สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หน้าจอแสดงผลข้อความจาก SMS

SMS MESSAGE DISPLAY

โดย

นาย กิตติพงศ์ เด่นแพทย์ชรางกูร เลขประจำตัว 47010044

นาย คุณนิธิ สมพงศ์ เลขประจำตัว 47010080

นาย จรัสพงษ์ งามเหมาะ เลขประจำตัว 47010084

อาจารย์ที่ปรึกษา

อ.ชินภัทร นันทจิวากรชัย

ปส.
ก1673ห
2550

เลขหมู่.....
เลขทะเบียน 82474
วัน,เดือน,ปี 11 ก.ค. 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำภาคเรียนที่ 2 ปีการศึกษา 2550

b.11946805
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานภาษาไทย หน้าจอแสดงผลข้อความจาก SMS

ชื่อโครงงานภาษาอังกฤษ SMS DISPLAY

จัดทำโดย

นายกิตติพงศ์ เด่นแพทย์ชรางกูร เลขประจำตัว 47010044

นายคุณนิธิ สมพงษ์ เลขประจำตัว 47010080

นาย จรัสพงษ์ งามเหมาะ เลขประจำตัว 47010084

อาจารย์ที่ปรึกษา

อาจารย์ ชินภัทร นันทจิวากรชัย

รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ชื่อ.....

อาจารย์ที่ปรึกษา

(.....)

วันที่ ..... / ..... / ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

รายงานนี้จัดทำขึ้นเพื่อให้ได้ความรู้ในการทำงานของการทำงานของไมโครคอนโทรลเลอร์ซึ่งในปัจจุบันนี้มีการใช้งานไมโครคอนโทรลเลอร์อย่างกว้างขวาง โดยในโครงงานนี้ใช้ MCS-51 และการทำงานของระบบ SMS โดยที่สามารถนำความรู้นี้ไปประยุกต์ใช้ได้อย่างกว้างขวางมากมาย โดยผู้ที่ได้ศึกษารายงานนี้ก็จะรู้ถึงการถอดรหัส SMS การแปลงเลขฐาน 16 เพื่อถอดรหัส SMS และการรู้จักกับ MCS-51 ซึ่งจะเป็นตัวที่เก็บข้อมูลและประมวลผลการทำงาน

นาย กิตติพงศ์ เด่นแพทย์ชรางกูร
นาย คุณนิธิ สมพงศ์
นาย จรัสพงษ์ งามเหมาะ

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอแสดงผลข้อความจาก sms

นาย กิติพงศ์ เด่นแพทย์ชรางกูร รหัส 47010044
นาย คุณนิธิ สมพงศ์ รหัส 47010080
นาย จรัสพงษ์ งามเหมาะ รหัส 47010084

อ. ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา

ปีการศึกษา 2550

บทคัดย่อ

โครงงานหน้าจอแสดงผลข้อความจาก SMS นี้ใช้ควบคุมโดยไมโครคอนโทรลเลอร์ MCS-51 เป็นตัวประมวลผล โดยจะแบ่งเป็น 3 หลักการ คือ ส่วนแรกเป็นการเชื่อมต่อกับโทรศัพท์มือถือซึ่งจะใช้รับ SMS จากมือถือ ส่วนที่สองเป็นการถอดรหัส PDU ซึ่งเป็นข้อความที่รับได้จากโทรศัพท์มือถือให้เป็น TEXT และ ส่วนที่สาม เป็นส่วนแสดงผลทางหน้าจอ ซึ่งนำข้อความที่ที่เป็น TEXT จากส่วนที่ 2 มาขึ้นจอ LED dot-matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMS MESSAGE DISPLAY

Mr. Kittipong Denphatcharangkul ID 47010044
Mr. Kunniti Sompong ID 47010080
Mr. Jaratpong Ngammoh ID 47010084
Mr. Shinnapat Nantajiwagornchai Advisor
Education Year 2550

ABSTRACT

This project has implemented SMS MESSSAGE DISPLAY, controlled by microcontroller MCS-51 for processing. The project is composed of 3 major parts. The first part is connecting to mobile phone by get SMS from mobile phone. The second part is decoding PDU code, which are message form mobile phone, to TEXT. The third part is showing message to LED dot-matrix display, the message are TEXT from the second part.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	1
Abstract	2
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล	3
1.2 ขอบข่ายของงาน	4
1.3 แนวทางการดำเนินงาน	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ไอซีไมโครคอนโทรลเลอร์ AT89C51ED2	5
2.1.1 หน้าที่และการใช้งานของไมโครคอนโทรลเลอร์	5
2.1.2 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	8
2.2 การแสดงผลแบบสแกน	17
2.3 หลักการรับส่ง SMS	18
2.4 PDU MODE	20
2.4.1 การรับข้อมูล SMS	20
2.4.2 การถอดรหัสตัวอักษรชนิด 7 บิต	22
2.5 AT-COMMAND	22
2.5.1 AT COMMAND ที่เกี่ยวข้องกับการรับ-ส่ง SMS	23
2.6 SERIAL COMMUNICATION	25
2.7 รหัส ASCII	26
2.8 ไอซี 74HC595	27
บทที่ 3 การออกแบบวงจร	30
3.1 วงจรแสดงผล SMS แบบ LED	30
3.1.1 วงจรส่วนที่รับส่งข้อความจากมือถือ	31
3.1.2 วงจรส่วนที่แสดงผลทาง LED	33
บทที่ 4 การทดสอบวงจรและโปรแกรม	35
4.1 การทดสอบส่วนของโปรแกรม	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 ทดสอบส่วนที่ส่ง AT_COMMAND 35
4.1.2 การทดสอบการถอดรหัส PDU 37
4.1.3 การทดสอบการแสดงผลทาง LED 38
4.1.4 การทดสอบโดยการเชื่อมต่อกับมือถือผ่านอินฟาเรด 39
4.2 การทดสอบกับการต่อวงจรของจริง 40
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 44
5.1 สรุปผลการทดลอง 44
5.2 สิ่งที่ได้รับจากการทำโครงงาน 44
5.3 ข้อเสนอแนะในการทำโครงงาน 45
บรรณานุกรม 46
ภาคผนวก 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 ภาพรวมของโครงงาน 3
รูปที่ 2.1 แสดงขาต่างของไมโครคอนโทรลเลอร์ AT89C51ED2 5
รูปที่ 2.2 แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2 6
รูปที่ 2.3 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0 9
รูปที่ 2.4 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 1 9
รูปที่ 2.5 แสดงสัญญาณการส่งข้อมูลในโหมด 2 10
รูปที่ 2.6 อัตรารับส่งโหมด 0 14
รูปที่ 2.7 อัตรารับส่งโหมด 2 14
รูปที่ 2.8 การแสดงผลตัวอักษร R โดยการแสกน ซึ่งจะแสดงผลทีละหลักไปเรื่อยๆ จนครบ 5 columns 18
รูปที่ 2.9 การแสดงผลตัวอักษร R เมื่อแสกนด้วยความเร็วสูง 18
รูปที่ 2.10 หลักการรับ-ส่ง SMS 19
รูปที่ 2.11 ข้อมูลใน SMS ในรูปแบบ PDU MODE 20
รูปที่ 2.12 ตัวอย่างคำสั่งพื้นฐานและ LIST ข้อความใน STO SENT 24
รูปที่ 2.13 ตัวอย่างการอ่านข้อความ 24
รูปที่ 2.14 ขั้นที่ 1 ของการทำงาน 74HC595 27
รูปที่ 2.15 ขั้นที่ 2 ของการทำงาน 74HC595 28
รูปที่ 2.16 ขั้นที่ 3 ของการทำงาน 74HC595 28
รูปที่ 2.17 ขั้นที่ 4 ของการทำงาน 74HC595 29
รูปที่ 3.1 วงจรแสดงผล SMS 30
รูปที่ 3.2 วงจรส่วนที่รับส่งข้อความจากมือถือ 31
รูปที่ 3.3 SIEMENS C35 32
รูปที่ 3.4 ตัวเชื่อมระหว่าง C35 กับ MCS51 32
รูปที่ 3.5 แบบการแสดงผลแบบ 16*16 33
รูปที่ 3.6 แผนผังการทำงาน 35
รูปที่ 4.1 แสดงผลการทดสอบโปรแกรมในการรับส่งข้อความ 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 การทดสอบการถอดรหัส PDU 37
รูปที่ 4.3 การทดลองการแสดงผลทาง LED 38
รูปที่ 4.4 การทดลองส่ง AT COMMAND ให้กับมือถือ 39
รูปที่ 4.5 การแสดงผล LED จากการทดสอบจริง 40
รูปที่ 4.6 การแสดงผลทาง LCD โดยแสดงส่วนที่เป็นตัวแรกหลังจากการส่ง AT COMMAND ให้แสดง SMS 41
รูปที่ 4.7 การแสดงผลทาง LCD โดยแสดงส่วนที่เป็นรหัส PDU 41
รูปที่ 4.8 หลังจากที่ถอดรหัส PDU 41
รูปที่ 4.9 ภาพโดยรวมการแสดงผล LED และการแสดงผลทางจอ LCD 42
รูปที่ 4.10 ประกอบวงจรลงกล่อง 43
รูปที่ 4.11 ทีเด็ดของวงจรทำแล้วลด noise ได้อย่างดี 43
รูปที่ 4.10 ทีมงานและผลงาน 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register) 11
ตาราง 2.2 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน 11
ตารางที่ 2.3 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล 13
ตารางที่ 2.4 แสดงตัวอย่างอัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้บอดเรทค่ามาตรฐานต่างๆ 16
ตารางที่ 2.5 ตัวอย่างเลขฐานสองที่จะแสดงผลทางจอ LED 17
ตารางที่ 2.6 ความหมายของรหัส SMS ที่ได้รับ 21
ตารางที่ 2.7 หน้าที่ของขาพินแต่ละขา 25
ตารางที่ 2.8 ตารางการแปลงรหัสASCII 26

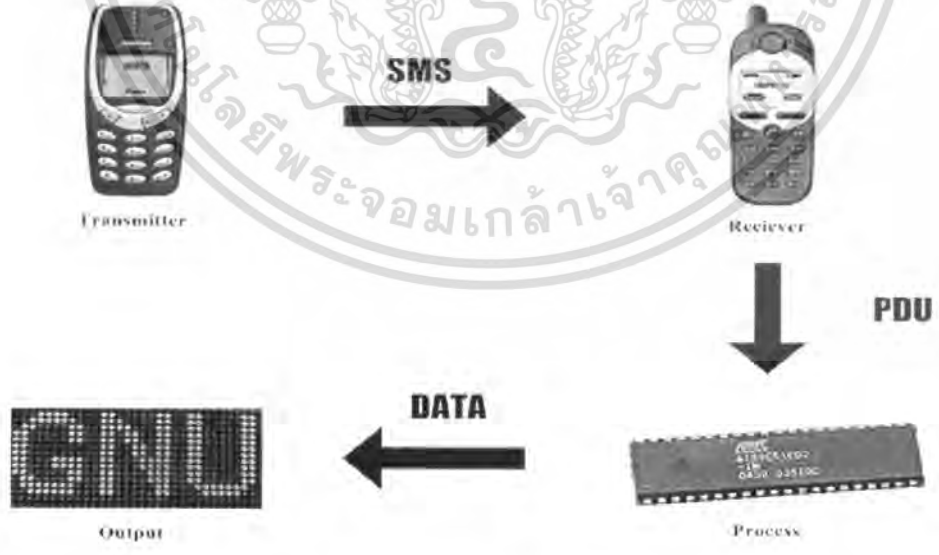
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

โทรศัพท์มือถือ เป็นอุปกรณ์ที่เข้ามามีบทบาทในชีวิตประจำวันของคนส่วนใหญ่อย่างมาก เนื่องจากความสะดวก รวดเร็วมีประสิทธิภาพสูง ทำให้การติดต่อสื่อสารเป็นไปอย่างง่ายดาย และแพร่กระจายไปได้แทบทั่วทุกมุมโลก เนื่องจากในปัจจุบันแทบจะทุกคนมีโทรศัพท์มือถือติดตัวเพื่อสะดวกในการติดต่อสื่อสารกับคนอื่น และสิ่งที่คนยังนิยมทำกับมือถือคือการส่งข้อความสั้น (SMS : Short Message Service) ซึ่งการส่งนี้สามารถส่งไปได้ในระยะทางที่ไกลและไม่มีปัญหาสัญญาณรบกวน

ดังนั้นจึงได้มีการทำโครงงานที่เกี่ยวข้อง กับ SMS เพื่อที่จะนำไปประยุกต์ใช้การแสดงข้อความจากมือถือ เพราะในบางครั้งเราอาจจะอยากบอกอะไรในระยะทางไกลๆออกไปอาจจะส่ง SMS เพื่อแสดงผลหน้าจอให้เพื่อแสดงออกแก่คนหมู่มากให้ได้รู้ หรืออาจจะไว้ประกาศตามหาคนแจ้งการเตือนภัย หรือความต้องการของตนเองก็ได้ โดยการทำงานคร่าวๆจะเป็นดังรูป



รูปที่ 1.1 ภาพรวมของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขอบข่ายของงาน

- เขียนโปรแกรมเพื่อศึกษาการส่งคำสั่ง AT_COMMAND เพื่อที่จะให้มือถือแสดงผล SMS
- เขียนโปรแกรมเพื่อรับข้อความรหัส PDU ที่ได้รับจากมือถือเพื่อที่จะถอดรหัส
- เขียนโปรแกรมเพื่อทำการศึกษาวงจรจอแสดงผล LED Dot matrix สำหรับไมโครคอนโทรลเลอร์ MCS-51
- ออกแบบวงจรจอแสดงผล LED Dot matrix สำหรับไมโครคอนโทรลเลอร์ MCS-51
- สร้างวงจรจอแสดงผล LED Dot matrix โดยที่เชื่อมต่อกับมือถือทางพอร์ตอนุกรม
- ทำการทดสอบจอแสดงผล LED Dot matrix โดยรับค่า SMS แล้วแสดงผล

1.3 แนวทางการดำเนินงาน

- ศึกษาการเขียนโปรแกรมภาษา C ในการเขียโปรแกรมสั่งการ Microcontroller
- ศึกษาการต่อพอร์ตอนุกรมเพื่อการเชื่อมต่อกับมือถือ
- ศึกษาและทดสอบวงจรบอร์ดแสดงผล LED Dot matrix
- สร้างบอร์ดแสดงผลและทดสอบการทำงานการเชื่อมต่อกับมือถือ
- สร้างบอร์ดแสดงผลและทดสอบการทำงานของบอร์ดแสดงผล LED Dot matrix
- ทดสอบการทำงานของ ไมโครคอนโทรลเลอร์ โดยรับ SMS และแสดงผลขึ้นจอ LED Dot matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

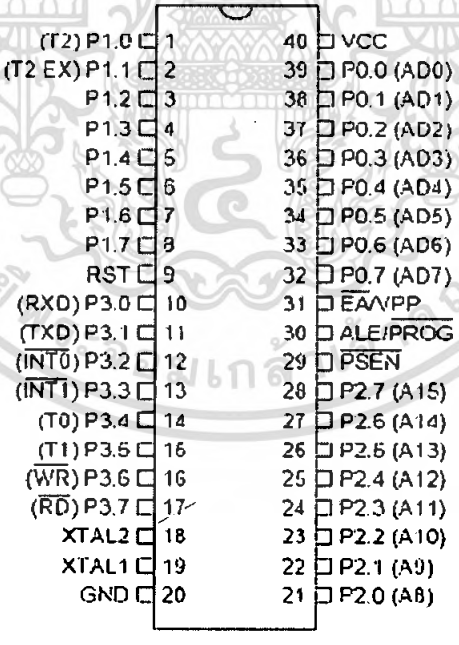
2.1 ไอซีไมโครคอนโทรลเลอร์ AT89C51ED2

AT89C51ED2 จะแตกต่างกับ AT89C52 ตรงที่มีหน่วยความจำโปรแกรม Flash Memory ที่มากกว่า โดย AT89C52 จะมีขนาด 8 KB แต่ AT89C51ED2 จะมีถึง 64 KB

