



หุ่นยนต์ไต่ผนังลบกระดานไวท์บอร์ด

SMOOTH WALL CLIMBING ROBOT FOR CLEANING WHITEBOARD

นาย เหล็กกล้า ศรีพิจิตร

นาย วิสรุช แสงมณี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

พ.ศ. 2555

หุ่นยนต์ใต้น้ำลงกระดานไวท์บอร์ด

นาย เหล็กกล้า ศรีพิจิตร

นาย วิสรุช แสงมณี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

พ.ศ. 2555

SMOOTH WALL CLIMBING ROBOT FOR CLEANING WHITEBOARD

MR. WITSARUT SANGMANEE

MR. LEKKLA SRIPIJIT

THIS PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE BACHELOR DEGREE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI

YEAR 2012

หัวข้อปริญญานิพนธ์    หุ่นยนต์ใต้น้ำเรียงลบกระดานขาว  
นักศึกษา                    นายเหล็กกล้า ศรีพิจิตร  
                                  นายวิสรุช    แสงมณี  
อาจารย์ที่ปรึกษา        อาจารย์เจษฎา อรุณฤกษ์

---

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคล  
ธัญบุรี อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

.....หัวหน้าภาควิชาฯ  
(อาจารย์มาโนช ประชา)

**คณะกรรมการสอบปริญญานิพนธ์**

.....ประธานกรรมการ  
(ดร.สุทินัน พรอนุรักษ์)

.....กรรมการ  
(อาจารย์สมชาติ หรั่งเจริญ)

.....กรรมการ  
(อาจารย์วีระชัย เข้มมาจี)

.....กรรมการและอาจารย์ที่ปรึกษา  
(อาจารย์เจษฎา อรุณฤกษ์)

ลิขสิทธิ์ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

หัวข้อปริญาานิพนธ์	หุ่นยนต์ไต่ผนังลบกกระดานไวท์บอร์ด	
นักศึกษา	นายเหล็กกล้า ศรีพิจิตร	รหัส 115110462043-0
	นายวิสรุช แสงมณี	รหัส 115110462016-6
อาจารย์ที่ปรึกษา	อาจารย์เจษฎา อรุณฤกษ์	
ปีการศึกษา	2555	

### บทคัดย่อ

เนื่องจากการสำรวจในพื้นที่สูงและเป็นผนังเรียบเหนือจากพื้นราบต่ำกว่า  $90^\circ$  มีความยากลำบากเพราะเป็นจุดที่มนุษย์เข้าไปได้ แต่อาจเกิดความเลี้ยสูงอีกทั้งอุปกรณ์ที่สามารถเคลื่อนที่และยึดเกาะพื้นผิวได้ดียังมีน้อยและยังต้องเคลื่อนที่หนีแรงดึงดูดของโลกอีกด้วย ทางผู้จัดทำจึงได้มีความคิดที่จะทำหุ่นยนต์ไต่ผนังเรียบที่สามารถเคลื่อนที่ตามที่เรากำหนดไปยังจุดต่างๆได้ซึ่งสามารถนำหุ่นยนต์ไต่กำแพงไปใช้ในงานสำรวจจริงได้อย่างรูปธรรม เพื่อเป็นความรู้แก่ผู้จัดทำและผู้ที่เกี่ยวข้อง วัตถุประสงค์ในการทำหุ่นไต่ผนังคือ

1. ออกแบบและพัฒนาหุ่นยนต์ไต่ผนัง
2. ศึกษาการทำงานและการใช้งานของไมโครคอนโทรลเลอร์
3. ศึกษาการทำงานของตัวควบคุมกระแสไหลวน
4. เพื่อนำไปประยุกต์ใช้งานจริง

หุ่นยนต์ไต่ผนังสามารถเคลื่อนที่บนผนังกระจกในแนวตั้ง โดยที่หุ่นยนต์ใช้ระบบกระแสไหลวนและระบบสูญญากาศจับและยึดตัวหุ่นยนต์ลงบนพื้นผิว หุ่นยนต์ใช้มอเตอร์สองตัวเพื่อควบคุมการเคลื่อนที่ในทิศทางขึ้น, ลง, ซ้ายและขวา โดยใช้ไมโครคอนโทรลเลอร์ตระกูล PIC เป็นตัวใช้ควบคุมการทำงานของมอเตอร์ในการเคลื่อนที่ในขั้นตอนต่างๆบนผนัง ผู้ใช้สามารถควบคุมหุ่นยนต์นี้ผ่านส่วนติดต่อผู้ใช้บนคอมพิวเตอร์โดยระบบการสื่อสารไร้สายแบบ Wireless และหุ่นยนต์สามารถทำงานอัตโนมัติได้

## กิตติกรรมประกาศ

ขอขอบคุณอาจารย์เจษฎา อรุณฤกษ์ ที่เป็นอาจารย์ปรึกษาโปรเจกของกลุ่มข้าพเจ้าซึ่งคอยแนะนำเกี่ยวกับการออกแบบหุ่น การวางแผนงาน ตรวจสอบงานว่าเป็นไปตามที่ต้องการหรือไม่ และยังให้คำแนะนำเกี่ยวกับอุปกรณ์ที่ควรจะใช้อีกด้วย ขอขอบพระคุณเป็นอย่างมาก แล้วที่จะขาดไม่ได้เลยต้องขอขอบพระคุณ คุณนิรุต กวีวัจน์ ตำแหน่งนักวิชาการ 5 และวิศวกรประจำกองนิทรรศการ องค์การพิพิธภัณฑ์วิทยาศาสตร์แห่งชาติ ที่ช่วยในเรื่องสถานที่ในการทำโปรเจกและคุณอภิชาติ ขำสวัสดิ์ ผู้ช่วยนักวิชาการและเจ้าหน้าที่ประจำห้องอิเล็กทรอนิกส์ ที่ช่วยในการให้ยืมอุปกรณ์ใช้สำหรับทำโปรเจก ขอขอบคุณอาจารย์วีระชัย เข้มวจิ ที่เป็นคนคอยเสนอแนวทางของโครงการและตรวจสอบเอกสารงานอีกทั้งยังคอยดูความคืบหน้าของโครงการและการจัดซื้อการเสนอโปรเจก เพื่อให้ทราบจุดที่ควรแก้ไข สุดท้ายขอขอบคุณ เพื่อนๆทุกคนที่คอยมาพูดคุยโปรเจก แลกเปลี่ยนความรู้เกี่ยวกับโปรเจกว่าควรไปในทิศทางใดและยังคอยให้กำลังใจในการทำโครงการ ขอขอบคุณไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญตาราง	ช
สารบัญรูป	ฅ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์	
1.3 ขอบเขต	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	1
บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง	2
2.1 ระบบสื่อสารไร้สาย	2
2.2 เซอร์โวมอเตอร์ (Servo Motor )	3
2.3 ระบบสัญญาณภาค	10
2.4 อะคริลิก	12
2.5 XBee	15
2.6 เซ็นเซอร์	19
2.7 แรงเสียดทาน	23
2.8 PIC	24
2.9 Visual Basic	30
2.10 Solid Work	36
2.11 ArtCAM	40

## สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีดำเนินงาน	41
3.1 แผนการดำเนินงาน	41
3.2 การออกแบบ / เครื่องมือ	42
3.3 ขั้นตอนการสร้าง / ขั้นตอนการดำเนินงาน	52
3.4 วิธีการทดสอบ / วิธีการวัดผล	53
บทที่ 4 ผลและการวิเคราะห์ผลการดำเนินงาน	54
4.1 ผลการดำเนินงาน	54
4.2 การวิเคราะห์ / การวิจารณ์	61
บทที่ 5 สรุปและข้อเสนอแนะ	63
บรรณานุกรม	66
ภาคผนวก ก Code โปรแกรม	67
ภาคผนวก ข Datasheet	82
ประวัติผู้ทำปฏิญานิพนธ์	100

## สารบัญตาราง

ตารางที่		หน้า
2.1	ตัวอย่างการใช้งานระบบสูญญากาศ	11
2.2	รายละเอียดโปรแกรม	32
3.1	แผนงานการดำเนินงาน	41
4.1	ตารางแสดงผลการถ่วงน้ำหนักกับหุ่นยนต์ใต้ผนังลบกระดาน	57
4.2	แรงคูดของพัคลมที่วัดได้	61

## สารบัญรูป

รูปที่		หน้า
2.1	ระบบการสื่อสารแบบไร้สาย (Wireless LAN).	2
2.2	ส่วนประกอบของ Servo Motor	3
2.3	รายละเอียดชิ้นส่วนใน Servo Motor	4
2.4	สัญญาณพัลส์	4
2.5	องศาของ Servo Motor กับ สัญญาณพัลส์	5
2.6	การถอดชิ้นส่วน Servo Motor	6
2.7	การปรับแต่ง Tap Stop	7
2.8	การเปลี่ยนตัวต้านทานของ Servo Motor	7
2.9	ถอดชิ้นส่วนของตัวต้านทานปรับค่าออก	8
2.10	การป้อนสัญญาณที่มีขนาดความกว้างพัลส์ 1 ms หรือ ให้น้อยกว่า 1.5 ms	9
2.11	ป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 2 ms หรือ ไม่ต่ำกว่า 1.5 ms	9
2.12	การควบคุมให้มอเตอร์หยุดหมุนโดยใช้ Logic	10
2.13	แม่พิมพ์ผลิตแผ่นอะคริลิกพลาสติกเรียงซ้อนกันหลายชั้น	14
2.14	ตัวอย่างผลิตภัณฑ์อะคริลิกบางชนิดที่อยู่ใกล้ตัวมาก	15
2.15	Star Network	16
2.16	Cluster Tree Network	16
2.17	Mesh Network	17
2.18	ภาพประกอบการอธิบาย การทดลอง รับส่งข้อมูลผ่าน ระบบเครือข่ายแบบ Mesh	18
2.19	ตัวอย่างตัวตรวจจับแบบไม่สัมผัสใช้หลักการเหนี่ยวนำ	20
2.20	แสดงส่วนประกอบของตัวตรวจจับแบบเหนี่ยวนำ	21
2.21	ตัวอย่างตัวตรวจจับแบบความจุแบบต่าง ๆ	21
2.22	ภาพตัดขวางด้านข้างส่วนตรวจจับของตัวตรวจจับแบบไม่สัมผัสแบบความจุไฟฟ้า	21

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.23	Infrared Sensor	22
2.24	แสดงโครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์	25
2.25	แสดงการจัดขาของไมโครคอนโทรลเลอร์ตระกูล PIC 16F684	27
2.26	การแบ่งหน่วยความจำของ PIC	27
2.27	Chip แบบ OTP	28
2.28	Chip แบบ EPROM	28
2.29	Chip แบบ EEPROM	29
2.30	ส่วนประกอบของโปรแกรม Visual Basic 6	32
2.31	ToolBar	33
2.32	ToolBox	34
2.33	คอนโทรล ActiveX (ActiveX Controls)	34
2.34	Form Designer	35
2.35	Properties Window	35
2.36	Form Layout	36
2.37	หมวดการทำงานของโปรแกรม Solid Work	39
2.38	Interface ของ Solid Work 2007	39
2.39	Interface ของ ArtCAM	40
3.1	เครื่อง CNC ใช้กัดชิ้นงาน	42
3.2	แบบหุ่นยนต์โดย Solid Work	43
3.3	ใช้ ArtCAM ในการสั่งงานให้เครื่อง CNC กัดชิ้นงาน	43
3.4	ใบพัด 3 ใบพัด	44
3.5	ด้านล่างของหุ่น	44
3.6	PIC 16F887และ XBee	45
3.7	ไฟแสดงสถานะ	45
3.8	กล่องส่งสัญญาณ	46
3.9	กล่องสวิตชิง 5 แอมป์	46

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.10	ลือของหุ่นยนต์	47
3.11	แขนกลและที่ลบกระดาน	47
3.12	ลายปริ้นบอร์ด	48
3.13	ติดตั้งเซ็นเซอร์ไว้ที่ปลายแปลงลบกระดาน	49
3.14	Block Diagram ของหุ่นยนต์ไต้ผนัง	49
3.15	Flow Chart ของหุ่นยนต์	50
3.16	Interface ของโปรแกรมควบคุมหุ่นยนต์	51
4.1	โครงประกอบของหุ่นไต้ผนัง	54
4.2	ทดสอบการเกาะติดกำแพง	55
4.3	ทดสอบการไต่ของหุ่นยนต์ไต้ผนัง	55
4.4	ปรับแต่งลือของรถไต้ผนัง	56
4.5	การทดสอบการถ่วงน้ำหนักกับการเปลี่ยนการทำงาน	56
4.6	แขนกลทั้ง 4 ระดับ	58
4.7	ทดสอบการเปลี่ยนโหมดเมื่ออุปกรณ์ครบ	58
4.8	XBee มีไฟเข้า เข้าสู่โหมดพื้นราบ	59
4.9	โหมดไต้ผนังไฟสีแดงดวงล่างติด	59
4.10	โหมดลบกระดานไฟที่ 2 จากข้างล่างติด	59
4.11	ทดสอบการลบกระดาน	60

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

เนื่องจากการสำรวจที่สูงมีลักษณะเป็นผนังเรียบมีความชันมากกว่า 0 องศา แต่ไม่เกิน 90 องศา มีความยากลำบากแม้ว่าเป็นจุดที่มนุษย์เข้าไปถึงได้ แต่อาจเกิดความเสี่ยงสูง อีกทั้งอุปกรณ์ที่สามารถเคลื่อนที่และยึดเกาะพื้นผิวได้ดียังมีน้อย อีกทั้งยังต้องเคลื่อนที่หนีแรงดึงดูดของโลกอีกด้วย ผู้จัดทำจึงได้มีความคิดที่จะทำหุ่นยนต์ไต่ผนังเรียบที่สามารถเคลื่อนที่ตามที่เรากำหนดไปยังจุดต่างๆได้ และนำมาประยุกต์ใช้ประโยชน์จริงกับการลดกระดานขาว

### 1.2 วัตถุประสงค์

- 1.2.1 ออกแบบและพัฒนาหุ่นยนต์ไต่ผนังและนำไปประยุกต์ใช้งานได้จริง
- 1.2.2 ศึกษาการทำงานและการใช้งานของไมโครคอนโทรลเลอร์
- 1.2.3 ศึกษาการทำงานของระบบสูญญากาศ

### 1.3 ขอบเขตของโครงการ

- 1.3.1 เคลื่อนที่พื้นราบด้วยล้อและใช้เซอร์โวมอเตอร์เป็นระบบในการขับเคลื่อนล้อของหุ่นยนต์
- 1.3.2 สามารถเปลี่ยนการเดินจากพื้นราบขึ้นไปยังผนังเรียบที่มีความชันไม่เกิน 90°
- 1.3.3 ใช้ระบบ Zero Gravity Traction Technology ในการเคลื่อนที่บนผนัง
- 1.3.4 ใช้ระบบการควบคุมด้วยคอมพิวเตอร์ผ่านการสื่อสารไร้สาย
- 1.3.5 หุ่นยนต์สามารถไต่บนกระดานขาวและลบกระดานขาวได้ด้วยระบบควบคุมด้วยมือ
- 1.3.6 หุ่นยนต์สามารถไต่บนกระดานขาวและลบกระดานขาวได้ด้วยระบบอัตโนมัติ
- 1.3.7 มีไฟแสดงสถานะการทำงานว่าเป็นพื้นราบหรือไต่ผนัง

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 หุ่นยนต์ที่สามารถเคลื่อนที่จากพื้นราบไปยังผนังที่มีความชันไม่เกิน 90° ได้
- 1.4.2 สามารถนำหุ่นยนต์ไปประยุกต์ใช้งานจริงได้
- 1.4.3 สร้างไว้เป็นต้นแบบเพื่อนำไปพัฒนาต่อได้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

หุ่นยนต์ใต้น้ำมีการทำงานที่ต้องมีที่สามารถเคลื่อนที่บนผนังกระจกในแนวดิ่ง โดยที่หุ่นยนต์ใช้ระบบสุญญากาศจับและยึดตัวหุ่นยนต์ลงบนพื้นผิว หุ่นยนต์ใช้มอเตอร์ประมาณอย่างน้อย 5 ตัว เพื่อควบคุมการเคลื่อนที่ในทิศทางขึ้น, ลง, ซ้าย, ขวา และแปรงลบกระดานโดยใช้ไมโครคอนโทรลเลอร์ตระกูล PIC เป็นตัวควบคุมการทำงานของหุ่นและใช้ควบคุมการทำงานของมอเตอร์ในการเคลื่อนที่ในชั้นตอนต่างๆบนผนัง ผู้ใช้สามารถควบคุมหุ่นยนต์นี้ผ่านส่วนติดต่อผู้ใช้บนคอมพิวเตอร์โดยระบบการสื่อสารไร้สาย และมีไฟสัญญาณบอกสถานะหุ่นว่าอยู่ในโหมดไหน จำเป็นต้องมีหลักการทฤษฎีที่เกี่ยวข้องดังนี้

### 2.1 ระบบสื่อสารไร้สาย

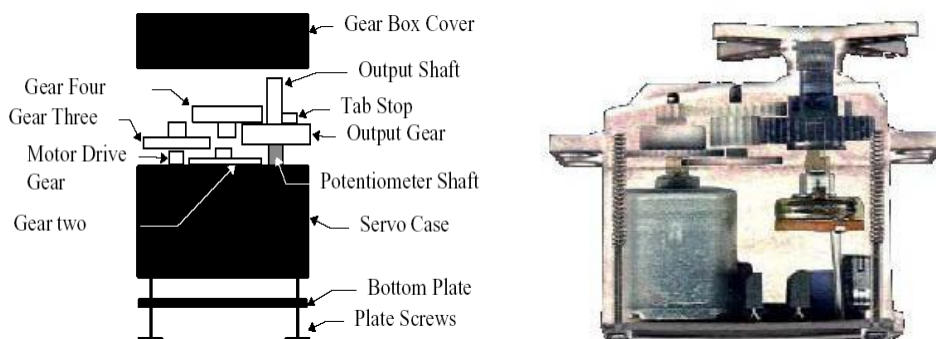
ระบบเครือข่ายไร้สาย (Wireless LAN : WLAN) หมายถึง เทคโนโลยีที่ช่วยให้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง หรือกลุ่มของเครื่องคอมพิวเตอร์สามารถสื่อสารกันได้ รวมถึงการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่ายคอมพิวเตอร์ด้วยเช่นกัน โดยปราศจากการใช้สายสัญญาณในการเชื่อมต่อ แต่จะใช้คลื่นวิทยุเป็นช่องทางการสื่อสารแทน การรับส่งข้อมูลระหว่างกันจะผ่านอากาศ ทำให้ไม่ต้องเดินสายสัญญาณ และติดตั้งใช้งานได้สะดวกขึ้นตามรูป 2.1



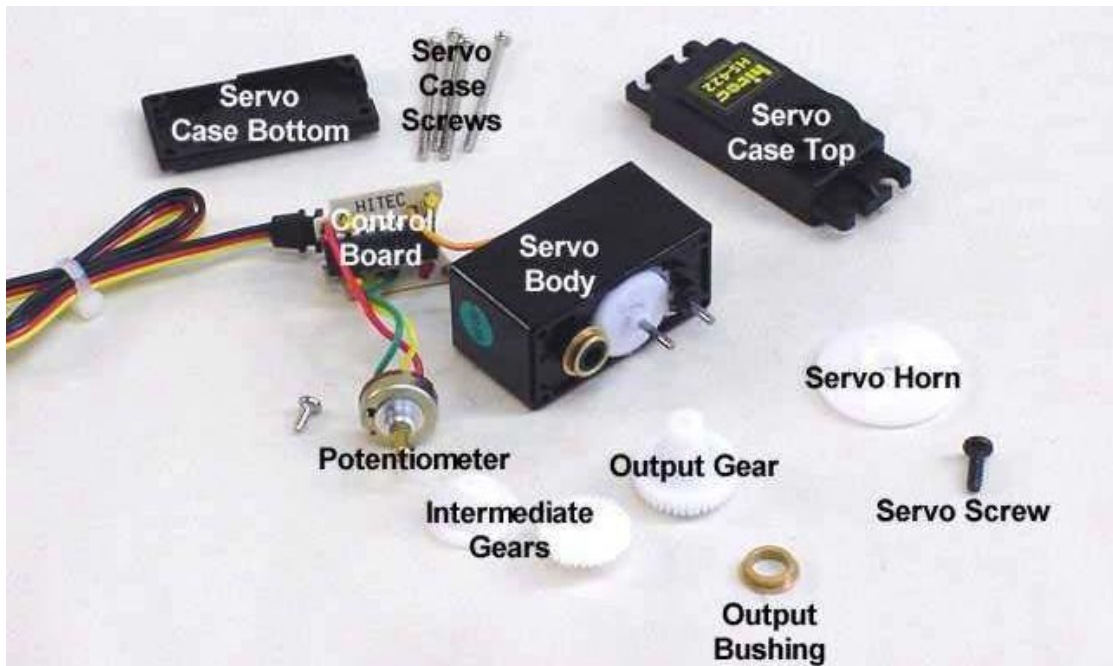
รูปที่ 2.1 ระบบการสื่อสารแบบไร้สาย (Wireless LAN)

## 2.2 เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ (Servo Motor) คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับชุดเกียร์และส่วนควบคุมต่างๆไว้ในโมดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC , GND และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วิดมอด (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลต์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือจะมีขนาดเล็กน้ำหนักเบา, ให้แรงบิดสูง, กินพลังงานน้อยและสามารถควบคุมด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่นๆ เพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่งหรือทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180° หรือครึ่งรอบเท่านั้น หรือบางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์และตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้นเซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา หรือครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้ แต่ถ้าหากเราต้องการให้มอเตอร์หมุนเป็นวงรอบ (360°) นั้นก็สามารถทำได้ โดยจะต้องทำการปรับแต่ง (Modify) ดัดแปลงชิ้นส่วนบางอย่างของมอเตอร์ ซึ่งวิธีการต่างๆจะได้กล่าวไว้ในภายหลังส่วนประกอบต่างๆ



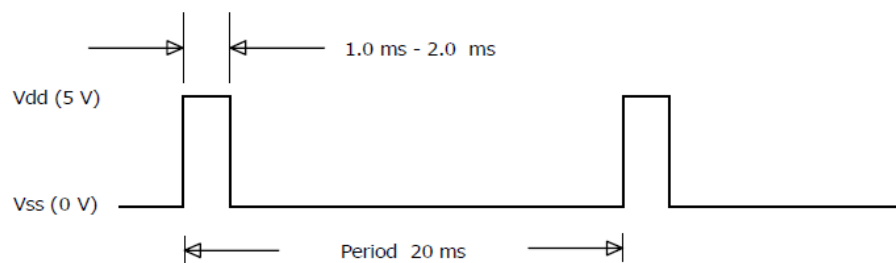
รูปที่ 2.2 ส่วนประกอบของ Servo Motor



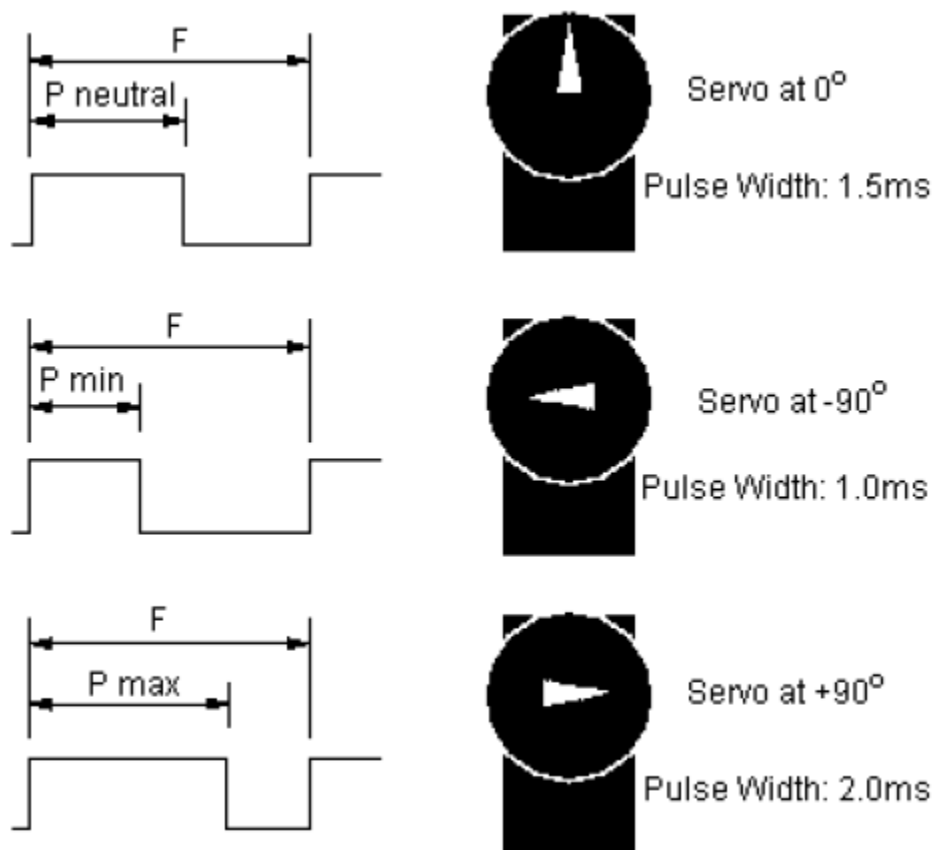
รูปที่ 2.3 รายละเอียดชิ้นส่วนใน Servo Motor

### 2.2.1 หลักการทำงานของ Servo Motor

การควบคุมการทำงานของเซอร์โวมอเตอร์ ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูป 2.4



รูปที่ 2.4 สัญญาณพัลส์



รูปที่ 2.5 องศาของ Servo Motor กับ สัญญาณพัลส์

จากรูป 2.5 จะพบว่า

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่น ๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงจากจุดทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่นถ้าต้องการให้มอเตอร์หมุนไปที่มุม 45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้นและสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้ โดยหลักการก็คือจะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายให้กับมอเตอร์ทาง

ขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงไปเป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของวงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control Line) มอเตอร์จึงจะหยุดหมุน

### 2.2.3 การปรับแต่ง Servo Motor

จากคุณสมบัติของ Servo Motor ที่ผลิตออกมาจากโรงงานจะสามารถหมุนได้แค่เพียงประมาณ 180 องศาหรือประมาณครึ่งรอบเท่านั้น หากเราต้องการนำเอา Servo Motor ไปใช้งานในลักษณะที่หมุนเป็นวงรอบนั้นก็สามารทำได้แต่ก็จะสูญเสียการควบคุมในเรื่องของการสั่งให้มอเตอร์หมุนไปในตำแหน่งหรือมุมที่ต้องการไปด้วยจะทำให้ได้ก็เพียงในเรื่องของการสั่งให้หมุนซ้าย, ขวาและหยุดเท่านั้น โดยการทำให้มอเตอร์สามารถหมุนเป็นวงรอบได้นั้นจะต้องทำการปรับแต่งหรือแก้ไขโครงสร้างภายในบางส่วนของมอเตอร์ซึ่งได้แก่

- 1) การต่อตัวต้านทานคงที่ 2 ตัวอนุกรม แทนตัวต้านทานปรับค่าได้
- 2) ตัดชิ้นส่วนของแกนเฟืองที่ทำหน้าที่หยุดมอเตอร์ (TAB STOP) ออก
- 3) การตัดแปลงตัวต้านทานปรับค่าได้ (VR) ให้สามารถหมุนได้รอบทิศทาง

### 2.2.4 ขั้นตอนการปรับแต่ง Servo Motor

- 1) ถอดชิ้นส่วนของ Servo Motor ออกเป็นส่วนๆ (ตามรูป 2.6)



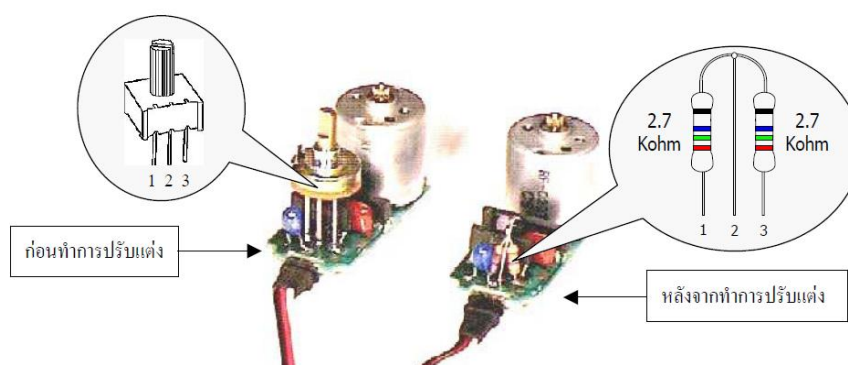
รูปที่ 2.6 การถอดชิ้นส่วน Servo motor

2) ตัดแกนที่ติดกับเฟือง (TAB STOP) ออกโดยแกนนี้มีหน้าที่ป้องกันไม่ให้มอเตอร์หมุนเกินมุม 180 องศาทั้งนี้เพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับตัวต้านทานปรับค่าได้เนื่องจากตัวต้านทานชนิดปรับค่าได้ไม่สามารถหมุนเป็นวงรอบได้ดังนั้นเพื่อให้มอเตอร์หมุนเป็นวงรอบได้จึงต้องตัด TAB STOP ในส่วนนี้ออก



รูปที่ 2.7 การปรับแต่ง TAB STOP

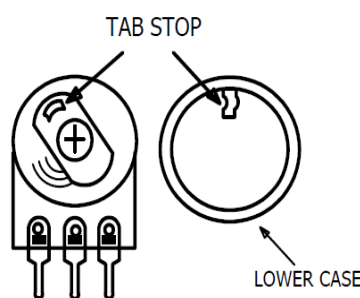
3) ถอดตัวต้านทานปรับค่าได้ (VR) ออกแล้วใส่ตัวต้านทานชนิดค่าคงที่ 2 ตัวที่ต่ออนุกรมกันเข้าไปแทนในตำแหน่งของตัวต้านทานปรับค่าได้ โดยตัวต้านทานชนิดค่าคงที่ที่นำมาต่อนี้จะต้องมีค่าอยู่ในช่วง  $2.2\text{ k}\Omega$  ถึง  $3.3\text{ k}\Omega$  ทั้งนี้เนื่องจากตัวต้านทานชนิดปรับค่าได้ที่อยู่ในบอร์ดควบคุมของ Servo Motor นั้นจะมีค่าความต้านทาน  $5\text{ k}\Omega$  ดังนั้นจึงต้องนำตัวต้านทานค่าคงที่ที่มาต่ออนุกรมกันเพื่อให้ได้ค่าความต้านทานใกล้เคียงกับของเดิม (รูป 2.7)



