สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์รักษาความปลอดภัย

SECURITY ROBOT

นายนพพล เศรษฐเสียร

นายบัณฑิต ศรีวัฒนางกูร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหมู่.....
เลขทะเบียน 46186
วัน, เดือน, ปี 20 มี.ค. 2546

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

b11288541

หุ่นยนต์รักษาความปลอดภัย

SECURITY ROBOT

โดย

นายนพพล เศรษฐเสียร

นายบัณฑิต ศรีวัฒนางกูร

อาจารย์ที่ปรึกษา

รศ. บรรจง ปิยธำรง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์รักษาความปลอดภัย

SECURITY ROBOT

ผู้จัดทำ

1. นายนพพล เศรษฐเสียร รหัสประจำตัว 41014205
2. นายบัณฑิต ศรีวัฒนางกูร รหัสประจำตัว 41014232

_____ อาจารย์ที่ปรึกษา

(รศ. บรรจง ปิยธำรง)

หุ่นยนต์รักษาความปลอดภัย

นายนพพล เศรษฐเสียร
นายบัณฑิต ศรีวัฒนางกูร
รศ.บรรจง ปิยธำรง อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

โครงงานนี้นำเสนอการควบคุมหุ่นยนต์รักษาความปลอดภัย โดยใช้คอมพิวเตอร์ควบคุมเชื่อมต่อกับอุปกรณ์สื่อสารไร้สาย ผู้ควบคุมสามารถสั่งงานให้หุ่นยนต์ทำงานต่างๆ ได้โดยผ่านโปรแกรมอินเตอร์เฟสควบคุม สำหรับตัวหุ่นยนต์จะถูกควบคุมแบบไร้สายโดยใช้คลื่นวิทยุ ในการรับ-ส่งสัญญาณควบคุม และใช้ไมโครคอนโทรลเลอร์ MCS-51 ในการควบคุมการทำงานต่างๆของหุ่นยนต์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

A Deductive Database Systems Base On A Conceptual Schema

Noppon Settasatean

Bundit Sriwattanangkoon

Assoc. Prof. Bunjong Piyathumrong Advisor

ABSTRACT

This project presents security robot controlling via wireless radio signal which consists of a computer acting as a controller unit. The user can control the robot via controller unit which connects to the radio frequency generator. The controlling signals, generated by the application of interface controling program, will be sent to the server which connects wirelessly to the robot using radio signal. Then the signals will be interpreted by MCS-51 to control the robot's functions.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จลงได้ หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้สำเร็จลงได้ก็คือ อาจารย์ บรรจง ปียธำรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณ อาจารย์ทุกๆท่าน เพื่อนๆที่คอยให้คำแนะนำ และใจกำลังใจมาโดยตลอด

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้าทั้งสองมีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักเป็นอย่างยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆด้าน ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายนพพล เศรษฐเสียร

นายบัณฑิต ศรีวัฒนางกูร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย		I
บทคัดย่อภาษาอังกฤษ		II
กิตติกรรมประกาศ		III
สารบัญ		IV
สารบัญตาราง		VI
สารบัญภาพ		VIII
บทที่ 1	บทนำ	1
	1.1 ความสำคัญและที่มา	1
	1.2 ขอบเขตของงานวิจัย	1
	1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
	1.4 วิธีการดำเนินงาน	1
บทที่ 2	การสื่อสารไร้สายด้วยคลื่นวิทยุความถี่สูงระยะไกล	4
	2.1 หลักการและทฤษฎีของระบบควบคุมระยะไกลแบบไร้สาย	4
	2.2 ความรู้พื้นฐานเกี่ยวกับอุปกรณ์ส่งคลื่นวิทยุ	5
	2.4 อุปกรณ์ส่งสัญญาณ	5
	2.5 สัญญาณจากสัญญาณรหัสกว้าง	7
	2.6 อุปกรณ์รับสัญญาณ (Receiver)	8
	2.7 การสื่อสารด้วยพัลส์	8
	2.8 วงจร Pulse Width Modulation	10
	2.9 วงจรกำเนิดความถี่วิทยุ (Radio Frequency Oscilator)	10
	2.10 การทำงานของภาคส่งสัญญาณสื่อสาร	11
	2.11 การทำงานของภาครับสัญญาณ	11
	2.12 ปัญหาการทำงานของการสื่อสารไร้สายด้วยคลื่นวิทยุ	11
บทที่ 3	ไมโครคอนโทรลเลอร์ตระกูล MCS-51	12
	3.1 คำสั่งที่ ใช้กับ MCS-51	12
	3.1 วิธีการเข้าถึงข้อมูลสำหรับ MCS-51	16
	3.3 การอินเตอร์รัพต์	16
	3.4 อิทเทอร์รัพท์อีนาเบิลรีจิสเตอร์ (Interrupt Enable Register : IE Register)	17
	3.5 ไทม์เมอร์/เคานท์เตอร์ คอนโทรลรีจิสเตอร์ (TCON)	19
	3.6 ไทม์เมอร์ / เคาน์เตอร์	20
	3.7 การใช้งานพอรต์สื่อสารอนุกรม	23

บทที่ 4	การสื่อสารข้อมูลแบบอนุกรม	26
	4.1 มาตรฐาน EIA RS-232	26
	4.2 รูปแบบของข้อมูลอนุกรมและอัตราบอดในการสื่อสารข้อมูลอนุกรม	28
	4.3 การเชื่อมต่อแบบอนุกรมและ UART	29
	4.4 MCS-51 กับการส่งข้อมูลแบบอนุกรม	32
	4.5 ซีเรียลพอร์ตคอนโทรลรีจิสเตอร์ (Seial Port Control Register)	33
	4.6 โหมดการนทำงาน (Mode Of Operation)	34
	4.7 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล	36
	4.8 อัตราการส่งข้อมูลทางพอร์ตอนุกรม	37
	4.9 การใช้ไทม์เมอร์ 1 กำหนด อัตราบอด Clock	38
	4.10 การเชื่อมต่อหน่วยความจำกับ MCS-51	39
	4.11 การเชื่อมต่อหน่วยความจำกับ MCS-51	40
	4.12 ถอดรหัสตำแหน่งของหน่วยความจำ	41
บทที่ 5	การทำงานของสเต็ปปิงมอเตอร์	44
	5.1 การสั่งงานควบคุมการหมุนของสเต็ปปิงมอเตอร์	44
	5.2 ชนิดของสเต็ปปิงมอเตอร์	47
	5.3 การเชื่อมต่อสเต็ปปิงมอเตอร์กับ ULN2803 (stepping driver motor)	47
บทที่ 6	การออกแบบการสร้าง และการคำนวณ	49
	6.1 ส่วนการทำงานของอุปกรณ์ควบคุมฮาร์ดแวร์	49
	6.2 วงจรเปลี่ยนระดับสัญญาณ RS-232 เป็นระดับสัญญาณ TTL	49
	6.3 วงจรแสดงผลข้อมูลที่ได้รับจากคอมพิวเตอร์	50
	6.4 ส่วนส่งสัญญาณควบคุมไร้สายและรับภาพไร้สาย	51
	6.5 ส่วนรถบังคับ	51
	6.6 วงจรขับเคลื่อนมอเตอร์	52
	6.7 การส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม	52
	6.8 โปรแกรมควบคุมหุ่นยนต์ผ่านพอร์ตอนุกรม	53
	6.9 โปรแกรมควบคุม Security Robot	54
บทที่ 7	ผลการทดลอง	56
	7.1 การติดต่อกับพอร์ตอนุกรม	56
	7.2 ผลการทดลองในส่วนของการรับภาพ	57
บทที่ 8	บทสรุปและวิจารณ์	63

สารบัญตาราง

หน้าที่

บทที่ 1 บทนำ 1

บทที่ 2 การสื่อสารไร้สายด้วยคลื่นวิทยุความถี่สูงระยะไกล 4

บทที่ 3 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 12

3.1 คำสั่งทางคณิตศาสตร์ 13

3.2 คำสั่งทางตรรกะ 13

3.3 คำสั่งทางการเคลื่อนย้ายข้อมูล 14

3.4 คำสั่งการประมวลผลแบบบูลีน 15

3.5 คำสั่งควบคุมลำดับการทำงาน 16

3.6 แสดงชื่อสัญญาณอินเตอร์รัพต์ 17

3.7 รายละเอียดการทำงานของ IE Register 18

3.8 รายละเอียดการทำงานของ IP Register 19

3.9 รายละเอียดการทำงานของ TCON รีจิสเตอร์ 20

3.10 รายละเอียดการทำงานของ TMODRegister 21

3.11 รายละเอียดของโหมดไทม์เมอร์ /เคาน์เตอร์ 21

3.12 รายละเอียดการทำงานของ SCON Register 24

3.13 แสดงรายละเอียดของโหมดการสื่อสารอนุกรม 24

3.14 แสดงการกำหนด Baud Rate โดยไทม์เมอร์ 1 25

บทที่ 4 การสื่อสารข้อมูลแบบอนุกรม 26

4.1 แสดงข้อมูลการเชื่อมต่ออนุกรม 9 ขา 27

4.2 อัตราบอด 29

4.3 อัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม 31

4.4 บิตต่างๆของรีจิสเตอร์ 34

4.5 โหมดต่างๆของการรับส่งแบบอนุกรม 34

4.6 ความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ 39

4.7 รหัสตำแหน่งหน่วยความจำขนาด 4 กิโลไบต์ 42

4.8 ค่าความจริงของ 74LS138 43

บทที่ 5 การทำงานของสเต็ปปิงมอเตอร์ 44

5.1ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบเวฟ 45

5.2 ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบ 2 เฟส 45

5.3 ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบ 4 เฟส 46

5.4 ตารางแสดงคุณสมบัติพื้นฐานของไอซี UNL2803 48

บทที่ 6 การออกแบบการสร้าง และการคำนวณ 49
บทที่ 7 ผลการทดลอง 55
7.1 การแปลงข้อมูลจาก ASCII ไปเป็นบิต 56
7.2 แสดงผลลัพธ์ที่เกิดขึ้นหลังจากกดปุ่ม 57
บทที่ 8 บทสรุปและวิจารณ์ 63

สารบัญภาพ

หน้าที่

บทที่ 1	บทนำ	1
	1.1 บล็อกไดอะแกรมแสดงการควบคุมโรบอท	2
	1.2 บล็อกไดอะแกรมการควบคุมโรบอทโดยใช้โปรแกรมติดต่อกับผู้ใช้	3
บทที่ 2	การสื่อสารไร้สายด้วยคลื่นวิทยุความถี่สูงระยะไกล	4
	2.1 บล็อกไดอะแกรมแสดงหลักการทำงานภาคส่งของระบบคลื่นวิทยุ	4
	2.2 บล็อกไดอะแกรมแสดงหลักการทำงานภาครับของระบบคลื่นวิทยุ	4
	2.3 บล็อกไดอะแกรมแสดงรูปแบบการส่งสัญญาณแบบพัลส์	5
	2.4 แสดงหลักการทำงานของระบบคลื่นวิทยุผ่านทางตัวกลางที่เป็นอากาศ	5
	2.5 การใช้สัญญาณพัลส์กระตุ้นให้อุปกรณ์ควบคุมสวิตซ์ทำงาน	6
	2.6 วงจรกำหนดความกว้างของพัลส์	6
	2.7 วงจรผลิตความถี่	7
	2.8 วงจรส่งคลื่นวิทยุที่มีทั้งวงจรผลิตความถี่และวงจรกำหนดความกว้างสัญญาณ	7
	2.9 วงจรแปลงสัญญาณคลื่นวิทยุ	8
	2.10 สัญญาณพัลส์	8
	2.11 วงจรปรับสัญญาณพัลส์	9
	2.12 วงจรกรองสัญญาณความถี่ต่ำ	10
	2.13 วงจรกำเนิดความถี่วิทยุ	10
บทที่ 3	ไมโครคอนโทรลเลอร์ตระกูล MCS-51	12
	3.1 แสดงโครงสร้าง IE Register	17
	3.2 แสดงโครงสร้าง IP Register	18
	3.3 แสดงโครงสร้าง TCON Register	19
	3.4 แสดงโครงสร้าง TMOD Register	20
	3.5 แสดงโครงสร้าง SCON Register	23
บทที่ 4	การสื่อสารข้อมูลแบบอนุกรม	25
	4.1การเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ DCE	28
	4.2 การเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ DTE	28
	4.3 การส่งข้อมูลแบบอนุกรม	29
	4.4 การแปลงข้อมูลอนุกรมไปเป็นข้อมูลแบบขนาน	30
	4.5 ข้อมูลอนุกรมขนาด 8 บิต	30
	4.6 การรับและส่งข้อมูลอนุกรมใน MCS-51	33
	4.7 การรับส่งข้อมูลออกทางพอร์ตอนุกรมใน MCS-51	35

4.8 โหมดต่างๆของอัตราการส่งข้อมูลออกทางพอร์ตอนุกรม 38
4.9 การเชื่อมต่อหน่วยความจำกับ MCS-51 41
4.10 การเชื่อมต่อหน่วยความจำกับ MCS-51ผ่านมัลติเพลกเซอร์ 41
4.11 ไอซี 74LS138 42
บทที่ 5 การทำงานของสเต็ปปิงมอเตอร์ 44
5.1 ULN2803 48
บทที่ 6 การออกแบบการสร้าง และการคำนวณ 49
6.1 วงจรเปลี่ยนระดับสัญญาณของ RS-232 เป็นระดับสัญญาณ TTL 50
6.2 การต่อขาคอนเนคเตอร์ที่ติดต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ 50
6.3 วงจรแสดงผลข้อมูลที่ได้รับจากคอมพิวเตอร์ 51
6.4 การทำงานของโปรแกรมส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม 53
6.5 แสดงการทำงานของโปรแกรมส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม 54
6.6 แสดง component ตางๆที่ทำงานร่วมกันในโปรแกรมควบคุม 55
บทที่ 7 ผลการทดลอง 56
7.1 ตัวอย่างลักษณะหน้าต่างโปรแกรมเมื่อมีการป้อนอินพุต 57
7.2 ผลการทดลองในการรับภาพ 58
7.3 แสดงส่วนขยายของปุ่ม Video DLG 59
7.4 แสดงส่วนขยายของปุ่ม Source DLG (1) 60
7.5 แสดงส่วนขยายของปุ่ม Source DLG (2) 61
7.6 แสดงส่วนขยายของปุ่ม Comp DLG 61
7.7 แสดงส่วนขยายของปุ่ม Audio DLG 62
บทที่ 8 บทสรุปและวิจารณ์ 63

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

หุ่นยนต์ หรือ โรบอต เป็นสิ่งประดิษฐ์ซึ่งมนุษย์สร้างขึ้นเพื่อทำหน้าที่ต่างๆแทนมนุษย์ ทั้งเพื่อความสะดวกสบาย และเพื่อทดแทนการทำงานของมนุษย์ เพื่อผลประโยชน์ทางธุรกิจ รวมทั้งเพื่อทำงานในที่เสี่ยงอันตรายแทนมนุษย์ได้ เช่น การส่งหุ่นยนต์ขึ้นไปสำรวจดาวอังคารเพื่อตรวจสอบสภาพแวดล้อม สิ่งมีชีวิตและแผนที่บางส่วนของดาวอังคาร เป็นต้น

ซีเคียวริตีโรบอต คือ หุ่นยนต์รักษาความปลอดภัยที่มีการทำงานคล้ายยามรักษาการณ์ โดยเราสามารถควบคุมโรบอตผ่านทางคอมพิวเตอร์เพื่อทำหน้าที่ในการตรวจรักษาความปลอดภัย เพื่อลดความเสี่ยงซึ่งอาจเกิดกับบุคคลในการเข้าตรวจสอบสถานที่ที่มีอันตรายต่างๆ เพื่อป้องกันรักษาชีวิตและทรัพย์สิน ซึ่งในการติดต่อเพื่อสั่งงานหุ่นยนต์จะใช้การสื่อสารแบบไร้สาย

1.2 ขอบเขตการทำงาน

ในโครงงานหุ่นยนต์รักษาความปลอดภัยนี้ จะทำการควบคุมหุ่นยนต์และรับสัญญาณภาพ ผ่านกล้องวิดีโอและอุปกรณ์สื่อสารไร้สายแบบคลื่นวิทยุ โดยสามารถสั่งงานได้จากทั้งคอมพิวเตอร์และคีย์บอร์ดหรือปุ่มควบคุม ทางด้านรับก็จะทำการแปลงสัญญาณที่รับได้เพื่อไปสั่งงานวงจรขับเคลื่อนมอเตอร์เพื่อควบคุมการทำงานของหุ่นยนต์ต่อไป

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถดูแลและตรวจสอบสถานที่ระยะไกลได้
- สามารถเฝ้าคอยระวังทรัพย์สินโดยที่ไม่ต้องจ้างคนมาคอยดูแล
- ตัดปัญหาเรื่องการว่าจ้างคนมาดูแลได้ทั้งหมด
- สามารถใช้งานได้ทันทีตลอดเวลาโดยที่อาจตั้งโปรแกรมเอาไว้ให้ทำงานเอง
- ตรวจสอบทรัพย์สินต่างๆด้วยตัวเองได้

1.4 วิธีการดำเนินงาน

โครงสร้างทางการทำงานของ Security Robot ทั้งหมดนั้นจะแบ่งได้เป็น สามส่วนคือ

- รถบังคับ
- การติดต่อสื่อสารไร้สาย
- โปรแกรมควบคุมการทำงานของรถและภาพ

รูปที่ 1.1 บล็อกไดอะแกรมแสดงการควบคุมโรบอต

การทำงานของรถ

การทำงานในส่วนของรถนั้น ประกอบด้วยส่วนย่อย 2 ส่วน ด้วยกัน คือจะมีอุปกรณ์สื่อสารไร้สาย ทำหน้าที่ส่งและรับข้อมูลที่จะถูกส่งมาจากวงจรสื่อสารด้วยคลื่นวิทยุ เพื่อควบคุมการทำงานของรถ ส่วนที่สองจะเป็นกล้องวีดีโอไร้สายที่จะทำหน้าที่บันทึกภาพ และส่งไปยังตัวรับสัญญาณภาพของตัวมันเอง ซึ่งจะทำการแสดงผลภาพในส่วนการทำงานในส่วนของโปรแกรม ซึ่งจะทำให้เราเห็นภาพออกมาเป็นสตรีมมิ่งวีดีโอ

รูปแบบของรถที่เลือกนำมาใช้นี้มีความแข็งแรงทนทาน สามารถเคลื่อนที่ไปในพื้นที่ที่มีสภาพไม่ดีได้ และมีอุปกรณ์ควบคุมการทำงานระยะไกลด้วยคลื่นสัญญาณวิทยุไร้สายติดมาด้วย

การสื่อสารไร้สาย

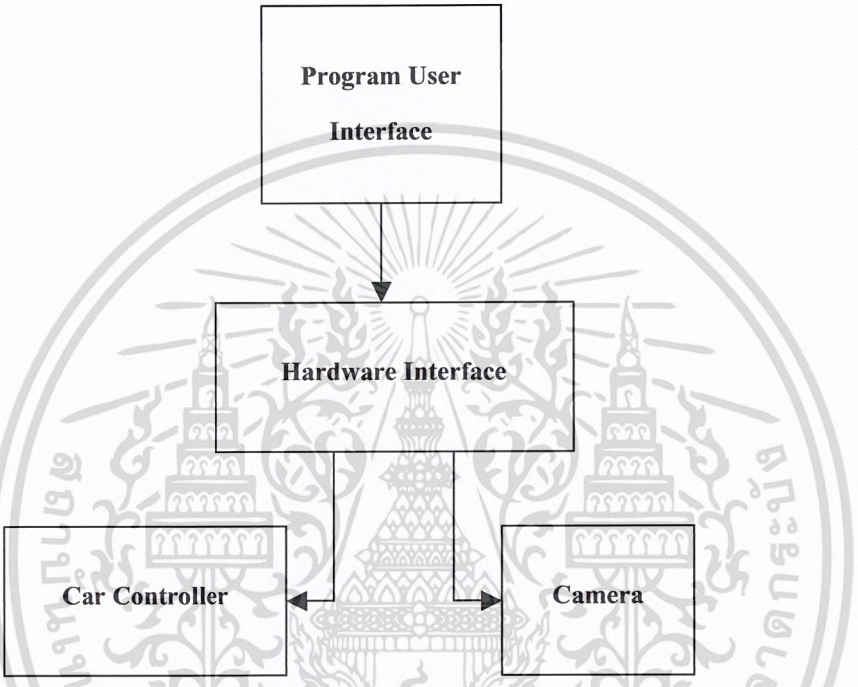
การทำงานในส่วนนี้จะแบ่งออกเป็น 2 ส่วนด้วยกัน ในแต่ละส่วนจะส่งข้อมูลโดยไม่ขึ้นต่อกัน เนื่องจากว่าอุปกรณ์การสื่อสารในแต่ละตัวนั้นมีชนิดของข้อมูลที่ส่งไม่เหมือนกัน การสื่อสารของกล้องถ่ายวิดีโอไร้สาย มีปริมาณของข้อมูลที่ส่งมากและต่อเนื่อง ทำให้จำเป็นต้องใช้ความถี่สูงในระดับกิกกะเฮิร์ซเป็ตัวกลางในการรับส่งข้อมูล การสื่อสารควบคุมการทำงานบังคับรถโดยที่จะใช้อุปกรณ์ที่มีให้อยู่แล้วกับตัวรถ ใช้บังคับรถ

อุปกรณ์ควบคุมฮาร์ดแวร์ (Hardware Controller)

ในการทำงานของส่วนนี้จะเป็นเรื่องของการใช้ไอซีและการเชื่อมต่ออุปกรณ์อินพุตและเอาต์พุตเข้าด้วยกัน เพื่อใช้ในการควบคุมการทำงานของวงจรสื่อสารไร้สายอีกทีหนึ่ง โดยการทำงานของหน่วยควบคุมศูนย์กลางนี้ จะประกอบไปด้วย การควบคุมสเต็ปปิงมอเตอร์2 ตัว ซึ่งสเต็ปปิงมอเตอร์ทั้ง 2 ตัวนี้จะไปบังคับตัวควบคุมบังคับรถไร้สายอีกทีหนึ่ง ส่วนถัดไปจะเป็นการติดต่อกับกล้องถ่ายวีดีโอไร้สาย โดยทั้งสองส่วนที่ได้กล่าวมานี้ จะทำหน้าที่เชื่อมต่อกับโปรแกรม ซึ่งโปรแกรมจะควบคุมการทำงานของหน่วยการทำงานควบคุมส่วนกลางอีกทีหนึ่ง

โปรแกรมส่วนติดต่อกับผู้ใช้

ในการติดต่อกับผู้ใช้ จะใช้โปรแกรม Visual Basic 6 ในการเขียนโปรแกรมติดต่อกับผู้ควบคุม และคำสั่งที่ได้จากการสั่งงานของผู้ควบคุมจะถูกส่งผ่าน Serial Port Com1 ไปยังวงจรแปลงระดับสัญญาณ RS-232 เป็นระดับสัญญาณ TTL และส่งต่อไปยัง MSC-51 เพื่อใช้ในการควบคุมสเต็ปปิงมอเตอร์ เพื่อนำไปบังคับทิศทางการเคลื่อนที่โรบอตอีกทีหนึ่ง

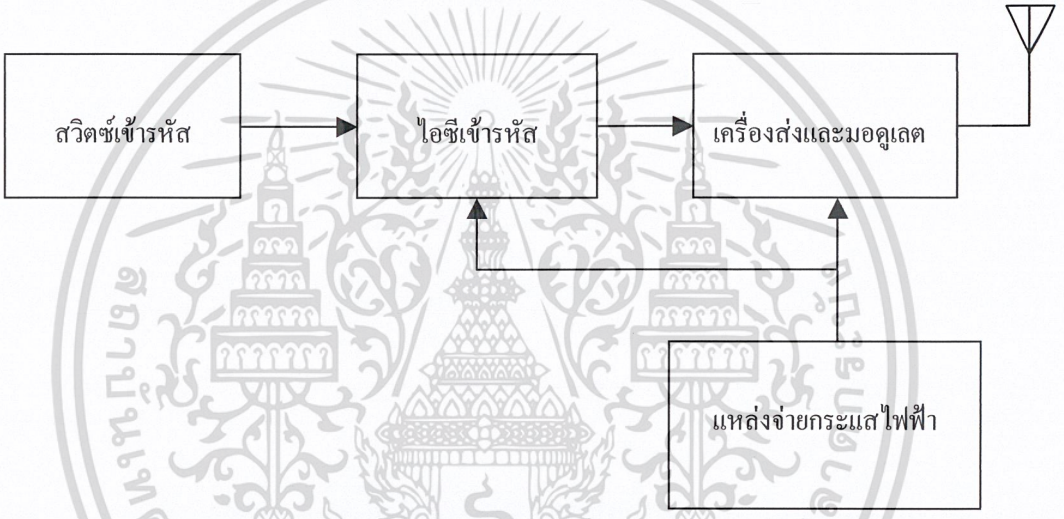


รูปที่ 1.2 บล็อกไดอะแกรมการควบคุมโรบอตโดยใช้โปรแกรมติดต่อกับผู้ใช้

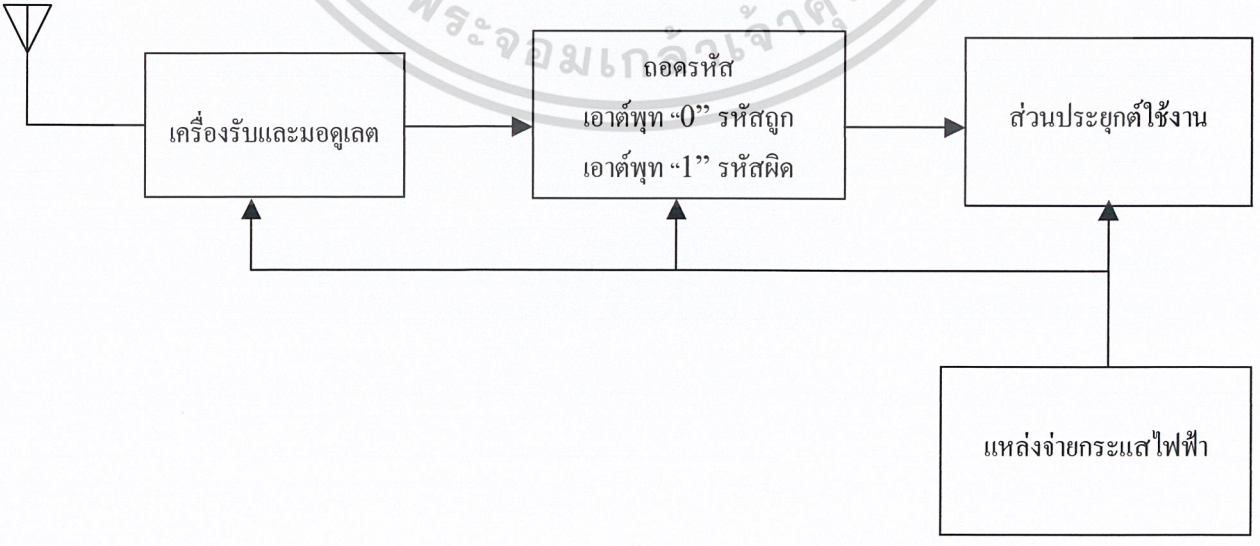
บทที่ 2
การสื่อสารไร้สายด้วยคลื่นวิทยุความถี่สูงระยะไกล