2.1.1 หน้าที่และการใช้งานของไมโครคอนโทรลเลอร์

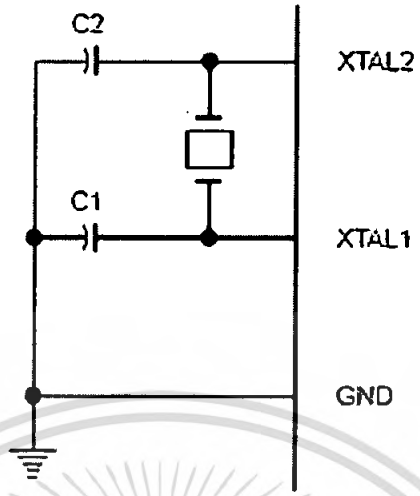
Vcc สำหรับต่อไฟ +5 Vdc เพื่อเลี้ยงไอซี

Vss (GND) สำหรับต่อลงกราวด์



รูปที่ 2.1 แสดงขาต่างของไมโครคอนโทรลเลอร์ AT89C51ED2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2

XTAL1 เป็นอินพุตของภาคขยายสัญญาณแบบอินเวอร์ส ของวงจรผลิตสัญญาณนาฬิกา

XTAL2 เป็นเอาต์พุตของภาคขยายสัญญาณแบบอินเวอร์ส ของวงจรผลิตสัญญาณนาฬิกา

RST สัญญาณรีเซต ไมโครคอนโทรลเลอร์จะถูกรีเซตเมื่อสัญญาณที่ขานี้มีค่าเป็นลอจิก "1" นานไม่ต่ำกว่า 2 แมชชีนไซเกิล

ALE/PROG (Address Latch Enable) เป็นสัญญาณเอาต์พุต ซึ่งไมโครคอนโทรลเลอร์จะส่งออกไปเป็นสัญญาณพัลส์เพื่อแลทช์ค่าตำแหน่งไบต์ต่ำที่อยู่ที่พอร์ต P0ในขณะติดต่อกับหน่วยความจำภายนอกสัญญาณนี้จะส่งออกไปด้วยอัตราคงที่คือ 1/6 เท่าของความถี่สัญญาณนาฬิกา ซึ่งสามารถนำไปใช้กับหน่วยความจำข้อมูลภายนอก (External Data Memory) และสัญญาณนี้จะใช้เป็นอินพุตเพื่อควบคุมการโปรแกรม PROM ภายในตัวไมโครคอนโทรลเลอร์ด้วย

PSEN (Program Store Enable) เป็นเอาต์พุต สำหรับส่งสัญญาณสโตรป (พัลส์ต่ำ) เพื่ออ่านข้อมูลจากหน่วย ความจำโปรแกรมภายนอก (External Data Memory) เมื่อซีพียูอ่านรหัสคำสั่งจากหน่วยความจำภายนอกจะส่งสัญญาณ สโตรปออกมา 2 ครั้งใน1 แมชชีนไซเกิล แต่สัญญาณสโตปทั้ง 2 ครั้งจะถูกข้ามไปหากเป็นช่วงที่ซีพียูติดกับ External Data Memory

EA (External Access) เป็นสัญญาณอินพุตใช้สำหรับควบคุมให้ไมโครคอนโทรลเลอร์เลือกติดต่อกับโปรแกรมที่เก็บอยู่ในหน่วยความจำโปรแกรมภายในหรือโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกตัวไมโครคอน โทรลเลอร์หากให้ค่าลอจิก "1" ที่ขานี้จะเป็นการเลือกใช้หน่วยความจำ โปรแกรมภายในตัวไมโครคอนโทรลเลอร์ หากต้องการให้ซีพียูติดต่อกับหน่วยความจำภายนอกต้องต่อสัญญาณเข้ากับลอจิก "0"หรือ Vss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต 0 เป็นพอร์ตอินพุต/เอาต์พุต 2 ทาง ข นาด 8 บิต (P0.1-P0.7) เมื่อใช้เป็นเอาต์พุต สามารถต่อกับไอซี TTL ตระกูล LS ได้ 8 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก "1" ออกไปที่พอร์ตก่อนเพื่อทำให้ลอยซึ่งจะเป็นอิมพีแดนซ์สูง

พอร์ต 1 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีใช้เป็น
เอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก "1"
ออกไปที่พอร์ตนี้ก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเป็น "0" เข้ามาจะทำให้พอร์ต P1 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้พอร์ต P1 ยังทำหน้าที่รับ ตำแหน่งด้านต่ำในช่วงของการโปรแกรม EPROM และช่วงการตรวจสอบโปรแกรมใน ROM หรือ EPROM อีกด้วย และที่ขา P1.0 กับ P1.1 จะทำหน้าที่เป็น T2 และ T2EX อีกหน้าที่หนึ่ง

พอร์ต 2 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีใช้เป็น
เอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก "1"
ออกไปที่พอร์ตนี้ก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเข้ามาเป็น "0" จะทำให้พอร์ต P2 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้พอร์ต P1 ยังทำหน้าที่รับตำแหน่งไบต์สูง ในช่วงของการโปรแกรม EPROM และช่วงการตรวจสอบโปรแกรมใน ROM หรือ EPROM อีกด้วย

พอร์ต 3 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีใช้เป็น
เอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก "1"
ออกไปที่พอร์ตนี้ก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเข้ามาเป็น "0" จะทำให้พอร์ต P3 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้ พอร์ต P3 ยังทำหน้าที่เป็นสัญญาณอื่นๆอีกดังนี้

P3.0 (RXD) เป็นอินพุตของพอร์ตอนุกรม

P3.1 (TRX) เป็นเอาต์พุตของพอร์ตอนุกรม

P3.2 (INT0) สัญญาณอินเตอร์รัพจากภายนอกตัวที่ 0

P3.3 (INT1) สัญญาณอินเตอร์รัพจากภายนอกตัวที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.4 (T0) เป็นอินพุตจากภายนอกของตัวตั้งเวลา 0

P3.5 (T1) เป็นอินพุตจากภายนอกของตัวตั้งเวลา 1

P3.6 (WR) เป็นสัญญาณการเขียนข้อมูลออกไปภายนอก

P3.7 (RD) เป็นสัญญาณการอ่านข้อมูลจากภายนอกเข้ามา

เมื่อต้องการใช้งานพอร์ต P3 ให้ทำหน้าที่เป็นสัญญาณต่างๆจะต้องส่งค่าลอจิก "1" ออกไปแลทช์ที่พอร์ต P3 ก่อนเพื่อพูลอัพภายใน หากเราให้ค่าลอจิก "0" จะทำให้สัญญาณที่ขาต่างๆมีค่าเป็น "0" ตลอดเวลา

2.1.2 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

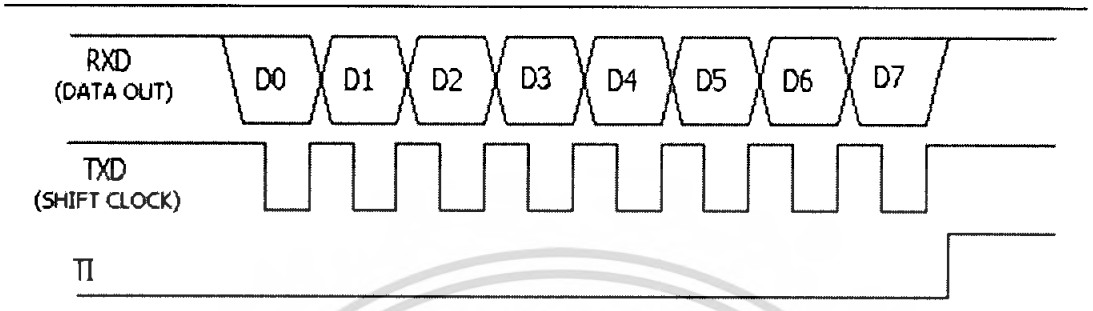
พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะมีโครงสร้างเป็นแบบฟูลดูเพล็ก ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้ โดยจะมีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรมโดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต เพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้ว จึงจะทำการเซตบิตโดยกำหนดค่าของแฟลก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไบต์ต่อไป ได้การรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้น โดยการกำหนดค่าของบิต REN ที่อยู่ในในรีจิสเตอร์SCON ให้มีค่าเป็นสถานะ "1" หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอก ก็จะทำการเลื่อนข้อมูลไปโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ที่รีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ "1" ซึ่งส่งผลให้เกิดการอินเตอร์รัพท์โปรแกรมขึ้น

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

โหมด 0: จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียว และจะไม่มีการส่งบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อน ข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพท์รีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุต ให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน

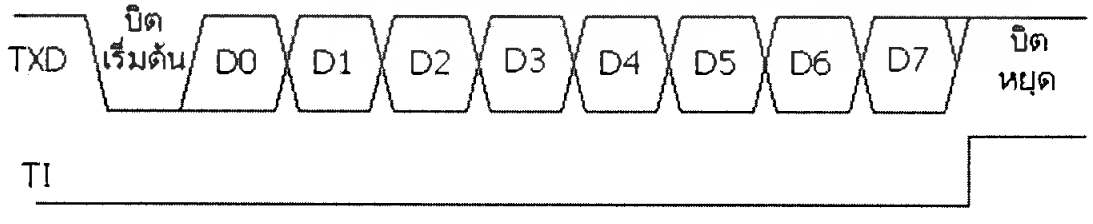
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะไอซีไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่นั้น มีจำนวนพอร์ตที่มากพอ และมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



รูปที่ 2.3 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

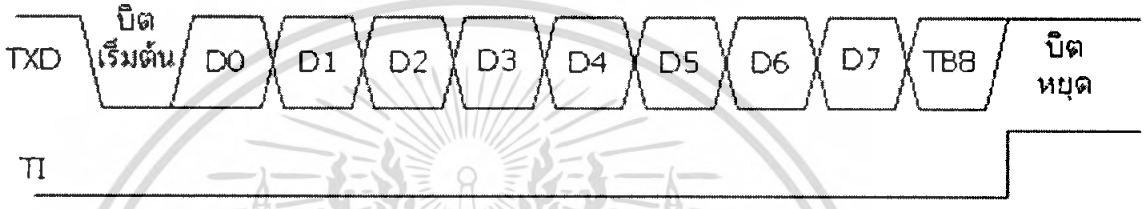
โหมด 1: จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver transmitters)สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ



รูปที่ 2.4 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2: จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters)ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วยบิตแรกจะเป็นบิตเริ่มต้น (Start bit ค่า 0), 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต(Stop bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit)



รูปที่ 2.5 แสดงสัญญาณการส่งข้อมูลในโหมด 2

โหมด 3: เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters)เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วใน การรับและส่งข้อมูลได้ตามต้องการ

UART (Universal Asynchronous Receiver Transmitters)

เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่ง และด้านรับ เป็นการส่งข้อมูล

โดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป (Start bit, Stop bit, Parity bit) เพื่อให้การรับ และการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้นการสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับและส่งข้อมูลทางอนุกรมมีอยู่ด้วยกัน 2 ระบบคือ

1.Single Processor System คือ ระบบการสื่อสารโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัว เชื่อมต่อหากัน

2.Multiprocessors System คือ ระบบการสื่อสารแบบมัลติโปรเซสเซอร์ โดยใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นMaster และสามารถที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็น Slave ได้อีกเป็นจำนวนหลายๆตัว

รีจิสเตอร์ที่ใช้งานในการติต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และรีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับส่ง โดยรีจิสเตอร์ แต่ละตัวจะมีหน้าที่ และการทำงานในแต่ละบิตดังต่อไปนี้

รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ(Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 (ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล (TB8) รายละเอียดของแต่ ละบิตมีดังต่อไปนี้

ตารางที่ 2.1 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0, SM1: (Serial port mode bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด

ตาราง 2.2 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน

SM0	SM1	โหมด การทำงาน อัตรารับ-ส่ง
0	0	MODE 0 Shift register $f_{osc}/12$
0	1	MODE 1 8 bit UART Variable
1	0	MODE 2 9 bit UART $f_{osc}/32$ หรือ $f_{osc}/64$
1	1	MODE 3 9 bit UART Variable

SM2: เป็นบิตที่ทำหน้าที่ควบคุมการทำงานและเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์แบบ Single Processor System หรือ Multi processors System โดยกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM2 = 1 เป็นการเลือกแบบ Multi processors System คือระบบการสื่อสารแบบใช้ซีพียูหลายๆตัว ร่วมกันทำงานจะใช้งานในโหมด 2 หรือโหมด 3

SM2 = 0 เป็นการเลือกแบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานในโหมด 0 ต้องกำหนดให้ SM2 = 0) ในกรณีที่เลือกให้ SM2 = 1 แบบ Multi processors System

- ถ้าข้อมูลที่รับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น "1" ทำให้แฟลกอินเตอร์รัพต์ทางด้านรับ จะถูกเซตให้เป็น 1 (RI = 1)

- ถ้าข้อมูลในบิตที่ 9 รับเข้ามามีค่าเป็น "0" จะทำให้แฟลกอินเตอร์รัพต์ทางด้านรับเป็น 0 (RI = 0)การทำงาน ในโหมด 1 ถ้าให้ SM2 = 1 แฟลกอินเตอร์รัพต์ทางด้านรับ (แฟลก RI) จะไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop bit)

REN: (Enable Serial Reception) เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์ คือ 1 = ให้มีการรับข้อมูล, 0 = ไม่ให้มีการรับข้อมูล

TB8: (Transmit bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

RB8: (Receive bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการรับข้อมูล โดยใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้จะเป็นค่าของ Stop Bit ที่รับเข้ามา สำหรับโหมด 0 จะไม่ใช้งานบิตนี้

TI: (Transmit Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพท์ด้านส่งข้อมูล และจะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 (Shift register) หรือเมื่อเริ่มต้นส่ง Stop bit ในโหมด 1,2 หรือ 3และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ ของการส่งข้อมูลเรียบร้อยแล้ว

RI: (Receive Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพท์ทางด้านรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ในโหมด 0 (Shift register) และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้งเมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ของการรับข้อมูลเรียบร้อยแล้ว หรืออาจกล่าวได้ว่าถ้าบิต RI ถูกเซตเมื่อใด หมายถึงข้อมูลได้เข้ามาเก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้ว สามารถที่จะอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

รีจิสเตอร์ SBUF (serial data buffer register) เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่99H และเข้าถึงข้อมูลแบบไบต์ได้อย่างเดียว จะทำหน้าที่รับและส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ในการอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะต้องอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งเป็นบัฟเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาได้จากภายนอก และในทำนองเดียวกันขณะที่ต้องการส่งข้อมูล เราก็จะนำเอาค่าข้อมูลที่ส่งออกไปไว้ที่ในรีจิสเตอร์ SBUFก่อน และหลังจากนั้นจึงจะส่งออกไป โดยจะใช้คำสั่งการโอนย้ายข้อมูลแบบไบต์เช่น MOV SBUF,#20H หรือ MOV SBUF,@R1 ก็ได้ การรับข้อมูลในโหมด 0 จะเริ่มต้นรับ เมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start bit เข้ามาที่ขา RXD

รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียวเท่านั้น โดยจะประกอบด้วยบิตดังต่อไปนี้

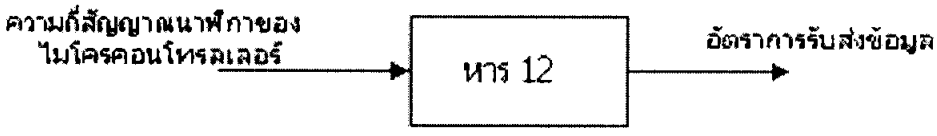
ตารางที่ 2.3 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

PC0N.7 SMOD: ในกรณีที่ใช้ไทม์เมอร์1 เป็นตัวกำหนดอัตรารับส่ง (Baud rate) และหากกำหนดให้บิตนี้มีค่าเป็น "0" ในการใช้งานกับพอร์ตสื่อสารอนุกรมโหมด 1, 2 และโหมด 3 ค่าอัตรารับส่ง (Baud rate) จะเพิ่มขึ้นเป็นสองเท่า รีจิสเตอร์ PCON ไม่สามารถอ้างตำแหน่งแบบบิตได้ แต่จะใช้คำสั่งทางลอจิกของการ ORเช่น ORL PCON,#80H จะเป็นการเซตบิตที่ 7 ของรีจิสเตอร์ PCON และการกำหนดให้บิตมีสถานะเป็น "0"หรือเคลียร์บิตจะใช้การ AND เช่น ANL PCON,#01111111B จะเป็นการเคลียร์บิตที่ 7 ของรีจิสเตอร์ PCON

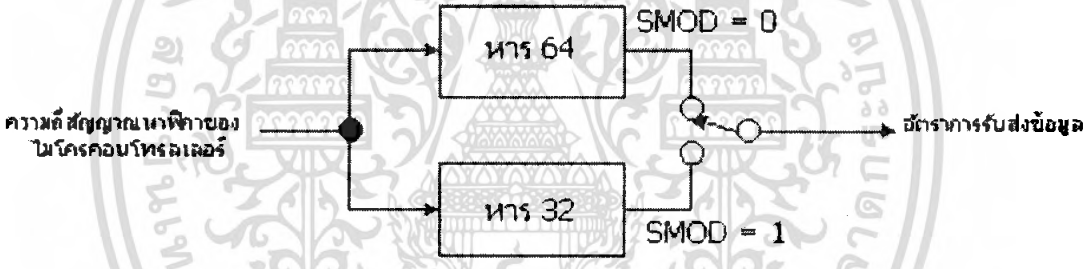
การกำหนดอัตรารับและส่งข้อมูล (Baud Rate Generator) อัตรารับส่ง (Baud rate) ในโหมด 0 ไม่สามารถกำหนดอัตรารับส่งเองได้ แต่จะมีค่าเท่ากับความถี่สัญญาณนาฬิกาไมโครคอนโทรลเลอร์ หารด้วย 12 หรืออาจกล่าวได้ว่าขึ้นกับค่าความถี่ของคริสตอลที่นำมาต่อใช้งาน แล้วหารด้วย 12 นั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 อัตรารับส่งโหมด 0

โหมด 2 อัตรารับส่ง (Baud rate) ในโหมด 2 เลือกได้ 2 อัตราความเร็วในการรับส่งข้อมูล โดยหาได้จากความถี่ของสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ หารด้วย 32 หรือหารด้วย 64 เรียกว่า SMOD0 และ SMOD1 ซึ่งจะขึ้นอยู่กับการกำหนดค่าสถานะของบิต SMOD ที่อยู่ใน รีจิสเตอร์ PCON เป็นตัวเลือก



รูปที่ 2.7 อัตรารับส่งโหมด 2

ถ้าบิต SMOD = 0 อัตรารับส่งโหมด 2 = 1/64 เท่าของความถี่สัญญาณนาฬิกา

ถ้าบิต SMOD = 1 อัตรารับส่งโหมด 2 = 1/32 เท่าของความถี่สัญญาณนาฬิกา

หลังจากที่เราทำการรีเซตระบบของไมโครคอนโทรลเลอร์ค่าข้อมูลในบิต SMOD จะเป็นสถานะ "0" เสมอ ดังนั้นเราสามารถที่จะเขียนเป็นสูตรสำหรับการคำนวณหาค่าอัตรารับส่ง (Baud rate) ได้ ดังสมการต่อไปนี้

อัตรารับส่งของโหมด 2 = [2SMODx(ความถี่สัญญาณนาฬิกา)]/64 (2.1)

ในกรณีที่เราใช้คริสตอลค่า 11.0592 MHz จะได้ค่าอัตรารับส่ง (Baud rate) สูงสุด = 345.6 K

ในกรณีที่เราใช้คริสตอลค่า 12 MHz จะได้ค่าอัตรารับส่ง (Baud rate) สูงสุด = 375 K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 1 และโหมด 3 จะมีอัตราการรับส่ง (Baud rate) ข้อมูลโดยถูกกำหนดได้ตามต้องการ โดยใช้อัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์1 หรือ ไทม์เมอร์ 2 เป็นตัวกำหนด

การใช้ไทม์เมอร์1 กำหนดอัตรารับส่ง (Baud rate) เมื่อใช้ไทม์เมอร์1 เป็นตัวสร้างอัตราการรับส่งข้อมูลของพอร์ตอนุกรมในโหมด 1 หรือโหมด 3 สังเกตได้ว่าเมื่อเกิดโอเวอร์โฟลว์ในไทม์เมอร์ตัวใด จะทำให้เกิดสัญญาณอินเตอร์รัพท์เพื่อบอกให้ซีพียูรับทราบ ดังนั้นในขณะใช้ ไทม์เมอร์1 เพื่อสร้างอัตรารับและส่งข้อมูล จะต้องไม่ให้มีการร้องขออินเตอร์รัพต์ที่เกิดจากไทม์เมอร์1 ในกรณีใดๆในระหว่างนั้นอีก (โดยการควบคุมที่ รีจิสเตอร์ IE) อัตราการรับและส่งข้อมูลจะมาจากอัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์1 และค่าของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON สามารถที่จะเขียนเป็นสูตร

สำหรับการคำนวณหาค่าอัตรารับส่ง (Baud rate) ได้ดังสมการต่อไปนี้อัตรารับส่งในโหมด 1 และ 3 = x (อัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์1)

อัตรารับส่งของโหมด 1 และ 3 = [2SMODx(อัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์1)] /32

(2.2)

ในการใช้งานสื่อสารข้อมูลแบบอนุกรมนั้น เราจะนิยมใช้งานไทม์เมอร์1 ในลักษณะของโหมด 2 (Autoreload)กำหนดค่าการควบคุมที่รีจิสเตอร์ TMOD = 0010XXXX) การคำนวณหาค่าอัตรารับส่ง (Baud rate) จะได้สมการดังต่อไปนี้

อัตรารับส่งของโหมด 1 และ 3 = [2SMODx(Oscillator)] /[32x12x(256-(TH1))] (2.3)

เราสามารถที่จะให้ไทม์เมอร์1 มีอัตราการรับส่งต่ำๆ ได้โดยใช้ ไทม์เมอร์1 ในโหมด1 ทำงานในลักษณะของตัวจับเวลาแบบ 16 บิต (มีค่าของการควบคุมที่รีจิสเตอร์ TMOD = 0001XXXX) และให้มีการอินเตอร์รัพต์จากไทม์เมอร์1 โดยให้โปรแกรมตอบสนองการอินเตอร์รัพต์ของไทม์เมอร์1 แล้วจะทำการกำหนดค่าเริ่มต้นใหม่ (Reload) ให้กับตัวจับเวลา ซึ่งเป็นการทำงานแบบ 16 บิตทางซอร์ฟแวร์ (Software reload) เนื่องจากการทำงานในโหมด 1 ของไทม์เมอร์1 ไม่สามารถโหลดค่าใหม่เองด้วยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงตัวอย่างอัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้บอดเรทค่ามาตรฐานต่างๆ

Baud rate	F_{osc} (MHz)	บิต SMOD	Timer1		
			บิต C/T	MODE	Reload Value
62.5K	12.00	1	0	2	FFH
19.25K	11.0592	1	0	2	FDH
9600	11.0592	0	0	2	FDH
4800	11.0592	0	0	2	FAH
2400	11.0592	0	0	2	F4H
1200	11.0592	0	0	2	F4H
137.5	11.0592	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	2	FEEBH

จะสังเกตได้ว่าการใช้งานจากความถี่คริสตอลเรามักจะนิยมจะเลือกใช้ค่าความถี่ 11.0592 MHz

เนื่องจากการกำหนดค่าอัตราการรับส่งข้อมูลจะสามารถกำหนดค่าบอดเรทในโหมด 1 และโหมด3 ได้ และเป็นค่าที่มาตรฐานเช่น 1200, 2400, 4800, 9600, 19200 ดังนั้นจึงเป็นเหตุผลที่เราเลือกใช้คริสตอลค่าความถี่ 11.0592 MHz มากกว่าค่า 12 MHz ในการกำหนดอัตรารับส่งข้อมูลยังสามารถที่จะใช้ไทม์เมอร์ 2 กำหนดค่าได้ แต่จะยังไม่ขอกล่าวในที่นี้เพราะไอซีที่เรานำมาทดลองนั้นไม่มีไทเมอร์ 2 ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2 การแสดงผลแบบสแกน

ในการแสดงผล LED ถ้าเราแสดงผลพร้อมๆกัน หลายๆหลัก จะทำให้ต้องใช้เอาต์พุตจำนวนมากและต้องการใช้กระแสไฟฟ้ามาก จึงมีการใช้การแสดงผลแบบแสกนขึ้น โดยให้แต่ละ Column พลัดกัน แสดงข้อมูลทีละหลักสลับกันไปด้วยความเร็วสูง ทำให้มองเห็นเหมือนว่าแสดงพร้อมๆกันทุกหลัก

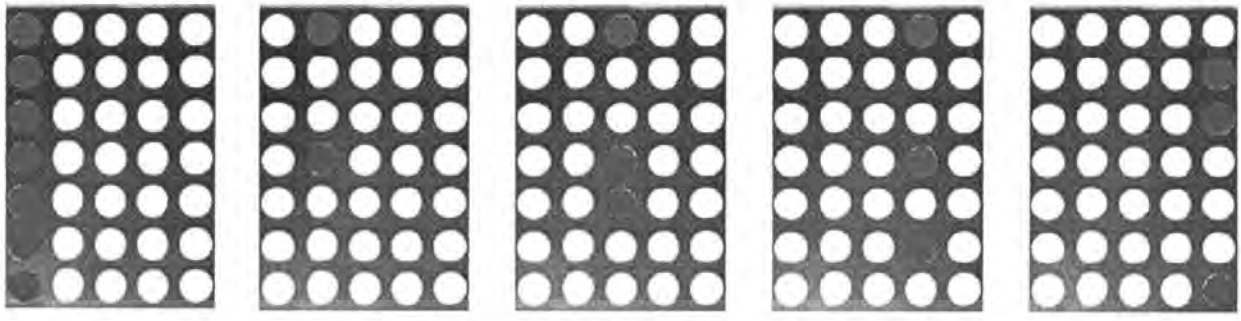
ในการแสดงผลตัวอักษร เราต้องออกแบบข้อมูลที่เราจะส่งออกไปแสดงผลก่อน กรณีนี้เราจะแสดงผลเป็นตัว R เราจะใช้ port 1 เป็น port ส่งข้อมูล โดยจะอ่านค่าในหน่วยความจำของไมโครคอนโทรเลอร์ AT89C51 โดยทำการทดลองโปรแกรมแสดงผลง่ายๆ (5 columns) ก่อน เช่น ข้อมูล Binary ตัวอักษร R ที่เก็บไว้คือ

ตารางที่ 2.5 ตัวอย่างเลขฐานสองที่จะแสดงผลทางจอ LED

Address 1	11111111
Address 2	10001001
Address 3	10011001
Address 4	10101001
Address 5	11000110

การแสดงผลของตัวอักษร จะเริ่มจากเปิดหลักที่ 1 (Logic ที่ขา IC ของ Column ที่ 1 มีค่าเท่ากับ "0")จากนั้นก็ส่งข้อมูลจากตำแหน่งที่ 1 ออกมาทาง port 1 หน่วงเวลาไว้ซักพักจากนั้นก็เคลียร์ค่าใน port 1 ต่อมาก็ปิดหลักที่ 1 และเปิดหลักที่ 2 (Logic ที่ขา IC ของ Column ที่ 1 มีค่าเท่ากับ "1" ส่วนที่ขา IC ของ Column ที่ 2 มีค่าเท่ากับ "0") จากนั้นก็ส่งข้อมูลจากตำแหน่งที่ 2 ออกมาทาง port 1 จากนั้นก็ทำอย่างเดียวกัน ทำอย่างนี้ไปเรื่อยๆ จนครบ 5 หลักก็เริ่มแสดงผลใหม่จากหลักที่ 1 ทำซ้ำไปเรื่อยๆ ด้วยความเร็วสูง เราก็จะเห็น LED Dot Matrix แสดงผลเป็นรูปตัวอักษร R ดังภาพ

82474
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การแสดงผลตัวอักษร R โดยการแสกน ซึ่งจะแสดงผลทีละหลักไปเรื่อยๆจนครบ 5 columns

รูปที่ 2.9 การแสดงผลตัวอักษร R เมื่อแสกนด้วยความเร็วสูง

2.3 หลักการรับส่ง SMS

SMS : Short Message Service คือ การให้บริการส่งข้อความสั้นไปยังโทรศัพท์มือถือแบบดิจิตอลโดยแต่ละข้อความสามารถบรรจุตัวอักษรสูงสุดได้ 160 ตัวอักษร (อักษรภาษาอังกฤษ) นอกจากนี้ยังสามารถส่งข้อความไปที่เครื่อง Fax, PC หรือ Internet address อีกด้วยระบบ SMS ในระบบเครือข่ายโทรศัพท์มือถือ รองรับโดยระบบ GSM (Global System for MobileCommunication), TDMA (time division multiple access) และ CDMA (code division multiple access)เมื่อ SMS ถูกส่งจากโทรศัพท์มือถือเครื่องหนึ่ง ข้อความนั้นจะถูกส่งไปที่ Short Message Service Center (SMSC) จากนั้นจึงจะส่งไปยังโทรศัพท์มือถือเครื่องรับอีกทอดหนึ่ง โดยมีกระบวนการดังนี้

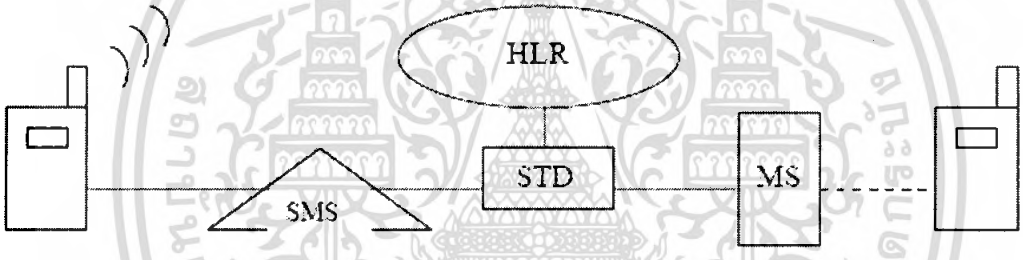
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. SMSC จะส่ง SMS Request ไปยัง Home location register (HLR) เพื่อหาตำแหน่งของผู้รับ

2. เมื่อ HLR ได้รับสัญญาณ Request ก็จะส่งสถานะของผู้รับ (subscriber's status) กลับมายังSMSC คือ

1) สถานะของเครื่องรับ Inactive หรือ Active

2) ตำแหน่งของเครื่องรับ



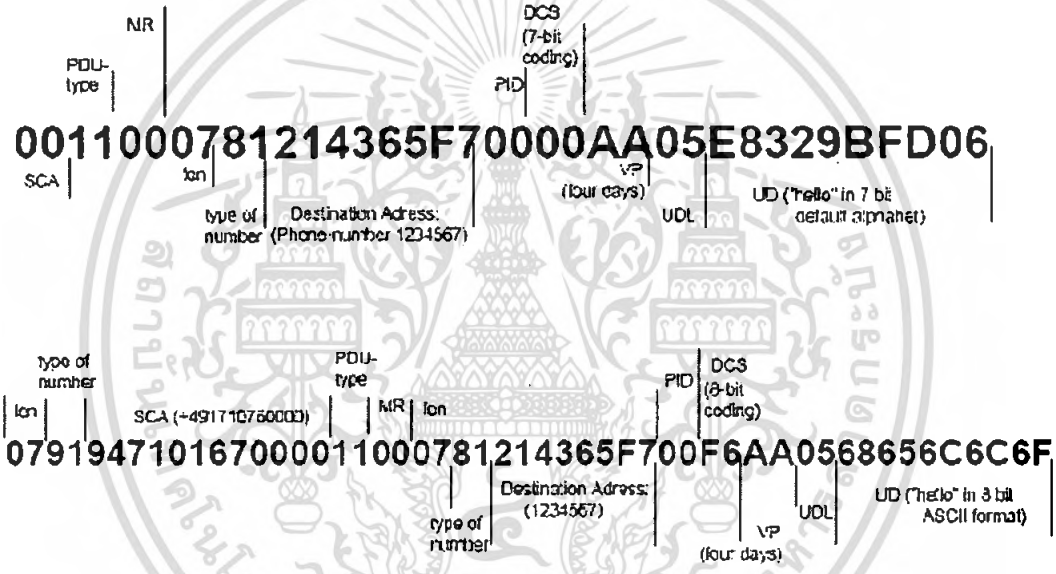
รูปที่ 2.10 หลักการรับ-ส่ง SMS

ถ้าสถานะของเครื่องรับเป็น Inactive แล้ว SMSC จะเก็บข้อความไว้ช่วงเวลาหนึ่ง และเมื่อใดที่เครื่องรับมีสถานะ Active แล้ว HLR จะส่ง SMS Notification ไปยัง SMSC และ SMSC ก็จะตอบรับข้อความนั้นไว้ จากนั้น SMSC จะส่งผ่านข้อความไปในรูปแบบ Short Message Delivery Point-to-Point ไปยังระบบบริการ โดยระบบจะทำการเรียกไปยังเครื่องรับ และถ้าเครื่องรับมีการตอบรับกลับมา ข้อความก็จะถูกส่งตามไปและ SMSC จะได้รับการตอบยืนยันว่า ข้อความได้ถูกรับโดยปลายทาเรียบร้อยแล้ว หลังจากนั้นข้อความจะมีสถานะเป็น SENT และจะไม่ถูกส่งอีกการรับส่งข้อมูล SMS มี 2 โหมด คือ Text Mode และ PDU Mode (Protocol Description Unit Mode) การส่งข้อมูล Text Mode นั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน แล้วค่อยส่งข้อมูลใน PDU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 PDU MODE

PDU : Protocol Description Unit คือ รูปแบบของข้อมูลที่ใช้ในการส่งข้อความสั้น ซึ่งเกิดจากการนำข้อมูลดิจิตอลมาเข้ารหัสเพื่อใช้ในการส่งผ่านข้อมูลผ่าน Air Interface โดยการรับและส่งข้อมูล SMS ในแบบPDU MODE จะมีส่วนประกอบและโครงสร้างที่แตกต่างกันPDU CODE ข้อมูล SMS ที่อยู่ใน PDU MODE ประกอบด้วยเลขฐาน 10 และเลขฐาน 16 โดยตัวเลขแต่ละคู่เรียกว่า Octet



รูปที่ 2.11 ข้อมูลใน SMS ในรูปแบบ PDU MODE

2.4.1 การรับข้อมูล SMS

ข้อมูลที่ได้รับจะประกอบด้วยข้อมูลของผู้ส่ง, ข้อมูล SMS Service Center (SMSC), Time Stampและอื่นๆ ตามด้วยส่วนของข้อความซึ่งจะอยู่ท้ายสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ความหมายของรหัส SMS ที่ได้รับ

ตัวอย่าง

0691668111808804 0A91662950268000004030212194348 20AE8329BFD4697D9E CC37

ความหมายของรหัสในแต่ละ Octet หรือ กลุ่ม Octet เป็นไปตามตาราง ดังนี้

กลุ่มตัวเลข 8 บิต (Octet)	รายละเอียด
06	ความยาวของ SMSC Information 6 Octet
91	รูปแบบของเลขหมาย SMSC 91 หมายถึงเลขหมายแบบสากล
66 81 11 80 88	เลขหมาย SMSC ซึ่งจะเป็นเลขฐาน 10 สลับ nibble ในกรณีนี้เลขหมายจริงของ Service Center คือ +6618110888
04	First octet of these SMS-DELIVER message
0A	ความยาวของเลขหมายผู้ส่ง (0A hex = 10 ตัว)
91	รูปแบบของเลขหมายผู้ส่ง 91 หมายถึงเลขหมายแบบสากล
66 29 50 26 80	เลขหมายผู้ส่ง เป็นเลขฐาน 10 สลับ nibble ในกรณีนี้เลขหมายจริงของผู้ส่งคือ +6692056208
00	TP-PID. (Protocol identifier)
00	TP-DCS (Data coding scheme) 00 คือเข้ารหัสข้อความแบบ 7 bits Default Alphabet
40 30 21 21 94 34 82	TP-SCTS. ข้อมูล Time stamp สลับ nibble
0A	TP-UDL. User data length จำนวนตัวอักษรของข้อความที่ส่งในที่นี้คือ 10 ตัว
E8329BFD4697D9ECC37	TP-UD. ข้อความ "hellohello" ที่เข้ารหัสแบบตัวอักษร 7 bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การถอดรหัสตัวอักษรชนิด 7 บิต

ในส่วนของข้อมูลที่เป็นข้อความสั้นในกรณีที่เป็นตัวอักษรภาษาอังกฤษ อักษรแต่ละตัวจะใช้รหัสขนาด7 บิต (7 bits default alphabet) และสำหรับอักษรภาษาอื่นๆ จะใช้ตัวอักษรชนิดอื่น เช่น 8 บิต หรือ 16 บิตสำหรับข้อความสั้นภาษาไทยนั้น ใช้รหัสตัวอักษรแบบ UNICODE ในที่นี้เราจะกล่าวถึงการถอดรหัสข้อความสั้น
ภาษาอังกฤษเท่านั้นโดยมีขั้นตอนการถอดรหัส ดังนี้

1) นำ PDU CODE ในส่วนของข้อความสั้น (TP-UD) ซึ่งเป็นเลขฐาน 16 มาเขียนเป็นเลขฐาน 2 ทีละ Octet

2) ตัวอักษรแรกเกิดจากบิตที่ 0 ถึง 6 ของ Octet แรก ตัวอักษรถัดมาเกิดจากบิตที่ 0 ถึง 5 ของOctet ที่ 2 และนำบิตที่ 7 ของ Octet ที่ 1 มาต่อท้าย ตัวอักษรถัดไปเกิดจากบิตที่ 0 ถึง 4 ของOctet ที่ 3 และนำบิตที่ 7 และ 6 ของ Octet ที่ 2 มาต่อท้าย ทำเช่นนี้ไปเรื่อยๆ

3) นำรหัสตัวอักษร 7 บิต จากข้อสอง เพิ่มบิตที่ 8 ด้วย '0' นำไปเทียบกับตารางรหัส ASCII โปรแกรม Microcontroller

2.5 AT-COMMAND

AT-COMMAND คือ ชุดคำสั่งมาตรฐาน ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็ม หรือ อุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่าหรือสั่งงานอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ และสำหรับการติดต่อกับโทรศัพท์มือถือ จะใช้ชุดคำสั่งที่เรียกว่า GSM AT COMMANDตัวอย่างคำสั่งที่เป็น BASIC AT COMMAND

AT ตรวจสอบความพร้อมของอุปกรณ์ ถ้าสามารถติดต่อกันได้อุปกรณ์จะตอบกลับมาว่า OK

ATDT phone number; โทรไปยังเลขหมายปลายทาง (phone number)

ATH วางสาย

ATA รับสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 AT COMMAND ที่เกี่ยวข้องกับการรับ-ส่ง SMS

ชุดคำสั่ง AT COMMAND ที่ใช้กับโทรศัพท์มือถือได้มีอยู่มากมาย ทั้งการอ่านรุ่นโทรศัพท์มือถือ,ตรวจสอบระดับแบตเตอรี่, ตรวจสอบระดับสัญญาณ แต่ในที่นี้จะกล่าวถึงเพียงคำสั่งที่เกี่ยวข้องกับการรับ-ส่ง SMS เท่านั้น

1) Message Format (AT+CMGF) เป็นคำสั่งกำหนดรูปแบบของข้อความที่จะให้แสดงออกมา โดย

AT+CMGF = 1 คือ แสดงข้อความในรูปแบบ TEXT

AT+CMGF = 0 คือ แสดงข้อความในรูปแบบ PDU CODE

2) List Message (AT+CMGL) เป็นคำสั่งที่ให้แสดงข้อความในสถานะต่างๆ โดยจะแสดงข้อความ
ทั้งหมด นั้นสถานะนั้น ดังนี้

AT+CMGL=0 คือ แสดงข้อความที่ได้รับแต่ยังไม่ได้อ่าน

AT+CMGL=1 คือ แสดงข้อความที่ได้รับและอ่านแล้ว

AT+CMGL=2 คือ แสดงข้อความที่เก็บไว้และยังไม่ได้ส่ง

AT+CMGL=3 คือ แสดงข้อความที่เก็บไว้และส่งออกไปแล้ว

AT+CMGL=4 คือ แสดงข้อความทั้งหมด (“ALL”)

หมายเหตุ .. หากกำหนด Message Format เป็น PDU CODE จะต้องเลือกสถานะโดยใช้ตัวเลข 0 ถึง4 แต่หากกำหนด Message Format เป็น Text จะต้องเลือกสถานะโดยใช้ตัวอักษรที่วงเล็บด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 ตัวอย่างคำสั่งพื้นฐานและ LIST ข้อความใน STO SENT

3) Read Message (AT+CMGR) เป็นคำสั่งที่ใช้อ่านข้อความที่เฉพาะเจาะจงได้โดยระบุตำแหน่งที่ข้อความนั้นถูกเก็บ

```
AT
OK
AT+CMGF=1
OK
AT+CMGR=5
+CMGR: "REC READ","+66149727877","04/05/02,23:37:02+28",0
It's ok wa.He understands u when u drank a lot 555;p he is a nice guy woy.

OK
```

รูปที่ 2.13 ตัวอย่างการอ่านข้อความ

4) Set the Message center number (AT+CSCA="XXX") เป็นคำสั่งที่ใช้ระบุหมายเลขของ SMS เพื่อจะทำการส่ง SMS ต่อไปได้

5) Send Message (AT+CMGS="YYY") เป็นคำสั่งที่ใช้ระบุเลขหมายปลายทางที่ต้องการส่ง SMS ไปถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 SERIAL COMMUNICATION

การเชื่อมต่อระหว่าง MCS-51 กับ มือถือนั้นจะทำการเชื่อต่อกันทางพอร์ตอนุกรมโดยที่มือถือจะมีขาพินดังตารางต่อไปนี้

Pin	Name	IN/OUT	Notes
1	GND		
2	SB	I/O	Charger coding and charger control.
3	POWER	I	Charging Current
4	FBatt+	O	Power supply for the accessories.
5	TX	O	Serial interface
6	RX	I	Serial interface
7	ZUB_CLK	I/O	Clock line for accessory bus Use as DTC In data operation
8	ZUB_DATA	I/O	Data line for accessory bus. Use as CTS in data operation
9	GND_MIC		For external microphone
10	HF_MIC	I	External microphone
11	AUDO	O	Trigger for external loudspeaker
12	GNDA		For external loudspeaker

ตารางที่ 2.7 หน้าที่ของขาพินแต่ละขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขาที่เราสนใจนั้นคือขา TX และ RX โดยที่ขา TX จะทำหน้าที่เป็นตัวส่งข้อมูล และขา RX จะทำหน้าที่รับข้อมูล ซึ่งในขาของ MCS-51 ก็มีสองขานี้อยู่แล้ว

2.7 รหัส ASCII

เมื่อถอดรหัส PDU มาได้จะได้เป็นเลขฐาน 16 ซึ่งเลขฐาน 16 นี้ เป็นรหัส ASCII ซึ่งจะสามารถถอดรหัส ASCII ให้เป็นภาษาที่ใช้สื่อสารกันโดยปกติได้โดยดูจากตารางนี้

ตารางที่ 2.8 ตารางการแปลงรหัสASCII

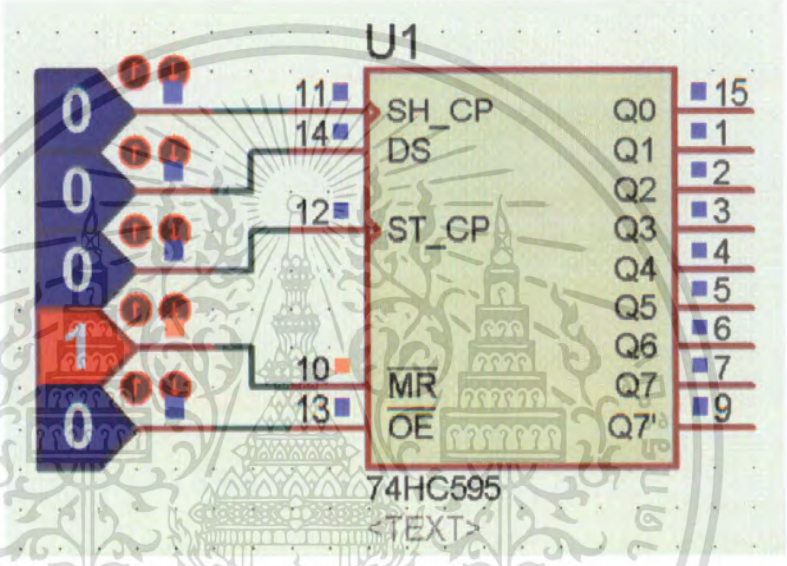
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ไอซี 74HC595

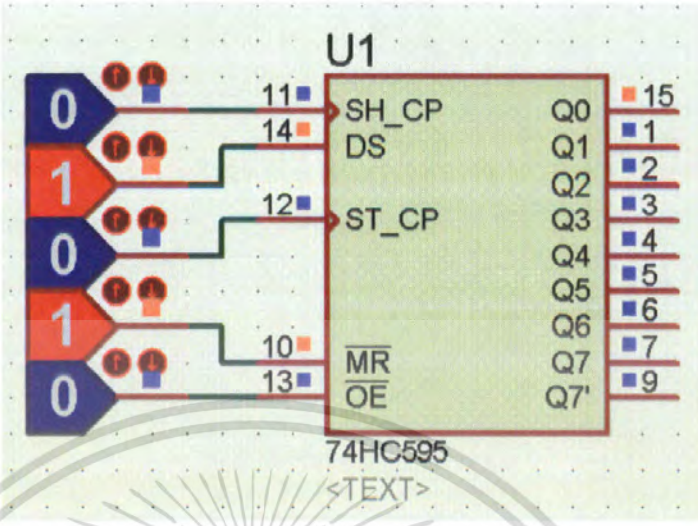
ไอซีเบอร์นี้จะเป็นไอซีที่สำคัญตัวหนึ่งในการแสดงผล LED แบบสแกนนิ่งโดยไอซีตัวนี้จะเป็นตัว Shift bit ทีละตัวเพื่อเป็นตัวกำหนดการที่จะให้เกิดการสแกนนิ่งตาทที่ได้กล่าวไปแล้วในหัวข้อ 2.2 โดยจะอธิบายการทำงานของมันโดยตัวอย่างภาพประกอบดังต่อไปนี้



