สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

DEVICE FOR HELPING VISUALLY IMPAIRED USER TO CROSS THE ROAD

T104282

โดย

นางสาวปทุมรัตน์ อุปวรพงษ์

นายศุภชัย วงษ์ม่วง

นายสุพัชร เหลี่ยงสกุล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน 104282

วัน,เดือน,ปี 30 ต.ค. 2552

.b.....

.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVICE FOR HELPING VISUALLY IMPAIRED USER TO CROSS THE ROAD

BY

MS. PATOOMRAT AUPPAWARAPONG

MR. SUPACHAI WONGMUONG

MR. SUPAT LEANGSAKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ อุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

Device for helping visually impaired user to cross the road

นักศึกษา	นางสาวปทุมรัตน์	อุปวรพงษ์	รหัสนักศึกษา	48010497
	นายศุภชัย	วงษ์ม่วง	รหัสนักศึกษา	48010909
	นายสุพัชร	เหลี่ยงสกุล	รหัสนักศึกษา	48011003

อาจารย์ที่ปรึกษา อาจารย์สรพงษ์ วชิรรัตนพรกุล

ระดับการศึกษา ปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

ภาควิชา วิศวกรรมสารสนเทศ

ปีการศึกษา 2551

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(อาจารย์สรพงษ์ วชิรรัตนพรกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ อุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

Device for helping visually impaired user to cross the road

นักศึกษา	นางสาวปทุมรัตน์	อุปวรพงษ์	รหัสนักศึกษา	48010497
	นายศุภชัย	วงษ์ม่วง	รหัสนักศึกษา	48010909
	นายสุพัชร	เหลี่ยงสกุล	รหัสนักศึกษา	48011003

อาจารย์ที่ปรึกษา อาจารย์สรพงษ์ วชิรรัตนพรกุล

ระดับการศึกษา ปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

ภาควิชา วิศวกรรมสารสนเทศ

ปีการศึกษา 2551

บทคัดย่อ

โครงงานนี้เป็นการใช้ชุดโมดูลการสื่อสารแบบไร้สายเพื่อช่วยเหลือผู้พิการทางสายตาและคนชราในการข้ามถนน ซึ่งผู้พิการทางสายตา หรือคนชราจะพกตัวส่งสัญญาณไร้สายติดตัว เมื่อเดินผ่านบริเวณเสาไฟสัญญาณจราจรที่มีการติดตั้งอุปกรณ์ตัวนี้อยู่ อุปกรณ์ตัวนี้จะทำการขอไฟสัญญาณจราจรเพื่อให้ผู้พิการ หรือคนชราข้ามถนน และยังมีลำโพงกระจายเสียง เพื่อช่วยให้ทราบว่าสามารถข้ามถนนได้เมื่อใด และเหลือเวลาในการข้ามอีกเท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Device for helping visually impaired user to cross the road
Student	Miss. Patoomrat Auppaworapong ID.48010497
	Mr. Supachai Wongmuong ID.48010909
	Mr.Supat Leangsakul ID.48011003
Advisor	Mr. Sorapong Wachirarattanapornkul
Graduate Level	Bachelor Degree of Information Engineering
Department	Information Engineering
Academic Year	2551

ABSTRACT

This project uses the wireless device for helping blinder and elder to cross the road. They will keep wireless sensor near them when they walking through traffic light area which has installed with this wireless device. It will send a request to traffic light to give blinder and elder a chance in cross road moreover, this wireless sensor has a loudspeaker to inform whether they can cross the road or still waiting for sign and the remaining time for crossing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีขอขอบพระคุณอาจารย์สรพงษ์ วชิรรัตนพรกุล ที่ให้คำแนะนำ และความช่วยเหลือที่ดีเสมอมาตลอดจนยังได้ชี้แนะแนวทางในการแก้ไขปัญหาต่างๆ ในการทำปริญญานิพนธ์มาโดยตลอด

ขอขอบพระคุณ พี่แปะ น้องกฤษณ์ ที่คอยให้คำปรึกษา และชี้แนวทาง ในการแก้ไขปัญหาต่างๆ ด้วยดีตลอดมา

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใยและให้การสนับสนุน ในการศึกษา รวมทั้งขอขอบคุณญาติสนิทและพี่ๆ ทุกคนที่เป็นกำลังใจ พร้อมทั้งให้คำช่วยเหลือ ในด้านต่างๆ มาโดยตลอด

สุดท้ายขอขอบพระคุณเพื่อนๆ คณะวิศวกรรมศาสตร์สารสนเทศชั้นปีที่ 4 ที่คอยช่วยเหลือมากันโดยตลอด

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ฌ
บทที่ 1 บทนำ	
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงงาน	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินโครงงาน	3
1.6 อุปกรณ์ที่ต้องใช้	4
1.6.1 ฮาร์ดแวร์(Hardware)	4
1.6.2 ซอฟต์แวร์(Software)	4
1.7 เนื้อหาภายในปริญญานิพนธ์	4
บทที่ 2 เทคโนโลยีและหลักการ	
2.1 ทฤษฎีระบบการสื่อสารไร้สาย	5
2.1.1 องค์ประกอบของระบบการสื่อสารไร้สาย	6
2.1.2 สื่อนำข้อมูลแบบไร้สาย	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 รูปแบบการสื่อสารไร้สาย 8
2.1.4 ทิศทางการส่งสัญญาณของระบบสื่อสาร 10
2.1.5 รูปแบบการส่งสัญญาณของระบบการสื่อสาร 11
2.1.6 การมอดดูเลตสัญญาณแบบดิจิตอล 12
2.2 ภาษา C 14
2.2.1 โครงสร้างของภาษา C 16
2.3 ET-BASE AVR ATmega64/128 17
2.3.1 คุณสมบัติของบอร์ด 17
2.3.2 โครงสร้างของบอร์ด 17
2.3.3 ขั้วต่อสัญญาณต่าง ๆ 19
2.3.4 การดาวน์โหลด Hex File ให้กับ MCU 20
2.3.5 การเชื่อมต่ออุปกรณ์สำหรับโปรแกรม Hex File 21
2.4 โมดูลการสื่อสารไร้สาย DCBT-24AX 22
2.4.1 คุณสมบัติทั่วไป 22
2.4.2 โครงสร้างการสื่อสารไร้สาย DCBT-24AX 23
2.4.3 ขั้วต่อสัญญาณต่างๆ 23
2.5 ไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น ATmega168V 24
2.6 ชิพรับส่งสัญญาณไร้สาย nRF24L01 28
2.6.1 คำสั่งการตั้งค่าการใช้งานโมดูล DCBT-24AX 29
2.7 บอร์ดบันทึกเสียง nRF24L01 33
2.7.1 ขั้วต่อสัญญาณต่างๆ 33
2.7.2 คุณสมบัติทั่วไป 34
2.7.3 โครงสร้าง nRF24L01 34
2.7.4 คุณสมบัติทั่วไป 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบและการคำนวณ	
3.1 การออกแบบโครงงาน	38
3.2 โมดูลการสื่อสารไร้สาย	39
3.2.1 ตัวส่งสัญญาณ	38
3.2.2 โมดูลในการส่งสัญญาณไร้สายภาครับ	40
3.3 ISD–4002	41
3.3.1 วิธีการบันทึกเสียงลงไอซี ISD-4002	41
3.3.2 วิธีการอ่านตำแหน่งที่บันทึกเสียงลงไอซี	42
3.4 ไมโครคอนโทลรเลอร์ AVR ATmega128	43
บทที่ 4 ผลการทดลอง	
4.1 ผลการทดลองตามวัตถุประสงค์หลัก	45
บทที่ 5 สรุปผลและแนวทางการพัฒนา	
5.1 สรุปผล	48
5.2 วิจารณ์	49
5.3 แนวทางการพัฒนา	49
บรรณานุกรม	50
ภาคผนวก ก. คู่มือการใช้งาน	51
ภาคผนวก ข. คู่มือการติดตั้ง	52
ภาคผนวก ค. วงจรชิ้นงานและ PCB	60
ภาคผนวก ง. Datasheet	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 1.1	คนพิการทางสายตาใช้อุปกรณ์อุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน	2
รูปที่ 2.1	การสื่อสารไร้สาย	6
รูปที่ 2.2	เปียโนเสมือนแบบพกพา(portable virtual piano)	7
รูปที่ 2.3	การสื่อสารของสัญญาณวิทยุ	7
รูปที่ 2.4	การสื่อสารผ่านดาวเทียม	8
รูปที่ 2.5	การรับส่งข้อมูลสัญญาณอนาลอค(เบสแบนด์)	9
รูปที่ 2.6	การรับส่งข้อมูลสัญญาณอนาลอค(แบบมอดูเลต)	9
รูปที่ 2.7	การสื่อสารทั้งแบบอนาลอกและดิจิตอล	10
รูปที่ 2.8	การส่งสัญญาณแบบ Simplex	11
รูปที่ 2.9	การส่งสัญญาณแบบ Half duplex	11
รูปที่ 2.10	การส่งสัญญาณแบบ Full duplex	11
รูปที่ 2.11	การส่งข้อมูลแบบอนุกรม	11
รูปที่ 2.12	การส่งข้อมูลแบบขนาน	12
รูปที่ 2.13	การมอดดูเลตสัญญาณแบบดิจิตอล	14
รูปที่ 2.14	วงจรบอร์ด ET-BASE AVR ATmega64/128	18
รูปที่ 2.15	โครงสร้างของบอร์ด ET-AVR ISP	20
รูปที่ 2.16	การจัดเรียงขาสัญญาณ	21
รูปที่ 2.17	การเลือก Jumper และการต่อสาย Download ของ ET-CAP10P (ซ้าย) V2.0(ขวา) V1.0 เพื่อใช้กับ AVR	22
รูปที่ 2.18	โครงสร้างชุดโมดูล DCBT-24AX	23
รูปที่ 2.19	การเชื่อมต่อการทำงานโมดูล DCBT 24-AX ระหว่าง ATmega168 กับ nRF24L01	23
รูปที่ 2.20	ขาต่างๆ ของ DCBT-24AX	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.21 โครงสร้างของไมโครคอนโทรลเลอร์ ATmega168V แบบ 32 ขา 26
รูปที่ 2.22 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ ATmega168V 27
รูปที่ 2.23 ลักษณะขาของชิพไอซี nRF24L01 28
รูปที่ 2.24 ขาต่างๆ ภายในไอซี ISD4002-120 33
รูปที่ 2.25 วงจรการทำงานของไอซีบันทึกเสียง ISD4002 34
รูปที่ 2.26 ลักษณะคลื่นสัญญาณของการสื่อสารของ ISD4002 36
รูปที่ 2.27 แสดง timing diagram ของการบันทึก 37
รูปที่ 3.1 บล็อกไดอะแกรมของโครงงานทั้งหมด 38
รูปที่ 3.2 Flowchart การทำงานของโมดูลสื่อสารไร้สายภาคส่ง 39
รูปที่ 3.3 Flow Chart การส่งสัญญาณไร้สายภาครับ 40
รูปที่ 3.4 Flow Chart การบันทึกเสียงลงไอซี ISD-4002 41
รูปที่ 3.5 Flow Chart การอ่านตำแหน่งที่บันทึกเสียง 42
รูปที่ 3.6 Flow Chart การทำงานของ AVR 44
รูปที่ 4.1 ตัวส่งสัญญาณไร้สาย 45
รูปที่ 4.2 ตัวรับสัญญาณ แบบไม่มีสวิตซ์ 46
รูปที่ 4.3 ตัวรับสัญญาณ แบบมีสวิตซ์ 46
รูปที่ 5.1 การเชื่อมต่ออุปกรณ์ทั้งหมดของโครงงาน 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 1.1	ขั้นตอนการดำเนินโครงงาน	5
ตารางที่ 2.1	เปรียบเทียบรายละเอียดภาษาต่างๆ	15
ตารางที่ 2.2	การจัดเรียงขาสัญญาณ	21
ตารางที่ 2.3	การทำงานของขาต่างๆ ระหว่าง AVR ATmega168V กับ nrf24l01	23
ตารางที่ 2.4	การทำงานของขาต่างๆ ของชิพ nRF24L01 ภายในโมดูล DCBT-24AX	29
ตารางที่ 2.5	คำสั่งต่างๆ ของอุปกรณ์รับ-ส่งข้อมูล ชิพ nRF24L01	30
ตารางที่ 2.6	ตำแหน่ง Registers ที่เกี่ยวข้อง	30
ตารางที่ 2.7	การตั้งค่า Configuration(00)	30
ตารางที่ 2.8	การตั้งค่า RF Gain(06)	31
ตารางที่ 2.9	การแสดงสถานะของชิพ nRF24L01(07)	31
ตารางที่ 2.10	การอ่านสถานะของชิพ nRF24L01(FF)	32
ตารางที่ 2.11	รายละเอียดการทำงานของขาต่างๆ ภายในไอซี ISD4002-120	33
ตารางที่ 2.12	รายละเอียดการทำงานของขาต่างๆ ภายในไอซี ISD4002-120	35
ตารางที่ 2.13	รายละเอียดคำสั่งต่างๆ ของ ISD4002	36
ตารางที่ 4.1	การตั้งค่ากำลังการส่งของ RF Module	45
ตารางที่ 4.2	การทดลองค่ากำลังการส่งของ RF Module	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

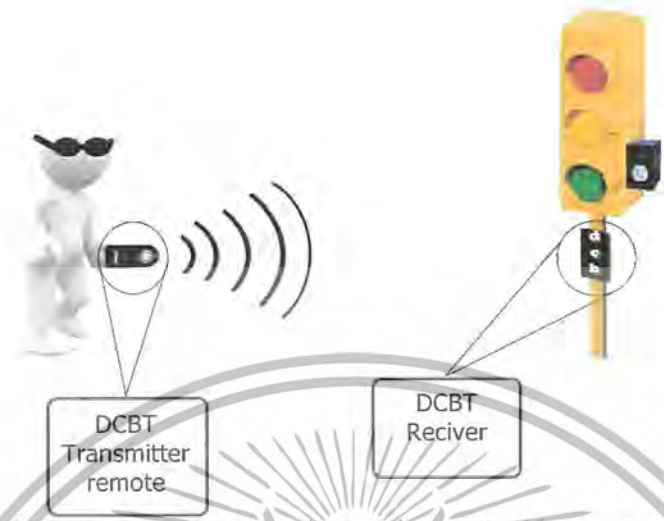
จากรายงานสถิติการเกิดอุบัติเหตุของสำนักงานเครือข่ายลดอุบัติเหตุพบว่า 1 วัน มีอุบัติเหตุเกิดขึ้นบนท้องถนนเฉลี่ย 300 ครั้ง มีผู้บาดเจ็บประมาณ 350 คน ซึ่งผู้พิการทางสายตา และคนชราที่จำเป็นต้องใช้ถนน จัดเป็นกลุ่มที่เสี่ยงต่อการเกิดอุบัติเหตุ เพราะผู้พิการทางสายตาจะขาดความสามารถในการมองเห็น ทำให้ไม่สามารถรับรู้ได้ว่ามียานพาหนะที่มาจากระยะทางไกลๆ ได้ และสำหรับคนชราที่ความสามารถในการรับรู้ และความสามารถในการเคลื่อนไหวได้ไม่ดี จึงเป็นอันตรายต่อการข้ามถนนเป็นอย่างมาก

จึงเป็นสาเหตุที่ทำให้คิดปริญญานิพนธ์นี้ขึ้นมา เพื่อช่วยเหลือผู้พิการทางสายตา และคนชรา ให้ข้ามถนนได้อย่างปลอดภัย โดยอาศัยโมดูลในการส่งแบบไร้สายเพื่ออำนวยความสะดวกในการใช้งาน ซึ่งโมดูลที่ใช้คือ การส่งโดยใช้สัญญาณไวเลส เนื่องจากเป็นโมดูลในการส่งที่มีต้นทุนต่ำ เหมาะแก่การนำไปใช้งานในจำนวนมาก และโดนรบกวนจากคลื่นต่างๆ ได้ยากแล้วยังสามารถทะลุสิ่งกีดขวางได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาการใช้งานโมดูลรับและส่งสัญญาณไร้สาย DCBT-24AX
2. เพื่อศึกษาการควบคุมไมโครคอนโทรลเลอร์ AVR ATmega128 เพื่อจำลองการทำงานของไฟสัญญาณจราจร
3. เพื่อศึกษาการใช้งานไอซีบันทึกเสียง ISD-4002
4. เพื่อแก้ปัญหาและป้องกันอุบัติเหตุให้แก่ผู้พิการทางสายตาและคนชราในการข้ามถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 คนพิการทางสายตาใช้อุปกรณ์อุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

1.3 ขอบเขตของโครงงาน

1. ใช้โมดูลการส่งสัญญาณไร้สายสร้างเป็นรีโมทส่งสัญญาณไร้สาย และสร้างตัวรับสัญญาณไร้สายเพื่อนำไปอินเทอรัพการทำงานของไมโครคอนโทรลเลอร์
2. ใช้ไอซีบันทึกเสียงบันทึกเสียง และอ่านตำแหน่งที่บันทึกเสียงเพื่อเล่นเสียงที่ตรงตามค่าที่ไมโครคอนโทรลเลอร์ส่งมา
3. ใช้ชุดอุปกรณ์ไมโครคอนโทรลเลอร์สร้างแบบจำลองการทำงานของเสาไฟสัญญาณจราจร โดยสามารถรับการอินเทอรัพจากโมดูลตัวรับสัญญาณไร้สายได้ และสามารถส่งค่าไปบังคับไอซีบันทึกเสียงเพื่อเล่นเสียงตามที่กำหนดได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้การทำงานของโมดูลรับ-ส่งสัญญาณไร้สาย DCBT-24AX
2. ได้เรียนรู้การทำงานของไอซีบันทึกเสียง ISD-4002
3. ได้เรียนรู้การใช้งานไมโครคอนโทรลเลอร์(AVR ATmega128) ในการจำลองไฟสัญญาณจราจร รับค่าอินเทอรัพภายนอก และส่งค่าไปควบคุมไอซีบันทึกเสียง
4. สามารถนำอุปกรณ์ที่พัฒนาขึ้น ไปใช้งานได้จริง เพื่อให้เกิดประโยชน์แก่สังคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนการดำเนินโครงงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินโครงงาน

ขั้นตอนที่	วิธีการดำเนินงาน	ระยะเวลาการดำเนินงาน							
		มิถุนายน 51	กรกฎาคม 51	สิงหาคม 51	กันยายน 51	พฤศจิกายน 51	ธันวาคม 51	มกราคม 52	กุมภาพันธ์ 52
1.	ศึกษาการทำงานของไฟสัญญาณจราจร	←	→						
2.	จัดทำแบบจำลองไฟสัญญาณจราจร	←	→	→					
3.	ศึกษาการทำงานของสัญญาณไวเลส และทำการเขียนโปรแกรมติดต่อ ระหว่างรีโมทและโมดูลตัวรับสัญญาณ ไวเลส			←	→				
4.	ทดสอบการใช้งานของอุปกรณ์ ตัวรับ-ส่งสัญญาณไวเลส กับแบบจำลอง ไฟสัญญาณจราจร			←	→				
5.	จัดทำเอกสารการสอบวิชาโครงงาน 1			←	→				
6.	ศึกษาและสรุปงานที่จะต้องทำเพิ่มเติม ในเทอม 2				←	→			
7.	จัดทำอุปกรณ์เพิ่มเสียงพูดให้กับตัว แบบจำลอง และกำหนดรูปแบบ ไฟสัญญาณจราจรในกรณีต่าง ๆ					←	→	→	
8.	ทดสอบการใช้งานจากสถานการณ์ จำลอง						←	→	
9.	จัดทำปริญญานิพนธ์ฉบับสมบูรณ์							←	→

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 อุปกรณ์ที่ต้องใช้

1.6.1 ฮาร์ดแวร์(Hardware)

-ชุดทดลองไมโครคอลโทรลเลอร์ AVR ATmega128 จำนวน 1 ชุด

-ชุดโมดูลสื่อสารไร้สาย DCBT-24AX จำนวน 3 ชุด

-ชุดบันทึกเสียง ISD-400 จำนวน 1 ชุด

1.6.2 ซอฟต์แวร์(Software)

-CodeVisionAVR C Compiler Evaluation

-AVRPROG

-Protel99SE

1.7 เนื้อหาภายในปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ประกอบด้วยเนื้อหาในบทต่างๆ ดังนี้

บทที่ 1 บทนำเป็นบทที่กล่าวถึงแนวคิดและที่มาของปริญญานิพนธ์วัตถุประสงค์ ขอบเขต ประโยชน์ที่ได้รับ ขั้นตอนการดำเนินงาน และอุปกรณ์ที่ใช้ในการทำปริญญานิพนธ์

บทที่ 2 นำทฤษฎีและหลักการที่เกี่ยวข้องกับปริญญานิพนธ์ฉบับนี้มารวบรวมไว้เพื่อศึกษาข้อมูลทฤษฎี และวิธีการใช้งานอุปกรณ์ต่างๆ ในการศึกษาหรือการทดลองตามปริญญานิพนธ์นี้

บทที่ 3 ออกแบบการทดลองในส่วนของทั้งฮาร์ดแวร์ และซอฟท์แวร์ บรรยายภาพรวมการทำงานของปริญญานิพนธ์ในรูปแบบบล็อกไดอะแกรม และวงจรในการเชื่อมต่อ เพื่อเตรียมชุดอุปกรณ์ให้พร้อมใช้งานในการทดลองจริง

บทที่ 4 เป็นการนำเสนอข้อมูลผลการทดลองตามความเป็นจริงที่ได้จากการทดลอง เพื่อเตรียมนำไปสรุปผล

บทที่ 5 บทสุดท้ายที่นำผลการทดลองในบทที่ 4 มาสรุปผล แล้วชี้แจงปัญหาและอุปสรรคที่เกิดขึ้นระหว่างการออกแบบและสร้างชุดอุปกรณ์ หรือระหว่างการทดลอง พร้อมชี้นำแนวทางในการศึกษาและพัฒนาชุดอุปกรณ์ต่อๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
เทคโนโลยีและหลักการ

ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อประยุกต์ใช้โมดูลการสื่อสารแบบไร้สาย และไอซีบันทึกเสียงทำงานร่วมกันโดยผ่านไมโครคอนโทรลเลอร์ตระกูล AVR ซึ่งการทำงานแบ่งได้เป็นสามส่วน คือส่วนแรกเป็นอุปกรณ์รับส่งสัญญาณไร้สาย เพื่อตรวจสอบว่าตัวส่งสัญญาณไร้สายได้ถูกปล่อยสัญญาณมาเมื่อใด ในส่วนนี้จะใช้โมดูล DCBT-24AX ที่ภายในประกอบด้วยไมโครคอนโทรลเลอร์ AVR เบอร์ Mega168V อยู่บนโมเดล เชื่อมต่อกับชิพ nRF24L01 เป็นโมดูลการส่งสัญญาณไร้สาย ในส่วนที่สองเป็นไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega128 ซึ่งใช้เป็นตัวจำลองสัญญาณไฟจราจร โดยเชื่อมต่อกับ DCBT-24AX ที่ขาอินเตอร์รัพ เมื่อได้รับสัญญาณไฟจาก DCBT-24AX จะทำการเรียกใช้งานไอซีบันทึกเสียงซึ่งอยู่ในส่วนที่สาม โดยใช้ไอซีบันทึกเสียงตระกูล ISD4002-120 ทำการบันทึกเสียงสถานะของสัญญาณไฟจราจรไว้ เมื่อได้รับสัญญาณไฟจาก ATmega128 จะเล่นไฟล์เสียงที่ทำการบันทึกไว้ผ่านลำโพง เพื่อให้สามารถใช้งานอุปกรณ์ได้ตามขอบเขตที่กำหนดไว้ควรศึกษาข้อมูลทฤษฎี และวิธีการใช้งานอุปกรณ์ต่างๆ ตามลักษณะการทำงานดังต่อไปนี้

2.1 ทฤษฎีระบบการสื่อสารไร้สาย

การสื่อสาร หมายถึง การนำสื่อหรือข้อความของฝ่ายหนึ่งส่งให้อีกฝ่ายหนึ่ง ประกอบด้วยผู้ส่งข่าวสารหรือแหล่งกำเนิดข่าวสาร ช่องทางการส่งข้อมูล ซึ่งเป็นสื่อกลางหรืออาจเป็นสายสัญญาณ และหน่วยรับข้อมูลหรือผู้รับสาร

ในปัจจุบันเทคโนโลยีได้พัฒนาและก้าวหน้าไปอย่างมาก ทั้งยังมีส่วนในการดำรงชีวิตประจำวันมากขึ้นเรื่อยๆ ในอดีตการสื่อสารและการส่งสัญญาณต่างๆ ต้องใช้สายนำสัญญาณเพื่อนำพาข้อมูลข่าวสารนั้นๆ ไปยังจุดหมายปลายทางที่ต้องการ ซึ่งมีข้อจำกัดมาก เช่น ระยะทางในการสื่อสาร หรือการควบคุมในระยะไกล เป็นจุดกำเนิดในการคิดค้นการสื่อสารไร้สาย(Wireless Communication) ขึ้นเพื่อสนองความต้องการของคนในยุคปัจจุบัน เนื่องจากการสื่อสารแบบไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถส่งข่าวสารข้อมูลผ่านทางอากาศโดยไม่ต้องใช้สายนำสัญญาณ ทำให้การส่งข่าวสารมีความสะดวกสบายขึ้น

2.1.1 องค์ประกอบของระบบการสื่อสารไร้สาย

เครื่องส่ง(Transmitter)

คือ อุปกรณ์ที่ถูกออกแบบสำหรับเปลี่ยนสัญญาณข้อมูล ให้เป็นสัญญาณที่เหมาะสมกับการส่งผ่านช่องสื่อสารหรือตัวกลางของแต่ระบบ

ช่องสัญญาณสื่อสารหรือตัวกลาง(Communication Channel)

คือ ส่วนที่ใช้ในการนำพาสัญญาณระหว่างผู้ส่งและผู้รับ

เครื่องรับ(Receiver)

คือ อุปกรณ์ที่ทำหน้าที่รับสัญญาณที่ผ่านช่องสื่อสารและแปลงเป็นสัญญาณที่เหมาะสมทำให้อุปกรณ์ปลายทางจัดการกับข้อมูลที่รับเข้ามาได้

รูปที่ 2.1 การสื่อสารไร้สาย

2.1.2 สื่อนำข้อมูลแบบไร้สาย

การสื่อสารข้อมูลแบบไร้สาย จะใช้อากาศเป็นตัวกลางของการสื่อสาร ลักษณะของการสื่อสารข้อมูลประเภทนี้ เช่น

