

รถสำรวจผ่านระบบปฏิบัติการแอนดรอยด์
THE EXPLORER CAR VIA ANDROID APPLICATION

โดย

นายยุทธพงษ์	คงศิริ
นายวัชร	จันทร์ขุ่ม
นายสร้อยยศ	แย้มขุนทอง

ปฏิญญาพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

รถสำรวจผ่านระบบปฏิบัติการแอนดรอยด์
THE EXPLORER CAR VIA ANDROID APPLICATION

โดย
นายยุทธพงษ์ คงศิริ
นายวัชร จันทร์ชุ่ม
นายสร้อยศ แยมขุนทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

รถสำรวจผ่านระบบปฏิบัติการแอนดรอยด์
THE EXPLORER CAR VIA ANDROID APPLICATION

โดย

นายยุทธพงษ์ คงศิริ	53011324
นายวัชร จันทร์ชุ่ม	53011446
นายสร้อยยศ แยมขุนทอง	53011647

อาจารย์ที่ปรึกษา
ผศ.ดร. สมเกียรติ ฤกษ์วัลญญ
รศ.ดร. พรชัย ทรัพย์นิธิ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ผ่านการตรวจรูปเล่มแล้ว
.....
อาจารย์ที่ปรึกษา
6/3/57
วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว
.....
กรรมการผู้ตรวจชิ้นงาน
17/01/2557
วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รถสำรวจผ่านระบบปฏิบัติการแอนดรอยด์

THE EXPLORER CAR VIA ANDROID APPLICATION

ผู้จัดทำ

- | | |
|--------------------------|----------|
| 1. นายยุทธพงษ์ คงศิริ | 53011324 |
| 2. นายวัชร จันทร์ขุ่ม | 53011446 |
| 3. นายสร้อยยศ แย้มขุนทอง | 53011647 |

.....
(ผศ.ดร. สมเกียรติ ฤกษ์วีระบุญ)

อาจารย์ที่ปรึกษา

.....
(รศ.ดร. พรชัย ทรัพย์นิธิ)

อาจารย์ที่ปรึกษาร่วม

กิตติกรรมประกาศ

โครงการสำเร็จลุล่วงไปได้ด้วยดี ได้รับความร่วมมือและช่วยเหลือจากบุคคลหลายท่าน ผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษา ผศ.ดร. สมเกียรติ ฤกษ์วีระบุญ และ รศ.ดร. พรชัย ทรัพย์นิธิ รวมทั้งพี่ๆ เพื่อนๆ ทุกคนที่คอยให้คำปรึกษา ชี้แนะเกี่ยวกับความรู้ ทฤษฎี และขั้นตอนการทำงานต่างๆ เป็นอย่างดี ผู้จัดทำพึงระลึกเสมอว่าโครงการนี้ไม่สามารถสำเร็จได้ หากขาดความช่วยเหลือจากบุคคลทุกท่านที่ได้กล่าวมา ขอขอบคุณอย่างยิ่ง

นายยุทธพงษ์ คงศิริ

นายวัชร จันทร์ชุม

นายสร้อยศ แยมขุนทอง

ผู้จัดทำ

รถสำรวจผ่านระบบปฏิบัติการแอนดรอยด์
THE EXPLORER CAR VIA ANDROID APPLICATION

โดย	นายยุทธพงษ์ คงศิริ	53011324
	นายวัชร จันทร์ขุ่ม	53011446
	นายสร้อยยศ แยมขุนทอง	53011647

อาจารย์ที่ปรึกษา ผศ.ดร. สมเกียรติ ฤกษ์วีระบุญ
รศ.ดร. พรชัย ทรัพย์นิธิ

บทคัดย่อ

โครงการนี้เป็นการสร้างโปรแกรมแอนดรอยด์ที่ใช้บนโทรศัพท์มือถือสมาร์ตโฟน เพื่อควบคุมการทำงานของรถสำรวจด้วยวิธีการสื่อสารแบบไร้สาย ระบบประกอบด้วย โปรแกรมที่ถูกสร้างขึ้นบนโทรศัพท์มือถือ โมดูลไร้สาย ไมโครคอนโทรลเลอร์ และวงจรควบคุมมอเตอร์ รถจะถูกสั่งการด้วยโปรแกรมแอนดรอยด์และรับสัญญาณไร้สาย เข้าสู่อุปกรณ์ภาครับสัญญาณ และเข้าสู่ไมโครคอนโทรลเลอร์เพื่อประมวลผล จากนั้นวงจรควบคุมมอเตอร์จะรับสัญญาณจากไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของรถให้เคลื่อนที่สำรวจสภาพแวดล้อมที่มนุษย์ไม่สามารถเข้าถึงได้

ABSTRACT

This project is an android application creation on a smart phone in order to control execution of the car by means of wireless communication. The system comprise an android application which is created on a smart phone, a wireless module, a microcontroller and a motor controlling circuit. The car is stimulated through an android application and receive signals in term of wireless form to a receiver. Finally, a microcontroller would process and then a motor controlling circuit would receive signals from a microcontroller to control the car to survey around inaccessible areas.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VI
บทที่ 1	1
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	2
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 มอเตอร์กระแสตรง	2
2.2 โพรโตคอล TCP/IP	8
2.3 Wireless LAN (WLAN)	15
2.4 การพัฒนา Application บนระบบปฏิบัติการแอนดรอยด์	20
2.5 Microcontroller	25
2.6 หลักการส่งข้อมูลแบบอนุกรม	34
2.7 Application กล้องที่ส่งภาพจากโทรศัพท์มือถือ Smart Phone ไปยัง โทรศัพท์มือถือ Smart Phone ที่ใช้บังคับ	39
2.8 หลักการ Multithread ในโปรแกรมภาษา JAVA	44
2.9 ระบบอัลตราโซนิก (Ultrasonic)	47
2.10 Accelerometer	58
บทที่ 3	62
การออกแบบและการจัดทำปริญญานิพนธ์	62
3.1 การออกแบบ	62
3.2 เครื่องมือที่ใช้ในการทดลอง	75
3.3 การจัดเก็บผลการทดลอง	76

สารบัญ (ต่อ)

		หน้า
บทที่ 4	ผลการทดลอง	79
	4.1 การทดสอบการส่งข้อมูลระหว่างโทรศัพท์มือถือ Smart Phone สองเครื่องผ่าน Application	79
	4.2 การตั้งค่าโมดูล Wi-Fi ผ่านโปรแกรม Putty	81
	4.3 การทดสอบการส่งข้อมูลโดยดูผลผ่านโปรแกรม Hyperterminal	82
	4.4 ผลการวัดสัญญาณที่ขา T_x ของโมดูล Wi-Fi	82
	4.5 ผลการจัดเก็บสัญญาณ Input และ Output ของ IC 298N	87
	4.6 การควบคุมมอเตอร์โดยใช้ Application บนโทรศัพท์มือถือ Smart Phone	90
	4.7 ผลการทดลองรับ-ส่งข้อมูลผ่านโปรแกรม Putty	91
	4.8 ผลการทดลองวัดสัญญาณ Input/Output ของ Ultrasonic Sensor	92
	4.9 ผลการทดลองวัดสัญญาณ Output ที่ระยะทางต่างๆ ของ Ultrasonic Sensor	94 95
	4.10 ผลการทดลองการวัดระยะทางเมื่อเปรียบเทียบระหว่างการวัดโดย Microcontroller กับการวัดโดยตลับเมตร	96
	4.11 ผลการทดลองกราฟความสัมพันธ์ระหว่างค่าความเร่งของโมดูล Accelerometer กับแรงดันทางไฟฟ้า	101
	4.12 จัดเก็บผลการทดลองวัดค่ามุมเอียงที่วัดได้จริงโดยใช้ Protractor เปรียบเทียบกับค่าความเอียงที่วัดได้จาก Microcontroller แสดงผลทาง Hyperterminal	102
	4.13 จัดเก็บระยะการรับ-ส่งสัญญาณระหว่างโมดูล Wi-Fi กับโทรศัพท์มือถือ Smart Phone และระยะรับ-ส่งภาพระหว่างโทรศัพท์มือถือ Smart Phone ทั้งสองเครื่องผ่าน Router	102
	4.14 การตั้งค่า Router เพื่อให้สามารถเชื่อมต่อภายในวง LAN เดียวกัน	
	4.15 การทดสอบ Application ในการรับข้อมูลจาก Sensor เมื่อเข้าถึงระยะที่กำหนด	
บทที่ 5	สรุปผลและข้อเสนอแนะ	105
	5.1 สรุปผล	105

สารบัญ (ต่อ)

	หน้า
5.2 ข้อเสนอแนะ	105
บรรณานุกรม	106
ภาคผนวก อุปกรณ์ที่ใช้	107

สารบัญรูป

รูปที่	หน้า	
2.1	เฟรมของมอเตอร์	2
2.2	ขดลวดพันรอบขั้วแม่เหล็ก	3
2.3	ส่วนของโรเตอร์ของมอเตอร์	4
2.4	แปรงถ่าน	4
2.5	การทำงานของมอเตอร์	5
2.6	วิธีการใช้กฎมือซ้ายเพื่อหาทิศในการหมุนของมอเตอร์	5
2.7	การทำงานของ H-Bridge Switching	6
2.8	วงจรมอเตอร์คอนเวอร์เตอร์ (Boost Converter Circuit)	7
2.9	ขั้นตอนการ Encapsulation และ Demultiplexing	9
2.10	โครงสร้าง TCP/IP	10
2.11	IP Header	11
2.12	ICMP Header	13
2.13	UDP Header	13
2.14	TCP Header	14
2.15	แสดงการมอดูเลชันแบบ OFDM	19
2.16	ไมโครคอนโทรลเลอร์ Atmega168	25
2.17	รายละเอียดขาต่างๆ ของ Atmega168	26
2.18	บอร์ด ET BASE AVR EASY168	27
2.19	ระดับแรงดันของสัญญาณ TTL	35
2.20	การรับ-ส่งข้อมูล	36
2.21	สัญญาณของข้อมูลที่ถูส่งแบบอนุกรม	36
2.22	การส่งข้อมูลแบบ Asynchronous	37
2.23	Frame ข้อมูลที่ส่งในรูปแบบ Synchronous	38
2.24	การส่งข้อมูลแบบ Synchronous	38
2.25	ระดับแรงดันของสัญญาณ TTL และสัญญาณ RS232	39
2.26	โค้ดที่ใช้สร้าง, ทำลาย และเปลี่ยนแปลงการทำงานของ Surface Layout	40
2.27	คำสั่ง Permission ในการเข้าใช้งานกล้องบนอุปกรณ์แอนดรอยด์	40
2.28	คำสั่งที่ใช้ประกาศ Class เพื่อเข้าใช้กล้องและคำสั่งต่างๆ ที่เกี่ยวข้อง	41

สารบัญรูป (ต่อ)

รูปที่	หน้า	
2.29	คำสั่งที่ใช้ถ่ายภาพ	41
2.30	คำสั่งที่ใช้ในการ Focus กล้อง	42
2.31	คำสั่งที่ใช้ในการเปิด-ปิดแฟลชของกล้อง	42
2.32	คำสั่งที่ใช้จัดเก็บภาพถ่ายลงในหน่วยความจำภายนอก	42
2.33	คำสั่งที่ใช้ในการเชื่อมต่อ Socket (IP/Port)	43
2.34	คำสั่งที่ใช้ในการสร้างเขียนข้อมูลลง Buffer และส่งข้อมูลออกไป	43
2.35	การทำงานของโปรแกรมแบบ Multithread	45
2.36	ช่วงชีวิต (การทำงานของ) ของ Thread	47
2.37	การเขียนสัญลักษณ์ของ Ultrasonic Transducer แบบต่างๆ	48
2.38	หลักการการทำงานของ Ultrasonic Sensor	50
2.39	Ultrasonic Sensor วงจรส่ง/รับ	51
2.40	Ultrasonic Sensor ที่วงจรเวลาคงที่	52
2.41	Ultrasonic Sensor ที่วงจรเวลาเปลี่ยนแปลงได้	53
2.42	Ultrasonic Sensor บนพื้นผิวตรง	53
2.43	Ultrasonic Sensor กับการตรวจจับสิ่งของ	54
2.44	Ultrasonic Sensor กับการเบี่ยงเบนคลื่นเสียง	54
2.45	สัญญาณไฟฟ้ากระแสตรง, สัญญาณสามเหลี่ยม และสัญญาณพัลส์ที่มีความกว้างคงที่	56
2.46	โครงสร้างของ Accelerometer	58
2.47	การตรวจวัดการสั่นสะเทือนและตรวจวัดอัตราเร่งของวัตถุ	59
2.48	โครงสร้างพื้นฐานของ Seismic Mass Accelerometer	59
2.49	โครงสร้างพื้นฐานของ Piezoelectric Accelerometer	61
3.1	Block Diagram โดยรวมของระบบ	62
3.2	Flowchart การทำงานโดยรวมของ Application	63
3.3	Flowchart การทำงานในส่วนการควบคุมของ Application	64
3.4	Flowchart การทำงานในส่วนกล้องของ Application	65
3.5	ส่วนต่างๆ ของ Application	67
3.6	Flowchart แสดงการประมวลผลโดยไมโครคอนโทรลเลอร์	69
3.7	วงจรขั้วมอดูเตอร์โดยใช้บิตอินพุต 3 Bit	72

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.8	ทิศทางการเร่งในแนวแกน	73
3.9	วงจรรวมภาครับสัญญาณ	74
4.1	ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเดิหน้า	79
4.2	ข้อมูลที่ส่งออกมาเมื่อกดปุ่มถอยหลัง	80
4.3	ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเลี้ยวซ้าย	80
4.4	ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเลี้ยวขวา	81
4.5	การเซ็ทโหมดการเชื่อมต่อและชื่อของ SSID	81
4.6	การเซ็ทรหัสผ่านของเครือข่าย WLAN ที่ใช้กับโมดูล	81
4.7	การเซ็ทให้เป็น IP แบบ Static	82
4.8	การส่งข้อมูลโดยคุณผลผ่านโปรแกรม Hyperterminal	82
4.9	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถเดิหน้า (F)	83
4.10	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถถอยหลัง (B)	83
4.11	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหมุนซ้าย (L)	84
4.12	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหมุนขวา (R)	84
4.13	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหยุด (S)	85
4.14	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล่องหมุนซ้าย (Q)	85
4.15	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล่องหมุนขวา (T)	86
4.16	สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล่องหยุดหมุน (n)	86
4.17	สัญญาณ Input (v_{en}) เทียบกับสัญญาณ Output ที่ขา 13	87
4.18	สัญญาณ Input (v_{en}) เทียบกับสัญญาณ Output ที่ขา 14	87
4.19	สัญญาณ Input (C) เทียบกับสัญญาณ Output ที่ขา 13	88
4.20	สัญญาณ Input (C) เทียบกับสัญญาณ Output ที่ขา 14	88

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.21 สัญญาณ Input (D) เทียบกับสัญญาณ Output ที่ขา 13	89
4.22 สัญญาณ Input (D) เทียบกับสัญญาณ Output ที่ขา 14	89
4.23 การหมุนของมอเตอร์เมื่อส่งผ่านโทรศัพท์มือถือ Smart Phone	90
4.24 การส่งข้อมูลจากโมดูล Wi-Fi ไปยังคอมพิวเตอร์เครื่องที่ 2	91
4.25 การรับข้อมูลที่คอมพิวเตอร์เครื่องที่ 1 และเครื่องที่ 2	91
4.26 การรับ-ส่งข้อมูลไปยัง Application	92
4.27 สัญญาณจาก Oscilloscope ที่ขาอินพุตและเอาต์พุตของ Ultrasonic Sensor	93
4.28 สัญญาณจาก Oscilloscope ที่ระยะทางต่างๆ จากขาเอาต์พุตของ Ultrasonic Sensor	95
4.29 กราฟความสัมพันธ์ระหว่างความเร่ง g_x กับ V(Volts) ในระนาบ X-Z	97
4.30 กราฟความสัมพันธ์ระหว่างความเร่ง g_z กับ V(Volts) ในระนาบ X-Z	98
4.31 กราฟความสัมพันธ์ระหว่างความเร่ง g_y กับ V(Volts) ในระนาบ Y-Z	99
4.32 กราฟความสัมพันธ์ระหว่างความเร่ง g_z กับ V(Volts) ในระนาบ Y-Z	100
4.33 การใส่ Username และ Password สำหรับเข้าไปโหมดตั้งค่า	103
4.34 การตั้งค่า SSID และ Password ของ Router	103
4.35 การตั้งค่า IP ให้เป็น Static เพื่อให้โมดูล Wi-Fi สามารถเชื่อมต่อได้	103
4.36 การแจ้งเตือนบนหน้าจอโทรศัพท์มือถือ Smart Phone เมื่อถึงระยะที่กำหนด	104
4.37 ภาพที่ถูกส่งมาจากโทรศัพท์มือถือ Smart Phone	104

สารบัญตาราง

ตารางที่	หน้า
2.1 ความหมายในแต่ละ Bit ของ Flag	15
2.2 คุณสมบัติของมาตรฐาน IEEE ชนิดต่างๆ	18
3.1 แสดงบิต Input 3 Bit ในการควบคุมมอเตอร์ 1 ตัว	71
4.1 เปรียบเทียบระหว่างระยะทางจากการวัดและคาบเวลาของ Pulse	94
4.2 การทดลองวัดระยะทางการทำงานของ Ultrasonic Sensor	96
4.3 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน X (g_x) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ X-Z	97
4.4 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน Z (g_z) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ X-Z	98
4.5 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน Y (g_y) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ Y-Z	99
4.6 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน Z (g_z) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ Y-Z	100
4.7 การทดลองวัดมุมเอียง	101
4.8 การทดลองวัดระยะทางไกลสุด	102

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันมีการเกิดภัยพิบัติทางธรรมชาติมากมาย โดยที่ไม่สามารถพยากรณ์หรือคาดเดาได้ว่าจะเกิดที่ใด แม้แต่พื้นที่ที่ไม่เคยเกิดขึ้นมาก่อนก็ยังสามารถเกิดขึ้นได้ และด้วยสาเหตุที่ไม่สามารถคาดเดาได้นั้น จึงยากแก่การเตรียมการรับมือของเจ้าหน้าที่และประชาชนในพื้นที่เสี่ยงภัย ทำให้เกิดปัญหาที่ตามมาคือ เจ้าหน้าที่ไม่สามารถเข้าไปให้ความช่วยเหลือได้ เพราะบางครั้งพื้นที่นั้นมีความอันตรายและไม่ทราบลักษณะของพื้นที่ จึงทำให้เจ้าหน้าที่หรือบุคคลที่เกี่ยวข้องไม่สามารถให้ความช่วยเหลือในพื้นที่ดังกล่าวได้

ดังนั้นเพื่อที่จะแก้ปัญหาดังกล่าวเราจึงได้คิดโครงการนี้ขึ้น เพื่อช่วยเหลือให้เจ้าหน้าที่สามารถสำรวจพื้นที่และวางแผนก่อนให้การช่วยเหลือผู้ประสบภัย อีกทั้งยังช่วยลดความเสี่ยงที่จะเดินทางเข้าไปยังพื้นที่ด้วยตนเองซึ่งอาจจะทำให้เกิดภัยซ้ำซ้อน โดยการนำรถบังคับที่เหมาะสมแก่การสำรวจภาคพื้นดินมาทำการประยุกต์เชื่อมต่อกับเครือข่าย WLAN ที่ปัจจุบันพบได้ทั่วไป แล้วบังคับผ่านโทรศัพท์มือถือระบบปฏิบัติการ Android เพื่อไปสำรวจพื้นที่ที่ไม่สามารถเข้าถึงได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ Microcontroller
2. เพื่อศึกษาการทำงานของระบบไร้สาย
3. เพื่อศึกษาการเขียนโปรแกรมลงในโทรศัพท์มือถือ Android เบื้องต้น
4. เพื่อใช้สำรวจพื้นที่ที่ไม่สามารถเข้าถึงได้
5. เพื่อศึกษาการทำงานของ Sensor

1.3 ขอบเขตของปริญญาณิพนธ์

1. สามารถบังคับรถผ่านอุปกรณ์ที่ใช้ระบบปฏิบัติการ Android ได้
2. สามารถใช้กล้องเพื่อตรวจสอบดูสภาพแวดล้อมที่เคลื่อนที่ผ่านไปได้
3. มี Sensor ป้องกันการชนที่อาจก่อให้เกิดความเสียหายแก่อุปกรณ์ได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 มอเตอร์กระแสตรง

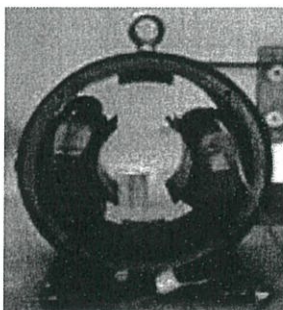
มอเตอร์ คือ เครื่องกลไฟฟ้าที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล โดยสร้างมอเตอร์ให้เหมือนกับเครื่องกำเนิดไฟฟ้ากระแสตรงทุกอย่าง จะมีข้อแตกต่างออกไปบ้างก็เพียงเล็กน้อยเท่านั้น ทั้งนี้เพราะว่าสภาพที่นำมาใช้งานแตกต่างกัน ตัวอย่างเช่น เครื่องกำเนิดไฟฟ้าทั่วไปจะเป็นชนิดเปิด (Open type) กล่าวคือขดลวดอาร์เมเจอร์และขดลวดสนามแม่เหล็กจะพันเป็นแบบเปิดทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดความเสียหายขึ้นกับขดลวด อย่างไรก็ตามเครื่องกลไฟฟ้ากระแสตรงเครื่องเดียวสามารถใช้ทำเป็นเครื่องกำเนิดไฟฟ้าหรือเป็นมอเตอร์ไฟฟ้าได้ โดยมอเตอร์จะมีอยู่หลายชนิดมาก ในที่นี้จะกล่าวถึงมอเตอร์กระแสตรงเท่านั้น

2.1.1 โครงสร้างและส่วนประกอบของมอเตอร์กระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงมีส่วนประกอบที่สำคัญ 2 ส่วน ดังนี้

2.1.1.1 ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator) ประกอบด้วย

1) เฟรมหรือโยค (Frame or Yoke) เป็นโครงสร้างภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆ ให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาเป็นรูปทรงกระบอก



รูปที่ 2.1 เฟรมของมอเตอร์

2) ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วน คือ แกนขั้วแม่เหล็ก และขดลวดส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์ เรียกว่า ขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มาก เป็นการทำให้มอเตอร์มีกำลังหมุน (Torque) ส่วนที่สองขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆ แกนขั้วแม่เหล็ก ขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้นและเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น

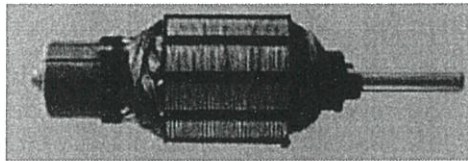


รูปที่ 2.2 ขดลวดพันรอบขั้วแม่เหล็ก

3) ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์ ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

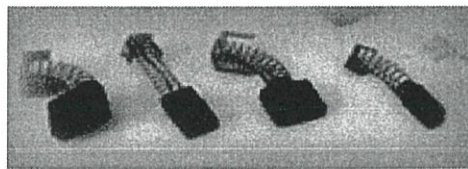
- (1) แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์ แกนเพลลานี้จะวางอยู่บนแบร์ริง เพื่อบังคับให้หมุนอยู่ในแนวนิ่งไม่มีการสั่นสะเทือนได้
- (2) แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด (Torque)
- (3) คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่ แต่ละซี่มีฉนวนไมก้า (Mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสายของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้ยึดแน่นติดกับแกนเพลลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยังขดลวดอาร์มาเจอร์ เพื่อสร้างเส้นแรงแม่เหล็ก อีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็กดังกล่าว เรียกว่า ปฏิกริยามอเตอร์ (Motor action)

(4) ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่อง (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้น ขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ ควรศึกษาต่อไปในเรื่องการพันอาร์มาเจอร์ (Armature Winding) ในโอกาสต่อไป



รูปที่ 2.3 ส่วนของโรเตอร์ของมอเตอร์

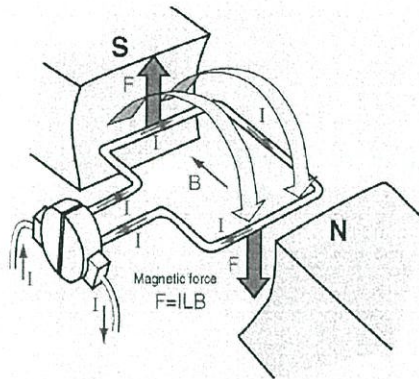
4) แปรงถ่าน (Brush) คือ ตัวสัมผัสกับคอมมิวเตเตอร์ ทำเป็นแท่งสี่เหลี่ยมผลิตจากคาร์บอนหรือแกรไฟต์ผสมผงทองแดงเพื่อให้แข็งและนำไฟฟ้าได้ดี มีสายตัวนำต่อร่วมกับแปรงถ่านเพื่อไปรับแรงดันกระแสไฟฟ้าตรงที่จ่ายเข้ามา แปรงถ่านทำหน้าที่รับแรงดันกระแสไฟฟ้าตรงจากแหล่งจ่ายเพื่อจ่ายผ่านไปให้คอมมิวเตเตอร์



รูปที่ 2.4 แปรงถ่าน

2.1.2 หลักการทำงานของมอเตอร์กระแสตรง

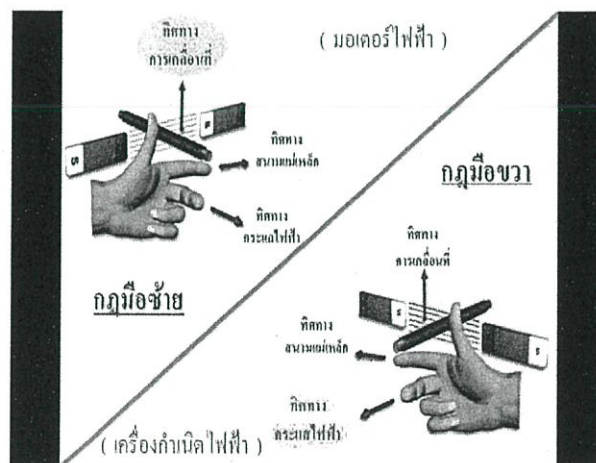
หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อป้อนแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะแปลงผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์เพื่อสร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวกันจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ และแกนเพลานี้จะสวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุนการที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์หรือโรเตอร์หมุนไปตามกฎมือซ้ายของเฟลมมิง (Fleming's left hand rule)



รูปที่ 2.5 การทำงานของมอเตอร์

2.1.2.1 กฎมือซ้ายของเฟลมมิง (Fleming's left hand rule)

ใช้สำหรับหาทิศทางของการเคลื่อนที่ของตัวนำในสนามแม่เหล็กหรือการหาทิศทางการหมุนของมอเตอร์ คือ เมื่อป้อนทิศทางกระแสไฟฟ้าเข้าไปในลวดตัวนำ ซึ่งลวดตัวนำอยู่ในสนามแม่เหล็ก สามารถหาทิศทางของการเคลื่อนที่ของลวดตัวนำได้โดยใช้กฎมือซ้ายของเฟลมมิงโดยยกมือซ้ายขึ้นมาแล้วให้นิ้วหัวแม่มือ นิ้วชี้ และนิ้วกลางของมือซ้ายวางตั้งฉากซึ่งกันและกัน ตามรูปที่ 2.6 กำหนดให้นิ้วชี้แทนทิศทางของสนามแม่เหล็ก นิ้วกลางแทนทิศทางของกระแสไฟฟ้า และนิ้วหัวแม่มือจะเป็นทิศทางของแรงการเคลื่อนที่หรือทิศทางการหมุนของมอเตอร์



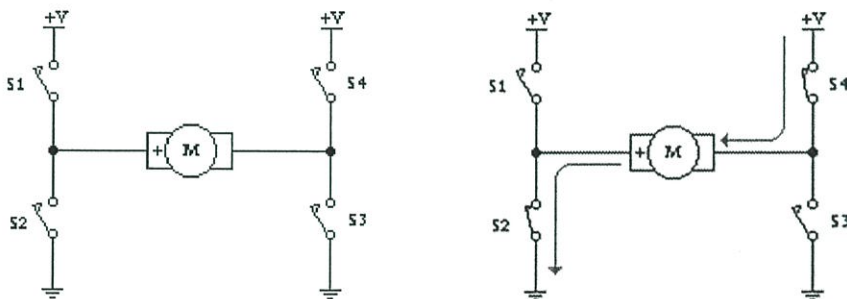
รูปที่ 2.6 วิธีการใช้กฎมือซ้ายเพื่อหาทิศในการหมุนของมอเตอร์

2.1.3 หลักการควบคุมมอเตอร์

การควบคุมมอเตอร์มีด้วยกันอยู่หลายวิธีแต่ในที่นี้จะกล่าวถึงเฉพาะหลักการที่ใช้ในการดำเนินงานเท่านั้น

2.1.3.1 ทฤษฎี H-Bridge Switching

หลักการของวงจรประกอบไปด้วยสวิทช์ 4 ตัว คือ S1, S2, S3 และ S4 ซึ่งในรูปแบบตัวอย่างจะใช้ดีซีมอเตอร์ (DC-Motor) เป็นโหลด (Load) ของวงจร ในสถานะเริ่มต้น สวิทช์ทุกตัวปิด (Off) อยู่ จะไม่มีอะไรเกิดขึ้นทั้งสิ้น เพราะไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์ และเมื่อทำการเปิด (On) สวิทช์ S1 และ S3 พร้อมกัน จะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้าไหลผ่านมอเตอร์จากขั้วบวกของมอเตอร์ไปยังขั้วลบของมอเตอร์ ทำให้มอเตอร์สามารถหมุนได้ในทิศทางไปข้างหน้า (Forward) จะหมุนแบบตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาขึ้นอยู่กับลักษณะของการพันขดลวดภายในมอเตอร์ และในทางกลับกันถ้าทำการเปิด (On) สวิทช์ S2 และ S4 พร้อมกัน จะเป็นการเชื่อมวงจรและทำให้เกิดกระแสไฟฟ้าไหลผ่านมอเตอร์จากขั้วลบของมอเตอร์ไปยังขั้วบวกของมอเตอร์ ทำให้มอเตอร์สามารถหมุนได้ในทิศทางกลับ (Reverse) ซึ่งจะกลับทิศทางกับกรณีแรก



รูปที่ 2.7 การทำงานของ H-Bridge Switching

2.1.3.2 การควบคุมความเร็วของมอเตอร์

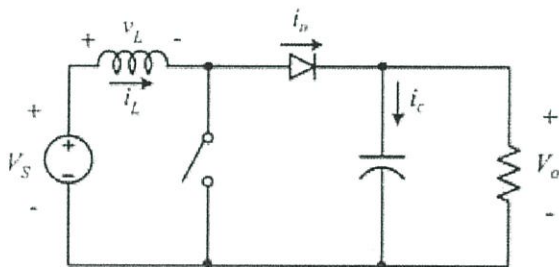
การควบคุมความเร็วของมอเตอร์สามารถทำได้โดยการควบคุมระดับแรงดันตกคร่อมตัวมอเตอร์และวิธีการที่นิยมใช้มีสามวิธี ดังนี้

1) การปรับระดับแรงดันโดยใช้หลักการเปลี่ยนค่าความต้านทานของวงจร ซึ่งทำให้แรงดันตกคร่อมเปลี่ยนไปด้วยแต่จะเห็นว่าวิธีนี้เป็น การสร้างความสูญเสียเนื่องจากกำลังไฟฟ้าตกคร่อมตัวต้านทาน (พลังงานไฟฟ้าเปลี่ยนเป็นพลังงานความร้อน)

2) การปรับระดับแรงดันโดยการเปิดปิดแหล่งจ่ายไฟ โดยระดับแรงดันเฉลี่ยที่ได้ขึ้นอยู่กับช่วงเปิด-ปิด เรียกว่าวิธีการนี้ว่า Pulse-Width Modulation และเรียกอัตราส่วนของช่วงเปิดและช่วงปิดว่า Duty cycle วิธีการนี้มีข้อเสียคือ หากเลือกความถี่ของพัลส์ไม่เหมาะสมอาจทำให้มอเตอร์สั่นได้

3) วงจรบูสต์คอนเวอร์เตอร์ (Boost Converter Circuit) คือ วงจรที่ทำการเพิ่มระดับแรงดันไฟฟ้า ซึ่งแรงดันไฟฟ้าด้านเอาต์พุตจะสูงกว่าแรงดันทางด้านอินพุต การวิเคราะห์การทำงานของวงจรบูสต์คอนเวอร์เตอร์ (Boost Converter Circuit) ในช่วงสภาวะอยู่ตัว จะมีการกำหนดเงื่อนไขเพื่อให้ง่ายต่อการวิเคราะห์ดังนี้

- (1) กระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำ ณ ตำแหน่งเดียวกันในแต่ละครั้งจะมีค่าเท่ากันและมีค่าเป็นบวกเสมอ
- (2) แรงดันไฟฟ้าเฉลี่ยตกคร่อมตัวเหนี่ยวนำในแต่ละคาบจะเท่ากับศูนย์
- (3) ตัวเก็บประจุต้องมีขนาดใหญ่พอทำให้แรงดันเอาต์พุตมีค่าคงที่
- (4) กำลังไฟฟ้าอินพุตเท่ากับกำลังไฟฟ้าด้านเอาต์พุต



รูปที่ 2.8 วงจรบูสต์คอนเวอร์เตอร์ (Boost Converter Circuit)

หลักการทำงานของวงจรบูสต์คอนเวอร์เตอร์เพื่อให้แรงดันด้านเอาต์พุตมีค่าตามต้องการและสามารถหาค่ากระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำได้โดยวิเคราะห์การทำงานของสวิตช์ในแต่ละโหมด จะเห็นได้ว่าการควบคุมความเร็วของมอเตอร์โดยปรับแรงดันตามค่าความต้านทานจะทำให้เกิดการสูญเสียพลังงาน (Power Loss) ในรูปของความร้อนสูง และการปรับแรงดันด้วยวงจรบูสต์คอนเวอร์เตอร์จะทำให้กระแสไฟฟ้าน้อยลงและทำให้แรงบิด (Torque) ของมอเตอร์ลดต่ำลงไปด้วย เนื่องจากกำลังไฟฟ้าด้านอินพุตเท่ากับกำลังไฟฟ้าด้านเอาต์พุต ดังนั้นจึงเลือกใช้วิธี Pulse-Width Modulation (PWM) เป็นตัวควบคุม และความกว้างของ Duty cycle เป็นตัวควบคุมความเร็วของมอเตอร์ ซึ่งเป็นวิธีที่ง่ายและมีประสิทธิภาพมากที่สุด

2.2 โพรโทคอล TCP/IP

2.2.1 ประวัติความเป็นมา

TCP/IP เป็นมาตรฐานของการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์ 2 ระบบที่มีจุดเริ่มต้นราว 30 ปีที่แล้ว เมื่อกระทรวงกลาโหมสหรัฐฯ ได้ทำการทดลองในปี ค.ศ. 1969 โดยเชื่อมโยงคอมพิวเตอร์ทางการทหารของแต่ละหน่วยให้สามารถติดต่อรับ-ส่งข้อมูลกันได้ รวมถึงใช้บริการอื่นๆ เช่น Remote, Login รวมถึงการรับ E-mail จุดประสงค์ของการทดลองนี้เพื่อใช้ในการส่งข้อมูลได้ แม้จะมีความเสียหายกับสายส่งข้อมูลบางส่วน ซึ่งเป็นคุณสมบัติที่จำเป็นมากในยามศึกสงคราม ในขณะนั้นกองทัพของสหรัฐฯ ทั้ง 3 เหล่า ที่ใช้คอมพิวเตอร์คนละระบบกันทำให้ไม่สามารถเชื่อมต่อสื่อสารกันเป็นเครือข่ายได้ จึงได้ให้ทุนเพื่อทำการคิดค้นการส่งข้อมูลระหว่าง 3 เครือข่าย โดยทำการส่งข้อมูลออกเป็นส่วนย่อยๆ เรียกว่า Packet ข้อมูลแต่ละส่วนจะถูกส่งไปให้คอมพิวเตอร์ฝั่งรับผ่านสายส่งข้อมูล คอมพิวเตอร์ปลายทางจะนำข้อมูลมาต่อรวมกันตามลำดับจนครบ หากเส้นทางที่ส่งข้อมูลเสียหายหรือเครื่องคอมพิวเตอร์บางส่วนเสียหาย ข้อมูลจะถูกร้องขอให้ส่งใหม่ โครงการนี้มีชื่อว่า Advanced Research Projects Agency Network (ARPANET) มีส่วนประกอบอยู่ 2 ส่วน คือ Transmission Control Protocol (TCP) และ Internet Protocol (IP) ซึ่ง TCP มีหน้าที่ตรวจสอบการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่งให้ได้ข้อมูลถูกต้องครบถ้วน หากมีการสูญหายจะแจ้งให้ต้นทางทำการส่งข้อมูลใหม่ ส่วน IP จะมีหน้าที่เลือกเส้นทางที่ใช้รับ-ส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบ Address โดยมีขนาดข้อมูล 4 ไบต์ (32 บิต) หลังจากนั้น TCP/IP ได้มีบทบาทกับการเชื่อมต่อคอมพิวเตอร์เป็นเครือข่าย จนกระทั่งกลายเป็นอินเทอร์เน็ตในปัจจุบัน

2.2.2 จุดประสงค์ TCP/IP

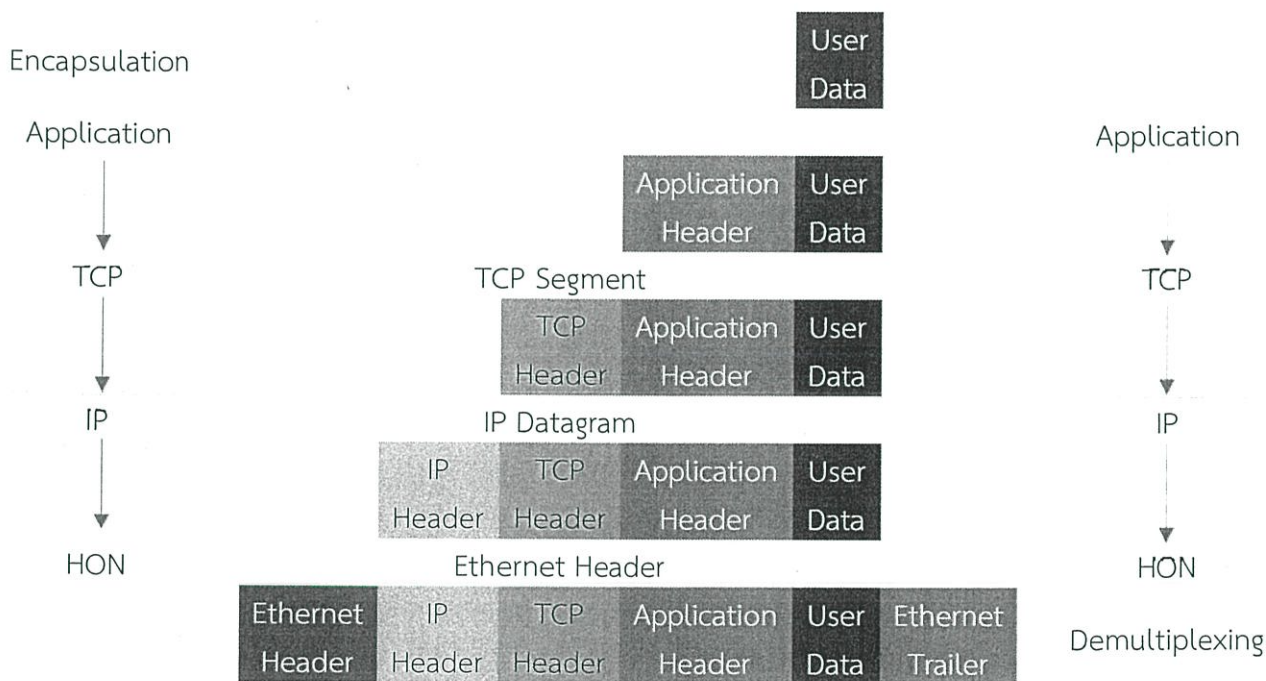
TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการ คือ

- 1) เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
- 2) ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกัน แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือการสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

3) มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิด ทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

2.2.3 Encapsulation/Demultiplexing

การส่งข้อมูลผ่านในแต่ละเลเยอร์จะทำการรวมข้อมูลที่รับมากับข้อมูลส่วนควบคุม ซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูล เรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูลจะเกิดกระบวนการทำงานย้อนกลับ คือ โปรโตคอลเดียวกับทางฝั่งผู้รับจะได้รับข้อมูลส่วนที่เป็น Header ก่อน แล้วนำไปประมวล เพื่อให้ทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



รูปที่ 2.9 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

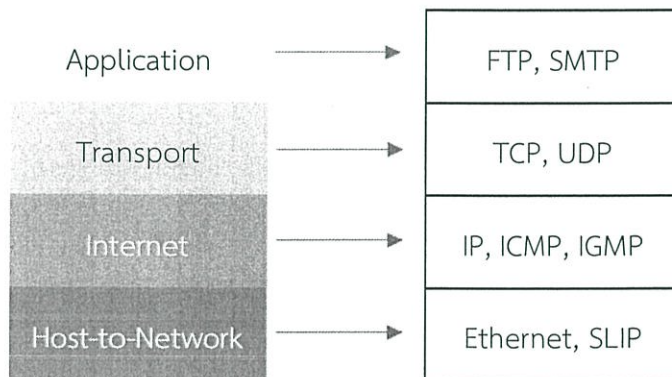
- 1) ข้อมูลที่มาจาก User หรือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data

2) เมื่อ Application ได้รับข้อมูลจาก User จะนำมารวมกับ Header ของ Application เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP

3) เมื่อโปรโตคอล TCP ได้รับ Application Data จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP

4) เมื่อโปรโตคอล IP ได้รับ TCP Segment จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer

5) ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ Flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป



รูปที่ 2.10 โครงสร้าง TCP/IP

2.2.4 โครงสร้าง TCP/IP

2.2.4.1 Host-to-Network Layer

โปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้ เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มา แล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทาง โดยผู้รับจะทำงานในทางกลับกันคือรับข้อมูลแล้วส่งให้กับโปรแกรมในชั้นสื่อสาร

2.2.4.2 Internet Layer

ใช้ประเภทของระบบการสื่อสารที่เรียกว่า Packet-Switching Network ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า

Packet สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบ จนถึงจุดหมายปลายทางได้โดยอิสระ หากมีการส่ง Packet ออกมาเป็นชุด โดยมีจุดหมายปลายทางเดียวกัน ระหว่างการเดินทางในเครือข่าย Packet แต่ละตัวในชุดนี้จะเป็นอิสระแก่กัน ดังนั้น Packet ที่ส่งไปถึงปลายทางอาจไม่เป็นไปตามลำดับได้

1) IP (Internet Protocol)

IP เป็นโปรโตคอลในระดับ Network Layer ทำหน้าที่จัดการเกี่ยวกับ Address และข้อมูล แล้วควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของ Packet ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุดและเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล มีระบบการแยกและประกอบ Datagram เพื่อรองรับการส่งข้อมูลระดับ Data link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet, Token Ring หรือ Apple Talk การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูลจะเป็นแบบ Connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆ ครั้งของการส่งข้อมูล 1 Datagram โดยไม่ทราบถึงข้อมูล Datagram ที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 Datagram อาจเกิดการส่งได้หลายครั้ง ในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (Fragmentation) และถูกนำไปรวมเป็น Datagram เดิมเมื่อถึงปลายทาง

4-bit Version	Header Length	8-bit Type of Service	16-bit Total Length in Byte	
16-bit Identification			3-bit Flag	16-bit Fragment Checksum
8-bit Time to Live (TTL)		8-bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Option				
Data				

รูปที่ 2.11 IP Header

Header ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม Option บางอย่าง Field ของ Header IP จะมีความหมายดังนี้

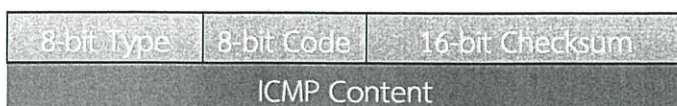
- (1) Version: ที่ใช้งานในปัจจุบันคือ Version 4 (IPv4) และ Version 6 (IPv6)
- (2) Header Length: โดยทั่วไปถ้าไม่มีส่วน Option จะมีค่าเป็น 5 (5*32 bits)

- (3) Type of Service (TOS): ใช้เป็นข้อมูลสำหรับ Router ในการตัดสินใจเลือกการ Route ข้อมูลในแต่ละ Datagram แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว
- (4) Length: ความยาวทั้งหมดเป็นจำนวน Byte ของ Datagram ซึ่งด้วยขนาด 16 bits ของ Field จะหมายถึงความยาวสูงสุดของ Datagram คือ 65535 byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆ ตามขนาดของ MTU ที่กำหนดใน Link Layer และนำมารวมอีกครั้ง เมื่อส่งถึงปลายทาง Application ส่วนใหญ่จะมีขนาดของ Datagram ไม่เกิน 512 byte
- (5) Identification: เป็นหมายเลขของ Datagram ในกรณีที่มีการแยก Datagram เมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี Identification เดียวกันมารวมกัน
- (6) Flag: ใช้ในกรณีที่มีการแยก Datagram
- (7) Fragment Offset: ใช้ในการกำหนดตำแหน่งข้อมูลใน Datagram ที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง
- (8) Time to Live (TTL): กำหนดจำนวนครั้งที่มากที่สุดที่ Datagram จะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่าง Network) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 ค่า เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และ Router สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด Time out ในระหว่างการส่งข้อมูล
- (9) Protocol: ระบุโปรโตคอลที่ส่งใน Datagram เช่น TCP, UDP หรือ ICMP
- (10) Header Checksum: ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน Header
- (11) Source IP Address: หมายเลข IP ของผู้ส่งข้อมูล
- (12) Destination IP Address: หมายเลข IP ของผู้รับข้อมูล
- (13) Data: ข้อมูลจากโปรโตคอลระดับบน