รูปที่ 2.14 ขั้นที่ 1 ของการทำงาน 74HC595

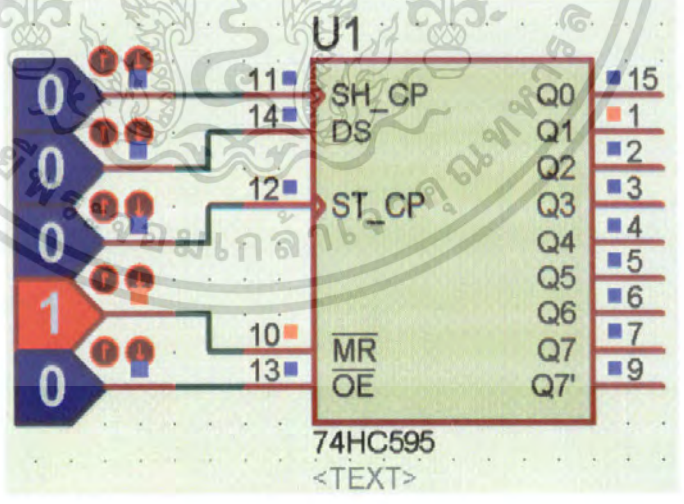
ขั้นที่ 1 จะทำการตั้งค่าลอจิก 1 ให้กับ $\overline{MR}$ ซึ่งจะทำให้ไม่เกิดการรีเซ็ต ตั้งค่าที่ $\overline{OE}$ ให้เป็น 0 เพื่อที่เราจะต้องการให้ค่าเอาท์พุตออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ขั้นที่ 2 ของการทำงาน 74HC595

ขั้นที่ 2 ขา DS ต้องเป็นลอจิก 1 และทำการทริก (เปลี่ยนแปลงจากลอจิก 0เป็น1 และเปลี่ยนกลับเป็น 0 อีกที) ที่ขา SH_CP และทำการทริกที่ขา ST_CP อีก 1 ทีก็จะทำให้เกิดลอจิก 1 ที่ Q0

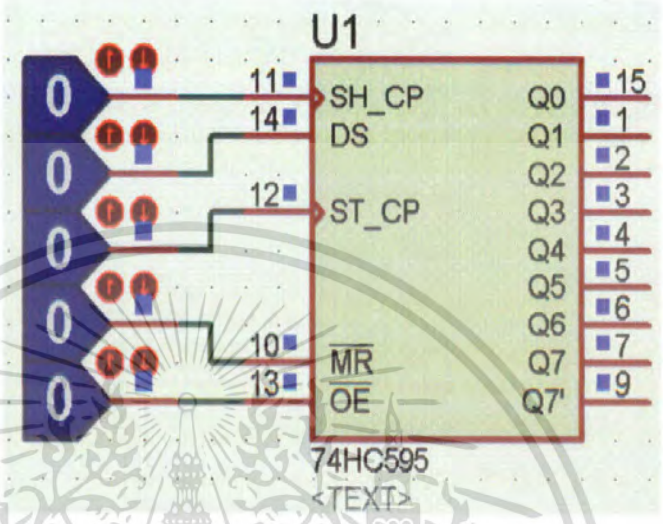


รูปที่ 2.16 ขั้นที่ 3 ของการทำงาน 74HC595

ขั้นที่ 3 ขา DS จะให้เป็นลอจิก 0 ต่อมาเราก็ทำการทริกขา SH_CP หนึ่งที และ ทริกที่ขา ST_CP อีกหนึ่งที ก็จะทำให้เกิดการเลื่อน ลอจิก 1 จาก Q0 ไป Q1 ได้ตามที่เราต้องการใช้งานกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LEDซึ่งถ้าเราจะต้องการเลื่อนอีกครั้งก็จะทำได้โดยการทริกที่ขา SH_CP หนึ่งที และ ขา ST_CP อีก 1 ที



รูปที่ 2.17 ขั้นที่ 4 ของการทำงาน 74HC595

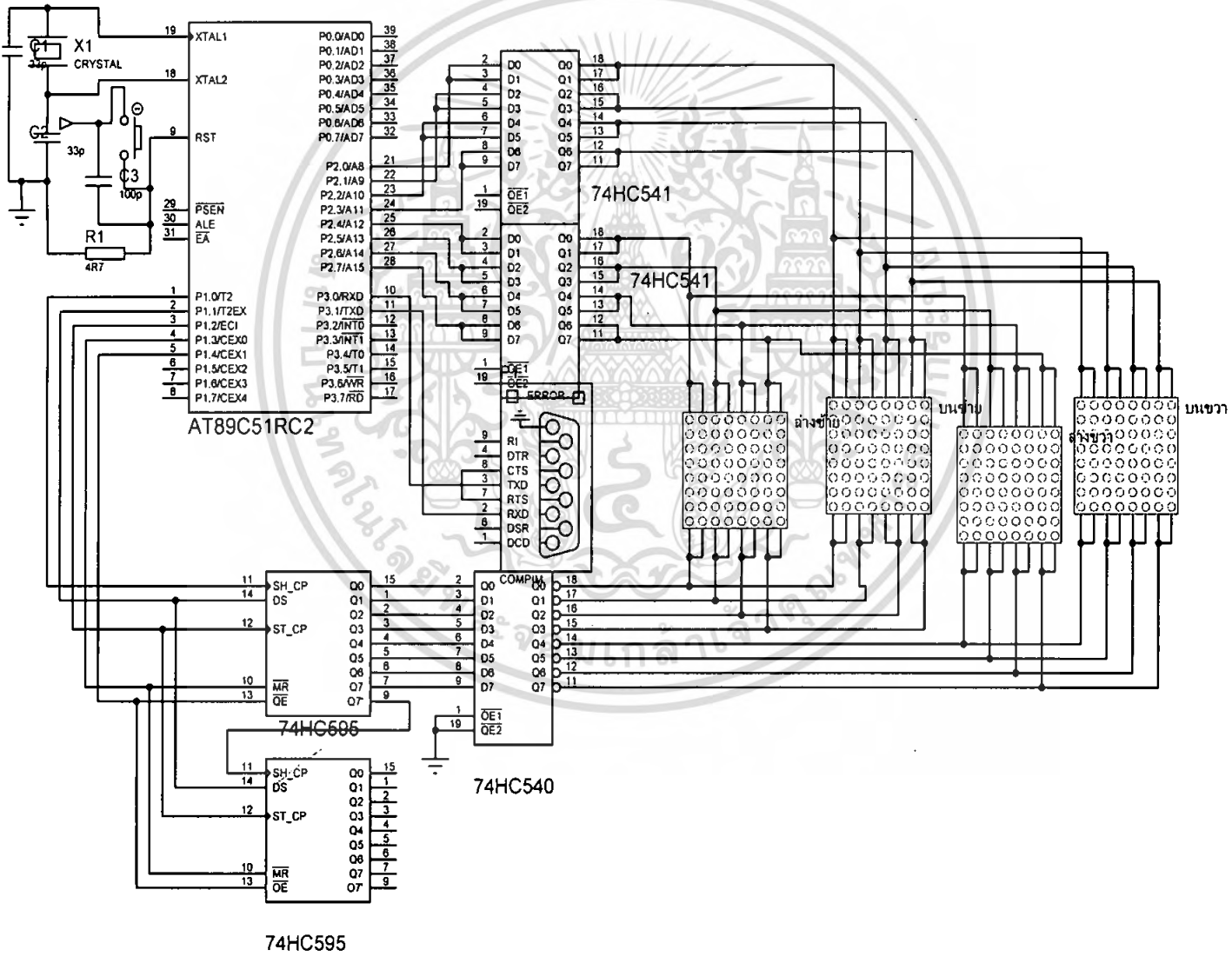
เมื่อเราต้องการตั้งค่าให้เกิดการรีเซ็ตค่าใหม่จะทำได้โดยให้ ลอจิก 0 ที่ $\overline{MR}$ และทำการทริกที่ขา SH_CP หนึ่งที และทริกที่ขา ST_CP ก็จะเกิดการรีเซ็ตที่ค่า เอาท์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบวงจร

3.1 วงจรแสดงผล SMS แบบ LED



รูปที่ 3.1 วงจรแสดงผล SMS

โดยวงจรจะสามารถแบ่งได้เป็น 2 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 วงจรส่วนที่รับส่งข้อความจากมือถือ

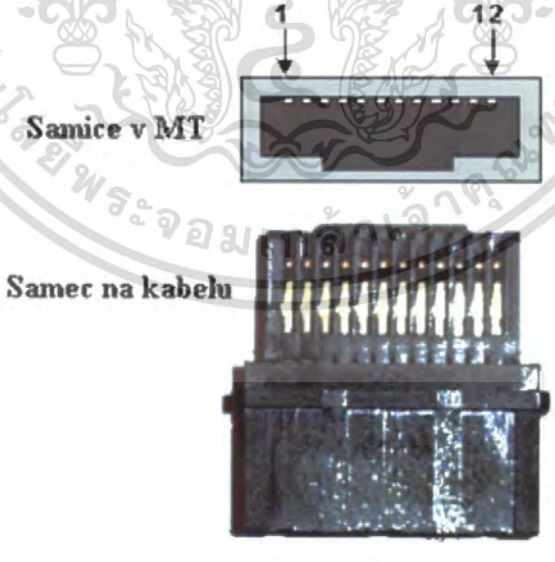
รูปที่ 3.2 วงจรส่วนที่รับส่งข้อความจากมือถือ

จากรูปวงจรขา Tx ของ MCS51 จะทำหน้าที่เป็นตัวส่งข้อความที่แป็นคำสั่ง AT_COMMAND ที่ให้มือถือแสดง SMS โดยจะส่ง รหัส PDU เข้ามาทางขาของ Rx MCS51 และ MCS51 จะทำการเก็บค่ารหัส PDU พร้อมกับถอดรหัสและเก็บค่าไว้อีกทีนึงเพื่อนำไปแสดงผลใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของวงจร LED อุปกรณ์มือถือที่ใช้จะเป็น SIEMENS รุ่น C35 โดยเราจะเชื่อมต่อกับ MCS51 ผ่านทาง HAND-FREE ที่แกะข้างในออกมาและบัดกรีขาที่เป็น ส่วน Tx และ Rx ดังรูป

รูปที่ 3.3 SIEMENS C35



รูปที่ 3.4 ตัวเชื่อมระหว่าง C35 กับ MCS51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

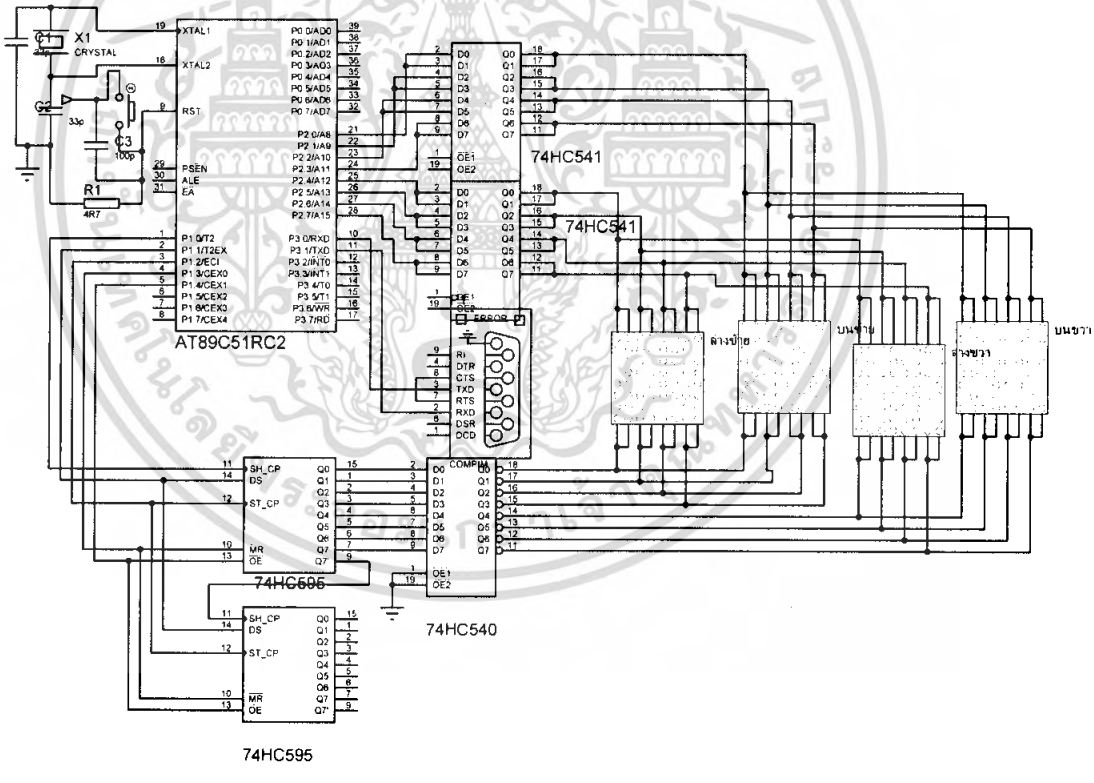
3.1.2 วงจรส่วนที่แสดงผลทาง LED

หลักการที่ใช้

1. การใช้ภาพติดตา กล่าวง่ายๆก็คือ การที่เรากระพริบไฟด้วยความถี่ที่ตาไม่สามารถตรวจจับได้ ทำให้เรามองเห็นไฟไม่กระพริบ ซึ่งต้องใช้ความถี่อย่างน้อย 1/10 วินาทีต่อหนึ่งภาพ ซึ่งในการฉายภาพยนตร์จะใช้ความถี่ประมาณ 1/20 วินาทีต่อหนึ่งภาพ

2.การแสกน โดยการที่เราจะมองเห็นเป็นตัวอักษรโดยการกระพริบนั้น เราจะต้องมีการกระพริบให้เป็นระบบ ซึ่งในโปรเจคนี้เราจะใช้การแสกน ตามแนวตั้งของตัวอักษร แสกนที่ละหลักจากซ้ายไปขวา

วงจรภาคแสดงผลใน dotmatrix



รูปที่ 3.5 แบบการแสดงผลแบบ 16*16

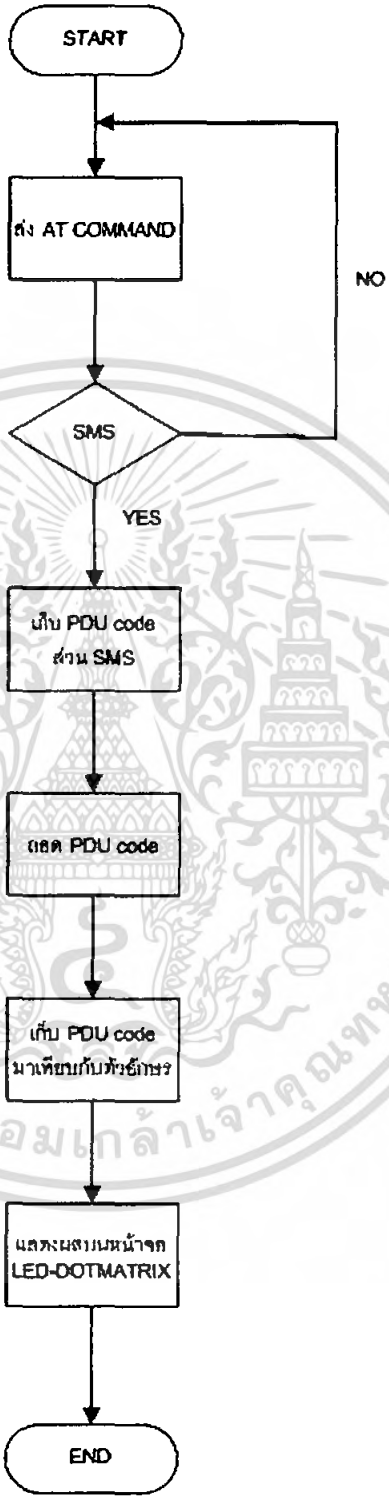
จากรูปที่ เป็นวงจรการแสดงผล แบบ 16*16 โดยในโปรเจคนี้จะทำการ ต่อเป็นแถวยาว โดยมี74HC595 เป็นตัวแสกน โดยต่อเหมือนกับในรูปที่ โดยจะเห็นได้ว่าการแสกนแต่ละหลักนั้นจะทำการเชื่อมต่อกันสองขากล่าวคือ จะแสกนที่ละ 2 แถว ในส่วนของ data ที่เข้ามาก็เช่นเดียวกันจะทำการเชื่อมกันไว้ติดกันเป็นคู่ๆ กล่าวคือการให้ติดทีละ 2 หลัก ฉะนั้นหากพิจารณาให้ดีแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pixel ต่ำสุดจะมีขนาดเท่ากับ 8*8 คือติดทีละ 4 จุด การที่ใช้ 74HC541 สองตัวก็เพราะ กระแสจาก1 IC ไม่พอที่จำเลี้ยง จุดในดอท เมทริกซ์ทั้ง 4 จุดได้ฉะนั้นเราจึงจำเป็นต้องใช้ IC 74HC541 สองตัว โดยส่วนแสกนนั้นจะใช้ 74HC540 เป็นตัวกลับเฟสของ 74HC595 เพื่อทำให้เกิดลักษณะกราววิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แผนผังการทำงานของโครงงาน



