สำนักหอสมุดกลาง พระจอมเกล้าฉลาดกระบัง

# หุ่นยนต์กู้ภัย

# Rescue Robot

จัดทำโดย

นายจักรพงศ์ วันทา

นายไตรภพ นุชน้อย

นายวินัย ปิงพิพัฒน์ตระกูล

ปพ.
TJ222ห
2550

เลขหมู่.....
เลขทะเบียน 82429
วัน,เดือน,ปี 11 ก.ค. 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

b.11944933
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หุ่นยนต์กู้ภัย

# Rescue Robot


จัดทำโดย

นายจักรพงศ์ วันทา

นายไตรภพ นุชน้อย

นายวินัย ปึงพิพัฒน์ตระกูล

อาจารย์ที่ปรึกษา

ผศ.ดร.ยุทธนา คิดใจเดียว



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ปริญญานิพนธ์** **ปีการศึกษา 2550**

**ภาควิชา** **อิเล็กทรอนิกส์**

**คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**เรื่อง** **หุ่นยนต์กู้ภัย Rescue Robot**


**ผู้จัดทำ**

1. **นายจักรพงศ์ วันทา**
2. **นายไตรภพ นุชน้อย**
3. **นายวินัย ปิงพิพัฒน์ตระกูล**


**.....**

**(ผศ.ดร.ยุทธนา คิดใจเดียว)**

**อาจารย์ที่ปรึกษา**


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หุ่นยนต์กู้ภัย ( Rescue Robot )

นายจักรพงศ์ วันทา รหัสนักศึกษา 48015155
นายไตรภพ นุชน้อย รหัสนักศึกษา 48015205
นายวินัย ปึงพิพัฒน์ตระกูล รหัสนักศึกษา 48015226
ผศ.ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา
ประจำปีการศึกษาที่ 2550

**บทคัดย่อ**

ในการทำหุ่นยนต์กู้ภัย (Rescue Robot) ได้มีแนวคิดมาจากการช่วยชีวิตผู้ประสบภัยที่เกิดจากธรรมชาติหรืออุบัติเหตุที่คนไม่สามารถจะเข้าไปถึงผู้ประสบภัยได้ จึงได้คิดค้นและสร้างหุ่นยนต์ที่จะสามารถเข้าไปสำรวจค้นหาในพื้นที่ที่คนไม่สามารถเข้าไปถึงได้และลดความเสี่ยงที่จะเกิดกับผู้ช่วยเหลือ การทำงานของตัวหุ่นจะแบบเป็นสองส่วนคือ ส่วนที่หนึ่งการทำงานของตัวหุ่นที่รับคำสั่งโดยผ่านสัญญาณ Wireless ที่สั่งจากภาคพื้นดิน ส่วนที่สองการControlจากภาคพื้นดิน

ความสามารถของตัวหุ่นยนต์ที่มี คือ ต้องสามารถบอกตำแหน่งที่แน่นอนของผู้บาดเจ็บและส่งภาพมายังภาคพื้นดินโดยมีกล้องเป็นตัวสำรวจ และใช้เข็มทิศอิเล็กทรอนิกส์เป็นตัวบอกตำแหน่งและทิศทาง หุ่นยนต์กู้ภัยตัวนี้จะสามารถเดินไปในพื้นที่ต่างๆ โดยมีสิ่งกรีดขวางสามารถปีนป่ายในพื้นที่สูงชันหรือบันไดได้ และยังสามารถบอกอุณหภูมิของผู้ได้รับบาดเจ็บได้ว่ายังมีชีวิตอยู่หรือไม่ โดยจะแสดงผลผ่านกล้องส่งผ่านมายังภาคพื้นดิน เมื่อตรวจสอบแล้วว่าผู้ประสบภัยยังมีชีวิตอยู่ การช่วยเหลือผู้ประสบภัยก็จะมีความระมัดระวังมากขึ้นเพื่อไม่ให้ผู้ได้รับบาดเจ็บมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Rescue Robot

**Mr. Chakkapong Wanta ID. 48015155**

**Mr. Tripop Nuchnoi ID. 48015205**

**Mr. Vinai Puengpattrakul ID. 48015226**

**Asst.Prof.Dr.Yuttana Kidjaidure Advisor**

**Educational Year 2007**

## Abstract

A Rescue Robot has a concept of helping the victim suffered from the disaster or accident places that human can not go into. So , the aim is to build the robot that can observe the land and reduce the risk that will happen. The robot consists of 2 parts the robot's command from the wireless and the control from the joystick.

The robot has to search the position of the victim and send the image to the controller. The electronic compass is used to tell the direction. This robot can walk into the place that has an obstacle and can climb to the steep place or stair. Furthermore , it can tell the temperature of the victim whether the victims are still alive. If they are still alive , the victims will be rescue with care.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ขอขอบพระคุณ ผศ.ดร. ยุทธนา คิดใจเดียว เป็นอย่างสูง ที่ได้ให้คำปรึกษาในการทำโครงงานและพี่ ๆ ที่คอยดูแลให้ความช่วยเหลือแทนอาจารย์ด้วยที่ได้คอยสอนและให้คำปรึกษา รวมถึงการเอื้อเฟื้อสถานที่ในการทำโครงงานหุ่นยนต์กู้ภัยนี้ เนื่องจากจำเป็นที่ต้องใช้ห้องเชื่อมเป็นระยะเวลานานในการทำหุ่น จึงต้องขอขอบคุณพี่ ๆ ด้วยเป็นอย่างสูง ที่ได้คอยให้ความช่วยเหลือตลอดระยะเวลาในการทำโครงงานจนกระทั่งสำเร็จลงด้วยดี

ผู้จัดทำ

นาย จักรพงศ์ วันทา รหัส 48015155

นาย ไตรภพ นุชน้อย รหัส 48015205

นาย วินัย ปิงพิพัฒน์ตระกูล รหัส 48015226



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| เรื่อง | หน้า |
|---|---|
| บทคัดย่อ | I |
| Abstract | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV, VI |
| สารบัญรูปภาพ | VII,VIII,IX |
| สารบัญตาราง | X |
| **บทที่ 1 บทนำ** | |
| 1.1 ความเป็นมาของโครงงาน | 1 |
| 1.2 วัตถุประสงค์ | 2 |
| 1.3 ขอบเขตของโครงงาน | 2 |
| 1.4 ประโยชน์ที่จะได้รับจากโครงงาน | 2 |
| 1.5 รายละเอียดโดยย่อของโครงงาน | 3 |
| 1.6 รายละเอียดของปริญญานิพนธ์ | 4 |
| **บทที่ 2 ทฤษฎีและหลักการทำงาน** | |
| 2.1 บล็อกไดอะแกรมการทำงาน | 5 |
| 2.2 อุปกรณ์ตรวจวัด | |
| 2.2.1 เซนเซอร์ตรวจจับหอุณภูมิ | 6 |
| 2.2.2 เข็มทิศดิจิตอล | 9 |
| 2.3 CPU (ไมโครคอนโทรลเลอร์ PIC16F877) | |
| 2.3.1 PIC16F877 | 18 |
| 2.3.2 สัญญาณนาฬิกา | 19 |
| 2.3.3 การจัดขาไอซี | 20 |
| 2.3.4 คุณสมบัติของ 16F877 | 21 |
| 2.3.5 โครงสร้างภายในชิพ | 23 |
| 2.4 โปรแกรมที่ใช้ในการควบคุมหุ่นยนต์ | |
| 2.4.1 Visual Basic 6 | 26 |
| 2.4.2 แนวคิดของ Visual Basic 6 | 26 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เรื่อง | หน้า |
|---|---|
| 2.4.3 โปรแกรม Visual Basic 6 | 26 |
| 2.4.4 เข้าสู่โปรแกรม Visual Basic 6 | 27 |
| 2.4.5 ทูลบาร์ | 28 |
| 2.4.6 Tool boxs | 29 |
| 2.4.7 คอนโทรลภายใน | 30 |
| 2.4.8 คอนโทรล Active | 30 |
| 2.4.9 Form Designer | 30 |
| 2.4.10 Project Explorer | 31 |
| 2.4.11 ส่วนประกอบของโปรเจ็กต์ | 32 |
| 2.4.12 Properties Window | 32 |
| 2.4.13 หน้าต่าง Form Layout | 33 |
| 2.4.14 Immediate Window | 33 |
| 2.4.15 หน้าต่าง New Project | 34 |
| 2.4.16 หน้าต่าง Code Editor | 35 |
| 2.5 ภาครับส่งข้อมูล | |
| 2.5.1 เครื่องรับ – ส่ง ที่ใช้ในการควบคุมหุ่นยนต์ (ETT) | 35 |
| 2.5.2 แหล่งจ่ายไฟเลี้ยง | 37 |
| 2.5.3 คู่มือการใช้ RS 232 to RF-Wireless | 38 |
| 2.5.4 โหมดการทำงาน | 39 |
| 2.5.5 การใช้งานเครื่อง ET-RF24G V2.0 ใน Run Mode | 39 |
| 2.5.6 การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode | 43 |
| 2.5.7 คู่มือการใช้งาน RS 232 to RF-Wireless | 46 |
| 2.5.8 ข้อแนะนำการกำหนดค่า Configuration | 47 |
| 2.5.9 การเชื่อมต่อสัญญาณ RS232 | 49 |
| 2.5.10 เครื่องรับ – ส่ง กล้องวงจรปิด | 54 |
| 2.6 คอมพิวเตอร์ | 58 |
| 2.7 Driver Motor | |
| 2.7.1 ชุด Driver Motor | 58 |
| 2.8 มอเตอร์ | |
| 2.8.1 ประเภทของมอเตอร์ | 59 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เรื่อง | หน้า |
|---|---|
| 2.8.2 ส่วนประกอบที่สำคัญของมอเตอร์ | 59 |
| 2.8.3 ลักษณะของมอเตอร์ | 60 |
| 2.8.4 การให้กระแสเข้าขดลวด | 60 |
| 2.8.5 หลักการทำงานของดีซีมอเตอร์ | 61 |
| 2.9 กล้องวงจรปิด | |
| 2.9.1 กล้อง CCTV | 63 |
| 2.9.2 กล้องสีธรรมดา | 67 |
| **บทที่ 3 การออกแบบส่วนต่างๆ ของหุ่นยนต์** | |
| 3.1 โครงสร้างการทำงานของหุ่นยนต์ | |
| 3.1.1 ส่วนต่างๆ ของหุ่นยนต์ | 68 |
| 3.1.2 ส่วนของการสื่อสารแบบไร้สาย แบ่งเป็น 2 ส่วน | 71 |
| 3.1.3 ส่วนของคอมพิวเตอร์ | 72 |
| **บทที่ 4 ผลการทดลอง** | |
| 4.1 ผลการทดลองทางด้านโปรแกรมควบคุมหุ่นยนต์ | 73 |
| 4.2 การทดลองคุณสมบัติของอุปกรณ์เข็มทิศอิเล็กทรอนิกส์ | 78 |
| 4.3 คุณสมบัติโดยรวมของหุ่นยนต์กู้ภัย | 78 |
| 4.4 รูปผลการทดลองโปรแกรมวาดแผนที่ด้วย Visual Basic 6.0 | 80 |
| **บทที่ 5 สรุปและวิจารณ์** | 81 |
| **ภาคผนวก** | 82 |
| **บรรณานุกรม** | 115 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ


| รูป | หน้า |
|---|---|
| รูปที่ 2.1 บล็อกไดอะแกรมแสดงการทำงานในส่วนต่างๆ | 5 |
| รูปที่ 2.2 บล็อกไดอะแกรมการทำงานของเครื่องตรวจจับอุณหภูมิ | 6 |
| รูปที่ 2.3 แสดงการวัดอุณหภูมิ ระยะทางเทียบกับวัตถุ | 7 |
| รูปที่ 2.4 แสดงถึงความสามารถในการแผ่รังสีอินฟราเรด | 8 |
| รูปที่ 2.5 แสดงเครื่องตรวจจับอุณหภูมิที่นำมาใช้งาน | 8 |
| รูปที่ 2.6 แสดงรูปร่างและตำแหน่งขาสำหรับการต่อใช้งาน ของเข็มทิศดิจิตอล | 10 |
| รูปที่ 2.7 แสดงวงจรของบอร์ด ADX-CMPS03 และการเชื่อมต่อกับโมดูล CMPS03 | 11 |
| รูปที่ 2.8 แสดงไทมิ่งไดอะแกรมของการติดต่อสื่อสารกับโมดูล CMPS03 ผ่านระบบ บัส $I^2C$ | 14 |
| รูปที่ 2.9แสดงหน้าจอของ Debug terminal แสดงข้อมูลที่อ่านได้จากโมดูล CMPS03 | 15 |
| รูปที่ 2.10 บอร์ด ADX-CMPS03 | 17 |
| รูปที่ 2.11 สายเชื่อมต่อ PCB3A | 17 |
| รูปที่ 2.12 แสดงตำแหน่งขา IC PIC 16F877 | 18 |
| รูปที่ 2.13 แสดงสัญญาณนาฬิกาของ PIC | 20 |
| รูปที่ 2.14 แสดงโครงสร้างขาของ 16F877 | 21 |
| รูปที่ 2.15 แสดงโครงสร้างแผนผัง IC PIC 16F877 | 23 |
| รูปที่ 2.16 วงจรควบคุมโดยไมโครคอนโทรเลอร์ | 25 |
| รูปที่ 2.17 เมื่อเลือกชนิดของโปรแกรมประยุกต์เป็นแบบ Standard EXE จะเข้าสู่หน้าต่างของ | 27 |
| รูปที่ 2.18 แสดงส่วนของ Visual Basi | 28 |
| รูปที่ 2.19 แสดงทูลบาร์ (Toolbars) | 28 |
| รูปที่ 2.20 แสดง Toolbox | 29 |
| รูปที่ 2.21 แสดงคอนโทรล ActiveX (ActiveX controls) | 30 |
| รูปที่ 2.22 แสดง Form Designer | 31 |
| รูปที่ 2.23 Project Explorer แบบโปรเจ็กต์เดียว และ แบบหลายโปรเจ็กต์ | 31 |
| รูปที่ 2.24 แสดง Properties Window | 32 |
| รูปที่ 2.25 แสดงหน้าต่าง Form Layout | 33 |
| รูปที่ 2.26 แสดง Immediate Window | 34 |
| รูปที่ 2.27 แสดง New Project | 34 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป** **หน้า**


รูปที่ 2.28 แสดงหน้าต่าง Code Editor 35

รูปที่ 2.29 แสดงรูปร่าง ETT 37

รูปที่ 2.30 แสดงการต่อสายสัญญาณ RS232 เพื่อใช้แหล่งจ่ายไฟจากบอร์ดไมโครฯ ของ ETT 37

รูปที่ 2.31 แสดงการต่อแหล่งจ่ายไฟรุ่น "ACH-4E"จากภายนอกให้กับเครื่อง ET-RF24G V1.0 38

รูปที่ 2.32 แสดงสายสัญญาณ RS232 เพื่อใช้กับ ET-RF24G ในโหมด RF Receive Only และ RF Transmit Only 41

รูปที่ 2.33 แสดงโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V2.0 44

รูปที่ 2.34 แสดง แผนผังการต่อสาย RS232เพื่อใช้งานกับ ET-RF24G V2.0 ในโหมด Auto Direction 49

รูปที่ 2.35 ตัวอย่างที่ 1 การรับส่งข้อมูล 2 ทิศทาง(Half Duplex) แบบ จุดต่อจุด (Point-to-Point) 50

รูปที่ 2.36 ตัวอย่างที่ 2 การรับส่งข้อมูล 2 ทิศทาง(Half Duplex) แบบ หลายๆจุด (RF Network) 51

รูปที่ 2.37 แสดงตัวอย่างที่ 3 การรับส่งข้อมูลแบบFull Duplex ด้วยโปรแกรมสำเร็จรูป 53

รูปที่ 2.38 แสดงเครื่องรับ – ส่ง สัญญาณภาพ 2 วัตต์ 54

รูปที่ 2.39 ส่วนประกอบของตัวรับ-ส่งสัญญาณภาพและเสียง 56

รูปที่ 2.40 วงจรเลือกช่องสัญญาณภาพ 57

รูปที่ 2.41 แสดงวงจรที่เป็นตัวเลือกกล้องทั้ง 2 ตัว 58

รูปที่ 2.42 วงจร Drive Motor ที่ทำจาก Relay 59

รูปที่ 2.43 แสดงหลักการทำงานของมอเตอร์ดีซี 60

รูปที่ 2.44 โครงสร้างพื้นฐานมอเตอร์ 62

รูปที่ 2.45 บล็อกไดอะแกรมการทำงานของภาครับ-ส่งสัญญาณภาพ 64

รูปที่ 2.46 แสดงรูปภาพของกล้อง CCTV 64

รูปที่ 2.47 โครงสร้างของ CMOS 65

รูปที่ 2.48 แสดงชิพบรรจุอยู่ในตัวถัง Wire Bonded แบบ IC ทั่วไปพร้อมใช้งาน 65

รูปที่ 2.49 แสดงภาพที่ได้จากกล้อง CMOS 66

รูปที่ 2.50 แสดงกล้องสีทั้ง ด้านหน้า ด้านบน ด้านข้าง 67

รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานในส่วนหลักๆ 68




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป** **หน้า**

**รูปที่ 3.2 แสดงโครงสร้างด้านข้างและการทำงานในส่วนต่างๆ** 69
**รูปที่ 3.3 แสดงรูปด้านบนและรายละเอียด** 70
**รูปที่ 3.4 ล้อที่ใช้เป็นตัวขับเคลื่อน** 70
**รูปที่ 3.5 โครงสร้างของแขนที่ใช้ยกกล้องวงจรปิด** 71
**รูปที่ 3.6 ส่วนประกอบของตัวรับ-ส่งสัญญาณภาพและเสียง** 72
**รูปที่ 4.1 ผลการทดลองโปรแกรมวาดแผนที่จากเข็มทิศเมื่อหุ่นขยับไป 48 องศา** 80
**รูปที่ 4.2 ผลการทดลองโปรแกรมวาดแผนที่จากเข็มทิศเมื่อหุ่นขยับไป 161 องศา** 80





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| ตาราง | หน้า |
|---|---|
| ตารางที่ 2.1 แสดงค่า Emissivity | 9 |
| ตารางที่ 2.2 แสดงตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03 | 13 |
| ตารางที่ 2.3 แสดงค่าต่างๆใน IC PIC 16F877 และเบอร์อื่นๆ | 19 |
| ตารางที่ 2.4 แสดงกำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V2.0 | 50 |
| ตารางที่ 2.5 แสดงค่า Configuration ของตัวเครื่อง ET-RF24G V2.0 | 52 |
| ตารางที่ 2.6 แสดงการกำหนด Configuration | 54 |
| ตารางที่ 4.1 การส่งข้อมูลออกไปและทำการแสดงผลเมื่อมีการกดปุ่มแป้นพิมพ์ | 73 |
| ตารางที่ 4.2 คุณสมบัติโดยรวมของหุ่นยนต์กู้ภัย | 78 |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาของโครงงาน

ในปัจจุบันพบว่าเทคโนโลยีใหม่ๆมีบทบาทในการพัฒนาต่อสังคมรวมไปถึงการพัฒนาประเทศ เนื่องจากเทคโนโลยีเป็นพื้นฐานสำคัญอย่างหนึ่งในการพัฒนาทางด้านต่างๆ ได้แก่ ด้านการแพทย์ การศึกษา การสื่อสารมวลชน การรักษาความปลอดภัย รวมไปถึงการป้องกันประเทศเป็นต้น ซึ่งเทคโนโลยีที่กล่าวมาได้มีการพัฒนาอยู่ตลอดเวลา แต่ในที่นี้เราจะกล่าวถึงเทคโนโลยีที่กำลังได้ความสนใจอยู่ในขณะนี้คือ หุ่นยนต์ ซึ่งเราจะสังเกตเห็นได้จากสื่อต่างๆ ถึงพัฒนาการของเทคโนโลยี ซึ่งบุคคลทุกเพศทุกวัยให้ความสนใจเป็นอย่างมากหุ่นยนต์ที่มีในปัจจุบันก็มีการทำงานที่แตกต่างกันออกไปหลายประเภท เช่น หุ่นยนต์ที่ใช้ในโรงงานอุตสาหกรรม หุ่นยนต์ที่ใช้เป็นของเล่น หุ่นยนต์รักษาความปลอดภัย รวมไปถึงหุ่นยนต์กู้ภัยที่เราจะกล่าวถึง เป็นต้น

ประโยชน์ของหุ่นยนต์ก็จะขึ้นอยู่กับประเภทของการใช้งาน โดยในที่นี้เราจะกล่าวถึงหุ่นยนต์กู้ภัย (Rescue Robot) ซึ่งจะมีประโยชน์ในการช่วยเหลือผู้ประสบภัยในด้านการค้นหาผู้รอดชีวิตในอุบัติภัยต่างๆ และบอกถึงสถานะภาพผู้ได้รับบาดเจ็บในพื้นที่ที่เกิดภัยพิบัติ อุบัติเหตุ ซึ่งเกิดจากธรรมชาติหรือการก่อการร้าย ที่คนไม่สามารถเข้าไปถึงผู้บาดเจ็บได้และลดความเสี่ยงทั้งชีวิตและทรัพย์ที่จะเกิดกับผู้ช่วยเหลือ ที่จะเข้าไปช่วยชีวิตผู้ได้รับบาดเจ็บและยังสามารถรู้ถึงที่อยู่พิกัดที่แน่นอน และการช่วยเหลือผู้ได้รับบาดเจ็บก็จะมีความระมัดระวังและรอบครอบมากขึ้นเพื่อไม่ให้ผู้ที่ได้บาดเจ็บมีอาการสาหัสมากขึ้นหรือเป็นอันตรายถึงชีวิต ดังนั้นหุ่นยนต์กู้ภัยจึงมีความจำเป็นอย่างมากในปัจจุบัน จึงทำให้ต้องมีการวิจัยเพื่อพัฒนาหุ่นยนต์กันมากขึ้นและยังมีการแข่งขันในระดับประเทศและระดับโลกเพื่อให้ได้หุ่นยนต์กู้ภัยที่มีประสิทธิภาพสูงสุด นอกจากประโยชน์ที่ได้กล่าวมาข้างต้นแล้วหุ่นยนต์ประเภทนี้ยังท้าทายความสามารถของผู้ทำและยังต้องใช้ความรู้หลากหลายแขนงวิชามารวมกัน ทั้งด้านเครื่องกล อิเล็กทรอนิกส์ ไฟฟ้า และคอมพิวเตอร์เป็นต้น อีกทั้งยังได้สัมผัสและ ศึกษาอุปกรณ์ที่มีเทคโนโลยีใหม่ๆ ที่ไม่เคยพบเห็นได้จากห้องเรียน จึงทำให้คณะผู้จัดทำมีความสนใจที่จะนำหุ่นยนต์กู้ภัยตัวนี้มาเป็นโครงงาน ซึ่งโครงงานนี้อาจจะยังไม่สามารถนำไปใช้งานจริง ๆ ได้ เนื่องจากอุปกรณ์ที่นำมาสร้างยังไม่เหมาะสมกับสถานการณ์จริง แต่ก็สามารถนำไปพัฒนาให้ใช้งานจริง ๆ ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1.2 วัตถุประสงค์**

1. **เพื่อศึกษาหลักการทำงานและประยุกต์การใช้งานในการควบคุมมอเตอร์**
2. **เพื่อศึกษาหลักการทำงานและประยุกต์การใช้งานในด้านการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์อิเล็กทรอนิกส์ที่อยู่บนตัวหุ่นยนต์**
3. **เพื่อศึกษาหลักการทำงานและประยุกต์การใช้งานอุปกรณ์เซนเซอร์ต่าง ๆ**
4. **เพื่อศึกษากลไกโครงสร้างตัวถังของหุ่นยนต์**
5. เพื่อศึกษาหลักการทำงานและประยุกต์การใช้งานอุปกรณ์ที่เป็นเทคโนโลยีใหม่ ๆ
6. **เพื่อศึกษาวิธีการเขียนโปรแกรมควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ PIC16F877**
7. **เพื่อศึกษาวิธีการเขียนโปรแกรมด้วย Visual Basic**

**1.3 ขอบเขตของโครงงาน**

**หุ่นยนต์กู้ภัย จะสามารถเคลื่อนที่ได้ทุกสภาพผิวและยังสามารถเคลื่อนที่เข้าไปในพื้นที่ที่ไม่มีแสงสว่างหรือพื้นที่ที่คนไม่สามารถเข้าไปถึงได้ เนื่องจากมีการติดตั้งไฟส่องทางไว้ด้วย ส่วนการควบคุมการเคลื่อนที่จะควบคุมด้วยจอยสติก โดยจะสื่อสารข้อมูลกันแบบไร้สายระหว่างหุ่นยนต์กับคอมพิวเตอร์ ซึ่งจะสามารถสื่อสารได้ในระยะทาง 50-150 เมตร ในส่วนของกล้องจะสามารถปรับมุมมองได้ถึง 360° และสามารถตรวจวัดอุณหภูมิโดยไม่ต้องสัมผัสกับวัตถุหรือเหยื่อผู้เคราะห์ร้ายได้ ซึ่งค่าของอุณหภูมิจะแสดงผลเป็นตัวเลขดิจิตอล มีหน่วยเป็นองศาเซลเซียส ผ่านจอมอนิเตอร์ ส่วนของการตรวจสอบเสียงของผู้เคราะห์ร้ายและเสียงอื่น ๆ จะใช้ไมล์ไร้สาย และส่วนของการบอกตำแหน่งของวัตถุหรือเหยื่อผู้เคราะห์ร้าย เราจะใช้แสงเลเซอร์ยิงเป็นเส้นขนานกับหุ่นยนต์โดยจำกัดระยะห่างระหว่างสองเส้นให้มีค่าเท่ากับ 50 เซนติเมตร และมีระยะห่างออกจากด้านหน้าของตัวหุ่นยนต์ไป 50 เซนติเมตร การบอกตำแหน่งเราจะบอกโดยจะอ้างอิงกับแผนที่ของสถานที่นั้น ๆ**

**1.4 ประโยชน์ที่จะได้รับจากโครงงาน**

**ประโยชน์ที่จะได้รับจากโครงงานนี้ คือได้นำความรู้ความสามารถในด้านทฤษฎีและปฏิบัติที่ได้จากการศึกษามาประยุกต์ใช้งานได้จริง ๆ เช่น การออกแบบวงจรต่างๆ การวิเคราะห์วงจร การออกแบบหุ่นยนต์ และการเขียนโปรแกรมควบคุมหุ่นยนต์ เป็นต้น รวมไปถึงการทำงานของอุปกรณ์ต่างๆ ด้วยไมโครคอนโทรลเลอร์PIC16F877 การเขียนโปรแกรมด้วย Visual Basic ซึ่งเป็นโปรแกรมที่เราต้องศึกษาด้วยตนเอง นอกจากนี้ยังได้ความรู้และประสบการณ์ใหม่ ๆ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะทำให้เรามีความสามารถและทักษะเพิ่มขึ้น และยังจะนำไปใช้ในการทำงานในอนาคตได้อีกด้วย

**1.5 รายละเอียดโดยย่อของโครงงาน**

หุ่นยนต์กู้ภัย จะมีโครงสร้างที่สร้างมาจากอะลูมิเนียมตันหนา 4 mm. หุ่นยนต์จะมีล้อทั้งหมด 10 ล้อ ด้านละ 5 ล้อ เพื่อลดช่วงความห่างระหว่างล้อทำให้สามารถข้ามผ่านพื้นผิวขรุขระหรือพื้นที่ที่เป็นขั้นบันไดได้ โดยมีขนาด กว้าง×ยาว×สูง = 40×90×35 cm. โดยมีน้ำหนักประมาณ 30 กิโลกรัม หุ่นยนต์จะขับเคลื่อนโดยใช้ดีซีมอเตอร์ 2 ตัว ส่วนของล้อคู่หน้าและหลังสามารถยกได้ 90° เพื่อเพิ่มประสิทธิภาพในการปีนป่ายให้ดีขึ้น ส่งกำลังโดยมอเตอร์ดีซี 2 ตัวใช้โซ่และเฟืองเป็นตัวดึงแขนทั้งหน้าและหลัง ส่วนของฐานกล้องก็ทำมาจากอะลูมิเนียมเช่นเดียวกัน โดยใช้ดีซีมอเตอร์ 3 ตัว โดยจะทำเป็นแขนสามารถยกได้สูงประมาณ 70 เซนติเมตร และสามารถหมุนได้ 360° เพื่อการมองเห็นได้รอบทิศทางโดยไม่ต้องขยับหุ่น ส่วนของกล้องเป็นกล้อง CCTV 1 ตัว ที่เป็นภาพสีและกล้องสีธรรมดาอีก 1 ตัว ซึ่งจะติดตั้งไว้ที่จอของตัวตรวจจับอุณหภูมิ โดยจะมีการควบคุมผ่านวงจร วีดีโอซีเล็กเตอร์ (Vidco Sclector) โดยจะต่อเข้ากับโทรทัศน์ และการส่งสัญญาณจะส่งโดยใช้ Video Sender มาแสดงภาพที่โทรทัศน์ ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ จะควบคุมโดยส่งข้อมูลผ่านพอร์ตอนุกรม และส่งไปยังหุ่นยนต์

โดยจะใช้เครื่องรับเครื่องส่ง ET-RF24G V1 RS 232 Wireless ของ ETT ส่วนของเสียงจะใช้ไมค์คอนเดนเซอร์ เป็นตัวตรวจสอบเสียง และส่วนของการหาตำแหน่งของวัตถุและสร้างแผนที่จะใช้เข็มทิศอิเล็กทรอนิกส์ ส่วนของอุปกรณ์ต่าง ๆ ก็จะมีการติดตั้งไฟส่องทางที่ด้านหน้าของหุ่นยนต์ และในกล่องกล้อง เพื่อใช้ในพื้นที่ที่ไม่มีแสงหรือแสงสว่างไม่เพียงพอให้สามารถเห็นวัตถุที่อยู่ด้านหน้า ส่วนของโปรแกรมการทำงานจะใช้ Visual Basic ควบคุมส่วนที่เป็นซอร์ฟแวร์ (Software) โดยจะใช้จอยสติกควบคุมทิศทางการเคลื่อนที่ของตัวหุ่นยนต์ ให้เดินหน้าถอยหลัง เลี้ยวซ้ายเลี้ยวขวา ยกล้อหน้าล้อหลัง ยกแขนกล้องหมุนกล้อง เปิดปิดไฟส่องทาง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1.6 รายละเอียดของปริญญานิพนธ์**

โครงงาน "หุ่นยนต์กู้ภัย" (Rescue Robot) เป็นโครงงานที่รวบรวมเอาความรู้หลากหลายแขนงมาใช้สร้างโครงงานนี้ จึงทำให้รายละเอียดของเนื้อหา และทฤษฎีที่เกี่ยวข้องกับโครงงานมีมากพอสมควร โดยจะมีการแบ่งเนื้อหาออกเป็นบทต่าง ๆ ดังต่อไปนี้

**บทที่ 1 บทนำ** จะกล่าวถึงรายละเอียดความเป็นมาของโครงงาน, วัตถุประสงค์, ขอบเขตของโครงงาน, ประโยชน์ที่จะได้รับจากโครงงาน, รายละเอียดโดยย่อของโครงงาน, และรายละเอียดของปริญญานิพนธ์

**บทที่ 2 ทฤษฎีและหลักการ** โดยจะกล่าวถึงทฤษฎีและหลักการทำงานพื้นฐานต่าง ๆ ที่นำมาใช้ในการออกแบบ และจะแสดงให้เห็นถึงความสัมพันธ์ในการทำงานของอุปกรณ์ในแต่ละส่วนที่ติดตั้งบนตัวหุ่นยนต์

**บทที่ 3 การออกแบบส่วนต่าง ๆ ของหุ่นยนต์** โดยจะกล่าวถึงขั้นตอนในการออกแบบส่วนต่างๆ โครงสร้างตัวถังของหุ่นยนต์ การออกแบบเขียนโปรแกรมควบคุมการทำงานส่วนต่าง ๆ ของหุ่นยนต์ และการออกแบบวงจรต่างๆ ที่อยู่ภายในหุ่นยนต์

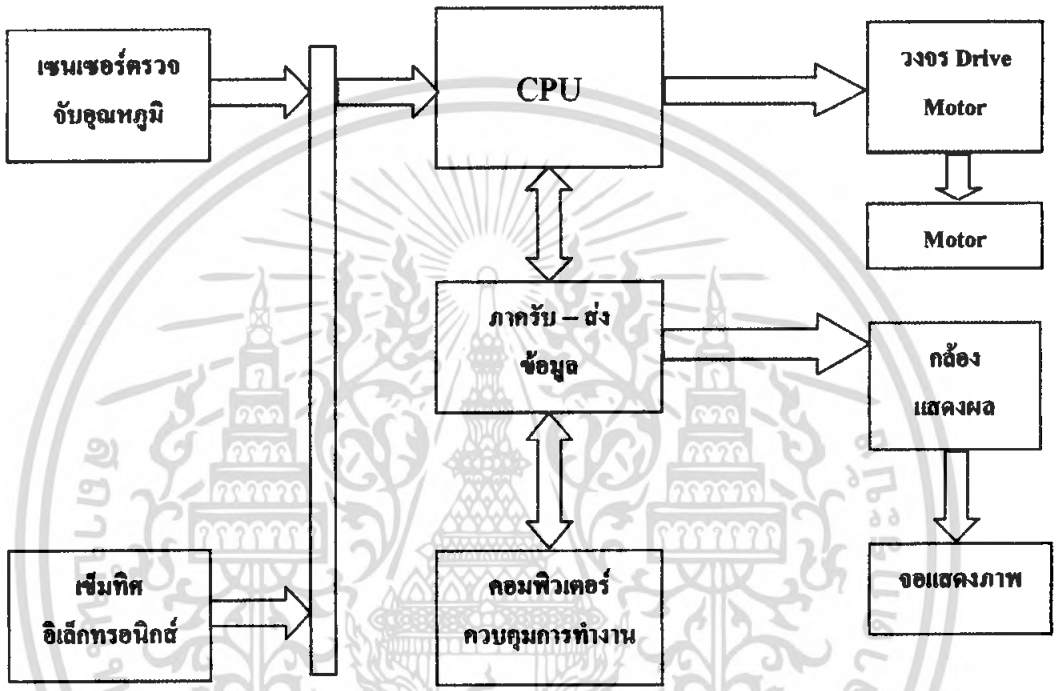
**บทที่ 4 ผลการทดสอบ** โดยจะกล่าวถึงผลจากการทดสอบวงจรและโปรแกรมต่าง ๆ ว่ามีการทำงานเป็นอย่างไรบ้าง