ส่วนของการสื่อสารไร้สายนี้เราสามารถแบ่งได้ด้วยกันเป็นสามส่วนใหญ่ เนื่องจากว่าในสองส่วนนั้นได้มาจากอุปกรณ์จากตัวรถวิทยุบังคับ และตัวกล้องถ่ายวีดีโอไร้สายด้วย เพราะฉะนั้นในส่วนนี้เราจะอธิบายถึงแต่เรื่องการส่งสัญญาณไร้สายระยะที่สามารถทำขึ้นมาเองได้ในเรื่องนี้

2.1 หลักการและทฤษฎีของระบบควบคุมระยะไกลแบบไร้สาย



รูปที่ 2.1 บล็อกไดอะแกรมแสดงหลักการทำงานภาคส่งของระบบคลื่นวิทยุ

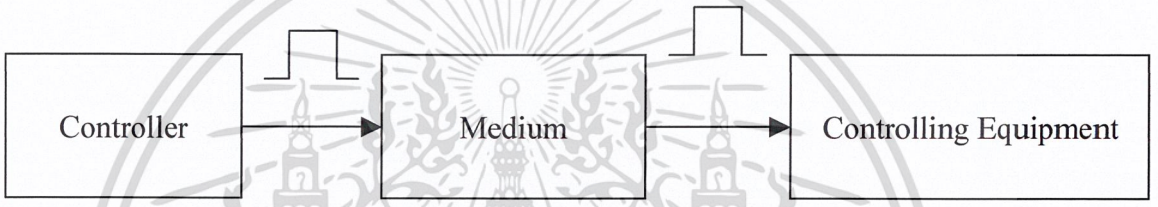


รูปที่ 2.2 บล็อกไดอะแกรมแสดงหลักการทำงานภาครับของระบบคลื่นวิทยุ

2.2 ความรู้พื้นฐานเกี่ยวกับอุปกรณ์ส่งคลื่นวิทยุ (Radio Controller)

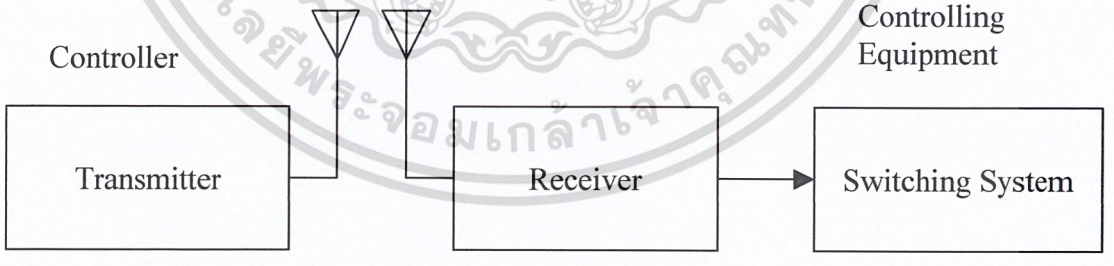
เป็นตัวควบคุมทางไฟฟ้าหรืออิเล็กทรอนิกส์ ซึ่งมีอุปกรณ์ที่ควบคุมอยู่ระยะไกลจากตัวควบคุม ระหว่างตัวควบคุมและตัวถูกควบคุมจะต้องมีตัวกลางเป็นตัวส่งผ่านสัญญาณในการควบคุม เช่น สัญญาณจากคลื่นวิทยุ คลื่นเสียงไมโครเวฟ หรืออื่นๆซึ่งตัวกลางนี้อาจเป็นอากาศ หรือ สายส่งสัญญาณก็ได้

จากรูปที่ 2.3 เป็นการแสดงถึงระบบของอุปกรณ์ส่งสัญญาณวิทยุ ซึ่งเป็นตัวผลิตสัญญาณที่ใช้ในการบังคับตัวถูกควบคุม โดยมีตัวกลาง เป็นตัวรับสัญญาณจากอุปกรณ์ส่งสัญญาณวิทยุ ไปยังวงจรที่ถูกควบคุม ในภาพ ตัวสัญญาณที่ถูกส่งมาออกมาจากอุปกรณ์ควบคุม จะเป็นในรูปของสัญญาณพัลส์ (Pulse) เมื่อสัญญาณพัลส์ เข้าไปในตัวอุปกรณ์ที่ถูกควบคุม แล้วสัญญาณพัลส์จะเป็นตัวกระตุ้นระบบสวิตช์ (Switch) ของ อุปกรณ์ควบคุม



รูปที่ 2.3 บล็อกไดอะแกรมแสดงรูปแบบการส่งสัญญาณแบบพัลส์

ในทางปฏิบัติตัวอุปกรณ์ควบคุมนี้อาจเป็นตัวผลิตสัญญาณพัลส์ ที่ตัวเครื่องส่ง (Transmitter) ตัวกลางอาจจะเป็นอากาศ และอุปกรณ์ที่ถูกควบคุมอาจเป็นเครื่องรับ (Receiver) และระบบสวิตช์ ดังรูป

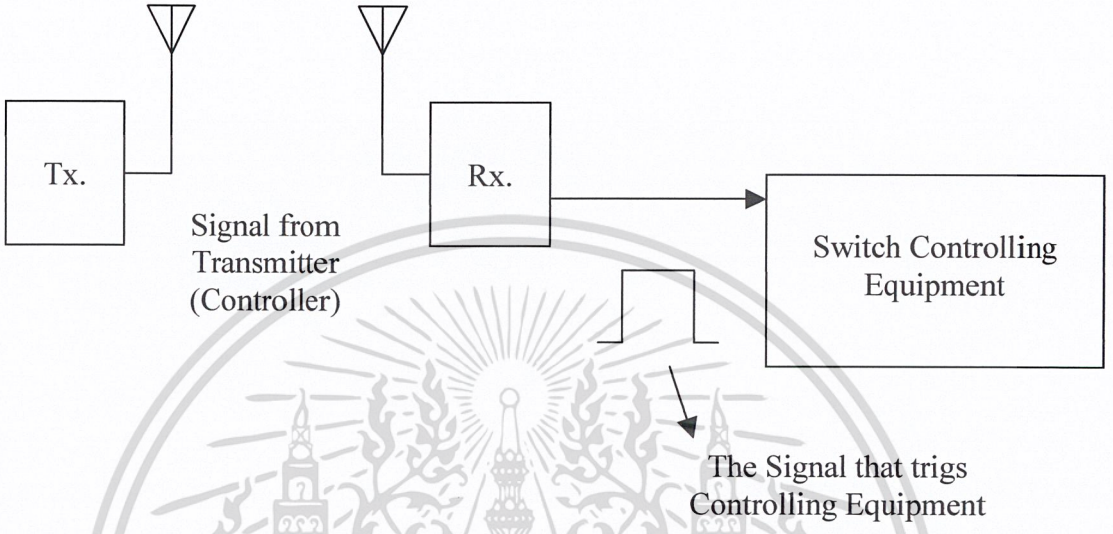


รูปที่ 2.4 แสดงหลักการทำงานของระบบคลื่นวิทยุผ่านทางตัวกลางที่เป็นอากาศ

2.3 อุปกรณ์ส่งสัญญาณ

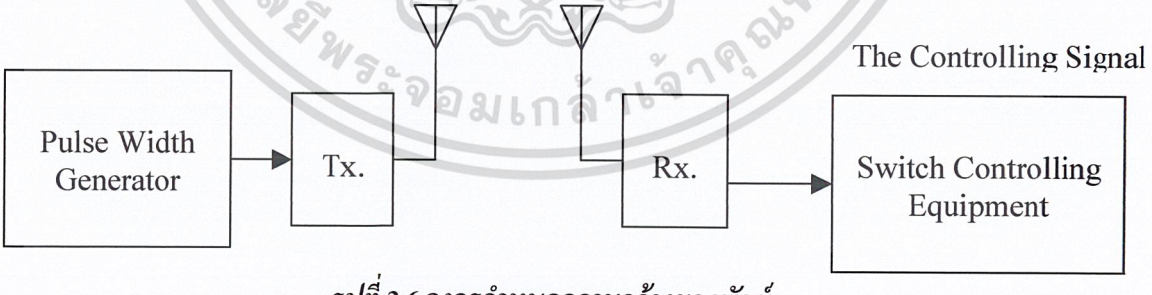
โดยอาศัยหลักการของเครื่องส่งวิทยุ เราได้ใช้ความถี่บางความถี่ที่สามารถเดินทางบนอากาศได้ เช่นความถี่ย่าน MF, RF, ไมโครเวฟ (Microwave), อินฟราเรด (Infrared) เป็นต้น เมื่อเราผลิตความถี่นี้ออกมาแล้วส่งให้มันเดินทางไปในอากาศ ดังนั้นสำหรับตัวเครื่องส่งแล้วจะต้องมีตัวที่ผลิตความถี่ที่จะส่งออก

อากาศ หรือ ตัวกำหนดสัญญาณวิทยุ และที่ตัวเครื่องรับก็เช่นเดียวกันกับเครื่องรับวิทยุคือ จะรับเอาคลื่นที่อยู่ในอากาศนั้นมาแล้วแปลความหมาย ซึ่งตัวผลิตความถี่ของเครื่องส่ง จะผลิตความถี่เท่าไรนั้น ขึ้นอยู่กับจุดประสงค์ในการใช้งานของเรา เพราะในย่านความถี่ต่างๆก็มีข้อดีและข้อเสียต่างกันออกไปในการส่งสัญญาณทางอากาศของเครื่องส่ง สำหรับอุปกรณ์ส่งสัญญาณวิทยุนั้น



รูปที่ 2.5 การใช้สัญญาณพัลส์กระตุ้นให้อุปกรณ์ควบคุมสวิตซ์ทำงาน

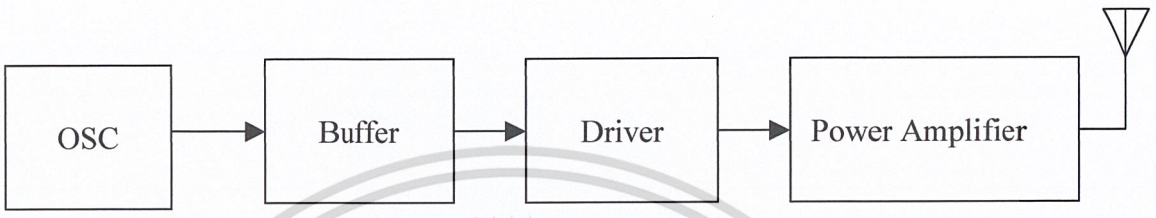
อาศัยความถี่ที่เราผลิตได้จากเครื่องส่ง เป็นเพียงคลื่นพาหะ (Carrier) เพื่อนำเอาสัญญาณพัลส์ ไปเป็นตัวกระตุ้นให้อุปกรณ์ควบคุมสวิตซ์ทำงานดังรูป



รูปที่ 2.6 วงจรกำหนดความกว้างของพัลส์

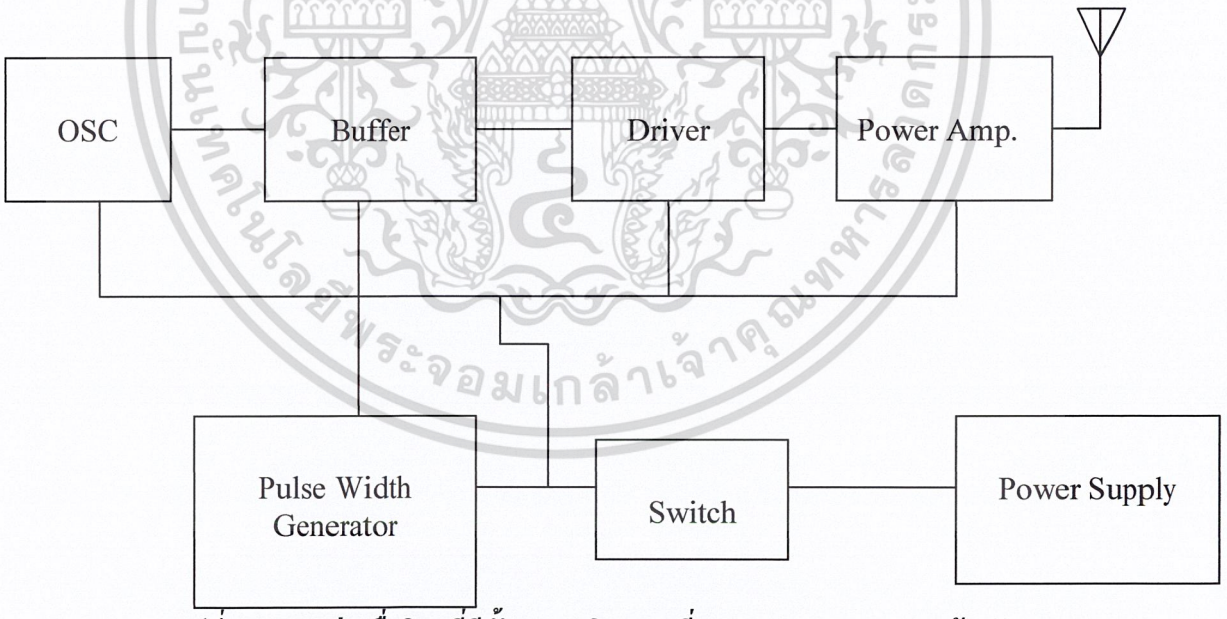
สรุปแล้วในภาคเครื่องส่งก็สามารถแบ่งลักษณะของเครื่องส่ง ออกได้เป็นสองชนิดคือ ชนิดที่มีเครื่องส่งอย่างเดียว และชนิดที่มีสัญญาณพัลส์กับเครื่องส่งประกอบกันดังรูปที่ 2.3และ 2.4 ซึ่ง Tx คือ เครื่องส่ง และ Rx คือ เครื่องรับ และ อุปกรณ์กำหนดความกว้างของสัญญาณพัลส์ (Pulse Width Generator) คือ สัญญาณพัลส์นั่นเอง สำหรับบล็อกไดอะแกรมของเครื่องส่งจะประกอบด้วยภาคความถี่

(Oscillator) ภาคที่กั้น OSC กับ บัฟเฟอร์ (Buffer), ภาคขับสัญญาณ (Driver), และภาคขยายกำลัง (Power Amplifier) ดังรูปที่ 2.7 ซึ่งเป็นตัวอย่างของเครื่องส่งอย่างง่ายๆ สำหรับวงจรที่มีการกำหนดความกว้างของสัญญาณก็เพิ่มอุปกรณ์กำหนดความกว้างของสัญญาณพัลส์เข้ามาอีก วงจรผลิตความถี่ (OSC) ผลิตความถี่ออกมา เพื่อที่จะผสมกับความกว้างสัญญาณ เราเรียกการผสมสัญญาณนั้นว่า ตัวปรับสัญญาณ (Modulator) แล้วสัญญาณจะผ่านทางบัฟเฟอร์ ภาคขับสัญญาณจนกระทั่งถึงภาคขยายกำลังเพื่อขยายสัญญาณให้มีขนาดสูงขึ้น จากนั้นก็ส่งออกอากาศดังรูป



รูปที่ 2.7 วงจรผลิตความถี่

2.4 สัญญาณจากสัญญาณรหัสกว้าง (Pulse Width)



รูปที่ 2.8 วงจรส่งคลื่นวิทยุที่มีทั้งวงจรผลิตความถี่และวงจรกำหนดความกว้างสัญญาณ

สัญญาณจากวงจรผลิตความถี่ (OSC) จะเป็นตัวกระตุ้นตัวถูกควบคุม (Controlling Equipment) ทำงาน สัญญาณจากสัญญาณรหัสกว้าง จะเป็นตัวถูกกระตุ้นให้ตัวถูกควบคุม (Controlling Equipment) ทำงาน ในทางปฏิบัติสัญญาณจากวงจรผลิตความถี่ และสัญญาณรหัสก็คือ ความถี่ทั้งคู่ แต่อาจจะมีความ

แตกต่างกันบ้าง โดยทั่วไปแล้วความถี่ของ วงจรผลิตความถี่ (OSC) จะสูงกว่า สัญญาณกว้าง และความถี่ของภาคกำหนดความกว้างของสัญญาณ จะออกมาในรูปของสัญญาณกว้าง

2.5 อุปกรณ์รับสัญญาณ (Receiver)

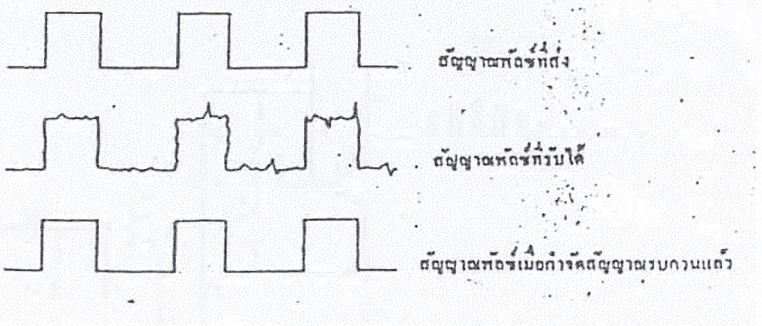
อุปกรณ์รับสัญญาณของวงจร Radio Controller จะเป็นตัวที่รับสัญญาณจากอากาศที่ถูกส่งมาโดย Transmitter ดังนั้นสำหรับภาครับจึงต้องมีความสามารถในการรับความถี่ที่ถูกส่งโดยอุปกรณ์ส่งสัญญาณเท่านั้น ทั้งนี้เพื่อป้องกันการรบกวน สัญญาณจากที่อื่นสำหรับอุปกรณ์รับสัญญาณ ส่วนประกอบใหญ่ของวงจรจะประกอบด้วย ภาครับและวงจรถอดรหัสสัญญาณ ภาครับจะรับเอาสัญญาณที่ส่งออกมาจากอุปกรณ์ส่งสัญญาณ เข้าไปในเครื่องและตัวถอดรหัสสัญญาณ จะทำการถอดเอาสัญญาณที่ได้รับออกมาเป็นพัลส์สัญญาณ เพื่อที่จะใช้งานต่อไปก่อนที่จะเอาไปใช้งานสัญญาณที่ได้ ควรจะได้รับการขยายก่อนเพื่อให้สัญญาณมีขนาดใหญ่ขึ้น (แรงมากขึ้น)

รูปที่ 2.9 วงจรแปลงสัญญาณคลื่นวิทยุ

2.6 การสื่อสารด้วยพัลส์

การสื่อสารด้วยพัลส์ได้เข้ามามีบทบาทอย่างมากในกิจการสื่อสารและโทรคมนาคม ตลอดจนการสื่อข้อมูลในรูปโทรเลข โทรพิมพ์ โทรศัพท์ การสื่อสารที่ใช้ในระบบราชการ การทหาร การพาณิชย์ การสื่อสารนี้มักจะส่งหรือสื่อสารกันด้วยสัญญาณพัลส์ ในบทนี้เราจะได้กล่าวถึงการสื่อสารที่ใช้พัลส์เป็นหลัก

ข้อดีของการสื่อสารแบบนี้เห็นได้ชัดในเรื่องความถูกต้องของข้อมูล ทั้งนี้เพราะเราส่งสัญญาณเป็น พัลส์ เมื่อมีสัญญาณอื่นรบกวนเข้ามาในระบบ การถอดรหัสออกมาเป็นพัลส์ก็ยังเหมือนเดิม ดังตัวอย่างในรูป



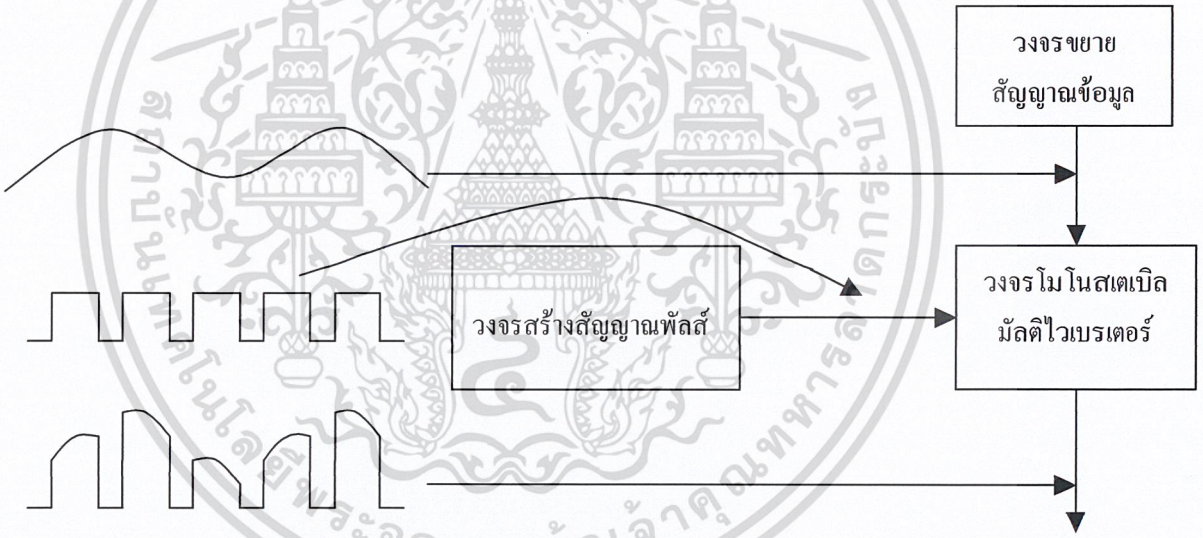
รูปที่ 2.10 สัญญาณพัลส์

นอกจานี้การสื่อสารด้วยพัลส์ยังมีข้อดีในเรื่องประสิทธิภาพของการส่งด้วย เพราะถ้าเป็นการส่งแบบวิทยุการทำงานของเอาต์พุตททรานซิสเตอร์จะอยู่ในลักษณะของสวิตซ์ซ์คัตออฟและอิ่มตัว ซึ่งทำให้กำลังงานสูญเสียในตัวเอาต์พุตทรานซิสเตอร์มีค่าน้อยกว่า

วงจรปรับสัญญาณพัลส์ (Pulse Amplitude Modulation)

การปรับสัญญาณพัลส์ คือ การสื่อสารอย่างหนึ่งที่นำคลื่นสัญญาณข้อมูลมามอดูเลตกับสัญญาณพัลส์ ซึ่งผลที่ได้คือ เอาต์พุตของสัญญาณจะประกอบด้วยสัญญาณพัลส์ที่มีแอมปลิจูดขึ้นอยู่กับคลื่นสัญญาณข้อมูลข่าวสาร โดยที่ความกว้างของสัญญาณพัลส์และความถี่ยังมีค่าคงเดิมการปรับสัญญาณพัลส์มีชื่อเรียกย่อๆว่า PAM (Pulse Amplitude Modulation)

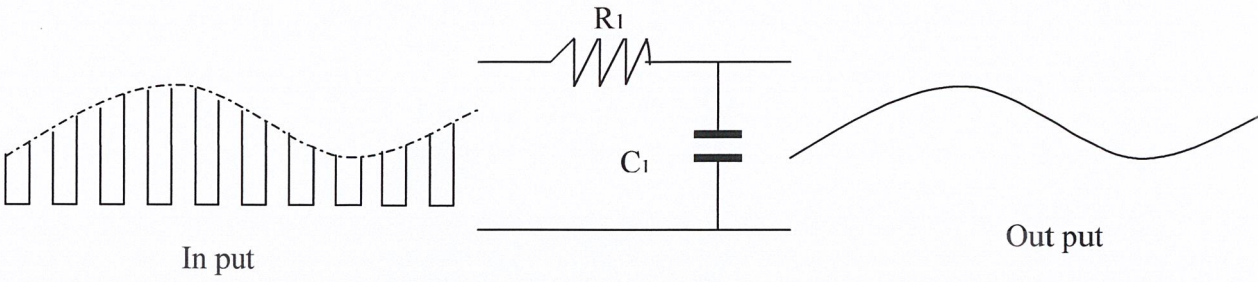
บล็อกไดอะแกรมวงจรปรับสัญญาณพัลส์ ที่แสดงในรูปที่ 2.11 ประกอบด้วยวงจรแหล่งสัญญาณที่มีความถี่ไม่สูงมากนัก เพื่อนำไปมอดูเลตกับสัญญาณพัลส์ที่ได้มาจากวงจรสัญญาณพัลส์ วงจรมอดูเลตที่ใช้ก็เป็นวงจรโมโนสเตเบิลที่ได้รับการทริกจากวงจรสัญญาณพัลส์นั่นเอง เอาพุตที่ได้รับจากวงจรโมโนสเตเบิ้ลจะมีลักษณะของแอมปลิจูดของพัลส์ไม่เท่ากันขึ้นอยู่กับสัญญาณข้อมูล



รูปที่ 2.11 วงจรปรับสัญญาณพัลส์

รูปนี้ เป็นการแสดงวงจรที่ใช้แทนบล็อกไดอะแกรมของรูปที่ 2.12 ถ้าพิจารณาให้ดีจะเห็นว่าส่วนของ Q1 และ Q2 ต่อเป็นวงจรโมโนเสตเบิ้ลมัลติไวเบรเตอร์ เอาต์พุตของโมโนสเตเบิ้ลจะขับทรานซิสเตอร์ Q3 โดยที่คอลเลคเตอร์ของ Q3 จะป้อนสัญญาณข้อมูลที่มันยกระดับมาทางไปตรงด้านบวก เมื่อ Q3 นำกระแส แรงดันตรงเอาต์พุต จะมีค่าเท่ากับแรงดันตกคร่อมตัวต้านทาน R6 ครั้นเมื่อ Q3 คัตออฟ เอาต์พุตจะได้เท่ากับแรงดันข้อมูลที่ป้อนเข้ามาทางคอลเลคเตอร์

เมื่อมีการส่งข้อมูลออกไปยังผู้รับปลายทาง ผู้รับก็ต้องทำการดีมอดูเลต การดีมอดูเลตสัญญาณ (PAM) นี้ทำได้โดยการผ่านวงจรสัญญาณความถี่ต่ำที่ประกอบด้วย RC วงจรก็จะกรองแต่สัญญาณส่วนขอบเช่นเดียวกับการตรวจจับสัญญาณ (Detect) ในการรับสัญญาณ AM



รูปที่ 2.12 วงจรกรองสัญญาณความถี่ต่ำ

2.7 วงจร Pulse Width Modulation (PWM)

เป็นวงจรปรับคลื่นสัญญาณ ทางความกว้างของพัลส์สัญญาณ ซึ่งจะแบ่งตามขนาดของสัญญาณข้อมูล นั่นคือ ถ้าข้อมูลมีแอมปลิจูด (Amplitude) ความกว้างของพัลส์ ก็จะมากขึ้นตามไปด้วย วงจรปรับความกว้างสัญญาณ (Pulse Width Modulation) แบบง่าย

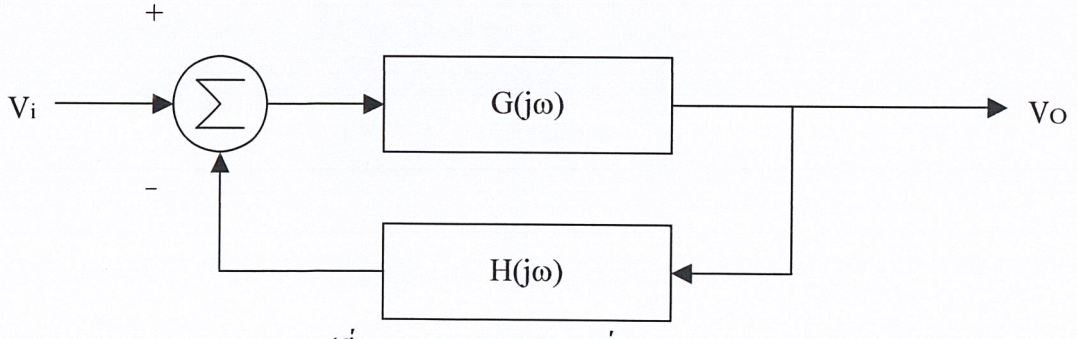
ปัจจุบันนี้เราสามารถเลือกใช้ IC ได้หลายเบอร์ด้วยกันมากมาย ซึ่งสามารถดัดแปลง ความกว้างของสัญญาณ พัลส์ ได้โดยตรงและมีประสิทธิภาพที่ดี การดัดแปลงนี้กระทำได้โดย การแปลงสัญญาณความกว้างให้เป็นการแปรค่าทางแอมปลิจูด แล้วใช้วงจรกรองความถี่ต่ำผ่าน เป็นตัวตรวจจับสัญญาณ