รูปที่ 2.8 การเปลี่ยนตัวต้านทานของ Servo Motor

4) ถึงแม้ว่าเราจะถอดตัวด้านทานปรับค่า (VR) ออกจากวงจรแล้วก็ตาม แต่เนื่องจากเรายังคงต้องใช้ตัวด้านทานปรับค่าได้นี้ไปเป็นแกนหมุนของมอเตอร์อยู่ ซึ่งตัวด้านทานปรับค่านี้ จะไม่สามารถหมุนเป็นวงรอบได้ ทำให้เราต้องแก้ไขเปลี่ยนแปลงบางส่วนของตัวด้านทานเพื่อให้ตัวด้านทานสามารถหมุนรอบตัวเองได้ เพื่อที่จะได้ไม่ไปขัดขวางการหมุนของมอเตอร์ซึ่งทำได้โดย

- ถอดชิ้นส่วนของตัวด้านทานปรับค่าออก



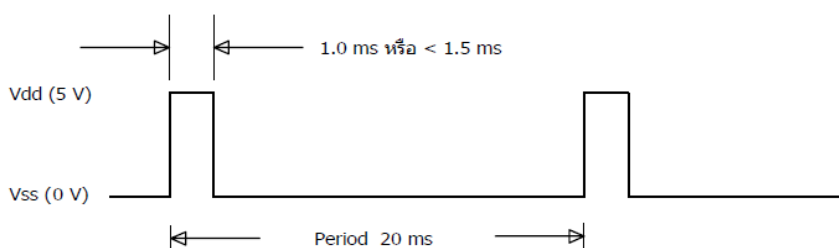
รูปที่ 2.9 ถอดชิ้นส่วนของตัวด้านทานปรับค่าออก

- ตัวด้านทานปรับค่าในมอเตอร์แต่ละรุ่นนั้นอาจจะใช้ไม่เหมือนกัน แต่จะมีหลักการเดียวกันโดยจะมีแท็บที่ทำหน้าที่หยุดการหมุนของตัวด้านทานอยู่ ให้เราทำการตัดส่วนนี้ออกแล้วทดลองหมุนแกนของตัวด้านทานปรับค่า ถ้าสามารถหมุนรอบตัวเองได้ก็ทำการประกอบตัวด้านทานเข้าไปเหมือนเดิม แต่ถ้ายังหมุนเป็นวงรอบไม่ได้ก็ให้พิจารณาว่ามีชิ้นส่วนใดที่ยังขัดขวางการหมุนของตัวด้านทานอยู่ เมื่อพบก็ให้เอาออกหรือทำลายได้เลยโดยไม่ต้องสนใจว่าจะทำให้ตัวด้านทานนี้พังเพราะเราไม่ได้ใช้ประโยชน์จากการเปลี่ยนแปลงค่าความต้านทานนี้อีกแล้ว นอกจากใช้เป็นแกนหมุนของเฟืองเท่านั้น
- จากนั้นตัดหรือพับขาของตัวด้านทานปรับค่า (VR) เพื่อป้องกันไม่ให้ขาของตัวด้านทานดังกล่าวไปช้อตกับแผงวงจรควบคุม

5) ประกอบชิ้นส่วนต่างๆเข้าที่เดิมและเพื่อความปลอดภัยในการประกอบตัวด้านทานปรับค่า (VR) ลงในกล่องของ Servo Motor ควรหาฉนวนรองตรงส่วนของขาที่เป็นโลหะของตัวด้านทานด้วยเพื่อไม่ให้ไปช้อตกับส่วนอื่นๆในแผงวงจรควบคุม เพียงเท่านี้มอเตอร์ของเราก็จะสามารถหมุนเป็นวงรอบ 360 องศาได้แล้วและในการนำไปใช้งานจะต้องระวังเรื่องขั้วของโพลที่นำมาต่อกับมอเตอร์ เพราะหากนำมอเตอร์ไปขับหรือยกโพลที่มีน้ำหนักมากเกินไป อาจจะทำให้

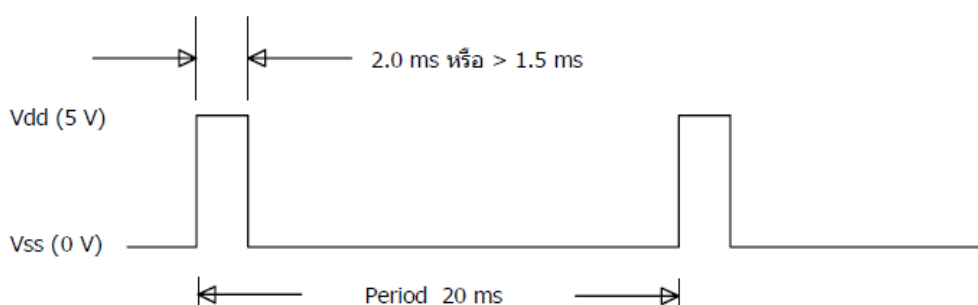
เกิดความเสียหายกับเฟือง หรือเกียร์ต่างๆของมอเตอร์ได้หลังจากเราได้ทำการปรับแต่งการทำงานของเซอร์โวมอเตอร์ให้สามารถหมุนเป็นวงรอบได้แล้ว วิธีในการควบคุมให้มอเตอร์หมุนจะมีลักษณะดังนี้

- การควบคุมให้มอเตอร์หมุนทางด้านซ้ายจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 1 ms หรือให้น้อยกว่า 1.5 ms โดยจะต้องป้อนสัญญาณพัลส์นี้ทุกๆ 20 ms (ในช่วงประมาณ 20 ms – 30 ms)



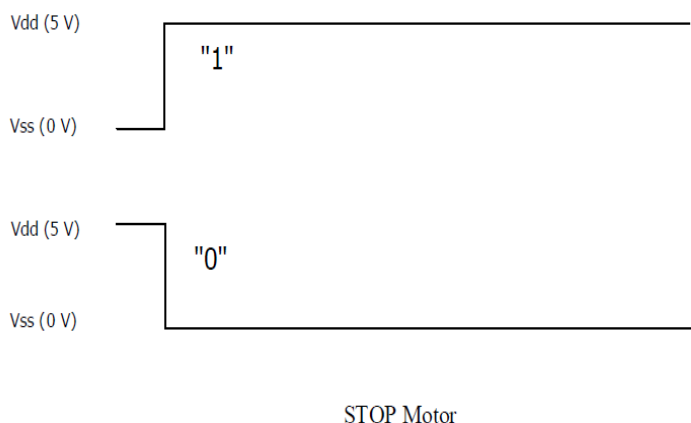
รูปที่ 2.10 การป้อนสัญญาณที่มีขนาดความกว้างพัลส์ 1 ms หรือ น้อยกว่า 1.5 ms

- การควบคุมให้มอเตอร์หมุนทางด้านขวาจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 2 ms หรือไม่ต่ำกว่า 1.5 ms และจะต้องป้อนสัญญาณพัลส์นี้ทุกๆ 20 ms (หรือในช่วงประมาณ 20 ms – 30 ms) เช่นกัน



รูปที่ 2.11 ป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 2 ms หรือ ไม่ต่ำกว่า 1.5 ms

- การควบคุมให้มอเตอร์หยุดหมุน ทำได้โดยการส่งลอจิก “0” หรือ “1” ให้กับมอเตอร์หรือก็คือการไม่จ่ายสัญญาณพัลส์ให้กับมอเตอร์นั่นเอง



รูปที่ 2.12 การควบคุมให้มอเตอร์หยุดหมุนโดยใช้ Logic

### 2.3 ระบบสุญญากาศ

คำว่าสุญญากาศ “Vacuum” มาจากภาษาละตินคำว่า “Vacua” ซึ่งมีความหมายว่า ที่ว่าง (empty) แต่ในทางปฏิบัติแล้วคำว่า Vacuum คือที่ซึ่งหรือบริเวณซึ่งว่างบางส่วน (Partially Empty Space) หรือบริเวณที่โมเลกุลของอากาศถูกเคลื่อนย้ายออกไป ทำให้บริเวณนั้นมีความดันต่ำกว่าความดันปกติ ที่ 760 มิลลิเมตรปรอท (760 mmHg) บริเวณที่โมเลกุลของก๊าซได้ถูกเคลื่อนย้ายออกไปโดยปกติเราเรียกว่า Work Chamber และ Work Chamber ทำหน้าที่แยกสุญญากาศ ภายในออกจากโลกภายนอกโดยปกติเราเห็นว่าโมเลกุลของก๊าซภายในมีแรงกระทำต่อผิวภายในภาชนะหรือ Work Chamber แรงนี้ทำให้เกิดความดัน เราสามารถวัดความดันใน Chamber ได้โดยการเทียบกับความดันบรรยากาศภายนอก วิธีเช่นนี้เราสามารถบอกได้ว่าโมเลกุลของก๊าซเหลือประมาณเท่าใดใน Vacuum Chamber ที่ความดันปกติ  $1.013 \times 10^5$  Pa หรือ 760 mmHg อากาศ 1 ลูกบาศก์เซนติเมตรจะมีประมาณ  $2 \times 10^{19}$  โมเลกุลและที่ความดัน  $1.013 \times 10^{-7}$  Pa อากาศ 1 ลูกบาศก์เซนติเมตร จะมีประมาณ  $2 \times 10^7$  โมเลกุล ความเป็นสุญญากาศมีระดับความแตกต่างแบ่งได้คร่าวๆ 3 แบบ คือ สุญญากาศหยาบ (Rough Vacuum) สุญญากาศสูง (High Vacuum) และสุญญากาศสูงสุด (Ultra High Vacuum) การเลือกใช้ชนิดหรือระดับของสุญญากาศขึ้นอยู่กับการใช้งาน แสดงให้เห็นว่ายังเป็นสุญญากาศดีก็ยังมีโมเลกุลของอากาศและก๊าซน้อย เราใช้สุญญากาศเมื่อเราจำเป็นต้องใช้ chamber ที่สะอาดปราศจากโมเลกุลของก๊าซที่รบกวนสิ่งที่เราจะทำงาน เช่นการผลิต การวิจัย ตัวอย่างเช่น เรามีเหล็กทิ้งไว้ในอากาศ เหล็กนี้จะทำปฏิกิริยากับ โมเลกุลของก๊าซเป็นผลให้เหล็กเกิด

สนิม แต่ถ้าเหล็กนี้อยู่ในสุญญากาศจะไม่เกิดสนิม อีกตัวอย่างหนึ่งเช่น หลอดภาพของโทรทัศน์ ถ้าหลอดภาพไม่เป็นสุญญากาศอิเล็กตรอนจะชนกับโมเลกุลของก๊าซจะทำให้อิเล็กตรอนสูญเสียพลังงานไม่สามารถเคลื่อนที่ถึงฉาก (หน้าจอ) ทำให้ไม่มีภาพวิธีที่ง่ายที่สุดที่จะให้นิยามที่สะอาด (clean) คือว่าทุกสิ่งมีสิ่งปนเปื้อนหรือสกปรกในบางระดับ ไม่ว่าสิ่งปนเปื้อนจะมากหรือน้อย ถ้าหากปนเปื้อนน้อย (Less Contamination) ก็หมายถึงสะอาดมาก (Cleaner) เท่านั้น

พิสัยของความดัน

พิสัยของความดันในทางปฏิบัติแบ่งออกเป็น 3 ช่วงง่ายๆ ดังนี้

- 1) สุญญากาศอย่างหยาบ (Rough/Low Vacuum) 759 ถึง 10 - 3 torr
- 2) สุญญากาศสูง (High Vacuum) 10<sup>-3</sup> ถึง 10<sup>-8</sup> torr
- 3) สุญญากาศสูงมาก(Ultra High Vacuum) น้อยกว่า 10<sup>-8</sup> torr

ตาราง 2.1 ตัวอย่างการใช้งานระบบสุญญากาศ

Rough Vacuum	High Vacuum	Ultrahigh Vacuum
1. ขบวนการผลิตอาหาร (Food processing)	1. การผลิตท่อ (Tube processing)	1. การวิจัยด้านอวกาศ (Space research)
2. การระเหยเป็นไอ (Evaporation)	2. การบำบัดความร้อน (Heat - treating)	2. การวิจัยทางวัสดุศาสตร์ (Materials research)
3. การทำให้แห้งขณะเย็นจัด (Freeze drying)	3. การผลิตวงจรรวม (Integrated circuit manufacturing)	3. โลหะวิทยา (Metallurgy)
4. การกลั่น (Distillation)	4. การเคลือบเพื่อความสวยงาม (Decorative coating)	4. การวิจัยทางฟิสิกส์ (Physics research)
5. การสับเตอริง (Sputtering)	5. เครื่องเร่งอนุภาค (Particle - acceleration)	5. การวิเคราะห์พื้นผิว (Surface analysis)
6. การนำไฟฟ้าของก๊าซ (Electrical conduction (neon lights))	6. การวิจัยทางเคมี (Chemistry research)	6. การปลูกฟิล์มบางโดย (MBE Molecular Beam Epitaxy)

## 2.4 อะคริลิก

บางคนเรียก อะคริลิกพลาสติก (Acrylic Plastic) หรือพลาสติกอะคริลิก ขณะที่อีกหลายคนเรียกว่า กระจกอะคริลิก (Acrylic glass) หรือเรียกย่อๆ แค่แผ่นอะคริลิก แต่ไม่ว่าจะเรียกชื่อแตกต่างกันอย่างไรก็ตาม ทั้งหมดก็หมายถึงพลาสติกชนิดเดียวกันคือ โพลีเมทิลเมทาไครเลต หรือพีเอ็มเอ็มเอ (Polymethyl Methacrylate), PMMA) และสูตรเคมีของพลาสติกชนิดนี้คือ  $C_5H_8O_2$  อะคริลิกพลาสติกหรือโพลีเมทิลเมทาไครเลตเป็นเทอร์โมพลาสติกชนิดหนึ่ง มีชื่อทางการค้าหลายชื่อด้วยกัน เช่น Plexiglas, Lucite, Perspex เป็นต้น พลาสติกชนิดนี้ถูกนำมาประยุกต์ใช้งานหลายอย่าง เช่น กระจกใสบนเครื่องบิน ป้ายโฆษณา กระจกตู้ปลา วัสดุทางการแพทย์ เป็นต้น เนื่องจากวัสดุมีสมบัติโดดเด่นในเรื่องความเหนียว (Toughness) ความโปร่งใส (Transparent) สามารถขึ้นรูปได้ง่าย และเมื่อผนวกกับการมีความหนาแน่นต่ำซึ่งเป็นสมบัติประจำตัวของวัสดุประเภทพลาสติกแล้ว อะคริลิกพลาสติกจึงเป็นวัสดุชนิดหนึ่งที่นิยมนำมาใช้แทนแก้วในงานหลายอย่าง

### 2.4.1 การพัฒนาอะคริลิก

อะคริลิกพลาสติกเป็นพลาสติกที่ได้จากการนำโมโนเมอร์ของเมทิลเมทาไครเลต (Methyl Methacrylate, MMA) มาทำปฏิกิริยาการเกิดโพลิเมอร์ (Polymerization) โดย 2 นักเคมีชาวเยอรมันคือ ฟิททิจ (Fittig) และพอล (Paul) สามารถสังเคราะห์โพลีเมทิลเมทาไครเลตได้ตั้งแต่ปี ค.ศ. 1877 แล้ว แต่การพัฒนาวิธีผลิตให้ได้แผ่นอะคริลิกพลาสติกออกมาต้องรอถึงปี ค.ศ. 1933 เมื่อออกโท เริห์ม (Otto Rohm) นักเคมีชาวเยอรมันของดิสทริบิวต์ริวิตผลิตแผ่นพลาสติกใสจากโพลีเมทิลเมทาไครเลตในชื่อทางการค้า Plexiglas หลังจากนั้นในปี ค.ศ. 1936 จึงมีการผลิตแผ่น Plexiglas ออกจำหน่ายในเชิงพาณิชย์ ด้วยความก้าวหน้าทางวิทยาศาสตร์และเทคโนโลยี ปัจจุบันสามารถผลิตอะคริลิกพลาสติกได้จากปฏิกิริยาการเกิดโพลิเมอร์หลายแบบ เช่น การเกิดโพลิเมอร์แบบอีมัลชัน (Emulsion Polymerization) การเกิดโพลิเมอร์แบบบัลค์ (Bulk Polymerization) เป็นต้น การผลิตอะคริลิกพลาสติกแบบ “แผ่น” มักใช้เทคนิคการเกิดโพลิเมอร์แบบบัลค์ โดยเติมโมโนเมอร์ของเมทิลเมทาไครเลตกับตัวเร่งปฏิกิริยาลงในแม่พิมพ์ (Mold) พร้อมกัน

### 2.4.2 สมบัติที่น่าสนใจของอะคริลิกพลาสติก

- 1) มีความหนาแน่นประมาณ 1.15-1.19 กรัม/ลูกบาศก์เซนติเมตร
- 2) มีจุดหลอมเหลวที่อุณหภูมิ 130-140 องศาเซลเซียส และจุดเดือดที่อุณหภูมิ 200 องศาเซลเซียส
- 3) มีความทนทานต่อการกระแทก (Impact Strength) สูงกว่าแก้วและโพลิสไตรีน แต่ต่ำกว่าโพลีคาร์บอเนตและพลาสติกวิศวกรรมชนิดอื่น

- 4) อะคริลิกพลาสติกมีเนื้ออ่อนจึงเกิดรอยขีดไ้ได้ง่าย
- 5) แสงสว่างสามารถส่องผ่านเนื้อพลาสติกได้ถึงร้อยละ 92 และมีการสะท้อนกลับที่ผิวประมาณร้อยละ 4
- 6) มีความทนทานต่อสภาพแวดล้อมดีกว่าพลาสติกชนิดอื่นเช่น โพลีคาร์บอเนตจึงนิยมใช้อะคริลิกพลาสติกกับงานกลางแจ้งด้วย
- 7) อะคริลิกพลาสติกไม่ทนทานต่อตัวทำละลายหลายชนิด

### 2.4.3 การผลิตแผ่นอะคริลิก

#### 1) การผลิตเป็นชุด (Batch Cell Bulk Polymerization)

- การผลิตแผ่นพลาสติกจะใช้แม่พิมพ์เป็นแผ่นแก้วหรือแผ่นโลหะผิวเรียบ 2 แผ่นประกบเข้าด้วยกัน โดยขอบนอกแม่พิมพ์มีลักษณะเป็นกรอบหน้าที่ยึดหดได้ ทั้งนี้เนื่องจากว่า ในระหว่างการเกิดโพลิเมอร์ เมื่อโมโนเมอร์หรือ โมเลกุลเล็กหลายตัวมาเชื่อมกันเข้ากลายเป็น โมเลกุลใหญ่จะทำให้ปริมาตรสารลดลง ดังนั้นกรอบแม่พิมพ์จึงต้องหดตัวตามปริมาตรพลาสติกที่เปลี่ยนไป
- โมโนเมอร์เหลวของเมทิลเมทาไครเลตกับตัวเร่งปฏิกิริยาถูกปล่อยเข้าไปในแม่พิมพ์ บางครั้งอาจเติมพรีโพลิเมอร์ (Prepolymer) ของเมทิลเมทาไครเลตเข้าไปด้วย (พรีโพลิเมอร์เป็นโพลิเมอร์น้ำหนักโมเลกุลต่ำ) เพื่อเร่งกระบวนการผลิตให้เร็วขึ้น
- เมื่อวัตถุดิบเข้าไปเต็มแม่พิมพ์แล้วปิดแม่พิมพ์ให้สนิท จากนั้นอาจมีการให้ความร้อนแก่แม่พิมพ์เพื่อกระตุ้นตัวเร่งปฏิกิริยาให้เริ่มทำงาน
- ขณะที่เกิดปฏิกิริยาจะมีการคายความร้อนออกมาจึงต้องระบายความร้อนด้วยการเป่าลม หรือแช่แม่พิมพ์ในน้ำ เพื่อควบคุมการเปลี่ยนแปลงอุณหภูมิของของเหลวในแม่พิมพ์ไม่ให้กลายเป็นไอ ซึ่งจะกลายเป็นฟองบนผิวพลาสติกในภายหลัง
- เมื่อพลาสติกแข็งตัว ผู้ผลิตจะปล่อยให้แผ่นพลาสติกเย็นก่อนจึงถอดออกจากแม่พิมพ์ ทั้งนี้อะคริลิกพลาสติกแผ่นบางจะใช้เวลาในการแข็งตัวนานประมาณ 10-12 ชั่วโมง ขณะที่พลาสติกแผ่นหนาต้องใช้เวลาหลายวันกว่าจะแข็งตัวหลังจากแกะแผ่นพลาสติกออกแล้ว แม่พิมพ์ จะถูกทำความสะอาดเพื่อเตรียมพร้อมสำหรับการผลิตชุดใหม่
- นำแผ่นอะคริลิกพลาสติกไปอบแอนนิล (Anneal) ที่อุณหภูมิประมาณ 80 องศาเซลเซียสเป็นเวลาหลายชั่วโมงเพื่อลดความเค้นตกค้าง (Residual Stress) ในพลาสติกที่อาจทำให้แผ่นพลาสติกบิดงอ หรือเสียรูปร่าง

- สูดทำยแผ่นพลาสติกจะถูกตัดแต่งครีบ หรือส่วนที่เกินออก จากนั้นปิดทับแผ่นพลาสติกด้วยกระดาษหรือฟิล์มพลาสติกเพื่อป้องกันสินค้าในระหว่างการขนส่งและการจัดเก็บ

2) การผลิตแบบต่อเนื่อง (Continuous Bulk Polymerization) มีขั้นตอนคล้ายการผลิตเป็นชุด แต่ใช้เวลาในการผลิตน้อยกว่า และใช้ผลิตแผ่นอะคริลิกพลาสติกขนาดบาง

- การผลิตแผ่นอะคริลิกพลาสติกแบบนี้ แม่พิมพ์มีลักษณะเป็นสายพานเหล็กกล้าซ้อนกัน 2 ชั้น โดยเว้นช่องห่างกันเล็กน้อยซึ่งระยะห่างระหว่างชั้นแผ่นเหล็กจะเป็นตัวควบคุมความหนาของแผ่นพลาสติก

- โมโนเมอร์เหลวกับตัวเร่งปฏิกิริยาจะถูกฉีดเข้าไปในช่องว่างระหว่างแผ่นเหล็ก และสายพานเหล็กที่บรรจุสารแล้วจะเลื่อนผ่านชุดอุปกรณ์ให้ความร้อน และระบายความร้อนสลับกันไปเพื่อให้ปฏิกิริยาเคมีเกิดอย่างสมบูรณ์

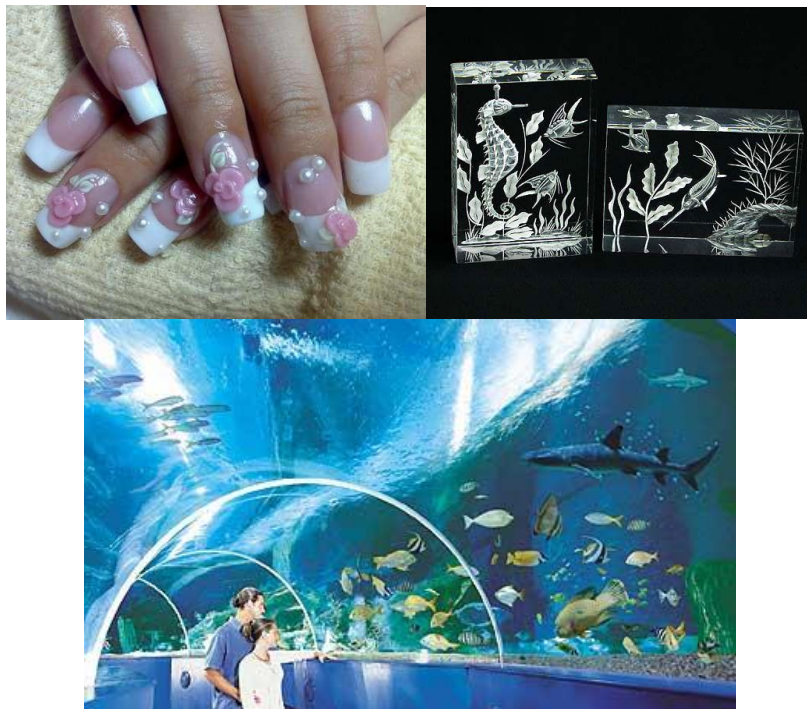
- หลังจากพลาสติกถูกนำออกจากแม่พิมพ์ จะถูกนำไปอบแอนนิลเพื่อลดความเค้นตกค้างในแผ่นพลาสติก

- แผ่นพลาสติกที่ผ่านการอบแล้วจะถูกตัดให้ได้ขนาดตามต้องการ และปิดผิวด้วยกระดาษหรือฟิล์มพลาสติกเพื่อปกป้องกันตัวสินค้าจากการขนย้าย



รูปที่ 2.13 แม่พิมพ์ผลิตแผ่นอะคริลิกพลาสติกเรียงซ้อนกันหลายชั้น

#### 2.4.4 การประยุกต์ใช้



รูปที่ 2.14 ตัวอย่างผลิตภัณฑ์อะคริลิกบางชนิดที่อยู่ใกล้ตัวมาก

## 2.5 XBee

XBee เป็นอุปกรณ์ที่ใช้ รับส่งข้อมูลแบบไร้สาย และสามารถสร้างระบบเครือข่ายขึ้นมาได้ โดย ZigBee ได้อ้างอิง มาตรฐานตาม IEEE 802.15.4 โดยสามารถแบ่ง หน้าที่การทำงานของ XBee ตามรูปแบบการสื่อสารในลักษณะต่างๆ ได้ อาทิ Coordinator , Router , End Device ในบทความนี้ ขอแนะนำการตั้งค่า Network แบบ Star , Cluster และ Mesh ด้วย XBee

### 2.5.1 XBee Topology

ในการสร้างโครงข่ายไร้สายของ ZigBee นั้น จะต้องประกอบด้วย โหนด จำนวน อย่างน้อยที่สุด 2 ชนิด คือ Coordinator node และ node ถูกข่าย ชนิดใดชนิดหนึ่ง (Router/End device) จึงจะสามารถสื่อสารและทำงานในรูปแบบของ PAN (Personal Area Network) ได้ โดย ZigBee สามารถแบ่งรูปแบบ เครือข่ายได้เป็น 3 รูปแบบ ดังนี้

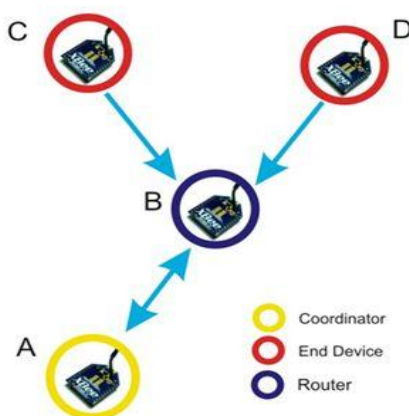
## 1) Star (Broadcast)



รูปที่ 2.15 Star Network

การเชื่อมต่อ แบบ Star หรือ แบบ Broadcast เป็นการรับส่งข้อมูลแบบไม่เฉพาะเจาะจง จุดหมายปลายทาง หรือ XBee ทุกตัวที่อยู่ในระบบเครือข่ายเดียวกันสามารถรับข้อมูลทุกข้อมูลได้ทุกตัว

## 2) Cluster Tree



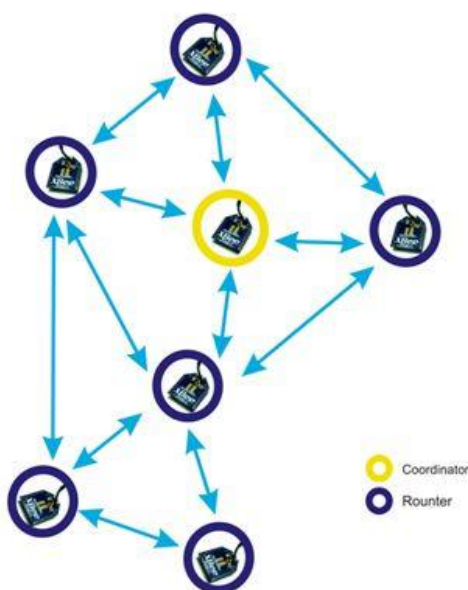
รูปที่ 2.16 Cluster Tree Network

เป็นการ รับส่งข้อมูล แบบ ส่งผ่าน เช่น A ต้องการติดต่อ กับ C แต่ C อยู่ไกลจาก A จน A ไม่สามารถ ติดต่อกับ C ได้ แต่พอดีมี B ที่อยู่ระหว่าง A กับ C ดังนั้น Cluster Tree จะใช้ B เป็น เหมือน ตัวกลาง เชื่อมการติดต่อ (Repeater) ระหว่าง A กับ C

จากรูปภาพด้านบน จะพบว่า มีการใช้งาน XBee อยู่ 3 ลักษณะด้วยกันคือ

- Coordinator
- End Device
- Routers

### 3) Mesh



รูปที่ 2.17 Mesh Network

การเชื่อมต่อเครือข่ายแบบ Mesh เป็นโครงข่ายที่มีประสิทธิภาพสูงเนื่องจาก ข้อมูลสามารถ ส่งไปถึงเป้าหมายได้หลายทาง ทำให้ ระบบนี้สามารถรับส่งข้อมูล ไปยังจุดหมายปลายทางได้ แม้จะ เกิดความเสียหายของระบบในบางส่วนก็ตาม (ขึ้นอยู่กับ การออกแบบระบบของผู้ใช้ด้วย) ระบบนี้ จึงเป็นระบบที่ได้รับความนิยมเป็นอย่างมาก

