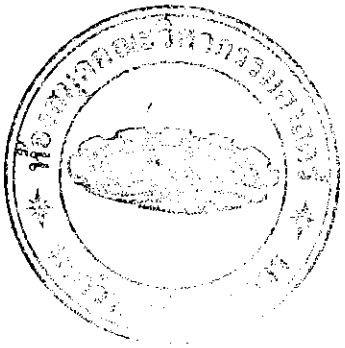
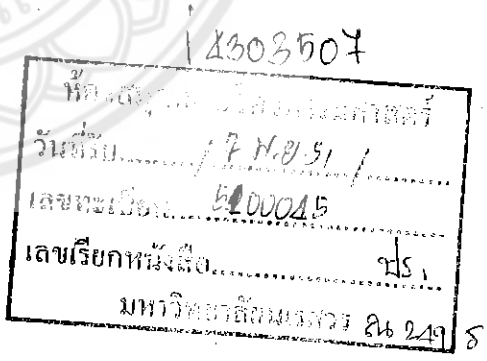


ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

Alert system for mobile application

นายณรงค์ศักดิ์ เทพช่วย รหัส 47361902

นายธีรยุทธ หอมจันทร์ รหัส 47361944



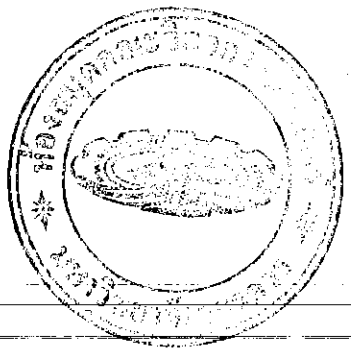
2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

ผู้ดำเนินโครงงาน นายณรงค์ศักดิ์ เทพช่วย รหัส 47361902

นายธีรยุทธ หอมจันทร์ รหัส 47361944

อาจารย์ที่ปรึกษา ดร. ไพศาล มุณีสว่าง

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2551

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร. ไพศาล มุณีสว่าง)

..... กรรมการ

(อาจารย์ ภาณุพงศ์ สอนคม)

..... กรรมการ

(อาจารย์ เศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ			
ผู้ดำเนินงาน	นายณรงค์ศักดิ์	เทพช่วย	รหัส	47361902
	นายธีรยุทธ	หอมจันทร์	รหัส	47361944
อาจารย์ที่ปรึกษา	ดร.ไพศาล	มุณีสว่าง		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2551			

บทคัดย่อ

จากสภาพปัญหาทางเศรษฐกิจในปัจจุบัน ก่อให้เกิดปัญหาทางสังคมเพิ่มมากขึ้น โดยปัญหาที่สำคัญเป็นลำดับต้นๆ คือ ปัญหาการโจรกรรม ทรัพย์สิน โดยการรักษาความปลอดภัยส่วนใหญ่จะเป็นหน้าที่ของเจ้าหน้าที่ตำรวจ ซึ่งมีข้อจำกัดหลายๆ ด้าน

เพื่อความปลอดภัยในชีวิตและทรัพย์สิน ทำให้เกิดแนวคิดในการออกแบบระบบแจ้งเตือนภัยโทรศัพท์มือถือโดยนำความรู้เกี่ยวกับไมโครคอนโทรลเลอร์มาประยุกต์ใช้ในการควบคุมการส่ง SMS ควบคู่ไปกับอุปกรณ์ตรวจจับความเคลื่อนไหว

ผลการดำเนินงานโครงงานวิศวกรรมในครั้งนี้จะได้เครื่องแจ้งเตือนภัยผ่านโทรศัพท์มือถือโดยจะส่ง SMS แจ้งเตือนภัยไปยังเบอร์โทรศัพท์มือถือที่เรากำหนดเมื่อมีการบุกรุกภายในตัวอาคารที่ทำการติดตั้งอุปกรณ์ตรวจจับความเคลื่อนไหว และจะมีการตรวจสอบเป็นระยะๆ เพื่อให้เกิดความน่าเชื่อถือมากขึ้น ซึ่งผู้ใช้สามารถหาวิธีแก้ไขได้ทันเหตุการณ์

Project Title	ALERT SYSTEM FOR MOBILE APPLICATION
Name	Mr.Narongsak Tepchuay ID. 47361902
	Mr.Theerayut Homjan ID. 47361944
Project Advisor	Paisarn Muneesawang, Ph.D.
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic Year	2008

ABSTRACT

Currently, there is the serious problem of economic crisis which reach many social problems especially, criminal therefore, the police have to guard for people in the world.

To have a safety of life and property, it leads to have a new idea for producing alert system. This system is going to send you the SMS via mobile phone by using Microcontroller to invent controller SMS together with infrared motion sensor.

The result of this engineer's project, the owner of that property would get SMS through their mobile phone while somebody try to invade your houses or offices which have installed infrared motion sensor. This system is periodly check to make the customers certain in this system.

กิตติกรรมประกาศ

การจัดทำโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ สำเร็จลุล่วงไปด้วยดีโดยทั้งนี้ คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาโครงงาน ดร. ไพศาล มุณีสว่าง เป็นอย่างสูง ที่ให้คำปรึกษาให้แนวความคิด แนวทางแก้ไขปัญหา ตลอดจนคำแนะนำต่างๆอันเป็นประโยชน์ต่อโครงงานนี้

ขอบคุณพี่รหัสที่คอยช่วยเหลือและแนะนำ ในการจัดทำโครงงานในครั้งนี้

ขอกราบขอบพระคุณบิดา มารดา ญาติพี่น้อง ที่คอยช่วยเหลือเป็นกำลังใจและทุนทรัพย์ที่เอื้อหนุนตลอดมา

ณรงค์ศักดิ์ เทพช่วย

ธีรยุทธ หอมจันทร์

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ก

บทคัดย่อภาษาอังกฤษ..... ข

กิตติกรรมประกาศ..... ค

สารบัญ..... ง

สารบัญตาราง..... ช

สารบัญรูปภาพ..... ซ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญโครงงาน..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 1

1.3 ขอบเขตของโครงงาน..... 1

1.4 ขั้นตอนการดำเนินงาน..... 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 3

1.6 งบประมาณที่ใช้..... 3

บทที่ 2 หลักการและทฤษฎีเบื้องต้น

2.1 หลักการทำงานของการแจ้งเตือนภัยผ่านโทรศัพท์เคลื่อนที่..... 4

2.2 GSM AT Command..... 5

2.3 หลักการรับส่ง SMS ของโทรศัพท์มือถือ..... 5

2.4 โหมดของการรับส่งข้อมูล SMS..... 5

2.5 รูปแบบในการส่งข้อมูลในรูป SMS ผ่าน AT Command..... 5

2.6 กลุ่มคำสั่งในการส่งเอสเอ็มเอส..... 7

2.7 กลุ่มคำสั่งในการควบคุมและดูสถานะของโทรศัพท์มือถือ..... 7

2.8 กลุ่มคำสั่งบริการเครือข่าย..... 7

2.9 การควบคุมการอินเตอร์รัพท์..... 7

2.10 อินเตอร์รัพท์..... 9

2.11 ประเภทของอินเตอร์รัพท์..... 9

สารบัญ(ต่อ)

หน้า

2.12 การจัดลำดับความสำคัญของการอินเตอร์รัพท์ 10
2.13 โปรแกรมให้บริการอินเตอร์รัพท์ 10
2.14 อินเตอร์รัพท์ MCS-51 11
2.15 การสื่อสารข้อมูลผ่านพอร์ตอนุกรม..... 16
2.16 รีจิสเตอร์ควบคุมพอร์ตอนุกรม (SCON) 16
2.17 การกำหนดอัตราบอด (Baud Rate)..... 17
2.18 การรับส่งข้อมูลผ่านพอร์ตอนุกรมด้วยวิธีอินเตอร์รัพท์ (Interrupt)..... 18
2.19 ส่วนควบคุมการส่ง SMS 19
2.20 ส่วนควบคุมการรับ SMS 20

บทที่ 3 วิธีดำเนินงานโครงงานวิศวกรรม
3.1 การวิเคราะห์ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ..... 21
3.2 การออกแบบระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ..... 21
3.3 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ..... 21
3.4 ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือด้านอุปกรณ์..... 22
3.5 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือด้านการโปรแกรม..... 25
3.6 คำอธิบายโปรแกรม..... 28

บทที่ 4 การทดสอบและการวิเคราะห์การทำงาน
4.1 ระบบตรวจจับความเคลื่อนไหว..... 30
4.2 การควบคุมให้ระบบส่งข้อความแจ้งเตือนภัยไปยังหมายเลขปลายทาง..... 31
4.3 ระบบแจ้งสถานะการส่งข้อความบนจอ LCD..... 32
4.4 ระบบ รับ-ส่ง ข้อความ..... 33
4.5 ผลการทดลอง..... 34

สารบัญ(ต่อ)

หน้า

บทที่ 5 บทสรุป

5.1 สรุปผลการดำเนินโครงงาน..... 35

5.2 ปัญหาและแนวทางแก้ไข..... 35

5.3 ข้อจำกัด..... 36

5.4 ข้อเสนอแนะในการพัฒนาต่อไป..... 36

เอกสารอ้างอิง..... 37

ภาคผนวก ก. AT COMMANDS..... 38

ภาคผนวก ข. PROGRAM CODE..... 40

ภาคผนวก ค. การดำเนินงานของโครงงาน..... 46

ประวัติผู้เขียนโครงงาน..... 53

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงหมายเลขอินเตอร์รัพท์ของ MCS-51/52	12
2.2 แสดงลำดับความสำคัญอินเตอร์รัพท์ของ MCS-51/52 ภายในระดับเดียวกัน	14
4.1 แสดงผลการทดลองอุปกรณ์ตรวจจับความเคลื่อนไหวในระยะต่างๆ	34

สารบัญรูปภาพ

รูปที่	หน้า
2.1 แสดงการเชื่อมต่อของอุปกรณ์ที่ใช้ในโครงงาน	4
2.2 ตัวอย่างการเข้ารหัส PDU ของคำว่า ALERT	6
2.3 แสดงการเปรียบเทียบระหว่างวิธี Polling กับวิธี Interrupt	8
2.4 แสดงช่วงเวลาทำงานของซีพียูเมื่อติดต่อกับอุปกรณ์ภายนอก	9
2.5 แสดงขา INT0 และ INT1	11
3.1 แสดงบอร์ดไมโครคอนโทรลเลอร์ เบอร์ P89V51RD2	22
3.2 วงจรไมโครคอนโทรลเลอร์ P89V51RD2	23
3.3 แสดงรูปภาพของอุปกรณ์ตรวจจับความเคลื่อนไหว	24
3.4 แสดงวงจรของอุปกรณ์ตรวจจับความเคลื่อนไหว (sensor)	24
3.5 แสดงโทรศัพท์เคลื่อนที่ Siemens รุ่น C35	25
3.6 แสดงจำนวนพอร์ตต่ออุปกรณ์ภายนอกของโทรศัพท์เคลื่อนที่ที่ใช้งาน	26
3.7 แสดง flow chart ของโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ	27
3.8 แผนผังแสดงขั้นตอนในการพัฒนาโปรแกรมระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ	29
4.1 แสดงรูปของอุปกรณ์ตรวจจับความเคลื่อนไหว	30
4.2 แสดงภาพเมื่อทำการต่ออุปกรณ์ต่างๆ เข้าด้วยกัน	31
4.3 จอ LCD แสดงผลสถานะของอุปกรณ์ตรวจจับความเคลื่อนไหวที่พร้อมใช้งาน	32
4.4 แสดงสถานะการส่งข้อความ	32
4.5 แสดงโทรศัพท์ที่ใช้รับข้อความเมื่อมีการแจ้งเตือนมายังเครื่องรับ	33
4.6 แสดงโทรศัพท์รุ่นซีเมนส์ C 35ที่ใช้ส่งข้อความแจ้งเตือนภัย	33

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยปัจจุบันโทรศัพท์มือถือได้มีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้นเรื่อยๆ ไม่เพียงเฉพาะการติดต่อสื่อสารเท่านั้น แต่ยังมีการพัฒนาในด้านต่างและเป็นเทคโนโลยีที่ไม่หยุดนิ่ง มีการเจริญทางด้าน ฮาร์ดแวร์และซอฟต์แวร์ อยู่ตลอดเวลา เพราะในปัจจุบันผู้คนใช้ชีวิตอย่างเร่งรีบ และปัจจุบันปัญหาด้านการโจรกรรมเริ่มมีปัญหากับชีวิตมากขึ้นเนื่องจากผู้คนต่างต้องออกจากบ้านเพื่อประกอบอาชีพจึงเกิดช่องโหว่ทำให้เป็นช่องทางให้คนร้ายกระทำการต่างๆได้อย่างง่ายดาย และหากเรานำเทคโนโลยีมาใช้ให้เกิดประโยชน์สูงสุดโดยการนำเอาเทคโนโลยีโทรศัพท์เคลื่อนที่เชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์และใช้อุปกรณ์ตรวจจับความเคลื่อนไหว (Sensor) เพื่อดูแลความเรียบร้อยและตรวจจับความผิดปกติภายในตัวอาคารหรือสำนักงานซึ่งจะทำให้มีความปลอดภัยเพิ่มมากขึ้น

ดังนั้นคณะผู้จัดทำจึงได้เล็งเห็นถึงคุณประโยชน์ที่จะได้รับจากการพัฒนาซอฟต์แวร์เพื่อใช้กับโทรศัพท์ มือถือเพื่อให้เกิดประโยชน์สูงสุดและเกิดความคุมค่ากับทรัพยากรที่มีอยู่อย่างจำกัดและเพื่อความปลอดภัยแก่ชีวิตและทรัพย์สิน

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อพัฒนาวงจรอิเล็คทรอนิคส์ในการควบคุมและรับข้อมูลจากฮาร์ดแวร์
2. เพื่อรักษาความปลอดภัยของทรัพย์สินภายในเคหะสถาน
3. เพื่อให้มีความรู้ความเข้าใจในการในการใช้ตัวไมโครคอนโทรเลอร์ MCS-51 ในการเชื่อมต่อกับอุปกรณ์ที่เป็นโทรศัพท์เคลื่อนที่และอุปกรณ์ตรวจจับความเคลื่อนไหว

1.3 เป้าหมายและขอบเขตของโครงงาน

เป้าหมายของโครงงาน

1. เพื่อช่วยในการรักษาความปลอดภัยผ่านระบบโทรศัพท์มือถือ
2. เพื่อช่วยให้ทรัพย์สินปลอดภัยจากเหล่ามิจฉาชีพ
3. เพื่อให้ใช้ทรัพยากรที่มีอยู่ได้อย่างเต็มประสิทธิภาพและเกิดประโยชน์สูงสุด
4. เพื่อได้เรียนรู้เกี่ยวกับเรื่อง AT COMMAND ในการส่ง SMS

ขอบเขตของโครงงาน

1. สามารถตรวจจับความเคลื่อนไหวที่เกิดขึ้นในอาคารได้
2. สามารถส่งข้อความเตือนผ่านโทรศัพท์มือถือได้
3. สามารถดูสถานะการทำงานการส่งข้อความบนหน้าจอ LCD ได้

1.4 ขั้นตอนการดำเนินการ

รายละเอียดขั้นตอนการดำเนินงาน มีดังต่อไปนี้

1. เสนอหัวข้อโครงงาน
2. ศึกษาและรวบรวมเนื้อหาที่จะต้องใช้พัฒนาอุปกรณ์ ศึกษาค้นคว้าหาข้อมูลเกี่ยวกับการเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์ ตระกูล MCS-51
3. ออกแบบและศึกษาระบบการส่ง SMS โดยใช้ไมโครคอนโทรลเลอร์
4. จัดหาแผงวงจรที่เกี่ยวข้องในการทำโปรเจค
5. ออกแบบและเขียนโปรแกรมเพื่อควบคุมการส่ง SMS
6. ต่ออุปกรณ์เข้ากับไมโครคอนโทรลเลอร์
7. ทดสอบและปรับปรุงโปรแกรม ทำการแก้ไขข้อผิดพลาดต่าง ๆ และตรวจสอบการทำงานของโปรแกรมว่าตรงความ ต้องการหรือไม่
8. สรุปผลและทำรายงานการสรุปผล
9. ทำเอกสารประกอบโปรแกรม
10. ทำเอกสารโปรแกรมและคู่มือการใช้

แผนการดำเนินงาน

การดำเนินงาน	2551						
	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
ศึกษาและรวบรวมเนื้อหาที่จะต้องใช้พัฒนาอุปกรณ์							
ออกแบบและศึกษาระบบการส่ง SMS							
จัดหาแผงวงจรที่เกี่ยวข้องในการทำโปรเจค							
ต่ออุปกรณ์เข้ากับไมโครคอนโทรลเลอร์							
ทดสอบและปรับปรุงโปรแกรม							
สรุปผลและทำรายงานการสรุปผล							
ทำเอกสารประกอบโปรแกรม							
รายงานผลการดำเนินงาน							

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มความปลอดภัยในชีวิตและทรัพย์สิน
2. ใช้ทรัพยากรที่มีอยู่ได้เต็มประสิทธิภาพ
3. ส่งเสริมเทคโนโลยีทางด้านโทรศัพท์มือถือ
4. เป็นแนวทางสำหรับบุคคลที่สนใจในการศึกษาค้นคว้าเพิ่มเติมหรือนำไปใช้พัฒนาต่อ