ปกติแล้วจะมีวงจรรวมสัญญาณ ที่ทำหน้าที่แปรสัญญาณพัลส์ ให้เป็นสัญญาณลาดถ้าความกว้างของพัลส์ มีมากแอมปลิจูด ของสัญญาณลาด ก็จะมีค่ามากตามไปด้วย เมื่อนำเอามาสัญญาณปรับความกว้าง ผ่านวงจรรวมสัญญาณ เราก็จะได้สัญญาณที่มีแอมปลิจูด เปลี่ยนแปรตามสัญญาณข้อมูลทันที

2.8 วงจรกำเนิดความถี่วิทยุ (Radio Frequency Oscilator)

หลักการสำคัญในการกำเนิดความถี่มีอยู่ สองประการ

1. เฟสชิฟต์ (Phase Shift) รอบวงของระบบที่มีค่ารวมกันเป็น 0 องศา
2. อัตราการขยายครบรอบวง (loop gain) มีค่าเป็นหนึ่งพอดี ถ้าต้องการให้ขนาดของสัญญาณที่ได้ จากการกำเนิด มีค่าคงที่



รูปที่ 2.13 วงจรกำเนิดความถี่วิทยุ

จากแผงผังวงจรในรูปจะเห็นได้ว่าส่วนของวงจรขยาย มีอัตราการขยายเปลี่ยนแปลงไปตามความถี่ G (jw) และมีอัตราการป้อนกลับเปลี่ยนแปรความถี่เป็น H (jw) เราสามารถกล่าวได้อีกนัยหนึ่งว่าในระบบที่มีการป้อนกลับแบบติดลบ ถ้าเฟสชิฟต์ของวงจรป้อนกลับมีค่าเป็น 180 องศา ที่บางความถี่วงจรจะกำเนิดที่ความถี่นั้น และสำหรับวงจรที่มีการป้อนกลับแบบบวกเฟสชิฟต์ ของการป้อนกลับต้องเป็น 0 องศา

อิมิตเตอร์ร่วม (Common Emitter) จะเห็นได้ว่าเฟสชิฟต์ ของวงจรป้อนกลับต้องมีค้า 180 องศา แต่ถ้าในกรณีที่เราใช้วงจรแบบ เบสร่วม (Common Base Emitter) แล้วต้องไม่มี เฟสชิฟต์ ระหว่างอินพุตและเอาต์พุต เพื่อให้กำเนิดสัญญาณ อย่างต่อเนื่องได้ แต่ถ้ามีเฟสชิฟต์วงจรก็จะป้อนกลับเป็น 0 องศา

2.9 การทำงานของภาคส่งสัญญาณสื่อสาร

สามารถแยกได้เป็นสองส่วนคือ ภาคเข้ารหัสโดยใช้ IC และภาคส่งซึ่งใช้ความถี่วิทยุโดยการวงจร (RF) Oscilator

2.10 การทำงานของภาครับสัญญาณ

วงจรภาครับจะรับสัญญาณ แบบ Regenerative และผ่านไปยังวงจรขยายแบบกลับ ซึ่งมีอัตราการขยายสูง แล้วต่อไปผ่านไปยังชมิตต์ทริกเกอร์ ซึ่งทำหน้าที่รับสัญญาณลูกคลื่นให้เป็นวงจร พัลส์ ที่ถูกต้องโดยผ่านตัวกลับสัญญาณ เพื่อกลับเฟสของสัญญาณ

2.11 ปัญหาการทำงานของการสื่อสารไร้สายด้วยคลื่นวิทยุ

ปัญหาในส่วนนี้นั้นเราได้พบว่าปัญหาส่วนมากจะเกิดขึ้นคล้ายๆกับตัวรถคือหาซื้ออุปกรณ์ได้ยาก และเป็นปัญหาทางด้านเทคนิคเฉพาะด้านแต่น้อยกว่ารถที่ประสบมา และปัญหาที่ไม่สามารถตอบได้เช่นต่อแล้วสัญญาณไม่ออกมา

บทที่ 3
ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ที่ บริษัทอินเทลได้พัฒนาขึ้นเพื่อการใช้งานทั่วไป ซึ่งต่อมาบริษัท แอทเมล (Atmel) ได้ซื้อลิขสิทธิ์ ไปเพื่อพัฒนาต่อ ซึ่งมีคุณลักษณะทั่วไปดังนี้

- ต้องการแหล่งจ่ายไฟ +5 โวลต์ชุดเดียว
- มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล 128กิโลไบต์
- มีไทม์เมอร์ / เคาน์เตอร์ 16 บิต 2 ชุดทำงานได้ 4 โหมด
- รับอินเตอร์รัปต์ได้ 6 แหล่ง 5 เวคเตอร์
- มีพอรต์รับส่งข้อมูลอนุกรม 2 พอรต์แบบ Full Duplex เลือกได้ 4 รูปแบบ

ภายในไมโครคอนโทรเลอร์ประกอบไปด้วย

- เอแอลยู (ALU : Arithmetic and Logic Unit) – แอคคิวมูเลเตอร์(Accumulator)
- รีจิสเตอร์ R0 – R7 จำนวน 4 กลุ่ม - แรมภายใน
- ตัวนับโปรแกรม (Program Counter : PC) - รอมภายใน
- ตัวชี้ สแต็ค (Stack Pointer : SP) - วงจร อินเตอร์รัปต์
- พอรต์อินพุต /เอาต์พุต (I/O Port) - วงจรนาฬิกา
- ไทม์เมอร์ / เคาน์เตอร์ - พอรต์อนุกรม

3.1 คำสั่งที่ใช้กับ MCS-51

คำสั่งที่ ใช้กับ MSC-51 สามารถแบ่งได้เป็น 5 กลุ่มดังนี้

- กลุ่มคำสั่งทางคณิตศาสตร์
- กลุ่มคำสั่งทางตรรกะ
- กลุ่มคำสั่งในการเคลื่อนย้ายข้อมูล
- กลุ่มคำสั่งการประมวลแบบบูลีน
- กลุ่มคำสั่งควบคุมลำดับการทำงาน

รายละเอียดคำสั่งต่างๆในแต่ละกลุ่มคำสั่งมีดังต่อไปนี้

คำสั่งทางคณิตศาสตร์

รูปแบบคำสั่ง	การทำงาน
ADD A,<BYTE>	A = A + <BYTE>
ADDC A,<BYTE>	A = A + <BYTE> + C
SUBB A,<BYTE>	A = A - <BYTE> -C
INC A	A = A + 1
INC <BYTE>	<BYTE> = <BYTE> + 1
INC DPTR	DPTR = DPTR + 1
DEC A	A = A – 1
DEC <BYTE>	<BYTE> = <BYTE> - 1
MUL AB	B:A = B x A
DIV AB	A = Int[A/B]
	B = Mod[A/B]
DA A	Decimal Adjust

ตารางที่ 3.1 คำสั่งทางคณิตศาสตร์

คำสั่งทางตรรกะ

รูปแบบคำสั่ง	การทำงาน
ANL A,<BYTE>	A = A AND <BYTE>
ANL <BYTE>,A	<BYTE> = <BYTE> AND A
ANL <BYTE>,#DATA	<BYTE> = <BYTE> AND #DATA
ORL A, <BYTE>	A = A OR <BYTE>
ORL <BYTE>,A	<BYTE> = <BYTE> OR A
ORL <BYTE>,#DATA	<BYTE> = <BYTE> OR #DATA
XRL A, <BYTE>	A = A XOR <BYTE>
ORL <BYTE>,A	<BYTE> = <BYTE> XOR A
ORL<BYTE>,#DATA	<BYTE> =<BYTE>XOR#DATA
CLRA	A=00H
CPLA	A=NOTA
RLA	เลื่อนค่าใน ACCไปทางซ้าย 1บิต
RLCA	เลื่อนค่าใน ACCไปทางซ้าย 1บิตผ่าน Carry
RRA	เลื่อนค่าใน ACCไปทางขวา 1บิต
RRCA	เลื่อนค่าใน ACCไปทางขวา 1บิตผ่าน Carry
SWAPA	สลับ 4บิตบนและล่างใน ACC

ตารางที่ 3.2 คำสั่งทางตรรกะ

คำสั่งในการเคลื่อนย้ายข้อมูล

รูปแบบคำสั่ง	การทำงาน
MOVA,<SRC>	A=<SRC>
MOV<DEST>,A	<DEST>=A
MOV<DEST>,<SCR>	<DEST>=<SRC>
MOVDPTR,#DATA(16)	DPTR=16บิตค่าคงที่
PUSH<SRC>	PUSH<SRC>เข้าสแต็ค
POP<DEST>	POPสแต็คเข้า <DEST>
XCHA,<DEST>	แลกเปลี่ยนค่าระหว่าง Aและ <DEST>
XCHDA,@Ri	แลกเปลี่ยนค่า 4บิตล่างระหว่าง Aและค่าใน ตำแหน่งที่ระบุโดย Ri
MOVXA,@Ri (อ้างตำแหน่ง 8บิต)	นำค่าในหน่วยความจำสำหรับเก็บข้อมูลภายนอกที่ระบุโดย Riมาไว้ที่ A
MOVX@Ri,A (อ้างตำแหน่ง 8บิต)	นำ ค่าใน Aออกไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกที่ระบุโดย Ri
MOVXA,@DPTR (อ้างตำแหน่ง 16บิต)	นำค่าในหน่วยความจำสำหรับเก็บข้อมูลภายนอกที่ระบุโดย DPTR มาไว้ที่ A
MOVX@DPTR,A	นำ ค่าใน Aออกไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกที่ระบุโดย DPTR
MOVCA,@A+DPTR	อ่านค่าจากหน่วยความจำโปรแกรมตำแหน่งที่ระบุโดย (A+DPTR)เข้า A
MOVCA,@A+PC	อ่านค่าจากหน่วยความจำโปรแกรมตำแหน่งที่ระบุโดย (A+ PC+1)เข้า A

ตารางที่ 3.3 คำสั่งทางการเคลื่อนย้ายข้อมูล

คำสั่งการประมวลผลแบบบูลีน

รูปแบบคำสั่ง	การทำงาน
SETB C	CY = 1
SETB <BIT>	<BIT> = 1
CLR C	CY = 0
CLR <BIT>	<BIT> = 0

CPL C	CY = CY
CPL <BIT>	<BIT> = ><BIT
MOV C,<BIT>	ย้ายข้อมูลในตำแหน่ง BIT เข้า CY
MOV <BIT>,C	ย้ายข้อมูลในตำแหน่ง CY เข้า <BIT>
ANL C.<BIT>	ข้อมูลในตำแหน่ง CY AND กับ <BIT>
ANL C,/<BIT>	ข้อมูลในตำแหน่ง CY AND กับ not <BIT>
ORL C,<BIT>	ข้อมูลในตำแหน่ง CY OR กับ <BIT>
ORL C,/<BIT>	ข้อมูลในตำแหน่ง CY OR กับ not <BIT>
JC REL	จะกระโดดถ้า CY = 1
JNC REL	จะกระโดดถ้า CY = 0
JB <BIT>,REL	จะกระโดดถ้าค่าในตำแหน่ง <BIT> = 1
JNB <BIT>,REL	จะกระโดดถ้าค่าในตำแหน่ง <BIT> = 0
JBC <BIT>,REL	จะกระโดดถ้าค่าในบิต = 1 แล้วเคลียร์บิตนั้นเป็น 0

ตารางที่ 3.4 คำสั่งการประมวลผลแบบบูลีน

คำสั่งควบคุมลำดับการทำงาน

รูปแบบคำสั่ง	**การทำงาน**
คำสั่งควบคุมลำดับการทำงานแบบมีบิตเงื่อนไข	
JZ REL	กระโดดถ้า A = 0
JNZ REL	กระโดดถ้า A /= 0
DJNZ <BYTE>,REL	ลดค่าใน <BYTE> ลงหนึ่ง ถ้าไม่เท่ากับศูนย์ให้ กระโดดไปยังตำแหน่ง (PC + 2 + REL)
CJNE A,<BYTE>,REL	เปรียบเทียบค่าใน A กับ <BYTE> ถ้าไม่เท่ากันให้ กระโดดไปยังตำแหน่ง (PC + 3 + REL) ถ้า A น้อยกว่า <BYTE> จะได้ CY = 0
CJNE <BYTE>,#DATA,REL	เปรียบเทียบค่าใน <BYTE> กับ #DATA ถ้าไม่เท่า กันให้กระโดดไปยังตำแหน่ง (PC + 3 + REL) ถ้า <BYTE> น้อยกว่า #DATA จะได้ CY = 0
คำสั่งควบคุมลำดับการทำงานแบบไม่มีบิตเงื่อนไข	
JMP ADDR	กระโดดไปยังตำแหน่ง ADDR มี3 รูปแบบ
- AJMP ADDR 11	กระโดดไปยังตำแหน่งในช่วง 2 11 ตำแหน่ง
- LJMP ADDR 16	กระโดดไปยังตำแหน่งในช่วง 2 16 ตำแหน่ง
- SJMP REL	กระโดดไปยังตำแหน่งในช่วง (-128 ถึง +128)

JMP @A + DPTR	กระโดดไปยังตำแหน่งที่ ระบุ โดยค่าใน (A + DPTR)
CALL ADDR	เรียกโปรแกรมย่อยที่ตำแหน่ง ADDR มี2 รูปแบบ
- ACALL ADDR 11	เรียกโปรแกรมย่อยในช่วง 2 11 ตำแหน่ง
- LCALL ADDR 16	เรียกโปรแกรมย่อยในช่วง 2 16 ตำแหน่ง
RET	กระโดดกลับจากโปรแกรมย่อย
RETI	กระโดดกลับจากโปรแกรมย่อยของอินเตอร์รัปต์
NOP	ไม่มีการทำงานใดๆ

ตารางที่ 3.5 คำสั่งควบคุมลำดับการทำงาน

3.2 วิธีการเข้าถึงข้อมูลสำหรับ MCS-51

ในการเขียนโปรแกรมใดๆก็ ตาม สิ่งหนึ่งที่จำเป็นต้องให้ความสนใจก็คือวิธีการเข้าถึงข้อมูล (Addressing Mode) ซึ่งวิธีการเข้าถึงข้อมูลสำหรับ MCS-51 มีดังนี้

- การเข้าถึงข้อมูลทางตรง (Direct Addressing) ทำได้กับ แรม ภายในและเอสเอฟอาร์ (SFR) เท่านั้นในลักษณะตำแหน่ง 8 บิต เช่น MOV A,7FH
- การเข้าถึงข้อมูลทางอ้อม (Indirect Addressing) นี้ จะใช้รีจิสเตอร์ R0 และ R1 ระบุตำแหน่งที่ต้องการในการเข้าถึงข้อมูลในแรม (RAM) เช่น MOV A,@R1
- ค่าคงที่ (Immediate Constant) สำหรับค่าคงที่ ใดๆ จะใช้เครื่องหมาย # นำหน้าค่าคงที่ และตามหลังรหัสคำสั่ง (Opcode) เช่น MOV A,#20H มีรหัส 7420H โดย 74 คือ รหัสคำสั่งสำหรับการ MOV และ 20 คือค่าคงที่ที่ต้องการ MOV
- รีจิสเตอร์คำสั่ง (Register Instruction) จะใช้รีจิสเตอร์ R0 ถึง R7 ในรีจิสเตอร์แบงค์ในการทำงาน ในรหัสคำสั่งของคำสั่งนี้ จะใช้ 3 บิตในการเปลี่ยนแปลงชื่อรีจิสเตอร์ เช่น INC R1 มีรหัสคำสั่งเป็น 09H หรือ 00001001B โดย 1 ในบิตที่ 3 แสดงการ INC และ 3 บิตล่างบอกถึงรีจิสเตอร์ตัวที่ 1
- รีจิสเตอร์คำสั่งเฉพาะ (Register-Specific Instruction) ชุดรหัสคำสั่งจะระบุชี้ชัดไปยังรีจิสเตอร์ตัวนั้นๆโดยเฉพาะ เช่น A และ DPTR ซึ่งจะพบว่าในชุ ดรหัสคำสั่งจะไม่มีไบต์บอกตำแหน่งเลย เช่น MOVC A,@A + DPTR จะมีรหัสคำสั่ง 93H หรือ 10010011B ซึ่งเป็นคำสั่งไบต์เดียว โดยไม่มีไบต์ระบุตำแหน่งเลย
- อินเดกซ์แอดเดรสซิ่ง (Indexed Addressing)ใน รอมเท่านั้นที่สามารถเข้าถึงข้อมูลในโหมดนี้ได้ และพบว่าสามารถอ่านข้อมูลได้อย่างเดียว มีจุดประสงค์ในการอ่านข้อมูลในตาราง ในรอมซึ่งจะระบุตำแหน่งโดยรีจิสเตอร์ 16 บิต โดยใช้ DPTR หรือ PC โดยจะใช้ค่าในรีจิสเตอร์ A เลื่อนระยะเข้าหาหรือห่างออกไป เช่น MOVC A,@A + PC

3.3 การอินเตอร์รัปต์

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การอินเตอร์รัปต์ คือ การขัดจังหวะโปรแกรมชั่วขณะเพื่อมาทำโปรแกรมบริการอินเตอร์รัปต์ (Interrupt Service Routine : ISR) การตรวจสอบสัญญาณการร้องขออินเตอร์รัปต์จะตรวจทุกๆแมชชีนไซเคิล (Machine Cycle) เมื่อพบแล้วในช่วงแมชชีนไซเคิลที่ 2 จะเป็นการตรวจสอบว่าเป็นของอุปกรณ์ใด แมชชีนไซเคิลที่ 3 จะกระโดดไปทำบริการอินเตอร์รัปต์ ดังแสดงค่าเวคเตอร์ดังตาราง อินเตอร์รัปต์ของ MCS-51 ได้มาจาก 6 แหล่งแต่มีเพียง 5 เวคเตอร์

ลำดับ	ชื่อสัญญาณอินเตอร์รัปต์	ตำแหน่งเวคเตอร์	Priority
1	IE0 (จากขา INT0)	0003H	สูงสุด
2	TF0 (ไทม์เมอร์ / เคาน์เตอร์ 0)	000BH	|
3	IE1 (จากขา INT1)	0013H	|
4	TF1 (ไทม์เมอร์ / เคาน์เตอร์ 1)	001BH	|
5	TI + RI (จากพอรต์อนุกรม)	0023H	V
6	TF2 + EXF2 (ไทม์เมอร์ / เคาน์เตอร์2)	002BH	ต่ำสุด

ตารางที่ 3.6 แสดงชื่อสัญญาณอินเตอร์รัปต์

3.4 อิทเทอร์รัพต์อีนาเบิลรีจิสเตอร์ (Interrupt Enable Register : IE Register)

ใช้ควบคุมอินเตอร์รัปต์ 6 แหล่ง 5 เวคเตอร์ คือจาก (TI,RI), TF0, TF1, IE0, IE1 เราสามารถสั่งห้ามหรือไม่ให้เกิดการอินเตอร์รัปต์ได้จากการตั้งค่ารีจิสเตอร์ชุดนี้ รายละเอียดมีดังนี้

EA	X	ET2	ES	ET	EX1	ET0	EX0

รูปที่ 3.1 แสดงโครงสร้าง IE Register

บิต	ชื่อบิต	การทำงาน
IE.7	$\overline{EA}$	= 1 ยอมให้เลือกการทำอินเตอร์รัปต์จากแหล่งต่างๆได้ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์จากแหล่งใดๆทั้งสิ้น
IE.6	X	ไม่ใช้งาน
IE5	ET2	= 1 ยอมให้ทำอินเตอร์รัปต์เมื่อ TF2 เกิดโอเวอร์โฟลว์ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์เมื่อ TF2 เกิดโอเวอร์โฟลว์
IE4	ES	= 1 ยอมให้ทำอินเตอร์รัปต์จากพอรต์สื่อสารอนุกรมได้ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์จากพอรต์สื่อสารอนุกรมได้
IE3	ET1	= 1 ยอมให้ทำอินเตอร์รัปต์เมื่อ TF1 เกิด โอเวอร์โฟลว์ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์เมื่อ TF1 เกิด โอเวอร์โฟลว์
IE2	EX1	= 1 ยอมให้ทำอินเตอร์รัปต์จากสัญญาณภายนอกหมายเลข 1

46186

IE1	ET0	= 0 ไม่ยอมให้ทำอินเตอร์รัปต์จากสัญญาณภายนอกหมายเลข 1 = 1 ยอมให้ทำอินเตอร์รัปต์เมื่อ TF0 เกิด โอเวอร์โฟลว์ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์เมื่อ TF0 เกิด โอเวอร์โฟลว์
IE.0	EX0	= 1 ยอมให้ทำอินเตอร์รัปต์จากสัญญาณภายนอกได้ (ขา INT0) = 0 ไม่ยอมให้ทำอินเตอร์รัปต์จากสัญญาณภายนอก (ขา INT0)

ตารางแสดงที่ 3.7 รายละเอียดการทำงานของ IE Register

จากโครงสร้างดังตาราง

- ถ้าต้องการ Disable อินเตอร์รัปต์ทั้งหมดจะทำได้โดยใช้คำสั่ง CLR EA เพียงคำสั่งเดียว
- ถ้าต้องการ Enable Timer0 และTimer1 ทำได้โดยใช้คำสั่ง

```
SETB EA ;Enable Interrupt
SETB ET0 ;Enable ET0
SETB ET1 ;Enable ET1
```

การอินเตอร์รัปต์ภายใน MCS-51 ได้จาก Timer0, Timer1 โดยตรวจสอบที่ TF0 และ TF1 และอินเตอร์รัปต์จากพอรต์อนุกรม โดยตรวจสอบที่ TI และ RI ส่วนขา INT0 และ INT1 เมื่อทำอินเตอร์รัปต์จะเป็นขาอินพุ ตของอินเตอร์รัปต์ภายนอก เมื่อไม่ทำอินเตอร์รัปต์จะใช้เริ่ม ไทม์เมอร์ / เคาน์เตอร์ หรือที่เราเรียกว่าการเริ่มการทำงานของฮาร์ดแวร์ (Hardware Start) การขัดจังหวะของการอินเตอร์รัปต์เราสามารถกำหนดลำดับความสำคัญได้จาก อินเตอร์รัปต์ไพรออริตีรีจิสเตอร์ (Interrupt Priority Register : IP Register) ดังรูป 3.2

PCT	X	PT2	PS	PT1	PX1	PT0	PX0

รูปที่ 3.2 แสดงโครงสร้าง IP Register

บิต	ชื่อบิต	การทำงาน
IP.7	PCT*	= 1 ยอมให้มีการจัดลำดับความสำคัญของการอินเตอร์รัปต์ (Priority) = 0 ไม่ยอมให้มีการจัดลำดับความสำคัญของการอินเตอร์รัปต์ * มีในชิปเบอร์ 83C154/83C154D เท่านั้น
IP.6	X	ไม่ใช้งาน
IP.5	PT2	= 1 ไทม์เมอร์ 2 มีลำดับความสำคัญสูงสุด = 0 ไทม์เมอร์ 2 มีลำดับความสำคัญต่ำสุด
IP.4	PS	= 1 ยอมให้ทำอินเตอร์รัปต์จากพอรต์สื่อสารอนุกรมได้ = 0 ไม่ยอมให้ทำอินเตอร์รัปต์จากพอรต์สื่อสารอนุกรม

IP.3	PT1	= 1 ไทม์เมอร์ 1 มีลำดับความสำคัญสูงสุด = 0 ไทม์เมอร์ 1 มีลำดับความสำคัญต่ำสุด
IP.2	PX1	= 1 อินเตอร์ภายนอกชนิด 1 มีลำดับความสำคัญสูงสุด = 0 อินเตอร์ภายนอกชนิด 1 มีลำดับความสำคัญต่ำสุด
IP.1	PT0	= 1 ไทม์เมอร์ 0 มีลำดับความสำคัญสูงสุด = 0 ไทม์เมอร์ 0 มีลำดับความสำคัญต่ำสุด
IP.0	PX0	= 1 อินเตอร์ภายนอกชนิด 0 มีลำดับความสำคัญสูงสุด = 0 อินเตอร์ภายนอกชนิด 0 มีลำดับความสำคัญต่ำสุด

ตารางแสดงที่ 3.8 รายละเอียดการทำงานของ IP Register

ถ้าทุกบิตถูกตั้งค่าเป็น 1 หมด จะถือว่ามีลำดับความสำคัญสูงสุดเท่ากัน เมื่อสัญญาณมาพร้อมกัน จะต้องทำการตัดสินตามตารางที่ 1 สำหรับการเลือกลักษณะของการทริค (Interrupt Type Control) ไม่ได้อยู่ในรีจิสเตอร์ที่ ใช้ควบคุมการอินเตอร์รัปต์ แต่ไปอยู่ในรีจิสเตอร์ TCON แทน

3.5 ไทม์เมอร์/เคาน์เตอร์ คอนโทรลรีจิสเตอร์ (TCON)

TCON อยู่ใน เอสเอฟอาร์ ตำแหน่งที่ 088H ใช้ทำงานหลายหน้าที่คือ

- TFx ใช้เป็นที่เก็บแฟลก ของ ไทม์เมอร์ 0,1
- TRx ใช้เริ่มไทม์เมอร์ / เคาน์เตอร์0,1
- IEx ใช้เป็นที่เก็บแฟลกของสัญญาณอินเตอร์รัปต์
- ITx ใช้เลือกลักษณะของการทริกที่ ขา INTx

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 3.3 แสดงโครงสร้าง TCON Register

บิต	ชื่อบิต	การทำงาน
TCON.7	TF1	Timer Overflow Flag แสดงการเกิดโอเวอร์โฟลว์ ของไทม์เมอร์ / เคาน์เตอร์1 จะเป็น 1 เมื่อเกิด โอเวอร์โฟลว์ และจะถูกเคลียร์เมื่อซีพียู ย้ายการทำงานไปที่ โปรแกรมบริการอินเตอร์รัปต์
TCON.6	TR1	บิตควบคุมการนับของไทเมอร์ 1 ควบคุมจากโปรแกรม ถ้าเป็น 1 ไทม์เมอร์ / เคาน์เตอร์ 1 เริ่มทำงานต่อ ถ้าเป็น 0 ไทม์เมอร์ / เคาน์เตอร์ 1 หยุ ดทำงาน
TCON.5	TF0	Timer Overflow Flag แสดงการเกิดโอเวอร์โฟลว์ ของไทม์เมอร์ / เคาน์เตอร์ 0 จะเป็น 1 เมื่อเกิดโอเวอร์โฟลว์ และจะถูกเคลียร์เมื่อซีพียูย้ายการทำงานไปที่ โปรแกรมบริการอินเตอร์รัปต์
TCON.4	TR0	บิตควบคุมการนับของไทม์เมอร์ 0 ควบคุมจากโปรแกรม