2) ICMP (Internet Control Message Protocol)

ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานะของ Datagram ในกรณีที่เกิดปัญหาเกี่ยวกับ Datagram เช่น Router ไม่สามารถส่ง Datagram ไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยัง Host ต้นทางเพื่อรายงานข้อผิดพลาดที่เกิดขึ้น แต่ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริง หากมีการส่ง Datagram ออกไปแล้วไม่มี ICMP Message บอก Error กลับมา มีความหมายได้ 2 กรณี คือ ข้อมูลถูกส่งไปถึงปลายทางเรียบร้อยแล้ว หรือมีปัญหาในการสื่อสารทั้งการส่ง Datagram และ ICMP Message ที่ส่งกลับมามีปัญหาระหว่างทาง ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Unreliable) ซึ่งเป็นหน้าที่ของโปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารมีความน่าเชื่อถือ

ในส่วนของ ICMP Message ประกอบด้วย Type ขนาด 8 bits, Checksum ขนาด 16 bits และส่วนของ Content ซึ่งมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป



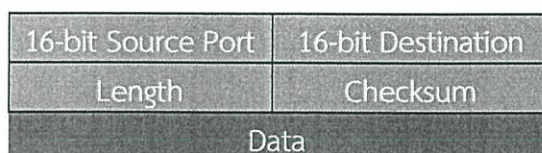
รูปที่ 2.12 ICMP Header

2.2.4.3 Transport Layer

แบ่งเป็นโปรโตคอล 2 ชนิด ดังนี้

1) UDP: (User Datagram Protocol) เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP จะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP ซึ่งจะไม่มีความสัมพันธ์กันระหว่าง Datagram และจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับ-ส่งข้อมูล

กลไกการตรวจสอบโดย Checksum ของ UDP เป็นการป้องกันข้อมูลที่อาจถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะรู้ว่ามียังข้อผิดพลาดเกิดขึ้น แต่จะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error จะให้ปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับ-ส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาด ในส่วนของ UDP จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด



รูปที่ 2.13 UDP Header

มีรายละเอียดดังนี้

- (1) Source Port Number: หมายเลขต้นทางที่ส่ง Datagram
- (2) Destination Port Number: หมายเลขปลายทางที่เป็นผู้รับ Datagram
- (3) UDP Length: ความยาวของ Datagram ทั้งส่วน Header และ Data หมายความว่า ค่าที่น้อยที่สุดใน Field คือ 8 ซึ่งเป็นขนาดของ Header
- (4) Checksum: เป็นตัวตรวจสอบความถูกต้องของ UDP Datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

(5) TCP: (Transmission Control Protocol) อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับ-ส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดย Datagram ของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับ-ส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นทางการ (Connection-Oriented)

16-bit Source Port Number								16-bit Source Destination Port	
32-bit Sequence Number									
32-bit Acknowledge Number									
Header Length	6-bit Reserved	URG	ACK	PUSH	RESET	SYN	FIN	16-bit Windows Size	
16-bit TCP Checksum								16-bit Urgent Pointer	
TCP Option									
Data									

รูปที่ 2.14 TCP Header

มีรายละเอียด ดังนี้

- (1) Source Port Number: หมายเลขต้นทางที่ส่ง Datagram
- (2) Destination Port Number: หมายเลขปลายทางที่เป็นผู้รับ Datagram
- (3) Sequence Number: Field ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับให้ถูกต้อง
- (4) Acknowledgment Number: ทำหน้าที่เช่นเดียวกับ Sequence Number แต่ใช้ในการตอบรับ
- (5) Header Length: โดยปกติความยาวของ Header TCP จะมีความยาว 20 Byte หรืออาจมากกว่านั้น แต่ต้องไม่เกิน 60 Byte
- (6) Flag: เป็นข้อมูลระดับ Bit ที่อยู่ใน Header TCP เป็นตัวบ่งบอกคุณสมบัติของ Packet TCP และใช้เป็นตัวควบคุมจังหวะการรับ-ส่งข้อมูล ซึ่ง Flag มีอยู่ทั้งหมด 6 bit แบ่งได้ดังนี้

ตารางที่ 2.1 ความหมายในแต่ละ Bit ของ Flag

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลใน Field Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (Reset) เนื่องจากเกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น Host มีปัญหาให้เริ่มสื่อสารใหม่ เป็นต้น
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

Flag ใน Header ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP Segment เนื่องจากข้อมูลใน Header ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งาน Field ไม่เหมือนกัน Flag จะเป็นตัวกำหนดว่าให้ใช้งาน Field ไหน เช่น Field Acknowledgment Number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลใน Field ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ถ้าไม่มี Flag เป็นตัวกำหนดอาจมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

2.2.4.4 Application Layer

มีโปรโตคอลสำหรับสร้างจอ Terminal เสมือน หรือเรียกว่า TELNET ช่วยให้ผู้ใช้สามารถติดต่อกับเครื่อง Host ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนว่ากำลังนั่งทำงานอยู่ที่เครื่อง Host โปรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP ช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องอื่น และโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบหรือรับข้อความที่มีผู้ส่งเข้ามา

2.3 Wireless LAN (WLAN)

ระบบเครือข่ายท้องถิ่นไร้สายหรือ WLAN หรือ Wireless LAN หรือ Wi-Fi คือระบบสื่อสารข้อมูลที่ส่งสัญญาณผ่านอากาศโดยใช้คลื่นวิทยุหรือคลื่นแม่เหล็กไฟฟ้า จึงไม่ต้องติดตั้งสายสัญญาณ ทำให้เครือข่ายนี้มีความยืดหยุ่นสูง สะดวกต่อการติดตั้งและใช้งาน ส่วนใหญ่จะนิยมติดตั้งเพิ่มเติมหรือแทนที่ระบบเครือข่ายท้องถิ่นแบบมีสาย (LAN) ระบบ WLAN จะใช้คลื่นวิทยุ (Radio Frequency) เป็นสัญญาณ และใช้อากาศเป็นตัวนำสัญญาณ ช่วยประหยัดค่าใช้จ่ายด้าน

สายนำสัญญาณ ปัจจุบันเครือข่ายไร้สายส่วนใหญ่สามารถรับ-ส่งข้อมูลได้ถึง 300 Mbps ซึ่งเพียงพอต่อเครื่อง Client ทั่วๆ ไป ข้อดีของระบบคือ ความสะดวกในการเคลื่อนย้าย และความง่ายต่อการขยายเครือข่าย แต่สิ่งที่ยังคงต้องพัฒนาต่อไป คือ เรื่องความปลอดภัยของข้อมูล ความเสถียร ระยะทางที่ครอบคลุม และความเร็ว

2.3.1 มาตรฐาน Wireless LAN IEEE 802.11

Wi-Fi หรือ เทคโนโลยีเครือข่ายแบบไร้สาย มาตรฐาน IEEE 802.11 ถือกำเนิดขึ้นในปี พ.ศ. 2528 จัดตั้งโดยองค์กร IEEE (สถาบันวิศวกรรมทางด้านไฟฟ้าและอิเล็กทรอนิกส์) มีความเร็ว 1 Mbps ในยุคเริ่มแรกนั้นให้ประสิทธิภาพการทำงานที่ค่อนข้างต่ำ ทั้งไม่มีการรับรองคุณภาพของการให้บริการที่เรียกว่า QoS (Quality of Service) และมาตรฐานความปลอดภัยต่ำ จากนั้นทาง IEEE จึงจัดตั้งคณะทำงานขึ้นมาปรับปรุงหลายกลุ่มด้วยกัน โดยกลุ่มที่มีผลงานเป็นที่น่าพอใจและได้รับการยอมรับอย่างเป็นทางการว่าได้มาตรฐาน ได้แก่ 802.11a, 802.11b และ 802.11g ส่วนมาตรฐานเกี่ยวกับการเข้ารหัสและคุณภาพของการให้บริการ (QoS) จะรวมอยู่ใน 802.11i และ 802.11e ซึ่งมีอธิบายได้ดังนี้

1) มาตรฐาน IEEE 802.11b

เสร็จสมบูรณ์เมื่อปี พ.ศ. 2542 ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ผสมกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับ-ส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุความถี่ 2.4 GHz (เป็นย่านความถี่ที่เรียกว่า ISM (Industrial Scientific and Medical) ซึ่งได้รับการจัดสรรไว้อย่างสากลสำหรับการใช้งานอย่างสาธารณะด้านวิทยาศาสตร์ อุตสาหกรรม และการแพทย์) โดยอุปกรณ์ที่ใช้ในย่านความถี่นี้ เช่น IEEE 802.11, Bluetooth, โทรศัพท์ไร้สาย และเตาไมโครเวฟ เป็นต้น มีระยะการส่งสัญญาณได้ไกลถึง 100 เมตร ปัจจุบันผลิตภัณฑ์หรืออุปกรณ์เครือข่ายไร้สายภายใต้มาตรฐานนี้ได้รับการผลิตออกมาเป็นจำนวนมาก และที่สำคัญแต่ละผลิตภัณฑ์มีความสามารถทำงานร่วมกันได้ อุปกรณ์ของผู้ผลิตทุกยี่ห้อต้องผ่านการตรวจสอบจากสถาบัน Wi-Fi Alliance เพื่อตรวจสอบมาตรฐานของอุปกรณ์และความเข้ากันได้ของแต่ละผู้ผลิต ปัจจุบันนิยมนำอุปกรณ์ WLAN ที่เป็นมาตรฐาน 802.11b ไปใช้ในองค์กรธุรกิจ, สถาบันการศึกษา, สถานที่สาธารณะ และกำลังแพร่เข้าสู่สถานที่พักอาศัยมากขึ้น มาตรฐานนี้มีระบบเข้ารหัสข้อมูลแบบ WEP ที่ 128 bit

2) มาตรฐาน IEEE 802.11a

เสร็จสมบูรณ์เมื่อปี พ.ศ. 2542 โดยออกเผยแพร่ช้ากว่าของมาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุง

ความเร็วในการส่งข้อมูลให้ได้สูงถึง 54 Mbps บนความถี่ 5 GHz ซึ่งจะมีคลื่นรบกวนน้อยกว่าความถี่ 2.4 GHz ที่มาตรฐานอื่นใช้กัน ที่ความเร็วนี้สามารถทำการแพร่ภาพและข่าวสารที่ต้องการความละเอียดสูงได้ อัตราความเร็วในการรับ-ส่งข้อมูลสามารถปรับระดับให้ช้าลงได้ เพื่อเพิ่มระยะทางการเชื่อมต่อให้มากขึ้น แต่ข้อเสียคือ ความถี่ 5 GHz นั้น หลายๆ ประเทศไม่อนุญาตให้ใช้ เช่น ประเทศไทย เพราะได้จัดสรรให้อุปกรณ์ประเภทอื่นไปแล้ว และยิ่งกว่านั้นระยะการส่งข้อมูลของ IEEE 802.11a ยังสั้นเพียง 30 เมตรเท่านั้น อีกทั้งอุปกรณ์ของ IEEE 802.11a ยังมีราคาสูงกว่า IEEE 802.11b ด้วย ดังนั้นอุปกรณ์ IEEE 802.11a จึงได้รับความนิยมน้อยกว่า IEEE 802.11b มาก จึงทำให้ไม่ค่อยเป็นที่ได้รับความนิยมเท่าที่ควร

3) มาตรฐาน IEEE 802.11g

เสร็จสมบูรณ์ในปี พ.ศ. 2546 ได้นำเอาเทคโนโลยี OFDM ของ 802.11a มาพัฒนาบนความถี่ 2.4 GHz ทำให้ใช้ความเร็ว 36-54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b ซึ่ง 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน มาตรฐานนี้เป็นที่ยอมรับจากผู้ใช้เป็นจำนวนมากและกำลังเข้ามาแทนที่ 802.11b ในอนาคต

4) มาตรฐาน IEEE 802.11n

เป็นมาตรฐานใหม่ที่ทาง Wi-Fi Alliance กำลังทดสอบ โดยคาดว่าจะมีความเร็วในการถ่ายโอนข้อมูลอยู่ที่ 74-248 Mbps หมายความว่ามีความเร็วกว่ารุ่นก่อนถึงประมาณ 5 เท่า นอกจากนี้ยังมีรัศมีทำการภายในอาคารที่ 70 เมตร และนอกอาคารที่ 160 เมตร เพิ่มความสามารถในการป้องกันสัญญาณก่อกวนจากอุปกรณ์อื่นๆ ที่ใช้ความถี่ 2.4 GHz เหมือนกัน และสามารถรองรับอุปกรณ์มาตรฐาน IEEE 802.11b และ IEEE 802.11g ได้ มาตรฐาน IEEE 802.11n นี้ได้เสร็จสมบูรณ์ในปี พ.ศ. 2552 แล้ว

5) มาตรฐาน IEEE 802.11ac

เป็นมาตรฐานล่าสุดที่ถูกพัฒนาขึ้นมาโดยเน้นไปที่การเพิ่มประสิทธิภาพของการรับ-ส่งคลื่น ทำให้ได้ความเร็วที่เพิ่มขึ้นจากเดิมมาก รวมไปถึงระยะที่ไกลขึ้นและสามารถทะลุผ่านสิ่งกีดขวางได้ดีขึ้น โดยใช้ความถี่ที่ย่าน 5 GHz โดยมีย่าน Bandwidth ที่กว้างขึ้นจากเดิม 40 MHz เป็น 80 MHz และ 160 MHz ซึ่งใช้หลักของ MIMO (Multiple-Input Multiple-Output) คือ การใช้เสามากกว่า 1 เสาในการรับ-ส่งสัญญาณ ยิ่งเสามากทำให้สัญญาณยิ่งแรงขึ้น ส่วนการ Modulation จะใช้ 256-QAM จากเดิมใช้ 64-QAM สำหรับ 802.11n

ตารางที่ 2.2 คุณสมบัติของมาตรฐาน IEEE ชนิดต่างๆ

มาตรฐาน	ความถี่ที่ใช้ (GHz)	Bandwidth (MHz)	อัตราการส่งข้อมูล สูงสุด (Mbps)	Modulation
IEEE 802.11	2.4	20	2	DSSS, FHSS
IEEE 802.11a	5	20	54	OFDM
IEEE 802.11b	2.4	20	11	DSSS
IEEE 802.11g	2.4	20	54	OFDM, DSSS
IEEE 802.11n	5	20/40	600	OFDM
IEEE 802.11ac	5	80/160	6930	256-QAM, OFDM

6) มาตรฐาน IEEE 802.11i

ได้ปรับปรุง MAC Layer ของ IEEE 802.11 ในด้านความปลอดภัย เนื่องจากเครือข่าย IEEE 802.11 WLAN มีช่องโหว่อยู่มาก โดยเฉพาะการเข้ารหัสข้อมูล (Encryption) ด้วย Key ที่ไม่มีการเปลี่ยนแปลง ทาง IEEE 802.11i นำเอาเทคนิคขั้นสูงมาใช้ในการเข้ารหัสข้อมูลด้วย Key ที่มีการเปลี่ยนค่าอยู่เสมอและการตรวจสอบผู้ใช้ที่มีความปลอดภัยสูง มาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุก Version ได้

7) มาตรฐาน IEEE 802.11e

ได้ปรับปรุง MAC Layer ของ IEEE 802.11 ให้สามารถรองรับการใช้งานหลักการ QoS (Quality of Service) สำหรับ Application เกี่ยวกับ Multimedia เนื่องจาก IEEE 802.11e เป็นการปรับปรุง MAC Layer ดังนั้นมาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุก Version ได้

2.3.2 การ Modulation แบบ Spread Spectrum

การส่งสัญญาณแบบ Spread Spectrum คือ การผสมผสานระหว่างเทคโนโลยี Narrowband และ Broadband กล่าวคือ ช่วงความถี่ที่ใช้จะกว้าง แต่แบ่งออกเป็นช่องสัญญาณย่อยหลายๆ ช่อง แล้วส่งสัญญาณโดยใช้ช่องสัญญาณทีละช่อง ระบบเครือข่ายไร้สายส่วนใหญ่จะใช้เทคโนโลยีนี้ จุดประสงค์ของการออกแบบ คือ การแลกเปลี่ยนกันระหว่างความเร็วกับความเชื่อถือ กล่าวคือ การส่งข้อมูลด้วยเทคโนโลยีนี้จะใช้ช่องความถี่ที่กว้างทำให้สัญญาณที่มีกำลังมากขึ้น ทำให้การรับสัญญาณง่ายขึ้น เทคโนโลยีนี้แบ่งออกเป็น 4 ประเภท คือ

1) Frequency-Hopping Spread Spectrum(FHSS)

เทคโนโลยีนี้จะใช้สัญญาณแบบ Narrowband แต่เปลี่ยนความถี่ตามลำดับที่ทั้งฝั่งรับและฝั่งส่งให้เข้าใจตรงกัน ซึ่งลำดับนี้ดูเหมือนกับเป็นแบบสุ่ม (Random) สำหรับสถานีอื่น

2) Direct-Sequence Spread Spectrum (DSSS)

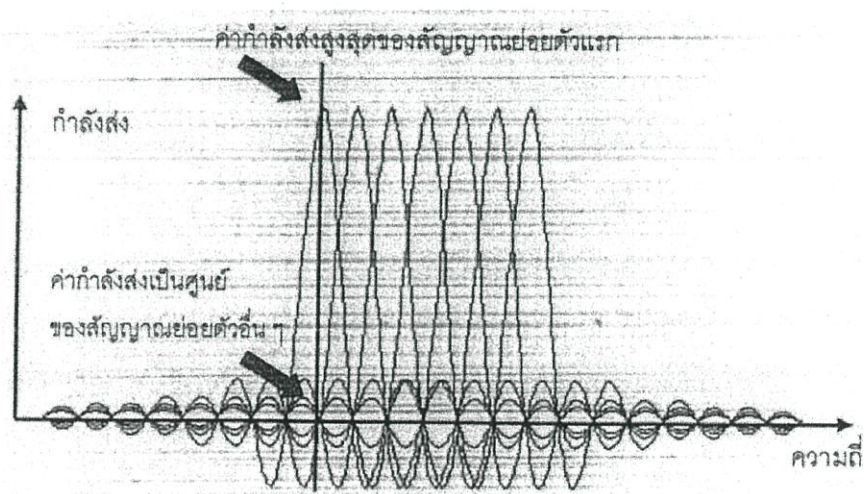
ใช้ช่วงความถี่ขนาดใหญ่ในการส่งสัญญาณ แต่มีวิธีการกู้คืนข้อมูลด้วยการสร้าง Bit เพิ่มเติม ซึ่งจะเรียกว่า Chipping Code ขนาด Code ยิ่งมากยิ่งทำให้ความเป็นไปได้ในการกู้ข้อมูลมีสูงขึ้น แต่จะใช้ Bandwidth เพิ่มขึ้นด้วย

3) High Rate/Direct Sequence Spread Spectrum(HR/DSSS)

เป็นเทคนิคการส่งที่คล้ายกับ DSSS แต่ส่งในอัตราข้อมูลที่สูงกว่า

4) Orthogonal Frequency Division Multiplexing (OFDM)

ใช้เทคนิคของการ Multiplex ความถี่ในการส่งสัญญาณ FDM แต่การส่งแบบนี้จะสิ้นเปลือง Bandwidth จึงพัฒนาการส่งที่ใช้สัญญาณย่อย (Sub-Carrier) เพื่อส่งข้อมูลไปพร้อมกัน โดยใช้อัตราการ Modulation ที่มีอัตราไม่สูงมาก (แต่ให้ผลรวมที่มีค่าสูง) ความถี่ที่ใช้เป็นช่วงความถี่ที่ซ้อนทับกัน โดยไม่มีช่องว่างระหว่างความถี่ ซึ่งเป็นการใช้ Bandwidth อย่างมีประสิทธิภาพมากขึ้น



รูปที่ 2.15 แสดงการมอดูเลชันแบบ OFDM

2.4 การพัฒนา Application บนระบบปฏิบัติการแอนดรอยด์

Android เป็นระบบปฏิบัติการที่มีพื้นฐานอยู่บน Linux ถูกออกแบบมาสำหรับอุปกรณ์ที่ใช้จอสัมผัส เช่น Smartphone และ Tablet ซึ่งถูกคิดค้นและพัฒนาโดยบริษัทแอนดรอยด์ (Android, Inc.) ซึ่งต่อมา Google ได้ทำการซื้อต่อบริษัทในปี พ.ศ. 2548 แอนดรอยด์ถูกเปิดตัวเมื่อปี พ.ศ. 2550 พร้อมกับการก่อตั้ง Open Handset Alliance ซึ่งเป็นกลุ่มของบริษัทผลิตฮาร์ดแวร์, ซอฟต์แวร์ และการสื่อสารคมนาคม ที่ร่วมมือกันสร้างมาตรฐานเปิดสำหรับอุปกรณ์พกพา โดย Smartphone ที่ใช้ระบบปฏิบัติการแอนดรอยด์เครื่องแรกของโลกคือ HTC Dream วางจำหน่ายเมื่อปี พ.ศ. 2551

2.4.1 เครื่องมือและโปรแกรมที่ใช้ในการพัฒนา Application

โปรแกรมหรือเครื่องมือที่ใช้พัฒนานั้นมีให้เลือกใช้หลายค่ายหลายทางเลือกด้วยกัน แต่ในที่นี้ได้ใช้โปรแกรมที่ google แนะนำ ซึ่งไม่เสียค่าใช้จ่ายและเป็นที่ยอมรับใช้กันอย่างแพร่หลายสำหรับนักพัฒนาทั่วไป เครื่องมือที่ต้องติดตั้งมีดังนี้

- 1) JDK (Java Development Kit) ชุดพัฒนาสำหรับภาษา Java
- 2) Android Developer Tools หรือ Eclipse เป็น IDE ช่วยในการเขียนโปรแกรมหรือ Application
- 3) Android SDK (Android Software Development Kit) เป็น Software Library สำหรับช่วยสร้างและทดสอบ Application
- 4) Android Development Tools plug-in (ADT) เป็น Plug-in สำหรับพัฒนา Application สำหรับแอนดรอยด์ บน Eclipse
- 5) Android Debugging Bridge (ADB) เป็น Driver สำหรับอุปกรณ์แต่ละเครื่อง ต้องทำการติดตั้งก่อนที่จะทำการพัฒนา Application

2.4.2 ภาษา Java

Java เป็นภาษาโปรแกรมที่พัฒนาขึ้นโดยทีมวิจัยชื่อ Green ของบริษัท Sun Microsystems ซึ่งมีวัตถุประสงค์เพื่อพัฒนาภาษาที่ใช้โปรแกรมหรือระบบที่สนับสนุนการทำงานของแต่ละระบบงานย่อยๆ ภาษา Java ถูกพัฒนาขึ้นบนคุณสมบัติดังนี้

- 1) ไม่เสียค่าใช้จ่ายในการพัฒนา
- 2) ง่ายต่อการเรียนรู้และเข้าใจเนื่องจากกลไกของภาษาไม่ซับซ้อน
- 3) มีความคงทนเนื่องจากการตรวจจับข้อผิดพลาดด้วยกลไกการจัดการข้อผิดพลาดและมีกลไกในการคืนพื้นที่ในหน่วยความจำอัตโนมัติ
- 4) มีความปลอดภัยสูง
- 5) ทำงานได้ในทุกระบบคือมีคุณลักษณะ Java Platform
- 6) มี Class และ Interface ให้ใช้เป็นจำนวนมาก

พื้นฐานในการเขียนโปรแกรมภาษา Java

การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming: OOP) เป็นการแบ่งขอบเขตของงานออกเป็นส่วนย่อยๆ ที่เป็นอิสระต่อกันโดยมองส่วนต่างๆ เป็นวัตถุหรือ Object ที่ไม่ขึ้นต่อกัน แต่มีการทำงานร่วมกัน

- 1) Class เป็นสิ่งที่ใช้อธิบายลักษณะและความสามารถของ Object เปรียบได้กับแม่แบบของ Object
- 2) Object คือสิ่งต่างๆ ที่ปรากฏอยู่รอบตัว มีคุณลักษณะและความสามารถในการทำงาน เช่น คน, รถยนต์ หรือเครื่องคอมพิวเตอร์ เป็นต้น
- 3) Attribute เป็นสิ่งที่ใช้อธิบายลักษณะของ Object
- 4) Method เป็นสิ่งที่ใช้อธิบายการทำงานของ Object

การสร้าง Class สามารถประกาศสร้างได้ดังนี้

```
[รูปแบบการเข้าถึง] Class ชื่อ Class
{
    [ การประกาศ Attribute]
    [ การประกาศ Method]
}
```

ตัวอย่างการสร้าง Class เช่น

```
Public class Cat
{
}
```

การประกาศ Attribute

```
[รูปแบบการเข้าถึง] ชนิดข้อมูล ชื่อ Attribute;
```

ตัวอย่างการประกาศ เช่น

```
Private String nameCat;
Public double heightCat;
```

การประกาศ Method

```
[รูปแบบการเข้าถึง] ชนิดข้อมูลที่ส่งกลับ ชื่อ Method (ข้อมูลที่รับเข้ามา)
{
    ชุดคำสั่ง;
}
```

ตัวอย่างการประกาศ เช่น

```
Public double downWeight (double dblWeight)
{
    weightCat = weightCat-dblWeight;
    return weightCat;
}
```

การสร้าง Object

```
ชื่อ Class ชื่อ Object;
ชื่อ Object = new ชื่อ Class;
```

หรือ

```
ชื่อ Class ชื่อ Object = new ชื่อ Class;
```

ตัวอย่างการสร้าง Object

```
Cat catThai;
CatThai = new Cat;
Cat catPersia = new Cat;
```

การใช้งาน Object

```
Cat catPersia =new Cat;
catPersia.setHeight (30.5);
catPersia.setWeight (2,3);
```

การใช้งานคำสั่ง try...catch

เป็นคำสั่งที่ไว้คอยตรวจสอบความผิดพลาด ไม่ว่าจะเกิดจากโปรแกรมหรือผู้ใช้ คำสั่งนี้ทำงานคล้ายกับ if statement ถ้าเกิดข้อผิดพลาดในคำสั่งที่ตามหลังคำสั่ง try โปรแกรมจะตรวจสอบที่คำสั่ง catch ว่าเป็นข้อผิดพลาดประเภทใด และทำงานที่ชุดคำสั่งหลังคำสั่ง catch ที่ตรงกับข้อผิดพลาด ซึ่งมีรูปแบบการใช้งานดังนี้

```
try {
    [statement]
}
catch (the Exception e) {
    [statements_n]
}
finally {
    [finalStatements]
}
```

โดยที่

Statement เป็นชุดคำสั่งที่ต้องการดักจับข้อผิดพลาดในขณะที่โปรแกรมทำงาน
 statements_n เป็นชุดคำสั่งที่กำหนดให้ทำงานเมื่อเกิดข้อผิดพลาดตรงตาม Exception
 ซึ่งสามารถมีได้มากกว่า 1 ชุดคำสั่ง
 finalStatements เป็นชุดคำสั่งที่ทำงานปกติไม่ว่าจะมี Exception เกิดขึ้นหรือไม่

คำสั่ง switch

เป็นคำสั่งควบคุมที่ให้โปรแกรมเลือกทำงานในชุดคำสั่งใดชุดคำสั่งหนึ่งจากหลาย
 ทางเลือก ในแต่ละทางเลือกจะมีการกำหนดเงื่อนไข โดยตรวจสอบเงื่อนไขแต่ละทางเลือก หาก
 พบว่าทางเลือกใดมีเงื่อนไขเป็นจริง จะทำงานที่ชุดคำสั่งภายใต้ทางเลือกนั้น โดยไม่พิจารณา
 ทางเลือกอื่นที่ยังไม่ได้ตรวจสอบ ซึ่งมีรูปแบบการทำงานดังนี้

```
switch (expression)
{
  case list_value_1;
    statement_1;
    break;
  case list_value_2;
    statement_2;
    break;
  .
  .
  .
  case list value _n;
    statement_n;
    break;
  default;
  statement;
}
```

โดยที่

Expression	เป็นนิพจน์ที่ต้องตรวจสอบค่าว่าตรงกับ list_value ไດ เพื่อจะได้ทำงานตามชุดคำสั่งภายใต้ list_value นั้น
list_value	เป็นค่าข้อมูลที่ใช้ตรวจสอบกับ expression
statement_n	เป็นชุดของคำสั่งที่ทำงานเมื่อ list_value นั้น มีค่าเท่ากับ expression
statement	เป็นชุดของคำสั่งที่ทำงานเมื่อไม่มี list_value ไດ มีค่าเท่ากับ expression
break	เป็นคำสั่งให้หยุดการทำงานและออกจาก Blog ของชุดคำสั่ง switch
default	เป็นคำสั่งจะถูกทำเมื่อค่าในเงื่อนไขของชุดคำสั่ง switch ไม่ตรงกับกรณีใดๆ ที่ถูกกำหนด จะมีหรือไม่มีก็ได้

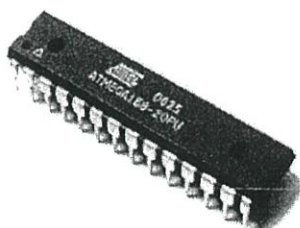
2.5 Microcontroller

2.5.1 ไมโครคอนโทรลเลอร์ ATmega168

ไมโครคอนโทรลเลอร์ ATmega168 คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอา CPU, หน่วยความจำ และ Port ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยมีหน้าที่ดังนี้

- 1) การรวบรวมข้อมูลจาก Input ที่สนใจ
- 2) ประมวลผล
- 3) ควบคุมการทำงานอุปกรณ์อื่น

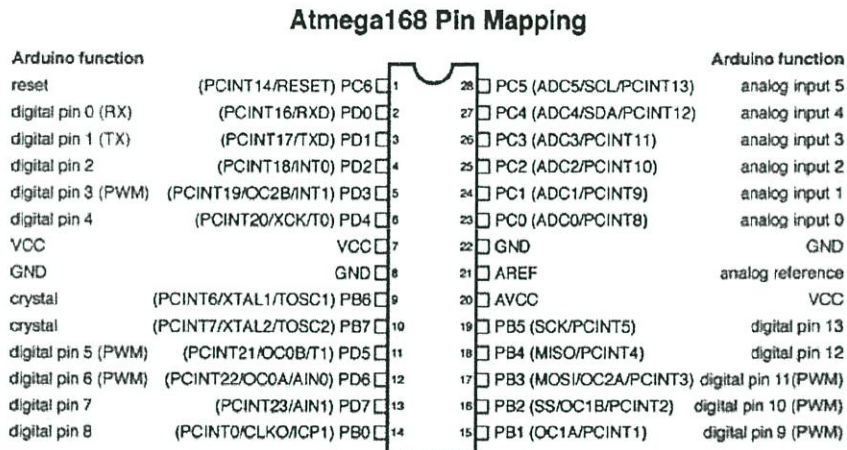
ซึ่งสามารถนำมาประยุกต์ใช้งานได้หลากหลาย ด้าน ไม่ว่าจะเป็นการประมวลผลเพื่อแสดงค่า การควบคุมการทำงานของอุปกรณ์เครื่องใช้ต่างๆ เป็นต้น โดยมีลักษณะภายนอกดังรูปที่ 2.16



รูปที่ 2.16 ไมโครคอนโทรลเลอร์ ATmega168

2.5.1.1 รายละเอียด Port ต่างๆ ของ ATmega168

แสดง Pin ของ ATmega168 ดังรูปที่ 2.17

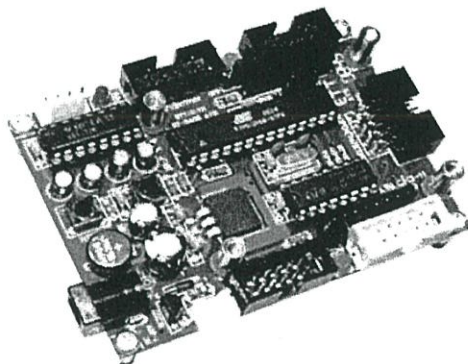


รูปที่ 2.17 รายละเอียดขาต่างๆ ของ ATmega168

หน้าที่แต่ละ Port (เฉพาะ Port ที่ใช้งาน)

PC0 (ADC0)	รับ Analog Input และมีฟังก์ชัน A/D
PD0 (Rx/D)	รับข้อมูลแบบอนุกรม (Digital)
PD1 (Tx/D)	ส่งข้อมูลแบบอนุกรม (Digital)
PD7	รับ Digital Input
Vcc	ไฟเลี้ยง
GND	Ground

2.5.1.2 ET BASE AVR EASY168



รูปที่ 2.18 บอร์ด ET BASE AVR EASY168

เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR ซึ่งแสดงในรูปที่ 2.18 โดยบอร์ดที่เลือกใช้ เบอร์ ATmega168 เป็นไมโครคอนโทรลเลอร์ประจำบอร์ด โดยมีจุดเด่น คือ เป็นไมโครคอนโทรลเลอร์ขนาดเล็ก แต่เพียบพร้อมไปด้วยทรัพยากรพื้นฐานต่างๆ อย่างครบถ้วน เหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆ ได้เป็นอย่างดี โดยมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM และ ADC ฯลฯ

2.5.1.3 คำสั่งหรือฟังก์ชันของ ATmega168

กลุ่มคำสั่งสำหรับใช้งาน Digital I/O

```
pinmode(pin,mode)
digitalWrite(pin,value)
int digitalRead(pin)
```

กลุ่มคำสั่งสำหรับใช้งาน Analog I/O

```
int analogRead(pin)
analogWrite(pin,value)
```

กลุ่มคำสั่งสำหรับหน่วงเวลา (Time)

```
delay(ms)
delayMicroseconds(us)
```

กลุ่มคำสั่งสื่อสารอนุกรม (Serial Communication)

```
Serial.begin(speed)
int Serial.available()
int Serial.read()
Serial.flush()
Serial.print(data)
Serial.print(data,format)
Serial.println(data)
Serial.println(data,format)
```

2.5.1.4 กลุ่มคำสั่งสำหรับใช้งาน Digital I/O

เป็นกลุ่มคำสั่งที่ใช้งาน Pin I/O ของ ATmega168 ในรูปแบบของ Digital I/O ซึ่งตามปกติแล้วในการกำหนดหน้าที่การใช้งานขาสัญญาณของไมโครคอนโทรลเลอร์ ATmega168 ซึ่งนิยมเรียกกันว่า Pin I/O ต้องเข้าไปกำหนดค่าให้กับ Register ต่างๆ ในตัว MCU โดยตรง เพื่อเลือกกำหนดรูปแบบการทำงานของขาสัญญาณของ Pin I/O ต่างๆ ให้มีคุณสมบัติตามที่ต้องการ

สำหรับ Digital I/O ใน ATmega168 มีจำนวน 14 Pin โดย ATmega168 ได้กำหนดรหัสของตัวเลขซึ่งเป็นตัวเลขจำนวนเต็ม มีค่าระหว่าง 0 ถึง 13 สำหรับอ้างถึง Digital I/O ทั้ง 14 Pin แต่ตามปกติแล้ว จะมี Digital I/O จำนวน 2 Pin ซึ่งถูกสงวนไว้สำหรับใช้ Port สื่อสารอนุกรม RS232 สำหรับใช้ Upload Code ของ Program ให้กับบอร์ด จึงเหลือ Digital I/O Pin สำหรับใช้งานเป็น Digital I/O Pin จริงๆ จำนวน 12 Pin คือ Digital I/O หมายเลข 2 ถึง 13 เท่านั้น

1) void pinMode(pin,mode)

ใช้ทำหน้าที่กำหนดการทำงานของขา I/O ที่เป็น Digital I/O Pin ของไมโครคอนโทรลเลอร์ ATmega168 ตามที่ ATmega168 กำหนดไว้ว่าต้องการใช้งานขา Digital I/O ขาใด เพื่อใช้งานเป็น Input หรือ Output ไม่มีการส่งค่าคืนกลับจากฟังก์ชัน

รูปแบบของคำสั่ง

```
pinMode(pin,mode)
```

ค่าพารามิเตอร์ที่ต้องการ

Pin หมายถึง หมายเลขรหัส Pin ของขาสัญญาณที่ทำหน้าที่เป็น Digital I/O Pin ซึ่งมีจำนวน 14 Pin คือ 0 ถึง 13 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นจำนวนเต็ม (int) ด้วย Mode หมายถึง หน้าที่การทำงานของ Digital I/O Pin ที่ต้องการกำหนด ซึ่งสามารถกำหนดได้ 2 หน้าที่ โดยใส่รหัสข้อความเป็น Input หรือ Output

2) void digitalWrite(pin,value)

ใช้กำหนดสถานะทาง Output ให้กับ Digital I/O Pin ว่าต้องการให้มีสถานะทาง Logic เป็น High หรือ Low ซึ่งขาสัญญาณที่จะส่งงานด้วยคำสั่งนี้ต้องถูกกำหนดคุณสมบัติให้ทำหน้าที่เป็น Output เรียบร้อยแล้ว ไม่มีการส่งค่าคืนกลับจากฟังก์ชัน

รูปแบบของคำสั่ง

```
digitalWrite(pin,value)
```

ค่าพารามิเตอร์ที่ต้องการ

Pin หมายถึง หมายเลขรหัส Pin ของขาสัญญาณที่ทำหน้าที่เป็น Digital I/O Pin ซึ่งมีจำนวน 14 Pin คือ 0 ถึง 13 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นจำนวนเต็ม (int) ด้วย Value หมายถึง ค่าสถานะทาง Output ของ Digital Output Pin ที่ต้องการกำหนด ซึ่งสามารถกำหนดค่าสถานะให้กับ Pin ได้ 2 ค่า คือ High และ Low

3) int digitalRead(pin)

ใช้อ่านค่าสถานะ Logic Input ของ Digital Input Pin ว่ามีค่าสถานะเป็น High หรือ Low ซึ่งขาสัญญาณที่อ่านค่าด้วยคำสั่งนี้ ต้องถูกกำหนดคุณสมบัติให้ทำหน้าที่เป็น Input เรียบร้อยแล้ว

รูปแบบของคำสั่ง

```
var = digitalRead(pin)
```

ค่าพารามิเตอร์ที่ต้องการ

Pin หมายถึง หมายเลขรหัส Pin ของขาสัญญาณที่ทำหน้าที่เป็น Digital I/O Pin ซึ่งมีจำนวน 14 Pin คือ 0 ถึง 13 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นจำนวนเต็ม (int) ด้วย

Var หมายถึง ตัวแปรแบบ int สำหรับใช้รรับค่าที่ส่งคืนกลับมาจากฟังก์ชัน ซึ่งเป็นค่าสถานะทาง Logic ของ Digital Input Pin ซึ่งมีค่าเป็น High หรือ Low

2.5.1.5 กลุ่มคำสั่งเกี่ยวกับ Analog I/O

คุณสมบัติของสัญญาณในกลุ่มนี้ ใช้วัดหรืออ่านค่าสัญญาณที่เป็นแบบแรงดันไฟฟ้า ซึ่งเป็นสัญญาณ Analog โดย Analog Input สามารถรับแรงดัน Input ที่มีขนาดแรงดันระหว่าง 0-5 โวลต์ แล้วแปลงค่าขนาดของแรงดันที่อ่านได้ มาเป็นข้อมูลแบบเลขจำนวนเต็ม ซึ่งมีค่าความละเอียด 1024 ค่า (0-1023) ซึ่งความสามารถของ ATmega168 ในการอ่านและวัดค่าระดับของแรงดันที่เป็น Analog นี้ จะช่วยทำให้สามารถนำไมโครคอนโทรลเลอร์มาประยุกต์ใช้งาน เกี่ยวกับการจับ Input แบบต่างๆ ได้มากมายหลายแบบ โดยใช้งานร่วมกับอุปกรณ์ตรวจจับ (Sensor) แบบต่างๆ ที่มีค่าออกมาเป็นแรงดัน

1) int analogRead(pin)

ใช้อ่านค่าระดับแรงดันที่ขาสัญญาณ Analog Input Pin ว่าระดับแรงดัน (Analog Input) มีค่าเป็นเท่าใด (0-Analog Reference) คือ ค่าผลลัพธ์ที่อ่านได้จากฟังก์ชันจะเป็นระดับแรงดันที่อ่านได้ที่ขา Analog Input Pin ซึ่งทำการแปลงค่าเป็นข้อมูลแบบดิจิตอลขนาด 10 bit เรียบร้อยแล้ว โดยผลลัพธ์ที่ได้มีค่า 1024 ระดับ โดยผลลัพธ์มีค่าระหว่าง 0-1023 ซึ่งค่า 1 ระดับของข้อมูลที่อ่านได้มีขนาดเท่ากับ ขนาดของแรงดันอ้างอิง/1024 ซึ่งถ้าใช้ค่าแรงดันอ้างอิง หรือ Analog Reference ตามค่า DEFAULT ซึ่งมีค่าเป็น +5V จะได้ขนาดของแรงดันต่อขนาดของข้อมูล 1 ระดับ ประมาณ $5V/1024$ หรือประมาณ $0.0049V$ (4.9mV)

โดยการทำงานของคำสั่งนี้จะใช้เวลาในการทำงานประมาณ $100 \mu s$ (100 ไมโครวินาที) ดังนั้นสามารถส่งค่า Analog Input ได้เร็วสุดประมาณ 10,000 รอบต่อวินาที

รูปแบบของคำสั่ง

```
var = analogRead(pin)
```

ค่าพารามิเตอร์ที่ต้องการ

Pin หมายถึง หมายเลขรหัส Pin ของขาสัญญาณที่ทำหน้าที่ Analog Input Pin ซึ่งมีจำนวน 6 Pin ถ้าใช้ชิพรุ่นที่มีตัวถัง 28 Pin โดยรหัสตัวเลข 0-5 และ 0-7 โดยต้องกำหนดรูปแบบของตัวเลขให้เป็นแบบจำนวนเต็ม (int) ด้วย

ตัวแปรแบบ int สำหรับใช้รอรับค่าที่ส่งคืนกลับมาจากฟังก์ชัน ซึ่งเป็นค่าแรงดัน Analog Input ที่ถูกแปลงเป็นค่าข้อมูล 10 bit แล้ว (1024 ค่า) โดยค่าผลลัพธ์ที่เป็นไปได้ คือ 0-1023 ซึ่งในการใช้งานต้องสร้างตัวแปรแบบ int ขึ้นมาเพื่อใช้รอรับค่าที่จะส่งกลับมาจากฟังก์ชันด้วย

2.5.1.6 กลุ่มคำสั่งเกี่ยวกับการหน่วงเวลา

1) void delay(ms)

ใช้สำหรับหน่วงเวลาโดยมีหน่วยเป็น ms (millisecond) ซึ่งสามารถกำหนดค่าการหน่วงเวลาเป็นค่าตัวเลขจำนวนเต็มระหว่าง 1 ถึง 4,294,967,295 ms โดยที่ค่าเวลาของ 1 ms จะมีค่าเท่ากับ 1/1000 วินาที ไม่มีการส่งค่าคืนกลับมาจากฟังก์ชัน

รูปแบบคำสั่ง

delay(ms)

ค่าพารามิเตอร์ที่ต้องการ

ms คือค่าเวลาที่ต้องการหน่วงเวลา มีหน่วยเป็น ms โดยค่าที่กำหนดให้กับคำสั่งเป็นค่าแบบ Unsigned Long ซึ่งสามารถกำหนดค่าได้สูงสุด 4,294,967,295 แต่ต้องไม่ลืมว่าการกำหนดค่าตัวเลขของภาษาซีที่ใช้กับ ATmega168 ตามปกตินั้น ถ้ากำหนดค่าโดยเขียนรูปแบบของตัวเลขปกติแล้ว โปรแกรม Compiler ของ ATmega168 จะถือว่าตัวเลขนั้นมีค่าเป็น Integer ซึ่งมีค่าไม่เกิน 32767 เท่านั้น เมื่อต้องการกำหนดค่าหน่วงเวลาให้มีค่ามากกว่า 32767 ต้องกำหนดรูปแบบของตัวเลขให้เป็น Unsigned Long ด้วย โดยใช้ตัวอักษร 'UL' ตามท้ายค่าตัวเลข

2) void delayMicrosecond(us)

ใช้หน่วงเวลาโดยมีหน่วยเป็น us (micro-second) ไม่มีการส่งค่าคืนกลับมาจากฟังก์ชัน

รูปแบบของคำสั่ง

delayMicrosecond(us)

ค่าพารามิเตอร์ที่ต้องการ

us คือ ค่าเวลาที่ต้องการหน่วงเวลา มีหน่วยเป็น microsecond

2.5.1.7 กลุ่มคำสั่งเกี่ยวกับการสื่อสารอนุกรม

ใช้สำหรับการติดต่อสื่อสารระหว่าง ATmega168 กับอุปกรณ์ภายนอก ในรูปแบบของการสื่อสารแบบอนุกรม ซึ่งอาจเป็นเครื่องอุปกรณ์ PC หรืออุปกรณ์ใดๆ ก็ได้ โดยเมื่อต้องการรับส่งข้อมูล ด้วยการสื่อสารแบบอนุกรม จะต้องสูญเสียขาสัญญาณไปจำนวน 2 เส้น คือ PD0 (Digital-0), PD1 (Digital-1) โดย

PD0 (Digital-0) ใช้ทำหน้าที่เป็นขารับสัญญาณข้อมูลอนุกรม (RXD)

PD1 (Digital-1) ใช้ทำหน้าที่เป็นขาส่งสัญญาณข้อมูลอนุกรม (TXD)

2.5.1.8 คำสั่งในกลุ่มของการสื่อสารอนุกรมของ ATmega168 มีทั้งหมด 5 คำสั่งหลักๆ คือ

Serial.begin(int speed)

Serial.print(data)

Serial.println(data)

int Serial.read()

Serial.flush()

1) Serial.begin(int speed)