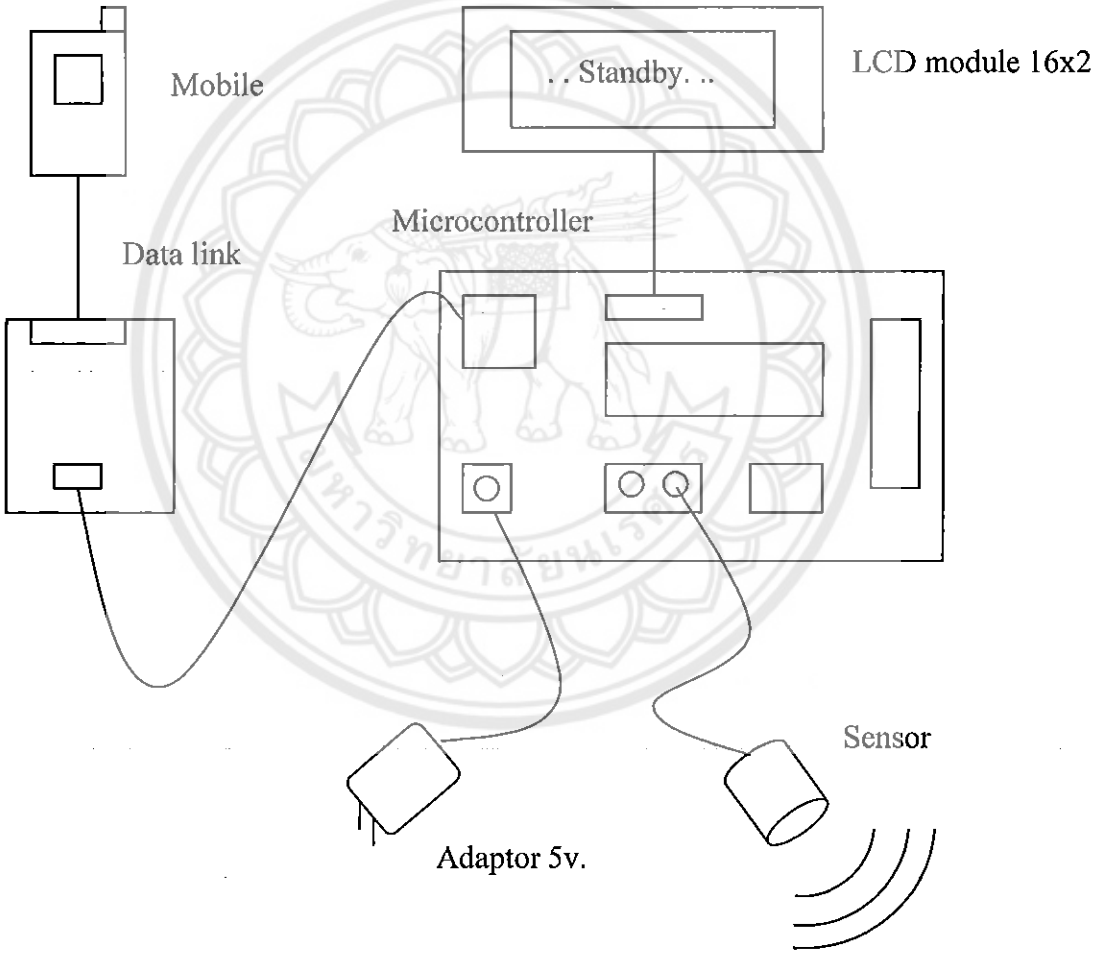
1.6 งบประมาณที่ต้องใช้ในการดำเนินงาน

1. ค่าอุปกรณ์ที่เกี่ยวข้องกับการทำโครงงาน	3,000 บาท
2. ค่าใช้จ่ายเกี่ยวกับเอกสาร	1,000 บาท
รวม	4,000 บาท

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 หลักการทำงานของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

อุปกรณ์ที่ใช้ในโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือโดยประกอบไปด้วย โทรศัพท์มือถือ บอร์ดคาต้าลิงค์ จอ LCD บอร์ดไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 อุปกรณ์ตรวจจับความเคลื่อนไหว (Sensor) และตัวอะแดปเตอร์ (Adapter 5V) ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการเชื่อมต่อของอุปกรณ์ที่ใช้ในโครงงาน

หลักการทำงานคือใช้ Sensor ตรวจจับความเคลื่อนไหวภายในห้อง เมื่อ Sensor ทำงานก็จะส่งสัญญาณเข้าไปที่บอร์ดไมโครคอนโทรลเลอร์ซึ่งถูกโปรแกรมไว้แล้วส่งข้อความไปยังโทรศัพท์เคลื่อนที่ที่ได้กำหนดหมายเลขปลายทางไว้ โดยมีจอ LCD แสดงสถานะการส่งข้อความว่าอยู่ในสถานะใด

ในโครงงานนี้ได้ใช้ข้อมูลเกี่ยวกับโทรศัพท์มือถือบวกกับความรู้ในส่วนของไมโครคอนโทรลเลอร์มาใช้งาน โดยมีหัวข้อสำคัญที่ควรทราบดังนี้

2.2 GSM AT Command [1]

เป็นชุดคำสั่งมาตรฐานที่ใช้ติดต่อสื่อสารกับโทรศัพท์มือถือ โดยส่วนมากมักใช้ในการสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) ในชุดคำสั่งพื้นฐานนั้นบริษัท Hayes ได้เป็นผู้ออกแบบคิดค้นเพื่อใช้กับโมเด็มของตนและต่อมาบริษัทผู้ผลิตมือถือยี่ห้อต่างๆได้พัฒนามาใช้กับผลิตภัณฑ์ของตนเป็นเหตุให้คำสั่งพิเศษบางคำสั่งไม่เหมือนกันในผลิตภัณฑ์ต่างยี่ห้อกัน และความสามารถของโทรศัพท์ในบางรุ่นจะไม่รองรับคำสั่งดังกล่าว เนื่องจากไม่ได้มีวงจรส่วนของโมเด็มบรรจุอยู่ภายใน

2.3 หลักการรับส่ง SMS ของโทรศัพท์มือถือ

SMS ย่อมาจาก Short Massage Service เป็นบริการส่งข้อความสั้นๆจากโทรศัพท์มือถือต้นทางผ่านชุมสายไปยังโทรศัพท์มือถือปลายทาง โดยสามารถส่งได้สูงสุด 160 ตัวอักษรต่อครั้งตามข้อกำหนดมาตรฐานขององค์การ ETSI (European Telecommunications Standards Institute)

2.4 โหมดของการรับส่งข้อมูล SMS [1]

แบ่งออกเป็น 2 โหมดคือ Text Mode และ PDU Mode (Protocol Description Unit Mode)

การส่งข้อความใน Text Mode นั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน(โดยตัวเครื่องเอง)แล้วจึงส่งข้อมูลในรูป PDU Mode อีกครั้งหนึ่ง แต่ในบางเครื่องก็ไม่สนับสนุนการส่งแบบ Text Mode ผ่านทาง AT Command แต่หากเป็น PDU Mode จะสามารถส่งได้ เนื่องจากเครื่องจะไม่ต้องทำอาศัยการแปลงข้อมูลอีกชั้น

2.5 รูปแบบในการส่งข้อมูลในรูป SMS ผ่าน AT Command

มี 2 รูปแบบ คือ Text Mode และ PDU Mode

1. **Text Mode** เป็นการส่งข้อมูลในรูปของตัวอักษรได้โดยตรง ซึ่งตัวเครื่องส่วนใหญ่ไม่รองรับการส่งข้อมูลรูปแบบนี้ผ่านทางเอทีคอมมานด์ จึงไม่สามารถใช้งานได้สมบูรณ์ เนื่องจากการส่งข้อความใน Text Mode นั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน (โดยตัวเครื่องเอง) แล้วจึงส่งข้อมูลในรูป PDU Mode อีกครั้งหนึ่ง แต่ในโทรศัพท์บางเครื่องก็ไม่สนับสนุนการส่งข้อความแบบ Text Mode ผ่านทาง AT Command แต่หากส่งข้อความเป็น PDU Mode จะสามารถส่งได้ เนื่องจากโทรศัพท์จะไม่ต้องมีการแปลงข้อมูลอีกชั้นหนึ่ง

2. PDU Mode PDU (Protocol Data Unit) คือโหมดการทำงานประเภทหนึ่ง ซึ่งจะทำการแปลงรหัสแอสกี (ASCII) ของตัวอักขระแต่ละตัวให้เป็นรหัส PDU ซึ่งรหัส PDU นั้น สามารถนำมาใช้งานได้กับชุดคำสั่ง เอทีคอมมานด์ ในการส่งเอสเอ็มเอส และสามารถใช้ได้กับโทรศัพท์มือถือทุกเครื่องที่รับคำสั่งเอทีคอมมานด์ได้ โดยที่การเข้ารหัส PDU มีขั้นตอนดังนี้

2.1 จะต้องทราบรหัสแอสกีแบบเลขฐาน 16 (Hexadecimal) ของแต่ละอักขระ

2.2 แปลงจากรหัสแอสกีแบบเลขฐาน 16 เป็นรหัสแอสกีแบบเลขฐาน 2 (Binary)

2.3 รหัสแอสกีแบบเลขฐาน 2 มาตัดบิตซ้ายสุดทิ้ง

2.4 แปลงเป็นรหัส PDU โดยนำบิตสุดท้ายของตัวอักขระตัวที่ 2 มาวางหน้า 7 บิตของอักขระตัวที่ 1 ซึ่งจะได้รหัส PDU ของอักขระตัวที่ 1 จากนั้นนำ 2 บิตสุดท้ายของอักขระตัวที่ 3 มาวางหน้า 6 บิตที่เหลืออยู่ของอักขระตัวที่ 2 ซึ่งจะได้รหัส PDU ของอักขระตัวที่ 2 จากนั้นนำ 3 บิตสุดท้ายของอักขระตัวที่ 4 มาวางหน้า 5 บิตที่เหลือของอักขระตัวที่ 3 ซึ่งจะได้รหัส PDU ของอักขระตัวที่ 3 จากนั้นทำตามขั้นตอนเดิมไปเรื่อยๆ จนได้รหัส PDU 8 บิต ของทุกอักขระ

2.5 แปลงรหัสPDU 8 บิตที่ได้ให้เป็นรหัส PDU แบบเลขฐาน 16 การเข้ารหัส PDU ของคำว่า ALERT จะเห็นว่ารหัส PDU ของคำว่าALERT คือ 4166514A05 ดังรูปที่ 2.2

Format	A	L	E	R	T
ASCII Hex	41	4C	45	52	54
ASCII Bin	0100 0001	0100 1100	0100 0101	0101 0010	0101 0100
บิตที่จะเข้ารหัส	100 0001	100 1100	100 0101	101 0010	101 0100

PDU	0100 0001	0110 0110	0101 0001	0100 1010	0000 0101
PDU Hex	41	66	51	4A	05

รูปที่ 2.2 ตัวอย่างการเข้ารหัส PDU ของคำว่า ALERT

2.6 กลุ่มคำสั่งในการส่ง SMS

- AT+CMGF เป็นคำสั่งที่ใช้เลือกรูปแบบของการส่งข้อความ ซึ่งมี 2 โหมด คือ SMS PDU Mode กับ SMS Text Mode

<mode> 0 คือ เลือกใช้ PDU mode

1 คือ เลือกใช้ Text mode

- AT+CMGS เป็นคำสั่งที่ใช้ส่งSMS โดยมีรูปแบบของคำสั่งดังนี้

<da> คือ หมายเลขโทรศัพท์ปลายทางที่จะส่งเอสเอ็มเอสไป

<text is entered> คือ ข้อความที่ต้องการจะส่ง

<Length> คือ ความยาวข้อมูลของข้อความใน PDU mode

<PDU is given> คือ ข้อมูลในรูปแบบรหัส PDU แบบเลขฐาน 16

<CR> คือ ปุ่ม Enter บนคีย์บอร์ด

<CTRL-Z> คือ ปุ่ม Ctrl และ ปุ่ม Z บนแป้นพิมพ์

2.7 กลุ่มคำสั่งในการควบคุมและดูสถานะของโทรศัพท์มือถือ

- AT+CPBS เป็นคำสั่งที่ใช้ในการเลือกที่เก็บหน่วยความจำของสมุดโทรศัพท์ (Select Phone Book Memory Storage)

- AT+CPBR เป็นคำสั่งที่ใช้อ่านข้อมูลจากสมุดโทรศัพท์ (Read Phone Book Entries)

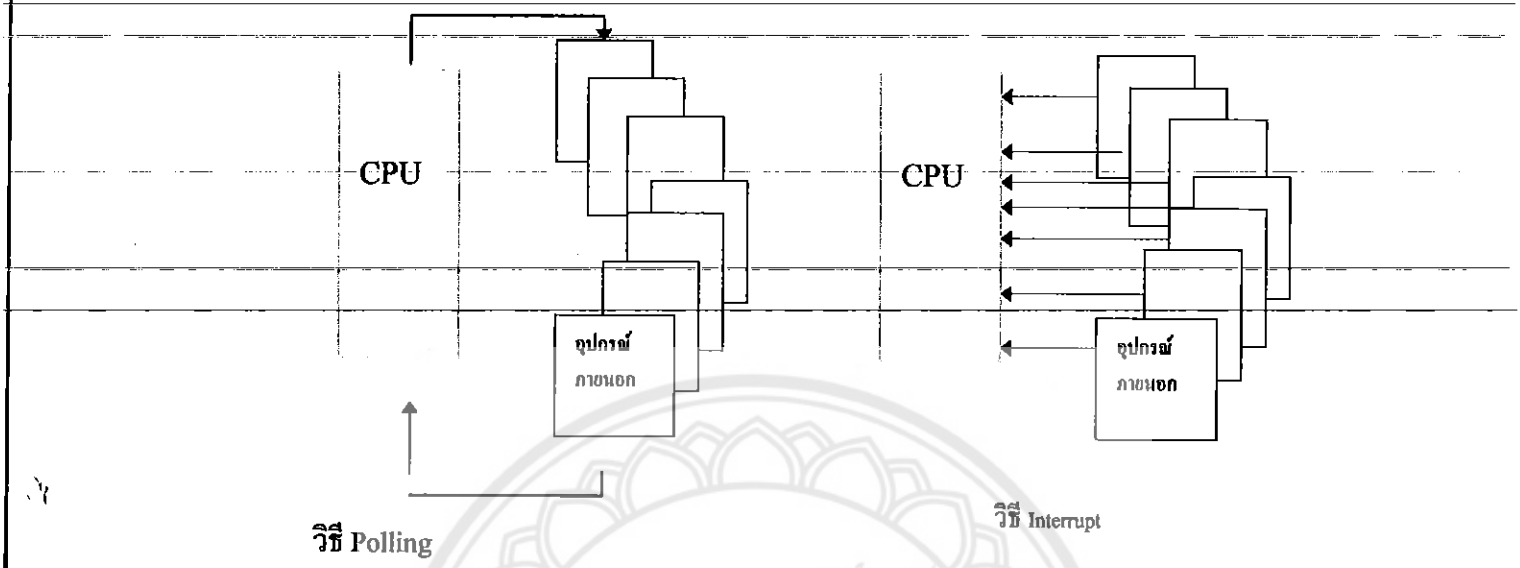
2.8 กลุ่มคำสั่งบริการเครือข่าย

- AT+CLIP เป็นคำสั่งที่ใช้ในการแสดงผลหมายเลขโทรศัพท์ที่โทรเข้า (Calling Line Identification Presentation)

2.9 การควบคุมการอินเตอร์รัพท์ [2]

จากที่เรารู้มาแล้วว่าองค์ประกอบของคอมพิวเตอร์ประกอบด้วย ซีพียู หน่วยความจำ และอุปกรณ์อินพุท เอาท์พุท วิธีที่ซีพียูใช้ติดต่อกับอุปกรณ์ภายนอก มีอยู่ 2 วิธีคือ การโพลลิ่ง (Polling) เป็นวิธีที่ซีพียูต้องคอยตรวจสอบอุปกรณ์อินพุทเอาท์พุทตลอดเวลาว่าอุปกรณ์ตัวไหนบ้างต้องการติดต่อ ทำให้เสียเวลามาก และอีกวิธีคือ การอินเตอร์รัพท์ (Interrupt) เป็นวิธีที่ซีพียูไม่ต้องคอยไปตรวจสอบอุปกรณ์อินพุทเอาท์พุท ถ้าอุปกรณ์ตัวไหนต้องการติดต่อ ก็เพียงแต่ส่งสัญญาณมาบอกซีพียู เรียกว่า สัญญาณร้องขออินเตอร์รัพท์ (Interrupt Request) เมื่อซีพียูได้รับสัญญาณ ก็ทำการตัดสินใจให้บริการอินเตอร์รัพท์ พร้อมกับส่งสัญญาณยอมให้มีการอินเตอร์รัพท์ (Interrupt Acknowledgement) กลับไปยังอุปกรณ์ เมื่ออุปกรณ์ได้รับก็ทำการติดต่อกับซีพียูต่อไป

จากตรงนี้อาจจะยังไม่เห็นภาพข้อแตกต่างระหว่างการโพลลิ่งและการอินเตอร์รัพท์ จึงขอยกตัวอย่างประกอบ แสดงดังรูปที่ 2.3



รูปที่ 2.3 แสดงการเปรียบเทียบระหว่างวิธี Polling กับวิธี Interrupt

อธิบายรูปที่ 2.3

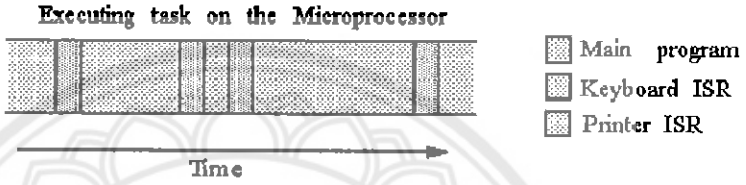
วิธีโพลลิ่ง ซีพียูจะทำการตรวจสอบอุปกรณ์ที่ต้องการติดต่อ โดยเริ่มจากตัวแรกไปเรื่อยๆ จนกระทั่งตัวสุดท้าย ถ้าสมมุติว่าอุปกรณ์ตัวสุดท้ายต้องการติดต่อ ก็จะต้องรอจนกว่าซีพียูจะตรวจสอบมาถึงทำให้อุปกรณ์ตัวนั้นเสียเวลาในการรอ

วิธีอินเตอร์รัพท์ ซีพียูไม่ต้องคอยไปตรวจสอบ ขณะที่ซีพียูกำลังทำงานอื่นอยู่ ถ้าอุปกรณ์ตัวไหนต้องการติดต่อก็จะส่งสัญญาณอินเตอร์รัพท์ไปบอกซีพียู ถ้าซีพียูต้องการติดต่อก็จะส่งสัญญาณตอบกลับไปยังอุปกรณ์ตัวนั้น ทำให้อุปกรณ์ไม่ต้องเสียเวลาในการรอจึงทำงานได้เร็ว และสมมุติว่าซีพียูต้องการติดต่อกับอุปกรณ์ตัวไหน เพียงแต่ส่งสัญญาณไปยังอุปกรณ์ตัวนั้น เช่นต้องการอ่านข้อมูลจากแผ่นดิสก์ ก็จะส่งสัญญาณการอ่านไปยังดิสก์ไดรฟ์ เมื่อไดรฟ์ได้รับสัญญาณแล้วก็จะทำการอ่านข้อมูลจากดิสก์เก็บไว้ใน Buffer ก่อน ช่วงระหว่างนี้ซีพียูสามารถไปทำงานอื่นได้ เมื่อดิสก์ไดรฟ์อ่านข้อมูลเสร็จ ก็จะส่งสัญญาณอินเตอร์รัพท์ไปบอกซีพียูให้มารับข้อมูลจาก Buffer ได้ จะเห็นว่าวิธีอินเตอร์รัพท์ จะช่วยให้ซีพียูทำงานสอดประสานกับกระบวนการอินพุทเอาท์พุทต่างๆได้ง่าย ถ้าปราศจากอินเตอร์รัพท์แล้ว หลายๆกระบวนการจะไม่สามารถทำงานได้ หรือทำได้ยากมาก