		ถ้าเป็น 1 ไทม์เมอร์ / เคาน์เตอร์ 0 เริ่มทำงานต่อ ถ้าเป็น 0 ไทม์เมอร์ / เคาน์เตอร์ 0 หยุดทำงาน
TCON.3	IE1	บิตแสดงสัญญาณอินเตอร์รัปต์ภายนอกหมายเลข 1 เมื่อมีสัญญาณอินเตอร์รัปต์เข้ามาที่ ขา INT1 และถูกเคลียร์เองโดยคำสั่ง RETI ที่อยู่ในโปรแกรมส่วนบริการอินเตอร์รัปต์
TCON.2	IT1	บิตเลือกประเภทการตรวจสอบสัญญาณอินเตอร์รัปต์ที่ ขา INT1 ถ้าเป็น 1 จะตรวจสอบการเปลี่ยนระดับสัญญาณจาก 1 เป็น 0 ที่ INT1 ถ้าเป็น 0 จะตรวจสอบระดับ 0 ของสัญญาณที่ INT1
TCON.1	IE0	บิตแสดงสัญญาณอินเตอร์รัปต์ภายนอกหมายเลข 0 เมื่อมีสัญญาณอินเตอร์รัปต์เข้ามาที่ ขา INT0 และถูกเคลียร์เองโดยคำสั่ง RETI ที่อยู่ในโปรแกรมส่วนบริการอินเตอร์รัปต์
TCON.0	IT0	บิตเลือกประเภทการตรวจสอบสัญญาณอินเตอร์รัปต์ที่ ขา INT0 ถ้าเป็น 1 จะตรวจสอบการเปลี่ยนระดับสัญญาณจาก 1 เป็น 0 ที่ INT0ถ้าเป็น 0 จะตรวจสอบระดับ 0 ของสัญญาณที่ ขา INT0

ตารางแสดงที่ 3.9 รายละเอียดการทำงานของ TCON รีจิสเตอร์

3.6 ไทม์เมอร์ / เคาน์เตอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีรีจิสเตอร์พิเศษที่ สามารถเลือกใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์อย่างใดอย่างหนึ่งโดยมีอยู่ด้วยกัน 2 ตัวคือ ไทม์เมอร์ / เคาน์เตอร์ 0 และ ไทม์เมอร์ / เคาน์เตอร์ 1 โดย ไทม์เมอร์ / เคาน์เตอร์ 0 ประกอบด้วย TH0, TL0 ไทม์เมอร์ / เคาน์เตอร์ 1 ประกอบด้วย TH1, TL1 ซึ่งมีคุณสมบัติเป็นเคาน์เตอร์รีจิสเตอร์แบบนับขึ้น (Up Counter Register) ซึ่งเราสามารถเลือกให้มีการทำงานเป็นไทเมอร์หรือเคาน์เตอร์ได้โดยเลือกที่บิต C/ T ในรีจิสเตอร์เฉพาะ TMOD (อยู่ใน เอสเอฟอาร์ตำแหน่งที่ 89H) ดังรูป

GATE	C/T	M1	M0	GATE	C/T	M1	M0

TIMER 1 **TIMER 0**

รูปที่ 3.4 แสดงโครงสร้าง TMOD Register

บิต	ชื่อบิต	การทำงาน
TMOD.7	GATE	เมื่อ TR1(ใน TCON)ถูกเซตเป็น 1และ GATE=1ไทม์เมอร์ / เคาน์เตอร์1จะทำงานเมื่อขา INT1เป็น ไฮ (High) (ควบคุมโดยฮาร์ดแวร์)เมื่อ GATE=0ไทม์เมอร์ /เคาน์เตอร์จะทำงานเมื่อ TR1=1 (ควบคุมโดยซอฟต์แวร์)
TMOD.6	C/ T	บิตเลือกการทำงานของไทม์เมอร์ เคาน์เตอร์ โดยเลือกดังนี้

		ถ้า C/ T =0เป็นการเลือกโหมดไทม์เมอร์ 1 ถ้า C/ T =1เป็นการเลือกโหมดเคาน์เตอร์ 1
TMOD.5 TMOD.4	M1 M0	เลือกโหมดการทำงานได้ 4โหมด
TMOD.3	GATE	เมื่อ TR0(ใน TCON)ถูกเซตเป็น 1และ GATE=1ไทม์เมอร์ / เคาน์เตอร์0จะทำงานเมื่อขา INT0เป็น ไฮ(ควบคุมโดยฮาร์ดแวร์) เมื่อ GATE=0ไทม์เมอร์ /เคาน์เตอร์จะทำงานเมื่อ TR0=1(ควบคุมโดยซอฟต์แวร์)
TMOD.2	C/ T	บิตเลือกการทำงานของไทม์เมอร์เคาน์เตอร์ โดยเลือกดังนี้ ถ้า C/ T =0เป็นการเลือกโหมดไทม์เมอร์ 0 ถ้า C/ T =1เป็นการเลือกโหมดเคาน์เตอร์ 0
TMOD.1 TMOD.0	M1 M0	เลือกโหมดการทำงานได้ 4โหมด

ตารางแสดงที่ 3.10 รายละเอียดการทำงานของ TMODRegister

M1	M0	โหมด	การทำงาน
0	0	0	13บิต ไทม์เมอร์หรือเคาน์เตอร์
0	1	1	16บิต ไทม์เมอร์หรือเคาน์เตอร์
1	0	2	2 8บิต ไทม์เมอร์หรือเคาน์เตอร์แบบโหลดซ้ำอัตโนมัติ
1	1	3	3 8 บิต ไทม์เมอร์หรือเคาน์เตอร์โดยใช้ TL0

ตารางแสดงที่ 3.11 รายละเอียดของโหมดไทม์เมอร์ /เคาน์เตอร์

โหมดไทม์เมอร์ รีจิสเตอร์เคาน์เตอร์แบบนับขึ้น THx, TLx จะถูกเพิ่มค่าทุกๆ 1 แมชชีนไซเคิล (12 คาบเวลาของสัญญาณนาฬิกาของ ซีพียู ที่ได้จาก Crystal) ในโหมดนี้ไม่ต้องป้อนสัญญาณจากภายนอกเข้ามาแต่จะใช้สัญญาณนาฬิกาของซีพียู (CPU) แทน วิธีเลือกโหมดนี้ต้องโปรแกรมให้บิต C/ T = 0 ส่วนการสั่งให้รีจิสเตอร์เคาน์เตอร์แบบนับขึ้นเริ่มทำงานจะควบคุมที่ TRx, GATE, โดย x คือ 0,1 เมื่อนับจนเกิดโอเวอร์โฟลว์ แล้วจะเซตบิต TFx ให้เป็น 1 และจะอินเตอร์รัปต์ได้ก็ต่อเมื่อเราเซตบิต EA และบิต Etx และรีจิสเตอร์ IE ไว้ก่อนแล้ว เมื่อเกิดการอินเตอร์รัปต์ โปรแกรมจะกระโดดมาที่ตำแหน่ง 000BH เมื่อ TF0โอเวอร์โฟลว์ และจะกระโดดมาที่ตำแหน่งเมื่อ TF1 โอเวอร์โฟลว์

การทำงานของไทม์เมอร์แบ่งได้เป็น 4 โหมดดังนี้

- **โหมด 0** (ไทม์เมอร์ 13 บิต) จะนับได้สูงสุดเท่ากับ 2 ยกกำลัง 13
- **โหมด 1** (ไทม์เมอร์ 16 บิต) จะเหมือนโหมด 0 ต่างกันที่รีจิสเตอร์เป็นแบบ 16 บิต
- **โหมด 2** (8 บิต โหลดซ้ำอัตโนมัติ) โหมดนี้ ใช้รีจิสเตอร์ TL1 และ TL0 เป็นรีจิสเตอร์แบบนับขึ้น ส่วนค่าที่ จะโหลดเข้ามาจะต้องเก็บไว้ก่อนใน TH0 และ TH1 โดยค่านี้ จะถูกโหลดเข้า TL1 และ TL0 โดยอัตโนมัติเมื่อเกิด โอเวอร์โฟลว์ (นับได้สูงสุด 256 แมชชีนไซเคิล)
- **โหมด 3** (เคาน์เตอร์และไทม์เมอร์ 8 บิต) โหมดนี้ จะใช้ TL0 ทำงานได้ทั้งโหมดเคาน์เตอร์และไทม์เมอร์ เมื่อเกิดโอเวอร์โฟลว์ จะเซตที่ TF0 ส่วน TH0 จะใช้ทำงานโหมดไทม์เมอร์ เมื่อเกิดอเวอร์โฟลว์ จะเซตที่ TF1

เราสามารถใช้ไทม์เมอร์สร้างสัญญาณนาฬิกาที่มีความถี่ ตามที่ เราต้องการได้โดยใช้ไทม์เมอร์โหมด 2 แล้วเลือกค่าโหลดเริ่มต้นที่ เหมาะสม

โหมดเคาน์เตอร์ รีจิสเตอร์เคาน์เตอร์แบบนับขึ้น THx, TLx จะถูกเพิ่มค่าที ละหนึ่งเมื่อป้อนสัญญาณนาฬิกาจากภายนอก 1 ลูกเข้ามาทางขา T0 (ขา 14) หรือ T1 (ขา 15) โดยไม่สนใจค่า Duty Cycle การตรวจสอบสัญญาณที่เข้ามาทางขานี้ จะตรวจสอบทุกๆ แมชชีนไซเคิล ดังนั้นการตรวจสอบสัญญาณนาฬิกา 1 ลูกต้องใช้ถึง 2 แมชชีนไซเคิล (24 คาบสัญญาณนาฬิกาของซีพียู) การเลือกใช้งานโหมดนี้ จะต้องโปรแกรมบิต C/ T = 1

การทำงานของเคาน์เตอร์แบ่งได้เป็น 4 โหมดดังนี้

- **โหมด 0** โหมดนี้ เคาน์เตอร์จะมีเพียง 13 บิต คือ TH0 8 บิตและ TL0 5 บิต โดย TL0 นับได้สูงสุดเพียง 3210 เท่านั้น และโหมดนี้นับได้สูงสุดเท่ากับ 819210
- **โหมด 1** (เคาน์เตอร์ 16 บิต) ทำงานเหมือนโหมด 0 ต่างกันที่ เคาน์เตอร์เป็นแบบ 16 บิต โดยแบ่งเป็น TL1 8 บิตและ TH1 8 บิต นับได้สูงสุด 6553610
- **โหมด 2** (8 บิต โหลดซ้ำอัตโนมัติ) โหมดนี้ จะใช้เคานท์เตอร์TL1 เป็นตัวนับซึ่งนับได้เพียง 8 บิต เวลาโหลดค่านับจะต้องโหลดเข้า TH1 เท่านั้นและโหลดเข้าได้เพียงครั้งเดียว เมื่อนับครบก็ จะเกิด โอเวอร์โฟลว์ ทำให้ TF1 ถูกเซต หลังจากนั้นค่าใน TH1 ก็ จะถูกโหลดเข้า TL1 อย่างอัตโนมัติ
- **โหมด 3** (เคาน์เตอร์ 8 บิต) โหมดนี้ จะใช้ TF0 เป็นแฟลก ของ TL0 ซึ่งเป็นเคาน์เตอร์ 8 บิต

3.7 การใช้งานพอรต์สื่อสารอนุกรม

พอรต์สื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่า ฟุลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TxD (P3.1) ทางด้านรับใช้ขา RxD (P3.0) และใช้SBUFเป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมพอรต์สื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM1และ SM0ซึ่งอยู่ในรีจิสเตอร์ควบคุม

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

รูปที่ 3.5 แสดงโครงสร้าง SCONRegister

การทำงานทั้ง 4โหมดของพอรต์สื่อสารอนุกรมมีดังนี้

- **โหมด 0** พอรต์สื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่งบิตเริ่มต้นแต่จะส่ง ชิฟต์คล็อก (Shift clock) ทางขา TxD0 (ความเร็ว 1/12 เท่าของสัญญาณนาฬิกาของ ซีพียู)
- **โหมด 1** พอรต์สื่อสารอนุกรม 10 บิต แบ่งเป็นข้อมูล 8 บิต บิตเริ่มต้น 1 บิต และบิตสิ้นสุด 1 บิต และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้โดยขึ้นอยู่กับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของไทม์เมอร์ 1
- **โหมด 2** พอรต์สื่อสารอนุกรม 11 บิต แบ่งเป็นข้อมูล 9 บิต บิตเริ่มต้น 1 บิต และบิตสิ้นสุด 1 บิต (TB8 นิยมนำมาใช้ส่ง Parity Bit)ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32และ 1/64ของสัญญาณนาฬิกาของ ซีพียูโดยขึ้นกับ SMODใน PCON
  BaudRateโหมด 2=(1/32)(ความถี่สัญญาณนาฬิกา) เมื่อ SMOD=1
  BaudRateโหมด 2=(1/64)(ความถี่สัญญาณนาฬิกา) เมื่อ SMOD=0
- **โหมด 3** พอรต์สื่อสารอนุกรม 11 บิต แบ่งเป็นข้อมูล 9 บิต บิตเริ่มต้น 1 บิต และบิตสิ้นสุด 1 บิตเหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของไทม์เมอร์ 1,2

บิต	ชื่อบิต	การทำงาน
SCON.7	SM0	บิตเลือกโหมดการทำงาน
SCON.6	SM1	
SCON.5	SM2	บิตเลือกการทำงานแบบ Single หรือ Multiprocessor 1 เลือก Multiprocessor ใช้ได้กับโหมด 2 และ 3 0 เลือก Single Processor ใช้ได้กับทุกโหมด
SCON.4	REN	บิตควบคุมให้รับหรือไม่รับข้อมูล 1 ให้รับข้อมูลได้

		0 ไม่รับข้อมูล
SCON.3	TB8	ข้อมูลบิตที่ 9 ที่ จะส่งออกไปในโหมด 2,3
SCON.2	RB8	ข้อมูลบิตที่ 9 จะรับเข้ามาในบิตนี้
SCON.1	TI	จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์
SCON.0	RI	จะเป็น 1 เมื่อสิ้นสุดการรับข้อมูล 1 ไบต์

ตารางแสดงที่ 3.12 รายละเอียดการทำงานของ SCON Register

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register อัตราเร็วในการรับหรือส่งข้อมูลเท่ากับ 1/12 ของสัญญาณนาฬิกาของ ซีพียู
0	1	1	การสื่อสารข้อมูลอนุกรม 8 บิต อัตราเร็วในการรับหรือส่งข้อมูลกำหนดได้จากไทม์เมอร์ 1,2
1	0	2	การสื่อสารข้อมูลอนุกรม 9 บิต อัตราเร็วในการรับหรือส่งข้อมูลเท่ากับ 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกาที่ขึ้นกับบิต SMOD ใน PCON
1	1	3	การสื่อสารข้อมูลอนุกรม 9 บิต อัตราเร็วในการรับหรือส่งข้อมูลกำหนดได้จากไทม์เมอร์ 1,2

ตารางที่ 3.13 แสดงรายละเอียดของโหมดการสื่อสารอนุกรม

Baud Rate	ความถี่สัญญาณนาฬิกา	SMOD ใน PCON	ไทม์เมอร์ 1		
			C/ T	โหมด	ค่าที่ โหลดซ้ำ
(โหมด 0)	12 MHz	X	X	X	X
(โหมด 1)	12 MHz	1	X	X	X
(โหมด 2)	12 MHz	0	X	X	X
(โหมด 1,3)					
62.5 kHz	12 MHz	1	0	2	FFH
19.2 kHz	11.0592 MHz	1	0	2	FDH
9.6 kHz	11.0592 MHz	0	0	2	FDH
4.8 kHz	11.0592 MHz	0	0	2	FAH
2.4 kHz	11.0592 MHz	0	0	2	F4H
1.2 kHz	11.0592 MHz	0	0	2	E8H
137.5	11.0592 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H

110	12 MHz	0	0	1	FEEBH

ตารางที่ 3.14 แสดงการกำหนด Baud Rate โดยไทม์เมอร์ 1

บทที่ 4
การสื่อสารข้อมูลแบบอนุกรม

การที่อุปกรณ์สองตัวจะสื่อสารกันได้นั้น จะต้องมี การเชื่อมต่อด้วยวิธี ใดวิธีหนึ่ง เพื่อให้สัญญาณไฟฟ้าที่ถูกส่งโดยฝ่ายหนึ่งสามารถถูกรับโดยอีกฝ่ายหนึ่งได้ การสื่อสารอาจเกิดขึ้นโดยตรงด้วยการเชื่อมต้ออุปกรณ์สองตัวด้วยสายสัญญาณหรือโดยอ้อมด้วยสื่อกลางที่ สอดแทรกเข้ามา สื่อกลางนี้มักจะเป็นระบบโทรศัพท์สาธารณะซึ่งในกรณีนี้ โมเด็ม (Modem) ทำหน้าที่ แปลงสัญญาณที่ ปลายด้านหนึ่งให้เป็นสัญญาณที่เหมาะสมกับการส่งผ่านสายโทรศัพท์ และเพื่อแปลงสัญญาณกลับที่ ปลายอีกด้านหนึ่ง สื่อชนิดอื่น เช่น เส้นใยนำแสงและการส่งผ้านคลื่นวิทยุ สามารถนำมาใช่ได้เช่นกัน อุปกรณ์การสื่อสารจะทำให้คอมพิวเตอร์สามารถสื่อสาร โดยใชสื่อเหล่านี้ เสมือนถูกเชื่อมต่อด้วยวิธีเดียวกับอุปกรณ์อนุกรมปกติ ดังนั้นหลักการที่กล่าวไว้สำหรับการสื่อสาร โดยตรงจะสามารถประยุกต์ใช้กับการสื่อสาร โดยอ้อมด้วยเช่นกันโดยต่อไปจะกล่าวถึงการเชื่อมต่ออุปกรณ์สองตัวโดยตรง สายสัญญาณและหัวต่อ (Connector) ที่ต้องใช้และมาตรฐานที่ ใช้กันทั่วไปในการเชื่อมต่อสาย

การเชื่อมต่อระหว่างพอร์ตอนุกรมแต่ละพอร์ตมีมาตรฐานอยู่สามแบบ คือ มาตรฐาน EIA RS-422, ระบบวงรอบกระแส และมาตรฐาน EIA RS-232 แต่เราส่งสัญญาณอนุกรมโดยใช้มาตรฐาน EIA RS-232 มากที่สุด โดยที่จะมีความยาวของสายได้ไม่เกิน 50 เมตร

4.1 มาตรฐาน EIA RS-232

มาตรฐาน EIA RS-232 หรือเรียกอีกอย่างว่า RS-232 กำหนดให้ระดับศักดาไฟฟ้าที่มีค่าเท่ากับ 3 โวลต์หรือสูงกว่า มีค่าทางตรรกะ เป็น 1 กำหนดให้ระดับศักดาไฟฟ้าที่มีค่าเท่ากับ -3 โวลต์หรือต่ำกว่า มีค่าทางตรรกะ เป็น 0 วงจรไอซีที่ใช้สร้างสัญญาณเหล่านี้ต้องการใช้แหล่งจ่ายไฟขนาด +12 โวลต์ RS-232 จะใช้สาย 1 เส้นสำหรับส่งข้อมูลและอีก 1 เส้นสำหรับรับข้อมูล ตารางต่อไปนี้แสดงตัวเชื่อมต่อแบบ D ชนิด 25 ขา (25 pin D-connector) และ สัญญาณต่างๆของ RS-232 บ่อยครั้งที่สัญญาณต่างของ RS-232 จะใช้สาย 1 เส้น สำหรับส่งข้อมูลและสายอีกหนึ่งเส้นจะใช้สำหรับรับข้อมูล โดยสัญญาณในแต่ละสายนี้จะถูกอ้างอิงเทียบกับกราวนด์ (ขาเบอร์ 7) มาตรฐาน RS-232 นี้ยังได้กำหนดสัญญาณ RS-232 ถึงแม้ว่าบ่อยครั้งที่สายสัญญาณส่ง RS-232 นี้ได้กำหนดไว้สัญญาณตอบรับเพื่อใช้ในการควบคุมการรับส่งข้อมูลด้วย

ในการเชื่อมต่อแบบ D ชนิด 25 ขา (25 pin D – connector) และสัญญาณต่างๆของ RS-232 ถึงแม้ว่าบ่อยครั้งที่สายส่งสัญญาณ RS-232 จะใช้ควบคู่กับตัวเชื่อมต่อแบบ 25 ขานี้ แต่ก็ยังไม่เป็นรูปแบบที่มาตรฐาน เนื่องจากตัวเชื่อมต่อแบบใหม่ที่มีขาเพียง 9 ขา (ดังตารางด้านล่างนี้) มาใช้แทน โดยเราจะพบได้ในเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นใหม่

มาตรฐาน RS-232 จะสามารถใช้ส่งข้อมูลได้ไกลที่สุด 50 เมตร ด้วยอัตรา 9600 บอดถ้าหากว่าจะส่งข้อมูลมากกว่า 50 เมตรขึ้นไปจะต้องทำการส่งที่เร็วกว่า 9600 บอด

9 Pin	EIA RS-232 Circuit	CCIT V.24 Circuit	RS-232 Description	Signal type & Direction
5	AB	102	Signal ground/common return	Ground/common
2	BB	104	Received data	Data from DCE
3	AA	103	Transmitted data	Data to DCE
1	CF	109	Received line signal detector	Control from DCE
4	CD	108,2	Data terminal ready	Control to DCE
6	CC	107	Data set ready	Control from DCE
7	CA	105	Request to send	Control to DCE
8	CB	106	Clear to send	Control from DCE
9	CE	125	Ring indicator	Control from DCE

ตารางที่ 4.1 แสดงข้อมูลการเชื่อมต่ออนุกรม 9 ขา

RS-232C

คุณสมบัติของ RS-232C

อัตราการรับส่งข้อมูล : 0-20000 บิต/วินาที

ระดับ แรงดันไฟฟ้าเอาต์พุตสูงสุดในภาวะไม่มีโหลด : -25 โวลต์ (ลอจิก 1)
+25 โวลต์ (ลอจิก 0)

ระดับแรงดันไฟฟ้าเอาต์พุตสำหรับโหลด 3-7 กิโลโอห์ม : ลอจิก“1” -15 โวลต์ (7 กิโลโอห์ม)
-5 โวลต์ (3 กิโลโอห์ม)
ลอจิก“0”+15 โวลต์ (7 กิโลโอห์ม)
+5 โวลต์ (3 กิโลโอห์ม)

กระแสเอาต์พุตเมื่อลัดวงจร : สูงสุด 500 มิลลิแอมป์เอาต์พุตอิมพีแดนซ์
เมื่อไม่จ่ายไฟเลี้ยงต่ำสุด 300 โอห์ม

สลูว์เรตทางเอาต์พุตสูงสุด : 30 โวลต์/ไมโครวินาที

ความต้านทานอินพุตของภาครับ : สูงสุด 7 กิโลโอห์ม
ต่ำสุด 3 กิโลโอห์ม

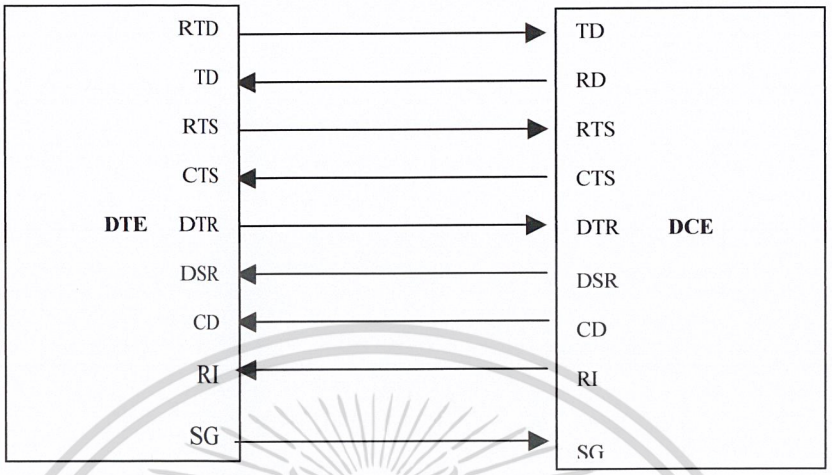
ค่าความจุอินพุตของภาครับ : สูงสุด 2500 กิโลฟารัด

ย่านแรงดันอินพุตของภาครับ : -25 โวลต์ ถึง +25 โวลต์

การจัดขาสัญญาณของ RS-232C

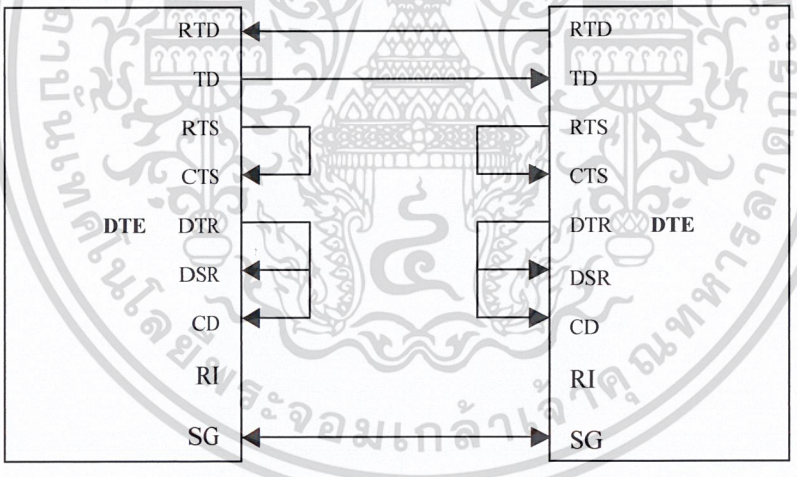
มีลักษณะเชื่อมต่อ 2 แบบด้วยกัน

1. การเชื่อมต่อกันระหว่างอุปกรณ์ ดีทีอี (DTE : Data terminal Equipment) เช่นคอมพิวเตอร์กับอุปกรณ์ ดีซีอี (DCE : Data Circuit Terminal) เช่น โมเด็ม แสดงดังรูปต่อไปนี้



รูปที่ 4.1 การเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ DCE

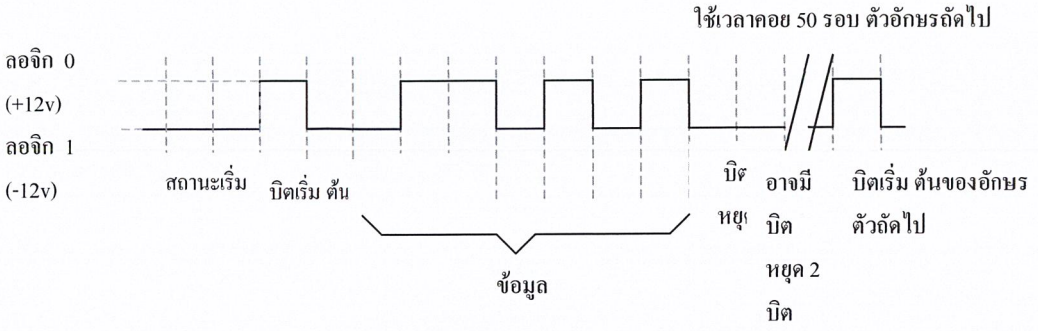
2. การเชื่อมต่อกันระหว่างอุปกรณ์ ดีทีอี (Data terminal Equipment) กับอุปกรณ์ ดีทีอี แสดงดังรูปต่อไปนี้



รูปที่ 4.2 การเชื่อมต่อกันระหว่างอุปกรณ์ DTE กับ DTE

4.2 รูปแบบของข้อมูลอนุกรมและอัตราบอดในการสื่อสารข้อมูลอนุกรม

อัตราบอด (baud rate) คือความเร็วในการส่งข้อมูลอนุกรม มีหน่วยเป็นบิตต่อวินาที



รูปที่ 4.3 การส่งข้อมูลแบบอนุกรม

เมื่อเราไม่ต้องการให้อุปกรณ์เชื่อมต่อภายนอกเชื่อมต่อโดยตรงกับระบบที่มีไมโครโปรเซสเซอร์เป็นพื้นฐาน เราจะต้องนำออปโตซิเลเตอร์ (optoisolator) มาใช้เป็นตัวป้องกันไมโครโปรเซสเซอร์ UART และวงจรอื่นๆในระบบที่มีศักดาไฟฟ้าสูงๆ เพื่อให้กระแสไหลเข้าที่สายกราวนด์