จากรูปภาพด้านบน จะพบว่า มีการใช้งาน XBee อยู่ 2 ลักษณะด้วยกันคือ

- Coordinator
- Routers

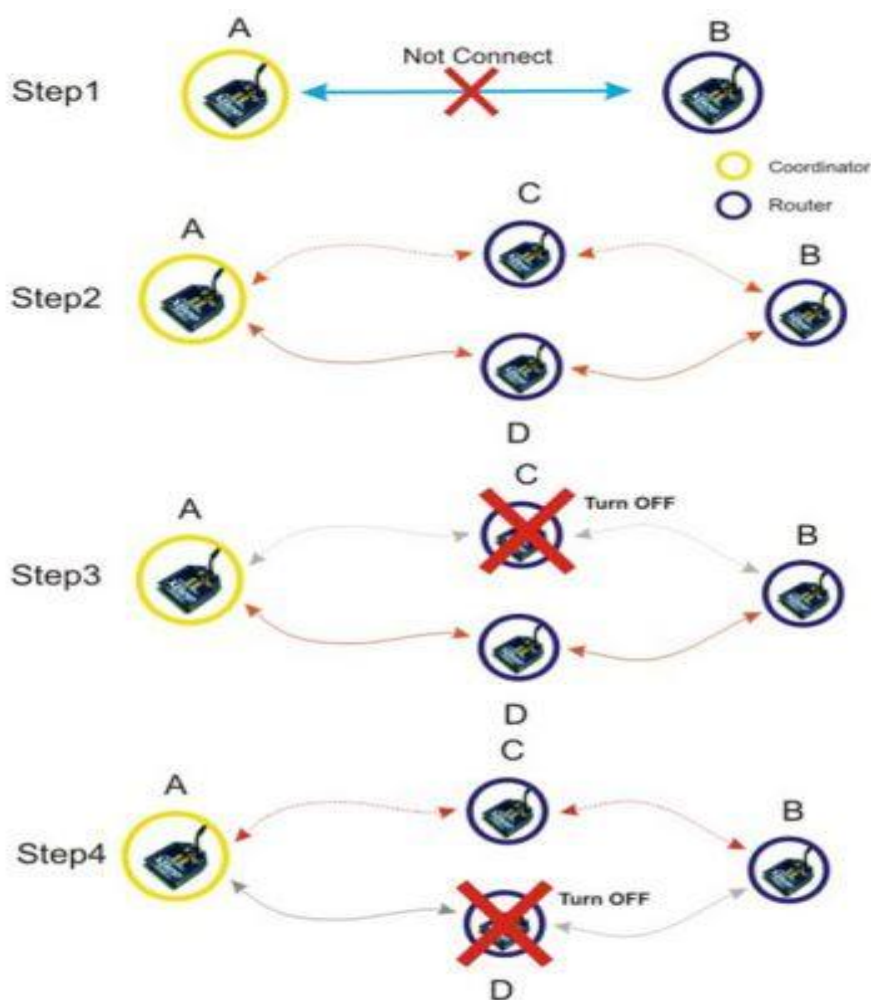
### 2.5.2 การทดลองรับส่งข้อมูลผ่าน ระบบเครือข่ายแบบ Mesh

1) นำ XBee ที่ตั้งค่าแล้ว คือ Coordinator (A) และ Router (B) สองตัวมาทดลองส่งข้อมูลหากันโดยตั้ง DH, DL ของทั้ง 2 ตัวให้สามารถส่งข้อมูลหากันได้ จากนั้น แยก XBee ทั้ง 2 ตัวออกจากกันจนไม่สามารถ ติดต่อกัน ได้อีก

2) จากนั้น นำเอา XBee อีก 2 ตัวที่ตั้งค่าเป็น Router (C, D) มาวางคั่นกลาง จะพบว่า เราจะสามารถ ติดต่อระหว่าง Coordinator (A) และ Router (B) ได้อีกครั้ง

3) ปิดแต่ C แล้ว A ยังสามารถคุยกับ B ได้ โดยผ่าน (Mesh) D

4) เมื่อปิดแต่ D แล้ว A ก็ยังสามารถคุยกับ B โดยผ่าน (Mesh) C จากการทดลองนี้ จะพบว่า ระบบเครือข่ายแบบ Mesh สามารถค้นหา และ เปลี่ยนแปลงเส้นทางการส่งข้อมูลได้เอง ทำให้เสถียรภาพของการรับส่งข้อมูลในรูปแบบนี้จึงมีประสิทธิภาพมาก



รูปที่ 2.18 ภาพประกอบการอธิบาย การทดลอง รับส่งข้อมูลผ่าน ระบบเครือข่ายแบบ Mesh

## 2.6 เซ็นเซอร์

คือตัวอุปกรณ์ตรวจวัดตัวแรกในระบบการวัด ซึ่งใช้ตรวจจับหรือรับรู้การเปลี่ยนแปลง ปริมาณทางกายภาพของตัวแปรต่าง ๆ เช่น ความร้อน แสง สี เสียง ระยะทาง การเคลื่อนที่ ความดัน การไหล เป็นต้น แล้วเปลี่ยนให้อยู่ในรูปของสัญญาณหรือข้อมูลที่สอดคล้องและเหมาะสมกับส่วนของการกำหนดเงื่อนไข ทางสัญญาณ ถ้าใช้เซ็นเซอร์วัดแบบสัมผัสกับตัวแปรโดยตรงเรียก ตัวตรวจรู้แบบปฐม (Primary Sensors) หรือตัวตรวจรู้ขั้นต้น หากมีการตรวจรู้โดยผ่านส่วนอื่นก่อน เช่น สเตทรนเกจตรวจรับแรงกดที่ต้องรับแรงถ่ายถอดจากแท่งโลหะที่รับแรงโดยตรงอีกทอดโดยใช้สเตรนเกจแปะติดกับแท่งโลหะดังกล่าวเพื่อวัดแรงนั้น เราจะเรียกสเตรนเกจในกรณีนี้ว่าเป็น ตัวตรวจรู้ทุติยภูมิ (Secondary Sensor) หรือ ตัวตรวจจับขั้นรอง การตรวจรู้จะอาศัยผลการเปลี่ยนแปลงของพารามิเตอร์ในตัวเซ็นเซอร์เองที่สามารถตรวจวัดได้ ซึ่งส่วนใหญ่จะเป็นพารามิเตอร์ทางไฟฟ้า เช่น แรงดัน กระแส ความต้านทาน ความจุ และความเหนี่ยวนำ เป็นต้น เมื่อค่าตัวแปรทางกายภาพเปลี่ยนแปลงแล้วพารามิเตอร์ของตัวเซ็นเซอร์เหล่านี้จะเปลี่ยนตาม เมื่อเราทราบค่าพารามิเตอร์ทางไฟฟ้าที่เปลี่ยนตาม อาจวัดได้โดยใช้มิเตอร์ หรือวงจรบริดจ์ต่าง ๆ ซึ่งเป็นการวัดตัวแปรทางอ้อม ด้วยวิธีทางไฟฟ้าโดยเราจะทำการเทียบหรือ ปรับแต่งปริมาณทางไฟฟ้านี้แทนค่าตัวแปรที่ทำการวัดอีกทอด เราจึงอาจเรียกว่าเป็นการวัดโดยวิธีอ้อมได้ กระบวนการนี้เรียกว่า การตรวจจับ (Sensing) กรณีนี้คำว่าทรานสดิวเซอร์จะถูกเรียกว่า เซ็นเซอร์จะเห็นว่าทรานสดิวเซอร์และเซ็นเซอร์แท้จริงคือ อุปกรณ์ที่ทำหน้าที่เดียวกันต่างตรงที่เราจะกล่าวถึงหลักการการทำงานหรือกล่าวถึงลักษณะการใช้งาน ปัจจุบันทรานสดิวเซอร์และเซ็นเซอร์เป็นคำกลาง ๆ ที่ใช้ร่วมกันโดยทรานสดิวเซอร์อาจจะรวมทั้งตัวเซ็นเซอร์และวงจรการปรับแต่งสัญญาณต่าง ๆ เข้าเป็นหน่วยเดียวกัน แล้วนำไปใช้ได้ทันที เช่น ทรานสดิวเซอร์ความดัน (Pressure Transducer) เมื่อมีความดันเข้ามาจะทำให้เป็นแรงดันไฟฟ้าหรือกระแสที่ด้านทางออก (Output) ที่แปรเป็นสัดส่วนกับความดันที่ต้องการวัด เป็นต้น

### 2.6.1 ประเภทตัวตรวจจับ

#### 1) ตัวตรวจจับแบบใช้แสง

ตัวตรวจจับแบบใช้แสง จะใช้ลำแสงที่มีขนาดเล็กตรวจสอบว่าวัตถุอยู่ในตำแหน่งที่ต้องการหรือไม่เป็นวิธีที่ไม่ละเอียดและมีความถูกต้องน้อยที่สุด ไม่นิยมใช้กับแขนกลที่ต้องการความละเอียดสูงการติดตั้งตัวตรวจจับแบบใช้แสงสามารถติดตั้งได้ง่ายโดยการติดตั้ง LED แบบอินฟราเรดที่ด้านหนึ่งและติดตั้งตัวรับแสงอินฟราเรดที่อีกด้านหนึ่ง โดยให้สิ่งที่ต้องการตรวจจับอยู่ระหว่างกลาง

## 2) ตัวตรวจจับแบบไม่สัมผัส

ตัวตรวจจับแบบไม่สัมผัส หรือที่นิยมเรียกตามศัพท์ตรง ๆ ว่า พร็อกซิมีตี้เซ็นเซอร์ ใช้หลักการทางไฟฟ้าดังนี้

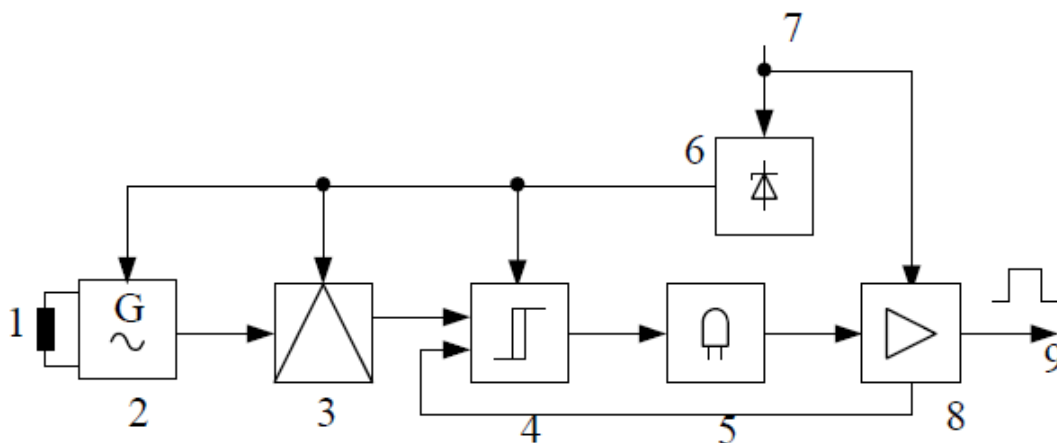
- ตัวตรวจจับแบบเหนี่ยวนำ (Inductive Sensors)



รูปที่ 2.19 ตัวอย่างตัวตรวจจับแบบไม่สัมผัสใช้หลักการเหนี่ยวนำ

ตัวตรวจจับแบบเหนี่ยวนำเรียกตามศัพท์ คือ อินดักทีฟเซ็นเซอร์ ทำงานโดยใช้หลักการเปลี่ยนแปลงค่าความเหนี่ยวนำของขดลวด วัตถุที่ตรวจจับได้จะเป็นโลหะเท่านั้น ลักษณะภายนอกแสดงดังรูปที่ 2.21 ส่วนประกอบภายในประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 2.22 ส่วนประกอบ

- 1) ส่วนตรวจจับ (ขดลวด) (Active Zone : Coil)
- 2) วงจรกำเนิดคลื่นความถี่ (Oscillator)
- 3) ส่วนการประมวลผล (Evaluator)
- 4) วงจรเปรียบเทียบและจุดชนวน (Trigger)
- 5) หลอดแสดงสถานะการทำงาน (Status display)
- 6) วงจรคงค่าแรงดันภายใน (Internal Constant Voltage Supply)
- 7) แหล่งจ่ายแรงดันภายนอก (External Voltage)
- 8) วงจรขยายและป้องกันด้านออก (Output and Protective)
- 9) สัญญาณด้านออก (เป็นแบบ ON-OFF)



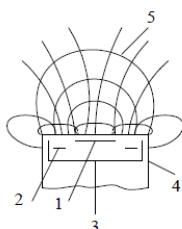
รูปที่ 2.20 แสดงส่วนประกอบของตัวตรวจจับแบบเหนี่ยวนำ

### 3) ตัวตรวจจับแบบความจุไฟฟ้า (Capacitive Sensor)

ตัวเหนี่ยวนำแบบนี้ใช้ตรวจจับวัตถุทั้งโลหะและไม่ใช่โลหะ ใช้หลักการตรวจจับค่าความจุที่เปลี่ยนแปลง เรียกทับศัพท์ด้านเทคนิคว่า คาปาซิทีฟเซ็นเซอร์



รูปที่ 2.21 ตัวอย่างตัวตรวจจับแบบความจุแบบต่าง ๆ



รูปที่ 2.22 ภาพตัดขวางด้านข้างส่วนตรวจจับของตัวตรวจจับแบบไม่สัมผัสแบบความจุไฟฟ้า

จากรูปมีส่วนประกอบดังนี้

- Active Electrode
- อิเล็กโทรดขดเชย
- Earth Electrode
- โครงของตัวตรวจจับ
- สนามไฟฟ้า

โครงสร้างและส่วนประกอบจะคล้ายกับตัวตรวจจับแบบความเหนียวนำ ต่างกันที่ ส่วนตรวจจับจะใช้หลักการของการเปลี่ยนค่าความจุแทน เมื่อวัตถุที่จะตรวจจับเคลื่อนที่เข้ามาที่ ระยะสนามไฟฟ้าของตัวเก็บประจุของตัวตรวจจับซึ่งเกิดจาก Active และ Earth Electrode และอาจ มีตัวนำขดเชยซึ่งทำหน้าที่ป้องกันและชดเชยผลของความชื้นที่ด้านหน้าของบริเวณตรวจจับ เมื่อมี วัตถุเคลื่อนที่เข้ามาในบริเวณตรวจจับ ค่าความจุของวงจรกำเนิดคลื่นความถี่จะเปลี่ยนแปลงไป

#### 2.6.2 หลักการทำงาน

จากรูปที่ 2.22 แสดงโครงสร้างและส่วนประกอบของตัวตรวจจับแบบความจุ เมื่อมี วัตถุใด ๆ เคลื่อนที่เข้ามาในบริเวณสนามไฟฟ้า จะทำให้ค่าความจุของวงจรกำเนิดความถี่มีการ เปลี่ยนแปลง ซึ่งขึ้นกับค่าระยะห่างระหว่างตัวกลางหรือวัตถุกับด้านหน้าของส่วนตรวจจับ ค่าคงที่ ทางไฟฟ้าของตัวกลาง (Dielectric Constant คือ ถ้ามีค่าคงที่มากระยะการตรวจจับก็จะมีระยะไกล ขึ้น) รวมทั้งขนาดและรูปร่างของตัวกลางตัวตรวจจับแบบความจุสามารถตรวจจับวัตถุตัวกลางได้ ทั้งที่เป็น โลหะและไม่ใช่โลหะ การทำงาน (ON) และ ไม่ทำงาน (OFF) นั้นได้จากสถานะของวงจร กำเนิดความถี่ ว่ามีการออสซิลเลตหรือไม่ โดยใช้หลักการเช่นเดียวกันกับตัวตรวจจับแบบเหนียวนำ ระยะการตรวจจับมาตรฐานได้จากการใช้แผ่นโลหะเป็นวัตถุตัวกลางเมื่อเปลี่ยนวัตถุตัวกลางเป็น วัสดุอื่นระยะทางก็จะต่างกันออกไป โดยการคูณค่าตัวประกอบ กับระยะมาตรฐานจะได้ระยะ ตรวจจับ



รูปที่ 2.23 Infrared Sensor

## 2.7 แรงเสียดทาน

แรงเสียดทาน หมายถึง แรงที่เกิดขึ้นระหว่างผิวสัมผัสของวัตถุ 2 ชิ้นที่สัมผัสกัน ซึ่งแรงนี้เป็นแรงที่ผิววัตถุผิวหนึ่งต้านทานการเคลื่อนที่ของผิววัตถุอีกผิวหนึ่ง ส่งผลทำให้วัตถุเคลื่อนที่ช้าลงเรื่อย ๆ จนกระทั่งหยุดนิ่งในที่สุด แรงเสียดทานจะกระทำในทิศตรงข้ามกับการเคลื่อนที่ของวัตถุ ถ้าไม่มีแรงเสียดทานวัตถุจะเคลื่อนที่ด้วยอัตราคงตัวตลอดการเคลื่อนที่ แต่เมื่อมีแรงเสียดทานวัตถุจะเคลื่อนที่ช้าลงเรื่อย ๆ จนหยุดนิ่งในที่สุดขนาดของแรงเสียดทานจะมากหรือน้อย ขึ้นอยู่กับลักษณะของผิวสัมผัส และน้ำหนักของวัตถุที่กดลงบนอีกพื้นผิวหนึ่งเป็นหลักหากน้ำหนักของวัตถุมาก แรงที่กดลงบนพื้นผิวอีกพื้นผิวหนึ่งก็จะมาก แรงเสียดทานที่เกิดขึ้นก็จะมาก อีกทั้งหากวัตถุต้องเคลื่อนที่บนพื้นผิวขรุขระมาก ก็จะมีแรงเสียดทานเกิดขึ้นมากกว่าตอนเคลื่อนที่ที่อยู่บนพื้นผิวที่ขรุขระน้อย

### 2.7.1 ประเภทของแรงเสียดทาน

แรงเสียดทานมี 2 ประเภท คือ

- 1) แรงเสียดทานสถิต (Static Friction) คือ แรงเสียดทานที่เกิดขึ้นระหว่างผิวสัมผัสของวัตถุ ในสภาวะที่วัตถุได้รับแรงกระทำแล้วอยู่นิ่ง
- 2) แรงเสียดทานจลน์ (Kinetic Friction) คือ แรงเสียดทานที่เกิดขึ้นระหว่างผิวสัมผัสของวัตถุในสภาวะที่วัตถุได้รับแรงกระทำแล้วเกิดการเคลื่อนที่ด้วยความเร็วคงที่

### 2.7.2 ปัจจัยที่เป็นผลต่อแรงเสียดทาน

- 1) น้ำหนักหรือแรงกดของวัตถุที่กดลงบนพื้น
 

ถ้าน้ำหนักหรือแรงกดของวัตถุมาก จะเกิดแรงเสียดทานมาก ถ้าน้ำหนักหรือแรงกดของวัตถุน้อยจะเกิดแรงเสียดทานน้อย
- 2) ลักษณะของพื้นผิวสัมผัส
  - ถ้าพื้นผิวเรียบ เช่น กระจก กระจก พลาสติก เป็นต้น จะเกิดแรงเสียดทานน้อย เนื่องจากพื้นผิวเรียบ มีการเสียดสีระหว่างกันน้อย
  - ถ้าพื้นผิวขรุขระ เช่น พื้นทราย พื้นหญ้า พื้นหินกรวด เป็นต้น จะเกิดแรงเสียดทานมาก เนื่องจากพื้นผิวขรุขระมีการเสียดสีระหว่างกันมาก จึงมีแรงเสียดทานที่ต้านการเคลื่อนที่ของวัตถุเกิดขึ้น

## 2.8 PIC

### 2.8.1 ไมโครคอนโทรลเลอร์คืออะไร

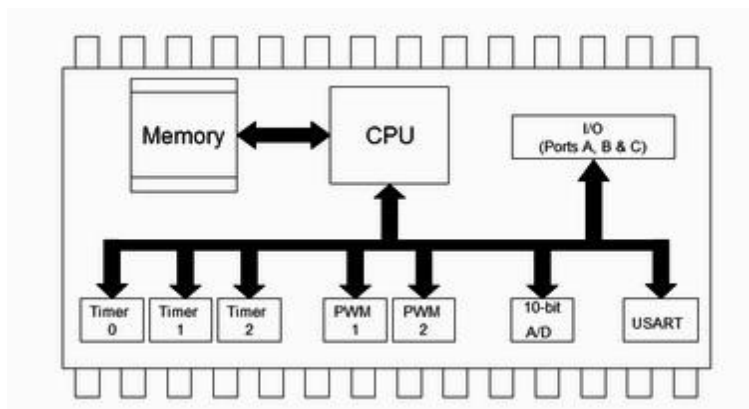
ไมโครคอนโทรลเลอร์คืออุปกรณ์ประเภทสารกึ่งตัวนำที่รวบรวมฟังก์ชันการทำงานต่างๆไว้ภายในตัวของมันเอง โดยมีโครงสร้างใกล้เคียงกับคอมพิวเตอร์คือภายในประกอบด้วยหน่วยรับข้อมูลและโปรแกรมหน่วยประมวลผล หน่วยความจำ หน่วยแสดงผล ซึ่งส่วนประกอบเหล่านี้มีความสมบูรณ์ในตัวของมันเอง ทำให้มีขนาดเล็ก และสามารถเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ต่างๆที่เชื่อมต่อกับตัวมันง่ายต่อการนำไปประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คำหนึ่งคือ ไมโคร (Micro) หมายถึงขนาดเล็กและคำว่า คอนโทรลเลอร์ (Controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์ จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ ที่คนโดยส่วนใหญ่คุ้นเคย กล่าวคือภายใน ไมโครคอนโทรลเลอร์ ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถึงเดียวกันความแตกต่างของ Microcontroller กับ Microcomputer คือ Microcontroller นั้นมีความสมบูรณ์ภายในตัวของมันเองคือ มีส่วนประกอบต่างๆ ครบถ้วนส่วน Microcomputer นั้นต้องทำงานร่วมกับอุปกรณ์ข้างเคียงที่เชื่อมต่อจากภายนอก เช่น แป้นพิมพ์ เครื่องอ่านเขียนแผ่นบันทึก หน่วยความจำ I/O ฯลฯ

### 2.8.2 ข้อแตกต่างระหว่างไมโครโปรเซสเซอร์กับไมโครคอนโทรลเลอร์

ไมโครโปรเซสเซอร์ที่ใช้อยู่ในปัจจุบัน จะไม่มีหน่วยความจำ RAM, ROM และ Port อยู่ในตัว ทำให้ต้องต่อหน่วยความจำโปรแกรมภายนอกเพิ่มและต้องใช้ ICs ขยายพอร์ตเพิ่มเติม ข้อดีคือ สามารถเพิ่มหน่วยความจำได้ตลอด ส่วนไมโครคอนโทรลเลอร์จะมีวงจรพื้นฐานประกอบอยู่ภายในชิป เช่น หน่วยความจำ RAM, ROM และ I/O Port ดังนั้น ในระบบไมโครคอนโทรลเลอร์ จึงมีขนาดเล็กกว่าและราคาต่ำกว่าระบบไมโครโปรเซสเซอร์

### 2.8.3 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้



รูปที่ 2.24 แสดงโครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

1) หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) เป็นวงจรอิเลคทรอนิกส์ที่ทำงานหรือประมวลผล ตามชุดของคำสั่งเครื่องจากซอฟต์แวร์ คำนี้เริ่มใช้ในอุตสาหกรรมคอมพิวเตอร์ตั้งแต่ต้นศตวรรษ 1960s หน่วยประมวลผลเปรียบเสมือนเป็นสมองของคอมพิวเตอร์ ในการทำหน้าที่ตัดสินใจหรือคำนวณ จากคำสั่งที่ได้รับมา เช่น การเปรียบเทียบ การกระทำการทางคณิตศาสตร์ ฯลฯ โดยมีกระบวนการพื้นฐานคือ

- อ่านชุดคำสั่ง (fetch)
- ตีความชุดคำสั่ง (decode)
- ประมวลผลชุดคำสั่ง (execute)
- อ่านข้อมูลจากหน่วยความจำ (memory)
- เขียนข้อมูล/ส่งผลการประมวลกลับ (write back)

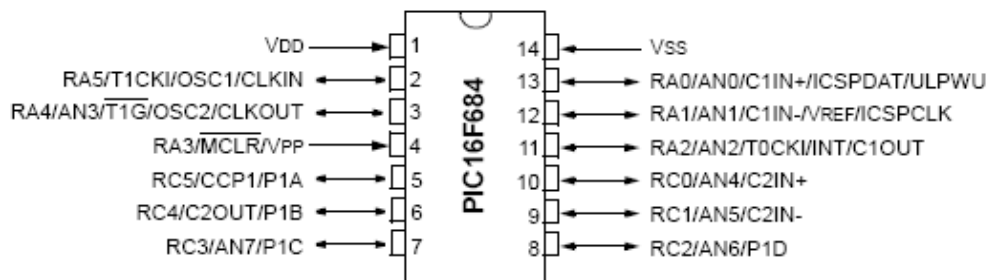
2) หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดานทศในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3) ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับ

อุปกรณ์ภายนอกถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4) ช่องทางเดินของสัญญาณหรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณ ข้อมูลระหว่าง ซีพียูหน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัว ไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม Control Bus )บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูล เพื่อการประมวลผลทั้งหมด ขนาดของบัสจะขึ้นอยู่กับความสามารถการประมวลผลของซีพียู สำหรับในงานทั่วไป ขนาดของบัสข้อมูลจะเป็น 8 บิต และในปัจจุบันได้มีการพัฒนาขึ้นมาจนถึง 16,32 และ 64 บิต บัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้น ซีพียู ต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ดังนั้นจำนวนสายสัญญาณของแอดเดรสจึงต้องมีจำนวนมาก ยิ่งมากเท่าไร ก็จะเป็นการแสดง ขนาดของหน่วยความจำที่ไมโครคอนโทรลเลอร์ สามารถติดต่อกับได้ โดยสามารถคำนวณได้จาก จำนวนแอดเดรสของหน่วยความจำ = 2 ยกกำลัง  $n$  ( $n$  คือจำนวนของเส้นทาง ) ยกตัวอย่าง ไมโครคอนโทรลเลอร์ตัวหนึ่งมีสายแอดเดรส 10 เส้น ดังนั้นไมโครคอนโทรลเลอร์ตัวนี้ สามารถติดต่อกับหน่วยความจำได้ 2 ยกกำลัง  $10 = 1,024$  ตำแหน่ง หากต้องการทราบความจุของหน่วยความจำจริงๆ จะต้องทราบถึงขนาดของบัสข้อมูลก่อนว่าเป็นเท่าใด หากเป็น 8 บิต ความจุของหน่วยความจำที่มีสายแอดเดรส 10 เส้น จะเท่ากับ  $8 \times 1024 = 8,192$  บิต และ 1 กิโลไบต์ เท่ากับ 1,024 ไบต์ ดังนั้นไมโครคอนโทรลเลอร์ดังกล่าว จึงมีความจุของหน่วยความจำเท่ากับ 8,192 บิต หรือ 1,024 ไบต์ หรือ 1 กิโลไบต์บัสควบคุมเป็นกลุ่มของสายสัญญาณควบคุมการติดต่อทั้งหมดของซีพียูกับหน่วยความจำและพอร์ต สำหรับสายสัญญาณเลือกควบคุมหลักได้แก่ สายสัญญาณเลือก-อ่าน-เขียนหน่วยความจำ สายสัญญาณเลือกเลือก อ่าน-เขียนข้อมูลกับพอร์ต

5) วงจรกำเนิดสัญญาณนาฬิกา (Clock) นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้น ส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย



รูปที่ 2.25 แสดงการจัดขาของไมโครคอนโทรลเลอร์ตระกูล PIC 16F684

Address	Bank 0	Bank 1	Address
00h	INDF	←	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	←	82h
03h	STATUS	←	83h
04h	FSR	←	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	Unimplemented	←	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	←	8Ah
0Bh	INTCON	←	8Bh
0Ch - 4Fh	GPR	←	8Ch - CFh

รูปที่ 2.26 การแบ่งหน่วยความจำของ PIC

#### 2.8.4 PIC ชนิดต่างๆ

MCU ในตระกูล PIC ถ้าแบ่งออกตามชนิดของ PROGRAM MEMORY แบ่งได้เป็น 3 แบบคือ

1. OTP (One Time Programmable)
2. EPROM (Erasable Programmable ROM)
3. EEPROM / Flash (Electrically Erasable Programmable ROM)

## 1) OTP

เป็น Chip ที่มีราคาถูกที่สุด ในสามประเภท สาเหตุก็มาจากว่า Chip แบบ OTP จะสามารถทำการโปรแกรมได้แค่ครั้งเดียวเท่านั้น หลังจาก Chip ได้ถูกโปรแกรมไปแล้วจะไม่สามารถโปรแกรมเข้าไปใหม่ได้อีก ดังนั้น Chip ประเภทนี้จะนิยมใช้หลังจากได้พัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆ ในโปรแกรม ไม่มีอีกแล้ว เพราะจะมีต้นทุนต่ำเมื่อเทียบตัว memory ประเภทอื่น จะมีตัวอักษร C แสดงบนตัว chip เช่น 16C84,16C74



รูปที่ 2.27 Chip แบบ OTP

## 2) EPROM

เป็น Chip ที่มี Program Memory ที่เมื่อเขียนโปรแกรมเข้าไปแล้วสามารถโปรแกรมใหม่ด้วยการลบโปรแกรมเดิมโดยให้แสง UV (Ultra Violet) ส่องผ่านเข้าไปยัง Chip ประมาณ 5-10 นาที ดังนั้นที่ด้านบนของ Chip จะมีกรอบกระจกเพื่อให้แสง UV สามารถส่องผ่านเข้าไปในตัว Chip ได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรมเช่นกัน เมื่อลบโปรแกรมด้วยแสง UV มากๆ เข้าก็จะเกิดการด้านคือโปรแกรมไม่เข้านั่นเอง จะมีตัวอักษร JW หรือว่าดูเอาว่ามีกรอบกระจกอยู่บน Chip หรือไม่



รูปที่ 2.28 Chip แบบ EPROM

### 3) EEPROM / Flash

เป็น Chip ที่ออกมาไม่กี่ปีนี่เอง ส่วนของ Program Memory สามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ใช้เวลาในการ ลบข้อมูลไม่กี่วินาที และสามารถลบ และเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดใน 3 ประเภท มีตัวอักษร F เป็นตัวบอก เช่น 16F84,16F877



รูปที่ 2.29 Chip แบบ EEPROM

#### 2.8.5 PIC เบอร์ต่างๆ

ปัจจุบัน MCU ของ PIC มีหลากหลายเบอร์ จนแทบจะเลือกใช้ไม่ถูก แต่ในความเป็นจริงแล้ว การออกเบอร์ใหม่ๆ ออกมาได้มีการกำหนดทิศทางที่แน่นอน ซึ่งเราสามารถ จะแบ่งออกเป็น 6 กลุ่ม ได้แก่

- 1) PIC12CXXX, PIC12FXXX (FLASH MCUs)
- 2) PIC16C5X
- 3) PIC16CXXX
- 4) PIC17CXXX
- 5) PIC16FXXX (FLASH MCUs)
- 6) PIC18CXXX , PIC18FXXX (FLASH MCUs)

ไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้งานควบคุมหลายประการเช่น

- ชิพไอซีและระบบที่ได้มีขนาดเล็ก
- ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์
- วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นจากการต่อ

วงจร

- มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุม โดยเฉพาะซึ่งใช้งานได้ง่าย
- ช่วยลดระยะเวลาในการพัฒนาระบบได้