2.10 อินเตอร์รัพท์

อินเตอร์รัพท์ คือเหตุการณ์ที่ไม่ปกติที่ต้องการให้ซีพียูหยุดการประมวลผลโปรแกรมที่ทำงานปัจจุบัน และไปทำงานบางอย่างที่เกี่ยวข้องกับเหตุการณ์ไม่ปกตินั้น หรือพูดง่ายๆ คือซีพียูกำลังทำงานบางอย่างอยู่ แต่อยู่ๆก็มีอุปกรณ์ภายนอกเข้ามาขอแทรกการทำงาน หรือขัดจังหวะการทำงาน ซีพียูก็ต้องพิจารณาว่าจะยอมให้อุปกรณ์นั้นขัดจังหวะหรือไม่ ถ้ายอมซีพียูก็ไปทำงานตามอุปกรณ์นั้น

ตัวอย่าง เครื่องคอมพิวเตอร์มีอุปกรณ์อินพุท คือ เครื่องพิมพ์ และอุปกรณ์เอาท์พุท คือ คีย์บอร์ด และอินเตอร์รัพท์เข้ามาช่วยทำงานได้อย่างไร พิจารณาการทำงานดังรูปที่ 2.4



รูปที่ 2.4 แสดงช่วงเวลาทำงานของซีพียูเมื่อติดต่อกับอุปกรณ์ภายนอก

จากรูปที่ 2.4 ซีพียูทำงานโปรแกรมหลักไปเรื่อยๆ และเมื่อมีคีย์บอร์ดอินเตอร์รัพท์เข้ามา (ผู้ใช้กดปุ่มคีย์บอร์ด) ซีพียูก็จะพักการทำงานโปรแกรมหลักไว้ก่อน และกระโดดไปยังโปรแกรมให้บริการอินเตอร์รัพท์คีย์บอร์ด (Keyboard ISR) เมื่อทำงานเสร็จซีพียูก็มาทำงานโปรแกรมหลักต่อ และเมื่อต้องการพิมพ์งานซีพียูก็จะส่งข้อมูลไปให้เครื่องพิมพ์ ขณะที่เครื่องพิมพ์กำลังพิมพ์งานอยู่ ซีพียูก็กลับมาทำงานโปรแกรมหลักต่อได้ เมื่อเครื่องพิมพ์ทำงานเสร็จก็ส่งสัญญาณอินเตอร์รัพท์มาบอกซีพียู และซีพียูก็หยุดทำงานโปรแกรมหลักและกระโดดไปยังโปรแกรมให้บริการอินเตอร์รัพท์เครื่องพิมพ์ (Printer ISR) เพื่อจัดการเกี่ยวกับเครื่องพิมพ์ เมื่อเสร็จก็กลับไปทำงานยังโปรแกรมหลักต่อไป

2.11 ประเภทของอินเตอร์รัพท์

อินเตอร์รัพท์แบ่งออกเป็น 2 ประเภท คือ maskable interrupt และ non-maskable interrupt Maskable interrupt คืออินเตอร์รัพท์ที่ซีพียูสามารถที่จะยกเลิกได้ ส่วน Non-maskable interrupt คืออินเตอร์รัพท์ที่ซีพียูไม่สามารถยกเลิกได้ ซึ่งต้องตอบสนองทันทีทันใด

เราสามารถกำหนดว่าจะให้ซีพียูทำงานอินเตอร์รัพท์แบบ Mask able interrupt หรือ non-maskable interrupt โดยการเซตบิต enable interrupt เมื่ออินเตอร์รัพท์ enable หรือบิตถูกเซต ซีพียูต้องให้บริการอินเตอร์รัพท์นั้น และถ้าอินเตอร์รัพท์ disable หรือบิตเคลียร์ ซีพียูจะ ก็จะยกเลิกอินเตอร์รัพท์นั้น

2.12 การจัดลำดับความสำคัญของการอินเตอร์รัพท์

ในกรณีที่มีอินเตอร์รัพท์เข้าหลายๆ อินเตอร์รัพท์พร้อมกัน แต่ซีพียูจะให้บริการเพียงอินเตอร์รัพท์เดียวเท่านั้น การที่ซีพียูจะให้อินเตอร์รัพท์ไหนทำงานก่อน จำเป็นต้องมีการจัดลำดับความสำคัญของอินเตอร์รัพท์ (Interrupt Priority) ตัวไหนจะมีลำดับความสำคัญสูงสุด ก็จะได้ก่อน ตัวอื่นก็จะได้ตามลำดับลงมา

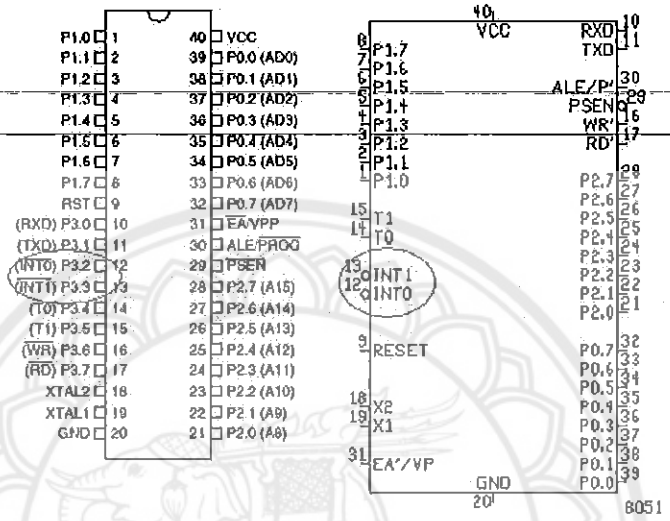
2.13 โปรแกรมให้บริการอินเตอร์รัพท์ [2]

ซีพียูจะเตรียมให้บริการสำหรับการอินเตอร์รัพท์ โดยการประมวลผลโปรแกรมที่เรียกว่า interrupt Service Routine (ISR) หรือโปรแกรมให้บริการอินเตอร์รัพท์ ขั้นตอนการอินเตอร์รัพท์ มีดังนี้

1. เก็บค่า PC ลง stack
2. เก็บสถานะของซีพียูลง stack (MCS-51 ไม่มีขั้นตอนนี้)
3. ตรวจสอบว่าเป็นอินเตอร์รัพท์แบบไหน
4. หาแอดเดรสเริ่มต้นของ ISR โดยไปอ่านจาก Interrupt Vector
5. ประมวลผลคำสั่งที่ ISR
6. คืนค่าสถานะซีพียูจาก stack (MCS-51 ไม่มีขั้นตอนนี้)
7. คืนค่า PC จาก stack
8. เริ่มขั้นตอนอินเตอร์รัพท์ใหม่

2.14 อินเตอร์รัพท์ MCS-51 [2]

ไมโครคอนโทรลเลอร์ MCS-51 มีการใช้งานอินเตอร์รัพท์อยู่ 2 ประเภทคือ อินเตอร์รัพท์จากภายนอก (External interrupt) จะเกิดเมื่อวงจรจากภายนอกส่งสัญญาณไปยังขาอินเตอร์รัพท์ของ MCS-51 ได้แก่ ขา INT0 และ INT1 และอินเตอร์รัพท์ภายใน (Internal interrupt) จะกำเนิดโดยฮาร์ดแวร์ที่อยู่ภายในชิป ได้แก่ไทเมอร์ 0 ไทเมอร์ 1 และพอร์ตสื่อสารอนุกรม แสดงดังรูปที่ 2.5



รูปที่ 2.5 แสดงขา INT0 และ INT1

เมื่ออินเตอร์รัพท์เกิดขึ้น ไมโครคอนโทรลเลอร์จะหยุดทำงานโปรแกรมปัจจุบัน และไปให้บริการอินเตอร์รัพท์ที่เข้ามา โดยการกระโดดไปยังตำแหน่งของโปรแกรมที่ให้บริการอินเตอร์รัพท์ (Interrupt Service Routine หรือ ISR) นั้นๆ ก่อนกระโดดไปจะต้องมีการเก็บค่าต่างๆ ที่จำเป็นที่ทำงานกับโปรแกรมปัจจุบันไว้ก่อน เช่นค่าในรีจิสเตอร์ PC เพราะเมื่อโปรแกรมให้บริการอินเตอร์รัพท์ทำงานเสร็จแล้ว จะได้กลับมาทำงานโปรแกรมเดิมต่อไปได้ จากนั้นก็กระโดดไปยังโปรแกรมให้บริการอินเตอร์รัพท์ เมื่อไมโครคอนโทรลเลอร์ทำงานเสร็จก็จะกระโดดกลับมาทำงานยังโปรแกรมปัจจุบันในตำแหน่งที่ต่อเนื่องจากเดิมก่อนที่จะเกิดอินเตอร์รัพท์ โดยการคืนค่าต่างๆ ที่บันทึกไว้กลับมา ไมโครคอนเลอร์ MCS-51 จะจองพื้นที่หน่วยความจำส่วนหนึ่งตั้งแต่แอดเดรสที่ 0003H ถึง 00FBH สำหรับเก็บแอดเดรสของโปรแกรมให้บริการอินเตอร์รัพท์หรืออินเตอร์รัพท์เวกเตอร์ (Interrupt vector) ซึ่งแต่ละเวกเตอร์จะใช้พื้นที่ขนาด 8 ไบต์ ดังนั้นจึงมีจำนวนเวกเตอร์ทั้งหมด 32 เวกเตอร์ โดยแอดเดรส 0003H ถึง 0033H ถูกจองไว้สำหรับอินเตอร์รัพท์เวกเตอร์ของ MCS-51/52 ขณะที่ตั้งแต่แอดเดรสที่ 003BH ถึง 00FB เราสามารถกำหนดเวกเตอร์ของโปรแกรมให้บริการอินเตอร์รัพท์ที่เราสร้างขึ้นเองได้อย่างอิสระ เพื่อความเข้าใจง่ายจะใช้ตัวเลขแทนแอดเดรสของเวกเตอร์ เช่น อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT0 (External INT0) จะอยู่ที่เวกเตอร์ 0 (0003H ถึง 000AH) หรือ อินเตอร์รัพท์ไทเมอร์ 0 อยู่ที่เวกเตอร์

1 (000BH ถึง 0012H) เนื่องจากแต่ละเวกเตอร์มีขนาด 8 ไบต์ ดังนั้นเวลาคิดหมายเลขเวกเตอร์ จะเอาแอดเดรสเริ่มต้นมาลบด้วย 3 จากนั้นหารด้วย 8

ตารางที่ 2.1 แสดงหมายเลขอินเตอร์รัพท์ของ MCS-51/52

หมายเลขอินเตอร์รัพท์	แอดเดรส	หมายเลขอินเตอร์รัพท์	แอดเดรส
0 (อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT0)	0003H	16	0083H
1 (อินเตอร์รัพท์ไทเมอร์ 0)	000BH	17	008BH
2 (อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT1)	0013H	18	0093H
3 (อินเตอร์รัพท์ไทเมอร์ 1)	001BH	19	009BH
4 (อินเตอร์รัพท์พอร์ตอนุกรม)	0023H	20	00A3H
5 (อินเตอร์รัพท์ไทเมอร์ 2)	002BH	21	00ABH
6 (PCA)	0033H	22	00B3H
7	003BH	23	00BBH
8	0043H	24	00C3H
9	004BH	25	00CBH
10	0053H	26	00D3H
11	005BH	27	00DBH
12	0063H	28	00E3H
13	006BH	29	00EBH
14	0073H	30	00F3H
15	007BH	31	00FBH

การใช้งานอินเตอร์รัพท์ของไมโครคอนโทรลเลอร์ MCS-51 เกี่ยวข้องกับรีจิสเตอร์ 3 ตัว คือ

1. รีจิสเตอร์ IE (Interrupt Enable) ไว้สำหรับควบคุมการ enable อินเตอร์รัพท์จากแหล่งต่างๆ ประกอบด้วยบิตต่าง ๆ ดังนี้

EA ถ้ากำหนดเป็นลอจิก '1' จะอินเตอร์รัพท์ทั้งหมด

ET2 ถ้ากำหนดเป็นลอจิก '1' จะ enable ไทเมอร์ 2 (สำหรับ MCS-52)

ES ถ้ากำหนดเป็นลอจิก '1' จะ enable พอร์ตอนุกรม

ET1 ถ้ากำหนดเป็นลอจิก '1' จะ enable ไทเมอร์ 1

EX1 ถ้ากำหนดเป็นลอจิก '1' จะ enable อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT1

ET0 ถ้ากำหนดเป็นลอจิก '1' จะ enable ไทเมอร์ 0

EX0 ถ้ากำหนดเป็นลอจิก '1' จะ enable อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT0

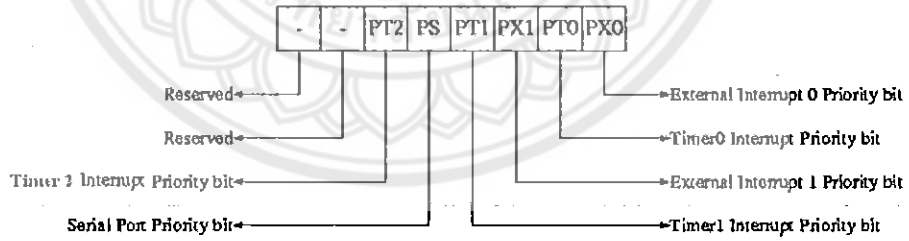
ตัวอย่าง ถ้าต้องการให้ไทเมอร์ 0 ทำงาน (enable) และไทเมอร์ 1 ไม่ทำงาน (disable) สั่งงานด้วยคำสั่ง C51 ดังนี้

```
ET0 = 0;   /* enable interrupt (Timer 0)
*/
ET1 = 1;   /* disable interrupt (Timer
0) */
```

ตัวอย่าง ถ้าต้องการให้อินเตอร์รัพท์ทั้งหมดทำงาน สั่งงานด้วยคำสั่ง C51 ดังนี้

```
EA = 1 ;   /* enable interrupt  */
```

2. รีจิสเตอร์ IP (Interrupt Priority) สำหรับกำหนดลำดับความสำคัญของการอินเตอร์รัพท์ ในกรณีที่มีอินเตอร์รัพท์เกิดขึ้นจากหลายแหล่งพร้อมกัน อินเตอร์รัพท์ที่มีลำดับความสูงสุดจะได้รับบริการก่อน การกำหนดลำดับความสำคัญจะกำหนดโดยเซตบิตในรีจิสเตอร์ IP ดังนี้



ถ้ากำหนดบิตเป็นลอจิก '1' หมายความว่า กำหนดเป็นลำดับความสำคัญสูงสุด
ถ้ากำหนดบิตเป็นลอจิก '0' หมายความว่า กำหนดเป็นลำดับความสำคัญต่ำสุด

แต่ละบิตอธิบายดังนี้

PT2 บิตสำหรับกำหนดไทเมอร์ 2 เป็นลำดับสูงสุด

PS บิตสำหรับกำหนดพอร์ตอนุกรม เป็นลำดับสูงสุด

PT1 บิตสำหรับกำหนดไทเมอร์ 1 เป็นลำดับสูงสุด

PX1 บิตสำหรับกำหนดอินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT1 เป็นลำดับสูงสุด

PT0 บิตสำหรับกำหนดไทเมอร์ 0 เป็นลำดับสูงสุด

PX0 บิตสำหรับกำหนดอินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT0 เป็นลำดับสูงสุด

การกำหนดโครงสร้างลำดับความสำคัญของอินเตอร์รัพท์อีกชั้นนึงเพื่อให้ซีพียูไว้ตรวจสอบว่าอินเตอร์รัพท์ไหนลำดับสูงกว่ากัน แสดงดังตารางที่ 2.2

ตารางที่ 2.2แสดงลำดับความสำคัญอินเตอร์รัพท์ของ MCS-51/52 ภายในระดับเดียวกัน

ชนิดอินเตอร์รัพท์	ลำดับความสำคัญของอินเตอร์รัพท์
อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT0	สูงสุด
อินเตอร์รัพท์ไทเมอร์ 0	
อินเตอร์รัพท์ภายนอกที่เข้ามาทางขา INT1	
อินเตอร์รัพท์ไทเมอร์ 1	ต่ำสุด
อินเตอร์รัพท์พอร์ตอนุกรม	
อินเตอร์รัพท์ไทเมอร์ 2	

ถ้ามี 2 อินเตอร์รัพท์ที่มีระดับความสำคัญต่างกันเข้ามาพร้อมๆกัน ซีพียูจะให้บริการกับอินเตอร์รัพท์ที่มีลำดับความสำคัญสูงสุดก่อน แต่ถ้า 2 อินเตอร์รัพท์มีระดับความสำคัญเท่ากันก็จะใช้วิธีการโพลลิ่งภายใน (Internal Polling)

1. รีจิสเตอร์ TCON (Timer Control)สำหรับควบคุมการใช้งานไทเมอร์และการเกิดอินเตอร์รัพท์ มีรายละเอียดดังนี้

Bit	7 (msb)	6	5	4	3	2	1	0 (lsb)
NAME	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 เป็นบิตโอเวอร์โฟลว์ของไทเมอร์ 1 จะเซต (กำหนดเป็นลอจิก '1') โดยฮาร์ดแวร์เมื่อไท เมอร์เกิดโอเวอร์โฟลว์ และจะเคลียร์ (กำหนดเป็นลอจิก '0') โดย ฮาร์ดแวร์ เมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมให้บริการ อินเตอร์รัพท์ (ISR) TR1 ทำโดยซอฟต์แวร์ ใช้ปิดเปิดไทเมอร์ 1 กำหนดเป็นลอจิก '1' กรณีเปิดไทเมอร์ 1 และกำหนดเป็นลอจิก '0' กรณีปิดไทเมอร์ 1