**บทที่ 5 สรุปและวิจารณ์** กล่าวถึงปัญหาที่พบในการทำงานเพื่อนำไปพัฒนาและปรับปรุงต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีและหลักการทำงาน

## 2.1 บล็อกไดอะแกรมการทำงาน




**รูปที่ 2.1 บล็อกไดอะแกรมแสดงการทำงานในส่วนต่างๆ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

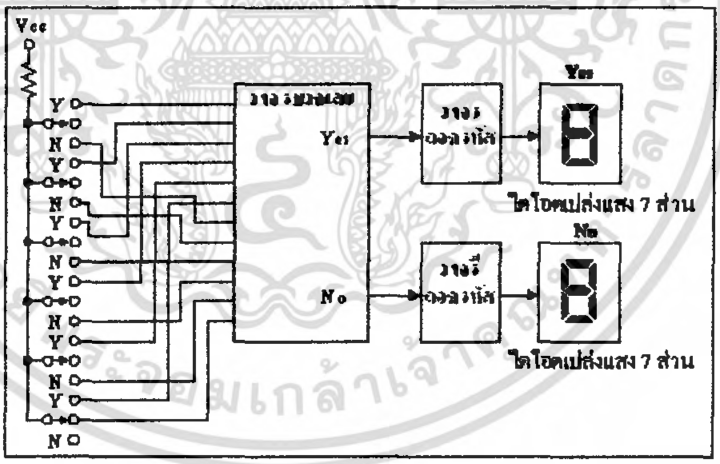
**2.2 อุปกรณ์ตรวจวัดประกอบด้วย 2 ส่วนดังนี้**

1. **เซนเซอร์ตัววัดอุณหภูมิ**
2. **เข็มทิศดิจิตอล**

**2.2.1 เซนเซอร์ตรวจจับอุณหภูมิ**

**2.2.1.1 หลักการทำงานของตัวตรวจจับอุณหภูมิ**

เทคโนโลยีการถ่ายภาพความร้อนเพื่อวัดอุณหภูมิโดยไม่ต้องสัมผัสช่วยตรวจหาบริเวณที่ร้อนผิดปกติของวัตถุต่างๆ ได้อย่างรวดเร็ว ในปัจจุบันได้นิยมใช้ในเชิงอุตสาหกรรมมากขึ้นโดยเฉพาะอย่างยิ่งในการบำรุงรักษาเครื่องจักรกลต่างๆ ซึ่งจะสามารถทราบถึงการทำงานของเครื่องจักรว่าผิดปกติหรือไม่ จะลดความเสียหายของเครื่องจักรและลดความเสี่ยงของบุคลากรภายในโรงงานอีกด้วย เครื่องตรวจจับอุณหภูมิเป็นเทคโนโลยีที่ไร้การสัมผัสโดยการวัดความยาวคลื่นอินฟราเรดเพื่อคำนวณอุณหภูมิซึ่งการแสดงผลภาพความร้อนที่อุณหภูมิแตกต่างกันนั้นจะแสดงด้วยสีที่แตกต่างกัน ภาพถ่ายความร้อนจะช่วยให้การตรวจสอบอุณหภูมิพื้นผิวนั้นๆ จะทำได้ง่ายและรวดเร็วขึ้นสามารถชี้จุดที่ร้อนผิดปกติได้อย่างแม่นยำ ซึ่งจุดที่มีความร้อนหรืออุณหภูมิที่เพิ่มสูงขึ้นผิดปกติมักจะบ่งบอกถึงความเสียหายที่เกิดขึ้นในไม่ช้า




**รูปที่ 2.2 บล็อกไดอะแกรมการทำงานของเครื่องตรวจจับอุณหภูมิ**

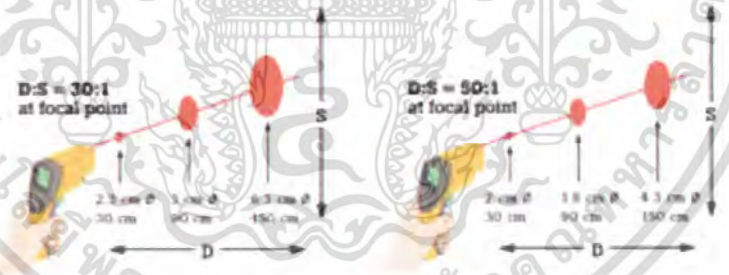
แต่ในที่นี้เรานำมาประยุกต์ใช้กับการกู้ภัยโดยจะติดตั้งอยู่บนตัวหุ่นยนต์เพื่อตรวจสอบอุณหภูมิของผู้ประสบภัยที่ติดอยู่ภายในพื้นที่ต่างๆ ว่ายังมีชีวิตอยู่หรือไม่ ถ้าผู้ประสบภัยยังมีชีวิตอยู่ตัวตรวจจับอุณหภูมิก็จะแสดงผลบนจอ LCD ในอุณหภูมิร่างกายปกติของคนเรา การใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือตรวจจับอุณหภูมิถึงแม้จะใช้ได้ง่าย แต่ก็มีพารามิเตอร์ที่สำคัญที่ต้องทำความเข้าใจอยู่สองค่าเพื่อให้สามารถวัดค่าอุณหภูมิได้อย่างแม่นยำ นั่นคือ

**1. ความละเอียดทางแสง (Optical resolution)** คือค่าอัตราส่วนของระยะทางต่อขนาดของจุดที่ถูกวัด ค่าอัตราส่วนยิ่งสูงโอกาสที่จะมีความคลาดเคลื่อนในการวัดก็น้อยลง ตัวอย่างเช่น Optical resolution 30 : 1 ถ้าต้องการได้ค่าการวัดที่ถูกต้องในระยะห่างจากจุดวัด 90 ซม. ขนาดเส้นผ่านศูนย์กลางของจุดที่จะทำการวัดไม่ควรเกิน 3 ซม.

**2. การแผ่รังสี (Emissivity)** คือความสามารถของวัตถุที่ทำให้เกิดพลังงานเป็นคลื่นอินฟราเรดออกมา ค่านี้ขึ้นอยู่กับชนิดของเนื้อวัสดุและลักษณะของพื้นผิว โดยมีค่า 1.0 สำหรับวัตถุดำมีดค่าการแผ่รังสีเปรียบได้กับค่าการสะท้อน หรือความมันวาวของพื้นผิวที่สังเกตในระดับที่ส่องด้วยกล้องไมโครสโคป ซึ่งวัตถุที่มีผิวขรุขระมีค่าการแผ่รังสีประมาณ 0.7 – 0.98 ในขณะเหล็กที่มันวาวจะมีค่า 0.1 วัตถุที่มีพื้นผิวมันจะสะท้อนแสงอินฟราเรดจากรอบๆ ตัวมัน รวมมากับรังสีอินฟราเรดจากตัววัตถุเองทำให้ค่าที่อ่านผิดไป เครื่องตรวจจับอุณหภูมิแบบอินฟราเรดโดยทั่วไปจะกำหนดค่าการแผ่รังสีคงที่ไว้ 0.95 เหมาะกับการวัดอุณหภูมิของพื้นที่ไม่สะท้อนแสงอย่างไรก็ตาม ในกรณีที่ต้องการวัดพื้นที่มันวาวสามารถแก้ไขโดยการนำผ้าสีดำมาคลุมปิดไว้เพื่อลดแสงสะท้อน




Optical Resolution เป็นค่าอัตราส่วนของระยะห่างจากวัตถุที่จะวัดอุณหภูมิกับขนาดของจุดที่จะวัด ตัวอย่างเช่น

- ค่า Optical Resolution 50:1 สามารถวัดอุณหภูมิของวัตถุ (พื้นที่) ขนาดเส้นผ่าศูนย์กลาง 2 ซ.ม. ได้ที่ระยะห่าง 30 ซ.ม.
- ค่า Optical Resolution ที่สูงขึ้น จะสามารถวัดอุณหภูมิขนาดที่เล็กได้ในระยะห่างที่ไกลยิ่งขึ้น

**รูปที่ 2.3 แสดงการวัดอุณหภูมิ ระยะทางเทียบกับวัตถุ**

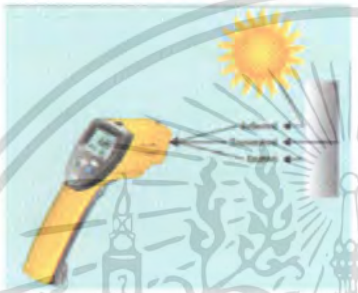
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Surface Emissivity

| Measured Surface | Emissivity |
|---|---|
| **METALS** | |
| **Aluminum** | |
| Oxidized | 0.2-0.4 |
| **Alloy A3003** | |
| Oxidized | 0.3 |
| Roughened | 0.1-0.2 |
| **Brass** | |
| Burnished | 0.3 |
| Oxidized | 0.5 |
| **Copper** | |
| Oxidized | 0.4-0.8 |
| Electrical | |
| Termina Blocks | 0.6 |
| **Haynes** | |
| Alloy | 0.3-0.8 |
| **Inconel** | |
| Oxidized | 0.7-0.95 |
| Sandblasted | 0.3-0.5 |
| Electropolished | 0.15 |
| **Iron** | |
| Oxidized | 0.5-0.9 |
| Rusted | 0.5-0.7 |
| **Iron, Cast** | |
| Oxidized | 0.6-0.95 |
| Unoxidized | 0.2 |
| Molten | 0.2-0.3 |
| **Iron, Wrought** | |
| Dull | 0.9 |
| **Lead** | |
| Rough | 0.4 |
| Oxidized | 0.7-0.9 |
| **Molybdenum** | |
| Oxidized | 0.2-0.9 |
| **Nickel** | |
| Oxidized | 0.2-0.5 |
| **Platinum** | |
| Black | 0.9 |

Table 2. Surface Emissivity (cont.)

| Measured Surface | Emissivity |
|---|---|
| **Steel** | |
| Cold Rolled | 0.7-0.9 |
| Ground Sheet | 0.4-0.6 |
| Polished Sheet | 0.1 |
| **Zinc** | |
| Oxidized | 0.1 |
| **NON-METALS** | |
| **Asbestos** | 0.95 |
| **Asphalt** | 0.95 |
| **Basalt** | 0.7 |
| **Carbon** | |
| Unoxidized | 0.8-0.9 |
| **Graphite** | 0.7-0.8 |
| **Carborundum** | 0.9 |
| **Ceramic** | 0.95 |
| **Clay** | 0.95 |
| **Concrete** | 0.95 |
| **Cloth** | 0.95 |
| **Glass** | |
| Plate | 0.85 |
| **Gravel** | 0.95 |
| **Gypsum** | 0.8-0.95 |
| **Ice** | 0.98 |
| **Limestone** | 0.98 |
| **Paper** | |
| (any color) | 0.95 |
| **Plastic** | |
| Opaque | 0.95 |
| **Soil** | 0.9-0.98 |
| **Water** | 0.93 |
| **Wood, (natural)** | 0.9-0.95 |



Emissivity เป็นค่าตัวเลขระหว่าง 0 จนถึง 1 ซึ่งแสดงถึงความสามารถในการแพร่พลังงานอินฟราเรด เพื่อให้แน่ใจว่าการวัดอุณหภูมิเที่ยงตรง ควรปรับค่านี้ให้ตรงกับผิวที่จะวัด

**รูปที่ 2.4 แสดงถึงความสามารถในการแผ่รังสีอินฟราเรด**

**2.2.1.2 เครื่องตรวจจับอุณหภูมิที่นำมาใช้เป็นของ Fluke รุ่น 62 ซึ่งมีคุณสมบัติดังนี้**

- วัดอุณหภูมิได้ถึง 500°C
- ความแม่นยำ 1 % Repeatability 0.5 %
- มีเลเซอร์ช่วยชี้ตำแหน่งการวัด
- คงค่าอุณหภูมิที่จอไว้อ่านภายหลังได้
- แสดงค่าปัจจุบัน / ค่าสูงสุด
- มีขนาดเล็ก น้ำหนักเบา

**รูปที่ 2.5 แสดงเครื่องตรวจจับอุณหภูมิที่นำมาใช้งาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1 แสดงค่า Emissivity**

| Measured surface | | Emissivity |
|---|---|---|
| **Metals** | | |
| Aluminum | Oxidized | 0.2-0.4 |
| Alloy A3003 | Oxidized | 0.3 |
| | Roughened | 0.1-0.3 |
| Brass | Burnished | 0.3 |
| | Oxidized | 0.5 |
| Copper | Oxidized | 0.4-0.8 |
| | Electrical terminal blocks | 0.5 |
| Haynes | Alloy | 0.3-0.8 |
| Inconel | Oxidized | 0.7-0.95 |
| | Sandblasted | 0.3-0.5 |
| | Electoropolished | 0.15 |
| Iron | Oxidized | 0.5-0.9 |
| | Rusted | 0.5-0.7 |
| Iron, Cast | Oxidized | 0.6-0.95 |
| | Unoxidized | 0.2 |
| | Molten | 0.2-0.3 |
| Iron, Wrought | Dull | 0.9 |
| Lead | Rough | 0.4 |
| | Oxidized | 0.2-0.6 |
| Molybdenum | Oxidized | 0.2-0.6 |
| Nickel | Oxidized | 0.2-0.5 |
| Platinum | Black | 0.9 |
| Steel | Cold-Rolled | 0.7-0.9 |
| | Ground Sheet | 0.4-0.6 |
| | Polished Sheet | 0.1 |
| Zinc | Oxidized | 0.1 |

| Non-Metals | | |
|---|---|---|
| Asbestos | | 0.95 |
| Asphalt | | 0.95 |
| Basalt | | 0.7 |
| Carbon | Unoxidized | 0.8-0.9 |
| Graphite | | 0.7-0.8 |
| Carborundum | | 0.9 |
| Ceramic | | 0.95 |
| Clay | | 0.95 |
| Concrete | | 0.95 |
| Cloth | | 0.95 |
| Glass | Plate | 0.85 |
| Gravel | | 0.95 |
| Gypsum | | 0.8-0.95 |
| Ice | | 0.98 |
| Limestone | | 0.98 |
| Paper | (any color) | 0.95 |
| Plastic | Opaque | 0.95 |
| Soil | | 0.9-0.98 |
| Water | | 0.93 |
| Wood | (natural) | 0.9-0.95 |

### 2.2.2 เข็มทิศดิจิตอล

#### 2.2.2.1 หลักการทำงานของเข็มทิศดิจิตอล

โมดูลเข็มทิศดิจิตอล Digital Compass Module (CMPS03) เป็นผลงานของ Devantech (www. radio-electronics.co.uk) ออกแบบมาเพื่อช่วยในกำหนดทิศทางการเคลื่อนที่ของหุ่นยนต์อัตโนมัติ และนำมาใช้ในการสร้างเครื่องมือวัดและตรวจสอบทิศระบบอิเล็กทรอนิกส์ โดยหัวใจสำคัญของโมดูล CMPS03 คือ ตัวตรวจจับสนามแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัว เพื่อให้มีความไวเพียงพอในการตรวจจับสนามแม่เหล็กโลก (Earth magnetic field) และไมโครคอนโทรลเลอร์เพื่อรับสัญญาณจากตัวตรวจจับมาประมวลผลเป็นข้อมูลดิจิตอล และสัญญาณพัลส์สำหรับแจ้งผลการวัดทิศทาง

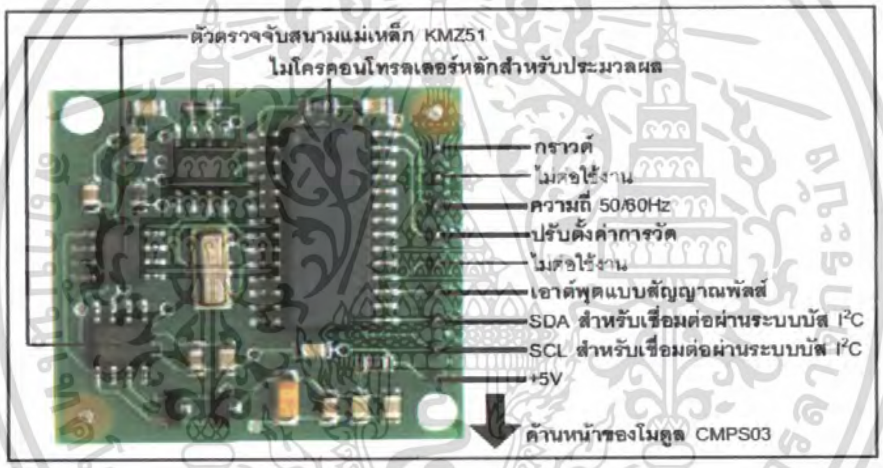
#### 2.2.2.2 ตำแหน่งขาและการต่อใช้งาน

ในรูปที่ 2.6 แสดงรูปร่างหน้าตาและการจัดขาของ CMPS03 โมดูลเข็มทิศดิจิตอล จะเห็นว่าเป็นแผงวงจรที่มีคอนเน็กเตอร์ต่อออกมาเพื่อให้เชื่อมต่อไปใช้งาน อย่างไรก็ตาม เพื่ออำนวยความสะดวกแก่ผู้ใช้งานกับบอร์ดควบคุมหุ่นยนต์ของบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

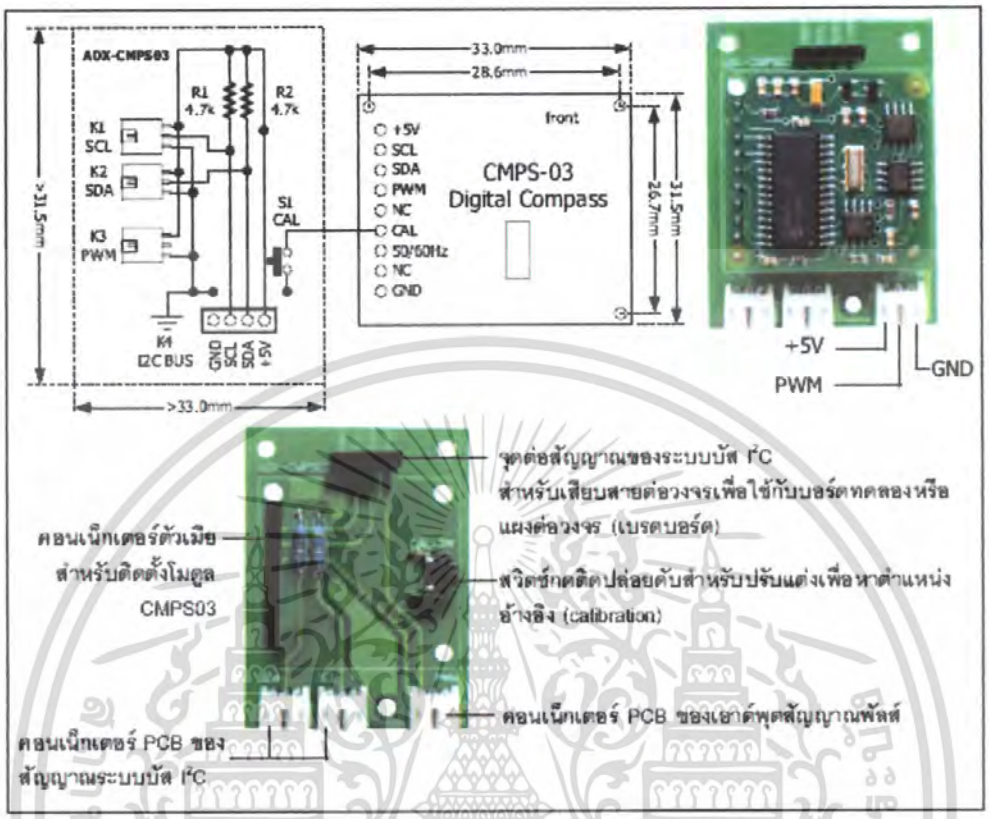
จำกัด (i-nex :เป็นตัวแทนจำหน่ายสินค้าของ Devantech ในประเทศไทยอย่างเป็นทางการ) จึงได้พัฒนาบอร์ดอะแดปเตอร์รุ่น ADX-CMPS03 เพื่อให้นำโมดูล CMPS03 มาติดตั้ง (โดยบอร์ด ADX-CMPS03 ต้องจัดซื้อแยก)

บนบอร์ด ADX-CMPS03 ได้จัดเตรียมคอนเน็กเตอร์ PCB 3 ขาตัวผู้สำหรับเชื่อต่อกับบอร์ดควบคุมหุ่นยนต์ และคอนเน็กเตอร์ IDC ตัวเมียแถวเดี่ยว 4 ขาสำหรับเสียบสายต่อวงจรเบอร์ AWG#22 เพื่อต่อกับแผงวงจรหรือเบรดบอร์ด นอกจากนั้นยังมีสวิตช์กดสำหรับปรับตั้งค่า (calibration) เพื่อกำหนดตำแหน่งทิศอ้างอิง โดยวงจรของบอร์ด ADX-CMPS03 แสดงในรูปที่ 2.7




**รูปที่ 2.6 แสดงรูปร่างและตำแหน่งขาสำหรับการต่อใช้งานของเข็มทิศดิจิตอล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.7 แสดงวงจรของบอร์ด ADX-CMPS03 และการเชื่อมต่อกับโมดูล CMPS03**

### 2.2.2.3 การปรับแต่งค่าทิศอ้างอิงแก่โมดูล CMPS03

เพื่อให้การวัดทิศทางของโมดูล CMPS03 มีความแม่นยำมากที่สุด จึงมีอินพุตสำหรับปรับแต่งค่าทิศทางอ้างอิง ทั้งนี้เพื่อประโยชน์ในการกำหนดทิศทางอ้างอิงเฉพาะสำหรับผู้ใช้งาน โดยต้องป้อนสัญญาณลอจิก "0" เข้าที่ขาอินพุตสำหรับปรับแต่งโมดูล CMPS03 ซึ่งก็คือขา 6 หากใช้บอร์ด ADX-CMPS03 กับโมดูล CMPS03 จะมีสวิตช์กดติดปล่อยดับติดตั้งไว้ให้แล้ว การปรับแต่งมีขั้นตอนดังนี้

(1) วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศเหนือ กดสวิตช์ 1 ครั้ง

(2) วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศตะวันออก กดสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศใต้ กดสวิตช์ 1 ครั้ง

(4) วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศตะวันตก กดสวิตช์

เป็นอันสิ้นสุดการปรับตั้งค่าทิศทางอ้างอิงของโมดูล CMPS03 โดยโมดูลค่าอ้างอิงนี้ไว้ในหน่วยความจำอีอีพรอมและไม่ต้องปรับตั้งค่าใหม่อีกเมื่อจ่ายไฟเลี้ยงครั้งใหม่

**2.2.2.4 การอ่านค่าสัญญาณเอาต์พุตของโมดูล CMPS03**

**2.2.2.4.1 การอ่านค่าทิศทางจากเอาต์พุตสัญญาณพัลส์**

การอ่านค่าสัญญาณในโหมดนี้ เป็นการนำค่าความกว้างพัลส์ที่ได้จากเอาต์พุตสัญญาณพัลส์ของโมดูล CMPS03 มาระบุตำแหน่งองศา จาก 0 ถึง 539.9 องศา โดยมีย่านของค่าความกว้างสัญญาณพัลส์จาก 1 มิลลิวินาทีไปจนถึง 36.99 มิลลิวินาที มีความละเอียด 0.1 มิลลิวินาทีต่อองศา ในสัญญาณพัลส์แต่ละไซเกิล มีช่วงลอจิก "0" กว้าง 65 มิลลิวินาที

ดังนั้นในการนำสัญญาณพัลส์มาประมวลผลเป็นค่ามุม จึงต้องใช้ความกว้างของสัญญาณพัลส์เป็นหลักในการคำนวณหาค่ามุมที่โมดูล CMPS03 วัดได้

**- ตัวอย่างโปรแกรมที่ใช้ควบคุมสำหรับเบสิกแสตมป์ 2XS และ i-Stamp**

การใช้งานร่วมกับเบสิกแสตมป์ 2XS และ i-Stamp นั้น จะใช้คำสั่ง PULSIN ในการนับสัญญาณพัลส์ โดยจะเพิ่มค่าการนับขึ้นทุกๆ 0.8 ไมโครวินาที ดังนั้นที่ความกว้างของพัลส์ที่ 1 มิลลิวินาทีสำหรับตำแหน่ง 0 องศา เบสิกแสตมป์ 2XS และ i-Stamp จะนับค่าได้เท่ากับ 1,250 จึงสามารถใช้ค่านี้เป็นจุดอ้างอิงที่ 0 องศา เมื่อต้องการทราบค่ามุมที่แท้จริงให้นำค่ามุมที่นับได้ลบด้วย 1,250 แล้วหารด้วย 125 ก็จะได้ค่ามุมในหน่วยองศาที่ต้องการ รายละเอียดของโปรแกรมแสดงในโปรแกรมที่ 1

ที่ความกว้างพัลส์สูงสุดคือ 36.99 มิลลิวินาที ค่าที่นับได้จากคำสั่ง PULSIN เท่ากับ 46,237 เมื่อลบด้วย 1,250 แล้วหารด้วย 125 เพื่อแปลงเป็นองศา ค่าสูงสุดที่แสดงเป็นผลลัพธ์ได้คือ 359 เป็นค่าหน่วยองศาสูงสุดนั่นเอง

**<u>หมายเหตุ : ในบางกรณี ค่าที่อ่านได้สูงสุดจากคำสั่ง PULSIN อาจไม่ถึง 46,237 ผู้ใช้งานสามารถปรับเปลี่ยนการคำนวณใหม่ให้ได้มุมเป็น 359.9 องศาได้</u>**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.2 แสดงตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03**

| ตำแหน่งรีจิสเตอร์ | รายละเอียด |
|---|---|
| 0 | ตัวเลขแสดงรุ่นของบอร์ด CMP03 |
| 1 | ส่งค่าตำแหน่งแบบหยาบ (0-255) |
| 2,3 | ส่งค่าตำแหน่งแบบละเอียดด้วยตัวเลข 16 บิต (0-3599) สามารถแปลงค่าเพื่อแสดงองศา 0-359.9 องศาได้โดยตรง |
| 4,5 | สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของ Sensor1 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 6,7 | สำหรับตรวจสอบค่าภายใน โดยจะแสดงค่าความต่างของ Sensor2 เป็นตัวเลข 16 บิตแบบคิดเครื่องหมาย |
| 8,9 | แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value1) เป็นตัวเลข 16 บิตแบบเครื่องหมาย |
| 10,11 | แสดงค่าตัวเลขการปรับแต่งภายใน (calibration value2) เป็นตัวเลข 16 บิตแบบเครื่องหมาย |
| 12,13 | ไม่ใช้งานอ่านค่าได้เป็น 0 |
| 14 | ไม่ใช้งาน ไม่ได้กำหนดค่าไว้ |
| 15 | คำสั่งสำหรับการปรับแต่งค่า โดยเมื่อต้องการปรับแต่งค่า ต้องเขียนข้อมูล 255 เข้าที่รีจิสเตอร์ตำแหน่งนี้ |

**2.2.2.4.2 การอ่านค่าทิศทางเป็นข้อมูลดิจิตอลผ่านระบบบัส $I^2C$**

**การอ่านค่าจากโมดูล CMPS03 ให้ได้ค่าที่มีความแม่นยำสูงควรเลือกเอาต์พุตข้อมูลดิจิตอลผ่านระบบบัส $I^2C$ โดยข้อมูล CMPS03 สามารถส่งข้อมูลของตำแหน่งองศามาที่ความละเอียดสูงสุด 0.1 องศาโดยไม่จำเป็นต้องมีการคำนวณหรือแปลงค่าใดๆ อีก**

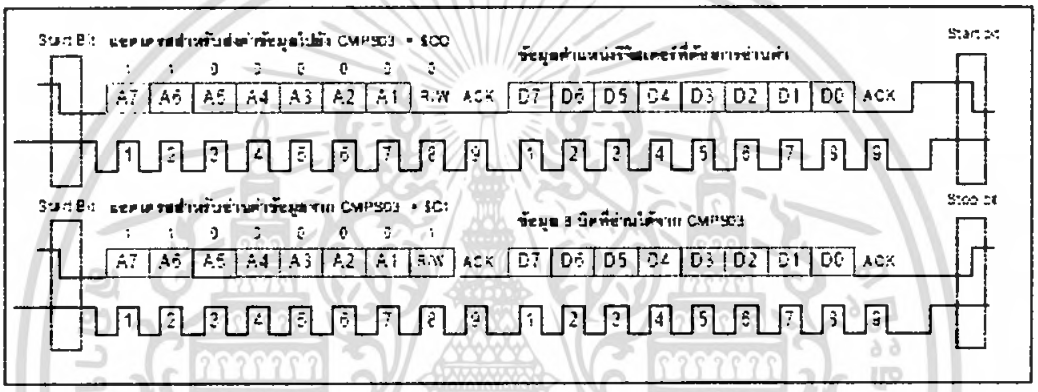
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **รูปแบบการสื่อสารข้อมูลบัส I²C**

บัส I²C เชื่อมต่อกับไมโครคอนโทรลเลอร์โดยใช้สายสัญญาณ 2 เส้น ได้แก่ขา SDA (รับและส่งข้อมูล) และ SCL (ขาสัญญาณนาฬิกา) โดยขาสัญญาณทั้งสองจะต้องต่อตัวต้านทานพูลอัปต่อไว้เพื่อกำหนดสถานะลอจิก "1" ให้กับระบบบัส

- **ลำดับขั้นการติดต่อ**

ค่าแอดเดรสของโมดูล CMPS03 คือ $C0 สำหรับการส่งข้อมูล และ $C0 สำหรับการอ่านค่าข้อมูล โดยขั้นตอนการติดต่อกับโมดูล CMPS03 เพื่ออ่านข้อมูลมีดังนี้

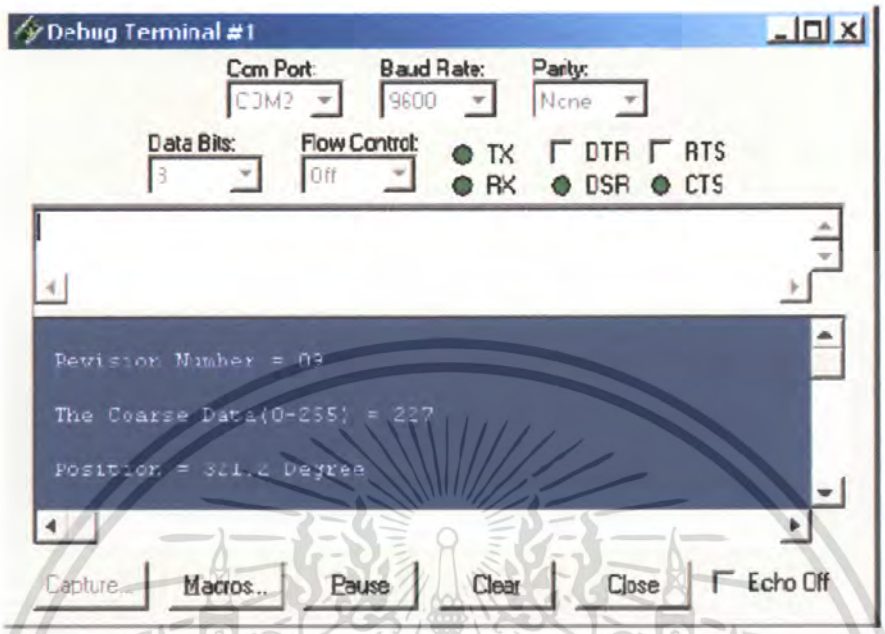



**รูปที่ 2.8 แสดงไทมิ่งไดอะแกรมของการติดต่อสื่อสารกับโมดูล CMPS03 ผ่านระบบ บัส I²C**

1. ส่งบิตเริ่มต้นหรือ Start bit เพื่อแจ้งให้ระบบบัส I²C เตรียมพร้อมรับข้อมูล
2. ส่งค่าแอดเดรส C0 เพื่อระบุว่าต้องการติดต่อเพื่อเขียนข้อมูลไปยังกับโมดูล CMPS03
3. ส่งค่าตำแหน่งรีจิสเตอร์ภายในโมดูล CMPS03 ที่ต้องการอ่านค่า ซึ่งมีรายละเอียดแสดงในตารางที่ 2.2
4. ส่งค่าแอดเดรส C1 เพื่อต้องการระบุค่าอ่านข้อมูลจากโมดูล CMPS03
5. อ่านค่าข้อมูลจากโมดูล CMPS03 มาเก็บไว้ในหน่วยความจำ
6. ส่งบิตหยุดหรือ Stop เพื่อหยุดการสื่อสารข้อมูล และกำหนดให้บัสอยู่ในสภาวะบัสว่าง

