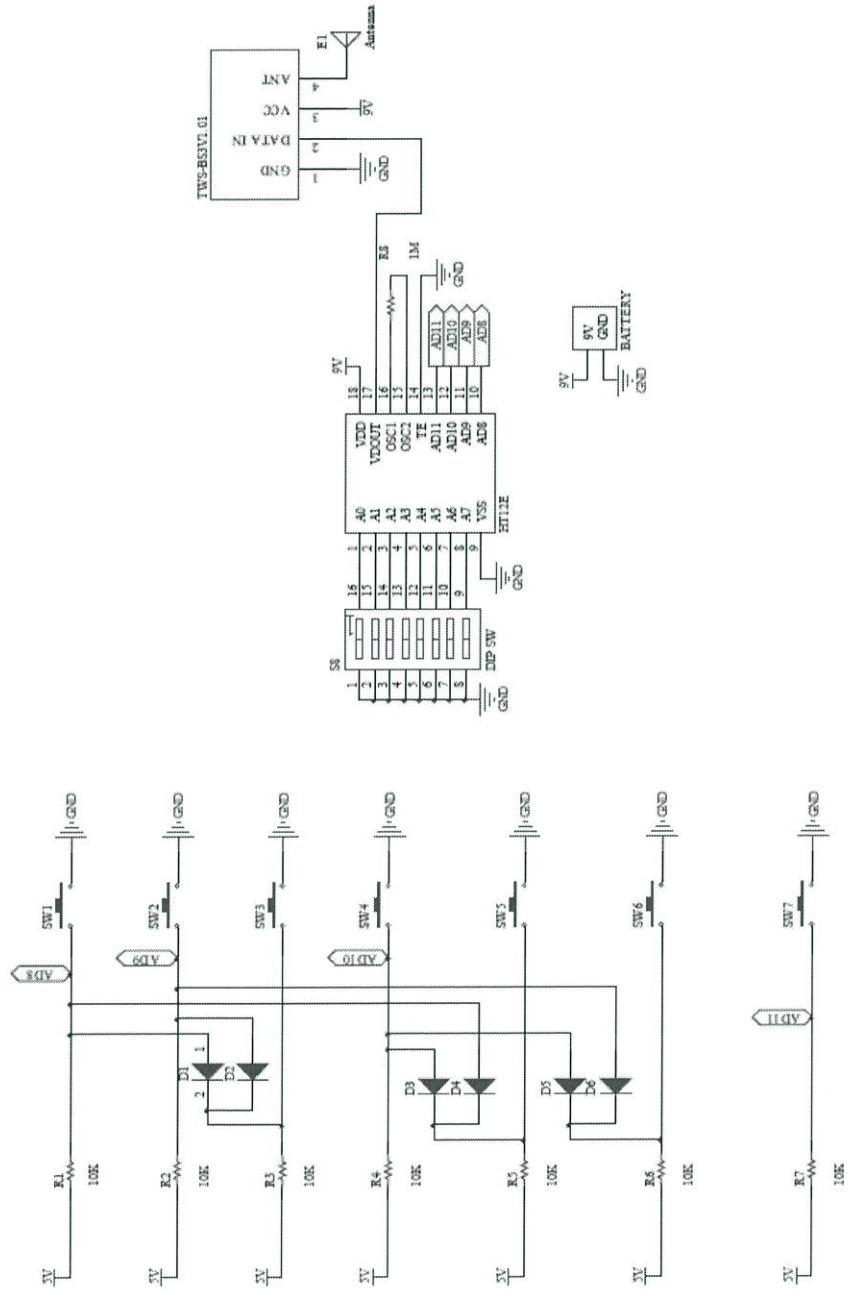
Flow Chart การทำงานในส่วนของภาครับ



รูปที่ 3.3 Flow Chart การทำงานในส่วนของภาครับ

3.2 ศึกษาค้นคว้าและออกแบบในส่วนของวงจร

3.2.1 วงจรภาคส่ง



รูปที่ 3.4 วงจรภาคส่ง

3.3 ค่าพารามิเตอร์ที่ใช้ในการเขียนโปรแกรม

ค่าพารามิเตอร์ที่ใช้ในการเขียนโปรแกรมตามสูตรคำนวณต่อไปนี้

3.3.1 ค่าความถี่

$$f = \frac{1}{T} \quad (3.1)$$

3.3.2 ค่าความถี่ %Duty Cycle

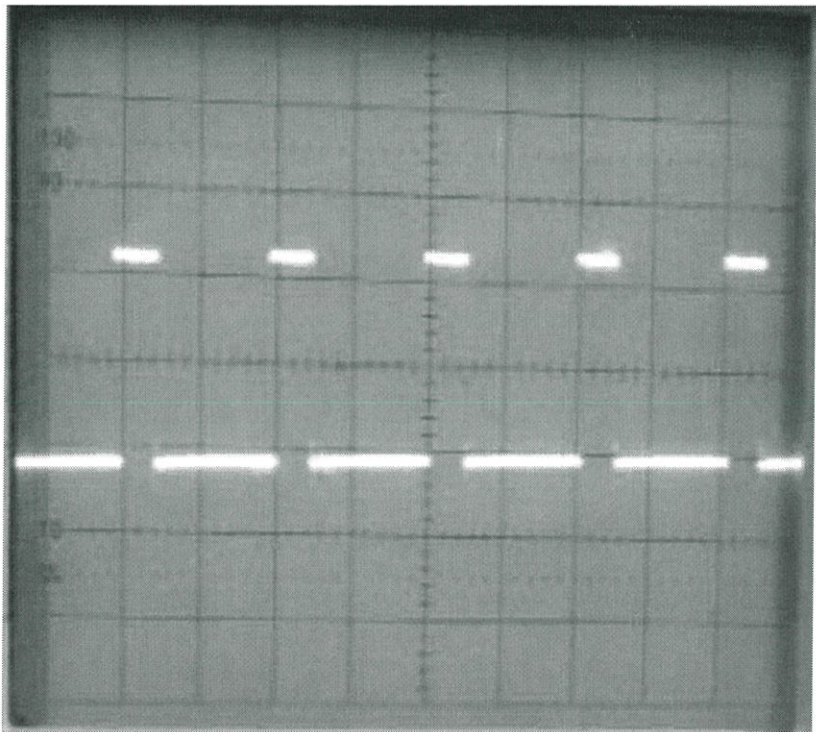
$$\%D = \frac{t_{on}}{T} \times 100\% \quad (3.2)$$

บทที่ 4

ผลการทดลอง

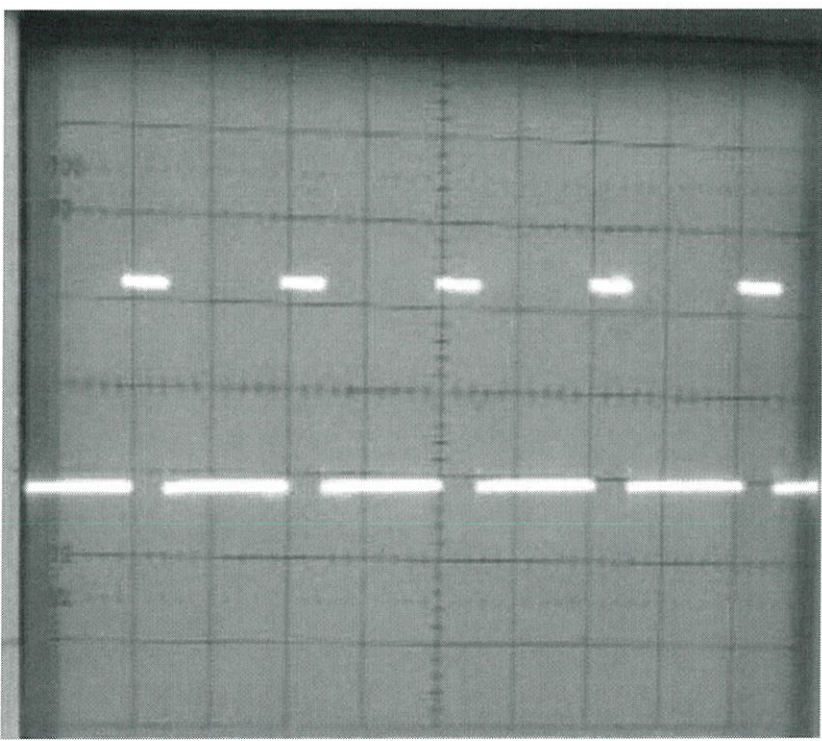
4.1 ผลการทดสอบวงจรขับเคลื่อนมอเตอร์จาก Microcontroller

ตารางที่ 4.1.1 SW_GO

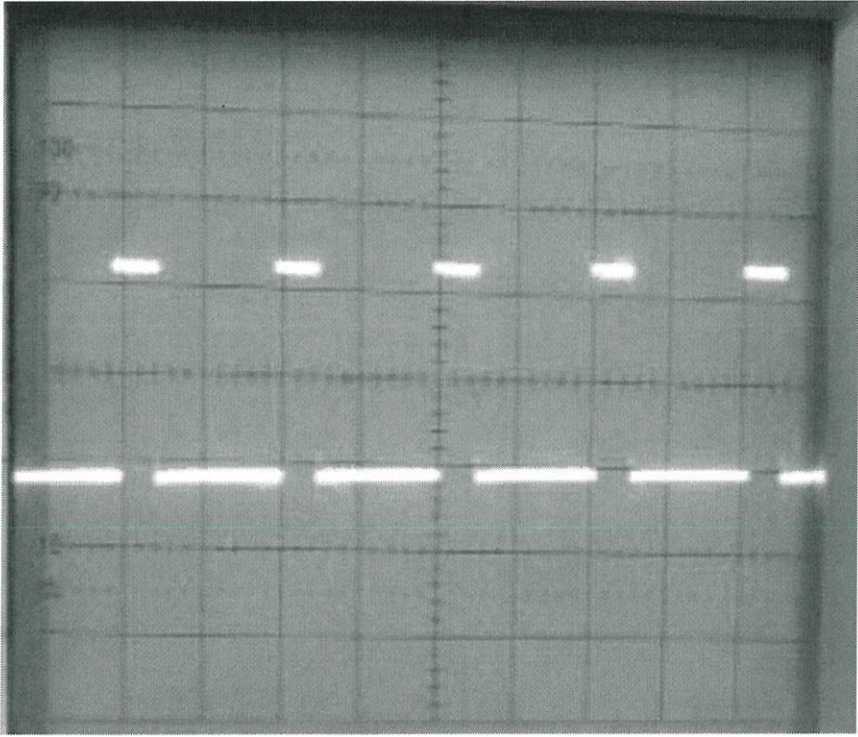
input	PIN_B4	High
	PIN_B5	High
	PIN_B6	High
	PIN_B7	Low
output	PIN_D3	High
	PIN_D2	Low
	C2 Volt/DIV = 2V Time/DIV = 1 ms f = 500 Hz	
Duty cycle	25%	

รูปที่ 4.1 การเกิด Pulse ที่ $T_o = 500 \mu$ (คาบ/s)

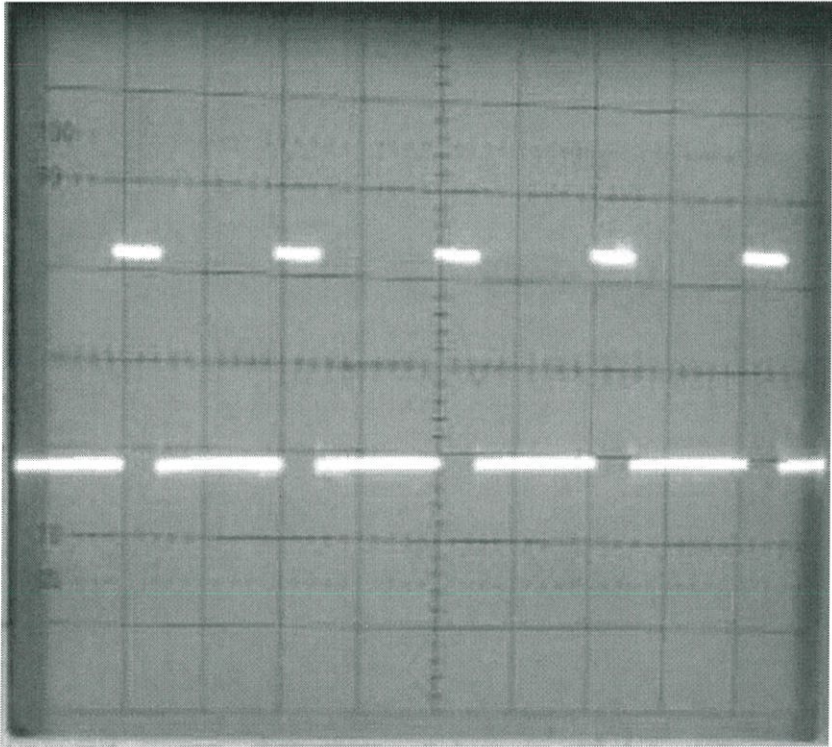
ตารางที่ 4.1.2 SW_BACK

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	High
output	PIN_D3	Low
	PIN_D2	High
	C2	 <p>รูปที่ 4.2 การเกิด Pulse ที่ $T_o = 500 \mu(\text{คาบ}/\text{s})$</p>
Duty cycle	25%	

