

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ
SECURITY SYSTEM AND ALERTS VIA MOBILE PHONE



T117548



นาย สิทธิชัย ลิขิตนุกูล
SITTICHAI LIKITNUKUL
นาย สุภกิจ พัฒนาดิษฐ์
SUPAKIT PATTANADISAI
นาย อนุชิต เจนป่า
ANUCHIT JENPA

เลขหมู่.....
เลขทะเบียน **117548**
วัน,เดือน,ปี **- 5 ต.ค. 2554**

b. 12343055
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SECURITY SYSTEM AND ALERTS VIA MOBILE PHONE



**THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

ระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ

รายนามนักศึกษา

นายสิทธิชัย ลิขิตนุกูล

รหัสนักศึกษา 50011672

นายสุภกิจ พัฒนาศิษฐ์

รหัสนักศึกษา 50011741

นายอนุชิต เจนป่า

รหัสนักศึกษา 50011839

ปริญญา

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมสารสนเทศ

พ.ศ.

2553

อาจารย์ที่ปรึกษาปริญญานิพนธ์

ผศ.ไพศาล สิทธิโยภาสกุล

รศ.นิกร สุขุมตันติ

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

(ผศ.ไพศาล สิทธิโยภาสกุล)

(รศ.นิกร สุขุมตันติ)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ

โดย นายสิทธิชัย ดิขิตนุกูล รหัส 50011672

นายสุภกิจ พัฒนาศิษย์ รหัส 50011741

นายอนุชิต เจนป่า รหัส 50011839

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศ

พ.ศ. 2553

อาจารย์ที่ปรึกษา ผศ. ไพศาล สิทธิโยภาสกุล

รศ. นิกร สุขุมตันติ

ปีการศึกษา 2553

บทคัดย่อ

ระบบรักษาความปลอดภัยและการแจ้งเตือนผ่าน โทรศัพท์มือถือ เป็นระบบที่ช่วยรักษาความปลอดภัยภายในบ้าน โดยหากมีเหตุฉุกเฉินเกิดขึ้น เช่น เกิดอัคคีภัย หรือ มีผู้บุกรุกเข้ามาภายในบ้าน ระบบก็จะทำการตรวจจับด้วยอุปกรณ์ตรวจจับควัน อุปกรณ์ตรวจจับความร้อน และอุปกรณ์แถบแม่เหล็ก แล้วส่งสัญญาณเข้ากับระบบควบคุมด้วยไมโครคอนโทรลเลอร์เพื่อทำการส่งข้อความแจ้งเตือนไปยังหมายเลขโทรศัพท์มือถือของเจ้าของบ้านให้รับทราบ และตั้งการให้ป้อนน้ำทำงานเพื่อช่วยแก้ไขเหตุการณ์ และยังมีส่วนควบคุมการปิด/เปิดของระบบ เพื่อให้ผู้ใช้งานได้รับความสะดวกในการใช้งาน ได้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Security system and Alerts via mobile phone		
Student	MR.SITTICHAI	LIKITNUKUL	NO. 50011672
	MR.SUPAKIT	PATTANADISAI	NO. 50011741
	MR.ANUCHIT	JENPA	NO. 50011839
Degree	Bachelor of Engineering		
Program	Information Engineering		
Adviser	MR. PAISAN	SITTIYOPASAKUL	
	MR.NIKORN	SUKUTAMATUNTI	
Year	2010		

ABSTRACT

Security system and Alerts via mobile phone is used for security purpose. It can detect emergency situation such as fire or stealth. In case of emergency, this system will detect fire by using smoke detector, heat detector and magnetic sensor to detect intruder in the house. After that, microcontroller will process these data and send the emergency message to the house keeper's mobile and water pump is working for help wrong situation. It has open/close control system for comfortable to use it.

กิตติกรรมประกาศ

ในการทำโครงการปริญาณิพนธ์นี้ ทางผู้จัดทำได้รับความอนุเคราะห์ช่วยเหลือ ในด้านคำแนะนำแนวทางในการดำเนินงาน และความช่วยเหลือในเรื่องสถานที่ในการทำโครงการ เครื่องมืออุปกรณ์ต่างๆ จากท่านผศ.ไพศาล สิทธิโยภาสกุล และรศ.นิกร สุขุมตันติ มาโดยตลอด จนกระทั่งสำเร็จเป็นโครงการนี้ขึ้นมา ผู้จัดทำจึงขอกราบขอบพระคุณท่านทั้งสองที่ให้ความกรุณาไว้ ณ ที่นี้ด้วย

ขอขอบเพื่อนๆนักศึกษาทุกคนที่ให้ความช่วยเหลือ และให้คำปรึกษาในการแก้ไขปัญหาอุปสรรคต่างให้สามารถผ่านไปได้อย่างดี และทำนี้ขอกราบขอบพระคุณ บิคา มารดา และทางสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งให้การสนับสนุนทางการเงิน

คณะผู้จัดทำ

นายสิทธิชัย ลิขิตนุกูล
นายสุภกิจ พัฒนาดิษฐ์
นายอนุชิต เจนป่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 อุปกรณ์ที่ต้องใช้	2
1.5 ขั้นตอนการดำเนินการ	2
1.6 ผลที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้	4
2.1 บอร์ด MCS-51	4
2.1.1 คุณสมบัติของบอร์ด และ MCU	4
2.1.2 โครงสร้างและการจัดสรร I/O ของบอร์ด	5
2.2 GSM/GPRS Module	10
2.2.1 คุณสมบัติต่างของ GR64 GSM/GPRS	11
2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232	12
2.3.1 ขั้วต่อสำหรับพอร์ต RS-232 และการเชื่อมต่อ	12
2.4 อุปกรณ์ตรวจจับสถานะการฉีกปกติ	15
2.4.1 อุปกรณ์ตรวจจับวัน	15
2.4.2 อุปกรณ์แถบแม่เหล็ก	18
2.4.3 เซนเซอร์วัดอุณหภูมิ DS18S20	19
2.5 อุปกรณ์ รับ-ส่ง สัญญาณรีโมท	25
2.6 รีเลย์ (Relay)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.7 อุปกรณ์ Digital Code Switch	26
2.8 ข้อความสั้น (SMS)	27
2.8.1 หลักการรับส่ง ข้อความสั้น (SMS) ของโทรศัพท์มือถือ	27
2.8.2 AT Command	28
2.9 ทราานซิสเตอร์ชนิด NPN	28
2.9.1 การทำงานของทรานซิสเตอร์	29
บทที่ 3 ออกแบบโครงงาน	30
3.1 การออกแบบการทำงานของระบบทั้งหมด	30
3.2 การทำงานของบอร์ดไมโครคอนโทรลเลอร์	32
3.3 การทำงานของ GSM/GPRS Module	33
3.4 การออกแบบวงจรอุปกรณ์แถบแม่เหล็ก	34
3.5 การออกแบบวงจรอุปกรณ์ตรวจจับควัน	36
3.6 การออกแบบวงจรอุปกรณ์ตรวจจับความร้อน	38
3.7 การออกแบบวงจรรีเลย์	40
3.8 การออกแบบวงจรอุปกรณ์รับ-ส่งสัญญาณรีโมท	41
3.9 การออกแบบวงจรอุปกรณ์ Digital Code Switch	43
3.10 การออกแบบวงจรทรานซิสเตอร์	45
บทที่ 4 ผลการทดลอง	47
4.1 การทดลองของอุปกรณ์แถบแม่เหล็ก	47
4.2 การทดลองของอุปกรณ์ตรวจจับควัน	50
4.3 การทดลองของอุปกรณ์ตรวจจับความร้อน	53
4.4 การทดลองของอุปกรณ์รับ-ส่งสัญญาณรีโมท	56
4.5 การทดลองของอุปกรณ์ Digital Code Switch	58
บทที่ 5 สรุปผลดำเนินโครงการ	60
5.1 บทวิจารณ์สรุปผล	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	60
5.3 แนวทางการแก้ไขปัญหาและพัฒนาต่อ	61
บรรณานุกรม	62



สารบัญรูป

รูปที่	หน้า
2.1 แสดงโครงสร้าง MCU AT89C51RE2	5
2.2 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P0	6
2.3 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P1	6
2.4 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P2	6
2.5 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P3	7
2.6 แสดงการ Set Jumper UART#2 เมื่อใช้งานขั้วต่อ RS232#2	8
2.7 แสดงการจัดเรียงขาขั้วต่อ RS232#1 และ RS232#2	8
2.8 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS422	8
2.9 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS485	9
2.10 แสดงการจัดเรียงขาของขั้วต่อ RS422/485	9
2.11 แสดงการจัดเรียงขาของขั้วต่อ CLCD แบบ 4 บิต	9
2.12 แสดงการจัดเรียงขาของขั้วต่อ #DS1307 4 PIN	10
2.13 ขั้วต่ออนุกรม 9 ขา หรือแบบ DB-9	13
2.14 ขั้วอนุกรม 25 ขา หรือแบบ DB-25	13
2.15 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมแบบ Null Modem	15
2.16 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม	15
2.17 การทำงานของระบบ Smoke Detector Ionization	16
2.18 เมื่อมีอนุภาควัณมาติดที่แผ่น Screen	16
2.19 ระบบการทำงานของ อุปกรณ์ตรวจจับควัน แบบกีดขวางแสง	17
2.20 ระบบการทำงานของ อุปกรณ์ตรวจจับควัน แบบหักเหแสง	17
2.21 Chart ขนาดอนุภาคควันต่างๆ หน่วยแกน x เป็น ไมครอน	18
2.22 แถบแม่เหล็ก	18
2.23 โครงสร้าง และขาของ DS18B20 ตัวถังแบบ TO-92	19
2.24 โครงสร้างรีจิสเตอร์ภายใน DS18S20	20
2.25 โครงสร้างภายในรีจิสเตอร์ Temperature	21
2.26 การต่อใช้งาน DS18S20	21
2.27 การเริ่มการติดต่อสื่อสารแบบ 1-wire ด้วย Reset Pulse และ Presence Pulse	22
2.28 การเขียนข้อมูลจาก Master ลง DS18S20	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.29 การอ่านข้อมูลจาก DS18S20	23
2.30 ขั้นตอนการแปลง และอ่านอุณหภูมิจาก DS18S20	24
2.31 แสดงเซ็นเซอร์ตรวจจับความร้อน	24
2.32 แสดงอุปกรณ์ตัวส่ง	25
2.33 แสดงลักษณะรีเลย์	26
2.34 แสดงรูปของอุปกรณ์ Digital Code Switch	27
2.35 โครงสร้างทรานซิสเตอร์ NPN	28
2.36 ทรานซิสเตอร์ NPN	29
2.37 แสดงทรานซิสเตอร์ชนิด NPN	29
3.1 แสดงภาพการทำงานของระบบทั้งหมด	30
3.2 แสดงแผนผังการทำงานของส่วนส่งข้อความ	31
3.3 แสดงลักษณะ โครงสร้างของบอร์ดไมโครคอนโทรลเลอร์	32
3.4 โครงสร้างของบอร์ด GSM/GPRS Module	33
3.5 โครงสร้างของบอร์ด GSM/GPRS Module ต่อกับ ไมโครคอนโทรลเลอร์	33
3.6 แผนผังแสดงการทำงานของแถบแม่เหล็ก	34
3.7 แสดงวงจรอุปกรณ์แถบแม่เหล็กต่อกับไมโครคอนโทรลเลอร์	35
3.8 แผนผังแสดงการทำงานของอุปกรณ์ตรวจจับควัน	36
3.9 แสดงวงจรอุปกรณ์ตรวจจับควัน	37
3.10 แสดงวงจรอุปกรณ์ตรวจจับควันต่อเข้ากับไมโครคอนโทรลเลอร์	37
3.11 แผนผังแสดงการทำงานของอุปกรณ์ตรวจจับความร้อน	38
3.12 แสดงวงจรอุปกรณ์ตรวจจับความร้อน	39
3.13 แสดงวงจรอุปกรณ์ตรวจจับความร้อนต่อกับไมโครคอนโทรลเลอร์	39
3.14 แสดงวงจรรีเลย์	40
3.15 แสดงวงจรรีเลย์ต่อกับไมโครคอนโทรลเลอร์	40
3.16 แสดงการทำงานของอุปกรณ์รับ-ส่งสัญญาณรีโมท	41
3.17 แสดงวงจรอุปกรณ์ส่งสัญญาณรีโมท	42
3.18 แสดงวงจรอุปกรณ์รับสัญญาณรีโมท	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.19 แสดงวงจรอุปกรณ์รับสัญญาณรีโมทต่อกับไมโครคอนโทรลเลอร์	43
3.20 แสดงแผนผังการทำงานของ Digital Code Switch	43
3.21 แสดงวงจรของ Digital Code Switch	44
3.22 แสดงวงจรของ Digital Code Switch ต่อกับไมโครคอนโทรลเลอร์	44
3.23 แสดงวงจรทรานซิสเตอร์	45
3.24 วงจรรวมของระบบ	46
4.1 แสดงอุปกรณ์แถบแม่เหล็กในขณะที่มีแรงดึงดูด	47
4.2 แสดงการไหลของกระแสของอุปกรณ์แถบแม่เหล็กในขณะที่ไม่มีแรงดึงดูด	48
4.3 แสดงอุปกรณ์แถบแม่เหล็กในขณะที่ไม่มีแรงดึงดูด	48
4.4 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์แถบแม่เหล็ก	49
4.5 แสดงอุปกรณ์ตรวจจับควันในขณะที่ยังไม่ทำงาน	50
4.6 แสดงการไหลของกระแสของอุปกรณ์ตรวจจับควันในขณะที่ทำงาน	51
4.7 แสดงอุปกรณ์ตรวจจับควันในขณะที่ทำงาน	51
4.8 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์ตรวจจับควัน	52
4.9 แสดงการทำงานของปั้มน้ำ	52
4.10 แสดงอุปกรณ์ตรวจจับความร้อนในขณะที่ยังไม่ทำงาน	53
4.11 แสดงการไหลของกระแสของอุปกรณ์ตรวจจับความร้อนในขณะที่ทำงาน	54
4.12 แสดงอุปกรณ์ตรวจจับความร้อนในขณะที่ทำงาน	54
4.13 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์ตรวจจับความร้อน	55
4.14 แสดงการทำงานของปั้มน้ำ	55
4.15 แสดงอุปกรณ์รับสัญญาณรีโมทในขณะที่ยังไม่ทำงาน	56
4.16 แสดงการไหลของกระแสของอุปกรณ์รับสัญญาณรีโมทในขณะที่ทำงาน	56
4.17 แสดงอุปกรณ์รับสัญญาณรีโมทในขณะที่ทำงาน	57
4.18 แสดงอุปกรณ์ Digital Code Switch ในขณะที่ยังไม่ทำงาน	58
4.19 แสดงอุปกรณ์ตรวจจับความร้อนในขณะที่ยังไม่ทำงาน	58
4.20 แสดงอุปกรณ์ Digital Code Switch ในขณะทำงาน	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

หน้า

2.1 จัดทำของสัญญาณพร้อมต่ออนุกรมในรูปแบบต่าง ๆ และหน้าที่การทำงาน

13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 แนวคิดและที่มาของปัญหา

สภาพสังคมของผู้คนในปัจจุบันมักจะใช้ชีวิตการทำงานอยู่นอกที่พักอาศัยเป็นจำนวนมาก และไม่ค่อยมีเวลาว่างที่จะดูแลที่พักอาศัย เมื่อเกิดเหตุร้ายขึ้นภายในที่พักอาศัยขณะที่เจ้าของไม่อยู่ เช่น การโจรกรรมทรัพย์สิน หรือการเกิดอัคคีภัย เจ้าของที่พักอาศัยก็จะไม่สามารถที่จะป้องกันหรือแก้ไขเหตุการณ์เหล่านี้ได้ ซึ่งเป็นสาเหตุที่ทำให้สูญเสียทรัพย์สินและอาจส่งผลกระทบต่อการดำเนินชีวิต เรื่องของความปลอดภัยในที่พักอาศัยจึงเป็นสิ่งที่เราไม่ควรมองข้ามและมีความจำเป็นอย่างยิ่งที่จะหาทางป้องกันแก้ไข ดังนั้นระบบรักษาความปลอดภัยที่ดีจึงเป็นสิ่งที่น่าสนใจอย่างยิ่ง ระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ น่าจะเป็นอีกทางเลือกหนึ่งที่จะช่วยป้องกันไม่ให้เกิดปัญหาหรือช่วยลดความเสียหายที่อาจจะเกิดขึ้น โดยระบบนี้จะทำการตรวจสอบประตูหรือหน้าต่างว่าถูกเปิดหรือไม่ โดยอุปกรณ์แถบแม่เหล็ก ในกรณีที่เกิดอัคคีภัยระบบจะตรวจสอบผ่านทางตัวตรวจจับของระบบ คือ อุปกรณ์ตรวจจับควัน และอุปกรณ์ตรวจจับความร้อน จากนั้นเมื่ออุปกรณ์สามารถตรวจจับความผิดปกติได้ จะทำการส่งข้อความแจ้งเตือนไปยังเจ้าของที่พักอาศัย เพื่อให้ได้รับทราบเหตุการณ์ที่เกิดขึ้น และมีส่วนของปั้มน้ำเพื่อให้สามารถแก้ไขเหตุการณ์ผิดปกตินั้นได้

1.2 วัตถุประสงค์

วัตถุประสงค์ของระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ มีดังนี้

1.2.1 เพื่อสร้างอุปกรณ์ที่ใช้สำหรับแจ้งเตือนเพื่ออำนวยความสะดวก

1.2.2 เพื่อนำเทคโนโลยีมาใช้ประโยชน์และประยุกต์ใช้ในชีวิตประจำวัน

1.2.3 เพื่อศึกษาการเขียน โปรแกรมที่ทำให้อุปกรณ์ทำงานได้อย่างมีประสิทธิภาพ

1.3 ขอบเขตของโครงการ

1.3.1 สามารถส่งข้อความสั้น (SMS) แจ้งเตือนเหตุการณ์ความผิดปกติที่เกิดขึ้นผ่านทางโทรศัพท์มือถือ โดยใช้คุณสมบัติของคำสั่ง AT Commands

1.3.2สามารถที่จะเพิ่มเติมอุปกรณ์ที่จะช่วยในการรักษาความปลอดภัยได้และสามารถแก้ไขเหตุการณ์ที่ผิดปกติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
1

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.3.3 สามารถใช้ตรวจสอบประตูหรือหน้าต่างว่าถูกเปิดอยู่หรือไม่
- 1.3.3 ไมโครคอนโทรลเลอร์สามารถสั่งการให้ปั้มน้ำทำงานได้
- 1.3.4 สามารถควบคุมการเปิด/ปิดของระบบได้

1.4 อุปกรณ์ที่ต้องใช้

1.4.1 ฮาร์ดแวร์

- ชุดบอร์ด CP-JR51RE2 V1.0
- ชุดบอร์ด ET-GSM SIM300CZ V1.0
- เครื่องคอมพิวเตอร์สำหรับพัฒนาโปรแกรม เชื่อมต่ออุปกรณ์และเน็ตเวิร์ค
- ปั้มน้ำ
- วงจรีเลย์
- วงจรทรานซิสเตอร์
- ชุดอุปกรณ์รับ-ส่งสัญญาณรีโมท
- ชุดอุปกรณ์ Digital Code Switch
- อุปกรณ์ตรวจแถบแม่เหล็ก, อุปกรณ์ตรวจจับควัน, อุปกรณ์ตรวจจับความร้อน

1.4.2 ซอฟต์แวร์

- Keil uVision3 , Filp 3.2.0

1.5 ขั้นตอนการดำเนินการ

1.5.1 ศึกษาค้นคว้าและจัดเตรียมอุปกรณ์ โดยหาข้อมูลเกี่ยวกับอุปกรณ์ต่างๆ เพื่อที่จะนำมาใช้สร้างในส่วนของฮาร์ดแวร์ และศึกษาการทำงานของวงจรอุปกรณ์

1.5.2 ศึกษาการใช้ภาษาซีและคำสั่ง AT Commands เพื่อใช้ควบคุมในการส่งข้อความสั้น (SMS) เข้าโทรศัพท์มือถือ

1.5.3 ออกแบบการทำงานในส่วนต่างๆของระบบ

1.5.4 ทดสอบการทำงานของอุปกรณ์ที่ได้ทำการออกแบบ

1.5.5สรุปการทำงานในส่วนต่างๆที่ได้ทำการออกแบบ

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 อุปกรณ์ที่สร้างขึ้นสามารถช่วยป้องกันหรือลดเหตุการณ์การถูกโจรกรรมและ
อัคคีภัยได้

1.6.2 สามารถนำไปประยุกต์ใช้กับหมู่บ้าน คอนโดมิเนียม หรือหอพักได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้

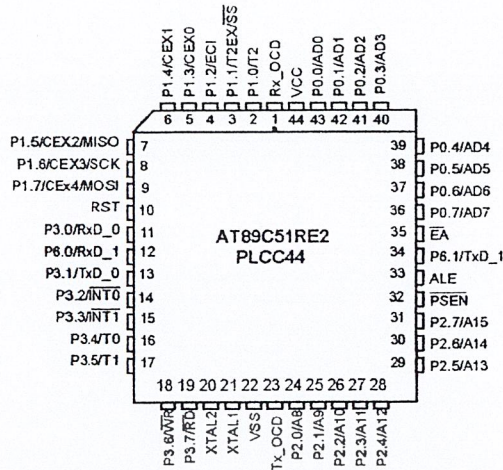
ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการทำโครงการ

2.1 MCS-51

บอร์ดไมโครคอนโทรลเลอร์รุ่นนี้จะใช้ MCU ขนาด 8 บิต ของ Atmel เบอร์ # AT89C51RE2 จะบรรจุในตัวถังแบบ PLCC ขนาด 44 ขา MCU เบอร์นี้คือ มี UART ให้ใช้งาน 2 แชนแนล, มี Timer/Counter ขนาด 16 บิต, มี Flash 128 Kbyte และมีขนาด RAM 8 มี Port I/O = 4 Port , Port RS232 = 2 Port , Port RS422/485 = 1 Port , Port LCD แบบ 4 บิต 1 Port และวงจรสำหรับในส่วนของ RTC ที่ใช้กับ # DS1307 การ Download โปรแกรมลงบอร์ดนั้นจะ Download ผ่านทาง Port RS232 โดยใช้โปรแกรม FlipV3.1.0 เป็นตัว Download และใช้คอมไพเลอร์ Keil μ Vision3 เป็นตัวพัฒนาโปรแกรมด้วยภาษา C

2.1.1 คุณสมบัติของบอร์ด และ MCU

- MCU เป็นตัวถังแบบ PLCC 44 Pin
- MCU ทำงานที่แรงดัน 2.7 - 5.5 V
- ความถี่ Crystal ที่ใช้งานบนบอร์ด 18.432 MHz
- หน่วยความจำ: Flash 128 KB, RAM 8KB
- การสื่อสารอนุกรมประกอบด้วย SPI 1 แชนแนล และ Uart 2 แชนแนล
- 16 บิต Timer/Counter สำหรับ Timer_0, Timer_1 และ Timer_2
- Watch-Dog Timer 14 บิต Counter
- PORT I/O 34 PIN (P0-P3 P6.0, P6.1)
- 11 Interrupt Source ซึ่งกำหนดระดับความสำคัญของ Interrupt ได้ 4 ระดับ
- Download โปรแกรมด้วย Flip V3.1.0 ผ่านทาง RS232
- MCU ทำงานที่อุณหภูมิ - 40 ถึง +85 องศาเซลเซียส



รูปที่ 2.1 แสดงโครงสร้าง MCU AT89C51RE2

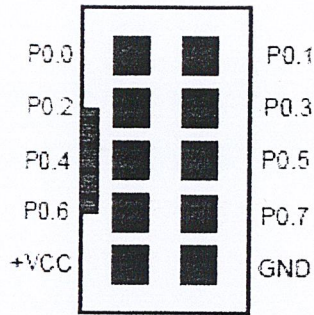
2.1.2 โครงสร้าง และการจัดสรร I/O ของบอร์ด

แหล่งจ่ายไฟ : สำหรับแหล่งจ่ายไฟของบอร์ดนี้สามารถต่อใช้งานได้ทั้งไฟกระแสตรงและสลับ โดยป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดันประมาณ 9-12 V ให้กับบอร์ด ซึ่งสามารถเลือกต่อกับขั้ว Connector แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านขั้ว Connector สำหรับ Adapter จ่ายไฟก็ได้เช่นกัน โดยจะแสดงผลการทำงานของแหล่งจ่ายไฟให้ทราบด้วย LED “PWR”

สัญญาณนาฬิกา : (CLOCK) ความถี่ของสัญญาณนาฬิกาที่จะป้อนให้กับ MCU #AT89C51RE2 นั้น ตามปกติแล้วสามารถป้อนค่าความถี่ของ Crystal ได้ถึง 40MHz ใน Standard Mode (12 Clock / 1 Machine Cycle) แต่ในกรณีที่ให้ MCU ทำงานใน X2 Mode จะสามารถใช้ค่าความถี่สูงสุดได้ที่ 20 MHz โดยสำหรับบอร์ด CP-JR51RE2 นั้นจะกำหนดให้ใช้ค่าความถี่ของ Crystal ที่ป้อนให้กับ MCU ด้วยค่าความถี่ 18.432 MHz เพื่อให้การสื่อสาร Port อนุกรมสามารถหาร Baud Rate ได้ลงตัว

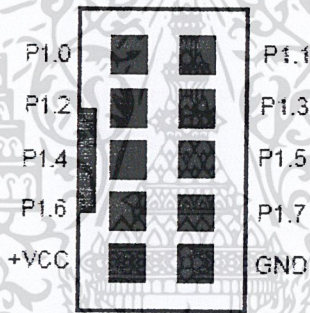
ขั้วต่อ I/O Port: สำหรับบอร์ดนี้จะต่อขาสัญญาณ I/O Port ของ MCU เข้ากับขั้วต่อ Connector 10 Pin เพื่อให้ผู้ใช้งานไปต่อใช้งานได้สะดวกขึ้น โดยมีการจัดขาสัญญาณดังนี้

Port-P0[0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 10 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก P0 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



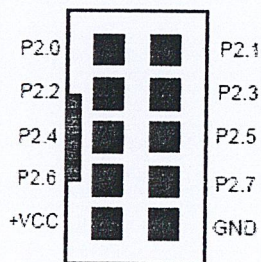
รูปที่ 2.2 แสดงการจัดเรียงขาเชื่อมต่อ 10 Pin ของ Port P0

Port-P1[0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขาต่อ Connector ขนาด 10 Pin โดยขาต่อนี้จะเชื่อมต่อสัญญาณมาจาก P1 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



รูปที่ 2.3 แสดงการจัดเรียงขาเชื่อมต่อ 10 Pin ของ Port P1

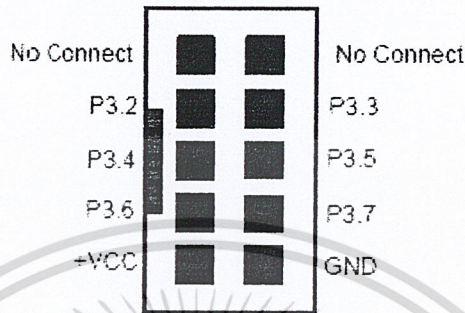
Port-P2 [0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขาต่อ Connector ขนาด 10 Pin โดยขาต่อนี้จะเชื่อมต่อสัญญาณมาจาก P2 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



รูปที่ 2.4 แสดงการจัดเรียงขาเชื่อมต่อ 10 Pin ของ Port P2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port-P3[2..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 10 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก P3 ของ MCU อยู่ 6 เส้น ส่วนอีก 2 เส้นที่เหลือคือ P3.0 และ P3.1 นั้นจะถูกต่อไปยัง ขั้วต่อ RS232 #1 ลักษณะการจัดขาสัญญาณแสดงดังรูป



รูปที่ 2.5 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P3

ขั้วต่อ RS232/RS422/RS485: สำหรับบอร์ดนี้ ได้จัดสรรขั้วต่อ สำหรับสื่อสารแบบอนุกรมไว้ 3 รูปแบบโดยรายละเอียดและหน้าที่การใช้งานของแต่ละแบบนี้จะเป็นดังนี้

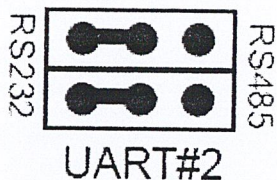
RS232#1: ขั้วต่อนี้จะถูกต่อไว้ที่ขั้ว Connector 4 Pin โดยจะเชื่อมต่อสัญญาณมาจาก P3.0 (Rx/D_0) และ P3.1 (Tx/D_0) ซึ่งขั้วต่อนี้จะนอกจะใช้ในการ สื่อสารทาง RS232 แบบปกติแล้วยัง ใช้สำหรับ Download โปรแกรม ลงใน MCU ด้วย โดยไม่ต้องทำการ Set Jumper ใดๆ เพียงแต่เวลาจะ Download โปรแกรม จะต้องทำการกดสวิตช์ PSEN และ RESET ดังต่อไปนี้เพื่อเข้าสู่ Monitor Mode

- กด SW. PSEN ค้างไว้
- ตามด้วยการกด SW. RESET ค้างไว้
- ปล่อย SW. RESET ในขณะที่ SW. PSEN ยังถูกกดค้างอยู่
- ปล่อย SW. PSEN เป็นลำดับสุดท้าย

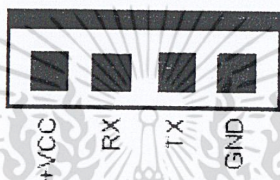
หลังจาก Download เสร็จเรียบร้อยให้ทำการกด SW. RESET เพื่อให้ MCU เริ่ม Run โปรแกรมที่เขียน ซึ่ง Port RS232#1 ก็จะเข้าสู่การทำงานแบบปกติ คือ การสื่อสารของ Port ก็จะถูกรควบคุมตามโปรแกรมที่ผู้ใช้เขียน

RS232#2: สำหรับขั้วต่อนี้ จะใช้สำหรับสื่อสารข้อมูลทาง RS232 เช่นกัน ซึ่งจะแยกอิสระกับขั้วต่อ RS232#1 โดยจะถูกต่อไว้ที่ขั้ว Connector 4 Pin ซึ่งขั้วต่อนี้จะเชื่อมต่อสัญญาณผ่าน Jumper UART#2 ไปยังขาสัญญาณ P6.0(R xD_1) และ P6.1 (Tx/D_1) เนื่องจาก

ขาสัญญาณ P6.0 และ P6.1 นี้จะถูกนำไปใช้งานสำหรับขั้วต่อ RS422/485 ด้วย ดังนั้นเวลาจะใช้งานขั้วต่อ RS232#2 จะต้อง Set Jumper UART#2 มาทางด้าน RS232 ดังแสดงในรูป

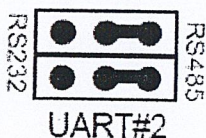


รูปที่ 2.6 แสดงการ Set Jumper UART#2 เมื่อใช้งานขั้วต่อ RS232#2



รูปที่ 2.7 แสดงการจัดเรียงขาขั้วต่อ RS232#1 และ RS232#2

RS422/485: ขั้วต่อนี้จะใช้สื่อสารข้อมูลแบบ RS422 หรือ RS485 ใดๆอย่างหนึ่ง ซึ่งการใช้งานขั้วต่อนี้จะต้องทำการต่อ IC Line Driver 75176 จำนวน 2 ตัว สำหรับ RS 422 หรือ 1 ตัวสำหรับ RS485 ลงใน Socket เสียก่อนจากนั้นก็ทำการ Set Jumper UART#2 มาทางด้าน RS485 ดังรูปที่ 2.8 ต่อมาก็ทำการ Set Jumper 422/485 ไปทางด้านที่จะใช้งาน ซึ่งเมื่อเลือกมาทางด้าน 422 จะต้อง Set Jumper FULL/HALF ไปทางด้าน FULL เพื่อให้ IC Line Driver 2ตัวในการรับข้อมูล(ICตัวใน) และส่งข้อมูล (ICตัวนอก) แต่ถ้าเลือกมาทางด้าน 485 ให้ Set Jumper FULL/HALF มาทางด้าน HALF เพื่อให้ IC Line Driver ตัวนอกตัวเดียวในการรับและส่งข้อมูล (แบบ 2 Line) โดยใช้ขา P3.7 เป็นตัวควบคุมการรับ-ส่งข้อมูล (P3.7 = 1 : Tx , P3.7 = 0 : Rx)



1) Set Jumper. UART#2

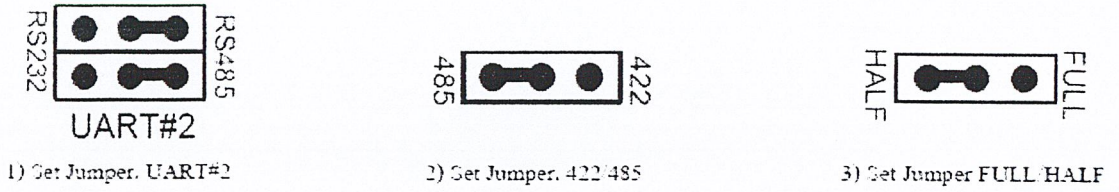


2) Set Jumper. 422/485

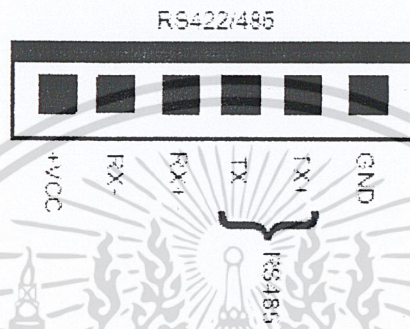


3) Set Jumper FULL/HALF

รูปที่ 2.8 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS422

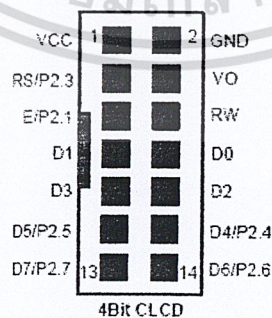


รูปที่ 2.9 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS485



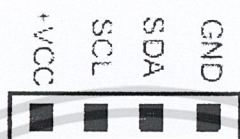
รูปที่ 2.10 แสดงการจัดเรียงขาของขั้วต่อ RS422/485

ขั้วต่อ CLCD: ขั้วต่อนี้จะใช้สำหรับต่อ DOT Matrix LCD โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 14 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก Port P2 ของ MCU การจัดวงจรของ Port CLCD นี้จะต่อในลักษณะแบบ 4 บิต มี VR ต่อไว้สำหรับปรับความเข้มของ LCD ให้ด้วย เมื่อผู้ใช้จะใช้งานจะต้องต่อ PIN ของ Module LCD ให้ตรงกับขาที่กำหนดไว้บน Port ด้วย โดยมีการจัดเรียงขาดังนี้



รูปที่ 2.11 แสดงการจัดเรียงขาของขั้วต่อ CLCD แบบ 4 บิต

Socket DS13 : สำหรับ Socket 1307 ที่จัดไว้บนบอร์ดนี้ จะเป็นการจัดวงจรไว้สำหรับรองรับการต่อใช้งาน RTC #DS1307 ซึ่งจะเป็นการสื่อสารแบบ I2C เมื่อผู้ใช้จะใช้งานจะต้องนำ IC RTC #DS1307 นี้มาเสียบที่ Socket จากนั้นก็ทำการต่อสาย SDA และ SCL จาก Connector 4 Pin ที่อยู่ข้างๆซึ่งได้เชื่อมต่อมาจาก ขา SDA และ SCL ของDS1307 ไว้แล้วไปเข้าขา I/O ของ MCU ที่จะใช้ควบคุม ซึ่ง MCU เมอร์รี่ไม่มีขา I2C ให้ใช้โดยตรงดังนั้นจะต้องสร้างขา SDA และ SCL จาก I/O เาเอง



รูปที่ 2.12 แสดงการจัดเรียงขาของขั้วต่อ #DS1307 4 PIN

2.2 GSM/GPRS Module

GR-64 เป็น โมดูล GSM/GPRS รองรับ 4 ความถี่ ได้แก่ GSM 850/900/1800/1900 MHz โดยปกติหากจะใช้งานตัว GR64 ซึ่งเป็น โมดูลที่มีขาต่อใช้งาน 60 pin GR-64 อาจยังไม่ครอบคลุมการใช้งานตัวโมดูล GR-64 ทุกประการเนื่องจากโมดูลตัวนี้สามารถนำไปประยุกต์ใช้งานที่หลากหลาย การใช้งานตัว GR-64 นี้สามารถติดต่อ โดยผ่านทาง RS-232 ด้วย AT Command ซึ่งสามารถใช้โปรแกรม STERM หรือจะใช้โปรแกรม Terminal อื่นๆ เช่น hyper terminal

GR64 GSM/GPRS Module เป็นบอร์ดใช้สำหรับรับส่งข้อมูลผ่านทางเครือข่ายโทรศัพท์มือถือถือ การต่อใช้งานต้องต่อใช้งานร่วมกับไมโครคอนโทรลเลอร์ผ่านการสื่อสารข้อมูลทาง RS232 เพื่อใช้ในการสั่งงานด้วยคำสั่ง AT Command สามารถสั่งการโทรออก-รับสาย ,การส่งข้อความสั้น (Short Message Service) ,การรับ-ส่งข้อมูลผ่านทางระบบ GPRS และการทำเป็นโมเด็มไร้สาย (Wireless modem) การจัดสรร Port I/O ของบอร์ดได้ต่อ Port RS232 ไว้สำหรับใช้งานจำนวน 3 Port แหล่งจ่ายไฟของบอร์ดสามารถเลือกต่อกับขั้ว Connector แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านขั้ว Connector แบบ Adaptor จ่ายไฟก็ได้ โดยการแสดงผลการทำงานของแหล่งจ่ายไฟและสถานะการทำงานให้ทราบด้วย LED

2.2.1 คุณสมบัติต่างของ GR64 GSM/GPRS

- คุณสมบัติด้านวิทยุ

- Quad Band GSM/GPRS Radio Device 850/900/1800/1900 MHz
- Mobile Class B
- Extended measurement reporting
- Compliant with 3GPP release 99 protocol stack
- รับส่ง ข้อความสั้น (SMS)
- Cell broadcast

- คุณสมบัติด้านการเชื่อมต่อ

- Dual microphone inputs and speaker outputs
- Up to 12 x programmable general purpose I/Os
- 4 x ADC inputs
- 1 x buzzer output
- Real time clock with alarm output
- 2 x UARTs Connectivity Serial baud rate from 1200 to 460,800 bits/s
- USB 2.0 Full speed end point compliant
- MMCX RF connector
- ใช้งานได้จำนวน 60 ขา
- ควบคุมด้วย AT command

- คุณสมบัติด้าน Data/GPRS

- GPRS Class 10
- Transparent and non-transparent CSD up to 9.6kbps
- Modem type; V21, V22, V33, V22bis, V26ter, V32, V34, V24
- V42bis compression
- GSM supplementary services supported
- GSM 27.010 multiplexing protocol
- USSD
- TCP/UDP/IP protocol stack
- Dynamic & static IP address allocation
- PPP protocol

- คุณสมบัติอื่นๆ

- SIM application toolkit class 2
- 1.8/3V SIM interface with SIM detection
- LED output
- ใช้งานได้ที่อุณหภูมิ -30 ถึง +75 องศาเซลเซียส

2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลแบบอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยความสามารถของ RS-232 ในอดีตนั้น ออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้ส่งผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดหนึ่งซึ่งอยู่ห่างไกล โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronics Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้ขั้วต่อเป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีค่าระดับสัญญาณตั้งแต่ -3 โวลต์จนถึง -12 โวลต์ แสดงว่ามีข้อมูล (Mark) และค่าระดับสัญญาณ +3 โวลต์จนถึง +12 โวลต์ แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ถูกใช้ในการกำหนดรูปแบบการสื่อสารข้อมูลระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

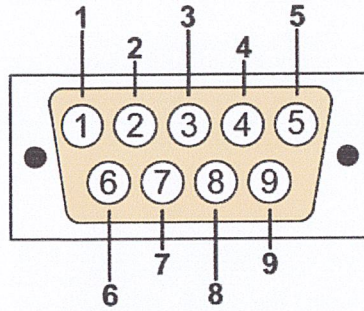
ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เราเห็นได้ชัด คือ ขั้วต่อของ DTE จะเป็นตัวผู้ ส่วนขั้วต่อของ DCE จะเป็นตัวเมีย โดยพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่โดยทั่วไปจะเป็นแบบ DTE ส่วนขั้วต่อที่อยู่ในโมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานคอมพิวเตอร์ พอร์ตอนุกรม RS-232 ถูกใช้เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์กับโมเด็ม แมส และเครื่องพิมพ์ที่สามารถใช้พอร์ตอนุกรมในการติดต่อได้

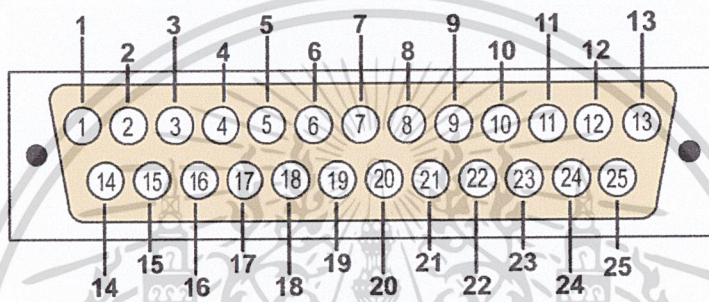
2.3.1 ขั้วต่อสำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้ขั้วต่อแบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งขั้วต่อแบบ DB-25 จะมีขาใช้งานเพียง 9 เส้น เช่นเดียวกับขั้วต่อแบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยมีการใช้งานมาในอดีต ในปัจจุบันไม่ค่อยมีความสำคัญมากนักจึงถูกยกเลิกไป ในรูปที่ 2.13 และรูปที่ 2.14 นั้นเป็นการแสดงตำแหน่งขาต่าง ๆ ของขั้วต่อทั้งสองแบบ โดยเป็นภาพที่มองจากด้านหลังคอมพิวเตอร์ และสำหรับตารางที่ 2.1 เป็นการแสดงหน้าที่การทำงานของขาต่างๆ ในขั้วต่อทั้งสองแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า



รูปที่ 2.13 ขั้วต่ออนุกรม 9 ขา หรือแบบ DB-9



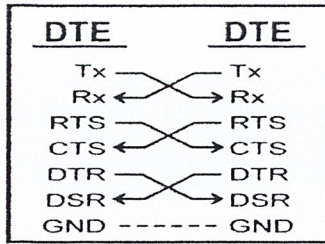
รูปที่ 2.14 ขั้วอนุกรม 25 ขา หรือแบบ DB-25

ตารางที่ 2.1 การจัดขาของสัญญาณพอร์ตอนุกรมในแบบต่างๆ และหน้าที่การทำงาน

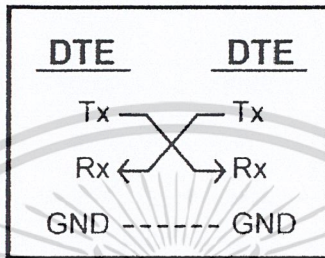
ขั้วต่อ DB-9	ขั้วต่อ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DTC	Input
2	3	Received Data : RXD	Input
3	2	Transmitted Data : TXD	Output
4	20	Data Terminal Ready : DTR	Output
5	7	Single Ground : GND	-
6	6	Data Set Ready : DSR	Input
7	4	Request To Send : RTS	Output
8	5	Clear To Send : CTS	Input
9	22	Ring Indicator : RI	Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 13
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา Data Carrier Detect: DCD หรืออาจจะเรียกว่า Carrier Detect: CD ขานี้จะทำงานเมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ถูกนำมาใช้งานมากนัก
- ขา Receive Data: RD หรือ RXD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยจะนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- ขา Transmitted Data: TD หรือ TXD ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์โดยทำการส่งข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- ขา Data Terminal Ready: DTR เป็นขาเอาต์พุตที่ใช้สำหรับรับส่งสัญญาณออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อกับอุปกรณ์ปลายทาง โดยขา DTR นี้จะต้องเชื่อมกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ และถ้าใช้การเชื่อมต่อแบบ 3 สาย ต้องเชื่อมต่อขา DTR และ DSR ของพอร์ตอนุกรมเข้าด้วยกัน และจะต้องเชื่อมต่อเข้ากับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห้
- ขา Single Ground: GND เป็นขาราวด์ของสัญญาณ
- ขา Data Set Ready: DSR ขานี้จะใช้ควบคู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอก
- ขา Request To Send: RTS เป็นขาเอาต์พุตสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมาให้คอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สาย จะต้องเชื่อมต่อกับขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- ขา Clear To Send: CTS เป็นขาอินพุตทำหน้าที่รอรับสัญญาณที่ส่งเข้ามา เมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ขานี้จะใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลแล้วหรือยัง
- ขา Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับ โมเด็มแล้วยังมีความต้องการตรวจสอบสัญญาณเรียกสายโทรศัพท์



รูปที่ 2.15 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมแบบ Null Modem



รูปที่ 2.16 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมใช้สายสัญญาณน้อยที่สุดเพียง 3 เส้น

การเชื่อมต่อในรูปที่ 2.15 เป็นการเชื่อมต่อแบบ Null Modem หรือการเชื่อมต่อโดยตรง โดยไม่ต้องผ่าน โมเด็ม ส่วนการเชื่อมต่อในรูปที่ 2.16 เป็นการเชื่อมต่อโดยสัญญาณน้อยสุดเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์

2.4 อุปกรณ์ตรวจจับสถานการณ์ผิดปกติ

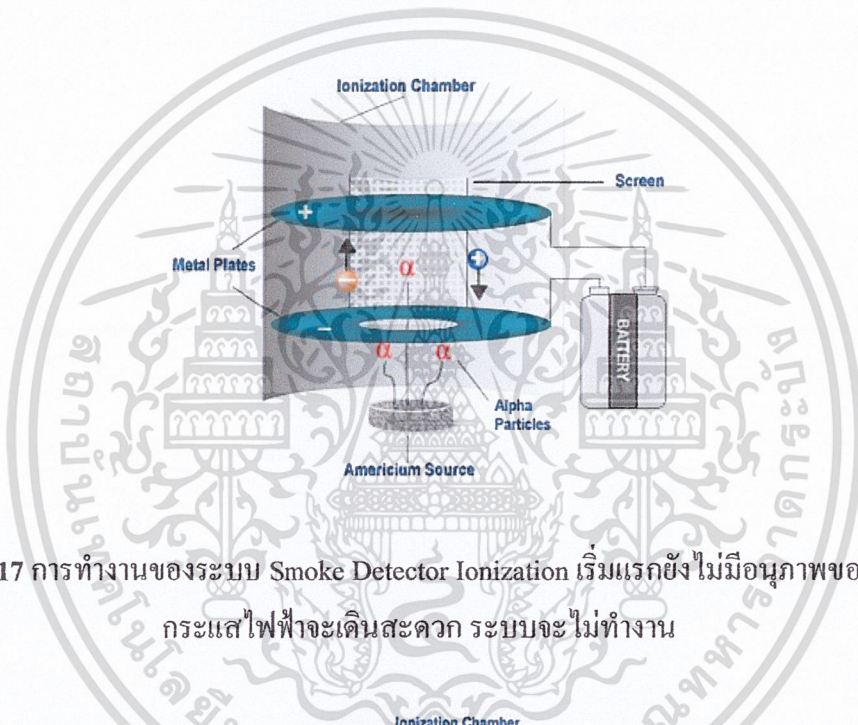
2.4.1 อุปกรณ์ตรวจจับควัน

อุปกรณ์ตรวจจับควัน (Smoke Detector) เป็นอุปกรณ์ที่ทำหน้าที่ตรวจสอบอนุภาคของควันโดยอัตโนมัติ โดยมากการเกิดเพลิงไหม้จะเกิดควันไฟก่อน จึงทำให้อุปกรณ์ตรวจจับควันสามารถตรวจการเกิดเพลิงไหม้ได้ในการเกิดเพลิงไหม้ระยะแรก แต่ก็มีข้อบกพร่องในการเกิดเพลิงไหม้บางกรณีจะเกิดควันไฟน้อยจึงไม่ควรนำ อุปกรณ์ตรวจจับควันไปใช้งาน เช่น การเกิดเพลิงไหม้จากสารเคมีบางชนิด

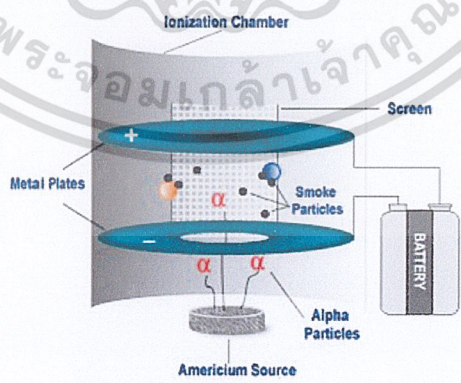
หลักการทำงาน โดยทั่วไปอุปกรณ์ตรวจจับควันจะทำงานโดยอาศัย หลักการคือเมื่อมีอนุภาคควัน ลอยเข้าไปในอุปกรณ์ตรวจจับควัน อนุภาควัสดุจะเข้าไปกีดขวางวงจรไฟฟ้า หรือกีดขวางระบบแสงในวงจร หรือใช้อนุภาคควันในการหักเหแสงไปที่ตัวรับแสงชนิดของอุปกรณ์

ตรวจจับควัน แบ่งได้ 2 ประเภทใหญ่ คือแบบ ชนิดไอโอไนเซชัน (Ionization), ชนิดโฟโตอิเล็กทริก (Photoelectric)

อุปกรณ์ตรวจจับควันชนิด ไอออนไนเซชัน (Smoke Detector Ionization Type) ภายในเป็นกล่อง (Chamber) มีแผ่นโลหะที่มีขั้วไฟฟ้าต่างกัน ที่มีสารกัมมันตภาพรังสี (Radioactive) ซึ่งจำทำหน้าที่กระตุ้นอากาศภายในให้เกิดการแตกตัวเป็นไอออน โดยไอออนในกล่องจะทำหน้าที่เป็นตัวนำไฟฟ้าให้กระแสไฟฟ้าไหลผ่านทั้งสองขั้ว เมื่อเกิดควันเข้าไปในกล่อง จะทำให้ค่าความนำไฟฟ้าของอากาศลดและกระแสไฟฟ้าจะลดลงเรื่อยตามปริมาณควันจนถึงค่าที่กำหนดไว้ ระบบจะทำงาน



รูปที่ 2.17 การทำงานของระบบ Smoke Detector Ionization เริ่มแรกยังไม่มือนุภาพของควัน กระแสไฟฟ้าจะเดินสะดวก ระบบจะไม่ทำงาน

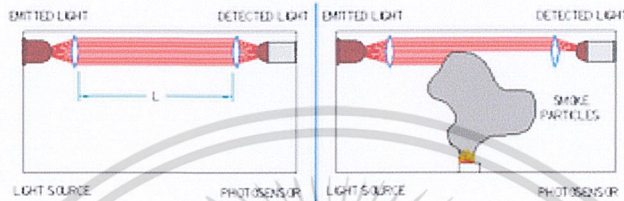


รูปที่ 2.18 เมื่อมือนุภาพควันมาติดที่แผ่น Screen (เห็นเป็นจุดดำๆ) จะเป็นตัวขัดขวางกระแสไฟฟ้า ทำให้กระแสไฟฟ้าลดต่ำลงจนถึงจุดที่ระบบทำงาน

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

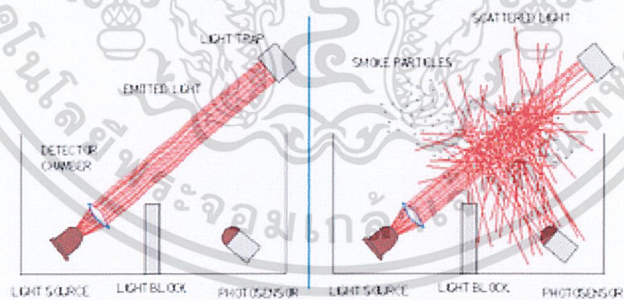
อุปกรณ์ตรวจสอบควัน ชนิดโฟโตอิเล็กทริก (Smoke Detector Photoelectric Type) มีหลักการทำงานสองแบบคือ แบบหักเหของแสง และแบบใช้ควันกีดขวางแสง

- แบบควันกีดขวางแสง (Light Obscuration) ทำงานโดยใช้แหล่งกำเนิดแสง (Emitted Light) ยิงเข้าที่ตัวรับแสง (Detector Light) เมื่อไม่มีควันไฟปริมาณแสงจะคงที่ๆค่าหนึ่งเสมอ เมื่อมีอนุภาคควันเข้ามาดังรูปขวามือ อนุภาคควันจะเข้าไปกีดขวางลำแสง แสงที่ส่องเข้าตัวรับจะต่ำลงเรื่อยจนถึงค่าที่กำหนดไว้ระบบจะทำงาน



รูปที่ 2.19 ระบบการทำงานของ อุปกรณ์ตรวจจับควัน แบบกีดขวางแสง (Light Obscuration)

- แบบหักเหแสง (Light Scattering) ทำงาน โดยมีแหล่งกำเนิดแสง แต่จะไม่ยิงไปที่ตัวรับแสงโดยตรง จะอาศัยหลักการที่ว่า เมื่อมีอนุภาคควันเข้ามาในอุปกรณ์ อนุภาคควันจะหักเหแสงบางส่วนไปที่ตัวรับแสง เมื่อมีควันมากขึ้นแสงก็จะหักเหเข้าตัวรับแสงมากขึ้นจนถึงจุดๆหนึ่งที่ระบบจะทำงาน

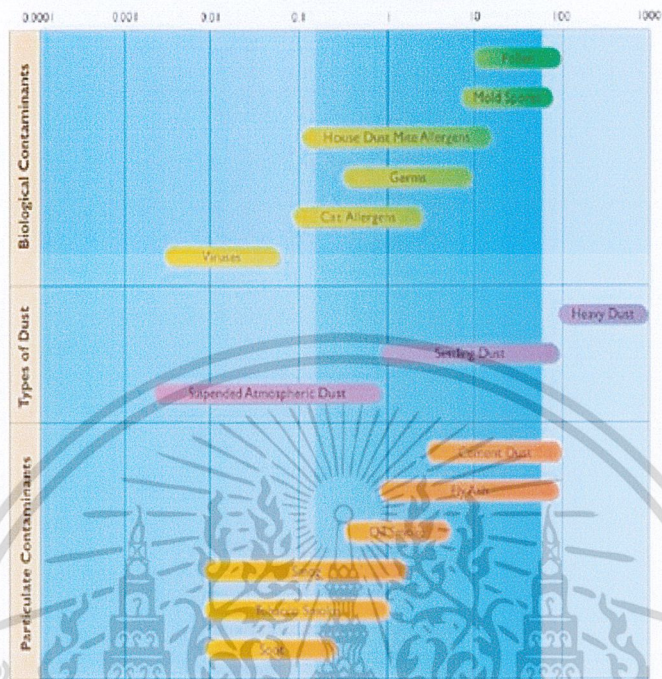


รูปที่ 2.20 ระบบการทำงานของ อุปกรณ์ตรวจจับควัน แบบหักเหแสง (Light Scattering)

ข้อดี เหมาะ กับการตรวจจับควันที่มีขนาดตั้งแต่ใหญ่ตั้งแต่ 1 ไมครอนขึ้นไป คือควันที่เกิดจากการสันดาปไม่สมบูรณ์ เช่นเกิดเพลิงไหม้ในที่อับอากาศ

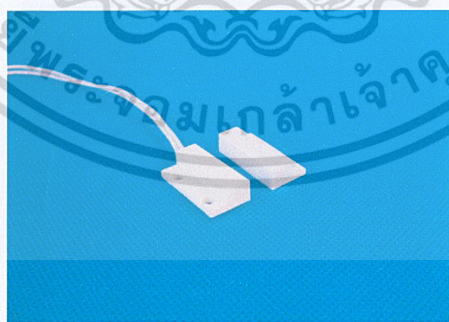
- แบบกล้องหมอกควัน (Cloud Chamber Type) ทำงานโดยการสูมตัวอย่างอากาศ โดยการดูดอากาศในพื้นที่เข้าไปในกล้องที่มีความชื้นสูงและอากาศที่ถูกดูดเข้าไปจะถูกลดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ลดความดันลงอย่างช้าๆ โดยถ้าอากาศที่ถูกดูดเข้ามีอนุภาคควันอยู่จะกลั่นตัวกลายเป็นหมอกถ้า อนุภาคควันมากความหนาแน่นของหมอกจะสูงจนถึงจุดที่ระบบทำงาน



รูปที่ 2.21 Chart ขนาดอนุภาคควันต่างๆ หน่วยแกน x เป็น ไมครอน

2.4.2 อุปกรณ์แอมป์หลัก



รูปที่ 2.22 แอมป์หลัก

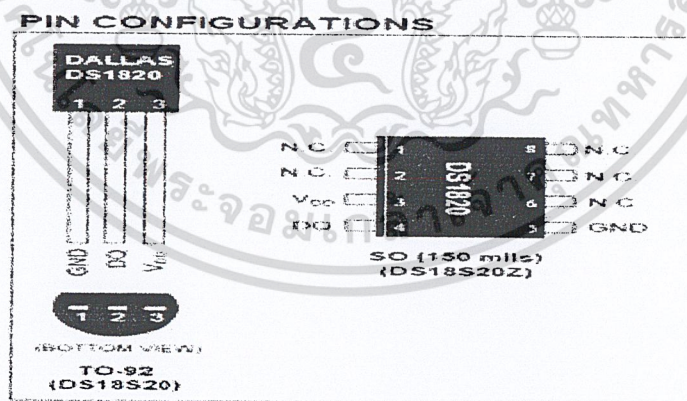
แอมป์หลักเป็นอุปกรณ์ที่ใช้ร่วมกับวงจรของระบบรักษาความปลอดภัย ทำงาน โดยต่อเข้ากับระบบ เพื่อให้กระแสไฟฟ้าไหลผ่านให้ครบวงจร ซึ่งจะแสดงผลโดยหลอด LED หากกระแสไฟฟ้าไหลผ่านครบวงจร หลอด LED จะติด หากแอมป์หลักหลุดออกจากกันกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไฟฟ้าจะไหลไม่ครบวงจร หลอด LED ก็จะดับ จากนั้นระบบจะทำการส่งข้อความสั้น (SMS) แจ้งเตือนไปให้ผู้ใช้งานทราบ

2.4.3 เซนเซอร์วัดอุณหภูมิ DS18S20

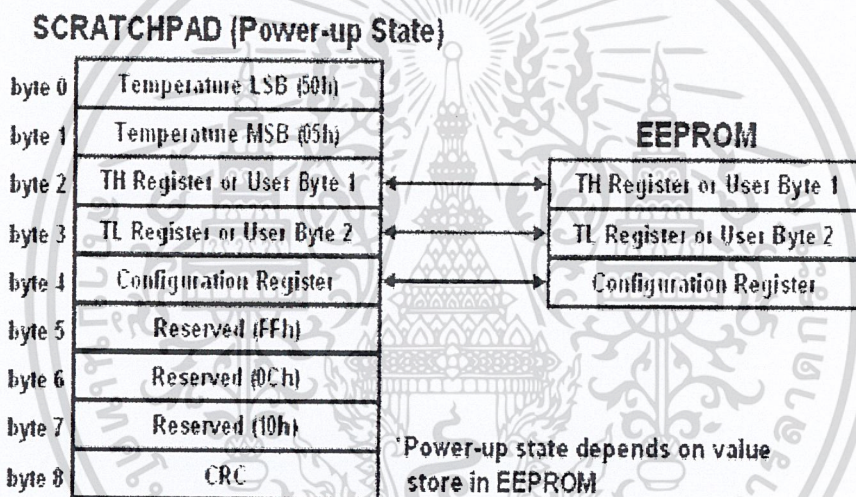
ไอซี DS18S20 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย ใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาดำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจนการถ่ายทอดข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายทอดข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน จุดที่วัดความแม่นยำของค่าที่วัดได้ใช้ไอซีเบอร์ DS18S20 ซึ่งมีระดับความผิดพลาดที่ 0.5 องศาเซลเซียส DS18S20 เป็น IC วัดอุณหภูมิแบบดิจิตอล ของ Dallas Semiconductor สามารถวัดอุณหภูมิเป็นหน่วยองศา C ในช่วง -55C ถึง 125 C ที่ความละเอียด 9-12 บิต และมีความแม่นยำอยู่ที่ 0.5C ในช่วง -10C ถึง 85 C ในกรณีที่เป็นตัวถังแบบ TO-92 นั้นจะมีโครงสร้าง และขาตั้งแสดงในรูปที่ 2.23



PIN	SYMBOL	Description
1	GND	Ground
2	DQ	Data Input/ Output pin
3	Vdd	Optional Vdd pin

รูปที่ 2.23 โครงสร้าง และขาของ DS18B20 ตัวถังแบบ TO-92

การสื่อสารและควบคุม DS18S20 นั้นสามารถทำได้โดยใช้บัสข้อมูลแบบ 1-wire ของ Dallas Semiconductor ซึ่งใช้สายสัญญาณเพียงเส้นเดียวเท่านั้น ภายใน DS18S20 แต่ละตัวมีโค้ดประจำตัวขนาด 64 บิต ทำให้สามารถใช้งาน DS18S20 หลายตัวทำงานบนบัสแบบ 1 wire พร้อมกันได้ นอกจากนี้ DS18S20 ยังสามารถทำงานในโหมดพาราสิติก (Parasite Power Mode) ซึ่งเป็นการทำงานโดยไม่ใช้ไฟเลี้ยง แต่ใช้พลังงานจากสายสัญญาณ 1-wire ซึ่งมีประโยชน์มากสำหรับการวัดอุณหภูมิระยะไกล หรือในการใช้งานในที่ ๆ มีเนื้อที่จำกัด โครงสร้างรีจิสเตอร์ภายในของ DS18S20 มีลักษณะดังแสดงในรูปที่ 2.24 จะเห็นได้ว่าประกอบไปด้วย SRAM Scratchpad ขนาด 9 ไบต์ และ EEPROM ขนาด 3 ไบต์ ซึ่งใช้เก็บค่าอุณหภูมิสูงสุด (TH) ต่ำสุด (TL) สำหรับเปรียบเทียบการเกิดสัญญาณเตือน และรีจิสเตอร์ควบคุม (Configuration Register)



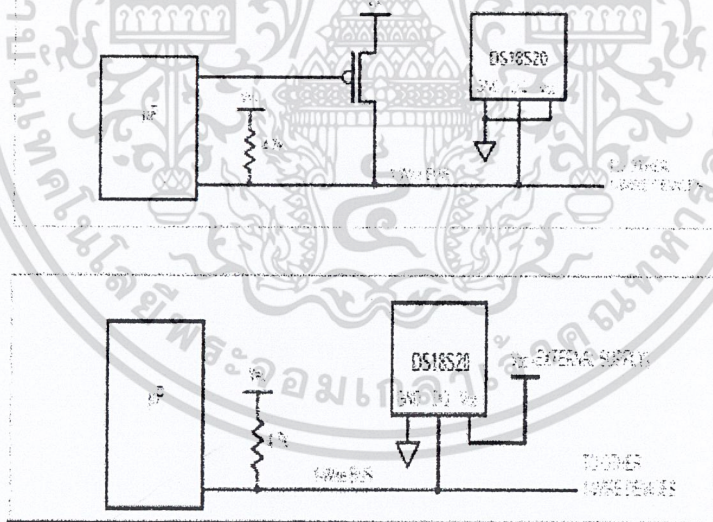
รูปที่ 2.24 โครงสร้างรีจิสเตอร์ภายใน DS18S20

ข้อมูลอุณหภูมิที่วัดได้จะถูกเก็บอยู่ในรีจิสเตอร์ Temperature ซึ่งมีขนาด 16 บิต ดังแสดงในรูปที่ 2.25 ถ้าข้อมูลอุณหภูมิเป็นบวก S จะเป็น “1” แต่ถ้าข้อมูลอุณหภูมิเป็นลบ S จะเป็น “0” ในกรณีที่ DS18S20 ทำงานในโหมดความละเอียด 12 บิต บิตทุกบิตในรีจิสเตอร์ Temperature จะถูกใช้หมด แต่ในกรณีที่ทำงานในโหมด 9-11 บิต บิตต่าง (บิต 0 – บิต 2) จะไม่ถูกใช้งาน ซึ่งในการกำหนดโหมดความละเอียดการทำงานของ DS18S20 นั้นสามารถกำหนดได้ที่รีจิสเตอร์ Configuration ซึ่งโดยปกติเริ่มต้น DS18S20 จะทำงานในโหมด 12 บิต

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

รูปที่ 2.25 โครงสร้างภายในรีจิสเตอร์ Temperature

การสื่อสารแบบ 1-wire เป็นระบบบัสข้อมูลแบบ Half-duplex นั่นคือสามารถสื่อสารได้ 2 ทิศทาง แต่ไม่สามารถรับ และส่งข้อมูลพร้อมกันในช่วงเวลาเดียวกันได้ ระบบบัสมีการทำงานเป็นแบบ Master/Slave โดยอุปกรณ์ Master จะเป็นตัวควบคุมสถานะ และจังหวะการรับส่งของบัสข้อมูล ในขณะที่อุปกรณ์ Slave จะทำงานตามการควบคุมของอุปกรณ์ Master เท่านั้น ในการใช้งานบัสแบบ 1 wire นี้ สายสัญญาณข้อมูล DQ จะต้องมีความถี่ที่ลอจิกสูง สามารถทำได้โดยการต่อตัวต้านทานประมาณ 5 กิโลโอห์ม พูลอัพไว้กับไฟเลี้ยง หรือในกรณีที่ใช้บัสแบบ 1 wire ต่อร่วมกับอุปกรณ์ DS18S20 หลายตัว ก็ยังสามารถทำได้ดังแสดงในรูปที่ 2.26

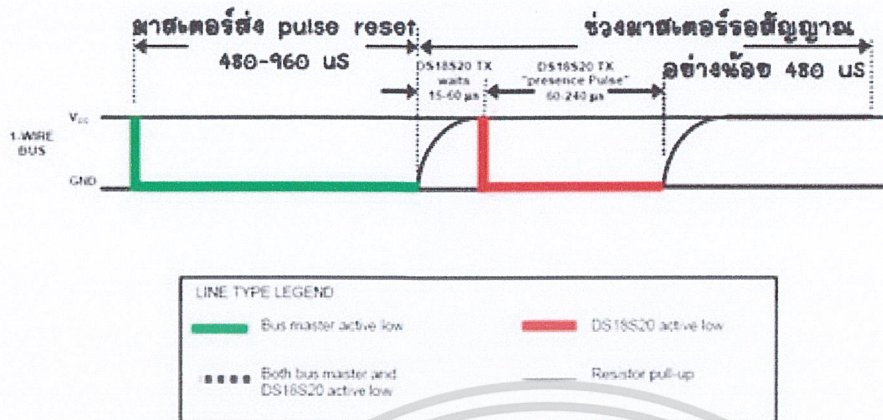


รูปที่ 2.26 การต่อวงจรเพื่อใช้งาน DS18S20

หลักการทำงานของเซนเซอร์วัดอุณหภูมิ DS18S20

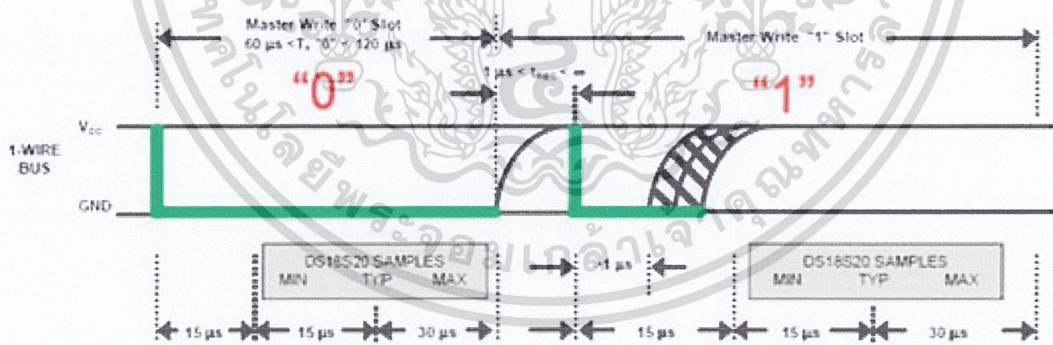
ในกระบวนการเริ่มต้นการสื่อสารแบบ 1-wire ทั้งหมดนั้น อุปกรณ์ Master ต้องขอเริ่มการสื่อสารด้วยการสร้าง Reset pulse ก่อน เมื่ออุปกรณ์ Slave ได้รับ Reset pulse ก็จะสร้างเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Presence pulse เพื่อตอบรับการขอเริ่มการสื่อสารนั้น ซึ่งมีรายละเอียดของช่วงเวลาต่าง ๆ ดังแสดงในรูปที่ 2.27



รูปที่ 2.27 การเริ่มการติดต่อสื่อสารแบบ 1-wire ด้วย Reset Pulse และ Presence Pulse

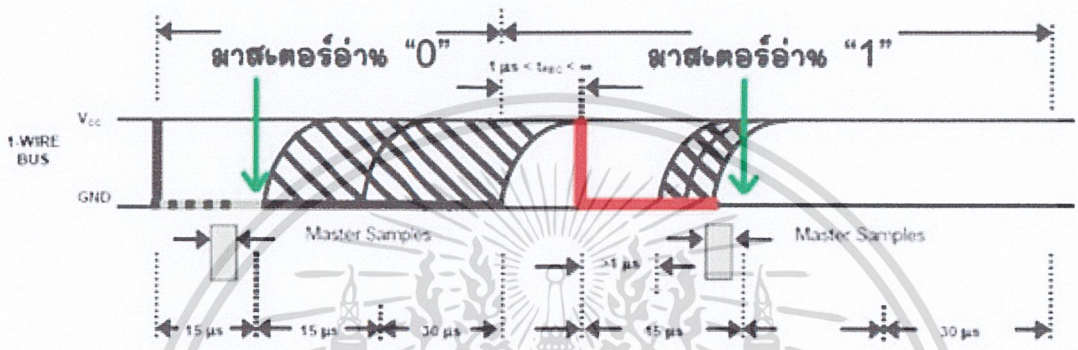
ในการเขียนข้อมูลแบ่งออกเป็น 2 ชนิดคือการเขียนข้อมูล “0” และการเขียนข้อมูล “1” ดังแสดงในรูปที่ 2.28 การเขียนข้อมูลลง DS18S20 ต้องใช้ช่วงเวลาของไทม์สล롯อย่างต่ำ 60 usec และต้องมีช่วงเวลาระหว่างไทม์สล롯อย่างต่ำ 1 usec



รูปที่ 2.28 การเขียนข้อมูลจาก Master ลง DS18S20

การเขียนข้อมูลทั้ง 2 ชนิด เริ่มแรกอุปกรณ์ Master ต้องดึงสัญญาณบนบัส 1-wire ลงมาให้อยู่ในสถานะลอจิกต่ำก่อน ในกรณีที่ต้องการเขียนข้อมูล “0” ลงใน DS18S20 อุปกรณ์ Master ต้องดึงสัญญาณบนบัสให้เป็นลอจิกต่ำต่อ จนกว่าจะครบช่วงเวลาไทม์สล็อต (อย่างต่ำ 60 usec)

ส่วนในกรณีที่ต้องการเขียนข้อมูล “1” ลง DS18S20 อุปกรณ์ Master ต้องปล่อยบัส เพื่อให้บัส กลับไปอยู่ในสถานะลอจิกสูงก่อนการ Sampling ของ DS18S20 ซึ่งจะอยู่ในช่วง 15 usec-60 usec หลังจากที่อุปกรณ์ Master ค้างสัญญาณบัส 1-wire ลงมาในการอ่านค่าภายใน SRAM ของ DS18S20 สามารถทำได้ก็ต่อเมื่ออุปกรณ์ Master ได้เขียนข้อมูลเพื่อขอทำการอ่านค่าใน SRAM (Read Scratchpad) ซึ่งมีค่าเป็น 0xBE ลงไปที่ DS18S20 เสียก่อน จากนั้นจึงเริ่มอ่านข้อมูลจากบัส 1-wire โดยไทม์สล๊อตของการอ่านต้องมีช่วงเวลาอย่างต่ำ 60 usec และต้องมีช่วงเวลาระหว่าง ไทม์สล๊อตอย่างต่ำ 1 usec ดังแสดงในรูปที่ 2.29



รูปที่ 2.29 การอ่านข้อมูลจาก DS18S20

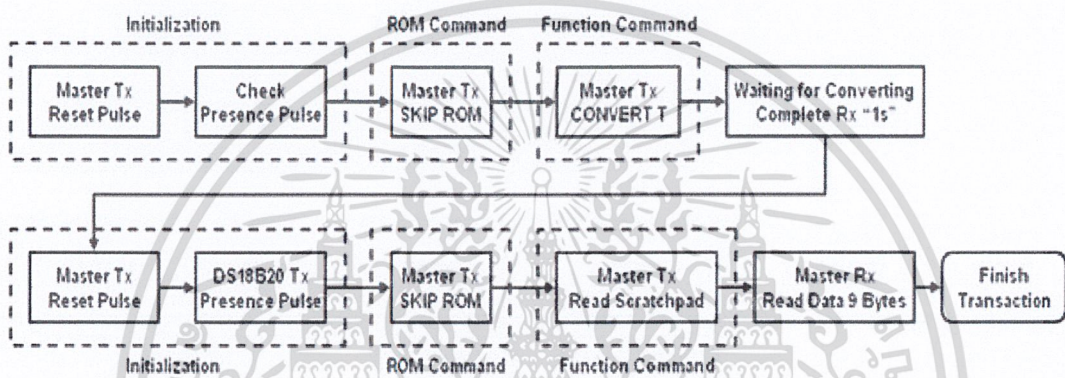
การอ่านข้อมูลจากบัส 1-wire เริ่มแรกอุปกรณ์ Master จะต้องดึงบัส 1-wire ลงให้อยู่ในสถานะลอจิกต่ำเป็นช่วงเวลาอย่างน้อย 1 usec จากนั้นจึงค่อยปล่อยบัส ในกรณีที่ DS18S20 ส่งข้อมูล “0” DS18S20 จะดึงบัสให้เป็นลอจิกต่ำจนจนกว่าจะสิ้นสุดไทม์สล๊อตถึงจึงจะปล่อยบัสให้กลับไปยังสถานะลอจิกสูง ส่วนในกรณีที่ DS18S20 ส่งข้อมูล “1” DS18S20 จะปล่อยบัสให้อยู่ในสถานะลอจิกสูงตลอด ในการ Sample เพื่อรับข้อมูลจาก DS18S20 ควรทำภายใน 15 usec หลังจากจุดเริ่มของไทม์สล๊อตดังแสดงในรูปที่ 2.29