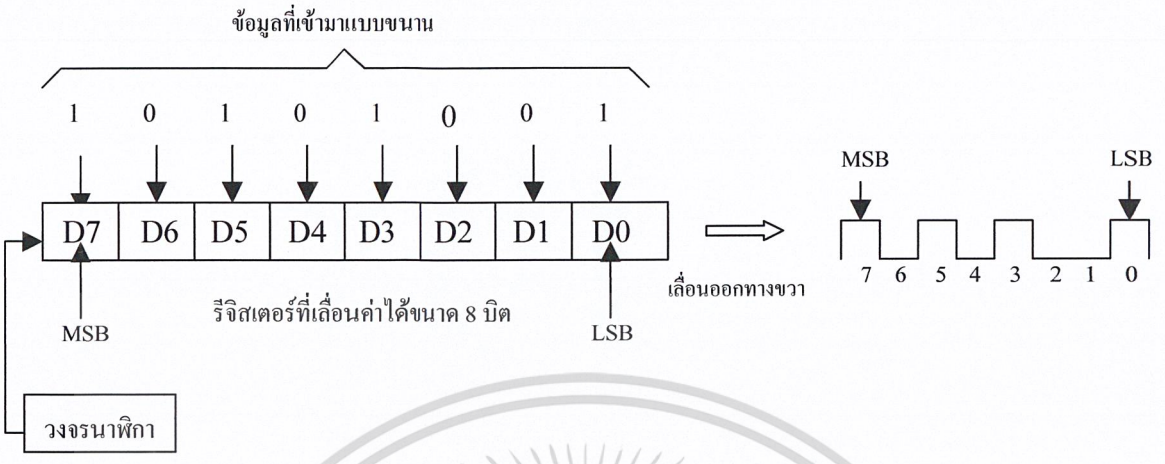
อัตราบอด	ช่วงเวลาของแต่ละบิต
110	9.91 ms
150	6.67 ms
300	3.33 ms
600	1.67 ms
1200	.833 ms
2400	.417 ms
4800	.208 ms
9600	.104 ms
19200	.052 ms

ตารางที่ 4.2 อัตราบอด

4.3 การเชื่อมต่อแบบอนุกรมและ UART

ในรูปด้านล่างนี้ แสดงการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยเริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ในชิฟต์รีจิสเตอร์ (Shift Register) หลังจากนั้นจะใช้สัญญาณนาฬิกาในการเลื่อนค่ารีจิสเตอร์ออกมาทีละบิต โดยเลื่อนค่าไปทางขวามือ บิตแรกที่ถูกเลื่อนออกมาคือบิต LSB ของข้อมูล

และบิตที่สองที่ถูกเลื่อนออกมาคือบิตที่อยู่ถัดจาก LSB และบิตต่อๆไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาคือบิด MSB ของข้อมูล

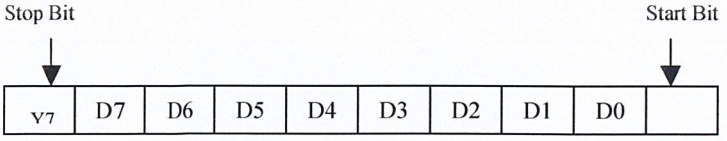


รูปที่ 4.4 การแปลงข้อมูลอนุกรมไปเป็นข้อมูลแบบขนาน

การเปลี่ยนแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลแบบขนานนั้นจะมีขั้นตอนตรงกันข้ามกับที่กล่าวมา นั่นคือข้อมูลแบบอนุกรมจะถูกเลื่อนเข้าเก็บใน ชิฟต์รีจิสเตอร์โดยใช้สัญญาณนาฬิกาเป็นตัวควบคุม และหลังจากที่ได้มีการเลื่อนข้อมูลทุกบิตเข้าไปใน ชิฟต์รีจิสเตอร์ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้จะถูกนำออกมาเป็นแบบขนานเพื่อนำไปใช้งานต่อไป

อุปกรณ์ที่ทำหน้าที่แปลงข้อมูลแบบอนุกรมเป็นแบบขนาน และแปลงข้อมูลแบบขนานเป็นแบบอนุกรม เรียกว่า UART (Universal Asynchronous Receiver-Transmitter) ซึ่งเป็นวงจร LSI ซึ่งนอกจากจะมีหน้าที่ในการเปลี่ยนแปลงข้อมูลแล้ว UART ยังมีหน่วยควบคุมและตรวจสอบการทำงานอีกด้วย

ในการขนส่งข้อมูลขนาด 8 บิตแบบอนุกรมนี้จะต้องมีบิตสตาร์ต (Start Bit) และบิตสตอป (Stop Bit) เพิ่มขึ้นมาซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงๆมีขนาด 10 บิต ในรูปต่อไปนี้ แสดงข้อมูลที่มีขนาด 8 บิต บิตสตาร์ต 1 บิต และบิตสตอป 1 บิต โดยที่บิตสตาร์ตมีค่าเป็น 0 บอก UART ที่ทำหน้าที่รับข้อมูลให้รู้ว่ามีข้อมูลกำลังเข้ามาและบิตสตอปมีค่า 1 บอกให้รู้ว่าการส่งข้อมูลได้เสร็จสิ้นลงแล้ว



รูปที่ 4.5 ข้อมูลอนุกรมขนาด 8 บิต

เราเรียกความเร็วของการส่งข้อมูลแบบ UART ว่าอัตราบอด (Baud Rate) มีหน่วยเป็นจำนวนบิตต่อวินาที (Bit per Second) ซึ่งจะบอกจำนวนบิตที่รับส่งในเวลา 1 วินาที เช่นการส่งข้อมูลด้วยอัตราบอด

1200 บอด ก็คือการส่งข้อมูลด้วยอักขระขนาด 10 บิต บิตสตาร์ต 1 บิต บิตข้อมูลที่มีขนาด 8 บิต และบิตสตอป 1 บิต) ได้ 120 ตัวอักขระใน 1 วินาที ซึ่งในตารางต่อไปนี้ แสดงอัตราบอดของ UART ที่ใช้กันโดยทั่วไป

อัตราบอด	ไบต์/วินาที
110	10
150	15
300	30
600	60
1200	120
2400	240
4800	480
9600	960
19200	1920
38400	3840

ตารางที่ 4.3 อัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

การส่งข้อมูลด้วยอัตรา 110 บอดนั้นจะมีรูปแบบแตกต่างกับอัตราบอดอื่นๆ ซึ่งในอัตรานี้จะต้องใช้บิตสตาร์ต 1 บิต และบิตสตอป 2 บิต ดังนั้นจึงทำให้ต้องส่งข้อมูลที่มีขนาด 11 บิต

เมื่อนำบิตที่ 8 ของข้อมูลมาใช้ตรวจสอบความผิดพลาดในการสื่อสารข้อมูล ซึ่งเรียกว่าบิตพาริตี (Parity Bit) UART ส่วนใหญ่สามารถสร้างและทำการตรวจสอบข้อมูลนั้นว่าเป็นพาริตีคู่หรือคี่ได้ ในการสร้างพาริตีคู่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูลทั้ง 8 บิต มีตัวเลข 1 เป็นจำนวนคู่ และในการสร้างพาริตีคี่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูลทั้ง 8 บิต มีตัวเลข 1 เป็นจำนวนคี่

เราสามารถใช่พาริตีคู่หรือคี่ในการตรวจสอบความผิดพลาดในการรับส่งข้อมูล โดยเมื่อ UART ได้รับข้อมูลก็จะทำการทดสอบว่าข้อมูลที่ได้รับมานั้นเป็นพาริตีคู่หรือคี่ ถ้า UART ตรวจสอบพบว่าข้อมูลที่ได้รับมานั้นมีพาริตีไม่ตรงตามที่กำหนด บิตพาริตีในรีจิสเตอร์ของ UART จะถูกเซ็ทเพื่อขอให้มีการส่งข้อมูลชุดเดิมนั้นกลับมาใหม่อีกครั้ง

เราสามารถแบ่ง UART ออกเป็นส่วนๆได้ 4 ส่วน ได้แก่ ส่วนที่ทำหน้าที่ส่งข้อมูล ส่วนรับข้อมูล ส่วนกำหนดสถานะ และส่วนที่เป็นวงจรควบคุม

ส่วนที่ทำหน้าที่ส่งข้อมูลของ UART แยกออกได้เป็น 2 ส่วน ซึ่งได้แก่บัฟเฟอร์ส่งข้อมูล (Transmitter data output buffer) กับรีจิสเตอร์ส่งข้อมูล (Transmitter register) โดยบิตข้อมูล 8 บิตจะถูกนำไปเก็บที่บัฟเฟอร์ส่งข้อมูล เมื่อสัญญาณควบคุมดาตาอินพุตสโตรบ (Data input strobe) เปลี่ยนค่าจาก 1 เป็น 0 และการส่งข้อมูลแบบอนุกรมจะเริ่มขึ้นเมื่อสัญญาณนี้เปลี่ยนค่าจาก 0 เป็น 1 หลังจากนั้น รีจิสเตอร์

ส่งข้อมูลจะทำหน้าที่เลื่อนข้อมูลส่งออกไปยังเส้นส่งข้อมูลอนุกรม โดยเริ่มจากบิตสตาร์ตถัดมาเป็นบิตข้อมูล D0 ถึง D7 และบิตสตอป

ส่วนที่ทำหน้าที่รับข้อมูลของ UART จะมีการทำงานที่ตรงกันข้ามกับส่วนที่ทำหน้าที่ส่งข้อมูล ข้อมูลที่เข้ามาทางเส้นรับข้อมูลอนุกรม (เข้า) จะถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล โดยการเลื่อนค่า 10 หรือ 11 ครั้ง การทำงานจะเริ่มเมื่อบิตสตาร์ตเข้ามา และเมื่อข้อมูลทั้งหมดถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล (Receiver register) แล้ว ข้อมูลในรีจิสเตอร์รับข้อมูลจะถูกนำไปเก็บบัฟเฟอร์รับข้อมูล (Receive-data output buffer) เมื่อมีสัญญาณควบคุมดาตาเอาต์พุตสโตรบ ทั้งรีจิสเตอร์รับข้อมูลและรีจิสเตอร์ส่งข้อมูลจะได้รับสัญญาณนาฬิกาขนาด 16 หรือ 64 เท่าของอัตราบอดที่ใช้ในการเลื่อนค่ารีจิสเตอร์ ทั้งรีจิสเตอร์รับข้อมูลและส่งข้อมูลจะได้รับสัญญาณนาฬิกาขนาด 16 หรือ 64 เท่าของอัตราบอดที่ใช้ในการเลื่อนค่ารีจิสเตอร์

รีจิสเตอร์ควบคุมการทำงานของ UART ทำให้เราสามารถกำหนดโหมดการทำงานของ UART ได้ โดยบิตควบคุมจำนวนข้อมูลทั้ง 2 บิตจะช่วยให้เราสามารถเลือกจำนวนของข้อมูลจริงที่รับมาว่าเป็น 5,6,7 หรือ 8 บิต ก็ได้

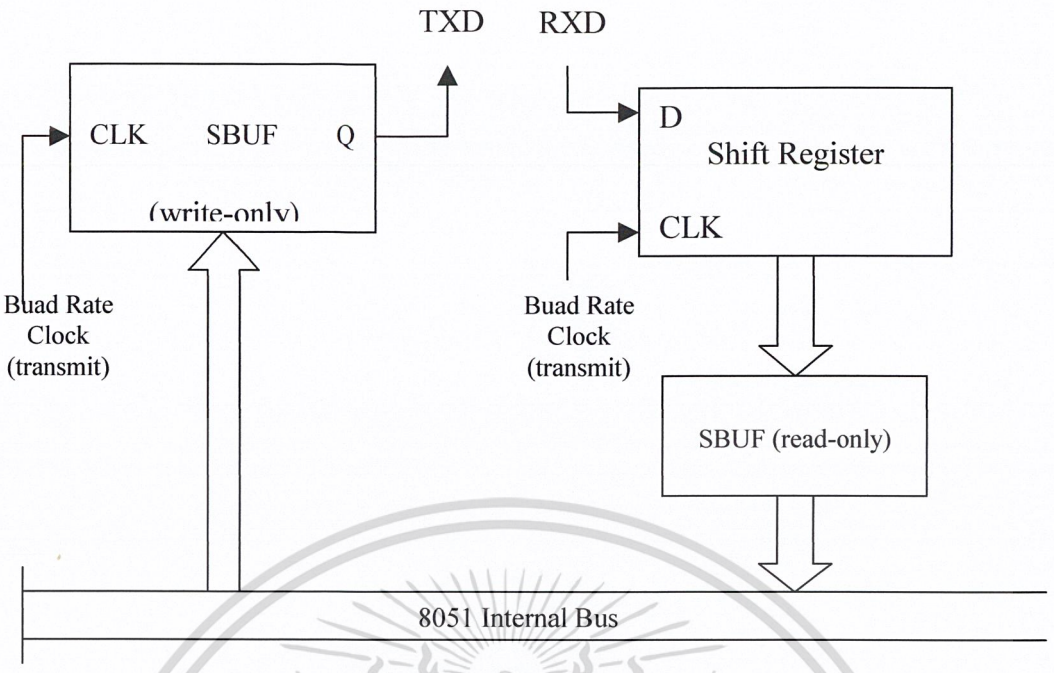
ในปัจจุบัน UART ถูกนำมารวมกับไอซีพิเศษที่ทำหน้าที่สร้างสัญญาณอัตราบอด โดยไอซีนี้จะสามารถสร้างสัญญาณที่มีอัตราบอดเป็น 16 หรือ 64 เท่าของอัตราบอดมาตรฐาน โดยสัญญาณนี้จะถูกสร้างจากวงจรกำเนิดสัญญาณแบบคริสตอล ทำให้ได้อัตราบอดที่มีอัตราคงที่และเที่ยงตรงมาก

4.4 MCS-51 กับการส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้นภายในชิป MCS-51 จะมี UART อยู่ในตัวซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ ถ้าเป็นไมโครโพรเซสเซอร์ เช่น เบอร์ Z-80 ต้องการส่งข้อมูลแบบอนุกรมจะต้องนำชิป UART มาประกอบกับพอร์ตอนุกรมของ MCS-51 ซึ่งจะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองข้างจะอยู่ในพอร์ต 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ตอนุกรมของ MCS-51 สมารถทำงานแบบ Full Duplex ๆได้ คือ สามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูล จะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน สเปเชียลฟังก์ชันรีจิสเตอร์ (Special Function Registers) โดยรีจิสเตอร์ซีเรียลพอร์ตบัฟเฟอร์(SBUF) จะอยู่ตำแหน่งที่ 99H ถ้าเขียนข้อมูลลงไปที่ตำแหน่งนี้ จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลออกจากตำแหน่งนี้ จะเป็นการรับข้อมูลจากพอร์ตอนุกรม โดยใน จะประกอบด้วยบัฟเฟอร์สองตัวสำหรับส่งและรับข้อมูลดังรูปที่ 2.10

สำหรับ ซีเรียลพอร์ตคอนโทรลรีจิสเตอร์ (SCON) ซึ่งอยู่ในตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลในระดับบิตได้ รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่างๆของการรับส่งข้อมูลแบบอนุกรม สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากการหารสัญญาณนาฬิกาที่ใช้กับ MCS-51



รูปที่ 4.6 การรับและส่งข้อมูลอนุกรมใน MCS-51

4.5 ซีเรียลพอร์ตคอนโทรลรีจิสเตอร์ (Seial Port Control Register)

MCS-51 มีโหมดการทำงานของพอร์ตอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมโหมดการทำงานได้โดยเขียนข้อมูลไปยังรีจิสเตอร์ SCON โดยความหมายของแต่ละบิตและโหมดการทำงานต่างๆ จะแสดงดังตารางต่อไปนี้

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตแฟลกกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับส่งข้อมูลในโหมด 2 และ 3 สามารถ set และ เคลียร์ ได้โดยใช้ software
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	TI	99H	บิตแฟลกแสดงการอินเทอรรัพต์ ภายหลังการส่งข้อมูลออกไป โดยจะ set เมื่อส่งข้อมูลออกไปหมดแล้ว และสามารถ เคลียร์ ด้วยการใช้ software
SCON.0	RI	98H	แฟลกแสดงการอินเตอร์รัปต์ ภายหลังรับข้อมูลเข้ามา สามารถ เคลียร์ ได้ด้วย software

ตารางที่ 4.4 บิตต่างๆของรีจิสเตอร์

SM0	SM1	MODE	ความหมาย	อัตราบอด
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้
0	1	1	8-bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจากไทม์เมอร์
1	0	2	9-bit UART	เปลี่ยนแปลงไม่ได้
1	1	3	9-bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจาก ไทม์เมอร์

ตารางที่ 4.5 โหมดต่างๆของการรับส่งแบบอนุกรม

ก่อนที่เราจะใช้พอร์ตอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อน เพื่อกำหนดโหมดการทำงานในลักษณะต่างๆ เช่น

MOV SCON,#01010010 B

เป็นการกำหนดให้พอร์ตอนุกรมทำงานในโหมด 1 และกำหนดให้มีการรับข้อมูล พร้อมทั้งกำหนด TI ให้เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้ว บิต TI จะถูก set เป็น 1 ในการส่งข้อมูล จะต้องคอยตรวจสอบบิต TI ถ้า TI ยังไม่เปลี่ยนเป็น 1 แสดงว่ายังส่งข้อมูลไปไม่หมด หากมีการเขียนข้อมูลไบต์ต่อไป ไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับการรับข้อมูล บิต REN จะต้อง set ให้เป็น 1 ยกเว้นโหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้วบิต R1 จะถูก set เป็น 1

4.6 โหมดการนทำงาน (Mode Of Operation)

ใน MCS-51 การสื่อสารทางพอร์ตอนุกรมจะมีอยู่ 4 โหมดโดยจะกำหนดที่บิต SM0 และ SM1 ใน SCON โดยจะมี 3 โหมดเป็นการสื่อสารแบบ Asynchronous จะมีบิตเริ่มต้น (start bit) และบิตจบ (stop bit) คล้ายการสื่อสารแบบ RS-232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะชิฟต์รีจิสเตอร์

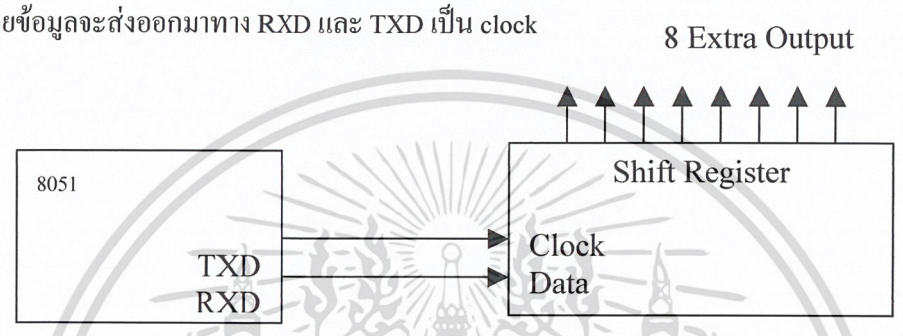
1. 8-Bit ชิฟต์รีจิสเตอร์(Mode 0)

การทำงานในโหมดนี้จะใช้ขา RXD ในการรับส่งข้อมูลโดยต่อกับ ชิฟต์รีจิสเตอร์ภายนอกส่วนขา TXD จะเป็น Output Shift Clock เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิต ถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่า Buad Rate เท่ากับ 1/12 ของความถี่ที่ใช้บน ชิป

ในการส่งข้อมูลจะทำได้โดยเขียนข้อมูลลงไปที่รีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD โดยจะสอดคล้องกับสัญญาณที่ออกมาจากขา TXD ซึ่งสัญญาณที่ถูกส่งออกมาจากขา TXD นี้จะส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก 0 ใน S3P1 และจะกลับเป็นลอจิก S6P1

สำหรับการรับข้อมูลจะรับได้เมื่อ set ขา Receiver Enable Bit (REN) เป็น 1 และ เคลียร์ ขา Receiver Interrupt Bit (RI) เป็น 0 ข้อมูลจะเข้าสู่ MCS-51 เมื่อ Clock Shift ถูกส่งออกไปทาง TXD ที่ขอบขาขึ้นของ Clock Shift บิตต่ำจะถูกส่งเข้ามาก่อน

ในการประยุกต์ใช้งานในโหมดนี้ จะต้องมี IC ชิฟต์รีจิสเตอร์มาต่อภายนอก เช่น หากต้องการส่งข้อมูลออกทางพอร์ตอนุกรม อาจต่อวงจรได้ดังรูปด้านล่างนี้ โดยใช้ไอซี Serial-to-parallel Shift Register ช่วย โดยข้อมูลจะส่งออกมาทาง RXD และ TXD เป็น clock



รูปที่ 4.7 การรับส่งข้อมูลออกทางพอร์ตอนุกรมใน MCS-51

2. 8-Bit UART with Variable Baud Rate (Mode 1)

ในโหมดนี้จะเป็นการส่งข้อมูลแบบ 10 บิต ประกอบด้วยบิตเริ่มต้น (0) ข้อมูล 8 บิต และบิตจบ (1) นอกจากนี้ยังสามารถกำหนดค่า อัตราบอด ได้โดยค่า อัตราบอด นี้จะแปรตามตัวจับเวลาตัวที่ 1 ในโหมดนี้จะส่งข้อมูลออกทาง TXD และรับเข้ามาทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัว stop bit จะเข้ามายังบิต RB8 ใน SCON

ถ้า Buad Rate ที่ใช้ในการรับส่งข้อมูลจะกำหนดโดย ไทม์เมอร์ 1 หลังจากโปรแกรมไปใน ไทม์เมอร์ 1 แล้วสามารถเลือกค่า อัตราบอด ได้อีกสองค่าคือ ค่าจากไทม์เมอร์ 1 โอเวอร์โฟลว์ หาร 32 กับค่าจาก ไทม์เมอร์ 1 โอเวอร์โฟลว์ หาร 16

การส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิต ไปที่ SBUF โดยบิตที่ 9 (stop bit) ให้เขียนลงใน TB8 ใน SCON จากนั้น ข้อมูลจะถูกส่งออกมาทางขา TXD โดยส่ง start bit ออกมาก่อน ตามด้วยข้อมูล 8 บิต และจบด้วย stop bit เมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag (TI) จะเป็น 1 ดังนั้นก่อนการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

ในการรับข้อมูล จะเริ่มจากเมื่อมีการเปลี่ยนลอจิกจาก 1 เป็น 0 ที่ขา RXD หมายความว่าเริ่มรับบิตเริ่มต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SUBF และ stop bit จะถูกเก็บลงใน RB8 ของรีจิสเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้ว บิต Interrupt Flag (RI) จะถูก set ดังนั้นการอ่านบิตข้อมูลจะทำได้หลังจากบิต RI ถูก set แล้ว เมื่ออ่านข้อมูลไปแล้วจะต้อง clean บิตนี้

3. 9-Bit UART with Fixed Baud Rate (Mode 2)

การทำงานในโหมดนี้ ไม่สามารถกำหนดค่าใน อัตราบอด ได้ ซึ่งค่า อัตราบอด จะมีสองค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิป การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต รวมบิตเริ่มต้นและหยุดเป็น 11 บิต โดยในข้อมูล 9 บิต จะเป็นข้อมูลจริงๆเพียง 8 บิต ส่วนอีก 1 บิตที่เหลือคือบิตที่สามารถโปรแกรมได้ ซึ่งจะใช้เป็น Parity Bit ในการส่งข้อมูลจะต้องเขียนไปที่บิต TB8 ในรีจิสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

4. 9-Bit UART with Variable Baud Rate (Mode 3)

การทำงานของโหมดนี้จะคล้ายกับโหมด 2 ต่างกันตรงที่สามารถกำหนด อัตราบอด ได้ โดยกรโปรแกรมไปที่ ไทม์เมอร์ 1 หลังจากโปรแกรมแล้วยังจะสามารถเลือกได้อีก 2 ค่าคือ ความถี่การ โอเวอร์โฟลว์ ของ ไทม์เมอร์ 1 หารด้วย 32 หรือ 16

4.7 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล

การรับข้อมูลถ้าจะให้ MCS-51 รับข้อมูลทางพอร์ตอนุกรมจะต้องโปรแกรมไปที่บิต Receiver Enable (REN) ในรีจิสเตอร์ SCON ให้เป็นลอจิก 1 ซึ่งทำได้ 2 วิธีดังนี้ SETB REN เป็นคำสั่งในการ set บิต REN ให้เป็น 1 หรืออาจทำได้โดยใช้คำสั่ง MOV SCON,#xxx1xxxxB ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า x หมายความว่า เป็นอะไรก็ได้ขึ้นกับการใช้งานในโหมดต่างๆ

ข้อมูลแบบ 9 บิต ในการรับส่งข้อมูลที่มีบิตข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และโหมด 3 ซึ่งการส่งข้อมูลบิตที่ 9 จะถูกเขียนใน TB8 โดยการเขียนโปรแกรมสำหรับการรับข้อมูล เมื่อข้อมูลเข้ามาถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

การเพิ่มบิต Parity การส่งข้อมูลแบบ 9 บิต สามารถใช้บิตที่ 9 เป็นบิต Parity ได้ ซึ่งบิต Parity จะอยู่ใน Program Status Word (PSW) โดยจะถูก set หรือ เคลียร์ ทุกๆ Machine Cycle ที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะส่งข้อมูลแบบ 8 บิต ตามด้วย Even Parity bit เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

```
MOV C, P          ;     อ่านค่าบิต P มาเก็บใน C
MOV TB8, C        ;     นำค่าบิต Parity มาเขียนลงใน TB8
MOV SBUF, A       ;     ส่งข้อมูลไปทางพอร์ตอนุกรม
```

ถ้าเป็นแบบ Odd Parity ให้แก้ไขข้อมูลที่อ่านได้จากบิต Parity เสียก่อนที่จะส่งออกไป ซึ่งเขียนโปรแกรมได้ดังนี้

```
MOV  C, P         ;     อ่านค่าบิต P มาเก็บใน C
CLP  C            ;     กลับค่าให้เป็น Odd Parity
MOV  TB8, C       ;     เขียนค่าลงใน TB8
MOV  SBUF, A      ;     ส่งข้อมูลไปทางพอร์ตอนุกรม
```

การส่งข้อมูลแบบมี Parity bit ด้วย สามารถส่งได้ในโหมด 1 ซึ่งส่งข้อมูลแบบ 8 บิต ได้ด้วย เช่น การส่งรหัส ASCII จะใช้บิตข้อมูล 7 บิต ส่วนบิตที่เหลืออีก 1 บิตนั้นจะเป็น Parity bit สามารถเขียนโปรแกรมได้ดังนี้

```
CLR  ACC.7          ;     เคลียร์ค่าบิต 7 เพื่อใช้เป็น Parity bit
MOV C, P            ;     อ่านค่าบิต P มาเก็บใน C
MOV TB8, C          ;     นำค่าบิต Parity มาเขียนลงใน TB8
MOV SBUF, A         ;     ส่งข้อมูลไปทางพอร์ตอนุกรม
```

