สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ตัวสังเกตระยะไกลในงานวัดระดับน้ำมันดิบด้วยเครือข่ายวิทยุ

# REMOTE OBSERVER FOR CRUDE OIL USING RADIO NETWORK




โดย

นายจิรศักดิ์ ธีไชยวงค์

นางสาวพรนภา ตั้งจาตุรันต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 55770
วัน,เดือน,ปี 25 พ.ค. 2548

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

# REMOTE OBSERVER FOR CRUDE OIL USING RADIO NETWORK

BY

MR.JIRASAK THICHAIWONG

MS.PORNNAPHA TUNGJATURUNT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

**หัวข้อปริญญานิพนธ์** ตัวสังเกตระยะไกลในงานวัดระดับน้ำมันดิบด้วยเครือข่ายวิทยุ

**ชื่อนักศึกษา** 1.นายจิรศักดิ์ ธิไชยวงค์ รหัสประจำตัว 43010066

2.นางสาวพรนภา ตั้งจาตุรันต์ รหัสประจำตัว 43010290

**อาจารย์ที่ปรึกษา** รองศาสตราจารย์ ดร.ปิติเขต สู้รักษา

อาจารย์บุณย์ชนะ ภู่ระหงษ์

**ระดับการศึกษา** ปริญญาตรี วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

**ภาควิชา** วิศวกรรมสารสนเทศ

**ปีการศึกษา** 2546

---

ปริญญานิพธ์ฉบับนี้ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(อาจารย์บุณย์ชนะ ภู่ระหงษ์)

อาจารย์ที่ปรึกษา

.....

(รองศาสตราจารย์ ดร.ปิติเขต สู้รักษา)

อาจารย์ที่ปรึกษา

**หัวข้อปริญญานิพนธ์** ตัวสังเกตระยะไกลในงานวัดระดับน้ำมันดิบด้วยเครือข่ายวิทยุ
**ชื่อนักศึกษา** 1.นายจิรศักดิ์ ธิไชยวงค์ รหัสประจำตัว 43010066
2.นางสาวพรนภา ตั้งจาตุรันต์ รหัสประจำตัว 43010290
**อาจารย์ที่ปรึกษา** รศ.ดร.ปิติเขต สู้รักษา
อาจารย์บุณย์ชนะ ภู่ระหงษ์
**ระดับการศึกษา** ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ
**ภาควิชา** วิศวกรรมสารสนเทศ
**ปีการศึกษา** 2546

## บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาและสร้างระบบสังเกตและวัดปริมาณน้ำมันดิบในฐานผลิตและขุดเจาะน้ำมัน โดยการใช้ไมโครคอนโทรลเลอร์ 89C51 และ SFR08 Ultra Sonic ติดต่อกับเครื่องคอมพิวเตอร์ผ่านทางอุปกรณ์รับ-ส่ง คลื่นวิทยุ และพอร์ตอนุกรม

**THESIS TITLE** REMOTE OBSERVER FOR CRUDE OIL USING RADIO NETWORK

**STUDENT** MR.JIRASAK THICHAIWONG NO.43010066

MS.PORNNAPHA TUNGJATURUNT NO.43010290

**ADVISOR** ASSOC.PROF.DR.PITIKHATE SOORAKSA

MR.BOONCHANA PURAHONG

**DEPARTMENT** Information Engineering

**YEAR** 2003

## Abstract

This project presents a study and implementation of remote observer for crude oil using radio network for detection and measurment product of fuels pump system. The IT automation consists of a with microcontroller AT89C51 module and SRF08 Ultra Sonic via radio communication 144-146 MHz propagated to serial ports of a personal computer.

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่สำเร็จด้วยดี หากไม่ได้รับความร่วมมือจาก อาจารย์ทุกท่าน รุ่นพี่ รุ่นน้อง และเพื่อนๆทุกคน ที่ได้ให้ความช่วยเหลือทั้ด้านข้อมูล อุปกรณ์ คำแนะนำติชมต่างๆอันเป็นประโยชน์ ขอขอบพระคุณ รองศาสตราจารย์ ดร.ปิติเขต สู้รักษา และ อาจารย์ บุณย์ชนะ ภู่ระหงษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ อาจารย์ กฤดากร กล่อมการ ที่คอยให้คำแนะนำ ให้คำปรึกษา และคอยกระตุ้นให้พวกเราได้มีความคิดและแนวทางในการแก้ไขปัญหาในการทำงาน ขอบคุณรุ่นพี่ทุกคนที่อยู่ใน ห้อง Info -Dynamics Laboratory ที่ให้คำแนะนำและคำปรึกษาต่างๆและที่สำคัญที่สุด ขอขอบคุณ บิดา มารดา ที่ได้เลี้ยงดูพวกเรามา จนสามารถมีทุกวันนี้ได้

นายจิรศักดิ์ ธิไชยวงค์

นางสาวพรนภา ตั้งจาตุรันต์

# สารบัญ


| เรื่อง | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | ก |
| บทคัดย่อภาษาอังกฤษ | ข |
| กิตติกรรมประกาศ | ค |
| สารบัญ | ง |
| สารบัญรูป | ฉ |
| บทที่ 1 | 1 |
| บทนำ | 1 |
| 1.1 แนวคิดและที่มาของปัญหา | 1 |
| 1.2 จุดประสงค์ | 3 |
| 1.3 ขอบเขตของโครงงาน | 3 |
| 1.4 ผลที่คาดว่าจะได้รับ | 3 |
| บทที่ 2 | 4 |
| ทฤษฎีและหลักการ | 4 |
| 2.1 หลักการทำงาน | 4 |
| 2.2 การตรวจสอบแท่นดูดน้ำมันโดยใช้ไมโครคอนโทรลเลอร์ 89C51 | 9 |
| 2.3 การส่งข้อมูลมายังเครื่องคอมพิวเตอร์ผ่านเครื่องรับส่งวิทยุ และพอร์อนุกรม | 11 |
| 2.4 การประมวลผลและรายงานสถานะ | 13 |
| 2.5 ไมโครคอนโทรลเลอร์ | 13 |
| 2.6 SRF80 Ultra Sonic | 17 |
| 2.7 Packet Radio | 19 |
| 2.8 Radio Module RDL200T | 24 |
| บทที่ 3 | 26 |
| การออกแบบโครงงาน | 26 |
| 3.1 การออกแบบทางด้านซอฟแวร์ | 26 |

บทที่ 4 35
ผลการทดลอง 35
4.1 การทดลอง RDL 200T 35
4.2 การทดลองอุปกรณ์ Ultra Sonic 36
4.3 ทดลองการทำงานของโปรแกรม 37
บทที่ 5 42
สรุปผล ปัญหาที่พบ และข้อเสนอแนะ 42
5.1 สรุปผล 42
5.2 ปัญหาที่พบ 42
5.3 ข้อเสนอแนะ 43
บรรณานุกรม
ภาคผนวก

# สารบัญรูป

รูป หน้า

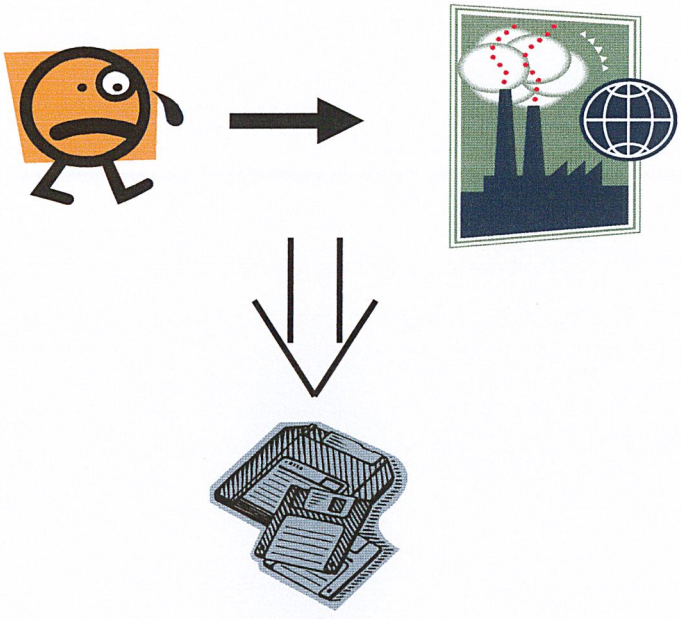

รูปที่ 1.1 แนวคิดและที่มาของปัญหา 1
รูปที่ 1.2 ภาพรวมของระบบที่พัฒนาใหม่ 2
รูปที่ 2.1 หลักการทำงานของระบบ 4
รูปที่ 2.2 เหตุผิดปกติที่ทำให้ประสิทธิภาพลดลง 6
รูปที่ 2.3 เหตุผิดปกติเมื่อระดับน้ำมันในถังไม่เปลี่ยนแปลง 7
รูปที่ 2.4 เหตุผิดปกติเมื่อไม่มีการตอบสนองจาก sensor 8
รูปที่ 2.5 การตรวจสอบแท่นดูดน้ำมันโดยใช้ไมโครคอนโทรลเลอร์ 9
รูปที่ 2.6 การส่งข้อมูลมายังเครื่องคอมพิวเตอร์ผ่านเครื่องรับส่งวิทยุ 11
รูปที่ 2.7 การรับส่งข้อมูลระหว่าง VB กับ89 C51 12
รูปที่ 2.8 ตำแหน่งขาต่างๆของไมโครคอนโทรลเลอร์ MCS-51 15
รูปที่ 2.9 CPU 89C51RD2-CM 15
รูปที่ 2.10 ลักษณะภายนอกของ SRF08-Ultra Sonic 17
รูปที่ 2.11 วงจรภาคส่งของอุปกรณ์ Ultra Sonic 18
รูปที่ 2.12 วงจรภาครับของอุปกรณ์ Ultra Sonic 18
รูปที่ 2.13 Packet Radio Station 20
รูปที่ 2.14 ชุดข้อมูลของ Packet Radio 23
รูปที่ 2 .15 RDL200T 24
รูปที่ 3.1 โปรแกรมติดต่อและควบคุมไมดครคอนโทรลเลอร์ 26
รูปที่ 3.2 ขั้นตอนที่ 1 การเขียนโปรแกรมติดต่อ I/O Serial Port 29
รูปที่ 3.3 ขั้นตอนที่ 2 การเขียนโปรแกรมติดต่อ I/O Serial Port 30
รูปที่ 3 .4 ขั้นตอนที่ 3 การเขียนโปรแกรมติดต่อ I/O Serial Port 31
รูปที่ 4.1 กราฟสัญญาณ FSK 35
รูปที่ 4.2 กราฟการทำงานของอุปกรณ์ Ultra Sonic 36

รูปที่ 4.3โปรแกรมติดต่อและควบคุมไมโครคอลโทรลเลอร์ 37
รูปที่ 4.4 หน้าจอการแสดงผ่านการวัดระดับน้ำมันที่ถัง ID 01 38
รูปที่ 4.5โปรแกรมการทำงานที่ไม่มีการตอบสนองของมโครคอนโทรลเลอร์ 39
รูปที่ 4.6 การทำงานของโปรแกรมเมื่อไม่มีการเปลี่ยนแปลงระดับน้ำมัน 40
รูปที่ 4.7 การทำงานของ โปรแกรมเมื่อ Sucker Rod มีปัญหา และระดับน้ำมันไม่เปลี่ยนแปลง 41

# บทที่ 1
# บทนำ

## 1.1 แนวคิดและที่มาของปัญหา

### 1.1.1 ระบบที่ใช้อยู่ในปัจจุบัน



รูปที่ 1.1 แนวคิดและที่มาของปัญหา

อธิบายรูปที่ 1.1

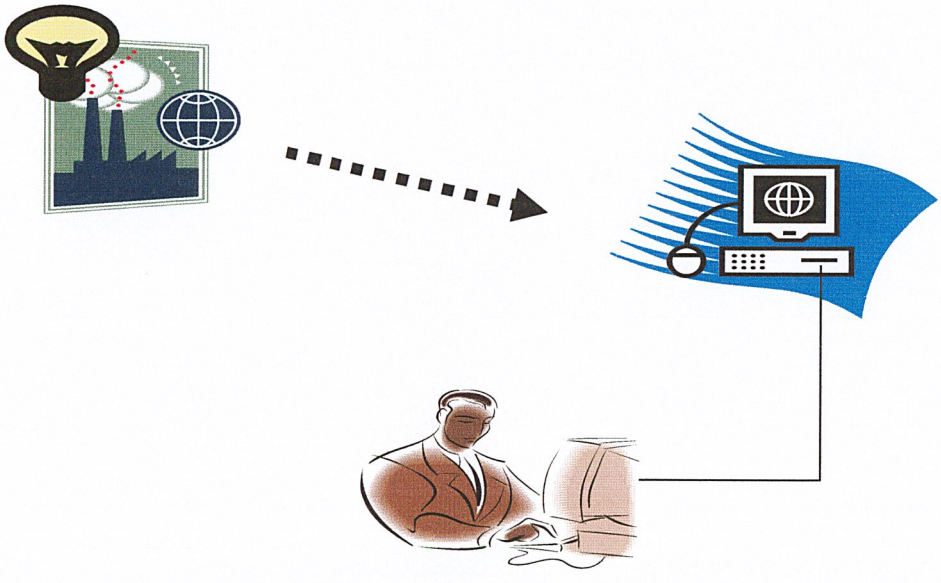
- เจ้าหน้าที่ไปตรวจสอบตำแหน่งต่างๆ ตามตารางเวลา
- ทำการจดบันทึกและแก้ไข เมื่อพบสิ่งผิดปกติ
- มีรถมาขนถ่ายน้ำมันตามตารางเวลา

จากระบบเดิม การตรวจสอบระดับน้ำมันในถังพักและการตรวจสอบการทำงานของแท่นดูดน้ำมันใช้วิธีการเดินไปตรวจสอบที่ตำแหน่งต่างๆ แล้วจึงทำการจดบันทึกการเปลี่ยนแปลงของระดับน้ำมันตามตารางเวลา และมีรถมาถ่ายน้ำมันออกจากถังตามตารางเวลาเช่นกัน การตรวจสอบนี้นอกจากจะใช้วัดปริมาณการผลิตแล้ว ยังจะเป็นการตรวจสอบอย่างคร่าวๆว่า ระบบหัวเจาะต่างๆยังทำงานดีอยู่หรือไม่ เมื่อเกิดปัญหาขึ้นมาจะไม่ทราบปัญหานั้นจนกว่าจะมีเจ้าหน้าที่มาตรวจสอบ และรถที่มาถ่ายน้ำมันไม่สามารถทราบแน่นอนว่าถังน้ำมันดังกล่าวมีน้ำมันเพียงพอหรือไม่ เมื่อใดก็ตามที่พบว่าถังน้ำมัน ไม่ผลิตน้ำมันตามปกติ จะต้องเปลี่ยนไปยังตำแหน่งอื่นที่อยู่ห่างกันในระยะไกล เพื่อถ่ายน้ำมันออกมา ทำให้เสียเวลาและค่าใช้จ่ายเป็นอย่างมาก

ปัญหาที่ต้องการแก้ไข

- เมื่อเกิดสิ่งผิดปกติขึ้นเจ้าหน้าที่จะไม่ทราบจนกว่าจะมีคนมาพบ
- รถที่มารับน้ำมันมาแล้วนำน้ำมันกลับไปตามที่ต้องการได้

**1.1.2 ระบบใหม่**



รูปที่ 1.2 ภาพรวมของระบบที่พัฒนาใหม่

คำอธิบายรูปที่ 1.2

- ตรวจสอบและวัดระดับน้ำมันโดยใช้ไมโครคอนโทรลเลอร์ 89C51
- ส่งข้อมูลมายังเครื่องคอมพิวเตอร์ที่อยู่ในระยะไกล ผ่านเครื่องรับส่งวิทยุ การประมวลผลและรายงานสถานะ

มีการนำเทคโนโลยีเข้ามาช่วยในการอำนวยความสะดวกคือ ติดตั้งอุปกรณ์ตรวจสอบสภาวะของแท่นดูดน้ำมัน และรายงานผลมายังเครื่องคอมพิวเตอร์ที่อยู่ในระยะไกล เพื่อประมวลผลและแสดงผลให้กับผู้ดูแล แท่นดูดน้ำมัน โดยที่ผู้ดูแลไม่ต้องไปตรวจสอบยังแท่นดูดน้ำมันน้ำมัน แต่สามารถทราบสภาวะต่างๆของถังน้ำมันได้

ระบบนี้ช่วยลดภาระการทำงานของเจ้าหน้าที่และลดความเสี่ยงของผู้ที่มาถ่ายน้ำมันออกจากถังพักเพราะระบบสามารถตรวจสอบได้ว่าขณะนั้นแท่นดูดน้ำมันมีการทำงานเป็นปกติและมีน้ำมันอยู่ในถังระดับใด อีกทั้งเมื่อใดก็ตามที่เกิดความผิดปกติขึ้น เจ้าหน้าที่สามารถวิเคราะห์และทราบความผิดปกติอย่างคร่าวๆได้

## 1.2 จุดประสงค์

1.2.1 เพื่อพัฒนาระบบการตรวจสอบแท่นดูดน้ำมัน

1.2.2 ลดขั้นตอนการทำงานของผู้ดูแลแท่นดูดน้ำมัน

## 1.3 ขอบเขตของโครงงาน

1.3.1 มีระบบตรวจสอบการทำงานของแท่นดูดน้ำมันทำงานเป็นปกติหรือไม่

1.3.2 รายงานผลที่ได้ผ่านทางเครื่องคอมพิวเตอร์ที่อยู่ในระยะไกล

## 1.4 ผลที่คาดว่าจะได้รับ

1.4.1 เป็นระบบที่สามารถนำไปใช้งานได้จริง

1.4.2 ช่วยในการดูแลและควบคุมการผลิตน้ำมัน

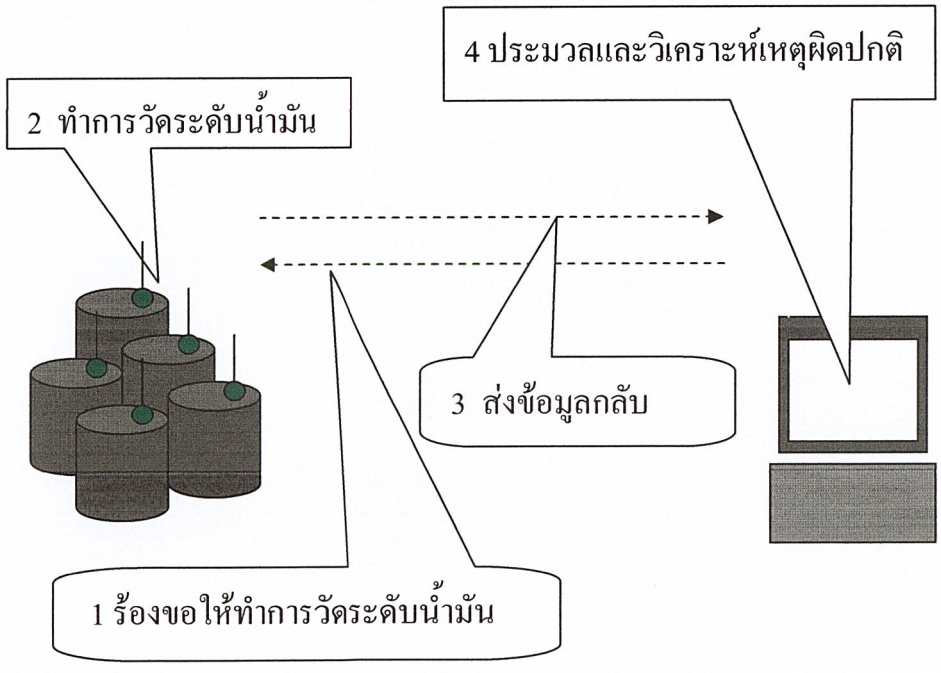
1.4.3 ลดภาระการทำงานของเจ้าหน้าที่

1.4.4 ประหยัดงบประมาณ

# บทที่ 2
# ทฤษฎีและหลักการ

บทนี้กล่าวถึงทฤษฎีและหลักการทำงานของของโครงงานนี้ ทั้งในส่วนของอุปกรณ์ที่นำมาใช้งาน หลักการทำงานโดยทั่วไปของอุปกรณ์ หลักการเขียนโปรแกรม การรับส่งข้อมูล การวิเคราะห์ถึงสิ่งผิดปกติที่อาจเกิดขึ้นกับระบบ และการประมวลผล

## 2.1 หลักการทำงาน




รูปที่ 2.1หลักการทำงานของระบบ

**2.1.1 การร้องขอให้ทำการวัดระดับน้ำมัน**

ผู้ใช้งานโปรแกรมทำการร้องขอให้ทำการวัดระดับน้ำมัน ผ่านโปรแกรม แล้วโปรแกรมจะทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ที่อยู่ที่ถังน้ำมัน ผ่านเครื่องรับส่งวิทยุ

**2.1.2 ทำการวัดระดับน้ำมัน**

ไมโครคอนโทรลเลอร์ได้รับสัญญาณจากคอมพิวเตอร์บอกให้ทำการวัดระดับน้ำมัน เมื่อทำการวัดระดับน้ำมันแล้วนำค่าที่วัดได้ส่งกลับมายังคอมพิวเตอร์

**2.1.3 ส่งข้อมูล**

การส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์ เป็นการส่งข้อมูลแบบไร้สายผ่านทาง เครื่องรับส่งวิทยุสื่อสารเพื่อให้การสื่อสาร

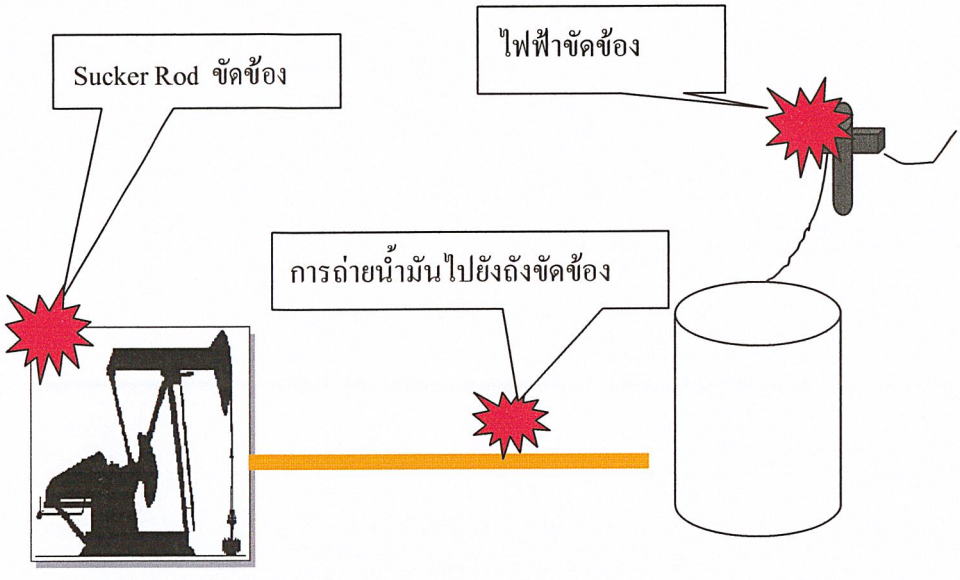
เมื่อไมโครคอนโทรลเลอร์ทำการวัดระดับน้ำมันได้แล้วส่งข้อมูลที่วัดได้ผ่านทางเครื่องรับส่งวิทยุ

**2.1.4 ประมวลผลและวิเคราะห์เหตุผิดปกติ**

ไมโครคอนโทรลเลอร์จะทำการควบคุมอุปกรณ์รับ-ส่งคลื่นอัลตราโซนิค เมื่อต้องการทราบระดับน้ำมัน ไมโครคอนโทรลเลอร์จะส่งสัญญาณ ไปยังอุปกรณ์อัลตราโซนิคแล้วจะทำการส่งคลื่นอัลตราโซนิคออกมา เมื่อกระทบกับระดับน้ำมันจะเกิดการสะท้อนกลับมายังตัวรับคลื่นสะท้อนของอุปกรณ์อัลตราโซนิค เมื่อรับคลื่นที่สะท้อนกลับมาจะมีสัญญาณส่งออกมาให้กับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะทำการนับเวลาตั้งแต่ส่งสัญญาณจนกระทั่งได้รับสัญญาณกลับเข้ามาอีกครั้ง แล้วส่งผลเวลาที่นับได้ให้กับคอมพิวเตอร์โดยส่งผ่านทางอุปกรณ์รับ-ส่งคลื่นวิทยุ และพอร์ทอนุกรม เพื่อประมวลผลหาระดับน้ำมันในถังพักน้ำมัน แล้วแสดงผลที่ได้ผ่านทางจอคอมพิวเตอร์

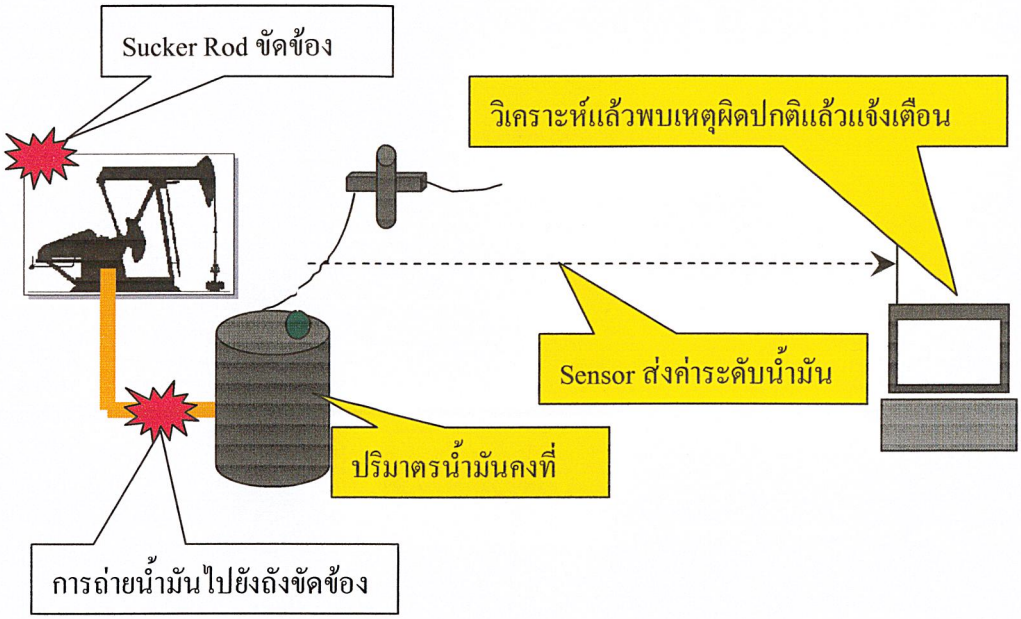
**2.4.1.1 เหตุผิดปกติที่ทำให้ประสิทธิภาพลดลง**




รูปที่ 2.2 เหตุผิดปกติที่ทำให้ประสิทธิภาพลดลง

ในระบบการทำงานของการนำน้ำมันดิบขึ้นมาเก็บไว้ในถังพัก สิ่งที่สามารถเกิดขึ้นได้และเป็นอุปสรรคในการทำงานของระบบที่ทำให้ระบบไม่สามารถดำเนินต่อไปได้ คือ ระบบไฟฟ้าขัดข้องทำให้ระบบหยุดทำงาน sucker rod ขัดข้องทำให้ไม่เกิดการปั๊มน้ำมันดิบขึ้นมาจากใต้ดิน หรือ ท่อส่งน้ำมันเกิดการชำรุดแตกหัก สิ่งต่างๆเหล่านี้ล้วนเป็นอุปสรรค และทำให้เกิดการสูญเสียทรัพยากรที่มีค่าอย่างเช่นน้ำมันดิบ การนำระบบเข้ามาช่วยในการตรวจสอบทำให้การดูและและควบคุมมีความง่ายมากยิ่งขึ้น

**กรณีที่ 1 ระดับน้ำมันในถังไม่เปลี่ยนแปลง**




รูปที่ 2.3 เหตุผิดปกติเมื่อระดับน้ำมันในถังไม่เปลี่ยนแปลง

เกิดจากสาเหตุ 3 ประการ คือ

**1. การถ่ายน้ำมันไปยังถังขัดข้อง**

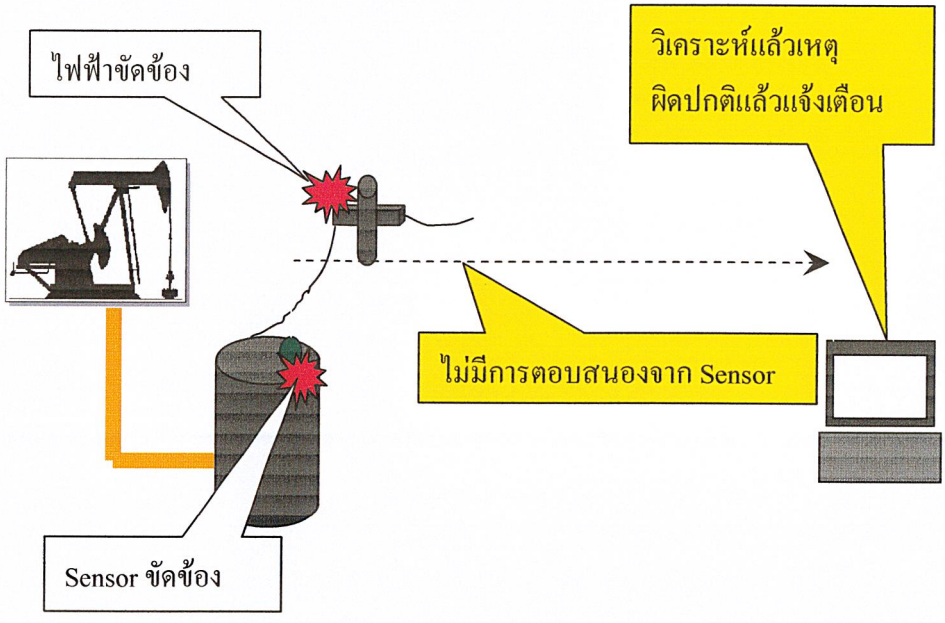
การถ่ายน้ำมันจากหัวเจาะไปยังถังขัดข้องอาจเกิดจากสาตุท่อส่งน้ำมันอุดตันระหว่างการส่ง ทำให้ระดับน้ำมันในถังไม่เกิดการเปลี่ยนแปลง หรือ เปลี่ยนแปลงน้อยมาก

**2. Sucker Rod ขัดข้อง**

Sucker Rod ขัดข้อง ทำให้ไม่เกิดการปั๊มน้ำมันจากใต้ดินเข้ามาเก็บไว้ในถังน้ำมัน อาจมีสาเหตุมาจาก Sucker Rod ขัดข้อง เสียหาย หรือหยุดทำงาน

**3. การถ่ายน้ำมันไปยังถังขัดข้อง และ Sucker Rod ขัดข้อง**

**กรณีที่ 2 ไม่มีการตอบสนองจาก Sensor**




รูปที่ 2.4 เหตุผิดปกติเมื่อไม่มีการตอบสนองจาก sensor

เกิดจากสาเหตุ 3 ประการคือ

**1. ระบบไฟฟ้าขัดข้อง**

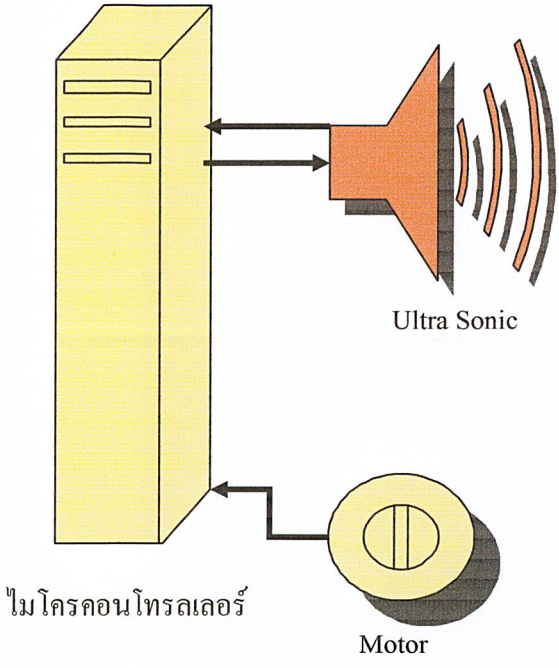
ทำให้ sensor ไม่สามารถทำงานได้จึงไม่สามารถวัด และส่งค่ากลับมายัง คอมพิวเตอร์ได้

**2. sensor ขัดข้อง**

sensor ขัดข้องไม่สามารถทำการวัด ระดับน้ำมันในถังได้ จึงไม่มีการส่งค่ากลับมายังคอมพิวเตอร์

**3. ระบบไฟฟ้า และ sensor ขัดข้อง**

## 2.2 การตรวจสอบแท่นดูดน้ำมันโดยใช้ไมโครคอนโทรลเลอร์ 89C51




รูปที่ 2.5 การตรวจสอบแท่นดูดน้ำมันโดยใช้ไมโครคอนโทรลเลอร์89 C51

### 2.2.1 89C51 กับ SFR08

ไมโครคอนโทรลเลอร์จะเริ่มจับเวลาและส่งสัญญาณไปยัง SFR08 เพื่อยิงคลื่นอัลตราโซนิค ออกมาเมื่อกระทบกับผิวของระดับน้ำมันจะเกิดการสะท้อนส่งกลับมายังตัวรับของ SRF08 แล้วส่งสัญญาณกลับมายังไมโครคอนโทรลเลอร์ ทันทีที่ไมโครคอนโทรลเลอร์ได้รับสัญญาณจะหยุดเวลาที่เริ่มนับตั้งแต่ส่งสัญญาณออกไป

### 2.2.2 89C51 กับ มอเตอร์ M1

มอเตอร์M1 เป็นมอเตอร์ที่ใช้ในการปั๊มน้ำมันเข้ามาเก็บไว้ในถังพัก จะทำการติดสวิตซ์ (detect switch) เมื่อมอเตอร์มีการทำงานหรือการหมุน จะส่งสัญญาณมายังไมโครคอนโทรลเลอร์ ทำให้สามารถตรวจจับการทำงานของมอเตอร์ได้

จากข้อมูลทั้ง 2 ส่วนนี้ไมโครคอนโทรลเลอร์จะทำการแปลงค่าและส่งเป็น บิท ข้อมูลมายังเครื่องคอมพิวเตอร์เพื่อนำข้อมูลที่ได้มาประมวลผลว่าแท่นดูดน้ำมันทำงานเป็นปกติหรือไม่ข้อมูลที่เครื่องคอมพิวเตอร์จะได้รับจากไมโครคอนโทรลเลอร์ มีดังนี้คือ

-หมายเลขรหัสของแท่นดูดน้ำมัน

-เวลาที่ได้จากการยิง อัลตราโซนิค เพื่อนำมาคำนวนหาระยะทาง

-สถานะมอเตอร์

**2.2.3 การพิจารณาการเลือกใช้ Timer/Counter**

1clock การทำงาน เท่ากับ 12 machinecycle

ใช้ Crystal 12 MHz จะได้ 1machinecycle = 1μS

Timer เพิ่มขึ้น 1 ค่า = 1 machinecycle = 1 μS

ระยะทางไป-กลับ)max) = 12 M , ความเร็วเสียง V=346 m/s

เวลาที่มากที่สุด t = 34682.0809 μS

ได้ t(max) = 34,683 = 34,683 แมชชีนไซเคิล

$2^{15} < 34683 < 2^{16}$

ใช้ Counter ขนาด 16 บิท ( mode 1 )

**2.2.4 การคำนวณหาระยะทาง**

Timer ใน MCS51 จะเพิ่มขึ้น 1 ทุกๆ 1 machinecycle เมื่อ Timer ทำการนับและหยุดเวลา จะได้ค่า ไบนารี ตามจำนวน machinecycle ที่นับได้ จึงนำค่า ไบนารีที่ได้นั้นมาทำการคำนวนหาระยะทางจริง ถ้าใช้ Crystal ขนาด 12 MHz จะได้ 1 machinecycle = 1 microsec

ตัวอย่างการคำนวณ

1111 1111 1010 0000= 2815 machinecycle

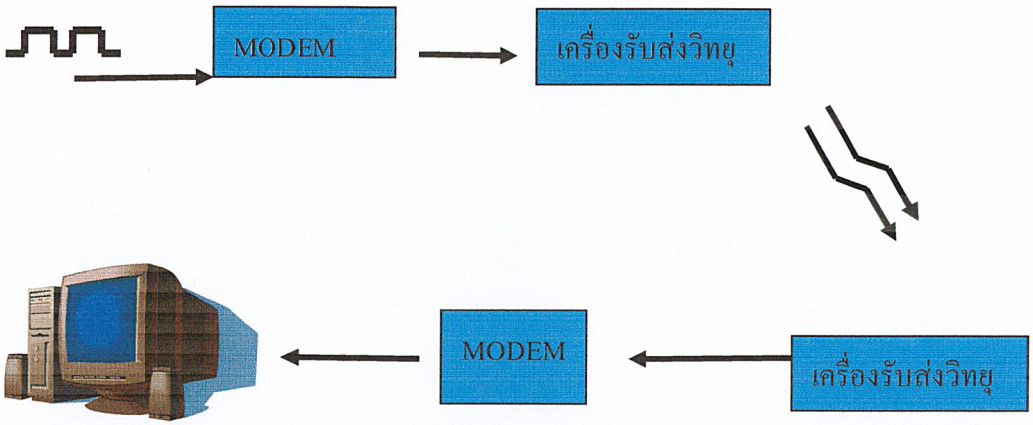
เวลาที่ใช้ใน 1 machinecycle คือ t = 2815 microsec

จาก S = V * t

ได้ ระยะทาง = $346*(2815*10^{-6}) = 0.973990$ M

ระยะทางจริง = 0.973990/2 =0.4869950

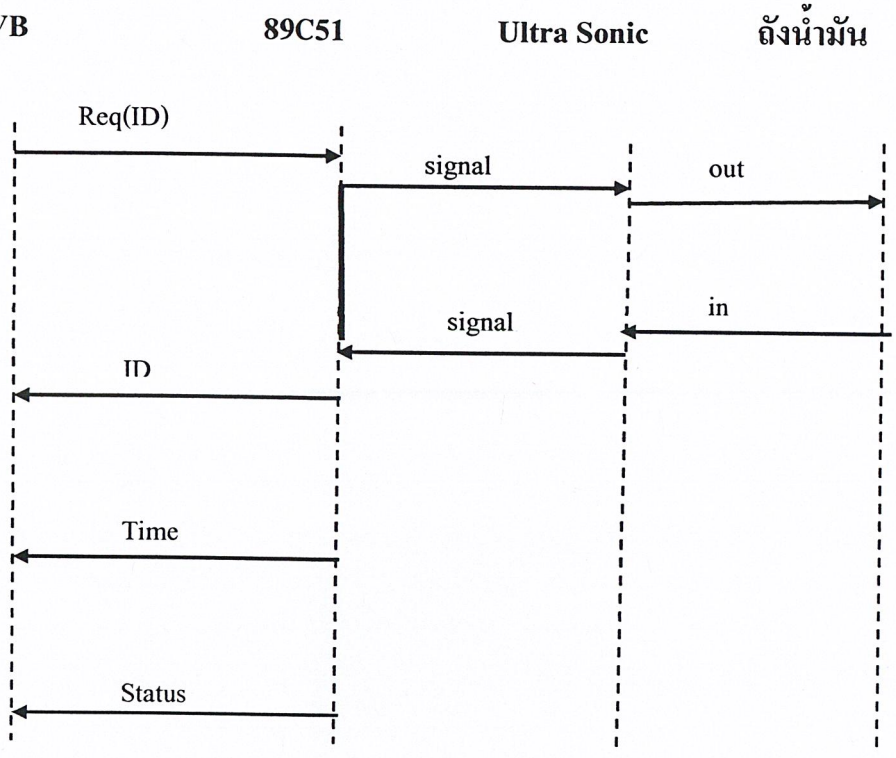
## 2.3 การส่งข้อมูลมายังเครื่องคอมพิวเตอร์ผ่านเครื่องรับส่งวิทยุ และพอร์อนุกรม




รูปที่ 2.6 การส่งข้อมูลมายังเครื่องคอมพิวเตอร์ผ่านเครื่องรับส่งวิทยุ และพอร์อนุกรม

สัญญาณที่ได้จาก 89C51 เป็นสัญญาณดิจิตอลถือว่าเป็นข้อมูลที่ต้องการส่งผ่านเครื่องรับส่งวิทยุในรูปแบบของ Packet Radio นำมาเปลี่ยนเป็นสัญญาณเสียง ด้วย radio module RDL200T จะได้สัญญาณ FSK ส่งผ่านคลื่นวิทยุ ที่ภาครับ รับด้วย RDL200T เพื่อแปลงสัญญาณกลับเป็นดิจิตอล และส่งไปยัง พอร์ตอนุกรม ( serial port )ไปยังเครื่องคอมพิวเตอร์ เพื่อทำการประมวลผลและรายงานสถานะต่อไป

### 2.3.1 การรับส่งข้อมูลระหว่าง VB กับ 89C51




รูปที่ 2.7 การรับส่งข้อมูลระหว่าง VB กับ89 C51

ผู้ใช้งานโปรแกรมต้องการตรวจสอบสถานะของถังเก็บน้ำมัน ทำการร้องขอผ่านทางโปรแกรม ( VB ) จากนั้นโปรแกรมจะรับการร้องขอนั้นและทำการติดต่อกับไมโครคอนโทลเลอร์ที่อยู่กับถังพักน้ำมันที่ทำการร้องขอไป โดยส่ง ชุดข้อมูล Req (ID) หมายถึงชุดคำสั่งร้องขอการติดต่อไปยังถังน้ำมันปลายทางทุกๆถัง ในลักษณะของการ Broadcast โดยในชุดข้อมูลนั้นจะมีการระบุ หมายเลข ถังน้ำมันที่ต้องการติดต่อ ชุด 89C51 เมื่อรับได้ข้อมูลดังกล่าวจะทำการตรวจสอบว่าเป็นหมายเลขถังของตนเองหรือไม่

ถ้าไม่ใช่จะทำการรอข้อมูลที่ระบุหมายเลยตนเอง ถ้าใช่ จะส่งสัญญาณให้กับอุปกรณ์ Ultra Sonic ในขณะเดียวกันก็สั่งให้ Timer เริ่มนับเวลา

เมื่ออุปกรณ์ Ultra Sonic ส่งสัญญาณกลับมายัง89 C51 ทำการหยุดเวลาใน Timer แล้วส่งค่าที่ได้ให้กับ VB โดยมีการส่ง หมายเลขถังของตนเอง (ID) ตามด้วยเวลาที่นับได้ ( Time ) และสถานะของมอเตอร์ ( Status ) ว่ายังทำงานอยู่หรือไม่

ในขณะเดียวกันเมื่อ VB ทำการร้องขอการติดต่อกับ 89C51 จะทำการนับเวลาไปพร้อมกันหากรอการตอบกลับเกินกว่าเวลาที่กำหนดจะทำการส่งข้อมูลชุดเดิมซ้ำอีกครั้งและทำการนับเวลาใหม่ หากรอการตอบกลับเกินจากช่วงเวลาที่กำหนดในครั้งที่ 5 จะฟ้องไปยังผู้ใช้งานโปรแกรม ว่า Disconnect หรือ ไม่มีการตอบสนองจากหมายเลขถังน้ำมันที่ระบุ และพยายามทำการติดต่อกับหมายเลขนั้น จน กว่าจะติดต่อได้หรือผู้ใช้งานสั่งยกเลิก

เมื่อ 89C51 ที่ไม่ใช่หมายเลขที่ถูกทำการร้องขอการติดต่อ อยู่ในสถานะรอข้อมูลสำหรับการร้องขอการติดต่อครั้งใหม่

## 2.4 การประมวลผลและรายงานสถานะ

เมื่อรับข้อมูลจากทางพอร์ตอนุกรมแล้ว นำข้อมูลที่ได้มาทำการประมวลผลว่าขณะนั้นการทำงานของแท่นดูดน้ำมันเป็นอย่างไร แล้วแสดงผลทางหน้าจอคอมพิวเตอร์สำหรับผู้ดูแล

### 2.4.1 การแสดงผล

- สถานะการทำงานของมอเตอร์
-ระดับน้ำมันในถัง
-หมายเลขถังน้ำมัน

## 2.5 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 (ที่มาจาก[4]) เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มีอุปกรณ์สนับสนุนประกอบอยู่ภายใน หลายอย่างได้แก่ หน่วยความจำ สำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับเวลา อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายในนี้นี่เอง ทำให้การใช้งานเพิ่มขึ้นและมีประสิทธิภาพมากขึ้นโดยไม่ต้องมีการเชื่อมต่ออุปกรณ์เพิ่มเติมมากเหมือนกับตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้เราต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่นเพิ่ม

เติมเช่น ไอซี 8255 หรือหน่วยความจำภายนอก เรายังสามารถนำมาเชื่อมต่อเพื่อเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีก

**2.5.1 คุณสมบัติทั่วไปที่สำคัญของไมโครคอนโทรลเลอร์ตระกูล MCS-51**

-เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต

-มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี

-มีขาสัญญาณอินพุทเอาร์พุทจำนวน 32 บิต

-สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (external data memory )โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 k

-สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก(external program memory) โดยอ้างตำแหน่ง แอดเดรสได้ถึง 64 k

-มีหน่วยความจำโปรแกรมภายในตัว (on-chip data memory)ขนาด 128 ไบท์ โดยเฉพาะเบอร์ 8032และ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบท์

-หน่วยความจะข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่ายส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น

-มีไทเมอร์/เคาน์เตอร์ (time/counter)ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8032 และ 8052 จะมีไทเมอร์/เคาน์เตอร์ จำนวน 3 ตัว

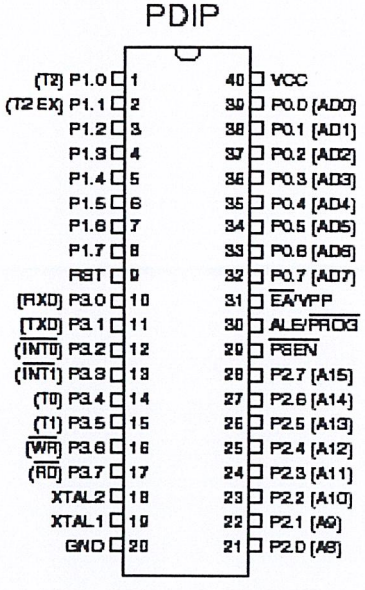
-การอินเตอร์รัปต์สามารถทำได้จาก 5แหล่งกำเนิดโดยเฉพาะเบอร์ 8032 และ 8052จะทำการอินได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัปยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ

-มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็ก(full duplex)

-มีคำสั่งในการคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์

-คำสั่งโดยส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่

**2.5.2โครงสร้างภายนอกของ mcs-51 (ทั่วไป)**




รูปที่ 2.8 ตำแหน่งขาต่างๆของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรเลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังแสดง ในรูป 2.2

### 2.5.3 CPU ในตระกูล MCS-51 ใหม่ 89C51RD2 - CM มีถึง 6 PORT ให้ใช้งาน




รูปที่ 2.9 CPU 89C51RD2-CM

**คุณสมบัติ**ของ CPU 89C51RD-2CM ของ ATMEL

- 80C52 COMPATIBLE
- 6 PORT ใช้งาน ( 48 BIT )
- 64 KBYTE ส่วนโปรแกรมแบบ FLASH
- 256 BYTE RAM, 1024 EXP RAM
- 2 KBYTE EEPROM บน CPU สามารถใช้เก็บข้อมูลได้
- 68 PIN PLCC TYPE
- RUN ความถี่ได้ 2 แบบ โดยสามารถเลือกได้จากโปรแกรม แบบ6 CLOCK หรือ แบบ 12 CLOCK สูงสุดได้40 MHz
- DUAL DATA POINTER ให้คุณเขียนโปรแกรมง่ายขึ้น
- PCA, HIGHT SPEED OUTPUT, COMPARE/CAPTURE, PULSE WIDTH MODULATOR, WDT

ในชุดของบอร์ดทดลอง ของ89 C51RD-CM

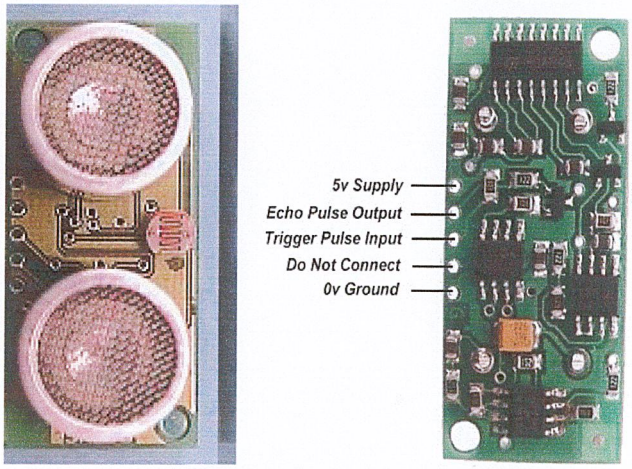
- **CP-JR51RD 2**ใช้ CPU ของบริษัท ATMEL ในรุ่น89 C51RD-2CM ตัวถังแบบ PLCC 68 PIN
- ใช้ X'TAL 18.432 MHz บนบอร์ด เลือกจากโปรแกรมได้ว่าจะให้ RUN ความถี่แบบ6 CLOCK ต่อคำสั่ง หรือ12 CLOCK ต่อคำสั่ง36.864 ) MHz หรือ18.432 MHz(

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



- 6 PORT ใช้งาน 48) BIT I/O PORT(
- 64 KBYTE โปรแกรมแบบ FLASH จากเครื่อง PC ได้, 1 + 256 KBYTE RAM ภายในใช้งาน
- 2 KBYTE EEPROM ภายใน CPU
- 4 PIN RS232 MAX232 ON BOARD
- 5 PIN ET DOWNLOAD PROGRAM
- 34 PIN I/O ET BUS จำนวน 2 ชุด
- 7805 POWER SUPPLY ON BOARD
- สามารถ DOWNLOAD โปรแกรมที่เขียนขึ้นบนคอมพิวเตอร์ PC ที่เป็น HEX FILE แล้วเข้ามายังบอร์ดได้โดยตรงผ่านทาง PORT RS232 ของเครื่อง PC เข้ายังPORT RS232 ชองบอร์ด CP-JR51RD2 ตัวโปรแกรมสามารถทำงานบนระบบ OS WINDOWS 95/98/ME/XP\2000

## 2.6 SRF80 Ultra Sonic



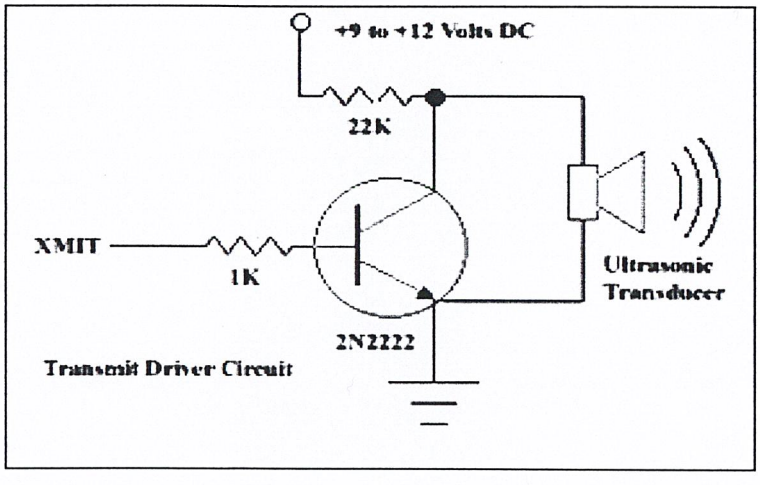

รูปที่ 2.10 ลักษณะภายนอกของ SRF08-Ultra Sonic(ที่มาจาก[6])

55770

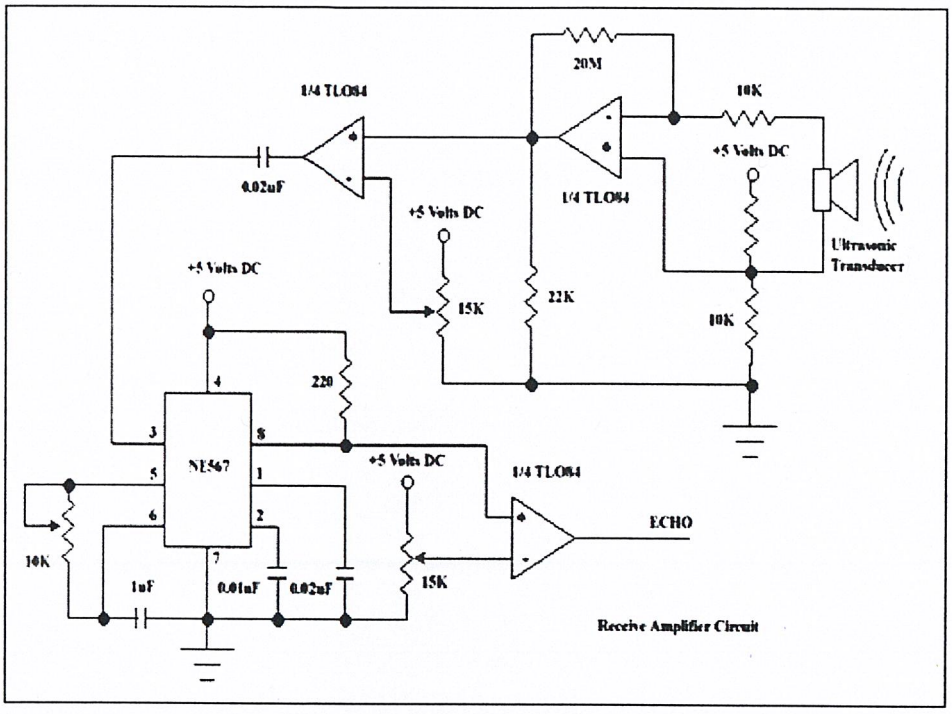
**2.6.1 ข้อมูลทางเทคนิค**

- ขนาด 43 มิลลิเมตร x 20 มิลลิเมตร x 17มิลลิเมตร
- จ่ายกระแสไฟ 5 V กระแสไฟขณะปกติ3 mA เมื่อส่งคลื่นอัลตราโซนิค 12-11 mA
- ความถี่ที่ใช้ 40 KHz
- ส่งคลื่นความถี่วัดระยะได้ใกล้ที่สุด 3 เซนติเมตร และไกลที่สุด 6 เมตร

เป็นอุปกรณ์ตรวจวัด)sensor) ชนิดหนึ่งที่มีการใช้กันอย่างแพร่หลาย ในการตรวจจับหรือวัดระยะต่างๆ เมื่อมีสัญญาณโลจิก5+) 1 V ) มาที่ขา SCL (Trigger Pulse Input) ชุดควบคุมของอุปกรณ์จะทำการสร้างคลื่นความถี่ค่าหนึ่ง 40) KHz (แล้วส่งออกมาทาง Ultra Sonic Transducers เมื่อคลื่นความถี่ดังกล่าวไปกระทบกับสิ่งกีดขวางใดๆ เช่น ของแข็ง น้ำ จะเกิดการสะท้อนกลับมาเมื่อตัวรับ)Ultra Sonic Transducers (จับสัญญาณได้ จะเปลี่ยนเป็นสัญญาณ DC +5 V ส่งไปยังขา SCL (Echo Pulse Output)




รูปที่ 2.11 วงจรภาคส่งของอุปกรณ์ Ultra Sonic(ที่มาจาก[6])




รูปที่ 2.12 วงจรภาครับของอุปกรณ์ Ultra Sonic(ที่มาจาก[6])

## 2.7 Packet Radio

Packet Radio (ที่มาจาก[3]) คือ รูปแบบหนึ่งของการติดต่อสื่อสาร กล่าวคือ แทนที่เราจะติดต่อกันด้วยเสียง (Phone) ที่คุ้นเคย โดยใช้ปากพูดโต้ตอบกัน แต่เราใช้คอมพิวเตอร์ในการรับส่งข้อมูล โดยมีตัวกลางที่เรียกว่า Modem (Modulate & Demodulate) ทำหน้าที่แปลงข้อมูล (DATA) จากเครื่องคอมพิวเตอร์ ให้เป็นเสียงส่งออกไปทางเครื่องวิทยุรับส่ง ขณะเดียวกัน ก็แปลงสัญญาณเสียงจากเครื่องวิทยุรับส่ง ให้เป็นข้อมูลปรากฏบนจอคอมพิวเตอร์ ยกตัวอย่างให้เข้าใจง่ายกว่านี้คือ

เมื่อเราต้องการส่งคำว่า “สวัสดีครับ” เราใช้ Keyboard พิมพ์คำว่า “สวัสดีครับ” แล้วตัว Modem จะแปลงคำว่า “สวัสดีครับ” เป็นเสียงดัง “แอ๊ด.....” ออกอากาศทางเครื่องวิทยุรับส่ง คู่สถานีจะต้องมีอุปกรณ์เหมือนกัน เมื่อรับสัญญาณเสียง “แอ๊ด.....” ได้แล้ว ตัว Modem ก็จะแปลงเป็นตัวอักษร “สวัสดีครับ” ปรากฏบนจอคอมพิวเตอร์

**2.7.1 คุณสมบัติของ Packet Radio**

- Keyboard to Keyboard พูดคุยโดยการพิมพ์ข้อความบนแป้นพิมพ์ (Keyboard) แทนการพูดคุยด้วยเสียง

- เก็บและบันทึกข้อมูลข่าวสารได้ แม้ว่าจะไม่มีใครอยู่หน้าเครื่องวิทยุ

- เรียกดูข่าวสารได้ตลอดเวลา

- รับส่งได้ทั้งตัวอักษร ภาพ และ Binary file (ไฟล์ที่เป็นรูปภาพ ไม่สามารถดูได้ทันที แต่ใช้โปรแกรมอื่นช่วยดู)

- หากเป็นเครื่อง TNC จะมีระบบ Mailbox ในตัวรับฝากจดหมาย โดยไม่จำเป็นต้องเปิด เครื่องคอมพิวเตอร์ทิ้งไว้ตลอดเวลา

- มีระบบสถานีถ่ายทอดข้อมูล (Digipeater) ทำให้ติดต่อสื่อสารได้ไกลยิ่งขึ้น

- ทำเป็นระบบเครือข่ายและศูนย์ข้อมูลได้ (Packet Bulletin Board System PBBS)

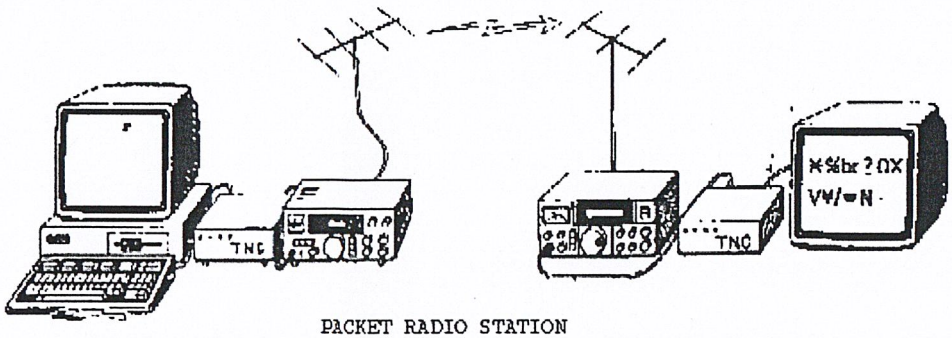
- ใช้งานได้หลายคู่สถานีในช่องความถี่เดียวกัน เนื่องจากทำงานแบบ CSMA (Carrier Sense Multiple Access)

- DX Packet Cluster คือการรายงานเกี่ยวกับข้อมูลสถานีทางไกล เช่น ขณะนี้ รับสถานีใดได้ที่ความถี่เท่าใด Mode ไหน ระดับสัญญาณเป็นเช่นใด เพื่อช่วยให้นักวิทยุสมัครเล่น ที่ใช้งานย่านความถี่ HF ที่ต้องการแลกเปลี่ยน QSL Card บางแห่ง มีโอกาสติดต่อกันได้ง่ายขึ้น

- Satellite Communication ติดต่อสื่อสารผ่านดาวเทียมวิทยุสมัครเล่น

- ใช้เครือข่าย Packet Radio เป็นเครือข่ายฉุกเฉิน

**2.7.2 สถานี Packet Radio**




รูปที่ 2.13 Packet Radio Station

**2.7.2.1 เครื่องวิทยุรับส่ง** ใช้ได้กับเครื่องวิทยุรับส่งทุกรุ่น ทุกยี่ห้อ ไม่ว่าเป็นแบบมือถือ ติดตั้งรถยนต์ หรือแบบประจำที่ เพียงแต่มีช่องสำหรับต่อไมโครโฟนและลำโพงภายนอก หรือบางรุ่นมีช่องสำหรับสื่อสารข้อมูลโดยเฉพาะ

**2.7.2.2 เครื่องคอมพิวเตอร์** สามารถใช้กับเครื่องคอมพิวเตอร์รุ่นเก่า Serial Card บางรุ่นอาจไม่สามารถใช้งานกับ Modem ได้ Software มีทั้งแบบทำงานบน Dos, Window95/98, Linux

**2.7.2.3 Packet Modem/TNC** ทำหน้าที่เป็นตัวกลางแปลงสัญญาณเสียงให้เป็นข้อมูล/ตัวอักษร/ภาพ และแปลงข้อมูล/ตัวอักษร/ภาพ ให้เป็นสัญญาณเสียงส่งออกอากาศ )ในที่นี้ใช้ TCM 3101(

**2.7.3 ข้อจำกัดระหว่างความเร็ว Modem กับวิทยุรับส่ง**

เนื่องจาก Modem ความเร็ว 1200/2400bps เป็นแบบ AFSK (Audio Frequency Shift Keying) จะใช้ช่อง Microphone กับลำโพง ในการรับส่งข้อมูล ข้อกำหนดทางเทคนิคของการติดต่อสื่อสารข้อมูลแบบ Packet Radio ให้ใช้แถบความกว้างของความถี่ไม่เกิน 6 kHz และข้อกำหนดของเครื่องวิทยุรับส่งแบบ FM ซึ่งมีแถบความกว้าง ของความถี่ไม่เกิน 16kHz (ช่วงห่างของความถี่แต่

ละช่อง) และความเบี่ยงเบนของความถี่สูงสุด (Maximum frequency deviation (FM) ไม่เกิน +/-5kHz เพราะฉะนั้น ต้องการใช้ช่องแค่ 6kHz แต่มีช่องความถี่ห่างถึง 16 kHz การรับส่งข้อมูล ผ่านช่อง Microphone ที่ความเร็ว 1200/2400bps จึงไม่เป็นปัญหาและไม่ต้องดัดแปลงเครื่องวิทยุรับส่งใด ๆ

สำหรับย่านความถี่ HF ซึ่งมีข้อกำหนดทางเทคนิคของวิทยุว่า ให้มีแถบความกว้างของความถี่สำหรับ SSB ไม่เกิน 3kHz แต่ Packet Radio ใช้ไม่เกิน 6 kHz ใช้จริงประมาณ 3 kHz การติดต่อผ่านช่อง Microphone ก็ยังใช้ได้อยู่ แต่ช่องห่างแต่ละช่องแคบเกินไป จะส่งไปเต็มที่ก็จะไปรบกวนช่องอื่น และย่านความถี่นี้ มีการรบกวนในบรรยากาศ สูงมาก เพื่อประสิทธิภาพและความสำเร็จในการรับส่ง จึงจำเป็นต้องลดความเร็วลงเหลือ 300bps )แต่ในที่นี้ใช้ช่วงความถี่ 144 MHz ซึ่งเป็นช่วงความถี่เฉพาะของราชการทหารไม่เกิดการรบกวนช่องอื่น จึงสามารถส่งได้ที่ความเร็ว1200 bps)

และความเร็วที่สูงกว่านี้ มีความจำเป็นต้องใช้แถบความถี่ที่กว้างมาก ก็ไม่สามารถส่งผ่านช่อง Microphone ได้ จึงจำเป็นต้องดัดแปลงเครื่องวิทยุรับส่ง ซึ่งปัจจุบัน การรับส่งความเร็วสูงที่ 9600bps มีความแม่นยำมากขึ้น ทำให้ผู้ผลิตเครื่องวิทยุรับส่ง ได้ทำช่องสำหรับสื่อสารข้อมูล (DATA Socket) ไว้ให้แล้ว

**2.7.3.1 ความเร็ว**

Modem Packet Radio ที่ใช้ในการติดต่อสื่อสารแบบไร้สาย มีความเร็วตั้งแต่ 300bps - 808,000bps ความเร็วในแต่ละระดับ ก็เหมาะสมกับข้อกำหนดทางเครื่องวิทยุรับส่งด้วย ในทางกิจการวิทยุสมัครเล่น ได้ใช้งานในแต่ละระดับความเร็วกับย่านความถี่ต่าง ๆ ดังนี้

1. **ความเร็ว 300bps** ความเร็วในระดับนี้ จะใช้กับย่านความถี่ HF (1.8-30MHz) ซึ่งมีการรบกวนค่อนข้างสูง และข้อจำกัดในเรื่องข้อกำหนดทางเทคนิคของวิทยุรับส่ง

2. **ความเร็ว 1200/2400bps** นิยมใช้งานตั้งแต่ย่านความถี่ VHF ขึ้นไป อุปกรณ์หาซื้อง่าย แต่สำหรับความเร็ว 2400bps นักวิทยุสมัครเล่นไม่ค่อยนิยมใช้งาน

3. **ความเร็ว 9600bps** จะใช้งานได้ตั้งแต่ย่าน UHF ขึ้นไป การใช้งานความเร็วระดับนี้จำเป็นต้องปรับแต่ง เครื่องวิทยุรับส่ง จึงจะใช้งานได้ แต่ปัจจุบันนี้ เครื่องวิทยุรุ่นใหม่ ได้ออกแบบ

สำหรับการสื่อสารข้อมูล ความเร็วสูง โดยมีช่องสื่อสารข้อมูล DATA SOCKET แยกไว้ต่างหาก แม้กระทั่ง เครื่องย่านความถี่ VHF ก็สามารถใช้ความเร็วระดับนี้ได้ ความเร็วระดับนี้ นิยมใช้สื่อสารระหว่างสถานีหลักมากกว่า ซึ่งมีปริมาณ ข้อมูลจำนวนมาก

สำหรับความเร็วที่สูงมากกว่านี้ มักจะใช้งานเฉพาะทาง ติดต่อแบบจุดต่อจุด เครื่องวิทยุ รับส่ง ก็ต้องใช้ความถี่สูงมาก ๆ ทั้งตัว Modem ความเร็วสูงกับวิทยุรับส่งหาซื้อยากมาก ส่วนใหญ่จะสร้างขึ้น มาใช้งานโดยเฉพาะ

**2.7.3.2 ความสามารถในการรับส่งข้อมูล**

ตัวอักษรไทยกับภาษาอังกฤษ 1 ตัว เรียกว่า 1 ไบต์ (Byte) และกำหนดไว้ว่า 1 byte มี 8 bit เพราะฉะนั้น 1200 bit จึงมีเท่ากับ 150 ตัวอักษร ความเร็ว 1200bps จึงเท่ากับความเร็วในการรับส่งข้อมูล 150 ตัวอักษรต่อวินาที หรือเท่ากับ 9000 ตัวอักษรต่อนาที หากเปรียบเทียบกับหน้ากระดาษ A4 จำนวน 1 หน้าที่ไม่มีรูปก็ประมาณหน้าละ 3-4000 ตัวอักษร

ในการส่งข้อมูลแต่ละครั้งจะทำการส่งข้อมูลออกไปเป็นชุดๆ และทุก ๆ ชุดที่ถูกส่งออกไป ไม่ได้มีแต่ข้อมูลที่จะส่งเท่านั้น ยังมีข้อมูลชื่อผู้ส่ง ผู้รับ และระบบตรวจสอบความผิดพลาด ถูกส่งออกไปทุกครั้ง รวมทั้งช่วงระยะเวลาต่าง ๆ หรือ A4 1 หน้า ในสภาวะปกติ ใช้เวลาประมาณ 1-2 นาที

| FLAG | ADDRESS | CONTROL | PID&DATA | FCS | FLAG |
|---|---|---|---|---|---|
| 01111110 | 112-560 BIT | 8BIT | 8*NBIT | 16BIT | 01111110 |

รูปที่ 2.14 ชุดข้อมูลของ Packet Radio

### 2.7.4 ความเร็วภาคส่งและความเร็วภาครับ

การรับส่งข้อมูลระหว่างสถานีต้องตั้งความเร็วของแต่ละสถานีให้ตรงกัน เนื่องจากรับส่งข้อมูลแบบ AFSK (Audio Frequency Shift Keying) ใช้ความถี่เสียง 2 เสียงที่ต่างกันในการ ส่ง ละ รับ โดยให้โทนเสียงหนึ่งแสดง สถานีเป็น 0 และอีกโทนเสียงหนึ่งแสดงสถานีเป็น1

-ความเร็ว 1200bps ใช้โทนเสียง 1200Hz กับ 2200Hz มีโทน เสียงต่างกัน 1000Hz

- ความเร็ว 300bps ใช้ โทน เสียง 2125Hz กับ 2295Hz มี โทน เสียงต่างกัน 170Hz

เพราะฉะนั้น โทน เสียงที่ไม่เหมือนกัน จึงไม่สามารถติดต่อกันได้

## 2.8 Radio Module RDL200T



รูปที่ 2.15 RDL 200T

สำหรับเครื่องส่งที่สถานีฐานนั้นจะใช้เป็นโมดูล (Module) ซึ่งมีคุณสมบัติดังนี้

- ความถี่ที่ใช้ คือ 433.92 MHz และ 434.33 MHz ซึ่งจะใช้ได้ครั้งละ 1 ความถี่
- อัตราการส่งข้อมูล 4800,9600,19200 bit/sec
- รูปแบบของข้อมูล ไม่มีพาริตี้บิต (None Parity) ส่งข้อมูล 8 บิต มี 1 Stop Bit ใช้การส่งข้อมูลแบบฮาฟดูเพล็ค (Half-duplex) บนช่องสัญญาณแบบอะซิงโครนัส (Asynchronous Mode)

- ระยะในการส่ง ประมาณ 50- 150 เมตร
- ใช้แรงดันไฟ 9 VDC
- กำลังในการส่ง 5 มิลลิวัตต์
- การมอดูเลตแบบ FSK

ซึ่งเครื่องส่ง เป็นตัวที่ใช้ติดต่อกับ คอมพิวเตอร์ผ่าน พอร์ตอนุกรม มีไฟที่แสดง สถานะต่างๆ สำหรับ ในการติดต่อ ระหว่างคอมพิวเตอร์กับเครื่องส่งนั้น ที่คอมพิวเตอร์จะต้องมีตัวที่จัดการการส่งข้อมูลให้เครื่องส่งที่ต่ออยู่นั้นให้สามารถทำงานด้วยกันได้ เช่นความเร็วในการส่งข้อมูลของทั้งสองตัวจะต้องเซตให้เท่ากันและการเซตอื่นๆจะต้องเหมือนกันด้วย ซึ่งตัวที่ใช้จัดการต่างๆในโครงงานนี้จะใช้โปรแกรมวิชวลเบสิค เวอร์ชัน 6 ( visual Basic 6.0)[1] เป็นตัวจัดการส่งข้อมูลไปยังเครื่องส่ง ซึ่งจะส่งข้อมูลแบบอนุกรม 8 บิต มี stop bit 1 บิต ซึ่งในโครงงานนี้จะใช้ความเร็วในการส่งข้อมูล 4800 บิตต่อวินาที ซึ่งอัตราการส่งข้อมูล 4800 บิตต่อวินาทีก็เพียงพอในการส่งข้อมูลสำหรับควบคุมหุ่นสำรวจแล้ว และที่เครื่องส่งที่ต่ออยู่กับคอมพิวเตอร์นั้นจะต้องเซตด้วยอัตราการส่งข้อมูล 4800 บิตต่อวินาทีด้วย โดยการเซตที่ดิปสวิตซ์ตัวที่ 1 ON ตัวที่ 2 และ 3เป็น Off

ส่วนของโมดูลภาครับ จะมีลักษณะเดียวกันกับโมดูลส่ง ซึ่งเป็นโมดูลที่ทำหน้าที่รับสัญญาณจากเครื่องส่งซึ่งอยู่ที่สถานีฐานแล้วจากนั้นก็จะดีมอดูเลตเพื่อให้ได้สัญญาณเหมือนกับที่คอมพิวเตอร์ส่งมา เป็นโมดูลขนาด 28 ขา สำหรับการต่อวงจร เพื่อใช้งานจะต้องมีการต่ออุปกรณ์ที่ใช้กับโมดูลรับก่อนจึงจะสามารถใช้งานได้ เช่น ดิปสวิตซ์ และไฟแสดงสถานะการส่งและรับข้อมูล

สำหรับการใช้งานนั้นจะต่อเครื่องรับเข้ากับ ไมโครคอนโทรลเลอร์ MCS-51 เพื่อทำการประมวลผลแล้วจากนั้นก็จะนำไปควบคุมอุปกรณ์ต่างๆ แสดงดังวงจรรูปที่2.15 ซึ่งการติดต่อกันระหว่าง โมดูลรับ และ ไมโครคอนโทรลเลอร์ จะเป็นแบบอนุกรม

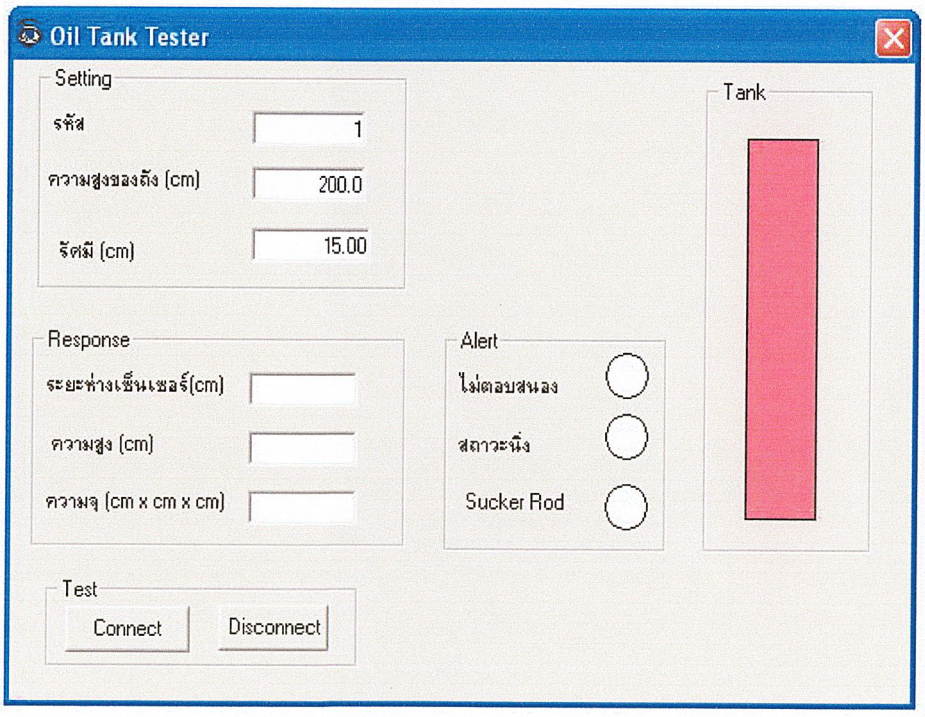
บทต่อไปจะกล่าวถึงการออกแบบโครงงาน ว่ามีรูปแบบการทำงานอย่างไร การใช้งานโปรแกรมและลักษณะการแจ้งเตือน เมื่อเกิดความผิดปกติต่างๆที่ตรวจสอบได้ รวมถึงการวิเคราะห์ถึงเหตุผิดปกติที่อาจเกิดขึ้นจากผลการตรวจสอบนั้น

# บทที่ 3
# การออกแบบโครงงาน

บทนี้กล่าวถึงการออกแบบโครงงานและการใช้งานโปรแกรมที่สร้างขึ้นสำหรับผู้ดูแลระบบ รวมถึงการกล่าวถึงลักษณะความผิดปกติที่ตรวจพบ รูปแบบการแจ้งเตือน และการวิเคราะห์ถึงความผิดปกติที่เป็นไปได้

## 3.1 การออกแบบทางด้านซอฟแวร์

### 3.1.1 User Interface




รูปที่ 3.1 โปรแกรมติดต่อและควบคุมไมโครคอลโทรลเลอร์

หน้าจอสำหรับการติดต่อกับผู้ใช้งานโปรแกรม แบ่งเป็น 4 ส่วน จากรูปที่ 3.1 คือ

#### 3.1.1.1 Setting

เป็นส่วนที่ผู้ใช้งานโปรแกรม ต้องกรอกข้อมูลให้ครบถ้วนสำหรับ ระบุหมายเลขของถังน้ำมันที่ต้องการตรวจสอบ ( รหัส ) ความสูงของถังน้ำมัน( หน่วยเป็นเซนติเมตร ) และรัศมีของถังน้ำมัน ( หน่วยเป็นเซนติเมตร ) สำหรับความสูงและรัศมีของถังน้ำมัน จะนำไปใช้ในการคำนวนณหาปริมาตรของน้ำมันและแสดงระดับน้ำมันในกราฟ

**3.1.1.2 Response**

เป็นส่วนที่แสดงผลจากการวัดที่ทำการคำนวณออกมาได้ให้กับผู้ใช้งานโปรแกรม

**ระยะห่างเซนเซอร์** บอกถึง ระยะห่างจากระดับผิวน้ำมันในถังกับเซนเซอร์( Ultra Sonic ) หน่วยเป็นเซนติเมตร

**ความสูง** บอกถึง ระดับความสูงของน้ำมันในถังน้ำมันขณะที่ทำการวัด หน่วยเป็นเซนติเมตร

**ปริมาตร** บอกถึง ปริมาตรของน้ำมันในถังที่มีอยู่ขณะทำการวัดหน่วยเป็น ลูกบาศก์เซนติเมตร

**3.1.1.3 Test**

ส่วนนี้ เมื่อผู้ใช้งานโปรแกรมต้องการตรวจสอบถังน้ำมัน และกรอกข้อมูลลงในส่วนของ Setting เรียบร้อยแล้ว ให้ทำการคลิกเมาส์ที่ปุ่ม Connect โปรแกรมจะส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ และรอรับ ผลจากการตรวจสอบ ตามคาบเวลาที่ทำการโปรแกรมไว้ เพื่อนำมาประมวลผลและแสดงผลให้กับผู้ใช้งานโปรแกรม เมื่อต้องการหยุดการตรวจสอบหรือต้องการตรวจสอบถังน้ำมันอื่น ให้คลิกเมาส์ที่ปุ่ม Disconnect และ คลิกเมาส์ที่ปุ่ม Connect ครั้งเมื่อต้องการตรวจสอบในครั้งต่อไป

**3.1.1.4 Alert**

เมื่อโปรแกรมได้รับผลการตรวจสอบต่างๆจากไมโครคอนโทรลเลอร์เรียบร้อยแล้ว จะทำการประมวลผล และ แสดงสถานะการทำงานของส่วนต่างๆ แบ่งเป็น 3 ส่วน คือ

**ไม่ตอบสนอง** เมื่อคลิกเมาส์ที่ปุ่ม Connect โปรแกรมจะพยายามทำการติดต่อกับคอนโทรลเลอร์ที่ระบุหมายเลขไว้ตลอดเวลา เหมือนกับการสร้างช่องทางการติดต่อขึ้น เมื่อใดก็ตามที่โปรแกรมไม่สามารถทำการเชื่อมต่อได้เป็นระยะเวลาหนึ่ง จะมีสัญลักษณ์วงกลมสีแดง เป็นการบอกให้รู้ว่าไม่สามารถติดต่อกับหมายเลขถังที่ระบุยังปลายทางได้ และจะพยายามทำการเชื่อมต่อตลอดเวลาถ้าทำการเชื่อมต่อสำเร็จ สัญลักษณ์วงกลมสีแดงจะหายไป

สภาวะนิ่ง ขณะที่ทำการวัดระดับน้ำมัน ระบบสูบน้ำมันเข้าถัง ยังคงทำงานอยู่ตลอดเวลาหมายความว่า ระดับน้ำมันต้องเกิดการเปลี่ยนแปลงตลอดเวลาหรือไม่อยู่ในสภาวะนิ่ง เนื่องจากผิวหน้าของน้ำมัน ต้องเกิดการเคลื่อนไหวตามการสูบน้ำมันเข้าถัง ถ้าทำการวัดระดับน้ำมันแล้วระดับน้ำมันไม่เกิดการเปลี่ยนแปลงหรืออยู่นิ่ง เป็นระยะเวลา 5 วินาที จะมีสัญลักษณ์วงกลมสีแดงเป็นการบอกว่าเกิดความผิดปกติกับระดับน้ำมันในถัง และสัญลักษณ์นี้จะหายไปเมื่อ ระดับน้ำมันไม่อยู่ในสภาวะนิ่ง หรือมีการเคลื่อนตัวอีกครั้ง

**Sucker Rod** sucker rod เป็นส่วนที่ทำหน้าที่ปั๊มน้ำมันเข้าไปยังถังพัก ถ้ายังคงมีการทำงานเป็นปกติ หรือ ทำการ ปั๊มน้ำมันตลอดเวลา จะมีสัญลักษณ์เป็นวงกลมสีเขียวแสดงว่ามีการทำงานเป็นปกติ แต่ถ้าส่วนนี้หยุดทำงาน เป็นระยะเวลา 5 วินาที ( ตามที่ตั้งเวลาไว้ในโปรแกรม ) จะมีสัญลักษณ์สีแดง ขึ้นมาแทนสีเขียว และ จะกลับเป็นสีเขียว เมื่อ sucker rod ทำงานอีกครั้ง

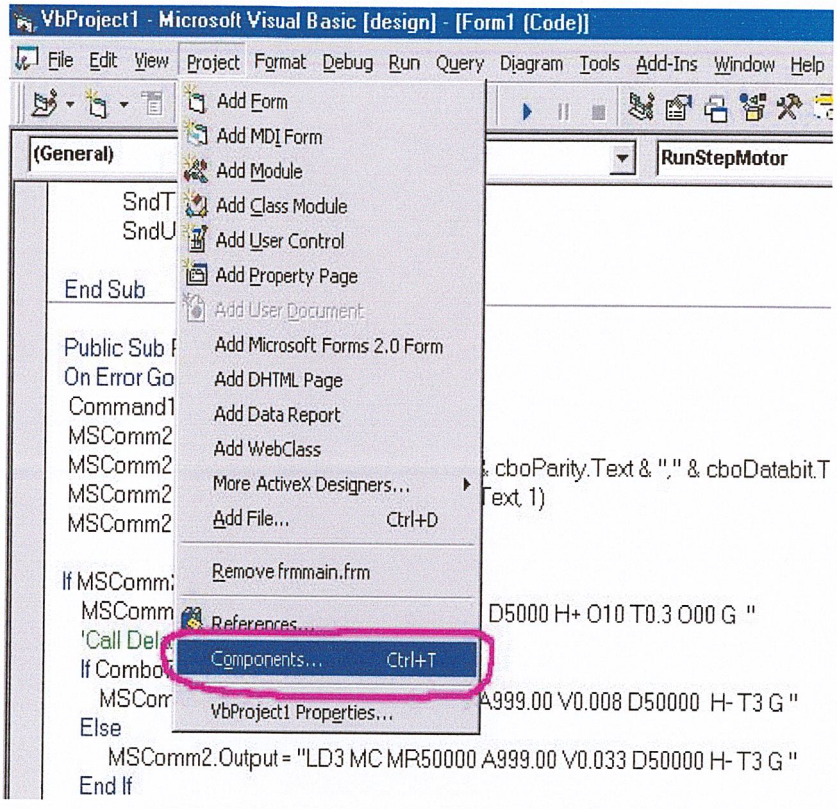
**3.1.1.5 Tank**

เป็นภาพแสดงระดับน้ำมันในถังเมื่อเทียบกับความสูงของถัง ที่ผู้ใช้ป้อนข้อมูลเข้ามา

### 3.1.2 วิชวลเบสิคเขียนโปรแกมติดต่อ I/O พอร์ตอนุกรม

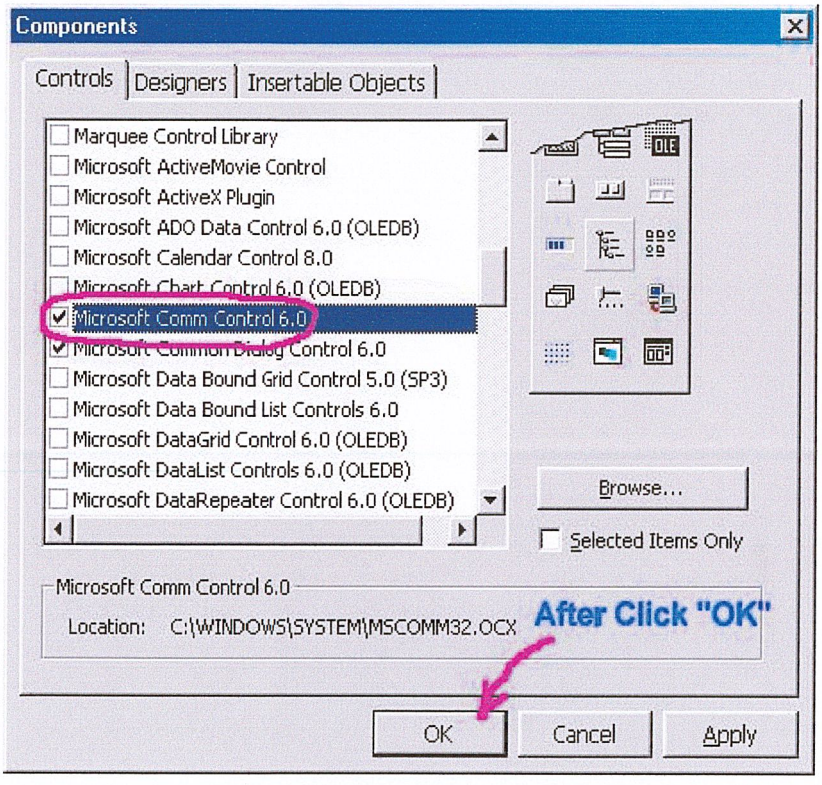
สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่คุณต้อง กำหนด Custom Control เข้าไปที่ เมนู Project--->Companents แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม(ที่มาจาก[1])

1.ขั้นตอนแรก เลือกที่เมนูบาร์ด้านบนของโปรแกรม **Visual Basic** ดังรูปด้านล่าง



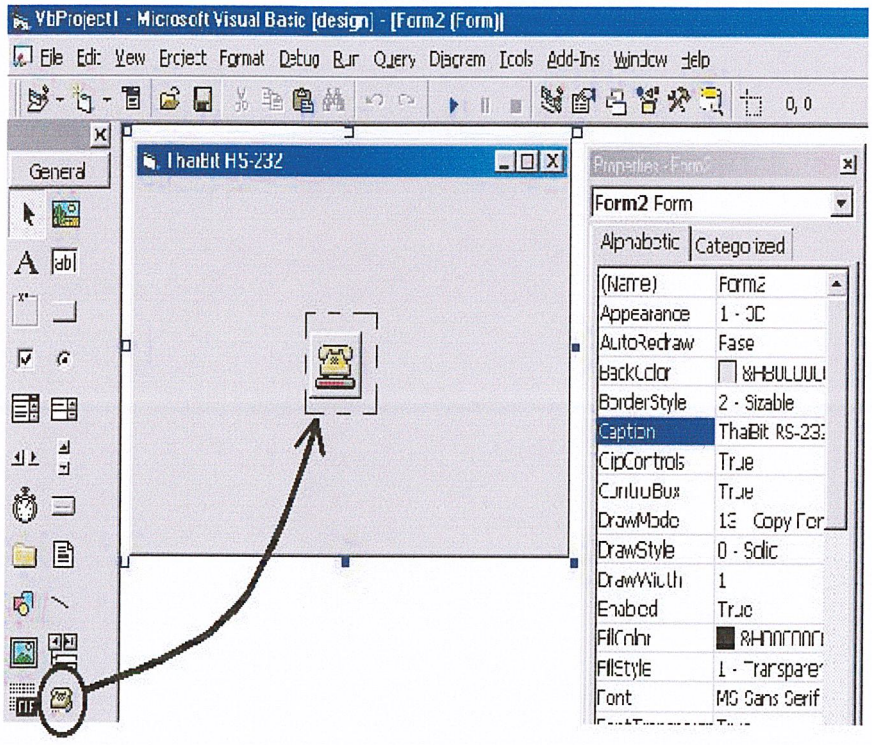

รูปที่ 3.2 ขั้นตอนที่ 1 การเขียนโปรแกรมติดต่อ I/O Serial Port (ที่มาจาก[5])

**2. ขั้นที่ 2 เลือกชื่อ Control ชื่อ Microsoft Comm Control 6 ดังรูปด้านล่าง**




รูปที่ 3.3 ขั้นตอนที่ 2 การเขียนโปรแกรมติดต่อ I/O Serial Port (ที่มาจาก[5])

**3. ขั้นที่สาม ลาก Control ชื่อ Microsoft Comm จาก ToolBox มาไว้บน Form ดังรูปด้านล่าง**




รูปที่ 3.4 ขั้นตอนที่ 3 การเขียนโปรแกรมติดต่อ I/O Serial Port (ที่มาจาก[5])

### 3.1.3 การเขียนโปรแกรมติดต่อกับ Serial Port

สามารถทำได้ 2 วิธี (ที่มาจาก[1]) คือ

**1.การติดต่อแบบอินเตอร์รัพต์**

ขบวนการอินเตอร์รัพท์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

**2.การติดต่อแบบโพลลิ่ง**

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า"Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ" Round Robin" แทน แต่ในVBเราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพต์ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive,comEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ

**3.1.4 องค์ประกอบในการใช้ MSComm**

การตั้งค่าติดต่อกับพอร์ต

- ComPort คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)รายละเอียดดูในเมนูด้านซ้ายSerialPortDetail
- Setting คือ เราต้องกำหนดอัตรา Baud,Parity,Data(จำนวนบิต(,Stop ตัวอย่าง 1200,n,8,1 เป็นต้น
- HandShaking คือ เราจะกำหนดได้ 4 แบบ 1.comNone 2.comXonXoff 3. comRTS 4.comTRSXonXoff

**3.1.5 การใช้ Buffer ในการรับส่งข้อมูล**

-InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา

-OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป

- Rthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา

- Sthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- Inputlen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- EOFEnable คือ การที่บอกว่าสิ้นสุดของไฟล์(EOF) End of File

**3.1.6ด้านฮาร์ดแวร์**

-ParityReplace คือ ค่าของคาเลกเตอร์ที่จะแทนในเมื่อเกิด Parity Error
-NullDiscard คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
-RTSEnableคือ ทำให้มีสัญญาณ RTS (Request To Send)
- DTSEnableคือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

**3.1.6.1 การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้**

1.Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน การเขียนโค้ดตัวอย่าง MSComm1.CommPort=1 ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมฯ

2. Property ชื่อ Settingsคือ การตั้งค่า ของการรับ-ส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่า อัตราบอดเรท ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่า ดังนี้ MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช้ใส่ N,จำนวนบิตข้อมูล,บิตสต๊อบ" การเขียนโค้ดกัน ตัวอย่าง MSComm1.Settings="1200,N,8,1"

3. Property ชื่อ InputLen คือ กำหนดขนาด ขณะที่มีข้อมูลเข้ามา ให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัพเฟอร์ การเขียนโค้ด ตัวอย่าง MSComm1.InputLen=1

4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด= True ถ้าปิด =False การเขียนโค้ด ตัวอย่าง MSComm1.PortOpen=True

5. Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล)Comport) ทำให้เกิด CommEvent ใน OnComm Event การเขียนโค้ดตัวอย่าง MSComm1.Rthreshold =1

จากรายละเอียดที่กล่าวมา เราจะมาเขียนใน โพรซีเดอร์ VB ซึ่งจะไว้ที่ Sub Form_Load() หรือจะสร้าง Sub ขึ้นใหม่ในกรณีที่จะเรียกใช้ภายหลัง

```
Private Sub Form_Load()
    MSComm1.Settings="1200,N,8,1"
    MSComm1.CommPort=1
    MSComm1.InputLen=1
    MSComm1.PortOpen=True
    MSComm1.Rthreshold =1
End Sub
```

### 3.1.6.2 วิธีของการรับส่งข้อมูลจาก Serial Port

จากวิธีเขียนโค้ดด้านบน เป็นการกำหนดค่าเริ่มต้น ให้กับคอมพอร์ต และเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้น ก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้ Output = ‘ ‘ ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต Input = ‘ ‘ เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS232 นั้นเอง

บทต่อไปจะกล่าวถึง การทดลองโครงงานนี้ จุดประสงค์ของการทดลอง และผลที่ได้จากการทดลอง มี ผลที่ได้จากการส่งข้อมูลในรูปแบบของ FSK การวัดสัญญาณของอุปกรณ์ Ultra Sonicและการทดสอบการทำงานของโปรแกรม

# บทที่ 4

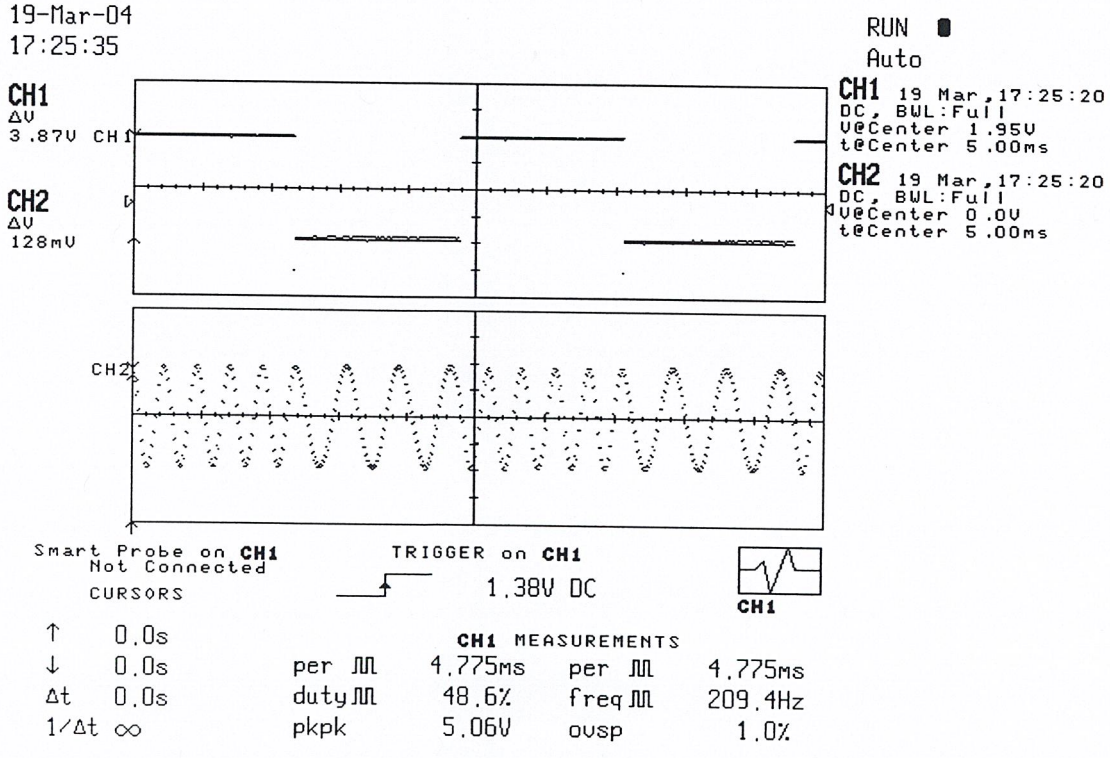
## ผลการทดลอง

บทนี้กล่าวถึงการทดลองต่างๆที่ได้ทำในโครงงานนี้ โดยแบ่งเป็น 3 หัวข้อการทดลอง คือ การทดลองการส่งข้อมูลด้วย FSK การทดลองสัญญาณของอุปกรณ์ Ultra Sonic และ การทดสอบการทำงานของโปรแกรม ว่ามีผลการทำงานเป็นเช่นไร

### 4.1 การทดลอง RDL 200T

**จุดประสงค์การทดลอง**

1. ทดสอบการทำงานของอุปกรณ์ RDL 200T
2. ตรวจสอบการทำงานที่ได้ว่าเป็นไปตามต้องการหรือไม่




รูปที่ 4.1 กราฟสัญญาณ FSK

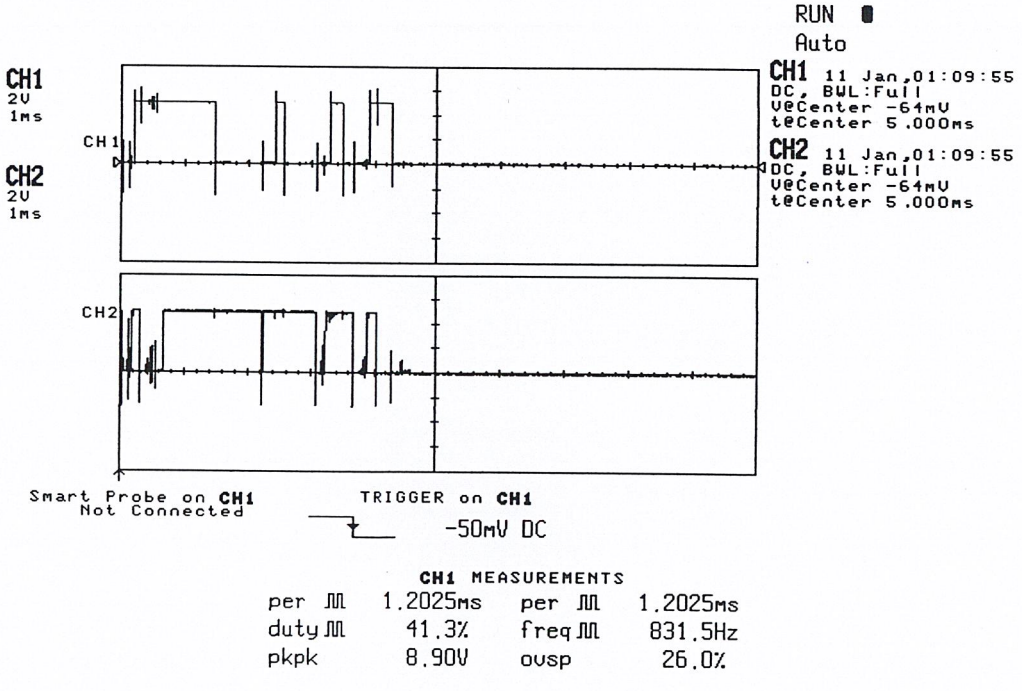
จากรูปที่ 4.1 CH2 เป็นสัญญาณ อินพุท ที่เป็น ไบนารี ของวงจร RLD200T และ CH1 เป็นสัญญาณ เอาท์พุทที่ได้ ( FSK )

เมื่อป้อนสัญญาณอินพุทเป็น 1 และ 0 สลับกัน สัญญาณที่ได้หลังจากการ Modurate สัญญาณอินพุท จะเป็น สัญญาณ FSK คือ ข้อมูลที่เป็น 1 จะมีความถี่ของสัญญาณมากกว่าข้อมูลที่มีค่าเป็น 0

## 4.2 การทดลองอุปกรณ์ Ultra Sonic

**จุดประสงค์การทดลอง**

1. ตรวจสอบการทำงานของอุปกรณ์



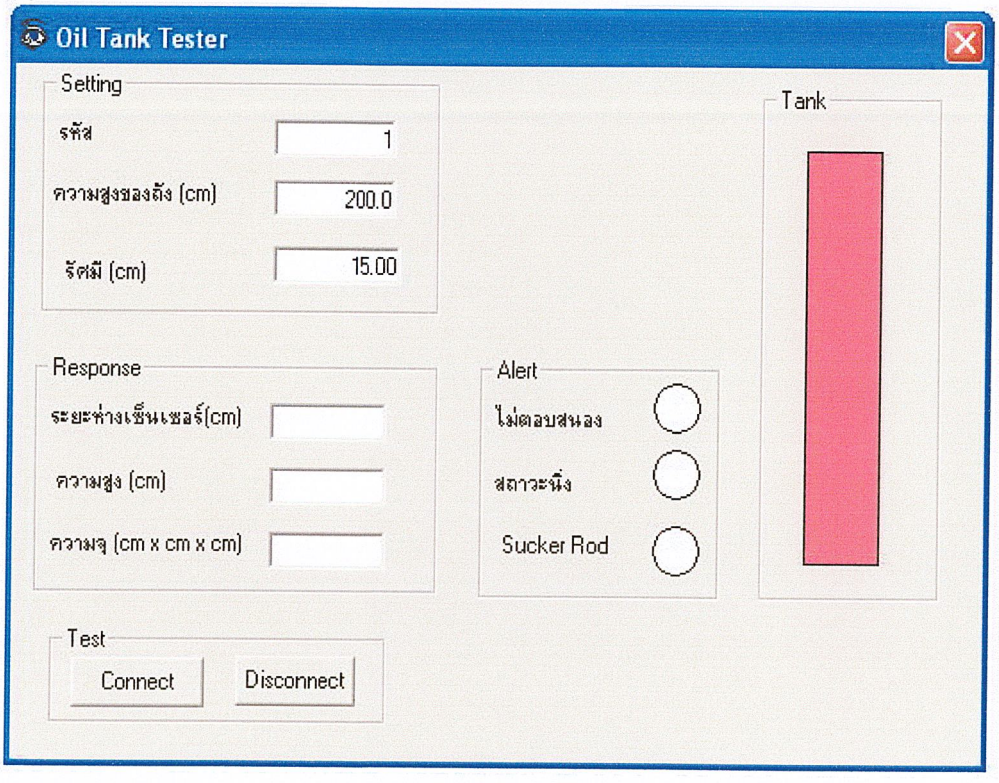

รูปที่ 4.2 กราฟการทำงานของอุปกรณ์ Ultra Sonic

รูปที่ 4.2 CH1 เป็นสัญญาณ ที่ขา Echo ของอุปกรณ์ Ultra Sonic เมื่อมีการป้อนสัญญาณพัลซ์(Pulse) เข้าที่ขา Triggler และเกิดการสะท้อนกลับ โดยที CH2 เป็นสัญญาณ ที่ป้อนมายังขา Triggler ของอุปกรณ์ Ultra Sonic

## 4.3 ทดลองการทำงานของโปรแกรม

**จุดประสงค์การทดลอง**

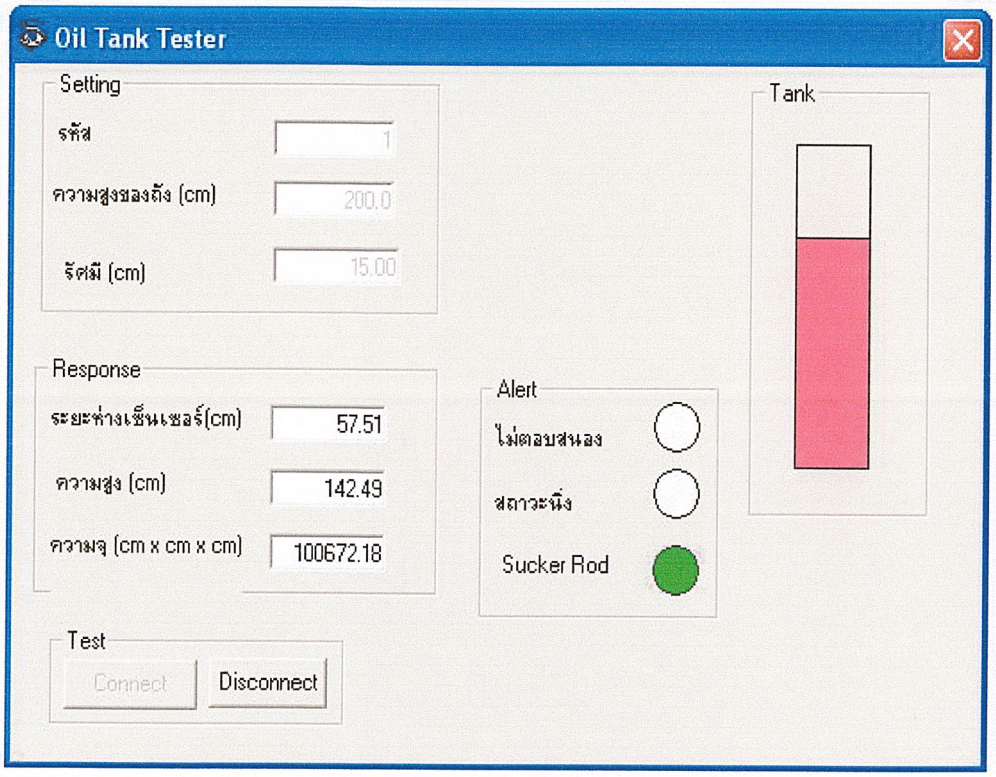
1. ทดลองการทำงานของโปรแกรมในการวัดระดับ
2. ทดลองการแจ้งเตือนเมื่อพบสิ่งผิดปกติ




รูปที่ 4.3 โปรแกรมติดต่อและควบคุมไมโครคอลโทรลเลอร์

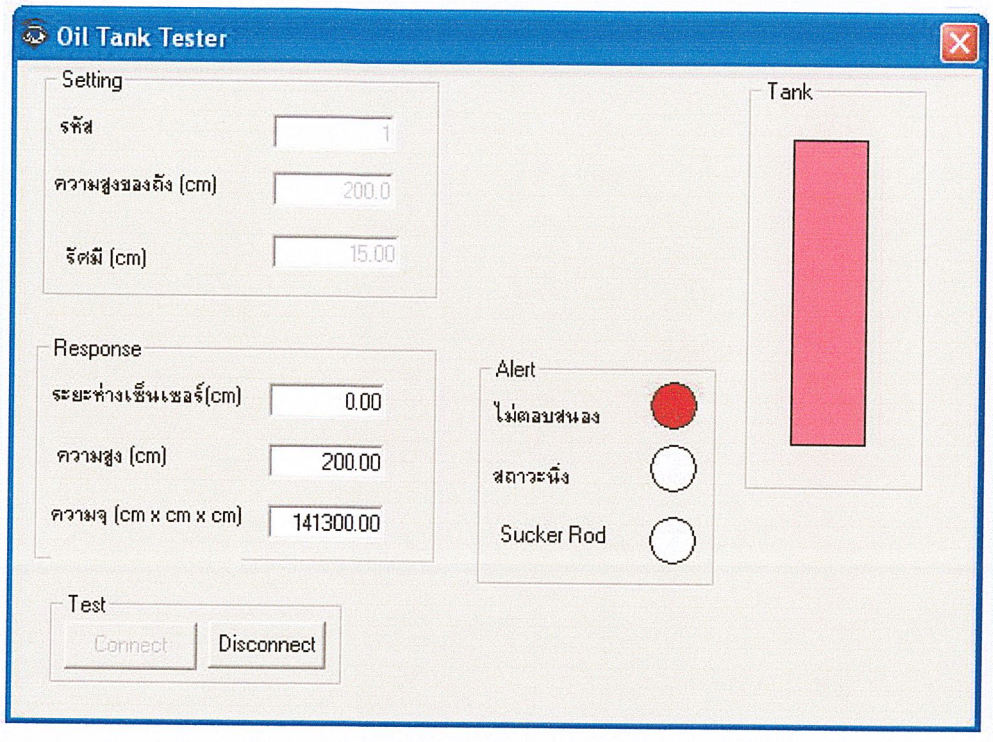
รูปที่ 4.3 เป็นหน้าต่างโปรแกรมเมื่อทำการเปิดโปรแกรมขึ้นมา ผู้ใช้งานโปรแกรม กรอกข้อมูลในส่วนของ Setting โดยระบุ หมายเลขถังน้ำมันที่ต้องการตรวจสอบ ( หมายเลข 1 ) ความสูงของถัง

( 200 ) รัศมีของถัง ( 15 ) ตามตัวอย่าง จากนั้นทำการ คลิกเมาส์ที่ปุ่ม Connect และรอผลการทำงานของโปรแกรม



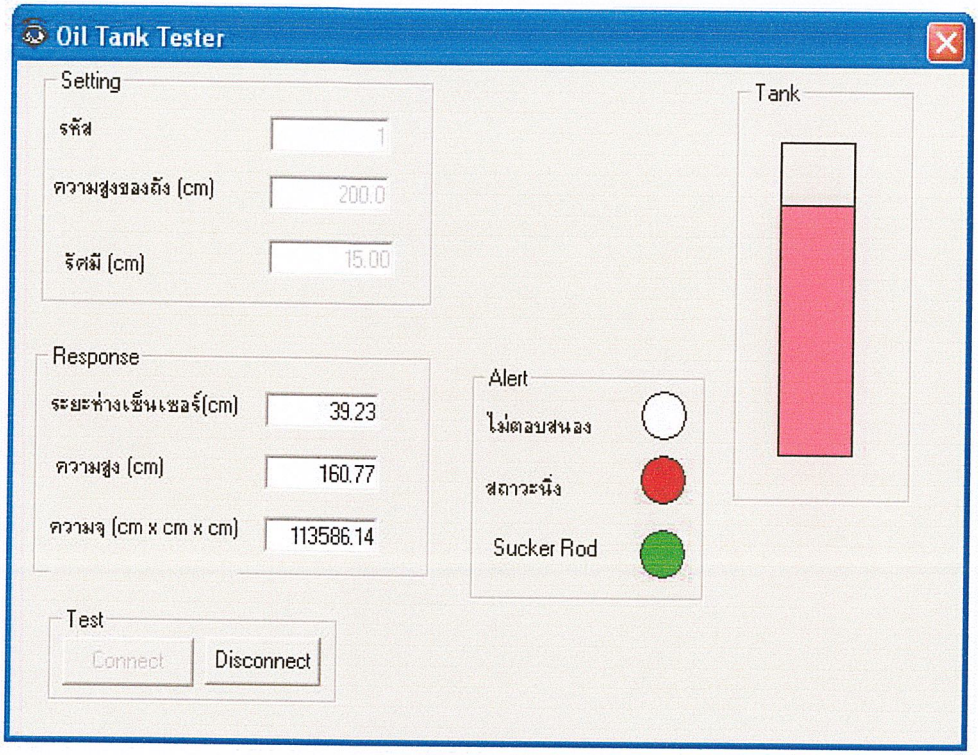

รูปที่ 4.4 หน้าจอการแสดงผ่านการวัดระดับน้ำมันที่ถัง ID 01

หน้าจอมีการแสดงผลของ ถังน้ำมันหมายเลข 1 แสดงสถานะของมอเตอร์ เป็น วงกลมสีเขียวหมายความว่า ในส่วนของ Sucker Rod มีการทำงานเป็นปกติ และในส่วนของการเตื่อนไม่ตอบสนอง และสภาวะนิ่ง ไม่มีการแจ้งเตือน การทำงานของระบบปั๊มน้ำมันยังคงเป็นปกติ ในส่วนของ Response จะแสดงผลของ ระยะห่างจาก เซนเซอร์ถึงระดับผิวน้ำมันในถัง คือ 57.51 เซนติเมตร ได้ความสูงของระดับน้ำมันในถังเป็น 142.49 เซนติเมตร และมีปริมาตร เป็น 100,672.18 ลูกบาศก์เซนติเมตร



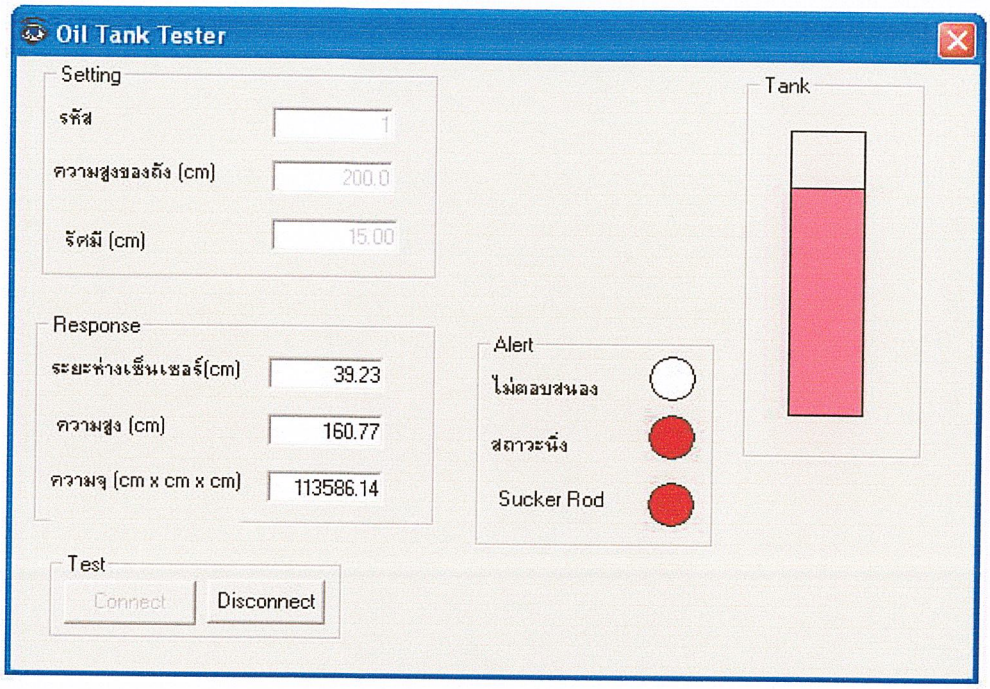

รูปที่ 4.5 โปรแกรมการทำงานที่ไม่มีการตอบสนองของมโครคอนโทรลเลอร์

รูปที่ 4.5 ปรากฏสีแดงขึ้นในช่อง ไม่ตอบสนอง แสดงว่าไม่สามารถทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ ที่ระบุได้ และยังสามารถวิเคราะห์ได้อีกว่าอาจเกิดจาก กระแสไฟฟ้า ที่จ่ายให้กับถังน้ำมันขัดข้อง หรือ ชุดอุปกรณ์ตรวจสอบเกิดการขัดข้อง




รูปที่ 4.6 การทำงานของโปรแกรมเมื่อไม่มีการเปลี่ยนแปลงระดับน้ำมัน

รูปที่ 4.6 สภาวะนิ่งเป็นสีแดง และ Sucker Rod เป็นสีเขียว หมายถึง sucker rod ทำงานเป็นปกติ แต่ระดับน้ำมันในถังคงที่ สามารถวิเคราะห์ได้ว่า ท่อส่งน้ำมันดิบ จาก sucker rod ไปยังถังน้ำมันผิดปกติ




รูป ที่ 4.7 การทำงานของ โปรแกรมเมื่อ sucker rod มีปัญหา และระดับน้ำมันไม่เปลี่ยนแปลง

รูปที่4.7 เป็นการแจ้งบอกว่า ขณะนี้ระดับน้ำมันในถังไม่เปลี่ยนแปลง และ sucker rod ไม่ทำงาน ทำให้รู้ได้ว่า sucker rod เกิดความผิดปกติขึ้น ไม่ทำการปั๊มและจ่ายน้ำมันไปยังถังน้ำมัน

# บทที่ 5
# สรุปผล ปัญหาที่พบ และข้อเสนอแนะ

## 5.1 สรุปผล

ได้ศึกษาการทำงานของระบบไมโครคอนโทรลเลอร์ที่มีการเชื่อมต่อกับคอมพิวเตอร์เพื่อใช้ในการสั่งการควบคุม และสื่อการระหว่างกันในหารทำงาน พื้นฐานทางด้านอิเลคโทรนิคและระบบการสื่อสารไร้สาย จากการทดลอง การสื่อสารส่งข้อมูลระหว่างคอมพิวเตอร์และ ไมโครคอนโทรลเลอร์เป็นอย่างดีแล้ว สามารถควบคุมการทำงานของอัลตราโซนิคให้ทำงานตามที่ต้องการได้ แต่ปัญหานั้นอยู่ที่ระบบสื่อสารที่ไม่สามารถส่งข้อมูลในอัตราที่สูงๆได้ ซึ่งไม่เป็นปัญหาเนื่องจากอัตราความเร็วในการส่งข้อมูลอยู่ที่ 1200 บิท / วินาทีก็เพียงพอแล้ว ปัญหาอีกอย่างหนึ่งก็คือ ปัญกาที่เกิดจากความผิดพลาดของอุปกรณ์อัลตราโซนิค ในทางทฤษฎีนั้นคลื่นสามารถแทรกสอด กระจัดกระจาย และเลี้ยวเบนได้ ทำให้คลื่นที่สะท้อนกลับมานั้น มีหลายทิศทาง และ มาไม่พร้อมกันทำให้เกิดข้อผิด-พลาดขึ้น อีกทั้งยังมีบีมที่กว้าง ถึง 30 องศา ทำให้ค่าระดับน้ำมันที่ได้นั้นไม่ตรงตามความเป็นจริงทั้งหมด แต่ค่าความคลาดเคลื่อนที่เกิดขึ้น มีน้อยมาก ( น้อยกว่า 10 เซนติเมตร) เมื่อเทียบกับระดับความสูงของ บ่อน้ำมัน ( 6 เมตร ) ที่เป็นต้นแบบของโครงงานนี้

การใช้ระบบตรวจสอบการทำงานในการช่วยดูแลการขุดเจาะน้ำมัน ทำให้เกิดความง่ายในการทำงานและการดูแลมากขึ้น อีกทั้งยังช่วยในการวิเคราะห์ถึงปัจจัยต่างๆที่อาจขึ้นได้ ทำให้ลดระยะเวลาการตรวจสอบ รวมถึงลดการสูญเสียและค่าใช้จ่ายที่ตามมาเป็นจำนวนมาก เนื่องจากว่าน้ำมันดิบมีมูลค่าสูง และเป็นทรัพยากรที่มีค่า

## 5.2 ปัญหาที่พบ

-อุปกรณ์ Ultra Sonic เสียง่าย เมื่อนำมาใช้งานตรวจสอบได้ยาก

-การส่งข้อมูล ไปกับคลื่นวิทยุ เกิดสัญญาณรบกวนง่าย ทำให้ข้อมูลที่ทำการส่งเกิดความผิดพลาดขึ้น ยังผู้รับ ดังนั้น ผู้ส่ง ต้องใช้ ช่วงคลื่นความที่ ที่เกิดความรบกวนน้อย และมี Guard Band มากพอที่จะทำการส่งข้อมูล

- อุปกรณ์ Ultra Sonic แต่ละตัวมีการตอบสนอง แตกต่างกัน ในการทำการวัดระดับน้ำที่ความสูงเดียวกัน ค่าที่คำนวณได้ จากอุปกรณ์แต่ละตัว มีความแตกต่างกัน ดังนั้นในการนำไปใช้งานจริงควร

ทำการตรวจสอบการตอบสนองของ อุปกรณ์ Ultra Sonic แต่ละตัวก่อนว่ามีความใกล้เคียงกันและถูกต้อง

**5.3 ข้อเสนอแนะ**

การที่ใช้การติดต่อสื่อสารแบบไร้สาย โดยใช้เครื่องรับส่งวิทยุผ่าน Packet Radio เป็น แนวคิดที่ประหยัดในการติดต่อสื่อสาร แต่ก็อาจจะไม่ได้สัญญาณที่มีความแน่นอนอาจจะมีความผิดพลาดได้ง่ายจึงไม่ควรนำไปใช้กับ งานที่มีความแน่นอนสูงๆ และ ต้องการความถูกต้องของข้อมูลสูง

# บรรณานุกรม


[1] อรรถพล บุษยะโภคา วรพจน์ วัฒนกุล ชัยวัฒน์ ลิ้มพรจิตรวิไล **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม** บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด ฉบับปรับปรุงครั้ง 2

[2] กิตติ ภักดีวัฒนะ จำลอง ครูอุตสาหะ **Visual Basic 6 ฉบับโปรแกรมเมอร์** กรุงเทพมหานคร บริษัทเคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด พิมพ์ครั้งที่ 10

[3] วิชเยนทร์ ตรีธารทิพย์วิกุล **"Packet Radio Handbook"**

[4] วรพจน์ กรแก้ววัฒนกุล ชัยวัฒน์ ลิ้มพรจิตรวิไล **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฉบับ AT89C5x ของ Atmel** กรุงเทพมหานคร: บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

[5] การเขียนโปรแกรมติดต่อกับอุปกรณ์ภายนอก **http://www.thaiio.com**

[6] SRF04 Ultrasonic Ranger Finder **http://www.robot-electronics.co.uk**

[an error occurred while processing this directive]

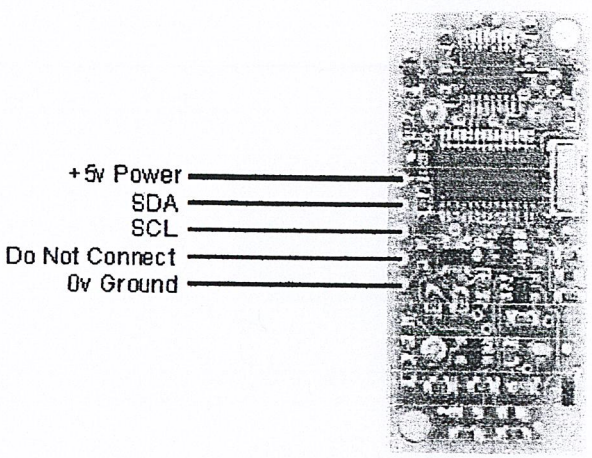
# SRF08 Ultra sonic range finder

**Technical Specification**

Communication with the SRF08 ultrasonic rangefinder is via the I2C bus. This is available on popular controllers such as the OOPic and Stamp BS2p, as well as a wide variety of micro-controllers. To the programmer the SRF08 behaves in the same way as the ubiquitous 24xx series eeprom's, except that the I2C address is different. The default shipped address of the SRF08 is 0xE0. It can be changed by the user to any of 16 addresses E0, E2, E4, E6, E8, EA, EC, EE, F0, F2, F4, F6, F8, FA, FC or FE, therefore up to 16 sonar's can be used. In addition to the above addresses, all sonar's on the I2C bus will respond to address 0 - the General Broadcast address. This means that writing a ranging command to I2C address 0 (0x00) will start all sonar's ranging at the same time. This should be useful in ANN Mode (See below). The results must be read individually from each sonar's real address. We have examples of using the SRF08 module with a wide range of popular controllers.

**Connections**

The "Do Not Connect" pin should be left unconnected. It is actually the CPU MCLR line and is used once only in our workshop to program the PIC16F872 on-board after assembly, and has an internal pull-up resistor. The SCL and SDA lines should each have a pull-up resistor to +5v somewhere on the I2C bus. You only need one pair of resistors, not a pair for every module. They are normally located with the bus master rather than the slaves. The SRF08 is always a slave - never a bus master. If you need them, I recommend 1.8k resistors. Some modules such as the OOPic already have pull-up resistors and you do not need to add any more.




**Registers**

The SRF08 appears as a set of 36 registers.

| Location | Read | Write |
|---|---|---|
| 0 | Software Revision | Command Register |
| 1 | Light Sensor | Max Gain Register (default 31) |
| 2 | 1st Echo High Byte | Range Register (default 255) |
| 3 | 1st Echo Low Byte | N/A |
| ~~~~ | ~~~~ | ~~~~ |
| 34 | 17th Echo High Byte | N/A |
| 35 | 17th Echo Low Byte | N/A |

Only locations 0, 1 and 2 can be written to. Location 0 is the command register and is used to start a ranging session. It cannot be read. Reading from location 0 returns the SRF08 software revision. By default, the ranging lasts for 65mS, but can be changed by writing to the range register at location 2. If you do so, then you will likely need to change the analogue gain by writing to location 1. See the **Changing Range** and **Analogue Gain** sections below.

Location 1 is the onboard light sensor. This data is updated every time a new ranging command has completed and can

be read when range data is read. The next two locations, 2 and 3, are the 16bit unsigned result from the latest ranging - high byte first. The meaning of this value depends on the command used, and is either the range in inches, or the range in cm or the flight time in uS. A value of zero indicates that no objects were detected. There are up to a further 16 results indicating echo's from more distant objects.

**Commands**

The are three commands to initiate a ranging (80 to 82), to return the result in inches, centimeters or microseconds. There is also an ANN mode (Artificial Neural Network) mode which is described later and a set of commands to change the I2C address.

| Command | | Action |
|---|---|---|
| Decimal | Hex | |
| 80 | 0x50 | Ranging Mode - Result in inches |
| 81 | 0x51 | Ranging Mode - Result in centimeters |
| 82 | 0x52 | Ranging Mode - Result in micro-seconds |
| | | |
| 83 | 0x53 | ANN Mode - Result in inches |
| 84 | 0x54 | ANN Mode - Result in centimeters |
| 85 | 0x55 | ANN Mode - Result in micro-seconds |
| | | |
| 160 | 0xA0 | 1st in sequence to change I2C address |
| 165 | 0xA5 | 3rd in sequence to change I2C address |
| 170 | 0xAA | 2nd in sequence to change I2C address |

**Ranging Mode**

To initiate a ranging, write one of the above commands to the command register and wait the required amount of time for completion and read as many results as you wish. The echo buffer is cleared at the start of each ranging. The first echo range is placed in locations 2,3. the second in 4,5, etc. If a location (high and low bytes) is 0, then there will be no further reading in the rest of the registers. The default and recommended time for completion of ranging is 65mS, however you can shorten this by writing to the range register before issuing a ranging command. Light sensor data at location 1 will also have been updated after a ranging command.

**ANN Mode**

ANN mode (Artificial Neural Network) is designed to provide the multi echo data in a way that is easier to input to a neural network, at least I hope it is - I've not actually done it yet. ANN mode provides a 32 byte buffer (locations 4 to 35 inclusive) where each byte represents the 65536uS maximum flight time divided into 32 chunks of 2048uS each - equivalent to about 352mm of range. If an echo is received within a bytes time slot then it will be set to no-zero, otherwise it will be zero. So if an echo is received from within the first 352mm, location 4 will be non-zero. If an object is detected 3m away the location 12 will be non-zero (3000/352 = 8) (8+4=12). Arranging the data like this should be better for a neural net than the other formats. The input to your network should be 0 if the byte is zero and 1 if its non-zero. I have a SOFM (Self Organizing Feature Map) in mind for the neural net, but will hopefully be useful for any type.

| Location 4 | Location 5 | Location 6 | Location 7 | Locations 8 - 35 |
|---|---|---|---|---|
| 0 - 352mm | 353 - 705mm | 706 - 1057mm | 1058 - 1410mm | and so on |

Locations 2,3 contain the range of the nearest object converted to inches, cm or uS and is the same as for Ranging Mode.

**Checking for Completion of Ranging**

You do not have to use a timer on your own controller to wait for ranging to finish. You can take advantage of the fact that the SRF08 will not respond to any I2C activity whilst ranging. Therefore, if you try to read from the SRF08 (we use the software revision number a location 0) then you will get 255 (0xFF) whilst ranging. This is because the I2C data line (SDA) is pulled high if nothing is driving it. As soon as the ranging is complete the SRF08 will again respond to the I2C bus, so just keep reading the register until its not 255 (0xFF) anymore. You can then read the sonar data. Your controller can take advantage of this to perform other tasks while the SRF08 is ranging.

**Changing the Range**

The maximum range of the SRF08 is set by an internal timer. By default, this is 65mS or the equivalent of 11 metres of range. This is much further than the 6 metres the SRF08 is actually capable of. It is possible to reduce the time the SRF08 listens for an echo, and hence the range, by writing to the range register at location 2. The range can be set in steps of about 43mm (0.043m or 1.68 inches) up to 11 metres.

The range is ((Range Register x 43mm) + 43mm) so setting the Range Register to 0 (0x00) gives a maximum range of

43mm. Setting the Range Register to 1 (0x01) gives a maximum range of 86mm. More usefully, 24 (0x18) gives a range of 1 metre and 140 (0x8C) is 6 metres. Setting 255 (0xFF) gives the original 11 metres (255 x 43 + 43 is 11008mm).
There are two reasons you may wish to reduce the range.
1. To get at the range information quicker
2. To be able to fire the SRF08 at a faster rate.
If you only wish to get at the range information a bit sooner and will continue to fire the SRF08 at 65ms of slower, then all will be well. However if you wish to fire the SRF08 at a faster rate than 65mS, you will definitely need to reduce the gain - see next section.
The range is set to maximum every time the SRF08 is powered-up. If you need a different range, change it once as part of your system initialization code.

**Analogue Gain**
The analogue gain register sets the *Maximum* gain of the analogue stages. To set the maximum gain, just write one of these values to the gain register at location 1. During a ranging, the analogue gain starts off at its minimum value of 94. This is increased at approx. 70uS intervals up to the maximum gain setting, set by register 1. Maximum possible gain is reached after about 390mm of range. The purpose of providing a limit to the maximum gain is to allow you to fire the sonar more rapidly than 65mS. Since the ranging can be very short, a new ranging can be initiated as soon as the previous range data has been read. A potential hazard with this is that the second ranging may pick up a distant echo returning from the previous "ping", give a false result of a close by object when there is none. To reduce this possibility, the maximum gain can be reduced to limit the modules sensitivity to the weaker distant echo, whilst still able to detect close by objects. The maximum gain setting is stored only in the CPU's RAM and is initialized to maximum on power-up, so if you only want do a ranging every 65mS, or longer, you can ignore the Range and Gain Registers.
*Note* - Effective in Ranging Mode only, in ANN mode, gain is controlled automatically.

| Gain Register | | Maximum Analogue Gain |
|---|---|---|
| Decimal | Hex | |
| 0 | 0x00 | Set Maximum Analogue Gain to 94 |
| 1 | 0x01 | Set Maximum Analogue Gain to 97 |
| 2 | 0x02 | Set Maximum Analogue Gain to 100 |
| 3 | 0x03 | Set Maximum Analogue Gain to 103 |
| 4 | 0x04 | Set Maximum Analogue Gain to 107 |
| 5 | 0x05 | Set Maximum Analogue Gain to 110 |
| 6 | 0x06 | Set Maximum Analogue Gain to 114 |
| 7 | 0x07 | Set Maximum Analogue Gain to 118 |
| 8 | 0x08 | Set Maximum Analogue Gain to 123 |
| 9 | 0x09 | Set Maximum Analogue Gain to 128 |
| 10 | 0x0A | Set Maximum Analogue Gain to 133 |
| 11 | 0x0B | Set Maximum Analogue Gain to 139 |
| 12 | 0x0C | Set Maximum Analogue Gain to 145 |
| 13 | 0x0D | Set Maximum Analogue Gain to 152 |
| 14 | 0x0E | Set Maximum Analogue Gain to 159 |
| 15 | 0x0F | Set Maximum Analogue Gain to 168 |
| 16 | 0x10 | Set Maximum Analogue Gain to 177 |
| 17 | 0x11 | Set Maximum Analogue Gain to 187 |
| 18 | 0x12 | Set Maximum Analogue Gain to 199 |
| 19 | 0x13 | Set Maximum Analogue Gain to 212 |
| 20 | 0x14 | Set Maximum Analogue Gain to 227 |
| 21 | 0x15 | Set Maximum Analogue Gain to 245 |
| 22 | 0x16 | Set Maximum Analogue Gain to 265 |
| 23 | 0x17 | Set Maximum Analogue Gain to 288 |
| 24 | 0x18 | Set Maximum Analogue Gain to 317 |
| 25 | 0x18 | Set Maximum Analogue Gain to 352 |
| 26 | 0x20 | Set Maximum Analogue Gain to 395 |

| | | |
|---|---|---|
| 27 | 0x21 | Set Maximum Analogue Gain to 450 |
| 28 | 0x22 | Set Maximum Analogue Gain to 524 |
| 29 | 0x23 | Set Maximum Analogue Gain to 626 |
| 30 | 0x24 | Set Maximum Analogue Gain to 777 |
| 31 | 0x25 | Set Maximum Analogue Gain to 1025 |

Note that the relationship between the Gain Register setting and the actual gain is not a linear one. Also there is no magic formula to say "use this gain setting with that range setting". It depends on the size, shape and material of the object and what else is around in the room. Try playing with different settings until you get the result you want. If you appear to get false readings, it may be echo's from previous "pings", try going back to firing the SRF08 every 65mS or longer (slower).
If you are in any doubt about the Range and Gain Registers, remember they are automatically set by the SRF08 to their default values when it is powered-up. You can ignore and forget about them and the SRF08 will work fine, detecting objects up to 6 metres away every 65mS or slower.

**Light Sensor**
The SRF08 has a light sensor on-board. A reading of the light intensity is made by the SRF08 each time a ranging takes place in either Ranging or ANN Modes ( The A/D conversion is actually done just before the "ping" whilst the +/- 10v generator is stabilizing). The reading increases as the brightness increases, so you will get a maximum value in bright light and minimum value in darkness. It should get close to 2-3 in complete darkness and up to about 248 (0xF8) in bright light. The light intensity can be read from the Light Sensor Register at location 1 at the same time that you are reading the range data.

**LED**
The red LED is used to flash out a code for the I2C address on power-up (see below). It also gives a brief flash during the "ping" whilst ranging.

**Changing the I2C Bus Address**
To change the I2C address of the SRF08 you must have only one sonar on the bus. Write the 3 sequence commands in the correct order followed by the address. Example; to change the address of a sonar currently at 0xE0 (the default shipped address) to 0xF2, write the following to address 0xE0; (0xA0, 0xAA, 0xA5, 0xF2 ). These commands must be sent in the correct sequence to change the I2C address, additionally, No other command may be issued in the middle of the sequence. The sequence must be sent to the command register at location 0, which means 4 separate write transactions on the I2C bus. When done, you should label the sonar with its address, however if you do forget, just power it up without sending any commands. The SRF08 will flash its address out on the LED. One long flash followed by a number of shorter flashes indicating its address. The flashing is terminated immediately on sending a command the SRF08.

| Address | | Long Flash | Short flashes |
|---|---|---|---|
| Decimal | Hex | | |
| 224 | E0 | 1 | 0 |
| 226 | E2 | 1 | 1 |
| 228 | E4 | 1 | 2 |
| 230 | E6 | 1 | 3 |
| 232 | E8 | 1 | 4 |
| 234 | EA | 1 | 5 |
| 236 | EC | 1 | 6 |
| 238 | EE | 1 | 7 |
| 240 | F0 | 1 | 8 |
| 242 | F2 | 1 | 9 |
| 244 | F4 | 1 | 10 |
| 246 | F6 | 1 | 11 |
| 248 | F8 | 1 | 12 |
| 250 | FA | 1 | 13 |
| 252 | FC | 1 | 14 |
| 254 | FE | 1 | 15 |

Take care not to set more than one sonar to the same address, there will be a bus collision and very unpredictable results.

**Current Consumption**
Average current consumption measured on our prototype is around 12mA during ranging, and 3mA standby. The module will automatically go to standby mode after a ranging, whilst waiting for a new command on the I2C bus. The actual measured current profile is as follows;

| Operation | Current | Duration |
|---|---|---|
| Ranging command received - Power on | 275mA | 3uS |
| +/- 10v generator Stabilization | 25mA | 600uS |
| 8 cycles of 40kHz "ping" | 40mA | 200uS |
| Ranging | 11mA | 65mS max |
| Standby | 3mA | indefinite |

The above values are for guidance only, they are not tested on production units.

You can have a look at the schematic and software

Your feedback/comments/criticisms and wish lists are very welcome, as always.
Regards, Gerry.
gerry@robot-electronics.co.uk