ขั้นตอนการใช้งาน DS18S20 มี 3 ขั้นตอนด้วยกันคือ

1. Initialization
2. ROM Command
3. DS18S20 Function Command

ขั้นตอนการอ่านค่าอุณหภูมิจาก DS18S20 นั้น แบ่งออกเป็น 2 ขั้นตอนใหญ่ ๆ คือ Converting Temperature ซึ่งเป็นการแปลงอุณหภูมิให้อยู่ในรูปแบบข้อมูลดิจิตอลเก็บไว้ใน Scratchpad และ Read Scratchpad ซึ่งเป็นการอ่านข้อมูลที่เก็บใน Scratchpad นั้นออกมา

จะเห็นได้ว่าไมโครคอนโทรลเลอร์ซึ่งเป็นอุปกรณ์ Master จะต้องทำการ Initialization จากนั้นจึงค่อยคำสั่ง SKIP ROM [CCh] ซึ่งเป็น ROM Command ที่ใช้ในกรณีที่ต่อใช้งาน DS18S20 เพียงตัวเดียวจากนั้นจึงค่อยส่ง Function Command CONVERT T [44h] เพื่อสั่งให้ DS18S20 ทำการแปลงอุณหภูมิให้อยู่ในรูปแบบดิจิทัลและเก็บไว้ใน Scratchpad กระบวนการแปลงนี้จะใช้เวลาประมาณ 750ms ซึ่งเราสามารถตรวจสอบสถานะการแปลงข้อมูลได้จากการเช็คสถานะของบัส 1-wire ในกรณีที่ DS18S20 ยังทำการแปลงข้อมูลอยู่บัส 1-wire จะมีสถานะเป็นลอจิกต่ำ แต่ถ้า DS18S20 ได้ทำการแปลงข้อมูลเสร็จเรียบร้อยแล้วบัส 1-wire จะกลับมาอยู่ในสถานะปกติคือลอจิกสูง



รูปที่ 2.30 ขั้นตอนการแปลง และอ่านอุณหภูมิจาก DS18S20

เมื่อ DS18S20 แปลงข้อมูลอุณหภูมิเป็นดิจิทัลเก็บไว้ใน Scratchpad แล้วไมโครคอนโทรลเลอร์สามารถอ่านข้อมูลภายใน Scratchpad นั้นโดยทำการ Initialization บัส 1-wire อีกครั้ง จากนั้นจึงส่งคำสั่ง SKIP ROM ซึ่งเป็น ROM Command และส่งคำสั่ง READ SCRATCHPAD ซึ่งเป็น Function Command เพื่อขออ่านข้อมูล Scratchpad ภายใน DS18S20 จากนั้นไมโครคอนโทรลเลอร์จึงค่อยทำการอ่านข้อมูลภายใน Scratchpad ของ DS18S20 มาทีละไบต์จนครบ 9 ไบต์ และแสดงผลข้อมูลอุณหภูมิที่อยู่ซึ่งเป็นข้อมูลในไบต์แรก และ ไบต์ 2 ที่อ่านมาจาก Scratchpad นั้นออกมาทางพอร์ตอนุกรม



รูปที่ 2.31 แสดงเซ็นเซอร์ตรวจจับความร้อน

2.5 อุปกรณ์ รับ-ส่ง สัญญาณรีโมท

รีโมทเป็นอุปกรณ์ที่ใช้ร่วมกับวงจรของระบบรักษาความปลอดภัย ทำงาน โดยต่อเข้ากับระบบ เพื่อให้สามารถ เปิดปิด การใช้งานของระบบรักษาความปลอดภัยได้

ลักษณะทั่วไป

- ใช้ไฟ DC 12V 800 MA ขึ้นไป (แนะนำ ให้ใช้เป็นอแดปเตอร์ 12V. กระแส 1A. ขึ้นไป)
- ใช้คลื่นความถี่ย่าน UHF
- ตัวแม่เรียนรู้อุปกรณ์ได้
- ตั้งให้ทำงานเป็นกดติด/กดดับและกดติด/ปล่อยดับได้
- ตั้งให้จำ สถานะเมื่อไฟดับได้
- ตั้งให้ทำงานเป็นซีเล็กเตอร์ได้
- ตัวลูกมีรหัส 6 หลัก (16 ล้านรหัส) ที่ฝาหลังด้านในถ้าต้องการส่งตัวลูกเพิ่ม โปรดแจ้งรหัสนี้มาด้วย
- รัศมีการใช้งานประมาณ 100 เมตร (ขึ้นอยู่กับพื้นที่ที่ใช้งานด้วย)

การนำไปใช้งาน

รีโมท ใช้สำหรับควบคุมการดำเนินการของระบบรักษาความปลอดภัย โดยไม่ใช้สายไฟ เป็นตัวส่งสัญญาณ แต่ใช้อินฟราเรดแทน ใช้พลังงานจากแบตเตอรี่ขนาดเล็ก



รูปที่ 2.32 แสดงอุปกรณ์ตัวส่ง

2.6 รีเลย์ (Relay)

เป็นอุปกรณ์ที่ใช้กระแสต่ำๆ เพื่อควบคุมสวิตช์ให้ตัดต่อโหลดที่มีกระแสสูงๆ โครงสร้างส่วนประกอบของรีเลย์ การทำงานของรีเลย์คือ เมื่อมีแรงดันกดคร่อมขดลวดจะทำให้เกิดกระแสไหลผ่านขดลวด ซึ่งจะทำให้หน้าสัมผัสเคลื่อนที่ (moving contact) เกิดสนามแม่เหล็กไฟฟ้าดูดหน้าสัมผัส NO (ปกติเปิดวงจร : normally open) ให้ต่อวงจร และเมื่อปลดแรงดันออกจากสนามแม่เหล็กก็จะหมดลงหน้าสัมผัสเคลื่อนที่ก็จะติดกลับมาต่อยังหน้าสัมผัส NC (ปกติต่อวงจร : normally close)



2.7 สวิตช์รหัสดิจิทัล

คุณสมบัติ

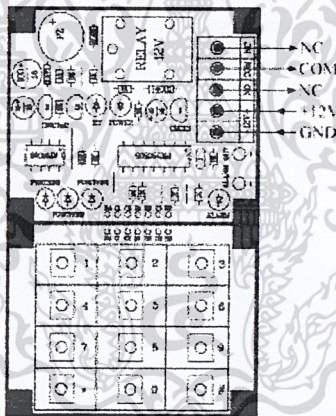
- ใช้ไฟเลี้ยงวงจร 12 DC V ใช้กระแสสูงสุดประมาณ 80 mA
- สามารถกำหนดรหัสได้มากกว่า 1,000,000 รหัสไม่ซ้ำกัน
- สามารถตั้งฟังก์ชันการทำงานได้ 3 ลักษณะ คือ กดรหัสถูกวงจรทำงาน กดรหัสถูกวงจรหยุดทำงาน, กดปุ่มใดๆก็ได้วงจรทำงานกดรหัสถูก วงจรหยุดทำงาน และกดรหัสถูก วงจรทำงานกดปุ่มใดๆ ก็ได้ วงจรหยุดทำงาน

- มีจุดเอาต์พุตที่สามารถควบคุมอุปกรณ์อื่นๆ ได้ ขนาด 12 DC V 5 A
- มีจุดต่อ ALARM เพื่อต่ออุปกรณ์เตือนภัยต่างๆ
- มี LED แสดงผลการทำงานของไฟเข้า, รีเลย์, alarm และฟังก์ชันต่างๆ
- มี IC MEMORY เพื่อช่วยในการเก็บข้อมูลขณะไฟดับ โดยไม่ต้องมีแบตเตอรี่แบ็คอัพ

- สามารถเปิดปิดเสียงสัญญาณขณะกดปุ่มได้และมีสัญญาณเตือนเมื่อมีการกดรหัสถูกหรือผิด

จุดต่อและไฟแสดงผลต่างๆ

- จุด 12V เป็นจุดต่อไฟตรงขนาด 12 V เพื่อนำไปเลี้ยงวงจรทั้งหมด
- จุด NO, COM และ NC เป็นจุดต่อกับอุปกรณ์ที่เราต้องการควบคุม
- จุด ALARM OUT ใช้สำหรับต่อเข้ากับวงจรเตือนภัยต่างๆ (โดยปกติจะมีแรงดันประมาณ 0 V แต่เมื่อมีการเตือนภัยจะมีแรงดันประมาณ 5 V)
- สวิตช์ต่างๆ ใช้ในการกดหมายเลขรหัสและกดเข้าฟังก์ชันต่างๆ
- LED POWER ใช้บอกสถานะ การทำงานของวงจรทั้งหมด
- LED RY ใช้บอกสถานะ การทำงานของรีเลย์
- LED ALARM ใช้บอกสถานะ การทำงานเมื่อมีการกดรหัสผิดเกินกว่า 3 ครั้ง
- LED FUNCTION ใช้บอกสถานะว่าในขณะที่วงจรถูกทำงานอยู่ในฟังก์ชันใด



รูปที่ 2.34 แสดงรูปของสวิตช์รหัสดิจิทัล

2.8 ข้อความสั้น (SMS)

2.8.1 หลักการรับส่ง ข้อความสั้น (SMS) ของโทรศัพท์มือถือ

ข้อความสั้น(SMS) ย่อมาจาก Short Message Service เป็นบริการส่งข้อความสั้นๆจากโทรศัพท์มือถือผ่านทางชุมสายไปยังโทรศัพท์มือถือปลายทาง โดยสามารถส่งได้สูงสุด 160 ตัวอักษรต่อครั้งตามข้อกำหนดมาตรฐานขององค์กร ETSI (European Telecommunications Standards Institute) โดยโหมดของการรับส่งข้อมูลแบบข้อความสั้น (SMS) ผ่าน AT Command ถูกแบ่งเป็น 2 โหมด คือ Text Mode และ PDU Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text Mode เป็นการส่งข้อมูลในรูปแบบตัวอักษรได้โดยตรง โดยการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อน แล้วจึงส่งข้อมูลในรูปแบบ PDU Mode อีกครั้งหนึ่งแต่ในบางเครื่องก็ไม่สนับสนุนการส่งแบบ Text Mode ผ่านทาง AT Command จึงไม่สามารถใช้งานได้สมบูรณ์

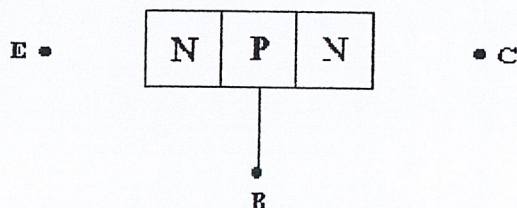
PDU Mode ย่อมาจาก Packet Data Unit เป็นรูปแบบการส่งข้อความสั้น (SMS) อีกรูปแบบหนึ่งที่ต้องมีการเข้ารหัสข้อมูลที่สลับซับซ้อนแต่ตัวเครื่องจะต้องรองรับ AT Command

2.8.2 AT Command

เป็นชุดคำสั่งมาตรฐานที่ใช้ติดต่อสื่อสารกับโทรศัพท์มือถือ โดยส่วนมากจะใช้ในการสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) ในชุดคำสั่งพื้นฐานนั้นบริษัท Hayes ได้เป็นผู้ออกแบบคิดค้นเพื่อใช้กับโมเด็มของตนและต่อมาบริษัทผู้ผลิตมือถือยี่ห้อต่างๆ ได้พัฒนามาใช้กับผลิตภัณฑ์ของตนเป็นเหตุทำให้คำสั่งพิเศษบางคำสั่งไม่เหมือนกันในผลิตภัณฑ์ยี่ห้ออื่น และความสามารถของโทรศัพท์ในบางรุ่นจะไม่รองรับคำสั่งดังกล่าวเนื่องจากไม่ได้มีวงจรส่วนตัวของโมเด็มบรรจุอยู่ใน AT Command มีรูปแบบการส่งข้อมูลแบบข้อมูลลำดับ Serial มักใช้ความเร็วและรูปแบบต่างกันแล้วแต่ยี่ห้อแต่ส่วนมากจะอยู่ที่ความเร็ว 9600 b/s การติดต่อโดยส่งรหัสตัวอักษรที่ขึ้นต้นด้วยคำว่า AT แล้วตามด้วยคำสั่งต่างๆ ซึ่งสามารถทดสอบการเขียนคำสั่งผ่านทางโปรแกรมที่สามารถใช้ติดต่อได้เช่น Hyper Terminal ,STEM

2.8 ทรานซิสเตอร์ชนิด NPN

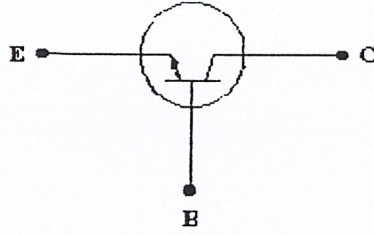
ทรานซิสเตอร์ สร้างมาจากวัสดุประเภทสารกึ่งตัวนำชนิด P และชนิด N มารวมกันโดยทำให้เกิดรอยต่อระหว่างเนื้อสารนี้สองรอยต่อ รอยต่อจากเนื้อสารทั้ง 3 นี้ มีจุดต่อเป็นขาทรานซิสเตอร์ เพื่อใช้เชื่อมโยงหรือบัดกรีกับอุปกรณ์อื่นดังนั้นทรานซิสเตอร์จึงมี 3 ขา มีชื่อเรียกว่า คอลเลกเตอร์ (สัญลักษณ์ C) อิมิตเตอร์ (สัญลักษณ์ E) และ เบส (สัญลักษณ์ B) รูปร่างโครงสร้างและสัญลักษณ์ของทรานซิสเตอร์ดังรูปที่ 2.35 , 2.36



รูปที่ 2.35 โครงสร้างทรานซิสเตอร์ NPN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

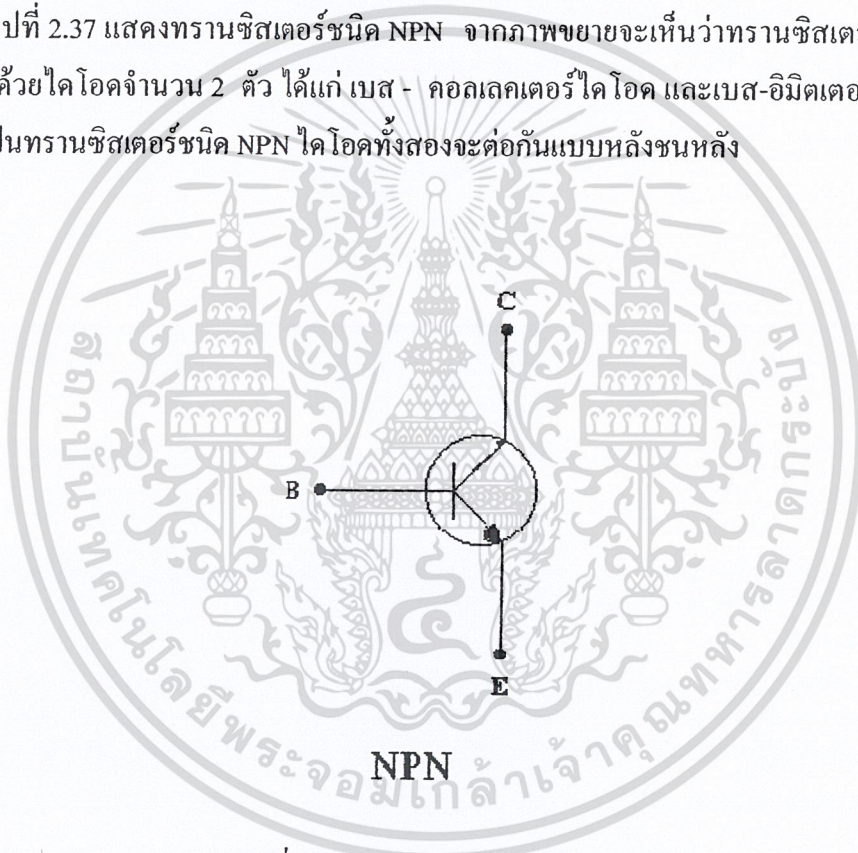
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 ทรานซิสเตอร์ NPN

2.8.1 การทำงานของทรานซิสเตอร์

รูปที่ 2.37 แสดงทรานซิสเตอร์ชนิด NPN จากภาพขยายจะเห็นว่าทรานซิสเตอร์นั้นจะประกอบด้วยไดโอดจำนวน 2 ตัว ได้แก่ เบส - คอลเลคเตอร์ไดโอด และเบส-อิมิตเตอร์ไดโอด โดยเมื่อเป็นทรานซิสเตอร์ชนิด NPN ไดโอดทั้งสองจะต่อกันแบบหลังชนหลัง

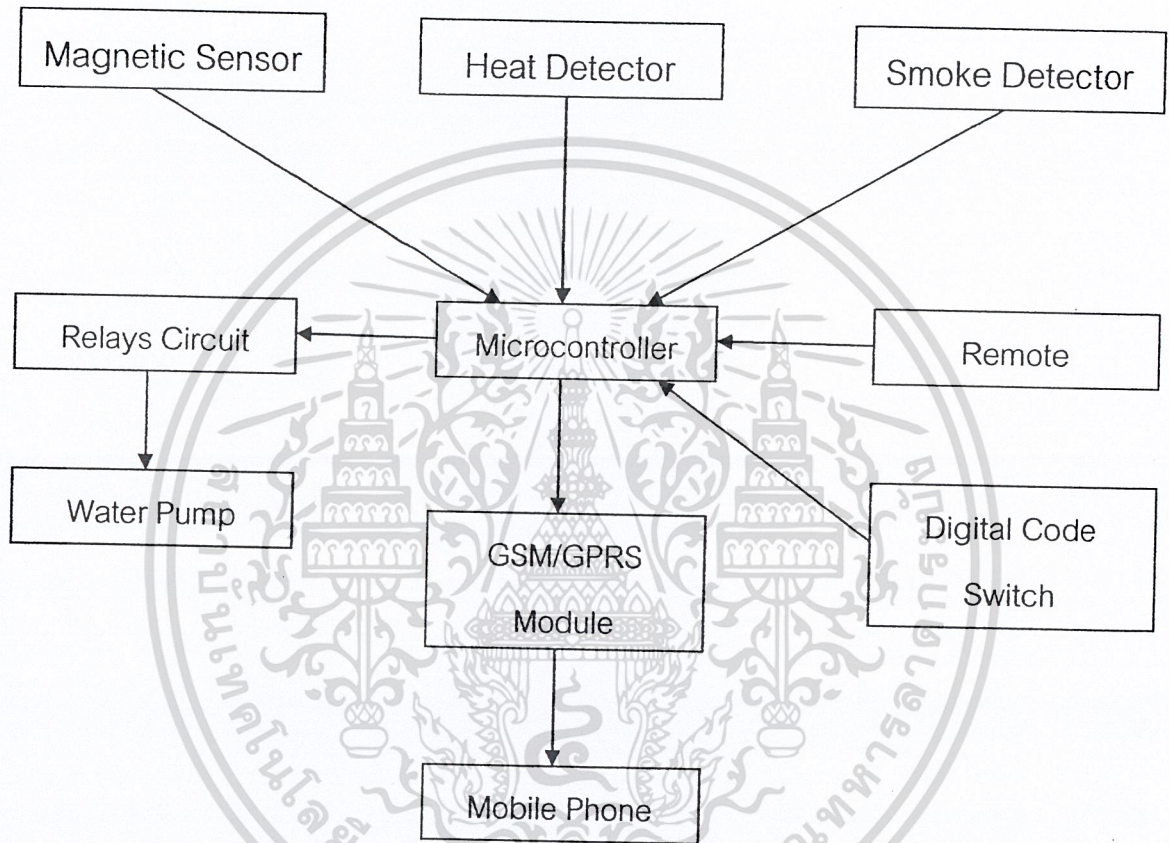


รูปที่ 2.37 แสดงทรานซิสเตอร์ชนิด NPN

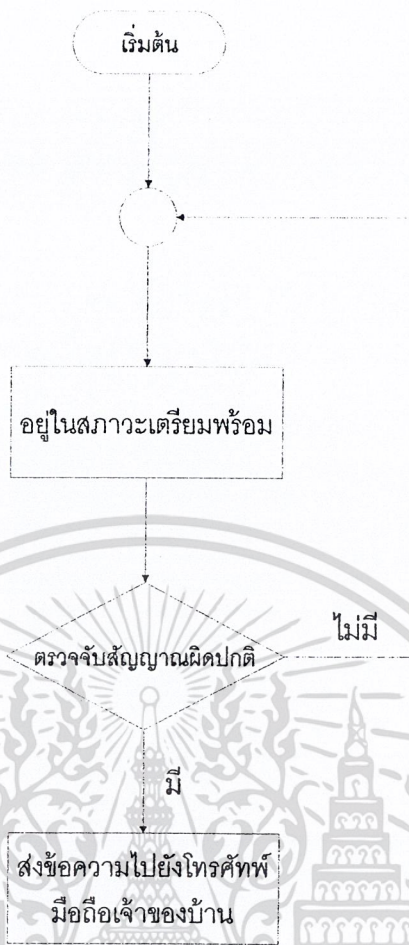
บทที่ 3

การออกแบบโครงงาน

3.1 การออกแบบการทำงานของระบบทั้งหมด



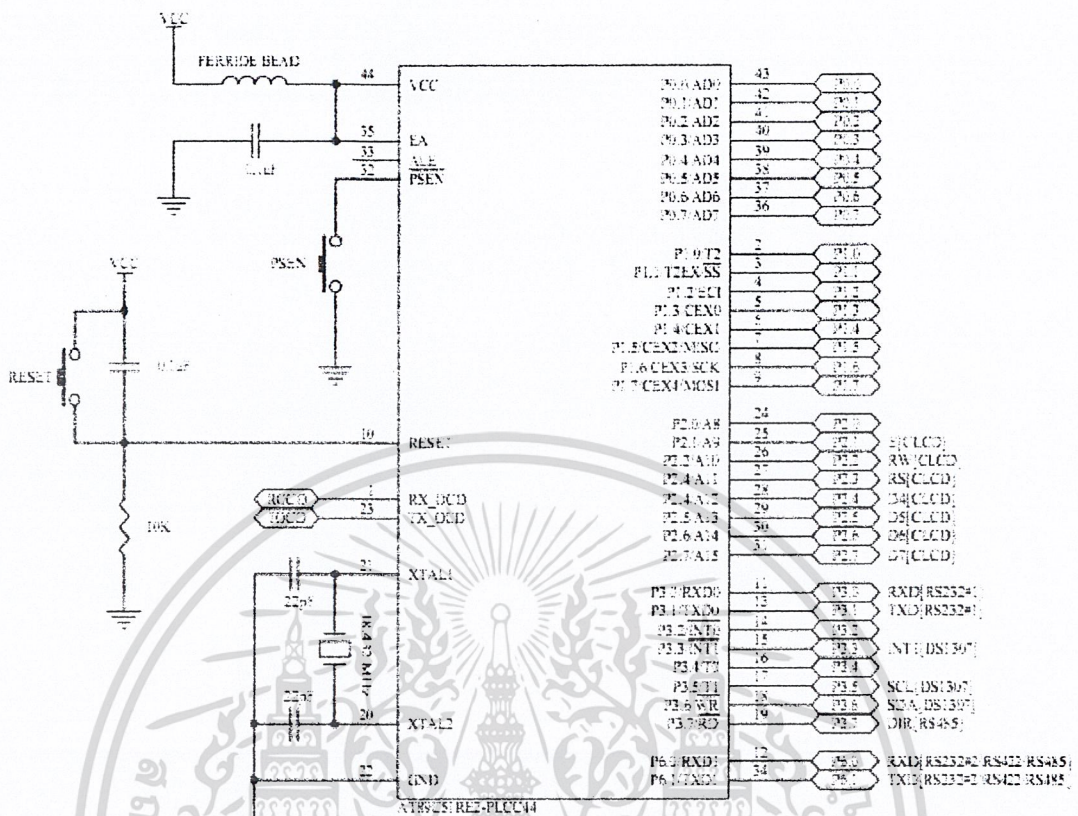
รูปที่ 3.1 แสดงภาพการทำงานของระบบทั้งหมด



รูปที่ 3.2 แสดงแผนผังการทำงานของส่วนส่งข้อความ

จากรูปที่ 3.1 แสดงการทำงานของระบบทั้งหมด สามารถอธิบายการทำงานของส่วนต่างๆ ได้ดังนี้คือ ในส่วนของอุปกรณ์ตรวจจับซึ่งประกอบอุปกรณ์ ตรวจจับควัน แลบแม่เหล็ก และ ตรวจจับความร้อน จะคอยตรวจจับความผิดปกติที่เกิดขึ้น ในกรณีที่เกิดควันหรือความร้อน แล้วอุปกรณ์ตรวจจับควัน และตรวจจับความร้อนทำงาน หรือ มีผู้ไม่หวังดีแอบบุกรุกเข้ามาในบ้าน แล้วแลบแม่เหล็กมีความผิดปกติ จากนั้นอุปกรณ์จะทำการส่งสัญญาณผ่านต่อไปยังชุดควบคุมด้วยไมโครคอนโทรลเลอร์แล้วส่งการให้ GSM/GPRS Module เพื่อทำการส่งข้อความแจ้งเตือนไปยัง โทรศัพท์มือถือของเจ้าของบ้านเพื่อให้ทราบ ดังรูปที่ 3.2 และ เนื่องจากไมโครคอนโทรลเลอร์ เป็นไอซีประเภท TTL สามารถทำงานได้ที่ความถี่สูง แต่จะกินกระแสมากและทำงานได้ที่แรงดัน 5 V เท่านั้น จึงจำเป็นต้องออกแบบวงจร ให้ค่าแรงดัน output จากเซนเซอร์ที่จะส่งต่อเข้าไปยังไมโครคอนโทรลเลอร์ เป็นแรงดันในระดับของ TTL

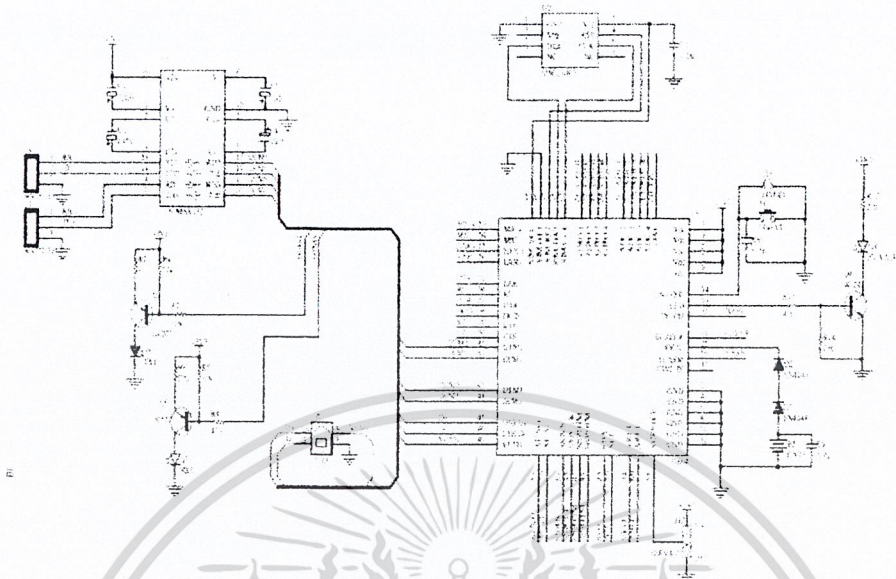
3.2 การทำงานของบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.3 แสดงลักษณะ โครงสร้างของบอร์ดไมโครคอนโทรลเลอร์

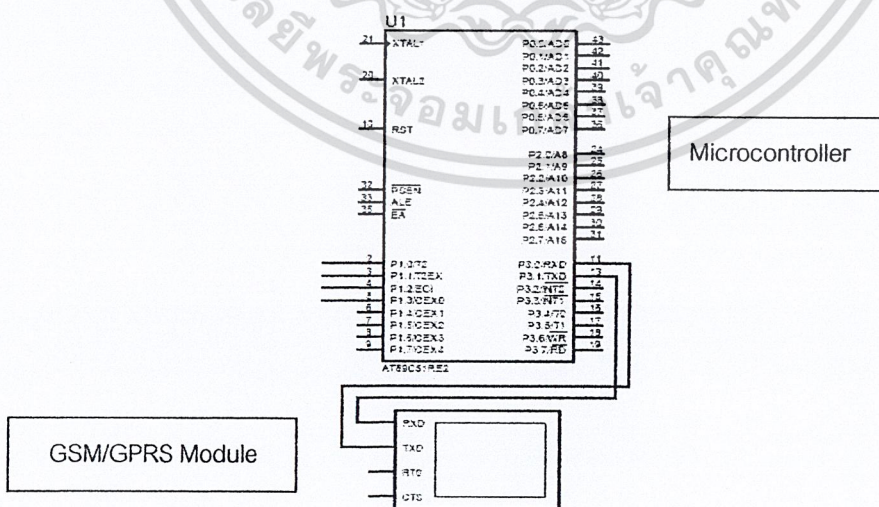
จากรูปที่ 3.3 การทำงานของไมโครคอนโทรลเลอร์นั้นจะใช้ในการรับ output จากตัวอุปกรณ์ตรวจจับความความร้อน และอุปกรณ์ตรวจจับวัน ทำงานไมโครคอนโทรลเลอร์ จะสั่งให้ปั้มน้ำทำงานด้วย และสั่งให้ GMS/GPRS Module ส่งข้อความแจ้งเตือนไปยังโทรศัพท์มือถือ ถ้าแถบแม่เหล็กหลุดออกจกกันจะสั่งให้ไมโครคอนโทรลเลอร์ไปสั่งให้ GMS/GPRS Module ทำการส่งข้อความแจ้งเตือนไปยังโทรศัพท์ มือถือ ส่วนในกรณีที่ รีโมท และสวิตช์ห้สติจิตอล บอร์ดไมโครคอนโทรลเลอร์จะรับ Input เข้ามาแล้วทำการสั่งให้ระบบทั้งหมดหยุดการทำงานทำให้สามารถเข้าออกประตูได้

3.3 การทำงานของ GSM/GPRS Module



รูปที่ 3.4 โครงสร้างของบอร์ด GSM/GPRS Module

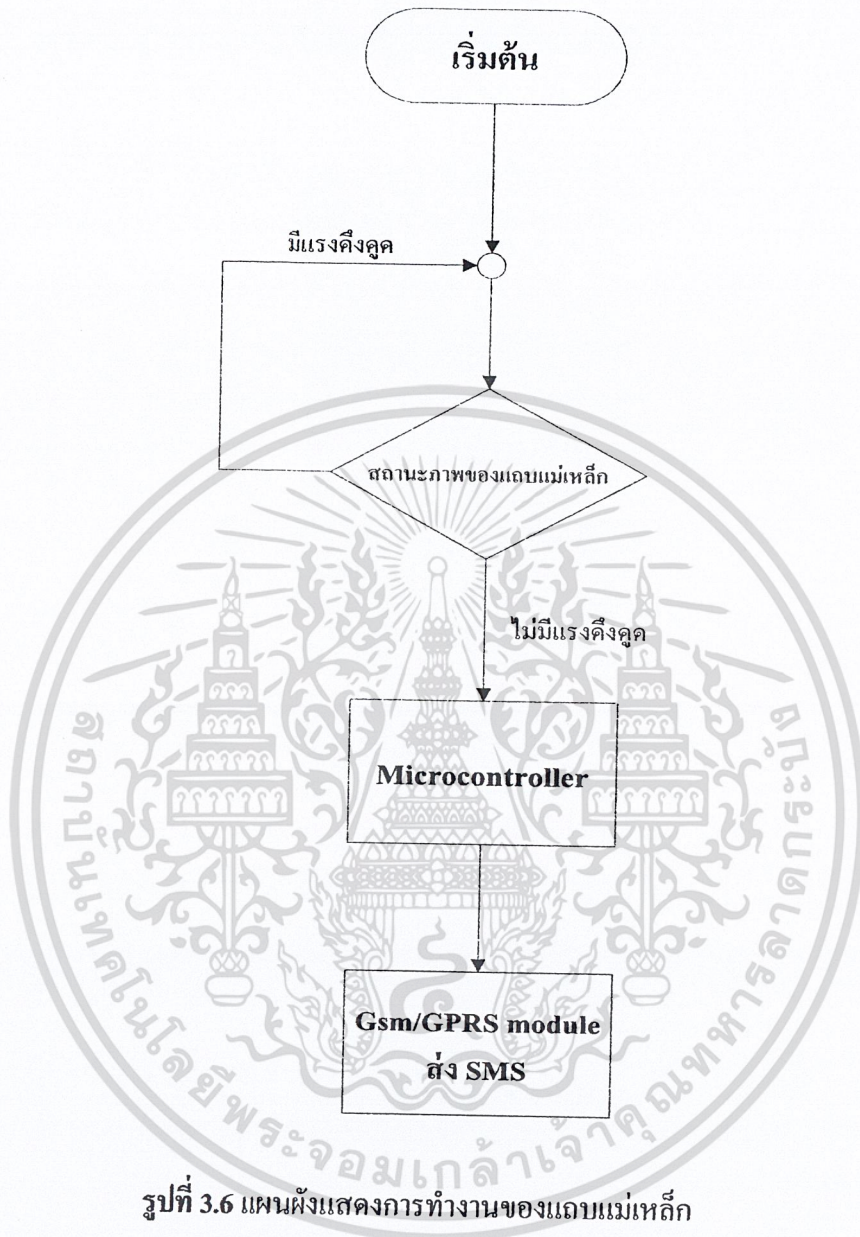
จากรูปที่ 3.4 การทำงานของ GSM/GPRS Module จะใช้ในส่วนของการส่งข้อความแจ้งเตือนไปยังโทรศัพท์ โดยเชื่อมต่อกับไมโครคอนโทรลเลอร์ทางพอร์ตอนุกรม RS232#1 ที่ขา Rx, Tx ซึ่ง GSM/GPRS Module จะใช้ไฟเลี้ยงจากไมโครคอนโทรลเลอร์ และ GSM/GPRS Module จะใช้คำสั่ง AT Commands ในการควบคุม Module



รูปที่ 3.5 GSM/GPRS Module ต่อกับไมโครคอนโทรลเลอร์

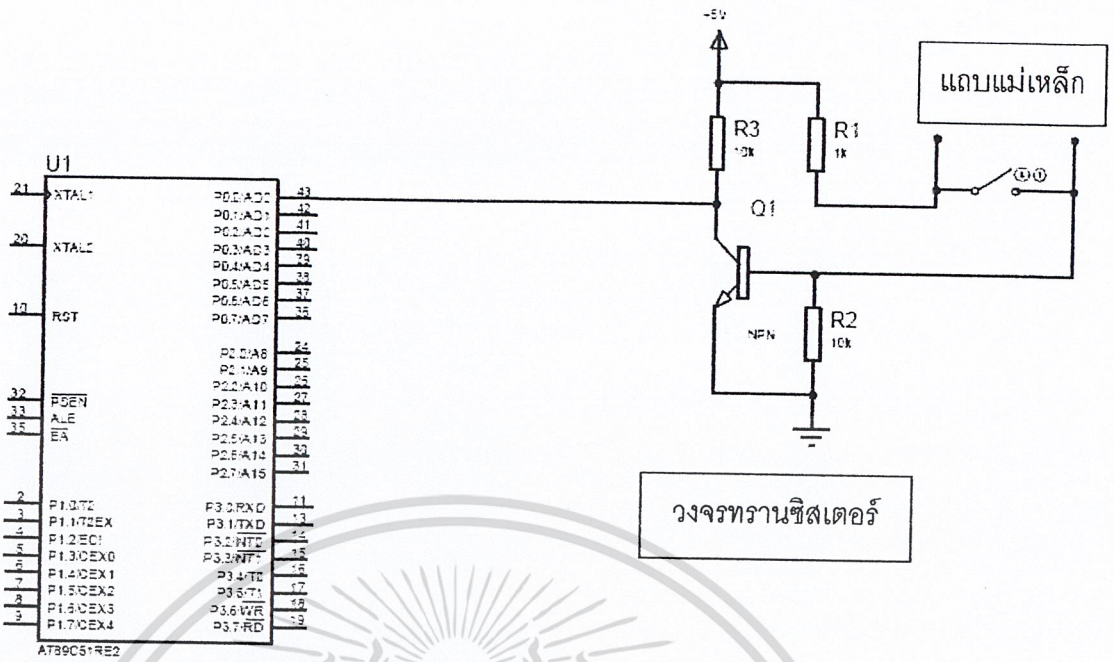
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบวงจรอุปกรณ์แอมป์หลัก