ใช้สำหรับเปิดพอร์ตสื่อสารอนุกรม พร้อมทั้งกำหนดค่าความเร็วในการสื่อสาร (Baudrate) ตามที่ผู้ใช้กำหนด ซึ่งต้องกำหนดค่าความเร็วนี้ให้ตรงกับความเร็วของอุปกรณ์อีกฝ่ายหนึ่งที่จะทำการสื่อสารด้วย ซึ่งตามปกติคือ เครื่องคอมพิวเตอร์ โดยค่าความเร็วที่เป็นมาตรฐานที่นิยมใช้งานกันทั่วไป ได้แก่ 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, และ 115200 ไม่มีการส่งค่าคืนกลับจากฟังก์ชัน

ค่าพารามิเตอร์ที่ต้องการ

speed คือ ค่าความเร็วในการสื่อสารข้อมูลของพอร์ตสื่อสารอนุกรม ซึ่งมีค่าเป็นตัวเลขจำนวนเต็ม ค่าที่แนะนำคือ 19200

2) Serial.print(data)

ใช้ส่งข้อมูลออกไปยังพอร์ตสื่อสารอนุกรม อย่างต่อเนื่องในลักษณะของการพิมพ์ โดยข้อมูลที่จะส่งพิมพ์ด้วยคำสั่งนี้ สามารถเป็นได้ทั้งตัวแปรหรือข้อความต่างๆ ที่กำหนดไว้ในพารามิเตอร์ภายในวงเล็บ () ออกจากพอร์ตสื่อสารอนุกรม

ค่าพารามิเตอร์ที่ต้องการ

data คือ ข้อมูลที่ต้องการส่งพิมพ์ อาจเป็นข้อความ ตัวแปร ในรูปแบบต่างๆ ที่ต้องการส่งพิมพ์

3) Serial.println(data)

ใช้ส่งข้อมูลออกไปยังพอร์ตสื่อสารอนุกรม อย่างต่อเนื่องในลักษณะของการพิมพ์ แต่จะมีการขึ้นบรรทัดใหม่ของข้อมูล โดยข้อมูลที่จะส่งพิมพ์ด้วยคำสั่งนี้สามารถเป็นได้ทั้งตัวแปรหรือข้อความต่างๆ ที่กำหนดไว้ในพารามิเตอร์ภายในวงเล็บ () ออกทางพอร์ตสื่อสารอนุกรม

ค่าพารามิเตอร์ที่ต้องการ

data คือ ข้อมูลที่ต้องการส่งพิมพ์ อาจเป็นข้อความ ตัวแปร ในรูปแบบต่างๆ ที่ต้องการส่งพิมพ์

4) int Serial.read()

ใช้สำหรับอ่านข้อมูลจาก Buffer ของพอร์ตสื่อสารอนุกรมออกมาประมวลผล โดยเมื่ออ่านข้อมูลออกมา 1 Byte จะทำให้พื้นที่ Buffer ที่ใช้เก็บข้อมูลของพอร์ตสื่อสารอนุกรมว่างลงอีก 1 Byte เสมอ โดยค่าของข้อมูลที่อ่านได้จะเป็นข้อมูลที่ถูกระบุไว้ใน Buffer เป็นลำดับแรก เมื่ออ่านแล้วข้อมูลถัดไปจะถูกเลื่อนตำแหน่งมารอไว้เป็นลำดับแรกแทน โดยปกติค่าของข้อมูลจะมีค่าเป็นเลขจำนวนเต็มระหว่าง 0-255 แต่ถ้าไม่มีข้อมูลเก็บไว้ใน Buffer ค่าข้อมูลที่อ่านได้จะมีค่าเป็น -1 แทน

รูปแบบคำสั่ง

```
var = Serial.read()
```

ค่าพารามิเตอร์ที่ต้องการ

ไม่มีการส่งค่าใดๆ ให้กับฟังก์ชัน

5) Serial.flush()

ใช้สำหรับล้างข้อมูลทั้งหมดใน Buffer ของพอร์ตสื่อสารอนุกรมให้ว่างลง เพื่อรอรับข้อมูลใหม่ที่ส่งเข้ามาหลังจากจบการทำงานของคำสั่งนี้

ค่าพารามิเตอร์ที่ต้องการ

ไม่มีการส่งค่าใดๆ ให้กับฟังก์ชัน

2.5.1.9 กลุ่มคำสั่งทางคณิตศาสตร์และตรีโกณมิติ เป็นกลุ่มคำสั่งสำหรับใช้ในการคำนวณทางคณิตศาสตร์และตรีโกณมิติ ซึ่งใช้สำหรับตรวจสอบค่า เปรียบเทียบค่า และการแปลงค่าตัวแปร เพื่อนำไปใช้งานตามเงื่อนไขที่ต้องการ

1) Abs()

ใช้ทำหน้าที่สำหรับคำนวณหาค่าสัมบูรณ์ของตัวแปร คือ การเปลี่ยนค่าลบให้เป็นค่าบวก

รูปแบบคำสั่ง

var = abs(x)

พารามิเตอร์ที่ต้องการ

x คือ ตัวเลขที่ต้องการตรวจสอบ

2) tan(rad)

ใช้สำหรับคำนวณหาค่า tangent ของมุมที่กำหนดไว้ใน rad โดยจะให้ค่าผลลัพธ์เป็นตัวเลขแบบ double โดยมีค่าระหว่าง $-\infty$ ถึง $+\infty$ โดยค่ามุมที่กำหนดให้กับคำสั่งจะเป็นแบบ เรเดียน (rad) ซึ่งถ้าต้องการกำหนดค่ามุมที่เป็นองศาต้องคูณด้วย $(180 \div \pi)$ เสมอจึงจะได้ค่าที่ถูกต้อง

รูปแบบคำสั่ง

var = tan(rad)

ค่าพารามิเตอร์ที่ต้องการ

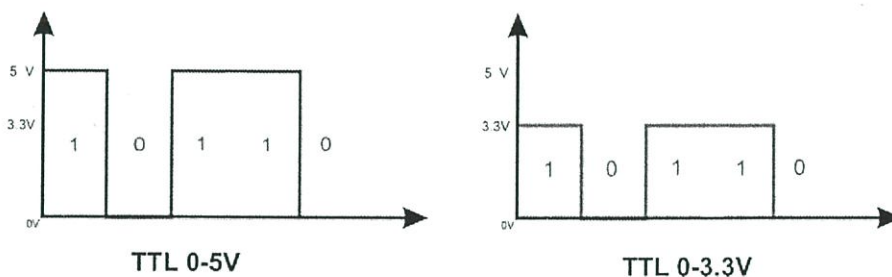
rad คือ ค่ามุมของ radian ที่ต้องการคำนวณหาค่า tangent ซึ่งต้องเป็นค่าแบบ float

2.6 หลักการส่งข้อมูลแบบอนุกรม

2.6.1 สัญญาณ TTL (Transistor-Transistor Logic)

TTL เป็นระดับแรงดันที่ถูกกำหนดขึ้นในยุคแรกๆ เพื่อใช้ระหว่าง Transistor กับ Transistor ภายในวงจรรวม (IC) ดังนั้น TTL จะใช้ระดับแรงดันอยู่ที่ 0–5 V แต่ในปัจจุบันมี

อุปกรณ์หลายเบอร์ที่ทำงานในช่วง 0–3.3 V (เรียกแรงดันระดับนี้ว่า LV-TTL) ซึ่งผู้ใช้ควรตรวจสอบจาก Datasheet ของอุปกรณ์ที่ใช้เสียก่อนว่าเป็นระดับแรงดันแบบใด เพราะหากใช้ผิดประเภทจะทำให้อุปกรณ์เสียหาย



(ก) ระดับแรงดันสัญญาณ TTL (0-5 V)

(ข) ระดับแรงดันสัญญาณ TTL (0-3.3 V)

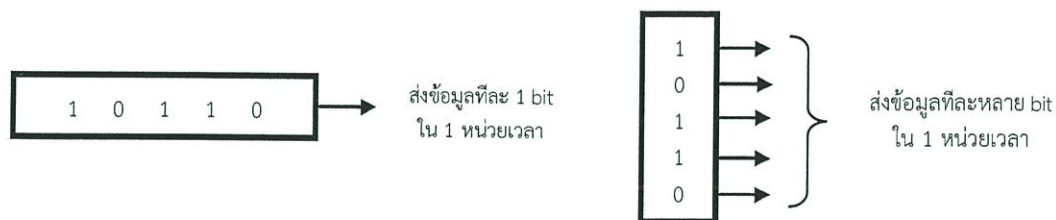
รูปที่ 2.19 ระดับแรงดันของสัญญาณ TTL

2.6.2 UART

ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter หมายถึง อุปกรณ์ที่ทำหน้าที่รับ-ส่งข้อมูลแบบ Asynchronous ซึ่งเป็นส่วนหนึ่งในการสื่อสารอนุกรมแบบ Asynchronous

การรับ-ส่งข้อมูลแบบอนุกรม (Serial Communication)

การรับ-ส่งข้อมูลแบบอนุกรมเป็นการรับ-ส่งข้อมูลที่ละ Bit แทนที่จะทำการส่งข้อมูลพร้อมกันหลาย bit ในเวลาเดียวกัน ตามรูปที่ 2.20(ข) ซึ่งการรับ-ส่งข้อมูลแบบอนุกรม จะมีความเร็วน้อยกว่าการรับ-ส่งข้อมูลแบบขนานที่ส่งข้อมูลที่ละหลาย Bit ในเวลาเดียวกัน ตามรูปที่ 2.20(ก) แต่จะประหยัดในการใช้จำนวนสายที่ใช้ในการส่งผ่านข้อมูล



(ก) การรับ-ส่งข้อมูลแบบอนุกรม

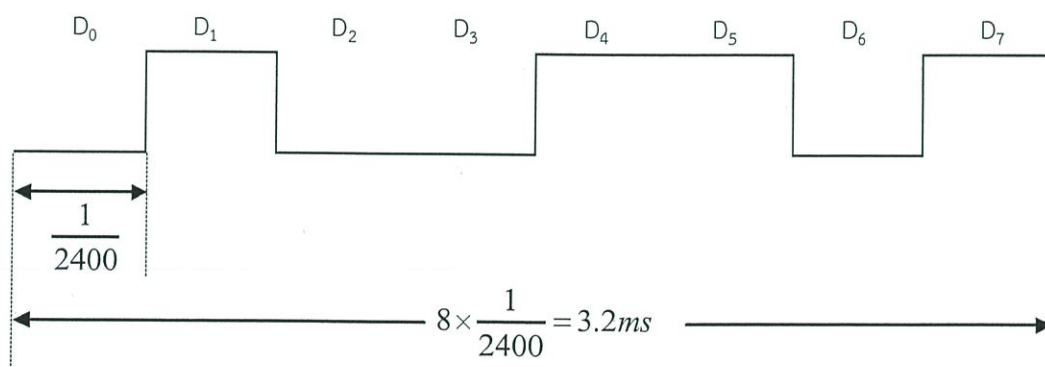
(ข) การรับ-ส่งข้อมูลแบบขนาน

รูปที่ 2.20 การรับ-ส่งข้อมูล

2.6.3 Baud Rate

คือ ความถี่ที่ใช้ในการส่งข้อมูล ซึ่งต้องสัมพันธ์กันระหว่างอุปกรณ์ที่ทำการรับ-ส่งข้อมูล ความถี่ที่ใช้มีชื่อเรียกว่า Baud Rate กล่าวคือ อัตราการรับ-ส่งข้อมูลเป็นจำนวน bit ใน 1 วินาที ถ้าเครื่องส่งและเครื่องรับใช้ Baud Rate ไม่ตรงกัน จะทำให้การรับ-ส่งข้อมูลเกิดผิดพลาด

โดยทั่วไปค่าของ Baud Rate จะใช้ค่าต่างๆ ดังนี้ 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600 และ 115200 เป็นต้น ตัวอย่างการส่งข้อมูลแสดงในรูปที่ 2.21



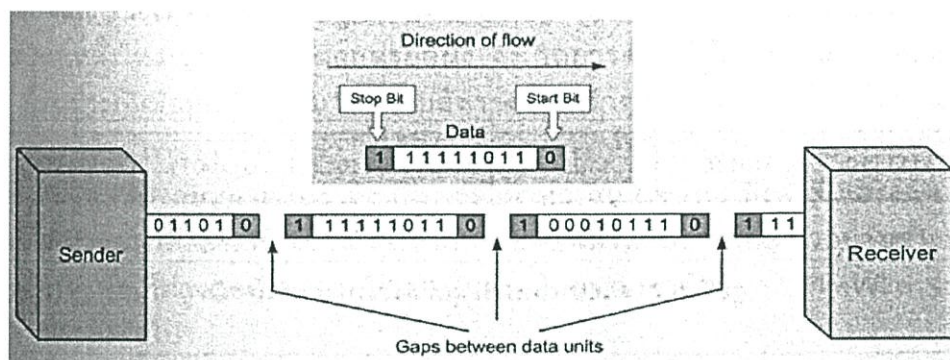
รูปที่ 2.21 สัญญาณของข้อมูลที่ถูกส่งแบบอนุกรม

2.6.4 วิธีต่างๆ ในการส่งข้อมูลแบบอนุกรม

2.6.4.1 การส่งข้อมูลแบบ Asynchronous

เป็นวิธีที่หลีกเลี่ยงปัญหาด้านเวลา ที่ฝั่งรับไม่ทราบเวลาที่แน่ชัดของข้อมูลที่ส่งมาจากฝั่งส่ง ฝั่งส่งและฝั่งรับไม่ต้องใช้สัญญาณนาฬิกาเดียวกันในการควบคุมจังหวะการรับ-ส่งข้อมูล โดย

เริ่มต้นที่ไม่มีการส่งข้อมูลใดๆ จะอยู่ในสภาวะนิ่งเฉย (Idle State) และกำหนดให้สัญญาณมีค่าเป็น 1 เมื่อมีการส่งข้อมูลระดับสัญญาณจะถูกกำหนดให้มีค่าเป็น 0 ทำให้เกิดเป็น Bit ขึ้นมา เรียกว่า Start Bit เพื่อบอกให้ทราบว่าจะมีข้อมูลส่งมา เมื่อฝั่งส่งได้ส่ง Bit ข้อมูลจนครบ (5-8 bit) จะส่งข้อมูลอีกหนึ่ง Bit ที่ระดับสัญญาณมีค่าเป็น 1 เป็นตัวปิดท้าย เรียกว่า Stop Bit การมี Start Bit และ Stop Bit เป็นกลไกที่ทำให้ฝั่งรับทราบถึงจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลในแต่ละ Byte อีกทั้งยังบอกให้รู้ว่า ได้ส่งข้อมูลครบตามจำนวน Byte แล้วตัวอย่างอุปกรณ์ที่ใช้สื่อสารด้วยวิธีนี้ คือ คีย์บอร์ด ซึ่งจะพบว่าแต่ละตัวอักษรที่พิมพ์จะมีช่วงเวลาที่แตกต่างกัน และเมื่อไม่มีการพิมพ์ข้อมูลใดๆ จะอยู่ในสภาวะ Idle

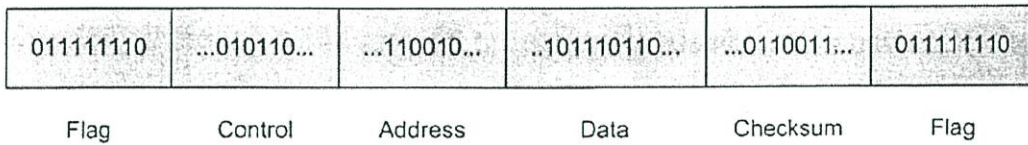


รูปที่ 2.22 การส่งข้อมูลแบบ Asynchronous

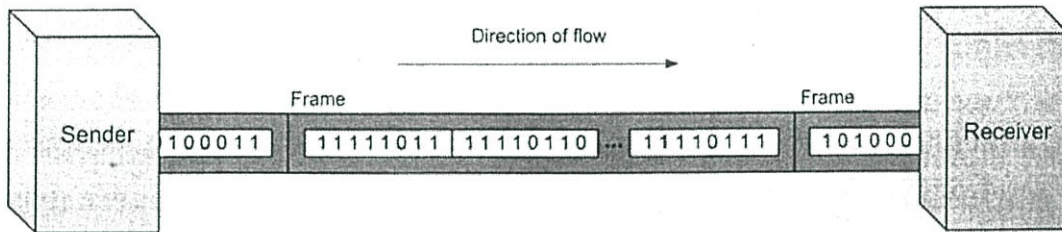
จากรูปที่ 2.22 ช่องว่างที่เกิดขึ้นระหว่าง Byte เรียกว่า Gap การส่งข้อมูลแบบ Asynchronous จะไม่มีการควบคุมจังหวะเวลาให้สอดคล้องกันในระดับ Byte แต่ในระดับ Bit ยังมีการควบคุมจังหวะเวลาให้สอดคล้องกันอยู่

2.6.4.2 การส่งข้อมูลแบบ Synchronous

เป็นการส่งกลุ่มข้อมูลแบบต่อเนื่องกัน โดย Bit ที่ทยอยส่งเข้ามาจะมีการรวมกันให้มีขนาดใหญ่ขึ้นเรียกว่า Frame หรือ Data block แสดงในรูปที่ 2.23 ซึ่งอาจมีจำนวนมากกว่า 1000 bit เมื่อข้อมูลส่งมาถึงปลายทาง ฝั่งรับจะทำหน้าที่นับจำนวน Bit และจับกลุ่มเป็น Byte ซึ่งการส่งวิธีนี้จะไม่มีช่องว่าง ไม่มี Start Bit และ Stop Bit ทำให้ฝ่ายรับไม่สามารถทราบได้เลยว่าข้อมูลที่ส่งมาครบหรือยัง ดังนั้นการควบคุมจังหวะเวลาให้สอดคล้องกันระหว่างอุปกรณ์จึงกลายเป็นสิ่งสำคัญคือ ทั้งฝั่งส่งและฝั่งรับจะต้องทำงานสอดคล้องกันตามจังหวะสัญญาณนาฬิกา ฝั่งรับจะได้รับสัญญาณนาฬิกาจากฝั่งส่ง โดยฝั่งส่งสามารถส่งสัญญาณนาฬิกาได้ 2 วิธี วิธีแรก คือ ส่งสัญญาณนาฬิกาแยกออกมาจากการส่งข้อมูล (ใช้งานได้ดีเมื่อส่งข้อมูลระยะใกล้) หรือ วิธีที่สอง คือ ส่งสัญญาณนาฬิกา รวมเข้ากับสัญญาณข้อมูล



รูปที่ 2.23 Frame ข้อมูลที่ส่งในรูปแบบ Synchronous



รูปที่ 2.24 การส่งข้อมูลแบบ Synchronous

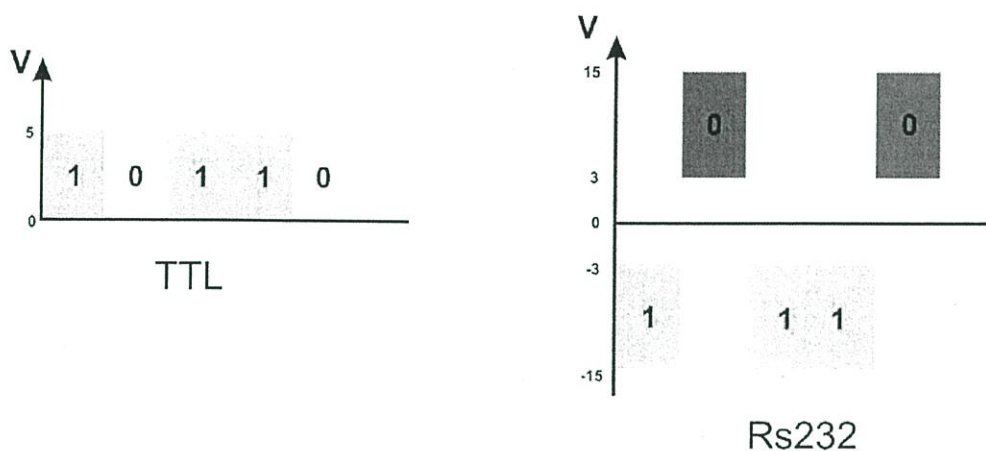
จากรูปที่ 2.24 ในด้านอัตราการความเร็วการส่งข้อมูลแบบ Synchronous จะมีความเร็วสูงกว่าแบบ Asynchronous เนื่องจากข้อมูลมีการส่งอย่างต่อเนื่อง และไม่มีการเพิ่ม Bit พิเศษต่างๆ เข้าไปในข้อมูล ทำให้ฝั่งรับไม่ต้องเสียเวลาในการนำ Bit พิเศษเหล่านั้นออกมา ดังนั้นการสื่อสารด้วยวิธีนี้จึงมีความเร็วสูง เหมาะกับอุปกรณ์สื่อสารด้วยความเร็วสูง เช่น การส่งข้อมูลไปมาระหว่างเครื่องคอมพิวเตอร์

2.6.5 RS232 (Recommended Standard 232)

RS232 คือ มาตรฐานการเชื่อมต่อข้อมูลแบบ Serial ใช้เพื่อเพิ่มระยะทางในการส่งข้อมูลแบบ Serial ให้สามารถส่งได้ระยะทางที่มากขึ้น โดยมีการเปลี่ยนระดับแรงดัน ของ Logic จากเดิมที่จะอยู่ในช่วง 0-5 V หรือ 0-3.3 V เป็นช่วง -15 ถึง 15 V โดยมีรายละเอียดดังนี้

Logic 0 ของ RS232 จะอยู่ในช่วง 3 ถึง 15V

Logic 1 ของ RS232 จะอยู่ในช่วง -3 ถึง -15V



(ก) ระดับแรงดันสัญญาณ TTL (0-5 V)

(ข)ระดับแรงดันสัญญาณ RS232

รูปที่ 2.25 ระดับแรงดันของสัญญาณ TTL และสัญญาณ RS232

2.7 Application กล้อง ที่ส่งภาพจากโทรศัพท์มือถือ Smart Phone ไปยังโทรศัพท์มือถือ Smart Phone ที่ใช้บังคับ

เนื่องจากในปัจจุบันโทรศัพท์มือถือ Smart Phone มีราคาที่ถูกลงพร้อมกับสเปคที่ดีขึ้นจึงได้นำโทรศัพท์มือถือ Smart Phone มาทำเป็น IP Camera แทนการใช้กล้อง ซึ่งมีน้ำหนักมากและราคาสูง แล้วได้ทำการต่อยอด Application โดยนำโทรศัพท์มือถือ Smart Phone เครื่องหนึ่งที่ทำหน้าที่เป็น IP Camera แล้วทำการส่งภาพไปยังโทรศัพท์มือถือ Smart Phone อีกเครื่องหนึ่งที่ทำหน้าที่บังคับรถ โดยคำสั่งที่ใช้มีดังต่อไปนี้

2.7.1 การติดต่อใช้งานกล้อง

สำหรับการใช้งานกล้องบนอุปกรณ์แอนดรอยด์ ภาพที่แสดงตอนเปิดหน้ากล้องเรียกว่า Surface View ดังนั้นในส่วนของ Layout จะต้องมีการ Surface View อยู่ด้วยจึงจะทำให้เห็นภาพจากกล้อง โดย Surface View จะมีอยู่ด้วยกัน 3 สถานะ คือ Surface Created, Surface Changed และ Surface Destroyed ซึ่งมีวิธีประกาศดังนี้

```

public void surfaceChanged(SurfaceHolder arg0
    , int arg1, int arg2, int arg3)
{

}

public void surfaceCreated(SurfaceHolder arg0)
{

}

public void surfaceDestroyed(SurfaceHolder arg0)
{

}

```

รูปที่ 2.26 โค้ดที่ใช้สร้าง, ทำลาย และเปลี่ยนแปลงการทำงานของ Surface Layout

และคำสั่ง Permission เพื่อขออนุญาตในการเข้าใช้กล้องบนอุปกรณ์แอนดรอยด์ คือ

```

<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />

```

รูปที่ 2.27 คำสั่ง Permission ในการเข้าใช้งานกล้องบนอุปกรณ์แอนดรอยด์

```

Camera mCamera;
mCamera = Camera.open(0); //ใช้ในการเปิดกล้อง

```

```
mCamera.release();//ใช้ในการปิดการใช้งานกล้องเพื่อปล่อยให้แอปอื่นทำงานต่อได้
mCamera.autofocus();//ใช้ในการทำ autofocus ของกล้อง
mCamera.startPreview();//เป็นคำสั่งที่ใช้แสดง surface ขึ้นมาบนหน้าจอเพื่อแสดงภาพที่ได้
```

รูปที่ 2.28 คำสั่งที่ใช้ประกาศ Class เพื่อเข้าใช้กล้องและคำสั่งต่างๆ ที่เกี่ยวข้อง

2.7.2 การส่งถ่ายภาพ, การเปิด-ปิดไฟแฟลช และการสั่งปรับโฟกัสของกล้อง

สำหรับ Application นอกจากจะส่งภาพได้แล้ว ยังได้มีการใส่ความสามารถให้ถ่ายภาพ (Snap) เป็นภาพนิ่งเก็บไว้ในโทรศัพท์ที่ทำหน้าที่เป็นกล้อง อีกทั้งยังสามารถโฟกัสภาพและเปิด-ปิดไฟแฟลชได้ ซึ่งคำสั่งที่ใช้มีดังต่อไปนี้

ในการถ่ายภาพนั้นจะใช้คำสั่งที่มีอยู่ในคลาส Camera คือ

```
mCamera.takePicture(ShutterCallback shutter, PictureCallback raw
, PictureCallback postView, PictureCallback jpeg);
```

รูปที่ 2.29 คำสั่งที่ใช้ถ่ายภาพ

จะเห็นว่าคำสั่ง takePicture แล้วมีคำสั่ง Callback อยู่ด้วยกัน 4 คำสั่ง คือ

- 1) ShutterCallback shutter: Callback จะทำงานเมื่อ Shutter ทำงาน โดยทำงานอยู่ภายใต้ฟังก์ชัน onShutter
 - 2) PictureCallback raw: Callback จะทำงานหลังจากถ่ายภาพแล้ว โดยข้อมูลภาพถ่ายที่ส่งเข้ามาจะอยู่ภายใต้ฟังก์ชัน onPictureTaken ซึ่งข้อมูลที่ส่งเข้ามาในฟังก์ชันนี้จะ เป็นข้อมูลดิบจากการถ่ายภาพ
 - 3) PictureCallback postView: จะเห็นว่าเป็น PictureCallback เหมือนกัน แต่สำหรับ Callback ตัวนี้จะส่งข้อมูลเป็นภาพที่แสดงให้เห็นชั่วคราว ให้นึกถึงเวลาถ่ายภาพ แล้วจะมีภาพแสดงให้เห็นสักครู่ว่าภาพถ่ายเป็นอย่างไร
 - 4) PictureCallback jpeg: เป็น Callback สำหรับส่งข้อมูลภาพถ่ายที่ได้จริงๆ โดยข้อมูลที่ส่งมาก็เป็นข้อมูลภาพ JPEG เพื่อนำไปสร้างเป็นไฟล์ภาพ
- คำสั่งในการโฟกัสกล้อง คือ

```
Camera mCamera = Camera.open();
mCamera.autoFocus(AutoFocusCallback cb);
```

รูปที่ 2.30 คำสั่งที่ใช้ในการ Focus กล้อง

คำสั่งในการเปิด-ปิดแฟลชของกล้อง จำเป็นต้องใช้ Class Parameters ของกล้อง ซึ่งสามารถเขียนได้ดังนี้

```
Camera.Parameters = params;//ประกาศคลาส Camera.Parameter โดยมีตัวแปรชื่อ
params
params.setFlashMode(Parameters.FLASH_MODE_TORCH); //เชื่อมต่อไฟเปิดค้างเป็นไฟฉาย
params.setFlashMode(Parameters.FLASH_MODE_OFF);//เชื่อมต่อไฟปิด
```

รูปที่ 2.31 คำสั่งที่ใช้ในการเปิด-ปิดแฟลชของกล้อง

2.7.3 การจัดเก็บภาพที่ถ่ายลงใน SD Card

```
File imagesFolder = new File(Environment.getExternalStorageDirectory(),
"DCIM/CameraRemote");//คำสั่งสร้างไฟล์ไปเก็บไว้ใน SD card โดยลงท้ายด้วยชื่อ directory
imagesFolder.mkdirs();//คำสั่งบันทึกภาพ
SimpleDateFormat sd = new SimpleDateFormat("yyyyMMdd-hhmmss");//เป็นการตั้งค่า
Format ของข้อมูลภาพ
```

รูปที่ 2.32 คำสั่งที่ใช้จัดเก็บภาพถ่ายลงในหน่วยความจำภายนอก

2.7.4 การเชื่อมต่อกันแบบ TCP/IP เพื่อส่งข้อมูลภาพและตัวอักษร

ในการส่งข้อมูลจะใช้การส่งข้อมูลแบบ TCP/IP เป็นโพรโทคอลในการส่งข้อมูล ซึ่งจะได้ภาพและตัวอักษรแล้วส่งกันระหว่างมือถือ 2 เครื่องและระหว่างโทรศัพท์มือถือ Smart Phone กับ Wifly Module RN-XV ซึ่งมีรายละเอียดดังนี้

ในการส่งข้อมูลจะต้องมีการเชื่อมต่อกันก่อน ซึ่งในการเชื่อมต่อต้องใช้ Class Socket ซึ่งจะทำหน้าที่เป็นตัวกำหนด IP และ Port ในการเชื่อมต่อ

```
socket s;//ประกาศใช้คลาส Socket โดยตัวแปรชื่อ s
Socket s = new Socket(etxtIP.getText().toString(), TCP_SERVER_PORT) ;//ประกาศคลาส
พร้อมประกาศใช้ตัวแปรเพื่อรับค่า IP/PORT
s.connct((new InetSocketAddress(InetAddress.getByName(ip), port_Number)),
timeconnection);//เป็นการเชื่อมต่อโดยกำหนดให้ IP เป็นไปตามตัวแปร ip โดย
Port_Number คือเลขพอร์ต ส่วน timeconnection คือเวลาที่ใช้ในการเชื่อมต่อ มีหน่วยเป็น ms
s.close();//เป็นคำสั่งใช้ปิดการเชื่อมต่อ
```

รูปที่ 2.33 คำสั่งที่ใช้ในการเชื่อมต่อ Socket (IP/Port)

เมื่อเชื่อมต่อได้ อีกส่วนหนึ่งคือการสร้าง Buffer เพื่อมาใช้เก็บข้อมูลก่อนที่จะส่งข้อมูลออกไปเป็นบางส่วนๆ ไปเรื่อยๆ トラバドที่ยังมีการเชื่อมต่อ

```
OutputStream out = s.getOutputStream();//ให้ s มารับข้อมูลเก็บไว้ในตัวแปร out
out.write(str.getBytes());//ใช้ตัวแปร out เพื่อเขียนข้อมูลที่ได้รับจาก str โดยรับมาเป็น ไบต์
out.flush ();//ใช้เคลียร์ค่าที่ค้างอยู่ใน Buffer เพื่อให้สามารถรับข้อมูลที่ถูกส่งเข้ามาใหม่ได้
```

รูปที่ 2.34 คำสั่งที่ใช้ในการสร้างเขียนข้อมูลลง Buffer และส่งข้อมูลออกไป

2.8 หลักการ Multithread ในโปรแกรมภาษา Java

เนื่องจากการส่งข้อมูลระหว่าง 2 อุปกรณ์ (IP) จึงต้องนำความรู้เรื่อง thread มาแบ่งการทำงานของโปรแกรมให้สามารถทำได้ 2 ภาระงาน ในโปรแกรมหรือ Application เดียวกัน ซึ่งมีหลักการดังนี้

2.8.1 ความแตกต่างของ Process กับ Thread

Process คือ สิ่งทีระบบปฏิบัติการประมวลผลโดยใช้ทรัพยากรของระบบ เช่น หน่วยความจำ เป็นต้น ทั้งนี้ Process จะทำให้ Application แต่ละตัวแยกออกจากกัน

Thread คือ การประมวลผลที่ทำให้ Application นั้นเกิดการดำเนินงานโดยในหนึ่ง Application สามารถมีได้หลายๆ Thread ที่สามารถใช้ข้อมูลเดียวกันประมวลผลไปพร้อมๆ กัน เรียกกรณีแบบนี้ว่า Multithread

2.8.2 ความแตกต่างระหว่าง Multitasking กับ Multithread

Multitasking คือ ระบบปฏิบัติการที่อนุญาตให้ผู้ใช้สามารถส่งโปรแกรมให้ระบบทำงานได้มากกว่าหนึ่งโปรแกรมพร้อมกัน ซึ่งโปรแกรมแต่ละโปรแกรมที่รันอยู่ในระบบปฏิบัติการจะสร้าง Process ขึ้น ระบบปฏิบัติการแบบ Multitasking จะมี Process หลายๆ Process เข้าคิวรอการทำงาน โดยระบบปฏิบัติการจะกำหนดลำดับของการทำงานของ Process เอง

Multithread คือ การทำงานหลายงานพร้อมกันโดยแต่ละงานจะเรียกว่า เธรด (Thread) ซึ่งเป็นแบบ Lightweight Process แตกต่างจาก Process ที่ทำงานภายใต้ระบบปฏิบัติการแบบ Multitasking ตรงที่ Process แต่ละ Process จะมีความเป็นอิสระต่อกัน แต่เธรดแต่ละเธรดอาจใช้ข้อมูลร่วมกันได้ โปรแกรมแบบเธรดจะมีหลักการการทำงานที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับระบบปฏิบัติการและจำนวน CPU

2.8.3 การสร้าง Thread

การสร้าง Thread ใน Java สามารถทำได้ 2 วิธีดังนี้

1) การ Extends คลาสที่ชื่อ Thread

สามารถสร้างได้โดยการสืบทอดคลาสที่ชื่อ Thread ซึ่งต้องมีการ Override เมธอด Run() โดยมีรูปแบบดังนี้

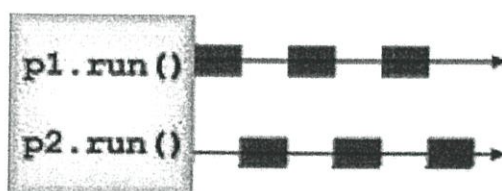
```
public class ClassName extends Thread
{
```

```

public void run()
{
    [statements]
}

```

การเรียกใช้เมธอด `p1.start()` และ `p2.start()` จะเป็นการส่ง Object แบบ เธรดทั้งสองเข้าไปจองคิวกับตัวจัดตารางเวลาเธรด ซึ่ง CPU จะแบ่งเวลาให้ Object ทั้งสอง มีโอกาสสลับกันทำงาน โดยการรันคำสั่งในเมธอด `run()` ของ Object



รูปที่ 2.35 การทำงานของโปรแกรมแบบ Multithread

2) การ Implements อินเตอร์เฟส Runnable

สามารถสร้างได้โดยการ Implements อินเตอร์เฟส Runnable ซึ่งเมธอดที่เขียนบล็อกรันคำสั่ง คือ เมธอด `run()` โดยมีรูปแบบดังนี้

```

public class ClassName implements Runnable
{
    public void run()
    {
        [statements]
    }
}

```

คำสั่งที่อยู่ในเมธอด `run()` คือ ชุดคำสั่งที่ต้องการให้โปรแกรมทำงานในลักษณะแบบ เธรดโปรแกรม Java ที่รัน Object ของคลาสแบบเธรด ในลักษณะนี้จะต้องมีการสร้าง Object ของคลาสที่ชื่อ Thread ก่อน

Constructor ของคลาสที่ชื่อ Thread จะมี Argument เพื่อรับ Object ของคลาสที่ Implements อินเตอร์เฟส Runnable โดยมีรูปแบบดังนี้

```
public Thread(Runnable obj)
```

ภายหลังจากที่สร้าง Object ของคลาสแบบเธรดแล้ว สามารถให้ Object (obj) เริ่มทำงานได้ โดยการเรียกใช้เมธอด Start() ที่อยู่ในคลาสที่ชื่อ Thread

คลาสที่ชื่อ Thread จะลงทะเบียน Object ลงในตัวจัดตารางเวลาเธรดและเมื่อ CPU รันโปรแกรมของ Object คำสั่งในเมธอด Run() จะถูกสั่งงานตามลำดับ

แต่ในการเขียนแอนดรอยด์จะใช้หลักการเดียวกัน แต่เขียนคำสั่งเป็น

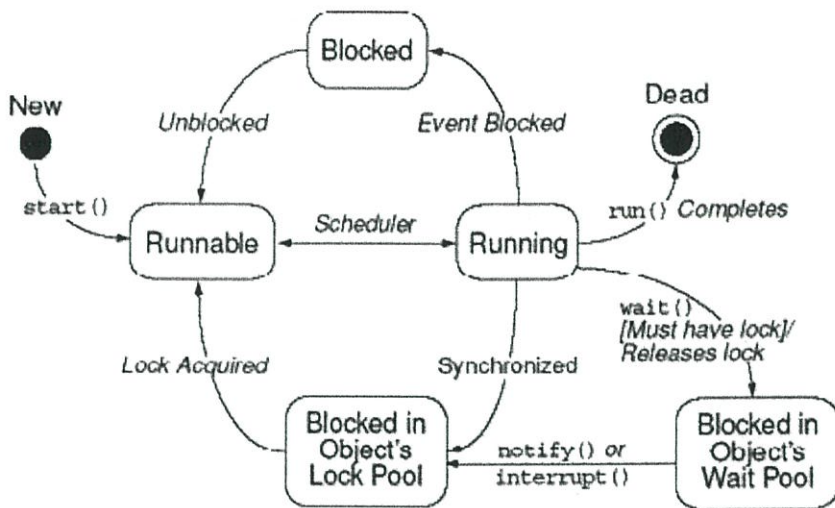
```
Runnable NameThread = new Runnable()
{
    Public void run()
    {
        [statement]; //คำสั่งที่จะทำงานของ Application
    }
};
new Thread(NameThread).start(); //คำสั่งในการเริ่มทำงานของ Thread
```

ในการใช้งาน Thread จะมี Method ที่สำคัญๆ ดังนี้

void start() คือ เมธอดที่เรียกใช้งานเพื่อกำหนดให้ Object แบบเธรดเริ่มทำงานโดยเข้าสู่สถานะ Runnable

void run() คือ เมธอดที่ถูกเรียกใช้เมื่อ CPU รันโปรแกรม Object แบบเธรด

static void sleep (long mills) คือ เมธอดที่กำหนดให้ Object แบบเธรดที่กำลังรันอยู่หยุดการทำงานชั่วคราวเป็นมิลลิวินาที โดยจะเข้าสู่สถานะ Blocked เพื่อเปิดโอกาสให้เธรดอื่นๆ ที่รออยู่ในสถานะ Runnable ได้มีโอกาสรัน



รูปที่ 2.36 ช่วงชีวิต (การทำงาน) ของ Thread

2.9 ระบบอัลตราโซนิก (Ultrasonic)

หมายถึง คลื่นเสียงที่มีความถี่สูงเกินกว่าที่มนุษย์จะได้ยิน โดยทั่วไปแล้วหูของมนุษย์โดยเฉลี่ยจะได้ยินเสียงสูงประมาณ 15 KHz คนที่อายุยังน้อยจะได้ยินเสียงที่มีความถี่สูงกว่านี้ได้ ดังนั้นโดยปกติคำว่าอัลตราโซนิกจึงหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20 KHz ขึ้นไป จะสูงขึ้นจนถึงเท่าใดไม่ได้ระบุจำกัดเอาไว้ สาเหตุที่มีการนำเอาคลื่นย่านอัลตราโซนิกมาใช้เพราะเป็นคลื่นที่มีทิศทางทำให้สามารถเล็งคลื่นเสียงไปยังเป้าหมายที่ต้องการได้โดยเจาะจง ซึ่งเป็นคุณสมบัติของคลื่นอย่างหนึ่ง ยิ่งคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะยิ่งสั้นลง เช่น คลื่นความถี่ 300 Hz ในอากาศจะมีความยาวคลื่นประมาณ 1 เมตร ซึ่งจะยาวกว่าช่องที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียง โดยทั่วไปคลื่นจะหักเบนที่ขอบด้านนอกของตัวกำเนิดเสียงทำให้เกิดการกระจายทิศทางของคลื่น แต่ถ้าความถี่สูงขึ้นจนอยู่ในย่านอัลตราโซนิก เช่น 40 KHz จะมีความยาวคลื่นในอากาศประมาณ 8 มิลลิเมตร ซึ่งเล็กกว่ารูเปิดของตัวกำเนิดเสียง ทำให้คลื่นเสียงไม่มีการเลี้ยวเบนที่ขอบจึงพุ่งออกมาเป็นลำแคบๆ เรียกว่า มีทิศทาง จึงสามารถนำไปใช้งานได้หลากหลาย เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic remote control) เครื่องล้างอุปกรณ์ (Ultrasonic cleaner) โดยให้น้ำสั่นที่มีความถี่สูง เครื่องวัดความหนาของวัตถุโดยส่งเกตระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล ใช้ในการหาตำแหน่งอวัยวะบางส่วนในร่างกาย และใช้ทดสอบการรั่วไหลของท่อ เป็นต้น

ซึ่งความถี่ที่ใช้จะขึ้นอยู่กับการใช้งาน เช่น คลื่นเสียงต้องเดินทางผ่านอากาศแล้วความถี่ที่ใช้จะจำกัดอยู่เพียงไม่เกิน 50 KHz เพราะที่ความถี่สูงขึ้นไปอากาศจะดูดกลืนคลื่นเสียง

เพิ่มขึ้นทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ต้องการรัศมีทำการสั้นๆ ก็อาจใช้ความถี่ในช่วง 1 MHz ถึง 10 MHz ขณะที่ความถี่เป็น GHz มีใช้กันในหลายๆ การใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่อากาศ

อุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไป-มา ทำให้เกิดคลื่นเสียงย่านอัลตราโซนิกกระจายไปในอากาศ หรือแปลงพลังงานทางกลให้มาเป็นพลังงานในรูปอื่น เรียกว่า Ultrasonic Transducer











2.9.1 ตัวส่ง

คือ Ultrasonic Transducer ที่ออกแบบเจาะจงมาให้แปลงคลื่นสัญญาณไฟฟ้าให้กับตัว โดยให้ออกมาเป็นคลื่นเสียงย่านอัลตราโซนิก หน้าที่ของตัวส่งจะคล้ายๆ กับเป็นลำโพง

2.9.2 ตัวรับ

คือ Ultrasonic Transducer ที่ออกแบบเจาะจงมาให้แปลงคลื่นเสียงย่านอัลตราโซนิกที่มาจากกระทบตัว ให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจะคล้ายๆ กับเป็นไมโครโฟน

ด้วยเหตุนี้เวลาเขียนสัญลักษณ์ของ Ultrasonic Transducer จึงนิยมเขียนตามหน้าที่คือ ถ้าเป็นตัวส่งจะเขียนสัญลักษณ์เป็นลำโพงและถ้าเป็นตัวรับจะเขียนสัญลักษณ์เป็นไมโครโฟน ดังรูป

แบบที่	ตัวส่ง	ตัวรับ
1		
2	 T _x	 R _x
3	 T _x	 R _x
4		
5		

รูปที่ 2.37 การเขียนสัญลักษณ์ของ Ultrasonic Transducer แบบต่างๆ

Ultrasonic Transducer แบบเซรามิกจะมีค่าความถี่เรโซแนนซ์ตั้งแต่ 23 KHz จนถึง 40 KHz โดยความถี่ 40 KHz เป็นรุ่นที่นิยมใช้มากที่สุดเพราะมีทิศทางดีกว่า

2.9.3 ข้อควรรู้ในการใช้งานตัวส่งและตัวรับ

เนื่องจากรายละเอียดต่างๆ ของ Ultrasonic Transducer หาได้ยาก จึงสรุปสิ่งที่ควรรู้ในขั้นต้นของ Ultrasonic Transducer เพื่อเป็นแนวทางในการใช้งานดังนี้

(1) ไม่ควรให้ Ultrasonic Transducer ได้รับการกระทบหรือตกจากที่สูงเพื่อป้องกันโครงสร้างภายในไม่ได้รับความเสียหาย

(2) ทรานสดิวเซอร์ที่มีขายกันโดยทั่วไปจะทนแรงดันตกคร่อมสูงสุดได้ไม่เกิน 20 Volts ดังนั้นขนาดของสัญญาณที่ป้อนให้กับ Ultrasonic Transducer ควรอยู่ในขีดจำกัด

(3) ความถี่เรโซแนนซ์ (ความถี่ที่ทำงานได้อย่างมีประสิทธิภาพสูงสุด) ของ Ultrasonic Transducer เป็น 40 KHz โดยทั่วไปจะผิดพลาดไม่เกิน ± 1 KHz และมีแถบความถี่ (Bandwidth) สำหรับตัวส่งประมาณ 4.5 KHz และสำหรับตัวรับประมาณ 5.0 KHz โดยแถบความถี่ของตัวรับจะกว้างกว่าของตัวส่งอยู่เล็กน้อย เพื่อให้แน่ใจว่าตัวรับสามารถรับความถี่ทั้งหมดที่ออกมาจากตัวส่งได้

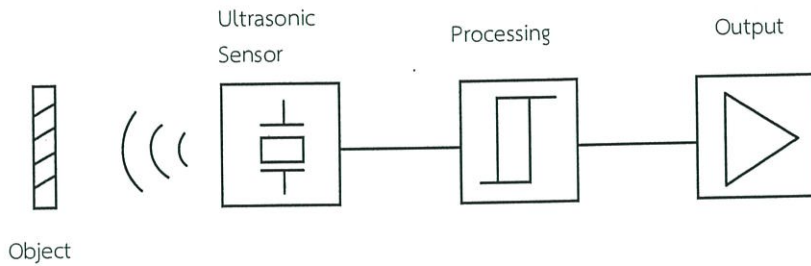
(4) อุณหภูมิที่ Ultrasonic Transducer สามารถใช้งานได้ควรอยู่ใน -20°C ถึง $+60^{\circ}\text{C}$

