ระบบเตือนภัยส่งผ่านโทรศัพท์

Security Systems by Telephone and Mobile

อนุชิต พนูวัฒนา

อรรถพ ศรีธารถกุล

เอกวิทย์ พลยมมา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

ระบบเตือนภัยส่งผ่านโทรศัพท์

Security Systems by Telephone and Mobile

อนุชิต หนูวัฒนา
อรรณพ ศรีสารสกุล
เอกวิทย์ พลยมมา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Security Systems by Telephone and Mobile

Anuchit Nhuwattana
Annop Srisarnsakul
Ekawit Ponyomma

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERRING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	ระบบเตือนภัยส่งผ่านโทรศัพท์			
	Security Systems by Telephone and Mobile			
นักศึกษาผู้จัดทำ	นายอนุชิต	หนูวัฒนา	**รหัสนักศึกษา**	52011386
	นายอรรณพ	ศรีสารสกุล	**รหัสนักศึกษา**	52011406
	นายเอกวิทย์	พลยมมา	**รหัสนักศึกษา**	52011476
ปริญญา	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมการวัดคุม			
ปีการศึกษา	2555			

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
รศ.ทรงชัย วีระทวีมาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบเตือนภัยส่งผ่านโทรศัพท์			
	Security Systems by Telephone and Mobile			
นักศึกษาผู้จัดทำ	นายอนุชิต	หนูวัฒนา	รหัสนักศึกษา	52011386
	นายอรรณพ	ศรีสารสกุล	รหัสนักศึกษา	52011406
	นายเอกวิทย์	พลยมมา	รหัสนักศึกษา	52011476
อาจารย์ที่ปรึกษา	รศ.ทรงชัย วีระทวีมาศ			
ปีการศึกษา	2555			

บทคัดย่อ

โครงงานนี้นำเสนอเรื่อง ระบบเตือนภัยส่งผ่านโทรศัพท์ เนื่องจากว่าโลกในปัจจุบันมีมิจฉาชีพที่กระทำการเกี่ยวกับการโจรกรรมมากขึ้น ประกอบกับเทคโนโลยีในด้านการติดต่อสื่อสารนั้นแพร่หลายมากขึ้นทุกวันโดยเฉพาะโทรศัพท์มือถือจนเหมือนกับเป็นปัจจัยที่ 5 ในชีวิตประจำวันของคนทั่วไป จึงได้คิดแนวทางในการที่จะนำเอาโทรศัพท์มือถือมาใช้ นอกจากนี้ระบบสัญญาณกันขโมยในปัจจุบันที่ส่วนใหญ่เป็นการส่งเสียงดังขึ้นขณะที่มีผู้บุกรุกนั้น มักจะเกิดสัญญาณดังขึ้นเองโดยไม่ได้ตั้งใจบ่อยครั้งจนคนรอบข้างไม่ให้ความสนใจอีกต่อไป รวมทั้งเจ้าของบ้านอาจจะไม่ได้ยินเสียงสัญญาณกันขโมย จากเหตุผลต่างๆดังกล่าวได้ก่อให้เกิดโครงงานระบบเตือนภัยการโจรกรรมด้วยเอสเอ็มเอสขึ้นเพื่อช่วยเพิ่มประสิทธิผลของระบบเตือนภัยโดยระบบเตือนภัยนี้สามารถส่งเอสเอ็มเอส ไปยังโทรศัพท์มือถือของเจ้าของบ้านได้โดยตรง ทันทีที่อุปกรณ์รับรู้ตรวจจับการโจรกรรมหรือการบุกรุกได้ ซึ่งอุปกรณ์รับรู้เหล่านี้จะต่อเข้ากับไมโครคอนโทรลเลอร์ ที่ทำหน้าที่ส่งคำสั่งไปยังโทรศัพท์มือถือที่ติดตั้งอยู่ในบ้านให้ส่งเอสเอ็มเอสออกไปยังโทรศัพท์มือถือที่อยู่กับเจ้าของบ้านพร้อมกับการส่งเสียงเตือนของไซเรน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Security Systems by Telephone and Mobile
Authors	Mr. Anuchit Nhuwattana
	Mr. Annop Srisarnsakul
	Mr. Ekawit Ponyomma
Thesis Advisor	Assoc.Prof. Songchai Weerathaweemas
Year	2012

ABSTRACT

This project present regarding security system with mobile phone. With the immense increase in crime rate throughout the world, In addition most of people use mobile phone. So we start to learning about basic security system with mobile that we can make it by ourselves. System of project, When sensor equipment detect to find something wrong in your home. This sensor equipment will send signal to microcontroller for command GSM module to be actived. After that GSM module will send SMS to mobile phone together with sound warning of siren. Our aim of project is to make a security system with mobile. First we want to this project can practicable. Second we want to learning regarding GSM module, microcontroller (STM 32) and Motion sensor. Next we need to decrease criminal problem, finally we hope this project will be developed in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาเป็นอย่างยิ่งจากรศ.ทรงชัย วีระทวีมาศ อาจารย์ประจำภาควิชาวิศวกรรมการวัดคุม ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ที่ได้ให้ คำแนะนำ คำปรึกษาแนวทางการแก้ปัญหาและติดตามการทำปริญญานิพนธ์อย่างใกล้ชิดตลอดมา นับตั้งแต่เริ่มต้นจนกระทั่งสำเร็จเรียบร้อยสมบูรณ์ ผู้จัดทำจึงรู้สึกซาบซึ้งในความกรุณาของท่านอาจารย์เป็นอย่างยิ่งและขอขอบพระคุณเป็นอย่างสูง ถึงแม้จะมีอุปสรรคและการทำงานที่ติดขัดอยู่บ้างแต่ทุกอย่างก็ผ่านไปได้ด้วยดี เพราะมีผู้ที่เกี่ยวข้องอีกมากมายที่ได้คอยให้ความช่วยเหลืออีกทางหนึ่ง ผู้จัดจึงทำขอขอบพระคุณผู้ที่เกี่ยวข้องทุกๆคนด้วยที่ให้ความช่วยเหลือและให้กำลังใจในการทำปริญญานิพนธ์นี้ตลอดมา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....1

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....3

2.1 AT-COMMAND.....3

2.1.1 ต้นกำเนิดของ AT-COMMAND.....3

2.1.2 AT-COMMANDที่เกี่ยวข้องกับการรับ-ส่งSMS.....3

2.2 SMS.....5

2.3 Q64.....7

2.4 ET-USB/RS232.....8

2.5 ARM Cortex-M3.....11

2.5.1 โปรเซสเซอร์ Cortex-M3 กับไมโครคอนโทรลเลอร์ Cortex-M3.....12

2.5.2 พัฒนาการสถาปัตยกรรมของ ARM โปรเซสเซอร์.....12

2.5.3 ชุดคำสั่งThumb-2:จุดเปลี่ยนสำคัญของการพัฒนาCortex-M3โปรเซสเซอร์...13

2.5.4 ไปป์ไลน์ของCortex-M3โปรเซสเซอร์.....13

2.5.5 ไดอะแกรมการทำงานของCortex-M3โปรเซสเซอร์.....13

2.5.6 ระบบไมโครคอนโทรลเลอร์Cortex-M3.....14

2.5.7 ระบบรีเซ็ตในไมโครคอนโทรลเลอร์Cortex-M3.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5.8 คุณสมบัติทางเทคนิคที่สำคัญของ STM32F103VBT6.....15
2.5.9 ไดอะแกรมการทำงานของ STM32F103VBT6.....15
2.6 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor).....18
2.7 TCRT5000.....19
2.8 ไซเรน.....19

บทที่ 3 วิธีการดำเนินงาน.....20
3.1 เตรียมอุปกรณ์ Q64 20
3.2 อุปกรณ์เชื่อมต่อร่วมกับคอมพิวเตอร์ 20
3.3 เขียนรหัสคำสั่งควบคุมโมเด็ม 20
3.4 ต่อเข้ากับอุปกรณ์ ARM Cortex-M3.....20
3.5 ติดตั้ง Motion Sensor 21
3.6 ติดตั้ง LCRT5000 21
3.7 ต่อวงจรขับไซเรน.....21
3.8 วงจรขับไซเรน.....22
3.9 บล็อกไดอะแกรมปริญญานิพนธ์ทั้งหมด.....23

บทที่ 4 ผลการทดลอง.....24
4.1 การทดลอง.....24
4.2 ผลการทดลอง.....25
4.3 สรุปผลการทดลอง.....28

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....29
5.1 สรุปผลปริญญานิพนธ์.....29
5.2 ข้อเสนอแนะ.....29

บรรณานุกรม.....30
ภาคผนวก.....32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญปริญญานิพนธ์

ทุกวันนี้เรื่องของความปลอดภัยในชีวิตและทรัพย์สิน เป็นเรื่องที่หลายๆคนให้ความสนใจเป็นอย่างยิ่ง โดยจะเห็นได้จากการที่มีเทคโนโลยีที่เกี่ยวกับเรื่องความปลอดภัยต่างๆมากขึ้น นั่นคือเรื่องของความปลอดภัยในบ้าน จากเหตุผลตรงนี้จึงมีความคิดที่จะทำโครงการที่เกี่ยวกับเรื่องการป้องกันการโจรกรรมบ้านขึ้นและนำเทคโนโลยีในปัจจุบันเข้ามาใช้ เพื่อช่วยในการแก้ปัญหาหรือทำให้ปัญหานั้นเบาบางลงไปโดยใช้การส่ง SMS เข้าโทรศัพท์มือถือของเจ้าของบ้านโดยตรง เมื่อบ้านนั้นกำลังถูกโจรกรรม

1.2 วัตถุประสงค์ปริญญานิพนธ์

1. เพื่อศึกษาหลักการทำงานของระบบการสื่อสารระหว่างโมดูลโทรศัพท์
2. เพื่อศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์
3. เพื่อศึกษาระบบเซนเซอร์ที่ใช้ในการเตือนภัยและสื่อสารกับไมโครคอนโทรลเลอร์ได้
4. เพื่อที่จะทำให้ปริญญานิพนธ์นี้สามารถนำไปใช้ประโยชน์ได้จริง

1.3 ขอบเขตปริญญานิพนธ์

สามารถทำให้โมดูลโทรศัพท์เชื่อมต่อกับไมโครคอนโทรลเลอร์ได้และสามารถส่งข้อความSMS ไปยังโทรศัพท์มือถือได้หากเซนเซอร์ตรวจจับความปลอดภัยทำงานขึ้นอีกทั้งยังทำให้ไซเรนสามารถทำงานส่งเสียงเตือนภัยขึ้น

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของระบบสื่อสาร
2. ศึกษาการเขียนโปรแกรมควบคุมตัวไมโครคอนโทรลเลอร์
3. ศึกษาระบบการทำงานของตัวเซนเซอร์ที่ใช้ในการเตือนภัยและสื่อสารกับตัวไมโครคอนโทรลเลอร์
4. ศึกษาการเชื่อมต่อระหว่างตัวอุปกรณ์ที่เชื่อมต่อกันในระบบ
5. ศึกษาการแสดงผลของแต่ละขั้นตอนในตัวอุปกรณ์แต่ละตัวที่เชื่อมต่อกันในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเรียนรู้และเข้าใจระบบการทำงานของตัวอุปกรณ์ที่ใช้ในการทำงานของระบบนี้ทุกๆอุปกรณ์
2. สามารถประยุกต์อุปกรณ์แต่ละตัวเข้ากับระบบการทำงานและการเตือนภัยประเภทต่างๆ
3. สามารถนำไปใช้ได้จริงในชีวิตประจำวันที่มีภัยอันตรายเกิดขึ้นได้ทุกเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 AT-COMMAND

คือ ชุดคำสั่งมาตรฐาน ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็ม หรืออุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่าหรือสั่งงานอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ และสำหรับการติดต่อกับโทรศัพท์มือถือ จะใช้ชุดคำสั่งที่เรียกว่า GSM AT-COMMAND ตัวอย่างคำสั่งที่เป็น BASIC AT-COMMAND; AT ตรวจสอบความพร้อมของอุปกรณ์ถ้าสามารถติดต่อกันได้ อุปกรณ์จะตอบกลับมาว่า OK ATDTphonenumber โทรไปยังเลขหมายปลายทาง (phonenumber) ATH วางสาย ATA รับสาย

2.1.1 ต้นกำเนิดของ AT-COMMAND

คือ รหัสคำสั่งควบคุมโมเด็มซึ่งมีต้นกำเนิดมาจากโมเด็มของบริษัทเฮย์ (Hayes) คำสั่งส่วนใหญ่จะขึ้นต้นด้วยรหัส AT เช่น ATDT หรือ ATZ

2.1.2 AT-COMMAND ที่เกี่ยวข้องกับการรับ-ส่งSMS

ชุดคำสั่ง AT COMMAND ที่ใช้กับโทรศัพท์มือถือได้มีอยู่มากมาย ทั้งการอ่านรุ่นโทรศัพท์มือถือ,ตรวจสอบระดับแบตเตอรี่, ตรวจสอบระดับสัญญาณ แต่ในที่นี้จะกล่าวถึงเพียงคำสั่งที่เกี่ยวข้องกับการรับ-ส่งSMSเท่านั้น

1) Message Format (AT+CMGF) เป็นคำสั่งกำหนดรูปแบบของข้อความที่จะให้แสดงออกมา โดย

AT+CMGF=1คือแสดงข้อความในรูปแบบTEXT

AT+CMGF=0คือแสดงข้อความในรูปแบบPDUCODE

2) List Message (AT+CMGL) เป็นคำสั่งที่ให้แสดงข้อความในสถานะต่างๆ โดยจะแสดงข้อความทั้งหมดมีลักษณะการใช้คำสั่งดังนี้

AT+CMGL=0คือแสดงข้อความที่ได้รับแต่ยังไม่ได้อ่าน(“RECUNREAD”)

AT+CMGL=1คือแสดงข้อความที่ได้รับและอ่านแล้ว(“RECREAD”)

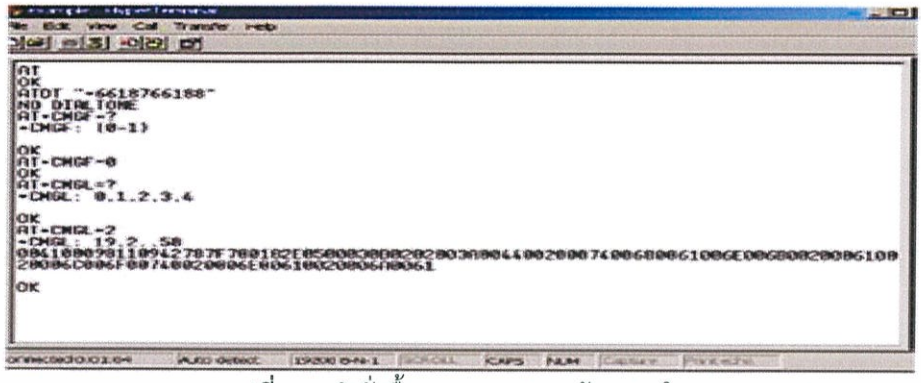
AT+CMGL=2คือแสดงข้อความที่เก็บไว้และยังไม่ได้ส่ง(“STOUNSENT”)

AT+CMGL=3คือแสดงข้อความที่เก็บไว้และส่งออกไปแล้ว(“STOSENT”)

AT+CMGL=4คือแสดงข้อความทั้งหมด(“ALL”)

หมายเหตุ หากกำหนด Message Format เป็น PDU CODE จะต้องเลือกสถานะโดยใช้ตัวเลข 0 ถึง 4 แต่หากกำหนด Message Format เป็น Text จะต้องเลือกสถานะโดยใช้ตัวอักษรที่วงเล็บด้านหลัง

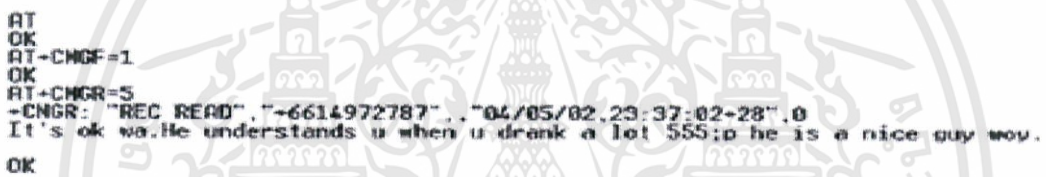
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 คำสั่งพื้นฐานและ LIST ข้อความใน STO SENT

ที่มา : http://www.narisa.com (2548)

3) Read Message (AT+CMGR) เป็นคำสั่งที่ใช้อ่านข้อความที่เฉพาะเจาะจงได้โดยระบุตำแหน่งที่ข้อความนั้นถูกเก็บไว้



รูปที่ 2.2 ตัวอย่างการอ่านข้อความ

ที่มา : http://www.narisa.com (2548)

4) Send Message (AT+CMGS= "XX") เป็นคำสั่งที่ใช้สำหรับส่งข้อความซึ่ง "XX" คือจำนวน Octet ของเลขฐาน 16 ที่ต้องการจะส่งทั้งหมด ยกเว้น Octet แรกที่เป็น "00"

Basic-transmission-of-SMS

A script which sets the module up is shown below: AT+CSDH=1 Enable the use of text mode parameters OK AT+CSMP=17,167 Set text mode parameter OK AT+CMGF=1 Switch the module to text mode OK AT+CMGS="07787154042" Sending SMS

> Test SMS The text must be ended, as in PDU mode, with a control-Z character +CMGS: 204 Confirmation that the SMS has been sent successfully OK

This text comes from GR47, GM47 AT-Commands Manual.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 SMS

รูปที่ 2.3 SMS

ที่มา : www.smsest.com (2556)

SMS ย่อมาจากคำว่า Short Message Service หรือเป็นบริการส่งข้อความสั้นๆ ลักษณะการใช้งานคล้ายกับการส่งอีเมล์แต่จะสามารถส่งข้อความได้ไม่เกิน 160 ตัวอักษรผ่านทางโทรศัพท์มือถือ

<u>จุดเด่นของบริการSMS</u>

สามารถส่งไปยังผู้รับโดยไม่ต้องกังวลว่าพื้นที่ของผู้รับจะมีสัญญาณหรือไม่ ในขณะนั้น หากทางปลายทางไม่มีสัญญาณระบบ SMS นี้จะเก็บข้อมูลไว้จนกว่าปลายทางมีสัญญาณทางระบบจึงจะทำการส่งข้อมูลไปในทันที นอกจากนี้แล้ว SMS ยังสามารถส่งข้อความที่ได้รับมาต่อไปยังหมายเลขอื่นๆได้อย่างไม่จำกัดอีกด้วย

<u>วิวัฒนาการของการส่งSMS</u>

เป็นที่ทราบกันดีอยู่แล้วว่าประเทศตะวันตกนั้นเป็นผู้พัฒนาโทรศัพท์มือถือขึ้น ฉะนั้นในยุคแรก ๆ ก็จะมีแต่การส่งความเป็นภาษาอังกฤษเท่านั้น แต่ถึงกระนั้นก็ยังมีการคิดค้นวิธีการส่งข้อความรูปแบบใหม่ๆ ไม่ว่าจะเป็นการใช้สัญลักษณ์ต่าง ๆ ที่อยู่ในเครื่อง มาทำเป็นตัวการ์ตูน หน้าคนที่แสดงอารมณ์ต่างๆ(Emoticon)และเริ่มมีการใช้ “คำย่อ” เพื่อเป็นการประหยัดเนื้อที่ในการส่ง SMS (SMS Abbreviation)ที่ส่งได้เพียง160 ตัวอักษร ต่อการส่ง 1 ครั้ง จนเป็นที่นิยมกับผู้ใช้มือถือทั่วไป

<u>ตัวอย่างสัญลักษณ์</u>

ส่งจูบ	:*)
คนผมหยิก	@:-)
หัวเราะ	:-D
ล้อเล่นนะ	;-)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างคำย่อ

AND (และ)			ก็จะย่อเป็น		N
Are	you	okay?	ก็จะย่อเป็น	ru	ok?
Kiss			ก็จะย่อเป็น		x

จนมาถึงยุคหลัง ๆ ที่โทรศัพท์มือถือได้รับความนิยมขึ้นเรื่อย ๆ สิ่งหนึ่งที่ผู้พัฒนาให้ความสำคัญก็คือ ทำอย่างไรให้ผู้ใช้มือถือชาวไทย สามารถส่งข้อความเป็นภาษาไทยได้ ในแรกเริ่มการส่งข้อความเป็นไปอย่างไม่สะดวกนัก เพราะมีการเรียงลำดับตัวอักษรภาษาไทย เหมือนกับภาษาอังกฤษ ในปุ่มกด 1 ปุ่ม (Alpha Numeric) ก็จะมีทั้ง ตัวเลข และ ตัวอักษรภาษาไทย เช่น ปุ่ม 2 ก็จะเป็น ตัว ฆ-ฉ |หรือ A-B-C ในภาษาอังกฤษ

รูปที่ 2.4 Alpha Numeric

ที่มา : www.barcodedatalink.com (2555)

สื่อโทรศัพท์มือถือ

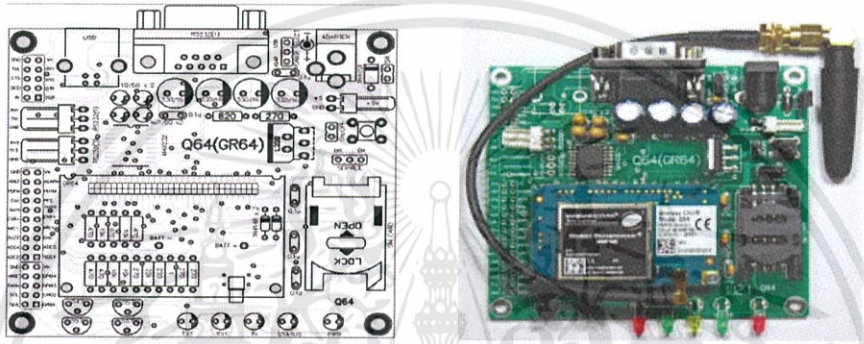
สื่อโทรศัพท์มือถือ คือ การรับ-ส่งสารผ่านช่องทางโทรศัพท์มือถือนั่นเองแต่สิ่งที่น่าสังเกต ก็คือ สื่อโทรศัพท์มือถือ นอกจากจะเป็นการสื่อสารระหว่างบุคคล (Interpersonal Communication) แล้ว ยังสามารถสื่อสารในระดับมวลชน (Mass Communication) ได้อีกด้วย โดยที่จะเป็นการเข้าถึงมวลชนใน ระดับรายบุคคล (One-to-one Communication)ได้ ซึ่งถือเป็นเครื่องมือสื่อสารชนิดแรกที่มีศักยภาพเพียงพอที่จะรวบเอาการสื่อสารในทุกระดับไว้ด้วยกัน โดยอาศัยเอกลักษณ์อันโดดเด่นของตัวเอง คือความเป็นมัลติมีเดีย อินเตอร์แอคทีฟ และขนาดที่เล็กจนสามารถพกพาไปไหนต่อไหนได้ ที่สำคัญยิ่งไปกว่า นั่นก็คือนอกจากโทรศัพท์มือถือจะสามารถทำตัวเป็น "สื่อ" ด้วยตัวของมันเองแล้วยังกลายเป็นศูนย์รวมของสื่ออื่นๆ มากมาย (Media Convergence) ทั้งสื่อสิ่งพิมพ์ สื่ออิเล็กทรอนิกส์ (โทรทัศน์ วิทยุ) หรือแม้แต่สื่อใหม่อย่างอินเทอร์เน็ต ดังคำกล่าวที่ทุกท่านคงจะเคยได้ยินกันอยู่บ่อยๆ ว่า "ไม่มีอะไรที่โทรศัพท์มือถือทำไม่ได้" เพราะ ณ ปัจจุบันนี้ การรับชมโทรทัศน์ ฟังวิทยุอ่านหนังสือพิมพ์ นิตยสาร หรือแม้กระทั่งเล่นอินเทอร์เน็ต ก็สามารถทำได้ครบครันผ่านหน้าจอเล็กๆ ของโทรศัพท์มือถือ สิ่งนี้เองที่ถือเป็นปรากฏการณ์ครั้งสำคัญที่จะต้องวิเคราะห์ถึงอัตลักษณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และบทบาทของสื่อใหม่นี้อย่างใกล้ชิด ในเบื้องต้นจะอยู่ในรูปแบบของตัวอักษร ผ่านทางข้อความสั้นไปยังโทรศัพท์มือถือ หรือที่เรียกกันว่า เอสเอ็มเอส มาร์เก็ตติ้ง (SMS Marketing)

