นาฬิกาบอกเวลาเป็นเสียงพูดควบคุมด้วยรีโมทไร้สายสำหรับผู้พิการทางสายตา

(Speaker Clock Controlled by Wireless Remote for Blind and Visually People)

โดย

นายประสิทธิ์ บุญเอนก B4506362

นางสาวเจือจิต จับกลาง B4509240

รายงานนี้เป็นส่วนหนึ่งของรายวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

ภาคการศึกษาที่ 3 ปีการศึกษา 2548

สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงงาน	นาฬิกาบอกเวลาเป็นเสียงพูดควบคุมด้วยรีโมทไร้สายสำหรับผู้พิการทางสายตา
ผู้ดำเนินงาน	นายประสิทธิ์ บุญเอนก B4506362
	นางสาวเจือจิต จับกลาง B4509240
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ชุติมา พรหมมาก
สาขาวิชา	วิศวกรรมโทรคมนาคม
ภาคการศึกษา	2/2548

บทคัดย่อ

ปัจจุบันความเจริญก้าวหน้าทางวิทยาศาสตร์ได้เข้ามามีบทบาทต่อการดำเนินชีวิตของมนุษย์อย่างต่อเนื่องและเครื่องอำนวยความสะดวกต่างๆก็ได้สร้างสรรค์ขึ้นมาเพื่อตอบสนองความต้องการในหลายๆด้าน แต่ในอีกด้านท่ามกลางความเจริญก็ยังมีบุคคลจำนวนไม่น้อยที่ต้องการความช่วยเหลืออยู่เช่นกันซึ่งก็คือผู้พิการนั่นเอง สำหรับโครงงานนี้มีชื่อว่า นาฬิกาบอกเวลาเป็นเสียงพูด สร้างขึ้นมาโดยการนำความรู้ทางด้านไมโครคอนโทรลเลอร์ตระกูล MCS-51 อุปกรณ์อิเล็กทรอนิกส์อย่างไอซีบันทึกและเล่นเสียง การเขียนโปรแกรมควบคุมการทำงานด้วยภาษาแอสเซมบลี และความรู้เกี่ยวกับคลื่นความถี่วิทยุซึ่งถูกนำมาใช้ในส่วนของรีโมทควบคุมโดยจุดประสงค์ในการสร้างนั้นเพื่ออำนวยความสะดวกและสนองความต้องการของผู้พิการในทุกเวลา

กิตติกรรมประกาศ

การจัดทำโครงงานนาฬิกาบอกเวลาเป็นเสียงพูดควบคุมด้วยรีโมทไร้สายสำหรับผู้พิการทางสายตา สำเร็จลุล่วงได้ด้วยความช่วยเหลือและความกรุณาของบุคคลหลายท่าน คณะผู้จัดทำใคร่ขอแสดงความขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน

บุคคลเหล่านั้นประกอบไปด้วย

- ผู้ที่ทำให้มีสาขาวิชาวิศวกรรมโทรคมนาคม
- อาจารย์ ดร. ชุติมา พรหมมาก อาจารย์ที่ปรึกษาโครงงานผู้เปิดโอกาสให้คณะผู้จัดทำได้เรียนรู้และชี้แนะสิ่งที่เป็นประโยชน์ตลอดการดำเนินงาน
- คณาจารย์ทุกท่านที่ประสิทธิ์ประสาทความรู้ตลอดช่วงชีวิตที่ผ่านมาจนทำให้มีโอกาสในการทำโครงงานนี้
- เพื่อนๆ ทุกคนสำหรับทุกความช่วยเหลือและกำลังใจ

และที่ขาดมิได้ ขอกราบขอบพระคุณบุพการีอันเป็นที่รัก ผู้ซึ่งมอบสิ่งที่ดีที่สุดโดยไม่เคยหยุดหย่อน

ประสิทธิ์ บุญเอนก
เจือจิต จับกลาง

สารบัญ

เรื่อง	หน้า
- บทคัดย่อ	ก
- กิตติกรรมประกาศ	ข
- สารบัญ	ค
- สารบัญภาพ	ง
- สารบัญตาราง	จ
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบข่ายโครงงาน	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎี	**3**
2.1 ไมโครคอนโทรลเลอร์ MCS-51	3
2.2 ไอซีสร้างฐานเวลาจริง (Real Time Clock)	3
2.3 ไอซีอัดเสียงและเล่นเสียง (ISD2590)	5
2.4 แอลอีดี 7 ส่วน แบบหลายหลัก	6
2.5 วงจรภาครับ-ส่งสัญญาณวิทยุ	8
บทที่ 3 การดำเนินงานและการทดสอบ	**11**
บทที่ 4 สรุปผลการทดลองและข้อเสนอแนะ	**20**
4.1 สิ่งที่ได้จากโครงงาน	20
4.2 ปัญหาและอุปสรรค	20
4.3 ข้อจำกัดของโครงงาน	20
4.4 ข้อเสนอแนะ	20
บรรณานุกรม	**21**
ภาคผนวก	**22**
ประวัติผู้เขียน	**60**

สารบัญรูปภาพ

หน้า

รูปที่1.1 แสดงแผนภาพรวมของระบบ 1
รูปที่2.1 แสดงรายละเอียดของไอซีสร้างฐานเวลาจริง (DS1302) 4
รูปที่2.2 แสดงสัญญาณขณะที่ไมโครคอนโทรลเลอร์ทำการอ่านข้อมูลจากไอซีสร้างฐานเวลาจริง 4
รูปที่2.3 แสดงสัญญาณขณะที่ไมโครคอนโทรลเลอร์ทำการเขียนข้อมูลลงไอซีสร้างฐานเวลาจริง 5
รูปที่2.4 แสดงสัญญาณที่ขาต่างๆขณะทำการเล่นเสียงในรูปแบบ M6 6
รูปที่2.5 แสดงวงจรการใช้งานในลักษณะการแสดงผลแบบมัลติเพล็กการใช้หลักการแสดงผลแบบมัลติเพล็ก(Multiplexed display) 7
รูปที่2.6 แสดงการใช้ทรานซิสเตอร์แบบ PNP กำหนดสถานะ การทำงานที่ขาคอมมอนแต่ละแบบของแอลอีดี 7 ส่วน 8
รูปที่2.7 แสดงโครงสร้างภายนอกและตำแหน่งขาของภาคส่งสัญญาณวิทยุ 8
รูปที่2.8 วงจรภาคส่งสัญญาณวิทยุ 9
รูปที่2.9 แสดงโครงสร้างภายนอกและตำแหน่งขาของภาครับสัญญาณวิทยุ 9
รูปที่2.10 วงจรภาครับสัญญาณวิทยุ 10
รูปที่3.1 แสดงขั้นตอนการอัดเสียง 12
รูปที่3.2 แสดงขั้นตอนการสร้างตัววงจร 13
รูปที่3.3 แสดงไฟล์ Schematic ที่เสร็จเรียบร้อยแล้ว 14
รูปที่3.4 แสดงวงจรที่ลงอุปกรณ์เสร็จเรียบร้อยแล้ว 15
รูปที่3.5 Flow Chart แสดงการทำงานของนาฬิกา 16
รูปที่3.6 Flow Chart แสดงการทำงานของรีโมท 17
รูปที่3.7 ภายในของรีโมท RF 18
รูปที่3.8 วงจรขยายสัญญาณเสียง 18
รูปที่3.9 แสดงการทดสอบใช้งานจริง 19

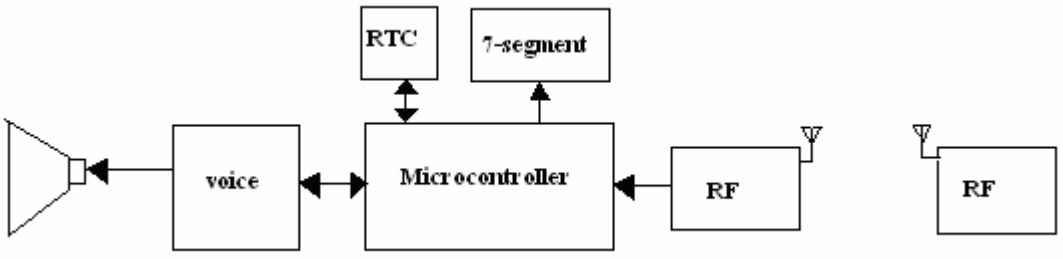
สารบัญตาราง

		หน้า
ตารางที่2.1 แสดงการทำงานของไอซีอัดเสียงและเล่นเสียง (ISD2590)ในรูปแบบต่างๆ รูปแบบ	6	5

บทที่ 1
บทนำ

1.1 ความเป็นมา

ผู้พิการทางสายตาคือผู้ที่ควรได้รับการดูแลเอาใจใส่มากกว่าคนปกติทั่วไป ผู้จัดทำจึงอยากมีส่วนร่วมในการช่วยเหลือผู้พิการทางสายตา โดยได้จัดทำนาฬิกาที่สามารถบอกเวลาเป็นเสียงพูดควบคุมด้วยรีโมทไร้สาย เพื่ออำนวยความสะดวกให้กับผู้พิการทางสายตาเนื่องจากนาฬิกาที่มีจำหน่ายในท้องตลาดนั้นส่วนใหญ่ผู้ผลิตไม่ได้ให้ความสำคัญกับผู้พิการทางสายตา ด้วยเหตุนี้คณะผู้จัดทำจึงได้คิดค้นเครื่องมือที่จะช่วยเหลือผู้พิการทางสายตาโดยเฉพาะ โดยอาศัยความรู้ที่ได้รับจากการศึกษาในสาขาวิชาวิศวกรรมโทรคมนาคมผนวกกับความสนใจด้านไมโครคอนโทลเลอร์เป็นพิเศษ ดังนั้นโครงงานนี้จึงได้จัดทำขึ้นเพื่อประดิษฐ์นาฬิกาที่สามารถบอกเวลาเป็นเสียงพูดได้ ซึ่งอุปกรณ์ทั้งหมดประกอบด้วย 4. ส่วน ดังนี้ 1.ไมโครคอนโทรลเลอร์ MCS-51 ทำหน้าที่ควบคุมการทำงานของส่วนประกอบอื่นๆ 2. ไอซีสร้างฐานเวลาจริง (Real Time Clock) ทำหน้าที่บอกเวลาที่เที่ยงตรงให้กับไมโครคอนโทรลเลอร์ 3.ไอซีบันทึกและเล่นเสียง และ 4.ส่วนรับ-ส่งสัญญาณวิทยุ โดยการทำงานเริ่มจากผู้ใช้กดปุ่มเพื่อส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ และจากนั้นไมโครคอนโทรลเลอร์ก็จะทำการอ่านค่าเวลาจากไอซีสร้างฐานเวลาจริง (Real Time Clock) แล้วสั่งงานให้ไอซีบันทึกเสียงซึ่งได้ทำการบันทึกเสียงไว้ก่อนหน้านี้ทำการแสดงเวลาในรูปแบบเสียง ณ ปัจจุบัน



รูปที่1.1 แสดงแผนภาพรวมของระบบ

1.2 วัตถุประสงค์

1. เพื่อสร้างสิ่งประดิษฐ์ที่สามารถช่วยเหลือผู้พิการทางสายตา
2. เพื่อศึกษาโปรแกรมภาษา Assembly ในการควบคุมไมโครคอนโทรลเลอร์ที่เชื่อมโยงระหว่างส่วนประกอบอื่นๆ
3. เพื่อศึกษาสถาปัตยกรรมไมโครโพรเซสเซอร์ ส่วนประกอบของระบบ หน่วยความจำ

การรับข้อมูลเข้าออก การเชื่อมต่อ และการทำโปรแกรม

4. เพื่อศึกษาการทำงานของไอซีสร้างฐานเวลาจริง ไอซีบันทึกและเล่นเสียง

5.เพื่อศึกษาการทำงานของการรับ-ส่งสัญญาณวิทยุ

1.3 ขอบข่ายโครงงาน

1. ศึกษาการทำงานของไอซีสร้างฐานเวลาจริง ไอซีบันทึกและเล่นเสียง
2. ศึกษาโปรแกรมภาษา Assembly
3. ออกแบบโปรแกรมในการแสดงผลเวลาที่ได้รับจากไอซีสร้างฐานเวลาจริงไปแสดงผลที่ 7 – Segment
4. ศึกษาการทำงานของไอซีบันทึกและเล่นเสียง
5. ศึกษาการทำงานของการรับ-ส่งสัญญาณวิทยุ
6. ออกแบบวงจรรวมของส่วนประกอบต่างๆให้สามารถใช้งานร่วมกันได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้สร้างสิ่งประดิษฐ์ที่สามารถช่วยเหลือผู้พิการทางสายตา
2. ได้ศึกษาโปรแกรมภาษา Assembly ในการควบคุมไมโครคอนโทรลเลอร์ที่เชื่อมโยงระหว่างส่วนประกอบอื่นๆ
3. ได้ศึกษาสถาปัตยกรรมไมโครโพรเซสเซอร์ ส่วนประกอบของระบบ หน่วยความจำ การรับข้อมูลเข้าออก การเชื่อมต่อ และการทำโปรแกรม
4. ได้ศึกษาการทำงานของไอซีสร้างฐานเวลาจริง ไอซีบันทึกและเล่นเสียง
5. ได้ศึกษาการทำงานของการรับ-ส่งสัญญาณวิทยุ

บทที่ 2
ทฤษฎี

2.1 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาท์พุต หน่วยความจำ และวงจรการกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ซึ่งในโครงงานนี้เราก็ได้เลือกไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เบอร์ AT89S8252 มาใช้ควบคุมวงจรในส่วนของนาฬิกาและไมโครคอนโทรลเลอร์เบอร์ AT89C2051 มาใช้ในส่วนของวงจรรีโมท ซึ่งเหตุผลในการเลือกใช้ไมโครคอนโทรลเลอร์ทั้ง 2 เบอร์นี้เนื่องจากได้เคยผ่านการศึกษาในรายวิชาไมโครคอนโทรลเลอร์ และใช้โปรแกรมภาษา Assembly มาใช้ในส่วนของโปรแกรมควบคุม

ในการนำไมโครคอนโทรลเลอร์มาใช้งานนั้นมีส่วนประกอบสำคัญที่จะทำให้ไมโครคอนโทรลเลอร์ทำงานได้ตามวัตถุประสงค์ที่ต้องการมีดังนี้

1.วงจรภาคจ่ายไฟโดยมีไอซีเรคกูเลเตอร์เบอร์ 7805 ในการปรับระดับแรงดันให้เอาท์พุตมีค่าเท่ากับ 5 โวลต์

นอกจากวงจรภาคจ่ายไฟหลักแล้วยังมีวงจรภาคจ่ายไฟสำรองที่ใช้แบตเตอรี่ขนาด 3 โวลต์จ่ายไฟเลี้ยงให้กับวงจรไอซีสร้างฐานเวลาจริง (Real Time Clock)

2.วงจรที่ทำให้ไมโครคอนโทรลเลอร์พร้อมใช้งานซึ่งประกอบด้วยวงจรรีเซต วงจรกำเนิดความถี่ (Crystal 11.0592 MHz) โดยขา 40, 31 เป็นแรงดัน 5 โวลต์ และขาที่ 20 เป็นกราวน์