รูปที่ 3.6 แผนผังแสดงการทำงานของแอมป์หลัก

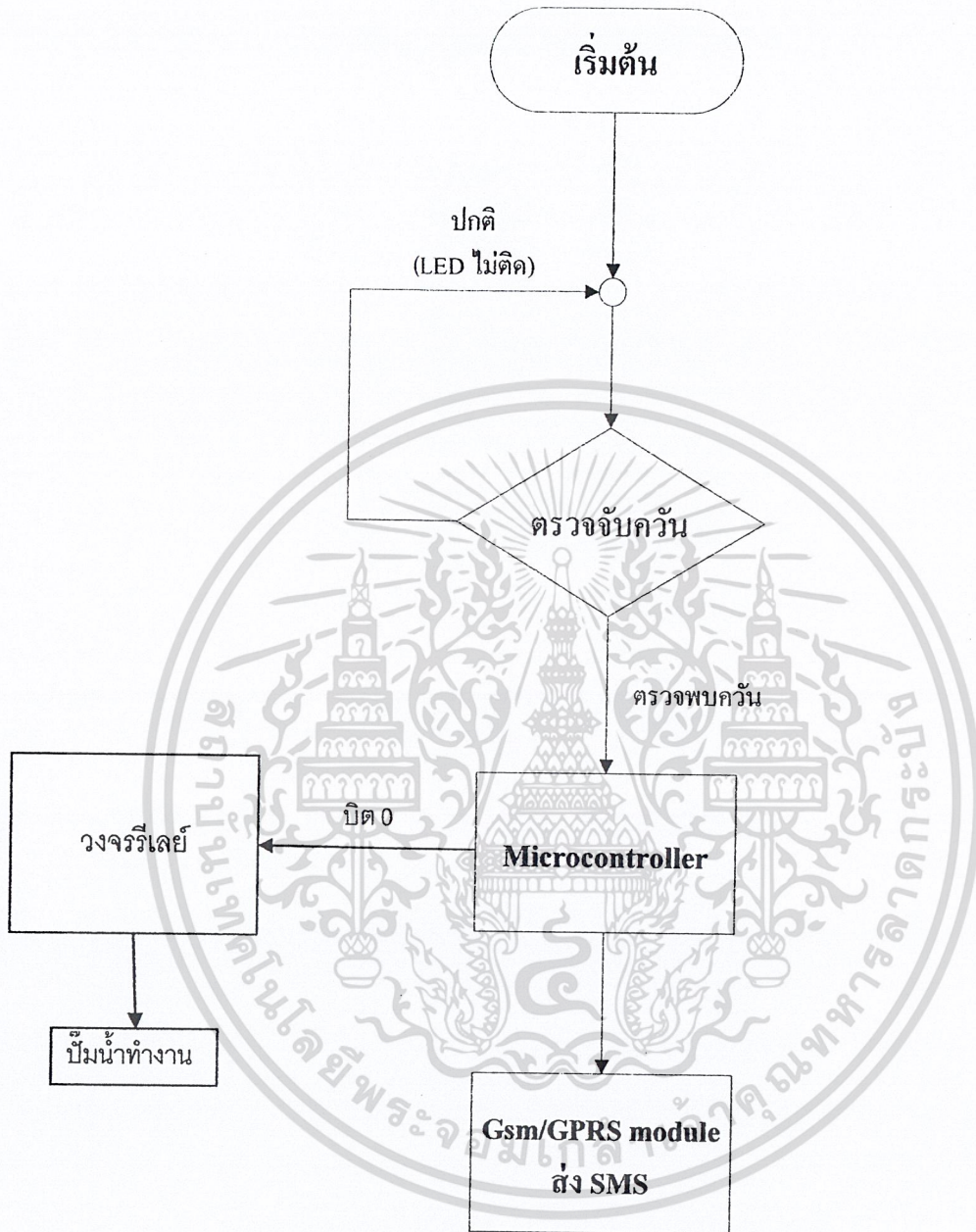
จากรูปที่ 3.6 การทำงานของอุปกรณ์แอมป์หลักจะเป็นเสมือนสวิตช์ทั่วไป เมื่อทำการจ่ายไฟเข้าไปในวงจร ถ้าแอมป์หลักมีแรงดึงดูดซึ่งกันกระแสไฟฟ้าก็จะไหลครบวงจรทำให้ขา Port มีสถานะเป็น Low แต่ถ้าแอมป์หลักอยู่ห่างกันจนไม่มีแรงดึงดูดซึ่งกันแล้ว จะทำให้กระแสไฟฟ้าไหลไม่ครบวงจร ทำให้ขา Port มีสถานะเป็น High ดังนั้นตัวไมโครคอนโทรลเลอร์ จะสั่งให้ GSM/GPRS Module ส่งข้อความแจ้งเตือนไปยังโทรศัพท์มือถือ



รูปที่ 3.7 แสดงวงจรอุปกรณ์แถบแม่เหล็กต่อกับไมโครคอนโทรลเลอร์

อุปกรณ์แถบแม่เหล็กจะต่อผ่านวงจรถานซิสเตอร์ก่อนที่จะต่อเข้ากับไมโครคอนโทรลเลอร์เพื่อควบคุมการไหลของกระแสไฟที่ไหลจากไมโครคอนโทรลเลอร์โดยจะต่อกับไมโครคอนโทรลเลอร์ทาง Port ที่ 0

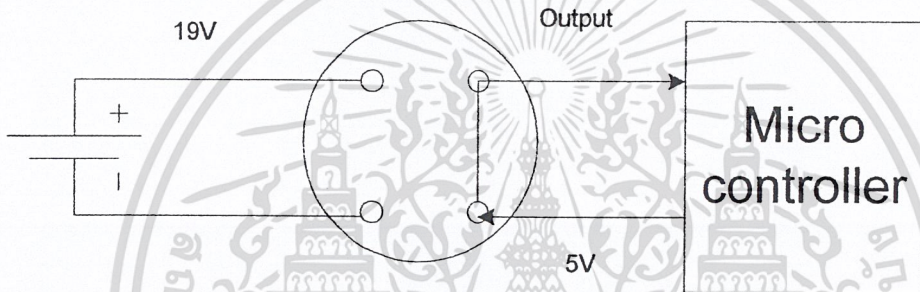
3.5 การออกแบบวงจรอุปกรณ์ตรวจจับควัน



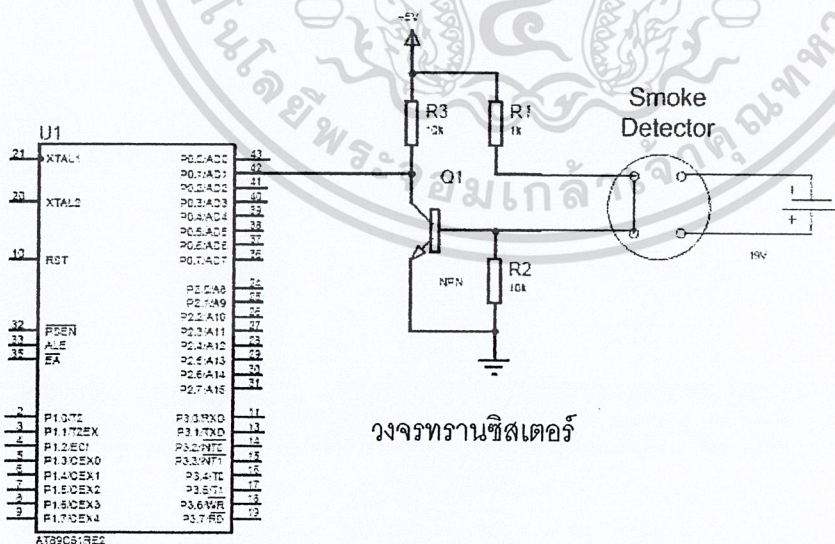
รูปที่ 3.8 แผนผังแสดงการทำงานของอุปกรณ์ตรวจจับควัน

จากรูปที่ 3.8 การทำงานของอุปกรณ์ตรวจจับควัน ถ้าอุปกรณ์อยู่ในสถานะที่ยังไม่มีการตรวจจับควันการแสงจะไม่ไหลออกมาทางขา Output เมื่ออุปกรณ์สามารถตรวจจับควันได้ กระแส +5V จะไหลออกมาทางขา Output ทำให้ ไมโครคอนโทรลเลอร์ สั่งงานให้ GSM/GPRS Module ส่งข้อความสั้น (SMS) ไปยังโทรศัพท์มือถือ และ จะส่งบิต 1 ให้วงจรรีเลย์ทำงาน ทำให้ปั้มน้ำทำงาน โดยอุปกรณ์ตรวจจับควันจะต่อเข้ากับวงจรถรานซิสเตอร์และผ่านเข้าไมโครคอนโทรลเลอร์ทาง Port 0

Smoke Detector

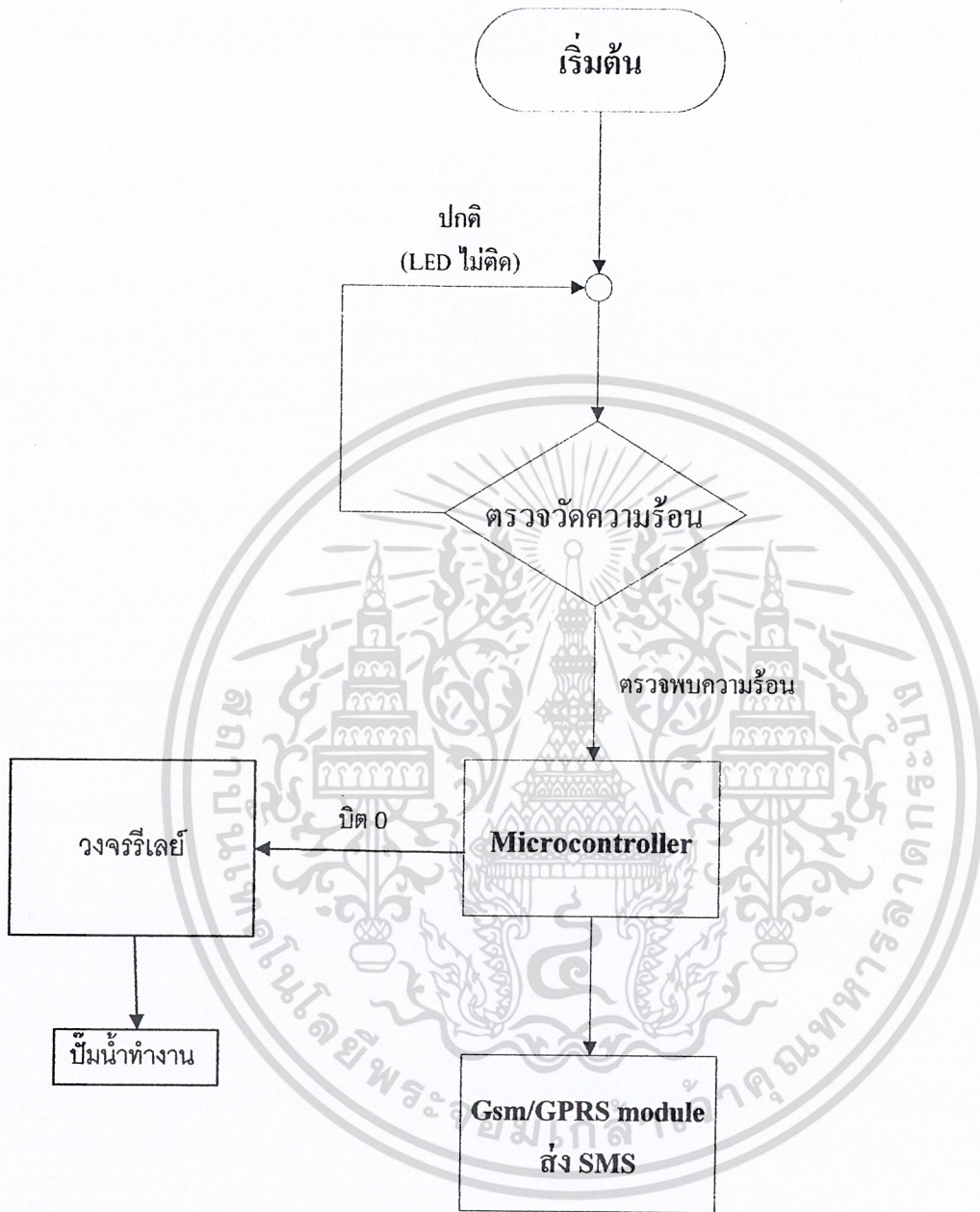


รูปที่ 3.9 แสดงวงจรอุปกรณ์ตรวจจับควัน



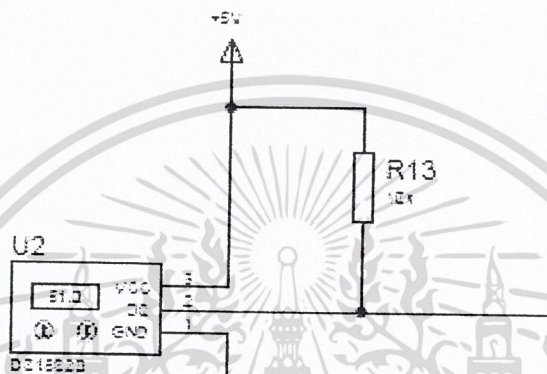
รูปที่ 3.10 แสดงวงจรอุปกรณ์ตรวจจับควันต่อเข้ากับ ไมโครคอนโทรลเลอร์

3.6 การออกแบบวงจรอุปกรณ์ตรวจจับความร้อน

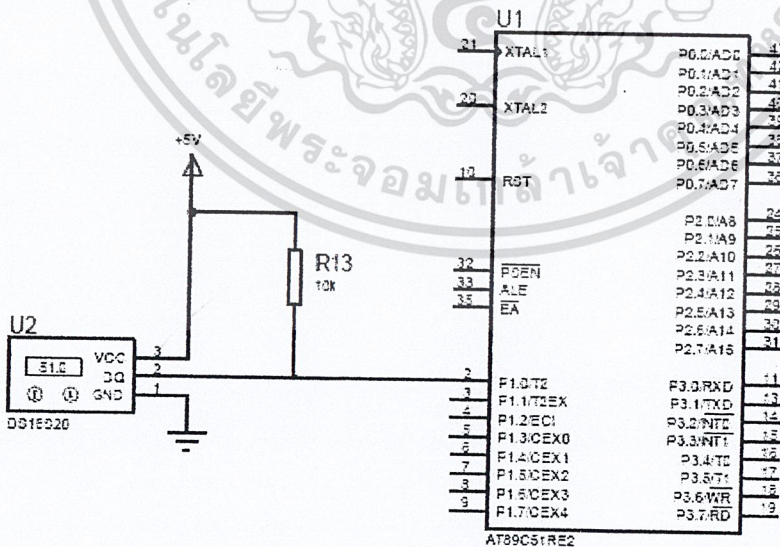


รูปที่ 3.11 แผนผังแสดงการทำงานของอุปกรณ์ตรวจจับความร้อน

จากรูปที่ 3.11 การทำงานของอุปกรณ์ตรวจจับความร้อนที่ใช้เป็นชนิดจับอัตราการเพิ่มของอุณหภูมิ โดยในการทดลองจะตั้งอุณหภูมิไว้ที่ 70 องศาเซลเซียส หากอุปกรณ์ตรวจจับความร้อนวัดอุณหภูมิได้เกินที่เราตั้งไว้ ก็จะทำการส่ง Output ให้กับไมโครคอนโทรลเลอร์ สั่งงานให้ GSM/GPRS Module ส่ง ข้อความสั้น (SMS) ไปยังโทรศัพท์มือถือ และ จะส่งบิต 1 ให้วงจรรีเลย์ทำงาน ทำให้ปั๊มน้ำทำงานโดยอุปกรณ์ตรวจจับความร้อนจะต่อเข้ากับตัวต้านทานขนาด 10 กิโลโอห์มและผ่านเข้าไมโครคอนโทรลเลอร์ทาง Port 1



รูปที่ 3.12 แสดงวงจรอุปกรณ์ตรวจจับความร้อน

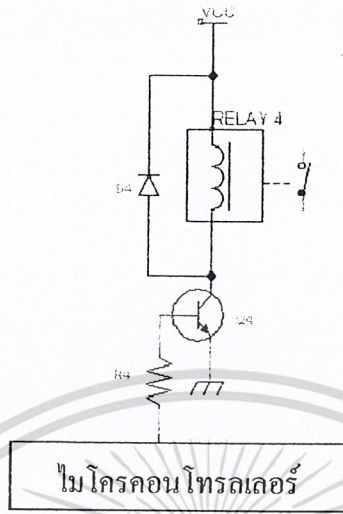


รูปที่ 3.13 แสดงวงจรอุปกรณ์ตรวจจับความร้อนต่อกับ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

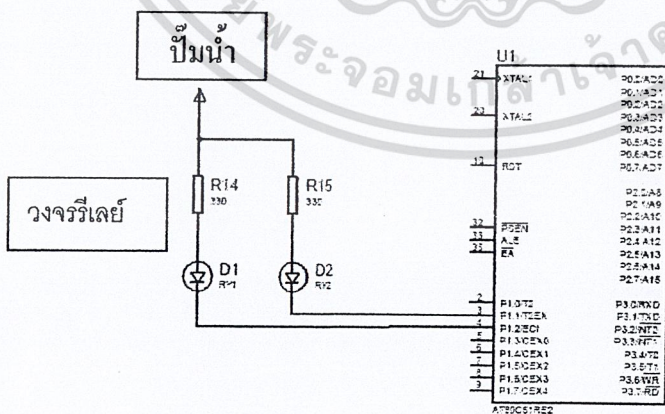
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การออกแบบวงจรรีเลย์



รูปที่ 3.14 แสดงวงจรรีเลย์

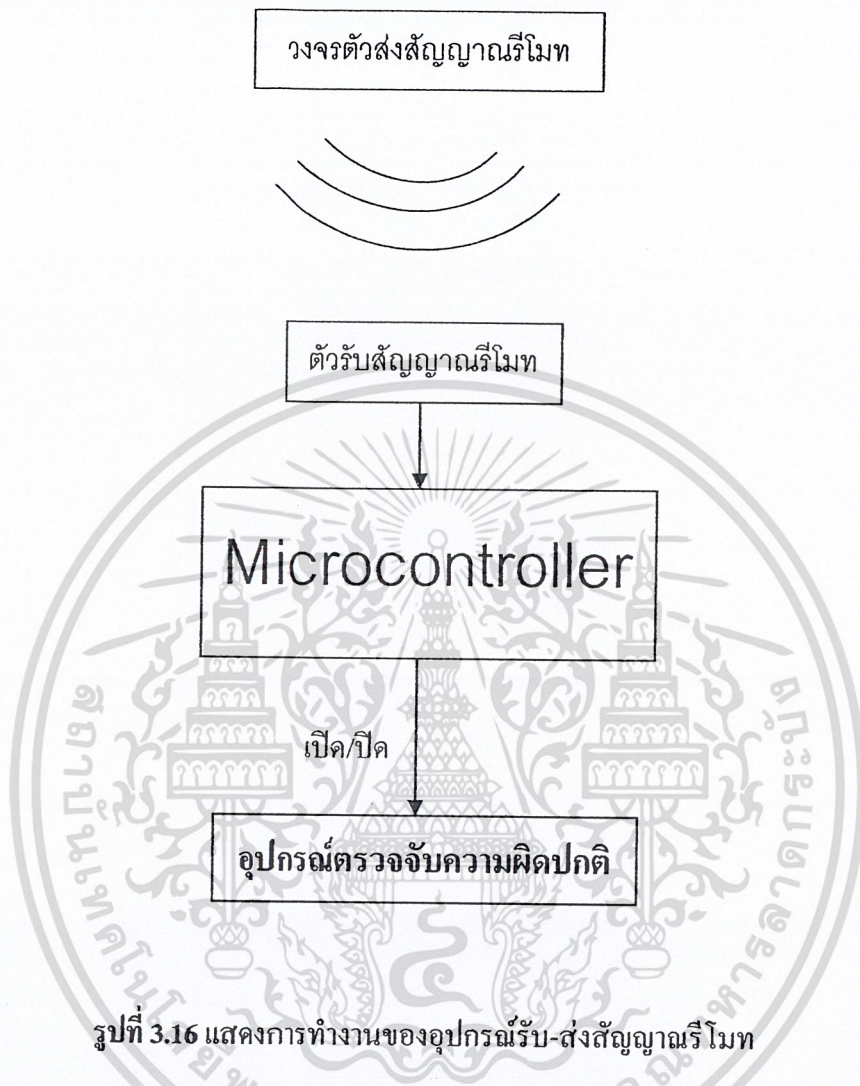
วงจรรีเลย์เป็นวงจรเพื่อใช้ควบคุม มีหน้าที่เป็นสวิตช์การเปิดปิดของปั้มน้ำ โดยเมื่ออุปกรณ์ตรวจจับความผิดปกติทำงาน จะสั่งงานให้ไมโครคอนโทรลเลอร์ส่งบิต 1 ไปยังวงจรรีเลย์ ทำให้วงจรรีเลย์เปิดให้ปั้มน้ำทำงาน แต่หากอุปกรณ์ตรวจจับไม่สามารถจับความผิดปกติได้ วงจรรีเลย์ก็จะไม่ทำงาน ปั้มน้ำจึงไม่ทำงาน โดยวงจรรีเลย์จะต่อเข้ากับ Port 1 ของไมโครคอนโทรลเลอร์



รูปที่ 3.15 แสดงวงจรรีเลย์ต่อกับไมโครคอนโทรลเลอร์

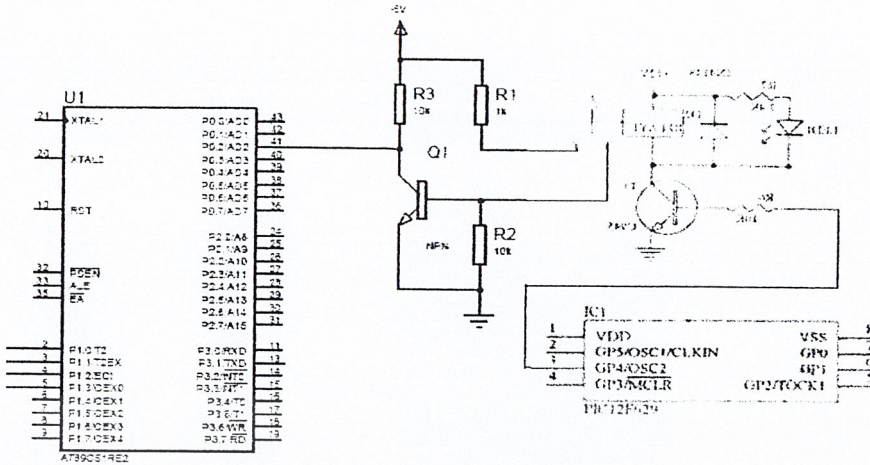
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบวงจรอุปกรณ์รับ-ส่งสัญญาณรีโมท



รูปที่ 3.16 แสดงการทำงานของอุปกรณ์รับ-ส่งสัญญาณรีโมท

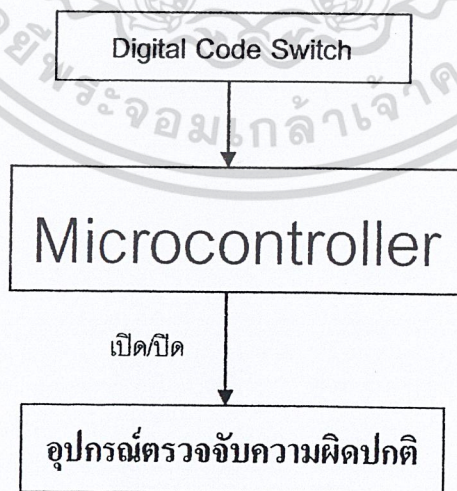
อุปกรณ์รับ-ส่งสัญญาณรีโมท ใช้เปิด-ปิดระบบ เพื่อให้เจ้าของบ้านสามารถเข้าออกจากบ้านได้อย่างสะดวก โดยอุปกรณ์รับส่งสัญญาณรีโมททำงานโดยใช้คลื่นอินฟราเรด โดยตัวรับสัญญาณจะต่อเข้ากับไมโครคอนโทรลเลอร์เมื่อกรีโมทและตัวรับสัญญาณรีโมทจับสัญญาณได้ ไมโครคอนโทรลเลอร์ก็จะสั่งเปิด/ปิดการทำงานของอุปกรณ์ตรวจจับความผิดปกติ



รูปที่ 3.19 แสดงวงจรอุปกรณ์รับสัญญาณรีโมทต่อกับไมโครคอนโทรลเลอร์

อุปกรณ์รับ-ส่งสัญญาณรีโมทจะต่อเข้ากับวงจรทรานซิสเตอร์ก่อนแล้วจึงต่อกับไมโครคอนโทรลเลอร์ทาง Port 0

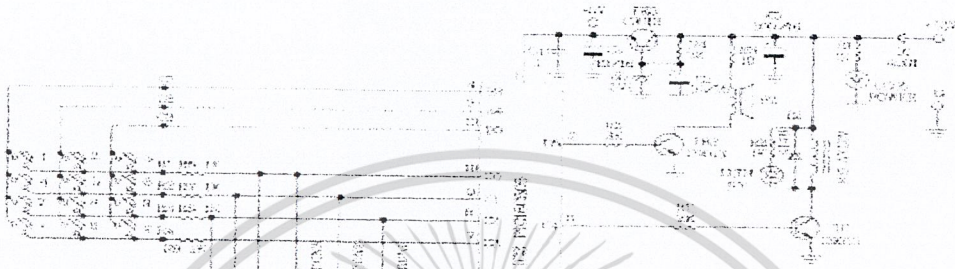
3.9 การออกแบบวงจรอุปกรณ์ Digital Code Switch



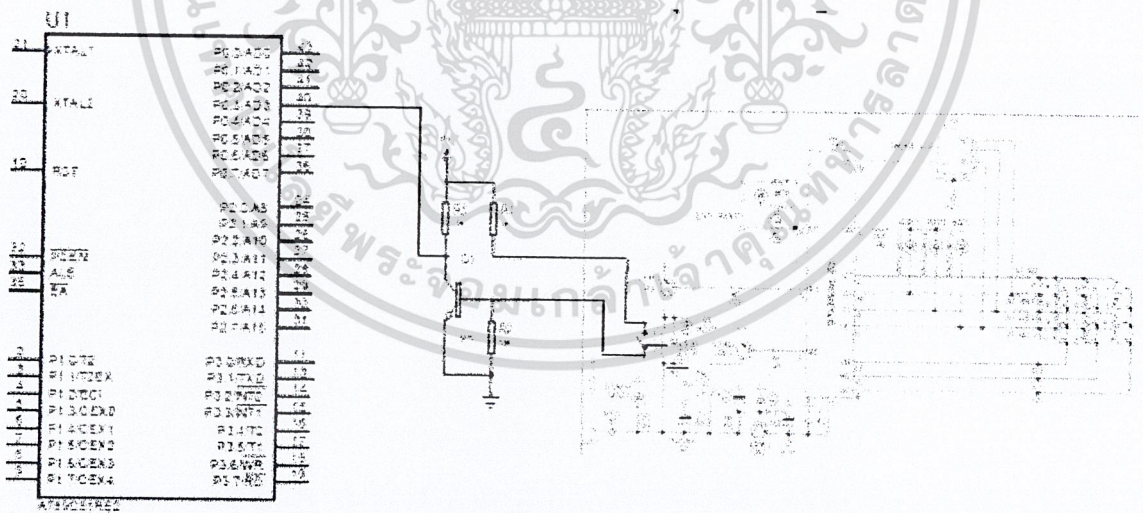
รูปที่ 3.20 แสดงแผนผังการทำงานของ Digital Code Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ Digital Code Switch ใช้เปิด-ปิดระบบ เพื่อให้เจ้าของบ้านสามารถเข้าออกจากบ้านได้อย่างสะดวก โดยอุปกรณ์ Digital Code Switch ทำงานโดยการครหัส โดยรหัสนั้นจะเป็นรหัส 4 ตัวตามที่เราได้กำหนดไว้ โดยเมื่อเรากรหัสผ่านถูกต้องตามที่เราได้ตั้งไว้ ไมโครคอนโทรลเลอร์ก็จะสั่งเปิด/ปิดการทำงานของอุปกรณ์ตรวจจับความผิดปกติ โดยอุปกรณ์ Digital Code Switch จะต่อเข้ากับวงจรทรานซิสเตอร์และต่อกับไมโครคอนโทรลเลอร์



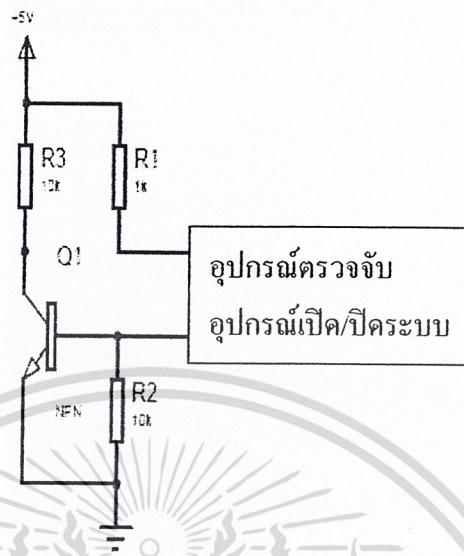
รูปที่ 3.21 แสดงวงจรของ Digital Code Switch



รูปที่ 3.22 แสดงวงจรของ Digital Code Switch ต่อกับไมโครคอนโทรลเลอร์

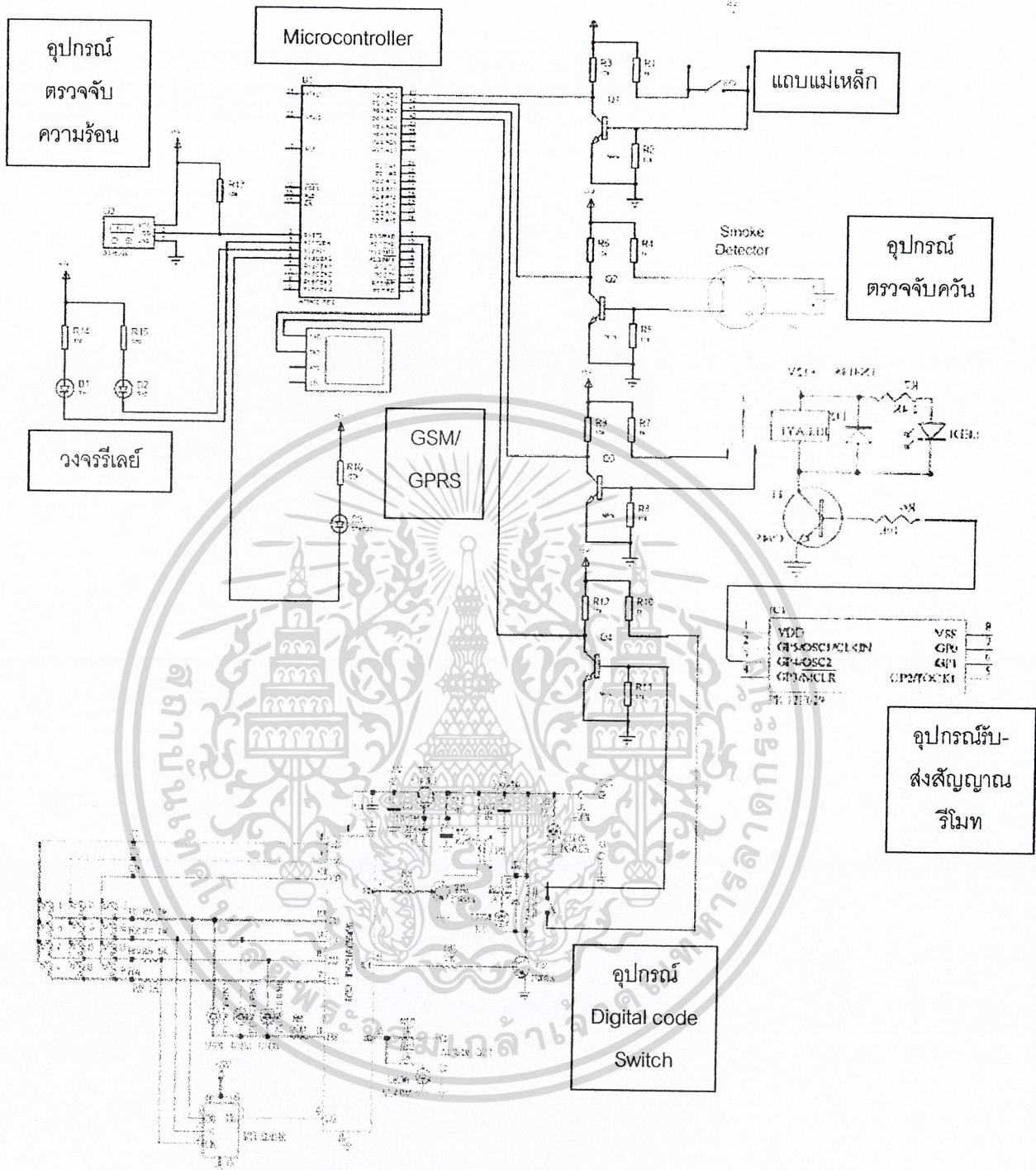
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การออกแบบวงจรทรานซิสเตอร์



รูปที่ 3.23 แสดงวงจรทรานซิสเตอร์

วงจรทรานซิสเตอร์ถูกออกแบบเพื่อเป็นตัวทำให้กระแสที่ไหลจาก Port 0 เพื่อควบคุมการไหลของกระแสให้คงที่ หากเราต่ออุปกรณ์ต่อเข้ากับไมโครคอนโทรลเลอร์โดยตรงอาจจะทำให้กระแสไหลไม่สม่ำเสมอ ซึ่งอาจเกิดจากสิ่งรบกวนภายนอก โดยอุปกรณ์ที่เป็น Input ทุกตัวจะต่อเข้ากับวงจรทรานซิสเตอร์ 1 ตัวต่อ 1 วงจร แล้วจึงต่อผ่านเข้าบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.24 วงจรรวมของระบบ

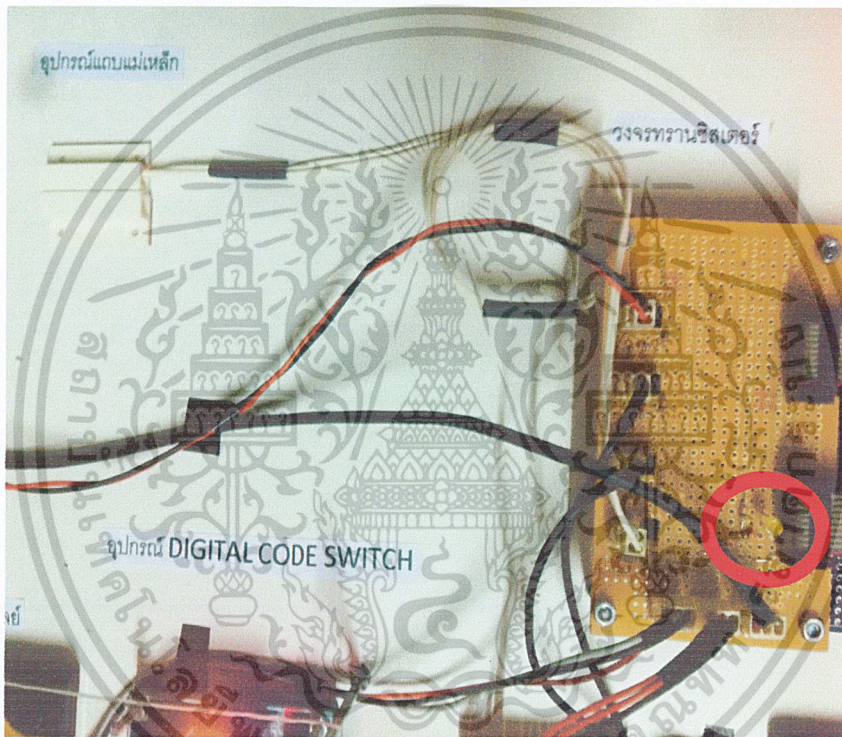
จากรูปที่ 3.24 เป็นการต่อวงจรรวมกันระหว่าง GSM/GPRS Module อุปกรณ์ตรวจจับความร้อน อุปกรณ์ตรวจจับควัน อุปกรณ์แม่เหล็ก อุปกรณ์รับ-ส่งสัญญาณรีโมท อุปกรณ์ Digital Code Switch วงจรรีเลย์และวงจรทรานซิสเตอร์เข้ากับ ไมโครคอนโทรลเลอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

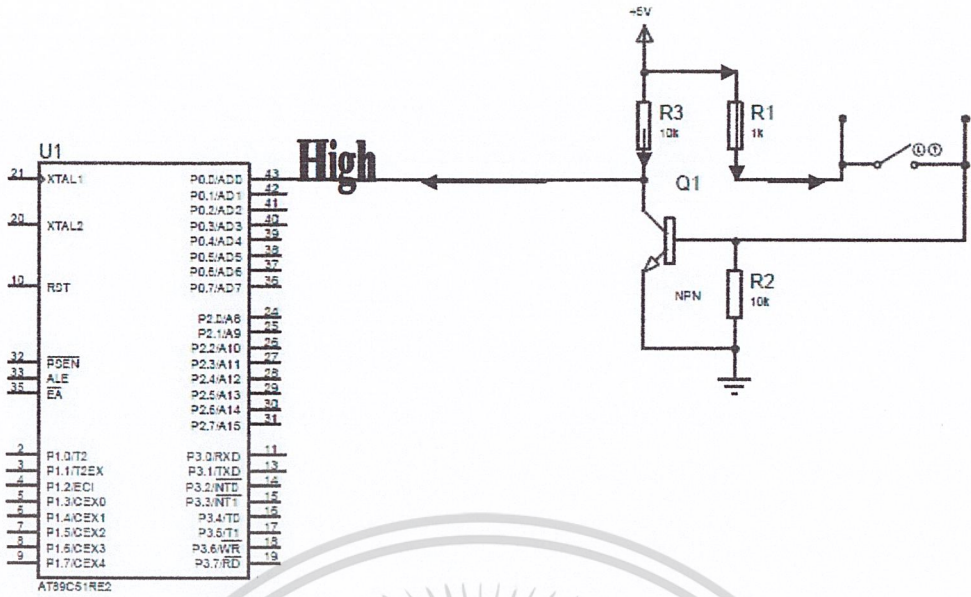
การทดลองการทำงานของระบบให้สอดคล้องกับเหตุการณ์ที่คาดว่าจะเกิดขึ้นตามที
ออกแบบไว้ จึงได้แบ่งการทดลองออกเป็นส่วนๆตามอุปกรณ์ ดังนี้

4.1 การทดลองของอุปกรณ์แถบแม่เหล็ก

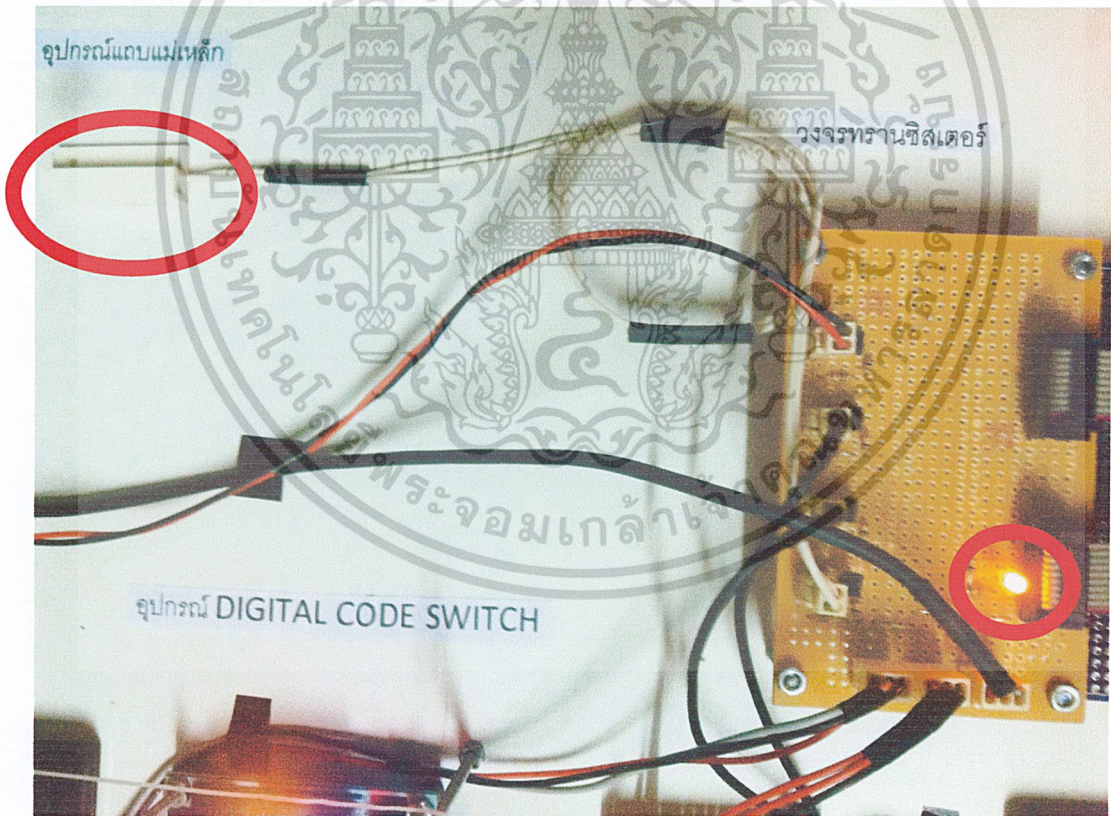


รูปที่ 4.1 แสดงอุปกรณ์แถบแม่เหล็กในขณะที่มีแรงดึงดูด

ในขณะที่แถบแม่เหล็กมีแรงดึงดูดกันนั้น กระแสไฟจะไหลครบวงจร หลอดไฟ LED
บอกสถานะของการทำงานจะดับ ตามรูปที่ 4.1

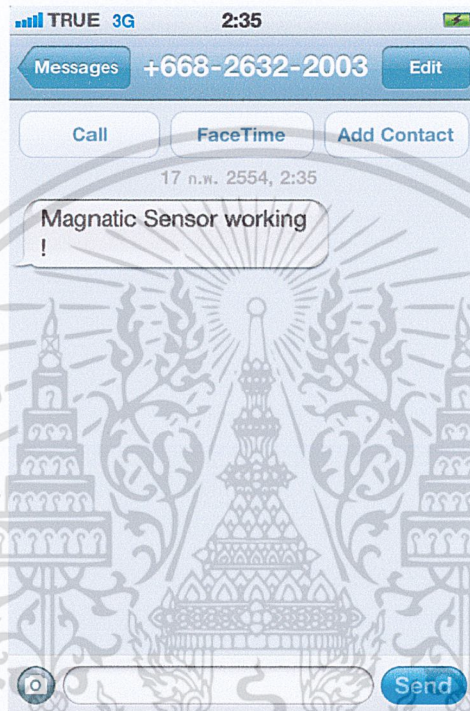


รูปที่ 4.2 แสดงการไหลของกระแสของอุปกรณ์แถบแม่เหล็กในขณะที่ไม่มีแรงดึงดูด



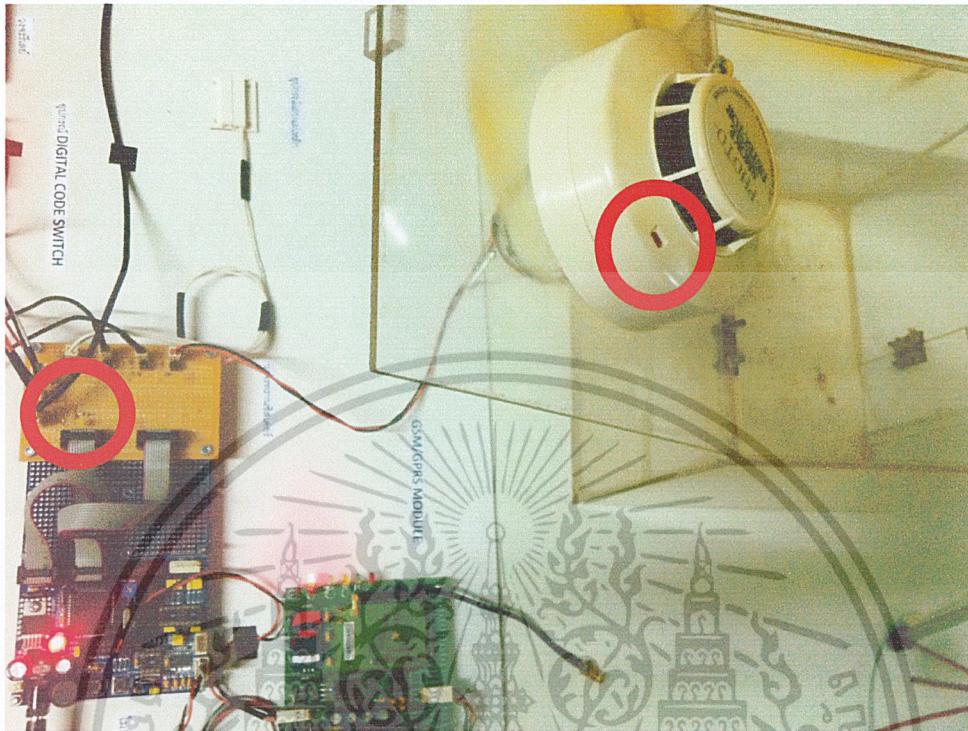
รูปที่ 4.3 แสดงอุปกรณ์แถบแม่เหล็กในขณะที่ไม่มีแรงดึงดูด

เมื่อแถบแม่เหล็กหลุดออกจากกัน กระแสไฟฟ้าจะไหลตามวงจร ดังรูปที่ 4.1 ซึ่งทำให้กระแสไฟฟ้าไหลตกคร่อมตัวต้านทาน และไม่สามารถไหลไปไหนได้ เป็นผลให้กระแสที่ขาที่ 3 Port0 มีสถานะเป็น High แล้วไมโครคอนโทรลเลอร์จะทำการสั่งให้ GSM/GPRS Module ทำการส่งข้อความสั้น (SMS) เข้าโทรศัพท์มือถือ ดังรูปที่ 4.4 และไฟบอกสถานะของการทำงาน จะติด ดังรูปที่ 4.3



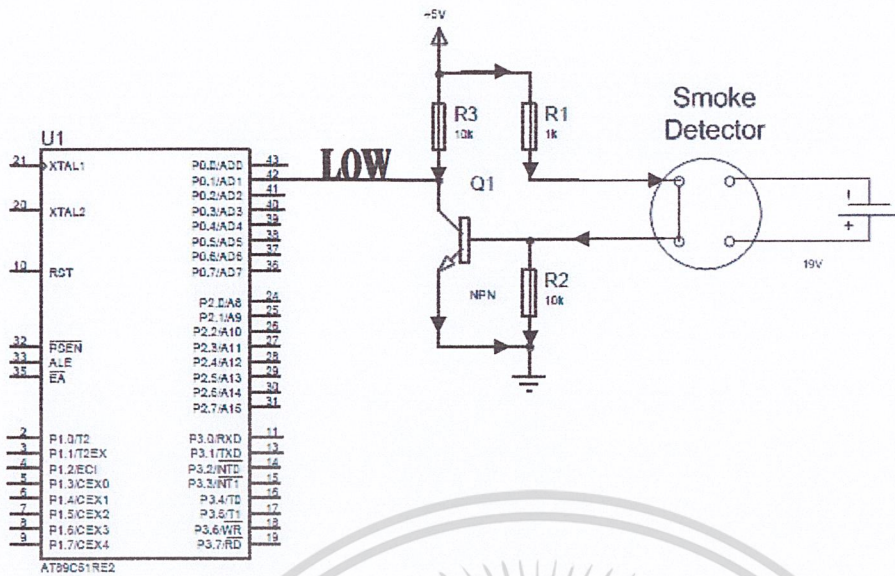
รูปที่ 4.4 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์แถบแม่เหล็ก

4.2 การทดลองของอุปกรณ์ตรวจจับควัน

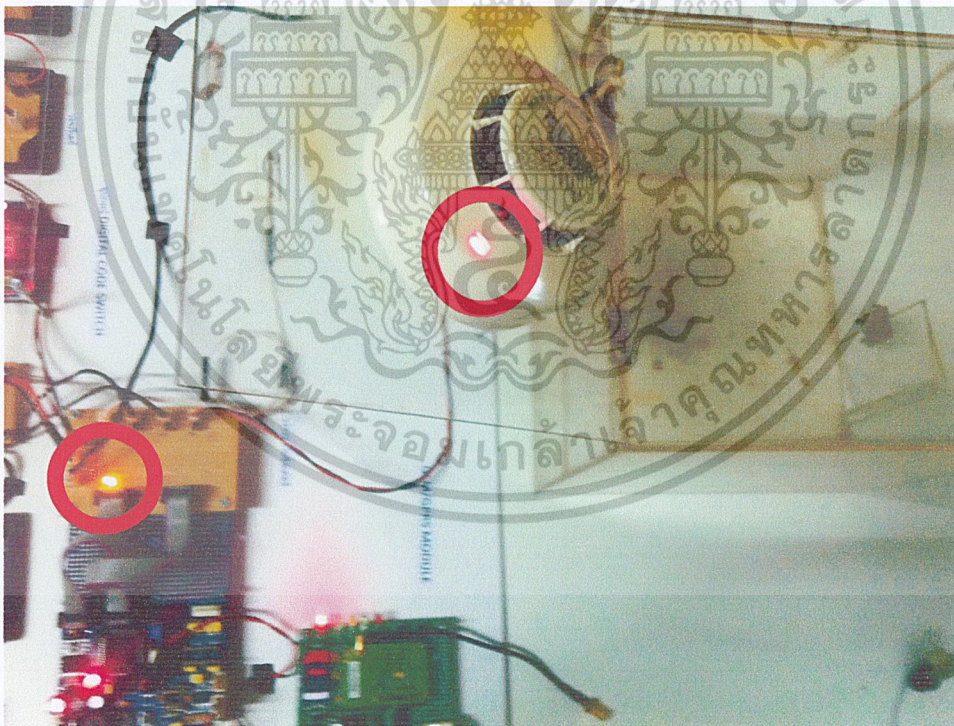


รูปที่ 4.5 แสดงอุปกรณ์ตรวจจับควันในขณะที่ยังไม่ทำงาน

ในขณะที่อุปกรณ์ตรวจจับควันนั้น ยังไม่สามารถตรวจจับสิ่งผิดปกติได้ ไฟบอกสถานะของอุปกรณ์และไฟสถานะของระบบจะดับ ดังรูปที่ 4.5 และระบบจะไม่ทำการ ส่งข้อความสั้น (SMS) และปั๊มน้ำก็จะไม่ทำงานด้วยเช่นกัน



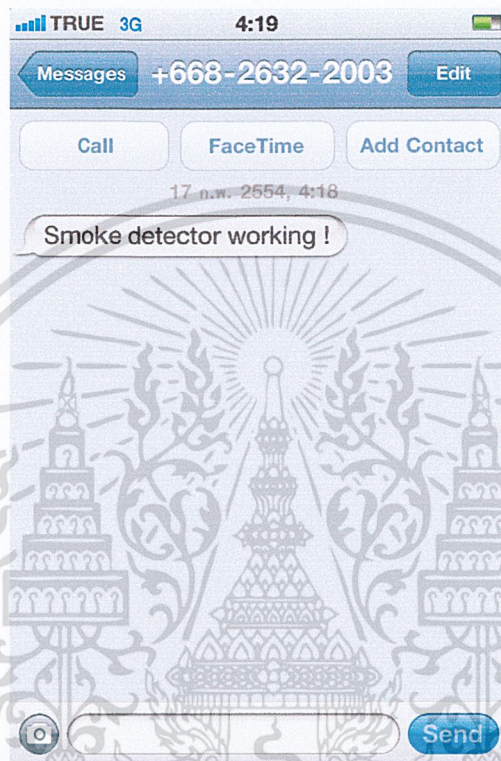
รูปที่ 4.6 แสดงการไหลของกระแสของอุปกรณ์ตรวจจับควันในขณะที่ทำงาน



รูปที่ 4.7 แสดงอุปกรณ์ตรวจจับควันในขณะที่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่ออุปกรณ์ตรวจจับควันทำงานจะทำให้กระแสไหล ดังรูปที่ 4.6 ทำให้กระแสทั้งหมด ไหลลง Ground เป็นผลให้ขาที่ 0 Port0 มีสถานะเป็น Low ไฟบอสถานะของการทำงานจะ แสดงดังรูปที่ 4.7 แล้วไมโครคอนโทรลเลอร์จะทำการสั่งให้ GSM/GPRS Module ทำการส่ง ข้อความสั้น (SMS) เข้าโทรศัพท์มือถือ และสั่งให้ ขาที่ 1 Port1 มีสถานะเป็น High เป็นผลให้ปั้มน้ำทำงานด้วย ดังรูปที่ 4.8 และ 4.9



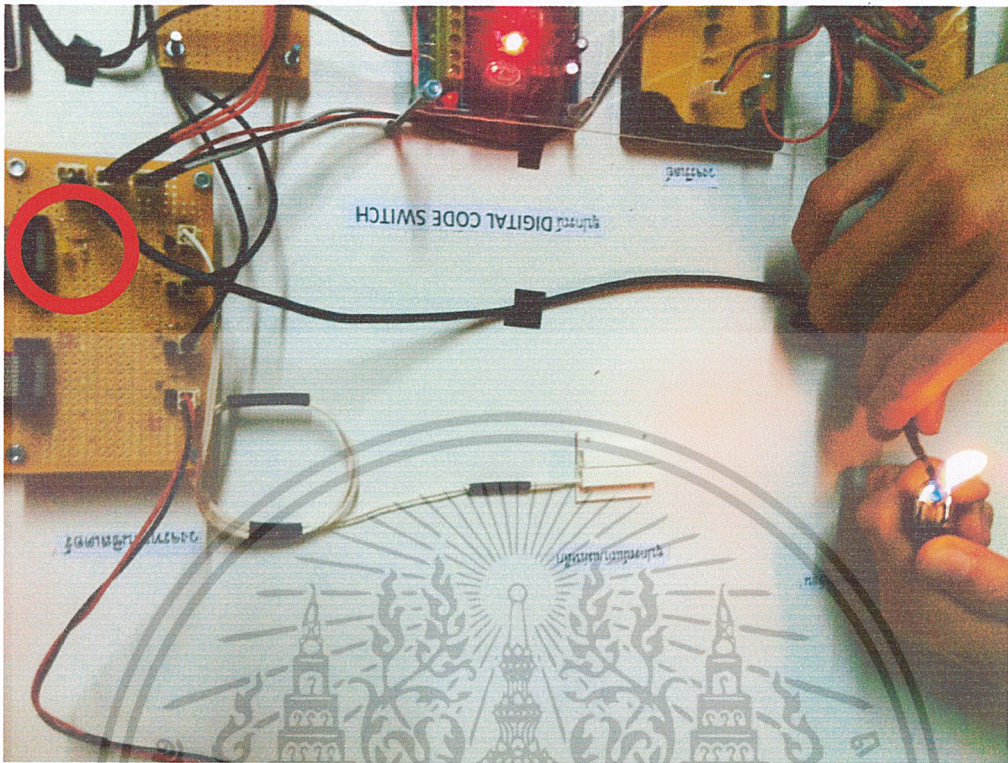
รูปที่ 4.8 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์ตรวจจับควัน



รูปที่ 4.9 แสดงการทำงานของปั้มน้ำ

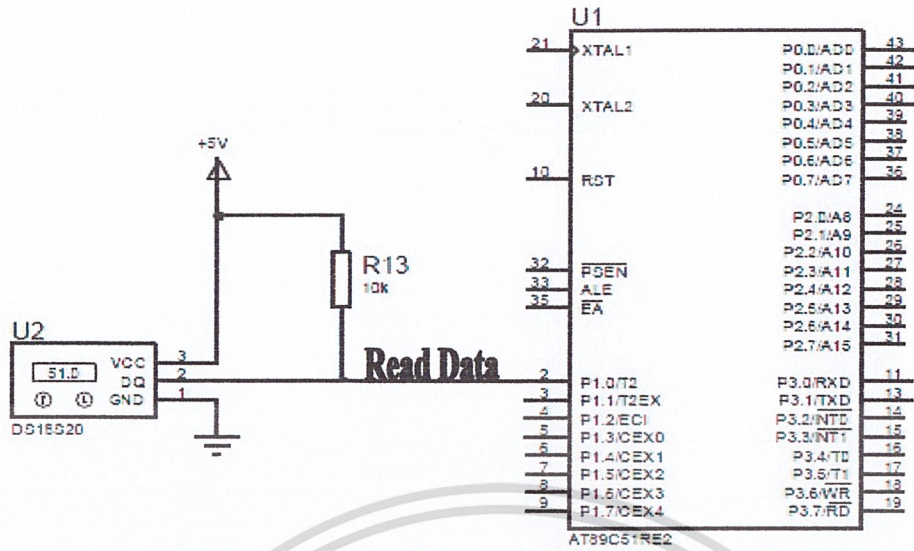
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองของอุปกรณ์ตรวจจับความร้อน



รูปที่ 4.10 แสดงอุปกรณ์ตรวจจับความร้อนในขณะที่ยังไม่ทำงาน

ในขณะที่อุปกรณ์ตรวจจับความร้อนนั้น ยังไม่สามารถตรวจจับสิ่งผิดปกติได้ ไฟบอกลสถานะของอุปกรณ์และไฟสถานะของระบบจะดับ ดังรูปที่ 4.10 และระบบจะไม่ทำการ ส่งข้อความสั้น (SMS) และปั๊มน้ำก็จะไม่ทำงานด้วยเช่นกัน



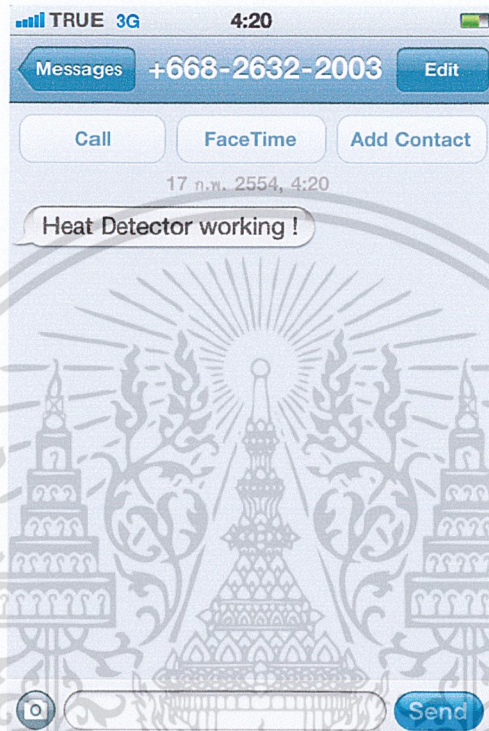
รูปที่ 4.11 แสดงการไหลของกระแสของอุปกรณ์ตรวจจับความร้อนในขณะทำงาน



รูปที่ 4.12 แสดงอุปกรณ์ตรวจจับความร้อนในขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ตรวจจับความร้อนจะทำงานเมื่อมีการอ่านข้อมูลของอุณหภูมิได้มากกว่า 70 องศาเซลเซียส เมื่ออุปกรณ์ตรวจจับความร้อนทำงาน ไฟบอกสถานะของการทำงานจะแสดงดังรูปที่ 4.12 แล้วไมโครคอนโทรลเลอร์ จะทำการสั่งให้ GSM/GPRS Module ทำการส่ง ข้อความสั้น (SMS) เข้าโทรศัพท์มือถือ และสั่งให้ ขาที่ 2 Port1 มีสถานะเป็น High เป็นผลให้ปั้มน้ำทำงานด้วย ดังรูปที่ 4.13 และ 4.14



รูปที่ 4.13 แสดงข้อความสั้น (SMS) จากการทำงานของอุปกรณ์ตรวจจับความร้อน

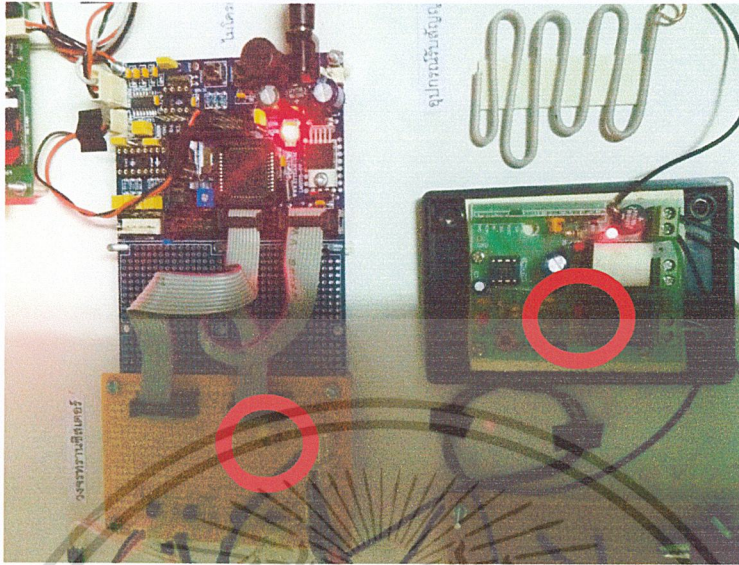


รูปที่ 4.14 แสดงการทำงานของปั้มน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
55

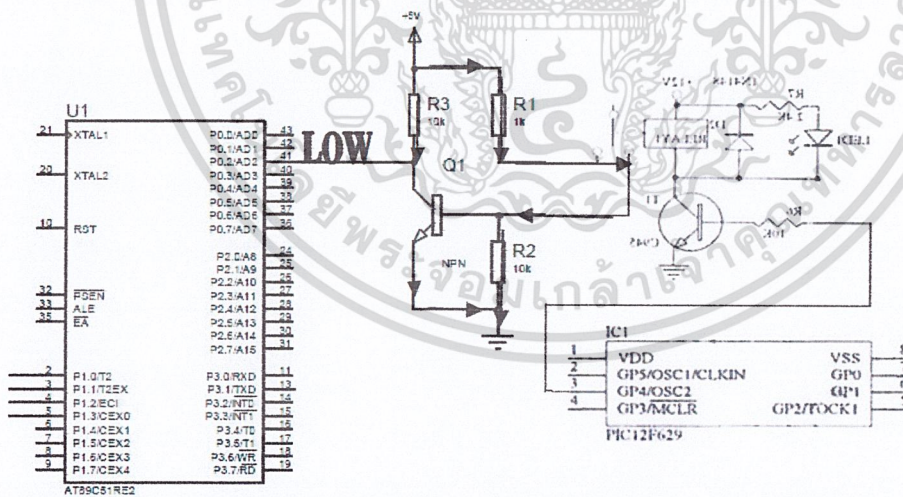
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองของอุปกรณ์รับ-ส่งสัญญาณรีโมท



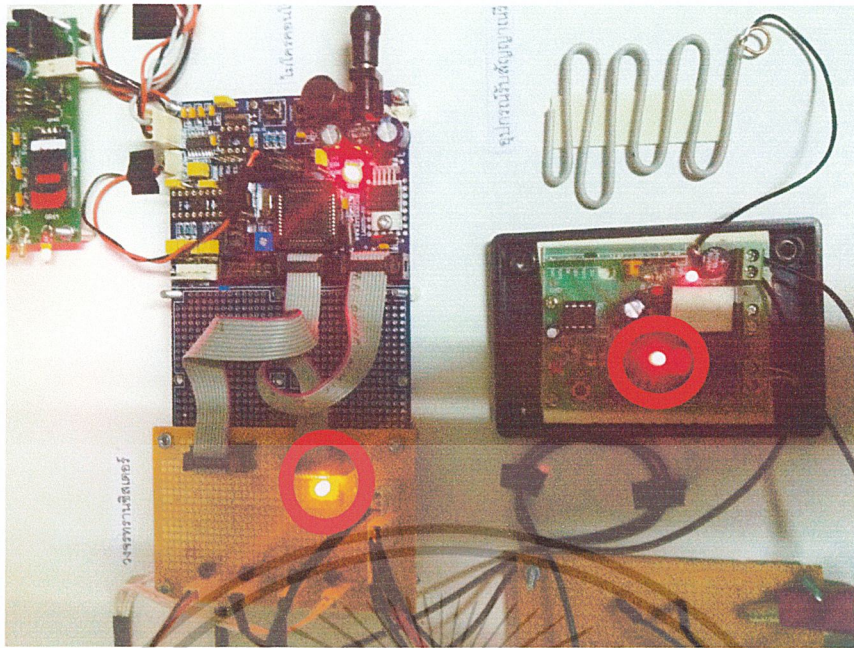
รูปที่ 4.15 แสดงอุปกรณ์รับสัญญาณรีโมทในขณะที่ยังไม่ทำงาน

ในขณะที่อุปกรณ์รับ-ส่งสัญญาณรีโมทนั้น ยังไม่ทำงาน ไฟบอกสถานะของอุปกรณ์และไฟสถานะของระบบจะดับ ดังรูปที่ 4.15 และระบบจะยังทำงานอยู่



รูปที่ 4.16 แสดงการไหลของกระแสของอุปกรณ์รับสัญญาณรีโมทในขณะที่ทำงาน

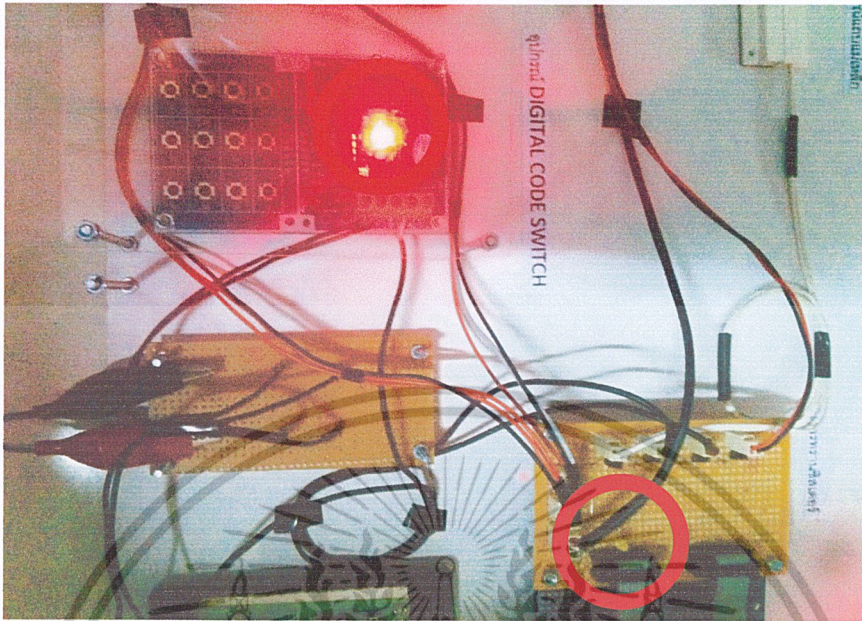
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงอุปกรณ์รับสัญญาณรีโมทในขณะที่ทำงาน

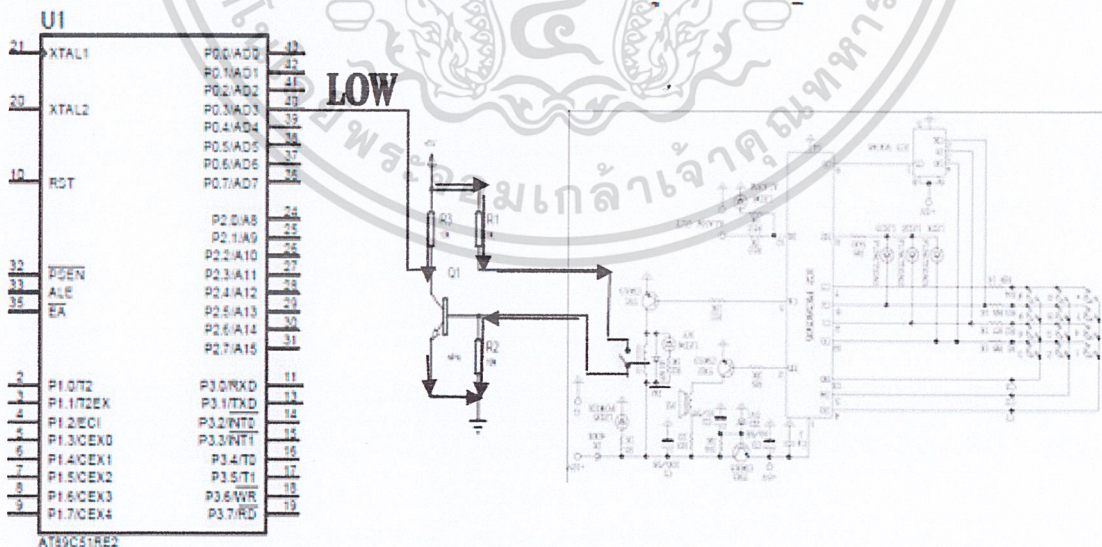
เมื่อตัวอุปกรณ์รับการรับ-ส่งสัญญาณรีโมทรับสัญญาณจากรีโมทได้จะทำให้กระแสไฟฟ้าไหลดังรูปที่ 4.16 ซึ่งทำให้ขาที่ 1 Port0 มีสถานะเป็น Low ไฟสถานะของการทำงานก็จะแสดงดังรูปที่ 4.17 แล้วไมโครคอนโทรลเลอร์ จะทำการสั่งให้ ปิดการทำงานทุกอุปกรณ์ ตรวจจับ เป็นผลให้สามารถเข้าออกจากที่พักได้โดยปกติ

4.5 การทดลองของอุปกรณ์ Digital Code Switch



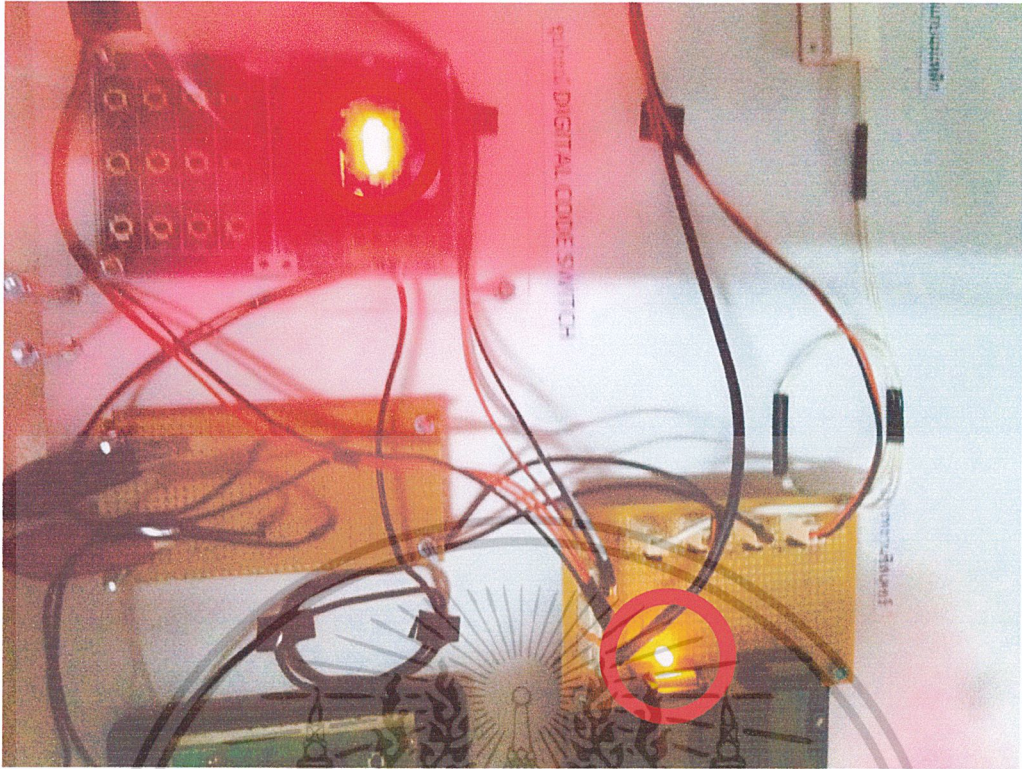
รูปที่ 4.18 แสดงอุปกรณ์ Digital Code Switch ในขณะที่ยังไม่ทำงาน

ในขณะที่อุปกรณ์ Digital Code Switch นั้น ยังไม่ทำงาน ไฟบอกสถานะของอุปกรณ์และไฟสถานะของระบบจะดับ ดังรูปที่ 4.18 และระบบจะยังทำงานอยู่



รูปที่ 4.19 แสดงการไหลของกระแสของอุปกรณ์ Digital Code Switch ในขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงอุปกรณ์ Digital Code Switch ในขณะทำงาน

เมื่อตัวอุปกรณ์ Digital Code Switch มีการทำงาน จะทำให้กระแสไหลดังรูปที่ 4.19 ซึ่งทำให้ขาที่ 2 Port0 มีสถานะเป็น Low ไฟสถานะของการทำงานก็จะแสดงดังรูปที่ 4.20 แล้วไมโครคอนโทรลเลอร์จะทำการสั่งให้ปิดการทำงานทุกอุปกรณ์ตรวจจับ เป็นผลให้สามารถเข้าออกจากที่พักได้โดยปกติ

บทที่ 5

สรุปผลดำเนินโครงการ

ในบทนี้จะกล่าวถึงปัญหาและอุปสรรคในการดำเนินพร้อมทั้งเสนอวิธีแก้ไข เพื่อให้เกิดประโยชน์ต่อผู้ที่ศึกษาโครงการนี้ต่อ เพื่อเป็นแนวทางในการพัฒนาต่อไป