2.3 Q64

เป็นโมดูล GSM/GPRS ที่รองรับความถี่ได้แก่ 850,900,1800,1900 MHZ โดยทางบริษัท ศิลารีเสิร์ช ได้นำตัวโมดูล Q64 มาทำในส่วนของบอร์ดเพิ่มเติม ได้แก่ วงจรในส่วนของ Supply, RS-232 ช่องสำหรับใส่ Sim card เพื่อให้สะดวกต่อการนำไปใช้งาน



รูปที่ 2.5 ภาพบอร์ดโมดูลโทรศัพท์ Q64 GSM/GPRS
ที่มา : บริษัท ศิลารีเสิร์ช (2556)

คุณสมบัติทั่วไป

รองรับระบบ GSM/GPRS ที่ความถี่ 850,900,1800,1900 MHz
รองรับระบบโปรโตคอลแบบ TCP/IP
มีช่องสำหรับใส่ Sim card รองรับทั้ง 3 ระบบ (AIS,DTAC,TRUEMOVE)
มีไฟแสดงสถานะ POWER,STATUS,RI,RX,TX
มีสวิตช์ ON-OFF สำหรับเปิด ปิด การทำงานของโมดูล
มีวงจร RS-232 และขั้วต่อสำหรับต่อกับคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์เพื่อสั่งงานด้วย AT Command

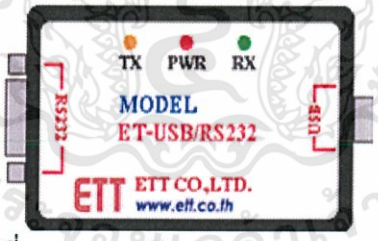
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ET-USB/RS232

รูปที่ 2.6 ET-USB/RS232
ที่มา : อรรณพ (2555)

คุณสมบัติของ ET-USB/RS232

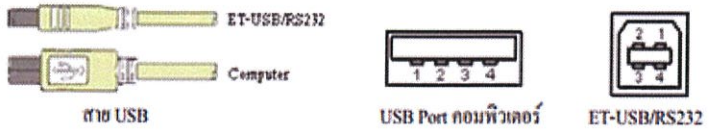
1) USB 1.1 and USB 2.0 Compatible
2) Data Rate 250 kbps (สามารถใช้งานได้ตั้งแต่ Baud Rate 300 BPS ถึง 128 KPBS)
3) USB CONNECTOR เป็นแบบ TYPE B, RS232 CONNECTOR เป็น TYPE DB 9 PIN ตัวผู้
4) 384 Byte Receive Buffer / 128 Byte Transmit Buffer ส่งผ่านข้อมูลความเร็วสูง
5) ใช้ไฟเลี้ยงวงจรจาก USB Port ได้โดยตรง ไม่ต้องต่อเพิ่มจากภายนอก
6) แสดงสถานการณ์ทำงานด้วย LED 3 สีคือการรับ (RX) สีเขียว การส่ง (TX) สีเหลือง และ Power (PWR) สีแดง
7) ขนาดตัวกล่อง 7.5 x 2.5 x 5 CM พร้อมสาย USB CABLE TYPE A TO TYPE B



รูปที่ 2.7 คุณสมบัติของ ET-USB/RS232
ที่มา : บริษัทอีทีที จำกัด (2556)

<u>การต่อใช้งาน</u>

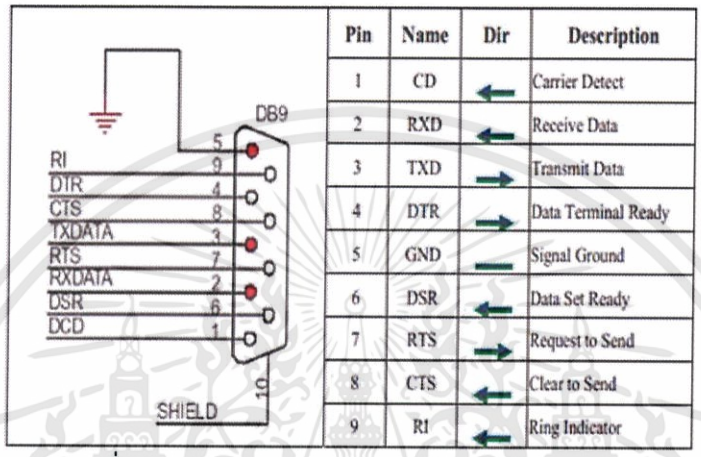
1.เสียบสาย USB เข้ากับ Port USB ของคอมพิวเตอร์และ Port USB ของ ET-USB/RS232 ให้ถูกต้อง ขณะนี้สังเกตไฟ แสดงสถานะ PWR จะยังไม่ติดสว่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ติดตั้ง Driver สำหรับ ET-USB/RS232 โดยดูจากคู่มือการติดตั้งไดร์ฟเวอร์ เมื่อติดตั้งเรียบร้อยแล้ว ไฟแสดงสถานะ PWR จะติดสว่าง แสดงว่าพร้อมใช้งานแล้ว

3. ต่อสาย Serial Port RS232 เพื่อใช้งาน วิธีการต่อใช้งานดังรูป



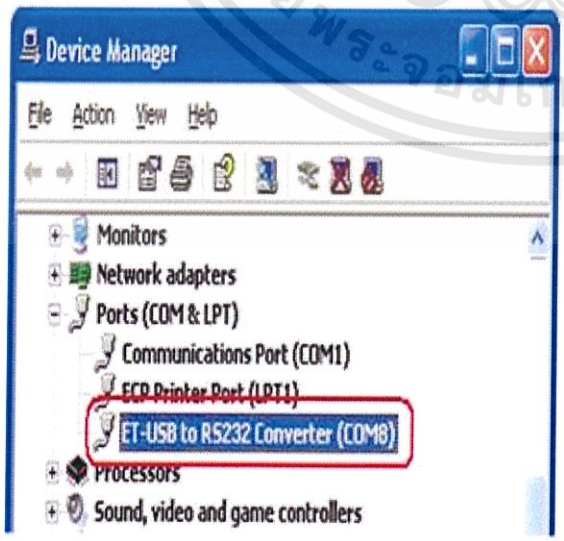
Pin	Name	Dir	Description
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	—	Signal Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

รูปที่ 2.8 PIN D-SUB MALE at the ET-USB/RS232

ที่มา : StarTech.com (2555)

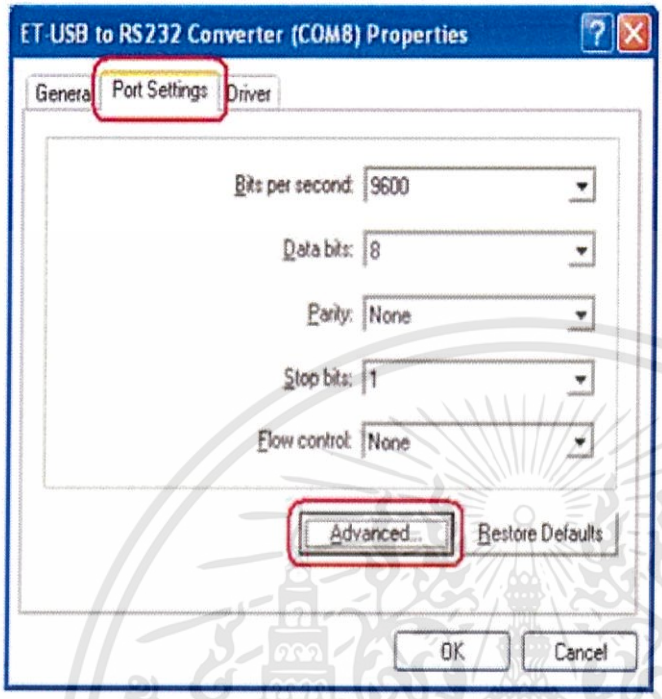
<u>การเปลี่ยนหมายเลข COM PORT</u>

1. ไปที่ Control Panel> System เลือกแท็ป Device Manager แล้วดับเบิลคลิกที่ ET-USB to RS232 Converter (COM8) (หมายเลข COM Port อาจมีการเปลี่ยนแปลง ซึ่งขึ้นอยู่กับคอมพิวเตอร์แต่ละเครื่อง) จะปรากฏ ไดอะล็อกบล็อกดังรูป

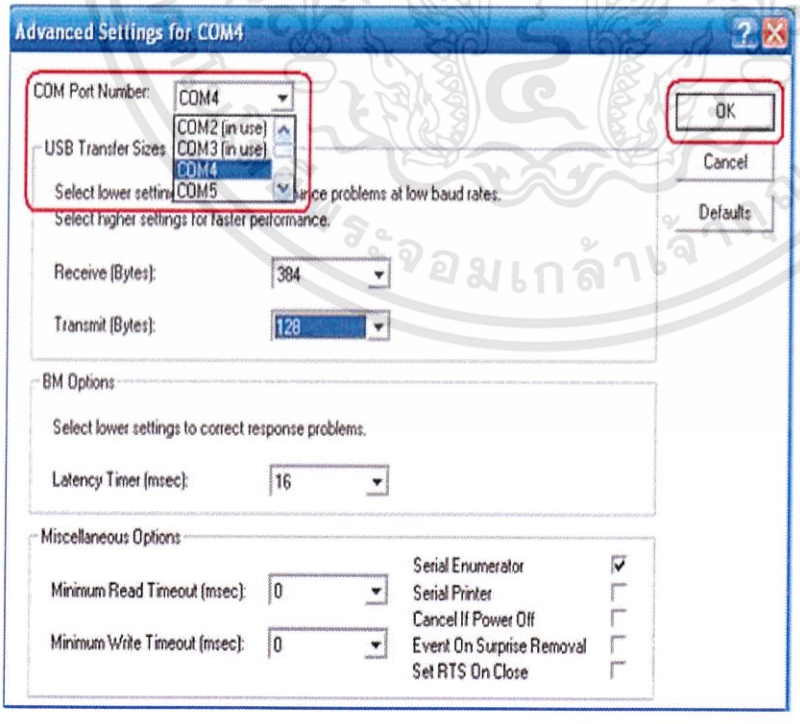


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แล้วเลือกที่ Advanced

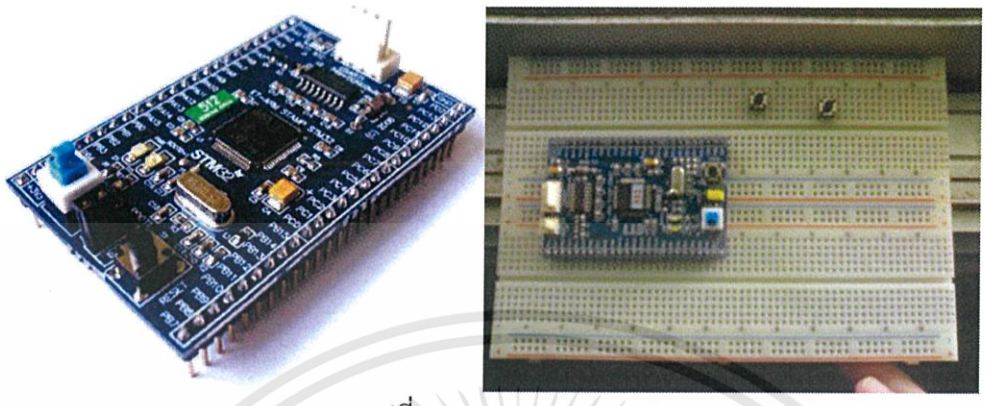


3.การเลือกเปลี่ยนพอร์ทโดยพอร์ตที่จะเปลี่ยนต้องยังไม่มีการใช้อยู่หรือใช้แบบที่ไดร์ฟเวอร์กำหนดให้ก็ได้จากนั้นก็คลิก OK เพื่อยืนยันอีกครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ARM Cortex-M3



รูปที่ 2.9 ARM Cortex-M3
ที่มา : http://en.wikipedia.org (2556)