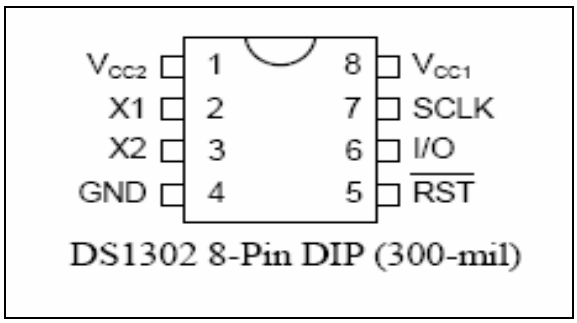
2.2 ไอซีสร้างฐานเวลาจริง (Real Time Clock)

ไอซีสร้างฐานเวลาจริง (DS1302) มีหน้าที่สร้างฐานเวลาจริงให้กับระบบไมโครคอนโทรลเลอร์ โดยจะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาในส่วนของวินาที นาที ชั่วโมง วันที่ วัน เดือน และปี โดยสามารถปรับให้ตรงตามปฏิทินได้อย่างถูกต้อง

คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

1. ให้ข้อมูลตั้งแต่วินาทีจนถึงปี และให้ข้อมูลได้อย่างเที่ยงตรงจนถึง คศ.2099
2. มีหน่วยความจำแรมอยู่ภายในสามารถใช้เก็บข้อมูลทั่วไปได้

3. มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ สามารถรักษาข้อมูลเวลาได้แม้ไม่มีไฟเลี้ยง

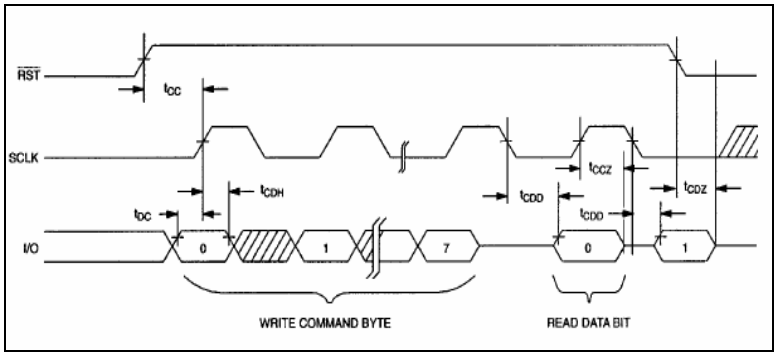


รูปที่2.1 แสดงรายละเอียดของไอซีสร้างฐานเวลาจริง DS1302

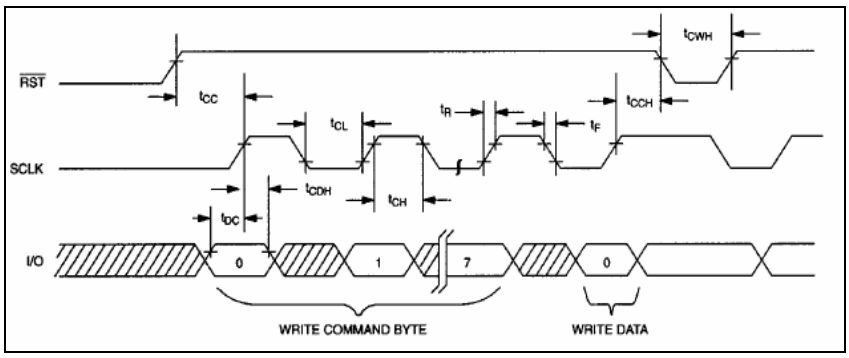
สำหรับการสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับไอซีสร้างฐานเวลาจริงนั้นเป็นการสื่อสารโดยใช้สายสัญญาณ 3 เส้น ดังนี้

1. SCK ใช้สำหรับรับสัญญาณคล็อก
2. I/O ใช้สำหรับรับส่งข้อมูล
3. RST ใช้สำหรับสร้างสัญญาณรีเซต

โดยในการสื่อสารมีการส่งสัญญาณระหว่างไมโครคอนโทรลเลอร์กับไอซีสร้างฐานเวลาจริงแสดงได้ดังรูป



รูปที่2.2 แสดงสัญญาณขณะที่ไมโครคอนโทรลเลอร์ทำการอ่านข้อมูลจากไอซีสร้างฐานเวลาจริง



รูปที่2.3 แสดงสัญญาณขณะที่ไมโครคอนโทรลเลอร์ทำการเขียนข้อมูลลงไอซีสร้างฐานเวลาจริง

2.3 ไอซีอัดเสียงและเล่นเสียง ISD2590

ในส่วนของการเล่นเสียง จะเป็นการนำเอาเสียงของตัวเลขแต่ละตัวมาต่อเรียงกันให้ตรงตามเวลาที่อ่านได้จากไอซีสร้างฐานเวลาจริง ในการเลือกไอซีที่จะนำมาใช้นั้นสิ่งสำคัญที่จะต้องคำนึงถึงคือ รูปแบบการอัดเสียงและเล่นเสียง จำนวนช่องสัญญาณ และ ปริมาณหน่วยความจำหรือเวลาที่สามารถอัดเสียงได้ สำหรับโครงงานนี้เราได้เลือกใช้ไอซีเบอร์ ISD 2590 ที่สามารถอัดเสียงได้นานถึง 90 วินาที และยังอัดเสียงได้โดยไม่จำกัดจำนวนช่องสัญญาณ รูปแบบการอัดเสียงและเล่นเสียงมีทั้งหมด 6 รูปแบบโดยแสดงได้ดังตารางด้านล่างนี้

Mode Control	Function	Typical Use	Jointly Compatible[1]
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete $\overline{\text{EOM}}$ markers	Position $\overline{\text{EOM}}$ marker at the end of the last message	M3, M4, M5, M6
M2	Not applicable	Reserved	N/A
M3	Looping	Continuous playback from Address 0	M1, M5, M6
M4	Consecutive addressing	Record/play multiple consecutive messages	M0, M1, M5
M5	$\overline{\text{CE}}$ level-activated	Allows message pausing	M0, M1, M3, M4
M6	Push-button control	Simplified device interface	M0, M1, M3

ตารางที่2.1 แสดงการทำงานของไอซีอัดเสียงและเล่นเสียง ISD2590ในรูปแบบต่างๆ 6 รูปแบบ

สำหรับโครงงานนี้เราได้ใช้รูปแบบในการอัดเสียงและเล่นเสียงอยู่ 2 รูปแบบด้วยกันคือ M0 (Message cueing) ใช้ในการกระโดดข้ามช่องสัญญาณไปช่องสัญญาณที่ต้องการก่อนเล่นเสียง และ M6 (Push-Button Control) ใช้ในการอัดเสียงและเล่นเสียงตามลำดับช่องสัญญาณ

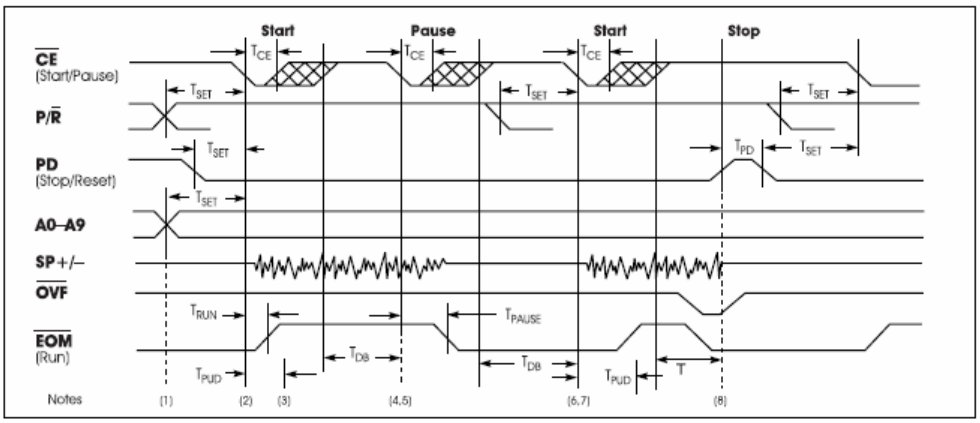
การควบคุมการทำงานสามารถทำได้โดยการต่อวงตามรูปจะเห็นว่ามีขาที่ใช้ในการควบคุมการทำงานอยู่ทั้งหมด 4 ขาด้วยกันคือ

1. Fast ใช้ในการเลือกรูปแบบการทำงานว่าจะทำงานในรูปแบบ M0 หรือ M6

2. CE ในรูปแบบ M0 ถ้ามีสัญญาณพัลส์ มาที่ขานี้จะทำการกระโดดข้ามไป 1 ช่องสัญญาณ และ ในรูปแบบ M6 ถ้ามีสัญญาณพัลส์มาที่ขานี้จะทำการเริ่มการเล่นเสียงและหยุดการเล่นเสียงโดยตำแหน่งช่องสัญญาณก็จะถูกเปลี่ยนไปตามลำดับ

3. PD ใช้เริ่มการทำงานของไอซี

4. EOM จะเป็นสัญญาณเตือนเมื่อทำการเล่นเสียงครบ 1 ช่องสัญญาณ

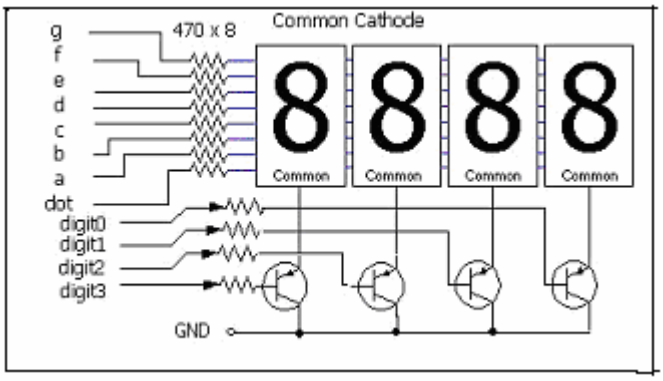


รูปที่2.4 แสดงสัญญาณที่ขาต่างๆขณะทำการเล่นเสียงในรูปแบบ M6

2.4 แอลอีดี 7 ส่วน แบบหลายหลัก

ในการใช้งานไมโครคอนโทรลเลอร์ เพื่อแสดงผลที่แอลอีดี 7ส่วนให้เป็นเอาต์พุต และถ้าหากเรานำแอลอีดี 7 ส่วนมาต่อใช้งานร่วมกันโดยตรงหลายๆหลัก ในกรณีที่แอลอีดีติดสว่างทุกเซกเมนต์ สว่างพร้อมกันทุกตัว จะต้องใช้ปริมาณของกระแสไฟที่จ่ายให้แอลอีดีมากขึ้น ดังนั้นในการใช้งานแอลอีดี 7 ส่วนถ้าหากต้องการแสดงผลหลายๆตัว พร้อมกัน จึงนิยมใช้วิธีการแสดงผลแบบมัลติเพล็ก(Multiplexed display) โดยเราจะใช้วิธีการต่อขาของแอลอีดี 7 ส่วนในแต่ละ

เซกเมนต์ขนานกับเซกเมนต์เดียวกันของแอลอีดี 7 ส่วนตัวอื่นๆ ทุกตัว พิจารณาวงจรจากรูปประกอบ



รูปที่2.5 แสดงวงจรการใช้งานในลักษณะการแสดงผลแบบมัลติเพล็กการใช้หลักการแสดงผลแบบมัลติเพล็ก(Multiplexed display)

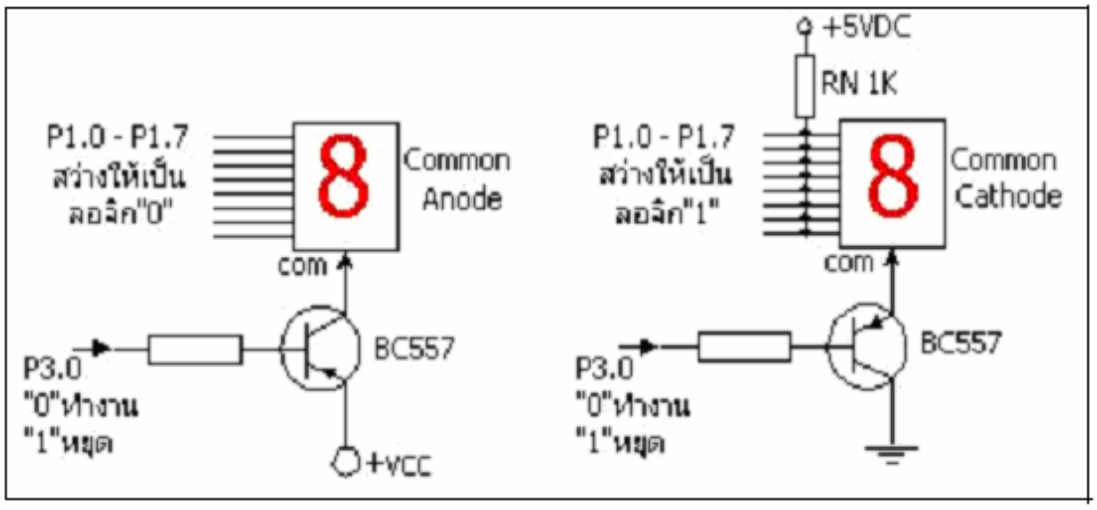
โดยให้แอลอีดี 7 ส่วนติดทีละหลัก จะควบคุมส่วนที่เป็นขาคอมมอน(Common) ของแอลอีดี 7 ส่วน แต่ละตัวให้ทำงาน การแสดงผลแบบมัลติเพล็กแต่ละหลักจะต้องใช้ความเร็วในการสลับหลัก โดยที่สามารถจะมองเห็นการดับของแอลอีดี 7 ส่วน แต่ละหลักได้ทัน ทำให้มองเห็นเหมือนกับว่าแอลอีดี 7 ส่วนแสดงผลทุกหลักติดพร้อมๆกัน ซึ่งการใช้งานในลักษณะการแสดงผลแบบมัลติเพล็ก(Multiplexed display) จะมีข้อดีหลายประการ ตัวอย่างเช่น เมื่อแอลอีดี 7 ส่วนติดได้เพียงทีละหลัก ซึ่งนั่นก็หมายความว่าจะใช้แหล่งจ่ายไฟให้แอลอีดี 7 ส่วนแค่หลักเดียว ถึงแม้จะมีแอลอีดี 7 ส่วนจำนวนหลายหลักก็ตาม และส่วนของตัวต้านทานที่ทำการจำกัดกระแสให้แต่ละเซกเมนต์ ก็จะใช้ทั้งหมดเพียงแค่ 8 ตัว(รวมทั้งที่แสดงผลเป็นจุด Dot) ดังนั้นวิธีนี้จะทำให้ประหยัดอุปกรณ์ที่นำมาต่อร่วม และปริมาณของกระแสไฟที่ต้องใช้ในระบบทั้งหมด

การต่อวงจรสามารถต่อได้ 2 แบบคือ

1.แบบคอมมอนแอโนด (Common Anode)

2. แบบคอมมอนแคโทด (Common Cathode)

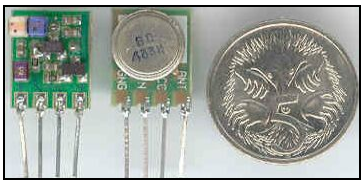
โดยในโครงงานนี้เราเลือกการต่อแบบคอมมอนแอโนด เนื่องจากกระแสที่ออกจากพอร์ตของไมโครคอนโทรลเลอร์นั้นไม่เพียงพอที่จะใช้กับวงจรแบบคอมมอนแคโทดได้

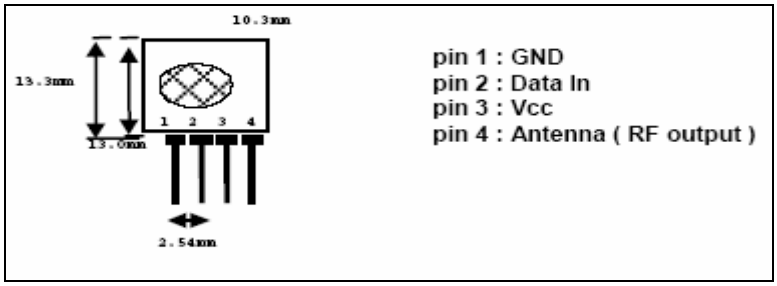


รูปที่2.6 แสดงการใช้ทรานซิสเตอร์แบบ PNP กำหนดสถานะ การทำงานที่ขาคอมมอนแต่ละแบบของแอลอีดี 7 ส่วน

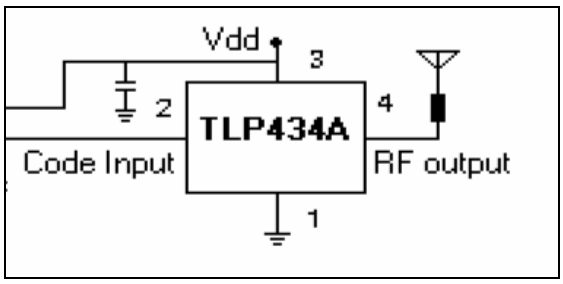
2.5 วงจรภาครับ-ส่งสัญญาณวิทยุ 433 MHz

วงจรภาครับส่งสัญญาณวิทยุนี้จะทำงานที่ความถี่ 433 MHz โดยระยะการทำงานในพื้นที่โล่งจะสามารถส่งสัญญาณได้ไกลถึง 100 เมตร แต่ในพื้นที่ที่มีสิ่งกีดขวางนั้นจะส่งสัญญาณได้ 30 เมตร ซึ่งระยะในการส่งสัญญาณนี้จะขึ้นอยู่กับสิ่งแวดล้อมโดยรอบ สายอากาศ และแรงดันที่ตัวส่งสัญญาณใช้ ซึ่งแรงดัน ที่ใช้จะอยู่ระหว่าง 2-12 V กระแส 20 mA ซึ่งวงจรภาคส่งสัญญาณนี้จะมีอัตราการส่งข้อมูล (Data Rate) 4.8 KB/s โดยมีกำลังเอาท์พุตเท่ากับ 16 DBm

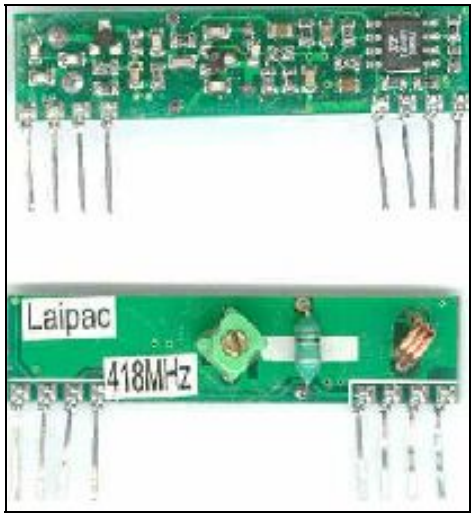


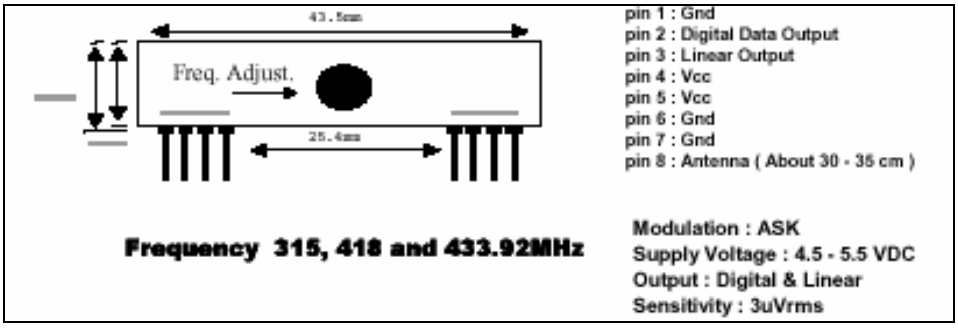


รูปที่2.7 แสดงโครงสร้างภายนอกและตำแหน่งขาของภาคส่งสัญญาณวิทยุ

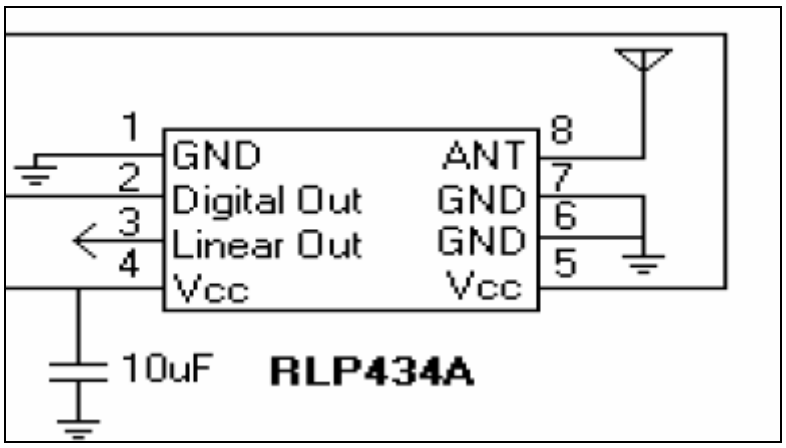


รูปที่2.8 วงจรภาคส่งสัญญาณวิทยุ





รูปที่2.9 แสดงโครงสร้างภายนอกและตำแหน่งขาของภาครับสัญญาณวิทยุ



รูปที่2.10 วงจรภาครับสัญญาณวิทยุ

บทที่ 3
การดำเนินงานและการทดสอบ

ในการทำโครงงานนี้เริ่มจากการออกแบบชิ้นงาน โดยการเลือกอุปกรณ์ที่เหมาะสมกับโครงงาน โดยการหาข้อมูลต่างๆเพื่อให้ทราบถึงคุณสมบัติและการใช้งานของอุปกรณ์ที่เราจะนำมาใช้ในโครงงานนี้ เมื่อเรามีข้อมูลมากพอเราจึงเริ่มลงมือสร้างชิ้นงานโดยเริ่มทีละส่วนดังนี้

ส่วนแรกจะเป็นส่วนของ ไอซีสร้างฐานเวลาจริง (Real Time Clock) โดยเราเลือกใช้ไอซีเบอร์ DS1302 แล้วเราก็ต่อวงจรเข้ากับบอร์ดไมโครคอนโทรลเลอร์ที่เราใช้เรียนในวิชาไมโครคอนโทรลเลอร์ ในขั้นตอนนี้เราทำเพื่อจะศึกษาการทำงานของตัวไอซีโดยออกแบบโปรแกรมให้ไมโครคอนโทรลเลอร์อ่านค่าเวลาจากไอซีสร้างฐานเวลาจริง แล้วส่งค่าออกมาทาง Serial Port แล้วแสดงผลที่โปรแกรม Hyper Terminal โดยในการออกแบบโปรแกรมนี้เราได้ทำการศึกษาจากตัวอย่างหลายๆตัวอย่างที่ได้หาข้อมูลมา

ส่วนที่สองที่เราจะทดลองวงจรแอลอีดี 7 ส่วนแบบหลายหลัก (7-Segment) เพื่อใช้ในการแสดงผลเวลาที่ได้จากการทดลองในส่วนแรก โดยการต่อ 7-Segment แบบ 4 หลักแบบ Common Anode เข้ากับวงจรในส่วนแรกแล้วทำการออกแบบโปรแกรมเพื่อทดสอบการทำงานของ 7-Segment ทั้ง 4 หลัก ให้แสดงค่าตัวเลขตั้งแต่เลข 0 เพิ่มขึ้นทีละ 1 ไปจนถึง 9

เมื่อเราออกแบบโปรแกรมในส่วน7-Segment จนแน่ใจได้ว่ามันสามารถทำงานได้ตามต้องการแล้ว เราจะทำการรวมการทำงานทั้งสองส่วนระหว่างไอซีสร้างฐานเวลาจริง (Real Time Clock) และวงจรแอลอีดี 7 ส่วน เพื่อเปลี่ยนการแสดงผลเวลาจากที่โปรแกรม Hyper Terminalไปเป็นที่ วงจรแอลอีดี 7 ส่วน โดยการออกแบบโปรแกรมให้ทำการนำค่าตัวเลขแต่ละหลักมาแสดงที่ 7-Segment ทั้ง 4 หลักคือ ชั่วโมง 2 หลัก และ นาที 2 หลัก

ส่วนที่สามคือส่วนของไอซีอัดเสียงและเล่นเสียง ในส่วนนี้เนื่องจากการทำงานของไอซี ตัวนี้มีการทำงานที่ซับซ้อนเราจึงได้ซื้อชุดวงจรสำเร็จมาศึกษาการทำงานของไอซี แต่เนื่องจากโค้ดโปรแกรมได้เป็นลิขสิทธิ์ของทางร้านจึงไม่สามารถนำโค้ดโปรแกรมมาเป็นตัวอย่างในการศึกษาได้ แต่ชุดวงจรนี้ก็ช่วยในส่วนของหน้าที่การทำงานของอุปกรณ์ได้เป็นอย่างดี

ทำการออกแบบโปรแกรมโดยศึกษาขั้นตอนการทำงานต่างๆตามคู่มือ (Data Sheet) โดยเริ่มจากการอัดเสียง 0-10 โดยใช้การควบคุมจากไมโครคอนโทรลเลอร์ตัวต้นแบบที่ได้จากทางร้านก่อนแล้วเราจึงทำการออกแบบโปรแกรมให้ทำการเล่นเสียงจากเลข 0-10 เป็นแบบตามลำดับ แล้วทำการอัดโปรแกรมให้กับไมโครคอนโทรลเลอร์อีกหนึ่งตัวซึ่งเป็นคนละตัวกันกับตัวแรกเพื่อจะเก็บตัวแรกไว้สำหรับทดสอบว่าบอร์ดสามารถทำงานได้ เมื่อทำการอัดโปรแกรมเสร็จแล้วจึงนำไป

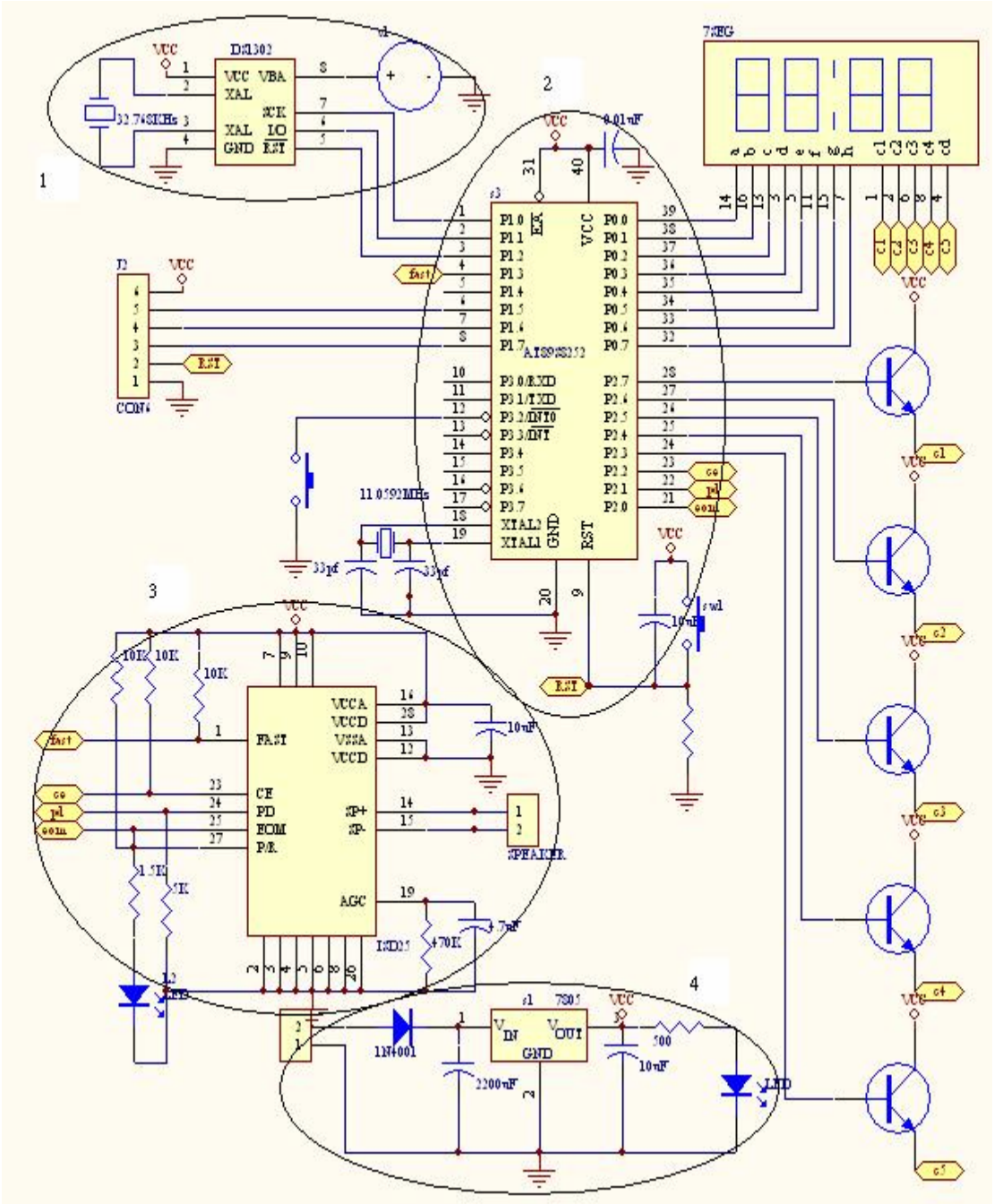
เปลี่ยนลงบอร์ดทดลองแล้วทดสอบดูว่าสามารถทำงานได้ตามต้องการหรือไม่ หากไม่ก็ทำการแก้ไขต่อไป

รูปที่3.1 แสดงขั้นตอนการอัดเสียง

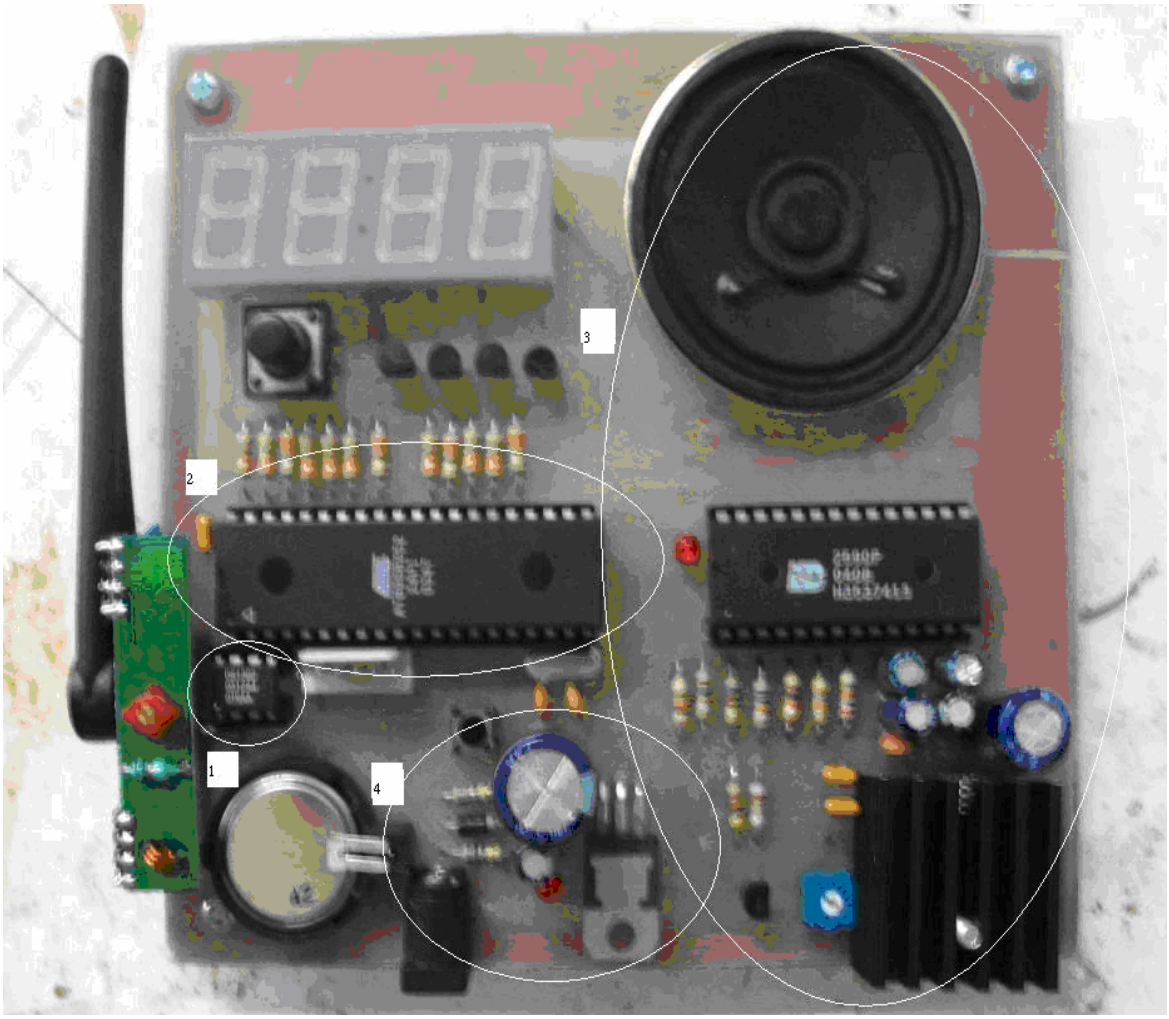
เมื่อเราเข้าใจการเล่นเสียงแบบตามลำดับแล้วขั้นตอนต่อไปเราก็จะออกแบบการเล่นเสียงในรูปแบบ M1 คือจะเป็นการกระโดดข้ามช่องสัญญาณหรือจะเป็นการเล่นแบบไม่ตามลำดับนั่นเองโดยปฏิบัติเหมือนขั้นตอนแรก

ขั้นตอนต่อไปหลังจากเราได้เข้าใจการทำงานของแต่ละส่วนแล้วเราจะทำการออกแบบวงจรที่จะรวมเอาอุปกรณ์ทุกตัวไว้ด้วยกันโดยจะต้องศึกษาการใช้โปรแกรมในการช่วยออกแบบลายวงจรเริ่มจากเขียนโครงร่างลายวงจรทั้งหมดลงในกระดาษเปล่าก่อนตรวจสอบให้แน่ใจว่าไม่มีจุดใดผิดพลาด จากนั้นทำการเขียนไฟล์ Schematic ใส่ Foot-Print ทำการสร้างไฟล์ PCB ขึ้นมาแล้วทำการปริ๊นลายวงจรลงแผ่นใสชนิดที่ทนความร้อนได้ นำไปรีดลงบนแผ่นปริ๊นและขั้นตอนสุดท้ายคือการนำไปแช่ในน้ำยากรดกัดปริ๊น เมื่อได้แผ่นปริ๊นที่มีลายวงจรตามที่ต้องการก็นำไปเจาะรูเพื่อลงอุปกรณ์

รูปที่3.2 แสดงขั้นตอนการสร้างตัววงจร



รูปที่3.3 แสดงไฟล์ Schematic ที่เสร็จเรียบร้อยแล้ว



รูปที่3.4 แสดงวงจรที่ลงอุปกรณ์เสร็จเรียบร้อยแล้ว

จากรูปแต่ละส่วนมีชื่อดังนี้

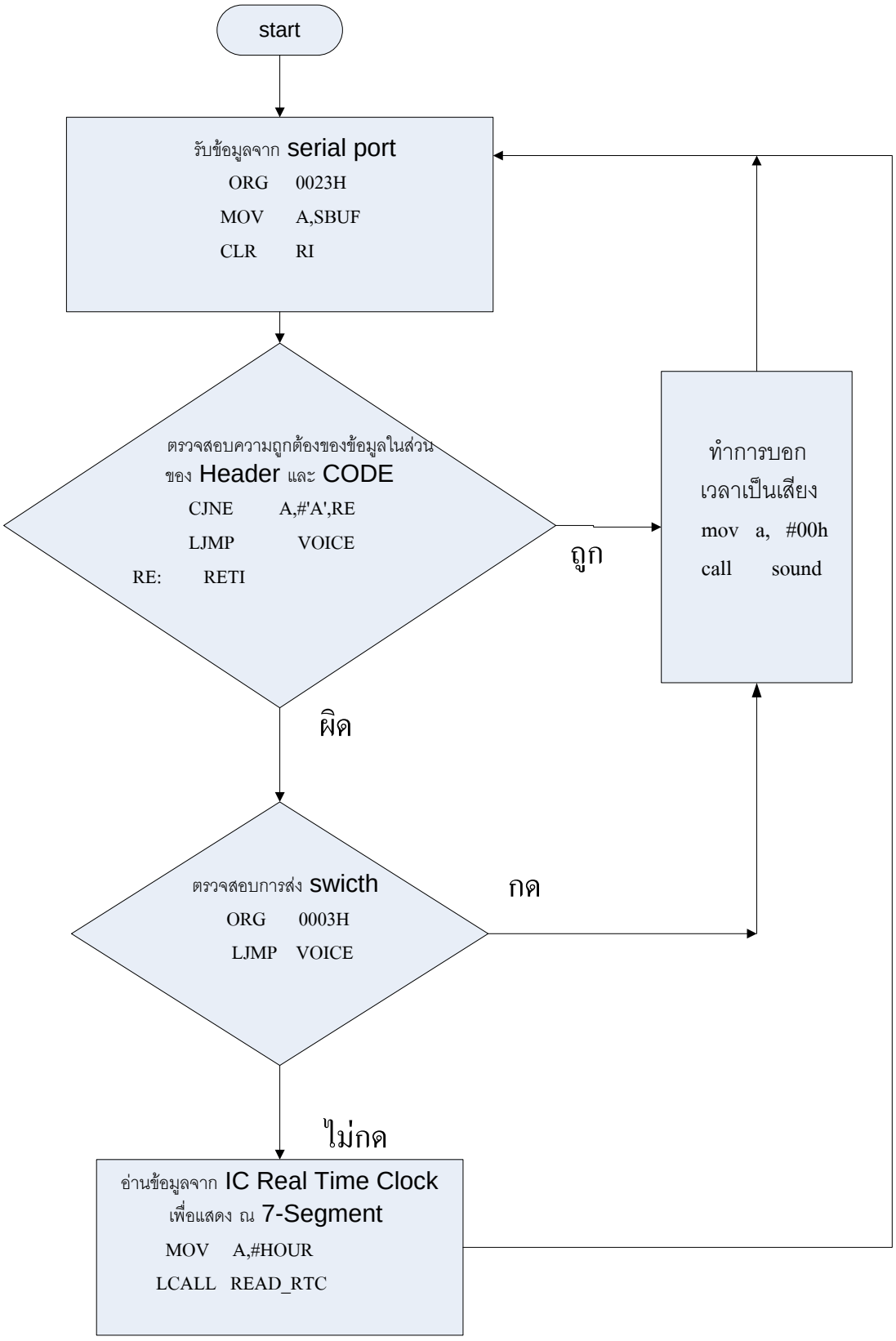
1.ส่วนของไอซีสร้างฐานเวลาจริง DS1302

2.ส่วนของไมโครคอนโทรลเลอร์

3.ส่วนของไอซีอัดเสียงและเล่นเสียง ISD2590

4.ส่วนของวงจรภาคจ่ายไฟหลัก

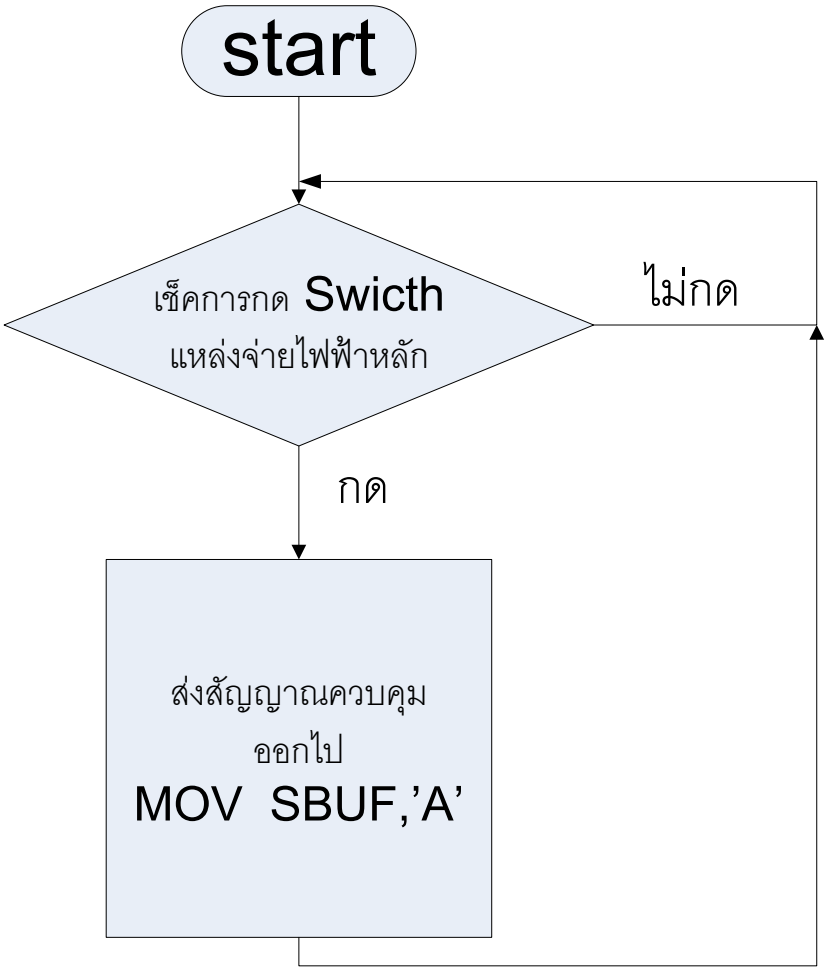
เมื่อได้อุปกรณ์ที่พร้อมใช้งานแล้วต่อไปจะเป็นการทดสอบการทำงานโดยการทดสอบทีละส่วนด้วยการใช้โค้ดโปรแกรมที่เคยออกแบบมาก่อนหน้านี้ เมื่อทดสอบได้ว่าใช้งานได้ครบทุกส่วนแล้วจึงออกแบบโปรแกรมที่มีการทำงานร่วมกัน



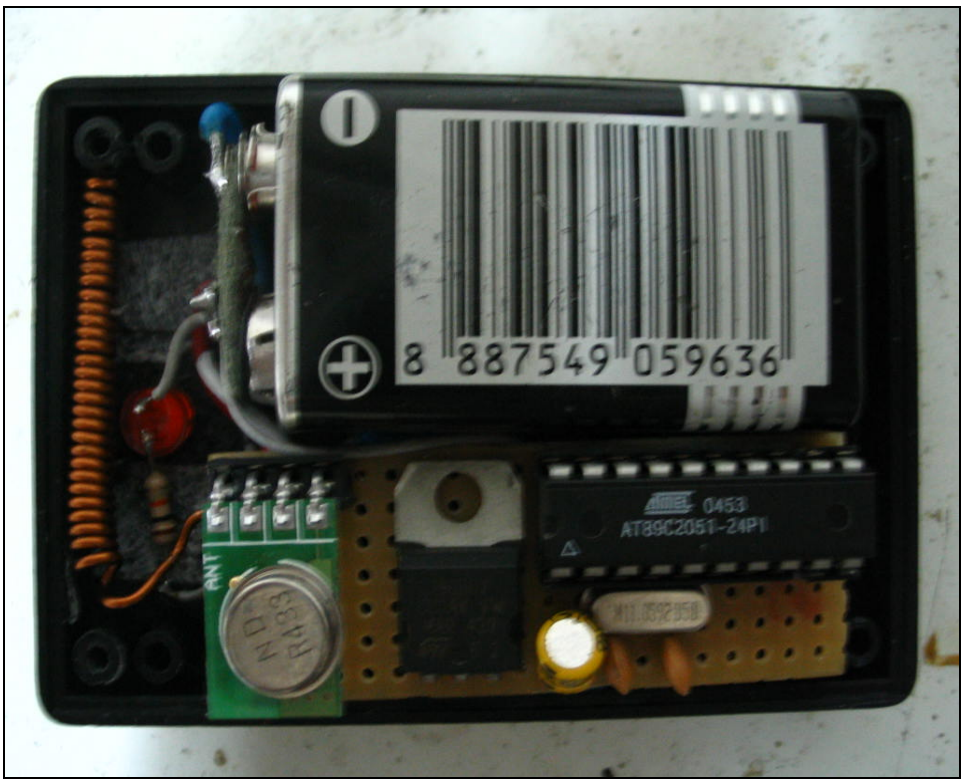
รูปที่3.5 Flow Chart แสดงการทำงานของนาฬิกา

เมื่อทุกส่วนสามารถทำงานรวมกันได้แล้วลำดับต่อไปจะเป็นการตรวจสอบชิ้นงานเพื่อหาข้อบกพร่อง และมีส่วนใดบ้างที่จะสามารถพัฒนาเพิ่มได้

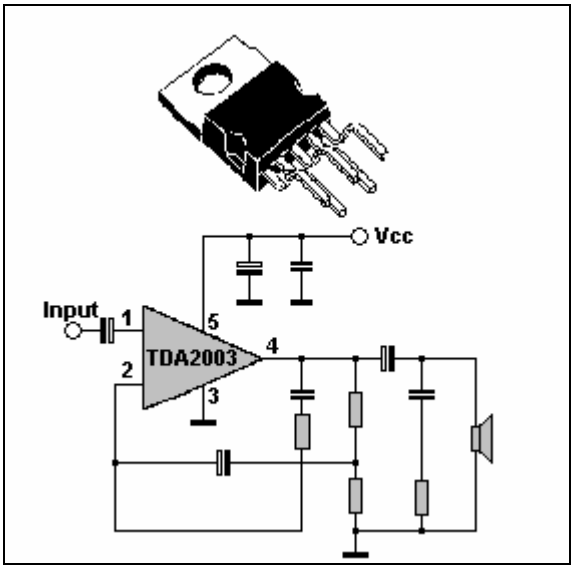
จากนั้นจึงออกแบบส่วนประกอบเพิ่มมาอีกสองส่วนคือ การควบคุมการทำงานโดยใช้การควบคุมผ่านสัญญาณวิทยุ ซึ่งจะมีอุปกรณ์ตัวส่งสัญญาณวิทยุย่านความถี่ 433MH เพิ่มเข้ามาโดยมีวงจรดังรูป และวงจรขยายกำลังของสัญญาณเสียงโดยใช้ไอซีเบอร์ TDA2003



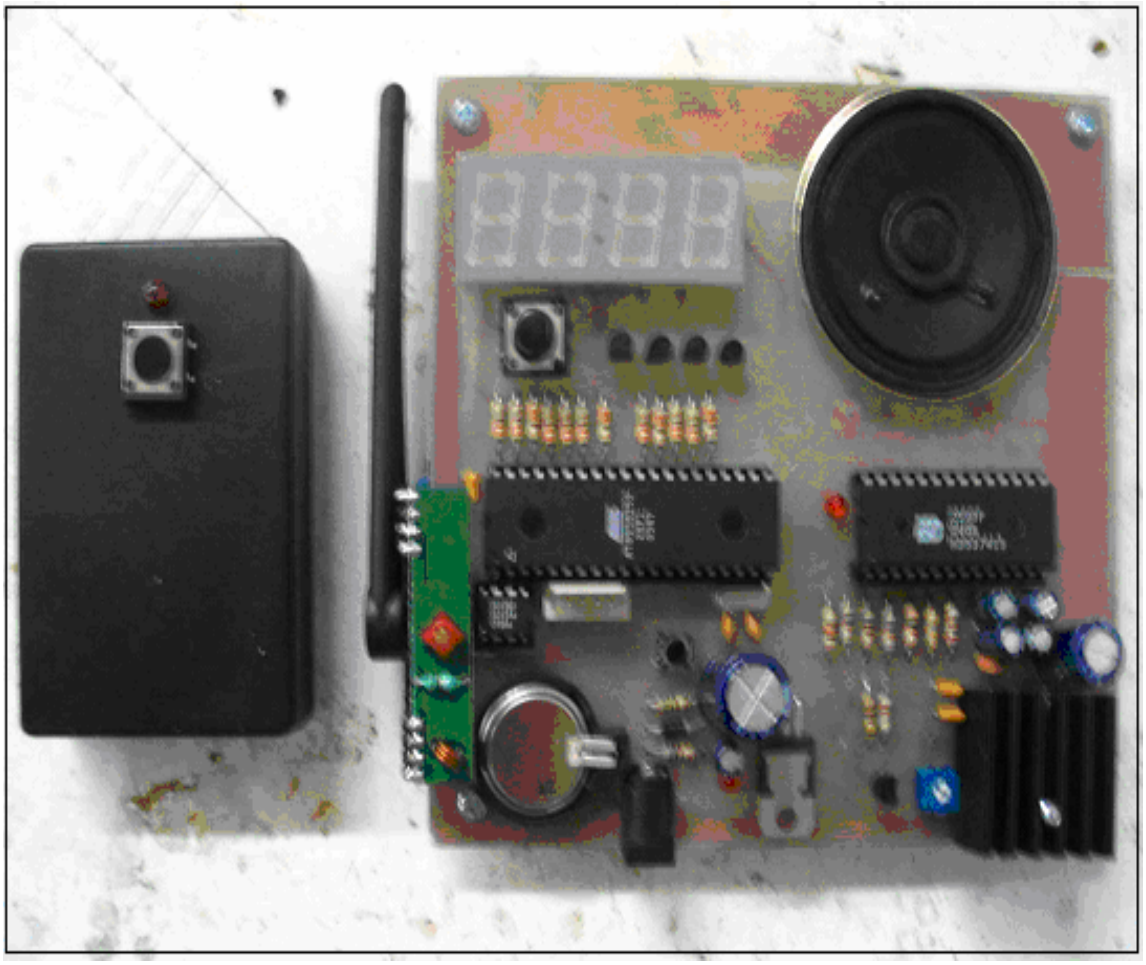
รูปที่3.6 Flow Chart แสดงการทำงานของรีโมท



รูปที่3.7 ภายในของรีโมท RF



รูปที่3.8 วงจรขยายสัญญาณเสียง



รูปที่3.9 แสดงการทดสอบใช้งานจริง

การทดสอบ

ในการนำโครงงานไปใช้จริงจำเป็นต้องทราบถึงขีดจำกัดในการใช้งานเพื่อเป็นประโยชน์สูงสุด ทั้งนี้เราจึงต้องทำการทดสอบเพื่อหาคำตอบเหล่านั้น

เริ่มจากการทดสอบระยะทางที่รีโมทสามารถควบคุมได้ภายในพื้นที่ห้องปฏิบัติการซึ่งระยะทางที่ไกลที่สุด คือ 33 เมตร

จากนั้นทำการทดสอบระยะการรับฟังเสียงได้อย่างชัดเจน ซึ่งระยะทางที่ไกลที่สุด คือ 17 เมตร

บทที่ 4
สรุปผลการทดลองและข้อเสนอแนะ

4.1 สิ่งที่ได้จากจากโครงงาน

- ได้รับความรู้ในการใช้โปรแกรมภาษา Assembly มากขึ้น
- ได้รับความรู้เรื่องการรับ-ส่งข้อมูลดิจิตอลผ่านตัวกลางที่เป็นอากาศ
- ได้รับความรู้เกี่ยวกับการทำงานของอุปกรณ์อิเล็กทรอนิกส์ต่างๆ
- ได้รับความรู้ในการประกอบอุปกรณ์ต่างๆ และสามารถเลือกใช้อุปกรณ์ที่เหมาะสมในโครงงาน
- ทำให้สามารถนำความรู้ที่ได้รับจากการเรียนทฤษฎีมาใช้ในการปฏิบัติจริง
- ทำให้รู้จักการทำงานร่วมกับผู้อื่นและการแก้ไขปัญหาเฉพาะหน้า
- สามารถนำความรู้ที่ได้จากโครงงานมาประยุกต์ใช้งานได้

4.2 ปัญหาและอุปสรรค

- ต้องใช้เวลามากในการเบิกจ่ายงบประมาณจึงทำให้การทำโครงงานล่าช้า
- การควบคุมค่าใช้จ่ายให้อยู่ในงบประมาณมีผลอย่างมากต่อคุณลักษณะของอุปกรณ์ที่นำมาใช้ทำโครงงาน
- ตำราและข้อมูลส่วนใหญ่เป็นภาษาอังกฤษจึงต้องใช้เวลาในการแปลและเรียบเรียงก่อนนำมาใช้

4.3 ข้อจำกัดของโครงงาน

- การติดต่อระหว่างอุปกรณ์ RF ในภาครับ-ส่ง ใช้เทคนิคการมอดูเลตแบบ ASK ซึ่งทนต่อสัญญาณรบกวนได้น้อย

4.4 ข้อเสนอแนะ

- ควรเพิ่มส่วนของการตั้งเวลาบนชุดวงจร
- ควรเพิ่มขนาดลำโพงให้มีคุณภาพที่ดีขึ้น
- ควรเพิ่มส่วนแสดงผลข้อมูล เช่น วันที่ วัน เดือน ปี เป็นต้น
- ควรทำให้มีขนาดเล็กและสามารถพกพาได้สะดวก

บรรณานุกรม

[1] บุญสืบ โพธิ์ศรี, สมหมาย ปานเขียว และโกมล ศิริสมบูรณ์เวช. (2546). ไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้น. กรุงเทพฯ: ศูนย์ส่งเสริมอาชีวะ

[2] ธีรวัฒน์ ประกอบผล.การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.สำนักพิมพ์ ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).2545

[3] ธีรวัฒน์ ประกอบผล.ภาษาแอสเซมบลีสำหรับ MSC-51.สำนักพิมพ์ ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).2545

[4] วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตรวิไล. (n.d.). เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช. กรุงเทพฯ: อินโนเวตีฟ เอ็กเพอริเมนต์

[5] Adisak.**แอลอีดี 7 ส่วน แบบหลายหลัก**.Available from : URL:http://www.adisak51.com/

[6] Adisak. **ไมโครคอนโทรลเลอร์ MCS-51**.Available from : URL:http://www.adisak51.com/

[7] **ISD2590**. Available from : www.winbond-usa.com/products/isd_products/chipcorder/datasheets/2560/ISD2560.pdf

[8] **AT89S8252**.Available from : www.ezl.com/~rsch/FTP/1302.pdf

[9] The **DS1302** Trickle Charge Timekeeping Chip contains a real time clock. **DS1302**.Available from : www.ezl.com/~rsch/FTP/1302.pdf

ภาคผนวก

DS1302

DALLAS SEMICONDUCTOR

DS1302
Trickle Charge Timekeeping Chip

www.maxim-ic.com

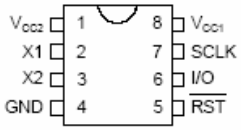
FEATURES

- Real-time clock (RTC) counts seconds, minutes hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 31-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Serial I/O for minimum pin count
- 2.0V to 5.5V full operation
- Uses less than 300nA at 2.0V
- Burst mode for reading/writing successive addresses in clock/RAM
- 8-pin DIP or optional 8-pin SOICs for surface mount
- Simple 3-wire interface
- TTL-compatible (V_{CC} = 5V)
- Optional industrial temperature range: -40°C to +85°C
- DS1202 compatible
- Underwriters Laboratory (UL) recognized

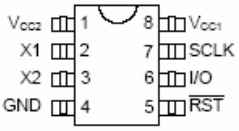
ORDERING INFORMATION

DS1302	8-Pin DIP (300-mil)
DS1302N	8-Pin DIP (Industrial)
DS1302S	8-Pin SOIC (200-mil)
DS1302SN	8-Pin SOIC (Industrial)
DS1302Z	8-Pin SOIC (150-mil)
DS1302ZN	8-Pin SOIC (Industrial)
DS1302S-16	16-Pin SOIC (300-mil)
DS1302SN-16	16-Pin SOIC (Industrial)

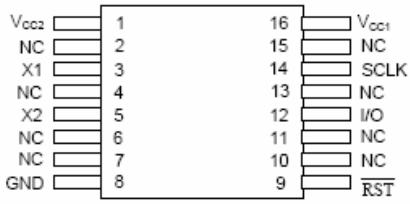
PIN ASSIGNMENT



DS1302 8-Pin DIP (300-mil)



DS1302 8-Pin SOIC (200-mil)
DS1302 8-Pin SOIC (150-mil)



DS1302 16-Pin SOIC (300-mil)

PIN DESCRIPTION

X1, X2	- 32.768kHz Crystal Pins
GND	- Ground
$\overline{RST}$	- Reset
I/O	- Data Input/Output
SCLK	- Serial Clock
V_{CC1}, V_{CC2}	- Power Supply Pins

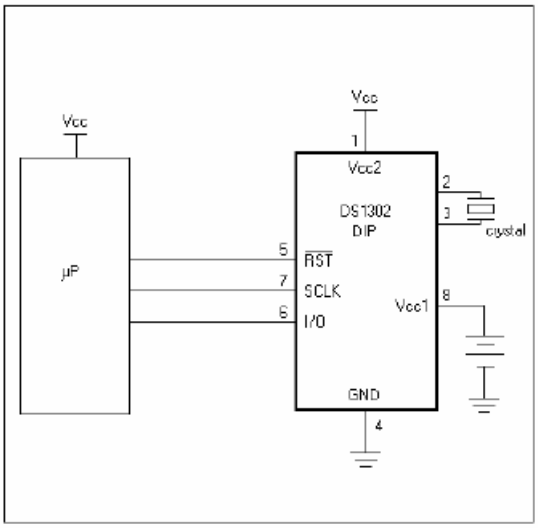
DESCRIPTION

The DS1302 Trickle Charge Timekeeping Chip contains an RTC/calendar and 31 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The RTC/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

Interfacing the DS1302 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: 1) $\overline{RST}$ (reset), 2) I/O (data line), and 3) SCLK (serial clock). Data can be transferred to and from the clock/RAM 1 byte at a time or in a burst of up to 31 bytes. The DS1302 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

The DS1302 is the successor to the DS1202. In addition to the basic timekeeping functions of the DS1202, the DS1302 has the additional features of dual-power pins for primary and back-up power supplies, programmable trickle charger for V_{CC1}, and seven additional bytes of scratchpad memory.

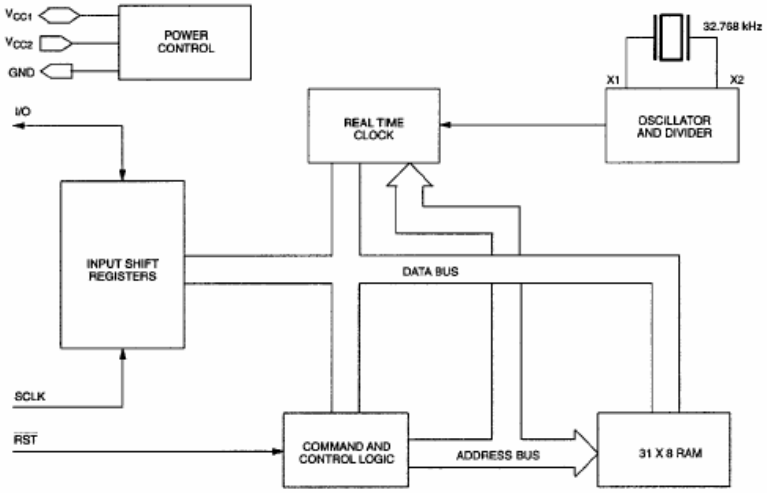
TYPICAL OPERATING CIRCUIT



OPERATION

The main elements of the serial timekeeper (i.e., shift register, control logic, oscillator, RTC, and RAM) are shown in Figure 1.

DS1302 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC1} – V_{CC1} provides low-power operation in single supply and battery-operated systems as well as low-power battery backup. In systems using the trickle charger, the rechargeable energy source is connected to this pin. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See "Conditions of Acceptability" at http://www.maxim-ic.com/TechSupport/QA/ntrl.htm.

V_{CC2} – V_{CC2} is the primary power supply pin in a dual-supply configuration. V_{CC1} is connected to a backup source to maintain the time and date in the absence of primary power.

The DS1302 will operate from the larger of V_{CC1} or V_{CC2}. When V_{CC2} is greater than V_{CC1} + 0.2V, V_{CC2} will power the DS1302. When V_{CC2} is less than V_{CC1}, V_{CC1} will power the DS1302.

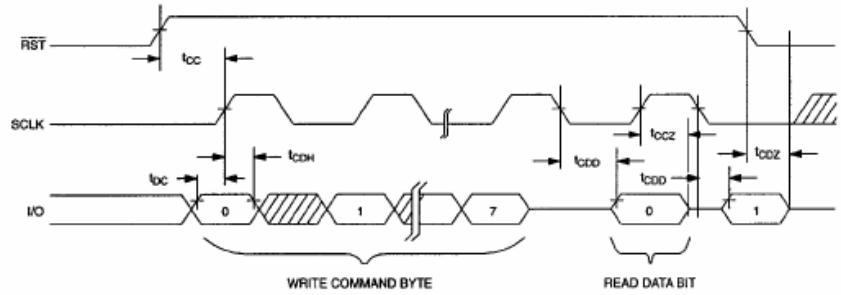
SCLK (Serial Clock Input) – SCLK is used to synchronize data movement on the serial interface. This pin has a 40kΩ internal pull-down resistor.

I/O (Data Input/Output) – The I/O pin is the bi-directional data pin for the 3-wire interface. This pin has a 40kΩ internal pull-down resistor.

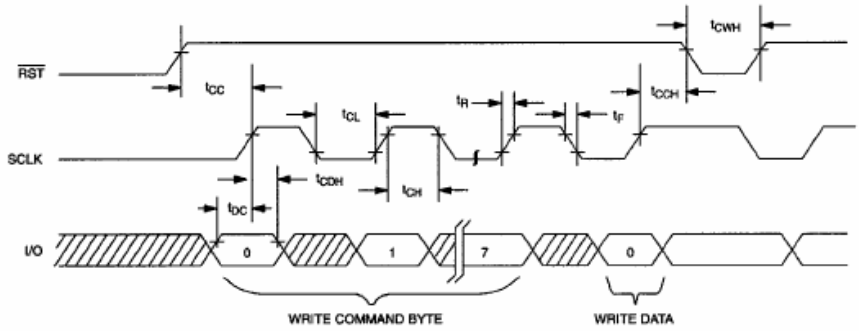
$\overline{RST}$ (Reset) – The reset signal must be asserted high during a read or a write. This pin has a 40kΩ internal pull-down resistor.

X1, X2 – Connections for a standard 32.768kHz quartz crystal. The internal oscillator is designed for operation with a crystal having a specified load capacitance of 6pF. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations

TIMING DIAGRAM: READ DATA TRANSFER Figure 5



TIMING DIAGRAM: WRITE DATA TRANSFER Figure 6



NOTES:

1. I_{CC1T} and I_{CC2T} are specified with I/O open, $\overline{RST}$ set to a logic 0, and clock halt flag = 0 (oscillator enabled).
2. I_{CC1A} and I_{CC2A} are specified with the I/O pin open, $\overline{RST}$ high, SCLK=2MHz at V_{CC} = 5V; SCLK = 500kHz, V_{CC} = 2.0V, and clock halt flag = 0 (oscillator enabled).
3. $\overline{RST}$, SCLK, and I/O all have 40kΩ pull-down resistors to ground.
4. Measured at V_{IH} = 2.0V or V_{IL} = 0.8V and 10ns maximum rise and fall time.
5. Measured at V_{OH} = 2.4V or V_{OL} = 0.4V.
6. Load capacitance = 50pF.
7. I_{CC1S} and I_{CC2S} are specified with $\overline{RST}$, I/O, and SCLK open. The clock halt flag must be set to logic one (oscillator disabled).
8. $V_{CC} = V_{CC2}$, when $V_{CC2} > V_{CC1} + 0.2V$; $V_{CC} = V_{CC1}$, when $V_{CC1} > V_{CC2}$.
9. V_{CC2} = 0V.
10. V_{CC1} = 0V.
11. Typical values are at 25°C.

ISD2590

ISD2560/75/90/120

SINGLE-CHIP, MULTIPLE-MESSAGES,

VOICE RECORD/PLAYBACK DEVICE

60-, 75-, 90-, AND 120-SECOND DURATION

Winbond Electronics Corp.

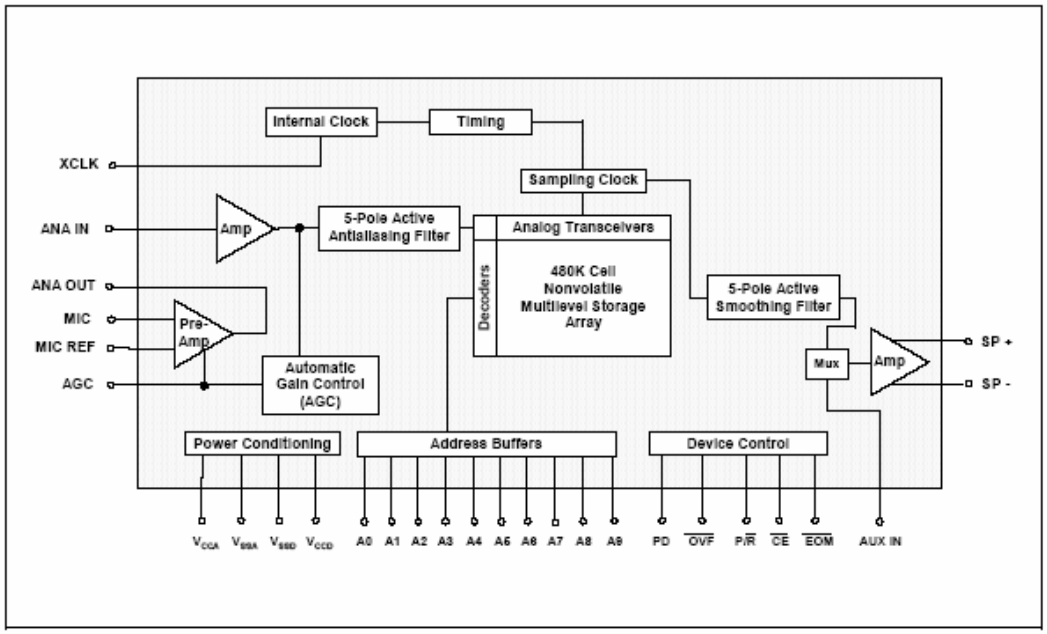
1. GENERAL DESCRIPTION

Winbond's ISD2500 ChipCorder® Series provide high-quality, single-chip, Record/Playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multi-level storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved. Recordings are stored into on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through Winbond's patented multilevel storage technology. Voice and audio signals are stored directly into memory in their natural form, providing high-quality, solid-state voice reproduction.

2. FEATURES

- Easy-to-use single-chip, voice record/playback solution
- High-quality, natural voice/audio reproduction
- Single-chip with duration of 60, 75, 90, or 120 seconds.
- Manual switch or microcontroller compatible
- Playback can be edge- or level-activated
- Directly cascadable for longer durations
- Automatic power-down (push-button mode)
  - Standby current 1 µA (typical)
- Zero-power message storage
  - Eliminates battery backup circuits
- Fully addressable to handle multiple messages
- 100-year message retention (typical)
- 100,000 record cycles (typical)
- On-chip clock source
- Programmer support for play-only applications
- Single +5 volt power supply
- Available in die form, PDIP, SOIC and TSOP packaging
- Temperature = die (0°C to +50°C) and package (0°C to +70°C)

3. BLOCK DIAGRAM



7. FUNCTIONAL DESCRIPTION

7.1. DETAILED DESCRIPTION

Speech/Sound Quality

The Winbond's ISD2500 series includes devices offered at 4.0, 5.3, 6.4, and 8.0 kHz sampling frequencies, allowing the user a choice of speech quality options. Increasing the duration within a product series decreases the sampling frequency and bandwidth, which affects the sound quality. Please refer to the ISD2560/75/90/120 Product Summary table below to compare the duration, sampling frequency and filter pass band.

The speech samples are stored directly into the on-chip nonvolatile memory without any digitization and compression associated like other solutions. Direct analog storage provides a very true, natural sounding reproduction of voice, music, tones, and sound effects not available with most solid state digital solutions.

Duration

To meet various system requirements, the ISD2560/75/90/120 products offer single-chip solutions at 60, 75, 90, and 120 seconds. Parts may also be cascaded together for longer durations.

TABLE 1: ISD2560/75/90/120 PRODUCT SUMMARY

Part Number	Duration (Seconds)	Input Sample Rate (kHz)	Typical Filter Pass Band * (kHz)
ISD2560	60	8.0	3.4
ISD2575	75	6.4	2.7
ISD2590	90	5.3	2.3
ISD25120	120	4.0	1.7

* 3db roll-off point

EEPROM Storage

One of the benefits of Winbond's ChipCorder® technology is the use of on-chip nonvolatile memory, providing zero-power message storage. The message is retained for up to 100 years typically without power. In addition, the device can be re-recorded typically over 100,000 times.

Microcontroller Interface

In addition to its simplicity and ease of use, the ISD2500 series includes all the interfaces necessary for microcontroller-driven applications. The address and control lines can be interfaced to a microcontroller and manipulated to perform a variety of tasks, including message assembly, message concatenation, predefined fixed message segmentation, and message management.

Programming

The ISD2500 series is also ideal for playback-only applications, where single or multiple messages are referenced through buttons, switches, or a microcontroller. Once the desired message configuration is created, duplicates can easily be generated via a gang programmer.

7.2. OPERATIONAL MODES

The ISD2500 series is designed with several built-in Operational Modes that provide maximum functionality with minimum external components. These modes are described in details as below. The Operational Modes are accessed via the address pins and mapped beyond the normal message address range. When the two Most Significant Bits (MSB), A8 and A9, are HIGH, the remaining address signals are interpreted as mode bits and not as address bits. Therefore, Operational Modes and direct addressing are not compatible and cannot be used simultaneously.

There are two important considerations for using Operational Modes. First, all operations begin initially at address 0 of its memory. Later operations can begin at other address locations, depending on the Operational Mode(s) chosen. In addition, the address pointer is reset to 0 when the device is changed from record to playback, playback to record (except M6 mode), or when a Power-Down cycle is executed.

Second, Operational Modes are executed when $\overline{CE}$ goes LOW. This Operational Mode remains in effect until the next LOW-going $\overline{CE}$ signal, at which point the current mode(s) are sampled and executed.

TABLE 2: OPERATIONAL MODES

Mode [1]	Function	Typical Use	Jointly Compatible [2]
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete $\overline{EOM}$ markers	Position $\overline{EOM}$ marker at the end of the last message	M3, M4, M5, M6
M2	Not applicable	Reserved	N/A
M3	Looping	Continuous playback from Address 0	M1, M5, M6
M4	Consecutive addressing	Record/playback multiple consecutive messages	M0, M1, M5
M5	$\overline{CE}$ level-activated	Allows message pausing	M0, M1, M3, M4
M6	Push-button control	Simplified device interface	M0, M1, M3

[1] Besides mode pin needed to be "1", A8 and A9 pin are also required to be "1" in order to enter into the related operational mode.

[2] Indicates additional Operational Modes which can be used simultaneously with the given mode.

7.2.1. Operational Modes Description

The Operational Modes can be used in conjunction with a microcontroller, or they can be hardwired to provide the desired system operation.

M0 – Message Cueing

Message Cueing allows the user to skip through messages, without knowing the actual physical addresses of each message. Each $\overline{CE}$ LOW pulse causes the internal address pointer to skip to the next message. This mode is used for playback only, and is typically used with the M4 Operational Mode.

M1 – Delete $\overline{EOM}$ Markers

The M1 Operational Mode allows sequentially recorded messages to be combined into a single message with only one $\overline{EOM}$ marker set at the end of the final message. When this Operational Mode is configured, messages recorded sequentially are played back as one continuous message.

M2 – Unused

When Operational Modes are selected, the M2 pin should be LOW.

M3 – Message Looping

The M3 Operational Mode allows for the automatic, continuously repeated playback of the message located at the beginning of the address space. A message can completely fill the ISD2500 device and will loop from beginning to end without $\overline{OVF}$ going LOW.

M4 – Consecutive Addressing

During normal operation, the address pointer will reset when a message is played through an $\overline{EOM}$ marker. The M4 Operational Mode inhibits the address pointer reset on $\overline{EOM}$, allowing messages to be played back consecutively.

M5 - $\overline{CE}$-Level Activated

The default mode for ISD2500 devices is for $\overline{CE}$ to be edge-activated on playback and level-activated on record. The M5 Operational Mode causes the $\overline{CE}$ pin to be interpreted as level-activated as opposed to edge-activated during playback. This is especially useful for terminating playback operations using the $\overline{CE}$ signal. In this mode, $\overline{CE}$ LOW begins a playback cycle, at the beginning of the device memory. The playback cycle continues as long as $\overline{CE}$ is held LOW. When $\overline{CE}$ goes HIGH, playback will immediately end. A new $\overline{CE}$ LOW will restart the message from the beginning unless M4 is also HIGH.

M6 – Push-Button Mode

The ISD2500 series contain a Push-Button Operational Mode. The Push-Button Mode is used primarily in very low-cost applications and is designed to minimize external circuitry and components, thereby reducing system cost. In order to configure the device in Push-Button Operational Mode, the two most significant address bits must be HIGH, and the M6 mode pin must also be HIGH. A device in this mode always powers down at the end of each playback or record cycle after $\overline{CE}$ goes HIGH.

When this operational mode is implemented, three of the pins on the device have alternate functionality as described in the table below.

TABLE 3: ALTERNATE FUNCTIONALITY IN PINS

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{CE}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse-activated)
$\overline{EOM}$	Active-HIGH Run Indicator

$\overline{CE}$ (START/PAUSE)

In Push-Button Operational Mode, $\overline{CE}$ acts as a LOW-going pulse-activated START/PAUSE signal. If no operation is currently in progress, a LOW-going pulse on this signal will initiate a playback or record cycle according to the level on the P/$\overline{R}$ pin. A subsequent pulse on the $\overline{CE}$ pin, before an $\overline{EOM}$ is reached in playback or an overflow condition occurs, will pause the current operation, and the address counter is not reset. Another $\overline{CE}$ pulse will cause the device to continue the operation from the place where it is paused.

PD (STOP/RESET)

In Push-Button Operational Mode, PD acts as a HIGH-going pulse-activated STOP/RESET signal. When a playback or record cycle is in progress and a HIGH-going pulse is observed on PD, the current cycle is terminated and the address pointer is reset to address 0, the beginning of the message space.

$\overline{EOM}$ (RUN)

In Push-Button Operational Mode, $\overline{EOM}$ becomes an active-HIGH RUN signal which can be used to drive an LED or other external device. It is HIGH whenever a record or playback operation is in progress.

Recording in Push-Button Mode

1. The PD pin should be LOW, usually using a pull-down resistor.

Good Audio Design Practices

Winbond products are very high-quality single-chip voice recording and playback systems. To ensure the highest quality voice reproduction, it is important that good audio design practices on layout and power supply decoupling be followed. See Application Information or below links for details.

Good Audio Design Practices

http://www.winbond-usa.com/products/isd_products/chipcorder/applicationinfo/apin11.pdf

Single-Chip Board Layout Diagrams

http://www.winbond-usa.com/products/isd_products/chipcorder/applicationinfo/apin12.pdf

8. TIMING DIAGRAMS

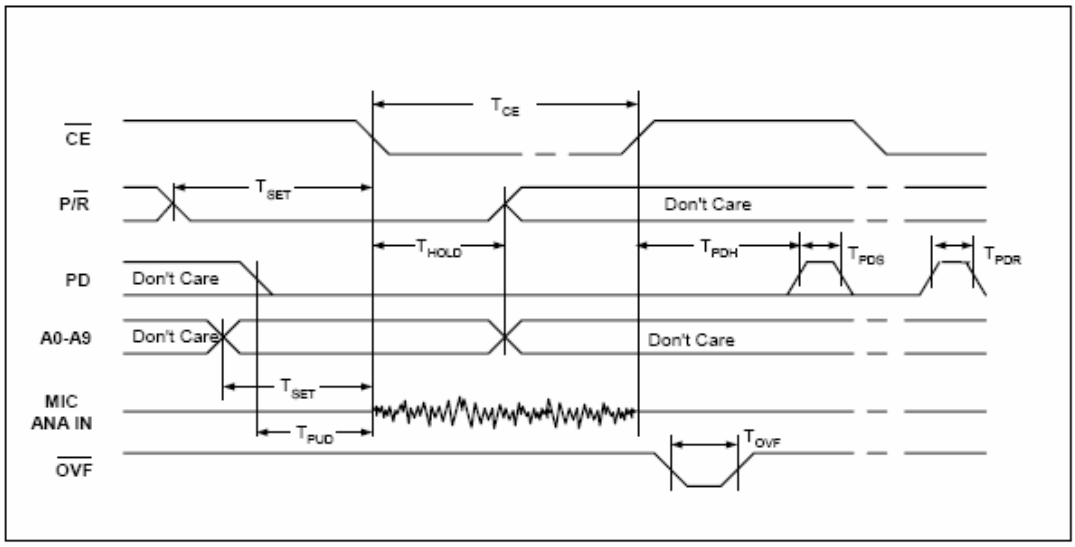


FIGURE 1: RECORD

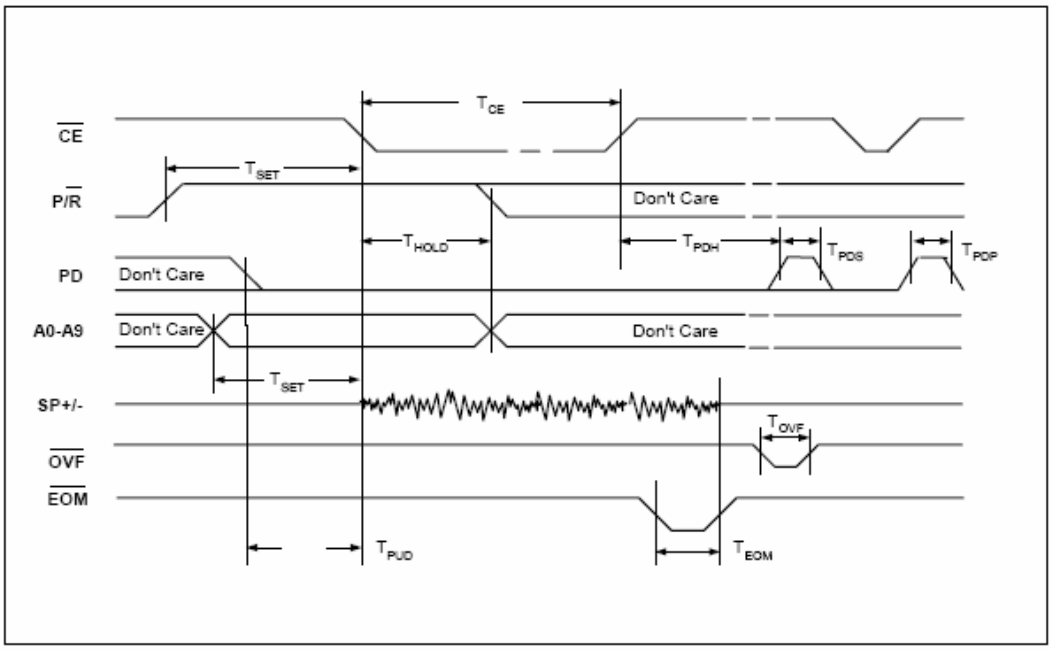


FIGURE 2: PLAYBACK

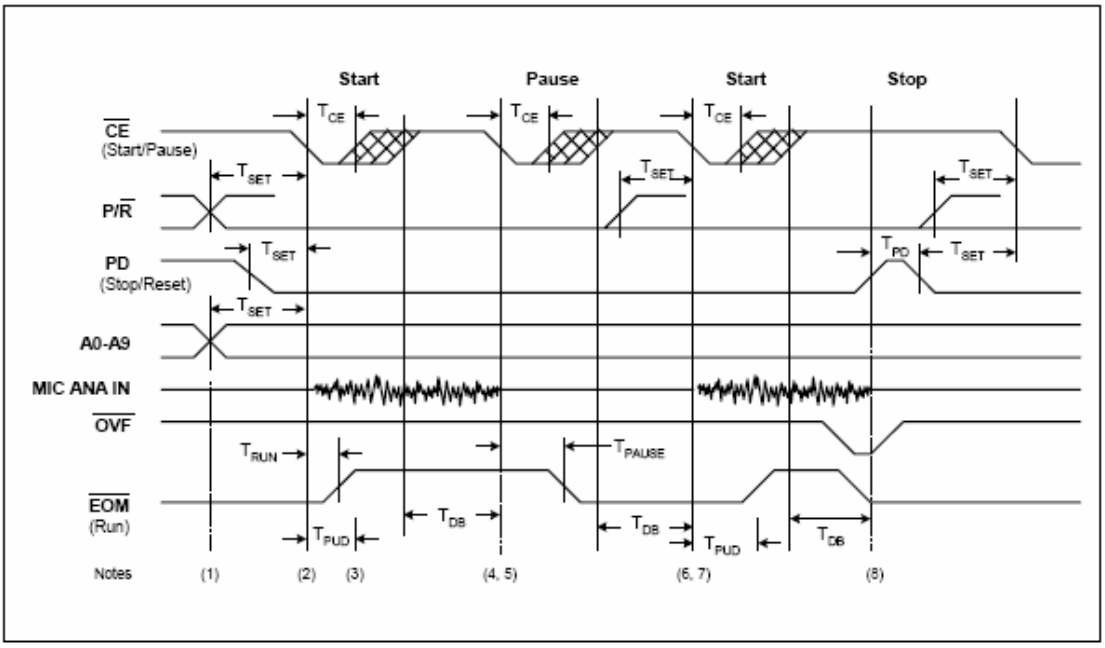


FIGURE 3: PUSH-BUTTON MODE RECORD

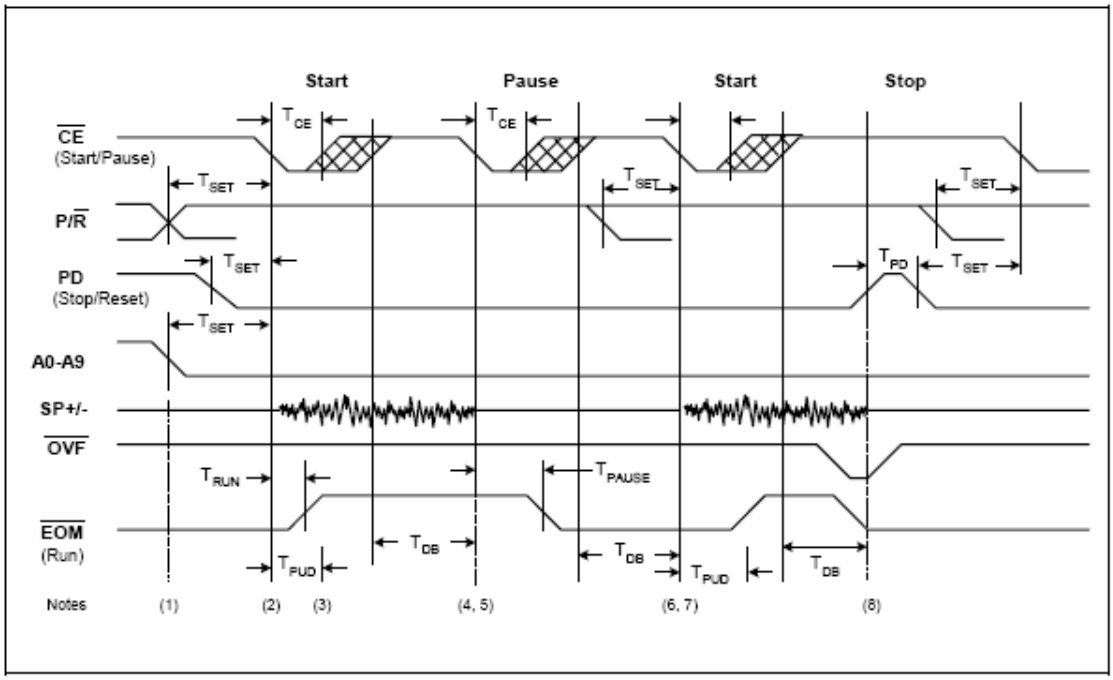


FIGURE 4: PUSH-BUTTON MODE PLAYBACK

Notes for Push-Button modes:

1. A9, A8, and A6 = 1 for push-button operation.
2. The first $\overline{CE}$ LOW pulse performs a start function.
3. The part will begin to play or record after a power-up delay T_{PUD}.
4. The part must have $\overline{CE}$ HIGH for a debounce period T_{DB} before it will recognize another falling edge of $\overline{CE}$ and pause.
5. The second $\overline{CE}$ LOW pulse, and every even pulse thereafter, performs a Pause function.
6. Again, the part must have $\overline{CE}$ HIGH for a debounce period T_{DB} before it will recognize another falling edge of $\overline{CE}$, which would restart an operation. In addition, the part will not do an internal power down until $\overline{CE}$ is HIGH for the T_{DB} time.
7. The third $\overline{CE}$ LOW pulse, and every odd pulse thereafter, performs a Resume function.
8. At any time, a HIGH level on PD will stop the current function, reset the address counter, and power down the device.

ไมโครคอนโทรลเลอร์ AT89S8252

Features

- Compatible with MCS®51 Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of downloadable Flash programmable and erasable read-only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed In-System through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcontroller, which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

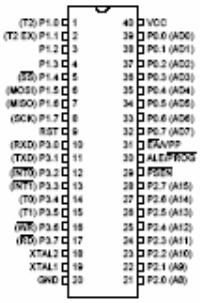
The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless lock bits have been activated.

8-bit Microcontroller with 8K Bytes Flash

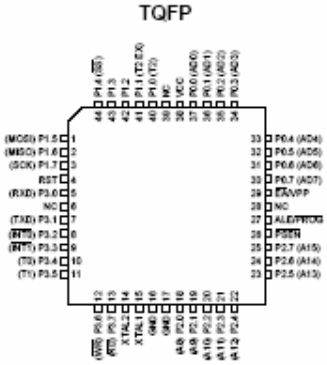
AT89S8252

Pin Configurations

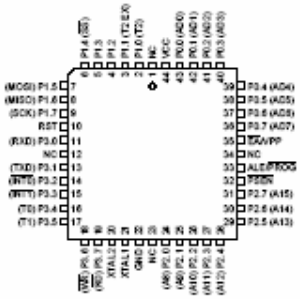
PDIP



TQFP



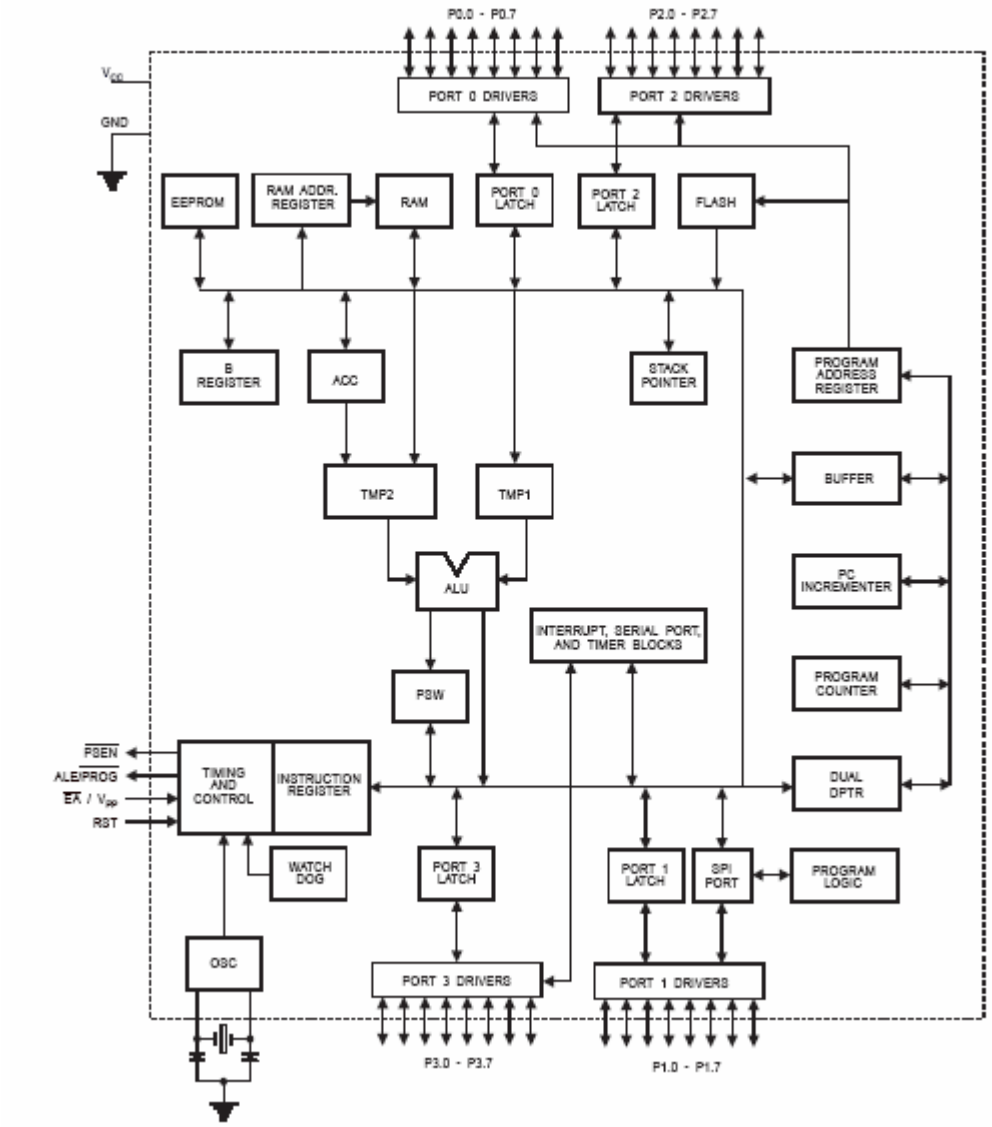
PLCC



Pin Description

VCC	Supply voltage.
GND	Ground.
Port 0	Port 0 is an 8-bit open drain bi-didirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.
Port 1	Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Block Diagram



Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

ไมโครคอนโทรลเลอร์ AT89C2051

Features

- Compatible with MCS®-51Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes
- Green (Pb/Halide-free) Packaging Option

1. Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read-only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

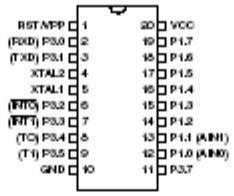
ATMEL®

8-bit Microcontroller with 2K Bytes Flash

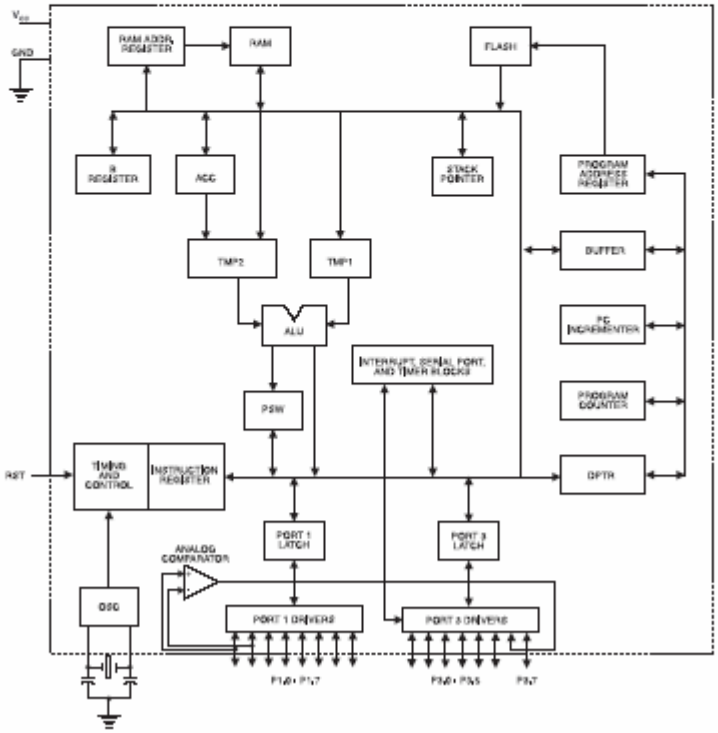
AT89C2051

2. Pin Configuration

2.1 20-lead PDIP/SOIC



3. Block Diagram



4. Pin Description

4.1 VCC

Supply voltage.

4.2 GND

Ground.

4.3 Port 1

The Port 1 is an 8-bit bi-directional I/O port. Port pins P1.2 to P1.7 provide internal pull-ups. P1.0 and P1.1 require external pull-ups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives code data during Flash programming and verification.

4.4 Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-directional I/O pins with internal pull-ups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general-purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and verification.

4.5 RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

4.6 XTAL1

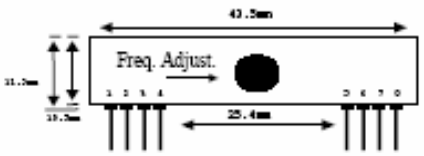
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

4.7 XTAL2

Output from the inverting oscillator amplifier.

ชุดรับส่งคลื่นวิทยุ ความถี่ 433 MHz

RLP-434 Receiver



pin 1 : Gnd
pin 2 : Digital Data Output
pin 3 : Linear Output
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 : Antenna (About 30 - 35 cm)

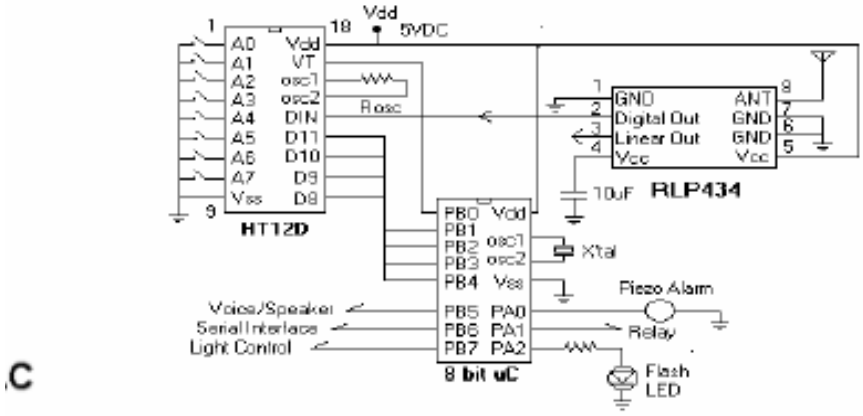
Frequency 315, 418 and 433.92MHz

Modulation : ASK
Supply Voltage : 4.6 - 5.5 VDC
Output : Digital & Linear
Sensitivity : 3uVrms

Symbol	Parameter	Conditions	Min	Typ	Max	
Vcc	Operating supply voltage		4.5	5	5.5	V
Itot	Operating Current		-	3.5	4.5	mA
Vdata	Data Out	Idata = +200 uA (High)	Vcc-0.5	-	Vcc	V
		Idata = -10 uA (Low)	-	-	0.3	V

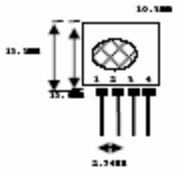
Electrical Characteristics					
Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC	315, 418 and 434			MHz
Sensitivity	Pref	-100	-103	-106	dBm
Channel Width			+-1.5		Khz
Receiver Turn On Time			5		ms
Noise equivalent BW	NEB		4		Khz
Baseboard Data Rate			3	5	Khz

Application Circuit II:
Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from Holtek Semiconductor Inc.



TLP434A Ultra Small Transmitter

Easy-Link Wireless



pin 1 : GND
pin 2 : Data In
pin 3 : Voo
pin 4 : Antenna (RF output)

Frequency 315, 418 and 433.92 Mhz

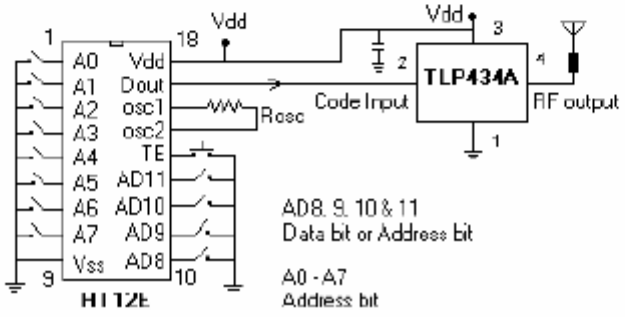
Modulation : ASK
Operation Voltage : 2 - 12 VDC

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Icc 1	Peak Current (2V)		-	-	1.64	mA
Icc 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	Idata= 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	Idata= 0 uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315Mhz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	External Encoding	512	4.8K	200K	bps

Notes : (Case Temperature = 25°C +- 2°C , Test Load Impedance = 50 ohm)

Application Circuit :

Typical Key-chain Transmitter using HT12E-18DIP, a Binary 12 bit Encoder from Holtek Semiconductor Inc.



Source Code

```
                ORG     0000H
                JMP     MAIN
                ORG     0003H
                LJMP    VOICE
                ORG     000BH
                MOV     P0,#11111111B
                CLR     H1
                CLR     H2
                CLR     M1
                CLR     M2
                DJNZ    R7,ll
                CPL     D
                CALLl   LOOP
                MOV     R7,#0FH
ll:             RETI
                ORG     0023H
                MOV     A,SBUF
                CLR     RI
                CJNE    A,#'A',RE
                SETB    D
                LJMP    VOICE
RE:             RETI
                ORG     0100H
MAIN:           RTC_RESET  EQU     P1.2          ; Chips Select DS1202
                RTC_DATA   EQU     P1.1          ; Data I/O DS1202
                RTC_CLOCK  EQU     P1.0          ; Clock DS1202
                SECOND     EQU     00H           ; Second 00-59
                MINUTE     EQU     01H           ; Minute 00-59
                HOUR       EQU     02H           ; Hour 01-12 or 00-23
```

```
CTRL_RTC   EQU     07H
H1         EQU     p2.6
H2         EQU     p2.5
M1         EQU     p2.4
M2         EQU     p2.3
D          EQU     p2.7
fast       equ     p3.4              ;set port isd2590
ce         equ     p3.5
pd         equ     p3.6
eom        equ     p3.7

SETB    RTC_CLOCK          ; Initial Port RTC-DS1202
SETB    RTC_DATA
CLR     RTC_RESET

mov     IE,#10010011b
mov     scon,#50h
mov     th1,#0fdh
mov     tmod,#021h
mov     th0,#0ffh
mov     tl0,#0ffh
clr     it0
setb    tr0
setb    tr1

LCALL     DISABLE           ; Write Protect OFF
MOV       A,#SECOND
LCALL     READ_RTC
CLR       ACC.7             ; Disable Clock Halt Flag
MOV       B,A
```

```
        MOV     A,#SECOND
        LCALL   WRITE_RTC
        LCALL   ENABLE          ; Write Protect ON
                                ;call    set
              setb    p3.3
              mov     r7,#0fh
              call    loop

loop2:
              mov     p0,#0ffh
              clr     M2
              setb    H1
              mov     a,R0
            LCALL   seven


              mov     p0,#0ffh
              CLR     H1
              setb    H2
              mov     a,r1
            LCALL   seven

             SETB    TR0
              mov     p0,#0ffh
              clr     H2
              setb    M1
              mov     a,r6
             LCALL   seven
              mov     p0,#0ffh
              clr     M1
```

```
        setb    M2
        mov     a,r3
       LCALL   seven
        SJMP    LOOP2
LOOP:
        MOV     A,#HOUR
        LCALL   READ_RTC
        SWAP    A
        ANL     A,#0FH
        mov     r0,a

        MOV     A,#HOUR
        LCALL   READ_RTC
        ANL     A,#0FH
        mov     r1,a

        MOV     A,#MINUTE
        LCALL   READ_RTC
        SWAP    A
        ANL     A,#0FH
        Mov     r6,a
        MOV     A,#MINUTE
        LCALL   READ_RTC
        ANL     A,#0FH
        mov     r3,a
          ret
seven:
          mov           dptr,#table
          movc          a,@a+dptr
          mov           p0,a
```

```
            nop
            RET

table:      db      0c0h,0f9h,0a4h,0b0h
            db      99h,92h,82h,0f8h
            db      80h,90h

voice:      mov     p0,#10111111b
            setb    H1
            setb    H2
            setb    M1
            setb    M2


            call    delay
            call    delay
            call    delay
            call    delay
            call    delay
            call    delay
            call    delay
            call    delay

;;;;;;;;;;;;;ขนะนี้เวลา;;;;;;;;;;;;;;;;;;
            mov     a,#00h
            call    sound

;;;;;;;;;;;;;ตัวเลขที่1;;;;;;;;;;;;;;;;
            cjne    r0,#00h,l11
            jmp     ll1
```

```
l11:    cjne    r1,#01h,l1

        mov     a,r0
        add     a,#12h
        call    sound
        jmp     l2
l1:
        mov     a,r0
        add     a,#0ch
        call    sound
        cjne    r1,#00h,ll1
        jmp     l2
ll1:    mov     a,r1
        add     a,#03h
        call    sound

;;;;;;;;;;;นาฬิกา;;;;;;;;;;;;;;;;;
l2:     mov     a,#01h
        call    sound
;;;;;;;;;;;ตัวเลนที่2;;;;;;;;;;;;;;;;
        cjne    r6,#00h,l33
        jmp     ll2
l33:    cjne    r3,#01h,l3

        mov     a,r6
        add     a,#12h
        call    sound
        jmp     l4
l3:     mov     a,r6
        add     a,#0ch
```

```
        call    sound
        cjne    r3,#00h,ll2
        jmp     l4
ll2:    mov     a,r3
        add     a,#03h
        call    sound
;;;;;;;;;;นาที;;;;;;;;;;;;;;;;;;;;;
l4:     mov     a,#02h
        call    sound
        setb    tr0
        reti

sound:
        clr     pd

        nop
        nop
xy:     cjne    a,#00h,xx
        jmp     yy
xx:     setb    fast
        nop
        nop
        clr     ce
        nop
        nop
        setb    ce
        nop
        nop
        nop
        call    delay
```

```
        dec     a
        jmp     xy
yy:     clr     fast
        call    delay
        clr     ce
        nop
        nop
        setb    ce
        call    delay
        call    delay
        call    delay
        call    delay
        call    delay
        call    delay
        call    delay
        call    delay
        clr   p3.3
        jb      eom,$
        setb    p3.3
        setb    pd
        call    delay
        ret
delay:
        MOV   R4,#04fh
DEl:    MOV   R5,#0ffh
        DJNZ  R5,$
        DJNZ  R4,DEl
        ret
```

```
set:                                ;set time
        LCALL    DISABLE
        MOV      B,#011h          ;set hour
        MOV      A,#HOUR          ;
        LCALL    WRITE_RTC
        MOV     B,#005h ;set minut
        MOV     A,#MINUTE        ;
        LCALL   WRITE_RTC
        LCALL   ENABLE
           clr      IE0
           ret
READ_RTC:  LCALL   START_RST          ; Start Communication
        MOV     R2,#8
        RL       A                ; Shift A0 to A1
        SETB    ACC.7             ; Command Status
        CLR     ACC.6             ; Ram Accress
        SETB    ACC.0             ; Read Command
        CLR     CY
READ_RTC1:  LCALL   CLOCK_LO          ; Falling Clock
                RRC     A                 ; Send D0..D7 (Read-Command)
                MOV     RTC_DATA,C        ; Prepare Port
                NOP                       ; Wait Bus Ready
               NOP
               NOP
               NOP
               LCALL   CLOCK_HI           ; Clock
               DJNZ    R2,READ_RTC1       ; Repeat write Command
        MOV     B,#0
        MOV     R2,#8
```

```
READ_RTC2:  SETB    RTC_DATA          ; Initial for Input
            NOP
            NOP
            SETB    RTC_DATA          ; Initial for Input
            LCALL   CLOCK_LO          ; Read data D0..D7 (Data)
            NOP                       ; Wait Data From RTC
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            MOV     C,RTC_DATA        ; Read data out bit
            MOV     A,B
            RRC     A                 ; Read D0..D7
            MOV     B,A               ; Backup Data
            LCALL   CLOCK_HI          ; Rising Clock
            DJNZ    R2,READ_RTC2
            MOV     A,B               ; Data Output 1-Byte
            LCALL   STOP_RST          ; Disable Communication
            RET
WRITE_RTC:  LCALL   START_RST         ; Enable Communication
            MOV     R2,#8
            RL      A                 ; Shift A0 to A1
            SETB    ACC.7             ; Command Status
            CLR     ACC.6             ; Ram Accress
            CLR     ACC.0             ; Write Command
            CLR     CY
WRITE_RTC1: LCALL   CLOCK_LO          ; Falling Clock
```

```
        RRC     A               ; Write D0..D7 (Command)
        MOV     RTC_DATA,C
        NOP                     ; Wait Bus Ready
        NOP
        NOP
        NOP
        LCALL   CLOCK_HI        ; Rising Clock
        DJNZ    R2,WRITE_RTC1   ; Repeat write Command
        MOV     A,B             ; Get Data
        MOV     R2,#8
        CLR     CY

WRITE_RTC2: LCALL   CLOCK_LO
        RRC     A               ; Write D0..D7
        MOV     RTC_DATA,C
        NOP                     ; Wait Bus Ready
        NOP
        NOP
        NOP
        LCALL   CLOCK_HI        ; Rising Clock
        DJNZ    R2,WRITE_RTC2   ; Repeat write Data
        LCALL   STOP_RST        ; Disable Communication
        RET

DISABLE:    MOV     A,#CTRL_RTC
            MOV     B,#00H      ; Disable Write-Protect
            LCALL   WRITE_RTC
            RET

ENABLE:     MOV     A,#CTRL_RTC
```

```
        MOV     B,#80H          ; Enable Write-Protect
        LCALL   WRITE_RTC
        RET

CLOCK_LO:  CLR    RTC_CLOCK       ; Falling Clock
        NOP
        NOP
        NOP
        NOP
        RET

CLOCK_HI:  SETB   RTC_CLOCK       ; Rising Clock
        NOP
        NOP
        NOP
        NOP
        RET

START_RST: CLR    RTC_RESET
        CLR    RTC_CLOCK
        NOP
        NOP
        NOP
        NOP
        SETB   RTC_RESET        ; Enable Communication
        NOP
        NOP
        NOP
        NOP
        RET
```

```
STOP_RST:  SETB   RTC_CLOCK
           NOP
           NOP
           NOP
           NOP
           CLR    RTC_RESET          ; Disable Communication
           NOP
           NOP
           NOP
           NOP
           RET

       END
```

ประวัติผู้เขียน

นายประสิทธิ์ บุญเอนก เกิดวันที่ 23 กุมภาพันธ์ พ.ศ. 2527 ภูมิลำเนาอยู่บ้านเลขที่ 730 ถนนราชดำเนิน ตำบลในเมือง อำเภอเมือง จังหวัดกำแพงเพชร จบการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนกำแพงเพชรพิทยาคม จังหวัดกำแพงเพชร ปีการศึกษา 2544 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

นางสาวเจือจิต จับกลาง เกิดวันที่ 1 กรกฎาคม พ.ศ. 2527 ภูมิลำเนาอยู่บ้านเลขที่ 146 หมู่ที่ 12 ตำบลบิง อำเภอโนนสูง จังหวัดนครราชสีมา จบการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนศรีสุขวิทยา จังหวัดนครราชสีมา ปีการศึกษา 2544 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา