

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบเตือนภัยควบคุมด้วยไมโครคอนโทรลเลอร์

Warning System By Microcontroller



รฟ.

๗๕๑๗

๒๕๔๙

เลขหมู่.....

เลขทะเบียน 72039

วัน,เดือน,ปี - 7 ส.ย. 2550

b. 117 b2251
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเตือนภัยควบคุมด้วยไมโครคอนโทรลเลอร์

Warning System By Microcontroller

ผู้จัดทำ

1. นายสุรพงศ์ ตริทิพย์สกุล 46010875
2. นายศักยวุฒิ ศิวาวรรณ์ 46010773
3. นางสาวสุรียพร พุ่มไสว 46010886



..... อาจารย์ที่ปรึกษา
(รศ.ดร.เกียรติศักดิ์ คมวัชระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเตือนภัยควบคุมด้วยไมโครคอนโทรลเลอร์

โดย นายสุรพงษ์ ตริทิพย์สกุล 46010875
นายศักยวุฒิ ศีลววรรณ 46010773
นางสาวสุรีย์พร พุ่มไสว 46010886

อาจารย์ที่ปรึกษา รศ.ดร.เกียรติศักดิ์ กมวัชระ

บทคัดย่อ

ในโลกปัจจุบัน โทรศัพท์มือถือได้เข้ามามีบทบาทในชีวิตปัจจุบันของมนุษย์มากขึ้น เพราะการสื่อสารเป็นสิ่งสำคัญในการดำรงชีวิตของมนุษย์ซึ่งที่เราสามารถเห็นได้จากการพกพาโทรศัพท์มือถืออย่างแพร่หลายในปัจจุบัน โดยโครงการนี้ได้นำเสนอระบบเตือนภัยผ่านโทรศัพท์มือถือด้วยระบบข้อความสั้น (SMS) ซึ่งจะทำการดำเนินชีวิตมีความปลอดภัย และ สะดวกสบายยิ่งขึ้น โดยเฉพาะโทรศัพท์มือถือนั้นยังสามารถที่จะนำไปประยุกต์ใช้ในงานด้านต่างๆ ไม่ว่าจะเป็นการควบคุมระบบภายในด้วยโทรศัพท์มือถือ เช่น ควบคุมการเปิด-ปิดอุปกรณ์ต่างๆ นอกจากนี้ยังสามารถที่จะนำไปประยุกต์ใช้ในการเตือนภัยในรูปแบบต่างๆ ได้ เช่น เตือนภัยรถยนต์ เตือนภัยธรรมชาติ เป็นต้น นอกจากนี้แล้วยังสามารถที่จะใช้ระบบนี้ในการควบคุมการให้อาหารสัตว์ผ่านระบบนี้ได้อีกด้วย อย่างไรก็ตาม ระบบนี้จะเกิดขึ้นกับ ชนิดของเหตุการณ์ ว่าเป็นแบบใด และต้องใช้เซนเซอร์อย่างไร จึงจะเหมาะสมที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Warning System By Microcontroller

By Mr. Surapong Tritipskul 46010875
 Mr. Sakkayavuth Silawan 46010773
 Mrs. Sureeporn pumeswai 46010886

Advisor

Assoc. Prof. Dr. Kiattisak Kumwachara

ABSTRACT

In present , mobile phone is very important for everyone. Because of communication is very important for everyone in social therefore we can see everyone who take the mobile phone around the world. In this project will present warning system through mobile phone in short message(SMS) which develop your safety and comfort especially for every part about mobile phone where control every part such that control switch and the other equipment. Aside of warning the other type such that warning vehicle system, natural system and you can apply for take care of any pet.It can be as same as care taker and security.However it's up to the type of accident that “ What type of accident ” and type of sensor that “ What type of sensor ” for the most comfortable for everyone best of the best.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับการสนับสนุนจากบุคคลหลายท่าน โดยทางผู้จัดทำขอขอบพระคุณ รศ.ดร.เกียรติศักดิ์ กมวัชระ คุณพบโชค ทิพย์วิชัย คุณดำรง กองเพชร คุณปรีชา สรีทิพย์อาจ ที่ให้ความกรุณาแนะนำที่เป็นประโยชน์ ตลอดจนให้ความสนใจใส่ดูแลสอบถามความก้าวหน้าอย่างสม่ำเสมอ พร้อมทั้งคณาจารย์ทุกท่าน ที่ได้ประสิทธิประสาทความรู้ในด้านต่างๆ และให้คำปรึกษาแก่ผู้จัดทำ ขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบคุณภาควิชา วิศวกรรมระบบควบคุม ที่มีเครื่องมือและอุปกรณ์ในการทำงานให้ใช้อย่างครบครัน ขอขอบคุณ พี่ๆ รุรการ ที่ให้ความสะดวกในเรื่องต่างๆ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดาของผู้จัดทำ ที่ได้ให้การอุปการะมาอย่างดีเยี่ยม และให้กำลังใจแก่ผู้จัดทำตลอดเวลา

คณะผู้จัดทำ	1. นายสุรพงศ์ ตริทิพย์สกุล	46010875
	2. นายศักขวุฒิ สีลาวรรณ	46010773
	3. นางสาวสุรีย์พร พุ่มไสว	46010886

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 เนื้อหาโดยสังเขปของปริญญานิพนธ์	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 รายละเอียดและขอบเขตของโครงการ	3
2.2 อินพุตที่ใช้ในการตรวจจับสิ่งแปลกปลอม	3
2.3 หลักการทำงานโดยรวมของวงจร	4
2.4 ไมโครคอนโทรลเลอร์	5
2.5 การจัดขาของไมโครคอนโทรลเลอร์MCSS1	6
2.6 การรับส่งข้อมูลแบบอนุกรม	8
2.6.1 การรับส่งข้อมูลแบบอนุกรม	8
2.6.2 มาตรฐานอาร์เอส-232c	9
2.6.3 อุปกรณ์ คีทีอี และ ซีซีอี	9
2.6.4 การกำหนดขาของ อาร์เอส 232 สำหรับ DB-9	10
2.7 การรับส่งข้อมูลสั้น sms	12
2.7.1 โหมดการรับส่งข้อมูล	15
2.7.2 การรับข้อความสั้น sms ใน PDU โหมด	16
2.8 การส่งข้อความ sms ใน PDU โหมด	17
2.8.1 รหัสตัวอักษรชนิด 7 บิต	18
2.8.2 การแปลงตัวอักษรชนิด 7 บิต เป็น ตัวอักษรชนิด 8 บิต	18
2.9 ไอซีถอดรหัสสัญญาณความถี่โทรศัพท์	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 ขั้นตอนการออกแบบ	25
3.1 หลักการออกแบบ	25
3.2 หลักการอินเตอร์เฟซโทรศัพท์มือถือ	26
3.3 วงจรที่ใช้ในโครงงาน	27
บทที่ 4 การทดลองและผลการทดลอง	30
บทที่ 5 บทวิจารณ์และสรุป	33
5.1 สรุปผลการทดลอง	33
5.2 ปัญหาในการดำเนินการ	33
เอกสารอ้างอิง	34
ภาคผนวก	35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

รูปที่	หน้า
2.1 เครื่องตรวจจับการเคลื่อนไหว	3
2.2 อุปกรณ์ตรวจจับการเปิดปิดประตูหน้าต่าง	4
2.3 เครื่องตรวจจับควันไฟ	4
2.4 ตำแหน่งขาของไมโครคอนโทรลเลอร์	6
2.5 การใช้มาตรฐานRS 232 เชื่อมต่อกับอุปกรณ์	9
2.6 คำอธิบายการทำงานของแต่ละขาที่ใช้งานของDB-9	10
2.7 คำอธิบายการทำงานของแต่ละขาที่ใช้งานของDB-25	11
2.8 แสดงโครงสร้างของไอซี MT 8870	21
2.9 แสดงตำแหน่งขาของไอซี MT 8870	21
3.1 วงจรAT89C51	27
3.2 วงจรพาวเวอร์ซัพพลาย	27
3.3 รูปแบบการทำงานของวงจรถาม	28
3.4 รูปแบบการทำงานของโปรแกรม	29
4.1 การกำหนดค่าสำหรับโทรศัพท์มือถือ	30
4.2 คำสั่งที่ใช้ควบคุมโทรศัพท์มือถือ	31
4.3 การเชื่อมต่อโทรศัพท์มือถือกับระบบควบคุม	31
4.4 วงจรสมบูรณ์ของโครงการ	32
4.5 ผลที่ได้จากโทรศัพท์มือถือภาครับ	32

สารบัญตาราง

ตารางที่	หน้า
2.1 การจัดการคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS232 แบบ DB9 และ DB25	10
2.2 รหัสตัวอักษร 7 บิต	13
2.3 รหัสตัวอักษร 7 บิต(ต่อ)	14
2.4 รหัสตัวอักษร 7 บิต(ต่อ)	15
2.5 ส่วนประกอบของข้อความสตริงการรับข้อความ sms	16
2.6 ส่วนประกอบของข้อมูลที่ส่ง	17
2.7 แสดงการแปลงตัวอักษร	19
2.8 แสดงตารางความจริงการถอดรหัสไอซี MT8870	24



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

สังคมในปัจจุบันนี้ มีปัญหาการโจรกรรมตามอาคารบ้านเรือนอยู่เสมอ ซึ่งสร้างความเดือดร้อนแก่เจ้าของบ้านอย่างมาก อีกทั้งเป็นการยากต่อการจับตัวผู้กระทำผิดมาลงโทษและชดใช้ค่าเสียหาย

เทคโนโลยีของการสื่อสารในปัจจุบันนั้น ได้พัฒนาไปอย่างรวดเร็ว จนสามารถกล่าวได้ว่าเป็นยุคของการสื่อสารไร้พรมแดน โทรศัพท์มือถือได้เข้ามามีบทบาทในการดำเนินชีวิตของผู้คนมากขึ้น และด้วยเหตุผลทางด้านราคาต่อหน่วยของเครื่องโทรศัพท์ที่ถูกลง จึงทำให้บุคคลทั่วไปสามารถเป็นเจ้าของโทรศัพท์มือถือได้ง่ายขึ้น

การพัฒนาระบบเตือนภัยอิเล็กทรอนิกส์สำหรับป้องกันการบุกรุก และระบบการส่งข้อความสั้นให้กับตัวเจ้าของบ้านเองจึงเป็นสิ่งที่จำเป็นต่อสังคมในตอนนี้ ดังจะเห็นได้จากการที่มีบริษัทรับติดตั้งระบบรักษาความปลอดภัยที่เพิ่มจำนวนอย่างต่อเนื่อง

ส่วนประกอบของโครงการชิ้นนี้มีดังนี้

1. ชุดควบคุมซึ่งใช้ Microcontroller ของ Atmel เบอร์ AT89C51
2. โทรศัพท์มือถือทางด้านส่ง (อยู่ภายในบริเวณที่มีการป้องกันการบุกรุก)
3. โทรศัพท์มือถือทางด้านรับ (อยู่กับตัวเจ้าของบริเวณที่มีการป้องกันการบุกรุก)

1.2 เนื้อหาโดยสังเขปของปฏิญญาพันธ

เนื้อหาของปฏิญญาพันธฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึง เหตุผลที่นำเอาเทคโนโลยีการส่งข้อความสั้น (SMS) บนโทรศัพท์มือถือมาประยุกต์ใช้ในโครงการนี้

บทที่ 2 กล่าวถึงหลักการ ความรู้ทางทฤษฎีต่างๆ ที่ใช้ในการทำปฏิญญาพันธฉบับนี้

บทที่ 3 การออกแบบ ขั้นตอนต่างๆ รวมไปถึงการสร้างโครงการขั้นนี้

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 บทสรุปและบทวิจารณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

2.1 รายละเอียดและขอบเขตของโครงการ

เป็นระบบเตือนภัยอิเล็กทรอนิกส์สำหรับป้องกันการบุกรุก โดยใช้ไมโครคอนโทรลเลอร์ ของ Atmel เบอร์ AT89C51 เป็นตัวควบคุม ระบบเตือนภัยอิเล็กทรอนิกส์ซึ่งประกอบไปด้วยตัวเซนเซอร์ต่างๆ เช่น อุปกรณ์ตรวจจับสิ่งต่างๆ (คนและสัตว์) รวมไปถึงระบบการเตือนภัยเมื่อมีผู้บุกรุกเข้ามายังภายในบ้าน

โดยเริ่มจากต่อเซนเซอร์เข้ากับขาของไมโครคอนโทรลเลอร์แล้วทำการโปรแกรมลงบนไมโครคอนโทรลเลอร์ หลังจากนั้นเมื่อเซนเซอร์ตรวจจับคน เซนเซอร์เหล่านั้นจะทำการส่งสัญญาณไปยังตัวควบคุม (Microcontroller) เมื่อตัวควบคุมได้รับสัญญาณก็จะทำการส่งสัญญาณให้เอาท์พุท ในที่นี้ก็คือการแจ้งเตือนข้อความสั้น (SMS) ผ่านทางโทรศัพท์มือถือ

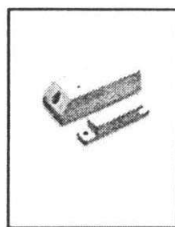
2.2 อินพุตที่ใช้ในการตรวจจับสิ่งแปลกปลอม

- เครื่องตรวจจับการเคลื่อนไหว (PIR Detector) (จากบริษัท Maxwell)

เป็นอุปกรณ์ตรวจจับความเคลื่อนไหวด้วยอินฟราเรด (Infrared) ในบางรุ่นสามารถตรวจจับได้ไกลถึง 12 – 15 เมตร และในบางรุ่นมีระบบป้องกันการทำลายอุปกรณ์ โดยจะส่งสัญญาณเตือนภัยเมื่อฝาครอบอุปกรณ์ถูกเปิดออก

หลักการของตัวตรวจจับความเคลื่อนไหว นั้นจะเป็นการใช้อุปกรณ์ ที่เรียกว่า PIR ซึ่งทำมาจากวัสดุประเภท Pyroelectric โดยจะทำการตรวจสอบรังสีอินฟราเรด (Infrared) จากร่างกายสิ่งมีชีวิต

รูปที่ 2.1



รูปที่ 2.2

- อุปกรณ์ตรวจจับการเปิดประตูหน้าต่าง (จากบริษัท วิกอมเทค จำกัด)

อุปกรณ์ติดตั้งที่ประตูหน้าต่าง ป้องกันการบุกรุก โดยจะส่งสัญญาณเตือนภัยเมื่ออุปกรณ์ทั้ง 2 ชั้นแยกห่างจากกัน (ประตู หน้าต่าง เปิดออก) ในบางรุ่นจะเตือนภัยผ่านเครื่องควบคุม และบางรุ่นจะมีเสียงเตือนภัยจากภายในอุปกรณ์



รูปที่ 2.3

- เครื่องตรวจจับควันไฟ (จากบริษัท วิกอมเทค จำกัด)

ใช้ตรวจจับควันไฟ และส่งสัญญาณเตือนภัยเมื่อมีความหนาแน่นของควันในระดับที่ตั้งค่าไว้ เป็นการป้องกันการเกิดเพลิงไหม้ในระยะเริ่มต้น

2.3 หลักการทำงานโดยรวมของวงจร

เริ่มต้น ส่วนของภาคจ่ายไฟนั้น เราได้ใช้ IC เบอร์ 7805 เพื่อทำการแปลงไฟจาก 12 โวลต์ให้เหลือ 5 volts เพื่อจ่ายไฟให้กับไมโครคอนโทรลเลอร์ โดยเมื่อไมโครคอนโทรลเลอร์รับไฟ 5 volt แล้ว จะทำงานได้ โดยเมื่อเรารับค่า จาก sensor แล้ว microcontroller จะทำการประมวลผลตามคำสั่งที่เราเขียนขึ้น และจะทำการติดต่อกับโทรศัพท์มือถือด้วยการต่อด้วยสายดาต้าลิงค์แบบ serialจากนั้นก็ทำการติดต่อกับโทรศัพท์มือถือด้วยโปรแกรมผ่านสาย serial โดยเมื่อมือถือติดต่อกับไมโครคอนโทรลเลอร์ได้แล้ว จะต้องทำการเขียนโปรแกรมควบคุมโทรศัพท์มือถือให้ส่งข้อความ(SMS) ด้วย กลุ่มคำสั่ง AT Command ซึ่งเราจะควบคุมการส่ง ข้อความไปยังภาครับ ซึ่งจะส่งไปโดยผ่าน IC ที่มีชื่อว่า MT8870 ซึ่งเป็นไอซีที่มีอยู่ในโทรศัพท์มือถือ ซึ่งทำหน้าที่ในการถอดรหัสความถี่ DTMF ให้เป็นเลขฐาน 2 จำนวน 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไมโครคอนโทรลเลอร์

Microcontroller AT89C51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ ซีพียูขนาด 18 บิต
- ภายในมีหน่วยความจำโปรแกรมแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์มีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบ 2 ทิศทาง สามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
- มีวงจรถ่ายโอนข้อมูลแบบฟูลดูเพล็กซ์
- ไทม์เมอร์/เคาน์เตอร์ ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแสงกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้ สูงสุด 64 กิโลไบต์
- มีวงจรถ่ายโอนสัญญาณพิกที่อยู่ภายในชิป
- มีวงจรถ่ายโอนข้อมูลแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวงจรถ่ายโอนพิกที่อยู่ในตัวสำหรับในอนุกรม AT89Sxx

ไมโครคอนโทรลเลอร์มีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0 ถึง พอร์ต 3 แต่ละพอร์ตจะมีขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง

INTERRUPT

คือการขัดจังหวะการทำงานของโปรแกรมหลักในไมโครคอนโทรลเลอร์ เพื่อไปทำงานในโปรแกรมย่อยๆ และเมื่อเสร็จการทำงานในโปรแกรมย่อยแล้วจะกลับมาทำงานในโปรแกรมหลักต่อจากบริเวณที่หยุดการทำงานไว้ต่อก่อนเข้า interrupt

Interrupt แบ่งเป็น Internal interrupt เช่น Timer Interrupt และ External interrupt

TIMER

สามารถโปรแกรมให้เป็นตัวนับ หรือ เป็นตัวจับเวลาก็ได้ เมื่อ Timer ถึงค่าที่กำหนด จะแสดงผลออกมาทางบิต Flag

การทำงานของ Timer นี้ ค่าใน register จะเพิ่มขึ้นทุกๆ machine cycle เช่นถ้าทำงานที่ 12 MHz และหนึ่ง machine cycle ใช้สัญญาณ 12 ลูก ดังนั้นค่าใน register timer จะเพิ่มขึ้น 1 ทุกๆ 1 us ถ้าหากเรา

“1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตยังใช้งานติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ตใดเป็นขาอินพุต สามารถทำได้โดย การเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่ในการใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P 3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 0 หรือขา INTO

P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณ ไทม์เมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขา รีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 เมกซีไนเซกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE/PROG (Address Latch Enable / Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับ โปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51

ขา PSEN (Program store Enable) นี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละเมกซีไนเซกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขา นี้ จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/ Vpp (External Access enable / Programming voltage input) ใช้สำหรับเลือกการติดต่อกับหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น

“1” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขา 1 นี้ยังเป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันไฟ +12 V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.6 การรับส่งข้อมูลแบบอนุกรม

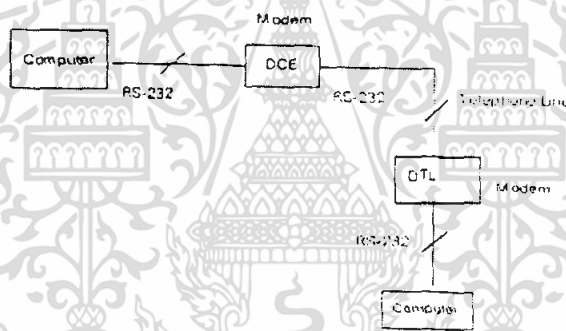
การรับส่งข้อมูลแบบอนุกรม ซึ่งมีข้อดีคือ ใช้สายสัญญาณน้อยและส่งได้เป็นระยะทางไกล สำหรับโครงการใช้การติดต่อกับไมโครคอนโทรลเลอร์ รับส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์เพื่อทำการเปลี่ยนระดับสัญญาณไฟฟ้าได้มาตรฐานในการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องไมโครคอมพิวเตอร์ ซึ่งสัญญาณข่าวสารที่ได้ถูกประมวลโดยโปรแกรมคอมพิวเตอร์ เพื่อแสดงออกทางหน้าจอคอมพิวเตอร์

2.6.1 การรับส่งข้อมูลแบบอนุกรม

โดยปกติเครื่องไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรมชื่อว่า อาร์เอส-232 อยู่ในตัวมันเองอยู่แล้ว ซึ่งพอร์ต อาร์เอส 232 นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรม (Universal Asynchronous Adapter) เหตุที่มีชื่อเรียกว่า RS-232 ก็เนื่องมาจาก สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ ของอเมริกา หรือ อีไอเอ (RS-232 : Recommended Standard Number 232, EIA : Electronic Industry Association) ได้กำหนดมาตรฐานของอุปกรณ์สื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า อาร์เอส-232 (ความจริงแล้วมาตรฐานของการรับส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับเครื่องไมโครคอมพิวเตอร์)

2.6.2 มาตรฐาน RS-232c

มาตรฐาน RS-232c ได้จัดพิมพ์ขึ้นเมื่อ ค.ศ 1969 อาร์เอสย่อมาจาก Recommended Standard ส่วน 232 คือหมายเลขบ่งบอกมาตรฐานตัวนี้ และ c เป็นหมายเลขฉบับของหมายเลขตัวนี้ จุดประสงค์ของมาตรฐาน อาร์เอส-232 ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูล (DCE : Data Communication Equipment) กับอุปกรณ์สื่อสารข้อมูล (DCE : Data Communication Equipment) สำหรับผู้ใช้ คอมพิวเตอร์ทั่วไป ดีทีอี ก็หมายถึง ตัวไมโครคอมพิวเตอร์ ส่วน ดีซีอี หมายถึง โมเด็ม (Modem) และอุปกรณ์อื่นๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรม อาจจะเป็นไปได้ทั้ง ดีทีอี และ ดีซีอี ซึ่งจะขึ้นอยู่กับผู้ผลิต สำหรับข้อแตกต่างของ ดีทีอี และ ดีซีอี จะเห็นได้จากรูปที่ 2.24 ซึ่งจากรูปจะเห็น ได้ว่า อาร์เอส-232c มีส่วนสำคัญอย่างมากสำหรับการสื่อสารข้อมูลระหว่างไมโครคอมพิวเตอร์



รูปที่ 2.5 การใช้มาตรฐาน อาร์เอส-232c เชื่อมต่ออุปกรณ์

2.6.3 อุปกรณ์ ดีทีอี และ ดีซีอี

ตามมาตรฐาน อาร์เอส-232c อุปกรณ์ ดีทีอีควรใช้หัวต่อตัวผู้และอุปกรณ์ ดีซีอี ควรใช้หัวต่อตัวเมีย อย่างไรก็ตามผู้ผลิตไม่ได้ปฏิบัติตามกฎนี้เสมอไปดังนั้นจึงไม่แยกแยะอุปกรณ์ ดีทีอี และ ดีซีอี โดยการมองผ่านๆ ได้เสมอไป เมื่อทราบว่าคุณสมบัติหนึ่งเป็น ดีทีอี และอีกตัวหนึ่งเป็น ดีซีอี ในทางทฤษฎีแล้วสามารถเชื่อมต่อได้อย่างง่ายดาย โดยการเชื่อมต่อสายที่มีหมายเลขตรงกัน เช่น เส้นที่ 2 กับ 2, 3 กับ 3 เป็นต้น เรียกว่า การเชื่อมต่อแบบตรงไปตรงมาแต่มีผู้ผลิตบางรายไม่ได้ทำตามมาตรฐานและทำให้เกิดปัญหาหลายอย่างเช่นเดียวกับวิธีการจัดการกับสถานการณ์ที่อุปกรณ์ทั้ง 2 เป็น ดีทีอี หรือ ดีซีอี

เหมือนกัน ในตอนนี้ถือว่าอุปกรณ์หนึ่งเป็น ดีทีอี และอีกตัวหนึ่งเป็น ดีซีอี แต่ละฝ่ายจะส่งสัญญาณที่อีกฝ่ายต้องการบนสายที่ตรงกัน

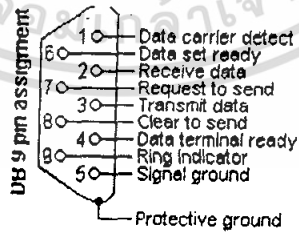
2.6.4 การกำหนดขา RS-232 สำหรับ DB-9

การกำหนดขาสัญญาณของคอนเน็กเตอร์ (connector) อนุกรม 9 ขา (DB-9) แสดงใน ตารางที่

2.1

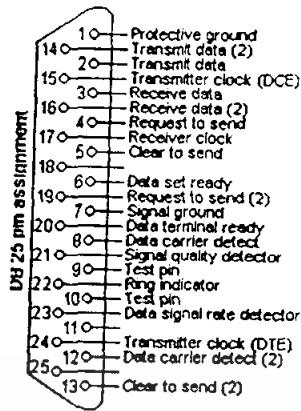
ตารางที่ 2.1 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน อาร์เอส-232 ทั้งแบบ ดีบี-9 และ ดีพี-25

ตำแหน่งขา		ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
DB-9	DR-25		
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Sent : CTS	อินพุต
9	22	Ring Indicator ; RI	อินพุต



รูปที่ 2.6 คำอธิบายการทำงานของแต่ละขาที่ใช้งานของ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 คำอธิบายการทำงานของแต่ละขาที่ใช้งานของ DB-25

- ดีซีอี (Data carrier Detect : DCD) หรืออาจเรียกว่าซีดี (carrier Detect : CD) ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ใช้งานมากนัก
- อาร์ดี (Received Data :RxD) เป็นทางสัญญาณเข้าไปยัง ดีทีอี หรือไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามาขานี้จะมีสถานะภาพลอจิกเป็น “1”
- ทีดี (Transmit Data : TD) เป็นสัญญาณที่ส่งออกจาก ดีทีอี (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อกับไมโครคอมพิวเตอร์ตัวอื่น เมื่อไม่มีสัญญาณส่งออกสถานะภาพของลอจิกที่ขานี้มีค่าเท่ากับ “1” หรือเทียบเท่า สต๊อป บิต
- ดีทีอาร์ (Data Terminal Ready ; DTR) เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยขา ดีทีอาร์นี้ต้องเชื่อมต่อกับขา ดีเอสอาร์ของอุปกรณ์ปลายทางและขาดีทีอาร์ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา ดีเอสอาร์ ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ นัล โมเด็ม (Null modem) ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา ดีทีอาร์ และ ดีเอสอาร์ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา ดีเอสดีด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจสัญญาณพาห์
- จีเอ็นดี (Signal Ground : GND) กราวด์ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ดีเอสอาร์ (Data set Ready : DSR) ขานี้จะใช้คู่กับขา ดีทีอาร์ เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์ปลายทาง ซึ่งขา ดีเอสอาร์ นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา ดีทีอาร์
- อาร์ทีเอส (Request to send :RTS) เป็นขาสำหรับสัญญาณร้องขอให้ทาง อุปกรณ์ ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ อาร์ทีเอส ก็คือขาซีทีเอส ในกรณีที่ใช้การเชื่อมต่อแบบ โมเด็มจะต้องเชื่อมต่อขา อาร์ทีเอส และ ซีทีเอส ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- ซีทีเอส (Clear To Sent :CTS) ขานี้จะคอยรับสัญญาณจาก อาร์ทีเอส เมื่อรับสัญญาณได้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้น ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมรับข้อมูลหรือไม่
- อาร์ไอ (Ring Indicator : RI) ใช้แสดงสถานะสัญญาณเรียกสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไป สายนี้จะไม่ถูกใช้งาน จะใช้ก็ต่อเมื่อมีการเชื่อมต่อ โมเด็มและโปรแกรมมีการตรวจสอบสัญญาณเท่านั้น

2.7 การรับส่งข้อความสั้น (SMS)

European Telecommunication Standard Institute (ETIS) เป็นองค์กรที่ไม่แสวงหาผลกำไรทำหน้าที่กำหนดมาตรฐานทางด้านโทรคมนาคมได้กำหนดมาตรฐานการส่ง เอสเอ็มเอส ไว้ในคู่มือ GSM 03.40 และ GSM03038 สามารถส่งได้สูงถึง 160 ตัวอักษร โดยแต่ละตัวอักษรใช้รหัสขนาด 7 บิต ที่กำหนดไว้ในตารางที่ 2.2-2.4 นอกจากนั้นยังมีการใช้ตัวอักษรชนิดอื่นๆเช่น ขนาด 8 บิต หรือ 16 บิต ซึ่งมีวัตถุประสงค์เพื่อการใช้งานที่แตกต่างออกไป ซึ่งในโครงการฉบับนี้จะพูดถึงเฉพาะ 7 บิต เท่านั้น

ตารางที่ 2.2 รหัสตัวอักษรชนิด 7 บิต (7 -bit default alphabet)

Hex	Dec	Character name	Character	ISO-8859-1
0x00	0	COMMERCIAL AT	@	64
0x01	1	POUND SIGN	£	163
0x02	2	DOLLAR SIGN	\$	36
0x03	3	YEN SIGN	¥	165
0x04	4	LATIN SMALL LETTER E WITH GRAVE	è	232
0x05	5	LATIN SMALL LETTER E WITH ACUTE	é	233
0x06	6	LATIN SMALL LETTER U WITH GRAVE	ù	249
0x07	7	LATIN SMALL LETTER I WITH GRAVE	ì	236
0x08	8	LATIN SMALL LETTER O WITH GRAVE	ò	242
0x09	9	LATIN CAPITAL LETTER C WITH	ç	199
0x0A	10	LINE FEED		10
0x0B	11	LATIN CAPITAL LETTER O WITH	Ø	216
0x0C	12	LATIN SMALL LETTER O WITH	ø	248
0x0D	13	CARRIAGE RETURN		13
0x0E	14	LATIN CAPITAL LETTER A WITH RING	Å	197
0x0F	15	LATIN SMALL LETTER A WITH RING	å	229
0x10	16	GREEK CAPITAL LETTER DELTA	Δ	
0x11	17	LOW LINE		95
0x12	18	GREEK CAPITAL LETTER PHI	Φ	
0x13	19	GREEK CAPITAL LETTER GAMMA	Γ	
0x14	20	GREEK CAPITAL LETTER LAMBDA	Λ	
0x15	21	GREEK CAPITAL LETTER OMEGA	Ω	
0x16	22	GREEK CAPITAL LETTER PI	Π	
0x17	23	GREEK CAPITAL LETTER PSI	Ψ	
0x18	24	GREEK CAPITAL LETTER SIGMA	Σ	
0x19	25	GREEK CAPITAL LETTER THETA	Θ	
0x1A	26	GREEK CAPITAL LETTER XI	Ξ	
0x1B	27	ESCAPE TO EXTENSION TABLE		
0x1B0A	27 10	FORM FEED		12
0x1B14	27 20	CIRCUMLEX ACCENT	^	94
0x1B28	27 40	LEFT CURLY BRACKET	{	123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 รหัสตัวอักษรชนิด 7 บิต (7 -bit default alphabet) (ต่อ)

0x1B29	27 41	RIGHT CURLY BRACKET	}	125
0x1B2F	27 47	REVERSE SOLIDUS (BACKSLASH)	\	92
0x1B3C	27 60	LEFT SQUARE BRACKET	[91
0x1B3D	27 61	TILDE	~	126
0x1B3E	27 62	RIGHT SQUARE BRACKET]	93
0x1B40	27 64	VERTICAL BAR		124
0x1B65	27 101	EURO SIGN	€	164(ISO-8859-
0x1C	28	LATIN CAPITAL LETTER AE	Æ	198
0x1D	29	LATIN SMALL LETTER AE	æ	230
0x1E	30	LATIN SMALL LETTER SHARP S	ß	223
0x1F	31	LATIN CAPITAL LETTER E WITH	É	201
0x20	32	SPACE		32
0x21	33	EXCLAMATION MARK	!	33
0x22	34	QUOTATION MARK	"	34
0x23	35	NUMBER SIGN	#	35
0x24	36	CURRENCY SIGN	¤	164(ISO-8859-1)
0x25	37	PERCENT SIGN	%	37
0x26	38	AMPERSAND	&	38
0x27	39	APOSTROPHE	'	39
0x28	40	LEFT PARENTHESIS	(40
0x60	96	INVERTED QUESTION MARK	¿	191
0x61	97	LATIN SMALL LETTER A	a	97
0x62	98	LATIN SMALL LETTER B	b	98
0x63	99	LATIN SMALL LETTER C	c	99
0x64	100	LATIN SMALL LETTER D	d	100
0x65	101	LATIN SMALL LETTER E	e	101
0x66	102	LATIN SMALL LETTER F	f	102
0x67	103	LATIN SMALL LETTER G	g	103
0x68	104	LATIN SMALL LETTER H	h	104
0x69	105	LATIN SMALL LETTER I	i	105
0x6A	106	LATIN SMALL LETTER J	j	106
0x6B	107	LATIN SMALL LETTER K	k	107
0x6C	108	LATIN SMALL LETTER L	l	108
0x6D	109	LATIN SMALL LETTER M	m	109
0x6E	110	LATIN SMALL LETTER N	n	110
0x6F	111	LATIN SMALL LETTER O	o	111
0x70	112	LATIN SMALL LETTER P	p	112
0x71	113	LATIN SMALL LETTER Q	q	113
0x72	114	LATIN SMALL LETTER R	r	114
0x73	115	LATIN SMALL LETTER S	s	115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 รหัสตัวอักษรชนิด 7 บิต (7-bit default alphabet) (ต่อ)

0x74	116	LATIN SMALL LETTER T	t	116
0x75	117	LATIN SMALL LETTER U	u	117
0x76	118	LATIN SMALL LETTER V	v	118
0x77	119	LATIN SMALL LETTER W	w	119
0x78	120	LATIN SMALL LETTER X	x	120
0x79	121	LATIN SMALL LETTER Y	y	121
0x7A	122	LATIN SMALL LETTER Z	z	122
0x7B	123	LATIN SMALL LETTER A WITH DIAERESIS	ä	228
0x7C	124	LATIN SMALL LETTER O WITH DIAERESIS	ö	246
0x7D	125	LATIN SMALL LETTER N WITH TILDE	ñ	241
0x7E	126	LATIN SMALL LETTER U WITH DIAERESIS	ü	252
0x7F	127	LATIN SMALL LETTER A WITH GRAVE	à	224

2.7.1 โหมดการรับส่งข้อมูล

การรับส่งข้อมูล เอสเอ็มเอส มีอยู่ด้วยกัน 2 โหมดคือ เท็กซ์ โหมด (Text Mode) และพีดียู โหมด (PDU: Protocol Description Unit Mode) การส่งข้อความในเท็กซ์โหมดนั้นจะเป็นการนำข้อความที่ต้องการมาเข้ารหัสก่อน แล้วค่อยส่งข้อมูลในพีดียูโหมดอีกที อย่างไรก็ตามในโทรศัพท์มือถือบางรุ่นไม่สนับสนุนการใช้งานเท็กซ์โหมด ซึ่งการเข้ารหัสส่ง และถอดรหัสรับ สำหรับในเท็กซ์โหมดนี้มีหลายแบบด้วยกันเช่น “PCCP437”, “PCDN”, “8859-1”, “IRA” และ “GSM” เมื่อเราเชื่อมต่อกับโทรศัพท์มือถือเพื่อที่จะอ่านข้อความเราสามารถตั้งค่าการเข้ารหัส/ถอดรหัส ได้โดยใช้กลุ่มคำสั่ง AT command แต่ถ้าเราอ่านข้อความจากจอของโทรศัพท์มือถือจะเลือกการถอดรหัสที่เหมาะสมให้เองโดยอัตโนมัติ

การเชื่อมต่อโทรศัพท์มือถือเพื่อรับส่งข้อความสามารถเลือกใช้ได้ทั้ง 2 โหมดแต่จะเห็นได้ว่าการเลือกใช้เท็กซ์โหมดจะมีข้อจำกัดทั้งจากการที่มือถือบางรุ่นอาจไม่สนับสนุนและยังถูกจำกัดวิธีการเข้ารหัสและถอดรหัส ซึ่งมีเพียงไม่กี่แบบตามที่ได้กล่าวมาแล้วข้างต้นซึ่งในบางกรณีอาจไม่สะดวกนัก แต่ถ้าเลือกพีดียูโหมดจะสามารถเลือกหรือสร้างการเข้ารหัสและถอดรหัสได้ทุกรูปแบบตามที่ต้องการโดยไม่มีข้อจำกัด

2.7.2 การรับข้อความสั้น (sms) ในพีดียูโทมด

ถ้าหากทำการเชื่อมต่อกับ โทรศัพท์มือถือแล้วทำการอ่านข้อความ sms ที่อยู่ใน Inbox โดยใช้คำสั่ง AT+CMGR ข้อมูลที่ได้รับจะอยู่ในรูปของสตริงที่ประกอบด้วยข้อมูลของผู้ส่ง, ข้อมูล SMS Service Center (SMSC), Time stamp และอื่น ๆ ที่จำเป็นและตามด้วยส่วนท้ายสุดของสตริงจะขอ ยกตัวอย่าง โดยใช้โทรศัพท์ เครื่องที่ใช้ในปริญญาณิพนธ์ฉบับนี้ซึ่งได้ใช้โทรศัพท์ SIMENS รุ่น c-35 ซึ่งข้อความที่ส่งมาคือ hellohello จากโทรศัพท์มือถืออีกเครื่องหนึ่ง ข้อมูลสตริงจะอยู่ในรูปของตัวเลขฐานสิบหกและฐานสิบในบางส่วนซึ่งมีรายละเอียดดังนี้