ARM Cortex-M3 เป็นชื่อของโปรเซสเซอร์ 32 บิตที่พัฒนาภายใต้สถาปัตยกรรม ARMv7-M มีคุณสมบัติที่น่าสนใจคือ

1) เป็นโปรเซสเซอร์แบบไฮเออราชิคอล (Hierachical) ที่รวมเอาซีพีคอร์เข้ากับระบบเพอริเฟอรัลขั้นสูง
2) ซีพียูคอร์เป็นสถาปัตยกรรมแบบฮาร์วาร์ด (Havard architecture) ไปร์ไลน์ 3 ระดับพร้อมรองรับการกระโดดไปทำงานอย่างทันทีทันใด รองรับคำสั่ง Thumb และ Thumb-2
3) หน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) มีโมดูลหารแบบฮาร์ดแวร์ (Hardware divider)
4) สามารคูณเลข 16 และ 32 บิตได้อย่างรวดเร็วเพียงหนึ่งไซเคิลของสัญญาณนาฬิกา
5) มีตัวควบคุมการอินเตอร์รัปต์แบบกำหนดค่าได้รองรับการเกิดอินเตอร์รัปต์ได้ถึง 240 อินเตอร์รัปต์
6) มีตัวควบคุมเวกเตอร์อินเตอร์รัปต์ซ้อน (NVIC :Nested Vectored Interrupt Controller) สามารถกำหนดนัยสำคัญได้ถึง 216 ระดับ
7) ใช้ระบบบัสแบบเมตริกซ์
8) มีไทเมอร์ของระบบที่ชื่อ SysTick สำหับรองรับการทำงานกับระบบปฏิบัติการเวลาจริงหรือ RTOS (Real-Time OS)
9) ใช้จำนวนขาในการดีบักทางฮาร์ดแวร์น้อยจากปกติ 9 ขาเหลือเพียง 2-3 ขา
10) ประสิทธิภาพความเร็วในการประมวลผล 1.2 DMIPS/MHz ซึ่งดีกว่า ARM7TDMI-S 70%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 โปรเซสเซอร์ Cortex-M3 กับไมโครคอนโทรลเลอร์ Cortex-M3

โปรเซสเซอร์ Cortex-M3 เป็นซีพียูที่ได้รับการออกแบบและพัฒนาจาก ARM Limited ในประเทศอังกฤษ โดยเพิ่มความจำ อุปกรณ์เพอริฟอรัล ระบบสัญญาณนาฬิกา และระบบอินพุตเอาต์พุตทำให้กลายเป็นไมโครคอนโทรลเลอร์ Cortex-M3

2.5.2 พัฒนาการสถาปัตยกรรมของ ARM โปรเซสเซอร์

ปี ค.ศ. 2006 ARM ได้พัฒนาคอร์ของซีพียู 32 บิต ด้วยการรองรับชุดคำสั่ง Thumb-2 ให้แก่คอร์ซีพียูตัวใหม่ ทำให้บริหารจัดการงานคำสั่งได้ดีขึ้น ลดจำนวนอุปกรณ์และพลังงานที่ใช้ลง แต่ความเร็วในการทำงานสูงขึ้น ทั้งยังมีการจัดกลุ่มซีพียูคอร์ใหม่เพื่อให้สามารถรองรับกับการประยุกต์ใช้งานที่ตรงจุดมากขึ้น ทั้งนี้เพื่อให้บรรลุถึงเป้าหมายคือ ซีพียูที่มีความสามารถสูงในราคาที่ต่ำที่สุด โดยมีการแบ่งออกเป็น 3 แบบหลักๆดังนี้

แบบ A หรือโปรไฟล์ A

เป็นซีพียูที่เน้นให้ใช้เป็นแอปพลิเคชันสำหรับงานที่มีความซับซ้อนสูงมากเช่นระบบปฏิบัติการ (OS: Operating System) เช่น Symbian โดยซีพียูสำหรับงานในลักษณะนี้ต้องมีความสามารถในการประมวลผลข้อมูลสูงสุด ระบบหน่วยความจำเสมือนหรือ Virtual memory system ต้องสามารถรองรับหน่วยบริหารหน่วยความจำหรือ MMU (Memory Management Unit) ได้เป็นอย่างดี ตัวอย่างของผลิตภัณฑ์ คือ โทรศัพท์เครื่องที่สมาร์ทโฟนและพ็อกเก็ตพีซี

แบบ R หรือโปรไฟล์ R

เป็นซีพียูที่รองรับการทำงานแบบเรียลไทม์ความเร็วสูง ซีพียูต้องมีเสถียรภาพและความน่าเชื่อถือในการทำงานระดับสูง ตัวอย่างการพัฒนาชิปไปสู่การใช้งานจริง คือ ชิปควบคุมฮาร์ดดิสก์ เนื่องจากฮาร์ดดิสก์ในปัจจุบันมีความจุสูง ดังนั้นการควบคุมระบบหยุดและเคลื่อนที่ของหัวอ่านต้องมีความเร็วและความแม่นยำสูงเพื่อให้สามารถเข้าถึงข้อมูลได้อย่างถูกต้องรวดเร็ว

แบบ M หรือโปรไฟล์ M

เป็นซีพียูที่ออกแบบมาเพื่อต้องการให้นำไปพัฒนาผลิตภัณฑ์ที่ต้องการความสามรถในการประมวลผลสูง งบประมาณต่ำ ใช้พลังงานต่ำ มีวัตถุประสงค์แต่ละแอปพลิเคชันที่เจาะจงอย่างชัดเจน ซึ่งนั่นก็คือ วัตถุประสงค์ในการทำงานของไมโครคอนโทรเลอร์นั่นเอง ซีพียูนั้นเหมาะสำหรับการนำไปพัฒนาเป็นชิปไมโครคอนโทรลเลอร์ความสามารถสูงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 ชุดคำสั่งThumb- 2: จุดเปลี่ยนสำคัญของการพัฒนาCortex-M3โปรเซสเซอร์

ARM ได้พัฒนาชุดคำสั่ง Thumb-2 ขึ้นมาตังแต่ปี2003 ซึ่งเป็นชุดคำสั่งที่สามรถรองรับการทำงานกับคำสั่งThumb ทั้ง16และ32บิต จุดมุ่งหมายสำคัญของการพัฒนาชุดคำสั่งนี้คือ เป็นคำสั่งที่ใช้งานง่าย มีความต้องการหน่วยความจำน้อย และเป็นคำสั่งที่มีประสิทธิภาพสูง

เหตุผลหลักประการหนึ่งของการพัฒนาชุดคำสั่งThumb-2คือต้องการให้ซีพียูARMv7-M สามารถทำงานกับคำสั่ง32บิตได้ โดยใช้หน่วยความจำที่น้อยและคล่องตัวกว่า

Cortex-M3โปรเซสเซอร์จึงเป็นโปรเซสเซอร์ที่มีความพร้อมในการรองรับชุดคำสั่งThumb-2 ได้อย่างสมบูรณ์นั่นจึงเป็นสาเหตุสำคัญที่ทำให้Cortex-M3โปรเซสเซอร์มีความแตกต่างจากARM โปรเซสเซอร์ดั้งเดิมเนื่องจากชุดคำสั่งThumb-2รองรับการทำงานทั้งคำสั่ง16และ32บิตทำให้Cortex-M3ไม่ต้องสลับการประมวลผลคำสั่งไปมาระหว่างชุดคำสั่งThumbมาตรฐาน(16บิต)กับARM(32บิต) ส่งผลให้การทำงานเร็วขึ้นมีค่าหน่วงเวลาสำหรับการประมวลผลลดลงสามารถสร้างโปรแกรมที่ซับซ้อนด้วยขนาดของหน่วยความจำที่น้อย

2.5.4 ไปป์ไลน์ของCortex-M3โปรเซสเซอร์

Cortex-M3โปรเซสเซอร์ มีไปป์ไลน์ 3 ช่วงประกอบด้วยช่วงอ่านหรือเฟตช์ (fetch) คำสั่งช่วงถอดรหัสคำสั่ง (decode)และช่วงเอ็กซิคิวต์(execute)หรือกระทำคำสั่ง

2.5.5 ไดอะแกรมการทำงานของCortex-M3โปรเซสเซอร์

ระบบของCortex-M3โปรเซสเซอร์ มิได้มีเพียงโปรเซสเซอร์ ARMv7-Mเท่านั้น ยังประกอบด้วยอุปกรณ์ที่ใช้ในการจัดการระบบจำนวนมากรวมถึงการรองรับกระบวนการดีบักด้วยและสามารถแบ่งได้3ส่วนหลักๆคือส่วนของระบบโปเซสเซอร์,ส่วนของระบบดีบักและบัสเชื่อมต่อของระบบ

ส่วนประกอบในระบบโปรเซสเซอร์ของCortexM3โปรเซสเซอร์ มีดังนี้

1) **Cortex-M3คอร์**:ประกอบด้วยรีจีสเตอร์,หน่วยคำนวณทางคณิตศาสตร์และลอจิก,บัสข้อมูล ,บัสคำสั่งและส่วนเชื่อมต่อระบบบัส
2) **ส่วนควบคุมอินเตอร์รัปต์**(NVIC:Nested Vectored Interrupt Controller):เป็นส่วนจัดการอินเตอร์รัปต์ทั้งหมดสามารถเปลี่ยนแปลงได้ในไมโครคอนโทรลเลอร์จากแต่ละผู้ผลิต
3) **SYSTICKไทเมอร์ระบบ**:เป็นไทเมอร์นับถอยหลังใช้ในการสร้างสัญญาณอินเตอร์รัปต์ตามช่วงเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) **ส่วนป้องกันหน่วยความจำMPU** (Memory Protection Unit):ใช้ในการป้องกันข้อมูลในหน่วยความจำภายในที่สำคัญของซีพียูเป็นส่วนเสริมที่อาจไม่ได้มีในไมโครคอนโทรลเลอร์ Cortex-M3ทุกรุ่นทุกเบอร์
5) **บัสเมตริกซ์**(BusMatrix):เป็นหัวใจของระบบบัสภายในของCortex-M3โปรเซสเซอร์เนื่องจากเป็นส่วนจัดการเชื่อมต่อทั้งหมดของบัสซีพียู(ABH:Advanced High-performance Bus)ในการถ่ายทอดข้อมูลจากส่วนต่างๆของโปรเซสเซอร์
6) **ส่วนเชื่อมต่อบัสซียูกับบัสเพอริเฟอรัล**(ABH-to-APBbusbridge):ใช้ในการเชื่อมต่ออุปกรณ์เพอริเฟอนัลเข้ากับบัสซีพียู

ส่วนประกอบในระบบดีบักและเทรซของCortex-M3โปรเซสเซอร์ มีดังนี้

1) ส่วนเชื่อมต่อJTAG
2) พอร์ตABH
3) โมดูลเทรซมาโครเซล
4) ส่วนตรวจสอบข้อมูลสำหรับการเทรซ
5) ส่วนจัดการกระบวนการวัดแบบเทรซ
6) ส่วนจัดการพอร์ตสำหรับเชื่อมต่อการเทรซ
7) ส่วนจัดการจุดหยุดทำงานและปรับปรุงข้อมูล
8) ตารางข้อมูลหน่วยความจำรอม

บัสเชื่อมต่อของCortex-M3โปรเซสเซอร์

1) บัสรหัสคำสั่ง
2) บัสข้อมูล
3) บัสระบบ
4) บัสเชื่อมต่ออุปกรณ์เพอริเฟอรัลภายนอก

2.5.6 ระบบไมโครคอนโทรลเลอร์Cortex-M3

2.5.7 ระบบรีเซ็ตในไมโครคอนโทรลเลอร์Cortex-M3

1) การรีเซ็ตจากเพาเวอร์รีเซ็ต
2) การรีเซ็ตระบบ
3) การรีเซ็ตทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.8 คุณสมบัติทางเทคนิคที่สำคัญของ STM32F103VBT6

เป็นไมโครคอนโทรลเลอร์ในอนุกรม STM32F10x มีคุณสมบัติเด่น ดังนี้

1) ซีพียูคอร์ : ARM 32 บิต Cortex-M3 ความถี่สัญญาณนาฬิกาสูงสุด 72 MHz ประสิทธิภาพความเร็วในการทำงาน 1.25 DMIPS/MHz มีตัวหารเลขแบบฮาร์ดแวร์ และหน่วยความประมวลผลทางคณิตศาสตร์
2) หน่วยความจำ : แบบแฟลช 128 กิโลไบต์ และสแตติกแรม 20 กิโลไบต์
3) ระบบสัญญาณนาฬิกา : 4 ถึง 16 MH/z เมื่อใช้คริสตอลออสซิลเลเตอร์,ออสซิลเลเตอร์ 32 kHz สำหรับโมดูลนาฬิกาจริงภายในชิปสามารถปรับเทียบได้
4) ไฟเลี้ยง : 2.0 ถึง 3.6 Vdc
5) ระบบจัดการพลังงาน : มีเพาเวอร์-ออนรีเซ็ต (POR), เพาเวอร์-ดาวน์รีเซ็ต (PDR) มีวงจรตรวจจับไฟเลี้ยงโปรแกรมได้ รองรับการทำงานในโหมดประหยัดพลังงาน 3 โหมด คือ Sleep,Stop,Standby
6) มีแหล่งจ่ายไฟ VBAT สำหรับโมดูลเวลานาฬิกาจริงภายในชิปและรีจิสเตอร์รักษาค่าเวลา
7) โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล : ความเร็ว 1 ไมโครวินาที ความละเอียด 12 บิต จำนวน 2 ชุด รวม 16 ช่อง
8) DMA : มีตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงหรือ DMA controller 7 ช่อง , รองรับการทำงานของไทเมอร์, โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล
9) ขาพอร์ตอินพุตเอาต์พุต : แบบความเร็วสูง 80 ขา รองรับแรงดัน +5V ได้
10) การดีบัก : รองรับการดีบักแบบ Serial wire debug (SWD) และ JTAG
11) ไทเมอร์ : 7 ชุด
12) โมดูลสื่อสาร 9 แบบ

2.5.9 ไดอะแกรมการทำงานของ STM32F103VBT6

หน่วยความจำ

มีหน่วยความจำแฟลชที่สามารถเขียนละลบได้ 128 กิโลไบต์ สำหรับเก็บทั้งข้อมูลและโปรแกรมควบคุม มีหน่วยความจำสแตติกแรม 20 กิโลไบต์ ที่สามารถอ่านเขียนได้อย่างรวดเร็วโดยไม่มีช่วงเวลารอคอย

ตัวควบคุมเวกเตอร์อินเตอร์รัปต์ (NVIC)

ด้วยความสามารถของตัวควบคุมอินเตอร์รัปต์แบบนี้ ทำให้ STM32F103VBT6 สามารถรองรับการอินเตอร์รัปต์แบบกำหนดได้มากถึง 43 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมการอินเตอร์รัปต์จากภายนอกและการเกิดสัญญาณอีเวนต์ (EXTI)

ตัวควบคุมการอินเตอร์รัปต์จากภายนอกและการเกิดสัญญาณอีเวนต์ ประกอบด้วย ตัวตรวจจับขอบสัญญาณ 16 ชุด เพื่อใช้ในการกำเนิดสัญญาณอินเตอร์รัปต์และสัญญาณอีเวนต์แล้วCต่จะร้องขอ แต่ละชุดจะทำงานเป็นอิสระต่อกัน

สัญญาณนาฬิกา

ไมโครคอนโทรลเลอร์ STM32F103VBT6 สามารถรองรับสัญญาณนาฬิกาได้ทั้งจากวงจร RC ภายในและต่อคริสตอลภายนอก

บูทโหมด (Boot mode)

เป็นกระบวนการเริ่มต้นการทำงาน หรือเรียกว่า การบูท (boot) ของไมโครคอนโทรลเลอร์ STM32F103VBT6

ไฟเลี้ยง (Supply voltage)

STM32F103VBT6 ต้องการไฟเลี้ยงหลัก (V_{DD}) และไฟเลี้ยงส่วนอนาล็อก (V_{DDA}) เป็นไฟตรงในช่วง 2 ถึง 3.6V

วงจรจัดการไฟเลี้ยง (Power Supply supervisor)

เนื่องจาก STM32F103VBT6 มีวงจรเพาเวอร์ออนรีเซต (POR) และ เพาเวอร์ดาวรีเซต (PDR) ซึ่งทำการแอกทีปเพื่อทำให้เกิดการรีเซตเมื่อไฟเลี้ยงเกิดการลดต่ำหรือเพิ่มขึ้นเป็น 2V จนกว่าระดับไฟเลี้ยงจะกลับเข้าสู่ระดับปกติ คือ 2.6V ขึ้นไป

วงจรควบคุมไฟเลี้ยง (Voltage regulator)

วงจรควบคุมไฟเลี้ยง STM32F103VBT6 มีโหมดการทำงาน 3 โหมด คือ

1) MR-โหมดการทำงานปกติหรือโหมดรัน
2) LPR-โหมดพลังงานต่ำ ใช้เมื่อกำหนดให้ซีพียูทำงานในโหมดหยุดทำงาน
3) PowerDown-โหมดลดพลังงานใช้เมื่อกำหนดให้ซีพียูทำงานในโหมดรอคอยหรือสแตนด์บาย

การทำงานในโหมดพลังงานต่ำ

STM32F103VBT6 มีโหมดการทำงาน 3 โหมด

1) โหมดสลีฟ (Sleep mode) ในโหมดนี้ซีพียูจะหยุดทำงาน แต่อุปกรณ์เพอริฟอรัสทั้งหมดยังทำงานอยู่
2) โหมดหยุดทำงาน (Stop mode)
3) โหมดรอคอยหรือสแตนด์บาย (Standby mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีลไทม์คล็อกและรีจีสเตอร์สำรองข้อมูล

STM32F103VBT6 มีวงจรฐานเวลานาฬิกาจริงหรือรีลไทม์คล็อกในตัวทำให้สามารถสร้างระบบนาฬิกาที่มีความแม่นยำสูงได้โดยไม่ต้องอาศัยอุปกรณ์ภายนอก รีจีสเตอร์สำรองข้อมูล จะถูกใช้สำรองข้อมูลที่ผู้ใช้งานต้องการให้รักษาไว้

วอตช์ด็อกอิสระ (independent Watchdog : IWDG)

เป็นตัวนับถอยหลังและทำงานร่วมกับปรีสสเกลเลอร์ 8 บิต โดยใช้สัญญาณนาฬิกาความถี่ 40 kHz จากวงจร RC

วินโดร์วอตช์ด็อก (Window Watching : WWDG)

เป็นตัวนับถอยหลังที่สามารถตั้งค่าได้ และเมื่อเกิดการไทม์เอาต์ก็จะทำให้เกิดสัญญาณไทม์เอาต์ก็จะทำให้เกิดสัญญาณรีเซตซีพียูได้ ความแตกต่างของ WWDG กับ IWDG คือ WWDG ใช้สัญญาณนาฬิกาจากระบบสัญญาณนาฬิกาหลัก ดังนั้นหากซีพียูทำงานในโหมดต่ำ WWDG จะหยุดทำงานด้วย ดังนั้นในการทำงานตามปกติ ผู้ใช้งานสามารถเลือกใช้งานวอตช์ด็อกได้2ตัวคือ ทั้งIWDGและWWDG โดยค่าเวลาไทม์เอาต์ของWWDGจะน้อยกว่าIWDGมาก ดังนั้นหากเอ็นเอเบิลการทำงานของWWDGผู้เขียนโปรแกรมต้องไม่ลืมเข้ามาทำการเคลียร์ค่าตัวนับในWWDGให้ทันเวลา มิเช่นนั้นหากWWDGนับค่าเวลาครบรอบ จะทำให้เกิดการรีเซ็ตขึ้น อาจส่งผลให้การทำงานไม่ต่อเนื่อง ตัวนับในWWDGสามารถหยุดการทำงานได้หากชิปต้องทำงานในโหมดดีบัก

ไทเมอร์ (SysTick)

เป็นไทเมอร์หรือตัวตั้งเวลาหลักของระบบ การทำงานเป็นตัวนับถอยหลังหรือนับลงขนาด 24 บิต ที่สามารถรีโหลดได้ สามารถสร้างสัญญาณอินเตอร์รัปต์ได้เมื่อค่าการนับเป็นศูนย์ และสามารถโปรแกรมเลือกแหล่งกำเนิดสัญญาณนาฬิกาได้ ตัวนับในไทเมอร์สามารถหยุดการทำงานได้หากชิปต้องทำงานในโหมดดีบัก

ไทเมอร์ใช้งานทั่วไป (TIMx)

ในไมโครคอนโทรลเลอร์ STM32F103VBT6 มีไทเมอร์ใช้งานทั่วไปขนาด 16 บิตให้เลือกใช้ 3 ตัวแต่ละตัวมีขาพอร์ตอินพุตให้ใช้งานอีก 4 ช่อง ดังนั้นใน STM32F103VBT6 จึงมีขาพอร์ตที่สามารถทำงานกับไทเมอร์ได้มากถึง 12 ขาหรือช่อง สามารถกำหนดให้ทำงานร่วมกันหรือทำงานให้สัมพันธ์หรือซิงโครไนซ์กันได้ โดยไทเมอร์นี้สามารถกำหนดให้นับขึ้นหรือลงก็ได้ สามารถป้อนค่าซ้ำได้ มีปรีสเกลเลอร์ 16 บิต สามารถนำมาใช้ในการนับค่าในการทำงานตรวจจับสัญญาณหรือสร้างสัญญาณเอาต์พุตเปรียบเทียบ รวมถึงนำไปใช้การสร้างสัญญาณ PWM ด้วยนอกจากนั้นไทเมอร์แต่ละตัวยังสามารถทำงานร่วมกับโมดูล DMA ได้อย่างแยกอิสระต่อกัน และตัวนับในไทเมอร์สามารถหยุดทำงานได้หากชิปต้องทำงานในโหมดดีบัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

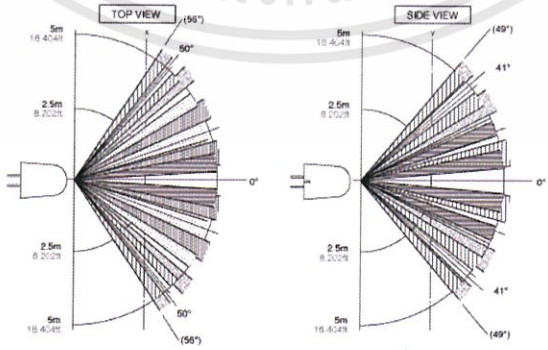
2.6 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)

รูปที่ 2.10 Motion Sensor
ที่มา : hhl.tarad.com (2556)

เซนเซอร์ตรวจจับความเคลื่อนไหว เป็นอุปกรณ์ที่แปลงการตรวจจับความเคลื่อนไหว เป็นสัญญาณไฟฟ้า โดยทั่วไปเซนเซอร์ตรวจจับความเคลื่อนไหวมี 3 ประเภทคือ

1) Passive infrared sensors (PIR) – เป็นเซนเซอร์ที่รับความร้อนจากร่างกายเมื่อเคลื่อนที่ ไม่มีการปล่อยพลังงานออกมาจากเซนเซอร์
2) Ultrasonic- เป็นเซนเซอร์ที่มีการปล่อยคลื่นอัลตร้าโซนิกออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่
3) Microwave - เป็นเซนเซอร์ที่มีการปล่อยคลื่นไมโครเวฟออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่

ในโครงงานนี้ได้เลือกใช้เซนเซอร์ตรวจจับความเคลื่อนไหวประเภท Passive infrared sensors (PIR sensor)เป็นอุปกรณ์ที่ตรวจจับความเคลื่อนไหวด้วยการตรวจวัดความร้อนในพื้นที่ที่ต้องการความร้อนวัดได้จากการเปลี่ยนแปลงระดับรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ เมื่อวัตถุเคลื่อนที่ (สิ่งมีชีวิตทุกชนิด จะแผ่รังสีอินฟราเรดออกมาจากตัวเอง การแผ่รังสีดังกล่าวเกิดจากการเคลื่อนที่ของอิเล็กตรอนในอะตอม ปริมาณรังสีจะมีมากน้อยตามแต่โครงสร้างทางเคมี และอุณหภูมิของวัตถุหรือสิ่งมีชีวิตนั้นๆ) จึงทำให้สามารถตรวจจับสัญญาณลอจิกที่เปลี่ยนแปลงที่ขาเอาต์พุตได้การตรวจจับระยะของ Motion Sensor ตัวนี้อยู่ในระยะตรวจจับ 10 เมตร



รูปที่ 2.11 ระยะตรวจจับเซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)
ที่มา : ปิยะชัย ควรปราโมทย์ (2553)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 TCRT5000

รูปที่ 2.12 TCRT5000

ที่มา : dangerousprototypes.com (2556)

มีทั้ง LED อินฟราเรดและโฟโต้ทรานซิสเตอร์ไว้ภายใน เมื่อจ่ายไฟให้วงจร LED จะส่งแสงอินฟราเรดในรัศมีที่กว้าง ถ้ามีวัตถุอยู่ด้านหน้าโฟโต้ทรานซิสเตอร์ จะได้รับแสงสะท้อนกลับมา ทำให้สามารถนำกระแสได้ จะมากหรือน้อย ก็ขึ้นอยู่กับความเข้มของแสงอินฟราเรดที่ตกกระทบโฟโต้ทรานซิสเตอร์ มีลักษณะดังนี้

-ห่อหุ้มด้วยตะกั่ว

-ตรวจจับแบบโฟโต้เซนเซอร์

-ระยะสูงสุด 2.5 ม.ม.