TF0 เหมือนกับ TF1 แต่ใช้กับไทเมอร์ 0

TR0 ทำโดยซอฟต์แวร์ ใช้ปิดเปิดไทเมอร์ 0 ถ้ากำหนดเป็นลอจิก '1' เปิดไทเมอร์ 0 และกำหนดเป็นลอจิก '0' ปิดไทเมอร์ 0

IE1 เป็นบิตแสดงว่ามีอินเตอร์รัพท์เกิดขึ้น จะเซต (กำหนดเป็นลอจิก '1') โดยฮาร์ดแวร์เมื่อมีสัญญาณอินเตอร์รัพท์ที่ขา INT1 และจะเคลียร์บิต (กำหนดเป็นลอจิก '0') เมื่อประมวลผลอินเตอร์รัพท์เสร็จเรียบร้อย

IT1 ใช้กำหนดรูปแบบของสัญญาณอินเตอร์รัพท์จากภายนอกที่เข้ามาทางขา INT1 ว่ามีลักษณะอย่างไร ถ้าเป็นลอจิก '1' หมายความว่าจะเกิดการอินเตอร์รัพท์เมื่อสัญญาณขอบขาลงเข้ามา ถ้าเป็นลอจิก '0' หมายความว่าจะเกิดการอินเตอร์รัพท์เมื่อสัญญาณขอบขาขึ้นเข้ามา

IE0 เหมือนกับ IE1 แต่จะใช้งานกับ INT0

IT0 เหมือนกับ IT1 แต่จะใช้งานกับ INT0

การคำนวณค่าเวลาที่จะกำหนดค่าให้ TH1 และ TL1 จะต้องพิจารณาจากช่วงเวลาที่ต้องการให้เกิดอินเตอร์รัพท์แต่ละครั้ง (จากตัวอย่าง 5 มิลลิวินาที) และความถี่ของสัญญาณนาฬิกา โดยทั่วไป MCS-51 จะอยู่ที่ 11.0592 MHz และหนึ่งแมชชีนไซเคิล (machine cycle) ใช้ สัญญาณนาฬิกา 12 ลูก ดังนั้นความถี่ของแมชชีนไซเคิล เท่ากับ 921.6 KHz (11.0592 MHz / 12) และเวลาต่อแมชชีนไซเคิล เท่ากับ 1.085 ไมโครวินาที (1/921.6 KHz) และค่าที่จะไปใส่ใน TH1 และ TL1 คำนวณดังนี้

1. จำนวนสัญญาณนาฬิกาที่ทำให้เกิดอินเตอร์รัพท์

$$\text{จำนวนสัญญาณนาฬิกา} = \frac{\text{เวลาที่นับ}}{\text{เวลาต่อแมชชีนไซเคิล}}$$

$$= 5 \text{ msec} / 1.085 \ \mu s$$

$$= 4608$$

2. คำนวณค่าที่จะโหลดลงไทเมอร์

$$\text{ค่าเวลา} = 65536 - \text{จำนวนสัญญาณนาฬิกา}$$

$$= 65536 - 4608$$

$$= 60928$$

$$= EE00 \text{ (ฐานสิบหก)}$$

ดังนั้น ค่าใน TH1 = EE (เก็บไบต์สูง)

TL1 = 00 (เก็บไบต์ต่ำ)

2.15 การสื่อสารข้อมูลผ่านพอร์ตอนุกรม [4]

ไมโครคอนโทรลเลอร์ MCS-51 มีอุปกรณ์ช่วยให้การติดต่อสื่อสารข้อมูลแบบอนุกรมกับอุปกรณ์อื่นอยู่ภายใน เรียกว่า UART (Universal Asynchronous Receiver/Transmitter) หรือที่เรารู้จักในชื่อ พอร์ตสื่อสารอนุกรม (Serial port) โดยก่อนการใช้งานเราต้องกำหนดโหมดการทำงานของพอร์ตอนุกรม และอัตราบอด (Baud rate) เสียก่อน

พอร์ตอนุกรมของ MCS-51 สามารถสื่อสารข้อมูลได้แบบทั้งรับและส่งในเวลาเดียวกัน เรียกว่าฟูลดูเพล็กซ์ (Full duplex) โดยจะใช้พอร์ต P3.0 สำหรับรับข้อมูล และพอร์ต P3.1 สำหรับส่งข้อมูล ซึ่งภายในจะมีรีจิสเตอร์สำหรับรับและส่งข้อมูลแยกต่างหาก แต่อย่างไรก็ตามในการใช้งานเราจะอ้างถึงรีจิสเตอร์ 2 ตัวนี้ ด้วยชื่อเดียวกันคือ SBUF (แอดเดรสที่ 99H ใน SFR) เช่น ถ้าต้องการส่งข้อมูลก็ให้เขียนข้อมูลไปที่ SBUF หรือถ้าต้องการรับข้อมูล ก็ให้อ่านข้อมูลจาก SBUF ดังนั้น ก่อนการใช้งานพอร์ตอนุกรม จะต้องกำหนดให้ MCS-51 รู้เสียก่อนว่าจำนวนบิตที่จะส่งแต่ละครั้งเท่าไร Baud rate ที่ใช้รับส่งข้อมูลเท่าไร และจะกำหนด Baud rate ที่ต้องการอย่างไร

ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม จะกำหนดผ่านรีจิสเตอร์ควบคุมพอร์ตอนุกรม หรือ SCON (Serial Port Control Register) อธิบายดังนี้

2.16 รีจิสเตอร์ควบคุมพอร์ตอนุกรม (SCON) [4]

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 (Serial port mode bit 0) สำหรับกำหนดโหมดการทำงานของพอร์ตอนุกรม โดยจะใช้ร่วมกับบิต SM1

SM1 (Serial port mode bit 1) ใช้ร่วมกับบิต SM1 ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม

SM2 (Serial port mode bit 2) ใช้ควบคุมเมื่อมีการเชื่อมต่อไมโครคอนโทรลเลอร์หลายตัว ในโหมด 2 และโหมด 3 นั่นคือ เมื่อเป็นลอจิก '0' จะทำงานในโหมด 0 และเป็นลอจิก '1' จะทำงานในโหมดหลายไมโครคอนโทรลเลอร์

REN (Received enable bit) ใช้เอ็นเอเบิลการรับข้อมูล นั่นคือเมื่อเป็นลอจิก '0' กำหนดให้รับข้อมูล และเป็นลอจิก '1' กำหนดให้ไม่รับข้อมูล

TB8 (Transmit data bit 8) สำหรับเก็บข้อมูลบิตที่ 9 เมื่อต้องการส่งออกพอร์ตอนุกรม ในโหมด 2 และ3

RB8 (Receive data bit 8) สำหรับเก็บข้อมูลบิตที่ 9 เมื่อมีการรับข้อมูลจากพอร์ตอนุกรม ในโหมด 2 และโหมด 3

TI (Transmit interrupt flag)เป็นบิตแสดงสถานการณ์ส่งข้อมูลออกพอร์ตอนุกรมเมื่อ MCS-51 ส่งข้อมูลเรียบร้อย ก็จะทำการเซตบิต TI เป็นลอจิก '1' (ไม่มีข้อมูลใน SBUF) เพื่อบอกให้โปรแกรมรู้ว่า จะสามารถส่งข้อมูลอื่นได้ (นำข้อมูลมาใส่ใน SBUF)

RI (Receive interrupt flag)) เป็นบิตใช้แสดงสถานะการรับข้อมูลจากพอร์ตอนุกรม เมื่อ MCS-51 รับข้อมูลเรียบร้อย (มีข้อมูลมาที่ SBUF) ก็จะทำการเซตบิต RI เป็นลอจิก '1' เพื่อบอกให้โปรแกรมรู้ว่าจะต้องอ่านข้อมูลจาก SBUF อย่างรวดเร็ว ก่อนที่ข้อมูลอื่นจะเข้ามาแทนที่

2.17 การกำหนดอัตราบอด (Baud Rate) [2]

หลังจากเลือกโหมดการทำงานพอร์ตอนุกรมแล้ว ต่อไปจะต้องทำการกำหนดอัตราบอด อัตราบอด คือความเร็วที่ใช้ในการรับส่งข้อมูลของพอร์ตอนุกรม การกำหนดอัตราบอดนี้จะทำเฉพาะในโหมด 1 และ 3 เท่านั้น ส่วนโหมด 0 และโหมด 2 จะถูกกำหนดโดยอัตโนมัติตามความถี่สัญญาณนาฬิกา (Crystal) ที่ป้อนให้กับ MCS-51 สำหรับในโหมด 0 ค่าอัตราบอดเกิดจากการนำเอาความถี่สัญญาณนาฬิกาหารด้วย 12 เช่น ถ้า Crystal ความถี่ 11.059 Mhz ค่าอัตราบอดที่ได้จะเป็น 921,583 bps (bps = baud per second) สำหรับในโหมด 2 ค่าอัตราบอดเกิดจากการนำเอาความถี่สัญญาณนาฬิกาหารด้วย 64 เช่นถ้าCrystal ความถี่ 11.059 Mhzค่าอัตราบอดที่ได้จะเป็น 172,797 bps

ส่วนในโหมด 1 และโหมด 3 ค่าอัตราบอดจะกำหนดโดยการทำให้ไทเมอร์ 1 เกิดการนับเกิน (Overflow) วิธีที่ทำให้เกิดการนับเกินของไทเมอร์ 1 มีหลายวิธี แต่ที่นิยมใช้คือ กำหนดให้ไทเมอร์ 1 ทำงานในโหมด 2 (8-bit auto-reload mode) และโหลดค่าที่จะทำให้เกิดการนับเกินลงใน TH1

การกำหนดค่าให้ TH1 เพื่อกำเนิดอัตราบอด กรณีที่บิต PCON.7 เป็นลอจิก '0' คำนวณดังนี้ TH1 = 256 - ((Crystal / 384) / Baud) ถ้า PCON.7 เป็นลอจิก '1' ค่าอัตราบอดจะเป็น 2 เท่า คำนวณดังนี้ TH1 = 256 - ((Crystal / 192) / Baud)

ตัวอย่าง สมมุติว่ามี Crytal ความถี่ 11.059 Mhz ต้องการกำหนดอัตราบอดที่ 19,200 bps จงคำนวณหาค่าที่จะโหลดลงใน TH1

วิธีคิด

TH1 = 256 - ((Crystal / 384) / Baud)

TH1 = 256 - ((11059000 / 384) / 19200)

TH1 = 256 - ((28,799) / 19200)

TH1 = 256 - 1.5 = 254.5

เนื่องจากได้ค่าเป็นเลขทศนิยม ต้องกำหนดเป็นจำนวนเต็ม ให้ทำการปัดขึ้นหรือปัดลง ถ้าปัดลงจะได้ค่า TH1 เป็น 254 คำนวณอัตราบอดแล้วจะได้ 14,400 และถ้าปัดขึ้นจะได้ค่า TH1 เป็น 255 คำนวณอัตราบอดแล้วจะได้ 28,800 จะเห็นว่ายังไม่ถูกต้อง ต้องเปลี่ยนการคำนวณเป็นแบบอัตราบอด 2 เท่า คำนวณใหม่ได้ดังนี้

TH1 = 256 - ((Crystal / 192) / Baud)

TH1 = 256 - ((11059000 / 192) / 19200)

TH1 = 256 - ((57699) / 19200)

TH1 = 256 - 3 = 253

ต้องแปลงเป็นฐานสิบหกก่อนโหลดลง TH1 จะได้ FD สรุปขั้นตอนการกำหนดอัตราบอด 19,200 ให้กับ MCS-51 ดังนี้

1. กำหนดพอร์ตอนุกรมทำงานในโหมด 1 หรือ 3
2. กำหนดให้ไทเมอร์ 1 ทำงานในโหมด 2 (8-bit auto-reload)
3. เซตค่า TH1 เป็น FD
4. เซตบิต PCON.7 (SMOD) เป็นลอจิก '1' เพื่อกำหนดอัตราบอดเป็นแบบ 2 เท่า สำหรับอัตราบอดแบบธรรมดา (บิต PCON.7 เป็นลอจิก '0')

หลังจากทำการกำหนดโหมดการทำงานและกำหนดอัตราบอดให้กับ MCS-51 สำหรับการสื่อสารผ่านพอร์ตอนุกรมเรียบร้อยแล้ว พอร์ตอนุกรมก็พร้อมสำหรับการรับส่งข้อมูล

2.18 การรับส่งข้อมูลผ่านพอร์ตอนุกรมด้วยวิธีอินเตอร์รัพท์ (Interrupt)

การรับส่งข้อมูลผ่านพอร์ตอนุกรมด้วยวิธีโพลลิ่ง เราต้องเขียนโปรแกรมคอยรับคอยส่งข้อมูลตลอดเวลา ถ้าข้อมูลยังมาไม่ถึงโปรแกรมก็จะต้องวนลูปรอจนกระทั่งมีข้อมูลมาที่ SBUF ทำให้ซีพียูไม่สามารถไปทำงานอื่นได้ ซึ่งไม่เหมาะกับงานที่ตอบสนองต่อเวลาจริง (real time system) แก้โดยใช้วิธีอินเตอร์รัพท์พอร์ตสื่อสารอนุกรม ซึ่งอยู่ในอินเตอร์รัพท์เวกเตอร์ลำดับที่ 4 (interrupt vector 4) หรือที่แอดเดรสที่ 0x0023 โดยอินเตอร์รัพท์พอร์ตอนุกรมนี้เกิดจากการเซตบิต RI หรือบิต TI นั่นหมายความว่า เมื่อบิตใดบิตหนึ่งถูกเซต (เป็นลอจิก '1') ก็ให้กระโดดไปทำงานในฟังก์ชันบริการอินเตอร์รัพท์ลำดับที่ 4 ทันที สำหรับภายในฟังก์ชันบริการอินเตอร์รัพท์ถ้าบิตไหนถูกเซต (RI หรือ TI) เมื่อทำการรับหรือส่งข้อมูลเสร็จแล้ว ก็ให้ทำการเคลียร์บิตนั้นทันทีเพื่อเตรียมพร้อมสำหรับการเกิดอินเตอร์รัพท์ครั้งต่อไป

2.19 ส่วนควบคุมการส่ง SMS [1]

ในส่วนของ SMS ที่จะส่งไปใน SMS การสั่งงานจะสั่งผ่านทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่โดยส่งเป็นชุดคำสั่ง AT-COMMAND ซึ่งชุดคำสั่งที่ใช้จะเป็นดังนี้

AT+CMGF เป็นคำสั่งในการเลือกโหมดของ Message ที่จะส่ง

AT+CMGL เป็นคำสั่งสำหรับอ่าน Message ที่เลือกไว้ จาก SIM card

AT+CMGS เป็นคำสั่งสำหรับส่ง Message ไปยัง Address ที่เลือกไว้

การส่ง SMS นั้นเราจะใช้ไมโครคอนโทลเลอร์ในการส่งโดยจะส่งชุดคำสั่ง AT-COMMAND เข้าไปทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่ที่ต่ออยู่กับไมโครคอนโทรเลอร์ แต่ก่อนที่ส่งนั้นเราต้องทำการกำหนด อัตรา Baud Rate ให้ตรงกับโทรศัพท์เคลื่อนที่เสียก่อน หลังจากนั้นเราจะใช้ AT+CMGL ในการตรวจเช็คข้อความที่มีการนำเข้า หลังจากนั้นจะทำการส่ง AT+CMGS พร้อมกับบอกจำนวนของข้อความ ตามด้วย "ข้อความที่เราต้องการส่ง" ตามด้วย Ctrl+Z เพื่อบอกว่าสิ้นสุดข้อความแล้ว

ในส่วนควบคุมการส่ง SMS นี้สามารถศึกษาได้ตามตัวอย่างการส่ง SMS ดังในตัวอย่างที่แสดง จะทำการส่ง SMS ที่มีข้อความว่า "TEST" ในรูปแบบของ PDU Mode ลักษณะดังนี้

ตัวอย่างการส่ง SMS

AT+CMGF=0	"เซ็ต PDU Mode"
OK	"โทรศัพท์เคลื่อนที่ตอบกลับมาว่า "OK"
AT+CMGS=22	"ส่ง SMS โดยความยาว TPDU = 22 Octests"
>0011000A8180737899750000AA0AE8329BFD4697D9EC3 +CMGS:64	

สามารถแยกข้อความได้เป็น 2 ส่วน คือ SMSC กับ TPDU ซึ่งมีลักษณะดังนี้

SMSC	: 00
00	: คือความยาวของเบอร์ SMSC = 0 คือให้ใช้ SMSC ที่เก็บอยู่ภายในเครื่อง
TPDU	: 11000A8180737899750000AA0AE8329BFD4697D9EC3
1100	: คือสถานะของ SMS ที่ SMSC
0A	: ความยาวของเบอร์โทรศัพท์เคลื่อนที่ที่ต้องการส่ง SMS = 10
81	: คือตัวบอกว่าเบอร์โทรศัพท์เคลื่อนที่เป็นแบบภายในประเทศ
8073789975	: คือเบอร์โทรศัพท์ปลายทางที่ทำการเข้ารหัส แล้ว 083-7879957
0000	: คือลักษณะการเข้ารหัสของข้อมูล (มาตรฐาน GSM)
AA	: คือตัวบอกลักษณะการเข้ารหัส
0A	: คือความยาวข้อความก่อนแปลงรหัส = 10
E8329BFD4697D9EC3	: คือข้อความที่แปลงรหัสเป็นเลขฐาน 16 แล้ว ก่อนแปลงคือคำว่า "hellohello"

2.20 ส่วนควบคุมการรับ SMS

ในส่วนของ SMS ที่จะรับ SMS เราจะใช้ชุดคำสั่ง AT-COMMAND ซึ่งชุดคำสั่งที่ใช้จะเป็นดังนี้