06916681118088040A8180041 104410000403021219434820AE8329BFD4697D9EC37

ตารางที่ 2.5 ส่วนประกอบของสตริงการรับข้อความ SMS

กลุ่มตัวเลข 8 บิต	รายละเอียด
06	ความยาวของ SMS Information 6 Octets (bytes)
91	รูปแบบของเลขหมาย Smsc 91 หมายถึงเลขหมายแบบสากล (International format)
6681118088	เลขหมาย smsc (แบบ decimal semi- octets) ซึ่งจะเป็นเลขฐาน 10 สลับ nibble ในกรณีเลขหมายจริงของ service center คือ +6681118088
04	First octet of this sms- deliver message
0A	ความยาวของเลขหมายผู้ส่ง (0A hex= 10 ตัวเลข)
81	รูปแบบของเลขหมายผู้ส่ง 81 หมายถึง เลขหมายแบบสากล (International format)
8004110441	เลขหมายผู้ส่ง (แบบ decimal semi-octets) เป็นเลขฐาน 10 สลับ nibble หมายถึงเลขผู้ส่งที่แท้จริงคือ+8004110441
00	TP-PID (Protocal identifier) ในกรณีนี้คือ 00
00	TP-UDL (data coding scheme) 00 คือเข้ารหัสข้อความแบบ 7 bit Default Alphabet
40302121943482	TP-UDL User data length จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 10 ตัว
E8329BFD4697D9EC37	TP-UD ข้อความ hellohello ที่เข้ารหัสแล้วจากตัวอักษรแบบ 7 bits เป็น ข้อมูล byte ขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลทั้งหมดในตารางเป็นเลขฐานสิบหกขนาดแปดบิต ยกเว้นหมายเลข service center, เลขหมายผู้ส่ง time stampจะเป็นเลขฐานสิบหกขนาดแปดบิต สลับหลักเป็นคู่ๆ (สลับ nibble) ชั้นส่วนของข้อมูลที่เป็นข้อความนั้นเป็นเลขฐานสิบหก ขนาดแปดบิต เช่นกัน โดยข้อมูลนี้จะใช้แสดงข้อความที่ประกอบไปด้วยตัวอักษรขนาดเจ็ดบิต ซึ่งผ่านการแปลง (เข้ารหัส) ข้อมูลจากรหัสตัวอักษรขนาดเจ็ดบิต

ในส่วนของข้อมูลที่เป็นเลขฐานสิบ เช่น เลขหมายผู้ส่งตัวเลข ในแต่ละคู่(1 byte) จะถูกสลับกัน เช่น เลขหมายจริง +66 040114014 จะถูกสลับในแต่ละคู่เป็น 66 04 11 04 41 (66คือรหัสประเทศ ส่วนเลขหมวดของหมายเลขโทรศัพท์มือถือจะถูกตัดเลข 0 ออก เช่น 04 จะเหลือแค่4 เป็นต้น แล้วจึงนำตัวเลขทั้งหมดมาต่อกันแล้วสลับคู่) เช่นเดียวกันกับ time stamp ข้อมูล 40 30 21 21 94 34 81 ซึ่งมีรูปแบบเป็น YY/MM/DD (HH:MM:SS:ss) หมายถึง ข้อความนี้ส่งเมื่อ 04/03/12 12 :49 :43 : 28

2.8 การส่งข้อความ SMS ในพีดียูโทมด

ในที่นี้จะขอยกตัวอย่างการใช้โทรศัพท์มือถือรุ่น siemens รุ่น C-35 โดยใช้โหมดพีดียูไปให้ผู้รับ หมายเลข +66066277916

AT+CMGF=0 หมายถึงการเลือกพีดียูโทมด

AT+CSMS=0 หมายถึงชี้ว่าโทรศัพท์มือถือสนับสนุนการส่ง sms หรือไม่

AT+CMGS=22 หมายถึงต้องการส่งทั้งหมด 22 byte (ในตัวอย่างนี้ใช้22 byte โดยที่ไม่รวมเลข00 ที่อยู่หน้าสุด)

>0011000A81806627297610000AA0AE8329BFD4697D9EC37

//เมื่อพิมพ์ข้อความครบแล้วกด Ctrl+z ส่วนประกอบอธิบายในตารางที่2.6

ตารางที่ 2.6 ส่วนประกอบของข้อมูลที่ส่ง ..

กลุ่มตัวเลข 8 บิต(Octet)	รายละเอียด
00	ความยาวของ SMSC Information 00 หมายถึงให้ใช้ SMSC Information ที่เก็บอยู่ในเครื่อง (ปกติเครื่องที่สามารถส่ง sms ได้มี SMSC ภายในเครื่องอยู่แล้ว)
11	First octet of the SMS- SUBMIT message
00	TP- Message-Reference 00 คือให้เครื่องหมายเลข
0A	Address-Length ความยาวของเลขหมายผู้รับ (10ตัว)
81	Type-of-address (91 indicates international format of the phone number)

8066729761	หมายเลขโทรศัพท์ของผู้รับ (แบบ semi-octets) หมายเลขที่แท้จริงคือ+66066277916
00	TP-PID (Protocol identifier) เป็น 00
00	TP-DCS (Data coding scheme) เป็น 00
00	TP-Validity-Period AA หมายถึง ช่วงเวลาหมดอายุของข้อความ 4 วัน ถ้าภายในช่วงเวลานี้ ยังส่งไม่ถึงปลายทางข้อความจะถูกยกเลิกโดยอัตโนมัติ
40302121943482	TP- User data length จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 10 ตัว
E8329BFD4697D9EC37	TP-User-Data เป็นข้อมูลในส่วนของข้อความตัวอักษรแบบ 7 บิต hellohello ที่ผ่านการแปลง (เข้ารหัส) เป็นข้อมูลแบบ 8 บิต แล้ว โดยมีวิธีการตามหัวข้อการแปลงตัวอักษรชนิด 7 บิต เป็นข้อมูล 8 บิต

2.8.1 รหัสตัวอักษรชนิด 7 บิต (7 bits default alphabet)

ตัวอักษรชนิด 7 บิต ถูกกำหนดโดยคู่มือ GSM 03.38 เป็นตารางข้างต้น ซึ่งเปรียบเทียบกับรหัสฐาน 10 ของ ISO-8859-1 (ASCII)

2.8.2 การแปลงตัวอักษรชนิด 7 บิต เป็น ข้อมูล 8 บิต (Octet)

จากตารางที่ 2.6 ในส่วนของ TP-User-Data จะเป็นส่วนที่เราสามารถใส่รหัสของข้อความที่ต้องการส่ง แต่เนื่องจากเราไม่สามารถนำรหัสของตัวอักษรแบบ 7 บิต ไปใส่ได้โดยตรง จำเป็นต้องผ่านการแปลงให้เป็นรหัสข้อมูลแบบ 8 บิต ก่อน โดยตัวอย่างต่อไปนี้เป็นการแปลงข้อความ hellohello ยาว 10 ตัวอักษร ซึ่งแต่ละตัวเป็นอักษรชนิด 7 บิต ให้เป็นข้อมูล 8 บิต สำหรับใช้ในการส่ง sms

การแปลงเริ่มจากรหัส 7 บิต ด้วยตัวอักษรแรก (h) มาเติมข้างหน้าด้วย 1 บิต ท้ายสุดของรหัส 7 บิต ของตัวอักษรตัวที่ 2 (e) จะได้ผลลัพธ์ 8 บิต (1 byte) เป็น E8

ขั้นตอนต่อมาให้เอา 6 บิต ที่เหลือของตัวอักษรตัวที่ 2 มาเติมข้างหน้าด้วย 2 บิต ท้ายของรหัส 7 บิต ของอักษรตัวที่ 3 (l) จะได้ผลลัพธ์ 8 บิตเป็น 32 และทำเช่นนี้เรื่อยไปโดยจำนวนบิตที่นำมากระทำจะเพิ่มขึ้นเป็น 3 บิต เป็น 4 บิต จนกระทั่งหมดชุดตัวอักษรดังตัวอย่างการแปลงตัวอักษร หลังจากแปลงข้อความ hellohello จะได้ข้อมูลเป็นฐาน 19 จำนวน 9 byte E8 32 9B FD 46 97 D9 EC 37

ตารางที่ 2.7 แสดงการแปลงตัวอักษร

Char	h	e	l	l	o	h	e	l	l	o
7bits(Dec)	104	101	108	108	111	104	101	108	108	111
7bits(Bin)	1101000	1100101	1101100	1101100	1101111	1101000	1100101	1101100	1101100	1101111
8bits(Bin)	11101000	00110010	10011011	11111101	01000110	10010111	11011001	11101100	11011111	
8bits(Hex)	E8	32	9B	FD	46	97	D9	EC	37	

คำอธิบาย

เริ่มต้นนำตัวอักษรมาตรวจสอบว่าแทนด้วยเลขฐานสิบตัวใด จากนั้นนำเลขฐานสิบมาแปลงเป็นเลขฐานสอง (ในที่นี้แสดงการส่งคำว่า hellohello) ซึ่งจะได้ตามตารางข้างบน

วิธีการแปลงจากเลขฐานสองชนิด 7 บิตเป็นเลขฐานสองชนิด 8 บิต

1. แปลงตัว h เป็นเลขฐานสอง 7 บิตได้ 1100101 แล้วพักไว้ก่อน
2. นำตัว e ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1100101 จากนั้นนำตัวหลังสุดของตัว e (ที่เป็นเลขฐานสอง 7 บิตแล้ว คือ 1100101) มา 1 ตัว ซึ่งคือเลข 1 ไปใส่ไว้ในข้างหน้าสุดของ h (1100101) กลายเป็น 11100101 และ 1100101 กลายเป็น 110010
3. นำตัว l ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101100 จากนั้นนำตัวหลังสุดของตัว l มา 2 ตัวซึ่งคือ 00 ไปใส่ไว้หน้าสุดของตัว e ที่เหลือ 6 ตัว (110010) กลายเป็น 00110010 และ 1101100 จะกลายเป็น 110111
4. นำตัว l ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101100 จากนั้นนำตัวหลังสุดของตัว l มา 3 ตัวซึ่งคือ 100 ไปใส่ไว้หน้าสุดของตัว l ที่เหลือ 5 ตัว (11011) กลายเป็น 10011011 และ 1101100 จะกลายเป็น 110111
5. นำตัว o ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101111 จากนั้นนำตัวหลังสุดของตัว o มา 4 ตัวซึ่งคือ 1111 ไปใส่ไว้หน้าสุดของตัว l ที่เหลือ 4 ตัว (1101) กลายเป็น 11111101 และ 1101111 จะกลายเป็น 110
6. นำตัว h ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101000 จากนั้นนำตัวหลังสุดของตัว h มา 5 ตัวซึ่งคือ 01000 ไปใส่ไว้หน้าสุดของตัว o ที่เหลือ 3 ตัว (110) กลายเป็น 01000110 และ 1101000 จะกลายเป็น 11
7. นำตัว e ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1100101 จากนั้นนำตัวหลังสุดของตัว e มา 6 ตัวซึ่งคือ 100101 ไปใส่ไว้หน้าสุดของตัว h ที่เหลือ 2 ตัว (11) กลายเป็น 10010111 และ 1100101 จะกลายเป็น 1

8. นำตัว 1 ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101100 จากนั้นนำตัวหลังสุดของตัว 1 มา 7 ตัวซึ่งคือ 1101100 ไปใส่ไว้หน้าสุดของตัว e ที่เหลือ 1 ตัว (1) กลายเป็น 11011001 และ 1101100 จะหมดพอดี
9. นำตัว 1 ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101100 พักไว้ก่อน
10. นำตัว 0 ซึ่งเขียนเป็นเลขฐานสอง 7 บิตคือ 1101111 จากนั้นนำตัวหลังสุดของตัว 0 มา 1 ตัวซึ่งคือ 1 ไปใส่ไว้หน้าสุดของตัว 1 ที่เหลือ 7 ตัว (1101100) กลายเป็น 11101100 และ 1101111 จะกลายเป็น 110111
11. นำตัวเลขที่แปลงมาแล้วมาแปลงเป็นฐาน 16 ซึ่งจะได้ E8329BFD4697D9EC37