-มีการกรองการปิดกั้นแสง2.10 ไซเรน 12 โวลต์

2.8 ไซเรน

รูปที่ 2.13 ไซเรน

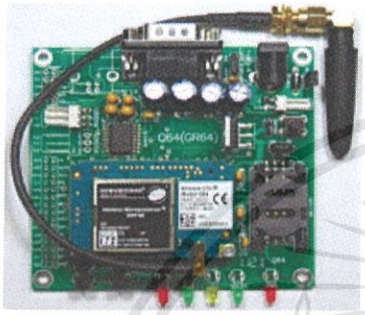
ที่มา : www.taradplaza.com (2556)

1) Tone: One tone, six tone.
2) Rated Voltage: DC 12V.
3) Power out: 15/20/25W.
4) Rated current: 600/1000/1200mA.

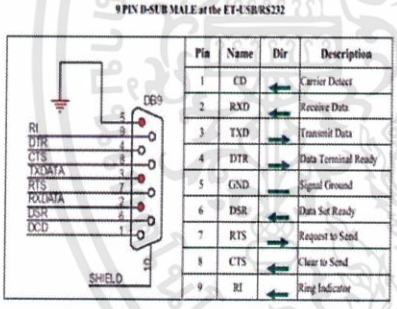
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

3.1 เตรียมอุปกรณ์ Q64 GSM/GPRS MODULE เชื่อมต่อสายเข้ากับตัวอุปกรณ์



3.2 ใช้อุปกรณ์ที่เป็นตัวเชื่อมต่อร่วมกับคอมพิวเตอร์ คือ ET-USB/RS232



Pin	Name	Dir	Description
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	—	Signal Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

การต่อเข้าขา

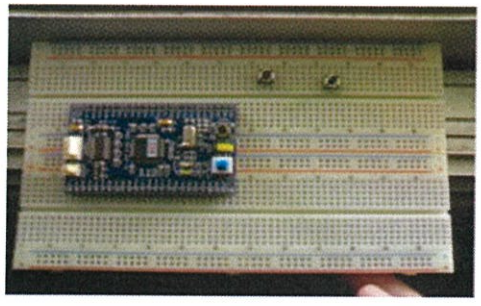
⇦ 2 – รับข้อมูล

⇦ 3 – ส่งข้อมูล

⇦ 5 – สัญญาณกราวด์

3.3 เขียนรหัสคำสั่งควบคุมโมเด็ม AT-COMMAND ในการส่ง SMS

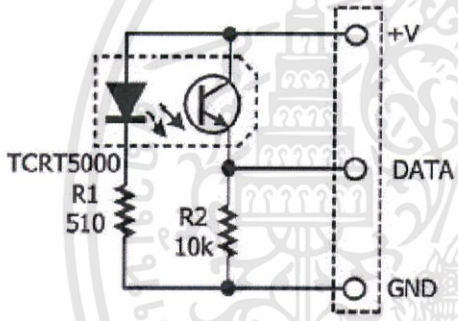
3.4 ต่อเข้ากับอุปกรณ์ ARM Cortex-M3 เป็นตัวที่ใช้ในการควบคุมภายในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ติดตั้ง Motion Sensor เข้ากับระบบ

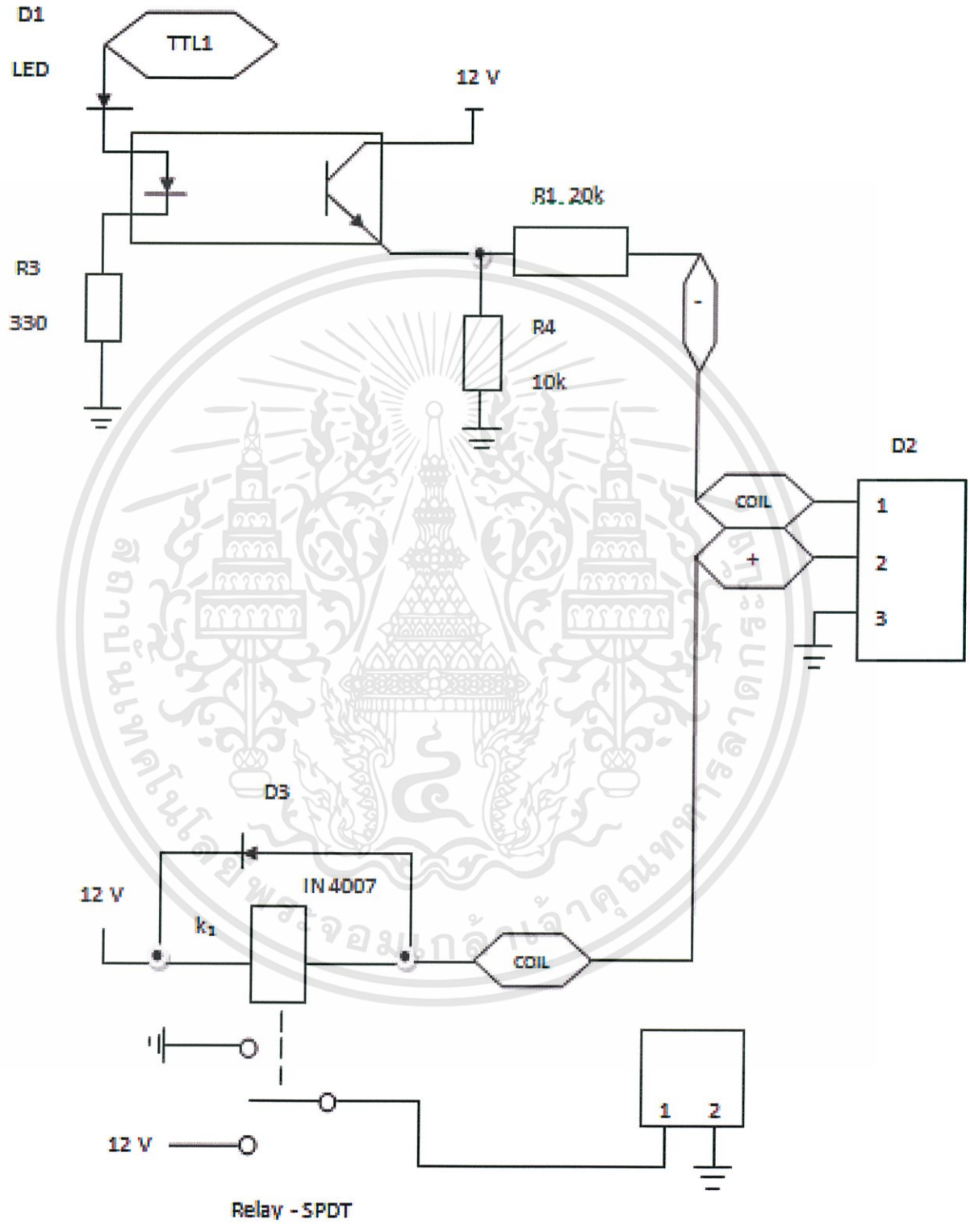
3.6 ติดตั้งเซนเซอร์ LCRT5000 เข้ากับระบบ



3.7 ต่อวงจรขับไซเรน

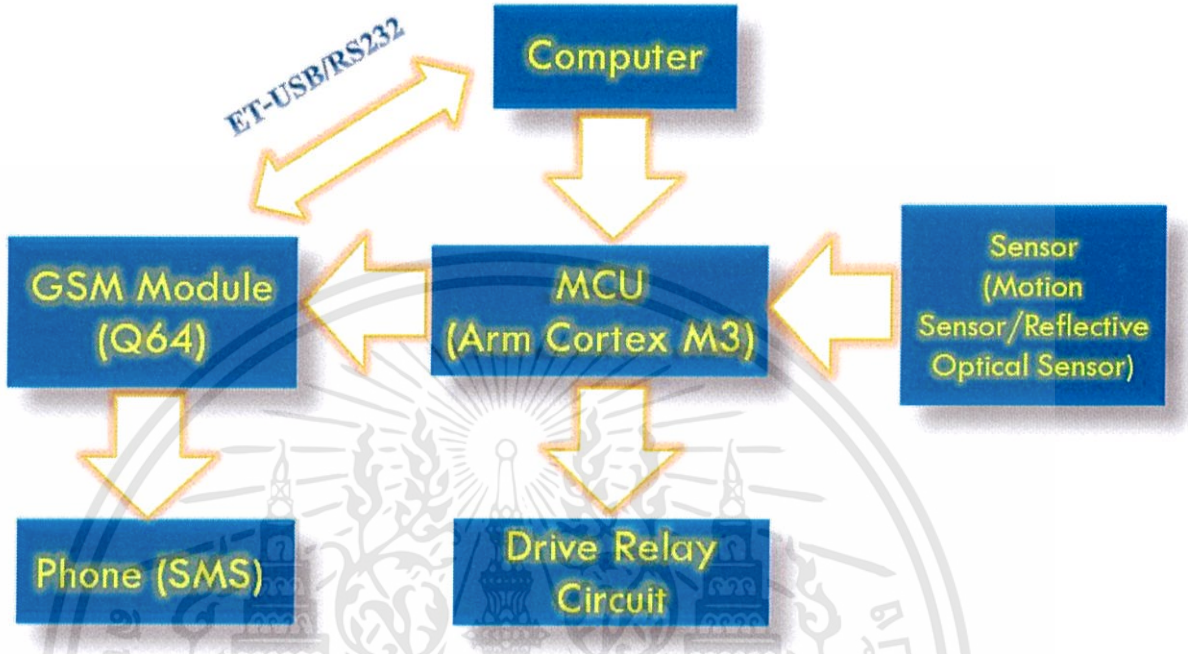
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 วงจรขับไซเรน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 บล็อกไดอะแกรมปริญญานิพนธ์ทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลอง

1) ทำการทดลองโดยการนำ Q64 ต่อสายเชื่อมตัวอุปกรณ์

2) นำตัว ET-USB/RS232 ต่อเข้าคอมพิวเตอร์เพื่อเชื่อมกับ Q64

3) ทำการเชื่อมต่อ Q64 กับ ARM Cortex-M3 เพื่อการควบคุมระบบ

4) ติดตั้งเซนเซอร์ Motion Sensor ใช้ในการตรวจจับภัยโดยเชื่อมต่อกับ ARM Cortex-M3

5) ติดตั้งเซนเซอร์ LCRT5000 เชื่อมต่อไปยัง ARM Cortex-M3

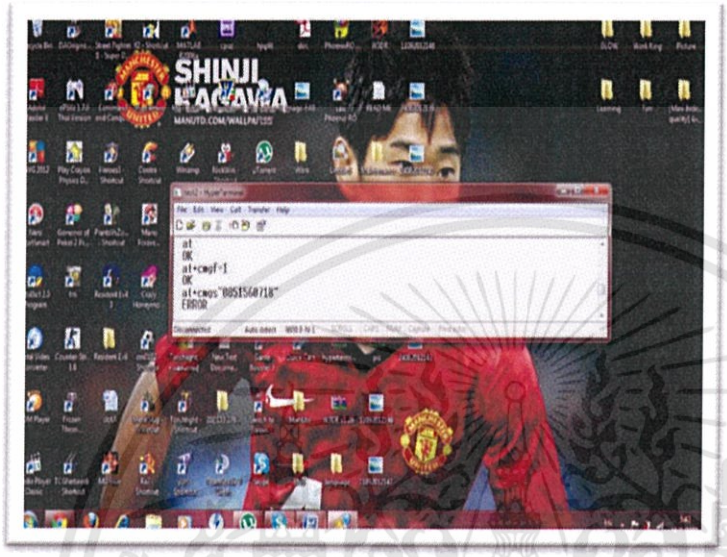
6) เขียนโปรแกรม Keil เบิร์นลง ARM Cortex-M3 เพื่อควบคุมการทำงานทั้งหมด