รูปที่ 3.6 แผนผังการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ในการทดสอบโปรแกรมการแสดงผลทาง LED โดยผ่าน Proteus เมื่อเราตั้งค่า delay ที่มากกว่าเดิมเล็กน้อยก็จะทำให้การแสดงผลไม่เป็นไปตามที่ตั้งใจไว้ ซึ่งมันจะแสดงเป็นแบบกระพริบๆแบบที่ตาเราสามารถมองเห็นได้แต่เมื่อเราทดสอบกับการต่อวงจรจริงๆปัญหานี้ก็จะไม่เกิดขึ้น

ในการทดสอบการรับข้อความจากมือถือ โดยทำการทดลอง โดยใช้คอมพิวเตอร์กับมือถือผ่านการเชื่อมต่อโดยอินฟาเรด เมื่อทำการทดลองโดยเชื่อมต่อกับมือถือผ่านCompim ก็จะไม่สามารถระบุได้ว่าการเขียนโปรแกรมของเรารับค่าที่ได้จะปรากฏบนจอ Virtual Terminal แต่เมื่อดำเนินการโปรแกรมผ่าน Proteus ก็จะแสดงการรับค่าจากมือถือที่ไม่มีความแน่นอนเลยทำให้ไม่สามารถระบุได้ว่าเราต้องปรับปรุงการเขียนโปรแกรมการรับอย่างไร

ในการทดสอบลองกับของจริงในเรื่องของการเชื่อมต่อพอร์ตอนุกรม ผลการทดลองจะแตกต่างกับการ ใช้โปรแกรม proteus โดยชิ้นเชิงเพราะว่าสามรถรับรหัส PDU ที่มือถือส่งมาได้ครบทุกตัวอักษร

5.2 สิ่งที่ได้รับจากการทำโครงงาน

มีความรู้ความเข้าใจในการทำงานของการใช้งานไมโครคอนโทรเลอร์ และการเขียนโปรแกรมภาษา C โดยใช้ Keil และการนำโปรแกรมที่เขียนไปทดสอบกับ Proteus และเข้าใจในหลักการเชื่อมต่อทางพอร์ตอนุกรมกับมือถือ โดยผ่านขา Tx และ Rx มีความเข้าใจในการถอดรหัส PDU เพื่อแปลงมาเป็นเลขฐาน 16 และนำไปแสดงผลเป็นตัวอักษรที่จอ LED และการแสดงผล LED จะต้องใช้ความรู้ในการใช้ความถี่หรือการ สแกนนิ่งเพื่อที่จะแสดงผล LED โดยประหยัด พอร์ตของ MCS51 มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะในโครงงาน

ในการทำโครงงานครั้งนี้จะสำเร็จได้เมื่อเราสามารถแก้ปัญหาดังต่อไปนี้

1.การเชื่อมต่อระหว่าง MCS51 กับมือถือ

โดยต้องทำการทดลองโดยส่ง at command ที่ชื่อว่า atd และตามด้วยหมายเลขปลายทาง ถ้าการทดลองได้ผลมือถือก็จะโทรออกไปที่เบอร์นั้น การทดลองนี้ปัจจัยสำคัญคือ การตั้งค่า baudrate ที่ถูกต้อง

2.การรับค่ารหัส PDU จากมือถือ

ทำการเขียนโปรแกรมเพื่อที่จะรับข้อความที่เป็นรหัส PDU โดยนำค่าที่ได้เก็บไว้ในตัวแปร array และนำค่านั้นมาเพื่อตรวจสอบโดยให้ตัวแปรนั้น ขึ้นจอ LCD

3.การถอดรหัส PDU

ทำการเขียนโปรแกรมเพื่อที่จะถอดรหัสข้อความรหัส PDU และนำค่าที่ได้ไปเก็บไว้ในตัวแปร array และนำค่านั้นไปแสดงผลที่จอ LCD

4. การแสดงผล LED

ทำการทดลองโดยเขียนโปรแกรมที่จะแสดงตัวอักษรที่เก็บไว้ใน array และทำการต่อวงจรสำหรับแสดงผล LED เพื่อทดสอบการทำงานของวงจรและโปรแกรม

สิ่งที่สำคัญที่สุดของการทำโครงงานนี้คือการที่เราต้องมีความพยายามในการลองผิดลองถูกและการศึกษาด้วยตัวเองให้ถึงที่สุดก่อน ถึงจะไปขอคำปรึกษาจากคนอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดสอบวงจรและโปรแกรม

4.1 การทดสอบส่วนของโปรแกรม

4.1.1 ทดสอบส่วนที่ส่ง AT_COMMAND

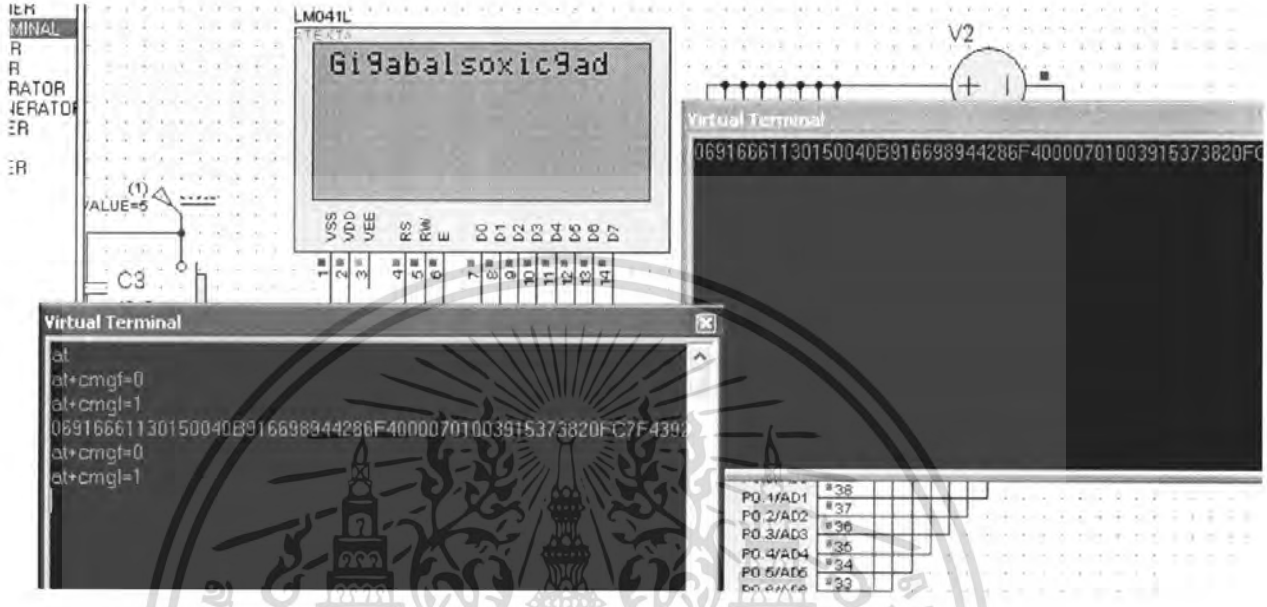
ในการทดลองครั้งนี้เราจะทำการจำลองโดยใช้ PROTEUS ในการจำลองโปรแกรมที่ได้เขียนขึ้นมาโดยเราจะลองทดลองโดยพิมพ์ข้อความเข้าไปแทนจำลองแทนการส่งรหัส PDU ของมือถือเพื่อที่จะทดสอบแค่ส่วนที่รับส่งข้อความเท่านั้น

รูปที่ 4.1 แสดงผลการทดสอบโปรแกรมในการรับส่งข้อความ

ภาพของ Virtual Terminal ทางซ้ายจะเป็นการส่งชุดคำสั่ง AT_COMMAND ภาพ Virtual Terminal ทางขวาจะแสดงผลของการที่เราจำลองการส่งข้อความเข้า MCS51 โดยผ่านการพิมพ์จากคีบบอร์ด โดยเมื่อรับข้อความได้แล้วก็จะ ไปแสดงผลที่จอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การทดสอบการถอดรหัส PDU

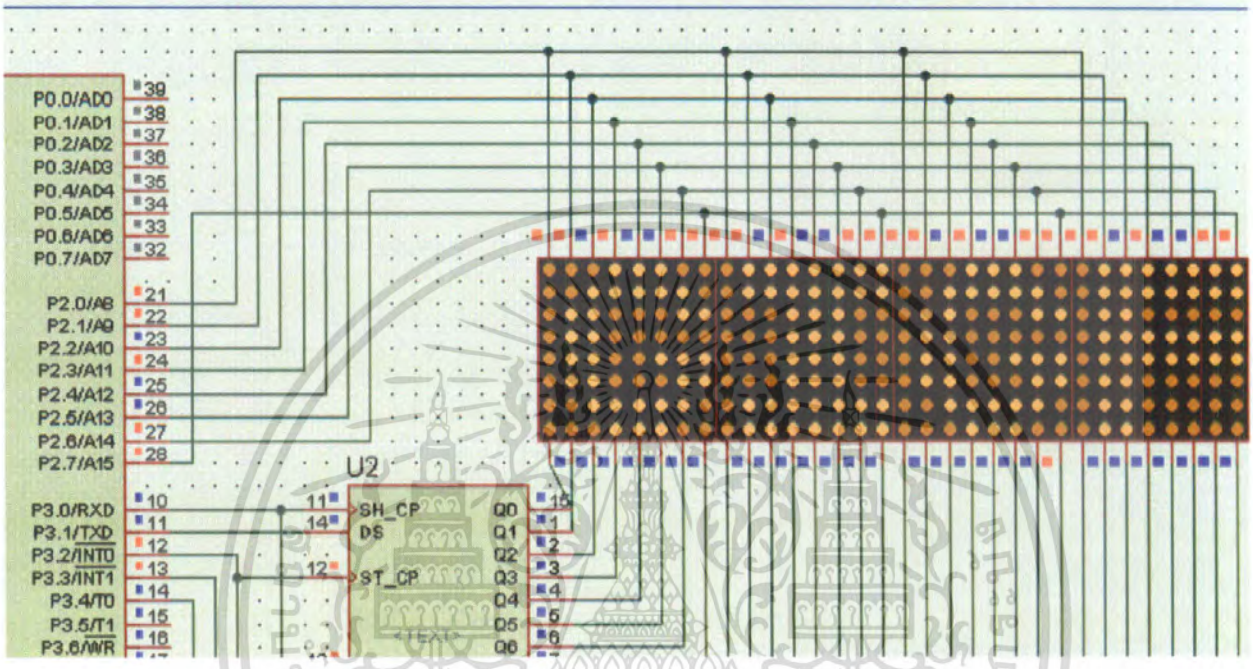


รูปที่ 4.2 การทดสอบการถอดรหัส PDU

จากรูปจะเป็นการทดสอบการถอดรหัสโดยจะทำการเก็บค่า PDU ที่ได้ไว้ ใน Array ตัวนึงก่อนซึ่งค่าที่เป็นส่วน Message จริงๆจะมีตั้งแต่ ตำแหน่งที่ 52 ขึ้นไป เมื่อเก็บค่านั้นมาได้แล้วเราก็จะทำการถอดรหัสตามทฤษฎีบทที่ 2 ที่ได้กล่าวไปแล้ว แล้วก็นำค่าที่ได้ไปแสดงผลผ่านจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การทดสอบการแสดงผลทาง LED

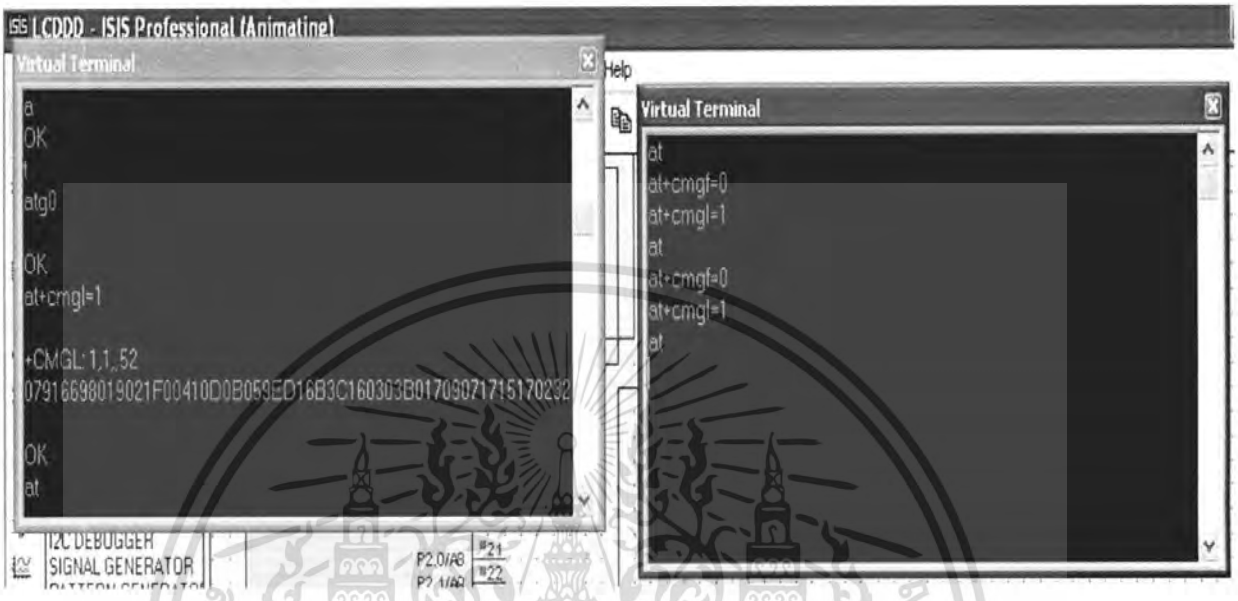


รูปที่ 4.3 การทดลองการแสดงผลทาง LED

จากรูปเราจะใช้ Port 2 เป็นตัวกำหนดหลัก และ Port 3 จะเป็นการกำหนดแถวซึ่งจะมี 74HC595 ในการ Shift bit ทีละตัวซึ่งดูได้จากรูปที่ส่วนล่างของ Dot Matrix จะมีไฟสีแดงวิ่งไปอย่างรวดเร็วจนเกิดภาพติดตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