AT+CMGL = 0 เป็นคำสั่งสำหรับอ่าน Message ที่เลือกไว้จาก SIM card

0691666113015004 0B916698761688F7 0000600171121351820AE8329BFD4697D9EC37

06	=	ความยาวของ SMSC
91	=	รูปแบบของเลขหมาย SMSC แบบระบบสากล
66 61 13 01 50	=	เลขหมายโทรศัพท์จริงจาก Service Center 66-16311005
04	=	รหัสตัวแรกของข้อมูล SMS
0B	=	ความยาวของหมายเลขโทรศัพท์ 11 ตัว
91	=	เลขหมายแบบต่างประเทศ
66 98 76 16 88 F7	=	หมายเลขโทรศัพท์ที่ส่งซึ่งเข้ารหัสอยู่ 66-089-6761887
00	=	Protocal ID
00	=	Data coding scheme
60 01 71 12 13 51 82	=	วันที่ 17 /10/06 เวลา 21:31:51.82
0A	=	ความยาวของตัวอักษร 10 ตัวอักษร
E8 32 9B FD 46 97 D9 EC 37	=	ข้อความที่ถูเข้ารหัสเป็น PDU Code อยู่

บทที่ 3

วิธีการดำเนินงานโครงงานวิศวกรรม

ในการจัดทำโครงงานเรื่องระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ประกอบไปด้วยวิธีการดำเนินงาน 4 ขั้นตอน ดังนี้

3.1 การวิเคราะห์ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.2 การออกแบบระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.3 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.4 การทดสอบและประเมินผลการดำเนินโครงงาน

3.1 การวิเคราะห์ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

ในขั้นแรกของการดำเนินโครงงานคือการวิเคราะห์ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ดังนี้

3.1.1 วิเคราะห์ปัญหาอันเป็นที่มาของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.1.2 วิเคราะห์องค์ประกอบของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือจากเอกสารและงานวิจัยที่เกี่ยวข้อง

3.1.3 วิเคราะห์การทำงานของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ โดยพิจารณาจากความต้องการของระบบ

3.2 การออกแบบระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

ในขั้นตอนของการการออกแบบระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ แบ่งออกเป็น 2 ขั้นตอนคือ

3.2.1 การออกแบบระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.2.2 การออกแบบการทำงานของโปรแกรม

3.3 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

ในขั้นตอนของการพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือนั้นสามารถแบ่งการพัฒนาออกเป็น 2 ด้านคือ

3.3.1 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ด้านอุปกรณ์

3.3.2 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ด้านการโปรแกรม

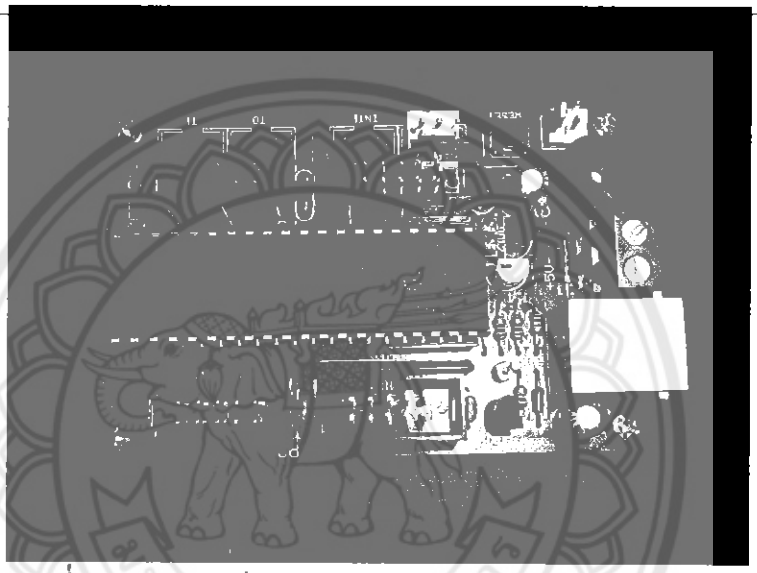
โดยเริ่มจากการพัฒนาระบบการแจ้งเตือนภัยผ่านโทรศัพท์มือถือด้านอุปกรณ์ดังนี้

3.4 ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือด้านอุปกรณ์

ในการพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือประกอบด้วยอุปกรณ์ดังนี้

3.4.1 บอร์ดไมโครคอนโทรลเลอร์ MCS-51 เบอร์ P89V51RD2

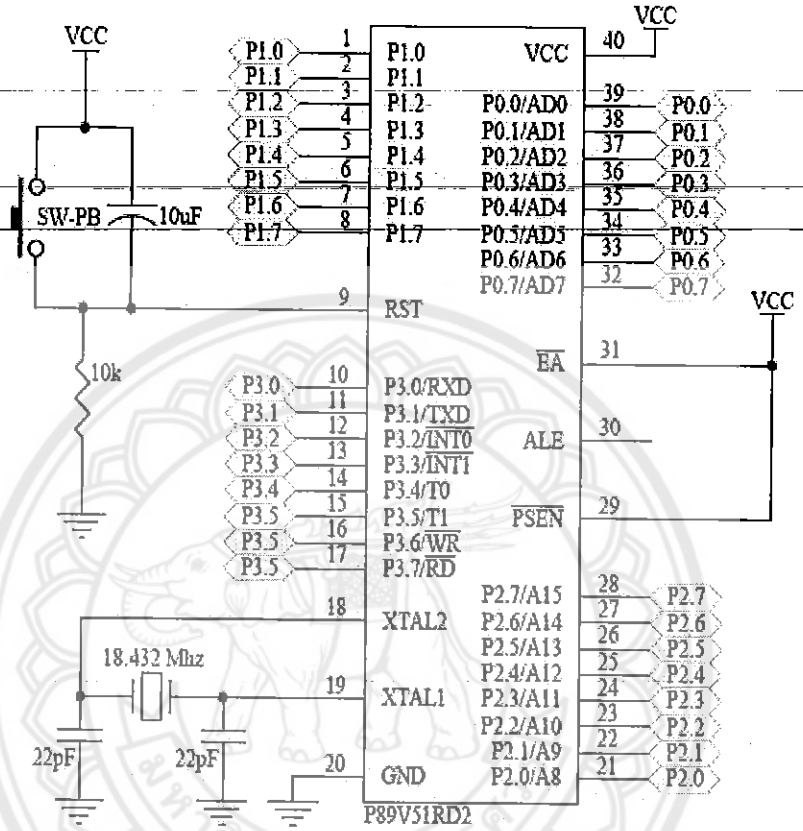
เป็นบอร์ดที่ใช้ในการทำโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือโดยมีพอร์ตอินพุต และเอาท์พุตอย่างละ 2 พอร์ต ดังรูปที่ 3.1



รูปที่ 3.1 แสดงบอร์ดไมโครคอนโทรลเลอร์ เบอร์ P89V51RD2

สำหรับไมโครคอนโทรลเลอร์ MCS-51 ที่นำมาใช้ในโครงงานนี้เป็นรุ่น P89V51RD2 ของบริษัท Philips ที่เลือกรุ่นนี้เนื่องจากเป็นรุ่นที่สามารถรองรับการดาวน์โหลดโปรแกรมแบบ ISP (In System Programming) ผ่านพอร์ตอนุกรมได้โดยตรง ไม่ต้องอาศัยอุปกรณ์ หรือวงจรเพิ่มเติมในการดาวน์โหลดโปรแกรม จึงทำให้สามารถใช้งานได้อย่างสะดวก รวมถึงราคาของ P89V51RD2 ที่ไม่แพง เมื่อเทียบกับความสามารถ และประสิทธิภาพของมัน P89V51RD2 สามารถทำงานในโหมด X2 ซึ่ง จะทำให้สามารถทำงานได้เร็วกว่า MCS-51 พื้นฐาน 2 เท่า (1 แมชชีนไซเคิล ใช้สัญญาณนาฬิกา 6 ลูก) เมื่อใช้คริสตอลความถี่ที่เท่ากัน ในการทำงานในโหมด X2 นี้ P89V51RD2 สามารถใช้คริสตอลความถี่สูงสุด 20MHz ส่วนในการทำงานในโหมด X1 สามารถใช้คริสตอลความถี่สูงสุด 40 MHz ภายใน P89V51RD2 มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 64 กิโลไบต์ นอกจากนั้นยังมี หน่วยความจำข้อมูลภายนอกเพิ่มเติมขนาด 1 กิโลไบต์ อยู่ภายในตัวชิพด้วย

วงจรไมโครคอนโทรลเลอร์ P89V51RD2 เป็นอุปกรณ์ที่ใช้ในการทำโครงงานซึ่งแสดงรายละเอียดของวงจรแสดงได้ดังรูปที่ 3.2

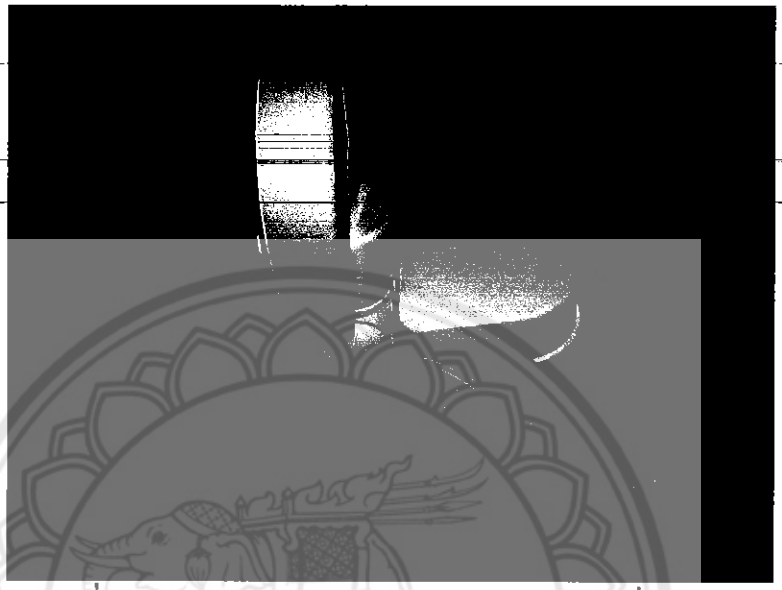


รูปที่ 3.2 วงจรไมโครคอนโทรลเลอร์ P89V51RD2

ในส่วนวงจรไมโครคอนโทรลเลอร์สามารถต่อได้ตามวงจรในรูปที่ 3 โดย Vcc 5 V นั้นใช้จาก Vcc ของวงจรจ่ายไฟ จะเห็นได้ว่าไมโครคอนโทรลเลอร์มีพอร์ต I/O แบบขนาน 8 บิตอยู่ทั้งหมด 4 พอร์ต (พอร์ต 0 – พอร์ต 3) แต่ละพอร์ตสามารถทำงานเป็นพอร์ต Input หรือ Output แล้วแต่ผู้ใช้งานจะเลือกใช้

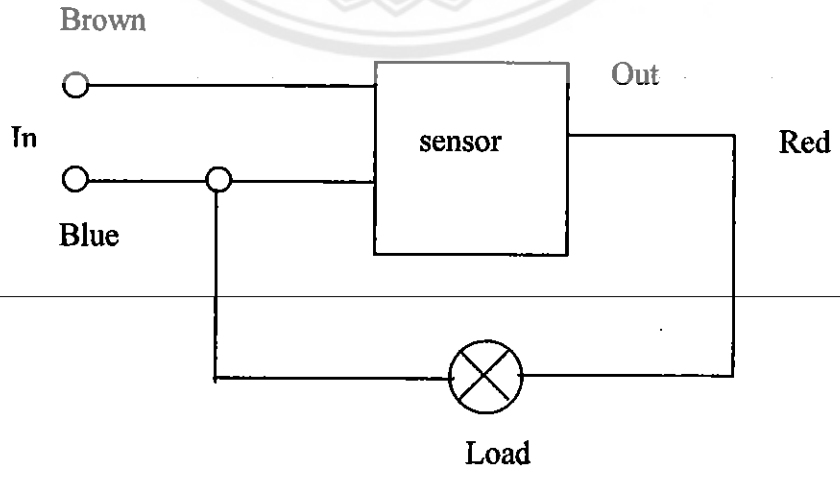
3.4.2 อุปกรณ์ตรวจจับความเคลื่อนไหว (sensor)

เป็นอุปกรณ์ต่อพ่วงกับบอร์ดไมโครคอนโทรลเลอร์ซึ่งในโครงงานวิศวกรรมนี้ผู้จัดทำได้ใช้เซนเซอร์แบบ Infrared motion sensor ซึ่งสามารถตรวจจับความเคลื่อนไหวต่างๆ ได้ดังแสดงในรูป 3.3



รูปที่ 3.3 แสดงรูปภาพของอุปกรณ์ตรวจจับความเคลื่อนไหว

อุปกรณ์ตรวจจับความเคลื่อนไหว (Sensor) แบบ INFRARED MOTION SENSOR มีการทำงานคือ เมื่อเราติดตั้งอุปกรณ์ตรวจจับความเคลื่อนไหว เมื่อมีการเคลื่อนไหวเกิดขึ้นก็จะส่งสัญญาณไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อดำเนินการต่อไปโดยมีวงจรดังแสดงในรูป 3.4



รูปที่ 3.4 แสดงวงจรของอุปกรณ์ตรวจจับความเคลื่อนไหว (sensor)

3.5 การพัฒนาระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือด้านการโปรแกรม

ในการสั่งงานจะสั่งผ่านทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่โดยส่งเป็นชุดคำสั่ง AT-COMMAND ซึ่งชุดคำสั่งที่ใช้จะเป็นดังนี้

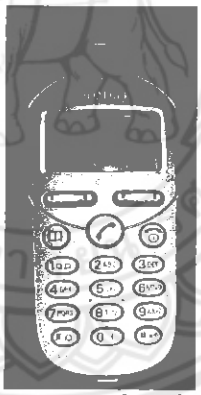
AT+CMGF เป็นคำสั่งในการเลือกโหมดของ Message ที่จะส่ง

AT+CMGL เป็นคำสั่งสำหรับอ่าน Message ที่เลือกไว้ จาก SIM card

AT+CMGS เป็นคำสั่งสำหรับส่ง Message ไปยัง Address ที่เลือกไว้

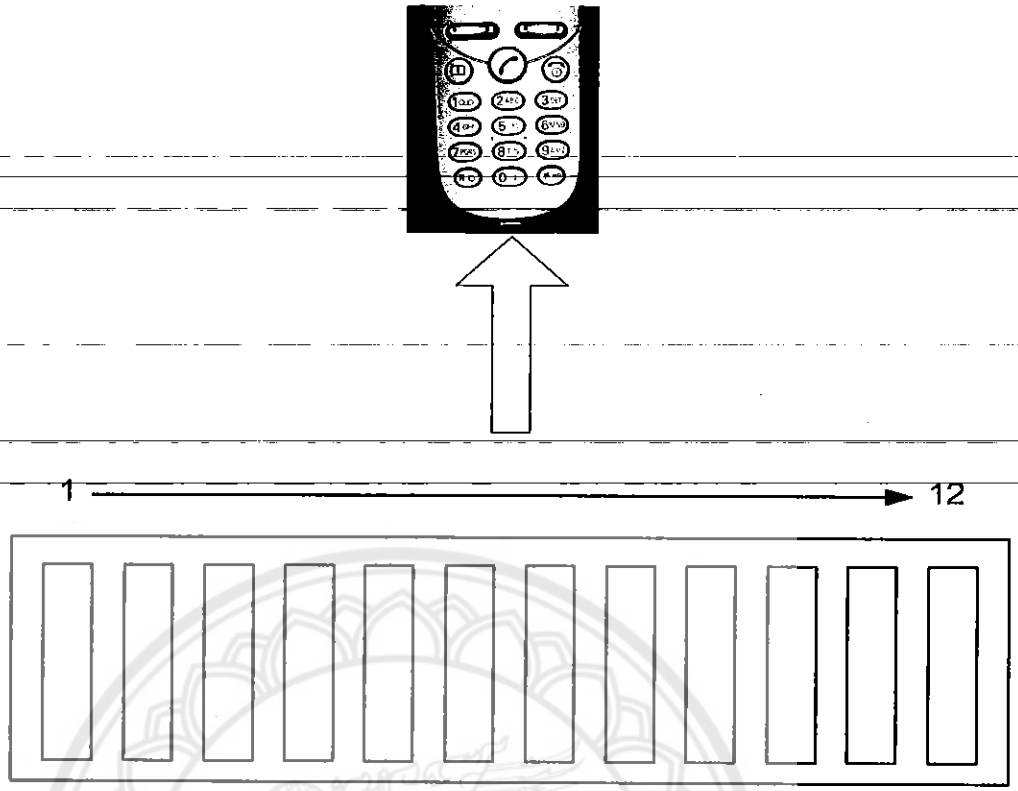
การส่ง SMS นั้นเราจะใช้ไมโครคอนโทลเลอร์ ในการส่งโดยจะส่งชุดคำสั่ง AT-COMMAND เข้าไปทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่ที่ต่ออยู่กับไมโครคอนโทรเลอร์ แต่ก่อนที่ส่งนั้นเราต้องทำการกำหนด อัตรา Baud Rate ให้ตรงกับโทรศัพท์เคลื่อนที่เสียก่อน หลังจากนั้นเราจะใช้ AT+CMGL ในการตรวจเช็คข้อความที่มีการนำเข้า หลังจากนั้นจะทำการส่ง AT+CMGS พร้อมกับบอกจำนวนของข้อความ ตามด้วย "ข้อความที่เราต้องการส่ง" ตามด้วย Ctrl+Z เพื่อบอกว่าสิ้นสุดข้อความ ในการทำงานโครงงานนี้ ได้ใช้โทรศัพท์เคลื่อนที่ Siemens รุ่น C35 ดังแสดงไว้ดังรูปที่ 3.5

14303507
ว.ร.
ค.2495
2551



รูปที่ 3.5 แสดงโทรศัพท์เคลื่อนที่ Siemens รุ่น C35

ซึ่งโทรศัพท์ที่สามารถใช้งานได้ควรเป็นโทรศัพท์ที่มีโมเด็มอยู่ในตัว จึงจะสามารถส่งข้อความได้ ที่สามารถหามาใช้งานในโครงงานนี้ได้เป็นโทรศัพท์ Siemens รุ่น C35 ซึ่งโทรศัพท์เคลื่อนที่รุ่นนี้จะมีพอร์ตที่สามารถต่อกับไมโครคอนโทรเลอร์ได้ โดยใช้สายส่งข้อมูลที่ซื้อตามรุ่นของโทรศัพท์เคลื่อนที่แต่ละเครื่องโดยต่อเข้ากับไมโครคอนโทรเลอร์ทางพอร์ตอนุกรมดังแสดงในรูป 3.6



รูปที่ 3.6 แสดงจำนวนพอร์ตต่ออุปกรณ์ภายนอกของโทรศัพท์เคลื่อนที่ที่ใช้งาน

ซึ่งในแต่ละขาสามารถอธิบายได้ดังนี้

ขาที่ 1 กราวด์ (GND)