## 2.9 Visual Basic

### 2.9.1 ประวัติความเป็นมาของ Visual Basic

Visual Basic เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดยบริษัท ไมโครซอฟท์ ซึ่งเป็นบริษัทยักษ์ใหญ่ที่สร้างระบบปฏิบัติการ Windows 95/98 และ Windows NT/XP ที่ใช้กันอย่างแพร่หลายในปัจจุบัน โดยตัวภาษาเองมีรากฐานมาจากภาษา Basic ซึ่งย่อมาจาก Beginner's All Purpose Symbolic Instruction ซึ่งหมายถึง ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น ภาษา Basic มีจุดเด่นคือผู้ที่ไม่มีพื้นฐานเรื่องการเขียนโปรแกรมเลยก็สามารถเรียนรู้และนำไปใช้งานได้โดยง่ายและรวดเร็ว Visual Basic เวอร์ชันแรกคือเวอร์ชัน 1.0 ออกสู่สายตาประชาชนตั้งแต่ปี 1991 โดยในช่วงแรกยังไม่มีความสามารถต่างจากภาษา QBASIC มากนัก แต่จะเน้นเรื่องเครื่องมือที่ช่วยในการเขียนโปรแกรมบน Windows ซึ่งปรากฏว่า Visual Basic ได้รับความนิยมและประสบความสำเร็จเป็นอย่างดี ไมโครซอฟท์จึงพัฒนา Visual Basic ให้ดีขึ้นเรื่อย ๆ ทั้งในด้านประสิทธิภาพ ความสามารถและเครื่องมือต่าง ๆ เช่น เครื่องมือตรวจสอบแก้ไขโปรแกรม สภาพแวดล้อมของการพัฒนาโปรแกรม การเขียนโปรแกรมแบบหลายวินโดวส์ย่อย (MDI) และอื่น ๆ อีกมากมาย Microsoft Visual Basic เป็นเครื่องมือในการสร้างโปรแกรมบนระบบปฏิบัติการ Windows ที่ใช้งานง่าย โดยการเลือกเครื่องมือต่าง ๆ มาออกแบบหน้าจอของโปรแกรมที่จะสร้าง การเขียนโปรแกรมลักษณะนี้เรียกว่า Visual Programming ซึ่งไม่จำเป็นต้องเขียนคำสั่งต่าง ๆ มากนัก และสามารถสร้างโปรแกรมได้อย่างรวดเร็ว

### 2.9.2 ความสามารถของ Visual Basic

1. สร้างโปรแกรมทางด้านกราฟฟิก โปรแกรมจัดการไฟล์ โปรแกรมคำนวณเลข พื้นฐานทั่วไปที่ทำงานบนระบบปฏิบัติการ Windows
2. สร้างโปรแกรมฐานข้อมูลได้อย่างง่าย เพราะมีเครื่องมือเกี่ยวกับฐานข้อมูลอย่างครบถ้วน และสามารถติดต่อกับฐานข้อมูลได้ทันที
3. มีคอมโพเนนต์ทางด้าน Active X ได้แก่ Active X Component , Active X Control และ Active X Document ซึ่งเป็นเครื่องมือที่สามารถนำส่วนของโปรแกรมที่สร้างไว้ ไปใช้ในโปรแกรมอื่น ๆ ได้
4. สร้างโปรแกรมที่รันบนอินเทอร์เน็ตได้อย่างง่าย โดยไม่ต้องเรียนรู้คำสั่งด้วยภาษา HTML หรือภาษาสคริปต์ที่ใช้งานบนอินเทอร์เน็ต

### 2.9.3 จุดเด่นของ Visual Basic

- 1) มีโครงสร้างภาษาที่ใกล้เคียงภาษามนุษย์ ทำให้เรียนรู้ได้ง่าย
- 2) รวบรวมเครื่องมือที่ช่วยพัฒนา Application ได้อย่างสะดวกรวดเร็วไว้ในตัว

3) สามารถสร้างไฟล์ .EXE ที่สามารถทำงานได้ด้วยตนเอง

4) ออกแบบส่วนหน้าจอติดต่อผู้ใช้ (Form) ได้ทันที โดยไม่ต้องรอเขียนรหัส

โปรแกรม

5) ใน Project บันทึกส่วนติดต่อผู้ใช้ (Form) และส่วนของรหัสโปรแกรม (Code) แยกกัน ทำให้สามารถนำ form เดิมไปปรับปรุงใช้กับ Project อื่นๆ ได้โดยไม่ต้องสร้างใหม่

6) สามารถพัฒนา Application ได้หลายแบบ

#### 2.9.4 คอมพิวเตอร์ที่เหมาะสมสำหรับใช้งาน Visual Basic 6.0

Visual Basic 6.0 ต้องการระบบที่มีคุณสมบัติต่อไปนี้

1) ระบบปฏิบัติการ Windows 95 หรือสูงกว่า สำหรับ Microsoft Visual Basic 6.0 ถ้าเป็น Visual Basic 2005 จะต้องใช้ระบบปฏิบัติการ Windows XP ขึ้นไป

2) หน่วยประมวลผลกลางรุ่น Pentium 90 MHz หรือสูงกว่า

3) เนื้อที่ในฮาร์ดดิสก์ อย่างน้อย 50 MB

4) CD – ROM Drive

5) การ์ดจอ VGA 640 x 480 หรือสูงกว่าที่ Windows สนับสนุน

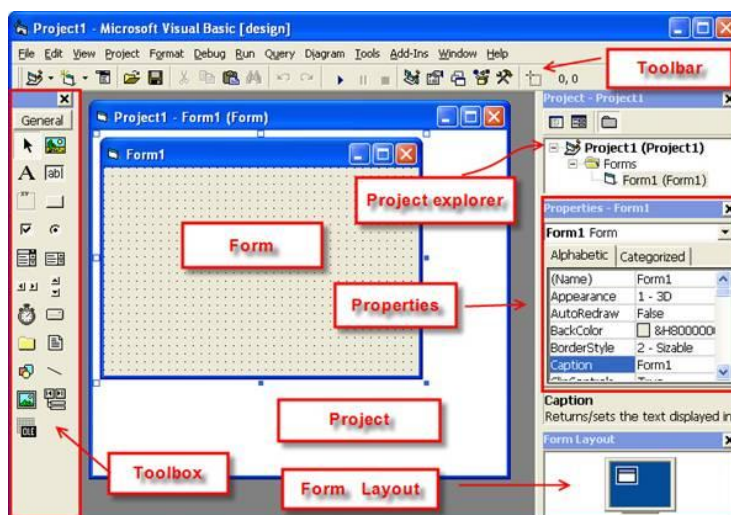
6) RAM 24 MB สำหรับ Windows 95 (ยิ่งมากยิ่งดี)

7) Microsoft Internet Explorer

8) เมาส์หรือ pointing device ที่สนับสนุนการทำงานบน Windows

การติดตั้งโปรแกรม Visual Basic สามารถทำได้เหมือนโปรแกรมทั่วไปที่ใช้ใน Windows คือเรียกไฟล์ Setup.exe จากแผ่น หรืออาจจะทำงานโดยอัตโนมัติเมื่อใส่แผ่นโปรแกรม โดย Visual Basic 6.0 มีทั้งเวอร์ชันที่ขายเดี่ยว ๆ และเวอร์ชันที่อยู่ในชุด Visual Studio 6.0 การติดตั้งจะต้องมี Internet Explorer 4.0 จึงจะติดตั้งโปรแกรมได้ นอกจากนั้น Help และโปรแกรมตัวอย่างต่าง ๆ จะแยกออกมาอยู่ในแผ่นซีดีต่างหาก

#### 2.9.4 ส่วนประกอบต่าง ๆ ของโปรแกรม Visual Basic



รูปที่ 2.30 ส่วนประกอบของโปรแกรม Visual Basic 6

ตารางที่ 2.2 รายละเอียดโปรแกรม

ส่วนประกอบ	รายละเอียด
Form	เป็นส่วนที่ใช้สำหรับจอภาพของโปรแกรมขึ้นใช้งาน โดยจะทำหน้าที่เป็น Background ของจอภาพ
Toolbox	เป็นส่วนที่ประกอบด้วย Icon ต่าง ๆ หรือ ที่เรียกว่า Control ที่จะนำไปใช้งานโดยการนำไปวางบน Form
Toolbar	เป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรม หรือเป็นเครื่องมือที่มีการเรียกใช้บ่อย ๆ
Project Explorer Window	เป็นส่วนที่ใช้สำหรับเรียก Form ต่าง ๆ ขึ้นมาแก้ไข ในกรณีที่มี Form มากกว่า 1 Form
Properties Window	เป็นจอภาพที่ใช้กำหนดคุณสมบัติต่าง ๆ ของ Project ที่เราได้ ออกแบบไว้เพื่อให้ทำงานตามความต้องการ
Form Layout Window	ใช้สำหรับกำหนดตำแหน่งของ Form ที่จะให้แสดงอยู่ในจอภาพเมื่อทำการ Run

1) ทูลบาร์ (Toolbar) เป็นแถบสัญลักษณ์ที่ใช้สำหรับเข้าถึงชุดคำสั่งของ Visual Basic ได้ทันที โดยจะนำคำสั่งที่ถูกใช้งานบ่อย ๆ มาแสดง



รูปที่ 2.31 ทูลบาร์

ทูลบาร์สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ ๆ คือ

- Standard Toolbars เป็นทูลบาร์มาตรฐานประกอบด้วยคำสั่งที่เกี่ยวกับการจัดการ Project
- Edit Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการเขียนโค้ดใน Code Editor
- Debug Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับตรวจสอบการทำงานการประมวลผลโปรแกรม
- Form Editor Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการปรับขนาด, ย้าย, เปลี่ยนตำแหน่งคอนโทรลต่าง ๆ ที่อยู่บนฟอร์ม

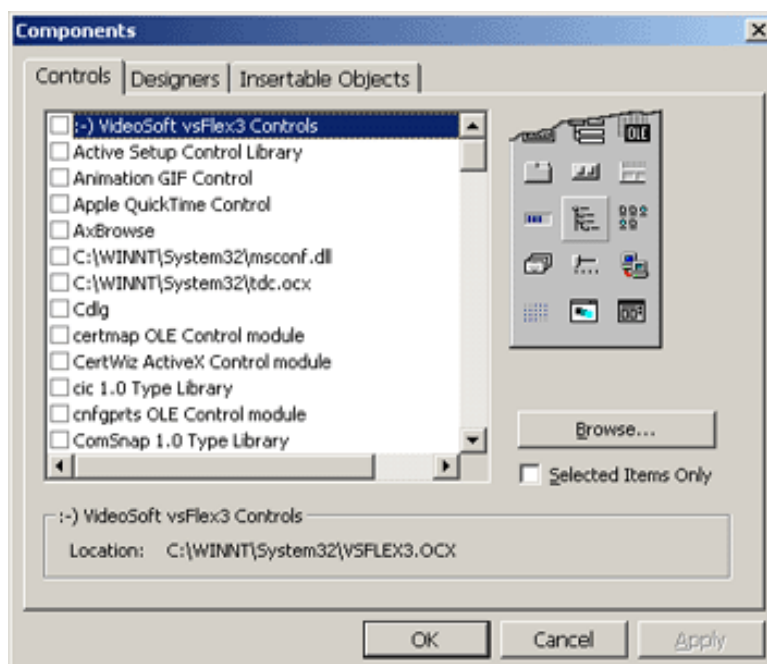
2) Toolbox คือ แถบสัญลักษณ์ Controls ต่าง ๆ ที่ใช้สำหรับพัฒนาโปรแกรมประยุกต์ แบ่ง เป็น 2 กลุ่ม คือ

- คอนโทรลภายใน (Intrinsic Controls) เป็นชุดคอนโทรลมาตรฐานของ Visual Basic ทุก ๆ ครั้งที่มีการเรียกใช้ Form เพื่อสร้างโปรแกรมประยุกต์ คอลโทรลชุดนี้จะถูกเรียกขึ้นมาอัตโนมัติ สามารถเลือกใช้งานคอลโทรลกลุ่มนี้ได้ทันทีตามรูปที่ 2.32



รูปที่ 2.32 ToolBox

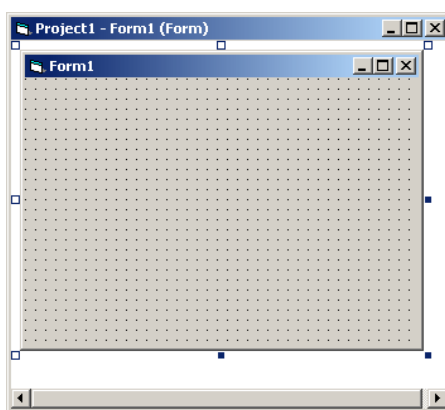
- คอนโทรล ActiveX (ActiveX Controls) เป็นชุดคอนโทรลเพิ่มเติมที่ไม่โครซอฟท์จัดเตรียมไว้ เพื่อเพิ่มขีดความสามารถในการพัฒนาโปรแกรมประยุกต์ การเพิ่มคอนโทรลกลุ่มนี้เข้ามาในทูลบ็อกซ์ทำได้โดยเลือกเมนู Project/Components



รูปที่ 2.33 คอนโทรล ActiveX (ActiveX Controls)

### 3) Form Designer

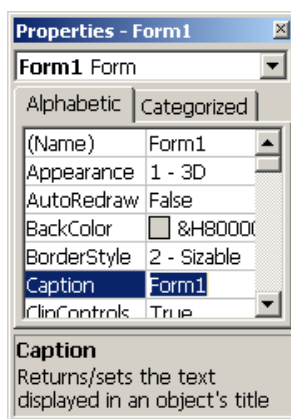
เป็นส่วนที่ใช้ออกแบบการแสดงผลส่วนที่ใช้ติดต่อกับผู้ใช้ ฟอร์มเป็นออบเจกต์แรกที่ถูกเตรียมไว้ให้ใช้งาน คอลโทรลทุกตัวที่ต้องการใช้งานจะต้องนำไปบรรจุไว้ในฟอร์ม นำคอลโทรลมาประกอบกันขึ้นเป็นโปรแกรมประยุกต์ ทุกครั้งที่เปิด Visual Basic ขึ้นมา หรือ สร้าง Project ใหม่จะมีฟอร์มว่าง 1 ฟอร์มถูกสร้างเตรียมไว้เสมอ



รูปที่ 2.34 Form Designer

### 4) Properties Window

หน้าต่างคุณสมบัติเป็นส่วนที่ใช้กำหนดคุณสมบัติของออบเจกต์ที่ถูกเลือก (Active) หรือได้รับความสนใจ (Focus) อยู่ขณะนั้น ซึ่งสามารถที่จะปรับเปลี่ยนค่าต่าง ๆ ของคอลโทรลเพื่อให้เกิดความเหมาะสมและตรงกับความต้องการใช้งานได้ทันที



รูปที่ 2.35 Properties Window

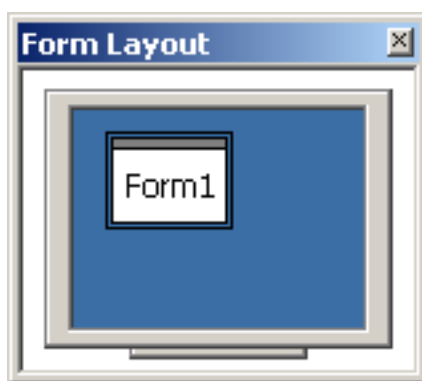
ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็บ 2 แท็บ คือ

1) แท็บ Alphabetic เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ

2) แท็บ Categorized เป็นแท็บที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของ คุณสมบัติที่มีหน้าที่คล้ายกัน

5) Form Layout

เป็นส่วนที่แสดงให้เห็นตำแหน่งของฟอร์ม และสามารถกำหนดตำแหน่งของ ฟอร์ม ที่ปรากฏบนจอภาพในขณะประมวลผลได้ โดยการเคลื่อนย้ายฟอร์มจำลอง ที่อยู่ในจอภาพ จำลองด้วยการ drag เม้าส์ ไปยังตำแหน่งที่คุณต้องการ โดยจะมีผลในขณะประมวลผลเท่านั้น



รูปที่ 2.36 Form Layout

## 2.10 Solid Work

Solid Works พัฒนาขึ้นในปี 1995 โดยบริษัท Dassault System ในฝรั่งเศสเป็นซอฟต์แวร์ เพื่อให้แก้ออกแบบใช้ เป็นเครื่องมือในการออกแบบทางวิศวกรรม เพื่อสร้างตัวอย่างผลิตภัณฑ์ จำลองใน Computer ก่อนที่จะสร้างผลิตภัณฑ์ต้นแบบจริง โดยตัวซอฟต์แวร์จะจัดอยู่ในตระกูล CAD (Computer Aided Design) ซึ่งสามารถสร้างชิ้นงานจำลองในรูปแบบ 3D Solid Models เป็นแบบงานแยกชิ้น (Part) และแบบงานประกอบ (Assembly) เพื่อนำไปสร้างเป็น 2D Standard Engineering (CADD = Computer Aided Design and Drafting) โปรแกรม Solid Work เป็นโปรแกรมที่มีความยืดหยุ่นในการทำงานสูงมาก คือ สามารถที่จะทำงานมากมายหลายรูปแบบ ไม่ว่าจะ เป็นชิ้นงานที่ต้องขึ้นเป็น solid หรือ surface ก็มีเครื่องที่รองรับเป็นอย่างดี เมื่อสร้างชิ้นงานเสร็จ เรียบร้อยก็สามารถที่จะประกอบชิ้นงานได้ใน Mode ของชุดคำสั่ง Assembly รวมทั้งผู้ต้องการ

Drawing ของชิ้นงาน ก็เพียงลากชิ้นงานมาวางในใบงานแล้วขนาด จะมองเห็นได้ว่าผู้ใช้งานสามารถ ที่จะประหยัดเวลาในการทำงานและสนุกกับการทำงานอีกด้วย

### 2.10.1 ประสิทธิภาพการทำงาน

ประสิทธิภาพของ Solid Works เป็นการเจาะลึกให้นักออกแบบสามารถสร้างชิ้นงานจำลองทางด้าน Mechanical Engineering Design ได้อย่างสมบูรณ์แบบ นอกจากนี้ยังสามารถนำไปใช้ในการคำนวณทางวิศวกรรม และการตรวจสอบความผิดพลาดของ 3D Solid Models เพื่อลดต้นทุนในการผลิต และลดระยะเวลาการทำงานในการออกแบบ เพื่อเพิ่มประสิทธิภาพการทำงานในบริษัทและองค์กร

### 2.10.2 ลักษณะการทำงาน

Solid Works แบ่งหมวดการทำงานหลักออกเป็น 3 หมวดคือ Part, Assembly และ Drawing โดยรูปแบบการทำงานทั้ง 3 หมวด มีลักษณะการใช้งานดังนี้

1) Part Mode เป็นหมวดการทำงานเริ่มต้นก่อนที่จะก้าวสู่การทำงานในหมวด Assembly และ Drawing ในขั้นนี้จะมีการแบ่งการทำงานออกเป็น 2 ส่วน คือ การใช้ 2D Sketch เพื่อนำไปสู่การสร้างเป็น 3D Feature และมีเงื่อนไขเป็น Feature-Based Modeling และ Parametric โดยมีการอ้างอิงจาก Solid Mode

- Feature-Based Modeling คือ การออกแบบซอฟต์แวร์ให้สามารถทราบถึงคุณสมบัติต่างๆของ Solid Model ที่สร้างขึ้นมา เพื่อให้ผู้ใช้งานสามารถเปลี่ยนแปลงและแก้ไข Model ในลำดับการทำงานแต่ละขั้นได้ง่ายและรวดเร็ว
- Parametric Model คือการออกแบบซอฟต์แวร์ซึ่งใช้เงื่อนไขทางคณิตศาสตร์ในการแก้ไขขนาดรูปร่าง ทางเรขาคณิตของ Model ที่สร้างขึ้นมา
- Solid Model คือแบบจำลองบนคอมพิวเตอร์ที่สามารถแสดงค่าต่างๆ เช่น Density, Material, Mass, Weight เป็นต้น และยังสามารถมองเห็น 3D Model ได้ทุกมุมมอง

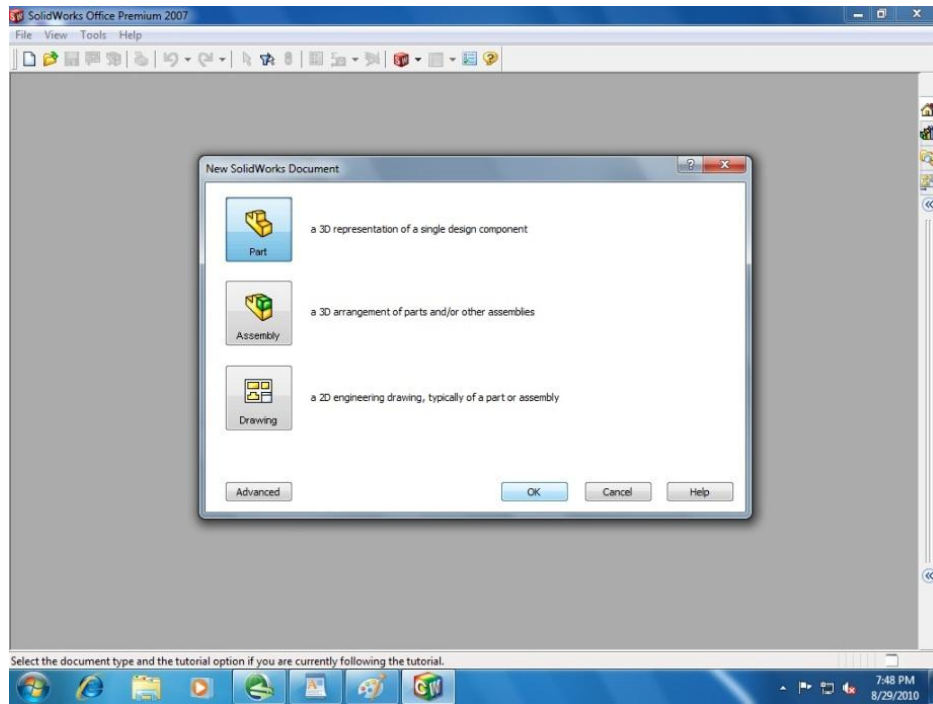
2) Assembly Mode เป็นหมวดการทำงานเพื่อนำ Part Model เข้าไปประกอบเป็นเครื่องจักรกลหรือกลไกต่างๆ และมีเงื่อนไขเป็น Feature Base และ Parametric เช่นเดียวกับ Part Model โดย Part Model และ Assembly จะมีความสัมพันธ์ซึ่งกันและกัน เมื่อทำการแก้ไขในหมวดใด อีก หรือมีการประกอบที่ซ้อนหรือทับกันหมวดจะมีการเปลี่ยนแปลงตามการแก้ไขไปด้วยการทำงานใน Assembly สามารถช่วยให้นักออกแบบหรือวิศวกรสามารถตรวจสอบความผิดพลาดในการสร้าง Part ได้โดยการใช้คำสั่งต่างๆ เช่น คำสั่ง Interference Detection เพื่อตรวจสอบการขัดกันเมื่อมีการเคลื่อนที่ โดยใช้คำสั่ง Move Component เพื่อตรวจสอบการเคลื่อนที่ของกลไก คำสั่ง

Simulation เพื่อจำลองต้นกำลังในการทำงานจริงของเครื่องจักร หรือหากชิ้นงานจำลองที่ออกแบบมีข้อผิดพลาด ก็สามารถแก้ไข Part ใน Assembly ได้เลย ทำให้การออกแบบเป็นเรื่องง่าย และผู้ออกแบบจะสนุกกับการทำงาน Design การทำงานใน Assembly Mode มีลักษณะการทำงาน 2 กรณีได้แก่

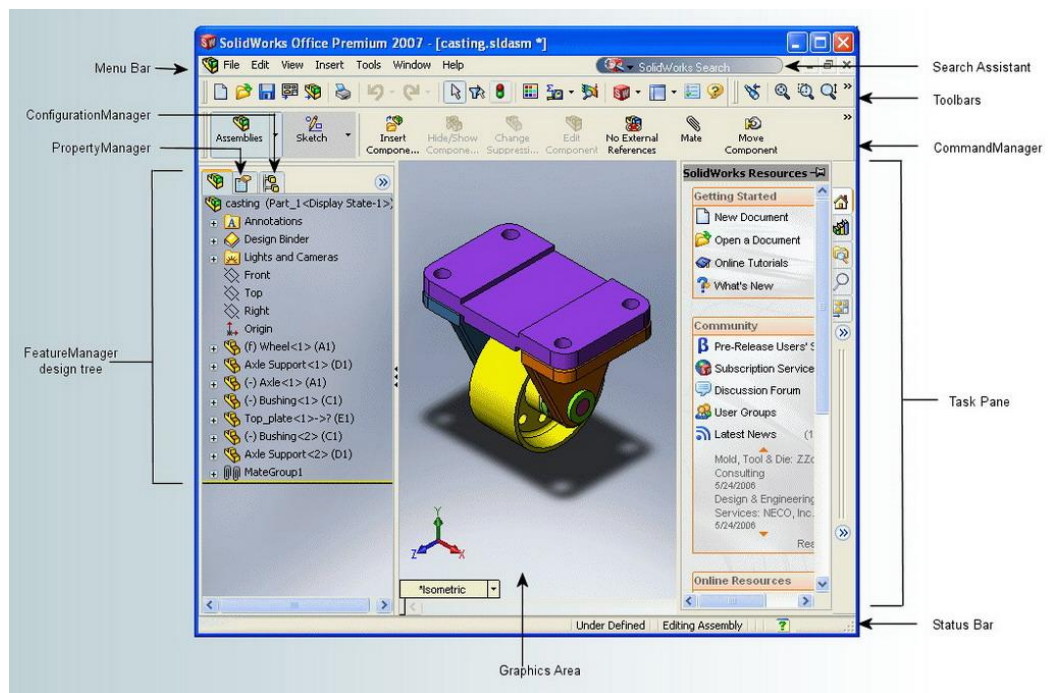
- Bottom-Up Assembly คือ การนำ 3D Models ต่างๆ ที่สร้างเสร็จแล้วใน Part Mode ไปวางในหน้าต่าง Assembly เพื่อทำการประกอบ โดยการใช้คำสั่ง Mate หรือ Smart Mate ซึ่งวิธีนี้จะเหมาะสำหรับผู้ใช้ในระดับเริ่มต้นหรือขั้น Basic
- Top-Down Assembly คือการสร้าง 2D Sketch เป็นโครงร่างระหว่างชิ้นส่วนต่าง ๆ ระหว่าง Part หรือการสร้าง Part ใน Assembly โดยให้มีขนาดและรูปร่างที่มีการอ้างอิงกับ Part อื่น ๆ ทั้งในส่วน Sketch และ Feature วิธีนี้เหมาะกับผู้ใช้ในระดับ Advance

3) Drawing Mode เป็นหมวดการทำงานเพื่อสร้าง 2D Standard Engineering โดยในหมวดนี้เป็นการสร้างมุมมองและกำหนดรายละเอียดตามระบบมาตรฐานต่าง ๆ โดยจะแบ่งการทำงานออกเป็น 2 ส่วนคือ

- Generative Drafting เป็นการสร้าง 2D Sketch และ Interaction Drafting ซึ่งเป็นการนำ 3D Model จาก Part และ Assembly มาวางใน Drawing เพื่อสร้างเป็น 2D Drafting จะมีลักษณะเป็น Parametric และ Relation เช่นกัน แต่จะไม่สามารถใช้คำสั่งใน Drawing Commands ได้ เพราะคำสั่งต่าง ๆ จะต้องอ้างอิงกับ 3D Model
- Interaction Drafting คือการนำ 3D Model จาก Part และ Assembly มาวาง Drawing เพื่อสร้างเป็น 2D Drafting การทำงานในหมวดนี้สามารถใช้คำสั่งจาก Annotation Command และ Drawing Command เพื่อสร้างมุมมองและกำหนดรายละเอียดได้โดยอัตโนมัติ



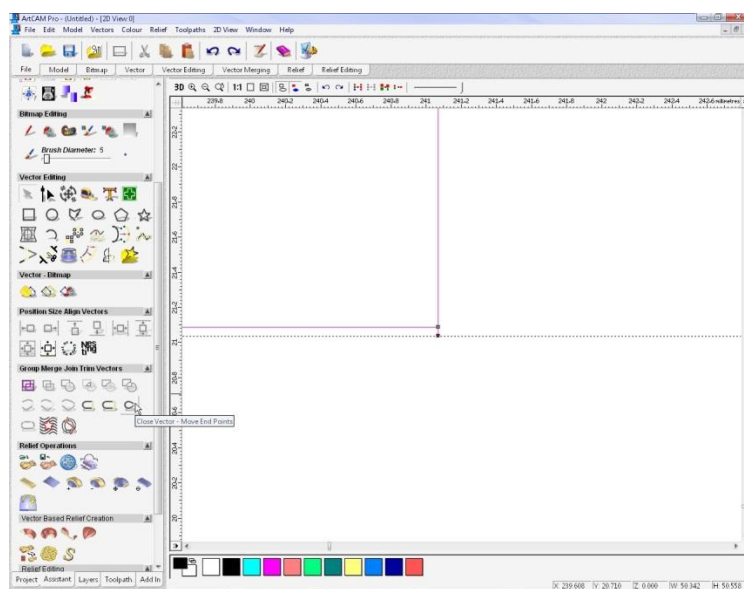
รูปที่ 2.37 หมวดการทำงานของโปรแกรม Solid Work



รูปที่ 2.38 Interface ของ Solid Work 2007

## 2.11 ArtCAM

เป็นซอฟต์แวร์ CAD/CAM ที่ช่วยสร้างสร้งงานทางด้านศิลปะ สามารถทำได้ทั้งการออกแบบและการกัดงาน โดยมีแนวการใช้งานช่างศิลป์ ซอฟต์แวร์นี้จะช่วยให้ผู้ใช้มีความคิดสร้างสรรค์ในด้านการออกแบบ งานศิลปะ งานอัญมณี งานแกะสลัก เพื่อเพิ่มผลผลิตยกระดับคุณภาพผลงานให้มีความทันสมัยยิ่งขึ้น มีเทคนิคการสร้างงานอัตโนมัติ พร้อมไปกับการสร้างผลงานจากทักษะเดิม ArtCAM Pro เป็นซอฟต์แวร์สร้างแบบงาน 3D ที่เพียบพร้อมไปด้วยวิธีการที่หลากหลายที่นำมาใช้แก้ปัญหาแบบเบ็ดเสร็จในงานแกะสลักและงานกัดด้วยเครื่อง CNC ทุกประเภทให้สามารถทำงานได้รวดเร็วและง่ายขึ้นและเพิ่มผลผลิตได้มากกว่า ไม่ว่าท่านจะสร้างงานแบบงานจากการร่างบนกระดาษหรือข้อมูลจากโปรแกรมใดๆก็สามารถทำงานกับซอฟต์แวร์ ArtCAM ได้อย่างสะดวกทั้งยังประกอบด้วยเครื่องมือที่มีประสิทธิภาพในการปรับแต่งเวกเตอร์และบิตแมพ ทำให้ท่านสามารถสร้างงานออกแบบใหม่หรือการแปลงข้อมูลรูปภาพรวมถึงการสร้างและปรับแต่งงานที่เป็นตัวอักษรทั้งตัวหนาและตัวจมนอกจากนั้นยังเน้นการเพิ่มมูลค่าและคุณภาพงานด้วยความสามารถถึงพื้นผิวให้เรียบเนียน กลมกลื่นและผสานระหว่างพื้นผิวอย่างลงตัว สำหรับการกัดงาน ArtCAM ก็จะช่วยประมวลผลในการสร้างทางเดินมีด (Toolpath) ได้อย่างรวดเร็วและแม่นยำด้วยวิธีการกัดงานที่หลากหลายและเลือกใช้เครื่อง CNC ได้ทุกยี่ห้อ