5.1 บทวิจารณ์และสรุปผล

ในโครงการระบบรักษาความปลอดภัยและการแจ้งเตือนผ่านโทรศัพท์มือถือ (Security system and alerts via mobile phone) ในสำหรับส่วนของวิชาโครงการที่ 2 คณะผู้จัดทำได้ทำการออกแบบและจัดทำวงจรต่างๆ ทั้งอุปกรณ์ตรวจจับความผิดปกติต่างๆ อุปกรณ์ควบคุมระบบ รวมถึงการเขียนโปรแกรมเพื่อควบคุมการทำงานต่างๆ จากการทดลอง แล้วได้ผลคือ อุปกรณ์ตรวจจับความผิดปกติทั้งสามชนิดคือ อุปกรณ์แถบแม่เหล็ก อุปกรณ์ตรวจจับควัน อุปกรณ์ตรวจจับความร้อน สามารถตรวจจับความผิดปกติได้อย่างแม่นยำ มีบางครั้งที่อุปกรณ์ตรวจจับความร้อนนั้นตรวจจับความผิดปกติผิดพลาด โดยเมื่อจับความผิดปกติได้ก็จะส่งผลไปยังไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์จะส่งใน GSM/GPRS Module ส่งข้อความสั้น (SMS) ไปยังอุปกรณ์มือถือได้ และยังสั่งให้ปั๊มน้ำทำงานได้ ในส่วนของไมโครคอนโทรลเลอร์สามารถสั่งงานได้อย่างถูกต้องตามที่ผู้จัดทำได้เขียนโปรแกรมไว้ แต่ในส่วนของอุปกรณ์ส่งข้อความสั้น (SMS) นั้นยังทำงานผิดพลาดอยู่บ้างในบางครั้ง คือ บางครั้งส่งข้อความสั้น (SMS) ได้ไม่ครบตามจำนวนเบอร์โทรศัพท์ที่ตั้งไว้ ส่วนของอุปกรณ์ควบคุมระบบ นั้นสามารถทำงานได้ดีแต่ยังไม่สะดวกเท่าที่ควร

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

- ในการออกแบบวงจรนั้นมี วงจรหลายส่วน จึงมีปัญหาในการต่อ วงจร
- การเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์นั้นทำได้ยาก เนื่องจาก มีส่วนของการควบคุมเป็นจำนวนมาก และเมื่อนำแต่ละส่วนมารวมกัน บางครั้งยังมีการทำงานผิดพลาดอยู่บ้าง
- การเขียนคำสั่งในการควบคุม GSM/GPRS Module นั้นต้องใช้คำสั่งเฉพาะสำหรับควบคุมการทำงาน ซึ่งมีความซับซ้อนและเป็นคำสั่งที่ไม่มีพื้นฐานมา

ก่อน จึงทำให้ต้องใช้เวลาศึกษาพอสมควร และยังคงความชำนาญในการใช้
อุปกรณ์จึงทำให้การใช้งานอุปกรณ์ยังมีความผิดพลาดอยู่บ้าง

- อุปกรณ์การส่งข้อความสั้น (SMS) นั้นยังต้องใช้ระบบเติมเงิน ถ้าเงินในระบบ
หมดก็ไม่สามารถรู้ได้จึงทำให้ส่งข้อความสั้น (SMS) ไม่ได้
- ในการทดลองอุปกรณ์ตรวจจับความร้อน เนื่องจากความไม่ถนัดในการใช้
อุปกรณ์จึงทำให้อุปกรณ์เสียหายและต้องเปลี่ยนอุปกรณ์อยู่บ่อยครั้ง และบางครั้ง
ยังวัดอุณหภูมิผิดพลาด
- อุปกรณ์ควบคุมระบบนั้นยังไม่สามารถเขียนโปรแกรมควบคุมได้อย่างสมบูรณ์
จึงทำให้ ควบคุมระบบได้อย่างจำกัด

5.3 แนวทางการแก้ไขปัญหาและพัฒนาต่อ

- ปรับปรุงและพัฒนาคำสั่งที่ใช้ควบคุมและสั่งการไมโครคอนโทรลเลอร์
- ศึกษาทำความเข้าใจอุปกรณ์ GSM/GPRS Module แล้วพัฒนาคำสั่งควบคุม
- เปลี่ยนไปใช้การส่ง ข้อความสั้น (SMS) แบบรายเดือนหรือ โปรโมชันการส่ง
ข้อความสั้น (SMS) แบบบุฟเฟต์
- เพิ่มชุดควบคุมการปิด/เปิดระบบ เพื่อความสะดวกในการใช้งาน

บรรณานุกรม

1. ประสิทธิ์ สุทธิปริญญาพันธ์. “พื้นฐานการรักษาความปลอดภัยในบ้าน” คอมพิวเตอร์อิเล็กทรอนิกส์เวิร์ลด์ ฉบับที่ 146,2537
2. **User’s Manual CPJR51RE2**
<http://www.ett.co.th/product/mcs51/man_CPJR51RE2_V1.pdf>
3. โปรแกรมตัวอย่างการใช้ AT Command <<http://nssurvey.tarad.com/article?id=44248&lang=th>>
4. การใช้งานพอร์ตอนุกรม RS232 <<http://www.thaimicrotron.com/CCS-628/Reference/RS232.htm>>
5. ข้อมูลของทรานซิสเตอร์ <http://www.chontech.ac.th/~electric/e-learn/unit12/unit12.htm>
6. ข้อมูลของเซ็นเซอร์จับความร้อน <http://www.maxim-ic.com/datasheet/index.mvp/id/2815>
7. ข้อมูลอุปกรณ์ตรวจจับควัน <http://www.allproducts.com/ee/safcea/ci-180.html>
และ Datasheet



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับการทำงานของโครงงาน

```
#include <at89c51xd2.h>
#include <stdio.h>

#define MAX_NUM 3

sbit SMOKE = P0^0;
sbit REMOTE = P0^1;
sbit PASS = P0^2;
sbit DOOR = P0^3;
sbit TMDAT = P1^0;
sbit RY1 = P1^1;
sbit RY2 = P1^2;
sbit LED = P1^3;

void delay_ms (unsigned int count);

#include "ds18s20.c"

bit TXOK; // tx flag
bit WORK;
bit s_smoke;
bit s_door;
bit s_temp;

unsigned char index;

code unsigned char BUFNUM1[]={"0860409401"}; // number 1
code unsigned char BUFNUM2[]={"0800457950"}; // number 2
code unsigned char BUFNUM3[]={"0894555856"}; // number 3
code unsigned char BUFNUM4[]={" "}; // number 4
code unsigned char BUFNUM5[]={" "}; // number 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
code unsigned char BUFMESSAGE1[]={"Smoke detector working !"};

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
code unsigned char BUFMESSAGE2[]={ "Heat Detector working !"};
code unsigned char BUFMESSAGE3[]={ "Magnatic Sensor working !"};
```

```

/*****/
void delay_ms (unsigned int count) { // mSec Delay
    unsigned int i;
    while (count) {
        i = 190; while (i>0) i--;
        count--;
    }
}
/*****/
char putchar (unsigned char x) { // change putchar for printf
    while (!TXOK);
    TXOK = 0;
    return (SBUF = x);
}
/*****/
void sim300_ok (void) {
    unsigned char i;
    while(1){
        WORK = 0;
        printf("AT\r");
        i = 0;
        while(!WORK){
            i++;
            LED= ~LED;
            if(i>=5)break;
            if(WORK) return;
            delay_ms(300);
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 LED = 1;
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(WORK) return;
    }
}

/*****/

void set_sms (unsigned char *num,unsigned char *MESSAGE) {
    sim300_ok ();                // check sim300 ok
    sim300_ok ();                // check sim300 ok
    delay_ms(200);
    printf("AT+CMGF=1\r");      // text mode
    delay_ms(100);
    printf("AT+CMGS=\"%s\r",num); // print number
    delay_ms(100);
    printf("%s",MESSAGE);      // print message
    delay_ms(200);
    putchar(0x1A);             //send sms // Ctrl z = 0x1A
    delay_ms(2500);
}

void send_sms (unsigned char *MESSAGE) {
    int i;
    for(i=1;i<=MAX_NUM;i++){
        switch(i) {
            case 1: set_sms (BUFNUM1,MESSAGE);
                    break;

            case 2: set_sms (BUFNUM2,MESSAGE);
                    break;

            case 3: set_sms (BUFNUM3,MESSAGE);
                    break;

            case 4: set_sms (BUFNUM4,MESSAGE);
                    break;

            case 5: set_sms (BUFNUM5,MESSAGE);
                    break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/*****/

void inputsim_ok (unsigned char a){
    if(a=='O') index = 'K';
    else if((index=='K') && (a=='K')) WORK = 1;
    else index = 0;
}

/*****/

void serial (void) interrupt 4 {    // serial interrupt
    unsigned char a;                // receive binary command
    if (TI) {                        // TXD
        TI = 0;
        TXOK = 1;
        return;
    }
    a = SBUF;                        // RXD
    RI = 0;
    if(a==0);
    else inputsim_ok (a);
}

/*****/

bit read_temp (unsigned char Alarm) {
    unsigned int x;
    unsigned char t;

    tmstart ();
    x = tmrtemp ();
    t = x;
    // printf("TEMP = %02bd\n\r",t);
    if( t >= Alarm ) return (0);
    else return (1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
void check_sensor (void) {
    bit temp;

    temp = read_temp (70);           //alarm 70.0 C

    if(temp==0){
        LED = ~LED;
        if(s_temp==0){
            s_temp = 1;
            RY2 = 1;
            send_sms (BUFMESSAGE2);
        }
    }
    else {
        LED = 1;
    }
}
/*-----*/
if(SMOKE==0){
    LED = ~LED;
    delay_ms(500);
    if(s_smoke==0){
        if(SMOKE==0){
            s_smoke = 1;
            RY1 = 1;
            send_sms (BUFMESSAGE1);
        }
    }
}
else {
    LED = 1;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(DOOR==1){
    LED = ~LED;
    delay_ms(500);
    if(s_door==0){
        if(DOOR==1){
            s_door = 1;
            send_sms (BUFMESSAGE3);
        }
    }
}
else {
    LED = 1;
    delay_ms(500);
    if(DOOR==0) s_door = 0;
}
}
/*****/
void Initial_Uart0 (void) {
    SCON = 0x50 ;           //Serial Mode1:8bit Uart
    TH1 = 0xFB ;           //Set Baud rate = 9600 at Crystal=18.432 Mhz
//    TL1 = 0xFB ;
    TMOD = 0x20 ;          //Use Sclk Timer1 Mode2
    TI = 1 ;               //SCON Reg.: SET Flag TI
    TR1 = 1 ;              //Set TCON Reg. On Timer1

    ES = 1;                //serial interrupt enable
    EA = 1;                // enable all interrupt
    delay_ms(1000);
}
/*****/
void main (void) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Initial_Uart0 ();
sim300_ok ();
// send_sms (BUFMESSAGE1);

/*
while(1){
    printf("-----\r\n");
    if(SMOKE) printf("SMOKE = 1\r\n");
    else printf("SMOKE = 0\r\n");

    if(REMOTE) printf("REMOTE = 1\r\n");
    else printf("REMOTE = 0\r\n");

    if(PASS) printf("PASS = 1\r\n");
    else printf("PASS = 0\r\n");

    if(DOOR) printf("DOOR = 1\r\n");
    else printf("DOOR = 0\r\n");

    read_temp (70);

    LED = ~LED;
    RY1 = ~RY1;
    RY2 = ~RY2;
    delay_ms(1000);

}
*/

read_temp (70);
read_temp (70);
while(1) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงบนสื่อใดๆ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
s_smoke = 0;
s_door = 0;
s_temp = 0;
RY1 = 0;
RY2 = 0;
LED = 0;
}
else {
    check_sensor();
    LED = 1;
}
delay_ms(500);
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับอุปกรณ์ตรวจวัดอุณหภูมิ

```
void tmreset (void) {          // Reset TX (Speed x2)
    unsigned int i;
    TMDAT = 0;
    i = 412; while (i>0) i--;   // Approx 900 uS
    TMDAT = 1;
    i = 10; while (i>0) i--;
}
```

```
bit tmpre (void) {           // Wait for Presence RX (Speed x2)
    unsigned char i;         // return 0=OK 1=ERROR
    unsigned int a;
    a = 0;
    while (TMDAT)
        if (a++>4000) return (1);
    while (~TMDAT);
    i = 10; while (i>0) i--;
    return (0);
}
```

```
bit tmrbit (void) {         // read one bit (Speed x2)
    unsigned int i;
    bit dat;
    TMDAT = 0; i++; i++;
    TMDAT = 1; i++; i++; i++; i++;
    dat = TMDAT;
    i = 36; while (i>0) i--;  // Approx 65 uS
    return (dat);
}
```

```
unsigned char tmrbyte (void) { // read one byte (Speed x1 x2)
```

```
    unsigned char i,j,dat;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=1;i<=8;i++) {
    j = tmrbit();
    dat = (j << 7) | (dat >> 1);
}
return (dat);
}

void tmwbyte (unsigned char dat) { // write one byte (Speed x2)
    unsigned int i;
    unsigned char j;
    bit testb;
    for (j=1;j<=8;j++) {
        testb = dat & 0x01;
        dat = dat >> 1;
        if (testb) {
            TMDAT = 0; // Write 1
            i++; i++; i++; i++; // Approx 4 uS
            TMDAT = 1;
            i = 36; while (i>0) i--; // Approx 65 uS
        }
        else {
            TMDAT = 0; // Write 0
            i = 36; while (i>0) i--; // Approx 65 uS
            TMDAT = 1;
            i++; i++; i++; i++; // Approx 4 uS
        }
    }
}

void tmstart (void) { // ds18s20 start convert
    tmreset ();
    if (tmpre ()==1) return;
    else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 delay_ms(1);
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    tmwbyte (0xcc);    // skip rom
    tmwbyte (0x44);    // convert
}
}

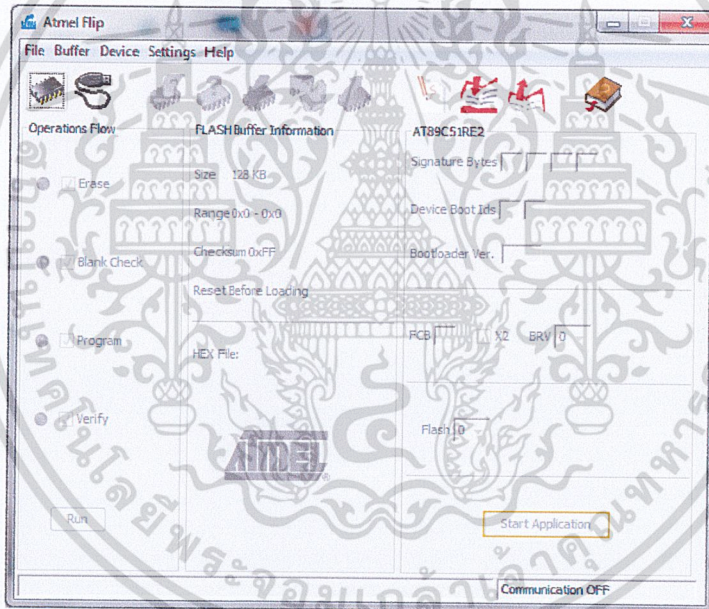
int tmrtemp (void) {    // read temp
    int x;
    unsigned char a,b;
    tmreset ();
    if (tmpr ()==1) return (0);
    else {
        dclay_ms(2);
        tmwbyte (0xcc);    // skip rom
        tmwbyte (0xbe);    // convert
        a = tmrbyte ();    // LSB
        b = tmrbyte ();    // MSB
        if (b==0xff) return (0); // don't care negative temp
        x = a & 0x1;    // 0=x.0 1=x.5
        a = a >> 1;
        if (a>99) {    // max 99.5 c
            a = 99;
            x = 1;
        }
        x = (x << 8) | a;
    }
    return (x);    // L-byte=0,1 R-byte=temp
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการ Download โปรแกรม

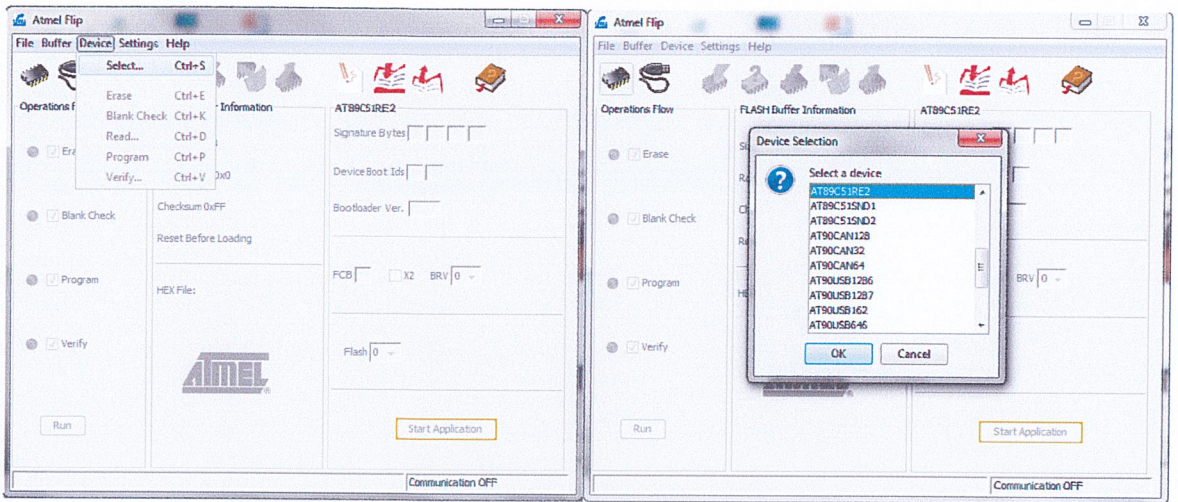
1. ทำการติดตั้งโปรแกรม Java ลงในเครื่อง
2. ทำการติดตั้งโปรแกรม Flip ลงในเครื่อง
3. ต่อสาย RS232 จาก PC เข้ากับขั้วต่อ RS232#1 ของบอร์ดคอนโทรลเลอร์
4. ทำการกดสวิทช์ PSEN และ RESET ดังต่อไปนี้เพื่อเข้าสู่ Monitor Mode
 - กด SW.PSEN ค้างไว้
 - ตามด้วยการกด SW.RESET ค้างไว้
 - ปลด SW.RESET ในขณะที่ SW.PSEN ยังถูกกดค้างอยู่
 - ปลด SW.PSEN เป็นลำดับสุดท้าย
5. ให้เปิดโปรแกรม Flip ขึ้นมา จะได้นหน้าต่างดังรูปที่ 1



รูปที่ 1

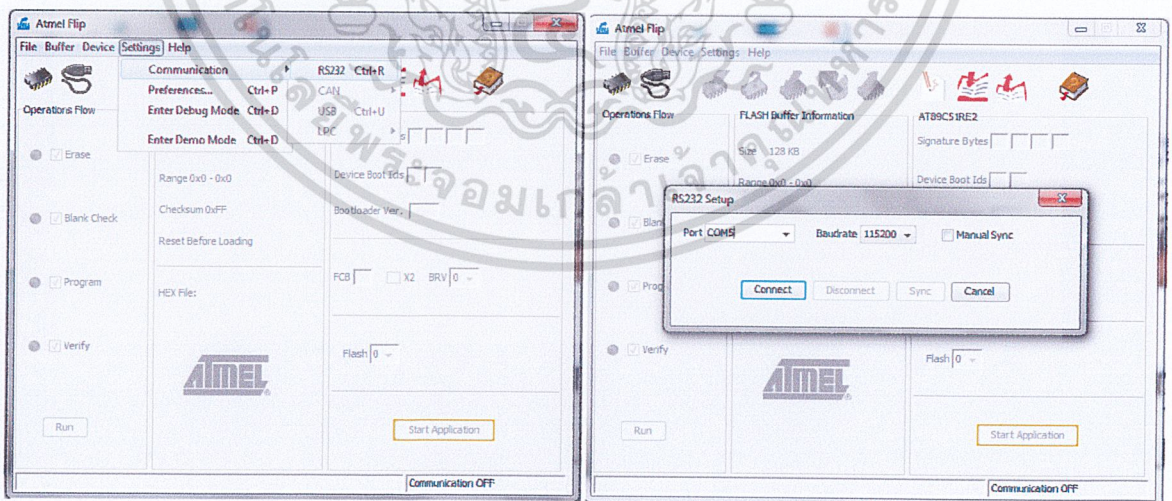
6. ให้คลิกที่ Device แล้วเลือก Select แล้วให้เลือกบอร์ดคอนโทรลเลอร์ เมื่อเลือกเสร็จให้กด OK.
ดังรูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



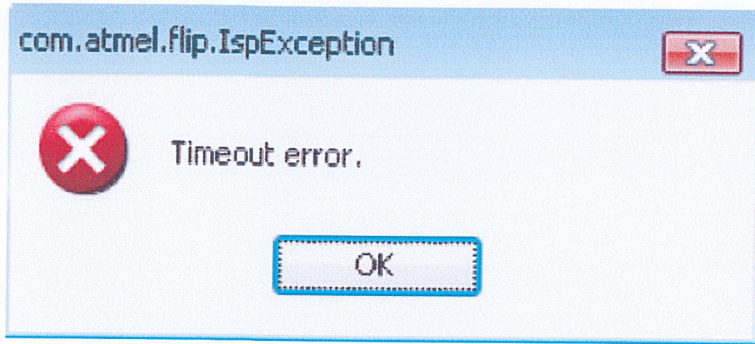
รูปที่ 2

7. ให้คลิกที่ Settings > Communication > RS232 แล้วจะได้หน้าต่าง RS232 Setup จากนั้นให้เลือก ComPort ของ PC แล้วเลือก Baud Rate (Default=115200) ที่จะใช้ในการ Download โปรแกรม เมื่อเลือกเรียบร้อยแล้วให้คลิก Connect ดังรูปที่ 3 ก่อนที่จะคลิก Connect ต้องให้คอนโทรลเลอร์ทำงานอยู่ใน Monitor Mode ก่อน โดยให้ทำตามขั้นตอนที่ 4 ถ้าการ Connect ผ่านไอคอนต่างๆ ในหน้าต่าง Atmel Flip ก็จะถูก Enable ให้ใช้งานได้ดังรูปที่ 4 แต่ถ้าการ Connect ไม่ผ่านก็จะมีหน้าต่างขึ้นมาเตือนดังรูปที่ 5 ซึ่งต้องทำการตรวจสอบการต่อสายของ RS232 และเข้าสู่ Monitor Mode ว่าถูกต้องหรือไม่แล้วทำการ Connect ใหม่อีกครั้ง

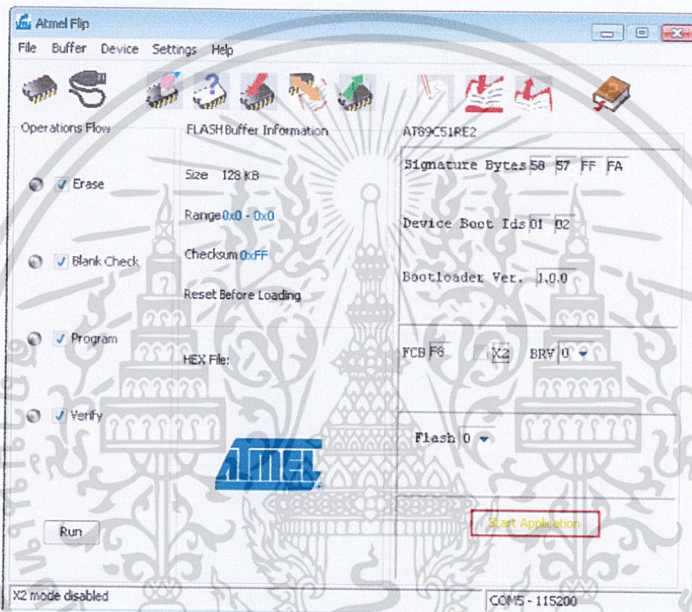


รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



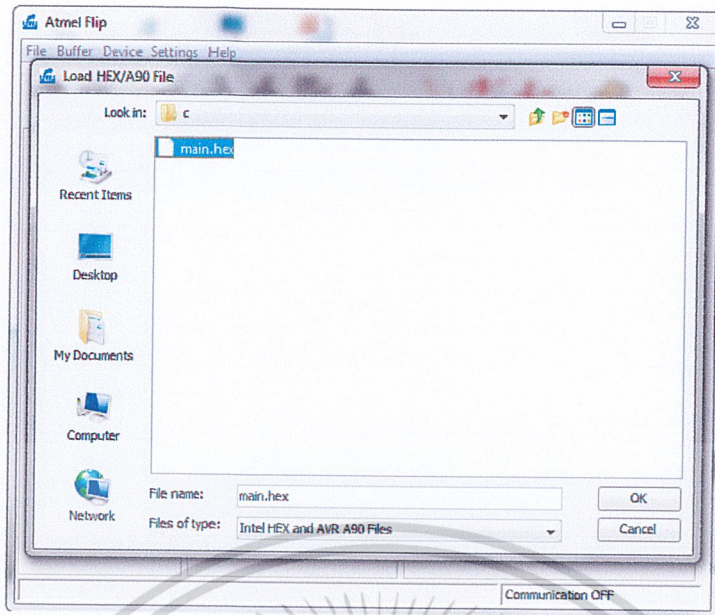
รูปที่ 4



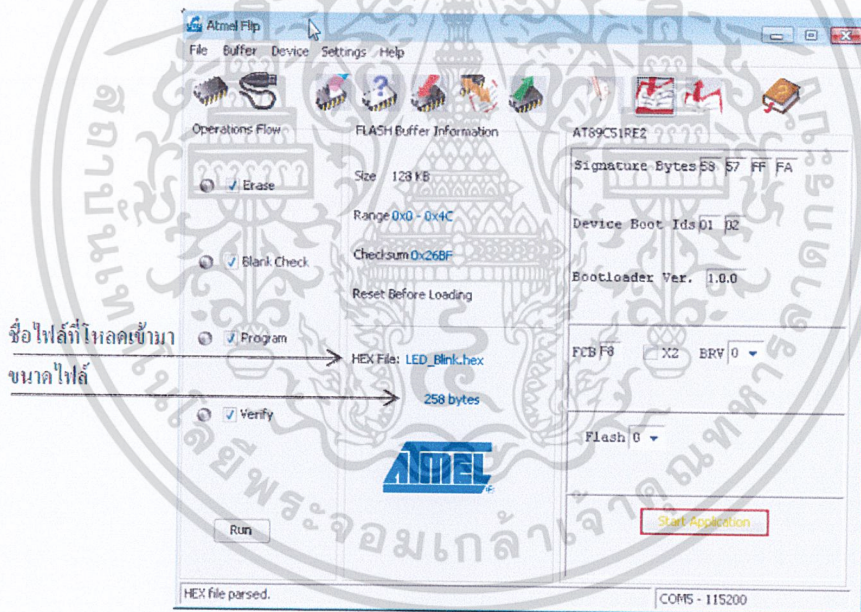
รูปที่ 5

8.หลังจาก Connect Port ได้เรียบร้อยแล้วให้คลิกที่ File > Load HEX file แล้วทำการเลือก file.Hex จากภายนอกเข้ามายังโปรแกรมดังรูปที่ 6 ซึ่งชื่อของ file ที่เลือกเข้ามาจะปรากฏที่หน้าต่าง Atmel Flip ในช่อง Hex File ดังรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



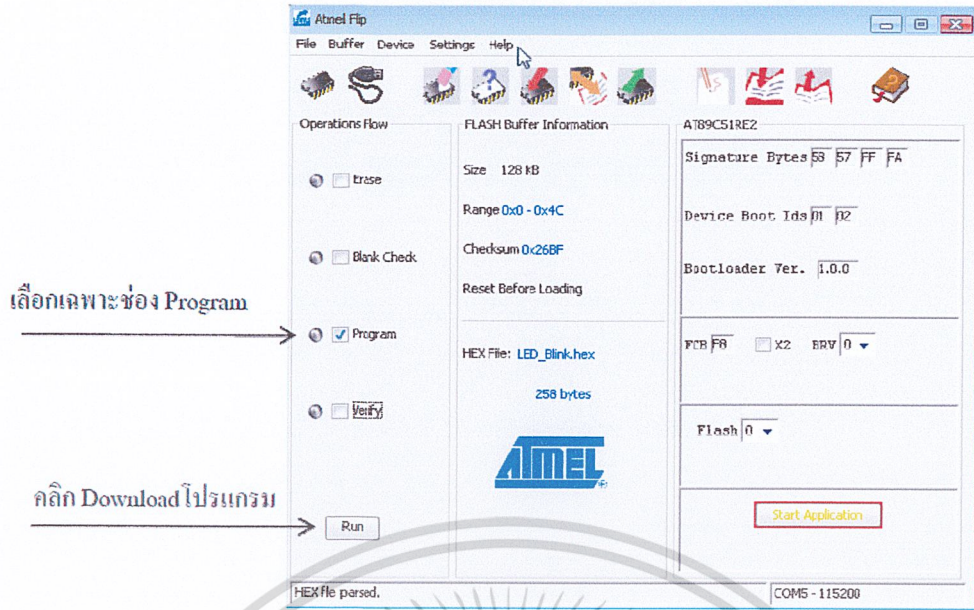
รูปที่ 6



รูปที่ 7

9.เมื่อโหลด file เข้ามาแล้ว ให้คลิกที่ปุ่ม Run เพื่อ Download โปรแกรมลงไปในไมโครคอนโทรลเลอร์ ซึ่งถ้าผู้ใช้ต้องการให้โหลดเร็วขึ้น จากหน้าต่าง Atmel Flip ก่อนที่จะคลิกปุ่ม Run ในช่อง Operation Flow ให้คลิกเลือกเฉพาะช่อง Program เพียงช่องเดียวก็พอ ส่วนช่องอื่นให้คลิกออกให้หมด ดังรูปที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8

10.หลังจาก Download เรียบร้อยแล้ว ให้กดที่สวิตซ์ RESET ของบอร์ดไมโครคอนโทรลเลอร์ เพื่อ Run โปรแกรมดูผลการทำงานของโปรแกรมที่เขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรม Keil uVision3 กับ AT89C51RE2

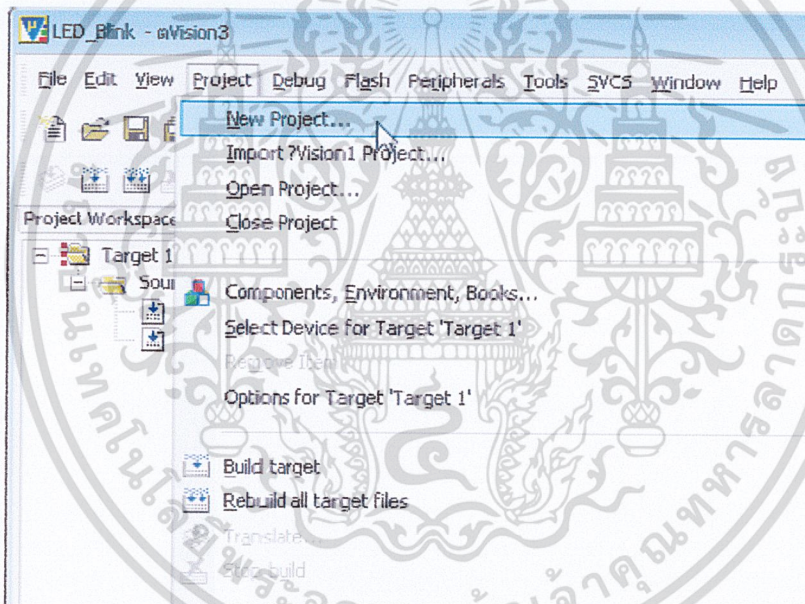
สำหรับโปรแกรม Keil uVision3 นี้จะเป็นโปรแกรมที่ใช้สำหรับพัฒนาการเขียนโปรแกรมของผู้ใช้ด้วยภาษาซี ซึ่งโปรแกรมตัวนี้จะทำหน้าที่เป็น Compiler ในตัว และจะสร้าง Hex ไฟล์ให้กับผู้ใช้เพื่อนำไปใช้ Download ลงในตัวไมโครคอนโทรลเลอร์ โดยตัวที่ให้ไปกับบอร์ดนั้นจะเป็นตัว DEMO สามารถ Flash Code ได้ไม่เกิน 2 K ก่อนที่จะเริ่มต้นใช้งาน Keil uVision3 เราจะมาทำความเข้าใจเกี่ยวกับไมโครคอนโทรลเลอร์ เบอร์นี้เสียก่อน คือ สำหรับ AT89C51RE2 เบอร์นี้จะมี RAM 8 K และพื้นที่ Flash Memory ภายในสำหรับเก็บ Code 128 K แต่ลักษณะของพื้นที่ Flash นั้น ไม่ได้เป็นพื้นที่ต่อเนื่อง แต่จะถูกแบ่งออกเป็น 4 Bank Bank ละ 32 K ได้แก่ Common Bank (0000h-7FFFh), Bank0 (8000h-FFFFh), Bank1 (8000h-FFFFh), Bank2(8000h-FFFFh) เมื่อผู้ใช้พัฒนาโปรแกรมด้วย Software Keil uVision3 ผู้ใช้จะต้องทำการกำหนดค่าต่างๆ ของ Project ให้ถูกต้องเพื่อให้สามารถใช้งานพื้นที่ Flash Memory ได้ครอบคลุม 128 K มิฉะนั้น พื้นที่ Flash จะถูกใช้งานได้เพียง 64 K เท่านั้น ถ้าโปรแกรมนั้นมีขนาดเกินกว่านี้ก็จะไม่สามารถ Flash ลงไปได้ถ้าไม่ทำการกำหนดคุณสมบัติของ Project File ให้ถูกต้องดังนั้นใน Software Keil uVision3 นี้จะมีวิธีการกำหนดคุณสมบัติให้กับตัว Project File เพื่อให้สามารถใช้งานพื้นที่ Flash Memory ได้ถึง 128 K ตามคุณสมบัติของไมโครคอนโทรลเลอร์ และสามารถที่จะเลือก Flash ลง Bank ใดก็ได้ ใน 4 Bank และในแต่ละ Bank นั้นสามารถเรียกใช้โปรแกรมน้อยที่เขียนอยู่ต่าง Bank กันได้โดยผู้ใช้ไม่ต้องเสียเวลาในการเขียนโปรแกรมสลับ Bank ไปมา ซึ่งอาจจะทำให้ผู้ใช้เขียนผิดพลาดได้เนื่องจากจะเห็นว่าโครงสร้างของไมโครคอนโทรลเลอร์ ตั้งแต่ Bank0-2 นั้น จะอ้างตำแหน่งแอดเดรสเดียวกัน ดังนั้นใน Software Keil uVision3 จะช่วยจัดการในส่วนต่างๆเหล่านี้ให้ ต่อไปเราจะมาพูดถึงการใช้งานในส่วนของโปรแกรม Keil uVision3 กันบ้าง ซึ่งเราจะขอแยกอธิบายการใช้งานออกเป็น 2 แบบ ดังนี้

1. การใช้งาน Keil uVision3 กับ AT89C51RE2 แบบไม่ใช้การ SWITCH Bank

ซึ่งการใช้งานในส่วนนี้จะเป็นการใช้งานแบบปกติ การกำหนดคุณสมบัติให้กับ Project File ที่เขียนขึ้นก็จะไม่ยุ่งยาก แต่จะมีข้อจำกัดอยู่ว่า พื้นที่ที่ใช้สำหรับ Flash Code ของ MCU จะใช้ได้เพียง 64 K ไม่เต็ม 128 K โดยพื้นที่ Flash 64 K ที่ใช้ได้นี้ ก็จะเป็นพื้นที่ในส่วนของ Common Bank 32 K ส่วนอีก 32 K ตัว Keil จะเป็นตัวสุ่มเองว่าจะใช้ Bank ไหน แต่ขอแนะนำให้ผู้ใช้งานกำหนดพื้นที่ Bank ที่เหลือนี้เองเพื่อเวลาเขียนโปรแกรมจะได้ไม่มีปัญหาในภายหลัง โดยสามารถกำหนดได้คือ ในการเขียนโปรแกรมทุกครั้งเมื่อเลือกใช้งานแบบ ไม่ใช้การ Switch Bank ก็ให้ทำการกำหนดค่าใน Register BMSEL = 0x00 ในตอนต้นของโปรแกรมเสีย (ดูได้จากตัวอย่างในเอกสารนี้) เพื่อกำหนดให้ใช้พื้นที่ Flash 64 K นี้ ใน Common Bank และ Bank0 เสมอ ต่อไปจะเป็น