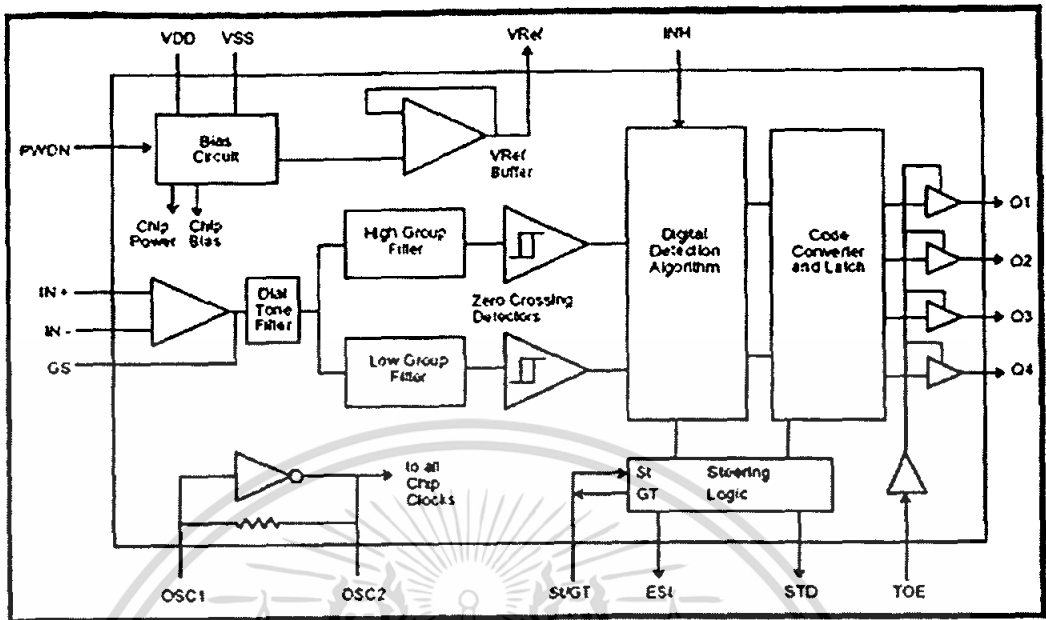
2.9 ไอซีถอดรหัสสัญญาณความถี่โทรศัพท์ (DTMF DECODER)

MT8870 เป็น ไอซีที่ใช้สำหรับถอดรหัสความถี่โทรศัพท์

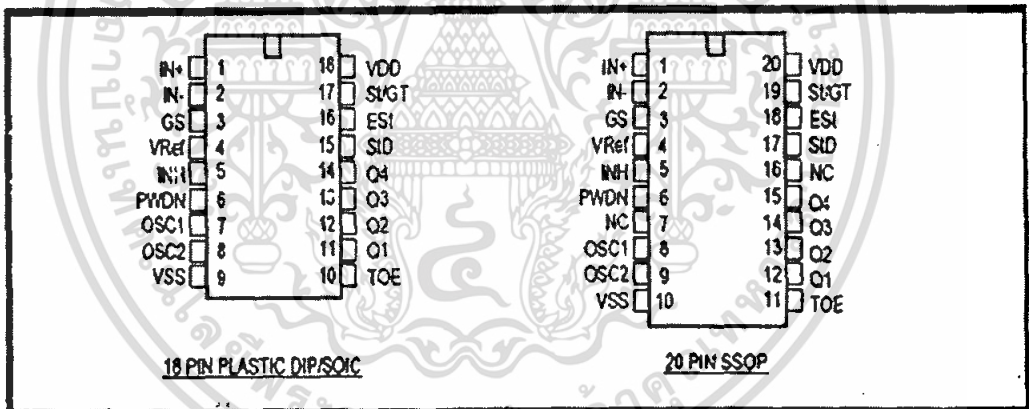
ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่มให้เป็นสัญญาณเลขฐานสอง 4 บิต คุณสมบัติการทำงานมีดังต่อไปนี้

- เป็นตัวรับและถอดรหัสความถี่โทรศัพท์
- ใช้กระแสไฟฟ้าน้อยและใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัว IC ได้
- เป็นไอซีมีคุณภาพสูง

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่ และวงจรถอดรหัสฟังก์ชันทาง ดิจิตอล เป็นไอซีที่สร้างด้วย เทคโนโลยี ISO-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของ สวิทช์คาปาซิเตอร์สำหรับกรองความถี่สูง และตัวส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อ ตรวจจับและถอดรหัส ทั้ง 16 ความถี่ออกเป็นฐานสองขนาด 4 บิตและเซกช่วงเวลาทีสัญญาณเข้ามาส่วน ภาคอินพุทเป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอก เอาท์พุทเป็นวงจรแลตซ์ 3 สถานะ



รูปที่ 2.8 แสดงโครงสร้างของ MT8870



รูปที่ 2.9 แสดงตำแหน่งขาของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยส่วนสำคัญ 5 ส่วน คือ

1. ภาคกรองความถี่
2. ภาคถอดรหัส
3. ภาคตรวจสอบสัญญาณ
4. ภาคกำเนิดความถี่
5. ภาคขยายสัญญาณความแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ภาคกรองความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่คือ ช่วงความถี่สูง และ ช่วงความถี่ต่ำโดยใช้วงจรกรองความถี่อันดับ 6 ชนิดสวิทช์คาปาซิเตอร์ ซึ่งความถี่ที่แยกได้ มี 2 ช่วงคือ ช่วงความถี่สูงและ ช่วงความถี่ต่ำ

2. ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิทัล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้องสัญญาณที่ขา EST ก็จะถูกแยกที่ฟ สำหรับค่าที่ถอดได้จากความถี่ต่าง ๆ

3. ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปทางเอาต์พุท จะมีการตรวจสอบความถี่ที่เข้ามาว่ามีระยะที่กำหนดให้โดยที่สังเกตจากระยะเวลาการกดปุ่มสวิทช์ความถี่ เพื่อให้มีช่วงความถี่ออกมาเป็นช่วงเวลาพอสมควรมิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้องส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอกสัญญาณที่ขา EST จะเป็น high นานใกล้เคียงกับระยะเวลาที่ความถี่ DTMF เข้ามาทำให้แรงดันสูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสจึงจะถอดรหัสออกมาเป็นตัวเลขขนาด 4 บิต

สัญญาณ DTMF ที่ถูกส่งมาจากภาคส่งสัญญาณจะถูกแยกองค์ประกอบของสัญญาณ เพื่อนำข้อมูลที่ได้เข้าไปในไมโครอีกครึ่ง ก่อนส่งไปควบคุมที่ภาคเอาต์พุตเกือบทั้งหมดนี้เราสามารถทำงานโดยใช้ไอซีเพียงตัวเดียวในปริณูณานิพนธ์ฉบับนี้คือ ใช้ไอซี MT 8870 ซึ่งเป็นไอซีถอดรหัสที่ใช้กันค่อนข้างแพร่หลาย

4. ภาคกำเนิดความถี่

5. ภาคขยายสัญญาณความแตกต่าง

การทำงานเริ่มจากสัญญาณภาคส่งสัญญาณถูกป้อนเข้ามาทางอินพุท ซึ่งจะขยายโดยออปแอมป์ จากนั้นสัญญาณที่ถูกขยายแล้วจะส่งต่อไปยังภาคกรองที่มี 2 สัญญาณ คือ ส่วนที่กรองสัญญาณ DTMF เป็นแบนด์พาสฟิลเตอร์ ทำหน้าที่กรองเอาเฉพาะกลุ่มความถี่ DTMF ออกมา เนื่องจากสัญญาณที่ส่งมาจากภาคส่งสัญญาณ อาจมีสัญญาณอื่นปะปนเข้ามาระหว่างทางได้จึงจำเป็นต้องกรองออกไปและ ส่วนที่

กรองสัญญาณความถี่สูง และ ความถี่ต่ำ ที่ภาคกรองความถี่สูงจะกรองเอาเฉพาะกลุ่มความถี่สูงของ DTMF และภาคกรองความถี่ต่ำก็จะกรองเอาเฉพาะกลุ่มความถี่ต่ำของ DTMF จากนั้นสัญญาณทั้ง 2 ส่วนที่ผ่านวงจรแล้ว จะถูกส่งต่อไปยังภาคถอดรหัสเพื่อทำการแปลงความถี่ DTMF ที่ได้ให้เป็น รหัสไบนารี

ถึงแม้ว่าภาคกรองความถี่จะมีถึงสองส่วน แต่ก็ไม่แน่ว่าความถี่ที่ผ่านวงจรกรองมาได้จะเป็น ความถี่ DTMF ที่ถูกต้องเสมอไปซึ่งอาจเป็นสัญญาณรบกวนที่มีความถี่ใกล้เคียงกันได้ทั้งนั้น จึงต้องมีการตรวจสอบสัญญาณเพื่อตรวจสอบว่าช่วงเวลาที่สัญญาณปรากฏ ถูกต้องตามที่ตั้งค่าไว้หรือไม่ กล่าวคือ ถ้าเวลาที่สัญญาณปรากฏสั้นกว่าเวลาที่ตั้งไว้ก็จะถือว่าสัญญาณนั้นมีความผิดพลาด ไม่ใช่ สัญญาณที่ต้องการ ก็จะไปควบคุมสถานะเอาต์พุตบัพเฟอร์ของไอซีให้เป็นไฮอิมพีแดนซ์นั่นคือจะไม่มี สัญญาณไบนารีปรากฏที่เอาต์พุตของไอซี IC MT8870 นี้มีอยู่ในโทรศัพท์มือถืออยู่แล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 แสดงตารางความจริงการถอดรหัสไอซี MT 8870

Digit	TOE	NIH	Est	Q4	Q3	Q2	Q1
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L				
B	H	H	L				
C	H	H	L				
D	H	H	L				

L= LOGIT LOW,H= LOGIT HIGH, Z= HIGH IMPEDANCE, X=DON'T CARE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนการออกแบบ

3.1 หลักการออกแบบ

การออกแบบโครงการนี้จะนำโทรศัพท์มือถือมาเชื่อมต่อกับไมโครคอนโทรลเลอร์และเซนเซอร์โดยใช้ไฟฟ้าตามบ้านหรือแบตเตอรี่ ซึ่งเซนเซอร์จะทำการตรวจจับสิ่งแปลกปลอมในพื้นที่ที่กำหนดไว้ และทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลตามที่ได้ตั้งโปรแกรมไว้ แล้วทำการส่งสัญญาณเอาต์พุตไปยังโทรศัพท์มือถือภาคส่ง โดยตัวไมโครคอนโทรลเลอร์จะทำการติดต่อกับโทรศัพท์มือถือภาคส่งผ่านทางสายดาต้าลิงค์เพื่อส่งการให้ทำการส่งข้อความสั้น (SMS) ต่อไป โดยข้อความที่จะถูกส่งออกไปยังโทรศัพท์มือถือนั้นจะถูกนำมาเข้ารหัสพีดียู (PDU code) และส่งออกโทรศัพท์มือถือซึ่งโทรศัพท์มือถือที่ใช้ระดับแรงดันเป็นแบบทีทีแอล (TTL) เช่นเดียวกับไมโครคอนโทรลเลอร์แต่เนื่องจากสายดาต้าลิงค์ (Data Link) ของโทรศัพท์โดยทั่วไปนั้นจะมีแมกซ์-232 อยู่ภายใน เพื่อใช้สำหรับแปลงระดับสัญญาณเมื่อติดต่อกับคอมพิวเตอร์ ดังนั้นเราจึงทำการตัดแปลงสายดาต้าลิงค์โดยตัดสายดาต้าลิงค์ส่วนที่มีวงจรมแมกซ์-232 อยู่ภายในออกและทำการต่อตรงเข้ากับไมโครคอนโทรลเลอร์เลย ที่ทำเช่นนี้ก็เพื่อเป็นการประหยัดไม่ต้องทำในส่วนของวงจรมอเตอร์อนุกรมเพิ่มเติม จากนั้นข้อมูลที่ถูกแปลงเป็นพีดียูโค้ดจะถูกส่งออกไปทางโทรศัพท์มือถือของภาคส่งแล้วดำเนินการส่งเอสเอ็มเอสต่อไป

ในส่วนของการเตือนภัยด้วยระบบเอสเอ็มเอส แบ่งออกเป็น 2 ส่วนด้วยกันคือ

1. ภาคส่ง
 - ส่วนของการคัดแยกข้อมูลและเข้ารหัสพีดียูโค้ดโดยใช้ไมโครคอนโทรลเลอร์
 - โทรศัพท์มือถือในส่วนของภาคส่ง
2. ภาครับ
 - โทรศัพท์มือถือในส่วนของภาครับ