รูปที่ 2.39 Interface ของ ArtCAM

**บทที่ 3**  
**วิธีการดำเนินงาน**

เนื่องจากโปรเจกต์นี้มีการทำงานหลายๆ ส่วน ทั้งเรื่องการทำเอกสารและการทำหุ่นยนต์ ซึ่งต้องอาศัยการจัดเวลาให้เหมาะสมเพื่อให้งานสำเร็จไปได้ด้วยดี ดังนั้นจึงจำเป็นต้องมีการแบ่งเวลาในการทำงานเป็นช่วงๆ

**3.1 แผนงานการดำเนินงาน**

ในการทำโปรเจกต์ครั้งนี้กลุ่มของข้าพเจ้าได้วางแผนการทำงานไว้เป็นส่วนๆ โดยแบ่งเป็นช่วงศึกษาข้อมูล, ลงมือทำงานและจัดทำรูปเล่มโครงงาน โดยมีการวางแผนงานตามตารางที่ 3.1

ตาราง 3.1 แผนงานการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน						
	พ.ศ. 2553		พ.ศ. 2555				
	พ.ย.	ธ.ค.	มี.ย	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1. เสนอหัวข้อโครงงาน	←-----→						
2. ค้นคว้าข้อมูลและรายละเอียดเกี่ยวกับสูญญากาศ			←-----→	←-----→			
3. ศึกษาข้อมูลรายละเอียดของ PIC			←-----→	←-----→			
4. ค้นคว้าโครงสร้างหุ่นยนต์			←-----→	←-----→			
5. ศึกษาเซนเซอร์ที่จะใช้			←-----→	←-----→			
6. ออกแบบและสร้างชิ้นงาน				←-----→	←-----→		
7. ทดสอบหาข้อผิดพลาด					←-----→	←-----→	
8. แก้ไขข้อผิดพลาด						←-----→	
9. จัดทำรูปเล่มโครงงาน					←-----→	←-----→	←-----→

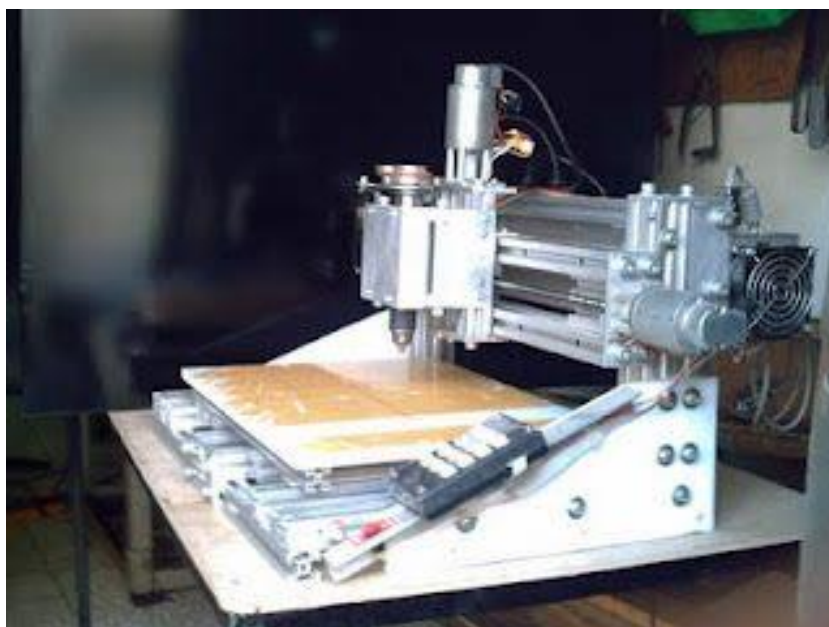
←-----→ แสดงแผนการดำเนินงาน      ←-----→ แสดงการดำเนินการจริง

### 3.2 การออกแบบ / เครื่องมือ

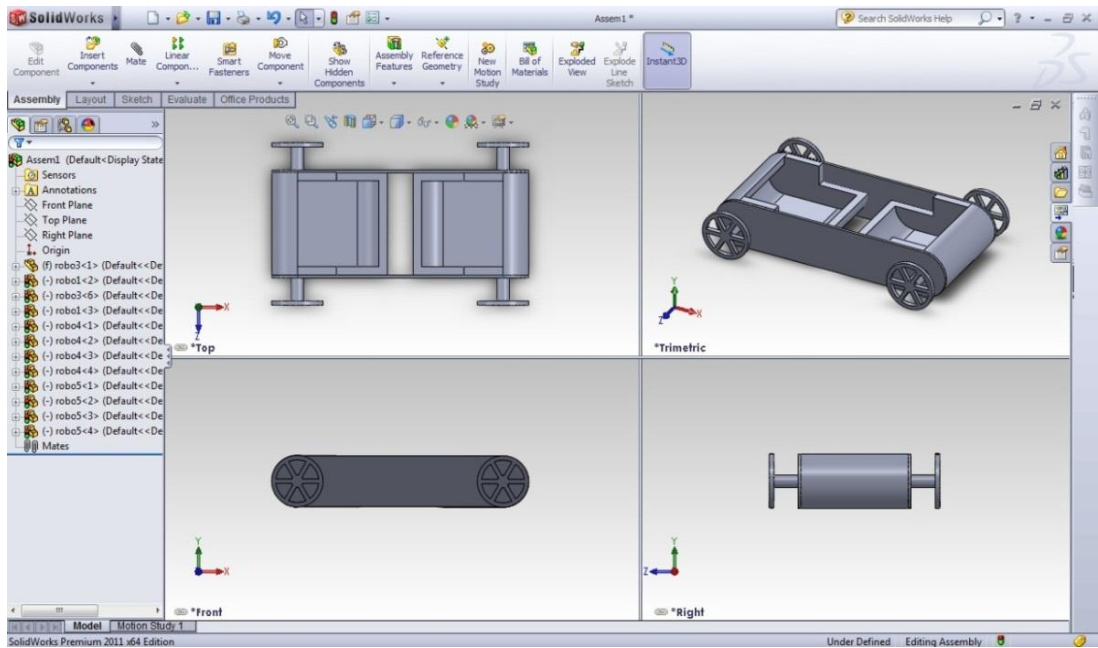
#### 3.2.1 ส่วนของฮาร์ดแวร์

##### 1) ขั้นตอนการสร้างหุ่น

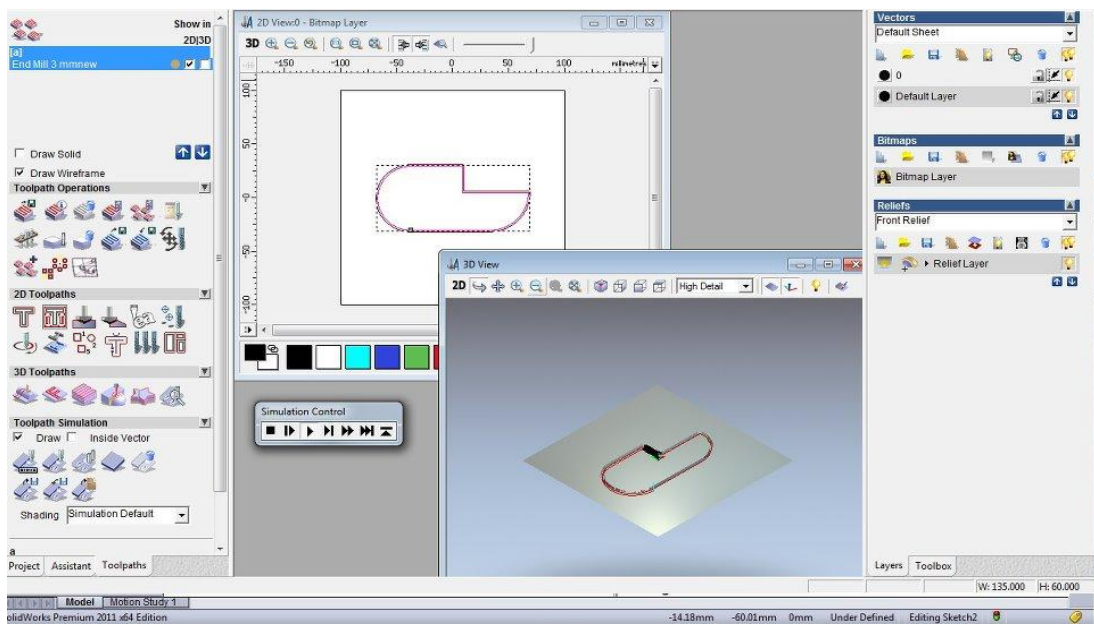
โครงสร้างหลักของตัวหุ่นยนต์ออกแบบสร้างจากอะคริลิก 1 มิลลิเมตร เพื่อให้มีน้ำหนักเบา ระบบสัญญาณใช้ มอเตอร์ 3 โบริต 2 ชุด ใช้เซอร์โวมอเตอร์ ต่อเข้ากับล้อ 4 ล้อ ทำให้สามารถเคลื่อนที่บนพื้นราบและเปลี่ยนเป็นเคลื่อนที่ บนผนังเรียบได้ ส่วนแขนกลลดกระดาศ ใช้ Servo Motor ขนาดเล็ก 1 ตัวควบคุมการยกแขนและลดกระดาศ หุ่นยนต์มีขนาด กว้าง 30 cm, ยาว 30 cm, สูง 13 cm โดยหุ่นยนต์จะมีน้ำหนัก 890 กรัม ออกแบบหุ่นโดยใช้ Solid Work เขียนแบบ และส่งไฟล์ไปยัง ArtCAM เพื่อเป็น G-Code และนำไปใช้ในเครื่องกัด CNC ทำเป็นโครงหุ่น



รูปที่ 3.1 เครื่อง CNC ใช้กัดชิ้นงาน โปรเจก



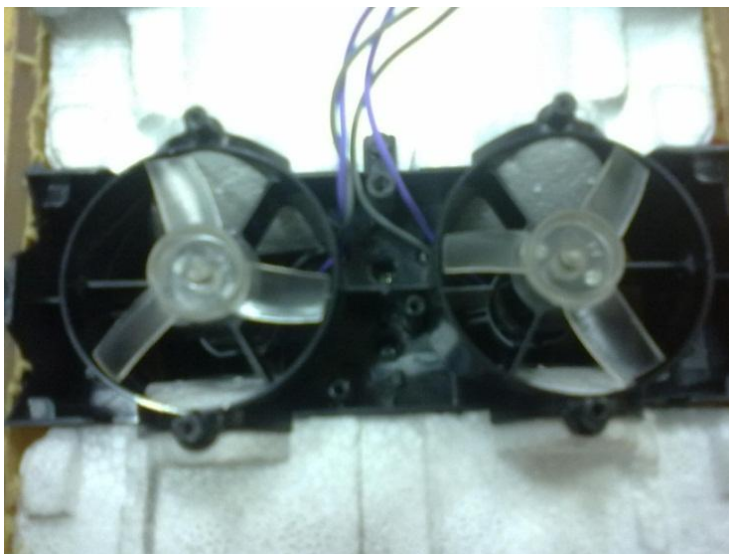
รูปที่ 3.2 แบบหุ่นยนต์โดย Solid Work



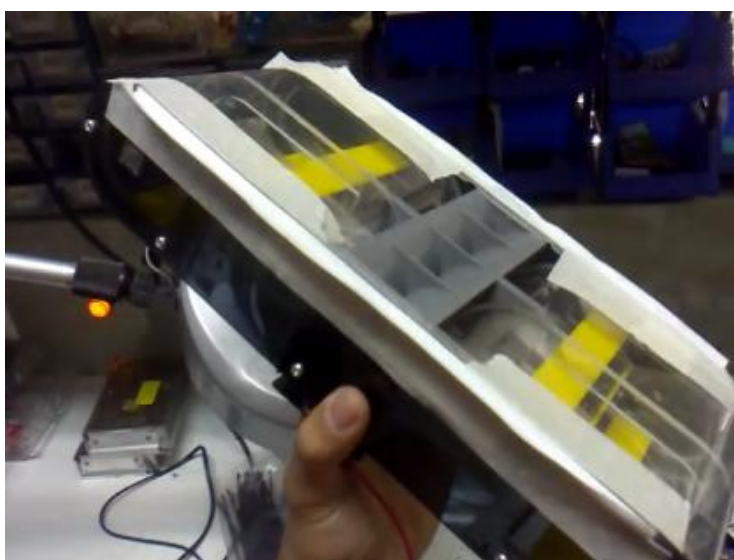
รูปที่ 3.3 ใช้ ArtCAM ในการสั่งงานให้เครื่อง CNC กัดชิ้นงาน

## 2) การออกแบบตัวดูดอากาศ

ตัวดูดอากาศของหุ่นยนต์ใช้ใบพัด 3 แฉก 2 ตัว ที่ซื้อมาจากสะพานเหล็ก และนำ  
ผ้ามาปิดด้านล่างเพื่อให้เกิดสุญญากาศเพื่อให้รถดูดติดกับผนังดังรูปที่ 3.4 และ รูปที่ 3.5



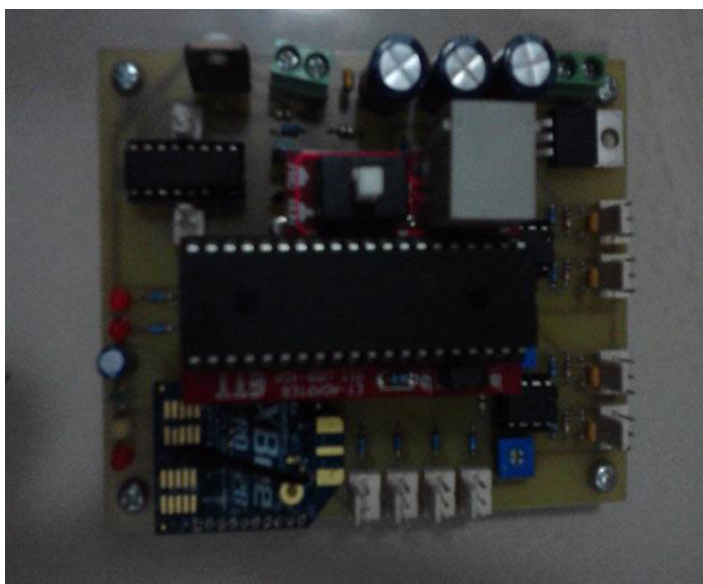
รูปที่ 3.4 ใบพัด 3 ใบพัด 2 ตัว



รูปที่ 3.5 ด้านล่างของหุ่น

### 3) บอร์ด สวิตซ์และตัวส่งสัญญาณ

ใช้บอร์ด PIC 16F887 ตามรูปที่ 3.6 ในการควบคุมหุ่นยนต์ให้เคลื่อนที่และในบอร์ดนี้มีตัวบอกสถานะของหุ่นยนต์ว่าอยู่ในสถานะใดๆ ซึ่งจากรูปที่ 3.7 เรียงจากขวาไปซ้ายคือ ไฟแดงแรกคือไฟเข้าบอร์ด, ไฟเหลืองคือ รับสัญญาณ ไฟสีเขียวคือส่งสัญญาณ และ ไฟแดง 2 อันสุดท้ายคือ โหมดลบกระดานและวิ่งพื้นตามลำดับ



รูปที่ 3.6 PIC 16F887 และ XBee



รูปที่ 3.7 ไฟแสดงสถานะ

ส่วนกล่องส่งสัญญาณตามรูปที่ 3.8 จะมี XBee ซึ่งเป็นตัวส่งทำหน้าที่ส่งสัญญาณไปหา XBee อีกตัวที่เป็นตัวรับอยู่บนบอร์ดของหุ่น ส่วนรูปที่ 3.9 เป็นรูปสวิตซ์ 5 แอมป์ที่ใช้กับหุ่นยนต์ตัวนี้



รูปที่ 3.8 กล่องส่งสัญญาณ



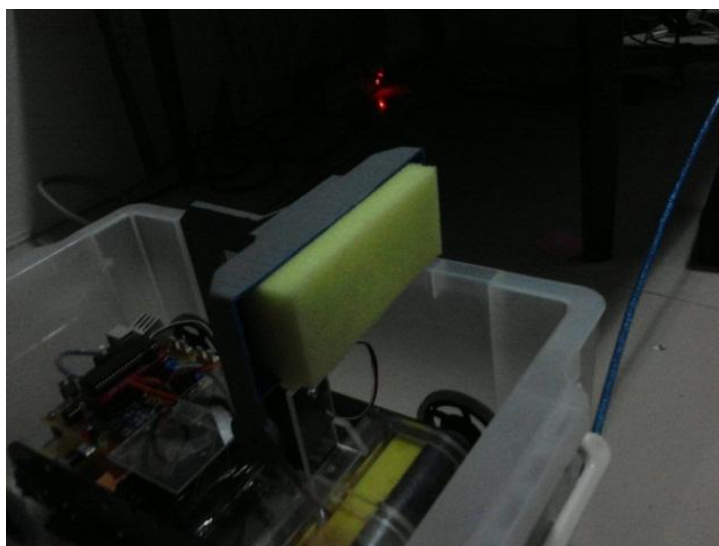
รูปที่ 3.9 กล่องสวิตซ์ 5 แอมป์

#### 4) ล้อและส่วนของที่ลบบกระดาน

ล้อที่ติดกับเซอร์โวมอเตอร์นั้นเป็นพลาสติกเบาหุ้มด้วยยางที่ทำจากซิลิโคนดังรูปที่ 3.10 ซึ่งมีทั้งหมด 4 ล้อ ส่วนที่ลบบกระดานนั้นเราจะใช้แกนกลที่ติดฟองน้ำล้างจานแทนการใช้ผ้า เพราะฟองน้ำล้างจานนั้นลบรอยหมึกออกได้ง่ายกว่าซึ่งส่วนของที่ลบบกระดานจะเป็นดังรูปที่ 3.11

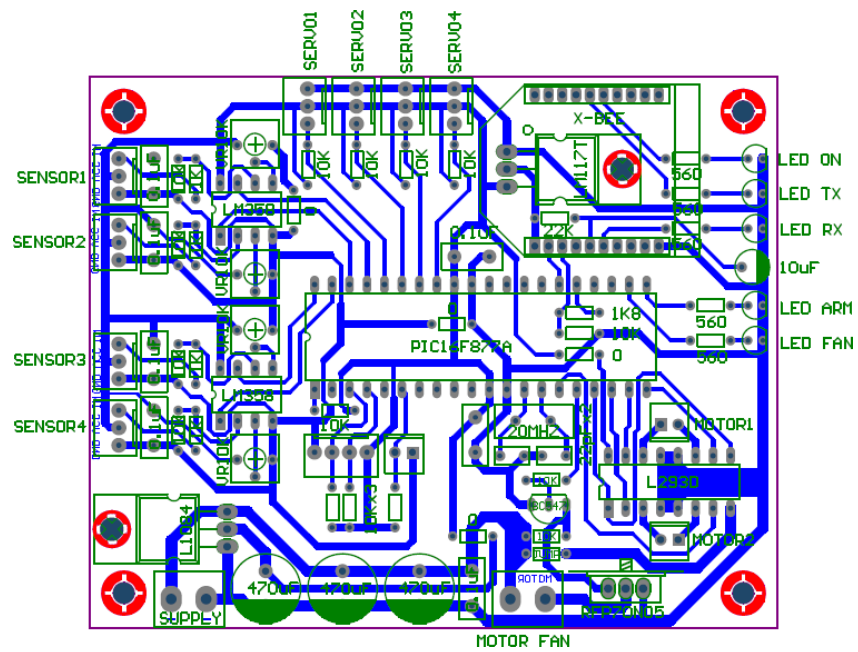
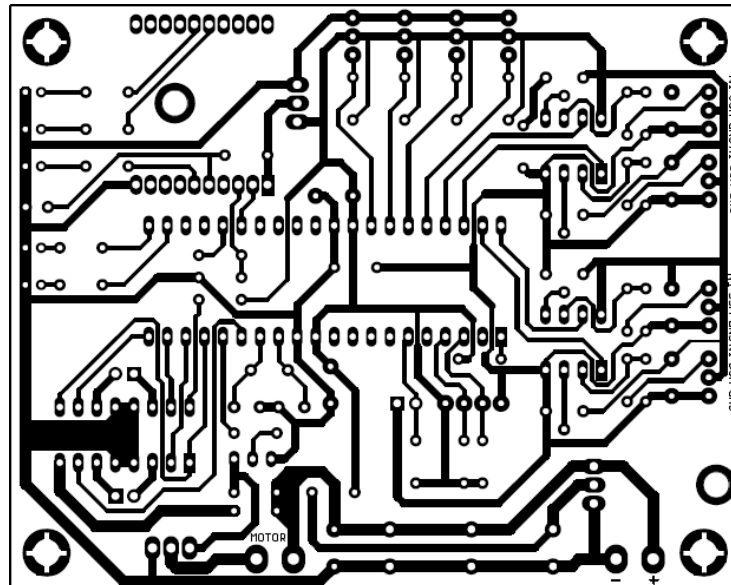


รูปที่ 3.10 ล้อของหุ่นยนต์



รูปที่ 3.11 แกนกลและที่ลบบกระดาน

5) ลายวงจรหุ่นยนต์



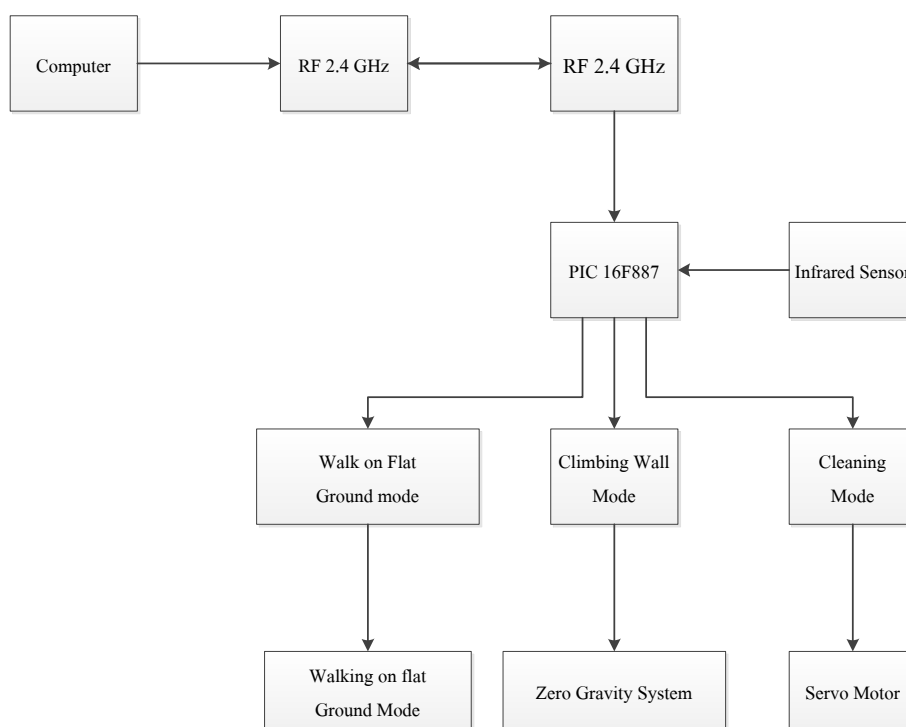
รูปที่ 3.12 ลายวงจรบอร์ด

6) ติดตั้งเซ็นเซอร์เพื่อให้หุ่นสามารถทำงานอัตโนมัติได้



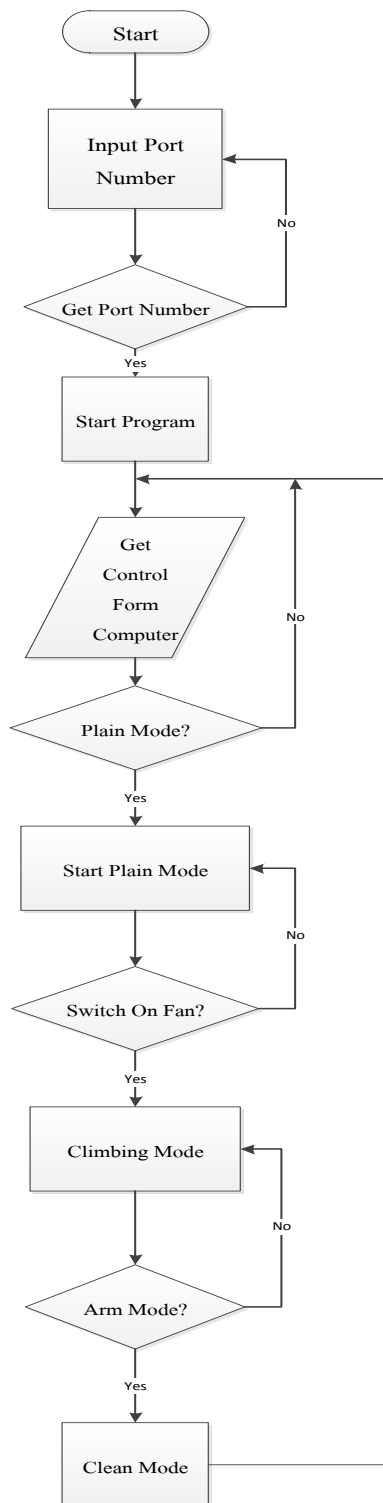
รูปที่ 3.13 ติดตั้งเซ็นเซอร์ไว้ที่ปลายแปลงลบกระดาน

### 3.2.2 Block Diagram



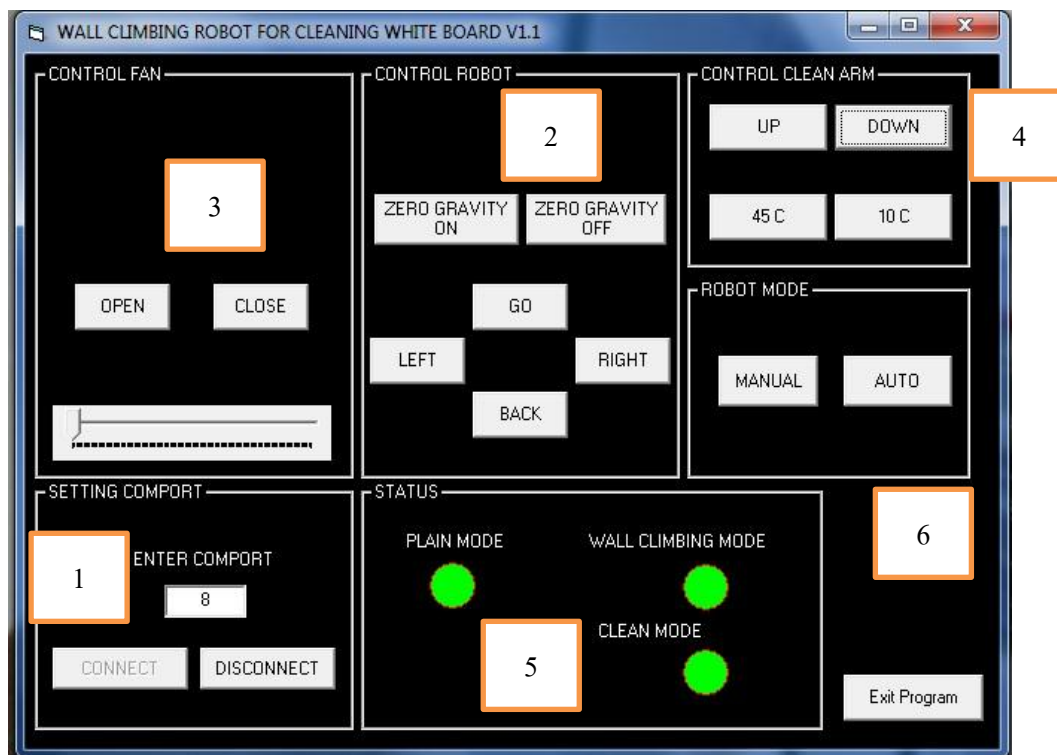
รูปที่ 3.14 Block Diagram ของหุ่นยนต์ใต้น้ำ

## 3.2.3 Flow Chart



รูปที่ 3.15 Flow Chart ของหุ่นยนต์

### 3.2.4 ตัวโปรแกรมควบคุมหุ่น



รูปที่ 3.16 Interface ของโปรแกรมควบคุมหุ่นยนต์

จากรูปที่ 3.16 เป็นรูป Interface ของโปรแกรมควบคุมหุ่นยนต์ โดยโปรแกรมมีการทำงานตามขั้นตอนตามเลขดังต่อไปนี้

1) ก่อนจะใช้โปรแกรมจะต้องนำ USB To Serial มาเสียบคอมพิวเตอร์และส่งสัญญาณ จากนั้นจึงหาเลข Comport ปรากฏเพื่อเริ่มโปรแกรม

2) ส่วนของ Control Robot จะมีปุ่มดังนี้

- ปุ่ม เดินหน้า, ถอยหลัง, เลี้ยวซ้ายและเลี้ยวขวา
- Zero Gravity เป็นปุ่มที่จะทำให้ลดการลื่นไถลของหุ่นเมื่อไต่ผนังและไม่ได้เดิน

3) ส่วนของ Control Fan ปุ่มเปิดกับปิดพัดลมดูดและ Scroll Bar ข้างล่างนั้นคือควบคุมการหมุนของพัดลม

4) ส่วนของ Control Clean Arm คือส่วนที่ควบคุมแขนกลเพื่อที่จะให้แขนกลที่ติดที่ลบกระดานทำงาน โดยแขนกลสามารถปรับได้ 4 ระดับคือ ปุ่ม Down คือระดับระนาบ, ปุ่ม 10 องศา, ปุ่ม 45 องศาและปุ่ม Up คือ 90 องศา

5) Status เป็นส่วนที่สร้างขึ้นเพื่อจะได้ทราบว่าหุ่นยนต์อยู่ในสถานะใดมี 3 สถานะคือ เดินบนพื้นราบ, ใต้ผืนง, โหมดลบกกระดาน โดยไฟสถานะจะมีสี 2 สี คือ สีแดงคือ ปิดระบบการทำงานและสีเขียวคือเปิดระบบการทำงาน

6) Robot Mode จะมีปุ่มกด 2 ปุ่มซึ่งจะทำให้หุ่นทำงานอย่างอัตโนมัติหรือเป็นแบบควบคุมด้วยมือ