แสงอินฟราเรด(Infrared)

อินฟราเรด เป็นการสื่อสารข้อมูลโดยใช้แสงอินฟราเรดเป็นสื่อกลางโดยในการส่งข้อมูลจำเป็นต้องมีอุปกรณ์ที่ทำหน้าที่ส่งข้อมูลและอุปกรณ์ที่รับข้อมูล ซึ่งคอมพิวเตอร์และอุปกรณ์ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบัน เช่น เมาส์ เครื่องพิมพ์ และกล้องดิจิตอล ซึ่งสื่อประเภทนี้นิยมใช้สำหรับการสื่อสารข้อมูลระยะใกล้

รูปที่ 2.2 เปียโนเสมือนแบบพกพา(portable virtual piano)

สัญญาณวิทยุ(Radio Wave)

สัญญาณวิทยุ เป็นสื่อนำข้อมูลแบบไร้สายที่มีการส่งข้อมูลเป็นสัญญาณคลื่นวิทยุไปในอากาศไปยังตัวรับสัญญาณ จึงทำให้ถูกสภาพแวดล้อมรบกวนข้อมูลได้ในช่วงที่สภาพอากาศไม่ดีการส่งสัญญาณวิธีนี้จะช่วยส่งข้อมูลในระยะทางไกล หรือในสภาพภูมิประเทศที่ไม่เอื้ออำนวยในการใช้สายส่งข้อมูล

รูปที่ 2.3 การสื่อสารของสัญญาณวิทยุ

ไมโครเวฟภาคพื้นดิน(Terrestrial Microwave)

ไมโครเวฟภาคพื้นดิน เป็นการสื่อสารโดยใช้สื่อนำข้อมูลแบบไร้สายอีกประเภทหนึ่ง การสื่อสารประเภทนี้จะมีเสาส่งสัญญาณไมโครเวฟที่อยู่ห่างๆ กัน ทำการส่งข้อมูลไปยังอากาศไปยังเสารับข้อมูล ในกรณีที่ระยะทางห่างกันมาก หรือมีสิ่งกีดขวางสัญญาณ จะต้องใช้สถานีทวนสัญญาณ(Repeater Station) เพื่อส่งสัญญาณต่อเป็นช่วงๆ การสื่อสารประเภทนี้สามารถส่งข้อมูลปริมาณมากได้ แต่ในบางครั้งอาจถูกสภาพแวดล้อมรบกวนได้เช่นกัน โดยเฉพาะช่วงฝนตกหรือมีพายุ จะทำให้การส่งข้อมูลทำได้ไม่ดีนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารผ่านดาวเทียม(Satellite Communication)

การสื่อสารผ่านดาวเทียม เป็นการสื่อสารจากพื้นโลกที่มีการส่งข้อมูลไปยังดาวเทียม โดนดาวเทียมจะทำหน้าที่เป็นสถานีทวนสัญญาณ เพื่อจัดส่งสัญญาณต่อไปยังสถานีภาคพื้นดินอื่นๆ ระยะทางจากโลกถึงดาวเทียมประมาณ22,000 ไมล์ ซึ่งเป็นระยะทางที่ไกลมาก ทำให้ข้อมูลที่ส่งไปยังดาวเทียมเกิดความล่าช้าขึ้นได้ โดยเฉลี่ยความล่าช้าที่เกิดขึ้นมีค่าประมาณ 2 วินาที การส่งข้อมูลวิธีนี้จะทำให้ส่งข้อมูลที่มีระยะทางไกลมากๆ ได้ การสื่อสารผ่านดาวเทียมนิยมใช้สำหรับการสื่อสารระหว่างประเทศรูป การสื่อสารผ่านดาวเทียม

รูปที่ 2.4 การสื่อสารผ่านดาวเทียม

2.1.3 รูปแบบการสื่อสารไร้สาย

การสื่อสารข้อมูลสามารถแบ่งออกได้เป็น 2 ประเภท คือ

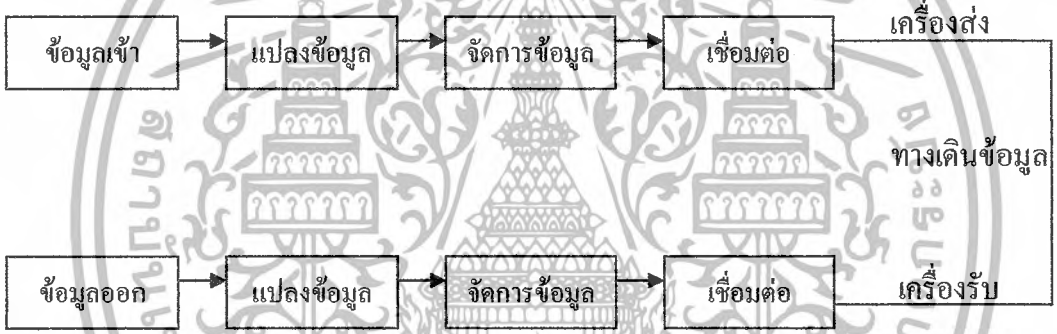
1. การสื่อสารแบบสัญญาณอนาลอก(analog)

สัญญาณอนาลอก เป็นสัญญาณแบบต่อเนื่อง มีลักษณะเป็นคลื่นไซน์(sine wave) โดยที่แต่ละคลื่นจะมีความถี่และความเข้มของสัญญาณที่ต่างกัน เมื่อนำสัญญาณข้อมูลเหล่านี้มาผ่านอุปกรณ์รับสัญญาณและแปลงสัญญาณ ก็จะได้ข้อมูลที่ต้องการได้ ตัวอย่างการส่งข้อมูลที่มีสัญญาณแบบอนาลอก คือการส่งข้อมูลผ่านระบบโทรศัพท์ เฮิรตซ์(hertz) คือ หน่วยวัดความถี่ของสัญญาณข้อมูลแบบอนาลอก วิธีวัดความถี่จะนับจำนวนรอบของสัญญาณที่เกิดขึ้นภายใน 1 วินาที เช่น สัญญาณ

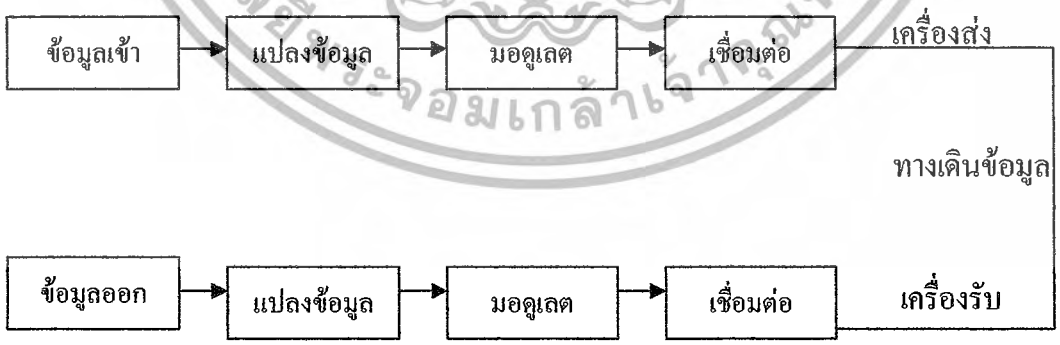
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่มีความถี่ 60Hz หมายถึงใน 1 วินาที สัญญาณมีการเปลี่ยนแปลงระดับสัญญาณ 60 รอบ(ขึ้นและลงนับเป็น 1 รอบ)

สิ่งที่จะใช้ในการพิจารณาระบบนี้ คือ อัตราส่วนของกำลังสัญญาณหลักต่อกำลังสัญญาณรบกวน(Signal to Noise; S/N) โดยถ้า S/N สูงแสดงว่าระบบนั้นมีประสิทธิภาพดีมีประสิทธิภาพดี แต่ถ้ามี S/N ต่ำแสดงว่าระบบมีประสิทธภาพไม่ดี และค่าแบนวิธ ที่หมายถึงช่วงความถี่ที่ครอบคลุมกำลังส่วนมาก หรือช่วงความถี่ที่อัตราขยาย หรือค่าลดทองเล็กน้อยในช่วงกลางๆ ของแบนวิธ โดยทั่วไปจะกำหนดเขตความกว้างที่ 3 dB หรือครึ่งหนึ่งของกำลังสูงสุด ซึ่งมีค่าแบนวิธเท่ากับ 3,000 เฮิรตซ์ ที่ 3 dB



รูปที่ 2.5 การรับส่งข้อมูลสัญญาณอนาลอค(เบสแบนด์)



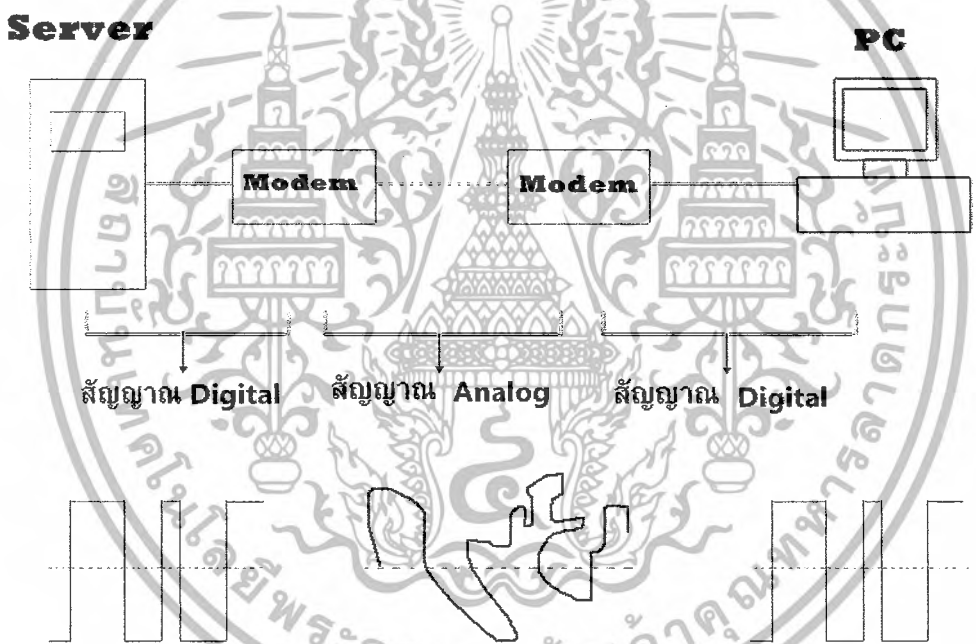
รูปที่ 2.6 การรับส่งข้อมูลสัญญาณอนาลอค(แบบมอดูเลต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ระบบการสื่อสารแบบดิจิตอล(Digital)

สัญญาณดิจิตอลเป็นสัญญาณแบบไม่ต่อเนื่อง รูปแบบของสัญญาณมีความเปลี่ยนแปลงที่ไม่ปะติดปะต่ออย่างสัญญาณอนาลอก โดยการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล จะใช้วิธีการสุ่มตัวอย่าง(Sampling) จากสัญญาณอนาลอกในการสื่อสารด้วยสัญญาณดิจิตอล ข้อมูลในคอมพิวเตอร์ซึ่งเป็นเลขฐานสอง(0 และ 1) จะถูกแทนด้วยสัญญาณดิจิตอล

Bit rate เป็นอัตราความเร็วในการส่งข้อมูลแบบดิจิตอล วิธีวัดความเร็วจะนับจำนวนBit ข้อมูลที่ส่งได้ในช่วงระยะเวลา 1 วินาที เช่น 14.4 Kbps หมายถึง มีความเร็วในการส่งข้อมูลจำนวน 14.4 KBit ในระยะเวลา 1 วินาที



รูปที่ 2.7 การสื่อสารทั้งแบบอนาลอกและดิจิตอล

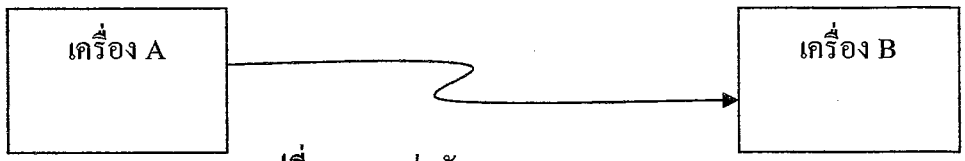
2.1.4 ทิศทางการส่งสัญญาณของระบบสื่อสาร

การส่งข้อมูลสามารถจำแนกทิศทางการส่งข้อมูลได้เป็น 3 รูปแบบดังนี้

1 การส่งแบบทิศทางเดียว(Simplex)

หมายถึง การส่งสัญญาณจากด้านส่ง ไปยังด้านรับเพียงด้านเดียว โดยที่ไม่สามารถโต้ตอบระหว่างกันได้ เช่น การกระจายเสียงของสัญญาณวิทยุ หรือโทรทัศน์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การส่งสัญญาณแบบ Simplex

2.การส่งแบบสองทิศทางแต่ต่างเวลากัน(Half duplex)

หมายถึงทั้งทางด้านส่ง ด้านรับสามารถโต้ตอบระหว่างกันได้ โดยมีข้อกำหนดว่าต้องมีด้านหนึ่งเป็นด้านรับเสมอ(ผลัดกันส่ง) เช่นวิทยุสมัครเล่น

รูปที่ 2.9 การส่งสัญญาณแบบ Half duplex

3. การส่งแบบสองทิศทางที่เวลาเดียวกัน(Full duplex)

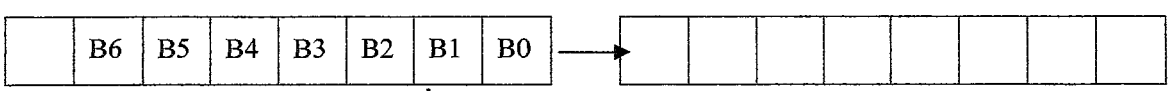
หมายถึง การที่ด้านรับและด้านส่งสามารถส่งสัญญาณได้พร้อมกันในเวลาเดียวกัน โดยที่ไม่ต้องผลัดกันส่งเหมือนแบบ Half duplex

รูปที่ 2.10 การส่งสัญญาณแบบ Full duplex

2.1.5 รูปแบบการส่งสัญญาณของระบบการสื่อสาร

แบ่งเป็น 2 ประเภท ดังนี้

1. การส่งข้อมูลแบบอนุกรม จะเป็นการส่งข้อมูลที่บิทเรียงกันไป



รูปที่ 2.11 การส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

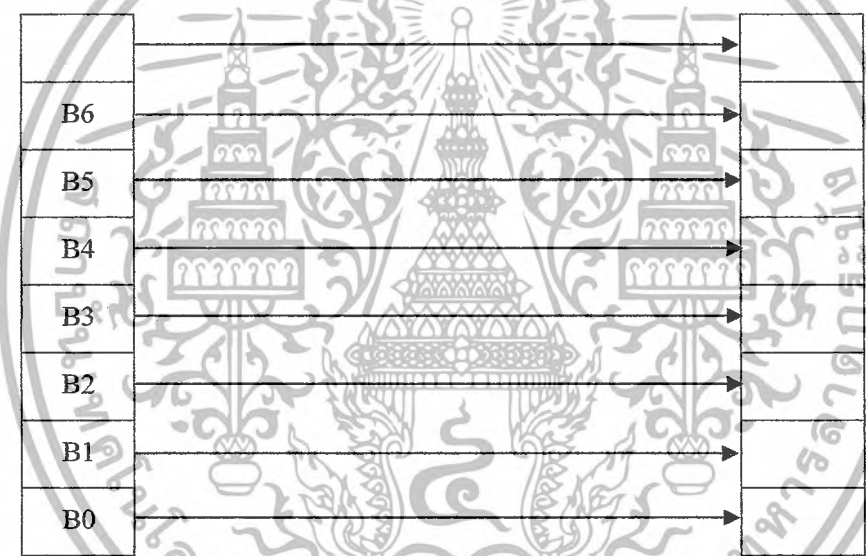
ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรมมีหน่วยเป็น บิตต่อวินาที(bps) ส่วนการเปลี่ยนแปลงของสัญญาณในหนึ่งวินาทีเรียกว่า บอดเรต(baud rate) การเปลี่ยนแปลงของสัญญาณหนึ่งครั้งอาจจะแสดงถึงการส่งข้อมูลแบบอนุกรมมากกว่า 1 บิตก็ได้ ถ้าเขียนในรูปของสมการคณิตศาสตร์จะได้

อัตราบิต(bit rate) = อัตราบอด(baud rate)*(บิตใน 1 บอด)

2. การส่งข้อมูลแบบขนาน

ทุกบิตของข้อมูลจะถูกแยกส่งไปตามทางติดต่อในเวลาเดียวกัน



รูปที่ 2.12 การส่งข้อมูลแบบขนาน

เปรียบเทียบทั้ง 2 วิธีแบบขนานจะใช้ค่าใช้จ่ายสูงกว่าแบบอนุกรม เนื่องจากต้องใช้สายนำข้อมูลมากแต่ก็มีข้อดี คือสามารถส่งข้อมูลได้รวดเร็วกว่าแบบอนุกรมเพราะส่งได้หลาย บิตในเวลาเดียวกัน เหมาะสำหรับการรับ–ส่งข้อมูลในระยะทางใกล้ๆ ส่วนแบบอนุกรมจะเหมาะกับการรับ–ส่งข้อมูลที่ไกลๆ

2.1.6 การมอดดูเลตสัญญาณแบบดิจิตอล

จากสมการทางคณิตศาสตร์ของคลื่นรูปซายน์ ที่เราใช้เป็นพาหะ

$$e = A\sin(\omega t+\Phi) \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะพบว่าเราสามารถทำการเปลี่ยนแปลงหรือมอดดูเลตได้ด้วยแอมพลิจูด, ความถี่เชิงมุม, และเฟส สัญญาณดิจิตอลเบสแบนด์เป็นรูปสี่เหลี่ยมมีขนนาดแปรตามค่า logic high-low ซึ่งสัญญาณที่เรามอดดูเลตจะเปลี่ยนค่าแอมพลิจูด, ความถี่เชิงมุม หรือเฟสไปตาม logic เช่นกัน

การมอดดูเลตข้อมูลที่เป็นดิจิตอลมีหลักการพื้นฐานอยู่ 3 แบบด้วยกันคือ

1. การมอดดูเลตทางขนาด(Amplitude Shift Keying; ASK)

เป็นวิธีมอดดูเลต วิธีแรกๆ ที่ทำให้สามารถส่งข่าวสารในรูปของ electric pulse ไปในระยะไกลๆ ได้ ASK ใช้เทคนิคของการปิด-เปิดสวิทช์ รูปแบบที่ง่ายที่สุดของ carrier คือ สัญญาณ DC ที่มีระดับแรงดัน 0 โวลต์ หรือระดับที่สูงสุดที่จะเป็นไปได้ ในปัจจุบันไม่ว่าจะเป็นการส่งโดยใช้สายส่งหรือไม่ใช้สายส่ง เราก็จะใช้สัญญาณ sinusoidal ที่มีความถี่ในย่าน AF ของ HF เป็น carrior การเปิดปิดสวิทช์ของสัญญาณ carrior อาจจะใช้เครื่องกำเนิดโดยตรง หรืออาจจะใช้สัญญาณ clock ควบคุมเครื่องกำเนิดสัญญาณ carrier อีกชั้นหนึ่ง

ในระบบ ASK สัญญาณไบนารีที่เป็น 1 แทนด้วยการส่งสัญญาณ carrier ที่เป็น sinusoid ที่มีขนาดของ Amplitude คงที่ A_c และความถี่คงที่ f_c สำหรับที่ลำดับของบิตจำนวน T_b seconds, เราสามารถเขียนสมการคณิตศาสตร์แทนสัญญาณ binary ASK ได้เป็น

$$s(t) = \begin{cases} A_c \cos(2\pi f_c t) & symbol\ 1 \\ 0 & symbol\ 0 \end{cases} \qquad (2.2)$$

2. การมอดดูเลตทางความถี่(Frequency Shift Keying; FSK)

ใน FSK จะใช้สัญญาณ carrier ที่เป็น sinusoid 2 ชุด ที่มีขนาด Amplitude A_c เท่ากัน แต่มีความถี่ f_1 และ f_2 ที่แตกต่างกันใช้แทนไบนารี 1 และ 0 ความถี่ทั้งสองจะถูก demodulated โดยใช้ frequency sensitive demodulator ซึ่งแต่ละตัวจะกำหนดให้กับระดับลอจิกหนึ่ง ความแตกต่างระหว่างความถี่ทั้งสองเรียกว่า shift ถ้ายิ่งค่าของ shift มีมากก็ยิ่งทำให้การ demodulation ง่ายยิ่งขึ้น แต่การส่งสัญญาณจะต้องใช้ bandwidth มากขึ้นด้วย เราสามารถแทน FSK ด้วยสมการข้างล่างนี้

$$s(t) = \begin{cases} A_c \cos(2\pi f_{c1} t) & symbol\ 1 \\ A_c \cos(2\pi f_{c1} t) & symbol\ 0 \end{cases} \qquad (2.3)$$

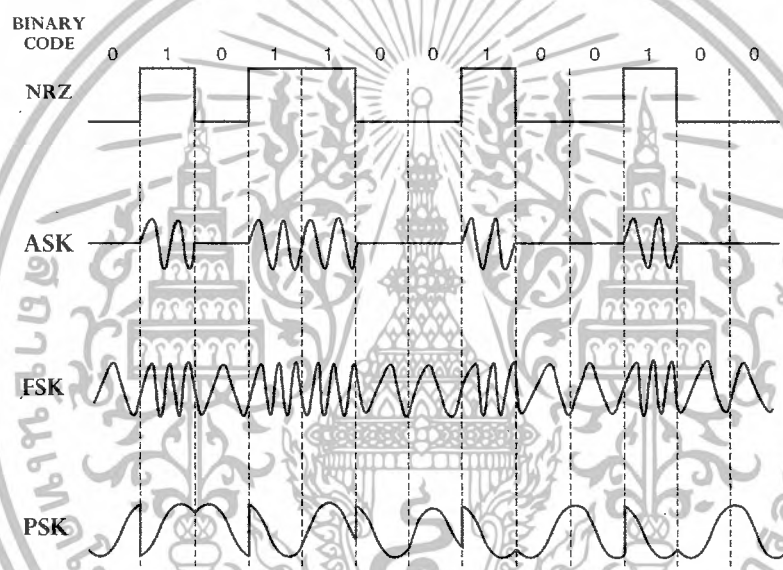
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การมอดดูเลตทางเฟส(Phase Shift Keying; PSK)

ใน PSK จะใช้สัญญาณ carrier ที่เป็น sinusoid ที่มีขนาด Amplitude คงที่ A_c และความถี่คงที่ f_c แทนทั้งไบนารี 1 และ 0 เพียงแต่ phase ของ carrier จะแตกต่างกัน 180 องศา สำหรับไบนารี 1 และ 0

ซึ่งสามารถแทนด้วยสมการข้างล่างนี้

$$s(t) = \begin{cases} A_c \cos(2\pi f_c t) & symbol\ 1 \\ A_c \cos(2\pi f_c t + \pi) & symbol\ 0 \end{cases} \qquad (2.4)$$



รูปที่ 2.13 การมอดดูเลตสัญญาณแบบดิจิตอล

2.2 ภาษา C

การเขียนโปรแกรม(Programming) คือ การเขียนคำสั่งเพื่อสั่งคอมพิวเตอร์ให้ทำตาม คำสั่งที่เขียนเสร็จแล้วเรียกว่าโปรแกรมหรือซอฟแวร์(Software)

ภาษาคอมพิวเตอร์แบ่งเป็น 3 ชนิด

1. Machine Language(ภาษาเครื่อง) ได้แก่ตัวเลขสองค่าเท่านั้น คือ 0 กับ 1 เป็นกลุ่มตัวเลขสองค่าเรียงกันแทนความหมายต่างๆได้แก่ ตัวอักษร ตัวเลข เครื่องหมาย และสัญลักษณ์ เรียกกลุ่มตัวเลข 0 กับ 1ว่าเลขฐานสอง(Binary) เพราะมีตัวเลขสองค่าคือ 0 กับ 1 ซึ่งต่างกับเลขฐานสิบ (Decimal) ที่มนุษย์ใช้ในชีวิตประจำวันซึ่งมีสิบค่าคือ 0 ถึง 9 ตัวอย่างเช่น เลขฐานสอง 00000001 มีค่าเท่ากับ 1 ของเลขฐานสิบ เลขฐานสอง 00000010 มีค่าเท่ากับ 2 ของเลขฐานสิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Low Level Language(ภาษาระดับต่ำ) ได้แก่ ภาษาแอสแซมบลี(Assembly) ใช้คำและอักษรย่อภาษาอังกฤษในการเขียนคำสั่ง เป็นภาษาที่เข้าใจยากกว่าภาษาระดับสูง เพราะต้องเข้าใจการทำงานของเครื่องคอมพิวเตอร์อย่างลึกซึ้ง แต่ข้อดีคือเร็วกว่าภาษาระดับสูง

3. High Level Language(ภาษาระดับสูง) ได้แก่ ภาษา C++ ภาษา Java และอีกหลายภาษา ใช้คำภาษาอังกฤษที่ใช้ในชีวิตประจำวันในการเขียนคำสั่ง เป็นภาษาที่นิยมใช้กันมากเพราะเข้าใจง่าย

ตารางที่ 2.1 เปรียบเทียบรายละเอียดภาษาต่างๆ

Human Language:Decimal (ภาษามนุษย:เลขฐานสิบ)	1	+	3	1 + 3
MachineLanguage:Binary (ภาษาเครื่อง:เลขฐานสอง)	00000001	10101010	00000011	00000001 10101010 00000011
Low Level Language:Assembly (ภาษาระดับต่ำ:ภาษาแอสแซมบลี)	MOV AX,1	ADD	MOV BX,3	ADD AX,BX
High Level Language:C++ (ภาษาระดับสูง:C++,Java)	x = 1	+	y = 3	x + y

ภาษาซี(C Programming Language) คือภาษาคอมพิวเตอร์ระดับสูง(High-Level Language) ที่นิยมใช้ในการเขียนคำสั่งคอมพิวเตอร์ มนุษย์ใช้ภาษาของมนุษย์ติดต่อสื่อสารกัน เช่น ภาษาอังกฤษ แต่ถ้าจะสื่อสารกับคอมพิวเตอร์จำเป็นต้องใช้ภาษาคอมพิวเตอร์ เช่น ภาษา C++