(5) ทั้งตัวส่งและตัวรับจะมีทิศทางคล้ายคลึงกัน คือ ที่ตำแหน่งเบนจากแนวแกนของตัวส่งไปประมาณ 30° ความแรงของคลื่นเสียงที่ถูกส่งออกไปจะลดลงจากแนวแกนประมาณ 10 dB ในทำนองเดียวกันถ้าคลื่นเสียงพุ่งเข้ามาในแนวที่เบี่ยงเบนไปจากแนวแกนขอตัวรับไปประมาณ 30° ความไวหรือขนาดแรงดันที่ออกมาจะลดลงประมาณ 10 dB ด้วย ดังนั้นในการควบคุมระยะไกลในที่โล่งจึงควรให้ทั้งตัวส่งและตัวรับอยู่ในแนวที่พุ่งตรงเข้าหากันให้มากที่สุด ในกรณีที่อยู่ในห้องอาจจะเบี่ยงเบนจากกันได้มาก เพราะคลื่นเสียงอัลตราโซนิกสามารถสะท้อนกับกำแพง พื้น และวัตถุที่อยู่ในห้อง ทำให้คลื่นเสียงเข้าไปหาตัวรับได้หลายทาง

(6) ในกรณีที่ใช้งานตัวรับ จะต้องมิตัวต้านทานต่อขนานกับตัวรับเพื่อทำหน้าที่เป็นโหลด ตามปกติแล้วตัวต้านทานตัวนี้ควรมีค่าอยู่ในช่วง 10 kW - 100 kW จากการทดลองพบว่าถ้าเปลี่ยนโหลดจาก 100 kW มาเป็น 10 kW ความไวจะลดลงประมาณ 10 ถึง 20 dB แต่แถบความถี่จะกว้างขึ้น ถ้าใช้ค่าความต้านทานต่ำลงความถี่เรโซแนนซ์จะลดลงไปจากที่ระบุไว้ ถ้าการใช้งานมีสัญญาณรบกวนมากควรใช้โหลดที่มีความต้านทานสูง เพื่อให้ตัวส่งมีความไวสูงและมีแถบความถี่แคบ

2.9.4 หน้าที่และการทำงานของ Ultrasonic Sensor

โดยจะมีรูปแบบต่างๆ ประกอบด้วยตัวตรวจจับด้วยคลื่นอัลตราโซนิก ชุดส่งสัญญาณ ชุดประมวลผล และชุด Output

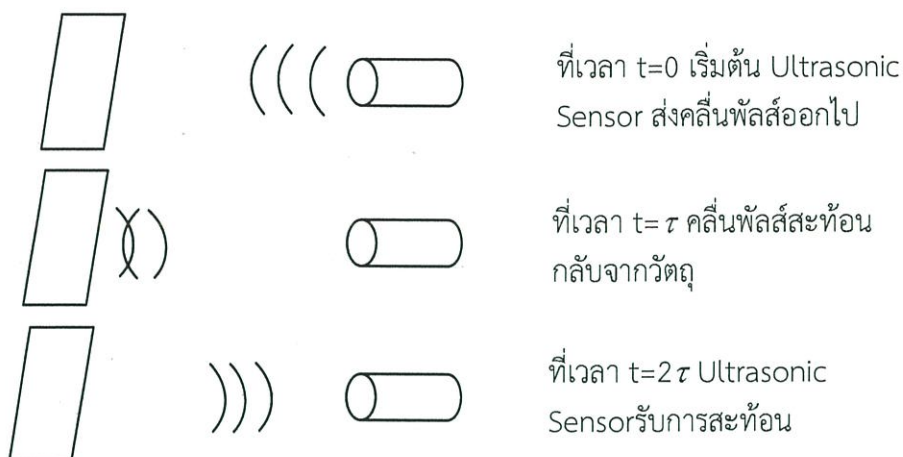


รูปที่ 2.38 หลักการทำงานของ Ultrasonic Sensor

จะใช้เป็นภาครับและภาคส่ง อาจมีระบบซึ่งประกอบด้วยส่วนหลักๆ อยู่ 2 ส่วน ในระหว่างการทำงาน Ultrasonic Sensor จะทำการส่งสัญญาณเสียงที่เรียกว่า Sound parcels ให้กับขบวนการทางอิเล็กทรอนิกส์ของเวลาทำงานไปเรื่อยๆ จนกระทั่งมีการรับการสะท้อนครั้งแรกเกิดขึ้น

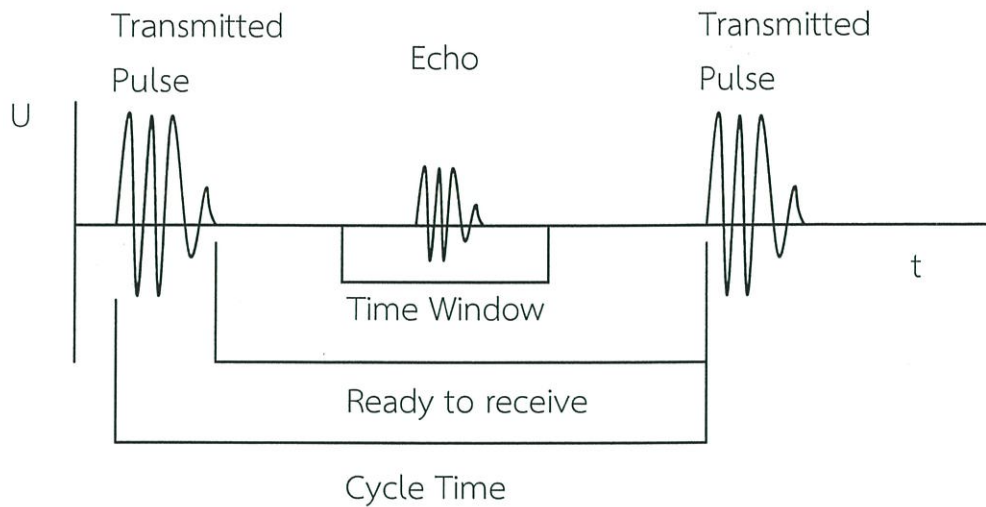
2.9.5 วงจรส่ง/รับ

สำหรับการทำงานเป็นวงจรของ Ultrasonic Sensor จะส่งผ่านคลื่นพัลส์เสียงในช่วงเวลาสม่ำเสมอหรือเวลาที่เปลี่ยนแปลง คลื่นเสียงที่ปล่อยออกมาจะถูกสะท้อนได้โดยวัตถุที่เหมาะสม โดย Ultrasonic Sensor และระบบการทำงานจะรับการสะท้อนของคลื่นเสียงที่สะท้อนกลับมา ความกว้างของคลื่นพัลส์เสียงอยู่ในช่วง 2-200 ไมโครวินาที



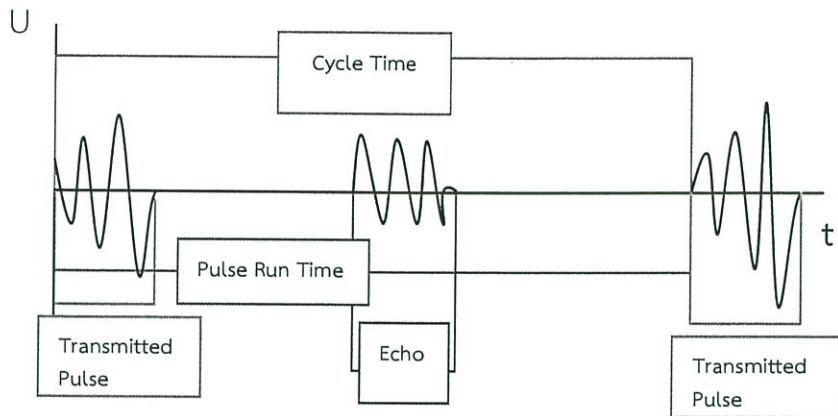
รูปที่ 2.39 Ultrasonic Sensor วงจรส่ง/รับ

เวลาในการเดินทางของคลื่นพัลส์เสียงเป็นการวัดระยะห่างจากวัตถุ โดยจะขึ้นอยู่กับชนิดของ Ultrasonic Sensor ระยะห่างนี้นำไปแสดงในรูปของ Analogue Signal เช่น 0-20 mA Logic Signal เช่น สัญญาณลอจิก 8 bit และ Serial Interface RS232 หรือการเปรียบเทียบค่าอ้างอิงในรูปของสวิทช์พัลส์ที่เรียกว่า Time Frame เนื่องจากขบวนการดำเนินเป็นไปตามเวลาที่คลื่นสะท้อนเดินทาง ไม่ได้เป็นไปตามความเข้มของคลื่นสะท้อน จึงจัดได้ว่า Ultrasonic Sensor มีข้อดีเหนือกว่า Optical Sensor ดังนั้นคุณลักษณะการสวิทช์ไม่เปลี่ยนไปแม้ในสถานะที่การสะท้อนเป็นไปอย่างไม่ดี คลื่นสะท้อนที่อ่อนจะมีผลต่อความถูกต้องในการตรวจจับวัตถุ ซึ่งอาจทำให้ไม่สามารถทำการตรวจจับวัตถุได้เลย ความเร็วที่เปลี่ยนไปของคลื่นพัลส์เสียงจะมีผลกระทบต่อประสิทธิภาพการทำงานของสวิทช์ (ระยะทาง) โดยตรงต่อ Ultrasonic Sensor ซึ่งทำงานด้วยวงจรเวลาที่คงที่ เช่น $t = 20 \text{ ms}$ จะส่งคลื่นพัลส์เสียงออกมาอย่างสม่ำเสมอ ดังรูปที่ 2.40 ดังนั้นวงจรเวลาจะเป็นตัวกำหนดช่วงและวงจรการทำงานของสวิทช์ Ultrasonic Sensor



รูปที่ 2.40 Ultrasonic Sensor ที่วงจรวจรเวลาที่

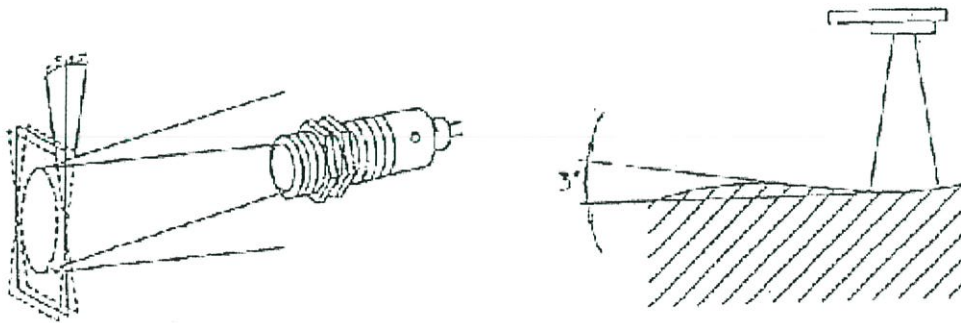
ยกตัวอย่าง คลื่นเสียงที่มีความเร็ว $v = 340 \text{ m/s}$ (ในสภาวะ 20 องศาเซลเซียส) ในช่วงเวลา $t = 20 \text{ ms}$ (50 Hz) จะเดินทาง $S = V \times t = 6.8 \text{ m}$ เนื่องจากระยะห่างระหว่าง Ultrasonic Sensor และวัตถุที่ได้จากการทำงานของ Ultrasonic Sensor คิดที่ไปและกลับ จึงได้ระยะทางสูงสุดสำหรับวงจรวจรเวลานี้เป็น 3.4 m แอมพลิจูดของส่วนคลื่นเสียงและ Sensitivity ของตัวรับต้องมีการพิจารณาเลือกใช้ เพื่อให้คลื่นสะท้อนที่เดินทางมาถึงหลังจกเวลาของวงจรวจรเวลาที่กำหนดไปแล้วจะไม่ได้รับการตรวจจับ เนื่องจากคลื่นสะท้อนนั้นอ่อนมากแล้วสัญญาณคลื่นนี้จะทำให้สวิทช์ Ultrasonic Sensor มีการทำงานผิดพลาดหรือให้ข้อมูลที่ผิด เพื่อให้การตรวจจับวัตถุเป็นไปอย่างถูกต้อง วัตถุต้องอยู่หนึ่งเป็นเวลาเพียงพอสำหรับสะท้อนอย่างน้อยหนึ่งส่วนของคลื่นเสียงภายในขอบเขตที่ Ultrasonic Sensor จะทำงานได้รอบมากที่สุดของการสวิทช์ ซึ่งจะเปลี่ยนแปลงกับอัตราส่วนของวัตถุต่อที่ว่างและจะพิจารณาให้มีค่าน้อยกว่ารอบของความถี่ที่จุดนี้ เวลาที่ขยายจะสิ้นสุดระหว่างการส่งผ่านของพัลส์และการรับคลื่นสะท้อนแรกจะถูกนำไปใช้วัดสำหรับวงจรวจรเวลา เมื่อเวลาดำเนินไปเท่ากับเวลาที่คลื่นสะท้อนเดินทางไป-กลับสิ้นสุด ส่วนของคลื่นสะท้อนต่อไปจะถูกส่งออก การหยุดลงช่วงระยะเวลาพิเศษที่คลื่นทำขึ้นเพื่อลดสัญญาณรบกวน (Noise) ที่มาจากตรวจจับวัตถุมากกว่าหนึ่งระยะ โดย Ultrasonic Sensor สามารถถูกปรับให้เหมาะสมตามสภาพแวดล้อม หมายถึง สำหรับการตรวจจับวัตถุที่อยู่ไกลเวลาการเดินทางนานเป็นผลให้ต้องการความถี่ต่ำ ในทางตรงกันข้ามความถี่ของวงจรวจรจะเพิ่มขึ้นเมื่อวัตถุเข้าใกล้ Ultrasonic Sensor จะใช้เวลาสั้นลงและพลังงานที่ส่งออกไปสามารถปรับในช่วงเวลาของคลื่นเสียงที่ปล่อยออกไปในเวลาเพิ่มขึ้นของแอมพลิจูดเมื่อมีการสวิทช์



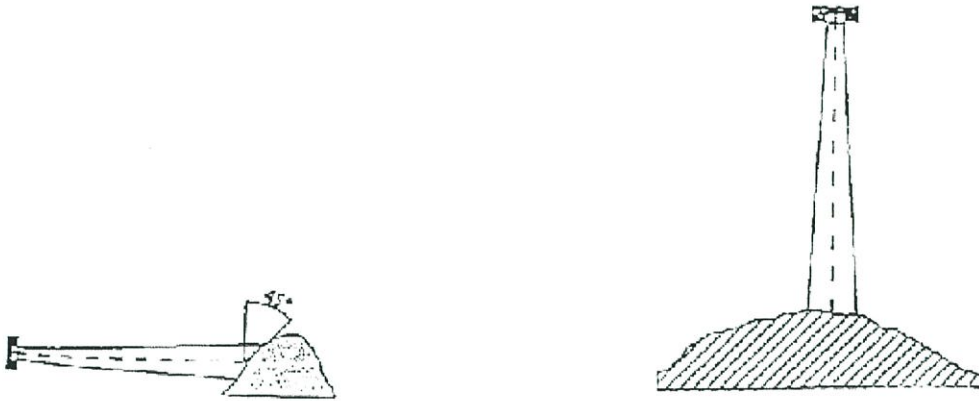
รูปที่ 2.41 Ultrasonic Sensor ที่วงจเวลาเปลี่ยนแปลงได้

2.9.6 การลดสัญญาณรบกวนและสภาวะการทำงาน

ผลที่เกิดจากคลื่นรบกวนและการแทรกสอดในการประยุกต์ใช้ Ultrasonic Sensor คือ การตรวจจับวัตถุได้ แต่ในระยะที่ใกล้กับ Ultrasonic Sensor และไม่สามารถตรวจจับวัตถุที่มีการสะท้อนได้ไม่ดี เนื่องจากความจริงที่ว่า คลื่นอัลตราโซนิกจะสะท้อนได้กับวัตถุเกือบทุกชนิดและง่ายต่อการเบี่ยงเบน วัตถุเหล่านั้นจะทำให้สวิทช์เปิด-ปิดเมื่อเข้าใกล้บริเวณที่ Ultrasonic Sensor สามารถตรวจจับได้

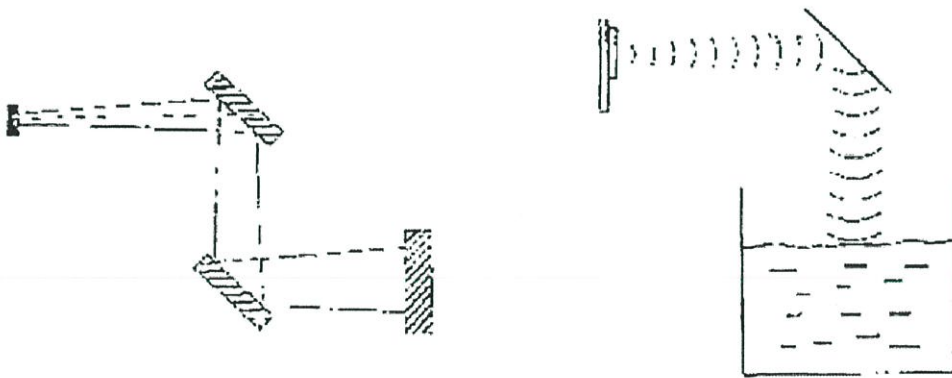


รูปที่ 2.42 Ultrasonic Sensor บนพื้นผิวตรง



รูปที่ 2.43 Ultrasonic Sensor กับการตรวจจับสิ่งของ

Ultrasonic Sensor สามารถนำมาเปียงเบนด้วยตัวสะท้อนอย่างง่าย ที่ทำจากวัสดุใดๆ พื้นที่ตรวจจับยังคงเท่าเดิม ทำให้ใช้กับตัวสะท้อนขนาดใหญ่ได้โดยใช้ตัวเปียงเบนไม่เกิน 2 ตัว ติดตั้งภายในทางเดินของคลื่นเสียงในแนวทางเดินตั้งฉากอย่างถูกต้อง



รูปที่ 2.44 Ultrasonic Sensor กับการเปียงเบนคลื่นเสียง

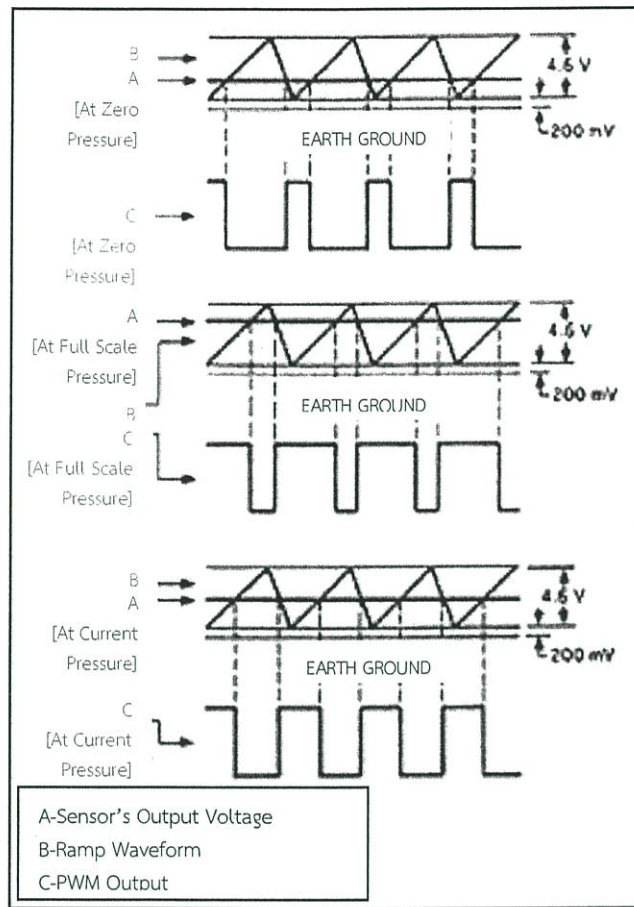
ด้วยวิธีนี้สามารถนำไปใช้ป้องกัน Sensor จากการใช้งานในสิ่งแวดล้อมที่เป็นอันตราย ต่อ Sensor และทำให้ลดคลื่นสะท้อนที่ไม่ต้องการจากวัตถุ การรวมคลื่นเสียงผ่านม่านกันหรือท่อ โดยปกติไม่สามารถทำได้เนื่องจากการหักเหของแสงบนขอบและกำแพง เพื่อหลีกเลี่ยงการแทรกสอดจากเครื่องมือที่ทำให้กำเนิดคลื่นเสียงอันอื่นๆ สัญญาณที่ได้รับจะถูกทดสอบความถี่ในตัวเอง วิธีการนี้ไม่สามารถทำให้สำเร็จได้เมื่อใช้ Sensor ชนิดเดียวกัน หรือมีย่านรบกวนกว้าง เช่น ใน

อากาศอัด เพื่อหลีกเลี่ยงผลกระทบระหว่าง Sensor จะต้องมีการติดตั้งที่ระยะปลอดภัย คลื่นรบกวนกับแอมพลิจูดขนาดใหญ่ สามารถปิดกั้น Sensor ไม่ให้รับคลื่นสะท้อนที่อ่อนกว่าคลื่นรบกวนได้ Sensor บางชนิดแก้ปัญหาการแทรกสอดนี้โดยสัญญาณเตือนที่ Output ที่แยกต่างหาก เพื่อชดเชยการเปลี่ยนแปลงความเร็วของเสียง จากอุณหภูมิที่เปลี่ยนแปลง Sensor บางชนิดจึงรวมเอา Sensor อุณหภูมิเข้าไปด้วย โดยจะทำการวัดอุณหภูมิบริเวณ Sensor และวัตถุ (ระยะสูงสุด 6 m) ขบวนการของสัญญาณจะทำให้เกิดคลื่นรบกวนที่เกิดขึ้นเป็นครั้งคราว เช่น คลื่นรบกวนเนื่องจากการเคลื่อนที่ของอากาศ หรือคลื่นรบกวนกับคลื่นอัลตราซาวด์ที่มีองค์ประกอบมาก (ในอากาศหรือเครื่องจักรกล) วิธีการนี้ Output ที่ได้จาก Sensor จะถูกกระตุ้นเมื่อจำนวนของคลื่นสะท้อนมีความเข้มเดียวกันและในการเดินทางเท่ากัน ข้อเสีย คือ ความถี่ของการทำงานค่อนข้างลดน้อยลงเมื่อทำการตรวจจับวัตถุที่เคลื่อนที่ผ่านอย่างรวดเร็ว ผ่านจุดที่ทำมุมตั้งฉากกับแกนของ Sensor ในกรณีนี้ความเร็วของวัตถุที่ยอมได้ จะกำหนดจากขนาดของวัตถุและระยะห่างจาก Sensor

หากวัตถุมีรูปร่างเป็นสี่เหลี่ยมจัตุรัสขนาดความยาวของแต่ละด้านเท่ากับ 100 mm. (แผ่นจานอ้างอิงมาตรฐานส่วนโค้ง B) ในระยะห่าง 2 m จาก Ultrasonic Sensor พร้อมกับย่านความไว 3 m นั้น ต้องการตรวจจับด้วย Ultrasonic Sensor จะต้องมีเวลาปฏิกิริยา 280 ms จะพิจารณาการใช้งานดังนี้ ระยะห่างช่องวัตถุในการตรวจจับ 1.24 m. เวลาค้นหาที่น้อยที่สุดที่ภายในช่วงการทำงานของ Ultrasonic Sensor (เวลาปฏิกิริยา) 280 ms. จะได้ความเร็วสูงสุด $v = 4.43 \text{ m/s}$ เพื่อที่จะตรวจจับแท่งไม้ที่มีขนาดเส้นผ่านศูนย์กลาง 25 mm. (ส่วนโค้ง D) ได้ ในระยะ 3 m จะคำนวณได้ดังนี้ $s = 0.19 \text{ m}$, $t = 280 \text{ ms}$, $v = 0.68 \text{ m/s}$

2.9.7 สัญญาณ PWM

ย่อมาจาก Pulse Width Modulation คือ การปรับความกว้างของพัลส์โดยการนำเอาสองสัญญาณมาเปรียบเทียบ คือ สัญญาณสามเหลี่ยมกับสัญญาณที่ต้องการปรับความกว้างของพัลส์ ดังรูปที่ 2.45



รูปที่ 2.45 สัญญาณไฟฟ้ากระแสตรง, สัญญาณสามเหลี่ยม และสัญญาณพัลส์ที่มีความกว้างคงที่

การสร้างสัญญาณ PWM สร้างโดยวงจร Comparator หรือวงจรเปรียบเทียบ ซึ่งวงจร Comparator ที่นิยมใช้ คือ Op-Amp ซึ่งเป็นพื้นฐานที่สามารถนำไปใช้งานได้

การสร้างสัญญาณ PWM เป็นการสร้างสัญญาณที่เกิดจากการผสมกันระหว่างสัญญาณรูปสามเหลี่ยม (Triangle wave) กับระดับสัญญาณไฟกระแสตรงหรือสัญญาณอื่นๆ ที่มีระดับสัญญาณไฟกระแสตรงเป็นตัวรองรับ ผลที่ได้จากการผสมสัญญาณดังกล่าวจะได้รูปคลื่นสัญญาณพัลส์ที่มี 2 สถานะ คือ ON กับ OFF เมื่อนำสัญญาณพัลส์ที่ได้ไปขับอุปกรณ์กำลัง เช่น หลอดไฟและมอเตอร์กระแสตรง จะได้ผลการควบคุม คือ เมื่อสัญญาณพัลส์มีสถานะ ON หลอดไฟจะติดสว่างเต็มที่ ถ้า OFF หลอดไฟจะดับ หากสัญญาณมีความต่อเนื่องตลอดจะได้การติด-ดับของหลอดไฟอย่างต่อเนื่อง และหากการติด-ดับมีความถี่สูงๆ จนสายตาสังเกตการติด-ดับได้ จะสังเกตเห็นได้เพียงแสงที่หรือลางหากทำให้ช่วงเวลา ON น้อยลง และในทางตรงกันข้ามหากทำให้ช่วงเวลา ON มากขึ้นเรื่อยๆ ก็จะทำให้หลอดไฟสว่างขึ้นเรื่อยๆ จนถึงสว่างเต็มที่เมื่อ ON ตลอด

รายละเอียดของสัญญาณ PWM ประกอบด้วยส่วนต่างๆ ดังนี้ คือ

(1) ความถี่ คือ จำนวนรูปคลื่นต่อวินาที (Cycle/Second) ในการใช้งานความถี่จะต้องคงที่ไม่เปลี่ยนแปลง ความถี่ที่เหมาะสมในการใช้งานควบคุมอุปกรณ์กำลังต่างๆ คือ ตั้งแต่ 400 Hz ถึง 10 KHz ส่วนความถี่ที่ใช้เป็นพาหะของสัญญาณเสียง (Audio) ควรจะสูงกว่า 50 KHz ถึง 1 MHz

(2) Duty Cycle เป็นค่าคาบเวลาช่วง ON ของ Cycle โดยคิดเป็นเปอร์เซ็นต์ของ Full-cycle

2.9.8 เสียง

(1) ธรรมชาติของเสียง เสียงเกิดจากการสั่นของวัตถุที่เป็นต้นกำเนิดเสียง เช่น การดีดสายกีตาร์ พลังงานในการดีดซึ่งเป็นพลังงานกล จะถูกถ่ายโอนให้กับสายกีตาร์ ทำให้สายกีตาร์สั่น พลังงานในการสั่นของสายกีตาร์จะเปลี่ยนเป็นพลังงานเสียงแผ่กระจายออกไปโดยรอบ พลังงานเสียงจะเคลื่อนที่ผ่านตัวกลางโดยการถ่ายโอนพลังงานไปยังโมเลกุลของตัวกลางนั้น เมื่อโมเลกุลของตัวกลางได้รับพลังงานโมเลกุลนั้นจะสั่นและถูกถ่ายทอดไปยังโมเลกุลตัวอื่นๆต่อไป การแผ่กระจายพลังงานเสียงออกไป ถูกส่งออกไปในลักษณะของคลื่นกล ซึ่งต้องอาศัยตัวกลางในการส่งผ่านพลังงาน ตัวกลางที่คลื่นเสียงผ่านได้มีทั้ง ของแข็ง ของเหลว และก๊าซ โดยตัวกลางที่เป็นของแข็งคลื่นเสียงผ่านได้ดีกว่าในของเหลว และ ก๊าซตามลำดับ โดยความถี่ของเสียงจะเท่ากับความถี่การสั่นของอนุภาคตัวกลาง(แบบซิมเปิลฮาร์มอนิก) และเท่ากับความถี่การสั่นของแหล่งกำเนิดเสียง

(2) อัตราเร็วของเสียง เนื่องจากเสียงเป็นคลื่นกล ซึ่งต้องใช้ตัวกลางในการส่งถ่ายพลังงานเสียงดังนั้นอัตราเร็วของเสียงจึงขึ้นอยู่กับสมบัติของตัวกลางดังนี้

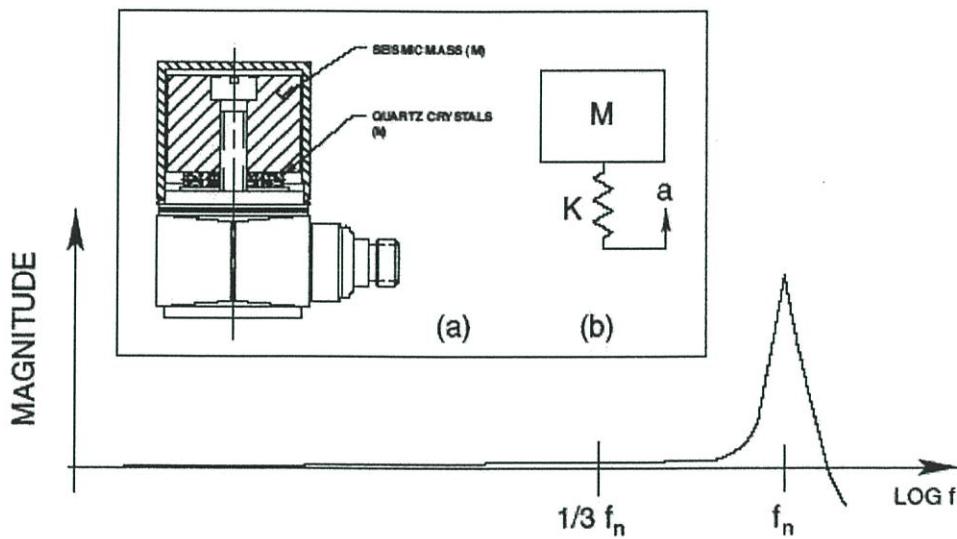
1. ชนิดของตัวกลาง เสียงเคลื่อนที่ได้เร็วในตัวกลางที่มีความหนาแน่นสูงดังนั้นเสียงจึงเคลื่อนที่ได้เร็วในตัวกลางที่เป็นของแข็ง และมีอัตราเร็วลดลงในของเหลวและก๊าซตามลำดับ

2. อุณหภูมิของตัวกลาง ถ้าตัวกลางมีอุณหภูมิสูงการสั่นสะเทือนของโมเลกุลจะมากขึ้นทำให้การเคลื่อนที่ของเสียงเร็วขึ้นด้วย สำหรับอัตราเร็วของเสียงในอากาศจากการทดลองพบว่า เสียงจะมีอัตราเร็วประมาณ 331 เมตร/วินาที ในอากาศที่อุณหภูมิ 0 องศาเซลเซียส จะอัตราเร็วจะเพิ่มขึ้น 0.6 เมตร/วินาทีทุก ๆ อุณหภูมิที่เพิ่มขึ้น 1 องศาเซลเซียส ดังนั้นในการคำนวณอัตราเร็วเสียงในอากาศที่ อุณหภูมิ t องศาเซลเซียส จึงหาได้จากสมการ $v = 331 + 0.6t$ เมื่อ v หมายถึง อัตราเร็วเสียงในอากาศที่ t องศาเซลเซียส มีหน่วยเป็นเมตร/วินาที (t หมายถึง อุณหภูมิของอากาศในหน่วย องศาเซลเซียส)

2.10 Accelerometer

2.10.1 โครงสร้างของ Accelerometer

จะประกอบด้วยสปริงและลูกตุ้มน้ำหนัก เมื่อมีการเคลื่อนที่ด้วยความเร่ง ลูกตุ้มน้ำหนักจะถูกกดไปอีกฝั่งตรงข้ามกับการเคลื่อนที่ สปริงก็ทำหน้าที่ดึงกลับเข้าที่อีกครั้งเมื่อหยุดการเคลื่อนที่ การเคลื่อนที่ด้วยความเร็วคงที่คือความเร่งเท่ากับศูนย์ ค่าที่วัดได้จะไม่เปลี่ยนแปลง

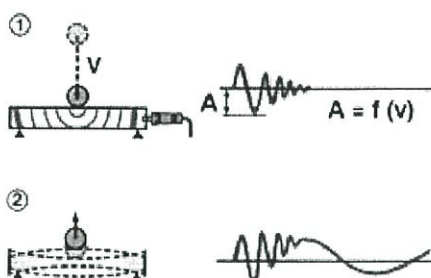


รูปที่ 2.46 โครงสร้างของ Accelerometer

ตัวเซ็นเซอร์ภายในที่จะใช้ในการตรวจวัดความเร่งของลูกตุ้มที่อยู่ในระบบนั้นมีหลายชนิด เช่น เพียโซอิเล็กทริก, สเตรนเกจ, ชนิดใช้แสงตรวจวัด, วัดแรงเฉือน เป็นต้น โดยที่สามารถแบ่งลักษณะการตรวจวัดได้ 2 ลักษณะ

1) การตรวจวัดการช็อก (Shock) และการสั่นสะเทือน (Vibration)

การช็อก คือ อัตราเร่งขนาดมหึมาที่เกิดขึ้นในช่วงเวลาสั้นๆ
การสั่นสะเทือน คือ อัตราเร่งขนาดเล็กที่เกิดขึ้นซ้ำกันไปเรื่อยๆ

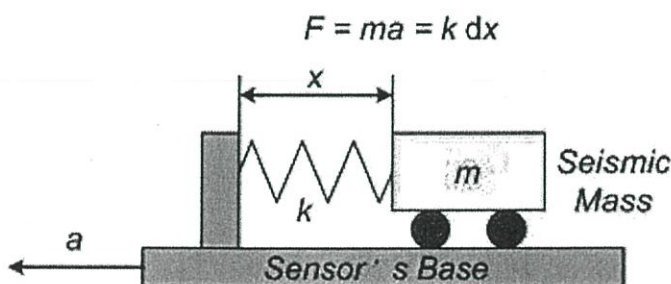


รูปที่ 2.47 การตรวจวัดการสั่นสะเทือนและตรวจวัดอัตราเร่งของวัตถุ

2) การตรวจวัดอัตราเร่งของวัตถุเพื่อนำข้อมูลไปใช้ในการระบุตำแหน่งความเร็ว และระยะทางที่ได้จากการเคลื่อนที่

2.10.2 ชนิดของมิเตอร์วัดความเร่ง

1) Seismic Mass Accelerometer อาศัยหลักการตรวจวัดระยะขจัดเชิงเส้นแล้วนำไปคำนวณหาอัตราเร่งที่เกิดขึ้นโดยเทคนิคดังกล่าวสามารถอธิบายง่ายๆ คือ วัตถุชิ้นหนึ่งจะมีความเร่งได้ ก็จะต้องมีแรงมากระทำ ยิ่งมีแรงมากระทำมากจะยิ่งมีความเร่งมาก ในขณะเดียวกันแรงต้านการเคลื่อนที่ก็จะมากด้วย นอกจากนี้เมื่อมีแรงมาทำให้วัตถุเกิดการเคลื่อนที่ ก็จะมีระยะขจัด ซึ่งก็จะแปรผันตรงกับแรงที่มากระทำที่วัตถุ ยิ่งแรงมากระยะขจัดยิ่งมาก จากความสัมพันธ์ดังกล่าวได้นำไปใช้เป็นหลักการพื้นฐานของ Seismic Mass Accelerometer ในการตรวจวัดอัตราเร่งของวัตถุในเทอมของระยะขจัดที่เกิดขึ้น



รูปที่ 2.48 โครงสร้างพื้นฐานของ Seismic Mass Accelerometer

โครงสร้างนี้มีมวล (m) เรียกว่า มวลตรวจการสั่นไหว (Seismic Mass) ยึดติดอยู่กับสปริงที่มีค่า Spring Constant เท่ากับ k และมวลนี้สามารถเคลื่อนที่ในแนวระดับได้ ซึ่งหลักการทำงาน คือ เมื่อตัว Sensor ถูกทำให้อัตราเร่งเกิดขึ้นจะส่งผลให้มวล (m) เคลื่อนที่ ซึ่งระยะที่

เคลื่อนที่ออกไปจะเป็นระยะขจัดเท่ากับ x และมีทิศทางตรงกันข้ามกับการเคลื่อนที่ของตัวมิเตอร์ ดังนั้นอัตราเร่ง (a) ของวัตถุสามารถคำนวณหาได้จากความสัมพันธ์ต่อไปนี้

$$a = xk/m \text{ โดยที่}$$

a คือ อัตราเร่งของวัตถุ หน่วย เมตร/วินาที

x คือ ระยะขจัดของมวล (m) หน่วย เมตร

k คือ ค่าคงที่ของสปริง หน่วย นิวตัน/เมตร

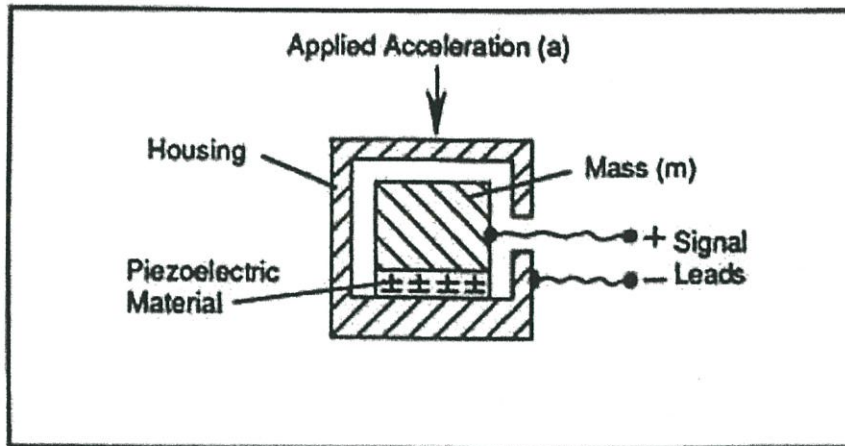
m คือ น้ำหนักของมวล (m) หน่วย กิโลกรัม

เมื่ออัตราเร่งของวัตถุมีค่าเพิ่มขึ้น ทำให้ระยะขจัดของมวล (m) มีค่าเพิ่มขึ้น เมื่ออัตราเร่งของวัตถุมีค่าลดลง ทำให้มวล (m) เคลื่อนที่ไปดันสปริง เมื่ออัตราเร่งของวัตถุหยุดลง ทำให้มวล (m) เคลื่อนที่กลับมาอยู่ตำแหน่งเดิม (ตำแหน่งอ้างอิง) แต่ในทางปฏิบัติสามารถวัดระยะขจัดของมวล (m) ได้โดยอาศัยมิเตอร์อีกชนิดหนึ่ง คือ มิเตอร์วัดระยะขจัดเชิงเส้น (LVDT, Potentiometer) ส่วนการวิเคราะห์หาค่าอัตราเร่งที่เกิดขึ้นสามารถคำนวณหาได้โดยใช้ Seismic Mass Accelerometer นี้จะนิยมใช้ในการตรวจวัดลักษณะการช็อคและลักษณะการสั่นสะเทือนที่มีความถี่ต่ำมากๆ เช่น ในเครื่องมือตรวจวัดแผ่นดินไหว หรือในเครื่องมือตรวจวัดการปะทุใต้ดินของภูเขาไฟ ฯลฯ

2) Piezoelectric Accelerometer

คุณสมบัติพื้นฐานทางไฟฟ้าของ Piezoelectric Crystal ถูกค้นพบโดย Pierre และ Jacques Curie ในราวปี ค.ศ.1880 ซึ่งเจ้า Piezoelectric Crystal นี้มันมีคุณสมบัติพิเศษ คือ เมื่อถูกแรงทางกลมากระทำจะสร้างประจุไฟฟ้าขึ้นมา โดยเป็นส่วนเดียวกับแรงกระทำนั้น ซึ่งจากคุณสมบัติพิเศษนี้ได้ถูกดัดแปลงนำไปใช้สร้างอุปกรณ์ต่างๆ มากมาย เช่น ใช้เป็นแบตเตอรี่จ่ายพลังงานไฟฟ้าให้กับนาฬิกาข้อมือดิจิทัลที่ใช้ทั่วไป และยังใช้สร้าง Piezoelectric Accelerometer ได้

โครงสร้างของ Piezoelectric Accelerometer ประกอบด้วย Seismic Mass ยึดติดกับ Piezoelectric Crystal และบรรจุอยู่ในตัวถังป้องกัน โดย Piezoelectric Crystal ที่นิยมนำมาใช้งาน ได้แก่ ผลึกควอตซ์ และผลึกโซเดียมโพตัสเซียมตาเตรต เพราะมีความทนทานต่อแรงกระทำ



รูปที่ 2.49 โครงสร้างพื้นฐานของ Piezoelectric Accelerometer

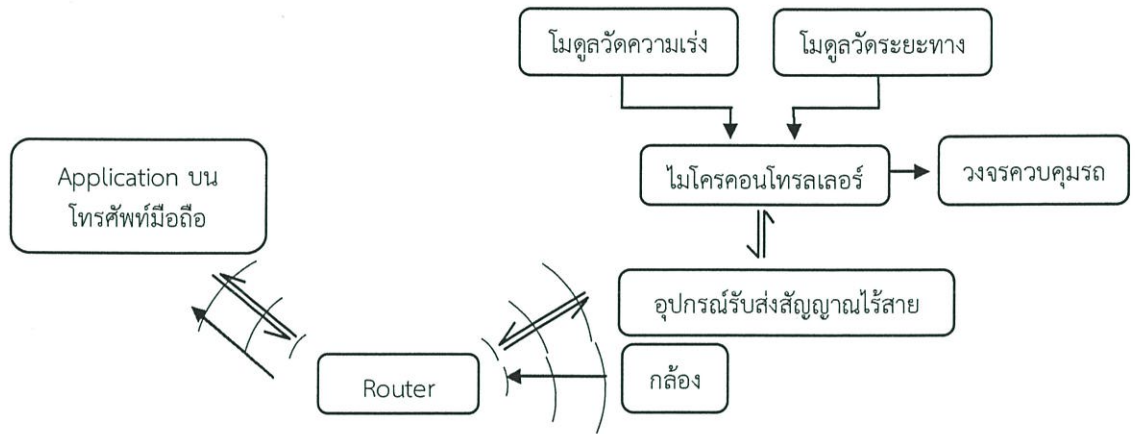
เมื่อ Seismic Mass (M) ถูกทำให้เกิดอัตราเร่งขึ้น (ถูกกด) จะส่งผ่านแรงกดไปกระทำกับ Piezoelectric Crystal ที่ถูกยึดติดอยู่ด้วยกัน ด้วยคุณสมบัติพิเศษนี้ จะทำให้ประจุไฟฟ้าถูกสร้างขึ้น และถูกสายนำสัญญาณออกไปยังเอาต์พุตของวงจร โดยที่ด้านเอาต์พุตจะต้องมีวงจรขยายประจุไฟฟ้า (Charge Amplifier) เพื่อขยายค่าประจุไฟฟ้าที่ได้ ให้เป็นแรงดันเอาต์พุตตามสัดส่วนของอัตราเร่งที่เกิดขึ้น จะแสดงผลด้วยโวลต์มิเตอร์ Piezoelectric Accelerometer ตอบสนองต่อทางด้านความถี่สูงได้ดี แต่ในทางกลับกันก็จะมีผลตอบสนองทางด้านความถี่ต่ำที่ไม่ดีนัก มีขนาดค่อนข้างเล็ก น้ำหนักเบา และสามารถใช้งานที่มีอัตราเร่งได้สูงถึง $250,000 \text{ ms}^{-2}$