7) นำ Siren มาต่อร่วมในระบบเพื่อส่งเสียงร้องเตือนภัย

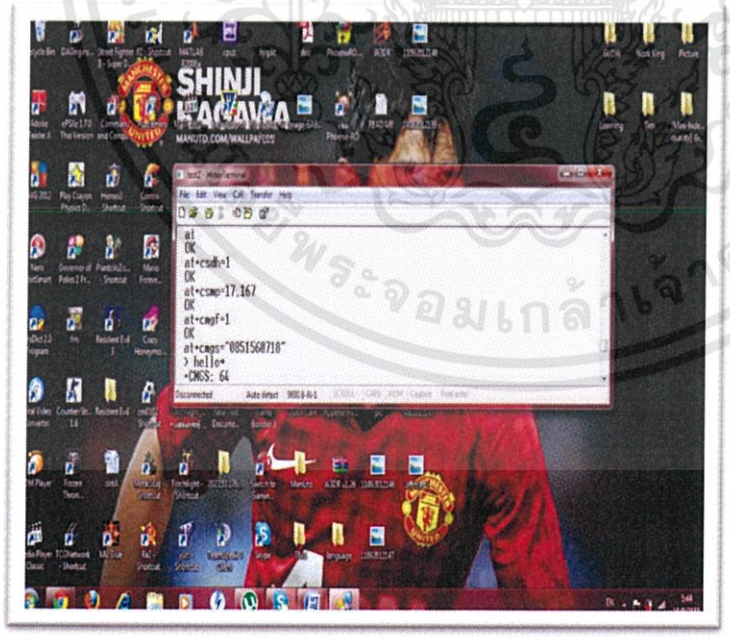
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

จากการทดลองเรารันผ่านโปรแกรมที่ใช้เป็นโปรแกรม Hyperterminal

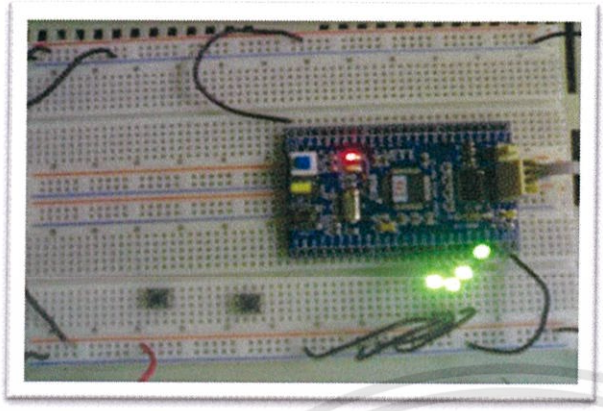


พิมพ์คำสั่ง AT ลงไปแล้วมีการตอบ OK กลับมาแสดงว่าการเชื่อมต่อของอุปกรณ์สมบูรณ์



ตัวอย่างการส่ง พิมพ์คำสั่ง AT ให้ส่ง SMS คำว่า “hello” เข้าโทรศัพท์มือถือ

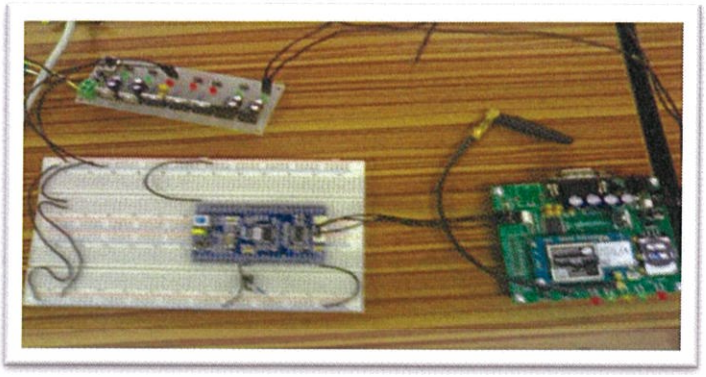
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อส่ง SMS สำเร็จจึงนำไมโครคอนโทรลเลอร์ ARM Cortex-M3 มาเรียนรู้การทำงานและการเขียนโปรแกรมเบื้องต้นก่อนโดยได้ทดลองเขียนโปรแกรมไฟกระพริบก่อน

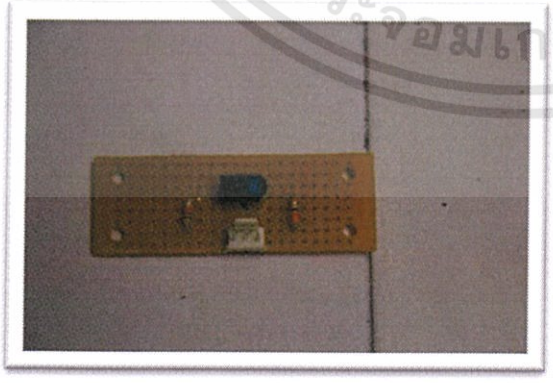
เมื่อเรียนรู้จนพอเข้าใจหลักการแล้วจึงทำการเขียนโปรแกรม Keil โดยใส่คำสั่งการส่งSMSลงไป จากนั้นจึงเบิร์นลง ARM Cortex-M3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ทำการเชื่อมเข้าต่อกับโมดูลโทรศัพท์ Q64 โดยการใช้ฟังก์ชัน USART ภายใน STM32 คือฟังก์ชันการสื่อสารข้อมูลแบบอะซิงโครนัสได้สองทิศทาง ทำให้โมดูลโทรศัพท์ Q64 สามารถสื่อสารกันระหว่างไมโครคอนโทรลเลอร์ได้

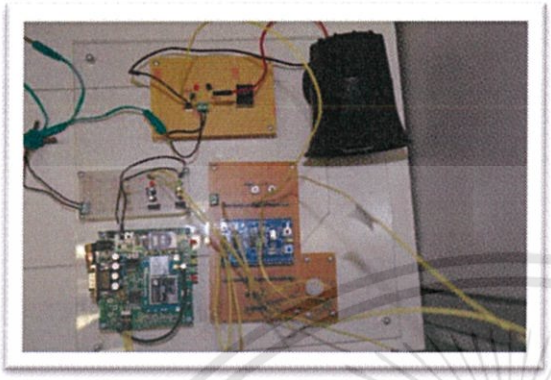
จากนั้นนำอุปกรณ์เซนเซอร์ชนิดตรวจจับความเคลื่อนไหว(Motion sensor)1ตัวมาเชื่อมต่อ



ติดตั้งอุปกรณ์ LCRT5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกทั้งยังติดอุปกรณ์ส่งสัญญาณเตือนภัยแบบเสียง(โดยต่อวงจรขับแรงดันให้ไซเรนทำงานได้ด้วย)



ติดตั้งอุปกรณ์รวมทั้งระบบ

4.3 สรุปผลการทดลอง

ศึกษาการใช้งานระบบและคำสั่งของการใช้โมดูลโทรศัพท์ Q64 จากนั้นนำไมโครคอนโทรลเลอร์ ARM Cortex-M3 เข้ามาเชื่อมต่อกับโมดูลโทรศัพท์ Q64 โดยการใช้ฟังก์ชัน USART ภายใน STM32 คือฟังก์ชันการสื่อสารข้อมูลแบบอะซิงโครนัสได้สองทิศทาง ทำให้โมดูลโทรศัพท์ Q64 สามารถสื่อสารกันระหว่างไมโครคอนโทรลเลอร์ได้ นำอุปกรณ์เซนเซอร์ชนิดตรวจจับความเคลื่อนไหว (Motion sensor) และลิมิตสวิตช์มาเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยใช้ฟังก์ชัน DMA พร้อมกับติดอุปกรณ์ส่งสัญญาณเตือนภัยแบบเสียงร่วมกัน ระบบการทำโดยรวมก็คือเมื่ออุปกรณ์เซนเซอร์นั้นทำงานหรือตรวจจับสิ่งผิดปกติได้ก็จะส่งสัญญาณให้กับไมโครคอนโทรลเลอร์สั่งการให้อุปกรณ์ส่งสัญญาณเตือนภัยแบบเสียงดังขึ้นและสั่งการให้โมดูลโทรศัพท์ส่งข้อความไปยังเบอร์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปการทดลองและข้อเสนอแนะ

5.1 สรุปการทดลอง

ปริญญานิพนธ์นี้ได้ทำการศึกษาเกี่ยวกับระบบเตือนภัยโดยใช้อุปกรณ์ที่สะดวกต่อการใช้งานมาช่วยในการเตือนภัยและยังนำไปประยุกต์กับอุปกรณ์ได้หลายๆอย่าง เนื่องจากสังคมสมัยนี้ภัยมีรอบตัวแม้กระทั่งอยู่ในบ้านพักอาศัยก็ยังจะมีภัยเข้าหาตัวได้ถึงจะมีการระวังภัยก็อาจเกิดได้เหนือความคาดหมาย จึงได้มีการทำสิ่งประดิษฐ์เพื่อมาช่วยในการป้องกันภัยสำหรับที่พักอาศัยต่างๆทำให้เจ้าของมีความสบายใจเมื่อยามที่ออกจากที่พักไปทำกิจกรรมต่างๆภายนอก และยังประหยัดค่าใช้จ่ายมากกว่าการป้องกันภัยในระบบอื่นๆ การทำงานของระบบก็ไม่ยากใช้งานสะดวกเหมาะแก่ผู้ใช้เกือบทุกวัย และหาซื้อมาใช้สะดวกด้วยอุปกรณ์ต่างๆหาง่าย ที่เลือกทำก็เพราะสาเหตุนี้เนื่องด้วยอยากนำไปใช้ในชีวิตประจำวันได้จริง ให้เกิดประโยชน์ต่อผู้อื่น

5.2 ข้อเสนอแนะ

เนื่องจากระบบการเตือนภัยนี้เป็นการเตือนภัยสำหรับมีการบุกรุก ถ้าเราต้องการให้ปลอดภัยจากภัยพิบัติอย่างไฟไหม้ เราก็จะต้องติดตั้งอุปกรณ์เสริมเข้าไป และการเตือนภัยในด้านอื่นๆ สามารถใช้ร่วมกันได้หลายๆอย่าง จะยิ่งช่วยในความปลอดภัยในชีวิตและทรัพย์สินของผู้ใช้มากขึ้น คุ้มค่าต่อการลงทุน อีกทั้งตัว GR64 นั้นสามารถเชื่อมต่ออินเทอร์เน็ตได้ด้วยอาจทำให้ระบบเตือนภัยนี้ใช้ระบบอินเทอร์เน็ตในการใช้งานทำให้มีความสะดวกมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

นคร ภักดีชาติ และชัยวัฒน์ ลิ้มพรจิตรวิไล. 2537. ปฏิบัติการไมโครคอนโทรลเลอร์ ARM Cortex-M3 กับ STM32, กรุงเทพฯ.

บริษัท อีทีที จำกัด: ETT Co.,Ltd. (2008). ARM Cortex-M3. [online] , Available: http://www.etteam.com [2013,January 8].

Sony Ericsson Mobile Communications International. (2003). GR47 GM47 AT Commands Manual pdf. [online], Available: http://ebookbrowse.com/gr47gm47atcommandsmanual-pdf[2013,January 15].

บริษัท อีทีที จำกัด: ETT Co.,Ltd. (2012). ARM Microcontroller. [online], Available: http://www.etteam.com [2013,January 10].

บริษัท อีเลคทรอนิคส์ ซอร์ซ จำกัด.(2009). [online], Available: http://www.es.co.th [2012, December 9].

Panasonic Corperation. (2012). [online], Available: http://pewa.panasonic.com/assets/pcsd/napioncatalog.pdf[2012,December 9].