ภาษาซี สามารถนำไปใช้ได้บนเครื่องทุก Platform ไม่ว่าจะเป็น Intel PC ที่วิ่ง Windows 95 หรือ Windows NT หรือ แม้แต่ Linux ทั้งเครื่อง Macintosh และเครื่องเวอร์คสเตชัน ตลอดจนเมนเฟรม เนื่องจากมี compiler ของภาษาซี อยู่ทั่วไป

ภาษาซี เป็นภาษาที่ง่ายๆ คือมีแต่ข้อกำหนดในการใช้งาน หรือ syntax แต่ไม่มีฟังก์ชันสำเร็จรูป(Built-in Function) ใดๆ ดังนั้นหากผู้ใช้ต้องการทำอะไรก็ตาม ต้องเขียนทุกอย่างขึ้นเอง หรือ อาจเรียก Library Functions มาใช้งาน โดย ฟังก์ชันที่เป็นงานที่ใช้บ่อยๆ จะถูกรวบรวมไว้ใน Library Functions เช่น การจัดการข้อความการดำเนินการเกี่ยวกับ Input/Output(I/O) การจองหน่วยความจำ(Memory Allocation) แต่ฟังก์ชันที่สมบูรณ์จะไม่มีใน Standard Library เช่น ฟังก์ชันที่จัดการ Graphics ทั้งนี้จะขึ้นกับระบบที่ใช้(เช่น เป็นระบบ UNIX หรือ Windows 95) และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งแวดล้อมในการทำงาน(เช่น GUI เป็น X-Windows หรือ Direct X) การทำเช่นนี้จะทำให้ภาษาซีเป็นภาษาที่เคลื่อนย้ายได้ง่าย(portable)

เมื่อภาษาซี ได้รับความนิยมมากขึ้น จึงมีผู้ผลิต compiler ภาษาซีออกมาแข่งขันกันมากมาย ทำให้เริ่มมีการใส่ลูกเล่นต่างๆ เพื่อดึงดูดใจผู้ซื้อ ทาง American National Standard Institute(ANSI) จึงตั้งข้อกำหนดมาตรฐานของภาษาซีขึ้น เรียกว่า ANSI C เพื่อคงมาตรฐานของภาษาไว้ไม่ให้เปลี่ยนแปลงไป

2.2.1 โครงสร้างของภาษา C

เฮดเดอร์ไฟล์(Header Files)

เป็นส่วนที่เก็บไลบรารี่มาตรฐานของภาษา C ซึ่งจะถูกดึงเข้ามารวมกับโปรแกรมในขณะที่กำลังทำการคอมไพล์ โดยใช้คำสั่ง

#include<ชื่อเฮดเดอร์ไฟล์> หรือ

#include"ชื่อเฮดเดอร์ไฟล์"

เฮดเดอร์ไฟล์นี้จะมีส่วนขยายเป็น .h เสมอ และเฮดเดอร์ไฟล์เป็นส่วนที่จำเป็นต้องมีอย่างน้อย 1 เฮดเดอร์ไฟล์ ก็คือ เฮดเดอร์ไฟล์ stdio.h ซึ่งจะเป็นที่เก็บไลบรารี่มาตรฐานที่จัดการเกี่ยวกับอินพุตและเอาท์พุต

ส่วนตัวแปรแบบ Global(Global Variables)

เป็นส่วนที่ใช้ประกาศตัวแปรหรือค่าต่าง ๆ ที่ให้ใช้ได้ทั้งโปรแกรม ซึ่งใช้ได้ทั้งโปรแกรม ซึ่งในส่วนไม่จำเป็นต้องมีก็ได้

ฟังก์ชัน(Functions)

เป็นส่วนที่เก็บคำสั่งต่าง ๆ ไว้ ซึ่งในภาษา C จะบังคับให้มีฟังก์ชันอย่างน้อย 1 ฟังก์ชันนั่นคือ ฟังก์ชัน Main() และในโปรแกรม 1 โปรแกรมสามารถมีฟังก์ชันได้มากกว่า 1 ฟังก์ชัน

ส่วนตัวแปรแบบ Local(Local Variables)

เป็นส่วนที่ใช้สำหรับประกาศตัวแปรที่จะใช้ในเฉพาะฟังก์ชันของตนเอง ฟังก์ชันอื่นไม่สามารถเข้าถึงหรือใช้ได้ ซึ่งจะต้องทำการประกาศตัวแปรก่อนการใช้งานเสมอ และจะต้องประกาศไว้ในส่วนนี้เท่านั้น

ตัวแปรโปรแกรม(Statements)

เป็นส่วนที่อยู่ถัดลงมาจากส่วนตัวแปรภายใน ซึ่งประกอบไปด้วยคำสั่งต่างๆ ของภาษา C และคำสั่งต่างๆ จะใช้เครื่องหมาย; เพื่อเป็นการบอกให้รู้ว่าจบคำสั่งหนึ่งๆ แล้ว ส่วนใหญ่ คำสั่งต่างๆ ของภาษา C เขียนด้วยตัวพิมพ์เล็ก เนื่องจากภาษา C จะแยกความแตกต่างของตัวพิมพ์เล็กและพิมพ์ใหญ่หรือ Case Sensitive นั่นเองยกตัวอย่างใช้ Test, test หรือจะถือว่าเป็นตัวแปรคนละตัวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

นอกจากนี้ภาษา C ยังไม่สนใจกับการขึ้นบรรทัดใหม่ เพราะฉะนั้นผู้ใช้สามารถพิมพ์คำสั่งหลายคำสั่งในบรรทัดเดียวกันได้ โดยไม่ต้องใช้เครื่องหมาย; เป็นตัวจบคำสั่ง

ค่าส่งกลับ(Return Value)

คือ ค่าที่ส่งกลับเมื่อฟังก์ชันนั้นๆทำงานเสร็จ ซึ่งเรื่องนี้ผู้เขียนจะยกไปกล่าวในเรื่องฟังก์ชั่นอย่างละเอียดอีกทีหนึ่ง

หมายเหตุ(Comment)/Remark

1. /*และ*/ ใช้สำหรับข้อความที่ยาวกว่า 1 บรรทัด โดยโปรแกรมจะถือว่าข้อความที่ตามหลัง/* จะเป็นหมายเหตุจนกว่าจะพบเครื่องหมาย*/ จึงจะแสดงว่าจบหมายเหตุแล้ว

2. // เหมาะสำหรับข้อความสั้นๆ 1 บรรทัด โดยถ้าบรรทัดใดขึ้นต้นด้วย// บรรทัดนั้นจะถือว่าเป็นหมายเหตุ

2.3 ET-BASE AVR ATmega128

ET-BASE AVR ATmega128 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR ของบริษัท Atmel โดยบอร์ดนี้ใช้ MCU เบอร์ ATmega64 และ ATmega128 ขนาด 64 Pin และเน้นการใช้งานทรัพยากรของตัว MCU เองเป็นหลัก จะมีการต่อขาสัญญาณ I/O ออกมาจัดเรียงให้เป็นพอร์ต PA PB PC PD PE PF และพอร์ต ET-CLCD เพื่อสะดวกต่อการใช้งาน พร้อมทั้งพอร์ตสำหรับการดาวน์โหลดโปรแกรม นอกจากนี้ยังได้เพิ่มวงจรไดร์ฟเวอร์ RS-232 เข้าไปด้วยเพื่อให้สามารถใช้งานทางด้านพอร์ตอนุกรม RS-232 ได้ง่ายและสะดวกขึ้น

2.3.1 คุณสมบัติของบอร์ด

เลือกใช้ MCU ตระกูล AVR เบอร์ ATmega64, ATmega128 ของ Atmel ที่เป็น MCU ขนาด 8–Bit โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ XTAL ค่า 16 MHz และคุณสมบัติเด่นๆ ของ MCU ได้แก่

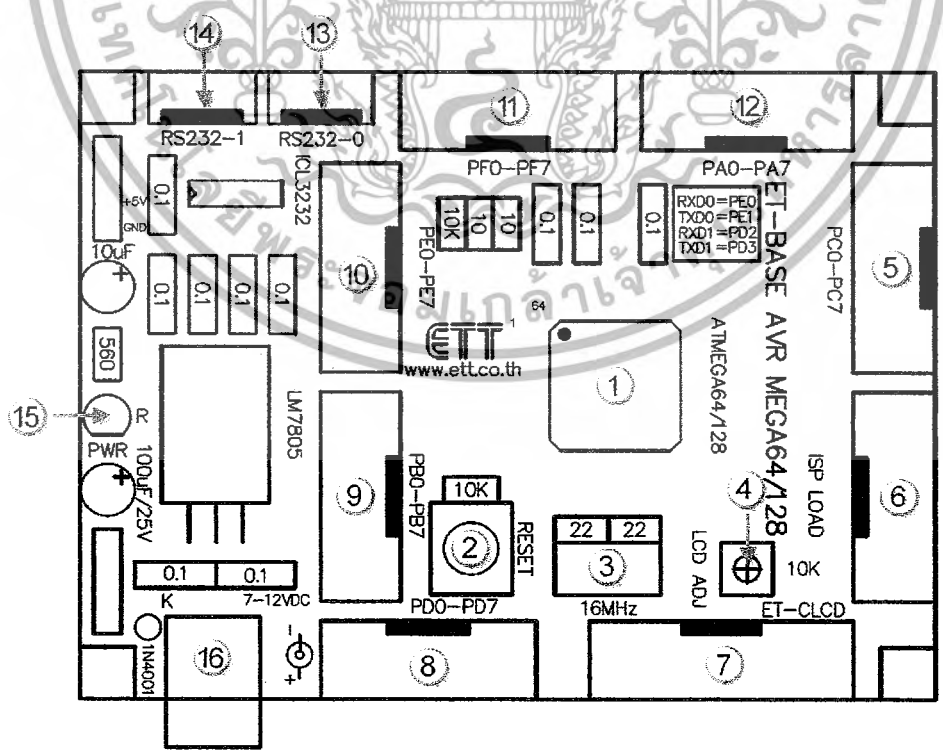
- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 64 Kbytes สำหรับ ATmega64 และ 128 Kbytes สำหรับ ATmega128 และมี RAM 4 Kbytes
- มีหน่วยความจำข้อมูลถาวรแบบ EEPROM ขนาด 2 Kbytes สำหรับATmega64 และ 4 Kbyte สำหรับ ATmega128 ซึ่งสามารถลบและเขียนซ้ำได้กว่า 100,000 ครั้ง
- จำนวน I/O สูงสุดถึง 53 I/O Pins
- มีวงจรสื่อสาร SPI จำนวน 1 ช่อง, I2C จำนวน 1 ช่อง, Programmable Serial USARTs จำนวน 2 ช่อง
- มี ADC ขนาด 10 Bit จำนวน 8 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

104282

- มี Timers/Counters 8-Bit จำนวน 2 ช่อง, Timers/Counters 16-Bit จำนวน 2 ช่อง, 8-Bit PWM 2 ช่อง, Watchdog Timer, Real Time Counter
- I/O PORT 10 PIN จำนวน 6 PORT ดังนี้ PA PB PC PD PE PF
- พอร์ต ISP LOAD สำหรับโปรแกรม MCU(ต้องใช้ร่วมกับET-AVR ISP หรือเครื่องโปรแกรม ISP อื่นที่มีการจัดเรียงขาสัญญาณเหมือนกัน)
- วงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม RS-232 จำนวน 2 ช่อง โดยเชื่อมต่อกับสัญญาณ PE0(RXD0) และ PE1(TXD0) จำนวน 1 ช่อง ส่วนที่เหลืออีก 1 ช่อง จะต่อกับสัญญาณ PD2(RXD1) และ PD3(TXD1) เพื่อให้ผู้ใช้สามารถต่อทดลองการติดต่อสื่อสาร RS-232
- วงจรเชื่อมต่อจอแสดงผล LCD แบบ Character(ET-CLCD) พร้อม VR ปรับความสว่างของ LCD ซึ่งใช้การเชื่อมต่อวงจรกับ LCD แบบ 4 Bit Interface
- วงจร Regulate ขนาด +5V / 1A สำหรับใช้งานเป็นแหล่งจ่ายไฟเลี้ยงวงจรให้กับจอแสดงผล LCD และอุปกรณ์ I/O ต่างๆที่ใช้กับแหล่งจ่ายขนาดขนาด +5V พร้อม LED แสดงสถานะสีแดง
- ขนาด PCB Size เล็กเพียง 8 × 6 cm

2.3.2 โครงสร้างของบอร์ด



รูปที่ 2.14 วงจรบอร์ด ET-BASE AVR ATmega128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-หมายเลข 1 คือ MCU เบอร์ ATmega64 หรือ ATmega128 ซึ่งเป็น MCU ตระกูล AVR จาก ATMEL

-หมายเลข 2 คือ Switch RESET ใช้สำหรับ Reset การทำงานของ MCU

-หมายเลข 3 คือ Crystal ค่า 16 MHz

-หมายเลข 4 คือ ตัวต้านทานสำหรับปรับค่าความสว่างให้ LCD

-หมายเลข 5 คือ PORT C มีขนาด 8 Bit คือ PC0-PC7

-หมายเลข 6 คือ พอร์ต ISP LOAD ใช้สำหรับดาวน์โหลด Hex File ให้กับ MCU

-หมายเลข 7 คือ พอร์ต ET-CLCD สำหรับเชื่อมต่อกับ LCD ชนิด Character Type ซึ่งใช้การเชื่อมต่อแบบ 4 Bit

-หมายเลข 8 คือ PORT D มีขนาด 8 Bit คือ PD0-PD7

-หมายเลข 9 คือ PORT B มีขนาด 8 Bit คือ PB0-PB7

-หมายเลข 10 คือ PORT E มีขนาด 8 Bit คือ PE0-PE7

-หมายเลข 11 คือ PORT F มีขนาด 8 Bit คือ PF0-PF7

-หมายเลข 12 คือ PORT A มีขนาด 8 Bit คือ PA0-PA7

-หมายเลข 13, 14 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป

-หมายเลข 15 คือ LED Power ใช้สำหรับแสดงสถานะของแหล่งจ่ายไฟ +5VDC

-หมายเลข 16 คือ ขั้วต่อแหล่งจ่ายไฟสำหรับเลี้ยงวงจรของบอร์ด

2.3.3 ขั้วต่อสัญญาณต่าง ๆ

สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้นจะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDC-Header ขนาด 10 Pin(2X5) จำนวน 6 ชุด คือ PA PB PC PD PE PF ตามลำดับ โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยจุดเชื่อมต่อกับสัญญาณภายนอกบอร์ดมีดังนี้

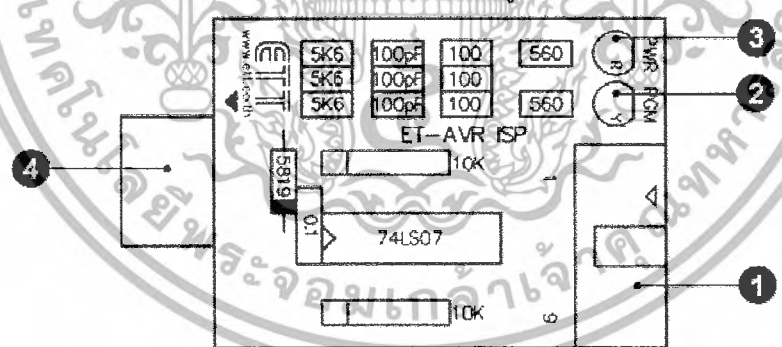
- ขั้วต่อแหล่งจ่ายไฟสำหรับเลี้ยงวงจรของบอร์ด
- ขั้วต่อ PORT A มีขนาด 8 Bit คือ PA0-PA7
- ขั้วต่อ PORT B มีขนาด 8 Bit คือ PB0-PB7
- ขั้วต่อ PORT C มีขนาด 8 Bit คือ PC0-PC7
- ขั้วต่อ PORT D มีขนาด 8 Bit คือ PD0-PD7
- ขั้วต่อ PORT E มีขนาด 8 Bit คือ PE0-PE7
- ขั้วต่อ PORT F มีขนาด 8 Bit คือ PF0-PF7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขั้วต่อ ET-CLCD สำหรับเชื่อมต่อกับ LCD ชนิด Character Type
- ขั้วต่อ RS 232 จำนวน 2 ช่อง โดยเชื่อมต่อกับสัญญาณ PE0(RXD0) และPE1(TXD0) จำนวน 1 ช่อง ส่วนที่เหลืออีก 1 ช่อง จะต่อกับสัญญาณ PD2(RXD1) และ PD3(TXD1) เพื่อให้ผู้ใช้สามารถต่อทดลองการติดต่อสื่อสาร RS232
- ขั้วต่อ ISP LOAD ใช้สำหรับดาวน์โหลด Hex File ให้กับ MCU

2.3.4 การดาวน์โหลด Hex File ให้กับ MCU

การดาวน์โหลด Hex File ให้กับ MCU นั้นจำเป็นจะต้องใช้ **ET-AVR ISP** หรือเครื่องโปรแกรมแบบ ISP อื่นๆ เช่น AVRISP ของ ATMEL เพื่อใช้ในการดาวน์โหลด Hex File ให้กับ MCU ตระกูล AVR ของ Atmel โดยใช้วิธีการแบบ Serial Programming ซึ่งการดาวน์โหลด Hex File ในกรณีที่ใช้ **ET-AVR ISP** จะกระทำผ่านทางพอร์ตขนานของคอมพิวเตอร์ โดยที่จะต้องใช้งานร่วมกับ **ET-CAP10P** ของอีทีที และ Software ที่ใช้ร่วมกับ ET-AVR ISP ก็คือ PonyProg2000 ซึ่ง PonyProg2000 เป็นโปรแกรม Download ข้อมูลแบบ HEX File ให้กับ CPU ตระกูล AVR โดยใช้วิธีการแบบ Serial Programming ซึ่งสามารถใช้งานกับบอร์ดตระกูล AVR ของ อีทีที ได้เป็นอย่างดี ซึ่งวิธีการใช้งานโปรแกรมโดยทั่วไปนั้น สามารถศึกษาได้จาก Help ของโปรแกรมได้เอง โดยในที่นี้จะขอแนะนำให้ทราบถึงวิธีการ Setup โปรแกรม PonyProg2000 เพื่อใช้งานกับบอร์ดตระกูล AVR ของ อีทีที ซึ่งสามารถใช้งานได้กับบอร์ดตระกูล AVR



รูปที่ 2.15 โครงสร้างของบอร์ด ET-AVR ISP

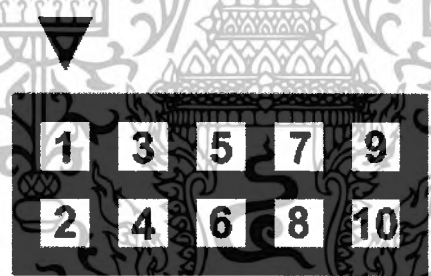
- **หมายเลข 1** คือ พอร์ตสำหรับเชื่อมต่อกับ **ET-CAP10P** ของอีทีที เพื่อโปรแกรม Hex File ให้กับ MCU
- **หมายเลข 2** คือ LED PGM(สีเขียว) แสดงสถานะของการโปรแกรมหรือดาวน์โหลด Hex File ลง MCU
- **หมายเลข 3** คือ LED PWR(สีแดง) แสดงสถานะของไฟเลี้ยงบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **หมายเลข 4** คือ พอร์ตสำหรับเชื่อมต่อกับบอร์ด Target ซึ่งสามารถใช้โปรแกรม Hex File ให้กับบอร์ด ET-AVR STAMP ATmega64 โดยเสียบบอร์ด ET-VR ISP เข้าที่พอร์ต ET-PSPI ซึ่งมีการจัดเรียงขาสัญญาณดังรูป 2.3

ตารางที่2.2 การจัดเรียงขาสัญญาณ

ตำแหน่งขา	ชื่อสัญญาณ
1	MOSI
2	VCC
3	ไม่ได้ใช้งาน
4, 6, 8, 10	GND
5	RESET
7	SCK
9	MISO

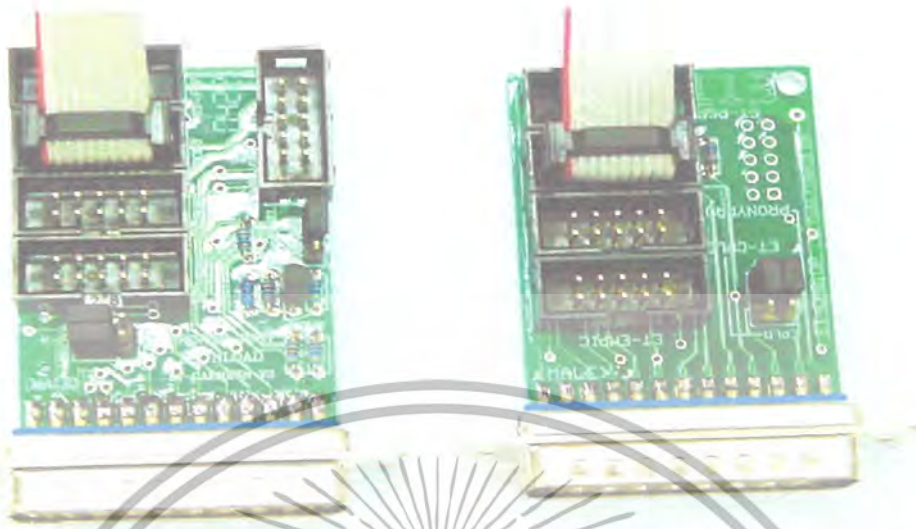


รูปที่2.16 การจัดเรียงขาสัญญาณ

2.3.5 การเชื่อมต่ออุปกรณ์สำหรับโปรแกรม Hex File

การโปรแกรมโค้ด(Hex File) ให้กับ AVR MCU ต้องใช้งานร่วมกับ ET-CAB10PIN และโปรแกรม PonyProg2000 โดยต่อ ET-CAP10PIN เข้ากับพอร์ต Printer พร้อมทั้งเลือก Jumper สำหรับใช้งานกับโปรแกรม PonyProg2000 แล้วต่อสาย Download ที่ขั้วต่อ ET-PSPI Download ของบอร์ด พร้อมทั้งจ่ายไฟเข้าบอร์ดให้เรียบร้อย ถ้ามีการต่ออุปกรณ์ภายนอกที่พอร์ต PB ให้ปลดออกก่อน โดยการเชื่อมต่อจะมีลักษณะดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การเลือก Jumper และการต่อสาย Download ของ ET-CAP10P(ซ้าย) V2.0(ขวา) V1.0 เพื่อใช้กับ AVR

2.4 โมดูลการสื่อสารไร้สาย DCBT-24AX

เป็น Module ตัวรับส่งข้อมูลผ่านคลื่นวิทยุระยะ ไกล ซึ่งมีความสามารถในการส่งข้อมูลที่รวดเร็ว และมีเสถียรภาพสูง ขนาดเล็ก ซึ่งมีตัวไมโครคอนโทรลเลอร์ AVR เบอร์ Mega168V อยู่บน Module เลยทำให้เราสามารถให้เป็นอุปกรณ์ควบคุมแบบไร้สาย แค่เพียงจ่ายไฟก็สามารถทำงานได้เลย

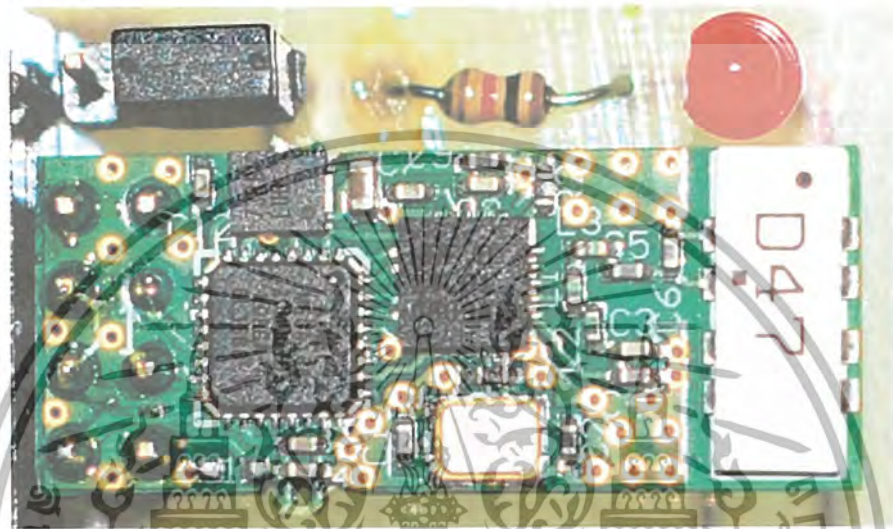
2.4.1 คุณสมบัติทั่วไป

- สามารถเป็นตัวรับและส่งในตัว
- ส่งได้ไกลถึง 30 m ในที่โล่ง
- ทำงานที่แรงดัน 2.7 – 5.25V
- กินกระแส 17 mA ขณะส่ง และกินกระแส 18 mA ขณะรับ
- ความถี่ในการส่ง 2.4GHz – 2.483GHz
- อัตราการส่ง 1Mbps – 2Mbps
- มีเสาอากาศในตัว
- ส่งได้ 84 Channel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

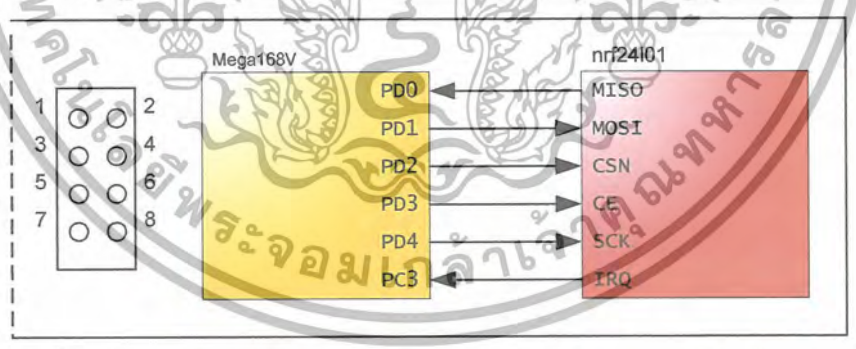
2.4.2 โครงสร้างการสื่อสารไร้สาย DCBT-24AX

ชุดโมดูล DCBT-24AX ประกอบด้วยไมโครคอนโทรลเลอร์ รุ่น AVR 8-bit RISC ATmega168V และชิพ nRF24L01 มีขนาดเล็ก น้ำหนักค่อนข้างเบาเหมาะสำหรับการใช้เป็นรีโมทเพื่อส่งสัญญาณไร้สาย