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 หลักการทำงานของระบบ

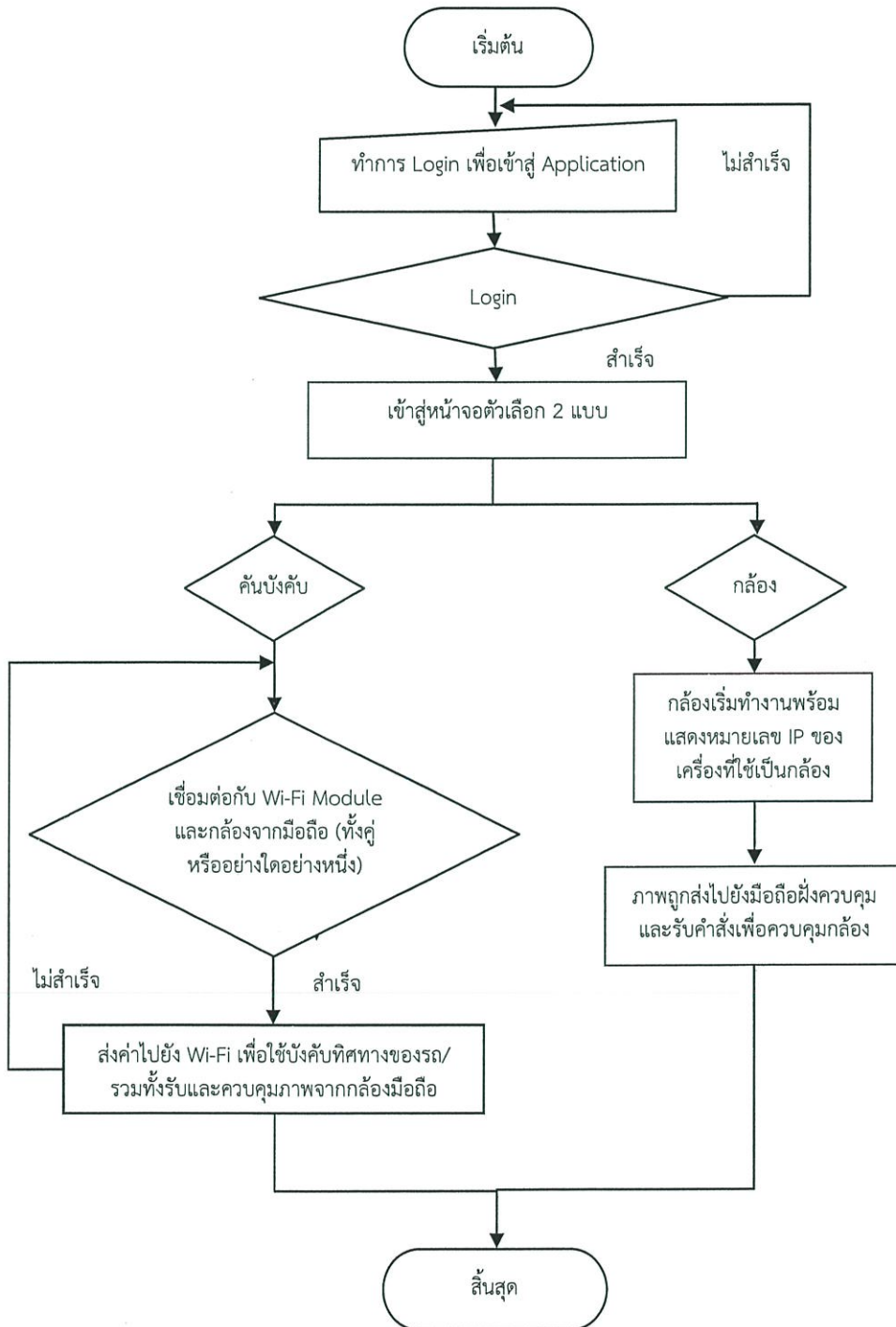


รูปที่ 3.1 Block Diagram โดยรวมของระบบ

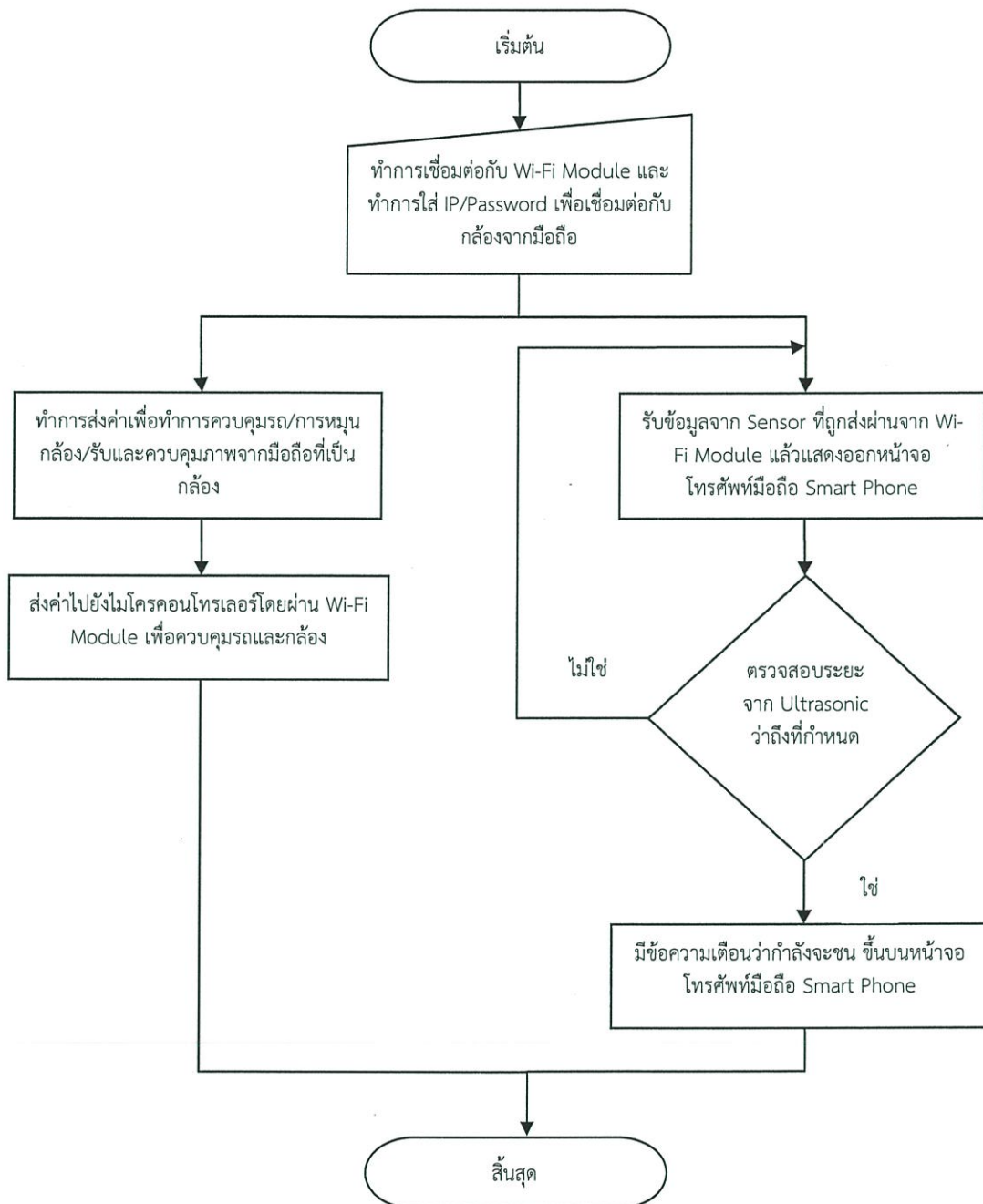
จากบล็อกไดอะแกรมการทำงานของระบบจะเริ่มจากการสร้างและการออกแบบ Application ลงบนโทรศัพท์มือถือ Smart Phone เพื่อควบคุมการทำงานของรถสำรวจ จากนั้นภาครับจะรับสัญญาณโดยอุปกรณ์รับ-ส่งสัญญาณแบบไร้สาย โดยสัญญาณที่รับได้จะถูกแปลงเป็นข้อมูลตัวอักษร จากนั้นไมโครคอนโทรลเลอร์จะทำหน้าที่รับข้อมูลตัวอักษรจากอุปกรณ์รับ-ส่งสัญญาณแบบไร้สาย และเก็บข้อมูลไว้ในบัพเฟอร์ของไมโครคอนโทรลเลอร์ จากนั้นจะอ่านข้อมูลและประมวลผลออกมาในรูปแบบบิตดิจิทัลจำนวน 6 bit ที่จะใช้เป็นบิตอินพุตให้กับวงจรขับมอเตอร์ต่อไป โดยมอเตอร์หนึ่งตัวจะใช้บิตในการควบคุม 3 bit

ในส่วนของการส่งข้อมูลจากไมโครคอนโทรลเลอร์เพื่อแสดงผลออกทางหน้าจอ Application บนโทรศัพท์มือถือ Smart Phone ข้อมูลที่ส่งจะประกอบไปด้วยข้อมูล 2 ชุด คือ ค่าระยะทางที่ Ultrasonic Sensor สามารถตรวจจับสิ่งกีดขวางได้ และค่ามุมเอียงของตัวรถ ข้อมูลจะถูกประมวลผลโดยไมโครคอนโทรลเลอร์และส่งข้อมูลนั้นออกทางพอร์ตอนุกรมเพื่อเข้าสู่โมดูล Wi-Fi จากนั้นโมดูล Wi-Fi จะส่งสัญญาณไปยังโทรศัพท์มือถือ Smart Phone ผ่านทาง Router ส่วนภาพจากกล้องจะถูกส่งมายังหน้าจอโทรศัพท์มือถือ Smart Phone โดยทั้งโมดูล Wi-Fi และโทรศัพท์มือถือ Smart Phone ต้อง Access เข้าอุปกรณ์ Router ก่อนการรับ-ส่งข้อมูล

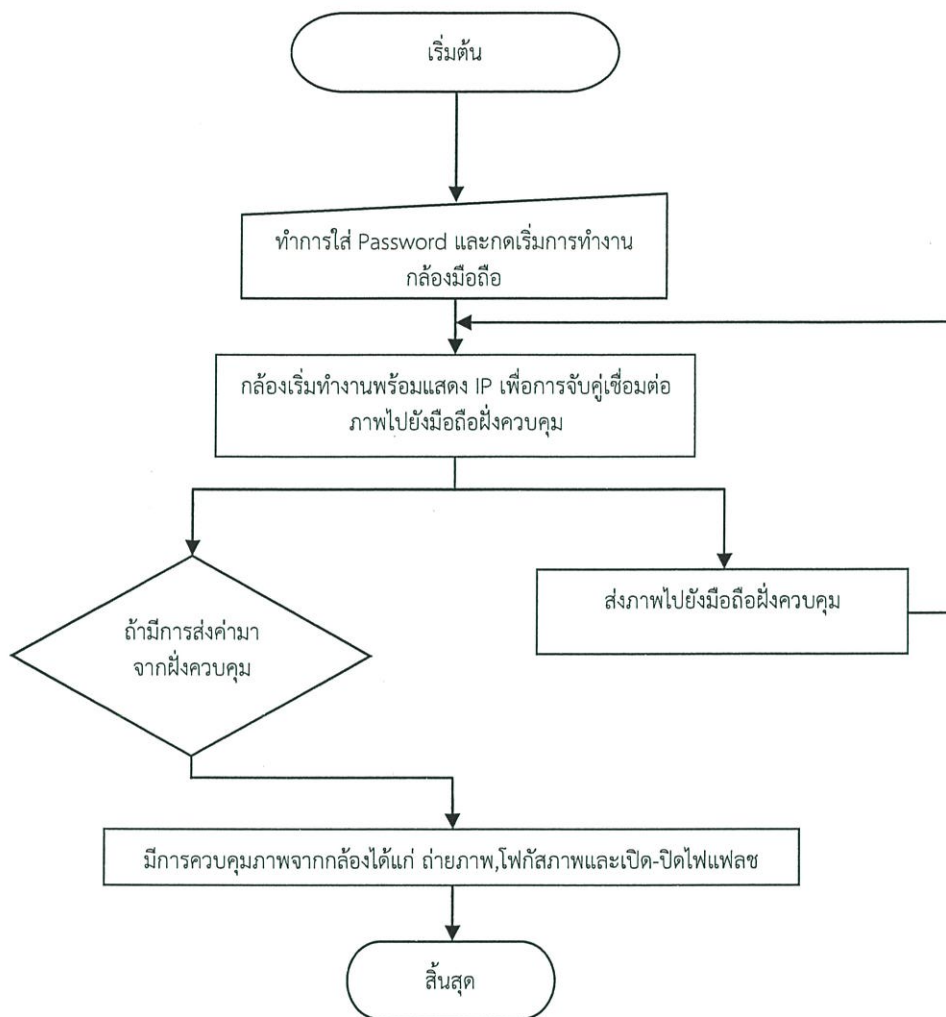
3.1.2 การออกแบบ Application



รูปที่ 3.2 Flowchart การทำงานโดยรวมของ Application



รูปที่ 3.3 Flowchart การทำงานในส่วนการควบคุมของ Application



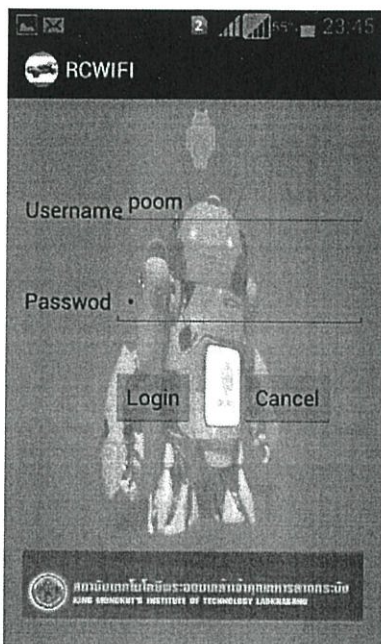
รูปที่ 3.4 Flowchart การทำงานในส่วนกล้องของ Application

เป็นการพัฒนา Application บนโทรศัพท์มือถือ Smart Phone เพื่อใช้ในการบังคับรถสำรวจโดยผ่านการเชื่อมต่อกับโมดูล Wi-Fi และทำการส่งภาพโดยโทรศัพท์มือถือ Smart Phone ที่ทำหน้าที่เป็นกล้องโดยการทำงานทั้งคู่จะทำงานคนละ IP กัน (โดย IP ของโมดูล Wi-Fi ได้ถูก fix ค่าไว้แล้ว) โดยใน Application ดังกล่าวจะแบ่งเป็น 2 ฟังก์ชัน คือ ฟังก์ชันควบคุมกับฟังก์ชันที่เป็นกล้อง เมื่อเข้าสู่หน้าต่างผู้ใช้งานจะมีการล็อกอินเพื่อให้สามารถเข้าไปสู่หน้าโปรแกรมหลักได้ โดย Application จะสามารถรับค่า Sensor จากไมโครคอนโทรลเลอร์มาแสดงผลออกหน้าจอโทรศัพท์มือถือ Smart Phone โดยเมื่ออยู่ภายในระยะห่างที่กำหนดจะสามารถแจ้งเตือนผู้ใช้ได้ โดยจะมีฟังก์ชันการใช้ในการควบคุมทั้งหมด 8 ฟังก์ชัน

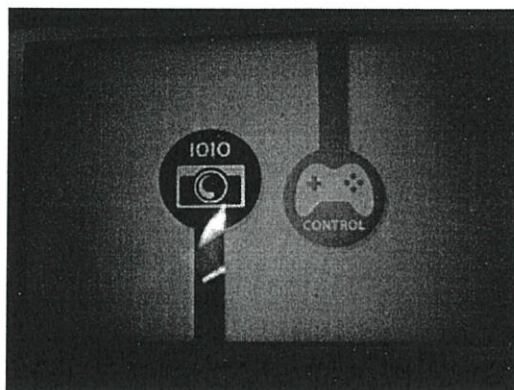
- 1) ฟังก์ชันเดินน้ารถ จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'F'
- 2) ฟังก์ชันถอยหลังรถ จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'B'
- 3) ฟังก์ชันหมุนซ้ายรถ จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'L'
- 4) ฟังก์ชันหมุนขวารถ จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'R'
- 5) ฟังก์ชันหยุดรถ จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'S' ทุกๆ การปล่อยปุ่มกด
- 6) ฟังก์ชันหมุนกล้องซ้าย จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'Q'
- 7) ฟังก์ชันหมุนกล้องขวา จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'T'
- 8) ฟังก์ชันหยุดหมุนกล้อง จะถูกกำหนดโดยการส่งข้อมูลตัวอักษร 'n'

ส่วนในด้านการใช้งานกล้องนั้นจะมีฟังก์ชันอยู่ 3 ฟังก์ชัน ได้แก่

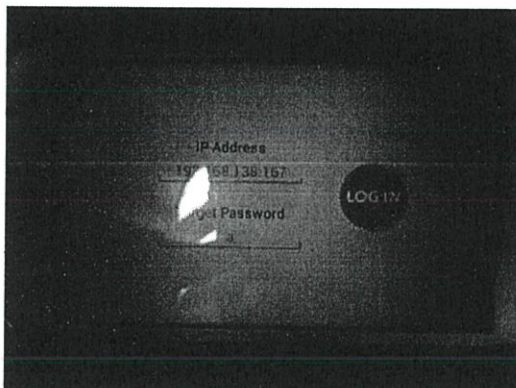
- 1) ฟังก์ชันถ่ายรูป ทำหน้าที่ถ่ายภาพนิ่งแล้วจัดเก็บไว้ในหน่วยความจำของเครื่องที่ทำหน้าที่เป็นกล้อง
- 2) ฟังก์ชัน Focus ทำหน้าที่ Focus ภาพจากกล้องให้ชัดเจนขึ้น
- 3) ฟังก์ชันเปิด-ปิดไฟแฟลช ทำหน้าที่ในการเปิด-ปิดไฟแฟลชของเครื่องที่ทำหน้าที่เป็นกล้อง



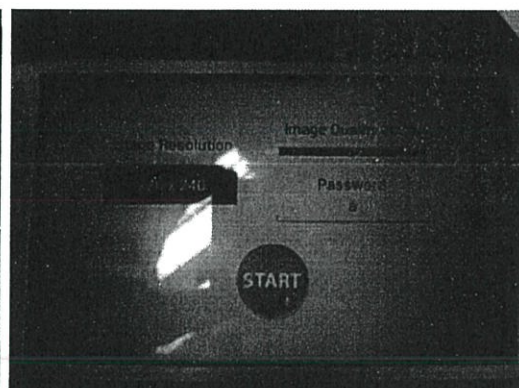
(ก)



(ข)



(ค)



(ง)

รูปที่ 3.5 ส่วนต่างๆ ของ Application

- (ก) หน้าต่างในการ Login ในการเข้าสู่ Application
- (ข) หน้าต่างในการเลือกฝั่งว่าจะบังคับหรือเป็นกล้อง
- (ค) หน้าต่างในการ Login เพื่อทำการบังคับ
- (ง) หน้าต่างในการ Login เข้าสู่กล้อง

3.1.3 การตั้งค่าโมดูล Wi-Fi สำหรับติดต่อกับโทรศัพท์มือถือ Smart Phone

เมื่อต้องการให้โมดูล Wi-Fi รับค่าจากโทรศัพท์มือถือ Smart Phone ต้องทำการตั้งค่าให้โมดูล Wi-Fi มีการเชื่อมต่อกับโทรศัพท์มือถือ Smart Phone โดยสามารถทำได้ 2 วิธี คือ

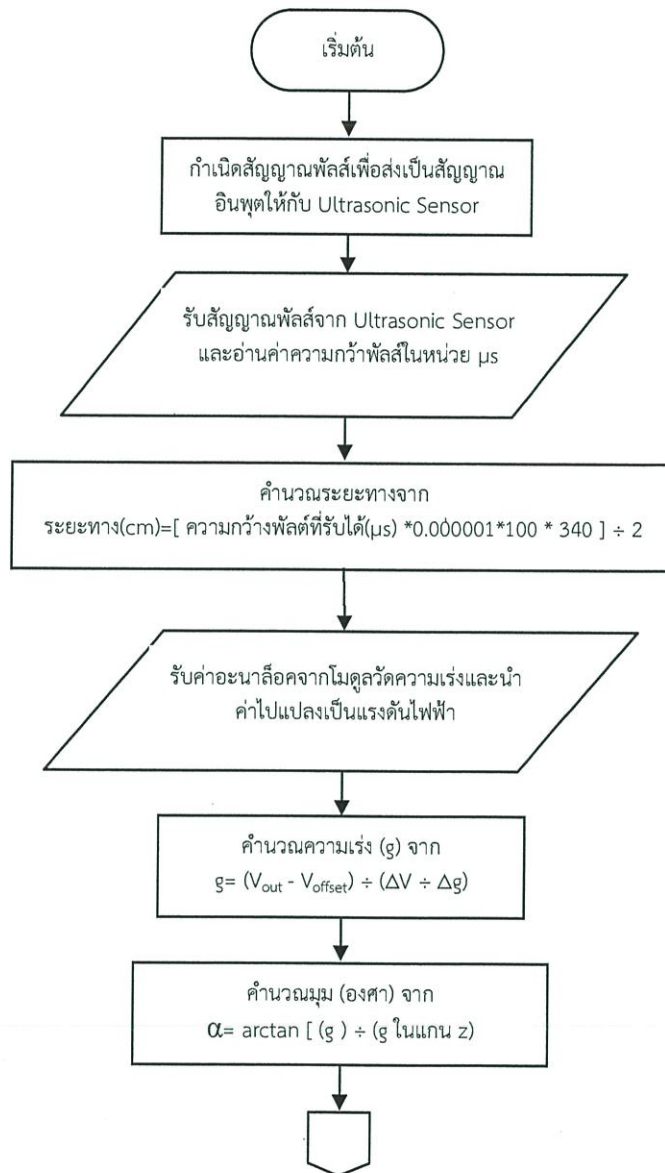
3.1.3.1 เขียนโปรแกรมผ่าน Microcontroller

3.1.3.2 ตั้งค่าผ่านโปรแกรม Putty โดยใช้ dongle ผ่าน PC

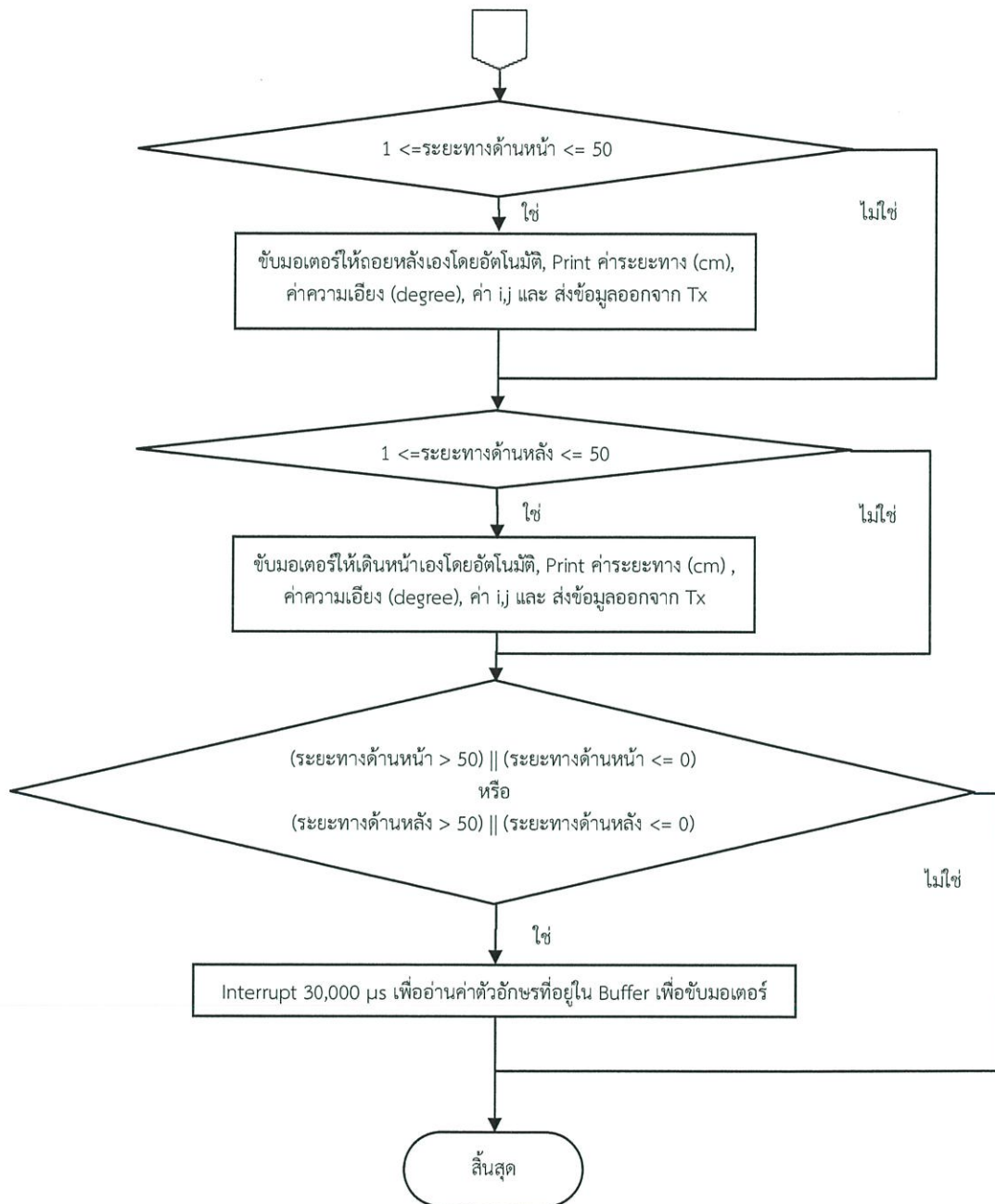
ในที่นี้จะใช้การตั้งค่าผ่านโปรแกรม Putty

- (1) เซ็ต SSID ของเครือข่าย WLAN ให้มีชื่อเป็น AndroidAP และเซตการ Connect เป็นแบบ Auto เมื่อทำการเปิดใช้งานโมดูล Wi-Fi
- (2) เซ็ตรหัสผ่านของตัวเครือข่าย WLAN โดยจะเซตให้เป็น 024294968
- (3) เซ็ต IP ให้เป็นแบบ Static ที่ IP 193.168.43.47 และทำการปิดโหมด DHCP

3.1.4 การออกแบบการควบคุมด้วยไมโครคอนโทรลเลอร์



รูปที่ 3.6 Flowchart แสดงการประมวลผลโดยไมโครคอนโทรลเลอร์



รูปที่ 3.6 Flowchart การประมวลผลโดยไมโครคอนโทรลเลอร์ (ต่อ)

ในกรณีการรับข้อมูลตัวอักษรจากโมดูล Wi-Fi ข้อมูลจะถูกส่งผ่านโดยพอร์ตอนุกรม และถูกเก็บไว้ในบัฟเฟอร์ของไมโครคอนโทรลเลอร์ จากนั้นจะใช้ซอฟต์แวร์ในการอ่านค่าข้อมูลที่ถูกเก็บและกำหนดเงื่อนไข ถ้าข้อมูลที่ถูกรับตรงกับข้อมูลที่ส่งมาจะถูกประมวลผลและไปกำหนดให้เอาต์พุตที่ได้เป็นสัญญาณดิจิทัลทั้งหมด 6 บิต ออกมาเป็นสภาวะลอจิก High และ Low โดยจะนำเอาบิตทั้งหมดไปเป็นบิตอินพุตให้กับวงจรในส่วนควบคุมมอเตอร์

ในกรณีการส่งข้อมูลไมโครคอนโทรลเลอร์จะถูกโปรแกรมให้อ่านค่าความกว้างพัลส์ที่รับได้ในหน่วยไมโครวินาที จากนั้นจะนำค่ามาคำนวณหาระยะทาง ในขณะที่เดียวกันไมโครคอนโทรลเลอร์จะรับสัญญาณอนาล็อกจากโมดูลวัดความแรงและทำการแปลงเป็นค่าตัวเลขดิจิทัลผ่านการทำ Analog to Digital Converter ค่าตัวเลขดิจิทัลที่ได้จะถูกนำมาแปลงให้เป็นแรงดันทางไฟฟ้าเพื่อนำไปคำนวณหาค่าความแรง โดยค่าแรงดันทางไฟฟ้าจะเปลี่ยนแปลงตามค่าความแรงเป็นแบบเชิงเส้น กระบวนการสุดท้ายจะนำความแรงที่ได้ไปคำนวณค่าความเอียงในหน่วยองศา ข้อมูลที่ได้สุดท้ายคือระยะทางที่ตรวจจับสิ่งกีดขวางและค่าความเอียงจะถูกส่งออกจากพอร์ตอนุกรมผ่านขา T_x ของไมโครคอนโทรลเลอร์เพื่อส่งไปยังขา R_x ของโมดูล Wi-Fi ต่อไป

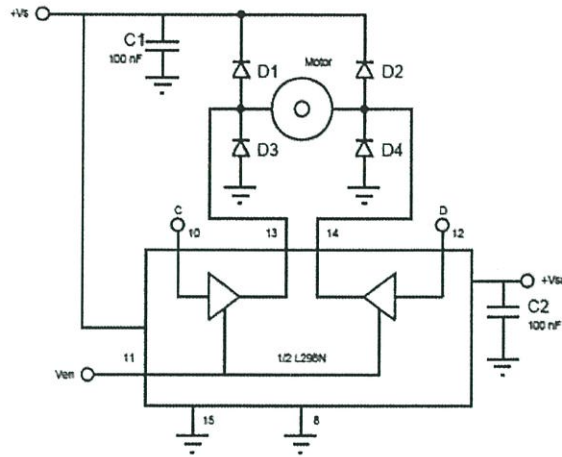
3.1.5 การออกแบบการควบคุมมอเตอร์

ตารางที่ 3.1 แสดงบิต Input 3 Bit ในการควบคุมมอเตอร์ 1 ตัว

V_{en}	สถานะของบิตอินพุต	ฟังก์ชัน
H	C=H ; D=L	Forward
	C=L; D=H	Reverse
	C=D	Fast Motor Stop
L	C=X ; D=X	Free Running Motor Stop

L= Low; H= High; X=Don't care

การออกแบบในส่วนวงจรขับเคลื่อนมอเตอร์โดยการใช้บิตในการควบคุม 3 bit ต่อมอเตอร์ 1 ตัว ได้แก่ บิต V_{en} , C และ D มอเตอร์จะถูก Forward เมื่อกำหนด V_{en} =High, C=High และ D=Low ในทางกลับกันมอเตอร์จะถูก Reverse เมื่อกำหนด V_{en} =High, C=Low และ D=High และมอเตอร์จะถูกหยุดอย่างอิสระ เมื่อกำหนด V_{en} =Low, C=D=Don't care ขณะที่มอเตอร์ถูก Forward ศักย์ไฟฟ้าที่ขา 13 จะมีค่ามากกว่าที่ขา 14 ในทางกลับกัน ถ้ามอเตอร์ถูก Reverse ศักย์ไฟฟ้าที่ขา 14 จะมากกว่าที่ขา 13 และศักย์ไฟฟาระหว่างขา 13 และขา 14 จะเท่ากัน เมื่อมอเตอร์หยุดหมุน



รูปที่ 3.7 วงจรขับมอเตอร์โดยใช้บิตอินพุต 3 Bit

3.1.6 การออกแบบวัดระยะทางโดย Ultrasonic Sensor

3.1.6.1 สร้างสัญญาณ PWM ที่มีความกว้างพัลส์ $\geq 10 \mu s$ จากไมโครคอนโทรลเลอร์เพื่อกระตุ้นการทำงานของ Ultrasonic Sensor

3.1.6.2 ไมโครคอนโทรลเลอร์จะอ่านค่าสัญญาณพัลส์ที่ได้รับได้จาก Ultrasonic Sensor ในหน่วยไมโครวินาที โดยความกว้างสัญญาณพัลส์ที่ได้รับได้คือระยะเวลาทั้งหมดที่คลื่นเดินทางไปที่ตกกระทบวัตถุและเดินทางกลับมายัง Ultrasonic Sensor

3.1.6.3 คำนวณค่าระยะทางจากสูตร ระยะทาง (m) = [ความเร็วเสียงในอากาศ (m/s) \times ระยะเวลาที่คลื่นเดินทางไป-กลับ(s) \div 2]

โดยค่าความเร็วเสียงในอากาศ (m/s) = $331 + (0.6 \times \text{อุณหภูมิหน่วยองศาเซลเซียส})$

3.1.7 การออกแบบวัดความเร่งโดย Accelerometer

3.1.7.1 อ่านค่าอนาล็อกและแปลงมาเป็นค่าตัวเลขข้อมูลขนาด 10 บิต 0-1024 ค่า ผ่านการทำ Analog to Digital Converter

3.1.7.2 คำนวณค่าแรงดันเอาต์พุตโดยการแปลงค่าตัวเลขให้อยู่ในรูปแบบแรงดันทางไฟฟ้า คำนวณจาก แรงดันเอาต์พุต (โวลต์) = [(แรงดันอ้างอิง (5 โวลต์) - แรงดันออฟเซต (โวลต์) \times ค่าตัวเลข (ข้อมูล)] \div 1024 โดยที่แรงดันออฟเซตเป็นแรงดันในสถานะที่

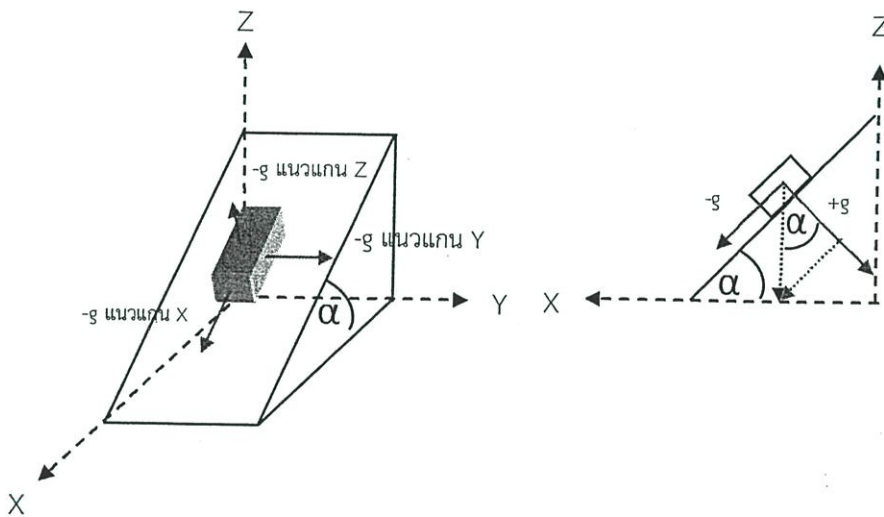
Accelerometer อยู่ในแนวระนาบพื้นดิน ดังนั้นค่าแรงดันเอาต์พุตที่ได้ คือ ค่าอัตราการเปลี่ยนแปลงแรงดัน

3.1.7.3 จากคุณสมบัติ Accelerometer อัตราการเปลี่ยนแปลงแรงดัน (ΔV) ต่อ อัตราการเปลี่ยนแปลงความเร่ง (Δg) เป็นแบบเชิงเส้นและมีค่า 330 mV/g

3.1.7.4 คำนวณหาค่า g จาก ($g = \text{แรงดันอินพุต (โวลต์)} \div 0.33$)

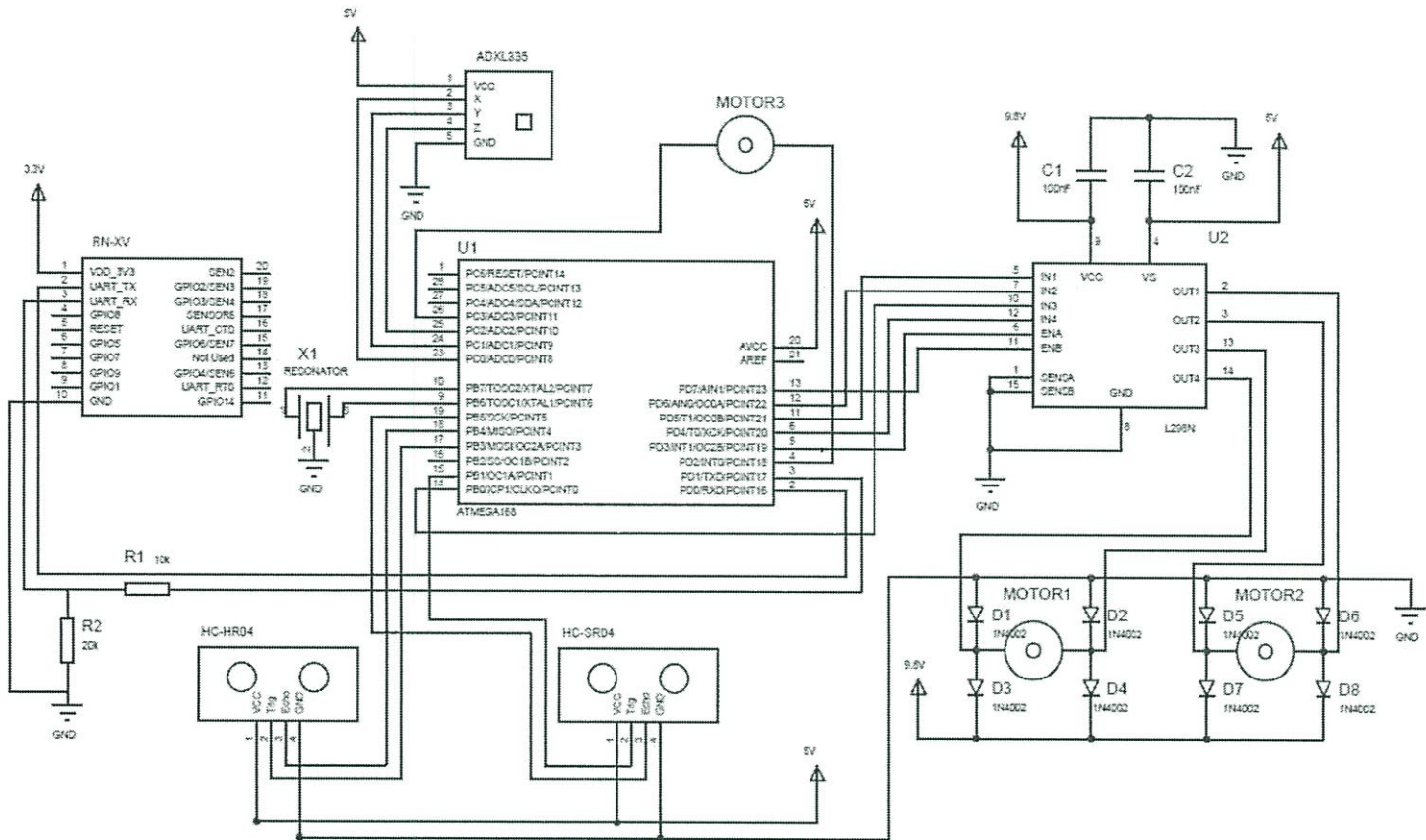
3.1.7.5 คำนวณหาค่ามุมเอียงจาก α (เรเดียน) = $\arctan (g \div g \text{ ในแนวแกน Z})$ โดยค่ามุมที่ได้จากการคำนวณจากไมโครคอนโทรลเลอร์จะอยู่ในหน่วยเรเดียน

3.1.7.6 คำนวณมุมในหน่วยองศา จาก α (องศา) = α (เรเดียน) $\times 180 \div \pi$



รูปที่ 3.8 ทิศทางความเร่งในแนวแกน

รูปที่ 3.9 วงจรรวมภาครับสัญญาณ



3.1.8 การออกแบบวงจรรวมภาครับสัญญาณ

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 เครื่องมือที่ใช้ในการพัฒนา Application และควบคุมด้วยโปรแกรม

- (1) Eclipse
- (2) Software Development Kit (SDK)
- (3) Java Development Kit (JDK)
- (4) Putty

3.2.2 โทรศัพท์มือถือ Smart Phone

ใช้ในการส่งสัญญาณแบบไร้สายไปยังโมดูล Wi-Fi โดยจะต้องเขียนโปรแกรม Application แล้วโหลดลงในโทรศัพท์มือถือ Smart Phone ก่อนใช้งาน

3.2.3 โมดูล Wi-Fi

ใช้ในการรับสัญญาณแบบไร้สายจากโทรศัพท์มือถือ Smart Phone แล้วแปลงสัญญาณออกมาเป็นข้อมูลตัวอักษรทางขา T_x แบบอนุกรม วิธีใช้ ต้องมีการเซตค่า IP ก่อนการใช้งาน เพื่อให้สามารถติดต่อสื่อสารระหว่างโทรศัพท์มือถือ Smart Phone กับโมดูล Wi-Fi

3.2.4 ไมโครคอนโทรลเลอร์

ใช้ในการรับค่าข้อมูลตัวอักษรจากโมดูล Wi-Fi ผ่านทางพอร์ตอนุกรม แล้วทำการประมวลผลออกมาเป็นสัญญาณบิตดิจิตอล

3.2.5 IC L298N

ใช้ในการควบคุมมอเตอร์แทนการใช้วงจรรีเลย์ เนื่องจากเป็น IC สำเร็จรูป ทำให้การใช้งานง่ายขึ้น โดยทำการป้อนบิตอินพุตจำนวน 3 bit ต่อการควบคุมมอเตอร์ 1 ตัว

3.2.6 มอเตอร์

ใช้ในการหมุนเฟืองและขับเคลื่อนล้อรถให้เดินหน้า ถอยหลัง และหยุด โดยมอเตอร์ที่นำมาประยุกต์ใช้งานนี้ จะ Forward และ Reverse เมื่อทำการป้อนแรงดันไฟฟ้าสลับชั่ววินาที โดยใช้มอเตอร์ DC 9 โวลต์

3.2.7 Ultrasonic Sensor

ใช้คำนวณค่าระยะทางที่รถเข้าใกล้วัตถุ โดยคำนวณจากความกว้างของสัญญาณพัลส์ที่ได้รับได้เป็นระยะเวลาทั้งหมดที่คลื่นส่งสัญญาณไป-กลับ แสดงได้ดังนี้

$$\text{ระยะทาง} = (\text{ความเร็วคลื่นเสียงในสุญญากาศ} \times \text{ระยะเวลาทั้งหมดที่คลื่นส่งสัญญาณไป-กลับ}) \div 2$$

3.2.8 โมดูลวัดความเร่ง

ใช้คำนวณหาค่าความเร่งในหน่วย g โดยรับสัญญาณอนาล็อกและแปลงมาเป็นสัญญาณดิจิตอล จากนั้นคำนวณออกมาเป็นโวลต์ โดยอัตราการเปลี่ยนแปลงความเร่งต่ออัตราการเปลี่ยนแปลงโวลต์เป็นแบบเชิงเส้น

3.2.9 Router

ใช้เป็นตัวกลางในการรับ-ส่งข้อมูลระหว่างโทรศัพท์มือถือ Smart Phone กับโมดูล Wi-Fi โดยเป็นตัวกระจายสัญญาณให้สามารถกระจายสัญญาณให้ไกลขึ้น โดยทั้งโทรศัพท์มือถือ Smart Phone และโมดูล Wi-Fi ต้อง Access เข้า Router ก่อนการรับ-ส่งข้อมูล

3.3 การจัดเก็บผลการทดลอง

3.3.1 จัดเก็บผลการส่งข้อมูลแล้วแสดงผลที่หน้าจอโทรศัพท์มือถือ Smart Phone ภาครัฐ

จัดเก็บผลการส่ง-รับข้อมูลตัวอักษรระหว่างโทรศัพท์มือถือ Smart Phone สองเครื่อง ผ่านการเชื่อมต่อ IP และ PORT ให้ตรงกัน เพื่อทดสอบว่าข้อมูลตัวอักษรที่ภาครัฐตรงกันกับข้อมูลที่ถูส่งมาจากภาคส่ง ตามที่ออกแบบโปรแกรม Application

3.3.2 จัดเก็บผลการทดลองจากการตั้งค่าโมดูล Wi-Fi ผ่านโปรแกรม Putty

จัดเก็บผลที่ได้และวิธีการในการเชื่อมต่อ IP เพื่อทำการเชื่อมต่อการรับ-ส่งข้อมูลระหว่างโทรศัพท์มือถือ Smart Phone กับโมดูล Wi-Fi

3.3.3 จัดเก็บผลการรับข้อมูลของโมดูล Wi-Fi แสดงบน HyperTerminal

จัดเก็บผลข้อมูลตัวอักษรที่ได้ผ่านการเชื่อมต่อระหว่างโทรศัพท์มือถือ Smart Phone กับโมดูล Wi-Fi โดยผลที่ได้จะถูกแสดงบนโปรแกรม HyperTerminal

3.3.4 จัดเก็บผลสัญญาณที่ขา T_x ของโมดูล Wi-Fi

จัดเก็บระดับสัญญาณที่ขา T_x ของโมดูล Wi-Fi ซึ่งเป็นการเชื่อมต่อแบบอนุกรม สัญญาณที่ได้จะมีลอจิก 10 bit จากนั้นจะทำการถอดรหัส ASCII ระดับสัญญาณ 8 bit ออกมาเป็น ข้อมูลตัวอักษรที่รับได้

3.3.5 จัดเก็บสัญญาณอินพุตและเอาต์พุตของ IC L298N

จัดเก็บสัญญาณอินพุตและเอาต์พุต ในขณะที่มีการส่งข้อมูลจากโทรศัพท์มือถือ Smart Phone เพื่อควบคุมมอเตอร์ให้ทำงาน

3.3.6 จัดเก็บผลการควบคุมมอเตอร์โดยใช้ Application บนโทรศัพท์มือถือ Smart Phone

แสดงการทดลองเมื่อกดปุ่มบนหน้าจอโทรศัพท์มือถือจะทำให้มอเตอร์หมุน โดย โทรศัพท์มือถือจะเชื่อมต่อกับ Wi-Fi Module ข้อมูลจะถูกส่งไปยังไมโครคอนโทรลเลอร์เพื่ออ่านค่า จากนั้นจะเข้าสู่วงจรขับเคลื่อนมอเตอร์เพื่อทำให้มอเตอร์หมุน

3.3.7 จัดเก็บผลการทดลองส่งข้อมูลผ่านโปรแกรม Putty

ทดลองการส่งข้อมูลโดยเริ่มจากเครื่องคอมพิวเตอร์ทั้งสองเครื่อง Access เข้า Router และโมดูล Wi-Fi แล้วเซตค่า IP Router จากนั้นจะทำการส่งข้อมูลระหว่างคอมพิวเตอร์ สองเครื่องจากแป้นพิมพ์คอมพิวเตอร์เครื่องที่ 1 ที่เชื่อมกับโมดูล Wi-Fi ทางพอร์ตอนุกรม ตรวจสอบการแสดงผลที่รับได้ของคอมพิวเตอร์เครื่องที่ 2 ว่ามีข้อมูลเข้ามาหรือไม่

3.3.8 จัดเก็บผลการทดลองวัดสัญญาณอินพุต/เอาต์พุตของ Ultrasonic Sensor

วัดสัญญาณอินพุตและเอาต์พุตจาก Ultrasonic Sensor เพื่อแสดงให้เห็นว่าเมื่อ ระยะทางที่เซนเซอร์ตรวจจับวัตถุได้เปลี่ยนไป ความกว้างสัญญาณพัลส์จะเปลี่ยนไปอย่างไร

3.3.9 จัดเก็บผลการทดลองวัดสัญญาณเอาต์พุตของ Ultrasonic Sensor

วัดสัญญาณเอาต์พุตจาก Ultrasonic Sensor อ่านค่าความกว้างของสัญญาณที่รับได้ และคำนวณค่าระยะทางโดยใช้ไมโครคอนโทรลเลอร์

