สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

มัคคุเทศก์อัจฉริยะ

Intelligence Guide

นายศรณัฐ เจนถนอมม้า
นายศุภกร ฤทธิชัย

วพ.
ศ1382
2549

เลขหมู่.....
เลขทะเบียน 72964
วัน,เดือน,ปี 26 มิ.ย. 2550

b 11772622
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัคคุเทศก์อัจฉริยะ

Intelligence Guide

จัดทำโดย

นายศรณัฐ เจนถนอมม้า รหัสประจำตัว 46010761

นายศุภกร ฤทธิชัย รหัสประจำตัว 46010789

อาจารย์ที่ปรึกษา

อ.เจริญ วงษ์ชุ่มเย็น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง มัคคุเทศก์อัจฉริยะ

Intelligence Guide

ผู้จัดทำ

1. นายศรณัฐ เจนถนอมม้า รหัสประจำตัว 46010761
2. นายศุภกร ฤทธิชัย รหัสประจำตัว 46010789

_____อาจารย์ที่ปรึกษา

(อ. เจริญ วงษ์ชุ่มเย็น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัคคุเทศก์อัจฉริยะ

นายศรณัฐ เจนถนอมม้า 46010761
นายศุภกร ฤทธิชัย 46010789
อ.เจริญ วงษ์ชุ่มเย็น อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

ในปัจจุบันนี้ (พ.ศ.2549) การท่องเที่ยวเป็นอุตสาหกรรมประเภทหนึ่งที่นำรายได้เข้าประเทศไทยได้อย่างมาก ดังนั้นเมื่อนักท่องเที่ยวจากประเทศต่างๆเดินทางมายังประเทศไทยเราจึงควรทำให้พวกเค้าเหล่านั้นประทับใจมากที่สุด แต่ปัญหาสำคัญก็คือความแตกต่างทางภาษาและวัฒนธรรม ทั้งนี้ด้วยคนไทยสามารถพูดภาษาต่างประเทศได้น้อย จึงทำให้นักท่องเที่ยวกระทำในสิ่งที่ขัดต่อวัฒนธรรมของคนไทย เช่น ปีนไปบนยอดเจดีย์เพื่อถ่ายภาพ เป็นต้น

ผู้พัฒนาจึงได้มีแนวความคิดในการออกแบบเครื่องมือซึ่งช่วยให้นักท่องเที่ยวได้ท่องเที่ยวในประเทศไทยได้อย่างไม่มีปัญหาทางด้านภาษา เราจึงได้ออกแบบอุปกรณ์ให้ข้อมูลแบบอัตโนมัติหรือมัคคุเทศก์อัจฉริยะ โดยระบบมัคคุเทศก์อัจฉริยะนั้นจะแบ่งออกเป็น 2 ส่วนด้วยกันคือ ส่วนอุปกรณ์พกพาและส่วนสถานีสัญญาณ โดยส่วนอุปกรณ์พกพานั้นนักท่องเที่ยวสามารถพกติดตัวไปได้ และส่วนสถานีสัญญาณนั้นจะต้องทำการติดตั้งไว้ยังพื้นที่ที่ต้องการ การทำงานของระบบก็คือเมื่ออุปกรณ์พกพานั้นเข้าไปอยู่ในพื้นที่ของสถานีสัญญาณใดๆ อุปกรณ์พกพานั้นจะทำการเล่นไฟล์เสียงคำบรรยายที่ตรงกับไอดีที่ได้รับมาจากสถานีสัญญาณมายังหูฟังของนักท่องเที่ยว

โดยผู้พัฒนาได้ทำการค้นคว้าวิจัย เปรียบเทียบเทคโนโลยีต่างๆเพื่อนำมาใช้ในการพัฒนา จากนั้นจะทำการออกแบบฮาร์ดแวร์แต่ละส่วน ทดสอบ จนแต่ละส่วนนั้นใช้งานได้ จากนั้นจึงทำการรวมระบบและทดสอบจนได้ตัวต้นแบบที่ใช้งานได้ เราหวังว่าอุปกรณ์ต้นแบบชิ้นนี้จะสามารถนำไปพัฒนาต่อและนำไปใช้งานเชิงพาณิชย์ได้จริง เป็นประโยชน์ต่ออุตสาหกรรมการท่องเที่ยวของประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Intelligence Guide

Soranat Jenthanomma 46010761

Supagorn Ritthichai 46010789

Charoen Vongchumyen Advisor

Academic Year 2006

ABSTRACT

Today (2006), one of the most important industrial of Thailand is traveling industrial. It earns much money for Thailand so when the traveler from other country came to Thailand; we must make them to impressive in that traveling. But the big problem is the different language and Thai people don't have good knowledge of foreign language enough. So we can't explain the information of the traveling place good enough. It cause that the foreign travelers incline to do something that resist to our culture such as climb up to the top of Je-dee to take a picture.

So that, we want to solve that problem by invent some device that help the foreign travelers to travel in Thailand without the problem of language. So we invent the "Intelligence Guide" or "iGUIDE". The iGUIDE system has two section, iGUIDE Portable-the portable device that travelers can portable it while they travel, and iGUIDE Station-the device that we install in the area that we want to informed the travelers. When iGUIDE Portable is in the iGUIDE Station's area, it will detect the ID of that area from iGUIDE Station. Then, iGUIDE Portable will get the information MP3 file of that area from MMC to play it to the traveler.

The developers have maked research, compare and selected technology. Then we designed hardware each sections and test them to make sure that they work. So, when all sections worked, we integrated all sections and tested it to be a worked prototype. We hope it can be developed in the future and wish this iGUIDE system can be use in real situation and be beneficial to Thailand's traveling.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์เสร็จลงได้ก็คือ อ.เจริญ วงษ์ชุ่มเย็น ซึ่งเป็นที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำความรู้และความช่วยเหลือในทุกๆด้านเสมอมา อีกทั้งยังคอยให้กำลังใจข้าพเจ้าตราบจนถึงวินาทีสุดท้าย ซึ่งต้องขอกราบขอบพระคุณเป็นอย่างยิ่ง

และต้องขอขอบพระคุณบุคคลสำคัญที่สุด ที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายศรณัฐ เจนถนอมม้า

นายสุภกร ฤทธิชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวเรื่อง	หน้าที่
บทคัดย่อ ภาษาไทย	I
บทคัดย่อ ภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	
1.1 ความเป็นมาของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตของงานวิจัย	2
1.5 ส่วนประกอบของรายงาน	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 โครงสร้างของไมโครคอนโทรลเลอร์ AT89C51SND1C	4
2.1.1 โครงสร้างขาสัญญาณภายนอก	4
2.1.2 โครงสร้างภาพรวมภายใน	6
2.1.3 โครงสร้างหน่วยความจำ	7
2.1.4 โครงสร้างระบบสัญญาณนาฬิกา	10
2.1.5 โครงสร้างส่วนการถอดรหัสเอ็มพีสาม (MP3 Decoder)	11
2.1.6 โครงสร้างส่วนติดต่อกับเอ็มเอ็มซี (MMC)	13
2.1.7 โครงสร้างส่วนการส่งสัญญาณเสียงออก(Audio Output)	21
2.2 การใช้งานเอ็มเอ็มซี	23
2.2.1 ความรู้ทั่วไปเกี่ยวกับเอ็มเอ็มซี	23
2.2.2 ลักษณะการเชื่อมต่อและโปรโตคอล	24
2.2.3 คำสั่งและการทำงาน	31
2.3 การเชื่อมต่อแบบไอสแควร์ซีแบบสองเส้น (Two-wire Interface I^2C)	32
2.4 การเชื่อมต่อภายนอกแบบเข้าจังหวะ (Synchronous Peripheral Interface)	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง	หน้าที่
2.5 การเชื่อมต่อแบบยูเอสบี (Universal Serial Bus Interface)	39
2.6 การใช้งานส่วนการติดต่อผ่านคลื่นสัญญาณวิทยุด้วยโมดูล TRW-2.4G	46
2.7 ทฤษฎีและการใช้งานไอซีเสียง ISD2590	51
บทที่ 3 การออกแบบและพัฒนา	
3.1 ภาพรวมของการออกแบบ	55
3.1.1 กำหนดคุณลักษณะการใช้งาน	55
3.1.2 พิจารณาเลือกใช้เทคโนโลยี	55
3.1.3 ออกแบบบล็อกไดอะแกรม (Block Diagram)	57
3.2 การออกแบบส่วนฮาร์ดแวร์	59
3.2.1 การออกแบบผังวงจรส่วนพกพา (iGUIDE Portable)	59
3.2.2 ออกแบบลายวงจรส่วนอุปกรณ์พกพา	65
3.2.3 ออกแบบผังวงจรส่วนสถานีสัญญาณ (iGUIDE Station)	66
3.2.4 ออกแบบลายวงจรส่วนสถานีสัญญาณ	68
3.3 การออกแบบส่วนโปรแกรมที่ใช้ควบคุมฮาร์ดแวร์ (Firmware)	69
3.3.1 ออกแบบสถานะการทำงาน (State Diagram)	69
3.3.2 ออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) ทั้งคีย์แพดและจอแอลซีดี	71
3.3.3 ออกแบบส่วนการทำงานของอุปกรณ์พกพา	73
3.3.4 ออกแบบส่วนการทำงานของสถานีสัญญาณ	74
บทที่ 4 การทดลองและผลการทดลอง	
4.1 ทดลองเขียน AT89C51SND1.h	76
4.2 ทดสอบโค้ดโปรแกรมส่วนแอลซีดี บนโปรแกรมจำลองการทำงานของไมโครคอนโทรลเลอร์ตระกูล 51	77
4.3 ทดสอบโค้ดไอสแควร์ซีบนโปรแกรมจำลองการทำงานของไมโครคอนโทรลเลอร์ตระกูล 51	78
4.4 การทดลองภาคจ่ายไฟ	79
4.5 การทดลองใช้งานไมโครคอนโทรลเลอร์ผ่านไอเอสพี	80
4.6 การทดลองใช้งานสวิทช์	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง	หน้าที่
4.7 การทดลองใช้งานส่วนเอ็มเอ็มซี	81
4.8 การทดลองส่วนสื่อสาร (TRW-2.4G)	82
4.9 การทดลองส่วนสัญญาณเสียง	84
4.10 การทดสอบรวมระบบ	84
บทที่ 5 บทวิจารณ์และสรุป	
5.1. บทสรุป	85
5.2. ปัญหาอุปสรรคและแนวทางแก้ไข	86
5.3. แนวทางการพัฒนาต่อ	87
ภาคผนวก ก ผังวงจรและลายวงจรส่วนต่างๆ	89
ภาคผนวก ข ซอร์สโค้ดโปรแกรมควบคุมฮาร์ดแวร์	91
บรรณานุกรม	97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้าที่
ตารางที่ 2.1	แสดง Extended RAM (ERAM)	9
ตารางที่ 2.2	แสดงข้อมูลของเฟรม	13
ตารางที่ 2.3	แสดงการตอบรับแบบปกติทั่วไป ความยาว 48 บิต มี CRC	16
ตารางที่ 2.4	แสดงการตอบรับการร้องขอ CID, CSD ความยาว 136 บิต มี CRC	16
ตารางที่ 2.5	แสดงการตอบรับการร้องขอ OCR ความยาว 48 บิต แต่ไม่มี CRC	16
ตารางที่ 2.6	แสดงการตอบรับอื่นเกี่ยวกับการรับ-ส่งข้อมูล ความยาว 48 บิตมี CRC	17
ตารางที่ 2.7	แสดงรูปแบบข้อมูลกับการตั้งค่าของส่วนส่งสัญญาณเสียงออก	23
ตารางที่ 2.8	ข้อเปรียบเทียบรูปแบบการเชื่อมต่อของเอ็มเอ็มซี	24
ตารางที่ 2.9	แสดงความต้องการการใช้พลังงานของเอ็มเอ็มซี	28
ตารางที่ 2.10	แสดงการเชื่อมต่อเอ็มเอ็มซีแบบ MMC Interface	29
ตารางที่ 2.11	แสดงการเชื่อมต่อเอ็มเอ็มซีแบบ เอสพีไอ	30
ตารางที่ 2.12	แสดงคำสั่งและการทำงานของเอ็มเอ็มซี	31
ตารางที่ 2.13	แสดงคำสั่งการตั้งค่าต่างๆของโมดูล TRW-2.4G	50
ตารางที่ 3.1	แสดงการพิจารณาเลือกใช้เทคโนโลยีเพื่อเชื่อมต่อกับคอมพิวเตอร์	55
ตารางที่ 3.2	แสดงการพิจารณาเลือกใช้เทคโนโลยีเพื่อติดต่อแบบไร้สาย กับสถานีสัญญาณ	56
ตารางที่ 3.3	แสดงการพิจารณาเลือกใช้เทคโนโลยีอุปกรณ์เก็บข้อมูล	57
ตารางที่ 3.4	ตารางแสดงการออกแบบส่วนติดต่อกับผู้ใช้	71
ตารางที่ 4.1	แสดงการใช้งานคำสั่งต่างและการตอบสนองของเอ็มเอ็มซี	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้าที่
รูปที่ 2.1	แสดงขาสัญญาณภายนอกของไมโครคอนโทรลเลอร์ AT89C51SND1C	5
รูปที่ 2.2	แสดงโครงสร้างภาพรวมภายในของไมโครคอนโทรลเลอร์ เบอร์ AT89C51SND1C	6
รูปที่ 2.3	แสดงโครงสร้างของหน่วยความจำ	8
รูปที่ 2.4	แสดง RAM ส่วนล่าง	8
รูปที่ 2.5	แสดง SFR ของ AT89C51SND1C	9
รูปที่ 2.6	แสดงโครงสร้างของระบบสัญญาณนาฬิกา	10
รูปที่ 2.7	แสดงการทำงานของสัญญาณนาฬิกาของ X2 Mode	10
รูปที่ 2.8	แสดงโครงสร้างของการสร้าง PLL Clock	11
รูปที่ 2.9	แสดงโครงสร้างส่วนถอดรหัสเอ็มพีสาม	12
รูปที่ 2.10	แสดงบล็อกไดอะแกรมของส่วน MP3CLK	12
รูปที่ 2.11	แสดงการทำงานการอ่านข้อมูลจากการ์ดแบบ Stream	14
รูปที่ 2.12	แสดงการทำงานการอ่านข้อมูลจากการ์ด แบบ Multi-block (มี CRC7 ต่อท้ายแต่ละ Block)	14
รูปที่ 2.13	แสดงการทำงานการเขียนข้อมูลลงการ์ดแบบ Stream (หลังจากการเขียนข้อมูลเสร็จการ์ด จะทำการเขียนลง Flash จริงเป็น Busy)	14
รูปที่ 2.14	แสดงการทำงานการเขียนข้อมูลลงการ์ดแบบ Multi-block (หลังจากการเขียนข้อมูลเสร็จการ์ดจะทำการเขียนลง Flash จริงเป็น Busy และมี CRC7)	15
รูปที่ 2.15	แสดงรูปแบบบิตของข้อมูลคำสั่ง	15
รูปที่ 2.16	แสดงรูปแบบบิตของข้อมูลการตอบสนอง	16
รูปที่ 2.17	แสดงโครงสร้างของเฟรมข้อมูล	17
รูปที่ 2.18	แสดงการสร้างสัญญาณสำหรับเอ็มเอ็มซี	17
รูปที่ 2.19	แสดงส่วนของ Command Line Controller	18
รูปที่ 2.20	แสดงส่วนของ Data Line Controller	19
รูปที่ 2.21	แสดงบล็อกไดอะแกรมของการหาสัญญานาฬิกาส่วนส่งสัญญาณเสียงออก	21
รูปที่ 2.22	แสดงภาพรวมของส่วนส่งสัญญาณเสียงออก	22
รูปที่ 2.23	แสดงขาสัญญาณที่บอกว่าเป็นข้อมูลของลำโพงซ้ายหรือขวา	22
รูปที่ 2.24	แสดงการตั้งค่าตามหลักของ I^2S	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

		หน้าที่
รูปที่ 2.25	แสดงบล็อกไดอะแกรมภายในของของเอ็มเอ็มซี	24
รูปที่ 2.26	แสดงลักษณะการเชื่อมต่อของเอ็มเอ็มซี	26
รูปที่ 2.27	แสดงการส่งข้อมูลในเอ็มเอ็มซีแบบต่างๆ	28
รูปที่ 2.28	แสดงรูปแบบของ Operating Conditions Register (OCR)	29
รูปที่ 2.29	แสดงลักษณะของสายสัญญาณที่เชื่อมต่อระหว่างเอ็มเอ็มซีและโฮสท์ของเอ็มเอ็มซี	30
รูปที่ 2.30	แสดงบล็อกไดอะแกรมการทำงานของเอสพีไอ	31
รูปที่ 2.31	แสดงการสื่อสารแบบเอสพีไอ	36
รูปที่ 2.32	แสดงการด่อพ่วงสลาฟของการสื่อสารแบบเอสพีไอ	37
รูปที่ 2.33	แสดงตัวอย่างซอร์สโค้ดการทำเอสพีไอ	39
รูปที่ 2.34	แสดงการจัดลำดับการรับส่งข้อมูลของอุปกรณ์แต่ละตัว	40
รูปที่ 2.35	แสดงลำดับและขั้นตอนการทำงานของซอฟต์แวร์ควบคุมการทำงานของพอร์ตยูเอสบี	42
รูปที่ 2.36	แสดงไดอะแกรมแสดงการทำงานของรูตฮับอย่างง่าย	44
รูปที่ 2.37	แสดงลักษณะของตัวโมดูล TRW-2.4G	47
รูปที่ 2.38	แสดงไดอะแกรมการส่งข้อมูลในโหมด ShockBurst ของตัวโมดูล TRW-2.4G	48
รูปที่ 2.39	แสดงไดอะแกรมการรับข้อมูลในโหมด ShockBurst ของตัวโมดูล TRW-2.4G	49
รูปที่ 2.40	แสดงขาสัญญาณต่างๆ รวมถึงทิศทางการวิ่งของข้อมูลของโมดูล TRW-2.4G	51
รูปที่ 2.41	แสดงถึงโครงสร้างภายในของไอซีเสียงเบอร์ ISD2590	52
รูปที่ 2.42	แสดงถึงโครงสร้างภายขาสัญญาณของไอซีเสียงเบอร์ ISD2590	52
รูปที่ 2.43	แสดงถึงแผนผังเวลา (Timing Diagram) การบันทึกเสียงของไอซีเสียงเบอร์ ISD2590	53
รูปที่ 2.44	แสดงถึงแผนผังเวลาการเล่นเสียงของไอซีเสียงเบอร์ ISD2590	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

		หน้าที่
รูปที่ 3.1	แสดงภาพรวมของบล็อกไดอะแกรมทั้งหมด	57
รูปที่ 3.2	แสดงบล็อกไดอะแกรมของอุปกรณ์	58
รูปที่ 3.3	แสดงการออกบล็อกไดอะแกรมส่วนของไมโครคอนโทรลเลอร์ เบอร์ AT89C51SND1C	59
รูปที่ 3.4	แสดงการออกแบบผังวงจรส่วนของไมโครคอนโทรลเลอร์ เบอร์ AT89C51SND1C	60
รูปที่ 3.5	แสดงการออกแบบบล็อกไดอะแกรมของส่วนภาคจ่ายไฟ	60
รูปที่ 3.6	แสดงการออกแบบผังวงจรส่วนของภาคจ่ายไฟ	61
รูปที่ 3.7	แสดงการออกแบบบล็อกไดอะแกรมของยูเอสบี	61
รูปที่ 3.8	แสดงการออกแบบผังวงจรของยูเอสบี	61
รูปที่ 3.9	แสดงการออกแบบบล็อกไดอะแกรมของส่วนเอ็มเอ็มซี	62
รูปที่ 3.10	แสดงการออกแบบผังวงจรของส่วนเอ็มเอ็มซี	62
รูปที่ 3.11	แสดงการออกแบบบล็อกไดอะแกรมของส่วนแอลซีดี	63
รูปที่ 3.12	แสดงการออกแบบผังวงจรของส่วนแอลซีดี	63
รูปที่ 3.13	แสดงการออกแบบบล็อกไดอะแกรมของส่วนหูฟัง	63
รูปที่ 3.14	แสดงการออกแบบผังวงจรของส่วนหูฟัง	64
รูปที่ 3.15	แสดงการออกแบบส่วนของ ISD2590	65
รูปที่ 3.16	แสดงส่วนการออกแบบของส่วนส่งข้อมูลของสถานีสัญญาณ	66
รูปที่ 3.17	แสดงการออกแบบผังวงจรของส่วนส่งข้อมูลของสถานีสัญญาณ	66
รูปที่ 3.18	แสดงการออกแบบบล็อกไดอะแกรมของส่วนรับข้อมูลของอุปกรณ์พกพา	67
รูปที่ 3.19	แสดงการออกแบบผังวงจรของส่วนตัวรับข้อมูลของอุปกรณ์พกพา	67
รูปที่ 3.20	แสดงการออกแบบลายวงจรของภาคการส่งและรับข้อมูล	68
รูปที่ 3.21	แสดงสถานการณ์การทำงานของโปรแกรม	69
รูปที่ 3.22	แสดงสถานการณ์ทำงานของส่วนอุปกรณ์พกพา	70
รูปที่ 3.23	แสดงผังการทำงานของโปรแกรมส่วนอุปกรณ์พกพา	73
รูปที่ 3.24	แสดงผังการทำงานของโปรแกรมส่วนสถานีสัญญาณ	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

		หน้าที่
รูปที่ 4.1	แสดงโปรแกรมที่ใช้ทำการทดลองของส่วนไฟล์เฮดเดอร์ AT89C51SND1.h	76
รูปที่ 4.2	แสดงผลการทดลองส่วนไฟล์เฮดเดอร์ AT89C51SND1.h	76
รูปที่ 4.3	แสดงโปรแกรมที่ใช้ทำการทดลองของส่วนจอแอลซีดี	77
รูปที่ 4.4	แสดงผลการทดลองของส่วนจอแอลซีดีจากโปรแกรมจำลอง	78
รูปที่ 4.5	แสดงผลการทดลองของส่วนไอสแควร์ซี	79
รูปที่ 4.6	แสดงผลการทดลองของส่วนไอสแควร์ซีจากโปรแกรมจำลองการทำงาน	79
รูปที่ 4.7	แสดงส่วนฟังก์ชันหลักที่ใช้ในการเขียนโปรแกรมสำหรับตัวส่ง	83
รูปที่ 4.8	แสดงส่วนฟังก์ชันหลักที่ใช้ในการเขียนโปรแกรมสำหรับตัวรับ	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันอุตสาหกรรมการท่องเที่ยวนั้นมีความสำคัญกับประเทศไทยเป็นอย่างมาก ทำรายได้เข้าประเทศได้ปีละมากๆ ในปัจจุบันปี 2549 คาดว่าจะมีนักท่องเที่ยวต่างประเทศมาท่องเที่ยวในประเทศไทยประมาณ 15.12 ล้านคน โดยเฉพาะการท่องเที่ยวชมโบราณสถานหรือพิพิธภัณฑ์ที่สวยงามของประเทศไทย จะเกิดปัญหาขึ้นกับนักท่องเที่ยวเป็นอย่างมากเนื่องมาจากการสื่อสารด้วยภาษาที่แตกต่างกัน ทำให้นักท่องเที่ยวไม่สามารถเข้าใจถึงประวัติความเป็นมาของแหล่งท่องเที่ยวนั้นๆได้อย่างถ่องแท้ หรือ นักท่องเที่ยวอาจไม่เข้าใจถึงกฎหรือข้อบังคับต่างๆของสถานที่นั้นๆ ทำให้เกิดการกระทำที่ไม่เหมาะสมต่อสถานที่หรือเป็นการลบหลู่ได้ เช่น ขึ้นไปยืนบนองค์พระ เพื่อถ่ายรูป ปีนป่ายหรือจับต้องของที่คนไทยเคารพ ทำให้เรามักไม่พอใจในเรื่องแบบนี้ แต่ปัญหาคือความไม่รู้ของเขา ของบางสิ่งในสถานที่ท่องเที่ยวมีความสำคัญมากกับเรา แต่เขาไม่รู้ความเป็นมาเขาก็เห็นเป็นแค่หินสลัก ปูนเก่าๆ บางคนอยากได้ของเก่าเหล่านี้ไปสะสมดูเล่น นำกลับไป ซื้อกลับไป เพราะเขาไม่รู้ค่าความสำคัญของของสิ่งนั้นต่อคนไทย

นอกจากนี้นักท่องเที่ยวบางคนชอบที่จะท่องเที่ยวคนเดียวไม่ชอบไปเป็นแบบทัวร์ที่เดินตามๆกันเป็นแถว ซึ่งการท่องเที่ยวแบบคนเดียวนี้นักท่องเที่ยวอาจไม่ได้รับข้อมูลที่ถูกต้องเพราะเค้าจะไม่มีมัคคุเทศก์ช่วยแนะนำสถานที่ต่างๆ อาจทำให้นักท่องเที่ยวชาวต่างชาติเหล่านี้ ไม่ได้รับความประทับใจในขณะที่ท่องเที่ยวเท่าที่ควรจะเป็น เพราะติดปัญหาด้านการสื่อสาร พูดจา ซึ่งเป็นปัญหาที่คนไทยไม่กล้าที่จะพูด กับไม่มีความรู้ความมั่นใจพอที่จะพูดให้ข้อมูลหรือให้ความช่วยเหลือนักท่องเที่ยวเพียงพอจนอาจทำให้นักท่องเที่ยวนั้นไม่มาเที่ยวที่ประเทศไทยอีก

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อเป็นหนึ่งความร่วมมือในการส่งเสริมการท่องเที่ยว

1.2.2 เพื่อศึกษาการทำงานไมโครคอนโทรลเลอร์ และนำไปประยุกค์ใช้ได้

1.2.3 เพื่อศึกษาทฤษฎีทางด้านคลื่นวิทยุ (RF), In-system Programming, การเชื่อมต่อภายนอกแบบเข้าจังหวะหรือเอสพีไอ (SPI) และเอ็มพีสาม

1.2.4 สร้างระบบของการนำเที่ยวในสถานที่ท่องเที่ยวหรือพิพิธภัณฑ์ต่างๆ ที่ผู้ท่องเที่ยวสามารถท่องเที่ยวด้วยตนเองได้ โดยไม่ต้องอาศัยมัคคุเทศก์เป็นผู้นำเดินชมสถานนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ได้รับความรู้ ความเข้าใจเกี่ยวกับงานทางด้านไมโครคอนโทรลเลอร์ และอุปกรณ์อิเล็กทรอนิกส์ ที่เกี่ยวข้อง

1.3.2 ได้รับความรู้ ความเข้าใจเกี่ยวกับคลื่นวิทยุ (RF), In-system Programming, เอสพีไอ และเอ็มพีสาม

1.3.3 สามารถออกแบบและสร้างตัวต้นแบบอุปกรณ์ มัคคุเทศก์อัจฉริยะ

1.3.4 ได้ฝึกฝนการวิเคราะห์และการแก้ปัญหาที่เกิดขึ้นในการทำงานวิจัย

1.3.5 แนวทางการแก้ปัญหา แผนการใช้งานอุปกรณ์เพื่อเป็นประโยชน์ต่อการท่องเที่ยว

1.4 ขอบเขตของงานวิจัย

ศึกษา ออกแบบและพัฒนาอุปกรณ์เพื่อใช้แทนมัคคุเทศก์ โดยตัวอุปกรณ์มีความสามารถเล่นไฟล์เสียงคำอธิบายสถานที่หรือห้องจัดแสดงในพิพิธภัณฑ์ ที่บุคคลผู้นั้นชมอยู่ โดยจะจัดทำ

1. สถานีสัญญาณ (iGUIDE Portable) เป็นอุปกรณ์ที่จะติดไว้ตามสถานที่ ห้อง หรือ งานเสดงต่างๆ ซึ่งตัวสถานีนี้จะใช้การส่งโดยโมดูลส่งสัญญาณคลื่นวิทยุซึ่งจะทำการส่งไอดีประจำสถานีนี้เพื่อระบุพื้นที่นั้นๆแบบทิศทางเดียวเท่านั้น

2. อุปกรณ์พกพา (iGUIDE Station) เป็นอุปกรณ์ที่พกพาไว้กับนักท่องเที่ยวที่จะเป็นตัวเล่นไฟล์เสียงคำอธิบายสถานที่ โดยในตัวโปรโตไทป์นี้จะใช้ไอซีเสียงเก็บไฟล์เสียงสั้นๆบอกแค่ว่าสถานีนี้คือสถานีอะไรหรือไม่ได้อยู่ในพื้นที่ของสถานี เมื่อนำอุปกรณ์พกพานี้เข้าไปยังพื้นที่ของสถานีใดๆก็จะมีการเล่นไฟล์เสียงออกทางหูฟังบ่งบอกว่าอยู่ในสถานีใด

1.5 ส่วนประกอบของรายงาน

รายงานฉบับนี้มีอยู่ 5 บทมีเนื้อหาดังนี้

เนื้อหาในบทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงงาน และส่วนประกอบของรายงานฉบับนี้

เนื้อหาในบทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วยทฤษฎีหลายๆทฤษฎีดังนี้คือ

1. ส่วนของไมโครคอนโทรลเลอร์ AT89C51SND1Cซึ่งประกอบด้วยส่วนโครงสร้างต่างๆ เช่น ส่วนของโครงสร้างของขาสัญญาณภายนอกซึ่งประกอบด้วยขาสัญญาณจำนวน 80 ขา ส่วนของโครงสร้างภาพรวมภายใน ซึ่งประกอบด้วยส่วนต่างๆทั้งส่วนที่เป็น C51 ส่วนของการกำเนิดสัญญาณนาฬิกา ส่วนอินเทอร์รัป ส่วนของหน่วยความจำภายในหรือ RAM ส่วนหน่วยความจำแบบแฟลชทั้ง flash ROMและ flash boot ส่วน UARTใช้จัดการการติดต่อแบบอนุกรม ส่วน Timer/Watchdog ซึ่งเป็นส่วนจัดการเกี่ยวกับวงจรนับภายในคอนโทรลเลอร์ ส่วน SPI/Data flash, TWI, ยูเอสบี, การติดต่อเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็มซี (MMC), คีย์บอร์ด, พอร์ตอินพุต/เอาท์พุต ซึ่งเป็นส่วนที่ใช้ติดต่ออุปกรณ์ภายนอก ส่วนถอดรหัสเอ็มพีสามใช้ถอดรหัสไฟล์เอ็มพีสามและส่วนสุดท้ายคือส่วนการส่งสัญญาณเสียงออก (Audio) เป็นส่วนที่ส่งสัญญาณเสียงออกไปภายนอก

2. ส่วนของการใช้งานเอ็มเอ็มซี ซึ่งจะกล่าวถึงความรู้ทั่วๆไปเกี่ยวกับเอ็มเอ็มซีและลักษณะการเชื่อมต่อและโปรโตคอลที่ใช้อย่างละเอียด และยังมีส่วนของคำสั่งและการทำงานของเอ็มเอ็มซี

3. การเชื่อมต่อแบบไอสแควร์ซีแบบสองเส้น (Two-wire Interface I^2C)

4. การเชื่อมต่อภายนอกแบบเข้าจังหวะ (SPI : Synchronous Peripheral Interface)

5. การเชื่อมต่อแบบยูเอสบี (Universal Serial Bus Interface)

6. การใช้งานส่วนการติดต่อผ่านคลื่นสัญญาณวิทยุด้วยโมดูล TRW-2.4G

7. การใช้งานไอซีเสียงเบอร์ ISD2590

เนื้อหาในบทที่ 3 จะกล่าวถึงชิ้นงานของโครงงานนี้ ส่วนที่ได้พัฒนาขึ้น โดยแบ่งเป็นส่วนภาพรวมของการออกแบบโดยจะต้องกำหนดคุณลักษณะการใช้งานก่อนและจากนั้นจะทำการเลือกเทคโนโลยีว่าเทคโนโลยีไหนจะเหมาะสมกับชิ้นงานทั้งทางด้านราคาและประสิทธิภาพ จากนั้นต้องทำการออกแบบส่วนของบล็อกไดอะแกรมเพื่อให้ได้เห็นภาพรวมของระบบทั้งหมด จากนั้นเมื่อได้ข้อสรุปแล้วว่าเราจะใช้เทคโนโลยีอะไร ภาพรวมของระบบเป็นอย่างไรแล้ว จะเริ่มทำการออกแบบส่วนของฮาร์ดแวร์โดยเริ่มต้นทำการออกแบบผังวงจร (Schematic design) ของทั้งระบบก่อน จากนั้นจะทำการออกแบบลายวงจร (Print Circuit Board) เป็นส่วนๆไป สุดท้ายเราจะทำการเขียนโปรแกรมเพื่อควบคุมฮาร์ดแวร์ (Firmware) โดยการออกแบบเขียนโปรแกรมนั้นจะต้องเริ่มจากการออกแบบสถานะทำงาน (State Diagram) เพื่อให้เห็นภาพรวมของการทำงานของฮาร์ดแวร์ก่อน จากนั้นจะทำการออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) ทำการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AT89C51SND1C เพื่อให้ไมโครคอนโทรลเลอร์ทำงานกับส่วนต่างๆได้อย่างถูกต้อง และทำส่วนของการติดต่อผ่านคลื่นสัญญาณวิทยุด้วยโมดูล TRW-2.4G กับการใช้งานไอซีเสียงเบอร์ ISD2590เป็นส่วนสุดท้าย

เนื้อหาในบทที่ 4 กล่าวถึงการทดลอง ผลการทดลอง ปัญหาที่เกิดจากการทดลอง และวิธีการแก้ปัญหานั้นๆ

เนื้อหาในบทที่ 5 เป็นบทสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน ปัญหาอุปสรรคและแนวทางแก้ไข และแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

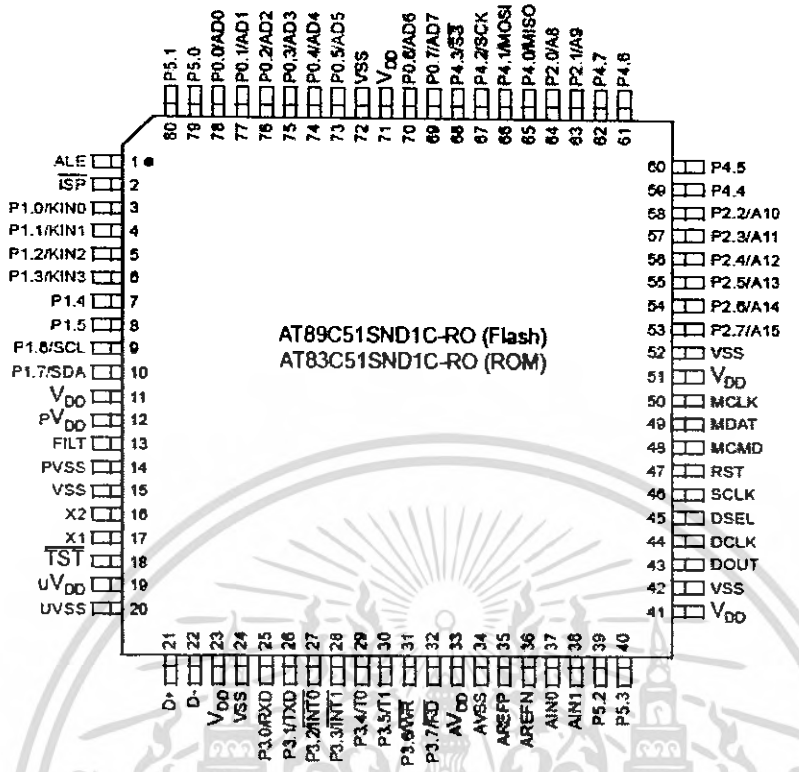
2.1 โครงสร้างของไมโครคอนโทรลเลอร์ AT89C51SND1C

คุณสมบัติของไมโครคอนโทรลเลอร์ AT89C51SND1C นี้มีความสามารถ ที่น่าสนใจอย่างยิ่ง และอำนวยความสะดวกต่อการพัฒนาเป็นอุปกรณ์เครื่องเล่นเอ็มพีสาม เพราะมีความสามารถในการถอดรหัสข้อมูลเสียงในรูปแบบเล่นเอ็มพีสาม ให้เป็นข้อมูลเสียงแบบสเตอริโอตามรูปแบบไอสแควร์เอส (I^2S) หรือพีซีเอ็ม (PCM) ได้ และยังมีส่วนจัดการการติดต่อกับอุปกรณ์ภายนอกหลากหลายชนิด เช่น สามารถเชื่อมต่อกับคอมพิวเตอร์ผ่านยูเอสบี1.1 (USB1.1) แบบเต็มความเร็ว (Full Speed) หรือ ผ่านพอร์ตอนุกรมก็ได้ สามารถเชื่อมต่อกับหน่วยความจำภายนอกได้ เช่น การ์ดหน่วยความจำเอสดีการ์ดหรือเอ็มเอ็มซี, ซีดีรอม หรือ ฮาร์ดดิสก์ ผ่านโปรโตคอลเอสพีไอ(SPI Protocol) และสื่อสารกับไอซีต่อพ่วงอื่นๆ ผ่านทางโปรโตคอลไอสแควร์ซี (I^2C Protocol) ได้อีกด้วย นับเป็นไมโครคอนโทรลเลอร์ที่มีประโยชน์ในการเรียนรู้ และควรนำมาประยุกต์ใช้งานอย่างยิ่ง

2.1.1 โครงสร้างขาสัญญาณภายนอก

ไมโครคอนโทรลเลอร์นี้ประกอบด้วย ขาสัญญาณภายนอกจำนวน 80 ขา ในผลิตภัณฑ์แบบ TQFP ประกอบด้วย 5 พอร์ต ครึ่ง คือ มี P0-P4 จำนวน พอร์ต ละ 8 บิต และ P5 มีขนาดเพียง 4 บิต และมีขาเชื่อมต่อโปรโตคอลภายนอกต่างๆ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงขาสัญญาณภายนอกของไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C

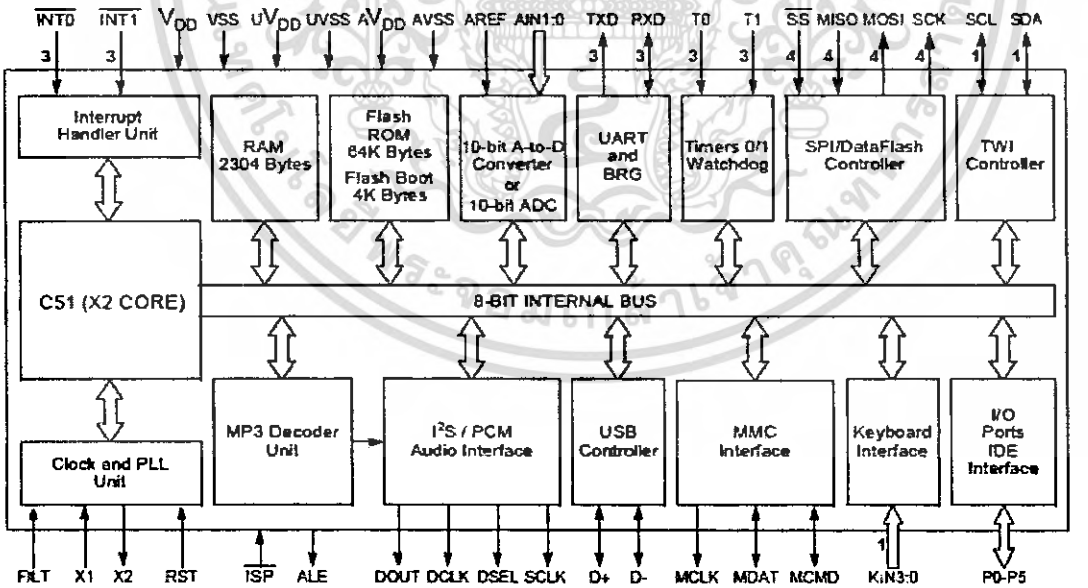
ขาสัญญาณของไมโครคอนโทรลเลอร์ AT89C51SND1C แบ่งเป็นกลุ่มๆตามการใช้งานได้ดังนี้