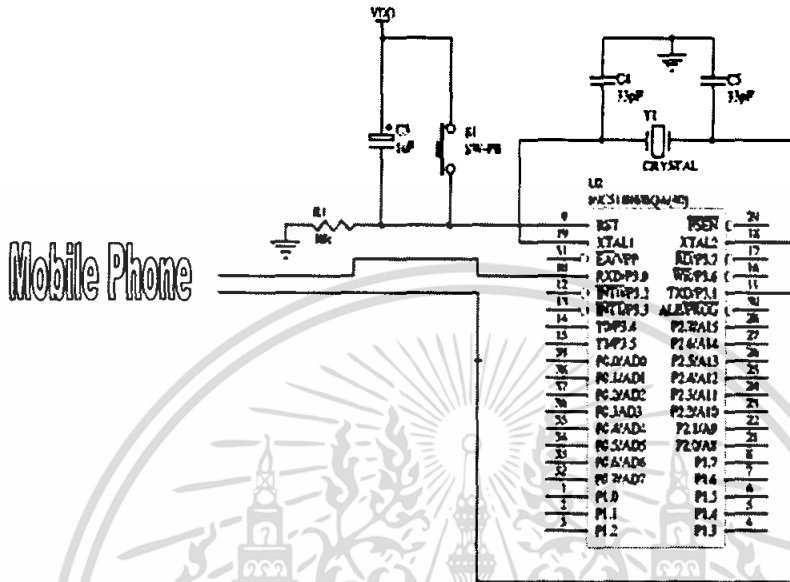
3.2 หลักการอินเตอร์เฟซโทรศัพท์มือถือ

ในที่นี้จะขอกล่าวถึงโทรศัพท์มือถือที่ใช้ในปริมาณพันธ์ ฉบับนี้คือ โทรศัพท์ มือถือยี่ห้อ siemens รุ่น c-35 ในส่วนของการติดต่อกันระหว่างโทรศัพท์มือถือกับไมโครคอนโทรลเลอร์นั้น เราใช้ สายดาต้าลิงค์ (data link) ในการส่งข้อมูลระหว่างกัน โดยส่วนของสัญญาณของโทรศัพท์มือถือและ ไมโครคอนโทรลเลอร์นั้นเป็นสัญญาณระดับ TTL ทั้งคู่ ดังนั้นจึงไม่ต้องแปลงสัญญาณใดๆและหาก สายส่งข้อมูล (data link) ที่ซื้อมามีวงจรแมกซ์-232 อยู่ภายในก็ต้องตัดส่วนนั้นทิ้งไปหรือจะประดิษฐ์ สายขึ้นมาเอง โดยในการใช้งานนี้จะใช้ทั้งหมด 5 ขา คือ ขา1,5,6,11 และ 12 ของพอร์ต ข้อมูลของ โทรศัพท์ มือถือ siemens รุ่น c-35 ดังตารางที่ 3.1 รูปแบบข้อมูลของโทรศัพท์ siemens รุ่น c-35 ส่ง ข้อมูลในลักษณะ 19200-N-8-1 คือมีอัตราการส่งข้อมูล 19200 บิตต่อวินาที ไม่มีพาริตีบิต ข้อมูลมีขนาด 8 บิต และมีบิตหยุด 1 บิต

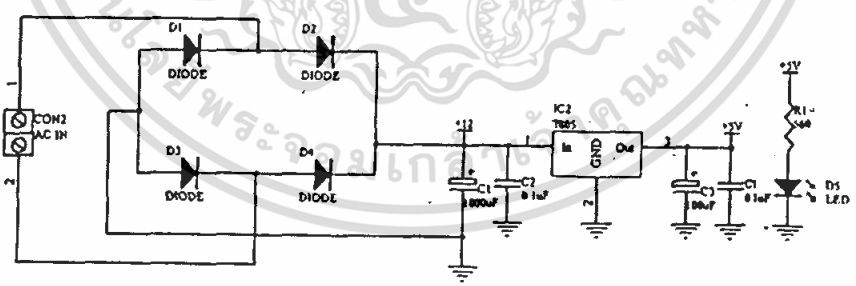


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรที่ใช้ในโครงการงาน

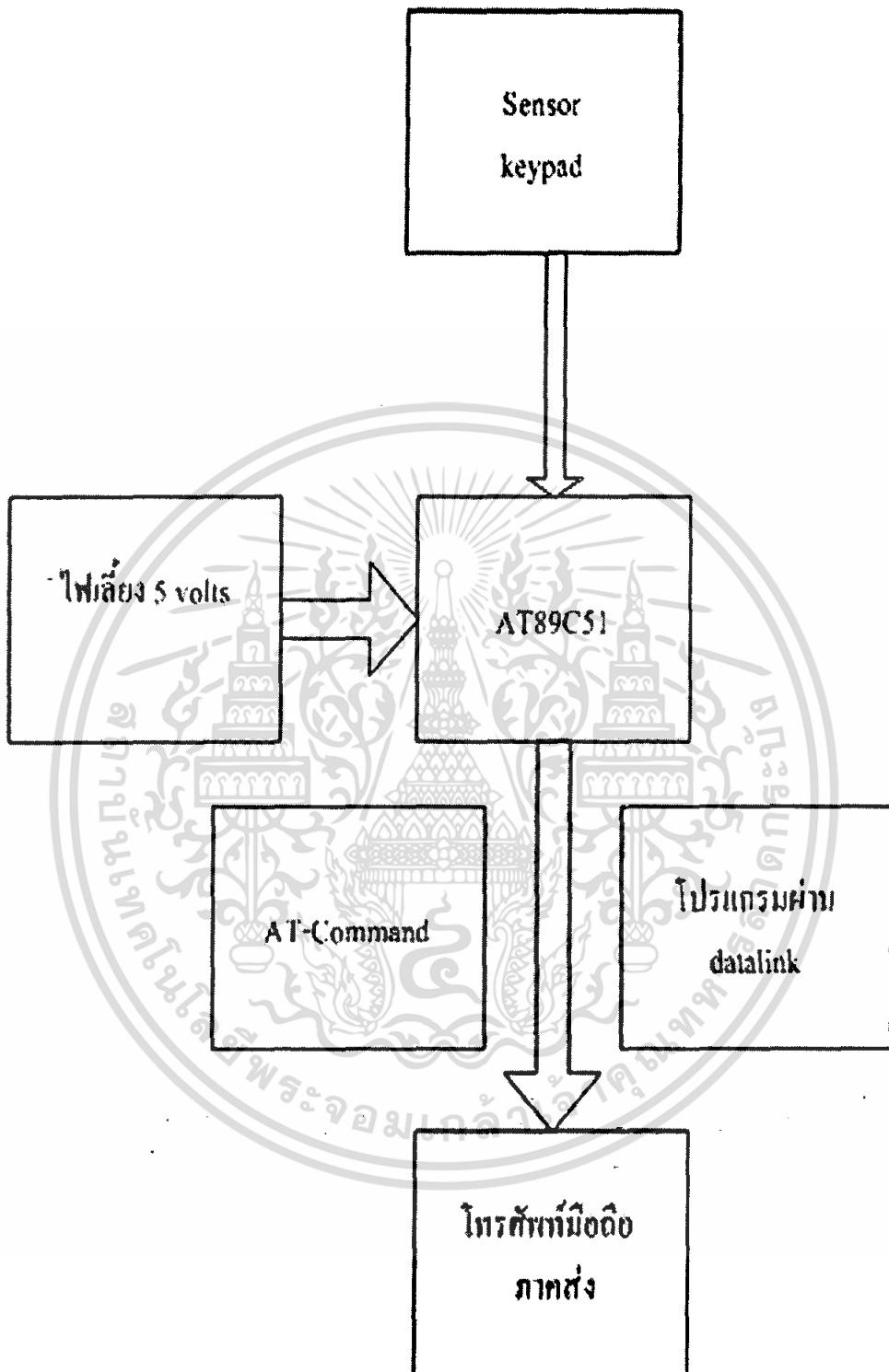


รูปที่3.1 วงจรAT89C51



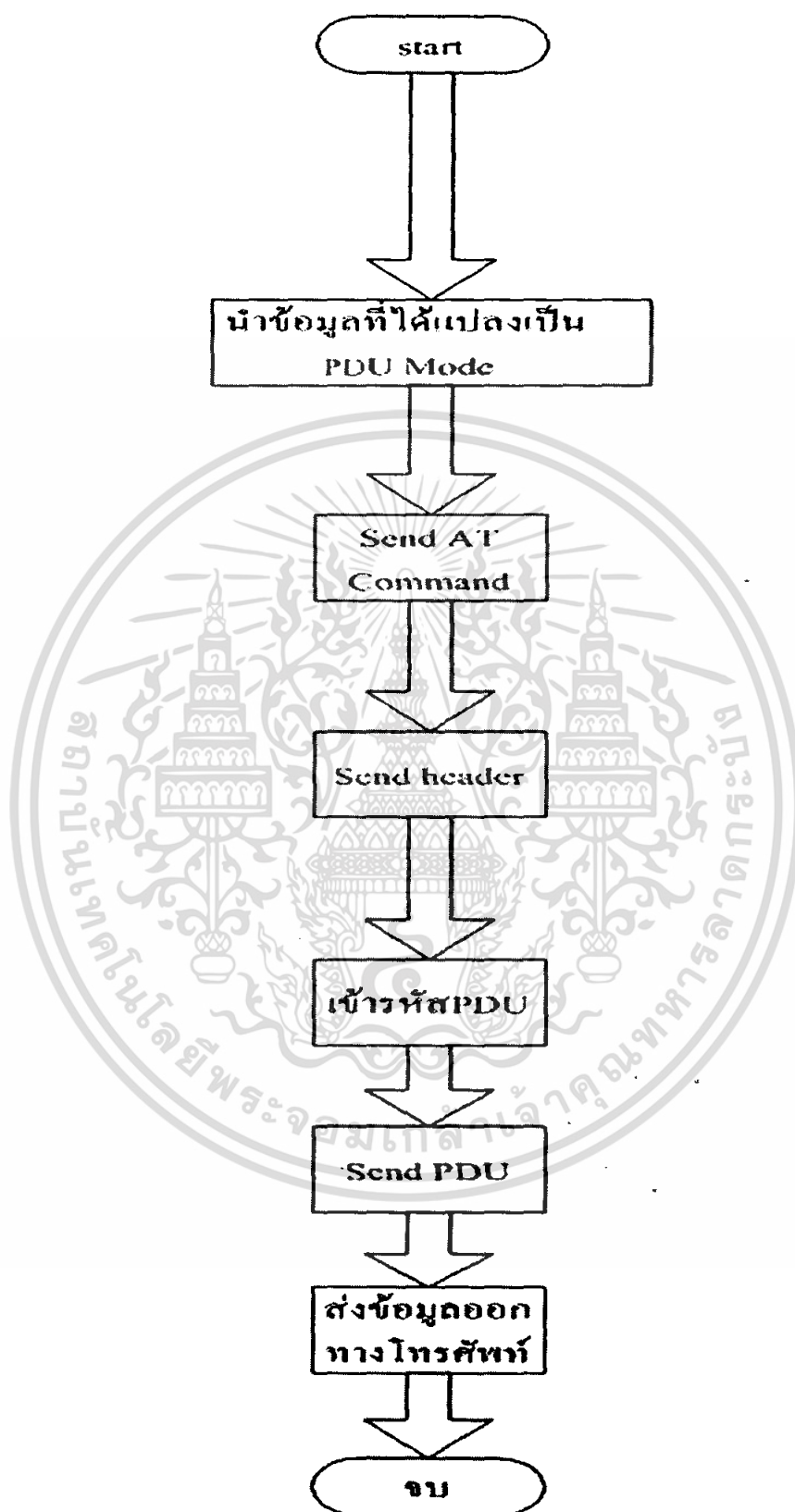
รูปที่3.2 วงจรพาวเวอร์ซัพพลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 รูปแบบการทำงานของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