### 3.3 ขั้นตอนการสร้าง / ขั้นตอนการดำเนินงาน

3.3.1 ศึกษาทำงานของหุ่นยนต์ใต้ผืนง เช่นวิธีบังคับหุ่นยนต์การเขียนโปรแกรม จะเลือกใช้ภาษาใดเขียน จะมีล้อ หรือ การจะเดินด้วยขา เป็นต้นซึ่งในขั้นตอนนี้ถือเป็นขั้นตอนที่สำคัญในขั้นตอนหนึ่งในการสร้างหุ่นยนต์ใต้ผืนง

3.3.2 วางแผนและออกแบบ เมื่อทราบวัตถุประสงค์ของการสร้างหุ่นยนต์ใต้ผืนงที่เราต้องการสร้างแล้ว ขั้นตอนนี้จะเป็นขั้นตอนในการวางแผนงาน และ การออกแบบ อาจจะวาดโครงสร้างแบบคร่าวๆในกระดาษก่อนขนาดของตัวหุ่นยนต์ขนาดเท่าไร อุปกรณ์ที่ใช้มีอะไรบ้าง

3.3.3 เตรียมเครื่องมือและอุปกรณ์สำหรับในขั้นตอนนี้เป็นการเตรียมเครื่องมือ และ วัสดุ อุปกรณ์สำหรับในการสร้างหุ่นยนต์ รวมทั้งเครื่องคอมพิวเตอร์ บอร์ดทดลองต่างๆ โปรแกรมที่ใช้ในการเขียน

3.3.4 ลงมือสร้างและประกอบส่วนต่างๆ สำหรับขั้นตอนนี้ จะเป็นขั้นตอนที่อาจจะใช้เวลาในการทำงานมากสักหน่อย เพราะต้องลงมือปฏิบัติจริง ดังนั้นทักษะต่างๆมักจะเกิดขึ้นในส่วนนี้เป็นส่วนใหญ่ ไม่ว่าจะเป็นการเขียนโปรแกรม, การประกอบวงจร, การบัดกรี, การใช้คีมตัดต่างๆ

3.3.5 ทดสอบการทำงานของหุ่นยนต์ใต้ผืนงเรียบ ขั้นตอนนี้จะเป็นขั้นตอนที่เรียกได้ว่า หุ่นยนต์ใต้ผืนงเรียบที่เราออกแบบมานั้นจะทำงานได้ดีขนาดไหนในลักษณะผนังต่างๆมีอะไรผิดพลาดบ้างหรือไม่ในการทดสอบสิ่งที่ต้องระวังส่วนมากจะเป็นเรื่องการต่อแหล่งจ่าย ผิดกันเป็นส่วนใหญ่ เช่น สลับขั้ว สายไฟหรือใช้แหล่งจ่ายไม่ตรงกับที่หุ่นยนต์และ ควรมีอุปกรณ์รองรับหุ่นยนต์ขณะใต้ผืนง เพื่อป้องกันความเสียหายจากการตกลงมากระแทก

3.3.6 ปรับแต่งและแก้ไขปัญหาที่เกิดขึ้น ขั้นตอนนี้จะเป็นขั้นตอนในการแก้ไขปัญหาที่เกิดขึ้น การแก้ปัญหาเมื่อหุ่นยนต์ไม่ทำงานหรือทำงานผิดพลาดหรือแก้ไขโปรแกรมเพื่อให้เกิดความถูกต้อง และเป็นไปตามแผนที่วางไว้

### 3.4 วิธีการทดสอบ / วิธีการวัดผล

งานวิจัยนี้เป็นการออกแบบหุ่นยนต์ให้เคลื่อนที่บนผนังเรียบในองศาต่างๆที่กำหนดและเปรียบเทียบค่าระยะทางในการเคลื่อนที่ซึ่งจะกำหนดไว้คงที่ต่อเวลาที่ใช้ในการเคลื่อนที่บนพื้นผิวต่างๆ ได้แก่ กระจก ไม้แผ่นเรียบ และเหล็กแผ่นเรียบในองศาต่างกัน โดยจะแบ่งการทดสอบออกเป็น หัวข้อหลักๆด้วยกันคือ

- 3.4.1 การทดสอบแรงดูดของหุ่นที่กระทำต่อผนัง
- 3.4.2 การทดสอบการเคลื่อนที่บนผนัง
- 3.4.3 การทดสอบการถ่วงน้ำหนักกับการเปลี่ยนการทำงาน
- 3.4.4 การทดสอบการเปลี่ยนการทำงานจากพื้นราบเป็นผนัง
- 3.4.5 การทดสอบควบคุมแขนกล
- 3.4.6 การทดสอบการเปลี่ยนการทำงานจากพื้นราบเป็นผนังเมื่อใส่อุปกรณ์ครบ
- 3.4.7 การทดสอบการแสดงสถานะของหุ่น
- 3.4.8 การทดสอบการไหลของหุ่น
- 3.4.9 การทดสอบการลบกกระดานขาวแบบควบคุมด้วยมือและอัตโนมัติ

โดยการทดสอบความเร็วของหุ่นยนต์ได้กำหนดค่าแรงเริ่มควบคุมให้หุ่นยนต์เคลื่อนที่ไปในระนาบ 0 องศา กับพื้นก่อนแล้วค่อยๆเพิ่มองศาขึ้นตามลำดับจนถึง 90 องศา ตั้งฉากกับพื้นและทดสอบการรับน้ำหนักของหุ่นยนต์ที่ระนาบ 90 องศา ตั้งฉากกับพื้น โดยจะค่อยๆ เพิ่มน้ำหนักขึ้นครั้งละ 50 กรัมแล้วบังคับให้หุ่นยนต์เดินและเพิ่มน้ำหนักไปเรื่อยๆ จนกว่าหุ่นจะไม่สามารถเคลื่อนที่ได้ หลังจากนั้นจึงเริ่มทดสอบการควบคุมหุ่น โดยผ่านระบบไร้สายและทดสอบระบบการส่งข้อมูลสถานะไปพร้อมๆกันเมื่อสามารถไต่ผนังได้แล้วจึงทดสอบการควบคุมหุ่นบนกระดานขาว โดยจะใช้วิธีควบคุมด้วยมือและทำระบบอัตโนมัติ และจากนั้นจึงนำไปทดสอบการลบกกระดานและเช็คความสะอาดของกระดาน

## บทที่ 4

### ผลการดำเนินงานและการวิเคราะห์

หลังจากที่ได้ดำเนินการตามแผนที่วางไว้ข้างต้นนั้นคือ การออกแบบชิ้นงาน เตรียมอุปกรณ์ และสร้างหุ่นยนต์ไต่ผนังลวดกระดานขาวตามแบบที่วางไว้ซึ่งหลังจากที่เราสร้างหุ่นยนต์ไต่ผนังลวดกระดานขาวแล้ว เราจึงได้ทำการทดสอบหุ่นยนต์ไต่ผนังนี้ตามที่วิธีทดสอบที่ได้กำหนดไว้เป็นข้อๆ เพื่อที่จะทำให้หุ่นยนต์ไต่ผนังลวดกระดานขาวนั้นออกมาสมบูรณ์ที่สุด

#### 4.1 ผลการดำเนินงาน

ในช่วงแรกๆนั้นเราได้ทำการศึกษาเรื่องนิวแมติกส์เพื่อที่จะใช้ในการทำให้หุ่นนั้นสามารถไต่ผนังได้ แต่เมื่อมาดูจากข้อมูลทั้งวิดีโอเกี่ยวกับระบบนิวแมติกส์ที่ทำให้หุ่นยนต์นั้นสามารถไต่ผนังได้ และค่าใช้จ่ายที่จะทำเป็นระบบนิวแมติกส์นั้น พบว่ามีราคาที่สูงและหุ่นนั้นความคล่องตัวและความเร็วของหุ่นที่ทำการเคลื่อนที่ไปบนผนังนั้นช้าเกินไป จึงได้หาข้อมูลเพิ่มเติมจนได้วิธีการทำระบบสุญญากาศ ซึ่งระบบสุญญากาศนี้ทำให้หุ่นไต่ผนังนั้นมีความคล่องตัวมากกว่าระบบนิวแมติกส์ มีความเร็วในการไต่ผนังมากกว่านิวแมติกส์และราคาก็ใกล้เคียงกัน ดังนั้นหุ่นจึงใช้ระบบสุญญากาศแทนที่จะใช้ระบบนิวแมติกส์ ในช่วงการลงมือทำหุ่นยนต์ไต่ผนังนั้น วัสดุ อุปกรณ์ และเครื่องมือการทำหุ่นนั้นทางกลุ่มของเราได้รับการช่วยเหลือจากพิพิธภัณฑ์วิทยาศาสตร์ คลอง 6 ปทุมธานี ทำให้การทำโครงหุ่น และใบพัดลม 3 แฉก 2 ตัวนั้นเป็นไปด้วยความราบรื่นเมื่อนำมาประกอบกันจะได้โครงหุ่นยนต์ดังรูปที่ 4.1 และทำการทดสอบดังนี้



รูปที่ 4.1 โครงประกอบของหุ่นไต่ผนัง

### 1) การทดสอบแรงดูดของหุ่นที่กระทำต่อผนัง

ซึ่งเมื่อเราประกอบหุ่นเข้ากับพัลลคมดูดเสร็จแล้วเราจึงนำไปทดสอบการดูดของพัลลคมว่าสามารถเกาะติดกับกำแพงหรือกระจกได้หรือไม่ซึ่งตอนแรกเมื่อหุ่นเกาะแล้วพบว่าสามารถเกาะติดได้ตามรูปที่ 4.2



รูปที่ 4.2 ทดสอบการเกาะติดกำแพง

### 2) ทดสอบการไต่ขึ้นพื้นกระจก

จากนั้นเมื่อติดล้อแล้วจึงนำมาทดสอบการไต่ขึ้นของหุ่นยนต์ พบว่าสามารถไต่ขึ้นได้ตามรูปที่ 4.3 และพบว่าหุ่นมีการไหลลงมาเล็กน้อยเวลาไต่ขึ้น และขึ้นได้ไม่เร็วเท่าไรนัก



รูปที่ 4.3 ทดสอบการไต่ของหุ่นยนต์ไต่ผนัง

แรงเสียดทานของล้อ 4 อันนั้นมีน้อยเกินไปหุ่นเกิดการไถลวิ่งไม่ตรงเส้นทางดังนั้นจึงได้เพิ่มอีกข้างละ 1 ล้อ เป็นทั้งหมด 6 ล้อดังรูปที่ 4.4 แต่ ก็ไม่สามารถทำให้ปัญหาหมดไปได้ จึงได้ทำระบบที่ให้เซอร์โวมอเตอร์นั้นวิ่งด้านแรงโน้มถ่วง ผลปรากฏว่าทำให้หุ่นสามารถเกาะติดกับผนังและกระดานได้ดีขึ้น และลดการไถลของหุ่นได้อย่างมาก



รูปที่ 4.4 ปรับแต่งล้อของรถไต่ผนัง

### 3) การทดสอบการถ่วงน้ำหนักกับการเปลี่ยนการทำงาน

จากการทดสอบการไต่ผนังของหุ่นทำให้เราทราบว่าหุ่นย่นคั้นนั้นอาจไต่ขึ้นผนังไม่ได้เพราะถ้าหัวหุ่นมีน้ำหนักมากกว่าท้ายหุ่น หุ่นจะขึ้นได้ยากรูปที่ 4.5



รูปที่ 4.5 การทดสอบการถ่วงน้ำหนักกับการเปลี่ยนการทำงาน

และได้ทำตารางสรุปการถ่วงน้ำหนักไว้ตามตารางที่ 4.1

ตารางที่ 4.1 ตารางแสดงผลการถ่วงน้ำหนักกับหุ่นยนต์ใต้อุปกรณ์ยกกระดาน

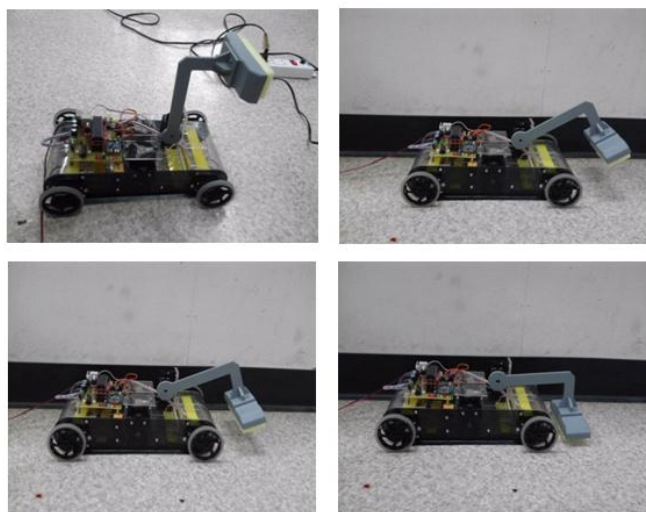
น้ำหนัก	ผล
100	สามารถขึ้นได้
150	สามารถขึ้นได้
200	สามารถขึ้นได้
250	สามารถขึ้นได้
300	สามารถขึ้นได้
350	ไม่สามารถขึ้นได้

#### 4) การทดสอบการเปลี่ยนการทำงานจากพื้นราบเป็นผนัง

จากการทดสอบให้หุ่นยนต์ทำงานจากพื้นราบเป็นทำงานบนผนังนั้นราบรื่นดีไม่มีปัญหาแต่อย่างใด อาจมีปัญหาคือสล็อตช่วงที่ยังไม่ใช้เซอร์โวมอเตอร์ ซึ่งตอนที่ใช้มอเตอร์เฟืองทคนั้นเวลาขึ้นแรงส่งจะน้อยกว่าตอนใส่เซอร์โวมอเตอร์และปัญหาเรื่องการถ่วงน้ำหนักโดยถ้าหากน้ำหนักของหุ่นมีมากเกินไปจะทำให้หุ่นหงายหลังได้เมื่อเวลาเป็นโหมคจากพื้นราบเป็นใต้อุปกรณ์ ดังนั้นจึงต้องออกแบบโดยคำนึงถึงน้ำหนักเป็นหลัก

#### 5) การทดสอบควบคุมแขนกล

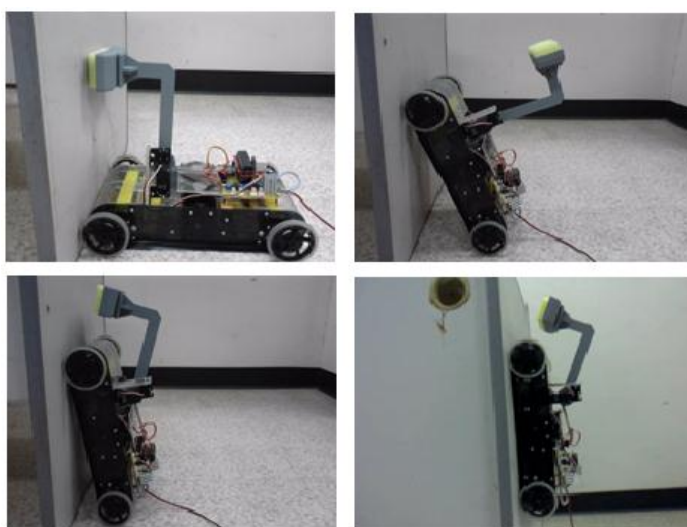
ผลออกมาเป็นไปได้ด้วยดีสามารถควบคุมแขนกลโดยการควบคุมการหมุนของเซอร์โวมอเตอร์ที่แขนกล โดยแขนกลสามารถปรับระดับได้ 4 ระดับคือ 90 องศา, 45 องศา, 10 องศาและ 0 องศา โดยการควบคุมด้วยมือเป็นหลัก ตามรูปที่ 4.6



รูปที่ 4.6 แขนกลทั้ง 4 ระดับ

6) การทดสอบการเปลี่ยนการทำงานจากพื้นราบเป็นผนังเมื่อใส่อุปกรณ์ครบ

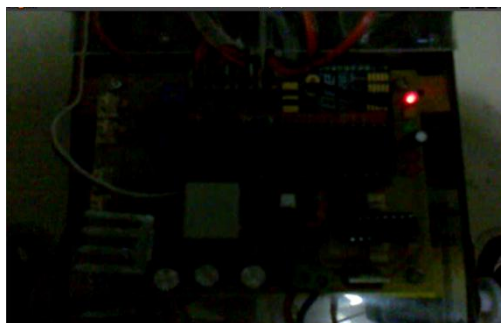
จากการทดสอบพบว่าในช่วงแรกนั้นตอนแรกที่ไม่ได้เขียนโปรแกรมควบคุมแขนกลนั้น เมื่อเปลี่ยนโหมดจากพื้นราบเป็นใต้ผนัง จะเกิดปัญหา เพราะน้ำหนักด้านบนของหุ่นเพิ่มมากขึ้นทำให้หุ่นเวลาจะเปลี่ยนโหมดไปใต้ผนังแล้วจะหงายหลังจนคิดว่า ดังนั้นจึงต้องเขียนโปรแกรมควบคุมแขนกลโดยการก่อนที่จะขึ้นจะทำการปรับองศาของแปรงลบกระดานก่อนที่จะทำการเปลี่ยนโหมด จึงทำให้ปัญหาเรื่องน้ำหนักของแขนกลนั้นหายไปตามรูปที่ 4.7



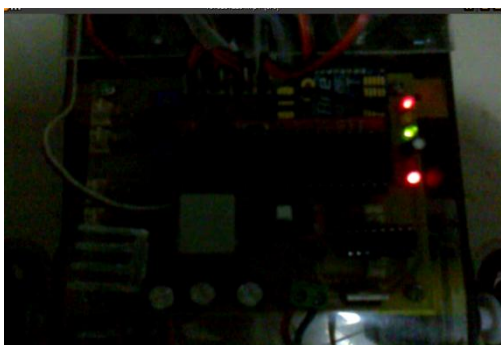
รูปที่ 4.7 ทดสอบการเปลี่ยนโหมดเมื่ออุปกรณ์ครบ

### 7) การทดสอบการแสดงผลสถานะของหุ่น

สามารถแสดงผลสถานะบอกการทำงานของหุ่นได้อย่างถูกต้องและรวดเร็วไม่มีข้อผิดพลาดใดๆ โดยรูปจะแสดงผลสถานะโดยไฟเขียวคือไฟแสดงรับสัญญาณจากคอม และไฟแสดงสถานะจะเป็นสีแดงดังเช่นรูปที่ 4.8 ถึงรูปที่ 4.10



รูปที่ 4.8 XBee มีไฟเข้า เข้าสู่โหมดพื้นราบ



รูปที่ 4.9 โหมดไต่ผนังไฟสีแดงดวงล่างติด



รูปที่ 4.10 โหมดหลบกระดานไฟที่ 2 จากข้างล่างติด

ระยะที่ XBee สามารถทำงานได้นั้นมีระยะที่ไกลมากซึ่งในคู่มือบอกไว้ว่าสามารถส่งสัญญาณได้ไกล 1.5 กิโลเมตร ซึ่งถ้ามีสิ่งกีดขวางก็จะลดระยะลงไปตามลำดับซึ่งทางเราได้ทดสอบความไกลของการส่งสัญญาณพบว่าสามารถควบคุมจากชั้น 1 ไปยังชั้น 2 ได้

#### 8) การทดสอบการไหลของหุ่น

ในช่วงที่ยังไม่มีระบบ Zero Gravity System หุ่นเกิดการไหลอย่างเห็นได้ชัด บางทีหุ่นวิ่งอยู่หุ่นก็จะไหลลงมาเรื่อยๆ ดังนั้นทางเราจึงได้เพิ่มปุ่มที่ทำการควบคุมเซอร์โวมอเตอร์ให้วิ่งอยู่ตลอดเวลาอย่างช้าๆ จึงทำให้ไม่ว่าจะวิ่งหรือหยุดนิ่งการไหลของหุ่นก็จะลดน้อยลงทำให้เมื่ออยู่โหมคได้ผนังจึงมีประสิทธิภาพมากขึ้น

#### 9) การทดสอบการลบกระดาน

สามารถลบกระดานได้จริง แต่ไม่สะดวกมากนักเพราะแรงกดของแปรงลบกระดานกับกระดานนั้นน้อยมากทำให้ต้องอาศัยการเคลื่อนที่ไปๆมาๆในเวลานาน ดังนั้นการควบคุมหุ่นอย่างชำนาญที่จะทำให้ผลออกมาดีที่สุด ส่วนระบบอัตโนมัติเราจะนำเทปดำมาปิดรอบกระดาน และปล่อยให้เดินไปโดยให้คำสั่งที่ว่าถ้าเซ็นเซอร์หุ่นเจอเทปดำที่ติดไว้ให้มันเดินถอยหลังแล้วเลี้ยว ปรากฏว่าเวลาที่เรเขียนปากกาคำ เรื่องความสะอาดในการลบกระดานนั้นเมื่อเรานำผ้าของแปรงลบกระดานมาติดใต้ห้องหุ่นจะได้ผลดังรูปที่ 4.11



รูปที่ 4.11 ทดสอบการลบกระดาน

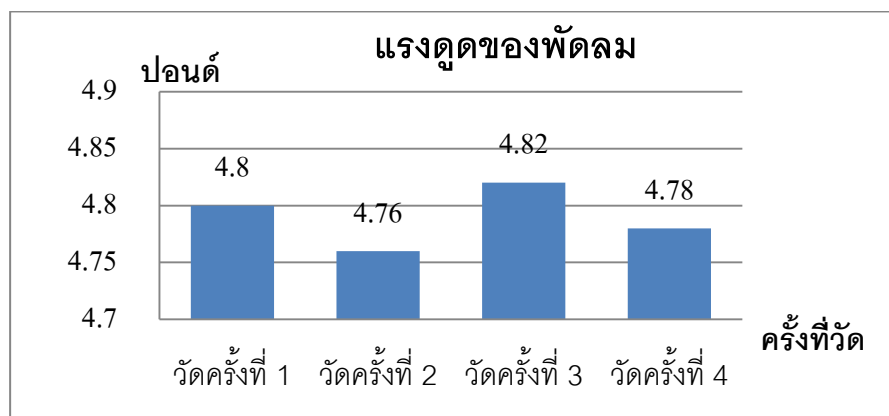
## 4.2 การวิเคราะห์ / การวิจารณ์

จากการวิเคราะห์หุ่นยนต์ไต่ผนังลงกระดาศขาที่ได้ทำขึ้นมานั้นพบว่าหุ่นนั้นต้องมีการแก้ปัญหาซึ่งมีปัญหาคือเป็นข้อๆดังนี้

1) ตอนช่วงเลือกวัสดุตอนแรกได้ลองใช้อะลูมิเนียมซึ่งได้พบว่าหุ่นมีน้ำหนักมากจึงได้แก้ปัญหาโดยการใส่อะคริลิกแทนเพราะทำให้หุ่นมีน้ำหนักแค่ 890 กรัม จากที่ใช้อะลูมิเนียมจะหนัก 1,800 กรัม ซึ่งลดน้ำหนักได้เกินครึ่งนึงแต่อะคริลิกนั้นราคาสูงมาก ในโปรเจกต์หุ่นยนต์ไต่ผนังลงกระดานนั้นน้ำหนักถือว่าเป็นสิ่งสำคัญมาก เพราะอาจทำให้หุ่นยนต์ไต่ขึ้นผนังไม่ได้ถ้าหากน้ำหนักมีมากเกินไป

2) ในช่วงแรกของการทดลองการยึดติดกับผนังโดยที่ไม่มีล้อซึ่งเราได้เห็นว่าเกาะได้แต่หุ่นยนต์นั้นมันค่อยๆ ไถลงมาเรื่อยๆ เพราะว่าแรงดูดพัลลมที่ดูดติดกับผนังนั้นยังน้อยจึงต้องทำการเพิ่มแรงดูดของหุ่นเมื่อได้แรงดูดที่พอเหมาะจึงได้ทำการวัดแรงดูดของหุ่นซึ่งได้ผลออกมาดังกราฟที่ 4.1

กราฟที่ 4.1 แรงดูดของพัลลมที่วัดได้



3) มีปัญหาที่ว่าจะเอาน้ำหนักถ่วงไว้ตำแหน่งในของหุ่นยนต์ ซึ่งต้องมาทดสอบหาจุดไหนที่สามารถถ่วงแล้วทำให้หุ่นไต่ผนังได้ เพราะต้องเอาบอร์ด PIC 16F887 วางบนหุ่นยนต์อีกทั้งตอนหลังต้องเอาแกนกลที่ติดตั้งเซอร์โวมอเตอร์อีก ตัวมาติดตั้งที่หุ่น ดังนั้นน้ำหนักของหุ่นจึงเป็นเรื่องสำคัญ

4) การติดต่อไร้สายนั้นค่อนข้างไม่ลำบากเท่าไรนักเพราะอุปกรณ์ Xbee นั้นมีการตั้งค่าการส่งข้อมูลจากหุ่นไปยังคอมพิวเตอร์ แต่ราคาค่อนข้างที่จะแพงฉะนั้นถ้าใช้อย่างอื่นได้ก็คิดแต่ที่เลือกใช้อันนี้เพราะง่ายและสะดวก

5) แรงเสียดทานของล้อหุ่นกับกระดานขาวนั้นสำคัญมากในช่วงแรกๆนั้นหุ่นเดินไปแล้วปรากฏว่าหุ่นจะไหลและหลุดออกจากเส้นทางซึ่งอาจเป็นเพราะสิ้น ทางกลุ่มจึงแก้ปัญหาด้วยการเปลี่ยน เซอร์โวมอเตอร์ให้กับล้อทั้ง 4 ข้างและเพิ่ม Mode Zero Gravity System คือการให้เซอร์โวมอเตอร์นั้นทำการหมุนล้อทำทิศตรงข้ามกับแรง โน้มถ่วงอย่างช้าๆเมื่อหุ่นวิ่งหรืออยู่นิ่ง เพื่อเพิ่มแรงให้หุ่นเกิดการไถลน้อยลง

6) เนื่องจากพื้นกระดานลื่นมากแม้ขณะควบคุมด้วยมืออาจมีปัญหาเรื่องการไถลของหุ่นเป็นบางครั้ง เพราะฉะนั้นการที่หุ่นเป็นแบบอัตโนมัตินั้นค่อนข้างที่ยากและเสียเวลามากมายในการทำให้หุ่นให้ระบบเสถียร

7) เมื่อทดสอบระบบอัตโนมัติของหุ่นยนต์ เราได้นำเทปกาวยึดไว้รอบๆขอบกระดานเพื่อหุ่นจะได้ทราบว่าสุดขอบกระดานขาวจากนั้นจึงปล่อยหุ่นให้วิ่งไปโดยอัตโนมัติบนกระดาน ซึ่งผลปรากฏว่าหุ่นสามารถไต่กระดานขาวขึ้นได้ แต่มีปัญหาด้านการทรงตัวเพราะเวลาหุ่นหมุนแล้วหุ่นจะเกิดการไถลและหุ่นจะวิ่งตกขอบล่างไป และเมื่อใช้ปากกาเขียนกระดานขาวที่เป็นสีดำหุ่นยนต์จะถอยกลับแล้วเลี้ยวทันที

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

โครงการวิจัยนี้เป็นการคิดที่อยากจะพัฒนาเกี่ยวกับหุ่นยนต์ในรูปแบบที่แตกต่างออกไปซึ่งหุ่นยนต์ใต้น้ำลอยกระดานขาวตัวนี้เป็นหุ่นที่สั่งการผ่านทางคอมพิวเตอร์ในระบบไร้สายโดยสามารถบอกสถานะของหุ่นว่าอยู่ในสถานะใด และควบคุมการลบกกระดานขาวของหุ่นยนต์ทั้งควบคุมด้วยมือและระบบอัตโนมัติ

#### 5.1 สรุป

จากโครงการที่ได้จัดทำขึ้นนั้น ได้เริ่มจากการศึกษาและรวบรวมข้อมูล หลายที่ทั้งในอินเทอร์เน็ต และสอบถามจากผู้ที่มีความรู้ในด้านหุ่นยนต์ออกมาในหลายลักษณะ ทำให้มีหุ่นยนต์หุ่นยนต์ออกมาในหลายลักษณะหลายทฤษฎีที่แตกต่างกันไป สุดท้ายแล้วจากการศึกษาทฤษฎี Zero Gravity Traction Technology น่าสนใจมากที่สุด และได้ทำการจัดทำขึ้น และมีผลที่ได้รับจากโครงการ คือ การสร้างหุ่นยนต์ใต้น้ำ กว้าง 12.5 cm, ยาว 30 cm, สูง 13 cm และสามารถสร้างโปรแกรมที่สามารถควบคุมการทำงานของหุ่นยนต์ใต้น้ำ เพื่อให้ควบคุมหุ่นยนต์ใต้น้ำและสามารถลบกกระดานขาวได้สะดวกและถูกต้อง และได้ทราบถึงหลักการเกี่ยวกับแรงเสียดทานและแรงโน้มถ่วงซึ่งมีผลต่อหุ่นยนต์ใต้น้ำลบกกระดานขาว นั่นคือถ้าแรงเสียดทานมีน้อยจนเกินไปหุ่นอาจจะตกลงมาได้และถ้าน้ำหนักหุ่นมากก็จะไต่ไม่ขึ้น ดังนั้นการใช้งานจึงต้องออกแบบให้หุ่นเบาที่สุดและล้อของหุ่นยนต์ใต้น้ำจะต้องมีแรงเสียดทานให้มากพอที่จะเกาะกับกระดานได้ ในการทำงานนั้นก็ได้อุปสรรคต่างๆ ในการทำงานซึ่งทางผู้จัดทำได้ทำการแก้ไขตามส่วนที่เกิดข้อผิดพลาดดังนี้ คือ ทำหุ่นให้เบาโดยใช้อะคริลิก เพิ่มจำนวนล้อ เพื่อให้มีแรงเสียดทานเพิ่มขึ้น มีการควบคุมการดูดของพัลลมเพื่อไม่ให้ดูดมากเกินไป ใช้ XBee ในการส่งสัญญาณหากันระหว่างหุ่นและคอมพิวเตอร์ที่ควบคุมหุ่นหรือการนำผ้าแปรงลบกกระดานมาติดได้รถเพื่อเพิ่มแรงดูดของหุ่นยนต์ และส่วนที่เป็นระบบอัตโนมัติทางกลุ่มได้ลองทดสอบแล้วพบว่าสามารถทำงานได้ แต่มีบางปัจจัยทำให้เกิดปัญหา เช่นการเขียนปากกาไวต์บอร์ดด้วยสีดำ และเรื่องการออกแบบหุ่นที่ทำให้ติดเซ็นเซอร์ได้น้อยรวมทั้งปัญหาล้อของรถที่บางล้อไม่ติดกระดาน