ศิลา.วิธีสืบค้นข้อมูลสารสนเทศ.[ออนไลน์],เข้าถึงได้จาก: http://www.silaresearch.com/accessory.php (วันที่ค้นข้อมูล: 19 ธันวาคม 2555)

เชาว์ชนะ ศรีปริทัศน์ และ นันทวุฒิ สระแก้ว. 2544. การเขียนโปรแกรมติดต่อสื่อสารระหว่างโทรศัพท์เคลื่อนที่กับคอมพิวเตอร์. [Online]. Available: http://202.28.94.55/web/320491/2547/seminar/g15/file/presentation.ppt

ความรู้อย่างย่อของโครงข่ายโทรศัพท์มือถือ.2550. [Online]. Available: http://www.ubmthai.com/leksoundsmf3/index.php?topic=10853.0;wap2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลิมพล ดอนนาม และ วิชัย สิทธิรุ่ง. เครื่องส่งข้อความเตือนกันขโมยรถยนต์ (ALERT SMS Protect the car). 2548. [Online]. Available: http://www.elecnet.chandra.ac.th/research/paper/2548/2548_150393211.doc

PCTEL. PCTEL AT Command Guide. 2543. [Online]. Available: http://linmodems.technion.ac.il/pctellinux/Pctel.ATCommand.Guide.6.23.pdf

AT-Command. 2553. [Online]. Available: http://pro-project.page.tl/AT_command.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เกี่ยวกับภาคผนวก

โค้ดที่ใช้ในการเขียนลงไมโครคอนโทรลเลอร์

```
#include "stm32f10x_lib.h"
#include "string.h"

#define ADC1_DR_Address    ((u32)0x4001244C)

vu16 ADC_Read[2];

void RCC_Setup(void);
void GPIO_Setup(void);
void USART1_Setup(void);
void DMA_Setup(void);
void ADC_Setup(void);
void NVIC_Setup(void);
void EXTI_Setup(void);

unsigned int k = 0;
unsigned int j = 0;

void delay(unsigned long ms)
{
        volatile unsigned long i,j;
        for(i=0; i<ms; i++)
        for(j=0; j<5525; j++);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int usart1_getc()
{
        while(USART_GetFlagStatus(USART1,USART_FLAG_RXNE)==RESET);       // Wait
untill receive data
   return(USART_ReceiveData(USART1));                    // Return charracter
}

void usart1_putc(unsigned char c)
{
        while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);        // Wait
untill transmittion ready
        USART_SendData(USART1,(int)c);                    // Send Charracter
}

void usart1_puts(unsigned char *s)
{
        while(*s)
                        // Check end of String
        {

                usart1_putc(*s++);
                        // Send charracter  1 time
        }
}

int Transfer(unsigned int k)
{
        if(k==0x30)
        {
                k = 0;
        }
        if(k==0x31)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        k = 1;
}
if(k==0x32)
{
        k = 2;
}
if(k==0x33)
{
        k = 3;
}
if(k==0x34)
{
        k = 4;
}
if(k==0x35)
{
        k = 5;
}
if(k==0x36)
{
        k = 6;
}
if(k==0x37)
{
        k = 7;
}
if(k==0x38)
{
        k = 8;
}
if(k==0x39)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                k = 9;
        }
        else
        {
                k = k;
        }
        return k;
}

int inTransfer(unsigned int k)
{
        if(k==0)
        {
                k = 0x30;
        }
        if(k==1)
        {
                k = 0x31;
        }
        if(k==2)
        {
                k = 0x32;
        }
        if(k==3)
        {
                k = 0x33;
        }
        if(k==4)
        {
                k = 0x34;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(k==5)
        {
                k = 0x35;
        }
        if(k==6)
        {
                k = 0x36;
        }
        if(k==7)
        {
                k = 0x37;
        }
        if(k==8)
        {
                k = 0x38;
        }
        if(k==9)
        {
                k = 0x39;
        }
        else
        {
                k = k;
        }
        return k;
}

int Show(unsigned long i)
{
                int a,b,c,d;
                if(i>=1000)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            a = i/1000;
            b = (i-(a*1000))/100;
            c = (i-((a*1000)+(b*100)))/10;
            d = (i-((a*1000)+(b*100)+(c*10)))/1;
    }
    else if((1000>i)&&(i>=100))
    {
            a = 0;
            b = i/100;
            c = (i-(b*100))/10;
            d = (i-((b*100)+(c*10)))/1;
    }
    else if((100>i)&&(i>=10))
    {
            a = 0;
            b = 0;
            c = i/10;
            d = (i-(c*10))/1;
    }
    else if((10>i)&&(i>=0))
    {
            a = 0;
            b = 0;
            c = 0;
            d = i;
    }

    a = inTransfer(a);
    b = inTransfer(b);
    c = inTransfer(c);
    d = inTransfer(d);
    usart1_putc(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                usart1_putc(b);
                usart1_putc(c);
                usart1_putc(d);

}

void SMS_Send(void)
{
                usart1_putc(0x41);
                delay(200);
                usart1_putc(0x54);
                delay(200);
                usart1_putc(0x0D);
                delay(200);
                delay(2000);

                usart1_puts("AT+CSDH=1");
                delay(200);
                usart1_putc(0x0D);
                delay(200);
                delay(2000);

                usart1_puts("AT+CSMP=17,167");
                delay(200);
                usart1_putc(0x0D);
                delay(200);
                delay(2000);

                usart1_puts("AT+CMGF=1");
                delay(200);
                usart1_putc(0x0D);
                delay(200);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                delay(2000);

                usart1_puts("AT+CMGS=");
                delay(200);
                usart1_putc(0x22);
                delay(200);
                usart1_puts("0865501789");
                delay(200);
                usart1_putc(0x22);
                delay(200);
                usart1_putc(0x0D);
                delay(200);
                delay(2000);

                usart1_puts("Alarm : RAK EKAWIT");
                delay(200);
                usart1_putc(0x1A);
                delay(200);
                usart1_putc(0x0D);
                delay(200);
                delay(2000);
}

void Alarm_On(void)
{
        GPIO_WriteBit(GPIOC,GPIO_Pin_4,1);
        delay(1000);
}

void Alarm_Off(void)
{
        GPIO_WriteBit(GPIOC,GPIO_Pin_4,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(1000);
}

void EXTI0_IRQHandler(void)
{
        if(EXTI_GetITStatus(EXTI_Line0) !=RESET)
        {
                delay(100);
                k = k + 1 ;
                EXTI_ClearITPendingBit(EXTI_Line0);
        }
}

/*void EXTI15_10_IRQHandler(void)
{
        if(EXTI_GetITStatus(EXTI_Line13) !=RESET)
        {
                delay(100);
                j = 1;
                EXTI_ClearITPendingBit(EXTI_Line13);
        }
}         */

int main(void)
{
        unsigned int a = 0;
    RCC_Setup();
    GPIO_Setup();
        USART1_Setup();
        ADC_Setup();
        NVIC_Setup();
        EXTI_Setup();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DMA_Setup();
    while (1)
    {

                if(k == 0)
                {
                        if(ADC_Read[0]>2000)    // Optical Sensor
                        {
                                if(a==0)
                                {
                                        Alarm_On();
                                        SMS_Send();
                                        a = 1;
                                }
                        }
                        else if(ADC_Read[1]>3000)       // Motion Sensor
                        {
                                if(a==0)
                                {
                                        Alarm_On();
                                        SMS_Send();
                                        a = 1;
                                }
                        }
                }
                else if(k != 0)
                {
                        Alarm_Off();
                        //a = 0;
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void RCC_Setup(void)
{
        ErrorStatus HSEStartUpStatus;
    RCC_DeInit();
    RCC_HSEConfig(RCC_HSE_ON);
    HSEStartUpStatus = RCC_WaitForHSEStartUp();
    if (HSEStartUpStatus == SUCCESS)
    {
     FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable);
     FLASH_SetLatency(FLASH_Latency_2);
     RCC_HCLKConfig(RCC_SYSCLK_Div1);
     RCC_PCLK2Config(RCC_HCLK_Div1);
     RCC_PCLK1Config(RCC_HCLK_Div4);
     RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_9);
     RCC_PLLCmd(ENABLE);
     while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET);
     RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);
     while (RCC_GetSYSCLKSource() != 0x08);
    }
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOC |
RCC_APB2Periph_AFIO | RCC_APB2Periph_GPIOB, ENABLE);
}

void GPIO_Setup(void)
{

    GPIO_InitTypeDef GPIO_InitStructure;

        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4 ;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
	GPIO_Init(GPIOC,&GPIO_InitStructure);

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 |GPIO_Pin_2 |GPIO_Pin_3 ;

	GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
	GPIO_Init(GPIOC,&GPIO_InitStructure);

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
	GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
	GPIO_Init(GPIOA,&GPIO_InitStructure);

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
	GPIO_Init(GPIOA,&GPIO_InitStructure);

	GPIO_InitStructure.GPIO_Pin = GPIO_Pin_15 | GPIO_Pin_14;
	GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
	GPIO_Init(GPIOB,&GPIO_InitStructure);

}

void USART1_Setup(void)
{
	USART_InitTypeDef USART_InitStructure;
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,ENABLE);	// Enable
USART1 Clock

	USART_InitStructure.USART_BaudRate = 9600;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        USART_InitStructure.USART_WordLength = USART_WordLength_8b;
        USART_InitStructure.USART_StopBits = USART_StopBits_1;
        USART_InitStructure.USART_Parity = USART_Parity_No;
        USART_InitStructure.USART_HardwareFlowControl                    =
USART_HardwareFlowControl_None;
        USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
        USART_Init(USART1,&USART_InitStructure);                    //
Configure the USART1

        USART_Cmd(USART1,ENABLE);
}


void ADC_Setup(void)
{
        ADC_InitTypeDef ADC_InitStructure;
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1,ENABLE);

        ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
        ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
        ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
        ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
        ADC_InitStructure.ADC_NbrOfChannel = 2;
        ADC_InitStructure.ADC_ScanConvMode = ENABLE;
        ADC_Init(ADC1, &ADC_InitStructure);

        ADC_RegularChannelConfig(ADC1,          ADC_Channel_11,          1,
ADC_SampleTime_13Cycles5);
        ADC_RegularChannelConfig(ADC1,          ADC_Channel_12,          2,
ADC_SampleTime_13Cycles5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ADC_Cmd(ADC1, ENABLE);
	ADC_DMACmd(ADC1, ENABLE);

	ADC_ResetCalibration(ADC1);
	while(ADC_GetResetCalibrationStatus(ADC1));
	ADC_StartCalibration(ADC1);
	while(ADC_GetCalibrationStatus(ADC1));
	ADC_SoftwareStartConvCmd(ADC1,ENABLE);

}

void DMA_Setup(void)
{
	DMA_InitTypeDef DMA_InitStructure;
	RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1,ENABLE);

   DMA_DeInit(DMA1_Channel1);
   DMA_InitStructure.DMA_PeripheralBaseAddr = (u32)&ADC1->DR;
   DMA_InitStructure.DMA_MemoryBaseAddr = (u32)&ADC_Read;
   DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
   DMA_InitStructure.DMA_BufferSize = 3;
   DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
   DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
   DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
   DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
   DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
   DMA_InitStructure.DMA_Priority = DMA_Priority_VeryHigh;
   DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
   DMA_Init(DMA1_Channel1, &DMA_InitStructure);
	DMA_Cmd(DMA1_Channel1,ENABLE);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void NVIC_Setup(void)
{
        NVIC_InitTypeDef NVIC_InitStructure;

        NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);

        NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQChannel;
        NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
        NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
        NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
        NVIC_Init(&NVIC_InitStructure);

/*      NVIC_InitStructure.NVIC_IRQChannel = EXTI15_10_IRQChannel;
        NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
        NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
        NVIC_Init(&NVIC_InitStructure);                              */

}

void EXTI_Setup(void)
{
        EXTI_InitTypeDef EXTI_InitStructure;
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO,ENABLE);
        EXTI_InitStructure.EXTI_Line = EXTI_Line0;
        EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
        EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
        EXTI_InitStructure.EXTI_LineCmd = ENABLE;
        EXTI_Init(&EXTI_InitStructure);

/*      GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EXTI_InitStructure.EXTI_Line = EXTI_Line13;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);          */
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้