- <u>กลุ่มขาสัญญาณพลังงาน</u> : ไฟเลี้ยง 3.3 V VDD, uVDD(ส่วนยูเอสบี), aVDD(ส่วนอนาลอก), pVDD(ส่วนจัดการ PLL) และ Ground 0 V VSS, uVSS(ส่วนยูเอสบี), aVSS(ส่วนอนาลอก), pVSS(ส่วนจัดการ PLL)
- <u>กลุ่มพอร์ตอินพุต-เอาท์พุตอเนกประสงค์</u> : P0-P4 ขนาด 8 บิต และ P5 ขนาด 4 บิต
- <u>กลุ่มสัญญาณนาฬิกาและการควบคุมไมโครคอนโทรลเลอร์</u> : X1,X2 ต่อกับคริสตอลขนาดไม่เกิน 20 เมกะเฮิร์ทซ์ ,FILT ขาสัญญาณต่อวงจร Filter ภายนอก, ISP ขาเลือกให้ไมโครคอนโทรลเลอร์ทำงานหรือโปรแกรมซอฟแวร์ลงในไมโครคอนโทรลเลอร์, RST ขาเลือกว่าให้ไมโครคอนโทรลเลอร์เริ่มการทำงานใหม่, TST ขาเลือกให้ไมโครคอนโทรลเลอร์ทำงานในโหมดทดสอบหรือไม่ ปกติใช้งานเลือกเป็นโลจิก 0
- <u>กลุ่มการอ้างหน่วยความจำภายนอก</u> : ALE ขาสัญญาณสั่งให้ทำงาน Address Latch ,WR สั่งให้หน่วยความจำเขียนข้อมูล, RD อ่านข้อมูลจากหน่วยความจำ P0,P2 ที่เป็น AD0-AD15 ใช้ในการอ้างอิงแอดเดรสขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- <u>กลุ่มการติดต่อยูเอสบี</u> : D+ และ D- ขาที่ใช้ในการสื่อสารกับคอมพิวเตอร์
- <u>กลุ่มการติดต่ออนุกรม</u> : RX และ TX ขาที่ใช้ในการสื่อสารกับคอมพิวเตอร์
- <u>กลุ่มการติดต่ออุปกรณ์ภายนอกด้วยเอสพีไอ</u> : MOSI, MISO, SS, SCK เป็นขาที่ใช้ในการสื่อสาร
- <u>กลุ่มการติดต่ออุปกรณ์ภายนอกด้วยไอสแควร์ซี</u> : SDA, SCL เป็นขาที่ใช้ในการสื่อสาร
- <u>กลุ่มการติดต่อเอ็มเอ็มซีด้วยเอ็มเอ็มซีโปรโตคอล</u> : MDAT, MCLK, MCMD เป็นขาที่ใช้ในการสื่อสาร
- <u>กลุ่มการติดต่ออุปกรณ์แปลงสัญญาณดิจิตอลเป็นอะนาล็อกเสียงแบบไอสแควร์เอส หรือพีซีเอ็ม (PCM)</u> : DOUT, DCLK, DSEL, SCLK เป็นขาที่ใช้ในการสื่อสาร

2.1.2 โครงสร้างภาพรวมภายใน



รูปที่ 2.2 แสดงโครงสร้างภาพรวมภายในของไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน C51 เป็นส่วนควบคุมการทำงานของไมโครคอนโทรลเลอร์ ให้ทำงานตามซอฟแวร์ที่โปรแกรมเข้าไป จะทำงานได้ต้องมีสัญญาณนาฬิกาจากส่วน Clock-PLL ทำงานร่วมกับอินเทอร์รัป (Interrupt) และข้อมูลที่ประมวลผลได้จะเข้าออกทางบัสภายใน

ส่วน Clock-PLL เป็นส่วนที่จัดการด้านสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์ โดยรับสัญญาณภายนอกจากคริสตอล

ส่วน อินเทอร์รัป จะคอยจัดการสัญญาณอินเทอร์รัป ที่เข้ามาขัดจังหวะการทำงานของ C51 โดยจะจัดตามลำดับความสำคัญ

ส่วน RAM หน่วยความจำภายใน เก็บข้อมูลระหว่างการทำงานของไมโครคอนโทรลเลอร์

ส่วน Flash ROM และ Flash Boot เป็นส่วนที่ใช้เก็บโปรแกรมการทำงานของไมโครคอนโทรลเลอร์ โดย Flash ROM เก็บซอฟแวร์ แต่ Flash Boot เป็นโปรแกรมที่ทำให้สามารถทำ In-System Programming ได้

ส่วน UART ใช้สำหรับจัดการเมื่อไมโครคอนโทรลเลอร์ต้องการติดต่อผ่านพอร์ตอนุกรม และใช้เมื่อมีการโปรแกรมผ่านพอร์ตอนุกรมด้วย

ส่วน Timer/Watchdog เป็นส่วนจัดการวงจรนับภายในไมโครคอนโทรลเลอร์ หรือตั้งค่าให้นับสัญญาณจากภายนอกได้

ส่วน SPI/Data flash, TWI (Two-wire Interface), ยูเอสบี, MMC Interface, Keyboard, พอร์ต I/O เป็นส่วนที่ใช้ในการติดต่ออุปกรณ์ภายนอก

ส่วน ส่วนถอดรหัสเอ็มพีสาม เป็นวงจรในการถอดรหัสไฟล์เอ็มพีสามโดยรับข้อมูลจาก Internal Bus แล้วป้อนข้อมูลเสียงที่ได้ออกไปที่ ส่วน Audio Interface ให้จัดการให้

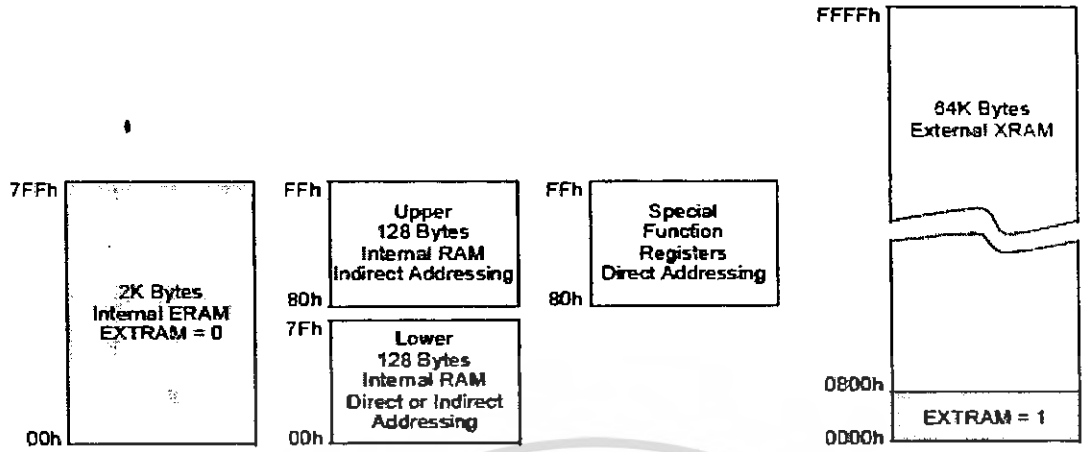
ส่วน Audio Interface เป็นวงจรสร้างสัญญาณดิจิตอล ตามรูปแบบของไอสแควร์เอสหรือพีซีเอ็มตามการตั้งค่า และนำข้อมูลที่แปลงได้ส่งออกไปภายนอก โดยสามารถรับข้อมูลเสียงที่ถอดรหัสแล้วได้ทั้งจากส่วนถอดรหัสเอ็มพีสามหรือ C51 ก็ได้

2.1.3 โครงสร้างหน่วยความจำ

การจัดหน่วยความจำในไมโครคอนโทรลเลอร์ AT89C51SND1C แบ่งออกเป็น 2 ชนิด คือ

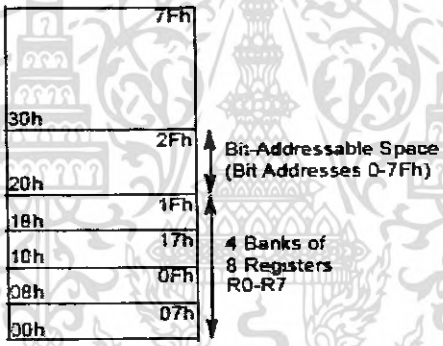
- *หน่วยความจำภายใน* มี 3 ชนิด Lower RAM 128 ไบต์, Upper RAM 128 ไบต์ และ Extended RAM (ERAM) อีก 2 กิโลไบต์
- *หน่วยความจำภายนอก* สามารถอ้างหน่วยความจำภายนอกได้ 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.3 แสดงโครงสร้างของหน่วยความจำ

Lower RAM มีขนาด 128 ไบต์ สามารถอ้างได้ทั้งแบบโดยตรงและโดยอ้อม มีหมายเลขแอดเดรสในช่วง 00h-7Fh โดยภายในแบ่งออกเป็นส่วนย่อยเพื่อจุดประสงค์เฉพาะทางอีกดังรูป



รูปที่ 2.4 แสดง RAM ส่วนล่าง

มี รีจิสเตอร์ ทั่วไปให้ใช้งาน 4 แถว แถวละ 8 ไบต์ โดยสามารถเลือกใช้งานได้โดยตั้งค่าที่ RS1 กับ RS0 นั่นคือ '00' ใช้ แถวที่ 0 (00h-07h), '01' ใช้ แถวที่ 1 (08h-0Fh), '10' ใช้ แถวที่ 2 (10h-17h), '11' ใช้ แถวที่ 3 (18h-1Fh) และพื้นที่ตั้งแต่ 20h-2Fh เรียกว่า Bit-Field เพราะสามารถอ้างตัวแปรที่เป็นแบบบิตได้ และส่วนที่เหลือใช้งานทั่วไปแบบไบต์ 30h-7Fh

Upper RAM 128 bit จะถูกซ้อนทับด้วย SFR ทำให้สามารถใช้งานได้แต่ต้องอ้าง แอดเดรสแบบโดยอ้อมเท่านั้น มีหมายเลข แอดเดรสตั้งแต่ 80h-FFh

SFR เป็น รีจิสเตอร์ เฉพาะที่ใช้งานเพื่อตั้งค่าและใช้งาน I/O ต่างๆ ของคอนโทรลเลอร์ แต่ซ้อนทับอยู่กับ Upper RAM จึงต้องอ้างแบบโดยตรงเท่านั้น หมายเลข แอดเดรส เหมือนกับ Upper RAM

Extended RAM (ERAM) เป็น RAM ขนาด 2 กิโลไบต์ สามารถกำหนดขนาดของ ERAM ได้ตามต้องการตามตารางนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดง Extended RAM (ERAM)

XRS1	XRS0	ERAM Size	Address
0	0	256 Bytes	0 to 00FFh
0	1	512 Bytes	0 to 01FFh
1	0	1K Byte	0 to 03FFh
1	1	2K Bytes	0 to 07FFh

การใช้งาน ERAM ต้องกำหนดขนาดก่อนแล้วใช้ DPTR อ้างอิง แอดเดรส 16 บิต แต่ต้องตั้งค่า EXTRAM = 0 เพราะว่าอ้าง DPTR โดยปกติหมายถึง แอดเดรส ภายนอก แต่ถ้ายังต้องการอ้างแอดเดรส ภายนอกก็ต้อง ให้ EXTRAM = 1

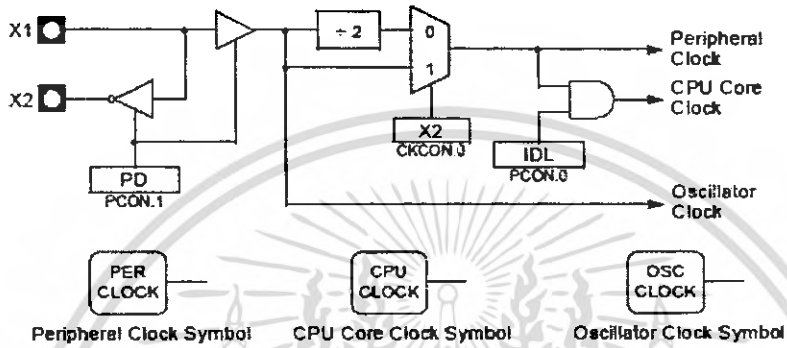
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	UEPINT 0000 0000	DAT16H XXXX XXXX		NVERS XXXX XXXX[2]					FFh
F0h	B[1] 0000 0000		ADCLK 0000 0000	ADCON 0000 0000	ADDL 0000 0000	ADDH 0000 0000			F7h
E8h		PLLCON 0000 1000	USBCLK 0000 0000	MP3CLK 0000 0000	AUDCLK 0000 0000	MMCLK 0000 0000	PLLNDIV 0000 0000	PLLRDIV 0000 0000	EFh
E0h	ACC[1] 0000 0000		UBYCTLX 0000 0000		MMCON0 0000 0000	MMCON1 0000 0000	MMCON2 0000 0000	MMINT 0000 0011	E7h
D8h	P5[1] XXXX 1111				MMDAT 1111 1111	MMCMD 1111 1111	MMSTA 0000 0000	MMMSK 1111 1111	DFh
D0h	PSW[1] 0000 0000	FCON[2] 1111 0000[4]			UEPCONX 1000 0000	UEPRST 0000 0000			D7h
C8h	MP3STA[2] 0000 0001						UEPSTAX 0000 0000	UEPDATX XXXX XXXX	CFh
C0h	P4[1] 1111 1111		UEPIEN 0000 0000	SPCON 0001 0100	SPSTA 0000 0000	SPDAT XXXX XXXX	USBADOR 0000 0000	UEPNUM 0000 0000	C7h
B8h	IPL0[1] X000 0000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEN 0001 0000		BFh
B0h	P3[1] 1111 1111	IEN1 0000 0000	IPL1 0000 0000	IPH1 0000 0000	MP3BAS 0000 0000	MP3MED 0000 0000	MP3TRE 0000 0000	IPH0 X000 0000	B7h
A8h	IEN0[1] 0000 0000	SADOR 0000 0000	MP3CON 0011 1111		MP3DAT 0000 0000	MP3ANC 0000 0000		MP3STA1 0100 0001	AFh
A0h	P2[1] 1111 1111		AUXR1 XXXX 00X0	KBCON 0000 1111	KBSTA 0000 0000		WDTRST XXX XXXX	WDTPRG XXXX X000	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	AUDCON0 0000 1000	AUDCON1 1011 0010	AUDSTA 1100 0000	AUDDAT 1111 1111	MP3VOL 0000 0000	MP3VOR 0000 0000	9Fh
90h	AUXCON[3] 1111 1111	BRL 0000 0000	BDRCON XXX0 0000	SSCON 0000 0000	SSSTA 1111 1000	SSDAT 1111 1111	SSADR 1111 1110		97h
88h	TCON[2] 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR X000 1101	CKCON 0000 000X[5]	8Fh
80h	P0[1] 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00XX 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

รูปที่ 2.5 แสดง SFR ของ AT89C51SND1C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

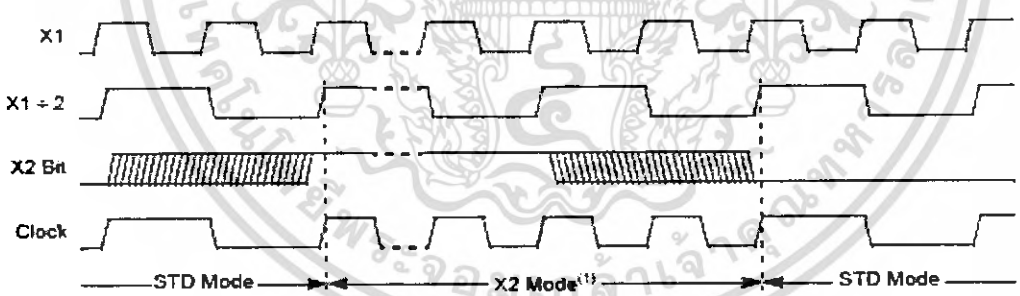
2.1.4 โครงสร้างระบบสัญญาณนาฬิกา

ภายในของไมโครคอนโทรลเลอร์ AT89C51SND1C มีการจัดการ PLL Phase Lock Loop เพื่อสร้างระบบสัญญาณนาฬิกาไปป้อนให้ระบบทั้งหมดภายในหลายส่วนทำงานได้ โดยจะรับสัญญาณเข้ามาจากคริสตอลทางขาสัญญาณ X1, X2 เพื่อสร้างสัญญาณนาฬิกาหลักออกมา 3 ชนิด คือ สัญญาณ PLL Clock, สัญญาณ CPU Clock และ Peripheral Clock ซึ่งจะใช้สร้างสัญญาณในส่วนของไทม์เมอร์, เอ็มเอ็มซี และเอสพีไอ



รูปที่ 2.6 แสดงโครงสร้างของระบบสัญญาณนาฬิกา

AT89C51SND1C มีความสามารถแตกต่างจาก 51 รุ่นมาตรฐานตรงที่มีคุณสมบัติ X2 นั่นคือสามารถ เพิ่มความถี่ให้กับ การต่อสัญญาณนาฬิกาให้กับอุปกรณ์ภายนอก และสัญญาณนาฬิกาของซีพียูจากที่รุ่นมาตรฐานที่ทำงาน 1 แมชชีนไซเคิล ใช้สัญญาณนาฬิกาOSC 12 ลูก แต่หากใช้คุณสมบัติ X2 จะเหลือเพียง สัญญาณนาฬิกา 6 ลูก = 1 แมชชีนไซเคิล



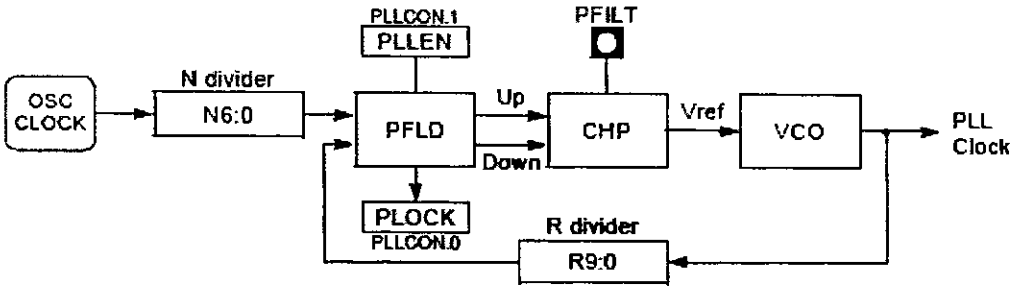
รูปที่2.7 แสดงการทำงานของสัญญาณนาฬิกาของ X2 Mode

PLL Clock เป็นสัญญาณนาฬิกาที่ป้อนให้กับส่วนถอดรหัสเอ็มพีสาม, ยูเอสบี, ส่วนส่งสัญญาณเสียงออก มีการคำนวณ ดังนี้

$$PLLclk = \frac{OSCclk \times (R+1)}{N+1} \qquad (2.1)$$

(2.1) สมการการคำนวณ PLL Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงโครงสร้างของการสร้าง PLL Clock

การตั้งค่าให้ PLL ทำงานได้มีขั้นตอนดังนี้

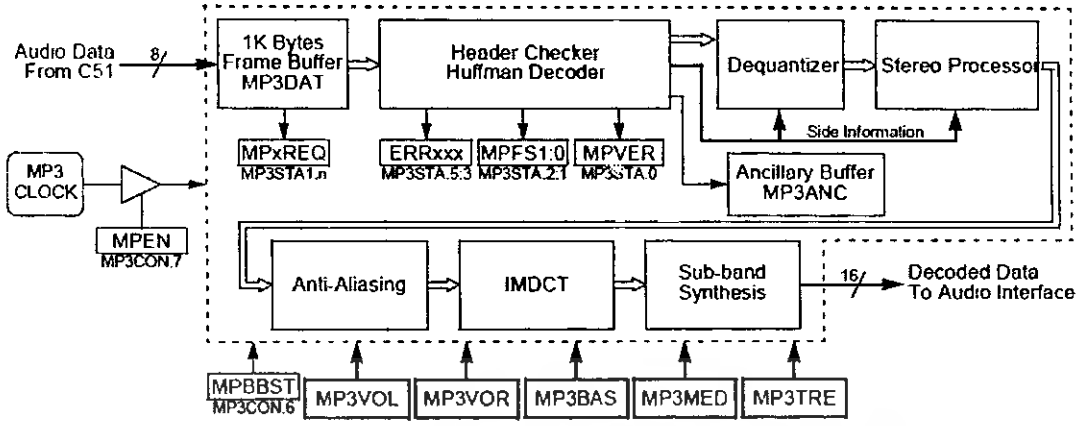
1. ตั้งค่า N6:0 และ R9:0 ให้เป็นค่าที่ต้องการ
2. เคลียร์ ค่าบิต PLLRES เพื่อยกเลิกการ รีเซต
3. เซต ค่าบิต PLLEN เพื่อเปิดการทำงาน PLL
4. ตัวไมโครคอนโทรลเลอร์เองจะวนอ่าน PLOCK บิตว่าเท่ากับ '1' หรือยังถ้ายังก็รอจนกว่าจะเป็น 1 ได้ถึงจะเริ่มทำงานได้

2.1.5 โครงสร้างส่วนถอดรหัสเอ็มพีสาม (MP3 Decoder)

2.1.5.1 บล็อกไดอะแกรมของส่วนถอดรหัสเอ็มพีสาม (MP3 Decoder Block Diagram)

กระบวนการทำงานจากข้อมูลเอ็มพีสามมีดังนี้ เขียนข้อมูลลงบน MP3DAT ซึ่งเป็นบัฟเฟอร์ขนาด 1 กิโลไบต์ แล้วไมโครคอนโทรลเลอร์จะผ่านเข้าไปที่ ส่วนตรวจสอบเฮดเดอร์และการถอดรหัสแบบฮาฟฟ์แมน (Haffman Decode) ถ้ามีข้อผิดพลาดก็จะบอกออกมาที่แฟลกต่างๆ หากมีค่าอื่นๆอีก จะนำค่านั้นออกมาที่ MP3ANC ที่สามารถอ่านได้อย่างเดียว แล้วก็นำไปผ่านกระบวนการด้านเสียง เช่น ปรับเสียงทั้งซ้ายและขวา ปรับแต่เบส, เทรบเบิล ตามต้องการสามารถเลือกที่จะใช้การปรับนี้หรือไม่ ไม่ใช้ MPBBST = '0' หากใช้ MPBBST = '1' แล้วข้อมูลเสียงจะออกมาในรูปของข้อมูลดิจิตอลแบบไอสแควร์เอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงโครงสร้างส่วนถอดรหัสเอ็มพีสาม

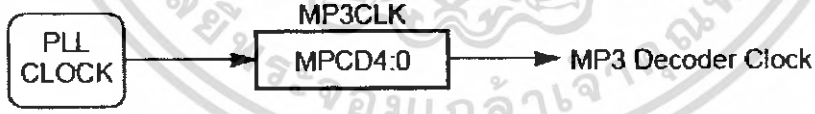
2.1.5.2 สัญญาณนาฬิกาเอ็มพีสาม(MP3 Clock)

ในส่วนนี้ต้องมีสัญญาณนาฬิกาเฉพาะส่วน แต่จะนำมาจาก PLL Clock แล้วนำมาใช้งาน สามารถคำนวณได้ดังนี้

$$f_{MP3Clock} = \frac{f_{PLLClock}}{MP3CLK + 1} \quad (2.2)$$

(2.2) สมการการคำนวณสัญญาณนาฬิกาของเอ็มพีสาม

โดย MP3CLK เป็นรีจิสเตอร์ขนาด 5 บิต จึงสามารถปรับได้ 32 ระดับ ตั้งแต่ 00000 – 11111 (0-32) และที่ควรรู้คือ MPEG 1 ต้องมีความถี่อย่างน้อย 21 เมกะเฮิร์ทซ์ และ MPEG II ต้องมีความถี่อย่างน้อย 10.5 เมกะเฮิร์ทซ์



รูปที่ 2.10 แสดงบล็อกไดอะแกรมของส่วน MP3CLK

2.1.5.3 การควบคุมความดังเบาเสียง (Volume Control)

MP3VOR และ MP3VOL คือการปรับความดังได้ ประมาณ 32 ระดับ เพราะแต่ละรีจิสเตอร์ ประกอบด้วย 5 บิต และ '00000' คือ Mute

2.1.5.4 การควบคุมอีควอไลซ์เซอร์ (Equalization Control)

ประกอบด้วย 3 รีจิสเตอร์ คือ MP3BAS ปรับเบส ได้ 32 ระดับ, MP3MED ปรับ Medium ได้ 32 ระดับ และ MP3TRE ปรับเทรบเบิล ได้ 32 ระดับ โดย '00000' = -infinity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.5 รูปแบบเฟรม (Frame Information)

เป็นข้อมูลที่ได้จาก MP3 Header โดยจะคืนค่ามาที่ MPVER, MPFS0:1 บอกข้อมูลเฉพาะของไฟล์ นั้นๆ ดังตารางนี้

ตารางที่2.2 แสดงข้อมูลของเฟรม

MPVER	MPFS1	MPFS0	Fs (kHz)
0	0	0	22.05 (MPEG II)
0	0	1	24 (MPEG II)
0	1	0	16 (MPEG II)
0	1	1	Reserved
1	0	0	44.1 (MPEG I)
1	0	1	48 (MPEG I)
1	1	0	32 (MPEG I)
1	1	1	Reserved

2.1.6 โครงสร้างส่วนติดต่อกับเอ็มเอ็มซี

2.1.6.1 เอ็มเอ็มซีรีจิสเตอร์ ในการ์ดประกอบด้วย 5 รีจิสเตอร์ คือ

OCR – Operation Conditions Register มีขนาด 32 บิต เป็นรีจิสเตอร์แบบอ่านอย่างเดียว เก็บข้อมูลเกี่ยวกับไฟเลี้ยง Vdd ของตัวการ์ด

CID – Card Identification มีขนาด 128 บิต เป็นรีจิสเตอร์แบบอ่านอย่างเดียว เพราะข้อมูลเกี่ยวกับตัวการ์ดจะถูกเขียนข้อมูลมาจากโรงงานในกระบวนการผลิต

CSD – Card-Specific Data มีขนาด 128 บิต เป็นรีจิสเตอร์ที่ใช้ตั้งค่าการทำงานเกี่ยวกับตัวการ์ด เช่น ขนาดของบล็อกที่ใช้อ่านหรือเขียน รูปแบบการเก็บข้อมูล ความเร็วในการอ่านเขียนและคุณสมบัติอื่นๆ สามารถเข้าถึงและอ่านเขียนได้

RCA – Relative Card Address Register มีขนาด 16 บิต เป็น แอดเดรส ของการ์ดที่โฮสต์ ให้กับการ์ด ขณะอยู่ในกระบวนการ Card Identification ตอนเริ่มต้นการเชื่อมต่อ และใช้อ้างอิงการ์ดนี้ขณะใช้งานตลอด

DSR – Driver Stage Register มีขนาด 16 บิต เป็นการตั้งค่าเกี่ยวกับการเชื่อมต่อของบัส เช่น อัตราการส่งข้อมูล ขนาดความยาวของบัส แต่เป็นรีจิสเตอร์ที่ใช้เพื่อปรับแต่งขั้นสูง เพื่อเพิ่มประสิทธิภาพเท่านั้น

2.1.6.2 Bus Line สายสัญญาณบัส ของการ์ดนั้นประกอบด้วยสาย 3 ประเภท คือ

Power คือ Vdd 1 เส้น, Vss 2 เส้น

Data คือ MDAT สายส่งข้อมูล 1 เส้นและ MCMD สายส่งคำสั่ง 1 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clock คือ MCLK ใช้สำหรับการ Synchronize สายข้อมูล

2.1.6.3 หลักการของโปรโตคอล (Protocol Concept)

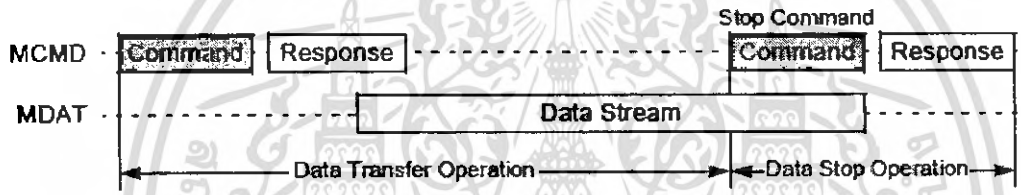
การติดต่อเริ่มขึ้นเมื่อไฟเลี้ยงส่งถึงการ์ด ในภาวะที่ไฟเลี้ยงคงที่ประมาณ 100 มิลลิวินาที แล้วโฮสต์ (ไมโครคอนโทรลเลอร์) ก็จะส่งคำสั่ง (Command) ไปร้องขอการเชื่อมต่อ หลักการของสายในการเชื่อมต่อมีแนวคิด ดังนี้

Command คือ คำสั่ง+Argument ที่ส่งจากโฮสต์ สู่การ์ดบนสาย MCMD

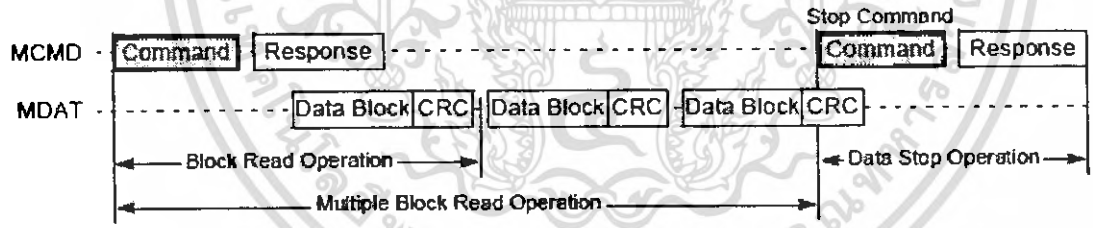
Response คือ การตอบรับ ที่ส่งจาก การ์ดสู่โฮสต์บนสาย MCMD

Data คือ ข้อมูลบนสาย MDAT ที่จะส่งจากทางไหนก็ได้ อีกฝ่ายก็รับ

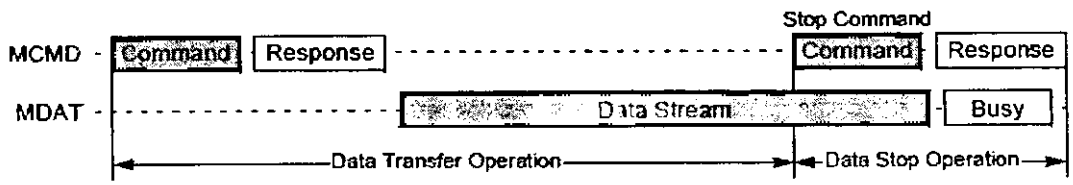
รูปภาพแสดงการใช้บัสไลน์โดยสีเทาเป็นของโฮสต์ และสีขาวเป็นของการ์ต



รูปที่ 2.11 แสดง การทำงานการอ่านข้อมูลจากการ์ดแบบสตรีม (Stream)

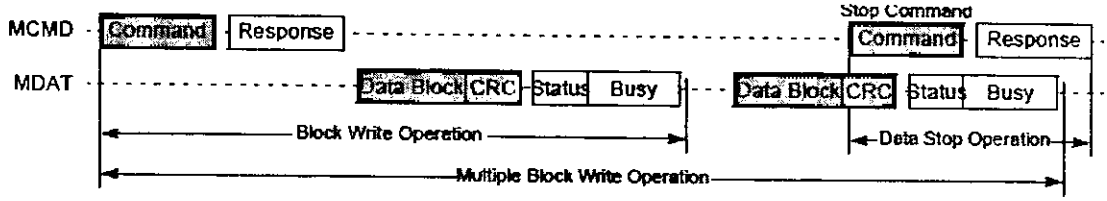


รูปที่ 2.12 แสดง การทำงานการอ่านข้อมูลจากการ์ด แบบมัลติบล็อก (Multi-block)
(มี CRC7 ต่อท้ายแต่ละบล็อก)



รูปที่ 2.13 แสดง การทำงานการเขียนข้อมูลลงการ์ด แบบสตรีม
(หลังจากการเขียนข้อมูลเสร็จการ์ด จะทำการเขียนลง Flash จริงเป็น Busy)

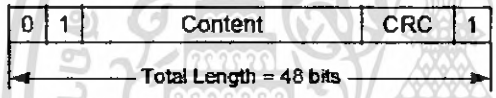
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดง การทำงานการเขียนข้อมูลลงการ์ด แบบมัลติบล็อก
(หลังจากการเขียนข้อมูลเสร็จการ์ด จะทำการเขียนลง Flash จริงเป็น Busy และมี CRC7)

2.1.6.4 รูปแบบของคำสั่ง (Command Format)

รูปแบบของคำสั่งนั้นมีรูปแบบเดียว คือ มีความยาว 48 บิต โดยจะมีบิตเริ่มต้น (Start Bit) เป็น '0' และ ทรานสมิชชั่นบิต (transmission bit) เป็น '1' มีส่วนข้อมูลยาว 38 บิต ตามมาด้วยส่วนตรวจสอบความถูกต้อง (CRC7) อีก 7 บิตและมี บิตหยุด (Stop bit) เป็น '1' ดังภาพ



Bit Position	47	46	45:40	39:8	7:1	0
Width (Bits)	1	1	6	32	7	1
Value	'0'	'1'	-	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Argument	CRC7	End bit

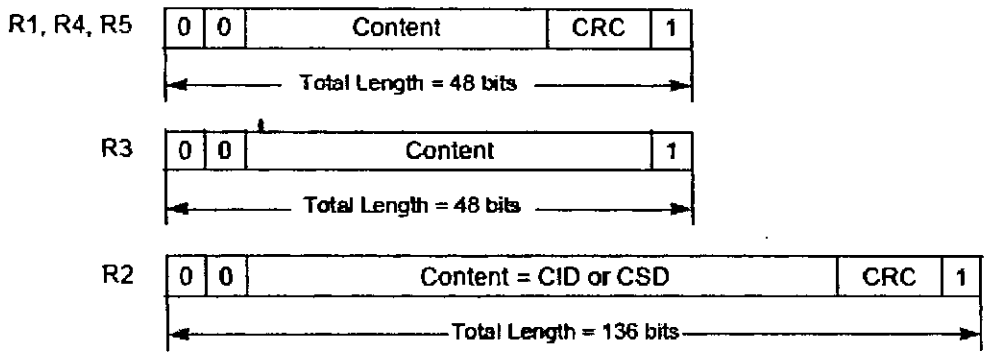
รูปที่ 2.15 แสดงรูปแบบบิตของข้อมูลคำสั่ง

สังเกตว่าในส่วนของข้อมูลของคำสั่งนั้น จะเป็น 2 ส่วนคือคำสั่งขนาด 6 บิต เช่น CMD7 = '000111' CMD12 = '001100' เป็นต้น และส่วนต่อจากนั้นเป็น อาร์กิวเมนต์ (Argument) ซึ่งถ้ามีก็เป็นไปตามแต่ละคำสั่ง หากไม่มีหรือมีไม่ครบ 32 บิด ก็จะเติม '0' ด่อท้าย

2.1.6.5 รูปแบบของเฟรมการตอบสนอง (Response Format)

Response คือ การตอบรับที่มาจากการ์ดบนสาย MCMD มีด้วยกัน 5 แบบ แบ่งเป็น 3 Format คือ ขนาด 48 บิตแบบมี CRC7, 48 บิดแบบไม่มี CRC7 และ 136 บิตมี CRC7 ดังรูป แต่ทุกแบบจะมี Start bit = '0', Transmission bit = '0', Stop bit = '1' เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงรูปแบบบิตของข้อมูลการตอบสนอง

ตารางที่ 2.3 แสดงการตอบรับแบบปกติทั่วไป (Card Status) มีความยาว 48 บิต มี CRC

Bit Position	47	46	45:40	39:8	7:1	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	-	-	-	'1'
Description	Start bit	Transmission bit	Command index	Card Status	CRC7	End bit

ตารางที่ 2.4 แสดงการตอบรับการร้องขอ CID หรือ CSD มีความยาว 136 บิต มี CRC
(แก้ไขส่วน Argument มี 128 บิต เพราะ CID และ CSD มีขนาด 128 บิต)

Bit Position	135	134	[133:128]	[127:1]	0
Width (bits)	1	1	6	32	1
Value	'0'	'0'	'111111'	-	'1'
Description	Start bit	Transmission bit	Reserved	Argument	End bit

ตารางที่ 2.5 แสดงการตอบรับการร้องขอ OCR มีความยาว 48 บิต แต่ไม่มี CRC
(สังเกตว่า ส่วนของ CRC เป็น '1111111' เพราะไม่ใช้)

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	'111111'	-	'1111111'	'1'
Description	Start bit	Transmission bit	Reserved	OCR register	Reserved	End bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

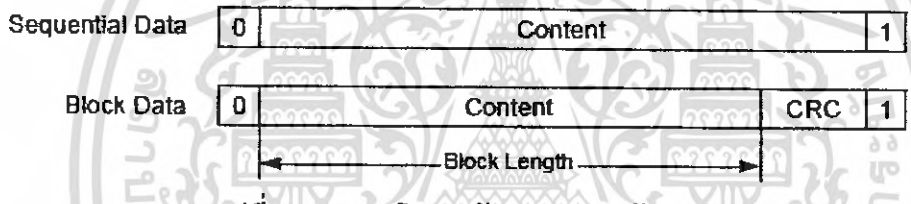
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.6 แสดงการตอบรับอื่นเกี่ยวกับการรับ-ส่งข้อมูล มีความยาว 48 บิต มี CRC

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	'100111'	-	-	'1'
Description	Start bit	Transmission bit	Command Index	Argument	CRC7	End bit

2.1.6.6 รูปแบบของเฟรมข้อมูล (Data Format)

ข้อมูลบน MDAT แบ่งเป็น 2 รูปแบบ คือ แบบสตรีมและแบบบล็อก ตามโหมดในการรับหรือส่งข้อมูล โดยทั้ง 2 รูปแบบมีขนาดเท่าใดก็ได้ แต่มีบิตเริ่มต้นเป็น '0' และบิตหยุดเป็น '1' ถ้าเป็นรูปแบบของสตรีมจะไม่มี CRC แต่ถ้าเป็นแบบบล็อก จะมี CRC ขนาด 16 บิต เรียกว่า CRC16 ต่อท้าย เพื่อตรวจสอบความถูกต้อง ดังรูป



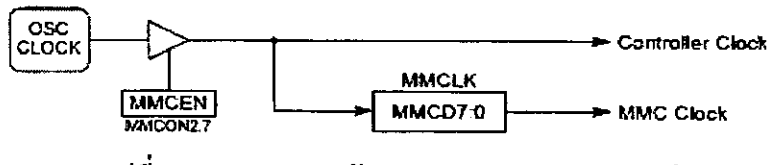
รูปที่ 2.17 แสดงโครงสร้างของเฟรมข้อมูล

2.1.6.7 Clock Generator

สัญญาณนาฬิกาที่เตรียมไว้สำหรับการติดต่อกับเอ็มเอ็มซีสามารถปรับแต่งได้โดยใช้รีจิสเตอร์ MMCLK มีขนาด 8 บิต โดยการสร้างสัญญาณนาฬิกานี้นำมาจากสัญญาณนาฬิกาOSC การคำนวณความถี่สัญญาณนาฬิกาบนสาย MCLK ดังนี้

$$f_{MCLK} = \frac{f_{OSC}}{MMCLK + 1} \quad (2.3)$$

(2.3) สมการคำนวณความถี่สัญญาณนาฬิกาบนสาย MCLK

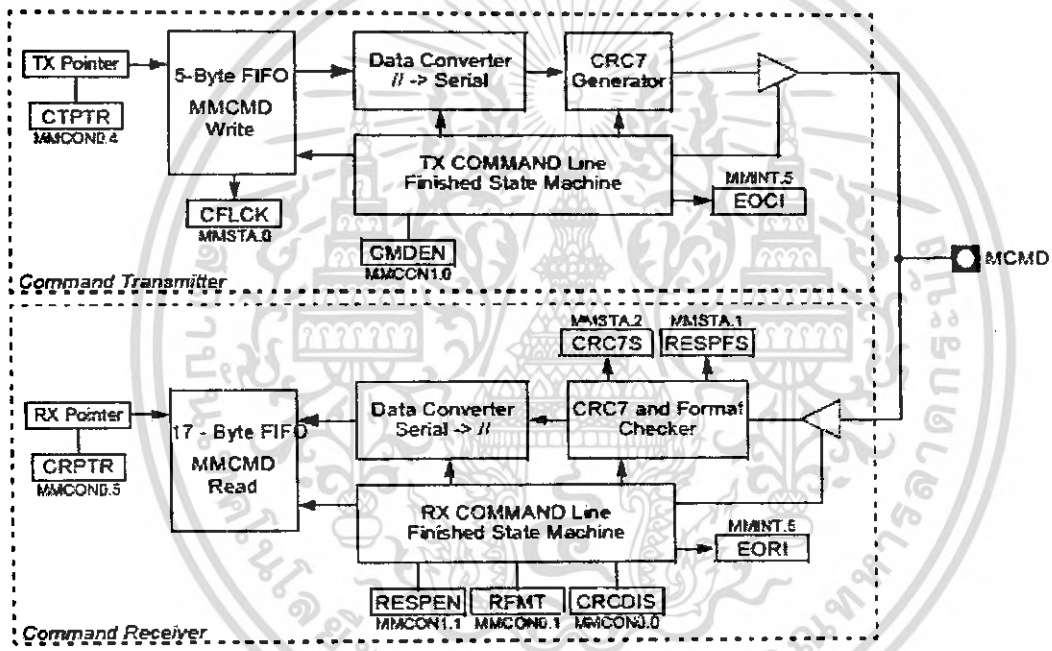


รูปที่ 2.18 แสดงการสร้างสัญญาณสำหรับเอ็มเอ็มซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
72964