แฟลกอิเทอร์รัพต์ เมื่อมีการรับส่งข้อมูลเสร็จสิ้นจะมีผลต่อแฟลกอินเตอร์รัปต์ (RI,TI) ในรีจิสเตอร์ SCON ซึ่งบิตเหล่านี้จะถูก set โดน ฮาร์ดแวร์แต่ต้อง เคลียร์ โดย Software

บิต RI ถ้าถูก set หมายความว่าบัฟเฟอร์ที่ใช้รับข้อมูลเต็มให้อ่านไปได้แล้ว และบิตนี้สามารถใช้อินเตอร์รัปต์ MCS-51 ได้ แต่ถ้าเขียนโปรแกรม จะใช้วิธีตรวจเช็กบิตนี้เป็น 1 หมายความว่าให้อ่านข้อมูลมาเก็บในรีจิสเตอร์ A ได้ แต่ก่อนอ่านต้อง เคลียร์ RI เสียก่อน เพื่อที่จะได้รับข้อมูลถัดไปซึ่งเขียนโปรแกรมได้ดังนี้

```
WAIT: JNB  RI, WAIT     ;     ถ้าไม่เป็น 1 จะทำงานอยู่ที่เดิม
      CLR  RI           ;     เคลียร์ RI
      MOV  SBUF, A      ;     อ่านค่ามาเก็บใน A
```

บิต TI เมื่อส่งข้อมูลออกไปแล้ว บิตนี้จะถูก set เป็นการแจ้งว่าบัฟเฟอร์ส่งข้อมูลว่างแล้ว ให้ส่งข้อมูลใหม่เขาไปได้ ซึ่งสามารถใช้บิตนี้อินเตอร์รัปต์ MCS-51 เช่นกัน ถ้าจะเขียนโปรแกรมคอยตรวจเช็กอาจเขียนได้ดังนี้

```
WAIT: JNB  TI, WAIT     ;     ตรวจว่า TI เป็น 1 แล้วหรือยัง
      CLR  TI           ;     เคลียร์ TI
      MOV  SBUF, A      ;     เขียนข้อมูลลงไป
```

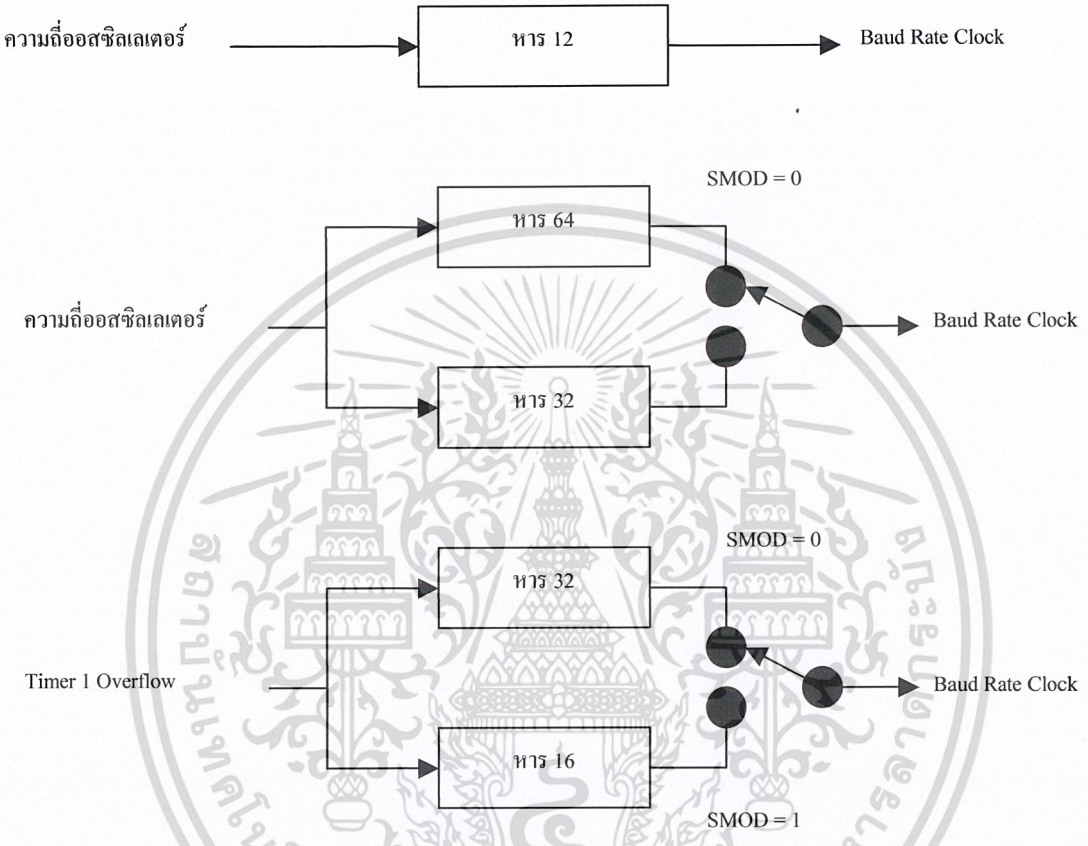
4.8 อัตราการส่งข้อมูลทางพอร์ตอนุกรม

จากการศึกษาการส่งข้อมูลข้อมูลในโหมดต่างๆ พบว่าโหมด 0 และ 2 ไม่สามารถกำหนด อัตราบอด เองได้ โดยในโหมด 0 ค่า อัตราบอด จะมีค่าเท่ากับความถี่ของ Oscillator หารด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ Oscillator หารด้วย 32 และหารด้วย 64 สองค่านี้เรียกว่า SMOD0 และ SMOD1 ซึ่งสามรถกำหนดได้จากรีจิสเตอร์ PCON บิตที่เจ็ด ในรีจิสเตอร์นี้ไม่สามารถเข้าถึงข้อมูลในระดับบิตได้ ถ้าเขียนข้อมูลลงไปทีละบิตต้องใช้วิธี Read-Modify-Write คืออ่านค่าขึ้นมาแล้วแก้ไขเขียนลงไปใหม่ อย่างเช่น

```
MOV  A,PCON         ;     อ่านค่าจาก PCONมาเก็บใจรีจิสเตอร์ A
SETB  ACC.7         ;     เซตบิต 7 (SMOD)
MOV  PCON,A         ;     เขียนค่าลงไปใหม่ใน PCON
```

สำหรับโหมด1และโหมด3 สามารถกำหนดค่า อัตราบอด ได้ด้วยการโปรแกรมลงไปใน ไทม์เมอร์ 1

ในการโปรแกรมแต่ละครั้ง จะมี SMOD สองค่าเช่นกัน ค่า อัตราบอด ของโหมดต่างๆแสดงได้ดังรูปต่อไปนี้



รูปที่ 4.8 โหมดต่างๆของอัตราการส่งข้อมูลออกทางพอร์ตอนุกรม

4.9 การใช้ไทม์เมอร์ 1 กำหนด อัตราบอด Clock

การกำหนดค่าลงในไทม์เมอร์ 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8-bit Auto Reload Maode (Mode 2) โดยเขียนค่าไปที่ TH1 ซึ่งสามารถโปรแกรมลงไปที่รีจิสเตอร์ TMOD ได้ดังนี้

MOD TMOD, #0010xxxxB (ค่า x เป็นค่าอะไรก็ได้เพราะบิตเหล่านี้ใช้ใน ไทม์เมอร์0)

ถ้าต้องการ อัตราบอด ต่ำๆเราก็สามารถใช้ 16-bit Mode ได้ โดยโปรแกรมเข้าไปเป็น TMOD = 0001xxxxB ค่า Buad Rate ที่ส่งออกมาจะมีค่าเท่ากับความถี่ของ ไทม์เมอร์ 1 เกิดโอเวอร์โฟลว์ หารด้วย 32 (หรือ 16 ถ้าเป็น SMOD = 1)

รูปแบบทั่วไปของการหาค่า อัตราบอด ในโหมด 1 และ 3 สามารถหาได้ดังนี้

Baud Rate = Timer 1 Overflow Rate/32

ถ้าต้องการ อัตราบอด เท่ากับ 1200 สามารถคำนวนค่าความถี่โอเวอร์โฟลว์ของไทม์เมอร์ 1 ได้ดังนี้

1200 = Timer 1 Overflow / 32

จะได้ Timer 1 Overflow = 38.4 กิโลเฮิรซ์

ถ้าระบบ MCS-51 ใช้ความถี่สัญญาณนาฬิกาจาก Crystal เท่ากับ 12 เมกกะเฮิรตซ์ตัวไทม์เมอร์ 1 จะได้รับ คล็อกเท่ากับ 1 เมกกะเฮิรตซ์ หรือ 100 กิโลเฮิรตซ์ถ้าเราต้องการไทม์เมอร์ 1 โอเวอร์โฟลว์ เท่ากับ 38.4 กิโลเฮิรตซ์ดังนั้นค่าอัตรา โอเวอร์โฟลว์ มีค่าเท่ากับ 1000 / 38.4 = 26.04 คล็อกโดยค่า โอเวอร์โฟลว์ จะเกิดเมื่อมีการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องให้ ไทม์เมอร์ 1 นับไป 26 ดังนั้น ค่าที่จะให้รีจิสเตอร์ TH มีค่าเท่ากับ –26 ซึ่งใช้เป็นค่ารีโหลด (Reload) จึงเขียนคำสั่งได้ดังนี้

```
MOV  TH1, # -26
```

ตัวโปรแกรมแอสเซมเบลอร์ทั่วๆไปจะแปลงค่า –26 เป็น 0E6H เอง จากที่ผ่านมาจะเห็นว่า ความถี่ อัตราบอด จะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาจาก Crystal ในตารางด้านล่างนี้เป็นค่าที่ต้องกำหนดใน ไทม์เมอร์ 1 เมื่อต้องการอัตราบอด ต่างๆ

ค่า อัตราบอด	Crystal	SMOD โหมด	ค่าใน TH1	ค่า อัตราบอด ที่ได้	ผิดพลาด
9600	12.000	1	-7 (F9H)	8923	7%
2400	12.000	0	-13 (F3H)	2404	0.16%
1200	12.000	0	-16 (E6H)	1202	0.16%
13200	11.059	1	-3 (FDH)	19200	0
9600	11.059	0	-3 (FDH)	9600	0
2400	11.059	0	-12 (F4H)	2400	0
1200	11.059	0	-24 (E8H)	1200	0

ตารางที่ 4.6 ความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ

4.10 การเชื่อมต่อหน่วยความจำกับ MCS-51

การจัดหน่วยความจำสำหรับ MCS-51 แบ่งได้เป็น 2 ส่วนใหญ่ๆได้แก่ Data Memory และ Program Memory

1. ดาตาเมโมรี (Data Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูล มีความจุได้ถึง 64 กิโลไบต์ เป็นหน่วยความจำประเภทอ่านและเขียนได้ (RAM) สัญญาณจาก MCS-51 ที่ใช้ในการอ่านข้อมูลจากแรมคือ RD\ และสัญญาณที่ใช้ในการเขียนข้อมูลลงแรม คือ WR\

แรม (Random Access Memory)

เป็นหน่วยความจำที่สามารถเขียนหรืออ่านข้อมูลลงไปได้ ข้อมูลจะสูญหายทันทีที่ขาดไฟเลี้ยง แบ่งออกเป็น 2 ชนิดคือสแตติกแรม (Static RAM)และไดนามิกแรม (Dynamic RAM)โดย สแตติกแรมเป็นหน่วยความจำที่ง่ายต่อการนำมาใช้งานแต่จะราคาสูงเมื่อเทียบกับไดนามิกแรมแต่จะนำใช้งานง่ายกว่า Stataicแรมเพราะไม่ต้องคอยรีเฟรช (Refresh) หน่วยความจำตลอดเวลาเหมือนไดนามิกแรม

2. โปรแกรมเมโมรี (Program Memory)

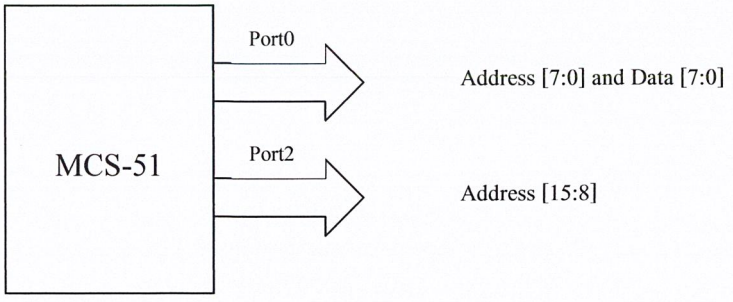
หน่วยความจำที่ใช้เก็บโปรแกรม เป็นหน่วยความจะซึ่งสามารถอ่านข้อมูลได้เพียงอย่างเดียว (ROM) โดยสัญญาณจาก MCS-51 ที่ใช้ในการอ่านรอม คือ PSEN\

รอม (Read Only Memory)

คือหน่วยความจำแบบถาวรสามารถรักษาข้อมูลไว้ได้โดยไม่ต้องใช้ไฟเลี้ยง ไม่สามารถเขียนข้อมูลลงไปได้สามารถอ่านข้อมูลได้เพียงอย่างเดียว

4.11 การเชื่อมต่อหน่วยความจำกับ MCS-51

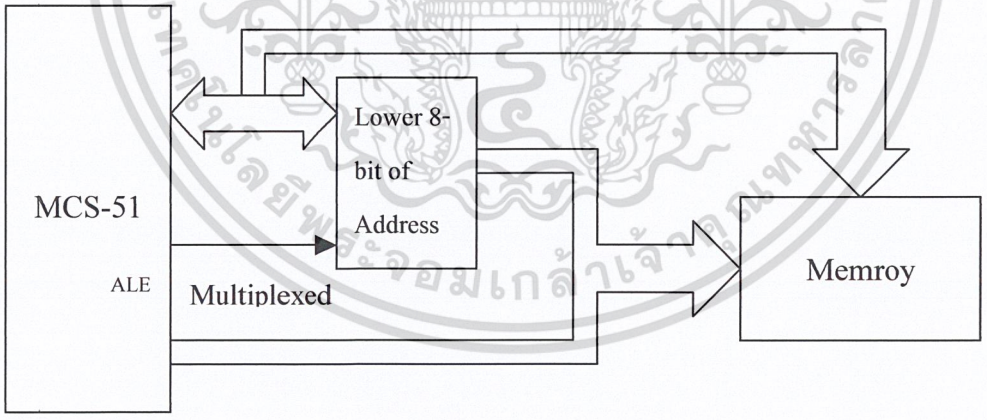
การเชื่อต่อกับหน่วยความจำจะต้องใช้สัญญาณหลายชนิดประกอบด้วย แอดเดรสบัส บัสข้อมูล และบัสควบคุมใน MCS-51 สามารถติดต่อกับสัญญาณภายนอกได้ 64 กิโลไบต์ โดยจะใช้สายสัญญาณแอดเดรสบัส 16 เส้น ส่งออกมาทางพอร์ต 0 และ 2 โดยพอร์ต 0 จะใช้มัลติเพล็กซ์ (Multiplex) ระหว่างแอดเดรสบัสและบัสข้อมูล โดยแอดเดรสบัสจะเป็นบิตต่ำ A0-A7 ดังรูปด้านล่าง



รูปที่ 4.9 การเชื่อมต่อหน่วยความจำกับ MCS-51

ในการเชื่อมต่อกับหน่วยความจำจะต้องมีอุปกรณ์ภายนอกมาแลช (Latch) สัญญาณแอดเดรสบัสที่ได้จากพอร์ต 0 เพื่อที่จะใช้พอร์ต 0 เป็นบัสข้อมูลต่อไป ในรูปต่อไปนี้จะเป็นการต่อหน่วยความจำประเภทรอม และแรมกับ MCS-51 โดยมีอุปกรณ์ภายนอกมาแลช ค่าแอดเดรสไบต์ต่ำเอาไว้เรียกว่า แอดเดรสแลชชิง (Address Latching) อุปกรณ์ที่นิยมใช้ได้แก่ ไอซี TTL เบอร์ 74LS373 โดนสัญญาณที่ใช้แลช คือ สัญญาณ ALE จาก MCS-51

การอ่านหน่วยความจำโปรแกรมและการอ่านเขียนหน่วยความจำข้อมูล การเชื่อมต่อขาแอดเดรสบัสและบัสข้อมูลจะเหมือนกัน แม้หน่วยความจำทั้งสองอยู่ในตำแหน่งเดียวกัน แต่จะแยกด้วยสัญญาณควบคุม



รูปที่ 4.10 การเชื่อมต่อหน่วยความจำกับ MCS-51ผ่านมัลติเพล็กเซอร์

4.12 ถอดรหัสตำแหน่งของหน่วยความจำ

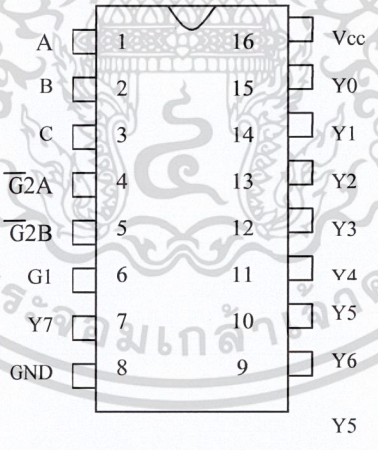
หน่วยความจำที่ถูกนำมาใช้ใน MCS-51 มีขนาดเล็กกว่า 64 กิโลไบต์ ดังนั้นจึงต้องนำมาเรียงกันแล้ว

ถอดรหัสของหน่วยความจำแยกออกเป็นแต่ละช่วง ถ้าใช้หน่วยความจำขนาด 4 กิโลไบต์ จำนวน 8 ตัวจะได้เท่ากับ 32 กิโลไบต์ ถ้าให้ตัวแรกเริ่มที่แอดเดรส 8000H ตัวที่สองก็จะเริ่มที่แอดเดรส 9000H ดังรายละเอียดในตารางด้านล่าง

EPROM ตัวที่	แอดเดรส	A15	A14	A13	A12	A11----------A0
1	8000-8FFF	1	0	0	0	X----------X
2	9000-9FFF	1	0	0	1	X----------X
3	A000-AFFF	1	0	1	0	X----------X
4	B000-BFFF	1	0	1	1	X----------X
5	C000-CFFF	1	1	0	0	X----------X
6	D000-DFFF	1	1	0	1	X----------X
7	E000-EFFF	1	1	1	0	X----------X
8	F000-FFFF	1	1	1	1	X----------X

ตารางที่ 4.7 รหัสตำแหน่งหน่วยความจำขนาด 4 กิโลไบต์

การถอดรหัสตำแหน่งของหน่วยความจำจะใช้ IC to 8-line Decoder เบอร์ 74LS138 ดังมีการจัดวางขาดังรูปและตารางต่อไปนี้



รูปที่ 4.11ไอซี 74LS138

Enable Inputs		Selecte Input			Outputs							
G1	G2	C	B	D	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H

H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

ตารางที่ 4.8 ค่าความจริงของ 74LS138

เมื่อพิจารณาจากตารางความจริงของ 74LS138 พบว่าจะต้องทำให้ขา G1 เป็น high และ G2A/ และ G2B/ เป็นโลว์ (low) แล้วนำอินพุตเข้าทางขา A,B,C ซึ่งอินพุตนี้ก็คือแอดเดรสของ MCS-51 นั่นเอง จากตารางที่ 1 พบว่าตารางที่ 1 เป็น high ตลอดให้นำมาต่อที่ขา G1 ได้เลย และจากตารางความจริงของ 74LS138 พบว่าขา A,B,C ทั้งสามขานี้เป็นศูนย์ทั้งหมด ขา Y0 จะแอคทีฟโลว์ ดังนั้นเมื่อพิจารณาจากตารางที่ 1 พบว่า A15,A14,A13 เป็นศูนย์ทั้งหมดซึ่งตรงกันพอดี ดังนั้นจึงได้การติดต่อกันอย่างสมบูรณ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 5
การทำงานของสเต็ปปิงมอเตอร์ (Stepping Motor)

สเต็ปปิงมอเตอร์คือมอเตอร์ชนิดหนึ่งที่ทำการหมุนเป็นขั้นๆทีละน้อยตามเส้นรอบวงและหยุด แตกต่างกับมอเตอร์ทั่วๆไปซึ่งจะหมุนไปเรื่อยๆตลอดเวลา ซึ่งเราจะสามารถควบคุมการหมุนสเต็ปปิงมอเตอร์ได้อย่างละเอียดโดยคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ สเต็ปปิงมอเตอร์สามารถใช้ในงานละเอียดต่างๆ เช่น ควบคุมการหมุนกระดาษในพรินเตอร์หรือใช้ในการเคลื่อนที่หัวอ่านและเขียนของฟลอปปี้ ดิสก์ เป็นต้นสเต็ปปิงมอเตอร์ ใช้งานลักษณะระบบเปิด (Open Loop System) คือ สเต็ปปิงมอเตอร์สามารถทำงานได้โดยไม่ต้องมีการ ป้อนค่าพารามิเตอร์กลับมา (Feed back) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอนนั้นล่ะ จะต้องการป้อนกลับไปยังระบบและตัวบอก ตำแหน่งว่าถูกต้องหรือผิดผลาดให้รับทราบ

การพันขดลวดบนสเตเตอร์ของสเต็ปปิงมอเตอร์ จะเห็นว่าการพันมีด้วยกัน 2 วิธี คือ แบบไบโพลาร์ (Bipolar) กับแบบยูนิโพลาร์ (Unipolar)

- -แบบไบโพลาร์ (Bipolar)จะมีการพันขดลวดหนึ่งขด(จำนวนรอบแล้วแต่การใช้งาน)ในแต่ขั้วแม่เหล็กของสแตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้น ที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียง การกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรสวิตซ์ชิ่งให้กลับขั้วไฟฟ้า
- แบบยูนิโพลาร์ (Unipolar)แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม เช่นกันกับการกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรสวิตซ์ซึ่งให้สลับหนึ่งไปยังอีกขั้วหนึ่งแทนกัน
- การพันขดลวดทั้ง 2 แบบที่กล่าวมาต่างกันคือ แบบยูนิโพลาร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพลาร์ ความแตกต่างที่รูปร่างของสเต็ปปิงมอเตอร์ทั้ง 2 แบบก็คือสายไฟที่ต่อมาจากตัว สเต็ปปิงมอเตอร์ซึ่งแบบไบโพลาร์จะมี 4 สาย ส่วนเป็นแบบยูนิโพล่าร์จะมี 5 สายหรือ 6 สาย

5.1 การสั่งงานควบคุมการหมุนของสเต็ปปิงมอเตอร์

ความยาวของขั้นการหมุนของสเต็ปปิงมอเตอร์ทั่วๆไปนั้นจะมีระยะตั้งแต่ 0.9 องศา ถึง 30 องศา โดยสเต็ปปิงมอเตอร์จะหมุนเป็นไปขั้นๆ โดยใช้การเปลี่ยนแปลงของกระแสในสนามแม่เหล็กของมอเตอร์สเต็ปปิงมอเตอร์ที่ใช้กันทั่วไปนั้นจะมี 3 แบบคือ แบบเวฟ, 2 เฟส และ 4 เฟส

- แบบเวฟ (wave)จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ตัวอย่างเช่น ขดที่ 1 , 2 , 3 , 4 ,1, 2 , 3 , 4 เป็นลำดับอย่างนี้ หรือ ขด 1 , 4 , 3 , 2 , 1 , 4 , 3 ,

2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า ดังในรูปของวงจรการจ่ายไฟ ที่อยู่ด้านบนนั้น เราสามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	-	-	-
2	-	ON	-	-
3	-	-	ON	-
4	-	-	-	ON
5	ON	-	-	-
6	-	ON	-	-

ตารางที่ 5.1 ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบเวฟ

- แบบ 2 เฟส (2 Phase) เป็นการกระตุ้นขดลวดทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป จะยกตัวอย่างการกระตุ้นขดลวดในลักษณะเรียงกันไป ให้ดูดังนี้ 12, 23, 34, 41, 12, 23, 34, 41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็น 14, 43, 32, 21, 14, 43, 32, 21 เรียงกันไปเรื่อยๆเช่นกัน

ข้อดีของการกระตุ้นแบบ 2 เฟส การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่าแบบเวฟ ซึ่งโรเตอร์จะหมุนด้วยแรง ดึงแบบเต็มๆแรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน

ข้อเสีย แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ ก็เป็นไปตามธรรมชาติ ได้อย่างก็ต้องเสียอย่าง นั้นล่ะครับ

เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางต่อไปนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	ON	-	-
2	-	ON	ON	-
3	-	-	ON	ON
4	ON	-	-	ON
5	ON	ON	-	-
6	-	ON	ON	-

ตารางที่ 5.2 ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบ 2 เฟส

- แบบ 4 เฟส (4 phase) แบบ 4-เฟส ในการที่เราจะทำให้สเต็ปปิงมอเตอร์แบบ4-เฟสหมุนตามเข็มหรือทวนเข็มนาฬิกาไปได้ในแต่ละขั้นนั้นเราจะต้องใช้สวิตซ์ 4 ตัว ในการควบคุมในที่นี้จะแทนด้วย SW1, SW2, SW3, และ SW4

สมมุติว่า SW1, SW2 เปิดอยู่ ถ้าเราต้องการที่จะหมุนสเต็ปปิงมอเตอร์ไปตามเข็มนาฬิกายังขั้นต่อไปเราก็ต้อง ปิด SW2 และเปิด SW4 และถ้าเราเปลี่ยนมาเปิด SW3 และ SW2 ก็จะทำให้สเต็ปปิงมอเตอร์หมุนไปอีกขั้นหนึ่ง ถ้าต้องการให้สเต็ปปิงมอเตอร์หมุนทวนเข็มนาฬิกาไปหนึ่งขั้นเราก็ต้องทำย้อนขั้นตอนที่กล่าวมาแล้ว ตารางการหมุนของ 4-เฟส สเต็ปปิงมอเตอร์ทั้งตามและทวนเข็มนาฬิกาจะแสดงในตารางต่อไปนี้

Step	Switch			
	sw4	sw3	sw2	sw1
1	0	0	1	1
2	1	0	0	1
3	1	1	0	0
4	0	1	1	0
1	0	0	1	1

ตามเข็ม ทวนเข็ม

Step	SW4	SW3	SW2	SW1
1	-	-	ON	ON
2	-	-	-	ON
3	ON	-	-	ON
4	ON	-	-	-
5	ON	ON	-	-
6	-	ON	-	-
7	-	ON	ON	-
8	-	ON	ON	-
1	-	-	ON	ON

ตารางที่ 5.3 ขั้นตอนการหมุนของเสต็ปปิงมอเตอร์แบบ 4 เฟส

หลังจากที่เราใส่ค่าสวิตซ์ต่างๆไปครั้งหนึ่งแล้ว เราจะต้องรออีก 2-3 มิลลิวินาทีก่อนที่จะใส่ค่าสวิตซ์ในขั้นตอนต่อไป โดยที่เราสามารถย้อนทำตามขั้นตอนที่แสดงไว้ในตารางข้างต้น จนกว่าจะถึงขั้นที่ต้องการ สำหรับสเต็ปปิงมอเตอร์ที่นำมาใช้ในโครงงานนี้เป็นแบบ 4-เฟส จะต้องทำการเคลื่อนที่ 48 step จึงจะครบรอบ 360 องศา หรือ มีการเคลื่อนที่ 7.5 องศาต่อ 1 step

ข้อดี การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเต็ปที่มีระยะสั้นลงอีกประการหนึ่งแต่ละ สเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย

ข้อเสีย ก็คงจะเช่นเดียวกับแบบ 2 เฟส ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟหรือจะใช้เท่ากับแบบ 2เฟส นั้นเอง