รูปที่ 2.18 โครงสร้างชุดโมดูล DCBT-24AX

2.4.3 ขั้วต่อสัญญาณต่างๆ



รูปที่ 2.19 การเชื่อมต่อการทำงานโมดูล DCBT 24-AX ระหว่าง ATmega168 กับ nRF24L01

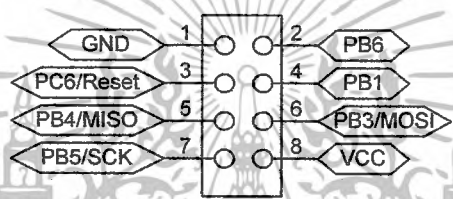
ตารางที่ 2.3 การทำงานของขาต่างๆ ระหว่าง AVR ATmega168V กับ nrf24l01

ตำแหน่งขา	หน้าที่
CE	เป็นขาควบคุมการทำงานของการรับ(Rx) ส่ง(Tx)
CSN	เป็นขาควบคุมการสื่อสารของ Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การทำงานของขาต่างๆ ระหว่าง AVR ATmega168V กับ nrf24l01(ต่อ)

ตำแหน่งขา	หน้าที่
SCK	เป็นขาสัญญาณนาฬิกา
MOSI	เป็นขารับข้อมูลจาก MCU
MISO	เป็นขาส่งข้อมูลไปให้ MCU
IRQ	เมื่อมีข้อมูลเข้ามาขานี้จะมีสถานะเป็น"0"



รูปที่ 2.20 ขาต่างๆ ของ DCBT-24AX

ขาที่ 1(GND) คือ ขา Ground ไว้ต่อสายดิน(0 V)

ขาที่ 2(PB6) คือ ขาสำหรับต่อพอร์ตใช้งานทั่วไป

ขาที่ 3(PC6/Reset) คือ ขาสำหรับโปรแกรมตัวชิพ ATmega168V

ขาที่ 4(PB1) คือ ขาสำหรับต่อพอร์ตใช้งานทั่วไป

ขาที่ 5(PB4/MISO) คือ ขาสำหรับโปรแกรมตัวชิพ ATmega168V

ขาที่ 6(PB3/MOSI) คือ ขาสำหรับโปรแกรมตัวชิพ ATmega168V

ขาที่ 7(PB5/SCK) คือ ขาสำหรับโปรแกรมตัวชิพ ATmega168V

2.5 ไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น ATmega168V

ATmega 168 เป็นไมโครคอนโทรลเลอร์ตระกูล AVR ของบริษัท Atmel มีคุณสมบัติดังนี้

สถาปัตยกรรมขั้นสูงแบบ RISC

- สถาปัตยกรรมภายในถูกออกแบบให้ใช้แบบ RISE(Reduce Instruction Set Computer) RISE คือ ทำให้การประมวลผลมีความเร็ว 1 คำสั่ง / 1 Clock หรือ CPU สามารถประมวลคำสั่งได้ 1 MIPS / MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดคำสั่ง 131 คำสั่ง โดยส่วนใหญ่ทำงานที่ 1 รอบสัญญานาฬิกา(clock cycle)
- รีจิสเตอร์ขนาด 8 Bit จำนวน 32 ตัว
- ความเร็วในการประมวลผลมากกว่า 20 ล้านคำสั่งต่อวินาที(Million Instructions Per Second : MIPS)

หน่วยความจำ

- แบบ Flash 16 Kbytes สามารถเขียน/ลบโปรแกรมได้ 10,000 ครั้ง
- แบบ EEPROM 512 Bytes สามารถเขียน/ลบโปรแกรมได้ 100,000 ครั้ง
- แบบ SRAM 1 Kbytes

ไฟเลี้ยง

- ระหว่าง 1.8 ถึง 5.5 VDC

ความถี่สัญญาณนาฬิกา

- 0 ถึง 4 MHz

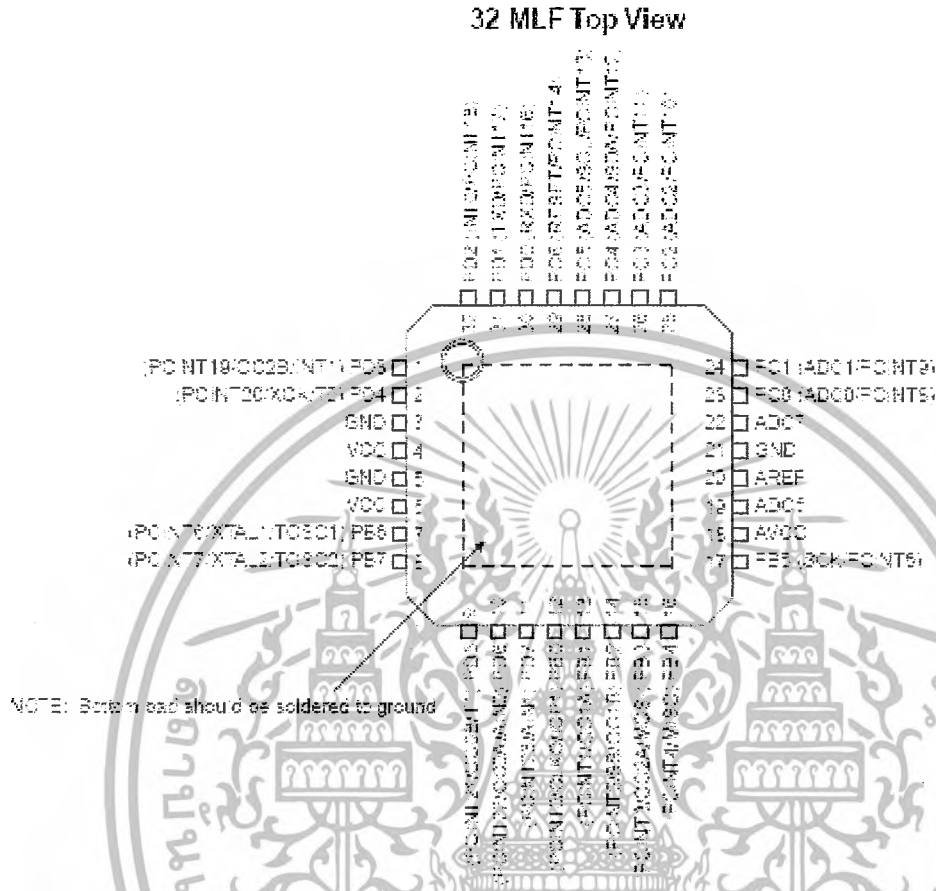
การรองรับอุปกรณ์ต่อพวง

- ตัวจับเวลา(Timer-counter) ขนาด 8 บิต 2 ตัว และขนาด 16 บิต 1 ตัว
- อุปกรณ์สื่อสารข้อมูลแบบอนุกรมแบบ SPI และ USART

อื่นๆ

- ระบบ Reset แบบอัตโนมัติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้าไมโครคอนโทรลเลอร์(Power-on Reset)
- ฟังก์ชั่นตรวจสอบแรงดัน(Brown-out Detection)
- ระบบการขัดจังหวะจากภายในและภายนอก(Internal and External Interrupt)
- ระบบตรวจจับการทำงานผิดพลาดของซีพียู(Watchdog Timer)
- โหมดอนุรักษ์พลังงาน 5 โหมด ได้แก่ Idle, ADC Noise Reduction, Power-save, Power-down และ Standby
- ช่วงอุณหภูมิการทำงาน -40°C ถึง 85°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 โครงสร้างของไมโครคอนโทรลเลอร์ ATmega168V แบบ 32 ขา

ขาที่ 1(CE) คือ ขาควบคุมการทำงานของการรับ(Rx) และการส่ง(Tx) ข้อมูล

ขาที่ 2(CSN) คือ ขาควบคุมการสื่อสารของโมดูลระหว่างไมโครคอนโทรลเลอร์ ATmega168V และชิพ nRF24L01

ขาที่ 3(SCK) คือ ขาสัญญาณนาฬิกา

ขาที่ 4(MOSI) คือ ขารับข้อมูลจากไมโครคอนโทรลเลอร์ ATmega168V

ขาที่ 5(MISO) คือ ขาส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ ATmega168V

ขาที่ 6(IRQ) คือ เมื่อมีข้อมูลเข้ามา ขานี้จะตั้งค่าสถานะเป็น “0”

ขาที่ 7(VDD) คือ ไฟเลี้ยง 1.9 V ถึง 3.6 V

ขาที่ 8(VSS) คือ ขา Ground ต่อสายดิน(0 V)

ขาที่ 9(XC2) คือ ขาต่อ Crystal2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ 10(XC1) คือ ขาต่อ Crystal1

ขาที่ 11(VDD_PA) คือ ไฟเลี้ยง 1.8 V ให้ภายในชิพ nRF24L01 และต้องเชื่อมต่อกับเสาอากาศทั้ง 2 ตัว(ANT1 และ ANT2)

ขาที่ 12(ANT1) คือ ขาต่อเสาอากาศ1

ขาที่ 13(ANT2) คือ ขาต่อเสาอากาศ2

ขาที่ 14(VSS) คือ ขา Ground ต่อสายดิน(0 V)

ขาที่ 15(VDD) คือ ไฟเลี้ยง 1.9 V ถึง 3.6 V

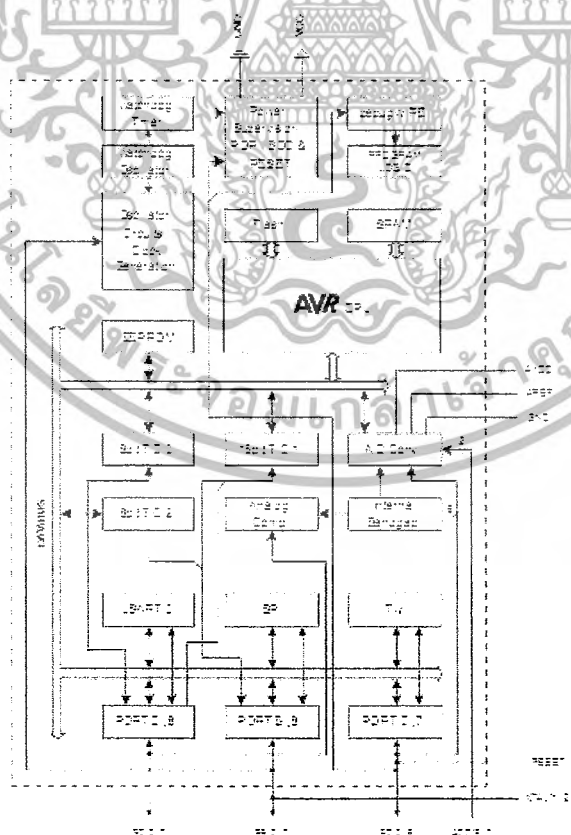
ขาที่ 16(IREF) คือ กระแสอ้างอิง โดยต่อกับตัวต้านทาน 22 kΩ ต่อลง Ground

ขาที่ 17(VSS) คือ ขา Ground ต่อสายดิน(0 V)

ขาที่ 18(VDD) คือ ไฟเลี้ยง 1.9 V ถึง 3.6 V

ขาที่ 19(DVDD) คือ ไฟเลี้ยงสำหรับ de-coupling

ขาที่ 20(VSS) คือ ขา Ground ต่อสายดิน(0 V)



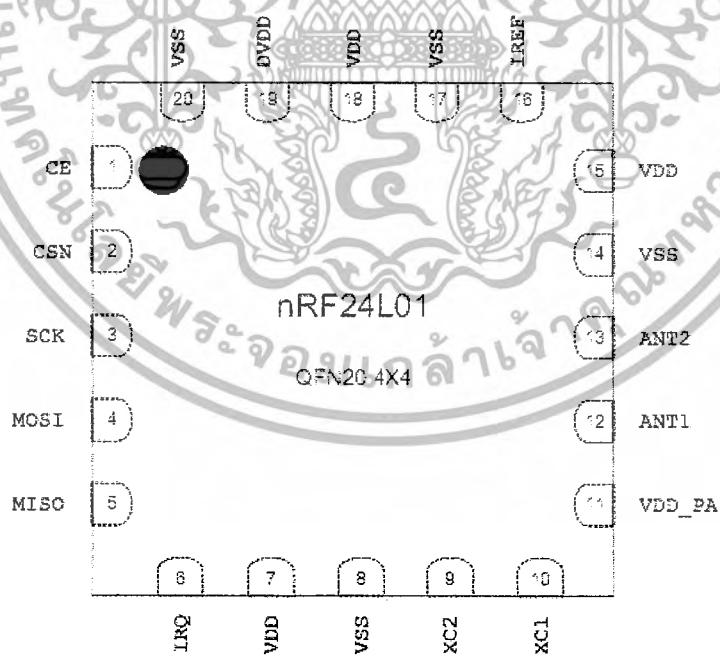
รูปที่ 2.22 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ ATmega168V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ชิพรับส่งสัญญาณไร้สาย nRF24L01

เป็นชิพไอซีที่ใช้สำหรับรับส่งข้อมูลในแบบอนุกรม ผ่านคลื่นวิทยุระยะไกล ใช้กับย่านความถี่ 2.4 GHz พร้อมมีเสาอากาศในตัว สามารถส่งข้อมูลรวดเร็ว มีเสถียรภาพสูง มีคุณสมบัติดังนี้

1. มีสถาปัตยกรรมแบบ QFN Package 20 Pin ขนาด 4x4 mm
2. ความถี่ในการใช้งานที่ 2.4 GHz
3. ความเร็วในการรับส่งข้อมูล 1 Mbps และ 2 Mbps
4. สามารถใช้เป็นตัวรับและส่งในตัว
5. มีรูปแบบความเร็วสูงในการรับส่งข้อมูล โดยส่งแบบ GFSK(Gaussian Frequency Shift Keying)
6. ทำงานที่ความต่างศักย์ทางไฟฟ้า 1.9 – 3.6 V
7. กำลังงานเอาต์พุตปรับได้ 4 ระดับ คือ 0 dBm, -6 dBm, -12 dBm และ -18 dBm
8. สามารถตั้งช่องความถี่ ในการใช้งานด้วยโปรแกรมได้ถึง 126 ช่องสัญญาณ
9. โหมดของการรับส่งข้อมูล เป็น โหมด ShockBurst เพราะเกิดการผิดพลาดในการส่งข้อมูลต่ำ ส่วนมากนิยมใช้ในการส่งข้อมูลชนิดที่ต้องการความถูกต้องสูง



รูปที่ 2.23 ลักษณะขาของชิพไอซี nRF24L01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 การทำงานของขาต่างๆ ของชิพ nRF24L01 ภายในโมดูล DCBT-24AX

ขาที่	ชื่อ	รายละเอียด
1	CE	ขาควบคุมการทำงานของการรับ(Rx) และการส่ง(Tx) ข้อมูล
2	CSN	ขาควบคุมการสื่อสารของโมดูล ระหว่างไมโครคอนโทรลเลอร์ ATmega168V และ ชิพ nRF24L01
3	SCK	ขาสัญญาณนาฬิกา
4	MOSI	ขารับข้อมูลจากไมโครคอนโทรลเลอร์ ATmega168V
5	MISO	ขาส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ ATmega168V
6	IRQ	เมื่อมีข้อมูลเข้ามา ขานี้จะตั้งค่าสถานะเป็น “0”
7	VDD	ไฟเลี้ยง 1.9 V ถึง 3.6 V
8	VSS	ขา Ground ต่อสายดิน(0 V)
9	XC2	ขาต่อ Crystal2
10	XC1	ขาต่อ Crystal1
11	VDD_PA	ไฟเลี้ยง 1.8 V ให้ภายในชิพ nRF24L01 และต้องเชื่อมต่อกับเสาอากาศทั้ง 2 ตัว(ANT1 และ ANT2)
12	ANT1	ขาต่อเสาอากาศ1
13	ANT2	ขาต่อเสาอากาศ2
14	VSS	ขา Ground ต่อสายดิน(0 V)
15	VDD	ไฟเลี้ยง 1.9 V ถึง 3.6 V
16	IREF	กระแสอ้างอิง โดยต่อกับตัวต้านทาน 22 kΩ ต่อลง Ground
17	VSS	ขา Ground ต่อสายดิน(0 V)
18	VDD	ไฟเลี้ยง 1.9 V ถึง 3.6 V
19	DVDD	ไฟเลี้ยงสำหรับ de-coupling
20	VSS	ขา Ground ต่อสายดิน(0 V)

2.6.1 คำสั่งการตั้งค่าการใช้งานโมดูล DCBT-24AX

ในการตั้งค่าใช้งานเริ่มต้นของโมดูล DCBT-24AXนั้น มีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 คำสั่งต่างๆ ของอุปกรณ์รับ-ส่งข้อมูล ชิพ nRF24L01

คำสั่ง	Code	Comment
อ่านค่าจาก Register	0x00 + Address	Address = 0-0x1F
เขียนค่าลง Register	0x20 + Address	Address = 0-0x1F
อ่านค่าจาก Buffer การรับ	0x61	อ่านได้สูงสุด 32 ไบต์
เขียนค่าลง Buffer การส่ง	0xA0	เขียนได้สูงสุด 32 ไบต์
ล้าง Buffer ของการรับเข้า	0xE2	-
ล้าง Buffer ของการส่งออก	0xE1	-
อ่านสถานะ	0xFF	-

ตารางที่ 2.6 ตำแหน่ง Registers ที่เกี่ยวข้อง

ตำแหน่ง	ชื่อ Register	Comment
00	Configuration	ตั้งค่าการทำงานของชิพ nRF24L01
06	RF Gain	ตั้งค่ากำลังการส่ง
0A	RX Address P0	ตั้งรหัสของตัวรับหมายเลข 0
0B	RX Address P1	ตั้งรหัสของตัวรับหมายเลข 1
10	TX Address	ตั้งรหัสของตัวส่ง
11	RX Pipeline P0	ตั้งค่า Buffer ของการรับหมายเลข 0
12	RX Pipeline P1	ตั้งค่า Buffer ของการรับหมายเลข 1

Configuration(00) การตั้งค่าการทำงานของชิพ nRF24L01 จำนวน 8 บิต โดยแต่ละบิตมีหน้าที่ ดังนี้

ตารางที่ 2.7 การตั้งค่า Configuration(00)

บิตที่	การทำงาน
7	ไม่ได้ใช้
6	"0" คือ เมื่อได้รับข้อมูลแล้ว จะให้ขา IRQ เป็น "0"
5	"0" คือ เมื่อส่งข้อมูลเสร็จแล้ว จะให้ขา IRQ เป็น "0"
4	"0" คือ เมื่อส่งข้อมูลค่าเดิมเสร็จแล้ว จะให้ขา IRQ เป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 การตั้งค่า Configuration(00)(ต่อ)

บิตที่	การทำงาน
3	"1" คือ ให้มีการตรวจสอบ CRC
2	จำนวนไบต์ตัวถอดรหัส CRC, "0" คือ ข้อมูล 1 ไบต์ และ "1" คือ ข้อมูล 2 ไบต์
1	"1" คือ Power Up และ "0" คือ Power Down
0	"1" คือ ตั้งให้เป็นตัวรับข้อมูล และ "0" คือ ตั้งให้เป็นตัวส่งข้อมูล

RF Gain(06) การตั้งค่ากำลังการส่งของชิพ nRF24L01 จำนวน 8 บิต โดยแต่ละบิตมีหน้าที่ ดังนี้

ตารางที่ 2.8 การตั้งค่า RF Gain(06)

บิตที่	การทำงาน
7	ไม่ได้ใช้ มีค่าเป็น "0"
6	ไม่ได้ใช้ มีค่าเป็น "0"
5	ไม่ได้ใช้ มีค่าเป็น "0"
4	ใช้สำหรับการทดสอบอุปกรณ์
3	ตั้งอัตราการส่งข้อมูล "0" คือ 1 Mb/s , "1" คือ 2 Mb/s
2	ตั้งค่ากำลังวัตต์ของการส่ง "00" คือ -18 dBm, "01" คือ -12 dBm, "10" คือ -6 dBm , "11" คือ 0 dBm
1	
0	เป็น "1" เมื่อต้องการตั้งค่ากำลังการส่ง

Status(07) สถานะของชิพ nRF24L01จำนวน 8 บิต โดยแต่ละบิตมีหน้าที่ ดังนี้

ตารางที่ 2.9 การแสดงสถานะของชิพ nRF24L01(07)

บิตที่	การทำงาน
7	สงวนไว้
6	เป็น "1" เมื่อได้รับข้อมูล และจะถูกยกเลิกเมื่อเขียนค่า "1" ลงไป
5	เป็น "1" เมื่อได้ส่งข้อมูล และจะถูกยกเลิกเมื่อเขียนค่า "1" ลงไป
4	เป็น "1" เมื่อได้ส่งข้อมูลเกินกว่า Buffer ของตัวส่ง และจะยกเลิกเมื่อเขียนค่า "1" ลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 การแสดงสถานะของชิพ nRF24L01(07)(ต่อ)

บิตที่	การทำงาน
3	จำนวนข้อมูลของ Pipeline สำหรับการเตรียมข้อมูล เพื่อการอ่านค่าจาก Buffer การรับ 000 – 101 : จำนวนข้อมูลของ Pipeline 110 : ไม่ได้ใช้ 111 : Buffer การรับว่างเปล่า
2	
1	
0	เป็น “1” เมื่อ Buffer การส่งเต็ม

Read Status(FF) อ่านสถานะของชิพ nRF24L01จำนวน 8 บิต โดยแต่ละบิตมีหน้าที่ ดังนี้

ตารางที่ 2.10 การอ่านสถานะของชิพ nRF24L01(FF)

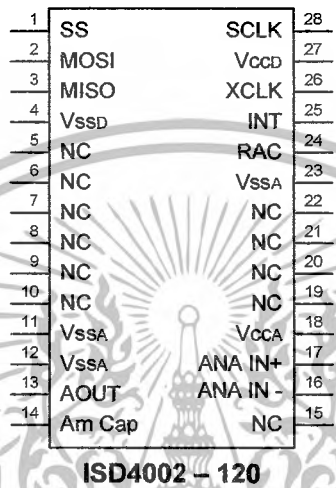
บิตที่	การทำงาน
7	ไม่ได้ใช้ มีค่าเป็น “0”
6	เป็น “1” เมื่อต้องการส่งข้อมูลล่าสุดออกไปอีกครั้ง และจะถูก Reset เป็น “0” เมื่อมีการล้างข้อมูลใน Buffer ของการส่ง
5	เป็น “1” เมื่อ Buffer ของการส่งเต็ม
4	เป็น “1” เมื่อ Buffer ของการส่งว่างเปล่า
3	ไม่ได้ใช้ มีค่าเป็น “0”
2	ไม่ได้ใช้ มีค่าเป็น “0”
1	เป็น “1” เมื่อ Buffer การรับเต็ม
0	เป็น “1” เมื่อ Buffer การรับว่างเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 บอร์ดบันทึกเสียง ISD4002-120

เป็นไอซีที่ทำหน้าที่บันทึกและเล่นเสียง(record/play) ใช้ไอซีบันทึกเสียงตระกูล ISD4002-120

2.7.1 ขั้วต่อสัญญาณต่างๆ



รูปที่ 2.24 ขาต่างๆ ภายในไอซี ISD4002-120

ตารางที่ 2.11 รายละเอียดการทำงานของขาต่างๆ ภายในไอซี ISD4002-120

ตำแหน่งขา	หน้าที่
SS,SCLK,MISO,MOSI	เป็นขาสัญญาณที่ใช้ในการควบคุม ISD4002
XCLK	เป็นขาที่รับสัญญาณนาฬิกาเพื่อการ Sampling สัญญาณเสียง แต่โดยปกติแล้วเราจะใช้สัญญาณนาฬิกาภายใน ดังนั้นถ้าขานี้ไม่ใช้จะต่อลง Ground
INT	ขานี้จะเป็นลอจิก “0” เมื่อเล่นจนหมดหน่วยความจำ หรือเล่นจนหมดในแต่ละ ข้อความนั้น
ANA IN+, ANA IN-	เป็นขาอินพุทเพื่อรับสัญญาณเสียงจากภายนอกเพื่อการบันทึกเสียง
AOUT	สัญญาณเสียงจะออกจากขานี้ เมื่ออยู่ในโหมดของการ Play
AM CAP	เป็นขาที่ใช้ในการลดสัญญาณรบกวนขณะเล่นเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

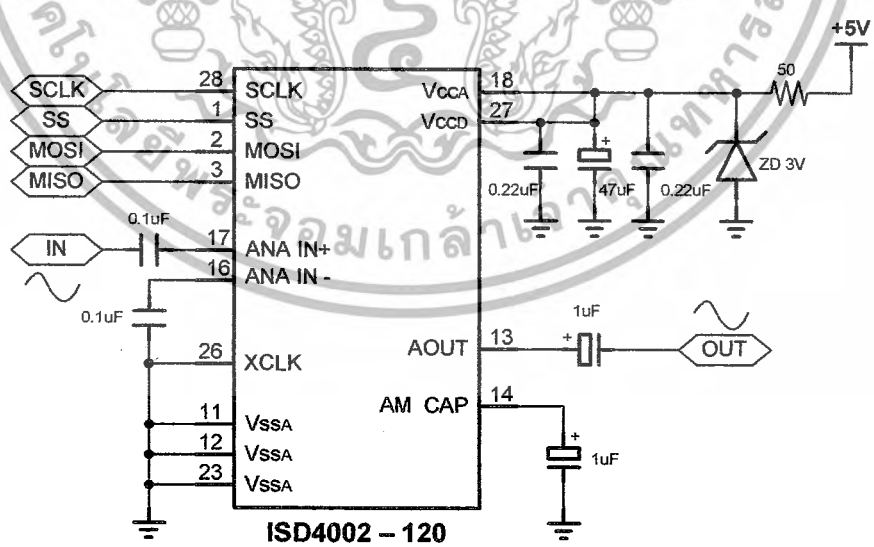
ตารางที่ 2.12 รายละเอียดการทำงานของขาต่างๆ ภายในไอซี ISD4002-120(ต่อ)

VccD,VccA	เป็นขาไฟเลี้ยงของไอซีทำงานที่แรงดัน 3 V
VssD,VssA	เป็นขา Ground ของไอซี
NC	ไม่ต้องต่อ

2.7.2 คุณสมบัติทั่วไป

- สามารถเล่นและบันทึกเสียงได้ในตัวเดียว
- ทำงานที่แรงดัน 3 V
- กินกระแส 15 mA ขณะเล่น และกินกระแส 25 mA ขณะบันทึก
- บันทึกได้นาน 2, 3, 4 นาที ตามขนาดของไอซี
- บันทึกซ้ำได้มากกว่า 100,000 ครั้ง
- จดจำได้นานถึง 100 ปี
- ติดต่อสื่อสารแบบ SPI(Serial Peripheral Interface)

2.7.3 โครงสร้าง nRF24L01



รูปที่ 2.25 วงจรการทำงานของไอซีบันทึกเสียง ISD4002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.12 รายละเอียดการทำงานของขาต่างๆ ภายในไอซี ISD4002-120

ตำแหน่งขา	หน้าที่
SS,SCLK,MISO,MOSI	เป็นขาสัญญาณที่ใช้ในการควบคุม ISD4002
XCLK	เป็นขาที่รับสัญญาณนาฬิกาเพื่อการ Sampling สัญญาณเสียง แต่โดยปกติแล้วเราจะใช้สัญญาณนาฬิกาภายใน ดังนั้นถ้าขานี้ไม่ใช้จะต่อลง Ground
INT	ขานี้จะเป็นลอจิก "0" เมื่อเล่นจนหมดหน่วยความจำ หรือเล่นจนหมดในแต่ละ ข้อความนั้น
ANA IN+, ANA IN-	เป็นขาอินพุตเพื่อรับสัญญาณเสียงจากภายนอกเพื่อการบันทึกเสียง
AOUT	สัญญาณเสียงจะออกจากขานี้ เมื่ออยู่ในโหมดของการ Play
AM CAP	เป็นขาที่ใช้ในการลดสัญญาณรบกวนขณะเล่นเสียง
VccD,VccA	เป็นขาไฟเลี้ยงของไอซีทำงานที่แรงดัน 3 V
VssD,VssA	เป็นขา Ground ของไอซี
NC	ไม่ต้องต่อ

2.7.4 คำสั่งต่างๆ ของ ISD4002

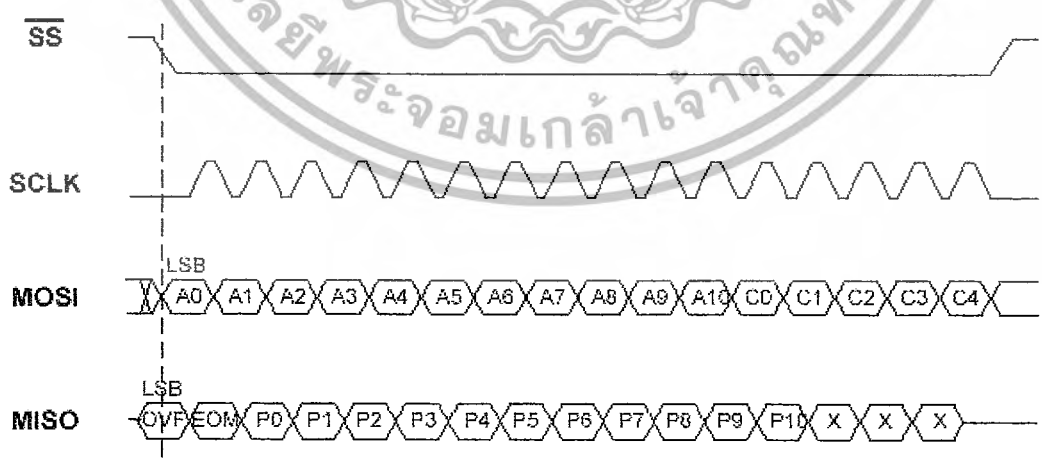
- Power Up : เริ่มต้นทำงาน
- SETPLAY : กำหนด Address ที่จะเล่น
- PLAY : เล่นเสียง
- SETREC : กำหนด Address ที่จะบันทึก
- REC : บันทึกเสียง
- STOP : หยุดการเล่นหรือบันทึกเสียง
- Power Down : หยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.13 รายละเอียดคำสั่งต่างๆ ของ ISD4002

คำสั่ง	คำสั่ง					แอดเดรส					ข้อมูล
	C4	C3	C2	C1	C0	A10	A9	.	.	A0	
Power Up	0	0	1	0	0	000 0000 0000					0x2000
SETPLAY	1	1	1	0	0	A10 – A0					0xExxx
PLAY	1	1	1	1	0	000 0000 0000					0xF000
SETREC	1	0	1	0	0	A10 – A0					0xAxxx
REC	1	0	1	1	0	000 0000 0000					0xB000
STOP	0	0	1	1	0	000 0000 0000					0x3000
PWRDOWN	0	0	0	1	0	000 0000 0000					0x1000

2.7.4 รูปแบบการสื่อสารของ ISD4002



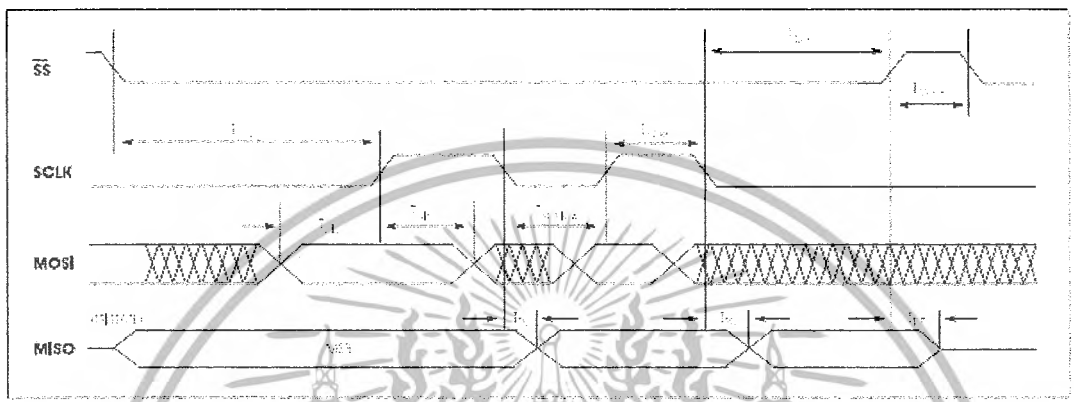
รูปที่ 2.26 ลักษณะคลื่นสัญญาณของการสื่อสารของ ISD4002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OVF - Overflow เป็น “1” เมื่อเล่นจนหมดหน่วยความจำ

EOM - End of Massage เป็น “1” เมื่อ สิ้นสุดข้อความเสียงที่เล่น

P10-P0 - เป็น Address ปัจจุบันที่เล่นหรือบันทึกอยู่



รูปที่ 2.27 แสดง timing diagram ของการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการคำนวณ

3.1 การออกแบบโครงงาน

รูปที่ 3.1 บล็อกไดอะแกรมของโครงงานทั้งหมด

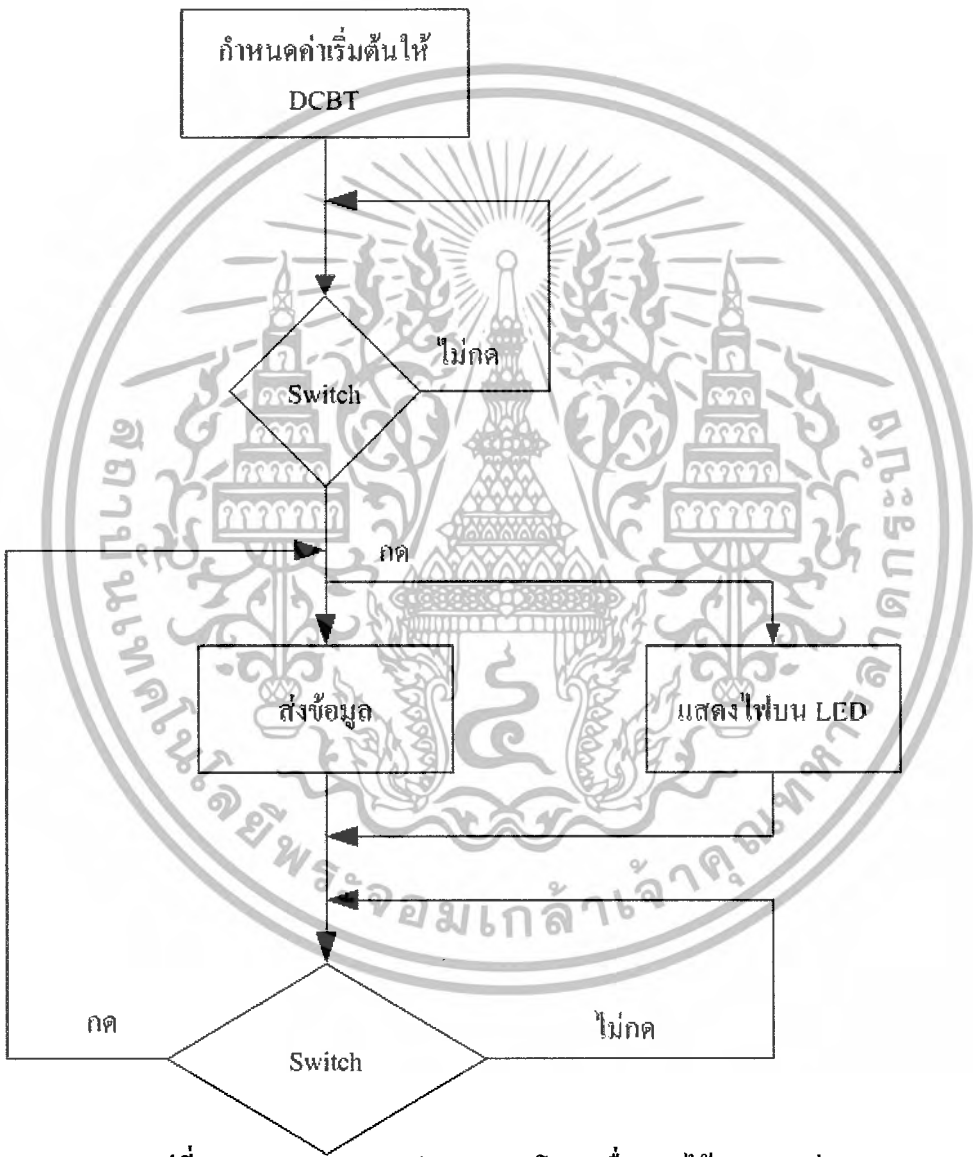
ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อประยุกต์ใช้โมดูลการสื่อสารแบบไร้สาย และไอซีบันทึกเสียงทำงานร่วมกันโดยผ่านไมโครคอนโทรลเลอร์ตระกูล AVR ซึ่งการทำงานแบ่งได้เป็นสามส่วน คือส่วนแรกเป็นอุปกรณ์รับส่งสัญญาณไร้สาย เพื่อตรวจสอบว่าตัวส่งสัญญาณไร้สายได้ถูกปล่อยสัญญาณมาเมื่อใด ในส่วนนี้จะใช้โมดูล DCBT-24AX ที่ภายในประกอบด้วยไมโครคอนโทรลเลอร์ AVR เบอร์ Mega168V อยู่บนโมเดล เชื่อมต่อกับชิพ nRF24L01 เป็นโมดูลการส่งสัญญาณไร้สาย ในส่วนที่สองเป็นไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega128 ซึ่งใช้เป็นตัวจำลองสัญญาณไฟจราจร โดยเชื่อมต่อกับ DCBT-24AX ที่ขาอินเตอร์รัพ เมื่อได้รับสัญญาณไฟจาก DCBT-24AX จะทำการเรียกใช้งานไอซีบันทึกเสียงซึ่งอยู่ในส่วนที่สาม โดยใช้ไอซีบันทึกเสียงตระกูล ISD4002-120 ทำการบันทึกเสียงสถานะของสัญญาณไฟจราจรไว้ เมื่อได้รับสัญญาณไฟจาก ATmega128 จะเล่นไฟล์เสียงที่ทำการบันทึกไว้ผ่านลำโพง เพื่อให้สามารถใช้งานอุปกรณ์ได้ตามขอบเขตที่กำหนดไว้ควรศึกษาข้อมูลทฤษฎี และวิธีการใช้งานอุปกรณ์ต่างๆ ตามลักษณะการทำงานดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โมดูลการสื่อสารไร้สาย

ชุดโมดูลการสื่อสารไร้สาย DCBT-24AX จะถูกแบบเป็นสองส่วน ได้แก่ตัวส่งสัญญาณ และตัวรับสัญญาณ

3.2.1 ตัวส่งสัญญาณ

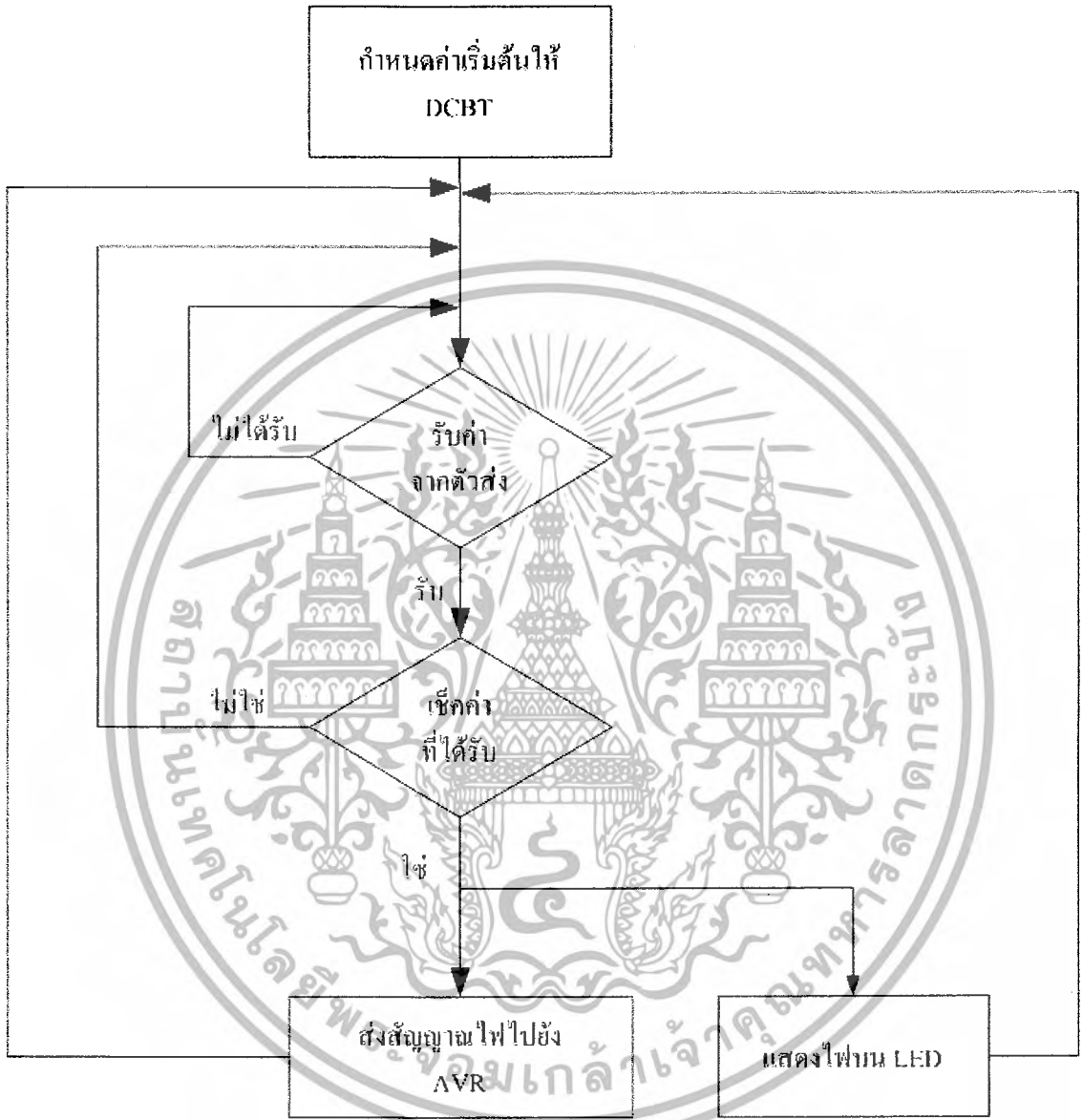


รูปที่ 3.2 Flowchart การทำงานของโมดูลสื่อสารไร้สายภาคส่ง

โมดูลการสื่อสารไร้สายภาคส่งจะรอจนกว่ามีการกดสวิทซ์ เมื่อมีการกดสวิทซ์จะทำการปล่อยสัญญาณไร้สายไปยังภาครับ พร้อมกับแสดงไฟ LED เพื่อแสดงให้ทราบว่าตัวส่งสัญญาณได้ทำการส่งสัญญาณแล้ว และจะทำการหยุดส่งสัญญาณเมื่อทำการปล่อยสวิทซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โมดูลในการส่งสัญญาณไร้สายภาครับ



รูปที่ 3.3 Flow Chart การส่งสัญญาณไร้สายภาครับ

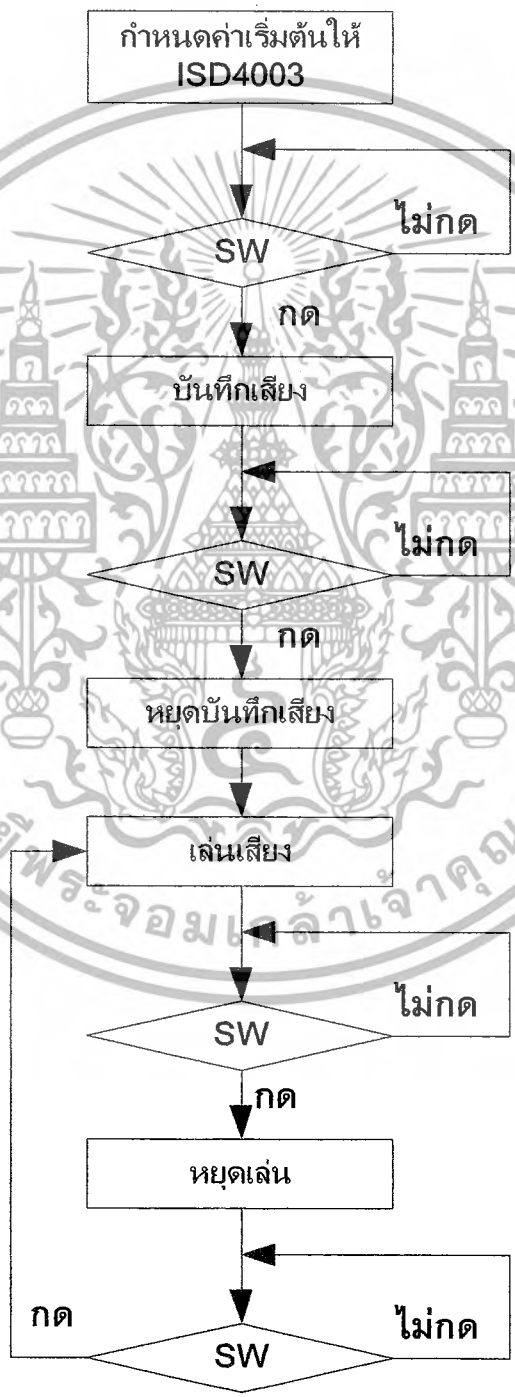
โมดูลการสื่อสารไร้สายภาครับ จะทำการตรวจจับว่ามีการส่งสัญญาณมาจากภาคส่งหรือไม่ หากตรวจจับได้ว่ามีการส่งสัญญาณมาจากภาคส่งแล้ว จะนำค่าที่ได้รับมาจากภาคส่งมาทำการตรวจสอบว่ามีค่าตรงตามกับที่กำหนดไว้ร่วมกันในตอนแรกหรือไม่ เพื่อเป็นการป้องกันไม่ให้เกิดการรับสัญญาณจากตัวส่งสัญญาณอื่น จากนั้นจึงทำการส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ตระกูล AVR พร้อมกับแสดงผลไฟ LED เพื่อเป็นการบอกให้ทราบว่ารับสัญญาณได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ISD–4002

ก่อนนำไอซีบันทึกเสียงมาใช้งานร่วมกับ AVR จะต้องทำการบันทึกเสียงเสียก่อน โดยจะต้องเขียนโปรแกรมเพื่อทำการบันทึกเสียง ซึ่งสามารถอธิบายได้ดังนี้

3.3.1 วิธีการบันทึกเสียงลงไอซี ISD-4002

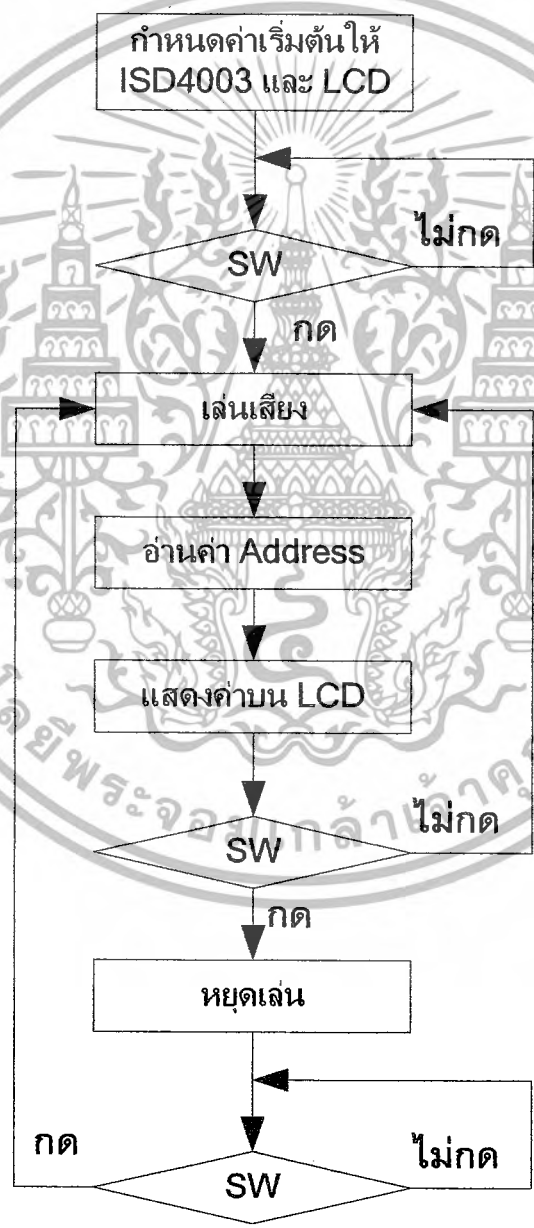


รูปที่ 3.4 Flow Chart การบันทึกเสียงลงไอซี ISD-4002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราทำการกดสวิทซ์ INT0 โดยมี LED เป็นตัวบอกสถานะการบันทึก ในขณะเดียวกันก็ให้เราปล่อยสัญญาณเสียง จาก Line Out ของ Computer และถ้าเราต้องการหยุดบันทึกและเล่นเสียงกลับ ก็ให้เรากดปุ่ม INT0 อีกครั้งหนึ่ง สัญญาณเสียงจะออกจากบอร์ดของเรา ซึ่งเป็นเสียงที่เราบันทึกไว้เมื่อสักครู่นี้ และถ้าเราต้องการหยุดเล่นก็สามารถกดปุ่ม INT0 ได้เรื่อยๆ

3.3.2 วิธีการอ่านตำแหน่งที่บันทึกเสียงลงไอซี



รูปที่ 3.5 Flow Chart การอ่านตำแหน่งที่บันทึกเสียง

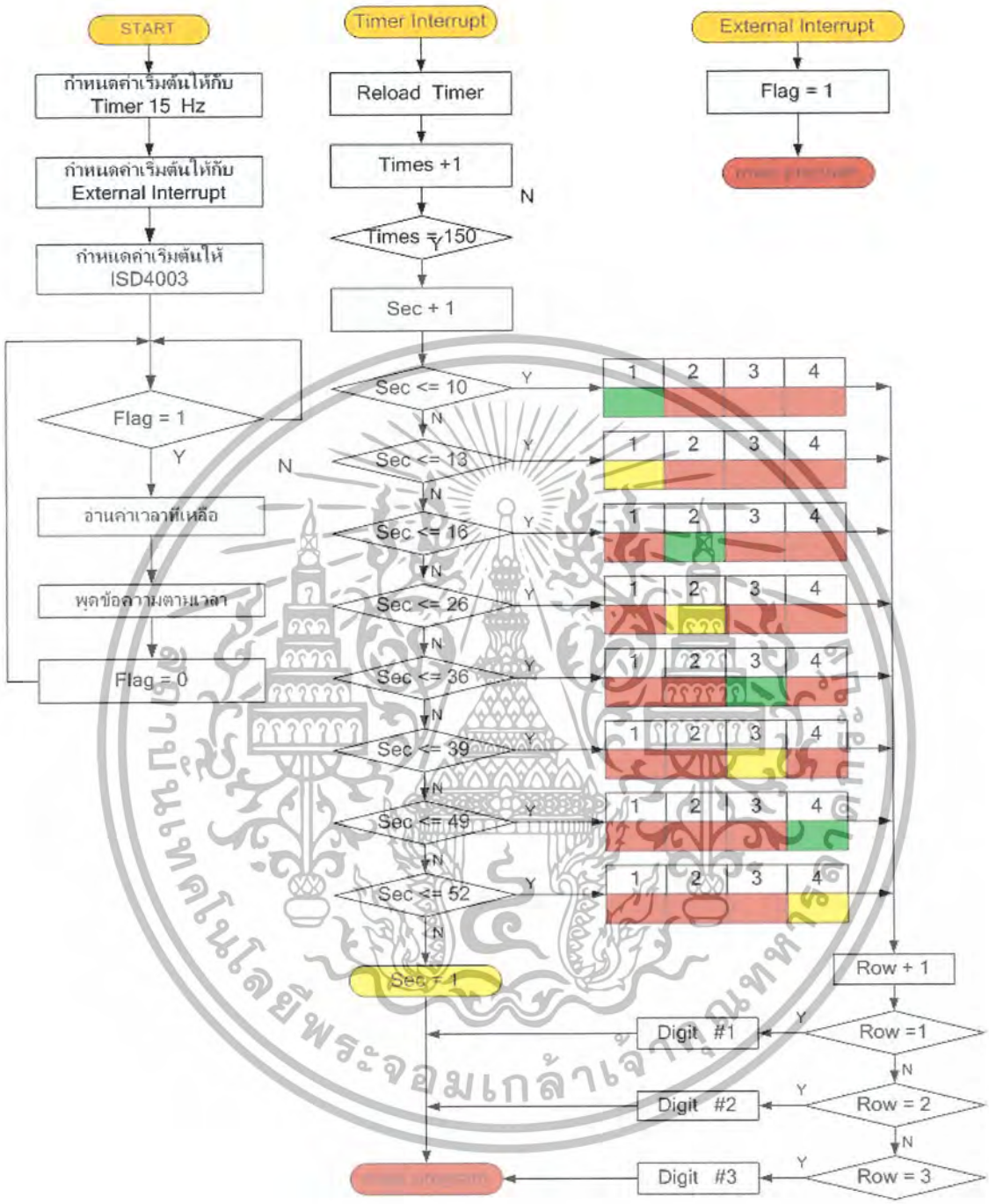
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เราบันทึกเสียงเรียบร้อยแล้ว ซึ่งเสียงเราบันทึกนั้นเป็นข้อความที่ต่อเนื่อง แต่ถ้าเกิดเราต้องการเล่นข้อความบางช่วงบางข้อความ เราสามารถทำได้โดยต้องรู้ว่าข้อความที่เราบันทึก อยู่ในช่วง Address ไหนของหน่วยความจำที่อยู่ใน ISD4002 เราจะเรียกใช้ฟังก์ชัน Play ที่สามารถส่งค่า Address ปัจจุบันที่กำลังเล่นอยู่มาแสดงบน LCD ทำให้เรารู้ว่าข้อความที่เราต้องการนั้นอยู่ที่ Address ไหน

3.4 ไมโครคอนโทรลเลอร์ AVR ATmega128

ส่วนนี้เป็นส่วนที่เชื่อมโยงอุปกรณ์ทุกส่วนไว้ด้วยกัน โดยส่วนแรกคือการใช้ AVR จำลองการทำงานของสัญญาณไฟจราจร ส่วนที่สองคือโมดูลการรับส่งสัญญาณไร้สาย ซึ่งจะนำสัญญาณไฟจากตัวรับสัญญาณไร้สายต่อเข้ากับขา INT0 ของ AVR เมื่อมีการส่งสัญญาณเข้ามาจะทำการอินเทอร์รัพการทำงานของ AVR โดย AVR จะส่งค่าสถานะไฟสัญญาณจราจรไปยังไอซีบันทึกเสียงเพื่อให้ไอซีบันทึกเสียงเล่นไฟล์เสียงจากตำแหน่งที่ตรงกับค่าที่ได้รับมาจาก AVR ผ่านลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Flow Chart การทำงานของ AVR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 ผลการทดลองตามวัตถุประสงค์หลัก

เนื่องจากอุปกรณ์ที่ใช้ในการสื่อสารระหว่างผู้พิการทางสายตากับเสาไฟสัญญาณจราจรเป็นรีโมทตัวส่งสัญญาณไร้สาย DCBT-24AX ซึ่งมีระยะในการส่งตามค่าที่กำหนดไว้ในโปรแกรมโดยที่เราไม่สามารถรู้ได้ว่าการกำหนดค่าต่างๆ จะส่งผลกับระยะทางในการรับส่งเท่าไรบ้าง กอรปกับโครงสร้างของระบบทั้งหมด คือการที่ตัวรับสัญญาณไร้สายรอรับสัญญาณจากตัวส่งสัญญาณไร้สาย ซึ่งจะทำการติดตั้งไว้ตามเสาไฟสัญญาณจราจร โดยที่เสาไฟสัญญาณจราจรแต่ละเสามีระยะห่างประมาณ 20 เมตร และควบคุมไม่ให้ระยะส่งสั้นเกินไป เพราะจะทำให้ผู้พิการทางสายตาต้องเดินเข้ามาใกล้เสาไฟสัญญาณจราจรมากเกินไปจนอาจทำให้เกิดอันตรายได้ ดังนั้นจึงต้องทดลองหาค่าที่ควบคุมระยะในการส่งให้อยู่ในระยะที่เหมาะสม

ตารางที่ 4.1 การตั้งค่ากำลังการส่งของ RF Module

บิทที่	การทำงาน	ค่า
7	ไม่ได้ใช้	0
6	ไม่ได้ใช้	0
5	ไม่ได้ใช้	0
4	ใช้สำหรับการทดสอบอุปกรณ์	0
3	อัตราการส่งข้อมูล 0 = 1Mb/s, 1 = 2Mb/s	1
2	ตั้งค่ากำลังวัตต์ของการส่ง 00 = -18 dbm, 01 = -12 dbm 10 = -6 dbm, 11 = 0 dbm	0
1		0
0	1 = เมื่อต้องการตั้งค่ากำลังการส่ง	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.1 การใช้ค่ากำลังการส่งของ RF Module นำมาการทดลองป้อนค่าและวัดระยะ

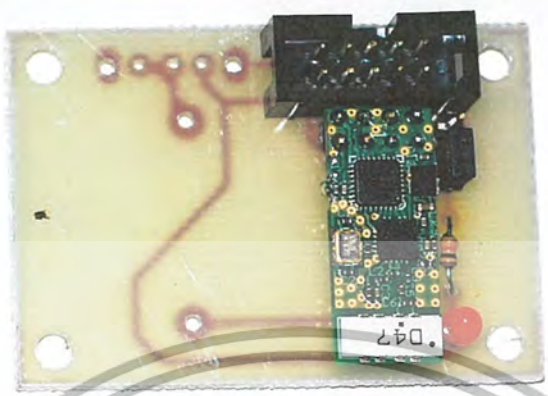
ตารางที่ 4.2 การทดลองค่ากำลังการส่งของ RF Module

Rf gain	ระยะทาง(เมตร)
0x09	3
0x0B	4
0x0D	5
0xFF	6
0x01	6
0x03	8
0x05	10
0x07	12

จากตารางที่ 4.2 จึงเลือกค่า 0x0b เพราะเป็นระยะที่เหมาะสมในการปล่อยสัญญาณจากรีโมท ไปยังตัวรับ

รูปที่ 4.1 ตัวส่งสัญญาณไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ตัวรับสัญญาณ แบบไม่มีสวิตซ์

รูปที่ 4.3 ตัวรับสัญญาณ แบบมีสวิตซ์

จากรูปด้านบน รูปที่ 4.2 เป็นตัวรับสัญญาณแบบไม่มีสวิตซ์ ทำงานได้โดยใช้ไฟเลี้ยงจากไมโครคอนโทรลเลอร์ AVR ATmega64/128 ส่วนในรูปที่ 4.3 เป็นตัวรับสัญญาณแบบมีสวิตซ์ โดยทำงานได้โดยใช้ไฟจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและแนวทางการพัฒนา

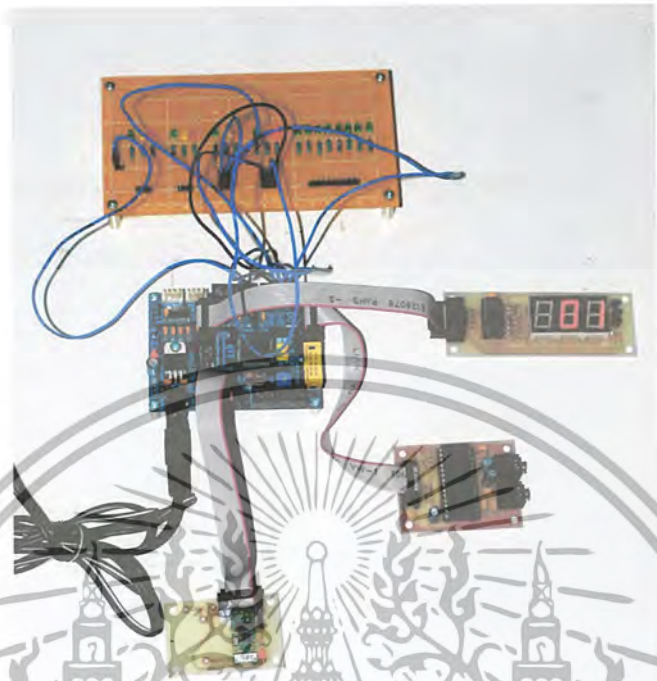
5.1 สรุปผล

จากวัตถุประสงค์ของปริญญานิพนธ์นี้คือการช่วยเหลือผู้พิการทางสายตาข้ามถนน จึงต้องสร้างอุปกรณ์ที่ง่ายต่อการใช้งาน เช่น รีโมท โดยใช้การส่งสัญญาณแบบไร้สาย ดังนั้นจึงเลือกใช้งานโมดูลการรับส่งสัญญาณไร้สาย DCBT-24AX เพราะใช้ไฟน้อย มีขนาดเล็ก พกพาสะดวก และในส่วนของไอซีบันทึกเสียง ได้เลือกใช้งาน ISD-4002 เนื่องจากสามารถบันทึกเสียงได้ในตัว มีการเล่นไฟล์เสียงที่เร็ว และวิธีใช้งานไม่ซับซ้อน

ในปริญญานิพนธ์นี้ใช้งานไมโครคอนโทรลเลอร์ AVR ATmega128 เป็นอุปกรณ์เชื่อมโยงส่วนต่างๆ เข้าด้วยกัน เนื่องจากมีพอร์ตการทำงานหลายพอร์ตจึงเหมาะกับปริญญานิพนธ์นี้เพราะต่อเชื่อมโยงกับอุปกรณ์หลายชิ้น และมีโปรแกรมที่ช่วยในการพัฒนาที่ใช้งานง่าย เช่น CodeVisionAVR C Compiler, AVR Studio

เมื่อเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกัน ไมโครคอนโทรลเลอร์จะจำลองการทำงานของสัญญาณไฟจราจร ประกอบด้วยไฟสัญญาณจราจร และไฟ 7 segment ที่ใช้ในการนับถอยหลังสถานะของไฟสัญญาณจราจร ที่ตัวรับสัญญาณไร้สายและรีโมทตัวส่งสัญญาณไร้สายได้ติดไฟ LED ไว้ เพื่อให้ทราบว่าเมื่อกดสวิทซ์ที่ตัวส่ง และเมื่อตัวรับได้รับสัญญาณ อุปกรณ์ได้ทำงานอย่างถูกต้องหรือไม่ เมื่อตัวรับสัญญาณได้รับสัญญาณจากรีโมทตัวส่งสัญญาณไร้สายที่ให้ผู้พิการทางสายตาไว้แล้ว จะทำการส่งค่าไปยังไมโครคอนโทรลเลอร์ หลังจากนั้นไมโครคอนโทรลเลอร์จะส่งสถานะไฟสัญญาณจราจรไปยังไอซีบันทึกเสียง และไอซีบันทึกเสียงก็จะเล่นไฟล์เสียงตามสถานะไฟสัญญาณจราจรผ่านลำโพง เพื่อให้ผู้พิการทางสายตาตัดสินใจว่าจะข้ามหรือไม่ข้ามถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 การเชื่อมต่ออุปกรณ์ทั้งหมดของโครงงาน

5.2 วิจารณ์

การใช้งานอุปกรณ์ต่างๆ ในปริญญานิพนธ์นี้ในชีวิตจริงอาจจะเกิดปัญหาได้ เช่น การถูกรบกวนจากสัญญาณคลื่นวิทยุทำให้ตัวรับสัญญาณไร้สายเกิดความผิดพลาดได้ และเสียงของลำโพงอาจจะถูกเสียงของผู้คนหรือเสียงของรถยนต์กลบเอาได้

5.3 แนวทางการพัฒนา

สามารถพัฒนาให้รีโมทเพิ่มกำลังส่ง และเพิ่มตัวรับไว้ในจุดเสี่ยงอันตรายต่างๆ เมื่อเวลาที่ผู้พิการทางสายตา หรือคนชราได้รับอุบัติเหตุ เพียงกดรีโมทก็จะเป็นการแจ้งไปยังสถานีตำรวจเพื่อให้ได้รับการรักษาอย่างทันท่วงที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ประจิน พลังสันติกุล. **C Programming for AVR Microcontroller and WinAVR(C Compiler)**.กรุงเทพ : แอพซอฟต์เทค, 2549.
2. ประภาพร ช่างไม้. **คู่มือเขียนโปรแกรมภาษา C.** กรุงเทพฯ : วิตตี้ กรุ๊ป, 2551.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

คู่มือการใช้งานอุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

คู่มือการใช้งาน

1.1 การเตรียมอุปกรณ์

1. ติดตั้งไอซีบันทึกเสียงเข้ากับอุปกรณ์จำลองสัญญาณไฟจราจร และนำไอซีบันทึกเสียงต่อเข้ากับลำโพงอีกที

2. ติดตั้งตัวรับสัญญาณไร้สายเข้ากับอุปกรณ์จำลองสัญญาณไฟจราจร ให้สูงจากพื้นประมาณ 1 เมตร เพื่อให้สามารถรับสัญญาณได้ง่าย

3. เมื่อจ่ายไฟให้กับอุปกรณ์จำลองสัญญาณไฟจราจรแล้ว อุปกรณ์ทั้งหมดจะพร้อมทำงาน เนื่องจากอุปกรณ์ทุกชิ้นได้ใช้ไฟจากอุปกรณ์จำลองสัญญาณไฟจราจรเป็นไฟเลี้ยง

1.2 การใช้งาน

1. การใช้งานสำหรับผู้พิการทางสายตา จะให้ผู้พิการทางสายตาพกรีโมทตัวส่งสัญญาณไร้สายติดตัวไว้

2. ผู้พิการทางสายตาจะสามารถรู้ได้ว่ามาถึงบริเวณสี่แยกจากสัญลักษณ์ที่มีอยู่ที่พื้น

3. เมื่อผู้พิการต้องการข้ามถนน ผู้พิการเพียงกดรีโมท เมื่อตัวรับสัญญาณได้รับสัญญาณแล้วจะส่งลอจิกไปยังอุปกรณ์จำลองสัญญาณไฟจราจร เพื่อให้ส่งสถานะของสัญญาณไฟจราจรไปยังไอซีบันทึกเสียงอีกที จากนั้นไอซีบันทึกเสียง จะเล่นเสียงที่บอกสถานะของไฟจราจร ณ ขณะที่ผู้พิการทางสายตาได้กดรีโมท

4. หลังจากที่ผู้พิการได้ยินสถานะของไฟจราจรแล้ว ผู้พิการจะต้องตัดสินใจเองว่า จะข้ามถนนในเวลานั้นได้ทันหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

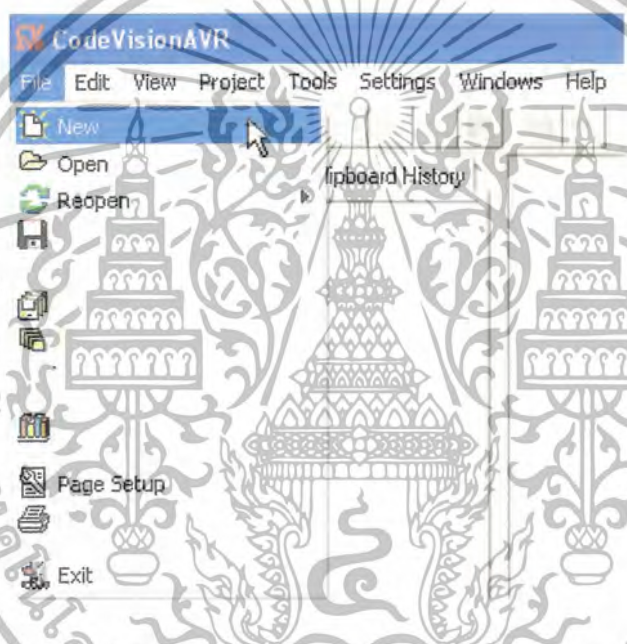
คู่มือการติดตั้งอุปกรณ์ช่วยเหลือผู้พิการทางสายตาข้ามถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

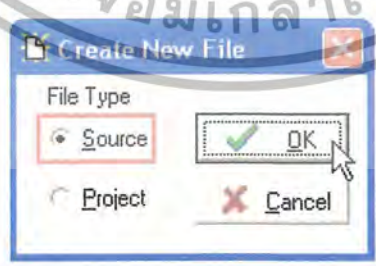
ตัวอย่างการพัฒนาโปรแกรมด้วย CodeVisionAVR

สำหรับโปรแกรมที่จะใช้ในการเขียนโปรแกรมภาษาซี ก็คือโปรแกรม CodeVisionAVR ซึ่งซอฟต์แวร์ตัวนี้ทางบริษัท HP InfoTech มีเวอร์ชั่นทดลองใช้ซึ่งผู้ทดลองสามารถเข้าไปดาวน์โหลดโปรแกรมได้ที่เว็บไซต์ www.hpinfotech.com แต่อย่างไรก็ตามทางทีมงานได้รวบรวมโปรแกรมนี้ไว้ในแผ่น CD-ROM แล้ว ซึ่งเป็นโปรแกรม CodeVisionAVR เวอร์ชั่น 1.24.7e โดยข้อจำกัดของเวอร์ชั่นทดลองคือสามารถคอมไพล์ Source code ได้ไม่เกิน 2kbytes ซึ่งขั้นตอนการใช้งานโปรแกรมมีดังนี้

1. เปิดโปรแกรม CodeVisionAVR C Compiler และคลิกเลือกที่เมนูคำสั่ง File → New ดังรูป



2. เลือก File Type เป็น Source เพื่อสร้างไฟล์ภาษาซีใหม่และคลิกปุ่ม OK ดังรูป



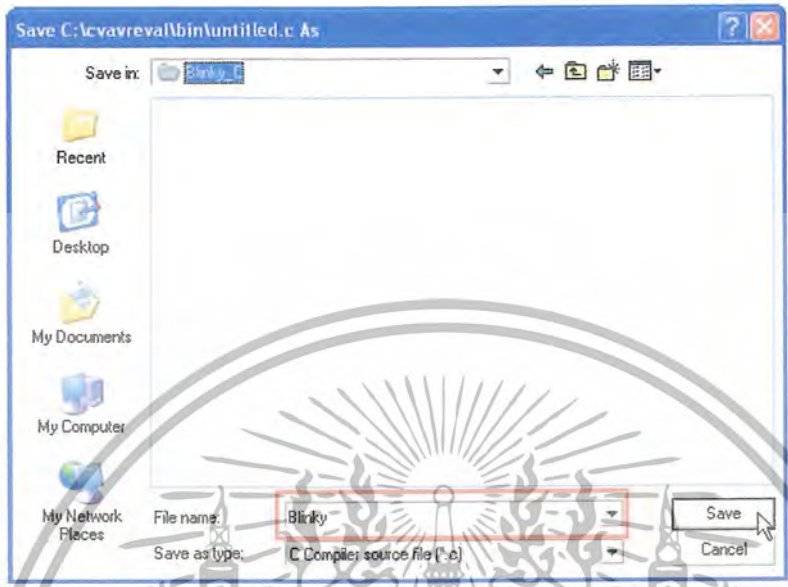
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นจะปรากฏหน้าต่าง Editor ให้ทำการเขียนโปรแกรมดังตัวอย่าง ซึ่งเป็นตัวอย่างโปรแกรมไฟกระพริบที่ PORTB.0