3.3.10 จัดเก็บผลการทดลองวัดระยะทางเมื่อเปรียบเทียบระหว่างการวัดโดยไมโครคอนโทรลเลอร์กับวัดโดยตลับเมตร

เปรียบเทียบกับค่าระยะทางที่วัดได้จริงและค่าระยะทางที่วัดได้จากไมโครคอนโทรลเลอร์ แล้วนำมาหาค่าเฉลี่ย หาเปอร์เซ็นต์ความผิดพลาด พร้อมทั้งตรวจสอบระยะทางไกลสุดที่ Ultrasonic Sensor วัดได้แม่นยำที่สุด

3.3.11 จัดเก็บผลการทดลองเป็นกราฟความสัมพันธ์ระหว่างค่าความเร่งของ Accelerometer กับแรงดันทางไฟฟ้า

วัดความเร่งของ Accelerometer โดยเอียงตัว Accelerometer ให้มีค่ามุมเอียงตั้งแต่ 0 องศา ถึง 90 องศา โดยเอียงทุกๆ 10 องศา และเก็บค่าแรงดันทางไฟฟ้าและค่าความเร่ง g จากนั้นพล็อตกราฟและหาค่าความชันของกราฟทั้งระนาบ X และ Y

3.3.12 จัดเก็บผลการทดลองวัดค่ามุมเอียงที่วัดได้จริงโดยใช้ Protractor เปรียบเทียบกับค่าความเอียงที่วัดได้จาก Microcontroller แสดงผลทาง Hyperterminal

วัดค่ามุมเอียงโดยใช้ Protractor กับค่าความเอียงที่วัดได้โดย Microcontroller เพื่อทดสอบว่าอุปกรณ์เซนเซอร์มีความเที่ยงตรงมากน้อยเพียงใด

3.3.13 จัดเก็บผลการทดลองวัดค่าความเอียงที่วัดได้จริงเปรียบเทียบกับค่าความเอียงที่วัดได้จากไมโครคอนโทรลเลอร์

จัดฐาน Accelerometer ให้อยู่ในระนาบพื้นดินจากนั้นเอียงตัว Accelerometer ตั้งแต่ 0 องศา ถึง 90 องศา จัดเก็บค่ามุมเอียงที่วัดได้ทุกๆ 10 องศา ทั้งระนาบ X และ Y เก็บค่ามุมเอียงที่วัดได้ที่แสดงผลโดยไมโครคอนโทรลเลอร์ แล้วนำมาหาค่าเฉลี่ยและเปอร์เซ็นต์ความผิดพลาด

3.3.14 จัดเก็บระยะการรับ-ส่งสัญญาณระหว่างโมดูล Wi-Fi กับโทรศัพท์มือถือ Smart Phone และระยะรับ-ส่งภาพระหว่างโทรศัพท์มือถือ Smart Phone ทั้งสองเครื่องผ่าน Router

เพื่อทดสอบระยะทางที่รถสามารถเข้าถึงพื้นที่ใดๆ ได้ไกลสูงสุดทั้งแบบ Line-of-sight และ Non-line-of-sight

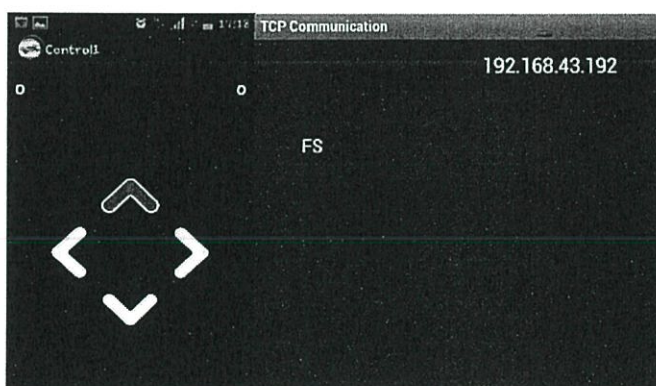
บทที่ 4

ผลการทดลอง

4.1 การทดสอบการส่งข้อมูลระหว่างโทรศัพท์มือถือ Smart Phone สองเครื่องผ่าน Application

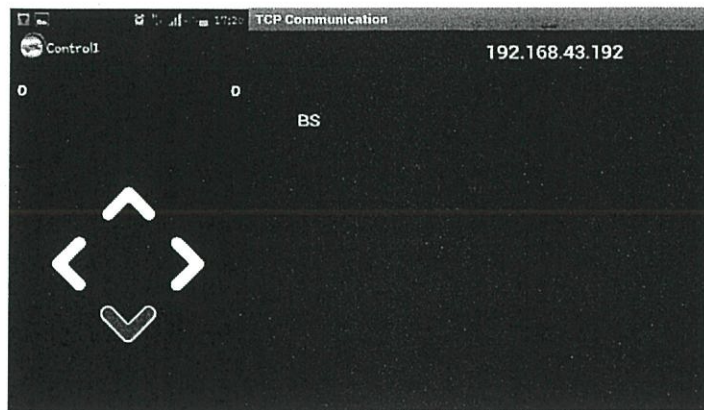
ทำการทดสอบโดยให้ฝั่งหนึ่งเป็นฝ่ายส่งข้อมูล อีกฝั่งหนึ่งเป็นฝ่ายรับข้อมูล ซึ่งเป็นการทดสอบว่า Application ที่สร้างขึ้น โดยได้ทำการส่งข้อมูลออกไปและทำการเก็บภาพจากโทรศัพท์มือถือ Smart Phone ทั้งสองเครื่องในกรณีต่างๆ ดังนี้ เมื่อกดปุ่มเดินหน้า, ถอยหลัง, หมุนซ้าย และหมุนขวา ตามลำดับ

4.1.1 เมื่อกดปุ่มเดินหน้าจนกระทั่งปล่อยปุ่ม ฝั่งส่งจะทำการส่งค่า 'F' และ 'S' ออกมาตามลำดับ



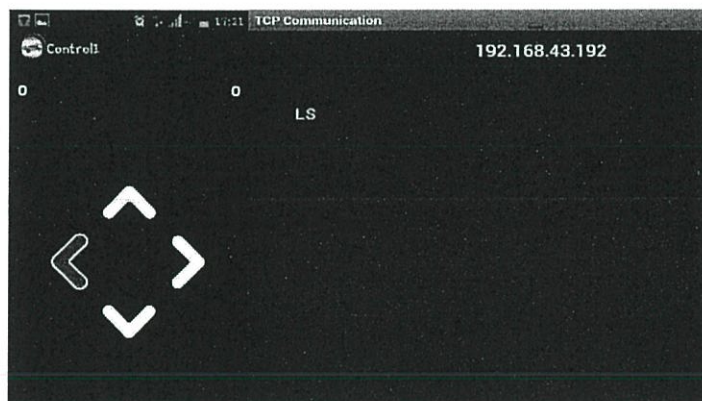
รูปที่ 4.1 ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเดินหน้า

4.1.2 เมื่อกดปุ่มถอยหลังจนกระทั่งปล่อยปุ่ม ฝั่งส่งจะทำการส่งค่า 'B' และ 'S' ออกมาตามลำดับ



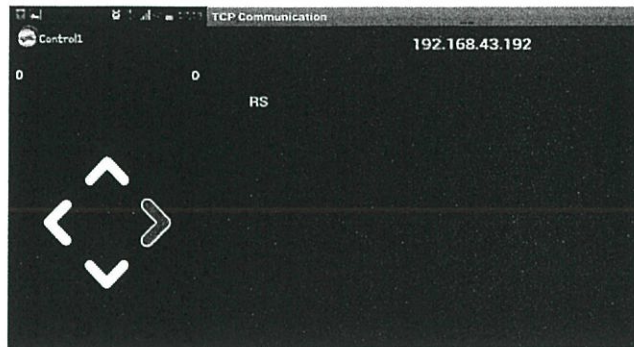
รูปที่ 4.2 ข้อมูลที่ส่งออกมาเมื่อกดปุ่มถอยหลัง

4.1.3 เมื่อกดปุ่มหมุนซ้ายจนกระทั่งปล่อยปุ่ม ฝั่งส่งจะทำการส่งค่า 'L' และ 'S' ออกมาตามลำดับ



รูปที่ 4.3 ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเลี้ยวซ้าย

4.1.4 เมื่อกดปุ่มหมุนขวาจนกระทั่งปล่อยปุ่ม ฝั่งส่งจะทำการส่งค่า 'R' และ 'S' ออกมาตามลำดับ



รูปที่ 4.4 ข้อมูลที่ส่งออกมาเมื่อกดปุ่มเลี้ยวขวา

4.2 การตั้งค่าโมดูล Wi-Fi ผ่านโปรแกรม Putty

เมื่อต้องการให้โมดูลรับค่าจากโทรศัพท์มือถือ Smart Phone ต้องทำการตั้งค่าให้โมดูลมีการเชื่อมต่อกับโทรศัพท์มือถือ Smart Phone โดยสามารถทำได้ 2 วิธี คือ

4.2.1 เชียนโปรแกรมผ่าน Microcontroller

4.2.2 ตั้งค่าผ่านโปรแกรม Putty โดยใช้ Dongle ผ่าน PC

ในที่นี้จะใช้การตั้งค่าผ่านโปรแกรม Putty

4.2.2.1 เช็ต SSID ของเครือข่าย WLAN ให้มีชื่อเป็น AndroidAP และเช็ตการ Connect เป็นแบบ Auto เมื่อทำการเปิดใช้งานโมดูล

```
<2.32> set wlan join 1
ACK
<2.32> set wlan ssid AndroidAP
ACK
```

รูปที่ 4.5 การเช็ตโหมดการเชื่อมต่อและชื่อของ SSID

4.2.2.2 การเช็ตรหัสผ่านของเครือข่าย WLAN โดยจะเช็ตให้เป็น 024294968

```
<2.32> set wlan phrase 024294968
ACK
```

รูปที่ 4.6 การเช็ตรหัสผ่านของเครือข่าย WLAN ที่ใช้กับโมดูล

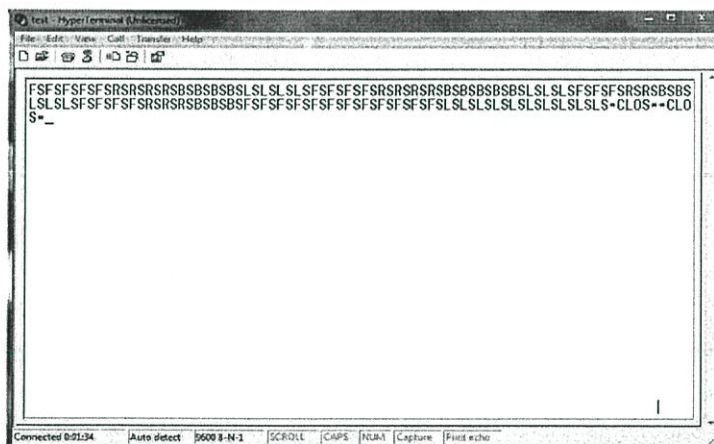
4.2.2.3 การเซต IP ให้เป็นแบบ Static ที่ IP 193.168.43.47 และทำการปิดโหมด DHCP

```
<2.32> set ip address 192.168.43.47
AOK
<2.32> set ip dhcp 0
AOK
```

รูปที่ 4.7 การเซตให้เป็น IP แบบ Static

4.3 การทดสอบการส่งข้อมูลโดยดูผลผ่านโปรแกรม Hyperterminal

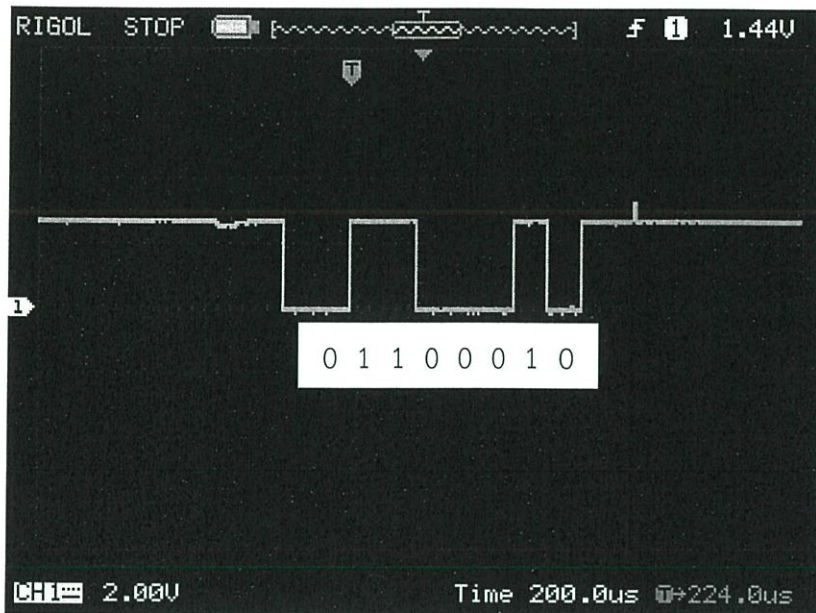
ทำการทดสอบเพื่อดูการส่งข้อมูลว่าสามารถส่งจากตัวโมดูลมายัง Serial Port ของ Microcontroller เพื่อที่จะทำการรับข้อมูลที่ส่งมาจากฝั่งส่ง แล้วนำไปใช้ควบคุมการหมุนของมอเตอร์



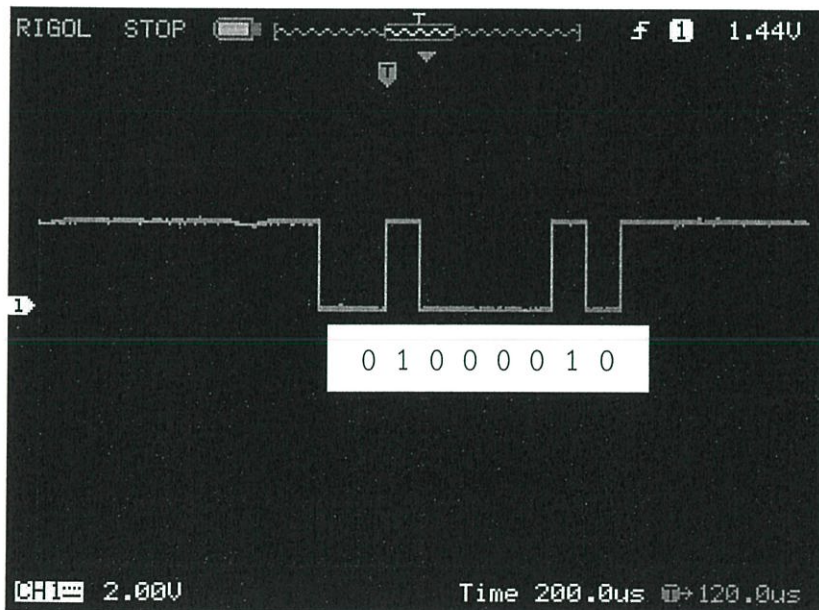
รูปที่ 4.8 การส่งข้อมูลโดยดูผลผ่านโปรแกรม Hyperterminal

4.4 ผลการวัดสัญญาณที่ขา T_x ของโมดูล Wi-Fi

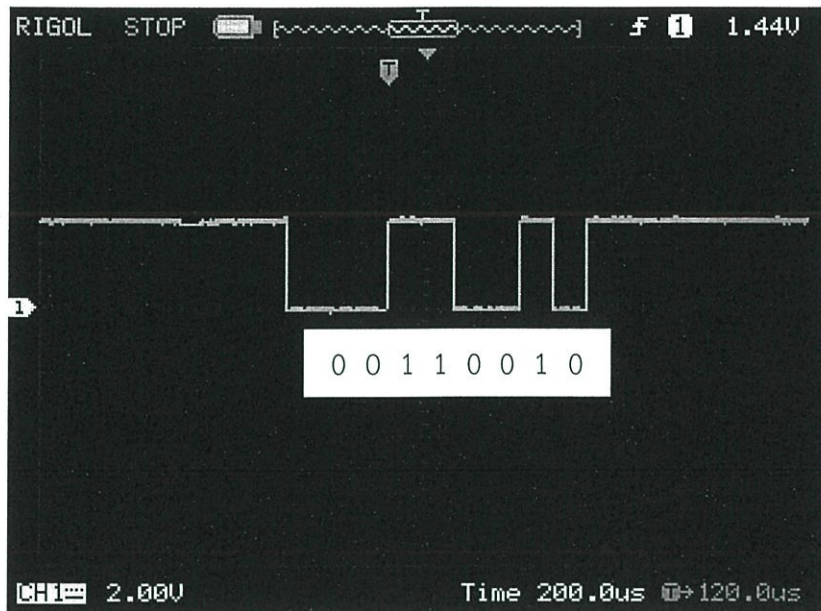
ทำการวัดสัญญาณที่ขา T_x ของตัวโมดูล Wi-Fi RN-XV เพื่อดูว่าสัญญาณที่ส่งมาจากโทรศัพท์มือถือ Smart Phone นั้นมีความถูกต้องตามมาตรฐานของ ASCII ซึ่งสามารถทำได้โดยวัดที่ขา 2 ของโมดูล ซึ่งมีทั้งสิ้น 5 คำสั่ง คือ เดินหน้า ถอยหลัง หมุนซ้าย หมุนขวา และหยุด



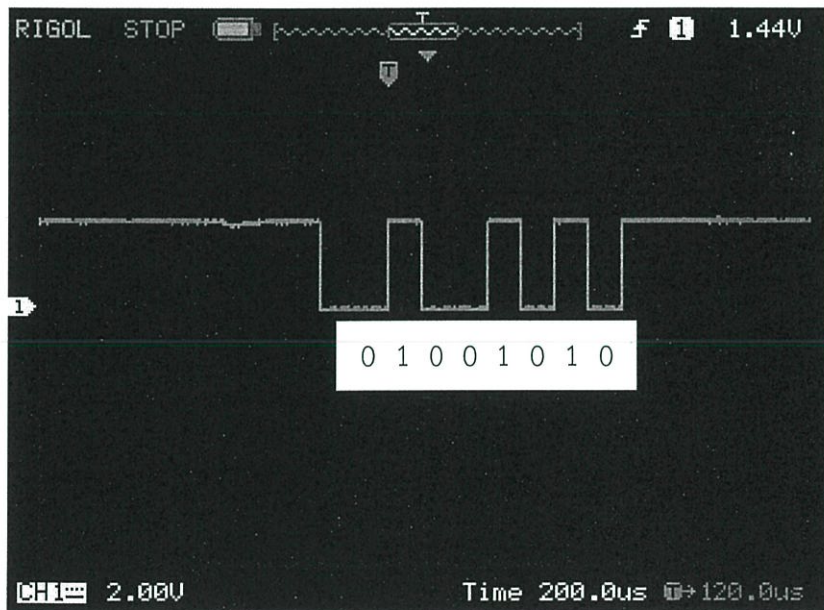
รูปที่ 4.9 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถเดินหน้า (F)



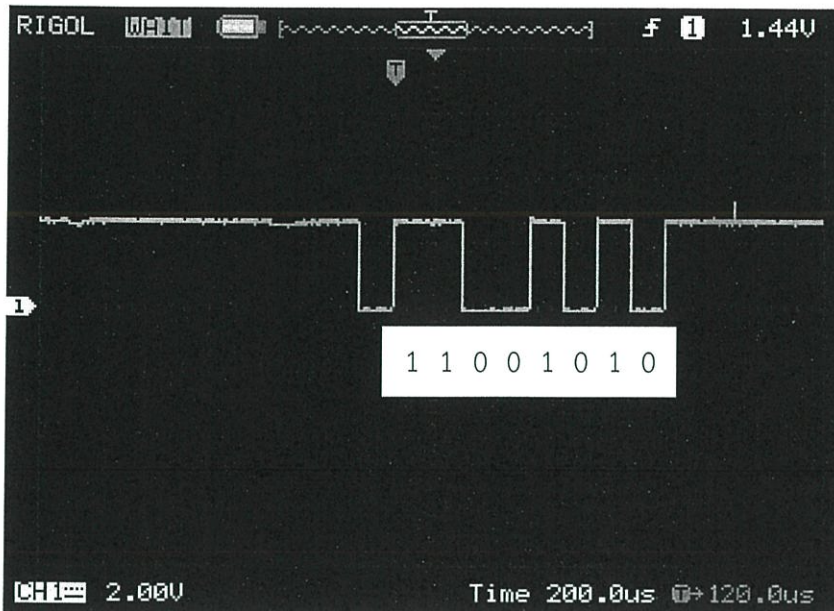
รูปที่ 4.10 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถถอยหลัง (B)



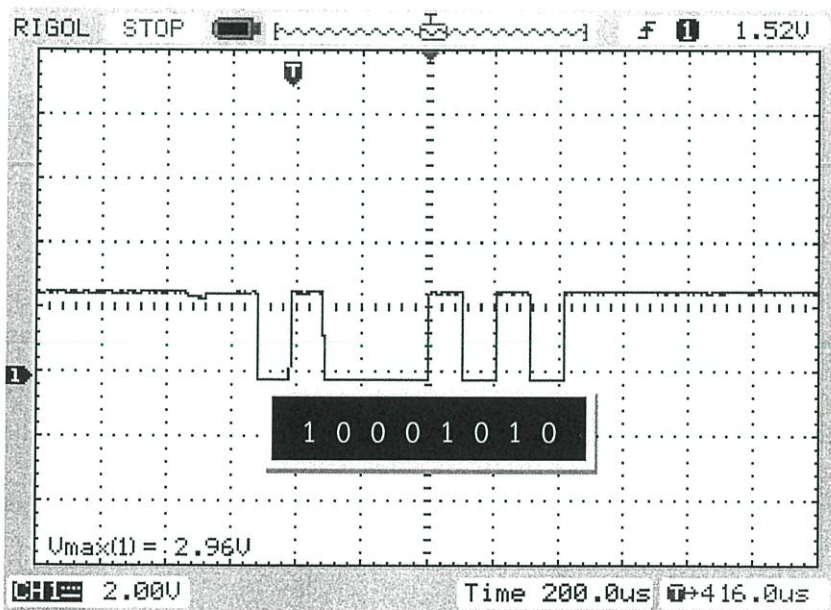
รูปที่ 4.11 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหมุนซ้าย (L)



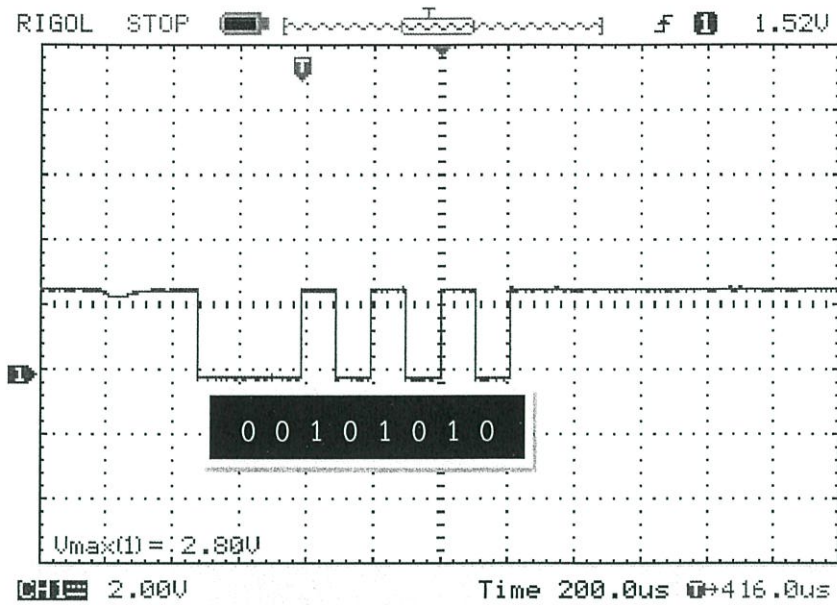
รูปที่ 4.12 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหมุนขวา (R)



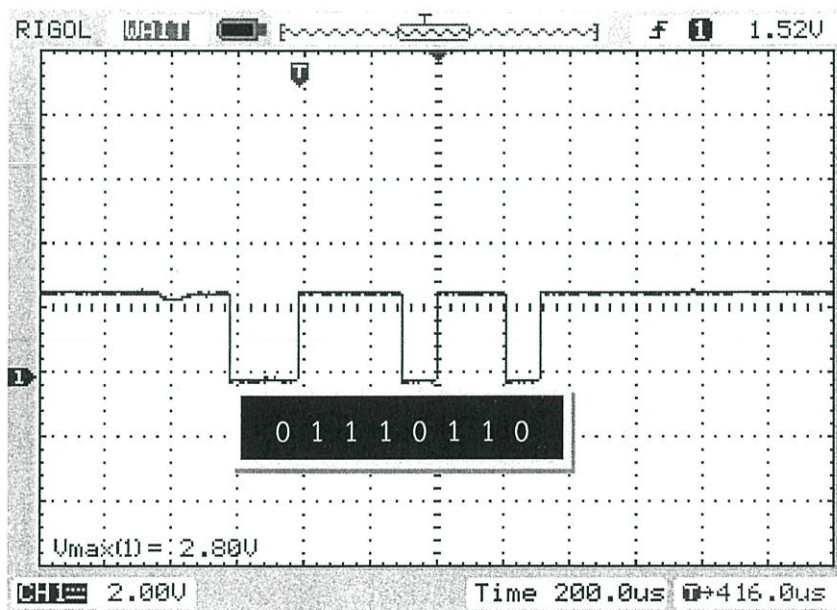
รูปที่ 4.13 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้รถหยุด (S)



รูปที่ 4.14 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล้องหมุนซ้าย (Q)



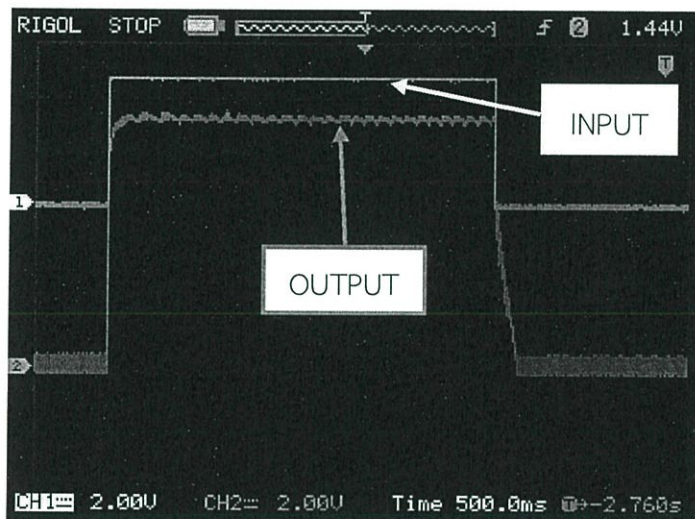
รูปที่ 4.15 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล้องหมุนขวา (T)



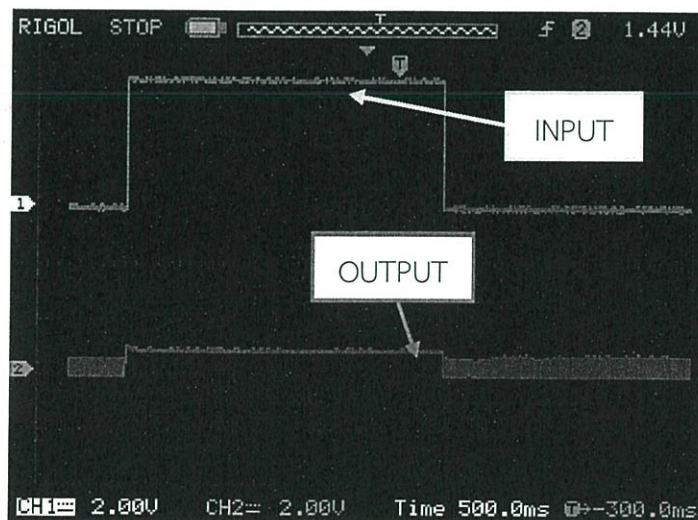
รูปที่ 4.16 สัญญาณข้อมูลจากโทรศัพท์มือถือ Smart Phone เมื่อสั่งให้กล้องหยุดหมุน (n)

4.5 ผลการจัดเก็บสัญญาณ Input และ Output ของ IC 298N

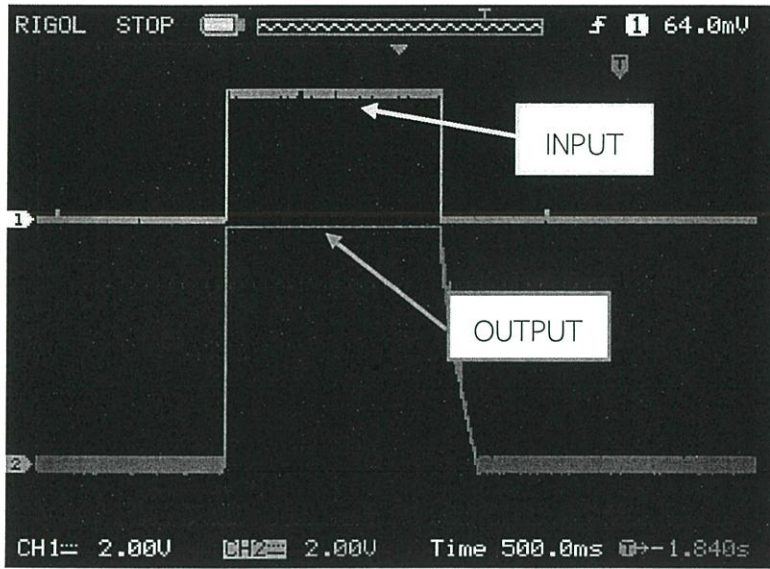
ทำการจัดเก็บสัญญาณที่ขา Input และ Output เพื่อดูว่าสัญญาณที่ออกมาเป็นไปตามสเปคของ IC 298N โดยในที่นี้จะยกตัวอย่างการกดเดบีนหน้า แล้วทำการวัดสัญญาณ Input ทั้งหมด 3 bit ซึ่งมีหน้าที่ในการควบคุมมอเตอร์ และวัดสัญญาณ Output ทั้งหมด 2 จุด ที่ขา 13 และขา 14 ของ IC ซึ่งจะถูกต่อเข้ากับขั้วมอเตอร์



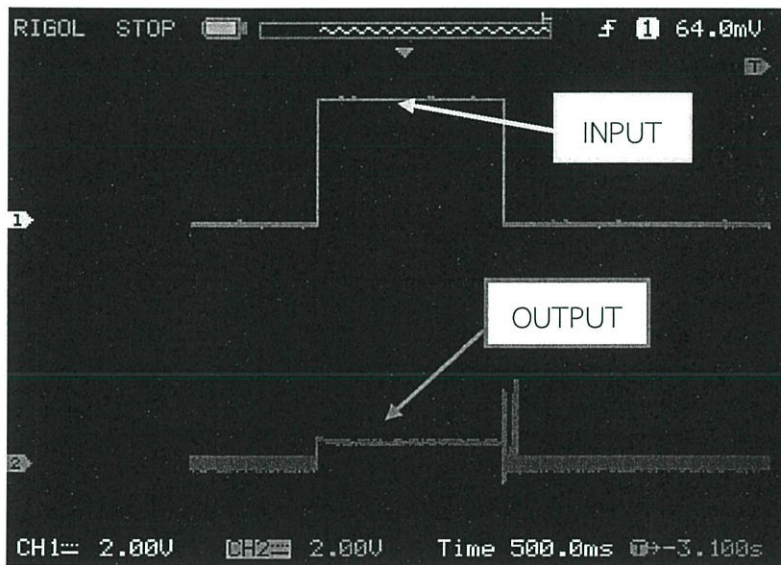
รูปที่ 4.17 สัญญาณ Input (V_{en}) เทียบกับสัญญาณ Output ที่ขา 13



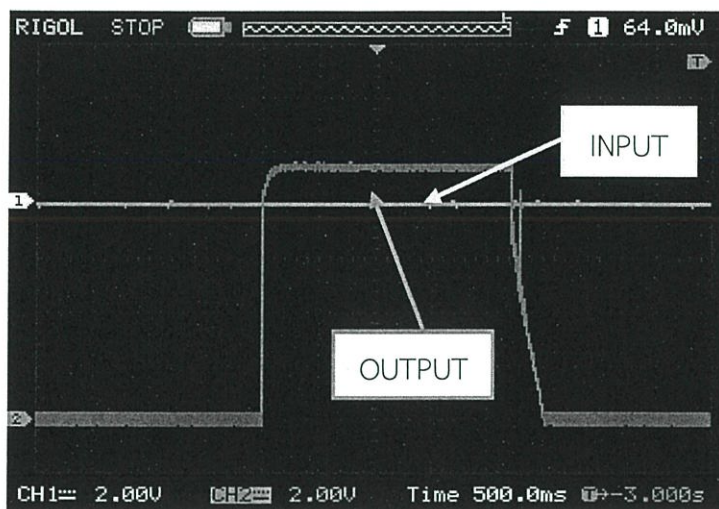
รูปที่ 4.18 สัญญาณ Input (V_{en}) เทียบกับสัญญาณ Output ที่ขา 14



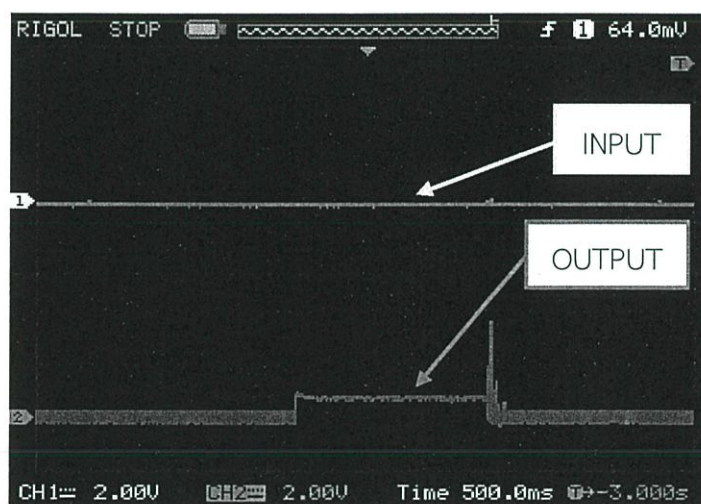
รูปที่ 4.19 สัญญาณ Input (C) เทียบกับสัญญาณ Output ที่ขา 13



รูปที่ 4.20 สัญญาณ Input (C) เทียบกับสัญญาณ Output ที่ขา 14



รูปที่ 4.21 สัญญาณ Input (D) เทียบกับสัญญาณ Output ที่ขา 13

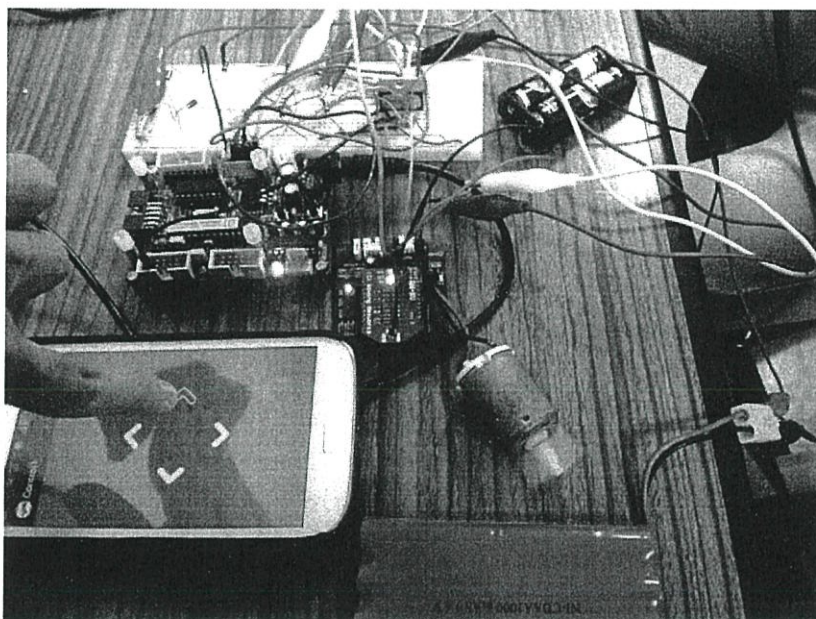


รูปที่ 4.22 สัญญาณ Input (D) เทียบกับสัญญาณ Output ที่ขา 14

จากรูปที่ 4.17 – 4.22 เป็นรูปอธิบายการป้อนสัญญาณ INPUT ทั้งหมด 3 บิต ให้กับ IC L298N โดยใช้ สัญญาณ INPUT (V_{in}) เป็นสถานะ High, สัญญาณ INPUT (C) เป็นสถานะ High และ สัญญาณ INPUT (D) เป็นสถานะ Low สัญญาณ OUTPUT ที่ออกจากขาที่ 13 และขาที่ 14 ของ IC L298N จะมีระดับแรงดันเป็น 9 โวลต์ และ 0 โวลต์ ตามลำดับ เพื่อขับมอเตอร์ให้หมุนไปด้านหน้า

4.6 การควบคุมมอเตอร์โดยใช้ Application บนโทรศัพท์มือถือ Smart Phone

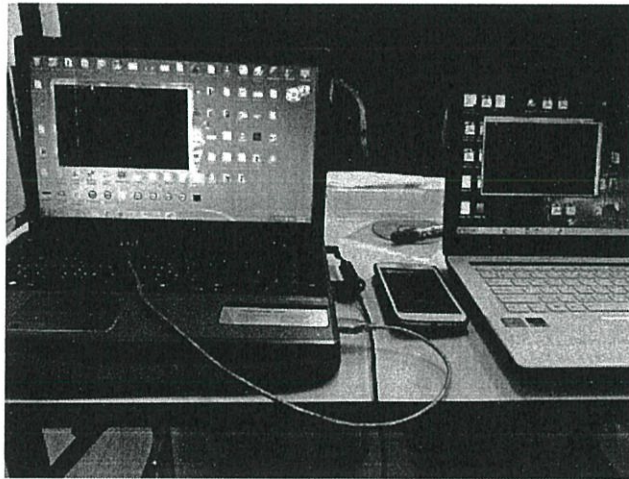
ทำการทดสอบเพื่อดูว่า Application สามารถที่จะควบคุมการหมุนของมอเตอร์ตามทีออกแบบไว้



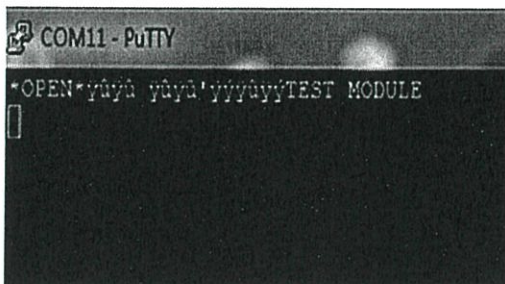
รูปที่ 4.23 การหมุนของมอเตอร์เมื่อสั่งผ่านโทรศัพท์มือถือ Smart Phone

4.7 ผลการทดลองรับ-ส่งข้อมูลผ่านโปรแกรม Putty

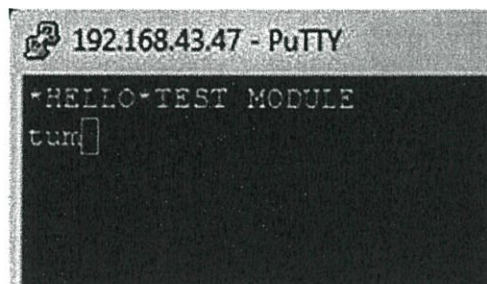
ทดลองการส่งข้อมูลทางคีย์บอร์ดจากเครื่องคอมพิวเตอร์ที่ 1 ไปยังคอมพิวเตอร์เครื่องที่ 2 โดยคอมพิวเตอร์เครื่องที่ 1 จะเชื่อมต่อกับโมดูล Wi-Fi ทางพอร์ตอนุกรม (Serial Port) สัญญาณจากโมดูล Wi-Fi จะผ่าน Router และแสดงผลข้อมูลไปยังหน้าจอคอมพิวเตอร์เครื่องที่ 2 ผ่านโปรแกรม Putty โดยใช้โหมด Telnet โดยก่อนการส่งข้อมูล อุปกรณ์โมดูล Wi-Fi และเครื่องคอมพิวเตอร์ทั้ง 2 เครื่องต้อง Access เข้า Router ก่อน ซึ่งในที่นี้คือโทรศัพท์มือถือ Smart Phone



รูปที่ 4.24 การส่งข้อมูลจากโมดูล Wi-Fi ไปยังคอมพิวเตอร์เครื่องที่ 2



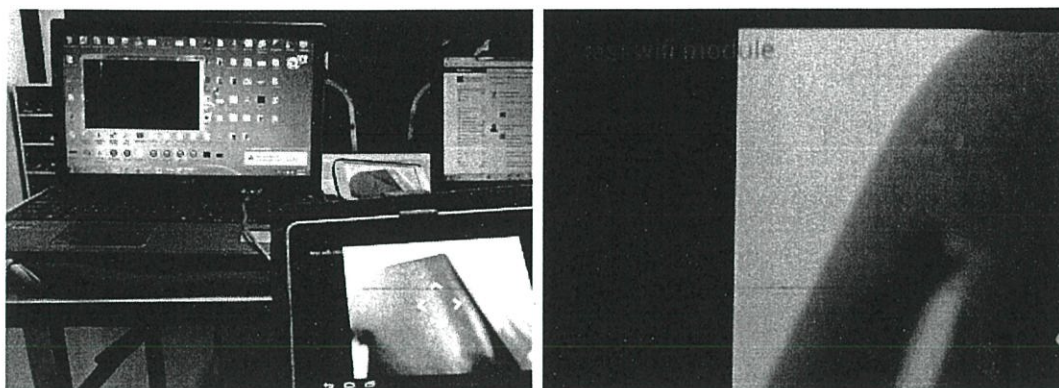
(ก) การรับข้อมูลที่คอมพิวเตอร์เครื่องที่ 1



(ข) การรับข้อมูลที่คอมพิวเตอร์เครื่องที่ 2

รูปที่ 4.25 การรับข้อมูลที่คอมพิวเตอร์เครื่องที่ 1 และเครื่องที่ 2

ทดลองการรับข้อมูลข้อมูลจากคีย์บอร์ดบนคอมพิวเตอร์เครื่องที่1ไปยังหน้าจอ Application บนโทรศัพท์มือถือ Smart Phone โดยคอมพิวเตอร์เครื่องที่1จะเชื่อมต่อกับโมดูล Wi-Fi ทางพอร์ตอนุกรม สัญญาณจากโมดูล Wi-Fi จะผ่าน Router และแสดงผลข้อมูลไปยังหน้าจอ Application จากรูป (ข) เป็นหน้าจอ Application ที่รับข้อมูล “test wifi module” ที่ถูกส่งมาจากคอมพิวเตอร์เครื่องที่ 1 โดยการพิมพ์ตัวอักษรผ่าน Serial Port



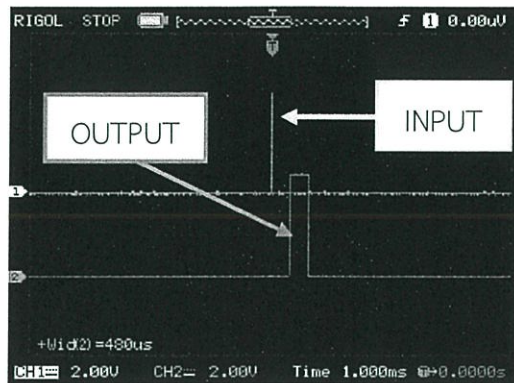
(ก) การส่งข้อมูลบน Application

(ข) การรับข้อมูลบน Application

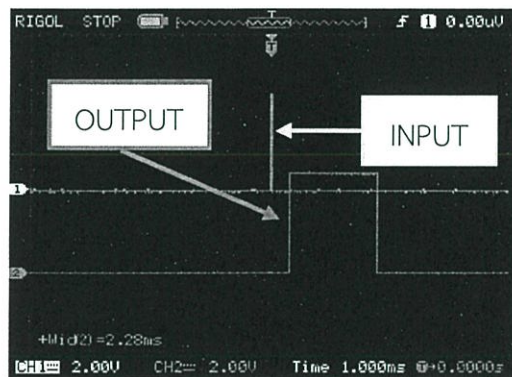
รูปที่ 4.26 การรับ-ส่งข้อมูลไปยัง Application

4.8 ผลการทดลองวัดสัญญาณ Input/Output ของ Ultrasonic Sensor

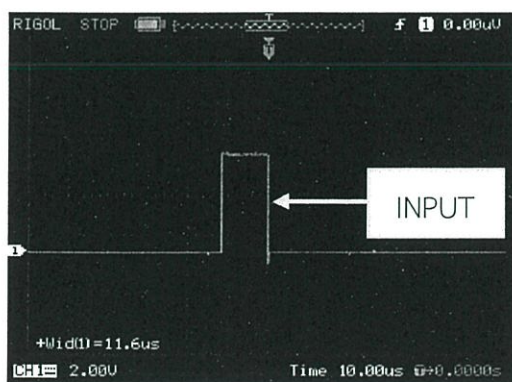
เป็นการป้อนสัญญาณอินพุตให้กับ Ultrasonic Sensor จากนั้นสัญญาณเอาต์พุตจาก Ultrasonic Sensor จะเป็นสัญญาณพัลส์ที่มีความกว้างพัลส์ตามระยะทางที่เปลี่ยนไป เมื่อระยะทางระหว่าง Sensor กับวัตถุมีค่ามากขึ้น สัญญาณพัลส์ที่เอาต์พุตของ Sensor จะมีความกว้างพัลส์เพิ่มมากขึ้น