ขาที่ 2 ไม่นำมาใช้งาน

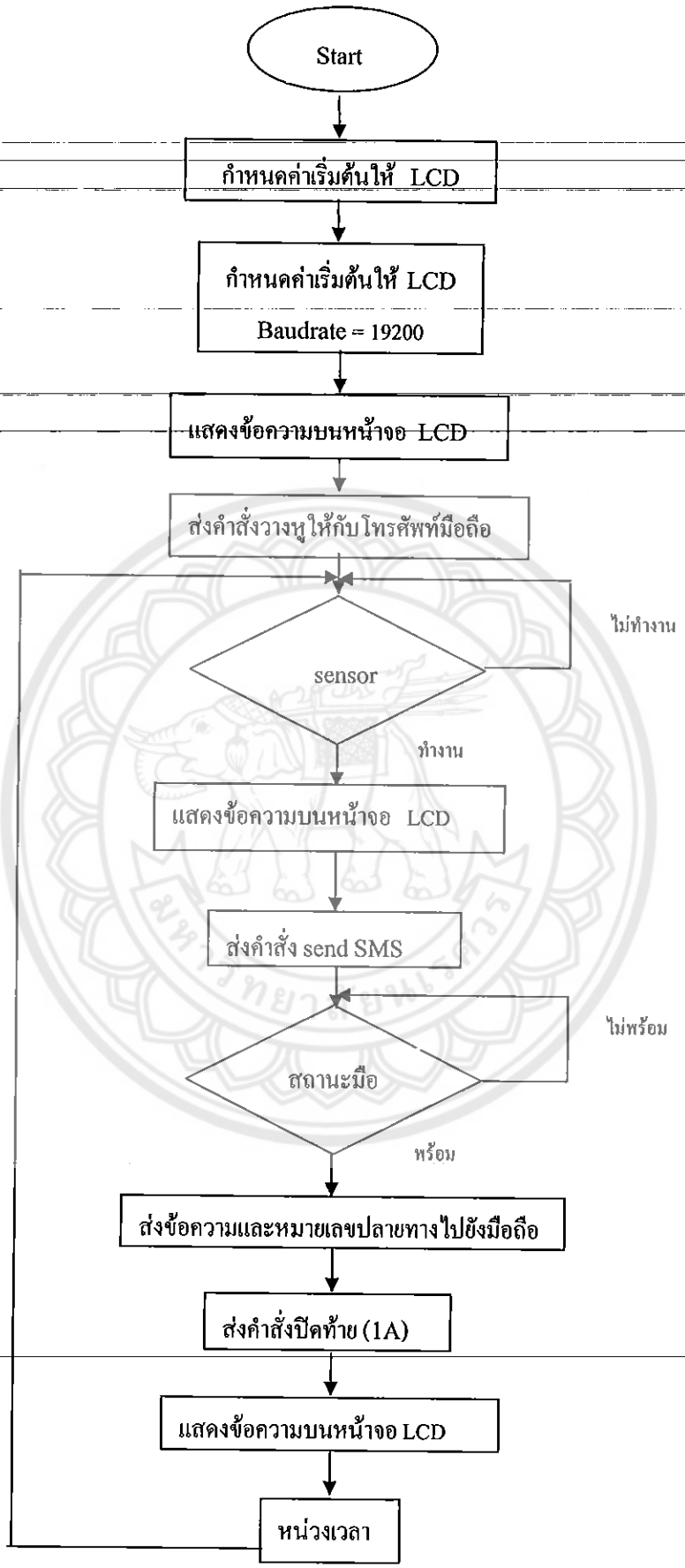
ขาที่ 3 ขั้วบวกใช้ในการชาร์จแบตเตอรี่

ขาที่ 4 ไม่นำมาใช้งาน

ขาที่ 5 (TX) ใช้ในการส่งข้อมูล

ขาที่ 6 (RX) ใช้ในการรับข้อมูล

ขาที่ 7-11 ไม่นำมาใช้งาน

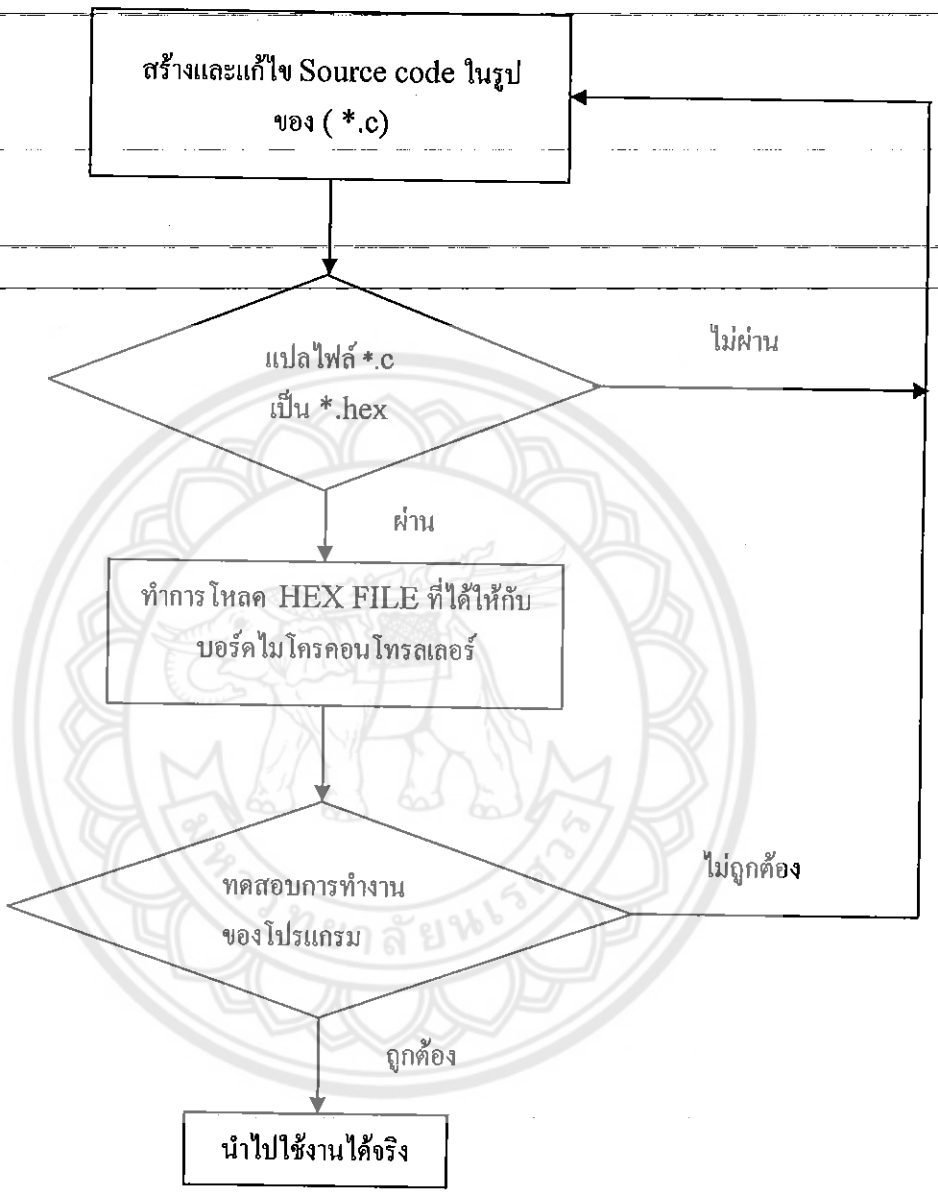


รูปที่ 3.7 แสดง flow chart ของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

3.6 คำอธิบายโปรแกรม

เมื่อเริ่มต้นการทำงาน อันดับแรกจะเป็นการกำหนดค่าเริ่มต้นให้กับ LCD ก่อนแล้วตั้งค่า Baud -rate โดยกำหนดให้มีค่าเท่ากับ 19200 Hz เพื่อให้มีค่าตรงกับบอร์ดไมโครคอนโทรลเลอร์ และให้หน้าจอ LCD แสดงข้อความเพื่อแสดงสถานการณ์ทำงานของอุปกรณ์ตรวจจับความเคลื่อนไหว ขั้นตอนต่อไปจะทำการตรวจสอบโทรศัพท์มือถือที่ใช้ทำการส่ง SMS โดยเราจะส่งคำสั่งวางหู เพื่อให้มือถือนั้นทำการเคลียร์สถานะ การโทรออก หลังจากนั้นก็จะทำการตรวจสอบว่าอุปกรณ์ตรวจจับความเคลื่อนไหวนั้นได้ต่ออยู่กับชุดบอร์ดไมโครคอนโทรลเลอร์หรือไม่ ถ้าได้ต่ออุปกรณ์ตรวจจับความเคลื่อนไหวแล้วให้ทำการแสดงข้อความบนจอ LCD แต่ถ้าไม่ได้ต่อไว้ก็จะให้เริ่มกลับไปตรวจสอบใหม่อีกครั้งจนกว่าจะทำการต่ออุปกรณ์ตรวจจับความเคลื่อนไหวแล้ว เมื่อตรวจสอบแล้วว่าอุปกรณ์จรวจจับความเคลื่อนไหวถูกเชื่อมต่อไว้แล้วก็จะทำการแสดงข้อความบนหน้าจอ LCD ถ้าเกิดอุปกรณ์ตรวจจับความเคลื่อนไหวจับได้ว่ามีการบุกรุกก็จะทำการส่งค่าบิตเพื่อให้ส่งคำสั่ง AT- COMMAND โดยก่อนส่งข้อความไปจะเช็คสถานะของโทรศัพท์มือถือที่ใช้ส่งข้อความเตือนว่าพร้อมใช้งานก็จะส่งข้อความแจ้งเตือนไปแล้วแสดงข้อความที่หน้าจอ LCD ว่าได้ส่งข้อความแจ้งเตือนภัยออกไปแล้ว สุดท้ายโปรแกรมก็จะหน่วงเวลาเพื่อให้อุปกรณ์ตรวจจับความเคลื่อนไหวทำงานอีกครั้ง

การพัฒนาโปรแกรมสำหรับใช้ในการส่ง SMS นั้นจะมีลำดับขั้นตอนในการพัฒนาโปรแกรมดังรูปที่ 3.8



รูปที่ 3.8 แผนผังแสดงขั้นตอนในการพัฒนาโปรแกรมระบบแจ้งเตือนภัยผ่าน SMS

ในกรณีที่ใช้ MCU เบอร์ P89V51RD2 ของ Philips นั้น วิธีการ Download โปรแกรมให้กับบอร์ดจะทำได้โดยง่ายกว่า MCU เบอร์อื่นๆ เนื่องจากสามารถใช้วงจรสื่อสารอนุกรม RS232 ปรกติเหมือนกับการใช้งานทั่วๆไป ในการ Download โปรแกรมได้ทันที โดยไม่ต้องจัดวงจรควบคุมอื่นๆ ให้กับ MCU อีก

บทที่ 4

การทดสอบและการวิเคราะห์การทำงาน

หลังจากประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน จนสามารถทดสอบการทำงานของระบบได้ โดยการกำหนดรูปแบบข้อความที่ต้องการส่งเมื่ออุปกรณ์ตรวจจับความเคลื่อนไหวเริ่มทำงาน จากนั้นบอร์ดไมโครคอนโทรลเลอร์จะสั่งให้มือถือส่งข้อความเตือนภัยไปยังมือถือปลายทาง

สำหรับระบบการทำงานสามารถแบ่งออกเป็น 4 ส่วนหลัก ๆ คือ

4.1 ระบบตรวจจับความเคลื่อนไหว

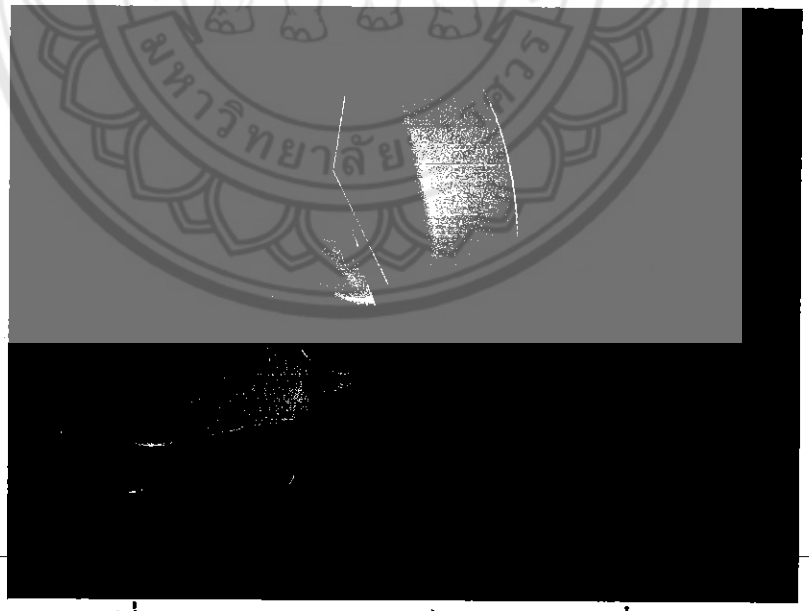
4.2 การควบคุมให้ระบบส่งข้อความแจ้งเตือนภัยไปยังหมายเลขปลายทาง

4.3 ระบบแจ้งสถานะ การส่งข้อความบนจอ LCD

4.4 ระบบ รับ-ส่ง ข้อความ

4.1 ระบบตรวจจับความเคลื่อนไหว

ในการทำโครงงานวิศวกรรมนี้เราจะใช้อุปกรณ์ตรวจจับความเคลื่อนไหวดังที่เรียกว่า Infrared motion sensor ดังแสดงในรูป 4.1



รูปที่ 4.1 แสดงรูปของอุปกรณ์ตรวจจับความเคลื่อนไหว

อุปกรณ์ตรวจจับความเคลื่อนไหวนี้จะทำงานเมื่อมีการเคลื่อนไหว เมื่อมีการ Interrupt อุปกรณ์ตรวจจับความเคลื่อนไหวจะส่งลอจิก 0 เข้าไปที่ขา Int0 ที่บอร์ดไมโครคอนโทรลเลอร์

4.2 การควบคุมให้ระบบส่งข้อความแจ้งเตือนภัยไปยังหมายเลขปลายทาง

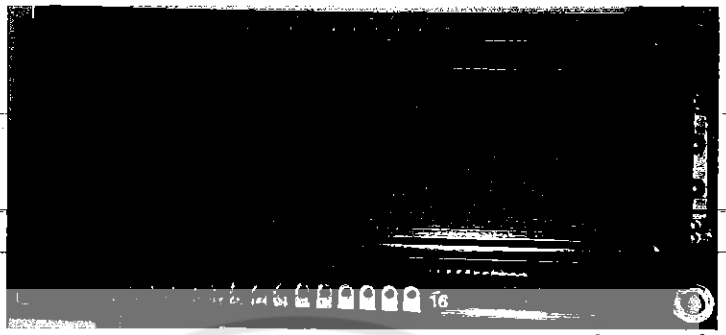
เมื่อต่ออุปกรณ์เข้าด้วยกันแล้ว ดังรูปที่ 4.2 จากนั้นจะสามารถส่งข้อความแจ้งเตือนภัยได้

รูปที่ 4.2 แสดงภาพเมื่อทำการต่ออุปกรณ์ต่างๆ เข้าด้วยกัน

เมื่อบอร์ดไมโครคอนโทรลเลอร์ได้รับสัญญาณลอกจิก 0 จากอุปกรณ์ตรวจจับความเคลื่อนไหว บอร์ดไมโครคอนโทรเลอร์ที่ได้ทำการโปรแกรมแล้ว จะทำการส่งคำสั่งให้โทรศัพท์มือถือที่ต่อพ่วงอยู่กับบอร์ดทำการส่งข้อความแจ้งเตือนภัยไปยังหมายเลขโทรศัพท์ที่ได้กำหนดไว้

4.3 ระบบแจ้งสถานะการส่งข้อความบนจอ LCD

เป็นหน้าจอ LCD แบบ 8 บิต แสดงได้ 2 บรรทัดโดยข้อความที่จะแสดงสามารถกำหนดโดยการโปรแกรมแสดงผลได้ ดังรูปที่ 4.3



รูปที่ 4.3 จอ LCD แสดงผลสถานะของอุปกรณ์ตรวจจับความเคลื่อนไหวที่พร้อมใช้งาน

ขณะที่อุปกรณ์ตรวจจับความเคลื่อนไหวพร้อมใช้งานไมโครคอนโทรลเลอร์จะสั่งให้จอ LCD แสดงผลสถานะการทำงานของอุปกรณ์ตรวจจับความเคลื่อนไหว

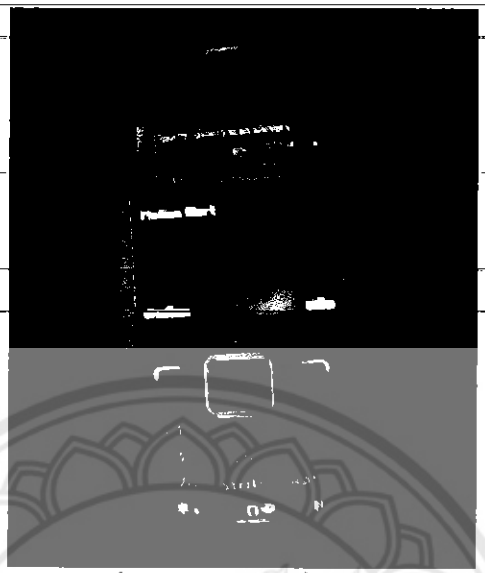
หลังจากมีการส่งข้อความแจ้งเตือนไปยังเลขหมายปลายทางหน้าจอ LCD จะเปลี่ยนสถานะ โหมดการทำงานดังรูปที่ 4.4