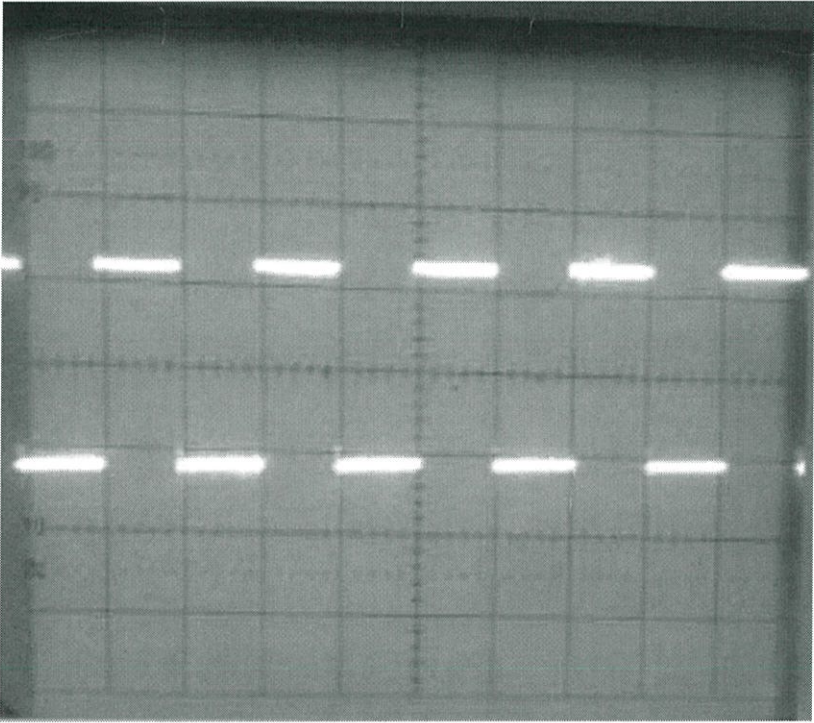
ตารางที่ 4.1.3 SW_STOP

input	PIN_B4	High
	PIN_B5	Low
	PIN_B6	High
	PIN_B7	High
output	PIN_D3	Low
	PIN_D2	Low
	C2	 <p>รูปที่ 4.3 การเกิด Pulse ที่ $T_o = 500 \mu(\text{คาบ/s})$</p>
Duty cycle	25%	

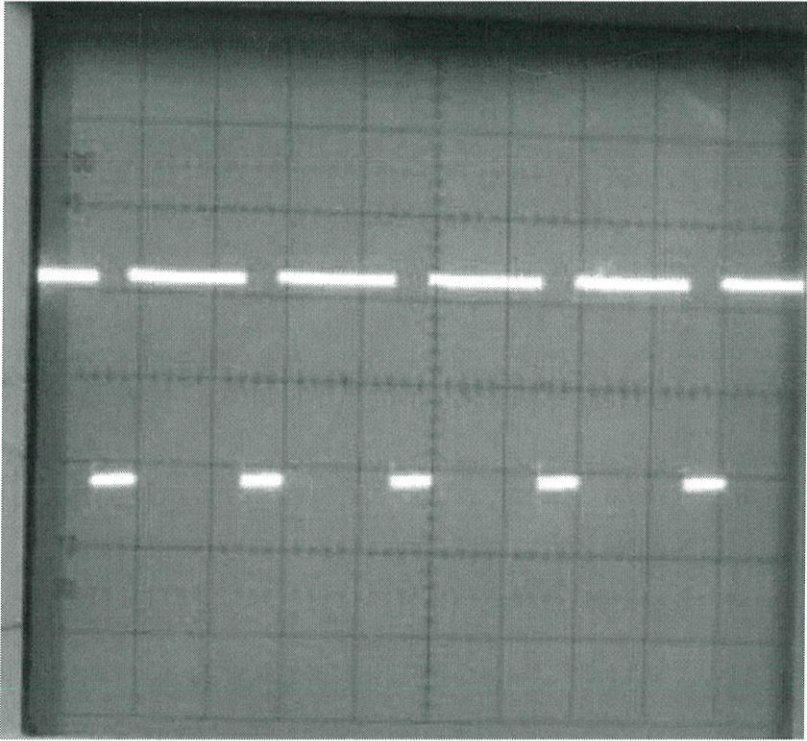
ตารางที่ 4.1.4 SW_SPEED

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	Low
output	J	0
	C2	 <p>รูปที่ 4.4 การเกิด Pulse ที่ $T_o = 500 \mu(\text{คาบ/s})$</p>
	Duty cycle	25%
DC Motor	กำลังงาน (P)	$P = IV = 0.6 \times 4 = 2.4 \text{ W}$

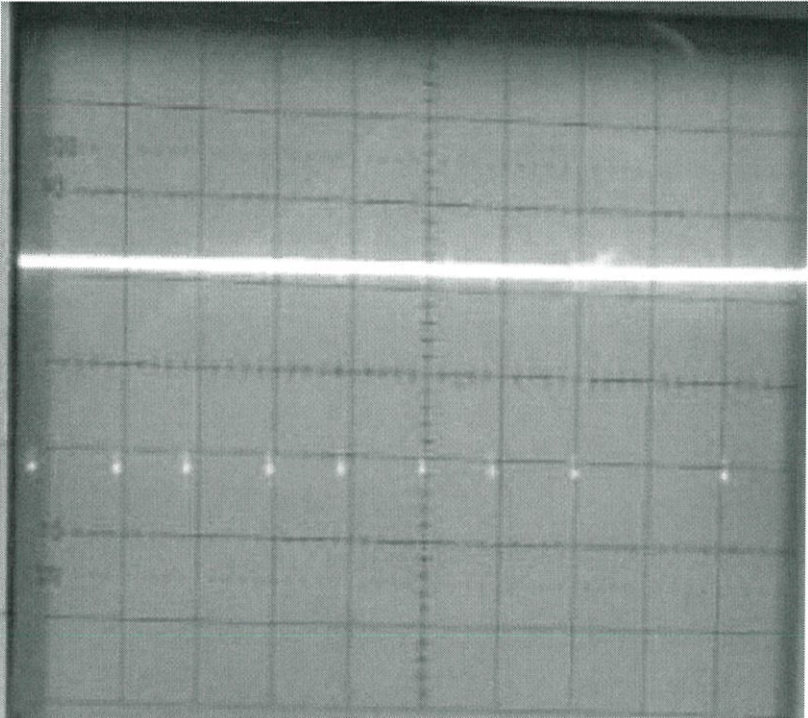
เมื่อทดลองกดสวิทช์ครั้งที่หนึ่ง

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	Low
output	J	1
	C2	 <p>รูปที่ 4.5 การเกิด Pulse ที่ $T_o = 1000 \mu(\text{คาบ/s})$</p>
	Duty cycle	25%
DC Motor	กำลังงาน (P)	$P = IV = 0.7 \times 4.2 = 2.94 \text{ W}$

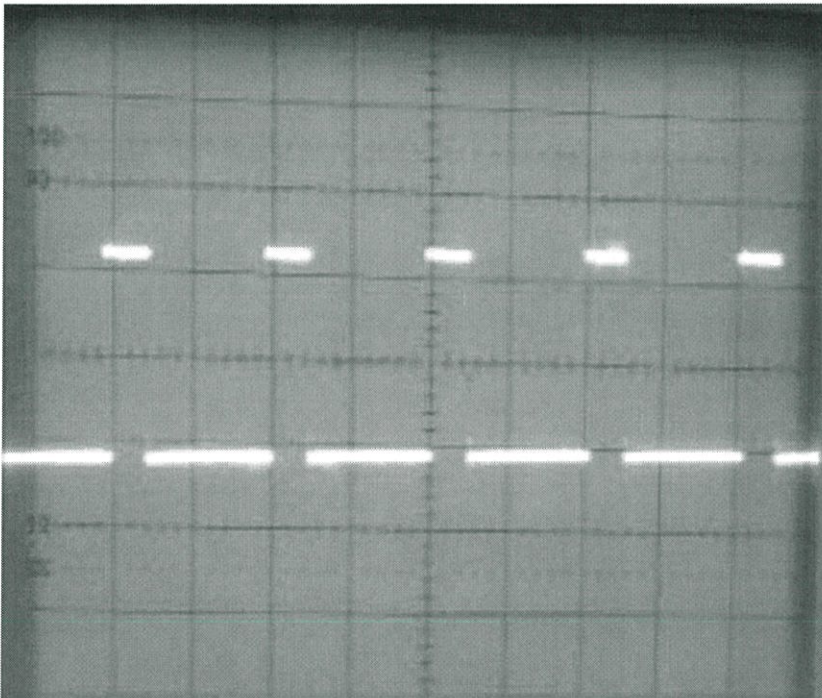
เมื่อทดลองกดสวิทช์ครั้งที่สอง

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	Low
output	J	2
	C2	 <p>รูปที่ 4.6 การเกิด Pulse ที่ $T_o = 1500 \mu(\text{คาบ/s})$</p>
	Volt/DIV = 2V Time/DIV = 1 ms f = 500 Hz	
Duty cycle	25%	
DC Motor	กำลังงาน (P)	$P = IV = 0.8 \times 4.35 = 3.48 \text{ W}$

เมื่อทดลองกดสวิทช์ครั้งที่สาม

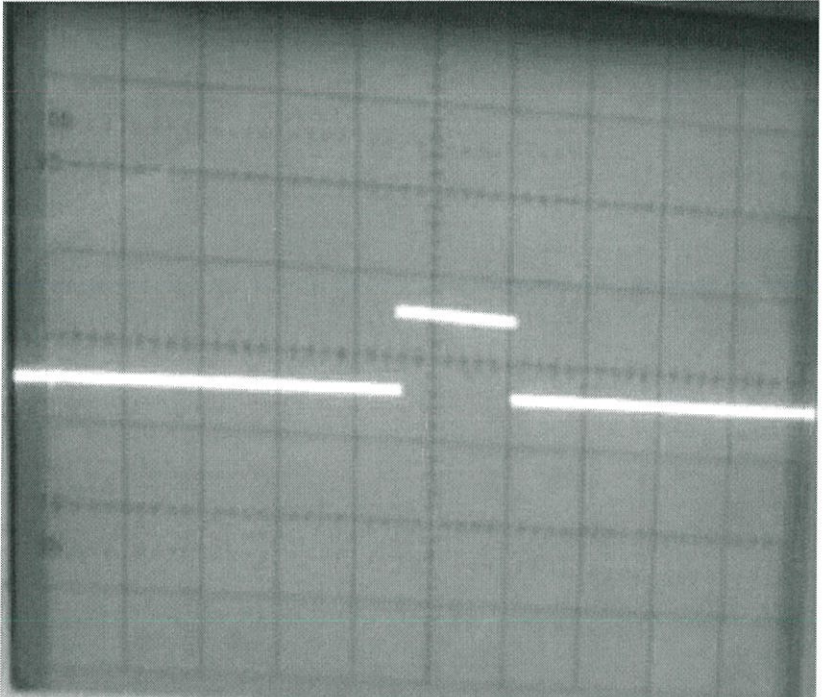
input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	Low
output	J	3
	C2 Volt/DIV = 2V Time/DIV = 1 ms f = 500 Hz	 <p>รูปที่ 4.7 การเกิด Pulse ที่ $T_o = 2000$ น(คาบ/s)</p>
	Duty cycle	25%
DC Motor	กำลังงาน (P)	$P = IV = 0.9 \times 4.48 = 4.03 \text{ W}$

เมื่อทดลองกดสวิตช์มากกว่าสามครั้ง

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	Low
	PIN_B7	Low
output	J	>3
	C2	 <p>Volt/DIV = 2V Time/DIV = 1 ms f = 500 Hz</p> <p>รูปที่ 4.8 การเกิด Pulse ที่ $T_o = 500$ μ(คาบ/ส)</p>
	Duty cycle	25%
DC Motor	กำลังงาน (P)	$P = IV = 0.6 \times 4 = 2.4 \text{ W}$

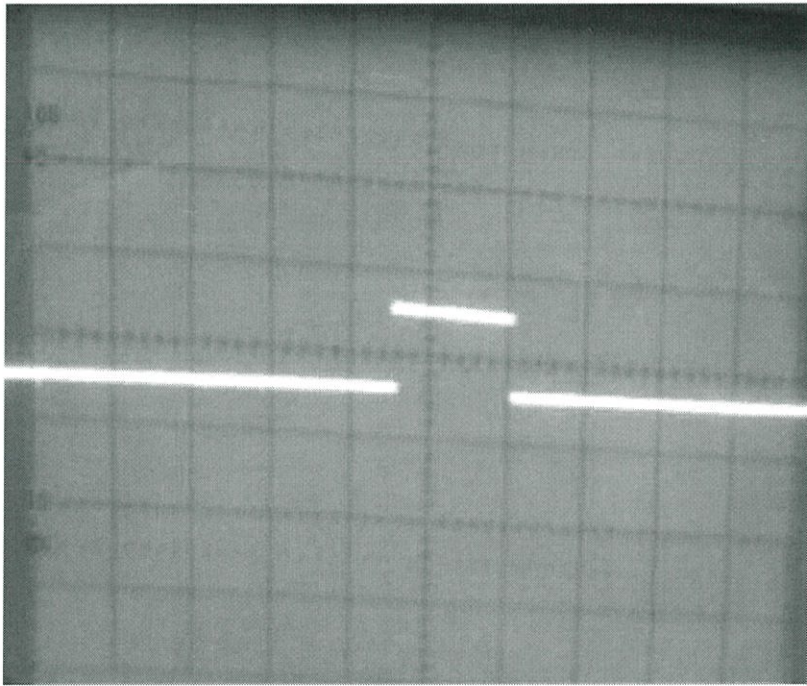
4.2 ผลการทดสอบ PWM Servo Motor

ตารางที่ 4.2.1 เมื่อยังไม่ได้กดสวิตช์ "0°"

input	PIN_B4	Low
	PIN_B5	High
	PIN_B6	High
	PIN_B7	High
output	C1 Volt/DIV = 5V Time/DIV = 0.5 ms	
	Duty cycle	25%


รูปที่ 4.9 การเกิด Pulse ที่ "0°"

ตารางที่ 4.2.2 SW_LEFT = "90°"

input	PIN_B4	High
	PIN_B5	Low
	PIN_B6	High
	PIN_B7	Low
output	C1 Volt/DIV = 5V Time/DIV = 0.5 ms	
	Duty cycle	25%

รูปที่ 4.10 การเกิด Pulse ที่ "90°"

ตารางที่ 4.2.3 SW_RIGHT = " - 90°"

input	PIN_B4	High
	PIN_B5	Low
	PIN_B6	Low
	PIN_B7	High
output	C1 Volt/DIV = 5V Time/DIV = 0.5 ms	
	Duty cycle	25%

รูปที่ 4.11 การเกิด Pulse ที่ " - 90°"

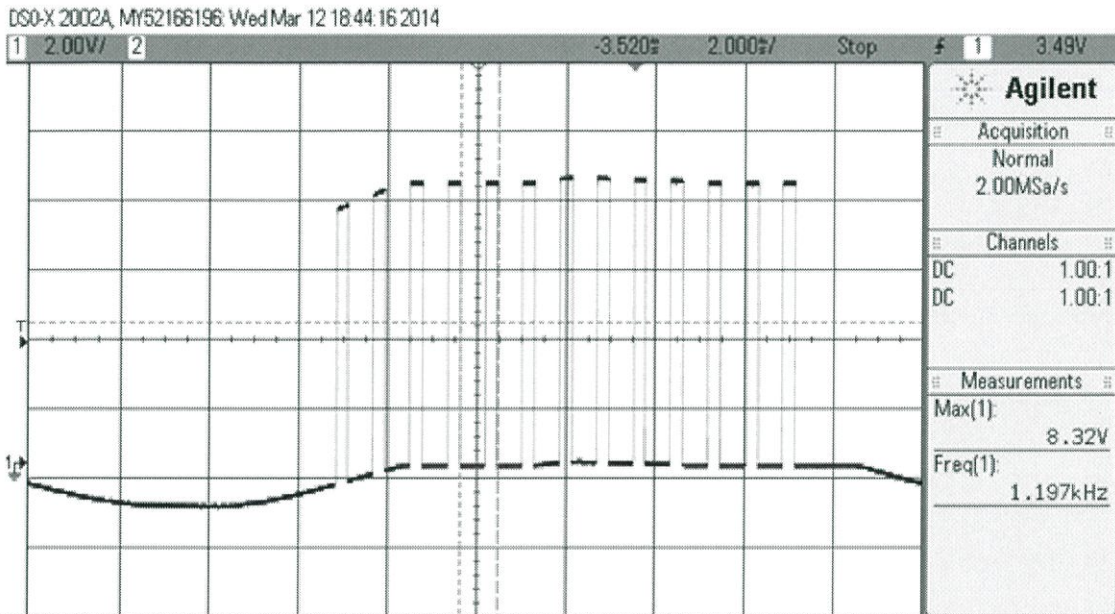
ตารางที่ 4.2.3 ขณะสวิตช์ยังไม่ถูกกด (ไม่มี output)

input	PIN_B4	High
	PIN_B5	High
	PIN_B6	High
	PIN_B7	High

4.3 ผลการทดลองวัดสัญญาณจาก Module ตัวส่งและตัวรับ

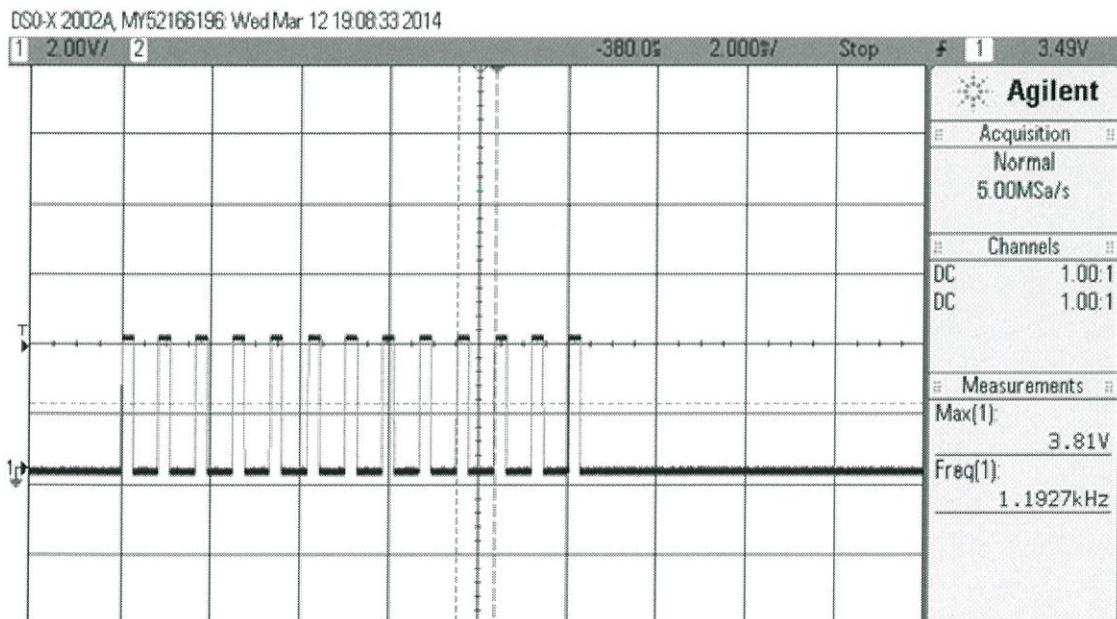
4.3.1 ยังไม่กดสวิตช์ (1111)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.12 สัญญาณจากขา data ตัวส่ง ขณะที่ยังไม่กดสวิตช์ (1111)

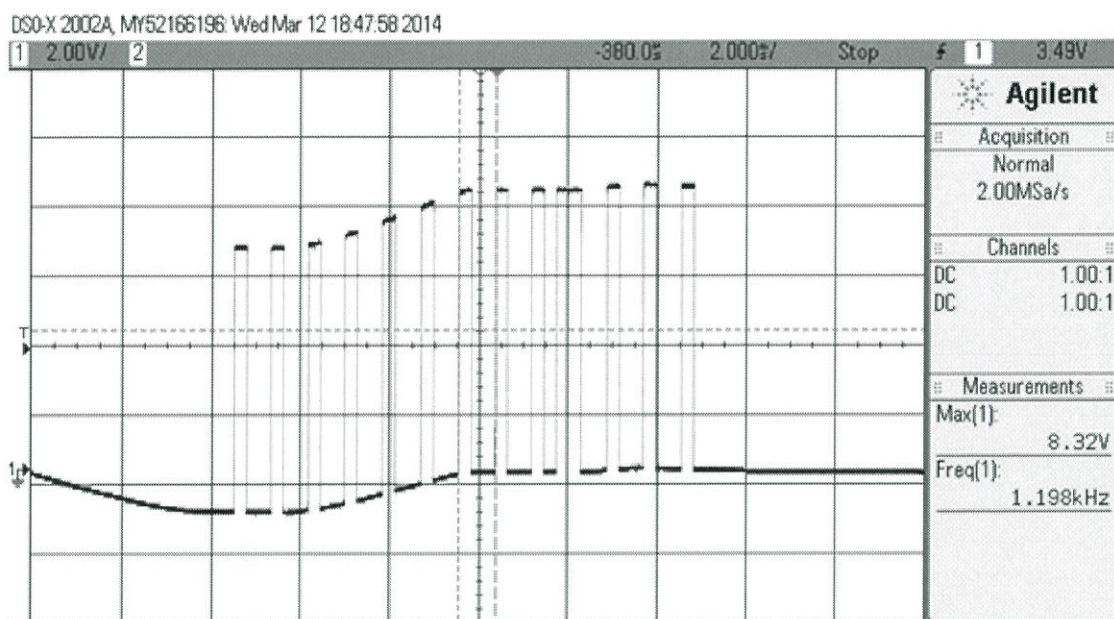
สัญญาณจากขา data ตัวรับ



รูปที่ 4.13 สัญญาณจากขา data ตัวรับ ขณะที่ยังไม่กดสวิตช์ (1111)

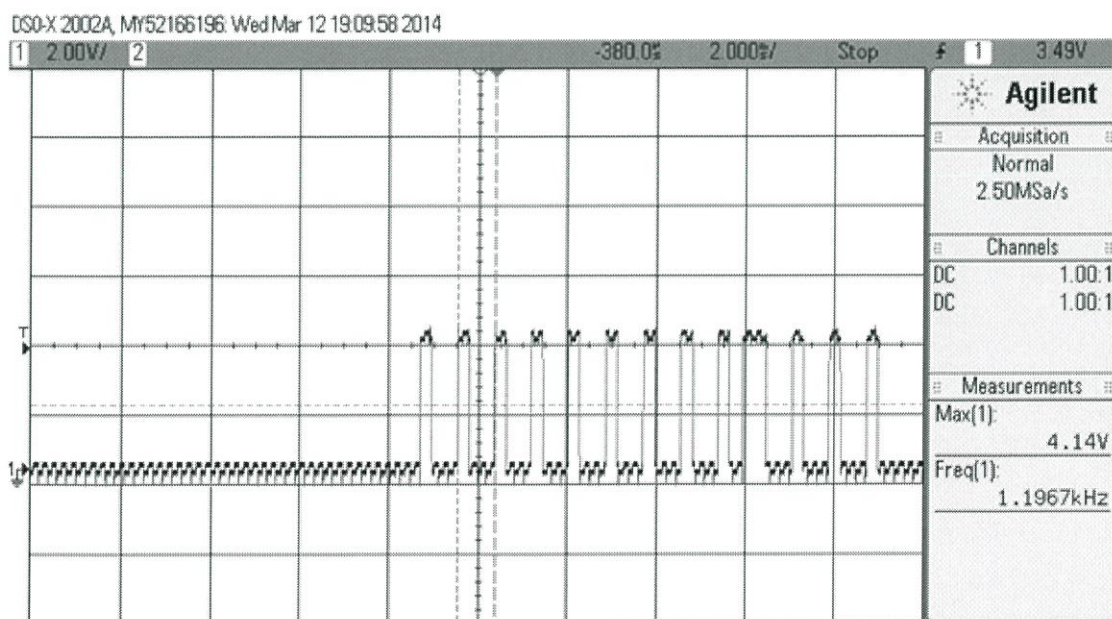
4.3.2 กดสวิตช์ Go (1110)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.14 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Go (1110)

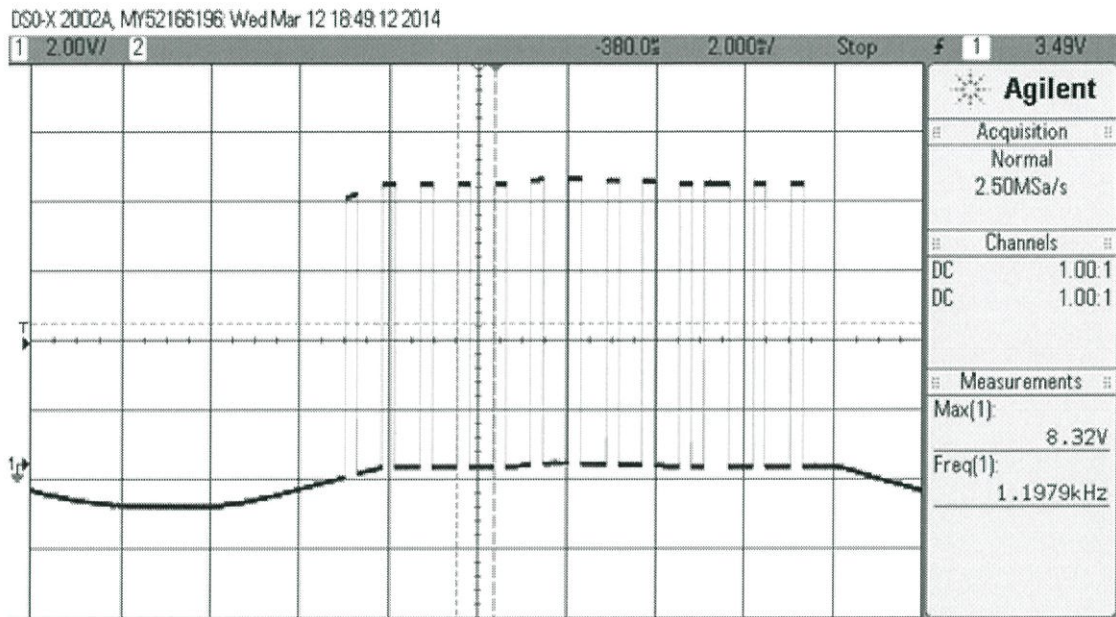
สัญญาณจากขา data ตัวรับ



รูปที่ 4.15 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Go (1110)

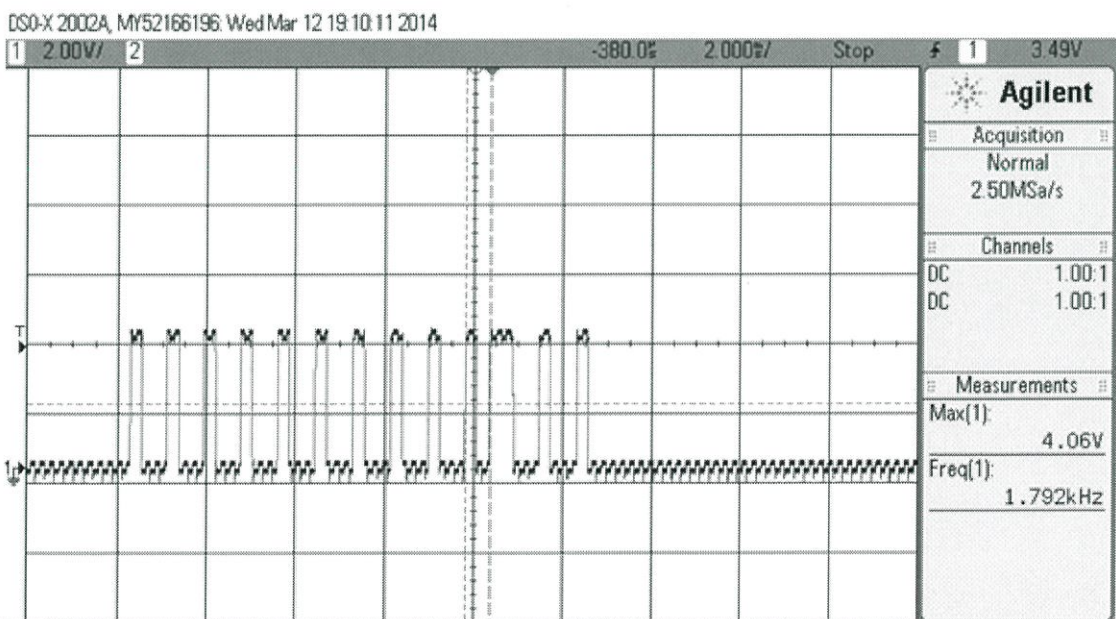
4.3.3 กดสวิตช์ Back (1101)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.16 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Back (1101)

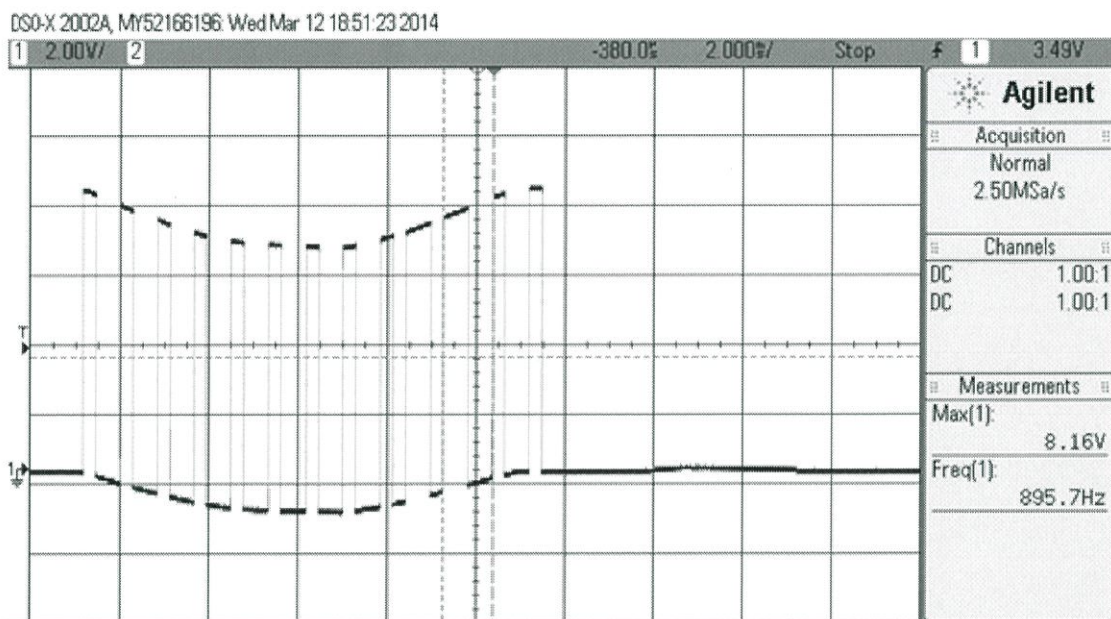
สัญญาณจากขา data ตัวรับ



รูปที่ 4.17 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Back (1101)

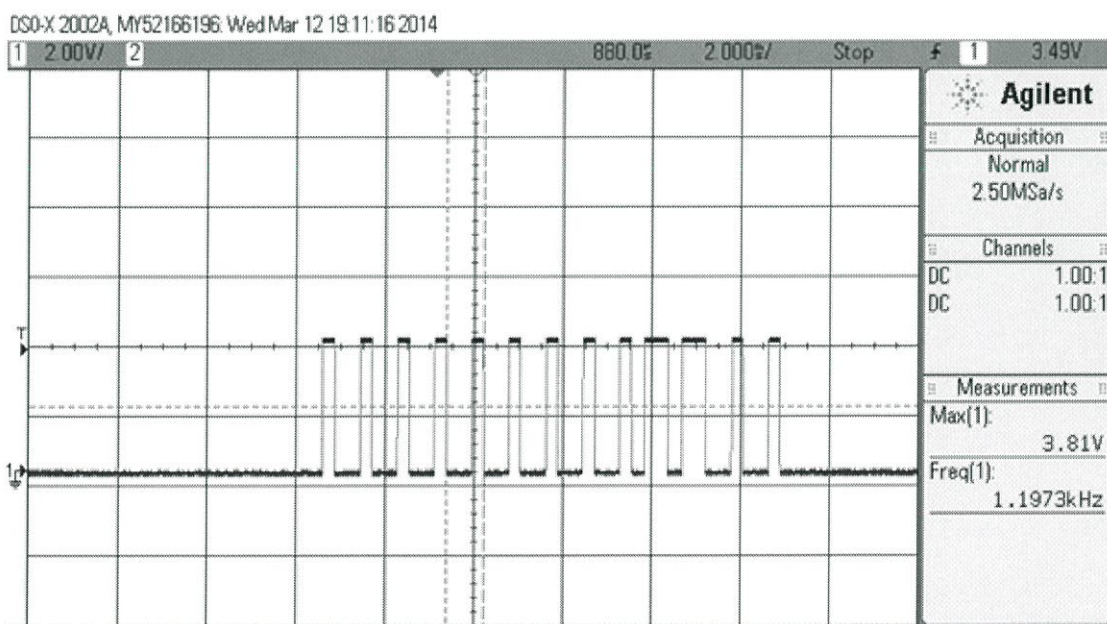
4.3.4 กตสวิตซ์ Speed (1100)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.18 สัญญาณจากขา data ตัวส่ง เมื่อกตสวิตซ์ Speed (1100)

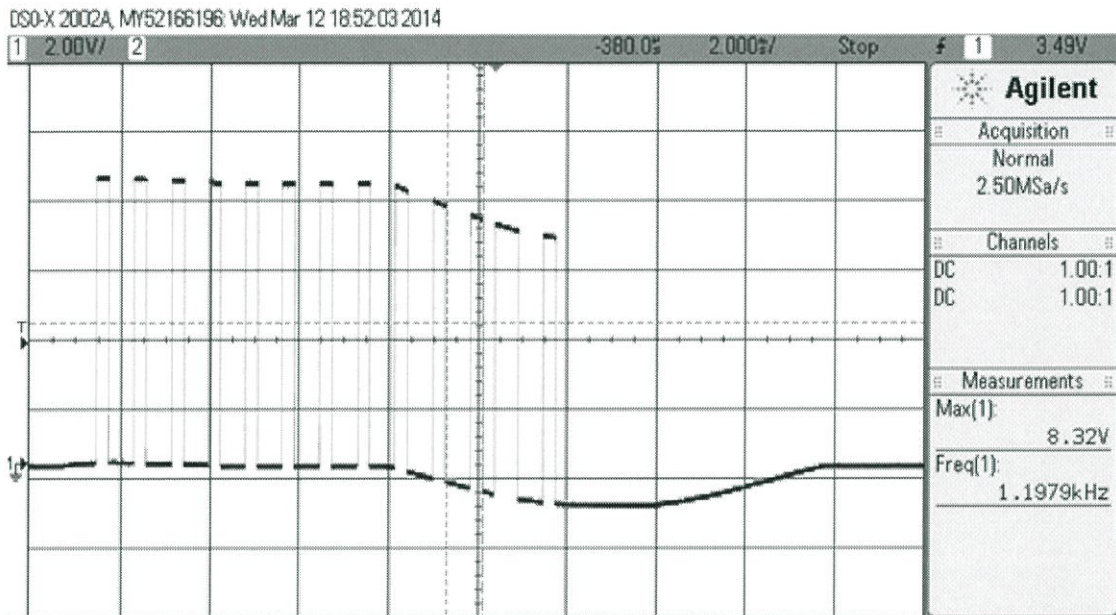
สัญญาณจากขา data ตัวรับ



รูปที่ 4.19 สัญญาณจากขา data ตัวรับ เมื่อกตสวิตซ์ Speed (1100)

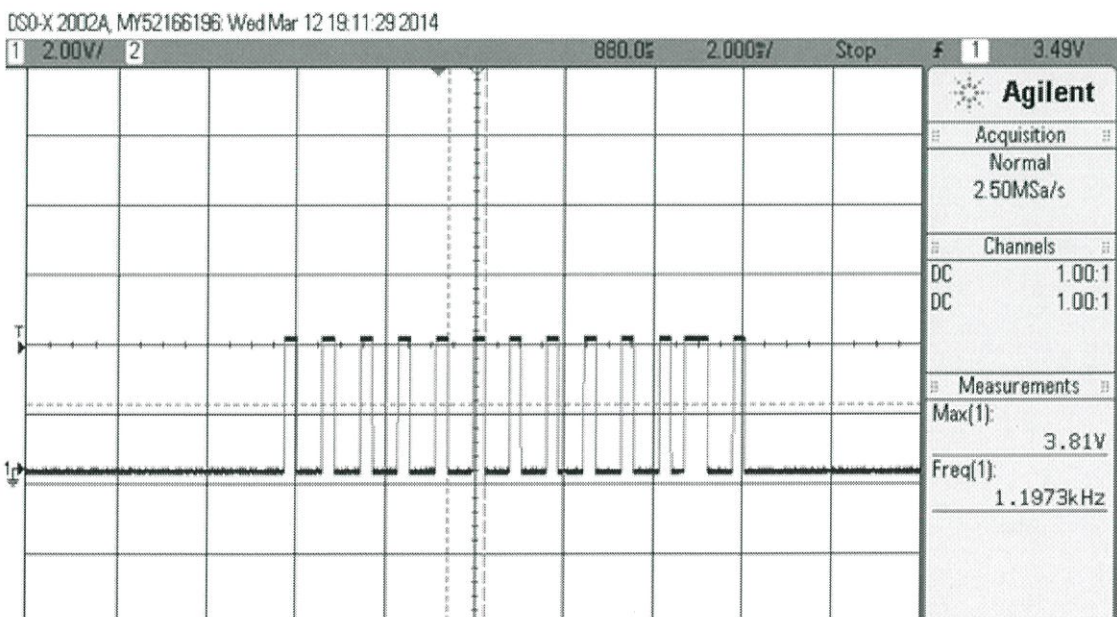
4.3.5 กดสวิตช์ Stop (1011)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.20 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Stop (1011)

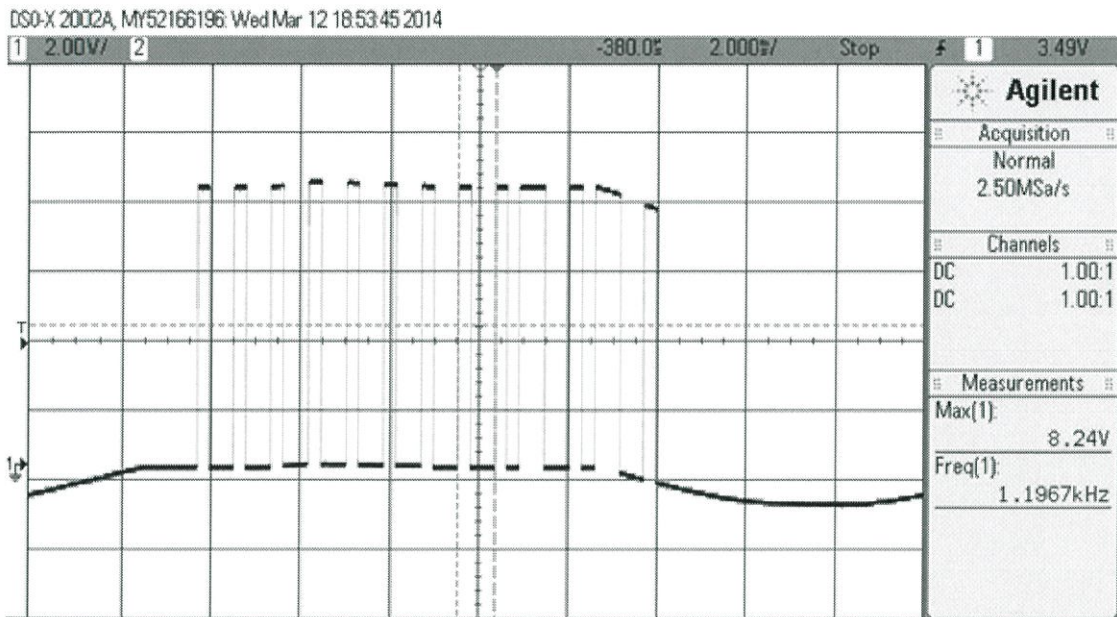
สัญญาณจากขา data ตัวรับ



รูปที่ 4.21 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Stop (1011)

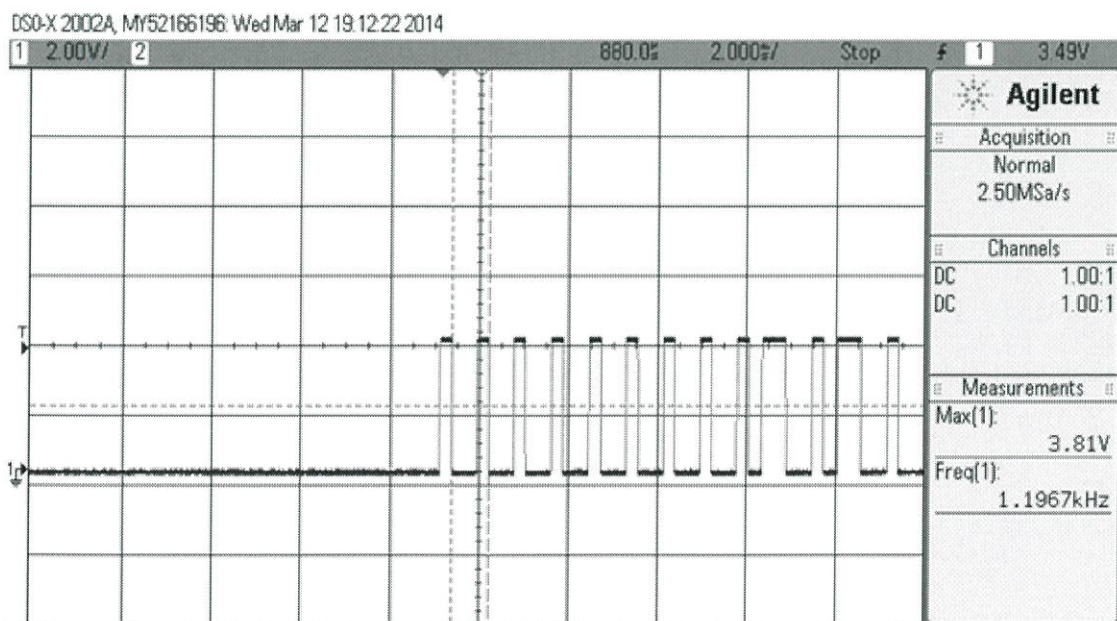
4.3.6 กดสวิตช์ left (1010)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.22 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ left (1010)

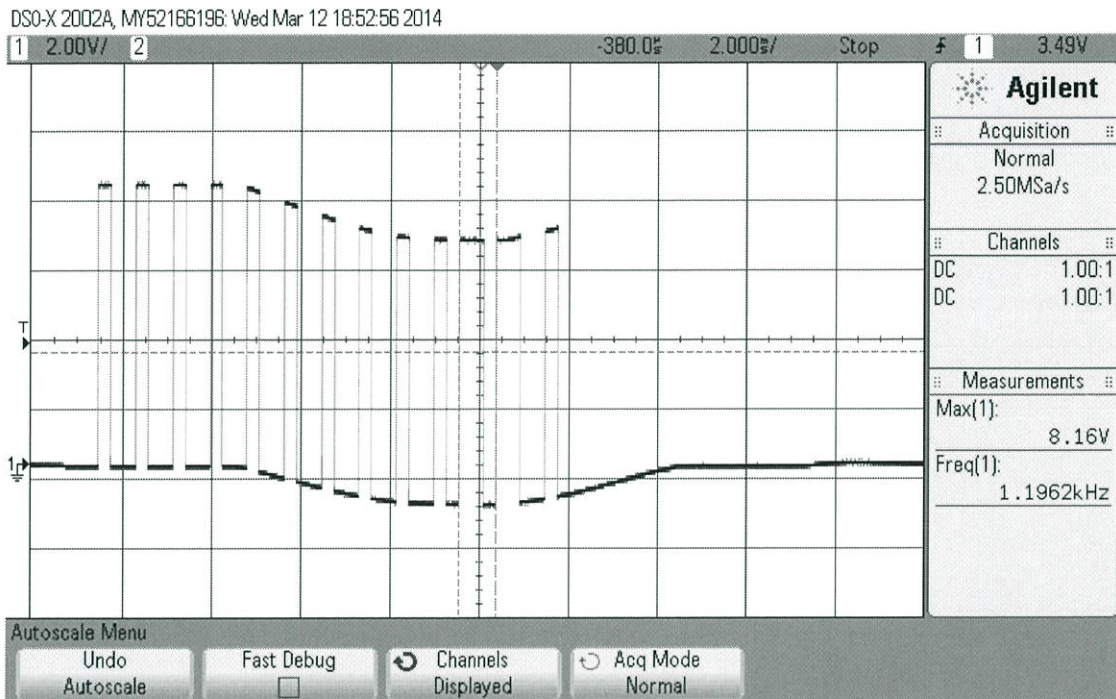
สัญญาณจากขา data ตัวรับ



รูปที่ 4.23 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Left (1010)

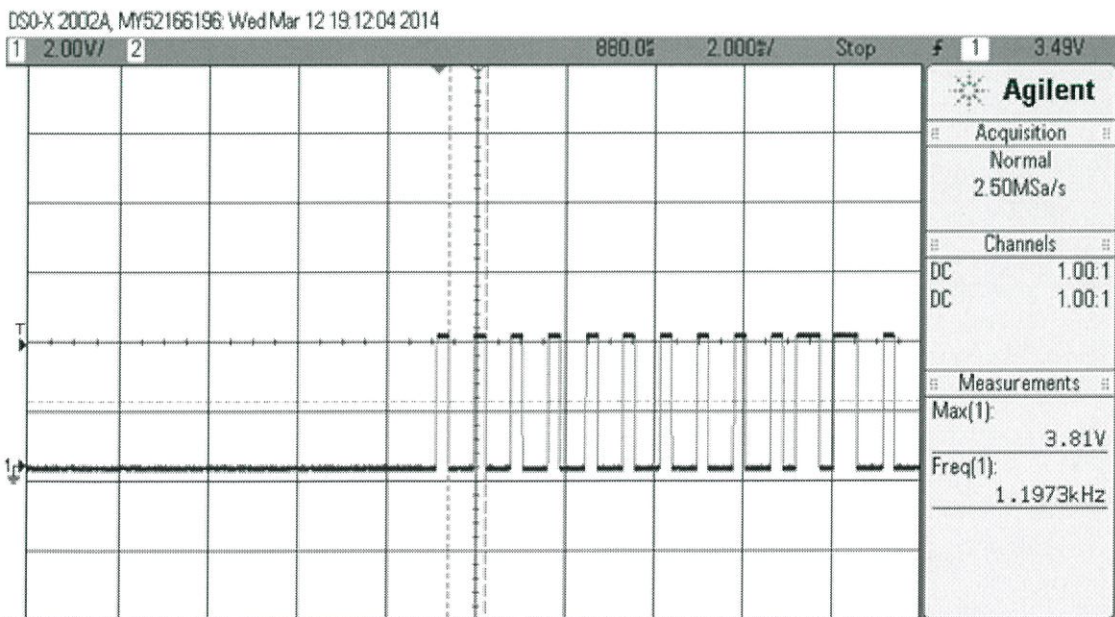
4.3.7 กดสวิตช์ right (1001)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.24 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Right (1001)

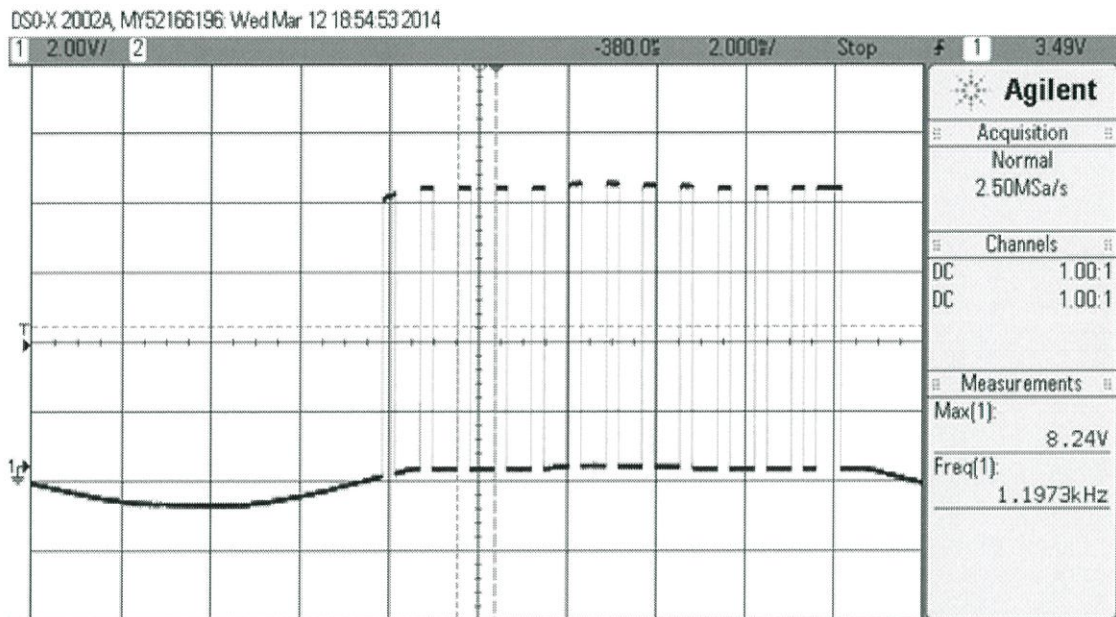
สัญญาณจากขา data ตัวรับ



รูปที่ 4.25 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Right (1001)

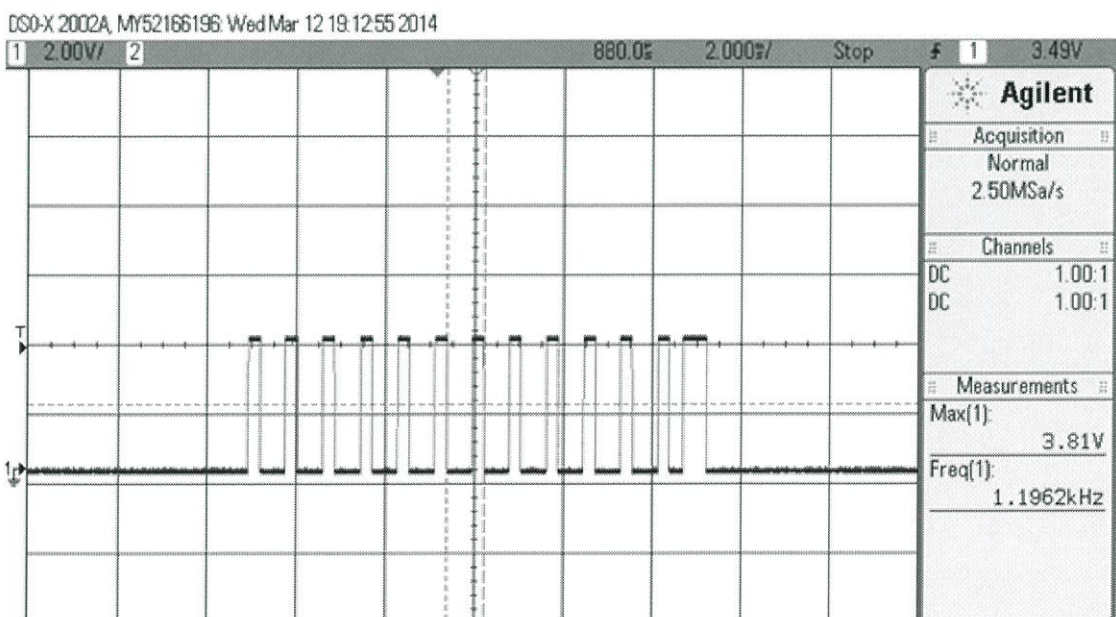
4.3.8 กดสวิตช์ reset (0111)

สัญญาณจากขา data ตัวส่ง



รูปที่ 4.26 สัญญาณจากขา data ตัวส่ง เมื่อกดสวิตช์ Reset (1011)

สัญญาณจากขา data ตัวรับ



รูปที่ 4.27 สัญญาณจากขา data ตัวรับ เมื่อกดสวิตช์ Reset (1011)

บทที่ 5

สรุปผลการทดลอง

เรือบังคับวิทยุในรายงานนี้ เป็นการรับ-ส่งสัญญาณคลื่นวิทยุขนาด 433.92 MHz โดยใช้รีโมทคอนโทรลในการส่งสัญญาณ และใช้ HT12E ในการเข้ารหัส และกำหนด Address ประมวลผลส่งผ่านโมดูล TWS-BS3V1.01 ที่มีการมอดูเลตแบบ ASK และ RWS-374-6 เมื่อรับสัญญาณมา ก็จะส่งข้อมูลให้กับ HT12D เพื่อใช้ในการถอดรหัส และส่งต่อไปในส่วนของ microcontroller เพื่อประมวลผลในการควบคุมการทำงานของมอเตอร์และเซอร์โวมอเตอร์

ผลที่ได้จากการทดลอง คือการควบคุมการเคลื่อนที่ของเรือบังคับวิทยุ โดยสามารถ เดินหน้า ถอยหลัง เพิ่มความเร็วของมอเตอร์ได้ตั้งแต่ 0 รอบต่อวินาที ไปจนถึง 500 รอบต่อวินาที โดยขึ้นอยู่กับค่าเฉลี่ยแรงดันตั้งแต่ 0 ถึง 12V สามารถที่จะเลี้ยงซ้ายและเลี้ยงขวาได้เป็นมุมถึง 90 องศา จึงสามารถทำงานได้ตามการที่กำหนดไว้ และสามารถส่งสัญญาณได้ในระยะ 10 เมตร

ภาคผนวก

ผ.ก โปรแกรมควบคุม

ผ.ก.1 โปรแกรมควบคุมภาครับ

```
#include <18f4520.h>

#fuses HS,NOWDT,NOPROTECT

#use delay(clock=20000000)

int j=0,k;

#INT_RB

void ON_Change_ISR()

{

    int1 sw[5];

    sw[0]=input(PIN_B4);

    sw[1]=input(PIN_B5);

    sw[2]=input(PIN_B6);

    sw[3]=input(PIN_B7);

    if(sw[0]==1&&sw[1]==1&&sw[2]==1&&sw[3]==0) // 1110 go

    {

        output_high(PIN_D3);

        delay_ms(10);

        output_low(PIN_D2);
```

```
        delay_ms(10);

    break;
}

if(sw[0]==1&&sw[1]==1&&sw[2]==0&&sw[3]==1) //1101 back
{
    output_high(PIN_D2);

    delay_ms(10);

    output_low(PIN_D3);

    delay_ms(10);

    break;
}

if(sw[0]==1&&sw[1]==1&&sw[2]==0&&sw[3]==0) //1100 speed
{
    j++;

    break;
}

if(sw[0]==1&&sw[1]==0&&sw[2]==1&&sw[3]==1) //1011 stop
{
    output_low(PIN_D2);

    delay_ms(10);

    output_low(PIN_D3);

    delay_ms(10);

    break;
}
```

```
}

if(sw[0]==1&&sw[1]==0&&sw[2]==1&&sw[3]==0) //1010 left
{
    for(k=0;k<10;k++)
        {output_high(PIN_C1);
            delay_ms(2);
            output_low(PIN_C1);
            delay_ms(20);}
}

if(sw[0]==1&&sw[1]==0&&sw[2]==0&&sw[3]==1) //1001 right
{
    for(k=0;k<10;k++)
        {output_high(PIN_C1);
            delay_ms(1);
            output_low(PIN_C1);
            delay_ms(20);}
}

if(sw[0]==0&&sw[1]==1&&sw[2]==1&&sw[3]==1) //0111 reset
{
    for(k=0;k<10;k++)
        {output_high(PIN_C1);
            delay_us(1500);
            output_low(PIN_C1);
```

```
        delay_ms(20);}

    }

}

void main()

{

    int i;

    long delay[]={100,125,150,200};

    set_tris_b(0xf0);

    set_tris_d(0x00);

    set_tris_c(0x00);

    output_d(0x00);

    enable_interrupts(GLOBAL);

    enable_interrupts(INT_RB);

    while(TRUE)

    {

        for(i=0;i<100;i++)

        {

            output_high(PIN_C2);

            delay_us(delay[j]);

            output_low(PIN_C2);

            delay_us(200-delay[j]);

            if(j>3)j=0;
```

```

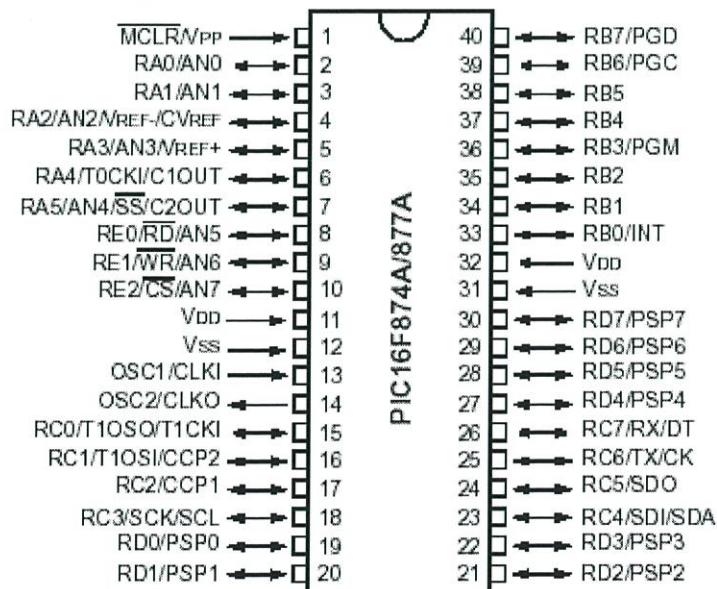
}
}
}

```

ผ.1 ข้อมูลเกี่ยวกับ PIC16F877A

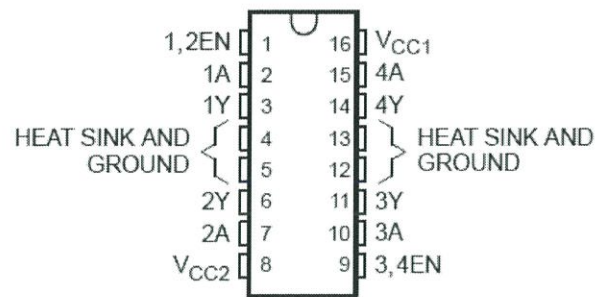
- รองรับสัญญาณนาฬิกาภายนอกได้สูงสุดเป็น 20 MHz
- รองรับการเชื่อมต่อแบบ UART
- มีส่วนแปลงแรงดันอนาล็อกไปเป็นดิจิตอลขนาด 10 Bit ภายในตัว
- มีตัวจับเวลาภายในแยกอิสระต่อกัน 3 ตัว
- รองรับการสร้างสัญญาณขัดจังหวะจากภายนอก
- สามารถทำงานได้ที่ระดับแรงดัน 2.0 ถึง 5.5 V
- รองรับการเชื่อมต่อแบบ SPI และ I^2C
- ตัวจับเวลาภายในสามารถทำงานได้เมื่ออยู่ในสถานะ SLEEP (เฉพาะเมื่อทำการต่อสัญญาณนาฬิกาภายนอก)

40-Pin PDIP



รูปที่ ผ.1.1 การจัดเรียงขาของไมโครคอนโทรลเลอร์ PIC16F877A

ผ.2 ข้อมูลเกี่ยวกับ L293D



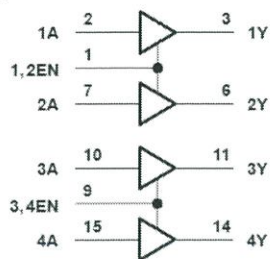
รูปที่ ผ.2.1 โครงสร้างของ L293D

L293D เป็นอุปกรณ์ที่เป็นวงจรรวม ซึ่งมีขาทั้งหมด 16 ขา และมี 4 ขาที่ถูกใช้งานเป็นหลัก ๆ เชื่อมต่อเข้าด้วยกัน และยังเป็นอุปกรณ์ที่ถูกนำมาใช้สำหรับการระบายความร้อน L293D นั้น มีช่องทางสำหรับการขับกระแสสูง 4 ช่องทาง ซึ่งถูกออกแบบมาให้เข้ากับการใช้งานกับ DTL หรือ TTL logic levels ซึ่งเป็นรูปแบบมาตรฐาน และใช้ในการเป็นตัวขับกระแสให้กับโหลด ที่ถูกนำเข้ามา ยกตัวอย่าง เช่น relays solenoids, DC และ มอเตอร์ เป็นต้น อีกทั้งเรายังสามารถนำ L293D มาใช้งานกับ Switching Power transistors ได้เป็นอย่างดีอีกด้วย

เพื่อให้ง่ายต่อการนำมาใช้งาน L293D จึงถูกออกแบบมาให้ มีขา enable input อยู่ 2 ด้าน มีไดโอดอยู่ภายใน และมีการแยก input ของ supply ไว้อย่างชัดเจน เพื่อประโยชน์สำหรับการป้องกันสัญญาณ logic การใช้งาน L293D นั้นเราสามารถนำมาใช้งานกับความต่างศักย์ต่ำๆ ได้

L293D เป็นอุปกรณ์ที่เหมาะสมสำหรับการใช้งานกับ Switching applications ที่มีความถี่ต่ำๆ ขึ้นไปจนถึง 5 kHz

logic diagram

FUNCTION TABLE
(each driver)

INPUTS [‡]		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

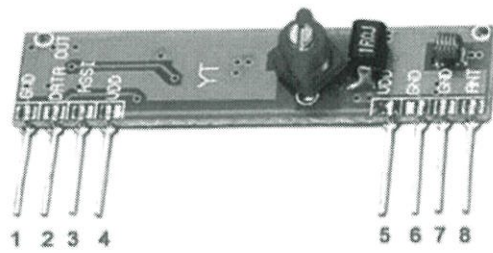
H = high-level, L = low level,
 X = irrelevant, Z = high-impedance (off)
[‡] In the thermal shutdown mode, the
 output is in the high-impedance state
 regardless of the input levels.

รูปที่ ผ.2.2 logic diagram ของ L293D

ผ.3 ข้อมูลเกี่ยวกับ RWS-BS Series Module

- รองรับความถี่ในช่วง 433.92 MHz (RF ASK)
- รองรับการรับส่งสัญญาณมอดดูเลตแบบ ASK
- สามารถทำงานได้ที่ระดับแรงดัน 5 V
- เป็น Circuit Shape แบบ LC
- รับค่า data rate ได้สูงสุด 4800 bps

➤ Pin Assignment:



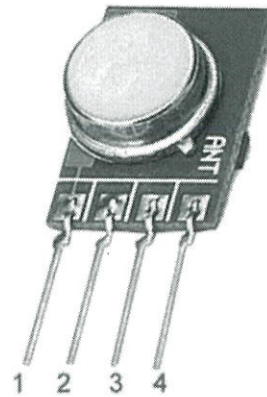
Pin	Function	Description
1	GND	
2	Digital Output	
3	Linear Out	
4	VCC	
5	VCC	
6	GND	
7	GND	
8	ANT	About 13cm

รูปที่ ผ.3.1 แสดงการจัดเรียงขาของ RWS-BS Module

ผ.4 ข้อมูลเกี่ยวกับ TWS-BS Series Module

- รองรับความถี่ในช่วง 433.92 MHz (RF ASK)
- รองรับการทำงานรับส่งสัญญาณมอดดูเลตแบบ ASK
- สามารถทำงานได้ที่ระดับแรงดัน 5 V ถึง 12 V
- เป็น Circuit Shape แบบ SAW
- รับค่า data rate ได้สูงสุด 8 Kbps

➤ **Pin Assignment:**

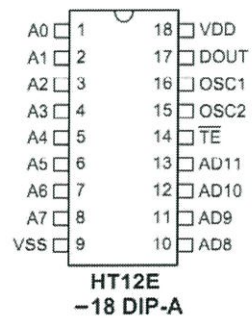


Pin	Function
1	GND
2	Data In
3	VCC
4	ANT

รูปที่ ผ.4.1 แสดงการจัดเรียงขาของ TWS-BS Module

ผ.5 ข้อมูลเกี่ยวกับ HT12E

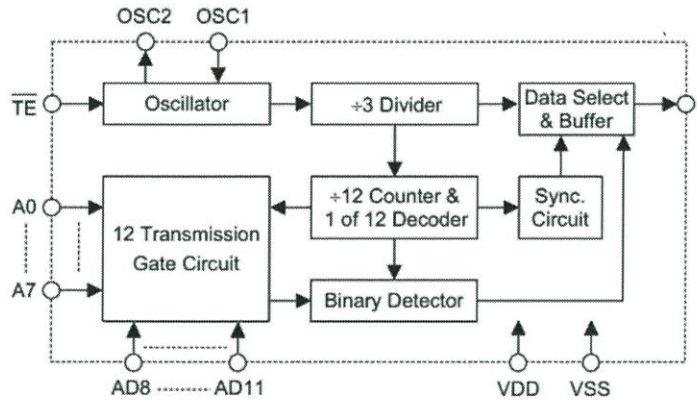
**8-Address
4-Address/Data**



รูปที่ ผ.5.1 โครงสร้างของ HT12E

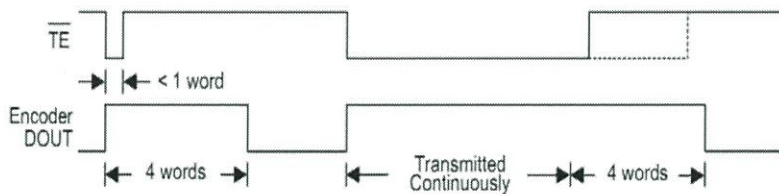
เป็นไอซีที่ใช้ในการเข้ารหัสสัญญาณ (Encoder) ซึ่งสามารถกำหนดแอดเดรส (Address) ได้ทั้งหมด 8 บิต มีตาต้า (Data) จำนวน 4 บิต และมีขา TE เป็นตัว Trigger