## 5.2 ปัญหาที่พบ

- 5.2.1 พบปัญหาเรื่องการใช้วัสดุอุปกรณ์เพราะเนื่องจากต้องการใช้วัสดุที่เบาและแข็งแรง
- 5.2.2 ปัญหาเกี่ยวกับยางที่จะใช้กับหุ่นว่าจะใช้วัสดุชนิดใดในการทำยาง
- 5.2.3 ปัญหาด้านการใช้มอเตอร์ในหุ่นยนต์เพราะมอเตอร์ที่ใช้จะต้องมีแรงบิดที่มากเพื่อให้หุ่นสามารถไต่ขึ้นกำแพง ซึ่งใช้มอเตอร์เฟืองทดแล้วยังเกิดปัญหา
- 5.2.4 ปัญหาเรื่องการควบคุมแรงคูดของพัลลม ถ้าแรงคูดที่มากเกินไปหุ่นจะไม่เดิน ถ้าน้อยไปก็จะไม่มีแรงคูดติดกับผนังและกระดาน
- 5.2.5 ปัญหาเรื่อง IC Regulator เมื่อหุ่นทำงานหนักเกินไปจะทำให้ IC ร้อนและเสีย
- 5.2.6 ปัญหาหุ่นไถลเมื่อไต่ขึ้นผนังและกระดาน
- 5.2.7 การลบกกระดานของหุ่นยนต์โดยใช้แปรงที่คิดแขนกลอาจจะสะอาดไม่พอ

## 5.3 การแก้ปัญหา

- 5.3.1 ในช่วงแรกได้ทดสอบใช้อะลูมิเนียมพบว่าน้ำหนักมากเกินไปจึงเปลี่ยนมาใช้อะคริลิกแทนพบว่าน้ำหนักหุ่นลดลงไปครึ่งนึง แต่ราคาอะคริลิกค่อนข้างที่จะแพง
- 5.3.2 วัสดุที่ทำเป็นยางแล้วพบว่ายางทำจากซิลิโคนนั้นดีที่สุดเพราะมีแรงเสียดทานมากที่สุด
- 5.3.3 แก้ปัญหาโดยใช้เซอร์โวมอเตอร์ เพราะมีความทนทานมากกว่ามอเตอร์เฟืองทดสามารถปรับแต่งการหมุนของเซอร์โวมอเตอร์ได้และแรงขับของเซอร์โวมอเตอร์นั้นมีมาก แต่ราคาของมอเตอร์นั้นอาจจะค่อนข้างแพง
- 5.3.4 เขียนโปรแกรมให้สามารถควบคุมการหมุนของมอเตอร์โดยสามารถปรับความแรงของการหมุนของมอเตอร์ที่ใบพัดติดอยู่ได้ซึ่งอาจจะต้องระวังเรื่องในการควบคุมความเร็วของมอเตอร์เพราะถ้าให้หมุนมากเกินไปมอเตอร์อาจเสียหายได้
- 5.3.5 ติด Heatsink ให้กับ IC Regulator เพื่อป้องกันการเสียหายที่จะเกิดขึ้นกับ IC Regulator เนื่องจากถ้าหากวงจรนี้เสียหาย หุ่นจะดับและอาจทำให้ตกลงมาได้เมื่อเวลาไต่ผนังหรือกระดาน
- 5.3.6 ใช้วิธีการให้ล้อหมุนไปข้างหน้าอย่างช้าๆเพื่อเป็นการเลี้ยงตัวหุ่นยนต์เพื่อไม่ให้ไถลลงมาเกินไป หรือไถลลงทันที
- 5.3.7 นำเอาผ้าวัสดุที่ใช้ในการลบกกระดานมาแปะไว้ใต้ท้องรถเพื่อให้ลบล้างสะอาดมากขึ้น และทำให้รัศมีแรงคูดมากขึ้นเพราะเป็นการลดขนาดของช่องอากาศที่จะทำให้อากาศออกทำให้แรงคูดของหุ่นมากขึ้น

## 5.4 แนวทางการพัฒนาต่อ

5.4.1 อาจจะสามารถเพิ่มฟังก์ชันให้หุ่นสามารถเขียนและลบได้ โดยผ่านคอมพิวเตอร์

5.4.2 เพิ่มเติมโปรแกรมให้กับหุ่นยนต์เพื่อให้ประสิทธิภาพ เช่นระบบ Image Processing เพื่อให้หุ่นยนต์ทำงานได้ถูกต้องมากขึ้น

5.4.3 เปลี่ยนหรือเพิ่มล้อของหุ่นยนต์และเพิ่มความต้านทานให้กับล้อ เพื่อให้หุ่นสามารถเกาะติดกระดานได้ดีขึ้นไม่ลื่นไถล

5.4.4 ระบบอัตโนมัติแบบใช้เซนเซอร์ควรรู้ถึงสีของปากกา และโปรแกรมควรจะสามารถใส่ขนาดของกระดานให้หุ่นทราบได้

5.4.5 ควรทำโซล่อพติดที่ล้อให้กับหุ่นเพื่อให้ล้อของหุ่นสามารถสัมผัสกับผนังแล้วกระดานตลอดเวลา

5.4.6 อาจให้หุ่นใช้แบตเตอรี่แทนการใช้สวิตซ์ แต่ต้องคำนวณการใช้ไฟหุ่นให้หุ่นให้ถูกต้อง เพื่อไม่เกิดปัญหาไฟเลี้ยงไม่พอ

5.4.7 อาจใช้ตัวรับส่งสัญญาณที่เป็นตัวอื่นที่ไม่ใช่ XBee เพราะ XBee มีราคาแพงมาก ถึงแม้จะสามารถส่งสัญญาณได้ถึง 1.5 กิโลเมตร แต่มันก็เกินความจำเป็น

5.4.8 ควรศึกษาระบบสุญญากาศให้มากกว่านี้เพราะจะช่วยในเรื่องการออกแบบหุ่นให้เหมาะสมต่อการทำหุ่นยนต์ไต่ผนัง ซึ่งทางกลุ่มของข้าพเจ้ามีจุดอ่อนเรื่องหุ่นที่มีขนาดเล็กเกินไป การเพิ่มเติมอุปกรณ์ให้หุ่นจึงเป็นไปได้ด้วยความยากลำบาก

## บรรณานุกรม

- [1] Liptak, Bela. 1995. **Instrument Engineers' Handbook: Process Control**. Radnor, Pennsylvania: Chilton Book Company. pp. 20-29.
- [2] มนูญ ชื่นชม. 2540. **นิวแมติกส์ไฟฟ้าเบื้องต้น**, พิมพ์ครั้งที่ 1. กรุงเทพฯ :สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- [3] ร.อ. วาที ปริยพงศ์. 2547. **เส้นทางสู่นักประดิษฐ์หุ่นยนต์**, พิมพ์ครั้งที่ 3. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- [4] จุฑาทิพย์ ถมยา. 2546. **นิวแมติกส์และนิวแมติกส์ไฟฟ้าเบื้องต้น**, พิมพ์ครั้งที่ 1. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท.
- [5] โกวิท ไชวสุวรรณ. **คู่มือนักสร้างหุ่นยนต์**, พิมพ์ครั้งที่ 2 กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท.
- [6] ชวพงษ์ สิงหแพทย. 2538. **การควบคุมอัตโนมัติ**, พิมพ์ครั้งที่ 5. กรุงเทพฯ : ฟิสิกส์เซ็นเตอร์การพิมพ์.
- [7] รศ. วุฒิชัย กปิลกาญจน์. 2536. **กลไกและพลศาสตร์ของเครื่องจักรกล**, พิมพ์ครั้งที่ 4. กรุงเทพฯ : สำนักพิมพ์ฟิสิกส์ เซ็นเตอร์.
- [8] มงคล ทรายพันธ์. 2540. **กลศาสตร์เครื่องจักรกล เล่ม 1**, กรุงเทพฯ : หจก. เม็ดทรายพรินติ้ง.
- [9] จิตาภัส สัมพันธ์สมโภช และคณะ. 2548. **การพัฒนาซอฟต์แวร์ด้วยโปรแกรมสำเร็จรูป**, พิมพ์ครั้งที่ 1. กรุงเทพฯ : สำนักพิมพ์วังอักษร
- [10] ฉัททวุฒิ พิษผล และคณะ. 2544. **คู่มือเรียน Visual Basic 6.0**, พิมพ์ครั้งที่ 5. กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด .
- [11] บุญรักษ์ กาญจนวรวณิชย์. 2551. **อะคริลิก**. [ออนไลน์] เข้าถึงได้จาก : [http://www.mtec.or.th/index.php?option=com\\_content&task=view&id=577&Itemid=36](http://www.mtec.or.th/index.php?option=com_content&task=view&id=577&Itemid=36).
- [12] ช่าง ทมทิตชงศ์. 2544. **แรงเสียดทาน**. [ออนไลน์] เข้าถึงจาก : [http://www.myfirstbrain.com/student\\_view.aspx?ID=75483](http://www.myfirstbrain.com/student_view.aspx?ID=75483).
- [13] Thaeasyelectronic. 2554. **การ Configuration XBee**. [ออนไลน์] เข้าถึงจาก : <http://www.thaeasyelec.com/Embedded-Electronics-Application/Xbee-Basic-Configuration-in-Network-Application.html>.
- [14] วิทยาลัยเทคนิคสุโขทัย. 2550. **PIC คืออะไร**. [ออนไลน์] เข้าถึงจาก : [km.sukhothaitc.ac.th/files/1010190994103\\_10101914140254.pdf](http://km.sukhothaitc.ac.th/files/1010190994103_10101914140254.pdf).

ภาคผนวก ก

**Code Program**

## 1. Source Code ส่วนควบคุมหุ่นยนต์บน PIC

```

#include <16F887.h>

#device adc=8

#FUSES NOWDT           //Watch Dog Timer
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPUT          //No Power Up Timer
#FUSES NOMCLR         //Master Clear pin enabled
#FUSES PROTECT        //Code protected from reads
#FUSES NOCPD          //No EE protection
#FUSES NOBROWNOUT     //No brownout reset
#FUSES IESO           //Internal External Switch Over mode enabled
#FUSES FCMEN          //Fail-safe clock monitor enabled
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES NOWRT          //Program memory not write protected
#FUSES BORV40         //Brownout reset at 4.0V

#use delay(clock=2000000)

#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7)

#use fast_io(B)

#define LED_ARM(x)    output_bit(pin_d2,x)
#define LED_FAN(x)    output_bit(pin_d3,x)
#define SENSOR()     input(PIN_A4)
#define CONTROL_DIRECTON    '1'
#define CONTROL_FAN        '2'
#define CONTROL_ARM        '3'
#define SET_SPEED_FAN      '4'

#define CMD_FORWARD        '1'

```

```
#define CMD_BACKWARD      '2'
#define CMD_TURNLEFT     '3'
#define CMD_TURNRIGHT    '4'
#define CMD_STOP         '5'
#define CMD_EN_ZR_GRAVITY '6'
#define CMD_DS_ZR_GRAVITY '7'

#define CMD_FAN_OPEN     '1'
#define CMD_FAN_CLOSE   '2'

#define CMD_UP_ARM       '1'
#define CMD_DW_ARM       '2'
#define CMD_CN_ARM       '3'
#define CMD_CN1_ARM      '4'
#define fan(x)           set_pwm2_duty(127-x)

#define servo1 = 0x06.0
#define servo2 = 0x06.1
#define servo3 = 0x06.2
#define servo4 = 0x06.3
#define servo5 = 0x05.2

#define T15 61989
#define T05 64286
#define T23 59786
#define T20 60536
#define T21 60286 // Time for control servo : 0.5 - 2.3 ms
#define TMR1H = 0x0F
#define TMR1L = 0x0E
unsigned long servo_val[5] = {T15+30,T15+20,T15-20,T15+5,T15};
```

```
// ser1 right back
// ser2 right font

int1 fan_flag = 0;
int speed_fan= 100;

int1 x = 0;
int1 mode = 0;

/*****
/
/* FUNCTION : main () */
/*****
/

void main (void)
{
    set_tris_a(0x00);
    set_tris_c(0x00);
    set_tris_b(0x00);

    setup_timer_1(T1_INTERNAL|T1_DIV_BY_2);
    set_timer1(0);

    setup_ccp1(CCP_PWM);
    setup_ccp2(CCP_PWM);
    setup_timer_2(T2_DIV_BY_16,127,1);
    set_timer2(0);

    enable_interrupts(INT_TIMER1);
```

```

enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);

set_pwm1_duty(0);
set_pwm2_duty(127);

while(true){

    // printf("\n%f %lu %lu %lu %lu ",servo_val[0],servo_val[1],servo_val[2],servo_val[3]);
    // delay_ms(100);
    if(fan_flag == 1){
        fan(speed_fan);
    }else
        fan(0);
}

}

/*****
/
/* FUNCTION: get_command() */
/*****
/
#INT_RDA
void get_command(void)
{
char cmd;
static int stat = 0;

```

```
cmd = getc();

/*
switch(cmd){
    case 'q': servo_val[0]++; break;
    case 'a': servo_val[0]--; break;

    case 'w': servo_val[1]++; break;
    case 's': servo_val[1]--; break;

    case 'e': servo_val[2]++; break;
    case 'd': servo_val[2]--; break;

    case 'r': servo_val[3]++; break;
    case 'f': servo_val[3]--; break;
}
*/

switch(stat){ // Select Control
case 0: if(cmd == '$'){
    stat = 1;
    }
    break;

case 1: switch(cmd){
    case CONTROL_DIRECTON : stat = 2; break;
    case CONTROL_FAN      : stat = 3; break;
    case CONTROL_ARM      : stat = 4; break;
    case SET_SPEED_FAN    : stat = 5; break;
    default                : stat = 0; break;
}
```

```
    }
```

```
    break;
```

```
case 2: switch(cmd){
```

```
    case CMD_FORWARD : servo_val[0] = T05; servo_val[1] = T05; servo_val[2] = T23;
servo_val[3] = T23;      x = 1; break;
```

```
    case CMD_BACKWARD : servo_val[0] = T23; servo_val[1] = T23; servo_val[2] = T05;
servo_val[3] = T05;      x = 1; break;
```

```
    case CMD_TURNLEFT : servo_val[0] = T05; servo_val[1] = T05; servo_val[2] = T05;
servo_val[3] = T05;      x = 1; break;
```

```
    case CMD_TURNRIGHT: servo_val[0] = T23; servo_val[1] = T23; servo_val[2] = T23;
servo_val[3] = T23;      x = 1; break;
```

```
    case CMD_STOP : servo_val[0] = T15+30; servo_val[1] = T15+20; servo_val[2] =
T15-20; servo_val[3] = T15+5; x = 0; break;
```

```
    case CMD_EN_ZR_GRAVITY : mode = 1; break;
```

```
    case CMD_DS_ZR_GRAVITY : mode = 0; break;
```

```
    }
```

```
    stat = 0;
```

```
    break;
```

```
case 3: switch(cmd){
```

```
    case CMD_FAN_OPEN : fan_flag = 1; LED_FAN(1); break;
```

```
    case CMD_FAN_CLOSE : fan_flag = 0; LED_FAN(0); break;
```

```
    }
```

```
    stat = 0;
```

```
    break;
```

```
case 4: switch(cmd){
```

```

        case CMD_UP_ARM: servo_val[4] = T15; LED_ARM(0); break;
        case CMD_DW_ARM: servo_val[4] = T23; LED_ARM(1); break;
        case CMD_CN_ARM: servo_val[4] = T20; LED_ARM(0); break;
        case CMD_CN1_ARM: servo_val[4] = T21; LED_ARM(0); break;
    }
    stat = 0;
    break;

case 5: speed_fan = cmd;
    stat = 0;
    break;
}

}

/*****
/
/* FUNCTION: control_servo() */
/*****
/
#INT_TIMER1
void control_servo(void)
{
    static int stat = 0;
    unsigned long Time;

    switch(stat){

        case 0: TMR1H = servo_val[0]>>8;

```

```
TMR1L = servo_val[0];  
servo1 = 1;  
stat++;  
break;
```

```
case 1: TMR1H = servo_val[1]>>8;  
TMR1L = servo_val[1];  
servo2 = 1;  
if((mode==1)||(x==1))servo1 = 0;  
stat++;  
break;
```

```
case 2: TMR1H = servo_val[2]>>8;  
TMR1L = servo_val[2];  
servo3 = 1;  
if((mode==1)||(x==1))servo2 = 0;  
stat++;  
break;
```

```
case 3: TMR1H = servo_val[3]>>8;  
TMR1L = servo_val[3];  
servo4 = 1;  
if((mode==1)||(x==1))servo3 = 0;  
stat++;  
break;
```

```
case 4: TMR1H = servo_val[4]>>8;  
TMR1L = servo_val[4];  
servo5 = 1;  
if((mode==1)||(x==1))servo4 = 0;
```

```

stat++;
break;

case 5: Time = 15536 - servo_val[0] - servo_val[1] - servo_val[2] - servo_val[3] -
servo_val[4];
TMR1H = Time>>8;
TMR1L = Time;
servo5 = 0;
stat = 0;
break;
}
}

```

## 2. Source Code ส่วน User Interface บน PC พัฒนาบน Visual Basic 2006

```

Private Sub Command1_Click()
    MSComm1.Output = "$21"
    Shape1.BackColor = &HFF00&
End Sub

Private Sub Command10_Click()
    If MSComm1.PortOpen = True Then
        MSComm1.PortOpen = False
        Shape3.BackColor = &HFF&
        Command1.Enabled = False
        Command2.Enabled = False
        Command3.Enabled = False
        Command4.Enabled = False
        Command5.Enabled = False
    End If
End Sub

```

```
Command6.Enabled = False
Command7.Enabled = False
Command8.Enabled = False
Command9.Enabled = True
Command10.Enabled = False
Command12.Enabled = False
Command13.Enabled = False
Command14.Enabled = False
Command15.Enabled = False
Slider1.Enabled = False
Shape1.BackColor = &HFF&
Shape1.BackColor = &HFF&
Shape1.BackColor = &HFF&
End If
End Sub
```

```
Private Sub Command11_Click()
End
End Sub
```

```
Private Sub Command12_Click()
    MSComm1.Output = "$33"
End Sub
```

```
Private Sub Command13_Click()
    MSComm1.Output = "$34"
End Sub
```

```
Private Sub Command14_Click()
    MSComm1.Output = "$16"
```

```
End Sub
```

```
Private Sub Command15_Click()
```

```
    MSComm1.Output = "$17"
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    MSComm1.Output = "$22"
```

```
    Shape1.BackColor = &HFF&
```

```
End Sub
```

```
Private Sub Command3_MouseDown(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
    ' go
```

```
    MSComm1.Output = "$11"
```

```
End Sub
```

```
Private Sub Command3_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)
```

```
    ' ungo
```

```
    MSComm1.Output = "$15"
```

```
End Sub
```

```
Private Sub Command4_MouseDown(Button As Integer, Shift As Integer, X As Single, Y  
As Single)
```

```
    ' back
```

```
    MSComm1.Output = "$12"
```

```
End Sub
```

```
Private Sub Command4_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
    ' unback  
    MSComm1.Output = "$15"  
End Sub
```

```
Private Sub Command5_MouseDown(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
    ' left  
    MSComm1.Output = "$13"  
End Sub
```

```
Private Sub Command5_MouseUp(Button As Integer, Shift As Integer, X As Single, Y  
As Single)  
    ' unleft  
    MSComm1.Output = "$15"  
End Sub
```

```
Private Sub Command6_MouseDown(Button As Integer, Shift As Integer, X As Single,  
Y As Single)  
    ' right  
    MSComm1.Output = "$14"  
End Sub
```

```
Private Sub Command6_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
    ' unlight  
    MSComm1.Output = "$15"
```

End Sub

```
Private Sub Command7_Click()  
    Shape2.BackColor = &HFF00&  
    MSComm1.Output = "$32"  
End Sub
```

```
Private Sub Command8_Click()  
    Shape2.BackColor = &HFF&  
    MSComm1.Output = "$31"  
End Sub
```

```
Private Sub Command9_Click()  
    On Error GoTo err  
    MSComm1.CommPort = Val(Text1.Text)  
    MSComm1.Settings = "9600,n,8,1"  
    MSComm1.PortOpen = True  
    Shape3.BackColor = &HFF00&  
    Command9.Enabled = False  
    Command10.Enabled = True  
    Slider1.Enabled = True  
    Command1.Enabled = True  
    Command2.Enabled = True  
    Command3.Enabled = True  
    Command4.Enabled = True  
    Command5.Enabled = True  
    Command6.Enabled = True  
    Command7.Enabled = True  
    Command8.Enabled = True  
    Command12.Enabled = True
```

```
Command13.Enabled = True
```

```
Command14.Enabled = True
```

```
Command15.Enabled = True
```

```
err:
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Slider1.Min = 0
```

```
Slider1.Max = 127
```

```
Command1.Enabled = False
```

```
Command2.Enabled = False
```

```
Command3.Enabled = False
```

```
Command4.Enabled = False
```

```
Command5.Enabled = False
```

```
Command6.Enabled = False
```

```
Command7.Enabled = False
```

```
Command8.Enabled = False
```

```
Command9.Enabled = True
```

```
Command10.Enabled = False
```

```
Command12.Enabled = False
```

```
Command13.Enabled = False
```

```
Command14.Enabled = False
```

```
Command15.Enabled = False
```

```
Slider1.Enabled = False
```

```
End Sub
```

```
Private Sub Slider1_Click()
```

```
MSComm1.Output = "$4" & Chr(Slider1.Value)
```

```
End Sub
```

**ภาคผนวก ข**

**Datasheet**



**L293D**  
**L293DD**

## PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

### DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

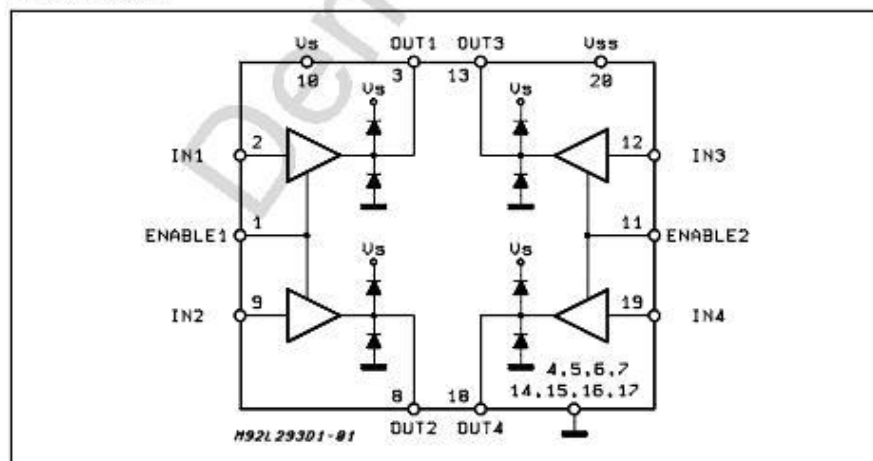
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

### BLOCK DIAGRAM

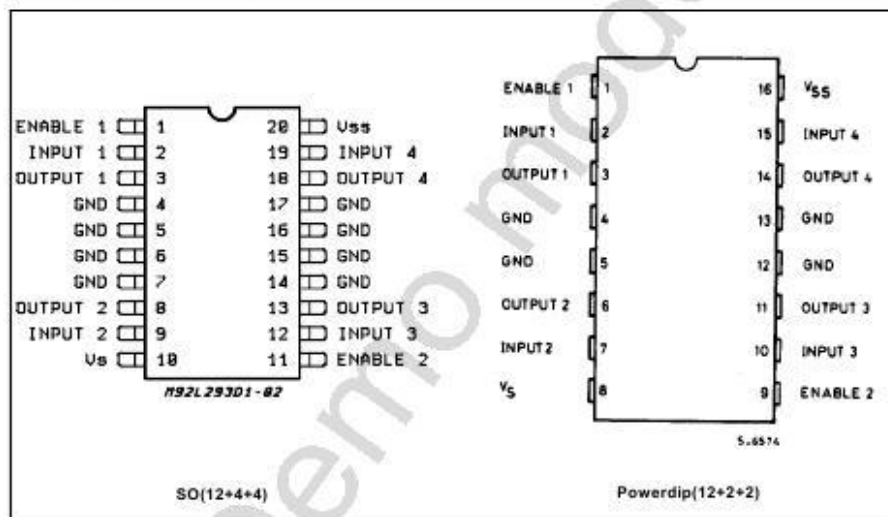


## L293D - L293DD

## ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_S$	Supply Voltage	36	V
$V_{SS}$	Logic Supply Voltage	36	V
$V_i$	Input Voltage	7	V
$V_{en}$	Enable Voltage	7	V
$I_o$	Peak Output Current (100 $\mu$ s non repetitive)	1.2	A
$P_{tot}$	Total Power Dissipation at $T_{pms} = 90$ °C	4	W
$T_{stg}, T_j$	Storage and Junction Temperature	- 40 to 150	°C

## PIN CONNECTIONS (Top view)



## THERMAL DATA

Symbol	Description	DIP	SO	Unit
$R_{th(j-pins)}$	Thermal Resistance Junction-pins	max. -	14	°C/W
$R_{th(j-amb)}$	Thermal Resistance junction-ambient	max. 80	50 (*)	°C/W
$R_{th(j-case)}$	Thermal Resistance Junction-case	max. 14	-	

(\*) With 6sq. cm on board heatsink.

## L293D - L293DD

**ELECTRICAL CHARACTERISTICS** (for each channel,  $V_S = 24\text{ V}$ ,  $V_{SS} = 5\text{ V}$ ,  $T_{\text{amb}} = 25\text{ }^\circ\text{C}$ , unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_S$	Supply Voltage (pin 10)		$V_{SS}$		36	V
$V_{SS}$	Logic Supply Voltage (pin 20)		4.5		36	V
$I_S$	Total Quiescent Supply Current (pin 10)	$V_i = L$ ; $I_O = 0$ ; $V_{en} = H$		2	6	mA
		$V_i = H$ ; $I_O = 0$ ; $V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
$I_{SS}$	Total Quiescent Logic Supply Current (pin 20)	$V_i = L$ ; $I_O = 0$ ; $V_{en} = H$		44	60	mA
		$V_i = H$ ; $I_O = 0$ ; $V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
$V_L$	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
$V_{IH}$	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		$V_{SS}$	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
$I_{L}$	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_L = 1.5\text{ V}$			-10	$\mu\text{A}$
$I_{IH}$	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{IH} \leq V_{SS} - 0.6\text{ V}$		30	100	$\mu\text{A}$
$V_{enL}$	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
$V_{enH}$	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		$V_{SS}$	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
$I_{enL}$	Low Voltage Enable Current (pin 1, 11)	$V_{enL} = 1.5\text{ V}$		-30	-100	$\mu\text{A}$
$I_{enH}$	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			$\pm 10$	$\mu\text{A}$
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = +0.6\text{ A}$		1.2	1.8	V
$V_F$	Clamp Diode Forward Voltage	$I_O = 600\text{ nA}$		1.3		V
$t_r$	Rise Time (*)	0.1 to 0.9 $V_O$		250		ns
$t_f$	Fall Time (*)	0.9 to 0.1 $V_O$		250		ns
$t_{on}$	Turn-on Delay (*)	0.5 $V_i$ to 0.5 $V_O$		750		ns
$t_{off}$	Turn-off Delay (*)	0.5 $V_i$ to 0.5 $V_O$		200		ns

(\*) See fig. 1.