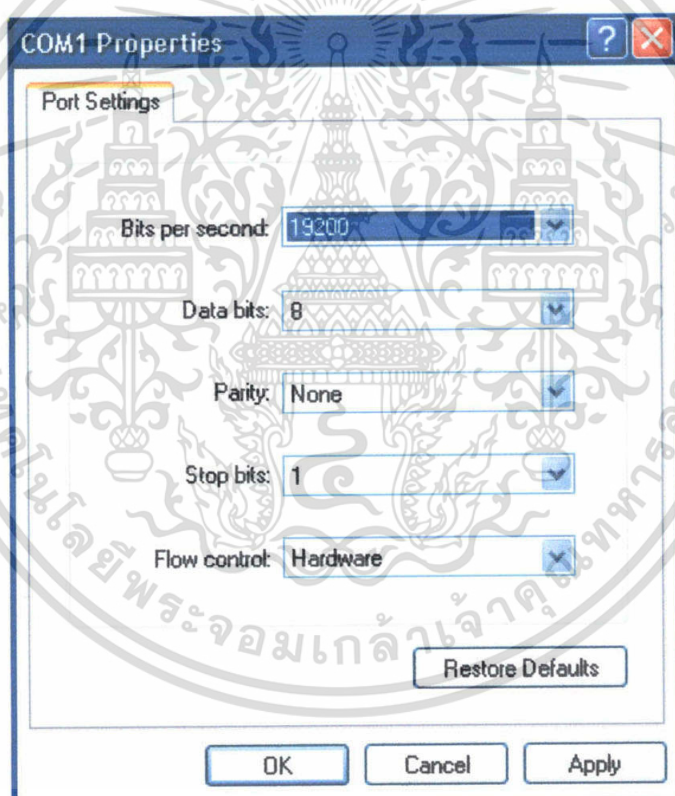
รูปที่3.4 รูปแบบการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เริ่มต้นจะต้องทดลองโทรศัพท์มือถือกับคอมพิวเตอร์ดูว่า สามารถใช้ชุดคำสั่ง เอที คอมมานด์ (AT-COMMAND) ได้หรือไม่ ซึ่งเราสามารถทดลองได้โดยการต่อสายดาต้าลิงค์ ระหว่าง คอมพิวเตอร์ กับ โทรศัพท์มือถือ จากนั้นเข้าโปรแกรมไฮเปอร์เทอร์มินอล แล้วกำหนดค่าบอดเรตให้กับ โทรศัพท์มือถือ ซึ่งต้องให้ค่าเท่ากับ 19,200 เฮิร์ตซ์ เพราะเนื่องจากทดลองกับ โทรศัพท์มือถือยี่ห้อ ซีเมนส์ ซี 35 ตามรูปที่ 4.1



รูปที่4.1 การกำหนดค่าสำหรับโทรศัพท์มือถือ

จากนั้นก็ทำการทดลองโดยการพิมพ์คำสั่ง เอที คอมมานด์ลงไปโปรแกรมซึ่งถ้า แสดงว่า โอเค นั่นคือ โทรศัพท์มือถือสามารถรองรับคำสั่งได้ แต่ถ้าขึ้นคำว่า เอเอเรอ์ แสดงว่า โทรศัพท์มือถือนี้ ไม่รองรับคำสั่งดังกล่าวตามรูปที่4.2

```

x - HyperTerminal
File Edit View Call Transfer Help
[at+cmgf=1
ERROR
at+cmgf=0
OK
at+csms=0
+CSMS: 1,1,1
OK
-
Connected 0:05:11 Auto detect 19200 8-N-1 SCROLL CAPS NUM Pause First echo

```

รูปที่4.2 คำสั่งที่ใช้ควบคุม โทรศัพท์มือถือ

จากรูป4.2 จะเห็นได้ว่า โทรศัพท์มือถือรุ่นนี้ รองรับได้แก่ ฟิสิกส์ โหมด เพราะ AT+ CMGF=0 ซึ่งหมายถึงการเลือก ฟิสิกส์โหมด ส่วน AT+ CSMS=0 นั้นหมายถึงเป็นการเลือกข้อความในโหมดที่ 1 ซึ่งดูจากตัวหลังสุด

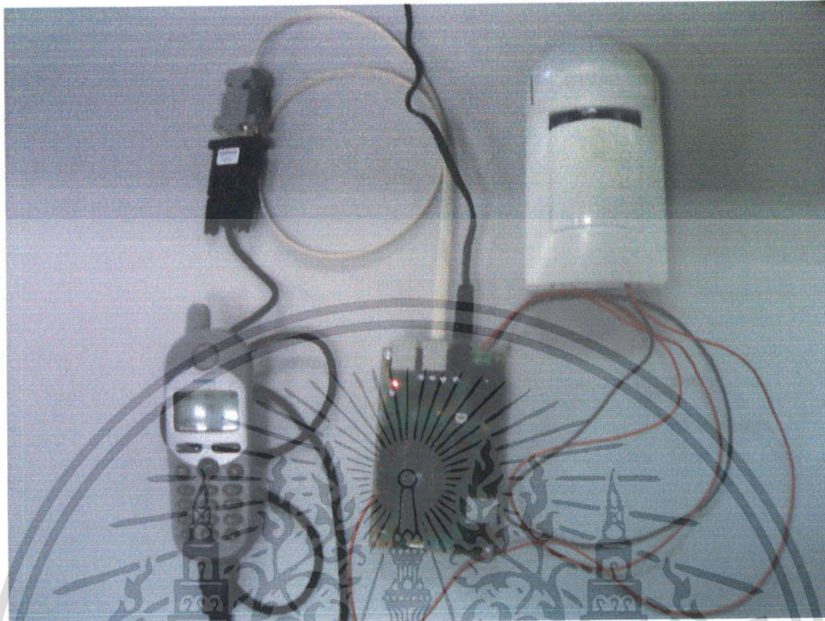
จากนั้นเราจะทำการเชื่อมต่อระหว่างโทรศัพท์มือถือกับไมโครคอนโทรลเลอร์โดยผ่าน ซีเรียลพอร์ต



รูปที่4.3 การเชื่อมต่อ โทรศัพท์มือถือกับระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการต่อเซนเซอร์เข้ากับขาอินเทอร์รับของไมโครคอนโทรลเลอร์ ตามรูป4.4



รูปที่4.4 วงจรสมบูรณ์ของโครงการ

จากนั้นทำการเบิร์นไมโครคอนโทรลเลอร์ที่โปรแกรมลงไปแล้วจะได้ผลตามรูป4.5



รูปที่4.5 ผลที่ได้จากโทรศัพท์มือถืออีกฉบับ

นั่นก็คือ ระบบจะทำการเตือนภัยด้วยข้อความ เอสเอ็มเอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและบทวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลองพบว่า การส่งข้อความเตือนภัยนั้น ให้ผลที่ถูกต้องชัดเจน และน่าเชื่อถือ ประสิทธิภาพของระบบนี้ ได้ทดลองโดยใช้ไมโครคอนโทรลเลอร์ตระกูล AT89C51 ซึ่งไมโครคอนโทรลเลอร์จะติดต่อกับโทรศัพท์มือถือ โดยสายดาต้าถึงชนิดซีเรียลพอร์ท โดยไมโครคอนโทรลเลอร์จะทำการส่งรหัสพีดียูไปยังโทรศัพท์มือถือภาคส่ง โดยที่โทรศัพท์มือถือภาคส่งจะทำการถอดรหัสดีพียูกลับไปเป็นข้อความ แล้วทำการส่งข้อความออกไป นอกจากนี้ยังสามารถที่จะพัฒนาเป็นระบบส่งงานอัตโนมัติได้

5.2 ปัญหาในการดำเนินการ

ถ้าโทรศัพท์อยู่ในสถานที่ที่อับสัญญาณการส่งข้อความเอสเอ็มเอสอาจมีปัญหา ส่งไม่ได้ หรือการส่งเกิดการล่าช้าได้

เอกสารอ้างอิง

- [1] เอกสารเซมิคอนดักเตอร์ฉบับที่ 264 เรื่องระบบเตือนภัยทางโทรศัพท์มือถือ
- [2] วรพจน์ กรแก้ววัฒนกุล , ชัยวัฒน์ ลิ้มพรจิตรวิไล “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS51”
- [3] www.gunkamoy.com
- [4] www.ett.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมที่ใช้งาน

```

#include "reg52.h"
#include "absacc.h"
#include "function.c"
#include "LCD.C"

unsigned char Text[42];
unsigned char idata Pdu[29];
unsigned char idata QBuf[60];
bit Bits;
float TCount;
unsigned int Index;
unsigned char Ascii,Count,i,j,k,Temp,Temp2;
unsigned char idata GBuf[36],l;
unsigned char Ch;
sbit gate = P3^4;
sbit Load1 = P2^0;
sbit Load3 = P2^2;
sbit Load4 = P2^3;

// ----- //
unsigned char HexToAsc(unsigned char X)
{
unsigned char Temp;

Temp = X;

if(Temp > 9){Temp = Temp + 0x37;}
else {Temp = Temp + 0x30;}

return(Temp);
}

// ----- //

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void FindText(unsigned char Count)
{
    unsigned char i;
    for(i=0;i<Count;i++)
    {
        QBuf[i*2] = HexToAsc(Pdu[i]>>4);
        QBuf[(i*2)+1] = HexToAsc(Pdu[i]&0x0F);
    }
}

unsigned char Convert(unsigned char j)
{
    if(j>0x40) { j = j - 55; }else j = j - 48;
    return(j);
}

unsigned char ASCHex(unsigned char X,Y)
{
    unsigned char i;
    i = (Convert(X)<<4);
    i = i+ Convert(Y);
    return(i);
}

void FindPDU(unsigned char Count)
{
    unsigned char i,k,j,Temp,Temp2;

    j=0;
    for(i=0;i<(Count-1);i++)
    {
        k = i % 7;
        if(i==0){k=0;QBuf[30] = QBuf[0] & 0x7F;j++;}
        else if(k==0){QBuf[j+30]=QBuf[i] & 0x7F;j++;}
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Temp2 = QBuf[i] >> (7-k);
Temp = (QBuf[i+1] << (k+1)) & 0x7F;
QBuf[j+30] = Temp | Temp2;
j++;

}

//for (i=0;i<(Count);i++) Tx_Byte(QBuf[i+30]);
}

```

```

void AIS_PDU(unsigned char Count)

```

```

{ unsigned char i;

```

```

for(i=0;i<(Count/2);i++)

```

```

{

```

```

QBuf[i+30] = QBuf[(i*2)+1];

```

```

}

```

```

}

```

```

// ----- //

```

```

// - Input Text[..]

```

```

// - Output ASC[..]

```

```

void TextToPDU(unsigned char Count)

```

```

{ TCount = (Count*7)/8.0;

```

```

Index = TCount*10;

```

```

if((Index%10)>0){Index= Index+10;}

```

```

Count = Index / 10;

```

```

i=0;

```

```

for(j=0;j<Count;j++)

```

```

{ k = j % 7;

```

```

if((j!= 0)&&(k==0)){i++;}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Temp2 = Text[i] >> k;
Temp = (Text[i+1] << (7-k));
Pdu[j] = Temp | Temp2;
i++;
}
FindText(Count);

```

```

}
//-----//
main()
{
unsigned char i,Data;
unsigned char l;
Init_LCD();
Light = 1;
Wr_String(" ..STANDBY.. ",1);
Tx_Byte(0x0D);Tx_Byte(0x0D);
Init_Serial(192);Tx_String("ATH");
Tx_Byte(0x0D);
Delay(1500);
while(1)
{
Init_Serial(192);
gate = 1;
RI = 0;
if(INT0 == 0)
{ Wr_String(" Fire Alert ",2);
Tx_String("ATH");
Tx_Byte(0x0D);Delay(500);
Tx_String("AT+CMGS=22");//12+ 10 Lenght Message
Tx_Byte(0x0D);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while(Rx_Byte()=='>'){
            Delay(500);

Tx_String("0011000A8180766506390000AA0AC6B4BC0C0AB2CB723A");
    Tx_Byte(0x1A);
        Delay(3000);
    }

Tx_String("AT+CMGL=0"); Tx_Byte(0x0D); // Command Check New