2.1.6.8 Command line Controller

การทำงานและควบคุมการทำงานบน MCMD นั้นต้องควบคุมการส่งคำสั่งและรับ การตอบสนองจากการ์ดได้ด้วย ดังนั้นจึงแบ่งเป็น 2 ส่วน คือ ส่วนส่ง Transmission โดยมีบัฟเฟอร์รองรับการส่งคำสั่งอย่างต่อเนื่องได้โดยมีบัฟเฟอร์แบบ FIFO ขนาด 5 ไบต์ และส่วนสร้าง CRC7 อัตโนมัติ จากคำสั่งที่ส่งเข้ามา พร้อมด้วย รีจิสเตอร์สถานะ (Register Status) และอินเทอร์รัปที่เกี่ยวข้อง เช่น CFLCK ควบคุม FIFO การส่ง และ EOCI เป็นอินเทอร์รัปที่จะเป็น '1' เมื่อส่ง คำสั่งเสร็จแล้ว และอีกส่วนหนึ่ง คือ ส่วนรับการตอบสนองก็จะมีกระบวนการคล้ายการส่ง เช่น มีส่วนตรวจ CRC7 มีบัฟเฟอร์ FIFO แต่มีขนาดมากกว่าเป็น 17 ไบต์ ดังรูป



รูปที่ 2.19 แสดงส่วนของ Command Line Controller

2.1.6.9 การส่งคำสั่ง (Command Transmitter) การจะส่งข้อมูลได้นั้นเราต้องทำการขั้นตอน ดังนี้

1. ตั้งค่า RESPEN (Response Enable) ว่าต้องการการตอบรับกลับ หรือไม่? ('1'-Yes, '0'-No)
2. ตั้งค่า RFMT (Response Format) ว่าต้องการการตอบรับรูปแบบ Format ใด?('1'-48 bit, '0'-136 bit)
3. ตั้งค่า CRCDIS (CRC Disable) ว่าต้องการใช้ CRC หรือไม่ ('1'-No, '0'-Yes)
4. นำ Command Index + Argument ที่ต้องการ ไปที่ MMCMD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

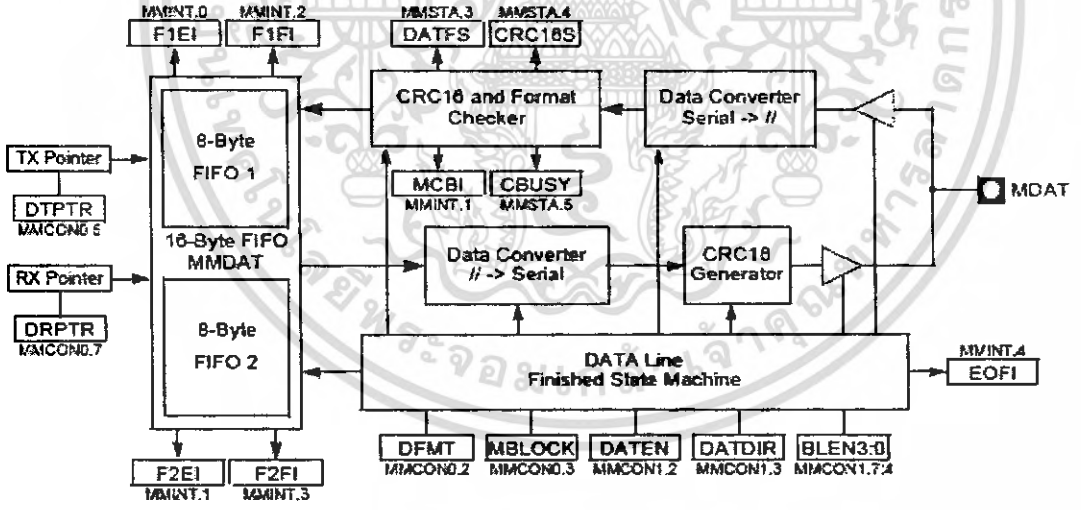
5. เซต ค่า CMDEN = '1' และ เคลียร์ ค่า CMDEN = '0' เพื่อเริ่มทำการส่ง
6. ในขณะการส่งสามารถ ยกเลิกการส่งได้โดย เคลียร์ ค่า CTPTR = '0'
7. ถ้าการส่งเสร็จสิ้นจะไมโครคอนโทรลเลอร์จะ เซต ค่า EOCI = '1' เพื่อ อินเทอร์รัป

2.1.6.10 Command Receiver (Response)

1. เมื่อมีข้อมูลเข้ามาแล้วไมโครคอนโทรลเลอร์จะใช้ข้อมูล RESPEN, RFMT, CRCDIS มาตรวจสอบข้อมูลที่รับเข้ามา ถ้าถูก EORI จะ เซต อัตโนมัติที่อินเทอร์รัป
2. เราสามารถอ่านข้อมูลจาก MMCMD ได้
3. หากเราจะทำการยกเลิกการรับ ให้เคลียร์ค่า CRPTR

2.1.6.11 Data Line Controller

MMDAT จะเป็นรีจิสเตอร์แบบ FIFO ขนาด 16 ไบต์ แต่ถูกแบ่งออกเป็นส่วนส่งและส่วนรับอย่างละ 8 ไบต์ มีการทำงานคล้ายกับส่วนจัดการคำสั่งคือ มี CRC, บัฟเฟอร์ และสถานะต่างๆ เพื่อควบคุมและตรวจการทำงานของสาย MDAT ดังภาพ



รูปที่ 2.20 แสดงส่วนของ Data Line Controller

สังเกตว่าที่บัฟเฟอร์MMDAT เป็น FIFO นั้นจะมีการดูแลสถานะโดยแบ่งเป็น

- F1EI : บอก FIFO ฝั่งส่ง ว่าง
- F1FI : : บอก FIFO ฝั่งส่ง เต็ม
- F2EI : บอก FIFO ฝั่งรับ ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- F2FI : : บอก FIFO ฝั่งรับ เต็ม

แบ่งการทำงานออกเป็น 3 แบบตามโหมดของ เอ็มเอ็มซี คือ

- ถ้ารับ-ส่งแบบ Stream (DFMT = '0')
- ถ้ารับ-ส่งแบบ Single-Block (DFMT= '1',MBLOCK= '0',BLEN=ความยาว Block)
- ถ้ารับ-ส่งแบบ Multi-Block (DFMT= '1',MBLOCK= '1',BLEN=ความยาวบล๊อก)

2.1.6.12 Data Transmitter

1. ตั้งค่า DATDIR ทิศทางของข้อมูล (DATDIR= '1')
2. เขียนข้อมูลลง MMDAT ทีละ 16 ไบต์
3. เริ่มส่งข้อมูลโดย เซต DATEN = '1' แล้ว เคลียร์ DATEN = '0'
4. ถ้าส่งข้อมูลเสร็จเรียบร้อยแล้ว EOFI จะ เซต ถ้าเป็น Stream Mode ก็ถือว่าเสร็จเรียบร้อยแล้ว แต่ถ้าเป็น Block Mode ก็ยังตั้ง Check ที่ DATFS และ CRC16S ว่าถูกต้องหรือไม่ ถ้าทำเสร็จแล้วทำการทดสอบว่าถูกต้องก็จะ เซต
5. หลังจากส่งข้อมูลเสร็จแล้ว การ์ด จะเป็น Busy Status สามารถอ่านได้จาก CBUSY bit
6. หากต้องการยกเลิกการส่ง ต้อง เคลียร์ DTPTR = '0'

2.1.6.13 Data Receiver

1. ตั้งค่า DATDIR ทิศทางของข้อมูล (DATDIR= '0')
2. ขณะรับข้อมูลอยู่จะไม่สามารถอ่านได้ แต่สังเกตสถานะที่ DATFS (Frame Format) ถูกต้องหรือไม่ และ CRC16S การคำนวณ CRC ว่าถูกต้องหรือไม่
3. จะอ่านข้อมูลได้ต่อเมื่อ F2FI บอกเต็มแสดงว่าสามารถอ่านข้อมูลจาก MMDAT ได้
4. หากต้องการยกเลิกการรับ ต้อง เคลียร์ DRPTR = '0'

2.1.6.14 โฟลว์คอนโทรล (Flow Control):

ไมโครคอนโทรลเลอร์มีการจัดการโฟลว์คอนโทรลดังนี้ และจะทำงานเมื่อเรา เซต FLOWC โดยจะทำการควบคุมดังนี้

<u>ขณะการส่ง</u>

- MMCLK จะหยุดเมื่อ F1EI เซต บอกว่า FIFO ว่าง
- MMCLK จะเริ่มทำงานเมื่อ F1FI เซต บอกว่า FIFO เต็มส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะการรับ

- MMCLK จะหยุดเมื่อ F2FI เซต บอกว่า FIFO เต็มแล้ว
- MMCLK จะเริ่มทำงานเมื่อ F2EI เซต บอกว่า FIFO ว่างแล้วรับได้

2.1.7 โครงสร้างส่วนการส่งสัญญาณเสียงออก (Audio Output)

ส่วนนี้สามารถทำงาน โดยสามารถเลือกอินพุตที่เป็นข้อมูลเสียงได้ 2 แหล่ง คือ จากส่วนควบคุมจากสายสัญญาณข้อมูล (Data bus) หรือ ส่วนถอดรหัสเอ็มพีสามทำให้สามารถตั้งค่าให้เล่นเพลงเอ็มพีสามได้เลยโดยไม่ต้อง นำข้อมูลที่ผ่านการถอดรหัสแล้วกลับมาป้อนให้ส่วนการส่งสัญญาณเสียงออกเอง การทำงานของส่วนการส่งสัญญาณเสียงออกเองนี้ให้ผลลัพธ์ เป็นข้อมูลดิจิตอลแบบ พีซีเอ็ม และแบบ ไอสแควร์เอส โดยทั้งสองแบบนี้จะเป็นแบบสเตอริโออยู่แล้ว เราจึงต้องมีวงจรมาแปลงสัญญาณดิจิตอลให้กลายเป็น อนาล็อกอีกทีหนึ่งภายนอก ส่วนของการส่งสัญญาณเสียงออกเองมีด้วยกัน 4 ขาสัญญาณด้วยกัน คือ

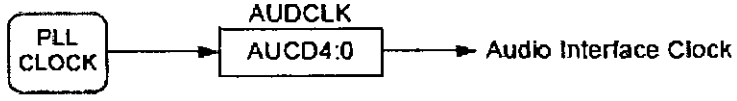
- SCLK เป็นขาสัญญาณ System clock หรือ Sampling rate
- DCLK เป็นขาสัญญาณ Bit Clock สัญญาณนาฬิกาสำหรับการอ่านข้อมูล
- DSEL เป็นขาสัญญาณ Select Channel เลือกว่าสัญญาณนี้เป็นซ้ายหรือขวา
- DOUT เป็นขาสัญญาณข้อมูลดิจิตอล

2.1.7.1 สัญญาณนาฬิกาที่ใช้ในส่วนการส่งสัญญาณเสียงออก

สัญญาณนาฬิกานั้นรับมาจาก PLL Clock แล้วนำมาหารด้วย รีจิสเตอร์ AUDCLK มีขนาด 5 บิต จึงได้สัญญาณนาฬิกาที่ใช้ในส่วน ออดิโอ นี้

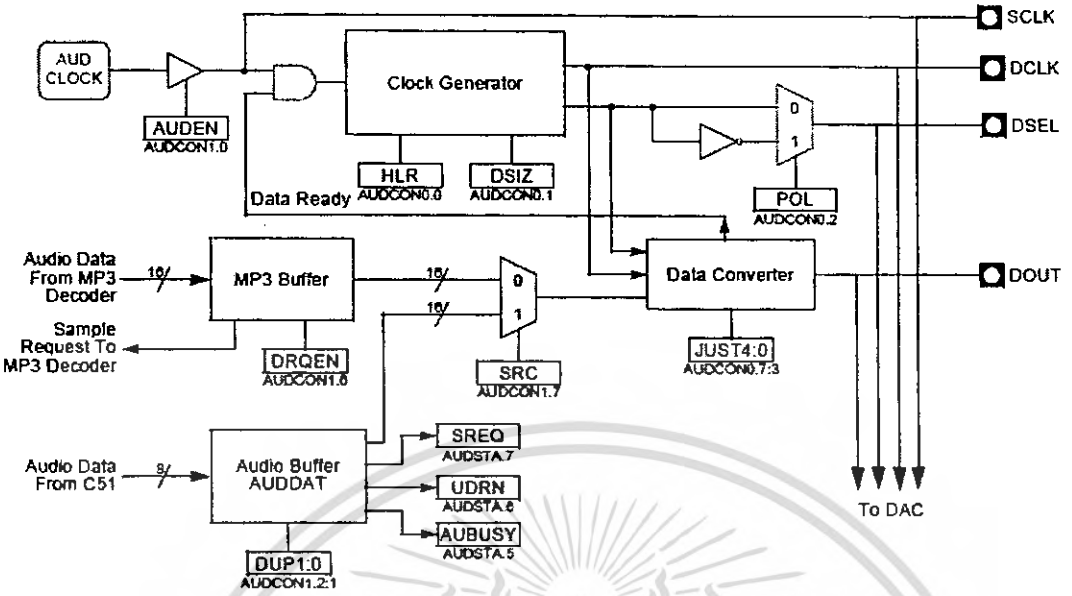
$$AUDCLOCK = \frac{PLLCLOCK}{AUDCLK + 1} \qquad (2.4)$$

(2.4) สมการแสดงการคำสัญญาณนาฬิกาในส่วนของ ส่วนส่งสัญญาณเสียงออก



รูปที่ 2.21 แสดงบล็อกไดอะแกรมของการหาสัญญานาฬิกาส่วนส่งสัญญาณเสียงออก

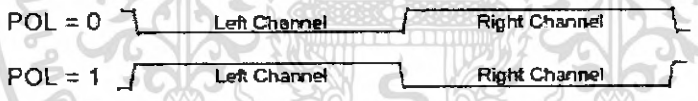
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 แสดงภาพรวมของส่วนส่งสัญญาณเสียงออก

2.1.7.2 การตั้งค่าส่วนการส่งสัญญาณเสียงออก

ที่ขา DSEL เป็นขาที่บอกสัญญาณว่าเป็นข้อมูลของ ซ้ายหรือขวา มีการตั้งค่าด้วยบิต POL ได้ดังภาพ



รูปที่ 2.23 แสดงขาสัญญาณที่บอกว่าเป็นข้อมูลของลำโพงซ้ายหรือขวา

Bit DSIZ เป็นบิตที่ใช้เลือกว่าต้องการข้อมูลที่มีความยาวเท่าใด ถ้า DSIZ = 0 แสดงว่าข้อมูลต้องการ 16 บิต แต่ถ้าเป็น 1 หมายถึงข้อมูลมากกว่า 16 บิต

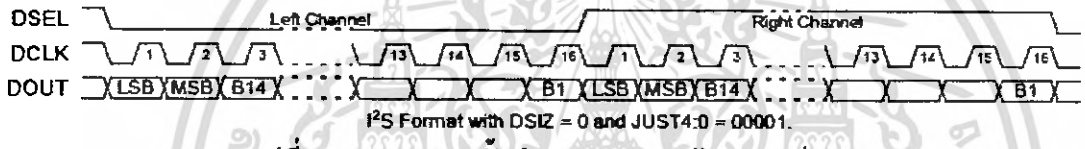
ส่วน JUST4:0 เป็นรีจิวเตอร์ที่ใช้ตั้งค่าปรับแต่งการจัดวางของข้อมูลในแต่ละแชนแนล ว่าจัดแบบ MSB หรือ LSB มาก่อนและเติม '0' ข้างหน้าหรือข้างหลังหากข้อมูลไม่ครบเท่า Bit Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 แสดงรูปแบบข้อมูลกับการตั้งค่าของส่วนส่งสัญญาณเสียงออก

Dac Format	POL	DSIZ	JUST4:0
16-bit I^2S	0	0	00001
> 16-bit I^2S	0	1	00001
16-bit PCM	1	0	00000
18-bit PCM LSB justified	1	1	01110
20-bit PCM LSB justified	1	1	01100
20-bit PCM MSB justified	1	1	00000

ในโครงงานนี้เลือก แบบ ไอสแควร์เอส ขนาด 16 บิดเพราะสามารถเข้ากันได้กับ PCM1725 ที่นำมาแปลงสัญญาณภายนอก



รูปที่ 2.24 แสดงการตั้งค่าตามหลักของ ไอสแควร์เอส

การตั้งค่าให้สามารถเล่น เอ็มพีสาม ได้จาก ส่วนถอดรหัสเอ็มพีสาม ได้เลย ทำดังนี้

1. ตั้งค่าบิต HLR ที่AUDCON.0 บอกว่า '1' – PLL Clock = 384Fs '0' – PLL Clock = 256Fs
2. DSIZ = '0' เพราะใช้ข้อมูลขนาด 16 บิต
3. POL = '0' เพราะตาม รูปแบบ ของ ไอสแควร์เอส
4. JUST4:0 = '00001' ตาม รูปแบบ ของ ไอสแควร์เอส
5. SRC = '0' เพราะเลือกซอร์ซจาก เอ็มพีสาม แต่ถ้าหากใช้จาก C51 เลือก '1'
6. AUDEN = '1' เริ่มการทำงานของส่วน ออดิโอ
7. DRQEN = '1' นั่นคือหาก เอ็มพีสาม ไม่มีข้อมูลก็ไม่แปลงออกเสียง แต่หากมีข้อมูลก็แปลงเป็นเสียง

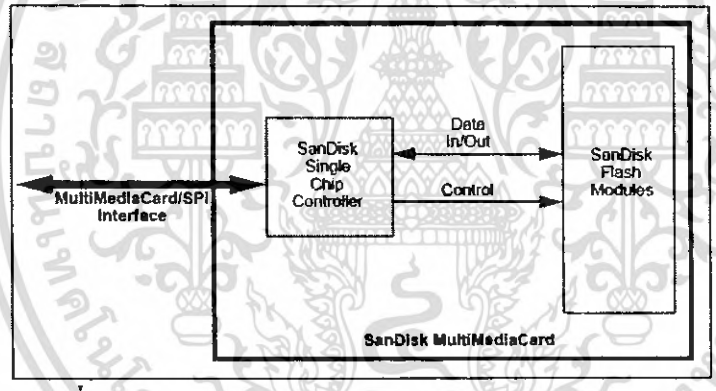
2.2 การใช้งานเอ็มเอ็มซี

2.2.1 ความรู้ทั่วไปเกี่ยวกับเอ็มเอ็มซี

เอ็มเอ็มซีเป็นอุปกรณ์เก็บข้อมูลที่มีขนาดเล็กมาก และได้ถูกออกแบบมาให้มีขนาดเท่ากับอุปกรณ์เก็บข้อมูลแบบอื่นๆ เพื่อให้สามารถใช้ร่วมกันได้บนอุปกรณ์ที่ใช้งานด้านดิจิตอล เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องดิจิตัล คอมพิวเตอร์ โทรศัพท์มือถือ และอุปกรณ์ชนิดอื่นอีกมากมาย ทำให้เอ็มเอ็มซีเป็นที่แพร่หลาย อีกทั้งราคาเอ็มเอ็มซีมีราคาถูก หากเทียบกับความจุของข้อมูลที่สามารถเก็บได้ และข้อดีของอุปกรณ์ลักษณะนี้ คือ สามารถพกพกสะดวก เนื่องจากมีขนาดเล็ก กะทัดรัด น้ำหนักเบา ไม่ต้องใช้พลังงานในการเก็บข้อมูลไว้ และในขณะใช้งานอ่านหรือเขียนข้อมูลก็ใช้พลังงานเพียงเล็กน้อย สามารถทนแรงสั่นสะเทือนได้ การใช้งานเอ็มเอ็มซีนั้นเราสามารถอ่านหรือเขียนข้อมูลเอ็มเอ็มซีโดยผ่านโปรโตคอลที่เอ็มเอ็มซียอมรับ ซึ่งสามารถทำได้ 2 แบบ คือ แบบเฉพาะตัวของเอ็มเอ็มซีเอง (MultiMediaCard Interface) หรือแบบ เอสพีไอ (SPI :Serial Peripheral Interface) เป็นแบบทั่วไปที่ใช้ได้ทั้งในเอสดีการ์ดด้วย โดยในเอ็มเอ็มซีนั้นจะมีส่วนควบคุมการทำงานอยู่แล้ว จะทำงานควบคุมการอ่าน-เขียน ส่งสถานะการทำงาน ควบคุมข้อผิดพลาดที่เกิดขึ้น ตรวจสอบข้อมูลที่ได้รับจากโฮสต์ ใช้ CRC และสร้าง CRC ขณะที่จะส่งข้อมูลกลับไปให้โฮสต์ เพื่อรับประกันว่าข้อมูลที่อ่านและเขียนทุกครั้งนั้นถูกต้องเสมอ ควบคุมเรื่อง Write-Protection ป้องกันข้อมูลในส่วนที่ห้ามเขียนหรือลบข้อมูล และสามารถต่อเอ็มเอ็มซีบนสายสื่อสารชุดเดียวกันได้หลายตัว



รูปที่ 2.25 แสดงบล็อกไดอะแกรมภายในของของเอ็มเอ็มซี

2.2.2 ลักษณะการเชื่อมต่อและโปรโตคอล

ตารางที่ 2.8 ข้อเปรียบเทียบรูปแบบการเชื่อมต่อของเอ็มเอ็มซี

MMC Interface	SPI Interface
ใช้สายสัญญาณ 3 เส้น (Clock,Command,Data)	ใช้สายสัญญาณ 3 เส้น (Clock,DataIn,DataOut) และขา CS เพื่อเลือกใช้งาน การ์ด นั้น
ใช้การ์ดบนสายชุดเดียวกันได้มากถึง 65,5369 ตัว	ใช้ได้หลายตัวขึ้นอยู่กับขา CS มีแค่ไหน
สามารถแยกแยะการ์ดได้ง่าย	ไม่มีความสามารถนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MMC Interface	SPI Interface
มีส่วนจัดการกับ Error ที่เกิดขึ้น	ไม่มีความสามารถนี้
สามารถอ่านเขียนข้อมูลได้ 3 แบบ คือแบบ - Data Stream - Single Block - Multiple Block	สามารถอ่านเขียนข้อมูลได้ 2 แบบ คือแบบ - Single Block - Multiple Block

2.2.2.1 Flash Technology Independence

1 เซคเตอร์ ของ เอ็มเอ็มซี มีขนาด 512 ไบต์ซึ่งเท่ากับขนาด เซคเตอร์ ของฮาร์ดดิสก์ ทำให้สามารถอ่าน-เขียนข้อมูลด้วยกันได้ง่ายและ เอ็มเอ็มซี มีส่วนรับการเขียนข้อมูลที่ส่งมาด้วยตัวเองแล้ว ทำให้ โฮสต์ ไปต้องจัดการอ่านหรือเขียนเอง แต่เป็นหน้าที่ของส่วนควบคุมบน เอ็มเอ็มซี ดังนั้นไม่ว่าโฮสต์ จะเปลี่ยนซอฟต์แวร์ หรือลักษณะการเก็บข้อมูลแบบใดก็ไม่กระทบต่อ เอ็มเอ็มซี เพราะสามารถเข้ากันได้เลย และไม่ว่าการเก็บข้อมูลในการ์ด จะเป็นเช่นใดผู้สร้างแอพพลิเคชั่นไม่ต้องคำนึงถึงเพราะเรายังใช้การติดต่อแบบเดิม

2.2.1.2 Defect and Error Management

เอ็มเอ็มซี มีส่วนตรวจสอบและควบคุม Error เพื่อให้การอ่าน-เขียนข้อมูลมีความน่าเชื่อถือ โดยใช้ CRC ในการควบคุม ถ้าหากโฮสต์จะส่งข้อมูลไปเขียนใน เอ็มเอ็มซี ก็จะต้องมีการส่ง CRC ตามหลังข้อมูลไปเพื่อรับประกันความถูกต้อง แล้วส่วนควบคุมจะทำการตรวจสอบข้อมูลที่ได้มากับ CRC ที่ได้รับหากไม่ถูกต้องจะตอบ Error กลับไปที่โฮสต์ทำให้ทราบว่าการส่งไม่ถูกต้อง และมีปัญหาใด และถ้าเกิด Error ในกรณีอื่นๆ เช่นโฮสต์พยายามเขียนหรือลบในส่วนที่เป็น Write-Protection ก็จะมีการส่ง Error กลับไปบอกโฮสต์หรือส่งข้อมูลในสภาวะที่ เอ็มเอ็มซี ไม่พร้อม เช่น กำลังเขียนข้อมูลลง หน่วยความจำแบบแฟลช เป็นต้น

2.2.1.3 ความทนทาน (Endurance)

เอ็มเอ็มซี สามารถเขียนข้อมูลลงในแต่ละ เซคเตอร์ ได้ประมาณ 100,000 ครั้ง แต่อ่านได้ไม่จำกัด จึงสามารถนำไปใช้อ่านเขียนข้อมูลในโทรศัพท์มือถือ เครื่องอัดเสียง ได้ จากการทดสอบสามารถใช้งานได้ 34 ปี หากมีการเขียนข้อมูลชั่วโมงละ 3 ครั้ง วันละ 8 ชั่วโมง ปีละ 365 วัน จึงถือว่าใช้งานได้นานมาก

2.2.1.4 Using the Erase Command

เอ็มเอ็มซี มีคำสั่งลบข้อมูลโดยเฉพาะ ซึ่งทำงานได้เร็วกว่า คำสั่งเขียนข้อมูลปกติ โดยสามารถลบเป็น เซคเตอร์ ไป

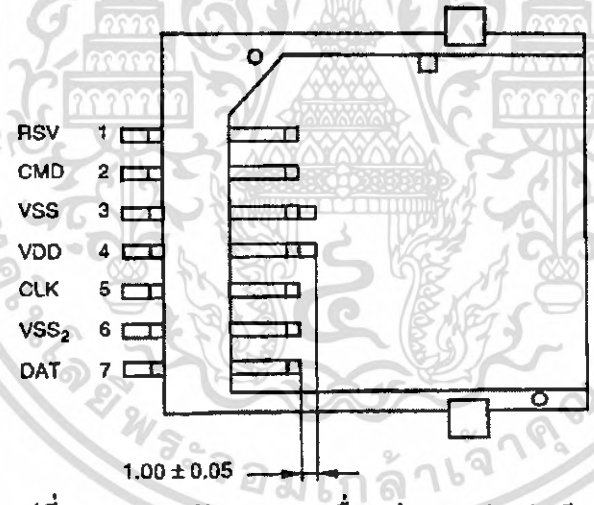
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.5 Automatic Sleep Mode

คุณสมบัติประหยัดพลังงาน เมื่อไม่มีมีคำสั่งหรือการทำงานอื่นภายใน 5 มิลลิวินาที เอ็มเอ็มซี เองจะเข้าสู่สภาวะ Sleep mode เพื่อประหยัดพลังงาน และออกจากสภาวะ Sleep Mode เมื่อมีคำสั่งใดๆ มาก็ได้ เอ็มเอ็มซี จะเข้าสู่สภาวะปกติและทำงานตามคำสั่งนั้นทันที ทำให้โฮสต์ไม่ต้องคำนึงว่า เอ็มเอ็มซี อยู่ในสภาวะ Sleep mode หรือไม่ เพราะ เอ็มเอ็มซี จะเข้าและออก โหมด นี้โดยอัตโนมัติ

2.2.1.6 Hot Insertion

หมายถึงการเสียบ เอ็มเอ็มซี เข้าไปในโฮสต์ที่กำลังใช้งานอยู่นั้นช่องเสียบ เอ็มเอ็มซี ได้ถูกออกแบบมาให้ ขา Power นั้นมีความยาวมากกว่า ขาอื่นทำให้ เอ็มเอ็มซี ได้รับไฟเลี้ยงเพียงพอก่อนที่ขาอื่นจะถูกต่อ ส่วนการดึงออกนั้นขา Power ก็จะหลุดเป็นขาสุดท้าย ทำให้ขาอื่นๆ นั้นหลุดออกหมดแล้ว ไม่ว่าจะเป็นการต่อแบบ เอ็มเอ็มซีอินเทอร์เฟซ หรือแบบ เอสพีไอ ก็มีลักษณะนี้ ลักษณะการออกแบบนี้คล้ายกับ PCMCIA



รูปที่ 2.26 แสดงลักษณะการเชื่อมต่อของเอ็มเอ็มซี

2.2.1.7 Negotiating Operation Conditions

เอ็มเอ็มซี สามารถปรับสภาวะของไฟเลี้ยงที่กำลังใช้ติดต่อสื่อสารอยู่นั้นด้วยการปรับรีจิสเตอร์ ภายในให้ เอ็มเอ็มซี รับไฟเลี้ยงปริมาณเท่าใดได้ และสามารถถอดการ์ดออกโดยการสั่งให้ เอ็มเอ็มซี ไปอยู่ในสภาวะ inactive MMC จะไม่ตอบสนองใด และไม่รับไฟเลี้ยงเปรียบเสมือนถอดออกจากระบบแล้ว โดยคำสั่งนี้ คือ GO_INACTIVE_STATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.8 Card Status

โฮสต์สามารถอ่านค่าสถานะของ เอ็มเอ็มซี ในขณะนั้นได้โดยตลอด สถานะของ เอ็มเอ็มซี มีขนาด 32 บิต อ่านได้โดยใช้คำสั่ง SEND_STATUS โดยโฮสต์มักจะถามก่อนที่จะอ่านหรือเขียนข้อมูล

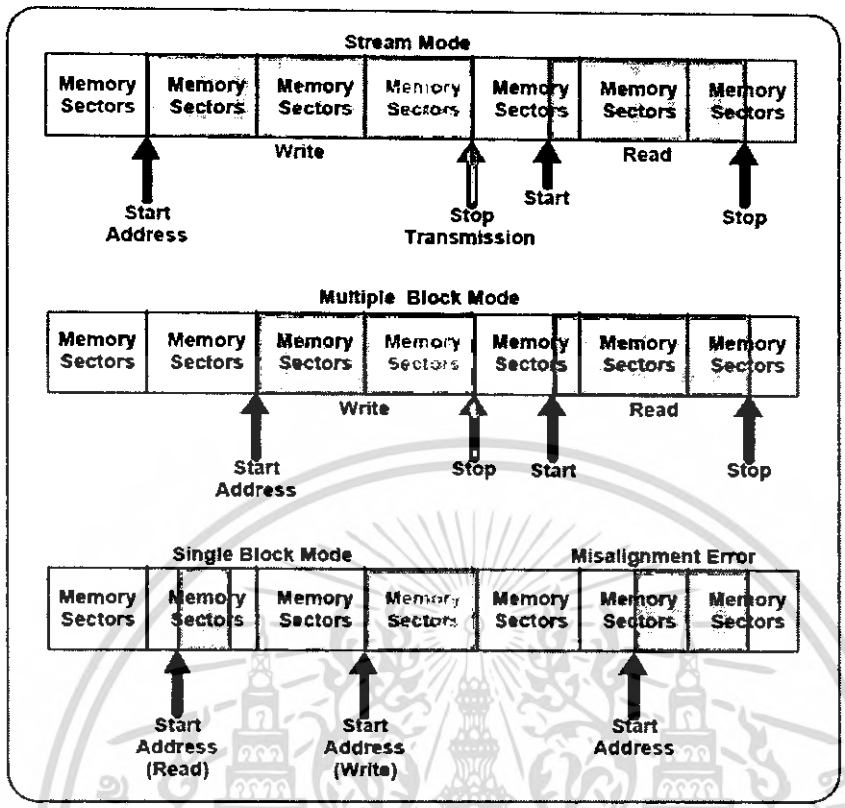
2.2.1.9 CSD and CID Register

CSD คือ รีจิสเตอร์ภายใน เอ็มเอ็มซี ที่อยู่ในส่วนควบคุม เพื่อโฮสต์จะปรับแต่งค่าของเอ็มเอ็มซี ตามต้องการทั้งความเร็วในการติดต่อสื่อสาร หรือข้อมูลเฉพาะอื่นๆ ส่วน CID คือ รีจิสเตอร์ที่เป็นที่เก็บข้อมูลเอกลักษณ์ของการ์ด ที่มาจากโรงงานบ่งบอก S/N โรงงานที่ผลิต รุ่น หรือข้อมูลอื่นๆอีกสามารถอ่านและเขียน(SEND_CSD, PROGRAM_CSD) CSD ได้แต่ CID อ่านได้อย่างเดียว

2.2.1.10 Data Transfer

- **Stream mode** คือการอ่านหรือเขียนแบบต่อเนื่องโฮสต์ส่ง แอดเดรส เริ่มต้นในการอ่านหรือเขียน แล้วจะเริ่มส่งข้อมูลกัน จนกว่าโฮสต์จะสั่งหยุดการส่งข้อมูล แต่การรับส่งข้อมูลแบบนี้ไม่มีการรับประกันความถูกต้องมาควบคุม ทั้งนี้การอ่านนั้นจะเริ่มจากที่ใดก็ได้ แต่การเขียนต้องเริ่มที่ต้น เซคเตอร์ เท่านั้น
- **Single Block** คือการอ่านหรือเขียนเพียง I บล็อก นั่นคือไม่เกิน 512 ไบต์ โฮสต์ส่งเพียง แอดเดรส เริ่มต้น จะเริ่มอ่านและเขียนจนกว่าจะครบ I บล็อก หรือหมดข้อมูล และมีการับประกันความถูกต้องด้วย CRC ขนาด 16 บิต
- **Multiple Block** คือ การอ่านหรือเขียนข้อมูล แบบหลายบล็อก โฮสต์ ต้องส่งทั้ง Start และ Stop เหมือน Stream แต่มีการรับประกันความถูกต้องทีละบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงการส่งข้อมูลในเอ็มเอ็มซีแบบต่างๆ

2.2.1.11 Erase

การลบแบ่งออกเป็น 2 ขั้นตอนดังนี้

- **Tagging (Selecting the Sector for Erasing)** คือ การเลือกที่ที่ต้องการลบ จากแอดเดรสเริ่มต้น ถึงแอดเดรสสุดท้าย และจะถูกยกเลิกเมื่อ ใช้คำสั่ง UNTAG
- **Erasing (Starting Erase Process)** คือ ขั้นตอนการลบ โดยจะถูกลบเมื่อ นำ Tagging ที่เลือกไว้มาลบ

ตารางที่ 2.9 แสดงความต้องการการใช้พลังงานของเอ็มเอ็มซี

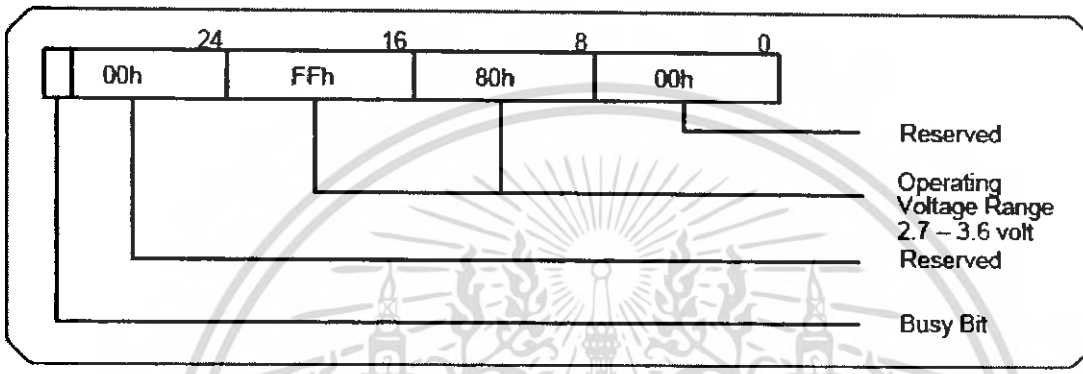
Operation	3.3 V	2.7 V
Read/อ่าน	< 58 mA	< 58 mA
Write/เขียน	< 60 mA	< 58 mA
Sleep/ประหยัดพลังงาน	50 – 150 uA	40 – 120 uA

เอ็มเอ็มซี ควรใช้ไฟเลี้ยงในช่วง 2.7 – 3.6 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.12 Operating Conditions Register (OCR)

เป็นรีจิสเตอร์ขนาด 32 บิต ที่ใช้เก็บค่าเกี่ยวกับไฟเลี้ยง VDD สามารถใช้คำสั่งใน CMD1 (Voltage recognition procedure) ในการตั้งค่า สามารถตั้งค่าไฟเลี้ยงได้ตั้งแต่ในช่วง 2 ถึง 2.6 Volts โดยปกติ การตั้งค่าจะอยู่ประมาณ 2.7 – 3.6 Volts และยังสามารถอ่านค่าว่าการ์ดทำงานอยู่หรือเปล่า (Busy bit) จาก OCR ได้ด้วย ดังภาพ



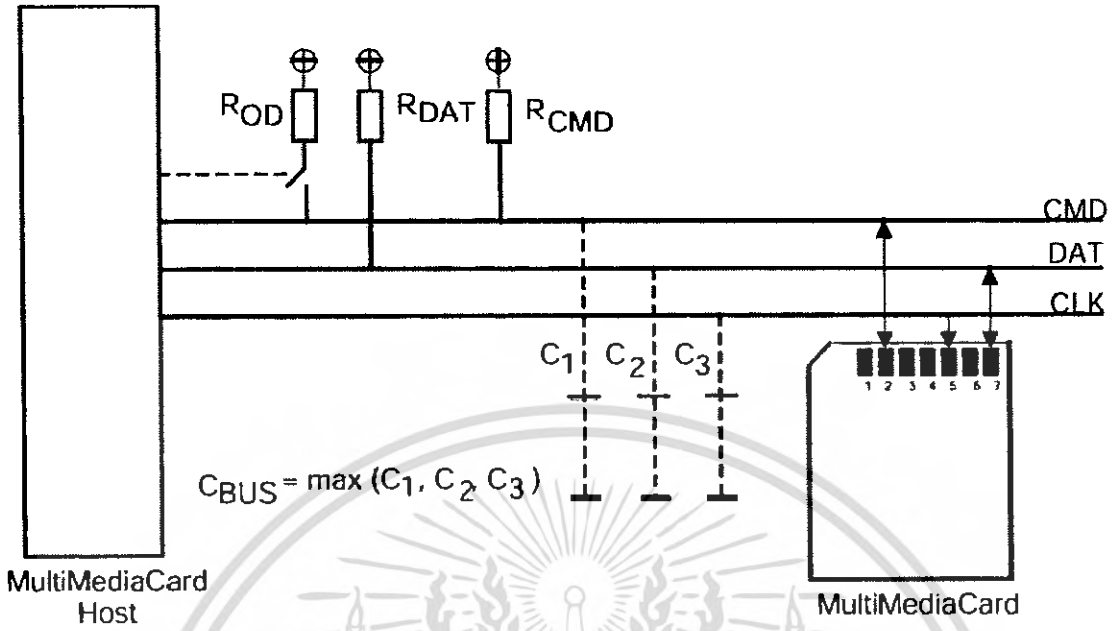
รูปที่ 2.28 แสดงรูปแบบของ Operating Conditions Register (OCR)

ตารางที่ 2.10 แสดงการเชื่อมต่อเอ็มเอ็มซีแบบ MMC Interface

Pin#	Name	Type	Description
1	RSV	NC	ไม่ต้องเชื่อมต่อ ปล่อยเป็น logic 1
2	CMD	I/O/PP/OD	คำสั่ง/ตอบรับคำสั่ง
3	VSS1	S	ต่อ Ground
4	VDD	S	ต่อไฟเลี้ยง
5	CLK	I	ขาสัญญาณนาฬิกา
6	VSS2	S	ต่อ Ground
7	DAT	I/O/PP	ขาข้อมูล

คำอธิบาย NC= “not connect” PP = “push-pull” OD = “Open-drain” S= “Supply”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



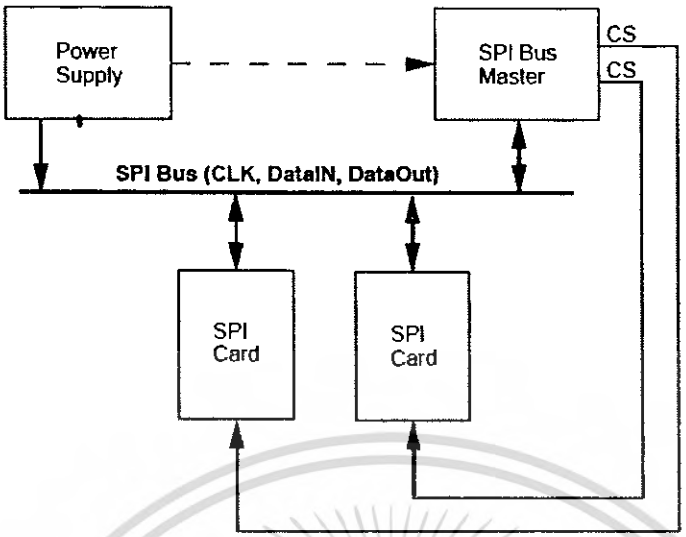
รูปที่ 2.29 แสดงลักษณะของสายสัญญาณที่เชื่อมต่อระหว่างเอ็มเอ็มซีและโฮสต์ของเอ็มเอ็มซี

ตารางที่ 2.11 แสดงการเชื่อมต่อเอ็มเอ็มซีแบบ เอสพีไอ

Pin#	Name	Type	Description
1	CS	I	เลือกใช้ การ์ด นี้(Active low)
2	Data In	I	คำสั่ง/ข้อมูล จาก โฮสต์ สู่ เอ็มเอ็มซี
3	VSS1	S	ต่อ Ground
4	VDD	S	ต่อไฟเลี้ยง
5	CLK	I	ขาสัญญาณนาฬิกา
6	VSS2	S	ต่อ Ground
7	Data Out	O	ตอบรับ/ข้อมูล จาก เอ็มเอ็มซี สู่ โฮสต์

คำอธิบาย S= “Supply”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 แสดงบล็อกไดอะแกรมการทำงานของ เอสพีไอ

2.2.3 คำสั่งและการทำงาน

ตารางที่ 2.12 แสดงคำสั่งและการทำงานของเอ็มเอ็มซี

คำสั่ง	อาร์กิวเมนต์	ตอบสนอง	คำอธิบาย
CMD0	-	-	รีเซต การ์ด สู่ Idle State
CMD1	OCR	R3	อ่าน OCR จากทุก การ์ด
CMD2	-	R2	อ่าน CID จากทุก การ์ด
CMD3	RCA[31:16]	R1	ตั้ง Relation Address ให้ การ์ด
CMD7	RCA[31:16]	R1	เปลี่ยนสถานะระหว่าง Stand by กับ Transfer
CMD9	RCA[31:16]	R2	อ่าน CSD จาก การ์ด ที่ต้องการ
CMD10	RCA[31:16]	R2	อ่าน CID จาก การ์ด ที่ต้องการ
CMD11	Data Address	R1	อ่าน Stream ข้อมูล
CMD12	-	R1	หยุดการส่งข้อมูล
CMD13	RCA[31:16]	R1	อ่านสถานะจาก การ์ด ที่ต้องการ
CMD15	RCA[31:16]	-	เปลี่ยนสถานะ การ์ด เป็น Inactive
CMD16	Block length	R1	ตั้งค่า Block length
CMD17	Data Address	R1	อ่านข้อมูล Single Block
CMD18	Data Address	R1	อ่านข้อมูล Multiple Block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง	อาร์กิวเมนต์	ตอบสนอง	คำอธิบาย
CMD20	Data Address	R1	เขียน Stream ข้อมูล
CMD24	Data Address	R1	เขียนข้อมูล Single Block
CMD25	Data Address	R1	เขียนข้อมูล Multiple Block
CMD27	-	R1	ตั้งค่า CSD
CMD32	Data Address	R1	ตั้งค่า แอดเดรส เริ่มต้น เพื่อเริ่มลบข้อมูล
CMD33	Data Address	R1	ตั้งค่า แอดเดรส สุดท้าย เพื่อเริ่มลบข้อมูล
CMD34	Data Address	R1	ยกเลิกข้อมูลที่เลือกไว้
CMD35	Data Address	R1	ตั้งค่า แอดเดรส เริ่มต้น เพื่อเริ่มลบข้อมูล แบบ Group
CMD36	Data Address	R1	ตั้งค่า แอดเดรส สุดท้าย เพื่อเริ่มลบข้อมูล แบบ Group
CMD37	Data Address	R1	ยกเลิกข้อมูลที่เลือกไว้ แบบ Group
CMD38	-	R1	ลบข้อมูลที่เลือกไว้ทั้งหมด

2.3 การเชื่อมต่อไอสแควร์ซีแบบสองเส้น (Two-wire Interface I^2C)

I^2C ย่อมาจาก **Inter-IC Communication** หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัสไอสแควร์ซีได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงานและควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือสายข้อมูล อีกเส้นหนึ่งคือสายสัญญาณนาฬิกาที่ใช้กำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัสไอสแควร์ซีทำได้ง่ายมากเพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหรดแอสเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอสเดรสของแต่ละตัว

สายข้อมูลบนบัสไอสแควร์ซีมีชื่อเรียกอย่างเป็นทางการว่า **สายข้อมูลอนุกรมหรือ SDA (Serial Data line)** ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า**สายสัญญาณนาฬิกาอนุกรม** หรือ **SCL (Serial clock line)** ในการอธิบายต่อไปนี้จะเรียกสายสัญญาณทั้งสองว่า**สาย SDA และ SCL**

2.3.1 คุณสมบัติโดยทั่วไปของบัสไอสแควร์ซี

สาย SDA และ SCLเป็นสายสัญญาณสองทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอับกับแรงดัน +5V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาส์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของอุปกรณ์ที่อยู่บนบัสไอสแควร์ซีต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collecotr)

อัตราการถ่ายทอดข้อมูลบนบัสไอสแควร์ซีสูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติและสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่อร่วมอยู่บนบัสไอสแควร์ซีจะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400pF การเข้าถึงอุปกรณ์บนบัสบนบัสไอสแควร์ซีใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-bit Addressing) หรือ 10 บิต (10-bit Addressing)

2.3.2 หลักการของบัสไอสแควร์ซี

บัสไอสแควร์ซีประกอบด้วยสายสัญญาณ 2 เส้นคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้

2.3.2.1 ข้อมูลพื้นฐานที่สำคัญของบัสไอสแควร์ซี

สภาวะที่เกิดขึ้นบนบัสไอสแควร์ซีมีด้วยกัน 5 สถานะ ดังนี้

2.3.2.1.1 บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสภาวะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่าการถ่ายทอดข้อมูลเริ่มขึ้นได้

2.3.2.1.2 เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (START)

2.3.2.1.3 ข้อมูลดำรงอยู่บนบัส(data valid) สภาวะที่เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือเป็น “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้เมื่อสาย SCL เป็นลอจิกต่ำ แต่มื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลที่สมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่ SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปรความหมายเป็นสถานะหยุดหรือสภาวะเริ่มต้นได้ ทำให้ข้อมูลที่เกิดการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1.4 รับรู้ข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์โดยตัวส่งจะทำการส่งข้อมูล 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่ง สัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ขณะนั้นก็จะกำเนิดบิตรับรู้ที่มีสถานะที่มีลอจิกต่ำที่ทำการตอบสนองให้ทราบว่าได้รับข้อมูลเรียบร้อยแล้ว

2.3.2.1.5 หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

2.3.2.2 การทำงานบนบัสไอสแควร์ซี

เริ่มต้นด้วยการเข้าถึงอุปกรณ์เสียก่อน โดยการเข้าถึงอุปกรณ์บนบัสไอสแควร์ซีนั้นจะใช้การเข้าถึงแบบ 7 หรือ 10 บิต ในกรณีที่มีอุปกรณ์ที่ต่ออยู่บนบัสไม่มาก ใช้การเข้าถึงแบบ 7 บิตเพียงพอ แต่ในบางอุปกรณ์ต้องใช้การเข้าถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มถ่ายทอดข้อมูลกันต่อไป

2.3.2.2.1 การเข้าถึงแบบ 7 บิต (7-bit Addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ โดยมีรูปแบบแสดงในรูป ใน 7 บิตบนรวมทั้งบิตMSB ด้วยจะเป็นข้อมูลแอสเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็นบิตกำหนดแอสเดรสคงที่ (fixed Address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตแอสเดรสที่สามารถโปรแกรมได้ (programmable Address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัสไอสแควร์ซีส่วนในบิต LSB เป็นบิตที่ใช้กำหนดอ่านหรือเขียนข้อมูลกับอุปกรณ์ตัวสเลฟนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้นเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่าบิตใดเป็นอินพุต บิดใดเป็นเอาส์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้การทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

ข้อมูลในไบต์ต่อมาคือข้อมูลที่ทำการถ่ายทอดจริง (data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ทำการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง

2.3.2.2.2 การเข้าถึงแบบ 10 บิต

จะมีข้อมูลเพิ่มเติมเข้ามาเล็กน้อย โดยในไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนเป็นข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอสเดรสของอุปกรณ์ที่ต้องการติดต่อในบิต LAB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอสเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็เป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกับการเข้าถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

อุปกรณ์ใดติดต่อกันอยู่และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะข้ออธิบายลักษณะหน้าที่และนิยามของอุปกรณ์ที่อยู่บนบัสไอสแควร์ซีเพื่อเป็นข้อตกลงก่อนอธิบายการทำงานของบัสไอสแควร์ซีต่อไป

- อุปกรณ์ที่เป็นผู้สร้างข้อมูลเรียกว่าตัวส่ง (transmitter)
- อุปกรณ์ที่เป็นตัวรับข้อมูลเรียกว่าตัวรับ (receiver) อุปกรณ์บนบัสไอสแควร์ซีสามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับอย่างเดียวจะไม่มีอุปกรณ์ใดบนบัสไอสแควร์ซีที่ทำหน้าที่เป็นตัวส่งเป็นอย่างเดียว
- อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัสไอสแควร์ซีเรียกว่า มาสเตอร์ (master)
- อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์หรือต่อพ่วงเข้าไปบนบัสไอสแควร์ซีเรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัสไอสแควร์ซีคือ

(1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

(2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้นสัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเชื่อมต่อภายนอกแบบเข้าจังหวะ (Synchronous Peripheral Interface)

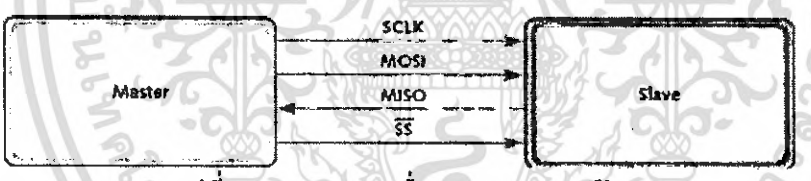
เป็นกระบวนการติดต่อสื่อสารระหว่างอุปกรณ์ ด้วยอัตราการสื่อสารที่มีความเร็วต่ำ คล้ายการติดต่อสื่อสารแบบไอสแควร์ซีแต่มีลักษณะ Full-Duplex คือสามารถรับและส่งพร้อมกันได้ และมีความเร็วสูงกว่า

แนะนำการสื่อสารแบบไอสแควร์ซีเพื่อเป็นพื้นฐานและในการทำความเข้าใจง่ายขึ้นไอสแควร์ซีเป็นการสื่อสารระหว่างอุปกรณ์ แบบ Master-Slave ที่ใช้สายเพียง 2 เส้น คือ

- SDA เป็นสายส่งข้อมูลทั้งไปและกลับดังนั้นจึงสามารถรับหรือส่งได้ไม่พร้อมกัน (Half-Duplex)
- SCL เป็นสายส่งสัญญาณนาฬิกาเพื่อทำให้ทั้ง Master-Slave เข้าใจตรงกันว่าสามารถอ่านข้อมูลบนสาย SDA ได้ (Synchronous Clock)

แต่จุดด้อยของไอสแควร์ซีคือการที่ใช้ แอดเดรส ของตัวอุปกรณ์ส่งไปตามสาย SDA มีขนาด 8 บิตทำให้จำกัดอุปกรณ์ที่ใช้ต่อพ่วง เป็น slave ได้แตกต่างกันมากที่สุดเพียง 256 อุปกรณ์

เอสพีไอ เป็นการสื่อสารแบบ Master-Slave เหมือนไอสแควร์ซีแต่ใช้สัญญาณ ในระบบจำนวน 4 เส้น คือ



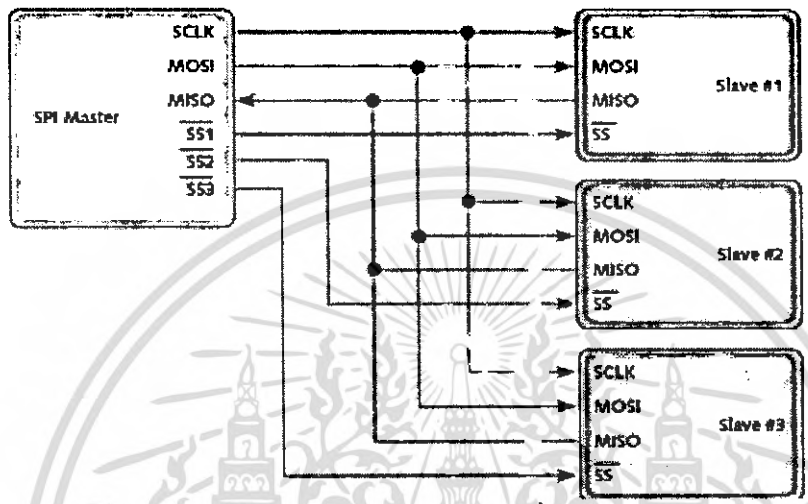
รูปที่ 2.31 แสดงการสื่อสารแบบ เอสพีไอ

- **SCLK** เป็นสายสัญญาณนาฬิกาเพื่อ Synchronous เช่นเดียวกับไอสแควร์ซีส่งออกทาง Master
- **MOSI** (Master-out Slave-in) ตามชื่อเป็นสายสัญญาณที่ใช้ส่งอข้อมูลจาก Master ไปหา Slave
- **MISO** (Master-in Slave-out) เป็นสายสัญญาณที่ Slave ใช้ส่งข้อมูลกลับมาหา Master
- **\SS** (Slave Select) เป็นขาที่ทำงานที่ low-level ที่ใช้เลือกว่า Slave ตัวใดที่มีสิทธิ์ติดต่อกับ Master ที่ Master จะไม่มีขานี้ แต่ใช้ขาใดก็ได้ใช้ส่งสัญญาณ low-level ออกมาไปที่ Slave ที่ต้องการ การออกแบบที่ง่ายที่สุดคือ ถ้ามี Slave 5 ตัว Master จะต้องมี pin ทำหน้าที่นี้ 5 pin เพื่อเลือก Slave หรือใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AND Gate สร้างเป็นระบบ แอดเดรส ก็ได้จะประหยัดขาสัญญาณกว่า แต่เปลืองเกต

สังเกตเห็นว่า เอสพีไอ จะมีการรับส่งข้อมูล 2 เส้นทำให้สามารถทำงานใน โหมด Full-Duplex ได้ และสามารถต่อ Slave พ่วงได้มากเพราะใช้การเลือก Slave ที่ต้องการสื่อสารด้วย



รูปที่ 2.32 แสดงการต่อพ่วงสลาฟของการสื่อสารแบบ เอสพีไอ

เอสพีไอ สามารถปรับแต่งการทำงานของการเข้าจังหวะของสัญญาณนาฬิกาได้โดยใช้ CPOL (Clock Polarity) และ CPHA (Clock Phase) ให้ทำการรับข้อมูลที่ low-level หรือ high-level ได้

วิธีการสื่อสารโดย เอสพีไอ ทำได้โดย Master ทำการส่งสัญญาณ low-level ไปเลือก Slave ก่อนแล้วสร้าง Clock ออกไปเรื่อย พร้อมๆ กับ ข้อมูลบน สาย MOSI ให้ Slave รับข้อมูลนั้นเข้าไป หากต้องการรับข้อมูลจาก Slave Master จะส่ง Clock ออกไปรอแต่สัญญาณใน MOSI จะเป็น 0x00 แล้ว รอรับข้อมูลจากทาง MISO ที่ Slave จะส่งมาให้ เมื่อได้รับข้อมูลแล้ว เอสพีไอ จะทำการเปลี่ยน Status flag ของ เอสพีไอ และมี interrupt 9 แต่ เอสพีไอ ไม่มีการรับประกันว่าข้อมูลส่งถึงผู้รับเหมือนไอสแควร์ซีที่มีการตอบ ACK or NAK หากต้องการทำการรับประกันต้องไปหาช่องทางอื่นมาช่วย เอสพีไอ ให้สามารถรู้สถานะได้

```
//Master
#include "reg_c51.h"
char serial_data;
char data_example=0x55;
char data_save;
bit transmit_completed= 0;

void it_SPI(void) interrupt 9 /* interrupt Address is 0x004B */
{
       switch( SPSTA )
       {
              case 0x80:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                serial_data=SPDAT;
                transmit_completed=1;
                break;
                case 0x10:
                        /* put here for mode fault tasking */
                break;
                case 0x40:
                        /* put here for overrun tasking */
                break;
        }
}
void main(void)
{
        SPCON |= 0x10;          /* Master Mode */
        P1_1=1;                 /* enable master */
        SPCON |= 0x82;          /* Fclk Periph/128 */
        SPCON &= ~0x08;         /* CPOL=0; transmit mode example */
        SPCON |= 0x04;          /* CPHA=1; transmit mode example */
        IEN1 |= 0x04;           /* enable spi interrupt */
        SPCON |= 0x40;          /* run spi */
        EA=1;           /* enable interrupts */
        while(1)
        {
                /* Send MOSI */
                SPDAT=data_example;
                while(!transmit_completed); /* wait end of transmition */
                transmit_completed = 0;
                /* Receive MISO */
                SPDAT=0x00;  /* data is send to generate SCK signal */
                while(!transmit_completed); /* wait end of transmition */
                transmit_completed = 0;
                data_save = serial_data;  /* save receive data */
        }
}
```

//Slave

```
#include "reg_c51.h"
bit transmit_completed;
char serial_data;
void it_SPI(void) interrupt 9
{
        switch( SPSTA ) /* spi status register */
        {
                case 0x80:
                        serial_data=SPDAT;  /* read receive data */
                        transmit_completed=1;       /* set software flag */
                        break;
                case 0x10:
                        /* put here for mode fault tasking */
                        break;
                case 0x40:
                        /* put here for overrun tasking */
                        break;
        }
        SPDAT=serial_data; /* needed to complete clearing sequence */
}
void main(void)
{
        SPCON &= ~0x10;         /* slave mode */
        SPCON &= ~0x08;         /* CPOL=0; transmit mode example*/
        SPCON |= 0x04;          /* CPHA=1; transmit mode example*/
        IEN1 |= 0x04;
        SPCON |= 0x40;
        transmit_completed = 0;
        EA=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while(1)
        {
                if(transmit_completed)
                {
                        SPDAT = serial_data; /* echo data to master */
                        transmit_completed = 0;
                }
        }
}
```

รูปที่ 2.33 แสดงตัวอย่างซอร์สโค้ดการทำ เอสพีไอ

2.5 การเชื่อมต่อแบบยูเอสบี (Universal Serial Bus Interface)

USB ย่อมาจาก Universal Serial Bus แปลแบบตรงความหมายจะได้ว่า "บัสอนุกรมอเนกประสงค์" คุณสมบัติต่างๆที่ทำให้สามารถกำหนดชื่อนี้ให้กับพอร์ตชนิดนี้ได้มีดังนี้

- ใช้คอนเน็กเตอร์เพียงชนิดเดียวซึ่งมี 2 รูปแบบสำหรับเชื่อมต่ออุปกรณ์ทุกๆชนิดเข้ากัดคอมพิวเตอร์

- สามารถเชื่อมอุปกรณ์หลายๆชนิดรวมเข้าสู่คอนเน็กเตอร์ตัวเดียวสูงสุด 127 ตัว

- ไม่เกิดการชัดแย้งกันของการเข้าใช้ทรัพยากรของระบบ (IRQ)

- ตรวจสอบการเชื่อมต่อและตั้งค่าทำงานต่างๆ อัตโนมัติ ระหว่างที่เครื่องกำลังทำงานอยู่ (hot attachment)

- ความเร็วในการถ่ายโอนข้อมูลจะขึ้นอยู่กับมาตรฐาน ดังนี้

-มาตรฐาน ยูเอสบี 1.0/1.1 : มีอัตราในการถ่ายโอนข้อมูลความเร็วต่ำ (low speed)เท่ากับ 1.5 Mbit/sec และความเร็วเต็ม (full speed) เท่ากับ 12 Mbit/sec

-มาตรฐาน ยูเอสบี 2.0 : มีอัตราในการถ่ายโอนเพิ่มขึ้นอีก 1 ระดับคือ ความเร็วสูง (high speed)เท่ากับ 480 Mbit/sec

- ที่ขาพอร์ต ยูเอสบี มีแรงดันไฟตรง +5V จ่ายออกมาด้วย ทำให้อุปกรณ์ต่อพ่วงที่ใช้พลังงานไม่มากนัก สามารถใช้แรงดันจากพอร์ต ยูเอสบี นี้เป็นไฟเลี้ยงทำงานได้ โดยไม่ต้องอาศัยแหล่งจ่ายไฟเพิ่มเติมจากภายนอก

2.5.1 การส่งข้อมูลภายในบัส ยูเอสบี 1.0/1.1

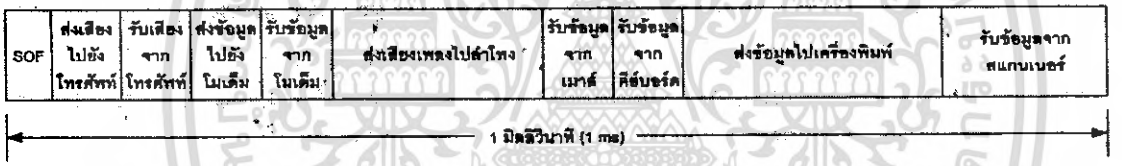
ยูเอสบี เป็นการส่งข้อมูลที่มีรูปแบบการเชื่อมต่อในระบบบัสคือ *อุปกรณ์ทุกๆตัวจะต้องส่งสัญญาณรวมกันไปในสายส่งสัญญาณเพียงคู่เดียว* ดังนั้นอุปกรณ์ทุกตัวที่เชื่อมต่อกับบัสจะต้องส่งข้อมูลเรียงลำดับกันไปเพื่อไม่ให้เกิดการชนกันของข้อมูล และเนื่องจากยูเอสบีเป็นระบบบัสที่ใช้สายส่งสัญญาณเพียงคู่เดียว (2 เส้น) ทำให้ช่วงเวลาหนึ่งๆ จะมีข้อมูลวิ่งไปได้ในเพียงทิศทางเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่านั้นไม่สามารถรับส่งข้อมูลไปได้ในเวลาเดียวกันได้ หรือที่เรียกว่า *การส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (half-duplex)*

จังหวะการรับส่งข้อมูลของระบบบัส ยูเอสบี ทั้งหมดจะควบคุมจาก โฮสต์ ซึ่งก็คือ เครื่องคอมพิวเตอร์ที่เป็นจุดรวมของอุปกรณ์ทุกตัวที่เชื่อมต่ออยู่นั่นเอง ดังนั้นจึงไม่สามารถเชื่อมต่อเครื่องคอมพิวเตอร์ 2 เครื่องให้รับส่งข้อมูลถึงกันได้โดยตรง เพราะถ้าเครื่องคอมพิวเตอร์ 2 เครื่องทำหน้าที่เป็นโฮสต์ จะเกิดการชนกันของข้อมูลภายในบัส เนื่องจากแต่ละเครื่องก็พยายามกำหนดจังหวะในการรับส่งของตัวเองขึ้นมา ดังนั้นจะเชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกันจะต้องมีอุปกรณ์ตัวกลางเพื่อทำการซิงโครไนซ์ตัวเองให้เข้ากับโฮสต์ทั้งสองให้ได้

การรับส่งข้อมูลจะถูกกำหนดเป็นเฟรม โดยทุกๆ 1 มิลลิวินาที จะเกิดการรับส่งข้อมูลขึ้น 1 เฟรม ในแต่ละเฟรมจะแบ่งย่อยออกเป็นแพ็กเก็ต (packet) เริ่มต้นของแต่ละเฟรมโดยโฮสต์จะส่งสัญญาณเริ่มต้นเฟรมหรือ SOF (Start Of Frame) ออกไปเพื่อให้อุปกรณ์ทุกตัวรู้จังหวะการเริ่มเฟรม หลังจากนั้น โฮสต์ก็จะเริ่มส่งหรือรับข้อมูลต่างๆตามที่ได้จัดลำดับความสำคัญไว้อุปกรณ์ต่างๆ ที่อยู่ภายในบัสจะต้องทำงานตามจังหวะที่โฮสต์กำหนดไว้เท่านั้น การส่งข้อมูลกลับไปยังโฮสต์จะสามารถทำได้ก็ต่อเมื่อได้รับการถามหรือร้องขอจากโฮสต์



รูปที่ 2.34 แสดงการจัดลำดับการรับส่งข้อมูลของอุปกรณ์แต่ละตัว

แต่เนื่องจากแต่ละเฟรมข้อมูลจะต้องรับส่งเสร็จภายใน 1 มิลลิวินาที หมายความว่าข้อมูลของอุปกรณ์ทุกๆตัวที่เชื่อมต่อกับบัสจะต้องถูกกำหนดขนาดไม่ให้ใหญ่เกินกว่าที่จะสามารถรับส่งได้ภายใน 1 มิลลิวินาที และเล็กพอที่จะทำให้อุปกรณ์ทุกๆตัวสามารถใช้งานบัสไปพร้อมๆกันได้ ดังนั้นในระบบบัสยูเอสบี จึงจำเป็นต้องอาศัยซอฟต์แวร์เข้ามาจัดการในด้านนี้ และยังต้องอาศัยฮาร์ดแวร์ที่จะคอยกระจายการส่งและรวบรวมการรับส่งข้อมูลจากทุกอุปกรณ์ทุกๆ ตัวในระบบ ซึ่งซอฟต์แวร์และฮาร์ดแวร์ที่จำเป็นสำหรับระบบยูเอสบี มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ส่วนซอฟต์แวร์</u>

1.ไดรเวอร์อุปกรณ์ยูเอสบี (USB device drivers)

2.ไดรเวอร์ยูเอสบี (USB driver)

3.ไดรเวอร์โฮสต์คอนโทรลเลอร์ (USB host controller driver)

<u>ส่วนฮาร์ดแวร์</u>

1.ยูเอสบีโฮสต์คอนโทรลเลอร์ (USB host controller) / รูตฮับ (root hub)

2.ยูเอสบีฮับ (USB hub)

3.อุปกรณ์ยูเอสบี (USB device)

2.5.2 ส่วนประกอบทางซอฟต์แวร์

2.5.2.1 ไดรเวอร์อุปกรณ์ยูเอสบี (USB Device Driver)

ไดรเวอร์อุปกรณ์ยูเอสบี คือ โปรแกรมเก็บข้อมูลที่จำเป็นในการติดต่อไปยังอุปกรณ์แต่ละตัว เมื่อโปรแกรมใดมีความต้องการที่จะติดต่อกับอุปกรณ์ต่างๆ จะต้องแจ้งความต้องการนั้นๆมายังไดรเวอร์อุปกรณ์ยูเอสบี เนื่องจากตัวไดรเวอร์นี้จะรู้ว่าถ้าต้องการติดต่อกับอุปกรณ์จะต้องติดต่อผ่านเอ็นด์พอยต์ (Endpoint) ไหน ด้วยรูปแบบใด (การทำงานของอุปกรณ์ยูเอสบี จะติดต่อสั่งงานผ่านเอ็นด์พอยต์ของตัวอุปกรณ์ซึ่งอุปกรณ์ต่างๆจะมีชนิดและจำนวนเอ็นด์พอยต์ที่แตกต่างกัน รายละเอียดจะกล่าวถึงในบทต่อๆไป) ดังนั้นอุปกรณ์แต่ละตัวก็จะมีไดรเวอร์อุปกรณ์ยูเอสบีเฉพาะตัว ซึ่งเมื่อถึงคราวต้องนำอุปกรณ์นั้นมาต่อใช้งานกับเครื่องคอมพิวเตอร์จริงๆ ก็จะต้องนำไดรเวอร์ตัวเดียวกันมาติดตั้งเพิ่มเข้ากับระบบปฏิบัติการในเครื่องคอมพิวเตอร์เพื่อให้ระบบรู้จักและติดต่อกับอุปกรณ์ที่ติดตั้งเข้ามาใหม่ได้ เช่น ถ้าต้องการติดต่อเพื่อรับข้อมูลจากคีย์บอร์ด ตัวไดรเวอร์อุปกรณ์ยูเอสบี จะรู้ว่าต้องรับส่งข้อมูลด้วยอัตราเร็วต่ำ (slow speed) โดยใช้รูปแบบการถ่ายทอดข้อมูลแบบอินเทอร์รัป (interrupt transfer type) ผ่านเอ็นด์พอยต์ตัวหนึ่งของคีย์บอร์ต และตรวจสอบข้อมูลการเกิดเป็นช่วงระยะห่างค่าหนึ่ง

แต่ในบางอุปกรณ์ที่เป็นอุปกรณ์พื้นฐานของเครื่องคอมพิวเตอร์ เช่น เมาส์และคีย์บอร์ด จะมีการบรรจุไดรเวอร์อุปกรณ์ยูเอสบี ของอุปกรณ์เหล่านี้ภายในไบออสของเครื่องคอมพิวเตอร์เรียบร้อยแล้ว จึงไม่ต้องติดตั้งไดรเวอร์เพิ่มเติมสำหรับอุปกรณ์เหล่านี้ เพียงแต่เข้าไปเปิดการทำงานไบออสก็จะทำให้เครื่องคอมพิวเตอร์รู้จักอุปกรณ์เหล่านี้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

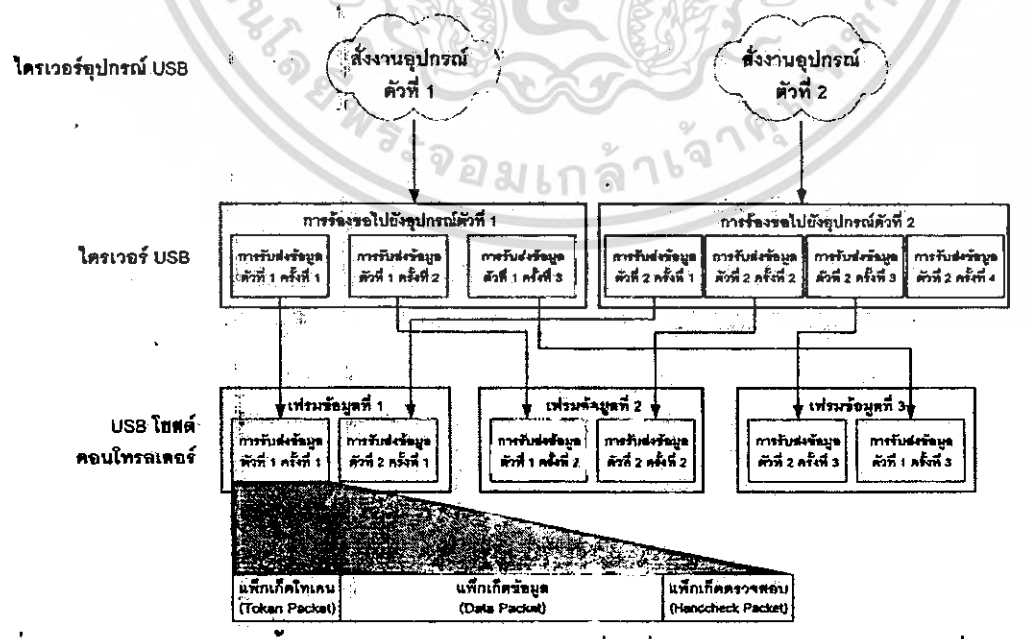
2.5.2.2 ไดรเวอร์ยูเอสบี (USB Driver)

การทำงานยูเอสบี นั้นเป็นการต่อร่วมกันของอุปกรณ์หลายๆชนิดบนสายสัญญาณเพียงคู่เดียว ดังนั้นการส่งข้อมูลของอุปกรณ์แต่ละชนิดจะต้องมีการแบ่งสันปันส่วนกันอย่างพอเหมาะพอดี เพื่อให้อุปกรณ์ทุกตัวสามารถทำงานไปพร้อมๆกัน และแน่นอนว่าต้องมีซอฟต์แวร์ที่เข้ามาทำหน้าที่นี้คือ ไดรเวอร์ ยูเอสบี นั่นเอง

ไดรเวอร์อุปกรณ์ยูเอสบี ของอุปกรณ์แต่ละตัวจะส่งการร้องขอเพื่อการติดต่อ (request) ลงมายังไดรเวอร์ยูเอสบี และเมื่อไดรเวอร์ยูเอสบี รับทราบความต้องการติดต่อของอุปกรณ์ครบทุกๆตัวที่เชื่อมต่ออยู่กับบัสแล้ว ก็จะพิจารณาว่ารอบการรับส่งข้อมูลหนึ่งๆนั้นอุปกรณ์แต่ละตัวสามารถรับส่งข้อมูลได้มากเท่าใด หากปริมาณข้อมูลที่ต้องการรับส่งมีขนาดเล็กมากก็จะตัดออกเป็นส่วนๆแล้วเก็บไว้รอส่งในรอบถัดไปโดยปริมาณที่ส่งได้ของอุปกรณ์แต่ละตัวจะถูกพิจารณาจากชนิดของการถ่ายทอดข้อมูล (transfer type) ว่าอุปกรณ์ใดใช้การถ่ายทอดข้อมูลแบบใดและการรับส่งข้อมูลชนิดนั้นมีลำดับความสำคัญมากน้อยเพียงใด

2.5.2.3 ไดรเวอร์คอนโทรลเลอร์ (Controller Driver)

หลังจากไดรเวอร์ยูเอสบี พิจารณาแล้วว่าอุปกรณ์แต่ละตัวส่งข้อมูลได้เท่าใดบ้าง มันจะส่งข้อมูลของอุปกรณ์แต่ละตัวที่ติดต่อในรอบการติดต่อนั้นๆ มายังไดรเวอร์โฮสต์คอนโทรลเลอร์ จากนั้นโฮสต์คอนโทรลเลอร์จะจัดเรียงลำดับข้อมูลของอุปกรณ์แต่ละชนิดลงเป็นเฟรมข้อมูล เพิ่มเติมส่วนประกอบต่างๆของเฟรมให้ครบตามมาตรฐานการถ่ายทอดข้อมูลแบบยูเอสบี แล้วส่งข้อมูลนั้นไปยังฮาร์ดแวร์ยูเอสบี โฮสต์คอนโทรลเลอร์เพื่อส่งข้อมูลทั้งหมดออกไปยังอุปกรณ์ต่างๆ ในรูป 2.35 แสดงลำดับและขั้นตอนการทำงานของซอฟต์แวร์ควบคุมการทำงานของพอร์ตยูเอสบี



รูปที่ 2.35 แสดงลำดับและขั้นตอนการทำงานของซอฟต์แวร์ควบคุมการทำงานของพอร์ตยูเอสบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 ส่วนประกอบทางฮาร์ดแวร์ของยูเอสบี

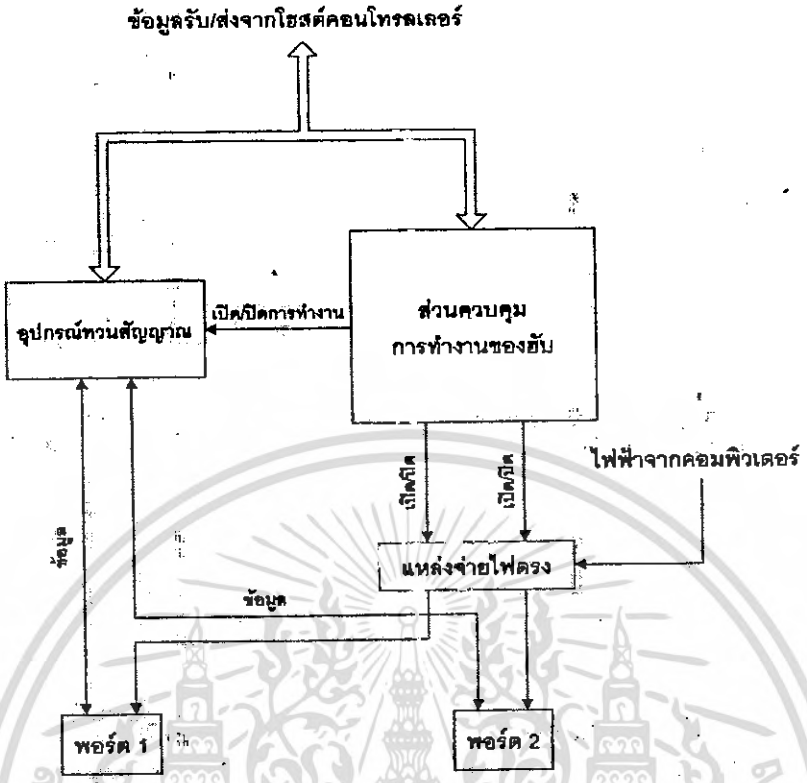
จากส่านประกอบ 3 ส่วนที่ผ่านมาเป็นส่วนประกอบด้านซอฟต์แวร์ ซึ่งจะคอยควบคุมจัดการการทำงานของอุปกรณ์ที่มาต่อร่วมกันทั้งหมดให้เป็นไปอย่างราบรื่น ในส่วนถัดไปจะเป็นหน้าที่ของส่วนประกอบทางฮาร์ดแวร์ที่ทำหน้าที่รับส่งรับส่งสัญญาณกับตัวอุปกรณ์ต่างๆ โดย ส่วนประกอบตัวแรกที่จะกล่าวถึงคือ ยูเอสบีโฮสต์คอนโทรลเลอร์ (USB controller) และ รูตฮับ (USB root hub)

2.5.3.1 ยูเอสบีโฮสต์คอนโทรลเลอร์ / ยูเอสบีรูตฮับ

ยูเอสบีโฮสต์คอนโทรลเลอร์มีหน้าที่สร้างสัญญาณข้อมูลทางไฟฟ้า แล้วส่งต่อไปยังรูตฮับเพื่อกระจายออกไปยังอุปกรณ์ต่างๆ โดยมันจะสร้างสัญญาณข้อมูลการติดต่อรูปแบบต่างๆตามที่ไดรเวอร์โฮสต์คอนโทรลเลอร์กำหนดมาให้ จากนั้นแปลงข้อมูลที่จะส่งจากแบบขนานเป็นแบบอนุกรมเพื่อใช้ในการส่งต่อไป เมื่อสัญญาณที่ต้องการส่งมาถึงรูตฮับ รูตฮับจะส่งสัญญาณนั้นออกไปยังบัสเพื่อส่งต่อไปยังอุปกรณ์ต่างๆ นอกจากนั้นรูตฮับยังมีหน้าที่สำคัญอีก 4 อย่างคือ