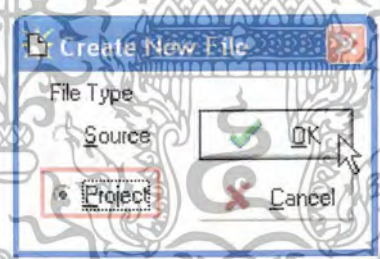
```
//*********************************************/;
//*Hardware    : ET-AVR STAMP (ATmega64)       */;
//*CPU         : ATMEL-ATmega64                */;
//*X-TAL       : 16.00 MHz                     */;
//*Filename    : Main.C                        */;
//*Complier    : CodeVisionAVR V1.24.7e        */;
//*Last Update : 9-02-2006 (ETT CO.,LTD)       */;
//*            : WWW.ETT.CO.TH                 */;
//*Description : Example LED Blink on Portb.0  */;
//*********************************************/;
//*CodeVisionAVR Complier Option Setting       */;
//*Chip type              : ATmega64           */;
//*Program type           : Application        */;
//*Clock frequency        : 16.000000MHz       */;
//*Memory model           : Small              */;
//*External SRAM size     : 0                  */;
//*Data Stack size        : 1024               */;
//*********************************************/;

#include <mega64.h>                 // ATmega64 MCU
#include <delay.h>                  // Delay functions

void main(void)
{
  PORTB=0x00;                       // PB7..0 = 0
  DDRB=0x01;                        // PB0 = Output

 //Loop Blink LED on PB0
  while (1)
  {
     PORTB |= 0x01;                 // PB0 = 1 (OFF LED)
     delay_ms(200);                 // Display LED Delay
     PORTB &= 0xFE;                 // PB0 = 0 (ON LED)
     delay_ms(200);                 // Display LED Delay
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

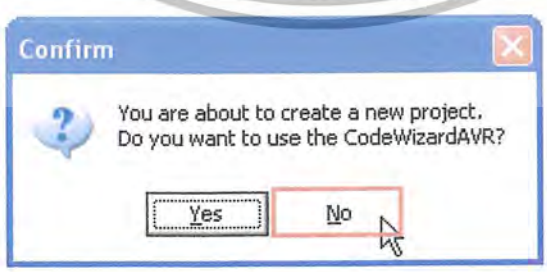
4. ทำการบันทึกโปรแกรมภาษาซีที่เขียนโดยเลือกเมนู File → Save ทำการตั้งชื่อไฟล์และกดปุ่ม Save ดังรูป



5. เลือกที่เมนู File → New และเลือก File Type เป็น Project เพื่อสร้างโปรเจกต์ใหม่และคลิกปุ่ม OK ดังรูป

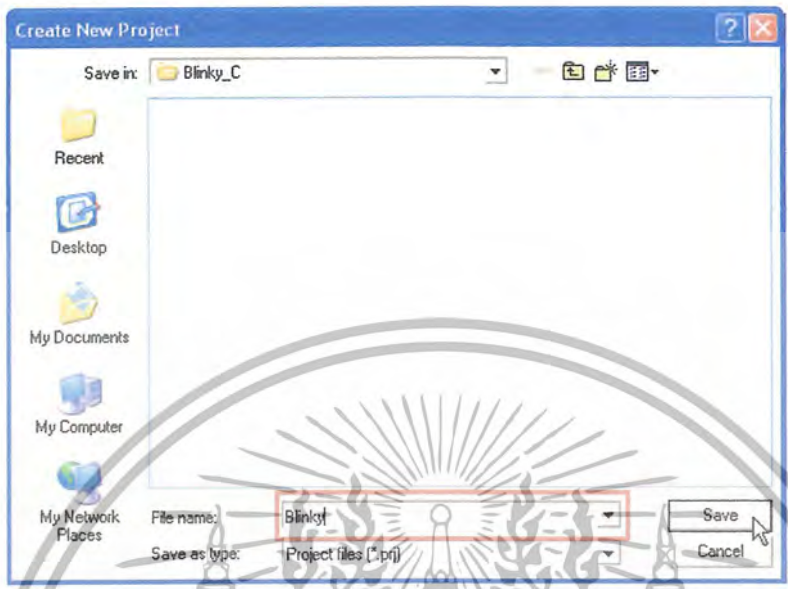


6. คลิกปุ่ม No เพื่อไม่ใช้ตัวช่วยในการสร้างโปรเจกต์ (CodeWizard)

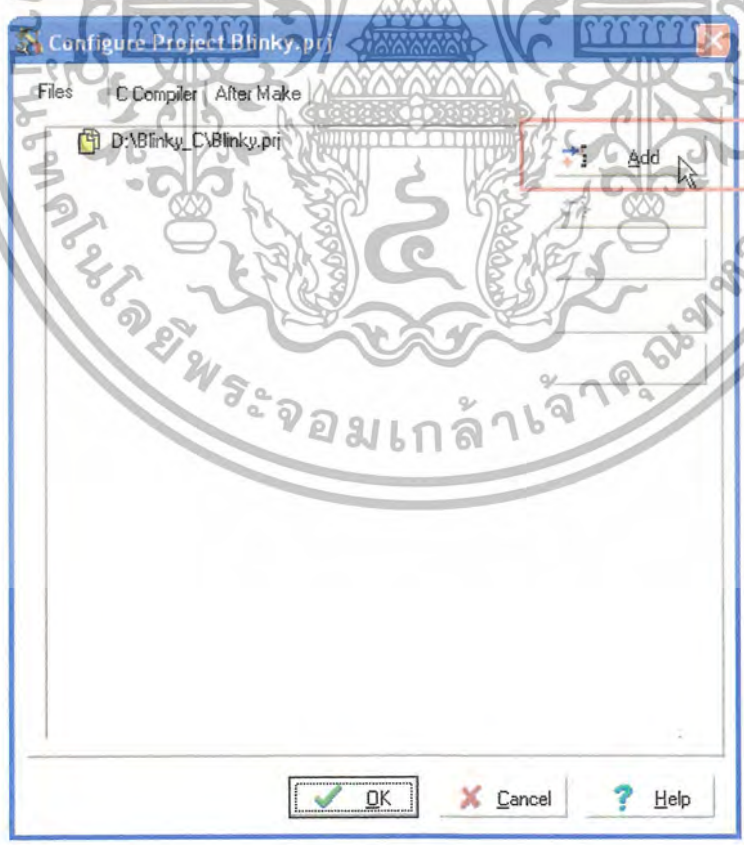


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

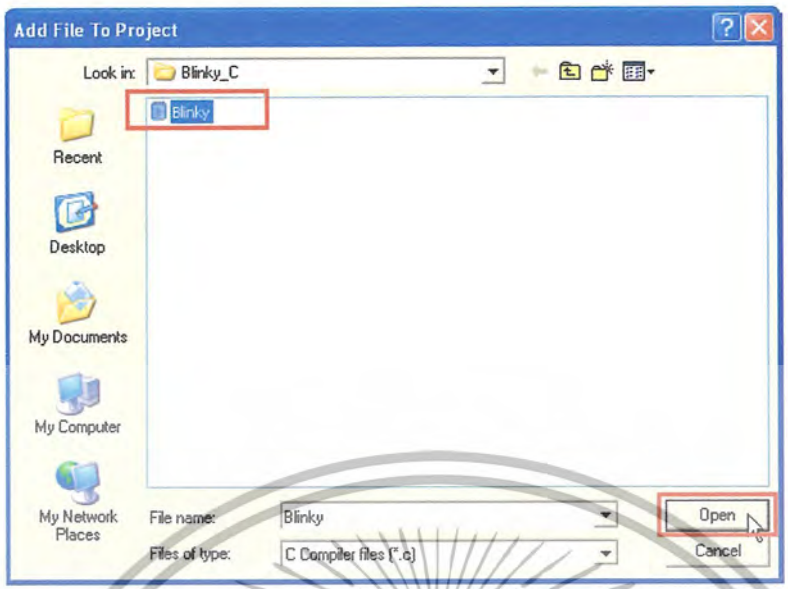
7. ทำการตั้งชื่อโปรเจกต์ตามต้องการและคลิกปุ่ม Save ดังรูป



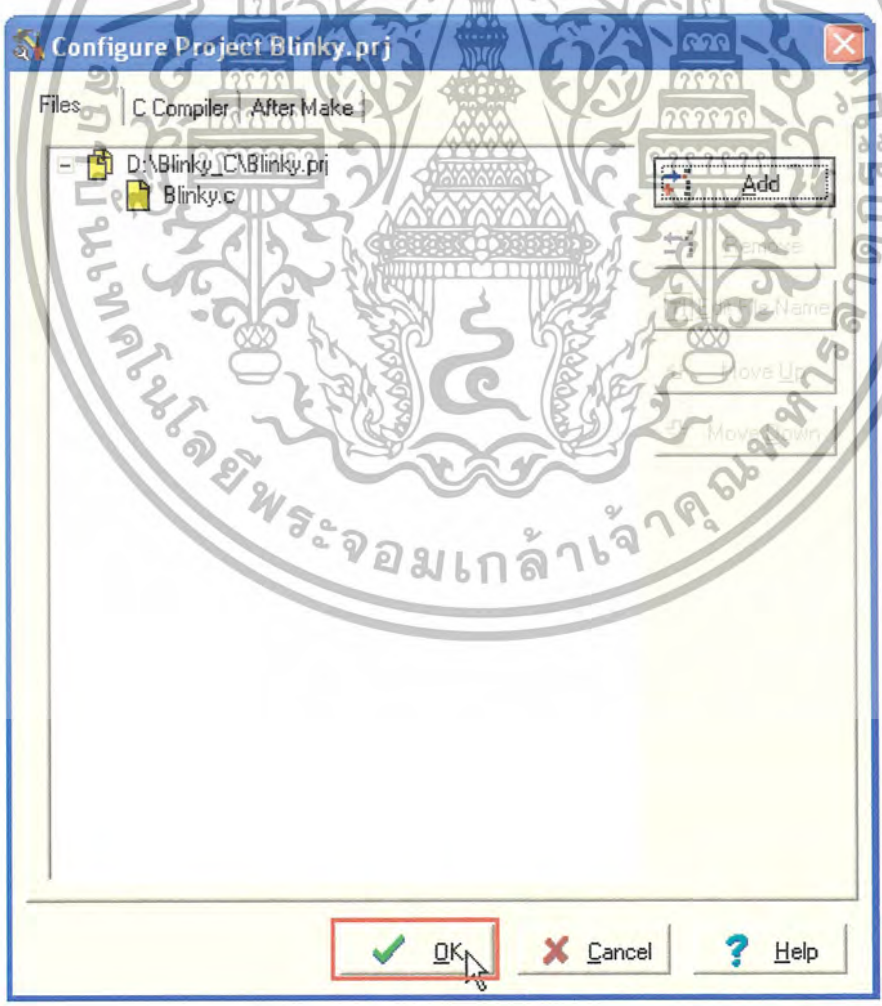
8. ทำการเพิ่มไฟล์ภาษาซีที่เขียนไว้ก่อนหน้านี้เข้ามาในโปรเจกต์โดยการคลิกปุ่ม Add ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

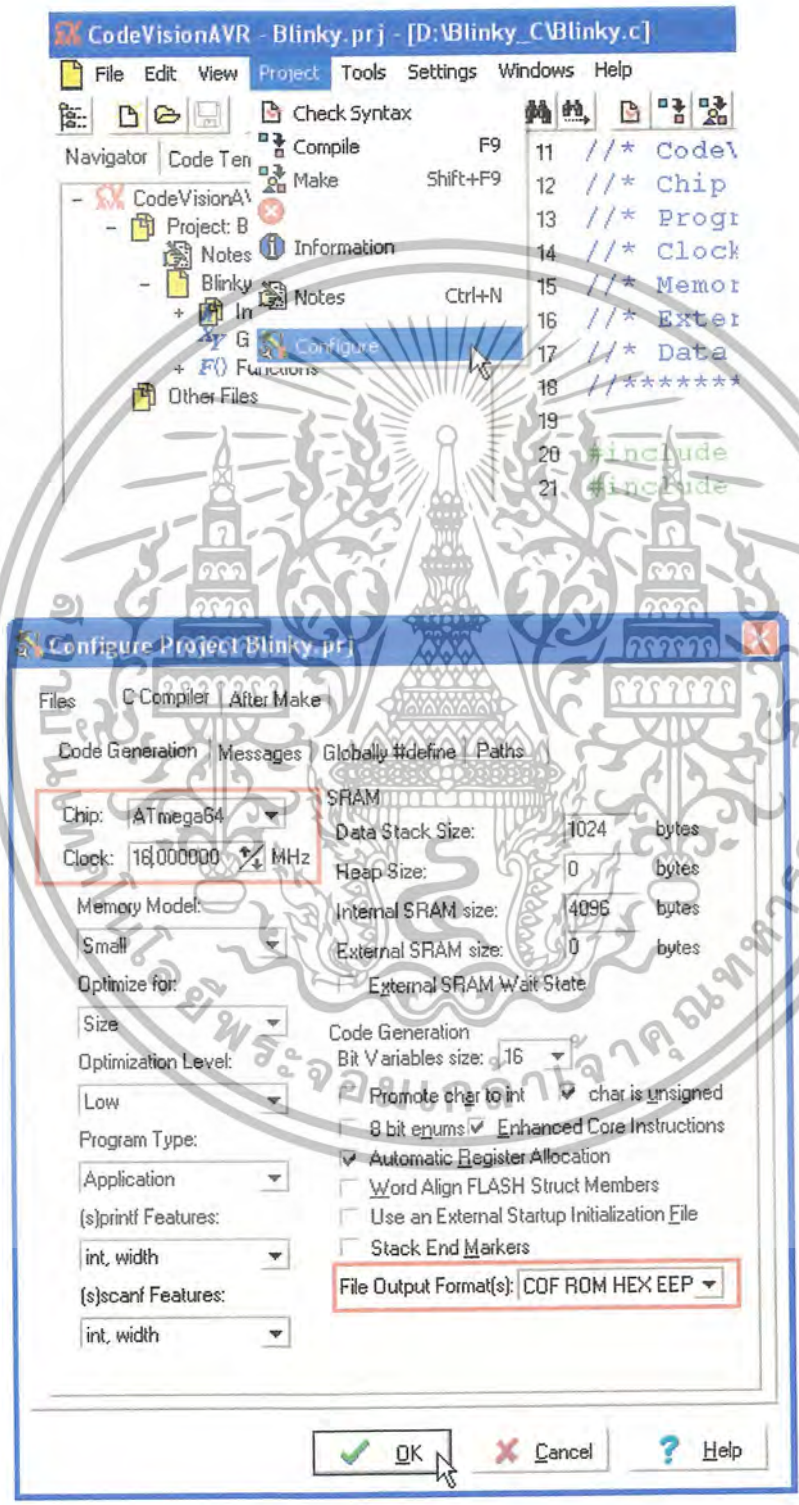


9. เมื่อทุกอย่างเรียบร้อยคลิกปุ่ม OK ดังรูป



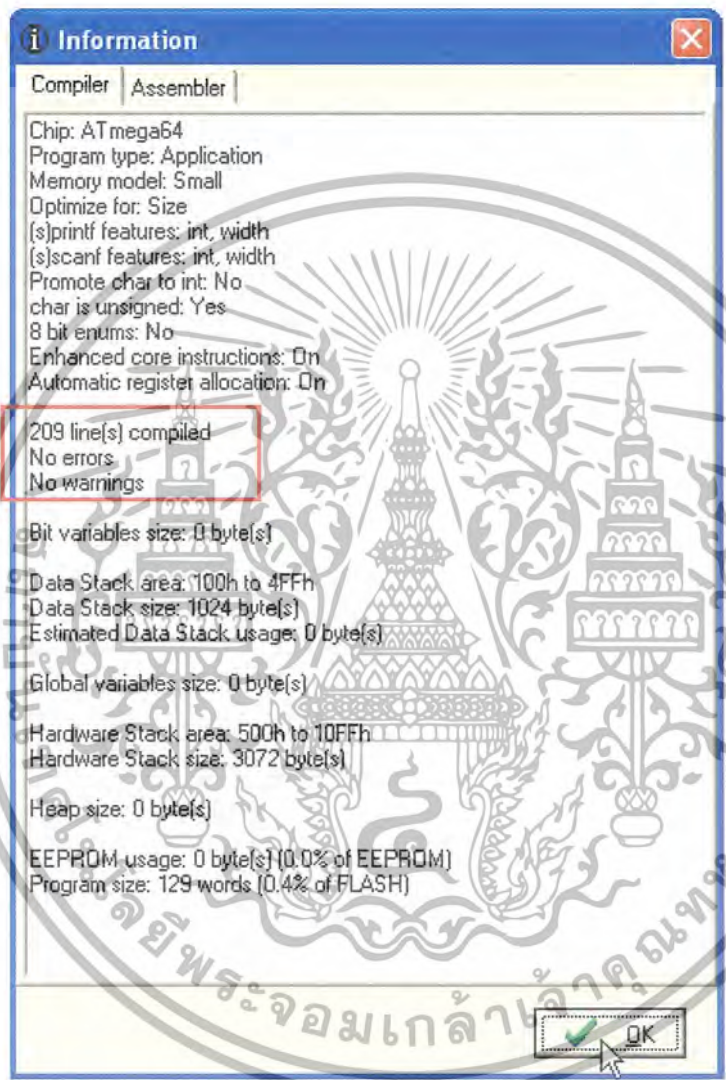
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ทำการตั้งค่าต่างของโปรเจกต์โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Configure จากนั้น ทำการกำหนดเบอร์ MCU เป็น ATmega64 ค่าคริสตอลเท่ากับ 16.000000 MHz และ File Output Format(s) เป็น COF ROM HEX EEP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

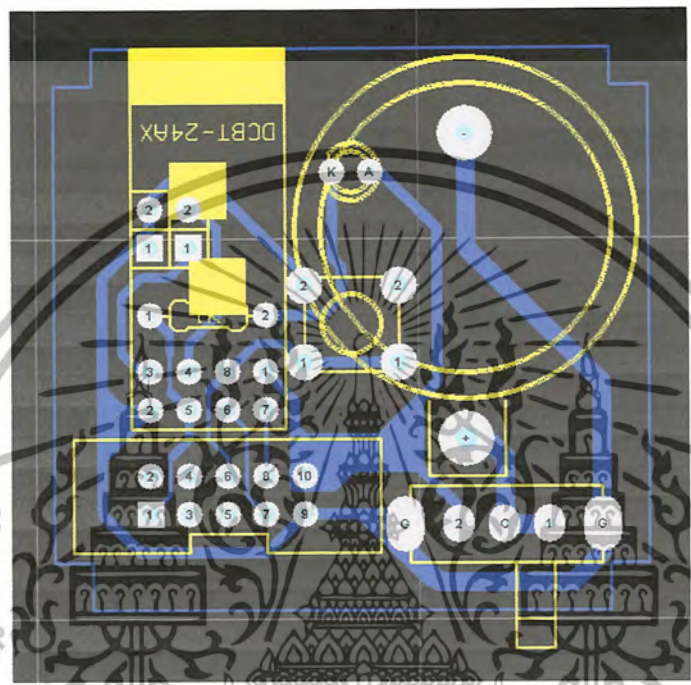
10. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้น โดยการคลิกเมาส์ที่เมนูคำสั่ง Project → Make ซึ่งหลังจากแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใด ๆ จะปรากฏข้อความ No errors, No warnings ต่อจากนี้ผู้ใช้ก็สามารถนำ Hex File ที่ได้จากสั่งแปลโปรแกรมนี้ไปทำการ Download ลง MCU ได้ทันที



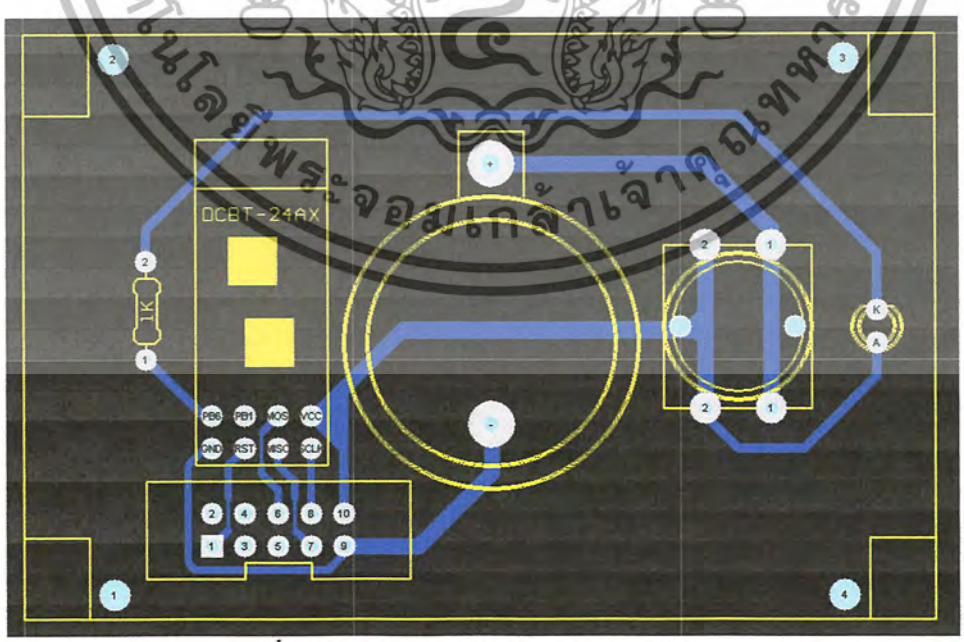
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

วงจรชิ้นงานและ PCB

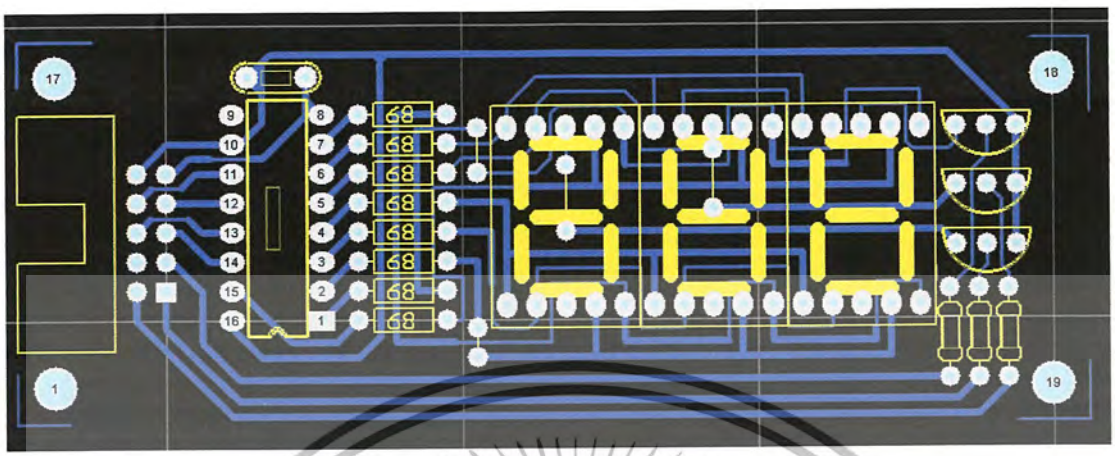


รูปที่ ค.1 ลายวงจรอุปกรณ์รับสัญญาณไร้สาย

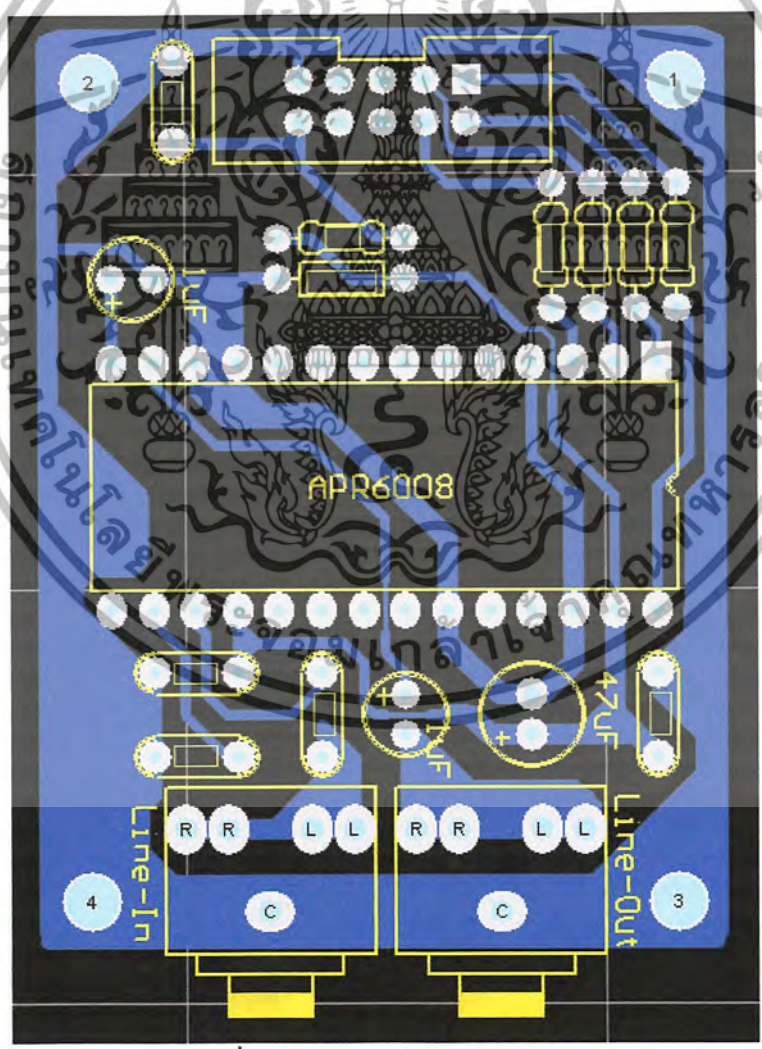


รูปที่ ค.2 ลายวงจรอุปกรณ์ส่งสัญญาณไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.3 ลายวงจรตัวแสดงเวลา 7 segment



รูปที่ ค.4 ลายวงจรไอซีบันทึกเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega128L
  - 4.5 - 5.5V for ATmega128
- Speed Grades
  - 0 - 8 MHz for ATmega128L
  - 0 - 16 MHz for ATmega128

8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

ATmega128
ATmega128L

Summary

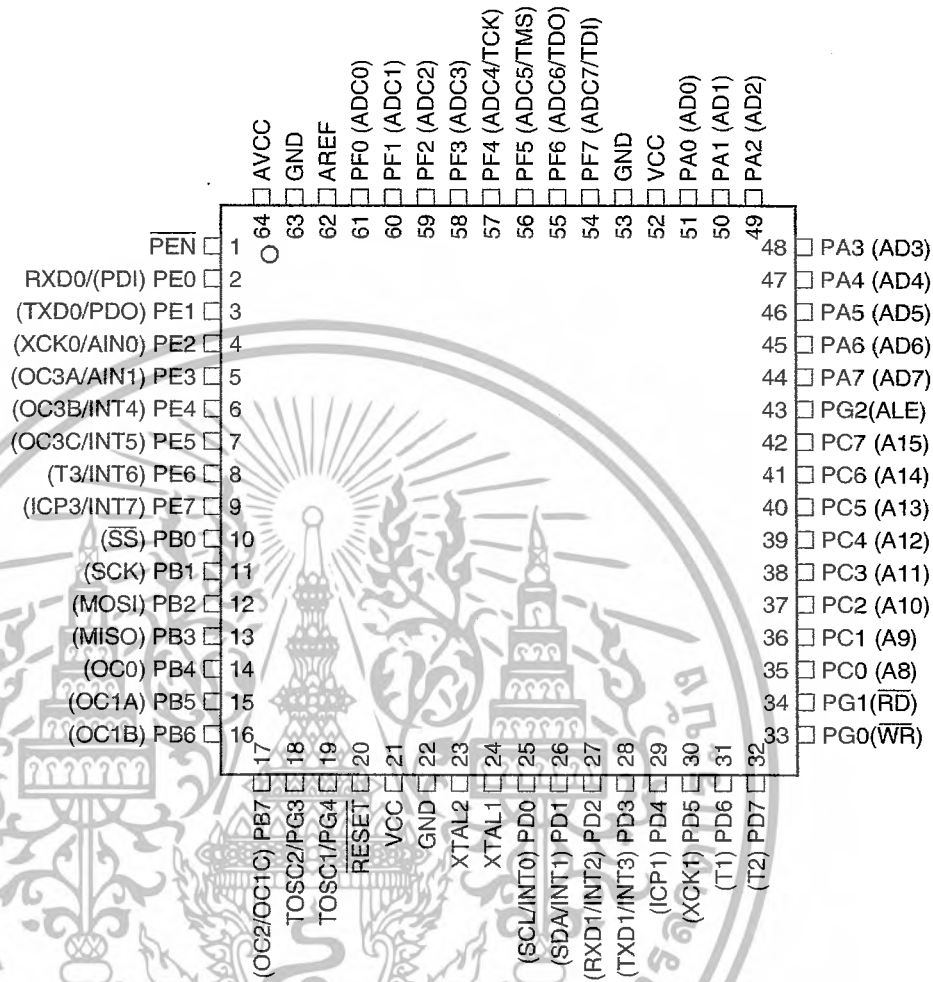
Rev. 2467MS–AVR–11/04

Note: This is a summary document. A complete document is available on our Web site at www.atmel.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

Figure 1. Pinout ATmega128



Note: The bottom pad under the MLF package should be soldered to ground.

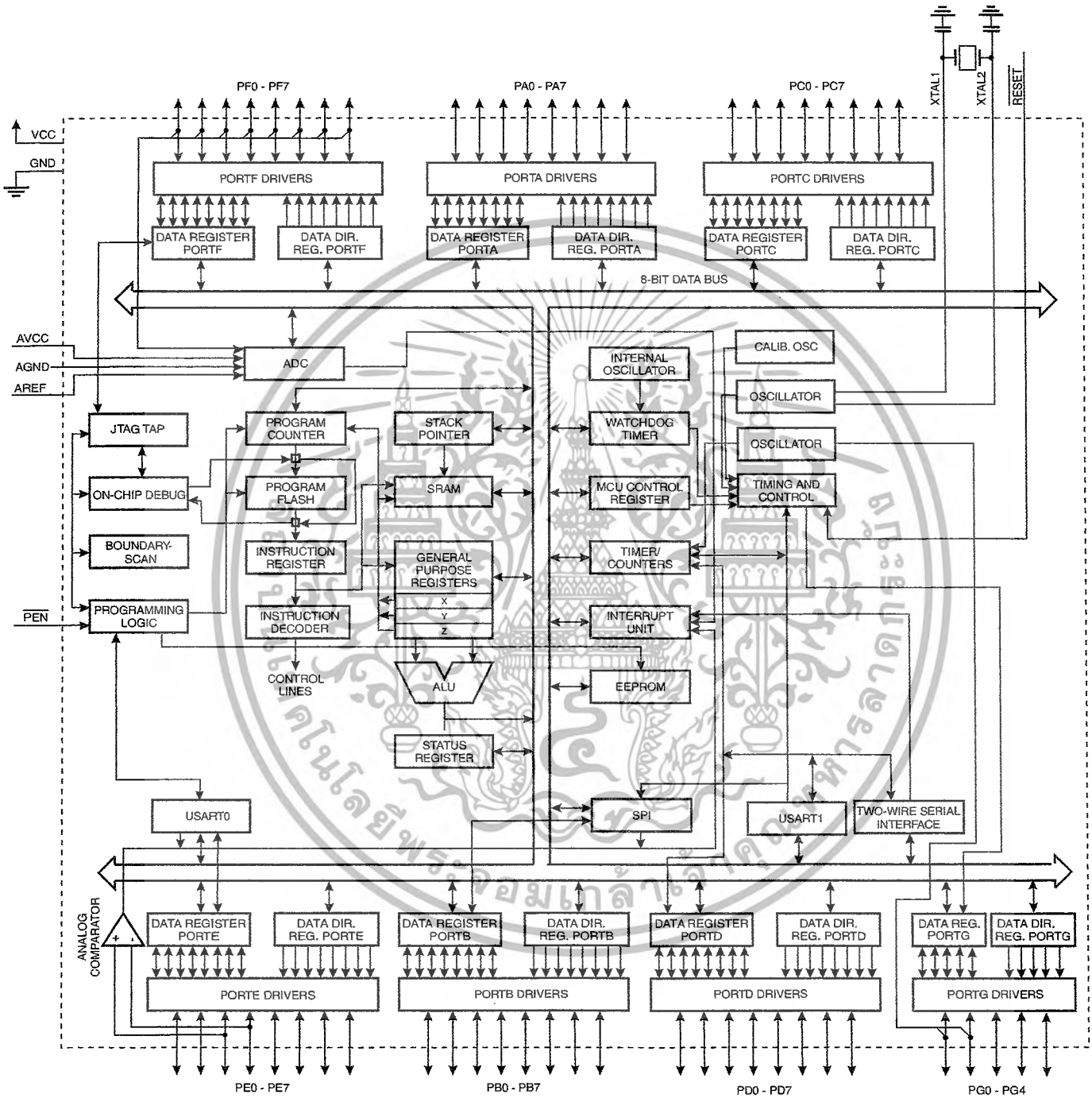
Overview

The ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram

Figure 2. Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega128 provides the following features: 128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 4K bytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

ATmega103 and ATmega128 Compatibility

The ATmega128 is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128. Most additional I/O locations are added in an Extended I/O space starting from $60 to $FF, (i.e., in the ATmega103 internal RAM space). These locations can be reached by using LD/LDS/LDD and ST/STS/STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the Extended Interrupt vectors are removed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

ATmega103 Compatibility Mode

By programming the M103C fuse, the ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

In addition, there are some other minor differences to make it more compatible to ATmega103:

- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega128 as listed on page 70.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega128 as listed on page 71.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega128 as listed on page 74. In ATmega103 compatibility mode, Port C is output only, and the port C pins are **not** tri-stated when a reset condition becomes active.

Note: The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega128 as listed on page 75.

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega128 as listed on page 78.

Port F (PF7..PF0)

Port F serves as the analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

Port G (PG4..PG0)

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32 kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

$\overline{\text{RESET}}$ Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 48. Shorter pulses are not guaranteed to generate a reset.

XTAL1 Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 Output from the inverting Oscillator amplifier.

AVCC AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF AREF is the analog reference pin for the A/D Converter.

PEN PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high . By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. $\overline{\text{PEN}}$ has no function during normal operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($FF)	Reserved	–	–	–	–	–	–	–	–	
..	Reserved	–	–	–	–	–	–	–	–	
($9E)	Reserved	–	–	–	–	–	–	–	–	
($9D)	UCSR1C	–	UMSEL1	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	191
($9C)	UDR1	USART1 I/O Data Register								189
($9B)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	189
($9A)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	190
($99)	UBRR1L	USART1 Baud Rate Register Low								193
($98)	UBRR1H	–	–	–	–	USART1 Baud Rate Register High				193
($97)	Reserved	–	–	–	–	–	–	–	–	
($96)	Reserved	–	–	–	–	–	–	–	–	
($95)	UCSR0C	–	UMSEL0	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	191
($94)	Reserved	–	–	–	–	–	–	–	–	
($93)	Reserved	–	–	–	–	–	–	–	–	
($92)	Reserved	–	–	–	–	–	–	–	–	
($91)	Reserved	–	–	–	–	–	–	–	–	
($90)	UBRR0H	–	–	–	–	USART0 Baud Rate Register High				193
($8F)	Reserved	–	–	–	–	–	–	–	–	
($8E)	Reserved	–	–	–	–	–	–	–	–	
($8D)	Reserved	–	–	–	–	–	–	–	–	
($8C)	TCCR3C	FOC3A	FOC3B	FOC3C	–	–	–	–	–	135
($8B)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	131
($8A)	TCCR3B	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	134
($89)	TCNT3H	Timer/Counter3 – Counter Register High Byte								136
($88)	TCNT3L	Timer/Counter3 – Counter Register Low Byte								136
($87)	OCR3AH	Timer/Counter3 – Output Compare Register A High Byte								136
($86)	OCR3AL	Timer/Counter3 – Output Compare Register A Low Byte								136
($85)	OCR3BH	Timer/Counter3 – Output Compare Register B High Byte								137
($84)	OCR3BL	Timer/Counter3 – Output Compare Register B Low Byte								137
($83)	OCR3CH	Timer/Counter3 – Output Compare Register C High Byte								137
($82)	OCR3CL	Timer/Counter3 – Output Compare Register C Low Byte								137
($81)	ICR3H	Timer/Counter3 – Input Capture Register High Byte								137
($80)	ICR3L	Timer/Counter3 – Input Capture Register Low Byte								137
($7F)	Reserved	–	–	–	–	–	–	–	–	
($7E)	Reserved	–	–	–	–	–	–	–	–	
($7D)	ETIMSK	–	–	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	138
($7C)	ETIFR	–	–	ICF3	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	139
($7B)	Reserved	–	–	–	–	–	–	–	–	
($7A)	TCCR1C	FOC1A	FOC1B	FOC1C	–	–	–	–	–	135
($79)	OCR1CH	Timer/Counter1 – Output Compare Register C High Byte								136
($78)	OCR1CL	Timer/Counter1 – Output Compare Register C Low Byte								136
($77)	Reserved	–	–	–	–	–	–	–	–	
($76)	Reserved	–	–	–	–	–	–	–	–	
($75)	Reserved	–	–	–	–	–	–	–	–	
($74)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	206
($73)	TWDR	Two-wire Serial Interface Data Register								208
($72)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	208
($71)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	207
($70)	TWBR	Two-wire Serial Interface Bit Rate Register								206
($6F)	OSCCAL	Oscillator Calibration Register								39
($6E)	Reserved	–	–	–	–	–	–	–	–	
($6D)	XMCRA	–	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11		29
($6C)	XMCRB	XMBK	–	–	–	–	XMM2	XMM1	XMM0	31
($6B)	Reserved	–	–	–	–	–	–	–	–	
($6A)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	87
($69)	Reserved	–	–	–	–	–	–	–	–	
($68)	SPMCSR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	279
($67)	Reserved	–	–	–	–	–	–	–	–	
($66)	Reserved	–	–	–	–	–	–	–	–	
($65)	PORTG	–	–	–	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	86
($64)	DDRG	–	–	–	DDG4	DDG3	DDG2	DDG1	DDG0	86
($63)	PING	–	–	–	PING4	PING3	PING2	PING1	PING0	86
($62)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($61)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	86
($60)	Reserved	–	–	–	–	–	–	–	–	
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	9
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	12
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
$3C ($5C)	XDIV	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	41
$3B ($5B)	RAMPZ	–	–	–	–	–	–	–	RAMPZ0	12
$3A ($5A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	88
$39 ($59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	89
$38 ($58)	EIFR	INTF7	INTF6	INTF5	INTF4	INTF3	INTF	INTF1	INTF0	89
$37 ($57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	106, 138, 158
$36 ($56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	106, 139, 158
$35 ($55)	MCUCR	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	29, 42, 61
$34 ($54)	MCUCSR	JTD	–	–	JTRF	WDRF	BORF	EXTRF	PORF	51, 256
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	101
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								103
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								103
$30 ($50)	ASSR	–	–	–	–	AS0	TCN0UB	OCR0UB	TCR0UB	104
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	131
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	134
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								136
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								136
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								136
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								136
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								136
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								136
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								137
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								137
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	156
$24 ($44)	TCNT2	Timer/Counter2 (8 Bit)								158
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								158
$22 ($42)	OCDR	IDRD/OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	253
$21 ($41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	53
$20 ($40)	SFIOR	TSM	–	–	–	ACME	PUD	PSR0	PSR321	70, 107, 143, 228
$1F ($3F)	EEARH	–	–	–	–	EEPROM Address Register High				19
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								19
$1D ($3D)	EEDR	EEPROM Data Register								20
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	20
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	84
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	84
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	84
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	84
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	84
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	84
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	84
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	84
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	85
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	85
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	85
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	85
$0F ($2F)	SPDR	SPI Data Register								168
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	168
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	166
$0C ($2C)	UDR0	USART0 I/O Data Register								189
$0B ($2B)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	189
$0A ($2A)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	190
$09 ($29)	UBRR0L	USART0 Baud Rate Register Low								193
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	228
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	244
$06 ($26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	245
$05 ($25)	ADCH	ADC Data Register High Byte								246
$04 ($24)	ADCL	ADC Data Register Low byte								246
$03 ($23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	85
$02 ($22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	85
$00 ($20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	86

Notes: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

2. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if ($REG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Inc	Rd ← (RAMPZ:Z), RAMPZ:Z ← RAMPZ:Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package[1]	Operation Range
8	2.7 - 5.5V	ATmega128L-8AC ATmega128L-8MC	64A 64M1	Commercial (0°C to 70°C)
		ATmega128L-8AI ATmega128L-8AU[2] ATmega128L-8MI ATmega128L-8MU[2]	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	4.5 - 5.5V	ATmega128-16AC ATmega128-16MC	64A 64M1	Commercial (0°C to 70°C)
		ATmega128-16AI ATmega128-16AU[2] ATmega128-16MI ATmega128-16MU[2]	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)

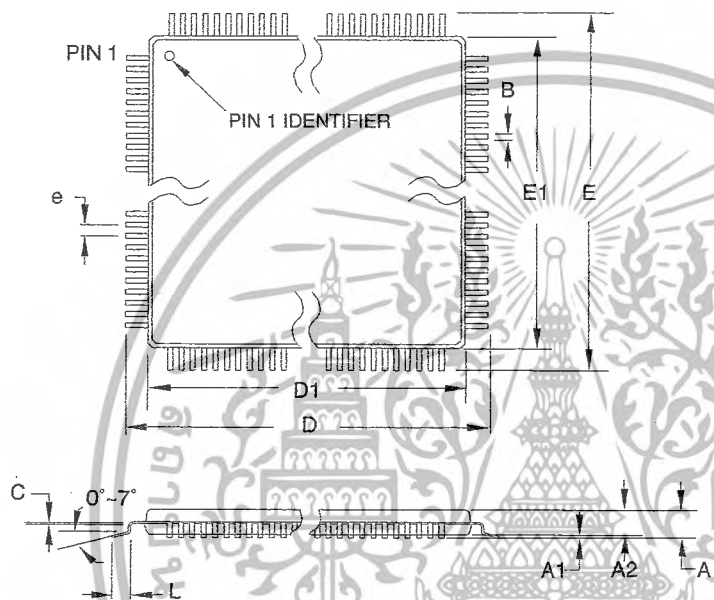
Notes: 1. The device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

Package Type	
64A	64-lead, 14 x 14 x 1.0 mm, Thin Profile Plastic Quad Flat Package (TQFP)
64M1	64-pad, 9 x 9 x 1.0 mm, Micro Lead Frame Package (MLF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

64A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

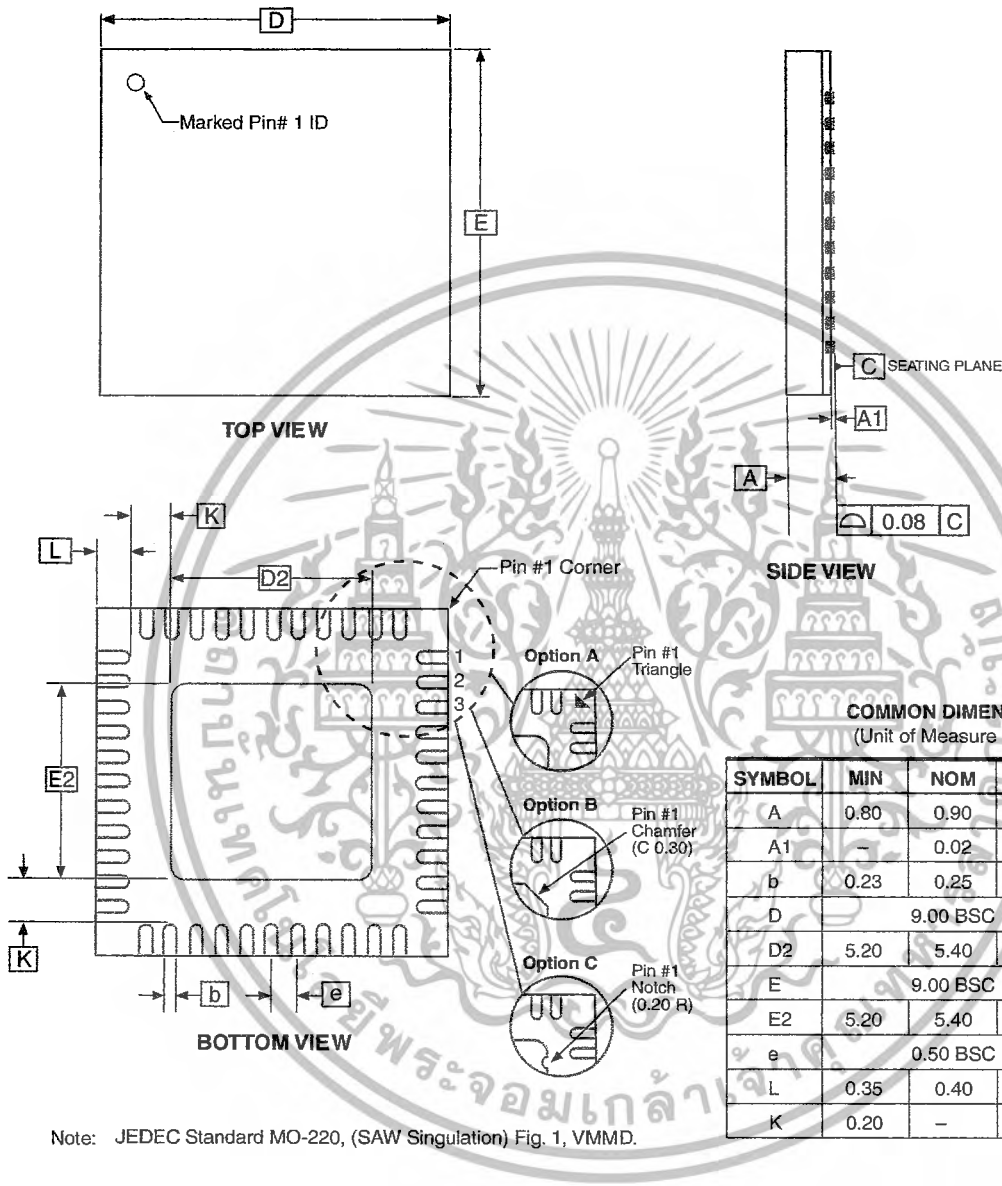
Notes:
1. This package conforms to JEDEC reference MS-026, Variation AEB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**64A,** 64-lead, 14 x 14 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	64A	B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

64M1



Note: JEDEC Standard MO-220, (SAW Singulation) Fig. 1, VMMD.

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
b	0.23	0.25	0.28	
D		9.00 BSC		
D2	5.20	5.40	5.60	
E		9.00 BSC		
E2	5.20	5.40	5.60	
e		0.50 BSC		
L	0.35	0.40	0.45	
K	0.20	–	–	

8/19/04

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**64M1**, 64-pad, 9 x 9 x 1.0 mm Body, Lead Pitch 0.50 mm, 5.40 mm Exposed Pad, Micro Lead Frame Package (MLF)	64M1	D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Errata

The revision letter in this section refers to the revision of the ATmega128 device.

ATmega128 Rev. I

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. Stabilizing time needed when changing XDIV Register

After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

1.Clear the I bit in the SREG Register.

2.Set the new pre-scaling factor in XDIV register.

3.Execute 8 NOP instructions

4.Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

Assembly Code Example:

```
CLI                 ; clear global interrupt enable
OUT  XDIV, temp     ; set new prescale value
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
SEI                 ; clear global interrupt enable
```

2. Stabilizing time needed when changing OSCCAL Register

After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The behavior follows errata number 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **Stabilizing time needed when changing OSCCAL Register**

   After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

   Problem Fix / Workaround

   The behavior follows errata number 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller) to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. G

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. **Stabilizing time needed when changing XDIV Register**

   After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

   Problem Fix / Workaround

   The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

   1.Clear the I bit in the SREG Register.

   2.Set the new pre-scaling factor in XDIV register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller) to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. H

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. Stabilizing time needed when changing XDIV Register

After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

1.Clear the I bit in the SREG Register.

2.Set the new pre-scaling factor in XDIV register.

3.Execute 8 NOP instructions

4.Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

Assembly Code Example:

```
CLI                 ; clear global interrupt enable
OUT  XDIV, temp     ; set new prescale value
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
SEI                 ; clear global interrupt enable
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.Execute 8 NOP instructions

4.Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

Assembly Code Example:

```
CLI                 ; clear global interrupt enable
OUT  XDIV, temp     ; set new prescale value
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
SEI                 ; clear global interrupt enable
```

2. **Stabilizing time needed when changing OSCCAL Register**

After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The behavior follows errata number 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller) to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. F

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. Stabilizing time needed when changing XDIV Register

After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

1.Clear the I bit in the SREG Register.

2.Set the new pre-scaling factor in XDIV register.

3.Execute 8 NOP instructions

4.Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

Assembly Code Example:

```
CLI                 ; clear global interrupt enable
OUT  XDIV, temp     ; set new prescale value
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
NOP                 ; no operation
SEI                 ; clear global interrupt enable
```

2. Stabilizing time needed when changing OSCCAL Register

After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

Problem Fix / Workaround

The behavior follows errata number 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the pre-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller) to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

Changes from Rev. 2467L-05/04 to Rev. 2467M-11/04

1. Removed "analog ground", replaced by "ground".
2. Updated Table 11 on page 38, Table 114 on page 287, Table 128 on page 306, and Table 132 on page 323. Updated Figure 114 on page 239.
3. Added note to "Port C (PC7..PC0)" on page 6.
4. Updated "Ordering Information" on page 14.

Changes from Rev. 2467K-03/04 to Rev. 2467L-05/04

1. Removed "Preliminary" and "TBD" from the datasheet, replaced occurrences of ICx with ICPx.
2. Updated Table 8 on page 36, Table 19 on page 48, Table 22 on page 54, Table 96 on page 243, Table 126 on page 302, Table 128 on page 306, Table 132 on page 323, and Table 134 on page 325.
3. Updated "External Memory Interface" on page 24.
4. Updated "Device Identification Register" on page 255.
5. Updated "Electrical Characteristics" on page 321.
6. Updated "ADC Characteristics" on page 327.
7. Updated "ATmega128 Typical Characteristics" on page 335.
8. Updated "Ordering Information" on page 14.

Changes from Rev. 2467J-12/03 to Rev. 2467K-03/04

1. Updated "Errata" on page 17.

Changes from Rev. 2467I-09/03 to Rev. 2467J-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 39.

Changes from Rev. 2467H-02/03 to Rev. 2467I-09/03

1. Updated note in "XTAL Divide Control Register – XDIV" on page 41.
2. Updated "JTAG Interface and On-chip Debug System" on page 46.
3. Updated values for V_{BOT} (BODLEVEL = 1) in Table 19 on page 48.
4. Updated "Test Access Port – TAP" on page 248 regarding JTAGEN.
5. Updated description for the JTD bit on page 257.
6. Added a note regarding JTAGEN fuse to Table 118 on page 290.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Updated R_{PU} values in "DC Characteristics" on page 321.
8. Added a proposal for solving problems regarding the JTAG instruction IDCODE in "Errata" on page 17.

Changes from Rev. 2467G-09/02 to Rev. 2467H-02/03

1. Corrected the names of the two Prescaler bits in the SFIOR Register.
2. Added Chip Erase as a first step under "Programming the Flash" on page 318 and "Programming the EEPROM" on page 319.
3. Removed reference to the "Multipurpose Oscillator" application note and the "32 kHz Crystal Oscillator" application note, which do not exist.
4. Corrected OCn waveforms in Figure 52 on page 123.
5. Various minor Timer1 corrections.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Various minor TWI corrections.
8. Added reference to Table 124 on page 293 from both SPI Serial Programming and Self Programming to inform about the Flash Page size.
9. Added note under "Filling the Temporary Buffer (Page Loading)" on page 282 about writing to the EEPROM during an SPM Page load.
10. Removed ADHSM completely.
11. Added section "EEPROM Write During Power-down Sleep Mode" on page 23.
12. Updated drawings in "Packaging Information" on page 15.

Changes from Rev. 2467F-09/02 to Rev. 2467G-09/02

1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

Changes from Rev. 2467E-04/02 to Rev. 2467F-09/02

1. Added 64-pad MLF Package and updated "Ordering Information" on page 14.
2. Added the section "Using all Locations of External Memory Smaller than 64 KB" on page 31.
3. Added the section "Default Clock Source" on page 35.
4. Renamed SPMCR to SPMCSR in entire document.
5. When using external clock there are some limitations regards to change of frequency. This is descried in "External Clock" on page 40 and Table 131, "External Clock Drive," on page 323.
6. Added a sub section regarding OCD-system and power consumption in the section "Minimizing Power Consumption" on page 45.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. **Corrected typo (WGM-bit setting) for:**
   "Fast PWM Mode" on page 96 (Timer/Counter0).
   "Phase Correct PWM Mode" on page 98 (Timer/Counter0).
   "Fast PWM Mode" on page 150 (Timer/Counter2).
   "Phase Correct PWM Mode" on page 152 (Timer/Counter2).

8. **Corrected Table 81 on page 192 (USART).**

9. **Corrected Table 102 on page 261 (Boundary-Scan)**

10. **Updated Vil parameter in "DC Characteristics" on page 321.**

Changes from Rev. 2467D-03/02 to Rev. 2467E-04/02

1. **Updated the Characterization Data in Section "ATmega128 Typical Characteristics" on page 335.**

2. **Updated the following tables:**
   Table 19 on page 48, Table 20 on page 52, Table 68 on page 157, Table 102 on page 261, and Table 136 on page 328.

3. **Updated Description of OSCCAL Calibration Byte.**
   In the data sheet, it was not explained how to take advantage of the calibration bytes for 2, 4, and 8 MHz Oscillator selections. This is now added in the following sections:
   Improved description of "Oscillator Calibration Register – OSCCAL" on page 39 and "Calibration Byte" on page 291.

Changes from Rev. 2467C-02/02 to Rev. 2467D-03/02

1. **Added more information about "ATmega103 Compatibility Mode" on page 5.**

2. **Updated Table 2, "EEPROM Programming Time," on page 21.**

3. **Updated typical Start-up Time in Table 7 on page 35, Table 9 and Table 10 on page 37, Table 12 on page 38, Table 14 on page 39, and Table 16 on page 40.**

4. **Updated Table 22 on page 54 with typical WDT Time-out.**

5. **Corrected description of ADSC bit in "ADC Control and Status Register A – ADCSRA" on page 245.**

6. **Improved description on how to do a polarity check of the ADC differential results in "ADC Conversion Result" on page 242.**

7. **Corrected JTAG version numbers in "JTAG Version Numbers" on page 256.**

8. **Improved description of addressing during SPM (usage of RAMPZ) on "Addressing the Flash During Self-Programming" on page 280, "Performing Page Erase by SPM" on page 282, and "Performing a Page Write" on page 282.**

9. **Added not regarding OCDEN Fuse below Table 118 on page 290.**

10. **Updated Programming Figures:**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 135 on page 292 and Figure 144 on page 304 are updated to also reflect that AVCC must be connected during Programming mode. Figure 139 on page 299 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the PROG_PAGELOAD and PROG_PAGEREAD instructions on page 310.**

12. **Added Calibrated RC Oscillator characterization curves in section "ATmega128 Typical Characteristics" on page 335.**

13. **Updated "Two-wire Serial Interface" section.**

    More details regarding use of the TWI Power-down operation and using the TWI as master with low TWBRR values are added into the data sheet. Added the note at the end of the "Bit Rate Generator Unit" on page 204. Added the description at the end of "Address Match Unit" on page 205.

14. **Added a note regarding usage of Timer/Counter0 combined with the clock. See "XTAL Divide Control Register – XDIV" on page 41.**

Changes from Rev. 2467B-09/01 to Rev. 2467C-02/02

1. **Corrected Description of Alternate Functions of Port G**

   Corrected description of TOSC1 and TOSC2 in "Alternate Functions of Port G" on page 82.

2. **Added JTAG Version Numbers for rev. F and rev. G**

   Updated Table 100 on page 256.

3 **Added Some Preliminary Test Limits and Characterization Data**

   Removed some of the TBD's in the following tables and pages:

   Table 19 on page 48, Table 20 on page 52, "DC Characteristics" on page 321, Table 131 on page 323, Table 134 on page 325, and Table 136 on page 328.

4. **Corrected "Ordering Information" on page 14.**

5. **Added some Characterization Data in Section "ATmega128 Typical Characteristics" on page 335.**

6. **Removed Alternative Algortihm for Leaving JTAG Programming Mode.**

   See "Leaving Programming Mode" on page 318.

7. **Added Description on How to Access the Extended Fuse Byte Through JTAG Programming Mode.**

   See "Programming the Fuses" on page 320 and "Reading the Fuses and Lock Bits" on page 320.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

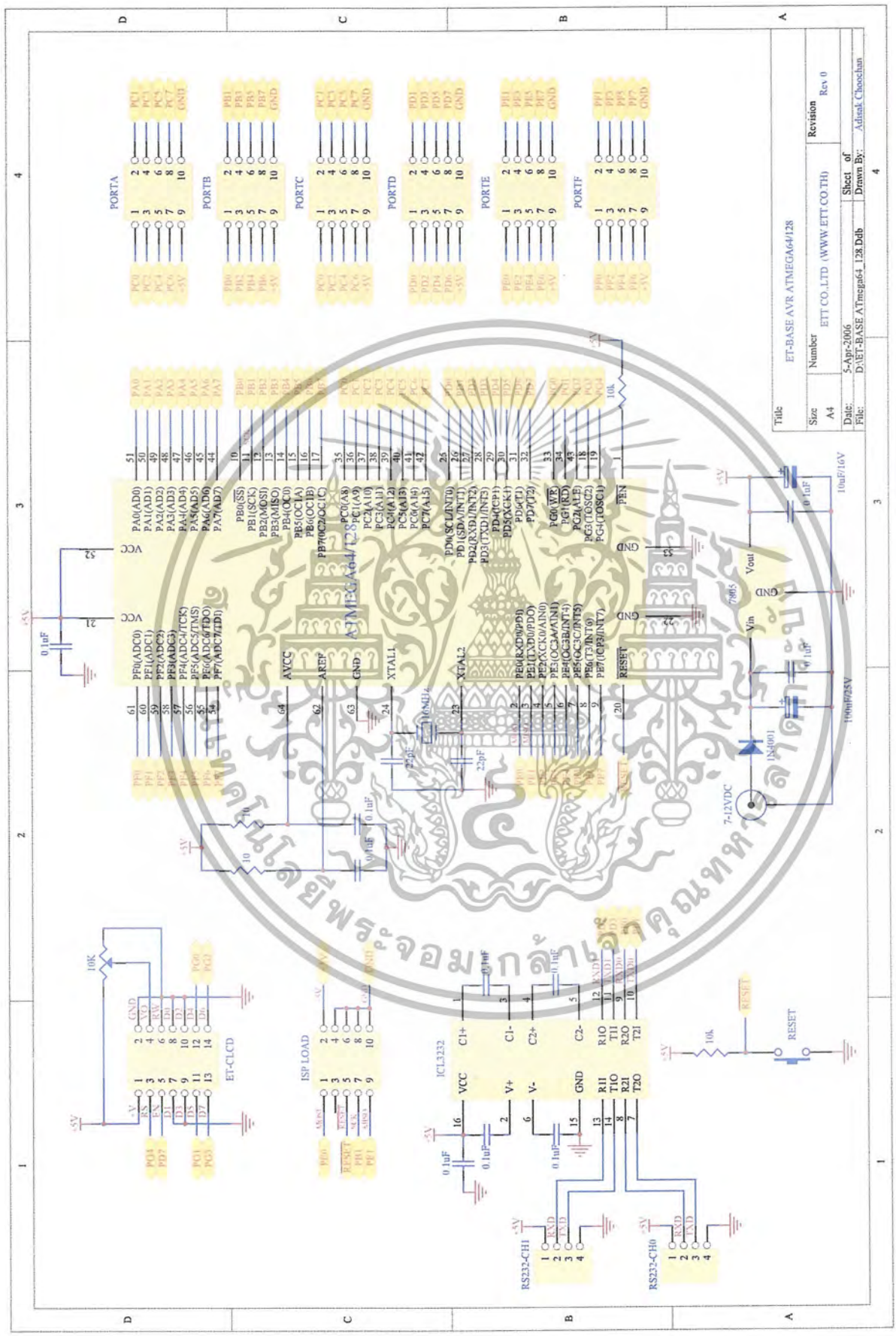
Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2004. All rights reserved. Atmel®, logo and combinations thereof, AVR®, and AVR Studio® are registered trademarks, and Everywhere You Are℠ are the trademarks of Atmel Corporation or its subsidiaries. Microsoft®, Windows®, Windows NT®, and Windows XP® are the registered trademarks of Microsoft Corporation. Other terms and product names may be trademarks of others.

Printed on recycled paper.

2467MS–AVR–11/04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้