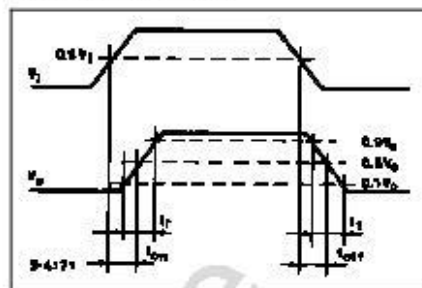
**L293D - L293DD**

**TRUTH TABLE (one channel)**

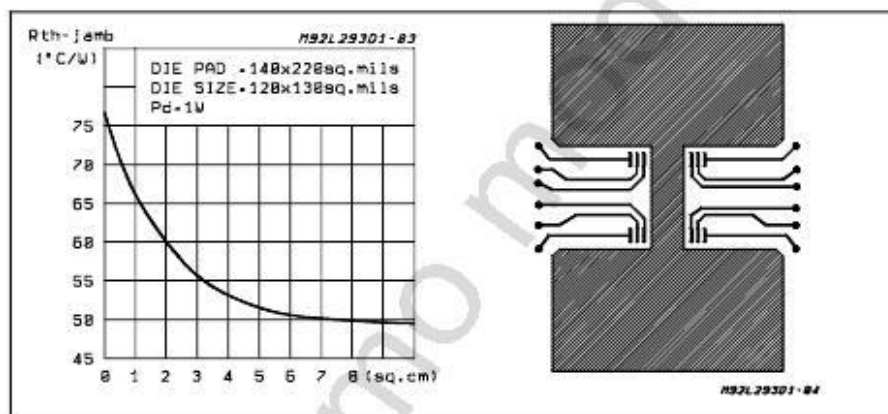
Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

Z = High output impedance  
 (\*) Relative to the considered channel

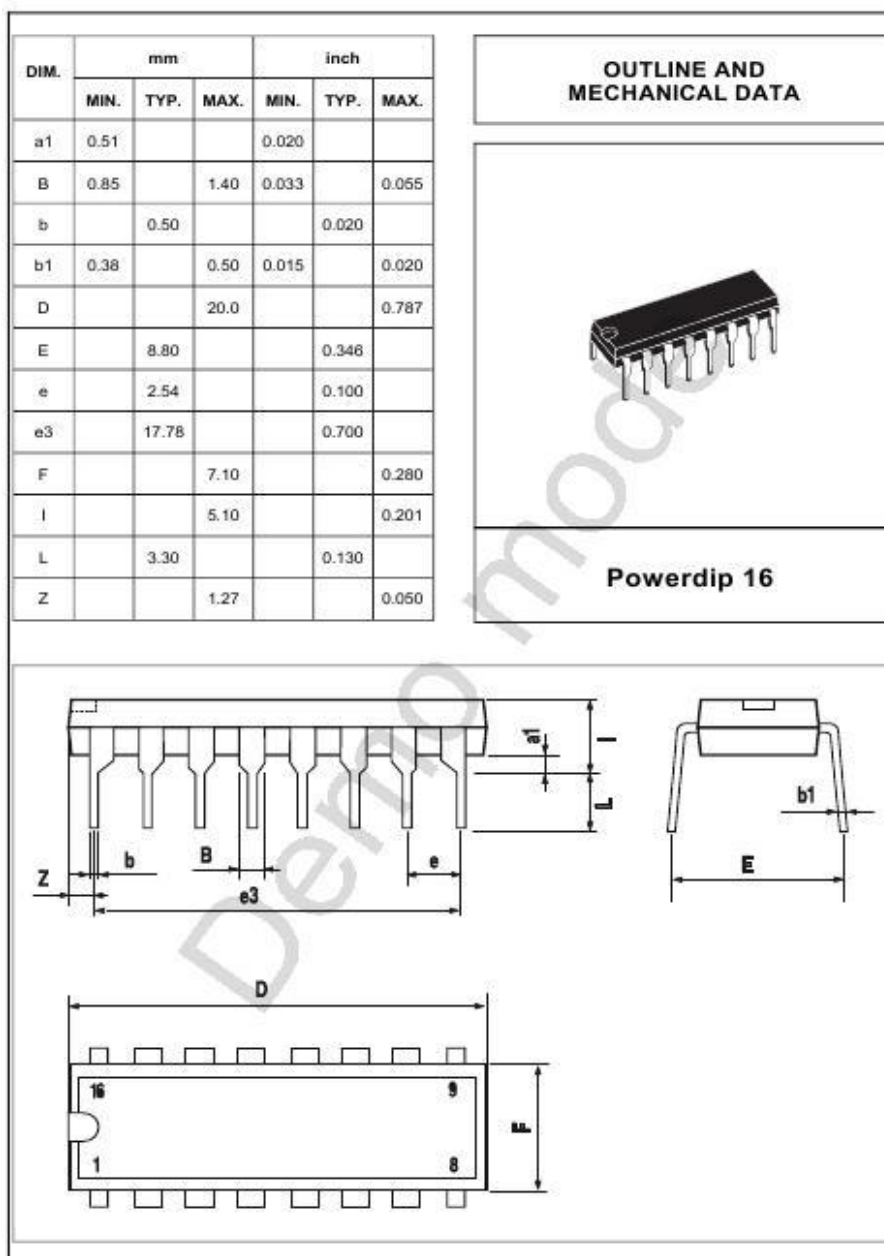
**Figure 1: Switching Times**



**Figure 2: Junction to ambient thermal resistance vs. area on board heatsink (SO12+4+4 package)**



## L293D - L293DD



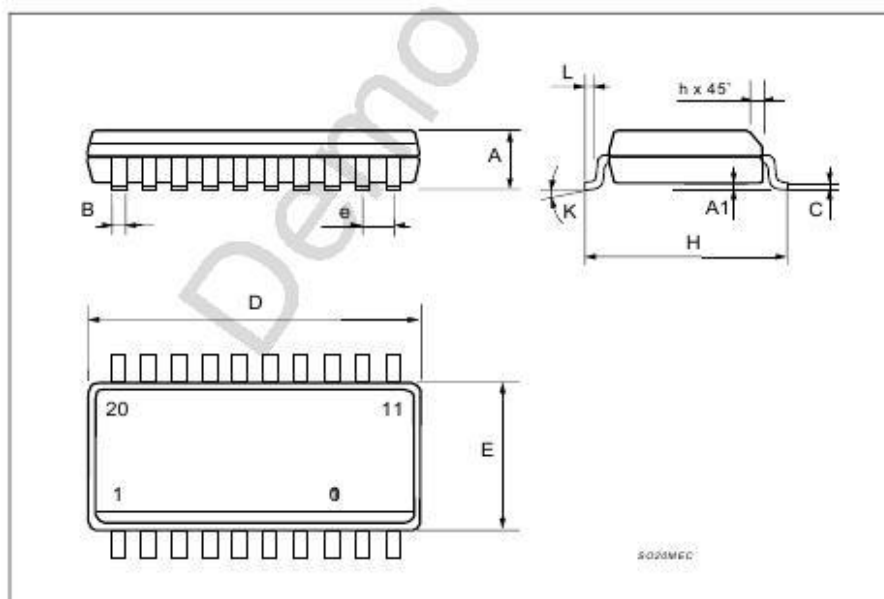
## L293D - L293DD

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	2.35		2.65	0.093		0.104
A1	0.1		0.3	0.004		0.012
B	0.33		0.51	0.013		0.020
C	0.23		0.32	0.009		0.013
D	12.6		13	0.496		0.512
E	7.4		7.6	0.291		0.299
e		1.27			0.050	
H	10		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
L	0.4		1.27	0.016		0.050
K	0° (min.) 8° (max.)					

## OUTLINE AND MECHANICAL DATA



SO20



Demo mode

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

© 2003 STMicroelectronics - Printed in Italy - All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

<http://www.st.com>





February 2005

## IRF840B/IRFS840B

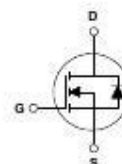
### 500V N-Channel MOSFET

#### General Description

These N-Channel enhancement mode power field effect transistors are produced using Fairchild's proprietary, planar, DMOS technology. This advanced technology has been especially tailored to minimize on-state resistance, provide superior switching performance, and withstand high energy pulse in the avalanche and commutation mode. These devices are well suited for high efficiency switch mode power supplies, power factor correction and electronic lamp ballasts based on half bridge.

#### Features

- 8.0A, 500V,  $R_{DS(on)} = 0.8\Omega$  @  $V_{GS} = 10V$
- Low gate charge (typical 41 nC)
- Low  $C_{rss}$  (typical 35 pF)
- Fast switching
- 100% avalanche tested
- Improved dv/dt capability



#### Absolute Maximum Ratings

$T_C = 25^\circ\text{C}$  (unless otherwise noted)

Symbol	Parameter	IRF840B	IRFS840B	Units
$V_{DS}$	Drain-Source Voltage	500		V
$I_D$	Drain Current - Continuous ( $T_C = 25^\circ\text{C}$ )	8.0	8.0	A
	- Continuous ( $T_C = 100^\circ\text{C}$ )	5.1	5.1	A
$I_{DM}$	Drain Current - Pulsed (Note 1)	32	32	A
$V_{GS}$	Gate-Source Voltage	$\pm 30$		V
$E_{AS}$	Single Pulsed Avalanche Energy (Note 2)	320		mJ
$I_{AR}$	Avalanche Current (Note 1)	8.0		A
$E_{AR}$	Repetitive Avalanche Energy (Note 1)	13.4		mJ
dv/dt	Peak Diode Recovery dv/dt (Note 3)	5.5		V/ns
$P_D$	Power Dissipation ( $T_C = 25^\circ\text{C}$ )	134	44	W
	- Derate above $25^\circ\text{C}$	1.08	0.35	W/°C
$T_J, T_{STG}$	Operating and Storage Temperature Range	-55 to +150		°C
$T_L$	Maximum lead temperature for soldering purposes, 1/8" from case for 5 seconds	300		°C

\* Drain current limited by maximum junction temperature.

#### Thermal Characteristics

Symbol	Parameter	IRF840B	IRFS840B	Units
$R_{\theta JC}$	Thermal Resistance, Junction-to-Case Max.	0.93	2.86	°C/W
$R_{\theta CS}$	Thermal Resistance, Case-to-Sink Typ.	0.5	--	°C/W
$R_{\theta JA}$	Thermal Resistance, Junction-to-Ambient Max.	62.5	62.5	°C/W

Electrical Characteristics		T <sub>c</sub> = 25°C unless otherwise noted				
Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
<b>Off Characteristics</b>						
BV <sub>DSS</sub>	Drain-Source Breakdown Voltage	V <sub>GS</sub> = 0 V, I <sub>D</sub> = 250 μA	500	--	--	V
ΔBV <sub>DSS</sub> / ΔT <sub>J</sub>	Breakdown Voltage Temperature Coefficient	I <sub>D</sub> = 250 μA, Referenced to 25°C	--	0.55	--	V/°C
I <sub>DSS</sub>	Zero Gate Voltage Drain Current	V <sub>DS</sub> = 500 V, V <sub>GS</sub> = 0 V V <sub>DS</sub> = 400 V, T <sub>C</sub> = 125°C	--	--	10	μA
I <sub>DSSF</sub>	Gate-Body Leakage Current, Forward	V <sub>GS</sub> = 30 V, V <sub>DS</sub> = 0 V	--	--	100	nA
I <sub>DSSR</sub>	Gate-Body Leakage Current, Reverse	V <sub>GS</sub> = -30 V, V <sub>DS</sub> = 0 V	--	--	-100	nA
<b>On Characteristics</b>						
V <sub>GS(th)</sub>	Gate Threshold Voltage	V <sub>DS</sub> = V <sub>GS</sub> , I <sub>D</sub> = 250 μA	2.0	--	4.0	V
R <sub>DS(on)</sub>	Static Drain-Source On-Resistance	V <sub>GS</sub> = 10 V, I <sub>D</sub> = 4.0 A	--	0.65	0.8	Ω
g <sub>FS</sub>	Forward Transconductance	V <sub>DS</sub> = 40 V, I <sub>D</sub> = 4.0 A (Note 4)	--	7.3	--	S
<b>Dynamic Characteristics</b>						
C <sub>iss</sub>	Input Capacitance	V <sub>DS</sub> = 25 V, V <sub>GS</sub> = 0 V	--	1400	1800	pF
C <sub>oss</sub>	Output Capacitance	f = 1.0 MHz	--	145	190	pF
C <sub>rss</sub>	Reverse Transfer Capacitance		--	35	45	pF
<b>Switching Characteristics</b>						
t <sub>d(on)</sub>	Turn-On Delay Time	V <sub>DD</sub> = 250 V, I <sub>D</sub> = 8.0 A, R <sub>G</sub> = 25 Ω	--	22	55	ns
t <sub>r</sub>	Turn-On Rise Time		--	65	140	ns
t <sub>d(off)</sub>	Turn-Off Delay Time		--	125	260	ns
t <sub>f</sub>	Turn-Off Fall Time	(Note 4, 5)	--	75	160	ns
Q <sub>g</sub>	Total Gate Charge	V <sub>DS</sub> = 400 V, I <sub>D</sub> = 8.0 A, V <sub>GS</sub> = 10 V	--	41	53	nC
Q <sub>gs</sub>	Gate-Source Charge		--	6.5	--	nC
Q <sub>gd</sub>	Gate-Drain Charge	(Note 4, 5)	--	17	--	nC
<b>Drain-Source Diode Characteristics and Maximum Ratings</b>						
I <sub>S</sub>	Maximum Continuous Drain-Source Diode Forward Current		--	--	8.0	A
I <sub>SM</sub>	Maximum Pulsed Drain-Source Diode Forward Current		--	--	32	A
V <sub>SD</sub>	Drain-Source Diode Forward Voltage	V <sub>GS</sub> = 0 V, I <sub>S</sub> = 8.0 A	--	--	1.4	V
t <sub>rr</sub>	Reverse Recovery Time	V <sub>GS</sub> = 0 V, I <sub>S</sub> = 8.0 A,	--	390	--	ns
Q <sub>rr</sub>	Reverse Recovery Charge	dI <sub>F</sub> / dt = 100 A/μs (Note 4)	--	4.2	--	μC
<b>Notes:</b>						
1. Repetitive Rating: Pulse width limited by maximum junction temperature						
2. L = 0.0nH, I <sub>GS</sub> = 8.0A, V <sub>DS</sub> = 50V, R <sub>G</sub> = 25 Ω, Starting T <sub>J</sub> = 25°C						
3. I <sub>D</sub> = 8.0A, dI <sub>D</sub> / dt ≤ 200A/μs, V <sub>DS</sub> = BV <sub>DSS</sub> , Starting T <sub>J</sub> = 25°C						
4. Pulse Test: Pulse width ≤ 300μs, Duty cycle ≤ 2%						
5. Essentially independent of operating temperature						

Typical Characteristics

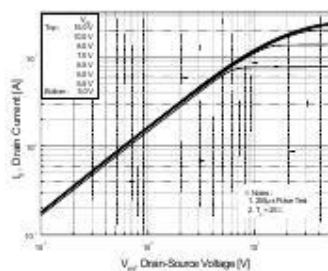


Figure 1. On-Region Characteristics

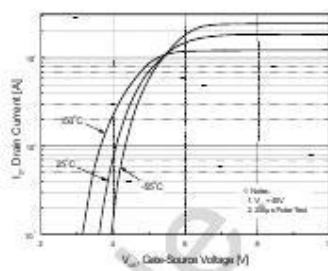


Figure 2. Transfer Characteristics

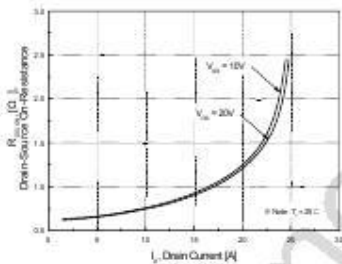


Figure 3. On-Resistance Variation vs Drain Current and Gate Voltage

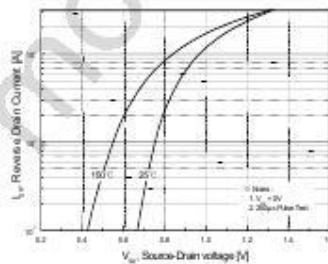


Figure 4. Body Diode Forward Voltage Variation with Source Current and Temperature

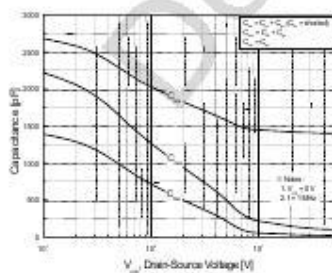


Figure 5. Capacitance Characteristics

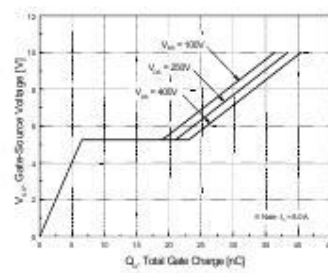


Figure 6. Gate Charge Characteristics

Typical Characteristics (Continued)

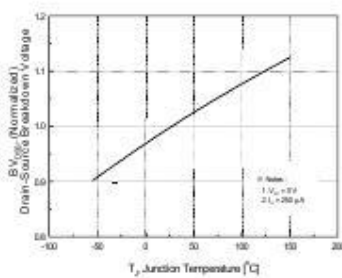


Figure 7. Breakdown Voltage Variation vs Temperature

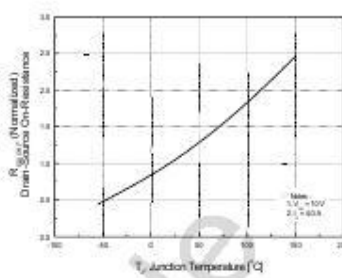


Figure 8. On-Resistance Variation vs Temperature

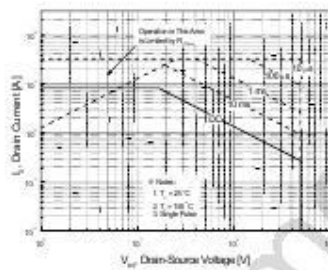


Figure 9-1. Maximum Safe Operating Area for IRF840B

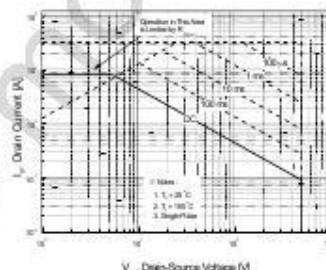


Figure 9-2. Maximum Safe Operating Area for IRFS840B

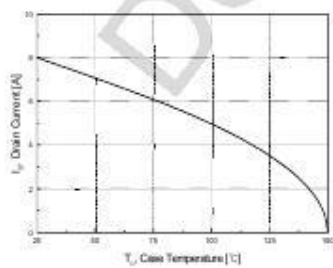


Figure 10. Maximum Drain Current vs Case Temperature

Typical Characteristics (Continued)

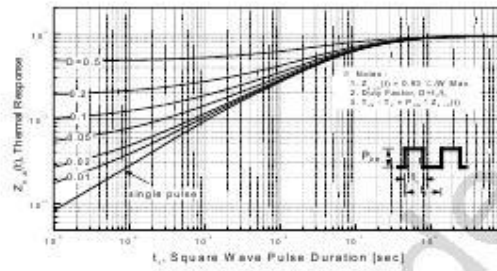


Figure 11-1. Transient Thermal Response Curve for IRF840B

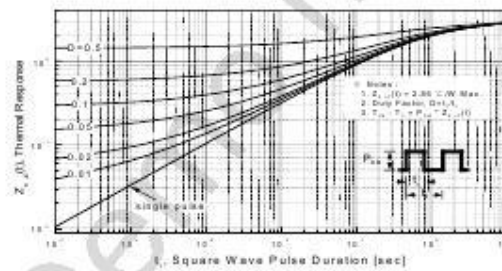
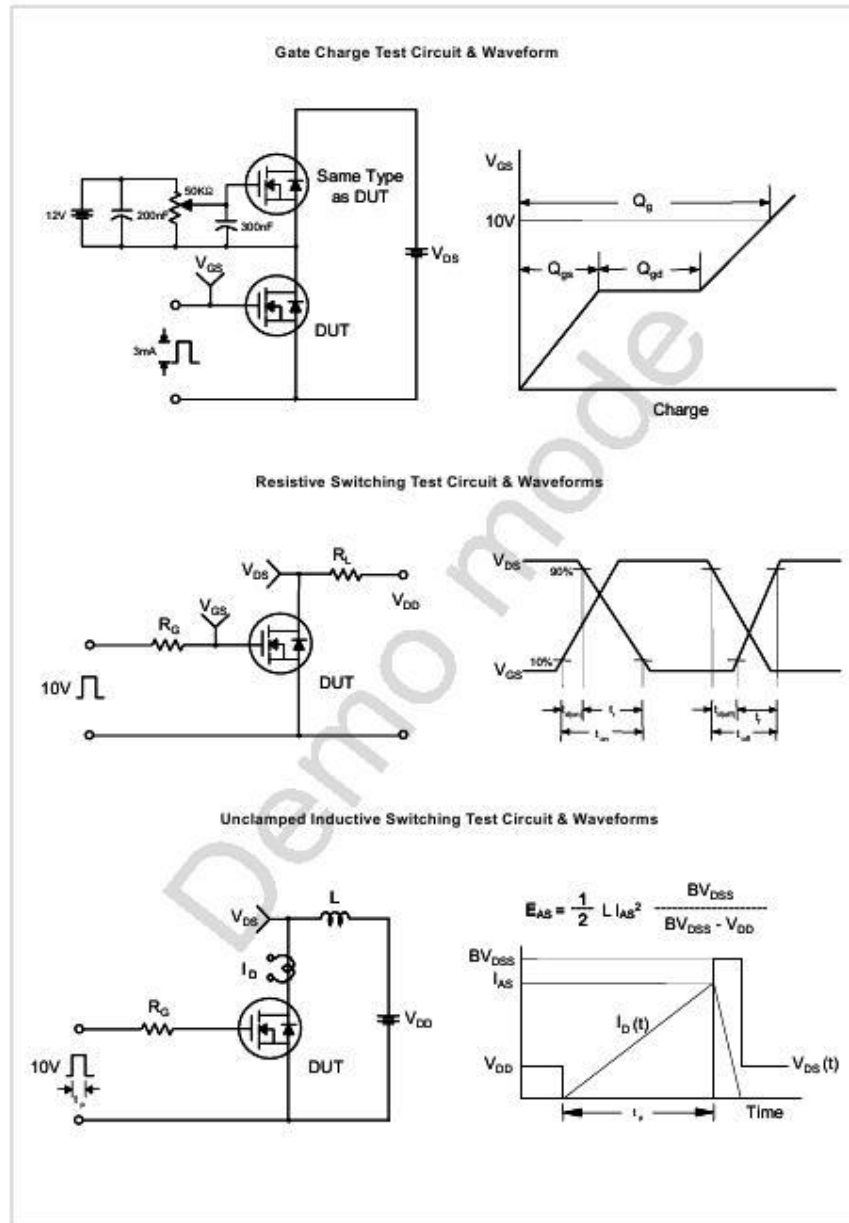
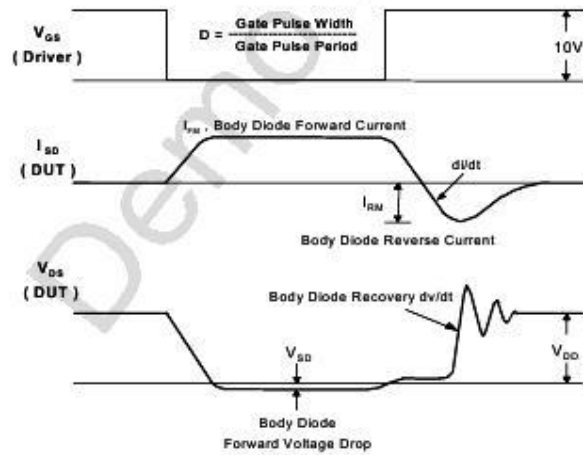
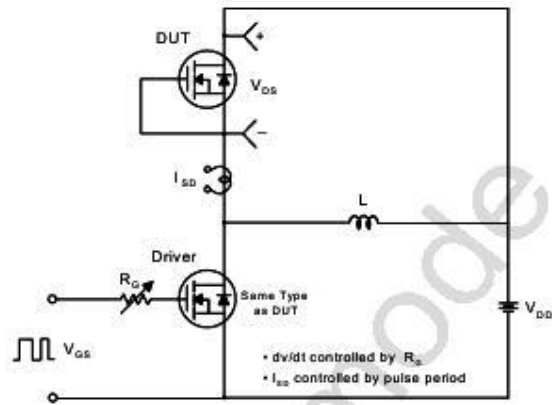
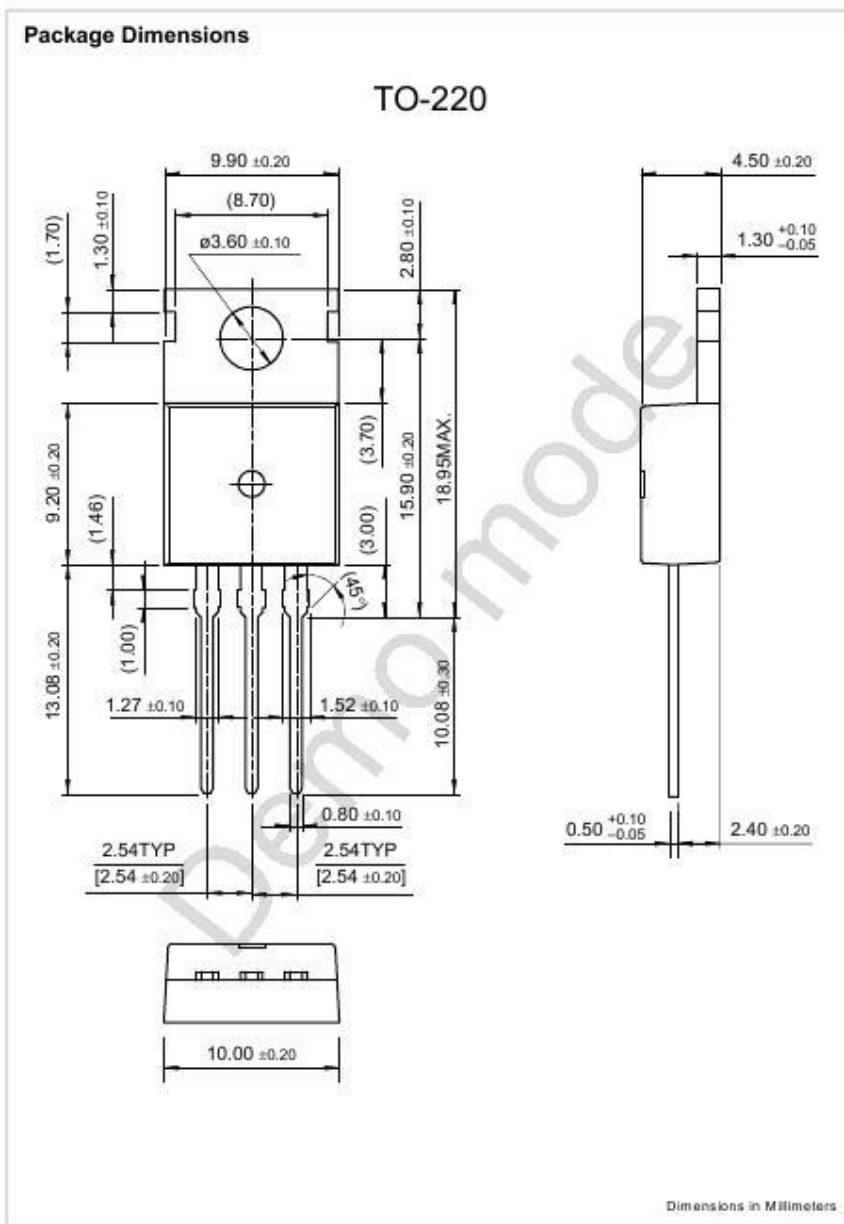


Figure 11-2. Transient Thermal Response Curve for IRFS840B

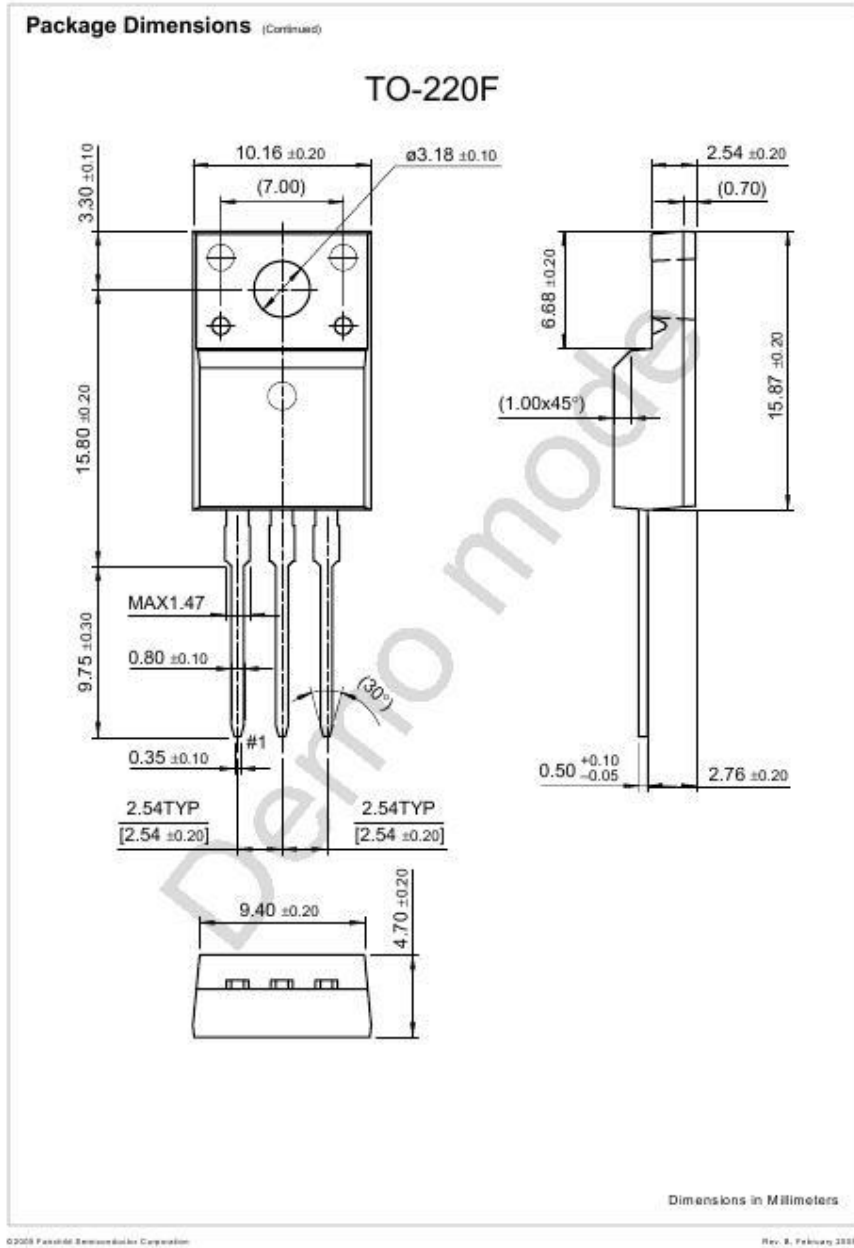


Peak Diode Recovery dv/dt Test Circuit & Waveforms





IRF 840B/IRFS 840B



IRF 840B/IRFS 840B

## TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FAST®	IntelliMAX™	POP™	SPM™
ActiveArray™	FASTr™	ISOPLANAR™	Power247™	Stealth™
Bottomless™	FPS™	LittleFET™	PowerEdge™	SuperFET™
CoolFET™	FRFET™	MICROCOUPLER™	PowerSaver™	SuperSOT™-3
CROSSVOLT™	GlobalOptoisolator™	MicroFET™	PowerTrench®	SuperSOT™-6
DOME™	GTO™	MicroPak™	QFET®	SuperSOT™-8
EcoSPARK™	HiSeC™	MICROWIRE™	QS™	SyncFET™
E <sup>2</sup> C MOS™	IC™	MSX™	QT Optoelectronics™	TinyLogic®
EnSigna™	i-Lo™	MSXPro™	Quiet Series™	TINYOPTO™
FACT™	ImpliedDisconnect™	OCX™	RapidConfigure™	TruTranslation™
FACT Quiet Series™		OCXPro™	RapidConnect™	UHC™
Across the board. Around the world.™		OPTOLOGIC®	µSerDes™	UltraFET®
The Power Franchise®		OPTOPLANAR™	SILENT SWITCHER®	UniFET™
Programmable Active Droop™		PACMAN™	SMART START™	VCX™

## DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

## LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

## PRODUCT STATUS DEFINITIONS

### Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not in Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

ประวัติผู้จัดทำปริยญาณิพนธ์

## ประวัติผู้จัดทำปฏิญานิพนธ์



ชื่อ	นายเหล็กกล้า ศรีพิจิตร รหัส 115110462043-0
สาขาวิชา/ภาควิชา	ภาควิชาวิศวกรรมคอมพิวเตอร์
วัน-เดือน-ปี เกิด	วันที่ 22 มีนาคม 2533
สถานที่เกิด	จังหวัดชุมพร
ที่อยู่	27/11-12 ถ.ปรมิทรมรรคา ต.ท่าตะเภา อ.เมือง จ.ชุมพร 86000
ประวัติการศึกษา	ประกาศนียบัตรวิชาชีพ (ปวช.) วิทยาลัยเทคนิคชุมพร ปีการศึกษา 2550

## ประวัติผู้จัดทำปริญญาานิพนธ์



ชื่อ	นายวิสรุช แสงมณี รหัส 115110462016-6
สาขาวิชา/ภาควิชา	ภาควิชา วิศวกรรมคอมพิวเตอร์
วัน-เดือน-ปี เกิด	วันที่ 23 พฤศจิกายน 2532
สถานที่เกิด	จังหวัดกรุงเทพมหานคร
ที่อยู่	49/23 หมู่ 9 ต.ลำผักชี อ.หนองจอก จ.กรุงเทพมหานคร 10530
ประวัติการศึกษา	มัธยมศึกษาตอนปลาย (ม.6) ร.ร.เตรียมอุดมศึกษาพัฒนาการ สุวินทวงศ์ ปีการศึกษา 2550