รูปที่ 4.4 แสดงสถานการณ์ส่งข้อความ

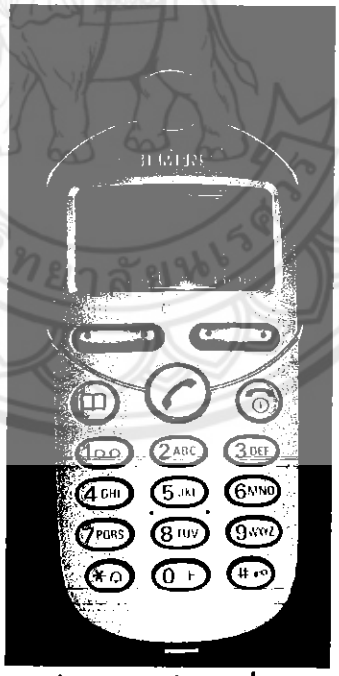
เมื่ออุปกรณ์ตรวจจับความเคลื่อนไหวทำงาน บอร์ดไมโครคอนโทรลเลอร์จะสั่งให้โทรศัพท์มือถือที่เชื่อมต่ออยู่กับบอร์ดส่งข้อความแจ้งเตือนภัยไปยังหมายเลขโทรศัพท์ที่ได้กำหนดไว้ ซึ่งจอ LCD ก็จะแสดงสถานะว่าระบบกำลังส่งข้อความแจ้งเตือนไปยังหมายเลขที่ได้กำหนด

4.4 ระบบ รับ-ส่ง ข้อความ

เป็นโทรศัพท์ที่ใช้ในการรับหรือส่งข้อความแสดงดังรูปที่ 4.5 และ 4.6



รูปที่ 4.5 แสดงโทรศัพท์ที่ใช้รับข้อความเมื่อมีการแจ้งเตือนมายังเครื่องรับ



รูปที่ 4.6 แสดงโทรศัพท์รุ่นซีเมนส์ C 35ที่ใช้ส่งข้อความแจ้งเตือนภัย

อุปกรณ์รับ-ส่งข้อความ ประกอบไปด้วย อุปกรณ์ส่งข้อความและอุปกรณ์รับข้อความ โดย อุปกรณ์ที่ใช้รับข้อความจะเป็นโทรศัพท์มือถือที่สามารถรับข้อความได้ เมื่อเครื่องส่งทำการส่ง ข้อความแจ้งเตือนภัยมายังเครื่องรับแสดงได้ดังรูปที่ 4.5 ส่วนอุปกรณ์ที่ใช้ส่งข้อความจะเป็น

โทรศัพท์มือถือรุ่นที่มีโมเด็มภายในตัวเครื่อง ซึ่งในโครงงานนี้ใช้โทรศัพท์มือถือ ซีเมนส์ C35 ในการส่งข้อความ

4.5 ผลการทดลอง

จากการทดลองใช้อุปกรณ์ตรวจจับความเคลื่อนไหวในระยะต่างๆ จะได้ผลดังนี้

ตารางที่ 4.1 แสดงผลการทดลองอุปกรณ์ตรวจจับความเคลื่อนไหวในระยะต่างๆ

ระยะทาง (เมตร)	ผลการทดลอง
2	Detect ได้
4	Detect ได้
6	Detect ได้
8	Detect ได้
10	Detect ได้บ้าง
12	Detect ไม่ได้
14 +	Detect ไม่ได้

ผลการทดลองที่ระยะ 2-8 เมตรอุปกรณ์ตรวจจับความเคลื่อนไหวสามารถ Detect สัญญาณได้ดีและที่ระยะทาง 10 เมตร สามารถ Detect สัญญาณได้บ้าง ที่ระยะมากกว่า 12 เมตร อุปกรณ์ตรวจจับความเคลื่อนไหวไม่สามารถ Detect สัญญาณได้เลย สรุปได้ว่าอุปกรณ์ตรวจจับความเคลื่อนไหวทำงานดีที่สุดที่ระยะ 2-8 เมตร

บทที่ 5

บทสรุป

ในการทำโครงงานระบบแจ้งเตือนภัยด้วยโทรศัพท์มือถือ คณะผู้จัดทำโครงงานได้แบ่งหัวข้อในบทนี้เป็น 4 หัวข้อด้วยกันคือ

5.1 สรุปผลการดำเนินงานโครงงาน

5.2 ปัญหาและแนวทางการแก้ไข

5.3 ข้อจำกัดของระบบ

5.4 ข้อเสนอแนะในการใช้งานต่อไป

5.1 สรุปผลการดำเนินงานโครงงาน

จากการทำโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ได้ข้อสรุปจากการดำเนินงานดังนี้

1. ได้ระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ ที่นำไปประยุกต์ใช้ได้จริงในอนาคต
2. ได้รับความรู้ ความเข้าใจ ในหลักการทำงานของระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ มากยิ่งขึ้น
3. ได้รับความรู้ความเข้าใจ ในระบบตรวจจับความเคลื่อนไหว , การโปรแกรมภาษา c และการเชื่อมต่อไมโครคอนโทรลเลอร์มากยิ่งขึ้น
4. ได้นำความรู้พื้นฐานที่มีอยู่เดิม และประยุกต์ใช้อุปกรณ์ที่สามารถหาทั่วไปมาใช้ให้เกิดประโยชน์มากยิ่งขึ้น

5.2 ปัญหาและแนวทางการแก้ไข

จากการทำโครงงานระบบแจ้งเตือนภัยผ่านโทรศัพท์มือถือ พบปัญหาและอุปสรรคต่างๆ ดังนี้

1. การดำเนินงานมีความล่าช้าเนื่องจากขาดความรู้ความเข้าใจในการเลือกอุปกรณ์ โดยโทรศัพท์มือถือรุ่นที่มีคุณสมบัติเหมาะสมในการพัฒนาโครงงานหาได้ยาก แก้ไขได้โดยสอบถามผู้ที่เชี่ยวชาญในด้านนี้
2. การดำเนินงานในด้านโปรแกรม บางครั้งเกิดปัญหาข้อผิดพลาดในด้านการพัฒนาโปรแกรมทำให้เกิดความล่าช้าในการดำเนินงาน แก้ไขโดยศึกษาและสอบถามผู้ที่เชี่ยวชาญในด้านนี้

5.3 ข้อจำกัดของระบบ

1. ระบบมีระยะเวลาการทำงานจำกัด ขึ้นอยู่กับพลังงานของโทรศัพท์มือถือ
2. ระบบไม่สามารถส่งข้อความได้ หากมีค่าบริการของโทรศัพท์มือถือที่ใช้ในการส่งข้อความมีไม่พอ
3. ระบบไม่สามารถหยุดการทำงานได้โดยอัตโนมัติ ซึ่งผู้ใช้จะต้องหยุดการทำงานของระบบด้วยตนเอง

5.4 ข้อเสนอแนะในการพัฒนาต่อไป

1. ในอนาคตระบบการสื่อสารไร้สายน่าจะเข้ามามีบทบาทกับชีวิตประจำวันอย่างมาก ดังนั้นการพัฒนา ระบบที่เกี่ยวข้องน่าจะเป็นประโยชน์อย่างมากเช่นกัน
2. อาจเพิ่มอุปกรณ์ตรวจจับความเคลื่อนไหวเพื่อเพิ่มประสิทธิภาพในการรักษาความปลอดภัย
3. ในอนาคตระบบโทรศัพท์มือถือน่าจะมีการเพิ่มระยะเวลาการทำงานที่นานขึ้น ดังนั้นระบบย่อม สามารถ ทำงานได้นานยิ่งขึ้นเช่นกัน
4. ในกรณีที่เครือข่ายไม่พร้อมให้บริการจะไม่สามารถส่ง SMS ได้ ควรพัฒนาด้านโปรแกรมให้ระบบ สามารถส่ง SMS ได้เมื่อเครือข่ายพร้อมให้บริการ
5. ควรเพิ่มอุปกรณ์รับข้อมูลเพิ่มเติม เพื่อให้ระบบยืดหยุ่นและสะดวกต่อการใช้งาน

เอกสารอ้างอิง

[1] "**รูปแบบการส่ง SMS ผ่าน AT COMMAND.**" [ออนไลน์].
เข้าถึงได้จาก : www.elecnet.chandra.ac.th/research/paper/2548/2548_150393211.doc.
เข้าถึงเมื่อวันที่ 2 กันยายน 2551.

[2] อุดม จีนประดับ. **ไมโครคอนโทรลเลอร์ MCS-51.** ครั้งที่ 1 .กรุงเทพมหานคร: ศูนย์ผลิตตำรา สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.2541.

[3] ธีรวัฒน์ ประกอบผล. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.**กรุงเทพมหานคร: สำนักพิมพ์ สสท.2543

[4] ทวีชัย ภูรีทิพย์. ไขปัญหา **RS-232**. กรุงเทพมหานคร: บริษัท เอช.เอ็น.กรุ๊ป จำกัด. 2538

ภาคผนวก ก

AT COMMANDS

ชุดคำสั่ง AT Commands

1. Call control

Commands	Description
ATA	Answer Command
ATD	Dial Command
ATH	Hang Up Call
ATL	Monitor Speaker Loudness
ATX	Call Progress Monitoring Control
ATO	Return To Online Data Mode
AT + CHUP	Hang Up Call
At + CLCC	List Current Calls
AT + CSNS	Single Numbering Scheme
AT + CSTA	Select type of phone number
AT + CVHU	Voice Hang up
AT + VTD	DTMF tone duration
AT + VTS	DTMF and Tone generation

2. Short Message Services – Point to Point

Command	Description
AT*E2SMSRI	Ring Indicator for SMS
AT+CGSMS	Select Service for MO SMS Messages
AT+CMGF	Message Format
AT+CMGW	Write Message to Memory
AT+CMGC	Send Command
AT+CMGS	Send Messages
AT+CMSS	Send from Storage
AT+CMGD	Delete Message
AT+CMGL	List Message
AT+CMGR	Read Message
AT+CNMI	New Message Indication to TE
AT+CMTI	New Message Indication Unsolicited Response
AT+CPMS	Preferred Message Storage
AT+CSCA	Service Center Address
AT+CSCS	Select Character Set
AT+CSDH	Show Text Mode Parameters
AT+CSMP	Set Text Mode Parameter
AT+CSMS	Select Message Service
AT*E2CMGA	Modify Message Attribute
AT*E2CMGL	List Message, without marking message Read
AT*E2CMGR	Read Message without Read mark

3. Short Message Service – Cell Broadcast

Command	Description
AT+CSCB	Select Cell Broadcast Message Type
AT*EMBOX	Mailbox Numbers
AT*EMWI	Message Waiting Indication

ภาคผนวก ข

PROGRAM CODE

โปรแกรมภาษา c ของโครงงานระบบการแจ้งเตือนภัยผ่านโทรศัพท์มือถือ

```
#include "reg52.h"
#include "absacc.h"
#include "string.h"
sbit   Piezo  =  P3^6;
bit    Light;

//------- Waiting ----------
void Wait(unsigned int x)
{unsigned int i;
    for(i=0;i<x;i++) {}
}
// ---  Delay ---------- */
void  Delay(unsigned int x )
{ unsigned int   i,j;
    for (i=0;i<x;i++)
     {
        for (j=0;j<115;j++) { }
     }
}
// ---------------------------------------
// -------  LCD  Zone ------------------
// ---------------------------------------
sbit EN_LCD    =        P0^2;
sbit RS_LCD    =        P0^1;
sbit   LIGHT    =        P0^3;
void  Wr_Register(unsigned char Data)    // Wr_Command to LCD
{unsigned char Temp;
```

```
    Temp    =    Data;
    Temp    =    Temp | 0x0F;
    P0      =    Temp;
  if (Light) { LIGHT = 0;}
    RS_LCD  =    0;
    EN_LCD  =    0;
    Delay(1);
    EN_LCD  =    1;
    Temp    =    Data;
    Temp    =  Temp << 4;
    Temp    =    Temp | 0x0F;
    P0      =    Temp;
  if (Light) { LIGHT = 0;}
    RS_LCD  =    0;
    EN_LCD  =    0;
    Delay(1);
    EN_LCD  =    1;
}
void  Wr_Char(unsigned char Data)        // Wr_ Data to LCD
{unsigned char Temp;
  Temp=      Data;
    Temp    =    Temp | 0x0F;
    P0      =    Temp;
  if (Light) { LIGHT = 0;}
    RS_LCD  =    1;
    EN_LCD  =    0;
    Delay(1);
    EN_LCD  =    1;
    Temp    =    Data;
    Temp    =  Temp << 4;
    Temp    =    Temp | 0x0F;
    P0      =    Temp;
```

```
    if (Light) { LIGHT = 0;}
    RS_LCD    =        1;
    EN_LCD    =        0;
    Delay(1);
    EN_LCD    =        1;
}
void    Init_LCD(void)          // initial  LCD Module
{
    Delay(200);
    Wr_Register(0x33);
    Wr_Register(0x32);
    Wr_Register(0x28);
    Wr_Register(0x0C);
    Wr_Register(0x06);
    Wr_Register(0x01);
    Delay(100);
}
void    Goto_XY(unsigned char x,y) // goto cursor
{
    if(x == 1)
    {
     Wr_Register(0x80+(y-1));
    }
    if(x == 2)
    {
    Wr_Register(0xC0+(y-1));
    }
   if(x == 3)
    {
     Wr_Register(0x94+(y-1));
    }
    if(x == 4)
```

```
    {
     Wr_Register(0xD4+(y-1));
    }
}
void  Wr_String(unsigned char  *dptr,Row)        // show string on LCD
{unsigned  char  i,Count;
    Count = strlen(dptr);
    Goto_XY(Row,1);
      for (i=0;i<Count;i++)
        {   Wr_Char(*dptr);
                    dptr++;
        }
 }

// ---------- Serial  Zone ----------

void  Tx_Byte(unsigned char Data)  // send  1 byte to Serial Port
{
    TI          =       1;
    SBUF        =       Data;
    TI          =       0;
    while (!TI) {}
    TI          =       0;
}
unsigned char Rx_Byte(void)        // get 1 byte from Serial Port
{unsigned char Data;
    while (!RI) {}
    RI          =       0;
    Data        =       SBUF;
    return(Data);
}
void Init_Serial(unsigned  char  br)  // Initial Serial Port
```

```
{
    SCON        = 0x50;
    TMOD        = (TMOD & 0x0F)+0x20;
    TH1         = 0xFD;
    TR1         = 1;
    EA          = 1;
  if (br == 48)   TH1  = 0xFA;
  if (br == 96)   TH1  = 0xFD;
  if (br == 192)  {PCON = 0x80;} else {PCON = 0x00;}
}
void  Tx_String(unsigned char  *dptr)      // Send String to serial Port
{unsigned char i,Count;
  Count = strlen(dptr);
  for(i=0;i<Count;i++)
   {
    Tx_Byte(*dptr);
    dptr++;
   }
}
//-----------------------------------------
//--------- Main Program ----------------
//-----------------------------------------
void  main()
{
 Init_LCD();          // Initt  LCD
 Init_Serial(192);           // Initial Serial Port Baudrete = 19200 hz
 Wr_String("  ..STAND BY..  ",1);         // Show  String

 Tx_String("ATH");                              // Hook off  Mobile Command
 Tx_Byte(0x0D); Delay(1500);
```

```
}

        while(1)
        {if((INT0 == 0)||(INT1==0))   // Check Sensor Active( 0 )
            {  Wr_String(" Motion Alert ",2);     // Show String
               Tx_Byte(0x0D);Delay(1000);
                  RI = 0;                                  // Clear Serial Flag
              Tx_String("AT+CMGS=24");  // send SMS Command  = 12 + 12 Charactor )
                  Tx_Byte(0x0D);
                  while(Rx_Byte()!='>'){}   // Check '>' Respone from Mobile to Ready
                  Delay(1000);
                  //08-72-01-38-48 : Motoin Alert
                  // Send  Data  to Mobile No. 0872013848

     Tx_String("0011000A8180271083840000AA0CCD373DFD768382ECB29C0E");
          Delay(1000);
                  Tx_Byte(0x1A); // End  Command          ctrl Z
                  Tx_Byte(0x0D);Delay(5000);
                  RI = 0;                                  // Clear  Serial Flag
          Tx_String("AT+CMGS=24");  // send SMS Command  = 12 + 12 Charactor )
                  Tx_Byte(0x0D);
                  while(Rx_Byte()!='>'){}   // Check '>' Respone from Mobile to Ready
                  Delay(1000);
                  // Send  Data  to Mobile No. 0866791320

       Tx_String("0011000A8180669731020000AA0CCD373DFD768382ECB29C0E");
           Delay(1000);
                  Tx_Byte(0x1A); // End  Command          ctrl Z
                  Wr_String("  Sent   Alert  ",2); // Show  String
                  Delay(60000);  // delay  10 sec   1000 = 1Sec
                  Wr_String("  ..STAND BY..  ",1);          // Show  String
        }
     }}
```

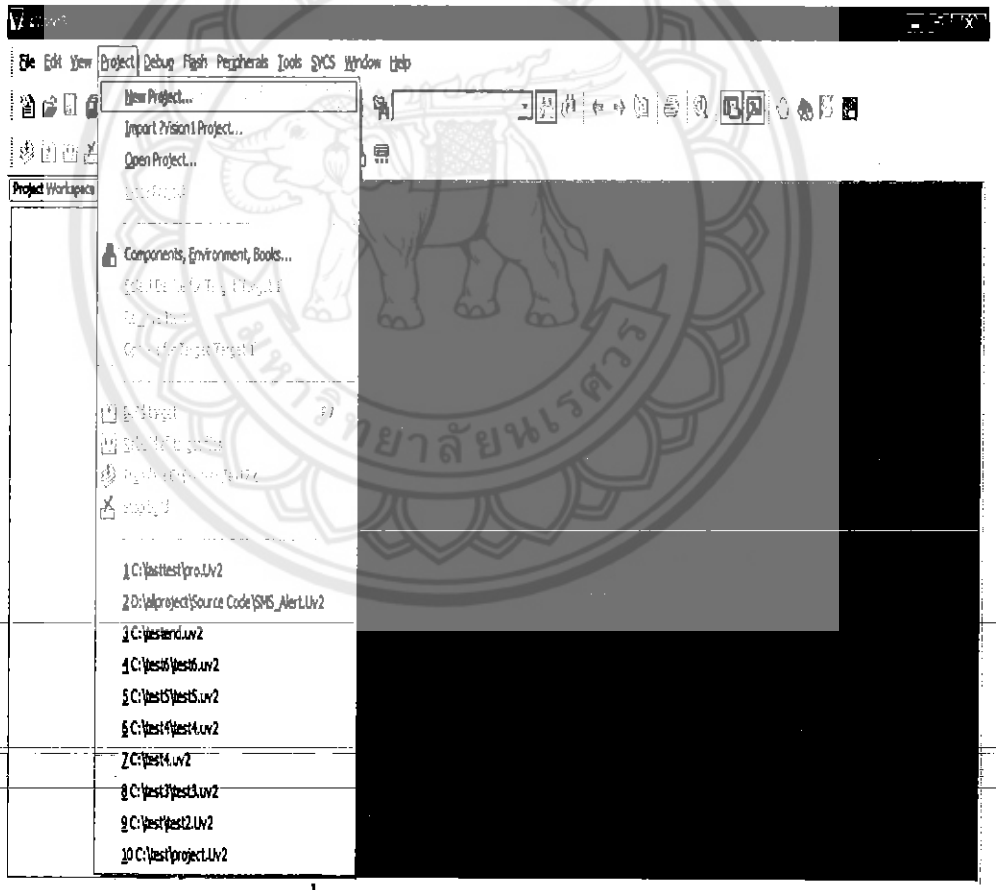
ภาคผนวก ค

การดำเนินงานของโครงงาน

1. การทดลอง Microcontroller MCS-51 (P89V51RD2)