1. ควบคุมการใช้พลังงานของอุปกรณ์ที่นำมาต่อ
2. ตรวจสอบการเชื่อมต่อของอุปกรณ์ว่ามีอุปกรณ์ต่ออยู่หรือไม่
3. เปิดหรือเอเนเบิลการใช้งานพอร์ตเมื่อมีอุปกรณ์ต่ออยู่ และปิดหรือดิสเอเบิลการใช้งานเมื่อปลดอุปกรณ์ออกไปแล้ว
4. รายงานสถานะของแต่ละพอร์ตเมื่อไดรเวอร์โฮสต์คอนโทรลเลอร์ร้องขอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 แสดงไดอะแกรมแสดงการทำงานของรูตฮับอย่างง่าย

2.5.3.2 ยูเอสบีฮับ

หน้าที่หลักๆของยูเอสบีฮับ คือ การขยายการเชื่อมต่อให้อุปกรณ์จำนวนมากๆสามารถเชื่อมต่อเข้ากับระบบบัสได้ โดยหลักการทำงานของยูเอสบีฮับนั้นมีอยู่ 2 ส่วนคือ ทำหน้าที่เป็นตัวทวนสัญญาณ (repeater) และตัวจัดการพลังงาน (power managerment)

ในส่วนของการทวนสัญญาณยูเอสบีฮับจะต้องรับสัญญาณจากโฮสต์มา แล้วส่งกระจายออกไปยังพอร์ตทุกๆพอร์ต และรับสัญญาณจากแต่ละพอร์ต แล้วจับมารวมกันเพื่อส่งกลับไปที่โฮสต์ สำหรับส่วนของการจัดการพลังงานนั้นมีหน้าที่เหมือนรูตฮับก็คือ ตรวจสอบว่ามีการต่ออยู่ของอุปกรณ์ที่พอร์ตใดบ้าง หากมีอุปกรณ์ต่ออยู่ก็เปิดการใช้งานพอร์ตนั้นๆ หากไม่มีอุปกรณ์ตอบอยู่ก็ปิดการใช้งาน ตรวจสอบการเชื่อมต่อหรือปลดออกของอุปกรณ์เพื่อรายงานผลเมื่อโฮสต์คอนโทรลเลอร์ร้องขอ และป้องกันอุปกรณ์ที่ต่ออยู่ในแต่ละพอร์ตไม่ให้ดึงกระแสไฟฟ้าเกินกว่าที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.3 อุปกรณ์ยูเอสบี

ส่วนประกอบทางฮาร์ดแวร์ส่วนสุดท้ายที่ต้องรู้จักคือ อุปกรณ์ยูเอสบีซึ่งอุปกรณ์ต่างๆที่เชื่อมต่อกับคอมพิวเตอร์ด้วยพอร์ตยูเอสบีนั่นเอง สามารถแบ่งเป็น 2 ชนิดตามความเร็วในการถ่ายทอดข้อมูลคือ

1. อุปกรณ์ความเร็วต่ำ (low-speed device) ถ่ายทอดข้อมูลด้วยความเร็ว 1.5 Mbit/sec

2. อุปกรณ์ความเร็วเต็มที่ (full-speed device) ถ่ายทอดข้อมูลด้วยความเร็ว 12 Mbit/sec

ปัจจุบันนี้อุปกรณ์ยูเอสบี ที่มีจำหน่ายอยู่ตามท้องตลาดมีอยู่เป็นจำนวนมาก เช่น คีย์บอร์ด, เมาส์, จอยสติ๊ก เหล่านี้คืออุปกรณ์ยูเอสบี ความเร็วต่ำ ส่วนจอมอนิเตอร์, ลำโพง, เครื่องพิมพ์, กล้องถ่ายรูปดิจิตอล, ซีดีรอมไดรฟ์, เครื่องเล่น เอ็มพีสาม จัดเป็นอุปกรณ์ยูเอสบี ความเร็วสูง อุปกรณ์บางตัวจะบรรจุความสามารถของยูเอสบีฮับเข้าไปด้วย ทำให้สามารถนำอุปกรณ์อื่นๆมาเชื่อมต่อได้เหมือนกับการต่อเข้ากับฮับ อุปกรณ์ลักษณะนี้เรียกว่า Computer USB Device ตัวอย่างของอุปกรณ์ที่มีฮับอยู่ภายในได้แก่ จอมอนิเตอร์ หรือเครื่องพิมพ์

นอกจากการแบ่งชนิดของอุปกรณ์ตามความเร็วในการถ่ายทอดข้อมูลแล้วอาจจะแบ่งกลุ่มตามการใช้พลังงานของตัวอุปกรณ์เองก็ได้ ซึ่งสามารถแบ่งได้อีก 2 ชนิด คือ

1. *อุปกรณ์จ่ายไฟเลี้ยงจากบัส* (Bus powered device) คือ อุปกรณ์ที่ใช้ไฟเลี้ยงจากบัสโดยตรง ไม่ต้องมีแหล่งจ่ายภายนอกเพิ่มเติม

2. *อุปกรณ์ที่ใช้ไฟเลี้ยงจากตัวเอง* (Self powered device) คือ อุปกรณ์ที่มีแหล่งจ่ายไฟในตัวไม่ต้องอาศัยไฟเลี้ยงจากบัส

2.5.4 การส่งสัญญาณในบัสยูเอสบี

สัญญาณที่ปรากฏบนสายสัญญาณระหว่าง รูตฮับและอุปกรณ์จะส่งไปในแบบสัญญาณผลต่าง (Differential signaling) เพื่อการแพร่กระจายสนามแม่เหล็กไฟฟ้า (EMI:Electromagnetic Interference) เนื่องจากตามธรรมชาติของสัญญาณไฟฟ้า เมื่อมีการเปลี่ยนแปลงระดับสัญญาณด้วยความเร็วมากๆ(ซึ่งก็คือการส่งข้อมูลที่มีความเร็วมากๆ) จะทำให้เกิดการแพร่กระจายของสนามแม่เหล็กและสนามไฟฟ้าออกมารอบๆสายส่งสัญญาณ ซึ่งอาจรบกวนการทำงานของอุปกรณ์ต่างๆ รอบข้างได้ ด้วยการส่งสัญญาณแบบนี้จะทำให้การเปลี่ยนแปลงของสัญญาณในสายนำสัญญาณเกิดขึ้นพร้อมกันและเกิดในลักษณะตรงกันข้าม ทำให้สนามแม่เหล็กหักล้างกัน ไม่แพร่ออกมาภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 กระบวนการกำหนดการทำงานของอุปกรณ์

เมื่อเริ่มการทำงานของยูเอสบี หรือเมื่อมีการเชื่อมต่ออุปกรณ์ตัวใหม่เข้ามาในระบบ ลำดับขั้นตอนคร่าวๆจะเป็นดังนี้

2.5.5.1 ฮับตรวจสอบพบว่ามีการเชื่อมต่ออุปกรณ์ตัวใหม่เข้าสู่ระบบ แล้วแจ้งผลกลับไปยังโฮสต์คอนโทรลเลอร์

2.5.5.2 โฮสต์คอนโทรลเลอร์สั่งให้ฮับเปิดการทำงานของแหล่งจ่ายไฟในโหมดประหยัด เพื่อให้อุปกรณ์ที่อาศัยพลังงานจากบัสสามารถทำงานได้

2.5.5.3 โฮสต์คอนโทรลเลอร์สั่งให้ฮับรีเซตพอร์ตที่อุปกรณ์ตัวใหม่มาเชื่อมต่อเพื่อให้อุปกรณ์รีเซตค่าแอดเดรสและเอนด์พอยต์ของตัวเองให้เป็นค่าเริ่มต้น (default)

2.5.5.4 โฮสต์อ่านดิสคริปเตอร์ต่างๆจากตัวอุปกรณ์และพิจารณาว่าทรัพยากรของระบบพอเพียงต่อความต้องการของอุปกรณ์หรือไม่ ทรัพยากรในที่นี้คือ พลังงานไฟฟ้าและปริมาณข้อมูลที่จะส่งของตัวอุปกรณ์ หากพิจารณาแล้วว่าไม่สามารถทำงานได้ก็จะสั่งให้ฮับปิดการทำงานของพอร์ตนั้น

2.5.5.5 เมื่อโฮสต์พิจารณาแล้วว่าสามารถให้บริการแก่อุปกรณ์ตัวที่มาเชื่อมต่อได้จะควบคุมให้แหล่งจ่ายไฟจ่ายพลังงานตามที่อุปกรณ์ต่างๆต้องการรวมไปถึงการตั้งค่าแอดเดรสและกำหนดค่าต่างๆ

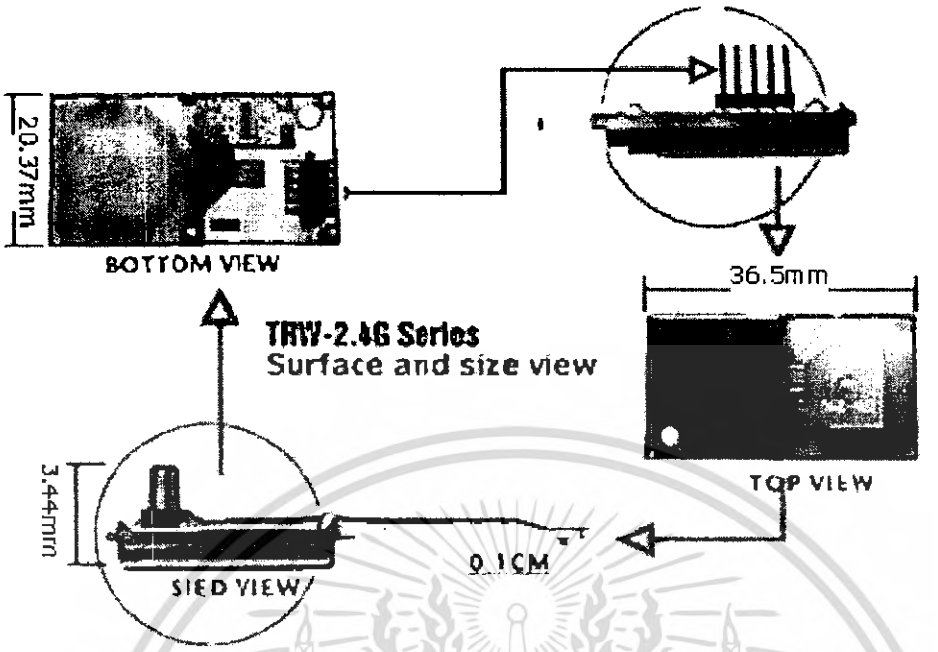
2.5.5.6 หลังจากตั้งค่าเรียบร้อย คอมพิวเตอร์จะรู้จักกับอุปกรณ์ตัวใหม่และสามารถติดต่อกับอุปกรณ์ได้ทันทีโดยไม่ต้องทำการรีบูตเครื่องใหม่

2.6 การใช้งานส่วนการติดต่อผ่านคลื่นสัญญาณวิทยุด้วยโมดูล TRW-2.4G

ในส่วนของการติดต่อสื่อสารระหว่างตัว Portable กับ Station หลังจากที่เราได้เปรียบเทียบข้อดี-ข้อเสีย ของแต่ละเทคโนโลยีที่ใช้สื่อสารไร้สายปรากฏว่าเทคโนโลยี RF เป็นเทคโนโลยีที่เหมาะสมในโปรเจคนี้ที่สุด ทั้งในด้านราคา ด้านความง่ายในการใช้งานและการจัดซื้อ

TRW-2.4G เป็น EF Module ที่มีขายในประเทศไทยใช้ความถี่ 2.4-2.524 GHz สื่อสาร RF ด้วยเทคนิค FSK (Frequency Shift keying) มีความเร็วในการส่งข้อมูล 250 กิโลบิตต่อวินาที ในช่วง 280 m แต่มีความเร็วสูงขึ้นเป็น 1 เมกะบิตต่อวินาที ในระยะทำงาน 150 m ใช้ไฟล์เสียง Module ≈ 3V สามารถตั้งค่าให้เป็น transmitter หรือ Receiver ก็ได้และมี mode การทำงานแบบ ShochBurst แบบ Duo Ceiver มี Antenna ภายใน Moduleราคาประมาณ 400 กว่าบาท ที่ ES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



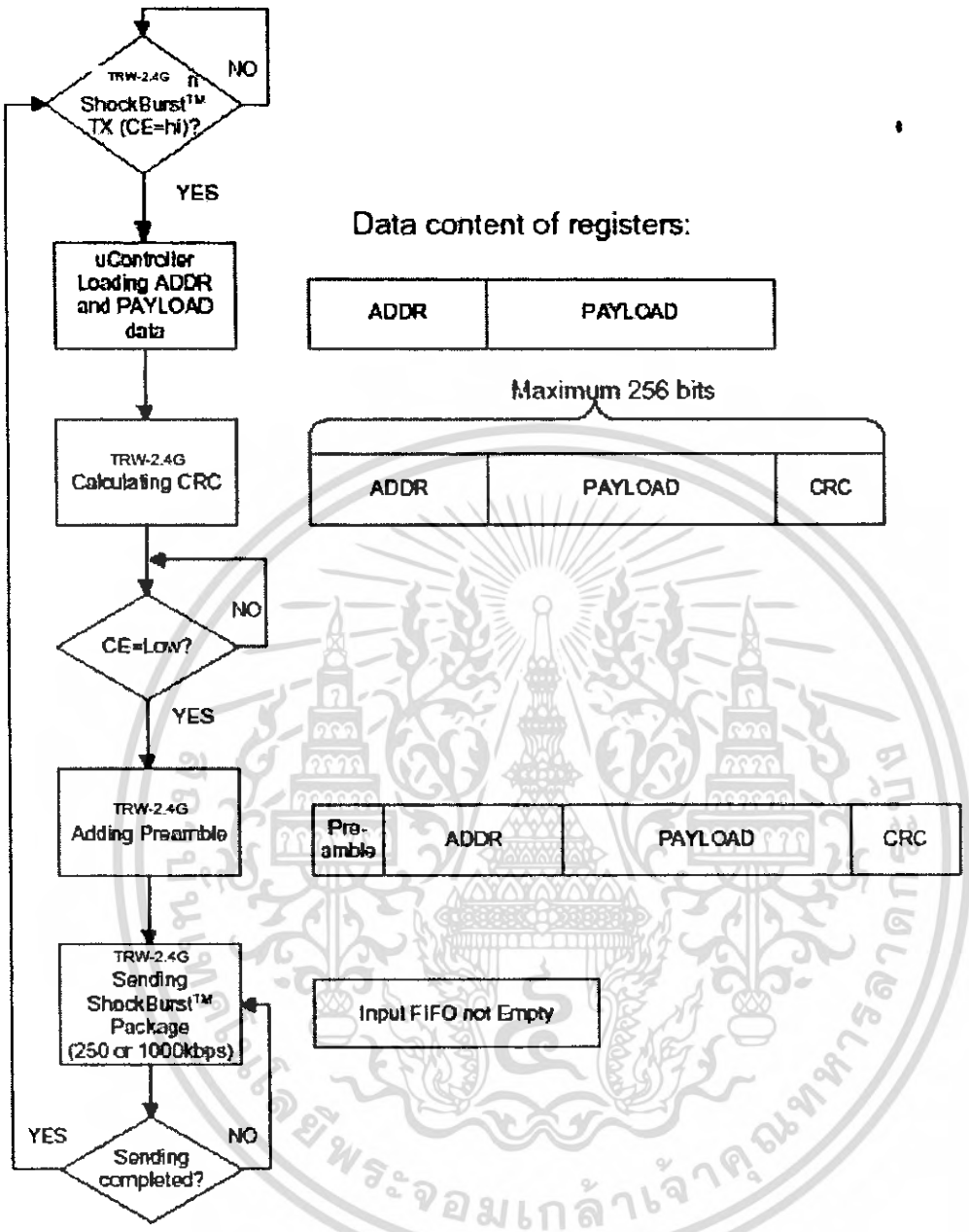
รูปที่ 2.37 แสดงลักษณะของตัวโมดูล TRW-2.4G

การทำงานของ โหมด ShockBurst

ต้องการส่งข้อมูล (ใช้ Pin CE, CLK1, DATA)

- สั่งPin CE = 1 (เปิด โหมด ShockBurst)
- สั่งข้อมูล(แอดเดรส และ PAYLOAD) ทาง pin DATA โดยใช้ CLK1 เป็นตัว Synchronous Data ที่ส่งให้ TRW-2.4G
- TRW-2.4G จะคำนวณ CRC ให้(ขนาดตามที่เรา config ไว้)
- สั่ง pin CE = 0 (ปิด โหมด ShockBurst) TRW-2.4G ใส่ Preamble และเริ่มส่ง RF จนกว่าจะเสร็จแล้วกลับมา Standby

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 แสดงไดอะแกรมการส่งข้อมูลในโหมด ShockBurst ของตัวโมดูล TRW-2.4G

มีข้อมูลเข้ามาที่ตัวรับ (ใช้ Pin CE, DRI, CLK1, DATA)

- เมื่อมีข้อมูลเข้ามาที่ตัวรับ TRW-2.4 จะทำการตรวจสอบ ADDR ที่ส่งมาค่าตรงกับของตัวมันที่เรา Config ไว้หรือไม่ หากตรงจะทำงานต่อ หากไม่ตรงก็จะเพิกเฉยรอ package ใหม่
- ทำการ Check CRC ที่ส่งมาว่าถูกต้องหรือไม่ ถ้าถูกทำงานต่อหากผิดจะไม่มีการทำงานตอบกลับว่า Error ใดๆ ทั้งสิ้น

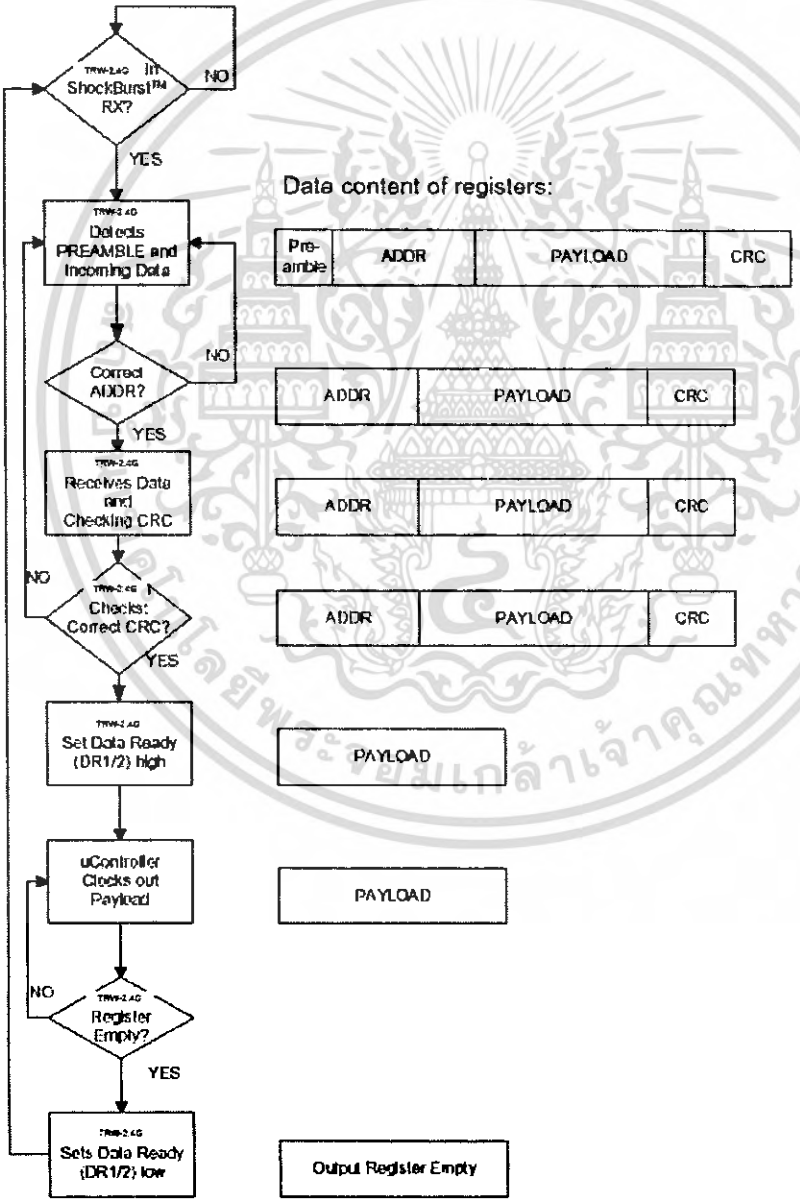
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TRW-2.4G จ set ขา DR1 เป็น 1 ดังนั้น microcontroller จึงคอยวน check ขา DR1 ว่าเป็น 1 เมื่อใดแสดงว่า Data Ready แล้ว
- Microcontroller ต้องสร้าง check มาที่ CLK1 เพื่ออ่านข้อมูลจากทาง DATA ออกไป ถ้าครบแล้ว DR1 จะกลายเป็น 0 อีกครั้ง

ถ้าหากใช้ Duo Ceiver (คือ สามารถกำหนดให้ TRW-2.4G ตัวนี้มีช่องรับ 2 ทาง) การแบ่งการทำงานออกเป็น 2 แชนแนล แต่ละ แชนแนล มี ADDR ของมันเอง โดยที่ใช้ขาสัญญาณไม่ตรงกัน

Rx1 ใช้ pin (DR1, CLK1, DATA)

Rx2 ใช้ pin (DR2, CLK2, DOUT2)



รูปที่ 2.39 แสดงไดอะแกรมการรับข้อมูลในโหมด ShockBurst ของตัวโมดูล TRW-2.4G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Config TRW-2.4G

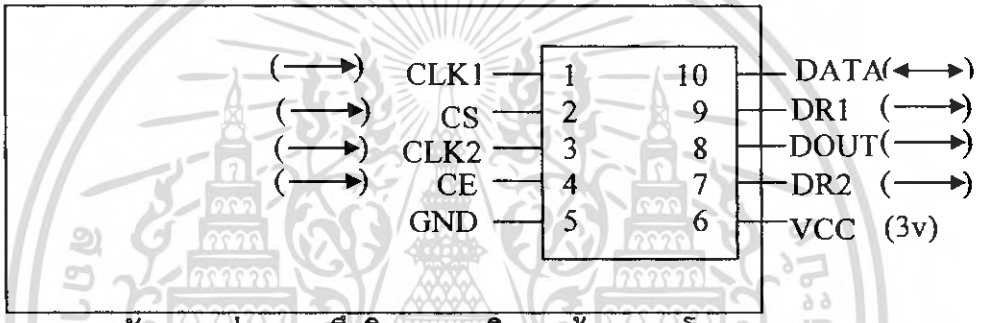
- สั่งให้ขา CS = 1 (เริ่มการ config) แต่ขา CE ต้องเป็น 0 ด้วยเพราะจะไม่อนุญาตให้ config และรับส่งข้อมูลพร้อมกันได้
- สั่งขา DATA ที่ใช้ config 144bit เข้าไปทาง DATA pin เริ่มจาก MSB และใช้ CLK1 ในการ Synchronous
- เมื่อเสร็จสิ้นให้ set CS = 0 เหมือนเดิม

ตารางที่ 2.13 แสดงคำสั่งการตั้งค่าต่างๆของโมดูล TRW-2.4G

ตำแหน่ง bit	จำนวน bit	ชื่อ	คำอธิบาย
143:120	24	Test	สงวนใช้Test
119:112	8	DATA2_1	ความยาวของ payload คำนวนจาก Payload = 256-ADDR_W-CRC
111:104	8	DATA1_W	
103:64	40	ADDR2	ค่าของADDR ของตัวนรับ ตั้งค่าได้ทั้ง แชนแนล 1 และ 2 มีขนาดสูงสุดได้ 5 ไบต์
63:24	40	ADDR1	
23:18	6	ADDR_W	กำหนดความยาวของ ADDR ได้
17	1	CSC_L	ความยาวของ CRC '0' = 8 bit '1' = 16 bit
16	1	CRR_EN	กำหนดใช้หรือไม่ใช้ CRC
15	1	RX2_EN	กำหนดใช้กี่ แชนแนล(รับ)
14	1	CM	เปิดปิด โหมดช็อคเบิสท์
13	1	RFDR_SB	กำหนด อัตราการส่งข้อมูล '0' = 250 กิโลบิตต่อวินาที '1' = 1 เมกะบิตต่อวินาที
12:10	3	XO_F	กำหนดเกี่ยวกับ CRYTAL แต่กำหนดตายตัวเป็น 011
9:8	2	RF_PWR	กำหนด Power ในการส่ง '00' (-20dBm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			'01' (-1dBm) '10' (-5dBm) '11' (0dBm)
7:1	7	RF_CH	กำหนดความถี่ที่ใช้ในการส่งคำนวณดังนี้ $RF_{ส่ง}$ = 2400+RF_CH เมกะเฮิร์ทซ์
0	1	RE_EN	กำหนดให้ TRW เป็น Tx-Rx '0'(Tx) '1'(RX)

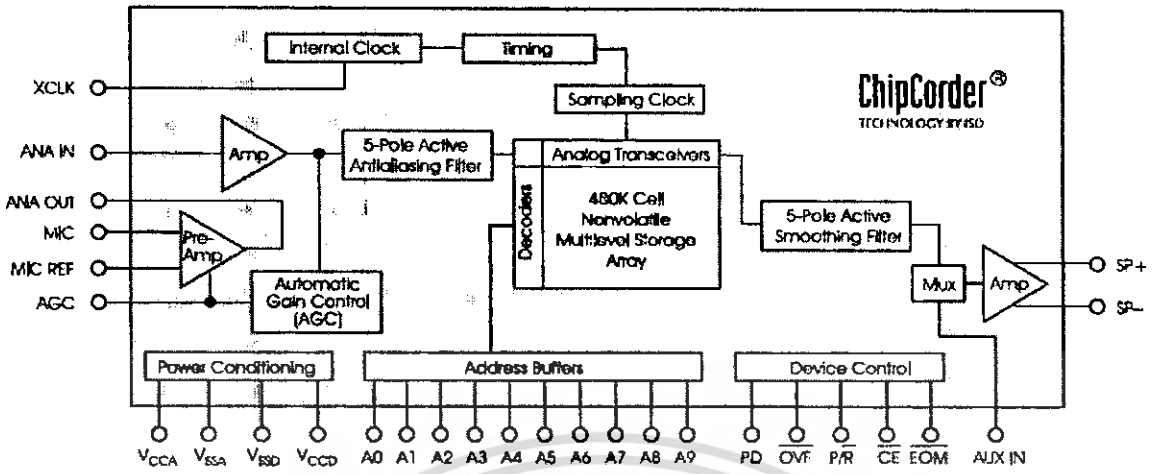


รูปที่ 2.40 แสดงขาสัญญาณต่างๆรวมถึงทิศทางการวิ่งของข้อมูลของโมดูล TRW-2.4G

2.7 ทฤษฎีและการใช้งานไอซีเสียง ISD2590

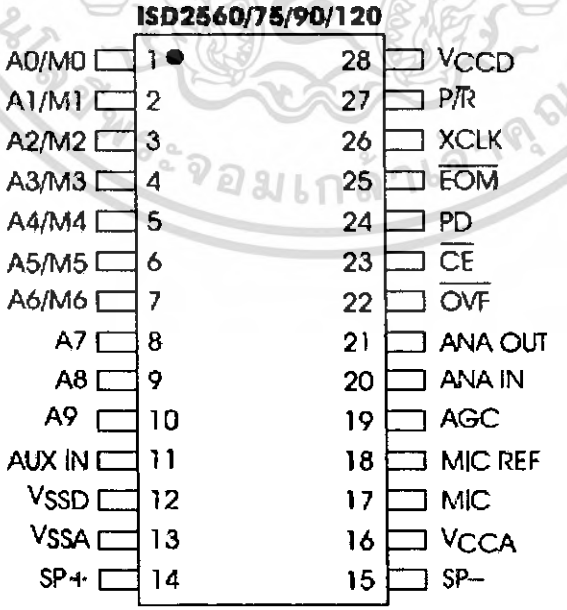
ในส่วนของการส่งสัญญาณเสียง ไอซีเสียง ISD2590 นี้มีความสามารถในการบันทึกและเล่นเสียงได้ทั้งหมด 90 วินาทีลงในหน่วยความจำภายใน สามารถเลือก แอดเดรส ในการบันทึกและเล่นเสียงได้ทั้งหมด 1024 แอดเดรส โดยใช้การควบคุมทั้งหมดจากขาสัญญาณภายนอก ทำให้ใช้งานได้ง่ายและสะดวก เกิดความผิดพลาดน้อย สามารถควบคุมการเล่น หรือหยุดเล่นด้วยขาสัญญาณ และใช้ไฟเลี้ยงไอซี 5 โวลต์ ทำให้ทำงานร่วมกับวงจรที่ออกแบบได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 แสดงถึงโครงสร้างภายในของไอซีเสียงเบอร์ ISD2590

จากรูปแสดงถึงโครงสร้างภายในของไอซีเสียง สังเกตเห็นว่าจะมีหน่วยความจำ ROM ภายในทำการบันทึกสัญญาณเสียงที่ได้รับจากไมโครโฟนจากขาสัญญาณ MIC เข้ามาบันทึกไว้ตามแอดเดรส ที่ได้รับการตั้งค่าจากขาสัญญาณ A0-A9 โดยใช้สัญญาณนาฬิกาจากขา XCLK ทำการสร้าง Sampling แล้วเปลี่ยนสัญญาณเสียงจาก อนาลอก เป็น ดิจิตอล เก็บลงหน่วยความจำ และสามารถส่งสัญญาณเสียงออกทาง SP+/SP- โดยมีวงจรขยายสัญญาณภายในทำให้สามารถต่อหูฟังได้ทันที และมีส่วนควบคุมผ่านทางขาสัญญาณ PD, OVF, P/R, CE และ EOM และ ISD2590 นี้มีอัตราการแซมปลิ้ง 5.3 กิโลเฮิร์ทซ์ ต้องการสัญญาณนาฬิกาภายนอก 682.7 กิโลเฮิร์ทซ์



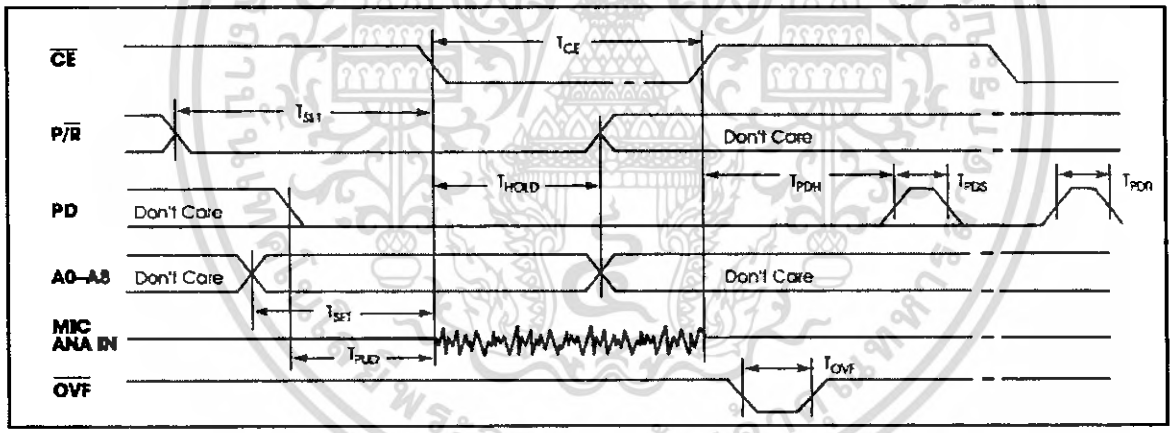
รูปที่ 2.42 แสดงถึงโครงสร้างภายขาสัญญาณของไอซีเสียงเบอร์ ISD2590

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISD2590 เป็นไอซีที่มีขาสัญญาณ 28 ขา บนแพกเกจแบบ DIP จึงค่อนข้างมีขนาดใหญ่แต่สามารถใช้งานได้ง่าย ขาสัญญาณแบ่งออกเป็นกลุ่มๆได้ดังนี้ กลุ่มการควบคุม (PD, OVF, P/R, CE และ EOM) กลุ่มไฟเลี้ยง (VSSD, VSSA, VCCD และ VCCA) กลุ่มสัญญาณอินพุต (AUX IN, AGV, MIC, MIC REF, ANA IN และ XCLK) กลุ่มสัญญาณเอาท์พุต (SP+, SP-, ANA OUT)

2.7.1 การใช้งานไอซีเสียงทำการบันทึกเสียง

a. ตั้งค่าให้ CE เป็น 1 เพื่อปิดการทำงานของไอซี และ PD (Power down) เป็น 0
b. ตั้งค่าให้ขา P/R เป็น 0 เป็นการเลือกทำการบันทึก
c. ตั้งค่า แอดเดรส ที่ต้องการผ่านขา A0-A9
d. สั่งให้ขา CE เป็น 0 ตลอดเมื่อต้องการบันทึกเสียง เมื่อบันทึกเสียงเสร็จสิ้นให้ขา CE กลับมาเป็น 1 โดยทันที เพื่อปิดการบันทึก
e. เมื่อบันทึกเสียงเสร็จจะมีสัญญาณพัลส์ OVF ออกมาจากไอซีเป็น 0



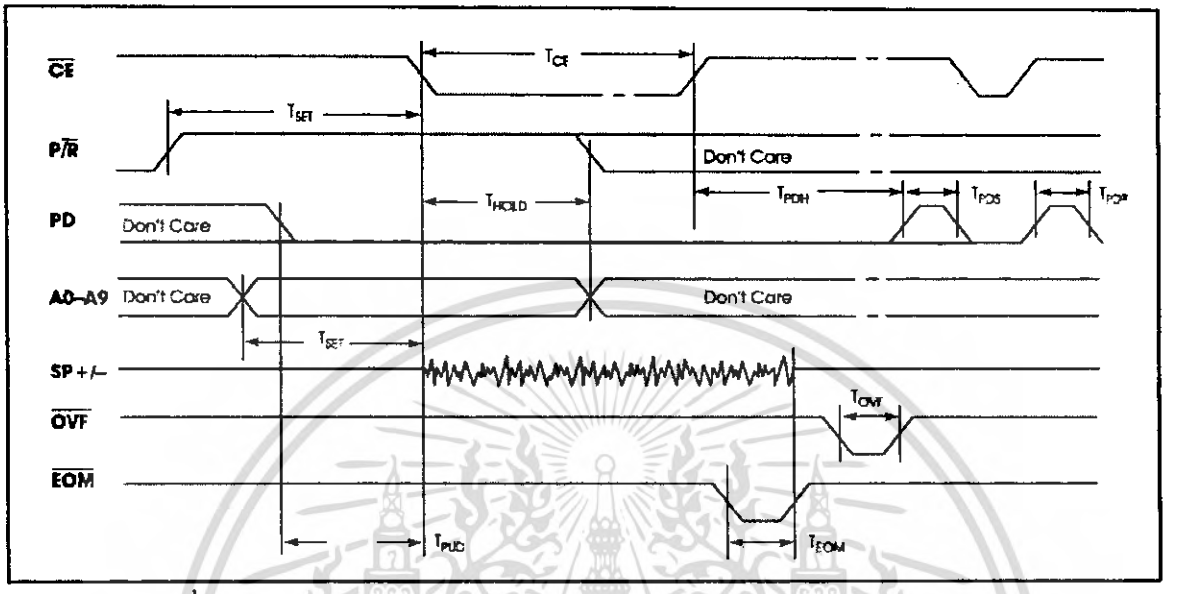
รูปที่ 2.43 แสดงถึงแผนผังเวลา (Timing Diagram) การบันทึกเสียงของไอซีเสียงเบอร์ ISD2590

2.7.2 การใช้งานไอซีเสียงทำการเล่นเสียง

a. ตั้งค่าให้ CE เป็น 1 เพื่อปิดการทำงานของไอซี และ PD (Power down) เป็น 0
b. ตั้งค่าให้ขา P/R เป็น 1 เป็นการเลือกทำการเล่นเสียง
c. ตั้งค่า แอดเดรส ที่ต้องการผ่านขา A0-A9
d. สั่งให้ขา CE เป็น 0 เมื่อต้องการเริ่มเล่นเสียงเท่านั้น ไอซีเสียงจะทำการเล่นเสียงออกทาง SP+/SP- จนจบสัญญาณ โดยขณะเล่นเสียงจะไม่รับคำสั่งใดๆทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e. เมื่อเล่นเสียงเสร็จจะมีสัญญาณพัลส์ OVF ออกมาจากไอซีเป็น 0 และสัญญาณ EOM (End of Message) จะเป็น 0 ขณะจบสัญญาณ



รูปที่ 2.44 แสดงจึงแผนผังเวลาการเล่นเสียงของไอซีเสียงเบอร์ ISD2590

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการพัฒนา

3.1 ภาพรวมของการออกแบบ

3.1.1 กำหนดคุณลักษณะการใช้งาน

3.1.1.1 อุปกรณ์แบ่งออกเป็น 2 ส่วน คือ ส่วนพกพา (iGUIDE Portable) และส่วนสถานีสัญญาณ (iGUIDE Station)

3.1.1.2 ส่วนพกพาจะประกอบด้วย การ์ดข้อมูลเสียงเอ็มเอ็มซี ตัวเครื่อง และหูฟัง

3.1.1.3 ส่วนพกพาสามารถเชื่อมต่อกับคอมพิวเตอร์ได้

3.1.1.4 ส่วนพกพาเมื่อได้รับสัญญาณบอกไอดี (ID) จากสถานี จะทำการเล่นไฟล์เสียงที่ตรงตามสัญญาณบอกเอกลักษณ์นั้นๆ ได้

3.1.1.5 ส่วนพกพาสามารถปรับความดังของเสียงได้

3.1.1.6 เมื่อไม่มีสัญญาณจากสถานีใดเลย ให้เครื่องพกพาสามารถเล่นเพลงหรือเสียงอื่นๆ ได้

3.1.1.7 ส่วนสถานีสามารถตั้งค่าหมายเลขเอกลักษณะได้

3.1.1.8 หน่วยความจำที่ต้องการ ประมาณ 128 เมกกะไบต์

3.1.2 พิจารณาเลือกใช้เทคโนโลยี

ตารางที่ 3.1 แสดงการพิจารณาเลือกใช้เทคโนโลยีเพื่อเชื่อมต่อกับคอมพิวเตอร์

ตัวเลือก	ราคา(บาท)	ความเร็ว	ระยะทาง
USB	**60**	**1.5Mbps - 12Mbps**	**Cable**
Serial	35	9.6 kbps - 19.2kbps	Cable
Bluetooth	1000-3000	1 Mbps -3 Mbps	1-100 m
IR	500-800	9.6 kbps - 16 Mbps	1-100 cm

โดยเราเลือกใช้ ยูเอสบี เพื่อใช้ติดต่อกับเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงการพิจารณาเลือกใช้เทคโนโลยีเพื่อติดต่อแบบไร้สายกับสถานีสัญญาณ

รูปแบบ	ราคา(บาท)	ความเร็ว	ระยะทาง
Bluetooth	1000	1 Mbps -3 Mbps	1-100 m
IR	500-800	9.6 kbps - 16 Mbps	1-100 cm
RFID	>5000	ขึ้นอยู่กับความถี่	1-300 cm
RF	**400-1000**	**ต่ำ ขึ้นอยู่กับความถี่**	**30-100 m**

โดยเราเลือกใช้สัญญาณวิทยุ (RF) เพื่อใช้ในส่วนการติดต่อระหว่างส่วนพกพากับส่วนสถานีสัญญาณ

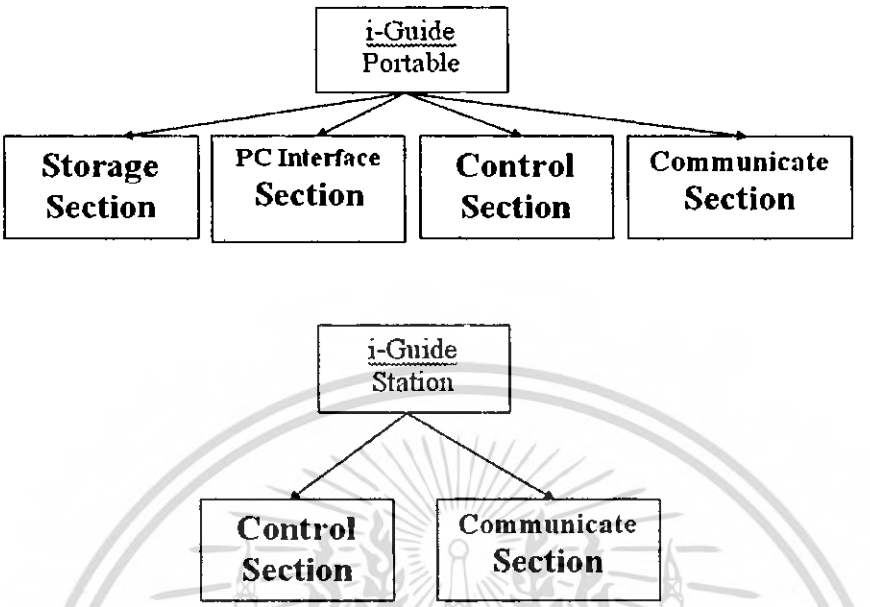
ตารางที่ 3.3 แสดงการพิจารณาเลือกใช้เทคโนโลยีอุปกรณ์เก็บข้อมูล

ชนิด	ความจุ	ความเร็ว	ขนาด	ราคา(บาท)
SD	256 MB	66x(10 MB/s),150x(15MB/s)	20 x 21.5 x 1.4 mm	1000
	512 MB			1700
	1 GB			2900
MMC		9 MB/s	**20 x 21.5 x 1.4 mm**	
	128 MB			360
	256 MB			700
	512 MB			**1000**
Harddisk	> 20 GB	133 MB/s	Very Large	> 2000
Compact Flash	256 MB	-	43 x 36 x 3.3 mm	1200
	512 MB			1800
Flash Drive	128 MB	แล้วแต่ความเร็วของยูเอสบีพอร์ต	แล้วแต่ยี่ห้อ	360
	256 MB			420
	512 MB			580
	1GB			1050
	2 GB			2090

โดยเราเลือกใช้เอ็มเอ็มซีใช้งานในส่วนเก็บข้อมูลเนื่องด้วยเรามีความเข้าใจในโปรโตคอลของเอ็มเอ็มซีมากกว่าโปรโตคอลของสื่อเก็บข้อมูลแบบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ออกแบบบล็อกไดอะแกรม (Block Diagram)



รูปที่3.1 แสดงภาพรวมของบล็อกไดอะแกรมทั้งหมด

จากภาพจะเห็นว่ามีการแบ่งภาพรวมออกเป็นสองส่วนคือ

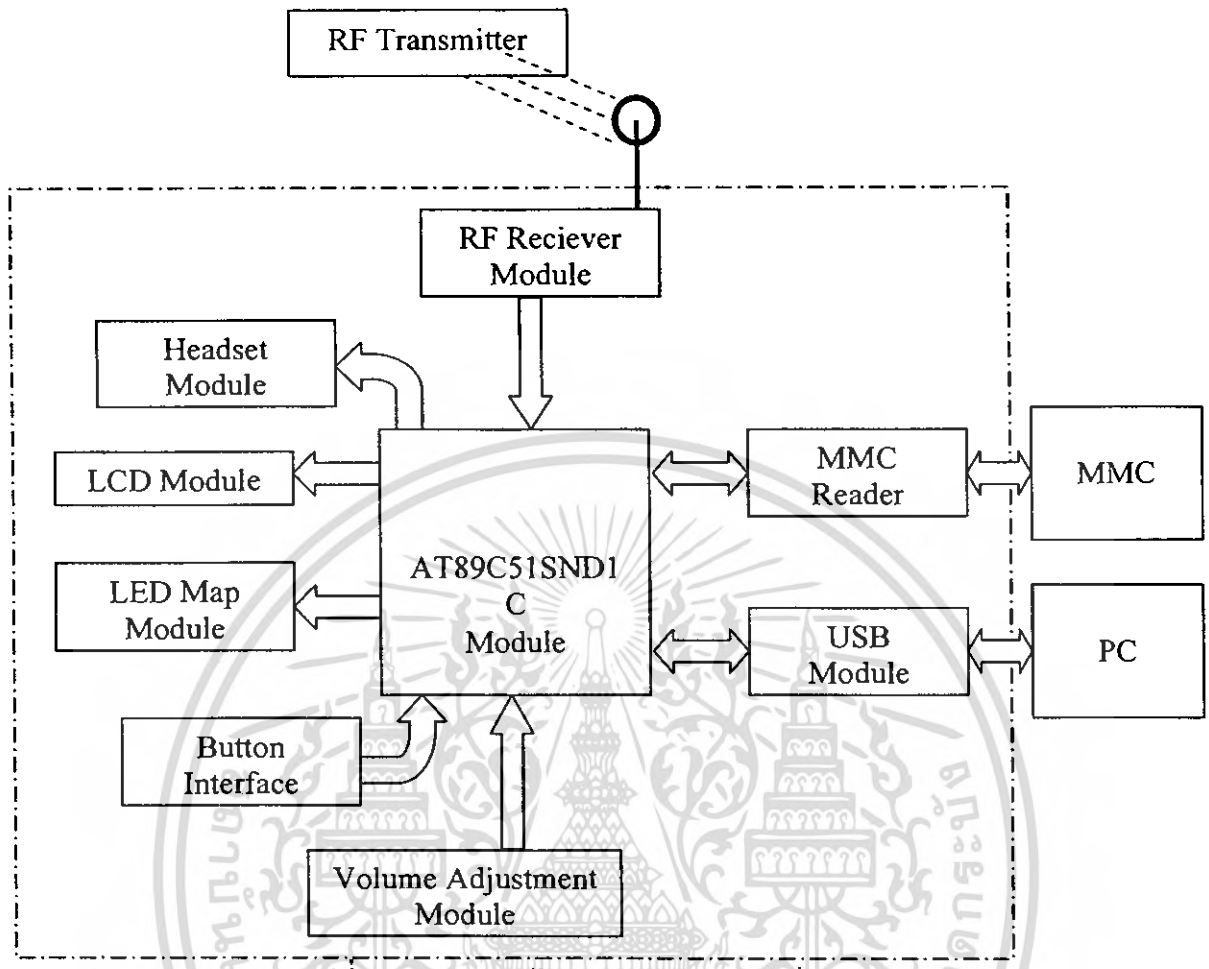
3.1.3.1 ส่วนอุปกรณ์พกพา (iGUIDE Portable) โดยแบ่งเป็นส่วนย่อยๆ คือ

- ส่วนเก็บข้อมูล (Storage section)
- ส่วนติดต่อกับคอมพิวเตอร์ (PC Interface section)
- ส่วนควบคุม (Control section)
- ส่วนติดต่อกับสถานีสัญญาณ (Communicate section) เป็นภาคการรับสัญญาณเท่านั้น

3.1.3.2 ส่วนสถานีสัญญาณ (iGUIDE Station) แบ่งเป็นส่วนย่อยๆ คือ

- ส่วนติดต่อกับอุปกรณ์พกพา (Communicate section) เป็นภาคการส่งสัญญาณเท่านั้น
- ส่วนควบคุม (Control section)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2 แสดงบล็อกไดอะแกรมของอุปกรณ์

สังเกตจากบล็อกไดอะแกรมว่าอุปกรณ์จะมี AT89C51SND1C Module เป็นศูนย์การในการทำงานทั้งหมดโดย AT89C51SND1C Module จะติดต่อกับส่วนการทำงานต่างๆดังนี้

ส่วนการติดต่อแบบทิศทางเดียว (Simplex)

- **Headset Module** ส่วนควบคุมลำโพงหูฟัง
- **LCD Module** ส่วนควบคุมจอแอลซีดี
- **LED Map Module** ส่วนควบคุม หลอดแอลอีดี
- **Button Interface** ปุ่มกดควมคุมการทำงานของอุปกรณ์
- **Volume Adjustment Module** ส่วนการปรับความดังเบาของเสียง
- **RF Receiver Module** ส่วนรับสัญญาณวิทยุจาก สถานีสัญญาณ เพื่อจะรู้ได้ว่าอยู่ในพื้นที่ของ แอดเดรส ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการติดต่อแบบสองทิศทาง (Duplex)

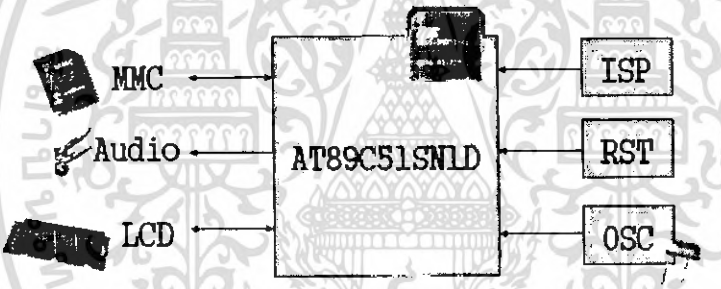
- **MMC Reader** ส่วนการจัดการการอ่านและเขียนข้อมูลไปยังเอ็มเอ็มซี
- **USB Module** เป็นส่วนเชื่อมต่อระหว่างอุปกรณ์และคอมพิวเตอร์

3.2 การออกแบบส่วนฮาร์ดแวร์

3.2.1 การออกแบบผังวงจรส่วนพกพา (iGUIDE Portable)

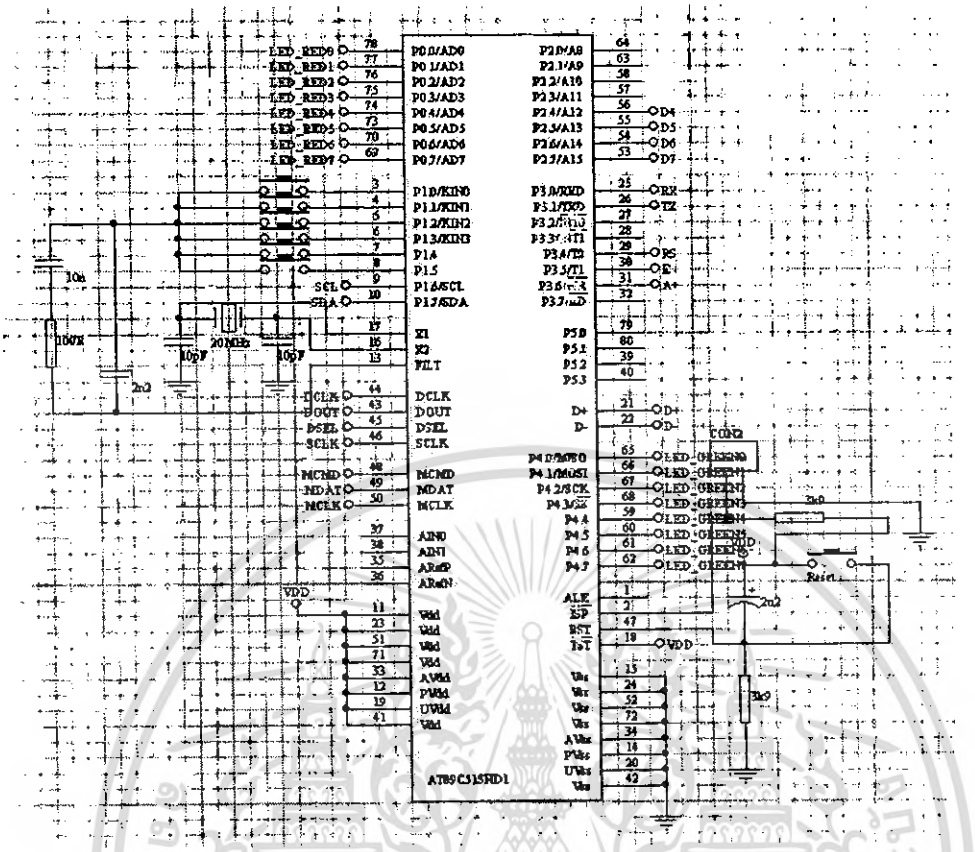
เราใช้ไมโครคอนโรลเลอร์เบอร์ AT89C51SND1C เป็นตัวควบคุมการทำงานของอุปกรณ์พกพา ทั้งหมดและใช้ไมโครคอนโทรลเลอร์เบอร์ AT89Cx041 เป็นตัวควบคุมการทำงานของส่วนสถานีสัญญาณ โดยทำการวาดผังวงจรทั้งหมดด้วยโปรแกรม Protel99 ก่อน (สามารถดูผังวงจรของส่วนอุปกรณ์พกพาได้จากภาคผนวก) จากนั้นจึงทำการออกแบบลายวงจร ในขั้นตอนต่อไป

3.2.1.1 ส่วนไมโครคอนโทรลเลอร์



รูปที่ 3.3 แสดงการออกบล็อกไดอะแกรมส่วนของไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C

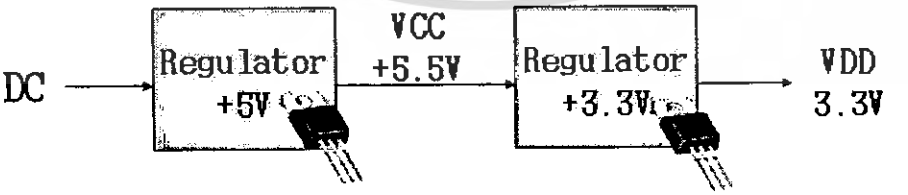
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงการออกแบบผังวงจรส่วนของไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C

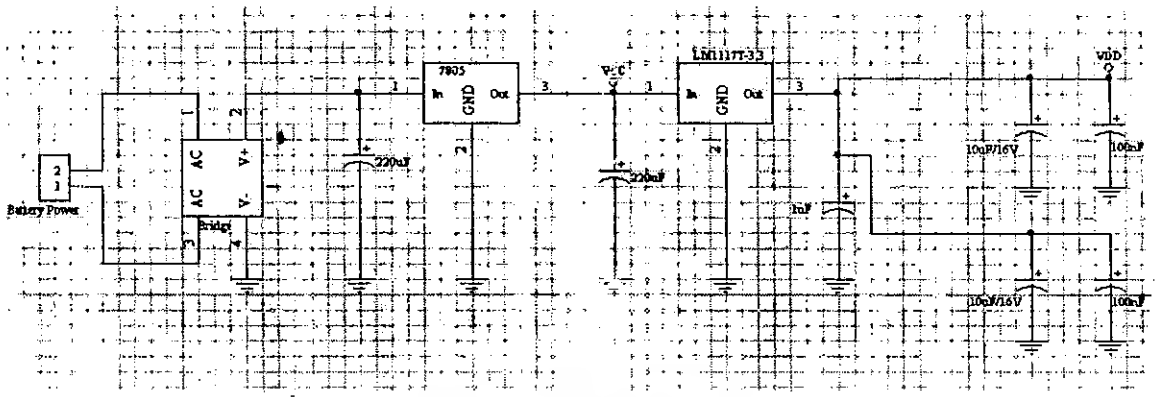
ส่วนนี้เป็นส่วนของไมโครคอนโทรลเลอร์หลัก ประกอบด้วยวงจรในส่วน RST เพื่อใช้ในการรีเซต ให้ไมโครคอนโทรลเลอร์กลับไปเริ่มทำงานใหม่ และส่วนของไอเอสพี เพื่อให้สำหรับตั้งค่าการทำ In-System Programming และส่วนของการสร้างสัญญาณนาฬิกาเพื่อให้ไมโครคอนโทรลเลอร์ทำงานได้ และต้องจ่ายไฟเลี้ยงวงจรตามจุดต่างๆ ให้ครบเพื่อให้ทำงานได้

3.2..1.2 ส่วนภาคจ่ายไฟ



รูปที่ 3.5 แสดงการออกแบบบล็อกไดอะแกรมของส่วนภาคจ่ายไฟ

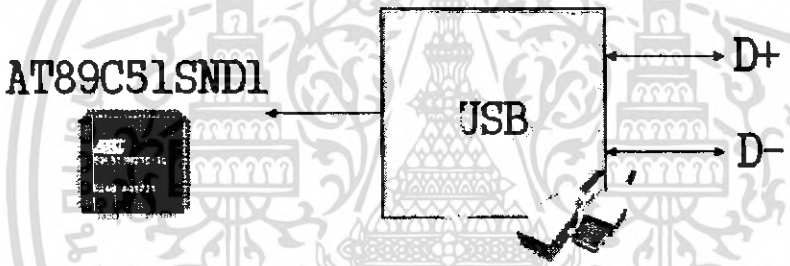
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



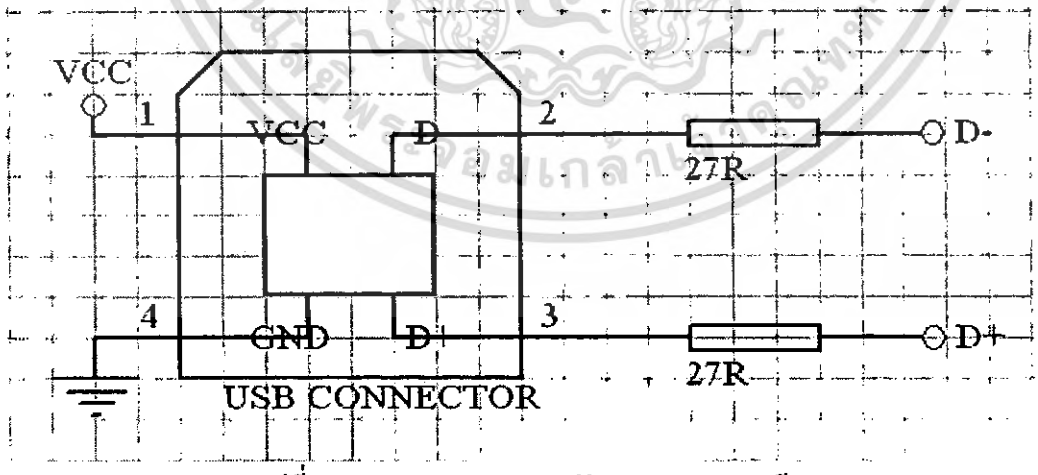
รูปที่ 3.6 แสดงการออกแบบผังวงจรส่วนของภาคจ่ายไฟ

ส่วนนี้ออกแบบให้สามารถจ่ายไฟได้ 2 แบบ คือ 5V และ 3.3 V (VDD) เพื่อไมโครคอนโทรลเลอร์และเอ็มเอ็มซี ส่วนของ 5V (VCC) ในส่วนอื่นๆ

3.2.1.3 ส่วนของยูเอสบี



รูปที่ 3.7 แสดงการออกแบบบล็อกไดอะแกรมของยูเอสบี

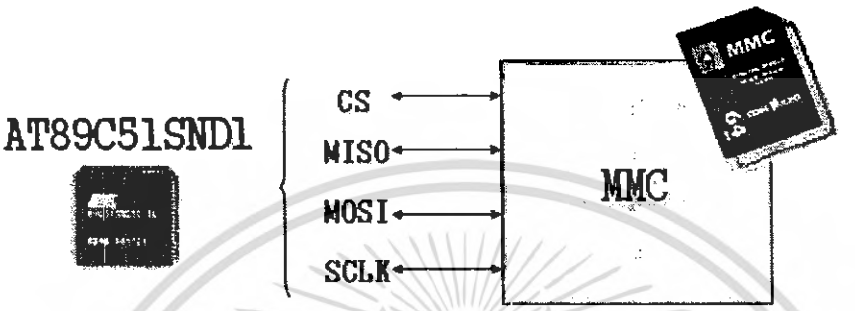


รูปที่ 3.8 แสดงการออกแบบผังวงจรของยูเอสบี

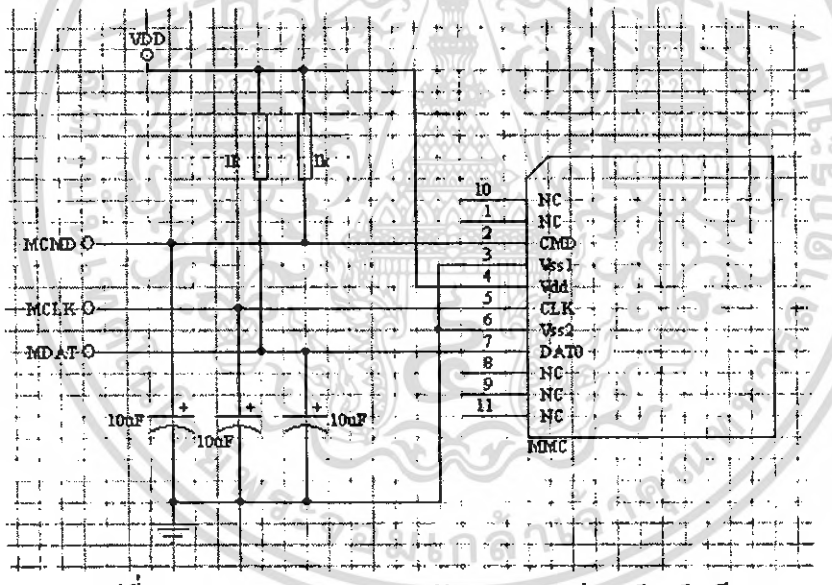
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้มีเพียงหัวยูเอสบีและวงจรเพียงเล็กน้อย ตัวจัดการโปรโตคอลทั้งหลายนั้นถูกรวมไว้ในไมโครคอนโทรลเลอร์ โดยยูเอสบี แบ่งออกเป็น 4 ขา 2 ขาเป็นไฟเลี้ยงและอีก 2 ขาเป็นขาสัญญาณ D+ และ D-

3.2.1.4 ส่วนของเอ็มเอ็มซี



รูปที่ 3.9 แสดงการออกแบบบล็อกไดอะแกรมของส่วนเอ็มเอ็มซี

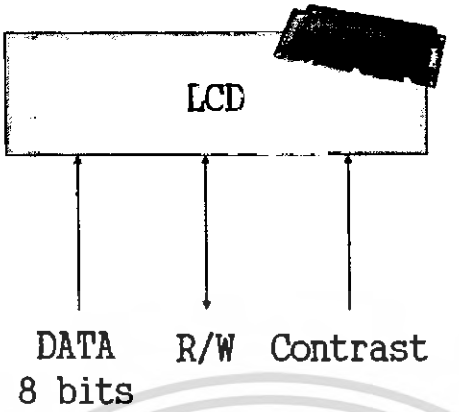


รูปที่ 3.10 แสดงการออกแบบผังวงจรของส่วนเอ็มเอ็มซี

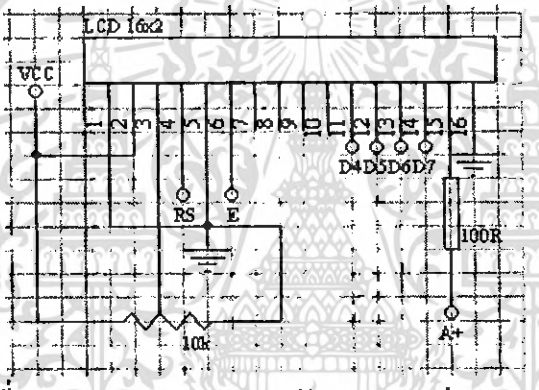
ส่วนนี้เชื่อมต่อกับไมโครคอนโทรลเลอร์ ให้สามารถติดต่อแล้วควบคุมการทำงานของเอ็มเอ็มซีได้โดยผ่านโปรโตคอลเอ็มเอ็มซีจะประกอบด้วย 3 เส้นสัญญาณ คือ MDAT เป็นเส้นสัญญาณข้อมูล MCMD เป็นเส้นสัญญาณเพื่อใช้ในการสั่งงานและรับการตอบสนองกลับจาก เอ็มเอ็มซี และ MCLK เป็นขาสัญญาณเพื่อส่งสัญญาณนาฬิกามาควบคุมการสื่อสารของโปรโตคอลนี้ โดยในแต่ละเส้นสัญญาณจะมีตัวต้านทานมาต่อ pull-up เพื่อให้สัญญาณชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.5 ส่วนแสดงผลของแอลซีดี



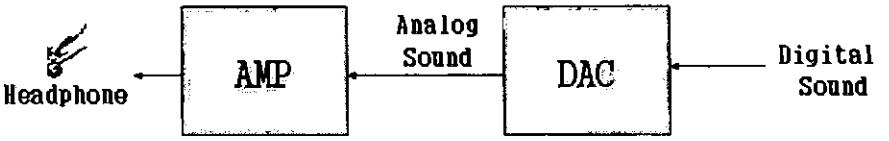
รูปที่ 3.11 แสดงการออกแบบบล็อกไดอะแกรมของส่วนแอลซีดี



รูปที่ 3.12 แสดงการออกแบบผังวงจรของส่วนแอลซีดี

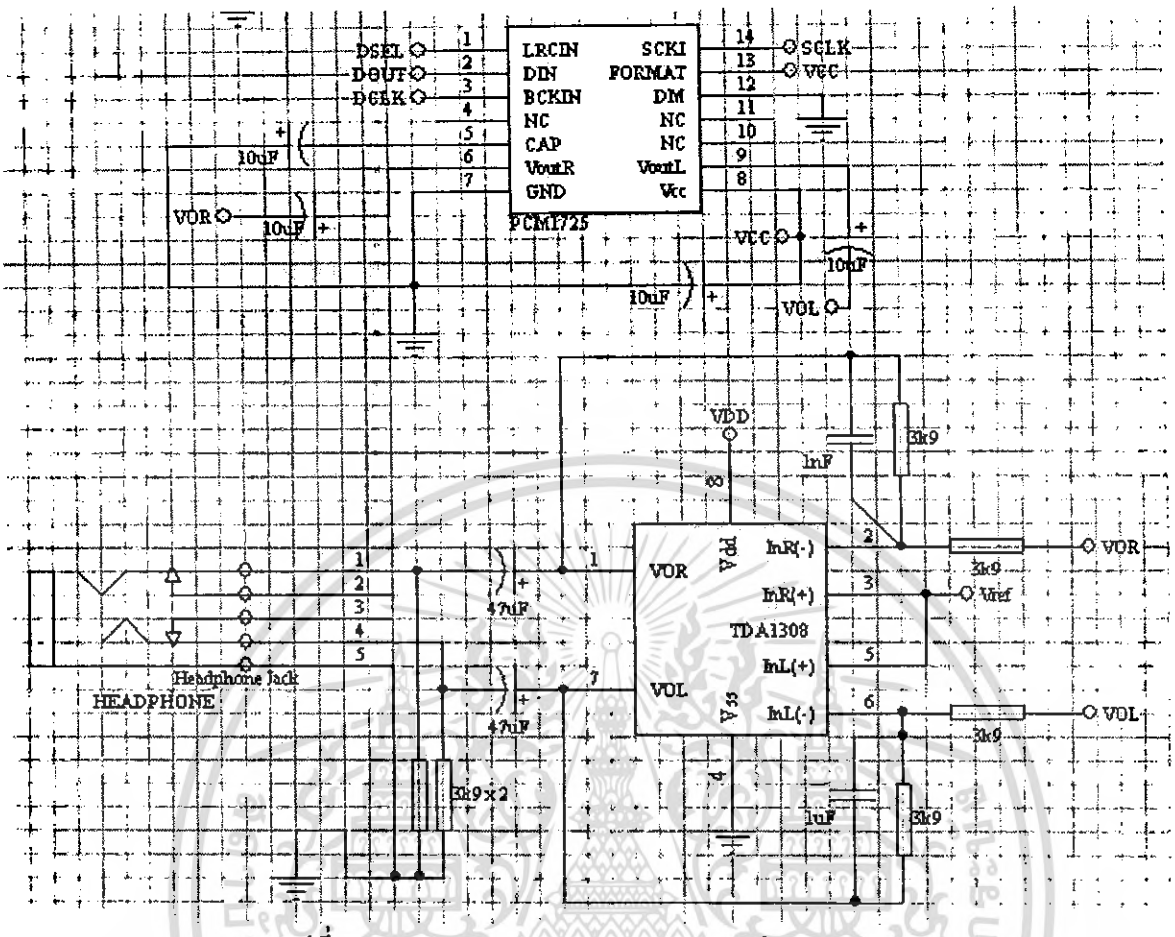
เป็นส่วนที่ต่อวงจรเพื่อให้แอลซีดีทำงานได้ โดยมีขาส่งสัญญาณข้อมูลไปแอลซีดี 8 เส้น หมายถึงข้อมูล 8 บิต ที่ส่งไป และขาสัญญาณควบคุมอื่นๆ เช่น อ่านหรือเขียน เปิดการติดค่อ ปรับค่าความเข้มของแอลซีดีและไฟเลี้ยง

3.2.1.6 ส่วนแสดงผลด้านเสียงออกทางหูฟัง



รูปที่ 3.13 แสดงการออกแบบบล็อกไดอะแกรมของส่วนหูฟัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

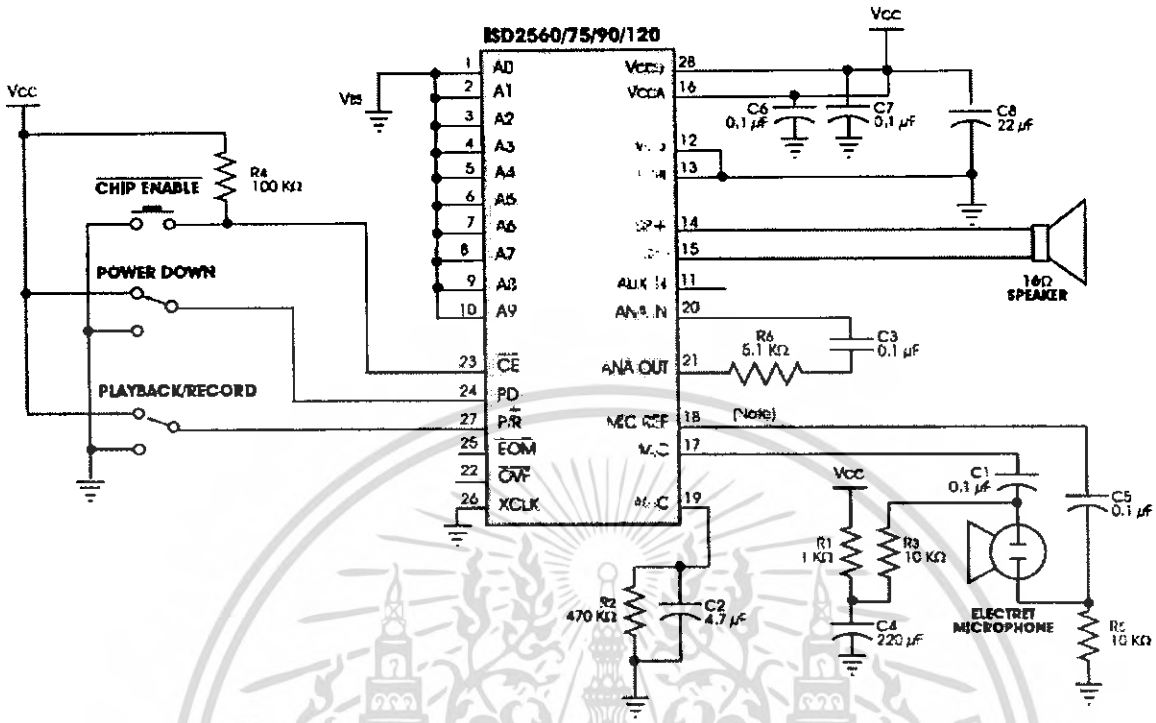


รูปที่ 3.14 แสดงการออกแบบผังวงจรของส่วนหูฟัง

ส่วนนี้ประกอบไปด้วยไอซีทำหน้าที่ต่อเนื่องกัน ตัวแรกทำงานแปลงสัญญาณดิจิตอลเป็นอนาล็อกแล้ว ตัวที่สองทำงานขยายสัญญาณเสียง โดยสุดท้ายส่งสัญญาณเสียงออกไปทางแจ็คหูฟัง เพื่อเชื่อมต่อไปสู่หูฟัง โดยการต่อในส่วนอื่นๆเป็นไปตามข้อมูลที่ได้ในเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.7 ส่วนของไอซีเสียง ISD2590



รูปที่ 3.15 แสดงการออกแบบส่วนของ ISD2590

ส่วนนี้เป็นการออกแบบส่วนไอซีเสียง ISD2590 เพื่อให้สามารถทำการบันทึกและเล่นเสียงได้ โดยประกอบด้วยส่วนอินพุต ไมโครโฟน ส่วนเอาท์พุต ลำโพงหรือหูฟัง และมีสวิตท์ควบคุมให้สามารถเลือกให้ทำการบันทึกหรือเล่นเสียงได้ และสวิตท์ควบคุม CE ส่วนของ แอดเดรส จะทำการต่อพ่วงกับไมโครคอนโทรลเลอร์เพื่อควบคุมผ่านไมโครคอนโทรลเลอร์

โดยผังวงจรรวมที่ใช้ไอซีเสียงนี้จะแสดงอยู่ในภาคผนวก

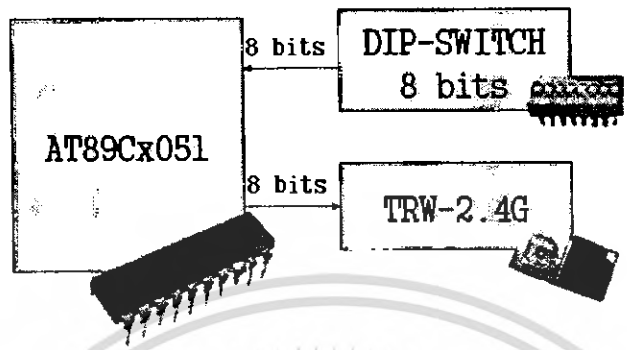
3.2.2 ออกแบบลายวงจรส่วนอุปกรณ์พกพา

เมื่อทำการออกแบบผังวงจรเสร็จเรียบร้อยแล้วจากนั้นจึงทำการวาดลายวงจรในโปรแกรม Protel99 อีกเช่นกันซึ่งลายวงจรนี้จะสามารถนำไปสั่งกัดเป็นแผ่นลายวงจร (PCB) ได้ (สามารถดูแผ่นลายวงจรได้จากภาคผนวก)

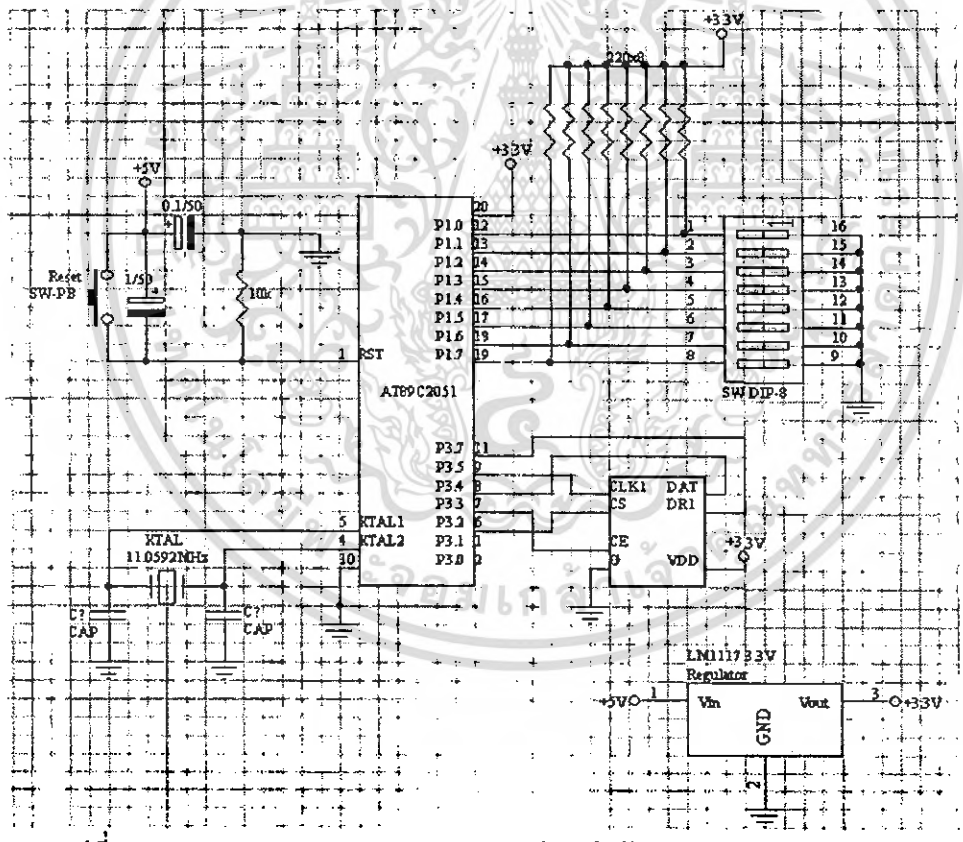
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ออกแบบผังวงจรส่วนสถานีสัญญาณ (ส่วนสื่อสารด้วยโมดูล TRW-2.4G)

3.2.3.1 ส่วนภาคการส่งข้อมูล



รูปที่ 3.16 แสดงส่วนการออกแบบของส่วนส่งข้อมูลของสถานีสัญญาณ



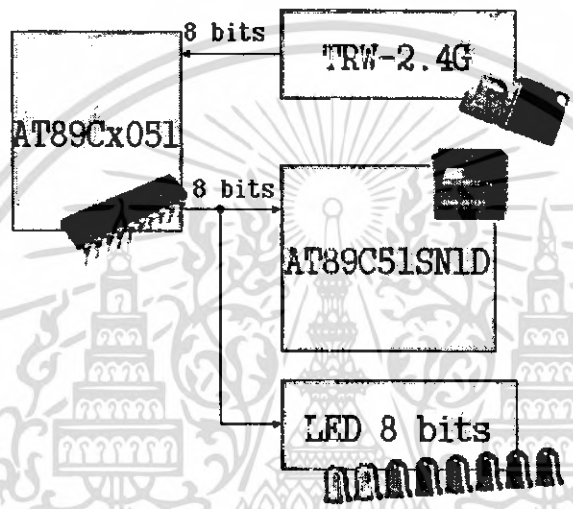
รูปที่ 3.17 แสดงการออกแบบผังวงจรของส่วนส่งข้อมูลของ สถานีสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

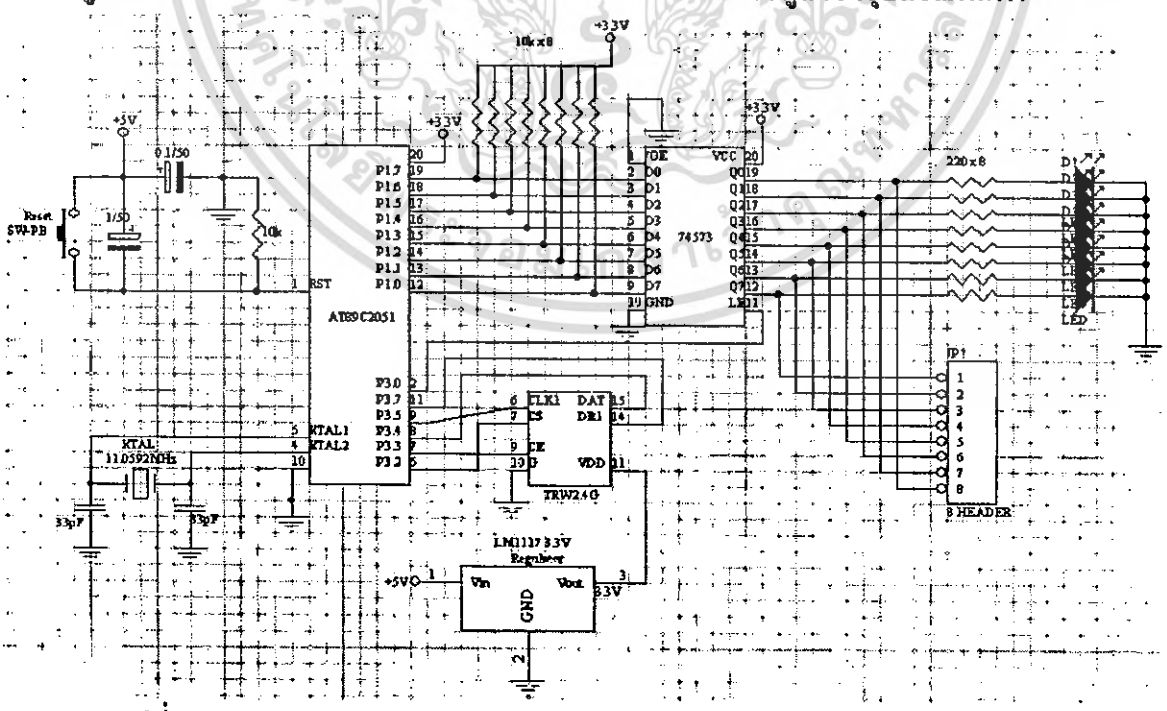
จากรูปในส่วนภาคการส่งข้อมูลจะมีการใช้ พอร์ต1 ในการติดต่อรับข้อมูลจาก DIP-SWITCH ขนาด 8 บิต เป็นส่วนติดต่อกับผู้ใช้ เพื่อสามารถตั้งค่าไอดี ของสถานีสัญญาณจาก DIP-SWITCH ได้

ส่วนของพอร์ต 3 ใช้ติดต่อกับโมดูล TRW-2.4G เพื่อใช้ในการตั้งค่าโมดูลให้ทำหน้าที่เป็นตัวส่งข้อมูลด้วยสัญญาณวิทยุ (ส่งเป็นแบบทางเดียว)

3.2.3.2 ส่วนภาคการรับข้อมูล



รูปที่ 3.18 แสดงการออกแบบบล็อกไดอะแกรมของส่วนรับข้อมูลของอุปกรณ์พกพา



รูปที่ 3.19 แสดงการออกแบบผังวงจรของส่วนตัวรับข้อมูลของอุปกรณ์พกพา

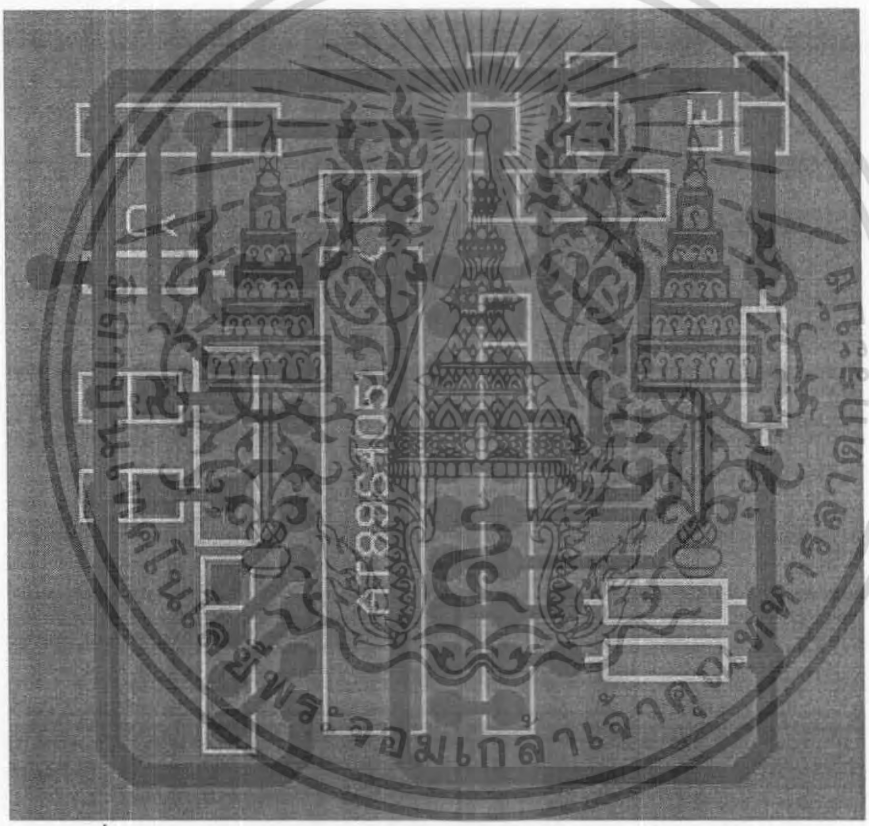
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปในส่วนภาคการรับข้อมูลจะมีการใช้พอร์ต 1 ในการติดต่อกับไอซี 74573 เพื่อใช้ทดสอบขับหลอดแอลอีดีขนาด 8 บิตเพื่อทดสอบว่าส่งสัญญาณได้ถูกต้องและเพิ่มขาเชื่อมต่อเพื่อไว้ใช้ต่อออกไปติดต่อกับส่วนบอร์ดใหญ่ของอุปกรณ์พกพาด้วย

ส่วนของพอร์ต 3 ใช้ติดต่อกับโมดูล TRW-2.4G เพื่อใช้ในการตั้งค่าโมดูลให้ทำหน้าที่เป็นตัวรับข้อมูล

3.2.4 ออกแบบลายวงจรส่วนสถานีสัญญาณ (iGUIDE Station)

ออกแบบลายวงจรของส่วนสถานีสัญญาณและส่วนรับข้อมูลของสถานีสัญญาณด้วยโปรแกรม Protel99

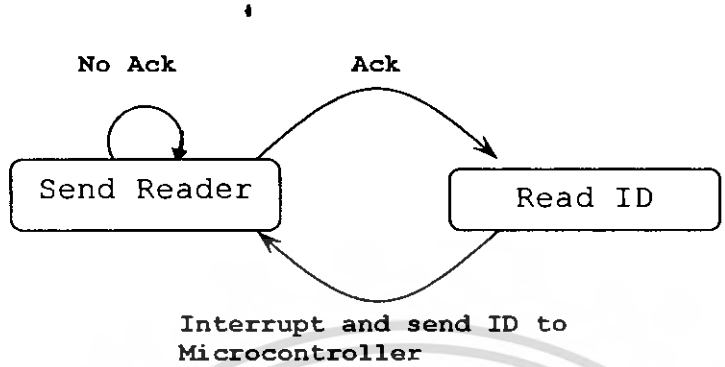


รูปที่ 3.20 แสดงการออกแบบลายวงจรของภาคการส่งและรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบส่วนโปรแกรมที่ใช้ควบคุมฮาร์ดแวร์ (Firmware)

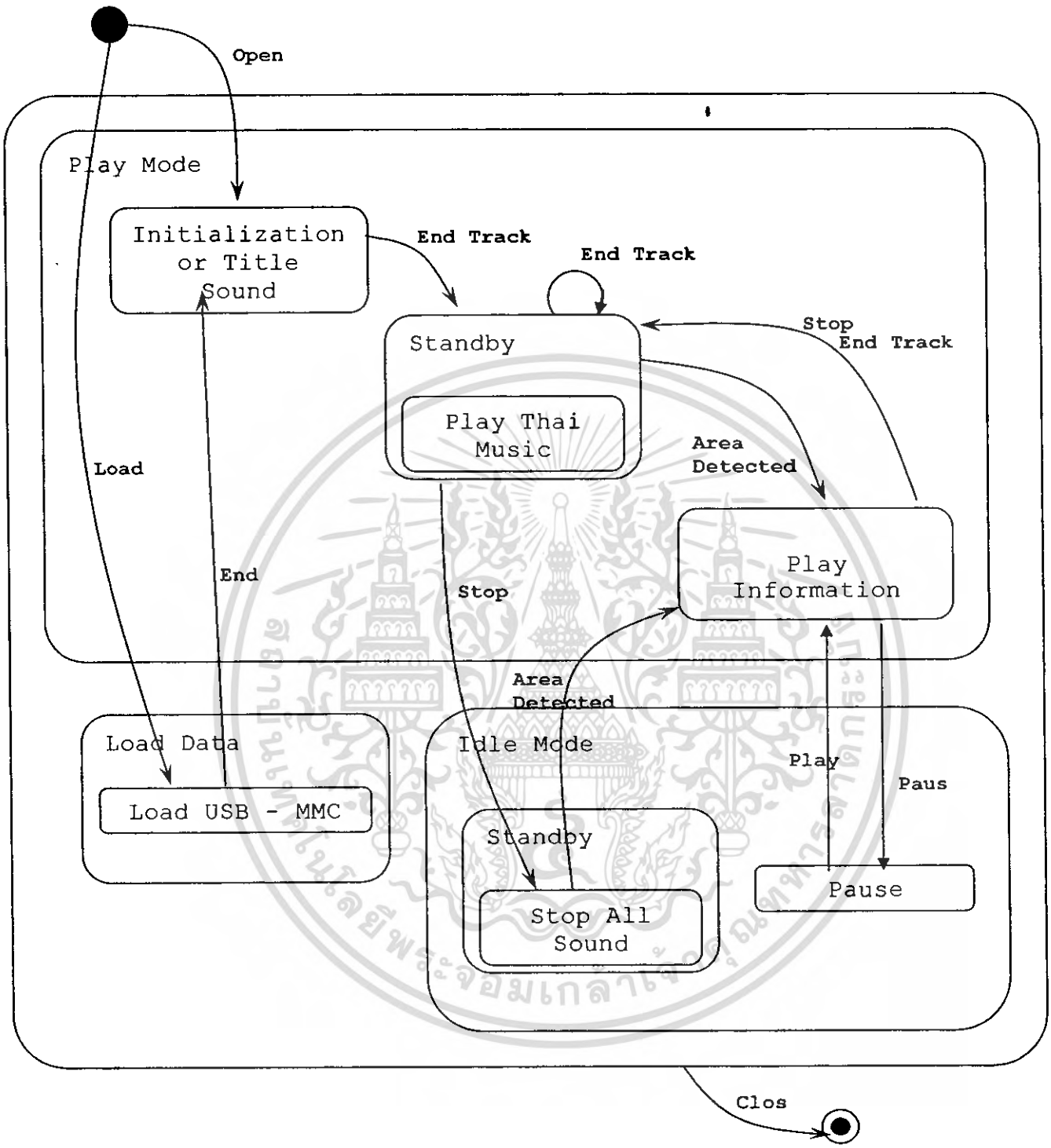
3.3.1 ออกแบบสถานะการทำงาน (State Diagram)



รูปที่ 3.21 แสดงสถานการณ์การทำงานของโปรแกรม

จากรูปข้างบนซึ่งแสดงสถานะการทำงานจะเห็นได้ว่าส่วนของ อุปกรณ์พกพา จะทำการอ่านค่า ไอดี จาก สถานีสัญญาณ อยู่เสมอเมื่อเข้าในเขตที่มีสัญญาณของพื้นที่นั้นๆแต่ถ้าไม่ได้อยู่ในเขตสัญญาณของ สถานีสัญญาณ ไม่มีอ่านค่าสัญญาณ อุปกรณ์พกพา จะประมวลผลในสิ่งที่ทำอยู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แสดงสถานการณ์ทำงานของส่วนอุปกรณ์พกพา

จากรูปแสดงสถานการณ์ทำงานด้านบนนี้จะเห็นได้ว่าเมื่อเราเปิดเครื่อง อุปกรณ์พกพาขึ้นมานั้นจะเกิดการทำงานได้ 2 สถานะคือ

1.ถ้าเราไม่ได้ต่อสายยูเอสบีกับเครื่องคอมพิวเตอร์อยู่นั้นจะอยู่ในสถานะ Play mode และจะเข้าสู่ส่วนการเล่น Title Sound ของการเปิดเครื่อง เมื่อเล่นTitle track จบก็จะเข้าสู่โหมด Standby เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอรับสัญญาณ ไอดี จาก สถานีสัญญาณ เพื่อนำ ไอดี นั้นมาใช้ในการเลือกไฟล์เอ็มพีสามที่ถูกต้องกับพื้นที่ที่ส่ง ไอดี มาไปเล่นให้ผู้ใช้ฟังซึ่งจะเข้าสู่โหมด Play Information ซึ่งในโหมดนี้นั้นผู้ใช้สามารถเพิ่มหรือลดเสียงหรือหยุดการเล่นชั่วคราว (Pause)ได้ซึ่งจะเข้าสู่ Idle mode หรืออาจจะกดปุ่มยกเลิกการเล่นไฟล์เอ็มพีสามนี้ (Stop) ซึ่งจะกลับเข้าไปสู่ Standby mode อีกที

2.ถ้าต่อสาย ยูเอสบี กับเครื่องคอมพิวเตอร์อยู่นั้นก็จะเข้าสู่สถานะ Load Data แทนซึ่งจะสามารถนำข้อมูลจากเสียงจากเครื่องคอมพิวเตอร์ลงใส่ใน เอ็มเอ็มซี ได้เมื่อโอนถ่ายข้อมูลเสร็จก็จะกลับไปสู่ส่วนการเล่น Title sound อีกที

3.3.2 ออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) ทั้งคีย์แพดและจอแอลซีดี

ตารางที่ 3.4 ตารางแสดงการออกแบบส่วนติดต่อกับผู้ใช้

Type	Present	Input	Next
1	WELCOME TO XXXX 00/00/00 00:00	Signal Detection	PLACE NAME X >00:00
2	PLACE NAME X >02:45	End of Track	WELCOME TO XXXX 00/00/00 00:00
3	PLACE NAME X >00:05	Stop	WELCOME TO XXXX 00/00/00 00:00
4	PLACE NAME X >00:05	Play/Pause	PLACE NAME X #00:05
5	PLACE NAME X #00:05	Play/Pause	PLACE NAME X >00:06
6	PLACE NAME X >00:05	Reset	PLACE NAME X >00:00
7	Any Page(Save State)	+ −	VOLUME #########
8	VOLUME #########	2 Second	Save State

จากตารางจะแสดงให้เห็นว่าการใช้ปุ่มต่างๆจะแสดงผลอย่างไรบ้าง

แบบที่ 1 เมื่ออยู่ในโหมดการเล่น Title sound แล้วมีการเข้าสู่พื้นที่ที่มีสัญญาณของ สถานีสัญญาณ จอแอลซีดีจะแสดงชื่อของสถานที่หรือส่วนที่เป็นพื้นที่ของ สถานีสัญญาณ นั้นๆและแสดงเวลาของ track นั้นๆด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบที่ 2 เมื่ออยู่ในโหมดเล่นไฟล์เสียงนั้นๆอยู่แล้วกดปุ่ม End of track เครื่องก็จะกลับเข้ามาสู่โหมดการเล่นเพลง Title sound อีกครั้ง

แบบที่ 3 เมื่ออยู่ในโหมดแล่นไฟล์เสียงนั้นๆอยู่แล้วกดปุ่ม Stop เครื่องก็จะกลับเข้ามาสู่โหมดการเล่นเพลง Title sound เช่นกัน

แบบที่ 4 เมื่ออยู่ในโหมดแล่นไฟล์เสียงนั้นๆอยู่แล้วกดปุ่ม Play/Pause เครื่องก็จะหยุดการเล่น แทร็ค นั้นๆชั่วคราวโดยจะขึ้นสัญลักษณ์ # อยู่หน้าเวลาของ แทร็ค นั้น

แบบที่ 5 เมื่ออยู่ในโหมดการหยุดเล่น แทร็ค ชั่วคราวแล้วกดปุ่ม Play/Pause อีกทีก็จะกลับไปสู่โหมดการเล่น แทร็ค นั้นต่อจากที่หยุดไว้

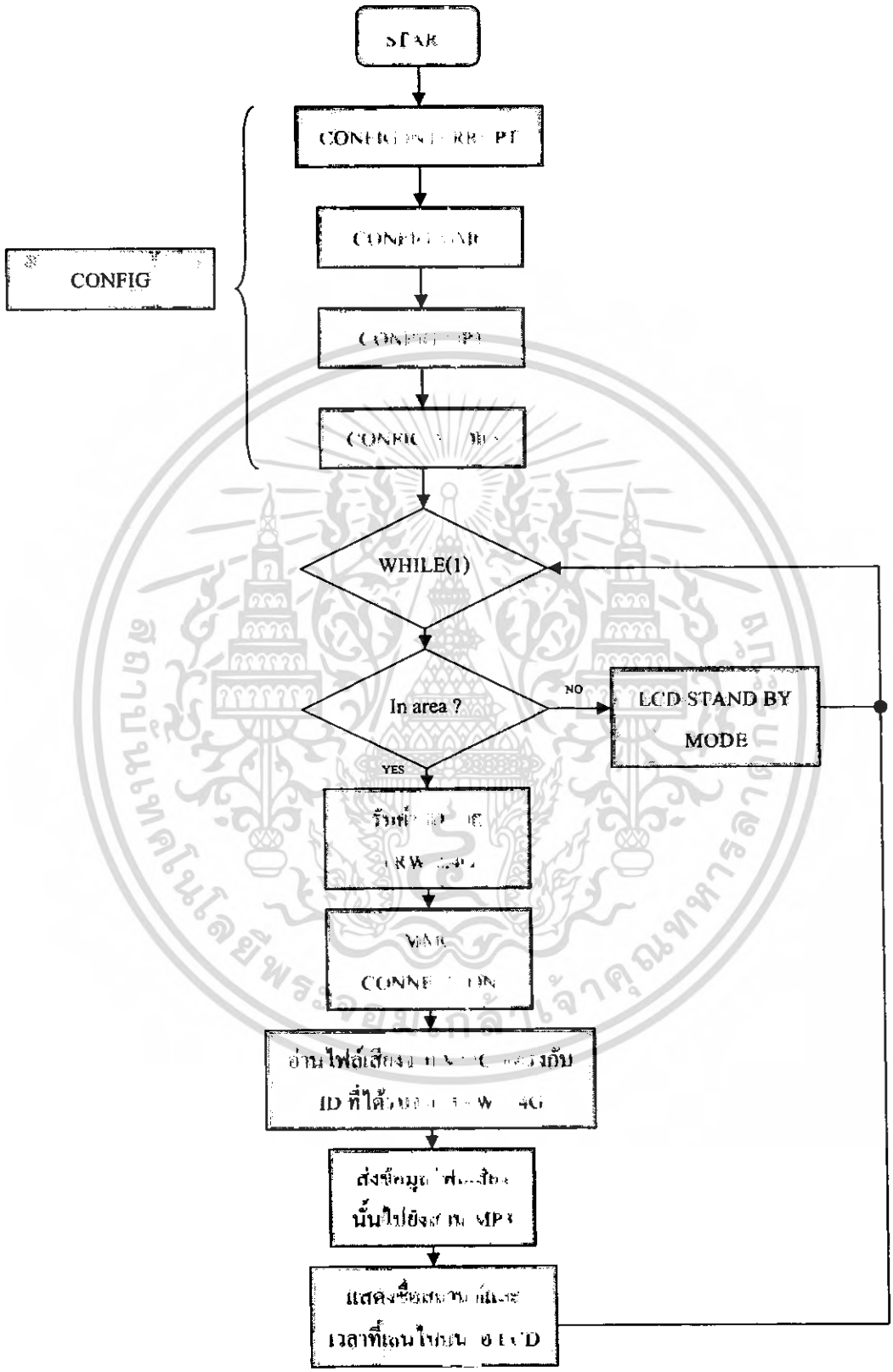
แบบที่ 6 เมื่ออยู่ในโหมดแล่นไฟล์เสียงนั้นๆอยู่แล้วกดปุ่ม รีเซต เครื่องก็จะหยุดการเล่น แทร็ค นั้นๆและกลับไปเล่น แทร็ค นี้ตั้งแต่ต้นอีกที

แบบที่ 7 สามารถกดปุ่มเพิ่มหรือลดเสียง (+/-) ได้ในทุกโหมด

แบบที่ 8 และเมื่อปล่อยปุ่มเพิ่มหรือลดเสียง (+/-) เป็นเวลา 2 วินาทีก็จะกลับเข้าสู่โหมดเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ออกแบบส่วนการทำงานของอุปกรณ์พกพา



รูปที่ 3.23 แสดงผังการทำงานของโปรแกรมส่วนอุปกรณ์พกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมในส่วนอปุกรณพกพานั้นในขั้นตอนแรกเราต้องทำการตั้งค่าเริ่มต้น (Configuration) ของส่วนต่างๆของอุปกรณ์ คือ

-ส่วนการทำงานของ อินเทอร์รัป

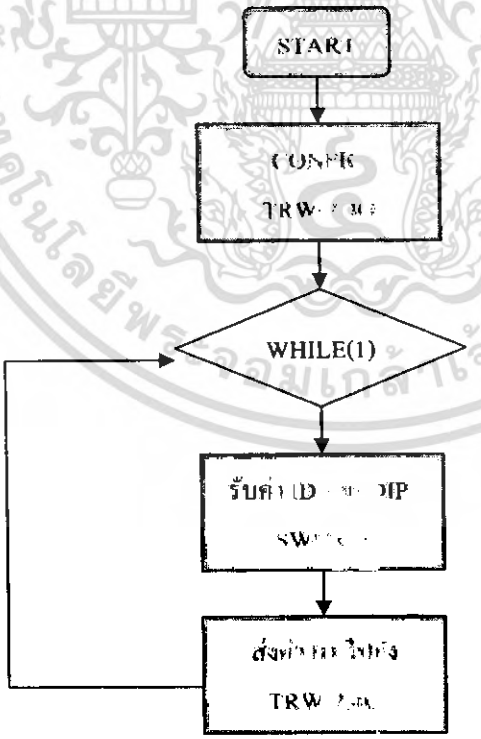
-ส่วนนการควบคุมเอ็มเอ็มซี

-ส่วนการควบคุมการทำงานกับไฟล์เสียงเอ็มพีสาม

-ส่วนการควบคุมส่งสัญญาณเสียงออก

เมื่อทำการตั้งค่าเริ่มต้นส่วนต่างๆเสร็จเรียบร้อยแล้วต่อไปโปรแกรมก็จะเข้าสู่ลูปการทำงานโดยเมื่อเข้าสู่ลูปการทำงานแล้วนั้นจะโปรแกรมควบคุมส่วนการทำงานของจอแอลซีดี ให้อยู่ในโหมด Stand by ต่อจากนั้นจะทำการรับค่าไอดี จากโมดูลการสื่อสารด้วยสัญญาณวิทยุ (TRW-2.4G) เพื่อรับค่าไอดี ที่ส่งมาจากสถานีสัญญาณจากนั้นเมื่อได้ค่าไอดีเรียบร้อยแล้วโปรแกรมจะนำไอดีนั้นไปดึงข้อมูลไฟล์เสียงเอ็มพีสามจากเอ็มเอ็มซีที่มีไอดีตรงกับไอดีที่ได้รับมาเพื่อนำข้อมูลไฟล์เสียงนั้นๆส่งไปยังส่วนการเล่นไฟล์เอ็มพีสามเพื่อส่งออกไปยัง ส่วนส่งสัญญาณเสียงออกหูฟังและส่งข้อมูลไฟล์เสียงที่เล่นนั้นไปยังหน้าจอแอลซีดีด้วย

3.3.4 ออกแบบส่วนการทำงานของสถานีสัญญาณ



รูปที่ 3.24 แสดงผังการทำงานของโปรแกรมส่วนสถานีสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการออกแบบโปรแกรมในส่วน สถานีสัญญาณ นั้นเริ่มต้นจะทำการตั้งค่าเริ่มต้น (Configuration) ของส่วนโมดูล TRW-2.4G ก่อนโดยตั้งค่าให้เป็นตัวส่งข้อมูล (TX mode) ซึ่งโปรแกรมจะวนลูปรับค่า ไอดี ซึ่งสามารถตั้งค่าได้จาก DIP-SWITCH 8 บิต แล้วจะทำการส่งข้อมูลไอดี ไปยังส่วนการส่งข้อมูลออกไปให้กับ อุปกรณ์พกพา ผ่านโมดูล TRW-2.4G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดลองเขียน AT89C51SND1.h

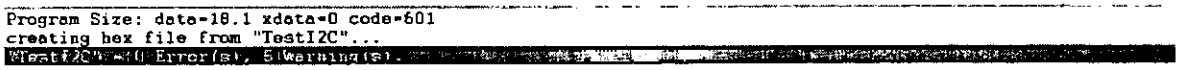
ขั้นตอนการทดลอง

- จากเอกสารของไมโครคอนโทรลเลอร์ AT89C51SND1C เราสามารถนำที่อยู่ของ รีจิสเตอร์ เหล่านั้นมาสร้างเชื่อมโยงกับชื่อของ รีจิสเตอร์ เพื่อง่ายต่อการใช้เขียนโปรแกรม โดยใช้ Keil เราจึงทดลองสร้างไฟล์ at89c51snd1.h
- นำไฟล์นี้มาทดสอบเขียนโปรแกรมอย่างง่าย แล้วอ้างอิง ชื่อ รีจิสเตอร์ต่างเพื่อทดสอบว่าไฟล์นี้สามารถใช้ ทำงานร่วมกับคอมไพล์เลอร์ได้จริง

รูปที่ 4.1 แสดงโปรแกรมที่ใช้ทำการทดลองของส่วนไฟล์เฮดเดอร์ AT89C51SND1.h

ผลการทดลอง

- เมื่อนำไฟล์นี้พบว่า สามารถใช้ ทำงานร่วมกับคอมไพล์เลอร์ได้จริง ไม่มีข้อผิดพลาด ดังภาพ



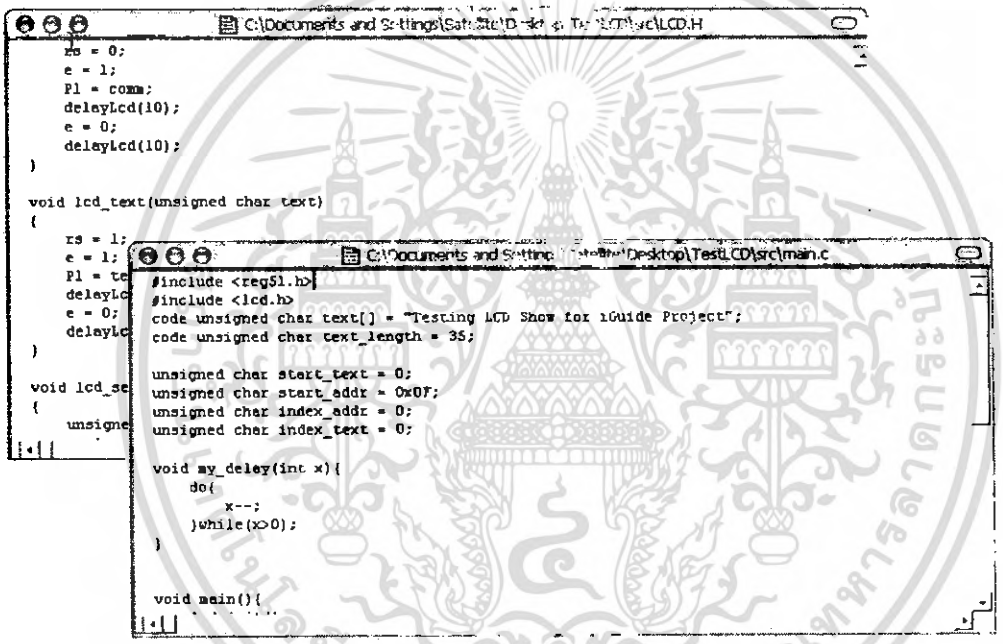
รูปที่4.2 สดงผลการทดลองส่วนไฟล์เฮดเดอร์ AT89C51SND1.h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดสอบโค้ดโปรแกรมส่วนแอลซีดี บนโปรแกรมจำลองการทำงานของไมโครคอนโทรลเลอร์ตระกูล 51

ขั้นตอนการทดลอง

- จากเอกสารของไมโครคอนโทรลเลอร์ AT89C51SND1C เรานำความรู้นั้นมาศึกษาและประยุกต์ใช้ ติดต่อกับจอ แอลซีดี เพื่อแสดงผล
- แล้วนำไฟล์โปรแกรมมาทดลองใช้กับโปรแกรม จำลองการทำงาน เพื่อทดสอบการทำงานของโปรแกรมว่าสามารถทำงานได้จริง โดยให้แสดงผลคำว่า "Testing LCD Show for iGuide Project" แล้วเลื่อนไปทางซ้ายมือจนจบประโยค

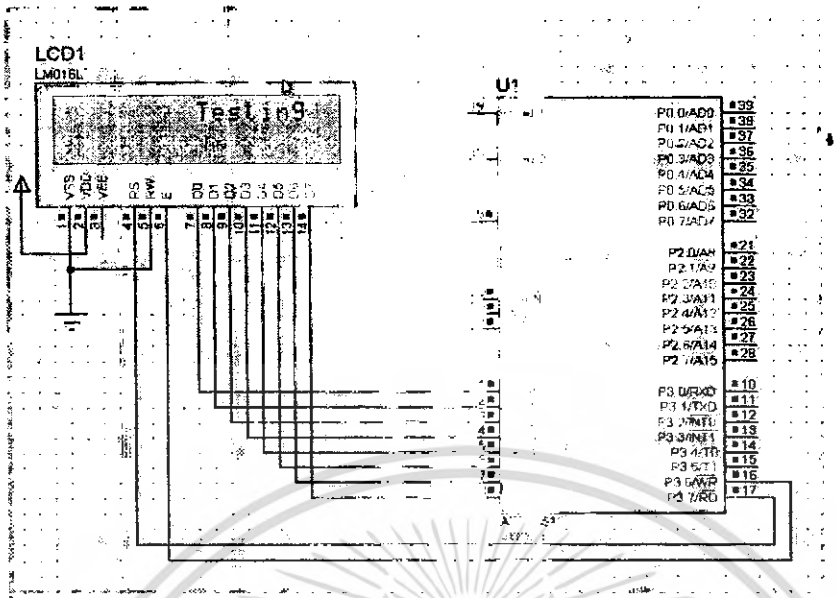


รูปที่4.3 แสดงโปรแกรมที่ใช้ทำการทดลองของส่วนจอแอลซีดี

ผลการทดลอง

- สำเร็จ โปรแกรมสามารถทำได้ตามจุดประสงค์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



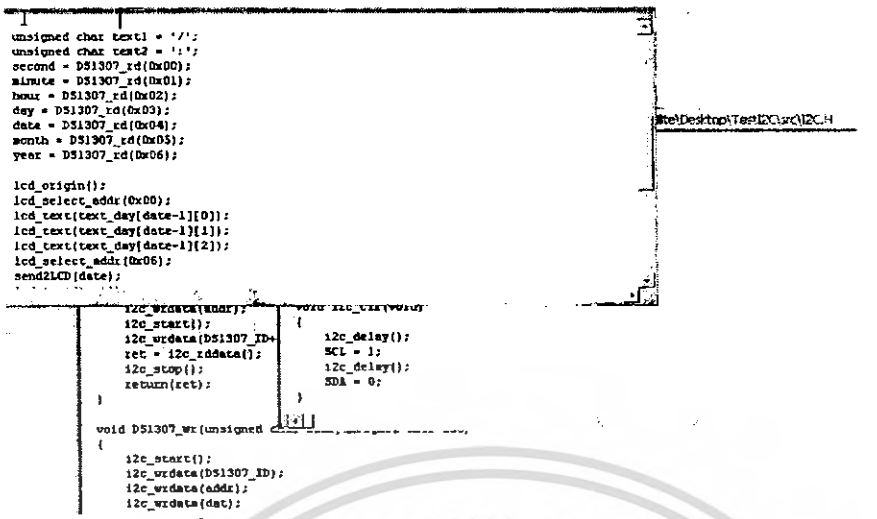
รูปที่ 4.4 แสดงผลการทดลองของส่วนจอแอลซีดีจากโปรแกรมจำลองการทำงาน

4.3 ทดสอบโค้ดไอสแควร์ซีบนโปรแกรมจำลองการทำงานของไมโครคอนโทรลเลอร์ตระกูล 51

ขั้นตอนการทดลอง

- จากเอกสารของไมโครคอนโทรลเลอร์ AT89C51SND1C เรานำความรู้นั้นมาศึกษาและประยุกต์ใช้ เพื่อติดต่อกับไอซีฐานเวลาจริง เรียกเวลาขึ้นมาแสดงบนจอแอลซีดี
- แล้วนำไฟล์โปรแกรมมาทดลองใช้กับโปรแกรม จำลองการทำงาน เพื่อทดสอบการทำงานของโปรแกรมว่าสามารถทำงานได้จริง โดยให้อ่านเวลามาจากไอซีฐานเวลาจริง แล้วนำมาแสดง วันในสัปดาห์ วันที่/เดือน/ปี และเวลา

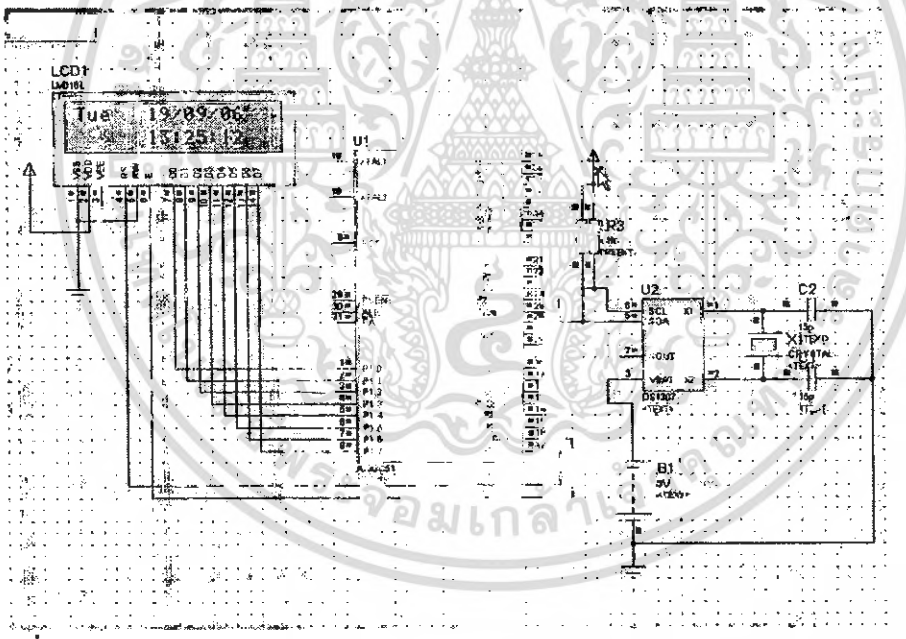
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงผลการทดลองของส่วนไอสแควร์ซี

<u>ผลการทดลอง</u>

- สำเร็จ โปรแกรมสามารถทำได้ตามจุดประสงค์ที่ต้องการ



รูปที่ 4.6 แสดงผลการทดลองของส่วนไอสแควร์ซีจากโปรแกรมจำลองการทำงาน

4.4 การทดลองภาคจ่ายไฟ

ขั้นแรกทดสอบการจ่ายไฟและอุปกรณ์ประกอบไมโครคอนโทรลเลอร์

4.4.1 ตรวจสอบการเชื่อมกัน และ ไม่เชื่อมกันของทุกจุดตามแบบ หลังลงอุปกรณ์แล้ว

<u>ผลการตรวจสอบ</u> ทุกจุดเป็นไปตามแบบที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 ตรวจสอบ ไฟ +5 V และ +3.3V ที่จุดต่างๆ เทียบกับ GND

ผลการตรวจสอบ ไฟ +5V ถูกต้องทุกจุดแต่ +3.3V ผิดพลาดบางจุดเป็น 0 บางจุด 4V บางจุด 2V

สาเหตุของปัญหา พบว่า Regulator 3.3 V ขา 1 2 และ 3 ผิดหน้าที่ สลับเปลี่ยนขากันอยู่

วิธีแก้ไข สลับขาให้ถูกต้องแต่ต้องย้าย Regulator 3.3 V ออกมานอกบอร์ด

ผลการแก้ไข ทุกจุดถูกต้องตามต้องการ

4.4.3 ตรวจสอบขา ISP ต่อกับ jumper ถูกต้อง

ผลการตรวจสอบ ต่อถูกต้องและ หากใส่ Jumper ขา ISP = 0 ไม่ใส่ ISP = 3.3V ถูกต้อง

4.4.4 ตรวจสอบส่วน คริสตอล

ผลการตรวจสอบ ต่อถูกต้องตามรูปแบบ

4.4.5 ตรวจสอบขา RST

ผลการตรวจสอบ ต่อถูกต้องและ หากกดปุ่ม รีเซต ขา RST = 3.3V ไม่กด RST = 0 V ถูกต้อง

4.4.6 ตรวจสอบส่วน ยูเอสบี

ผลการตรวจสอบ ต่อถูกต้องตามแบบและมีไฟเลี้ยง 5V ที่ขา 1 และ 0V ที่ขา 4

4.5 การทดลองใช้งานไมโครคอนโทรลเลอร์ผ่านไอเอสพี

4.5.1 ตรวจสอบการเชื่อมต่อ ยูเอสบี กับ คอมพิวเตอร์

ผลการตรวจสอบ คอมพิวเตอร์ ไม่สามารถหาพบอุปกรณ์ไม่เกิดอะไรขึ้นเลย

สาเหตุของปัญหา พบว่าขา D+ ต้องการ pull-up ขนาด 1.5 กิโลโอห์ม

วิธีแก้ไข เพิ่ม Pull-up ขนาด 1.0 กิโลโอห์ม ที่ขา D+ นอกบอร์ด

ผลการแก้ไข คอมพิวเตอร์ ค้นพบอุปกรณ์ต่อใหม่แต่หา ไดรเวอร์อุปกรณ์ ไม่ได้ เป็น ไม่รู้จักอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 ใช้งาน Flip 2.4.6 ทำการทดสอบการเชื่อมต่อ คอมพิวเตอร์

ผลการตรวจสอบ Flip 2.4.6 ไม่สามารถเปิดพอร์ตยูเอสบี ได้

4.5.3 ใช้งาน Flip 3.0 ทำการทดสอบการเชื่อมต่อ คอมพิวเตอร์

ผลการตรวจสอบ Flip 3.0 สามารถหาอุปกรณ์เจอเป็น ATMEL DFU ประเภท AT89C51SND1C/ 2C แต่ไม่สามารถเปิด Flip 3.0 ได้

สาเหตุของปัญหา พบว่า Flip 3.0 ต้องการ JVM 1.5

วิธีแก้ไข ดาวน์โหลดและติดตั้ง JRE 1.5

ผลการแก้ไข คอมพิวเตอร์ ค้นพบอุปกรณ์ ATMEL DFU

4.5.4 ทดลองโปรแกรมอย่างง่ายลงบน AT89C51SND1C

ผลการตรวจสอบ สามารถทำงานได้ตามโปรแกรมที่ได้โปรแกรมลงไป

4.6 การทดลองใช้งานสวิทช์

4.6.1 ทดสอบการทำงานของสวิทช์ทั้ง 6 ตัว

ผลการตรวจสอบ สามารถใช้งานได้เพียง 5 ตัว

สาเหตุของปัญหา ตัวที่ใช้งานไม่ได้เพราะขาของอุปกรณ์หักไป

4.7 การทดลองใช้งานส่วนเอ็มเอ็มซี

4.7.1 ทดสอบการทำงานร่วมกับเอ็มเอ็มซีโดยใช้คำสั่งผ่านทาง MCMD

ผลการตรวจสอบ ไม่สามารถใช้งานได้ ไม่มีการตอบสนองจากฝั่งเอ็มเอ็มซี

สาเหตุของปัญหา การตอบสนองเมื่อรับข้อมูลจะเข้ามาเป็น อินเทอร์รัป แต่จำเป็นต้องเปิด Mask bit ก่อน

วิธีแก้ไข โปรแกรมใหม่ แก้ไขที่ MMMSK = 0x00

ผลการแก้ไข มีการตอบสนองกลับจากเอ็มเอ็มซีว่ามีการรับข้อมูลเข้ามา (EORI = 1 เมื่อมีการรับข้อมูลเสร็จสิ้น)

4.7.2 ทดสอบการทำงานร่วมกับเอ็มเอ็มซีโดยใช้คำสั่งผ่านทาง MCMD แล้วทำการตรวจสอบข้อมูลที่ได้รับ

ผลการตรวจสอบ ข้อมูลที่ได้รับผิด ทั้งฟอร์แมทและ CRC

สาเหตุของปัญหา ต้องการตัวต้านทาน open-drain mode และปัญหาอื่นๆ ไม่ทราบที่มาอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีแก้ไข ต่อ ตัวต้านทาน open-drain mode และนำเอาตัวเก็บประจุออก เพิ่มตัว หน่วงเวลาในโปรแกรม เพิ่มหน่วงเวลาของ การตอบสนองเอ็มเอ็มซี

ผลการแก้ไข ข้อมูลเข้ามาของการตอบสนองพื้นฐานถูกต้อง ทั้ง CRC และ ฟอร์แมท

ตารางที่ 4.1 แสดงการใช้งานคำสั่งต่างและการตอบสนองของเอ็มเอ็มซี

คำสั่ง	ข้อมูลตอบสนอง
0x00	-
0x01	0x1F 00 7F 80 00 7F (6 Bytes) if busy
0x01	0x1F 80 7F 80 00 7F (6 Bytes) if idle
0x02	0x1F FF FF FF FF FF FF FF FF FF FF FF FF FF FF 03 (17 Bytes)
0x03	CRC และ Format ผิด

4.8 การทดลองส่วนสื่อสาร (TRW-2.4G)

4.8.1 ทดสอบส่วนฮาร์ดแวร์ของภาคการติดต่อสื่อสาร

ผลการตรวจสอบ ไม่สามารถส่งข้อมูลจากฝั่งส่งไปยัง ฝั่งรับได้

สาเหตุของปัญหา ไฟเลี้ยงผิดพลาดจากตอนเริ่มต้นการทดลองลองใช้ ไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C ซึ่งทำงานบนไฟเลี้ยง +5V ซึ่งไม่เข้ากันกับ โมดูล TRW-2.4G ที่ใช้สำหรับตัวภาคส่งและตัวรับ

วิธีแก้ไข แก้ไขไฟเลี้ยงโดยใช้ Regulator เบอร์ LM1117 3.3V เพื่อลดค่าความต่าง ศักย์ลงมาโดยที่กระแสยังสามารถใช้งานได้อยู่ และเปลี่ยนไมโครคอนโทรลเลอร์เป็นเบอร์ AT80C2051 ทำงานบนไฟเลี้ยง +3.3V แทน

ผลการแก้ไข สามารถรับและส่งข้อมูลขนาด 8 บิตผ่านสัญญาณวิทยุด้วยโมดูล TRW-2.4Gได้

4.8.2 ทดสอบส่วนซอฟท์แวร์ของภาคการติดต่อสื่อสาร

ทดลองเขียนโปรแกรมเพื่อใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์เบอร์ AT89C2051 เพื่อให้ให้ควบคุมตัวโมดูล TRW-2.4G (สามารถดูฟังก์ชันของโปรแกรม ทั้งหมดได้ในภาคผนวก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main()
{
    unsigned char id;          //define id variable
    Init_TRW24G();             //Initialize TRW-2.4G Module
    SetMode_TRW24(MODE_TX); //Set TRW-2.4G Module to TX mode
    while (1) {
        id = P1;               //Get id From DIP-SWITCH 8 bits
        Send_TRW24(id);        //Send that id
    }

}
```

รูปที่ 4.7 แสดงส่วนฟังก์ชันหลักที่ใช้ในการเขียนโปรแกรมสำหรับตัวส่ง

จากส่วนของโปรแกรมจะเห็นว่าจะเริ่มต้นด้วยการตั้งค่าเริ่มต้นให้กับโมดูล TRW-2.4G ก่อน จากนั้นทำการตั้งโหมดของโมดูลเป็นโหมดการส่งสัญญาณ

จากนั้นในส่วนของลูป while จะทำการอ่านค่าจาก พอร์ต1 ซึ่งต่อไว้กับ DIP-SWITCH เพื่อรับค่าการตั้งค่า ไอดี ของตัวเครื่อง สถานีสัญญาณ จาก ส่วนติดต่อกับผู้ใช้ภายนอก แล้วทำการส่งผ่านฟังก์ชั่น Send_TRW24() ออกไป

```
void main()
{
    unsigned char id;          //Define id variable
    Init_TRW24G();             //Initialize TRW-2.4G module
    SetMode_TRW24(MODE_RX); //Set TRW-2.4G Module to RX mode
    while (1) {
        while (!DR1)()         //Wait for DR1=1
        id = Read_TRW24();     //Recieve id from TRW-2.4G Module
        P1 = id;               //Send id to connector and show id to LED
    }

}
```

รูปที่ 4.8 แสดงส่วนฟังก์ชันหลักที่ใช้ในการเขียนโปรแกรมสำหรับตัวรับ

จากส่วนของโปรแกรมจะเห็นว่าจะเริ่มต้นด้วยการตั้งค่าเริ่มต้นให้กับโมดูล TRW-2.4G ก่อน จากนั้นทำการตั้งโหมดของโมดูลเป็นโหมดการรับสัญญาณ

จากนั้นในส่วนของลูป while จะต้องรอให้ขา DR1 (Data Ready1) ของโมดูลมีค่าเป็น I ก่อนซึ่งหมายความโมดูลอีกฝั่งพร้อมจะส่งข้อมูลมายังโมดูลของเราแล้ว จากนั้นทำการรับค่า ไอดี ด้วยฟังก์ชั่น Read_TRW24() และนำไปแสดงผลออกทาง หลอดแอลอีดี เพื่อตรวจสอบความถูกต้องของ ไอดี และเพื่อต่อออกไปยัง คอนเนกเตอร์ เพื่อต่อไปยังส่วนเล่นเอ็มพีสามของ อุปกรณ์พกพา ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การทดลองส่วนสัญญาณเสียง

4.9.1 ทดสอบส่วนฮาร์ดแวร์และการใช้งานของไอซีเสียง

ผลการตรวจสอบ สามารถบันทึกและเล่นเสียงได้ตาม แอดเดรส ที่กำหนด

4.9.2 ทดสอบไอซีเสียงร่วมกับไมโครคอนโทรลเลอร์โดยใช้สวิตท์ที่ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการเลือกเล่นเสียงแบบต่อเนื่องและใช้ แอลซีดี แสดงผลตัวเลขที่เล่นอยู่ขณะนั้น

ผลการตรวจสอบ สามารถบันทึกและเล่นเสียงได้ตามการควบคุมของสวิตท์ และแอลซีดี แสดงผลได้ถูกต้อง

4.10 การทดสอบการทับกันของการส่งไอดีของสถานีสัญญาณและการรวมระบบ

4.10.1 ทดสอบโดยนำส่วนรับ-ส่งสัญญาณวิทยุรวมเข้ากับส่วนแสดงสัญญาณเสียงและแอลซีดี

ผลการตรวจสอบ สามารถทำงานได้แต่เกิดปัญหาเมื่อสองสถานีอยู่ในบริเวณเดียวกันเกิดการรบกวนกันทำให้เสียงที่ออกมาสลับไปมาระหว่างสองสถานี

สาเหตุของปัญหา ข้อมูลที่ได้รับจากสองสถานีรบกวนกัน

วิธีแก้ไข แก้ไขในซอฟแวร์ให้รับรู้ถึงการรบกวนกันของสถานี โดยหากอยู่ในส่วนที่รบกวนกันให้ดำเนินการทำงานเดิมต่อไป และจะเปลี่ยนสถานีเมื่อได้รับสัญญาณจากเพียงแหล่งเดียว

ผลการแก้ไข สามารถทำงานได้ถูกต้องตามต้องการ แก้ปัญหาการรบกวนกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1. บทสรุป

จากการทำโครงงานมัคคุเทศก์อัจฉริยะนี้เราจะเข้าใจถึงเทคโนโลยีทุกอย่างที่ใช้ คือ ส่วนของการใช้งานไมโครคอนโทรลเลอร์ AT89C51SND1C โดยได้ทำการศึกษาทั้งโครงสร้างภายนอก ขาสัญญาณต่างๆ โครงสร้างภายในส่วนของหน่วยความจำ สัญญาณนาฬิกา ส่วนการถอดรหัสเอ็มพีสาม ส่วนของเอ็มเอ็มซี และส่วนของส่งสัญญาณเสียงออก (Audio Output) ซึ่งเราได้ทำการศึกษาทั้งรีจิสเตอร์ต่างๆที่เกี่ยวข้อง หลักการใช้งาน หลักการเขียนโปรแกรมควบคุม อีกส่วนหนึ่งที่สำคัญคือส่วนของการใช้งานเอ็มเอ็มซีซึ่งเราได้ทำการศึกษาทั้งลักษณะทั่วไปของตัวเอ็มเอ็มซีเอง หลักการเชื่อมต่อและโปรโตคอล การเขียนคำสั่งและการทำงานทั้งการอ่าน เขียน และลบข้อมูลบนการ์ด ส่วนต่อมาทำการศึกษาหลักการเชื่อมต่อภายนอกแบบเข้าจังหวะ (SPI : Synchronous Peripheral Interface) ซึ่งเป็นกระบวนการติดต่อสื่อสารระหว่างอุปกรณ์ ด้วยอัตราการสื่อสารที่มีความเร็วต่ำ คล้ายการติดต่อสื่อสารแบบ I^2C แต่มีลักษณะ Full-Duplex คือสามารถรับและส่งพร้อมกันได้ และมีความเร็วสูงกว่า และส่วนสุดท้ายวิธีการใช้งานโมดูล TRW-2.4G เพื่อติดต่อสื่อสารข้อมูลผ่านสัญญาณวิทยุ โดยการควบคุมผ่านไมโครคอนโทรเลอร์ AT89C51

จากนั้นได้ทำการทดลองเขียนโปรแกรมควบคุมไอสแควร์ซี จอแอลซีดี และลองทดสอบต่ออุปกรณ์ในส่วนของโมดูล TRW-2.4G บนโพรโตบอร์ดจนสามารถใช้งานได้ จึงได้ทำการออกแบบผังวงจร (Schematic design) ของแต่ละส่วนของอุปกรณ์ เมื่อได้ผังวงจรแต่ละส่วนแล้ว ก็ทำมารวมเข้าด้วยกันเป็นผังวงจรรวมของระบบทั้งหมด เมื่อได้ผังวงจรรวมแล้วก็สามารถออกแบบลายวงจร ได้ และเมื่อได้ข้อสรุปเป็นลายวงจร แล้วนั้นเราจะสามารถไปซื้ออุปกรณ์ (ไอซีต่างๆ) เพื่อทำบอร์ดจริงได้ และสามารถสั่งทำแผ่นวงจรได้

ในส่วนของตัวรับข้อมูลและเล่นคำบรรยายนั้น ผู้พัฒนาได้ทำการออกแบบวงจร และทำการทดสอบแต่ละส่วน โดยในส่วนของการติดต่อกับเครื่องคอมพิวเตอร์เพื่อโปรแกรมการทำงานลงไปให้กับไมโครคอนโทรลเลอร์ ใช้โปรโตคอล ไอเอสพี นั้นสามารถทำงานได้ และสามารถควบคุมไมโครคอนโทรลเลอร์ให้ทำงานได้ตามต้องการ แต่ในการติดต่อกับเอ็มเอ็มซีนั้นเกิดปัญหากับโปรโตคอลเอ็มเอ็มซี เนื่องจากวงจรไม่สามารถทำให้การ์ดตอบรับได้ในคำสั่งเลือกการ์ด และผลมาจากการที่ไม่สามารถติดต่อกับการ์ดได้ ทำให้ไม่สามารถนำข้อมูล เอ็มพีสาม มาทดลองกับส่วนของการสร้างเสียงได้ เพราะข้อมูล เอ็มพีสาม ไม่สามารถสร้างจำลองขึ้นมาเองภายในไมโครคอนโทรลเลอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายจึงต้องทำการแก้ไขโดยใช้ไอซีเสียง เป็นส่วนบันทึกเสียงแทน เพราะสามารถใช้งานได้ง่ายและเกิดความผิดพลาดได้น้อยกว่า แต่มีข้อเสียตรงไม่สามารถบันทึกเสียงที่มีเวลานานได้ แต่สามารถทำให้ผู้ใช้รับรู้ถึงสถานี และสถานที่ที่ตนอยู่ประกอบกับข้อมูลบางส่วน ทำให้โครงงานนี้สามารถนำมาใช้งานได้ตามวัตถุประสงค์ที่ตั้งไว้

5.2. ปัญหาอุปสรรคและแนวทางแก้ไข

ทางด้านอุปสรรคและแนวทางการแก้ไขในการทำงานนั้น มีดังนี้คือ

5.2.1 ปัญหาด้านการหาข้อมูลทางเทคนิคต่างๆ

ข้อมูลที่หาได้มักไม่ตรงตามที่ต้องการและมักไม่เป็นข้อมูลเชิงเทคนิคเพราะส่วนใหญ่มักจะเป็นข้อมูลเชิงการค้าเสียมากกว่า แนวทางการแก้ไขคือ การค้นหาดาต้าชีทจากบริษัทผู้ผลิตซึ่งจะทำให้เราได้ข้อมูลทางเทคนิคที่ตรงตามที่ต้องการมากกว่า

5.2.2 อุปกรณ์บางตัวไม่สามารถหาซื้อได้ในประเทศไทย

เมื่อเราสามารถหาอุปกรณ์ตรงตามคุณสมบัติที่เราต้องการได้แล้วนั้นกลับพบปัญหาต่อมาคือ อุปกรณ์บางตัวนั้นไม่มีผู้ค้านำเข้ามาขายในประเทศไทย ทำให้เรานั้นต้องทการสั่งซื้อทางไปรษณีย์ ซึ่งต้องเสียค่าจัดส่งและเสียเวลาในการรอสินค้าแต่ละรอบที่จะส่งมาให้ทางเรือ

5.2.3 อุปกรณ์บางตัวไม่มีลายขาอุปกรณ์ที่จะใช้ทำการออกแบบ แผ่นลายวงจร

เมื่อได้อุปกรณ์ที่ต้องการแล้ว เนื่องจากอุปกรณ์นั้นไม่เป็นที่นิยมในท้องตลาดเท่าใดนัก ทำให้ไม่สามารถใช้ลายขาอุปกรณ์ในโปรแกรมที่ใช้ออกแบบ แผ่นลายวงจร ได้ ทำให้ต้องวัดขนาดและสร้างลายขาอุปกรณ์ขึ้นเอง

5.2.4 ปัญหาเกี่ยวกับแพ็กเกจของตัวไมโครคอนโทรลเลอร์เบอร์ AT89C51SND1C และของเอ็มเอ็มซีสล็อต

เนื่องจากไมโครคอนโทรลเลอร์ที่ใช้นั้นเป็นแพ็กเกจแบบ TQPF80 และสล็อตของเอ็มเอ็มซีก็เป็นแบบเซอร์เฟซเมาท์ (surface mount) ซึ่งไม่สามารถบัดกรีเองหรือต่อลงโปรโตบอร์ดได้ ทำให้เกิดความยากลำบากในการทดลองเพราะไม่สามารถนำโปรแกรมที่เขียนเพื่อทดลองใช้งานไมโครคอนโทรเลอร์และเอ็มเอ็มซี อย่างง่ายๆก่อนได้ ดังนั้นเราจึงแก้ปัญหาด้วยการออกแบบ แผ่นลายวงจร ทั้งหมดให้เสร็จก่อนเพื่อจะได้ทำการลงอุปกรณ์รอบเดียวเพื่อจะได้ทำการทดลองด้วยโปแกรมง่ายๆได้

5.2.5 ปัญหาเกี่ยวกับส่วนของภาคจ่ายไฟ

เรกูเลเตอร์ 3.3V นั้นมีการวางตำแหน่งของขาไม่เหมือนเรกูเลเตอร์โดยทั่วไปจึงต้องมีการต่อสายออกมานอกบอร์ดเพื่อนสลับขาของ ให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.6 ปัญหาเกี่ยวกับส่วนใช้งานไมโครคอนโทรลเลอร์ผ่าน ไอเอสพี

เกิดปัญหาเกี่ยวกับการเชื่อมต่อกับคอมพิวเตอร์ซึ่งเป็นปัญหาส่วนของ ไดรเวอร์อุปกรณ์ แต่สามารถแก้ไขได้โดยการดาวโหลดไดรเวอร์ของจาวาที่เวอร์ชั่นใหม่กว่ามาคือ JVM1.5

5.2.7 ปัญหาเกี่ยวกับการติดต่อกับเอ็มเอ็มซี

การติดต่อเอ็มเอ็มซี สามารถทำได้ 2 รูปแบบ คือ ด้วยโปรโตคอลของเอ็มเอ็มซีเอง และโปรโตคอลเอสพีไอ โปรโตคอลเอ็มเอ็มซี นั้นมีประสิทธิภาพที่ดีกว่าและไมโครคอนโทรลเลอร์ที่เลือกใช้มีฮาร์ดแวร์ในส่วนที่จัดการโปรโตคอลนี้ด้วยจึงทำการเลือกใช้ แต่เนื่องจากโปรโตคอลนี้ไม่ได้รับความนิยม ทำให้ไม่มีข้อมูลที่แน่นอน และในเมืองไทยไม่มีเอกสารที่เขียนเกี่ยวกับโปรโตคอลนี้เลย แต่ได้ทำการศึกษาจากเอกสารจากผู้ผลิตเอ็มเอ็มซี ในต่างประเทศ พบว่ามีหลายเอกสารที่ไม่ตรงกัน จึงได้ทำการทดลองหลายต่อหลายครั้ง แต่ไม่ประสบความสำเร็จ สามารถติดต่อได้บางคำสั่ง ซึ่งมีการตอบรับที่ถูกต้อง แต่คำสั่งสำคัญในการเลือกการ์ดไม่สามารถทำได้ จึงต้องทำการเปลี่ยนโปรโตคอลที่ใช้

5.2.8 ปัญหาเกี่ยวกับข้อมูล เอ็มพีสาม

เนื่องมาจากข้อมูล เอ็มพีสามไม่สามารถสร้างจำลองขึ้นเองได้ ทำให้เมื่อไม่สามารถติดต่อเอ็มเอ็มซีได้ก็ไม่สามารถนำข้อมูลเอ็มพีสามออกมาได้ ทำให้การทดสอบส่วนการทำงานของเอ็มพีสามและส่วนส่งสัญญาณเสียงไม่สามารถทดสอบและพัฒนาต่อได้ จึงได้นำเอาไอซีเสียงเข้ามาบันทึกและเล่นเสียงแทนส่วนของเอ็มพีสามและเอ็มเอ็มซี เพราะใช้งานได้ง่าย เกิดความผิดพลาดน้อย ทำให้โครงงานนี้สามารถพัฒนาต่อจนใช้งานได้

5.2.9 ปัญหาพื้นที่การของส่วนส่งสัญญาณของสถานีส่งสัญญาณซ้อนทับกัน

ส่วนของการส่งสัญญาณนั้นจะมีส่วนที่ซ้อนทับกันอยู่ ซึ่งเราต้องทางแก้ไขโดยการเขียนโปรแกรมแก้ไขในจุดนี้ให้ตัวอุปกรณ์พกพานั้นประมวลการทำงานที่ทำอยู่ให้จบก่อนที่จะรับข้อมูลใหม่เข้ามาและให้อุปกรณ์นั้นทำการตรวจสอบให้แน่ใจว่าอุปกรณ์ได้เข้ามาอยู่ในพื้นที่หนึ่งๆแล้ว

5.3. แนวทางการพัฒนาต่อ

โครงงานเครื่องให้ข้อมูลอัตโนมัติหรือมัคคุเทศก์อัจฉริยะนี้ ได้รับการออกแบบ ทดสอบและพัฒนาแล้ว อีกทั้งมีข้อมูลด้านเทคนิคครบถ้วน สามารถนำไปพัฒนาต่อเพื่อการใช้ประโยชน์จริง ปัญหาและอุปสรรคที่ได้รับได้ทำการบันทึกไว้ เพื่อให้ผู้พัฒนาต่อได้ตรวจสอบและแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงจุด โครงงานนี้สามารถนำไปพัฒนาต่อเป็น เครื่องให้ข้อมูลในการเข้าชมงานในนิทรรศการ ในพิพิธภัณฑ์หรือเยี่ยมชมโรงงาน และยังสามารถนำไปปรับปรุงด้านความเร็วในการติดต่อสื่อสาร และจำกัดวงของการสื่อสารให้เล็กลงจะทำให้เกิดความละเอียดในการนำเสนอมากขึ้น การปรับปรุงส่วนติดต่อผู้ใช้ หรือทำให้สะดวกต่อการพกพา สามารถนำโครงงานนี้ไปพัฒนาต่อได้

****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Atmel Corporation. "AT89C51SND1 details." [Online].
Available : http://www.atmel.com/dyn/products/product_card.asp?part_id=2598. 2006

[2] Sandisk Corporate. "AT89C51SND1 Details." [Online].
Available : www.compsys1.com/support/docs/sandisk.pdf. 2006

[3] ElecToday.com. "Electoday" [Online].
Available : http://www.electoday.com/modules.php?name=Forums&file=viewtopic&t=907. 2006.

[4] Micro4dev.com. "รู้จักกับพอร์ต ยูเอสบี" [Online].
Available : http://www.micro4dev.com/www/index.php?option=com_content&task=view&id=95&Itemid=46. 2004.

[5] ComSys.com. "MMC Project" [Online].
Available : http://www.compsys1.com/workbench/On_top_of_the_Bench/MMC_Project/mmc_project.html. 2006.

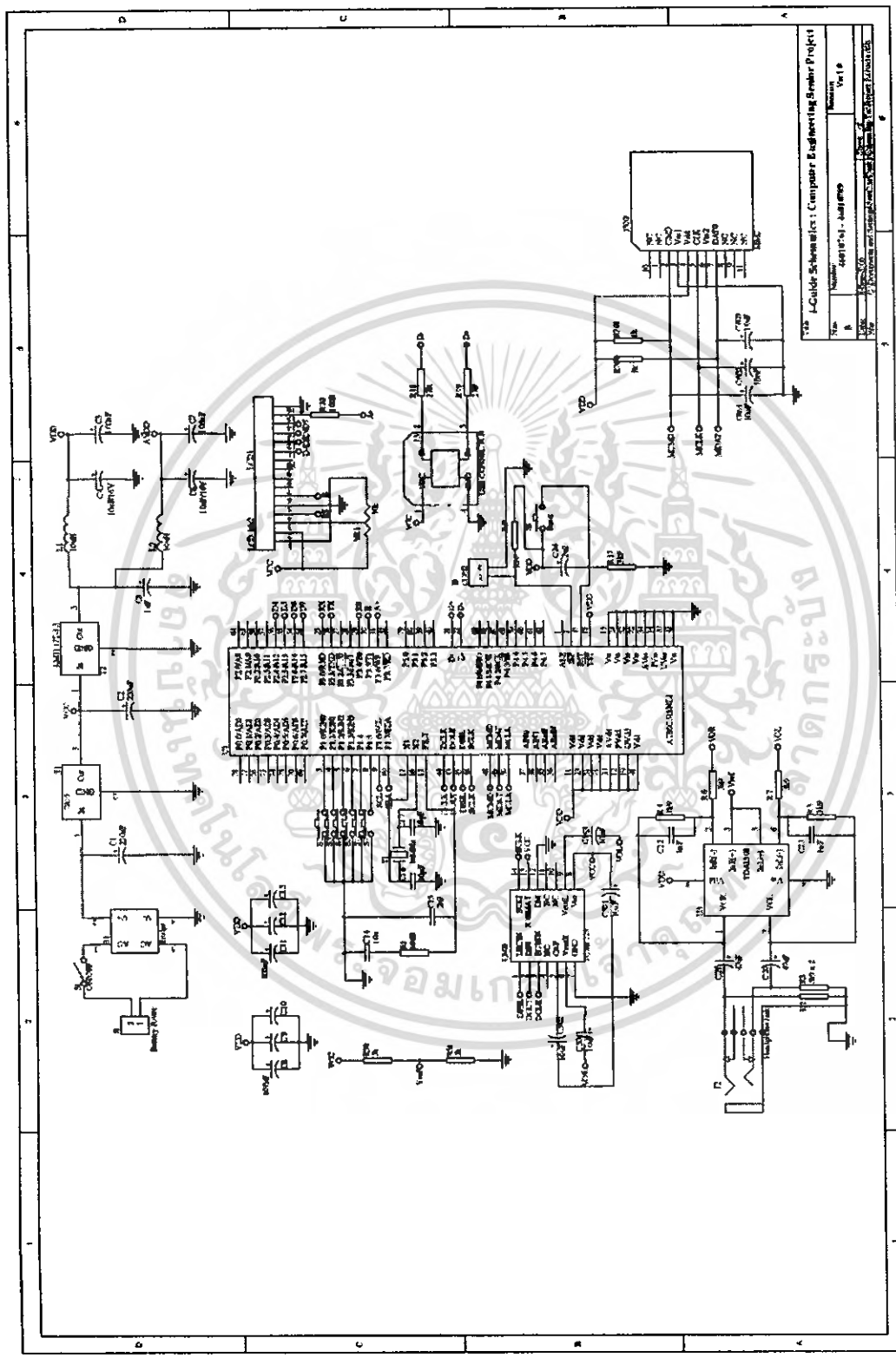
[6] Electronics Source Co,Ltd. "High Quality Product" [Online].
Available : http http://www.es.co.th/Detail.asp?Prod=AT89C51SND1C%2DROTIL. 2005.

[7] ลภน สุภาพ, อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ต ยูเอสบี ขั้นพื้นฐาน** สำนักพิมพ์ อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด กรุงเทพฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

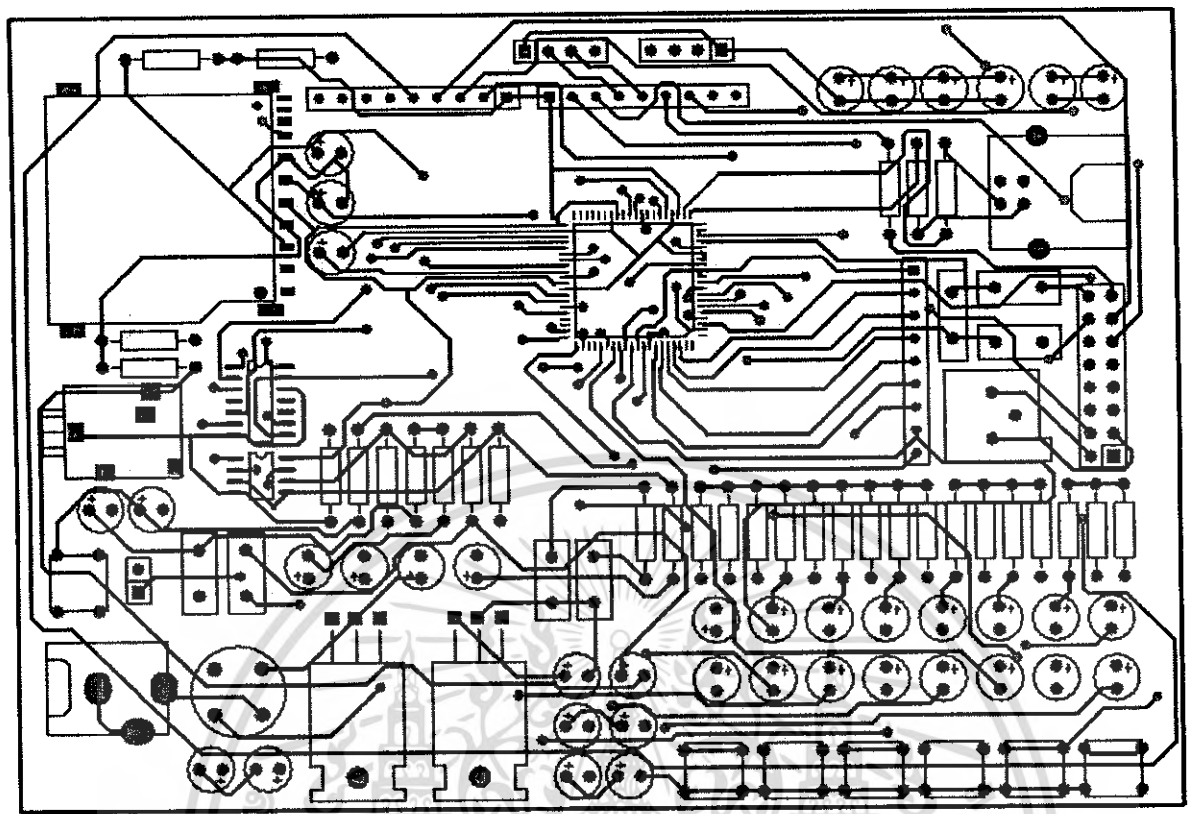
ภาคผนวก ก

ผังวงจรและลายวงจรส่วนต่างๆ

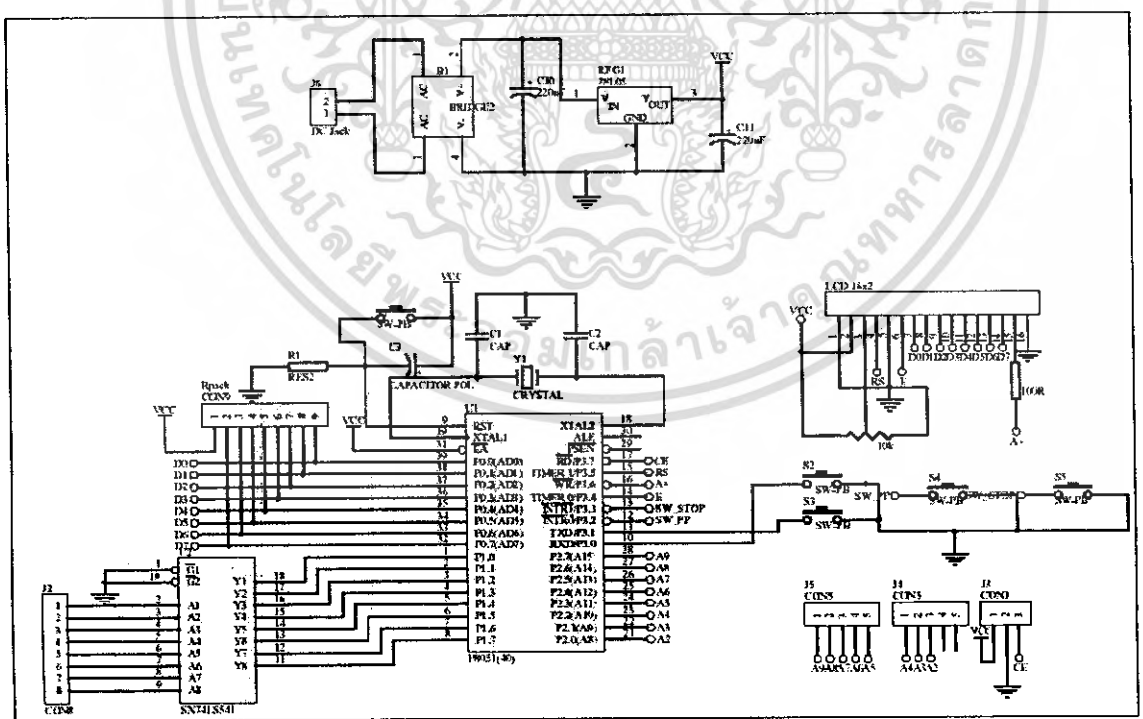


ภาคผนวก ก รูปที่ 1 แสดงผังวงจรของส่วน อุปกรณ์พกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก รูปที่ 2 แสดงลายวงจรของส่วน อุปกรณ์พกพา



ภาคผนวก ก รูปที่ 3 แสดงผังวงจรของส่วน อุปกรณ์พกพา
เมื่อเปลี่ยนเป็นใช้ไอซีเสียงเบอร์ ISD2590

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ซอร์สโค้ดโปรแกรมควบคุมฮาร์ดแวร์

```
#include <reg51.h>

sbit CS = P3^2;
sbit CE = P3^3;
sbit DAT = P3^4;
sbit CLK = P3^5;
sbit DR1 = P3^7;
sbit latch = P3^0;

#define MODE_TX 0
#define MODE_RX 1

void Wait(unsigned char n)
{
    while (n--) ;
}

void Init_TRW24G(void)
{
    CE  = 0;
    CS  = 0;
    CLK = 0;
    DAT = 0;
    DR1 = 0;
    Wait(1);
}

void CLK_TRW24(void)
{
    CLK = 0;
    CLK = 1;
}

void Write_TRW24(unsigned char Data)
{
    unsigned char i;
    bit Out;
    for (i=0;i<8;i++)
    {
        Out = Data & 0x80;
        DAT = Out;
        CLK_TRW24();
        Data = Data << 1;
    }
}
unsigned char Read_TRW24(void)
{
    unsigned char i,Temp;
    bit Out;
    DAT = 1;
    for (i=0;i<8;i++)
    {
        Temp = Temp << 1;
        CLK = 1; Wait(1);
        Out = DAT;
        if (Out) { Temp = Temp + 0x01;}
        CLK = 0; Wait(1);
    }
    return(Temp);
```

ภาคผนวก ข รูปที่ 1 แสดงฟังก์ชันต่างๆที่ใช้ในการเขียนโปรแกรมควบคุม TRW-2.4G (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void SetMode_TRW24( unsigned char Mode)
{
    Wait(1);
    CE = 0;
    CS = 1;
    Write_TRW24(0x8E); /* Reserved for testing */
    Write_TRW24(0x08); /* Reserved for testing */
    Write_TRW24(0x1C); /* Reserved for testing */

    Write_TRW24(0x08); /* Length of Bit Ch 2 */
    Write_TRW24(0x08); /* Length of Bit Ch 1 */

    Write_TRW24(0xD1); /* Address 5 Byte Ch 2 */
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);

    Write_TRW24(0xB5); /* Address 5 Byte Ch 1 */
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);

    Write_TRW24(0xA3); /* Number of Address bit + CRC */
    Write_TRW24(0x4F); /* RF Programming */
    Write_TRW24(0x0A+Mode);

    DAT = Mode;
    DR1 = Mode;
    CE = Mode;

    CS = 0;
    Wait(1);
}

void Send_TRW24(unsigned char Data)
{
    Wait(1);
    CS = 0;
    CE = 1;
    Write_TRW24(0xB5); /* Address 5 Byte Ch 1 */
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);

    Write_TRW24(Data); /* Data 8 bit */
    Wait(1);
    CLK = 0;
    CE = 0;
    Wait(1);
}
```

ภาคผนวก ข รูปที่ 2 แสดงฟังก์ชันต่างๆที่ใช้ในการเขียนโปรแกรมควบคุม TRW-2.4G (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define SOUND_ADDR P2
void delay(int tick)
{
    unsigned int i,j;
    for(i=0;i<tick;i++)
    for(j=0;j<250;j++);
}
sbit SOUND_CE = P3^7;
/*void sound_delay(){
    unsigned char i,j;
    for(i=0;i<100;i++){
        for(j=0;j<255;j++);
    }
}*/

void sound_init(){
    SOUND_CE = 1;
    SOUND_ADDR = 0x00;
}

void play_addr(unsigned char addr){
    SOUND_ADDR = addr;
    SOUND_CE = 0;
    //sound_delay();
    SOUND_CE = 1;
}
```

ภาคผนวก ข รูปที่ 3 แสดงโปรแกรมควบคุมส่วนการส่งสัญญาณเสียงออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define LCD_PORT P0
sbit e = P3^4;  //Edit to E-pin
sbit rs = P3^5; //Edit to RS-pin
void delayLcd(int tick)
{
    unsigned int i,j;
    for(i=0;i<tick;i++)
    for(j=0;j<250;j++);
}
void lcd_command(unsigned char comm)
{
    rs = 0;
    e = 1;
    LCD_PORT = comm;
    delayLcd(10);
    e = 0;
    delayLcd(10);
}
void lcd_text(unsigned char text)
{
    rs = 1;
    e = 1;
    LCD_PORT = text;
    delayLcd(10);
    e = 0;
    delayLcd(10);
}
void lcd_select_addr(unsigned char addr)
{
    unsigned char real;
    real = 0x80 + addr;
    lcd_command(real);
}
void lcd_clear(void)
{
    lcd_command(0x01);
}
void lcd_origin(void)
{
    lcd_command(0x02);
}
void lcd_init(void)
{
    delayLcd(250); //not 500 because char = 1 byte 0-255
    delayLcd(250);
    lcd_command(0x38);
    lcd_command(0x0C);
    lcd_command(0x01);
}
```

ภาคผนวก ข รูปที่ 4 โปรแกรมส่วนควบคุมจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <reg51.h>
#include <lcd.h>
#include <sound.h>
#define IN_RF P1
code unsigned char first_line[] = "I Guide";
code unsigned char volume_line[] = "Volume";
code unsigned char station[] = "Station : ";
code unsigned char no_signal[] = "No Signal";
code unsigned char sound_addr[] = {
    0x00,   /*0 Station*/
    0x03,   /*1*/
    0x05,   /*2*/
    0x07,   /*3*/
    0x09,   /*4*/
    0x0B,   /*5*/
    0x0D,   /*6*/
    0x0F,   /*7*/
    0x20,   /*8*/
    0x22,   /*9*/
    0x24,   /*10*/
    0x26,   /*11 100*/
    0x28,   /*12 2-*/
    0x2A,   /*13 -1*/
    0x2C};  /*14 No Station*/

sbit sw_up =P3^0;
sbit sw_down =P3^1;
sbit sw_pp =P3^2;
sbit sw_stop =P3^3;
bit update = 1;
unsigned char addr = 0;
unsigned char addr_sh = 0;
unsigned char vol_level = 5;
unsigned char input_rf=0;

void sendNum2LCD(unsigned char num){
    unsigned char x = 0;
    x = num/100;
    lcd_text(0x30 | x);
    x = (num%100)/10;
    lcd_text(0x30 | x);
    x = num%10;
    lcd_text(0x30 | x);
}
void sendNum2Sound(unsigned char num){
    unsigned char x;
    unsigned char y;
    unsigned char z;
    bit have = 0;
    play_addr(sound_addr[0]);
    delay(1200);
    x = num/100;
    if(x>0){
        play_addr(sound_addr[x]);
        delay(500);
        play_addr(sound_addr[11]);
        delay(500);
        have = 1;
    }
    y = (num%100)/10;
    if(y>0){
        if(y>2){
            play_addr(sound_addr[y]);
            delay(750);
        }else if(y>1){
            play_addr(sound_addr[12]);
            delay(750);
        }
        play_addr(sound_addr[10]);
        delay(500);
        have = 1;
    }
    z = num%10;
    if(z>0){
        if(z==1 && have)
            play_addr(sound_addr[13]);
        else
            play_addr(sound_addr[z]);
        delay(500);
    }
}
```

ภาคผนวก ข รูปที่ 5 แสดงโปรแกรมส่วนควบคุมไมโครคอนโทรลเลอร์ MCS51 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main(){
    unsigned char i = 0;
    unsigned char temp = 0;
    unsigned char index_sound = 0;
    lcd_init();
    sound_init();
    lcd_select_addr(0x00);
    for(i=0;i<7;i++) lcd_text(first_line[i]);
    while(1){
        input_rf = IN_RF;
        temp = input_rf&0xC0;
        if(temp == 0x80){
            input_rf = input_rf&0x3F;
        }else{
            input_rf = 0x00;
        }
        if(addr != input_rf){
            addr = input_rf;
            update = 1;
        }
        if(update){
            lcd_clear();
            lcd_select_addr(0x00);
            for(i=0;i<7;i++) lcd_text(first_line[i]);
            if(addr == 0x00){
                lcd_select_addr(0x40);
                for(i=0;i<9;i++) lcd_text(no_signal[i]);
                play_addr(sound_addr[14]);
                delay(100);
            }else{
                lcd_select_addr(0x40);
                for(i=0;i<7;i++) lcd_text(station[i]);
                lcd_select_addr(0x48);
                sendNum2LCD(addr);
                sendNum2Sound(addr);
            }
            update = 0;
        }
        if(sw_up==0){
            lcd_clear();
            if(vol_level<10) vol_level++;
            lcd_select_addr(0x00);
            for(i=0;i<6;i++) lcd_text(volume_line[i]);
            lcd_select_addr(0x40);
            for(i=0;i<vol_level;i++) lcd_text('#');
            while(sw_up==0);
            update = 1;
        }
        if(sw_down==0){
            lcd_clear();
            if(vol_level>0) vol_level--;
            lcd_select_addr(0x00);
            for(i=0;i<6;i++) lcd_text(volume_line[i]);
            lcd_select_addr(0x40);
            for(i=0;i<vol_level;i++) lcd_text('#');
            while(sw_down==0);
            update = 1;
        }
    }
}
```

ภาคผนวก ข รูปที่ 6 แสดงโปรแกรมส่วนควบคุมไมโครคอนโทรลเลอร์ MCS51 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้