Block Diagram



รูปที่ ๕.๒ Block Diagram ของ HT12E

ลักษณะการส่งข้อมูลของ HT12E

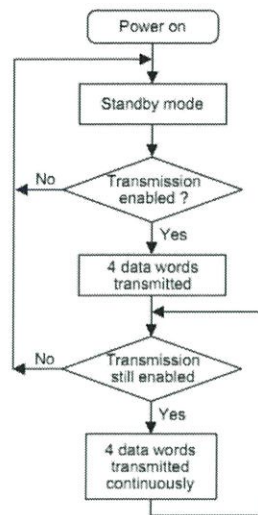


Transmission timing for the HT12E

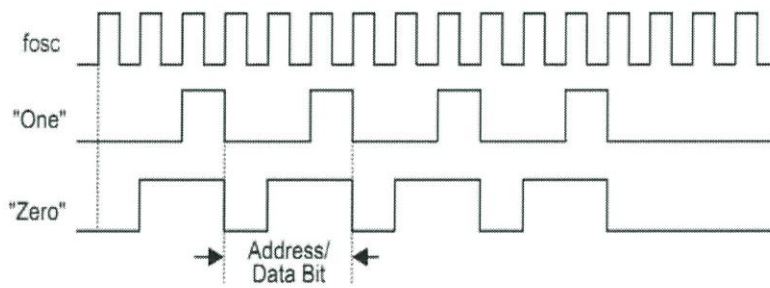
รูปที่ ๕.๓ การส่งข้อมูลของ HT12E

ขา TE Trigger นั้นเป็น active low หมายความว่า ถ้าสัญญาณเป็น low จะทำให้ HT12E นั้นส่งข้อมูลออกไป โดยการส่งจะส่งออกไปคราวละ 4 words ในขณะที่ขา TE มีสถานะเป็น low เป็นเวลาน้อยกว่า 1 word เมื่อขา TE Trigger มีสถานะเป็น low การส่งข้อมูลออกไปของ HT12E นั้นก็จะเป็นไปอย่างต่อเนื่องตลอดเวลา ต่อเมื่อ ขา TE Trigger ได้กลับสถานะจาก low เป็น high การส่งข้อมูลของ HT12E จะไม่หยุดส่งข้อมูลโดยทันทีแต่จะยังมีการส่งข้อมูลต่อเนื่องไปอีก 4 words จึงจะหยุดการส่งข้อมูล

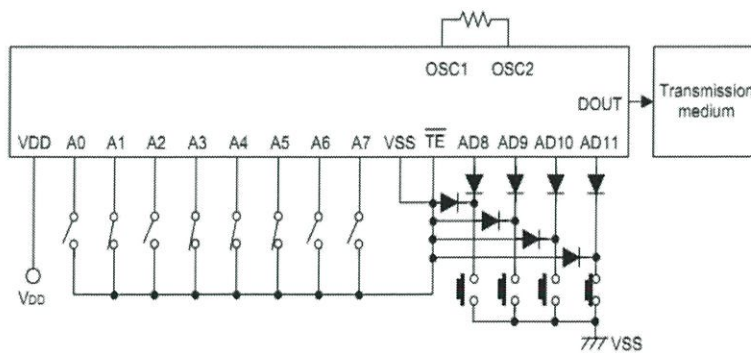
Flowchart



รูปที่ ๕.๔ Flow chart การทำงานของ HT12E

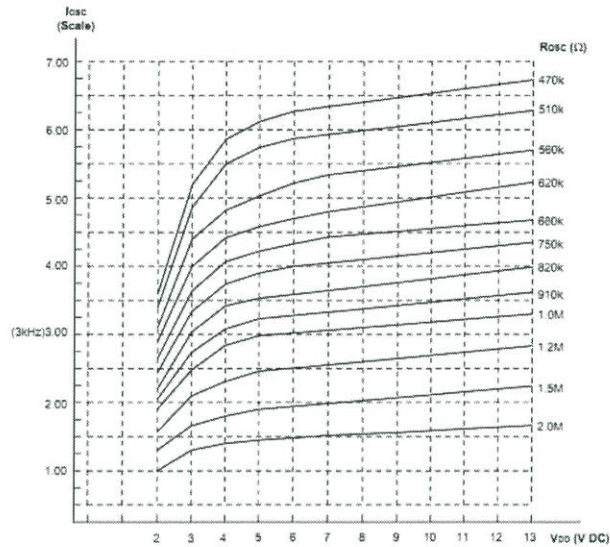


รูปที่ ๕.๕ แสดงรูปคลื่นของ Address/data bit จาก HT12E



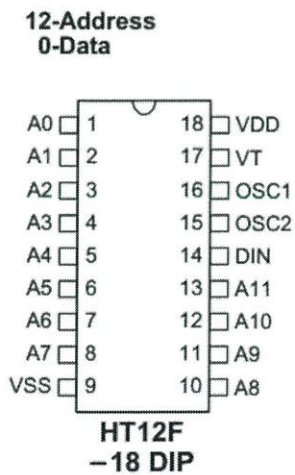
รูปที่ ๕.๖ แสดงการประยุกต์ใช้งาน HT12E

Oscillator Frequency vs. Supply Voltage



รูปที่ ผ.5.7 แสดงการเลือกใช้ R_{osc} (f_{OSCD} (decoder) = 50 f_{OSCE} (HT12E encoder))

ผ.6 ข้อมูลเกี่ยวกับ HT12D

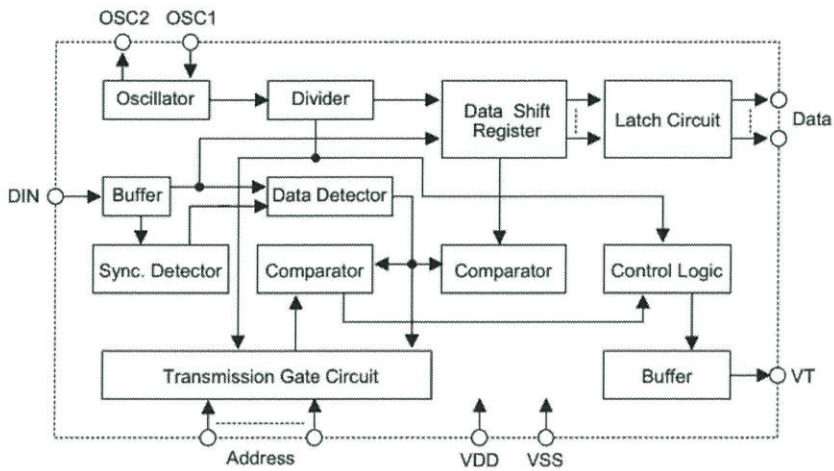


รูปที่ ผ.6.1 โครงสร้างของ HT12D

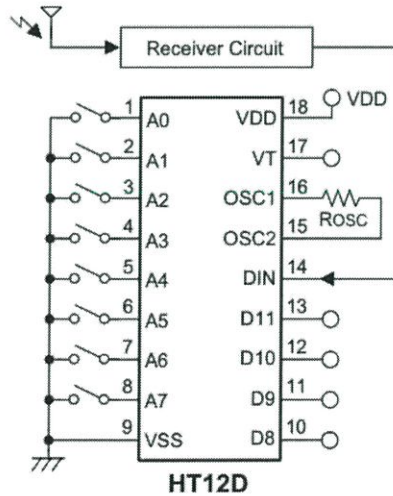
เป็นไอซีที่ใช้ในการถอดรหัส (Decoder) ซึ่งสามารถกำหนด address ได้ทั้งหมด 8 บิต ซึ่ง address นี้จะต้องตั้งให้ตรงกับ address ในส่วนของ HT12E ถ้าตั้ง address ของด้านส่งและด้านรับไม่ตรงกันจะเกิดความผิดพลาดและจะไม่มี output ออก มี data จำนวน 4 บิต โดย output ของ data จะ

มีสถานะติดค้าง (Latch) และมีขา VT เป็นตัวแสดงให้รู้ว่าข้อมูลที่ได้รับมานั้นถูกต้องหรือไม่ โดยถ้าข้อมูลที่รับมามีความถูกต้อง ขา VT จะมีสถานะเป็น High การเลือกใช้ RO5C นั้นจะต้องใช้ FO5C ที่มีค่า 50 เท่า ของ FO5C ที่ใช้กับ HT12E