จากลำดับขั้นการติดต่อสื่อสารข้างต้น สามารถนำมาเขียนเป็นโปรแกรมตัวอย่างเพื่ออ่านข้อมูลจากโมดูล CMPS03 โดยใช้เบสิกแสตมป์ 2SX หรือ i-Stamp ได้ดังแสดงในโปรแกรมที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.9 แสดงหน้าจอของ Debug terminal แสดงข้อมูลที่อ่านได้จากโมดูล CMPS03**

**- การทำงานของโปรแกรมที่ 2 ติดต่อกับโมดูล CMPS03 ผ่านระบบบัส $I^2C$ ด้วย i-Stamp**

เนื่องจากเบสิกแสตมป์ 2SX และ i-Stamp ไม่มีคำสั่งติดต่อกับระบบบัส $I^2C$ ดังนั้น โปรแกรมติดต่อกับโมดูล CMPS03 ผ่านระบบบัส $I^2C$ จึงค่อนข้างยาว เนื่องจากต้องสร้างโปรแกรมย่อยสำหรับการสื่อสารข้อมูลกับระบบบัส $I^2C$ หลังจากนี้ผู้ใช้งานสามารถนำโปรแกรมย่อยนี้ไปประยุกต์ใช้งานกับอุปกรณ์ตัวอื่นๆ ที่ได้ใช้การติดต่อระบบบัส $I^2C$ ได้ทันที สำหรับขั้นตอนการทำงานหลักๆ ของโปรแกรมที่ 2 มีดังนี้

1. กำหนดรีจิสเตอร์เป็น 0 เพื่อติดต่ออ่านค่าเวอร์ชันของโมดูล CMPS03 จากนั้นมาอ่านค่ามาแสดงที่หน้าต่าง Debug terminal

2. ให้โปรแกรมวนลูปที่โปรแกรมหลัก จากนั้นส่งค่ารีจิสเตอร์เท่ากับ 1 เพื่ออ่านค่าข้อมูลแบบหยาบออกมา แล้วนำมาแสดงผลที่หน้าต่าง Debug terminal

3. ค่ารีจิสเตอร์เท่ากับ 2 เพื่ออ่านค่าข้อมูลแบบละเอียด จากนั้นสั่งอ่านค่าข้อมูลจากบัส $I^2C$ แบเวิร์ด (อ่านข้อมูลออกมา 16 บิต)

4. นำค่าข้อมูลที่ได้หารด้วย 10 ก่อนเพื่อแปลงค่าที่ได้เป็นองศา แสดงที่หน้าต่าง Debug terminal จากนั้นนำค่าหลักสุดท้ายมาแสดง ซึ่งตำแหน่งสุดท้ายเป็นตำแหน่งของจุดทศนิยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การปรับแต่งค่าของโมดูล CMPS03 ผ่านทางระบบบัส $I^2C$

การปรับแต่งค่าทำได้โดยการส่งค่า 0xFF ไปยังรีจิสเตอร์ 15 ของโมดูล CMPS03 โดยจะต้องส่งค่า 4 ครั้งและระบุทิศทางหลักๆ 4 ทิศทางเช่นเดียวกับการกำหนดค่าด้วยสวิตช์โดยตรง มีขั้นตอนดังนี้

1. วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศเหนือ จากนั้นเขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
2. วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศตะวันออก เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
3. วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศใต้ กดสวิตช์ เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15
4. วางโมดูล CMPS03 ขนานกับพื้น หักด้านหน้าของโมดูลไปทางทิศตะวันตก กดสวิตช์ เขียนค่า 255 (0xFF) ไปยังรีจิสเตอร์ 15

หลังจากปรับแต่งค่าแล้วค่าที่ปรับแต่งจะเก็บไว้ที่หน่วยความจำอีอีพรอม ดังนั้น แม้ไม่จ่ายไฟให้กับตัวบอร์ด ข้อมูลที่ปรับแต่งแล้ว จะยังคงอยู่ต่อไป

**2.2.3.5 คุณสมบัติ โมดูลเข็มทิศดิจิตอลDigital Compass Module**

- ใช้ไฟเลี้ยง +5V ต้องการกระแสไฟฟ้า 20mA
- ใช้ตัวตรวจจับกระแสแม่เหล็กเบอร์ KMZ51 ของ Philips จำนวน 2 ตัว เพื่อให้สามารถตรวจจับสนามแม่เหล็กโลกได้อย่างสมบูรณ์และมีความละเอียดมากพียงพอ
- ความละเอียดของมุม 0.1 องศา
- ค่าความผิดพลาด 3-4 องศาโดยประมาณ หลังจากการปรับแต่ง
- เอาต์พุตแบบสัญญาณพัลส์ ความกว้าง 1 ถึง 37 มิลลิวินาที โดยมีอัตราเพิ่มครั้งละ 0.1 มิลลิวินาที
- เอาต์พุตข้อมูลดิจิตอลผ่านการติดต่อระบบบัส $I^2C$ รองรับสัญญาณนาฬิกาความถี่สูงถึง 1MHz โดยให้ข้อมูล 2 รูปแบบคือ 0-255 และ 0-3599
- ขนาดเล็กเพียง 32 × 35 มิลลิเมตร
- สื่อสารกับไมโครคอนโทรลเลอร์ยอดนิยมได้ทุกตระกูล อาทิ เบสิกแสตมป์ 2SX/2P, PIC, MCS-51, PSoC, 68HC11 ทั้งผ่านระบบบัส $I^2C$ และด้วยการวัดสัญญาณพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


**2.2.2.6 อุปกรณ์เสริม**

- บอร์ด ADX-CMPS03 ซึ่งเป็นบอร์ดอะแดปเตอร์สำหรับอำนวยสะดวกในการเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์และแผงต่อวงจรหรือเบรดบอร์ด



**รูปที่ 2.10 บอร์ด ADX-CMPS03**

- สาย PCB3A สำหรับการเชื่อมต่อกับบอร์ดควบคุมหุ่นยนต์ของ i-nex

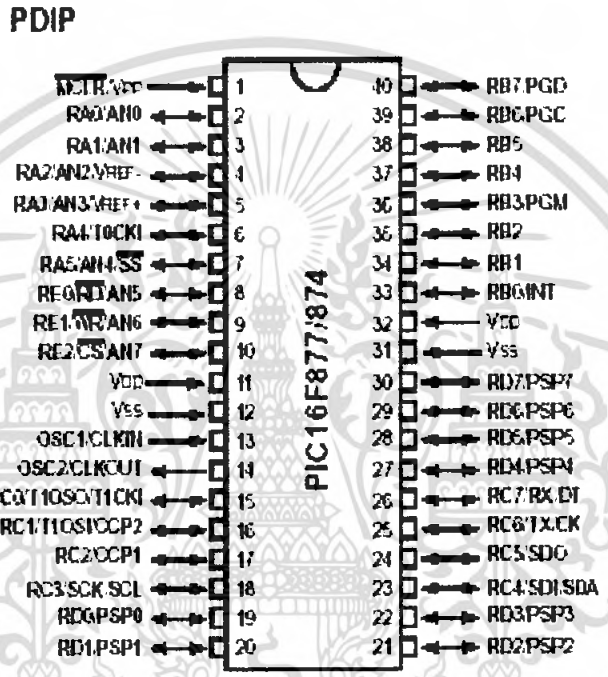
**รูปที่ 2.11 สายเชื่อมต่อ PCB3A**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
82429

## 2.3 CPU (ไมโครคอนโทรลเลอร์ PIC 16F877)

### 2.3.1 PIC 16F877

พื้นฐานการทำงานของไมโครคอนโทรลเลอร์ก็ คือ ระบบดิจิตอลโดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกต่างๆ มากมาย




**รูปที่ 2.12 แสดงตำแหน่งขา IC PIC 16F877**

จากรูปขาของ PIC16f877 กับ 16F874 เหมือนกันแต่ความสามารถบางประการจะไม่เหมือนกัน สามรถดูได้จาก data sheet แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออกเป็น PORT A , PORT B , PORT C , PORT D, PORT E โดยพื้น ฐานแล้วพอร์ตแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุตเป็นดิจิตอล ยกเว้น PORT A และ PORT E ที่สามารถทำงานเป็นตัวรับสัญญาณอนาลอกแปลงเป็นค่าดิจิตอลเพื่อนำมาวัดปริมาณทางฟิสิกส์ต่างๆ ที่เห็นได้อย่างชัดเจน คือ นำมาวัดความต่างศักย์ครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.3 แสดงค่าต่างๆใน IC PIC 16F877 และเบอร์อื่นๆ**

| ชิพ | หน่วยความจำโปรแกรม | หน่วยความจำข้อมูล | | CAN Module | จำนวน I/O (บิต) | OSC (MHz) | Timer | PLL |
|---|---|---|---|---|---|---|---|---|
| | | RAM | EEPROM | | | | | |
| 16F84 | 1K Word | 68 | 64 | ไม่มี | 13 | 4-10 | 1 | ไม่มี |
| 16F877 | 8K Word | 368 | 256 | ไม่มี | 33 | 4-20 | 3 | ไม่มี |
| 16F422 | 16 KB | 768 | 256 | ไม่มี | 34 | 40 | 4 | มี |
| 16F458 | 32 KB | 1536 | 256 | มี | 34 | 40 | 4 | มี |

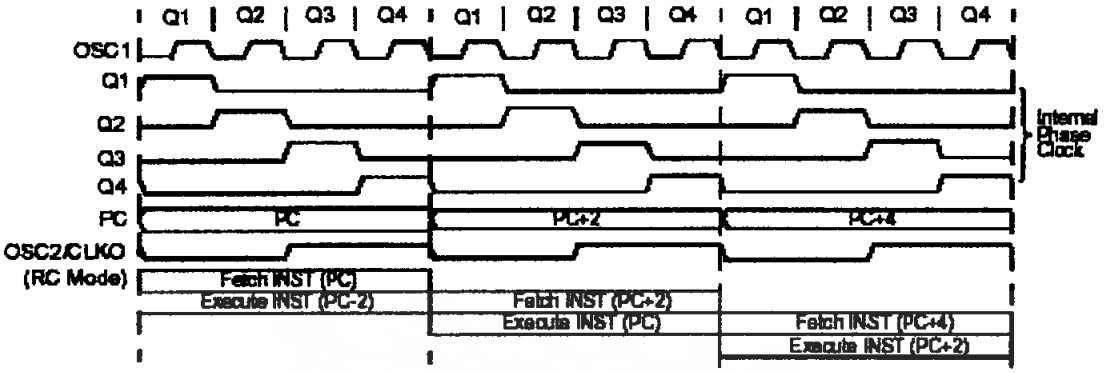
**หมายเหตุ**

1. 16F84/877 นั้น ใช้หน่วยความจำโปรแกรมขนาด 14 บิต ต่อ 1 คำ (Word)
2. CAN Module คือ
3. PLL (Phase Lock Loop) คือ วงจรที่สร้างความถี่ของสัญญาณนาฬิกาเป็น 4 เท่าจาก XTAL ดังนั้น ถ้าเราเลือกโหมด PLL ก็จะใช้ XTAL ได้ไม่เกิน 10MHz

**2.3.2 สัญญาณนาฬิกา**

PIC จะใช้สัญญาณนาฬิกา โดยมองเป็นลักษณะของ วงรอบ (Cycle) ซึ่งระบุเอาไว้ว่า 1 คำสั่งนั้นจะประกอบไปด้วย 1-2 วงรอบ โดยแต่ละวงรอบนั้นจะแบ่งเป็น 4 ส่วน คือ Q1, Q2, Q3 และ Q4 ด้วยเหตุนี้ ความเร็วโดยรวมของ PIC จึงเท่ากับ ค่าความถี่ของสัญญาณนาฬิกา หาร ด้วย 4 หรือหากจะพิจารณาความเร็วของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถประมวลผลต่อหนึ่งคำสั่งเท่ากับ 1/4 เท่าของความถี่ออสซิเลเตอร์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.13 แสดงสัญญาณนาฬิกาของ PIC**

**Oscillator Type**

- LP (Low Power Crystal) คริสตอลพลังงานต่ำ
- XT (Crystal/Resonator) คริสตอล หรือ เรโซเนเตอร์
- HS (High Speed Crystal/Resonator) คริสตอง หรือ เรโซเนเตอร์ความเร็วสูง
- RC (External Resistor/Capacitor) วงจร RC ภายนอก
- EC (External Clock) สัญญาณนาฬิกาจากภายนอก
- H4 (HS + PLL : High Speed Crystal/Resonator with PLL enabled)คูณ 4 PLL คือจะทำการคูณสัญญาณนาฬิกาที่เข้ามา ด้วย 4 เช่น OSC ความถี่ 10 MHz เมื่อผ่านกระบวน การนี้จะทำให้ได้ความถี่เท่ากับ 40 MHz(คุณสมบัตินี้มีเพียงบางเบอร์เท่านั้น เช่น 18F458)
- EC Clockout ใช้คริสตอลภายนอก และกำหนดให้ขา OSC2 เป็นเอาต์พุต
- INTRC (IN) วงจร RC ภายในไมโครคอนโทรลเลอร์
- EXTRC (ER) วงจร RC ภายในไมโครคอนโทรลเลอร์และกำหนดให้ขา OSC2 เป็นเอาต์พุต
- EXTRC Clockout วงจรตัวต้านทานภายนอกโดยกำหนดค่าความถี่จากค่าความต้านทานที่นำมาต่อ และ กำหนดให้ขา OSC2 เป็นเอาต์พุตส่วนการกำหนดชนิดของออสซิเลเตอร์ในโปรแกรมสมารถทำได้โดยใช้คำสั่ง #fuses Oscillator Type เช่น #fuses XT คือใช้คริสตอล หรือ เรโซเนเตอร์

### 2.3.3 การจัดขาของไอซี

หลังจากที่ทำความเข้าใจเกี่ยวกับสัญญาณนาฬิกากันไปแล้ว เรามาดูเรื่องการจัดขาของชิพกันดีกว่าครับ ในรูปนั้นถ้าขาใดเป็น Active Low ผมจะใส่เครื่องหมาย' เอาไว้ด้านหลังนะครับ เช่น MCLR' ก็คือ ขาที่ Active แบบ Low นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.14 แสดงโครงสร้างขาของ 16F877**

**2.3.4 คุณสมบัติของ 16F877**

1. มีคำสั่งให้ใช้งาน 35 คำสั่ง
2. คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 Cycle
3. ทำงานได้สูงสุดที่ 20MHz (PIC16F877-20/P ไม่ใช่ 16F877-04/p)
4. ทำงานแบบ Pipe-line (มี 2 ท่อ) ทำให้ ณ เวลาหนึ่งทำงาน 2 อย่างพร้อมๆกันได้
5. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8KWord (1 word=14 บิต)
6. มี RAM ขนาด 368 ไบต์ ให้เราใช้งาน
7. มี EEPROM ขนาด 256 ไบต์
8. ตอบสนองกับอินเตอร์รัพได้ทั้งหมด 14 แหล่ง
9. มี Stack ให้ใช้ได้สูงสุด 8 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up timer

11. Watchdog timer

12. มีระบบ Code Protection กันใครก๊อบเอาไปแจก

13. มีโหมดประหยัดเงิน (ประหยัดพลังงาน ก็ประหยัดเงินไปด้วย)

14. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้

15. สามารถโปรแกรมด้วยไฟ +5VDC ได้

16. ใช้การโปรแกรมแบบ In-Circuit Serial Programming

17. ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5VDC

18. Current Sink และ Current Source อยู่ที่ 25mA (ขับ LED ได้สบายเลยล่ะครับ ... ไม่เหมือน MCS-5!)

19. มี Timer/Counter 3 ตัว

20. มีโมดูล Capture/Compare/PWM อีก 2 ชุด (มี PWM นี่คุม DC Motor ได้)

21. มี A-TO-D Converter แบบ 10 บิต จำนวน 8 ช่องนำเข้า ในตัวเอง (ประหยัดเงินค่าไอซี A-TO-D Converter ไปหลายบาท)

22. มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232 หรือดีกว่า

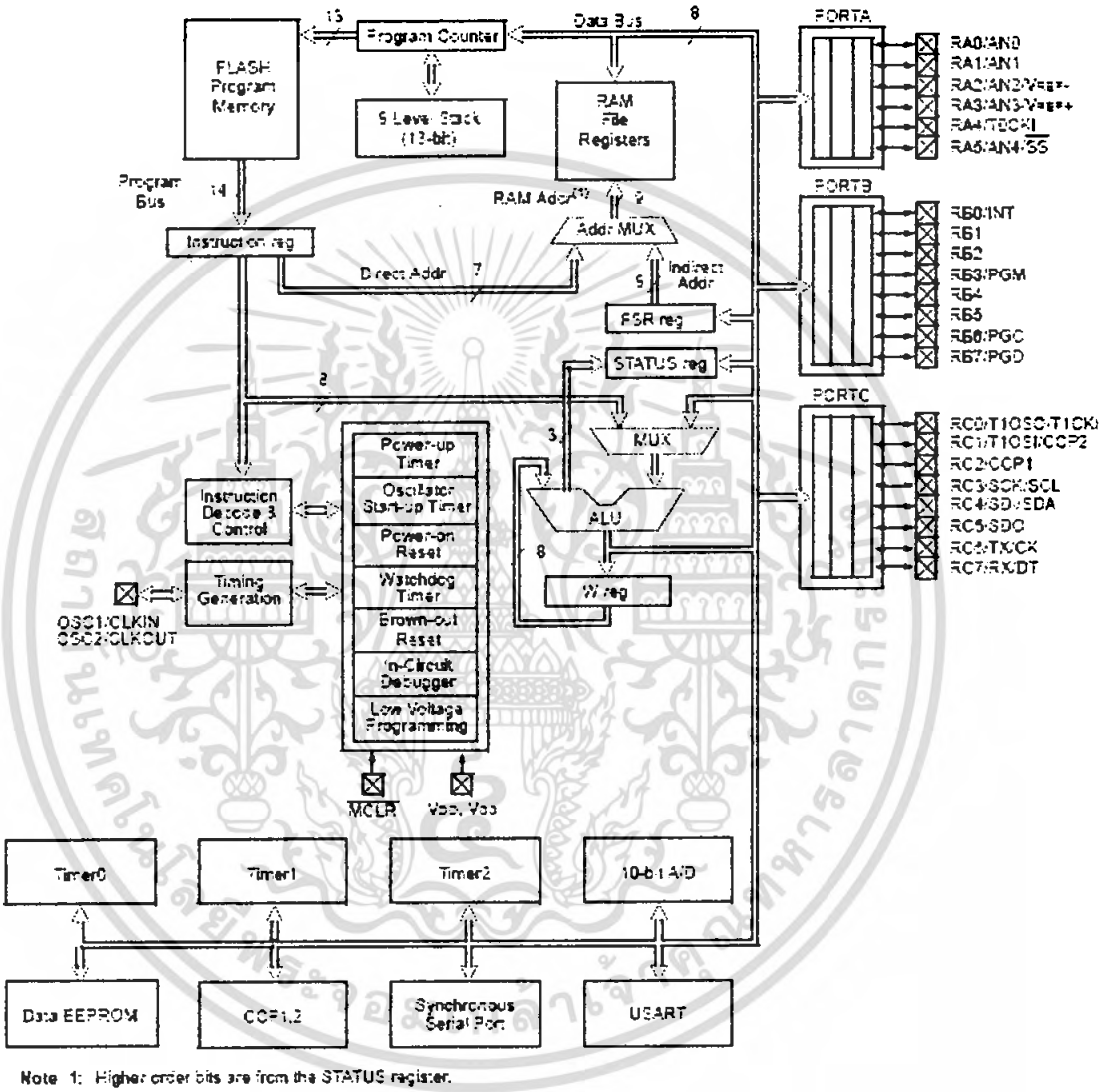
23. มีระบบตรวจระดับไปเลี้ยง (Brown-out reset)

24. มี I/O พอร์ตทั้งหมด 5 พอร์ต (แต่ว่าแต่ละพอร์ตมีจำนวนบิตไม่เท่ากันนะครับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5 โครงสร้างภายในชิพ

บล็อกไดอะแกรมของ PIC 16F877




**รูปที่ 2.15 แสดงโครงสร้างแผนผัง IC PIC 16F877**

จากผังจะมี Register สำคัญๆ คือ W ซึ่งเป็น Register ที่ใช้ในการทำเป็น Input ให้กับ ALU และเป็นตัวเก็บผลลัพธ์จากการทำงานของ ALU, STATUS เป็น Register ที่ใช้เก็บสถานะ การทำงานของคำสั่ง ว่าเมื่อคำสั่งทำงานเสร็จแล้วเกิดอะไรขึ้นมาบ้าง ซึ่งมีประโยชน์ในการเขียนโปรแกรมแบบมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไข, PC หรือ Program Counter เป็น Register อีกตัวหนึ่งที่มีความสำคัญ เนื่องจากใช้สำหรับเป็นตัวชี้ว่า คำสั่งที่จะนำมาประมวลผลนั้นอยู่ ณ ตำแหน่งใดในหน่วยความจำ

### 2.3.6 หน้าที่ของพอร์ตที่เราใช้งาน

**RA0-RA3 และ RA5** จะใช้งานเป็น I/O ปกติ และทำหน้าที่เป็นขาอินพุตของสัญญาณอนาล็อก (AN0-AN4)

**RA4** เป็นขา I/O

**RA6/OSC2/CLKO** ทำหน้าที่ในหลายส่วน คือ เป็นขา OSC2 และ CLKO จะนำมาใช้เป็นขาสัญญาณ I/O ได้ก็ต่อเมื่อเราใช้คริสตอลออสซิลเลเตอร์ แบบที่เป็นโมดูลสำเร็จสามารถต่อเข้ากับขา OSC1/CLKIN ได้เลยโดยที่ไม่ต้องต่อกับขา RA6/OSC2 ทำให้ ขา RA6 ว่างและนำไปใช้เป็น I/O ได้

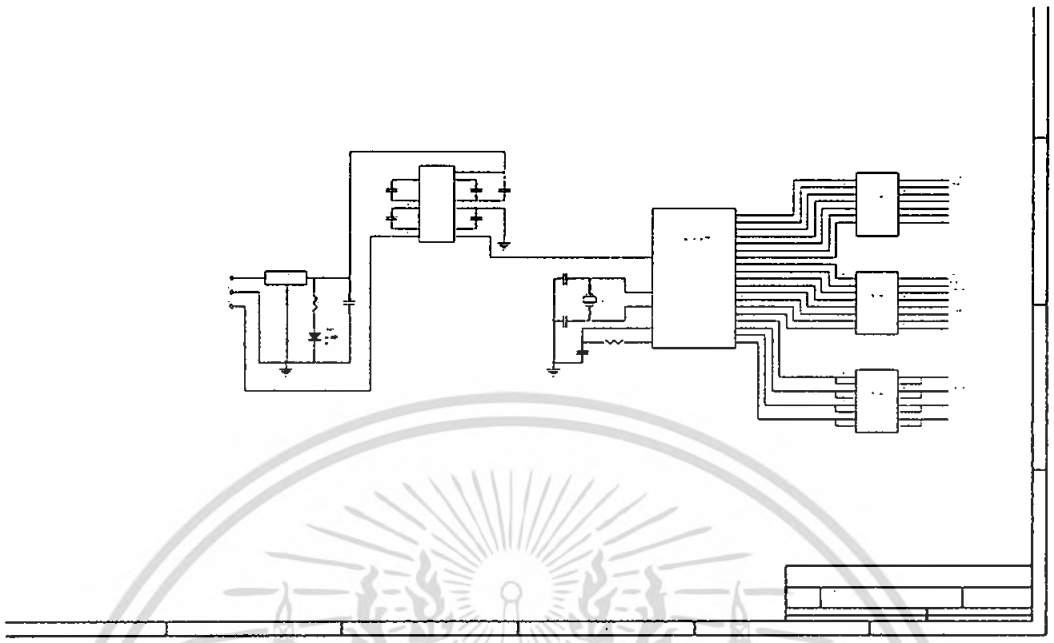
**RB0-RB7** สามารถใช้งานเป็น I/ แต่มีคุณสมบัติ พิเศษคือวงจรมีพูลอัพ (Pull-Up) ภายใน และ เป็นแหล่งกำเนิดสัญญาณอินเตอรัพท์ต่างๆ ดังนี้

- RB0/INT0 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 0
- RB1/INT1 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 1
- RB2/INT2 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 2
- RB3/INT3 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 3 (18F442)
- RB4-RB7 เป็นขาที่สามารถกำเนิดสัญญาณอินเตอร์รัพท์ได้

จะเห็นว่าผมแนะนำเฉพาะพอร์ต A และ B เท่านั้น ทั้งนี้เนื่องจาก ในบทความเรื่อง PIC ผมจะใช้พอร์ต A (RA) ต่อกับพวก Switch และ พอร์ต B ต่อเข้ากับ LED เพื่อใช้ในการเรียนรู้คำสั่ง ของ PIC

เมื่อสัญญาณไร้สายส่งผ่านเข้าที่ตัวรับ ตัวรับนั้นได้ทำการแปลงคำสั่งที่เป็นสัญญาณแอสกี้เป็นตัวคำสั่งสัญญาณตามเดิมที่มาจากการกดสัญญาณคำสั่งจากคอมพิวเตอร์ เมื่อสัญญาณเข้ามาทางวงจรควบคุมและประมวลผลโดยวงจรคอนโทรลเลอร์ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 2.16 วงจรควบคุมโดยไมโครคอนโทรเลอร์**

วงจรที่แสดงเป็นวงจรไมโครคอนโทรเลอร์ที่แปลงสัญญาณประมวลผลของสัญญาณที่ส่งมาแล้วทำงานตามคำสั่งที่เราได้กำหนดไว้ตั้งแต่ตอนแรกจากคอมพิวเตอร์ โดยการทำงานของคอนโทรลเลอร์เป็นการประมวลผลการทำงานตามคำสั่งของรหัสแอสกี้ที่รับเข้ามาโดยถ้าเงื่อนไขของรหัสแอสกี้ที่เข้ามานั้นถูกต้อง ก็จะส่งแรงดันออกไปที่พอร์ตตามที่เรากำหนดไว้เพื่อใช้ในการควบคุมมอเตอร์ตามจุดต่างๆ แต่ถ้ารหัสแอสกี้ไม่ตรงกับเงื่อนไขที่กำหนดไว้มันก็จะกระโดดข้ามไปดูเงื่อนไขตัวต่อไปและจะกระโดดไปเรื่อยๆ จนกว่ารหัสแอสกี้ที่ส่งเข้ามาจะตรงกับเงื่อนไขที่กำหนดไว้ ตัวอย่างเช่นรหัส “W” เมื่อวงจรไมโครคอนโทรลเลอร์ได้รับสัญญาณแล้วแปลงสัญญาณเป็นรหัสแอสกี้ ก็จะนำรหัส “W” ไปเข้า PIC16F877 เพื่อทำการประมวลผลตรวจสอบเงื่อนไขตามที่กำหนดไว้ในแต่ละบรรทัด ถ้าเงื่อนไขของ “W” ที่เข้ามาครั้งแรกเป็นจริงก็จะทำตามเงื่อนไขของบรรทัดนั้นๆ เพื่อทำการส่งแรงดันออกพอร์ตแต่ละพอร์ตเพื่อควบคุมมอเตอร์แต่ละตัวให้ทำงาน แต่ถ้าเงื่อนไขที่ตรวจสอบครั้งแรกเป็น “S” ซึ่งไม่ตรงกับเงื่อนไขครั้งแรกที่วางเอาไว้ ก็จะทำการกระโดดข้ามไปตรวจสอบเงื่อนไขอันถัดไปทันทีและจะตรวจสอบเงื่อนไขอย่างนี้เรื่อยจนกว่าจะมีเงื่อนไขใดเป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 โปรแกรมที่ใช้ในการควบคุมหุ่นยนต์

### 2.4.1 Visual Basic 6

Visual Basic เป็นโปรแกรมที่ใช้สร้างโปรแกรมประยุกต์ สำหรับระบบปฏิบัติการ Windows

**Visual** เป็นส่วนที่หมายถึงเมธอดในการติดต่อแบบ graphical user interface (GUI) ซึ่งการสร้างทำได้โดยการเพิ่มอ๊อบเจค ลงบนฟอร์มที่ทำหน้าที่ติดต่อกับผู้ใช้ผ่านจอภาพ

**Basic** เป็นส่วนที่หมายถึงภาษา BASIC (Beginners ALL Purpose Symbolic Instruction Code) โดยVisual Basic ได้เปลี่ยนแปลงจากภาษา BASIC ดั้งเดิม ด้วยการเพิ่มประโยค คำสั่ง ฟังก์ชัน และคีย์เวิร์ด ที่เกี่ยวข้องโดยตรงกับ GUI

### 2.4.2 แนวคิดของ Visual Basic

โปรแกรมประยุกต์ Visual Basic เป็นการพัฒนาในสภาพแวดล้อมของ windows ซึ่ง แนวคิดพื้นฐานในการทำงานของระบบ Windows ที่สำคัญมี 3 ประการ คือ window, events และ ข่าวสาร (message)โปรแกรมประยุกต์ Visual Basic มีการทำงานแบบ Event-Driven ที่เป็นการประมวลผลตามคำสั่งในแต่ละส่วนเพื่อตอบสนองต่อ event ซึ่ง event เหล่านี้สามารถเปลี่ยนโดยการทำงานของผู้ใช้ ข่าวสารของระบบหรือโปรแกรมประยุกต์อื่น หรือภายในโปรแกรมเดียวกัน ลำดับการทำงานของ event จะจัดลำดับโดยจากการประมวลคำสั่ง

### 2.4.3 โปรแกรม Visual Basic (VB)

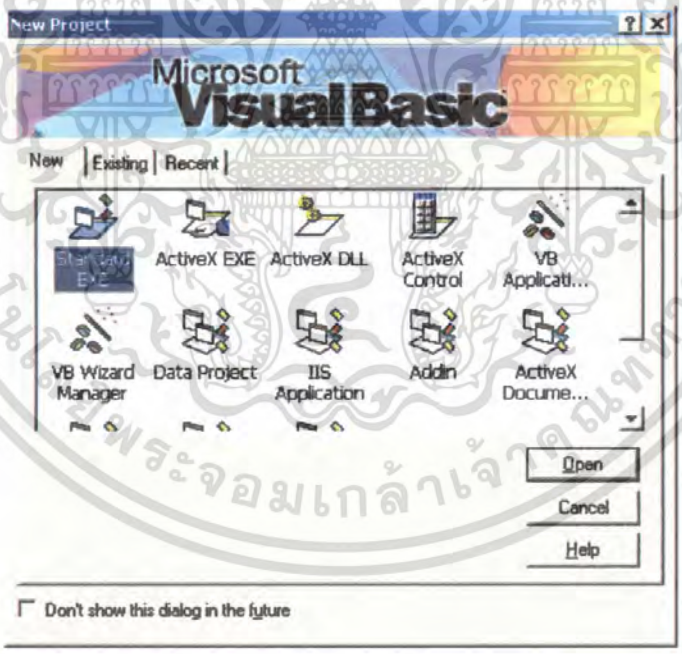
เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่ นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิค หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจ็กต์) นั่นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจ็กต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก้ไขคุณสมบัติของออบเจ็กต์นั้นได้โดยตรง ในทุกๆ ออบเจ็กต์จะมีคุณสมบัติ (propcrtics)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจ็กต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจ็กต์ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเดอร์ (procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจ็กต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

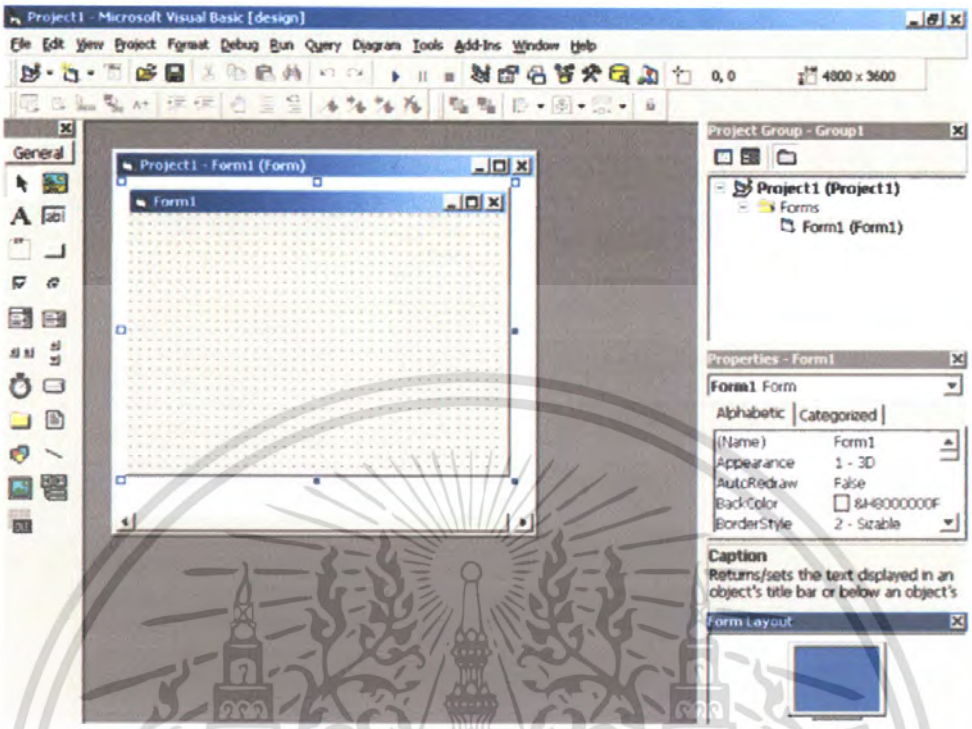
### 2.4.4 เข้าสู่โปรแกรม Visual Basic

เมื่อเข้าสู่โปรแกรม Visual Basic จะแสดงกรอบโต้ตอบสำหรับเลือกชนิดของโปรแกรมประยุกต์ ที่ต้องการ




**รูปที่ 2.17 เมื่อเลือกชนิดของโปรแกรมประยุกต์เป็นแบบ Standard EXE จะเข้าสู่หน้าต่างของ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

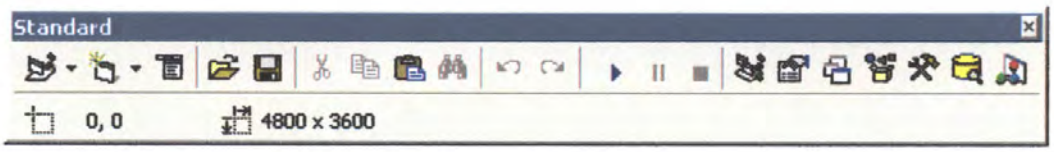



**รูปที่ 2.18 แสดงส่วนของ Visual Basic**

ในแต่ละส่วนของ Visual Basic จะมีหน้าที่แตกต่างกันไป ซึ่งในระหว่างการพัฒนาโปรแกรมประยุกต์ จะต้องใช้ส่วนต่าง ๆ เหล่านี้ ในการพัฒนาโปรแกรมประยุกต์

### 2.4.5 ทูลบาร์ (Toolbars)

เป็นแถบสัญลักษณ์ที่ใช้สำหรับเข้าถึงชุดคำสั่งของ Visual Basic ได้ทันที โดยจะนำคำสั่งที่ถูกใช้งานบ่อย ๆ มาแสดง




**รูปที่ 2.19 แสดงทูลบาร์ (Toolbars)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.5.1 ทูลบาร์สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ ๆ คือ**

**1. Standard Toolbars** เป็นทูลบาร์มาตรฐานประกอบด้วยคำสั่งที่เกี่ยวกับการจัดการ Project

**2. Edit Toolbars** เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการเขียนโค้ดใน code editor

**3. Debug Toolbars** เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับตรวจสอบการทำงานการประมวลผลโปรแกรม

**4. Form Editor Toolbars** เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการปรับขนาด, ย้าย, เปลี่ยนตำแหน่งคอนโทรลต่าง ๆ ที่อยู่บนฟอร์ม

**2.4.6 Toolboxs**

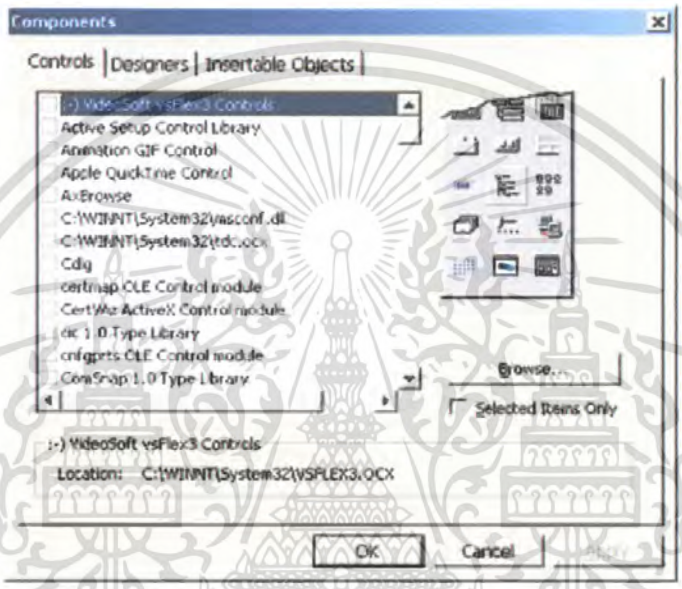
คือแถบสัญลักษณ์ Controls ต่าง ๆ ที่ใช้สำหรับพัฒนาโปรแกรมประยุกต์ แบ่งเป็น 2 กลุ่ม คือ

**รูปที่ 2.20 แสดง Toolboxs**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.7 คอนโทรลภายใน (Intrinsic controls)

เป็นชุดคอนโทรลมาตรฐานของ Visual Basic ทุก ๆ ครั้งที่มีการเรียกใช้ Form เพื่อสร้างโปรแกรมประยุกต์ คอนโทรลชุดนี้จะถูกเรียกขึ้นมาอัตโนมัติ สามารถเลือกใช้งานคอนโทรลกลุ่มนี้ได้ทันที




**รูปที่ 2.21 แสดงคอนโทรล ActiveX (ActiveX controls)**

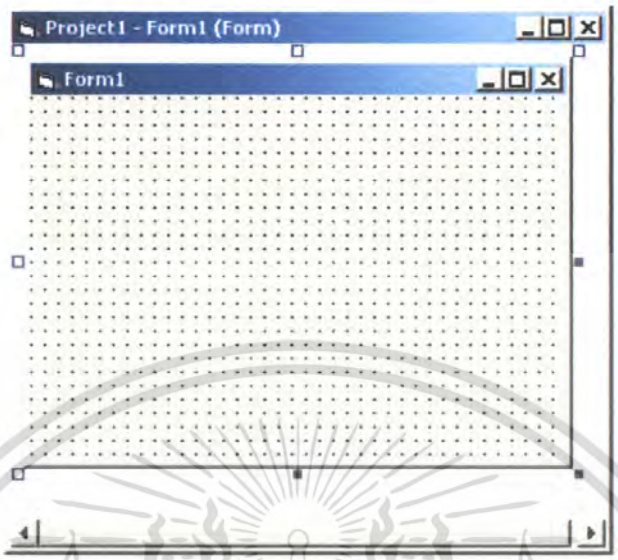
### 2.4.8 คอนโทรล ActiveX (ActiveX controls)

เป็นชุดคอนโทรลเพิ่มเติมที่ไมโครซอฟท์จัดเตรียมไว้เพื่อเพิ่มขีดความสามารถในการพัฒนาโปรแกรมประยุกต์การเพิ่มคอนโทรลกลุ่มนี้เข้ามาในทูลบ๊อกซ์ทำโดยเลือกเมนู Project/Components (หรือคลิ๊กขวาตรงแถบทูลบ๊อกซ์เลือกคำสั่ง

### 2.4.9 Form Designer

เป็นส่วนที่ใช้ออกแบบการแสดงผลส่วนที่ใช้ติดต่อกับผู้ใช้ ฟอร์มเป็นออบเจ็กต์แรกที่ถูกเตรียมไว้ให้ใช้งาน คอลโทรลทุกตัวที่ต้องการใช้งานจะต้องนำไปบรรจุไว้ในฟอร์ม นำคอลโทรลมาประกอบกันขึ้นเป็นโปรแกรมประยุกต์ ทุกครั้งที่เปิด Visual Basic ขึ้นมา หรือ สร้าง Project ใหม่ จะมีฟอร์มว่าง 1 ฟอร์มถูกสร้างเตรียมไว้เสมอ

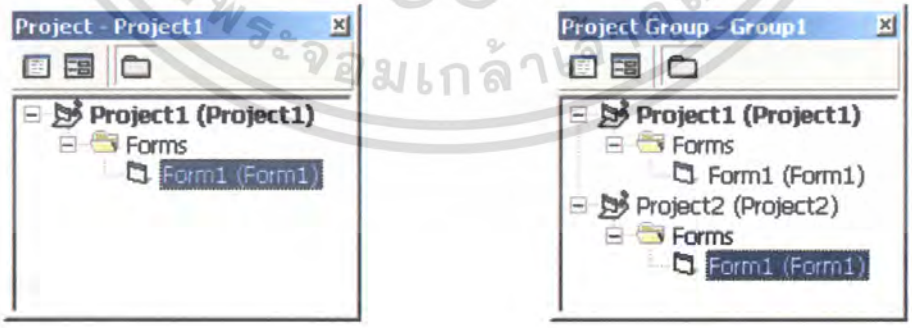
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.22 แสดง Form Designer**

### 2.8.10 Project Explorer

Project Explorer ใช้สำหรับบริหารและจัดการโปรเจ็กซ์ โดยจะแสดงองค์ประกอบของแต่ละโปรเจ็กต์แบบโครงร่างต้นไม้ (tree-view)ตัวโปรเจ็กตจะหมายถึงโปรแกรมประยุกต์ซึ่งจะอยู่ส่วนบนสุด ถัดมา จะแสดงส่วนประกอบต่าง ๆ ของโปรเจ็กต์นั้น ๆ ว่าประกอบด้วยอะไรบ้าง เช่น ฟอร์มโมดูล รายงาน เป็นต้น ถ้ามี 2 โปรเจ็กต์ขึ้นไป ก็จะแสดงแยกออกเป็นส่วนต่างหากอีกโปรเจ็กต์ ถ้าต้องการใช้งานส่วนใด ของโปรเจ็กต์ไหนก็สามารถคลิ๊กเลือกได้ทันที




**รูปที่ 2.23 Project Explorer แบบโปรเจ็กต์เดียว และ แบบหลายโปรเจ็กต์**

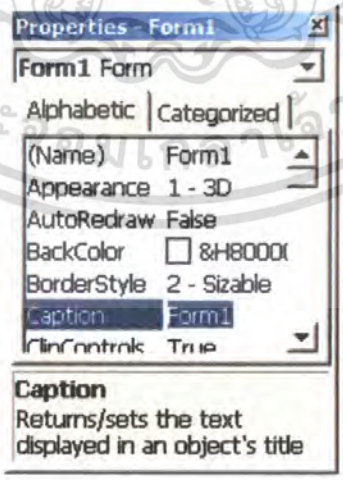
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.11 ส่วนประกอบของโปรเจ็กต์

| | |
|---|---|
| Project(n) | คือโปรแกรมประยุกต์ที่พัฒนาอยู่ มีนามสกุล .vbp |
| Form(n) .frm | เป็นฟอร์มที่มีอยู่ในโปรเจ็กต์นั้น ๆ ใน 1 โปรเจ็กต์อาจมีมากกว่า 1 ฟอร์มก็ได้ มีนามสกุล |
| Modules | เป็นที่เก็บชุดคำสั่งที่คุณเขียนขึ้นมา โดยจะเก็บชุดคำสั่งที่ใช้บ่อย ๆมีนามสกุล .bas |
| Class Modules | เป็นโมดูลชนิดพิเศษที่มีลักษณะเป็นอ๊อบเจ็กต์ ที่สามารถสร้างขึ้นมาได้ จะมีนามสกุล .cls |
| User controls | เป็นส่วนที่เก็บคอนโทรล ActiveX ที่คุณสร้างขึ้นมา มีนามสกุล .ctl |
| Designers | เป็นส่วนของรายงานที่ถูกสร้างขึ้นมีนามสกุลเป็น .dsr |

### 2.4.12 Properties Window

หน้าต่างคุณสมบัติเป็นส่วนที่ใช้กำหนดคุณสมบัติของออบเจ็กต์ที่ถูกเลือก (adtive) หรือได้รับความสนใจ (focus) อยู่ขณะนั้น ซึ่งสามารถที่จะปรับเปลี่ยนค่าต่าง ๆ ของคอลโทรลเพื่อให้เกิดความเหมาะสมและตรงกับความต้องการใช้งานได้ทันที




**รูปที่ 2.24 แสดง Properties Window**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

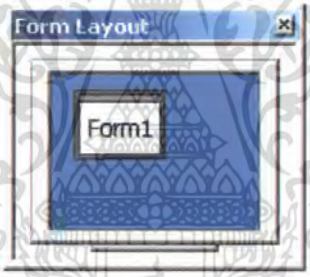
ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็ป 2 แท็ป คือ

1. แท็ป Alphabetic เป็นแท็ปที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ

2. แท็ป Categorized เป็นแท็ปที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน

**2.4.13 หน้าต่าง Form Layout**

เป็นส่วนที่แสดงให้เห็นตำแหน่งของฟอร์ม และสามารถกำหนดตำแหน่งของฟอร์ม ที่ปรากฏบนจอภาพในขณะประมวลผลได้ โดยการเคลื่อนย้ายฟอร์มจำลอง ที่อยู่ในจอภาพจำลองด้วยการ drag เมาส์ ไปยังตำแหน่งที่คุณต้องการ โดยจะมีผลในขณะประมวลผลเท่านั้น




**รูปที่ 2.25 แสดงหน้าต่าง Form Layout**

**2.4.14 Immediate Window**

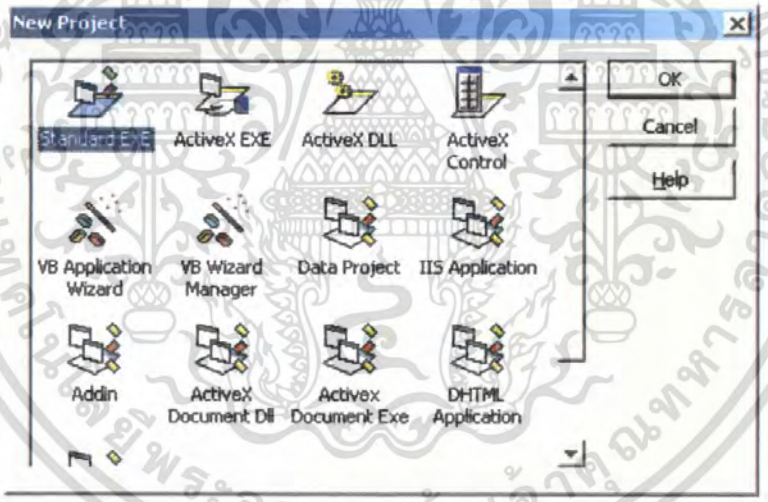
เป็นหน้าต่างที่ให้ประโยชน์ ในกรณีที่คุณต้องการทราบผล การประมวลผลโดยทันที เช่น การทดสอบโปรแกรมย่อยต่าง ๆ เป็นต้น เมื่อคุณสั่งประมวลผลโปรเจ็กต์ หน้าต่างนี้จะปรากฏขึ้นโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.26 แสดง Immediate Window**

### 2.4.15 หน้าต่าง New Project

หน้าต่าง New Project จะปรากฏขึ้นมาเมื่อเลือกเมนู File/New Project กรอบโต้ตอบนี้ จะแสดงชนิดของโปรแกรมประยุกต์ ที่คุณต้องการพัฒนา ซึ่งจะคล้ายกับตอนที่เปิดโปรแกรม Visual Basic ขึ้นมาครั้งแรก

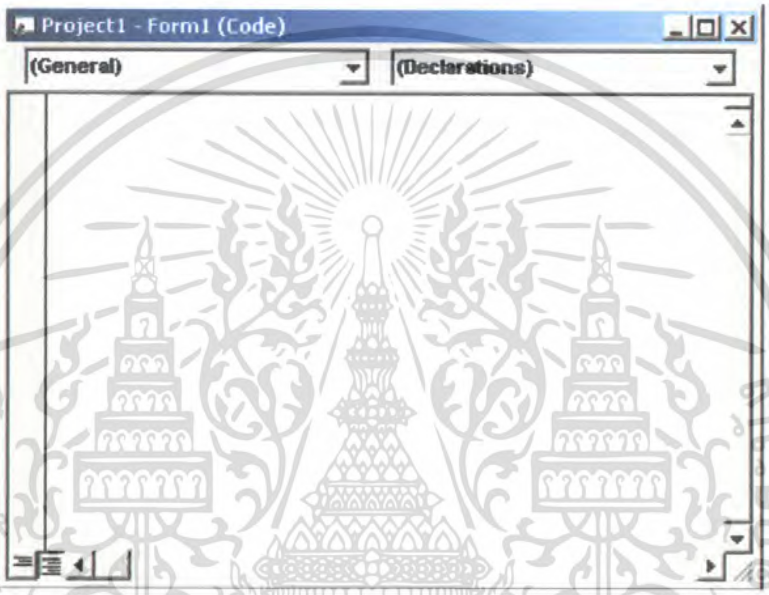



**รูปที่ 2.27 แสดง New Project**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.16 หน้าต่าง Code Editor

เป็นส่วนที่ใช้ในการเขียนชุดคำสั่งสำหรับการประมวลผล และควบคุมการทำงานของคอนโทรล




รูปที่ 2.28 แสดงหน้าต่าง Code Editor

## 2.5 ภาครับส่งข้อมูล

แบ่งเป็น 2 ส่วน ดังนี้

1. เครื่องรับ – ส่ง ที่ใช้ในการควบคุมหุ่นยนต์ (ETT)
2. เครื่องรับ – ส่ง กล้องวงจรปิด (CCTV)

### 2.5.1 เครื่องรับ – ส่ง ที่ใช้ในการควบคุมหุ่นยนต์ (ETT)

ET-RF24G V2.0 เป็นชุด Signal Converter สำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ชุด ET-RF24G V2.0 ก็จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่ (GFSK) จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วยซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณ ET-RF24G V2.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรงโดยจะมีข้อดีกว่า คือ สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญ คือไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้น จะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมากแต่อย่างไรก็ตามการรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางในการสื่อสารนั้น ก็มีข้อจำกัดบางประการเหมือนกันโดยเฉพาะอย่างยิ่ง เรื่องความน่าเชื่อถือของข้อมูลที่รับส่งกัน ซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากในการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลางในการรับส่งข้อมูลแต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาสที่ข้อมูลจะเกิดการรบกวนจากสัญญาณอื่นๆที่มีย่านความถี่ใกล้เคียงกันแล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน ซึ่งระบบการจัดการข้อมูลของเครื่อง ET-RF24G V2.0 นั้น มีระบบการเข้ารหัสและถอดรหัสข้อมูลที่มีความน่าเชื่อถืออยู่ในเกณฑ์ที่จัดว่าดี โดยข้อมูลแต่ละ Byte ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกัน

เนื่องจากกลไกในการรับส่งข้อมูลของเครื่อง ET-RF24G V2.0 นั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่ง ว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามีการสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆใหม่อีกครั้งหนึ่ง ก็จะสามารถแก้ไขปัญหาดังกล่าวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.29 แสดงรูปร่าง ETT**

**2.5.2 แหล่งจ่ายไฟเลี้ยง (Power Supply)**

สำหรับการต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V2.0 นั้น จะสามารถเลือกต่อแหล่งจ่ายไฟให้กับตัวเครื่องได้ 2 ทางด้วยกัน โดยเครื่อง ET-RF24G V2.0 นั้น ต้องการไฟเลี้ยงวงจร ซึ่งเป็นแหล่งจ่ายกระแสตรง ขนาดประมาณ +5VDC ถึง +9VDC โดยจุดเชื่อมต่อแหล่งจ่ายไฟของเครื่อง ET-RF24G V2.0 นี้ สามารถเชื่อมต่อได้ 2 จุดด้วยกัน โดยผู้ใช้สามารถเลือกต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V2.0 จุดใดจุดหนึ่งก็ได้ ในกรณีที่นำเครื่อง ET-RF24G V2.0 ไปเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆของ อีทีที นั้น

สามารถใช้แหล่งจ่ายไฟจากบอร์ดไมโครคอนโทรลเลอร์ เพื่อจ่ายให้กับตัวเครื่อง ET-RF24G V2.0 ได้ทันที โดยไม่ต้องใช้แหล่งจ่ายไฟจากภายนอก เนื่องจากขั้วต่อสัญญาณ RS232 ของบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆ ของบริษัท อีทีที นั้น ได้จัดเตรียมแหล่งจ่ายไฟตรง ขนาด +5V เตรียมไว้ให้ด้วยแล้ว โดยผู้ใช้เพียงแต่นำสายสัญญาณRS232 ซึ่งทำการต่อสายสัญญาณครบทั้ง 4 เส้น ดังรูปมาเชื่อมต่อก็สามารถใช้งานได้แล้ว

**รูปที่ 2.30 แสดงการต่อสายสัญญาณ RS232**
**เพื่อใช้แหล่งจ่ายไฟจากบอร์ดไมโครฯ ของ ETT**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.3 คู่มือการใช้งาน RS232 to RF-Wireless (RF2.4GHz)รุ่น ET-RF24G V1.0 ETT.CO.,LTD**

แต่สำหรับกรณีที่นำเครื่อง ET-RF24G V2.0 ไปต่อใช้งานกับอุปกรณ์อื่นๆที่ไม่ได้มีการจัดเตรียมจุดต่อไฟเลี้ยงไว้ให้ด้วย ผู้ใช้จำเป็นต้องจัดหา Adapter จ่ายไฟจากภายนอกมาต่อให้กับเครื่อง ET-RF24G V2.0 ต่างหากด้วย โดยให้เลือกแหล่งจ่ายไฟที่มีขนาดแรงดันไฟตรงประมาณ +5VDC และสามารถจ่ายกระแสได้ประมาณ300mAเป็นอย่างน้อย ซึ่งในกรณีนี้ขอแนะนำให้เลือกใช้ Power Supply รุ่น "ACH-4E" ซึ่งเป็นแหล่งจ่ายไฟแบบSwitching Power ใช้กับไฟบ้าน 220VAC และให้เอาต์พุตเป็นไฟกระแสตรง ขนาดประมาณ 5VDC / 750mA เพราะPower Supply รุ่นนี้ สามารถใช้งานร่วมกับเครื่อง ET-RF24G V2.0 ได้อย่างต่อเนื่องเป็นเวลานานๆ

โดยไม่เกิดความร้อนสะสมที่วงจร Regulate ของบอร์ด ET-RF24G V2.0 มากนัก ซึ่งถ้าผู้ใช้เลือกใช้แหล่งจ่ายไฟรุ่นอื่นๆ ทีมีขนาดแรงดันสูงกว่า +5V มากๆ ซึ่งถึงแม้ว่าจะสามารถใช้งานร่วมกันกับเครื่อง ET-RF24G V2.0 ได้ แต่ถ้ามีการใช้งานอย่างต่อเนื่องเป็นเวลานานๆแล้ว อาจทำให้เกิดความร้อนสะสมที่ตัวไอซี Regulate มากเกินไป จนอาจทำให้ภาคPower ของเครื่อง ET-RF24G V2.0 หยุดจ่ายไฟทำให้เครื่องหยุดทำงานได้

**รูปที่ 2.31 แสดงการต่อแหล่งจ่ายไฟรุ่น "ACH-4E" จากภายนอกให้กับเครื่อง ET-RF24G V1.0**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.4 โหมดการทำงาน**

สำหรับโหมดการทำงานของ ET-RF24G V2.0 นั้นจะแบ่งออกเป็น 2 โหมด ด้วยกัน โดยการกำหนดโหมดการทำงานของ ET-RF24G V2.0 นั้นจะกระทำผ่าน Switch เลือกโหมด ซึ่งอยู่ด้านใต้กล่อง โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับ ET-RF24G V2.0 ด้วยเสมอ เนื่องจากการทำงานของเครื่อง ET-RF24G V2.0 นั้นจะทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เลือกโหมด เฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เลือกโหมด หลังจากทำการจ่ายไฟให้กับ ET-RF24G V2.0 ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานของเครื่อง ET-RF24G V2.0 นั้นจะมี LED แสดงสถานะการทำงานของเครื่องจำนวน 2 หลอด คือ LED POWER ซึ่งเป็น LED สีแดง โดยที่ LED POWER นี้จะติดสว่างให้เห็นตลอดเวลาที่มีการจ่ายไฟเลี้ยงให้เครื่องทำงานอยู่ ส่วน LED อีกดวงหนึ่งนั้นจะเป็น LED สีเขียว ใช้แสดงสถานะการทำงานของเครื่อง ซึ่งเรียกว่าLED STATUS โดย LED STATUS นี้จะเกิดการกระพริบตามจังหวะของการรับส่งข้อมูลกันในแต่ละครั้งโดยในสภาวะปรกตินั้น ถ้าเครื่องทำงานอยู่ใน RUN MODE หลอด LED STATUS จะดับอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล แต่ถ้าตัวเครื่องทำงานอยู่ใน SETUP MODE หลอด LED STATUS จะติดอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูลโดยโหมดการทำงานของ ET-RF24G V2.0 จะมีอยู่ด้วยกัน 2 โหมด คือ

**2.5.5 การใช้งานเครื่อง ET-RF24G V2.0 ใน Run Mode**

การใช้งานใน Run Mode ซึ่งเป็นโหมดของการใช้งานตามปรกติของเครื่อง โดยเมื่อเครื่อง ET-RF24GV2.0 เข้าทำงานในโหมดนี้แล้ว จะสังเกตเห็นหลอดไฟแสดงสถานะของการทำงานหรือ LED STATUS ดับอยู่ แต่เมื่อมีการ รับ หรือ ส่ง ข้อมูล เกิดขึ้น สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูลนั้นๆ แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะดับอยู่ตลอดเวลาสำหรับการทำงานใน Run Mode นั้น จะแบ่งลักษณะการทำงานออกเป็น 3 แบบด้วยกัน โดยลักษณะการทำงานนี้ จะถูกกำหนดไว้แล้วใน Configuration ของเครื่องใน Setup Mode ดังนั้นก่อนการใช้งานเครื่อง ในครั้งแรกจะต้องทำการกำหนดค่า Configuration ต่างๆให้เรียบร้อยเสียก่อน โดยเมื่อเครื่อง ET-RF24G V2.0 เริ่มต้นเข้าทำงานใน Run Mode แล้วมันจะทำการอ่านค่า Configuration ที่เก็บไว้ออกมา เพื่อใช้เป็นเงื่อนไขในการทำงานตามค่าที่ได้กำหนดไว้ โดยลักษณะการทำงานใน Run Mode แบ่งออกเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.5.1 การทำงานแบบ RF Receive Only**

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้ จะเป็นการรอรับข้อมูลความถี่แบบ GFSK จากด้าน RF แล้วเปลี่ยนเป็นข้อมูลอนุกรมส่งออกไปทางขา TX (Transmit) ของ RS232 โดยการทำงานจะวนรอบอยู่ เช่นนี้ไปตลอด ซึ่งในการใช้งานเครื่อง ET-RF24G V2.0 ในโหมดนี้จะต้องนำสัญญาณ TX(Transmit) ไปต่อกับขาสัญญาณ RX (Receive) ของอุปกรณ์ด้านตรงข้าม (RS232 ของคอมพิวเตอร์ PC) โดยในโหมดนี้ การทำงานของขาสัญญาณ RX ด้าน RS232 ของเครื่อง ET-RF24G V2.0 จะถูกเปลี่ยนหน้าที่เป็นสัญญาณ CTS (Clear To Send)ใช้ตรวจสอบความพร้อมในการส่งข้อมูลไปให้อุปกรณ์ด้านตรงข้ามแทน ซึ่งในการใช้งานจะต้องนำสัญญาณนี้ไปต่อเข้ากับสัญญาณ RTS (Ready To Send) ของอุปกรณ์ด้านตรงข้าม โดยเครื่อง ET-RF24G V2.0 จะทำการตรวจสอบสถานะของสัญญาณ RX ซึ่งในโหมดนี้เปรียบเสมือน CTS ว่ามีค่าเป็น "0" หรือไม่ โดยถ้าพบว่าเป็น "0"จึงจะส่งข้อมูลออกไปให้ทางขา TX แต่ถ้าพบว่าสถานะของขาสัญญาณนี้มีค่าเป็น "1" แสดงว่าอุปกรณ์ด้านตรงข้ามยังไม่พร้อมรับข้อมูลก็จะรอจนกว่าจะพบว่าสถานะของสัญญาณดังกล่าวมีค่าเป็น "0" จึงจะส่งข้อมูลออกไปให้ โดยเครื่อง ET-RF24G V2.0 จะสามารถจัดเก็บข้อมูลไว้ใน Buffer เพื่อรอการส่งได้สูงสุด 64 Byte เท่านั้น ซึ่งถ้าในระหว่างที่รอความพร้อมอยู่นั้น มีข้อมูลด้าน RF ส่งเข้ามาเกินกว่า 64 Byte จะทำให้ข้อมูลที่เกินมานั้นสูญหายไป

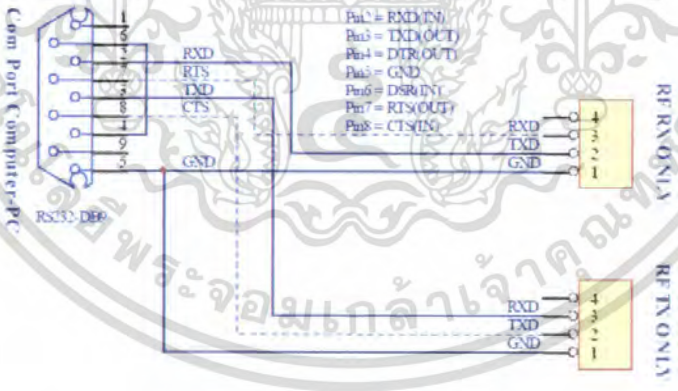
**2.5.5.2 การทำงานแบบ RF Transmit Only**

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้จะมีลักษณะตรงกันข้ามกับ RF Receive Onlyกล่าวคือ เครื่อง ET-RF24G V2.0 จะทำหน้าที่รอรับข้อมูลจากขา RX (Receive) ด้าน RS232 แล้วเปลี่ยนเป็นข้อมูลแบบ GFSK ส่งออกไปทางด้าน RF โดยการใช้งานเครื่องในโหมดนี้ จะต้องนำสัญญาณ TX (Transmit) ซึ่งเป็นขาส่งข้อมูลจาก RS232 ของอุปกรณ์ด้านตรงข้ามมาต่อเข้ากับขา RX(Receive) ของเครื่อง ET-RF24G V2.0 ส่วนขาสัญญาณ TX จะถูกเปลี่ยนหน้าที่เป็น RTS (Ready To Send) เพื่อใช้แสดงสถานะความพร้อมในการรับข้อมูลจากด้าน RS232 ซึ่งในการใช้งานจะต้องนำสัญญาณ TX ซึ่งในขณะนี้เปรียบเสมือนกับ RTS นำไปต่อเข้ากับสัญญาณCTS (Clear To Send) ของอุปกรณ์ด้านตรงข้าม เพื่อใช้ในการตรวจสอบความพร้อมในการรับข้อมูล โดยอุปกรณ์ด้านตรงข้ามจะต้องทำการตรวจสอบสถานะของสัญญาณ RTS นี้ เพื่อตรวจสอบความพร้อมในการรับข้อมูลของเครื่อง ET-RF24G V2.0 ด้วย โดยถ้าเครื่อง ET-RF24G V2.0 พร้อมรับข้อมูลจาก RS232 มันจะส่งสัญญาณ RTS ให้มีค่าเป็น "0" รอไว้ และเมื่อใดก็ตามที่การรับข้อมูลทางด้านของ RS232 มีจำนวนข้อมูลที่ยังไม่สามารถเปลี่ยนเป็นGFSK เพื่อส่งออกไปทางด้าน RF ได้ทันจนเกือบจะเต็ม Buffer แล้ว เครื่อง ET-RF24G V2.0 จะทำการส่งสัญญาณRTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้มีค่าเป็น “1” ออกไปบอกให้อุปกรณ์ด้านตรงข้ามทราบเพื่อจะได้หยุดการส่งข้อมูลออกมา โดยอุปกรณ์ด้านตรงข้ามจะต้องหยุดการส่งข้อมูลและรอจนกว่าสถานะของสัญญาณ RTS จะกลับเป็น “0” จึงจะเริ่มต้นส่งข้อมูลออกมาใหม่

ซึ่งหลังจากที่เครื่อง ET-RF24G V2.0 ส่งสัญญาณ RTS ด้วยค่า “1” ออกไปแล้ว จะยังคงสามารถรับข้อมูลได้เพิ่มเติมอีกไม่เกิน 16 Byte เท่านั้น ซึ่งถ้าอุปกรณ์ด้านตรงข้ามยังส่งข้อมูลต่อเนื่องมาอีกจนเกินขนาดของBuffer ที่เครื่อง ET-RF24G V2.0 จะรับไว้ได้จะทำให้ข้อมูลที่เกินมานั้นเกิดการสูญหายได้โดยเราสามารถนำเครื่อง ET-RF24G V2.0 จำนวน 4 ชุด มาต่อใช้งานร่วมกันเพื่อใช้งานในการรับส่งข้อมูลกันแบบ Full Duplex โดยแบ่งการใช้งานออกเป็น 2 ด้าน คือ ต้นทางและ ปลายทาง ด้านละ 2 ชุด โดยแต่ละด้านให้กำหนดหน้าที่การทำงานเป็น RF Receive Only 1 ชุด และ RF Transmit Only อีก 1 ชุด




**รูปที่ 2.32 แสดงสายสัญญาณ RS232 เพื่อใช้กับ ET-RF24G ในโหมด RF Receive Onlyและ RF Transmit Only**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.5.2 การทำงานแบบ RF Auto Direction**

เป็นการทำงานชนิด 2 ทิศทาง แบบ Half Duplex หรือ ผลัดกันรับผลัดกันส่ง ซึ่งสามารถใช้รับส่งข้อมูลระหว่างต้นทาง และ ปลายทาง ได้ โดยใช้เครื่อง ET-RF24G V2.0 ด้านละ 1 ชุด เท่านั้น เพียงแต่การรับส่งข้อมูลแบบนี้จะไม่สามารถส่งข้อมูลสวนทางกันได้เหมือนกับแบบ Full Duplex แต่จะต้องใช้วิธีการผลัดกันรับข้อมูลและส่งข้อมูลแทน โดยเมื่อฝ่ายรับทำการรับข้อมูลได้จนครบแล้วจึงจะสลับหน้าที่เป็นฝ่ายส่งเพื่อส่งข้อมูลย้อนกลับไปโดยในโหมดนี้ เครื่อง ET-RF24G V2.0 จะทำหน้าที่เป็นทั้ง ฝ่ายรับ และ ฝ่ายส่ง ข้อมูล แบบอัตโนมัติ โดยในสภาวะปรกติจะอยู่ในสภาวะของการรอรับข้อมูล ทั้งด้าน RF และ RS232 ซึ่งถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้านของ RF ก็จะนำข้อมูลนั้นส่งออกไปทางด้านขา TX ของ RS232 ทันที และในทำนองเดียวกัน ถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้าน RX ของ RS232 มันก็จะทำการรับข้อมูลนั้นจาก RS232 พร้อมกับเปลี่ยนทิศทางของอุปกรณ์ RF จากการรอรับข้อมูลให้ทำหน้าที่เป็นตัวส่งข้อมูลแทน เพื่อทำการส่งข้อมูลที่รับได้จาก RS232 ออกไปทาง RF ในทันที ซึ่งหลังจากที่เครื่อง ET-RF24G V2.0 ทำการสลับโหมดการทำงานของอุปกรณ์ด้าน RF

จากการรอรับเป็นการส่งและทำการเริ่มต้นส่งข้อมูลออกไปทางด้าน RF เรียบร้อยแล้ว มันจะวนกลับไปตรวจสอบการรับข้อมูลจากด้าน RS232 อีกว่ายังมีข้อมูลส่งเข้ามาอีกหรือไม่ ถ้าพบว่ายังมีข้อมูลส่งเข้ามีอีกก็จะทำการแปลงข้อมูลนั้นเพื่อส่งออกไปยังด้าน RFต่อไปอีกจนกว่าการส่งข้อมูลด้าน RS232 จะสิ้นสุดลง ซึ่งข้อมูลด้าน RS232 ที่ส่งเข้ามานั้น ควรส่งอย่างต่อเนื่องโดยเมื่อเครื่อง ET-RF24G V2.0 ทำการส่งข้อมูลแต่ละ Byte ออกไปทางด้าน RF เรียบร้อยแล้วมันจะวนรอบรอรับข้อมูล Byte ถัดไปจาก RS232 ภายในเวลา 2.5 mS ถ้าไม่พบข้อมูลส่งเข้ามาอีกภายในระยะเวลาดังกล่าวมันจึงจะทำการเปลี่ยนหน้าที่ของอุปกรณ์ด้าน RF ให้กลับมาทำหน้าที่เป็นการรอรับข้อมูลตามเดิม โดยในขณะที่อุปกรณ์ด้าน RFถูกกำหนดให้เป็นฝ่ายส่งข้อมูลอยู่นั้น จะไม่สามารถทำการรับข้อมูลจาก RF ได้ ซึ่งถ้ามีการส่งข้อมูลเข้ามาในขณะนั้นก็จะไม่สามารถรับได้ โดยค่าเวลาที่จะใช้ในการสลับโหมดการทำงานของ RF จากฝ่ายส่งข้อมูลให้เป็นฝ่ายรับข้อมูลนั้นจะมีค่าเป็น 2.5mS ดังนั้นเมื่อฝ่ายรับสามารถรับข้อมูลได้ครบหมดแล้วก่อนที่จะทำการส่งข้อมูลเพื่อตอบกลับไปยังฝ่ายตรงข้ามนั้น ควรทำการหน่วงเวลาไว้ไม่น้อยกว่า 3mS นับจากรับข้อมูล Byte สุดท้ายได้เรียบร้อยแล้วจึงเริ่มต้นส่งข้อมูล Byte แรกย้อนกลับไป ซึ่งถ้าฝ่ายรับทำการส่งข้อมูลตอบกลับไปยังฝ่ายตรงข้ามเร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทันสำหรับการใช้งานเครื่อง ET-RF24G V2.0 ในโหมด RF Auto Direction นี้ การ รับ และ ส่ง ข้อมูล ด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

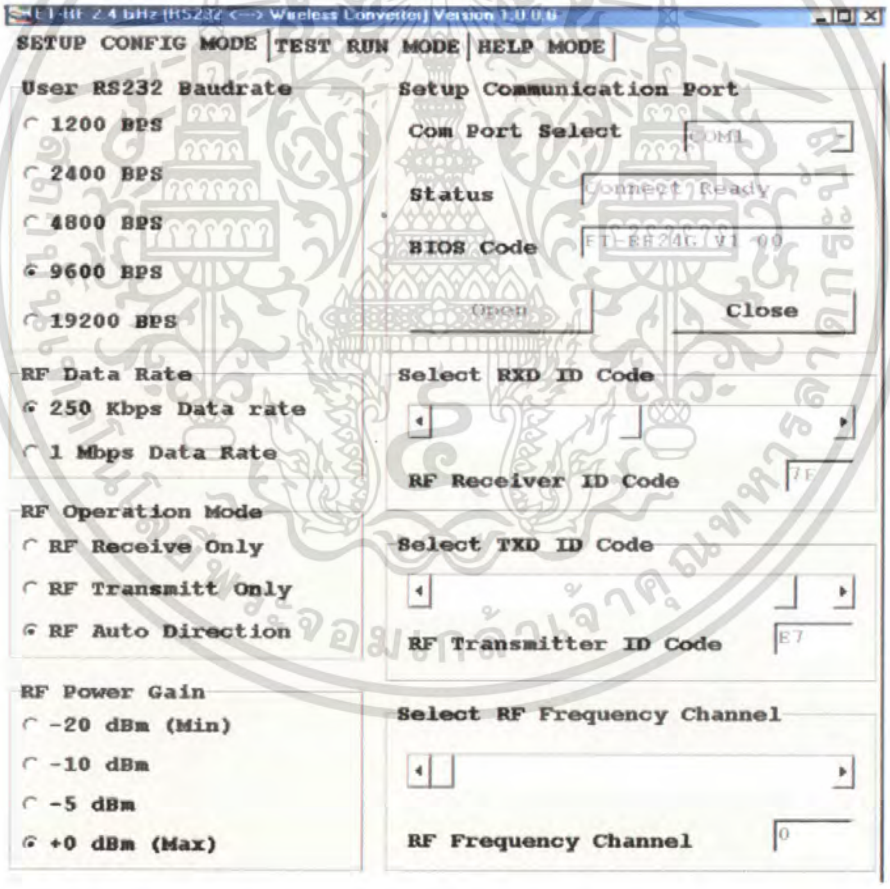
RS232 จะไม่มีการตรวจสอบความพร้อมของฝ่ายรับและส่ง ด้วยสัญญาณทางไฟฟ้า (CTS/RTS) เหมือนกับการใช้งานใน 2 โหมดที่ผ่านมาแล้ว โดยเมื่อมันสามารถรับข้อมูลจาก RF ได้ ก็จะทำการส่งข้อมูลนั้นออกไปทางขา TX(Transmit) ของ RS232 ในทันที โดยไม่สนใจว่า อุปกรณ์ที่ต่อไว้ด้าน RS232 จะพร้อมรับข้อมูลหรือไม่ ซึ่งถ้าด้าน RS232 ไม่พร้อมรับข้อมูลก็จะทำให้ข้อมูล Byte นั้นสูญหายไปทันที ซึ่งในการใช้งานนั้น ผู้ใช้ควรกำหนดค่า ความเร็วในการรับส่งข้อมูลด้าน RS232 ที่จะใช้กับเครื่อง ET-RF24G V2.0 ทุกๆตัวด้วยค่าความเร็วที่เท่ากันด้วยเพื่อให้การรับและส่งข้อมูลเกิดความสัมพันธ์กันอย่างเหมาะสมสำหรับความสามารถในการรอรับข้อมูลจาก RS232 ของเครื่อง ET-RF24G V2.0 ในโหมดนี้ จะสามารถรับข้อมูลได้อย่างต่อเนื่องสูงสุด ไม่เกิน 64 Byte ดังนั้นในกรณีที่มีการส่งข้อมูลจากด้าน RS232 ด้วยข้อมูลจำนวนมากกว่า 64 Byte ต่อเนื่องกันนั้น ควรทำการแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดชุดละไม่เกิน 64 Byte ซึ่งหลังจากทำการส่งข้อมูลอย่างต่อเนื่องไปได้ 1 ชุด (64 Byte) แล้วควรทำการหน่วงเวลาไว้ชั่วขณะหนึ่งอย่างน้อย 1mS แล้วจึงเริ่มส่งข้อมูลชุดถัดไป สลับกับการหน่วงเวลา อย่างนี้เรื่อยๆ เพื่อให้เครื่อง ET-RF24G V2.0 สามารถนำ ข้อมูลที่รับได้จากด้าน RS232 ส่งออกไปทางด้าน RF ได้ทัน ซึ่งถ้าทำการส่งข้อมูลอย่างต่อเนื่องโดยไม่มีการหน่วงเวลาเลยอาจทำให้ข้อมูลบาง Byte เกิดการสูญหายไปได้

### 2.5.6 การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode

การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode ซึ่งเป็นโหมดสำหรับใช้กำหนดค่า Configurationต่างๆ สำหรับควบคุมการทำงานของเครื่อง ET-RF24G V2.0 ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode โดยในการ Setup ค่า Configuration ต่างๆนั้นจะกระทำร่วมกับโปรแกรม "ET_RF24G_V1.EXE" ของ อีทีที ซึ่งเมื่อ เครื่อง ET-RF24G V2.0 เข้าทำงานในโหมด Setup แล้ว จะสังเกตเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LEDSTATUS ติดสว่างค้างอยู่ตลอดเวลา แต่เมื่อมีการสั่งอ่านหรือเขียนข้อมูลกับบอร์ด สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูล แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะติดค้างอยู่ตลอดเวลา ซึ่งการกำหนดค่า Configuration ให้กับ ET-RF24G V2.0 นั้น จะต้องกระทำในขณะที่ตัวเครื่องทำงานอยู่ใน Setup Mode เท่านั้น (เลือก Switch กำหนดโหมดไว้ทางด้าน Setup แล้วจ่ายไฟให้เครื่องเริ่มต้นทำงาน) โดยค่าของ Configuration ต่างๆนั้นจะถูกใช้สำหรับเป็นเงื่อนไขในการทำงานของ ET-RF24G V2.0ในขณะที่อยู่ใน Run Mode ดังนั้น ก่อนการเริ่มต้นใช้งานเครื่องในครั้งแรกนั้น จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดค่าของ Configuration ต่างๆให้ถูกต้องและตรงกับความต้องการที่จะใช้งานเสียก่อน โดยเมื่อทำการกำหนดค่าตัวเลือกต่างๆของ Configuration เรียบร้อยแล้ว ก็สามารถเปลี่ยนโหมดการทำงานของตัวเครื่องกลับเป็นRun Mode พร้อมกับการปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟที่จ่ายให้กับตัวเครื่อง (Power-OFF) ชั่วขณะหนึ่ง จากนั้นจึงเริ่มต้นจ่ายไฟให้กับตัวเครื่องใหม่ (Power-ON) ก็สามารถใช้งาน ET-RF24G V2.0 ตามค่าของ Configuration ที่กำหนดไว้แล้วได้ทันที โดยค่าตัวเลือกต่างๆของ Configuration ที่ได้กำหนดไว้แล้วจะถูกเก็บไว้ภายในตัวเครื่องอย่างถาวร ถึงแม้ว่าจะไม่ได้ทำการจ่ายไฟให้กับตัวเครื่องแล้วก็ตาม ดังนั้นเมื่อทำการกำหนดค่า Configuration ต่างๆเรียบร้อยแล้ว ถ้าไม่มีการเปลี่ยนแปลงเงื่อนไขการทำงานของตัวเครื่องต่างไปจากเงื่อนไขเดิมที่ได้กำหนดไว้แล้ว ก็ไม่จำเป็นต้องทำการกำหนดค่า Configuration ใหม่อีกแต่อย่างใด โดยทุกๆ ครั้งที่เริ่มต้นจ่ายไฟเข้าเครื่องในครั้งแรกนั้น การทำงานของET-RF24G V2.0 จะเป็นไปตามเงื่อนไขที่กำหนดไว้ใน Configuration เสมอทุก ๆครั้ง โดยคุณสมบัติของ Configuration ต่างๆนั้นมีดังนี้




**รูปที่ 2.33 แสดงโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V2.0**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **User RS232 Baudrate** ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

• **RF Data Rate** ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V2.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V2.0 ทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนดค่าความเร็วต่างกันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps
- 1 Mbps

• **RF Operation Mode** ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V2.0 ซึ่งสามารถกำหนด หน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ

- **RF Receive Only** เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้านRF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา
- **RF Transmit Only** เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา
- **RF Auto Direction** เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้เครื่อง ET-RF24G V2.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลา โดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RFจากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.7 คู่มือการใช้งาน RS232 to RF-Wireless (RF2.4GHz) CONVERTER รุ่น ET-RF24G V2.0**

**ETT.CO.,LTD -11-** www.etteam.com

• **RF Power Gain** เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBm

เป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ

- -20dBm (กำลังส่งต่ำสุด)
- -10dBm
- -5dBm
- +0dBm (กำลังส่งสูงสุด)

• **RXD ID Code** เป็นรหัส ID Code ของเครื่อง ET-RF24G V2.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V2.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V2.0 ที่อยู่ทางด้านรับทำ การรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่ รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอา เฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24G V2.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **TXD ID Code** เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V2.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **RF Frequency Channel** เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V2.0 จะทำการ รับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และ ใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมีประโยชน์เป็น อย่างมากในกรณีที่มีการใช้งานเครื่อง ET-RF24G V2.0 จำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของ ET-RF24G V2.0 กลุ่มที่จะสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร่วมกันไว้ที่ช่องความถี่เดียวกันส่วนกลุ่มอื่นๆก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกัน

**2.5.8 ข้อแนะนำในการกำหนดค่า Configuration**

การกำหนดค่า Configuration ให้กับเครื่อง ET-RF24G V2.0 นั้น สามารถเลือกกำหนดได้ตามความต้องการและจุดประสงค์ของการใช้งาน โดยแต่ละโหมดของการใช้งานนั้นจะมีค่า Configuration ที่เหมาะสมต่างกัน ซึ่งขอแนะนำวิธีการกำหนดค่า Configuration ดังแนวทางต่อไปนี้

- ความเร็วในการรับส่งข้อมูลด้าน RS232 หรือ User RS232 Baudrate ที่ความเร็ว 19200 Bps นั้น เหมาะกับการใช้งาน ET-RF24G V2.0 แบบ Receive Only หรือ Transmit Only ซึ่งมีการตรวจสอบความพร้อมของสัญญาณในการรับส่งข้อมูลกันด้วย แต่ถ้าต้องการใช้งานเครื่อง ET-RF24G V2.0 ในโหมด Auto Direction นั้น ควรกำหนดค่า User RS232 Baudrate ไว้ที่ความเร็วไม่เกิน 9600 Bps จะดีที่สุด และควรกำหนดค่า Baudrate ของทั้งสองฝ่ายให้มีค่าเท่ากันด้วย

- ค่าความเร็วของการรับส่งข้อมูลด้าน RF หรือ RF Data Rate ที่สามารถรับส่งข้อมูลกันได้ระยะทางไกล มากที่สุด และ มีโอกาสผิดพลาดน้อยที่สุด คือ 250Kbps

- ค่า RF Power Gain ที่ดีที่สุดคือ 0dBm ซึ่งเป็นค่ากำลังส่งสูงสุด ซึ่งจะทำให้สามารถส่งข้อมูลได้ระยะทางไกลที่สุด แต่ถ้าระยะการรับส่งข้อมูลไม่ไกลกันมาก และมีการใช้งานเครื่อง ET-RF24G V2.0 จำนวนหลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ก็อาจทำการลดกำลังส่งให้ต่ำลงเพื่อลดปัญหาการรบกวนกัน หรือกำหนดช่องความถี่ RF Frequency Channel ให้ห่างกันมากๆ

- ในกรณีที่มีการใช้เครื่อง ET-RF24G V2.0 หลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ควรกำหนดช่องความถี่ใน การใช้งาน หรือ RF Frequency Channel ให้ห่างกันด้วยเพื่อป้องกันการรบกวนกัน

- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ถ้ามีการส่งข้อมูลจำนวนมากๆ ควรจัดแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดข้อมูลชุดละไม่เกิน 64 Byte โดยในการส่งข้อมูลแต่ละชุดนั้น ให้ทำการส่งข้อมูลอย่างต่อเนื่องโดยให้ข้อมูลแต่ละ Byte มีระยะเวลาห่างกันไม่เกิน 2.5mS เนื่องจากถ้า ข้อมูลขาดหายไปนานกว่านี้ เครื่อง ET-RF24G V2.0 จะทำการเปลี่ยนโหมดของการส่งข้อมูลกลับเป็น โหมดของการรับข้อมูลแทน ซึ่งเมื่อมีการส่งข้อมูล Byte ถัดไปมาอีกก็จะต้อง

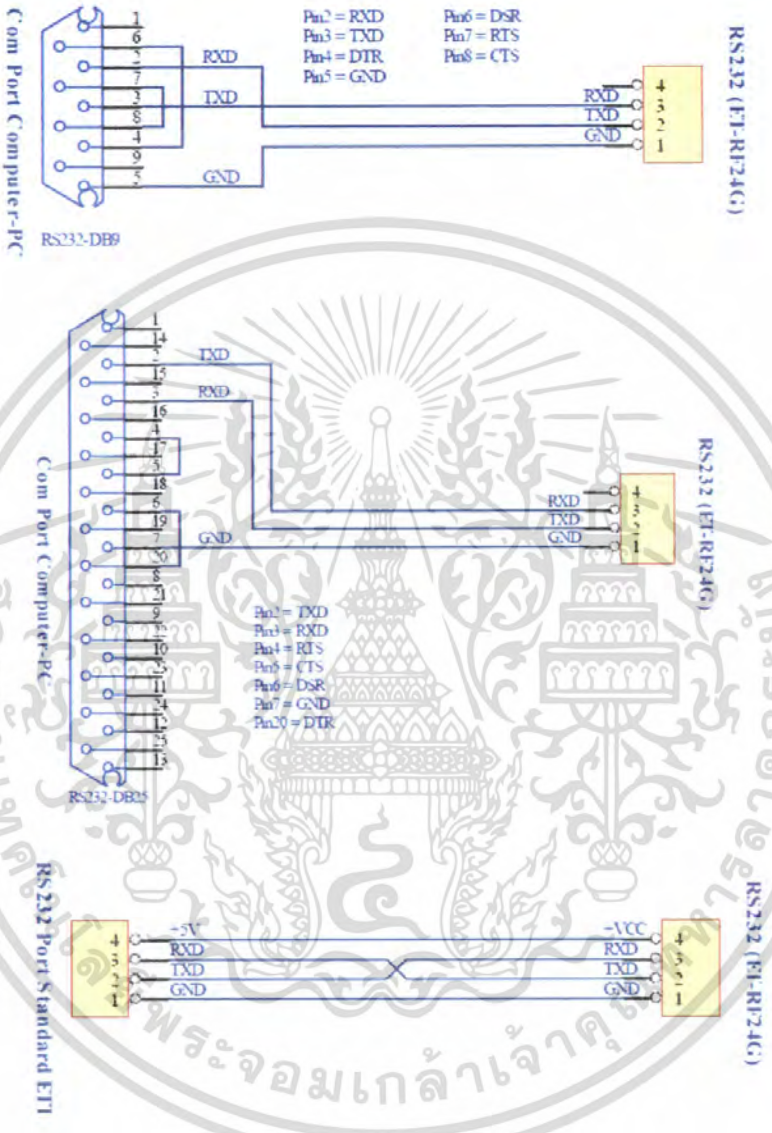
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียเวลาในการสลับโหมด จากฝ่ายรอรับข้อมูลให้เป็นฝ่ายส่งข้อมูลอีก ซึ่งจะทำให้ประสิทธิภาพในการจัดส่งข้อมูลลดลงเนื่องจาก ต้องเสียเวลาในการสลับโหมดการทำงานของวงจรภาค RF อยู่ตลอดเวลา โดยที่เมื่อทำการจัดส่งข้อมูล ครบ 64 Byte แล้ว ให้ทำการหน่วงเวลาไว้ชั่วขณะหนึ่ง ประมาณ 1mS-2mS แล้วจึงส่งข้อมูลชุดถัดไป อีกอย่างนี้เรื่อยๆ จะทำให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด

- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ควรหน่วงเวลาในการสลับโหมดจากฝ่าย ของการรอรับข้อมูลเป็นฝ่ายส่งข้อมูล อย่างน้อยที่สุด 3mS – 5mS ซึ่งถ้าส่งข้อมูลย้อนกลับด้วยเวลาที่ เร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

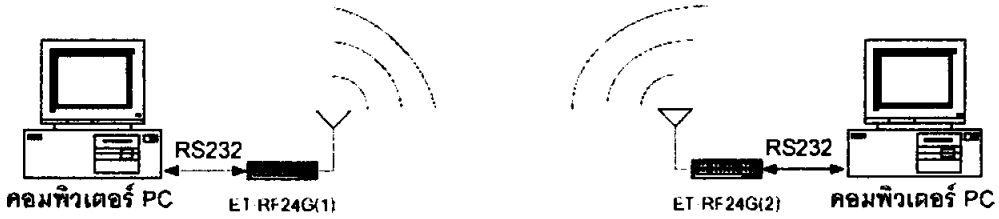
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.9 การเชื่อมต่อสัญญาณ RS232




**รูปที่ 2.34 แสดง แผนผังการต่อสาย RS232**
**เพื่อใช้งานกับ ET-RF24G V2.0 ในโหมด Auto Direction**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.35 ตัวอย่างที่ 1 การรับส่งข้อมูล 2 ทิศทาง**
**(Half Duplex) แบบ จุดต่อจุด (Point-to-Point)**

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบRS232 จำนวน 2 ชุดโดยต้องใช้รูปแบบการสื่อสารแบบ Half Duplex หรือ ผลัดกันรับ ผลัดกันส่ง กล่าวคือ ด้านรับจะต้องทำการรอรับข้อมูลจากด้านส่งจนครบทั้งหมด แล้วจึงจะส่งข้อมูลตอบกลับไปได้ ซึ่งจะไม่สามารถส่งข้อมูลสวนทางกลับไปในขณะที่กำลังรับข้อมูลอยู่ได้ โดยการสื่อสารแบบนี้ฝ่ายรับข้อมูลจะต้องรอให้รับข้อมูลได้ครบทั้งหมดเสียก่อน จากนั้น จึงจะส่งข้อมูลตอบกลับไปได้ โดยให้กำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V2.0 เป็นดังนี้

**ตารางที่ 2.4 แสดงกำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V2.0**

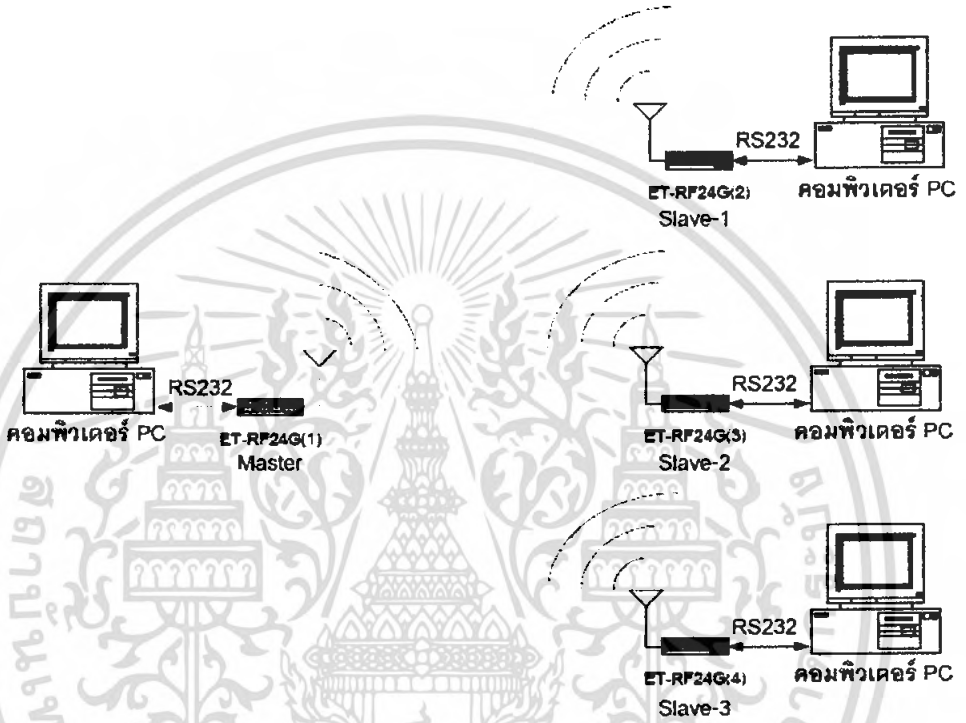
**ข้อสังเกตในการกำหนด Configuration**

| ค่า Configuration | ET-RF24G V2.0 ตัวที่1 | ET-RF24G V2.0 ตัวที่2 |
|---|---|---|
| User RS232 Baudrate | 9600 Bps | 9600 Bps |
| RF Data Rate | 250 Kbps | 250 Kbps |
| RF Operation Mode | Auto Direction | Auto Direction |
| RF Power Gain | +0dBm | +0dBm |
| RXD ID Code | 01 | 02 |
| TXD ID Code | 02 | 01 |
| RF Frequency Channel | 0 | 0 |

- ค่า RF Frequency Channel ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RXD ID Code ของตัวที่1 ต้องตรงกับ TXD ID Code ของตัวที่2
- ค่า TXD ID Code ของตัวที่1 ต้องตรงกับ RXD ID Code ของตัวที่2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทดสอบการทำงานด้วย Hyper Terminal นั้นให้ทดลองกดคีย์ใดๆ ในขณะที่ Run โปรแกรม Hyper Terminal อยู่ โดยจะสังเกตุเห็นตัวอักษรจากแป้นพิมพ์ของฝ่ายที่เป็นฝ่ายส่งข้อมูล จะถูกส่งออกไปแสดงผลที่หน้าจอโปรแกรม Hyper Terminal ของอีกฝ่ายหนึ่งในทันที




**รูปที่ 2.36 ตัวอย่างที่ 2 การรับส่งข้อมูล 2 ทิศทาง (Half Duplex) แบบ หลายๆจุด (RF Network)**

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวนหลายๆตัวร่วมกัน โดยหลักการสื่อสารแบบนี้จะให้ตัว Master เป็นตัวควบคุมการสื่อสารกับ Slave แต่ละตัวในระบบ โดยเมื่อ Master จะทำการส่งข้อมูลออกไปจะมีการใส่รหัส ID Code ของ Slave ที่ต้องการสื่อสารด้วย รวมไปในชุดข้อมูลนั้นๆด้วย ซึ่ง Slave ทุกๆตัวจะรับข้อมูลจาก Master ได้เหมือนๆกัน แต่จะมี Slave เพียงตัวเดียวที่ตอบสนองต่อข้อมูลนั้นๆ โดยให้กำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V2.0 เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.5 แสดงค่า Configuration ของตัวเครื่อง ET-RF24G V2.0**

| ค่า Configuration | ET-RF24G V2.0 ตัวที่1(Master) | ET-RF24G V2.0 ตัวที่2 (Slave1) | ET-RF24G V2.0 ตัวที่3 (Slave2) | ET-RF24G V2.0 ตัวที่4 (Slave3) |
|---|---|---|---|---|
| User RS232 Baudrate | 9600 Bps | 9600 Bps | 9600 Bps | 9600 Bps |
| RF Data Rate | 250 Kbps | 250 Kbps | 250 Kbps | 250 Kbps |
| RF Operation Mode | Auto Direction | Auto Direction | Auto Direction | Auto Direction |
| RF Power Gain | -0dBm | -0dBm | -0dBm | -0dBm |
| RXD ID Code | 01 | 02 | 02 | 02 |
| TXD ID Code | 02 | 01 | 01 | 01 |
| RF Frequency Channel | 0 | 0 | 0 | 0 |

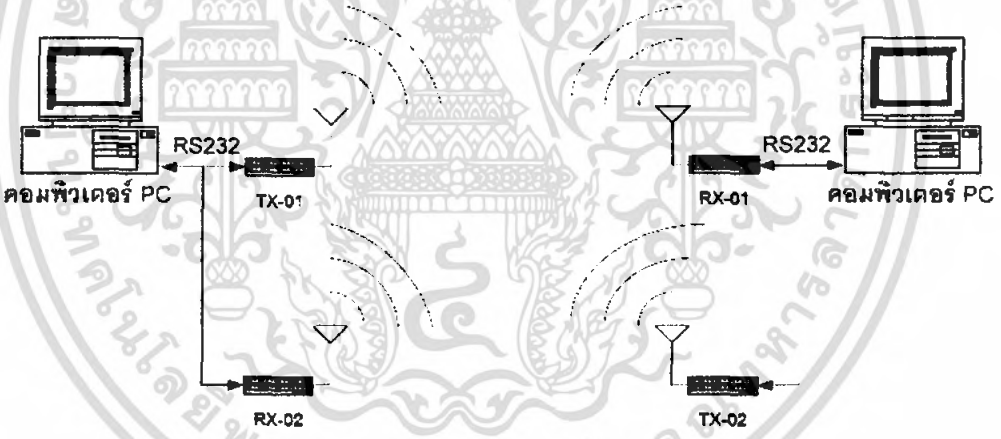
**ข้อสังเกตในการกำหนด Configuration**

- ค่า RF Frequency Channel ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RXD ID Code ของตัวที่1(Master) ต้องตรงกับ TXD ID Code ของตัวที่2-4(Slave1-3)
- ค่า TXD ID Code ของตัวที่1(Master) ต้องตรงกับ RXD ID Code ของตัวที่2-4(Slave1-3)

สำหรับการสื่อสารแบบนี้ จะต้องมีการกำหนด Protocol ขึ้นมาใช้ในการรับส่งข้อมูลกันด้วย ตัวอย่างเช่น กำหนดให้ใช้รหัส เครื่องหมาย '*' เป็นรหัสเริ่มต้นของชุดข้อมูล ตามด้วยรหัสหมายเลข ID Code ของ Slave ปลายทางเป็นตัวเลข 2 หลัก และจบด้วยรหัส Enter ดังนั้น ในการส่งข้อมูลแต่ละครั้ง Master จะต้องทำการส่งข้อมูลขนาด 4 Byte เสมอ และทางด้าน Slave ก็จะต้องรอรับข้อมูล โดยจะรอรับรหัสเครื่องหมาย '*' เป็นอันดับแรก ซึ่งเมื่อรับรหัสเครื่องหมาย '*' ได้แล้ว จึงรอรับข้อมูลถัดไปอีก 2 Byte จากนั้นจึงรอรับข้อมูล Byte ที่ 4 ซึ่งจะต้องตรวจสอบว่าเท่ากับรหัส Enter หรือไม่ ซึ่งถ้าใช่ก็แสดงว่ารับข้อมูลได้ถูกต้อง จากนั้นจึงทำการตรวจสอบข้อมูลในByte ที่ 2 และ 3 ว่าตรงกับรหัส ID Code ของตัวเองหรือไม่ โดย Slave-1 จะมีรหัสเป็น '0','1' ส่วน Slave-2 และSlave-3 ก็จะมีรหัส ID Code เป็น '0','2' และ '0','3' ตามลำดับ ซึ่งถ้าตรวจสอบแล้วพบว่าข้อมูลใน Byte ที่ 2 และ 3ตรงกับค่ารหัส ID Code ของตนเอง ก็ให้ตอบกลับด้วย รหัส ID Code ตามด้วยข้อความ 'OK' ซึ่งจากตัวอย่าง Protocol ข้างต้นจะได้ว่า เมื่อ Master ต้องการส่งข้อมูลไปยัง Slave-1 จะต้องมีการระบุ หมายเลข ID Code ของ Slave-1 รวมไปในชุดข้อมูลด้วย โดยจะส่งข้อมูลเป็น '*','0','1',Enter ออกไป ซึ่งข้อความดังกล่าวที่ส่งออกไปจาก Master นั้น ตัว Slave ทุกตัวจะสามารถรับข้อมูลได้เหมือนกันทั้งหมด ซึ่ง Slave ทุกตัวจะต้องทำการวิเคราะห์ข้อมูลที่รับได้ ซึ่งในที่นี้ Slave-1 จะต้องตอบกลับด้วยข้อมูล '*','0','1','O','K',Enter เป็นต้นซึ่งจากตัวอย่างที่ได้กล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายมานี้ เป็นเพียงตัวอย่างแนวทางขั้นต้น เท่านั้น ซึ่งในการนำไปใช้งานจริงนั้นจะต้องมีการดัดแปลงและเพิ่มเติมข้อกำหนดต่างๆเข้าไปในชุดข้อมูลอีก เช่น รหัสคำสั่ง รหัสตรวจสอบความถูกต้องของข้อมูล (Checksum) เป็นต้น ซึ่งข้อกำหนดต่างๆเหล่านี้ ผู้ใช้สามารถ คิดค้น ออกแบบรูปแบบของข้อมูลและคำสั่งต่างๆขึ้นมาใช้งานได้เองตามต้องการสำหรับการทดสอบการทำงาด้วยโปรแกรม Hyper Terminal นั้น อันดับแรกให้ทดสอบกดแป้นพิมพ์จากเครื่องที่เป็น Master ด้วยข้อความ *01 และ Enter คู ซึ่งจะเห็นข้อความดังกล่าวไปแสดงอยู่ที่หน้าจอโปรแกรมที่เป็นของตัว Slave ทุกๆตัวเหมือนกันหมด จากนั้นให้ทดลองคีย์ข้อความ *01OK และ Enter จาก Slave-1 ซึ่งก็จะเห็นข้อความนั้นไปปรากฏที่หน้าจอโปรแกรมของตัว Master ทันที ซึ่งในการทดสอบการทำงานนั้นจะเห็นได้ว่า เมื่อมีการส่งข้อมูลใดๆจาก Master ข้อมูลนั้นจะถูกส่งไปแสดงผลยังหน้าจอของ Slave ทุกตัวเหมือนกันหมด และเมื่อมีการส่งข้อมูลใดๆจาก Slave ไม่ว่าตัวใด ข้อมูลนั้นๆก็จะถูกส่งไปแสดงผลยังหน้าจอของ Master เช่นเดียวกัน แต่ข้อมูลที่ถูกส่งจาก Slave จะไม่ถูกส่งไปแสดงผลที่หน้าจอของ Slave ตัวอื่นๆเลย




**รูปที่ 2.37 แสดงตัวอย่างที่ 3 การรับส่งข้อมูลแบบ Full Duplex ด้วยโปรแกรมสำเร็จรูป**

ในตัวอย่างนี้จะเป็นการประยุกต์ใช้งานเครื่อง ET-RF24G V2.0 สำหรับทำการรับส่งข้อมูลกันแบบ Full Duplex โดยกำหนดโหมดการใช้งานเป็น RF Receive Only และ RF Transmit Only ฝ่ายละ 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.6 แสดงการกำหนด Configuration**

| ค่า Configuration | ET-RF24G V2.0 ฝั่งต้นทาง | | ET-RF24G V2.0 ฝั่งปลายทาง | |
|---|---|---|---|---|
| | ตัวที่1 (RF RX1) | ตัวที่2 (RF TX1) | ตัวที่3 (RF RX2) | ตัวที่4 (RF TX2) |
| User RS232 Baudrate | 19200 Bps | 19200 Bps | 19200 Bps | 19200 Bps |
| RF Data Rate | 250 Kbps | 250 Kbps | 250 Kbps | 250 Kbps |
| RF Operation Mode | RF RX Only | RF TX Only | RF RX Only | RF TX Only |
| RF Power Gain | +0dBm | +0dBm | +0dBm | +0dBm |
| RXD ID Code | 01 | - | 02 | - |
| TXD ID Code | - | 02 | - | 01 |
| RF Frequency Channel | 0 | 124 | 124 | 0 |

**ข้อสังเกตในการกำหนด Configuration**

- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RF Frequency Channel ของ ตัวรับ ด้านต้นทาง ต้องกำหนดให้ตรงกับ ตัวส่ง ด้านปลายทาง
- ค่า RF Frequency Channel ของ ตัวส่ง ด้านต้นทาง ต้องกำหนดให้ตรงกับ ตัวรับ ด้านปลายทาง
- ค่า RXD ID Code ของ ตัวรับ ด้านต้นทาง ต้องตรงกับ TXD ID Code ของ ตัวส่ง ด้านปลายทาง
- ค่า TXD ID Code ของ ตัวส่ง ด้านต้นทาง ต้องตรงกับ RXD ID Code ของ ตัวรับ ด้านปลายทาง

**2.5.10 เครื่องรับ – ส่ง กล้องวงจรปิด**

ตัวรับส่งสัญญาณภาพเป็นอุปกรณ์รับส่งสัญญาณภาพที่สามารถส่งสัญญาณภาพจากกล้องวงจรปิดมายังจอแสดงผลโดยส่งสัญญาณแบบไร้สาย ประกอบด้วย 2 ตัว ซึ่งประกอบด้วยตัวรับและตัว

**รูปที่ 2.38 แสดงเครื่องรับ – ส่ง สัญญาณภาพ 2 วัตต์**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชุดเพิ่มระยะทาง แบบไร้สายโดยให้ภาพที่คมชัดเหมือนต้นฉบับ สามารถใช้กับ กล้องวงจรปิด /กล้องถ่ายวีดีโอ / กล้องดิจิตอล / โปรเจ็คเตอร์และเครื่องเล่น / เครื่องเสียงและอื่นๆ

**2.5.10.1 คุณสมบัติการทำงานของเครื่องรับ – ส่ง สัญญาณภาพ 2 วัตต์**

- ทำงานด้วยคลื่นวิทยุในย่านความถี่ 2.4 GHz-2.4835 GHzไม่รบกวนความถี่อื่น
- สามารถผ่านกำแพงตึก/ชั้นต่างๆ / ห้องต่างๆได้ดี ระยะทำการที่หวังผลไกลมากกว่า 200 เมตร ถึง 300 เมตร
- ใช้แหล่งจ่ายไฟในการทำงาน 9-12 v กระแส 500 mA
- สามารถเลือกช่องสัญญาณได้ทั้งหมดสี่ช่องเพื่อไม่ให้ไปรบกวนกับกล้องตัวอื่นที่เป็นระบบเดียวกัน

| | | | | |
|---|---|---|---|---|
| Channel | 1 | – | 2.414 | GHz |
| Channel | 2 | – | 2.432 | GHz |
| Channel | 3 | – | 2.450 | GHz |
| Channel | 4 | – | 2.468 | GHz |

- สามารถส่งได้ทั้งสัญญาณภาพ และสัญญาณเสียงระบบเสตอริโอแยกซ้าย-ขวาได้
- ส่งสัญญาณแบบ FM
- Video input level : 1Vp-p
- Audio input level : 1Vp-p
- Video Impedance : 75Ω
- Audio Impedance : 600Ω
- Input Port: RCA to RCA line jacks
- Antenna: Dipole antenna

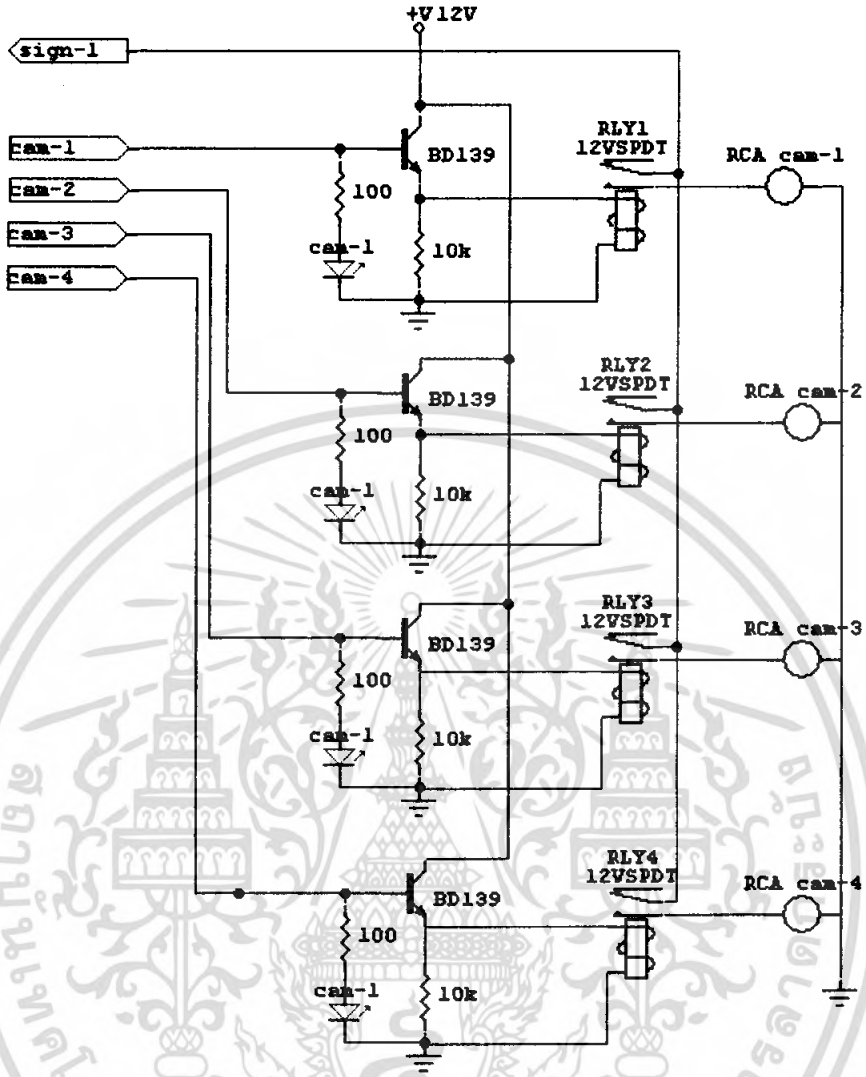
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.39 ส่วนประกอบของตัวรับ-ส่งสัญญาณภาพและเสียง**

**2.5.10.2 วงจรตัดต่อการทำงานของชุดเลือกสัญญาณภาพ**

ในหุ่นยนต์กู้ภัยนี้ที่ได้ทำขึ้นมาอย่างที่กล่าวไว้ข้างต้นแล้วว่าใช้กล้องในการทำงานทั้งหมด 3 ตัวโดยแต่ละตัวทำหน้าที่ต่างกัน แต่ใช้ชุดรับส่งสัญญาณเพียงชุดเดียวดังนั้น ที่ตัวหุ่นยนต์กู้ภัยจึงจำเป็นต้องมีวงจรเพื่อที่จะตัดต่อเลือกช่องสัญญาณกล้องที่ต้องการใช้งาน โดยมีวงจรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.40 วงจรเลือกช่องสัญญาณภาพ**

**โดยมีหลักการทำงานของวงจรดังนี้ เมื่อต้องการเลือกกล้องเพื่อที่จะให้กล้องทำงานเรา สามารถเลือกได้โดย ป้อนสัญญาณทริกซ์ที่ cam-1, cam-2, cam-3, cam-4 เพื่อเลือกกล้องตามลำดับ จากตัวที่หนึ่งถึงตัวที่สี่ โดยสัญญาณจะทริกซ์เลือกให้ รีเลย์ทำงานโดยรีเลย์จะเป็นตัวเลือกกล้องแล้ว ส่งข้อมูลไปที่เครื่องส่ง และส่งข้อมูลออกไปให้กับเครื่องรับเพื่อจะแสดงผลต่อไป**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.41 แสดงวงจรที่เป็นตัวเลือกกล้องทั้ง 2 ตัว

เนื่องจากว่าถ้าเราทำขึ้นมาเองโดยใช้แจ๊ค RCA มันจะทำให้เกิดสัญญาณกวนกันจึงต้องซื้อเป็นแผงมาแล้วมาทำการต่อวงจรใส่ที่แผงด้านหลังเองเลือกใช้แจ๊ค RCA ที่เป็นสีเหลืองที่ใช้สัญญาณภาพทั้ง 4 ช่องเพียงอย่างเดียว

## 2.6 คอมพิวเตอร์

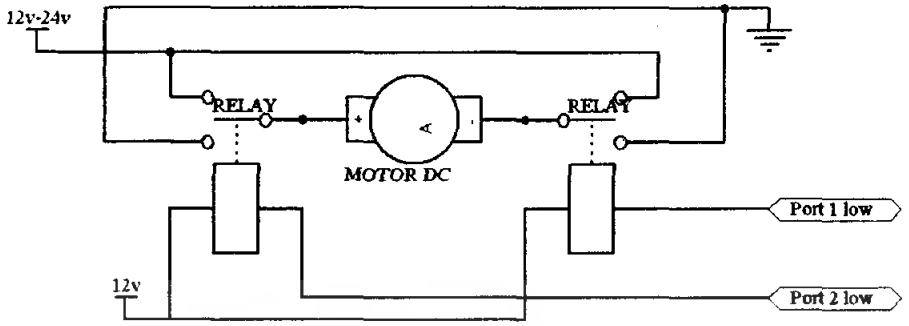
เป็นหัวใจหลักของการควบคุม เนื่องในการควบคุมหุ่นยนต์นั้นเราจะควบคุมผ่านแป้นพิมพ์ของเครื่องคอมพิวเตอร์โดยเราจะเชื่อมต่อเครื่องรับเครื่องส่งเข้ากับเครื่องคอมพิวเตอร์ และยังใช้เครื่องคอมพิวเตอร์เป็นตัววาดแผนที่โดยคำสั่งจะส่งมาจากเข็มทิศอิเล็กทรอนิกส์

## 2.7 Driver Motor

### 2.7.1 ชุด Driver Motor

วงจร Drive Motor นั้นประกอบไปด้วย Relay 2 ตัว จะสามารถควบคุมมอเตอร์ได้ 1 ตัว เนื่องจาก Relay ตัวที่ 1 จะทำหน้าที่หมุนซ้าย Relay ตัวที่ 2 จะทำหน้าที่หมุนขวา โดยมีหลักการทำงานคือ แรงดันทั้งสองข้างไหลเข้า Relay ทั้งสองตัวเพื่อเป็นตัวป้อนไฟให้กับมอเตอร์ จากนั้น Relay ทั้งสองตัวต้องการให้มอเตอร์ตัวใดตัวหนึ่งทำงาน ก็จะมีสัญญาณไฟลบมาทริกที่ Relay มอเตอร์ตัวนั้นก็จะทำงาน โดยจะมี Post หนึ่งป้อนเป็นไฟลบ Post หนึ่งป้อนเป็นไฟบวกสลับกัน โดยสมมุติว่าต้องการให้มอเตอร์หมุนไปข้างหน้าดังนั้นแรงดันที่ Post 1 และ Post 2 จะเป็นไฟลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.42 วงจร Drive Motor ที่ทำจาก Relay**

**2.8 มอเตอร์**

**คือ เครื่องกลที่ใช้สำหรับเปลี่ยนพลังงานไฟฟ้า ออกมาเป็นพลังงานกลโดยขดลวดที่พันรอบแกนโลหะที่วางอยู่ระหว่างขั้วแม่เหล็ก โดยเมื่อผ่านไฟฟ้าไปยังขดลวดที่อยู่ระหว่างขั้วแม่เหล็ก จะทำให้ขดลวดหมุนไปและเมื่อสลับขั้วไฟฟ้า การหมุนของขวดลวดจะหมุนกลับทิศทางกัน มอเตอร์มี 2 ประเภท คือ**

**2.8.1 ประเภทของมอเตอร์**

**1. มอเตอร์ดีซี (Motor DC) หรือมอเตอร์กระแสตรง**

**เป็นมอเตอร์ที่ต้องการใช้ไฟฟ้ากระแสตรงผ่านขดลวดอาร์เมเจอร์เพื่อทำให้เกิดการดูดและผลักกันของมเหล็กถาวรกับไฟฟ้าที่เกิดจากขดลวดมอเตอร์จึงหมุนได้**

**2. มอเตอร์เอซี (Motor AC) หรือมอเตอร์กระแสสลับ**

**เป็นมอเตอร์ที่ต้องการใช้ไฟฟ้ากระแสสลับหลักการดูดผลักกันของแม่เหล็กถาวรกับแม่เหล็กไฟฟ้าจากขดลวดจะเกิดการหมุนของมอเตอร์**

**2.8.2 ส่วนประกอบที่สำคัญของมอเตอร์มี 4 ส่วนดังนี้**

**1. ขั้วแม่เหล็ก N และ S ซึ่งทำหน้าที่สร้างสนามแม่เหล็ก ในมอเตอร์ ขั้วแม่เหล็ก อาจเป็น แม่เหล็กถาวร หรืออาจทำจากแม่เหล็กไฟฟ้าก็ได้ ในมอเตอร์ เรียกขั้วแม่เหล็ก N และ S นี้ว่า สเตเตอร์ (Stator)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขดลวดอาร์เมเจอร์ (Armature) ซึ่งหมุนได้รอบตัว เมื่อมีกระแสไฟฟ้าผ่านเข้าไป ในขดลวดอาร์เมเจอร์ ที่วางอยู่ในสนามแม่เหล็ก จะทำให้เกิดแรงกระทำต่อขดลวด แล้วเกิดโมเมนต์คู่ควบ หมุนขดลวดอาร์เมเจอร์

3. วงแหวนผ่าซีก หรือ Commutator เป็นส่วนประกอบสำคัญ ที่จะทำให้กระแสที่ไหลผ่านขดลวดอาร์เมเจอร์ ไหลในทิศที่ทำให้เกิดโมเมนต์คู่ควบ หมุนขดลวดอาร์เมเจอร์ในทิศเดียวกันตลอดเวลา

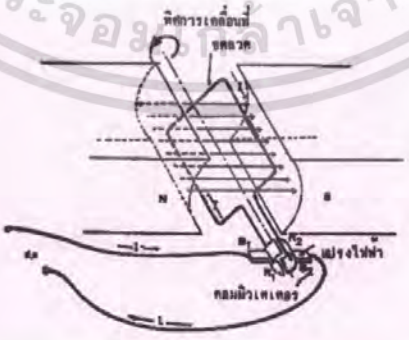
4. แปรงคาร์บอน ทำหน้าที่สัมผัสเบาๆ กับ Commutator โดยที่แปรงทั้งสองอยู่กับที่ และใช้สำหรับต่อกับวงจรไฟฟ้า

### 2.8.3 ลักษณะของมอเตอร์

ลักษณะของมอเตอร์นั้น คล้ายไดนาโม แต่มีส่วนที่สำคัญคือ แหวนครึ่งซีก เพื่อทำหน้าที่บังคับให้กระแสวิ่งอยู่ทางเดียว ถ้าไม่มีแหวนครึ่งซีกแล้ว ขดลวดจะพลิกกลับไปมา เริ่มแรก ลวดด้าน AB อยู่ติดกับแหวน E ลวดด้าน CD อยู่ติดกับแหวน F ตามรูป กระแสเข้าตามทิศทาง EAB (เข้าไปข้างใน) และกระแสออก ทางด้าน CDF (ออกมาข้างนอก)

### 2.8.4 การให้กระแสเข้าขดลวด

เริ่มหมุนในทิศทวนเข็มนาฬิกา สมมติลวดหมุนได้ครึ่งรอบ จะ เห็นว่าลวด CD มาแทน AB และ AB มาแทน CD ฉังหวะนี้ ลวด AB จะได้กระแสตามทิศ CDF, ลวด CD จะได้กระแสทิศ EAB ทำให้ขดลวดนี้ สามารถ หมุนไปได้ทางเดียวเรื่อยๆ ถ้าหากไม่มีแหวนครึ่งซีก คือเป็นแหวน 2 วง กระแส ไม่มีถูกตัดช่วง ลวดแต่ละฝ่ายจะได้รับกระแสทางเดียวตลอด ทำให้ขดลวดพลิกกลับไปกลับมา




**รูปที่ 2.43 แสดงหลักการทำงานของมอเตอร์ดีซี**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.8.5 หลักการทำงานของดีซีมอเตอร์**

ดีซีมอเตอร์เป็นทรานสดิวเซอร์แรงบิดซึ่งมีการออกแบบให้คุณลักษณะพิเศษคือ แรงบิดของเพลาของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ ความสัมพันธ์ระหว่างแรงบิดของเพลาและกระแสคือ

$$T = K\Phi I$$

เมื่อ T คือ แรงบิดของเพลา หน่วย นิวตัน-เมตร
I คือ กระแสที่ไหลผ่านอาร์เมเจอร์ หน่วย แอมแปร์
$\Phi$ คือ เส้นแรงแม่เหล็ก หน่วย เวเบอร์
K คือ ค่าคงที่

จะเห็นได้ว่า แรงบิดของเพลาเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส เมื่อขดลวดเคลื่อนที่ในสนามแม่เหล็ก จะทำให้เกิดแรงดันตกคร่อม ซึ่งแรงดันนี้เป็นสัดส่วนกับความเร็วของเพลาของมอเตอร์และเส้นแรงแม่เหลก มีความสัมพันธ์ดังนี้

$$E = K\Phi\omega$$

เมื่อ E คือ แรงดันย้อนกลับ หน่วย โวลต์
$\omega$ คือ ความเร็วของมอเตอร์ หน่วย เรเดียน
$\Phi$ คือ เส้นแรงแม่เหล็ก หน่วย เวเบอร์
K คือ ค่าคงที่

**2.8.5.1 ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่**

ฟิลด์ (Field) ของมอเตอร์ชนิดนี้นิยมใช้แม่เหล็กถาวร มอเตอร์ชนิดนี้จึงอาจจะเรียกว่า มอเตอร์แบบแม่เหล็กถาวร (Permanent – Magnet Motors) มอเตอร์ชนิดนี้ไม่ต้องใช้การกระตุ้นที่ทำให้เกิดสนามแม่เหล็ก เนื่องจากมีแกนขั้ว ซึ่งทำจาก อัลลอย ที่มีคุณสมบัติรักษาความเป็นแม่เหล็กหรือขดลวดฟิลด์มอเตอร์ชนิดนี้มีประสิทธิภาพสูงกว่ามอเตอร์ชนิดชันท์ หรือมอเตอร์คอมปาวด์ที่มีพิกัดที่เท่ากัน นอกจากนี้ยังสามารถควบคุมความเร็วและทิศทางการหมุนได้ง่าย โดยป้อนขนาดและขั้วแรงดันที่ป้อนให้แก่มอเตอร์ นั่นคือ ถ้ามีการกลับขั้วของแรงดัน ทำให้ทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของกระแสที่ไหลผ่านมอเตอร์ (หรือไหลผ่านอาร์เมเจอร์) เปลี่ยนแปลงไปอย่างเดียวเท่านั้น แต่ไม่ได้ทำให้ฟลักซ์เปลี่ยนแปลง จึงได้อัตราส่วนผสมระหว่างกระแสอาร์เมเจอร์ และแรงบิดที่มีค่าคงที่ ดังนี้

$$T=KTi \text{ และ } E=Ke$$

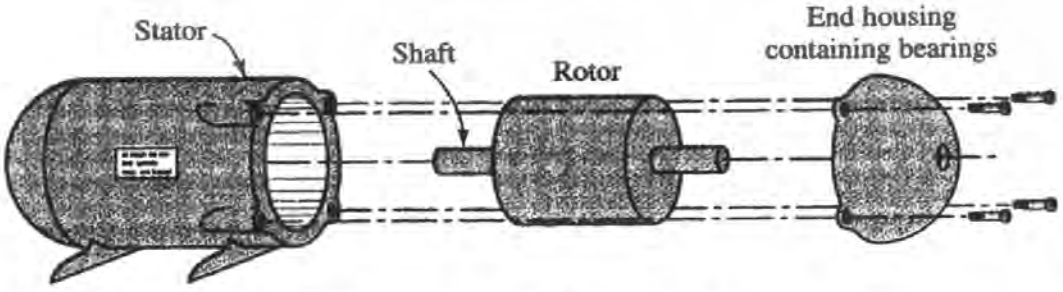
ระบบนี้จะให้ความสัมพันธ์ระหว่างกระแสอาร์เมเจอร์, แรงบิด และความเร็วอยู่ในลักษณะเชิงเส้น สมการทางไฟฟ้าของดีซีมอเตอร์จึงเขียนเป็น

$$V(t)+Ke+L[\, di(t)dt \,]+Ri(t)$$

เมื่อ V(t) คือ แรงดันที่ป้อน หน่วย โวลต์
L คือ ค่าความเหนี่ยวนำของขดอาร์เมเจอร์ หน่วย เฮนรี
R คือ ค่าความต้านทานที่ขั้วของมอเตอร์ หน่วย โอห์ม
i(t) คือ กระแสที่ไหลผ่านอาร์เมเจอร์ หน่วย แอมแปร์

2.8.5.2 ดีซีมอเตอร์แบบอาร์เมเจอร์เป็นแกนหลัก

โครงสร้างของโรเตอร์และสเตเตอร์ของมอเตอร์แบบแกนหลัก โครงสร้างของมอเตอร์ชนิดนี้มีโมเมนท์ของแรงเฉื่อยสูงสุด และมีค่าอินดัคแตนซ์ของโรเตอร์สูงสุดด้วย ดังนั้นมอเตอร์นี้จึงมีปริมาณการดูดความร้อนได้สูง และสามารถทนต่อโอเวอร์โหลดได้ในระยะเวลาที่ยาวนานได้โดยไม่ทำให้มอเตอร์เสียหายได้




รูปที่ 2.44 โครงสร้างพื้นฐานมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.8.5.3 โครงสร้างพื้นฐานมอเตอร์ประกอบด้วย**

**1. สเตเตอร์ (Stator) คือ ส่วนที่อยู่กับที่ ประกอบด้วยขดลวดที่พันโดยรอบ**

**2. โรเตอร์ (Rotor) คือ ส่วนที่สามารถหมุนรอบโดยเชื่อมต่อกับแกน และโหลด ประกอบด้วยขดลวดที่พันโดยรอบเช่นเดียวกัน**

**3. ฝาครอบลูกปืน (Bearings) คือ ส่วนที่รองรับแกนโรเตอร์ ซึ่งสามารถหมุนได้อย่างอิสระ**

**2.9 กล้องวงจรปิด**

**แบ่งได้เป็น 2 ชนิด**

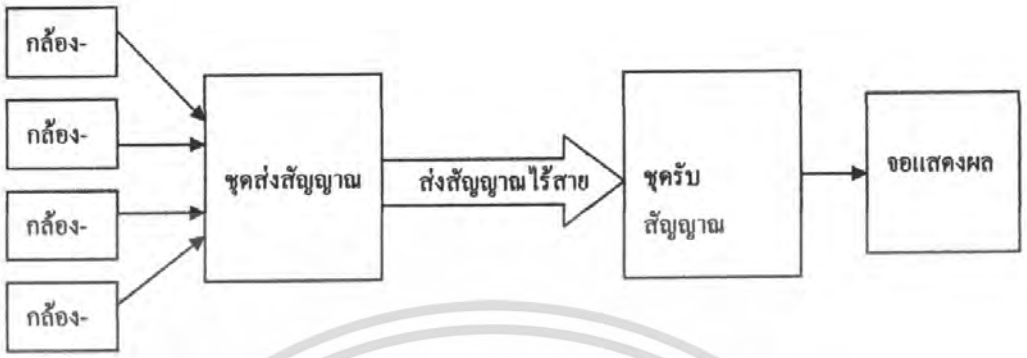
**1. กล้อง CCTV**

**2. กล้องสีแบบธรรมดา**

**2.9.1 กล้อง CCTV**

**ในหุ่นยนต์กู้ภัยเราสามารถดูภาพเหตุการณ์ต่างๆที่อยู่ภายพื้นที่ที่หุ่นยนต์กู้ภัยได้จากกล้องที่ส่งสัญญาณแบบไร้สาย และการงานของชุดรับ-ส่งสัญญาณภาพที่ใช้ในหุ่นยนต์กู้ภัยจะเป็นลักษณะการทำงานแบบทางเดียวคือ ใช้กล้องจากตัวหุ่นยนต์ดูภาพและก็ส่งแบบไร้สายไปที่ตัวรับสัญญาณได้ทางเดียวเพราะเราได้ใช้งานเพียงทางเดียว โดยสัญญาณกล้องไร้สายได้ใช้ตัวกล้องและเครื่องส่งสัญญาณที่สามารถส่งสัญญาณภาพ โดยเหตุผลที่ต้องใช้กล้องแบบไร้สายเพราะว่าเราได้ทำการบังคับหุ่นยนต์กู้ภัยแบบไร้สายเพราะว่าการมองพื้นที่ที่เรามองไม่เห็น เราสามารถมองผ่านพื้นที่ภายในที่ประสบภัยหรือที่เกิดอุบัติภัย โดยเป็นการเสี่ยงที่จะส่งมนุษย์เข้าไปสำรวจในเหตุการณ์จริง โดยกล้องที่ใช้ต้องมีความทนทานต่อการสั่นสะเทือนเพราะว่าหุ่นยนต์กู้ภัยเมื่อเคลื่อนที่เดินทางไปในแต่ละจุดในพื้นที่ เป็นพื้นที่ที่ขรุขระทำให้เกิดการสั่นสะเทือน และตัวส่งสัญญาณต้องเป็นตัวส่งสัญญาณที่มีคุณภาพเพราะจะต้องส่งสัญญาณได้ระยะไกลๆและมีความถี่ที่ส่งสัญญาณได้อย่างเข้มข้นเพราะว่าจะต้องทนทานต่อสัญญาณรบกวนต่างๆที่จะมารบกวนเพราะว่าการใช้งานจะต้องไม่ผิดเพี้ยน ไม่เกิดการรบกวนเพราะการทำงานจะเกิดความผิดพลาด แล้วผลที่ได้จะไม่ตรงตามความต้องการ กล้องต้องสามารถตรวจสอบสีได้เพื่อที่จะดูพื้นที่ที่ประสบภัยได้ชัดเจนและบอกได้ว่าจุดไหนคืออะไร เป็นสีอะไรเพื่อจะได้กำหนดเป้าหมายได้ถูกต้องและเป็นไปตามเป้าหมายที่ต้องการ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.45 บล็อกไดอะแกรมการทำงานของภาครับ-ส่งสัญญาณภาพ**

**2.9.1.1 หลักการทำงาน**

โดยในตัวหุ่นยนต์กู้ภัยตัวนี้ที่ทางสมาชิกในทีมได้ร่วมกันทำได้ใช้กล้องทั้งหมด 4 ตัว โดยมีกล้อง CCD 1 ตัว และมีกล้องสีธรรมดา 3 ตัวที่ใช้งาน 2 ตัวจาก 3 ตัวใช้มองฝั่งซ้ายและขวา เพื่อความเร็วในการสังเกตพื้นที่รอบข้างได้เร็วขึ้น และง่ายในการตัดสินใจเลือกเส้นทางที่จะเดินไปของหุ่นยนต์ว่าควรไปทางไหนเพื่อจะได้เร็วในการทำงาน โดยมีข้อมูลของกล้องและตัวส่งแยกได้เป็นส่วนๆ คือ

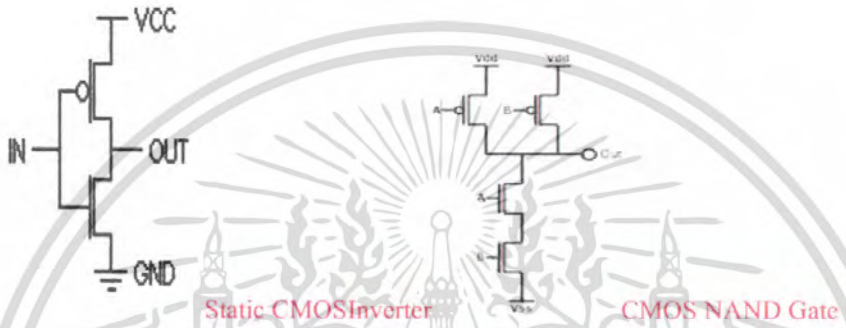
**2.9.1.2 ข้อมูลกล้อง CCTV**

กล้องที่ใช้เป็นกล้อง CCTV ที่สามารถนำมาต่อสายโดยตรงหรือสามารถนำมาต่อกับเครื่องรับ-ส่งสัญญาณเพื่อทำการส่งข้อมูลแบบไร้สายได้ ในที่นี้เราได้ต่อร่วมกับชุดรับ – ส่งสัญญาณเพราะว่าได้ทำการนำไปใช้งานแบบไร้สาย

**รูปที่ 2.46 แสดงรูปภาพของกล้อง CCTV**

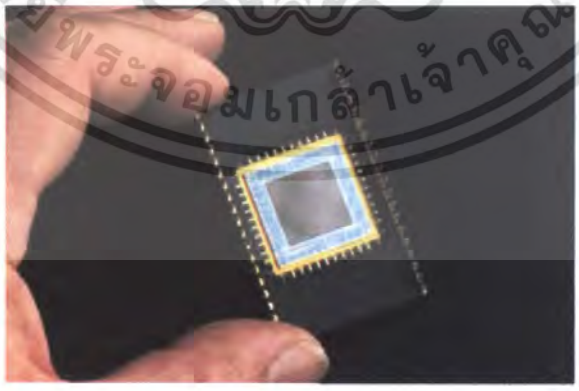
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องที่เรานำมาใช้นั้นเป็นกล้องชนิด CMOS ย่อมาจากคำว่า Complementary Metal Oxide Semiconductor ซึ่งมีพื้นฐานมาจาก เทคโนโลยีการผลิตสารกึ่งตัวนำ มีคุณสมบัติเด่นในเรื่องของการบริโภคพลังงานต่ำและมีความร้อนสะสมต่ำ โดยการทำงานอาศัยทรานซิสเตอร์พื้นฐานหลายๆ ตัว นิยมมากในการนำมาใช้งานกับอุปกรณ์ประเภท Microprocessor, Microcontroller และ Static RAM




**รูปที่ 2.47 โครงสร้างของ CMOS**

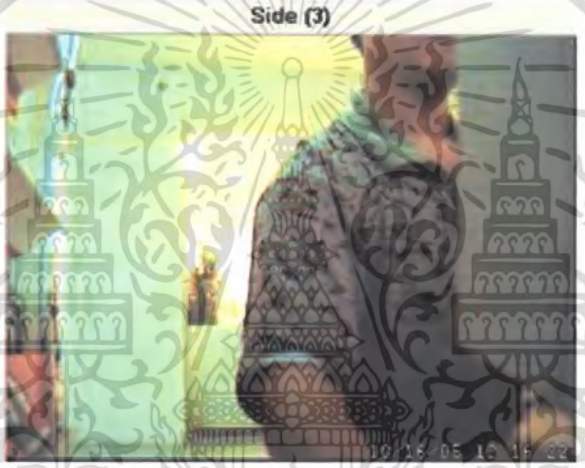
ซึ่งผลิตขึ้นโดยเฉพาะเพื่อจุดประสงค์ให้เป็นอุปกรณ์รับแสงในรูปแบบต่างๆ โดยประกอบด้วย IC ที่จัดเรียงแถวเชื่อมต่อ หรือจับคู่กันเป็นจำนวนมาก และตัวเก็บประจุที่ไวต่อแสง ที่ถูกสร้างขึ้นมาเพื่อใช้กับกล้องถ่ายภาพด้วยรังสี Ultraviolet ตัวชิพบรรจุอยู่ในตัวถัง Wire Bonded แบบ IC ทั่วไปพร้อมใช้งาน



**รูปที่ 2.48 แสดงชิพบรรจุอยู่ในตัวถัง Wire Bonded แบบ IC ทั่วไปพร้อมใช้งาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า เทคโนโลยี CMOS จะมีจุดเด่นที่มีราคาถูก ใช้พลังงานต่ำ และทำให้มีขนาดเล็กได้ง่ายกว่า ยิ่งถ้าหากใช้เทคโนโลยีล่าสุดมาใช้ในการผลิตก็สามารถลดสัญญาณรบกวนได้มากกว่าในอดีต (ตัวอย่างเช่นกล้องดิจิตอล Canon หลายรุ่นใช้ CMOS Sensor ในการผลิตตัวรับแสง) แต่สำหรับกล้อง CMOS ที่อยู่ในตลาดกล้องวงจรปิด โดยเฉพาะหากผลิตมาจากประเทศจีน คงคาดหวัง "latter technology" ได้ยาก การใช้งานจึงต้องเน้นรองรับการใช้งานแบบ "พอดูได้" ว่ามีคนหรือวัตถุอยู่ตรงหน้ากล้องเท่านั้น หากต้องลงรายละเอียดมากก็คงไม่ได้ รวมทั้งการเดินสายยาวๆ ก็จะไม่เหมาะสมทั้งกำลังของสัญญาณที่ต่ำกว่าและการถูกรบกวนของสัญญาณที่ถูกรบกวนได้ง่ายกว่า




**รูปที่ 2.49 แสดงภาพที่ได้จากกล้อง CMOS**

ส่วนกรณีของ CCD พบว่ามีจุดเด่นที่ มีความไวแสงสูงกว่า สัญญาณรบกวนต่ำกว่า และมีช่วงกว้างในการรับแสงที่กว้างกว่า คือตั้งแต่ UV จนถึง IR ได้เลยทีเดียว ซึ่งเป็นคุณสมบัติหลักใหญ่ใจความ ที่อุปกรณ์รับแสงในกล้องพึงจะมี การใช้งานจึงให้ภาพที่สว่างใสกว่า ให้รายละเอียดมิติมากกว่า และสามารถเดินสายไกลได้ยาวกว่ากล้อง CMOS แต่ราคาของกล้อง CCD ก็สูงกว่าด้วยเช่นกัน ดังนี้เองผู้ผลิตชิพรับภาพรายใหญ่ ไม่ว่าจะเป็น Sony, Panasonic, Sharp หรือ Samsung ต่างก็ผลิตชิพภายใต้เทคโนโลยีแบบ CCD ทั้งสิ้น ในขณะที่ CMOS กลายเป็นชิพในตลาดล่างที่โรงงานห้องแถวที่ไหนก็สามารถซื้อหามาผลิตกล้องขายได้เช่นกันส่วนเรื่องอายุการใช้งาน แม้ว่ากล้อง CMOS จะมีการสะสมความร้อนที่ต่ำกว่า แต่ด้วยมาตรฐานการผลิตที่สูงกว่า ของ CCD ทำให้อายุการใช้งานของกล้อง CCD ที่ผลิตจากโรงงานที่น่าเชื่อถือ สามารถใช้งานได้ยาวนานกว่าเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**คุณสมบัติ มีดังต่อไปนี้**

-1/4 นิ้ว SHARP CCD

-ใช้ได้กับ TV ทั้งระบบ PAL(CCIR) และ NTSC(EIA)

-Picture Element 500(H)x582(V) สามารถเปลี่ยนเลนส์ได้

-S/N ration : Mor than 48dB

-Lens : C/CS Mount

-White Balance : Auto

-Scan Frequency 50Hz

-Video output 1.0 Vp-p,75Ohm,BNC connector

-Power sorce DC 12v

**2.9.2 กล้องสีธรรมดา**

เป็นกล้องที่ใช้ทำงานเพิ่มเติม ทำหน้าที่มอง ซ้าย-ขวา และอุณหภูมิ ซึ่งมีขนาดเล็กน้ำหนักเบาและราคาถูก

**รูปที่ 2.50 แสดงกล้องสีทั้ง ด้านหน้า ด้านบน ด้านข้าง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การออกแบบส่วนต่างๆ ของหุ่นยนต์

## 3.1 โครงสร้างการทำงานของหุ่นยนต์

หุ่นยนต์กู้ภัยนี้มีโครงสร้างการทำงานที่สลับซับซ้อนมาก แต่มีโครงสร้างการทำงานหลักๆ อยู่ 3 ส่วนใหญ่ๆ สามารถแบ่งได้เป็น

1. หุ่นยนต์
2. การสื่อสารแบบไร้สาย
3. คอมพิวเตอร์

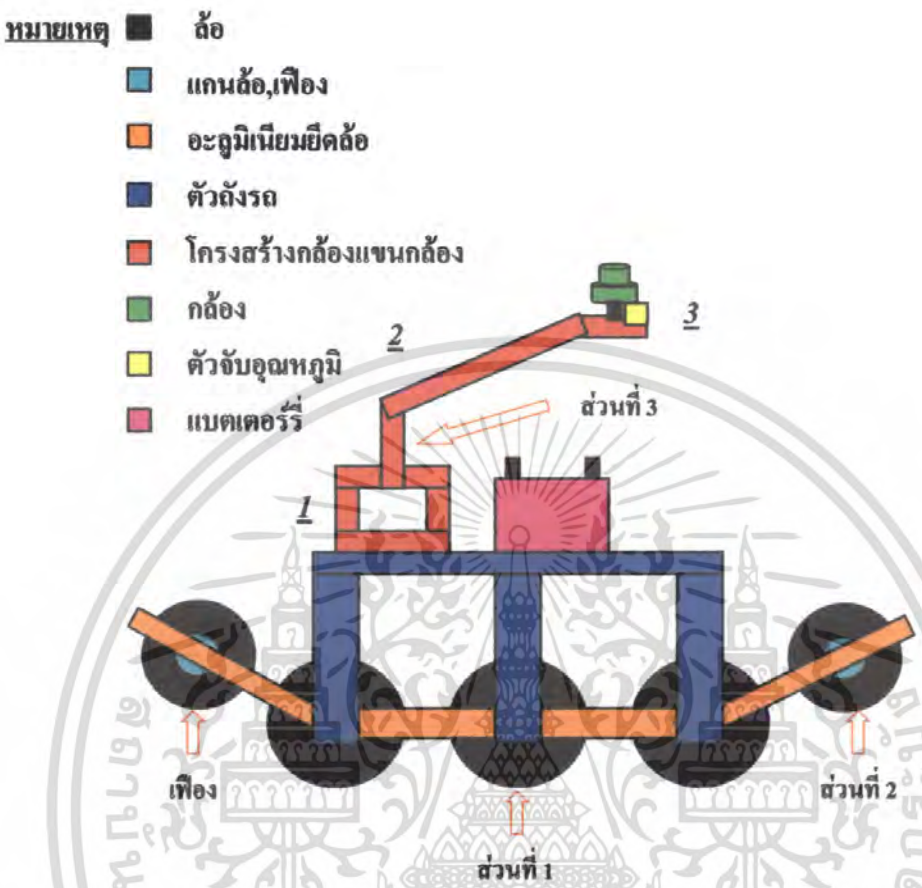
แสดงให้เห็นในบล็อกไดอะแกรมด้านล่าง ดังนี้

**รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานในส่วนหลักๆ**

### 3.1.1 ส่วนต่างๆ ของหุ่นยนต์

โครงสร้างในส่วนของหุ่นยนต์ก็สามารถแบ่งออกเป็นรายละเอียดปลีกย่อยอีกมากมาย เช่น การขับเคลื่อน แขนที่ให้ยกกล้องวงจรปิด แขนที่ทำหน้าปืนป่าย เป็นต้น เราจะแสดงโครงสร้างในส่วนต่างๆ ให้เห็นได้ในรูปด้านล่าง ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.2 แสดงโครงสร้างด้านข้างและการทำงานในส่วนต่างๆ**

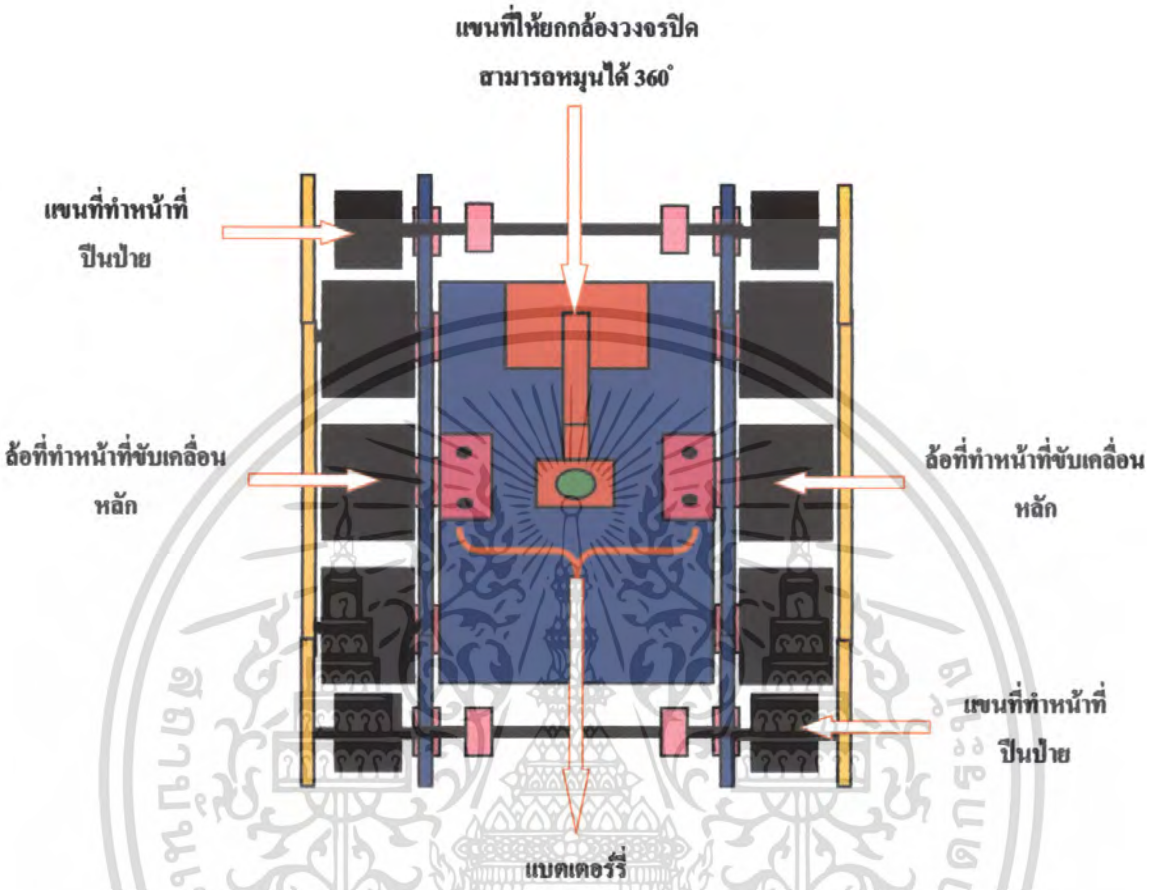
การทำงานทั้งหมดของ Robot จะสามารถทำงานได้ทุกส่วน เช่น ล้อจะสามารถขยับได้โดยเราจะใช้โซ่ในการขับเคลื่อนโดยจะมีมอเตอร์ที่จะทำการหมุนเดินหน้าและถอยหลังเราสามารถแบ่งเป็นส่วนต่างๆ ได้ดังนี้

ส่วนที่ 1 เป็นส่วนที่ทำหน้าที่หลักคือการขับเคลื่อนหุ่นยนต์ให้เดินหน้าถอยหลัง

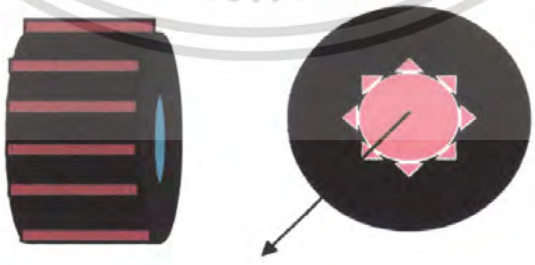
ส่วนที่ 2 เป็นส่วนที่สามารถขยับขึ้นลง เพื่อทำการปีนป่ายในพื้นที่สูงชันหรือเครื่องกีดขวางได้ และล้อก็สามารถหมุนไปล้อในตามส่วนที่ 1

ส่วนที่ 3 เป็นส่วนที่ทำหน้าที่เป็นแขนของกล้องวงจรปิดสามารถหมุนได 360° ส่วนที่เป็นข้อต่อจะสามารถขยับได้ทุกส่วน สามารถที่จะ ก้ม,หงาย กล้องได้เพื่อจะสะดวกในการมองเห็น เลข *1, 2, 3* นั้นคือส่วนที่สามารถขยับได้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



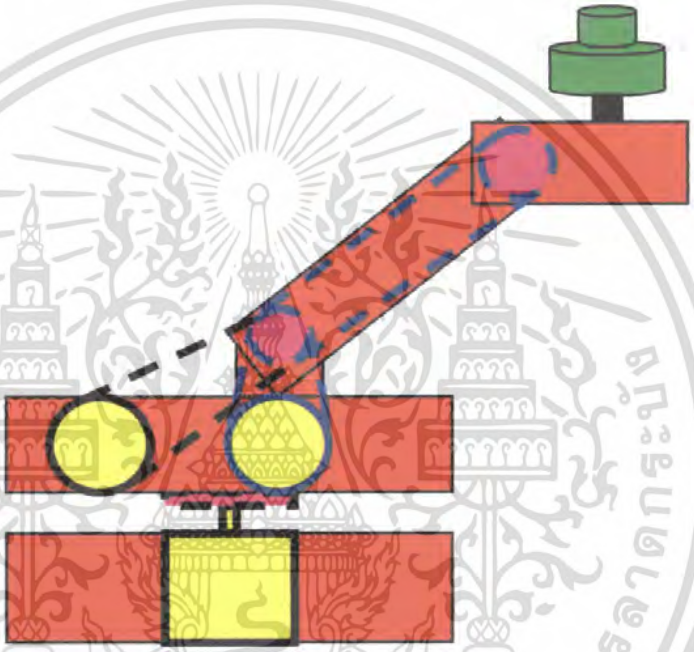

รูปที่ 3.3 แสดงรูปด้านบนและรายละเอียด



รูปที่ 3.4 ล้อที่ใช้เป็นตัวขับเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล้อที่นำมาใช้ทำมาจากท่อประปา (PVC) เนื่องจากน้ำหนักเบา ราคาถูก และทำได้ง่ายกว่าล้อที่เป็นเหล็ก เราทำการติดยึดเฟืองข้างใดข้างหนึ่งเพื่อใช้เป็นเฟืองที่ส่งกำลังมาจากโซ่ที่มาจากตัวขับเคลื่อนหลัก(มอเตอร์)



รูปที่ 3.5 โครงสร้างของแขนที่ใช้ยกกล้องวงจรปิด

3.1.2 ส่วนของการสื่อสารแบบไร้สาย แบ่งเป็น 2 ส่วน คือ

1. การสื่อสารแบบไร้สายที่ใช้ควบคุมหุ่นยนต์
2. การสื่อสารแบบไร้สายที่ส่งภาพจากกล้องวงจรปิดมายังจอแสดงภาพ

**3.1.2.1 การสื่อสารแบบไร้สายที่ใช้ควบคุมหุ่นยนต์**

ที่เราควบคุมหุ่นยนต์กู้ภัยโดยใช้การสื่อสารแบบไร้สาย เพื่อสะดวกในควบคุมสามารถเข้าไปในพื้นที่ที่จำกัดได้ เพื่อความปลอดภัยของผู้ควบคุมเอง เป็นต้น การสื่อสารแบบไร้สายที่เรานำมาใช้ในการควบคุมหุ่นยนต์นี้มีชื่อว่า ETT สามารถพัฒนารูปแบบการเชื่อมต่อได้ 3 รูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.1.2.2 การสื่อสารแบบไร้สายที่ส่งภาพจากกล้องวงจรปิดมายังจอแสดงภาพ**

เป็นส่วนที่ใช้ส่งภาพที่ได้จากกล้อง CCTV มายังตัวส่งและจะนำภาพที่ได้ทำการส่งภาพมายังตัวรับเพื่อแสดงผลออกจอมอนิเตอร์ ซึ่งมีกำลังส่ง 2 วัตต์ สามารถส่งภาพและเสียงได้ และเลือกช่องในการส่งได้ถึง 4 ช่องสัญญาณ เพื่อหลีกเลี่ยงช่องสัญญาณอื่นที่ไม่ต้องการ

| | | | | |
|---|---|---|---|---|
| Channel | 1 | – | 2.414 | GHz |
| Channel | 2 | – | 2.432 | GHz |
| Channel | 3 | – | 2.450 | GHz |
| Channel | 4 | – | 2.468 | GHz |

**รูปที่ 3.6 ส่วนประกอบของตัวรับ-ส่งสัญญาณภาพและเสียง**

**3.1.3 ส่วนของคอมพิวเตอร์**

เป็นส่วนที่สำคัญอีกส่วนหนึ่งคือ ทำหน้าที่ควบคุมส่วนต่างๆ ของหุ่นยนต์ตามที่เราสั่งงานโดยแป้นพิมพ์ จากการเขียนโปรแกรมสั่งงานให้ทำงานในส่วนต่างๆ ได้โดยผ่านการเชื่อมต่อกับการสื่อสารแบบไร้สาย และรับข้อมูลจากส่วนของเซ็นเซอร์เข็มทิศอิเล็กทรอนิกส์ มาเขียนเป็นแผนที่เพื่อระบุถึงตำแหน่งที่หุ่นยนต์อยู่ ทิศทางในการเคลื่อนที่ และตำแหน่งเป้าหมายที่ถูกค้นพบโดยจะแสดงโปรแกรมไว้ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# ผลการทดลอง

## 4.1 ผลการทดลองทางด้านโปรแกรมควบคุมหุ่นยนต์

ในการควบคุมหุ่นยนต์นี้เราจะใช้แป้นพิมพ์เป็นตัวควบคุมการทำงานของหุ่นยนต์ โดยเมื่อมีการกดปุ่มบนแป้นพิมพ์เมื่อไร มันจะทำการส่งข้อมูลในรูปแบบรหัสแอสกี้ ที่เป็นแบบลักษณะตัวอักขระตามค่าที่เรากดไปนั้นเข้าทางพอร์ตอนุกรม (RS-232) ไปยังคอมพิวเตอร์ซึ่งทำหน้าที่รับข้อมูลเพื่อจะได้นำสัญญาณข้อมูลไปควบคุมหุ่นยนต์ และควบคุมโปรแกรมวาดแผนที่

ตารางดังต่อไปนี้เป็นการแสดงให้เห็นว่าเมื่อเรากดปุ่มปุ่มนั้นเมื่อไร ข้อมูลจะถูกส่งออกไปเพื่อทำให้มอเตอร์หรือ Selector นั้นทำงานดังตารางที่ 4.1

**ตารางที่ 4.1** การส่งข้อมูลออกไปและทำการแสดงผลเมื่อมีการกดปุ่มแป้นพิมพ์

| เมื่อกดปุ่มบนแป้นพิมพ์ | ข้อมูลที่ส่งออกจาก MScom1 | การแสดงผลบนโปรแกรมวาดแผนที่ | การแสดงผลของหุ่นยนต์ | หมายเหตุ |
|---|---|---|---|---|
| F | "F" รหัสแอสกี้ | - | - | จะเร่งความเร็วแต่จะไม่มีผลถ้าไม่มีการเคลื่อนไหว |
| W | "W" รหัสแอสกี้ | เส้นสีน้ำเงินเคลื่อนที่ไปด้านหน้า | หุ่นยนต์วิ่งไปข้างหน้า | - |
| S | "S" รหัสแอสกี้ | เส้นสีน้ำเงินเคลื่อนที่ถอยหลัง | หุ่นยนต์วิ่งถอยหลัง | - |
| A | "A" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้าย | เข็มทิศเบี่ยงไปทางซ้าย |
| D | "D" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวา | เข็มทิศเบี่ยงไปทางขวา |
| Y | "Y" รหัสแอสกี้ | - | มอเตอร์ล้อหน้ายกขึ้น | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.1** การส่งข้อมูลออกไปและทำการแสดงผลเมื่อมีการกดปุ่มแป้นพิมพ์

| เมื่อกดปุ่มบนแป้นพิมพ์ | ข้อมูลที่ส่งออกจาก MScom1 | การแสดงผลบนโปรแกรมวาดแผนที่ | การแสดงผลของหุ่นยนต์ | หมายเหตุ |
|---|---|---|---|---|
| H | "H" รหัสแอสกี้ | - | มอเตอร์ล้อหน้ายกลง | - |
| T | "T" รหัสแอสกี้ | - | มอเตอร์ล้อหลังยกขึ้น | - |
| G | "G" รหัสแอสกี้ | - | มอเตอร์ล้อหลังยกลง | - |
| I | "I" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 1 ยกขึ้น | - |
| K | "K" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 1 ยกลง | - |
| J | "J" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 1 หมุนไปทางซ้าย | - |
| L | "L" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 1 หมุนไปทางขวา | - |
| X | "X" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 2 ยกขึ้น | - |
| C | "C" รหัสแอสกี้ | - | มอเตอร์แขนท่อนที่ 2 ยกลง | - |
| V | "V" รหัสแอสกี้ | - | - | No Use |
| B | "B" รหัสแอสกี้ | - | - | No Use |
| N | "N" รหัสแอสกี้ | - | - | No Use |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เมื่อกดปุ่มบนแป้นพิมพ์ | ข้อมูลที่ส่งออกจาก MScom1 | การแสดงผลบนโปรแกรมวาดแผนที่ | การแสดงผลของหุ่นยนต์ | หมายเหตุ |
|---|---|---|---|---|
| M | "M" รหัสแอสกี้ | - | - | No Use |
| Q+M | - | - | - | No Use |
| Q+S | - | - | - | No Use |
| F+W | "a" รหัสแอสกี้ | เส้นสีน้ำเงินเคลื่อนที่ไปด้านหน้าอย่างรวดเร็ว | หุ่นยนต์วิ่งไปข้างหน้าด้วยความเร็วสูง | - |
| F+S | "b" รหัสแอสกี้ | เส้นสีน้ำเงินเคลื่อนที่ไปถอยหลังอย่างรวดเร็ว | หุ่นยนต์วิ่งถอยหลังด้วยความเร็วสูง | - |
| F+A | "c" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้ายอย่างรวดเร็ว | เข็มทิศเบี่ยงไปทางซ้ายอย่างรวดเร็ว |
| F+D | "d" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวาอย่างรวดเร็ว | เข็มทิศเบี่ยงไปทางขวาอย่างรวดเร็ว |
| W+Y | "e" รหัสแอสกี้ | - | หุ่นยนต์วิ่งไปข้างหน้าพร้อมยกล้อหน้าขึ้น | - |
| W+H | "f" รหัสแอสกี้ | - | หุ่นยนต์วิ่งไปข้างหน้าพร้อมยกล้อหน้าลง | - |
| W+T | "g" รหัสแอสกี้ | - | หุ่นยนต์วิ่งไปข้างหน้าพร้อมยกล้อหลังขึ้น | - |
| W+G | "h" รหัสแอสกี้ | - | หุ่นยนต์วิ่งไปข้างหน้าพร้อมยกล้อหลังลง | - |
| S+Y | "i" รหัสแอสกี้ | - | หุ่นยนต์วิ่งถอยหลังพร้อมยกล้อหน้าขึ้น | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เมื่อกดปุ่มบนแป้นพิมพ์ | ข้อมูลที่ส่งออกจาก MScom1 | การแสดงผลบนโปรแกรมวาดแผนที่ | การแสดงผลของหุ่นยนต์ | หมายเหตุ |
|---|---|---|---|---|
| S+H | "j" รหัสแอสกี้ | - | หุ่นยนต์วิ่งถอยหลังพร้อมยกล้อหน้าลง | - |
| S+T | "k" รหัสแอสกี้ | - | หุ่นยนต์วิ่งถอยหลังพร้อมยกล้อหลังขึ้น | - |
| S+G | "l" รหัสแอสกี้ | - | หุ่นยนต์วิ่งถอยหลังพร้อมยกล้อหลังลง | - |
| A+Y | "m" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้ายพร้อมยกล้อหน้าขึ้น | - |
| A+H | "n" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้ายพร้อมยกล้อหน้าลง | - |
| A+T | "o" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้ายพร้อมยกล้อหลังขึ้น | - |
| A+G | "p" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวซ้ายพร้อมยกล้อหลังลง | - |
| D+Y | "q" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวาพร้อมยกล้อหน้าขึ้น | - |
| D+H | "r" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวาพร้อมยกล้อหน้าลง | - |
| D+T | "s" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวาพร้อมยกล้อหลังขึ้น | - |
| D+G | "t" รหัสแอสกี้ | - | หุ่นยนต์เลี้ยวขวาพร้อมยกล้อหลังลง | - |
| Y+T | "u" รหัสแอสกี้ | - | หุ่นยนต์ยกล้อหน้าและล้อหลังขึ้นพร้อมกัน | - |
| Y+G | "v" รหัสแอสกี้ | - | หุ่นยนต์ยกล้อหน้าขึ้นและล้อหลังลงพร้อมกัน | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เมื่อกดปุ่มบนแป้นพิมพ์ | ข้อมูลที่ส่งออกจาก MScom1 | การแสดงผลบนโปรแกรมวาดแผนที่ | การแสดงผลของหุ่นยนต์ | หมายเหตุ |
|---|---|---|---|---|
| H+T | "w" รหัสแอสกี้ | - | หุ่นยนต์ยกล้อหน้าลงและล้อหลังขึ้นพร้อมกัน | - |
| H+G | "x" รหัสแอสกี้ | - | หุ่นยนต์ยกล้อหน้าและล้อหลังลงพร้อมกัน | - |
| 1 | - | วางตำแหน่งผู้ประสบภัยคนที่ 1 | - | - |
| 2 | - | วางตำแหน่งผู้ประสบภัยคนที่ 2 | - | - |
| 3 | - | วางตำแหน่งผู้ประสบภัยคนที่ 3 | - | - |
| 4 | - | วางตำแหน่งผู้ประสบภัยคนที่ 4 | - | - |
| 5 | - | วางตำแหน่งผู้ประสบภัยคนที่ 5 | - | - |
| 6 | - | วางตำแหน่งผู้ประสบภัยคนที่ 6 | - | - |
| 7 | - | วางตำแหน่งผู้ประสบภัยคนที่ 7 | - | - |
| 8 | - | วางตำแหน่งผู้ประสบภัยคนที่ 8 | - | - |
| 9 | - | เลื่อนเข็มทิศไป +1 องศา | - | - |
| 0 | - | เลื่อนเข็มทิศไป -1 องศา | - | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2 การทดลองคุณสมบัติของอุปกรณ์เข็มทิศอิเล็กทรอนิกส์**

จากผลการทดลองสรุปได้ว่าเข็มทิศอิเล็กทรอนิกส์นั้นสามารถที่จะบอกตำแหน่งของหุ่นยนต์ที่กำลังเคลื่อนที่ไปอยู่ว่าหุ่นยนต์นั้นกำลังเคลื่อนที่ไปยังทิศใดของพื้นที่นั้น ซึ่งเข็มทิศอิเล็กทรอนิกส์นั้นในวงจรถูกตั้งโปรแกรมไว้อยู่แล้วเพื่อให้สะดวกในการใช้งานไม่ต้องเสียเวลาออกแบบโปรแกรมตั้งค่าเอง

**4.3 คุณสมบัติโดยรวมของหุ่นยนต์กู้ภัย**

จากการทำงานของหุ่นยนต์โดยภาพรวมสามารถสรุปคุณสมบัติต่างๆภายในหุ่นยนต์ได้ดังนี้

**ตารางที่ 4.2** คุณสมบัติโดยรวมของหุ่นยนต์กู้ภัย

| คุณสมบัติโดยรวม | ค่าที่ได้จากการทดลอง | ปัญหาที่เกิดขึ้น |
|---|---|---|
| ความเร็วในการเดินหน้าของหุ่นยนต์กู้ภัย | 0.1 m/s | - |
| ความเร็วในการถอยหลังของหุ่นยนต์กู้ภัย | 0.1 m/s | - |
| ความเร็วในการเลี้ยวซ้ายของหุ่นยนต์กู้ภัย | 0.1 m/s | - |
| ความเร็วในการเลี้ยวขวาของหุ่นยนต์กู้ภัย | 0.1 m/s | - |
| ความเร็วในการหมุนแขนกล้องทางซ้ายของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการหมุนแขนกล้องทางขวาของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกแขนกล้องท่อนล่างขึ้นของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกแขนกล้องท่อนล่างลงของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกแขนกล้อง | 0.05 m/s | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

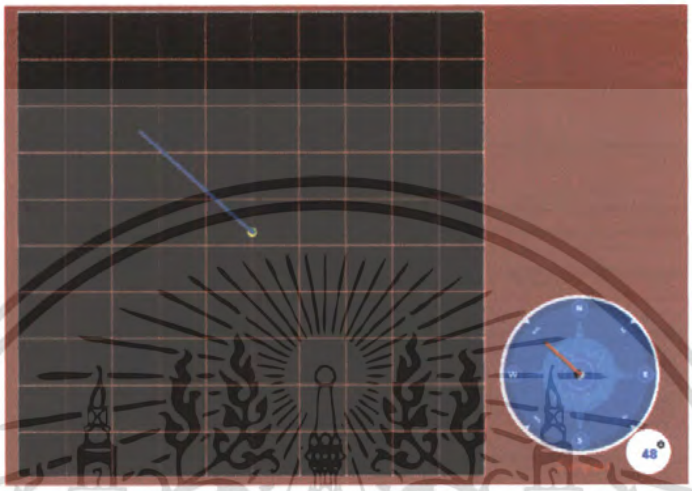
| | | |
|---|---|---|
| ท่อนบนขึ้นของหุ่นยนต์กู้ภัย | | |
| ความเร็วในการยกแขนกล้องท่อนบนลงของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกล้อหน้าขึ้นของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกล้อหน้าลงของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ความเร็วในการยกล้อหลังขึ้นของหุ่นยนต์กู้ภัย | 0.05 m/s | - |

ตารางที่ 4.2 คุณสมบัติโดยรวมของหุ่นยนต์กู้ภัย ( ต่อ)

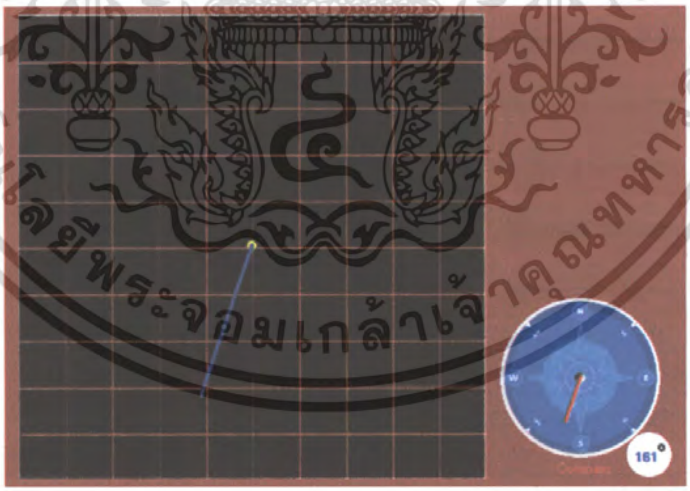
| คุณสมบัติโดยรวม | ค่าที่ได้จากการทดลอง | ปัญหาที่เกิดขึ้น |
|---|---|---|
| ความเร็วในการยกล้อหลังลงของหุ่นยนต์กู้ภัย | 0.05 m/s | - |
| ระยะทางที่ตัวส่งสามารถส่งสัญญาณข้อมูลไปยังตัวรับได้ | 100 m | - |
| ระยะทางที่กล้องจากหุ่นยนต์สามารถส่งภาพเข้ามาได้ | 100 m | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 รูปผลการทดลองโปรแกรมวาดแผนที่ด้วย Visual Basic 6.0




**รูปที่ 4.1 ผลการทดลองโปรแกรมวาดแผนที่จากเข็มทิศเมื่อหุ่นขยับไป 48 องศา**




**รูปที่ 4.2 ผลการทดลองโปรแกรมวาดแผนที่จากเข็มทิศเมื่อหุ่นขยับไป 161 องศา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# สรุปและวิจารณ์

การทำหุ่นยนต์กู้ภัย ได้มีเป้าหมายเพื่อป้องกันไม่ให้มนุษย์เกิดอันตรายจากการเข้าไปกู้ภัยในที่เกิดเหตุที่คาดเดาอะไรไม่ได้ จึงได้มีการสร้างหุ่นยนต์กู้ภัยขึ้นโดยหน้าที่ที่หุ่นยนต์กู้ภัยที่ได้ทำการสร้างขึ้นมานี้ โดยหุ่นยนต์กู้ภัยตัวนี้ มีความสามารถคือ

- หุ่นยนต์กู้ภัยนี้ สามารถเคลื่อนที่ไปในสภาพพื้นผิว ที่ขรุขระ และพื้นเอียง และได้มีพัฒนาเพื่อที่จะให้เคลื่อนที่ขึ้นบันไดได้

- หุ่นยนต์กู้ภัยเป็นการบังคับแบบไร้สาย และมองผ่านกล้องไร้สาย จึงใช้หุ่นยนต์เดินเข้าไปเพื่อตรวจสอบสภาพภายในพื้นที่เกิดเหตุได้ โดยไม่ต้องเสี่ยง

- หุ่นยนต์กู้ภัย สามารถวาดแผนที่เพื่อที่จะทราบสภาวะภายในพื้นที่เกิดเหตุ เพื่อที่ว่าเมื่อสภาพพื้นที่ภายในมืดมองไม่ค่อยชัด อาจจะบอกได้ว่ามีอะไรอยู่ที่ด้านข้าง ๆ จะได้ระมัดระวัง และในหุ่นยนต์กู้ภัย มีตัวแสดงทิศทางเพื่อบอกให้ทราบว่าเดินไปในทิศทางใน และมีสิ่งกีดขวางอยู่หรือเปล่าก็จะช่วยในการตัดสินใจได้

- เมื่อเจอผู้ประสบภัยในพื้นที่เกิดเหตุ เราสามารถตรวจสอบอุณหภูมิในร่างกายได้ เพื่อที่จะตรวจสอบสภาวะของผู้ประสบภัย ว่าอยู่ในขั้นใด ว่ายังมีชีวิตหรือว่าเสียชีวิตไปแล้ว เพราะอุณหภูมิจะต่างกัน

วิจารณ์ หุ่นยนต์กู้ภัยตัวนี้เป็นตัวที่สร้างขึ้นเองโดยนักศึกษา โดยโครงสร้างอาจไม่ได้สัดส่วนที่แน่นอน เพราะสร้างตามสภาพเครื่องมือที่มีและความสามารถของตัวเองทุกชิ้นส่วน ในหุ่นยนต์กู้ภัย นี้การทำงานส่วนใหญ่ ใช้ ไมโครคอนโทรลเลอร์ควบคุม เป็นปัญหาในการควบคุมบางครั้งต้องรีเซ็ตค่าใหม่บ่อย ๆ และการทำเป็นการศึกษาขั้นต้นสามารถนำไปเป็นต้นแบบในการพัฒนาต่อได้เพื่อที่จะได้สมบูรณ์แบบสามารถทำงานได้เต็มที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

1) เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ฉบับ AT89C5x/AT89Sxxxx , วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , อินโนเวตีฟ เอ็กเพอริเมนต์ , 399 หน้า , 2548.

2) คู่มือเรียน Visual Basic 6 [ปรับปรุงใหม่] , ฉัททวุฒิ พืชผล พิชิต สันติกุลานนท์ และ พร้อมเลิศ หล่อวิจิตร , โปรวิชั่น , 560 หน้า , 2547.

3) เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic , อภิชาติ ภู่พลับ , อินโฟเพรส , 226 หน้า , 2546.

4) ไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ P89V51RD2 , นคร ภักดีชาติ และ ชัยวัฒน์ ลิ้มพรจิตรวิไล , อินโนเวตีฟ เอ็กเพอริเมนต์ , 284 หน้า , 2549.

5) PSoC การเขียนโปรแกรมไมโครคอนโทรลเลอร์ PSoC ด้วยภาษาซี , อุกฤษฎ์ ตันทสุทธานนท์ และ ทีมงาน , MRT , 220 หน้า , 2549.

6) เรียนรู้และเข้าใจ PSoC Microcontroller ด้วย ภาษา Assembly และ ภาษา C , นายวัชรินทร์ เดารพ , บริษัท อีทีที จำกัด , 340 หน้า , 2548.

7) MAXIM Datasheet

8) ST Datasheet

9) Philips Datasheet

10) Fairchild Semiconductor Datasheet

11) Motorola Datasheet

12) National Semiconductor Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โปรแกรมที่ 2.1 แสดงการอ่านค่าสัญญาณพัลส์จากโมดูล CMPS03**

```
'{$STAMP BS2sx}
'{$PBASIC 2.5}
bearing  VAR  WORD
main:
     PULSIN 4, 1, bearing              ' Get reading
     bearing = (bearing-1250)/125      ' BS2sx  - Calculate Bearing
                                       ' in degrees
     DEBUG "Compass Bearing  ", DEC3 bearing ,CR
                                       ' Display Compass Bearing
     GOTO main
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โปรแกรมที่ 2.2 โปรแกรมอ่านค่าจากโมดูล CMPS03 ผ่านระบบบัส I²C ของ I - Stamp**

```
'{$STAMP BS2ex}
'{$PBASIC 2.5}
SDA             CON     6           ' I2C serial data line
SCL             CON     7           ' I2C serial clock line
WrCMPS03        CON     $C0         ' write to compass
RdCMPS03        CON     $C1         ' read from compass
Ack             CON     0           ' acknowledge bit
Nak             CON     1           ' no ack bit
i2cSDA          VAR     NIB         ' I2C serial data pin
i2cData         VAR     WORD        ' data to/from device
REGISTER        VAR     BYTE        ' register address
i2cWork         VAR     BYTE        ' work byte for TX routine
i2cAck          VAR     BIT         ' Ack bit from device
temp            VAR     WORD        ' for rj printing
digits          VAR     NIB
width           VAR     NIB

Init:
   PAUSE 250
   i2cSDA = SDA                                 ' define SDA pin
   REGISTER = 0                                 ' compass revision number
   GOSUB Read_Byte
   DEBUG 2,1,1, "Revision Number = " ,DEC2 i2cData
Main:
   REGISTER = 1                                 ' Show Data 0-255 for 0-360 degree
   GOSUB Read_Byte                              ' Read Byte From I2C
   DEBUG 2,1,3, "The Coarse Data(0-255) = " ,DEC i2cData
   REGISTER = 2                         ' get Data in degrees, 0.0 - 359.9 Degree
   GOSUB Read_Word
   DEBUG 2,1,5, "Position = ",DEC i2cData/10,".",DEC1 i2cData," Degree"
   PAUSE 250
   GOTO Main

' Compass Access Subroutines
' Writes low byte of i2cData to REGISTER
Write_Byte:
   GOSUB I2C_Start
   i2cWork = WrCMPS03
   GOSUB I2C_TX_Byte                            ' send device address
   i2cWork = REGISTER
   GOSUB I2C_TX_Byte                            ' send register number
   i2cWork = i2cData.LOWBYTE
   GOSUB I2C_TX_Byte                            ' send the data
   GOSUB I2C_Stop
   RETURN

' Writes i2cData to REGISTER
Write_Word:
   GOSUB I2C_Start
   i2cWork = WrCMPS03
   GOSUB I2C_TX_Byte                          ' send device address
   i2cWork = REGISTER
   GOSUB I2C_TX_Byte                          ' send register number
   i2cWork = i2cData.HIGHBYTE
   GOSUB I2C_TX_Byte                          ' send the data - high byte
   i2cWork = i2cData.LOWBYTE
   GOSUB I2C_TX_Byte                          ' send the data - low byte
   GOSUB I2C_Stop
   RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

```
' Read i2cData (8 bits) from REGISTER
Read_Byte: GOSUB I2C_Start
           i2cWork = WrCMPS03
           GOSUB I2C_TX_Byte                    ' send compass address
           i2cWork = REGISTER
           GOSUB I2C_TX_Byte                    ' send register number
           GOSUB I2C_Start                      ' repeat start (sets register)
           i2cWork = RdCMPS03
           GOSUB I2C_TX_Byte                    ' send read command
           GOSUB I2C_RX_Byte_Nak
           GOSUB I2C_Stop
           i2cData = i2cWork                    ' return the data
           RETURN

' Read i2cData (16 bits) from REGISTER
Read_Word: GOSUB I2C_Start
           i2cWork = WrCMPS03
           GOSUB I2C_TX_Byte                    ' send compass address
           i2cWork = REGISTER
           GOSUB I2C_TX_Byte                    ' send register number
           GOSUB I2C_Start                      ' repeat start (sets register)
           i2cWork = RdCMPS03
           GOSUB I2C_TX_Byte                    ' send read command
           GOSUB I2C_RX_Byte
           i2cData.HIGHBYTE = i2cWork           ' read high byte of data
           GOSUB I2C_RX_Byte_Nak
           GOSUB I2C_Stop
           i2cData.LOWBYTE = i2cWork            ' read low byte of data
           RETURN
' Low Level I2C Subroutines
'Start
I2C_Start: INPUT i2cSDA                         ' I2C start bit sequence
           INPUT SCL
           LOW i2cSDA                           ' SDA -> low while SCL high
Clock_Hold:
   IF (INS.LOWBIT(SCL) = 0) THEN Clock_Hold     ' device ready?
   RETURN

'Transmit
I2C_TX_Byte:
   SHIFTOUT i2cSDA,SCL,MSBFIRST,[i2cWork\8]     ' send byte to device
   SHIFTIN i2cSDA,SCL,MSBPRE,[i2cAck\1]         ' get acknowledge bit
   RETURN

'Receive
I2C_RX_Byte_Nak: i2cAck = Nak                   ' no Ack = high
                 GOTO I2C_RX
I2C_RX_Byte:     i2cAck = Ack                   ' Ack = low
I2C_RX:
   SHIFTIN i2cSDA,SCL,MSBPRE,[i2cWork\8]        ' get byte from device
   SHIFTOUT i2cSDA,SCL,LSBFIRST,[i2cAck\1]      ' send ack or nak
   RETURN

'Stop
I2C_Stop:  LOW i2cSDA                           ' I2C stop bit sequence
           INPUT SCL
           INPUT i2cSDA                         ' SDA --> high while SCL high
           RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างคำสั่งโปรแกรมควบคุม ROBOT

```
Dim a As Currency
Dim b As Currency
Dim compass As Integer
Dim ang As Currency
Dim cal As Integer
Dim px As Currency
Dim py As Currency
Dim xx As Integer
Dim yy As Integer
Dim turn As Currency
Dim boo As Boolean

Const speed1 = 4
Const speed2 = 8
Const midX = 12390
Const midY = 8780
Const PI = 3.1459

Private Sub Form_load()
MSComm1.PortOpen = True
MSComm2.PortOpen = True
Image2.Left = -1000
Image2.Top = -1000
Image3.Left = -1000
Image3.Top = -1000
Image4.Left = -1000
Image4.Top = -1000
Image5.Left = -1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Image5.Top = -1000
Image6.Left = -1000
Image6.Top = -1000
Image7.Left = -1000
Image7.Top = -1000
Image8.Left = -1000
Image8.Top = -1000
Image9.Left = -1000
Image9.Top = -1000
cal = 90
End Sub

Private Sub Label1_Click()
Picture1.Cls

End Sub

Private Sub Label4_Click()
End
End Sub

Private Sub Picture1_DragDrop(Source As Control, X As Single, Y As Single)

Shape1.Left = X - (Shape1.Width / 2)
Shape1.Top = Y - (Shape1.Height / 2)
a = Shape1.Left + (Shape1.Width / 2)

b = Shape1.Top + (Shape1.Height / 2)

px = (a / 1000)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
py = 10 - (b / 996.5)

Label3.Caption = px
Label5.Caption = py
End Sub
Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Shape1.Drag vbBeginDrag

End Sub

Private Sub Picture1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Shape1.Drag vbEndDrag

End Sub

Private Sub Picture1_KeyDown(KeyCode As Integer, Shift As Integer)

px = (a / 1000)
py = 10 - (b / 1000)

Label3.Caption = px
Label5.Caption = py


Select Case KeyCode

  Case vbKeyF:    MSComm1.Output = "F"

  Case vbKeyW:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSComm1.Output = "W"

a = a - (speed1 * Sin(turn))
b = b - (speed1 * Cos(turn))
If boo Then
Picture1.PSet (a, b), vbBlue
Else

Picture1.PSet (a, b), vbBlue

Picture1.DrawWidth = 5
End If

Case vbKeyS:   MSComm1.Output = "S"

a = a + -(speed1 * Sin(-turn))
b = b + (speed1 * Cos(-turn))
If boo Then
Picture1.PSet (a, b), vbBlue
Else

Picture1.PSet (a, b), vbBlue
Picture1.DrawWidth = 5

End If

Case vbKeyA:   MSComm1.Output = "A"

Case vbKeyD:   MSComm1.Output = "D"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case vbKeyY:    MSComm1.Output = "Y"

Case vbKeyH:    MSComm1.Output = "H"

Case vbKeyT:    MSComm1.Output = "T"

Case vbKeyG:    MSComm1.Output = "G"

Case vbKeyI:    MSComm1.Output = "I"

Case vbKeyK:    MSComm1.Output = "K"

Case vbKeyJ:    MSComm1.Output = "J"

Case vbKeyL:    MSComm1.Output = "L"

Case vbKeyX:    MSComm1.Output = "X"

Case vbKeyC:    MSComm1.Output = "C"

Case vbKeyV:    MSComm1.Output = "V"

Case vbKeyB:    MSComm1.Output = "B"

Case vbKeyN:    MSComm1.Output = "N"

Case vbKeyM:    MSComm1.Output = "M"

Case (vbKeyQ & vbKeyW):
        a = a - (6 * Sin(turn))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            b = b - (6 * Cos(turn))
            If boo Then
            Picture1.PSet (a, b), vbBlue
            Else

            Picture1.PSet (a, b), vbBlue

            Picture1.DrawWidth = 5

            End If


Case (vbKeyQ & vbKeyS):
            a = a + -(6 * Sin(-turn))
            b = b + (6 * Cos(-turn))
            If boo Then
            Picture1.PSet (a, b), vbBlue
            Else

            Picture1.PSet (a, b), vbBlue
            Picture1.DrawWidth = 5
            End If
Case (vbKeyF & vbKeyW): MSComm1.Output = "a"
            a = a - (speed2 * Sin(turn))
            b = b - (speed2 * Cos(turn))
            If boo Then
            Picture1.PSet (a, b), vbBlue
            Else

            Picture1.PSet (a, b), vbBlue
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Picture1.DrawWidth = 5

            End If

Case (vbKeyF & vbKeyS): MSComm1.Output = "b"
            a = a + -(speed2 * Sin(-turn))
            b = b + (speed2 * Cos(-turn))
            If boo Then
            Picture1.PSet (a, b), vbBlue
            Else

            Picture1.PSet (a, b), vbBlue
            Picture1.DrawWidth = 5

            End If
Case (vbKeyF & vbKeyA): MSComm1.Output = "c"

Case (vbKeyF & vbKeyD): MSComm1.Output = "d"

Case (vbKeyW & vbKeyY): MSComm1.Output = "e"

Case (vbKeyW & vbKeyH): MSComm1.Output = "f"

Case (vbKeyW & vbKeyT): MSComm1.Output = "g"

Case (vbKeyS & vbKeyY): MSComm1.Output = "i"
Case (vbKeyS & vbKeyH): MSComm1.Output = "j"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case (vbKeyS & vbKeyT): MSComm1.Output = "k"

Case (vbKeyS & vbKeyG): MSComm1.Output = "l"

Case (vbKeyA & vbKeyY): MSComm1.Output = "m"

Case (vbKeyA & vbKeyH): MSComm1.Output = "n"

Case (vbKeyA & vbKeyT): MSComm1.Output = "o"

Case (vbKeyA & vbKeyG): MSComm1.Output = "p"

Case (vbKeyD & vbKeyY): MSComm1.Output = "q"

Case (vbKeyD & vbKeyH): MSComm1.Output = "r"

Case (vbKeyD & vbKeyT): MSComm1.Output = "s"

Case (vbKeyD & vbKeyG): MSComm1.Output = "t"

Case (vbKeyY & vbKeyT): MSComm1.Output = "u"

Case (vbKeyY & vbKeyG): MSComm1.Output = "v"

Case (vbKeyH & vbKeyT): MSComm1.Output = "w"

Case (vbKeyH & vbKeyG): MSComm1.Output = "x"

Case vbKey1:    Label10(0).ForeColor = &HFFFFFF
                xx = (a + 500) / 1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Label11(0).Caption = xx
            yy = (10500 - b) / 1000
            Label12(0).Caption = yy

            Image2.Left = ((Label11(0).Caption * 1000) - 500) - (Image2.Width / 2)

            Image2.Top = (9900 - (Label12(0).Caption * 1000)) + (Image2.Height / 2)

Case vbKey2:    Label10(1).ForeColor = &HFFFFFF
            xx = (a + 500) / 1000
            Label11(1).Caption = xx
            yy = (10500 - b) / 1000
            Label12(1).Caption = yy

            Image3.Left = ((Label11(1).Caption * 1000) - 500) - (Image3.Width / 2)

            Image3.Top = (9900 - (Label12(1).Caption * 1000)) + (Image3.Height / 2)

Case vbKey3:    Label10(2).ForeColor = &HFFFFFF
            xx = (a + 500) / 1000
            Label11(2).Caption = xx
            yy = (10500 - b) / 1000
            Label12(2).Caption = yy

            Image4.Left = ((Label11(2).Caption * 1000) - 500) - (Image4.Width / 2)

            Image4.Top = (9900 - (Label12(2).Caption * 1000)) + (Image4.Height / 2)

Case vbKey4:    Label10(3).ForeColor = &HFFFFFF
            xx = (a + 500) / 1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Label11(3).Caption = xx
        yy = (10500 - b) / 1000
        Label12(3).Caption = yy

        Image5.Left = ((Label11(3).Caption * 1000) - 500) - (Image5.Width / 2)

        Image5.Top = (9900 - (Label12(3).Caption * 1000)) + (Image5.Height / 2)

Case vbKey5:    Label10(4).ForeColor = &HFFFFFF
        xx = (a + 500) / 1000
        Label11(4).Caption = xx
        yy = (10500 - b) / 1000
        Label12(4).Caption = yy

        Image6.Left = ((Label11(4).Caption * 1000) - 500) - (Image6.Width / 2)

        Image6.Top = (9900 - (Label12(4).Caption * 1000)) + (Image6.Height / 2)

Case vbKey6:    Label10(5).ForeColor = &HFFFFFF
        xx = (a + 500) / 1000
        Label11(5).Caption = xx
        yy = (10500 - b) / 1000
        Label12(5).Caption = yy

        Image7.Left = ((Label11(5).Caption * 1000) - 500) - (Image7.Width / 2)

        Image7.Top = (9900 - (Label12(5).Caption * 1000)) + (Image7.Height / 2)

Case vbKey7:    Label10(6).ForeColor = &HFFFFFF
        xx = (a + 500) / 1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Label11(6).Caption = xx
            yy = (10500 - b) / 1000
            Label12(6).Caption = yy

            Image8.Left = ((Label11(6).Caption * 1000) - 500) - (Image8.Width / 2)

            Image8.Top = (9900 - (Label12(6).Caption * 1000)) + (Image8.Height / 2)

    Case vbKey8:    Label10(7).ForeColor = &HFFFFFF
            xx = (a + 500) / 1000
            Label11(7).Caption = xx
            yy = (10500 - b) / 1000
            Label12(7).Caption = yy

            Image9.Left = ((Label11(7).Caption * 1000) - 500) - (Image9.Width / 2)

            Image9.Top = (9900 - (Label12(7).Caption * 1000)) + (Image9.Height / 2)

    Case vbKey0:    cal = cal + 1

    Case vbKey9:    cal = cal - 1

End Select

End Sub
Private Sub Picture1_keyup(KeyCode As Integer, Shift As Integer)
Select Case KeyCode

    Case vbKeyF:    MSComm1.Output = "Z"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case vbKeyW:    MSComm1.Output = "Z"

Case vbKeyS:    MSComm1.Output = "Z"

Case vbKeyA:    MSComm1.Output = "Z"

Case vbKeyD:    MSComm1.Output = "Z"

Case vbKeyY:    MSComm1.Output = "Z"

Case vbKeyH:    MSComm1.Output = "Z"

Case vbKeyT:    MSComm1.Output = "Z"

Case vbKeyG:    MSComm1.Output = "Z"

Case vbKeyI:    MSComm1.Output = "Z"

Case vbKeyK:    MSComm1.Output = "Z"

Case vbKeyJ:    MSComm1.Output = "Z"

Case vbKeyL:    MSComm1.Output = "Z"

Case vbKeyX:    MSComm1.Output = "Z"

Case vbKeyC:    MSComm1.Output = "Z"

Case vbKeyV:    MSComm1.Output = "Z"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case vbKeyB:    MSComm1.Output = "Z"

Case vbKeyN:    MSComm1.Output = "Z"

Case vbKeyM:    MSComm1.Output = "Z"

Case (vbKeyQ & vbKeyW): MSComm1.Output = "Z"

Case (vbKeyQ & vbKeyS): MSComm1.Output = "Z"

Case (vbKeyQ & vbKeyA): MSComm1.Output = "Z"

Case (vbKeyQ & vbKeyD): MSComm1.Output = "Z"

Case (vbKeyF & vbKeyW): MSComm1.Output = "Z"

Case (vbKeyF & vbKeyS): MSComm1.Output = "Z"

Case (vbKeyF & vbKeyA): MSComm1.Output = "Z"

Case (vbKeyF & vbKeyD): MSComm1.Output = "Z"

Case (vbKeyW & vbKeyY): MSComm1.Output = "Z"

Case (vbKeyW & vbKeyH): MSComm1.Output = "Z"

Case (vbKeyW & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyS & vbKeyY): MSComm1.Output = "Z"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case (vbKeyS & vbKeyH): MSComm1.Output = "Z"

Case (vbKeyS & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyS & vbKeyG): MSComm1.Output = "Z"

Case (vbKeyA & vbKeyY): MSComm1.Output = "Z"

Case (vbKeyA & vbKeyH): MSComm1.Output = "Z"

Case (vbKeyA & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyA & vbKeyG): MSComm1.Output = "Z"

Case (vbKeyD & vbKeyY): MSComm1.Output = "Z"

Case (vbKeyD & vbKeyH): MSComm1.Output = "Z"

Case (vbKeyD & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyD & vbKeyG): MSComm1.Output = "Z"

Case (vbKeyY & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyY & vbKeyG): MSComm1.Output = "Z"

Case (vbKeyH & vbKeyT): MSComm1.Output = "Z"

Case (vbKeyH & vbKeyG): MSComm1.Output = "Z"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Select
End Sub

Private Sub Timer1_Timer()
MSComm2.DTREnable = True
MSComm2.DTREnable = False
If MSComm2.InBufferCount Then
ang = Asc(MSComm2.Input) * 1.42
compass = ang + 90 - cal
turn = (360 - compass) * 0.01746

        Label6.Caption = compass

      Line23.X1 = midX
      Line23.Y1 = midY

      Line23.X2 = 980 * Cos(PI * (ang - cal + 1) / 180) + midX
      Line23.Y2 = 980 * Sin(PI * (ang - cal + 1) / 180) + midY

End If
End Sub
```

โค้ดเบิร์นpicควบคุมเข็มทิศ

```
        LIST P=16F877
        #INCLUDE <P16F877.INC>
;       CONFIG1 = _CP_OFF&_WDT_OFF&_BODEN_ON&_PWRTE_ON
;       CONFIG2 = _XT_OSC&_WRT_ENABLE_ON&_LVP_OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        __CONFIG 0X0D41

CBLOCK      0X20
NUM1
ENDC

#DEFINE     MOTOR_Fw            PORTB,7
#DEFINE     MOTOR_Bw            PORTB,6
#DEFINE     MOTOR_Lw            PORTB,5
#DEFINE     MOTOR_Rw            PORTB,4
#DEFINE     Leray_Sp            PORTB,3
#DEFINE     MOTOR_UpF           PORTB,2
#DEFINE     MOTOR_DownF         PORTB,1
#DEFINE     MOTOR_UpB       PORTB,0
#DEFINE     MOTOR_DB            PORTD,7
#DEFINE     MOTOR_K1up      PORTD,6
#DEFINE     MOTOR_K1down        PORTD,5
#DEFINE     MOTOR_K2L           PORTD,4
#DEFINE     MOTOR_K2R       PORTD,3
#DEFINE     MOTOR_K3up      PORTD,2
#DEFINE     MOTOR_K3down        PORTE,0
#DEFINE     MOTOR_K4up      PORTE,1
#DEFINE     MOTOR_K4down        PORTE,2
#DEFINE     MOTOR_K5up      PORTD,0
#DEFINE     MOTOR_K5down        PORTD,1

        ORG     0X000

;       MOVLW       .7
;       MOVWF       CMCON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BSF    STATUS,RP0
        MOVLW      0XF0
        MOVWF      TRISC
        CLRF   TRISB
        CLRF   TRISD
        CLRF   TRISE
        MOVLW      .7
        MOVWF      ADCON1
        BCF    STATUS,RP0

        CLRF   PORTC
        CLRF   PORTB
        CLRF   PORTD
        CLRF   PORTE
        CALL   INIT232

MAIN
        MOVF RCREG,W
        MOVWF      NUM1

N0      MOVF NUM1,W
        SUBLW      'F'
        BTFSS STATUS,Z
        GOTO N1
        BSF    Leray_Sp
        GOTO MAIN

N1      MOVF NUM1,W
        SUBLW      'W'
        BTFSS STATUS,Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GOTO  N2
        BSF   MOTOR_Fw
        GOTO  MAIN

N2      MOVF  NUM1,W
        SUBLW       'S'
        BTFSS STATUS,Z
        GOTO  N3
        BSF   MOTOR_Bw
        GOTO  MAIN

N3      MOVF  NUM1,W
        SUBLW       'A'
        BTFSS STATUS,Z
        GOTO  N4
        BSF   MOTOR_Lw
        GOTO  MAIN

N4      MOVF  NUM1,W
        SUBLW       'D'
        BTFSS STATUS,Z
        GOTO  N5
        BSF   MOTOR_Rw
        GOTO  MAIN

N5      MOVF  NUM1,W
        SUBLW       'Y'
        BTFSS STATUS,Z
        GOTO  N6
        BSF   MOTOR_UpF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GOTO  MAIN

N6      MOVF  NUM1,W
        SUBLW       'H'
        BTFSS STATUS,Z
        GOTO  N7
        BSF   MOTOR_DownF
        GOTO  MAIN

N7      MOVF  NUM1,W
        SUBLW       'T'
        BTFSS STATUS,Z
        GOTO  N8
        BSF   MOTOR_UpB
        GOTO  MAIN

N8      MOVF  NUM1,W
        SUBLW       'G'
        BTFSS STATUS,Z
        GOTO  N9
        BSF   MOTOR_DB
        GOTO  MAIN

N9      MOVF  NUM1,W
        SUBLW       'T'
        BTFSS STATUS,Z
        GOTO  N10
        BSF   MOTOR_K1up
        GOTO  MAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
N10   MOVF NUM1,W
      SUBLW       'K'
      BTFSS STATUS,Z
      GOTO N11
      BSF   MOTOR_K1down
      GOTO MAIN

N11   MOVF NUM1,W
      SUBLW       'J'
      BTFSS STATUS,Z
      GOTO N12
      BSF   MOTOR_K2L
      GOTO MAIN

N12   MOVF NUM1,W
      SUBLW       'L'
      BTFSS STATUS,Z
      GOTO N13
      BSF   MOTOR_K2R
      GOTO MAIN

N13   MOVF NUM1,W
      SUBLW       'X'
      BTFSS STATUS,Z
      GOTO N14
      BSF   MOTOR_K3up
      GOTO MAIN

N14   MOVF NUM1,W
      SUBLW       'C'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BTFSS STATUS,Z
        GOTO N15
        BSF   MOTOR_K3down
        GOTO MAIN

N15     MOVF NUM1,W
        SUBLW       'V'
        BTFSS STATUS,Z
        GOTO N16
        BSF   MOTOR_K4up
        GOTO MAIN

N16     MOVF NUM1,W
        SUBLW       'B'
        BTFSS STATUS,Z
        GOTO N17
        BSF   MOTOR_K4down
        GOTO MAIN

N17     MOVF NUM1,W
        SUBLW       'N'
        BTFSS STATUS,Z
        GOTO N18
        BSF   MOTOR_K5up
        GOTO MAIN

N18     MOVF NUM1,W
        SUBLW       'M'
        BTFSS STATUS,Z
        GOTO N19
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BSF     MOTOR_K5down
        GOTO    MAIN

N19     MOVF    NUM1,W
        SUBLW           'a'
        BTFSS   STATUS,Z
        GOTO    N20
        BSF     Leray_Sp
        BSF     MOTOR_Fw
        GOTO    MAIN

N20     MOVF    NUM1,W
        SUBLW           'b'
        BTFSS   STATUS,Z
        GOTO    N21
        BSF     Leray_Sp
        BSF     MOTOR_Bw
        GOTO    MAIN

N21     MOVF    NUM1,W
        SUBLW           'c'
        BTFSS   STATUS,Z
        GOTO    N22
        BSF     Leray_Sp
        BSF     MOTOR_Lw
        GOTO    MAIN

N22     MOVF    NUM1,W
        SUBLW           'd'
        BTFSS   STATUS,Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GOTO  N23
        BSF   Leray_Sp
        BSF   MOTOR_Rw
        GOTO  MAIN

N23     MOVF  NUM1,W
        SUBLW       'e'
        BTFSS STATUS,Z
        GOTO  N24
        BSF   MOTOR_Fw
        BSF   MOTOR_UpF
        GOTO  MAIN

N24     MOVF  NUM1,W
        SUBLW       'f'
        BTFSS STATUS,Z
        GOTO  N25
        BSF   MOTOR_Fw
        BSF   MOTOR_DownF
        GOTO  MAIN

N25     MOVF  NUM1,W
        SUBLW       'g'
        BTFSS STATUS,Z
        GOTO  N26
        BSF   MOTOR_Fw
        BSF   MOTOR_UpB
        GOTO  MAIN

N26     MOVF  NUM1,W
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SUBLW       'h'
        BTFSS STATUS,Z
        GOTO N27
        BSF   MOTOR_Fw
        BSF   MOTOR_DB
        GOTO MAIN

N27     MOVF NUM1,W
        SUBLW       'i'
        BTFSS STATUS,Z
        GOTO N28
        BSF   MOTOR_Bw
        BSF   MOTOR_UpF
        GOTO MAIN

N28     MOVF NUM1,W
        SUBLW       'j'
        BTFSS STATUS,Z
        GOTO N29
        BSF   MOTOR_Bw
        BSF   MOTOR_DownF
        GOTO MAIN

N29     MOVF NUM1,W
        SUBLW       'k'
        BTFSS STATUS,Z
        GOTO N30
        BSF   MOTOR_Bw
        BSF   MOTOR_UpB
        GOTO MAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
N30   MOVF NUM1,W
      SUBLW       'l'
      BTFSS STATUS,Z
      GOTO N31
      BSF   MOTOR_Bw
      BSF   MOTOR_DB
      GOTO MAIN

N31   MOVF NUM1,W
      SUBLW       'm'
      BTFSS STATUS,Z
      GOTO N32
      BSF   MOTOR_Lw
      BSF   MOTOR_UpF
      GOTO MAIN

N32   MOVF NUM1,W
      SUBLW       'n'
      BTFSS STATUS,Z
      GOTO N33
      BSF   MOTOR_Lw
      BSF   MOTOR_DownF
      GOTO MAIN

N33   MOVF NUM1,W
      SUBLW       'o'
      BTFSS STATUS,Z
      GOTO N34
      BSF   MOTOR_Lw
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BSF     MOTOR_UpB
        GOTO    MAIN

N34     MOVF    NUM1,W
        SUBLW       'p'
        BTFSS   STATUS,Z
        GOTO    N35
        BSF     MOTOR_Lw
        BSF     MOTOR_DB
        GOTO    MAIN

N35     MOVF    NUM1,W
        SUBLW       'q'
        BTFSS   STATUS,Z
        GOTO    N36
        BSF     MOTOR_Rw
        BSF     MOTOR_UpF
        GOTO    MAIN

N36     MOVF    NUM1,W
        SUBLW       'r'
        BTFSS   STATUS,Z
        GOTO    N37
        BSF     MOTOR_Rw
        BSF     MOTOR_DownF
        GOTO    MAIN

N37     MOVF    NUM1,W
        SUBLW       's'
        BTFSS   STATUS,Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        GOTO  N38
        BSF   MOTOR_Rw
        BSF   MOTOR_UpB
        GOTO  MAIN

N38     MOVF  NUM1,W
        SUBLW       't'
        BTFSS STATUS,Z
        GOTO  N39
        BSF   MOTOR_Rw
        BSF   MOTOR_DB
        GOTO  MAIN
N39     MOVF  NUM1,W
        SUBLW       'u'
        BTFSS STATUS,Z
        GOTO  N40
        BSF   MOTOR_UpF
        BSF   MOTOR_UpB
        GOTO  MAIN
N40     MOVF  NUM1,W
        SUBLW       'v'
        BTFSS STATUS,Z
        GOTO  N41
        BSF   MOTOR_UpF
        BSF   MOTOR_DB
        GOTO  MAIN
N41     MOVF  NUM1,W
        SUBLW       'w'
        BTFSS STATUS,Z
        GOTO  N42
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BSF     MOTOR_DownF
        BSF     MOTOR_UpB
        GOTO    MAIN
N42     MOVF    NUM1,W
        SUBLW           'x'
        BTFSS   STATUS,Z
        GOTO    N43
        BSF     MOTOR_DownF
        BSF     MOTOR_DB
        GOTO    MAIN

N43     MOVF    NUM1,W
        SUBLW           'Z'
        BTFSS   STATUS,Z
        GOTO    MAIN
        CLRF    PORTB
        CLRF    PORTD
        CLRF    PORTE
        GOTO    MAIN

;/////////////// RS-232 SUBPROGRAM ////////////////////////
;********* initial RS232 **********************
INIT232         BSF     STATUS,RP0
        MOVLW           .25                     ;BIT RATE = 19200
        MOVWF           SPBRG
        BCF     TXSTA,SYNC
        BSF     TXSTA,BRGH
        BSF     TXSTA,TXEN
        BCF     STATUS,RP0
        BSF     RCSTA,CREN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BSF     RCSTA,SPEN              ;ENABLE SERIAL PORT
RETURN

END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้