(ก) สัญญาณอินพุตเทียบกับสัญญาณเอาต์พุตที่ระยะทาง 8 cm



(ข) สัญญาณอินพุตเทียบกับสัญญาณเอาต์พุตที่ระยะทาง 40 cm



(ค) สัญญาณอินพุต PWM ความกว้างพัลส์ 10 μ s

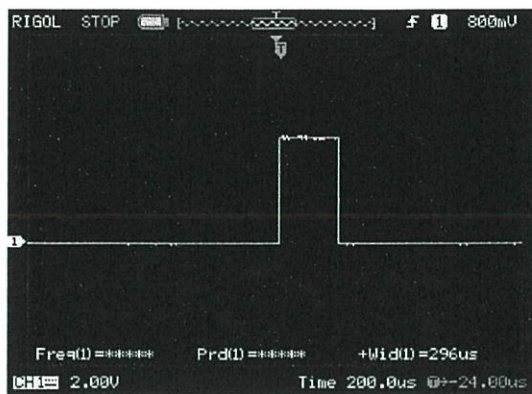
รูปที่ 4.27 สัญญาณจาก Oscilloscope ที่ขาอินพุตและเอาต์พุตของ Ultrasonic Sensor

4.9 ผลการทดลองวัดสัญญาณ Output ที่ระยะทางต่างๆ ของ Ultrasonic Sensor

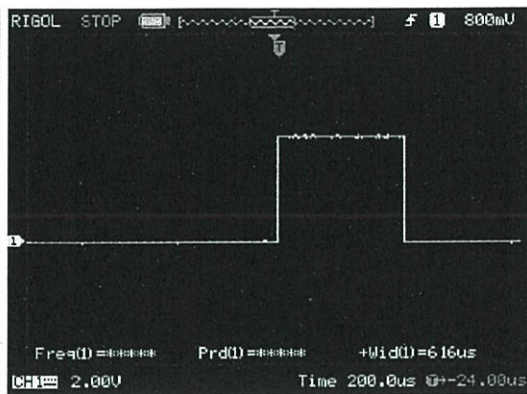
ตารางที่ 4.1 เปรียบเทียบระหว่างระยะทางจากการวัดและคาบเวลาของ Pulse

ระยะทาง (cm)	2	3	4	5	10	20	30	40	50
คาบเวลา ของ Pulse (ms)	0.11	0.17	0.23	0.30	0.62	1.16	1.74	2.32	2.86
	0.12	0.18	0.23	0.29	0.59	1.16	1.74	2.32	2.87
	0.12	0.17	0.24	0.29	0.58	1.16	1.74	2.32	2.90

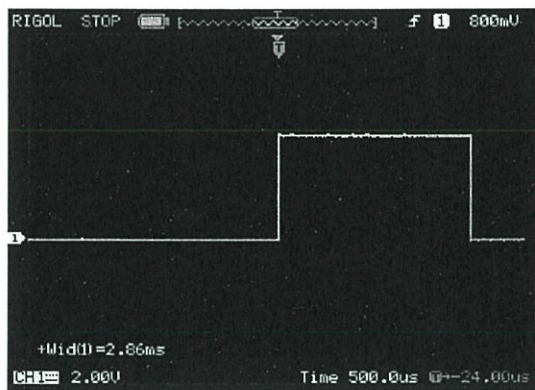
ระยะทาง (cm)	60	70	80	90	100	110	120	150	200
คาบเวลา ของ Pulse (ms)	3.49	4.07	4.64	5.23	5.83	6.61	7.20	9.20	12.50
	3.49	4.06	4.64	5.23	5.82	6.60	7.19	9.15	12.48
	3.49	4.07	4.65	5.24	5.80	6.59	7.20	9.18	12.40



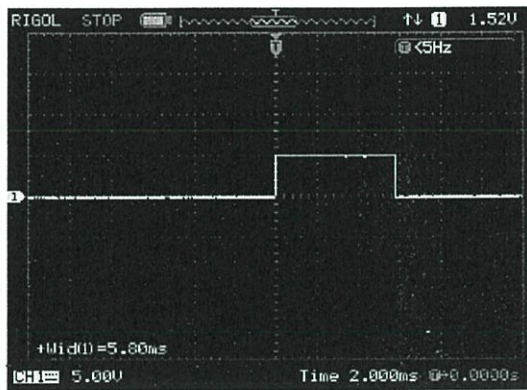
(ก) ระยะ 5 cm



(ข) ระยะ 10 cm



(ค) ระยะ 50 cm



(ง) ระยะ 100 cm

รูปที่ 4.28 สัญญาณจาก Oscilloscope ที่ระยะทางต่างๆ จากขาเอาต์พุตของ Ultrasonic Sensor

4.10 ผลการทดลองการวัดระยะทางเมื่อเปรียบเทียบระหว่างการวัดโดย Microcontroller กับการวัดโดยตลับเมตร

คำนวณระยะทางจากความกว้าง Pulse ที่รับได้จาก Sensor จากนั้นนำค่าระยะทางที่วัดได้จาก Microcontroller มาเปรียบเทียบกับค่าระยะทางที่ได้จากการวัดโดยตลับเมตร ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 การทดลองวัดระยะทางการทำงานของ Ultrasonic Sensor

ระยะทาง (เซนติเมตร)						ค่าความผิดพลาด (%)
วัดโดย ตลับเมตร	วัดโดย Microcontroller			ระยะทาง เฉลี่ย	ค่าความ ผิดพลาด	
	ครั้งที่1	ครั้งที่2	ครั้งที่3			
2.00	1.90	2.07	2.07	2.01	0.01	0.50
3.00	2.93	3.10	2.93	2.99	0.01	0.33
4.00	3.97	3.97	4.14	4.03	0.03	0.75
5.00	5.17	5.00	5.00	5.06	0.06	1.20
10.00	10.67	10.17	10.00	10.28	0.28	2.80
20.00	20.00	20.00	20.00	20.00	0.00	0.00
30.00	30.00	30.00	30.00	30.00	0.00	0.00
40.00	40.00	40.00	40.00	40.00	0.00	0.00
50.00	49.31	49.48	50.00	49.60	0.40	0.80
60.00	60.17	60.17	60.17	60.17	0.17	0.28
70.00	70.17	70.00	70.17	70.11	0.11	0.16
80.00	80.00	80.00	80.17	80.57	0.57	0.71
90.00	90.17	90.17	90.34	90.23	0.23	0.26
100.00	100.52	100.34	100.00	100.29	0.29	0.29
110.00	113.97	113.79	113.62	113.79	3.79	3.44
120.00	124.14	123.97	124.14	124.08	4.08	3.40
150.00	158.62	157.76	158.28	158.22	8.22	5.48
200.00	215.52	215.17	213.79	214.83	14.83	7.41

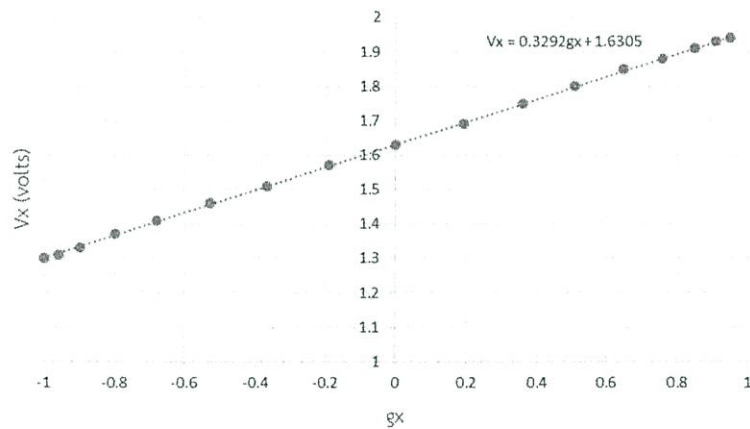
4.11 ผลการทดลองกราฟความสัมพันธ์ระหว่างค่าความเร่งของไมโคร Accelerometer กับแรงดันทางไฟฟ้า

เก็บค่าความเร่ง (g) และค่าแรงดันทางไฟฟ้า ซึ่งได้จากการคำนวณโดย Microcontroller และค่าแรงดันทางไฟฟ้าจะเปลี่ยนแปลงตามค่าความเร่ง (g) เพื่อแสดงถึงความ เป็นเชิงเส้นและอัตราการเปลี่ยนแปลงของค่าแรงดันทางไฟฟ้าเมื่อเทียบกับค่าความเร่ง (g)

ตารางที่ 4.3 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน X (g_x) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ X-Z

มุม (องศา)	-80	-70	-60	-50	-40	-30	-20	-10	0
g_x	-1.00	-0.96	-0.90	-0.80	-0.68	-0.53	-0.37	-0.19	0.00
V (Volt)	1.30	1.31	1.33	1.37	1.41	1.46	1.51	1.57	1.63

มุม (องศา)	10	20	30	40	50	60	70	80	90
g_x	0.19	0.36	0.51	0.65	0.76	0.85	0.91	0.95	0.95
V (Volt)	1.69	1.75	1.80	1.85	1.88	1.91	1.93	1.94	1.94

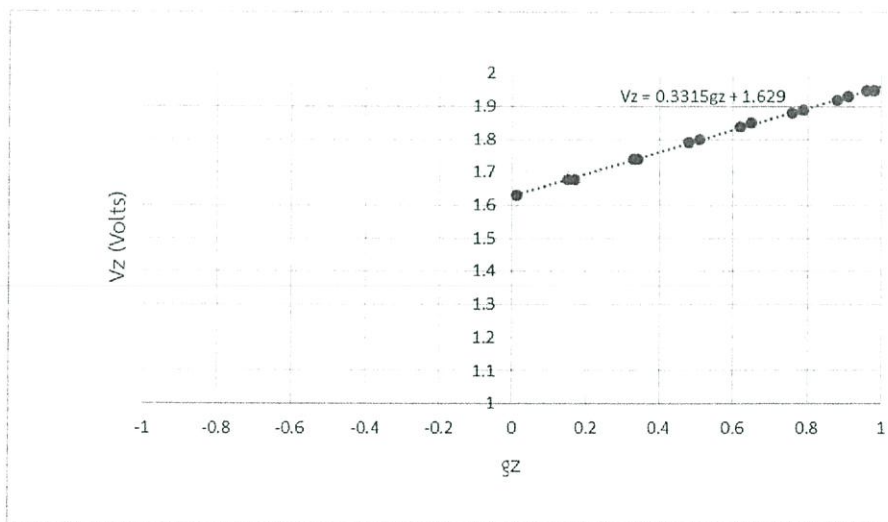


รูปที่ 4.29 กราฟความสัมพันธ์ระหว่างความเร่ง g_x กับ V (Volts) ในระนาบ X-Z

ตารางที่ 4.4 ความสัมพันธ์ระหว่างความเร่ง g_z ในแนวแกน Z (g_z) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ X-Z

มุม (องศา)	-80	-70	-60	-50	-40	-30	-20	-10	0
g_z	0.17	0.34	0.51	0.65	0.79	0.91	0.98	1.00	1.00
V (Volt)	1.68	1.74	1.80	1.85	1.89	1.93	1.95	1.97	1.97

มุม (องศา)	10	20	30	40	50	60	70	80	90
g_z	1.00	0.96	0.88	0.76	0.62	0.48	0.33	0.15	0.01
V (Volt)	1.96	1.95	1.92	1.88	1.84	1.79	1.74	1.68	1.63

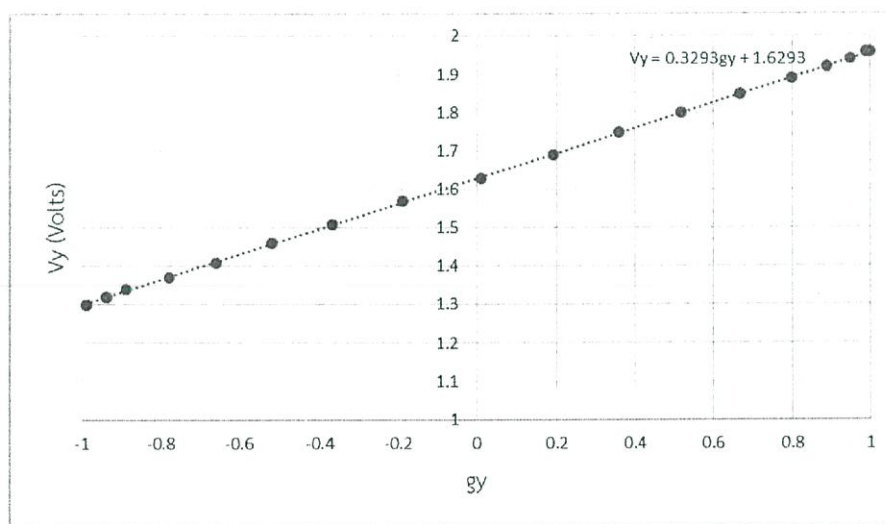


รูปที่ 4.30 กราฟความสัมพันธ์ระหว่างความเร่ง g_z กับ V (Volts) ในระนาบ X-Z

ตารางที่ 4.5 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน Y (g_y) กับค่าแรงดันทางไฟฟ้า (V) ที่มีมุมต่างๆ ในระนาบ Y-Z

มุม (องศา)	-80	-70	-60	-50	-40	-30	-20	-10	0
g_y	-0.99	-0.94	-0.89	-0.78	-0.66	-0.52	-0.37	-0.19	0.00
V (Volt)	1.30	1.31	1.33	1.37	1.41	1.46	1.51	1.57	1.63

มุม (องศา)	10	20	30	40	50	60	70	80	90
g_y	0.19	0.36	0.52	0.67	0.80	0.89	0.95	0.99	1.00
V (Volt)	1.69	1.75	1.80	1.85	1.89	1.92	1.94	1.96	1.96

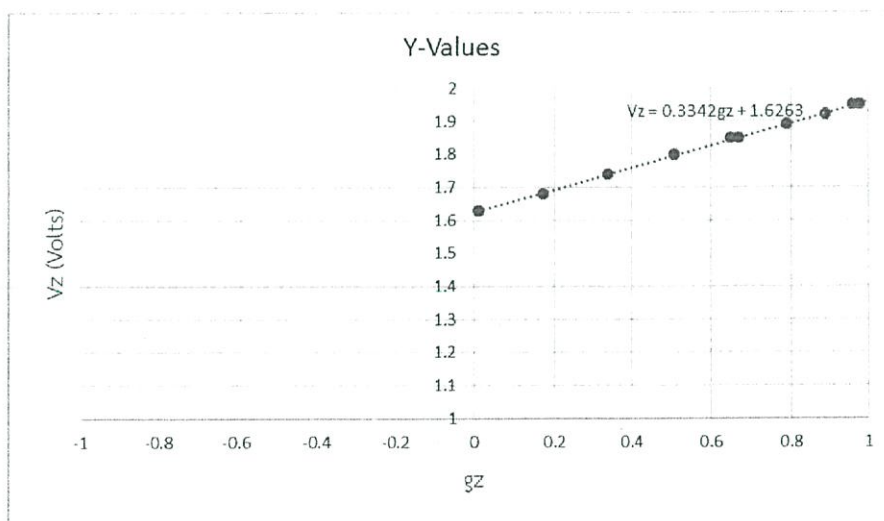


รูปที่ 4.31 กราฟความสัมพันธ์ระหว่างความเร่ง g_y กับ V (Volts) ในระนาบ Y-Z

ตารางที่ 4.6 ความสัมพันธ์ระหว่างความเร่ง g ในแนวแกน Z (g_z) กับค่าแรงดันทางไฟฟ้า (V) ที่มุมต่างๆ ในระนาบ Y-Z

มุม (องศา)	-80	-70	-60	-50	-40	-30	-20	-10	0
g_z	0.17	0.34	0.51	0.65	0.79	0.89	0.96	1.00	1.00
V (Volt)	1.68	1.74	1.80	1.85	1.89	1.92	1.95	1.97	1.97

มุม (องศา)	10	20	30	40	50	60	70	80	90
g_z	1.00	0.98	0.89	0.79	0.67	0.51	0.34	0.17	0.01
V (Volt)	1.97	1.95	1.92	1.89	1.85	1.80	1.74	1.68	1.63



รูปที่ 4.32 กราฟความสัมพันธ์ระหว่างความเร่ง g_z กับ V (Volts) ในระนาบ Y-Z

4.12 จัดเก็บผลการทดลองวัดค่ามุมเอียงที่วัดได้จริงโดยใช้ Protractor เปรียบเทียบกับค่าความเอียงที่วัดได้จาก Microcontroller แสดงผลทาง Hyperterminal

เก็บค่ามุมเอียงที่วัดได้โดย Microcontroller เปรียบเทียบกับมุมเอียงที่วัดได้จริงโดยใช้ Protractor

ตารางที่ 4.7 การทดลองวัดมุมเอียง

มุมเอียง (องศา)					ค่าความผิดพลาด	ค่าความผิดพลาด (%)
วัดโดย ไม้วัดมุม	วัดโดย Microcontroller			มุมเอียงเฉลี่ย		
	ครั้งที่1	ครั้งที่2	ครั้งที่3			
-80	-80	-80	-80	-80	0	0
-70	-70	-70	-70	-70	0	0
-60	-60	-60	-60	-60	0	0
-50	-50	-50	-50	-50	0	0
-40	-40	-40	-40	-40	0	0
-30	-30	-30	-30	-30	0	0
-20	-20	-20	-20	-20	0	0
-10	-10	-10	-10	-10	0	0
0	0	0	0	0	0	0
10	10	10	10	10	0	0
20	20	20	20	20	0	0
30	30	30	30	30	0	0
40	40	40	40	40	0	0
50	50	50	50	50	0	0
60	60	60	60	60	0	0
70	70	70	70	70	0	0
80	80	80	80	80	0	0
90	90	90	90	90	0	0

4.13 จัดเก็บระยะเวลาการรับ-ส่งสัญญาณระหว่างโมดูล Wi-Fi กับโทรศัพท์มือถือ Smart Phone และระยะรับ-ส่งภาพระหว่างโทรศัพท์มือถือ Smart Phone ทั้งสองเครื่อง ผ่าน Router

เก็บผลเพื่อทดสอบระยะทางที่รถสามารถเข้าถึงพื้นที่ใดๆ ได้ระยะทางสูงสุดทั้งแบบ Line-of-sight และ Non-line-of-sight โดยภาพจะแบ่งเป็นความละเอียด 176×144 และ 320×240 ตามลำดับ

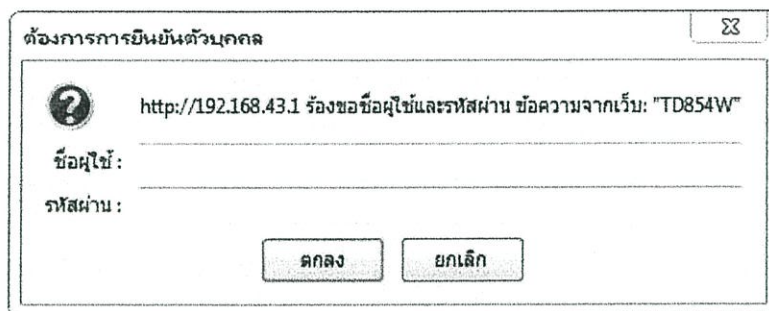
ตารางที่ 4.8 การทดลองวัดระยะทางไกลสุด

ครั้งที่	ระยะรับภาพ (เมตร)		ระยะรับ-ส่งข้อมูล (เมตร)	
	Line-of-sight	Non-line-of-sight	Line-of-sight	Non-line-of-sight
1	36.0	28.0	108.0	53.0
	29.0	22.0		
2	35.0	30.0	110.0	55.0
	27.0	23.0		
3	38.0	27.0	102.0	50.0
	30.0	20.0		
เฉลี่ย	36.3	28.3	106.7	52.7
	28.7	21.7		

4.14 การตั้งค่า Router เพื่อให้สามารถเชื่อมต่อภายในวง LAN เดียวกัน

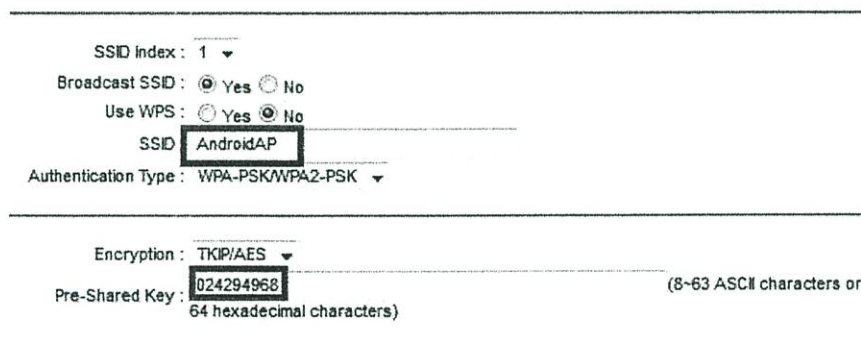
เนื่องจากการใช้โทรศัพท์มือถือ Smart Phone ให้มีหน้าที่ในการกระจายสัญญาณมีระยะทางที่ไม่มาก จึงได้ใช้ Router ซึ่งทำหน้าที่กระจายสัญญาณมาทำหน้าที่แทน โดยก่อนใช้งาน ต้องมีการตั้งค่าดังนี้

1) เสียบสาย LAN เข้า Router แล้วเชื่อมต่อไปยังคอมพิวเตอร์ แล้วเปิด Browser จากนั้นพิมพ์เลข IP ของ Router (ในที่นี้คือ 192.168.43.1) จากนั้นใส่ Username และ Password เพื่อเข้าสู่โหมดการตั้งค่า



รูปที่ 4.33 การใส่ Username และ Password สำหรับเข้าไปโหมดตั้งค่า

2) เข้าไปเซต SSID ของ Router ให้มีชื่อว่า “AndroidAP” และมี Password “024294968”



รูปที่ 4.34 การตั้งค่า SSID และ Password ของ Router

3) เข้าไปเซตให้ IP 192.168.43.47 (Static IP) เป็นของโมดูล Wi-Fi โดยนำเลข MAC Address มาใช้ในการตั้งค่า

Hostname	IP Address	MAC Address	Status	Expire Time
	192.168.43.46	Manual Config	Static	
	192.168.43.47	00:06:66:80:34:EF	Static	N/A
Poomtelecom-PC	192.168.43.45	C8:0A:A9:82:F1:53	Auto	2days, 23:55:21

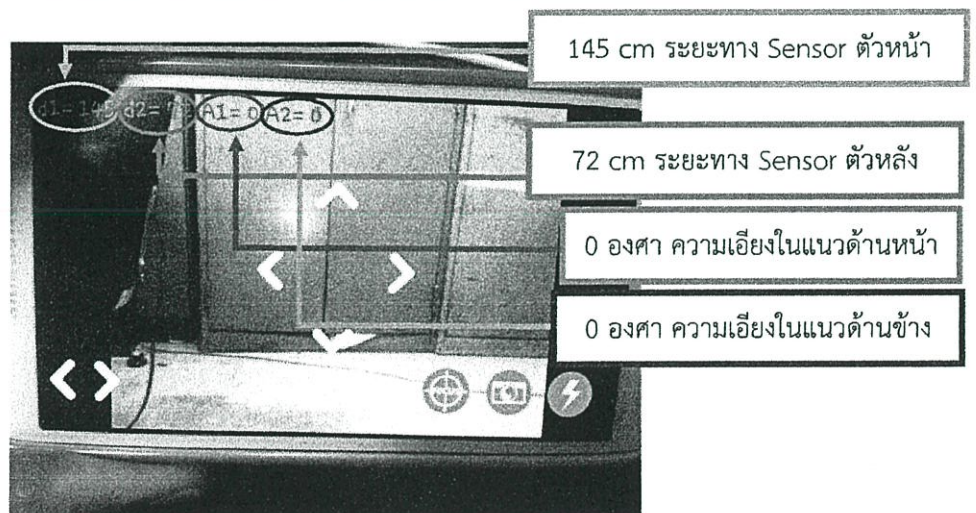
รูปที่ 4.35 การตั้งค่า IP ให้เป็น Static เพื่อให้โมดูล Wi-Fi สามารถเชื่อมต่อได้

4.15 การทดสอบ Application ในการรับข้อมูลจาก Sensor เมื่อเข้าถึงระยะที่กำหนด

เนื่องจากการออกแบบ Application ได้มีการออกแบบให้แสดงภาพออกมาอย่างฝั่งบังคับและสามารถรับค่าจาก Sensor มาแสดงบนหน้าจอโทรศัพท์มือถือ Smart Phone และเมื่อถึงระยะที่กำหนดจะมีการแจ้งเตือนผู้ใช้ซึ่งผลของหน้าจอเป็นดังนี้



รูปที่ 4.36 การแจ้งเตือนบนหน้าจอโทรศัพท์มือถือ Smart Phone เมื่อถึงระยะที่กำหนด



รูปที่ 4.37 ภาพที่ถูกส่งมาจากโทรศัพท์มือถือ Smart Phone

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการนี้เป็นกรออกแบบการควบคุมรถสำรวจเพื่อใช้สำรวจในพื้นที่ที่เข้าไม่ถึง ซึ่งประกอบไปด้วย 5 องค์ประกอบหลักคือ 1. Application Android ที่ใช้ในการควบคุม 2. โมดูล Wifly RN-XV 3. Microcontroller 4. วงจรควบคุมการหมุนของมอเตอร์ 5. Sensor ป้องกันความเสียหายแก่อุปกรณ์ ในภาคเรียนที่ 1 ในส่วนของ Application ได้มีการออกแบบให้สามารถควบคุมรถโดยการควบคุมผ่านโปรโตคอล TCP/IP โดยผ่านการเชื่อมต่อ Hotspot ระหว่างโทรศัพท์มือถือกับตัวโมดูล Wifly RN-XV จากนั้นค่าที่ได้จะถูกส่งไปยังส่วนของ Microcontroller เพื่อใช้ควบคุมมอเตอร์ 2 ตัว ทำให้เราสามารถบังคับรถผ่านโทรศัพท์มือถือระบบปฏิบัติการ Android ได้ ส่วนในภาคเรียนที่ 2 เราได้เพิ่มในส่วนการติดตั้งกล้องเพื่อตรวจสอบสภาพแวดล้อมโดยรอบที่รถวิ่งเข้าไปถึง โดยการพัฒนา Application ให้ทำหน้าที่เป็น IP Camera และได้มีการเพิ่มส่วนของ Sensor ได้แก่ Ultrasonic Sensor และโมดูลวัดความแรงซึ่งทำหน้าที่วัดความเอียง พร้อมทั้งสามารถส่งค่าของ Sensor ไปยังหน้าจอโทรศัพท์มือถือ Smart Phone เพื่อแจ้งเตือนให้ผู้ใช้เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นแก่อุปกรณ์ นอกจากนี้พบว่าในการส่งภาพด้วยการเชื่อมต่อแบบ Hotspot บนโทรศัพท์มือถือ Smart Phone ไม่สามารถส่งภาพได้อย่างมีประสิทธิภาพ จึงได้มีการเพิ่ม Wi-Fi Router เพื่อทำหน้าที่แทน Hotspot โทรศัพท์มือถือ Smart Phone ซึ่งทำให้ภาพที่ส่งนั้นมีประสิทธิภาพมากขึ้น

5.2 ข้อเสนอแนะ

1. อาจใช้เครือข่ายอินเทอร์เน็ต เพื่อเพิ่มระยะทางการควบคุมให้มากขึ้น
2. เสริมฟังก์ชันที่เป็นประโยชน์แก่การสำรวจ เช่น วัดอุณหภูมิ เป็นต้น
3. ทำให้รถสามารถเปลี่ยนแปลงความเร็วในการวิ่งได้
4. สามารถใช้พลังงานทางเลือกเป็นแบตเตอรี่ในการทำงานของระบบ เช่น พลังงานแสงอาทิตย์

ภาคผนวก
อุปกรณ์ที่ใช้

FEATURES

3-axis sensing

Small, low profile package

4 mm \times 4 mm \times 1.45 mm LFCSP

Low power : 350 μ A (typical)

Single-supply operation: 1.8 V to 3.6 V

10,000 g shock survival

Excellent temperature stability

BW adjustment with a single capacitor per axis

RoHS/WEEE lead-free compliant

APPLICATIONS

Cost sensitive, low power, motion- and tilt-sensing applications

Mobile devices

Gaming systems

Disk drive protection

Image stabilization

Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of $\pm 3 g$. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_X , C_Y , and C_Z capacitors at the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm \times 4 mm \times 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

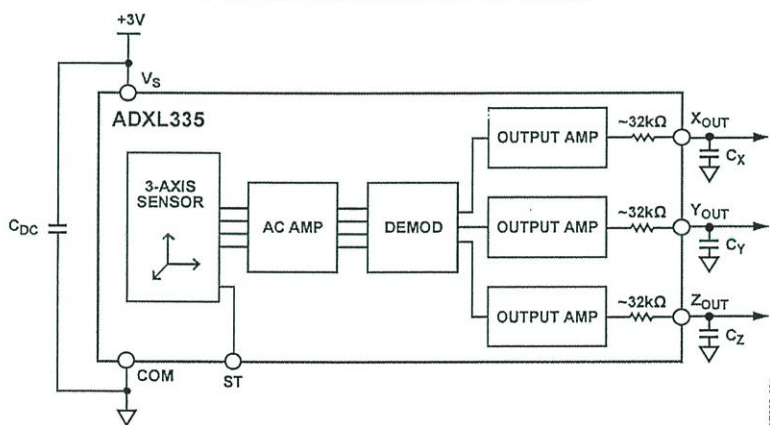


Figure 1.

Rev. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 3\text{ V}$, $C_X = C_Y = C_Z = 0.1\ \mu\text{F}$, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis	± 3	± 3.6		g
Nonlinearity	% of full scale		± 0.3		%
Package Alignment Error			± 1		Degrees
Interaxis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ¹			± 1		%
SENSITIVITY (RATIOMETRIC)²					
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	Each axis $V_S = 3\text{ V}$	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	$V_S = 3\text{ V}$		± 0.01		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X_{OUT} , Y_{OUT}	$V_S = 3\text{ V}$	1.35	1.5	1.65	V
0 g Voltage at Z_{OUT}	$V_S = 3\text{ V}$	1.2	1.5	1.8	V
0 g Offset vs. Temperature			± 1		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density X_{OUT} , Y_{OUT}			150		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
Noise Density Z_{OUT}			300		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
FREQUENCY RESPONSE⁴					
Bandwidth X_{OUT} , Y_{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z_{OUT} ⁵	No external filter		550		Hz
R_{FILT} Tolerance			$32 \pm 15\%$		k Ω
Sensor Resonant Frequency			5.5		kHz
SELF-TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3\text{ V}$		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S .

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X , C_Y , C_Z).

⁵ Bandwidth with external capacitors = $1/(2 \times \pi \times 32\text{ k}\Omega \times C)$. For C_X , $C_Y = 0.003\ \mu\text{F}$, bandwidth = 1.6 kHz. For $C_Z = 0.01\ \mu\text{F}$, bandwidth = 500 Hz. For C_X , C_Y , $C_Z = 10\ \mu\text{F}$, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S .

⁷ Turn-on time is dependent on C_X , C_Y , C_Z and is approximately $160 \times C_X$ or C_Y or $C_Z + 1\text{ ms}$, where C_X , C_Y , C_Z are in microfarads (μF).

ADXL335

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 <i>g</i>
Acceleration (Any Axis, Powered)	10,000 <i>g</i>
V _s	-0.3 V to +3.6 V
All Other Pins	(COM - 0.3 V) to (V _s + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	-55°C to +125°C
Temperature Range (Storage)	-65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

ADXL335

USE WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL335 is tested and specified at $V_s = 3\text{ V}$; however, it can be powered with V_s as low as 1.8 V or as high as 3.6 V. Note that some performance parameters change as the supply voltage is varied.

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At $V_s = 3.6\text{ V}$, the output sensitivity is typically 360 mV/g. At $V_s = 2\text{ V}$, the output sensitivity is typically 195 mV/g.

The zero g bias output is also ratiometric, thus the zero g output is nominally equal to $V_s/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/g) increases while the noise voltage remains constant. At $V_s = 3.6\text{ V}$, the X-axis and Y-axis noise density is typically $120\text{ }\mu\text{g}/\sqrt{\text{Hz}}$, whereas at $V_s = 2\text{ V}$, the X-axis and Y-axis noise density is typically $270\text{ }\mu\text{g}/\sqrt{\text{Hz}}$.

Self-test response in g is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at $V_s = 3.6\text{ V}$, the self-test response for the ADXL335 is approximately -560 mV for the X-axis, $+560\text{ mV}$ for the Y-axis, and $+950\text{ mV}$ for the Z-axis.

At $V_s = 2\text{ V}$, the self-test response is approximately -96 mV for the X-axis, $+96\text{ mV}$ for the Y-axis, and -163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at $V_s = 3.6\text{ V}$ is $375\text{ }\mu\text{A}$, and typical current consumption at $V_s = 2\text{ V}$ is $200\text{ }\mu\text{A}$.

AXES OF ACCELERATION SENSITIVITY

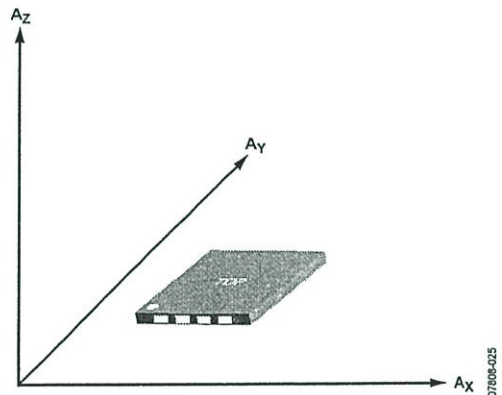


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

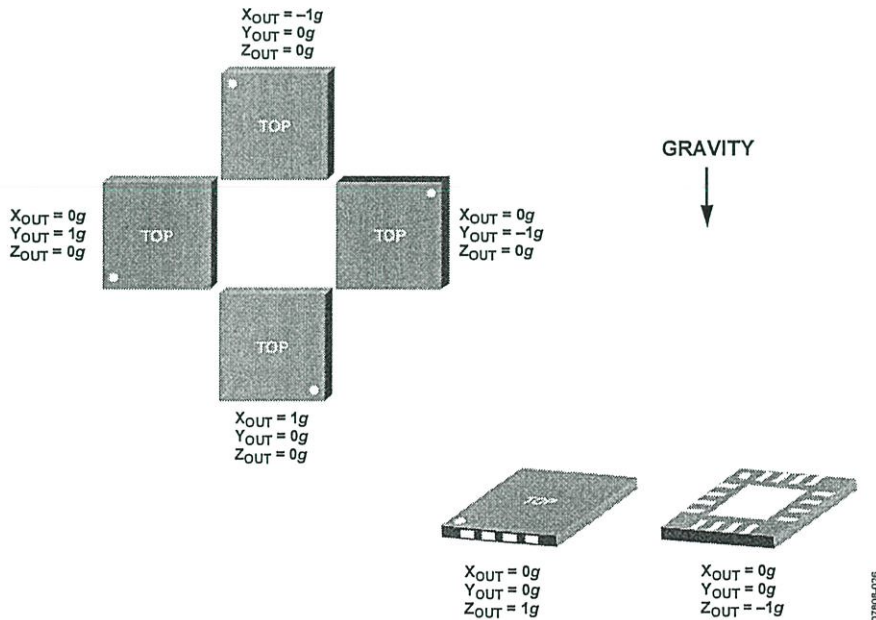


Figure 24. Output Response vs. Orientation to Gravity

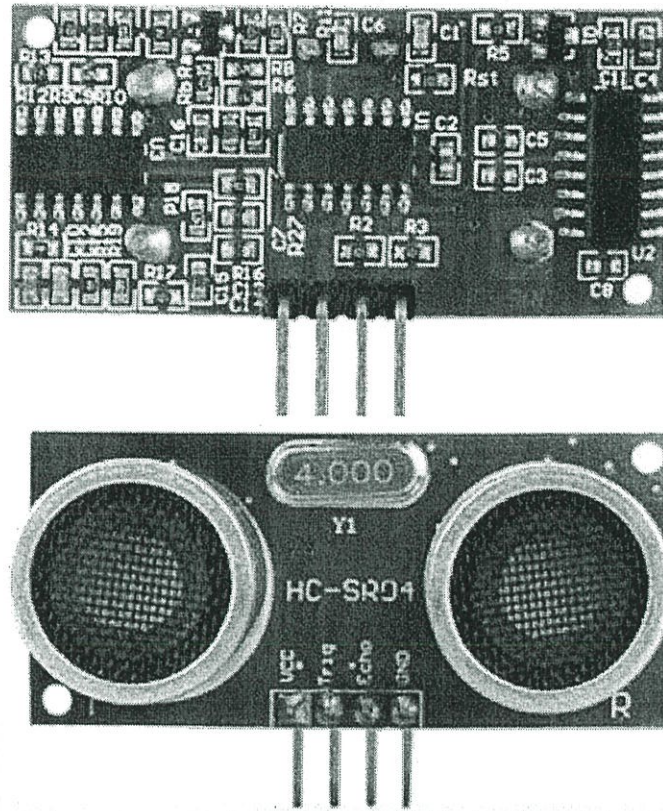
1.0 INTRODUCTION

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1" to 13 feet. Its operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

Features:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1" - 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

3.0 PRODUCT LAYOUT

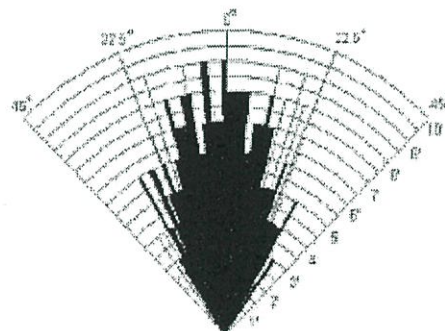
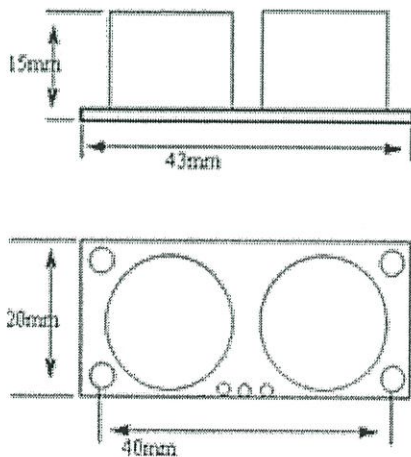


VCC = +5VDC

Trig = Trigger input of Sensor

Echo = Echo output of Sensor

GND = GND



Practical test of performance,
Best in 30 degree angle

4.0 PRODUCT SPECIFICATION AND LIMITATIONS

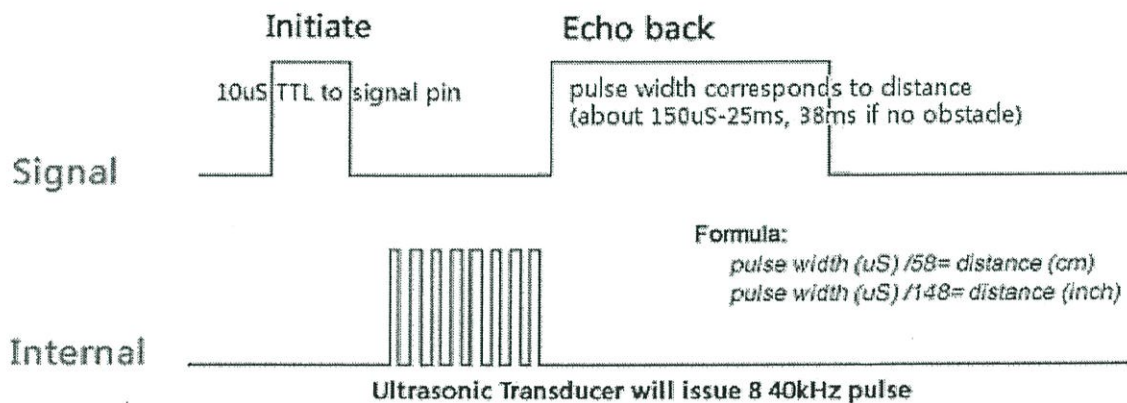
Parameter	Min	Typ.	Max	Unit
Operating Voltage	4.50	5.0	5.5	V
Quiescent Current	1.5	2	2.5	mA
Working Current	10	15	20	mA
Ultrasonic Frequency	-	40	-	kHz

5.0 OPERATION

The timing diagram of HC-SR04 is shown. To start measurement, Trig of SR04 must receive a pulse of high (5V) for at least 10us, this will initiate the sensor will transmit out 8 cycle of ultrasonic burst at 40kHz and wait for the reflected ultrasonic burst. When the sensor detected ultrasonic from receiver, it will set the Echo pin to high (5V) and delay for a period (width) which proportion to distance. To obtain the distance, measure the width (Ton) of Echo pin.

Time = Width of Echo pulse, in uS (micro second)

- Distance in centimeters = Time / 58
- Distance in inches = Time / 148
- Or you can utilize the speed of sound, which is 340m/s



Note:

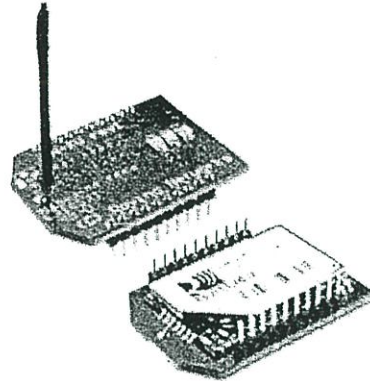
- Please connect the GND pin first before supplying power to VCC.
- Please make sure the surface of object to be detect should have at least 0.5 meter² for better performance.

Features

- Drop in Wi-Fi solution for existing systems currently using 802.15.4 modules
- Based on Roving Networks' robust RN-171 Wi-Fi module
- Based on pseudo-standard footprint
- Onboard TCP/IP stack provides internet access to every node
- No custom profiles needed
- Ultra-low power: 4uA sleep, 40mA Rx, 180 mA Tx at 10dBm
- Configurable transmit power: 0dBm to +12dBm
- Hardware interface: TTL UART
- Data rate: 464Kbps using hardware flow control
- Through hole board simplifies system integration
- 8 general purpose digital I/O
- 3 analog sensor interfaces.
- Real-time clock for wakeup and time stamping
- Complete TCP/IP networking stack
- Wi-Fi Alliance certified for WEP, WPA and WPA2-PSK
- WPS push button for easy configuration
- FCC / CE/ ICS certified and RoHS compliant.

Applications

- Industrial metering
- HVAC control
- Room temperature sensors
- Pump configuration and control
- Telemetry
- PV/Solar controllers
- Robotics



Description

The RN-XV is a 802.11 b/g solution especially designed for customer who want to migrate their existing 802.15.4 architecture to a more standard TCP/IP based platform without having to redesign their existing hardware.

Based on a pseudo standard footprint often found in embedded applications, the RN-XV module from Roving Networks allows for Wi-Fi connectivity using 802.11 b/g standards in legacy and existing designs that may have been based upon 802.15.4 standard.

The RN-XV module is based upon Roving Networks' robust RN-171 Wi-Fi module and incorporates 802.11 b/g radio, 32 bit SPARC processor, TCP/IP stack, real-time clock, crypto accelerator, power management unit and analog sensor interface.

The module offers additional functionality through its onboard programmable GPIOs (10) and ADCs (8). The ADCs provide 14-bit resolution while the GPIOs can be configured to provide standard functionality or status signaling to a host microcontroller to reduce the need for serial polling between the Wi-Fi module and host microcontroller.

The module is pre-loaded with firmware to simplify integration and minimize development time of your application. In the simplest configuration, the hardware only requires four connections (PWR, TX, RX and GND) to create a wireless data connection.