5.2 ชนิดของสเต็ปปิงมอเตอร์

สเต็ปปิงมอเตอร์แบ่งตามพื้นฐานออกได้ 3 ชนิดคือ วาริเอเบิลรีลัดแตนซ์ (variable reluctance : VR), เพอร์มาเนนท์แมกเน็ท (permanent magnet : PM) และแบบไฮบริด (hybrid)

ชนิดวาริเอเบิลรีลัดแตนซ์มีโครงสร้างของโรเตอร์แบบมัลติทูธ (multi-tooth) ทำจากเหล็กอ่อน เราจะทดสอบว่าเป็นมอเตอร์ชนิดนี้ได้ โดยการใช้นิ้วหมุนเพลาของมอเตอร์ ถ้าเป็นมอเตอร์ชนิดนี้จะไม่เกิดปรากฏการณ์ทางแม่เหล็ก (magnatism) มันจะหมุนได้ตลอดโดยไม่ติดขัดต่างจากชนิดเพอร์มาเนนท์แมกเน็ทและไฮบริด ซึ่งมีสนามแม่เหล็กที่โรเตอร์ เมื่อหมุนจะรู้สึกขัดๆเหมือนเป็นฟันเฟือง สเต็ปปิงมอเตอร์ชนิดนี้มีจุดด้อยเรื่องความถูกต้องของตำแหน่งและทำงานได้มีดีนักเมื่อมีสเต็ปในการหมุนสูง

ชนิดเพอร์มาเนนท์แมกเน็ทมีโครงสร้างของโรเตอร์แบบเรียบ ไม่มีซี่ขั้วแม่เหล็ก และบนโรเตอร์จะเป็นแม่เหล็กถาร ควบคุมได้โดยการป้อนกระแสไฟกระตุ้นที่ขดลวดบนสเตเตอร์ เช่น ถ้าเป็นสเตเตอร์แบบ 4 เฟส จะมีขั้วแม่เหล็กอยู่ 4 ขั้วซึ่งคอยล์พันอยู่แยกจากกัน ขั้วแม่เหล็กแบบถาวรบนโรเตอร์จะถูกแรงดึงดูดจากขั้วแม่เหล็กบนเสตอร์นั้นแม้ว่าจะไม่ป้อนกระแสไฟฟ้าให้อีกต่อไป ทำให้เกิดแรงยึดหน่วงขึ้น สเต็ปปิงมอเตอร์ชนิดนี้มีข้อดีในเรื่องความถูกต้องของตำแหน่ง และความเร็วมากขึ้นเมื่อเปรียบเทียบกับชนิดอื่น

ชนิดไฮบริดเป็นชนิดที่นิยมใช้งานมากที่สุด โดยเฉพาะกับอุปกรณ์ที่อยู่ในเครื่องคอมพิวเตอร์ มอเตอร์ชนิดนี้มีโครงสร้างภายในซึ่งได้จากการรวมโครงสร้างของมอเตอร์ทั้งสองแบบข้างต้นประกอบเข้าด้วยกัน จึงทำให้เป็นมอเตอร์ที่มีแรงยึดเหนี่ยวสูง และมีแรงบิตสูง ทำงานได้ดีแม้จะมีสเต็ปต่อรอบในการหมุนสูง

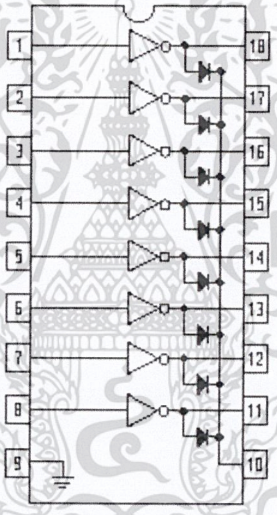
5.3 การเชื่อมต่อสเต็ปปิงมอเตอร์กับ ULN2803 (stepping driver motor)

เนื่องจากสเต็ปปิงมอเตอร์ต้องการใช้กระแสไฟฟ้าถึง 12โวลต์ แต่ไมโครโปรเซสเซอร์ MCS51 สามารถจ่ายกระแสไฟฟ้าออกมาได้เพียง 5 โวลต์เท่านั้น เราจึงจำเป็นต้องมีวงจรอินเตอร์เฟสเพื่อที่จะนำมาควบคุมการจ่ายกระแสไฟฟ้าให้กับสเต็ปปิงมอเตอร์ เพื่อจ่ายความต่างศักย์ไฟฟ้าเพิ่มขึ้นมาให้เพียงพอถึง

12 โวลต์ ในที่นี้ก็คือไอซี UNL2803 ซึ่งมีตารางและรูปแสดงคุณสมบัติพื้นฐานของไอซีตัวนี้มีดังต่อไปนี้คือ

Rating	Symbol	Value	Unit
กระแสไหลเข้า	VO	50	Volt
กระแสไหลออก (ยกเว้น ULN2801)	VI	30	Volt
กระแสรวม	IC	500	mA
กระแสเบส	IB	25	mA
ทำงานได้ที่อุณหภูมิ	TA	0 ถึง 70	^{o}C
เก็บที่อุณหภูมิ	Tstg	-55 ถึง 150	^{o}C

ตารางที่ 5.4 ตารางแสดงคุณสมบัติพื้นฐานของไอซี UNL2803



รูปที่ 5.1 ULN2803

บทที่ 6
การออกแบบการสร้าง และการคำนวณ

ในการออกแบบโครงงานนี้ เราจะแบ่งการออกแบบออกเป็นส่วนหลักๆ 3 ส่วนดังนี้

1. ส่วนรถบังคับ
2. ส่วนส่งสัญญาณควบคุมไร้สายและรับภาพไร้สาย
3. ส่วนโปรแกรมควบคุมการทำงานและรับภาพ

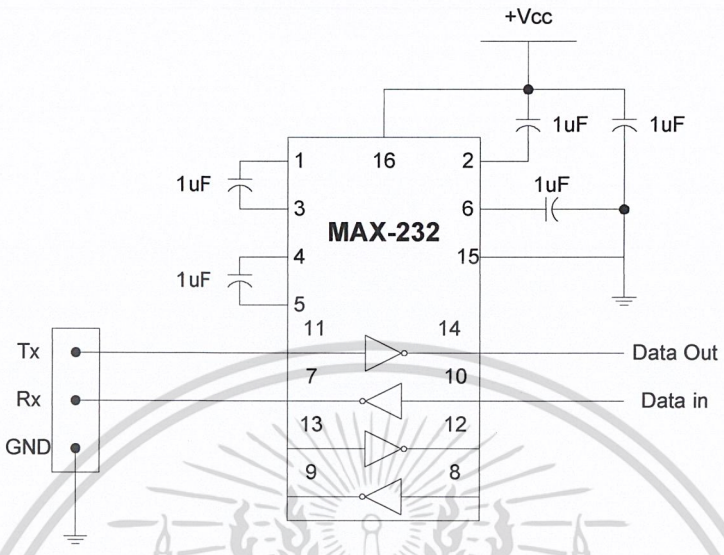
โดยการทำงานจะรับการควบคุมจากส่วนติดต่อกับผู้ใช้ ซึ่งจะใช้โปรแกรม Visual Basic 6 เป็นโปรแกรมติดต่อกับผู้ควบคุม ซึ่งคำสั่งที่ได้จากการสั่งงานของผู้ใช้จะถูกส่งผ่าน คอมพอร์ต (ComPort) ของคอมพิวเตอร์ ซึ่งเป็นการส่งข้อมูลแบบอนุกรม (Serial Communication) มาผ่านวงจรแปลงระดับสัญญาณ RS-232 เพื่อส่งไปยังภาครับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 โดยสัญญาณคำสั่งที่ได้จะถูกส่งไปยังส่วนวงจรขับเคลื่อนสเต็ปปิงมอเตอร์ เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ตามที่ผู้ใช้สั่งงาน

6.1 ส่วนการทำงานของอุปกรณ์ควบคุมฮาร์ดแวร์

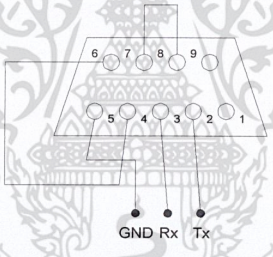
การควบคุม ประกอบด้วยส่วนรับสัญญาณข้อมูลมาจากคอมพิวเตอร์ และจะนำสัญญาณข้อมูลที่ได้มาผ่านวงจรแปลงระดับสัญญาณ RS-232 เป็นระดับสัญญาณ TTL และนำสัญญาณที่ได้ไปเข้าวงจรควบคุมการส่งสัญญาณควบคุม ซึ่งจะทำให้ได้สัญญาณควบคุม และจะนำสัญญาณนี้มาทำการมอดูเลตกับสัญญาณนาฬิกาที่ความถี่ 38 กิโลเฮิรตซ์ และนำสัญญาณที่ได้ทำการมอดูเลตแล้วไปทำการส่งโดยส่งผ่านวงจรส่งอินฟราเรด เพื่อส่งสัญญาณควบคุมแบบ ไร้สายไปยังภาครับสัญญาณควบคุม โดยจะแสดงการทำงานของวงจรแต่ละส่วนได้ดังนี้

6.2 วงจรเปลี่ยนระดับสัญญาณ RS-232 เป็นระดับสัญญาณ TTL

ในการติดต่อสื่อสารกับคอมพิวเตอร์ ข้อมูลที่ได้จากพอร์ตอนุกรมนั้นจะมีระดับสัญญาณที่ไม่สามารถใช้กับระดับสัญญาณ TTL ได้ จึงต้องทำการแปลงระดับสัญญาณให้มีค่าที่เหมาะสมเสียก่อน โดยต้องนำมาผ่านวงจรเปลี่ยนระดับสัญญาณ RS-232 เป็นระดับสัญญาณ TTL โดยในวงจรจะใช้ไอซี MAX-232 เป็นตัวแปลงระดับสัญญาณ



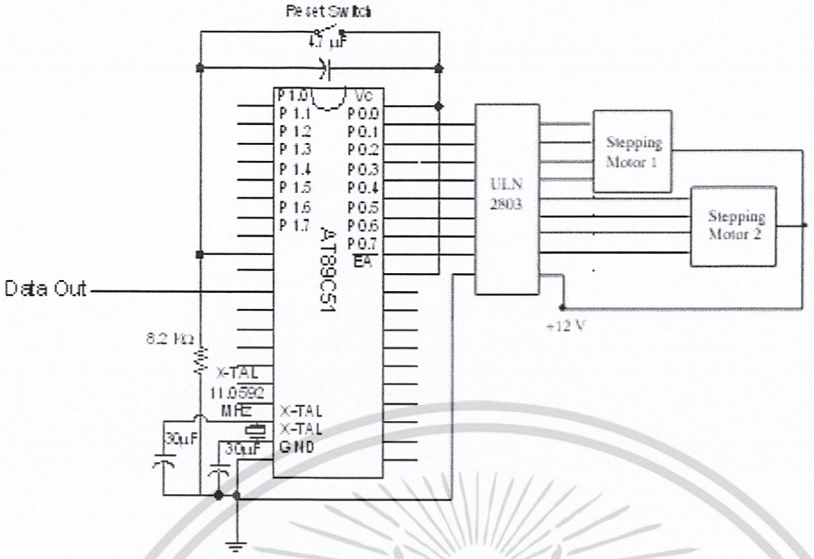
รูปที่ 6.1 วงจรเปลี่ยนระดับสัญญาณของ RS-232 เป็นระดับสัญญาณ TTL



รูปที่ 6.2 การต่อขาคอนเน็คเตอร์ที่ติดต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

6.3 วงจรแสดงผลข้อมูลที่ได้รับจากคอมพิวเตอร์

จะเป็นส่วนที่ใช้ในการตรวจเช็กสถานะของข้อมูลที่ได้รับมาจากคอมพิวเตอร์ ในวงจรนี้จะใช้ไอซีไมโครคอนโทรลเลอร์เบอร์ 89C51 ในการรับสัญญาณข้อมูลแบบอนุกรม เข้าที่ขา RXD (P3.0) และจะนำสัญญาณที่ได้ไปแสดงผลที่พอร์ต 1 ของไมโครคอนโทรลเลอร์ และจะส่งต่อไปยังวงจรควบคุมการส่งสัญญาณข้อมูลทางพอร์ต 2 การตรวจเช็กสถานะของข้อมูลที่ได้รับจากคอมพิวเตอร์ทำให้เราสามารถตรวจเช็กการทำงานของวงจร และโปรแกรมได้เร็วขึ้นเมื่อมีการผิดพลาดของการทำงานเกิดขึ้นโดยการนำมาต่อ กับไดโอดส่องแสง(LED)



รูปที่ 6.3 วงจรแสดงผลข้อมูลที่ได้รับจากคอมพิวเตอร์

6.4 ส่วนส่งสัญญาณควบคุมไร้สายและรับภาพไร้สาย

ส่วนส่งสัญญาณควบคุมไร้สายและรับภาพไร้สายนั้นเราจะใช้อุปกรณ์ด้านฮาร์ดแวร์ที่มีมาให้ในการติดต่อซึ่งมีอยู่ด้วยกันสองส่วนด้วยกันคือ

- อุปกรณ์บังคับรถไร้สาย
- อุปกรณ์รับภาพไร้สาย

โดยการใช้สเต็ปปิงมอเตอร์สองตัวในการบังคับอุปกรณ์บังคับรถไร้สาย ผ่านพอร์ตอนุกรม และอุปกรณ์รับภาพไร้สายทางพอร์ตยูเอสบี

6.5 ส่วนรถบังคับ

การเคลื่อนที่ของหุ่นยนต์สามารถแบ่งได้เป็น 2 แบบด้วยกัน คือ
แบ่งตามการเคลื่อนที่คือ

1. **เดินหน้า** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านซ้าย
2. **ถอยหลัง** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านขวา
3. **เดินหน้าเลี้ยวซ้าย** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านซ้าย แล้วสเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้าย
4. **เดินหน้าเลี้ยวขวา** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านซ้าย แล้วสเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้ายหันขวา

5. **ถอยหลังเลี้ยวซ้าย** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านขวา แล้ว สเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้าย
6. **ถอยหลังเลี้ยวขวา** ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านขวา แล้วสเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้ายหันขวา

แบ่งตามการควบคุม

1. เดินหน้า ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านซ้าย

2. ถอยหลัง ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ ตัวแรกเลื่อนบิตไปด้านขวา

3. หักล้อไปทางซ้าย ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้าย

4. หักล้อไปทางขวา ทำได้โดยสั่งให้สเต็ปปิงมอเตอร์ตัวที่สองเลื่อนบิตไปด้านซ้ายหันขวา

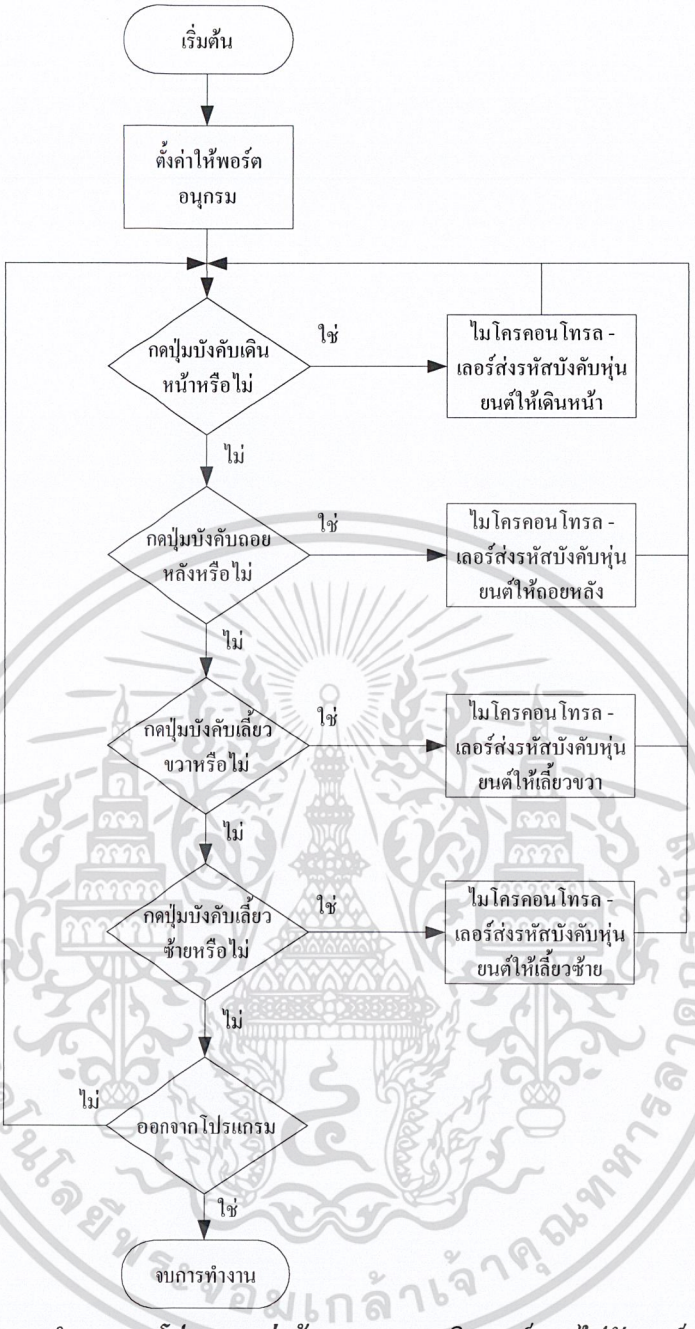
6.6 วงจรขับเคลื่อนมอเตอร์

ทำหน้าที่ขับเคลื่อนสเต็ปปิงมอเตอร์ 2 ตัว โดยวงจรขับเคลื่อนมอเตอร์ประกอบด้วยด้วย วงจรบัฟเฟอร์ซึ่งใช้ไอซีเบอร์ ULN2803 และ วงจรเพาเวอร์สวิตช์ดังที่กล่าวมาแล้ว โดยหลักการทำงานเป็นดังนี้ โดยสเต็ปปิงมอเตอร์จะประกอบด้วยขดลวด 4 เฟสซึ่งจะทำหน้าที่แปลงพลังงานไฟฟ้าเป็นพลังงานกล โดยเริ่มแรกสัญญาณที่ได้จากMCS-51 จะป้อนเข้าอินพุตของวงจรอินเวอร์ตเตอร์ซึ่งจะทำหน้าที่เป็นสวิตช์ตัวล่าง ซึ่งถ้าหากอินพุตที่เข้ามาเป็น "1" ทรานซิสเตอร์ในวงจรอินเวอร์ตเตอร์จะทำงานก็จะดึงกระแสลงกราวด์ เฟตก็จะไม่ทำงานเนื่องจากโวล์ตเทจตกคร่อมไม่ถึง V_{TH} ทำให้ไม่มีกระแสจ่ายให้แก่ขดลวด ทำให้ขดลวดในเฟสนั้นไม่ทำงาน

แต่ถ้าป้อนอินพุตเป็น "0" ทรานซิสเตอร์ในวงจรอินเวอร์ตเตอร์ก็จะไม่ทำงาน ทำให้โวลต์เตจตกคร่อมเฟสมีค่ามากกว่า V_{TH} ทำให้มีกระแสจ่ายให้แก่ขดลวดและสามารถทำงานได้ ซึ่งวงจรขับเคลื่อนมอเตอร์สามารถต่อวงจรตามรูปที่ 3.15

6.7 การส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม

โปรแกรมที่ทำหน้าที่ส่งข้อมูลจากคีย์บอร์ดไปยังพอร์ตอนุกรมนั้น เขียนขึ้นโดยอาศัยโปรแกรม Visual Basic 6.0 โดยการทำงานเริ่มจากการตั้งค่าให้กับพอร์ตอนุกรม ซึ่งก็ได้แก่หมายเลขคอมพอร์ต (ComPort) ที่ต้องการใช้ ซึ่งได้พอร์ต 1 และรายละเอียดของข้อมูลอนุกรมที่จะส่ง ได้แก่ อัตราบอดกำหนดเป็น 2400, การใช้พาริตีบิตซึ่งในที่นี้ไม่มีการใช้บิตดังกล่าว, จำนวนบิตข้อมูลที่จะส่ง กำหนดเป็น 8 บิต และค่ามาตรฐานการส่งข้อมูล กำหนดเป็น 1 เมื่อตั้งค่าแล้ว โปรแกรมจะรอรับสัญญาณจากคีย์บอร์ดหากมีการกดปุ่มหนึ่งปุ่มใดตามที่กำหนดไว้ ค่าของตัวแปรของปุ่มนั้นๆจะถูกส่งออกไปยังพอร์ตอนุกรมทันที ลักษณะการทำงานของโปรแกรมเป็นดังโฟลว์ชาร์ต



รูปที่ 6.4 การทำงานของโปรแกรมส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม

6.8 โปรแกรมควบคุมหุ่นยนต์ผ่านพอร์ตอนุกรม

สำหรับโปรแกรมที่ใช้ควบคุมหุ่นยนต์ผ่านพอร์ตอนุกรมซึ่งเขียนด้วยโปรแกรม Visual Basic 6.0 นั้นมีคุณสมบัติที่ดีคือ เขียนง่ายสามารถเรียกใช้ คอมโพเน็นต์ต่างๆได้โดยง่าย เช่นการติดต่อกับอุปกรณ์รับภาพไร้สายโดยเรียกใช้คอมโพเน็นต์ ezvid60 ซึ่งนำมาจากในอินเตอร์เน็ต สามารถดึงภาพสู่หน้าจอได้โดยง่าย และ คอมโพเน็นต์ MS Comm Control ใช้สำหรับการติดต่อกับพอร์ต อนุกรม

```
        MOV   SCOM, #50H          - set initial value of retrive data
        MOV   TMOD, #20H
        MOV   TH1, #0FDH          - set ค่า bit rate = 9600 b/s
        MOV   TL1, #0FDH
        MOV   A, PCON
        CLR   ACC7
        MOV   PCON, 9
        SETB  EA                  - set enable interupt switch
        SETB  ES
        SETB  TRI

LOOP :
        PUSH  ACC                 - keep old value
        PUSH  PSW
        MOV   P2, A               - set data port
        CLR   RI
        MOV   P2, A
        MOV   P1,A
        POP   PSW                 - retrive old value
        POP   ACC
        RETI                      - go to loop
        END.
```

รูปที่ 6.5 แสดงการทำงานของโปรแกรมส่งข้อมูลจากคอมพิวเตอร์ออกไปยังพอร์ตอนุกรม

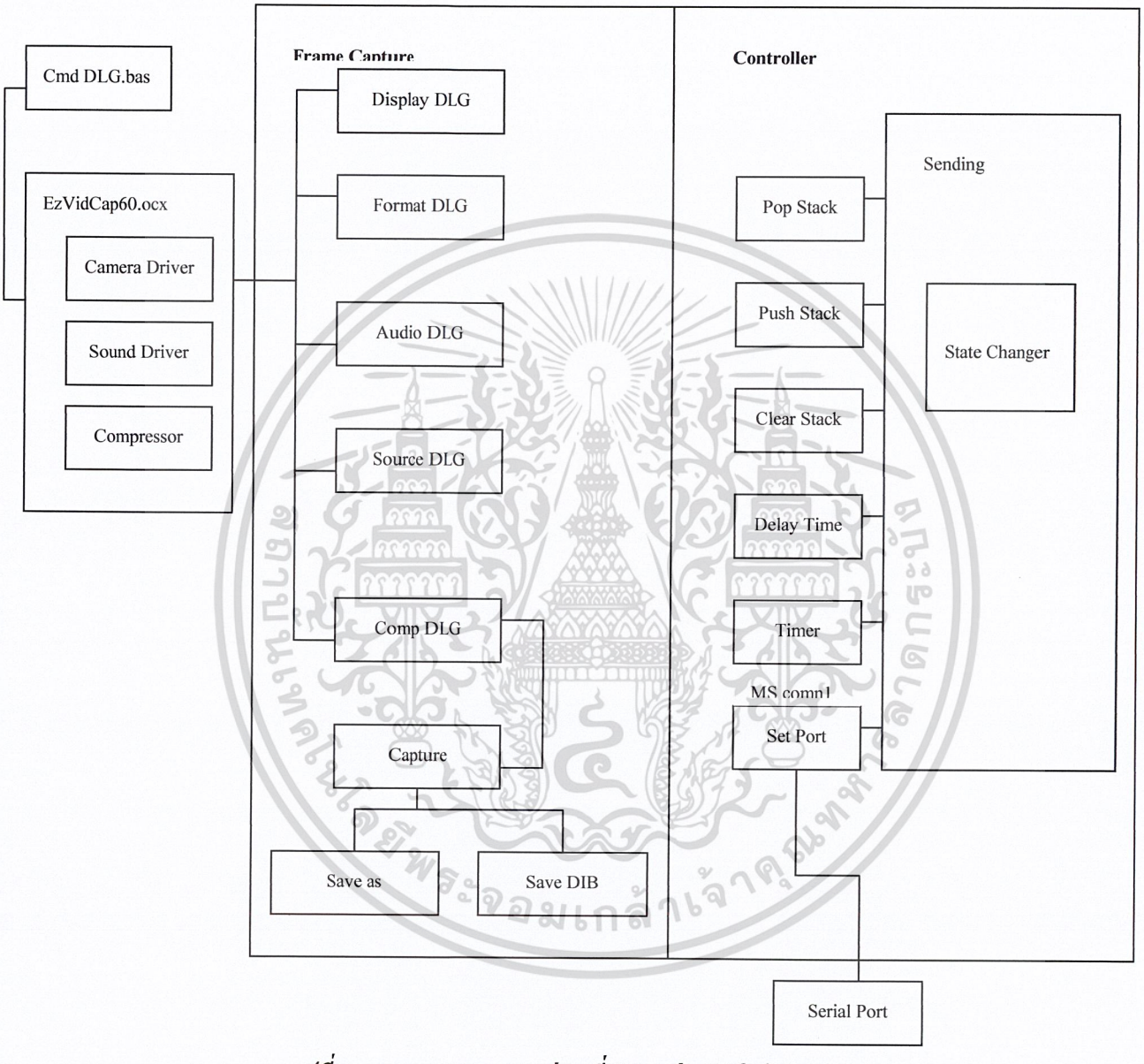
6.9 โปรแกรมควบคุม Security Robot

ในการควบคุม Security Robot นั้นเราจะใช้ภาษา Visual Basic ในการควบคุมและรับภาพออกทางโปรแกรม โดยที่จะแบ่งได้ออกเป็นสองทางดังนี้

ส่วนรับภาพ เป็นส่วนที่ทำหน้าที่รับสัญญาณจากกล้องวิดีโอ ผ่านตัวรับภาพไร้สายโดยที่มีอุปกรณ์ติดต่อพอร์ต USB และ ไดร์ฟเวอร์ ให้มาอยู่แล้ว เราจำเป็นต้องเรียกคอมโพเน็นต์ ที่ทำหน้าที่รับสัญญาณภาพแล้วทำการส่งภาพออกหน้าจอ

ส่วนควบคุม นั้นจะทำหน้าที่ไปบังคับส่วนของ ฮาร์ดแวร์ที่มี สเต็ปปิงมอเตอร์อยู่สองตัวเพื่อไปจับบังคับกับอุปกรณ์ทำการควบคุม ผ่านทางพอร์ตอนุกรม

การทำงานของทั้งสองส่วนนั้นจะแยกออกจากกันโดยพอร์ตสื่อสารทั้งสองทาง ในทางด้านส่วนรับภาพนั้นจะมีฟังก์ชันที่ไว้ใช้ปรับแต่งค่าต่างๆของตัวกล้องแต่ทำหน้าที่บันทึกภาพทั้งแบบ AVI ซึ่งเป็นรูปแบบภาพเคลื่อนไหว และแบบ bmp ซึ่งแบบภาพนิ่งได้ โดยส่วนการควบคุมนั้นจะทำหน้าที่บังคับการเคลื่อนที่ของตัวรถเพียงอย่างเดียวเท่านั้น