การใช้งาน Keil แบบไม่ใช้การ SwitchBank ตามขั้นตอนดังต่อไปนี้

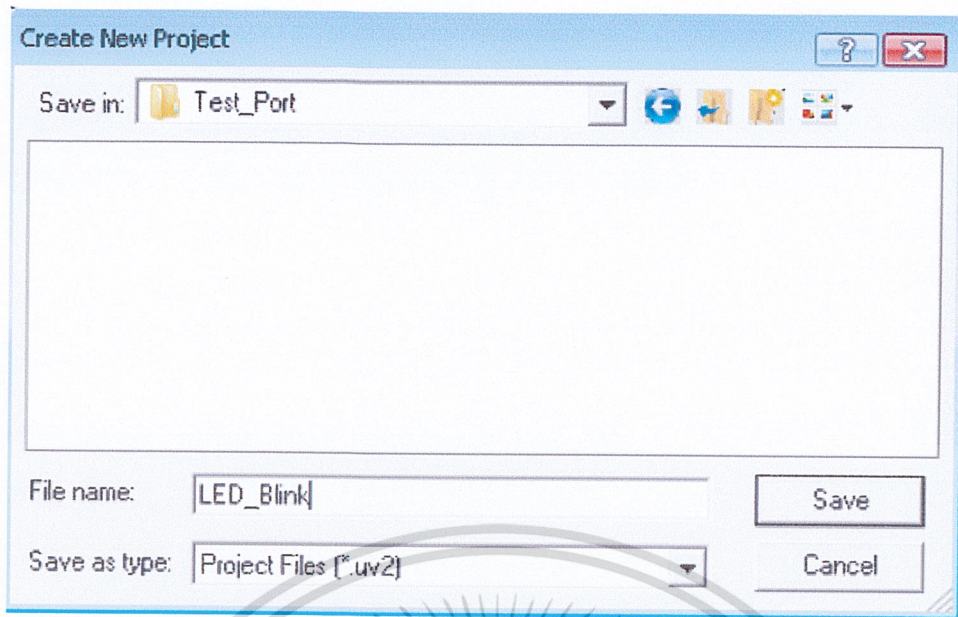
1. ติดตั้งโปรแกรม Keil uVision3 ลงบนเครื่อง PC เมื่อติดตั้งเสร็จโดยปกติโปรแกรมจะถูกเก็บไว้ที่ C:\Keil
2. ให้ทำการ Copy File: at89c51re2.h ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\INC\Atmel เพื่อเอาไว้เรียกใช้ในส่วนของ Include File เมื่อเขียนโปรแกรมจะได้ไม่เสียเวลาประกาศตัวแปรต่างๆอีก
3. Copy File: STARTUP.A51 และ L51_BANK.A51 ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\LIB ซึ่งได้ทำการแก้ไขให้รองรับ MCU เบอร์นี้ไว้แล้ว แต่อาจจะไม่รองรับเบอร์อื่น ดังนั้นก็ควร Copy 2 ไฟล์นี้ ของเดิมที่มาพร้อมกับการติดตั้งเก็บไว้ด้วย เวลาจะเปลี่ยนไปใช้ MCU เบอร์อื่นจะได้นำมาวางทับได้เลย ไม่ต้องเสียเวลาติดตั้งโปรแกรมใหม่
4. เปิดโปรแกรม Keil uVision3 ขึ้นมา เลือกที่เมนู Project และเลือก New Project... ดังรูปที่ 1



รูปที่ 1

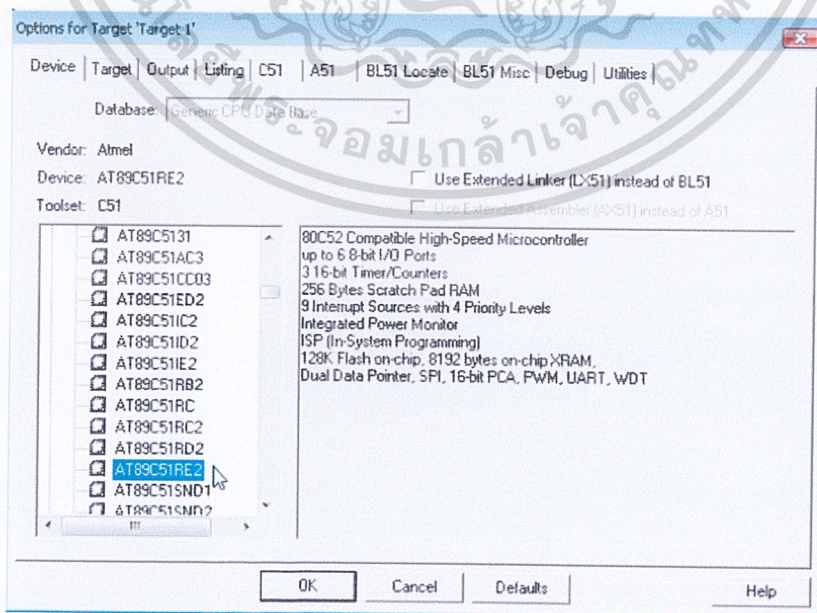
5. จะปรากฏ หน้าต่าง Create New Project ขึ้นมาดังรูปที่ 2 ให้เลือก Folder ที่จะเก็บ Project File และตั้งชื่อ Project File ในตัวอย่างจะตั้งชื่อ Project file คือ LED_Blink และเลือกเก็บที่ Folder Test Port จากนั้น กด Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

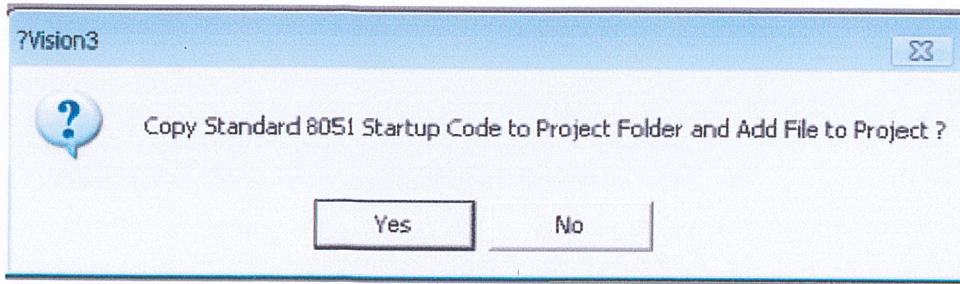


รูปที่ 2

6. หลังจากกด Save แล้ว จะปรากฏหน้าต่าง Select Device for Target 'Target1' ขึ้นมา ดังรูปที่ 3 เพื่อให้ผู้ใช้เลือกเบอร์อุปกรณ์ โดยในช่อง Data base ให้ผู้ใช้เลือกที่ Atmel และเลือกที่เบอร์ AT89C51 RE2 ให้สังเกต เมื่อเลือกแล้วเบอร์จะแสดงในช่อง Device: ด้านบน จากนั้นคลิก OK จะมีหน้าต่าง Pop-Up ขึ้นมาดังรูปที่ 4 ถ้ากด YES จะเป็น करना File Start Up (STARTUP.A51) มาตรฐานของ C51 เข้ามายัง Project หรือกด No ก็ได้ถ้าไม่ต้องการนำ File Start Up เข้ามายัง Project ในตัวอย่างนี้จะขอเลือก No

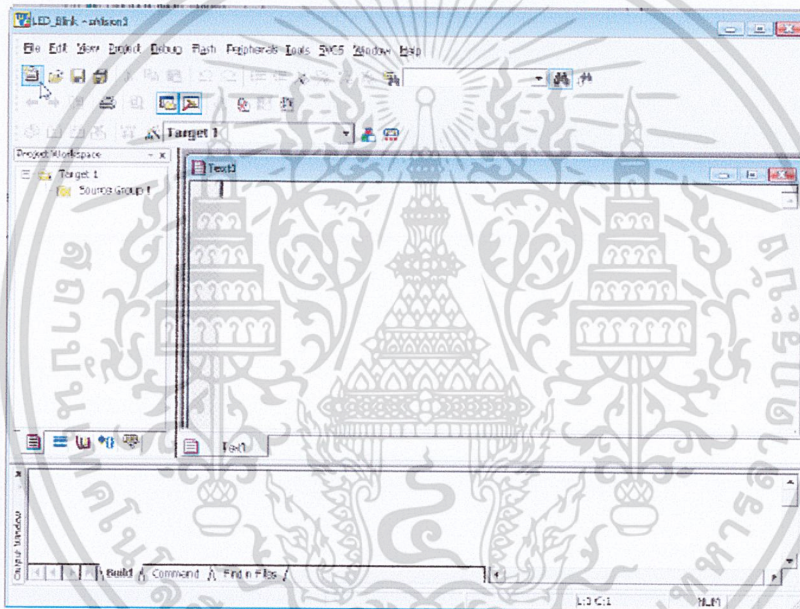


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 3 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



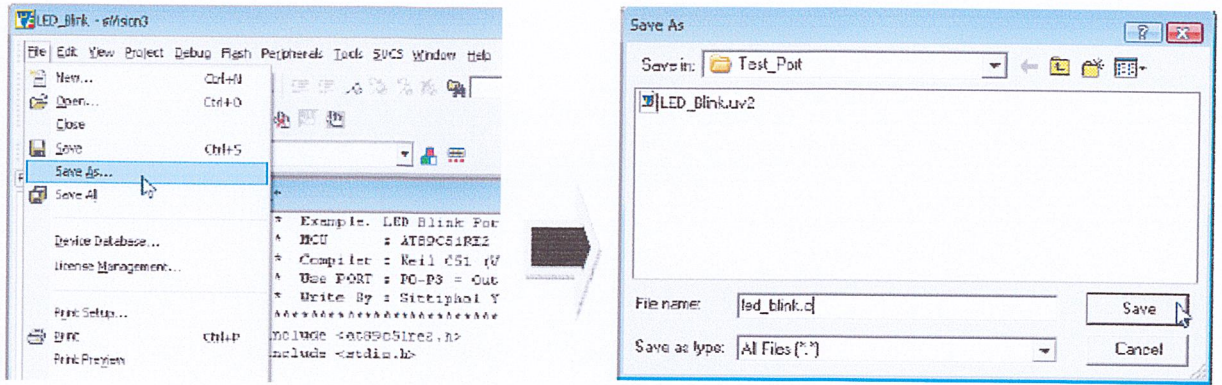
รูปที่ 4

7. จากนั้นให้คลิกที่ไอคอน Create a New File จะได้หน้าต่าง Text1 ออกมาดังรูปที่ 5 ซึ่งจะใช้สำหรับเขียน โปรแกรม



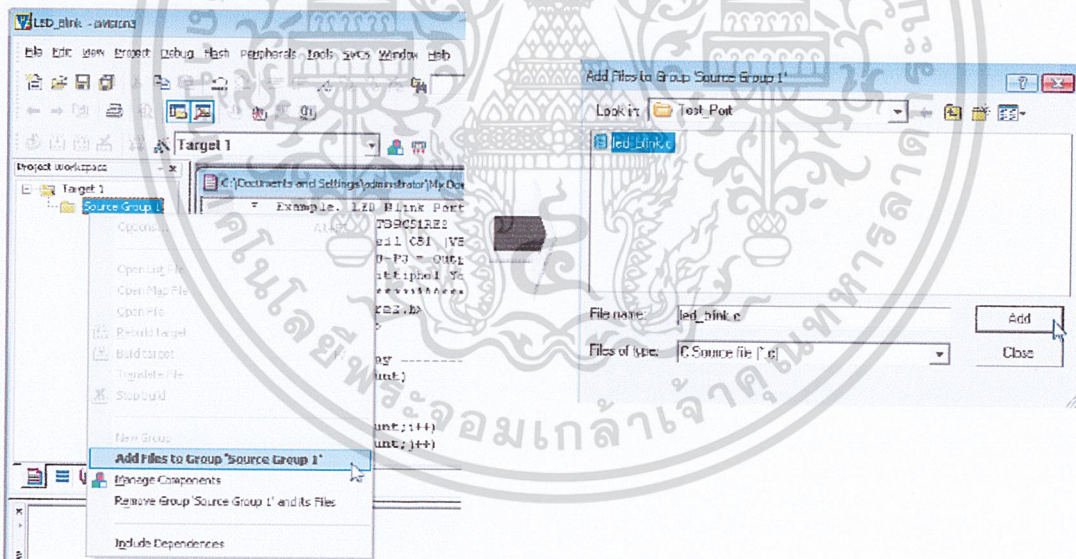
รูปที่ 5

8. ทำการเขียนโปรแกรมตามตัวอย่างด้านล่างลงในหน้าต่าง Text1 เมื่อเขียนโปรแกรมเรียบร้อยแล้วให้ไปที่เมนู File เลือก save as... จะได้หน้าต่างดังรูปที่ 6 ให้เลือก Save ไว้ใน Folder เดียวกับ Project File ในตอนแรก จากนั้นทำการตั้งชื่อ File เป็นนามสกุลจุด C ในตัวอย่างจะตั้งชื่อเป็น led_blink.c และกด Save



รูปที่ 6

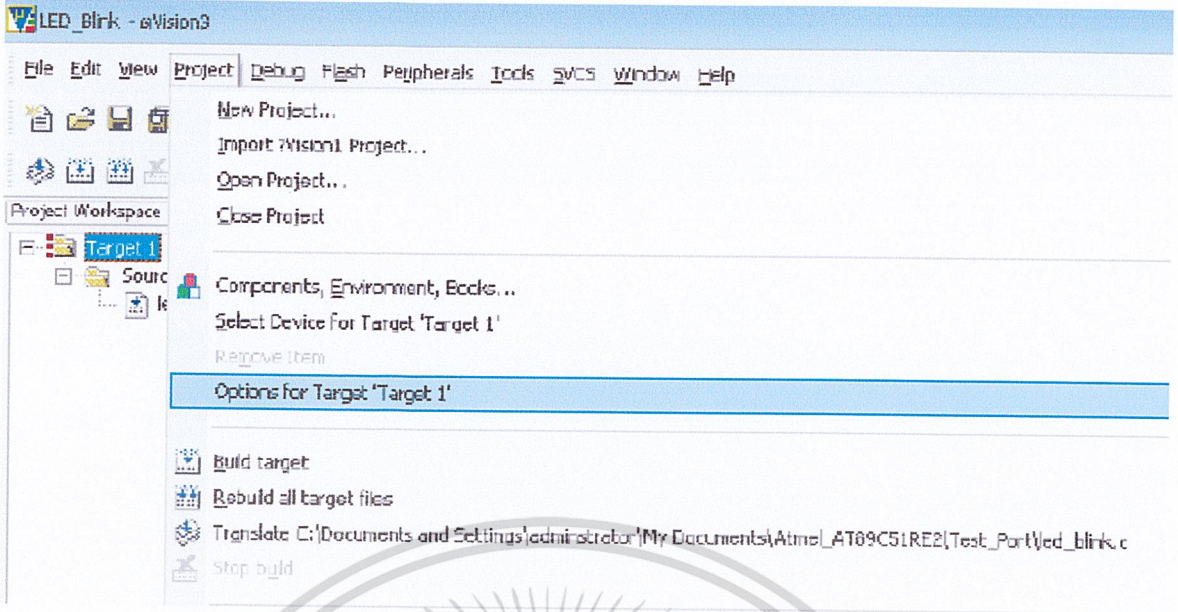
9. เมื่อ Save File แล้วให้ทำการ Add File led_blink.c เข้ามายัง Project โดยให้ Double Click หรือคลิกขวาที่ Folder Source Group 1 ที่อยู่ในหน้าต่างด้านซ้ายมือ จากนั้นเลือก Add Files to Group 'Source Group 1' จะปรากฏหน้าต่างให้ Add file ขึ้นมาดังรูปที่ 7 ให้ผู้ใช้เลือก File led_blink.c แล้วกด Add จากนั้นกด Close เพื่อปิดหน้าต่าง Add File ไฟล์ที่ Add ก็จะมาอยู่ใน Folder Source Group 1



รูปที่ 7

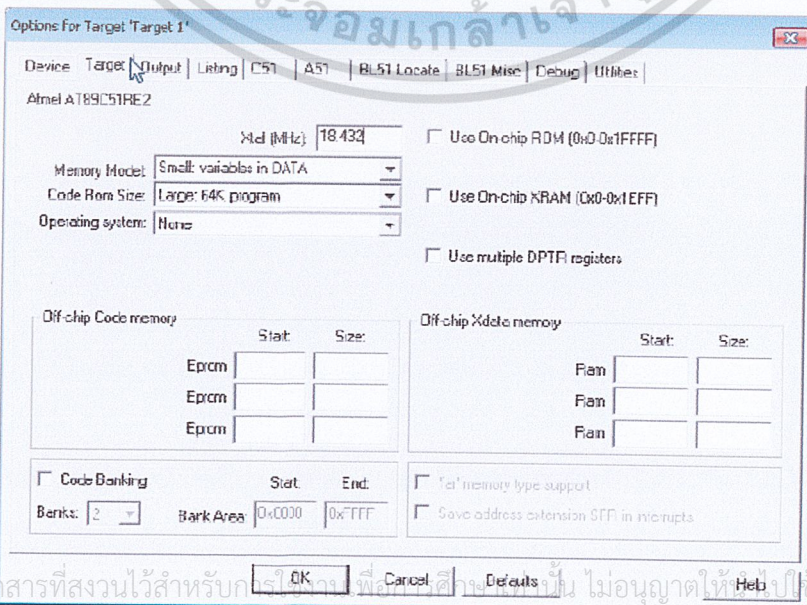
10. เมื่อแอดไฟล์ที่เขียนเข้ามาเรียบร้อยแล้วให้ไปคลิกที่ Folder Target 1 ในช่องด้านซ้ายมือให้เป็นแถบสีน้ำเงินจากนั้นไปที่เมนู Project แล้วเลือก Options for Target 'Target 1' ดังแสดงในรูปที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8

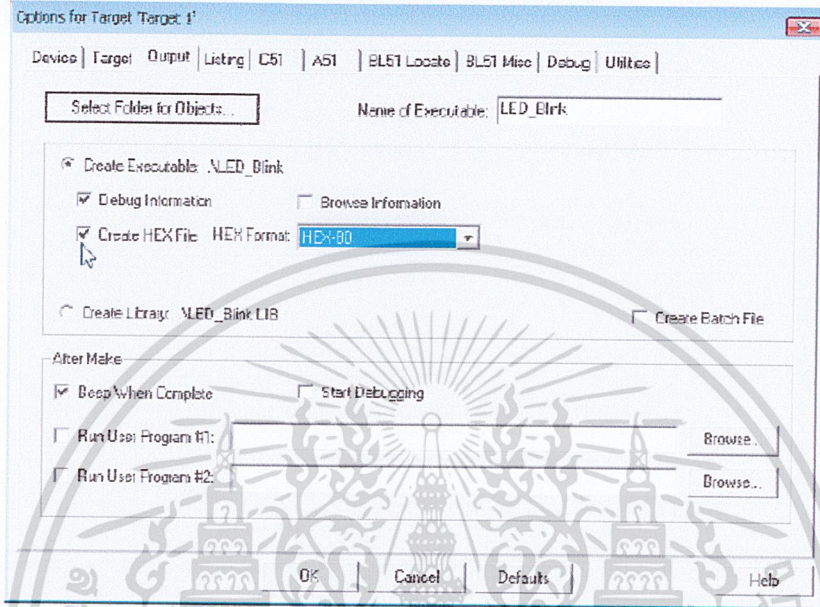
11. จะได้นหน้าต่าง Options for Target 'Target1' ออกมา ให้คลิกที่ TAB Target แล้วทำการกำหนดค่าตามรูปที่ 9 ให้สังเกตในช่อง Memory Model : ซึ่งในช่องนี้จะเป็นการเลือกขนาดของ Ram ที่ใช้ในการเขียนโปรแกรม ถ้าผู้ใช้เลือก Small: variables in DATA ตัวแปรที่ ประกาศในโปรแกรม จะถูกเก็บไว้ยังพื้นที่ RAM ภายใน ซึ่งขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 128 byte , ถ้าเลือก Compact : variable in PDATAขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 256 Byte , และถ้าเลือก Large : variable in XDATA ตัวแปรก็จะถูกเก็บไว้ในพื้นที่ XRAM ดังนั้น ขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 8 K byte ซึ่งแล้วแต่ผู้ใช้จะเลือกใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีสอนเท่านั้น ไม่อนุญาตให้เผยแพร่ไปประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

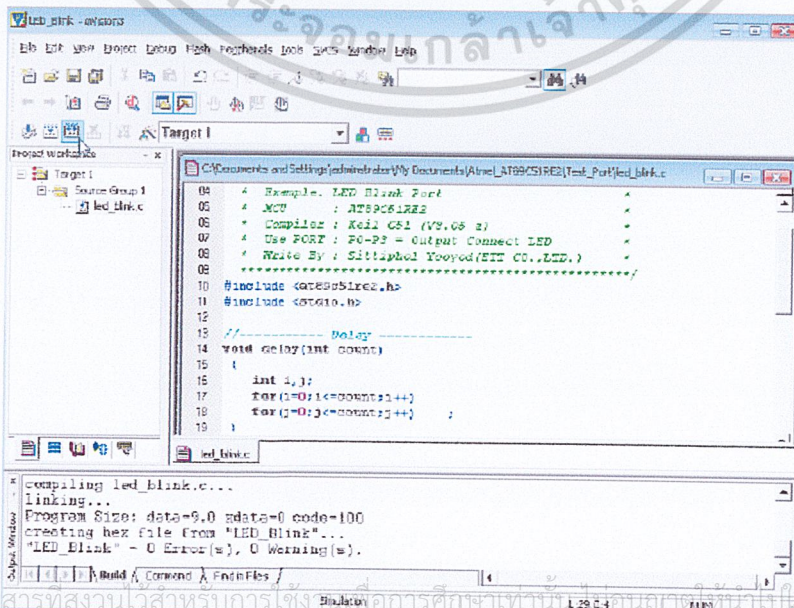
รูปที่ 9

12. คลิกที่ TAB Output แล้วทำการ Tick เครื่องหมายถูกหน้าช่อง Create HEX File ส่วนช่อง HEX Format : ให้เลือก HEX-80 ส่วนช่องอื่นๆก็ให้กำหนดเหมือนในรูป เสร็จแล้วกด OK



รูปที่ 10

13. เมื่อ Set ค่าให้กับ Project เรียบร้อยแล้ว ให้ทำการ Compile โปรแกรมเพื่อตรวจสอบว่ามี Error หรือไม่ โดยคลิกที่ไอคอน Rebuild all target files ถ้าโปรแกรมที่เขียนไม่มีปัญหา ก็จะแสดง Error เป็น 0 อยู่ในหน้าต่างด้านล่าง ดังรูปที่ 11



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 11

14. เมื่อ Compile เรียบร้อย ก็ให้ทำการ Download โปรแกรมลงในไมโครคอนโทรลเลอร์ได้ โดยใช้โปรแกรม Flip คู่มือการ Download ได้ในหัวข้อที่ 4 การ Download โปรแกรม ด้วย FLIP

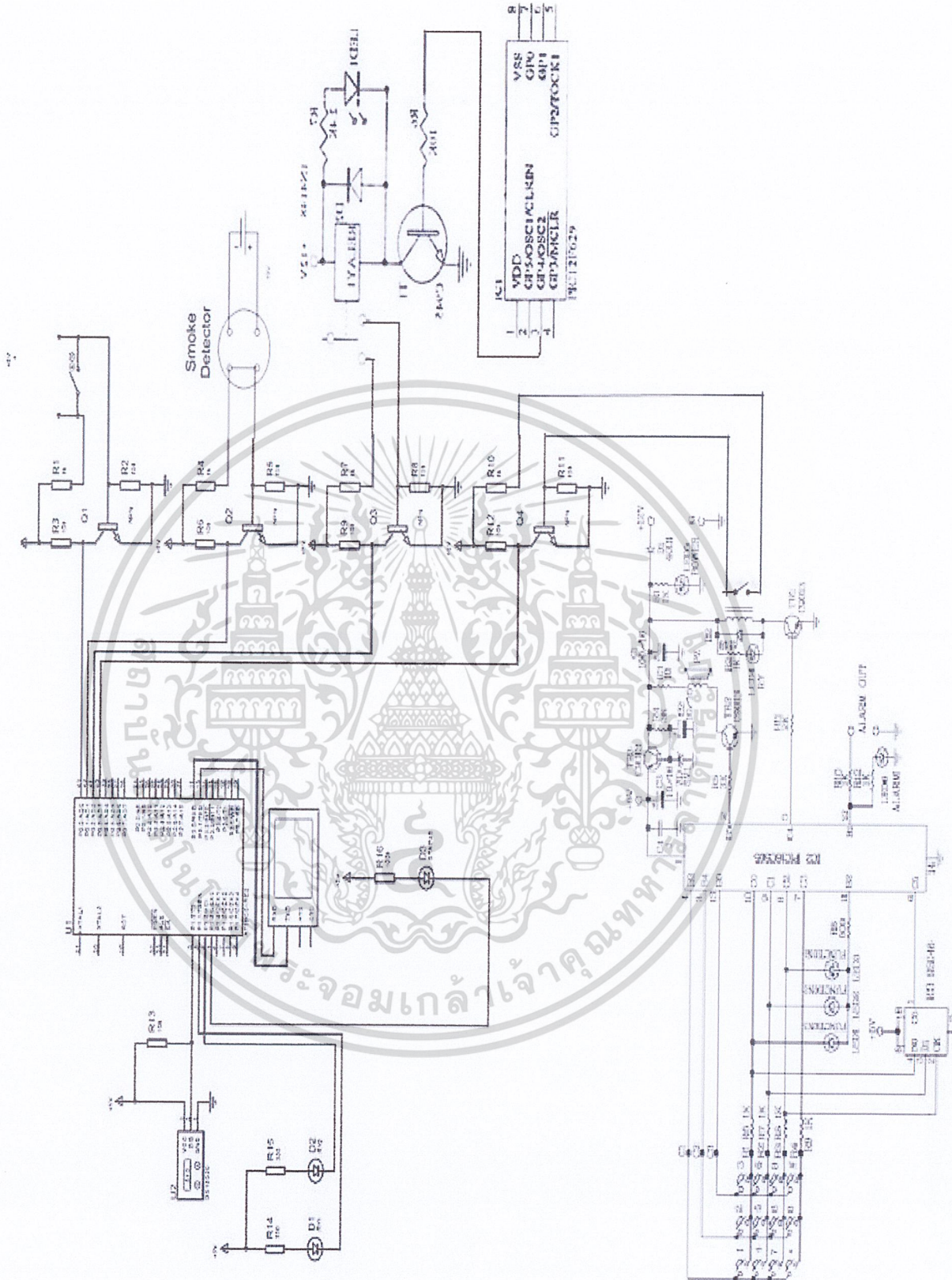


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.
วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค.

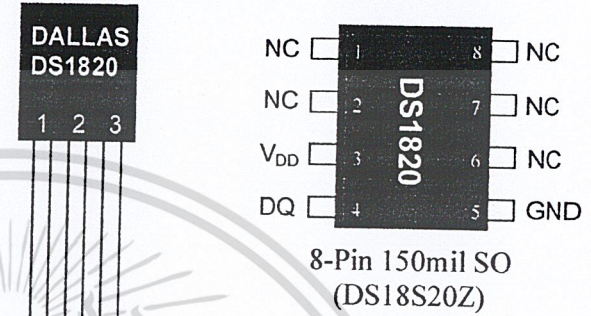
Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES

- Unique 1-Wire® interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an onboard ROM
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C accuracy from -10°C to +85°C
- 9-bit thermometer resolution
- Converts temperature in 750ms (max.)
- User-definable nonvolatile (NV) alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



PIN DESCRIPTION

- GND - Ground
- DQ - Data In/Out
- V_{DD} - Power Supply Voltage
- NC - No Connect

DESCRIPTION

The DS18S20 Digital Thermometer provides 9-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-Wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

1-Wire is a registered trademark of Dallas Semiconductor.

DETAILED PIN DESCRIPTIONS Table 1

8-PIN SOIC*	TO-92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	Data Input/Output Pin. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	3	V _{DD}	Optional V_{DD} Pin. V _{DD} must be grounded for operation in parasite power mode.

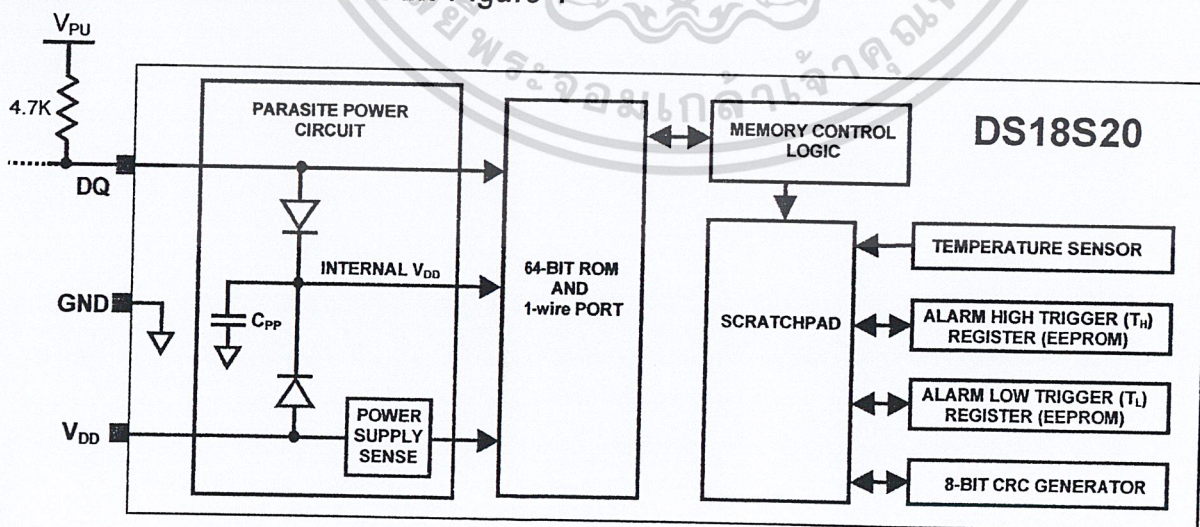
*All pins not specified in this table are "No Connect" pins.

OVERVIEW

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L). The T_H and T_L registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Dallas' exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-WIRE BUS SYSTEM* section of this datasheet.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V_{DD}.

DS18S20 BLOCK DIAGRAM Figure 1

OPERATION — MEASURING TEMPERATURE

The core functionality of the DS18S20 is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to 0.5°C steps. The DS18S20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its idle state. If the DS18S20 is powered by an external supply, the master can issue “read-time slots” (see the *1-WIRE BUS SYSTEM* section) after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18S20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *POWERING THE DS18S20* section of this datasheet.

The DS18S20 output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two’s complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 2 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT REMAIN and COUNT PER °C registers in the scratchpad. Note that the COUNT PER °C register is hard-wired to 16 (10h). After reading the scratchpad, the TEMP_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 2). The extended resolution temperature can then be calculated using the following equation:

$$TEMPERATURE = TEMP_READ - 0.25 + \frac{COUNT_PER_C - COUNT_REMAIN}{COUNT_PER_C}$$

Additional information about high-resolution temperature calculations can be found in *Application Note 105: High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors*.

TEMPERATURE REGISTER FORMAT Figure 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	S	S	S

TEMPERATURE/DATA RELATIONSHIP Table 2

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+85.0°C*	0000 0000 1010 1010	00AAh
+25.0°C	0000 0000 0011 0010	0032h
+0.5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1111	FFFFh
-25.0°C	1111 1111 1100 1110	FFCEh
-55.0°C	1111 1111 1001 0010	FF92h

*The power-on reset value of the temperature register is +85°C

OPERATION — ALARM SIGNALING

After the DS18S20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers $S = 0$ and for negative numbers $S = 1$. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the MEMORY section of this datasheet.

T_H AND T_L REGISTER FORMAT Figure 3

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^5	2^5	2^2	2^1	2^0

Only bits 8 through 1 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than or equal to T_L or higher than T_H , an alarm condition exists and an alarm flag is set inside the DS18S20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18S20s on the bus by issuing an Alarm Search [ECh] command. Any DS18S20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18S20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

POWERING THE DS18S20

The DS18S20 can be powered by an external supply on the V_{DD} pin, or it can operate in “parasite power” mode, which allows the DS18S20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18S20's parasite-power control circuitry, which “steals” power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18S20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18S20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

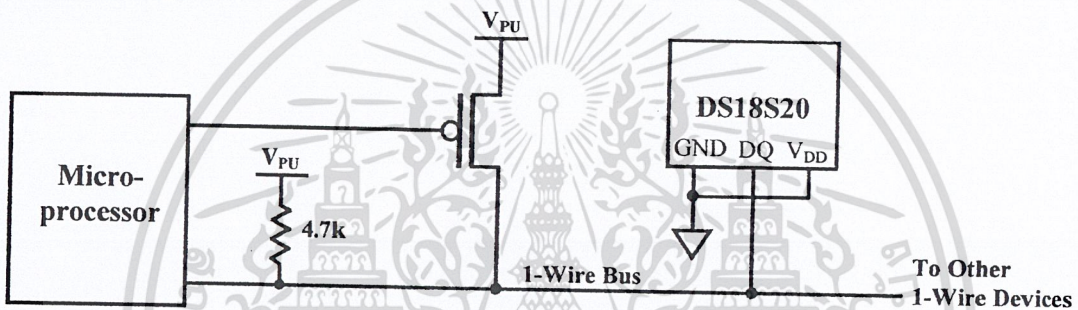
In parasite power mode, the 1-Wire bus and C_{PP} can provide sufficient current to the DS18S20 for most operations as long as the specified timing and voltage requirements are met (refer to the *DC ELECTRICAL CHARACTERISTICS* and the *AC ELECTRICAL CHARACTERISTICS* sections of this data sheet). However, when the DS18S20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by C_{PP} . To assure that the DS18S20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-Wire bus must be switched to the strong pullup within 10 μ s (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{conv}) or data transfer ($t_{wr} = 10$ ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18S20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

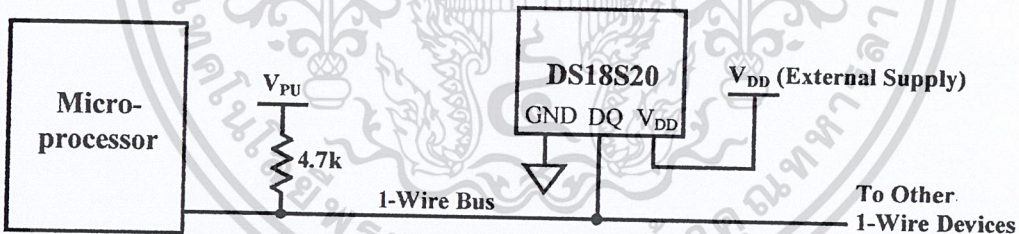
The use of parasite power is not recommended for temperatures above 100°C since the DS18S20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18S20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18S20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read-time slot”. During the read-time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

SUPPLYING THE PARASITE-POWERED DS18S20 DURING TEMPERATURE CONVERSIONS Figure 4



POWERING THE DS18S20 WITH AN EXTERNAL SUPPLY Figure 5



64-BIT LASERED ROM CODE

Each DS18S20 contains a unique 64-bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20’s 1-Wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the *CRC GENERATION* section. The 64-bit ROM code and associated ROM function control logic allow the DS18S20 to operate as a 1-Wire device using the protocol detailed in the *1-WIRE BUS SYSTEM* section of this datasheet.

64-BIT LASERED ROM CODE Figure 6

8-BIT CRC		48-BIT SERIAL NUMBER			8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB	

MEMORY

The DS18S20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L). Note that if the DS18S20 alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the *DS18S20 FUNCTION COMMANDS* section.

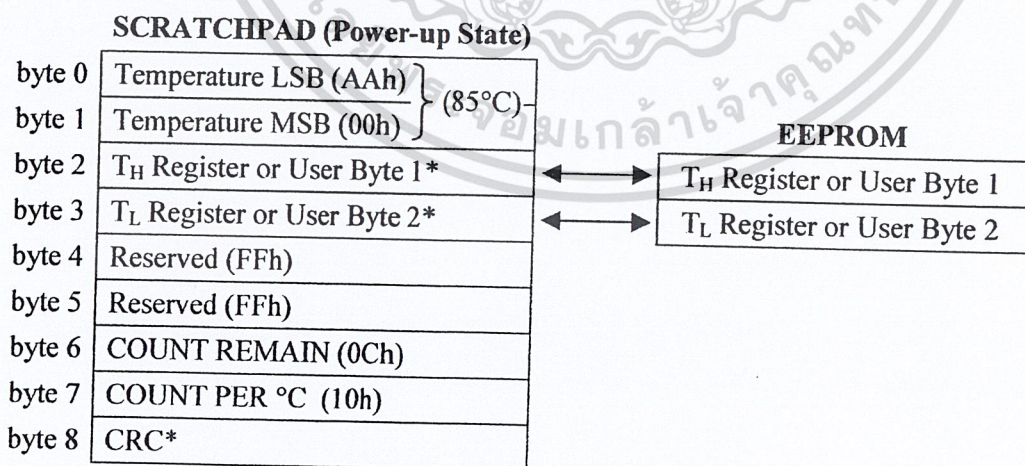
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Bytes 4 and 5 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read. Bytes 6 and 7 contain the COUNT REMAIN and COUNT PER °C registers, which can be used to calculate extended resolution results as explained in the *OPERATION — MEASURING TEMPERATURE* section.

Byte 8 of the scratchpad is read-only and contains the cyclic redundancy check (CRC) code for bytes 0 through 7 of the scratchpad. The DS18S20 generates this CRC using the method described in the *CRC GENERATION* section.

Data is written to bytes 2 and 3 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18S20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H and T_L data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E² [B8h] command. The master can issue “read-time slots” (see the *1-WIRE BUS SYSTEM* section) following the Recall E² command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

DS18S20 MEMORY MAP Figure 7



*Power-up state depends on value(s) stored in EEPROM

CRC GENERATION

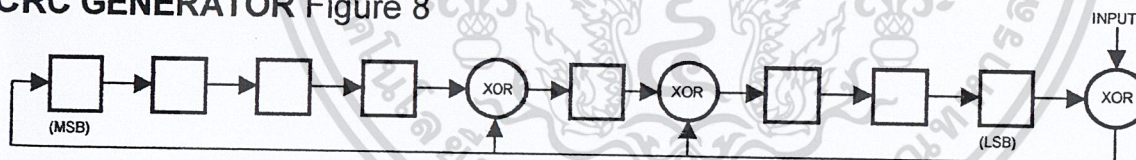
CRC bytes are provided as part of the DS18S20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18S20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18S20 that prevents a command sequence from proceeding if the DS18S20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18S20 using the polynomial generator shown in Figure 8. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18S20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Dallas 1-Wire cyclic redundancy check is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products*.