Overview

- Host Data Rate up to 460Kbps over UART
- Intelligent built-in power management with programmable wakeup events (timers and I/O)
- Real time clock for time stamping, auto-sleep and auto-wakeup modes
- Configuration over WiFi or UART using simple ASCII commands
- Over the air firmware upgrade via FTP
- Secure WiFi authentication: WEP, WPA-PSK (TKIP), WPA2-PSK
- Built in networking applications DHCP, DNS, ARP, ICMP UDP, Telnet, FTP, HTML client
- 802.11 b/g power save and roaming functions
- Configurable transmit power: -2dBm to 12dBm
- WPS push button mode for easy and secure wireless setup
- Built-in HTML web client to send GPIO, UART and sensor data to remote web servers

Environmental Conditions

Parameter	Value
Temperature Range (Operating)	-40 °C - 85 °C
Temperature Range (Storage)	-40°C - 85 °C
Relative Humidity (Operating)	≤90%
Relative Humidity (Storage)	≤90%

Electrical Characteristics

Supply Voltage	Min	Typ.	Max.	Unit
Input Power	3.0	3.3	3.7	V
Power consumption				
Sleep		4		uA
Standby (doze)		15		mA
Connected (idle, RX)		40		mA
Connected (TX)		180 at 10dBm		mA

Analog Sensor Inputs

Parameter	Value
Sense 0,1,2,3 wakeup detect threshold	500mV
AD sense 0-4 measurement range	0-400mV
Precision	14 bits = 12uV
Accuracy	5% un-calibrated, .01% calibrated
Minimum conversion time	35uS (5kHz over Wi-Fi)
Sensor Power (pin 33) output resistance 3.3V	10 ohms, max current = 50mA

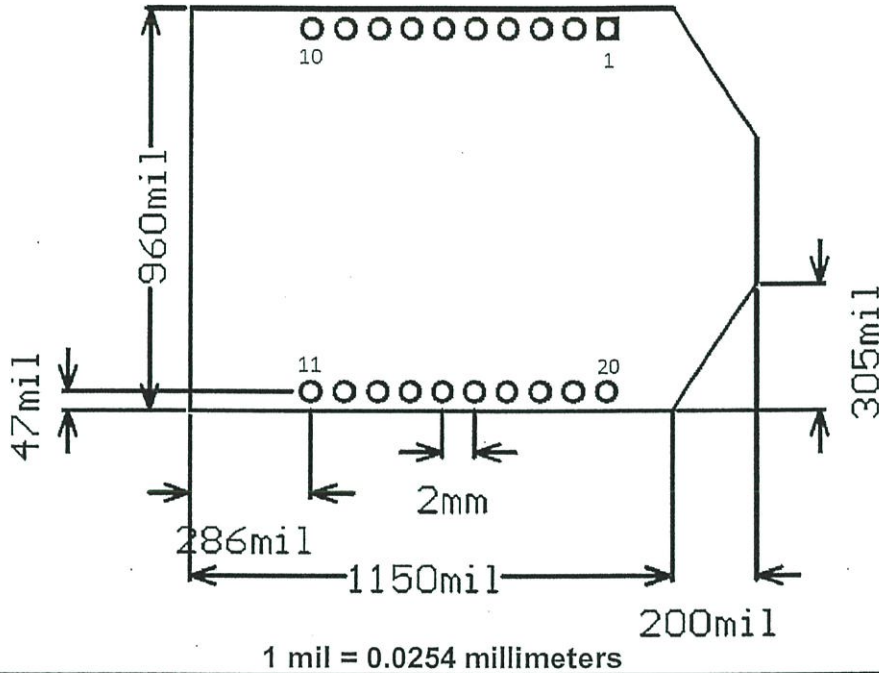
Radio Characteristics

Parameter	Specifications
Frequency	2402 ~ 2480MHz
Modulation	802.11b compatibility : DSSS(CCK-11, CCK-5.5, DQPSK-2, DBPSK-1) 802.11g : OFDM (default)
Channel intervals	5MHz
Channels	1 - 14
Transmission rate (over the air)	1 – 11Mbps for 802.11b / 6 – 54Mbps for 802.11g
Receive sensitivity	-83dBm typ.
Output level (Class1)	0dBm to +12dBm (software configurable)

Transmit Power

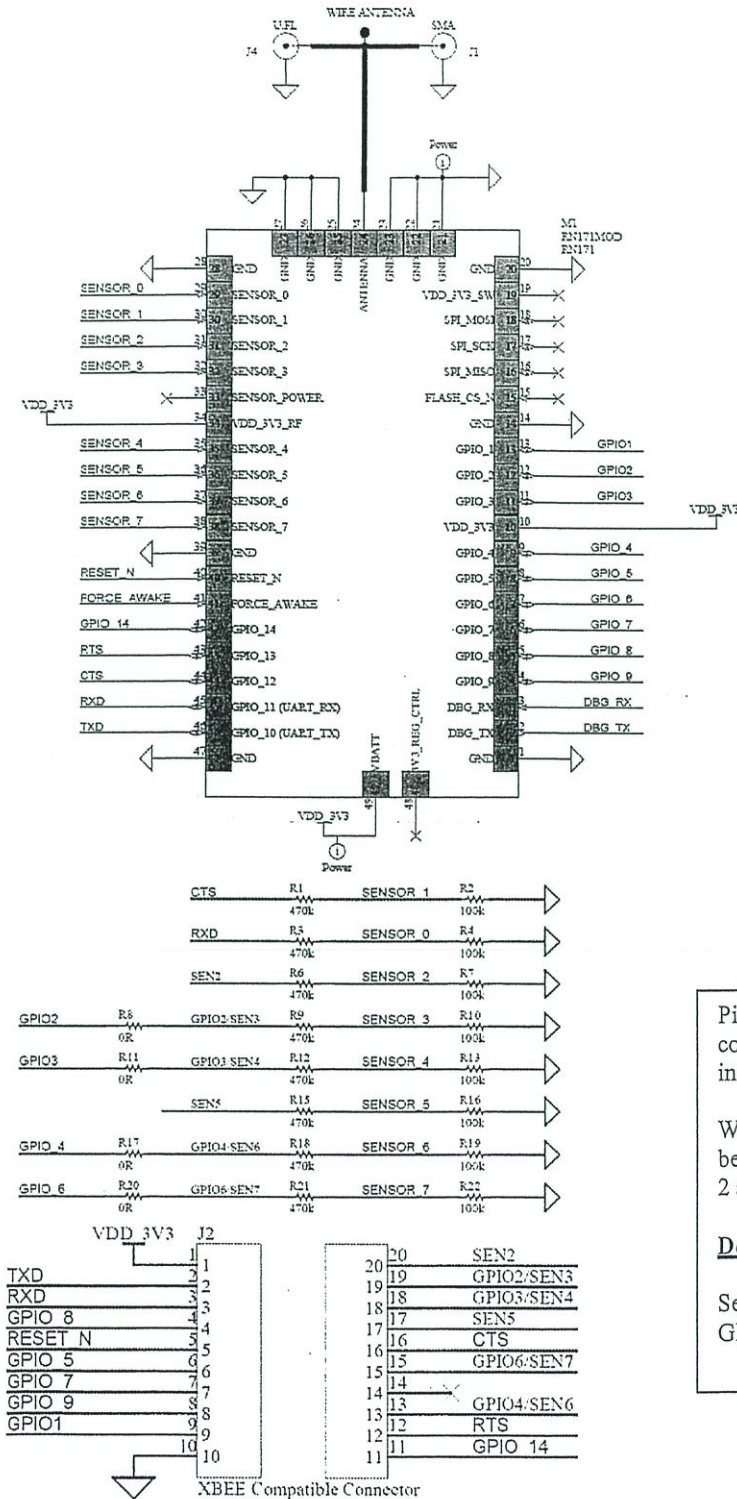
Output Power	802.11 b (2Mbps) Current in mA*	802.11 g (24Mbps) Current in mA*
0	120	135
2	130	150
4	170	190
6	175	200
8	180	210
10	185	225
12	190	240

* Measured at 3.3VDC input. The power consumption is the average power, active during actual power consumption

Physical Dimensions and pin out table


Pad Number	Signal Name	Description	Optional Function	Direction
1	VDD 3V3	3.3V regulated power input to the module		POWER
2	UART TX	UART TX, 8mA drive, 3.3V tolerant		OUT →
3	UART RX	UART RX, 3.3V tolerant		IN ←
4	GPIO 8	GPIO, 24mA drive, 3.3V tolerant		IN / OUT
5	RESET	Optional Module Reset Signal (active low), 100k Pull up, apply pulse of at least 160us, 3.3V Tolerant		INPUT
6	GPIO 5	GPIO, 24mA drive, 3.3V tolerant	Data TX/RX	OUT
7	GPIO 7	GPIO, 24mA drive, 3.3V tolerant		IN / OUT
8	GPIO 9	Enable Adhoc mode, Restore factory defaults, 8mA drive, 3.3V tolerant		IN / OUT
9	GPIO 1	GPIO, 8mA drive, 3.3V tolerant		IN / OUT
10	GND	Ground		GND
11	GPIO 14	GPIO, 8mA drive, 3.3V tolerant		IN / OUT
12	UART RTS	UART RTS flow control, 8mA drive, 3.3V tolerant		OUT →
13	GPIO 4/SEN 6	GPIO, 24mA drive, 3.3V tolerant/ADC input , (3.3V tolerant). Defaults to GPIO 4	Association Status	IN / OUT
14	Not Used			No Connect
15	GPIO 6/SEN 7	GPIO, 24mA drive, 3.3V tolerant/ADC input , (3.3V tolerant). Defaults to GPIO 6	Connection Status	POWER
16	UART CTS	UART CTS flow control, 3.3V tolerant		IN ←
17	SENSOR 5	Sensor Interface, Analog input to module, (3.3V tolerant)		INPUT
18	GPIO 3/SEN 4	GPIO, 8mA drive, 3.3V tolerant/ADC input (3.3V tolerant). Defaults to GPIO 3		IN / OUT
19	GPIO 2/SEN 3	GPIO, 8mA drive, 3.3V tolerant/ADC input (3.3V tolerant). Defaults to SEN 3		IN / OUT
20	SEN 2	Sensor Interface, Analog input to module, 3.3V tolerant		INPUT

Module Schematic



Pins 13, 15, 18 and 19 on the RN-XV connector can be configured as GPIOs or as sensor inputs depending on the installed resistors.

When configured as GPIOs, ONLY Column 1 resistors MUST be installed. When configured as sensor inputs, ONLY Column 2 and Column 3 resistors MUST be stuffed.

Default configuration:

Sensor inputs – Pins 19 and 20
 GPIO – Pins 13, 15 and 18

Design Concerns

1. **Powering the RN-XV module:** Apply ONLY 3.3V±10% regulated power to pin 1 (VDD) and pin 10 (Ground). The module does not have any voltage regulator on board and hence MUST be powered from a regulated 3.3V power supply.
2. **Antenna:** The RN-XV ships with a wire antenna. Custom antenna configurations such as chip, U.FL. and SMA connector are available with extended lead times

Ordering Information

Part Number	Description
RN-XV-W	Standard configuration, industrial Temperature (- 40 to + 85 C) 802.15.4 replacement solution with ¼ inch wire antenna
RN-XV-U	Custom configuration, industrial Temperature (- 40 to + 85 C) 802.15.4 replacement solution with U.FL. connector
RN-XV-S	Custom configuration, industrial Temperature (- 40 to + 85 C) 802.15.4 replacement solution with SMA connector
RN-XV-C	Custom configuration, industrial Temperature (- 40 to + 85 C) 802.15.4 replacement solution with chip antenna

Copyright © 2011 Roving Networks. All rights reserved.

Roving Networks reserves the right to make corrections, modifications, and other changes to its products, documentation and services at any time. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Roving Networks assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Roving Networks components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Roving Networks products are not authorized for use in safety-critical applications (such as life support) where a failure of the Roving Networks product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use.

All other trademarks are property of their respective owners.

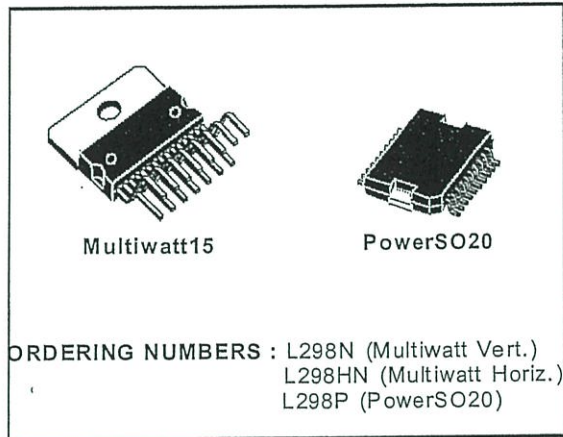


DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

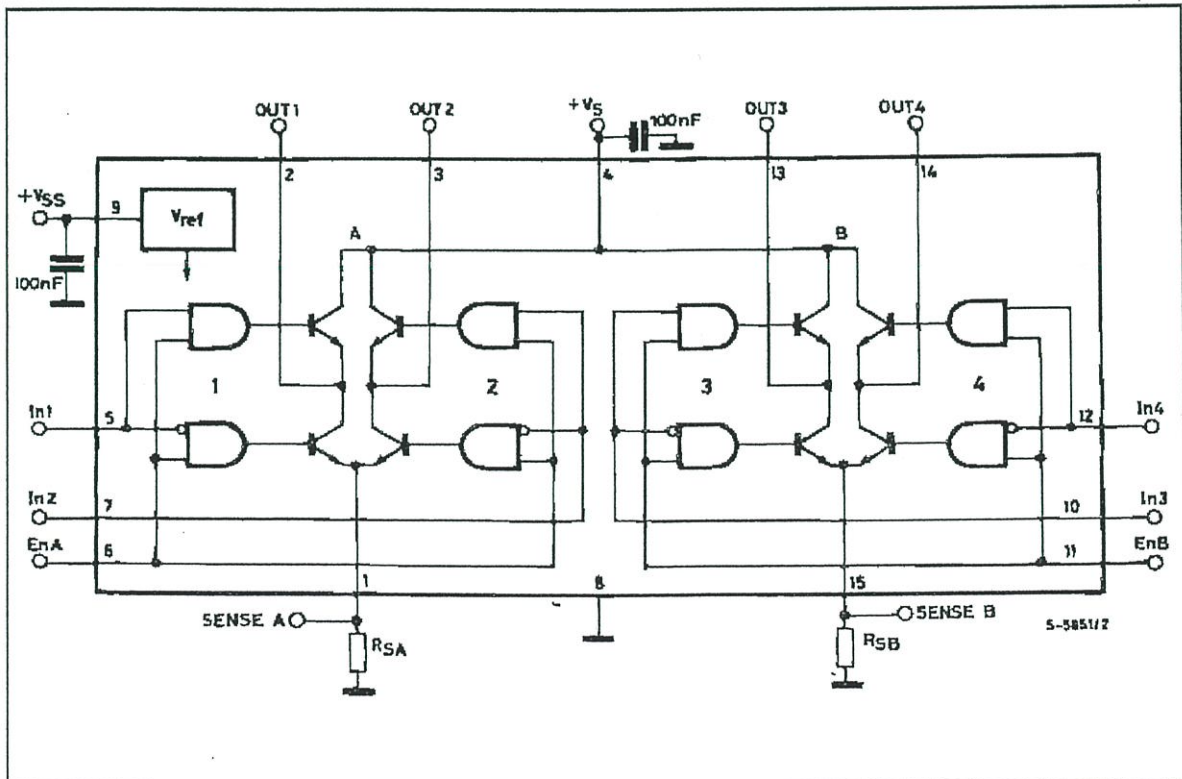
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

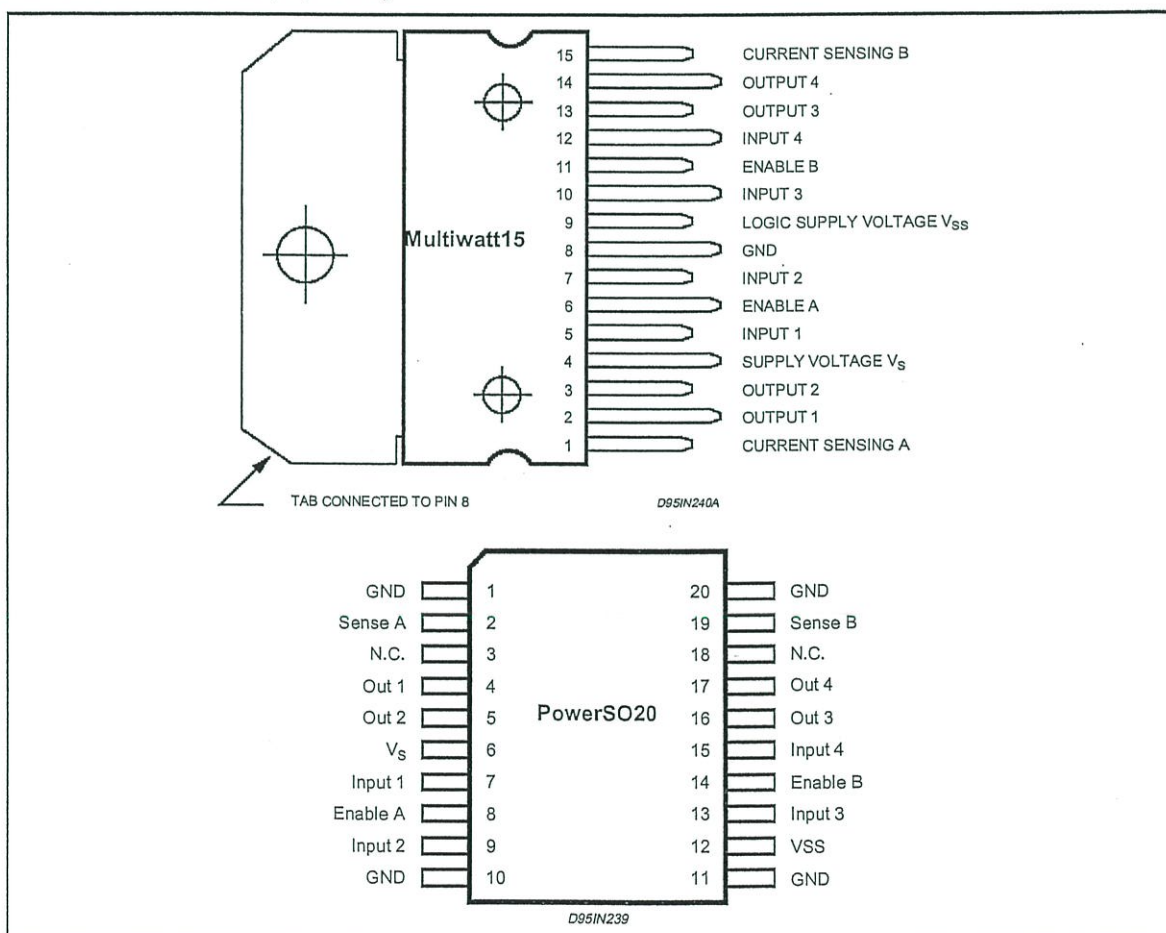
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{thj-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{thj-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _s	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_s = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _s	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _s	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0	V _i = L	13	22	mA
			V _i = H	50	70	mA
		V _{en} = L	V _i = X		4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0	V _i = L	24	36	mA
			V _i = H	7	12	mA
		V _{en} = L	V _i = X		6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat(H)}	Source Saturation Voltage	I _L = 1A	0.95	1.35	1.7	V
		I _L = 2A		2	2.7	V
V _{CEsat(L)}	Sink Saturation Voltage	I _L = 1A (5)	0.85	1.2	1.6	V
		I _L = 2A (5)		1.7	2.3	V
V _{CEsat}	Total Drop	I _L = 1A (5)	1.80		3.2	V
		I _L = 2A (5)			4.9	V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

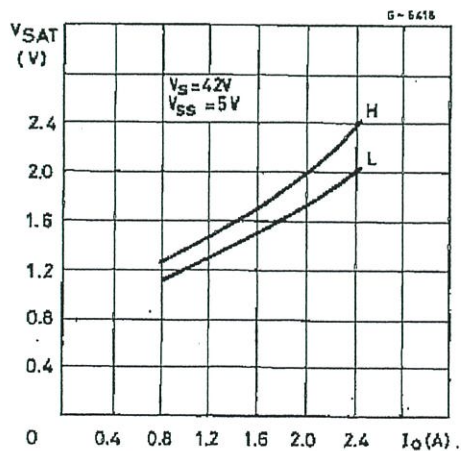
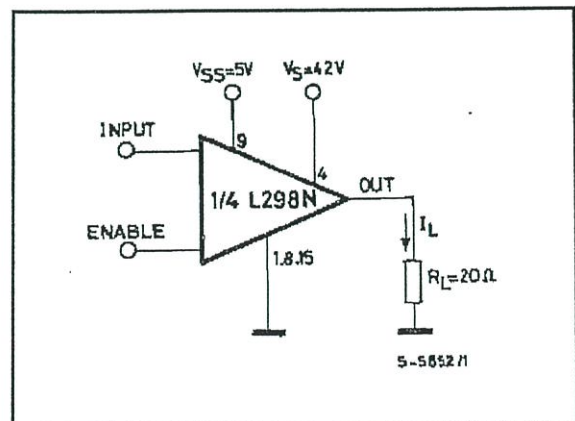


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H



Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

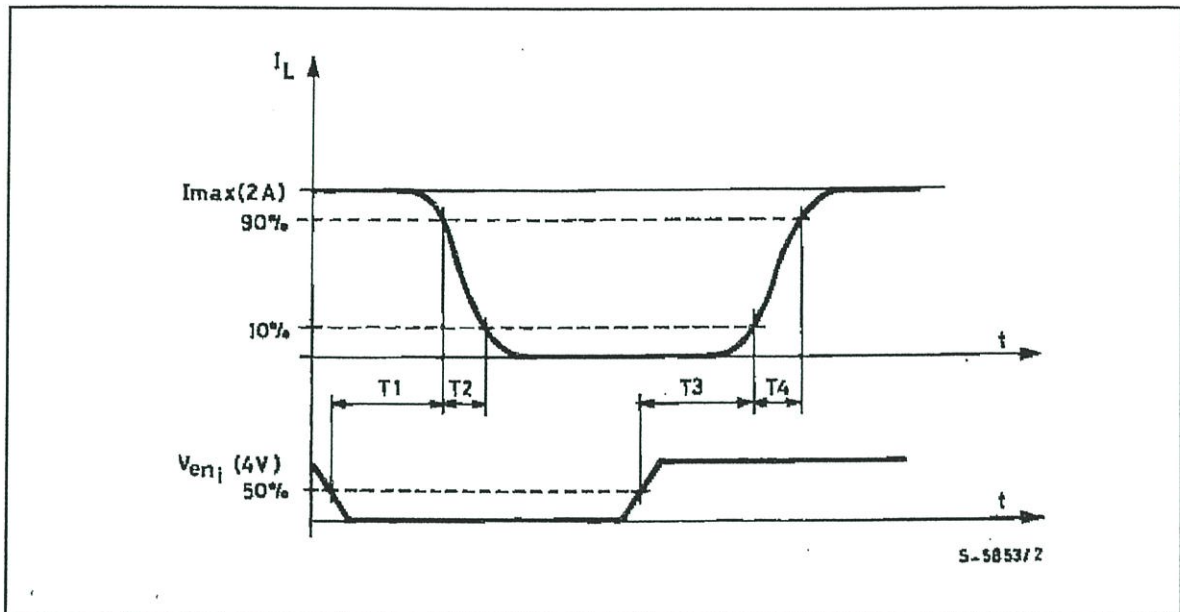
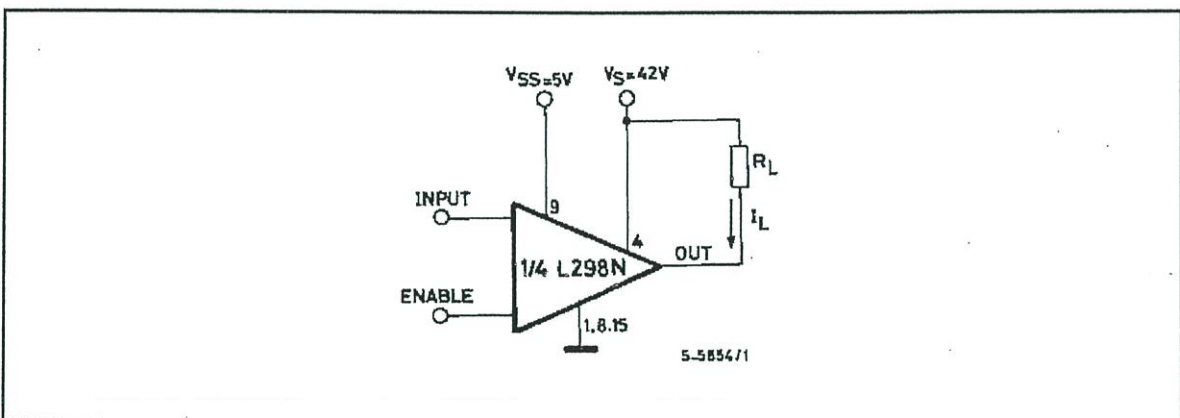


Figure 4 : Switching Times Test Circuits.



Note: For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

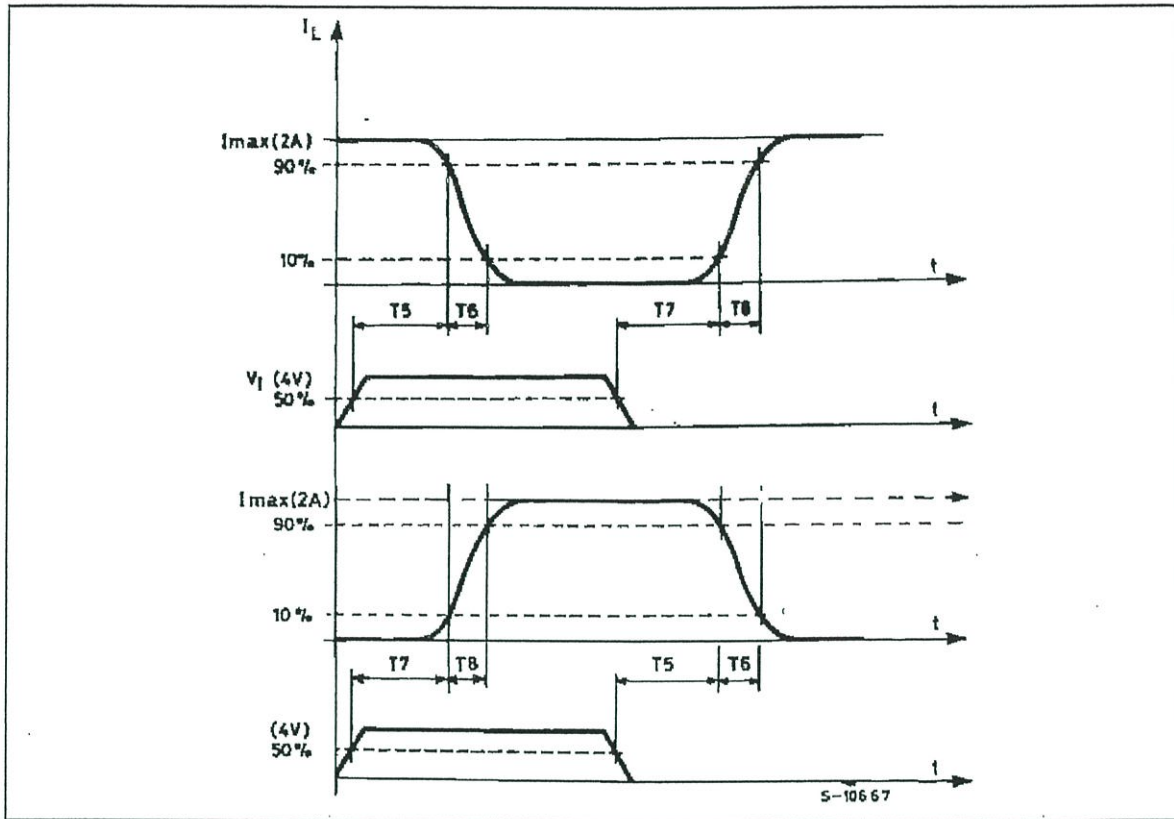


Figure 6 : Bidirectional DC Motor Control.

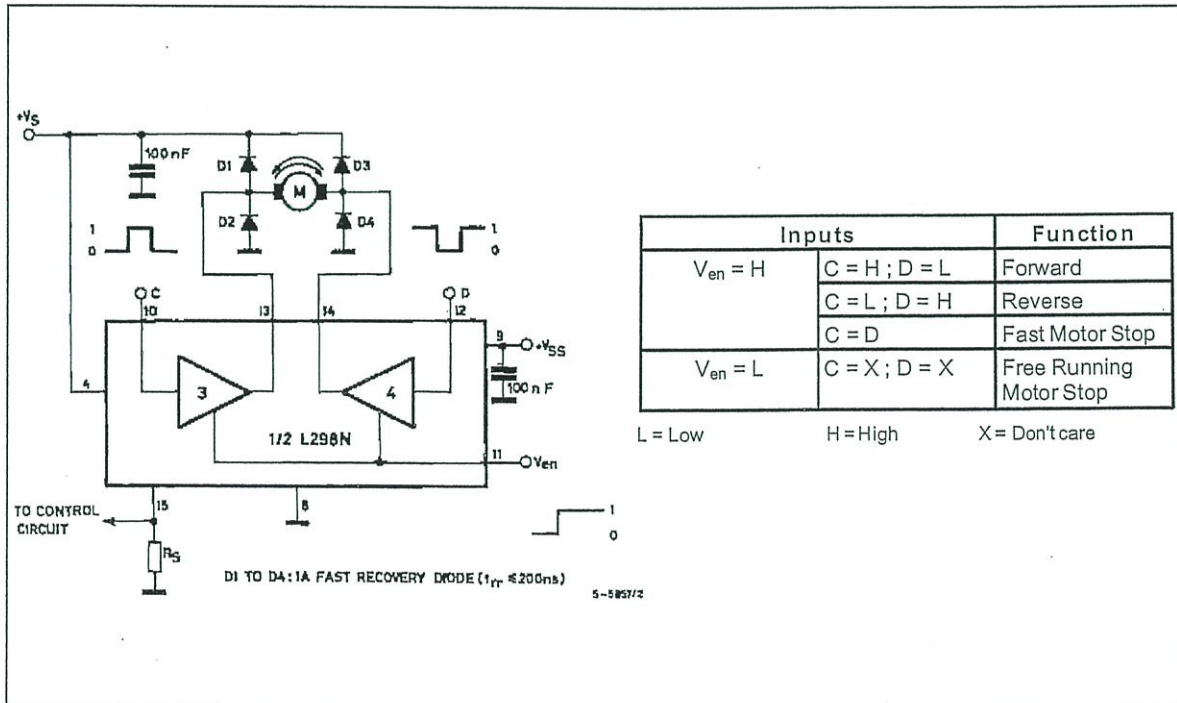
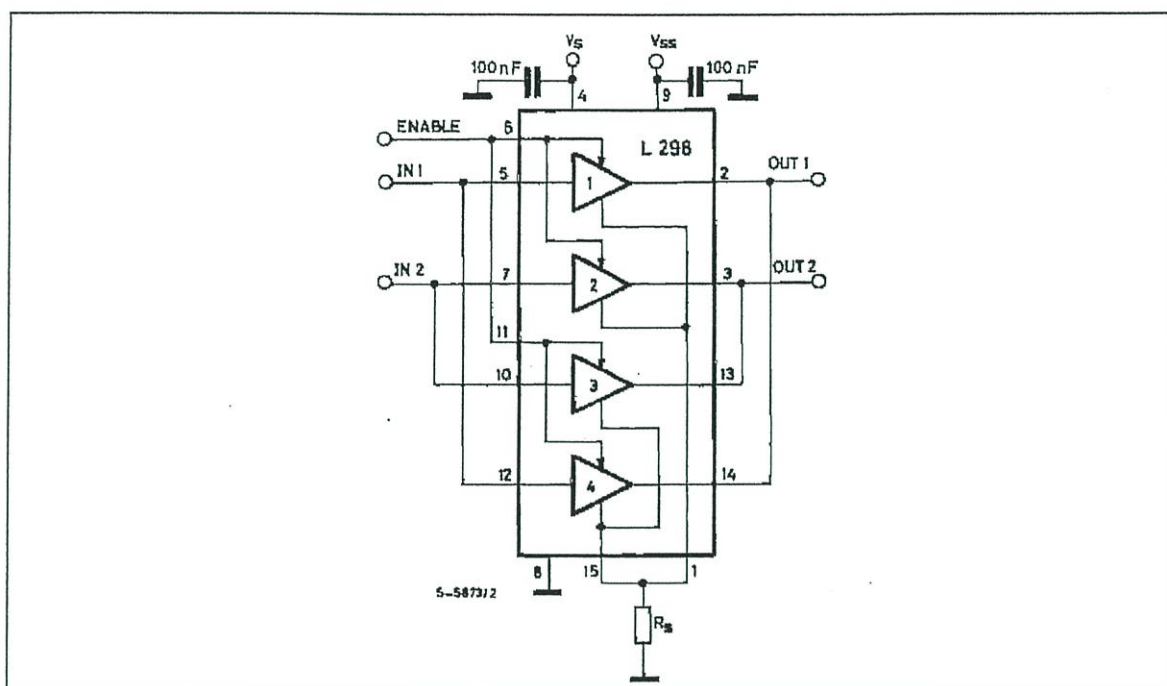


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are IN_1 ; IN_2 ; EN_A and IN_3 ; IN_4 ; EN_B . The IN inputs set the bridge state when The EN input is high; a low state of the EN input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off: Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{tr} \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

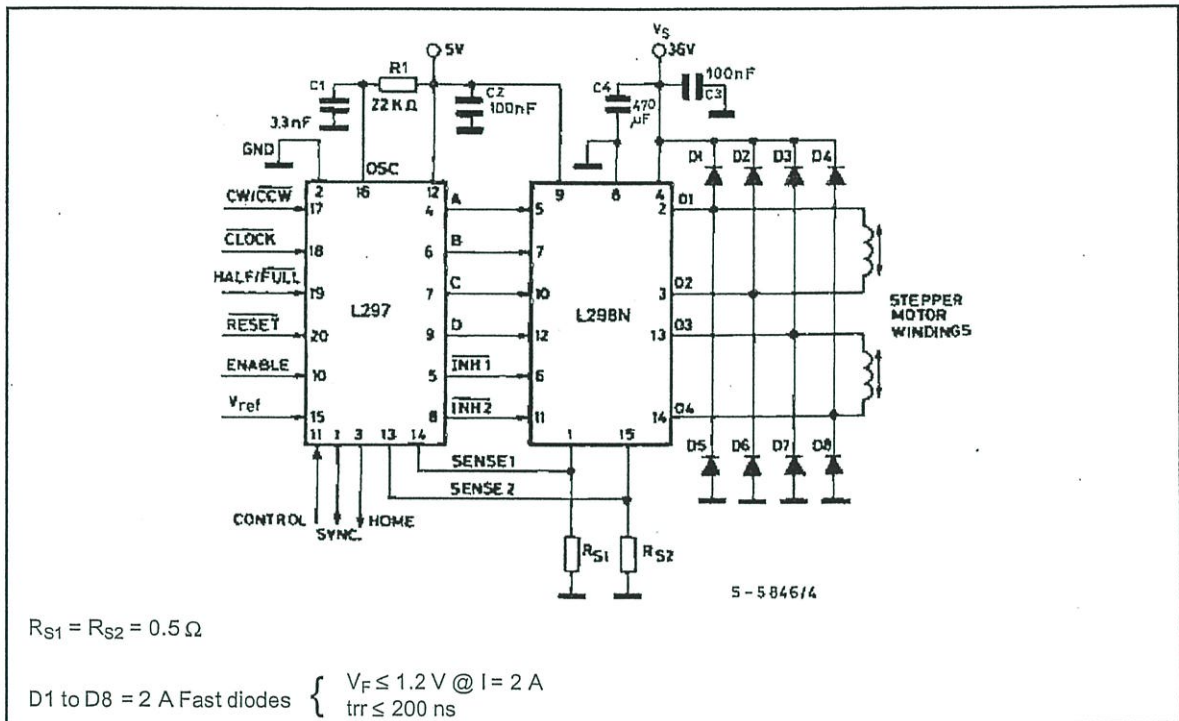


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

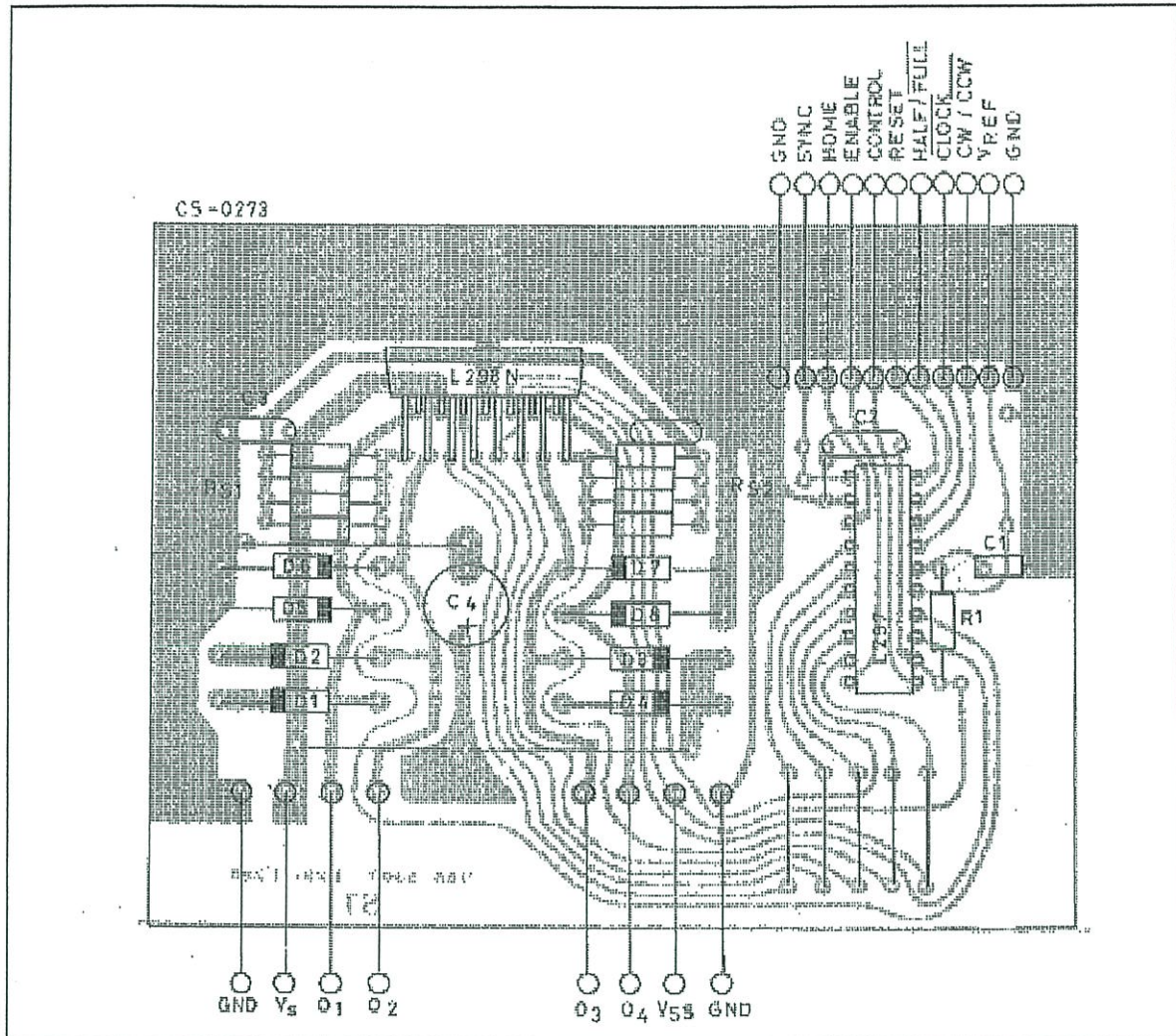
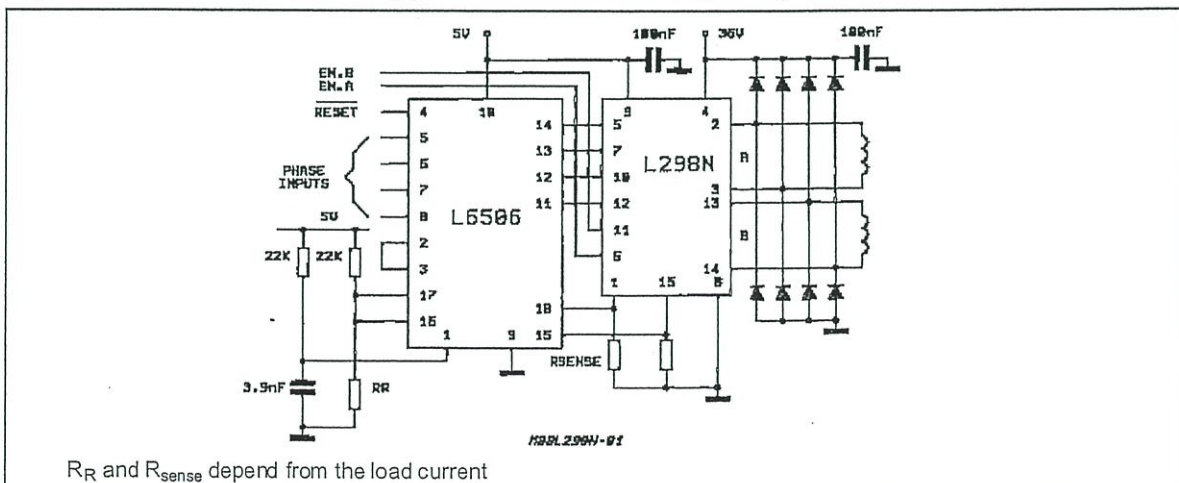
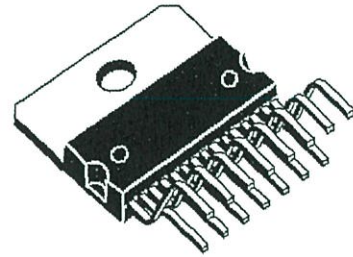


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

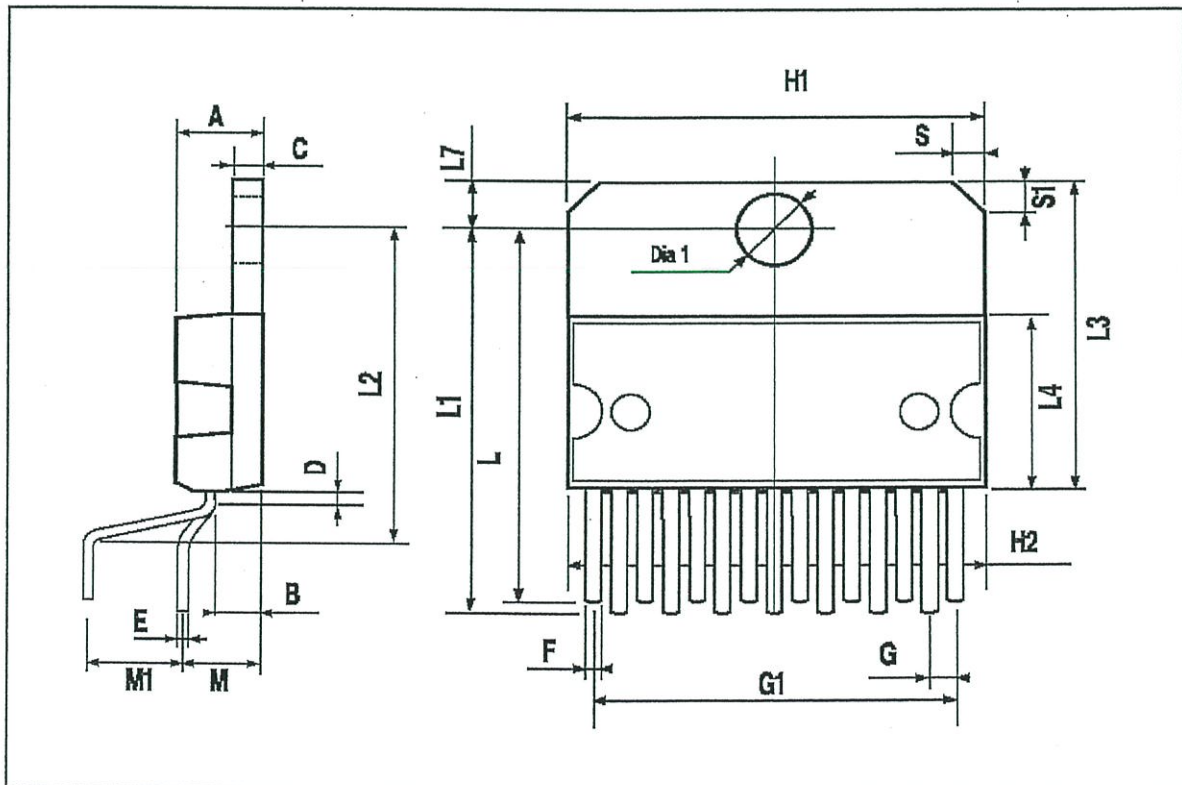


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

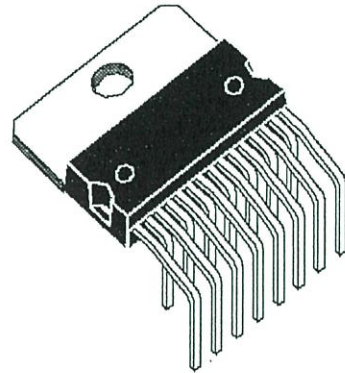


Multiwatt15 V

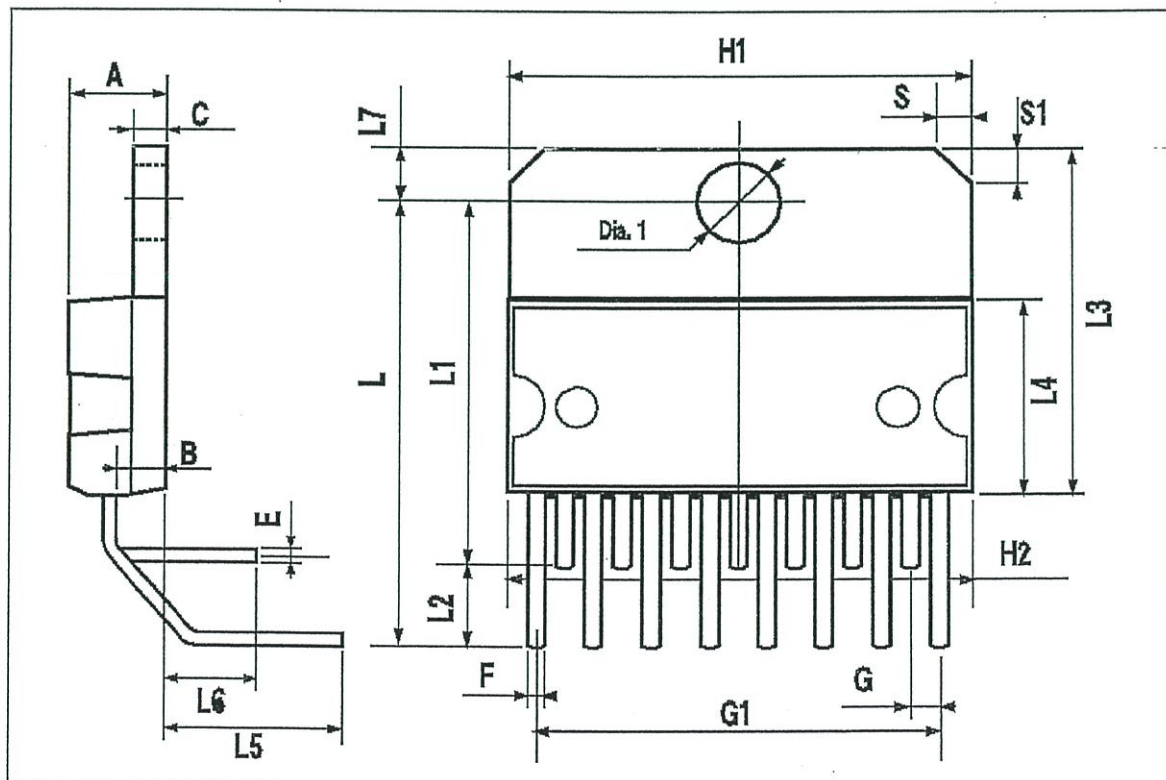


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



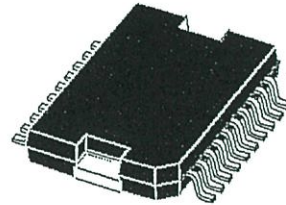
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20