Block Diagram

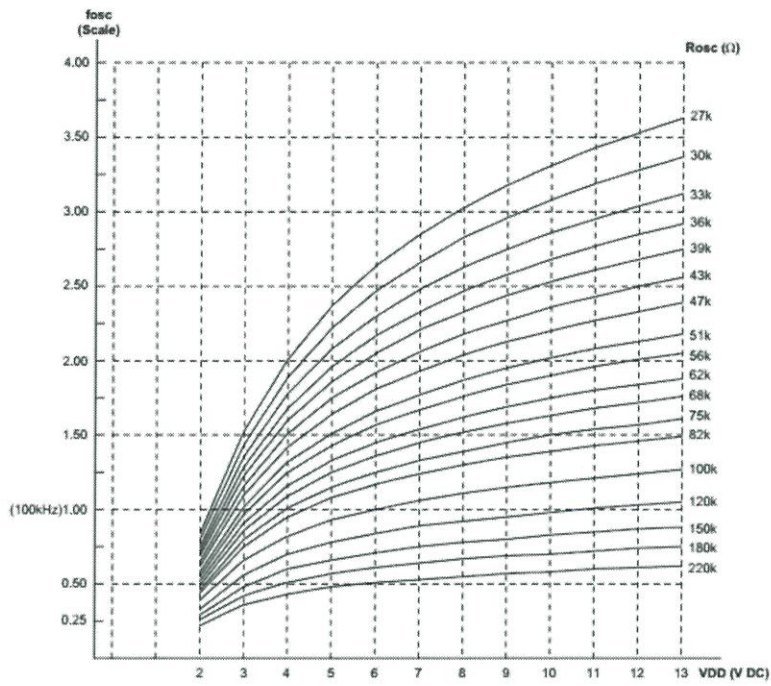


รูปที่ ผ.6.2 Block diagram ของ HT12D



รูปที่ ผ.6.3 แสดงการประยุกต์ใช้งาน HT12D

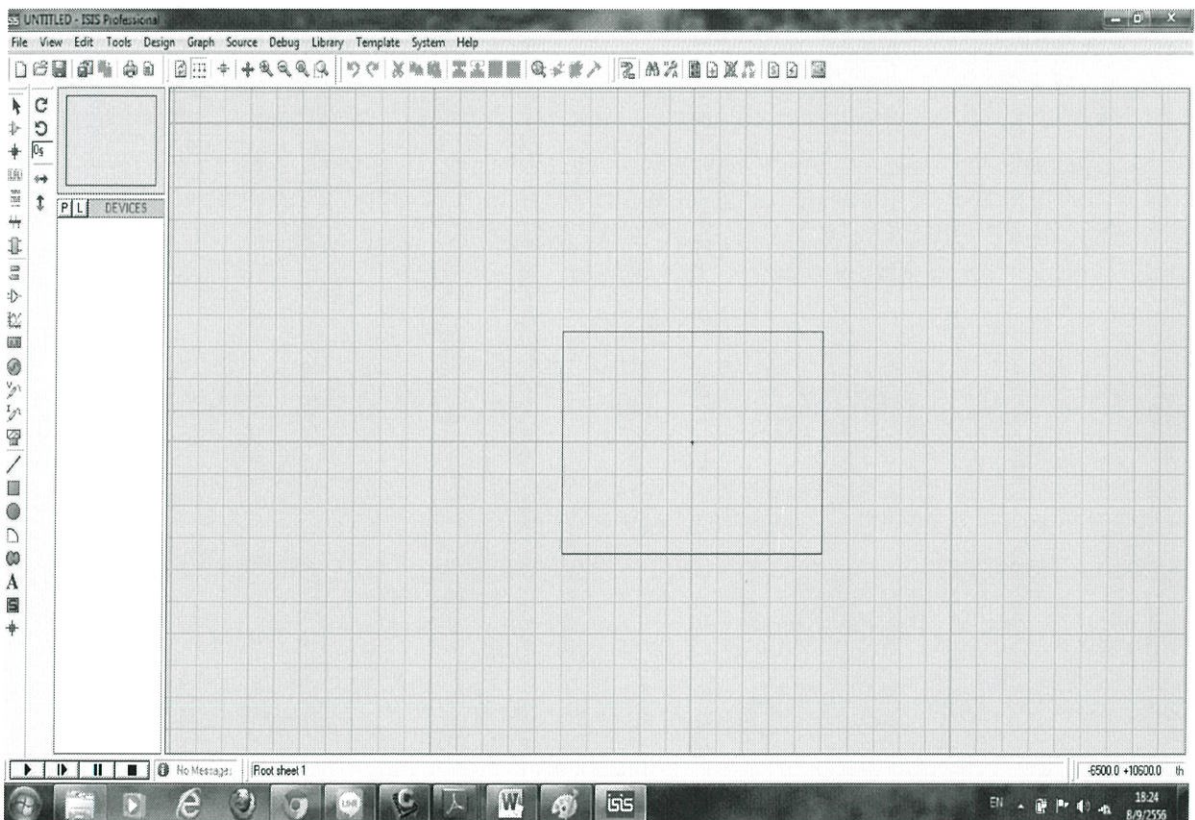
Oscillator frequency vs supply voltage

รูปที่ ผ.6.4 แสดงการเลือกใช้ R_{osc}

ผ.7 ซอฟแวร์ที่ใช้ในการทำงาน

ผ.7.1 โปรแกรม Proteus

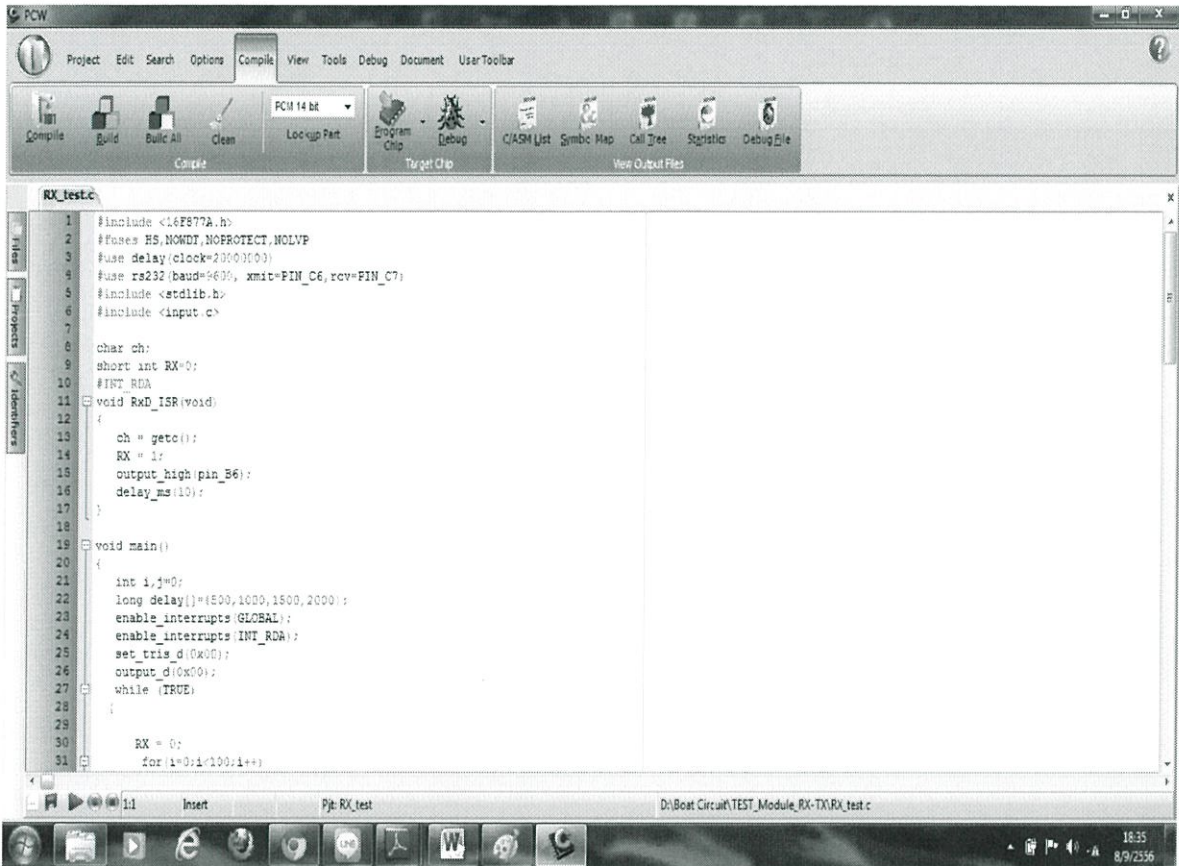
โปรแกรม Proteus เป็นโปรแกรมยอดนิยมของนักออกแบบวงจรอิเล็กทรอนิกส์ วงจรดิจิทัล รวมไปถึงวงจรไมโครคอนโทรลเลอร์ เนื่องจากโปรแกรม Proteus เป็นโปรแกรมที่มีความสามารถครบวงจร นั่นคือ สามารถจำลองการทำงานของวงจรได้ทั้งวงจรรอนาลอกและวงจรดิจิทัล สามารถจำลองการทำงานของไมโครคอนโทรลเลอร์ได้หลายตระกูล สามารถโหลด *.hex ลงไปและดูผลการทำงานของโปรแกรมที่เขียนขึ้นได้ สามารถออกแบบลายวงจรหรือ PCB ได้



รูปที่ ผ.7.1 หน้าจอแสดงผลของโปรแกรม Proteus

ผ.7.2 โปรแกรม CCS PIC C Compiler

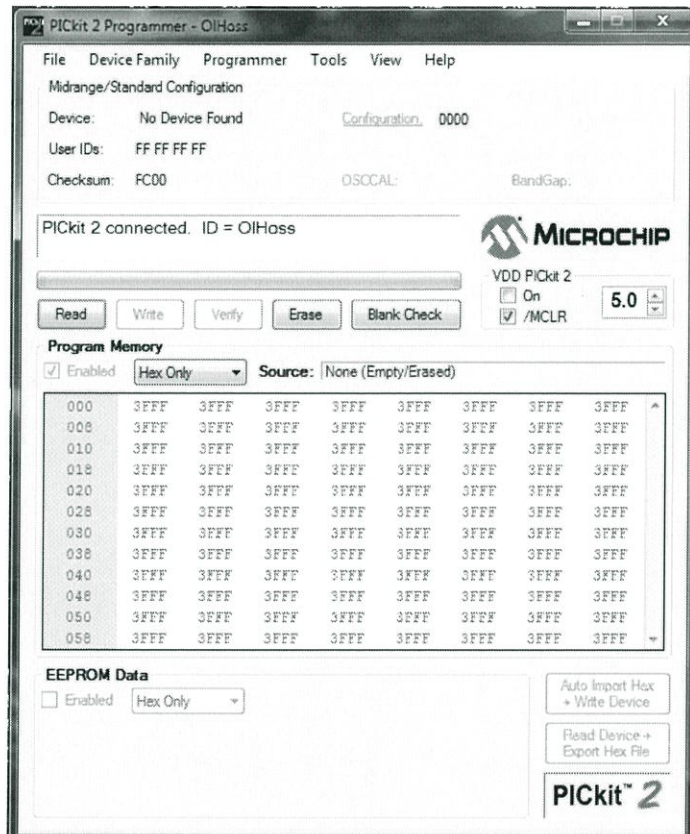
โปรแกรม CCS PIC C Compiler เป็นซอฟต์แวร์สำหรับแปลโปรแกรมภาษา C ของไมโครคอนโทรลเลอร์ PIC เป็นรหัสเครื่องหรือแมคชีนโค้ด (machine code) ผลิตโดย Custom Computer Services สหรัฐอเมริกา



รูปที่ ผ.7.2 หน้าจอแสดงผลของโปรแกรม CCS PIC C Compiler

ผ.7.3 โปรแกรม PICKit 2

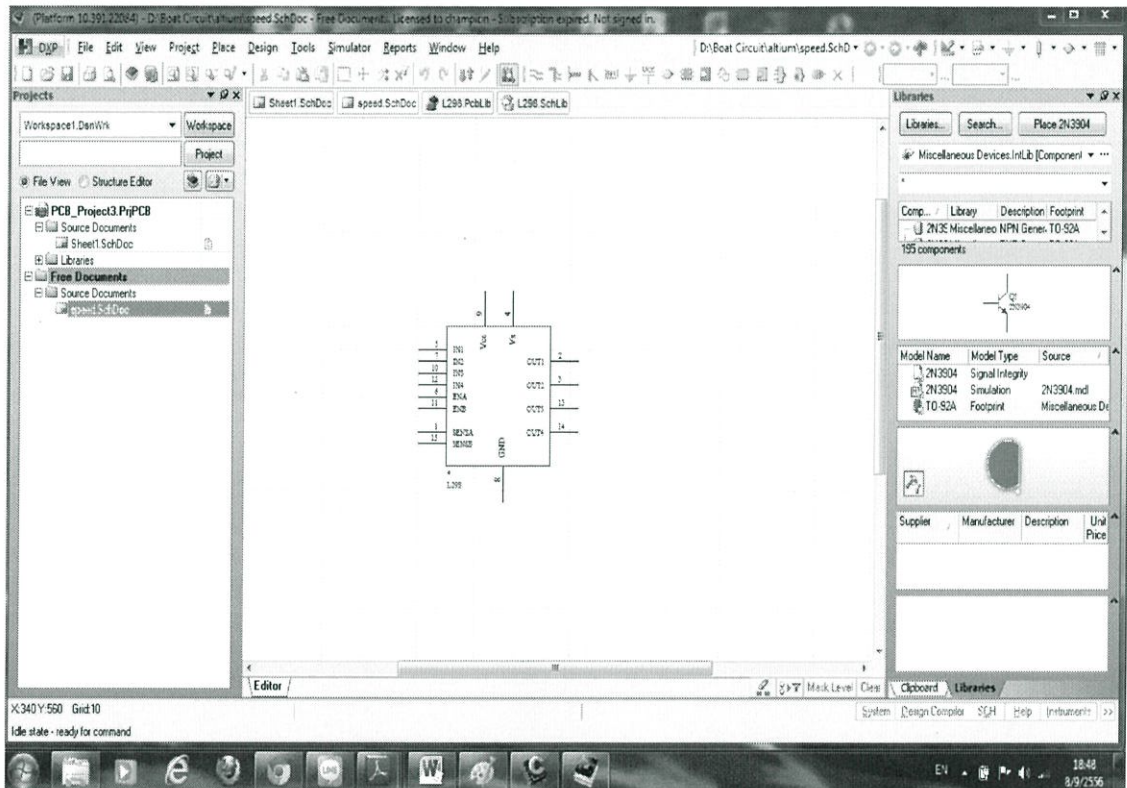
เป็นโปรแกรมประเภท Freeware ที่ใช้สำหรับการดาวน์โหลดหรือโปรแกรมไฟล์นามสกุล .hex ลงบนไมโครคอนโทรลเลอร์อีกหนึ่งโปรแกรม ซึ่งสนับสนุนการทำงานร่วมกับไมโครคอนโทรลเลอร์ เบอร์ PIC10F, PIC12F5xx, PIC16F5xx, PIC12F6xx, PIC16F, PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 ของบริษัท Microchip



รูปที่ ผ.7.3 หน้าจอแสดงผลของโปรแกรม PICKit 2

ผ.7.4 โปรแกรม Altium Designer 10

Altium Designer โปรแกรมที่ใช้สำหรับออกแบบวงจรอิเล็กทรอนิกส์ขนาด 32 บิต ใช้งานบนโปรแกรมวินโดว์ NT/XP โปรแกรม Altium Designer มีชุดเครื่องมือออกแบบที่ทำให้ ท่านสามารถออกแบบวงจรตั้งแต่ขั้นวางแผนความคิด จนกระทั่งเป็นชิ้นงานแผ่นวงจรได้อย่างง่ายดาย เครื่องมือทั้งหมดของโปรแกรม Altium Designer ใช้งานภายใต้โปรแกรมประยุกต์เดียวคือ Design Explorer ทั้งนี้ที่เริ่มใช้งาน หน้าต่าง Design Explorer จะถูกเปิดขึ้น เครื่องมือออกแบบต่างๆจะปรากฏให้ท่านเลือกใช้งานได้ อย่างง่าย ภายใต้สภาวะการใช้งานที่เป็นไปในแนวเดียวกัน



รูปที่ ผ.7.4 หน้าจอแสดงผลของโปรแกรม Altium Designer 10

เอกสารอ้างอิง

1. http://www.premier-ac.co.th/index.php?lay=boardshow&ac=webboard_show&No=1195843
2. http://www.kknic.ac.th/~kkvisart_poon/Webpage/wachira/htdoc/less_5/Servo%20moter.pdf
3. <http://www.sut.ac.th/engineering/electrical/courses/429296/old/course/motor.pdf>
4. http://www.il.mahidol.ac.th/e-media/computer/network/net_datacom3.html
5. https://www.google.co.th/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&ved=0CDwQFjAC&url=http%3A%2F%2Fsuanpalm3.kmutnb.ac.th%2Fteacher%2FFileDL%2Fart176255410054.ppt&ei=NOsGU93nKKqOiAfqkYDABO&usg=AFQjCNHjxzLG FHLpgnYZ2rqEpy8XOmbKFO&sig2=OIO_zOAY3UCslH-sl-HRZw&bvm=bv.61725948,d.aGc