รูปที่ 6.6 แสดง component ต่างๆที่ทำงานร่วมกันในโปรแกรมควบคุม

บทที่ 7

ผลการทดลอง

7.1 การติดต่อกับพอร์ตอนุกรม

การติดต่อกับพอร์ตอนุกรมแบ่งเป็นสองส่วนด้วนกันคือ ติดต่อในรูปแบบของฮาร์ดแวร์และการติดต่อในรูปแบบซอฟแวร์โดยการติดต่อกับส่วนฮาร์ดแวร์นั้นผลการทดลองมีดังนี้

ผลการทดลองในการควบคุมการเคลื่อนที่

การทำงานของอุปกรณ์ฮาร์ดแวร์ที่ใช้ในการบังคับการเคลื่อนที่

สเต็ปปิงมอเตอร์ ซึ่งตรวจสอบการหมุนของสเต็ปปิงมอเตอร์โดยทำการทดลองโดยป้อนไฟเป็นแบบ 1 เฟส และแบบ 2 เฟสโดยที่ สเต็ปปิงมอเตอร์ที่ได้มาต้องวนครบ 24 เฟสด้วยกันถึงจะได้หนึ่งรอบของการหมุน

ไอซี MAX 232 ทำหน้าที่แปลงค่าจากสัญญาณ TTL เป็นรูปแบบบิต แต่ยังเป็น การส่งข้อมูลในรูปแบบการส่งแบบอนุกรมอยู่

MCS-51 ประมวลผลและแปลงค่าข้อมูลจาก MAX 232 ที่เป็นแบบอนุกรม และแยกออกมาในแต่ละพอร์ตที่จะทำงาน

ULN 2803 จ่ายไฟเพิ่มให้อุปกรณ์ที่ต้องการกำลังไฟมากขึ้น

อุปกรณ์ฮาร์ดแวร์ต้องใช้ซอฟแวร์ในการตรวจสอบก่อนหน้าที่เราจะเขียนโปรแกรมติดต่อควบคุมนั้นซึ่งเราใช้โปรแกรม Hyper Terminal ในการป้อนข้อมูลลงไปผ่านพอร์ตอนุกรม โดยทำการตรวจสอบในแต่ละบิตที่ได้ส่งออกไปด้วยค่าแอสกี เนื่องจาก สเต็ปปิงมอเตอร์ ทำการตรวจสอบได้ยาก เราจึงทำการต่อ LED แล้วตรวจสอบผ่านอีกพอร์ตหนึ่งของ MCS-51 ซึ่งได้ค่าที่ถูกต้องตามที่ได้ทดลองป้อนลงไปใน4บิตล่าง

ค่า ASCII	บิต
3	0011
6	0110
12	1100
9	1001

ตารางที่ 7.1 การแปลงข้อมูลจาก ASCII ไปเป็นบิต

การทำงานในส่วนของซอฟแวร์นั้นมีการควบคุมเกี่ยวกับสเต็ปปิงมอเตอร์ซึ่งใช้คีย์บอร์ดในการควบคุมโดยที่ต้องส่งค่าที่เป็นแอสกี แล้วแปลงค่านั้นให้เป็นบิตที่จะทำให้สเต็ปปิงมอเตอร์นั้นหมุนอย่างถูกต้องโดยที่ต้องทำงานเป็นทีละลำดับขั้นในการส่งข้อมูลไปยังอุปกรณ์ฮาร์ดแวร์

การส่งข้อมูลนั้นเราใช้ พอร์ตเดียวสำหรับการบังคับสเต็ปปิงมอเตอร์ซึ่งในรูปแบบภาษา Visual Basic นั้นจรึงต้องแยกค่าใน 1 ไบต์ แบบ 4 บิตล่างและ 4 บิตบนด้วยกัน เพื่อแยกในการคำนวณค่าในการหมุนของ สเต็ปปิงมอเตอร์

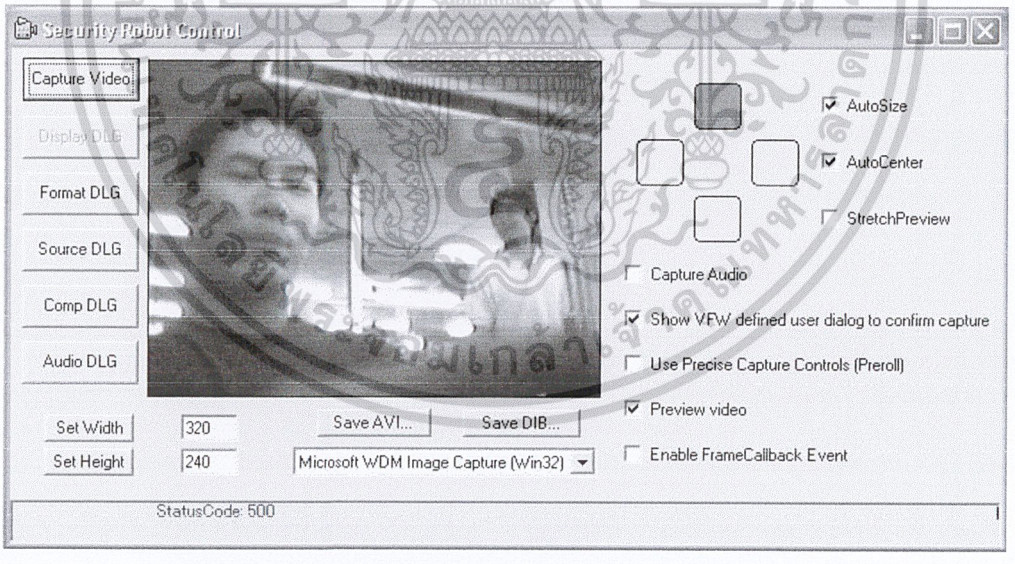
เมื่อทำการกดคีย์บอร์ดแล้วจะทำการส่งค่าดังนี้

กดปุ่ม Up Arrow และ Down Arrow จะทำการเปลี่ยน 4 บิตบนที่ทำการส่งไปยังพอร์ตอนุกรม

กดปุ่ม Left Arrow และ Right Arrow จะทำการเปลี่ยน 4 บิตล่างที่ทำการส่งไปยังพอร์ตอนุกรม เมื่อทำการกดปุ่มจะมีการ เปลี่ยนแปลงของบิตที่ส่งไปและทำให้รถเคลื่อนที่

สัญญาณควบคุม	การทำงานของรถบังคับ
เมื่อไม่ได้ทำการกดปุ่ม	ไม่มีการเคลื่อนที่
เมื่อกดปุ่มที่ 1	เดินหน้า
เมื่อกดปุ่มที่ 2	เลี้ยวขวา
เมื่อกดปุ่มที่ 3	ถอยหลัง
เมื่อกดปุ่มที่ 4	เลี้ยวซ้าย

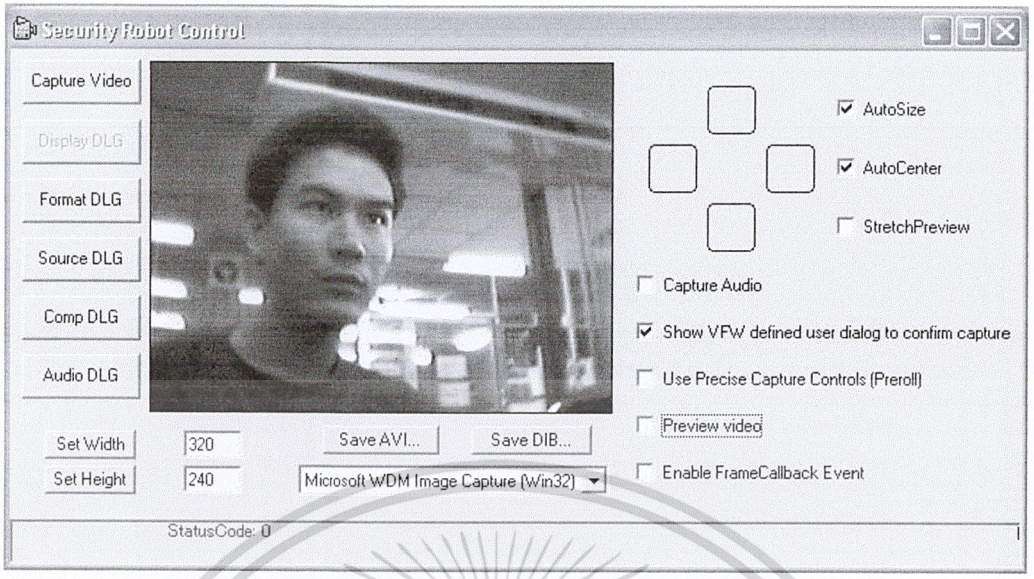
ตารางที่ 7.2 แสดงผลลัพธ์ที่เกิดขึ้นหลังจากกดปุ่ม



รูปที่ 7.1 ตัวอย่างลักษณะหน้าต่างโปรแกรมเมื่อมีการป้อนอินพุต

7.2 ผลการทดลองในส่วนของการรับภาพ

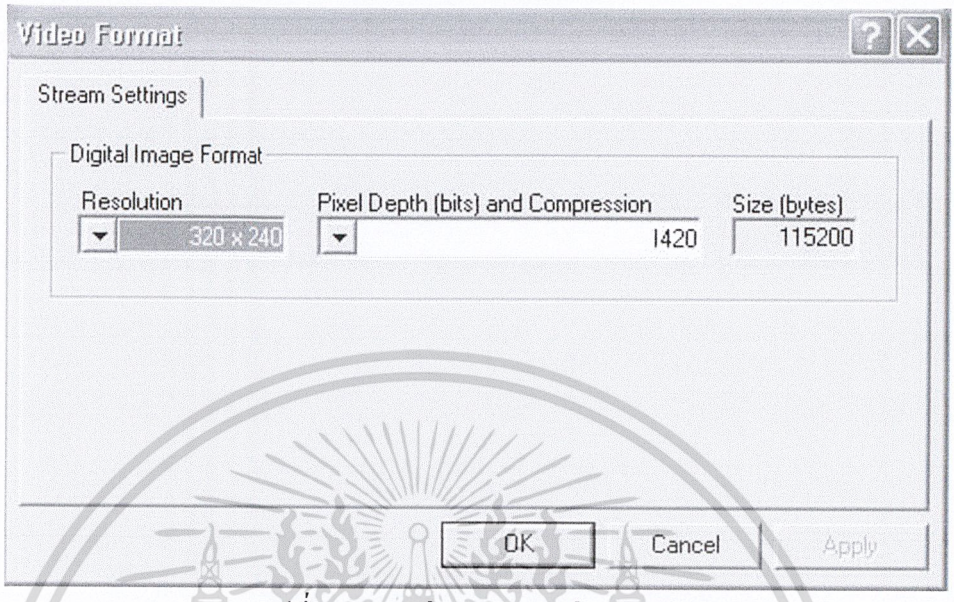
การทดลองการรับภาพนั้นเราจะต้องทำการติดต่อกับอุปกรณ์กล้องตลอดทำการเขียนโปรแกรม เนื่องจาก คอมโพเน็นต์ ezvid60 สามรถดึงภาพมาใช้ได้โดยง่าย และยังมีฟังก์ชันต่างๆให้ใช้มากมาย



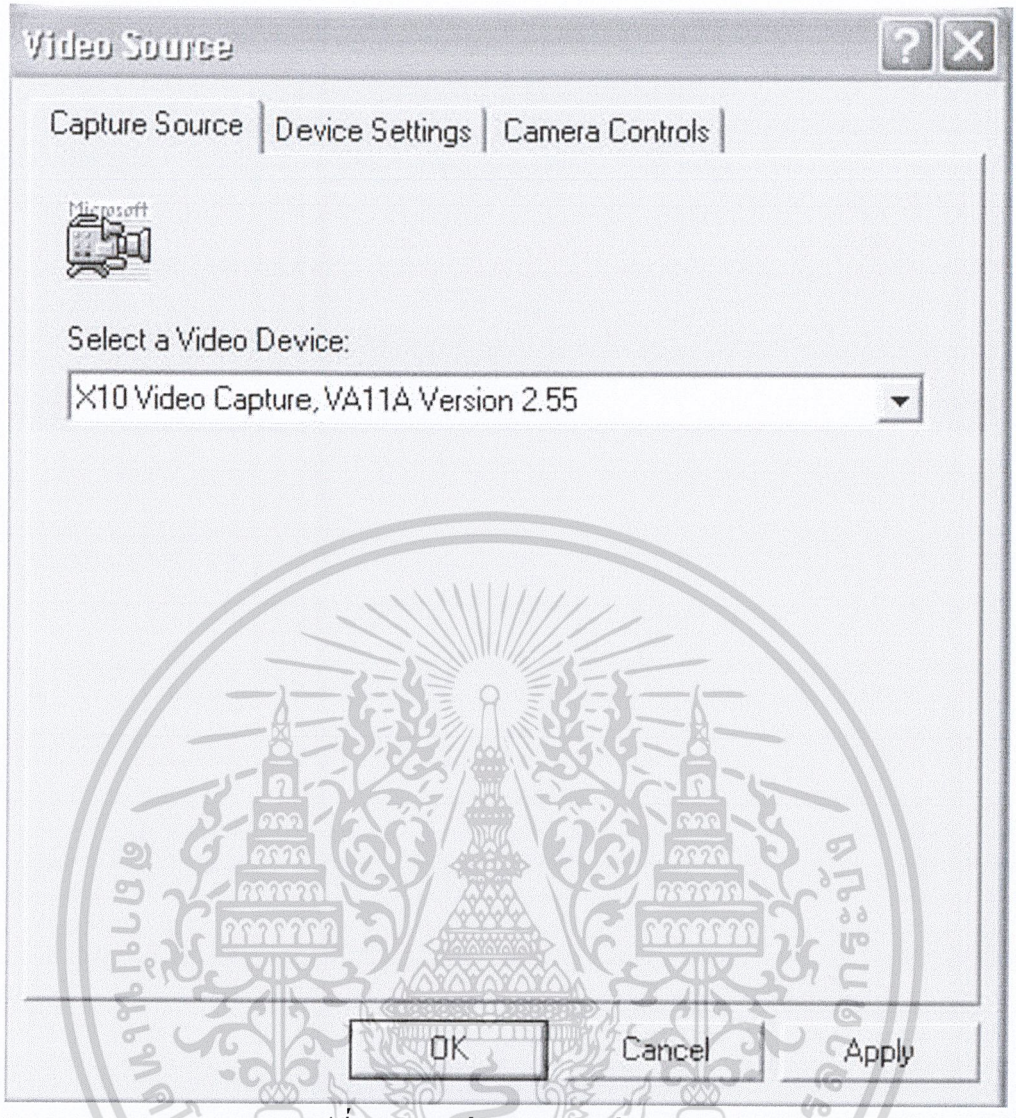
รูปที่ 7.2 ผลการทดลองในการรับภาพ

การใช้งานโปรแกรมควบคุม Security Robot ในส่วนรับภาพ

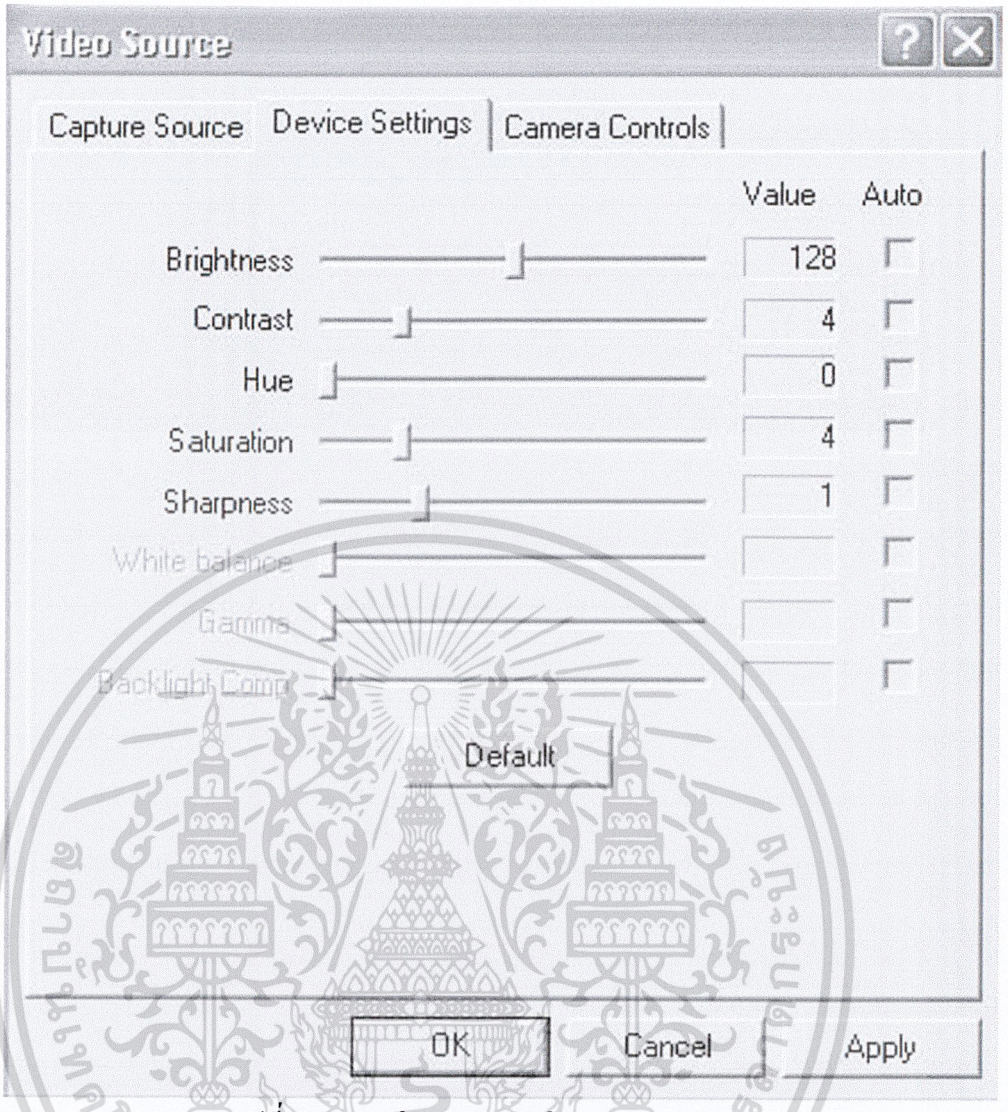
- ปุ่ม Capture Video ทำหน้าที่บันทึกภาพโดยภาพที่ถูกบันทึกนั้นในช่วงแรกจะถูกบันทึกอยู่ในชื่อ Capture.avi ที่ไดร์ฟซี แล้วเมื่อใส่ชื่อไฟล์ที่จะเซฟแล้วจริงทำการเปลี่ยนชื่อและเปลี่ยนที่อยู่ของ ไฟล์ให้ที่หลังที่กดเซฟ
- ปุ่ม Format DLG ทำหน้าที่เลือกรูปแบบการแสดงภาพออกหน้าจอว่าจะให้เป็นแบบมาตรฐานใดและกำหนดค่าต่างๆในการส่งภาพออกหน้าจอโดยสามารถเลือกทั้งขนาดของหน้าจอ และจำนวนพิกเซลต่อเฟรม
- ปุ่ม Display DLG เมื่อไม่มีภาพปรากฏบนหน้าจอให้กดปุ่มนี้เพื่อแสดงภาพ
- ปุ่ม Source DLG ทำหน้าที่ต่างๆเกี่ยวข้องกับอุปกรณ์ที่ใช้ส่งสัญญาณภาพ สามารถเลือกอุปกรณ์ส่งสัญญาณที่ต้องการจะใช้งาน และกำหนดค่าต่างๆการแสดงภาพออกหน้าจอ เช่น ความสว่างหรือความคมชัด ให้กับอุปกรณ์ตัวนั้นๆได้
- ปุ่ม Comp DLG ทำหน้าที่เลือกชนิด และปรับแต่งค่าในการบีบอัดข้อมูลภาพและเสียง
- ปุ่ม Audio DLG ทำหน้าที่ปรับค่าและตั้งค่าเกี่ยวกับเสียง
- ปุ่ม Set Width ทำหน้าที่ ปรับขนาดด้านกว้างของภาพบนหน้าจอ
- ปุ่ม Set Hight ทำหน้าที่ ปรับขนาดด้านแนวตั้งของภาพบนหน้าจอ
- ปุ่ม Save AVI ทำหน้าที่เก็บบันทึกภาพเคลื่อนไหวในชื่อไฟล์ที่ต้องการ และที่อยู่ที่ต้องการ
- ปุ่ม Save DBI ทำหน้าที่เก็บบันทึกภาพนิ่งในชื่อไฟล์ที่ต้องการ และที่อยู่ที่ต้องการ



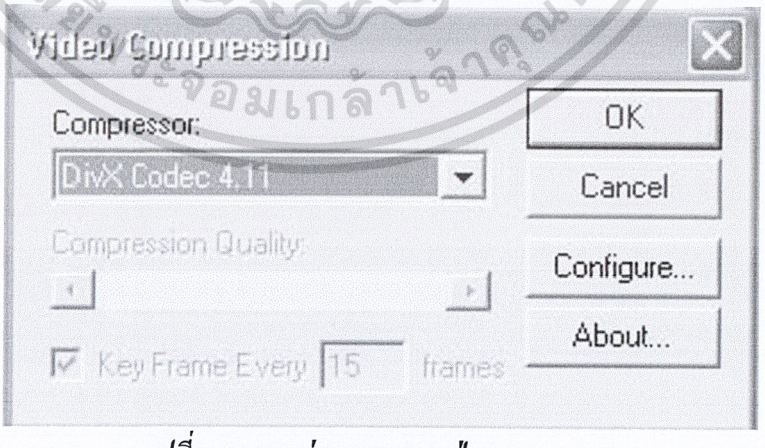
รูปที่ 7.3 แสดงส่วนขยายของปุ่ม Video DLG



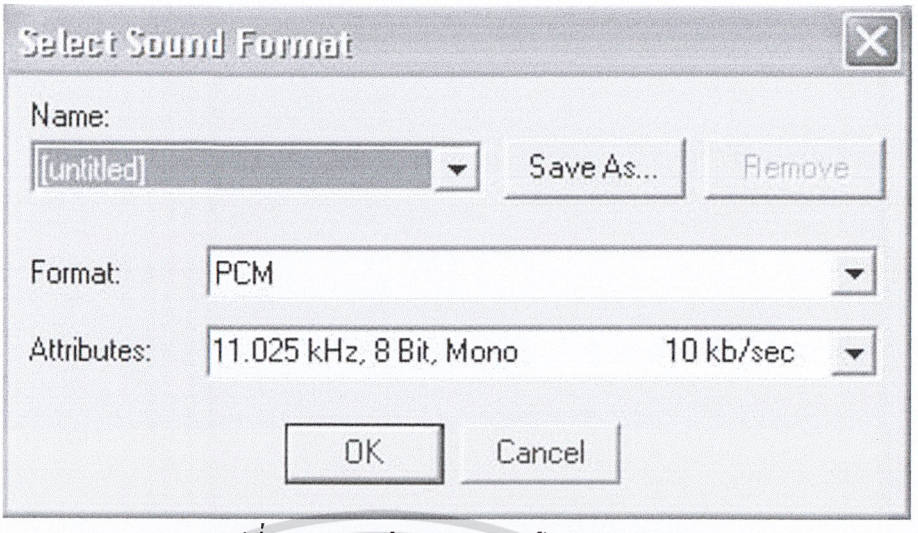
รูปที่ 7.4 แสดงส่วนขยายของปุ่ม Source DLG (1)



รูปที่ 7.5 แสดงส่วนขยายของปุ่ม Source DLG (2)



รูปที่ 7.6 แสดงส่วนขยายของปุ่ม Comp DLG



รูปที่ 7.7 แสดงส่วนขยายของปุ่ม Audio DLG

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 8
บทสรุปและวิจารณ์

โครงงานนี้เกี่ยวกับการควบคุมหุ่นยนต์แบบไร้สายผ่านคอมพิวเตอร์โดยใช้อุปกรณ์สื่อสารไร้สายเป็นตัวส่งข้อมูลในการควบคุมการทำงานระหว่างคอมพิวเตอร์กับหุ่นยนต์ และส่งสัญญาณภาพกลับมาแสดงออกทางหน้าจอ เพื่อใช้สำหรับรักษาความปลอดภัยจากระยะไกล

จากการดำเนินงาน โปรแกรมติดต่อกับผู้ใช้ ทำงานได้อย่างถูกต้อง สามารถส่งข้อมูลควบคุมในการทำงานของหุ่นยนต์ได้ สามารถรับสัญญาณภาพจากกล้องแล้วแสดงออกทางหน้าจอได้อย่างชัดเจน และส่วนควบคุมการทำงานภาคส่งก็สามารถตอบสนองการทำงานได้อย่าง ถูกต้อง แต่ยังมีปัญหาในการใช้งานอยู่บ้าง คือ ใช้งานได้ยาก ส่วนวงจรควบคุมการเคลื่อนที่ของหุ่นยนต์ หุ่นยนต์สามารถเคลื่อนที่ เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ได้อย่างถูกต้องตามโปรแกรมในส่วนควบคุมการเคลื่อนที่

แนวทางในการแก้ไขปัญหา คือ แก้ไขโปรแกรมในส่วนติดต่อกับผู้ใช้ใหม่ ให้ง่ายต่อการใช้งานมากขึ้น และถ้าโปรแกรมส่วนติดต่อกับผู้ใช้สามารถใช้งานได้ง่ายมากขึ้น ก็จะทำให้สามารถควบคุมการทำงานของหุ่นยนต์ได้อย่างถูกต้องมากขึ้นไปด้วย

แนวทางในการพัฒนาโครงงาน คือ เขียนโปรแกรมในส่วนของการควบคุมผ่านอินเตอร์เน็ต เพื่อให้สามารถควบคุมหุ่นยนต์ได้ในระยะไกล ติดตั้งอุปกรณ์ชาร์ตพลังงานไฟฟ้าอัตโนมัติเพิ่มให้กับหุ่นยนต์ และทำการติดตั้งอุปกรณ์เสริม เช่น แขนกล เพื่อเพิ่มความสามารถในการทำงานของหุ่นยนต์

บรรณานุกรม

[1] สัจจะ จรัสรุ่งรวีวร: “คู่มือการเขียนโปรแกรมและใช้งาน VISUAL BASIC 6”, อินโฟเพรส, 2544

[2] สมศักดิ์ ศรีขจรเกียรติ : “Advanced visual basic 6”, บิบลิโอไฟล์ พับลิชชิ่ง, 2544

[3] วรานันต์ วงศ์วิศว์ : “รู้ลึกรู้จริง Microsoft Visual Basic 6 Expert Edition”, สำนักพิมพ์ เอส.พี.ซี.บุ๊คส์, 2/2543

[4] Guy Eddon , Henry Eddon : “การพัฒนาคอมโพเนนต์ ด้วย Microsoft Visual Basic 6.0”, Microsoft Press, 2544

[5] รศ. ยืน ภู่วรวรรณ : “ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์”, ซีเอ็ดยูเคชั่น, 2538

[6] ชัชวาล ศุภเกษม : “ไขปัญหา RS-232 วิธีการใช้พอร์ตอนุกรมอย่างละเอียด”, ซีเอ็ดยูเคชั่น, 2538

[7] พิพัฒน์ เลาหสงคราม : “ ไมโครคอนโทรลเลอร์ MCS-48 MCS-51”, ซีเอ็ดยูเคชั่น, 2/2537

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง