

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบแจ้งเหตุเพลิงไหม้ โดยแจ้งเตือนผ่านข้อความสั้น  
FIRE ALARM SYSTEM BY SHORT MESSAGE SERVICE



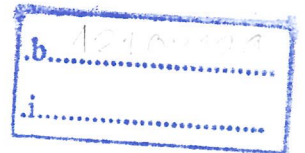
โดย

นาย ชีรคม ใจปลอด

นาย วิเศษฐ์ สถิตเกษมสานต์

นาย อภิรัตน์ เจียวศิริกุล

เลขหมู่.....  
เลขทะเบียน **104360**  
วัน,เดือน,ปี... - 2 พ.ย. 2552



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2551

ภาควิชา  
วิศวกรรมโทรคมนาคม  
ผ่านการตรวจแปลนแล้ว  
(ลงชื่อ).....ผู้ตรวจ

ระบบแจ้งเหตุเพลิงไหม้ โดยแจ้งเตือนผ่านข้อความสั้น  
**FIRE ALARM SYSTEM BY SHORT MESSAGE SERVICE**

โดย

|     |          |               |          |
|-----|----------|---------------|----------|
| นาย | ธีรคม    | ใจปลอด        | 48012022 |
| นาย | วิเชษฐ์  | สถิตเกษมสานต์ | 48012035 |
| นาย | อภิรัตน์ | เจียวศิริกุล  | 48012043 |

อาจารย์ที่ปรึกษา

ผศ. ดร. สุทธิชัย นพนาถิพงษ์

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2551

ปริญญาโทชั้นปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแจ้งเหตุเพลิงไหม้โดยแจ้งเตือนผ่านข้อความสั้น

**FIRE ALARM SYSTEM BY SHORT MESSAGE SERVICE**

**ผู้จัดทำ**

- |                              |          |
|------------------------------|----------|
| 1. นาย ชีรคม ใจปลอด          | 48012022 |
| 2. นาย วิเชษฐ์ สถิตเกษมสานต์ | 48012035 |
| 3. นาย อภิรัตน์ เจียวศิริกุล | 48012043 |



.....อาจารย์ที่ปรึกษา

( ผศ.ดร. สุทธิชัย นพนาถพงษ์ )

**ระบบแจ้งเหตุเพลิงไหม้ โดยแจ้งเตือนผ่านข้อความสั้น**  
**( FIRE ALARM SYSTEM BY SHORT MESSAGE SERVIC )**

|                           |          |
|---------------------------|----------|
| จัดทำโดย นาย ชีรคม ใจปลอด | 48012022 |
| นาย วิเชษฐ์ สถิตเกษมสานต์ | 48012035 |
| นาย อภิรัตน์ เขียวศิริกุล | 48012043 |

**อาจารย์ที่ปรึกษา ผศ.ดร. สุทธิชัย นพนาศิพงษ์**

**บทคัดย่อ**

โครงการนี้เป็นการประยุกต์ใช้ไมโครคอนโทรลเลอร์มาใช้ในการควบคุมระบบแจ้งเหตุเพลิงไหม้ ซึ่งในทุกวันนี้เราต้องพบกับภัยหลายรูปแบบ โดยเฉพาะภัยที่สร้างความเสียหายแก่ชีวิตและทรัพย์สิน เช่น การเกิดอุบัติเหตุจากเพลิงไหม้ จึงได้มีแนวคิดที่จะทำโครงการที่ใช้ในการเตือนภัยและแจ้งเหตุจากการเกิดเพลิงไหม้ขึ้น

เมื่อเกิดเหตุเพลิงไหม้ อุปกรณ์เซนเซอร์จะทำหน้าที่ในการตรวจจับควัน หรือสิ่งผิดปกติที่เกิดขึ้นและส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของระบบ โดยสั่งให้สัญญาณเตือนภัยดัง และส่งข้อความสั้น (SMS) ไปยังเลขหมายปลายทางที่ได้บันทึกไว้ เพื่อแจ้งให้ผู้รับข้อความทราบ ในกรณีที่เกิดเหตุไม่สามารถใช้ไฟฟ้าได้ ในอุปกรณ์จะมีการต่อแบตเตอรี่สำรองเข้ากับแหล่งจ่ายไฟเพื่อใช้ในกรณีฉุกเฉินได้

**ABSTRACT**

This project is presents microcontroller application use control fire alarm system. Now a day we meet many harm especially about life and property such as the fire accident. Therefore we have idea to create project about fire alarm system and inform.

When has fire the sensor equipment will smoke detects or abnormal that happen and send signal to microcontroller in order to control system. As the order give fire alarm loud and send Short Message Service (SMS) to record number for report the user know. bWhen circuit can not use electric current the equipment has connect backup battery for use them it has emergency.

# สารบัญ

| หัวข้อ   | หน้า |
|--|------|
| บทที่ 1 บทนำ   | 1    |
| บทที่ 2 ทฤษฎีและหลักการ  | 3    |
| 2.1 ระบบสื่อสาร  | 3    |
| 2.2 ความรู้พื้นฐานของการสื่อสารระบบไร้สาย                              | 4    |
| 2.3 ระบบคลื่นพาหะที่เป็นคลื่นวิทยุ (Radio Carrier)                     | 4    |
| 2.4 การมอดูเลตชัน (Modulation Technique)                               | 4    |
| 2.4.1 การมอดูเลตชันทางแอมพลิจูด (Amplitude Modulation)                 | 5    |
| 2.4.2 การมอดูเลตชันทางความถี่ (Frequency Modulation)                   | 9    |
| 2.5 ความถี่ข้างเคียงและดัชนีการมอดูเลตชัน                              | 10   |
| 2.6 เครื่องส่งสัญญาณคลื่นวิทยุ (Radio Transmitters)                    | 11   |
| 2.6.1 วงจรสัญญาณพาหะ (Carrier Generation)                              | 11   |
| 2.6.2 วงจรมอดูเลตสัญญาณ (Modulator)                                    | 11   |
| 2.6.3 วงจรขยาย (Amplifier)   | 11   |
| 2.6.4 วงจรอิมพีแดนซ์แมตชิ่ง (Impedance Matching Circuit)               | 13   |
| 2.6.5 เครื่องรับสัญญาณ (Communication Receives)                        | 13   |
| 2.7 ระบบสัญญาณ   | 14   |
| 2.7.1 สัญญาณระหว่างผู้เข้ากับชุมสาย (Subscriber Signaling)             | 14   |
| 2.7.2 สัญญาณติดต่อระหว่างชุมสายกับชุมสาย (Inter Exchange Signaling)    | 15   |
| 2.7.3 สัญญาณติดต่อกันระหว่างผู้เรียก (Calling) และผู้ถูกเรียก (Called) | 16   |
| 2.8 การทำงานของโทรศัพท์  | 16   |
| 2.9 ระบบโทรศัพท์แบบ DTMF   | 18   |
| 2.10 ชุมสายโทรศัพท์  | 20   |
| 2.11 พื้นฐานระบบเซลลูลาร์  | 20   |
| 2.12 หลักการจัดแบ่งเซลล์ในระบบโทรศัพท์เซลลูลาร์                        | 22   |
| 2.13 การใช้งานระบบ GSM   | 24   |
| 2.14 ลักษณะของการส่งข้อความสั้น  | 25   |
| 2.14.1 การรับ-ส่งข้อความสั้นแบบโหมคพีดียู (PDU-Mode)                   | 25   |
| 2.14.2 ตัวอย่างการส่งข้อความสั้นแบบโหมคพีดียู                          | 25   |
| 2.14.3 ตัวอย่างการรับข้อความสั้นแบบโหมคพีดียู                          | 29   |

## สารบัญ (ต่อ)

| หัวข้อ   | หน้า      |
|--|-----------|
| 2.14.4 ตัวอย่างการรับข้อความสั้นแบบภาษาไทย                             | 33        |
| 2.15 คำสั่ง AT Command กับมือถือ                                       | 36        |
| 2.16 สถาปัตยกรรมไมโครคอนโทรลเลอร์ AVR                                  | 45        |
| 2.16.1 คุณสมบัติที่สำคัญ   | 46        |
| 2.16.2 ขาพอร์ตอินพุตเอาต์พุต   | 47        |
| 2.16.3 ขาพอร์ตอินพุตเอาต์พุต   | 47        |
| 2.16.4 ไทมเมอร์/เคาน์เตอร์   | 48        |
| 2.16.5 โหมดการทำงานของไทมเมอร์/เคาน์เตอร์                              | 49        |
| 2.16.6 การกำหนดโหมดการทำงาน  | 50        |
| 2.17 ระบบสัญญาณเตือนอัคคีภัย (Fire Alarm System)                       | 52        |
| 2.17.1 ส่วนประกอบของระบบสัญญาณเตือนอัคคีภัย                            | 53        |
| 2.18 ชนิดอุปกรณ์เริ่มสัญญาณแบบอัตโนมัติ (Automatic Initiation Devices) | 55        |
| 2.18.1 อุปกรณ์ตรวจจับควัน (Smoke Detector)                             | 55        |
| 2.18.2 อุปกรณ์ตรวจจับความร้อน (Heat Detector)                          | 56        |
| 2.18.3 อุปกรณ์ตรวจจับเปลวไฟ (Flame Detector)                           | 57        |
| 2.19 การออกแบบระบบสัญญาณเตือนอัคคีภัย                                  | 57        |
| 2.19.1 การจัดแบ่งโซน   | 58        |
| 2.19.2 การออกแบบติดตั้ง Manual Station                                 | 58        |
| 2.19.3 การกำหนดตำแหน่งอุปกรณ์แจ้งสัญญาณ                                | 59        |
| 2.19.4 ตำแหน่งการติดตั้งตู้ควบคุม (Fire Alarm Control Panel)           | 59        |
| 2.20 โครงสร้างของโทรศัพท์ NOKIA 3310                                   | 59        |
| 2.20.1 UI หรือ USER INTERFACE  | 59        |
| 2.20.2 แฟลชคอนเนคเตอร์ FLASH CONNECTOR                                 | 60        |
| 2.20.3 SIM CARD  | 61        |
| 2.20.4 ภาคแปลงสัญญาณ ( COBBA )   | 61        |
| 2.20.5 อุปกรณ์เชื่อมต่อกับ COBBA                                       | 63        |
| 2.21 ภาควิทยุ RF ( RADIO FREQUENCY )                                   | 63        |
| 2.21.1 ภาครับ-ส่งความถี่ ( HAGAR )                                     | 63        |
| <b>บทที่ 3 การคำนวณและการสร้าง</b>                                     | <b>67</b> |
| 3.1 การทำงานของระบบ  | 67        |

## สารบัญ (ต่อ)

| หัวข้อ  | หน้า       |
|---|------------|
| 3.2 วงจรภาครับ – ส่งสัญญาณวิทยุ                                     | 68         |
| 3.2.1 หลักการทำงานของภาคส่ง   | 69         |
| 3.2.2 หลักการออกแบบวงจรภาคส่งสัญญาณวิทยุ                            | 70         |
| 3.2.3 การออกแบบวงจรภาคส่งสัญญาณวิทยุ                                | 75         |
| 3.2.4 หลักการทำงานของภาครับสัญญาณวิทยุ                              | 81         |
| 3.2.5 การออกแบบของภาครับสัญญาณวิทยุ                                 | 83         |
| 3.2.4 การออกแบบการทำงานของภาครับสัญญาณวิทยุ                         | 85         |
| 3.3 วงจรควบคุมไมโครคอนโทรลเลอร์ และเปิด-ปิดสัญญาณเตือนภัย           | 89         |
| 3.3.1 วงจรควบคุมการส่ง SMS และเปิด-ปิดสัญญาณเตือนภัย                | 90         |
| 3.3.2 การออกแบบด้านอินพุตของวงจรควบคุม                              | 91         |
| 3.3.3 การออกแบบด้านเอาต์พุตของวงจรควบคุม                            | 94         |
| 3.3.4 การออกแบบวงจรซาร์จแบตเตอรี่ โทรศัพท์มือถือ                    | 96         |
| 3.3.5 แหล่งจ่ายไฟ   | 101        |
| 3.3.6 วงจรลดระดับแรงดันและไฟสำรอง                                   | 102        |
| 3.4 การทำงานของโปรแกรมควบคุม  | 103        |
| 3.5 โหมดการรับ - ส่งข้อมูล SMS                                      | 107        |
| 3.5.1 รูปแบบทั่วไปของการส่ง SMS                                     | 107        |
| 3.5.2 การส่งข้อความ SMS ในพีดียูโหมด                                | 107        |
| 3.5.3 การแปลงข้อมูลแอสกีไค้ดเป็นพีดียูไค้ด (ASCII Code to PDU Code) | 108        |
| 3.5.4 การส่ง SMS  | 111        |
| <b>บทที่ 4 การทดลองและผลการทดลอง</b>                                | <b>113</b> |
| 4.1 การทดสอบวงจรโดยโปรแกรม Proteus 7 Professional                   | 113        |
| 4.2 การประกอบวงจรลงแผ่น PCB (Print Circuit Board )                  | 119        |
| 4.2.1 การลงอุปกรณ์ของวงจร ไมโครคอนโทรลเลอร์                         | 119        |
| 4.2.2 การลงอุปกรณ์ของวงจรลดระดับแรงดัน                              | 120        |
| 4.2.3 การประกอบส่วนเชื่อมต่อระหว่างมือถือ กับวงจรควบคุม             | 121        |
| 4.2.4 การลงอุปกรณ์ของวงจรรับ – ส่งสัญญาณวิทยุ                       | 123        |
| 4.2.5 การประกอบกล่องควบคุม  | 124        |
| 4.2.6 วงจรประกอบกล่องเอนกประสงค์และการติดตั้ง                       | 125        |
| 4.3 การทดลองวงจรตรวจจับวัน  | 126        |

## สารบัญ (ต่อ)

| หัวข้อ  | หน้า       |
|---|------------|
| 4.4 การทดสอบการส่งข้อความสั้นผ่าน โปรแกรม HyperTerminal     | 128        |
| 4.5 ผลการทดลองการวัดสัญญาณ                                  | 135        |
| 4.5.1 การวัดสัญญาณส่วนวงจรรับ – ส่งสัญญาณ                   | 135        |
| 4.5.2 การวัดสัญญาณของวงจรควบคุมไมโครคอนโทรลเลอร์            | 142        |
| <b>บทที่ 5 สรุปผลการทดลอง</b>                               | <b>149</b> |
| 5.1 สรุปผลการทดลองโดยโปรแกรม Proteus                        | 149        |
| 5.2 สรุปผลการทดลองการประกอบวงจร และการวัดผล                 | 151        |
| 5.3 สรุปผลการทดลองผ่าน โปรแกรม HyperTerminal                | 154        |
| 5.4 สรุปผลการวัดสัญญาณ ของวงจรรับ – ส่ง สัญญาณ              | 155        |
| 5.4 สรุปผลการวัดสัญญาณ ของวงจรควบคุมไมโครคอนโทรลเลอร์       | 157        |
| 5.4 แนวทางการพัฒนาและวิจารณ์ผลการทดลอง                      | 159        |
| 5.4.1 เกี่ยวกับการเขียนโปรแกรมในไมโครคอนโทรลเลอร์           | 159        |
| 5.4.2 เกี่ยวกับการพัฒนาในส่วนของวงจรรับ – ส่งสัญญาณ         | 159        |
| 5.4.3 เกี่ยวกับการพัฒนาในส่วนของวงจรควบคุมไมโครคอนโทรลเลอร์ | 159        |
| <b>กิตติกรรมประกาศ</b>                                      | <b>160</b> |
| <b>บรรณานุกรม</b>   |            |
| <b>ภาคผนวก 1</b>  |            |
| <b>ภาคผนวก 2</b>  |            |

## สารบัญรูปรูปภาพ

| รูปภาพที่  | หน้า     |
|--|----------|
| <b>บทที่ 1 บทนำ</b>  | <b>1</b> |
| รูปที่ 1.1 แสดงบล็อกไดอะแกรมการทำงานของระบบ                                  | 2        |
| <b>บทที่ 2 ทฤษฎีและหลักการ</b>   | <b>3</b> |
| รูปที่ 2.1 แสดงรูปแบบตัวกลางแบบมีสาย   | 3        |
| รูปที่ 2.2 แสดงรูปแบบตัวกลางแบบไร้สาย  | 4        |
| รูปที่ 2.3 แสดงสัญญาณข้อมูล สัญญาณพาหะ และสัญญาณที่ได้จากการมอดูเลต          | 5        |
| รูปที่ 2.4 สัญญาณแอมพลิจูดมอดูเลชันในแกนความถี่                              | 7        |
| รูปที่ 2.5 แสดงแถบความถี่ของของสัญญาณ FM                                     | 10       |
| รูปที่ 2.6 สัญญาณพื้นฐานของชุมสายโทรศัพท์                                    | 15       |
| รูปที่ 2.7 กระแสไฟตรงที่เลี้ยงคู่สาย ขณะยกหูและวางหู                         | 16       |
| รูปที่ 2.8 การทำงานของเครื่องโทรศัพท์  | 17       |
| รูปที่ 2.9 ระบบโทรศัพท์แบบ DTMF  | 18       |
| รูปที่ 2.10 ความถี่ของโทนเสียงเมื่อกดหมายเลขต่างๆ                            | 18       |
| รูปที่ 2.11 การทำงานพื้นฐานของชุมสายโทรศัพท์                                 | 20       |
| รูปที่ 2.12 แสดงโครงสร้างของเครือข่ายระบบ โทรศัพท์เซลลูลาร์                  | 21       |
| รูปที่ 2.13 การแบ่งบริเวณพื้นที่ออกเป็นรูปสามเหลี่ยม สี่เหลี่ยม และหกเหลี่ยม | 22       |
| รูปที่ 2.14 การจัดเซลล์ที่มีความถี่แตกต่างกันในแต่ละช่องสัญญาณ               | 23       |
| รูปที่ 2.15 ลักษณะและตำแหน่งการจัดเซลล์ใช้ความถี่ซ้ำๆ ของขนาด K = 7          | 24       |
| รูปที่ 2.16 แสดงลักษณะขั้นตอนของการส่งข้อความสั้น                            | 25       |
| รูปที่ 2.17 แสดงบล็อกไดอะแกรม AVR ( ATmega16 )                               | 45       |
| รูปที่ 2.18 ขาพอร์ต AVR ( ATmega16 ) ตัวถังแบบ PDIP                          | 47       |
| รูปที่ 2.19 รายละเอียดขาของ ATmega 16  | 48       |
| รูปที่ 2.20 แสดงส่วนประกอบของระบบเตือนอัคคีภัย                               | 53       |
| รูปที่ 2.21 แสดงการประกอบอุปกรณ์ชนิดต่างๆ                                    | 54       |
| รูปที่ 2.22 อุปกรณ์ตรวจจับควันไฟ (Smoke Detector)                            | 55       |

## สารบัญรูปภาพ (ต่อ)

| รูปภาพที่  | หน้า      |
|--|-----------|
| รูปที่ 2.23 อุปกรณ์ตรวจจับความร้อน (Heat Detector)                   | 56        |
| รูปที่ 2.24 ส่วนติดต่อผู้ใช้งาน                                      | 59        |
| รูปที่ 2.25 โครงสร้างของโทรศัพท์ NOKIA 3310                          | 60        |
| รูปที่ 2.26 การเชื่อมต่อข้อมูลระหว่างคอมพิวเตอร์และมือถือ            | 61        |
| รูปที่ 2.27 การเชื่อมต่อ Sim Card                                    | 61        |
| รูปที่ 2.28 ส่วนเชื่อมต่อระหว่างภาคเบสแบนด์กับภาควิทยุ               | 62        |
| รูปที่ 2.29 การเชื่อมต่อ ไมโครโฟนและลำโพง                            | 63        |
| รูปที่ 2.30 บล็อกไดอะแกรม ภาควิทยุ RF ( RADIO FREQUENCY )            | 66        |
| <b>บทที่ 3 การคำนวณและการสร้าง</b>                                   | <b>67</b> |
| รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ                              | 67        |
| รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของ วงจรภาครับ-ส่งสัญญาณวิทยุ        | 68        |
| รูปที่ 3.3 วงจรรวมของภาคส่งสัญญาณวิทยุ                               | 69        |
| รูปที่ 3.4 บล็อกไดอะแกรมภายในของ IC HA 17555                         | 70        |
| รูปที่ 3.5 บล็อกไดอะแกรมการใช้งานวงจรรวม แบบอะสเตเบิลมัลติไวเบรเตอร์ | 73        |
| รูปที่ 3.6 ไดอะแกรมเวลาของวงจร ในการคายและเก็บประจุ                  | 73        |
| รูปที่ 3.7 แสดงการต่อของวงจรภาคส่ง                                   | 75        |
| รูปที่ 3.8 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้                        | 76        |
| รูปที่ 3.9 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้                        | 77        |
| รูปที่ 3.10 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้                       | 78        |
| รูปที่ 3.11 วงจรส่งสัญญาณคลื่นวิทยุ                                  | 80        |
| รูปที่ 3.12 วงจรรวมของภาครับสัญญาณวิทยุ                              | 81        |
| รูปที่ 3.13 วงจรภาครับสัญญาณ   | 82        |
| รูปที่ 3.14 บล็อกไดอะแกรมภายในของ IC NE567                           | 82        |
| รูปที่ 3.15 แสดงการต่อ $R_1$ และ $C_1$ กับ IC NE567                  | 83        |
| รูปที่ 3.16 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรล็อกเฟส  | 85        |
| รูปที่ 3.17 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรล็อกเฟส  | 86        |
| รูปที่ 3.18 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรล็อกเฟส  | 87        |

## สารบัญรูปภาพ (ต่อ)

| รูปภาพที่   | หน้า |
|---|------|
| รูปที่ 3.19 วงจรถอดรหัสสัญญาณ   | 88   |
| รูปที่ 3.20 บล็อกไคอะแกรมส่วนควบคุมการส่ง SMS และควบคุมการเปิด - ปิด สัญญาณเตือนภัย           | 89   |
| รูปที่ 3.21 วงจรไมโครคอนโทรลเลอร์ที่ควบคุมอุปกรณ์   | 90   |
| รูปที่ 3.22 วงจรแปลงข้อมูลจากแบบขนานเป็นแบบอนุกรม   | 91   |
| รูปที่ 3.23 บล็อกไคอะแกรมของ IC SN74HC165N  | 91   |
| รูปที่ 3.24 ไคอะแกรมเวลาของการทำงาน IC SN74HC165N   | 92   |
| รูปที่ 3.25 บล็อกไคอะแกรมของไมโครคอนโทรลเลอร์ ATmega168V                                      | 93   |
| รูปที่ 3.26 แสดงรายละเอียดของขาไมโครคอนโทรลเลอร์  | 94   |
| รูปที่ 3.27 วงจรแปลงข้อมูลจากแบบอนุกรมเป็นแบบขนาน   | 94   |
| รูปที่ 3.28 บล็อกไคอะแกรมของ IC TPIC6B595N  | 95   |
| รูปที่ 3.29 วงจรชาร์จแบตเตอรี่โทรศัพท์มือถือ  | 96   |
| รูปที่ 3.30 บล็อกไคอะแกรมและสัญญาณเอาต์พุตของ ไอซี KA34063A                                   | 96   |
| รูปที่ 3.31 วงจรการแปลงระดับความต่างศักย์   | 97   |
| รูปที่ 3.32 แสดงการวนของกระแสในวงจรเมื่อ TR “ ON ”  | 97   |
| รูปที่ 3.33 แสดงการวนของกระแสในวงจรเมื่อ TR “ OFF ”   | 98   |
| รูปที่ 3.34 การต่อของอุปกรณ์กับ IC KA34063A   | 98   |
| รูปที่ 3.35 แสดงการวางอุปกรณ์ เพื่อให้ได้ค่า Vout ตามที่คำนวณไว้                              | 99   |
| รูปที่ 3.36 แสดงการวางอุปกรณ์ R <sub>sc</sub>   | 100  |
| รูปที่ 3.37 วงจรแหล่งจ่ายไฟ   | 101  |
| รูปที่ 3.38 วงจรลดระดับแรงดันและไฟสำรอง   | 102  |
| รูปที่ 3.39 บล็อกไคอะแกรมของส่วนควบคุมการส่งข้อความสั้น                                       | 103  |
| รูปที่ 3.40 บล็อกไคอะแกรมของส่วนควบคุมการอ่านข้อความสั้น                                      | 104  |
| รูปที่ 3.41 บล็อกไคอะแกรมของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ติดต่อกับ<br>โทรศัพท์มือถือ         | 105  |
| รูปที่ 3.42 บล็อกไคอะแกรมของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ติดต่อกับ<br>โทรศัพท์มือถือ ( ต่อ ) | 106  |
| รูปที่ 3.43 แสดงการรับ - ส่งข้อความแบบ Text Mode  | 107  |
| รูปที่ 3.44 แสดงการรับ - ส่งข้อความแบบ PDU Mode   | 107  |

## สารบัญรูปภาพ (ต่อ)

| รูปภาพที่   | หน้า       |
|---|------------|
| <b>บทที่ 4 การทดลองและผลการทดลอง</b>  | <b>113</b> |
| รูปที่ 4.1 แบบจำลองการทำงานของวงจร ภายในโปรแกรม Proteus                             | 113        |
| รูปที่ 4.2 แสดง Source Code ที่เขียนบน โปรแกรม AVR Studio                           | 113        |
| รูปที่ 4.3 ไฟล์ .hex ของ Source Code ที่ Compiled                                   | 114        |
| รูปที่ 4.4 โปรแกรม PonyProg 2000 ที่แสดงข้อมูลเลขฐาน 16 ที่ทำการ Compiled จากภาษา C | 115        |
| รูปที่ 4.5 แสดงการใส่โปรแกรมลงในวงจรจำลองของโปรแกรม Proteus                         | 116        |
| รูปที่ 4.6 การทดสอบการเข้ารหัส PDU เมื่อกด IN0                                      | 117        |
| รูปที่ 4.7 แสดงความแตกต่างของรหัส PDU ที่ส่งตาม Zone1 – Zone8 จากการรับ IN0 – IN7   | 117        |
| รูปที่ 4.8 แสดงขั้นตอนการลงอุปกรณ์  | 119        |
| รูปที่ 4.9 ลายแผ่นทองแดงที่ผ่านการกัด และเจาะรู ของวงจรลวดระดับแรงดัน               | 120        |
| รูปที่ 4.10 แสดงการลงอุปกรณ์ของวงจรลวดระดับแรงดัน                                   | 120        |
| รูปที่ 4.11 ขาดอนเน็คเตอร์ที่เชื่อมต่อไปยัง โทรศัพท์มือถือ                          | 121        |
| รูปที่ 4.12 สายคาต้าลังค์ของ โทรศัพท์มือถือ   | 122        |
| รูปที่ 4.13 แสดงพอร์ตอนุกรมสื่อสารของ โทรศัพท์มือถือ                                | 122        |
| รูปที่ 4.14 วงจรส่วนภาคส่งทั้ง 3 ช่อง   | 123        |
| รูปที่ 4.15 วงจรส่วนภาครับสัญญาณวิทยุ   | 123        |
| รูปที่ 4.16 วงจรภาคส่งประกอบลงกล่อง   | 123        |
| รูปที่ 4.17 แสดงการนำอุปกรณ์ลงกล่อง ของวงจรควบคุม                                   | 124        |
| รูปที่ 4.18 การนำอุปกรณ์ลงกล่องควบคุมของวงจรภาคส่ง                                  | 125        |
| รูปที่ 4.19 การติดตั้งวงจรภาคส่งพร้อมทั้งส โมคคิตเทคเตอร์                           | 125        |
| รูปที่ 4.20 แสดงการนำอุปกรณ์ลงกล่องของวงจรส่งสัญญาณควบคุม                           | 125        |
| รูปที่ 4.21 การประกอบวงจรพร้อมการใช้งานของภาคส่ง                                    | 126        |
| รูปที่ 4.22 ข้อความที่ส่งจากวงจรควบคุมถึงโทรศัพท์มือถือปลายทาง                      | 126        |
| รูปที่ 4.23 การทดสอบการส่งการเตือนภัย ผ่าน sms ทั้งหมด 8 จุด ที่แตกต่างกัน          | 127        |
| รูปที่ 4.24 Dialog Box Setting จาก โปรแกรม Hyper Terminal                           | 128        |
| รูปที่ 4.25 แสดงการตอบรับจากโทรศัพท์มือถือเมื่อได้รับคำสั่ง AT                      | 129        |
| รูปที่ 4.26 ผลการทดลองเมื่อพิมพ์คำสั่ง AT+CMGF ผ่าน Hyper Terminal                  | 130        |
| รูปที่ 4.27 แสดงการทดสอบการส่งข้อความใน PDU Mode ผ่าน Hyper Terminal                | 131        |
| รูปที่ 4.28 แสดงข้อความที่โทรศัพท์มือถือปลายทางได้รับข้อความว่า #01>Zone1           | 131        |

## สารบัญรูปภาพ (ต่อ)

| รูปภาพที่   | หน้า |
|---|------|
| รูปที่ 4.29 แสดงการทดสอบการส่งข้อความใน PDU Mode ผ่าน Hyper Terminal                  | 134  |
| รูปที่ 4.30 แสดงข้อความที่โทรศัพท์มือถือปลายทางได้รับข้อความว่า #01>Zone2             | 134  |
| รูปที่ 4.31 ขาของ IC HA17555 แต่ละส่วนที่ทำการต่อจากวงจรของภาคส่งสัญญาณ               | 135  |
| รูปที่ 4.32 TWS – 434A ทำหน้าที่มอดูเลตสัญญาณและทำการส่งไปยังภาครับ                   | 135  |
| รูปที่ 4.33 โมดูล RLP – 434A ทำหน้าที่รับสัญญาณ                                       | 136  |
| รูปที่ 4.34 แสดงช่องที่ทำการส่งสัญญาณออกของภาครับ                                     | 136  |
| รูปที่ 4.35 แสดงจุดที่วัดสัญญาณของวงจรภาคส่ง  | 137  |
| รูปที่ 4.36 แสดงจุดที่วัดสัญญาณของวงจรภาครับ  | 137  |
| รูปที่ 4.37 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 1                     | 128  |
| รูปที่ 4.38 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 1                               | 138  |
| รูปที่ 4.39 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 2                     | 139  |
| รูปที่ 4.40 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 2                               | 139  |
| รูปที่ 4.41 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 3                     | 140  |
| รูปที่ 4.42 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 3                               | 140  |
| รูปที่ 4.43 สัญญาณที่ยังไม่ได้ทำการมอดูเลต  | 141  |
| รูปที่ 4.44 สัญญาณที่ทำการมอดูเลตกับคลื่นพาหะ   | 141  |
| รูปที่ 4.45 แสดงวงจรที่ทำการวัดสัญญาณอินพุต ที่เข้าทาง IC SN74HC165                   | 142  |
| รูปที่ 4.46 วงจรไมโครคอนโทรลเลอร์ที่ต่อกับอุปกรณ์พร้อมทำการทดลอง                      | 143  |
| รูปที่ 4.47 สัญญาณขาที่ 11 ของ IC SN74HC165 ขณะที่ยังไม่มีสัญญาณอินพุตเข้ามา          | 144  |
| รูปที่ 4.48 สัญญาณขาที่ 11 ของ IC SN74HC165 ขณะที่มีการสัญญาณอินพุตเข้ามา             | 144  |
| รูปที่ 4.49 แสดงวงจรที่ทำการวัดสัญญาณที่ IC SN74HC165 ส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ | 145  |
| รูปที่ 4.50 สัญญาณขาที่ 9 ของ IC SN74HC165 สัญญาณเอาต์พุต                             | 145  |
| ที่แปลงจากแบบขนานเป็นแบบอนุกรม  |      |
| รูปที่ 4.51 แสดงวงจรไมโครคอนโทรลเลอร์ที่ทำการติดต่อพอร์ตสื่อสารกับโทรศัพท์มือถือ      | 146  |
| รูปที่ 4.52 พอร์ตสื่อสารของโทรศัพท์มือถือที่อยู่ด้านหลัง                              | 146  |
| รูปที่ 4.53 สายคาต้าลิงค์ที่เชื่อมต่อกับโทรศัพท์มือถือ                                | 146  |
| รูปที่ 4.54 วงจรไมโครคอนโทรลเลอร์ที่ต่อกับพอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือ         | 147  |
| รูปที่ 4.55 สัญญาณที่ขาที่ 3 (PD1) ของไมโครคอนโทรลเลอร์ ขณะที่ยังไม่มีสัญญาณออกมา     | 148  |
| รูปที่ 4.56 สัญญาณที่ขาที่ 3 (PD1) ของไมโครคอนโทรลเลอร์ ขณะที่มีการสัญญาณออกมา        | 148  |

## สารบัญรูปภาพ (ต่อ)

| รูปภาพที่  | หน้า       |
|--|------------|
| <b>บทที่ 5 สรุปผลการทดลอง</b>  | <b>149</b> |
| รูปที่ 5.1 แสดงวงจรจำลองที่สร้างขึ้นบน โปรแกรม Proteus   | 149        |
| รูปที่ 5.2 แสดงผลคอบสนองของ โปรแกรมเมื่อทำการทดลอง   | 150        |
| รูปที่ 5.3 การนำอุปกรณ์ทั้งหมดลงในกล่องใส่อุปกรณ์เพื่อความเรียบร้อย                                    | 151        |
| รูปที่ 5.4 การประกอบวงจรควบคุมที่สมบูรณ์   | 151        |
| รูปที่ 5.5 วงจรภาคส่งที่นำอุปกรณ์ลงกล่องเรียบร้อย  | 152        |
| รูปที่ 5.6 วงจรภาคส่งที่ติดตั้งพร้อมใช้งาน   | 152        |
| รูปที่ 5.7 แสดงการต่อวงจรพร้อมใช้งาน และข้อความที่ได้รับเมื่อมีการแจ้งเตือน                            | 153        |
| รูปที่ 5.8 แสดงข้อความที่ได้รับ  | 154        |
| รูปที่ 5.9 แสดงค่าที่สามารถรับได้จากการส่งของ ไมโครคอนโทรเลอร์กับ โทรศัพท์มือถือ                       | 154        |
| รูปที่ 5.10 แสดงการเปรียบเทียบสัญญาณภาคส่ง และภาครับ ของภาคส่งตัวที่ 1                                 | 154        |
| รูปที่ 5.11 แสดงการเปรียบเทียบสัญญาณภาคส่ง และภาครับ ของภาคส่งตัวที่ 2                                 | 155        |
| รูปที่ 5.12 แสดงการเปรียบเทียบสัญญาณภาคส่ง และภาครับ ของภาคส่งตัวที่ 3                                 | 155        |
| รูปที่ 5.13 สัญญาณที่ส่งออกทางสายอากาศขณะยังไม่มียอดูเลต   | 156        |
| รูปที่ 5.14 สัญญาณที่ส่งออกทางสายอากาศเมื่อมียอดูเลต   | 156        |
| รูปที่ 5.15 แสดงการเปรียบเทียบสัญญาณเมื่อไม่มีอินพุตเข้ามา และเมื่อมีสัญญาณอินพุตเข้ามา                | 157        |
| รูปที่ 5.16 สัญญาณที่ส่งออกไปยัง ไมโครคอนโทรเลอร์ในแบบอนุกรม   | 157        |
| รูปที่ 5.17 แสดงการเปรียบเทียบสัญญาณเมื่อไม่มีเอาต์พุต และเมื่อมีสัญญาณเอาต์พุตออกจาก ไมโครคอนโทรเลอร์ | 158        |
| รูปที่ 5.18 แสดงการได้รับข้อความของ โทรศัพท์ปลายทาง  | 158        |

## สารบัญตาราง

| ตารางที่  | หน้า     |
|---|----------|
| <b>บทที่ 2 ทฤษฎีและหลักการ</b>  | <b>3</b> |
| ตารางที่ 2.1 แสดงความถี่ที่มอดูเลตกันเมื่อคหหมายเลขต่างๆ                                      | 19       |
| ตารางที่ 2.2 แสดงส่วนประกอบของชุดข้อมูลในการส่งข้อความสั้นแบบ โหมดพีดียู                      | 26       |
| ตารางที่ 2.3 ส่วนประกอบของข้อมูลที่ส่ง  | 28       |
| ตารางที่ 2.4 ส่วนประกอบของสตรีมการรับข้อความสั้น  | 29       |
| ตารางที่ 2.5 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิตข้อความ “hellohello”            | 31       |
| ตารางที่ 2.6 ชุดของตัวแปรของมาตรฐาน GSM   | 32       |
| ตารางที่ 2.7 แสดงตารางรหัสตัวอักษรของการเข้ารหัสแบบ UCS2 (16 บิต)                             | 33       |
| ตารางที่ 2.8 ส่วนประกอบของข้อมูลที่รับแบบ UCS2 (16 บิต)                                       | 34       |
| ตารางที่ 2.9 ส่วนประกอบของข้อมูลที่ส่งแบบ UCS2 (16 บิต)                                       | 35       |
| ตารางที่ 2.10 แสดงวิธีการแปลงตัวอักษร โดยวิธีการเข้ารหัสแบบ UCS2 (16bit)<br>ข้อความ “สวีตตี้” | 36       |
| ตารางที่ 2.11 ลักษณะชุดคำสั่งของ AT+CNMI  | 37       |
| ตารางที่ 2.12 ลักษณะชุดคำสั่งของ AT+CSCB  | 38       |
| ตารางที่ 2.13 ลักษณะชุดคำสั่งของ AT+CSGF  | 38       |
| ตารางที่ 2.14 ลักษณะชุดคำสั่งของ AT+CSCA  | 39       |
| ตารางที่ 2.15 ลักษณะชุดคำสั่งของ AT+CMGL  | 39       |
| ตารางที่ 2.16 ลักษณะชุดคำสั่งของ AT+CMGR  | 40       |
| ตารางที่ 2.17 ลักษณะชุดคำสั่งของ AT+CMGS  | 40       |
| ตารางที่ 2.18 ลักษณะชุดคำสั่งของ AT+CMSS  | 41       |
| ตารางที่ 2.19 ลักษณะชุดคำสั่งของ AT+CMGW  | 42       |
| ตารางที่ 2.20 ลักษณะชุดคำสั่งของ AT+CMGD  | 42       |
| ตารางที่ 2.21 ลักษณะชุดคำสั่งของ AT+CSMS  | 43       |
| ตารางที่ 2.22 ลักษณะชุดคำสั่งของ AT+CPMS  | 43       |
| ตารางที่ 2.23 ลักษณะชุดคำสั่งของ AT+CMGC  | 44       |
| ตารางที่ 2.24 แสดงบิตของรีจิสเตอร์ TCCR0  | 50       |

## สารบัญตาราง (ต่อ)

| ตารางที่  | หน้า       |
|---|------------|
| ตารางที่ 2.25 บิตกำหนดรูปแบบสัญญาณของไทมเมอร์/เคาน์เตอร์                                | 50         |
| ตารางที่ 2.26 แสดงรายละเอียดของ ค่า BOTTOM, MAX และ TOP                                 | 51         |
| ตารางที่ 2.27 บิตกำหนดแหล่งสัญญาณนาฬิกาของไทมเมอร์                                      | 51         |
| ตารางที่ 2.28 แสดงบิตของรีจิสเตอร์ TIFR   | 52         |
| ตารางที่ 2.29 คุณสมบัติของตัวตรวจจับชนิดต่างๆ   | 57         |
| <br>  |            |
| <b>บทที่ 3 การคำนวณและการสร้าง</b>  | <b>67</b>  |
| ตารางที่ 3.1 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ด้วยข้อความ “#01>Zone1 ” | 109        |
| ตารางที่ 3.2 ชุดของตัวแปรของมาตรฐาน GSM 03.38   | 110        |
| ตารางที่ 3.3 แสดงส่วนประกอบของข้อมูลที่ส่งข้อความสั้นภาษาอังกฤษของคำว่า “#01>Zone1 ”    | 111        |
| <br>  |            |
| <b>บทที่ 4 การทดลองและผลการทดลอง</b>  | <b>113</b> |
| ตารางที่ 4.1 แสดงส่วนประกอบของข้อมูลที่ส่งข้อความสั้นภาษาอังกฤษของคำว่า “#01>Zone1 ”    | 132        |
| ตารางที่ 4.2 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ด้วยข้อความ “#01>Zone1 ” | 133        |

## บทที่ 1

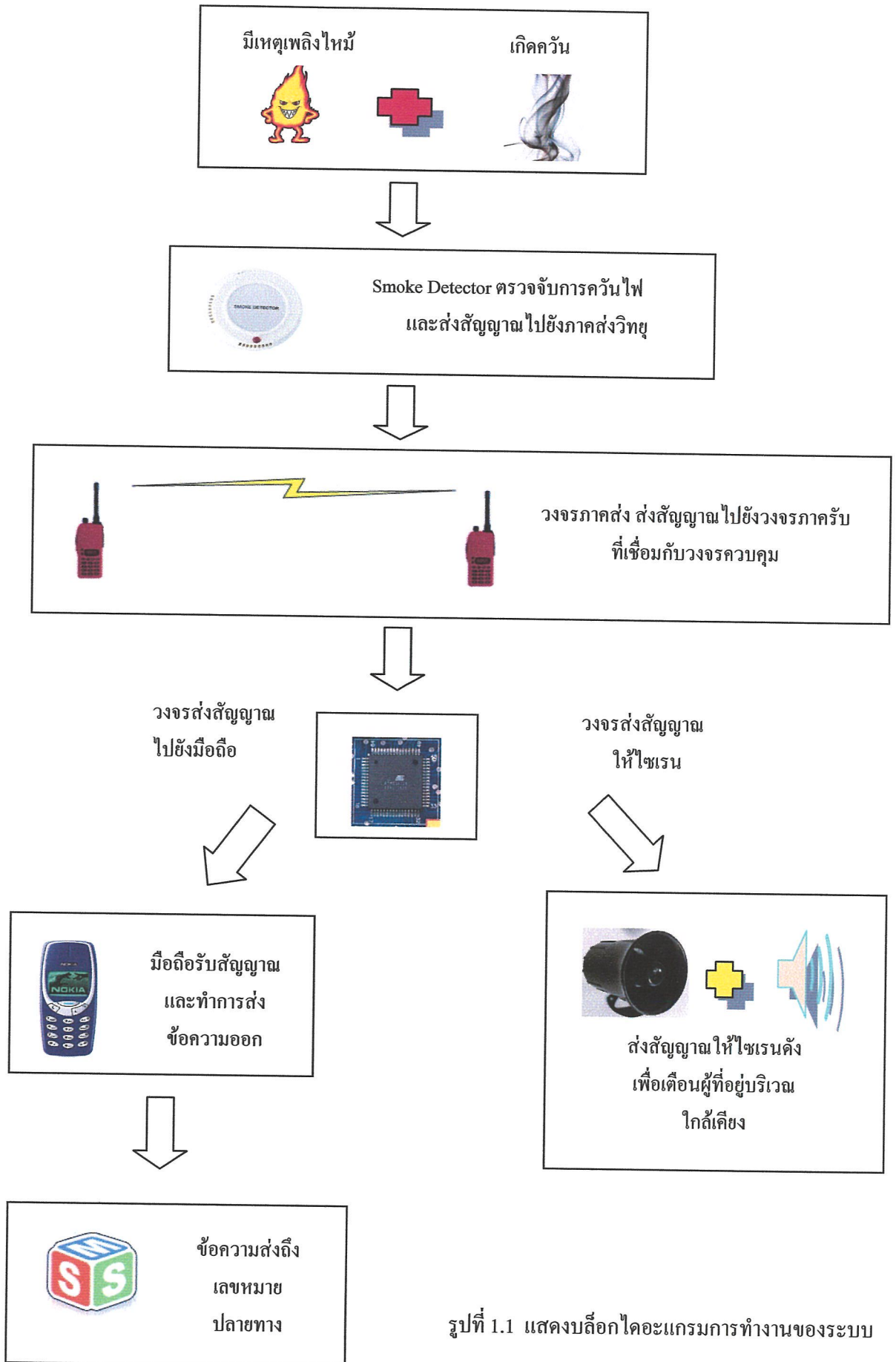
### บทนำ

ในปัจจุบันผู้คนส่วนใหญ่ต้องมีภาระที่ต้องเร่งรีบ ในแต่ละวันต้องออกไปทำงานในสถานที่ต่างๆ นอกบ้าน ทำให้ความปลอดภัยของทรัพย์สินต่างๆที่อยู่ภายในบ้านนั้นดูแลได้ไม่ทั่วถึง โดยเฉพาะเรื่องเกี่ยวกับอัคคีภัย ซึ่งเป็นภัยที่อันตราย และควบคุมได้ยาก โดยในแต่ละปี การสูญเสียจากการเกิดอัคคีภัยนั้นมีมูลค่ามาก

ดังนั้นจึงเป็นเหตุให้เกิดแนวความคิดในการทำโครงการเรื่อง ระบบแจ้งเหตุเพลิงไหม้ และแจ้งเตือนผ่านข้อความสั้น **FIRE ALARM SYSTEM BY SHORT MESSAGE SERVICE(SMS)** เนื่องจากสาเหตุของการเกิดอัคคีภัยส่วนใหญ่ มาจากขณะที่เริ่มเกิดเพลิงไหม้จะไม่มีใครคนอยู่หรือเกิดในบริเวณที่ไม่มีคนเห็น กว่าที่รู้ตัวเพลิงก็ลุกลาม จนเกินกำลังที่คนไม่กี่คนหรืออุปกรณ์ดับเพลิงขนาดเล็กที่มีอยู่ในอาคารจะทำการสกัดไฟได้ ดังนั้นจึงจำเป็นต้องมีระบบสัญญาณเตือนอัคคีภัยติดตั้งไว้ในอาคาร เพื่อให้สามารถรับรู้เหตุการณ่วงหน้าก่อนที่ไฟจะลุกลามจนควบคุมไม่ได้ โดยการแจ้งเตือนผ่าน SMS

เมื่อเกิดเหตุอัคคีภัยในบริเวณต่างๆ สโมคดิเท็คเตอร์จะตรวจจับควันไฟที่เกิดขึ้นในบริเวณนั้น และแปลงเป็นสัญญาณส่งไปยังวงจรภาคส่งวิทยุซึ่งประกอบและติดตั้งอยู่กับตัวสโมคดิเท็คเตอร์ ส่งสัญญาณวิทยุไปยังภาครับเพื่อทำการส่งสัญญาณเข้าไปยังวงจรควบคุม เมื่อวงจรควบคุมรับค่าสัญญาณแจ้งเตือน จะส่งสัญญาณไปยังไซเรนให้ดังเพื่อแจ้งให้ผู้ที่อยู่ในบริเวณใกล้เคียงได้ทราบ และส่งสัญญาณไปยังมือถือที่เชื่อมต่อกับระบบ เพื่อทำการส่งข้อความสั้น ไปยังเลขหมายปลายทางที่บ้านทักไว้

การทำงานของระบบ สามารถอธิบายในรูปแบบของบล็อกไดอะแกรมดังรูปที่ 1.1



รูปที่ 1.1 แสดงบล็อกไดอะแกรมการทำงานของระบบ

## บทที่ 2 ทฤษฎีและหลักการ

การสื่อสาร หมายถึง การส่ง การรับ และการประมวลผลข้อมูลระหว่างจุด 2 จุด ระบบของการสื่อสารข้อมูลทุกชนิดจะมีรูปแบบพื้นฐานที่ประกอบไปด้วยเครื่องส่ง (Transmitter) เครื่องรับ (Receiver) และช่องทางการสื่อสาร (Channel)

### 2.1 ระบบสื่อสาร

ระบบสื่อสารมีส่วนประกอบต่างๆดังนี้

#### เครื่องส่ง (Transmitter)

คือ อุปกรณ์และวงจรอิเล็กทรอนิกส์ที่ได้ถูกออกแบบสำหรับเปลี่ยนสัญญาณข้อมูล(Intelligence) ให้กลายเป็นสัญญาณที่เหมาะสมกับการส่งผ่านช่องสื่อสารผ่านตัวกลางของแต่ละระบบ

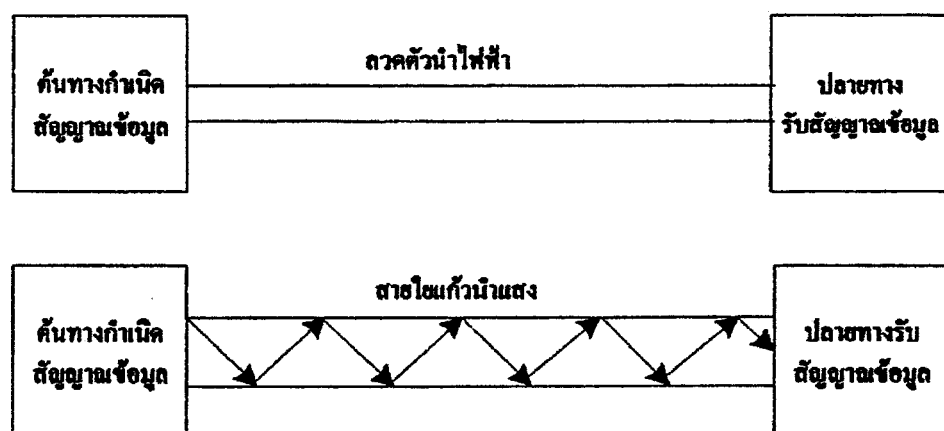
#### เครื่องรับ (Receiver)

คือ อุปกรณ์และวงจรอิเล็กทรอนิกส์อีกชุดหนึ่ง ซึ่งจะรับสัญญาณที่ผ่านช่องสื่อสารและแปลงให้เป็นสัญญาณที่เหมาะสมกับตัวรับของระบบ

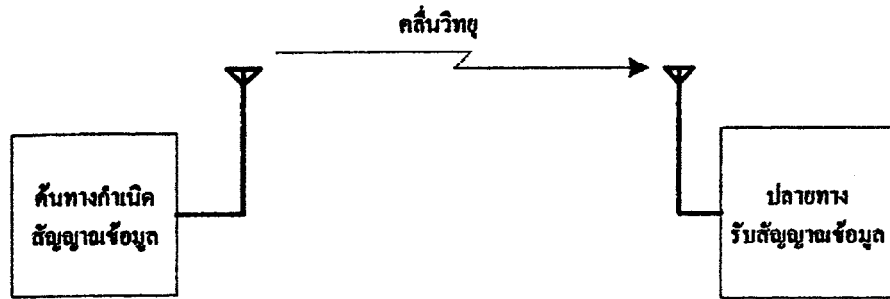
#### ช่องสื่อสารหรือตัวกลาง (Communication channel)

คือ ตัวกลางที่สัญญาณข้อมูลเดินทางผ่านไป จากจุดหนึ่งไปยังอีกจุดหนึ่งในระบบสื่อสารนั้น ตัวกลางของการสื่อสารสามารถมีได้หลายรูปแบบ เช่น

- แบบมีสาย (Wire or Cable)
- แบบไร้สาย (Wireless or Radio)



รูปที่ 2.1 แสดงรูปแบบตัวกลางแบบมีสาย



รูปที่ 2.2 แสดงรูปแบบค้วกลำงเบบไร้สาย

ซึ่งในที่นี่จะกล่าวถึงทฤษฎีพื้นฐานที่สำคัญสำหรับการสื่อสารแบบไร้สายดังต่อไปนี้

## 2.2 ความรู้พื้นฐานของการสื่อสารระบบไร้สาย

การทำความเข้าใจระบบสื่อสารไร้สายควรที่จะทราบว่าสิ่งใดเป็นองค์ประกอบเบื้องต้นบ้าง โดยมีรูปแบบการใช้พาหะในการสื่อสารเป็นหลักใหญ่ๆ อยู่สองแบบนั่นคือ

1. ระบบที่ใช้คลื่นวิทยุเป็นพาหะ ( Radio Carrier ) ได้แก่ ระบบโทรศัพท์เคลื่อนที่ ระบบวิทยุติดตามตัว ระบบสื่อสารผ่านดาวเทียม ระบบวิทยุกระจายเสียง และระบบคลื่นไมโครเวฟ
2. ระบบที่ใช้คลื่นแสงเป็นพาหะ ( Light Carrier ) ระบบสื่อสารผ่านแสงเลเซอร์อินฟราเรด

## 2.3 ระบบคลื่นพาหะที่เป็นคลื่นวิทยุ ( Radio Carrier )

การสื่อสารด้วยคลื่นวิทยุ อาจเรียกได้ว่าเป็นการพัฒนาให้มนุษย์สามารถส่งข้อมูลหรือเสียงไปได้ไกลโดยไม่ต้องพึ่งสาย โดยธรรมชาติของคลื่นวิทยุ นั้นเป็นคลื่นแม่เหล็กไฟฟ้าขั้นต้นตอน คือ มีการแพร่กระจายพลังงานคลื่นจากสายอากาศด้านผู้ส่ง ซึ่งคลื่นสามารถเดินทางได้เร็วเท่าความเร็วของแสง นั่นคือ 186,000 ไมล์ต่อวินาที หรือ 300,000,000 เมตรต่อวินาที เมื่อคลื่นเดินทางมาถึงสายอากาศด้านผู้รับจะเกิดค่าความต่างศักย์ไฟฟ้าจำนวนน้อยๆค่าหนึ่ง ซึ่งหากมีการขยายและแปลงสัญญาณที่มีการส่งมา ก็จะได้สัญญาณเดิมกลับมา ส่วนข้อมูลข่าวสารหรือเสียงของมนุษย์นั้นไม่มีวิธีการ คือ นำเสียงพูดรวมเข้าไปกับสัญญาณคลื่นวิทยุ ที่เป็นสื่ออีกช่วงความถี่หนึ่ง เราเรียกขั้นตอนนี้ดังกล่าวว่า การมอดูเลต ( Modulation )

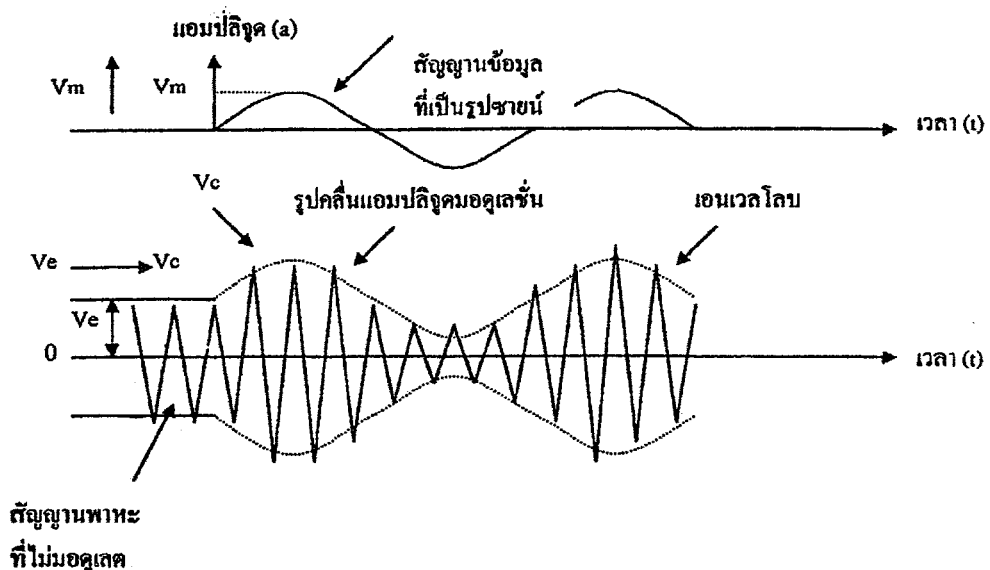
## 2.4 การมอดูเลตขั้น ( Modulation Technique )

จากความต้องการติดต่อสื่อสารระยะไกล สัญญาณเสียงพูดของมนุษย์ซึ่งอยู่ในช่วงความถี่ต่ำมีความยาวคลื่นมาก หากต้องการส่งเสียงพูดออกไปในอากาศ ก็จำเป็นต้องใช้เสาอากาศที่มีขนาดสูงมากซึ่งไม่เหมาะสมและอาจมีสัญญาณรบกวนได้ง่าย จึงมีการคิดค้นเทคนิคที่จะทำให้เปลี่ยนความถี่ของเสียงพูดให้ไปอยู่ในอีกช่วงหนึ่งบนแถบความถี่ซึ่งเหมาะสมสำหรับการส่งออกอากาศ

การมอดูเลชันเทคนิคที่สำคัญ เป็นวิธีการเปลี่ยนรูปแบบของสัญญาณหนึ่ง ( ในที่นี้คือ สัญญาณพาหะ Carrier ) ให้เป็นไปตามรูปแบบของสัญญาณอีกชุดหนึ่งซึ่งเป็นข้อมูลหรือเสียงพูดที่ต้องการส่ง สัญญาณข้อมูลหรือเสียงพูดจะถูกเรียกว่า มอดูเลตติ้งซิกแนล ( Modulating Signal ) ส่วนสัญญาณจะถูกเรียกว่าแคเรียร์ ( Carrier ) หรือ มอดูเลตเวฟ ( Modulate Wave )

#### 2.4.1 การมอดูเลชันทางแอมพลิจูด ( Amplitude Modulation )

แอมพลิจูดมอดูเลชันเป็นวิธีการที่คิดค้นได้ก่อนในบรรดาเทคนิคการมอดูเลตอื่นๆ เป็นการนำสัญญาณ 2 ชุดมาผสมกัน โดยสัญญาณแรกเป็นข้อมูลหรือเสียงที่ต้องการส่ง สัญญาณที่สองเป็นสัญญาณพาหะ แอมพลิจูดมอดูเลชันเป็นการนำสัญญาณข้อมูลมาปรับเปลี่ยนสัญญาณแอมพลิจูดของสัญญาณพาหะ โดยที่ความถี่ของสัญญาณพาหะไม่มีการเปลี่ยนแปลง ในที่นี้สมมุติว่าสัญญาณข้อมูลที่ต้องการทำการมอดูเลตชันมีความถี่ค่าหนึ่ง คือ  $f_s$  และสัญญาณข้อมูลที่เป็นพาหะมีความถี่อีกค่าหนึ่งคือ  $f_c$  ผลลัพธ์จากการมอดูเลชันจะเป็นดังรูป



รูปที่ 2.3 แสดงสัญญาณข้อมูล สัญญาณพาหะ และสัญญาณที่ได้จากการมอดูเลต

จากรูป เส้นประซึ่งเกิดจากการต่อของแอมพลิจูดสัญญาณลัพธ์จะมีชื่อเรียกว่า เอนเวลโลป (Envelope) ซึ่งมีลักษณะเช่นเดียวกับคลื่นของ  $f_s$

ลักษณะของสัญญาณที่ได้รับจากการทำแอมพลิจูดมอดูเลชัน

สัญญาณรูปขายนั่น โดยปกติจะสามารถอธิบายได้ด้วยสมการดังนี้

$$v = V \sin(2\pi ft)$$

เมื่อกำหนดให้

$v_s$  = แอมพลิจูดสูงสุดของสัญญาณรูปขายน้ความถี่  $f_s$  เป็นความต่างศักย์มีหน่วยเป็น โวลต์

$v_c$  = แอมพลิจูดของสัญญาณรูปขายน้ความถี่  $f_c$  เป็นความต่างศักย์มีหน่วยเป็น โวลต์

$V_c$  = แอมพลิจูดสูงสุดของสัญญาณรูปขายน้ความถี่  $f_c$  เป็นความต่างศักย์มีหน่วยเป็น โวลต์

$t$  = เป็นเวลาใดๆ มีหน่วยเป็นวินาที

จะได้ว่า  $v_s = V_s \sin(2\pi f_s t)$

$$v_c = V_c \sin(2\pi f_c t)$$

### ข้อสังเกต

ค่าของแอมพลิจูดสูงสุดของความถี่สัญญาณข้อมูล ไม่ควรที่จะมากกว่าแอมพลิจูดสูงสุดของความถี่สัญญาณพาหะ ( Carrier ) มิฉะนั้นจะเกิดการผิดเพี้ยนของสัญญาณข้อมูลได้ หลังจากทำการ มอดูเลชัน หากค่าแอมพลิจูดของสัญญาณข้อมูลสูงกว่าของสัญญาณพาหะมากๆ จะเรียกได้ว่าการ โอเวอร์มอดูเลชัน ( Over - modulation ) และความสัมพันธ์ของค่าความต่างศักย์ของสัญญาณทั้งสองนั้น มีการกำหนดขึ้นเรียกว่าดัชนีการมอดูเลต ( Modulation Index ) ซึ่งเป็นอัตราส่วนระหว่างค่าความต่างศักย์ของสัญญาณข้อมูลกับความต่างศักย์ของสัญญาณพาหะ ดังสมการข้างล่างนี้

$$m = V_s / V_c$$

ค่าของดัชนีการมอดูเลชัน จะอยู่ในช่วง  $0 < m < 1$  เพราะหาก  $m$  มากกว่า 1 ก็จะมีการ โอเวอร์มอดูเลชัน และนิยามคูณ 100 เข้าไปเพื่อแสดงค่าเป็นเปอร์เซ็นต์การมอดูเลชัน

จากรูปคลื่นผลลัพธ์ทำให้เราพอสรุปได้ว่า ค่าความต่างศักย์ที่เวลาใดๆของสัญญาณเกิดจากผลรวมของค่าแอมพลิจูดสูงสุดของสัญญาณพาหะ กับแอมพลิจูดที่เวลาใดๆ ของสัญญาณข้อมูล จะได้ค่าตามสมการข้างล่างนี้

$$v1 = V_c + v_s$$

$$v1 = V_c + v_s \sin(2\pi f_s t)$$

ค่าของ  $v_1$  เป็นค่าของแอมพลิจูดของสัญญาณผลลัพ์ตามที่เรียกว่า เอนเวลโลป ดังนั้น สมการที่แสดงถึงรูปคลื่นผลลัพ์คือ

$$v_2 = v_1 \sin(2\pi f_c t)$$

$$v_2 = V_c + V_s \sin(2\pi f_s t)$$

หากพิจารณาจากสมการจะเห็นว่า สัญญาณพาหะ + ( สัญญาณข้อมูล x สัญญาณพาหะ )

นั่นคือ ส่วนแรกที่เป็นสัญญาณพาหะ และส่วนที่สองจะเป็นผลคูณของสัญญาณพาหะและสัญญาณข้อมูล ดังนั้นวงจรที่จะนำมาใช้สร้างแอมพลิจูดมอดูเลชั่นจะต้องเป็นวงจรคูณสัญญาณ มีชื่อเรียกว่า วงจรสำหรับทำ AM ว่ามอดูเลเตอร์ (Modulator) และจากสูตรตรีโกณมิติ ที่ผลคูณของซายน์

$$\sin A \sin B = \frac{1}{2} [\cos(A - B) - \cos(A + B)]$$

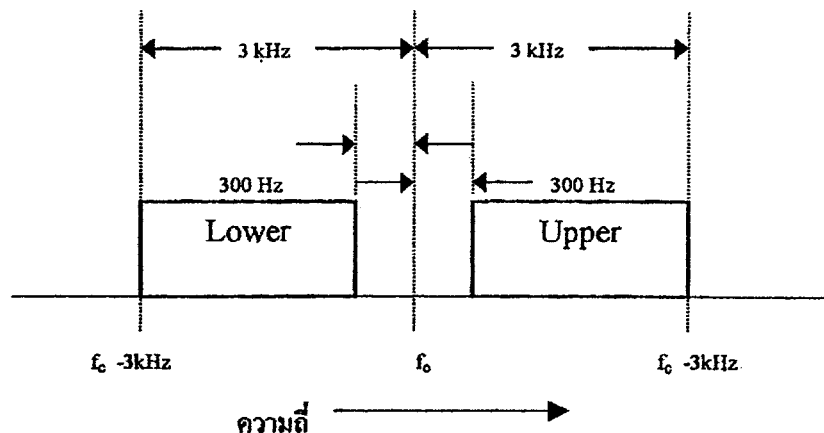
$$v_2 = V_c \sin(2\pi f_c t) + 1/2 V_s \cos\pi t(f_c - f_s) - V_s \cos 2t(f_c + f_s)$$

จากสมการข้างต้นชี้ให้เห็นว่า ผลที่ได้จากการทำแอมพลิจูดมอดูเลชั่นจะได้สัญญาณที่ความถี่ใกล้เคียงกับสัญญาณพาหะ (Carrier) ออกมาด้วย เป็นสัญญาณที่ความถี่ต่ำและสูงกว่าสัญญาณพาหะ เรียกว่า ไซด์แบนด์หรือไซด์ฟรีควนซ์ (Sideband or Side Frequency)

$$f_{\text{USB}} = \text{Upper Sideband Frequency} = (f_c + f_s)$$

$$f_{\text{LSB}} = \text{Lower Sideband Frequency} = (f_c - f_s)$$

จากการวัดด้วยออสซิลอโคป (Oscilloscope) สัญญาณที่ได้จะไม่สามารถแยกได้ว่ามีความถี่ข้างต้นเกิดขึ้นด้วย แต่ถ้าใช้เครื่องมือสเปกตรัมอนาลิเซอร์ (Spectrum Analyzer) วิเคราะห์สัญญาณของแอมพลิจูดมอดูเลชั่น จะได้รูปกราฟของฟรีควนซ์โดเมน (Frequency Domain) ดังรูป



รูปที่ 2.4 สัญญาณแอมพลิจูดมอดูเลชั่นในแกนความถี่

จากรูปทำให้ทราบว่า สัญญาณที่ได้จากการมอดูเลตจะมีความถี่ใกล้เคียงเกิดขึ้นด้วยทั้งด้านบนและด้านล่างของความถี่พาหะ และจะได้ว่า AM ใช้ความกว้างของแถบสัญญาณหรือที่เรียกว่าแบนด์วิดท์ (Bandwidth) เป็นจำนวนสองเท่าของความถี่สูงสุดของสัญญาณข้อมูล ถ้าหากสัญญาณข้อมูลมีความถี่  $f_s = 5 \text{ kHz}$  จะได้ว่า ต้องใช้แบนด์วิดท์เท่ากับ  $10 \text{ kHz}$  สำหรับสัญญาณ AM

เมื่อพิจารณาถึงกำลัง (Power) ที่ส่งออกอากาศ จะประกอบด้วยกำลังของคลื่นพาหะบวกกับกำลังของคลื่นความถี่ข้างเคียงทั้งสองด้าน (Sideband) เนื่องจากกำลังส่งของความถี่ข้างเคียงมีค่า ขึ้นอยู่กับค่าของดัชนีมอดูเลชัน ( $m$  : Modulation Index) และค่าของกำลังส่งไซด์แบนด์มีค่าดังนี้

$$P_{LSB} = P_{USB} = P_c(m)^2/4$$

|       |           |   |  |
|-------|-----------|---|--|
| เมื่อ | $P_c$     | = | ค่ากำลังของสัญญาณพาหะ                        |
|       | $P_{LSB}$ | = | ค่ากำลังส่งของสัญญาณด้านที่ต่ำกว่าสัญญาณพาหะ |
|       | $P_{USB}$ | = | ค่ากำลังส่งของสัญญาณด้านที่สูงกว่าสัญญาณพาหะ |

จากสมการของกำลังข้างต้นทำให้ทราบว่า ยิ่งเปอร์เซ็นต์มอดูเลชันสูงมากเท่าใด กำลังส่งของสัญญาณข้างเคียงก็จะยิ่งสูง ทำให้การรับและการส่งของสัญญาณดีขึ้น

แม้ว่า AM จะมีความง่ายและได้ผลที่ดีวิธีหนึ่ง แต่เป็นการส่งสัญญาณที่ไม่ค่อยมีประสิทธิภาพ หากพิจารณากำลังส่งทั้งหมด เมื่อ  $P_t$  = กำลังส่งสัญญาณทั้งหมด

$$P_t = P_c + P_{LSB} + P_{USB}$$

$$P_t = P_c + P_c(m)^2/4 + P_c(m)^2/4$$

ที่  $m = 1$  หรือ 100% มอดูเลชันจะได้สัญญาณไซด์แบนด์ที่มีกำลังสูงสุด

$$P_t = P_c + P_c/2$$

หากกำลังส่งของสัญญาณพาหะเป็น 100 วัตต์ กำลังส่งของสัญญาณไซด์แบนด์จะมีค่า 50 วัตต์ จากผลรวมทั้งหมด 150 วัตต์ ซึ่งเป็นสัญญาณข้อมูลเพียง 50 วัตต์ มีการสูญเสียไปกับกำลังส่ง ของสัญญาณถึง 2 ใน 3 ของพลังงานทั้งหมด โดยไม่มีข้อมูลใดๆ อยู่ในสัญญาณนี้เลย ด้วยสาเหตุข้างต้นจึงมีการลดหรือเอาสัญญาณพาหะออกไปซึ่งเรียกว่าการซัพเพรสชัน (Suppression) ทำให้สัญญาณที่ได้มีเฉพาะสัญญาณไซด์แบนด์ สัญญาณ AM ที่มีการเอาสัญญาณพาหะออก เรียกว่าดับเบิลไซด์แบนด์ซัพเพรสส์แคร์เรียร์ (Double sideband suppressed carrier : DSB) วงจรที่ใช้ในการสร้าง DSB เรียกว่า บาลานซ์มอดูเลเตอร์ (Balanced Modulator) ด้วยการ ใช้ DSB จะทำให้ได้กำลังส่งในสัญญาณไซด์แบนด์ได้มากขึ้น และหากพิจารณาต่อไปอีกจะพบว่าสัญญาณข้อมูลนั้นมีการซ้อนกันทั้งความถี่บน (Upper

Sideband) และความถี่ด้านล่าง (Lower Sideband) จึงมีการคิดค้นให้ส่งสัญญาณเพียงด้านเดียว เรียกว่า ซิงเกิลไซด์แบนด์ซัพเพรสส์แคร์เรีย (Single Sideband Suppressed Carrier : SSSC หรือ SSB) เมื่อมีการใช้สัญญาณเพียงข้างเดียว ทำให้สามารถประหยัดการครอบครองแถบความถี่ (Spectrum) ไปได้ครึ่งหนึ่ง จากเดิมที่สัญญาณ AM ต้องการใช้ทั้ง DSB และ SSB มีประสิทธิภาพมากกว่าในแง่ของกำลังส่งที่ใช้ เนื่องจากมีการประหยัดกำลังส่งของสัญญาณพาหะ ทำให้สามารถนำกำลังส่งไปใส่ในไซด์แบนด์ได้

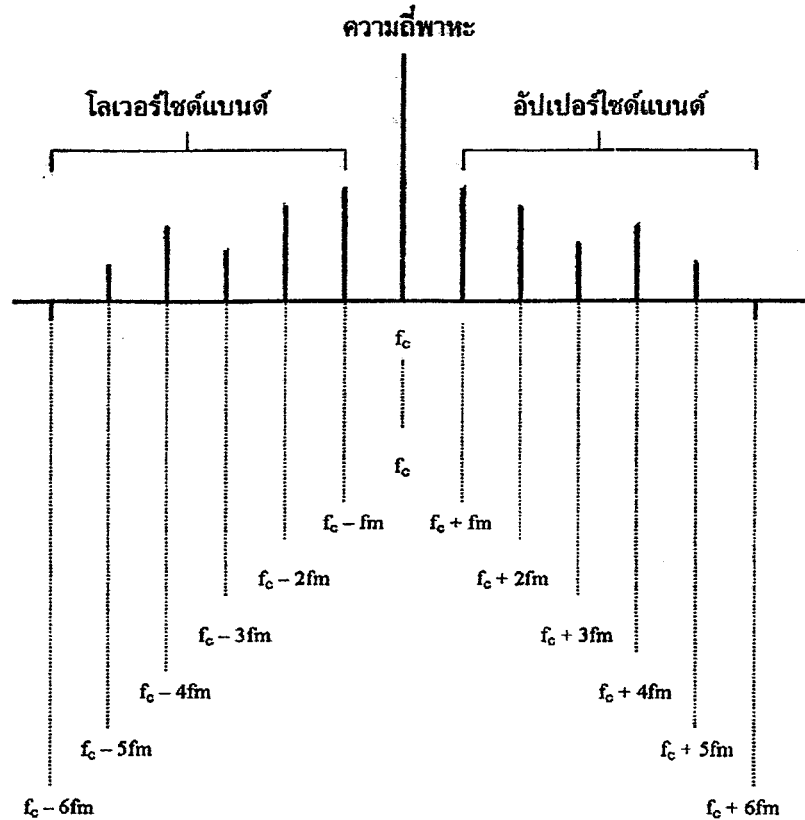
สิ่งที่เกี่ยวข้องกับการมอดูเลชันทางความถี่อีกเรื่องหนึ่งคือ ฟริควเอนซีคอนเวอร์ชัน (Frequency Conversion) เป็นการแปลงสัญญาณข้อมูลหรือเสียงพูดไปยังความถี่สูงหรือต่ำกว่า ในขณะที่ยังคงรักษาข้อมูลเบื้องต้นไว้ทั้งหมด เป็นเพียงการเลื่อนความถี่ไปเท่านั้น ในการแปลงความถี่ให้สูงขึ้นนั้น มีการใช้งานในการส่งสัญญาณออกอากาศ ส่วนการแปลงความถี่ให้ต่ำลงนั้น มักมีใช้ในเครื่องรับวิทยุเพื่อทำให้สามารถเลือกช่องสัญญาณ (Selectivity) ได้สะดวกขึ้น และเพิ่มค่ากำลังขยาย (Gain) ให้ดีขึ้น ซึ่งฟริควเอนซีคอนเวอร์ชันก็เป็นรูปแบบหนึ่งของ แอมพลิฟิเคชันมอดูเลชัน และกระทำได้โดยใช้วงจรที่เรียกว่ามิกเซอร์ (Mixer) ในอุปกรณ์บางชนิด มิกเซอร์หมายถึงตัวแปลง (Converter) และฟังก์ชันการทำงานของมิกเซอร์ เรียกว่า เฮเทอโรไดน์ (Heterodyning)

#### 2.4.2 การมอดูเลชันทางความถี่ (Frequency Modulation)

เป็นการเปลี่ยนความถี่ของสัญญาณพาหะตามการเปลี่ยนแปลงของสัญญาณข้อมูล โดยที่ค่าแอมพลิจูดของสัญญาณยังคงที่ การเปลี่ยนแปลงความถี่ของสัญญาณพาหะจะมีความสัมพันธ์กับค่าแอมพลิจูดของสัญญาณ ข้อมูล เช่น เมื่อสัญญาณข้อมูลมีแอมพลิจูดสูง สัญญาณพาหะก็จะมีความถี่สูงขึ้นด้วย ขณะที่ไม่มีสัญญาณข้อมูลเข้ามาทำการมอดูเลตสัญญาณพาหะจะมีค่าความถี่กึ่งกลางอยู่ตำแหน่ง ซึ่งเรียกว่า เซนเตอร์ฟริควเอนซี (Center Frequency) เมื่อสัญญาณข้อมูลเข้ามาก็จะทำความถี่เปลี่ยนแปลงสูงขึ้น ถูกเรียกว่าค่าความถี่เบี่ยงเบน (Frequency Deviation) ค่าสูงสุดของความถี่เบี่ยงเบนจะเกิดขึ้นเมื่อสัญญาณข้อมูลมีแอมพลิจูดสูงสุด ความถี่ของสัญญาณข้อมูล จะเป็นตัวกำหนดให้ทราบถึงจำนวนครั้งในหนึ่งวินาทีที่เกิดการเบี่ยงเบนความถี่ขึ้น ไปสูงและต่ำกว่าค่าความถี่กลาง และอัตราดังกล่าวถูกเรียกว่าอัตราความถี่เบี่ยงเบน (Frequency Deviation Rate) ความถี่ของสัญญาณข้อมูลจะมีผลต่ออัตราการเบี่ยงเบนทางความถี่ดังกล่าว แต่จะไม่มีผลต่อปริมาณของความถี่เบี่ยงเบนไปจากค่าความถี่กลาง ซึ่งขึ้นอยู่กับแอมพลิจูดของสัญญาณข้อมูลเท่านั้น

## 2.5 ความถี่ข้างเคียงและดัชนีการมอดูเลตขั้น

ในการมอดูเลตทางความถี่จะทำให้เกิดความถี่ข้างเคียงขึ้น เป็นความถี่ของผลรวมและผลต่างของสัญญาณพาหะและสัญญาณข้อมูล ตามทฤษฎีแล้วจะเกิดคู่ของความถี่มีการใช้งานแถบความถี่ ( Spectrum ) มากกว่าสัญญาณที่มอดูเลตทางแอมพลิจูด



รูปที่ 2.5 แสดงแถบความถี่ของของสัญญาณ FM

จากรูปจะเห็นว่า แถบความถี่ของสัญญาณมอดูเลตทางความถี่โดยทั่วไปที่สัญญาณพาหะถูกมอดูเลตด้วยสัญญาณรูปไซน์ที่มีความถี่เดียว จะเห็นว่าความถี่ข้างเคียงจะเว้นระยะห่างจากสัญญาณความถี่พาหะและความถี่ข้างเคียงด้วยกันเป็นระยะเท่าๆกัน เท่ากับความถี่สัญญาณข้อมูล และแอมพลิจูดของสัญญาณข้างเคียง ไม่เท่ากัน หากสมมุติว่าสัญญาณความถี่ข้างเคียงต่างๆ เป็นสัญญาณรูปไซน์ที่มีความถี่และแอมพลิจูดเฉพาะตัวเมื่อนำมารวมกันจะได้ผลลัพธ์เป็นสัญญาณที่ถูกมอดูเลตทางความถี่นั่นเอง

ลักษณะต่างๆของจำนวนความถี่ข้างเคียง ขนาดแอมพลิจูดของความถี่ข้างเคียง และระยะห่างระหว่างความถี่จะขึ้นอยู่กับค่าเบี่ยงเบนความถี่และความถี่ของสัญญาณข้อมูล แม้ว่าสัญญาณมอดูเลตทางความถี่จะมีความถี่ข้างเคียงจำนวนอนันต์ แต่ความถี่ข้างเคียงที่มีแอมพลิจูดสูงเพียงพอเท่านั้นจึงจะถือว่าเป็นสัญญาณที่มีข้อมูลอยู่ โดยปรกติสัญญาณที่มีแอมพลิจูดต่ำกว่า 1% ของสัญญาณพาหะที่ยังไม่ถูกมอดูเลต จะถือว่าไม่มีความสำคัญ

จากความสัมพันธ์ข้างต้น ทำให้เกิดอัตราส่วนระหว่างค่าเบี่ยงเบนความถี่ของสัญญาณข้อมูลซึ่งเรียกว่าดัชนีการมอดูเลต ( Modulation Index : m )

$$m = f_d / f_m$$

|       |       |   |                           |
|-------|-------|---|---------------------------|
| เมื่อ | $f_d$ | = | ค่าความถี่เบี่ยงเบน       |
|       | $F_m$ | = | ค่าความถี่ของสัญญาณข้อมูล |

ค่าเบี่ยงเบนความถี่และค่าความถี่ของสัญญาณข้อมูลมักจะใช้ค่ามากที่สุด เพื่อนำมาคำนวณดัชนีการมอดูเลต และนอกจากนั้นค่าดัชนีดังกล่าวก็ถูกเรียกว่าอัตราการเบี่ยงเบนด้วย

สัญญาณที่ถูกมอดูเลตทางความถี่ ( FM ) จะใช้แถบความถี่กว้างเท่าใดขึ้นอยู่กับดัชนีการมอดูเลต หากค่าดัชนีการมอดูเลตมากก็จะใช้แถบความถี่กว้างมาก ดังนั้นสามารถประหยัดแถบความถี่ได้ด้วยการจำกัด ค่าของดัชนีการมอดูเลต

แถบความถี่กว้างของความถี่ ( Bandwidth ) =  $2f_m$  x จำนวนของความถี่ข้างเคียงที่สำคัญ

## 2.6 เครื่องส่งสัญญาณคลื่นวิทยุ ( Radio Transmitters )

จุดเริ่มต้นของการส่งคลื่นวิทยุ นั้น ประกอบด้วย การสร้างสัญญาณที่เป็นสื่อหรือพาหะที่เรียกว่า แครีเรียร์เจเนอเรชัน ( Carrier Generation ) นำสัญญาณมามอดูเลตกับสัญญาณข้อมูลหรือเสียง แล้วจึงทำการขยายสัญญาณเพื่อส่งออกไปยังสายอากาศหรือตัวนำคลื่นเป็นสัญญาณความถี่วิทยุ ( RF )

### 2.6.1 วงจรสัญญาณพาหะ ( Carrier Generation )

โดยส่วนมากเป็นวงจรคริสตัลอสซิลเลเตอร์ ( Crystal Oscillator ) ซึ่งจะให้สัญญาณความถี่ที่ต้องการได้เที่ยงตรงและมีเสถียรภาพดี โดยส่วนมากมักมีการต่อวงจรขยายบัฟเฟอร์เข้าไปเพื่อแยกวงจรออสซิลเลเตอร์ออกจากโหลด เป็นการป้องกันการเปลี่ยนความถี่เนื่องจากค่าโหลดของวงจรออสซิลเลเตอร์มีการเปลี่ยนแปลงไป

### 2.6.2 วงจรมอดูเลตสัญญาณ ( Modulator )

ทำการแปลงคุณสมบัติของสัญญาณพาหะให้มีลักษณะตามการเปลี่ยนแปลงของข้อมูลหรือเสียงพูดที่ต้องการส่ง

### 2.6.3 วงจรขยาย ( Amplifier )

เป็นการขยายสัญญาณในรูปแบบต่างๆ ในขั้นตอนที่แตกต่างกัน สำหรับวงจรขยายมีการจัดการออกเป็นหลายแบบ มีการเรียกเป็น ( Class ) ตามวิธีการไบอัสของวงจรขยายดังนี้

### วงจรรขยายคลาสเอ ( Class A )

เป็นวงจรรขยายที่ต่อทรานซิสเตอร์แบบที่มีกระแสไหลผ่านขาคอลเลคเตอร์ ( Collector ) หรือที่เรียกว่ากระแสเดรน ( Drain Current ) เป็นวงจรรขยายแบบเชิงเส้น ( Linear Amplifier ) เนื่องจากสัญญาณที่ได้ในขาออก เป็นสัดส่วนโดยตรงกับสัญญาณขาเข้า แต่วงจรรขยายคลาสเอ ก็เป็นวงจรที่ไม่ค่อยมีประสิทธิภาพ เนื่องจากวงจรมีการขยายสัญญาณ โดยการทำงานตลอดทุกคลื่นสัญญาณขาเข้า หรือที่เรียกว่าครบ 360 องศา ดังนั้นวงจรรขยาย คลาสเอ จึงไม่เหมาะที่จะเป็นวงจรรขยายกำลัง ( Power Amplifier ) เพราะ โดยปรกติมักใช้ในวงจรรขยาย ขั้นต้นที่มีสัญญาณความถี่ต่ำหรือวงจรรขยายกำลังต่ำ ( Low – Power Amplifier ) ตัวอย่างเช่น วงจรรขยายแบบบัฟเฟอร์ ( Buffer Amplifier )

### วงจรรขยายคลาสบี ( Class B )

เป็นวงจรรขยายที่ต่อทรานซิสเตอร์ แบบที่มีกระแสไหลให้ทรานซิสเตอร์อยู่ในช่วงคัตออฟ ( Cutoff ) เป็นสถานะที่ไม่มีกระแสไหล โดยปรกติเมื่อไม่มีสัญญาณขาเข้าก็จะมีกระแสไหลที่ขาคอลเลคเตอร์ ตัวทรานซิสเตอร์จะทำงานนำกระแสเพียงช่วงครึ่งลูกของสัญญาณขาเข้าที่ต้องการขยาย จึงมีการต่อวงจรที่เรียกว่า พูชแอนด์พูล ( Push and Pull ) โดยใช้วงจรรขยายคลาสบี 2 ชุด ทำงานทั้งในช่วงสัญญาณบวกและสัญญาณลบ สลับต่อเนื่องกันไป วงจรรขยายคลาสบีมีประสิทธิภาพดีกว่าวงจรรขยายคลาสเอ เนื่องจากการไหลของกระแส จะเกิดเพียงช่วงหนึ่งของสัญญาณเท่านั้น ซึ่งเหมาะสำหรับวงจรรขยายกำลัง แต่สัญญาณที่ได้ก็ยังมีบิดเบือน ( Distortion ) ดังนั้นจึงมีการต่อวงจรรขยายแบบพูชแอนด์พูลเพื่อลดความบิดเบือนของสัญญาณด้วย

### วงจรรขยายคลาสบี ( Class AB )

เป็นวงจรรขยายที่มีกระแสไหลให้ทรานซิสเตอร์ให้อยู่ในช่วงที่เกือบจะคัตออฟจึงมีกระแสไหลที่ขาคอลเลคเตอร์เพียงเล็กน้อย ทำให้เมื่อมีสัญญาณขาเข้ามาถึงก็จะทำงานในช่วงของสัญญาณที่มากกว่า 180 องศาแต่ไม่ถึง 360 องศาของลูกคลื่นขาเข้า และก็มีมีการต่อใช้ในรูปของวงจรรขยายแบบพูชแอนด์พูล เช่นเดียวกับคลาสบี ซึ่งทำให้มีความบิดเบือนของสัญญาณน้อยกว่าในคลาสบี นั่นคือมีความเป็นเชิงเส้นมากกว่าแต่ยังมีประสิทธิภาพน้อยกว่าในแบบคลาสบี

วงจรรขยายคลาสเอ คลาสบี คลาสเอบี เป็นวงจรรขยายเชิงเส้นที่มักใช้ในการขยายสัญญาณคลื่นวิทยุที่มีการเปลี่ยนแปลงแอมพลิจูด เช่น วงจรรขยาย AM แบบกำลังต่ำหรือแบบซิงเกิลไซด์แบนด์ ( SSB ) วงจรรขยายแบบไม่เชิงเส้น เช่น วงจรรขยายคลาสซี ( Class C ) ที่เป็นวงจรซึ่งใช้มากในเครื่องส่งแบบ AM และ FM สำหรับการขยายกำลังในรูปของวงจรรขับ ( Driver ) วงจรควบคุมความถี่ และวงจรรขยายภาคสุดท้าย

### วงจรรขยายคลาสซี ( Class C )

เป็นวงจรรทรานซิสเตอร์ที่ถูกไบแอสให้มีการนำสัญญาณเพียงส่วนน้อยกว่า 180 องศา ของลูกคลื่นขาเข้า วิธีการไบแอสทรานซิสเตอร์สำหรับคลาสซีมีอยู่ด้วยกัน 3 วิธี คือ ไบแอสด้วยสัญญาณ ( Signal Bias ) ไบแอสด้วยวงจรรภายนอก ( External Bias ) และ ไบแอสด้วยตนเอง ( Self Bias ) และปรกติมีช่วงทำงานของวงจรรอยู่ในช่วง 90 – 150 องศา นั้นหมายถึงมีเพียงสัญญาณพัลส์เล็กๆเท่านั้น

ออกมาที่ขาออก ดังนั้นการทำให้มีสัญญาณขยายเต็มลูกคลื่นจะต้องมีการนำวงจรเรโซแนนซ์ ( Resonant Tuned Circuit ) มาต่อที่ขาออกเพื่อที่จะได้สัญญาณลูกคลื่นขยายเต็มลูกคลื่น ที่เหนือกว่าคลาสอื่นๆ ก็คือ คลาสซีที่มีการขยายสัญญาณขาเข้าเพียงช่วงสั้นๆเท่านั้น

#### วงจรขยายคลาสดี ( Class D )

เป็นวงจรขยายแบบสวิตช์ เหตุที่เรียกชื่อดังที่กล่าวมานี้ก็เนื่องจากว่าทรานซิสเตอร์จะทำงานในลักษณะออนและออฟสลับกัน ทำให้มีการสูญเสียกำลังงานน้อย

#### 2.6.4 วงจรอิมพีแดนซ์แมตชิ่ง ( Impedance Matching Circuit )

เป็นวงจรที่ใช้สำหรับเชื่อมต่อระหว่างวงจรขยายความถี่วิทยุ ( RF Amplifier ) ในแต่ละภาค เพื่อให้ได้กำลังส่งที่มากที่สุด การที่จะทำให้มีการถ่ายพลังงานมากที่สุดจากวงจรขยายชุดหนึ่งไปยังอีกชุดหนึ่ง จะต้องมีค่าของอิมพีแดนซ์ของวงจรแรกเท่ากับค่าอิมพีแดนซ์ขาเข้าของวงจรถัดไป วงจรอิมพีแดนซ์แมตชิ่ง โดยทั่วไปเป็นวงจรของตัวนำและตัวเก็บประจุ LC ( Inductors and Capacitors ) ที่มีรูปแบบการต่อต่างๆกัน เช่น โครงข่ายรูป L และ โครงข่ายรูป T

#### 2.6.5 เครื่องรับสัญญาณ ( Communication Receives )

หน้าที่ของเครื่องรับสัญญาณคือทำการเลื่อนช่องสัญญาณที่ต้องการออกมาจากสัญญาณอื่นๆ ที่ถูกส่งออกมาในอากาศและขยายสัญญาณกลับไปเป็นสัญญาณข้อมูลที่ส่งมาได้ โดยปกติเครื่องรับจะมีปัจจัย 2 ประการที่ต้องคำนึงถึง ดังนี้

##### ค่าซีเล็กติวิตี ( Selectivity )

หมายถึง ความสามารถในการรับสัญญาณ โดยเลือกเอาเฉพาะช่องสัญญาณที่ต้องการเข้ามาเท่านั้น หากเครื่องรับสัญญาณมีค่าซีเล็กติวิตีที่ดีก็จะสามารถรับสัญญาณช่องที่ต้องการและกำจัดช่องสัญญาณข้างเคียงออกไปได้

##### ค่าเซนซิวิตี ( Sensitivity )

หมายถึง ความสามารถในการรับสัญญาณที่ต้องการที่มีขนาดเล็กหรือสัญญาณอ่อนแล้วนำมาขยายให้ได้สัญญาณที่มีความแรงมากขึ้น โดยปรกติค่าเซนซิวิตีจะแสดงถึงการขยายสัญญาณด้วย นั่นคือยังมีกำลังขยายมาก ค่าเซนซิวิตีก็ยิ่งดี และจะแสดงอยู่ในรูปของค่าแรงดันของสัญญาณขาเข้า ที่มีขนาดเล็กที่สุดซึ่งจะสามารถขยายได้มากกว่า 10 เท่าของสัญญาณรบกวน

## 2.7 ระบบสัญญาณ

สัญญาณ (Signaling) คือ ข่าวสารที่ใช้ติดต่อกันระหว่างเครื่องโทรศัพท์กับชุมสาย หรือข่าวสารที่ใช้ติดต่อกันระหว่างชุมสายกับชุมสาย

### 2.7.1 สัญญาณระหว่างผู้เข้ากับชุมสาย (Subscriber Signaling)

#### สัญญาณที่ส่งจากผู้เข้าไปยังชุมสาย

- Off Hook คือ สภาพผู้เช่ายกหูโทรศัพท์ สายจะมีสภาพ Closed Loop (Low Impedance)
- On Hook คือ สภาพผู้เช่าวางหูหรือสภาพว่าง สายจะมีสภาพ Open Loop (High Impedance)
- Dialing คือ สภาพที่ผู้เช่าหมุนหมายเลขเข้าเครื่องเป็น Rotary Dial สัญญาณจะเป็น Pulsing

ค่าความต้านทานจะสูง-ต่ำ สลับกันไปตามที่หมุนหมายเลข ถ้าเครื่องเป็นแบบกดปุ่ม Touch Tone สัญญาณออกจะเป็นความถี่ DTMF ส่งออกไปยังชุมสาย

#### สัญญาณที่ส่งจากชุมสายไปยังเครื่องรับ

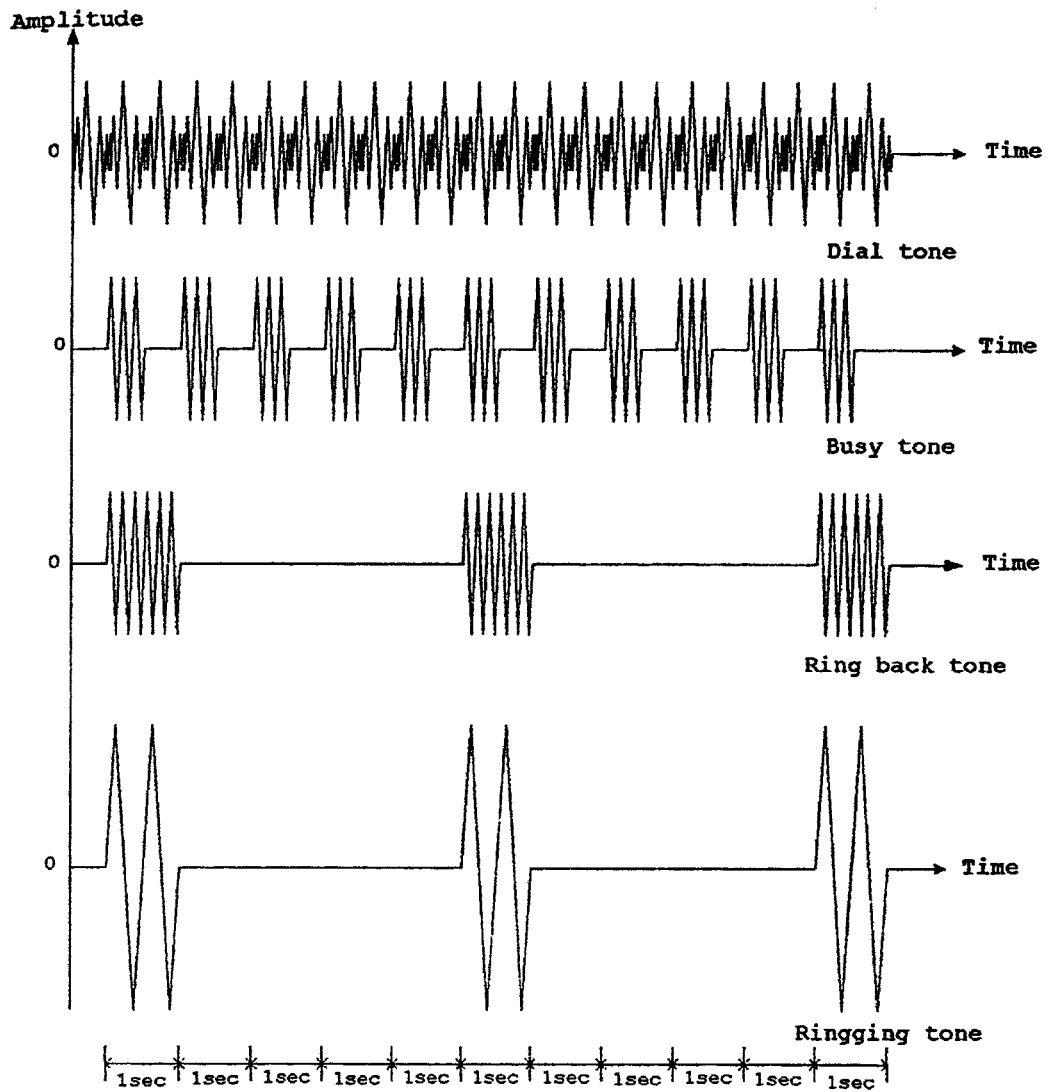
- สัญญาณให้หมุน (Dial Tone) เป็นสัญญาณเพื่อแสดงให้ผู้เรียกทราบว่า ขณะนี้ผู้ใช้สามารถที่จะเรียกไปยังหมายเลขที่ต้องการได้ ซึ่งเป็นสัญญาณความถี่ต่อเนื่องขนาด 400 Hz มอดูเลตกับความถี่ 50 Hz แบบ AM

- สัญญาณไม่ว่าง (Busy Tone) เป็นสัญญาณเพื่อให้ผู้เรียกทราบว่า หมายเลขที่ต้องการติดต่อด้วยในขณะนี้ยังไม่ว่างควรจะวางหูระยะหนึ่งก่อนแล้วจึงทำการติดต่อไปใหม่อีกครั้ง ซึ่งเป็นสัญญาณความถี่ขนาด 400 Hz โดยมีช่วงเวลากการส่งประมาณ 0.5 วินาที แล้วหยุดประมาณ 4 วินาที สลับกัน

- สัญญาณเรียกกลับ (Ring Back Tone) เป็นสัญญาณเพื่อแสดงให้ผู้เรียกทราบว่า สามารถที่จะติดต่อกับผู้ที่จะสนทนาด้วยได้แล้ว แต่ยังคงอยู่ในระหว่างรอการยกหูจากปลายทางซึ่งเป็นสัญญาณความถี่ต่อเนื่องขนาด 400 Hz โดยมีช่วงเวลากการส่งประมาณ 1 วินาที แล้วหยุดประมาณ 4 วินาที สลับกัน

- สัญญาณกระดิ่งเรียก (Ringing Tone) ใช้พร้อมกับสัญญาณ Ring Back Tone เมื่อมีสัญญาณกระดิ่งเรียกดังขึ้นก็จะมีสัญญาณเรียกกลับดังขึ้นพร้อมๆกัน แต่สัญญาณนี้จะมีเสียงดังมากเพื่อให้วงจรเสียงกระดิ่งในเครื่องโทรศัพท์ทำงานได้ ซึ่งเป็นสัญญาณความถี่ขนาด 50 Hz โดยมีช่วงเวลากการส่งประมาณ 1 วินาที แล้วหยุดประมาณ 4 วินาที สลับกันเหมือนกับ Ring Back Tone

- สัญญาณโทนอื่นๆ เช่น Nu Tone (Number Unobtained Tone) บอกให้ทราบว่าหมายเลขที่หมุนไปนั้นยังไม่มีบริการใช้งานลักษณะของสัญญาณจะเป็นความถี่ประมาณ 400 Hz โดยมีช่วงเวลากการส่งประมาณ 1 วินาที แล้วหยุดประมาณ 4 วินาที อย่างต่อเนื่อง



รูปที่ 2.6 สัญญาณพื้นฐานของชุมสายโทรศัพท์

### 2.7.2 สัญญาณติดต่อระหว่างชุมสายกับชุมสาย (Inter Exchange Signaling)

สัญญาณพื้นฐานมีอยู่ 5 ประเภทคือ

1. Seizer (สัญญาณจับเวลา) เป็นสัญญาณให้ชุมสายปลายทางทราบว่า คู่สายขณะนี้ถูกใช้งานอยู่ ชุมสายปลายทางจะจัดการจัดเตรียมอุปกรณ์ที่รับเลขหมายของผู้ถูกเรียกที่จะส่งมา
2. Address Information เป็นสัญญาณบอกเลขหมายหรือประเภทของผู้เช่า
3. Address Signal (สัญญาณตอบรับ) สัญญาณนี้ถูกส่งเมื่อผู้ถูกเรียกทำการยกหูรับ
4. สัญญาณยกเลิกการต่อตรง (Clear Forward) จะถูกส่งเมื่อผู้เรียกทำการวางหู ผลของสัญญาณนี้จะทำให้วงจรทางด้านปลายทำการยกเลิกการต่อวงจรต่างๆ
5. สัญญาณยกเลิกการตอบกลับ (Clear Back) จะถูกส่งเมื่อผู้ถูกเรียกทำการวางหู ผลของสัญญาณนี้จะทำให้ชุมสายต้นทางเริ่มต้นจับเวลา เมื่อเวลาผ่านไปประมาณ 90-120 วินาที ชุมสายต้นทางจะ

ยกเลิกการติดต่อมาพร้อมกับส่งสัญญาณ Clear Forward ออกไปเพื่อให้ชุมสายปลายทางยกเลิกเช่นกัน

### 2.7.3 สัญญาณติดต่อกันระหว่างผู้เรียก (Calling) และผู้ถูกเรียก (Called)

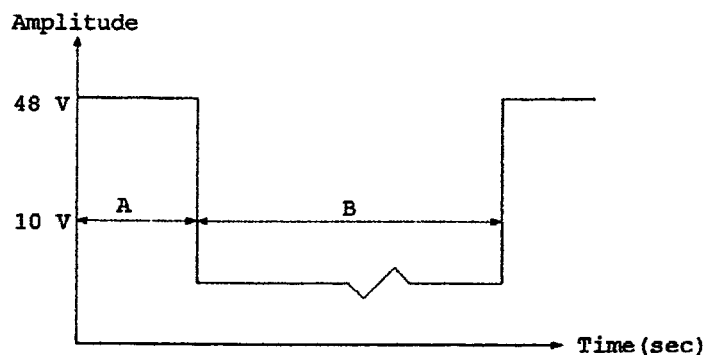
ขั้นตอนการทำงานของโทรศัพท์แบ่งออกเป็น 2 กรณี คือ ผู้เรียกและผู้ถูกเรียก

#### กรณีผู้เรียก (Calling subscriber)

ขณะที่โทรศัพท์ที่วางอยู่นั้น ที่คู่สายโทรศัพท์จะมีกระแสไฟตรงตกคร่อมอยู่ 48 โวลต์ และเมื่อหูโทรศัพท์ที่ถูกยกขึ้น กระแสไฟตรงที่คร่อมอยู่นั้นจะตกลงมาเหลือประมาณ 5-10 โวลต์ ทั้งนี้ขึ้นอยู่กับระบบชุมสายโทรศัพท์ย่อยแต่ละพื้นที่ ขณะเดียวกันนั้นก็จะมีสัญญาณเสียงให้หมุนหมายเลข(Dial Tone) ถ้าหมายเลขที่ถูกเรียกไม่ว่าง ผู้เรียกก็จะได้ยินเสียงสัญญาณไม่ว่าง (Busy Tone) และถ้าได้ยินเสียงเรียกกลับ (Ring Back Tone) ตอบกลับมาก็แสดงว่าเลขหมายที่เรียกพร้อมที่จะทำการสนทนาได้เพียงแต่รอจนกว่าผู้ถูกเรียกจะทำการยกหูรับ ก็สามารถเริ่มทำการสนทนาได้

#### กรณีผู้ถูกเรียก (Called subscriber)

ขณะที่โทรศัพท์ที่วางอยู่นั้นจะมีกระแสไฟตกคร่อมอยู่ 48 โวลต์ และเมื่อมีการถูกเรียกจากคันทาง ทางชุมสายจะทำการเชื่อมต่อให้และจะส่งสัญญาณกระดิ่ง (Ringing Tone) เป็นแรงดันไฟสลับประมาณ 110-150 โวลต์ และเมื่อมีการยกหูโทรศัพท์ ซึ่งจะทำให้เกิดค่าความต้านทานประมาณ 600 โอห์ม ต่อเข้ากับชุมสาย (แรงดันตกคร่อมจะลดลงเหลือประมาณ 5-10 โวลต์) ทำให้ชุมสายรู้ว่ามีการยกหูโทรศัพท์แล้วชุมสายก็จะหยุดส่งสัญญาณกระดิ่ง และทำการต่อคู่สาย โทรศัพท์ให้เพื่อเริ่มต้นการสนทนา

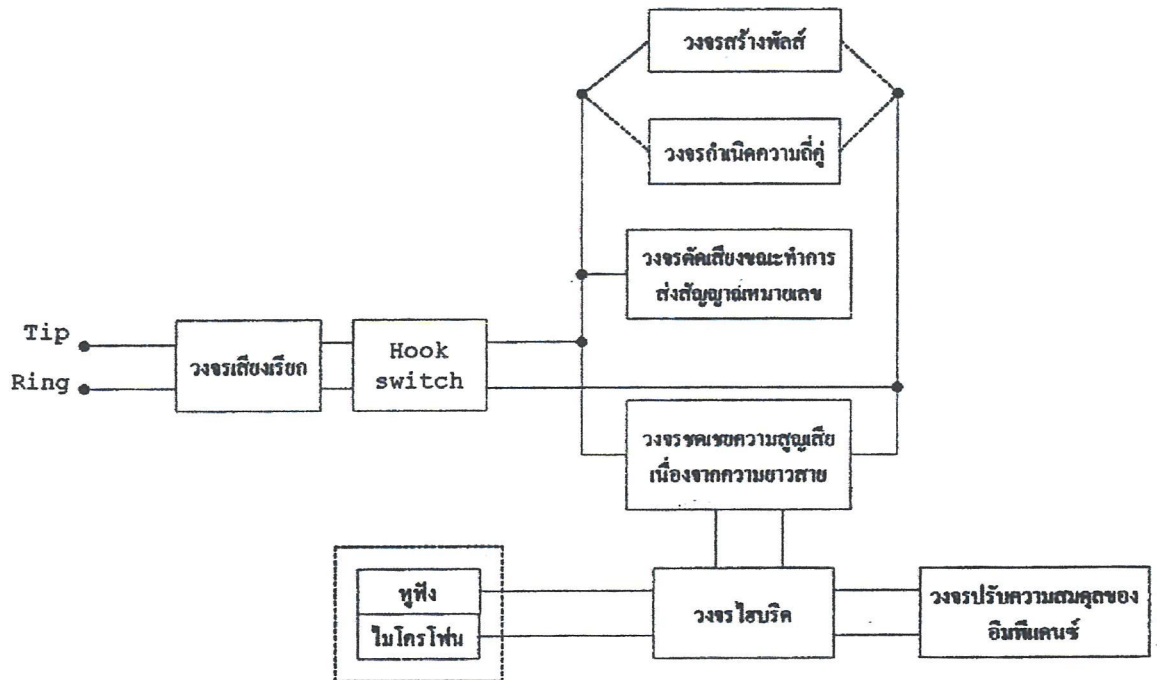


รูปที่ 2.7 กระแสไฟตรงที่เลี้ยงคู่สาย ขณะยกหูและวางหู

## 2.8 การทำงานของโทรศัพท์

รูปที่ 2.8 เป็นบล็อกไดอะแกรมการทำงานของโทรศัพท์ ไม่ว่าจะเป็นระบบพัลส์หรือโทนก็มีลักษณะเหมือนกัน แต่แตกต่างกันตรงที่การกำเนิดสัญญาณเลขหมายว่าจะป็นระบบพัลส์หรือ DTMF วงจรเสียงเรียกทำหน้าที่แจ้งให้ผู้โทรศัพท์ทราบว่ามีการเรียกเข้ามา สุกสวิทช์เป็นค้วบอกให้ชุมสาย

โทรศัพท์รับรู้ว่ามีการยกหูใช้งานโทรศัพท์แล้ว ก็จะทำการตัดต่อคู่สายให้ติดต่อกันได้ ในส่วนของวงจรตัดเสียงขณะทำการส่งหมายเลขจะช่วยให้การส่งหมายเลขให้มีความชัดเจนถูกต้องไม่ถูกรบกวนด้วยสัญญาณเสียงพูด



รูปที่ 2.8 การทำงานของเครื่องโทรศัพท์

ในขณะที่วงจรชดเชยความสูญเสียเนื่องจาก ความยาวของสายส่ง จะทำให้สัญญาณที่ติดต่อระหว่างต้นทางและปลายทางมีความแรงและชัดเจนมากที่สุด แม้ว่าต้นทางและปลายทางจะมีระยะห่างกันแค่ไหนก็ตามส่วนวงจรไฮบริดทำหน้าที่เสมือนเป็นวงจรขยาย 2 ทิศทาง หรือสามารถให้สัญญาณผ่านเข้าออกได้ตลอดเวลาจึงมีเสียงจากปลายทางมาปรากฏที่หูฟัง ในขณะที่สัญญาณจากปากพูดก็สามารถผ่านออกไปทางคู่สายได้และวงจรไฮบริดยังทำหน้าที่ลดสัญญาณไซด์โทนหรือเสียงสะท้อนได้อีกด้วย ส่วนวงจรปรับความสมดุลของความต้านทาน มีไว้เพื่อทำให้ความต้านทานของส่วนต่างๆ ในโทรศัพท์มีความเหมาะสมเพื่อการถ่ายทอดสัญญาณเป็นไปได้อย่างมีประสิทธิภาพมากที่สุด

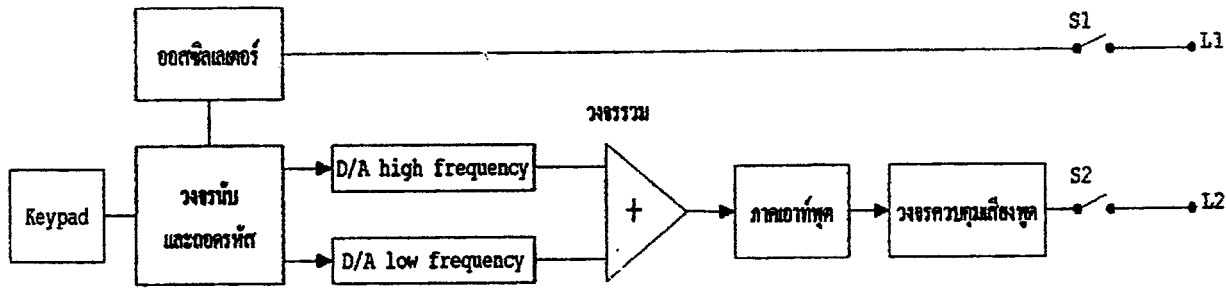
ส่วนประกอบหลักของเครื่องโทรศัพท์แบ่งออกได้เป็น 3 ส่วนดังนี้

1. ส่วนรับ-ส่งสัญญาณเสียงพูด (Speech Transmission)
2. ส่วนกำเนิดสัญญาณหมายเลขของผู้เรียก (Generator Tone Code)
3. ส่วนที่รับสัญญาณเรียกจากชุมสาย (Ringing Tone)

นอกจากนี้เครื่องโทรศัพท์ยังแบ่งออกเป็น 2 ชนิด ดังที่จะอธิบายต่อไป

### 2.9 ระบบโทรศัพท์แบบ DTMF

บล็อกไดอะแกรมของระบบ DTMF โดยใช้หลักการของวงจรดิจิทัลแปลงรหัสทางดิจิทัลเป็นสัญญาณอนาลอก ซึ่งมีความแม่นยำในการถอดรหัส จากการพัฒนานี้เองทำให้ปัจจุบันขนาดของโทรศัพท์เล็กลงมากเพราะไม่ต้องมีขดลวดกระดิ่งใหญ่ๆ เหมือนกับระบบพัลส์เนื่องจากการใช้ทรานซิสเตอร์เป็นตัวส่งเสียงแทน



รูปที่ 2.9 ระบบโทรศัพท์แบบ DTMF

โทรศัพท์ชนิดนี้สร้างสัญญาณ DTMF (Dual Tone Multiple Frequency) ในการส่งเลขหมายโดยการกดแต่ละหมายเลขบนหน้าปัดโทรศัพท์ จากการกดปุ่มแต่ละปุ่มจะมีสองความถี่ส่งออกไปพร้อมกัน ความถี่แต่ละคู่ที่ส่งออกไปจะมีค่าเวลาประมาณ 40 มิลลิวินาทีเป็นอย่างต่ำ โทรศัพท์แบบกดปุ่มจึงทำงานเร็วกว่าแบบหมุนอยู่ประมาณ 10 เท่า ในการออกแบบระบบควบคุมดังกล่าวจำเป็นต้องทราบการทำงานของระบบกล่าวคือ โทรศัพท์ชนิดนี้เป็นโทรศัพท์ชนิดกดปุ่มระบบ Tone ประกอบด้วยปุ่มกดจำนวน 12 ปุ่ม โดยการทำงานจะเป็นแบบ Dual Multi Frequency เป็นกคจะแบ่งออกเป็น 3 แถวกับ 4 คอลัมน์ ดังรูป

| ความถี่<br>Hz | รหัสหรือหมายเลข |      |      |               |
|---------------|-----------------|------|------|---------------|
| 697           | 1               | 2    | 3    |               |
| 770           | 4               | 5    | 6    |               |
| 852           | 7               | 8    | 9    |               |
| 941           | *               | 0    | #    |               |
|               | 1209            | 1336 | 1477 | ความถี่<br>Hz |

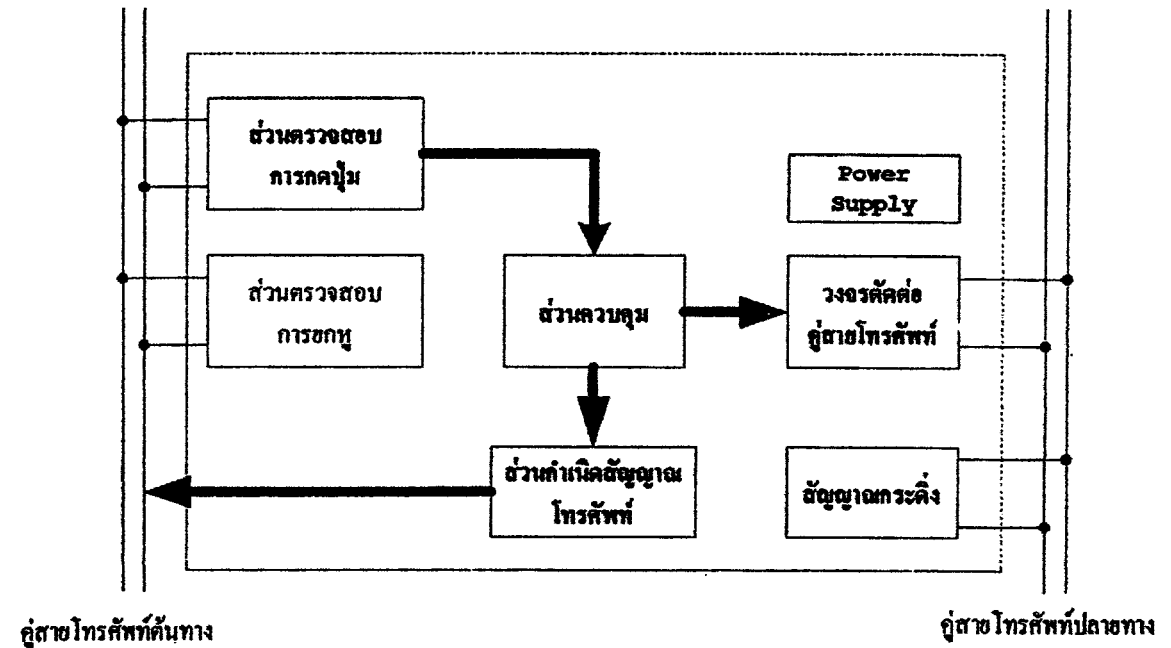
รูปที่ 2.10 ความถี่ของ โทนเสียงเมื่อกดหมายเลขต่างๆ

เมื่อคหหมายเลขใดหมายเลขหนึ่งจะประกอบด้วยโทนเสียง 2 ความถี่ด้วยกันคือ ความถี่สูงและความถี่ต่ำ (ความถี่ทางด้านคอลัมน์และแถวตามลำดับ)

| หมายเลข | ความถี่ต่ำ(Hz) | ความถี่สูง (Hz) |
|---------|----------------|-----------------|
| 1       | 697            | 1209            |
| 2       | 697            | 1366            |
| 3       | 697            | 1477            |
| 4       | 770            | 1209            |
| 5       | 770            | 1366            |
| 6       | 770            | 1477            |
| 7       | 852            | 1209            |
| 8       | 852            | 1336            |
| 9       | 852            | 1447            |
| 0       | 941            | 1336            |
| *       | 941            | 1209            |
| #       | 941            | 1477            |
| A       | 697            | 1633            |
| B       | 770            | 1633            |
| C       | 852            | 1633            |
| D       | 941            | 1633            |

ตารางที่ 2.1 แสดงความถี่ที่มีคู่เลขกันเมื่อคหหมายเลขต่างๆ

## 2.10 ชุมสายโทรศัพท์



รูปที่ 2.11 การทำงานพื้นฐานของชุมสายโทรศัพท์

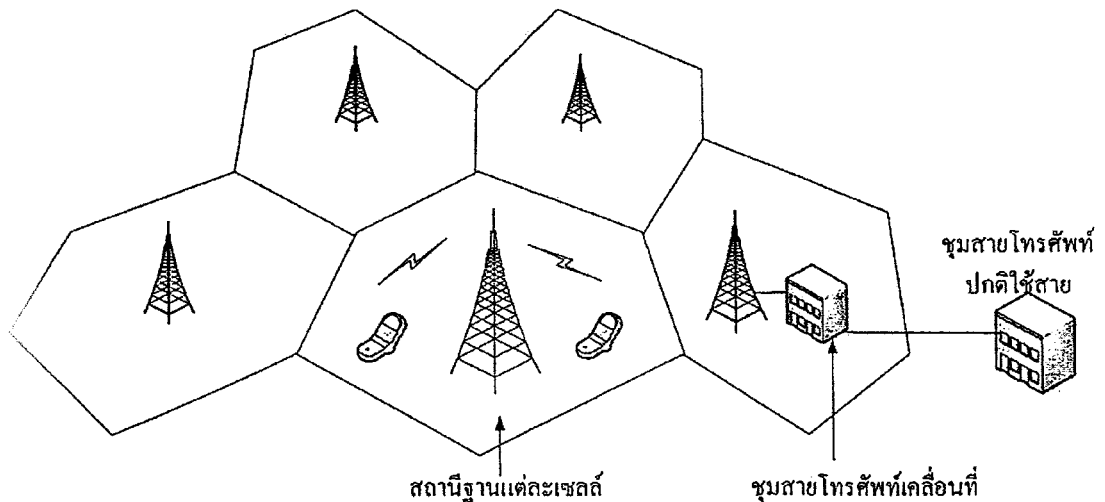
ชุมสายโทรศัพท์เป็นอุปกรณ์ที่มีความสำคัญมากในระบบโทรศัพท์ เพราะเป็นอุปกรณ์ที่ทำหน้าที่ตัดต่อคู่สายโทรศัพท์ให้ต้นทางและปลายทางสามารถติดต่อพูดคุยกันได้ จากระบบชุมสายใหญ่ๆ ก็มีการประยุกต์ใช้ในสำนักงาน เพื่อให้การติดต่อระหว่างแผนกต่างๆ ในสำนักงานได้อย่างคล่องตัว จากเดิมที่ใช้รีเลย์เป็นตัวตัดต่อคู่สาย ทำให้ชุมสายมีขนาดใหญ่โตมาก จึงได้มีการพัฒนามาเป็นอิเล็กทรอนิกส์สวิตซ์

## 2.11 พื้นฐานระบบเซลลูลาร์

ระบบโทรศัพท์มือถือ คือ ระบบโทรศัพท์แบบหนึ่ง ซึ่งมีลักษณะแตกต่างจากระบบโทรศัพท์ธรรมดา คือ การส่งสัญญาณเสียงระหว่างผู้รับกับผู้ส่งไม่จำเป็นต้องใช้สายโทรศัพท์ แต่การส่งสัญญาณจะอยู่ในรูปของคลื่นวิทยุความถี่สูง ซึ่งจะมีสถานีเครือข่ายทำหน้าที่รับและส่งสัญญาณวิทยุไปยังผู้รับและผู้ส่ง โดยสถานีเครือข่ายจะสามารถส่งสัญญาณครอบคลุมพื้นที่บริเวณหนึ่งเท่านั้น ซึ่งพื้นที่ที่สถานีเครือข่ายแต่ละสถานีครอบคลุม เรียกว่าเซลล์ (Cell) โดยในเซลล์หนึ่งจะมีรัศมีครอบคลุมประมาณ 2.5-10 กิโลเมตร ในกรณีที่ต้องการให้ครอบคลุมพื้นที่กว้างกว่านี้จะต้องตั้งสถานีรับส่งสัญญาณให้ครอบคลุมในเขตพื้นที่ที่ต้องการ โดยย่านความถี่ที่ใช้มีหลายย่านความถี่และมีช่องสัญญาณต่างกันขึ้นอยู่กับผู้ให้บริการ โดยปัจจุบันย่านความถี่ที่ใช้งานจะถูกนำกลับมาใช้ใหม่เมื่อมีการวางสายหรือเลิกใช้แล้วในเขตพื้นที่ต่าง ๆ กันทำให้ใช้งานย่านความถี่ที่จำกัดได้เต็มที่ซึ่งระบบนี้ เรียกว่า ระบบเซลลูลาร์ (Cellular Mobile Telephone) ปัจจุบันได้ใช้งานระบบนี้กันอย่างแพร่หลาย

โทรศัพท์เซลลูลาร์เป็นระบบโทรศัพท์เคลื่อนที่ระบบใหม่ ที่ถูกพัฒนาขึ้นมาให้เหมาะสมกับการใช้งานในระบบสื่อสารวิทยุโทรศัพท์เคลื่อนที่ในปัจจุบัน สามารถครอบคลุมพื้นที่ให้บริการได้กว้างมากขึ้น ขยายขอบเขตการให้บริการแก่ผู้ใช้บริการได้ต่อเนื่อง ไม่มีขีดจำกัด กล่าวคือในเขตพื้นที่ที่มีประชากรหนาแน่น เช่น กรุงเทพมหานคร มีความต้องการใช้โทรศัพท์เคลื่อนที่เป็นจำนวนมาก ก็ออกแบบให้มีจำนวนเซลล์มากขึ้นเพื่อรองรับอัตราใช้บริการหรือ ทราฟฟิก (Traffic) ที่เพิ่มขึ้นส่วนในเขตพื้นที่ที่มีประชากรเบาบาง เช่น ในต่างจังหวัดมีความต้องการใช้โทรศัพท์เคลื่อนที่เป็นจำนวนน้อยก็ออกแบบเซลล์ให้มีขนาดใหญ่ขึ้น แต่ละเซลล์ที่ติดกันจะใช้ย่านความถี่ที่แตกต่างกันเพื่อป้องกันการรบกวนซึ่งกันและกัน ส่วนเซลล์ที่อยู่ห่างออกไปจะนำความถี่เดิมมาใช้ซ้ำอีก เรียกว่า การนำความถี่กลับมาใช้อีก โดยไม่เกิดการสอดแทรกและสูญหายของสัญญาณเมื่อต้องการใช้โทรศัพท์เคลื่อนที่เพิ่มขึ้นก็แบ่งจำนวนเซลล์ออกเป็นเซลล์ย่อยๆ (Cell Splitting) ให้มากขึ้นได้ตามต้องการ ทำให้การทำงานของระบบโทรศัพท์เซลลูลาร์สามารถนำความถี่มาใช้งานอย่างคุ้มค่า

ระบบโทรศัพท์เซลลูลาร์ใช้หลักการติดตั้งเครื่องรับส่งวิทยุ (Transceiver) ที่กำลังส่งต่ำมากๆ จำนวนมากกระจายเป็นจุดๆ ทั่วพื้นที่ให้บริการ จุดที่ติดตั้งเครื่องรับส่งวิทยุเหล่านี้ เรียกว่า สถานีฐาน (Base Station) หรืออาจเรียกว่าที่ตั้งเซลล์ (Cell Site) ซึ่งทำหน้าที่เป็นศูนย์กลางเซลล์ ลักษณะโครงสร้างของระบบโทรศัพท์เซลลูลาร์แสดงดังรูปที่ 2.12



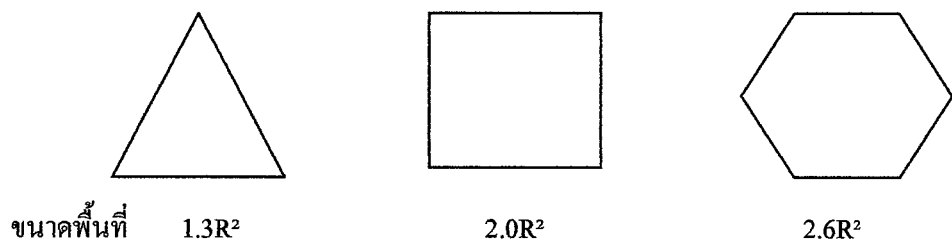
รูปที่ 2.12 แสดง โครงสร้างของเครือข่ายระบบ โทรศัพท์เซลลูลาร์

จากรูปที่ 2.12 แสดง โครงสร้างของเครือข่ายระบบ โทรศัพท์เซลลูลาร์แต่ละเซลล์จะเชื่อมต่อกันซึ่งแต่ละเซลล์มีรัศมีทำการกว้างหรือแคบขึ้นอยู่กับความหนาแน่นของการใช้โทรศัพท์เซลลูลาร์ ถ้าการใช้โทรศัพท์เซลลูลาร์มีความหนาแน่นมากจำนวนเซลล์จะเพิ่มขึ้นรัศมีของแต่ละเซลล์ก็จะเล็กลง และถ้าการใช้โทรศัพท์เซลลูลาร์มีความหนาแน่นน้อยจำนวนเซลล์ลดลงรัศมีของแต่ละเซลล์จะใหญ่ขึ้น

มาตรฐานทั่วไปที่นิยมใช้ขนาดของเซลล์มีรัศมีตั้งแต่ 250 เมตร ถึง 30 กิโลเมตร แต่ละเซลล์ที่ถูกแบ่งออกจะต้องเพิ่มสถานีฐานเข้าไปทุกเซลล์ เครื่องรับส่งแต่ละเซลล์สามารถทำงานได้ที่กำลังส่งครอบคลุมอยู่ในเฉพาะพื้นที่เล็กๆ ของเซลล์ตัวเองเท่านั้น ทำให้เครื่องรับส่งที่ใช้งานมีขนาดเล็กลง

## 2.12 หลักการจัดแบ่งเซลล์ในระบบโทรศัพท์เซลลูลาร์

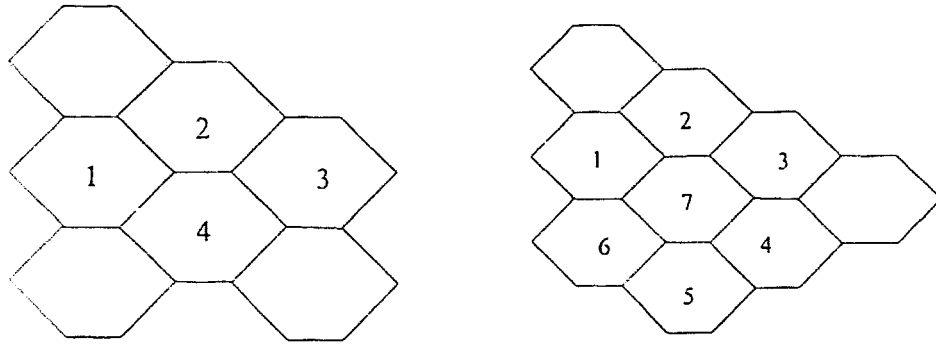
จุดมุ่งหมายหลักของการแบ่งพื้นที่ให้บริการออกเป็นเซลล์ย่อยก็เพื่อที่จะสามารถใช้ประโยชน์จากสเปกตรัมที่มีอยู่อย่างจำกัดให้ได้ประสิทธิภาพมากที่สุด ลักษณะการแบ่งพื้นที่นั้นสามารถทำได้หลายรูปแบบ เช่น การแบ่งเป็นพื้นที่รูปสามเหลี่ยม สี่เหลี่ยม และหกเหลี่ยม ดังรูปที่ 2.13 พิจารณาจากรูปจะเห็นว่า การแบ่งบริเวณพื้นที่ที่ออกเป็นรูปหกเหลี่ยมเมื่อกำหนด  $R$  ให้มีขนาดเท่ากัน ซึ่งหมายความว่า สำหรับขนาดพื้นที่ให้บริการที่เท่ากันเราจะใช้จำนวนอุปกรณ์รับส่งสัญญาณที่น้อยลงได้ นอกจากนี้รูปหกเหลี่ยมยังมีความคล้ายคลึงกับรูปร่างกลมมากที่สุด ซึ่งสอดคล้องใกล้เคียงกับลักษณะการแพร่กระจายของคลื่นสัญญาณมากขึ้น และที่สำคัญมากที่สุดคือรูปหกเหลี่ยมจะมีปัญหาของสัญญาณรบกวนจากเซลล์รอบข้างน้อยกว่ารูปแบบอื่น



รูปที่ 2.13 การแบ่งบริเวณพื้นที่ออกเป็นรูปสามเหลี่ยม สี่เหลี่ยม และหกเหลี่ยม

ระบบโทรศัพท์เซลลูลาร์นั้นต้องนำความถี่ที่ใช้แล้วกลับมาในเซลล์ต่างๆ ที่อยู่ห่างกันออกไป จำนวนช่องสัญญาณมากขึ้นขึ้นอยู่กับจำนวนแถบคลื่นความถี่ที่จัดสรรให้โดยองค์กรที่รับผิดชอบและช่วงห่าง ของช่องสัญญาณที่เป็นมาตรฐาน ใช้กันอยู่เป็นเครือข่ายช่องสัญญาณ ในกลุ่มเซลล์ที่ติดกันจะต้องใช้ความถี่ที่แตกต่างกัน ระบบการทำงานของโทรศัพท์เซลลูลาร์ต้องระมัดระวังการสอดแทรกของช่องสัญญาณที่ใช้ความถี่เดียวกันในบริเวณที่ใกล้เคียงกัน (Co - Channel Interference) ต้องให้อยู่ในขีดจำกัดที่ยอมรับได้ตามมาตรฐานการสื่อสารในระบบเซลลูลาร์

ในการนำความถี่เดิมมาใช้ใหม่จำเป็นต้องเลือกจำนวนช่องเซลล์สัญญาณที่แตกต่างกัน (K) ควรให้มีมากที่สุดที่จะจัดได้เท่าที่จำเป็น จะต้องไม่ให้เกิดการทับกันหรือเกิดช่องว่างระหว่างเซลล์จำนวนช่องเซลล์สัญญาณที่แตกต่างกัน (K) ที่ยอมรับได้และนิยมใช้งานคือ  $K = 4, 7, 12$  และ  $19$  เซลล์ เป็นต้น ลักษณะการจัดวางเซลล์สัญญาณที่แตกต่างกันแสดงดังรูปที่ 2.14



รูปที่ 2.14 การจัดเซลล์มีความถี่แตกต่างกันในแต่ละช่องสัญญาณ

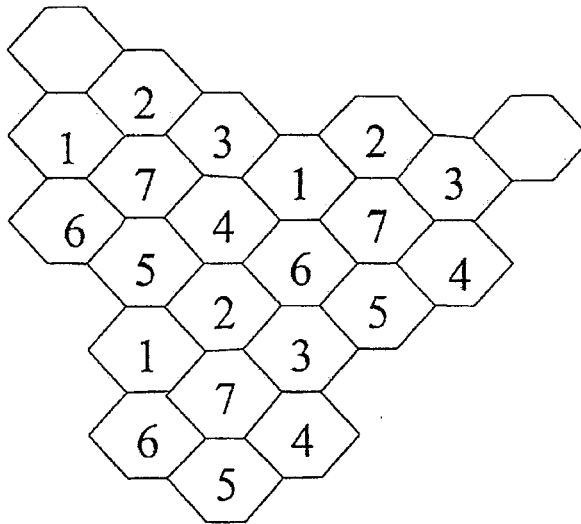
การใช้งานของโทรศัพท์เซลล์ดูราร์นิยมใช้ค่า  $K$  น้อยค่า เพราะค่า  $K$  น้อยค่านั้นทำให้จำนวนช่องสัญญาณสื่อสารของแต่ละเซลล์มีจำนวนมากขึ้น สามารถให้บริการผู้ใช้โทรศัพท์เซลล์ดูราร์ภายในเซลล์มากหมายเลขแต่ถ้าค่า  $K$  เพิ่มขึ้นทำให้จำนวนช่องสัญญาณสื่อสารของแต่ละเซลล์มีจำนวนน้อยลงให้บริการผู้ใช้โทรศัพท์เซลล์ดูราร์ภายในเซลล์น้อยหมายเลขลง

ระยะห่างของการใช้เซลล์มีความถี่ซ้ำ ( $D$ ) หาได้จากค่ารัศมีของเซลล์ที่ใช้ และค่าจำนวนช่องเซลล์สัญญาณที่แตกต่างกัน ( $K$ ) หาโดยเขียนเป็นสมการได้ ดังนี้

$$D = R \times \sqrt{3K}$$

- เมื่อ  $D$  = ระยะห่างของการใช้เซลล์มีความถี่ซ้ำ  
 $R$  = รัศมีของเซลล์  
 $K$  = จำนวนช่องสัญญาณที่แตกต่างกัน

ค่า  $D$  ของระบบจะมากขึ้นหรือน้อยลงขึ้นอยู่กับค่า  $K$  ที่ใช้งานอยู่ค่า  $K$  มีค่าน้อยระยะห่างของ  $D$  น้อยลงด้วย และค่า  $K$  มากขึ้นทำให้ค่า  $D$  มากขึ้นตามไปด้วย ลักษณะและตำแหน่งการจัดเซลล์ใช้ความถี่ซ้ำๆ

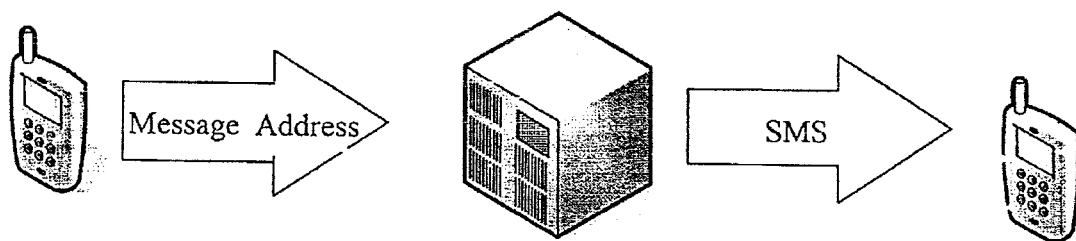


รูปที่ 2.15 ลักษณะและตำแหน่งการจัดเซลล์ใช้ความถี่ซ้ำๆ ของขนาด  $K = 7$

จากรูปที่ 2.15 แสดงลักษณะและตำแหน่งการจัดเซลล์ใช้ความถี่ซ้ำๆ ของขนาดเซลล์สัญญาณที่แตกต่างกัน 7 เซลล์ ( $K = 7$ ) จะเห็นได้ว่ากลุ่มเซลล์ที่มีความถี่แตกต่างกันถูกจับกลุ่มให้มีเซลล์ติดต่อกัน การใช้เซลล์ความถี่ซ้ำกลุ่มต่อไปจำเป็นต้องวางตำแหน่งเซลล์ความถี่ซ้ำให้มีระยะห่างจากเซลล์เดิมเป็นระยะที่เท่าๆ กันอย่างสม่ำเสมอ เพื่อเป็นการป้องกันการสอดแทรกของช่องสัญญาณที่ใช้ความถี่เดียวกัน การวางตำแหน่งกลุ่มเซลล์สามารถจัดได้แบ่งออกเป็นกลุ่มๆ

### 2.13 การใช้งานระบบ GSM

ระบบ GSM นั้นแตกต่างจากระบบอื่นที่ย่านความถี่ต่ำ 900 MHz และวิธีการแบ่งช่องสัญญาณแบบ TDMA และระบบ GSM ได้นำระบบรักษาความปลอดภัยแบบ Subscriber Identification Module (SIM) โดยระบบนี้เป็นการเก็บความจำเกี่ยวกับหมายเลข หมายเลขเครื่อง รหัสลับ และตัวอักษรอื่นที่ต้องการเก็บลงใน ซิมการ์ด (SIM Card) ข้อดีของระบบ SIM คือ การใช้งานเราสามารถพกพาซิมการ์ดของเราไปใช้งานเครื่องโทรศัพท์มือถือเครื่องใดก็ได้ที่อยู่ในระบบเดียวกัน และเมื่อเครื่องโทรศัพท์มือถือเสียหายหรือชำรุดก็สามารถเปลี่ยนซิมการ์ดมาไว้เครื่องใหม่ โดยข้อมูลเดิมยังอยู่ครบจากข้อดีของระบบ SIM นี้ทำให้ GSM มีการใช้งานที่แตกต่างจากระบบอื่น เช่น มีการใช้งานการส่งข้อความสั้น โดยสามารถส่งข้อความจำนวนตัวอักษรไม่เกิน 160 ตัวอักษรในแต่ละครั้งของการส่ง ซึ่งข้อดีของการใช้งานข้อความสั้น คือข้อความถึงผู้รับแน่นอน ถึงแม้ว่าผู้รับจะปิดเครื่องแต่ข้อความจะแสดงเมื่อเปิดเครื่อง เนื่องจากการทำงานของการส่งข้อความสั้น จะเป็นดังนี้ เริ่มต้นเมื่อเราเขียนข้อความและกดส่งจากโทรศัพท์มือถือข้อความ เบอร์โทรศัพท์มือถือที่เราต้องการส่งให้ เบอร์โทรศัพท์มือถือของผู้รับจะถูกหัดหลังจากนั้น ศูนย์บริการส่งข้อความสั้นจะคิดค่าใช้จ่ายและส่งข้อความไปยังเบอร์โทรศัพท์มือถือที่เราต้องการส่ง โดยแสดงรูปแบบได้ตามรูปที่ 2.16



รูปที่ 2.16 แสดงลักษณะขั้นตอนของการส่งข้อความสั้น

จากความสามารถของระบบ GSM ที่สามารถส่งข้อความสั้นไปยังผู้รับได้นี้เอง เราจึงเลือกที่จะใช้ระบบ GSM เป็นเครือข่ายสำหรับส่งข้อความสั้นเชิงเตือนภัย

## 2.14 ลักษณะของการส่งข้อความสั้น

การส่งข้อความสั้น (Short Message Service) คือการส่งข้อความสั้นๆ หรือข้อมูลสั้นจากเครื่องโทรศัพท์มือถือผู้ส่ง ไปยังเครื่องโทรศัพท์มือถือของผู้รับโดยส่งผ่านเครือข่ายศูนย์บริการ (SMSC) โดยการส่งแบบข้อความสั้นนี้เราสามารถเลือกได้ว่าจะส่งข้อความสั้น หรือ รูปภาพโลโก้ เสียงเพลงริงค์โทน ซึ่งจะมีวิธีการส่งที่แตกต่างกัน 2 แบบ คือโหมดตัวอักษร (text-Mode) และโหมดพีดียู (Protocol Data Unit : PDU) โดยโหมดตัวอักษร คือ โหมดที่เราสามารถส่งข้อความสั้นๆ ประมาณ 160 ตัวอักษร ไปยังเครื่องโทรศัพท์มือถือของผู้รับ โดยลักษณะข้อความนั้นจะอยู่ในรูปแบบรหัสแอสกี (ASCII) ส่วนในโหมดพีดียู คือ โหมดที่สามารถส่งได้ทั้งข้อความสั้นๆ ส่งรูปภาพและเพลงริงค์โทนได้ ซึ่งโหมดพีดียูนี้จะมีรูปแบบการวางข้อมูลที่จะส่งแตกต่างกับโหมดตัวอักษร คือ โหมดพีดียูจะมีการเข้ารหัสที่จะแปลงข้อความในรูปแบบของเลขฐานสิบหกและต้องมีการส่งหัวข้อของชุดข้อมูล (Heading) แต่ในโหมดตัวอักษรจะเป็นการส่งแบบรหัสแอสกีและไม่จำเป็นต้องส่งหัวข้อของชุดข้อมูล

### 2.14.1 การรับ-ส่งข้อความสั้นแบบโหมดพีดียู (PDU-Mode)

ในโครงการนี้เราจะใช้การส่งข้อความสั้นแบบโหมดพีดียู ซึ่งรูปแบบการจัดรูปแบบนั้นจะซับซ้อนกว่าแบบโหมดตัวอักษรมาก แต่การส่งแบบโหมดพีดียูนั้นเราสามารถใช้ได้กับโทรศัพท์มือถือได้ทุกรุ่น โดยการส่งข้อความสั้นแบบโหมดพีดียูมีรายละเอียดดังนี้ คือในนี้จะต้องมีการสร้างหัวข้อของชุดข้อมูลสำหรับส่ง ซึ่งประกอบด้วยส่วนของศูนย์บริการข้อความสั้นกับส่วนของชุดข้อความ (Transfer Protocol Unit : TPDU) โดยทั้งสองส่วนจะมีลักษณะเป็นเลขฐานสิบหกซึ่งจะวางลำดับตามนี้

|                                      |                                   |                             |                                |
|--------------------------------------|-----------------------------------|-----------------------------|--------------------------------|
| หัวข้อของชุดข้อมูล<br>(Heading : Cr) | ส่วนของศูนย์บริการ<br>ข้อความสั้น | ส่วนของชุดข้อความ<br>(TPDU) | บิตหยุด<br>(Stop bit : Ctrl-Z) |
|--------------------------------------|-----------------------------------|-----------------------------|--------------------------------|

ตารางที่ 2.2 แสดงส่วนประกอบของชุดข้อมูลในการส่งข้อความสั้นแบบโหมคพีดียู

ในส่วนของชุดข้อความก็จะประกอบด้วยส่วนย่อยๆ ซึ่งจะเป็นตัวกำหนดรูปแบบของการส่งข้อความสั้นที่ข้อความสั้นที่จะส่ง โดยถ้าเราต้องการที่จะส่งเป็นข้อความจะต้องจัดรูปแบบเรียงตามนี้

1. โปรโตคอลพารามิเตอร์ คือ พารามิเตอร์ที่บอกว่าโปรโตคอล (Protocol) ที่ใช้ส่งเป็นแบบใด กรณีส่งแบบ TPDU = 0x01
2. ตัวเลขอ้างอิงข้อความ ในกรณีที่มีข้อความหลายๆ ข้อความ เราสามารถจัดลำดับข้อความโดยใช้ตัวเลขอ้างอิงข้อความได้ (มีค่าปกติ = 0x00)
3. ความยาวของเบอร์โทรศัพท์มือถือของหมายเลขปลายทาง
4. รูปแบบของเบอร์โทรศัพท์มือถือของหมายเลขปลายทาง ซึ่งจะเป็นตัวบอกลักษณะของเบอร์โทรศัพท์มือถือที่เราต้องการส่งข้อความไปให้ โดยส่งแบบสากลจะใช้ค่า = 0x91
5. หมายเลขโทรศัพท์มือถือของหมายเลขปลายทางที่ต้องการจะส่ง โดยหมายเลขโทรศัพท์นี้ จะมีการเข้ารหัสแบบสลับ (Nibble Swapped)
6. ตัวแสดงรูปแบบชุดข้อมูล
7. ลักษณะการเข้ารหัสของข้อมูล คือ พารามิเตอร์ที่บอกว่าเราจะส่งเป็นภาษาใด (มาตรฐาน คือ ระบบ GSM)
8. ความยาวของข้อความที่ต้องการส่ง (ก่อนเข้ารหัส)
9. ข้อความที่ต้องการส่ง (หลังเข้ารหัส)

จะเห็นว่าการส่งแบบโหมคพีดียูมีการเข้ารหัสที่ซับซ้อน เช่น การเข้ารหัสสลับ และการเข้ารหัสของชุดข้อความที่จะส่งโดยการเข้ารหัสแบบสลับมีลักษณะดังนี้ โดยจะนำการสลับเบอร์โทรศัพท์ที่ติดกันเป็นคู่ๆ และถ้าเหลือเศษจะเติมค่า F เข้าไปก่อนรหัสตัวสุดท้าย เช่น เบอร์โทรศัพท์ คือ 123456789 เมื่อเข้ารหัสสลับแล้วจะกลายเป็น 21436587F9 ส่วนการเข้ารหัสของชุดข้อความจะต้องทำการแปลงข้อความที่เป็นแอสกี มาเป็นเลขฐานสองหลักจากนั้นก็ทำการเข้ารหัส

ส่วนของศูนย์บริการข้อความสั้นจะเป็นที่กำหนดเครือข่ายการใช้บริการว่าจะใช้บริการผ่านศูนย์บริการ ส่วนของศูนย์บริการข้อความสั้นใดๆ โดยจะประกอบด้วยส่วนย่อยๆ ดังนี้ คือ

1. ความยาวของเบอร์ศูนย์บริการ
2. รูปแบบของเบอร์ศูนย์บริการ (ส่งแบบสากลจะใช้ค่า = 0x91)
3. เบอร์ศูนย์บริการ โดยจะมีการเข้ารหัสแบบสลับ

เมื่อผู้รับได้รับข้อความสั้นที่มีการส่งแบบโหมคพีดียูรูปแบบของข้อความก็จะอยู่ในลักษณะของโหมคพีดียูเราจำเป็นต้องศึกษาถึงรูปแบบของข้อความที่ได้รับดังนี้ คือ ข้อความที่ได้รับนี้จะประกอบด้วยส่วนสำคัญสองส่วน คือ ส่วนของศูนย์บริการข้อความสั้นกับส่วนของชุดข้อความ โดยทั้งสองส่วนจะมีลักษณะเป็นเลขฐานสิบหกซึ่งจะเหมือนกับการส่ง แต่ชุดข้อมูลบางชุดเพิ่มเติมเข้ามา คือ เวลา วันเดือนปีที่ได้รับข้อความ และเบอร์โทรศัพท์ของผู้ส่ง

ในส่วนของชุดข้อความก็จะประกอบด้วยส่วนย่อยๆ ซึ่งจะเป็นตัวกำหนดรูปแบบของข้อความที่รับมาโดยในส่วนนี้จะมีส่วนที่แตกต่างจากการส่ง คือ เพิ่มเวลา วันเดือนปีที่ได้รับข้อความและเปลี่ยนจากเบอร์ที่ต้องการส่งเป็นเบอร์ที่ส่งมาจากต้นทาง โดยจัดรูปแบบเรียงตามนี้

1. โปรโตคอลพารามิเตอร์ คือ พารามิเตอร์ที่บอกว่าโปรโตคอล (Protocol) ที่ใช้ส่งเป็นแบบใด กรณีส่งแบบ TPDU = 0x01
2. ตัวเลขอ้างอิงข้อความ ในกรณีที่มีข้อความหลายๆ ข้อความเราสามารถจัดลำดับข้อความโดยใช้ตัวเลขอ้างอิงข้อความได้ (มีค่าปกติ = 0x00)
3. ความยาวของเบอร์โทรศัพท์มือถือของหมายเลขต้นทาง
4. รูปแบบของเบอร์โทรศัพท์มือถือของหมายเลขต้นทาง ซึ่งจะเป็นตัวบอกลักษณะของเบอร์โทรศัพท์มือถือที่เราต้องการส่งข้อความไปให้ โดยส่งแบบสากลจะใช้ค่า = 0x91
5. หมายเลขโทรศัพท์มือถือของหมายเลขต้นทางที่ต้องการจะส่ง โดยหมายเลขโทรศัพท์นี้จะมีการเข้ารหัสแบบสลับ
6. ตัวแสดงรูปแบบชุดข้อมูล
7. ลักษณะการเข้ารหัสของข้อมูล คือ พารามิเตอร์ที่บอกว่าเราจะส่งเป็นภาษาใด (มาตรฐานคือระบบ GSM)
8. เวลาและวันเดือนปีที่ได้รับข้อความ เช่น 0x99 0x20 0x21 0x50 0x75 0x03 0x21 หมายถึง 12. Feb 1999 05:57:30 GMT + 3
9. ความยาวของข้อความที่ต้องการส่ง (ก่อนเข้ารหัส)
10. ข้อความที่ต้องการส่ง (หลังเข้ารหัส)

โดยปกติการส่งข้อความสั้นนี้เราสามารถกดส่งจากเครื่องโทรศัพท์มือถือของเราได้โดยเริ่มจากการเขียนข้อความ เมื่อข้อความเสร็จแล้วจะมีให้เลือกที่จะส่งไปเบอร์โทรศัพท์หมายเลขใด นอกจากนี้เราต้องกดส่งจากโทรศัพท์มือถือแล้วเรายังสามารถเลือกที่จะส่งข้อความสั้นได้อีกแบบ คือ ในเครื่องโทรศัพท์บางรุ่นที่มีอยู่ในปัจจุบันจะมีพอร์ตอนุกรมนี้เป็นตัวเชื่อมต่อระหว่างคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ได้ พอร์ตอนุกรมที่มีนี้ทำให้เราง่ายในการส่งข้อความสั้นอย่างมาก คือ เราไม่จำเป็นต้องกดปุ่มที่เครื่องโทรศัพท์มือถือเพียงแค่ว่าส่งชุดคำสั่งเป็นรหัสแอสกีเข้าไปทางพอร์ตอนุกรมนี้เราก็สามารถสั่งงานให้เครื่องโทรศัพท์มือถือส่งข้อความสั้น

### 2.14.2 ตัวอย่างการส่งข้อความสั้นแบบโทมคพีดียู

โดยจะทำการส่งข้อความสั้นคำว่า “hellohello” โดยใช้โทมคพีดียูไปยังหมายเลข “+66 092056208”

AT+CMGF=0 // เพื่อเลือกโทมคพีดียู

AT+CMMS=0 // เช็คว่ามีมือถือสนับสนุนการส่งข้อความสั้น หรือไม่

AT+CMGS=22 // ต้องการส่งทั้งหมด 22 bytes (ไม่รวมตัวเลข 00 ที่อยู่ข้างหน้าสุด)

> 0011000A9166295026800000AA0AE8329BFD4697D9EC37 // เมื่อพิมพ์ข้อความครบแล้ว

กด Ctrl + z ส่วนประกอบของข้อมูลที่ส่งอธิบายในตารางที่ 2.3

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด  |
|---------------------------|---|
| 00                        | ความยาวของ SMSC Information 00 หมายถึง ให้ใช้ SMSC Information ที่เก็บอยู่ภายในเครื่อง(ปกติเครื่องที่สามารถส่ง SMS ได้จะมีข้อมูล SMSC ภายในเครื่องอยู่แล้ว) |
| 11                        | First octet of the SMS-SUBMIT message   |
| 00                        | TP-Message-Reference “00” คือ ให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง  |
| 0A                        | Address-Length ความยาวของเลขหมายผู้รับ (0A hex = 10)  |
| 91                        | Type-of-Address (91 indicates international format of the phone number)   |
| 66 29 50 26 80            | เลขหมายผู้รับ (แบบ Decimal Semi-Octets)เป็นเลขฐาน 10 สลับ Nibble หมายเลขที่แท้จริง คือ +66092056208   |
| 00                        | TP-PID (Protocol Coding identifier) เป็น 00   |
| 00                        | TP-DCS (Data Coding Scheme) เป็น 00   |
| AA                        | TP-Validity-Period “AA” หมายถึงช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทางข้อความจะถูกยกเลิกโดยอัตโนมัติ                         |
| 0A                        | TP-User-Data-Lengthจำนวนตัวอักษรของข้อความที่ส่ง (10 ตัว)   |
| E8329BFD4697D9EC37        | TP-UD ข้อความ “hellohello” ที่เข้ารหัสแล้วจากตัวอักษรแบบ 7 บิตเป็นข้อมูลไบนารี ขนาด 8 บิต   |

ตารางที่ 2.3 ส่วนประกอบของข้อมูลที่ส่ง

### 2.14.3 ตัวอย่างการรับข้อความสั้น แบบโหมคทีดียู

ถ้าหากเราเชื่อมต่อกับมือถือแล้วทำการอ่านข้อความสั้นที่อยู่ในภาคเข้า โดยใช้คำสั่ง AT+CMGR ข้อมูลที่ได้รับจะอยู่ในรูปของสตริงที่ประกอบไปด้วยข้อมูลของผู้ส่ง ข้อมูล SMS Service Center (SMSC) Time Stamp และอื่นๆ ที่จำเป็นและตามด้วยส่วนของข้อความซึ่งจะอยู่ที่ท้ายสุดของสตริง ตัวอย่างสตริงต่อไปนี้ข้อความส่งมาคือ “hellhello” จากมือถืออีกเครื่องหนึ่งของข้อมูลสตริงนี้จะอยู่ในรูปของตัวเลขฐาน 16 และฐาน 10 (ในบางส่วน) โดยจะเรียกตัวเลขแต่ละคู่ว่า Octet ซึ่งมีรายละเอียดดังตารางที่ 2.4

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด  |
|---------------------------|---|
| 06                        | ความยาวของ SMSC Information 06 Octet (ไบนารี)   |
| 91                        | รูปแบบของเลขหมาย SMSC 91 หมายถึง เลขหมายแบบสากล (International Format)  |
| 66 81 11 80 88            | เลขหมาย SMSC (แบบ decimal semi-octets) ซึ่งจะเป็นเลขฐาน 10 สลับ Nibble ในกรณีนี้เลขหมายจริงของ Service Center คือ +6618110888 |
| 04                        | First octet of the SMS-DELIVER message  |
| 0A                        | ความยาวของเลขหมายผู้ส่ง (0A hex = 10)   |
| 91                        | รูปแบบของเลขหมายผู้ส่ง 91 หมายถึง เลขหมายแบบสากล (International Format)   |
| 66 29 50 26 80            | เลขหมายผู้ส่ง (แบบ Decimal Semi-Octets) เป็นเลขฐาน 10 สลับ Nibble หมายเลขผู้ส่งที่แท้จริง คือ +66092056208                    |
| 00                        | TP-PID (Protocol Coding identifier) เป็น 00   |
| 00                        | TP-DCS (Data Coding Scheme) เป็น 00 คือเข้ารหัสข้อความแบบ 7 บิต   |
| 40 30 21 21 94 34 82      | TP-SCTS ข้อมูล Time Stamp (แบบ Decimal Semi-Octets)   |
| 0A                        | User data length จำนวนตัวอักษรของข้อความที่ส่งในกรณีนี้คือ 10 ตัว   |
| E8329BFD4697D9EC37        | TP-UD ข้อความ “hellhello” ที่เข้ารหัสแล้วจากตัวอักษรแบบ 7 บิตเป็นข้อมูลไบนารีขนาด 8 บิต                                       |

ตารางที่ 2.4 ส่วนประกอบของสตริงการรับข้อความสั้น

ข้อมูลทั้งหมดในตารางเป็นเลขฐาน 16 ขนาด 8 บิต ยกเว้นหมายเลขของศูนย์บริการข้อความสั้น เลขหมายผู้ส่ง Timestamp จะเป็นเลขฐาน 10 ขนาด 8 บิตสลับหลักเป็นคู่ๆ ในส่วนของข้อมูลที่เป็นข้อความนั้นเป็นเลขฐาน 16 ขนาด 8 บิต เช่นกัน โดยข้อมูลนี้จะใช้แสดงข้อความที่ประกอบไปด้วย ตัวอักษรขนาด 7 บิต ซึ่งผ่านการแปลง (เข้ารหัส) ข้อมูลจากตัวอักษรขนาด 7 บิต ให้เป็นเลขฐาน 16 ขนาด 8 บิต มาแล้ว

ในส่วนของข้อมูลที่เป็นเลขฐาน 10 เช่น เลขหมายผู้ส่งตัวเลขในแต่ละคู่ (1 ไบต์) จะถูกสลับหลักกัน เช่น เลขหมายจริง "+66092056208" จะถูกสลับในแต่ละคู่เป็น "66 29 50 26 80" (66 คือ รหัสประเทศ ส่วนเลขหมวดของหมายเลขมือถือจะถูกตัดเลข 0 ออก เช่น 09 จะเหลือแค่ 9 เป็นต้น แล้วจึงนำตัวเลขทั้งหมดมาต่อกันแล้วสลับคู่) เช่นเดียวกันกับ Time Stamp ข้อมูล "40 30 21 21 94 34 82" ซึ่งจะมีรูปแบบเป็น "YY/MM/DD HH:MM:SS: ss" หมายถึง ข้อความนี้ส่งเมื่อ "04/03/12 12:49:43:28"

การแปลงตัวอักษรชนิด 7 บิตข้อมูล 8 บิต (Octet) โดยจากตารางที่ 2.4 ในส่วนของชุดข้อความจะเป็นส่วนที่เราสามารถใส่รหัสข้อความที่ต้องการส่ง แต่เนื่องจากเราไม่สามารถนำรหัสของตัวอักษรแบบ 7 บิตใส่ไปได้โดยตรงจำเป็นต้องผ่านการแปลงให้เป็นรหัสข้อมูลแบบ 8 บิตก่อนโดยตัวอย่างต่อไปนี้เป็นการแปลงข้อความ "hellhello" ยาว 10 ตัวอักษรซึ่งแต่ละตัวจะเป็นอักษรชนิด 7 บิต ให้เป็นข้อมูล 8 บิต สำหรับใช้ในการส่ง SMS การแปลงเริ่มจากการนำรหัส 7 บิตของตัวอักษรตัวแรก (h) มาเติมข้างหน้าด้วย 1 บิตท้ายสุดของรหัส 7 บิต ของอักษรตัวที่ 2 (e) จะได้ผลลัพธ์ 8 บิต (1 ไบต์) เป็น "E8" ขั้นตอนต่อมาให้เอา 6 บิตที่เหลือของอักษรตัวที่ 2 มาเติมข้างหน้าด้วย 2 บิตท้ายของรหัส 7 บิตของตัวอักษรตัวที่ 3 (l) จะได้ผลลัพธ์ 8 บิต เป็น "32" และทำเช่นนี้เรื่อยไปโดยจำนวนบิตที่นำมากระทำจะเพิ่มขึ้นเป็น 3 บิต 4 บิต จนกระทั่งถึง 7 บิต แล้วเริ่มกระบวนการใหม่จนกระทั่งหมดชุดตัวอักษรหลังจากการแปลงข้อความ "hellhello" จะได้ข้อมูลเป็นเลขฐาน 16 จำนวน 9 ไบต์ เป็น E8 32 9B FD 46 97 D9 EC 37 โดยมีวิธีการแปลงแสดงดังตารางที่ 2.5 โดยที่ตัวอักษรชนิด 7 บิตถูกกำหนดโดยมาตรฐาน GSM 03.38 ดังตารางที่ 2.6

|                  |                 |                 |                 |                 |                 |                |                |         |                 |
|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|---------|-----------------|
| h                | e               | l               | l               | o               | h               | e              | l              | l       | o               |
| 104              | 101             | 108             | 108             | 111             | 104             | 101            | 108            | 108     | 111             |
| 1101000          | 1100101         | 1101100         | 1101100         | 1101111         | 1101000         | 1100101        | 1101100        | 1101100 | 1101111         |
| 1101000          | 110010 <u>1</u> | 11011 <u>00</u> | 1101 <u>100</u> | 110 <u>1111</u> | 11 <u>01000</u> | <u>1100101</u> | <u>1101100</u> | 1101100 | 110111 <u>1</u> |
| <u>1</u> 1101000 | 00110010        | 10011011        | 11111101        | 01000110        | 10010111        | 11011001       | 11101100       | 110111  |                 |
| E8               | 32              | 9B              | FD              | 46              | 97              | D9             | EC             | 37      |                 |

ตารางที่ 2.5 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิตข้อความ "hellohello"

| Dec |     | 0  | 16    | 32 | 48 | 64 | 80 | 96 | 112 |
|-----|-----|----|-------|----|----|----|----|----|-----|
|     | Hex | 0  | 10    | 20 | 30 | 40 | 50 | 60 | 70  |
| 0   | 0   | @  | Δ     | SP | 0  | i  | P  |    | p   |
| 1   | 1   | £  | _     | !  | 1  | A  | Q  | a  | q   |
| 2   | 2   | \$ | €     | "  | 2  | B  | R  | b  | r   |
| 3   | 3   | ¥  | [     | #  | 3  | C  | S  | c  | s   |
| 4   | 4   | è  | ]     | ¤  | 4  | D  | T  | d  | t   |
| 5   | 5   | é  | Ω     | %  | 5  | E  | U  | e  | u   |
| 6   | 6   | ù  | Π     | &  | 6  | F  | V  | f  | v   |
| 7   | 7   | ì  | ¤     | '  | 7  | G  | W  | g  | w   |
| 8   | 8   | ò  | ~     | v  | 8  | H  | X  | h  | x   |
| 9   | 9   | Ç  | ß     | )  | 9  | I  | Y  | i  | y   |
| 10  | A   | LF | Þ     | *  | :  | J  | Z  | j  | z   |
| 11  | B   | Ø  | <ESC> | +  | ;  | K  | Ä  | k  | ä   |
| 12  | C   | ø  | ¢     | ,  | <  | L  | Ö  | l  | ö   |
| 13  | D   | CR | Æ     | -  | =  | M  | Ñ  | m  | ñ   |
| 14  | E   | Å  | £     | .  | >  | N  | Ü  | n  | ü   |
| 15  | F   | å  | É     | /  | ?  | O  | §  | o  | à   |

ตารางที่ 2.6 ชุดของตัวแปรของมาตรฐาน GSM 03.38

โดยที่ถ้าต้องการส่งข้อความเป็นภาษาไทยก็สามารถทำได้ แต่จากตารางที่ 2.6 จะสังเกตเห็นว่าเป็นตารางสากลจึงไม่มีอักษรภาษาไทยอยู่ ถ้าต้องการส่งข้อความภาษาไทยจึงต้องเปลี่ยนวิธีการเข้ารหัสข้อความเป็นแบบ UCS2 (16 บิต) ซึ่งจะสามารถส่งข้อความภาษาไทยได้โดยมีรหัสของตัวอักษรต่างๆ แสดงดังตารางที่ 2.7

|   | 0E0 | 0E1 | 0E2 | 0E3 | 0E4 | 0E5 | 0E6 | 0E7 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | -   | ฐ   | ภ   | ะ   | เ   | ๐   | -   | -   |
| 1 | ก   | ฑ   | ม   | ะ   | แ   | ๑   | -   | -   |
| 2 | ข   | ฒ   | ย   | า   | โ   | ๒   | -   | -   |
| 3 | ฃ   | ณ   | ร   | ำ   | ใ   | ๓   | -   | -   |
| 4 | ค   | ค   | ฤ   | ำ   | ใ   | ๔   | -   | -   |
| 5 | ค   | ค   | ล   | ำ   | ำ   | ๕   | -   | -   |
| 6 | ฅ   | ถ   | ภ   | ำ   | ำ   | ๖   | -   | -   |
| 7 | ง   | ท   | ว   | ำ   | ำ   | ๗   | -   | -   |
| 8 | จ   | ธ   | ศ   | ง   | '   | ๘   | -   | -   |
| 9 | ฉ   | น   | ษ   | ง   | ๗   | ๙   | -   | -   |
| A | ช   | บ   | ส   | .   | ๗   | ๗   | -   | -   |
| B | ช   | ป   | ห   | -   | +   | ๗   | -   | -   |
| C | ฅ   | ผ   | พ   | -   | ๗   | -   | -   | -   |
| D | ฅ   | ฝ   | อ   | -   | ๗   | -   | -   | -   |
| E | ฅ   | พ   | ฮ   | -   | ๗   | -   | -   | -   |
| F | ฅ   | ฟ   | ๗   | ฿   | ๗   | -   | -   | -   |

ตารางที่ 2.7 แสดงตารางรหัสตัวอักษรของการเข้ารหัสแบบ UCS2 (16 บิต)

#### 2.14.4 ตัวอย่างการรับข้อความสั้นแบบภาษาไทย

ทำการอ่านข้อความสั้น ที่อยู่ในภาคเข้า โดยใช้คำสั่ง AT+CNRR ข้อมูลที่ได้รับจะอยู่ในรูปของสตริงที่ประกอบไปด้วยข้อมูลของผู้ส่ง ข้อมูลส่วนของศูนย์บริการข้อความสั้น Time Stamp และอื่นๆ ที่จำเป็นและตามด้วยส่วนของข้อความซึ่งจะอยู่ที่ท้ายสุดของสตริง ตัวอย่างสตริงต่อไปนี้คือข้อความที่ส่งมา คือ “สวัสดี” จากมือถืออีกเครื่องหนึ่ง ข้อมูลสตริงนี้จะอยู่ในรูปของตัวเลขฐาน 16 และฐาน 10 (ในบางส่วน) โดยจะเรียกตัวเลขแต่ละคู่ว่า Octet ซึ่งจะมีรายละเอียดดังตารางที่ 2.8 ซึ่งมีตัวเลขต่างๆ ดังนี้

0691 6619001902040A916676804975000860200222292230C0E2A0E270E3100E2A0E140E35

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด   |
|---------------------------|--|
| 06                        | ความยาวของ SMSC Information 06 Octet (ไบต์)  |
| 91                        | รูปแบบของเลขหมายผู้ส่ง 91 หมายถึง เลขหมายแบบสากล (International Format)  |
| 66 19 00 19 02            | เลขหมาย SMSC (แบบ decimal semi-octets) ซึ่งจะเป็นเลขฐาน 10 สลับ Nibble ในกรณีนี้เลขหมายจริงของ Service Center คือ +66191009120 |
| 04                        | First octet of the SMS-DELIVER message   |
| 0A                        | ความยาวของเลขหมายผู้ส่ง (0A hex = 10)  |
| 91                        | รูปแบบของเลขหมายผู้ส่ง 91 หมายถึง เลขหมายแบบสากล (International Format)  |
| 66 76 80 49 75            | เลขหมายผู้ส่ง (แบบ Decimal Semi-Octets) เป็นเลขฐาน 10 สลับ Nibble หมายเลขผู้ส่งที่แท้จริงคือ +6667089457                       |
| 00                        | TP-PID (Protocol Coding identifier) เป็น 00  |
| 08                        | TP-DCS (Data Coding Scheme) เป็น 08 คือเข้ารหัสแบบ UCS2 (16 บิต)   |
| 60 20 02 22 22 92 23      | TP-SCTS ข้อมูล Time Stamp (แบบ Decimal Semi-Octets) สลับ Nibble  |
| 0C                        | TP-User-Data-Length จำนวนตัวอักษรของข้อความที่ส่ง  |
| 0E2A0E270E3100E2A0E140E35 | TP-UD ข้อความ “สวัสดี” ที่เข้ารหัสแล้วโดยดูเปรียบเทียบกับได้จากตารางที่ 2.7  |

ตารางที่ 2.8 ส่วนประกอบของข้อมูลที่รับแบบ UCS2 (16 บิต)

ส่วนตัวอย่างการส่งข้อความสั้นแบบภาษาไทย แสดงดังตารางที่ 2.9

โดยจะทำการส่งข้อความสั้นว่า “สวัสดี” โดยใช้โหมดพีดียูไปยังหมายเลข “+66 092056208”

AT+CMGF=0 // เพื่อเลือกโหมดพีดียู

AT+CMMS=0 // เช็คว่ามีมือถือสนับสนุนการส่งข้อความสั้น หรือไม่

AT+CMGS=25 // ต้องการส่งทั้งหมด 22 bytes (ไม่รวมตัวเลข 00 ที่อยู่ข้างหน้าสุด)

> 0011000A9166295026800008AA0C0E2A0E270E310E2A0E140E35 // เมื่อพิมพ์ข้อความ

ครบแล้วกด Ctrl+z ส่วนประกอบของข้อมูลที่ส่งอธิบายในตารางที่ 2.9

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด  |
|---------------------------|---|
| 00                        | ความยาวของ SMSC Information 00 หมายถึง ให้ใช้ SMSC Information ที่เก็บอยู่ภายในเครื่อง(ปกติเครื่องที่สามารถส่ง SMS ได้จะมีข้อมูล SMSC ภายในเครื่องอยู่แล้ว) |
| 11                        | First octet of the SMS-SUBMIT message   |
| 00                        | TP-Message-Reference “00” คือให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง   |
| 0A                        | Address-Length ความยาวของเลขหมายผู้รับ (0A hex = 10)  |
| 91                        | Type-of-Address (91 indicates international format of the phone number)   |
| 66 29 50 26 80            | เลขหมายผู้รับ (แบบ Decimal Semi-Octets)เป็นเลขฐาน 10 สลับ Nibble หมายเลขที่แท้จริงคือ +66092056208  |
| 00                        | TP-PID (Protocol Coding identifier) เป็น 00   |
| 08                        | TP-DCS (Data Coding Scheme) เป็น 08 คือเข้ารหัสแบบ UCS2 (16 บิต)  |
| AA                        | TP-Validity-Period “AA” หมายถึงช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทางข้อความจะถูกยกเลิกโดยอัตโนมัติ                         |
| 0C                        | TP-User-Data-Length จำนวนตัวอักษรของข้อความที่ส่ง   |
| 0E2A0E270E310E2A0E140E53  | TP-UD ข้อความ “สวัสดี” ที่เข้ารหัสแล้วจากตัวอักษรแบบ 7 บิตเป็นข้อมูลไบต์ ขนาด 8 บิต   |

ตารางที่ 2.9 ส่วนประกอบของข้อมูลที่ส่งแบบ UCS2 (16 บิต)

ตัวอย่างการแปลงตัวอักษรข้อความภาษาไทยโดยวิธีการเข้ารหัสแบบ UCS2 (16 บิต) ข้อความ “สวัสดิ์” แสดงดังตารางที่ 2.10

|                 | ส    | ว    | ั    | ศ    | ค    | ี    |
|-----------------|------|------|------|------|------|------|
| UCS2<br>(16bit) | 0E2A | 0E27 | 0E31 | 0E2A | 0E14 | 0E35 |

ตารางที่ 2.10 แสดงวิธีการแปลงตัวอักษรโดยวิธีการเข้ารหัสแบบ UCS2 (16bit) ข้อความ “สวัสดิ์”

## 2.15 คำสั่ง AT Command กับมือถือ

การสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) นั้นสามารถใช้ชุดคำสั่งที่เป็นมาตรฐานที่เรียกว่า AT Command ในการติดต่อเพื่อโต้ตอบตั้งค่าหรือส่งอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการโดยชุดคำสั่งพื้นฐานจะถูกกำหนดไว้ใน Hayes AT Command ซึ่งบริษัท Hayes เป็นผู้คิดค้นชุดคำสั่งนี้เพื่อใช้กับโมเด็มของคนและต่อมาได้กลายเป็นมาตรฐานสำหรับผู้ผลิตโมเด็มรายอื่นๆ โดยอาจจะมีชุดคำสั่งขยายเพื่อใช้เป็นการเฉพาะสำหรับผู้ผลิตรายนั้นๆ

การติดต่อกับมือถือก็เช่นกันเราสามารถใส่ชุดคำสั่งที่กำหนดไว้ใน GSM AT Command ซึ่งมีคำสั่งเพิ่มเติมที่เหมาะสมสำหรับการใช้งานและควบคุมมือถือ และเนื่องจากมีรายละเอียดที่ค่อนข้างมาก จึงจะพูดถึงเฉพาะคำสั่งที่จำเป็นสำหรับโครงงานนี้เท่านั้น การเชื่อมต่อคอมพิวเตอร์กับมือถือนั้น จะทำผ่านสายข้อมูล (Data Link) ซึ่งจะเป็นการเชื่อมต่อแบบอนุกรมโดยใช้โปรแกรมเทอร์มินอลต่างๆ เช่น ไฮเปอร์เทอร์มินอล (Hyper Terminal) ของ Windows ส่วนความเร็วในการสื่อสารมักจะใช้ 19200 bps

คำสั่ง AT-COMMAND สำหรับการส่งข้อความสั้นจาก GSM 07.05

|         |  |
|---------|--|
| AT+CNMI | เป็นคำสั่งเลือกสัญญาณข้อความสั้นใหม่                           |
| AT+CSCB | เป็นคำสั่งในการเลือกข้อความ Cell Broadcast                     |
| AT+CMGF | เป็นคำสั่งในการเลือกโหมดของข้อความที่จะส่ง                     |
| AT+CSCA | เป็นคำสั่งในการดูค่าของ ศูนย์บริการข้อความสั้น                 |
| AT+CMGL | เป็นคำสั่งเรียกอ่าน ข้อความ โดยให้แสดงตามชนิดที่ต้องการเรียกดู |
| AT+CMGR | เป็นคำสั่งที่ใช้เรียกอ่านข้อความที่ละอันเฉพาะอันที่ต้องการอ่าน |
| AT+CMGS | เป็นคำสั่งที่ใช้ในการส่งข้อความไปยัง Address ที่เลือกไว้       |
| AT+CMSS | เป็นคำสั่งที่ใช้ส่งข้อความจากซิมการ์ดไปยัง Address ที่เลือกไว้ |
| AT+CMGW | เป็นคำสั่งสำหรับเขียนข้อความเก็บไว้ในซิมการ์ด                  |
| AT+CMGD | เป็นคำสั่งสำหรับลบข้อความที่เก็บไว้ในซิมการ์ด                  |
| AT+CSMS | เป็นคำสั่งที่ใช้เลือกบริการข้อความ                             |
| AT+CPMS | เป็นคำสั่งที่ใช้เลือกหน่วยความจำของข้อความสั้น                 |
| AT+CMGC | เป็นคำสั่งที่ใช้ส่งคำสั่ง ข้อความสั้น                          |

### ลักษณะชุดคำสั่งของ AT+CNMI

| คำสั่ง  | คำตอบสนอง  |
|---|--|
| AT+CNMI=?                                     | +CNMI: (list of supported <mode>s), (list of supported <mt>s), (list of supported <bm>s), (list of supported <ds>s), (list of supported <bfr>s), |
| AT+CNMI?                                      | +CNMI: <mode>, <mt>, <bm>, <ds>, <bfr>   |
| AT+CNMI=[<mode>][,<mt>][,<bm>][,<ds>][,<bfr>] | OK/ERROR/+CMS ERROR  |

#### ตารางที่ 2.11 ลักษณะชุดคำสั่งของ AT+CNMI

|        |   |
|--------|---|
| <mode> | <p>0 ถ้าบัพเฟอร์เต็มแล้วมีสัญญาณเข้ามาใหม่จะเข้ามาแทนอันที่เก่าที่สุด</p> <p>1 ไม่รับ Unsolicited Result Code ใหม่เมื่อ TA-TE link ถูกจองไว้หรือไม่เช่นนั้นก็ Forward ไปยัง TE โดยตรง</p> <p>2 เก็บ Unsolicited Result Code ในบัพเฟอร์ของ TA เมื่อ TA-TE link ถูกจองไว้แล้วถูกส่งไปยัง TE เมื่อสิ้นสุดการจอง</p> <p>3 ทำการ Forward ค่า Unsolicited Result Code ไปที่ TE โดยตรง</p> |
| <mt>   | กฎสำหรับการเก็บข้อความสั้น ที่รับเข้ามาขึ้นอยู่กับวิธีการเข้ารหัสของข้อมูล การตั้งค่า Memory Format และค่านี้   |
| <bm>   | กฎสำหรับการเก็บ CBMs ที่รับเข้ามาขึ้นอยู่กับวิธีการเข้ารหัสของข้อมูล การเลือกรูปแบบของ CBM และค่านี้  |
| <ds>   | <p>0 ไม่มี SMS-STATUS-REPORT ส่งไปยัง TE</p> <p>1 SMS-STATUS-REPORT ส่งไปยัง TE โดยใช้ Unsolicited Result Code + CDS: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;PDU&gt; (PDU Mode Enable)</p> <p>2 ถ้า SMS-STATUS-REPORT ส่งไปใน ME/TA สัญญาณของ Location ของหน่วยความจำถูกส่งไป TE โดยใช้ Unsolicited Result Code</p>  |
| <bfr>  | 1 TA บัพเฟอร์ของ Unsolicited Result Code จะถูกจำกัดความในคำสั่งนี้  |

### ลักษณะชุดคำสั่งของ AT+CSCB

| คำสั่ง                             | ค่าตอบสนอง                         |
|------------------------------------|------------------------------------|
| AT+CSCB=?                          | +CSCB: (list of supported <mode>s) |
| AT+CSCB?                           | +CSCB:<mode>,<mids>,<dcss>         |
| AT+CSCB=[<mode>[,<mids>[,<dcss>]]] | OK/ERROR                           |

#### ตารางที่ 2.12 ลักษณะชุดคำสั่งของ AT+CSCB

|        |  |
|--------|--|
| <mode> | 0 รับข้อความ<br>1 ไม่รับข้อความ          |
| <mids> | ค่า CBM Message IDs: รูปแบบสตริง         |
| <dcss> | ค่า CBM Data Coding Schemes: รูปแบบสตริง |

### ลักษณะชุดคำสั่งของ AT+CMGF

| คำสั่ง           | ค่าตอบสนอง                         |
|------------------|------------------------------------|
| AT+CMGF=?        | +CMGF: (list of supported <mode>s) |
| AT+CMGF?         | +CMGF:<mode>                       |
| AT+CMGF=[<mode>] | OK/ERROR                           |

#### ตารางที่ 2.13 ลักษณะชุดคำสั่งของ AT+CMGF

|        |  |
|--------|--|
| <mode> | 0 เป็นโหมดพีดียู<br>1 เป็นโหมดตัวอักษร |
|--------|--|

### ลักษณะชุดคำสั่งของ AT+CSCA

| คำสั่ง                 | ค่าตอบสนอง          |
|------------------------|---------------------|
| AT+CSCA=?              | OK                  |
| AT+CSCA?               | +CSCA:<sca>,<tosca> |
| AT+CSCA=<sca>[,<mode>] | OK/ERROR            |

ตารางที่ 2.14 ลักษณะชุดคำสั่งของ AT+CSCA

<sca> Service-Center Address In String Format

<tosca> Service-Center Address Format

### ลักษณะชุดคำสั่งของ AT+CMGL

| คำสั่ง           | ค่าตอบสนอง  |
|------------------|---|
| AT+CMGL=?        | +CMGL: (list of supported <mode>s)  |
| AT+CMGL=[<stat>] | If PDU mode (+CMGF=0)<br>+CMGL:<index>,<stat>,[<alpha>],<length><CR><br><LF><pdu><CR><LF> |

ตารางที่ 2.15 ลักษณะชุดคำสั่งของ AT+CMGL

<stat> ตัวบอกสถานะของข้อความที่อยู่ในซิมการ์ด

0 ข้อความที่ได้รับมาแล้วยังไม่ได้อ่าน

1 ข้อความที่ได้รับมาแล้วอ่านแล้ว

2 ข้อความที่เก็บไว้สำหรับส่งแต่ยังไม่ได้ส่ง

3 ข้อความที่ส่งไปแล้ว

4 ข้อความทุกชนิด

<index> ตัวบอกตำแหน่งที่เราต้องการเลือกกว่าเป็นข้อความที่เท่าไรในซิมการ์ด

<length> ความยาวของส่วนของชุดข้อความ โดยจะนับแบบ octet

<pdu> ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นรวมกับส่วนของชุด

ข้อความ

### ลักษณะชุดคำสั่งของ AT+CMGR

| คำสั่ง          | ค่าตอบสนอง  |
|-----------------|---|
| AT+CMGR=?       | OK  |
| AT+CMGR=<index> | If PDU mode (+CMGF=0)<br>+CMGR:<stat>,[<alpha>],<length><CR><br><LF><pdu> |

### ตารางที่ 2.16 ลักษณะชุดคำสั่งของ AT+CMGR

|          |  |
|----------|--|
| <stat>   | ตัวบอกสถานะของข้อความที่อยู่ในซิมการ์ด<br>0 ข้อความที่ได้รับมาแล้วยังไม่ได้อ่าน<br>1 ข้อความที่ได้รับมาแล้วอ่านแล้ว<br>2 ข้อความที่เก็บไว้สำหรับส่งแต่ยังไม่ได้อ่าน<br>3 ข้อความที่ส่งไปแล้ว<br>4 ข้อความทุกชนิด |
| <index>  | ตัวบอกตำแหน่งที่เราต้องการเลือกกว่าเป็นข้อความที่เท่าไรในซิมการ์ด  |
| <length> | ความยาวของส่วนของชุดข้อความ โดยจะนับแบบ Octets   |
| <pdu>    | ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นรวมกับส่วนของชุดข้อความ  |

### ลักษณะชุดคำสั่งของ AT+CMGS

| คำสั่ง  | ค่าตอบสนอง  |
|---|---|
| AT+CMGS=?   | OK  |
| If PDU mode (+CMGF=0)<br>AT+CMGS:<length><CR>PDU is given<br><ctrl-Z/ESC> | If sending is successful:<br>+CMGS:<mr><br>If sending is not successful:<br>+CMS ERROR: <err> |

### ตารางที่ 2.17 ลักษณะชุดคำสั่งของ AT+CMGS

|                  |   |
|------------------|---|
| <length>         | ความยาวของส่วนของชุดข้อความ โดยจะนับแบบ Octets                          |
| <pdu><br>ข้อความ | ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นพร้อมกับส่วนของชุดข้อความ |
| <mr>             | จำนวนครั้งที่เราส่งข้อความสั้นหรือตัวอ้างอิงข้อความ                     |

### ลักษณะชุดคำสั่งของ AT+CMSS

| คำสั่ง                         | ค่าตอบสนอง  |
|--------------------------------|---|
| AT+CMSS=?                      | OK  |
| AT+CMSS:<index>[,<da>[,<tda>]] | If sending is successful:<br>+CMSS:<mr><br><br>If sending is not successful:<br>+CMS ERROR: <err> |

### ตารางที่ 2.18 ลักษณะชุดคำสั่งของ AT+CMSS

|         |   |
|---------|---|
| <index> | ตัวบอกตำแหน่งที่เราต้องการเลือกกว่าเป็นข้อความที่เท่าไรในซิมการ์ด   |
| <da>    | เป็นหมายเลขโทรศัพท์ที่เราต้องการส่งข้อความสั้น โดยจะอยู่ในรูป “หมายเลข” ซึ่งอยู่ในรหัสแอสกี                   |
| <tda>   | เป็นหมายเลขโทรศัพท์ที่เราต้องการส่งข้อความสั้น โดยจะอยู่ในรูป “+รหัสประเทศตามด้วยหมายเลข” ซึ่งอยู่ในรูปตัวเลข |
| <mr>    | จำนวนครั้งที่เราส่งข้อความสั้นหรือตัวอ้างอิงข้อความ   |

### ลักษณะชุดคำสั่งของ AT+CMGW

| คำสั่ง   | ค่าตอบสนอง                        |
|--|-----------------------------------|
| AT+CMSS=?  | OK                                |
| If PDU mode (+CMGF=0)<br>AT+CMGW=<length>[,<stat>]<CR>PDU is given<br><ctrl-Z/ESC> | +CMGW:<index><br>+CMS ERROR:<err> |

#### ตารางที่ 2.19 ลักษณะชุดคำสั่งของ AT+CMGW

|          |   |
|----------|---|
| <stat>   | ตัวบอกสถานะของข้อความที่อยู่ในซิมการ์ด<br>0 ข้อความที่ได้รับมาแล้วยังไม่ได้อ่าน<br>1 ข้อความที่ได้รับมาแล้วอ่านแล้ว<br>2 ข้อความที่เก็บไว้สำหรับส่งแต่ยังไม่ได้ส่ง<br>3 ข้อความที่ส่งไปแล้ว<br>4 ข้อความทุกชนิด |
| <index>  | ตัวบอกตำแหน่งที่เราต้องการเลือกกว่าเป็นข้อความที่เท่าไรในซิมการ์ด   |
| <length> | ความยาวของชุดข้อความ โดยจะนับแบบ Octets   |
| <pdu>    | ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นร่วมกับส่วนของชุดข้อความ  |

### ลักษณะชุดคำสั่งของ AT+CMGD

| คำสั่ง          | ค่าตอบสนอง           |
|-----------------|----------------------|
| AT+CMGD=?       | OK                   |
| AT+CMGD=<index> | OK/+ERROR/+CMS ERROR |

#### ตารางที่ 2.20 ลักษณะชุดคำสั่งของ AT+CMGD

|         |   |
|---------|---|
| <index> | ตัวบอกตำแหน่งที่เราต้องการเลือกกว่าเป็นข้อความที่เท่าไรในซิมการ์ด |
|---------|---|

### ลักษณะชุดคำสั่งของ AT+CSMS

| คำสั่ง              | ค่าตอบสนอง                                     |
|---------------------|--|
| AT+CSMS=?           | +CSMS: (list of supported <service>s)          |
| AT+CSMS?            | +CSMS: <service>, <mt>, <mo>, <bm>             |
| AT+CSMS=[<service>] | +CSMS: <mt>, <mo>, <bm><br>OK/ERROR/+CMS ERROR |

ตารางที่ 2.21 ลักษณะชุดคำสั่งของ AT+CSMS

|           |   |
|-----------|---|
| <service> | 0 GSM 3.40 และ 3.41   |
| <mt>      | 1 รองรับรูปแบบ Mobile Terminate Message<br>2 ไม่รองรับรูปแบบ Mobile Terminate Message |
| <mo>      | 1 รองรับรูปแบบ Mobile Originate Message<br>2 ไม่รองรับรูปแบบ Mobile Originate Message |
| <bm>      | 1 รองรับรูปแบบ Broadcast Type Message<br>2 ไม่รองรับรูปแบบ Broadcast Type Message     |

### ลักษณะชุดคำสั่งของ AT+CPMS

| คำสั่ง                               | ค่าตอบสนอง  |
|--------------------------------------|---|
| AT+CPMS=?                            | +CPMS: (list of supported <mem1>s), (list of supported <mem2>s), (list of supported <mem3>s), |
| AT+CPMS?                             | +CPMS:<mem1>,<use1>,<total1>,<mem2>,<use2>,<total2><br>,<mem3>,<use3>,<total3>                |
| AT+CPMS=<mem1>[,<mem2>,<br>[<mem3>]] | +CPMS:<use1>,<total1>,<use2>,<total2>,<use3>,<total3><br>OK/ERROR/+CMS ERROR                  |

ตารางที่ 2.22 ลักษณะชุดคำสั่งของ AT+CPMS

|          |   |
|----------|---|
| <mem1>   | ส่วนความจำสำหรับอ่านและลบข้อความ            |
| <mem2>   | ส่วนความจำสำหรับเขียนและส่งข้อความ          |
| <mem3>   | ส่วนความจำสำหรับข้อความที่รับมาเก็บไว้      |
| <memx>   | ส่วนความจำสำหรับอ่านและลบข้อความ            |
| <usex>   | จำนวนข้อความที่เก็บอยู่ใน <memx>            |
| <totalx> | จำนวนข้อความทั้งหมดที่สามารถเก็บได้ใน<memx> |

### ลักษณะชุดคำสั่งของ AT+CMGC

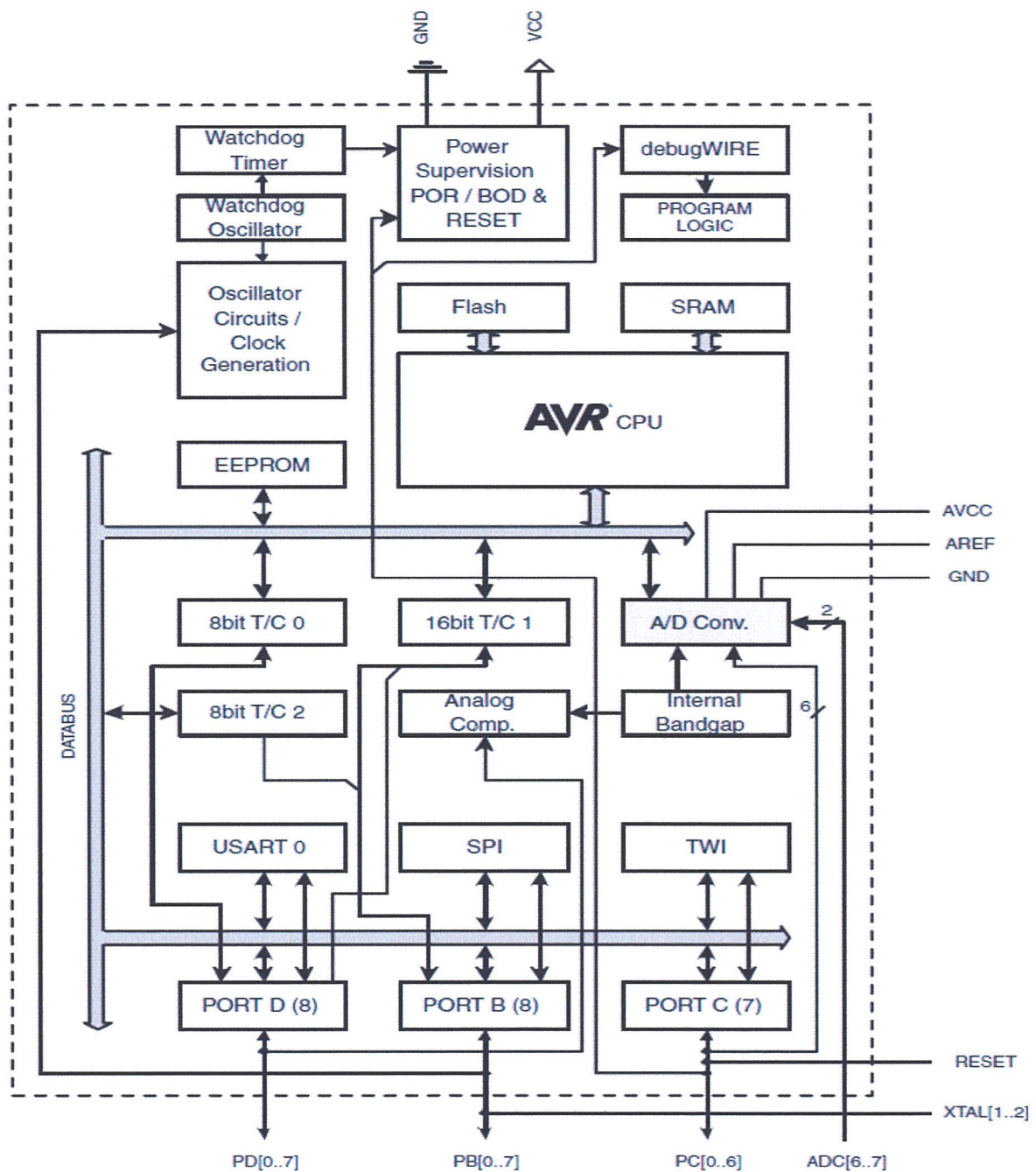
| คำสั่ง  | ค่าตอบสนอง  |
|---|---|
| AT+CMGC=?   | OK  |
| If PDU mode (+CMGF=0)<br>AT+CMGC=<length><CR>PDU is given<br><ctrl-Z/ESC> | If sending is successful:<br>+CMGC:<mr><br><br>If sending is not successful:<br>+CMS ERROR: <err> |

### ตารางที่ 2.23 ลักษณะชุดคำสั่งของ AT+CMGC

|                  |   |
|------------------|---|
| <length>         | ความยาวของชุดข้อความ โดยจะนับแบบ Octets                                 |
| <pdu><br>ข้อความ | ข้อความที่เป็นส่วนของศูนย์บริการส่งข้อความสั้นพร้อมกับส่วนของชุดข้อความ |
| <mr>             | จำนวนครั้งที่เราส่งข้อความสั้นหรือตัวอ้างอิงข้อความ                     |

## 2.16 สถาปัตยกรรมไมโครคอนโทรลเลอร์ AVR

ไมโครคอนโทรลเลอร์ AVR เป็นหนึ่งในไมโครคอนโทรลเลอร์ ที่ผลิตโดยบริษัท ATMEL (ผู้นำทางด้านไมโครคอนโทรลเลอร์ตระกูล MCS-51) AVR จัดเป็นไมโครคอนโทรลเลอร์ตระกูลใหม่ จาก ATMEL มีสถาปัตยกรรมแบบ RISC (reduce instruction set computer) คือหนึ่งคำสั่งงานใช้สัญญาณนาฬิกาเพียงลูกเดียว เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพและความสามารถสูงแบ่งออกเป็นหลายอนุกรม ในแต่ละอนุกรมยังแบ่งออกเป็นหลายเบอร์ เพื่อรองรับความต้องการที่แตกต่างของผู้ใช้งาน ในขณะที่ยังคงความประสิทธิภาพที่เท่ากัน สำหรับไมโครคอนโทรลเลอร์ AVR ที่นำเสนอในเล่มนี้จะเป็นเบอร์ ATmega16 รายละเอียดและคุณสมบัติภายในไมโครคอนโทรลเลอร์ ATmega16 แสดงดังบล็อกไดอะแกรมรูปที่ 2.17



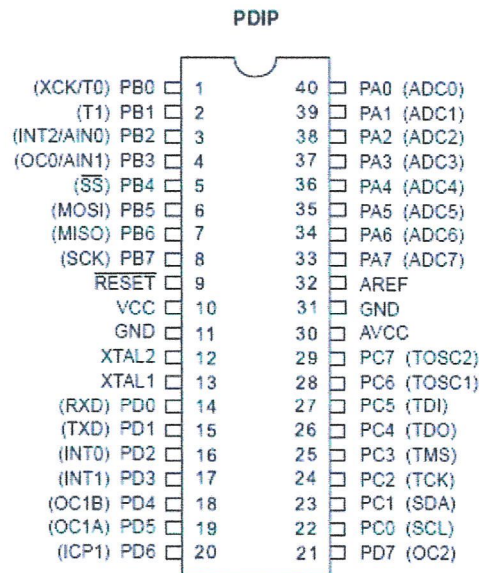
รูปที่ 2.17 แสดงบล็อกไดอะแกรม AVR ( ATmega16 )

### 2.16.1 คุณสมบัติที่สำคัญ

- สถาปัตยกรรมภายในเป็นแบบ Advanced RISC (Reduce Instruction Set Computer)
- มีคำสั่งควบคุมการทำงานมากกว่า 100 คำสั่ง โดยมีความเร็วในการประมวลผล 1 คำสั่ง ต่อ 1 สัญญาณนาฬิกา (1 MIP/1 MHz)
- มีรีจิสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว (ทำให้สะดวกต่อการพัฒนาโปรแกรมด้วยภาษา C เป็นอย่างมาก)
- ความเร็วในการทำงาน 1 MIPS ต่อ 1 MHz และมากถึง 16 MIPS เมื่อใช้ความถี่ 16 MHz (ความสามารถในการใช้งานความถี่สัญญาณนาฬิกาขึ้นอยู่กับเบอร์ที่เลือกใช้งาน)
- หน่วยความจำ ROM แบบ Flash (มีโหมดป้องกันหน่วยความจำ) ขนาด 16 กิโลไบต์ (เขียน/ลบได้ 1,000 ครั้ง)
- หน่วยความจำข้อมูลแบบ EEPROM (มีโหมดป้องกันหน่วยความจำ) ขนาด 512 ไบต์ (เขียน/ลบได้ 10,000 ครั้ง)
- หน่วยความจำข้อมูลแบบ SRAM 1 กิโลไบต์
- ไทเมอร์/เคาน์เตอร์ทั้งแบบ 8 บิตและ 16 บิต พร้อมปริสเกลเลอร์
- มีระบบตรวจสอบความผิดพลาดในการทำงานของซอฟต์แวร์ (Watchdog Timer with On-Chip Oscillator)
- โมดูลสร้างสัญญาณ PWM (Pulse Width Modulator) มีจำนวน 4 ช่อง
- มีโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC) ขนาด 10 บิต มากถึง 8 ช่อง
- โมดูลเปรียบเทียบแรงดันอะนาลอก (Analog Comparator)
- การสื่อสารข้อมูลอนุกรมมีทั้งแบบ UART (Universal Asynchronous Receiver Transmitters) หรือแบบ RS232, SPI และแบบ PC เป็นต้น
- พอร์ตอินพุตเอาต์พุตขึ้นอยู่กับเบอร์ AVR ที่เลือกใช้งาน มีตั้งแต่ 8 ขา จนมากกว่า 100 ขาพอร์ต (AT mega 16 มีขาพอร์ตอินพุต 32 ขา)
- แรงดันไฟเลี้ยงและความเร็วในการทำงานขึ้นอยู่กับเบอร์ AVR ที่เลือกใช้งาน
- ระบบการอินเตอร์รัพท์จากภายในและภายนอก (Internal and External Interrupt)
- ระบบการรีเซ็ตแบบฮาร์ด โนมัลติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้าไมโครคอนโทรลเลอร์ (Power on reset)

### 2.16.2 ขาพอร์ตอินพุตเอาต์พุต

ขาพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ ATmega16 มีจำนวน 40 ขา โดยแบ่งเป็น ขาพอร์ตอินพุตเอาต์พุตอิสระ จำนวน 32 ขาประกอบไปด้วย PA, PB, PC, PD ขนาด 8 บิต และขาพอร์ตที่เกี่ยวข้องกับสัญญาณนาฬิกาจำนวน 2 ขาพอร์ต คือ AREF และ AVCC รายละเอียดขาพอร์ตทั้งหมดแสดงดังรูปที่ 2.18

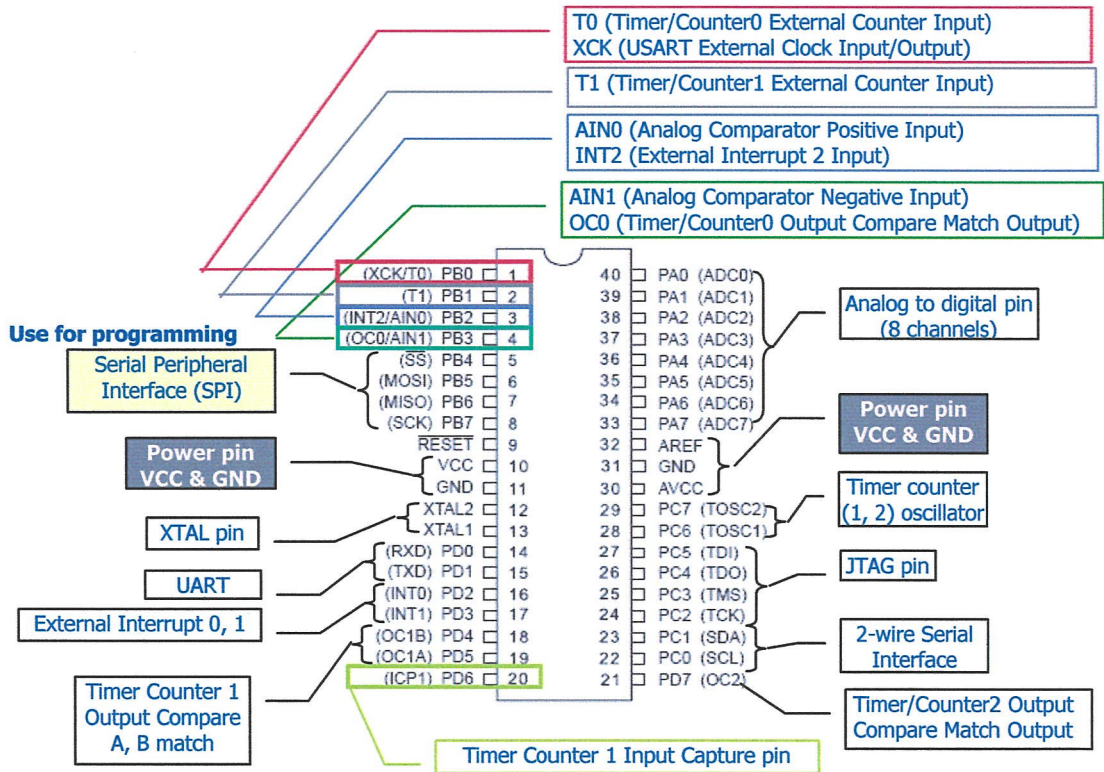


รูปที่ 2.18 ขาพอร์ต AVR (ATmega16) ตัวถังแบบ PDIP

รายละเอียดในแต่ละขาพอร์ตมีดังนี้

- **VCC** - ขาแรงดันไฟตรงให้กับ CPU
- **GND** - ขากราวด์
- **Port A (PA0..PA7)** - ขาพอร์ตเป็นอินพุตเอาต์พุตดิจิทัล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และสามารถกำหนดใช้งานเป็นพอร์ตอินพุตสัญญาณอนาล็อก (A/D Converter) ได้
- **Port B (PB0..PB7)** - เป็นขาพอร์ตอินพุตดิจิทัล กำหนดการพูลอัพภายในขาพอร์ตได้ และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาสำหรับโปรแกรมชิพ ขาป้อนสัญญาณนาฬิกาภายนอก เป็นต้น
- **Port C (PC0..PC7)** - นอกจากจะเป็นขาอินพุตเอาต์พุตดิจิทัล ที่กำหนดการพูลอัพภายในขาพอร์ตได้ แล้วยังเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อกับดีบั๊กและโปรแกรมด้วยการเชื่อมต่อแบบ JTAG เป็นต้น
- **Port D (PD0..PD7)** - เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัล กำหนดการพูลอัพภายในขาพอร์ตได้ และขาพอร์ตทำหน้าที่พิเศษ เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์ เนื่องจากสัญญาณภายนอก เป็นต้น

- **RESET** – ขารีเซตวงจร
- **XTAL1** – ขาคอนกรีตอลอสซิลเลเตอร์ ช่องที่ 1 ด้านอินพุต
- **XTAL2** – ขาคอนกรีตอลอสซิลเลเตอร์ ช่องที่ 2 ด้านเอาต์พุต
- **AVCC** – ขาแรงดันสำหรับพอร์ต A และ โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัล
- **AREF** – ขาแรงดันอนาลอกอ้างอิงสำหรับ โมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัล



รูปที่ 2.19 รายละเอียดขาของ ATmega 16

### 2.16.3 ไทเมอร์/เคาน์เตอร์

โมดูลไทเมอร์/เคาน์เตอร์ กับสัญญาณ PWM ขนาด 8 บิต ของ AVR นอกจากจะมีคุณสมบัติใช้งานเป็นไทเมอร์หรือเคาน์เตอร์แล้ว ยังมีคุณสมบัติเพิ่มเติมในการสร้างสัญญาณ PWM โดยอาศัยโมดูลการเปรียบเทียบข้อมูลในการสร้างสัญญาณ คุณสมบัติที่สำคัญของไทเมอร์/เคาน์เตอร์ มีดังนี้

1. ใช้งานเป็นไทเมอร์นับสัญญาณนาฬิกาภายใน (ตัวจับเวลา) หรือเป็นเคาน์เตอร์นับสัญญาณนาฬิกาภายนอก(ตัวนับสัญญาณ) ผ่านทางขาพอร์ต PBO (XCK/T0)
2. ใช้งานเป็นฐานเวลาในโหมดเปรียบเทียบข้อมูลเพื่อสร้างสัญญาณที่ขาพอร์ต PB3 (OC0/AIN1) เพื่อกำหนดหน้าที่ใดหน้าที่หนึ่ง เช่น เมื่อเปรียบเทียบข้อมูลตรงกันจะสลับสถานะขาพอร์ต เมื่อเปรียบเทียบข้อมูลตรงกันจะเซตขาพอร์ต (Set OC0 on compare math) หรือเมื่อเปรียบเทียบข้อมูลตรงกันจะทำการเคลียร์ขาพอร์ต (Clear OC0 on compare math)

- ใช้งานเป็นฐานเวลาในโหมดสร้างสัญญาณ PWM ผ่านทางขาพอร์ต OC0 ในรูปแบบ Fast PWM Mode หรือ Phase Correct PWM Mode

### รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานไทเมอร์/เคาน์เตอร์ มีดังนี้

- รีจิสเตอร์ TCNT0 (Timer/Counter0 Register)  
รีจิสเตอร์ที่ใช้ในการนับ การนับขึ้นหรือลงขึ้นอยู่กับการเซตบิตในรีจิสเตอร์ TCCR0
- รีจิสเตอร์ OCRT0 (Output Compare Register)  
รีจิสเตอร์ ORC0 ใช้กำหนดค่าเพื่อเปรียบเทียบกับค่าในรีจิสเตอร์ TCNT0 เมื่อตรงกัน จะให้ผลลัพธ์ตามเงื่อนไขที่กำหนดไว้ในรีจิสเตอร์ TCCR0
- รีจิสเตอร์ TCCR0 (Timer/Counter Control Register)  
รีจิสเตอร์กำหนดแหล่งสัญญาณนาฬิกาที่ใช้และกำหนดโหมดการทำงาน
- รีจิสเตอร์ TIMSK (Timer/Counter Interrupt Mask Register)  
รีจิสเตอร์เอ็นเอเบิลการใช้งานอินเตอร์รัปต์ของไทเมอร์/เคาน์เตอร์0 โดยกำหนดเงื่อนไขการเกิดอินเตอร์รัปต์ได้ 2 รูปแบบคือ
  - บิต OCIE0 (Timer/ Counter0 Output Compare Match Interrupt Enable)  
เอ็นเอเบิลการใช้งานอินเตอร์รัปต์เนื่องจากเปรียบเทียบข้อมูลได้ตรงกัน
  - บิต TOIE0 (Timer/ Counter0 Overflow Interrupt Enable)  
เอ็นเอเบิลการใช้งานอินเตอร์รัปต์จากการเกิดโอเวอร์โฟลว์เนื่องจากการนับ
- รีจิสเตอร์ TIFR (Timer/Counter Interrupt Flag Register)  
รีจิสเตอร์เอ็นเอเบิลการใช้งานอินเตอร์รัปต์ของไทเมอร์/เคาน์เตอร์
  - แฟล็ก OCF0 (Output Compare Flag 0) ถูกเซตเมื่อเกิดเหตุการณ์ ข้อมูลเปรียบเทียบตรงกัน
  - แฟล็ก TOV0 (Timer/ Counter0 Overflow Flag) ถูกเซตเมื่อข้อมูลเกิดโอเวอร์โฟลว์

#### 2.16.4 โหมดการทำงานของไทเมอร์/เคาน์เตอร์

ไทเมอร์/เคาน์เตอร์ กำหนดโหมดการทำงานได้ 4 โหมดคือ

- โหมดไทเมอร์/เคาน์เตอร์ หรือ โหมดปกติ (Normal Mode)
- โหมดเคลียร์ไทเมอร์เมื่อเปรียบเทียบข้อมูลตรงกัน (Clear Timer on Compare Match (OTC) Mode)
- โหมดสร้างสัญญาณ PWM แบบ Fast PWM Mode
- โหมดสร้างสัญญาณ PWM แบบ Phase Correct PWM Mode

### 2.16.5 การกำหนดโหมดการทำงาน

การกำหนดโหมดการทำงานของไทเมอร์/เคาน์เตอร์ จะเกี่ยวข้องกับรีจิสเตอร์ TCCR0 โดยมีรายละเอียดดังนี้

#### รีจิสเตอร์ TCCR0 (Timer/Counter Control Register)

| บิตที่      | 7    | 6     | 5     | 4     | 3     | 2    | 1    | 0    |
|-------------|------|-------|-------|-------|-------|------|------|------|
| ชื่อบิต     | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
| Read/Write  | W    | R/W   | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  |
| ค่าเริ่มต้น | 0    | 0     | 0     | 0     | 0     | 0    | 0    | 0    |

ตารางที่ 2.24 แสดงบิตของรีจิสเตอร์ TCCR0

- บิตที่ 7 : บิต FOC0 (Force Output Compare)  
บิต FOC0 จะเขียนได้อย่างเดียวและทำงานเมื่อไม่ได้อยู่ในโหมดการสร้างสัญญาณ PWM หรือ WGM00 เท่ากับ 0 บิตนี้จะใช้เป็นเครื่องมือเกี่ยวกับสัญญาณ สไตรบ
- บิตที่ 3,6 : บิต WGM01 และ WGM00 (Waveform Generation Mode)  
จะกำหนดโหมดสร้างสัญญาณของไทเมอร์/เคาน์เตอร์ กับสัญญาณ PWM รายละเอียดดังตารางที่ 2.25

| โหมด | WGM01<br>(CTC0) | WGM00<br>(PWM0) | การทำงานของ<br>ไทเมอร์/เคาน์เตอร์ | ค่าสูงสุด<br>(TOP) | การปรับปรุ้ค่าของ<br>รีจิสเตอร์ OCR0 | การเซต<br>แฟล็ก<br>TOV0 |
|------|-----------------|-----------------|-----------------------------------|--------------------|--------------------------------------|-------------------------|
| 1    | 0               | 0               | Normal                            | 0xFF               | Immediate                            | MAX                     |
| 2    | 0               | 1               | PWM,Phase<br>Correct              | 0xFF               | TOP                                  | BOTTOM                  |
| 3    | 1               | 0               | CTC                               | OCR0               | Immediate                            | MAX                     |
| 4    | 1               | 1               | Fast PWM                          | 0xFF               | TOP                                  | MAX                     |

ตารางที่ 2.25 บิตกำหนดรูปแบบสัญญาณของไทเมอร์/เคาน์เตอร์

โดยที่ค่า BOTTOM, MAX และ TOP มีรายละเอียดดังนี้

|        |  |
|--------|--|
| BOTTOM | มีค่าเท่ากับ 0x00 (เท่ากับ 0 ในฐานสิบ)   |
| MAX    | มีค่าเท่ากับ 0xFF (เท่ากับ 255 ในฐานสิบ)   |
| TOP    | ค่าสูงสุดของการนับ โดยกำหนดให้มีค่าสูงสุดเท่ากับ MAX หรือมีค่าเท่ากับที่กำหนดในรีจิสเตอร์ OCR0 (ค่าที่กำหนดนี้จะขึ้นอยู่กับโหมดการทำงาน) |

ตารางที่ 2.26 แสดงรายละเอียดของ ค่า BOTTOM, MAX และ TOP

- บิตที่ 5,4 : บิต COM01, COM00 (Compare Match Output Mode)  
บิตสำหรับกำหนดโหมดการเปรียบเทียบการสร้างสัญญาณเกี่ยวกับบิตที่ 3 และ 6 เมื่อทำงานในโหมด ไทเมอร์/เคาน์เตอร์หรือโหมดปกติ บิต COM01 และ COM00 จะเซตเป็น “0” หรือเป็นการปิดการใช้งานโหมด CTC และ โหมดสร้างสัญญาณ PWM
- บิตที่ 2-0 : บิต CS02 ,CS01 และ CS00 (Clock Slect)  
บิตกำหนดแหล่งสัญญาณนาฬิกาที่ใช้งานหากกำหนดให้เป็นการใช้งานสัญญาณนาฬิกาภายในก็จะทำงานในโหมดไทเมอร์และหากกำหนดให้ใช้เป็นแหล่งสัญญาณนาฬิกาจากภายนอกจะเป็นการกำหนดให้ทำงานในโหมดเคาน์เตอร์ ดังมีรายละเอียดดังตารางที่ 2.27

| CS02 | CS01 | CS00 | รายละเอียด  |
|------|------|------|---|
| 0    | 0    | 0    | ไม่ใช่สัญญาณนาฬิกา (ไทเมอร์/เคาน์เตอร์หยุดทำงาน)        |
| 0    | 0    | 1    | Clk I/O /1 (ไม่ใช่ ปรีสเกลเลอร์)                        |
| 0    | 1    | 0    | Clk I/O /8 (สัญญาณนาฬิกาหารด้วย ปรีสเกลเลอร์ 8)         |
| 0    | 1    | 1    | Clk I/O /64 (สัญญาณนาฬิกาหารด้วย ปรีสเกลเลอร์ 64)       |
| 1    | 0    | 0    | Clk I/O /256 (สัญญาณนาฬิกาหารด้วย ปรีสเกลเลอร์ 256)     |
| 1    | 0    | 1    | Clk I/O /1024 (สัญญาณนาฬิกาหารด้วย ปรีสเกลเลอร์ 1024)   |
| 1    | 1    | 0    | ใช้สัญญาณนาฬิกาจากภายนอกที่ขา T0 ทำงานที่ขอบขาของสัญญาณ |
| 1    | 1    | 1    | ใช้สัญญาณนาฬิกาจากภายนอกที่ขา T0 ทำงานที่ขอบขาของสัญญาณ |

ตารางที่ 2.27 บิตกำหนดแหล่งสัญญาณนาฬิกาของไทเมอร์

\*หมายเหตุ ปรีสเกลเลอร์ (Prescaler) คือตัวหารหรือลดทอนสัญญาณนาฬิกา (สัญญาณนาฬิกาจากคริสตอลที่ขา XTAL1 และ XTAL2)

### รีจิสเตอร์ TIFR (Timer/Counter Interrupt Flag Register)

| บิตที่      | 7    | 6    | 5    | 4     | 3     | 2    | 1    | 0    |
|-------------|------|------|------|-------|-------|------|------|------|
| ชื่อบิต     | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 |
| Read/Write  | R/W  | R/W  | R/W  | R     | R     | R    | R/W  | R/W  |
| ค่าเริ่มต้น | 0    | 0    | 0    | 0     | 0     | 0    | 0    | 0    |

ตารางที่ 2.28 แสดงบิตของรีจิสเตอร์ TIFR

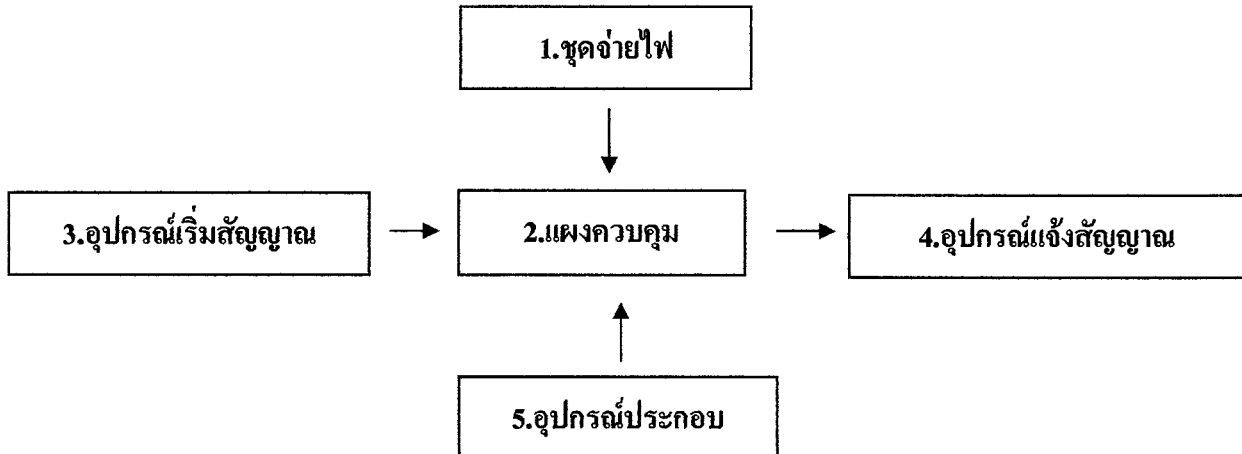
- บิตที่ 1 : บิต OCF0 (Timer/Counter Compare Flag)  
บิต OCF0 จะถูกเซตเมื่อเกิดการเปรียบเทียบข้อมูลระหว่างรีจิสเตอร์ TCNT0 และ OCR0 มีค่าเท่ากัน และถูกเคลียร์ค่าด้วยกระบวนการทางฮาร์ดแวร์ เมื่อมีการเปิดใช้งานฟังก์ชันอินเตอร์รัปต์หรือเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ โดยการเขียนค่า "1" ไปที่บิต OCF0
- บิตที่ 0 : บิต TOV0 (Timer/Counter Overflow Flag)  
บิตแสดงการเกิดโอเวอร์โฟลว์ เนื่องจากการนับของรีจิสเตอร์ TCNT0 การเกิดโอเวอร์โฟลว์ คือการเพิ่มค่าจนถึง 0xFF (255) บิต TOV0 จะถูกเซต และถูกเคลียร์ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อมีการใช้งานอินเตอร์รัปต์ฟังก์ชันเนื่องจากการเกิดโอเวอร์โฟลว์ และมีการเปิดใช้งานอินเตอร์รัปต์ดังกล่าว (เซตบิต TOIE0 ในรีจิสเตอร์ TIMSK และบิต I ในรีจิสเตอร์ SREG) หรือเคลียร์ด้วยกระบวนการทางซอฟต์แวร์โดยการเขียนค่า "1" ไปที่บิต TOV0

#### 2.17 ระบบสัญญาณเตือนอัคคีภัย (Fire Alarm System)

อัคคีภัยก่อให้เกิดความสูญเสียทั้งชีวิตและทรัพย์สิน สาเหตุส่วนใหญ่มาจากขณะที่เริ่มเกิดเพลิงไหม้จะไม่มีใครอยู่หรือเกิดในบริเวณที่ไม่มีคนเห็น กว่าจะรู้ตัวเพลิงก็ลุกลาม จนเกินกำลังที่คนไม่ก็คนหรืออุปกรณ์ดับเพลิงขนาดเล็กที่มีอยู่ในอาคารจะทำการสกัดไฟได้ ดังนั้นจึงจำเป็นต้องมีระบบสัญญาณเตือนอัคคีภัยติดตั้งไว้ในอาคาร เพื่อให้สามารถรับรู้เหตุการณ์ล่วงหน้าก่อนที่ไฟจะลุกลามจนควบคุมไม่ได้

### 2.17.1 ส่วนประกอบของระบบสัญญาณเตือนอัคคีภัย

ส่วนประกอบที่สำคัญของระบบเตือนอัคคีภัยมี 5 ส่วนใหญ่ๆ ซึ่งทำงานเชื่อมโยงกัน ดังแสดงในแผนภาพ



รูปที่ 2.20 แสดงส่วนประกอบของระบบเตือนอัคคีภัย

#### ชุดจ่ายไฟ (Power Supply)

ชุดจ่ายไฟ เป็นอุปกรณ์แปลงกำลังไฟฟ้าของแหล่งจ่ายไฟมาเป็นกำลังไฟฟ้ากระแสตรง ที่ใช้ปฏิบัติงานของระบบและจะต้องมีระบบไฟฟ้าสำรอง เพื่อให้ระบบทำงานได้ในขณะที่ไฟดับ

#### แผงควบคุม (Fire Alarm Control Panel)

เป็นส่วนควบคุมและตรวจสอบการทำงานของอุปกรณ์และส่วนต่างๆ ในระบบทั้งหมด จะประกอบด้วยวงจรควบคุมคอยรับสัญญาณจากอุปกรณ์เริ่มสัญญาณ, วงจรทดสอบการทำงาน, วงจรป้องกันระบบ, วงจรสัญญาณแจ้งการทำงานในสภาวะปกติ และภาวะขัดข้อง เช่น สายไฟจากอุปกรณ์ตรวจจับขาด, แบตเตอรี่ต่ำหรือ ไฟจ่ายตู้แผงควบคุม โคนตัดขาด เป็นต้น ตู้แผงควบคุม (FCP) จะมีสัญญาณไฟและเสียงแสดงสภาวะต่างๆ บนหน้าตู้ เช่น

- Fire Lamp : จะติดเมื่อเกิดเพลิงไหม้
- Main Sound Buzzer : จะมีเสียงดังขณะแจ้งเหตุ
- Zone Lamp : จะติดค้างแสดงโซนที่เกิด Alarm
- Trouble Lamp : แจ้งเหตุขัดข้องต่างๆ
- Control Switch : สำหรับการควบคุม เช่น เปิด/ปิดเสียงที่ตู้และกระดิ่ง, ทดสอบการทำงานตู้, ทดสอบ
- Battery Reset ระบบหลังเหตุการณ์เป็นปกติ

### อุปกรณ์เริ่มสัญญาณ (Initiating Devices)

เป็นอุปกรณ์ต้นกำเนิดของสัญญาณเตือนอัคคีภัย ซึ่งแบ่งออกเป็น 2 ชนิด คือ

- อุปกรณ์เริ่มสัญญาณจากบุคคล (Manual Station) ได้แก่ สถานีแจ้งสัญญาณเตือนอัคคีภัยแบบใช้มือกด (Manual Push Station)

- อุปกรณ์เริ่มสัญญาณโดยอัตโนมัติ เป็นอุปกรณ์อัตโนมัติที่มีปฏิกิริยาไวต่อสภาวะตามระยะต่างๆ ของการเกิดเพลิงไหม้ ได้แก่ อุปกรณ์ตรวจจับควัน(Smoke Detector) อุปกรณ์ตรวจจับความร้อน (Heat Detector) อุปกรณ์ตรวจจับเปลวไฟ(Flame Detector) อุปกรณ์ตรวจจับแก๊ส(Gas Detector)

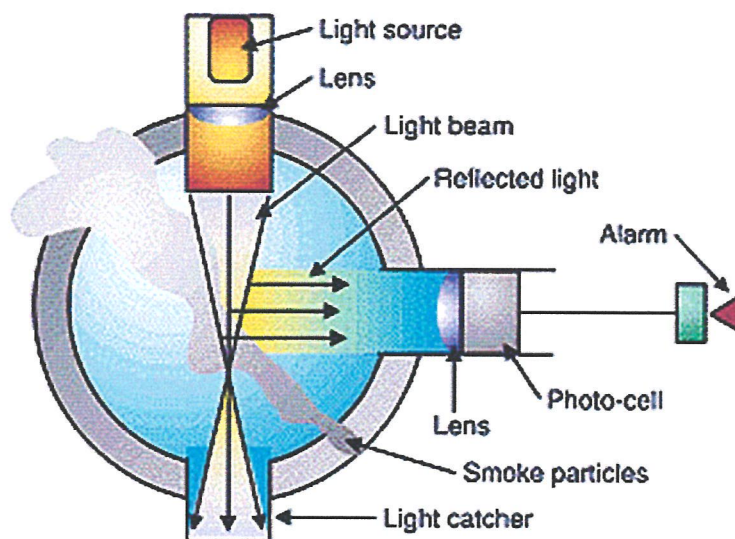
### อุปกรณ์แจ้งสัญญาณด้วยเสียงและแสง (Audible & Visual Signalling Alarm Devices)

หลังจากอุปกรณ์เริ่มสัญญาณทำงานโดยส่งสัญญาณมายังตู้ควบคุม (FCP) แล้ว FCP จึงส่งสัญญาณออกมาโดยผ่านอุปกรณ์ ได้แก่ กระดิ่ง, ไซเรน, ไฟสัญญาณ เป็นต้น เพื่อให้ผู้อยู่อาศัย, ผู้รับผิดชอบหรือเจ้าหน้าที่ดับเพลิงได้ทราบว่าเกิดเหตุเพลิงไหม้เกิดขึ้น

### อุปกรณ์ประกอบ (Auxiliary Devices)

เป็นอุปกรณ์ที่ทำงานเชื่อมโยงกับระบบอื่นที่เกี่ยวข้องกับการควบคุมป้องกัน และดับเพลิงโดยจะถ่ายทอดสัญญาณระหว่างระบบเตือนอัคคีภัยกับระบบอื่น เช่น

- ส่งสัญญาณกระตุ้นการทำงานของระบบบังคับลิฟต์ลงชั้นล่าง, การปิดพัดลมในระบบปรับอากาศ, เปิดพัดลมในระบบระบายอากาศ, เปลี่ยนแปลงเพื่อควบคุมควันไฟ, ควบคุมเปิดประตูทางออก, เปิดประตูหนีไฟ, ปิดประตูกันควันไฟ, ควบคุมระบบกระจายเสียง และการประกาศแจ้งข่าว, เปิดระบบดับเพลิง เป็นต้น
- รับสัญญาณของระบบอื่นมากระตุ้นการทำงานของระบบสัญญาณเตือนอัคคีภัย เช่น จากระบบพ่นน้ำปัดดับเพลิง ระบบดับเพลิงด้วยสารเคมีชนิดอัตโนมัติ เป็นต้น



รูปที่ 2.21 แสดงการประกอบอุปกรณ์ชนิดต่างๆ

## 2.18 ชนิดอุปกรณ์เริ่มสัญญาณแบบอัตโนมัติ (Automatic Initiation Devices)

### 2.18.1 อุปกรณ์ตรวจจับควัน (Smoke Detector) แบ่งออกเป็น 2 แบบดังนี้



รูปที่ 2.22 อุปกรณ์ตรวจจับควันไฟ (Smoke Detector)

- **อุปกรณ์ตรวจจับควันชนิดไอออนไนเซชัน (Ionization Smoke Detector)** อุปกรณ์ชนิดนี้เหมาะสำหรับใช้ตรวจจับสัญญาณควันในระยะเริ่มต้นที่มีอนุภาคของควันเล็กมาก Ionization Detector ทำงานโดยใช้หลักการเปลี่ยนแปลงคุณสมบัติทางไฟฟ้า โดยใช้สารกัมมันตภาพรังสีปริมาณน้อยมากซึ่งอยู่ใน Chamber ซึ่งจะทำปฏิกิริยากับอากาศที่อยู่ระหว่างขั้วบวกและลบ ทำให้ความนำไฟฟ้า (Conductivity) เพิ่มขึ้นมีผลให้กระแสสามารถไหลผ่านได้โดยสะดวก เมื่อมีอนุภาคของควันเข้ามาใน Sensing Chamber นี้ อนุภาคของควันจะไปรวมตัวกับ ไอออน จะมีผลทำให้การไหลของกระแสลดลงด้วย ซึ่งทำให้ตัว ตรวจจับควันแจ้งสถานะ Alarm ทันที

- **อุปกรณ์ตรวจจับควันชนิดโฟโตอิเล็กทริก (Photoelectric Smoke Detector)** เหมาะสำหรับใช้ตรวจจับสัญญาณควันในระยะที่มีอนุภาคของควันที่ใหญ่ขึ้น Photoelectric Smoke Detector ทำงานโดยใช้หลักการสะท้อนของแสง เมื่อมีควันเข้ามาในตัวตรวจจับควันจะไปกระทบกับแสงที่ออกมาจาก Photoemitter ซึ่งไม่ได้ส่องตรงไปยังอุปกรณ์รับแสง Photo receptor แต่แสงดังกล่าวบางส่วนจะสะท้อนอนุภาคควันและหักเหเข้าไปที่ Photo receptor ทำให้วงจรตรวจจับของตัวตรวจจับควันส่งสัญญาณแจ้ง Alarm

### 2.18.2 อุปกรณ์ตรวจจับความร้อน (Heat Detector)

อุปกรณ์ตรวจจับความร้อน เป็นอุปกรณ์แจ้งอัคคีภัยอัตโนมัติรุ่นแรกๆ มีหลายชนิด ซึ่งนับได้ว่าเป็นอุปกรณ์ที่ราคาถูกที่สุดและมีสัญญาณหลอก (Fault Alarm) น้อยที่สุดในปัจจุบัน อุปกรณ์ตรวจจับความร้อนที่นิยมใช้กันมีดังต่อไปนี้



รูปที่ 2.23 อุปกรณ์ตรวจจับความร้อน (Heat Detector)

- **อุปกรณ์ตรวจจับความร้อนชนิดจับอัตราการเพิ่มของอุณหภูมิ (Rate-of-Rise Heat Detector)**  
 อุปกรณ์ชนิดนี้จะทำงานเมื่อมีอัตราการเพิ่มของอุณหภูมิเปลี่ยนแปลงไปตั้งแต่ 10 องศาเซลเซียส ใน 1 นาที ส่วนลักษณะการทำงาน อากาศในส่วนด้านบนของส่วนรับความร้อนเมื่อถูกความร้อน จะขยายตัวอย่างรวดเร็วมากจนอากาศที่ขยายไม่สามารถเล็ดลอดออกมาในช่องระบายได้ ทำให้เกิดความดันสูงมากขึ้นและไปดันแผ่นไดอะแฟรมให้ดันขาตอนที่ตะกัน ทำให้อุปกรณ์ตรวจจับความร้อน นี้ส่งสัญญาณไปยังผู้ควบคุม

- **อุปกรณ์ตรวจจับความร้อนชนิดจับอุณหภูมิคงที่ (Fixed Temperature Heat Detector)**  
 อุปกรณ์ชนิดนี้จะทำงาน เมื่ออุณหภูมิของ Sensors สูงถึงจุดที่กำหนดไว้ซึ่งมีตั้งแต่ 60 องศาเซลเซียสไปจนถึง 150 องศาเซลเซียส การทำงานอาศัยหลักการของโลหะสองชนิด เมื่อถูกความร้อนแล้วมีสัมประสิทธิ์การขยายตัวแตกต่างกัน เมื่อนำโลหะทั้งสองมาแนบติดกัน (Bimetal) และให้ความร้อนจะเกิดการขยายตัวที่แตกต่างกัน ทำให้เกิดบิด โค้งงอไปอีกด้านหนึ่ง เมื่ออุณหภูมิลดลงก็จะคืนสู่สภาพเดิม

- **อุปกรณ์ตรวจจับความร้อนชนิดรวม (Combination Heat Detector)** อุปกรณ์ชนิดนี้รวมเอาคุณสมบัติของ Rate of Rise Heat และ Fixed Temp เข้ามาอยู่ในตัวเดียวกันเพื่อตรวจจับความร้อนที่เกิดได้ทั้งสองลักษณะ

### 2.18.3 อุปกรณ์ตรวจจับเปลวไฟ (Flame Detector)

โดยปกติจะนำไปใช้ในบริเวณพื้นที่อันตรายและมีความเสี่ยงในการเกิดเพลิงไหม้สูง (Heat Area) เช่นคลังจ่ายน้ำมัน, โรงงานอุตสาหกรรม, บริเวณเก็บวัสดุที่เมื่อติดไฟจะเกิดควันไม่มาก หรือบริเวณที่ง่ายต่อการระเบิดหรือง่ายต่อการลุกลาม อุปกรณ์ตรวจจับเปลวไฟจะตรวจจับความถี่คลื่นแสงในย่านอุลตราไวโอเล็ต ซึ่งมีความยาวคลื่นอยู่ในช่วง 0.18-0.36 ไมครอนที่แผ่ออกมาจากเปลวไฟเท่านั้นแสงสว่างที่เกิดจากหลอดไฟและแสงอินฟราเรดจะไม่มีผลทำให้เกิด Fault Alarm ได้การพิจารณาเลือกติดตั้งอุปกรณ์ตรวจจับในบริเวณต่างๆ เราจะคำนึงเรื่องความปลอดภัยของชีวิต, ความเสี่ยงต่อการเกิดอัคคีภัยในบริเวณต่างๆ และลักษณะของเพลิงที่จะเกิด เพื่อที่จะติดตั้งอุปกรณ์ตรวจจับที่เหมาะสมสถานที่และไม่สิ้นเปลืองค่าใช้จ่ายมากเกินไป

### 2.19 การออกแบบระบบสัญญาณเตือนอัคคีภัย

ปัจจัยที่ต้องพิจารณาในการออกแบบ

**ความสูงของเพดาน :** มีผลกับจำนวนอุปกรณ์ตรวจจับที่ต้องใช้ต่อพื้นที่ ความร้อนหรือควันที่ลอยขึ้นมาถึงอุปกรณ์ตรวจจับที่ติดตั้งบนเพดานสูงจะต้องมีปริมาณความร้อน หรือควันที่มากกว่าเพดานต่ำเพื่อให้อุปกรณ์ตรวจจับทำงานในเวลาเท่ากัน จึงต้องลดระยะห่างระหว่างตัวตรวจจับ เพื่อให้ระบบเสริมกำลังตรวจจับให้ละเอียดยิ่งขึ้น เราจะพิจารณากำหนดระยะจัดวางตัวตรวจจับที่ติดบนเพดานโดยอ้างอิงจากตารางต่อไปนี้

| ชนิดตัวตรวจจับ                | พื้นที่การตรวจจับ<br>(ตารางเมตร) | ระยะห่างระหว่างอุปกรณ์<br>(เมตร) | ความสูงเพดาน<br>(เมตร) |
|-------------------------------|----------------------------------|----------------------------------|------------------------|
| ตัวจับควัน(smoke detector)    | 150                              | 9                                | 0.4                    |
| ตัวจับควัน(smoke detector)    | 75                               | 4.5                              | 4.0                    |
| ตัวจับร้อน(heat detector)     | 70                               | 6                                | 0.4                    |
| ตัวจับความร้อน(heat detector) | 35                               | 3                                | 4.9                    |

ตารางที่ 2.29 คุณสมบัติของตัวตรวจจับชนิดต่างๆ

**สภาพแวดล้อม :** อุณหภูมิ, ไอน้ำ, ลม, ฝุ่น, สิ่งบดบัง, ประเภทวัสดุที่อยู่บริเวณนั้น ฯลฯ จะมีผลกับการเลือกชนิดของอุปกรณ์ตรวจจับและตำแหน่งการติดตั้ง เช่น ตัวจับควันจะไม่เหมาะกับบริเวณที่มีฝุ่น, ไอน้ำและลม Rate of Rise Heat Detector ไม่เหมาะที่จะติดตั้งในห้องBoiler ถ้าเป็นสารที่ติดไฟแต่ไม่มีควันก็จำเป็นต้องใช้ Flame Detector ดังนั้นเราจะต้องมีพื้นฐานเข้าใจหลักการทำงานของตัวตรวจจับแต่ละชนิด

**ระดับความสำคัญและความเสี่ยง :** เราควรเลือกใช้อุปกรณ์ที่ตรวจจับได้ไวที่สุด เพื่อรับรู้เหตุการณ์ทันทีก่อนที่จะลุกลามใหญ่โต ในบางสถานที่อาจมีปัจจัยเสี่ยงต่ำ เช่น เป็นพื้นที่ที่อยู่ในระยะของสายตาของเจ้าหน้าที่ประจำตลอดเวลาบริเวณที่ไม่มีวัตถุติดไฟหรือติดไฟยาก สำหรับบริเวณที่อาจเสี่ยงต่อการสูญเสียชีวิตเราจะต้องใช้อุปกรณ์ที่แจ้งเหตุได้เร็วที่สุดไว่ก่อน ได้แก่ ตัวจับควัน

**เงินงบประมาณที่ตั้งไว้ :** งบประมาณเป็นข้อจำกัดทำให้ไม่สามารถเลือกอุปกรณ์ตรวจจับ ชนิดที่ดีที่สุด ติดตั้งไว้ทุกจุดในอาคารเพราะราคาสูง จำต้องยอมเลือกชนิดที่มีราคาถูกไปแพ่งดังนี้

1. Fix Temperature Heat Detector --> 2. Rate of Rise Heat Detector --> 3. Combination Heat Detector --> 4. Photo Electric Smoke Detector --> 5. Ionization Smoke Detector -->6. Flame Detector -->7. Beam Smoke Detector

อุปกรณ์ที่รับรู้เหตุได้ไวจะมีราคาแพงกว่าแต่อาจจะไม่เหมาะสมกับบางสถานที่ เราจะต้องพิจารณากับข้ออื่นด้วย

### 2.19.1 การจัดแบ่งโซน

การที่สามารถค้นหาจุดเกิดเหตุได้เร็วเท่าไร นั้นหมายถึงความสามารถในการระงับเหตุก็จะมากขึ้นด้วยดังนั้น การจัดโซนจึงเป็นความสำคัญในการออกแบบระบบ Fire Alarm กรณีเกิดเหตุเริ่มต้นจะทำให้กระดิ่งดังเฉพาะโซนนั้นๆ ถ้าคุณสถานะการณ์ไม่ได้จึงจะสั่งให้กระดิ่ง โซนอื่นๆดังตาม

**แนวทางการแบ่งโซนมีดังนี้**

1. ต้องจัดโซน อย่างน้อย 1 โซนต่อ 1 ชั้น
2. แบ่งตามความเกี่ยวข้องของพื้นที่ ที่เป็นที่เข้าใจสำหรับคนในอาคารนั้น เช่น โซน Office, โซน Workshop
3. ถ้าเป็นพื้นที่ราบบริเวณกว้าง จะแบ่งประมาณ 600 ตารางเมตร ต่อ 1 โซน เพื่อสามารถมองเห็นหรือค้นพบจุดเกิดเหตุโดยเร็ว

4. คนที่อยู่ใน โซนใดๆ ต้องสามารถได้ยินเสียงกระดิ่ง Alarm ในโซนนั้นได้ชัดเจน

### 2.19.2 การออกแบบติดตั้ง Manual Station

ระบบ Fire Alarm จะต้องมีสวิทช์กดฉุกเฉิน(Manual Station)ด้วยอย่างน้อยโซนละ 1 ชุด สำหรับกรณีที่คนพบเหตุการณ์ก่อนที่ Detector จะทำงานหรือไม่มี Detector ติดตั้งไว้ในบริเวณนั้น

**Manual Station จะต้องมีลักษณะดังนี้**

1. เป็นการง่ายต่อการสังเกต โดยใช้สีแดงเข้ม ดูเด่นหรือมีหลอดไฟ(Location Light) ติดแสดงตำแหน่งในที่มืดหรือยามค่ำคืน
2. ตำแหน่งที่ติดตั้ง ต้องอยู่บริเวณทางออก ทางหนีไฟ ที่สามารถมองเห็นได้ชัดเจน
3. ระดับติดตั้งง่ายกับการกดแจ้งเหตุ (สูงจากพื้น 1.1-1.5 เมตร)
4. กรณีระบบมากกว่า 5 โซน ควรมีแจ้งโทรศัพท์เพื่อใช้ติดต่อระหว่างเจ้าหน้าที่บริเวณที่เกิดเหตุกับห้องควบคุมของอาคาร เพื่อรายงานสถานะการณ์และสั่งให้เปิดสวิทช์ General Alarm ให้กระดิ่งดังทุกโซน

### 2.19.3 การกำหนดตำแหน่งอุปกรณ์แจ้งสัญญาณ

อุปกรณ์แจ้งสัญญาณมีหลายชนิด ได้แก่ กระดิ่ง ไซเรน ไฟสัญญาณกระพริบ โดยทั่วไปเราจะนิยมติดตั้งกระดิ่งไว้บริเวณใกล้เตียงหรือที่เดียวกับ Manual Station ในระดับหูหรือเหนือศีรษะ เราจะมีกระดิ่งอย่างน้อย 1 ตัว ต่อโซนหรือเพียงพอ เพื่อให้คนที่อยู่เขตพื้นที่โซนนั้นได้ยินเสียงชัดเจนทุกคน (รัศมีความดังระดับที่พอเพียงของกระดิ่งขนาด 6 นิ้วจะไม่เกิน 25 เมตร) ส่วนไซเรนเราจะติดตั้งไว้ได้ชยาคาด้านนอก เพื่อแจ้งเหตุให้บุคคลที่อยู่นอกอาคารได้รับทราบว่าเกิดเหตุผิดปกติ โดยเราจะกำหนดให้ไซเรนดังทันทีทุกครั้งที่เกิดเหตุก่อนจากนั้นจึงจะรอการตัดสินใจว่าจะให้โซนอื่นๆดังตามหรือไม่

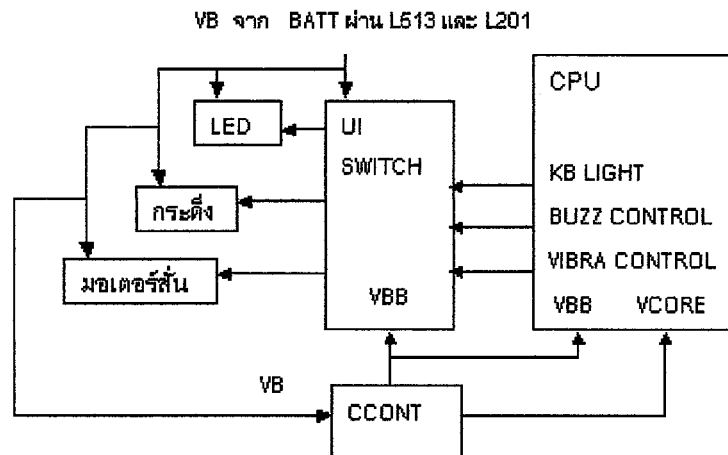
### 2.19.4 ตำแหน่งการติดตั้งตู้ควบคุม (Fire Alarm Control Panel)

เราจะติดตั้งตู้ควบคุม (FCP) ไว้บริเวณที่มีเจ้าหน้าที่รักษาความปลอดภัย หรือช่างควบคุมระบบอาคารหรือห้องเจ้าหน้าที่รักษาความปลอดภัยจากที่กล่าวมาข้างต้น จะเห็นได้ว่าระบบสัญญาณเตือนอัคคีภัยเป็นสิ่งจำเป็นที่ผู้ใช้ตระหนักถึงความปลอดภัยจะต้องคำนึงถึงและเลือกใช้ให้เหมาะสม

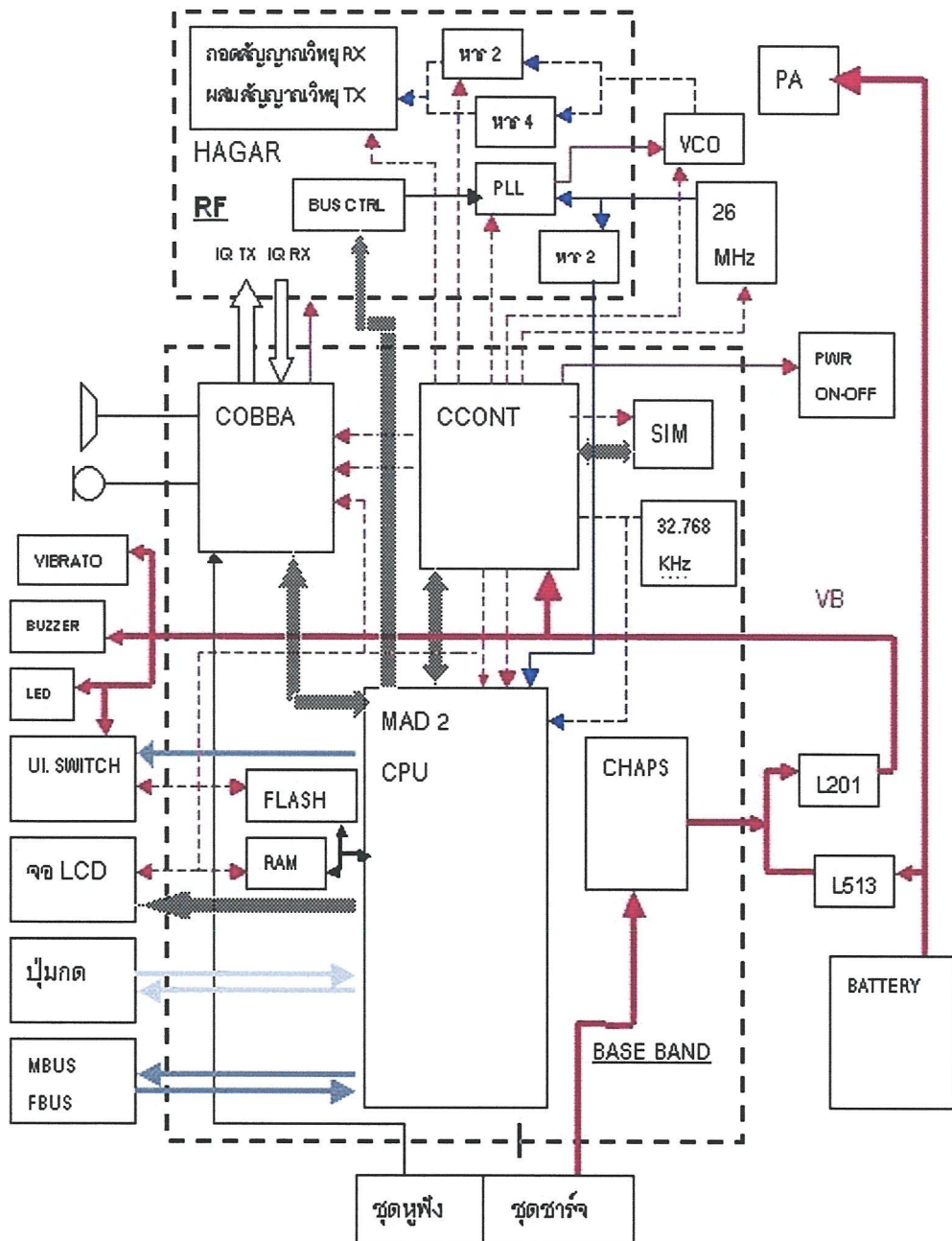
## 2.20 โครงสร้างของโทรศัพท์ NOKIA 3310

### 2.20.1 UI หรือ USER INTERFACE

- UI SWITCH (N400)
- หลอดไฟ LED ทำหน้าที่ให้แสงสว่างแก่ปุ่มกดและจอ LCD
- มอเตอร์สั่น VIBRATOR
- กระดิ่ง BUZZER



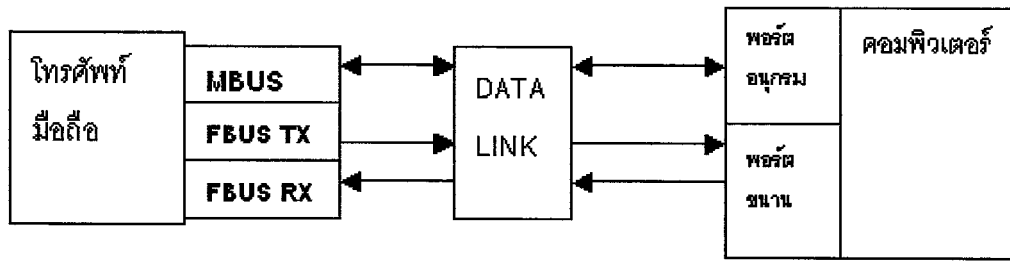
รูปที่ 2.24 ส่วนติดต่อผู้ใช้งาน



รูปที่ 2.25 โครงสร้างของโทรศัพท์ NOKIA 3310

### 2.20.2 แฟลชคอนเนคเตอร์ FLASH CONNECTOR

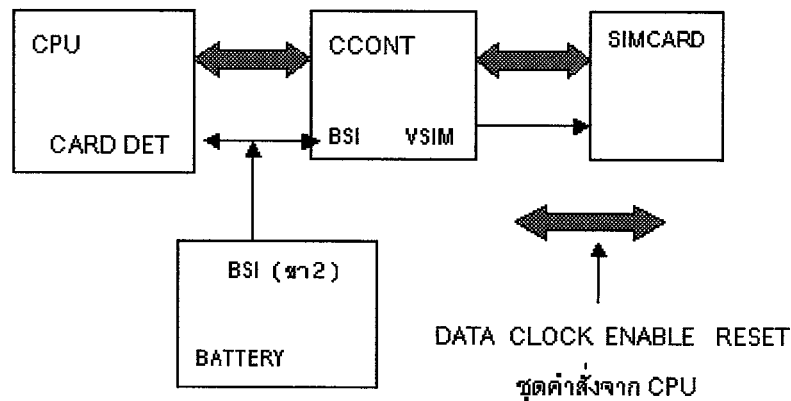
- MBUS ชูคเชื่อมต่อโยง หรือถ่ายโอนข้อมูลระหว่างมือถือกับคอมฯ
- FBUS TX ชูคเชื่อมต่อโยง หรือถ่ายโอนข้อมูลจากมือถือเข้าคอมฯ
- FBUS RX ชูคเชื่อมต่อโยง หรือถ่ายโอนข้อมูลจากคอมฯเข้ามือถือ



รูปที่ 2.26 การเชื่อมต่อข้อมูลระหว่างคอมพิวเตอร์และมือถือ

### 2.20.3 SIM CARD

SUBSCRIBER IDENTITY MODULE เป็นส่วนหนึ่งของอุปกรณ์โทรศัพท์ภายในเป็น CHIP IC MEMORY ขนาดเล็ก แต่เนื่องจากหมายเลขโทรศัพท์เป็นเลขส่วนบุคคลจึงต้องนำ SIM ไปทำการจดทะเบียนหรือ REGISTER ก่อนแล้วจึงนำ SIM ไปใส่ในโทรศัพท์ซึ่ง SIM จะติดต่อกับ CCONT โดยตรง และ CCONT จะเชื่อมโยงหรือ INTERFACE ข้อมูลไปยัง CPU และ SIM จะทำงานได้ก็ต่อเมื่อมีไฟเลี้ยง SIM จาก CCONT ชื่อ VSIM โดยทั่วไป SIM มี 2 ชนิด คือ SIM แผ่นใหญ่ ( รุ่นเก่า ) ขนาดเท่าบัตร ATM ใช้ไฟเลี้ยง 5 V. และ SIM ที่ใช้ในปัจจุบันคือ SIM ขนาดเล็ก หรือ PLUG IN SIM ใช้ไฟเลี้ยง 3V. นอกจากนี้ SIM ถูกควบคุมการทำงานโดยขา 2 ของแบตเตอรี่ ( BSI ) ด้วย ถ้าขณะที่ใช้งานขา 2 ของแบตเตอรี่หลุดหรือเคลื่อนจากตำแหน่ง หน้าจอ LCD จะแสดงคำว่า “ INSERT SIM CARD ” ทันที



รูปที่ 2.27 การเชื่อมต่อ Sim Card

### 2.20.4 ภาคแปลงสัญญาณ (COBBA)

เป็น ASIC IC หรือ APPLICATION SPECIFIC INTEGRATED CIRCUIT ทำหน้าที่ เชื่อมโยงหรือ INTERFACE ระหว่าง ภาคเบสแบนด์กับภาควิทยุ แปลงสัญญาณเสียงหรือ AUDIO FREQUENCY (AF) ซึ่งมาจาก ไมโครโฟนผ่านกระบวนการกรองความถี่ต่ำแล้วแปลงเป็นสัญญาณดิจิทัล A/D เข้าสู่กระบวนการแปลงสัญญาณเป็น PCM หรือ PULSE CODE MODULATOR โดย การควบคุมของวงจร

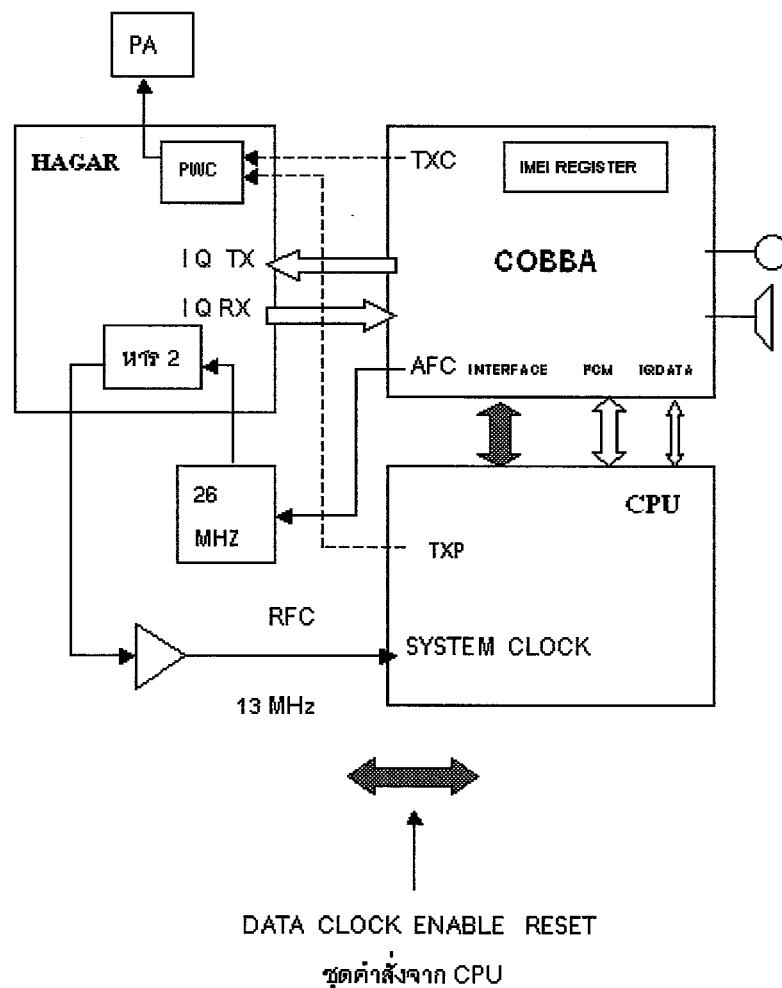
DSP หรือ DIGITAL SIGNAL PROCESSING จาก CPU แล้วแปลงให้เป็นสัญญาณ IQ หรือ INPHASE QUADRATURE ส่งต่อให้ HAGAR เราเรียกภาคนี้ว่าภาคส่งหรือ TX

ทำหน้าที่รับสัญญาณ IQ หรือ INPHASE QUADRATURE จาก HAGAR แปลงกลับมาเป็น PCM หรือ PULSE CODE MODULATOR ซึ่งเป็นสัญญาณดิจิทัล D/A ผ่านกระบวนการกรองความถี่ต่ำผ่านวงจรมอดูเลเตอร์ ลำโพงหรือหูฟัง เราเรียกภาคนี้ว่า ภาครับ หรือ RX

ควบคุมหรือชดเชยความถี่อัตโนมัติ หรือ AFC (AUTO FREQUENCY CONTROL) สำหรับ 26 MHZ

ควบคุมวงจรปรับระดับกำลังส่ง (PWC) ใน HAGAR เพื่อให้ HAGAR จ่ายแรงดันไฟไปควบคุม PA หรือ POWER AMP ในระดับที่ต่างกัน สำหรับภาคส่ง หรือ TXC

ทำหน้าที่จัดเก็บเลข IMEI หรือ IMEI REGISTER



รูปที่ 2.28 ส่วนเชื่อมต่อระหว่างภาคเบสแบนด์กับภาควิทยุ

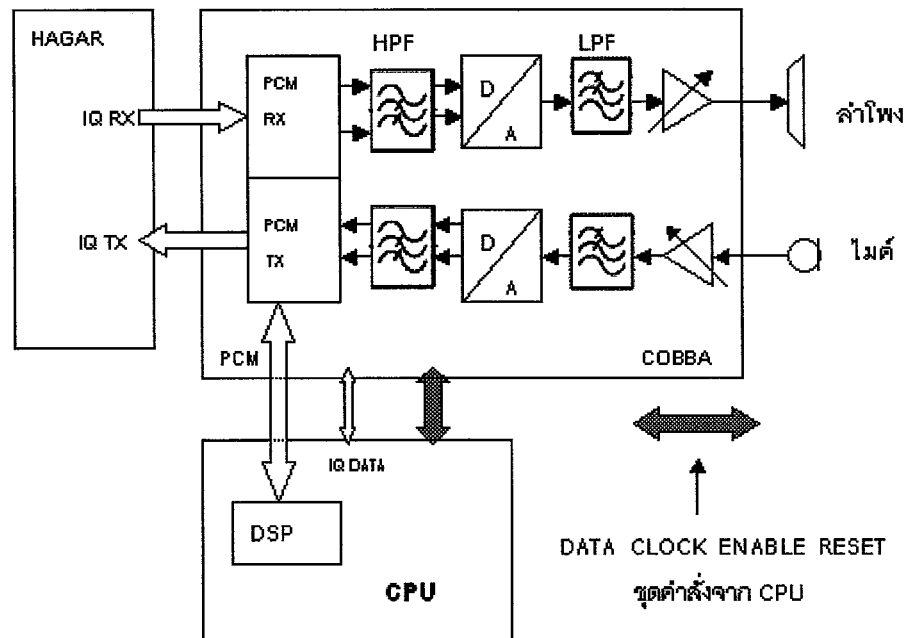
## 2.20.5 อุปกรณ์เชื่อมต่อกับ COBBA

### ไมโครโฟน MICROPHONE

ทำหน้าที่แปลงความถี่เสียงให้เป็นสัญญาณไฟฟ้าหรือ AF (AUDIO FREQUENCY)

### หูฟังหรือลำโพง EARPIECE , SPEAKER

ทำหน้าที่แปลงสัญญาณไฟฟ้าให้เป็นความถี่เสียง หรือ AF โดยผ่านวงจรขยายเสียงหรือ AMPLIFIER



รูปที่ 2.29 การเชื่อมต่อไมโครโฟนและลำโพง

## 2.21 ภาควิทยุ RF ( RADIO FREQUENCY )

ประกอบด้วยอุปกรณ์หลักดังนี้

### 2.21.1 ภาครับ-ส่งความถี่ (HAGAR)

เป็น IC โปรเซสเซอร์ ซึ่งรวมเอาภาครับและภาคส่งและภาคสังเคราะห์ความถี่ หรือภาคผลิตความถี่ท้องถิ่นเข้าด้วยกันดังมีรายละเอียดดังนี้

ภาครับ RX (RECEIVER) ทำหน้าที่ รับสัญญาณวิทยุหรือ RF (RADIO FREQUENCY) ซึ่งมาจากเสาอากาศผ่าน ฟิลเตอร์ 1 ผ่านวงจรขยายสัญญาณรบกวนต่ำ หรือ LNA ผ่านฟิลเตอร์ 2 ผ่านบาลัน ซึ่งภาคต่างๆ ที่กล่าวมานี้เรียกว่า ฟรอนท์เอนด์ (FRONT END) จากนั้นก็จะทำการถอดสัญญาณวิทยุ หรือ หักล้างสัญญาณวิทยุ (DEMODULATOR) แปลงให้เป็นสัญญาณ IQ (INPHASE หรือ สัญญาณเฟสรวม) และ (QUADRATURE หรือ สัญญาณต่างเฟส 90°) จากนั้นก็จะทำการปรับแต่งรูปสัญญาณ IQ ให้เหมาะสมส่งต่อไปยัง COBBA เพื่อให้ COBBA แปลงสัญญาณ IQ ให้เป็นสัญญาณเสียงออกลำโพงหรือหูฟังต่อไป

ภาคส่ง TX (TRANSMITTER) เริ่มจากไมโครโฟนที่แปลงสัญญาณเสียงหรือ AF (AUDIO FREQUENCY) ผ่านกระบวนการดิจิตอลแปลงให้เป็นสัญญาณ IQ (INPHASE QUADRATURE) ซึ่งมาจาก COBBA และส่งต่อมายัง HAGAR จากนั้น HAGAR ทำการผสมสัญญาณ IQ กับความถี่วิทยุ หรือ RF (RADIO FREQUENCY) การผสม IQ + RF นี้เรียกว่า MODULATOR แล้วส่งสัญญาณนี้ออกจาก HAGAR ผ่าน BALUN ไปยังวงจรรขยาย ( สำหรับระบบ GSM) แล้วไปขยายสัญญาณอีกครั้งที่ PA (POWER AMPLIFIER) สำหรับระบบ 1800 สัญญาณจะออกจาก HAGAR เข้า PA โดยตรงเลยแต่เนื่องจากโทรศัพท์ แต่ละเครื่องอยู่ห่างเสารับสัญญาณไม่เท่ากัน จึงจำเป็นต้องเพิ่มหรือลดระดับของกำลังส่งให้พอเหมาะสมกับระยะห่างของเสา โดยการตรวจสอบ (DETECT) ความแรงของสัญญาณจากภาครับ (RX FILTER CALIBRATION) เพื่อให้ภาครับและส่งสัญญาณสมดุลกันเพราะโทรศัพท์รับและส่งสัญญาณไปยัง เครือข่าย (CELL SITE) เดียวกัน เพราะฉะนั้น PA (POWER AMP) ซึ่งเป็นวงจรรขยายกำลังส่ง จึงจำเป็นต้องมีแรงดันไฟไปเลี้ยงในระดับที่ต่างกัน เพื่อให้พอเหมาะกับระยะห่างของเครือข่าย (CELL SITE) ซึ่งไฟเลี้ยง PA (VPAC) มาจากวงจรควบคุมกำลังส่ง (PWC) ใน HAGAR โดยมี TXC จาก COBBA ทำหน้าที่เพิ่มหรือลดระดับกำลังส่ง และมี TXP จาก CPU ทำหน้าที่เปิดวงจรควบคุมกำลังส่งให้กับวงจร PWC ใน HAGAR กำลังส่งที่ส่งออกจาก PA จะส่งรูปแบบของสนามแม่เหล็กส่งผ่านไปยัง สวิตช์แอนเทนน่า หรือ DIPLEXER ออกเสาอากาศของโทรศัพท์มือถือ ส่งต่อไปยังเครือข่าย (CELL SITE) ต่อไป

ภาคสังเคราะห์ความถี่ (SYNTHESIZER) หรือภาคผลิตความถี่ท้องถิ่น (LOCAL OSCILLATOR) คือภาคที่ผลิตความถี่ LO เพื่อนำความถี่ที่ผลิตหรือสังเคราะห์ได้เข้าไปหาร 2 สำหรับ (ระบบ 1800) และหาร 4 สำหรับ (ระบบ GSM) ใน HAGAR สำหรับภาครับความถี่ที่ได้ จะนำไปหักล้างหรือถอดสัญญาณวิทยุกับความถี่ที่มาจากเครือข่ายให้เป็นสัญญาณ IQ เราเรียกภาคนี้ว่า DEMODULATOR สำหรับภาคส่ง ความถี่ที่ได้จะนำไปผสมหรือรวมกับสัญญาณ IQ เราเรียกภาคนี้ว่า MODULATOR หัวใจหลักของการผลิตความถี่คือ วงจร PLL (PHASE LOCKED LOOP) หรือ เฟส ล็อก ลูป เป็นระบบป้อนกลับที่บังคับให้วงจรผลิตความถี่ OSC (OSCILLATOR) มีความถี่หรือเฟสของความถี่เปลี่ยนแปลงไปตามความถี่หรือเฟสของสัญญาณอ้างอิงภายนอก เฟส ล็อก ลูป (PHASE DETECTOR)

เนื่องจากโทรศัพท์มีการรับส่งสัญญาณในระดับความถี่และช่องสัญญาณที่ต่างกัน ดังนั้นการเข้าช่องสัญญาณหรือกำหนดช่องสัญญาณต้องแม่นยำ จึงจำเป็นต้องใช้ CPU เข้ามากำหนดช่องสัญญาณในภาค BUS CONTROL และที่สำคัญมากก็คือความถี่อ้างอิง 26 MHz ต้องมีความเที่ยงตรงมากด้วยเช่นกัน ใน IC HAGAR วงจร เฟส ล็อก ลูป (PLL) เป็นแบบปริสเกลเลอร์ 2 DIVIDER หรือ 2 โมดูลัส ระหว่าง P กับ P+1 ซึ่งวงจรมันจะถูกรักษาการหาร โดย CPU ปริสเกลเลอร์ตัวนี้สามารถหารด้วยตัวเลข 2 ตัว ซึ่งต่างกันอยู่ 1 เช่นหาร 10 หรือ 11 เรียกว่า 10/11 แต่สำหรับ HAGAR ใช้ 64/65 สังเกตว่าตัวหารทั้งคู่ ต่างกันอยู่เอาต์พุตของปริสเกลเลอร์จะป้อนไปให้แก่วงจรนับ หรือวงจรเคาน์เตอร์บ้างครั้งก็เรียกว่าวงจรหารตัวหนึ่งเป็นเคาน์เตอร์หลัก (MAIN COUNTER) ตัวย่อคือ M ส่วนอีกตัวเป็นเคาน์เตอร์เสริม (AUXILARY COUNTER) ตัวย่อคือ A ตัวเคาน์เตอร์เสริมจะเป็นตัวบังคับให้ ปริสเกลเลอร์หารด้วย

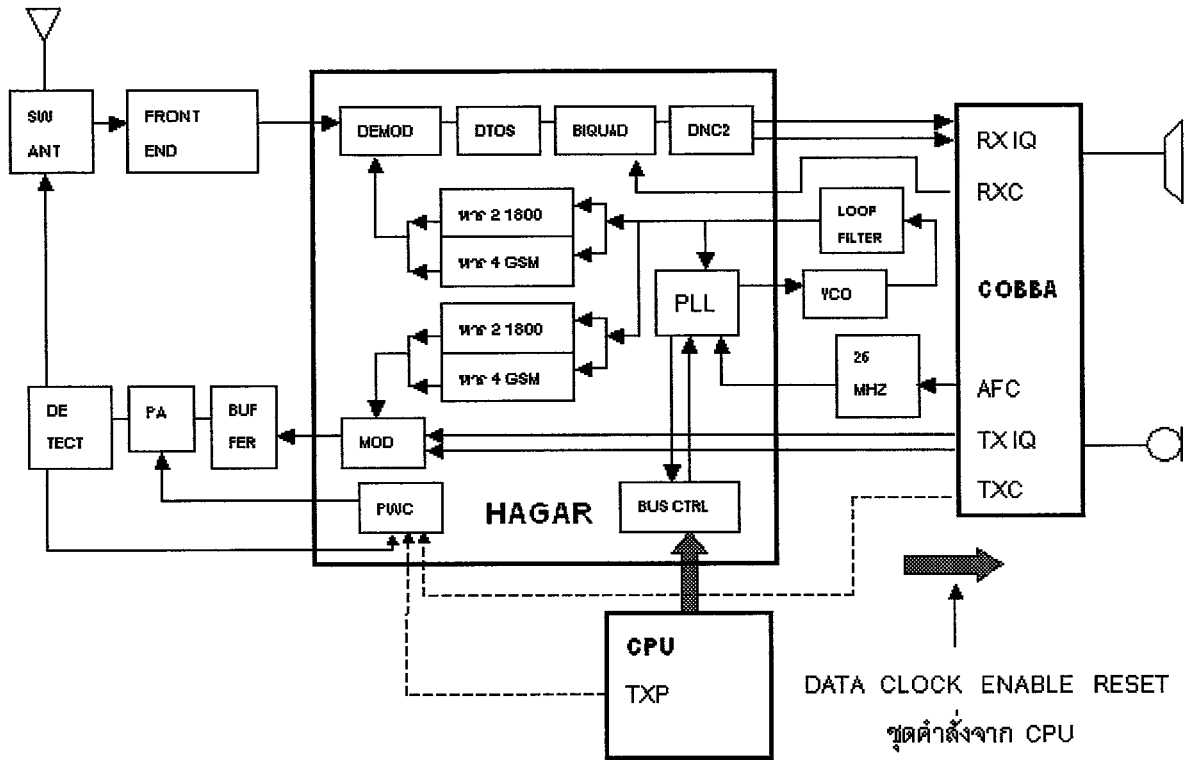
ตัวหาร (DIVIDER) คือ 64 หรือ 65 เช่นสมมุติว่า CPU ป้อนข้อมูล ( ความถี่ ) หรือปริสเกลเลอร์ใช้ 65 เป็นตัวหาร เมื่อเคาน์เตอร์เสริมหยุดนับ จึงจะส่งคำสั่ง ไปบังคับให้ปริสเกลเลอร์เปลี่ยนเป็นหารด้วย 64 ตัว เคาน์เตอร์หลัก (A) ก็เช่นเดียวกันจะค่อยๆ นับถอยหลังไปเรื่อยๆ จนเป็น 0 เมื่อเคาน์เตอร์หลักและเสริม (M) และ (A) นับถึงศูนย์เมื่อใดทั้งคู่จะถูก CPU เซตตัวเลขข้อมูล ( ความถี่ ) เนื่องจาก เคาน์เตอร์เสริม จะต้องนับถึง 0 ก่อน ดังนั้นตัวเลขที่ CPU เซตให้เคาน์เตอร์เสริม (A) จะต้องน้อยกว่าตัวเลขที่ CPU เซตให้ เคาน์เตอร์หลัก (M)ตัวเลขที่เซตให้แก่เคาน์เตอร์หลัก (M) และเคาน์เตอร์เสริม (A) เริ่มแรกนั้นให้ปริสเกล เลอร์อยู่ในภาวะหาร 65 ไปจนกว่าเคาน์เตอร์เสริม (A) จะนับลงเป็น 0 นั่นคือเวลาที่ใช้ในการนับ เคาน์เตอร์เสริม ที่เป็น 0 โดยคิดจากจำนวนรอบหรือไซเคิลของ VCO ที่ผ่านไปเท่ากับจำนวน 65 คูณด้วย A ไซเคิลหลังจากนั้นปริสเกลเลอร์จะถูกบังคับให้เปลี่ยนตัวหารเป็น 64 โดยเคาน์เตอร์เสริม (A) และ ในขณะที่เคาน์เตอร์หลักนับผ่านเคาน์เตอร์เสริม (A) ไปแล้ว พร้อมกับกับเคาน์เตอร์เสริม เช่นกัน ยัง เหลืออยู่อีก (M-A) ไซเคิลก่อนที่นับเป็นศูนย์นั้นคือจะต้องใช้เวลาในการนับเคาน์เตอร์หลัก (M) ให้เป็น ศูนย์ต่อไปอีกคิดเป็นจำนวนไซเคิล หรือจำนวนรอบของ VCO ที่ผ่านไปเท่ากับ 64 คูณด้วย (M-A) ฉะนั้น รอบเวลาที่ใช้จึงจำเป็นผลรอบของเวลาทั้ง 2 ข้างคั่น คือ

$$\text{VCO ไซเคิล} = 65A + 64(M-A) = 64M + A$$

ความถี่ของ VCO จะเท่ากับ  $(64M+A)$  เท่าความถี่อ้างอิงหรือ

$$F_{\text{synth}} = F_{\text{ref}}(64M+A)$$

สมการที่ยกตัวอย่างมาข้างต้นใช้กับ ปริสเกลเลอร์ 64/65 ซึ่งเป็นชนิด 2 โมดูลัสหรือ DUAL MODULUS DIVIDER และความถี่ที่ได้จะเข้าไปวงจรเปรียบเทียบ หรือ PHASE DETECT ซึ่งต่อเชื่อม กับวงจรชาร์จ์ปั๊ม CHARGE PUMP ซึ่งจะทำการเก็บและคายประจุแรงดันไฟในวงจรลูปฟิลเตอร์ (LOOP FILTER) แล้วจ่ายให้กับ VCO หลังจากนั้น VCO ก็จะผลิตความถี่ 3420-3840 MHz จ่ายผ่านบาลันเข้าไป ยังวงจรหาร 2 หรือหาร 4 เพื่อลดสัญญาณวิทยุ ในภาครับและผสมสัญญาณวิทยุในภาคส่งต่อไป ซึ่งวงจร ทั้งหมดที่กล่าวมานี้จะทำงานได้ก็ต้องมีแรงไฟมาเลี้ยงวงจร และไฟเลี้ยงวงจรทั้งหมดไม่ว่าจะเป็นวงจร ปริสเกลเลอร์ 64/65 วงจรเปรียบเทียบเฟส วงจรเคาน์เตอร์หรือวงจรมับวงจรหาร 2 และ 4 วงจร ชาร์จปั๊ม และ VCO มาจาก CCONT ทั้งหมดและคำสั่งที่กำหนดข้อมูล ( ความถี่ ) ให้กับวงจรเคาน์เตอร์หรือวงจร หาร มาจาก CPU ผ่าน BUS CONTROL ประกอบด้วย DATA CLOCK ENABLE และ RESET



รูปที่ 2.30 บล็อกไดอะแกรม ภาควิทยุ RF ( RADIO FREQUENCY )

### บทที่ 3

#### การคำนวณและการสร้าง

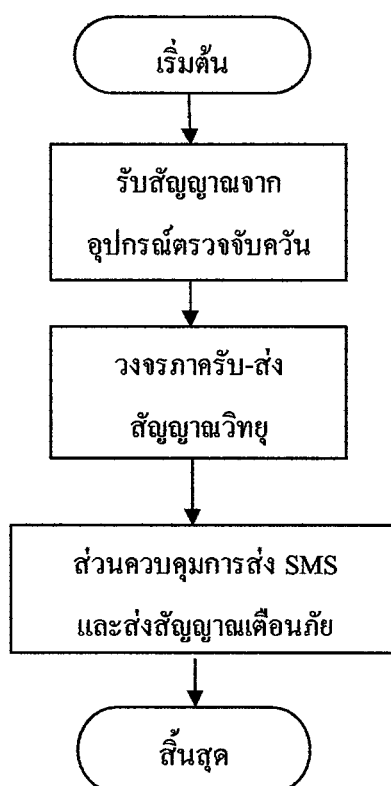
##### 3.1 การทำงานของระบบ

ระบบแจ้งเหตุเพลิงไหม้ โดยแจ้งเตือนผ่านข้อความสั้น แบ่งการทำงานออกเป็นสองส่วน คือ

1. ส่วนของวงจรรับ – ส่ง สัญญาณวิทยุ ทำหน้าที่ในการส่งสัญญาณวิทยุ จากการตรวจจับควันของสโมคดีเทคเตอร์ (Smoke Detector) เมื่อเกิดเหตุเพลิงไหม้ ส่งไปยังส่วนของวงจรส่วนควบคุม

2. วงจรส่วนควบคุม จะรับสัญญาณวิทยุจากวงจรภาครับสัญญาณแล้ว จะส่งเข้าไปที่ไมโครคอนโทรลเลอร์ AVR เพื่อทำหน้าที่ให้การแจ้งเตือนทั้งผู้ที่อยู่ในบริเวณใกล้เคียง โดยการส่งให้ไซเรนดัง และแจ้งเตือนผ่านทาง SMS เพื่อแจ้งในกรณีที่ไม่มีผู้ใดอยู่ในบริเวณนั้น พร้อมๆกัน

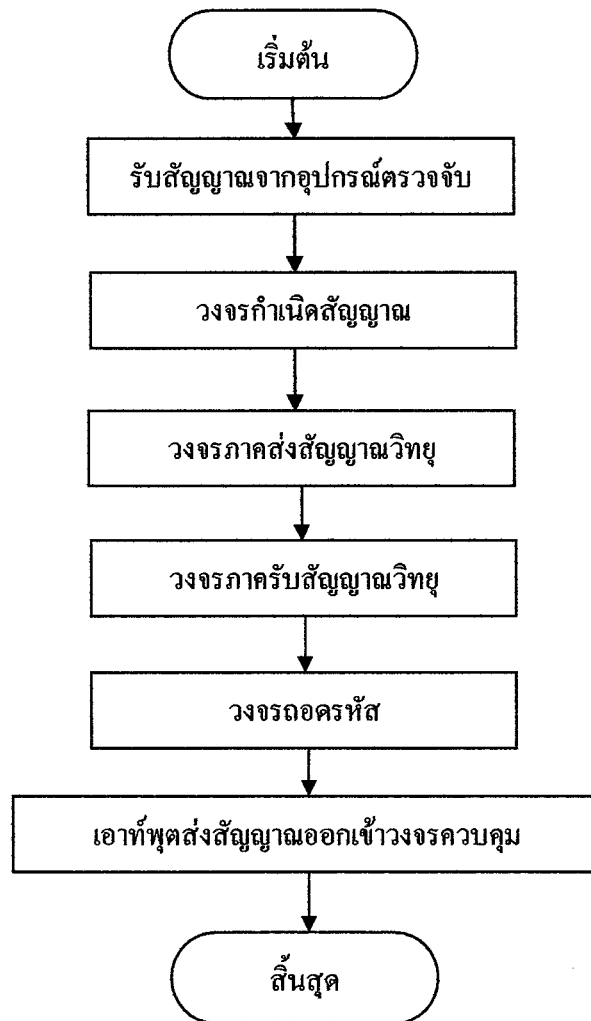
ข้อดีของการรวมการทำงานของทั้ง 2 ระบบเข้าด้วยกันคือ สามารถกระจายพื้นที่การติดตั้งของสโมคดีเทคเตอร์ (Smoke Detector) ในการตรวจจับควันได้ในบริเวณกว้างและติดตั้งได้ง่ายเพราะไม่ต้องเดินสายไฟแต่ใช้การส่งสัญญาณวิทยุแทน ซึ่งหากพิจารณาแล้วจะมีความสะดวกและปลอดภัยจากการลัดวงจรเนื่องจาก อายุการใช้งานของสายไฟ และสภาพแวดล้อมอื่นๆ โดยมีหลักการทำงานดังบล็อกไดอะแกรมที่ แสดงในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ

### 3.2 วงจรภาครับ – ส่งสัญญาณวิทยุ

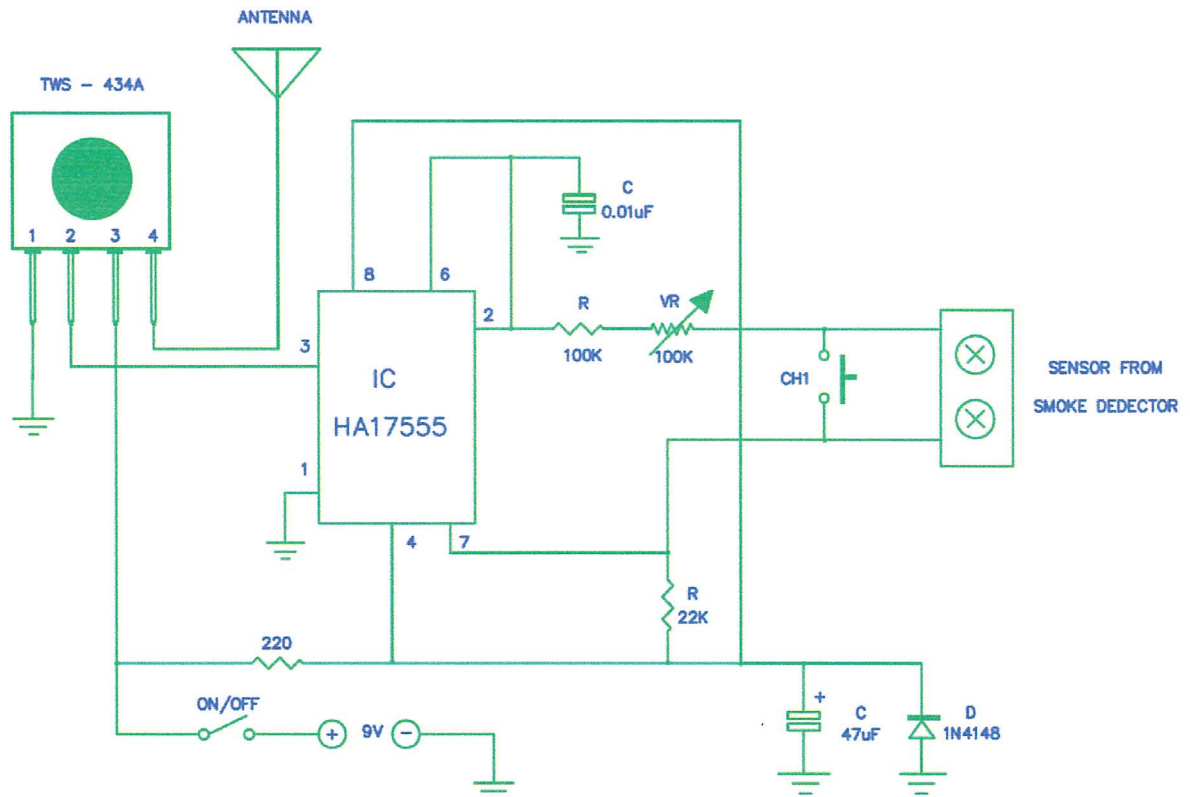
วงจรภาครับ-ส่งสัญญาณวิทยุ วงจรในส่วนนี้จะทำหน้าที่ รับสัญญาณจากอุปกรณ์ตรวจจับควัน (Smoke Detector) และส่งมายังภาครับ โดยมีหลักการการทำงานดังบล็อกไดอะแกรมที่ แสดงในรูปที่ 3.2



รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของ วงจรภาครับ-ส่งสัญญาณวิทยุ

### 3.2.1 หลักการทำงานของภาคส่ง

#### รูปวงจรรวมของภาคส่งสัญญาณวิทยุ

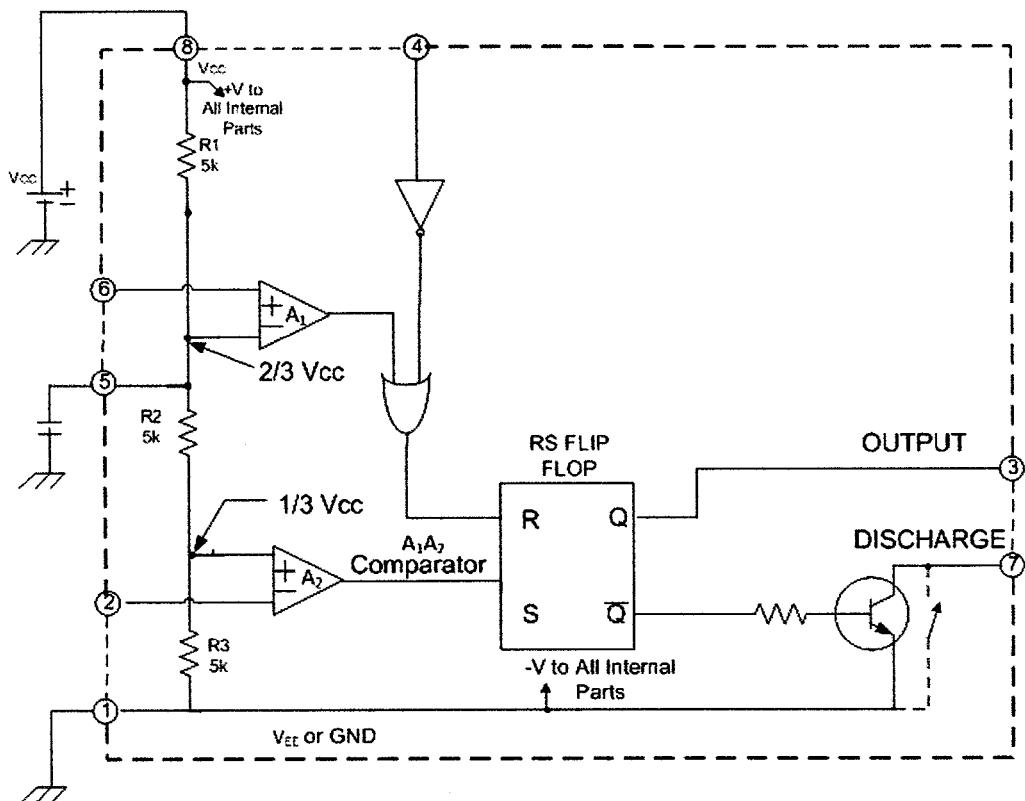


รูปที่ 3.3 วงจรรวมของภาคส่งสัญญาณวิทยุ

การทำงานของวงจรเริ่มจากการที่ วงจรภาคส่งสัญญาณวิทยุแต่ละช่องสัญญาณ ได้รับอินพุตจากอุปกรณ์ตรวจจับควัน โดยสัญญาณจะส่งเข้ามาจะทริกซ์ให้สวิดซ์ทำงาน เกิดการไหลเวียนของกระแสในวงจรสมบรูณ์ โดยกระแสจะผ่าน VR ซึ่งเป็นตัวต้านทานปรับค่าได้ โดยวงจรภาคส่งสัญญาณวิทยุแต่ละช่องสัญญาณจะมีค่าความแตกต่างของความต้านทาน จากการตั้งค่าทางด้านภาคส่งและภาครับ โดยการปรับความถี่ของสัญญาณให้ตรงกัน

เมื่อกระแสผ่านที่ตัวต้านทานปรับค่าได้ (VR) แล้วจะผ่านตัวต้านทาน (R) ขนาด 100K โอห์ม ( สำหรับช่องสัญญาณที่ 1, 2 ) และ 200K โอห์ม ( สำหรับช่องสัญญาณที่ 3 ) โดยทั้งตัวต้านทานแบบธรรมดาและแบบปรับค่าได้จะเป็นตัวที่ปรับระดับแรงดันกระแสให้แตกต่างกันในแต่ละช่องสัญญาณ

### 3.2.2 หลักการออกแบบวงจรแสงสัญญาณวิทยุ



รูปที่ 3.4 บล็อกไดอะแกรมภายในของ IC HA 17555

การทำงานของ IC HA17555 จะเป็นตัวกำเนิดสัญญาณความถี่ องค์ประกอบหลักของ IC HA17555 ประกอบด้วย วงจรเปรียบเทียบแรงดันไฟฟ้า ( Voltage Comparator ) 2 ชุด: A1, A2 และ RS ฟลิปฟลอป ดังแสดงในรูป 3.4 การทำงานของวงจรเปรียบเทียบแรงดัน สามารถอธิบายการทำงานดังสมการ

$$V_o \text{ Comparator} = \begin{cases} \text{Logic 1 หรือ High เมื่อ } V_{in}^+ > V_{in}^- \\ \text{Logic 0 หรือ Low เมื่อ } V_{in}^+ < V_{in}^- \end{cases}$$

ขา 1 :  $V_{EE}$  หรือ GND เป็นขาสำหรับจ่ายไฟขั้วลบให้กับวงจรภายในทั้งหมด โดยทั่วไปมักใช้เป็นกราวด์

ขา 8 :  $V_{CC}$  เป็นขาสำหรับจ่ายไฟขั้วบวกให้กับวงจรภายในทั้งหมด ค่าของแหล่งจ่ายไฟที่จ่ายให้กับ IC HA17555 สามารถใช้ได้ตั้งแต่ 5 V ถึง 15 V

$$5V \leq V_{pin 8} - V_{pin 1} = V_{CC} - V_{EE} \leq 15V$$

ขา 3 : ทำหน้าที่เป็นเอาต์พุต ซึ่งจะมีสถานะเป็น โลกิก “ 0 ” หรือ “ 1 ” จะมีค่าความต่างศักย์เทียบกับกราวด์ดังต่อไปนี้

Output “ 1 ” หรือ High ;  $V_{pin\ 3} = V_{CC} - 1.5\ V$

Output “ 0 ” หรือ Low ;  $V_{pin\ 3} = V_{EE} + 0.25\ V$

ขา 3 หรือ เอาต์พุตของ IC HA17555 สามารถนำไปจ่ายกระแสได้สูงสุดไม่เกิน  $\pm 200\ mA$

ขา 7 : **DISCHARGE** ภายใน IC HA17555 ระหว่างขา 7 กับ  $V_{EE}$  หรือ กราวด์ จะมีทรานซิสเตอร์ต่ออยู่ดังในรูปที่ 3.4 ทรานซิสเตอร์นี้จะทำหน้าที่เป็นสวิตช์ ซึ่งมักจะนำไปใช้ในการคายประจุของ C ในวงจรภายนอก ในขณะที่เอาต์พุต (ขา 3) เป็นสถานะ “ 0 ”  $\overline{Q}$  ของฟลิปฟล็อปจะเป็น “ 1 ” ทรานซิสเตอร์ Q ที่ต่ออยู่ระหว่างขา 7 กับ  $V_{EE}$  จะ “ ON ” เหมือนสวิตช์ปิดวงจร หรือเสมือนขา 7 ถูกช็อตกับ  $V_{EE}$  หรือกราวด์ ในทางตรงข้ามในขณะที่เอาต์พุต (Q ฟลิปฟล็อปเป็น “ 1 ”  $\overline{Q}$  จะเป็น “ 0 ” ทรานซิสเตอร์ Q จะ OFF หรือเปิดวงจร หรือกล่าวโดยสรุปได้ว่า

Output (ขา3) = 0 ทำให้ ขา 7 จะช็อตกับ  $V_{EE}$  โดยผ่าน Q

Output (ขา3) = 1 ทำให้ ขา 7 เสมือนเปิดวงจร

เอาต์พุตของ IC HA17555 (ขา 3) จะมีสถานะเป็น “ 0 ” หรือ “ 1 ” จะขึ้นอยู่กับศักย์ไฟฟ้าที่อินพุตขา 2 ขา 6 และ ขา 4

ขา 4 : **RESET** จะทำหน้าที่รีเซ็ตฟลิปฟล็อปโดยตรง

ถ้าขา 4 เป็นสถานะ “ 0 ” หรือ  $V_{pin4} \leq 0.7$  เอาต์พุต (ขา 3) จะมีสถานะเป็น 0 หรือถูกรีเซ็ต และขา 7 จะเสมือนถูกช็อตกับ  $V_{EE}$  หรือกราวด์ โดยไม่สนใจว่าขา 2 หรือ ขา 6 จะมีศักย์ค่าไฟฟ้าเป็นอย่างไร

ถ้าขา 4 เป็นสถานะ “ 1 ” หรือ  $V_{pin4} \geq 2$  เอาต์พุต (ขา 3) และทรานซิสเตอร์ที่ขา 7 จะถูกควบคุมโดยศักย์ค่าไฟฟ้าที่เข้ามาทางขา 2 และ ขา 6 กระแสที่ใช้ของ  $\overline{RST}$  ขา 4 มีค่าไม่เกิน  $100\ \mu A$

ตัวเปรียบเทียบแรงดัน  $A_1$  และ  $A_2$  จะทำหน้าที่เปรียบเทียบศักย์ค่าไฟฟ้าที่ทางเข้าขา 2 และขา 6 กับศักย์ค่าอ้างอิง  $2/3V_{CC}$  และ  $1/3V_{CC}$  ซึ่งศักย์ค่าอ้างอิง  $2/3V_{CC}$  และ  $1/3V_{CC}$  ได้มาจากวงจรแบ่งแรงดัน  $R_1, R_2, R_3$  ซึ่งศักย์ค่าอ้างอิงของ  $A_1$  คือ  $2/3V_{CC}$  และศักย์ค่าอ้างอิงของ  $A_2$  คือ  $1/3V_{CC}$

ขา 5 : **CONTROL VOLTAGE** ใช้ในการเปลี่ยนค่าศักย์ค่าอ้างอิงของ  $A_1$  และ  $A_2$  ( ถ้าต้องการจะเปลี่ยน ) โดยนำไปต่อกับแหล่งจ่ายแรงดันอ้างอิงภายนอก IC HA17555 ในการใช้งานปกติขา 5 จะเปิดวงจร ซึ่งทำให้ศักย์ค่าไฟฟ้าอ้างอิงของ  $A_1$  และ  $A_2$  มีค่า  $2/3V_{CC}$  และ  $1/3V_{CC}$  ตามลำดับ หรืออาจต่อ  $C_B$  ประมาณ  $0.01 - 0.1\ \mu F$  กับกราวด์ เพื่อให้ศักย์ค่าไฟฟ้าอ้างอิงของ  $A_1$  และ  $A_2$  ปราศจากสัญญาณ

รบกวนที่ความถี่สูง ( ที่ความถี่สูงอิมพีแดนซ์ของ  $C_B$  จะมีค่าต่ำมากเสมือนลัดวงจร แต่จะไม่มีผลต่อไฟตรง)

**ขา 2 : TRIGGER** ถ้าศักย์คาไฟฟ้าที่ขา 2 มีค่าเท่ากับหรือน้อยกว่า  $1/3V_{CC}$  จะทำให้ทางออกของ  $A_2$  มีสถานะเป็น “ 1 ” ฟลิปฟลอปจะถูกกระตุ้น หรือทริกเกอร์ หรือเซ็ต ทำให้สัญญาณที่ออกทาง (ขา 3) เป็นสถานะ 1 และขา 7 เสมือนเปิดวงจร

**ขา 6 : TREASHOLD** ถ้าศักย์คาไฟฟ้าที่ขา 6 มีค่าเท่ากับหรือมากกว่าศักย์คา TREASHOLD  $2/3V_{CC}$  เอาท์พุทของ  $A_1$  จะเป็น “ 1 ” ฟลิปฟลอปจะถูกรีเซ็ต ทำให้ทางออกมีสถานะเป็น “ 0 ” และขา 7 เสมือนช็อดกับกราวด์

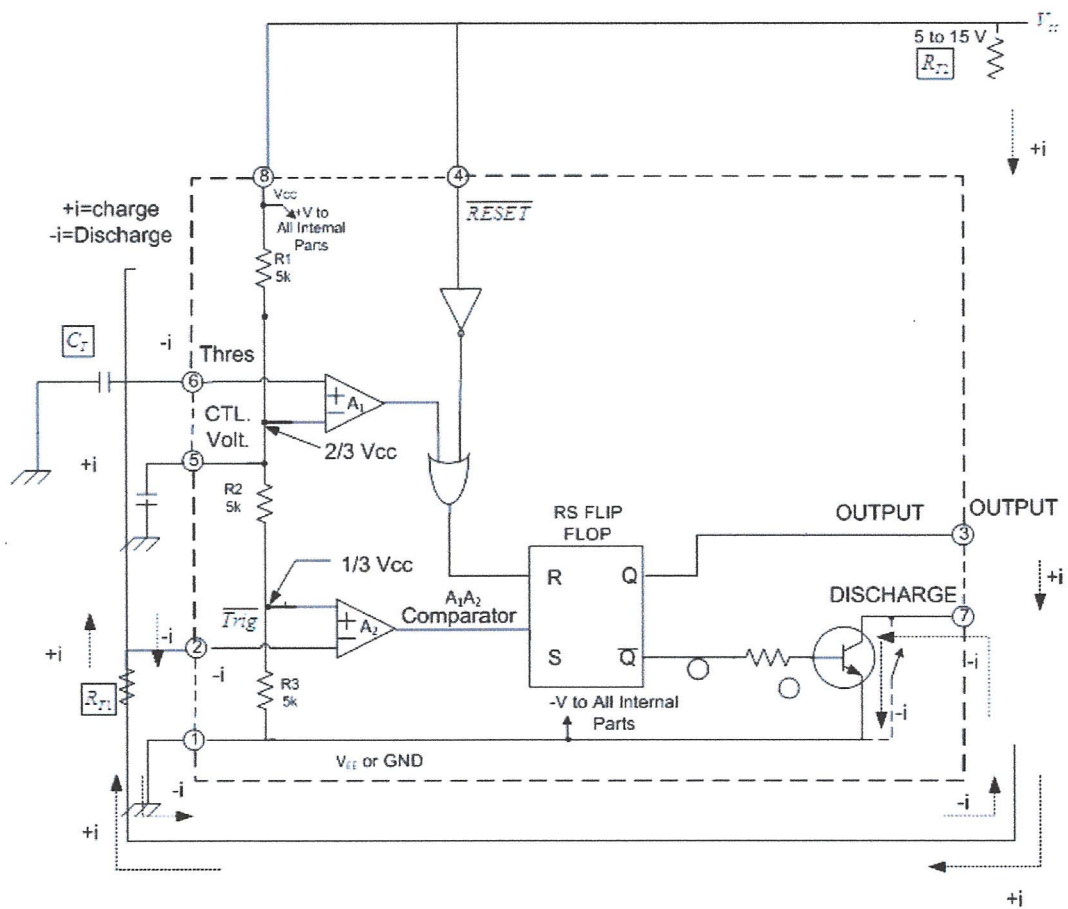
สามารถสรุปการควบคุมสถานะที่ออกทางขาของ IC HA17555 ได้ดังนี้คือ

$$\left. \begin{array}{l} V_{\text{TRIG}} (\text{ขา 2}) \leq 1/3V_{CC} \\ |I_{\text{TRIG}}| \leq 500 \text{ nA} \end{array} \right\} \begin{array}{l} \text{Output (ขา 3) = 1 และ} \\ \text{ขา 7 เสมือนเปิดวงจร} \end{array}$$

$$\left. \begin{array}{l} V_{\text{Thrs}} (\text{ขา 6}) \geq 2/3V_{CC} \\ |I_{\text{Thrs}}| \leq 100 \text{ nA} \end{array} \right\} \begin{array}{l} \text{Output (ขา 3) = 0 และ} \\ \text{ขา 7 เสมือนช็อดกับ } V_{EE} \end{array}$$

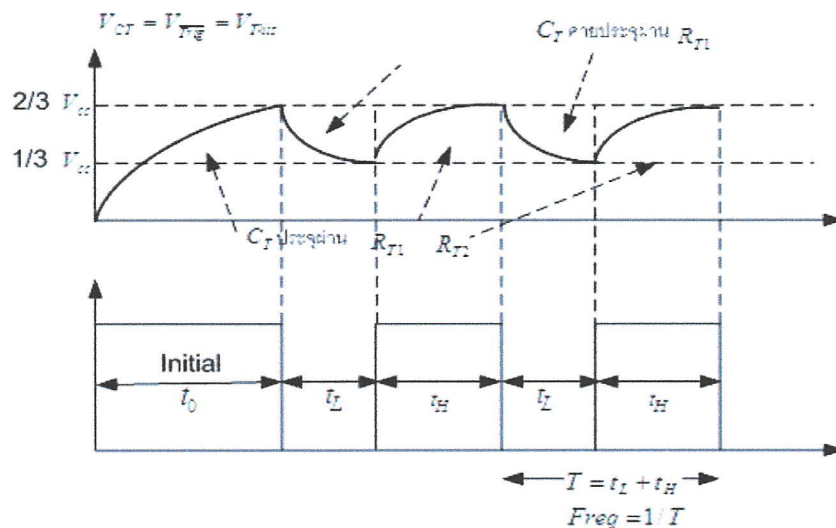
ค่ากระแสที่ใช้ในการทำงานของ TRIGGER (ขา 2) และ THRESHOLD (ขา 6) มีค่าไม่เกิน  $500 \text{ nA}$  ( $5 \times 10^{-7} \text{ A}$ )

ในการออกแบบวงจรเพื่อกำเนิดสัญญาณ โดยใช้งานเป็นวงจรอะสเตเบิลมัลติไวเบรเตอร์ คาบเวลา  $t_0$  เป็นช่วงเริ่มต้นการทำงาน ซึ่งจะมีค่าไม่แน่นอนโดยจะขึ้นอยู่กับค่าประจุที่  $C_T$  เก็บค้างไว้ก่อนที่เราจะจ่ายไฟให้กับวงจร แต่อย่างไรก็ตาม จะเป็นเหตุการณ์ที่เกิดขึ้นเพียงครั้งเดียวตลอดเวลาการทำงานของวงจร และไม่จำเป็นต้องสนใจค่าของ  $t_0$



รูปที่ 3.5 บล็อกไดอะแกรมการใช้งานวงจรรวม แบบอะอสเตเบิลต์มัลติไวเบเรเตอร์

ความเป็นจริง โดยทั่วไปหลังจากเราคัดไฟที่เต็มวงจร  $C_T$  จะคายประจุผ่าน  $R_{T2}$  ไปยังกราวด์ โดยผ่านขา  $V_{CC}$  และวงจรภายใน IC HA17555 ภายในเวลาไม่เกิน 1 วินาที มันจะคายประจุจนหมด หรือ  $V_{CT} = 0$  ก่อนที่เราจะจ่ายไฟให้กับวงจรในครั้งต่อไป เมื่อเริ่มต้นจ่ายไฟให้กับวงจร  $C_T$  จะถูกประจุจาก  $2/3V_{CC}$  สถานะการทำงานที่อยู่ตัว (Steady Stage) ก็จะเริ่มต้นขึ้น



รูปที่ 3.6 ไดอะแกรมเวลาของวงจร ในการคายและเก็บประจุ

จากรูปที่ 3.6 จะได้ว่า

$$V_{\text{Threshold}} \text{ (ขา 6)} = V_{\text{TRIG}} \text{ (ขา 2)} = V_{\text{CT}}$$

ทันทีที่  $V_{\text{CT}} = 2/3V_{\text{CC}}$  เอาท์พุทจะมีสถานะเป็น “ 0 ” และขา Discharge ( ขา 7 ) จะเสมือนชอร์ตกับกราวด์  $C_T$  จะคายประจุผ่าน  $R_{T1}$  ผ่านขา Discharge ( ขา 7 ) ไปยังกราวด์ ตลอดเวลาที่  $C_T$  คายประจุจาก  $2/3V_{\text{CC}}$  จนเหลือ  $1/3V_{\text{CC}}$  เอาท์พุทจะมีสถานะเป็น “ 0 ”

เมื่อ  $C_T$  คายประจุจนมีศักย์ไฟฟ้าเท่ากับ  $1/3V_{\text{CC}}$  เอาท์พุทจะมีสถานะเป็น “ 1 ” และขา Discharge ( ขา 7 ) จะเสมือนเปิดวงจร  $C_T$  จะถูกประจุจาก  $V_{\text{CC}}$  ผ่าน  $R_{T1}, R_{T2}$  ตลอดเวลาที่  $C_T$  ประจุจาก  $1/3V_{\text{CC}}$  ไปยัง  $2/3V_{\text{CC}}$  เอาท์พุทจะมีสถานะเป็น “ 1 ”

เมื่อ  $C_T$  ประจุจนมีศักย์ไฟฟ้าเท่ากับ  $2/3V_{\text{CC}}$  เอาท์พุทจะเปลี่ยนสถานะเป็น “ 0 ” อีกครั้งหนึ่ง  $C_T$  จะถูกประจุจาก  $2/3V_{\text{CC}}$  ไปยัง  $1/3V_{\text{CC}}$  สลับกันไป สถานะที่เอาท์พุทก็จะเป็น “ 1 ” , “ 0 ” สลับกันไปดังรูปที่

$T_L$  คือคาบเวลาที่ใช้ในการคายประจุ  $C_T$  ผ่าน  $R_{T1}$  ที่ทำให้ศักย์ไฟฟ้าของ  $C_T$  มีค่าลดลงจาก  $2/3V_{\text{CC}}$  ไปยัง  $1/3V_{\text{CC}}$  สามารถหาค่า  $T_L$  ได้จาก

$$T_x = -RC \ln \left( \frac{V_x - V_s}{V_c(0) - V_s} \right) = T_L = -R_{T1} C_T \ln \left( \frac{1/3V_{\text{CC}} - 0}{2/3V_{\text{CC}} - 0} \right) = -R_{T1} C_T \ln \left( \frac{1}{2} \right)$$

$$\text{ดังนั้น } T_L = 0.7 R_{T1} C_T \quad \text{s}$$

$T_H$  คือค่าเวลาที่ใช้ในการประจุ  $C_T$  จาก  $V_{\text{CC}}$  โดยผ่าน  $R_{T1} + R_{T2}$  ที่ทำให้ศักย์ไฟฟ้าของ  $C_T$  มีค่าเพิ่มขึ้นจาก  $1/3 V_{\text{CC}}$  เป็น  $2/3 V_{\text{CC}}$  สามารถหาค่า  $T_H$  ได้จาก

$$T_x = -RC \ln \left( \frac{V_x - V_s}{V_c(0) - V_s} \right) = T_L = -(R_{T1} + R_{T2}) C_T \ln \left( \frac{2/3V_{\text{CC}} - V_{\text{CC}}}{1/3V_{\text{CC}} - V_{\text{CC}}} \right)$$

$$= -(R_{T1} + R_{T2}) C_T \ln \left( \frac{1}{2} \right)$$

$$\text{ดังนั้น } T_H = 0.7 (R_{T1} + R_{T2}) C_T \quad \text{s}$$

คาบเวลาของ Pulse ที่วงจรผลิตหรือ Oscillate : T จะเป็น

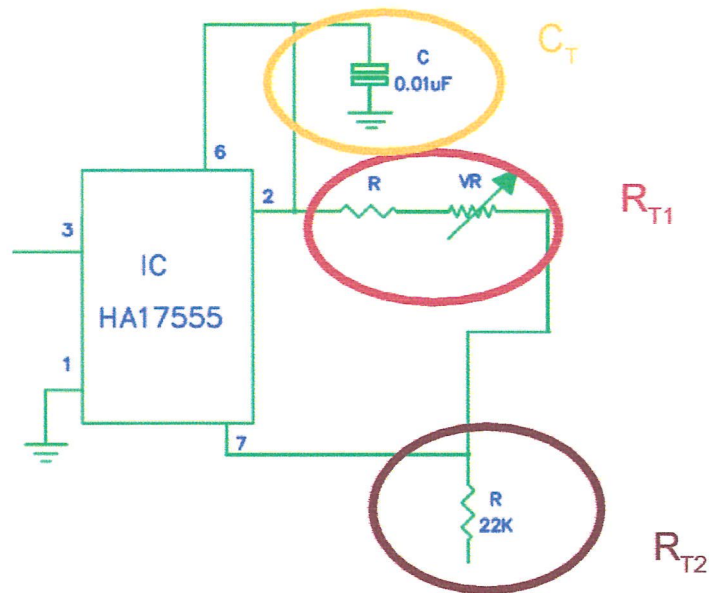
$$T = T_L + T_H$$

$$T = 0.7 C_T (2 R_{T1} + R_{T2}) \quad \text{s}$$

หรือความถี่จะมีค่าเท่ากับ

$$F = \frac{1}{T} = \frac{1}{0.7 C_T (2 R_{T1} + R_{T2})} \quad \text{Hz}$$

### 3.2.3 การออกแบบวงจรภาคส่งสัญญาณวิทยุ



รูปที่ 3.7 แสดงการต่อของวงจรภาคส่ง

จากรูปที่ 3.7 แสดงรูปแบบการต่อที่เทียบกับรูปที่ 3.5 โดยเป็นการต่อแบบ ออสเตเบิลมัลติไวยเบรเตอร์ (Astable Multivibrator) มี  $C_T = 0.01\mu\text{F}$  เข้าที่ขา 6 ซึ่งเป็น Threshold ของ IC HA17555 และมีการต่อ  $R_{T1}$  และ  $R_{T2}$  เข้าที่ขา 2 และขา 7 ซึ่งค่าของทั้ง 2 ตัวจะมีดังนี้คือ  $R_{T1}$  มีค่า  $R$  100 K $\Omega$  และ VR 100 K $\Omega$  สำหรับการปรับค่าให้ตรงตามที่ต้องการ โดย  $R_{T2}$  มีค่าเท่ากับ 22 K $\Omega$

การสร้างจะออกแบบให้ความถี่ของแต่ละช่องสัญญาณมีค่าที่แตกต่างกันคือ 300 Hz , 450 Hz และ 600 Hz โดยสามารถคำนวณการเลือกใช้อุปกรณ์ได้ดังนี้คือ

$$\text{จาก } T_L = 0.7R_{T1}C_T$$

$$T_L = 0.7R_{T1}(0.01 \times 10^{-6})$$

$$T_L = (0.007 \times 10^{-6}) R_{T1} \quad \text{s} \quad (\text{สมการที่ 1})$$

$$T_H = 0.7 (R_{T1} + R_{T2}) C_T$$

$$T_H = 0.7 (R_{T1} + 22 \times 10^3) (0.01 \times 10^{-6})$$

$$T_H = (0.007 \times 10^{-6}) (R_{T1} + 22 \times 10^3) \quad \text{s} \quad (\text{สมการที่ 2})$$

$$T = T_L + T_H$$

$$T = (0.007 \times 10^{-6}) R_{T1} + (0.007 \times 10^{-6}) (R_{T1} + 22 \times 10^3)$$

$$T = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3) \quad \text{s} \quad (\text{สมการที่ 3})$$

ความถี่ที่ต้องการในช่องสัญญาณที่ 1 คือ  $f = 300 \text{ Hz}$  หรือ  $T = 3.33 \times 10^{-3} \text{ s}$  โดย  $C_T = 0.01 \text{ uF}$

จากสมการที่ 3

$$T = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$3.33 \times 10^{-3} = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$\frac{3.33 \times 10^{-3}}{0.007 \times 10^{-6}} = (2 R_{T1} + 22 \times 10^3)$$

$$(475.714 - 22) \times 10^3 = 2 R_{T1}$$

$$\frac{543.71 \times 10^3}{2} = R_{T1}$$

$$R_{T1} = 226.85 \text{ K}\Omega$$

โดยสามารถแบ่งเป็นค่าดังนี้คือ  $R = 200 \text{ K}\Omega$  และ  $VR = 26.85 \text{ K}\Omega$

จากสมการที่ 1 สามารถหาช่วงเวลาของ  $T_L$  ได้ดังนี้

$$T_L = (0.007 \times 10^{-6}) R_{T1}$$

$$T_L = (0.007 \times 10^{-6}) (226.85 \times 10^3)$$

$$T_L = 1.588 \text{ ms}$$

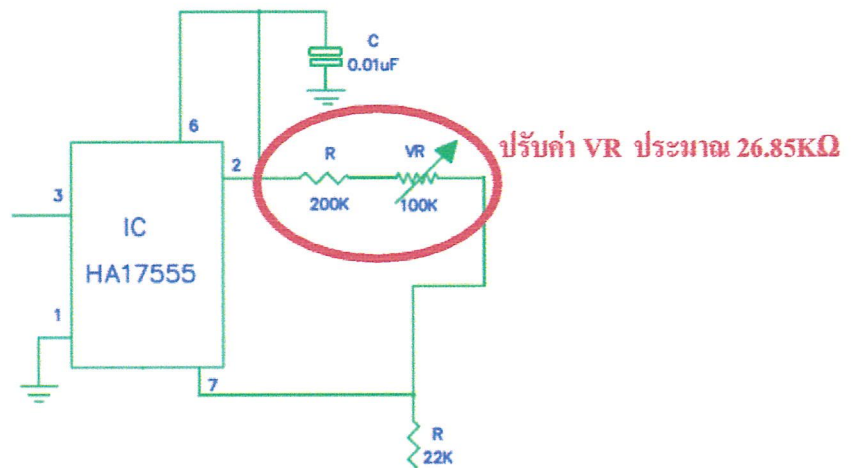
จากสมการที่ 2 สามารถหาช่วงเวลาของ  $T_H$  ได้ดังนี้

$$T_H = (0.007 \times 10^{-6}) (R_{T1} + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (226.85 \times 10^3 + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (248.85 \times 10^3)$$

$$T_H = 1.742 \text{ ms}$$



รูปที่ 3.8 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้

ความถี่ที่ต้องการในช่องสัญญาณที่ 2 คือ  $f = 450 \text{ Hz}$  หรือ  $T = 2.22 \times 10^{-3} \text{ s}$  โดย  $C_T = 0.01 \text{ uF}$   
จากสมการที่ 3

$$T = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$2.22 \times 10^{-3} = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$\frac{2.22 \times 10^{-3}}{0.007 \times 10^{-6}} = (2 R_{T1} + 22 \times 10^3)$$

$$(317.142 - 22) \times 10^3 = 2 R_{T1}$$

$$\frac{295.14 \times 10^3}{2} = R_{T1}$$

$$R_{T1} = 147.57 \text{ K}\Omega$$

โดยสามารถแบ่งเป็นค่าดังนี้คือ  $R = 100 \text{ K}\Omega$  และ  $VR = 47.57 \text{ K}\Omega$

จากสมการที่ 1 สามารถหาช่วงเวลาของ  $T_L$  ได้ดังนี้

$$T_L = (0.007 \times 10^{-6}) R_{T1}$$

$$T_L = (0.007 \times 10^{-6}) (147.57 \times 10^3)$$

$$T_L = 1.033 \text{ ms}$$

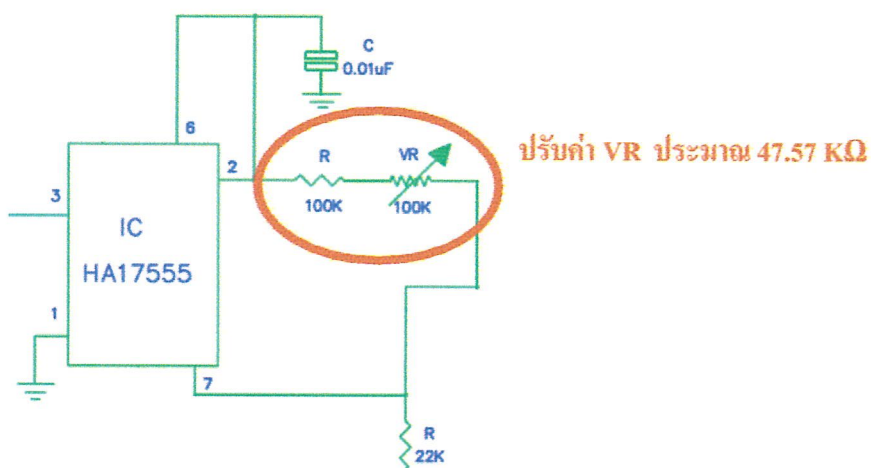
จากสมการที่ 2 สามารถหาช่วงเวลาของ  $T_H$  ได้ดังนี้

$$T_H = (0.007 \times 10^{-6}) (R_{T1} + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (147.57 \times 10^3 + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (169.57 \times 10^3)$$

$$T_H = 1.187 \text{ ms}$$



รูปที่ 3.9 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้

ความถี่ที่ต้องการในช่องสัญญาณที่ 3 คือ  $f = 600 \text{ Hz}$  หรือ  $T = 1.66 \times 10^{-3} \text{ s}$  โดย  $C_T = 0.01 \text{ uF}$   
จากสมการที่ 3

$$T = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$1.66 \times 10^{-3} = (0.007 \times 10^{-6}) (2 R_{T1} + 22 \times 10^3)$$

$$\frac{1.66 \times 10^{-3}}{0.007 \times 10^{-6}} = (2 R_{T1} + 22 \times 10^3)$$

$$(238.098 - 22) \times 10^3 = 2 R_{T1}$$

$$\frac{216.098 \times 10^3}{2} = R_{T1}$$

$$R_{T1} = 108.47 \text{ K}\Omega$$

โดยสามารถแบ่งเป็นค่าดังนี้คือ  $R = 100 \text{ K}\Omega$  และ  $VR = 8.47 \text{ K}\Omega$

จากสมการที่ 1 สามารถหาช่วงเวลาของ  $T_L$  ได้ดังนี้

$$T_L = (0.007 \times 10^{-6}) R_{T1}$$

$$T_L = (0.007 \times 10^{-6}) (108.47 \times 10^3)$$

$$T_L = 0.759 \text{ ms}$$

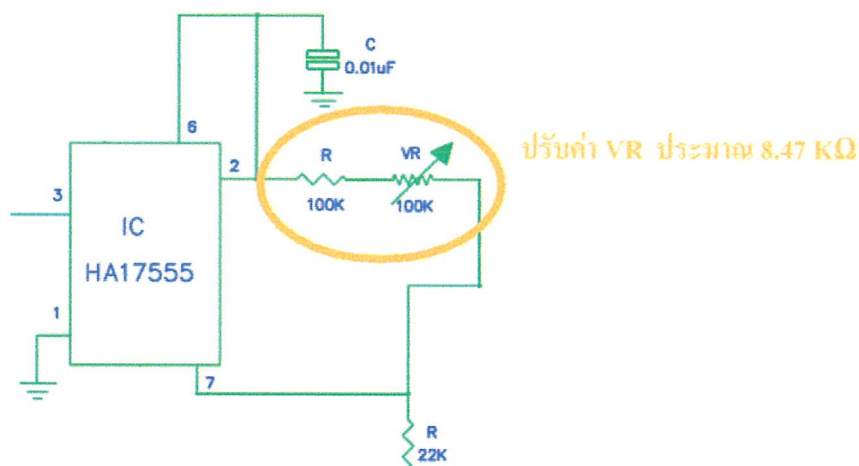
จากสมการที่ 2 สามารถหาช่วงเวลาของ  $T_H$  ได้ดังนี้

$$T_H = (0.007 \times 10^{-6}) (R_{T1} + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (108.47 \times 10^3 + 22 \times 10^3)$$

$$T_H = (0.007 \times 10^{-6}) (130.47 \times 10^3)$$

$$T_H = 0.913 \text{ ms}$$



รูปที่ 3.10 แสดงการวางอุปกรณ์ตามที่ได้คำนวณไว้

ช่วงการทำงาน (Duty Cycle) หาได้จากสูตร

$$D = \frac{T_H}{T_H + T_L} \times 100 \% \quad (\text{สมการ 4})$$

เมื่อ  $T_H$  = เวลาช่วง Logic “ 1 ” และเมื่อ  $T_L$  = เวลาช่วง Logic “ 0 ”

$$T = \text{คาบเวลาทั้งหมดคือ } T_H + T_L$$

ช่องสัญญาณที่ 1 คือ  $f = 300 \text{ Hz}$  ค่า Duty Cycle คือ

โดย  $T_H = 1.742 \text{ ms}$  และ  $T_L = 1.588 \text{ ms}$

$$D = \frac{1.742}{1.742 + 1.588} \times 100 \%$$

$$D = \frac{1.742}{3.33} \times 100 \%$$

$$D = 52.34 \%$$

ช่องสัญญาณที่ 1 คือ  $f = 450 \text{ Hz}$  ค่า Duty Cycle คือ

โดย  $T_H = 1.187 \text{ ms}$  และ  $T_L = 1.033 \text{ ms}$

$$D = \frac{1.187}{1.187 + 1.033} \times 100 \%$$

$$D = \frac{1.187}{2.22} \times 100 \%$$

$$D = 53.46 \%$$

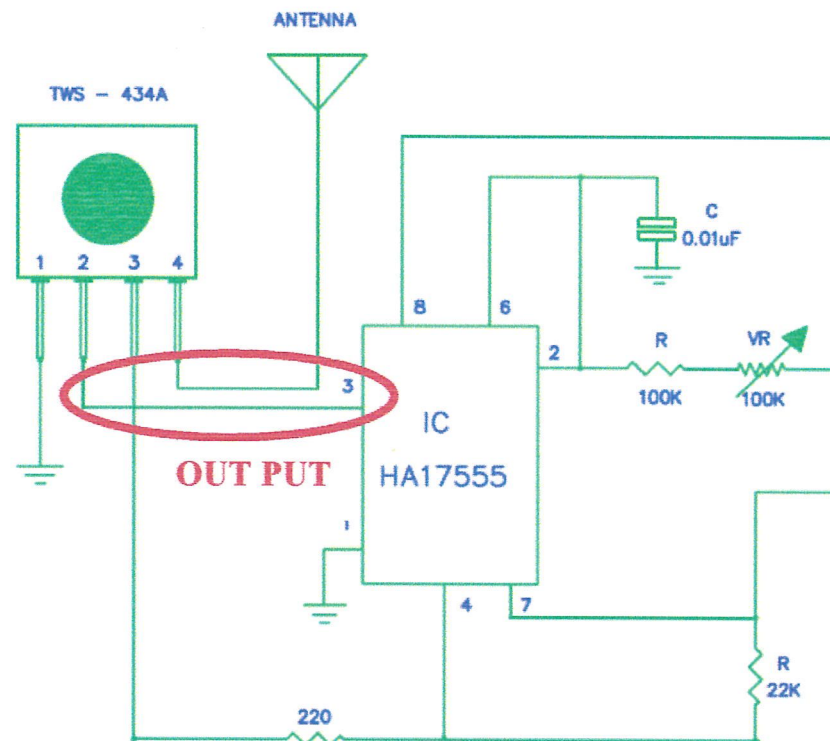
ช่องสัญญาณที่ 1 คือ  $f = 600 \text{ Hz}$  ค่า Duty Cycle คือ

โดย  $T_H = 0.913 \text{ ms}$  และ  $T_L = 0.759 \text{ ms}$

$$D = \frac{0.913}{0.913 + 0.759} \times 100 \%$$

$$D = \frac{0.913}{1.672} \times 100 \%$$

$$D = 54.60 \%$$



รูปที่ 3.11 วงจรส่งสัญญาณคลื่นวิทยุ

สัญญาณที่ถูกสร้างจาก IC HA17555 จะถูกส่งออกที่เอาต์พุตขา 3 และถูกส่งไปยังโมดูลภาคส่งสัญญาณคลื่นวิทยุ TLP - 434A ทางขา 2 (Data In) เมื่ออุปกรณ์รับอินพุตจะถูกผสมสัญญาณพาหะและทำการส่งออกยังขาที่ 4 (Antenna RF Output) ออกไปยังสายอากาศ ในรูปแบบของสัญญาณคลื่นวิทยุ

การทำงานของ TLP - 434A จะมอดูเลต (Modulate) สัญญาณแบบ (ASK : Amplitude-Shift Keying) สัญญาณพาหะที่ส่งไปยังภาครับผ่านทางสายอากาศ ใช้ความถี่ 430 MHz อยู่ในย่านความถี่ UHF (Ultra High Frequency ย่านความถี่สัญญาณวิทยุอยู่ระหว่าง 300 - 3000 MHz) ความยาวคลื่นที่ได้จากการคำนวณคือ

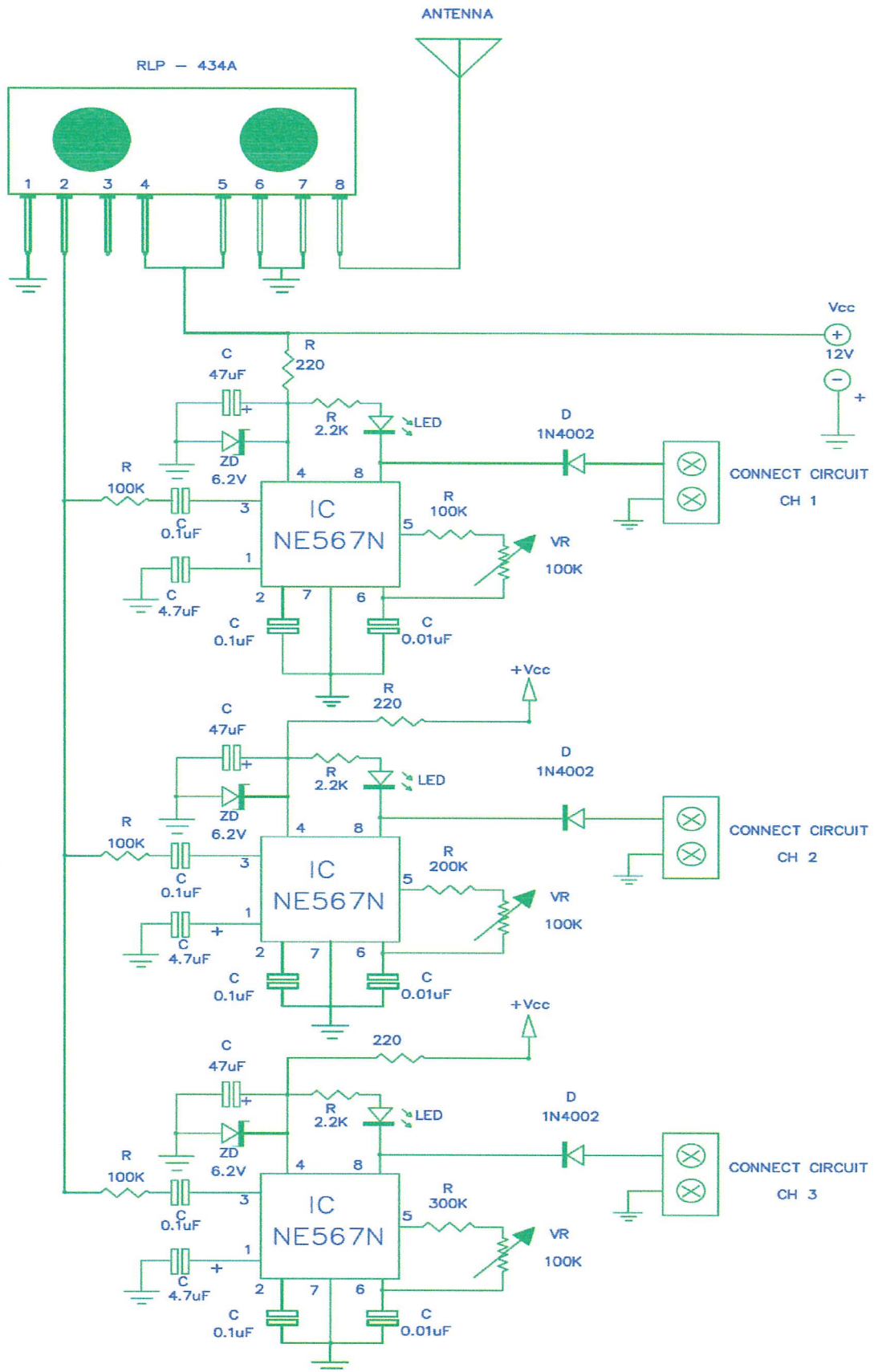
$$\lambda \text{ ความยาวคลื่น (m)} = C \text{ ความเร็วแสง (m/s)} / \text{ความถี่ (Hz)}$$

$$\lambda = 3 \times 10^8 / 430 \times 10^6$$

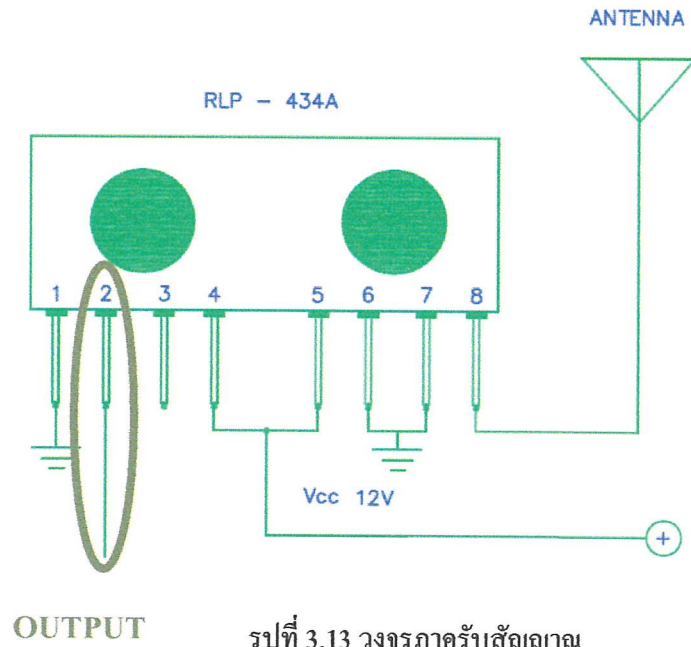
$$\lambda = 0.697 \text{ m}$$

ดังนั้นความยาวคลื่นคือ 0.697 เมตร

### 3.2.4 หลักการทำงานของภาครับสัญญาณวิทยุ



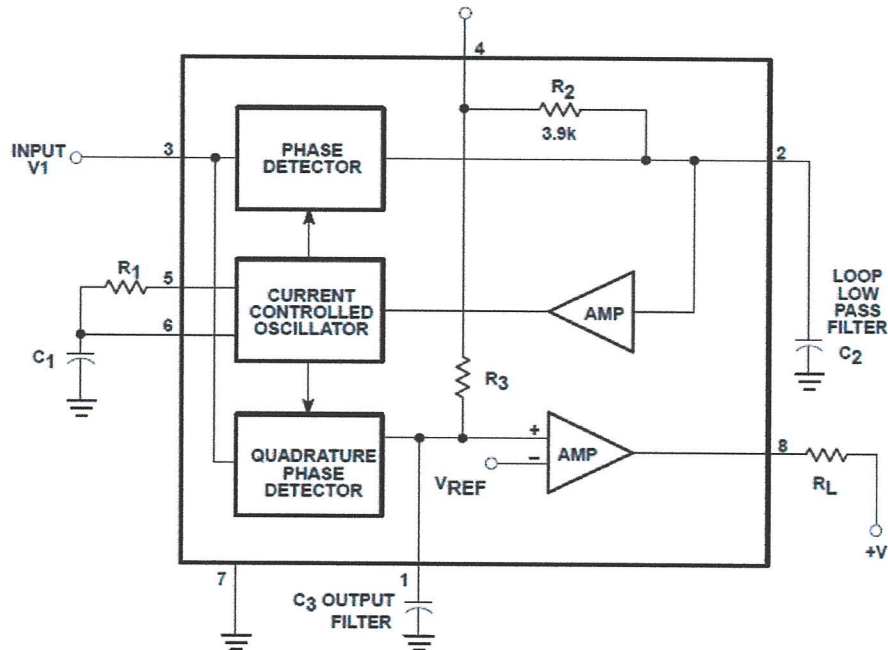
รูปที่ 3.12 วงจรรวมของภาครับสัญญาณวิทยุ



รูปที่ 3.13 วงจรภาครับสัญญาณ

ภาครับสัญญาณสัญญาณวิทยุ โดยการทำงานเริ่มจากการใช้โมดูลรับสัญญาณ RLP - 434A ทำหน้าที่รับคลื่นสัญญาณวิทยุที่ส่งเข้ามาผ่านทางขาที่ 8 ( Antenna RF Input ) เมื่อรับสัญญาณที่เข้ามาได้วงจรภายในของ RLP - 434A จะทำการดีมอดูเลต ( Demodulate ) สัญญาณพาหะ ( Carrier ) ออกจากสัญญาณข่าวสารที่ต้องการรับ

เมื่อแยกได้สัญญาณที่ต้องการแล้ว RLP - 434A ก็จะส่งเอาท์พุทท์ออก อุปกรณ์ชนิดนี้สามารถแยกข้อมูลที่ได้รับได้ทั้งในแบบดิจิทัล ( ออกทางขาที่ 2 ) และแบบอะนาล็อก ( ออกทางขาที่ 3 ( Linear Output) ) ซึ่งเอาท์พุทท์ที่ใช้นั้นจะส่งออกทางขา 2 ( Digital Data Output ) ไปยัง IC NE567N



รูปที่ 3.14 บล็อกไดอะแกรมภายในของ IC NE567

### 3.2.5 หลักการทำงานของภาครับสัญญาณวิทยุ

ลักษณะโดยทั่วไปของ IC NE567 คือเป็นตัวถอดรหัสความถี่ในลักษณะของ phase-locked loop กับการสัญญาณที่มีความสอดคล้องกันและสามารถส่งกำลังออกทางทางขาเอาต์พุตได้ รูปแบบพื้นฐานการทำงานคือการจับกำลังงาน โดยที่ เมื่อไหร่ก็ตามที่พบความถี่ที่สอดคล้องกับที่ตั้งวงจรไว้ ภายใน อุปกรณ์จะค้นหาแถบความถี่และเทียบกับสัญญาณอินพุต ศูนย์กลางช่วงความถี่ (bandwidth center frequency) และเอาต์พุตคือ โดยภายในนั้นมีส่วนประกอบหลักด้วยกันอยู่ 4 ส่วนคือ

**PHASE DETECTOR** : เมื่ออินพุตเข้ามาทางขาที่ 3 ของ IC NE567 แล้วจะผ่านส่วนของ PHASE DETECTOR เป็นอันดับแรก โดยจะมีหน้าที่ในการรับสัญญาณเข้ามาแล้วส่งผ่าน AMP เพื่อทำการขยายสัญญาณที่ได้มา มีกำลังมากขึ้น และส่งสัญญาณที่มีความถี่ไม่ได้อยู่ในช่วง 0.01 Hz ถึง 500 KHz ออกทางขาที่ 2

**AMP** : ทำหน้าที่ขยายสัญญาณภายในวงจร ซึ่งในวงจรมันประกอบด้วย AMP ตัว คืออยู่ระหว่าง PHASE DETECTOR กับ CURRENT CONTROLLED OSCILLATOR โดยหน้าที่ของตัวนี้จะทำหน้าที่ในการขยายสัญญาณที่รับได้และส่งเข้ามาเพื่อเปรียบเทียบกับสัญญาณที่ตั้งไว้

ส่วน AMP ที่ต่ออยู่ระหว่าง QUADRATURE PHASE DETECTOR กับ เอาต์พุตขา 8 เพื่อขยายสัญญาณที่ออกจาก QUADRATURE PHASE DETECTOR

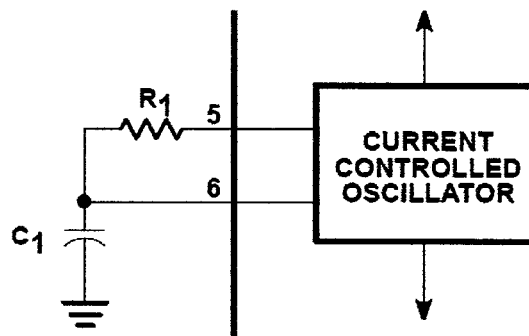
**CURRENT CONTROLLED OSCILLATOR** : เมื่อสัญญาณที่รับได้ เข้าผ่าน AMP แล้วจะส่งสัญญาณมาที่ CURRENT CONTROLLED OSCILLATOR โดยจะต่อกับ  $R_1$  และ  $C_1$  เพื่อสร้างวงจรเพื่อเปรียบเทียบกับความถี่ที่ตั้งไว้ กับความถี่ที่รับมาได้แล้วเทียบกัน โดยสามารถเขียนเป็นสมการในการออกแบบได้ดังนี้คือ

ความถี่กลาง (Center Frequency)

จาก  $T_0 = 1.1 R_1 C_1$

$$f_0 = \frac{1}{1.1 R_1 C_1} \quad \text{Hz} \quad (\text{สมการที่ 5})$$

โดย  $R_1$  ต่อกับขาที่ 5 และ  $C_1$  ต่อกับขาที่ 6 และลงกราวด์



รูปที่ 3.15 แสดงการต่อ  $R_1$  และ  $C_1$  กับ IC NE567

จากนั้นเมื่อทำการเปรียบเทียบเฟสของสัญญาณที่เข้ามากับเฟสที่ทำการล็อกไว้ หากสัญญาณความถี่มีความแตกต่างกับที่ต้องการจะส่งออกทาง PHASE DETECTOR และส่งออกทางขา 2 เพื่อทิ้งสัญญาณที่ไม่ต้องการ

แต่ถ้าสัญญาณตรงกับความถี่ในเฟสล็อกที่ตั้งไว้ สัญญาณความถี่จะส่งออกทาง QUADRATURE PHASE DETECTOR เพื่อออกทางเอาต์พุตต่อไป

**QUADRATURE PHASE DETECTOR** : ทำหน้าที่ในการรวมสัญญาณความถี่แปลงให้เป็นสัญญาณเอาต์พุต โดยมี  $C_3$  ทำหน้าที่ในการกรองสัญญาณที่ไม่ต้องการออก และส่งออกเข้าที่ AMP เพื่อทำการขยายและส่งออกที่เอาต์พุตขาที่ 8 ของ IC NE567 โดยเมื่อปรกติขณะที่ไม่มีความถี่สัญญาณใดๆ เอาต์พุตจะมีค่าเป็น “HIGH” แต่เมื่อมีสัญญาณเข้ามาจะมีหารเปลี่ยนเป็น “LOW”

หน้าที่ของแต่ละขาของ IC NE567

**ขา 3 : INPUT** ทำหน้าที่ในการรับอินพุตเข้ามาในวงจร

**ขา 2 : LOW-PASS FILTER CAPACITOR** เป็นส่วนที่ทำหน้าที่ในการนำสัญญาณที่ไม่ตรงกับที่ต้องการออก หรือสัญญาณรบกวนออกโดยผ่านทาง  $C_2$  ลงกราวด์

**ขา 4 :  $V_{CC}$**  สำหรับจ่ายไฟชั่วคราวให้กับวงจรภายในทั้งหมด สามารถป้อนความต่างศักย์ไฟฟ้าที่ 5 - 15 V และสามารถใส่กระแสอยู่ในช่วง 100 – 300 mA ซึ่งในการใช้งานจะใช้ศักย์ไฟฟ้าอยู่ที่ 12 V

**ขา 7 : GND** เป็นขาสำหรับจ่ายไฟชั่วคราวให้กับวงจรภายในทั้งหมด โดยทั่วไปมักใช้เป็นกราวด์

**ขา 5 และ ขา 6 : TIMING ELEMENT** ใช้สำหรับต่อ  $R_1$  และ  $C_1$  เพื่อทำการล็อกเฟสของสัญญาณ ในวงจรเพื่อสร้างความสัมพันธ์เป็น Center Frequency การรวมกันของสัมประสิทธิ์ตัวแปรทั้งสองนี้เพื่อให้เกิดเสถียรภาพในการสร้างความถี่อ้างอิงนี้

**ขา 1 : OUTPUT FILTER CAPACITOR** ในส่วนนี้จะทำหน้าที่ในการตัดรูปแบบของสัญญาณที่เราต้องการจะส่งออกไปเป็นในลักษณะทางระดับสัญญาณไฟในแบบ Logic โดยมี  $C_3$  ทำหน้าที่ในการกรองสัญญาณออกลงกราวด์

**ขา 8 : OUTPUT** ทำหน้าที่ส่งสัญญาณที่ต้องการออก โดยสถานะจะมี 2 ค่าคือ “HIGH” จะอยู่ในสถานะปรกติ และ “LOW” จะส่งออกมาเมื่อมีสัญญาณเข้ามา

### 3.2.4 การออกแบบทำงานของภาครับสัญญาณวิทยุ

ความถี่ที่ต้องการเลือกในช่องสัญญาณที่ 1 คือ  $f = 300 \text{ Hz}$  หรือ  $T = 3.33 \times 10^{-3} \text{ s}$

โดยการออกแบบจะกำหนดให้ค่า  $C_1$  มีค่าเท่ากับ  $0.01 \times 10^{-6} \text{ F}$

จากสมการที่ 4

$$F_0 = \frac{1}{1.1R_1C_1}$$

$$300 = \frac{1}{1.1R_1(0.01 \times 10^{-6})}$$

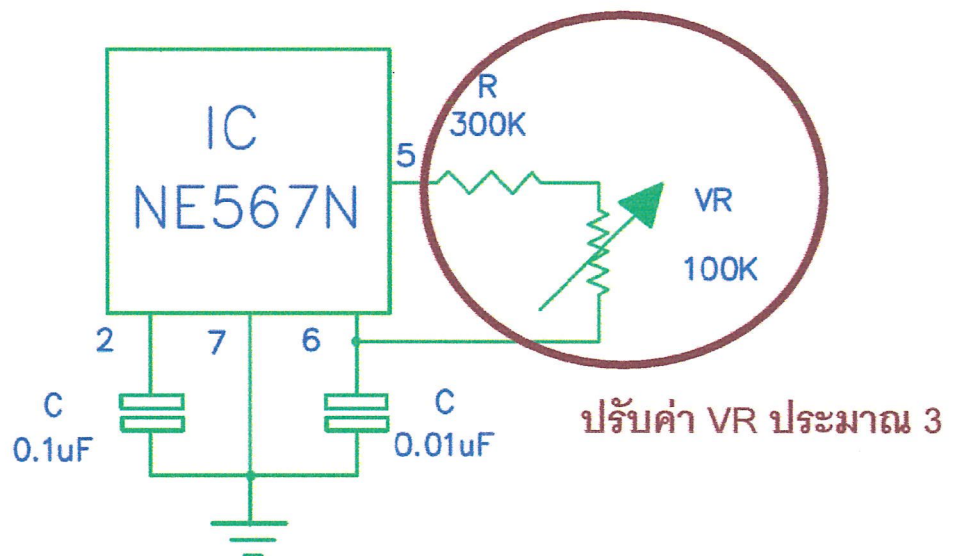
$$R_1 = \frac{1}{1.1(300)(0.01 \times 10^{-6})}$$

$$R_1 = \frac{1}{3.3 \times 10^{-6}}$$

$$R_1 = 303 \text{ K}\Omega$$

ดังนั้นความต้านทานในช่องสัญญาณของความถี่นี้คือ  $303 \text{ K}\Omega$  โดยแบ่งเป็น  $R = 300 \text{ K}\Omega$  และ

$VR = 3 \text{ K}\Omega$



รูปที่ 3.16 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรถูกเฟส

ความถี่ที่ต้องการเลือกในช่องสัญญาณช่องที่ 2 คือ  $f = 450 \text{ Hz}$  หรือ  $T = 2.22 \times 10^{-3} \text{ s}$

โดยการออกแบบจะกำหนดให้ค่า  $C_1$  มีค่าเท่ากับ  $0.01 \times 10^{-6} \text{ F}$

จากสมการที่ 4

$$F_0 = \frac{1}{1.1R_1C_1}$$

$$450 = \frac{1}{1.1R_1(0.01 \times 10^{-6})}$$

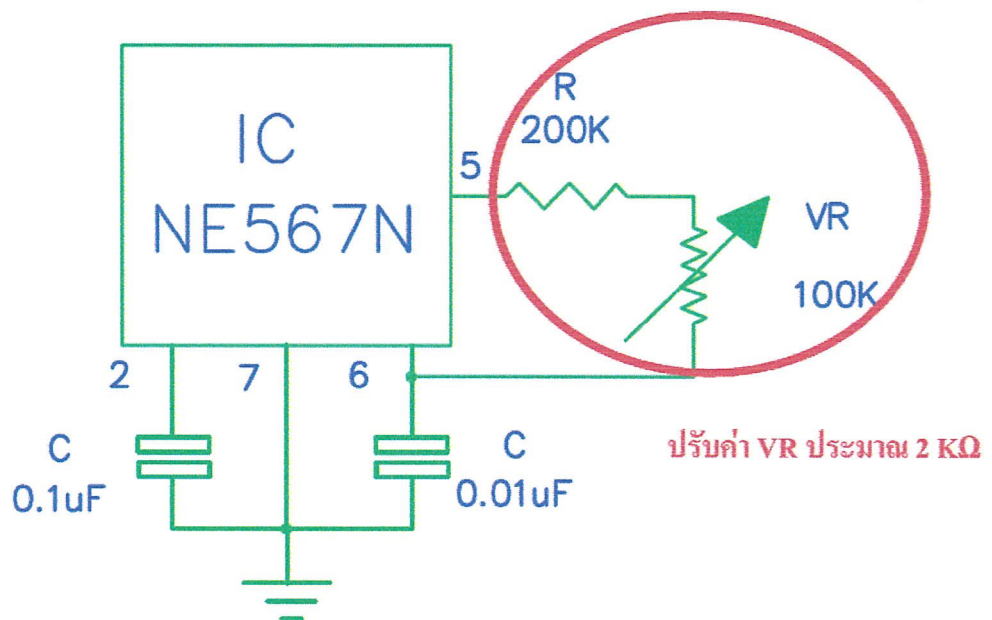
$$R_1 = \frac{1}{1.1(450)(0.01 \times 10^{-6})}$$

$$R_1 = \frac{1}{4.95 \times 10^{-6}}$$

$$R_1 = 202 \text{ K}\Omega$$

ดังนั้นความต้านทานในช่องสัญญาณของความถี่นี้คือ  $202 \text{ K}\Omega$  โดยแบ่งเป็น  $R = 200 \text{ K}\Omega$  และ

$VR = 2 \text{ K}\Omega$



รูปที่ 3.17 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรถอดเฟส

ความถี่ที่ต้องการเลือกในช่องสัญญาณช่องที่ 3 คือ  $f = 600 \text{ Hz}$  หรือ  $T = 1.66 \times 10^{-3} \text{ s}$

โดยการออกแบบจะกำหนดให้ค่า  $C_1$  มีค่าเท่ากับ  $0.01 \times 10^{-6} \text{ F}$

จากสมการที่ 4

$$F_0 = \frac{1}{1.1R_1C_1}$$

$$600 = \frac{1}{1.1R_1(0.01 \times 10^{-6})}$$

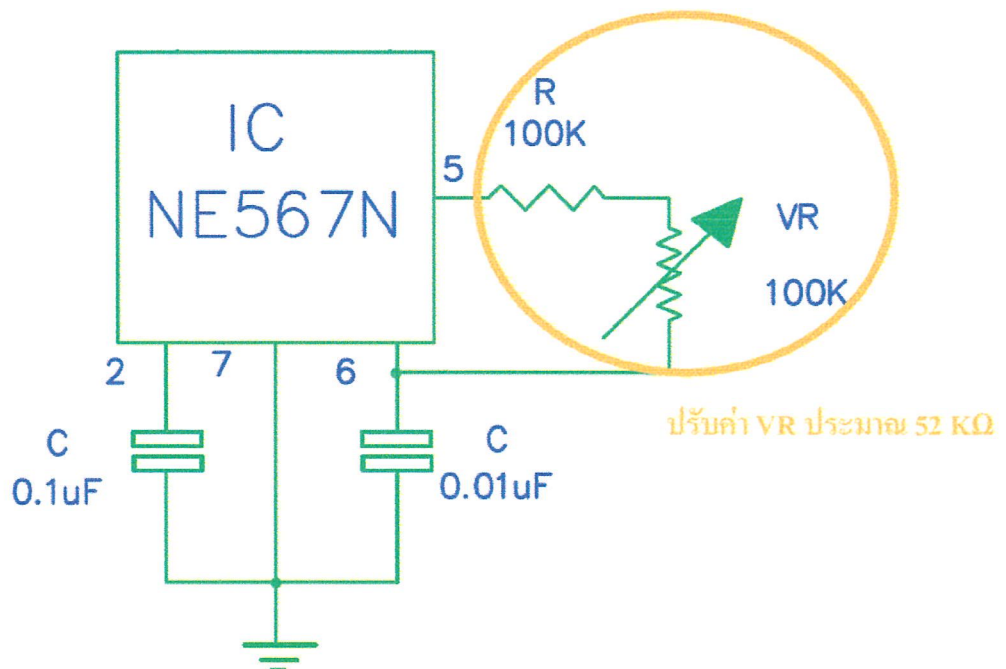
$$R_1 = \frac{1}{1.1(600)(0.01 \times 10^{-6})}$$

$$R_1 = \frac{1}{6.6 \times 10^{-6}}$$

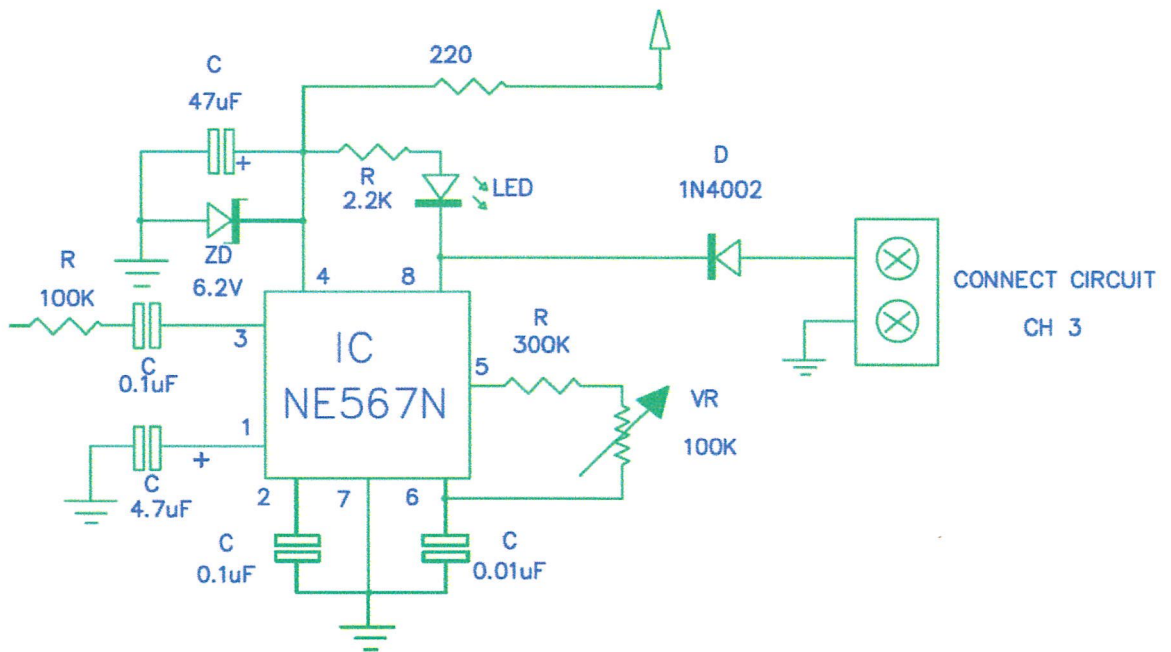
$$R_1 = 151.51 \text{ K}\Omega$$

ดังนั้นความต้านทานในช่องสัญญาณของความถี่นี้คือ  $151.51 \text{ K}\Omega$  โดยแบ่งเป็น  $R = 100 \text{ K}\Omega$

และ  $VR = 51.51 \text{ K}\Omega$



รูปที่ 3.18 แสดงการวางอุปกรณ์ภาครับที่คำนวณได้ในการสร้างวงจรถูกเฟส



รูปที่ 3.19 วงจรถอดรหัสสัญญาณ

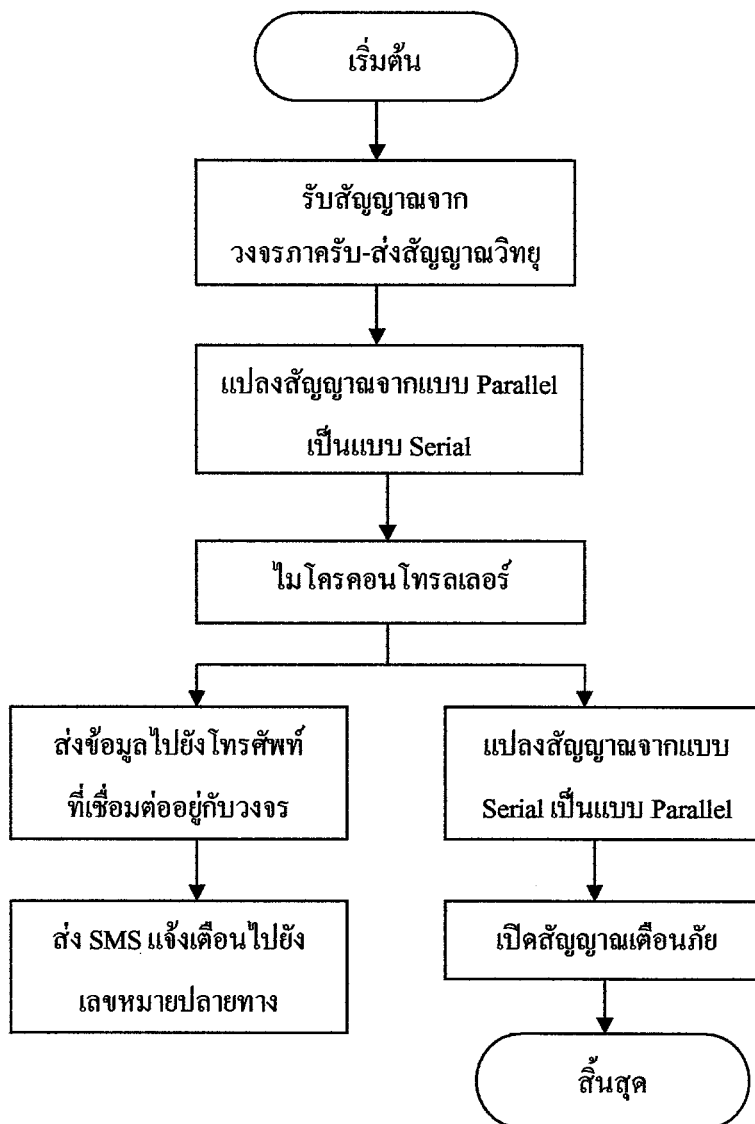
เอาต์พุตที่สัญญาณจาก RLP - 434A ผ่านตัวต้านทาน ( R ) ขนาด 100K โอมห์ และตัวเก็บประจุ ( C ) ขนาด 0.1 uF เพื่อปรับแอมพลิจูดของสัญญาณที่เข้ามาให้ตรงกับขนาดที่ต้องการ และส่งไปยัง IC NE567N เข้าเป็นอินพุตที่ทางขาที่ 3 ( Input )

หลักการทำงานของ IC NE567N จะเป็นเบอร์คู่กับ IC HA17555 ซึ่งจะเป็นตัวสร้าง และถอดรหัส หน้าทีของอุปกรณ์ชนิดนี้คือถอดรหัสสัญญาณที่รับเข้ามา ตัวต้านทานปรับค่าได้ ( VR ) ชนิดเกือกม้าทำหน้าที่กำหนดความถี่ของสัญญาณ ขนาดของความต้านทานมีการปรับค่าที่แตกต่างกัน ( โดยการปรับค่าสัญญาณในแต่ละช่อง ) และตัวต้านทาน ( R ) ในแต่ละช่องสัญญาณมีขนาดที่แตกต่างกันคือ 100K, 200K, 300K โอมห์ ตามลำดับ จะสามารถกำหนดการถอดรหัสสัญญาณที่มีความถี่แตกต่างกัน โดยการถอดรหัสนั้นจะนำตัวต้านทานต่อคร่อมขาที่ 5 (Timing Element R) และขาที่ 6 (Timing Element R and C ) ซึ่งทำให้เกิดการเปรียบเทียบในสัญญาณที่เข้ามาแต่ละตัว เพื่อทำการถอดรหัส ว่าเป็นสัญญาณของช่องสื่อสารใด

เมื่อทำการถอดรหัสที่แล้ว IC NE567N จะส่งเอาต์พุตออกที่ขา 8 (Output) จากนั้นสัญญาณที่ถูกส่งจะทำหน้าที่ควบคุมการทำงานของงานรีเลย์ ใช้สำหรับเป็นอินพุตในการส่งข้อมูลเข้าที่วงจรควบคุม

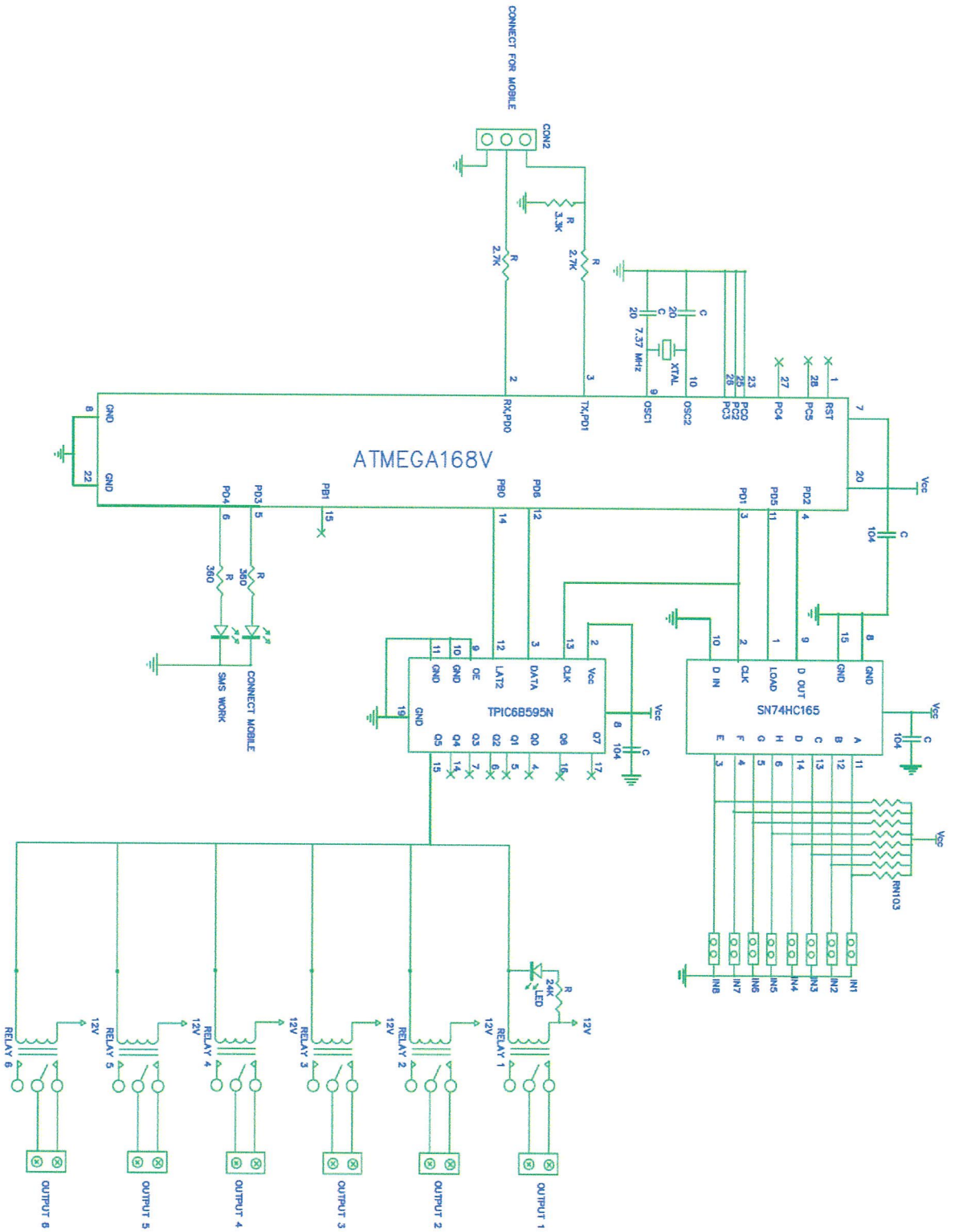
### 3.3 วงจรควบคุมไมโครคอนโทรลเลอร์ และ เปิด-ปิดสัญญาณเตือนภัย

วงจรในส่วนนี้จะทำหน้าที่รับสัญญาณ จากวงจรรับ-ส่งสัญญาณวิทยุ แล้วมาทำการวิเคราะห์และแจ้งเหตุทางการส่ง SMS และควบคุมการเปิด-ปิด สัญญาณเตือนภัย



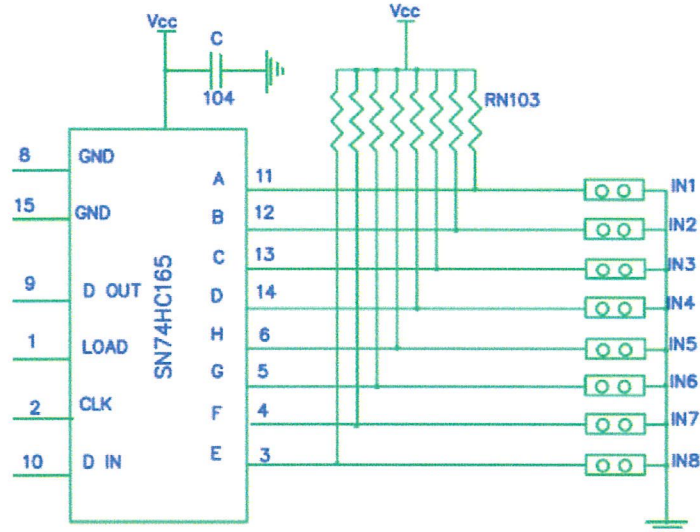
รูปที่ 3.20 บล็อกไดอะแกรมส่วนควบคุมการส่ง SMS และควบคุมการ เปิด - ปิด สัญญาณเตือนภัย

### 3.3.1 วงจรควบคุมการส่ง SMS และเปิด-ปิดสัญญาณเตือนภัย รูปวงจรทั้งหมดของวงจรควบคุมการทำงาน



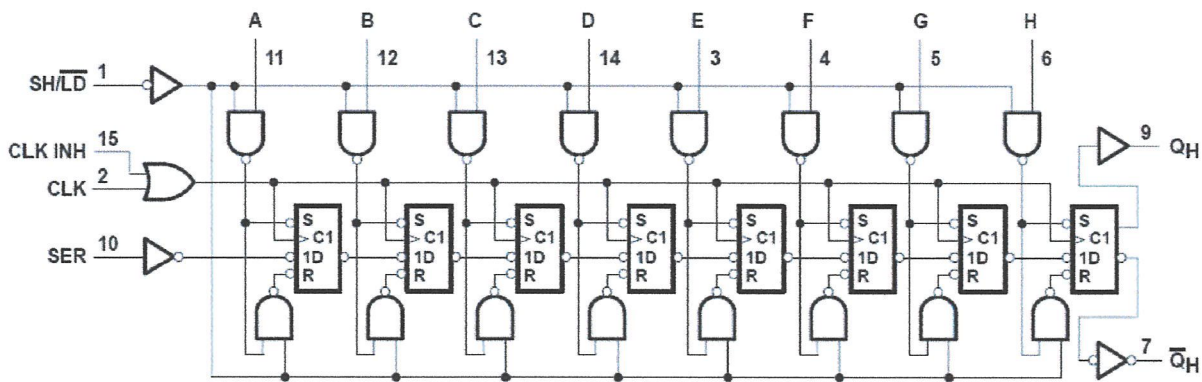
รูปที่ 3.21 วงจรไมโครคอนโทรลเลอร์ที่ควบคุมอุปกรณ์

### 3.3.2 การออกแบบด้านอินพุตของวงจรควบคุม



รูปที่ 3.22 วงจรแปลงข้อมูลจากแบบขนานเป็นแบบอนุกรม

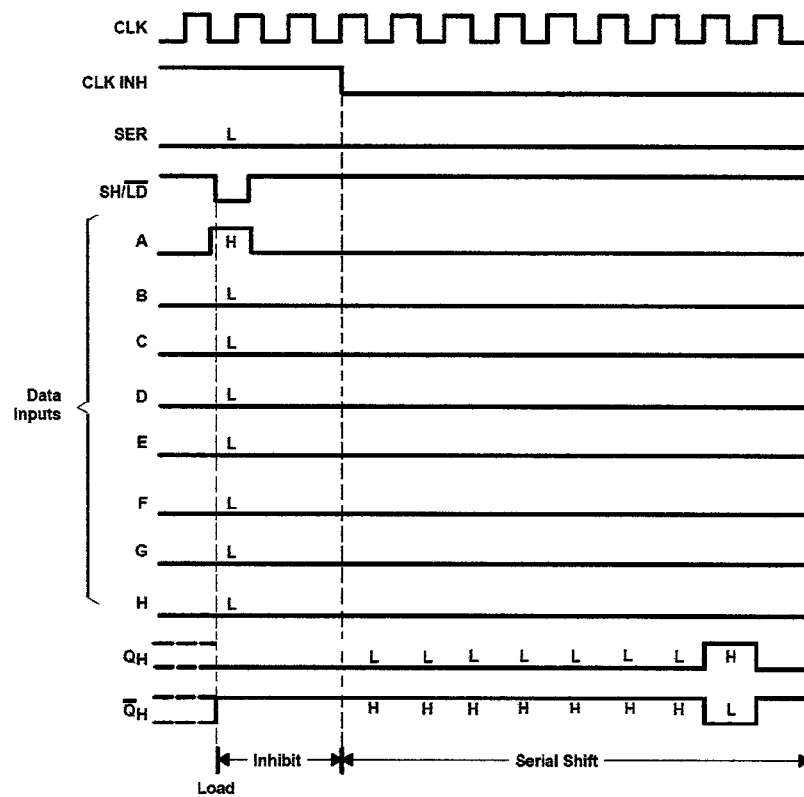
สัญญาณอินพุตที่รับเข้ามา จะรับเข้ามาแบบขนาน ( Parallel ) จึงต้องทำการแปลงเป็นแบบอนุกรม (Serial) โดยใช้ IC SN74HC165N ซึ่งมีคุณสมบัติ สามารถแปลงข้อมูลแบบขนาน 8 bit ขา 3,4,5,6,11,12,13,14 ให้เป็นแบบอนุกรม ขา 9 โดยใช้สัญญาณนาฬิกา ( CLK ) ขา 2 ในการเลื่อนตำแหน่งของข้อมูล และใช้สัญญาณจากไมโครคอนโทรลเลอร์ AVR ที่ส่งมาทางขา SH/LD ขา 1 สำหรับการใช้ในการกำหนดการโหลดข้อมูล โดยถ้าสัญญาณที่ขา SD/LD เป็น LOW จะเป็นการสั่งให้โหลดข้อมูลเข้าไปภายในตัวไอซี



FUNCTION TABLE

| INPUTS |     |         | FUNCTION      |
|--------|-----|---------|---------------|
| SH/LD  | CLK | CLK INH |               |
| L      | X   | X       | Parallel load |
| H      | H   | X       | No change     |
| H      | X   | H       | No change     |
| H      | L   | ↑       | Shift↑        |
| H      | ↑   | L       | Shift↑        |

รูปที่ 3.23 บล็อกไดอะแกรมของ IC SN74HC165N

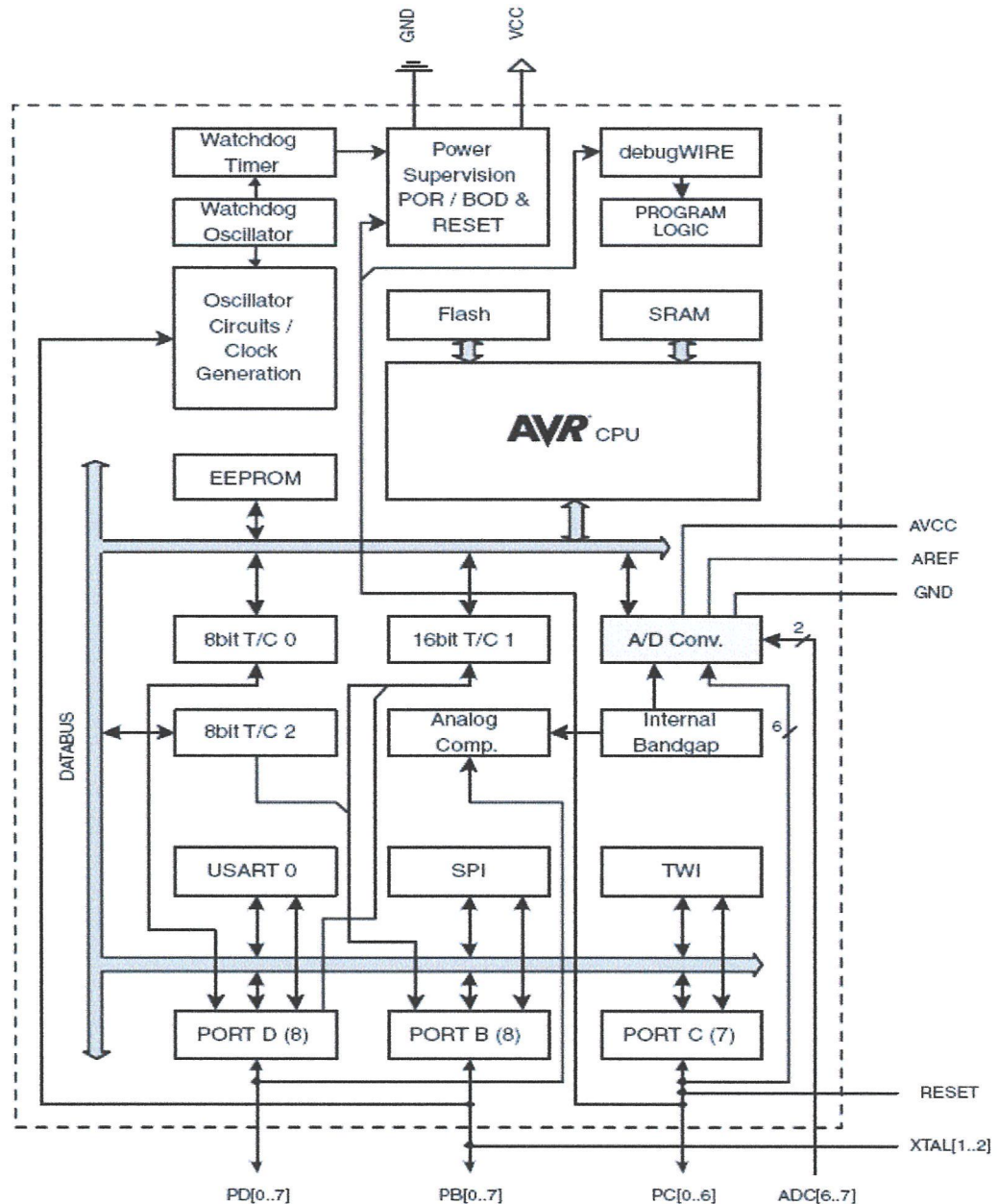


รูปที่ 3.24 ไคอะแกรมเวลาของการทำงาน IC SN74HC165N

การทำงานเริ่มจากให้ ขา A เป็นตัวรับข้อมูล ซึ่งโดยปกติทุกขาสัญญาณจะแสดงไว้เป็น “ LOW ” เมื่อมีอินพุตเข้ามาที่ขาใดขาหนึ่ง ในการสมมุติจะให้ขา A หรือ ขา 11 ทำหน้าที่ในการรับสัญญาณ โดยเมื่อได้มาขา A จะเปลี่ยนเป็น “ HIGH ” และ ส่งสัญญาณเข้ามาผ่าน AND GATE โดยการรับจะมี AND GATE 2 ตัว คือตัวแรกจะออกทางขา  $Q_H$  แสดงค่าที่ออกตามที่ได้รับมา และอีกตัวออกทางขา  $\overline{Q_H}$  ที่ต่ออยู่กับตัว INVERTER เพื่อแสดงค่าที่ตรงข้ามกับขา  $Q_H$  โดยจากความสัมพันธ์ของ AND GATE นำมาใช้ในการต่อกับ  $SH/LD$  โดยให้มีค่าเป็น “ HIGH ” เสมอเมื่อมีอินพุตเข้ามาที่ขาใดก็ตามจะมีการเปรียบเทียบกับภายใน AND GATE โดยถ้าตัวใดมีค่าเป็น “ HIGH ” AND GATE ก็จะส่งค่าลอจิกออกมาจากตัวนั้น เป็น “ 1 ” เช่น ขา A ได้รับอินพุตเข้ามาเข้า AND GATE และทำการเปรียบเทียบกับ  $SH/LD$  ปรากฏว่ามีค่าเป็น “ 1 ” ทั้งคู่ ทำให้ AND GATE ส่งค่าออกมาเป็น “ 1 ” ในทางตรงข้ามถ้าขา A ไม่มีอินพุตเข้ามา ค่าที่แสดงออกทางขา A จะเป็น “ LOW ” เมื่อนำมาเปรียบเทียบกับ  $SH/LD$  ที่มีค่าเป็น “ HIGH ” AND GATE ก็จะส่งค่าออกมาเป็น “ 0 ”

เมื่อ AND GATE ของขา A รับค่าแล้วจะส่งเข้าไปที่ SDR เพื่อทำการเข้าจังหวะทางสัญญาณ (Synchronize) โดย 1 cycle มีค่าเท่ากับ 20 ns โดยการรับค่าจะรับค่าทีละตัว คือรับค่าของ A เข้ามาไว้ใน SDR ตัวที่ 1 ( จากทางซ้ายมือของรูปที่ 3.23 ) เมื่อมีสัญญาณ CLK ที่ 2 เข้ามาในลูกถัดไปขา B รับค่าที่อินพุต ส่วน SDR ตัวที่ 1 ส่งค่าให้ SDR ตัวที่ 2 ต่อมาในสัญญาณ CLK ที่ 3 ขา B ส่งสัญญาณมาให้ SDR ตัวที่ 2 และสัญญาณของขา A จาก SDR ตัวที่ 2 จะส่งให้ SDR ตัวที่ 3 และขา C รับค่าจากอินพุต โดยทำเช่นนี้ ไปเรื่อยๆจนครบทั้ง 8 ขาและส่งค่าออกทาง  $Q_H$  หรือ  $\overline{Q_H}$  ดังแสดงรูปที่ 3.24

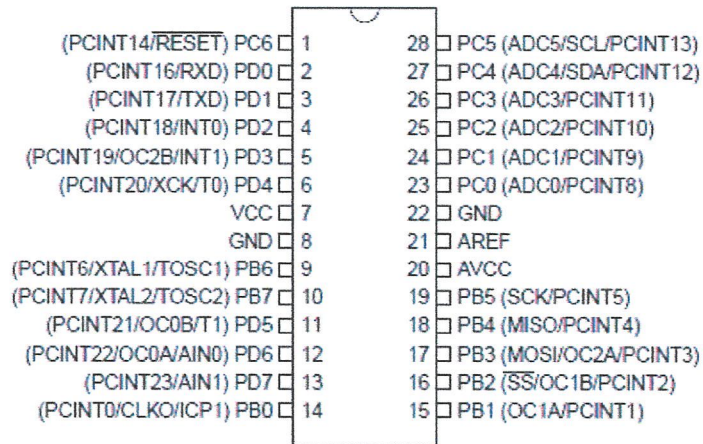
จากนั้นค่าจะถูกส่งออกไปยังไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์ ATMEGA168V ที่ใช้มีคุณสมบัติที่สำคัญ คือ มีความเร็วในการทำงาน 1 MIPS ต่อ 1 MHz ( สูงสุด 20 MIPS ต่อ 20 MHz ) ,หน่วยความจำ ROM แบบ Flash ขนาด 16 กิโลไบต์ ,หน่วยความจำข้อมูลแบบ EEPROM ขนาด 512 ไบต์ และหน่วยความจำข้อมูลแบบ SRAM ขนาด 1 กิโลไบต์ ซึ่งเพียงพอต่อการใช้งาน



รูปที่ 3.25 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ ATmega168V

เมื่อได้ข้อมูลแบบอนุกรม ( Serial ) แล้วข้อมูลจะส่งเข้าสู่ ไมโครคอนโทรลเลอร์ AVR ทาง Port D ( PD2 ) ซึ่งพอร์ตนี้อีกมีคุณสมบัติ คือ ทำหน้าที่เชื่อมต่อพอร์ตอนุกรม จากสัญญาณภายนอกจากนั้นไมโครคอนโทรลเลอร์ จะประมวลผลทีละ 8 บิต และตรวจสอบว่าสัญญาณอินพุตที่เข้ามามาจากช่องใดบ้าง และจึงทำการส่งข้อมูลตามที่ได้กำหนดไว้ในโปรแกรมที่เขียน ออกทางพอร์ต D ( PD1 ) เชื่อมต่อ

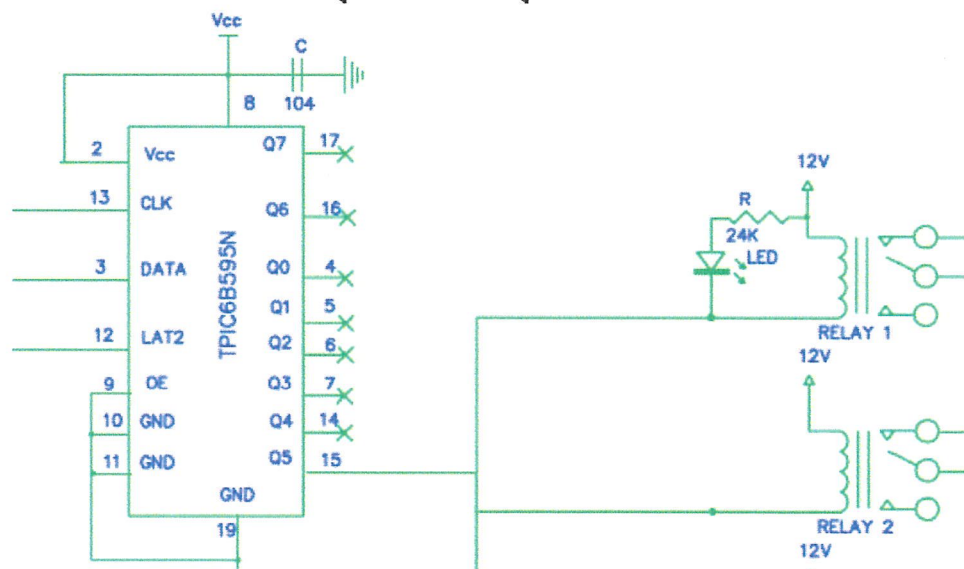
ผ่านทางสายคาลิงค์ ไปยังโทรศัพท์ และโทรศัพท์จะส่ง SMS แจ้งเตือนไปยังเลขหมายปลายทางที่กำหนดไว้และไม่โครคอนโทรลเลอร์จะส่งสัญญาณเพื่อสั่งให้สัญญาณเตือนภัยดัง ทางพอร์ต D ( PD6 ) ไปยัง IC TPIC6B595N โดยไอซีตัวนี้จะทำหน้าที่รับสัญญาณอินพุตแบบอนุกรม ( Serial ) ขา 3 และแปลงเป็นสัญญาณเอาต์พุตแบบขนาน ( Parallel ) ข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์ AVR จะเป็นแบบอนุกรม ดังนั้นข้อมูลแต่ละบิตจะผ่านเข้าไปในตัวไอซี โดยใช้สัญญาณนาฬิกา ( CLK ) ขา 13 ในการเลื่อนตำแหน่งของข้อมูล จากนั้นข้อมูลจึงส่งออกจากตัวไอซีแบบขนาน โดยที่ขา 15 จะต่ออยู่กับรีเลย์เพื่อใช้ในการเปิด-ปิดสัญญาณเตือนภัย



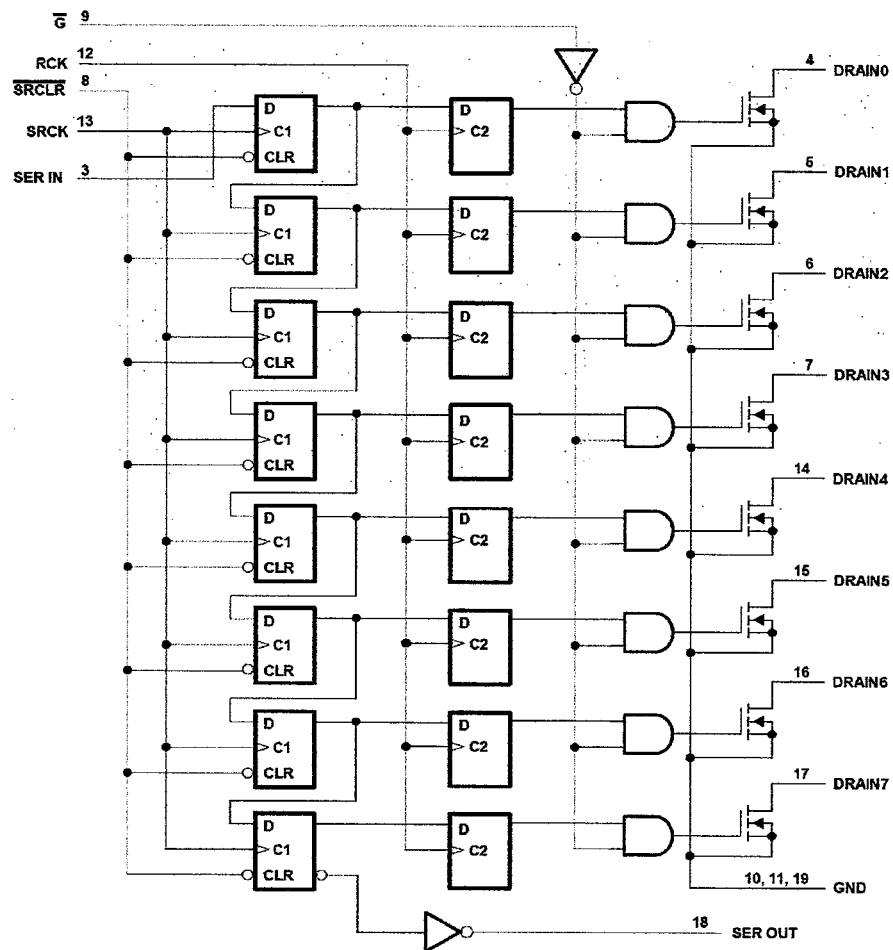
รูปที่ 3.26 แสดงรายละเอียดของขาไมโครคอนโทรลเลอร์

การตั้งค่าการทำงานต่างๆของไมโครคอนโทรลเลอร์ AVR กำหนดผ่านทาง พอร์ต บี เนื่องจากพอร์ตนี้นี้มีคุณสมบัติ คือ ทำหน้าที่โปรแกรมชิพ และป้อนสัญญาณนาฬิกาจากภายนอก ( PB6, PB7 ) การตั้งค่าการทำงาน ต่างๆกำหนดโดยโปรแกรมที่เขียนไว้ในตัวไมโครคอนโทรลเลอร์

### 3.3.3 การออกแบบด้านเอาต์พุตของวงจรควบคุม



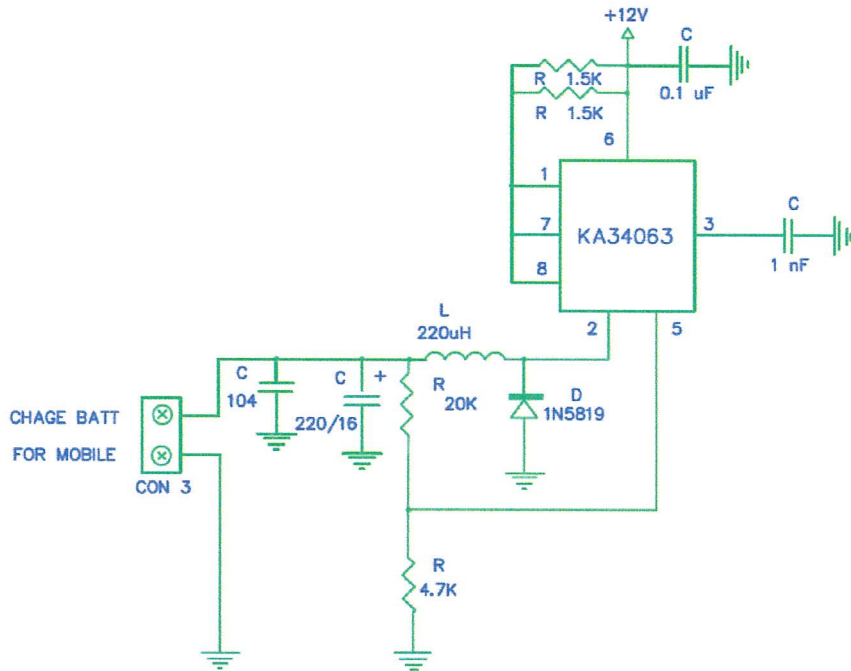
รูปที่ 3.27 วงจรแปลงข้อมูลจากแบบอนุกรมเป็นแบบขนาน



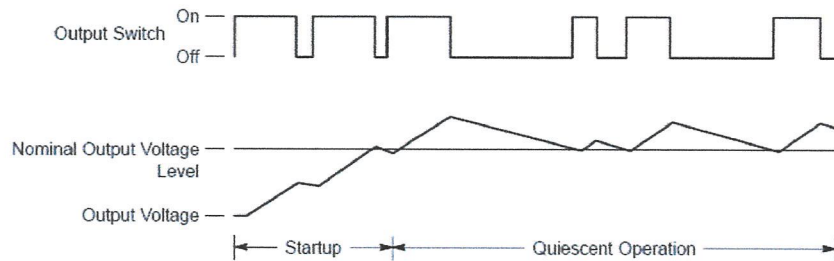
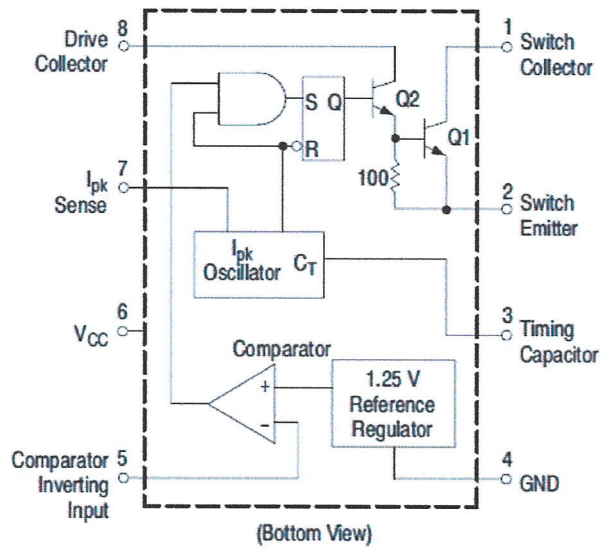
รูปที่ 3.28 บล็อกไดอะแกรมของ IC TPIC6B595N

การทำงานของ IC TPIC6B595N โดยเมื่อข้อมูลอินพุตเข้ามาที่ขา 12 โดยใช้สัญญาณ CLK ร่วมเมื่ออินพุตเข้ามาทางตัว C1 โดยสัญญาณ CLK 1 ลูก จะเลื่อนค่าเข้ามาที่ละตัวลงมาเรื่อยๆจนครบทั้ง 8 ตัว เมื่อ C1 รับค่าแล้วก็จะส่งออกเข้าที่ C2 โดยเอาที่พุตที่ออกจะไปยัง AND GATE โดยอีกขาของ AND GATE จะต่ออยู่กับ  $\bar{Q}$  โดยต่อผ่านตัว INVORTER ซึ่งทุกตัวจะให้ค่าทางลอจิกเป็น " 1 " เมื่อเอาที่พุตของ C2 ที่ออกมาเข้ามาเปรียบเทียบกับวงจรกับ AND GATE ถ้ามีค่าเป็น " 1 " เมื่อเปรียบเทียบกับลอจิกที่ออกจากตัว INVORTER ที่มีค่าเป็น " 1 " ผลที่ออกมาได้จะมีค่าเป็น " 1 " แต่ถ้าเอาที่พุตของ C2 มีค่าเป็น " 0 " เมื่อเปรียบเทียบกับ AND GATE แล้วค่าที่ออกมาจะมีค่าเป็น " 0 "

3.3.4 การออกแบบวงจรชาร์จแบตเตอรี่โทรศัพท์มือถือ



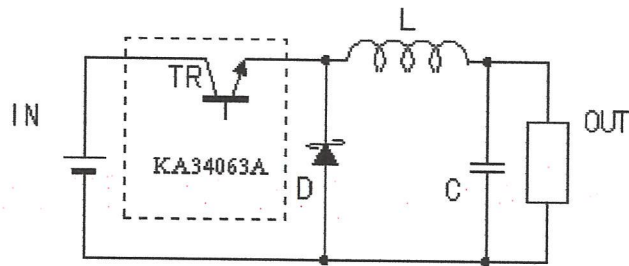
รูปที่ 3.29 วงจรชาร์จแบตเตอรี่โทรศัพท์มือถือ



รูปที่ 3.30 บล็อกไดอะแกรมและสัญญาณเอาต์พุตของ ไอซี KA34063A

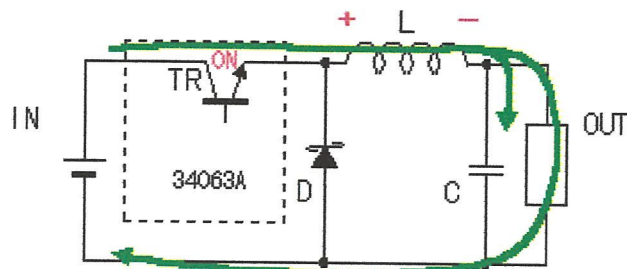
การชาร์จแบตเตอรี่ของโทรศัพท์มือถือ ใช้ IC KA34063A โดยทำงานแบบ Switching Regulator ขั้นตอนการลดระดับพลังงานที่ทำให้สวิตช์คาที่ออกจากเอาต์พุตซึ่งต่ำกว่า หรือสูงกว่าสวิตช์คาของอินพุตที่ป้อนเข้า จะสามารถทำได้ในระดับ +2V ถึง +10V โดยอินพุตที่ป้อนเข้าจะอยู่ในช่วง 3V ถึง 40V DC โดยในที่นี้จะให้อินพุตที่มีค่าเท่ากับ 12V และต้องการให้อาต์พุตออกมา 6V

ขั้นตอนการแปลงระดับความต่างศักย์



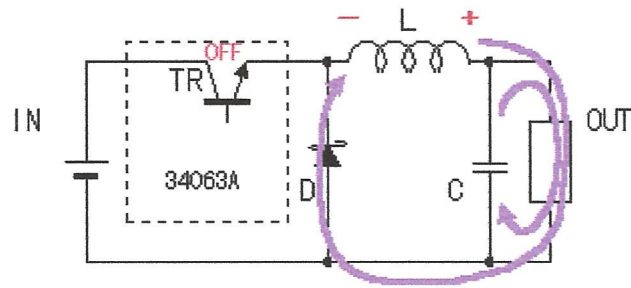
รูปที่ 3.31 วงจรการแปลงระดับความต่างศักย์

จากรูปที่ 3.31 คือวงจรแปลงระดับความต่างศักย์ อินพุตจะถูกปรับเปลี่ยนระดับพลังงานโดย Coil ที่ประกอบกับวงจร เอาต์พุตจะเก็บกำลังของอินพุตเมื่อ TR มีค่า “ ON ” เมื่อระดับพลังงานไฟฟ้าที่เก็บไว้ลดลง TR จะเปลี่ยนเป็น “ OFF ” เอาต์พุตจะรับพลังงานเข้ามา ตัวเก็บประจุที่ต่อคร่อมเอาต์พุตจะกระจายกำลังให้ได้ระดับที่ต้องการร่วมกับ Coil



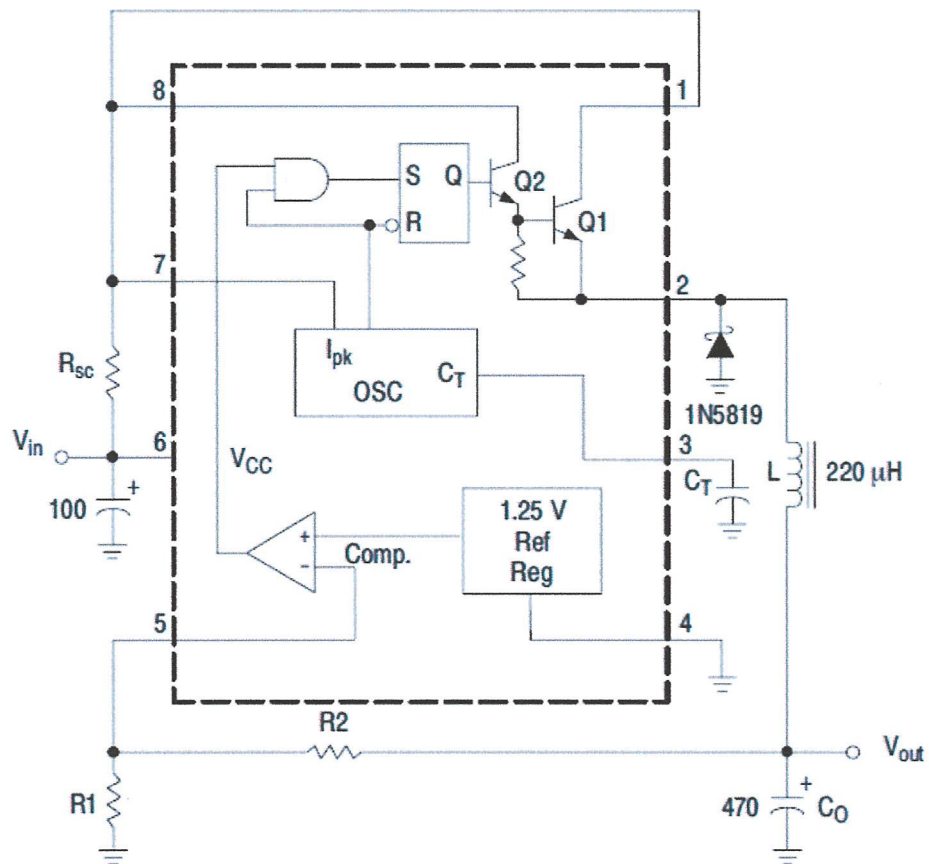
รูปที่ 3.32 แสดงการวนของกระแสในวงจรเมื่อ TR “ ON ”

เมื่อ TR “ ON ” พลังงานจากอินพุตจะเข้ามาประจุที่ตัวเก็บประจุ ซึ่ง Coil เป็นตัวส่งกำลังมาให้ ในเวลานี้พลังงานทั้งหมดจะถูกเก็บสะสมไว้ใน Coil และตัวเก็บประจุ ซึ่งในขณะเดียวกันสวิตช์คาทางอินพุตจะมีค่าเป็น + และทางเอาต์พุตจะมีค่าเป็น -



รูปที่ 3.33 แสดงการวนของกระแสในวงจรเมื่อ TR “OFF”

เมื่อ TR เริ่มจะมีค่าเป็น “OFF” Coil จะปล่อยพลังงานที่เก็บไว้โดยให้เกิดการปรับสมดุล ในเวลานี้อินพุตจะมีค่าเป็น - และเอาต์พุตจะมีค่าเป็น + โดยศักย์ค่าทางไฟฟ้าจะพยายามปรับเปลี่ยนให้เกิดการเลื่อนไหลของกระแสอยู่ตลอดเวลา พลังงานที่ถูกเก็บไว้ที่ Coil จะถูกส่งมายังไดโอด ในขณะเดียวกับพลังงานที่เก็บไว้กับตัวเก็บประจุก็สามารถที่จะปล่อยออกมาด้วยเช่นกัน หากไม่มีไดโอดพลังงานที่เก็บไว้จะไม่สามารถส่งออกมาได้



รูปที่ 3.34 การต่อของอุปกรณ์กับ IC KA34063A

การคำนวณ

โดย  $V_{out} = 1.25 \left( 1 + \frac{R_2}{R_1} \right)$  (สมการที่ 6)

ถ้าให้  $R_2 = 20 \text{ K}\Omega$  และต้องการให้  $V_{out} = 6 \text{ V}$

$$6.5 = 1.25 \left( 1 + \frac{20 \times 10^3}{R_2} \right)$$

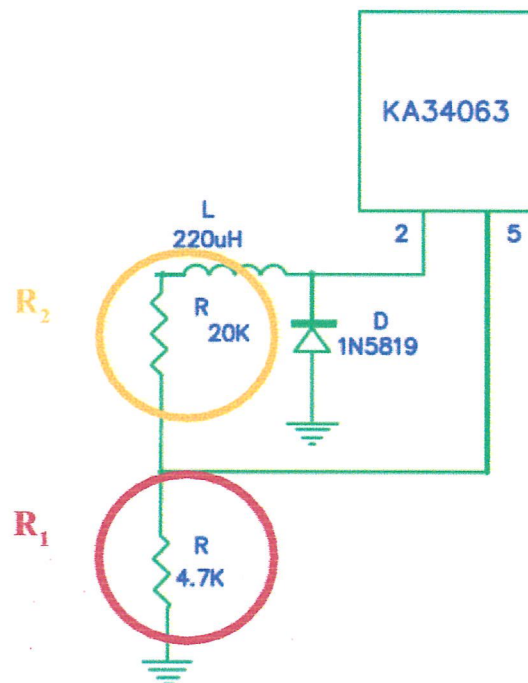
$$\frac{6.5}{1.25} = \left( 1 + \frac{20 \times 10^3}{R_2} \right)$$

$$5.2 - 1 = \frac{20 \times 10^3}{R_2}$$

$$R_2 = \frac{20 \times 10^3}{4.2}$$

$$R_2 = 4.7 \text{ K}\Omega$$

ดังนั้น  $R_1$  และ  $R_2$



รูปที่ 3.35 แสดงการวางอุปกรณ์ เพื่อให้ได้ค่า  $V_{out}$  ตามที่คำนวณไว้

การหาค่า  $R_{SC}$

จาก 
$$R_{SC} = \frac{0.3}{I_{PK(SWITCH)}} \quad \Omega \quad (\text{สมการที่ 7})$$

โดย 
$$I_{PK(SWITCH)} = 2 I_{OUT(MAX)} \quad A \quad (\text{สมการที่ 8})$$

ให้  $I_{OUT(MAX)}$  มีค่าเท่ากับ 0.1 mA

$$I_{PK(SWITCH)} = 2 I_{OUT(MAX)}$$

$$I_{PK(SWITCH)} = 2 \times (0.1 \times 10^{-3})$$

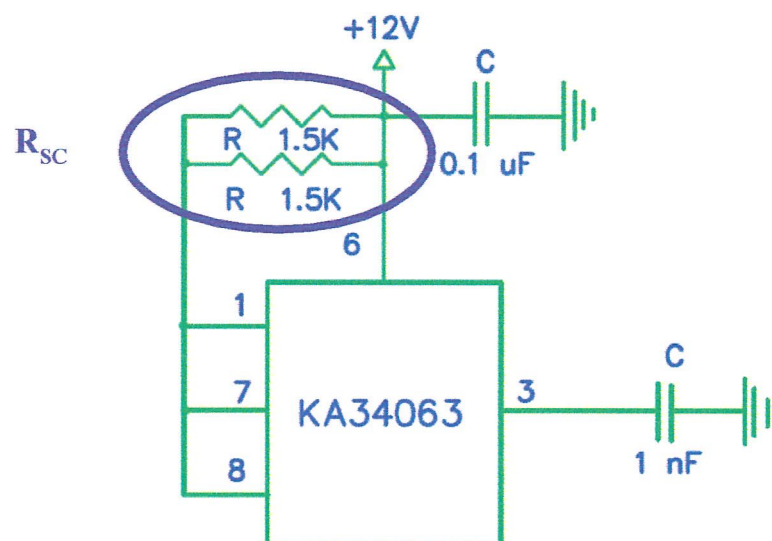
$$I_{PK(SWITCH)} = 0.2 \text{ mA}$$

ดังนั้น

$$R_{SC} = \frac{0.3}{I_{PK(SWITCH)}}$$

$$R_{SC} = \frac{0.3}{0.2 \times 10^{-3}}$$

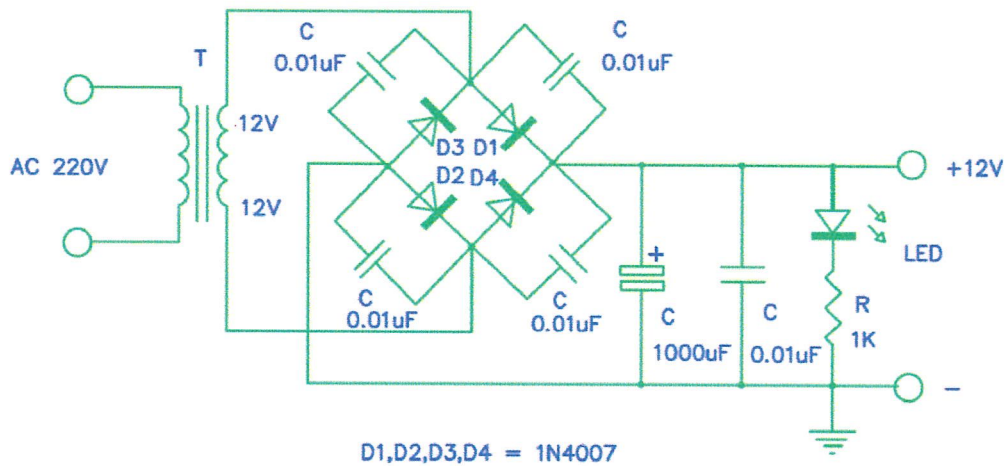
$$R_{SC} = 1.5 \text{ K } \Omega$$



รูปที่ 3.36 แสดงการวางอุปกรณ์  $R_{SC}$

### 3.3.5 แหล่งจ่ายไฟ

ไฟ AC 220 โวลต์ จะจ่ายเข้าวงจรระดับแรงดัน โดยใช้หม้อแปลงและไดโอด เป็น DC 12V 1A เพื่อใช้เป็นไฟเลี้ยง ไอซีและสัญญาณเตือนภัย

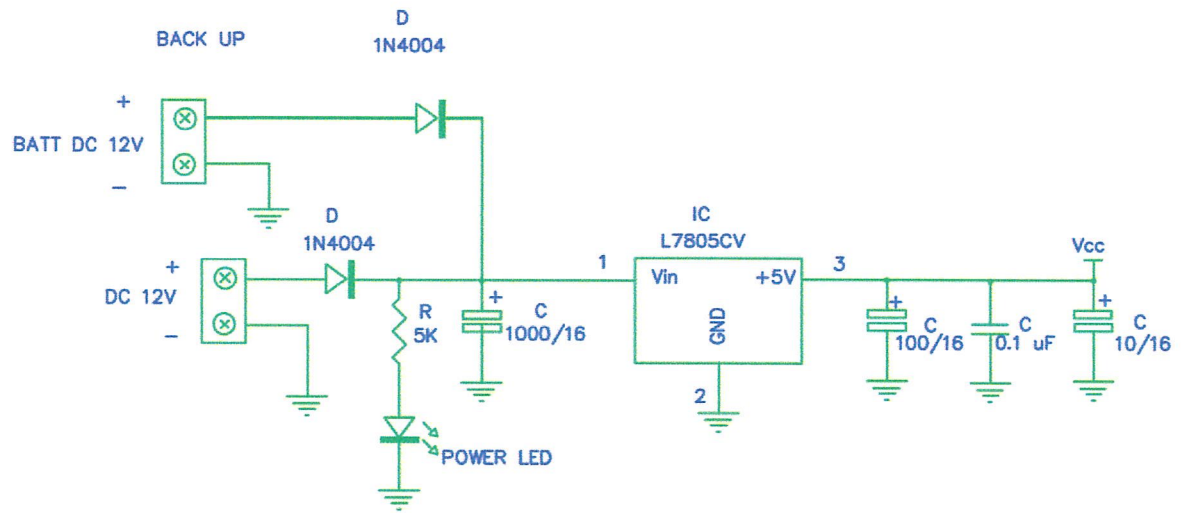


รูปที่ 3.37 วงจรแหล่งจ่ายไฟ

ไดโอด D1,D4 จะกรองกระแสทางด้านบวก และไดโอด D2,D3 จะกรองกระแสทางด้านลบ

เมื่อผ่านไดโอดแล้วตัวเก็บประจุจะเก็บและคายประจุ ทำให้กระแสเรียบขึ้นเพื่อส่งกระแสออกไปเลี้ยงวงจรในส่วนอื่นต่อไป เช่นเดียวกันถ้าหากว่าต้องการใช้ไฟตรงที่เรียงกระแสออกมาเรียบขึ้นเราก็ต้องใช้ตัวเก็บประจุค่ามากๆ มาเป็นวงจรกรองกระแส ยิ่งตัวเก็บประจุมีค่ามากการคายประจุจะมากขึ้น จึงทำให้ไฟกระแสตรงที่ออกมาเรียบที่สุด

### 3.3.6 วงจรลดระดับแรงดันและไฟสำรอง

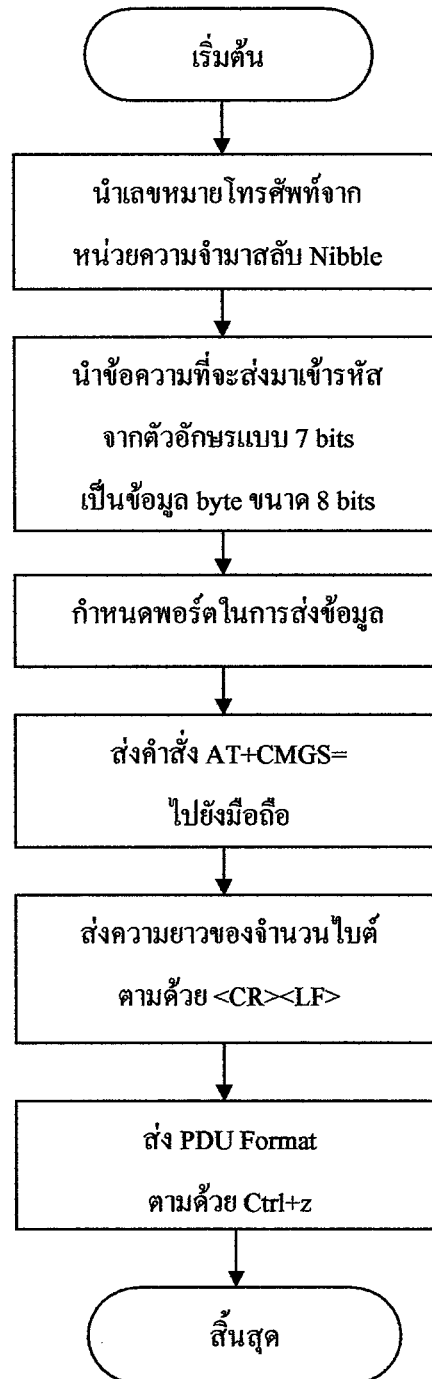


รูปที่ 3.38 วงจรลดระดับแรงดันและไฟสำรอง

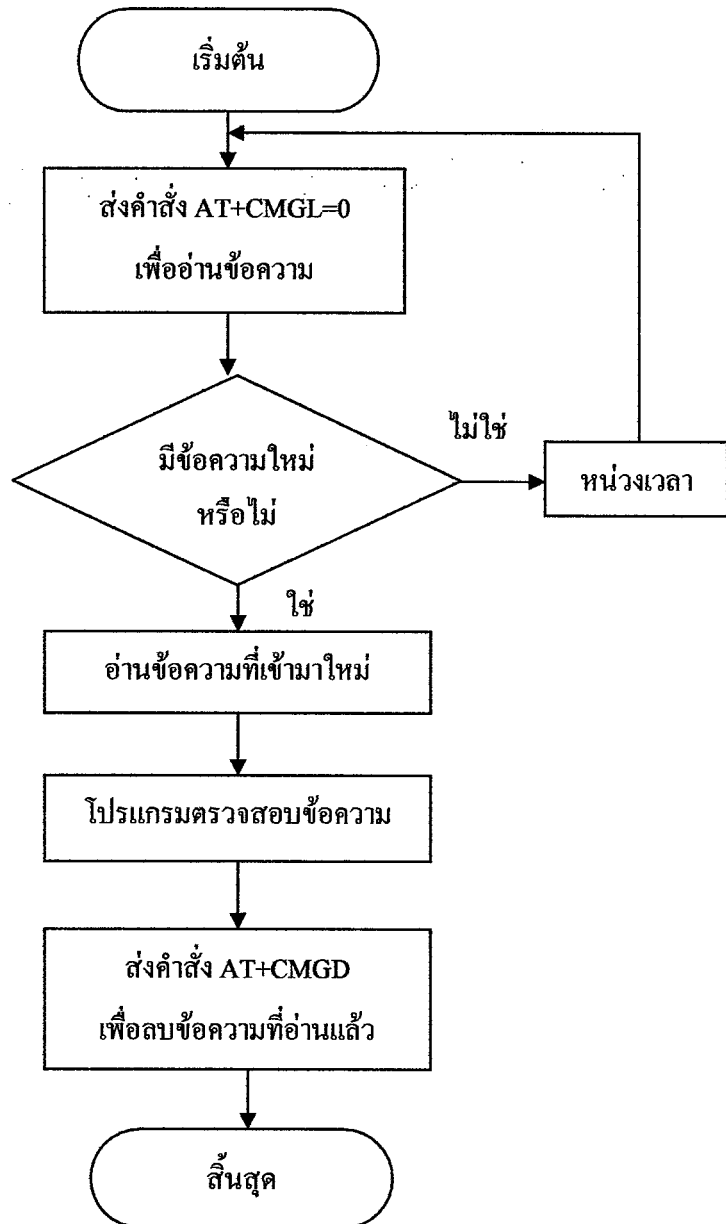
ไฟ DC 12V 1A จะจ่ายเข้าวงจรลดระดับแรงดัน โดยใช้ไอซี L7805CV แปลงเป็น ไฟ DC 5V เพื่อใช้เป็นไฟเลี้ยง ไอซีภาคควบคุมและภาครับสัญญาณ

โดยหลักการการทำงานจะรับไฟเข้าที่ 12V และผ่านเข้าที่ IC L7805CV ได้ไฟออกมา 5V และผ่านตัวเก็บประจุเพื่อกรองให้กระแสมีความมากเรียบขึ้น

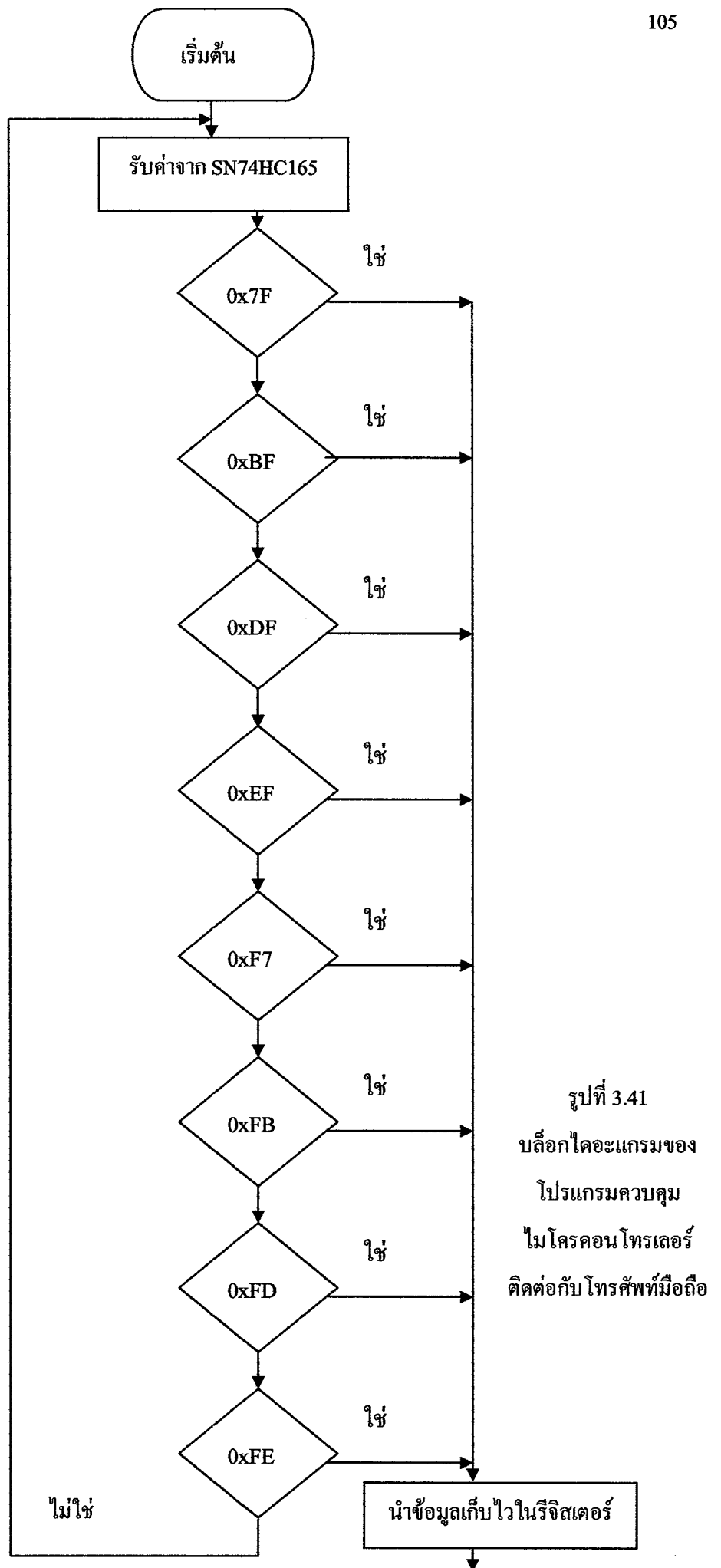
### 3.4 การทำงานของโปรแกรมควบคุม

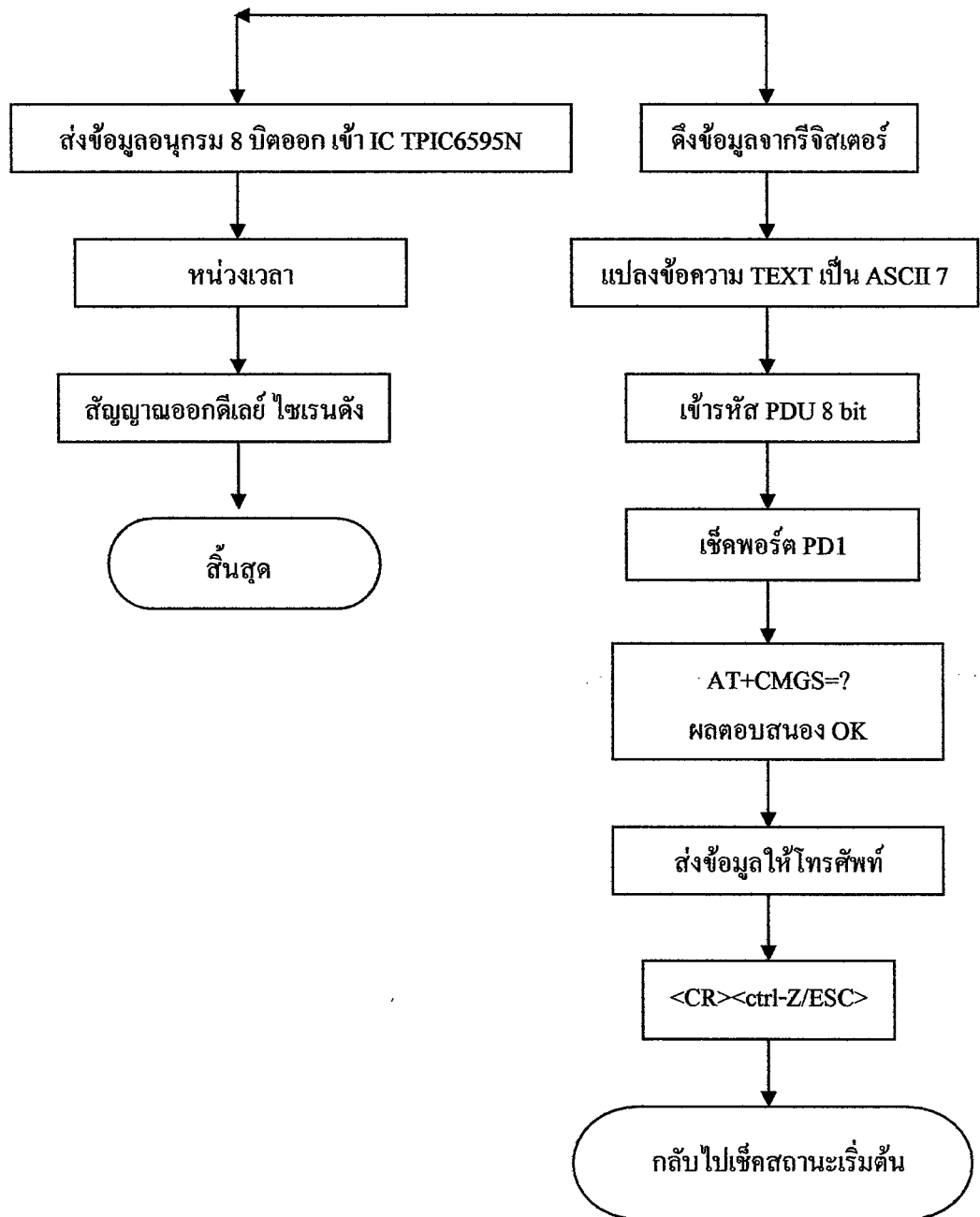


รูปที่ 3.39 บล็อกไดอะแกรมของส่วนควบคุมการส่งข้อความสั้น



รูปที่ 3.40 บล็อกไดอะแกรมของส่วนควบคุมการอ่านข้อความสั้น



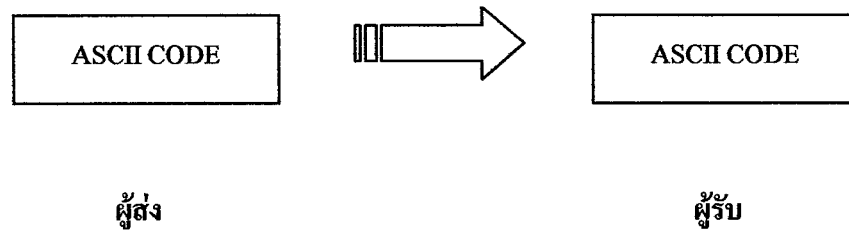


รูปที่ 3.42 บล็อกไดอะแกรมของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ติดต่อกับโทรศัพท์มือถือ (ต่อ)

### 3.5 โหมดการรับ - ส่งข้อมูล SMS

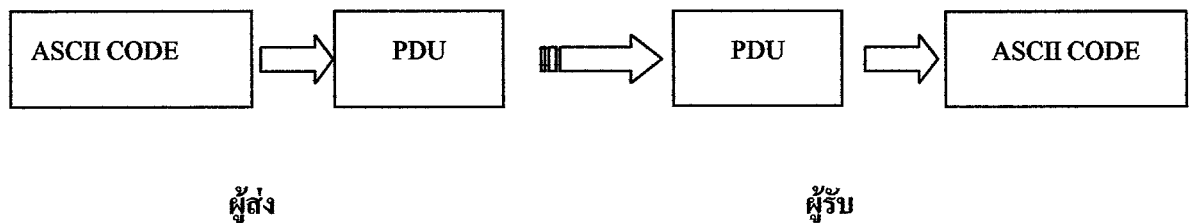
#### 3.5.1 รูปแบบทั่วไปของการส่ง SMS

1. **Text Mode** การส่งข้อมูลในรูปแบบของตัวอักษร โดยตรงซึ่งตัวเครื่องนั้นจะนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน แล้วจึงถูกส่งโดย PDU Mode ซึ่งตัวเครื่องโทรศัพท์ที่มีถือส่วนใหญ่ไม่รองรับการส่งข้อมูลในรูปแบบนี้ผ่านทาง **AT Command**



รูปที่ 3.43 แสดงการรับ - ส่งข้อความแบบ Text Mode

2. **PDU Mode ( Protocol Description Unit Mode )** การส่งข้อมูลในรูปแบบหนึ่งที่ต้องมีการนำเอาข้อมูลดิจิทัลมาทำการเข้ารหัสก่อนเพื่อใช้ในการส่งข้อมูลผ่านทาง Air Interface



รูปที่ 3.44 แสดงการรับ - ส่งข้อความแบบ PDU Mode

#### 3.5.2 การส่งข้อความ SMS ในทีดียูโหมด

PDU MODE คือ รูปแบบของข้อมูลที่ใช้ในการส่งข้อความสั้นซึ่งเกิดจากการนำเอาข้อมูลดิจิทัลมาเข้ารหัสเพื่อใช้ในการส่งข้อมูลผ่านทาง AIR Interface โดยการรับข้อมูลและส่งข้อมูล SMS แบบ PDU Mode จะมีโครงสร้างที่แตกต่างกัน ในส่วนการส่งข้อความ SMS ในทีดียูโหมดนั้น การส่งงานจะส่งผ่านทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่โดยส่งเป็นชุดคำสั่ง **AT Command** ซึ่งชุดคำสั่งที่ใช้เป็นดังนี้

**AT+CMGF** เป็นคำสั่งในการเลือกโหมดของข้อความที่จะส่ง

**AT+CMGS** เป็นคำสั่งสำหรับส่งข้อความไปยัง Address ที่เลือกไว้

การส่ง SMS นั้นเราจะใช้ไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ในการส่ง โดยจะส่งชุดคำสั่ง **AT Command** เข้าไปทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่ แต่ก่อนที่จะส่งนั้นเราต้องทำการกำหนดอัตราในการรับส่งข้อมูล (Baud Rate) ของไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ให้ตรงกับอัตราการรับส่งข้อมูลของ โทรศัพท์เคลื่อนที่เสียก่อน

### 3.5.3 การแปลงข้อมูลแอสกีโค้ดเป็นทีดียูโค้ด (ASCII Code To PDU Code)

การแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ( Octet ) โดยข้อความจากตารางจะเป็นส่วนที่เราสามารถใส่รหัสของข้อความที่ต้องการส่ง แต่เนื่องจากเราไม่สามารถนำรหัสของตัวอักษรแบบ 7 บิต ไปได้โดยตรงจำเป็นต้องผ่านการแปลงให้เป็นรหัสข้อมูลแบบ 8 บิตก่อน โดยในการส่งข้อความนี้เป็นการส่งข้อความจากเครื่องส่งที่เกิดเหตุไปยังเครื่องรับด้วยคำว่า **#01>Zone1** ซึ่งสามารถบ่งบอกถึงเครื่องรับแล้วสถานที่เกิดเหตุในแต่ละส่วนที่แบ่งไว้ได้ 9 ตัวอักษรซึ่งแต่ละตัวอักษรเป็นชนิด 7 บิตให้เป็นข้อมูล 8 บิต สำหรับใช้ในการส่ง SMS การแปลงเริ่มจากการนำรหัส 7 บิตของตัวอักษรแรก ( # ) มาเติมข้างหน้าด้วย 0 จากบิตสุดท้ายของอักษรตัวที่ 2 ( 0 ) จะได้ผลลัพธ์ 8 บิต ( 1 ไบต์ ) เป็น "23" ขั้นตอนต่อมาให้เอา 6 บิตที่เหลือของอักษรตัวที่ 2 มาเติมข้างหน้าด้วย 2 บิตท้ายของรหัส 7 บิตของอักษรตัวที่ 3 ( 1 ) จะได้ผลลัพธ์ 8 บิตเป็น "40" และทำเช่นนี้ไปเรื่อยๆ โดยจำนวนที่นำมากระทำจะเพิ่มขึ้นเป็น 3 บิต จนกระทั่งถึง 7 บิต แล้วเริ่มกระบวนการใหม่จนกระทั่งหมดชุดตัวอักษร หลังจากการแปลงข้อความ **"#01>Zone1 "** จะได้ข้อมูลเป็นเลขฐาน 16 จำนวน 8 ไบต์เป็น 23 40 C0 A7 7D BB CB 1 โดยวิธีการแปลง จะแสดงที่ตาราง โดยที่ตัวอักษรชนิด 7 บิต ถูกกำหนดโดยมาตรฐาน GSM 03.38 ดังตารางที่ 3.1

ตัวอย่างการแปลงเลขฐานสองเป็นเลขพีคียูโค้ดด้วยข้อความ “#01>Zone1 ”

| เลขฐานสิบหก<br>(แอสกีโค้ด) | เลขฐานสอง<br>(แอสกีโค้ด) | เลขฐานสอง(พีคียู) | พีคียูโค้ด | ข้อความ |
|----------------------------|--------------------------|-------------------|------------|---------|
| 23                         | 00100011                 | 0100011           | 35         | #       |
| 40                         | 01000000                 | 0000000           | 0          | 0       |
| C0                         | 11000000                 | 0000001           | 1          | 1       |
| A7                         | 10100111                 | 0111110           | 62         | >       |
| 7D                         | 01111101                 | 1011010           | 90         | Z       |
| BB                         | 10111011                 | 1101111           | 111        | o       |
| CB                         | 11001011                 | 1101110           | 110        | n       |
| 1                          | 00000001                 | 1100101           | 101        | e       |
|                            | -                        | 0000001           | 1          | 1       |

ตารางที่ 3.1 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ด้วยข้อความ “#01>Zone1 ”

| Dec |     | 0  | 16    | 32 | 48 | 64 | 80 | 96 | 112 |
|-----|-----|----|-------|----|----|----|----|----|-----|
|     | Hex | 0  | 10    | 20 | 30 | 40 | 50 | 60 | 70  |
| 0   | 0   | @  | Δ     | SP | 0  | i  | P  |    | p   |
| 1   | 1   | £  | _     | !  | 1  | A  | Q  | a  | q   |
| 2   | 2   | \$ | €     | "  | 2  | B  | R  | b  | r   |
| 3   | 3   | ¥  | l     | #  | 3  | C  | S  | c  | s   |
| 4   | 4   | è  | l     | □  | 4  | D  | T  | d  | t   |
| 5   | 5   | é  | Ω     | %  | 5  | E  | U  | e  | u   |
| 6   | 6   | ù  | Π     | &  | 6  | F  | V  | f  | v   |
| 7   | 7   | ì  | α     | '  | 7  | G  | W  | g  | w   |
| 8   | 8   | ò  | ~     | v  | 8  | H  | X  | h  | x   |
| 9   | 9   | ç  | ß     | )  | 9  | I  | Y  | i  | y   |
| 10  | A   | LF | P     | *  | :  | J  | Z  | j  | z   |
| 11  | B   | Ø  | <ESC> | +  | ;  | K  | Ä  | k  | ä   |
| 12  | C   | ø  | ¢     | ,  | <  | L  | Ö  | l  | ö   |
| 13  | D   | CR | Æ     | -  | =  | M  | Ñ  | m  | ñ   |
| 14  | E   | Å  | £     | .  | >  | N  | Ü  | n  | ü   |
| 15  | F   | å  | É     | /  | ?  | O  | §  | o  | à   |

ตารางที่ 3.2 ชุดของตัวแปรของมาตรฐาน GSM 03.38

### 3.5.4 การส่ง SMS

ส่วน ควบคุมการส่ง SMS นี้สามารถศึกษาได้ตามตัวอย่างการส่ง SMS ในรูปแบบของ PDU Mode ดังต่อไปนี้

ตัวอย่างนี้ใช้คอมพิวเตอร์ส่งคำสั่ง AT Command ให้โทรศัพท์เคลื่อนที่ทำการส่งข้อความ

AT+CMGF=0 “เซ็ต PDU Mode”

OK “โทรศัพท์เคลื่อนที่ตอบกลับมาว่า “OK”

AT+CMGS=22 “ส่ง SMS โดยความยาว TPDU = 22 Octets”

0011000B916638292021F30000AA092340C0A77DBBCB1 เมื่อพิมพ์ข้อความครบแล้วกด ctrl+z

ส่วนประกอบของข้อมูลที่ส่งอธิบายไว้ในตารางที่ 3.3

ส่วนประกอบของข้อมูลที่ส่งคือ 0011000A9166930212320000AA092340C0A77DBBCB1

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด  |
|---------------------------|---|
| 00                        | ความยาวของ SMSC Information โดยที่ 00 หมายถึง ให้ใช้ SMSC Information ที่เก็บอยู่ภายในเครื่อง (ปกติแล้วเครื่องที่สามารถส่ง SMS ได้มีข้อมูล SMSC ภายในเครื่องอยู่แล้ว) |
| 11                        | First octet of the SMS-SUBMIT message   |
| 00                        | TP-message-Reference โดยที่ 00 คือ ให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง   |
| 03                        | Address-Length คือความยาวของเลขหมายผู้รับ (10 ตัว)  |
| 91                        | ใช้แสดงว่าเบอร์โทรศัพท์เคลื่อนที่เป็นแบบภายในประเทศ   |
| 66 38 29 20 21 F3         | เบอร์โทรศัพท์เคลื่อนที่ปลายทางที่ทำการเข้ารหัสแบบสลับแล้ว โดยหมายเลขที่แท้จริงคือ +66 39202123  |
| 00                        | TP-PID (Protocol identifier) ในกรณีนี้เป็น 00   |
| 00                        | TP-DCS (Data coding scheme) เป็น 00 คือเข้ารหัสข้อความแบบ 7 บิต Default Alphabet  |
| AA                        | TP-Validity-Period โดยที่ AA หมายถึง ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทาง ข้อความจะถูกยกเลิกโดยอัตโนมัติ                            |
| 09                        | TP-User-Data-Length คือความยาวของข้อความก่อนแปลงรหัส นั่นคือ "#01>Zone1" = 9 ตัวอักษร   |
| 2340C0A77DBBCB1           | TP-User-Data เป็นข้อมูลในส่วนของข้อความตัวอักษรแบบ 7 บิต นั่นคือคำว่า "#01>Zone1" ที่ผ่านการแปลง (เข้ารหัส) เป็นข้อมูลแบบ 8 บิต แล้ว                                  |

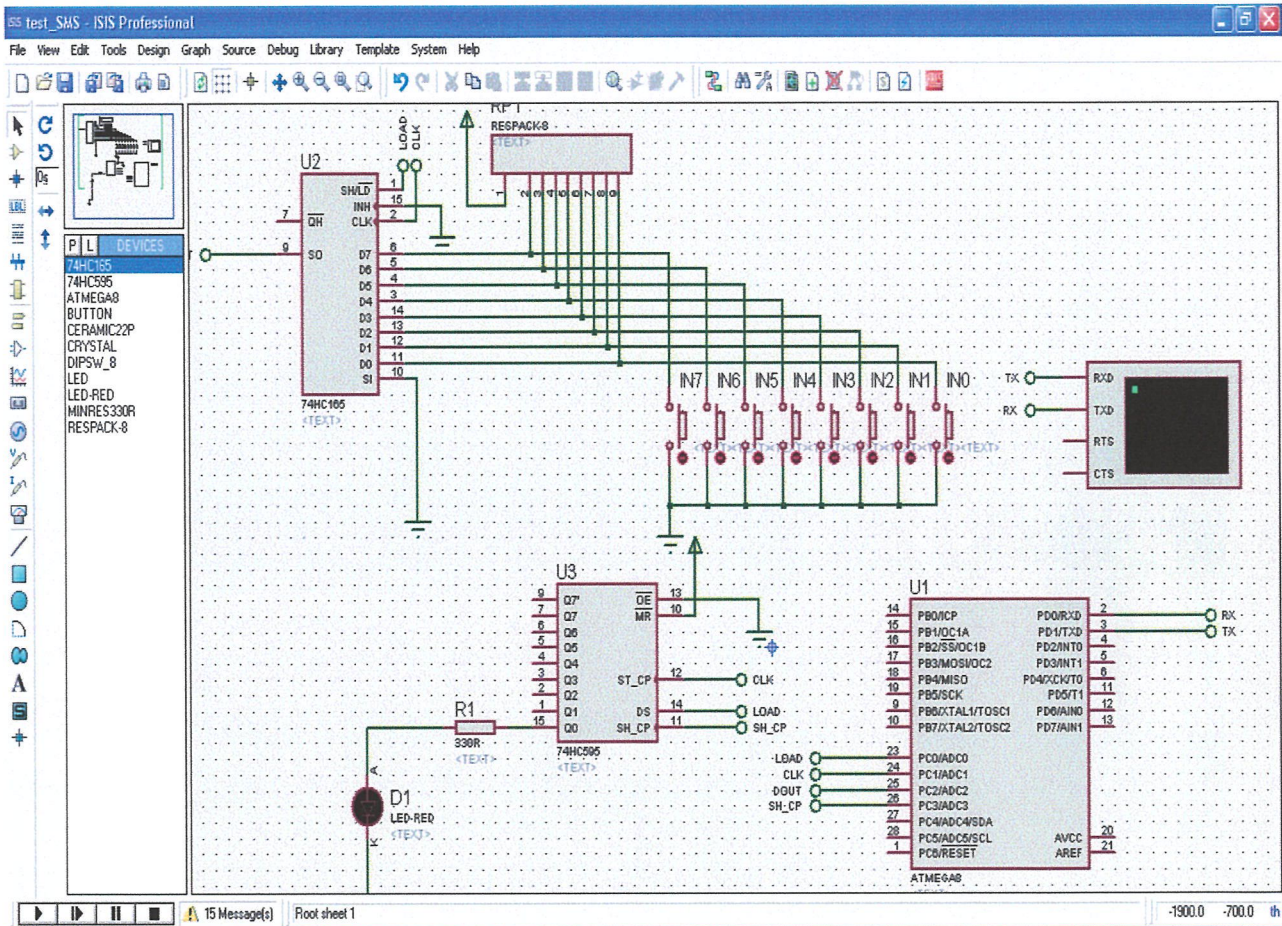
ตารางที่ 3.3 แสดงส่วนประกอบของข้อมูลที่ส่งข้อความสั้นภาษาอังกฤษของคำว่า "#01>Zone1"

## บทที่ 4

### การทดลองและผลการทดลอง

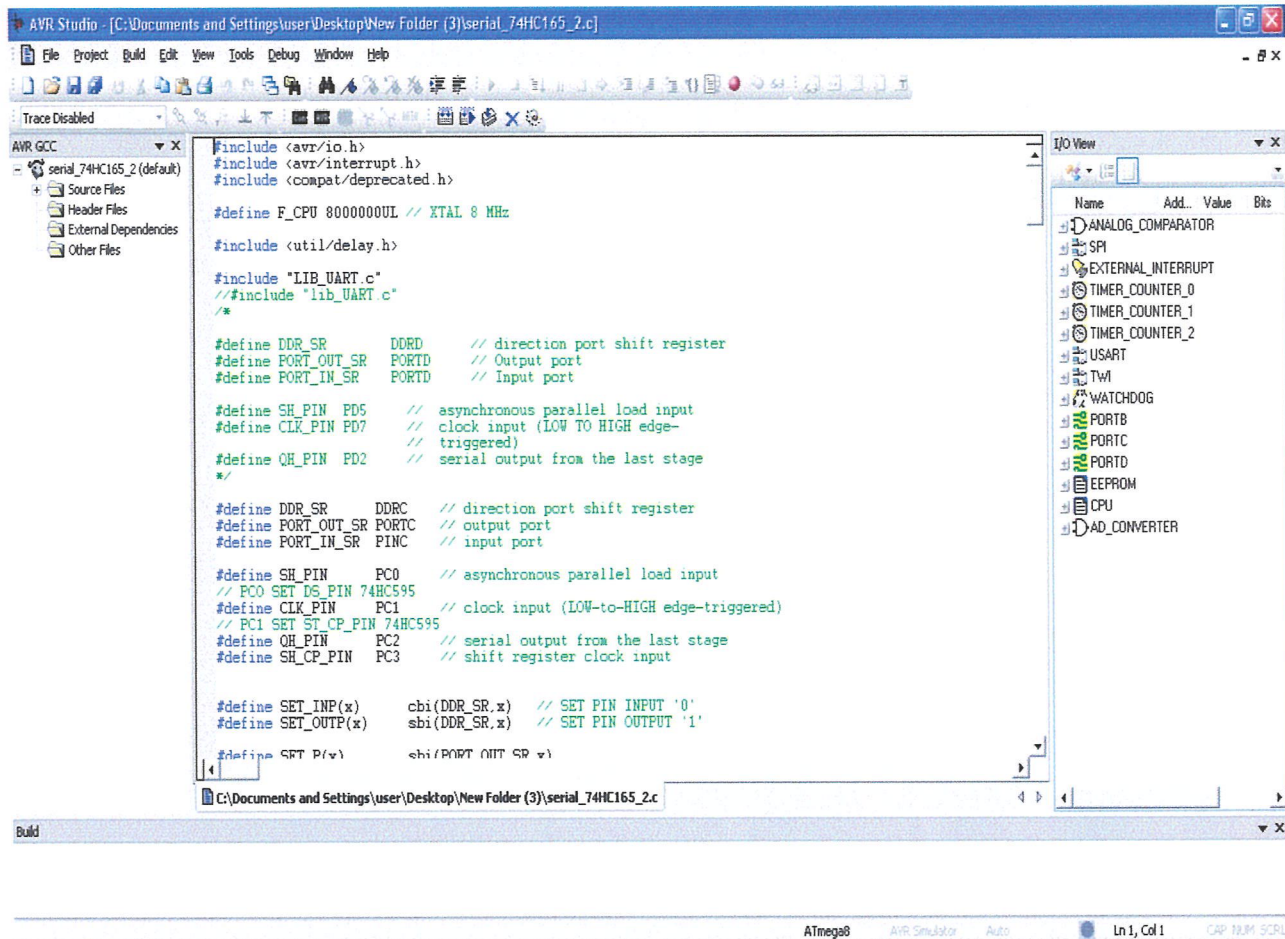
#### 4.1 การทดสอบวงจรโดยโปรแกรม Proteus 7 Professional

การทดลองในส่วนนี้จะเป็นส่วนเริ่มแรกของการศึกษา ออกแบบ และทดสอบระบบการทำงานของไมโครคอนโทรลเลอร์ ด้วยโปรแกรม Proteus ในการจำลองวงจรภายในโปรแกรมที่ได้เขียนขึ้นมาดังแสดงในรูปที่ 4.1



รูปที่ 4.1 แบบจำลองการทำงานของวงจร ภายใน โปรแกรม Proteus

## โดยการทดลองเริ่มจากการเขียน Source Code ของตัวไมโครคอนโทรลเลอร์ AVR ด้วยโปรแกรม AVR ดังรูปที่ 4.2

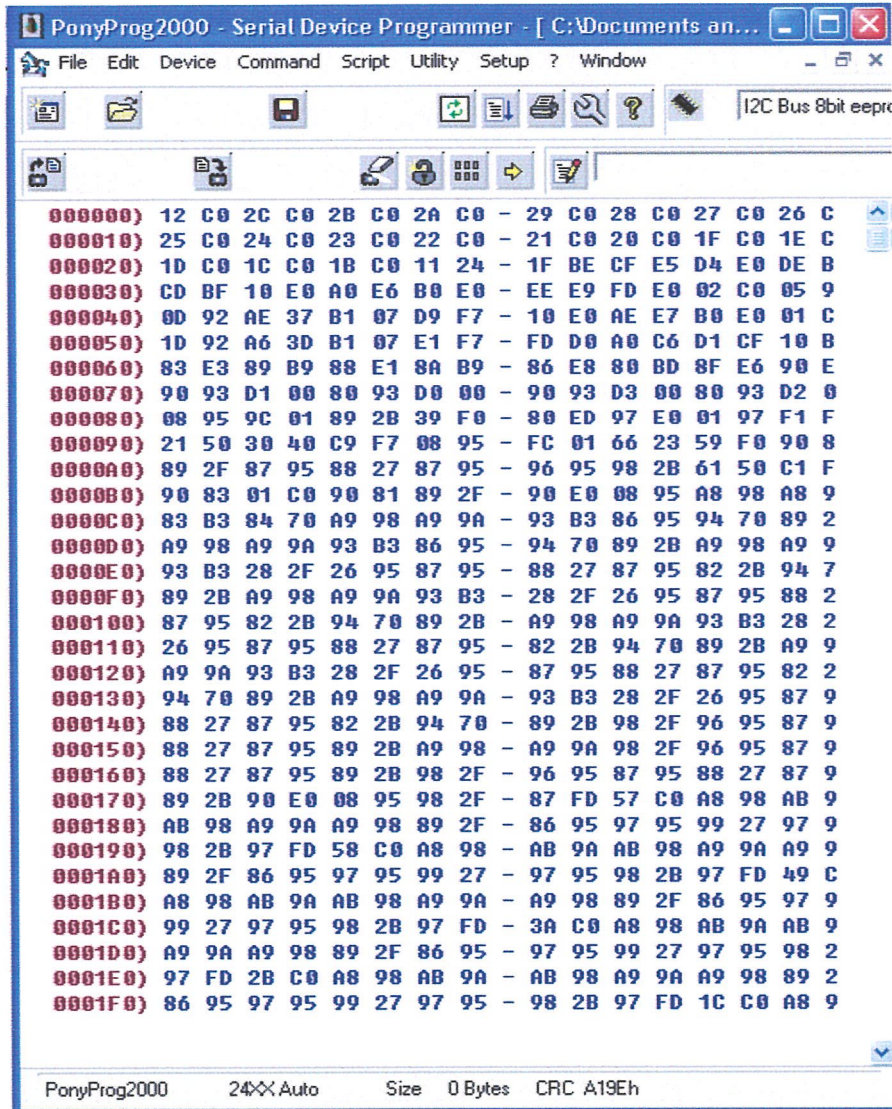


รูปที่ 4.2 แสดง Source Code ที่เขียนบนโปรแกรม AVR Studio

เมื่อ Source Code ที่เขียนสามารถ Compiled ผ่านจะได้ไฟล์ที่แปลงออกมาในรูปของไฟล์ .hex ซึ่งสามารถที่จะนำไปใช้กับตัวไมโครคอนโทรลเลอร์ได้ โดยเมื่อทำการ Compiled แล้ว Source Code ที่เขียนจะถูกแปลงให้เป็นเลขฐาน 16 โดยใช้โปรแกรม PonyProg 2000 ซึ่งเป็นโปรแกรมที่ออกแบบมาสำหรับการใส่โปรแกรม .hex ที่เขียนเข้าสู่ไมโครคอนโทรลเลอร์ AVR



รูปที่ 4.3 ไฟล์ .hex ของ Source Code ที่ Compiled

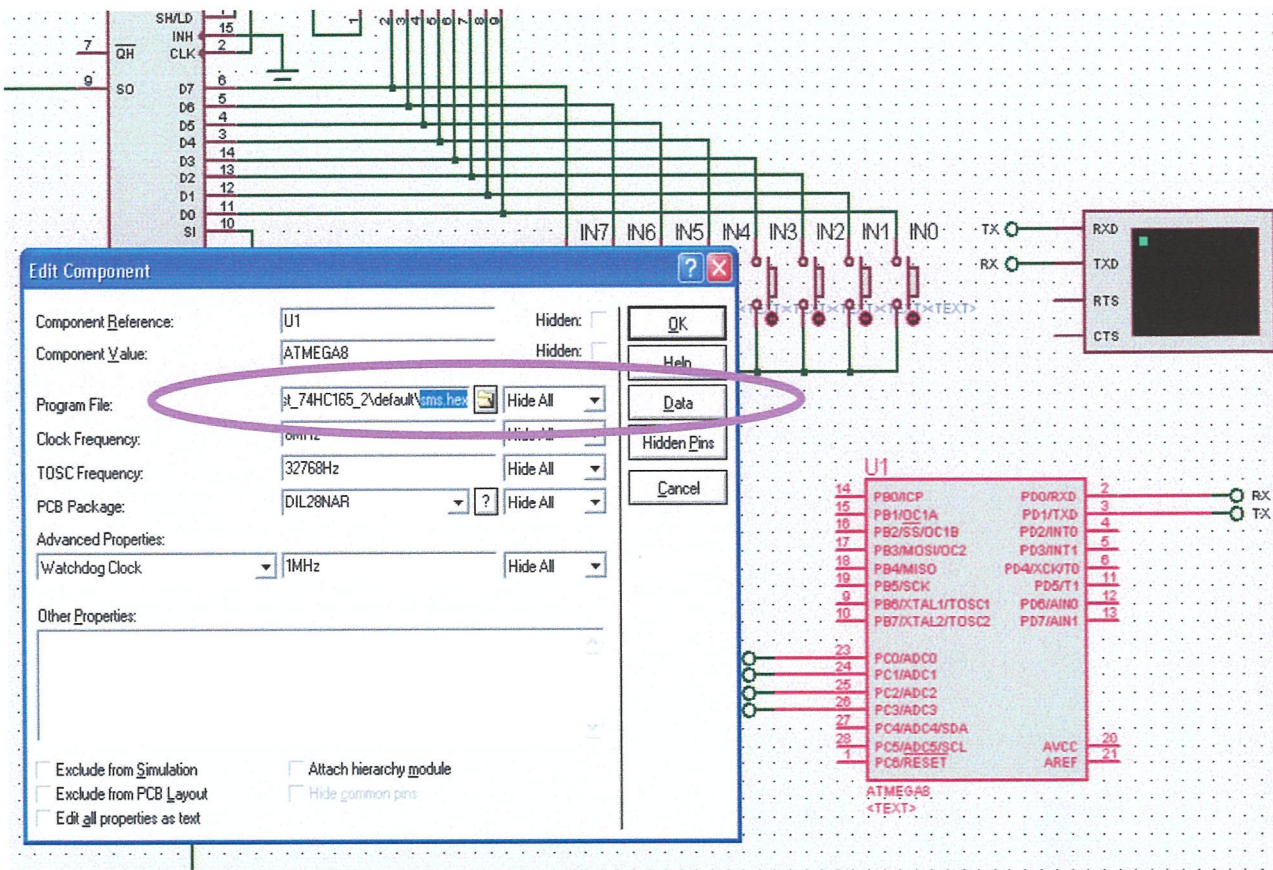


รูปที่ 4.4 โปรแกรม PonyProg 2000 ที่แสดงข้อมูลเลขฐาน 16 ที่ทำการ Compiled จากภาษา C

โปรแกรม Proteus ก็เปรียบเสมือนการทดลองการทำงานของวงจรที่เขียนขึ้นมานั้น สามารถทดสอบโดยการใส่ค่า .hex ลงไปในไมโครคอนโทรเลอร์ AVR ที่จำลองขึ้น โดย

คลิกขวา > Edit Component > Program file > เลือกไฟล์ .hex ที่ต้องการ

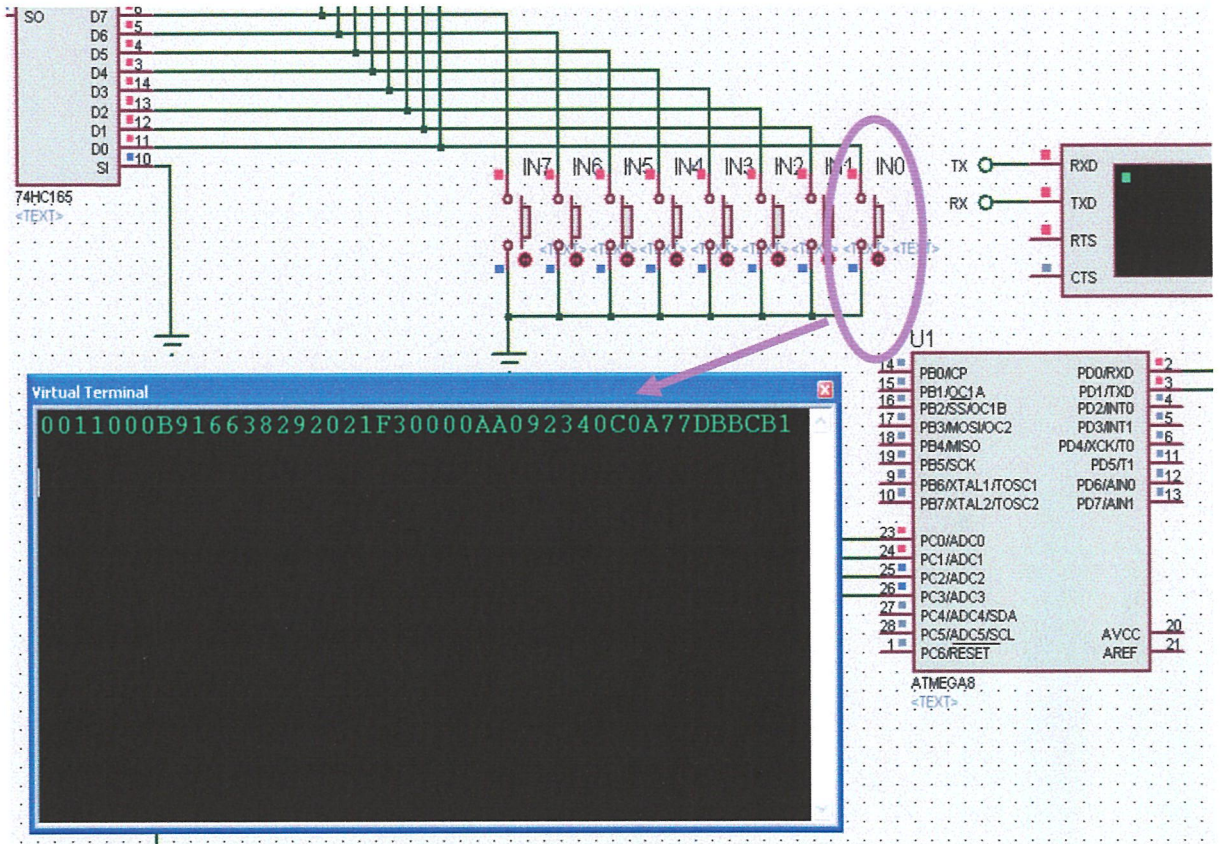
ซึ่งข้อดีของ โปรแกรมนี้คือผู้ออกแบบไม่จำเป็นต้องเขียนโปรแกรมลงในตัวไมโครคอนโทรเลอร์ และสามารถแก้ไขใหม่ได้เมื่อเกิดข้อผิดพลาด ทำให้เกิดความสะดวกและยังช่วยในเรื่องของความแม่นยำในผลการทดลองที่ได้อีกด้วย



รูปที่ 4.5 แสดงการใส่โปรแกรมลงในวงจรจำลองของโปรแกรม Proteus

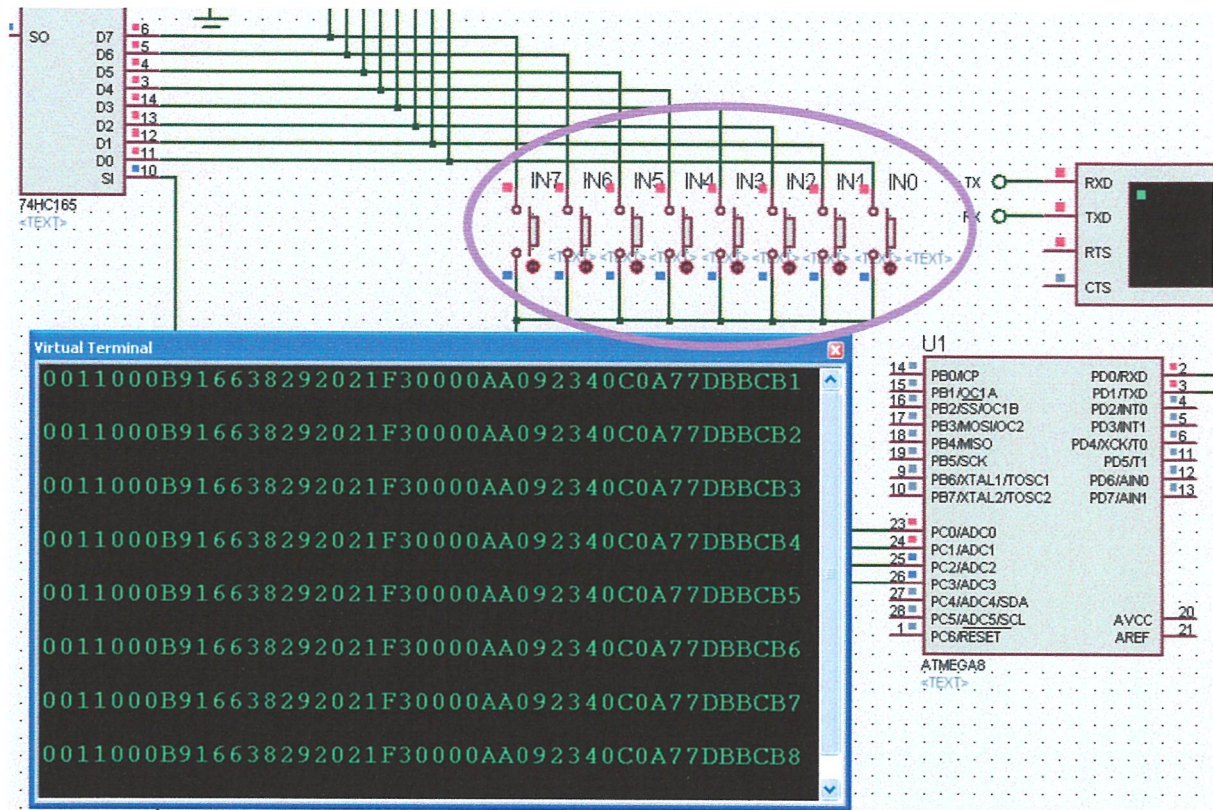
จากนั้นจะเป็นการแสดงผลของการจำลองในโหมดของ Animating โดยการทดลองในโหมดนี้จะเป็นการจำลองการทำงานเริ่มจากการรับค่า Input จากภายนอกเข้ามาที่วงจรและผ่าน IC 74HC165 ซึ่งรับค่าที่มีการส่งเข้ามาจากขาทุกตัวนั้นจะเป็น 1 (เป็นค่า High ) โดยทั้ง 8 ขา เมื่อไม่มี Input เข้ามาสัญญาณจะออกแบบในทางลจิกคือ 11111111 หรือ FF เมื่อมีสัญญาณเข้ามา (เกิดการลัดวงจรในขา 1) ทำให้มี Input และจะทำให้ค่าลจิกในส่วนนั้นที่เป็น 1 (high) ลงมาเป็น 0 (Low) เช่น ในวงจรจำลอง มี Input เข้าที่ขา IN0 (ขาที่ 1) ของวงจรถ่ายค่าทางลจิกทั้ง 8 ขาจะเปลี่ยนเป็น 01111111 ซึ่งจะได้ตัวเลขเป็น 7F และเมื่อขา IN1 (ขาที่ 2) มีการเปลี่ยนทางลจิกก็จะได้เป็น 10111111 ซึ่งจะได้ BF เปลี่ยนไปเรื่อยๆ หลังจากนั้นในวงจรจำลองจะส่งค่าไปให้ยังไมโครคอนโทรลเลอร์ AVR ที่ทำการทดลอง เมื่อไมโครคอนโทรลเลอร์ AVR รับค่าได้ก็จะส่งข้อความออก โดยข้อความที่ส่งนั้นจะเป็นการแปลงรหัส PDU เพื่อให้ตรงกับการติดต่อ SMS ของมือถือ

โดยการจำลองนั้นจะใช้โหมด Animating แสดงการทำงานโดยภาพหน้าต่างของ Virtual Terminal ในโปรแกรม Proteus และจะทำการใส่ Input เข้าที่ขา IN0 และดูค่าที่แสดงออกมา โดยค่าที่แสดงเป็นการแปลงคือรหัส PDU เพื่อส่งออกไปยังโทรศัพท์มือถือ



รูปที่ 4.6 การทดสอบการเข้ารหัส PDU เมื่อกด IN0

จากการทดลองนี้ ค่าแสดงใน Virtual Terminal ที่ถอดรหัสแล้วจะมีความหมายคือ #01>Zone1 โดย Input ที่เข้าขา IN0 ถึง IN7 ก็จะมีค่าแตกต่างกันตามลำดับ Zone1 – Zone8 ตามการออกแบบวงจร เพื่อให้ผู้รับข้อความทราบถึงส่วนของที่เกิดเหตุที่แตกต่างกัน ซึ่งแต่ละค่า Input ที่เปลี่ยนไปก็จะทำให้รหัส PDU ใน Virtual Terminal เปลี่ยนไปดังจะเห็นได้จากรูปที่ 4.7



รูปที่ 4.7 แสดงความแตกต่างของรหัส PDU ที่ส่งตาม Zone1 – Zone8 จากการรับ IN0 – IN7

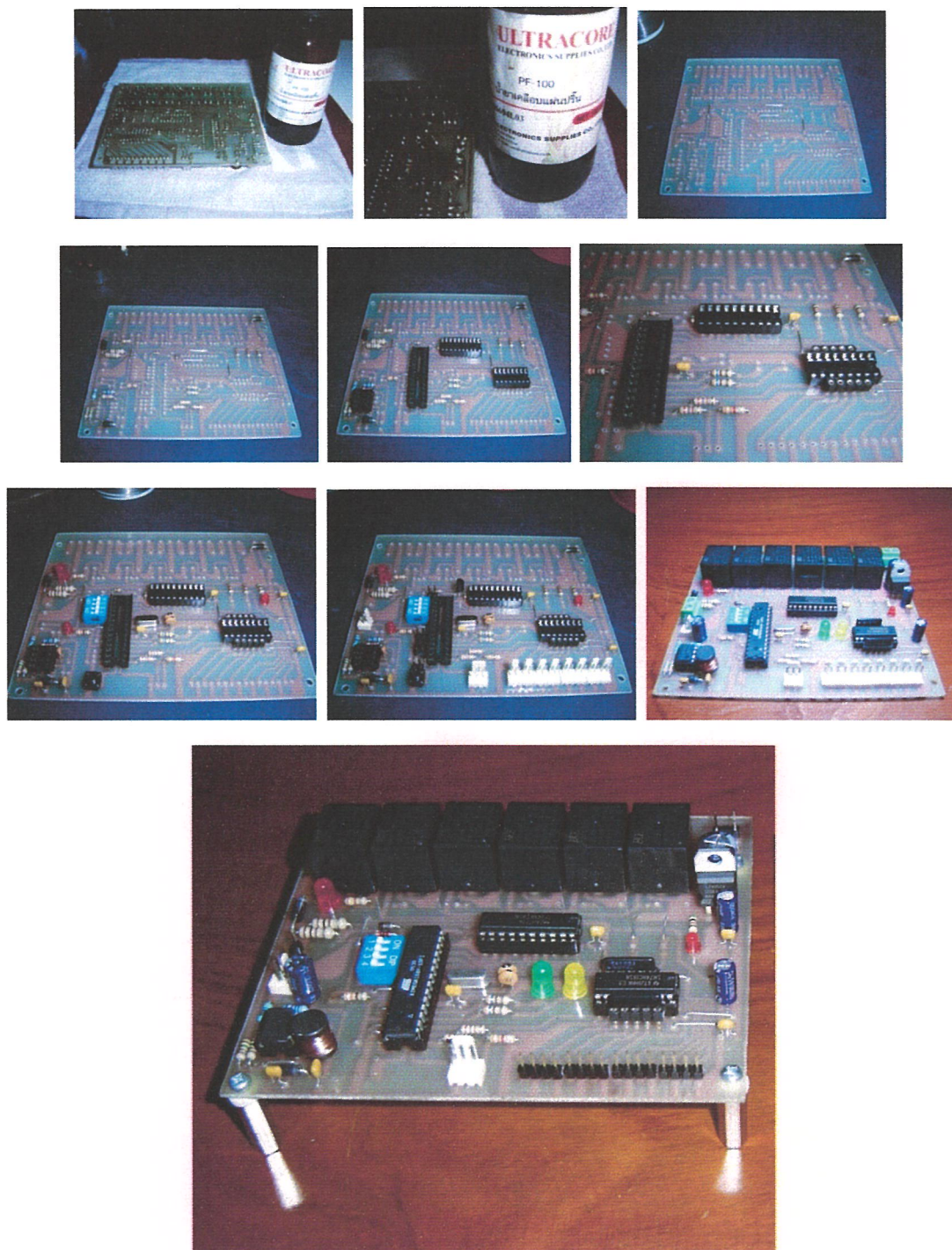
จากการทดลองในรูปที่ 4.7 แสดงให้เห็นถึงความแตกต่างกันในแต่ละส่วนที่รับสัญญาณอินพุต แล้วส่งออกมา โดยสังเกตจากข้อความที่แสดงใน Virtual Terminal ในส่วนของตัวอักษรสุดท้ายคือ B1 – B8 แสดงถึงข้อความที่ส่งที่แตกต่างกันซึ่งแปลความหมายจากโหมดการส่งแบบ PDU ออกมาเป็นข้อความที่แสดงในหน้าจอเมื่อถึงนั้น จะได้ข้อความคือคำว่า **Zone1 – Zone8** ซึ่งแสดงได้ถึง ช่องของสัญญาณอินพุต ที่รับสัญญาณเข้ามาในวงจรจำลองของโปรแกรม Proteus

จากนั้นเมื่อทำการจำลองโดยใช้โปรแกรม Proteus เป็นที่สำเร็จแล้ว ขั้นตอนต่อไปคือการทำในส่วนอุปกรณ์ และการทดลองจริง

## 4.2 การประกอบวงจรลงแผ่น PCB (Print Circuit Board )

### 4.2.1 การลงอุปกรณ์ของวงจรไมโครคอนโทรลเลอร์

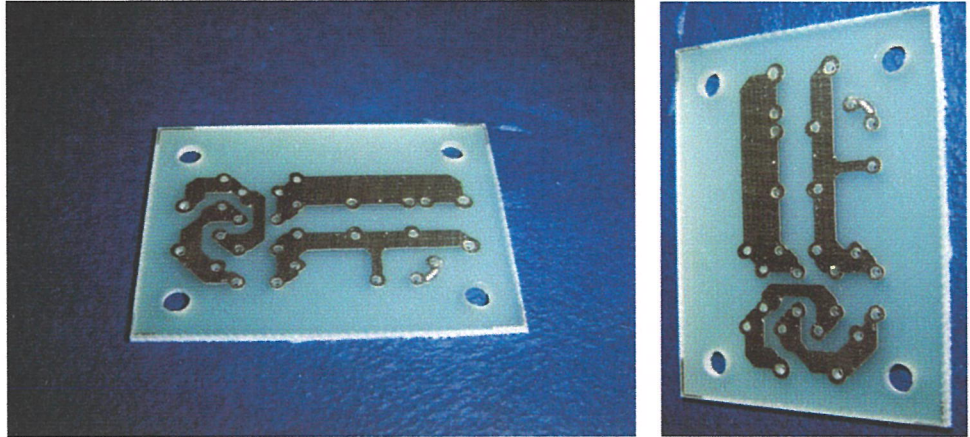
หลังจากที่ได้ทำการทดสอบการทำงานของวงจรโดยการจำลองผ่านโปรแกรม Proteus ในขั้นตอนนี้จะทำการประกอบวงจรทั้งหมดลงบนแผ่นปริ้นเอนกประสงค์ (Print Circuit Board ) รายละเอียดการประกอบจะแสดงตามขั้นตอนดังรูป



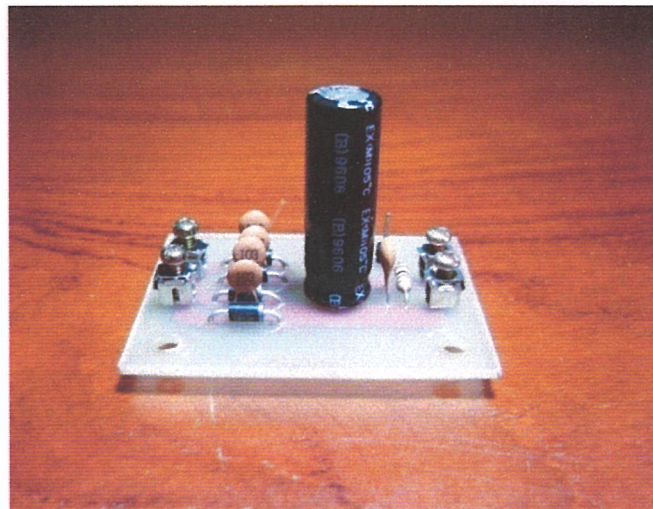
รูปที่ 4.8 แสดงขั้นตอนการลงอุปกรณ์

#### 4.2.2 การลงอุปกรณ์ของวงจรระดับแรงดัน

ในขั้นตอนของการลงวงจรระดับแรงดันจะสามารถอธิบายตามขั้นตอนได้ดังต่อไปนี้



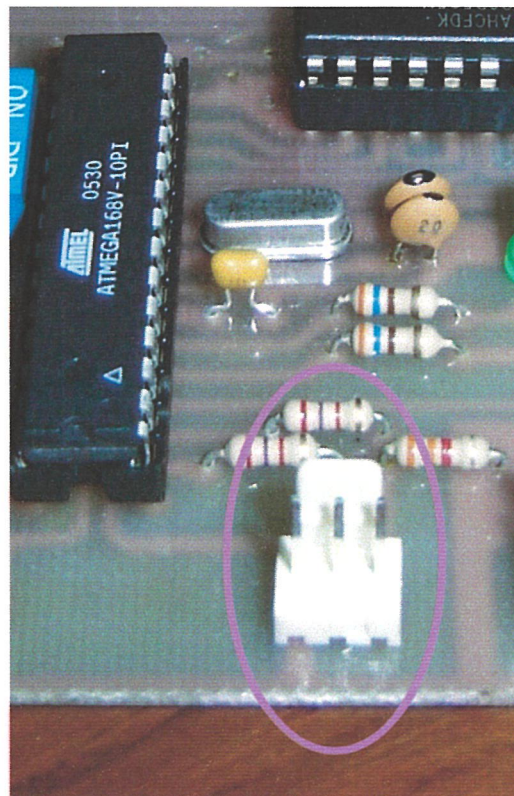
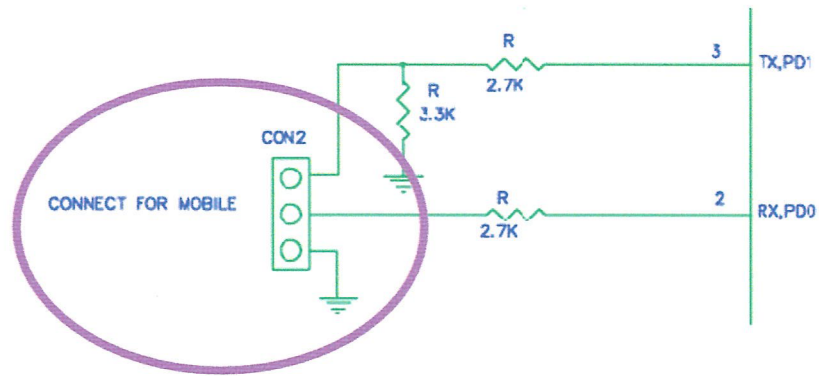
รูปที่ 4.9 ลายแผ่นทองแดงที่ผ่านการกัด และเจาะรู ของวงจรระดับแรงดัน



รูปที่ 4.10 แสดงการลงอุปกรณ์ของวงจรระดับแรงดัน

#### 4.2.3 การประกอบส่วนเชื่อมต่อระหว่างมือถือ กับวงจรควบคุม

จากวงจรควบคุม ถึงพอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือนั้น วงจรควบคุมจะมีขาคอนเน็คเตอร์อยู่ 3 ขา ที่แสดงการเชื่อมต่อจากไมโครคอนโทรลเลอร์เข้าสู่ตัวสายคาต้าลิงค์ (Data Link) ของมือถือ คือ ขาการเชื่อมต่อของสายสัญญาณส่งข้อมูล (Tx) ขาสายสัญญาณรับข้อมูล (Rx) และขากราวด์ (Ground)



รูปที่ 4.11 ขาคอนเน็คเตอร์ที่เชื่อมต่อไปยังโทรศัพท์มือถือ

เมื่อต่อสายสื่อสารจากวงจรควบคุมแล้วก็จะเข้าสู่ส่วนพอร์ตสื่อสารของโทรศัพท์มือถือ ซึ่งใช้สายคาต้าลิง (Data Link) เป็นตัวกลางเชื่อมการติดต่อเข้าสู่พอร์ตของมือถือ



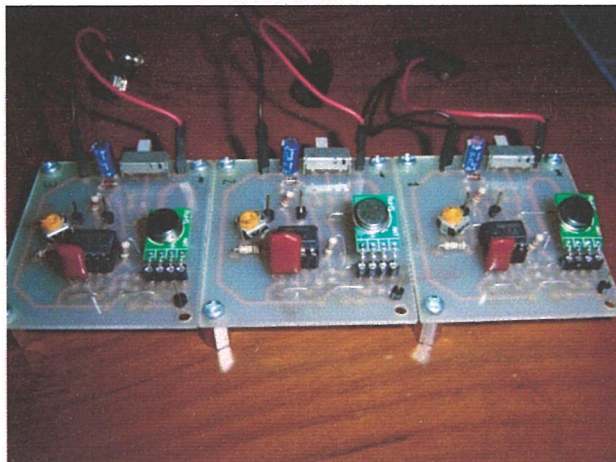
รูปที่ 4.12 สายคาต้าลิงค้ของโทรศัพท์มือถือ



รูปที่ 4.13 แสดงพอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือ

#### 4.2.4 การลงอุปกรณ์ของวงจรรับ – ส่งสัญญาณวิทยุ

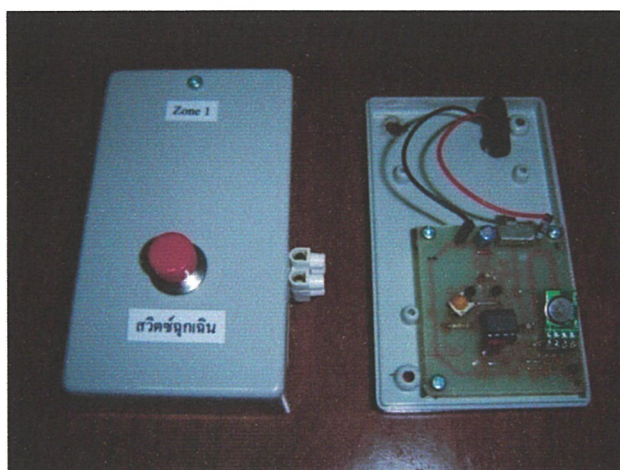
วงจรรับส่งสัญญาณ ประกอบด้วย 2 ส่วนหลักคือ วงจรภาคส่ง และวงจรภาครับ มีรายละเอียดของวงจรในการลงอุปกรณ์ดังต่อไปนี้



รูปที่ 4.14 วงจรส่วนภาคส่งสัญญาณทั้ง 3 ช่อง



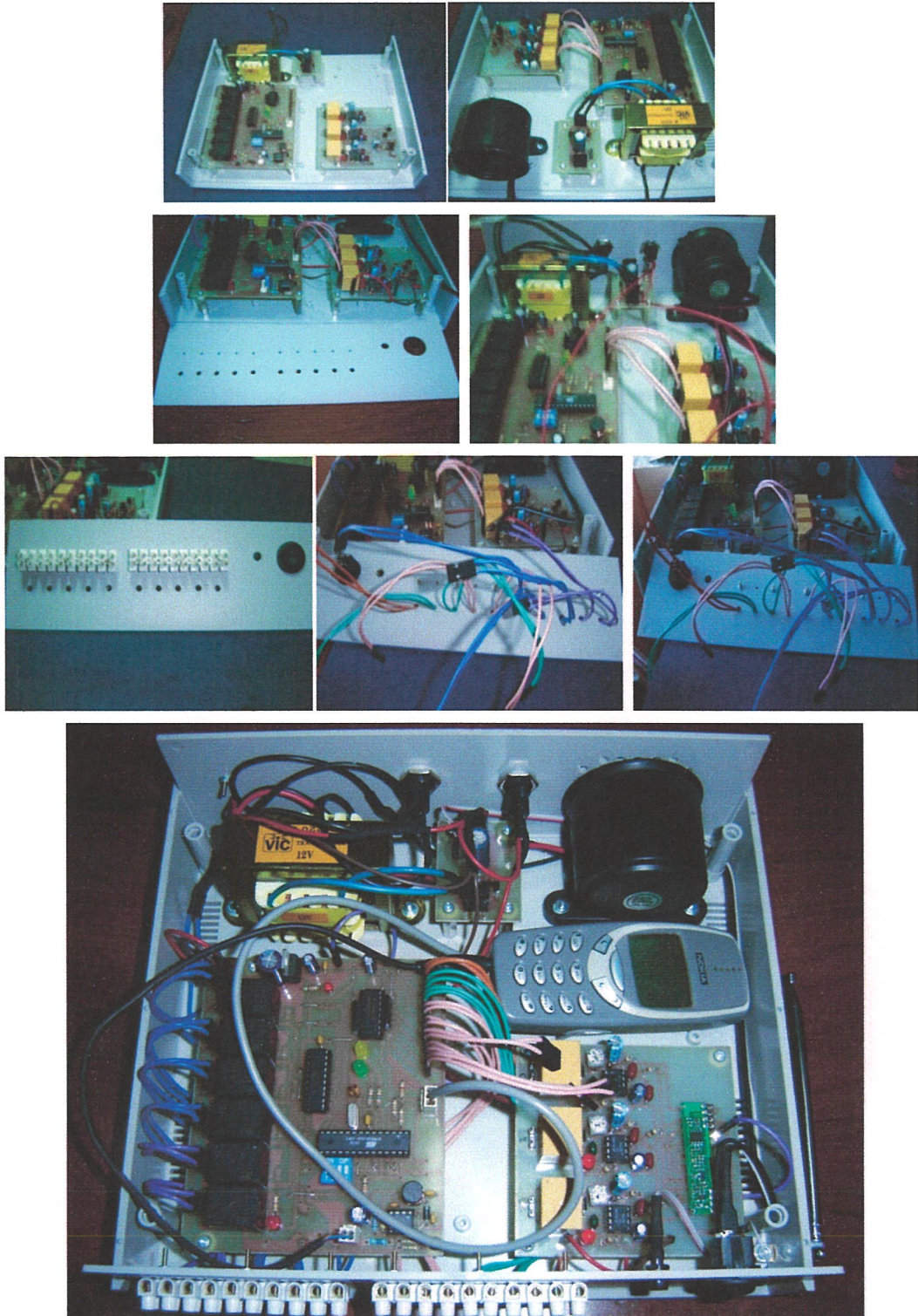
รูปที่ 4.15 วงจรส่วนภาครับสัญญาณวิทยุ



รูปที่ 4.16 วงจรภาคส่งประกอบลงกล่อง

#### 4.2.5 การประกอบกล่องควบคุม

ประกอบไปด้วยวงจรไมโครคอนโทรลเลอร์ วงจรภาครับสัญญาณ ไซเรน วงจรลดระดับแรงดัน และหม้อแปลง



รูปที่ 4.17 แสดงการนำอุปกรณ์ลงกล่อง ของวงจรควบคุม

#### 4.2.6 การนำวงจรภาคส่งลงกล่องเอนกประสงค์ และการติดตั้ง



รูปที่ 4.18 การนำอุปกรณ์ลงกล่องควบคุมของวงจรภาคส่ง



รูปที่ 4.19 การติดตั้งวงจรภาคส่งพร้อมทั้งสโมคคีเทคเตอร์



รูปที่ 4.20 แสดงการนำอุปกรณ์ลงกล่องของวงจรส่งสัญญาณควบคุม

### 4.3 การทดลองวงจรตรวจจับควัน

การทดลองเริ่มจากการประกอบ และติดตั้งในส่วนของสัญญาณภาคส่ง โดยการนำวงจรส่งสัญญาณวิทยุ ติดตั้งกับสโมคคีเทคเตอร์ในการพร้อมที่จะตรวจจับควัน



รูปที่ 4.21 การประกอบวงจรพร้อมการใช้งานของภาคส่ง

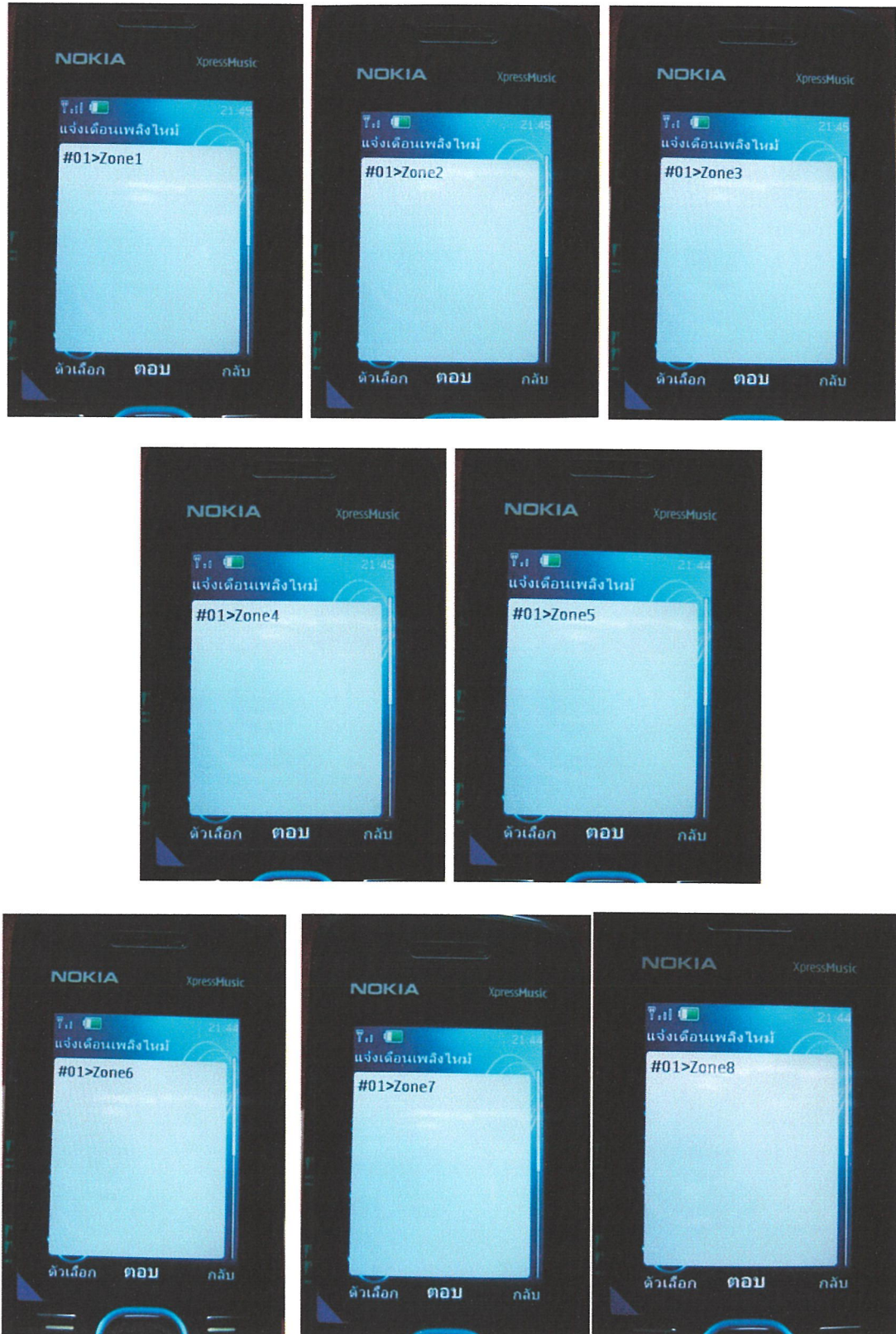
เมื่อเกิดเหตุเพลิงไหม้ ไซเรนจะทำการตรวจจับควันและทำการส่งอินพุตมายังวงจรภาคส่ง ในภาคส่งนั้นก็จะทำการส่งสัญญาณวิทยุในคลื่นความถี่ที่แตกต่างกันเพื่อแสดงถึงสถานที่ในการเกิดเหตุ

เมื่อสัญญาณวิทยุถูกส่งออกมายังภาครับ ก็จะทำการแปลงรหัสตามช่องสัญญาณเพื่อเข้าที่วงจรควบคุม แสดงค่าอินพุตในส่วนต่างๆที่เกิดเหตุ

จากนั้นไมโครคอนโทรลเลอร์จะทำการประมวลผลแล้วส่งไปเปิดการทำงานของไซเรนเพื่อแจ้งเตือนต่อไป อีกทั้งไมโครคอนโทรลเลอร์จะรับสัญญาณอินพุตและส่งข้อความไปยังโทรศัพท์มือถือ โดยการแสดงข้อความนั้นจะใช้คำว่า #01>Zone1 หมายถึง เกิดเหตุเพลิงไหม้ในบริเวณที่ 1



ที่ 4.22 ข้อความที่ส่งจากวงจรควบคุมถึงโทรศัพท์มือถือปลายทาง



รูปที่ 4.23 การทดสอบการแสดงผลการส่งการเตือนภัยผ่าน sms ทั้งหมด 8 จุด ที่แตกต่างกัน

#### 4.4 การทดสอบการส่งข้อความสั้นผ่านโปรแกรม HyperTerminal

การใช้งานโปรแกรม HyperTerminal สำหรับใช้ในงานที่ต้องการติดต่อแบบอนุกรมกับ ไมโครคอนโทรลเลอร์เชื่อมกับคอมพิวเตอร์ โดยการทดลองใช้คำสั่ง AT Command ในการติดต่อผ่านสาย Data Link และดูผลการตอบสนองจากโทรศัพท์มือถือ

ขั้นตอนแรกคือการตั้งค่าการรับส่งข้อมูลระหว่างเครื่องโทรศัพท์มือถือกับคอมพิวเตอร์ให้ตรงกับ อัตราการรับส่งข้อมูลสามารถตั้งค่าได้จาก Port Setting โดยค่าที่กำหนดคือ

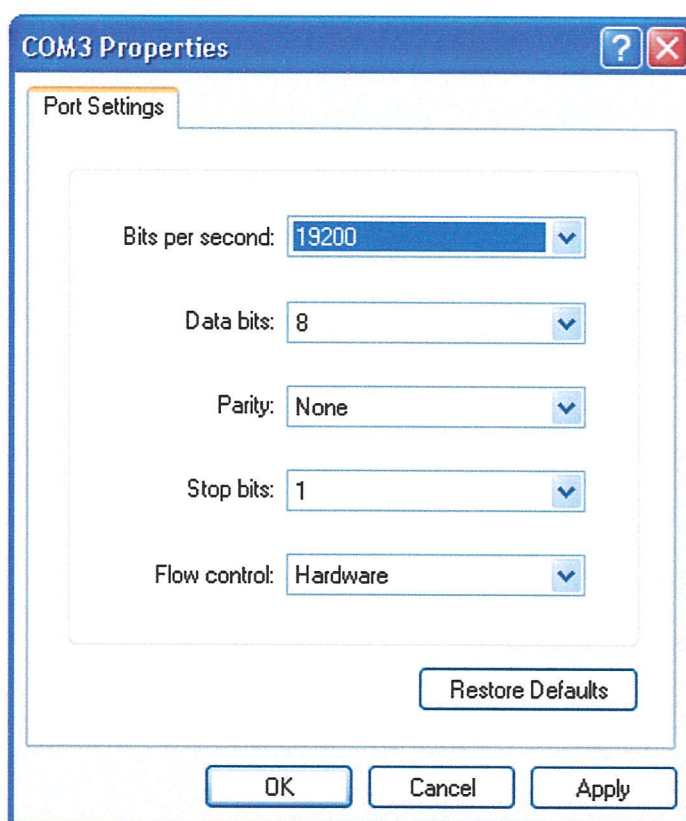
Bits per second มีค่าเท่ากับ 19200

Data bits มีค่าเท่ากับ 8

Parity bits มีค่าเท่ากับ None

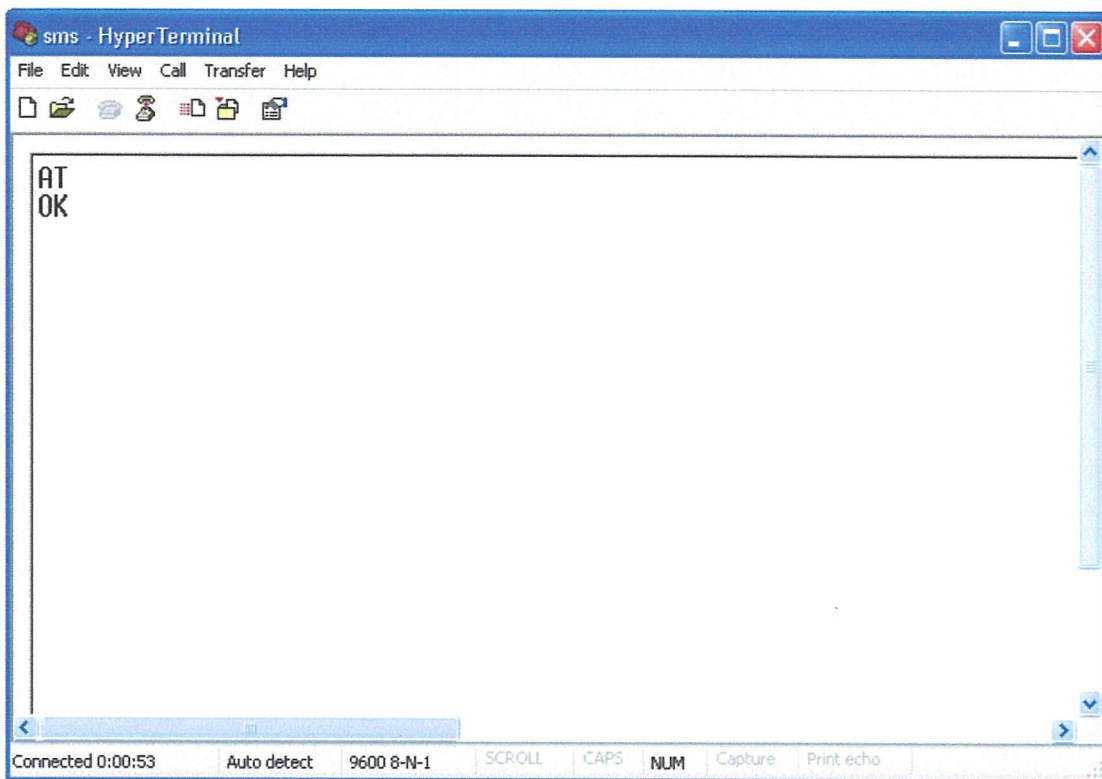
Stop bits มีค่าเท่ากับ 1

Flow control มีค่าเท่ากับ None



รูปที่ 4.24 Dialog Box Setting จาก โปรแกรม Hyper Terminal

เริ่มทำการทดสอบ โดยชุดคำสั่ง AT Command ไปยังโทรศัพท์มือถือโดยคำสั่งที่ใช้ในการทดสอบความพร้อมของอุปกรณ์คือ พิมพ์ AT แล้วกด Enter หากสามารถเชื่อมต่อกับโทรศัพท์มือถือได้ โทรศัพท์จะตอบกลับมาว่า OK




รูปที่ 4.25 แสดงการตอบรับจากโทรศัพท์มือถือเมื่อได้รับคำสั่ง AT

การทดสอบโหมดของการส่งข้อความสั้น โดยการพิมพ์คำสั่ง AT+CMGF ผ่านโปรแกรม Hyper Terminal

ค่าที่ได้รับการตอบกลับคือ +CMGF:(0) แสดงถึงชนิดของโหมดที่ใช้ในการรับ - ส่งข้อมูลที่เครื่องสามารถรองรับได้ซึ่งในที่นี้รองรับเฉพาะการรับ - ส่งในโหมด PDU

และจากการตอบกลับ +CMGF:0 หมายถึงโหมดการรับส่งที่ใช้อยู่ในขณะนี้คือ PDU Mode



The screenshot shows a HyperTerminal window titled 'sms - HyperTerminal'. The window contains the following text:

```
AT
OK

at+cmgf=?
+CMGF: (0)

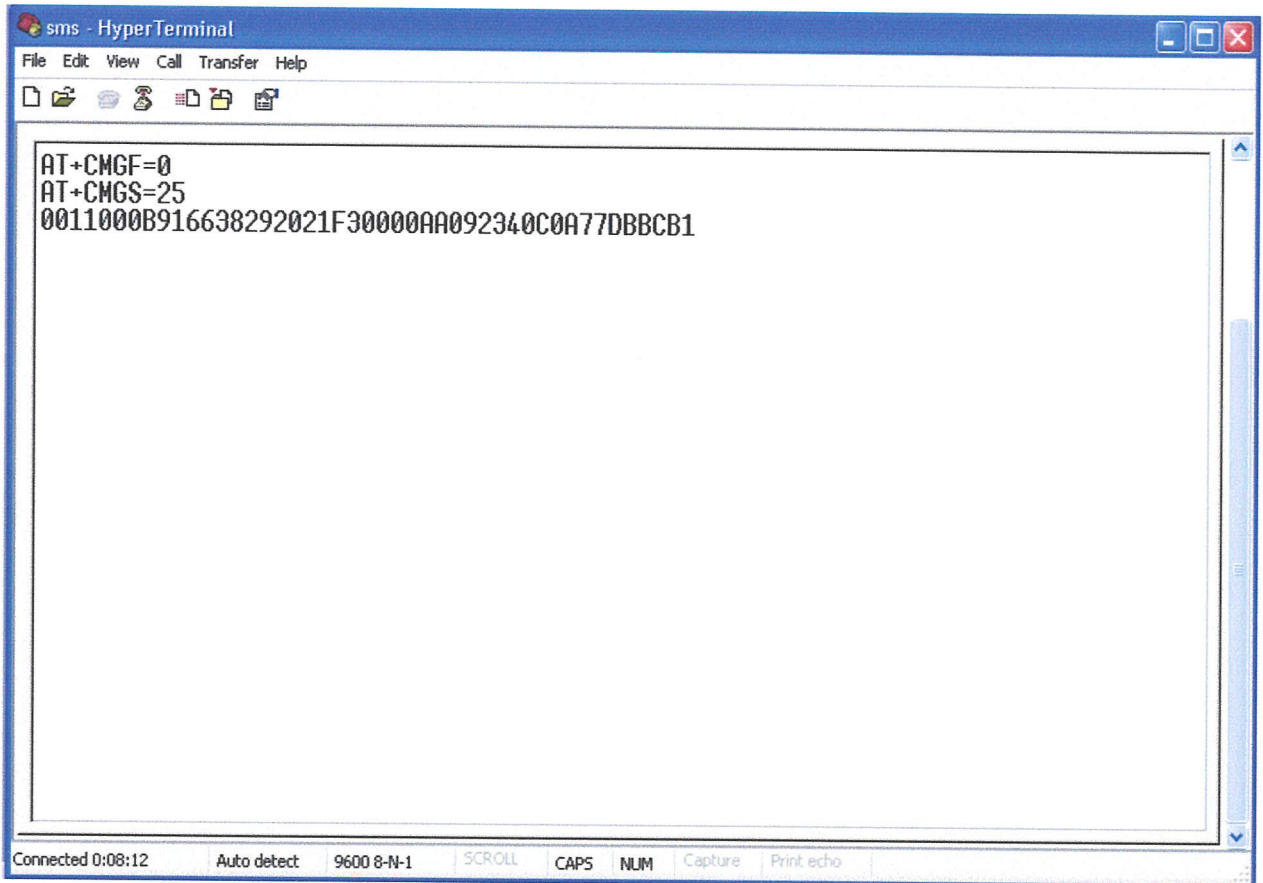
OK
at+cmgf?
+CMGF: 0

OK_
```

The status bar at the bottom of the window displays: 'Connected 0:07:33', 'Auto detect', '9600 8-N-1', 'SCROLL', 'CAPS', 'NUM', 'Capture', and 'Print echo'.

รูปที่ 4.26 ผลการทดลองเมื่อพิมพ์คำสั่ง AT+CMGF ผ่าน Hyper Terminal

การทดสอบข้อความที่ส่งไปยังโทรศัพท์มือถือนั้น จะทำการตรวจสอบผ่านทาง Hyper Terminal โดยในรูปแบบนี้จะเป็นการตรวจสอบการถอดรหัสโดยจะแสดงได้จากรูปที่ 4.27



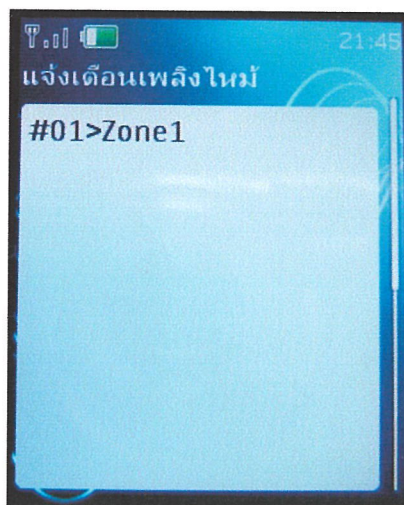
The screenshot shows a HyperTerminal window titled 'sms - HyperTerminal'. The menu bar includes 'File', 'Edit', 'View', 'Call', 'Transfer', and 'Help'. Below the menu bar are several icons for file operations. The main text area contains the following AT command output:

```
AT+CMGF=0
AT+CMGS=25
0011000B916638292021F30000AA092340C0A77DBBCB1
```

At the bottom of the window, a status bar shows 'Connected 0:08:12', 'Auto detect', '9600 8-N-1', 'SCROLL', 'CAPS', 'NUM', 'Capture', and 'Print echo'.

รูปที่ 4.27 แสดงการทดสอบการส่งข้อความใน PDU Mode ผ่าน Hyper Terminal

โดยข้อความที่วางจรวดควบคุมส่งผ่านไปยังโทรศัพท์มือถือ นั่นคือ #01>Zone1 ซึ่งแสดงถึงบริเวณที่เกิดเหตุเพลิงไหม้ โดยข้อความจะปรากฏบนโทรศัพท์มือถือปลายทาง เป็นข้อความแจ้งเตือนให้ทราบ



รูปที่ 4.28 แสดงข้อความที่โทรศัพท์มือถือปลายทางได้รับข้อความว่า #01>Zone1

ส่วนประกอบของข้อมูลที่ส่งคือ 0011000B91663829202F20000AA092340C0A77DBBCB1

| กลุ่มตัวเลข 8 บิต (Octet) | รายละเอียด   |
|---------------------------|--|
| 00                        | ความยาวของ SMSC Information โดยที่ 00 หมายถึง ให้ใช้ SMSC Information ที่เก็บอยู่ในเครื่อง (ปกติแล้วเครื่องที่สามารถส่ง SMS ได้มีข้อมูล SMSC ภายในเครื่องอยู่แล้ว) |
| 11                        | First octet of the SMS-SUBMIT message  |
| 00                        | TP-message-Reference โดยที่ 00 คือให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง   |
| 0B                        | Address-Length คือความยาวของเลขหมายผู้รับ (11 ตัว)   |
| 91                        | ใช้แสดงว่าเบอร์โทรศัพท์เคลื่อนที่เป็นแบบภายในประเทศ  |
| 66 38 29 20 21 F3         | เบอร์โทรศัพท์เคลื่อนที่ปลายทางที่ทำการเข้ารหัสแบบสลับแล้ว โดยหมายเลขที่แท้จริงคือ +66 39202123   |
| 00                        | TP-PID (Protocol identifier) ในกรณีนี้เป็น 00  |
| 00                        | TP-DCS (Data coding scheme) เป็น 00 คือเข้ารหัสข้อความแบบ 7 บิต Default Alphabet   |
| AA                        | TP-Validity-Period โดยที่ AA หมายถึง ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังส่งไม่ถึงปลายทาง ข้อความจะถูกยกเลิกโดยอัตโนมัติ                         |
| 09                        | TP-User-Data-Length คือความยาวของข้อความก่อนแปลงรหัส นั่นคือ “#01>Zone1 ” = 9 ตัวอักษร   |
| 2340C0A77DBBCB1           | TP-User- Data เป็นข้อมูลในส่วน of ข้อความตัวอักษรแบบ 7 บิต นั่นคือคำว่า “#01>Zone1 ” ที่ผ่านการแปลง (เข้ารหัส) เป็นข้อมูลแบบ 8 บิต แล้ว                            |

ตารางที่ 4.1 แสดงส่วนประกอบของข้อมูลที่ส่งข้อความสั้นภาษาอังกฤษของคำว่า “#01>Zone1 ”

การแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ( Octet ) โดยข้อความจาดตารางจะเป็นส่วนที่เราสามารถใส่รหัสของข้อความที่ต้องการส่ง แต่เนื่องจากเราไม่สามารถนำรหัสของตัวอักษรแบบ 7 บิต ไปได้โดยตรงจำเป็นต้องผ่านการแปลงให้เป็นรหัสข้อมูลแบบ 8 บิตก่อนโดยในการส่งข้อความนี้เป็นการส่งข้อความจากเครื่องส่งที่เกิดเหตุไปยังเครื่องรับด้วยคำว่า **#01>Zone1** ซึ่งสามารถบ่งบอกถึงเครื่องรับแล้วสถานที่เกิดเหตุในแต่ละส่วนที่แบ่งไว้ได้ 9 ตัวอักษรซึ่งแต่ละตัวอักษรเป็นชนิด 7 บิตให้เป็นข้อมูล 8 บิต สำหรับใช้ในการส่ง SMS การแปลงเริ่มจากการนำรหัส 7 บิตของตัวอักษรแรก ( # ) มาเติมข้างหน้าด้วย 0 จากบิตสุดท้ายของอักษรตัวที่ 2 ( 0 ) จะได้ผลลัพธ์ 8 บิต ( 1 ไบต์ ) เป็น “23” ขึ้นตอนต่อมาให้เอา 6 บิตที่เหลือของอักษรตัวที่ 2 มาเติมข้างหน้าด้วย 2 บิตท้ายของรหัส 7 บิตของอักษรตัวที่ 3 ( 1 ) จะได้ผลลัพธ์ 8 บิตเป็น “40” และทำเช่นนี้ไปเรื่อยๆ โดยจำนวนที่นำมากระทำจะเพิ่มขึ้นเป็น 3 บิต จนกระทั่งถึง 7 บิต แล้วเริ่มกระบวนการใหม่จนหมดชุดตัวอักษร หลังจากการแปลงข้อความ “**#01>Zone1**” จะได้ข้อมูลเป็นเลขฐาน 16 จำนวน 8 ไบต์เป็น 23 40 C0 A7 7D BB CB 1 โดยวิธีการแปลง จะแสดงที่ตาราง โดยที่ตัวอักษรชนิด 7 บิต ถูกกำหนดโดยมาตรฐาน GSM 03.38 ดังตารางที่ 4.2

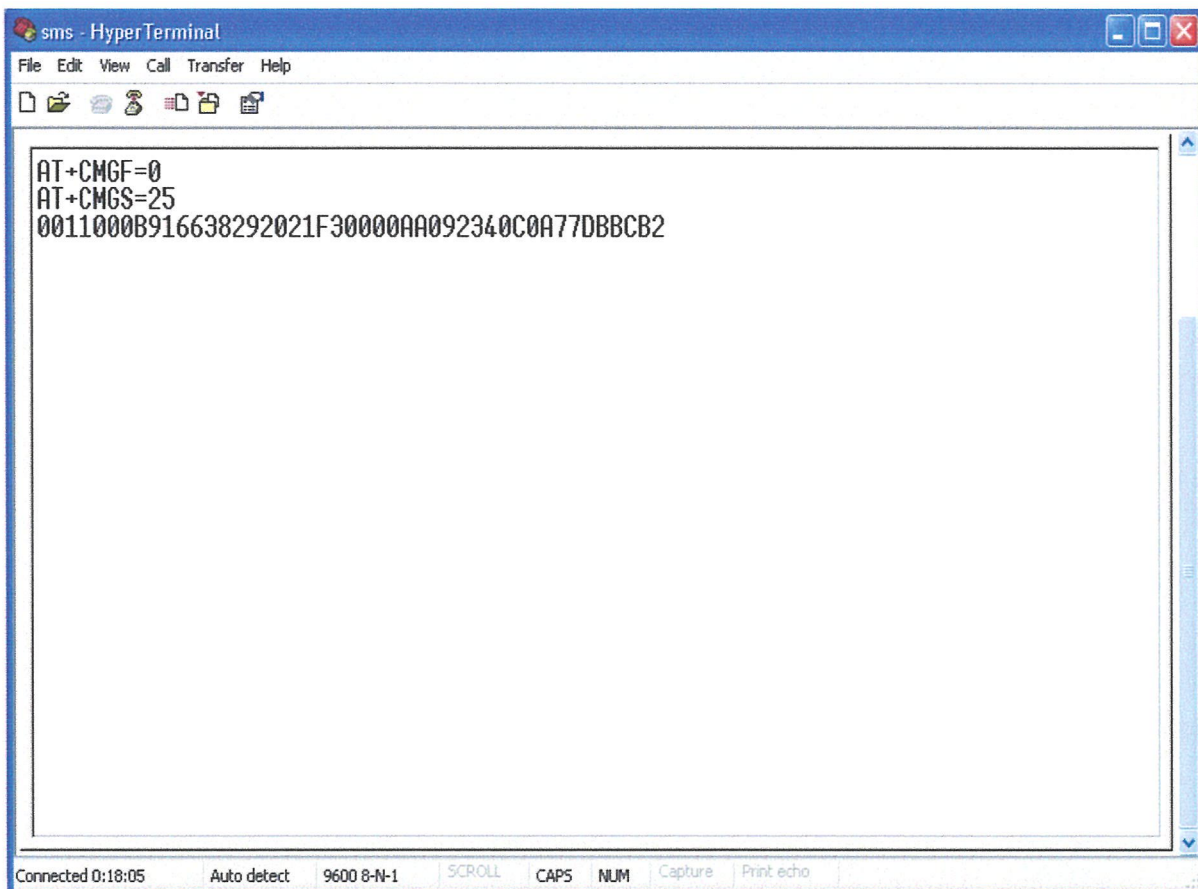
ตัวอย่างการแปลงเลขฐานสองเป็นเลขพิตียูโค้ดด้วยข้อความ “**#01>Zone1**”

| เลขฐานสิบหก<br>(แอสกีโค้ด) | เลขฐานสอง<br>(แอสกีโค้ด) | เลขฐานสอง(พิตียู) | พิตียูโค้ด | ข้อความ |
|----------------------------|--------------------------|-------------------|------------|---------|
| 23                         | 00100011                 | 0100011           | 35         | #       |
| 40                         | 01000000                 | 0000000           | 0          | 0       |
| C0                         | 11000000                 | 0000001           | 1          | 1       |
| A7                         | 10100111                 | 0111110           | 62         | >       |
| 7D                         | 01111101                 | 1011010           | 90         | Z       |
| BB                         | 10111011                 | 1101111           | 111        | o       |
| CB                         | 11001011                 | 1101110           | 110        | n       |
| 1                          | 00000001                 | 1100101           | 101        | e       |
|                            | -                        | 0000001           | 1          | 1       |

ตารางที่ 4.2 แสดงวิธีการแปลงตัวอักษรชนิด 7 บิตเป็นข้อมูล 8 บิต ด้วยข้อความ “**#01>Zone1**”

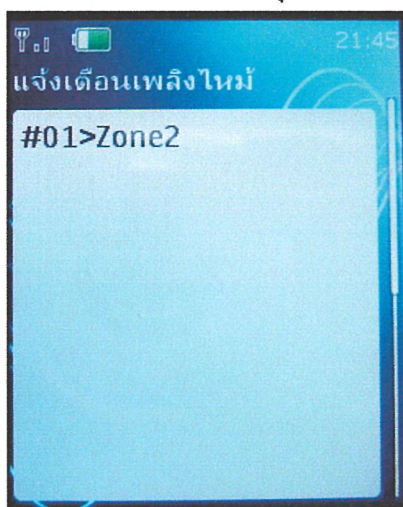
การทดสอบ วัตถุประสงค์ข้อมูลที่ส่งจากไมโครคอนโทรลเลอร์ไปยังโทรศัพท์มือถือ โดยใช้โปรแกรม

Hyper Terminal



รูปที่ 4.29 แสดงการทดสอบการส่งข้อความใน PDU Mode ผ่าน Hyper Terminal

ข้อความที่ส่งออกไปยังโทรศัพท์มือถือปลายทางคือ #01>Zone02 เป็นการแสดงส่วนองที่เกิดเหตุในโซนที่ 2 ที่เซนเซอร์ได้จับตำแหน่งไว้ โดยข้อความที่จะส่งไปยังเลขหมายปลายทาง ในการระบุส่วนท้ายของข้อความจะเปลี่ยนแปลงเพื่อแจ้งถึงสถานที่เกิดเหตุที่แตกต่างกัน



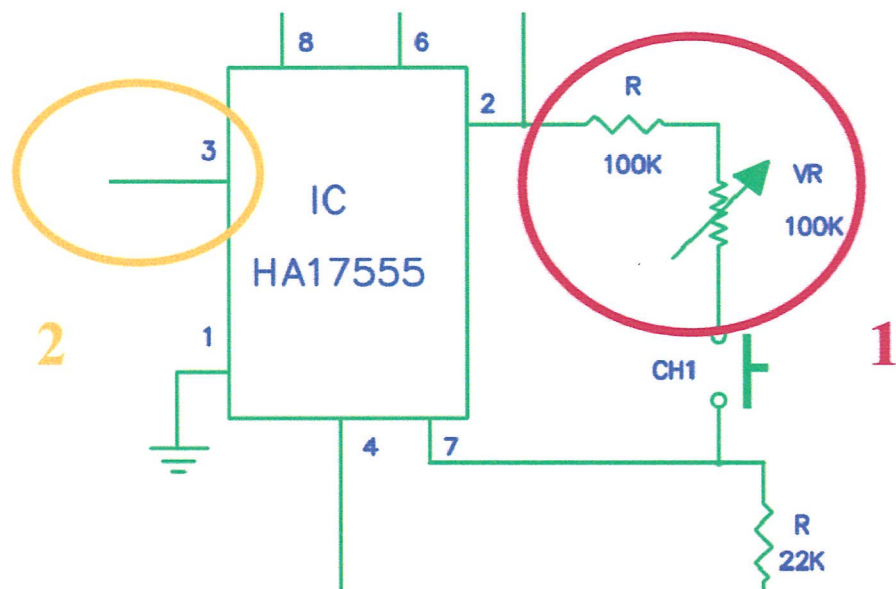
รูปที่ 4.30 แสดงข้อความที่โทรศัพท์มือถือปลายทางได้รับข้อความว่า #01>Zone2

#### 4.5 ผลการทดลองการวัดสัญญาณ

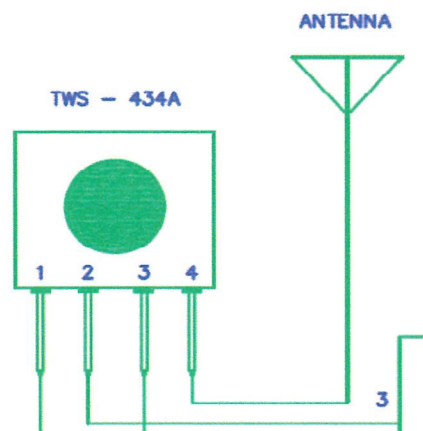
##### 4.5.1 การวัดสัญญาณส่วนวงจรรับ – ส่งสัญญาณ

การทดลองเริ่มจากการการวัดสัญญาณที่ส่งออกจากวงจรรับ – ส่ง สัญญาณ โดยนำภาคส่ง และภาครับ มาเปรียบเทียบกัน

โดยภาคส่งจะทำการส่งความถี่ของแต่ละช่องสัญญาณที่แตกต่างกัน ขึ้นอยู่กับการปรับเปลี่ยนค่าตัวต้านทานปรับค่าได้ ( VR ) และตัวต้านทานแบบธรรมดา ( R ) ที่ต่ออนุกรมกัน ที่ทำให้ค่าความต้านทานนั้นมีความแตกต่างกัน เพื่อควบคุมอุปกรณ์ จากนั้นสัญญาณที่มีความถี่เฉพาะของแต่ละช่องสัญญาณที่สร้างด้วย IC HA17555 จะส่งออกที่ขา 3 เข้าที่โมดูล TWS – 434A เพื่อทำการมอดูเลตแล้วส่งออกเป็นสัญญาณวิทยุ

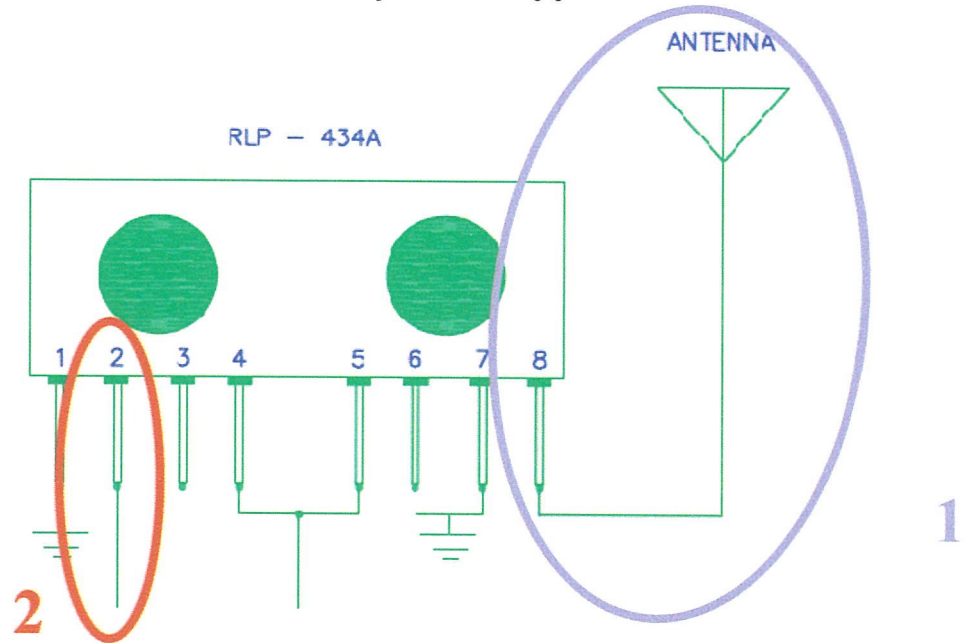


รูปที่ 4.31 ขาของ IC HA17555 แต่ละส่วนที่ทำการต่อจากวงจรของภาคส่งสัญญาณ ส่วนที่ 1 การปรับเปลี่ยนค่า VR และ R ซึ่งทำให้เกิดความแตกต่างของสัญญาณความถี่ที่ส่ง ส่วนที่ 2 สัญญาณความถี่ ที่ทำส่งออกจาก IC HA17555 โดยค่าความถี่ที่แตกต่างกันทางขา 3



รูปที่ 4.32 TWS – 434A ทำหน้าที่ในมอดูเลตสัญญาณและทำการส่งไปยังภาครับ

ในส่วนภาครับเมื่อมีสัญญาณจากภาคส่งเข้ามา RLP - 434A จะทำหน้าที่ในการรับสัญญาณที่เข้ามาทางขาที่ 8 ของอุปกรณ์ จากนั้นจะทำการตีโมดูเลต และส่งสัญญาณข่าวสารออกทางขาที่ 2

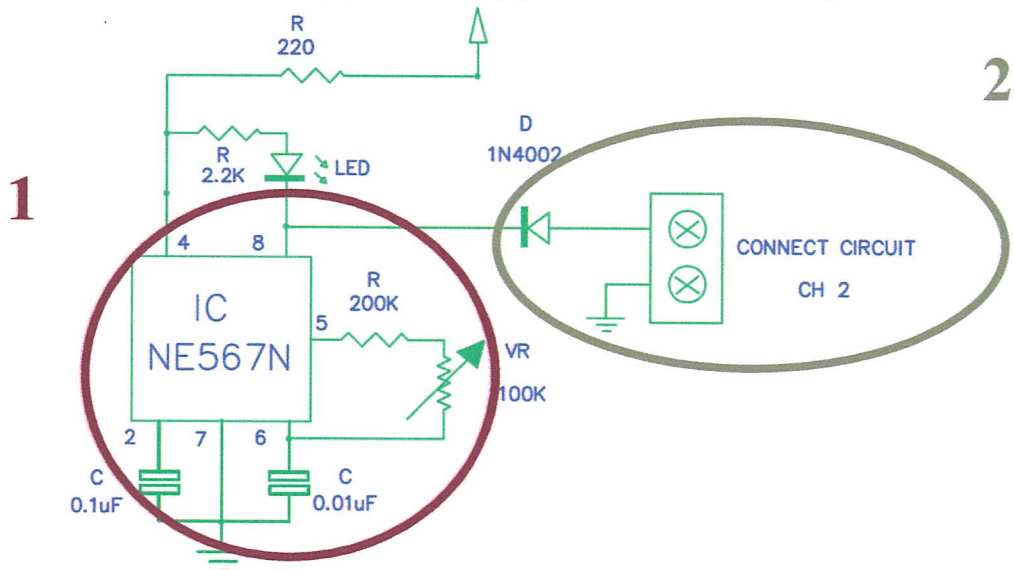


รูปที่ 4.33 โมดูล RLP - 434A ทำหน้าที่รับสัญญาณ

ส่วนที่ 1 ส่วนรับสัญญาณที่ได้เข้ามา ทางขาที่ 8

ส่วนที่ 2 สัญญาณที่ถูกแปลงจะส่งข้อมูลออกทางขาที่ 2

เมื่อสัญญาณที่เข้ามาจะถูกนำมาเปรียบเทียบกับความถี่ของแต่ละช่องสัญญาณความถี่ โดย IC NE567N ซึ่งเป็นตัวถอดรหัสสัญญาณ และส่งสัญญาณออกไปยังวงจรส่วนควบคุม



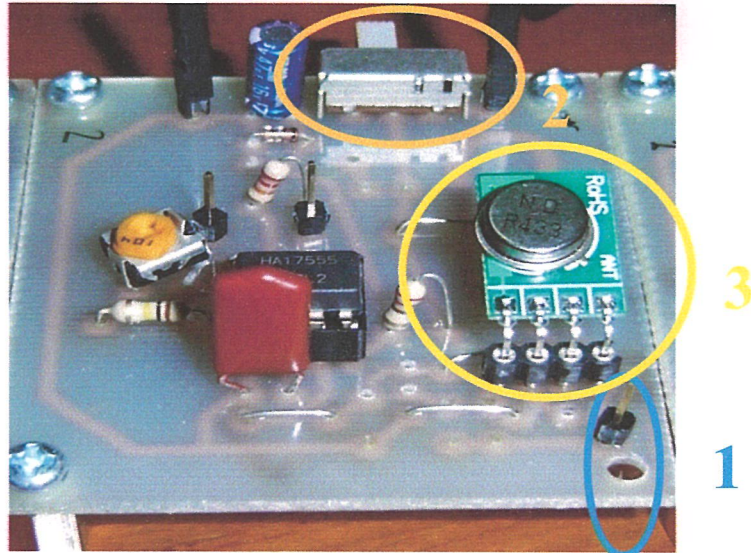
รูปที่ 4.34 แสดงช่องที่ทำการส่งสัญญาณออกของภาครับ

ส่วนที่ 1 IC NE567N ที่ทำการถอดรหัสสัญญาณ

ส่วนที่ 2 ส่งข้อมูลที่เข้าวงจรควบคุม

ระยะทางของภาคส่ง และภาครับ โดยการทดลองการวัด พบว่าระยะทางที่ไกลที่สุดที่สามารถรับ และส่งได้ในพื้นที่ไม่มีสิ่งกีดขวางนั้นอยู่ที่ 50 เมตร และในบริเวณที่มีสิ่งกีดขวาง เช่น กำแพงตึก อยู่ที่ 30 เมตร

การทดลองจะทำการวัดสัญญาณของภาคส่งขณะที่ มีสัญญาณมอดูเลตและไม่มีสัญญาณเพื่อ เปรียบเทียบรูปสัญญาณที่วัดได้ และสัญญาณของภาครับ ที่รับได้ จากวงจรภาคส่งทั้ง 3 ตัว

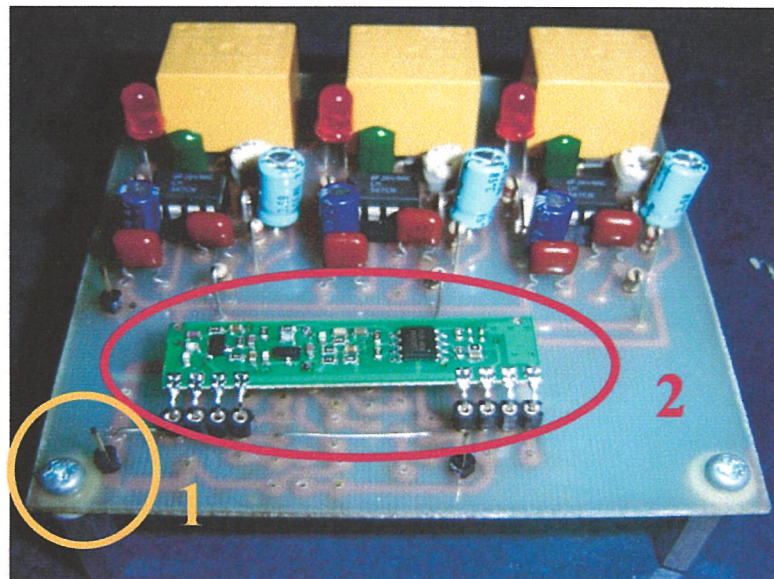


รูปที่ 4.35 แสดงจุดที่วัดสัญญาณของวงจรภาคส่ง

ส่วนที่ 1 จุดเชื่อมต่อสายอากาศที่ทำการส่งสัญญาณความถี่ออกไป

ส่วนที่ 2 สวิตช์ ปิด - เปิด ของวงจรภาคส่ง

ส่วนที่ 3 โมดูลภาคส่ง TWS - 434A ที่มีอคูเลตและส่งสัญญาณออกทางสายอากาศ

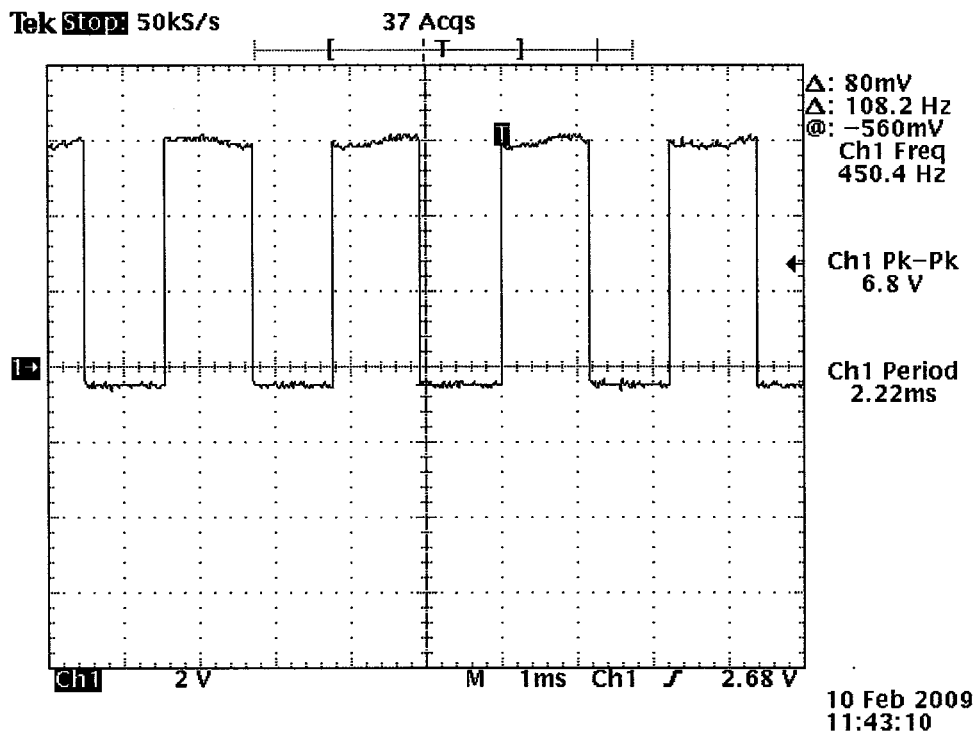


รูปที่ 4.36 แสดงจุดที่วัดสัญญาณของวงจรภาครับ

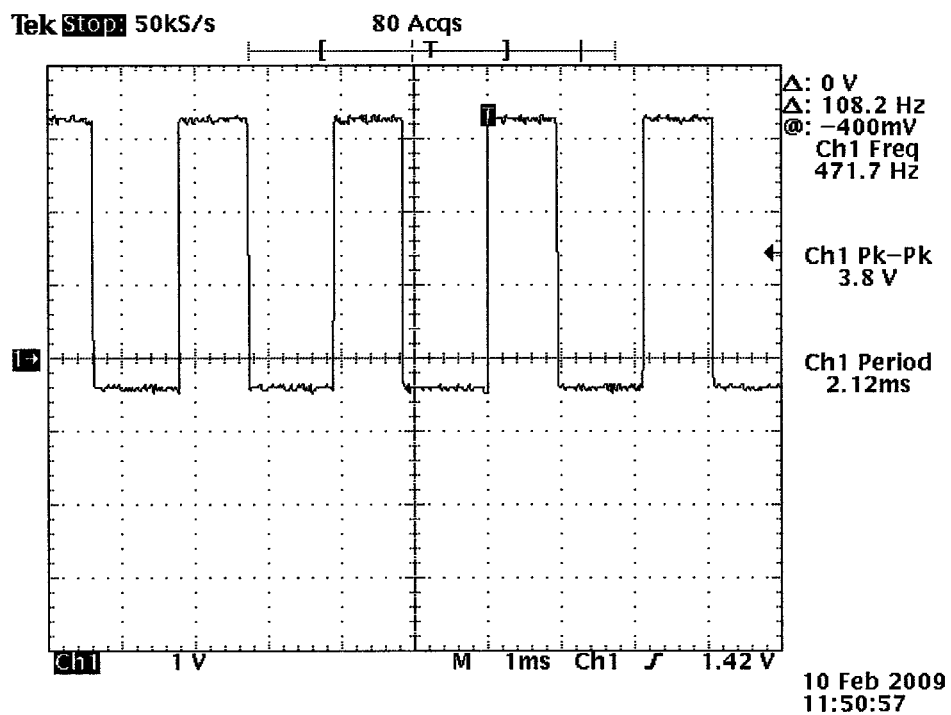
ส่วนที่ 1 จุดเชื่อมต่อสายอากาศภาครับสัญญาณ

ส่วนที่ 2 โมดูล RLP - 434A เป็นตัวดีมอดูเลตสัญญาณและส่งเข้าที่ IC NE567N เพื่อถอดรหัส

การทดลองเปรียบเทียบสัญญาณของวงจรภาคส่งและภาครับตัวที่ 1

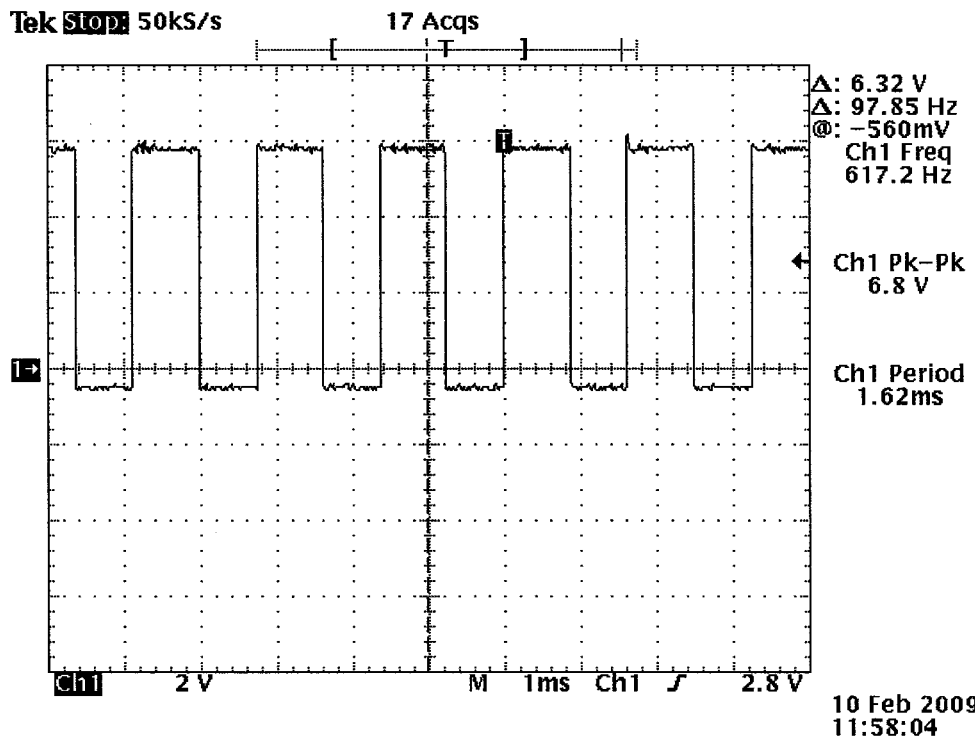


รูปที่ 4.37 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 1

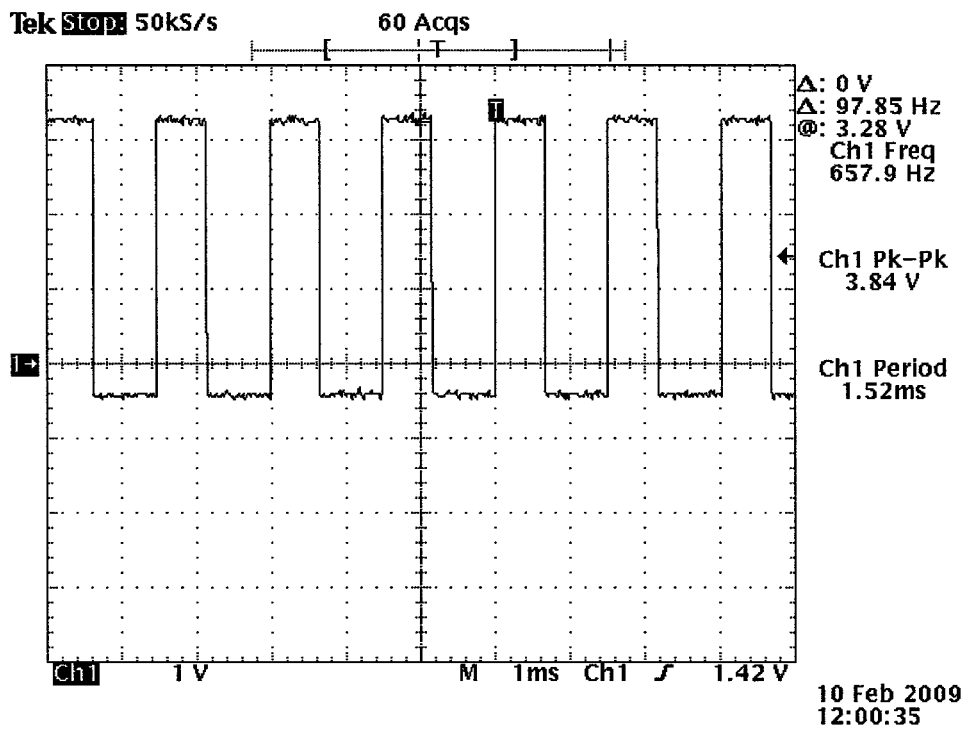


รูปที่ 4.38 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 1

การทดลองเปรียบเทียบสัญญาณของวงจรภาคส่งและภาครับตัวที่ 2

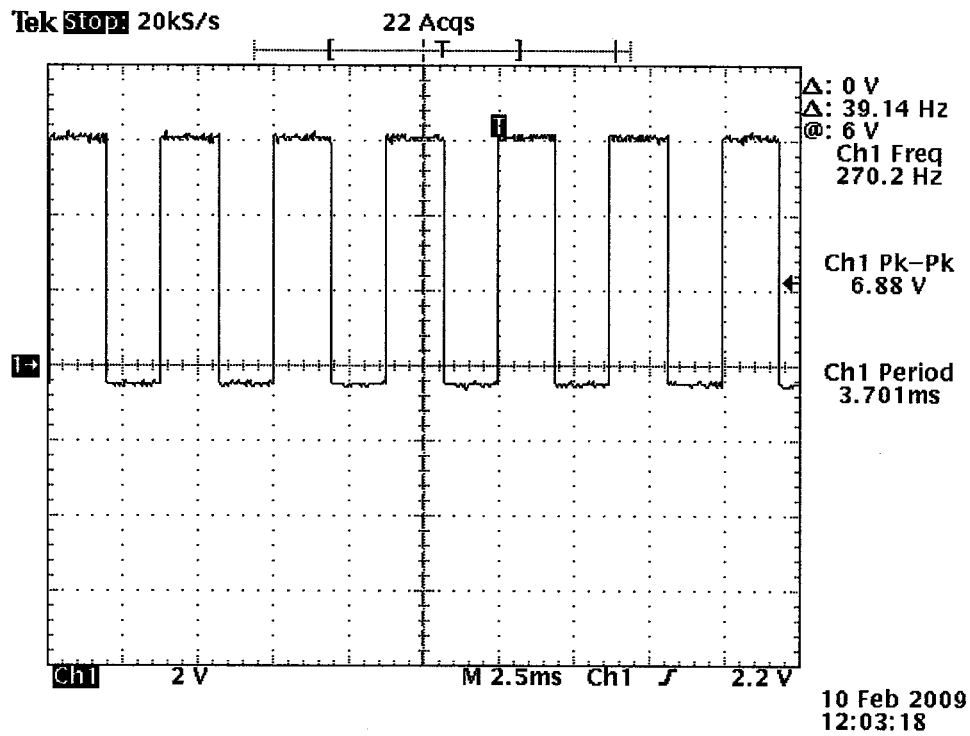


รูปที่ 4.39 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 2

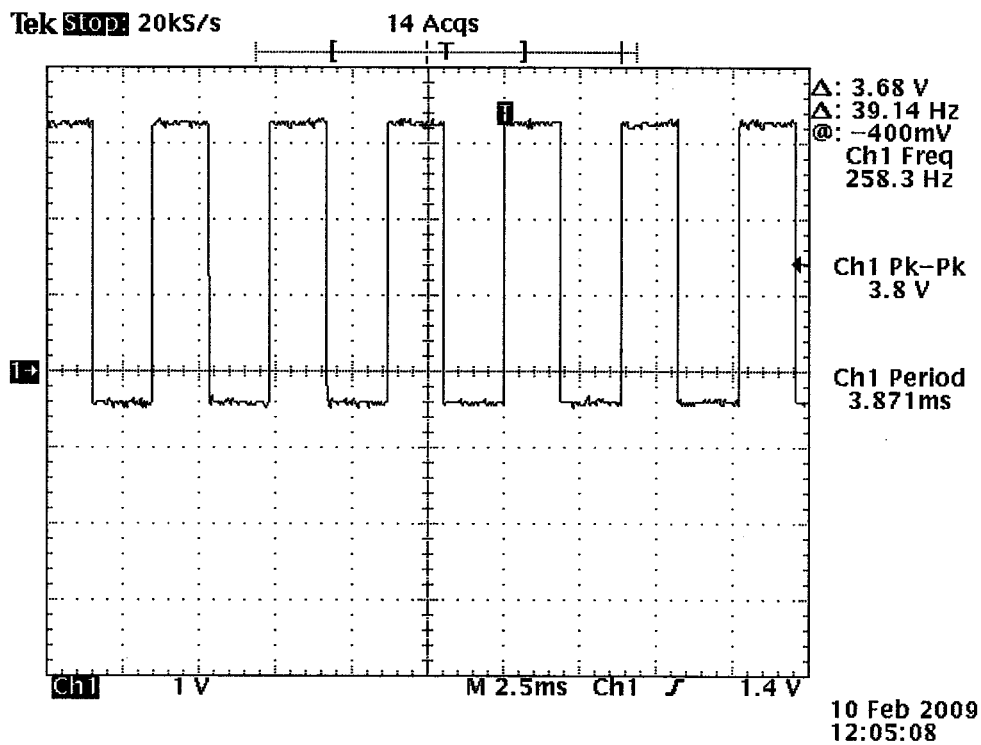


รูปที่ 4.40 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 2

การทดลองเปรียบเทียบสัญญาณของวงจรภาคส่งและภาครับตัวที่ 3

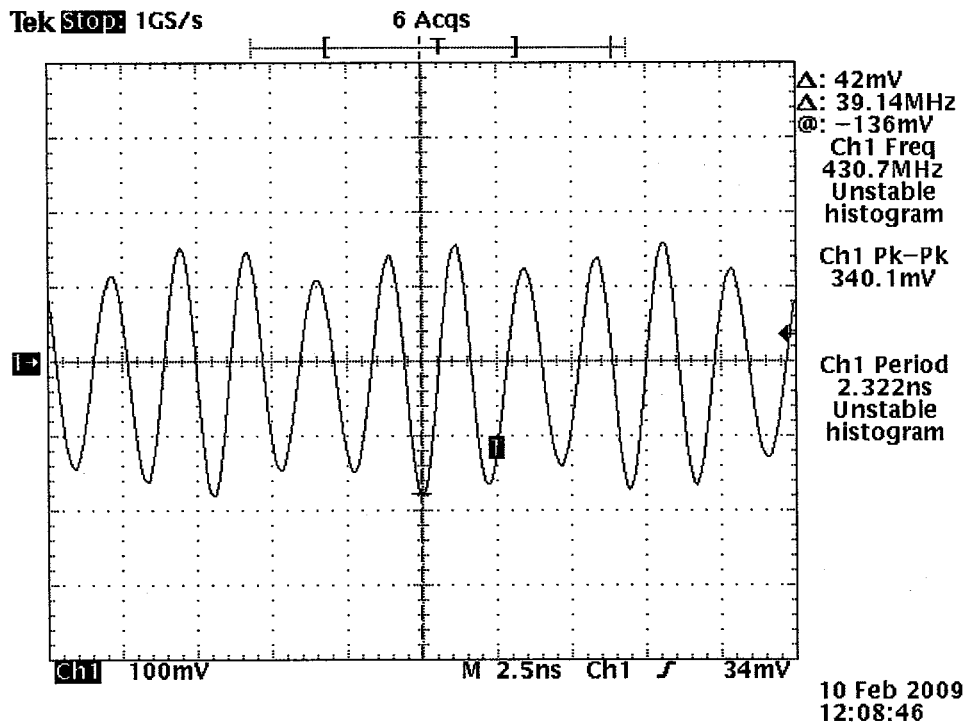


รูปที่ 4.41 สัญญาณที่ออกจากขา 3 ของ IC HA17555 วงจรภาคส่งตัวที่ 3

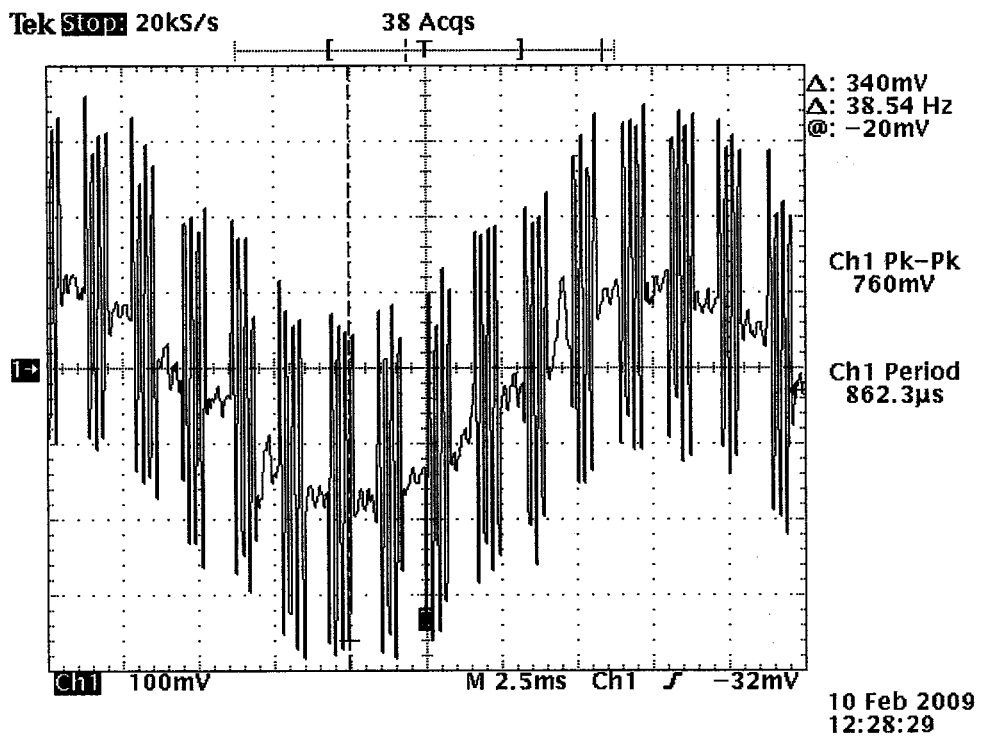


รูปที่ 4.42 สัญญาณอินพุตของ IC NE567N ของภาครับตัวที่ 3

การเปรียบเทียบสัญญาณที่ส่งออกจากสายอากาศ และสัญญาณที่ทำการมอดูเลตแล้ว



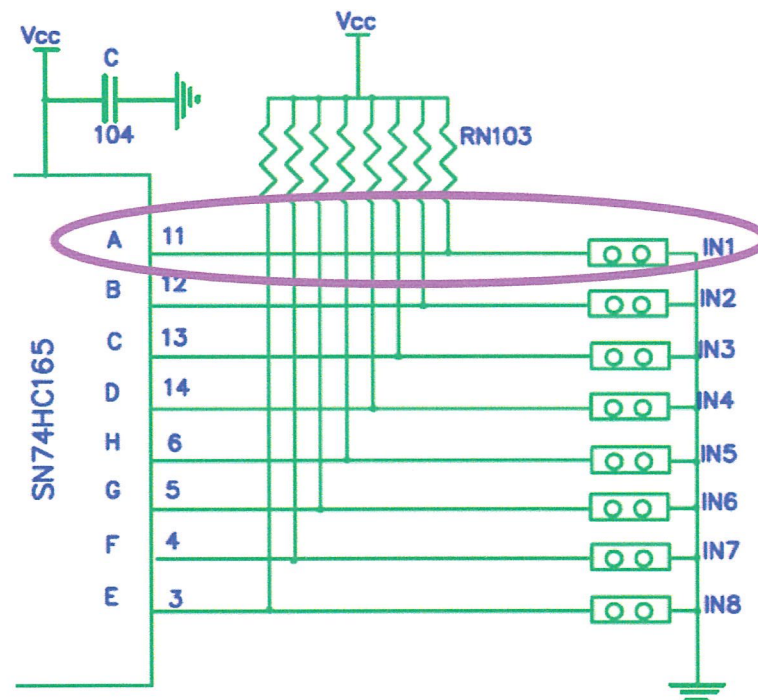
รูปที่ 4.43 สัญญาณที่ยังไม่ได้ทำการมอดูเลต



รูปที่ 4.44 สัญญาณที่ทำการมอดูเลตกับคลื่นพาหะ

#### 4.5.2 การวัดสัญญาณของวงจรถวนคุมไมโครคอนโทรลเลอร์

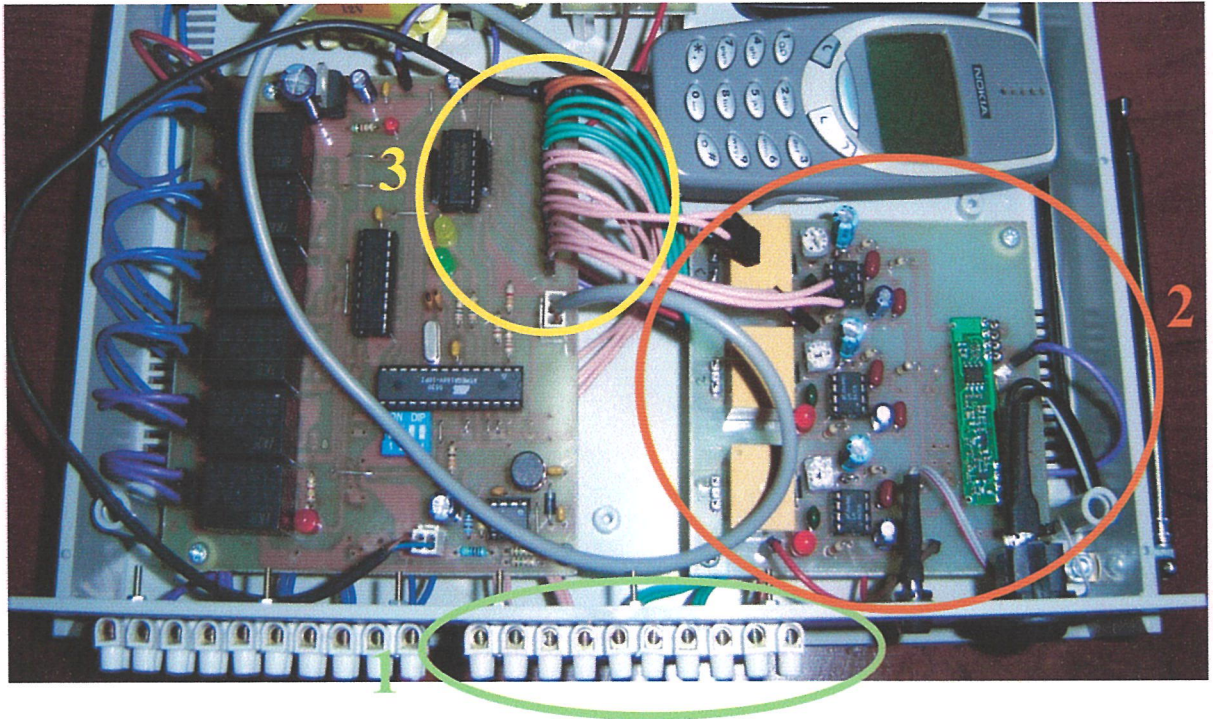
การทดลองวัดสัญญาณ จะเริ่มจากการวัดสัญญาณจากอินพุตที่เข้ามาที่ IC SN74HC165 โดยการรับสัญญาณของตัว IC นี้ ข้อมูลที่รับเข้ามาเป็นแบบขนานในสภาวะปกติ โดยขา A, B, C, D, E, F, G, H จะมีค่าเป็น High แสดงค่าลอจิกคือ 11111111 และเมื่อมีการรับข้อมูลจากภายนอกเข้าที่วงจรถวนคุมเริ่มต้นอินพุตจะทำการเปลี่ยนจาก High เป็น Low เพื่อแสดงถึงการรับข้อมูลในแต่ละส่วนที่มีการเปลี่ยนแปลง



รูปที่ 4.45 แสดงวงจรที่ทำการวัดสัญญาณอินพุต ที่เข้าทาง IC SN74HC165

จากการทดลองจะให้ขา A หรือ ขาที่ 11 มีการรับข้อมูลจากภายนอก และทำการส่งข้อมูลที่รับเข้าที่ไมโครคอนโทรลเลอร์ ดังนั้นก่อนที่จะมีการเปลี่ยนแปลง ทุกขาจะแสดงข้อมูลออกเป็นลักษณะในรูปแบบลอจิกคือ 11111111 คือทุกขาแสดงค่าเป็น High แต่เมื่อขา A มีการเปลี่ยนจากการได้รับข้อมูลจากภายนอก จะเปลี่ยนค่าลอจิกจาก 1 เป็น 0 ซึ่งข้อมูลที่รับเมื่อมีการเปลี่ยนแปลงจะแสดงเป็น 01111111

การทำงานของวงจรในรูปที่ 4.46 เริ่มจากการรับสัญญาณอินพุตที่อยู่กับอุปกรณ์ภายนอก ไม่ว่าจะเป็น การส่งผ่านทางสาย หรือการส่งแบบไร้สาย เมื่อมีสัญญาณเข้ามาก็จะส่งข้อมูลแบบขนานเข้ามาที่ IC SN74CH165F จากนั้นก็จะเข้าที่ไมโครคอนโทรลเลอร์เพื่อทำการประมวลผล



รูปที่ 4.46 วงจรไมโครคอนโทรลเลอร์ที่ต่อกับอุปกรณ์พร้อมทำการทดลอง

ส่วนที่ 1 แสดงอินพุตที่ต่อกับวงจร ไมโครคอนโทรลเลอร์ โดยแต่ละช่องของส่วนรับแสดงถึงช่องรับสัญญาณอินพุตแต่ละช่อง

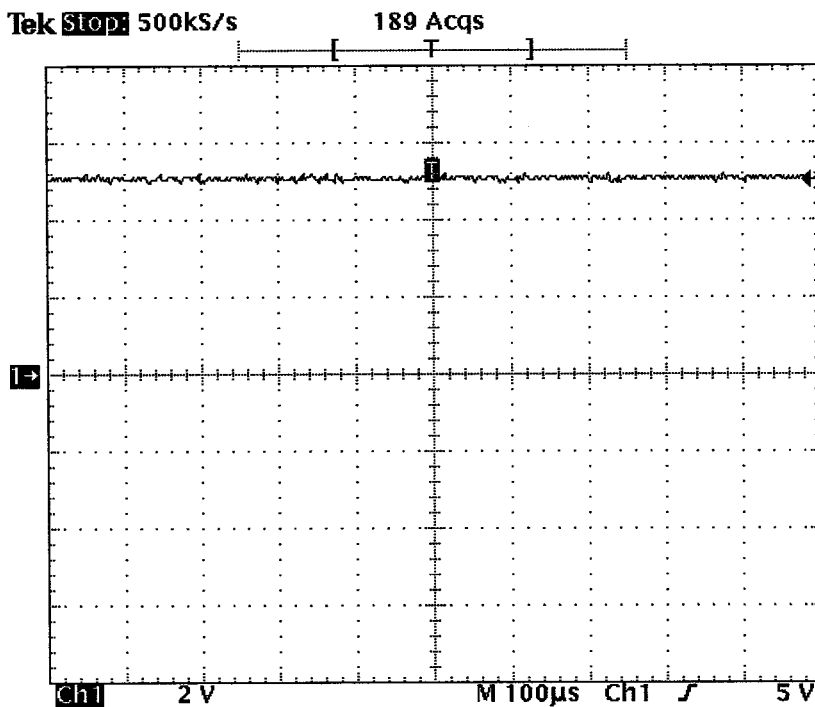
ส่วนที่ 2 แสดงวงจรภาครับสัญญาณ เมื่อมีสัญญาณเข้าก็จะส่งข้อมูลมาที่อินพุตของวงจรไมโครคอนโทรลเลอร์ ในรูปหมายเลข 1 แทนการใช้สาย

ส่วนที่ 3 เมื่อทำการรับข้อมูลจากอินพุตแล้ว ข้อมูลที่ได้จะถูกส่งในรูปแบบขนานเข้า IC SN74HC165 แล้วทำการแปลง เพื่อส่งข้อมูลไปยังไมโครคอนโทรลเลอร์

การทดลองวัดสัญญาณ ดังแสดงในรูปที่ 4.49 และ 4.50 ผลของสัญญาณก่อนการรับอินพุต และหลังการรับอินพุต ของ IC SN74HC165 สังเกตการณ์เปลี่ยนแปลงได้คือ ระดับไฟจาก 5V จะมีการลดระดับลง เป็น 0V

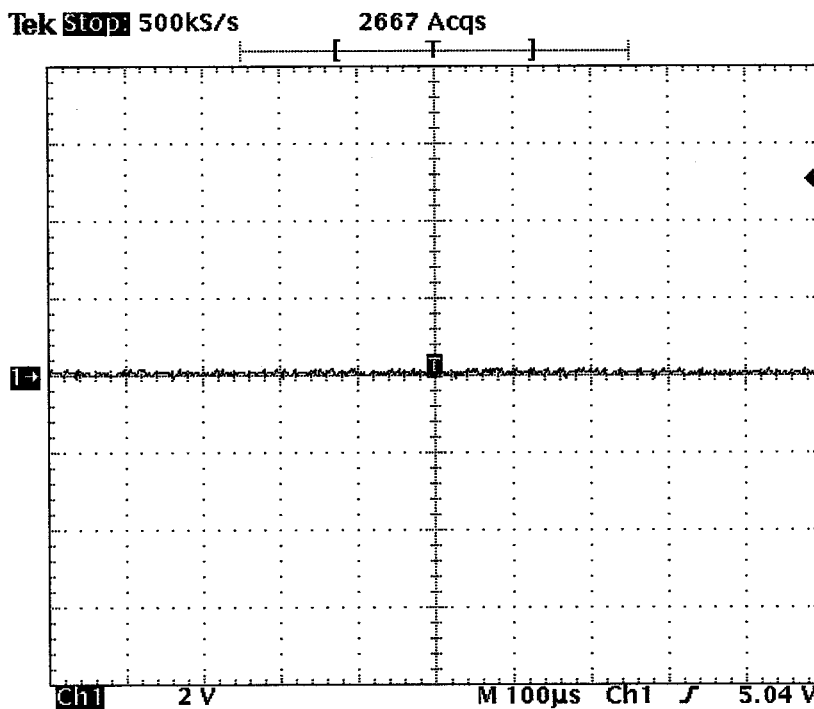
แสดงให้เห็นทราบว่ามีการตอบสนองต่อสัญญาณอินพุตที่เข้ามาในระบบวงจร คือการเปลี่ยนจากค่า High เป็น Low หรือเปลี่ยนจาก 1 เป็น 0 ในทางลอจิก

การทดลองการวัดสัญญาณขณะที่ไม่มีอินพุต และมีอินพุตของขาที่ 11 ของ IC SN74HC165



21 Jan 2009 14:26:08

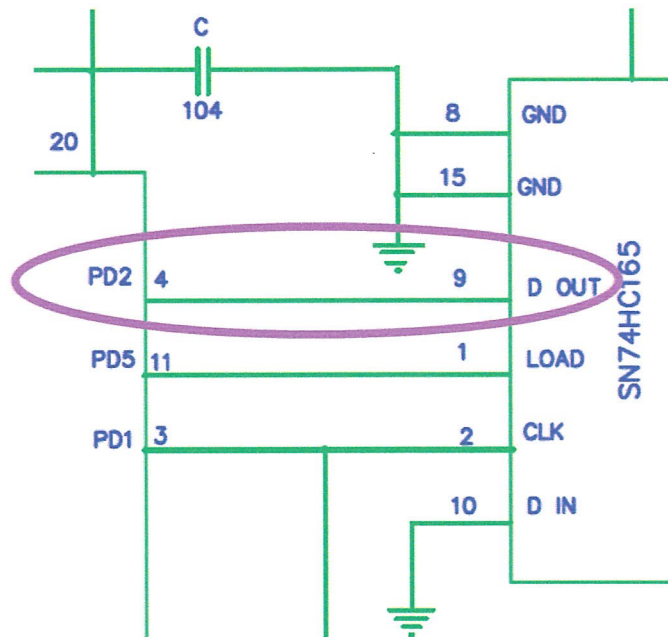
รูปที่ 4.47 สัญญาณขาที่ 11 ของ IC SN74HC165 ขณะที่ยังไม่มีสัญญาณอินพุตเข้ามา



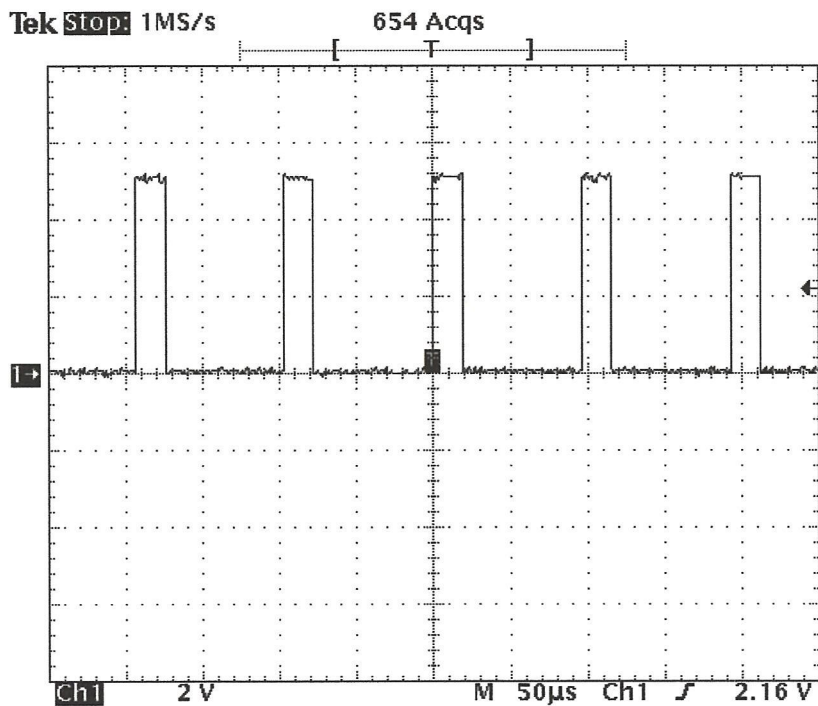
21 Jan 2009 14:56:45

รูปที่ 4.48 สัญญาณขาที่ 11 ของ IC SN74HC165 ขณะที่ มีสัญญาณอินพุตเข้ามา

เมื่อระบบมีการรับอินพุตจากภายนอก IC SN74HC165 จะทำการแปลงข้อมูลที่ได้รับ แล้วส่งข้อมูลที่ได้ออกมาทางขาที่ 9 ( D OUT ) กับขาที่ 4 ( PD2 ) ของไมโครคอนโทรลเลอร์ ในรูปแบบการส่งแบบอนุกรม



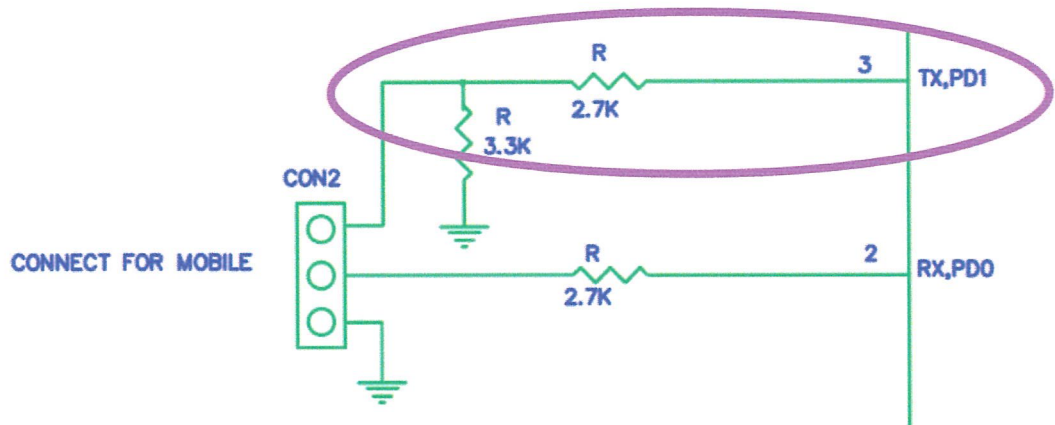
รูปที่ 4.49 แสดงวงจรที่ทำการวัดสัญญาณที่ IC SN74HC165 ส่งข้อมูลไปยังไมโครคอนโทรลเลอร์



21 Jan 2009  
15:00:27

รูปที่ 4.50 สัญญาณขาที่ 9 ของ IC SN74HC165 สัญญาณเอาต์พุตที่แปลงจากแบบขนานเป็นแบบอนุกรม

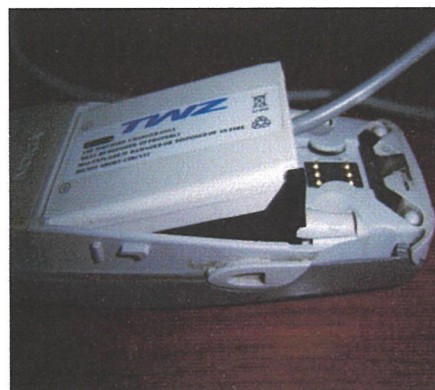
เมื่อไมโครคอนโทรลเลอร์รับค่าอินพุตจาก IC SN74HC165 จะทำการแปลงข้อมูลแล้วส่งออกทางขาที่ 3 (PD1) ของไมโครคอนโทรลเลอร์ ซึ่งเป็นขา Tx เพื่อเชื่อมต่อกับพอร์ตสื่อสารของโทรศัพท์มือถือ โดยผ่าน สายคาต้าลิงค์



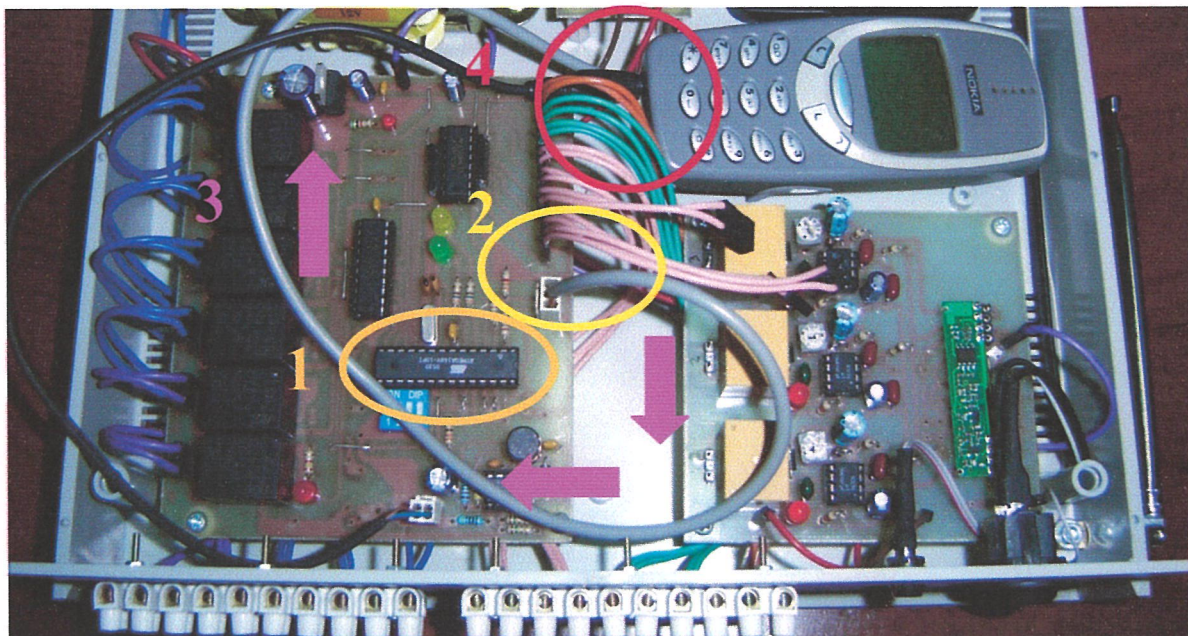
รูปที่ 4.51 แสดงวงจรไมโครคอนโทรลเลอร์ที่ทำการติดต่อพอร์ตสื่อสารกับโทรศัพท์มือถือ



รูปที่ 4.52 พอร์ตสื่อสารของโทรศัพท์มือถือที่อยู่ด้านหลัง



รูปที่ 4.53 สายคาต้าลิงค์ที่เชื่อมต่อกับโทรศัพท์มือถือ



รูปที่ 4.54 วงจรไมโครคอนโทรเลอร์ที่ต่อกับพอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือ

ส่วนที่ 1 ไมโครคอนโทรเลอร์ทำการส่งข้อมูลออกไปทางขาที่ 3 ( PD1 )

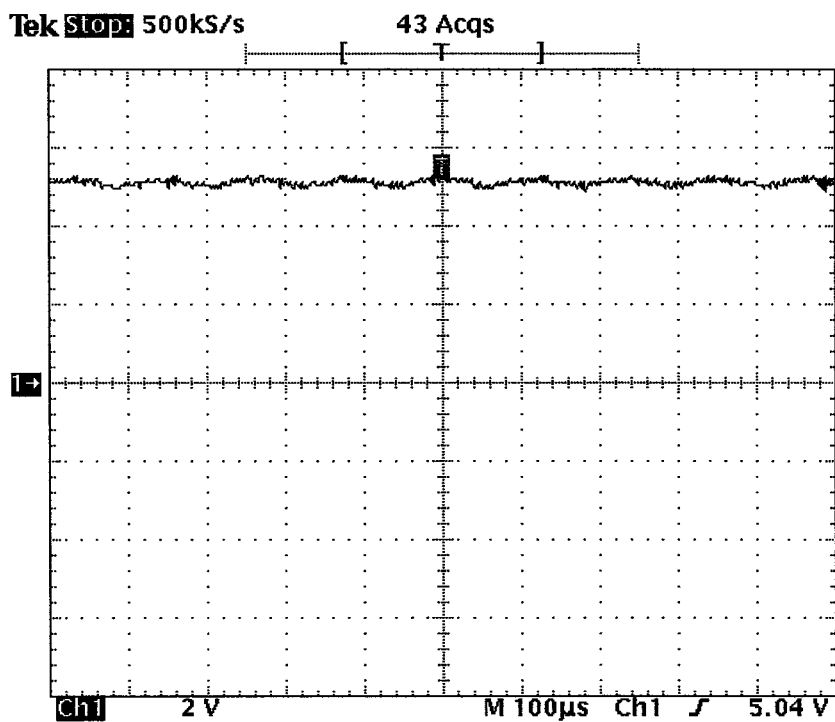
ส่วนที่ 2 แสดงการเชื่อมต่อข้อมูลจากวงจรไมโครคอนโทรเลอร์ ออกไปยังโทรศัพท์มือถือ

ส่วนที่ 3 เส้นทางเชื่อมต่อของวงจรไมโครคอนโทรเลอร์กับโทรศัพท์มือถือ

ส่วนที่ 4 การรับข้อมูลทางพอร์ตสื่อสารของโทรศัพท์มือถือ ที่ได้จากวงจรไมโครคอนโทรเลอร์

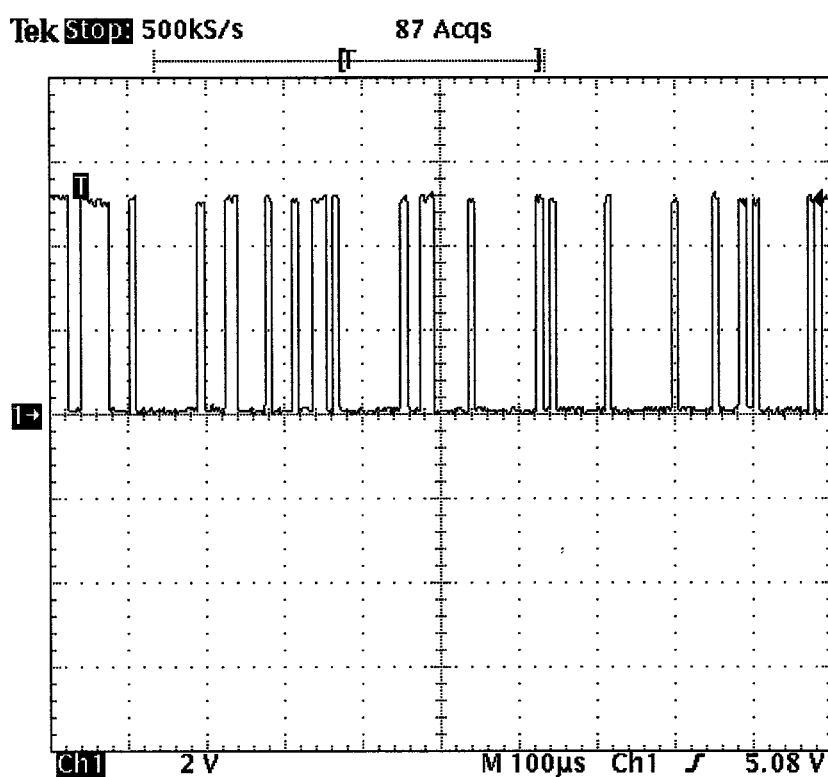
การทดลองวัดสัญญาณข้อมูลที่ออกจากขาที่ 3 ของไมโครคอนโทรเลอร์ โดยวัดในขณะที่มีสัญญาณและไม่มีสัญญาณ ที่ออกจากไมโครคอนโทรเลอร์

แสดงให้เห็นทราบว่า ไมโครคอนโทรเลอร์มีการตอบสนองต่ออินพุตที่รับจาก IC SN74HC165 ทางขาที่ 4 ( PD2 ) และมีการนำข้อมูลที่รับได้แบบอนุกรมนำมาประมวลผล แล้วส่งออกเพื่อแจ้งเตือนให้ทราบทางขาที่ 3 ( PD1 ) เพื่อเข้าทางพอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือ



21 Jan 2009  
15:02:24

รูปที่ 4.55 สัญญาณที่ขาที่ 3 (PD1) ของไมโครคอนโทรลเลอร์ ขณะที่ยังไม่มีสัญญาณออกมา



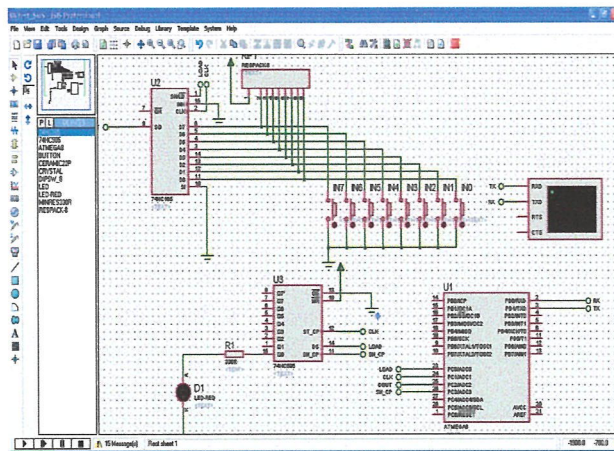
21 Jan 2009  
15:10:55

รูปที่ 4.56 สัญญาณที่ขาที่ 3 (PD1) ของไมโครคอนโทรลเลอร์ ขณะที่เริ่มสัญญาณออกมา

# บทที่ 5 สรุปผลการทดลอง

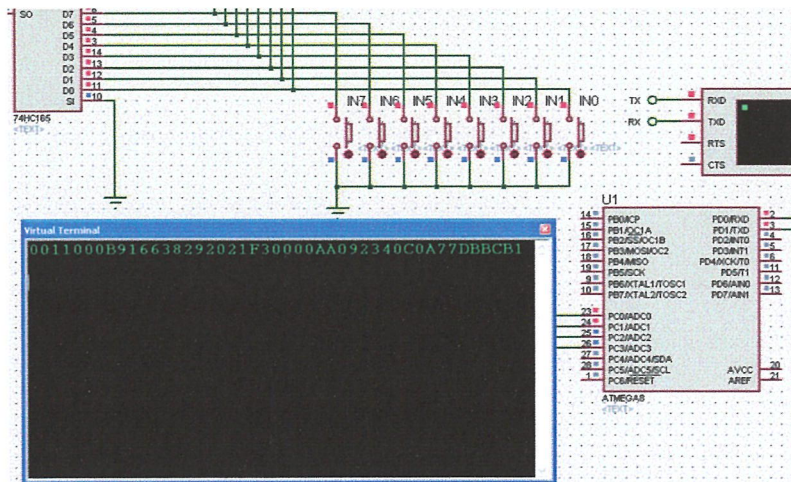
การทดลองได้แบ่งออกเป็นหัวข้อต่างๆ ทั้งในส่วนของภาคโปรแกรม และวงจร ทั้งนี้จึงขอสรุปการทดลองต่างๆตามส่วนของการทดลอง

## 5.1 สรุปผลการทดลองโดยโปรแกรม Proteus



รูปที่ 5.1 แสดงวงจรจำลองที่สร้างขึ้นบนโปรแกรม Proteus

โปรแกรม Proteus คือโปรแกรมที่ทดสอบการทำงานของไมโครคอนโทรลเลอร์ โดยเริ่มจากการเขียนวงจรจำลองขึ้นในโปรแกรม Proteus และจึงนำไฟล์ .hex ที่ Compile ของการเขียน Source Code การทำงานของไมโครคอนโทรลเลอร์ในโปรแกรม WinAVR มาใส่เข้าไปในตัวแบบจำลองการทำงานของวงจร จากนั้นทำการทดลองในโหมด Animating เพื่อแสดงการทำงาน ผลที่ได้ออกมาคือ หน้าจอ Virtual Terminal ในโปรแกรม Proteus จะเป็นตัวแสดงถึงตำแหน่งที่ทำการรับค่าอินพุต ซึ่งจะสามารถแบ่งได้ทั้งหมด 8 ไชน และส่งค่าออกเป็นเลขฐาน 16 ในหมวด PDU เพื่อส่งเข้าที่มีถือต่อไป ทั้งนี้ยังสามารถแสดงให้เห็นว่า ในส่วนที่แตกต่างกันวงจรสามารถตอบสนองและสามารถส่งข้อความเพื่อบ่งบอกในส่วนสถานที่ที่แตกต่างกันเพื่อแจ้งให้ทราบ ตามการออกแบบ ได้อีกด้วย



รูปที่ 5.2 แสดงผลตอบสนองของโปรแกรมเมื่อทำการทดลอง

ลักษณะของผลที่ออกจากโปรแกรมนั้นสามารถแสดงและตีความหมายได้ดังนี้คือ

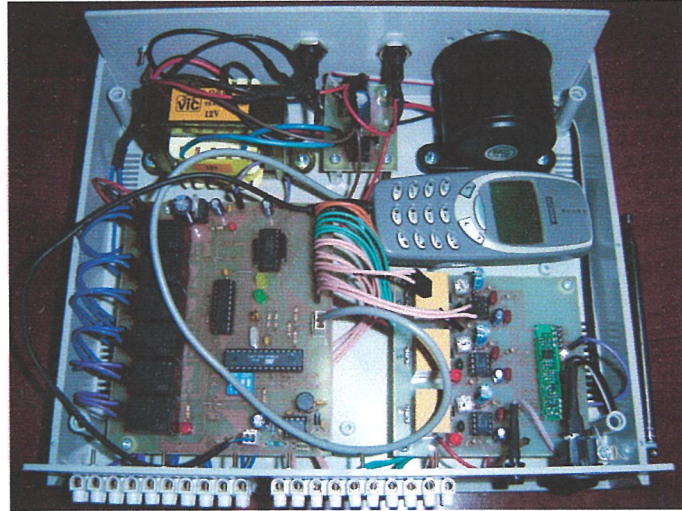
0011000B916638292021F30000AA092340C0A77D BBCB1 ความหมายที่แสดงผลคือ #01>Zone1  
 0011000B916638292021F30000AA092340C0A77D BBCB2 ความหมายที่แสดงผลคือ #01>Zone2  
 0011000B916638292021F30000AA092340C0A77D BBCB3 ความหมายที่แสดงผลคือ #01>Zone3  
 0011000B916638292021F30000AA092340C0A77D BBCB4 ความหมายที่แสดงผลคือ #01>Zone4  
 0011000B916638292021F30000AA092340C0A77D BBCB5 ความหมายที่แสดงผลคือ #01>Zone5  
 0011000B916638292021F30000AA092340C0A77D BBCB6 ความหมายที่แสดงผลคือ #01>Zone6  
 0011000B916638292021F30000AA092340C0A77D BBCB7 ความหมายที่แสดงผลคือ #01>Zone7  
 0011000B916638292021F30000AA092340C0A77D BBCB8 ความหมายที่แสดงผลคือ #01>Zone8

โดย 00 หมายถึงให้ใช้ SMSC Information ที่เก็บอยู่ภายในเครื่อง  
 11 First octet of the SMS-SUBMIT message  
 00 TP-message-Reference โดยที่ 00 คือให้เครื่องตั้งหมายเลขอ้างอิงข้อความขึ้นเอง  
 0B Address-Length คือความยาวของเลขหมายผู้รับ (11 ตัว)  
 91 ใช้แสดงว่าเบอร์โทรศัพท์เคลื่อนที่เป็นแบบภายในประเทศ  
 663829202 F3 เบอร์โทรศัพท์ปลายทางที่ทำการเข้ารหัสแบบสลับแล้ว  
 00 TP-PID (Protocol identifier) ในกรณีนี้เป็น 00  
 00 TP-DCS (Data coding scheme) เป็น 00 คือเข้ารหัสข้อความแบบ 7 บิต  
 AA ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ยังไม่ส่งถึงปลายทาง  
 09 ความยาวของข้อความก่อนแปลงรหัส นั่นคือ "#01>Zone1" = 9 ตัวอักษร  
 2340C0A77DBCB1 ข้อมูลในส่วน of ข้อความตัวอักษรแบบ 7 บิต นั่นคือคำว่า  
 "#01>Zone1" ที่ผ่านการแปลง (เข้ารหัส) เป็นข้อมูลแบบ 8 บิต แล้ว

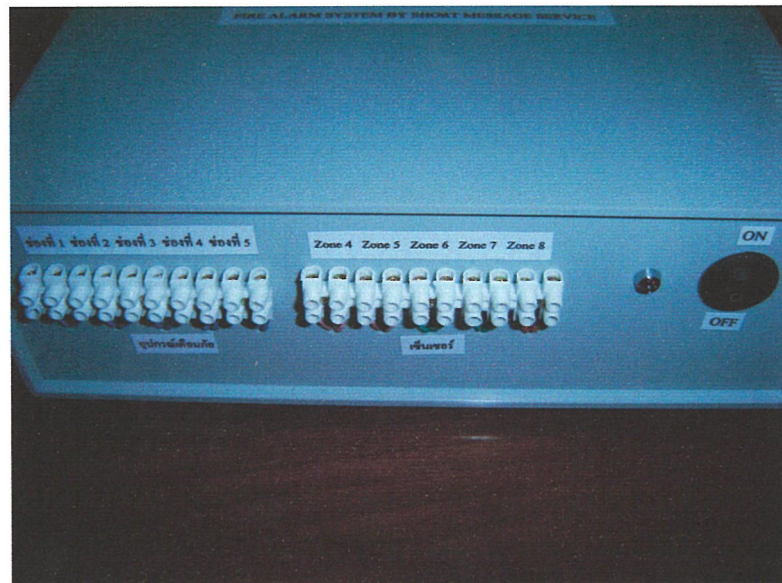
ข้อดีของการทำการทดลองโดยโปรแกรม Proteus คือ ทำให้สามารถทราบว่าผลที่ได้ออกมาจากการทำการทดลองนั้นจะเป็นอย่างไร โดยที่เราไม่จำเป็นต้องทำการทดลองจริง ทั้งนี้เพื่อความสะดวกในการปรับแต่งในส่วนการใช้งานต่างๆให้เกิดประสิทธิภาพดีขึ้น เพราะการที่ต้องนำไมโครคอนโทรลเลอร์มาบันทึก หรือลบข้อมูลใหม่หลายครั้งจะทำให้เสียเวลาในการทดลองมากเนื่องการถอดและประกอบตัวไมโครคอนโทรลเลอร์หลายครั้ง และอาจจะทำให้ขาของไมโครคอนโทรลเลอร์เกิดการหัก ช็อต หรือเสียหายเนื่องจากการถอดและประกอบเข้าไปในวงจรใหม่ อีกทั้งยังแก้ปัญหาในส่วนของบทร่องเกี่ยวกับการไม่ทำงานของวงจร โดยการจำลองการทำงานของวงจรจะทำให้ทราบถึงผลการตอบสนอง และผลที่แสดงผลออกมา การทดลองในส่วนการเขียนโปรแกรมนั้นว่ามีความถูกต้องหรือไม่ เราจึงสามารถที่จะตัดข้อสงสัยนี้ออกไปได้และสามารถมุ่งประเด็นไปในส่วนอื่นได้ คือวงจรอาจเกิดการช็อตกัน ตัวต้านทาน หรือตัวเก็บประจุ เกิดความเสียหาย ทำให้ไม่สามารถที่จะทำงานได้ ดังนั้นในการทดลองการออกแบบวงจรโดยโปรแกรมจึงเป็นข้อดี ก่อนที่จะลงมือปฏิบัติจริง

## 5.2 สรุปผลการทดลองการประกอบวงจร และการวัดผล

การทดลองในส่วนนี้เริ่มจากการประกอบวงจรควบคุมไมโครคอนโทรลเลอร์ วงจรระดับแรงดัน วงจรรีโมทควบคุมการเปิด – ปิดอุปกรณ์ ทั้งภาคส่งและภาครับและส่วนที่ทำการเชื่อมต่อกับโทรศัพท์มือถือ ลงในแผ่น PCB และทำการประกอบอุปกรณ์ทั้งหมดลงกล่อง ดังแสดงในรูป

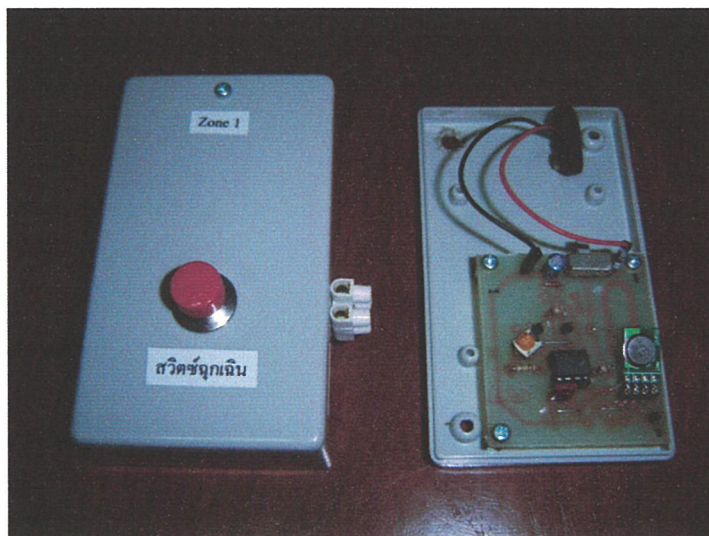


รูปที่ 5.3 การนำอุปกรณ์ทั้งหมดลงในกล่องใส่อุปกรณ์เพื่อความเรียบร้อย



รูปที่ 5.4 การประกอบวงจรควบคุมที่สมบูรณ์

ในขั้นตอนต่อมา จะเริ่มทำการทดลองโดยการต่อวงจร ไมโครคอนโทรลเลอร์กับพอร์ตสื่อสารของโทรศัพท์มือถือและทำการส่งอินพุตเข้าที่เซนเซอร์วงจรไมโครคอนโทรลเลอร์ ซึ่งได้ทำการออกแบบในการรับค่าได้มากที่สุด 8 ช่องในการทำงาน โดยแบ่งออกเป็นแบบใช้สาย 5 ช่องและแบบไร้สาย 3 ช่อง โดยใช้สัญญาณวิทยุแทน เพื่อเพิ่มการครอบคลุมพื้นที่ในการตรวจจับการเกิดอัคคีภัยได้มากขึ้น



รูปที่ 5.5 วงจรภาคส่งที่นำอุปกรณ์ลงกล่องเรียบร้อยแล้ว

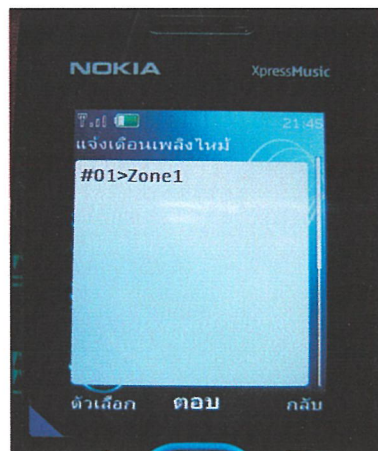


รูปที่ 5.6 วงจรภาคส่งที่ติดตั้งพร้อมใช้งาน



รูปที่ 5.7 แสดงการต่อวงจรพร้อมใช้งาน และข้อความที่ได้รับเมื่อมีการแจ้งเตือน

เมื่อมีการส่งสัญญาณจากตัวรับค่าจากอินพุตส์ที่ทำการตรวจจับอยู่ ข้อมูลที่ส่งเข้าที่ไมโครคอนโทรลเลอร์จะเป็นแบบขนาน เมื่อตัวไมโครคอนโทรลเลอร์รับก็จะแปลงรหัสแล้วส่งออกไปยังขาที่ 3 เข้าที่พอร์ตอนุกรมสื่อสารของโทรศัพท์มือถือ และส่งข้อความไปยังโทรศัพท์มือถือปลายทางเพื่อทำการแจ้งเตือนให้ทราบได้ทันที การแจ้งเตือนผ่านโทรศัพท์มือถือสามารถระบุตำแหน่งของบริเวณที่เกิดเหตุได้



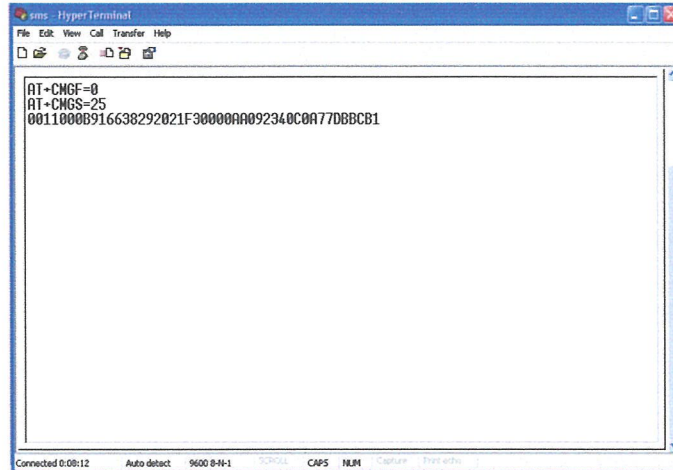
รูปที่ 5.8 แสดงข้อความที่ได้รับ

โดยข้อความที่ได้รับคือ #01>Zone01 แบ่งอธิบายได้ 2 ส่วนคือคำว่า #01 หมายถึงหมายเลขของเครื่องเตือนภัยว่ามาจากเครื่องใด และ Zone1 หมายถึงส่วนของเซนเซอร์ที่ทำการตรวจจับการเกิดเหตุเพลิงไหม้ นั้นอยู่ในส่วนที่ 1 ซึ่งในการออกแบบสามารถแบ่งได้ถึง 8 โชน ข้อความส่วนนี้จึงมีขอบเขตตั้งแต่ Zone1 ถึง Zone 8 ตามค่าเซนเซอร์ที่สามารถรับได้

จากผลการทดลองทั้งในส่วนของวงจรเซนเซอร์ที่ติดกับตัววัด และวงจรส่งสัญญาณไร้สาย ที่ทำการส่งสัญญาณมาให้เซนเซอร์ ผลที่ได้เป็นไปตามที่ออกแบบไว้ ทั้งนี้การตอบสนอง และการประมวลผลของวงจรไมโครคอนโทรลเลอร์ สามารถแสดงตำแหน่งของการเกิดในแต่ละส่วนได้อย่างถูกต้อง และมีการส่งข้อความแจ้งเตือนมายังโทรศัพท์มือถือปลายทางเพื่อแจ้งเหตุอีกด้วย

### 5.3 สรุปผลการทดสอบผ่านโปรแกรม HyperTerminal

การทดสอบผ่านโปรแกรม HyperTerminal เพื่อทำการตรวจสอบข้อความที่วงจรถูกคุม หรือ ไมโครคอนโทรลเลอร์ส่งค่าผ่านอนุกรมสื่อสารเพื่อทำการติดต่อกับโทรศัพท์มือถือด้วยคำสั่ง AT Command



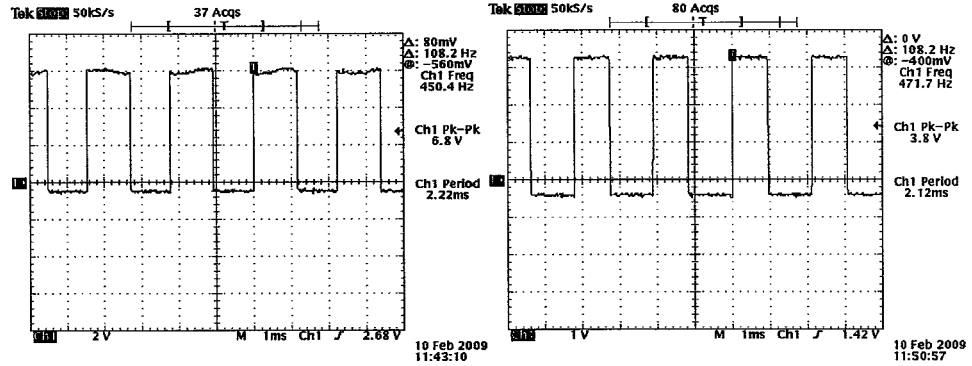
รูปที่ 5.9 แสดงค่าที่สามารถรับได้จากการส่งของไมโครคอนโทรลเลอร์กับโทรศัพท์มือถือ

ข้อความที่สามารถรับได้จากโปรแกรม Hyper Terminal นี้จะอยู่ในรูปแบบของการเข้ารหัสแบบ PDU Mode แสดงข้อความที่ได้รับคือ 0011000B916638292021F30000AA092340C0A77DBBCB1 โดยถอดรหัสออกมาแล้วจะให้ความหมายคือคำว่า #01>Zone1

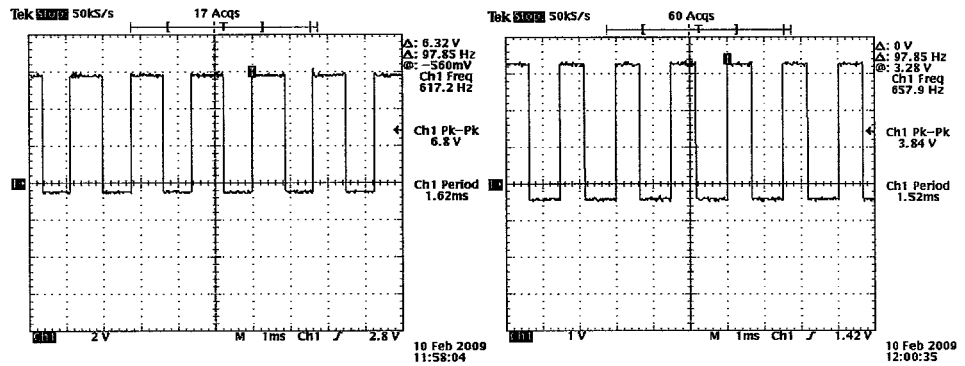
การทดสอบโดยโปรแกรม Hyper Terminal สามารถทราบถึงข้อความที่ส่งออกไปยังหมายเลข หรือโทรศัพท์ที่ต้องการ เพราะเป็นโปรแกรมที่สามารถเชื่อมต่อเข้ากับพอร์ตสื่อสารได้ทุกชนิด นอกจากนี้ยังสามารถรับ – ส่งข้อความ ไฟล์ หรือสามารถอำนวยความสะดวกในการสื่อสารได้อีกด้วย ทำให้ทราบถึงความเป็นไปในการส่งว่าข้อมูลที่เรากำลังส่งนั้นมีการแปลงเป็นรูปแบบอะไรบ้าง นอกจากนี้ยังสามารถบันทึกรูปแบบการส่งข้อความกลับเข้ามาได้อีกด้วย การทดสอบโดยโปรแกรม Hyper Terminal จึงเป็นข้อดีของการสื่อสารแบบอนุกรม

### 5.4 สรุปผลการวัดสัญญาณ ของวงจรรับ – ส่ง สัญญาณ

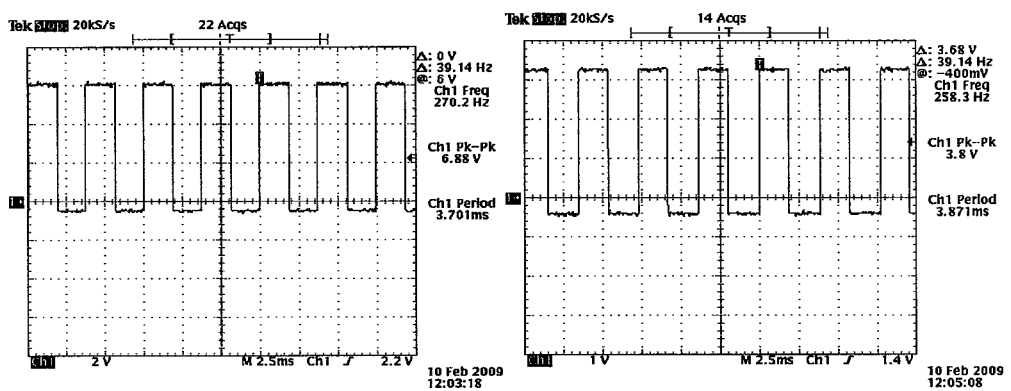
การทดลองวัดสัญญาณในส่วนแรกจะเริ่มวัดสัญญาณที่ออกจาก IC HA17555 ของวงจรภาคส่ง และนำสัญญาณมาเปรียบเทียบกับวงจรภาครับทุกตัว



รูปที่ 5.10 แสดงการเปรียบเทียบสัญญาณภาคส่ง (ซ้าย) และภาครับ (ขวา) ของภาคส่งตัวที่ 1

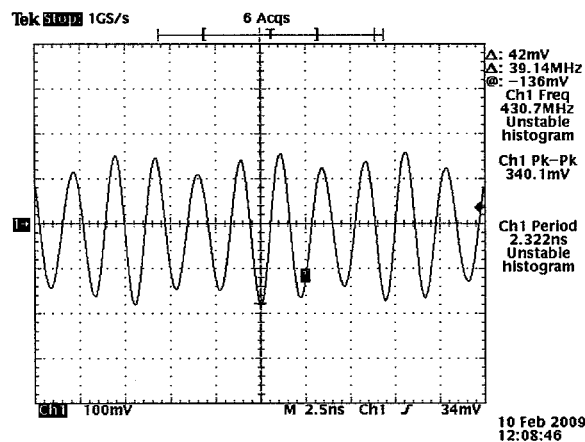


รูปที่ 5.11 แสดงการเปรียบเทียบสัญญาณภาคส่ง (ซ้าย) และภาครับ (ขวา) ของภาคส่งตัวที่ 2

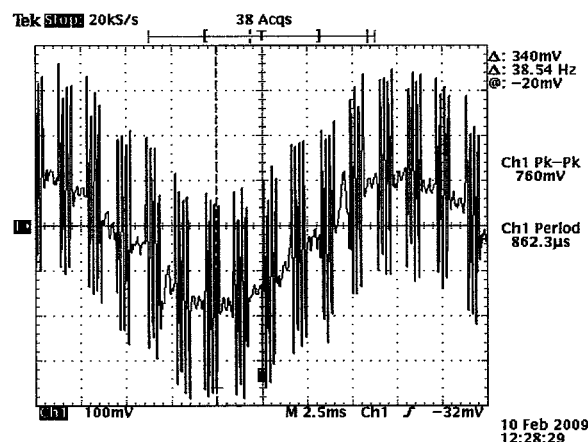


รูปที่ 5.12 แสดงการเปรียบเทียบสัญญาณภาคส่ง (ซ้าย) และภาครับ (ขวา) ของภาคส่งตัวที่ 3

### การทดสอบการวัดจากสัญญาณที่ส่งออกทางสายอากาศของภาคส่ง



รูปที่ 5.13 สัญญาณที่ส่งออกทางสายอากาศขณะยังไม่มีการมอดูเลต



รูปที่ 5.14 สัญญาณที่ส่งออกทางสายอากาศเมื่อมีการมอดูเลต

ค่าความถี่การส่งคลื่นวิทยุของภาครับนั้นจะมีความแตกต่างกัน โดยเครื่องวัดสัญญาณสามารถวัดความถี่ของคลื่นที่วงจรภาคส่งด้วยความถี่ 430 MHz อยู่ในช่วงย่านความถี่ของ UHF (ระหว่าง 300 - 3000 MHz) โดยการใช้ไมโคร TWS - 434 สร้างคลื่นพาหะที่มีการมอดูเลตแบบ ASK ความถี่ที่แตกต่างกันของแต่ละช่องสัญญาณนี้มาจากการเปลี่ยนค่า R ของวงจรภาคส่งซึ่งค่า R ที่ใช้นั้นคือ 100K, 200K และการปรับค่า VR เนื่องจากค่าที่แตกต่างกันทำให้กระแสที่เข้าไปใน IC HA17555 ต่างกัน ความถี่ที่ได้จึงมีค่าแตกต่างกัน และช่องสัญญาณที่รับได้นั้น เกิดจากการตั้งความถี่ของวงจรภาคส่ง และภาครับให้ตรงกันจึงสามารถที่จะรับ - ส่ง สัญญาณในแต่ละช่องได้

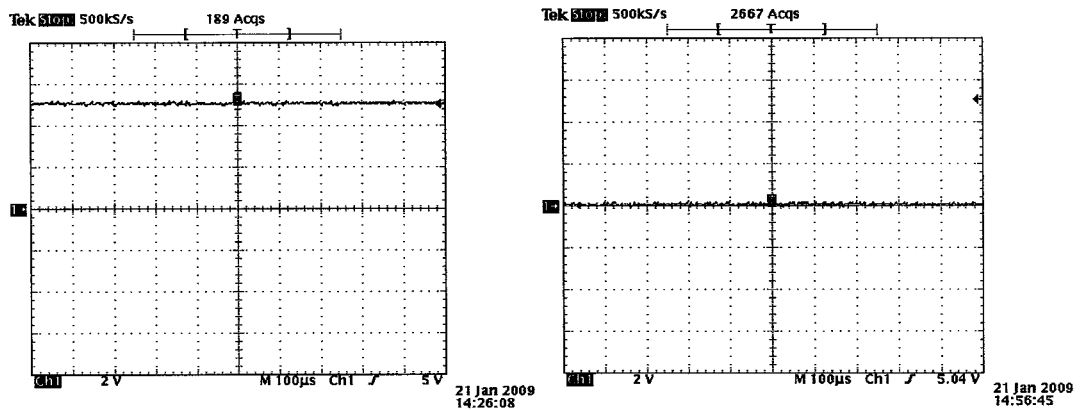
จากการเปรียบเทียบการวัดสัญญาณทั้งทางด้านส่ง และทางด้านรับพบว่า สัญญาณที่รับได้จะมีรูปร่างลักษณะของสัญญาณที่ใกล้เคียงกับสัญญาณเดิม โดยภาครับจะใช้ไมโครรับสัญญาณ RLP - 434 มาเป็นตัวรับสัญญาณและทำการแยกสัญญาณข่าวสารกับสัญญาณพาหะออกจากกัน จากนั้นวงจรภาครับจะส่งสัญญาณไปยังวงจรถอดรหัส โดยใช้ IC NE567N เป็นตัวถอดรหัสสัญญาณ โดยมีค่า R ขนาด 100K, 200K, 300K และ VR เข้ามาเป็นตัวกำหนดในการถอดค่าความถี่ที่แตกต่างกัน จากนั้นจะส่ง

สัญญาณควบคุมการทำงานของรีเลย์แยกตามความถี่ที่ได้รับได้ไปยังช่องสัญญาณต่างๆ และส่งเข้าที่ขาอินพุตของวงจรควบคุมเพื่อส่งไปยังไมโครคอนโทรลเลอร์

ระยะทางของภาคส่ง และภาครับ จากการทดลอง พบว่าระยะทางที่ไกลที่สุดที่สามารถรับ และส่งได้ในพื้นที่ไม่มีสิ่งกีดขวางนั้นอยู่ที่ 50 เมตรและในบริเวณที่มีสิ่งกีดขวาง เช่น กำแพงตึก อยู่ที่ 30 เมตร

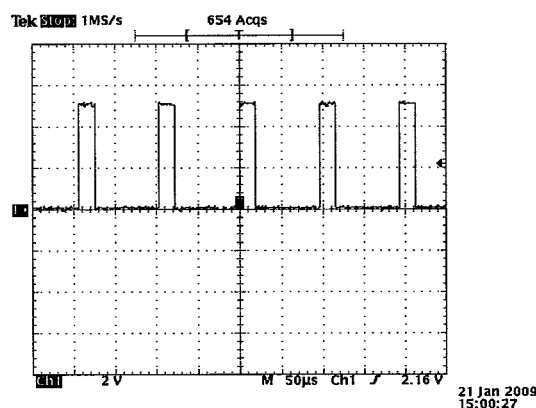
#### 5.4 สรุปผลการวัดสัญญาณ ของวงจรควบคุมไมโครคอนโทรลเลอร์

การทดลองวัดสัญญาณจะเริ่มจากการวัดสัญญาณจากอินพุตที่เข้ามาที่ IC SN74HC165 โดยการเปรียบเทียบสัญญาณ ในขณะที่ไม่มีอินพุตเข้ามาจะมีค่าเป็น High และเมื่อมีสัญญาณเข้ามาจะมีค่าเป็น Low



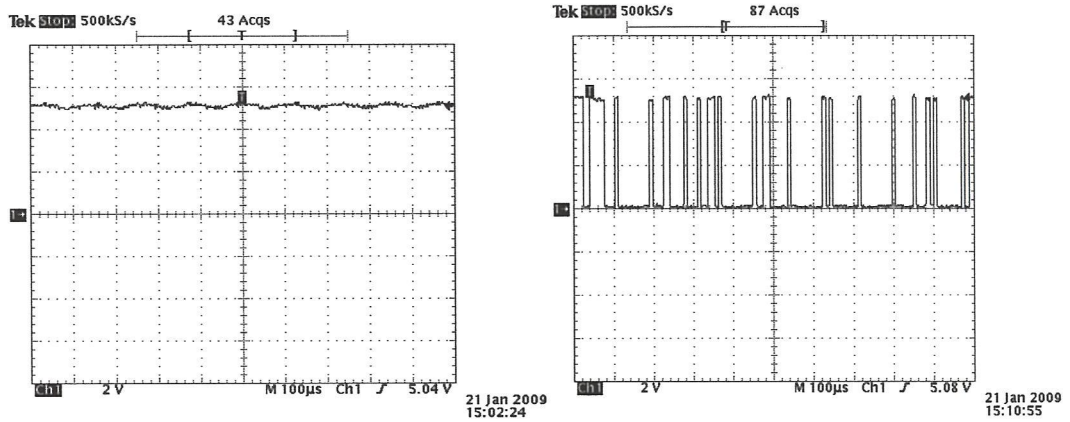
รูปที่ 5.15 แสดงการเปรียบเทียบสัญญาณเมื่อ ไม่มีอินพุตเข้ามา (ซ้าย) และเมื่อมีสัญญาณอินพุตเข้ามา (ขวา)

เมื่อมีการรับสัญญาณเข้ามาที่ IC SN74HC165 จะเป็นการรับข้อมูลแบบขนาน ดังนั้นจะต้องทำการแปลงสัญญาณแบบขนานให้เป็นสัญญาณแบบอนุกรมโดยสัญญาณที่ได้จะทำการส่งออกทางขา 9 ของ IC SN74HC165 เข้าไมโครคอนโทรลเลอร์เพื่อทำการประมวลผล



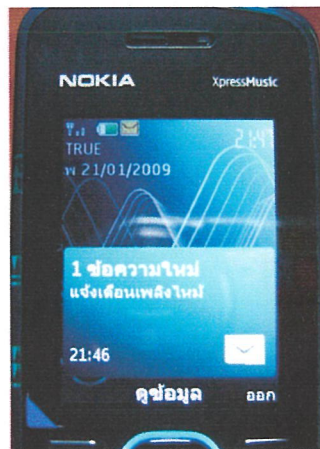
รูปที่ 5.16 สัญญาณที่ส่งออกไปยังไมโครคอนโทรลเลอร์ในแบบอนุกรม

เมื่อไมโครคอนโทรเลอร์รับค่าอินพุตจาก IC SN74HC165 จะทำการแปลงข้อมูลแล้วส่งออกมาที่ 3 (PD1) ของไมโครคอนโทรเลอร์ ซึ่งเป็นขา Tx เพื่อเชื่อมต่อกับพอร์ตสื่อสารของโทรศัพท์มือถือ โดยการเปรียบเทียบ จะใช้สัญญาณในขณะที่ไม่มีการรับค่าสัญญาณ และเมื่อมีการรับค่าสัญญาณจากอินพุต



รูปที่ 5.17 แสดงการเปรียบเทียบสัญญาณเมื่อไม่มีเอาต์พุต (ซ้าย) และเมื่อมีสัญญาณเอาต์พุต (ขวา) ออกจากไมโครคอนโทรเลอร์

โทรศัพท์ที่ต่ออยู่กับวงจรไมโครคอนโทรเลอร์นั้นจะทำการรับข้อมูล แล้วส่งข้อความไปยังผู้ใช้ปลายทางเพื่อเป็นการแจ้งเตือนในส่วนที่เกิดเหตุต่างๆ



รูปที่ 5.18 แสดงการได้รับข้อความของโทรศัพท์ปลายทาง

สรุปการทดลองในส่วนของวงจรไมโครคอนโทรเลอร์นั้น วงจรสามารถตอบสนองต่ออินพุตที่ทำการทดลองในจุดต่างๆ และส่งข้อความไปยังผู้ใช้ปลายทางได้อย่างถูกต้องทั้ง 8 จุด โดยในการต่ออินพุตของวงจรไมโครคอนโทรเลอร์จะต่อเชื่อมกับวงจรรับ - ส่งข้อมูลไร้สายไปตามจุดต่างๆ โดยที่เราไม่ต้องใช้สายทั้งนี้เพื่อความสะดวกและป้องกันการผิดพลาดเนื่องจากการชำรุดหรือความเสียหายของสาย

## 5.4 แนวทางการพัฒนาและวิจารณ์ผลการทดลอง

### 5.4.1 เกี่ยวกับการเขียนโปรแกรมไมโครคอนโทรลเลอร์

สามารถสรุปเป็นข้อได้ดังนี้คือ

- การพัฒนาในส่วนของโปรแกรม คือ สามารถบันทึกเบอร์โทรศัพท์ หรือเปลี่ยนเลขหมายโดยไม่ต้องแก้ไขโปรแกรมในตัวไมโครคอนโทรลเลอร์
- สามารถรายงานผลของจุดที่เกิดเหตุพร้อมกันได้ทีละหลายๆจุด
- เพิ่มหมายเลขผู้ใช้ปลายทาง เพื่อแจ้งเตือนเมื่อเกิดเหตุเพลิงไหม้ได้มากกว่า 1 หมายเลข
- ทำการเชื่อมต่อกับสถานีดับเพลิงเพื่อแจ้งเตือนได้โดยตรงโดยผู้ใช้ไม่จำเป็นต้องติดต่อด้วยตัวเอง

### 5.4.2 เกี่ยวกับการพัฒนาในส่วนของวงจรภาครับ – ส่งสัญญาณ

สามารถสรุปเป็นข้อได้ดังนี้คือ

- เพิ่มประสิทธิภาพของภาครับ – ภาคส่งให้มีระยะทางที่มากกว่าเดิม
- เซนเซอร์ทุกตัวสามารถใช้การสื่อสารแบบไร้สายแทนการใช้สายเพื่อความสะดวก และป้องกันการเสียหายของสาย

### 5.4.3 เกี่ยวกับการพัฒนาในส่วนของวงจรควบคุมไมโครคอนโทรลเลอร์

สามารถสรุปเป็นข้อได้ดังนี้คือ

- เพิ่มช่องรับเซนเซอร์ให้มีจำนวนที่มากขึ้นเพื่อเพิ่มพื้นที่ในการตรวจจับได้มากขึ้น
- ติดตั้งเซนเซอร์เพิ่ม เช่น อุปกรณ์ตรวจจับแก๊ส อุปกรณ์ตรวจจับความร้อน อื่นๆ

## กิตติกรรมประกาศ

ในการทำรายงานวิชา Project II เรื่อง ระบบแจ้งเหตุเพลิงไหม้ โคนแจ้งเตือนผ่านข้อความสั้น ( FIRE ALARM SYSTEM BY SHORT MESSAGE SERVICE ) จะไม่สามารถสำเร็จลุล่วงไปได้หากไม่ได้รับคำปรึกษา และคำแนะนำ ในการทำงานจาก อาจารย์ที่ปรึกษา ผศ.ดร. สุทธิชัย นพนาศิพงษ์ รวมทั้งอาจารย์ท่านอื่นที่ไม่ได้กล่าวมา ณ ที่นี้

นอกจากนี้รายงานจะไม่สมบูรณ์ หากมิได้ผู้ที่เกี่ยวข้องบุคคลอื่น อาทิเช่น เพื่อน รุ่นพี่ภาควิชา โทรคมนาถม และภาควิชาอื่นๆทุกท่าน ที่ให้คำแนะนำในการทดลอง การเขียนโปรแกรม รวมถึงการ ออกแบบวงจร ซึ่ง ทั้งนี้ทางผู้จัดทำขอขอบคุณในทุกคำแนะนำมา ณ ที่นี้

ในบางส่วนของรายงานอาจมีความผิดพลาด ทางผู้จัดทำต้องขออภัยต่อความผิดพลาดที่เกิดขึ้น และหวังเป็นอย่างยิ่งว่า รายงานฉบับนี้จะเป็นประโยชน์แก่นักศึกษารุ่นถัดไปของทางสถาบัน ไม่มากนัก น้อย ที่สนใจในการทำในเรื่องนี้ หรือต้องการนำเรื่องนี้ไปประยุกต์ใช้ในครั้งต่อไป

## บรรณานุกรม

- [1] ประจัน พลังสันติกุล – **C Programming for AVR Microcontroller and WinAVR ( C Compiler )**  
เล่ม 1
- [2] ประจัน พลังสันติกุล – **C Programming for AVR Microcontroller and WinAVR ( C Compiler )**  
เล่ม 2
- [3] Labcenter Electronics – **Intelligent Schematic Input System**
- [4] Claus Kuhnel – **AVR RISC Microcontroller handbook**
- [5] กำธร เรือนฝ้ายการ – **การใช้งาน AVR**
- [6] ชัชวาล สุขเกษม - **AutoCAD 2008i : ปฏิบัติการงานเขียนแบบ**
- [7] สุรพล เลิศดำรงชัย - **AutoCAD 2006i : tips & tricks**
- [8] ภาณุพงษ์ ปัดติสิงห์ - **คู่มือการใช้โปรแกรม AutoCAD 2006 : 2D drafting (ระดับต้น)**
- [9] อภิรัตน์ บางศิริ, ชัยณัฐพงศ์ ชาญญลักษณ์ - **เขียนแบบทางวิศวกรรมและสถาปัตยกรรมด้วย AutoCAD 2008 เรียนลัดเป็นเร็ว : ประยุกต์ใช้ได้กับเวอร์ชัน 2002-2008**
- [10] วิวัฒน์ กิรานนท์ – **วิเคราะห์วงจรไฟฟ้า**
- [11] วิวัฒน์ กิรานนท์ - **พื้นฐานการสื่อสาร Data Communicaion -**
- [12] วิวัฒน์ กิรานนท์ - **การสื่อสารข้อมูลดิจิทัล Digital data communication**
- [13] บัณฑิต จามรฤติ - **คู่มือการใช้ Protel for Windows**
- [14] Arnaud Henry-Labordere, Vincent Jonack - **SMS and MMS interworking in mobile networks**
- [15] นิกร สุขุมสันติ – **อิเล็กทรอนิกส์พื้นฐาน Basic electronics**
- [16] **ข้อมูลเกี่ยวกับพื้นฐาน IC จาก**

<http://www.alldatasheet.com/>

<http://www.datasheetcatalog.com/>

## ภาคผนวก 1

## Features

### High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller

#### Advanced RISC Architecture

- 131 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 20 MIPS Throughput at 20 MHz
- On-chip 2-cycle Multiplier

#### High Endurance Non-volatile Memory segments

- 4/8/16K Bytes of In-System Self-programmable Flash program memory
- 256/512/512 Bytes EEPROM
- 512/1K/1K Bytes Internal SRAM
- Write/Erase cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
- Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
- Programming Lock for Software Security

#### Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Six PWM Channels
- 8-channel 10-bit ADC in TQFP and QFN/MLF package
- 6-channel 10-bit ADC in PDIP Package
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

#### Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

#### I/O and Packages

- 23 Programmable I/O Lines
- 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

#### Operating Voltage:

- 1.8 - 5.5V for ATmega48V/88V/168V
- 2.7 - 5.5V for ATmega48/88/168

#### Temperature Range:

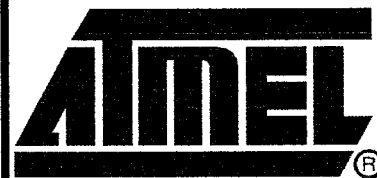
- -40°C to 85°C

#### Speed Grade:

- ATmega48V/88V/168V: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
- ATmega48/88/168: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V

#### Low Power Consumption

- Active Mode:
  - 250 µA at 1 MHz, 1.8V
  - 15 µA at 32 kHz, 1.8V (Including Oscillator)
- Power-down Mode:
  - 0.1µA at 1.8V



8-bit **AVR<sup>®</sup>**  
Microcontroller  
with 8K Bytes  
In-System  
Programmable  
Flash

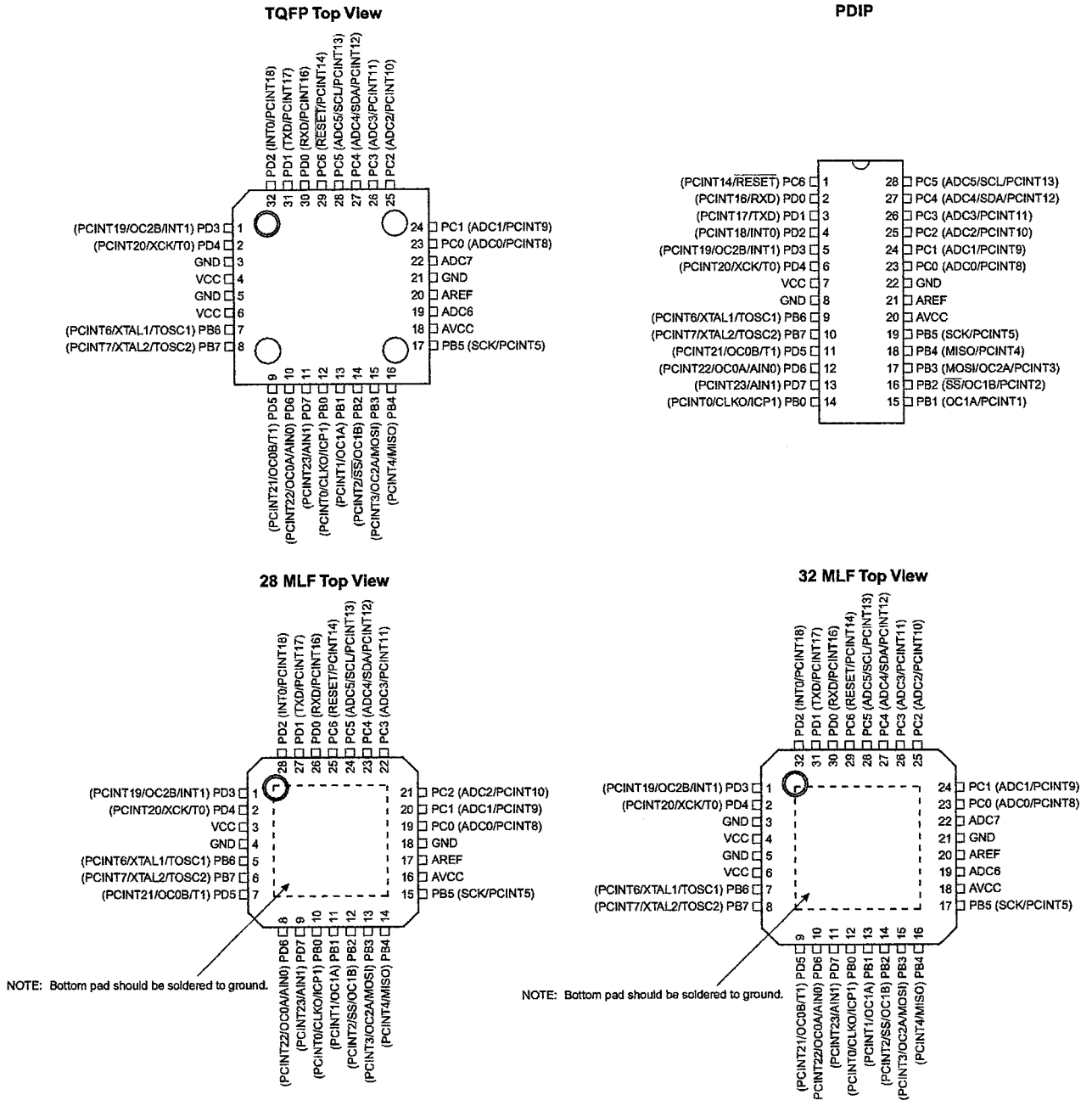
**ATmega48/V**  
**ATmega88/V**  
**ATmega168/V**

Rev. 2545M-AVR-09/07



# I. Pin Configurations

Figure 1-1. Pinout ATmega48/88/1682545M



## Theory and Applications of the MC34063 and $\mu$ A78S40 Switching Regulator Control Circuits



ON Semiconductor®

<http://onsemi.com>

Prepared by: Jade Alberkrack  
ON Semiconductor

### APPLICATION NOTE

This paper describes in detail the principle of operation of the MC34063 and  $\mu$ A78S40 switching regulator subsystems. Several converter design examples and numerous applications circuits with test data are included.

#### INTRODUCTION

The MC34063 and  $\mu$ A78S40 are monolithic switching regulator subsystems intended for use as dc to dc converters. These devices represent a significant advancement in the ease of implementing highly efficient and yet simple switching power supplies. The use of switching regulators is becoming more pronounced over that of linear regulators because the size reductions in new equipment designs require greater conversion efficiency. Another major advantage of the switching regulator is that it has increased application flexibility of output voltage. The output can be less than, greater than, or of opposite polarity to that of the input voltage.

#### PRINCIPLE OF OPERATION

In order to understand the difference in operation between linear and switching regulators we must compare the block diagrams of the two step-down regulators shown in Figure 1. The linear regulator consists of a stable reference, a high gain error amplifier, and a variable resistance series-pass element. The error amplifier monitors the output voltage level, compares it to the reference and generates a linear control signal that varies between two extremes, saturation and cutoff. This signal is used to vary the resistance of the series-pass element in a corrective fashion in order to maintain a constant output voltage under varying input voltage and output load conditions.

The switching regulator consists of a stable reference and a high gain error amplifier identical to that of the linear regulator. This system differs in that a free running oscillator and a gated latch have been added. The error amplifier again monitors the output voltage, compares it to the reference level and generates a control signal. If the output voltage is below nominal, the control signal will go to a high state and turn on the gate, thus allowing the oscillator clock pulses to drive the series-pass element alternately from cutoff to saturation. This will continue until the output voltage is pumped up slightly above its nominal value. At this time, the

control signal will go low and turn off the gate, terminating any further switching of the series-pass element. The output voltage will eventually decrease to below nominal due to the presence of an external load, and will initiate the switching process again. The increase in conversion efficiency is primarily due to the operation of the series-pass element only in the saturated or cutoff state. The voltage drop across the element, when saturated, is small as is the dissipation. When in cutoff, the current through the element and likewise the power dissipation are also small. There are other variations of switching control. The most common are the fixed frequency pulse width modulator and the fixed on-time variable off-time types, where the on-off switching is uninterrupted and regulation is achieved by duty cycle control. Generally speaking, the example given in Figure 1b does apply to MC34063 and  $\mu$ A78S40.

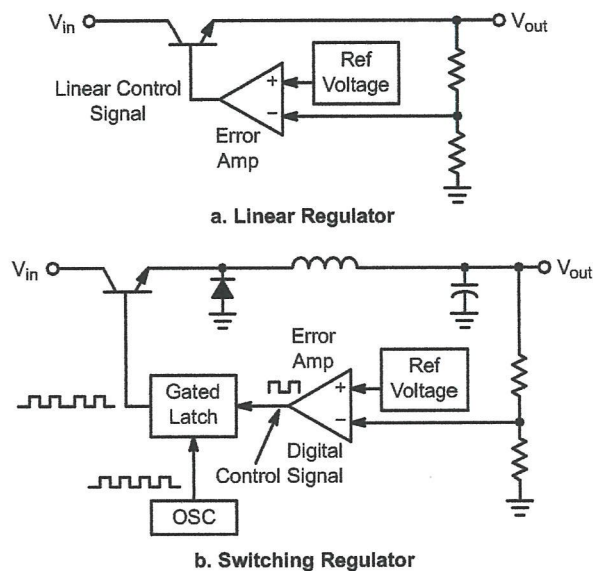


Figure 1. Step-Down Regulators

**GENERAL DESCRIPTION**

The MC34063 series is a monolithic control circuit containing all the active functions required for dc to dc converters. This device contains an internal temperature compensated reference, comparator, controlled duty cycle oscillator with an active peak current limit circuit, driver, and a high current output switch. This series was specifically designed to be incorporated in step-up, step-down and voltage-inverting converter applications. These functions are contained in an 8-pin dual in-line package shown in Figure 2a.

The  $\mu A78S40$  is identical to the MC34063 with the addition of an on-board power catch diode, and an uncommitted operational amplifier. This device is in a 16-pin dual in-line package which allows the reference and the noninverting input of the comparator to be pinned out. These additional features greatly enhance the flexibility of this part and allow the implementation of more sophisticated applications. These may include series-pass regulation of the main output or of a derived second output voltage, a tracking regulator configuration or even a second switching regulator.

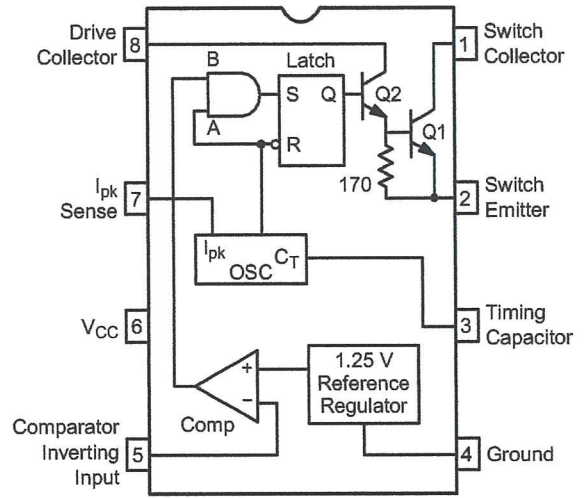
**FUNCTIONAL DESCRIPTION**

The oscillator is composed of a current source and sink which charges and discharges the external timing capacitor  $C_T$  between an upper and lower preset threshold. The typical charge and discharge currents are 35  $\mu A$  and 200  $\mu A$  respectively, yielding about a one to six ratio. Thus the ramp-up period is six times longer than that of the ramp-down as shown in Figure 3. The upper threshold is equal to the internal reference voltage of 1.25 V and the lower is approximately equal to 0.75 V. The oscillator runs continuously at a rate controlled by the selected value of  $C_T$ .

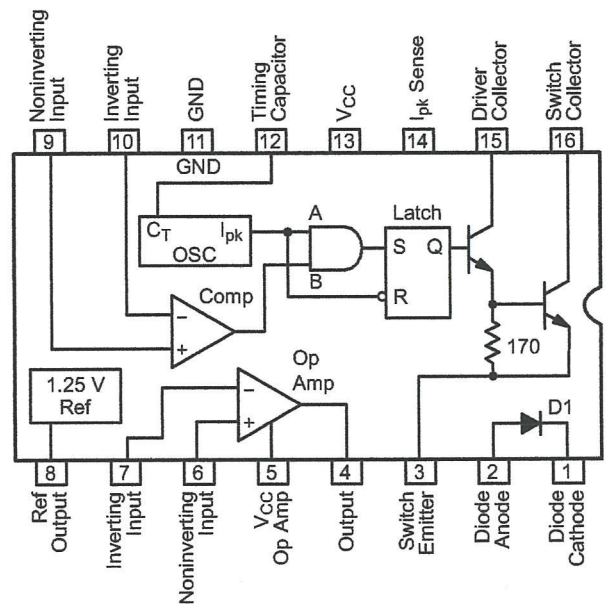
During the ramp-up portion of the cycle, a Logic "1" is present at the "A" input of the AND gate. If the output voltage of the switching regulator is below nominal, a Logic "1" will also be present at the "B" input. This condition will set the latch and cause the "Q" output to go to a Logic "1", enabling the driver and output switch to conduct. When the oscillator reaches its upper threshold,  $C_T$  will start to discharge and Logic "0" will be present at the "A" input of the AND gate. This logic level is also connected to an inverter whose output presents a Logic "1" to the reset input of the latch. This condition will cause "Q" to go low, disabling the driver and output switch. A logic truth table of these functional blocks is shown in Figure 4.

The output of the comparator can set the latch only during the ramp-up of  $C_T$  and can initiate a partial or full on-cycle of output switch conduction. Once the comparator has set the latch, it cannot reset it. The latch will remain set until  $C_T$  begins ramping down. Thus the comparator can initiate output switch conduction, but cannot terminate it and the latch is always reset when  $C_T$  begins ramping down. The comparator's output will be at a Logic "0" when the output voltage of the switching regulator is above nominal. Under

these conditions, the comparator's output can inhibit a portion of the output switch on-cycle, a complete cycle, a complete cycle plus a portion of one cycle, multiple cycles, or multiple cycles plus a portion of one cycle.

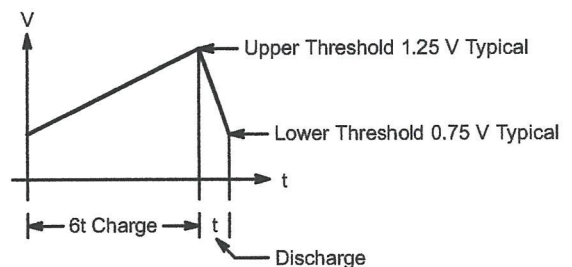


a. MC34063



b.  $\mu A78S40$

**Figure 2. Functional Block Diagrams**



**Figure 3.  $C_T$  Voltage Waveform**

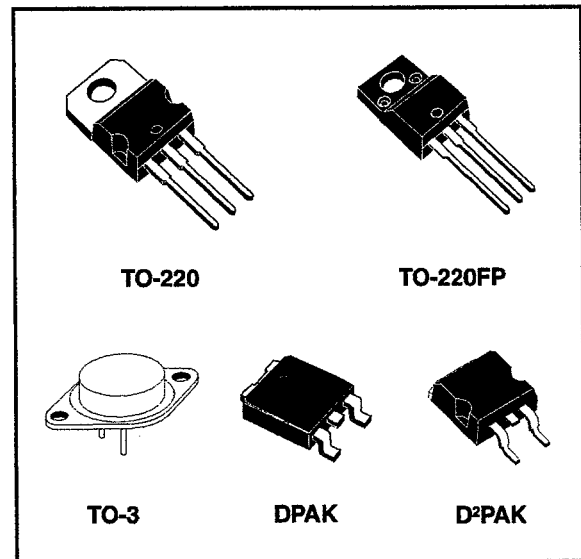
## Positive voltage regulators

### Features

- Output current to 1.5 A
- Output voltages of 5; 6; 8; 8.5; 9; 12; 15; 18; 24 V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection

### Description

The L78xx series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3, D<sup>2</sup>PAK and DPAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.



**Table 1. Device summary**

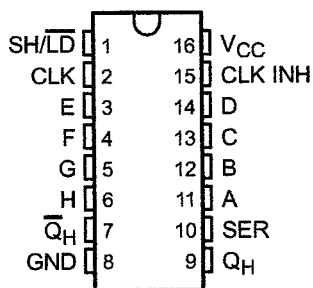
| Part numbers |        |
|--------------|--------|
| L7805        | L7809C |
| L7805C       | L7812C |
| L7806C       | L7815C |
| L7808C       | L7818C |
| L7885C       | L7824C |

# SN54HC165, SN74HC165 8-BIT PARALLEL-LOAD SHIFT REGISTERS

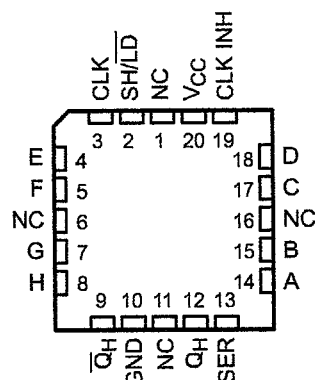
SCLS116E – DECEMBER 1982 – REVISED SEPTEMBER 2003

- Wide Operating Voltage Range of 2 V to 6 V
- Outputs Can Drive Up To 10 LSTTL Loads
- Low Power Consumption, 80- $\mu$ A Max  $I_{CC}$
- Typical  $t_{pd} = 13$  ns
- $\pm 4$ -mA Output Drive at 5 V
- Low Input Current of 1  $\mu$ A Max
- Complementary Outputs
- Direct Overriding Load (Data) Inputs
- Gated Clock Inputs
- Parallel-to-Serial Data Conversion

SN54HC165 . . . J OR W PACKAGE  
SN74HC165 . . . D, DB, N, NS, OR PW PACKAGE  
(TOP VIEW)



SN54HC165 . . . FK PACKAGE  
(TOP VIEW)



NC – No internal connection

## description/ordering information

The 'HC165 devices are 8-bit parallel-load shift registers that, when clocked, shift the data toward a serial ( $Q_H$ ) output. Parallel-in access to each stage is provided by eight individual direct data (A–H) inputs that are enabled by a low level at the shift/load (SH/LD) input. The 'HC165 devices also feature a clock-inhibit (CLK INH) function and a complementary serial ( $Q_H$ ) output.

## ORDERING INFORMATION

| TA             | PACKAGE†   |              | ORDERABLE PART NUMBER | TOP-SIDE MARKING |
|----------------|------------|--------------|-----------------------|------------------|
| –40°C to 85°C  | PDIP – N   | Tube of 25   | SN74HC165N            | SN74HC165N       |
|                |            | Tube of 40   | SN74HC165D            | HC165            |
|                | SOIC – D   | Reel of 2500 | SN74HC165DR           |                  |
|                |            | Reel of 250  | SN74HC165DT           |                  |
|                | SOP – NS   | Reel of 2000 | SN74HC165NSR          | HC165            |
|                | SSOP – DB  | Reel of 2000 | SN74HC165DBR          | HC165            |
|                | TSSOP – PW | Tube of 90   | SN74HC165PW           | HC165            |
| Reel of 2000   |            | SN74HC165PWR |                       |                  |
| Reel of 250    |            | SN74HC165PWT |                       |                  |
| –55°C to 125°C | CDIP – J   | Tube of 25   | SNJ54HC165J           | SNJ54HC165J      |
|                | CFP – W    | Tube of 150  | SNJ54HC165W           | SNJ54HC165W      |
|                | LCCC – FK  | Tube of 55   | SNJ54HC165FK          | SNJ54HC165FK     |

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated  
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

# TPIC6B595 POWER LOGIC 8-BIT SHIFT REGISTER

SLIS032 – JULY 1995

- Low  $r_{DS(on)}$  . . . 5  $\Omega$  Typical
- Avalanche Energy . . . 30 mJ
- Eight Power DMOS-Transistor Outputs of 150-mA Continuous Current
- 500-mA Typical Current-Limiting Capability
- Output Clamp Voltage . . . 50 V
- Devices Are Cascadable
- Low Power Consumption

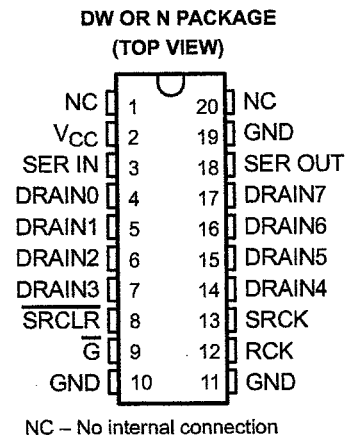
## description

The TPIC6B595 is a monolithic, high-voltage, medium-current power 8-bit shift register designed for use in systems that require relatively high load power. The device contains a built-in voltage clamp on the outputs for inductive transient protection. Power driver applications include relays, solenoids, and other medium-current or high-voltage loads.

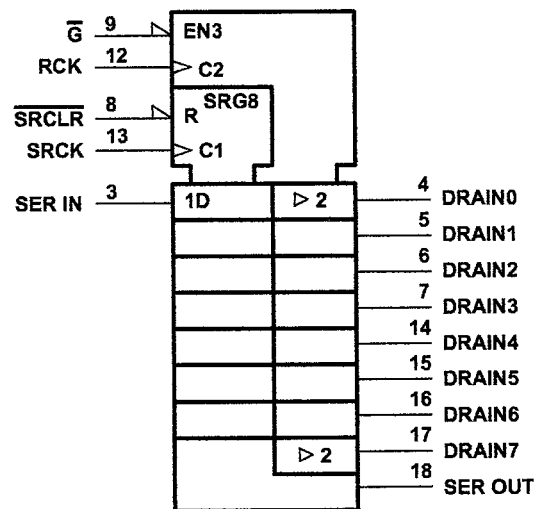
This device contains an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. Data transfers through both the shift and storage registers on the rising edge of the shift-register clock (SRCK) and the register clock (RCK), respectively. The storage register transfers data to the output buffer when shift-register clear ( $\overline{SRCLR}$ ) is high. When  $\overline{SRCLR}$  is low, the input shift register is cleared. When output enable ( $\overline{G}$ ) is held high, all data in the output buffers is held low and all drain outputs are off. When  $\overline{G}$  is held low, data from the storage register is transparent to the output buffers. When data in the output buffers is low, the DMOS-transistor outputs are off. When data is high, the DMOS-transistor outputs have sink-current capability. The serial output (SER OUT) allows for cascading of the data from the shift register to additional devices.

Outputs are low-side, open-drain DMOS transistors with output ratings of 50 V and 150-mA continuous sink-current capability. Each output provides a 500-mA typical current limit at  $T_C = 25^\circ\text{C}$ . The current limit decreases as the junction temperature increases for additional device protection.

The TPIC6B595 is characterized for operation over the operating case temperature range of  $-40^\circ\text{C}$  to  $125^\circ\text{C}$ .



## logic symbol



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1995, Texas Instruments Incorporated

---

# HA17555 Series

Precision Timer

# HITACHI

ADE-204-064 (Z)

Rev. 0

Dec. 2000

---

## Description

HA17555 Series are ICs designed for accurate time delays or oscillations. It provides both of trigger terminal and reset terminal in order to enable a wide scope of application including Mono Multi Vibrator and Astable Multi Vibrator, and the number of external components is fewer. Further, it's compatible with NE555 of singnetics.

## Features

- Mono multi vibrator can be constructed with one resistor and one capacitor.
- Astable multi vibrator can be constructed with two resistors and one capacitor.
- Delay time can be established widely from several  $\mu$  seconds to several hours.
- Pulse Duty can be controlled.
- The maximum value of both sink current and source current is 200mA.
- Direct connection of output to TTL is possible.
- Temperature/delay time ratio is 50 ppm/ $^{\circ}$ C (typ).
- Output is normally in the on and off states.

## Ordering Information

| Application    | Type No.  | Package |
|----------------|-----------|---------|
| Industrial use | HA17555PS | DP-8    |
|                | HA17555FP | FP-8D   |
| Commercial use | HA17555   | DP-8    |
|                | HA17555F  | FP-8D   |

---

---

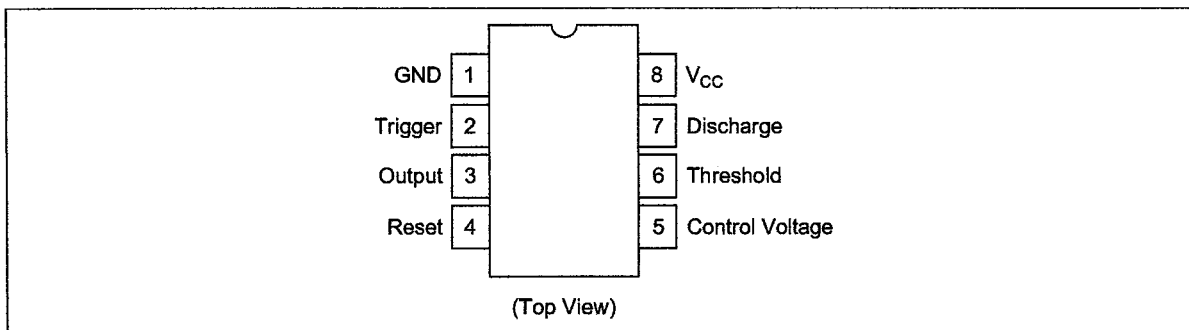
## HA17555 Series

---

### Applications

- Delay Time Generator (Mono Multi Vibrator)
- Pulse Generator (Astable Multi Vibrator)
- Pulse Width Modulator
- Pulse Location Modulator
- Miss Pulse Detector

### Pin Arrangement



### Pin Description

| Pin No. | Function            |
|---------|---------------------|
| 1       | Ground pin          |
| 2       | Trigger pin         |
| 3       | Output pin          |
| 4       | Reset pin           |
| 5       | Control voltage pin |
| 6       | Threshold pin       |
| 7       | Discharge pin       |
| 8       | V <sub>CC</sub> pin |

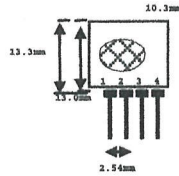
**HITACHI**



# TLP434A & RLP434A RF ASK Hybrid Modules for Radio Control ( New Version )

## TLP434A Ultra Small Transmitter

**Easy-Link  
Wireless**

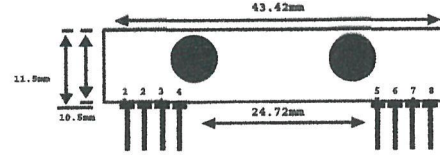


pin 1 : GND  
pin 2 : Data In  
pin 3 : Vcc  
pin 4 : Antenna ( RF output )

**Frequency 315, 418 and 433.92 Mhz**

Modulation : ASK  
Operation Voltage : 2 - 12 VDC

## RLP434A SAW Based Receiver



pin 1 : Gnd  
pin 2 : Digital Data Output  
pin 3 : Linear Output /Test  
pin 4 : Vcc  
pin 5 : Vcc  
pin 6 : Gnd  
pin 7 : Gnd  
pin 8 : Antenna

**Frequency 315, 418 and 433.92 Mhz**

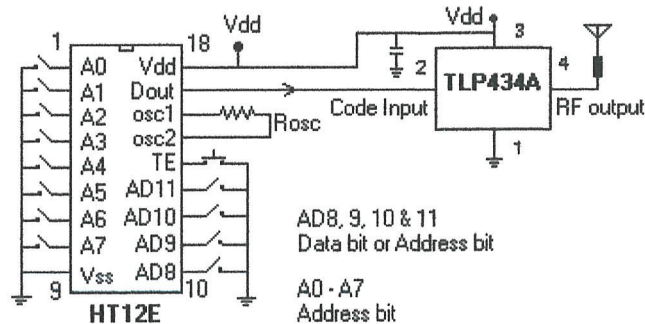
Modulation : ASK  
Supply Voltage : 3.3 - 6.0 VDC  
Output : Digital & Linear

| Symbol | Parameter                | Conditions          | Min     | Typ  | Max     | Unit |
|--------|--------------------------|---------------------|---------|------|---------|------|
| Vcc    | Operating supply voltage |                     | 2.0     | -    | 12.0    | V    |
| Icc 1  | Peak Current (2V)        |                     | -       | -    | 1.64    | mA   |
| Icc 2  | Peak Current (12V)       |                     | -       | -    | 19.4    | mA   |
| Vh     | Input High Voltage       | Idata= 100uA (High) | Vcc-0.5 | Vcc  | Vcc+0.5 | V    |
| VI     | Input Low Voltage        | Idata= 0 uA (Low)   | -       | -    | 0.3     | V    |
| FO     | Absolute Frequency       | 315Mhz module       | 314.8   | 315  | 315.2   | MHz  |
| PO     | RF Output Power- 50ohm   | Vcc = 9V-12V        | -       | 16   | -       | dBm  |
|        |                          | Vcc = 5V-6V         | -       | 14   | -       | dBm  |
| DR     | Data Rate                | External Encoding   | 512     | 4.8K | 200K    | bps  |

Notes : ( Case Temperature = 25°C +- 2°C , Test Load Impedance = 50 ohm )

### Application Circuit :

Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.



AD8, 9, 10 & 11  
Data bit or Address bit  
A0 - A7  
Address bit

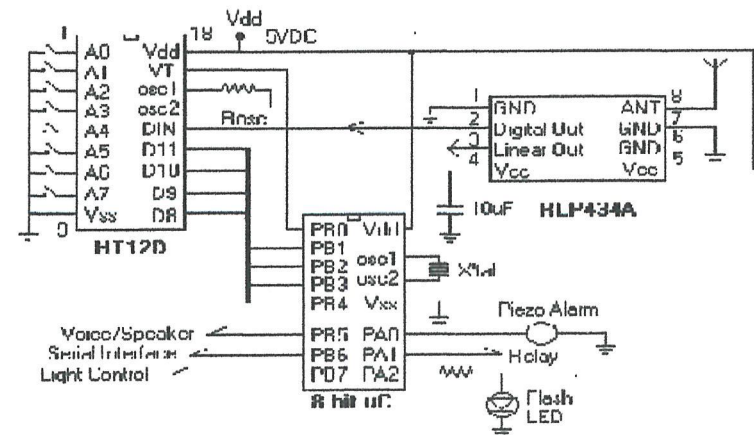
| Symbol | Parameter                | Conditions               | Min     | Typ  | Max | Unit |
|--------|--------------------------|--------------------------|---------|------|-----|------|
| Vcc    | Operating supply voltage |                          | 3.3     | 5.0V | 6.0 | V    |
| Ito    | Operating Current        |                          | -       | 4.5  | -   | mA   |
| Vdata  | Data Out                 | Idata = +200 uA ( High ) | Vcc-0.5 | -    | Vcc | V    |
|        |                          | Idata = -10 uA ( Low )   | -       | -    | 0.3 | V    |

### Electrical Characteristics

| Characteristics           | SYM  | Min                 | Typ | Max | Unit |
|---------------------------|------|---------------------|-----|-----|------|
| Operation Radio Frequency | FC   | 315, 418 and 433.92 |     |     | MHz  |
| Sensitivity               | Pref | -110                |     |     | dBm  |
| Channel Width             |      | ±500                |     |     | KHz  |
| Noise Equivalent BW       |      | 4                   |     |     | KHz  |
| Receiver Turn On Time     |      | 5                   |     |     | ms   |
| Operation Temperature     | Top  | -20                 |     |     | C    |
| Baseboard Data Rate       |      | 4.8                 |     |     | KHz  |

### Application Circuit :

Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.

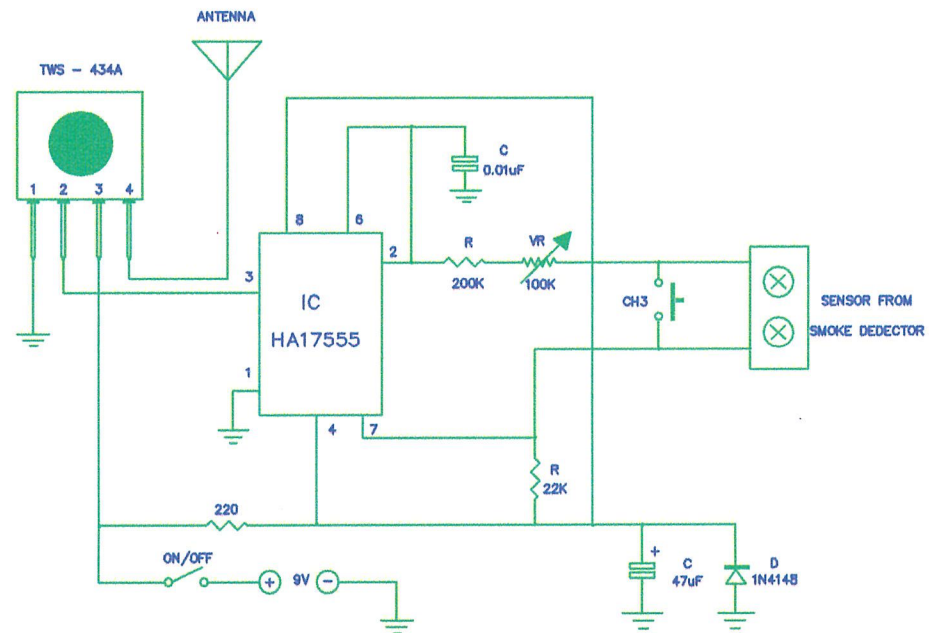
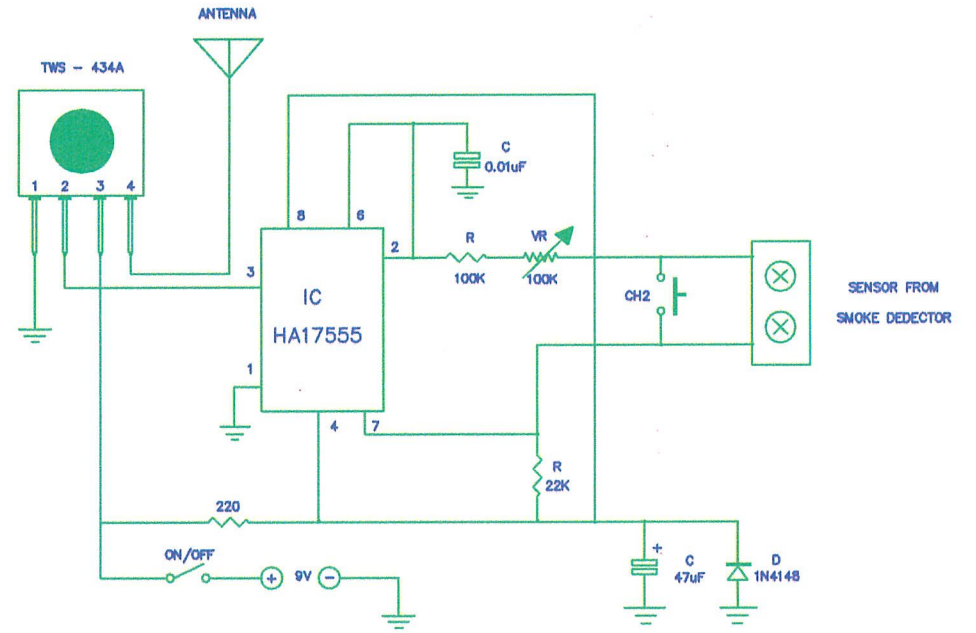
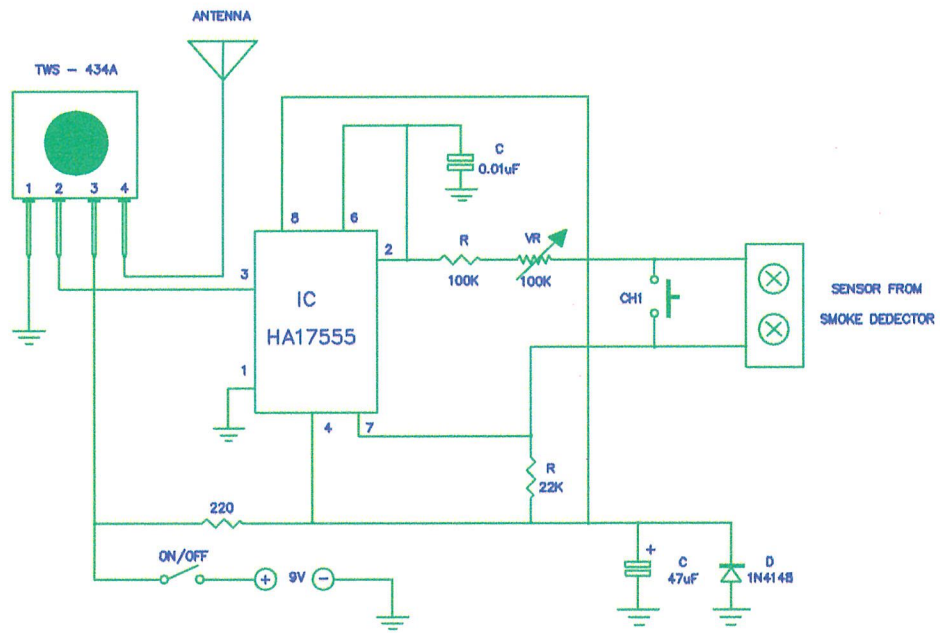


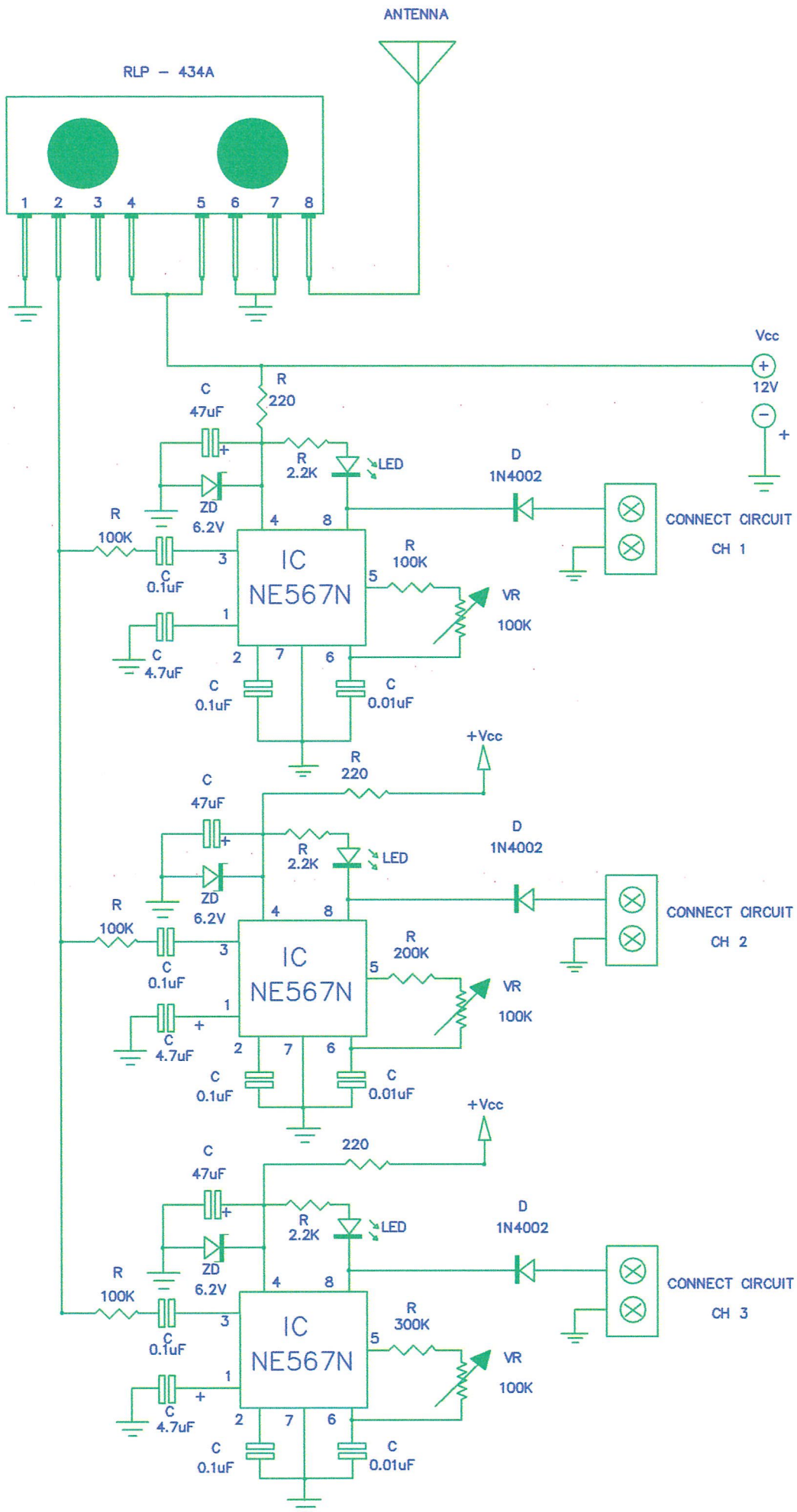
## Laipac Technology, Inc.

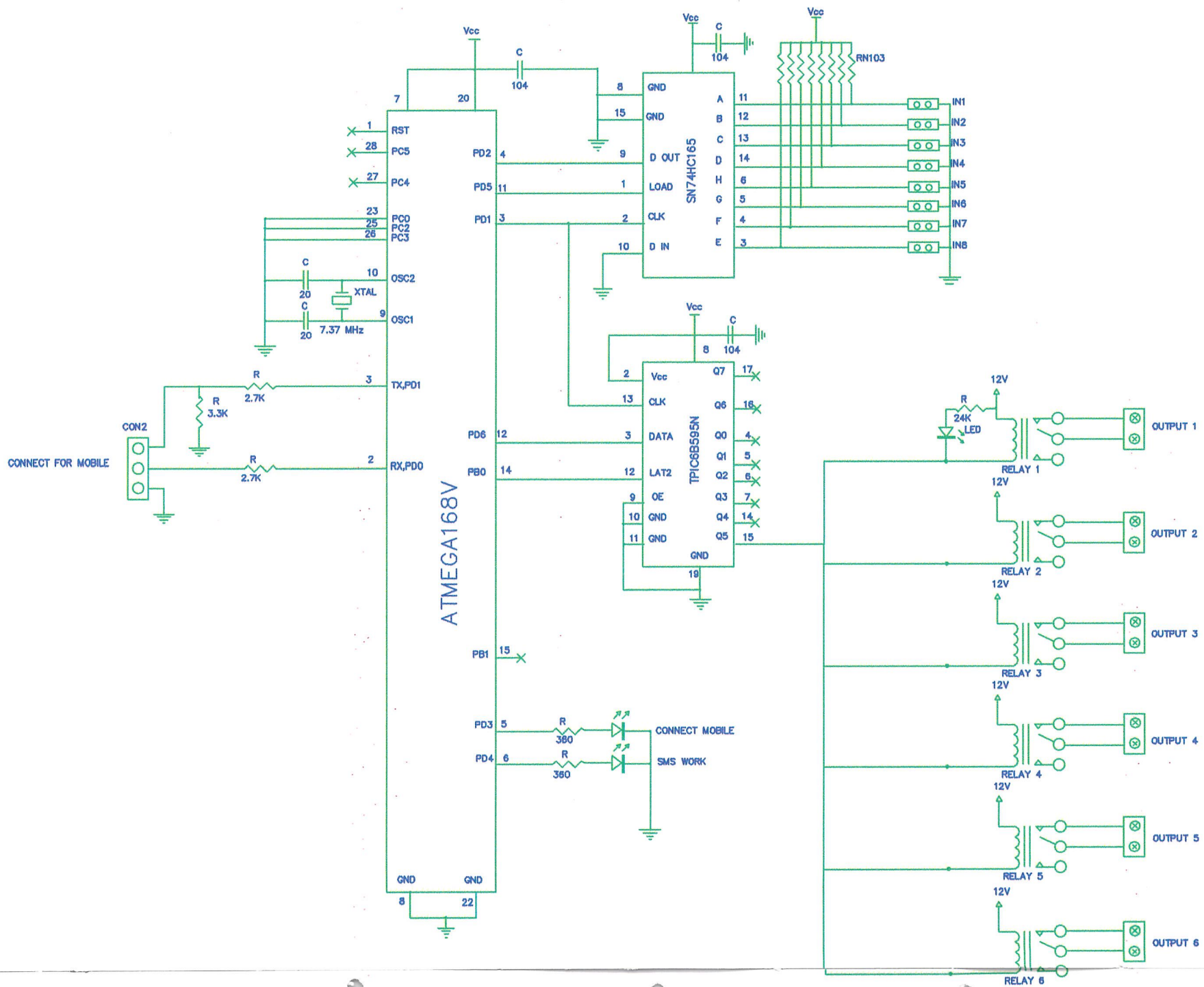
105 West Beaver Creek Rd. Unit 207 Richmond Hill Ontario L4B 1C6 Canada  
Tel: (905)762-1228 Fax: (905)763-1737 e-mail: info@laipac.com

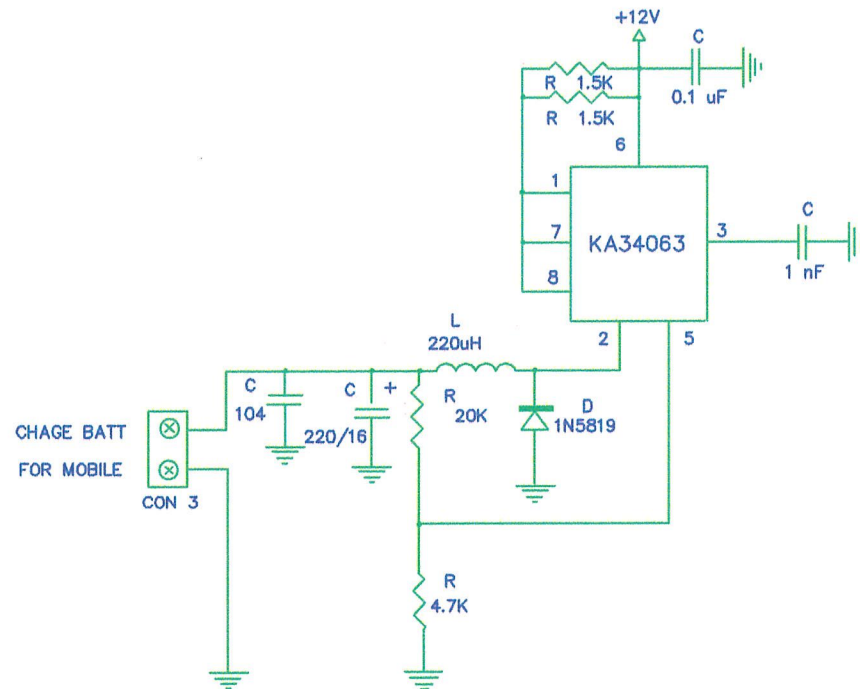
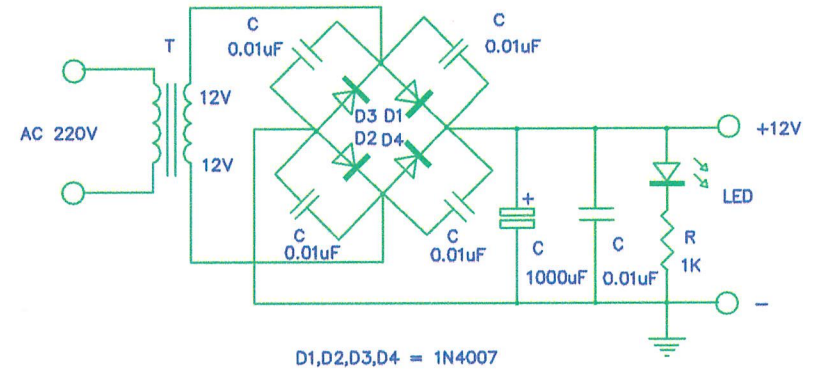
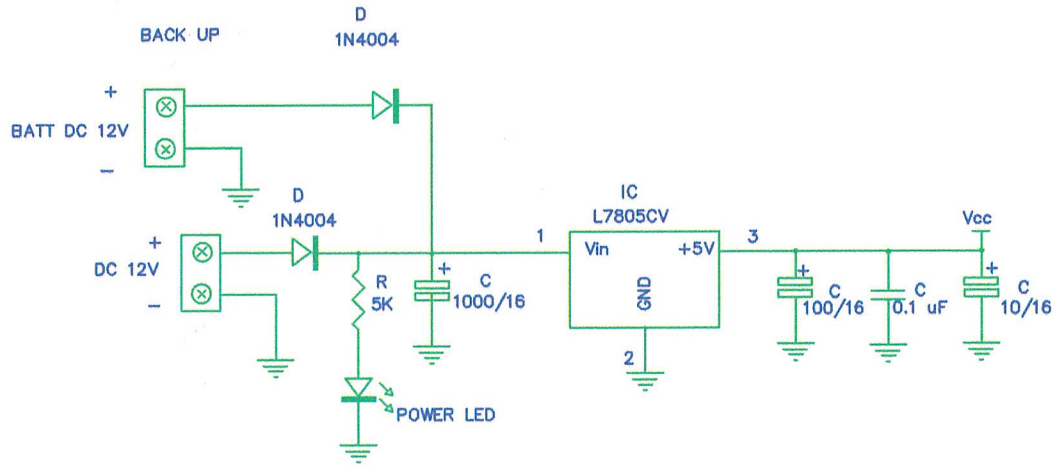


## ภาคผนวก 2









SOURCE CODE โปรแกรมควบคุมควบคุมไมโครคอนโทรลเลอร์

```

/*-----
File : โปรแกรมควบคุมไมโครคอนโทรลเลอร์
Compiler : AVR Studio/WinAVR
Target : ATmega16
-----
*/

//-----:INCLUDE

#include <avr/io.h>           / AVR device-specific IO definitions
#include <compat/deprecated.h> // Use sbi(),cbi() function

#define F_CPU 8000000UL      // XTAL 8 MHz
#include <util/delay.h>      // header file implement simple delay loops
#include <avr/io.h>         // AVR device – specific IO definitions

#define DDR_SR   DDRA        // direction port shift register
#define PORT_OUT_SR PORTA    // output port
#define PORT_IN_SR PINA      // input port

#define SH_PIN   PD5         // asynchronous parallel load input
#define CLK_PIN  PD1         // clock input (LOW-to-HIGH edge-triggered)
#define QH_PIN   PD2         // serial output from the last stage

#define SET_INP(x)  cbi(DDR_SR,x)
#define SET_OUTP(x) sbi(DDR_SR,x)

#define SET_P(x)    sbi(PORT_OUT_SR,x)
#define CLS_P(x)    cbi(PORT_OUT_SR,x)
#define READ_P(x)   (PORT_IN_SR&(1<<x))

//-----:FUNCTION

// delay milliseconds
void delay_ms(uint16_t i)
{
  for (;i > 0; i--)
    _delay_ms(1);
}

// rotate left
uint8_t rotate_left(uint8_t *x, uint8_t i)
while(1)

{

```

```
//INPUT = 0xFF;

INPUT = shift_in_data();// read data

if(INPUT != 0x7F)

{

    shift_out_dat(~INPUT);

    printf("%x\r\n",INPUT);

    while(INPUT != 0x7F)

    {

        INPUT = shift_in_data();// read data

        delay_ms(10);

    }

}

if(INPUT != 0xBF)

{

    shift_out_dat(~INPUT);

    printf("%x\r\n",INPUT);

    while(INPUT != 0xBF)

    {

        INPUT = shift_in_data();// read data

        delay_ms(10);

    }

}

if(INPUT != 0xDF)

{

    shift_out_dat(~INPUT);

    printf("%x\r\n",INPUT);
```

```
while(INPUT != 0xDF)
{
    INPUT = shift_in_data();// read data
    delay_ms(10);
}
}
if(INPUT != 0xEF)
{
    shift_out_dat(~INPUT);
    printf("%x\r\n",INPUT);
    while(INPUT != 0xEF)
    {
        INPUT = shift_in_data();// read data
        delay_ms(10);
    }
}
if(INPUT != 0xF7)
{
    shift_out_dat(~INPUT);
    printf("%x\r\n",INPUT);
    while(INPUT != 0xF7)
    {
        INPUT = shift_in_data();// read data
        delay_ms(10);
    }
}
```

```
if(INPUT != 0xFB)
{
    shift_out_dat(~INPUT);
    printf("%x\r\n",INPUT);
    while(INPUT != 0xFB)
    {
        INPUT = shift_in_data();// read data
        delay_ms(10);
    }
}
```

```
if(INPUT != 0xFD)
{
    shift_out_dat(~INPUT);
    printf("%x\r\n",INPUT);
    while(INPUT != 0xFD)
    {
        INPUT = shift_in_data();// read data
        delay_ms(10);
    }
}
```

```
if(INPUT != 0xFE)
{
    shift_out_dat(~INPUT);
    printf("%x\r\n",INPUT);
    while(INPUT != 0xFE)
    {
```

```
        INPUT = shift_in_data();// read data

        delay_ms(10);

    }

}

/*

// INPUT = shift_in_data();// read data

if(INPUT == 0xFF)

{

    printf("%x",INPUT);

}

else

{

}

delay_ms(1000);

*/

}

return(0);

}

// — Interrupt INTO (Port PD5) — //

ISR (INT0_vect)

{

    if(bit0)

        SET_P(QH_PIN);    // set input QH pin

    else

        CLS_P(QH_PIN);

    bit0 != bit0;

}
```

```

}

// Serial Data input
unsigned char shift_in_data()
{
    unsigned char i;
    unsigned char dat=0x00;

    CLS_P(SH_PIN);    // parallel load
    SET_P(SH_PIN);    // serial shift

    for(i=0;i<8;i++) {
        dat |= READ_P(QH_PIN); // data in

        rotate_left(&dat,1); // rotate left

        CLS_P(CLK_PIN);    // clock pulse
        SET_P(CLK_PIN);
    }

    rotate_left(&dat,2); // rotate left

    return(dat);
}

//-----FUNCTION MAIN
int main(void)
{
    unsigned char INPUT = 0x00;

    // ---- SET PIN CONTROL 74HC165 ---- //
    SET_OUTP(SH_PIN);    // set output PL pin
    SET_OUTP(CLK_PIN);    // set output CP pin
    SET_INP(QH_PIN);    // set input QH pin
    SET_OUTP(SH_CP_PIN );    // set output SH_CP_PIN pin

    Init_Serial(96);
    // DDRB = 0xFF;        // SET PORT B OUTPUT
    // DDRB = (1<<DDAB);

    // PORTB = 0x00;    // clear PORT A0. //
    // ---- SET PIN CONTROL 74HC165 ---- //
    // SET_OUTP(SH_PIN);    // set output PL pin
    // SET_OUTP(CLK_PIN);    // set output CP pin
    // SET_INP(QH_PIN);    // set input QH pin

    // DDRD = ((0 << DDD2)|(1 << DDD7)|(1 << DDD5));

```

```

// cbi(PORTD,0);
// printf("Test serial.");
/*
// ----- SET INTERRUPT INTO ----- //
MCUCR = (1<<ISCO1) | (1<<ISCO0);

GICR = (1<<INT0);
SREG = (0x08);
*/

while(1)
{
//INPUT = 0xFF;
INPUT = shift_in_data();// read data
if(INPUT != 0xFF)
{
shift_out_dat(~INPUT);
printf("%x\r\n",INPUT);
while(INPUT != 0xFF)
{
Int main(void)
{
char ch1;
ch1 = 8; // set 8 bit to register
while (1);

return 0;
}
}
}
}

//-----:

#define DS_PIN PD6 // serial data input
#define SH_CP_PIN PD1 // shift register clock input
#define ST_CP_PIN PB0 // storage register clock input

#define SET_INP(x) cbi(DDR_SR,x)
#define SET_OUTP(x) sbi(DDR_SR,x)

#define SET_P(x) sbi(PORT_SR,x)
#define CLS_P(x) cbi(PORT_SR,x)

// delay milliseconds
void delay_ms(uint16_t i)
{
for (;i > 0; i--)
_delay_ms(1);
}

```

```

// rotate left
uint8_t rotate_left(uint8_t *x, uint8_t i)
{
  for(;i>0;i--)
    *x = (*x>>1) | (*x<<7);
    X = 1 // Set 11...
  return (*x);
}

// Data transfers
void shift_out_data(uint8_t dat)
{
  unsigned char i;

  for(i=0;i<=8;i++) {
    if(dat & 0x80) // serial data in
      SET_P(DS_PIN);
    else
      CLS_P(DS_PIN);

    SET_P(SH_CP_PIN); // clock pulse
    CLS_P(SH_CP_PIN);

    SET_P(ST_CP_PIN); // latch pulse
    CLS_P(ST_CP_PIN);

    dat = rotate_left(&dat,1); // rotate left
  }
}
//-----:MAIN

int main(void)
{
  // set output port
  SET_OUTP(DS_PIN);
  SET_OUTP(SH_CP_PIN);
  SET_OUTP(ST_CP_PIN);

  while (1) { // loop forever
    shift_out_data(0x0F); // output
    delay_ms(100000); // delay 100s
    shift_out_data(0xF0);
    delay_ms(100000);
    shift_out_data(0xAA);
    delay_ms(100000);
    shift_out_data(0x55);
    delay_ms(100000);
  }

  return 0;
}

```

```

#include "includes.h" //ကတ်ကား PDU

/**
unsigned char ascii2hex( unsigned char a )
{
    if ( a >= '0' && a <= '9' )
        return ( a - '0' );
    else if ( a >= 'a' && a <= 'f' )
        return ( a - 'a' + 10 );
    else if ( a >= 'A' && a <= 'F' )
        return ( a - 'A' + 10 );
    else
        return 0;
}

/**
unsigned char hex2ascii( unsigned char h )
{
    if ( h <= 9 )
        return ( h + '0' );
    else if ( h >= 10 && h <= 15 )
        return ( h + 'A' - 10 );
    return '\0';
}

//*****
*****

typedef struct _PDU_RX_HEADER
{
    unsigned char smsc_info_len[2];
    unsigned char smsc_type[2];

```

```
    unsigned char smsc_number[12];  
    unsigned char first_octet[2];  
    unsigned char sender_num_len[2];  
    unsigned char sender_num_type[2];  
    unsigned char sender_num[12];  
    unsigned char protocol_identifier[2];  
    unsigned char data_coding[2];  
    unsigned char time_stamp[14];  
    unsigned char data_len[2];  
}PDU_RX_HEADER;
```

```
typedef struct _PDU_TX_HEADER  
{  
    unsigned char smsc_info_len[2];  
    unsigned char first_octet[2];  
    unsigned char message_reference[2];  
    unsigned char receiver_num_len[2];  
    unsigned char receiver_num_type[2];  
    unsigned char receiver_num[12];  
    unsigned char protocol_identifier[2];  
    unsigned char data_coding[2];  
    unsigned char validity_period[2];  
    unsigned char data_len[2];  
}PDU_TX_HEADER;
```

```
unsigned char ascii2hex( unsigned char a );  
unsigned char hex2ascii( unsigned char h );
```

```

unsigned char pdu_encode( unsigned char *plain_txt, unsigned char *pdu_txt );
void pdu_decode(unsigned char *pdu_txt, unsigned char *plain_txt);
unsigned char pdu_encode( unsigned char *plain_txt, unsigned char *pdu_txt )
{
    unsigned char i=8, plain_len=0, pdu_len=0;

    while( (plain_txt[plain_len]) )
    {
        plain_len++;
    }
    while ( plain_len-- )
    {
        if ( plain_txt[1] == '\0' )
        {
            if ( i != 1 )
            {
                #>Zone x = ascii
                X++
                pdu_txt[0] = hex2ascii( (plain_txt[0] & 0xf0)>>4 );
                pdu_txt[1] = hex2ascii( (plain_txt[0] & 0x0f) );
                pdu_txt[2] = '\0';
                pdu_len += 2;
            }
        }
        else
        {
            if ( i != 1 )
            {

```

```

        pdu_txt[0] = hex2ascii( ((plain_txt[0] | (plain_txt[1]) << (i-1))
& 0xf0)>>4 );

        pdu_txt[1] = hex2ascii( ((plain_txt[0] | (plain_txt[1]) << (i-1))
& 0x0f) );

        plain_txt[1] >>= 9-i;

        pdu_txt += 2;

        pdu_len++;

    }

}

plain_txt++;

if ( --i == 0 )

{

    i = 8;

}

}

//

return pdu_len;

}

//*****
*****

void pdu_decode(unsigned char *pdu_txt, unsigned char *plain_txt)

{

    unsigned char i, len=0;

    // convert ascii to hex

    i = 0;

    while ( (pdu_txt[i]) )

    {

        pdu_txt[len] = ascii2hex( pdu_txt[i] )<<4 | ascii2hex( pdu_txt[i+1] );

```

```

        i += 2;

        len++;

    }

    pdu_txt[len] = '\0';

    i = 8;

    while ( len-- )

    {

        if ( i == 8 )

        {

            plain_txt[0] = pdu_txt[0] & 0x7f;

        }

        else if ( i == 2 )

        {

            plain_txt[0] = ( pdu_txt[0] & ( 0xff >> (9-i) ) ) << (8-i) | ( pdu_txt[-1]
>> i);

            plain_txt[1] = pdu_txt[0] >> 1;

            plain_txt++;

        }

        else

        {

            plain_txt[0] = ( pdu_txt[0] & ( 0xff >> (9-i) ) ) << (8-i) | ( pdu_txt[-1]
>> i);

        }

        if ( --i == 1 )

        {

            i = 8;

        }

```

```

        plain_txt++;

        pdu_txt++;

    }

}

#include "includes.h" // SMS SEND

/**
*****
*****

// get variable value from sms,

// sms command protocol is [variable_1=value_1&variable_n&value_n&...]

// at_get_val function find a variable and value then save value to *value pointer

// if not found variable key function will be return with '0'

/**
*****
*****

unsigned char at_send_sms ( unsigned char *receiver_num, unsigned char *plain_txt )
{

    //unsigned char i;

    unsigned char tx_buffer[256];

    unsigned char rx_buffer[32];

    unsigned char *tx_buffer_ptr;

    // Length of SMSC information. Here the length is 0,

    // which means that the SMSC stored in the phone should be used.

    tx_buffer[0] = '0';

    tx_buffer[1] = '0';

    // First octet of the SMS-SUBMIT message.

    tx_buffer[2] = '1';

```

```
tx_buffer[3] = '1';
```

```
// TP-Message-Reference.
```

```
// The "00" value here lets the phone set the message reference number itself.
```

```
tx_buffer[4] = '0';
```

```
tx_buffer[5] = '0';
```

```
//The "0B" is Address-Length = 11
```

```
tx_buffer[6] = '0';
```

```
tx_buffer[7] = 'B';
```

```
//The "91" is number in thai
```

```
tx_buffer[8] = '9';
```

```
tx_buffer[9] = '1';
```

```
//Send to number +66839202123
```

```
tx_buffer[10] = '6';
```

```
tx_buffer[11] = '6';
```

```
tx_buffer[12] = '3';
```

```
tx_buffer[13] = '8';
```

```
tx_buffer[14] = '2';
```

```
tx_buffer[15] = '9';
```

```
tx_buffer[16] = '2';
```

```
tx_buffer[17] = '0';
```

```
tx_buffer[18] = '2';
```

```
tx_buffer[19] = '1';
```

```
tx_buffer[20] = 'F';
```

```
tx_buffer[21] = '3';
```

```
// TP-PID. Protocol identifier.

tx_buffer[22] = '0';

tx_buffer[23] = '0';

// TP-DCS Data coding scheme.

tx_buffer[24] = '0';

tx_buffer[25] = '0';

// TP-Validity-Period. "AA" means 4 days.

tx_buffer[26] = 'A';

tx_buffer[27] = 'A';

// TP-User-Data-Length

tx_buffer[28] = '0';

tx_buffer[29] = '9';

// encode plain text to pdu text, pdu length stored in i variable

i = pdu_encode ( plain_txt, (unsigned char*)&tx_buffer[sizeof(PDU_TX_HEADER)] );

// TP-User-Data-Length. Length of message.

// The TP-DCS field indicated 7-bit data,

// so the length here is the number of septets (10).

// If the TP-DCS field were set to 8-bit data or Unicode,

// the length would be the number of octets.

tx_buffer[30] = hex2ascii( i >> 4 );

tx_buffer[31] = hex2ascii( i & 0x0f);
```

```
// octet count and store in i variable

i = 0;

tx_buffer_ptr = (unsigned char*)&tx_buffer[2];

while ( (*tx_buffer_ptr) )

{

    i++;

    tx_buffer_ptr += 2;

}

// clear status flag

at_flag.byte = 0;

// sending sms by send AT+CMGS to phone

uart_puts_p ( PSTR ("AT+CMGS=") );

// number of octets to send

uart_putc ( hex2ascii( i/10 ) );

uart_putc ( hex2ascii( i%10 ) );

// put pdu code to phone

uart_puts_p ( PSTR ( "\r\n" ) );

delay_ms ( 200 );

uart_puts ( (unsigned char*)&tx_buffer );

// send sms by put Ctrl+Z(0x1a) to phone

delay_ms ( 200 );

uart_putc ( 0x1a );

// read status

delay_ms ( 2000 );
```

```
at_read_sms( (unsigned char*)&rx_buffer );

delay_ms(1);

if ( at_flag.bits.rx_error == 1 || at_flag.bits.rx_timeout == 1 )
    return 0;
else
    return 1;
}

union at_flag
{
    unsigned char byte;
    struct
    {
        unsigned char rx_ok:1;
        unsigned char rx_error:1;
        unsigned char rx_timeout:1;
        unsigned char unuse:5;
    }bits;
}at_flag;

unsigned char at_send_sms( unsigned char *tx_buffer );
unsigned char at_get_val ( unsigned char *src, unsigned char *key, unsigned char *value );
unsigned char at_sms_process ( unsigned char *sender_num );
unsigned char at_init ( void );
unsigned char at_send_sms ( unsigned char *receiver_num, unsigned char *plain_txt );
```

```

#define UART_BAUD 19200

//#define UART_BAUD 19200

คำสั่งส่งค่าออกทาง port PD1 เข้าโทรศัพท์มือถือ

void uart_init(void);

void uart_putc(unsigned char c);

char uart_getc(int timeOut);

void uart_puts(unsigned char *ptr);

void uart_puts_p(PGM_P ptr);

#include "includes.h"

/*

static int uart_getchar(FILE *stream) // read data from usart
{

    uint8_t c;

    char *cp, *cp2;

    static char b[RX_BUFSIZE];

    static char *rxp;

    if (rxp == 0)

        for (cp = b;;) {

            loop_until_bit_is_set(UCSRA, RXC);

            if (UCSRA & _BV(FE)) return _FDEV_EOF;

            if (UCSRA & _BV(DOR)) return _FDEV_ERR;

```

```
c = UDR;

// behaviour similar to Unix stty ICRNL

if (c == '\r') c = '\n';

if (c == '\n') {

    *cp = c;

    uart_putchar(c, stream);

    rxp = b;

    break;

} else if (c == '\t') c = ' ';

if ((c >= (uint8_t)' ' && c <= (uint8_t)'x7e') ||

    c >= (uint8_t)'xa0') {

    if (cp == b + RX_BUFSIZE - 1)

        uart_putchar('\a', stream);

    else {

        *cp++ = c;

        uart_putchar(c, stream);

    }

    continue;

}

{

    tx_buffer[ tx_index ] = UDR;

    if ( ++tx_index == 512 )

        tx_index = 0;
```

```
if ( tx_buffer[ tx_index-1] = '\n' &&
    tx_buffer[ tx_index-2] = '\r' )
{
    if ( rx_buffer[ (tx_index-3)] = 'K' &&
        tx_buffer[ (tx_index-4)] = 'O' )
    {
        if ( tx_buffer[ (tx_index-5)] = '\n' &&
            tx_buffer[ (tx_index-6)] = '\r' )
        {
            at_flag.bits.tx_complete = 1;
            at_flag.bits.tx_ok = 1;
        }
    }
}
else if ( rx_buffer[ rx_index-3] = 'R' &&
         rx_buffer[ tx_index-4] = 'O' &&
         rx_buffer[ tx_index-5] = 'R' &&
         rx_buffer[ tx_index-6] = 'R' &&
         rx_buffer[ tx_index-7] = 'E' )
{
    if ( tx_buffer[ tx_index-8] = '\n' &&
        tx_buffer[ tx_index-9] = '\r' )
    {
        at_flag.bits.tx_complete = 1;
        at_flag.bits.tx_error = 1;
    }
}
```

```

    }

static void USART_Init(unsigned int baud)
{
    // Set baud rate

    UBRRH = (unsigned char) (baud>>8);
    UBRL = (unsigned char) baud;

    // Enable receiver and transmitter
    UCSRB = (1<<RXEN) | (1<<TXEN);

    // Set frame format: 8data, NoneParity, 1stop bit
    UCSRC = (1<<URSEL) | (3<<UCSZ0);

    // Set address uart_str to stdout
    stdout = &uart_str;
}

/***** Main Functions
*/

int main(void)
{
    int i=99;

    USART_Init(25); // baudrate to 19200 bps using a 8MHz crystal
                   // UBRR = (( 8MHZ/(16x19200))-1 = 25
    fprintf(&uart_str, "\nUART Module : Serial communication device");
    fprintf(&uart_str, "\nCountdown....");

    while(1) { // Loop forever

```

```

    fprintf(&uart_str, "\nNumber %d", i);

    if (i-- <= 0) break;

    delay_1s(); // Delay 1 sec
}

UCSRB = (1<<RXEN)|(1<<TXEN); // Enable receiver and transmitter

// mode intrrup for check UDP

UCSRB |= (1<<RXCIE)|(1<<TXCIE); // Enable Interrupt

// Set I-bit global interrupt enable

sei();

return 0;
}

void uart_putc(unsigned char c)
{
    while ( !(UCSRA & _BV( UDRE )) );

    UDR = c;

    _delay_ms(1);
}

char uart_getc(int timeOut)
{
    while(timeOut)
    {
        if(UCSRA & _BV(RXC))
            return UDR;

        timeOut--;
    }
}

```

```
        return 0;
    }

void uart_puts(unsigned char *ptr)
{
    while(*ptr)
    {
        uart_putc(*ptr);
        ptr++;
    }
}

void uart_puts_p( PGM_P ptr )
{
    unsigned char c;

    while( (c = pgm_read_byte ( ptr++ )) )
    {
        uart_putc(c);
    }
}
```