```

Message

```

while(Rx_Byte() != 0x0A){
if(Rx_Byte() == 'O')
{
}
else
{QBuf[38]='a';
while(Rx_Byte() != 0x0A){ }
for(i=0;i<21;i++)
{
Rx_Byte();
}
i = 0;
QBuf[i] = Rx_Byte();
while(QBuf[i] != 0x0D)
{ i++;
QBuf[i] = Rx_Byte();
}
}
Wr_String("Tel = ",1);
Goto_XY(1,7);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

QBuf[0] = '0';
Wr_Char((QBuf[0]));
Wr_Char((QBuf[2]));
Wr_Char('-');
Wr_Char((QBuf[1]));
Wr_Char((QBuf[4]));
Wr_Char((QBuf[3]));
Wr_Char((QBuf[6]));
Wr_Char((QBuf[5]));
Wr_Char((QBuf[8]));
Wr_Char((QBuf[7]));
Wr_Char((QBuf[9]));

//if((QBuf[0]=='0')&&(QBuf[2]=='8')&&(QBuf[1]=='3')&&(QBuf[4]=='1')&&(QBuf[3]=='1')&
&(QBuf[6]=='5')&&(QBuf[5]=='1')&&(QBuf[8]=='5')&&(QBuf[7]=='2')&&(QBuf[9]=='7'))
{
    Count = ASCHex(QBuf[29],QBuf[30]);
    Index = ASCHex(QBuf[31],QBuf[32]);
    if(Index != 0)
    {
        TCount = (Count*7)/8.0;
        Index = TCount*10;
        if((Index%10)>0){Index= Index+10;
        Index = (Index/10)*2;
        }else {
            Index = (Count) * 2;
        }

        for(i=0;i<Index;i+=2)
        {
            QBuf[i/2] = ASCHex(QBuf[i+31],QBuf[i+32]);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

#include "string.h"

sbit Row0      =    P1^0;
sbit Row1      =    P1^1;
sbit Row2      =    P1^2;
sbit Row3      =    P1^3;

unsigned char data BufDisp[7];

void Wait(unsigned int x)
{unsigned int i;
  for(i=0;i<x;i++) {}

}

sbit Piezo = P3^6;
void Sound(void)
{unsigned char i;
  for (i=0;i<50;i++)
  {
    Piezo = ~Piezo;
    Wait(50);
  }
}

sbit Piezo2 = P2^5;
void SoundSmart(void)
{unsigned char i;
  for (i=0;i<50;i++)
  {
    Piezo2 = ~Piezo2;
    Wait(50);
  }
}

/* ----- Scan Key Sw. ----- */

unsigned char Scan_Key(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P1    =    0xEF;
if (Row0 == 0) {return(11);}
if (Row1 == 0) {return(9);}
if (Row2 == 0) {return(6);}
if (Row3 == 0) {return(3);}

```

```

P1    =    0xDF;
if (Row0 == 0) {return(0);}
if (Row1 == 0) {return(8);}
if (Row2 == 0) {return(5);}
if (Row3 == 0) {return(2);}

```

```

P1    =    0xBF;
if (Row0 == 0) {return(10);}
if (Row1 == 0) {return(7);}
if (Row2 == 0) {return(4);}
if (Row3 == 0) {return(1);}
return(0xff);

```

```

P1    =    0x7F;
if (Row0 == 0) {return(15);}
if (Row1 == 0) {return(14);}
if (Row2 == 0) {return(13);}
if (Row3 == 0) {return(12);}

return(0xff);

```

```

}

```

```

/* --- Delay ----- */

```

```

void Delay(unsigned int x )

```

```

{ unsigned int  i,j;

```

```

  for (i=0;i<x;i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (j=0;j<115;j++) { }
    }
}

void Key_Off(void)
{
    while (Scan_Key() != 0xFF) { }
    Delay(50);
}

unsigned char KeyPress(void)
{unsigned char Key;
    while (Scan_Key() == 0xFF) { }
    Key = Scan_Key();
    Key_Off();
    return(Key);
}

code unsigned char BufSEG[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,
                                0x82,0xF8,0x80,0x90,0xFF,0xC6};

void Display (void)
{
    P2 = 0xFF;          /* Display off */

    P0 = BufSEG[BufDisp[3]];

    P2 = 0xEF;          /* Scan 7 - Segment */
    Delay(4);

    P2 = 0xFF;          /* Display off */

    P0 = BufSEG[BufDisp[2]];

    P2 = 0xDF;          /* Scan 7 - Segment */
    Delay(4);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        P2 = 0xFF;          /* Display off */

P0 = BufSEG[BufDisp[1]];

P2 = 0xBF;          /* Scan 7 - Segment */

    Delay(4);

        P2 = 0xFF;          /* Display off */

P0 = BufSEG[BufDisp[0]];

P2 = 0x7F; /* Scan 7 - Segment */

    Delay(4);

        P2 = 0xFF;

}

void CountToDisp( unsigned long Count)
{
    BufDisp[6] = Count / 1000000;
    if(BufDisp[6] == 0) BufDisp[6] = 10;

    Count = Count % 1000000;

    BufDisp[5] = Count / 100000;
    if((BufDisp[6] == 10) && (BufDisp[5] == 0)) BufDisp[5] = 10;

    Count = Count % 100000;

    BufDisp[4] = Count / 10000;
    if((BufDisp[5] == 10)&& (BufDisp[4] == 0)) BufDisp[4] = 10;

    Count = Count % 10000;

    BufDisp[3] = Count / 1000;
    if((BufDisp[4] == 10) && (BufDisp[3] == 0)) BufDisp[3] = 10;

    Count = Count % 1000;

    BufDisp[2] = Count / 100;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if((BufDisp[3] == 10)&& (BufDisp[2] == 0)) BufDisp[2] = 10;

        Count    =    Count % 100;
        BufDisp[1] =    Count / 10;
        if((BufDisp[2] == 10) && (BufDisp[1] == 0)) BufDisp[1] = 10;

        BufDisp[0] =    Count % 10;

    }

    void ShowVal(unsigned int Num)
    {
        CountToDisp(Num);
        Display();
    }

    void Tx_Byte(unsigned char Data)
    {
        TI    =    1;
        SBUF  =    Data;
        TI    =    0;
        while (!TI) {}
        TI    =    0;
    }

    unsigned char Rx_Byte(void)
    {unsigned char Data;
        while (!RI) {}
        RI    =    0;
        Data  =    SBUF;
        return(Data);
    }

    void Init_Serial(unsigned char br)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TMOD = (TMOD & 0x0F)+0x20;
    TH1 = 0xFD;
    TR1 = 1;
    EA = 1;
    if (br == 48) TH1 = 0xFA;
    if (br == 96) TH1 = 0xFD;
    if (br == 192) {PCON = 0x80;} else {PCON = 0x00;}
}

void Tx_String(unsigned char *dptr)
{unsigned char i,Count;
  Count = strlen(dptr);
  for(i=0;i<Count;i++)
  {
    Tx_Byte(*dptr);
    dptr++;
  }
}

unsigned char _TH0,_TL0;
void Init_Timer0(unsigned int Freq)
{unsigned int Count;

  Count = 65536 - (921600 / Freq);
  TH0 = Count >> 8;
  TL0 = Count & 0xFF;
  _TH0 = TH0;
  _TL0 = TL0;
  TMOD = (TMOD & 0xF0)+0x01;
  TR0 = 1;
  EA = 1;
}

```

```
void Timer0_IntOn(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    ET0 = 1;
}

void Timer0_IntOff(void)
{
    ET0 = 0;
}

void Timer0_On(void)
{
    TR0 = 1;
}

void Timer0_Off(void)
{
    TR0 = 0;
}

void Serial_IntOn(void)
{
    ES = 1;
}

void Serial_IntOff(void)
{
    ES = 0;
}

sbit  EN_LCD    =    P0^3;
sbit  RS_LCD    =    P0^2;
sbit  LIGHT     =    P0^1;

```

bit Light;
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Wr_Register(unsigned char Data)
```

```
{unsigned char Temp;
```

```
Temp= Data;
```

```
Temp = Temp | 0x0F;
```

```
P0 = Temp;
```

```
if (Light) { LIGHT = 0;}
```

```
RS_LCD = 0;
```

```
EN_LCD = 0;
```

```
Delay(1);
```

```
EN_LCD = 1;
```

```
Temp = Data;
```

```
Temp = Temp << 4;
```

```
Temp = Temp | 0x0F;
```

```
P0 = Temp;
```

```
if (Light) { LIGHT = 0;}
```

```
RS_LCD = 0;
```

```
EN_LCD = 0;
```

```
Delay(1);
```

```
EN_LCD = 1;
```

```
}
```

```
void Wr_Char(unsigned char Data)
```

```
{unsigned char Temp;
```

```
Temp= Data;
```

```
Temp = Temp | 0x0F;
```

```
P0 = Temp;
```

```
if (Light) { LIGHT = 0;}
```

```
RS_LCD = 1;
```

```
EN_LCD = 0;
```

```
Delay(1);
```

```
EN_LCD = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Temp = Data;
Temp = Temp << 4;
Temp = Temp | 0x0F;
P0 = Temp;
if(Light) { LIGHT = 0;}
RS_LCD = 1;
EN_LCD = 0;
Delay(1);
EN_LCD = 1;
}
void Init_LCD(void)
{
Delay(200);
Wr_Register(0x33);
Wr_Register(0x32);
Wr_Register(0x28);
Wr_Register(0x0C);
Wr_Register(0x06);
Wr_Register(0x01);
Delay(100);
}
void Goto_XY(unsigned char x,y)
{
if(x == 1)
{
Wr_Register(0x80+(y-1));
}
if(x == 2)
{
Wr_Register(0xC0+(y-1));

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if(x == 3)
    {
        Wr_Register(0x94+(y-1));
    }
    if(x == 4)
    {
        Wr_Register(0xD4+(y-1));
    }
}

void Wr_Page(unsigned char *dptr)
{unsigned char i;
  Goto_XY(1,1);
  for (i=0;i<20;i++)
  { Wr_Char(*dptr);
    dptr++; }
  Goto_XY(2,1);
  for (i=0;i<20;i++)
  { Wr_Char(*dptr);
    dptr++; }
}

void Wr_String(unsigned char *dptr,Row)
{unsigned char i,Count;
  Count = strlen(dptr);
  Goto_XY(Row,1);
  for (i=0;i<Count;i++)
  { Wr_Char(*dptr);
    dptr++;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Cursor_Off(void)
{
    Wr_Register(0x0C);
}

void Cursor_On(void)
{
    Wr_Register(0x0F);
}

void SlideNum(void)
{
    BufDisp[6] = BufDisp[5];
    BufDisp[5] = BufDisp[4];
    BufDisp[4] = BufDisp[3];
    BufDisp[3] = BufDisp[2];
    BufDisp[2] = BufDisp[1];
    BufDisp[1] = BufDisp[0];
}

void CountToLCD( unsigned long Count)
{
    BufDisp[6] = (Count / 1000000)+0x30;
    if(BufDisp[6] == 0x30) BufDisp[0] = 0x20;

    Count = Count % 1000000;
    BufDisp[5] = (Count / 100000)+0x30;
    if((BufDisp[5] == 0x30) && (BufDisp[6] == 0x20)) BufDisp[5] =
0x20;

    Count = Count % 100000;
    BufDisp[4] = (Count / 10000)+0x30;
    if((BufDisp[4] == 0x30) && (BufDisp[5] == 0x20)) BufDisp[4] =
0x20;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Count = Count % 10000;
BufDisp[3] = (Count / 1000)+0x30;
if((BufDisp[3] == 0x30) && (BufDisp[4] == 0x20)) BufDisp[3] = 0x20;

```

```

Count = Count % 1000;
BufDisp[2] = (Count / 100)+0x30;
if((BufDisp[2] == 0x30) && (BufDisp[3] == 0x20)) BufDisp[2] =
0x20;

```

```

Count = Count % 100;
BufDisp[1] = (Count / 10)+0x30;
if((BufDisp[1] == 0x30) && (BufDisp[2] == 0x20)) BufDisp[1] =
0x20;

```

```

Count = Count % 10;
BufDisp[0] = Count+0x30;
}
void ShowNum(unsigned long Count,unsigned char Index)
{unsigned char i;
  CountToLCD(Count);
  for(i=0;i<Index;i++)
  {
    Wr_Char(BufDisp[Index-(i+1)]);
  }
}

```

```

unsigned long GetNum(unsigned char Count,X,Y)
{unsigned char i,j,Key;
  unsigned long Sum;
  for(i=0;i<Count;i++){BufDisp[i] = 0x20;}

```

```

i = 0;
while(Key != 11)
  {Goto_XY(X,Y+(Count-1));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Cursor_On();
Key = KeyPress();
Sound();

if((Key < 10)&&(i<Count))
{
    SlideNum();
    BufDisp[0] = Key;

    i++;
}
if(Key == 10)
{
    i = 0;
    for(j=0;j<Count;j++){BufDisp[j] = 0x20;}
}
Cursor_Off();
Goto_XY(X,Y);
for(j=0;j<Count;j++)
{
    if(BufDisp[Count-(j+1)] != 0x20){Wr_Char(BufDisp[Count-(j+1)]+0x30);}
    else
{Wr_Char(BufDisp[Count-(j+1)]);}
}

Key_Off();
}

Sum = 0;
for(i=0;i<Count;i++)
{
    if(BufDisp[Count-(i+1)]==0x20){BufDisp[Count-(i+1)]=0 ;}
    Sum = (Sum*10) + (BufDisp[Count-(i+1)]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Cursor_Off());  
return(Sum);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้