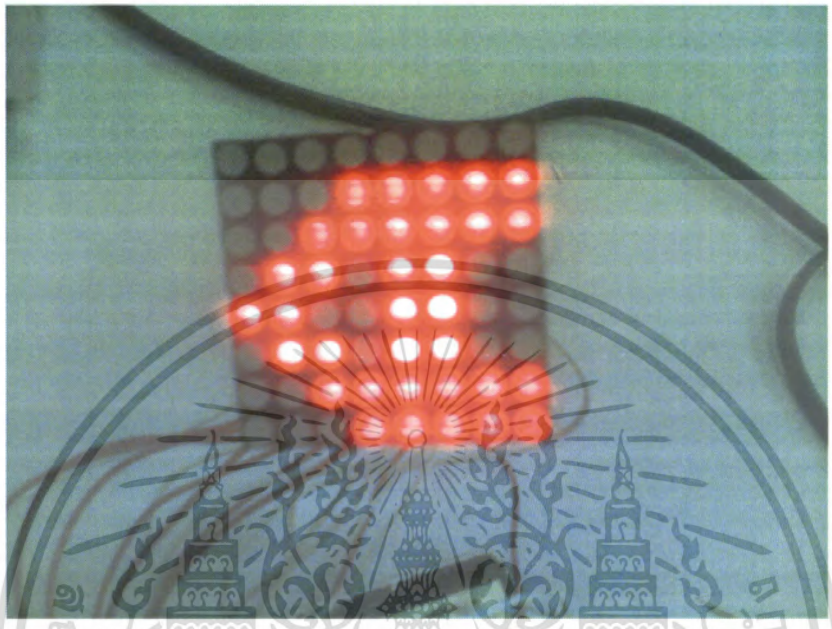
4.1.4 การทดสอบโดยการเชื่อมต่อกับมือถือผ่านอินฟาเรด



รูปที่ 4.4 การทดลองส่ง AT COMMAND ให้กับมือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

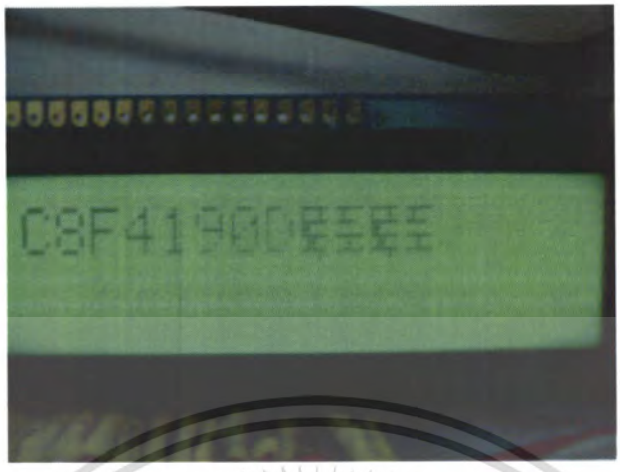
4.2 การทดสอบกับการต่อวงจรของจริง



รูปที่ 4.5 การแสดงผล LED จากการทดสอบจริง

รูปที่ 4.6 การแสดงผลทาง LCD โดยแสดงส่วนที่เป็นตัวแรกหลังจากการส่ง AT COMMAND ให้แสดง SMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

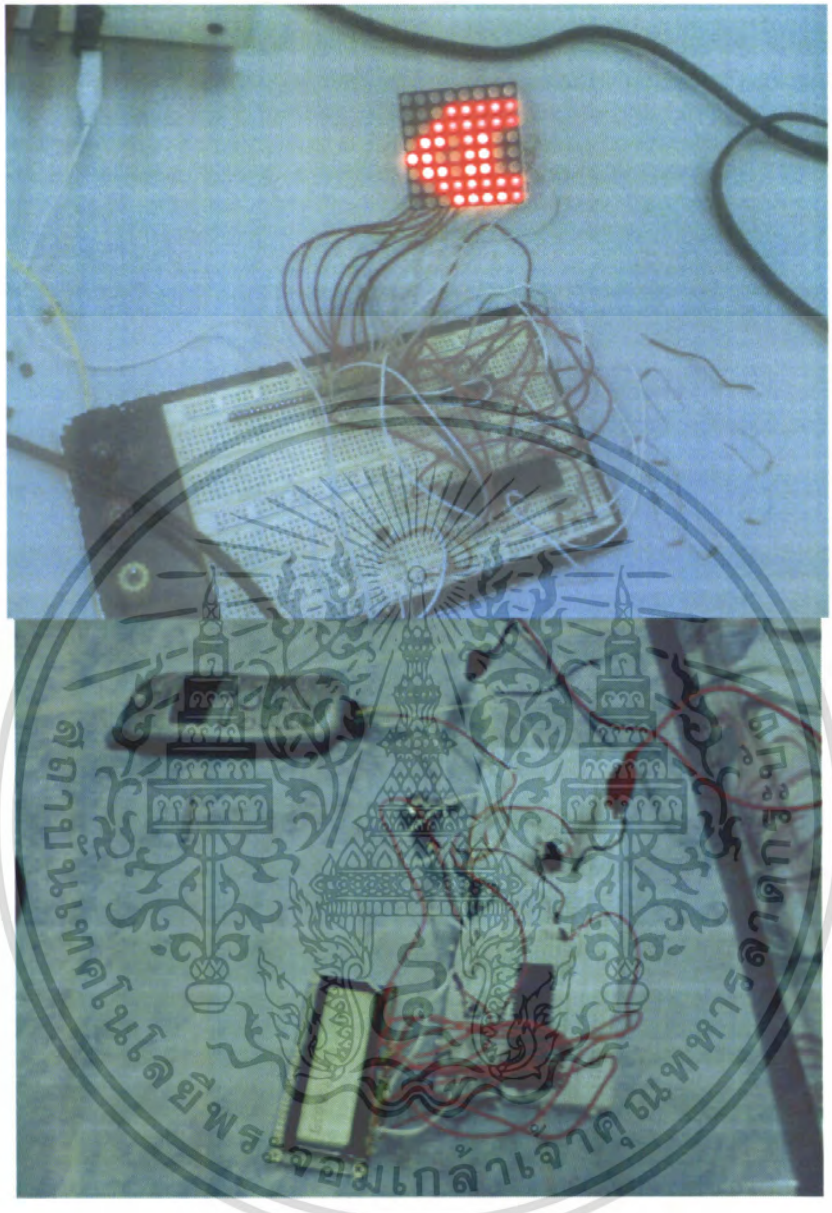


รูปที่ 4.7 การแสดงผลทาง LCD โดยแสดงส่วนที่เป็นรหัส PDU

โดยในการทดลองนี้จะเป็นรหัสของคำว่า "High" เลยทำให้มีรหัส PDU แค่ 8 ตัว และจะสังเกตได้อีกอย่างคือหลังจากที่มีการแสดงรหัส PDU ของ SMS ครบแล้วมันจะตามด้วยอักษรประหลาดๆ ซึ่งมันคือคำว่า "OK"

รูปที่ 4.8 หลังจากที่ถอดรหัส PDU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ภาพโดยรวมการแสดงผล LED และ การแสดงผลทาง LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.10 ประกอบวงจรลงกล่อง

รูปที่ 4.11 ทีเด็ดของวงจรทำแล้วลด noise ได้อย่างดี

รูปที่ 4.10 ทีมงานและผลงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.ศิวะพงศ์ เพ็ชรสงค์.(กันยายน,2546).ถอดรหัส SMS เรื่องลับที่ไม่ลับอีกต่อไป. PC MAGAZINE, หน้า 176-181.

2.อุดม รานอก. "ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์" , infopress , 296 หน้า , 2548

3.ประจิน พลังสันติกุล,ชัยวัฒน์ ลิ้มพรจิตรวิไล. " ไมโครคอรโทรลเลอร์ MCS-51 กับ Keil C51" .Inex, 367 หน้า ,2521

4. Keil programming . Keil.

Retrieved August 1 , 2007, from www.keil.org

5.Webboard for electronic people

Retrieved April 14 , 2007, from www.electoday.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

```
#include<reg52.h>
#include <stdio.h>
char sms[40];
char pdu[20];
char temp[100];
char gab[20];
int a,b,i,j,m,x,y,z,t ,b,time,d,t,c,next,d1,e,num;
int z1,z2,z3,z4,x1,x2,x3,y1,y2;
sbit SH=P1^0;
sbit DS=P1^1;
sbit ST=P1^2;
sbit MR=P1^3;
sbit OE=P1^4;
void delay(int tick)
{
        int i,j;
        for(i=0; i<tick; i++)
        for(j=0; j<250;j++);
}
void delay2()
{
int i,j;
for(i=0;i<5;i++)
for(j=0;j<10;j++)
i=i+0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void tid()
{
        OE=0;  delay2();
        ST=0;
        SH=1;
        SH=0;
        OE=1;
        ST=1;
}
void SHOWALL()
{
        next=1;
        time=time+1;
        while (next!=56)
        {
                P2=temp[t+next];
                tid();
                next=next+1;
        }

}
void SHOW_a()
{
temp[b]=0x00;
b++;
temp[b]=0x20;
b++;
temp[b]=0x54;
b=b+1;
temp[b]=0x54;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x78;
b=b+1;
}
void     SHOW_b()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x38;
b=b+1;
}
void SHOW_c()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x38;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x28;
b=b+1;
}
void SHOW_d()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x38;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x7F;
b=b+1;
}
void SHOW_e()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x38;
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x54;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x58;
b=b+1;
}

void SHOW_f()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x7E;
b=b+1;
temp[b]=0x05;
b=b+1;
temp[b]=0x01;
b=b+1;
}
void SHOW_g()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x98;
b=b+1;
temp[b]=0xA4;
b=b+1;
temp[b]=0xA4;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0xA4;
b=b+1;
temp[b]=0x78;
b=b+1;
}
void SHOW_h()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x78;
b=b+1;
}
void SHOW_i()
{

temp[b]=0x00;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x7D;
b=b+1;
temp[b]=0x40;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
}
void SHOW_j()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x80;
b=b+1;
temp[b]=0x84;
b=b+1;
temp[b]=0x7D;
b=b+1;
}
void SHOW_k()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x10;
b=b+1;
temp[b]=0x28;
b=b+1;
temp[b]=0x44;
b=b+1;
}
void SHOW_l()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x00;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x40;
b=b+1;
}
void SHOW_m()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7C;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x7C;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x78;
b=b+1;
}
void SHOW_n()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7C;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x04;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x78;
b=b+1;
}
void SHOW_o()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x38;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x38;
b=b+1;
}
void SHOW_p()
{
temp[b]=0x00;
b=b+1;
temp[b]=0xFC;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x24;
b=b+1;
temp[b]=0x24;
b=b+1;
temp[b]=0x24;
b=b+1;
temp[b]=0x18;
b=b+1;
}
void SHOW_q()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x18;
b=b+1;
temp[b]=0x24;
b=b+1;
temp[b]=0x24;
b=b+1;
temp[b]=0x24;
b=b+1;
temp[b]=0xFC;
b=b+1;
}
void SHOW_r()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x04;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x78;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x08;
}
void SHOW_s()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x48;
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x24;
b=b+1;
}
void SHOW_t()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x3f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x24;
b=b+1;
}
void SHOW_u()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3C;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x7C;
b=b+1;
}
void SHOW_v()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x0C;
b=b+1;
temp[b]=0x30;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x30;
b=b+1;
temp[b]=0x0C;
b=b+1;
}
void SHOW_w()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3C;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x3C;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x3C;
b=b+1;
}
void SHOW_x()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x28;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x10;
b=b+1;
temp[b]=0x28;
b=b+1;
temp[b]=0x44;
b=b+1;
}
void SHOW_y()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x1C;
b=b+1;
temp[b]=0xA0;
b=b+1;
temp[b]=0xA0;
b=b+1;
temp[b]=0xA0;
b=b+1;
temp[b]=0x7C;
b=b+1;
}
void SHOW_z()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x64;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x54;
b=b+1;
temp[b]=0x4C;
b=b+1;
temp[b]=0x44;
b=b+1;
}

void SHOWA    ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7E;
b=b+1;
temp[b]=0x11;
b=b+1;
temp[b]=0x11;
b=b+1;
temp[b]=0x11;
b=b+1;
temp[b]=0x7E;
b=b+1;
}
void SHOWB ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x11;
b=b+1;
temp[b]=0x11;
b=b+1;
temp[b]=0x11;
b=b+1;
temp[b]=0x7E;
b=b+1;
}
void SHOWC()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3E;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x22;
b=b+1;
}
void SHOWD()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x41;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x22;
b=b+1;
temp[b]=0x1C;
b=b+1;
}
void SHOWE()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x41;
b=b+1;
}
void SHOWF()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x09;
b=b+1;
temp[b]=0x09;
b=b+1;
temp[b]=0x09;
b=b+1;
temp[b]=0x01;
b=b+1;
}
void SHOWG()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3E;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x7A;
b=b+1;
}
void SHOWH()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x08;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x7F;
b=b+1;
}
void SHOWI()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x41;
b=b+1;
}
void SHOWJ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x30;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x41;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x3F;
b=b+1;
temp[b]=0x01;
b=b+1;
}
void SHOWK()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x14;
b=b+1;
temp[b]=0x22;
b=b+1;
temp[b]=0x41;
b=b+1;
}
void SHOWL()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
}
void SHOWM()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x02;
b=b+1;
temp[b]=0x0C;
b=b+1;
temp[b]=0x02;
b=b+1;
temp[b]=0x7F;
b=b+1;
}
void SHOWN()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x08;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x10;
b=b+1;
temp[b]=0x7F;
b=b+1;
}
void SHOWO()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3C;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x3C;
b=b+1;
}
void SHOWP()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7F;
b=b+1;
temp[b]=0x09;
b=b+1;
temp[b]=0x09;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x09;
b=b+1;
temp[b]=0x06;
b=b+1;
}
void SHOWQ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0xBE;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x61;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x3E;
b=b+1;
}
void SHOWR()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x7f;
b=b+1;
temp[b]=0x09;
b=b+1;
temp[b]=0x19;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x29;
b=b+1;
temp[b]=0x46;
b=b+1;
}
void SHOWS()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x26;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x32;
b=b+1;
}
void SHOWT()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x01;
b=b+1;
temp[b]=0x01;
b=b+1;
temp[b]=0x7F;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x01;
b=b+1;
temp[b]=0x01;
b=b+1;
}
void SHOWU()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3F;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x3F;
b=b+1;
}
void SHOWV()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x1F;
b=b+1;
temp[b]=0x20;
b=b+1;
temp[b]=0x40;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x20;
b=b+1;
temp[b]=0x1F;
b=b+1;
}
void SHOWW()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3F;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x30;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x3F;
b=b+1;
}
void SHOWX()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x63;
b=b+1;
temp[b]=0x14;
b=b+1;
temp[b]=0x08;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x14;
b=b+1;
temp[b]=0x63;
b=b+1;
}
void SHOWY()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x07;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x70;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x07;
b=b+1;
}
void SHOWZ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x61;
b=b+1;
temp[b]=0x51;
b=b+1;
temp[b]=0x49;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x45;
b=b+1;
temp[b]=0x43;
b=b+1;
}
void SHOWNULL ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
}
void SHOWNULLS ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
temp[b]=0x00;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void SHOW ()
{
temp[b]=0x01;
b=b+1;
temp[b]=0x02;
b=b+1;
temp[b]=0x04;
b=b+1;
temp[b]=0x08;
b=b+1;
temp[b]=0x10;
b=b+1;
temp[b]=0x20;
b=b+1;
temp[b]=0x40;
b=b+1;
temp[b]=0x80;
b=b+1;
}
void SHOW1 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x44;
b=b+1;
temp[b]=0x42;
b=b+1;
temp[b]=0x7f;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x40;
b=b+1;
temp[b]=0x40;
b=b+1;
}
void SHOW2 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x62;
b=b+1;
temp[b]=0x51;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x46;
b=b+1;
}
void SHOW3 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x22;
b=b+1;
temp[b]=0x41;
b=b+1;
temp[b]=0x49;
b=b+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp[b]=0x49;
b=b+1;
temp[b]=0x36;
b=b+1;
}

void SHOW4 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x18;
b=b+1;
temp[b]=0x14;
b=b+1;
temp[b]=0x12;
b=b+1;
temp[b]=0x7f;
b=b+1;
temp[b]=0x10;
b=b+1;
}
void SHOW5 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x27;
b=b+1;
temp[b]=0x45;
b=b+1;
temp[b]=0x45;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x45;
b=b+1;
temp[b]=0x39;
b=b+1;
}
void SHOW6 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x38;
b=b+1;
temp[b]=0x4c;
b=b+1;
temp[b]=0x4a;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x31;
b=b+1;
}
void SHOW7 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x01;
b=b+1;
temp[b]=0x71;
b=b+1;
temp[b]=0x09;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x05;
b=b+1;
temp[b]=0x03;
b=b+1;
}
void SHOW8 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x36;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x36;
b=b+1;
}
void SHOW9 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x06;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x49;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=b+1;
temp[b]=0x29;
b=b+1;
temp[b]=0x1e;
b=b+1;
}
void SHOW0 ()
{
temp[b]=0x00;
b=b+1;
temp[b]=0x3E;
b=b+1;
temp[b]=0x51;
b=b+1;
temp[b]=0x49;
b=b+1;
temp[b]=0x45;
b=b+1;
temp[b]=0x3e;
b=b+1;
temp[b]=0x00;
b=b+1;
}
void check()
{

        if      (gab[i]=='/')
        {
                SHOW();
        }else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if
        (gab[i]=='A')
{
        SHOWA();
}else
if      (gab[i]=='B')
{
        SHOWB();
}else
if      (gab[i]=='C')
{
        SHOWC();
}else
if      (gab[i]=='D')
{
        SHOWD();
}else
if      (gab[i]=='E')
{
        SHOWE();
}else
if      (gab[i]=='F')
{
        SHOWF();
}else
if      (gab[i]=='G')
{
        SHOWG();
}else
if      (gab[i]=='H')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        SHOWH();
}else
if      (gab[i]=='I')
{
        SHOWI();
}else
if      (gab[i]=='J')
{
        SHOWJ();
}else
if      (gab[i]=='K')
{
        SHOWK();
}else
if      (gab[i]=='L')
{
        SHOWL();
}else
if      (gab[i]=='M')
{
        SHOWM();
}else
if      (gab[i]=='N')
{
        SHOWN();
}else
if      (gab[i]=='O')
{
        SHOWO();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}else
if      (gab[i]=='P')
{
        SHOWP();
}else
if      (gab[i]=='Q')
{
        SHOWQ();
}else
if      (gab[i]=='R')
{
        SHOWR();
}else
if      (gab[i]=='S')
{
        SHOWF();
}else
if      (gab[i]=='T')
{
        SHOWT();
}else
if      (gab[i]=='U')
{
        SHOWU();
}else
if      (gab[i]=='V')
{
        SHOWV();
}else
if      (gab[i]=='W')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        SHOWW();
}else
if      (gab[i]=='X')
{
        SHOWX();
}else
if      (gab[i]=='Y')
{
        SHOWY();
}else
if      (gab[i]=='Z')
{
        SHOWZ();
}else
if      (gab[i]==' ')
{
        SHOWNULL();
}else
if      (gab[i]=='/')
{
        SHOW();
}else
if
        (gab[i]=='a')
{
        SHOW_a();
}else
if      (gab[i]=='b')
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SHOW_b();
}else
if          (gab[i]=='c')
{
            SHOW_c();
}else
if          (gab[i]=='d')
{
            SHOW_d();
}else
if          (gab[i]=='e')
{
            SHOW_e();
}else
if          (gab[i]=='f')
{
            SHOW_f();
}else
if          (gab[i]=='g')
{
            SHOW_g();
}else
if          (gab[i]=='h')
{
            SHOW_h();
}else
if          (gab[i]=='i')
{
            SHOW_i();
}else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if      (gab[i]=='j')
{
        SHOW_j();
}else
if      (gab[i]=='k')
{
        SHOW_k();
}else
if      (gab[i]=='l')
{
        SHOW_l();
}else
if      (gab[i]=='m')
{
        SHOW_m();
}else
if      (gab[i]=='n')
{
        SHOW_n();
}else
if      (gab[i]=='o')
{
        SHOW_o();
}else
if      (gab[i]=='p')
{
        SHOW_p();
}else
if      (gab[i]=='q')
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SHOW_q();
}else
if          (gab[i]=='r')
{
            SHOW_r();
}else
if          (gab[i]=='s')
{
            SHOW_s();
}else
if          (gab[i]=='t')
{
            SHOW_t();
}else
if          (gab[i]=='u')
{
            SHOW_u();
}else
if          (gab[i]=='v')
{
            SHOW_v();
}else
if          (gab[i]=='w')
{
            SHOW_w();
}else
if          (gab[i]=='x')
{
            SHOW_x();
}else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if      (gab[i]=='y')
{
        SHOW_y();
}else
if      (gab[i]=='z')
{
        SHOW_z();
}else
if      (gab[i]==' ')
{
        SHOWNULL();
}else
if      (gab[i]=='1')
{
        SHOW1();
}else
if      (gab[i]=='2')
{
        SHOW2();
}else
if      (gab[i]=='3')
{
        SHOW3();
}else
if      (gab[i]=='4')
{
        SHOW4();
}else
if      (gab[i]=='5')
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SHOW5();
        }else
        if      (gab[i]=='6')
        {
                SHOW6();
        }else
        if      (gab[i]=='7')
        {
                SHOW7();
        }else
        if      (gab[i]=='8')
        {
                SHOW8();
        }else
        if      (gab[i]=='9')
        {
                SHOW9();
        }else
        if      (gab[i]=='0')
        {
                SHOW0();
        }else
        SHOW();
}


void decode_pdu()
        {
        if (sms[t] == '0')
                x1 = 0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if   (sms[t] == '1')
            x1 = 0x10;
else if   (sms[t] == '2')
            x1 = 0x20;
else if   (sms[t] == '3')
            x1 = 0x30;
else if   (sms[t] == '4')
            x1 = 0x40;
else if   (sms[t] == '5')
            x1 = 0x50;
else if   (sms[t] == '6')
            x1 = 0x60;
else if   (sms[t] == '7')
            x1 = 0x70;
else if   (sms[t] == '8')
            x1 = 0x80;
else if   (sms[t] == '9')
            x1 = 0x90;
else if   (sms[t] == 'A')
            x1 = 0xA0;
else if   (sms[t] == 'B')
            x1 = 0xB0;
else if   (sms[t] == 'C')
            x1 = 0xC0;
else if   (sms[t] == 'D')
            x1 = 0xD0;
else if   (sms[t] == 'E')
            x1 = 0xE0;
else if   (sms[t] == 'F')
            x1 = 0xF0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (sms[a] == '0')
        x2 = 0x00;
else if   (sms[a] == '1')
        x2 = 0x01;
else if   (sms[a] == '2')
        x2 = 0x02;
else if   (sms[a] == '3')
        x2 = 0x03;
else if   (sms[a] == '4')
        x2 = 0x04;
else if   (sms[a] == '5')
        x2 = 0x05;
else if   (sms[a] == '6')
        x2 = 0x06;
else if   (sms[a] == '7')
        x2 = 0x07;
else if   (sms[a] == '8')
        x2 = 0x08;
else if   (sms[a] == '9')
        x2 = 0x09;
else if   (sms[a] == 'A')
        x2 = 0x0A;
else if   (sms[a] == 'B')
        x2 = 0x0B;
else if   (sms[a] == 'C')
        x2 = 0x0C;
else if   (sms[a] == 'D')
        x2 = 0x0D;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else if   (sms[a] == 'E')
                  x2 = 0x0E;
        else if   (sms[a] == 'F')
                  x2 = 0x0F;
}
void output1()
{
        t=1;
        a=2;
        j=1;
        i=1;
        x=1;
        decode_pdu();

        if(       (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                  |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                  |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                  |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1 | x2;
        gab[j] = pdu[i] & 0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[1]=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void output2()
{

        decode_pdu();

        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
                {
        pdu[i] = x1|x2;
        y1 = pdu[i-1] &0x80;
        z1 = y1>>7;
        z1 = z1 &0x01;     // if don't make this it will bug
        z2 = pdu[i]& 0x7f;
        z3 = z2<<1;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
                }
        else
        gab[2]=0x00;

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void output3()
{
        decode_pdu();

        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1] &0xc0;
        z1 = y1>>6;
        z1 = z1 & 0x03;
        z2 = pdu[i]&0x3f;
        z3 = z2<<2;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[3]=0x00;
}
void output4()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                    |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                    |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1] & 0xe0;
        z1 = y1>>5;
        z1 = z1 &0x07;
        z2 = pdu[i] & 0x1f;
        z3 = z2<<3;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[4]=0x00;
}
void output5()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        y1 = pdu[i-1]&0xf0;
        z1 = y1>>4;
        z1 = z1&0x0f;
        z2 = pdu[i]&0x0f;
        z3 = z2<<4;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[5]=0x00;
}
void output6()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1]&0xf8;
        z1 = y1>>3;
        z1 = z1&0x1f;
        z2 = pdu[i]&0x1f;
        z3 = z2<<5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[6]=0x00;
}
void output7()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1]&0xfc;
        z1 = y1>>2;
        z1 = z1&0x3f;
        z2 = pdu[i]&0x3f;
        z3 = z2<<6;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        a = a+2;
        x = x+2;
        }
        else
        gab[7]=0x00;
}
void output8()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1]&0xfe;
        z1 = y1>>1;
        z1 = z1&0x7f;
        z2 = pdu[i]&0x7f;
        z3 = z2<<7;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[8]=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void output9()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1]&0x7f;
        gab[j] = y1;
        i++;
        j++;
        t = t+2;
        a = a+2;
        }
        else
        gab[9]=0x00;
}
void output10()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] &0x80;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        z1 = y1>>7;
        z1 = z1&0x01;
        z2 = pdu[i-1]& 0x7f;
        z3 = z2<<1;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[10]=0x00;
}
void output11()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] &0xc0;
        z1 = y1>>6;
        z1 = z1&0x03;
        z2 = pdu[i-1]&0x3f;
        z3 = z2<<2;
        z4 = z3|z1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[11]=0x00;
}
void output12()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] & 0xe0;
        z1 = y1>>5;
        z1 = z1&0x07;
        z2 = pdu[i-1] & 0x1f;
        z3 = z2<<3;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        x = x+2;
        }
        else
        gab[12]=0x00;
}

void output13()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[a]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2]&0xf0;
        z1 = y1>>4;
        z1 = z1&0x0f;
        z2 = pdu[i-1]&0x0f;
        z3 = z2<<4;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[13]=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void output14()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[a]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2]&0xf8;
        z1 = y1>>3;
        z1 = z1&0x1f;
        z2 = pdu[i-1]&0x1f;
        z3 = z2<<5;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[14]=0x00;
}
void output15()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[a]=='B')|(sms[x]=='C')|(sms[x]=='D')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                    |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                    |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2]&0xfc;
        z1 = y1>>2;
        z1 = z1&0x3f;
        z2 = pdu[i-1]&0x3f;
        z3 = z2<<6;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[15]=0x00;
}
void output16()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[a]=='B')|(sms[x]=='C')|(sms[x]=='D')
                    |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                    |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                    |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        y1 = pdu[i-2]&0xfe;
        z1 = y1>>1;
        z1 = z1&0x7f;
        z2 = pdu[i-1]&0x7f;
        z3 = z2<<7;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[16]=0x00;
}
void output17()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-1]&0x7f;
        gab[j] = y1;
        i++;
        j++;
        t = t+2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        a = a+2;
        }
        else
        gab[17]=0x00;
}
void output18()
{
        decode_pdu();
        if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] &0x80;
        z1 = y1>>7;
        z1 = z1&0x01;
        z2 = pdu[i-1]& 0x7f;
        z3 = z2<<1;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[18]=0x00;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void output19()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
                        |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                        |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] &0xc0;
        z1 = y1>>6;
        z1 = z1&0x03;
        z2 = pdu[i-1]&0x3f;
        z3 = z2<<2;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[19]=0x00;
}
void output20()
{
        decode_pdu();
                if(     (sms[x]=='A')|(sms[x]=='B')|(sms[x]=='C')|(sms[x]=='D')
                        |(sms[x]=='E')|(sms[x]=='F')|(sms[x]=='1')|(sms[x]=='2')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    |(sms[x]=='3')|(sms[x]=='4')|(sms[x]=='5')|(sms[x]=='6')
                    |(sms[x]=='7')|(sms[x]=='8')|(sms[x]=='9')|(sms[x]=='0') )
        {
        pdu[i] = x1|x2;
        y1 = pdu[i-2] & 0xe0;
        z1 = y1>>5;
        z1 = z1&0x07;
        z2 = pdu[i-1] & 0x1f;
        z3 = z2<<3;
        z4 = z3|z1;
        gab[j] = z4&0x7f;
        i++;
        j++;
        t = t+2;
        a = a+2;
        x = x+2;
        }
        else
        gab[20]=0x00;
}
void start (void) {
SCON = 0x52; // set speacial function register
TMOD = 0x20;  /* TMOD: timer 1, mode 2, 8-bit reload     */
TH1 = 0xfd; // Baud Rate = 9600
TR1 = 1;
PCON=0x80;  // x2 baudrate
}
void phone_out(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char text1[3];
unsigned char text2[10];
unsigned char text3[10];
 text1[0]='a';
 text1[1]='t';
 text1[2]=0x0d;

text2[0]='a';
text2[1]='t';
text2[2]='+';
text2[3]='c';
text2[4]='m';
text2[5]='g';
text2[6]='f';
text2[7]='=';
text2[8]='0';
text2[9]=0x0d;

 text3[0]='a';
 text3[1]='t';
 text3[2]='+';
 text3[3]='c';
text3[4]='m';
 text3[5]='g';
 text3[6]='l';
 text3[7]='=';
 text3[8]='0';
 text3[9]=0x0d;

        for(i=0;i<3;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        SBUF=text1[i];

        while(~TI);
        TI=0;
        delay(50);
}
        delay(100);

for(i=0;i<10;i++)
        {
        SBUF=text2[i];

        while(~TI);
        TI=0;
        delay(50);
}
        delay(100);

for(i=0;i<10;i++)
        {
        SBUF=text3[i];

        while(~TI);
        TI=0;
        delay(50);
}
            delay(100);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void main(void)
{

        start();


        phone_out();
        RI=0;


        for(a=1; a<40+14+20-1; a++)         // delete junk
        {
        while(~RI);
        x = SBUF;
        RI=0;
        }
        a=1;
        b=0;
        while(b!=1)          // sms pdu
        {


        while(~RI);
        sms[a] = SBUF;
        RI=0;
                if((sms[a]!='A')&(sms[a]!='B')&(sms[a]!='C')&(sms[a]!='D')
                        &(sms[a]!='E')&(sms[a]!='F')&(sms[a]!='1')&(sms[a]!='2')
                        &(sms[a]!='3')&(sms[a]!='4')&(sms[a]!='5')&(sms[a]!='6')
                        &(sms[a]!='7')&(sms[a]!='8')&(sms[a]!='9')&(sms[a]!='0'))
                                {
                                        b=1;
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        a++;
        }
delay(100);
output1();
output2();
output3();
output4();
output5();
output6();
output7();
output8();
output9();
output10();
output11();
output12();
output13();
output14();
output15();
output16();
output17();
output18();
output19();
output20();
        time=0;
        c=0;
        t=0;
        b=1;
        i=1;
        SHOWNULLS();
        SHOWNULLS();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(gab[i]!=0x00)
{

        check();
        i++;
}
SHOWNULLS();
SHOWNULLS();
while(1)
{
        P2=0x00;
        OE=1;
        P1=0x00;
        MR=1;
        DS=1;
        SH=1;
        SH=0;
        DS=0;


        ST=1;
        OE=0;

        c=1     ;
        SHOWALL();
        if (time>=5)
        {
                time=0;
                t=t+1;
                if (b<t+56)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        {
                                        t=0;
                                        }
                    }
          }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้