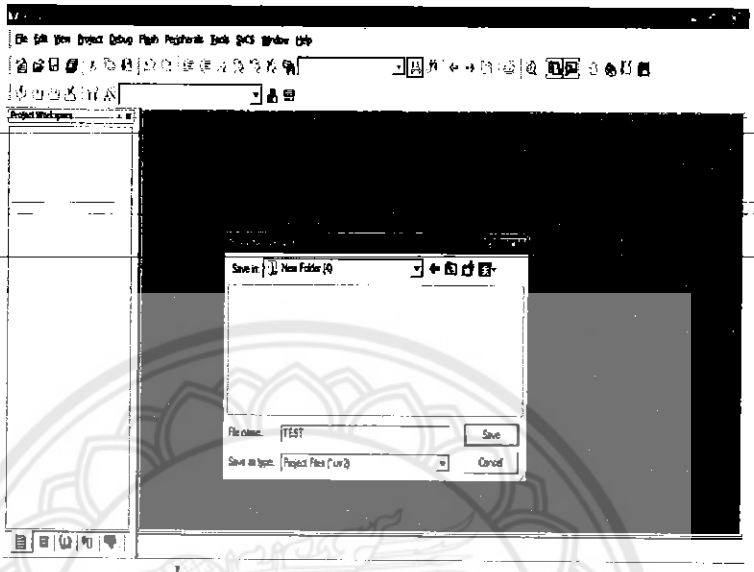
สำหรับการทดลอง Microcontroller MCS-51 (P89V51RD2) จะเป็นการทดลองการโหลด Code โปรแกรม จากเครื่องคอมพิวเตอร์ลงสู่บอร์ดไมโครคอนโทรลเลอร์ โดยสามารถติดต่อสื่อสารรับ-ส่ง ข้อมูลผ่านทางพอร์ตอนุกรม ในขั้นแรกทดสอบการใช้งานโปรแกรม Keil uVision3 ในการเขียนโปรแกรมโดยมีขั้นตอนดังนี้

1.1 เปิดโปรแกรม Keil uVision3 ขึ้นมา แล้วทำการเลือกที่ Project -> New Project ดังรูปที่ ค.1



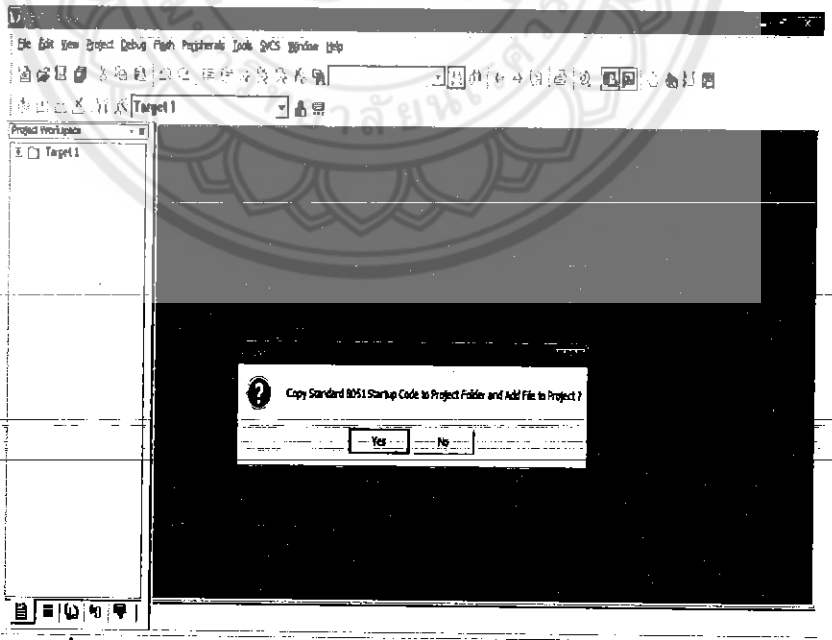
รูปที่ ค.1 แสดงการเปิดใช้งานโปรแกรม

1.2 ทำการสร้างโฟลเดอร์ขึ้นมาใหม่ ที่ Drive c แล้วตั้งชื่อ TEST จะได้ไฟล์ *.UV2 แล้วกด SAVE ดังรูปที่ ค.2



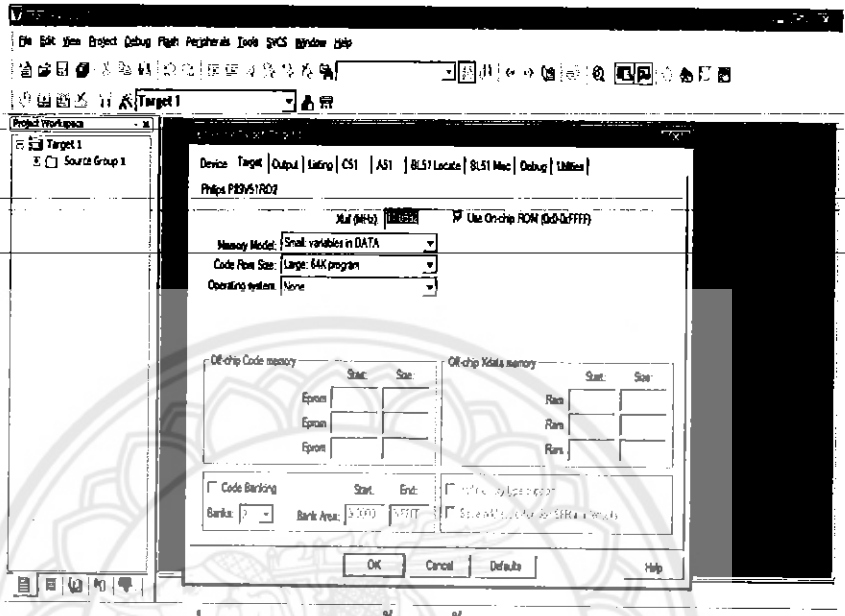
รูปที่ ค.2 แสดงการสร้างโฟลเดอร์ใหม่

1.3 เลือกเบอร์ชิพ แล้วจะมีหน้าต่าง เด้งขึ้นมาให้กด YES ดังรูปที่ ค.3



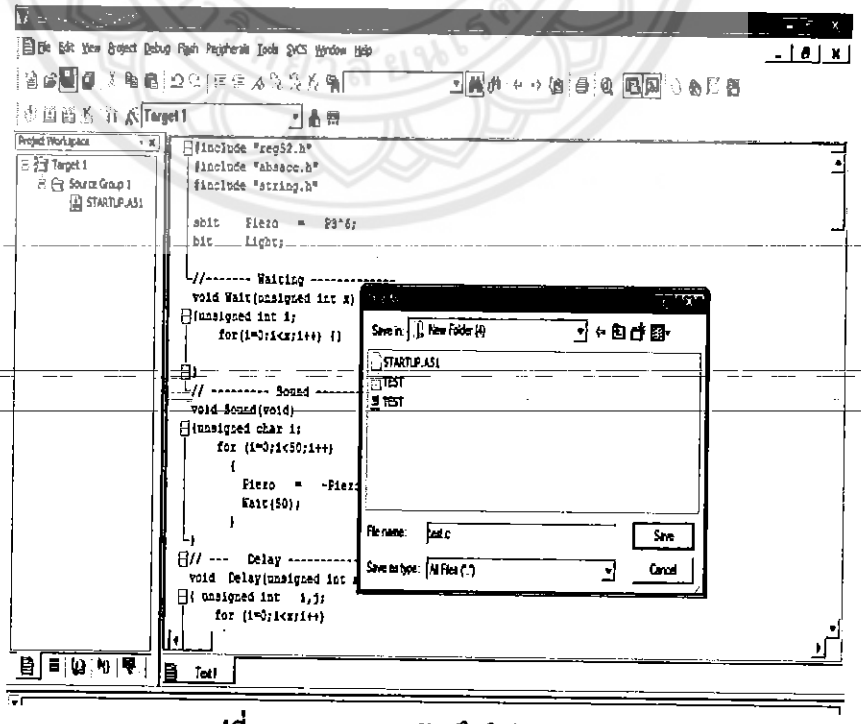
รูปที่ ค.3 แสดงการเลือกเบอร์ชิพที่ใช้กับบอร์ดไมโครคอนโทรลเลอร์

1.4 คลิกขวาที่ Target1 -> Option for Target 'Target1' แล้วทำการตั้งค่า ดังนี้ ที่แถบ Target ให้ตั้งค่า XTAL เป็น 11.0592 MHz และที่แถบ Output ให้ ทำเครื่องหมายที่ช่อง Create Hex File ดังรูปที่ ค.4



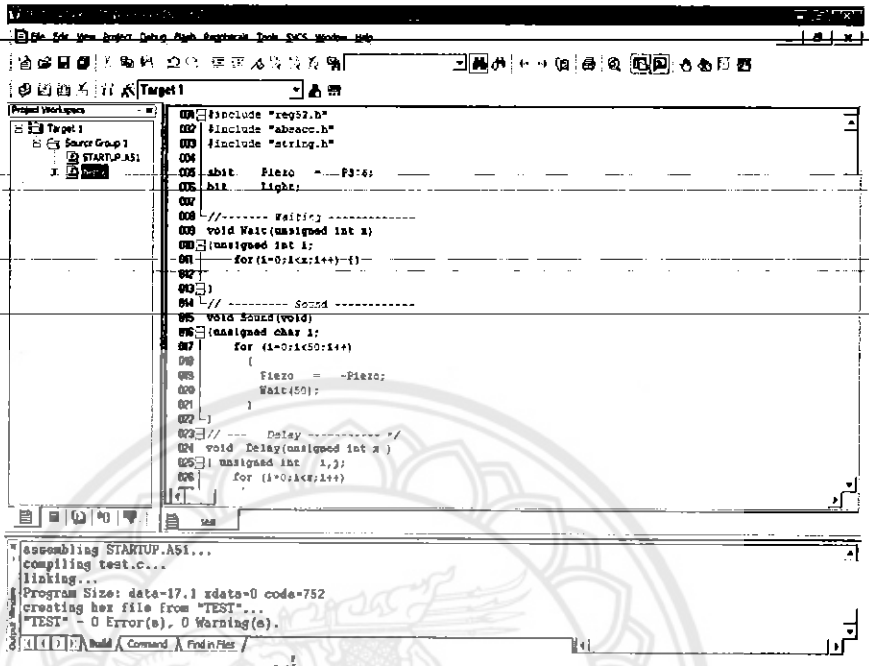
รูปที่ ค.4 แสดงการตั้งค่าเบื้องต้นของโปรแกรม

1.5 เลือก File -> New ทำการเขียนโปรแกรม หลังจากนั้นให้ save เป็น ไฟล์.c ดังรูปที่ ค.5



รูปที่ ค.5 แสดงการบันทึกโปรแกรม

1.6 คลิกขวาที่ Source Group1 แล้วเลือก Add File to group 'Source Group 1' หลังจากนั้น ทำการ Build Target แล้วจะได้ไฟล์ .Hex ขึ้นมา ดังรูปที่ ค.6



รูปที่ ค.6 แสดงการ Build Target

ต่อไปเป็นการโหลด ไฟล์.Hex ที่ได้ลงบอร์ดไมโครคอนโทรลเลอร์P89V51RD2 สำหรับในกรณีที่ใช้งานกับบอร์ดไมโครคอนโทรลเลอร์ นั้นจะสามารถทำการ Download โปรแกรมผ่านทางขั้วต่อ RS232 โดยมีขั้นตอนดังนี้

1. ต่อสายสัญญาณ DB - 9 ระหว่างคอมพิวเตอร์ PC กับ บอร์ดไมโครคอนโทรลเลอร์ที่ตำแหน่งของ ขั้วต่อสาย RS232 พร้อมกับจ่ายไฟให้กับบอร์ดเพื่อพร้อมรับคำสั่ง

2. สั่ง Run โปรแกรม Flash Magic

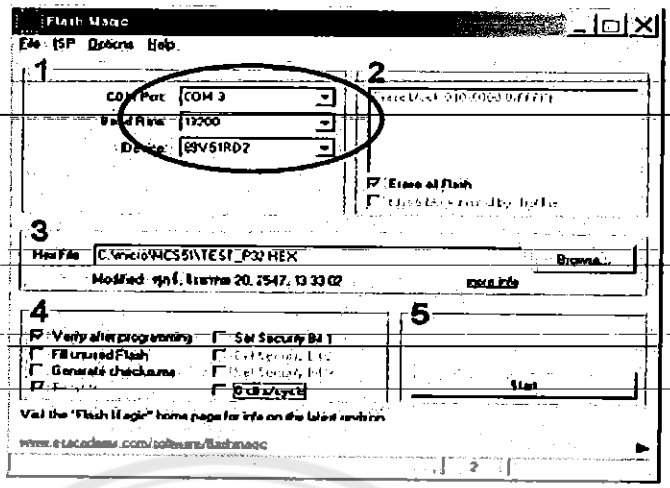
3. เลือกกำหนด Comport ตามที่ต่อสายไว้จริง และ เลือกกำหนด Baud rate เป็น 9600 หรือ 19200

4. เลือกกำหนด Device ตามที่ใช้จริง คือ P89V51RD2

5. เลือกกำหนด รูปแบบการลบข้อมูล ซึ่งถ้าไม่แน่ใจว่า CPU ถูก Lock ไว้หรือไม่ ให้เลือก Erase All Flash

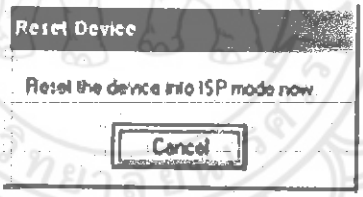
6. เลือกกำหนด Hex File ที่ต้องการ Download ตามต้องการ

7. เลือก Verify After Programming ดังรูปที่ ค.7



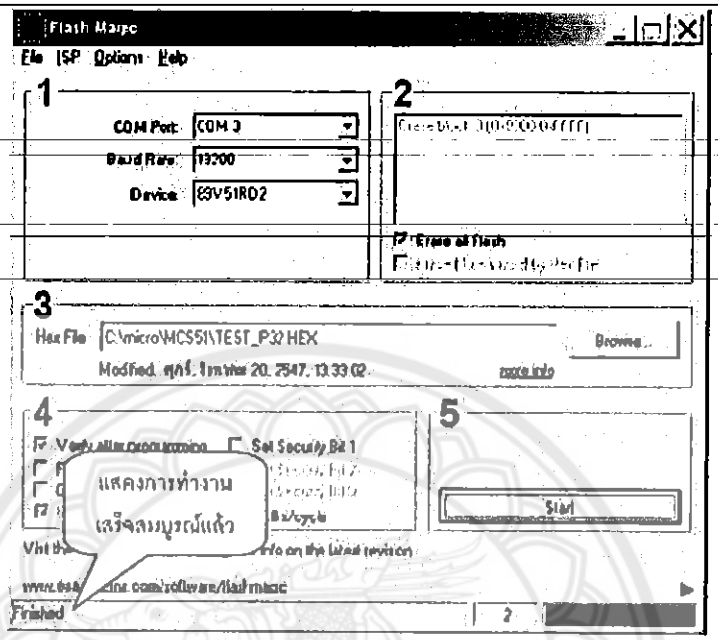
รูปที่ ค.7 แสดงการตั้งค่าในโปรแกรม Flash magic

8. เลือก Start เพื่อสั่ง Down load ข้อมูลให้กับ CPU ซึ่งจะปรากฏหน้าต่างบอกให้ RESET การทำงานของ MCU ให้เริ่มต้นทำงานใน ISP Mode ดังรูปที่ ค.8



รูปที่ ค.8 แสดงหน้าต่าง Reset

9. ให้ทำการกดสวิตช์ RESET ที่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งหน้าต่างข้างต้นจะหายไป และสังเกตเห็นโปรแกรมเริ่มต้นทำงาน ตามขั้นตอนต่างๆที่เลือกไว้ ซึ่งในขั้นตอนนี้ให้รอจนเสร็จ ดังรูปที่ ค.9



รูปที่ ค.9 แสดงการโหลดไฟล์ลงบอร์ดไมโครคอนโทรลเลอร์

การโปรแกรมลงบอร์ดก็ถือเป็นอันเสร็จสิ้น หลังจากนั้นก็จะสามารถนำบอร์ดไมโครคอนโทรลเลอร์ไปใช้งานได้

2. การเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกัน

เมื่อเชื่อมต่ออุปกรณ์ที่ใช้ในโครงงานทั้งหมดเข้าด้วยกันแล้ว จะได้ดังรูปที่ ค.10 หลังจากนั้นก็สามารถนำไปใช้งานได้

รูปที่ ค.10 แสดงการเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกัน

ประวัติผู้เขียนโครงงาน

ชื่อ นายณรงค์ศักดิ์ เทพช่วย

ภูมิลำเนา 4/101 หมู่ 1 ต.ศาลากลาง อ.บางกรวย
จ.นนทบุรี รหัสไปรษณีย์ 11130

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนบ้านส้มผ่อ
- จบประดับมัธยมศึกษาจากโรงเรียนยโสธรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **freeman_en@hotmail.com**

ชื่อ นายธีรยุทธ หอมจันทร์

ภูมิลำเนา 84 หมู่ 3 ต.หินฮาว อ. หล่มเก่า
จ.เพชรบูรณ์ รหัสไปรษณีย์ 67120

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนอนุบาลหล่มเก่า
- จบประดับมัธยมศึกษาจากโรงเรียนหล่มเก่าพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **oh_george5@hotmail.com**