CRC GENERATOR Figure 8



1-WIRE BUS SYSTEM

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18S20 is always a slave. When there is only one slave on the bus, the system is referred to as a “single-drop” system; the system is “multidrop” if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

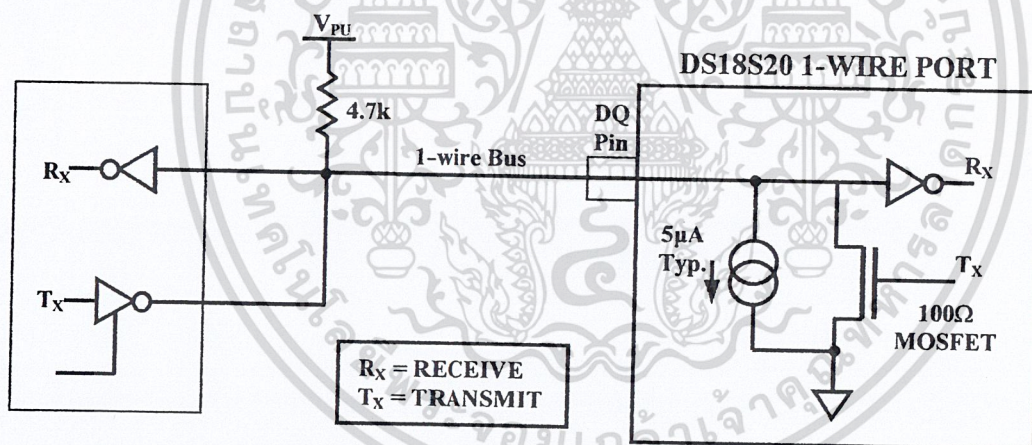
The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18S20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 9.

The 1-Wire bus requires an external pullup resistor of approximately $5k\Omega$; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than $480\mu s$, all components on the bus will be reset.

HARDWARE CONFIGURATION Figure 9



TRANSACTION SEQUENCE

The transaction sequence for accessing the DS18S20 is as follows:

Step 1. Initialization

Step 2. ROM Command (followed by any required data exchange)

Step 3. DS18S20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18S20 is accessed, as the DS18S20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

INITIALIZATION

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18S20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the *1-WIRE SIGNALING* section.

ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18S20 function command. A flowchart for operation of the ROM commands is shown in Figure 14.

SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the *iButton® Book of Standards* at www.ibutton.com/ibuttons/standard.pdf. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18S20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18S20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. Refer to the *OPERATION — ALARM SIGNALING* section for an explanation of alarm flag operation.

DS18S20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18S20 with which it wishes to communicate, the master can issue one of the DS18S20 function commands. These commands allow the master to write to and read from the DS18S20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18S20 function commands, which are described below, are summarized in Table 4 and illustrated by the flowchart in Figure 15.

CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{conv}) as described in the *POWERING THE DS18S20* section. If the DS18S20 is powered by an external supply, the master can issue read-time slots after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

WRITE SCRATCHPAD [4Eh]

This command allows the master to write 2 bytes of data to the DS18S20's scratchpad. The first byte is written into the T_H register (byte 2 of the scratchpad), and the second byte is written into the T_L register (byte 3 of the scratchpad). Data must be transmitted least significant bit first. Both bytes **MUST** be written before the master issues a reset, or the data may be corrupted.

READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad T_H and T_L registers (bytes 2 and 3) to EEPROM. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the *POWERING THE DS18S20* section.

RECALL E² [B8h]

This command recalls the alarm trigger values (T_H and T_L) from EEPROM and places the data in bytes 2 and 3, respectively, in the scratchpad memory. The master device can issue read-time slots following the Recall E² command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

READ POWER SUPPLY [B4h]

The master device issues this command followed by a read-time slot to determine if any DS18S20s on the bus are using parasite power. During the read-time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. Refer to the *POWERING THE DS18S20* section for usage information for this command.

DS18S20 FUNCTION COMMAND SET Table 4

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued	Notes
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18S20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2 and 3 (T_H and T_L).	4Eh	Master transmits 2 data bytes to DS18S20.	3
Copy Scratchpad	Copies T_H and T_L data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H and T_L data from EEPROM to the scratchpad.	B8h	DS18S20 transmits recall status to master.	
Read Power Supply	Signals DS18S20 power supply mode to the master.	B4h	DS18S20 transmits supply status to master.	

NOTES:

- 1) For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
- 2) The master can interrupt the transmission of data at any time by issuing a reset.
- 3) Both bytes must be written before a reset is issued.

1-WIRE SIGNALING

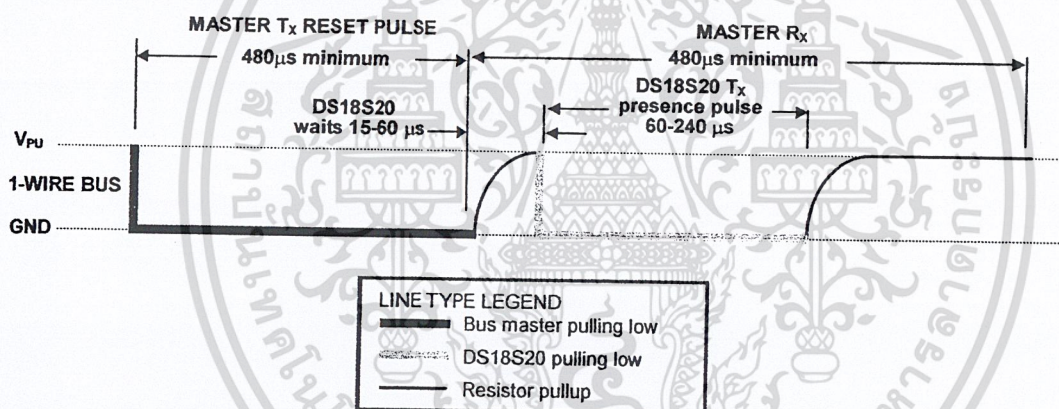
The DS18S20 uses a strict 1-Wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

INITIALIZATION PROCEDURE: RESET AND PRESENCE PULSES

All communication with the DS18S20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18S20. This is illustrated in Figure 10. When the DS18S20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_x) the reset pulse by pulling the 1-Wire bus low for a minimum of $480\mu\text{s}$. The bus master then releases the bus and goes into receive mode (R_x). When the bus is released, the 5k pullup resistor pulls the 1-Wire bus high. When the DS18S20 detects this rising edge, it waits $15\mu\text{s}$ to $60\mu\text{s}$ and then transmits a presence pulse by pulling the 1-Wire bus low for $60\mu\text{s}$ to $240\mu\text{s}$.

INITIALIZATION TIMING Figure 10



READ/WRITE TIME SLOTS

The bus master writes data to the DS18S20 during write time slots and reads data from the DS18S20 during read-time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

WRITE TIME SLOTS

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18S20 and a Write 0 time slot to write a logic 0 to the DS18S20. All write time slots must be a minimum of $60\mu\text{s}$ in duration with a minimum of a $1\mu\text{s}$ recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see Figure 11).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within $15\mu\text{s}$. When the bus is released, the 5k pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least $60\mu\text{s}$). The DS18S20 samples the 1-Wire bus during a window that lasts from $15\mu\text{s}$ to $60\mu\text{s}$ after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18S20. If the line is low, a 0 is written to the DS18S20.

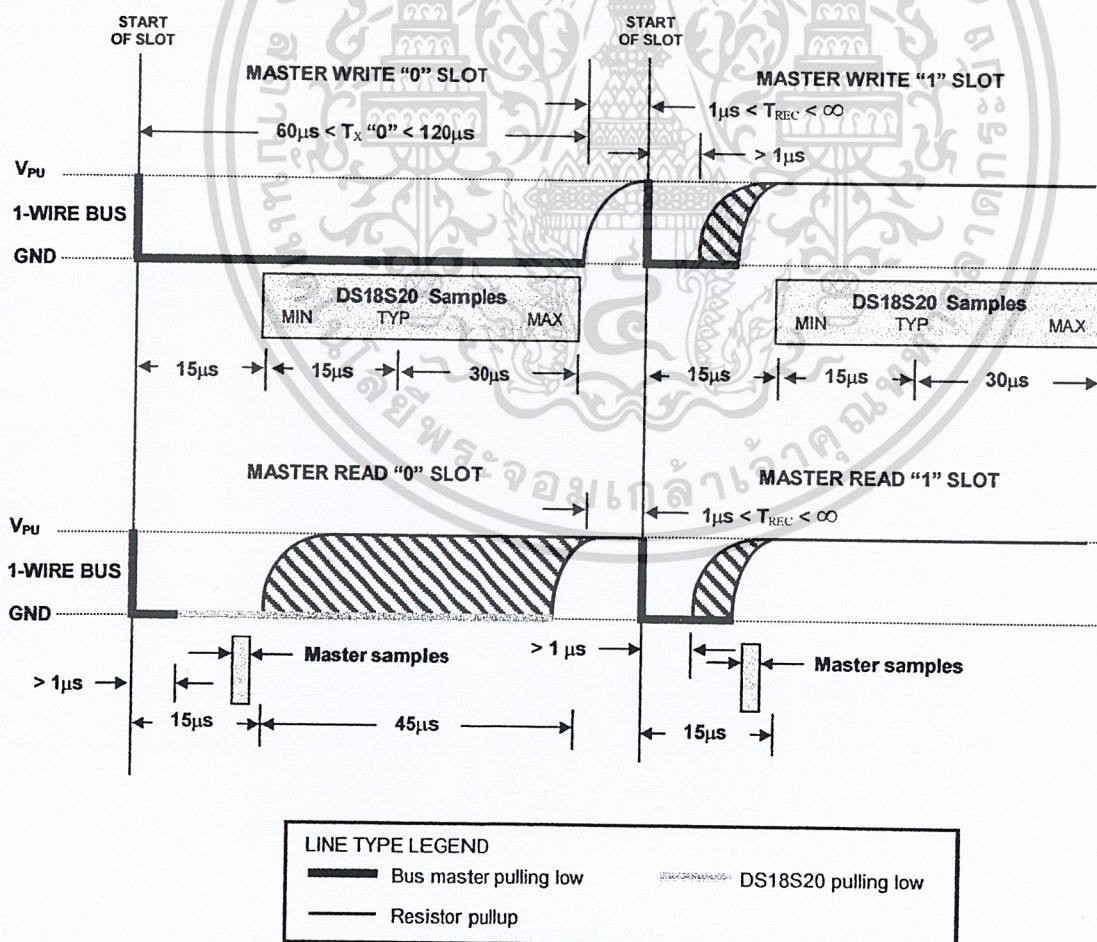
READ-TIME SLOTS

The DS18S20 can only transmit data to the master when the master issues read-time slots. Therefore, the master must generate read-time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18S20 can provide the requested data. In addition, the master can generate read-time slots after issuing Convert T [44h] or Recall E² [B8h] commands to find out the status of the operation as explained in the DS18S20 FUNCTION COMMAND section.

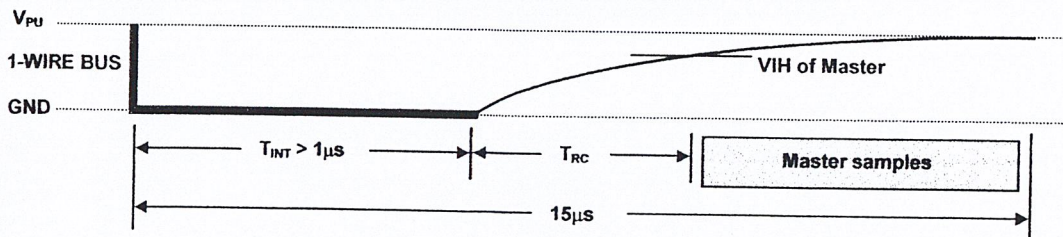
All read-time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read-time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see Figure 11). After the master initiates the read-time slot, the DS18S20 will begin transmitting a 1 or 0 on bus. The DS18S20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18S20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output data from the DS18S20 is valid for 15µs after the falling edge that initiated the read-time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

Figure 12 illustrates that the sum of T_{INIT} , T_{RC} , and T_{SAMPLE} must be less than 15µs for a read-time slot. Figure 13 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read-time slots towards the end of the 15µs period.

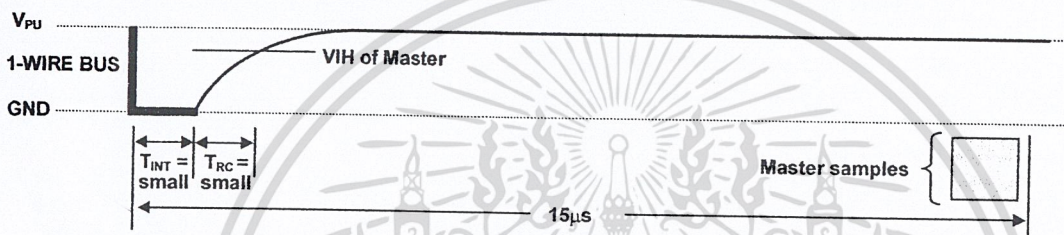
READ/WRITE TIME SLOT TIMING DIAGRAM Figure 11



DETAILED MASTER READ 1 TIMING Figure 12

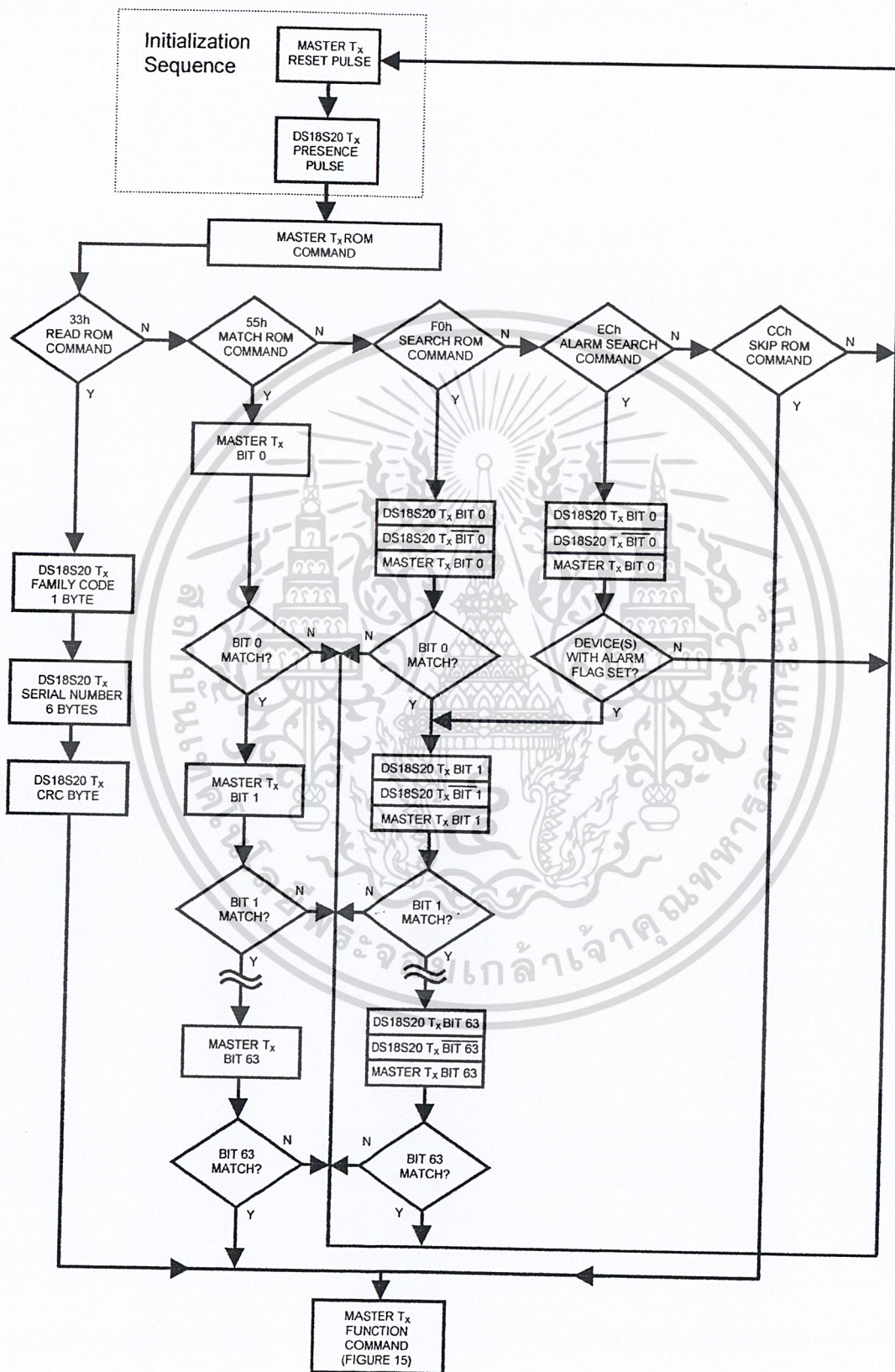


RECOMMENDED MASTER READ 1 TIMING Figure 13

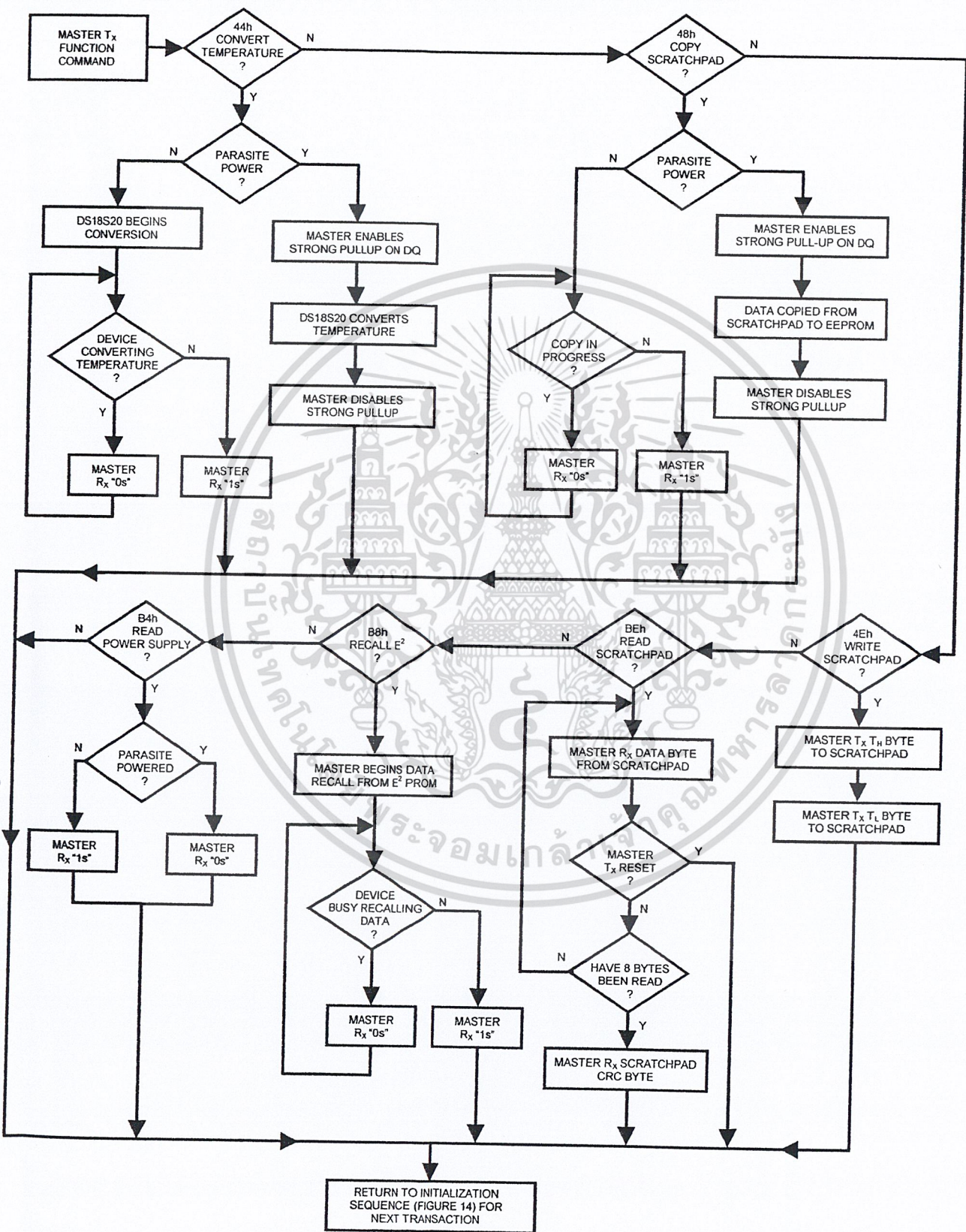


LINE TYPE LEGEND
 — Bus master pulling low
 — Resistor pullup

ROM COMMANDS FLOW CHART Figure 14



DS18S20 FUNCTION COMMANDS FLOW CHART Figure 15



DS18S20 OPERATION EXAMPLE 1

In this example there are multiple DS18S20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18S20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18S20 ROM code.
TX	44h	Master issues Convert T command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t_{conv}).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18S20 ROM code.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18S20 OPERATION EXAMPLE 2

In this example there is only one DS18S20 on the bus and it is using parasite power. The master writes to the T_H and T_L registers in the DS18S20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	4Eh	Master issues Write Scratchpad command.
TX	2 data bytes	Master sends two data bytes to scratchpad (T_H and T_L)
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	48h	Master issues Copy Scratchpad command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

DS18S20 OPERATION EXAMPLE 3

In this example there is only one DS18S20 on the bus and it is using parasite power. The bus master initiates a temperature conversion then reads the DS18S20 scratchpad and calculates a higher resolution result using the data from the temperature, COUNT REMAIN and COUNT PER °C registers.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
TR	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	44h	Master issues Convert T command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t_{conv}).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. The master also calculates the TEMP_READ value and stores the contents of the COUNT REMAIN and COUNT PER °C registers.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
-	-	CPU calculates extended resolution temperature using the equation in the <i>OPERATION — MEASURING TEMPERATURE</i> section of this datasheet.

RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS18S20. These notes can be obtained from the Dallas Semiconductor "Application Note Book," via the Dallas website at <http://www.dalsemi.com/>, or through our faxback service at (214) 450-0441.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product

Application Note 55: Extending the Contact Range of Touch Memories

Application Note 74: Reading and Writing Touch Memories via Serial Interfaces

Application Note 104: Minimalist Temperature Control Demo

Application Note 105: High-Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors

Application Note 106: Complex MicroLANs

Application Note 108: MicroLAN — In the Long Run

Application Note 162: Interfacing the DS18X20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Sample 1-Wire subroutines that can be used in conjunction with AN74 can be downloaded from the Dallas website or anonymous FTP Site.

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +6.0V
Operating Temperature Range	-55°C to +125°C
Storage Temperature Range	-55°C to +125°C
Solder Temperature	See IPC/JEDEC J-STD-020A
Reflow Oven Temperature	+220°C

*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD} = 3.0V$ to $5.5V$)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1, 2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic Low	V_{IL}		-0.3		+0.8	V	1, 4, 5
Input Logic High	V_{IH}	Local Power	+2.2		The lower of 5.5 or $V_{DD} + 0.3$	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	$V_{IO}=0.4V$	4.0			mA	1
Standby Current	I_{DDs}			750	1000	nA	7, 8
Active Current	I_{DD}	$V_{DD}=5V$		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		μA	10
Drift				±0.2		°C	11

NOTES:

- All voltages are referenced to ground.
- The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU} . In order to meet the V_{IH} spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
- See typical performance curve in Figure 16
- Logic low voltages are specified at a sink current of 4mA.
- To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
- Logic high voltages are specified at a source current of 1mA.
- Standby current specified up to 70°C. Standby current typically is 3μA at 125°C.
- To minimize I_{DDs} , DQ should be within the following ranges: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
- Active current refers to supply current during active temperature conversions or EEPROM writes.
- DQ line is high ("hi-Z" state).
- Drift data is based on a 1000 hour stress test at 125°C with $V_{DD} = 5.5V$.

AC ELECTRICAL CHARACTERISTICS: NV MEMORY(-55°C to +100°C; $V_{DD} = 3.0V$ to 5.5V)

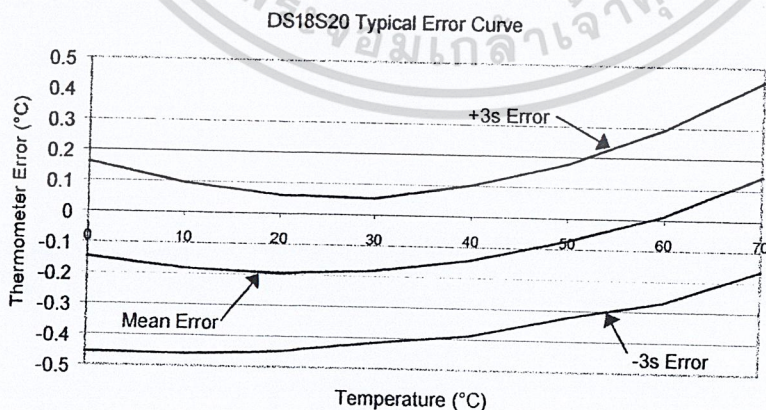
PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{wr}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EDR}	-55°C to +55°C	10			years

AC ELECTRICAL CHARACTERISTICS(-55°C to +125°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}				750	ms	1
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	μs	
Time Slot	t_{SLOT}		60		120	μs	1
Recovery Time	t_{REC}		1			μs	1
Write 0 Low Time	t_{LOW0}		60		120	μs	1
Write 1 Low Time	t_{LOW1}		1		15	μs	1
Read Data Valid	t_{RDV}				15	μs	1
Reset Time High	t_{RSTH}		480			μs	1
Reset Time Low	t_{RSTL}		480			μs	1, 2
Presence Detect High	t_{PDHIGH}		15		60	μs	1
Presence Detect Low	t_{PDLOW}		60		240	μs	1
Capacitance	$C_{IN/OUT}$				25	pF	

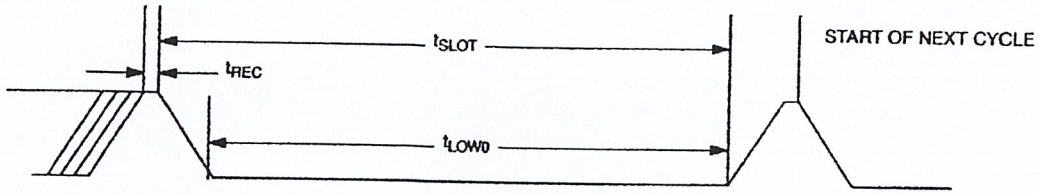
NOTES:

- 1) Refer to timing diagrams in Figure 17.
- 2) Under parasite power, if $t_{RSTL} > 960\mu s$, a power on reset may occur.

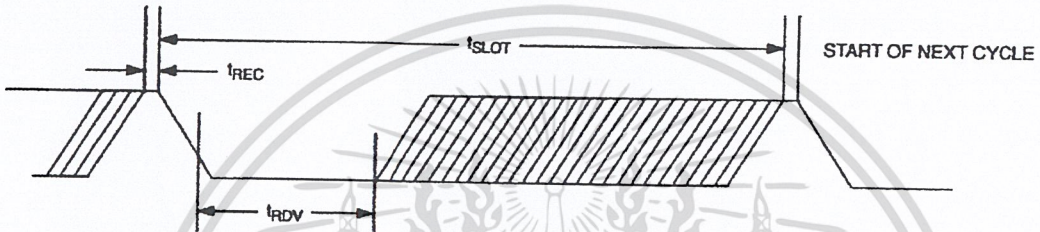
TYPICAL PERFORMANCE CURVE Figure 16

TIMING DIAGRAMS Figure 17

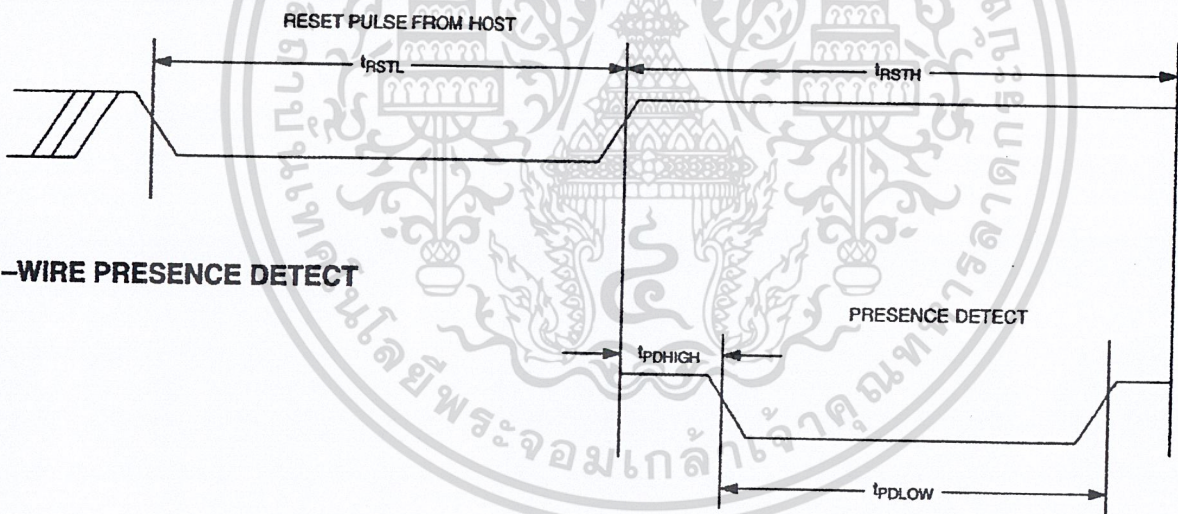
1-WIRE WRITE ZERO TIME SLOT



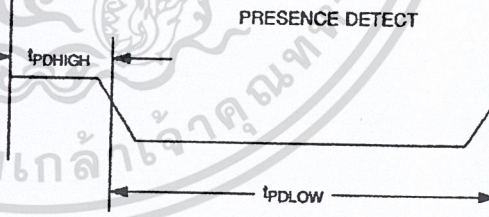
1-WIRE READ ZERO TIME SLOT























1-WIRE RESET PULSE



1-WIRE PRESENCE DETECT



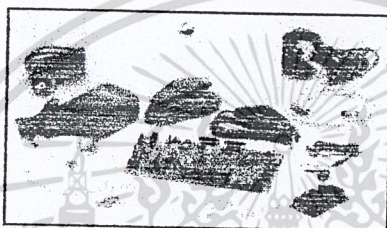
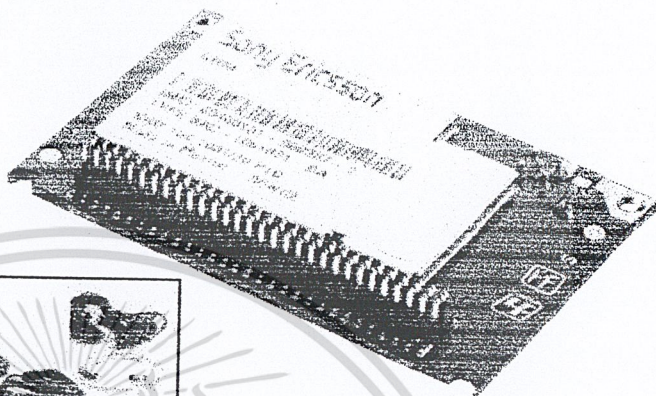
Product Features:

-  **Compact Size**
 -  **Quad Band**
850-900+
1800-1900
 -  **GSM/GPRS Class 10**
 -  **ARM-9 Processor**
 -  **SIM Holder**
Optional
 -  **2x UART Connectivity**
Asynchronous Universal Asynchronous Receiver/Transmitter
 -  **Integrated TCP/IP Stack**
 -  **Embedded Applications**
 -  **RoHS Compliant**
 -  **Fully Type Approved**
CE/FCC/TELEC
-
- Broad Spectrum:**
-  **Emergency Services**
 -  **Fleet & Freight**
 -  **Healthcare**
 -  **Industrial Systems**
 -  **Information Display**
 -  **Metering & Monitoring**
 -  **Power & Energy**
 -  **Security & Surveillance**
 -  **Traffic Control**
 -  **Vending & Retail**



Sony Ericsson

GR64 GSM/GPRS Radio Device



Universal Developer's Kit

Benefits

- **Functionality comparable to the GM47/GM48 and GR47/GR48 Radio Devices for Legacy users**
- **Powerful and fast enabling complex and time critical applications**
- **Minimize total solution cost by embedding application directly onto GR64**

Optimized for global M2M communications, the GR64 is a Quad Band GSM/GPRS Class 10 compact Radio Device. With functionality comparable to the Sony Ericsson GM47/GM48 and GR47/GR48 Radio Devices, the GR64 offers a broad range of voice and data features. Its intrinsic TCP/IP stack enables customers to make effective use of GPRS. The GR64 is configurable and possesses an extended range of input/output capabilities.

With its exceptionally powerful processor and enhanced memory, the GR64 is ideal for a broad spectrum of M2M applications including fleet and asset management, POS, vending, security, metering and other solutions.

Sony Ericsson's Universal Developer's Kit can be used with the GR64 to simplify application development. The kit provides everything required to design and develop innovative applications and to bring them to market in rapid time.

The GR64 is compliant with EU RoHS directives on the environment.

Expanding Wireless Possibilities



Expanding Wireless Possibilities

เอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GR64 GSM/GPRS Radio Device

Radio Features

- Quad Band GSM/GPRS Radio Device
 - GSM 850/900MHz power class 4 (33dBm)
 - GSM 1800/1900MHz power class 1 (30dBm)
- Mobile class B
- Extended measurement reporting
- Compliant with 3GPP release 99 protocol stack
- Battery charging support
- Overall dimensions (excluding connectors):
 - 50 x 33 x 3.2mm
- Weight: 9g

Environmental

- Power supply: 3.2 – 4.5VDC
- Temperature range:
 - -30°C to +75°C (operational functionality)
 - -40°C to +85°C (storage temperature)

Interfaces

- Dual microphone inputs and speaker outputs
- Up to 12 x programmable general purpose I/Os
- 4 x ADC inputs
- 1 x buzzer output
- Real time clock with alarm output
- 2 x UARTs
 - Serial baud rate from 1200 to 460,800 bits/s
 - Auto baud
 - Parity, data and stop bit programmable
- USB 2.0 full speed end point compliant (optional)
- MMCX RF connector
- 60-pin system connector as on GM47/GM48 and GR47/GR48
- Control by AT commands (GSM 27.005, 27.007 plus proprietary commands)

Data/GPRS Features

- GPRS class 10
 - Multiple simultaneous PDP contexts
 - GPRS coding schemes CS1-CS4
- Transparent and non-transparent CSD up to 9.6kbps
- Modem type; V21, V22, V23, V22bis, V26ter, V32, V34, V24
- V42bis compression
- GSM supplementary services supported
- GSM 27.010 multiplexing protocol
- USSD

Short Message Service (SMS) Features

- Text and PDU
- Point to point (MT/MO)
- Cell broadcast

Voice Features

- Telephony
- Emergency calls
- GSM vocoder support:
 - Full Rate, Enhanced Full Rate, Half Rate and Adaptive Multi Rate (FR/EFR/HR/AMR)
- Noise suppression and echo cancellation
- Sidetone/microphone amplification
- PCM audio
- Five customizable profiles enabling tuning of the audio interface

Embedded Applications (Optional)

- Embedded script interpreter
- Extensive API-library
- Multiple UART access
- Embedded application script downloads over the air using GPRS
- Our M2mpower software facilitates the development of cost effective wireless M2M applications

Internet Protocols

- TCP/UDP/IP protocol stack
- Extensive AT command access to TCP/IP stack
- Multi sockets (up to 15) through AT commands
- Non blocking listening/server capability
- IPv4 protocol
- Dynamic & static IP address allocation
- PPP protocol (PAP)

Additional Features

- SIM application toolkit class 2
- 1.8/3V SIM interface with SIM detection
- Optional SIM holder
- Dual SIM support with switching
- LED output
- Keypad matrix (4x4)
- Optional integrated level shifters

Universal Developer's Kit

- Our Universal Developer's Kit is available for rapid evaluation of products and applications on a limited scale

© Copyright 2006 Sony Ericsson Mobile Communications (USA) Inc. Reproduction in whole or in part without written permission is prohibited. All rights reserved.

Sony Ericsson, M2M Expanding Wireless Opportunities are registered trademarks of Sony Ericsson Mobile Communications.

All other trademarks are trademarks of their respective owners.

The information contained herein is for information purposes only. The datasheet contents are based upon sources which we believe to be reliable, but no representation, express or implied, is made with respect to the accuracy, completeness or reliability of the information or opinions in this datasheet. The information contained herein is subject to change without notice, and we assume no responsibility to update the information in our reports and no contractual commitment is formed simply by reading the datasheet. Sony Ericsson does not accept responsibility or liability for any direct, indirect, special, consequential or other losses or damages of whatsoever kind arising out of any information contained in the datasheet or your use thereof.

The products and services discussed in this datasheet may not be available in your particular country or locality, or at all. The reference to such products and services is for information purposes only and does not imply or warrant that these products or services will be available in your particular country, location or elsewhere.

L2T 123 8126 R1E

Sony Ericsson

mail: m2minfo@SonyEricsson.com

website: www.SonyEricsson.com/m2m

m2m

Expanding Wireless Possibilities

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้...
อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้