

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์สื่อสารสำหรับผู้พิการทางหู

COMMUNICATION DEVICE FOR HEARING DISABLES



T104169



2/ค.ค.
จ 699 ค
9551

เลขหมู่.....
เลขทะเบียน 104169
วัน,เดือน,ปี 30 ต.ค. 2552

12110334

ปฏิญญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์สื่อสารสำหรับผู้พิการทางหู

COMMUNICATION DEVICE FOR HEARING DISABLES



ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2551

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์สื่อสารสำหรับผู้พิการทางหู

ผู้จัดทำ

1. นายเจตพล อัจฉริยวงศ์ 49015229
2. นายณรงค์ศักดิ์ จินดาดี 49015232
3. นายพีรพงศ์ รักเมือง 49015240


.....อาจารย์ที่ปรึกษา
(คร.กิตติพล ชิตสกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์สื่อสารสำหรับผู้พิการทางหู

นาย เจตพล อัจฉริยวงศ์ 49015229

นาย ณรงค์ศักดิ์ จินดาดี 49015232

นาย พีรพงศ์ รักเมือง 49015240

ดร.กิตติพล ชิตสกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2551

บทคัดย่อ

ปฏิญานีพนธ์นี้ได้นำเสนออุปกรณ์ช่วยสื่อสารแบบพกพาลำหรับผู้ที่มีความบกพร่องทางด้านการพูด ซึ่งจะประกอบด้วย 3 ส่วนหลักคือ การแสดงภาพสี เสียง และการอ่านข้อมูลจากหน่วยความจำ SD Card มีเมนูแสดงหมวดภาพต่างๆ มีสวิตซ์ในการควบคุม 4 ปุ่มใช้สำหรับเปลี่ยนภาพและเล่นเสียง ในส่วนของภาพกราฟิกจะมีเสียงพูดที่สื่อถึงภาพนั้นๆ การทำงานโดยใช้ไมโครคอนโทรลเลอร์ AVR เบอร์ ATmega128L เป็นตัวควบคุมจอสีแสดงผล GLCD Color6610 มีความละเอียดหน้าจอ 132 x 132 pixel และมีไอซี Epson S1D15G0 เป็นตัวควบคุม LCD และในส่วนเสียงใช้ไอซีถอดรหัสไฟล์ MP3 ของ VLSI เบอร์ VS1002D ซึ่งสามารถถอดรหัสไฟล์ MP3 และ WAVE ได้ โดยการอ่านข้อมูลภาพและเสียงที่บันทึกไว้ใน SD Card และใช้ในการสแกนคีย์สวิตซ์ โดยเครื่องต้นแบบมีภาพและข้อความทั้งสิ้น 35 ภาพ 7 กลุ่มความหมาย

COMMUNICATION DEVICE FOR HEARING DISABLES

Mr. Jatapon Atchariyawong ID. 49015229

Mr. Narongsak Jindadee ID. 49015232

Mr. Perapong Rukmeung ID. 49015240

Dr. Kitiphol Chitsakul Advisor

Education Year 2008

Abstract

This thesis presents a mobile communication device for speech disable. This device consists of 3 main parts, color display, sound player, and SD card controller. There is a menu showing the groups of image with 4 control switches to select for images or sounds. On operation, the sound player generates sounds corresponding the meaning of each graphic. By using AVR ATmega128 as controller together with GLCD Color 6610 with 132 x 132 resolutions and Epson S1D15G0 to control LCD, the graphic can display with 4096 colors. For sounds generator, MP3 files are decoded by VLSI No.VS1002D IC which can decode both MP3 and WAVE files by reading graphics and sounds recorded form SD Card and key switch scanned. There are also 35 graphics in 7 meaning groups for this model.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลงได้ด้วยดีก็ด้วยความสนับสนุนช่วยเหลือและกำลังใจจากหลาย ๆ ฝ่ายด้วยกัน โดยการให้ข้อมูล อุปกรณ์สนับสนุน และชี้แนะข้อคิดเห็นต่างๆ ซึ่งมีประโยชน์อย่างมากในการดำเนินการ ไม่ว่าจะเป็นจากพี่ ๆ เพื่อน ๆ ในภาคอิเล็กทรอนิกส์และภาคอื่นๆ ที่ให้ความช่วยเหลือทุก ๆ ด้าน พี่ ๆ ปรวิญญาโทที่ช่วยกันให้คำปรึกษาและให้กำลังใจในการทำงาน โดยเฉพาะอย่างยิ่ง ดร.กิตติพล ชิตสกุล ที่ให้คำปรึกษาคัดลอกเวลา และอาจารย์ทุก ๆ ท่านที่ให้คำปรึกษาเป็นอย่างดี

ผู้จัดทำจึงขอขอบคุณทุก ๆ ท่านมา ณ โอกาสนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1) ความเป็นมาของโครงการ	1
1.2) วัตถุประสงค์	2
1.3) ขอบเขตของโครงการ	2
1.4) วิธีการดำเนินงาน	2
1.5) ประโยชน์ที่ได้รับ	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	3
2.1) ไมโครคอนโทรลเลอร์ AVR	3
2.1.1 คุณสมบัติที่สำคัญของ AVR AT mega 128	3
2.1.2 รายละเอียดขาพอร์ต AT mega 128	4
2.2) การเชื่อมต่ออุปกรณ์อนุกรม (SPI)	11
2.2.1 โหมดการทำงานของโมดูล SPI	13
2.2.2 คุณสมบัติของขาพอร์ต SS	13
2.2.3 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด SPI	14
2.2.4 โหมดการรับส่งข้อมูล	17
2.2.5 หลักการสำคัญของการใช้ SPI	19
2.3) การใช้งานจอ GLCD Color 6610	20
2.4) การปรับจอ GLCD (LCD Display Orientation)	21
2.5) การติดต่อ GLCD 6100 ด้วย ไมโครคอนโทรลเลอร์	22

สารบัญ (ต่อ)

2.6) การจัดการหน่วยความจำฟลิกเซล	23
2.7) ข้อมูลสี 12บิต	24
2.8) การใช้งานไอซีถอดรหัสไฟล์ MP3 และ WAVE (IC VLSI VS1002xx)	27
บทที่ 3 การติดต่อกับ SD การ์ดเบื้องต้น และการ FAT	30
3.1. ความรู้เบื้องต้นเกี่ยวกับ SD การ์ด	30
3.1.1 คุณสมบัติเด่นของ SD การ์ด	30
3.1.2 คุณสมบัติทางเทคนิคที่สำคัญของ SD การ์ด	34
3.1.3 ระบบบัสที่ใช้ติดต่อกับ SD การ์ด	34
3.1.4 ขาสัญญาณของ SD การ์ด	34
3.1.5 การจัดแบ่งพื้นที่ของ SD การ์ด	36
3.1.6 รีจิสเตอร์ของ SD การ์ด	38
3.1.7 คำสั่งของ SD การ์ด	41
3.1.8 กระบวนการอ่าน-เขียน SD การ์ด	44
3.2) การติดต่อกับ SD การ์ด	46
3.2.1 รูปแบบการติดต่อกับ SD การ์ดผ่านระบบบัส SPI	49
3.2.2 การเลือกโหมด	49
3.2.3 การอ่านข้อมูลในโหมด SPI	49
3.2.4 การเขียนข้อมูลในโหมด SPI	51
3.2.5 การลบและการจัดการป้องกันการเขียน	52
3.2.6 การอ่านรีจิสเตอร์ CID และ CSD	53
3.2.7 กระบวนการรีเซต	53
3.2.8 การควบคุมสัญญาณนาฬิกา	54
3.2.9 ค่าเวลาไทม์เอาต์ของการอ่าน-เขียน-ลบข้อมูลของ SD การ์ด	54
3.2.10 คำสั่งการติดต่อ SD การ์ดผ่านระบบบัส SPI	55
3.2.11 รูปแบบของข้อมูลตอบสนอง เมื่อติดต่อ SD การ์ดแบบ SPI	56
3.2.12 รูปแบบข้อมูลในการติดต่อกับ SD การ์ดผ่านบัส SPI	60

สารบัญ (ต่อ)

3.3) FAT (File Allocation Table)	
3.3.1 ความหมายของ FAT	
3.3.2 ประเภทของ FAT	65
3.3.3 ชื่อเรียกของ FAT 32	70
บทที่ 4 การออกแบบและหลักการทํางาน	71
4.1) การออกแบบและหลักการทํางานของวงจร	73
4.2) อุปกรณ์ที่นำมาใช้งาน	73
4.3) การออกแบบวงจรเชื่อมต่อกับ GLCD 6610	74
4.4) การออกแบบวงจรถอดรหัสไฟล์ MP3 และ WAVE	75
4.5) การออกแบบวงจร SD Card	75
4.6) การออกแบบวงจรรวม	76
4.7) การออกแบบโปรแกรมเพื่อทํางานร่วมกับอุปกรณ์	77
4.7.1 จุดประสงค์ในการออกแบบ	77
4.7.2 ซอฟต์แวร์ (Software) ที่ใช้ในการออกแบบ	77
4.7.3 การออกแบบ	77
4.8) แผนผังการทํางานของโปรแกรม	80
บทที่ 5 โครงสร้างทางฮาร์ดแวร์และซอฟต์แวร์ของเครื่อง	82
5.1) การออกแบบ PCB	82
5.2) การลงอุปกรณ์	84
5.3) การทดลองและการใช้งาน	86
5.4) การแสดงภาพ	89
5.5) การแสดงเสียง	91
5.6) การแสดงภาพและเสียงที่บันทึกโดยใช้ สวิตช์ควบคุม	93
บทที่ 6 บทสรุป	94
บรรณานุกรม	
ภาคผนวก	

สารบัญรูป

รูปที่ 2.1	รายละเอียดของ ATmega128	3
รูปที่ 2.2	การต่อหน่วยความจำแรม (SRAM) ภายนอกกับ AVR ATmega128	5
รูปที่ 2.3	บล็อกไดอะแกรม SPI	11
รูปที่ 2.4	การติดต่อระหว่างมาสเตอร์กับสเลฟใน โมดูล SPI	12
รูปที่ 2.5	รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 0	18
รูปที่ 2.6	รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 1	19
รูปที่ 2.7	การใช้งานคอลัมน์และแถวแบบปกติ	21
รูปที่ 2.8	การปรับการใช้งานคอลัมน์และแถว 180 องศา	21
รูปที่ 2.9	timing diagram จอ LCD 6100 ที่ใช้ IC PCF8833 ใช้การเชื่อมต่อแบบ SPI 9-bit	22
รูปที่ 2.10	การส่ง bit DC เพื่อออก LCD Controller	22
รูปที่ 2.11	แสดงหน่วยความจำพิกเซลจะมี 132 แถวและ 132 คอลัมน์	23
รูปที่ 2.12	แสดงการแปลรหัสพิกเซล 12บิต	24
รูปที่ 2.13	แสดงวิธีการส่งพิกเซลเดียวที่ใช้การแปลรหัส 12บิต และบันทึกว่า 4บิตสีแดงสุดท้ายจาก พิกเซลต่อไปถูกกลบ	25
รูปที่ 2.14	แสดงการแปลรหัสเป็นสีแดง 3บิต สีเขียว 3บิต สีน้ำเงิน 2บิต	25
รูปที่ 2.15	แสดงการแปลรหัสเป็น 5บิตสำหรับสีแดง 6บิต สำหรับสีเขียว และ 5บิตสำหรับสีน้ำเงิน	26
รูปที่ 2.16	ไอซี VS1002D	27
รูปที่ 2.17	โครงสร้างไอซี VS1002D	28
รูปที่ 3.1	แสดงไดอะแกรมการทำงานเบื้องต้นของ SD การ์ด	31
รูปที่ 3.2	แสดงลักษณะทางกายภาพของ SD การ์ด	32
รูปที่ 3.3	กระบวนการป้องกันการละเมิดลิขสิทธิ์ใน SD การ์ด	33
รูปที่ 3.4	ไดอะแกรมแสดงการจัดแบ่งพื้นที่ของ SD การ์ด	37
รูปที่ 3.5	แสดงกระบวนการอ่าน-เขียนข้อมูลของ SD การ์ด	45
รูปที่ 3.6	วงจรการเชื่อมต่อเบื้องต้นระหว่าง ไมโครคอนโทรลเลอร์ กับ SD การ์ดผ่านระบบบัส SD	47

สารบัญรูป (ต่อ)

รูปที่ 3.7 แสดงไดอะแกรมการติดต่อกับ SD การ์ดผ่านบัส SPI	48
รูปที่ 3.8 วงจรการเชื่อมต่อเบื้องต้นระหว่างไมโครคอนโทรลเลอร์ กับ SD การ์ดผ่านระบบบัส SPI	48
รูปที่ 3.9 กระบวนการอ่านข้อมูลแบบบล็อกเดียวจาก SD การ์ด	50
รูปที่ 3.10 กระบวนการอ่านข้อมูลแบบหลายบล็อกจาก SD การ์ด	50
รูปที่ 3.11 จังหวะการเขียนข้อมูลลงใน SD การ์ดแบบบล็อกเดียว	51
รูปที่ 3.12 จังหวะการเขียนข้อมูลลงใน SD การ์ดแบบหลายบล็อก	52
รูปที่ 3.13 แสดงจังหวะการทำงานของทั้งคำสั่งลบและป้องกันการเขียน ในทางปฏิบัติไม่ได้ใช้ร่วมหรือต่อกัน เพียงนำมาอธิบายต่อเนื่องกันเท่านั้น	53
รูปที่ 3.14 แสดงส่วนประกอบของฮาร์ดดิสก์	67

สารบัญตาราง

ตารางที่ 2.1 การกำหนดรูปแบบของชาพอร์ด	10
ตารางที่ 2.2 การกำหนดขอบขาสัญญาณ SCK	15
ตารางที่ 2.3 การกำหนดการอ่านตัวอย่างข้อมูล	15
ตารางที่ 2.4 การกำหนดสัญญาณความถี่	16
ตารางที่ 2.5 การกำหนดโหมดสัญญาณนาฬิกาในการรับส่งข้อมูล	17
ตารางที่ 3.1 สรุปรูปข้อมูลสำคัญของการติดต่อกับ SD การ์ดทั้งแบบบัส SD และ SPI	35
ตารางที่ 3.2 เป็นการจ้ดขาคเมื่อติดต่อกับ SD การ์ดด้วยบัส SD	35
ตารางที่ 3.3 เป็นการจ้ดขาคเมื่อติดต่อกับ SD การ์ดด้วยบัส SPI	36
ตารางที่ 3.4 แสดงความสัมพันธ์ของบิตข้อมูลในรีจิสเตอร์ OCR กับแรงดันของ SD การ์ด	38
ตารางที่ 3.5 รายละเอียดของรีจิสเตอร์ CID	39
ตารางที่ 3.6 รูปแบบข้อมูลทั้ง 48 บิตของคำสั่งของ SD การ์ด	41
ตารางที่ 3.7 แสดงสายสัญญาณของการติดต่อกับ SD การ์ดทั้งแบบผ่านบัส SD และ SPI	45
ตารางที่ 3.8 สรุปลำสั่งที่ใช้ในการติดต่อกับ SD การ์ด เมื่อทำงานผ่านระบบบัส SPI โดยแบ่งตามคลาส	55
ตารางที่ 3.9 แสดงตัวอย่างระบบปฏิบัติการกับระบบ ไฟล์ที่ใช้	66
ตารางที่ 3.10 แสดงการเปรียบเทียบขนาดระหว่างคลัสเตอร์ FAT32 และ FAT16	69

บทที่ 1

บทนำ

1.1) ความเป็นมาของโครงการ

สังคมปัจจุบันเป็นสังคมข่าวสารและสารสนเทศ ทำให้โลกไร้พรมแดน ประชาชนบนพื้นโลกสามารถติดต่อสื่อสารกันได้อย่างรวดเร็ว ดังจะเห็นได้จากข่าวต่างๆที่เกิดขึ้นที่มุมโลกหนึ่งสามารถสื่อสารไปยังอีกมุมโลกหนึ่งได้อย่างรวดเร็วเกือบจะทันที โดยผ่านเทคโนโลยีด้านภาพเสียง เครื่องมือในการติดต่อสื่อสารของมนุษย์ที่สำคัญ คือ ภาษาและเพื่อการพัฒนาด้านการสื่อสารที่มีประสิทธิภาพ ประชากรจึงต้องมีภาษาเป็นเครื่องมือในการถ่ายทอดความคิดจากบุคคลหนึ่งไปยังบุคคลหนึ่ง เพื่อให้เกิดความเข้าใจตรงกันและเพื่อความสำเร็จของคน เพราะในการบริหารงานนั้นการสื่อความหมายกันในองค์กรนับเป็นเรื่องสำคัญอย่างยิ่ง

ภาษาพูดเป็นเครื่องมือการถ่ายทอดข้อมูลข่าวสารพื้นฐานที่ตรงไปตรงมาและให้รายละเอียดได้มากที่สุด จากการซักถามและตอบโต้ แต่กลับเป็นอุปสรรคสำคัญในกรณีฝ่ายหนึ่งเป็นผู้พิการทางการได้ยินและอีกฝ่ายหนึ่งเป็นบุคคลปกติ ดังนั้นปัญหาสำคัญของผู้พิการที่พูดไม่ได้คือความสามารถในการสื่อสารกับคนปกติ ทำให้กลุ่มคนพิการดังกล่าวขาดโอกาสในการเข้าถึงข้อมูลและข่าวสารจากบุคคลอื่นๆในสังคม จึงทำให้เป็นจุดกำเนิดในการพัฒนาโปรแกรมช่วยเหลือเพื่อคนพิการสำหรับการสื่อสารกับคนปกติ

โครงการชิ้นนี้เป็นการสร้างอุปกรณ์สื่อสารอย่างง่ายสำหรับผู้พิการทางหูซึ่งสามารถพกพาไปที่ต่างๆได้ เป็นการพัฒนาต่อยอดจากเครื่อง “โอภา” ของศูนย์อิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติให้มีขนาดกระทัดรัดและราคาประหยัด โดยโครงการประกอบสองส่วนคือในส่วนของภาพและในส่วนของเสียง โครงการประกอบด้วยจอแสดงภาพโนเกียรุ่น 6610 และวงจรถอดรหัสไฟล์ MP3 และ WAVE และเล่นเสียงกลับนี้ต่ออินเทอร์เฟซกับไมโครคอนโทรลเลอร์ มีการเลือกปุ่มการทำงานผ่านทางสวิตช์ 4 ปุ่มเพื่อให้ผู้พิการที่ไม่สามารถใช้นิ้วและมือได้สะดวกสามารถใช้เครื่องนี้ได้

1.2) วัตถุประสงค์

1. ศึกษาการใช้จอสี GLCD Nokia color 6610 นำไปใช้ต่อเข้ากับ MCU ชนิดต่าง ๆ นำไปแสดงผลตามต้องการ โดยเปลี่ยนจอคริสตัลเหลวขาวดำ เป็นจอสี
2. ศึกษาการเขียนโปรแกรมประยุกต์สำหรับงานด้านการแสดงผลของกราฟิกและการแสดงเสียงทางลำโพง โดยอ่านข้อมูลจาก SD card
3. ศึกษาการใช้ระบบอิเล็กทรอนิกส์กินกำลังงานต่ำ
4. สร้างต้นแบบเครื่องช่วยสื่อสารด้วยภาพและเสียงอย่างง่าย

1.3) ขอบเขตของโครงการ

สร้างเครื่องต้นแบบอุปกรณ์ช่วยสื่อสาร แบบพกพาสำหรับผู้ที่มีบกพร่องทางการพูดบนโปรแกรมที่สร้างได้และวงจรที่ช่วยจำกัดแรงดันเพื่อประหยัดพลังงานในการใช้งาน ในส่วนของโปรแกรมประยุกต์ที่แสดงผลจะสามารถแสดงภาพกราฟิกต่างๆที่จำเป็นในชีวิตประจำวันพร้อมกับแสดงเสียงที่สอดคล้องกับข้อความภาพนั้นๆ

1.4) วิธีการดำเนินงาน

1. ศึกษาการอินเตอร์เฟสระหว่าง MCU กับ GLCD Nokia color 6610 และระหว่าง MCU กับไอซีถอดรหัสไฟล์ MP3 หรือ WAVE เพื่อแสดงภาพกราฟิกและข้อมูลเสียงที่บันทึกภายใน SD card
2. ศึกษาการเข้าถึงข้อมูลภายใน SD card เพื่ออ่านไฟล์ภาพและเสียงที่บันทึกไว้
3. เขียนโปรแกรมประยุกต์ที่ใช้ในการแสดงภาพกราฟิกบน GLCD Nokia color 6610 พร้อมทั้ง สวิตช์สำหรับเลือกการแสดงผลภาพและแสดงเสียงที่ต้องการจาก SD card

1.5) ประโยชน์ที่ได้รับ

1. เครื่องมีขนาดเล็กสามารถพกพาสะดวก
2. สร้างความสะดวกสบายแก่ผู้พิการ ในการติดต่อสื่อสารในชีวิตประจำวันและการเรียนการสอนได้
3. นำของทั้งหมดค้มาประยุกต์ใช้ให้เกิดประโยชน์มากขึ้น

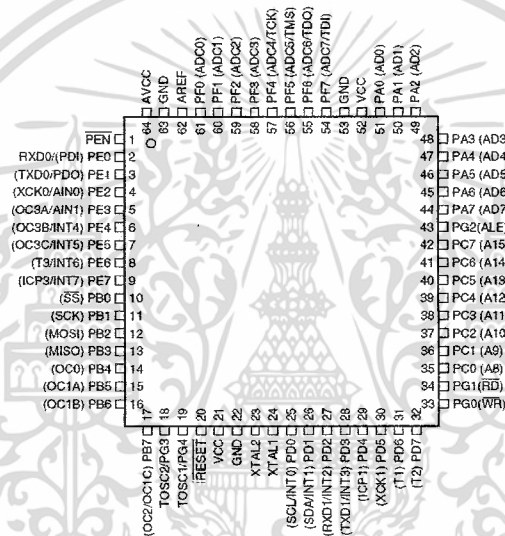
บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1) ไมโครคอนโทรลเลอร์ AVR

อินพุต/เอาต์พุต AVR ATmega128

ATmega128 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตตระกูล AVR สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer) เช่นเดียวกับ AT mega 16 แต่มีคุณสมบัติที่มากกว่าในตัวถังแบบ Thin Profile Plastic Quad Flat Package (TQFP) และ Micro Lead Frame Package (MLF) การจัดตำแหน่งขาพอร์ต์



รูปที่ 2.1 รายละเอียดขา ATmega128

2.1.1 คุณสมบัติที่สำคัญของ AVR AT mega 128

- ไมโครคอนโทรลเลอร์ขนาด 8 บิต กำล้างไฟต่ำ ประสิทธิภาพสูง
- สถาปัตยกรรมแบบ Advanced RISC (Reduce Instruction Set Computer)
 - 133 คำสั่ง มีความเร็วในการประมวลผล 1 คำสั่งต่อ 1 สัญญาณนาฬิกา
 - รีจิสเตอร์ใช้งาน 32 ตัวขนาด 8 บิต และรีจิสเตอร์ควบคุม
 - ความเร็วในการทำงานสูงถึง 16 MIPS ที่ความถี่สัญญาณนาฬิกา 16 MHz
- หน่วยความจำโปรแกรมแบบ Nonvolatile และหน่วยความจำข้อมูล
 - หน่วยความจำโปรแกรมแบบ FLASH ขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง
 - หน่วยความจำข้อมูล EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูล SRAM ขนาด 4 กิโลไบต์
- สามารถขยายหน่วยความจำได้มากถึง 64 กิโลไบต์
- มีวงจรถักอินเทอร์เฟซแบบ JTAG
- โมดูลใช้งานประกอบไปด้วย
 - โมดูลไทมเมอร์/เคาน์เตอร์ ขนาด 8 บิต 2 ตัวพร้อมปริสเกลเลอร์และโหมดเปรียบเทียบ
 - กำหนดให้ทำงานแบบ 16 บิตได้ ด้วยการ ใช้ไทมเมอร์ 2 ตัว ทำงานร่วมกัน พร้อมปริสเกลเลอร์ โหมดเปรียบเทียบและโหมดตรวจจับสัญญาณอินพุต
 - ตัวนับแบบเรียบใหม่กับออสซิลเลเตอร์แยกเฉพาะ
 - โมดูลสร้างสัญญาณ PWM 8 ช่อง
 - PWM 6 ช่องกำหนดความละเอียดได้ตั้งแต่ 2 ถึง 16 บิต
 - โมดูลเชื่อมต่อแบบอนุกรม TWI (Two-wire Serial), SPI และ USART
 - โมดูลวอตด็อกซ์ไทมเมอร์ (Watchdog Timer)
 - โมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 10 บิต
 - โมดูลเปรียบเทียบแรงดันอะนาลอก
- คุณสมบัติพื้นฐานสำหรับไมโครคอนโทรลเลอร์
 - เพาเวอร์ออนรีเซตและเบราร์วเอด์ดีเทคแบบโปรแกรมได้
 - RC ออสซิลเลเตอร์ภายในเพื่อสร้างสัญญาณความถี่
 - แหล่งอินเทอร์รัปต์ทั้งภายในและภายนอก
 - 6 โหมดสลีปประกอบด้วย โหมด Idle, ADC Noise Reduction, Power-save, Power-down, Standby และ Extended Standby
 - กำหนดโหมดสัญญาณความถี่นาฬิกาด้วยซอฟต์แวร์
- ขาพอร์ตอินพุต/เอาต์พุต 53 ขาประกอบด้วยพอร์ต A, B, C, D, E, F, G

2.1.2 รายละเอียดขาพอร์ต AT mega 128

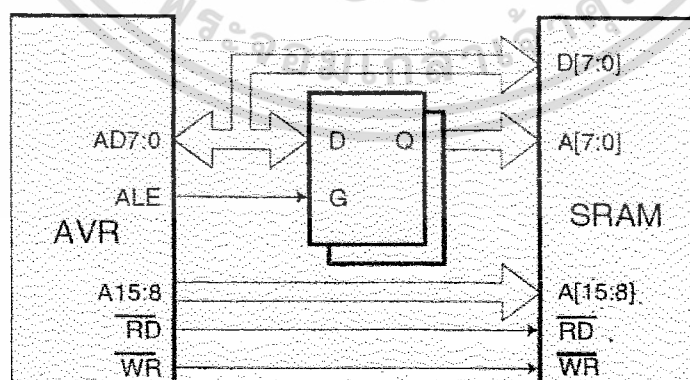
ไมโครคอนโทรลเลอร์ AVR AT mega 128 มีจำนวนขาพอร์ตอินพุตเอาต์พุตดิจิตอลทั้งสิ้น 53 ขาพอร์ตประกอบไปด้วยพอร์ต A ถึง F พอร์ตละ 8 บิต และพอร์ต G จำนวน 5 บิต โดยในแต่ละขาพอร์ตสามารถทำหน้าที่ได้มากกว่าหนึ่งหน้าที่ โดยหน้าที่พิเศษเพิ่มเติมจะอ้างอิงกับคุณสมบัติของโมดูลต่าง ๆ มีรายละเอียดดังนี้

ขา PORTA (PA0-PA7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PA7	AD7 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 7)
PA6	AD6 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 6)
PA5	AD5 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 5)
PA4	AD4 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 4)
PA3	AD3 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 3)
PA2	AD2 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 2)
PA1	AD1 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 1)
PA0	AD0 (ขาเชื่อมต่อหน่วยความจำภายนอก แอคเดรสและค่าข้อมูล บิตที่ 0)

ขา PORTA นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 8 บิตแล้ว ยังมีคุณสมบัติที่สำคัญดังนี้

- ใช้เป็นขาพอร์ตเพื่อต่อขยายหน่วยความจำภายนอกเพิ่มเติม โดยเป็นทั้งขาแอกเดรสและค่าข้อมูลขนาด 8 บิต รูปแบบการต่อใช้งานหน่วยความจำภายนอกเพิ่มเติม แสดงดังรูป ซึ่งเป็นการต่อขยายหน่วยความจำประเภท SRAM เพิ่มเติม โดยการต่อขยายหน่วยความจำจะใช้ขา PORTA เป็นทั้งขาแอกเดรสและค่าข้อมูล 8 บิตล่าง (บิตที่ 0 ถึงบิตที่ 7) และใช้ขา PORTA เป็นขาแอกเดรส 8 บิตบน (บิตที่ 8 ถึงบิตที่ 15)



รูปที่ 2.2 การต่อหน่วยความจำแรม (SRAM) ภายนอกกับ AVR ATmega128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา PORTB (PB0-PB7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PB7	OC2/OC1C (โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM สำหรับไทเมอร์/เคาน์เตอร์ 2 หรือ โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง C สำหรับไทเมอร์/เคาน์เตอร์ 1)
PB6	OC1B (โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง B สำหรับไทเมอร์/เคาน์เตอร์ 1)
PB5	OC1A (โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง A สำหรับไทเมอร์/เคาน์เตอร์ 1)
PB4	OC0 (โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM สำหรับไทเมอร์/เคาน์เตอร์ 0)
PB3	MISO (ขาสัญญาณ อินพุตมาสเตอร์/เอาต์พุตสเลฟ สำหรับบัส SPI)
PB2	MOSI (ขาสัญญาณ อินพุตมาสเตอร์/เอาต์พุตสเลฟ สำหรับบัส SPI)
PB1	SCK (ขาสัญญาณนาฬิกาสำหรับบัส SPI)
PB0	SS (ขาสัญญาณอินพุตเลือกสเลฟ สำหรับบัส SPI)

ขา PORTB นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 8 บิตแล้ว ยังมีคุณสมบัติที่สำคัญดังนี้

- ใช้เป็นขาพอร์ตสำหรับสร้างสัญญาณ Pulse Width Modulator (PWM) 3 ช่องทางคือ A, B และ C ซึ่งนำไปประยุกต์ใช้ในการควบคุมการทำงานของมอเตอร์ และใช้เป็นขาพอร์ตสำหรับเชื่อมต่อกับอุปกรณ์ภายนอกแบบอนุกรม (Serial Peripheral Interface) หรือที่นิยมเรียกว่าบัสแบบ SPI

ขา PORTC (PC0-PC7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PC7	A15 (ขาสัญญาณแอกเดรส บิตที่ 15)
PC6	A14 (ขาสัญญาณแอกเดรส บิตที่ 14)
PC5	A13 (ขาสัญญาณแอกเดรส บิตที่ 13)
PC4	A12 (ขาสัญญาณแอกเดรส บิตที่ 12)
PC3	A11 (ขาสัญญาณแอกเดรส บิตที่ 11)
PC2	A10 (ขาสัญญาณแอกเดรส บิตที่ 10)
PC1	A9 (ขาสัญญาณแอกเดรส บิตที่ 9)
PC0	A8 (ขาสัญญาณแอกเดรส บิตที่ 8)

ขา PORTC นอกจากจะเป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลแล้ว ยังมีคุณสมบัติสำคัญดังนี้

- ใช้งานเป็นขาพอร์ตสำหรับเชื่อมต่อหน่วยความจำภายนอก โดยเป็นขาแอกเดรส 8 บิตบน (บิตที่ 8 ถึงบิตที่ 15) ทำงานร่วมกับขา PORTA ดังที่กล่าวมาแล้วข้างต้น

ขา PORTD (PD0-PD7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PD7	T2 (อินพุตสัญญาณ สำหรับ ไทเมอร์/คาน์เตอร์ 2)
PD6	T1 (อินพุตสัญญาณ สำหรับ ไทเมอร์/คาน์เตอร์ 1)
PD5	XCK1 (ขาอินพุตเอาต์พุตสัญญาณนาฬิกาภายนอกสำหรับ โมดูล USART1)
PD4	ICP1 (ขาอินพุตตรวจจับสัญญาณของ ไทเมอร์/คาน์เตอร์ 1)
PD3	INT3/TXD1 (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 3 หรือขาส่งข้อมูลโมดูล UART1)
PD2	INT2/RXD1 (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 2 หรือขา รับข้อมูลโมดูล UART1)
PD1	INT1/SDA (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 1 หรือขาข้อมูลอนุกรมสำหรับ โมดูล TW1)
PD0	INT0/SCL (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 0 หรือ ขาสัญญาณนาฬิกาสำหรับ โมดูล TW1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา PORTD นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 8 บิตแล้ว ยังมีคุณสมบัติที่สำคัญดังนี้

- ขาพอร์ตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 0 ถึง 3
- ขาพอร์ตสำหรับอินพุตสัญญาณนาฬิกาจากภายนอกของไทเมอร์/เคาน์เตอร์ 1 และ 2 หรือที่เรียกว่าการทำงานในโหมดเคาน์เตอร์
- ขาตรวจจับสัญญาณอินพุต (Input Capture) ของไทเมอร์/เคาน์เตอร์ 1 เมื่อตรวจจับสัญญาณได้จะใช้เวลาในการจับสัญญาณมา สามารถนำมาประยุกต์ใช้ในการจับสัญญาณความถี่หรือหาความกว้างของสัญญาณ PWM ได้ เป็นต้น
- ขาพอร์ตสำหรับ โมดูลอนุกรม USART1 ช่องที่ 1 เพื่อใช้ในการสื่อสารข้อมูลอนุกรมกับอุปกรณ์ภายนอก แบบ RS-232 ช่องที่ 1
- ขาพอร์ตสำหรับการเชื่อมต่อด้วยสายสัญญาณ 2 เส้น (Two-wire Serial Interface (TWI)) หรือที่รู้จักกันในชื่อ I²C บัส

ขา PORTE (PE0-PE7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PE7	INT7/ICP3 (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 7 หรือขาอินพุตตรวจจับสัญญาณของไทเมอร์/เคาน์เตอร์ 3)
PE6	INT6/T3 (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 6 หรืออินพุตรับสัญญาณสำหรับไทเมอร์/เคาน์เตอร์ 3)
PE5	INT5/OC3C (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 5 หรือ โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง C สำหรับไทเมอร์/เคาน์เตอร์ 3)
PE4	INT4/OC3B (ขาอินพุตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 4 หรือ โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง B สำหรับไทเมอร์/เคาน์เตอร์ 3)
PE3	AIN1/OC3A (อินพุตเปรียบเทียบแรงดันอะนาลอกด้านลบหรือ โมดูลเปรียบเทียบและสร้างสัญญาณเอาต์พุต PWM ช่อง A สำหรับไทเมอร์/เคาน์เตอร์ 3)
PE2	AIN0/XCK0 (อินพุตเปรียบเทียบแรงดันอะนาลอกด้านบวกหรืออินพุตเอาต์พุตสัญญาณนาฬิกาจากภายนอกสำหรับ โมดูล USART0)
PE1	PDO/TXD0 (ขาโปรแกรมข้อมูลเอาต์พุต หรือขาส่งข้อมูล โมดูล UART0)
PE0	PDI/RXD0 (ขาโปรแกรมข้อมูลอินพุต หรือขารับข้อมูล โมดูล UART0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา PORTE นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 8 บิตแล้ว ยังมีคุณสมบัติที่สำคัญดังนี้

- ขาพอร์ตอินเทอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 4 ถึง 7
- ขาตรวจจับสัญญาณอินพุต (Input Capture) ของ ไทเมอร์/เคาน์เตอร์ 3
- ขาพอร์ตสร้างสัญญาณ PWM สำหรับ ไทเมอร์/เคาน์เตอร์ 3
- ขาอินพุตด้านบวกและลบสำหรับ โมดูลเปรียบเทียบแรงดันอะนาล็อก
- ขาพอร์ตสำหรับ โมดูลอนุกรม USART0 ช่องที่ 0

ขา PORTF (PF0-PF7)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PF7	ADC7/TDI (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 7 หรือขาทดสอบ JTAG สำหรับอินพุตข้อมูล)
PF6	ADC6/TDO (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 6 หรือขาทดสอบ JTAG สำหรับเอาต์พุตข้อมูล)
PF5	ADC5/TMS (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 5 หรือขาทดสอบ JTAG สำหรับเลือกโหมด)
PF4	ADC4/TCK (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 4 หรือขาทดสอบ JTAG สำหรับสัญญาณนาฬิกา)
PF3	ADC3 (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 3)
PF2	ADC2 (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 2)
PF1	ADC1 (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 1)
PF0	ADC0 (ขาอินพุตสัญญาณอะนาล็อกช่องที่ 0)

ขา PORTF นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 8 บิตแล้ว ยังมีคุณสมบัติที่สำคัญดังนี้

- ขาพอร์ตสำหรับวงจร JTAG สำหรับการดีบั๊ก โปรแกรม
- ขาพอร์ตสำหรับ โมดูลแปลงสัญญาณอะนาล็อกเป็นดิจิทัลขนาด 10 บิต 8 ช่อง

ขา PORTG (PG0-PG4)

ขาพอร์ต	ขาพอร์ตฟังก์ชันพิเศษ
PG4	TOSC1 (ขา RTC ออสซิลเลเตอร์ 1 สำหรับไทมเมอร์/เคาน์เตอร์ 0)
PG3	TOSC2 (ขา RTC ออสซิลเลเตอร์ 2 สำหรับไทมเมอร์/เคาน์เตอร์ 0)
PG2	ALE (ขาแลตซ์แอดเดรสสำหรับเอ็นเอเบิลหน่วยความจำภายนอก)
PG1	RD (ขาสโตร์ สัญญาณอ่านข้อมูลหน่วยความจำภายนอก)
PG0	WD (ขาสโตร์ สัญญาณเขียนข้อมูลหน่วยความจำภายนอก)

ขา PORTG นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตดิจิทัลขนาด 5 บิตแล้ว ยังมีคุณสมบัติสำคัญ ดังนี้

- ขาพอร์ตออสซิลเลเตอร์สำหรับ โมดูลสัญญาณนาฬิกาเวลาจริง Real-Time Clock (RTC)
- ขาพอร์ตสำหรับอ่าน-เขียนหน่วยความจำภายนอก ทำงานร่วมกับขา PORTA และ PORTC

ขาพอร์ตอินพุตเอาต์พุตดิจิทัลของ AT mega128 มีคุณสมบัติแบบเดียวกับ AT mega16 โดยกำหนดคุณสมบัติของขาพอร์ตผ่านทางรีจิสเตอร์ DDRx และ PORTx รวมถึงบิต PUD ในรีจิสเตอร์ SFIOR แสดงดังตาราง

ตารางที่ 2.1 การกำหนดรูปแบบของขาพอร์ต

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-Up	สถานะ
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

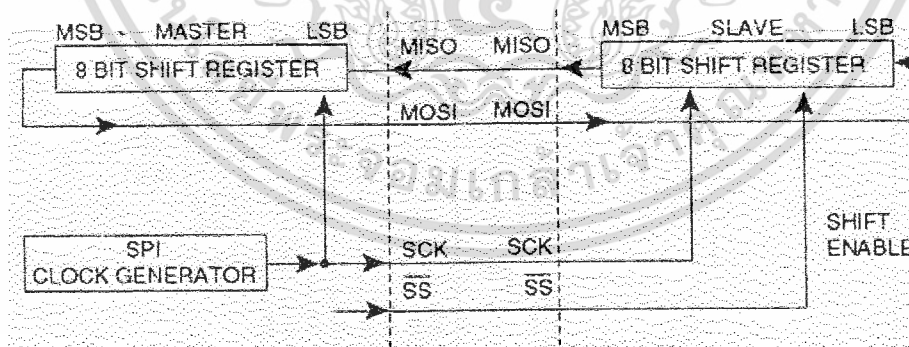
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปด้านบน การสื่อสารข้อมูล SPI ประกอบไปด้วยสายสัญญาณ 4 เส้น ดังนี้

- MOSI (Master Out - Slave In)
ใช้ในการส่งข้อมูลออกจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ
- MISO (Master In – Slave Out)
ใช้ในการรับข้อมูลจากอุปกรณ์สเลฟหรือมาสเตอร์อ่านข้อมูล
- SCK (Serial Clock)
สายสัญญาณนาฬิกา ที่ใช้งานร่วมกันระหว่างมาสเตอร์กับสเลฟ
- SS (Slave Select)
สายสัญญาณสำหรับเลือกอุปกรณ์สเลฟ

อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์มาสเตอร์จะต้องควบคุมอุปกรณ์สเลฟได้ อุปกรณ์สเลฟมักจะเป็น ไอซีหน้าที่พิเศษต่างๆ เช่น ไอซีวัดอุณหภูมิ ไอซีฐานเวลานาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำงานในโหมดสเลฟ ก็ได้เช่นเดียวกัน

การใช้งาน โมดูล SPI โดยปกติแล้วจะใช้สายสัญญาณทั้ง 4 เส้นที่กล่าวมา หากเป็นการต่อระหว่างไมโครคอนโทรลเลอร์ด้วยกัน โดยที่ไมโครคอนโทรลเลอร์ตัวหนึ่งทำหน้าที่เป็นมาสเตอร์ และอีกตัวทำหน้าที่สเลฟ จะใช้สายสัญญาณทั้ง 4 เส้น ดังรูปด้านล่างนี้



รูปที่ 2.4 การติดต่อระหว่างมาสเตอร์กับสเลฟในโมดูล SPI

สำหรับการเชื่อมต่อกับอุปกรณ์สเลฟที่ไม่ได้เป็นไมโครคอนโทรลเลอร์ จะมีขาสำหรับใช้ในการกำหนดการทำงานของอุปกรณ์สเลฟเรียกว่า สายสัญญาณ EN (ENABLE) เพิ่มเติมขึ้นมาและสายสัญญาณ SS จะไม่มีการใช้งาน ซึ่งจะเห็นได้ในตัวอย่างแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 โหมดการทำงานของโมดูล SPI

- **มาสเตอร์โหมด (Master Mode)**

เมื่อกำหนดให้ทำงานในโหมดมาสเตอร์ การควบคุมขาพอร์ต SS จะต้องควบคุมด้วยกระบวนการทางซอฟต์แวร์ (เขียนโปรแกรมควบคุมเอง) ก่อนที่จะเริ่มต้นการสื่อสารข้อมูล SPI การส่งข้อมูลจะเริ่มต้นโดยการเขียนข้อมูลไปที่รีจิสเตอร์ SPDR (SPI Data Register) สัญญาณนาฬิกาจะกำหนดการส่งข้อมูลโดยอัตรานิมิต เมื่อส่งข้อมูลเสร็จสมบูรณ์แฟล็ก SPIF จะถูกเซต หากมีการเปิดใช้งานอินเตอร์รัปต์โดยการเซตบิต SPIE (SPI Interrupt Enable bit) และเซตบิตอินเตอร์รัปต์โกลบอล (บิต I (Global Interrupt Enable) ในรีจิสเตอร์ SREG) จะเกิดอินเตอร์รัปต์ที่เกี่ยวกับโมดูล SPI ขึ้น

- **สเลฟโหมด (Slave Mode)**

เมื่อกำหนดให้ทำงานในโหมดสเลฟ ขาพอร์ต MISO จะเป็น tri-stated (ลอจิก 3 สถานะ คือ ลอจิก “0”, ลอจิก “1” หรือสภาพที่ไม่เป็นทั้งลอจิก “0” และ “1” ซึ่งเรียกสภาวะนี้ว่า อิมพีแดนซ์สูง (Hi-Z)) จนขาพอร์ต SS เป็นลอจิก “1” (High) ในสภาวะนี้ กระบวนการทางซอฟต์แวร์จะมีการปรับปรุงข้อมูลในรีจิสเตอร์ SPDR และมีการชิฟต์ (Shift) ข้อมูลเกิดขึ้น จนสัญญาณ SS เป็นลอจิก “0” (Low) เมื่อรับส่งข้อมูลเสร็จสมบูรณ์ บิต SPIF จะถูกเซต หากมีการเปิดใช้งานอินเตอร์รัปต์ ก็จะเกิดอินเตอร์รัปต์ขึ้น

2.2.2 คุณสมบัติของขาพอร์ต SS

เมื่อ SPI ทำงานในโหมดสเลฟ ขาพอร์ต SS (Slave Select) จะต้องกำหนดเป็นอินพุต หากทำงานในโหมดมาสเตอร์ ผู้ใช้งานสามารถกำหนดได้ว่าต้องการให้ขาพอร์ต SS เป็นอินพุตหรือเอาต์พุต หากกำหนดเป็นเอาต์พุตขาพอร์ต SS จะไม่มีผลกับโหมด SPI แต่หากกำหนดเป็นอินพุต จะต้องกำหนดให้ขาพอร์ตมีสถานะเป็นลอจิก “1” หรือ High เพื่อให้ทำงานในโหมดมาสเตอร์ โดยการทำงานของขาพอร์ตในแต่ละโหมดแสดงดังตารางด้านล่าง

ขาพอร์ต	ทิศทางขาพอร์ตโหมดมาสเตอร์	ทิศทางขาพอร์ตโหมดสเลฟ	ขาพอร์ต ATmega16
MOSI	ผู้ใช้กำหนด	อินพุต	PB5
MISO	อินพุต	ผู้ใช้กำหนด	PB6
SCK	ผู้ใช้กำหนด	อินพุต	PB7
SS	ผู้ใช้กำหนด	อินพุต	PB4

2.2.3 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด SPI

รีจิสเตอร์ SPCR (SPI Control Register)

รีจิสเตอร์ควบคุมการทำงานของโมดูล SPI เพื่อกำหนดโหมดการทำงาน การใช้งานอินเตอร์รัปต์รวมถึงการกำหนดความถี่สัญญาณนาฬิกาในการรับส่งข้อมูล

รีจิสเตอร์ SPSR (SPI Status Register)

รีจิสเตอร์แสดงสถานะการทำงานของโมดูล SPI เช่น การเซตแฟล็ก SPIF การตรวจสอบการชนกันของข้อมูล (Collision Flag) เป็นต้น

รีจิสเตอร์ SPDR (SPI Data Register)

รีจิสเตอร์ใช้ในการรับส่งข้อมูลของโมดูล SPI ข้อมูลการรับและส่งจะถูกกำหนดในรีจิสเตอร์นี้ โดยมีรายละเอียดของรีจิสเตอร์ดังต่อไปนี้

รีจิสเตอร์ SPCR (SPI Control Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7 : บิต SPIE (SPI Interrupt Enable)
บิตเปิดการใช้งานอินเตอร์รัปต์ SPI เมื่อบิต SPIF ถูกเซตจะเกิดอินเตอร์รัปต์ขึ้น หากมีการเปิดการใช้งานอินเตอร์รัปต์โดยรวม (บิต I ในรีจิสเตอร์ SREG)
- บิตที่ 6 : บิต SPE (SPI Enable)
เซตบิต SPE เป็น “1” เพื่อเปิดการใช้งานโมดูล SPI
- บิตที่ 5 : บิต DORD (Data Order)
บิตกำหนดการส่งข้อมูล LSB หรือ MSB ก่อนเป็นลำดับแรก เซตบิต DORD เป็น “1” บิต LSB ของข้อมูลจะถูกส่งออกไปก่อน เซตบิต DORD เป็น “0” บิต MSB ของข้อมูลจะถูกส่งออกไปก่อน
- บิตที่ 4 : บิต MSTR (Master/Slave Select)
บิตกำหนดโหมดการทำงานเป็นมาสเตอร์หรือสเลฟ เซตบิต MSTR เป็น “1” เมื่อต้องการทำงานในโหมดมาสเตอร์ และเซตบิต MSTR เป็น “0” เมื่อต้องการทำงานในโหมดสเลฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิตที่ 3 : บิต OCF3B (Timer/Counter3, Output Compare B Match Flag)
บิตกำหนดลักษณะของขอบขาสัญญาณนาฬิกาเมื่ออยู่ในสถานะไอดีล (Idle) เมื่อเซตเป็น “1” ขาสัญญาณ SCK จะเป็น Low เมื่ออยู่ในโหมดไอดีล เซตเป็น “0” ขาสัญญาณ SCK จะเป็น High เมื่ออยู่ใน โหมดไอดีล ตารางด้านล่าง

ตารางที่ 2.2 การกำหนดขอบขาสัญญาณ SCK

CPOL	ต้นของสัญญาณ 1 ลูก	ปลายของสัญญาณ 1 ลูก
0	ขอบขาขึ้น	ขอบขาลง
1	ขอบขาลง	ขอบขาขึ้น

- บิตที่ 2 : บิต CPHA (Clock Phase)
เซตบิต CPHA เพื่อกำหนดรูปร่างสัญญาณนาฬิกาหรือเฟสของสัญญาณนาฬิกาในการอ่านตัวอย่างข้อมูล (Sample) จากการเปลี่ยนแปลงสัญญาณนาฬิกาจาก High เป็น Low หรือจาก Low เป็น High ใดๆอย่างหนึ่ง การกำหนดบิต CPHA แสดงดังตารางด้านล่างนี้

ตารางที่ 2.3 การกำหนดการอ่านตัวอย่างข้อมูล

CPHA	ต้นของสัญญาณ 1 ลูก	ปลายของสัญญาณ 1 ลูก
0	อ่านข้อมูล	เปลี่ยนแปลงข้อมูล
1	เปลี่ยนแปลงข้อมูล	อ่านข้อมูล

- บิตที่ 1:0 : บิต SPR1, SPR0 (SPI Clock Rate Select 1 and 0)
บิตกำหนดความถี่สัญญาณนาฬิกา (f_{osc}) ในการรับส่งข้อมูลระหว่างอุปกรณ์ในโหมด SPI แสดงดังตาราง

ตารางที่ 2.4 การกำหนดสัญญาณความถี่

SPI2X	SPR1	SPR0	ความถี่ SCK
0	0	0	fosc/4
0	0	1	fosc/16
0	1	0	fosc/64
0	1	1	fosc/128
1	0	0	fosc/2
1	0	1	fosc/8
1	1	0	fosc/32
1	1	1	fosc/64

รีจิสเตอร์ SPSR (SPI Status Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	SPIF	WCOL	-	-	-	-	-	SPI2X
Read/Write	R	R	R	R	R	R	R	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7 : บิต SPIF (SPI Interrupt Flag)
บิต SPIF จะถูกเซตเมื่อการรับส่งข้อมูล SPI เสร็จสมบูรณ์ หากมีการกำหนดใช้งานอินเทอร์รัปต์ SPI จะเกิดอินเทอร์รัปต์ขึ้น
- บิตที่ 6 : บิต WCOL (Write COLLision Flag)
บิต WCOL จะถูกเซต เมื่อตรวจพบว่าการเขียนข้อมูลในรีจิสเตอร์ SPDR ระหว่างการส่งข้อมูล จะถูกเคลียร์เมื่อมีการอ่านรีจิสเตอร์ SPSR บิต WCOL และการเข้าถึงรีจิสเตอร์ SPDR (SPI Data Register)
- บิตที่ 5..1 : Res (Reserved Bits)
บิตที่ 1 ถึง 5 สงวนไว้ อ่านได้เป็น "0"
- บิตที่ 0 : บิต SPI2X (Double SPI Speed Bit)
หากกำหนดบิต SPI2X เป็น "1" จะเป็นการเพิ่มความถี่สัญญาณพิก้าเป็น 2 เท่า (2 fosc) เมื่อทำงานในโหมดมาสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SPD (SPI Data Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	MSB							LSB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	x	x	x	x	x	x	x	x

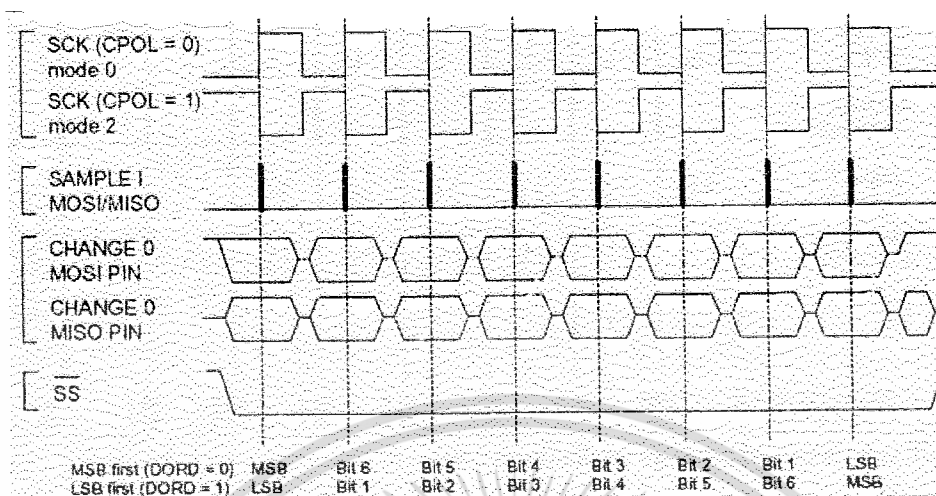
- รีจิสเตอร์ข้อมูล สำหรับเขียนอ่านเพื่อรับส่งข้อมูลในโหมด SPI ทั้งส่วนของ มาสเตอร์และสเลฟ

2.2.4 โหมดการรับส่งข้อมูล

การกำหนดรูปแบบของเฟสและลักษณะขอบขาของสัญญาณนาฬิกา จะถูกควบคุมผ่านทาง บิต CPHA และ CPOL โดยสามารถกำหนดได้ 4 รูปแบบหรือ 4 โหมดสัญญาณนาฬิกาแสดงดัง ตารางด้านล่างนี้ และรูปแบบสัญญาณดังรูปถัดไป

ตารางที่ 2.5 การกำหนดโหมดสัญญาณนาฬิกาในการรับส่งข้อมูล

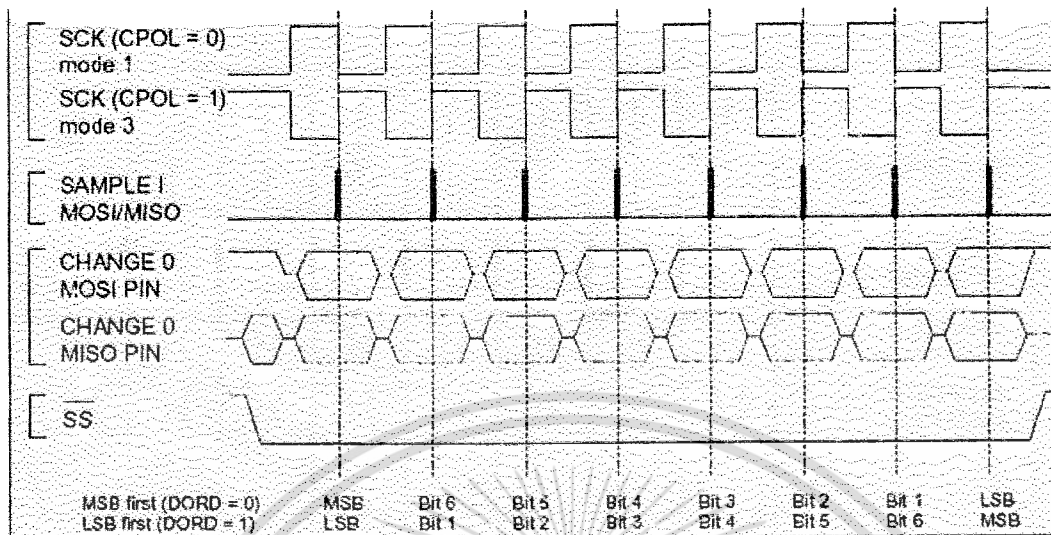
SPI โหมด	การกำหนดค่าบิต	ต้นสัญญาณ	ปลายสัญญาณ
0	CPOL = 0, CPHA = 0	อ่านข้อมูล (ขอบขาขึ้น)	อ่านข้อมูล (ขอบขาลง)
1	CPOL = 0, CPHA = 1	เปลี่ยนแปลงข้อมูล (ขอบขาขึ้น)	เปลี่ยนแปลงข้อมูล (ขอบขาลง)
2	CPOL = 1, CPHA = 0	อ่านข้อมูล (ขอบขาลง)	อ่านข้อมูล (ขอบขาขึ้น)
3	CPOL = 1, CPHA = 1	เปลี่ยนแปลงข้อมูล (ขอบขาลง)	เปลี่ยนแปลงข้อมูล (ขอบขาขึ้น)



รูปที่ 2.5 รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 0

จากรูปด้านบน สัญญาณ SPI ในโหมด 0 และ 2 จะเริ่มอ่านข้อมูลที่ต้นสัญญาณ SCK โดยมีรูปแบบการทำงานดังนี้

- เมื่อ SPI ทำงาน โหมด 0 (CPOL = 0 และ CPHA = 0) การอ่านข้อมูล (Sample I MOSI/MISO) ที่ขา MOSI/MISO จะเริ่มค้นอ่านข้อมูลเมื่อขา SCK ที่ขอบขาขึ้น (เปลี่ยนจาก Low เป็น High) โดยการเปลี่ยนแปลงข้อมูลที่ขา MOSI และ MISO จะอยู่ระหว่างสัญญาณ SCK เป็น High หรือ Low (ระหว่างเปลี่ยนจาก High เป็น Low)
- เมื่อ SPI ทำงาน โหมด 2 (CPOL = 0 และ CPHA = 0) การอ่านข้อมูลที่ขา MOSI/MISO จะเริ่มค้นอ่านข้อมูลเมื่อขา SCK ที่ขอบขาลง (เปลี่ยนจาก High เป็น Low) โดยการเปลี่ยนแปลงข้อมูลที่ขา MOSI และ MISO จะอยู่ระหว่างสัญญาณ SCK เป็น Low หรือ High (ระหว่างเปลี่ยนจาก Low เป็น High)
- การเลื่อนข้อมูลจะถูกกำหนดในบิต DORD ว่าต้องการให้บิต MSB หรือบิต LSB ของข้อมูลถูกเลื่อนเข้าไปก่อนหรือหลัง
- ขา SS จะต้องเป็น Low ระหว่างที่มีการส่งข้อมูล



รูปที่ 2.6 รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 1

จากรูปด้านบน สัญญาณ SPI ในโหมด 1 และ 3 จะเริ่มอ่านข้อมูลที่ต้นสัญญาณ SCK โดยมีรูปแบบการทำงานเช่นเดียวกับรูปก่อนหน้านี้

2.2.5 หลักการสำคัญของการใช้ SPI

สถานะเริ่มต้นการทำงานของอุปกรณ์สเลฟ (Slave) จะเริ่มต้นด้วย Low ที่ขาสัญญาณ ENABLE อุปกรณ์ SPI จะมีเงื่อนไขให้ส่งข้อมูลที่ขอบขาขึ้นหรือขอบขาลงของขาสัญญาณ SCK แตกต่างกันไปขึ้นอยู่กับข้อกำหนดของอุปกรณ์ SPI นั้น นอกจากนี้ต้องคำนึงถึงความถี่สูงสุดของสัญญาณนาฬิกาของอุปกรณ์สเลฟที่ใช้งาน โดยค่าความถี่สูงสุดของสัญญาณนาฬิกาที่ใช้ต้องมีค่าไม่เกินความถี่สัญญาณนาฬิกาของอุปกรณ์สเลฟ

2.3) การใช้งานจอ GLCD Color 6610

คุณสมบัติ

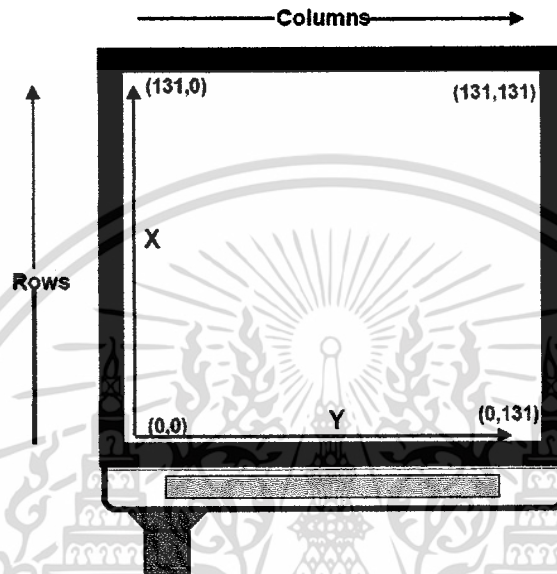
- ชิพประมวลผล Epson S1D15G10 Compatible Controller
- รองรับระบบไฟ 3.3 V ถึง 5.5 V
- ความละเอียดหน้าจอ 132x132 pixel
- ใช้การสื่อสารอนุกรมแบบ 9 Bit SPI Serial Interface
- เชื่อมต่อสัญญาณโดยตรงกับไมโครคอนโทรลเลอร์
ได้ทั้งระบบไฟ 3.3 และ 5 โวลท์
- ความละเอียดสี 4,096 เฉดสี

Pin No.	ชื่อขาสัญญาณ	คำอธิบาย
1	VBL	ขาสัญญาณไฟบวก ที่จ่ายไปเลี้ยงไฟ LED Back-Light ช่วงแรงดัน 3.3 V ถึง 5.5 V
2	VCC	ขาสัญญาณไฟเลี้ยง LCD ช่วงแรงดันที่ใช้ 3.3 V ถึง 5.5 V
3	GND	ขาสัญญาณกราวด์
4	RESET	ขาสัญญาณรีเซ็ตของ LCD ทำงานที่ลอจิก 0
5	SDATA	ขาสัญญาณอนุกรม Serial Data
6	SCLK	ขาสัญญาณอนุกรม Serial Clock
7	CS	ขาสัญญาณ Chip Select สำหรับเปิดปิดการรับส่งข้อมูลของ LCD 1 = ปิดพอร์ตการรับส่งข้อมูล (SDATA และ SCLK เป็น Hi-impedance) 0 = เปิดพอร์ตการสื่อสารข้อมูล
8	BL	ขาสัญญาณควบคุม เปิด-ปิด ไฟ LED Back-Light 1 = เปิดไฟ LED Back-Light 0 = ปิดไฟ LED Back-Light

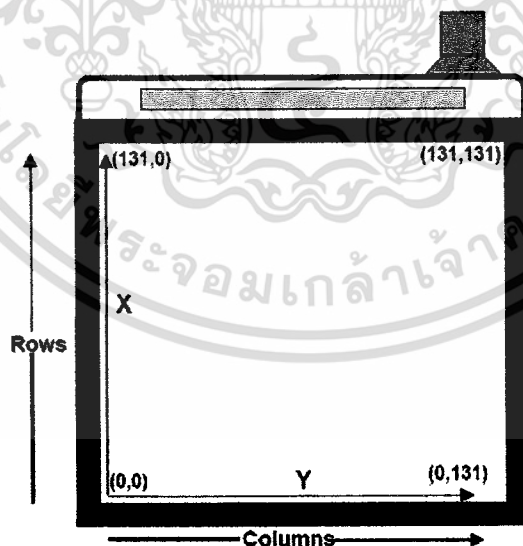
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4) การปรับจอ GLCD (LCD Display Orientation)

จอ GLCD Color 6610 มีความละเอียดจอแสดงผล 132x132 พิกเซล มีบิตสี 12บิต ซึ่งแบ่งเป็น บิตสีแดง 4บิต บิตสีเขียว 4บิต บิตสีน้ำเงิน 4บิต ซึ่งสามารถปรับคอล์มภ์และแถวได้สองแบบ ซึ่งการปรับสามารถปรับได้ 180 องศารูป



รูปที่ 2.7 การใช้งานคอล์มภ์และแถวแบบปกติ



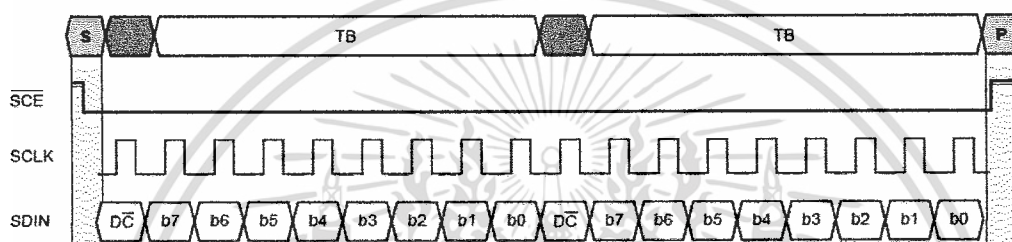
รูปที่ 2.8 การปรับการใช้งานคอล์มภ์และแถว 180 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5) การติดต่อ GLCD 6100 ด้วยไมโครคอนโทรลเลอร์

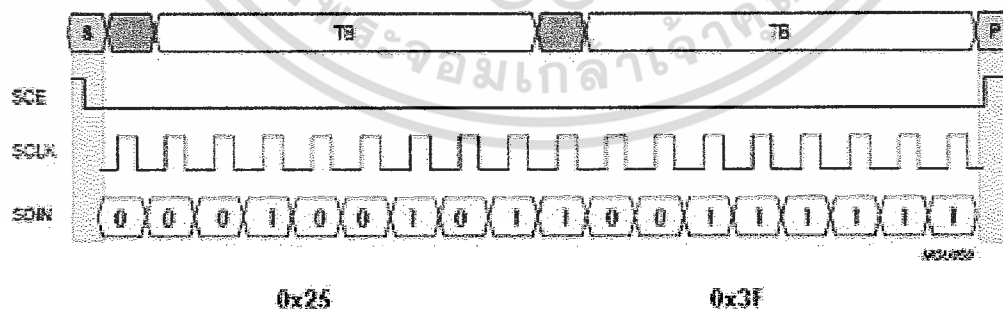
ภายในจอ LCD 6100 จะมี LCD Controller ที่ทำหน้าที่สแกนภาพออกที่จอ (เหมือนเวลาเราใช้ 7-segments) โดยจะมีหน่วยความจำ SRAM ในตัว และเวลาเราติดต่อกับจอ ก็คือการติดต่อกับตัว LCD Controller ซึ่งจริงๆแล้ว สามารถติดต่อได้ทั้งแบบ parallel และแบบอนุกรม แต่ผู้ผลิตจอ LCD 6100 ได้ต่อวงจรในส่วนนี้เรียบร้อยแล้ว และให้เราติดต่อได้ทางอนุกรมเท่านั้น

สำหรับจอ LCD 6100 ที่ใช้ IC PCF8833 ของ Phillips เป็น LCD Controller ใช้การเชื่อมต่อแบบ SPI 9-bit มี timing diagram ดังรูป



รูปที่ 2.9 timing diagram จอ LCD 6100 ที่ใช้ IC PCF8833 ใช้การเชื่อมต่อแบบ SPI 9-bit

จาก Timing diagram (รูปนำมาจาก datasheet ของ PCF8833) การส่งข้อมูลจะส่ง MSB ก่อน LSB และก่อนจะเป็นข้อมูล 8 บิต จะต้องส่ง bit DC เพื่อบอก LCD Controller ว่าเป็น ไบต์ที่จะส่งนั้น เป็นคำสั่ง หรือเป็นค่าพารามิเตอร์ สำหรับส่งคำสั่ง บิต DC จะต้องเป็นลอจิก 0 และสำหรับพารามิเตอร์ บิต DC จะต้องเป็น 1 เช่น ถ้าต้องการกำหนด ความเข้มของภาพ (contrast) จะต้องส่งคำสั่ง 0x25 ตามด้วย ค่าพารามิเตอร์ความสว่าง ในที่นี้กำหนดเป็น 0x3F



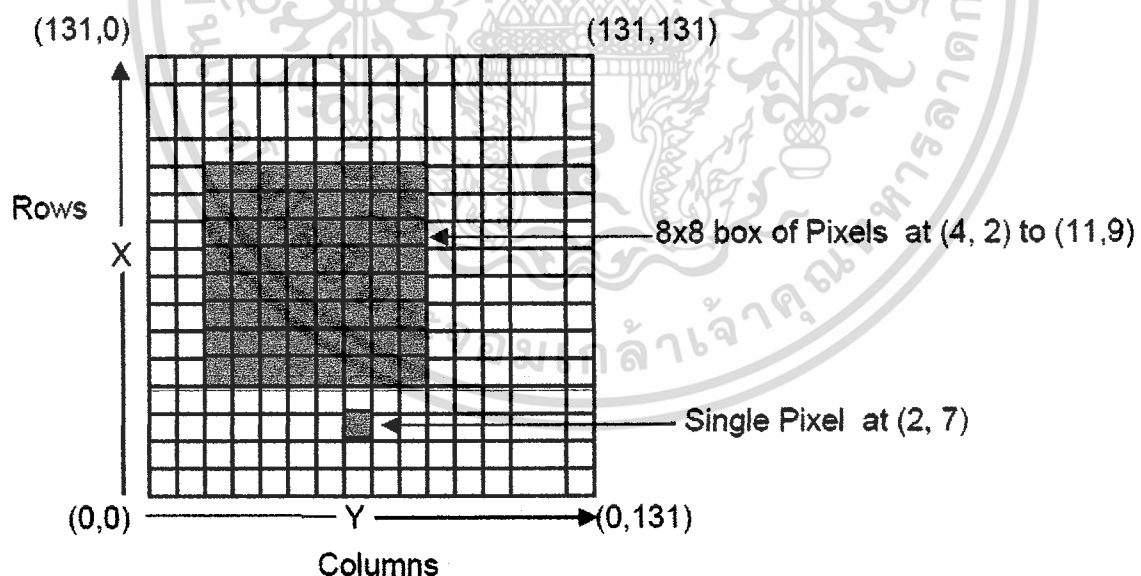
รูปที่ 2.10 การส่ง bit DC เพื่อบอก LCD Controller

สำหรับสัญญาณ CS (SCE ในรูป) ใช้กำหนดจังหวะการเริ่มต้นการส่งข้อมูล เพื่อให้ผู้ส่งและผู้รับ เข้าใจตรงกันว่าให้นับบิตไหนเป็นบิตแรก และแต่ละไบต์ สามารถส่งต่อเนื่องกัน โดยที่ไม่ต้องหยุดการส่งข้อมูลแล้วเริ่มใหม่ โดยสามารถเริ่มต้นครั้งเดียวแล้วส่งคำสั่งตามด้วยค่าพารามิเตอร์แล้วส่งคำสั่งถัดไปได้ทันที

จอ GLCD 6610 จะใช้จุดเชื่อม SPI 2 สาย ซึ่ง SPI ทางตัวควบคุม AVR จะสร้างสัญญาณการจับเวลาและข้อมูล และจอจะเป็นอุปกรณ์การทำงาน ซึ่ง Olimex จะเลือกเพื่อไม่ทำให้สัญญาณ MISOO เกิดผลที่จะขอมรับตัวควบคุม AVR เพื่ออ่านจากจอ LCD ดังนั้น จอแสดงผลของจอคือการเขียนเท่านั้น

2.6) การจัดการหน่วยความจำพิกเซล

ตัวควบคุม Epson S1D15G10 จะมีหน่วยความจำค่า 17424 [132x132] ที่แต่ละค่าคือ 12บิต (สี 4บิตสำหรับแดง เขียว น้ำเงิน) โดยการระบุที่อยู่ของพิกเซลกับ Page Address Set (กำหนดที่อยู่หน้า) และ Column Address Set (กำหนดที่อยู่คอลัมน์) หน่วยความจำพิกเซลจะมี 132แถวและ 132 คอลัมน์ตามที่ถูกแสดงในภาพ



รูปที่ 2.11 แสดงหน่วยความจำพิกเซลจะมี 132 แถวและ 132 คอลัมน์

ทันทีที่การวาดขอบเขตถูกสร้างขึ้น (เป็นพิกเซลเดี่ยวหรือกลุ่มพิกเซลสี่เหลี่ยม) การดำเนินการหน่วยความจำผลใดๆจะถูกกำหนดที่ขอบเขตนั้น หากพยายามเขียนพิกเซลมากกว่าถูกกำหนดโดยขอบเขต พิกเซลพิเศษจะถูกทิ้งโดยตัวควบคุม ซึ่งตัวควบคุม Epson S1D15G00 จะมีแผนงานหน่วยความจำเดียวกันเป็น Philips/NXP PCF8833

2.7) ข้อมูล 12 บิต

ตัวควบคุมจอ LCD Philips PCF8833 จะมี 3 ทางเพื่อระบุพิกเซล

12 บิตต่อพิกเซล (แบบเดิม)

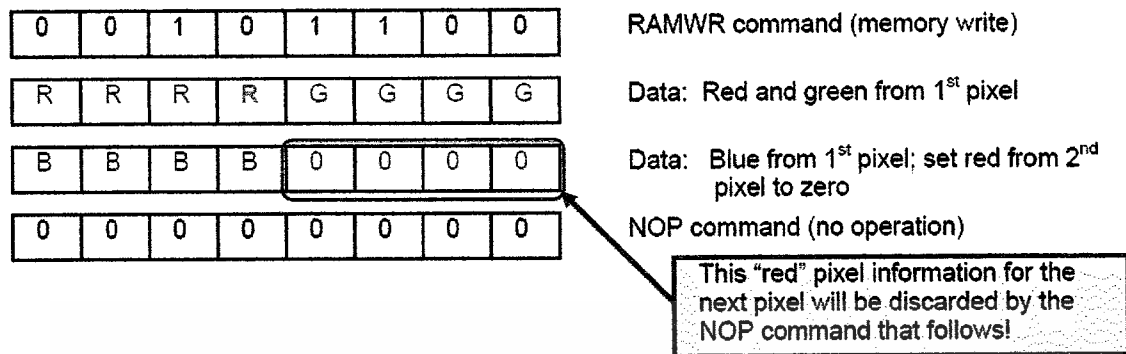
การเลือกแบบพิกเซล 12 บิตถูกทำให้สำเร็จโดยการส่ง Color interface Pixel Format [0x34] (การจัดพิกเซลจุดเชื่อมสี) โดยไบต์ข้อมูลเดี่ยวที่มีค่า 3 การแปลรหัสนี้ต้องการการเขียนหน่วยความจำและไบต์ข้อมูล 1.5 เพื่อระบุพิกเซลเดี่ยว ซึ่งไบต์ที่ถูกบรรจุครั้งที่ 2 พิกเซลจะมี 3 ไบต์และกระบวนการจะทำซ้ำจนกระทั่งการวาดขอบเขตถูก

0	0	1	0	1	1	0	0	RAMWR command (memory write)
R	R	R	R	G	G	G	G	Data: Red and green from 1 st pixel
B	B	B	B	R	R	R	R	Data: Blue from 1 st pixel; red from 2 nd pixel
G	G	G	G	B	B	B	B	Data: Green and blue from 2 nd pixel

รูปที่ 2.12 แสดงการแปลรหัสพิกเซล 12 บิต

ถ้าหากระบุพิกเซลเดียวกับ 2 ไบต์ข้อมูล และข้อมูลสีแดง 4 บิตจากพิกเซลต่อไป (ปกติจะกำหนดที่ศูนย์) จะทำให้พิกเซลข้างเคียงยุ่งเหยิงหรือไม่ ซึ่งคำตอบคือ PCF8833 ไม่ได้เขียนที่จอ RAM เมื่อมันได้รับพิกเซลที่สมบูรณ์ ซึ่งบิตสีแดงจากพิกเซลต่อไปได้รอคอยสำหรับการทำให้สีที่ยังคงอยู่สมบูรณ์

การปรากฏคำสั่งใดๆจะยกเลิกการดำเนินการก่อนหน้านี้และทิ้งข้อมูลพิกเซลที่ไม่ได้ใช้ และเพื่อความปลอดภัยนั้น เราได้เติม NOP ใน LCDSetPixel () เพื่อรับรองว่าข้อมูลสีแดงที่ไม่ได้ใช้จากพิกเซลต่อไปถูกทิ้ง



รูปที่ 2.13 แสดงวิธีการส่งพิกเซลเดี่ยวที่ใช้การแปลรหัส 12 บิต และบันทึกว่า 4 บิตสีแดงสุดท้ายจากพิกเซลต่อไปถูกละเลย

8 บิตต่อพิกเซล

การเลือกแบบพิกเซล 8 บิตถูกทำให้สำเร็จ โดยการส่งการจัดพิกเซลจุดเชื่อมสี โดยไบท์ข้อมูลเดี่ยวที่มีค่า 2 การแปลรหัสนี้ต้องการการเขียนหน่วยความจำและ ไบท์ข้อมูลเพื่อระบุพิกเซลเดี่ยว ซึ่งไบท์ข้อมูลจะมีข้อมูลสีสำหรับ 1 พิกเซล ซึ่งข้อมูลสีถูกแปลรหัสเป็นสีแดง 3 บิต สีเขียว 3 บิต สีน้ำเงิน 2 บิตตามที่แสดงในรูป



รูปที่ 2.14 แสดงการแปลรหัสเป็นสีแดง 3 บิต สีเขียว 3 บิต สีน้ำเงิน 2 บิต

สิ่งสำคัญเพื่อบันทึกในที่นี้คือการที่การแปลรหัส 8 บิตถูกเปลี่ยน ไปที่การแปลรหัส 12 บิต โดยตารางสีที่กำหนดล่วงหน้า ซึ่งตารางการกำหนดสีนี้จะเปลี่ยนสีแดง 3 บิต ไปเป็นสีแดง 4 บิต สีเขียว 3 บิต ไปเป็นสีเขียว 4 บิต และสีน้ำเงิน 2 บิต ไปเป็นสีน้ำเงิน 4 บิต ซึ่งสิ่งนี้ถูกทำโดยการระบุของตารางสีที่ 20 ในขั้นตอนการทำแรกเริ่มให้พิจารณาจุดที่ตามมา ซึ่งการแก้ปัญหาจอ โนเกีย 6100 คือ 132x132 พิกเซล, 12 บิต/พิกเซล และตั้งแต่การแปลรหัส 8 บิต/พิกเซลถูกเปลี่ยน โดยตารางสีไปที่ 12 บิต/พิกเซล ก็ไม่มีการบันทึกของหน่วยความจำจอ ซึ่งการแปลรหัส 8 บิต/พิกเซลจะใช้ไบท์ข้อมูล 1/3 เพื่อเติมพื้นที่ ดังนั้นมันจะเป็นการกระทำในจำนวน ไบท์ที่ถูกส่งผ่าน ซึ่งการแปลรหัส 8 บิต/พิกเซลจะสร้างรูปภาพที่ไม่ดี

16บิตต่อพิกเซล

การเลือกแบบ 16บิต/พิกเซลถูกทำให้สำเร็จโดยการส่งการกำหนดพิกเซลจุดเชื่อมต่อโดยไบต์ข้อมูลเดียวที่มีค่า 5 การแปลรหัสนี้ต้องการการเขียนหน่วยความจำและ 2 ไบต์ข้อมูลเพื่อระบุพิกเซลเดียว ซึ่งข้อมูลที่ถูกแปลรหัสเป็น 5บิตสำหรับสีแดง 6บิตสำหรับสีเขียว และ 5บิตสำหรับสีน้ำเงินตามที่ถูกแสดงในภาพ 10

0	0	1	0	1	1	0	0	RAMWR command (memory write)
R	R	R	R	R	G	G	G	Data: Red (5 bits), Green (6 bits)
G	G	G	B	B	B	B	B	Data: Green (6 bits), Blue (5 bits)

รูปที่ 2.15 แสดงการแปลรหัสเป็น 5บิตสำหรับสีแดง 6บิตสำหรับสีเขียว และ 5บิตสำหรับสีน้ำเงิน

การแปลรหัสนี้ถูกเปลี่ยนโดยตัวควบคุมที่ใช้เทคนิคการรวมจุดสีที่ข้อมูล 12บิตสำหรับพิกเซล RAM ซึ่งผลรวมคือทำให้การเปลี่ยนสี 65k ซึ่งมุมมองของเราคือการที่ไม่มีใครแสดงโมนาลิซ่าในจอแบนนี้ ดังนั้น การแปลรหัสสี 16บิตจะถูกใช้น้อยมาก ซึ่งตัวควบคุม Epson S1D15G00 จะสนับสนุนแบบ 8, 12บิตแต่ไม่สนับสนุนแบบ 16บิต

เราจะสามารถแสดงภาพตามต้องการได้ จากที่เรากำหนด โหมดสีเป็นแบบ 8 บิตนั้น ใน 8 บิต ประกอบด้วยโทนสีแดง (R) 3 บิต สีเขียว (G) 3 บิต และสีน้ำเงิน (B) 2 บิต โดยเรียงลำดับจาก MSB ไปยัง LSB เป็น RGB ซึ่งจะเห็นว่าเราสามารถระบุโทนสีแดง และเขียวได้ 8 โทน กับอีก 4 โทนสำหรับสีน้ำเงิน นอกจากนี้ เราสามารถสลับโทนสีแดง กับน้ำเงินได้ เพื่อให้สามารถแสดงรายละเอียดสีน้ำเงินได้มากขึ้น การใส่สีให้กับพิกเซล คือ การกำหนด code สี ให้กับพิกเซล เนื่องจากมี 8 บิต จึงสามารถแสดงได้ 256 สี การเขียน code สี ให้กับ pixel ก็คือการเขียน code สีลงไปในหน่วยความจำที่อยู่ใน LCD Controller จากนั้นในการแสดงผล LCD Controller จะนำ code สี ที่แต่ละตำแหน่ง มาเปิดตารางดูว่า code หรือ โทนของแต่ละสี เป็นระดับสีเท่าไร ใน 16 ระดับ (4 บิต สำหรับแต่ละสี) ซึ่งระดับ 0 ก็คือมืดที่สุด และระดับ 15 คือสูงที่สุด เช่น ถ้าเป็นองค์ประกอบสีแดง R = 15 ก็หมายถึงสีแดงสด ถ้า R = 0 คือสีดำ

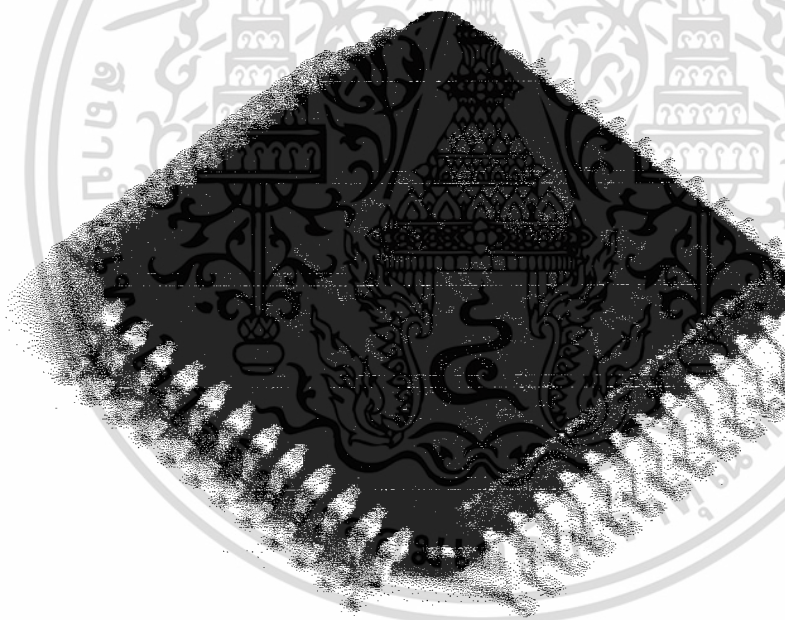
ดังนั้น สี 256 สี ที่เราสามารถแสดงได้ ก็คือ การเลือกสีมาจากสีทั้งหมด $16 \times 16 \times 16 = 4096$ สี นั่นเอง โดยเราสามารถกำหนดตารางสี สำหรับแต่ละ โทน ได้โดยใช้คำสั่ง RGBSET แต่เนื่องจาก PCF8833 นั้นได้ทำการกำหนดค่าเริ่มต้นของตารางสีดังกล่าวไว้แล้ว จึงไม่จำเป็นต้องกำหนดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

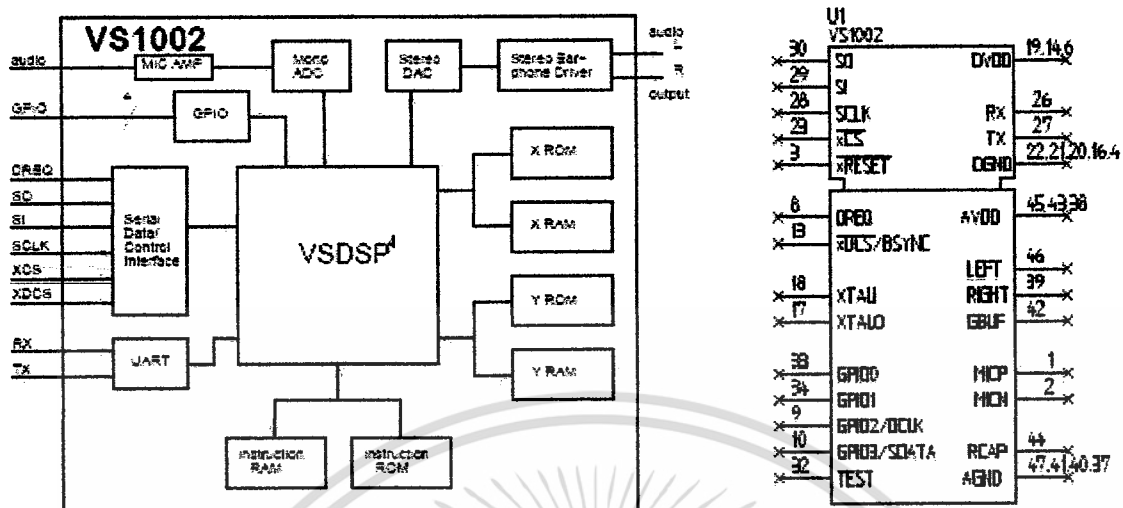
สำหรับการเขียนค่าสีลงในหน่วยความจำของจอ PCF8833 ได้กำหนดวิธีการไว้คือ เราต้องกำหนดคอลัมน์เริ่มต้น คอลัมน์สุดท้าย แถวแรก และแถวสุดท้าย แล้วจึงเขียนลงในหน่วยความจำที่เดียวอย่างต่อเนื่อง โดยเมื่อเขียนจนจบแถวแรก ตำแหน่งแอดเดรส จะเลื่อนไปยังแถวถัดไปทางซ้ายสุดให้เอง

หมายเหตุ จากคุณสมบัติดังกล่าว ในโครงการนี้ใช้ AVR Atmega128 เป็น MCU เชื่อมต่อกับจอ LCD และส่งข้อมูลแบบเป็น I/O คือไม่ได้ใช้ SPI เนื่องจากส่งได้ 8 บิต (ไม่พอสำหรับ 9 บิต)

2.8) การใช้งานไอซีถอดรหัสไฟล์ MP3 และ WAVE (IC VLSI VS1002xx)



รูปที่ 2.16 ไอซี VS1002D



รูปที่ 2.17 โครงสร้างไอซี VS1002D

คุณสมบัติของไอซีถอดรหัสไฟล์ MP3 และ WAVE

- ใช้ไอซีถอดรหัสไฟล์ MP3 ของ VLSI เบอร์ VS1002D
- สามารถถอดรหัสไฟล์ MPEG ซึ่งใช้การเข้ารหัสแบบ MPEG1.0 & 2.0 Audio layer III (CBR + VBR) รวมทั้ง WAV และ PCM ได้
- สามารถเข้ารหัสสัญญาณเสียงจาก ไมโครโฟนให้เป็นข้อมูลแบบมาตรฐาน ADPCM ได้
- รองรับการส่งถ่ายข้อมูลแบบต่อเนื่อง (Streaming Data) สำหรับไฟล์ข้อมูลแบบ MP3 หรือ WAVE ได้
- มีคำสั่งปรับแต่งเสียงทุ้ม (Bass Control)
- ทำงานด้วยสัญญาณนาฬิกา 12.288 MHz โดยสามารถ คุม ความถี่เป็นสองเท่าได้จาก PLL ภายใน
- มีวงจรแปลงกลับข้อมูลเป็นเสียงแบบ DAC คุณภาพสูงพร้อมวงจรภาคขยายเสียงแบบ Stereo
- ทำงานที่แรงดันไฟตรงระหว่าง 2.5V ถึง 3.6V
- รองรับการเชื่อมต่อสัญญาณกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม SPI ได้
- สามารถดัดแปลงการทำงานให้เป็นเครื่องเล่น MP3 แบบ Standalone โดยไม่ต้องใช้ การควบคุมการทำงานจากไมโครคอนโทรลเลอร์ได้ (ดูรายละเอียดเพิ่มเติมได้จาก Application Note ของ VLSI)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถอดรหัสไฟล์ MP3 และ WAVE เป็นชุดถอดรหัสไฟล์ MP3 เพื่อแปลงเป็นเสียง โดยได้เลือกนำเอาไอซีสำหรับถอดรหัสไฟล์ MP3 ของ VLSI เบอร์ VS1002D ซึ่ง VS1002D จัดว่าเป็นไอซีถอดรหัสไฟล์ MP3 ที่มีคุณภาพดี และ ใช้งานง่ายมากที่สุดตัวหนึ่ง โดย VS1002D สามารถถอดรหัสไฟล์ MP3 และ WAVE ได้ทันที พร้อมทั้งมีภาค Output แบบ Analog เป็นสัญญาณเสียงแบบ Stereo โดยสามารถนำไปต่อกับชุดหูฟัง หรือชุดขยายเสียงได้ทันที สำหรับการเชื่อมต่อกับไมโครคอนโทรลเลอร์นั้นก็สามารถทำได้โดยง่าย โดยการเชื่อมต่อกับพอร์ตอนุกรมแบบ SPI มาตรฐานได้ทันที จึงทำให้ผู้ใช้สามารถนำไปประยุกต์ใช้งานร่วมกับไมโครคอนโทรลเลอร์ต่างๆได้ตามต้องการ นำไปเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์รุ่นใดรุ่นหนึ่ง สามารถนำไปประยุกต์คัดแปลงใช้งานในรูปแบบต่างๆได้ โดยอิสระ



บทที่ 3

การติดต่อกับ SD การ์ดเบื้องต้น และการ FAT

ในบทนี้นำเสนอตัวอย่างการเขียนโปรแกรมเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ LPC 2148 กับ การ์ดหน่วยความจำชนิด SD (เพื่อความกระชับ ในที่นี้จึงขอเรียกว่า SD การ์ด) เพื่ออ่านและเขียนข้อมูล เพื่อให้เกิดความสมบูรณ์ จึงแบ่งเนื้อหาออกเป็น 3 ส่วนดังนี้

1. ความรู้เบื้องต้นเกี่ยวกับ SD การ์ด
2. ตัวอย่างการสร้างไลบรารีสำหรับอ่านเขียนข้อมูล SD การ์ดเบื้องต้น
3. ตัวอย่างการทดลอง

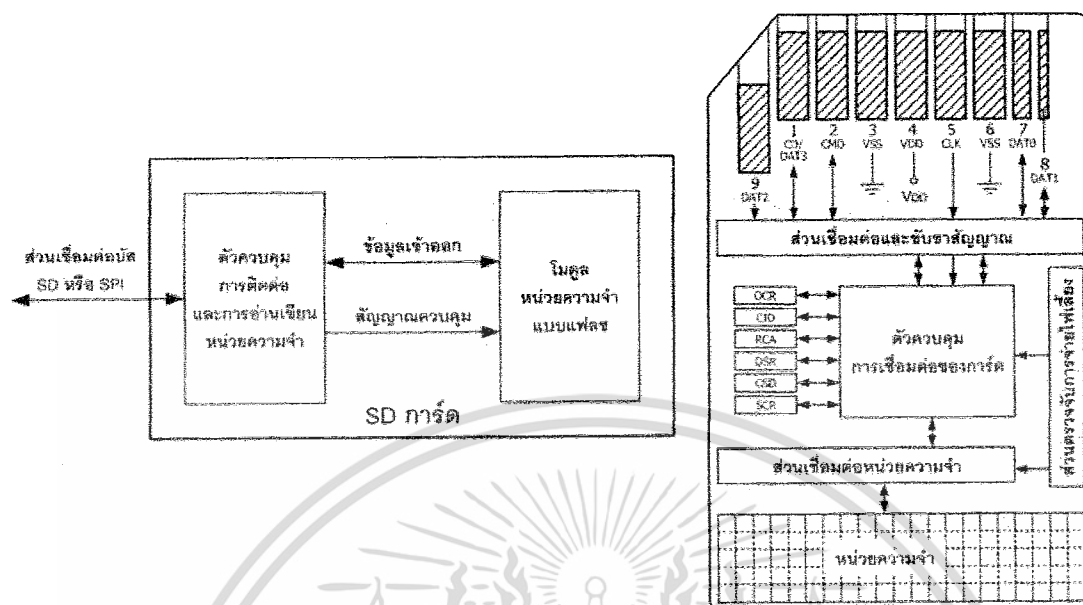
3.1) ความรู้เบื้องต้นเกี่ยวกับ SD การ์ด

SD การ์ดเป็นหน่วยความจำแบบเขียนและลบใหม่ได้แบบหนึ่งที่ใช้เทคโนโลยีหน่วยความจำแบบแฟลช ชื่อเต็มคือ Secure Digital Card มีลักษณะการทำงานและการติดต่อกคล้ายกับการ์ดหน่วยความจำแบบ MMC หรือ Multi Media Card หากแต่ใน SD การ์ดได้บรรจุส่วนการรักษาดูแลข้อมูลเข้าไปเพิ่มเติม ในรูปที่ 1 แสดงไดอะแกรมการทำงานของ SD การ์ด จะเห็นว่ามีส่วนประกอบหลัก 2 ส่วนคือ โมดูลหน่วยความจำแบบแฟลชและตัวควบคุม การติดต่อกับ SD การ์ดสามารถกระทำผ่านบัส SD หรือบัส SPI

3.1.1 คุณสมบัติเด่นของ SD การ์ด

SD การ์ดเกิดขึ้นจากความร่วมมือของ 3 บริษัทคือ Matsushita Electric Industrial (MEI), SanDisk Corporation (Sandisk) และ Toshiba Corporation (Toshiba) มีการกำหนดคุณสมบัติต่างๆ รวมถึงมาตรฐานการติดต่อที่ชัดเจนภายในการค้ากับดูแล โดย SD card Association (www.sdcard.org)

ในปัจจุบัน SD การ์ดได้รับความนิยมสูงมาก โดยเฉพาะในอุปกรณ์สารสนเทศสมัยใหม่ ไม่ว่าจะเป็นกล้องดิจิทัล โทรศัพท์เคลื่อนที่ เครื่องเล่น MP3 เป็นต้น ทั้งนี้เนื่องจาก SD การ์ดได้รับการออกแบบให้มีความโดดเด่นในทุกด้านที่หน่วยความจำชนิดนี้พึงมี 5 ประการ ดังนี้



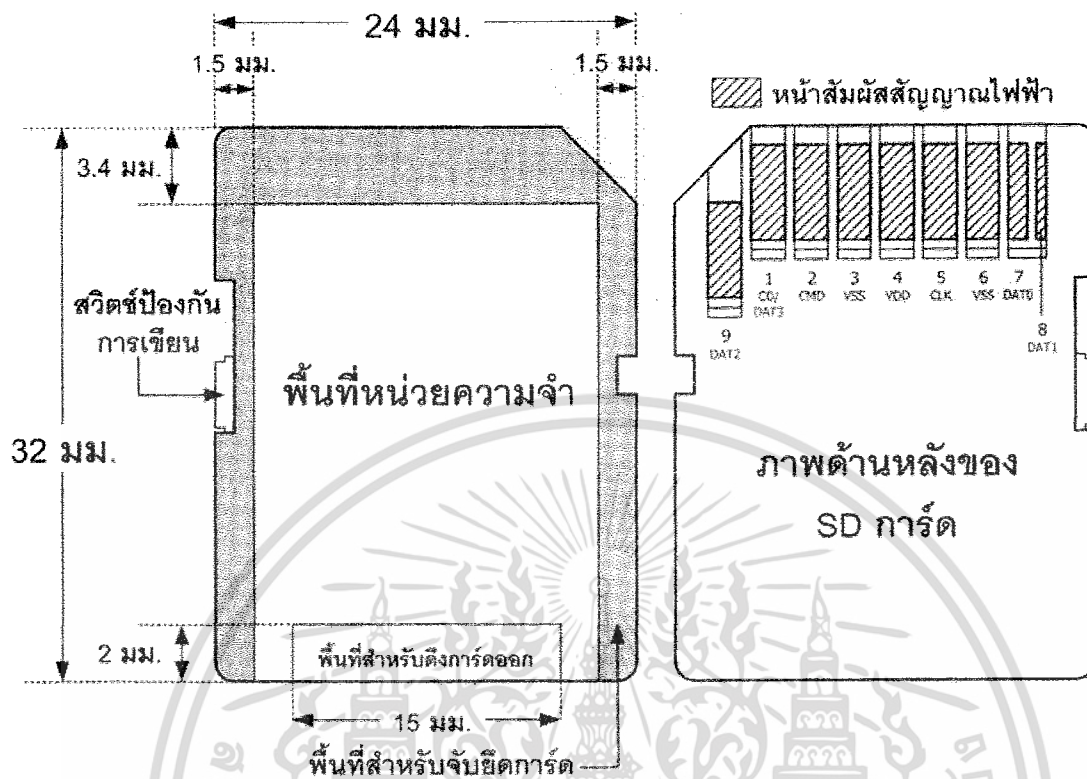
รูปที่ 3.1 แสดงโคโอะแกรมการทำงานเบื้องต้นของ SD การ์ด

ขนาดเล็กและบาง (compact & slim)

SD การ์ดรุ่นมาตรฐานมีขนาด 24 x 32 มิลลิเมตร และหนาเพียง 2.1 มิลลิเมตร ดังแสดงในรูปที่ 2 จะเห็นได้ว่ามีขนาดเล็กและบางมาก จึงทำให้สามารถพกพาได้สะดวก นอกจากนั้นได้มีการพัฒนา SD การ์ด ให้มีขนาดเล็กลงไปอีกเรียกว่า mini SD ซึ่งมีขนาดเพียง 20 x 21.5 มิลลิเมตร หนาเพียง 1.4 มิลลิเมตร ซึ่งเล็กและบางกว่ารุ่นมาตรฐานพอสมควรทีเดียว

ความจุสูง (large capacity)

แม้ว่า SD การ์ดจะมีขนาดเล็กและบาง แต่นั่นไม่ใช่ข้อจำกัดในด้านความจุ SD การ์ดมีความสูงมากเมื่อเทียบกับขนาด โดยในยุคแรก ๆ ของการพัฒนาจะมีความจุ 8 MB และได้รับการพัฒนาให้มีความจุสูงเพิ่มขึ้นเป็น 16 และ 32 MB ภายใต้เทคโนโลยีไบนารีแนนด์ (Binary NAND technologies) จากนั้นมีการใช้เทคโนโลยีใหม่ที่เรียกว่า Multi Level Cell (MLC) NAND technologies ทำให้สามารถเพิ่มความจุเป็น 64, 128, 256, 512 MB และไปถึงระดับ GB (ในขณะที่ทำเอกสารนี้มีการพัฒนาถึง 4 GB แล้ว และมีแนวโน้มสูงมากที่จะเพิ่มความจุสูงไปถึง 8 MB) เพื่อให้สามารถรองรับไฟล์ภาพ รวมไปถึงภาพยนตร์คุณภาพสูงในระดับ DVD



รูปที่ 3.2 แสดงลักษณะทางกายภาพของ SD การ์ด

อัตราเร็วในการถ่ายทอข้อมูลสูง (high-speed data transfer)

SD การ์ดมีอัตราการถ่ายทอข้อมูลในแบบสายสัญญาณเดี่ยวผ่านทางขา DAT0 25 เมกะบิตต่อวินาที ถ้าหากถ่ายทอข้อมูลแบบ 4 สาย อัตราเร็วจะเพิ่มขึ้นเป็น 100 เมกะบิตต่อวินาที หรือ 12 MB ต่อวินาที

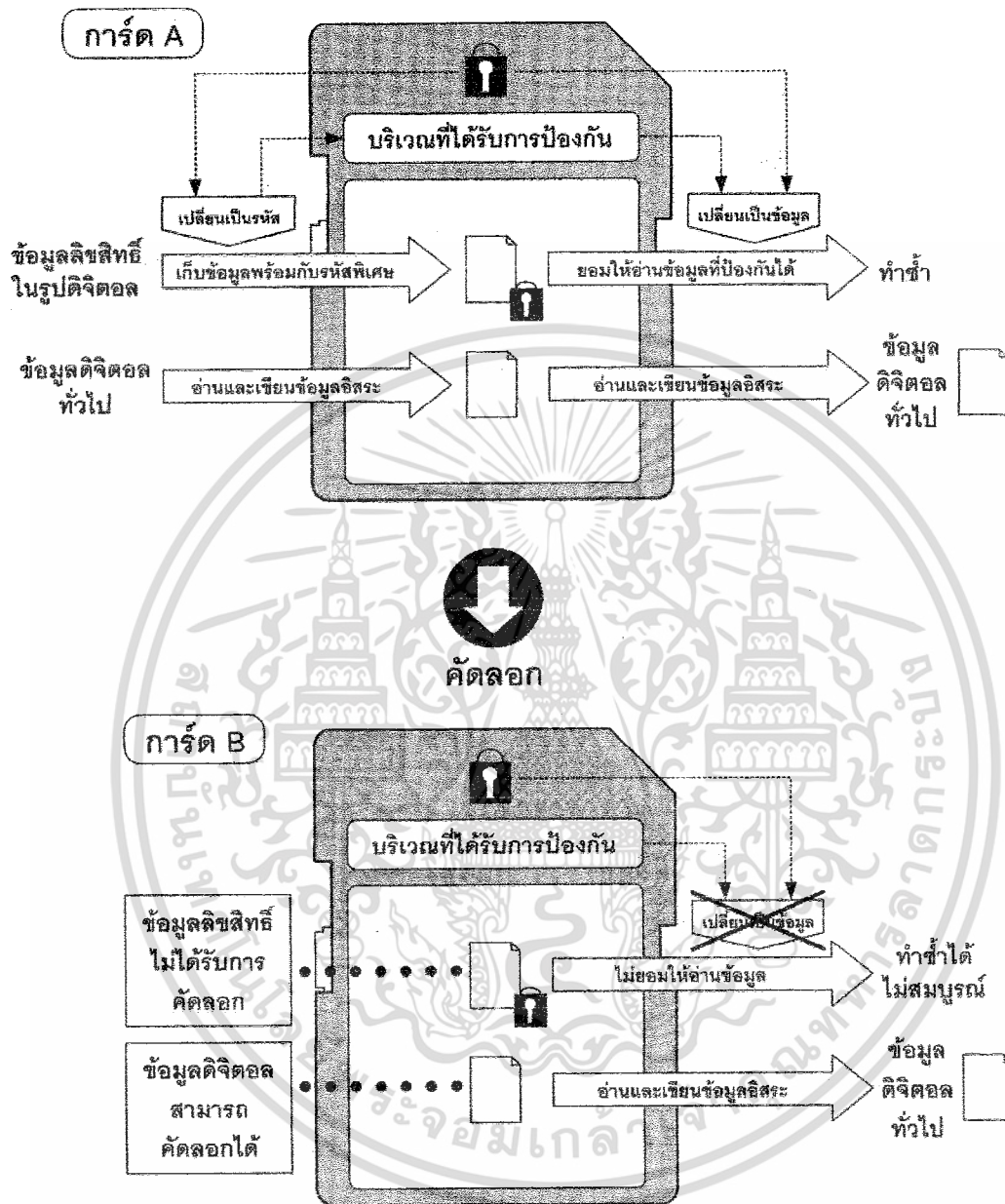
ป้องกันการละเมิดลิขสิทธิ์ (copyright protection)

SD การ์ดสามารถรองรับเทคโนโลยีการป้องกันลิขสิทธิ์ที่เรียกว่า CPRM หรือ Content Protection for recordable Media ซึ่งร่วมกันพัฒนาโดย IBM, Intel, Matsushita และ Toshiba นั่นคือ สามารถเลือกป้องกันเฉพาะข้อมูลดิจิทัลที่สงวนไว้เป็นส่วนแสดงลิขสิทธิ์ ไม่สามารถที่จะคัดลอกได้ แต่ยอมให้ข้อมูลดิจิทัลทั่วไปสามารถคัดลอก อ่าน เขียนได้ตามปกติ ดังแสดงกระบวนการนี้ในรูปที่ 3

สามารถขยายความสามารถได้ (expandability)

ใน SD การ์ดรุ่น SDIO (Secure Digital Input/Output) มีคุณสมบัติพิเศษคือ สามารถบรรจุหรืออุปกรณ์อินพุตเอาต์พุตลงใน SD การ์ด อาทิ ภาครับโทรทัศน์ดิจิทัล (digital TV tuner), ระบบแลนไร้สาย (WLAN 802.11b), วิทยุ FM เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 กระบวนการป้องกันการละเมิดลิขสิทธิ์ใน SD การ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 คุณสมบัติทางเทคนิคที่สำคัญของ SD การ์ด

- สามารถเก็บข้อมูลได้ถึง 8 GB (ในขณะที่จัดทำเอกสารนี้)
- รองรับการติดต่อแบบหนึ่งสายสัญญาณ และแบบ 4 สายสัญญาณ รวมทั้งแบบบัส SPI
- สามารถป้องกันการคัดลอกข้อมูลลิขสิทธิ์ได้
- สามารถลบ-เขียนใหม่ในแต่ละเซกเตอร์ได้ 100,000 ครั้ง
- สามารถเก็บรักษาข้อมูลได้นานมากกว่า 10 ปี
- แรงดันใช้งาน
- กรณีกระทำคำสั่งสื่อสารเบื้องต้น (CMD0, CMD15, CMD56 และ ACMD41) ใช้ 2.0 ถึง

3.6V

- กรณีกระทำคำสั่งอื่น รวมถึงเข้าถึงหน่วยความจำ ใช้ 2.7 ถึง 3.6V
- สามารถปรับสัญญาณนาฬิกาได้ในช่วง 0 ถึง 25MHz
- อัตราเร็วในการถ่ายถอดข้อมูลสูงสุด 12.5 MB ต่อวินาที (ในกรณีใช้การสื่อสารแบบ 4

สาย)

- สามารถติดต่อการ์ดได้พร้อมกันสูงสุด 10 แผ่น
- ป้องกันการเขียนด้วยสวิตช์
- สามารถกำหนดการป้องกันการเขียนได้ทางซอฟต์แวร์ ทั้งแบบชั่วคราวและถาวร
- มีสัญญาณแจ้งการถอดและใส่การ์ดกับซ็อกเก็ต

3.1.3 ระบบบัสที่ใช้ติดต่อกับ SD การ์ด

การติดต่อกับ SD การ์ดสามารถกระทำได้ 2 วิธี คือ ผ่านทางบัส SD และบัส SPI โดยสามารถสรุปได้ดังตารางที่ 1

3.1.4 ขาสัญญาณของ SD การ์ด

ขาสัญญาณมาตรฐานของ SD การ์ดมีทั้งสิ้น 9 ขา โดยมีลักษณะเป็นหน้าสัมผัสโลหะ ดังแสดงในรูปที่ 2 ส่วนการกำหนดชื่อและหน้าที่ของขาสัญญาณจะแตกต่างกันตามรูปแบบของการติดต่อดังสรุปได้ในตารางที่ 2 และ 3 โดยในตารางที่ 2 เป็นการจัดขาเมื่อติดต่อ SD การ์ดด้วยบัส SD ส่วนตารางที่ 3 เป็นการจัดขาเมื่อทำงานผ่านบัส SPI

ตารางที่ 3.1 สรุปข้อมูลสำคัญของการติดต่อกับ SD การ์ดทั้งแบบบัต SD และ SPI

การติดต่อ SD การ์ดด้วยบัต SD	การติดต่อ SD การ์ดด้วยบัต SPI
ใช้สายสัญญาณ 6 เส้น - สัญญาณนาฬิกา - สัญญาณคำสั่ง (Command) - สัญญาณข้อมูล 4 สาย	ใช้สายสัญญาณอนุกรม 3 เส้น - สัญญาณนาฬิกา - สัญญาณข้อมูลเข้า (DI) - สัญญาณข้อมูลออก (DO) - สัญญาณเลือกการ์ด CS
มีการป้องกันความผิดพลาดในการถ่ายทอข้อมูล	สามารถเลือกหรือไม่เลือกการป้องกันความผิดพลาดในการถ่ายทอข้อมูล
สามารถถ่ายทอข้อมูลได้ทั้งแบบบล็อกเดียวหรือหลายบล็อก	สามารถถ่ายทอข้อมูลได้ทั้งแบบบล็อกเดียวหรือหลายบล็อก

ตารางที่ 3.2 เป็นการจัดขาเมื่อติดต่อกับ SD การ์ดด้วยบัต SD

หมายเลขขา	ชื่อขาสัญญาณ	ชนิด	คำอธิบาย
1	CD/DAT3	อินพุต/เอาต์พุต	ตรวจสอบการ์ด/สายข้อมูลบิต 3
2	CMD	อินพุต/เอาต์พุต	สัญญาณคำสั่ง/ตรวจสอบการตอบสนอง
3	Vss	สายแหล่งจ่ายไฟ	กราวด์
4	VDD	สายแหล่งจ่ายไฟ	ไฟเลี้ยง
5	CLK	อินพุต	สัญญาณนาฬิกา
6	Vss	สายแหล่งจ่ายไฟ	กราวด์
7	DAT0	อินพุต/เอาต์พุต	สายข้อมูลบิต 0
8	DAT1	อินพุต/เอาต์พุต	สายข้อมูลบิต 1
9	DAT2	อินพุต/เอาต์พุต	สายข้อมูลบิต 2

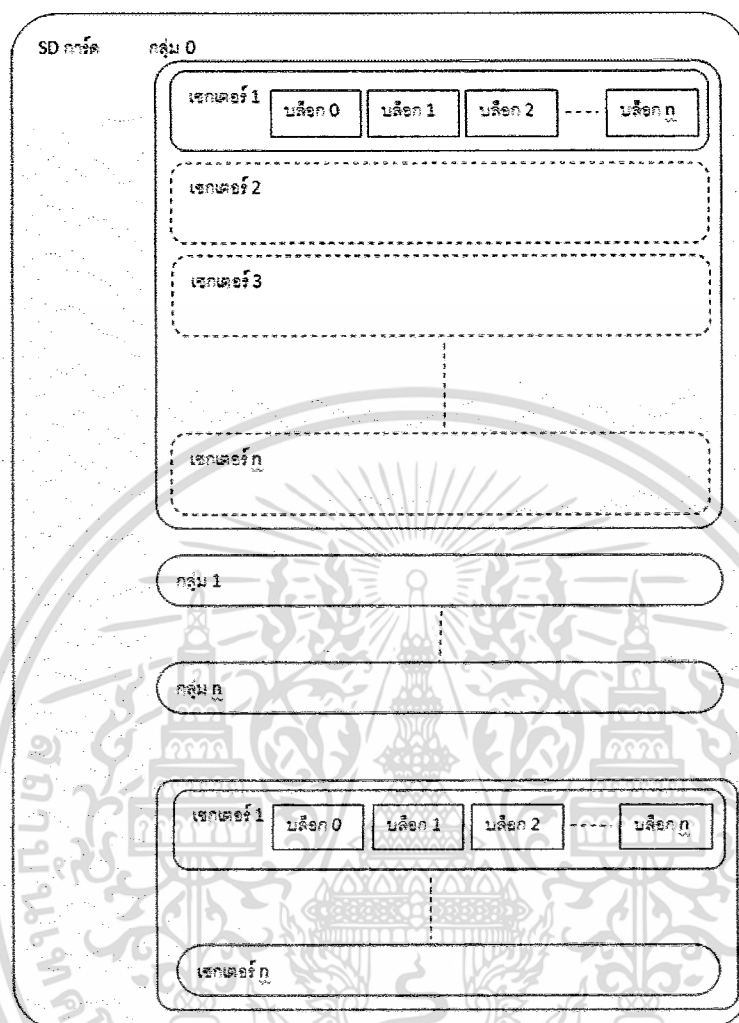
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 เป็นการจัดขาเมื่อติดต่อกับ SD การ์ดด้วยบัส SPI

หมายเลข ขา	ชื่อขาสัญญาณ	ชนิด	คำอธิบาย
1	CS	อินพุต	สัญญาณเลือกติดต่อกับ (ลอจิก "0")
2	DI	อินพุต	สัญญาณข้อมูลเข้าจากโฮสต์
3	Vss1	สายแหล่งจ่ายไฟ	กราวด์
4	VDD	สายแหล่งจ่ายไฟ	ไฟเลี้ยง
5	CLK	อินพุต	สัญญาณนาฬิกา
6	Vss2	สายแหล่งจ่ายไฟ	กราวด์
7	DO	เอาต์พุต	สัญญาณข้อมูลส่งออกจากการ์ด
8	RSV	อินพุต	สำรองไว้
9	RSV	อินพุต	สำรองไว้

3.1.5 การจัดแบ่งพื้นที่ของ SD การ์ด

หน่วยที่เล็กที่สุดของการถ่ายทอดข้อมูลใน SD การ์ดคือ 1 ไบต์ (byte) ส่วนการถ่ายทอดข้อมูลจริงนั้น ควรกระทำในลักษณะบล็อกข้อมูล โดยผู้ใช้งานสามารถกำหนดขนาดของบล็อกได้ โดยในแต่ละบล็อกสามารถบรรจุข้อมูลได้หลาย ๆ ไบต์ แต่โดยปกติแล้วมักจะเลือกใช้ที่บล็อกละ 512 ไบต์ ทั้งนี้เพื่อให้สอดคล้องกับระบบ FAT (File Allocation Table) หรือตารางสำหรับจัดวางแฟ้มข้อมูลซึ่งใช้ในระบบคอมพิวเตอร์ ในรูปที่ 4 เป็นไดอะแกรมแสดงการจัดแบ่งพื้นที่ของ SD การ์ด



รูปที่ 3.4 โค้ดแอมแสดงการจัดแบ่งพื้นที่ของ SD การ์ด

จะเห็นได้ว่าการจัดสรรเป็น 3 ส่วนหลัก คือ

(ก) บล็อกข้อมูล เป็นกลุ่มของข้อมูลที่ได้รับการกำหนดขนาดจากผู้ใช้งาน และนำไปใช้ในคำสั่งอ่านและเขียนบล็อกข้อมูล สำหรับการกำหนดและตรวจสอบขนาดของบล็อกข้อมูลสามารถกระทำได้ที่รีจิสเตอร์ CSD

(ข) เซกเตอร์ เป็นหน่วยของพื้นที่ข้อมูลใน SD การ์ดที่สัมพันธ์กับคำสั่งลบใน 1 เซกเตอร์มีหลายบล็อกข้อมูล โดยได้รับการกำหนดมาตายตัวจากผู้ผลิต ผู้ใช้งานสามารถตรวจสอบขนาดของเซกเตอร์ได้จากรีจิสเตอร์ CSD

(ค) กลุ่มป้องกันการเขียน (WP Group) เป็นพื้นที่ของหน่วยความจำที่ได้รับการจัดแบ่งให้น้อยที่สุดใช้เพื่อบรรจุข้อมูลลิขสิทธิ์ที่ไม่ต้องการให้เกิดการเขียนทับ ขนาดของพื้นที่ที่ได้รับการกำหนดมาตายตัวเช่นกัน ผู้ใช้งานสามารถตรวจสอบขนาดของพื้นที่ได้จากรีจิสเตอร์ CSD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

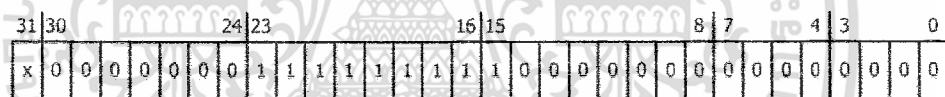
3.1.6 รีจิสเตอร์ของ SD การ์ด

มีทั้งสิ้น 6 ตัว โดยเป็นรีจิสเตอร์หลักที่ใช้ 4 ตัว, รีจิสเตอร์พิเศษ 1 ตัว และรีจิสเตอร์เสริมอีก 1 ตัว ดังแสดงในตารางที่ 5

รีจิสเตอร์ OCR (Operating Condition Register)

เป็นรีจิสเตอร์เก็บข้อมูลของค่าแรงดันไฟเลี้ยงของ SD การ์ด สำหรับตรวจสอบแรงดันของ SD การ์ด ดังแสดงในตารางที่ 6 ปกติแรงดันไฟเลี้ยงของ SD การ์ดอยู่ในช่วง 2.7 ถึง 3.6 ดังนั้นค่าของรีจิสเตอร์ OCR ควรเท่ากับ

ตารางที่ 3.4 แสดงความสัมพันธ์ของบิตข้อมูลในรีจิสเตอร์ OCR กับแรงดันของ SD การ์ด



↑ ส้ารองไว้ | เก็บข้อมูลของค่าแรงดันไฟเลี้ยงในย่าน 2.7 ถึง 3.6V | ส้ารองไว้
 บิตแสดงสถานะการจ่ายไฟเลี้ยง
 "0" หมายถึง การจ่ายไฟเลี้ยงให้แก่การ์ดยังไม่สมบูรณ์ หรือ BUSY
 "1" หมายถึง การจ่ายไฟเลี้ยงให้แก่การ์ดเสร็จสมบูรณ์

บิต OCR	แรงดัน VDD	บิต OCR	แรงดัน VDD	บิต OCR	แรงดัน VDD	บิต OCR	แรงดัน VDD	บิต OCR	แรงดัน VDD
0 ถึง 3		8	2.0 ถึง 2.1	13	2.5 ถึง 2.6	18	3.0 ถึง 3.1	23	3.5 ถึง 3.6
4	1.6 ถึง 1.7	9	2.1 ถึง 2.2	14	2.6 ถึง 2.7	19	3.1 ถึง 3.2	24 ถึง 30	ส้ารองไว้
5	1.7 ถึง 1.8	10	2.2 ถึง 2.3	15	2.7 ถึง 2.8	20	3.2 ถึง 3.3	31	บิตแสดงสถานะการจ่ายไฟเลี้ยงให้แก่การ์ด
6	1.8 ถึง 1.9	11	2.3 ถึง 2.4	16	2.8 ถึง 2.9	21	3.3 ถึง 3.4		
7	1.9 ถึง 2.0	12	2.4 ถึง 2.5	17	2.9 ถึง 3.0	22	3.4 ถึง 3.5		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 รายละเอียดของรีจิสเตอร์ CID

ตำแหน่งบิต	ชื่อพารามิเตอร์	ชนิดข้อมูล	ขนาด	รายละเอียด	ค่าของ CID
127 : 120	รหัสผู้ผลิต (MID : Manufacturer ID)	ไบนารี	8 บิต	รหัสผู้ผลิต SD การ์ด ได้รับการกำหนดจาก SD card Association (SDA)	0x03
119 : 104	รหัสการใช้งาน (OID : OEM/Application ID)	เฮกซ์	16 บิต	รหัสนี้ครอบคลุมถึงผลิตภัณฑ์ที่นำไปใช้งานด้วยโดยต้องได้รับการกำหนดจาก 3 บริษัทผู้ก่อตั้ง SDA หรือบริษัท 3C	SD รหัสเฮกซ์ที่ 0x53 และ 0x44
103 : 64	ชื่อผลิตภัณฑ์ (PNM : Product Name)	เฮกซ์	40 บิต	รวมด้วยข้อมูลรหัสเฮกซ์ 5 หลัก	SD128, SD064, SD032, SD016, SD008
63 : 56	รุ่นของผลิตภัณฑ์ (PRV : Product Revision)	BCD	8 บิต	รวมด้วยข้อมูลรหัส BCD 2 หลัก โดย 4 ไบนารีคือ เลขหลักหน่วย และ 4 บิตล่างคือเลขหลักสิบ โดยจะนำคีย์จุดทศนิยม	รุ่นของผลิตภัณฑ์
55 : 24	เลขรหัสอนุกรมประจำตัว (PSN : Serial Number)	ไบนารี	32 บิต	รวมด้วยข้อมูลเลขฐานสอง 32 บิตแบบไม่คิดเครื่องหมาย	เลขรหัสอนุกรม ประจำตัว
23 : 20	สำรองไว้		4 บิต		
19 : 8	รหัสวันผลิต (MDT : Manufacturer Date Code)	BCD	12 บิต	รหัสเดือนปีที่ผลิต 12 หลักหน้าคือปี เริ่มตั้งแต่ 00 หมายถึง ค.ศ. 2000 และหลักหลังคือเดือน โดย 7 คือ มกราคม	รหัสเดือนปีที่ผลิต 0x014 : เมษายน 2004
7 : 1	รหัสตรวจสอบ CRC7 (CRC7 checksum)	ไบนารี	7 บิต	คำนวณจาก CRC(6...0) = เศษกส่วนหารพหุนาม $(P(x)=x^7+G(x))$ โดยที่ $M(x) = (บิตที่ 1 \times x^{11}) + (บิตที่ 2 \times x^{10}) + \dots + (บิตสุดท้าย \times x^1)$ และ $G(x) = x^6 + x^5 + 1$	CRC7
0	ไม่ใช้งาน		1 บิต		

รีจิสเตอร์ CID (Card Identification Register)

เป็นรีจิสเตอร์ที่มีความยาว 16 ไบต์ ใช้ในการเก็บข้อมูลเฉพาะของ SD การ์ด ซึ่งกำหนดมาจากผู้ผลิต ดังแสดงในตารางที่ 7 ผู้ใช้งาน ไม่สามารถทำการเปลี่ยนแปลงได้ โดยค่าและความหมายของข้อมูลในรีจิสเตอร์ CID ใน SD การ์ดจะแตกต่างจาก MMC

รีจิสเตอร์ CSD (Card Specific Data)

เป็นรีจิสเตอร์ขนาด 16 ไบต์ (128 บิต) ที่ใช้เก็บข้อมูลคุณสมบัติเฉพาะของ SD การ์ด ซึ่งมีรายละเอียดค่อนข้างมากเพราะในรีจิสเตอร์นี้บรรจุข้อมูลเกี่ยวกับความจุ, อัตราเร็วในการถ่ายถอดข้อมูล, แรงดันและกระแสไฟฟ้าในขณะที่อ่านและเขียนข้อมูล, รูปแบบของไฟล์, การป้องกันข้อมูล, การลบ และข้อมูลเกี่ยวกับการเขียนข้อมูลลงใน SD การ์ด สำหรับในการทดลองนี้เลือกใช้ 2 ข้อมูลคือ C_SIZE (บิต 73:62) และ C_SIZE_MUL (บิต 49:47) เพื่อนำมาคำนวณหาความจุของ SD การ์ดที่ติดต่อด้วย

ส่วนข้อมูลอื่น ๆ เพิ่มเติมของรีจิสเตอร์ตัวนี้สามารถอ่านได้จากไฟล์ค้ำค่าซีตของ SD การ์ด ในซีดีรอมที่จัดมาพร้อมกับบอร์ด JX-2148

รีจิสเตอร์ RCA (Relative Card Address)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าแอดเดรสของหน่วยความจำแบบสัมพันธ์ ซึ่งทาง โฮสต์ (หมายถึง คอมพิวเตอร์หรือไมโครคอนโทรลเลอร์) สามารถเลือกกำหนดได้ อย่างไรก็ตาม หากเลือกการติดต่อกับ SD การ์ดแบบ SPI จะไม่สามารถติดต่อกับรีจิสเตอร์ตัวนี้ได้

รีจิสเตอร์ SCR (SD Configuration Register)

เป็นรีจิสเตอร์ขนาด 64 บิต ที่ใช้เก็บค่าคุณสมบัติพิเศษของ SD การ์ด ที่เพิ่มเติม นอกเหนือไปจากที่เก็บในรีจิสเตอร์ CSD ซึ่งข้อมูลทั้งหมดนี้ได้รับการกำหนดมาจากผู้ผลิตเช่นกัน มีทั้งสิ้น 5 ข้อมูล คือ ข้อมูลเวอร์ชันของ SCR (บิต 63:60 รวม 4 บิต), ข้อมูลเวอร์ชันของคุณสมบัติทางกายภาพของ SD การ์ด (บิต 59:56 รวม 4 บิต ใช้จริงบิตเดียว), ข้อมูลสถานะของข้อมูลหลังจากการลบ (1 บิตคือบิต 55), ข้อมูลกำหนดระดับการป้องกัน (บิต 54:52 รวม 3 บิต), ข้อมูลแจ้งการรองรับขนาดของข้อมูลที่ทำการถ่ายทอค์ได้ของ SD การ์ด (บิต 51:48 รวม 4 บิต) รวมเป็น 16 บิต ที่เหลือสำรองไว้อีก 16 บิต (บิต 47:32) และสำรองสำหรับใช้เฉพาะผู้ผลิตอีก 32 บิต (บิต 31:0)

รีจิสเตอร์ DSR (Driver Stage Register)

เป็นรีจิสเตอร์ขนาด 16 บิต สำหรับเก็บค่าคุณสมบัติของไดรเวอร์ทางเอาต์พุตของ SDIO การ์ดจะมีความแตกต่างกันไปในอุปกรณ์เอาต์พุตแต่ละตัว

คั้งนั้นรีจิสเตอร์หลัก ๆ ที่ใช้จะมี 3 ถึง 4 ตัวคือ OCR, CID, CSD และ RCA สำหรับการทดสอบเบื้องต้นจะใช้เพียง 2 ตัวคือ CID และ CSD

รีจิสเตอร์แสดงสถานะการทำงานของ SD การ์ด

มี 2 ตัว คือ Card Status และ SD_Status โดย

Card Status มีขนาด 32 บิต ใช้แสดงสถานะการทำงานปกติ มีการทำงานเหมือนกับของ MMC การ์ด

SD_Status มีขนาด 512 บิต สามารถแสดงสถานะการทำงานพิเศษที่เพิ่มเติมไปจาก Card Status โดยข้อมูลสถานะจะถูกส่งลงไปบนสายสัญญาณ DAT พร้อมกับรหัสตรวจสอบ CRC บิต

ตารางที่ 3.6 รูปแบบข้อมูลทั้ง 48 บิตของคำสั่งของ SD การ์ด

ตำแหน่งบิต	47	46	45 : 40	39 : 8	7 : 1	0
ขนาด (บิต)	1	1	6	32	6	1
ค่าข้อมูล	'0'	'1'	X	X	X	'1'
คำอธิบาย	บิต เริ่มต้น	บิตส่ง	ตัวชี้ คำสั่ง	อะกิวเมนต์	CRC7	บิตหยุด

3.1.7 คำสั่งของ SD การ์ด

มีด้วยกัน 4 กลุ่มคือ

- (1) คำสั่งกระจายแบบไม่ต้องตอบสนอง (broadcast commands : bc)
- (2) คำสั่งกระจายแบบต้องตอบสนอง (broadcast commands with response : bcr)
- (3) คำสั่งระบุแอดเดรสจุดต่อจุดแบบ ไม่มีการถ่ายทอดข้อมูล (addressed commands : ac)
- (4) คำสั่งระบุแอดเดรสจุดต่อจุดแบบมีการถ่ายทอดข้อมูล (addressed commands

data transfer on DAT : actc)

คำสั่งทั้งหมดจะต้องส่งผ่านขาสัญญาณ CMD เข้าไปใน SD การ์ด โดยเริ่มจากบิตซ้ายสุดของขบวนข้อมูลคำสั่ง

คำสั่งของ SD การ์ดมีขนาด 48 บิต และต้องใช้เวลาในการส่งผ่าน 2.4 ไมโครวินาที ที่ความถี่สัญญาณนาฬิกา 20MHz โดยมีรูปแบบดังแสดงในตารางที่ 10

คลาสของคำสั่ง

คำสั่งของ SD การ์ดจัดแบ่งเป็นกลุ่ม เรียกว่า คลาส (class) ถึงปัจจุบันมีทั้งสิ้น 11 คลาส ดังนี้

- คลาส 0 เป็นกลุ่มของคำสั่งเบื้องต้น
- คลาส 1, 3, 9, 10 และ 11 สำรองไว้
- คลาส 2 เป็นกลุ่มของคำสั่งอ่านบล็อกข้อมูล
- คลาส 4 เป็นกลุ่มของคำสั่งเขียนบล็อกข้อมูล
- คลาส 5 เป็นกลุ่มคำสั่งลบข้อมูล
- คลาส 6 เป็นกลุ่มคำสั่งป้องกันการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส 7 เป็นกลุ่มลือกการ์ด
 คลาส 8 เป็นกลุ่มคำสั่งประยุกต์เพิ่มเติม สำหรับ SDIO การ์ด (ไม่อธิบายในที่นี้)
 โดยในคลาส 0 ถึง 7 จะตรงกับคลาสคำสั่งของ MMC การ์ด

รูปแบบของข้อมูลตอบสนอง (Response)

ในคำสั่งของ SD การ์ดที่มีความต้องการการตอบสนองจาก SD การ์ด จะมีรูปแบบของข้อมูลที่ตอบสนองกลับมาแตกต่างกัน มีทั้งสิ้น 4 รูปแบบมาตรฐานคือ R1, R1b, R2 และ R3 ซึ่งตรงกับ MMC การ์ด สำหรับ SD การ์ดยังมีรูปแบบของข้อมูลตอบสนองเพิ่มอีก 1 แบบคือ R6

(ก) รูปแบบของข้อมูลตอบสนองแบบ R1

เป็นรูปแบบมาตรฐาน มีขนาดของข้อมูล 48 บิต ดังนี้

ตำแหน่งบิต	47	46	45 : 40	39 : 8	7 : 1	0
ขนาด	1 บิต	1 บิต	6 บิต	32 บิต	7 บิต	1 บิต
ค่า	'0'	'0'	X	X	X	'1'
คำอธิบาย	บิตเริ่มต้น	บิตส่งข้อมูล	ค่านีคำสั่ง	ข้อมูลสถานะของการ์ด	CRC7	บิตปิดท้าย

(ข) รูปแบบของข้อมูลตอบสนองแบบ R1b

เป็นรูปแบบมาตรฐานที่เหมือนกับ R1 แต่จะมีการส่งสัญญาณ “ไม่ว่าง – Busy” ลงบนสายสัญญาณข้อมูลเพิ่มเติม

(ค) รูปแบบของข้อมูลตอบสนองแบบ R2

ใช้กับรีจิสเตอร์ CID และ CSD มีขนาดของข้อมูล 136 บิตดังนี้

ตำแหน่งบิต	135	134	133 : 128	127 : 1	0
ขนาด	1 บิต	1 บิต	6 บิต	32 บิต	1 บิต
ค่า	'0'	'0'	'111111'	X	'1'
คำอธิบาย	บิตเริ่มต้น	บิตส่งข้อมูล	สำรองไว้	ข้อมูลของรีจิสเตอร์ CID หรือ CSD รวมกับรหัสตรวจสอบ CRC7	บิตปิดท้าย

ค่าของรีจิสเตอร์ CID จะตอบสนองกับคำสั่ง CMD2 และ CMD10 ในขณะที่ค่าของรีจิสเตอร์ CSD จะตอบสนองกับคำสั่ง CMD9 โดยข้อมูลที่สนใจคือ ข้อมูลขนาด 127 บิต (ในบิต 1 ถึง 127)

(ง) รูปแบบของข้อมูลตอบสนองแบบ R3

ใช้กับรีจิสเตอร์ OCR มีขนาดของข้อมูล 48 บิต แสดงในตารางต่อไปนี้

ตำแหน่งบิต	47	46	45 : 40	39 : 8	7 : 1	0
ขนาด	1 บิต	1 บิต	6 บิต	32 บิต	7 บิต	1 บิต
ค่า	'0'	'0'	'111111'	X	'111111'	'1'
คำอธิบาย	บิตเริ่มต้น	บิตส่งข้อมูล	สำรองไว้	ข้อมูลสำหรับรีจิสเตอร์ OCR	สำรองไว้	บิตปิดท้าย

ค่าของรีจิสเตอร์ OCR จะตอบสนองกับคำสั่ง ACMD41

(จ) รูปแบบของข้อมูลตอบสนองแบบ R6

ใช้กับรีจิสเตอร์ RCA มีขนาดของข้อมูล 48 บิต แสดงในตารางต่อไปนี้

ตำแหน่งบิต	47	46	45 : 40	39 : 8	7 : 1	0	
ขนาด	1 บิต	1 บิต	6 บิต	16 บิต	16 บิต	7 บิต	1 บิต
ค่า	'0'	'0'	X	X	X	x	'1'
คำอธิบาย	บิตเริ่มต้น	บิตส่งข้อมูล	ดัชนีคำสั่ง	ข้อมูลใหม่ในบิต 31 : 16 ของรีจิสเตอร์ RCA	ข้อมูลของรีจิสเตอร์ Card Status บิต 23, 22, 19 และ 12 : 0	CRC7	บิตปิดท้าย

3.1.8 กระบวนการอ่าน-เขียน SD การ์ด

SD การ์ดมีกระบวนการอ่าน-เขียนข้อมูล 2 โหมด ดังแสดงในรูปที่ 5 โดยมีอัตราการถ่ายเทข้อมูล 25 เมกะบิตต่อวินาทีในกรณีใช้สายข้อมูลเส้นเดียว (ติดต่อแบบบัส SPI) และ 100 เมกะบิตต่อวินาทีในกรณีใช้สายข้อมูล 4 เส้น (ติดต่อแบบบัส SD)

โหมดบล็อกข้อมูลเดี่ยว (single block mode)

ในโหมดนี้โฮสต์ (host) หรือไมโครคอนโทรลเลอร์จะอ่านหรือเขียนข้อมูล 1 บล็อก โดยการส่งแต่ละบล็อกข้อมูลจะได้รับการป้องกันด้วยรหัส CRC 16 บิต เพื่อให้การรับข้อมูลมีความถูกต้องในการอ่านข้อมูลออกจาก SD การ์ดจะถูกจำกัดขนาดของบล็อกข้อมูลไว้ไม่เกิน 512 ไบต์ และต้องอยู่ในเซกเตอร์เดียวกัน ในการเขียนข้อมูลในแต่ละบล็อกต้องมีการกำหนดขนาดของเซกเตอร์และแอดเดรสเริ่มต้นมาด้วยเสมอ

โหมดหลายบล็อก (multiple block mode)

การทำงานในโหมดนี้จะคล้ายกับโหมดบล็อกข้อมูลเดี่ยวแต่โฮสต์หรือไมโครคอนโทรลเลอร์สามารถอ่านหรือเขียนข้อมูลได้หลายบล็อก โดยจะต้องเป็นบล็อกที่ติดกัน นอกจากนั้นยังสามารถรองรับการเขียนข้อมูลข้ามเซกเตอร์ได้ (ดูรูปที่ 5 ประกอบ)

3.2) การติดต่อกับ SD การ์ด

โฮสต์หรือคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์สามารถติดต่อกับ SD การ์ดได้ 2 วิธีคือผ่านบัส SD และบัส SPI ใช้สายสัญญาณที่แตกต่างกันดังแสดงรายละเอียดในตารางที่ 19

อย่างไรก็ตามในเอกสารนี้จะอธิบายเน้นไปที่การติดต่อผ่านบัส SPI เป็นหลัก

การติดต่อ SD การ์ดผ่านบัส SD

มีไลออะแกรมการติดต่อตามรูปที่ 6 ใช้สายสัญญาณ 6 เส้น และสายพลังงาน 3 เส้น ดังนี้

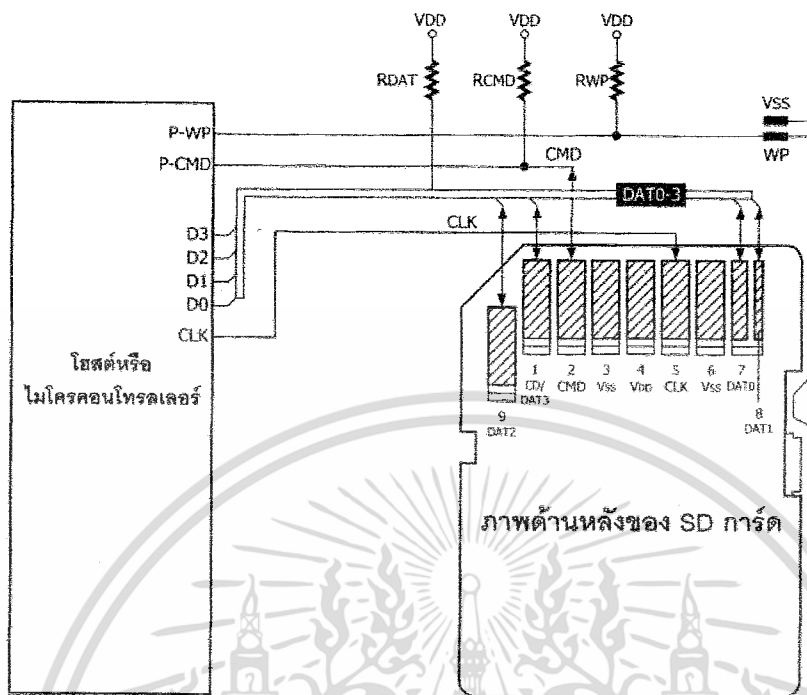
CMD : สายสัญญาณคำสั่ง เป็นสัญญาณ 2 ทิศทางติดต่อกันระหว่างโฮสต์กับ SD การ์ด

DAT0 ถึง DAT3 : สายสัญญาณข้อมูล เป็นสัญญาณ 2 ทิศทางเพื่อถ่ายทอดข้อมูลระหว่างโฮสต์กับ SD การ์ด มีทั้งสิ้น 4 เส้น

CLK : สายสัญญาณนาฬิกา สัญญาณนี้จะส่งออกจากโฮสต์เพื่อกำหนดจังหวะการทำงาน

V_{DD} : สายไฟเลี้ยง

GND : สายกราวด์ (ปกติมี 2 เส้น)



รูปที่ 3.6 วงจรการเชื่อมต่อเบื้องต้นระหว่าง ไมโครคอนโทรลเลอร์กับ SD การ์ดผ่านระบบบัส SD

การติดต่อ SD การ์ดผ่านบัส SPI

มีไดอะแกรมแสดงการติดต่อตามรูปที่ 8 ใช้สายสัญญาณ 4 เส้น และสายพลังงาน 3 เส้น ดังนี้

CS : สายสัญญาณเลือกการ์ด สัญญาณนี้จะส่งออกจากโฮสต์เพื่อเลือกติดต่อกับ SD การ์ด

DO (DAT0) : สายสัญญาณข้อมูลออกเป็นสัญญาณข้อมูลอนุกรมจากการ์ดส่งไปยังโฮสต์

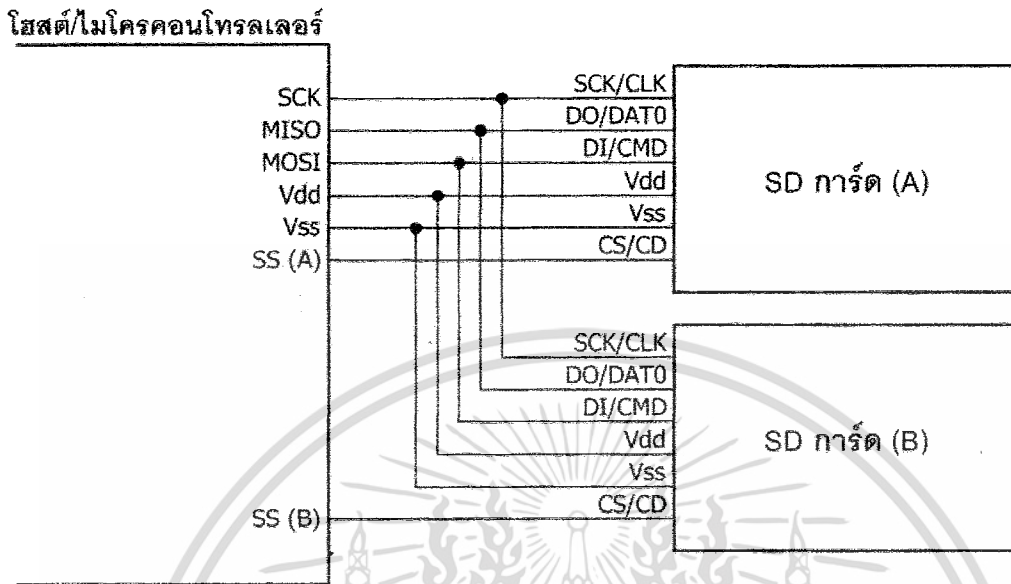
DI (CMD) : สายสัญญาณข้อมูลเข้า เป็นสัญญาณข้อมูลอนุกรมจากโฮสต์ส่งมายังการ์ด

SCK : สายสัญญาณนาฬิกา สัญญาณนี้จะส่งออกจากโฮสต์เพื่อกำหนดจังหวะการทำงาน

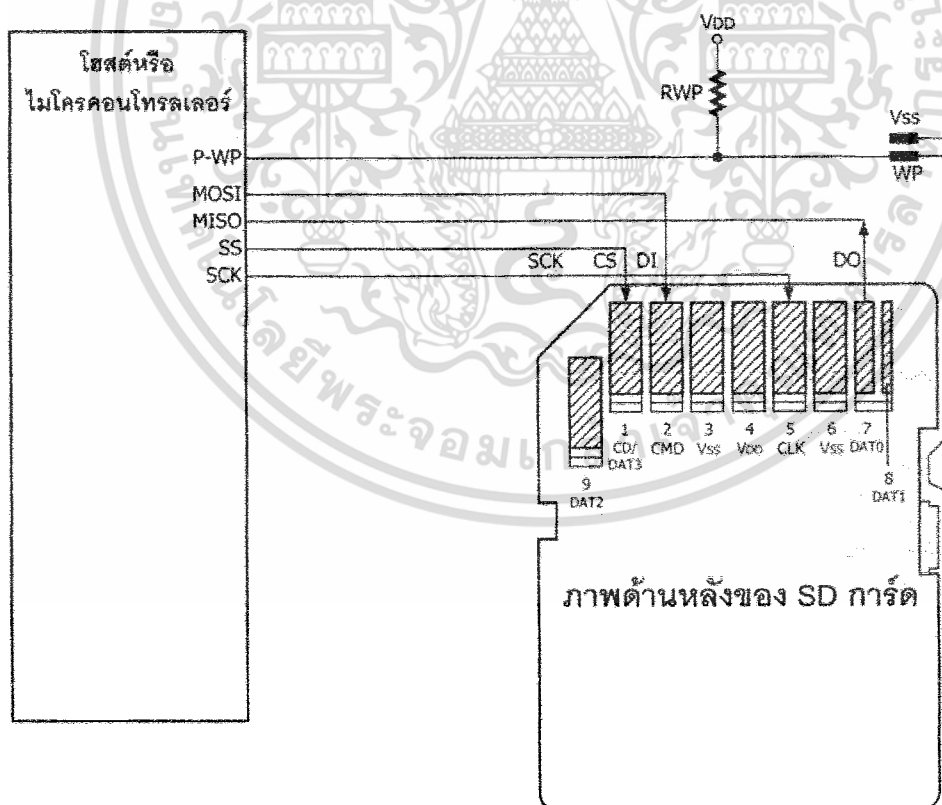
V_{DD} : สายไฟเลี้ยง

GND : สายกราวด์ (ปกติมี 2 เส้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงไต่แอมการติดต่อกับ SD การ์ดผ่านบัส SPI



รูปที่ 3.8 วงจรการเชื่อมต่อเบื้องต้นระหว่างไมโครคอนโทรลเลอร์กับ SD การ์ดผ่านระบบบัส SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 รูปแบบการติดต่อกับ SD การ์ดผ่านระบบบัส SPI

กลุ่มข้อมูลที่ใช้ในการติดต่อบนระบบบัส SPI หรือเรียกว่า SPI message ประกอบด้วย คำสั่ง (command), การตอบสนอง (response) และบล็อกข้อมูล (data-block) การสื่อสารระหว่าง โฮสต์หรือในที่นี้คือ ไมโครคอนโทรลเลอร์กับ SD การ์ดจะได้รับการกำหนดจังหวะจากโฮสต์ โดยโฮสต์จะเริ่มต้นการติดต่อสื่อสารด้วยการทำให้สายสัญญาณ CS เป็นลอจิก “0”

การตอบสนองของ SD การ์ดในการติดต่อผ่านระบบบัส SPI มีหลักเกณฑ์ดังนี้

- (1) SD การ์ดที่ถูกเลือกให้ติดต่อจะต้องตอบสนองต่อทุกคำสั่งเสมอ
- (2) ข้อมูลตอบสนองจะใช้ขนาด 8 หรือ 16 บิต
- (3) เมื่อ SD การ์ดประสบปัญหาในการกู้คืนข้อมูล มันจะแจ้งกลับด้วยข้อมูล

ตอบสนองผิดพลาด (error response) แทนที่จะเป็นบล็อกข้อมูล โดยมีค่าเวลาไทม์เอาต์ที่มากกว่า การติดต่อผ่านบัส SD

นอกจากนั้นในการตอบสนองคำสั่ง เมื่อทุก ๆ บล็อกข้อมูลถูกส่งไปยัง SD การ์ดในระหว่างการเขียน จะมีการตอบสนองด้วยสัญลักษณ์พิเศษ (special data response token) บล็อกของข้อมูลอาจมีขนาดใหญ่เท่ากับ 1 บล็อกข้อมูลปกติ หรือเล็กเพียง 1 ไบต์ ก็ได้

3.2.2 การเลือกโหมด

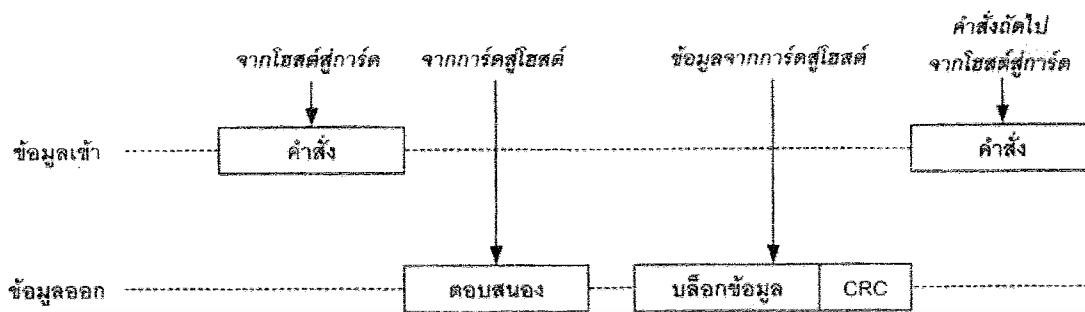
โดยปกติแล้ว SD การ์ดจะถูกกำหนดให้ใช้การติดต่อแบบบัส SD เป็นหลัก แต่สามารถเปลี่ยนโหมดการติดต่อเป็นบัส SPI ได้เมื่อ สายสัญญาณ CS เป็นลอจิก “0” ในระหว่างการรับคำสั่ง รีเซต (CMD0) และเมื่อเข้าสู่โหมด SPI ได้ จะมีการส่งรหัสตอบสนอง R1 กลับมายังโฮสต์

เมื่อ SD การ์ดทำงานในโหมด SPI แล้ว คำสั่งที่เกี่ยวข้องกับ โครงสร้างหรือโปรโตคอลจะไม่มี การตรวจสอบรหัส CRC ส่วนในคำสั่งอื่น ๆ สามารถที่เลือกได้ว่าจะให้มีการป้องกันตัวด้วยการ ตรวจสอบบิต CRC หรือไม่ โดยโฮสต์สามารถกำหนดได้โดยใช้คำสั่ง CRC_ON_OFF หรือ CMD59 ถ้าหากเลือกที่จะไม่มีการตรวจสอบ CRC แล้ว คุณภาพของสายสัญญาณและระดับ สัญญาณที่ใช้ในการติดต่อจะต้องดีและมีเสถียรภาพเพียงพอเพื่อให้การถ่ายถอดข้อมูลนั้นมีความ ถูกต้อง

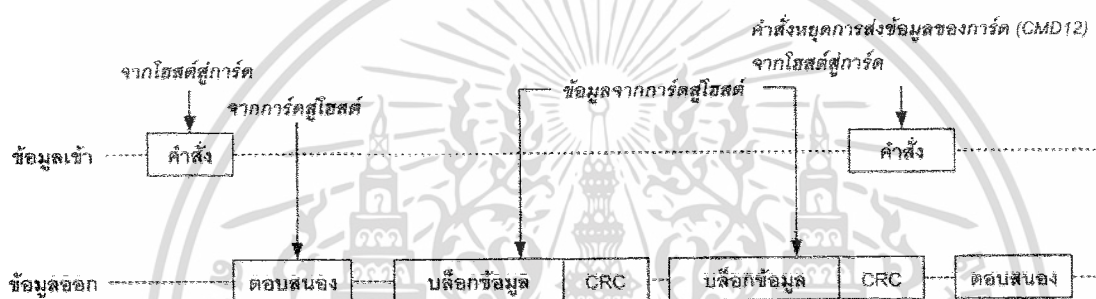
3.2.3 การอ่านข้อมูลในโหมด SPI

การอ่านข้อมูลจาก SD การ์ดในโหมดการติดต่อแบบ SPI นี้ สามารถอ่านได้ทั้งแบบบล็อก เดี่ยวและหลายบล็อก คำสั่งที่ใช้คือ CMD17 สำหรับบล็อกเดี่ยว และ CMD18 สำหรับหลายบล็อก เมื่อ SD การ์ดได้รับคำสั่งร้องขอเพื่ออ่านข้อมูลแล้ว มันจะส่งรหัสตอบสนองต่อด้วยบล็อกข้อมูลที่มีความ ยามตามที่กำหนดจากคำสั่ง CMD16 (SET_BLOCK_LENGTH) ปิดท้ายด้วยรหัส CRC ดัง แสดงในรูปที่ 3.9 อันเป็นกระบวนการอ่านข้อมูลแบบบล็อกเดี่ยวจาก SD การ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 กระบวนการอ่านข้อมูลแบบบล็อกเดียวจาก SD การ์ด



รูปที่ 3.10 กระบวนการอ่านข้อมูลแบบหลายบล็อกจาก SD การ์ด

สำหรับรหัส CRC 16 บิตนั้นจะถูกกำเนิดด้วยสมการ โพลีโนเมียลตามมาตรฐาน CCITT ดังนี้

$$x^{16} + x^{12} + x^5 + 1$$

ความยาวของบล็อกข้อมูลสูงสุดคือ 512 ไบต์ กำหนดโดย READ_BL_LEN หนึ่งในพารามิเตอร์ของรีจิสเตอร์ CSD ซึ่งมีขอบเขตในการกำหนดค่าได้ตั้งแต่ 1 จนถึงค่าของ READ_BL_LEN แต่โดยส่วนใหญ่แล้ว มักเลือกที่จะอ่านบล็อกข้อมูลความยาว 512 ไบต์ เพื่อความเร็วและต่อเนื่องในการทำงาน

แอดเดรสเริ่มต้นของการอ่านสามารถกำหนดที่แอดเดรสใด ๆ ก็ได้ภายในขอบเขตของ SD การ์ดใบนั้น ๆ ที่ทำการอ่าน

ในกรณีที่เกิดความผิดพลาดในกระบวนการอ่านข้อมูลเพื่อ SD การ์ดจะไม่ส่งข้อมูลใด ๆ ออกมาแต่จะส่งรหัสแจ้งความผิดพลาดกลับมายัง โฮสต์แทน และยกเลิกกระบวนการติดต่อเพื่ออ่านข้อมูลในทันที

ในกรณีที่มีการอ่านข้อมูลหลายบล็อก ข้อมูลในแต่ละบล็อกจะมีรหัส CRC16 บิตปิดท้ายเสมอเพื่อช่วยแยกข้อมูลให้ชัดเจน รวมทั้งช่วยในการตรวจสอบว่าข้อมูลที่ถูกอ่านออกไปถูกต้อง

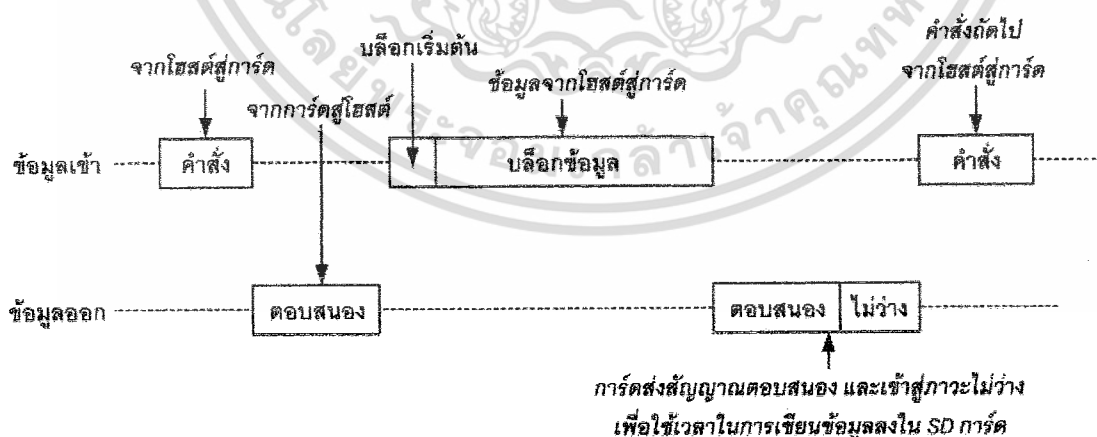
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมบูรณ์ดังแสดงในรูปที่ 11 และเมื่อต้องการหยุดอ่านข้อมูลต้องมีการส่งรหัสคำสั่งแจ้งแก่ SD การ์ดด้วย นั่นคือรหัสคำสั่ง CMD12 (Stop transmission command) หรือคำสั่งหยุดการส่งข้อมูลของ SD การ์ด

3.2.4 การเขียนข้อมูลในโหมด SPI

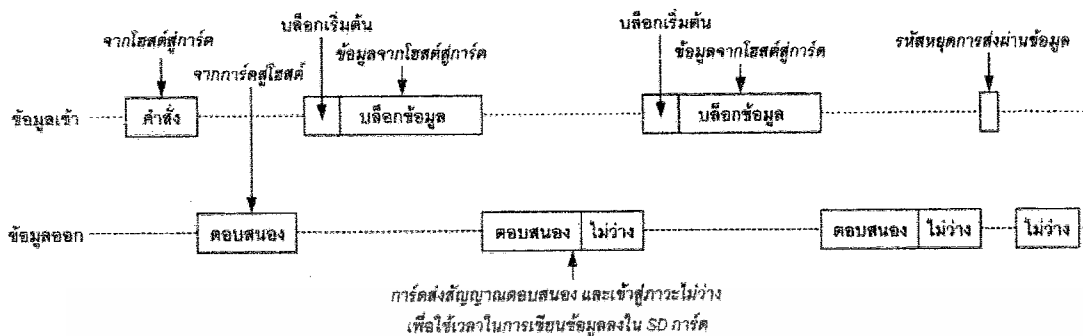
การเขียนข้อมูลไปยัง SD การ์ดในโหมดการติดต่อแบบ SPI นี้ สามารถเขียนได้ทั้งแบบบล็อกเดี่ยวและหลายบล็อก คำสั่งที่ใช้คือ CMD24 สำหรับบล็อกเดี่ยว และ CMD25 สำหรับหลายบล็อก เพื่อ SD การ์ดได้รับคำสั่งร้องขอเพื่อเขียนข้อมูลแล้ว มันจะส่งรหัสตอบสนอง จากนั้นจะรอบล็อกข้อมูลจากโฮสต์ ความยาวของบล็อกข้อมูลในกรณีเขียนนี้ต้องใช้ 512 ไบต์ เพื่อช่วยลดความผิดพลาดในการเขียนข้อมูลในครั้งถัดไป ในรูปที่ 12 แสดงจังหวะการเขียนข้อมูลลงใน SD การ์ดแบบบล็อกเดี่ยว

ในทุก ๆ บล็อกข้อมูลที่จะนำมาเขียนลงใน SD การ์ดต้องเริ่มต้นด้วยบล็อกเริ่มต้น (start block) มีความยาว 1 ไบต์ เมื่อข้อมูลถูกส่งมายัง SD การ์ด มันจะส่งสัญญาณตอบสนองตามด้วยสถานะไม่ว่าง (Busy) เพื่อใช้เวลาไปตรวจสอบว่า ในขณะที่นั้นการ์ดยังมีพื้นที่เหลือพอสำหรับเขียนข้อมูลใหม่ลงไปหรือไม่ ถ้ามีพอก็จะเขียนข้อมูล และปรับปรุงความจุของการ์ดที่เหลือหลังจากเขียนข้อมูลใหม่แล้ว ในจังหวะที่เกิดสถานะไม่ว่างนั้น ที่สายสัญญาณข้อมูลออก (หรือข้อมูลเอาต์พุต) จะได้รับการกำหนดให้เป็นลอจิกต่ำจนกว่าจะเสร็จสิ้นกระบวนการสถานะของสายสัญญาณจะกลับมาเป็นลอจิกสูง



รูปที่ 3.11 จังหวะการเขียนข้อมูลลงใน SD การ์ดแบบบล็อกเดี่ยว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 จังหวะการเขียนข้อมูลลงใน SD การ์ดแบบหลายบล็อก

หลังจากที่การเขียนข้อมูลเสร็จสิ้นลง โฮสต์ควรตรวจสอบผลการทำงานด้วย โดยส่งรหัสคำสั่ง CMD13 (SEND_STATUS) ไปยัง SD การ์ด โดยจะเน้นไปที่การตรวจสอบรหัส CRC และบิตแจ้งเตือนความผิดพลาดจากการเขียนข้อมูลลงในหน่วยความจำ

ในกรณีที่ต้องการเขียนข้อมูลแบบหลายบล็อก จะมีลักษณะการทำงานที่คล้ายกับแบบบล็อกเดียวหากแต่เมื่อจบบล็อกข้อมูลหนึ่ง แล้ว SD การ์ดส่งสัญญาณตอบสนอง จนกระทั่งเมื่อสถานะของบัสพร้อมก็สามารถส่งข้อมูลในบล็อกต่อไปที่ต้องการเขียนไปได้ทันที และเมื่อต้องการหยุดการเขียนโฮสต์ต้องส่งรหัสหยุดการส่ง (stop tran) ไปยัง SD การ์ด หลังจากที่ SD การ์ดเขียนบล็อกข้อมูลล่าสุดเรียบร้อยแล้ว หากเกิดความผิดพลาดในการเขียนขึ้น โฮสต์สามารถส่งรหัสคำสั่ง AMCD22 (SEND_NUM_WR_BLOCKS) เพื่อตรวจสอบจำนวนบล็อกข้อมูลที่เขียนข้อมูลได้อย่างสมบูรณ์ ในรูปที่ 13 แสดงจังหวะการเขียนข้อมูลลงใน SD การ์ดแบบหลายบล็อกผ่านระบบบัส SPI การรีเซตสัญญาณ CS ในขณะที่ทำการเขียนข้อมูลลงใน SD การ์ดนั้นกระบวนการเขียนข้อมูลจะไม่ถูกยกเลิก SD การ์ดยังสามารถเขียนข้อมูลต่อเนื่องไปได้

แต่ถ้าหากการรีเซตเกิดจากการส่งรหัสคำสั่ง CMD0 ในขณะที่กำลังเขียนข้อมูลลงใน SD การ์ดนั้น กระบวนการเขียนทั้งหมดจะถูกยกเลิกทันที นั่นหมายถึง ความเสียหายในระดับรูปแบบข้อมูลใน SD การ์ดถูกทำลาย ดังนั้นข้อจำกัดนี้คือสิ่งที่ผู้พัฒนาต้องให้ความสำคัญอย่างเต็มที่

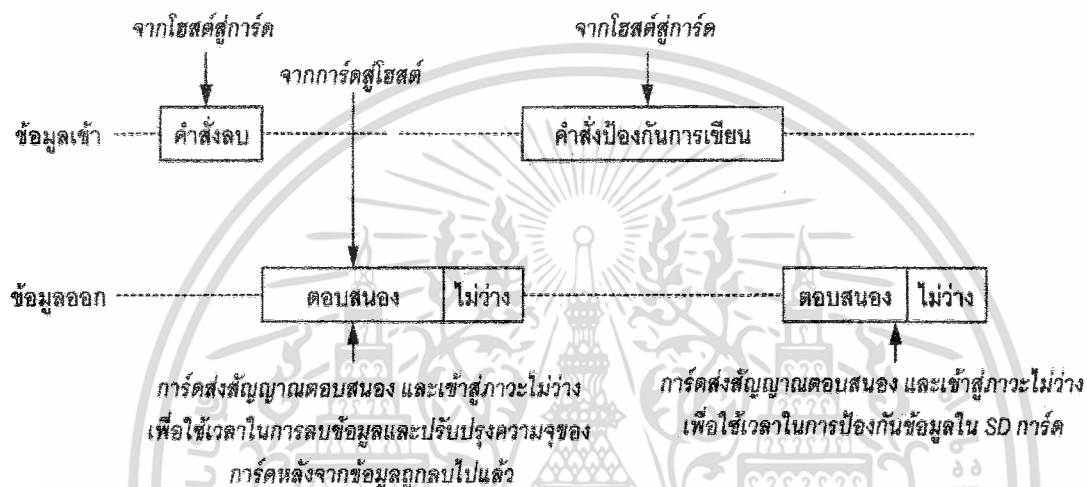
3.2.5 การลบและการจัดการป้องกันการเขียน

อีก 2 กระบวนการสำคัญที่ต้องมีในหน่วยความจำแบบเขียนใหม่ได้ทุกชนิดคือ การลบและการป้องกันการเขียน โดยทั้งสองกระบวนการนี้เป็นการติดต่อกับ SD การ์ดโดยไม่มีข้อมูลมาเกี่ยวข้อง ในรูปที่ 14 แสดงจังหวะการทำงานในกระบวนการทั้งสองนี้ โดยในทางปฏิบัติทั้งคำสั่งลบและป้องกันการเขียนคงไม่ได้ใช้ร่วมหรือต่อกัน เพียงแต่ในรูปนำมาอธิบายต่อเนื่อง เมื่อส่งคำสั่งลบข้อมูลไปยัง SD การ์ด จะมีการตอบสนองกลับมา แล้วเกิดสถานะไม่ว่าง เนื่องจาก SD การ์ด

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต้องดำเนินการลบข้อมูลออก แล้วปรับปรุงความจุของเนื้อที่ในการ์ดใหม่ เมื่อเรียบร้อยจึงคืนสถานะกลับมา

ทางด้าน การป้องกันการเขียนมีจังหวะการทำงานที่เหมือนกันคือ โฮสต์ส่งคำสั่งป้องกันการเขียนมา SD การ์ดก็จะตอบสนองแล้วสร้างสถานะไม่ว่างเพื่อดำเนินการตามคำสั่ง จนกระทั่งเรียบร้อย



รูปที่ 3.13 แสดงจังหวะการทำงานของทั้งคำสั่งลบและป้องกันการเขียน ในทางปฏิบัติไม่ได้ใช้ร่วมหรือต่อกัน เพียงนำมาอธิบายต่อเนื่องกันเท่านั้น

3.2.6 การอ่านรีจิสเตอร์ CID และ CSD

ในการติดต่อกับ SD การ์ดด้วยระบบบัส SPI จะต้องดำเนินการเข้าถึงเพื่ออ่านค่าของรีจิสเตอร์ CID และ CSD เอง โดยใช้กระบวนการเกี่ยวกับการอ่านข้อมูลปกติ เพียงแต่เลือกใช้คำสั่งที่อ่านข้อมูลอย่างถูกต้องเท่านั้น (ซึ่งจะได้กล่าวถึงคำสั่งนี้ในภายหลัง) เมื่อ SD การ์ดรับรู้คำสั่ง ก็จะส่งรหัสตอบสนองตามด้วยส่งบิตของข้อมูลขนาด 16 บิตต่อท้ายด้วยรหัส CRC 16 บิต

3.2.7 กระบวนการรีเซต

ในการติดต่อ SD การ์ด กระบวนการรีเซตเป็นเรื่องที่ต้องให้ความสนใจ การรีเซตจะเกิดขึ้นได้จาก 2 ทางคือ รีเซตทางฮาร์ดแวร์โดยกระบวนการเพาเวอร์-อนรีเซตหรือรีเซตเมื่อเริ่มต้นจ่ายไฟเลี้ยงให้แก่ SD การ์ด และการรีเซตทางซอฟต์แวร์ด้วยการส่งคำสั่งรีเซต CMD0 หลังจากการรีเซต SD การ์ดจะเข้าสู่สภาวะว่างหรือไอดีล (idle) ในช่วงเวลานี้ SD การ์ดสามารถตอบสนองได้เพียง 4 คำสั่งหลักคือ

CMD1	(SEND_OP_COND)
ACMD4	(SD_SEND_OP_COND)
CMD59	(CRC_ON_OFF)
CMD58	(READ_OCR)

โฮสต์ต้องทำการวนส่งรหัสคำสั่ง CMD1 อย่างต่อเนื่องจนกระทั่ง SD การ์ดอินิเชียลเรียบร้อยแล้ว โดยตรวจสอบจากระดับลอจิกของสายข้อมูล เปลี่ยนจากลอจิกสูงเป็นต่ำ ทั้งนี้เป็นเพราะรหัสคำสั่ง CMD1 ไม่มีการส่งค่ากลับมา แต่ยังมีอีกวิธีหนึ่งในการตรวจสอบความพร้อมของ SD การ์ดคือ ใช้รหัสคำสั่ง CMD58 เพื่ออ่านค่ารีจิสเตอร์ OCR เมื่อใดที่ SD การ์ดพร้อมทำงานก็จะสามารถอ่านค่าของรีจิสเตอร์ OCR ซึ่งรหัสคำสั่งนี้จะมีเฉพาะในการติดต่อผ่านบัส SPI เท่านั้น

3.2.8 การควบคุมสัญญาณนาฬิกา

ในการติดต่อ SD การ์ด ผ่านระบบบัส SPI สัญญาณนาฬิกามีความสำคัญมากเพราะเป็นตัวกำหนดจังหวะการทำงานทั้งหมด โดยโฮสต์เป็นอุปกรณ์ที่ทำหน้าที่กำเนิดสัญญาณนาฬิกาและสามารถเปลี่ยนแปลงความถี่ได้ตลอดเวลา และต้องมีค่าไม่เกินค่าความถี่ของการถ่ายทอดข้อมูลสูงสุดของ SD การ์ด

เมื่อกระบวนการในการติดต่อกับ SD การ์ดเสร็จสิ้นไปแล้วในช่วง โฮสต์จำเป็นต้องส่งสัญญาณนาฬิกาต่อเนื่องไปอีก 8 ลูก เพื่อรอให้ SD การ์ดทำงานตามคำสั่งที่กำหนดไปแล้วเสร็จอย่างสมบูรณ์จริง แม้ว่าในบางคำสั่งจะมีการตอบสนองหรือส่งสัญญาณตอบกลับมาจากโฮสต์ก็ตาม โดยสามารถสรุปได้ดังนี้

- (1) เมื่อโฮสต์ได้รับการตอบสนองจาก SD การ์ด หลังจากส่งคำสั่งไปให้โฮสต์ต้องส่งสัญญาณนาฬิกาไปอีก 8 ลูกหลังจากได้รับบิตสุดท้ายของการตอบสนองจาก SD การ์ด
- (2) ในกระบวนการอ่านข้อมูลจาก SD การ์ด โฮสต์ต้องส่งสัญญาณนาฬิกาไปอีก 8 ลูกหลังจากได้รับบิตสุดท้ายของบล็อกข้อมูล
- (3) ในกระบวนการเขียนข้อมูลไปยัง SD การ์ด โฮสต์ต้องส่งสัญญาณนาฬิกาไปอีก 8 ลูกหลังจากได้รับบิตสุดท้ายของรหัส CRC

3.2.9 ค่าเวลาไทม์เอาต์ของการอ่าน-เขียน-ลบข้อมูลของ SD การ์ด

ในทุกกระบวนการของการติดต่อกับ SD การ์ด จะมีพารามิเตอร์ตัวหนึ่งที่ควรทราบและต้องให้ความสำคัญคือ ค่าเวลาไทม์เอาต์ (time out) หรือค่าเวลาเกินกำหนดหลังจากที่ SD การ์ดได้รับคำสั่งแล้วไม่มีกาตอบสนองใดๆ โดยในแต่ละกระบวนการจะมีค่าเวลาไทม์เอาต์ที่แตกต่างกัน สรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) ค่าไทม์เอาต์ของการอ่านข้อมูลจาก SD การ์ด ควรมีค่ามากกว่า 100 เท่าของค่าเวลาเข้าถึงข้อมูลของ SD การ์ด (access time) หรือ 100 มิลลิวินาที

(2) ค่าไทม์เอาต์ของการเขียนและลบข้อมูลใน SD การ์ด ควรมีค่ามากกว่า 100 เท่าของค่าเวลาในการเขียนข้อมูลของ SD การ์ด หรือ 250 มิลลิวินาที

3.2.10 คำสั่งการติดต่อกับ SD การ์ดผ่านระบบบัส SPI

คำสั่งที่ใช้ในการติดต่อกับ SD การ์ดเมื่อทำงานผ่านระบบบัส SPI แบ่งออกเป็น 12 คลาส แต่มีการใช้งานจริง 7 คลาส รวม 60 รหัสคำสั่ง (CMD0 ถึง CMD59) ดังแสดงในตารางที่ 20 มีความยาวของแต่ละคำสั่งเท่ากับ 6 ไบต์ และมีรูปแบบดังนี้

ตารางที่ 3.8 สรุปคำสั่งที่ใช้ในการติดต่อกับ SD การ์ดเมื่อทำงานผ่านระบบบัส SPI โดยแบ่งตาม คลาส

คลาสรหัสคำสั่ง ของ SD การ์ด (CCC:Card CMD Class)	คำอธิบาย	รหัสคำสั่ง																											
		0	1	9	10	12	13	16	17	18	24	25	27	28	29	30	32	33	38	42	55	56	58	59					
คลาส 0	เป็นกลุ่มคำสั่งพื้นฐาน	+	+	+	+	+	+																						
คลาส 1	ไม่รองรับโหมด SPI																												
+	คำสั่งอ่านบล็อกข้อมูล							+	+	+																			
คลาส 3	ไม่รองรับโหมด SPI																												
คลาส 4	คำสั่งเขียนบล็อกข้อมูล										+	+	+																
คลาส 5	คำสั่งลบข้อมูล																	+	+	+									
คลาส 6	คำสั่งป้องกันการเขียน																		+	+	+								
คลาส 7	คำสั่งล๊อคการ์ด																										+		
คลาส 8	คำสั่งพิเศษเฉพาะ																											+	+
คลาส 9	ไม่รองรับโหมด SPI																												
คลาส 10-11	สำรองไว้																												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบนารีที่ 1			ไบนารีที่ 2 ถึง 5 (4 ไบนารี รวม 32 บิต)	ไบนารีที่ 6	
7	6	5 0	31	7	0
0	1	คำสั่ง	อะกิวเมนต์ของคำสั่ง	รหัส CRC	1

การส่งคำสั่งเริ่มต้นด้วยข้อมูลคำสั่งเป็น ไบนารีแรก ตามด้วยอาร์กิวเมนต์ของคำสั่งซึ่งมีขนาด 4 ไบนารี และปิดท้ายด้วยไบนารีของรหัส CRC โดยในการส่งคำสั่งไปยัง SD การ์ดแต่ละไบนารี จะเริ่มต้นด้วยบิต MSB หรือบิตนัยสำคัญสูงสุดก่อน

ข้อมูลของรหัส CRC7 บิต (CRC0-CRC6) กำหนดจากเศษของการหาร $(M(x) \cdot x^7) / G(x)$ โดยที่

$$M(x) = \text{ค่าของบิตเริ่มต้น} \cdot x^{39} + (\text{บิตของข้อมูลคำสั่งบิต MSB} \cdot x^{38}) + \dots + (\text{บิตสุดท้ายของอาร์กิวเมนต์ของคำสั่ง ก่อนไบนารี CRC} \cdot x^0)$$

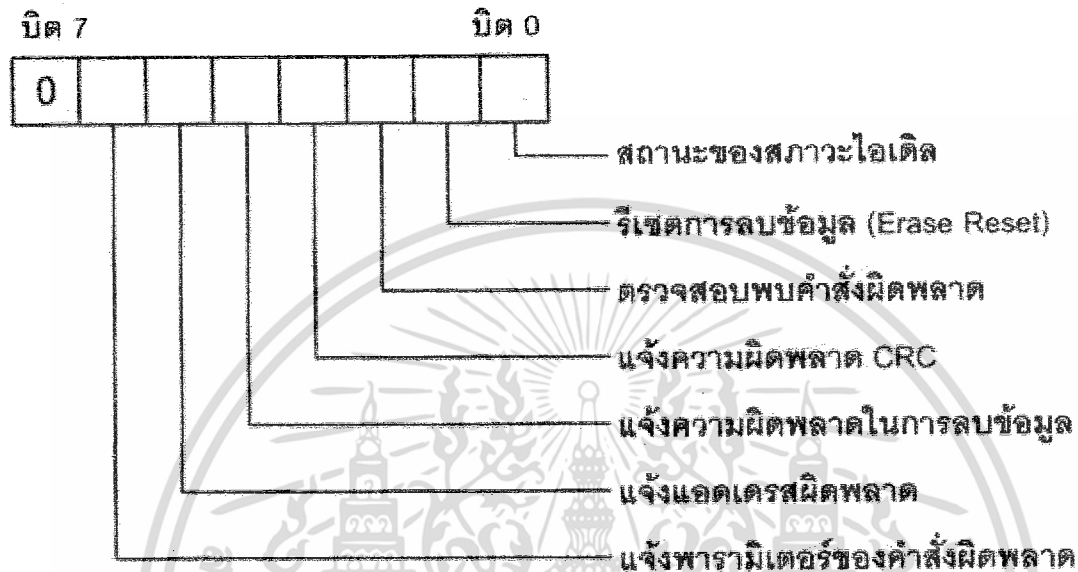
นอกจากนั้นในโหมดการติดต่อแบบ SPI ยังมีคำสั่งพิเศษเพิ่มเติมเช่นกัน โดยใช้รหัสคำสั่ง นำหน้าเป็น ACMD โดยแสดงไว้ในตารางที่ 25

3.2.11 รูปแบบของข้อมูลตอบสนอง (Response) เมื่อติดต่อ SD การ์ดแบบ SPI

ในคำสั่งของ SD การ์ดที่มีความต้องการตอบสนองจาก SD การ์ด จะมีรูปแบบของข้อมูลตอบสนองกลับมาแตกต่างกัน มีทั้งสิ้น 4 รูปแบบมาตรฐานคือ R1, R1b, R2 และ R3

(ก) รูปแบบของข้อมูลคอบสนองแบบ R1

เป็นรูปแบบมาตรฐาน โดยปกติใช้แจ้งการคอบสนองคำสั่งส่งข้อมูลของโฮสต์มายัง SD การ์ดหรือ SEND_STATUS มีขนาดของข้อมูล 8 บิต ดังนี้



ค่าเป็น "0" บิต 7 ซึ่งเป็นบิต MSB มีค่าเป็น "0" เสมอ ส่วนบิตสถานะอื่น ๆ เมื่อเกิดขึ้นจะมี

หรือเตรียมความพร้อม บิต 0 ใช้แจ้งว่า ขณะนี้ SD การ์ดอยู่ในสถานะไอเดิล และกำลังทำการอินิเชี่ยล

ก่อนเอ็กซิกิวต์คำสั่งลบข้อมูล บิต 1 ใช้แจ้งว่า ขณะนี้ SD การ์ดอยู่ในสถานะรีเซตการลบข้อมูล ซึ่งจะเกิดขึ้น

บิต 2 ใช้แจ้งว่า ขณะนี้ SD การ์ดได้รับคำสั่งที่ไม่ถูกต้อง

บิต 3 ใช้แจ้งว่า การตรวจสอบรหัส CRC ล้มเหลว

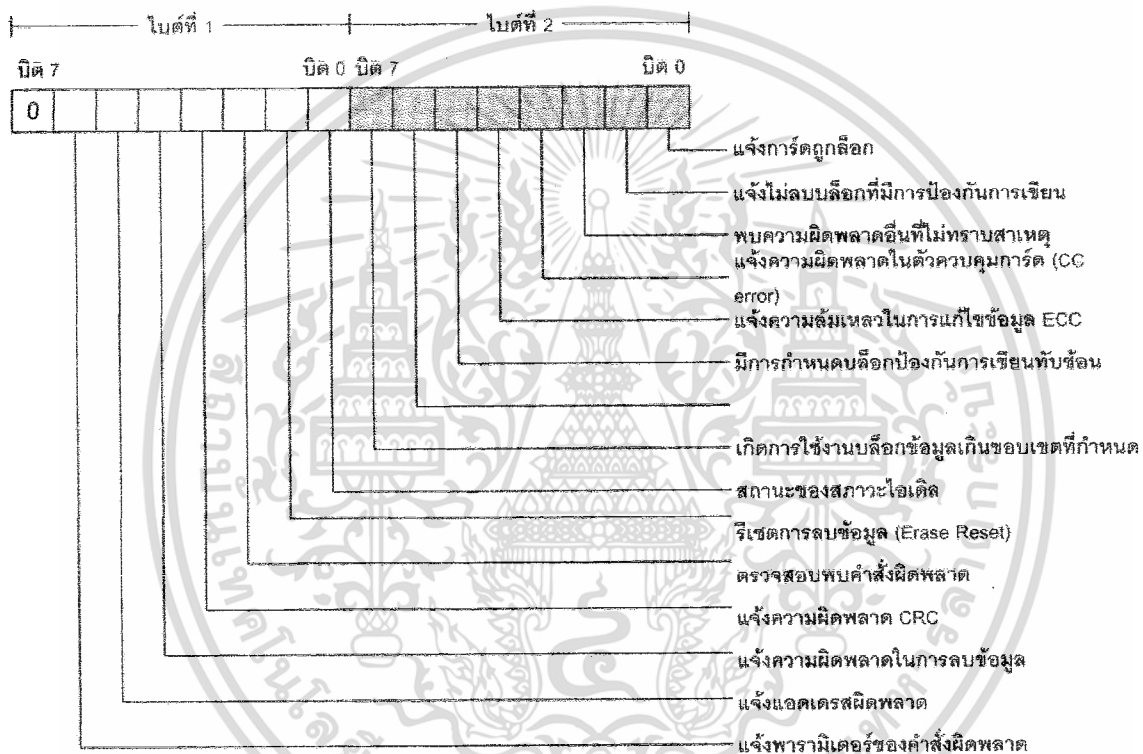
บิต 4 ใช้แจ้งว่า การลบข้อมูลใน SD การ์ดเกิดความผิดพลาด

บิต 5 ใช้แจ้งว่า มีค่าแอดเดรสที่ไม่ถูกต้องหรือตรงกับความยาวของบล็อกข้อมูลที่
ที่ใช้ในคำสั่งที่ส่งมา

บิต 6 ใช้แจ้งว่า มีการกำหนดค่าพารามิเตอร์ไม่ถูกต้องให้แก่คำสั่งที่ใช้ในการ
ติดต่อกับ SD การ์ด

(ข) รูปแบบของข้อมูลตอบสนองแบบ R1b

เป็นรูปแบบมาตรฐานที่เหมือนกับ R1 แต่จะมีการส่งสัญญาณ “ไม่ว่าง – Busy” ลงบนสายสัญญาณข้อมูลเพิ่มเติม โดยถ้าหากอยู่ในสถานะไม่ว่าง จะส่งสัญญาณลอจิก “0” ต่อท้ายข้อมูลตอบสนอง โดยความยาวของสัญญาณไม่ว่าง อาจยาวหนึ่งไบต์หรือมากกว่าก็ได้ขึ้นอยู่กับการทำงานที่เกี่ยวข้องในขณะนั้น



(ค) รูปแบบของข้อมูลตอบสนองแบบ R2

มีความยาว 2 ไบต์ โดยมีรายละเอียดของข้อมูลในแต่ละบิตดังนี้

โดยข้อมูลในไบต์แรกมีรูปแบบเหมือนกับข้อมูลตอบสนอง R1 ส่วนในไบต์ที่สองมีรายละเอียดเพิ่มเติมดังนี้

บิต 7 Out of Range : เกิดการใช้งานบล็อกรหัสเกินขอบเขตที่กำหนด

บิต 6 Erase Param : แจ้งความผิดพลาดเนื่องจากการลบข้อมูลในบล็อกรหัสข้อมูลที่มีอยู่จริง หรือไม่ได้เลือกบล็อกรหัสที่ต้องการลบ

บิต 5 Write protect violation : แจ้งความผิดพลาดเนื่องจากการกำหนดบล็อกรหัสข้อมูลที่มีการป้องกันการเขียนทับซ้อนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 4 Card ECC failed : แจ้งความผิดพลาดในการแก้ไขข้อมูล ECC

บิต 3 CC Error : แจ้งความผิดพลาดในตัวควบคุมการ์ด

บิต 2 Error : แจ้งพบความผิดพลาดอื่นที่ไม่ทราบสาเหตุ

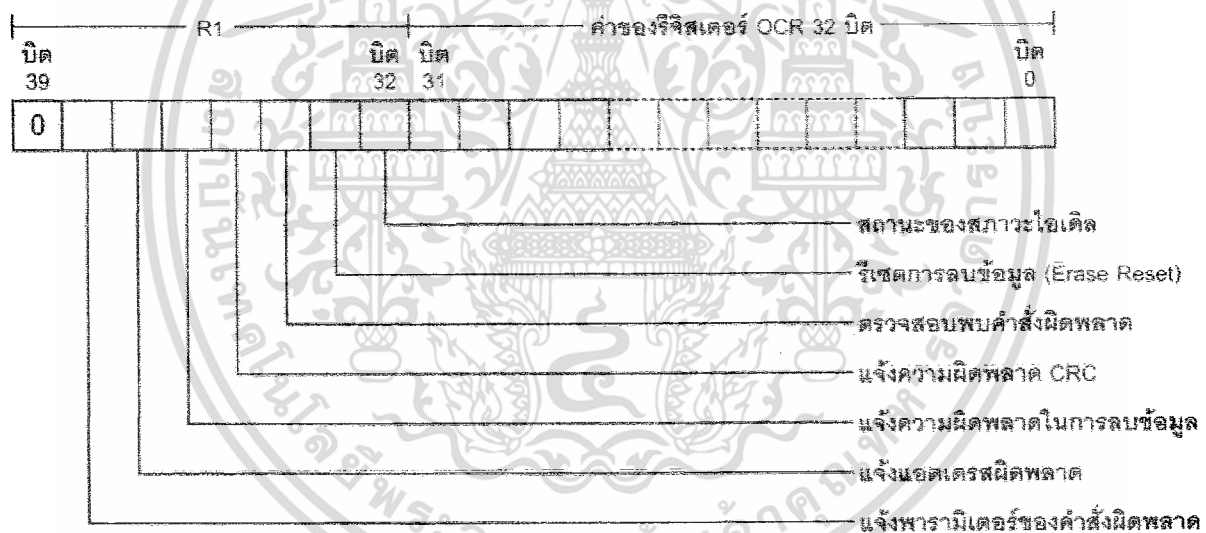
บิต 1 Write protect erase skip : แจ้งพบความผิดพลาดในกรณีที่แจ้งลบบล็อก

ข้อมูลที่มีการป้องกันการเขียน

บิต 0 Card is locked : แจ้ง SD การ์ดถูกล็อก (มีใช้ใน SD การ์ดของ SanDisk)

(ง) รูปแบบของข้อมูลตอบสนองแบบ R3

ใช้กับรีจิสเตอร์ OCR มีขนาดของข้อมูล 40 บิต โดยข้อมูล 8 บิตบนจะเป็นข้อมูลตอบสนอง R1 ส่วนอีก 32 บิตที่เหลือเป็นค่าของรีจิสเตอร์ OSR ข้อมูลตอบสนองแบบนี้จะเกิดขึ้นเมื่อ SD การ์ด ได้รับคำสั่ง READ_OCR



(จ) ข้อมูลตอบสนองเนื่องจากการเขียนบล็อกข้อมูล

ข้อมูลตอบสนองในลักษณะนี้จะเกิดขึ้นทุกครั้งที่มีการเขียนบล็อกข้อมูลลงใน SD การ์ด มีความยาว 1 ไบต์ โดยมีรูปแบบดังนี้

บิต 7							บิต 0
X	X	X	0				1

| บิตแจ้งสถานะ |

“010” - ได้รับข้อมูล

“101” - ปฏิเสธข้อมูลเนื่องจากความผิดพลาด CRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“110” - ปฏิเสธข้อมูลเนื่องจากการเขียนข้อมูลผิดพลาด

3.2.12 รูปแบบข้อมูลในการติดต่อกับ SD การ์ดผ่านบัส SPI

ในการกระทำคำสั่งอ่านและเขียนข้อมูลกับ SD การ์ด จะมีการถ่ายทอดและจัดกลุ่มข้อมูลให้เป็นสัดส่วนและรูปแบบ โดยทั้งการอ่านและเขียนข้อมูลจะเริ่มต้นการถ่ายทอดข้อมูลด้วยบิต MSB เสมอ

ข้อมูลที่มีการถ่ายทอดในระบบของ SD การ์ดมีขนาด 4 ถึง 515 ไบต์ โดยมีรูปแบบดังนี้

(ก) รูปแบบของข้อมูลในกรณีอ่าน-เขียนข้อมูลแบบบล็อกเดี่ยว และอ่านข้อมูลแบบหลายบล็อก

ข้อมูลไบต์แรก คือ บล็อกเริ่มต้น (Start Block) ต้องมีรูปแบบดังนี้

บิต 7								บิต 0
1	1	1	1	1	1	1	1	0

ข้อมูลไบต์ที่ 2 ถึง 513 (ขึ้นอยู่กับความยาวของบล็อกข้อมูล) เป็น ข้อมูลของผู้ใช้งาน

ข้อมูล 2 ไบต์สุดท้าย เป็นรหัสตรวจสอบ CRC 16 บิต

(ข) รูปแบบของข้อมูลในกรณีเขียนข้อมูลแบบหลายบล็อก

ข้อมูลไบต์แรกของแต่ละบล็อก จะเกิดขึ้นได้ 2 กรณีคือ

รูปแบบนี้
ถ้าเป็นการถ่ายทอดข้อมูล ข้อมูลไบต์แรกจะเป็นบล็อกเริ่มต้น มี

บิต 7								บิต 0
1	1	1	1	1	1	1	0	0

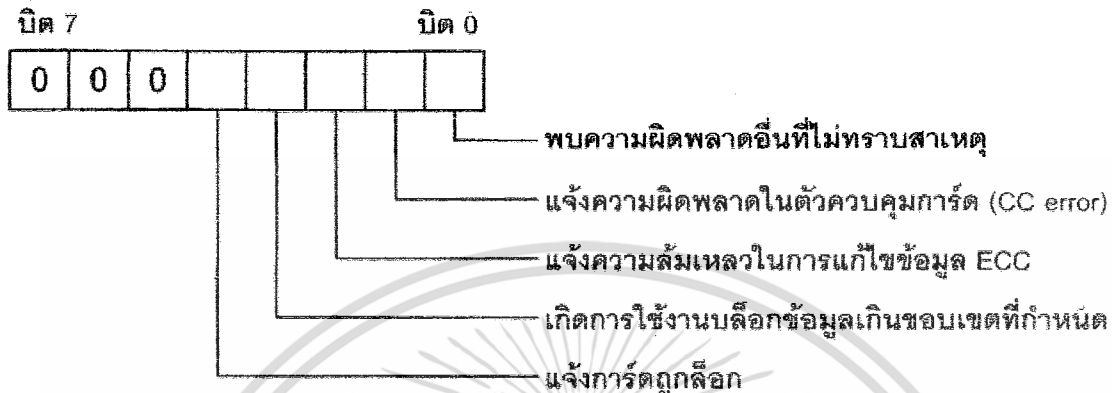
ถ้าเป็นการร้องขอให้หยุดถ่ายทอดข้อมูล ข้อมูลไบต์แรกจะเป็น บล็อกหยุดการถ่ายทอดข้อมูล (Stop Tran) มีรูปแบบดังนี้

บิต 7								บิต 0
1	1	1	1	1	1	1	0	1

อย่างไรก็ตาม รูปแบบข้อมูลของการร้องขอให้หยุดการถ่ายทอดข้อมูลในลักษณะนี้จะใช้เฉพาะกรณีหยุดเขียนข้อมูลแบบหลายบล็อกเท่านั้น ส่วนการหยุดอ่านข้อมูลแบบหลายบล็อกจะใช้คำสั่ง STOP_TRAN (รหัสคำสั่ง CMD12) แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อการอ่านข้อมูลจาก SD การ์ดล้มเหลว ทำให้ไม่สามารถอ่านข้อมูลได้ SD การ์ดจะส่งข้อมูลแจ้งความผิดพลาดหรือ Data error token ขนาด 1 ไบต์กลับมายังโฮสต์ มีรายละเอียดดังนี้



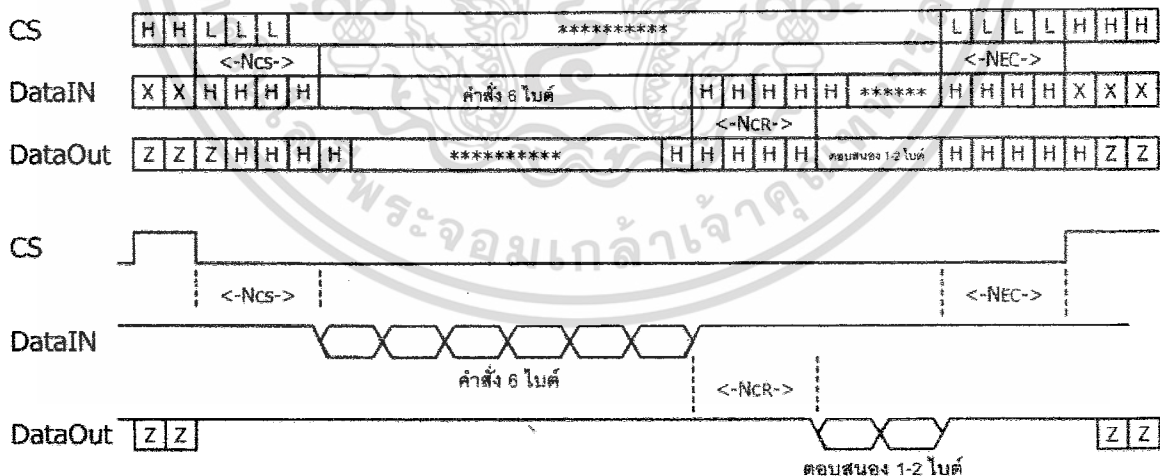
ไคอะแกรมเวลาของการติดต่อกับ SD การ์ดผ่านบัส SPI

เพื่อให้การทำความเข้าใจในการกระบวนการติดต่อกับ SD การ์ดง่ายขึ้น ไคอะแกรมเวลาเป็นสิ่งที่จำเป็นและช่วยได้อย่างมาก ซึ่งมีด้วยกัน 3 กลุ่มใหญ่ ๆ คือ

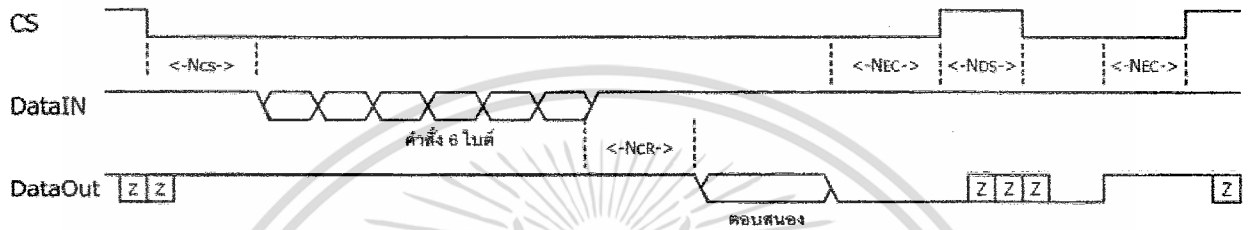
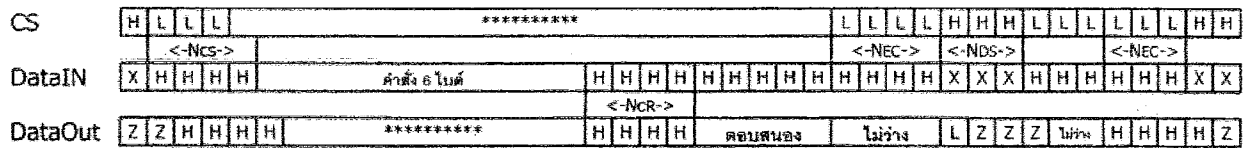
(1) ไคอะแกรมเวลาของการรับส่งคำสั่งและสัญญาณตอบสนอง มี 3 รูปแบบคือ

(1.1) โฮสต์ส่งคำสั่งมายัง SD การ์ด แล้วการ์ดพร้อมทำงาน มีไคอะแกรม

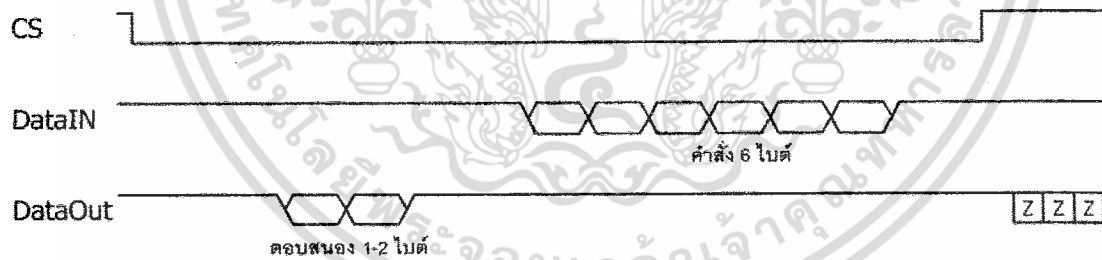
เวลาดังรูป

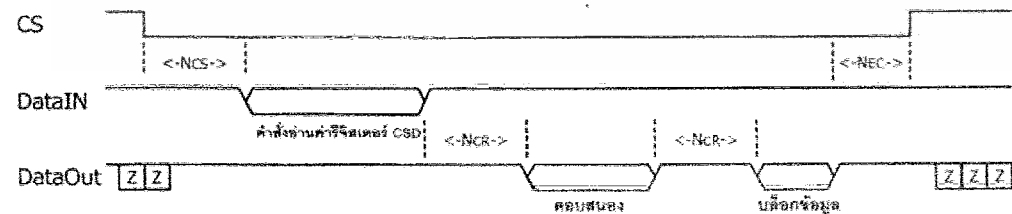
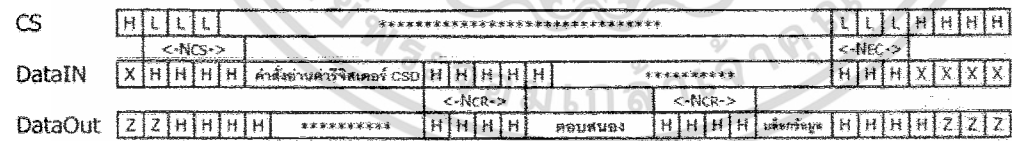
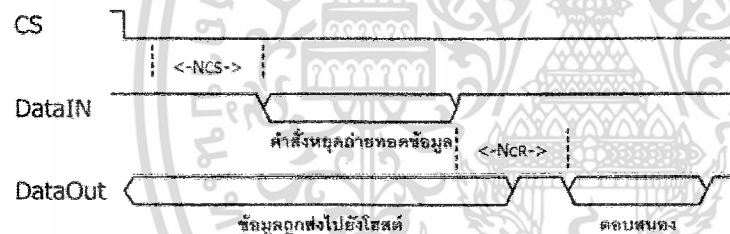
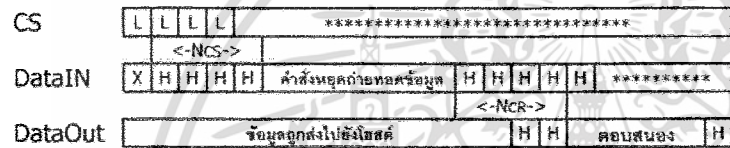
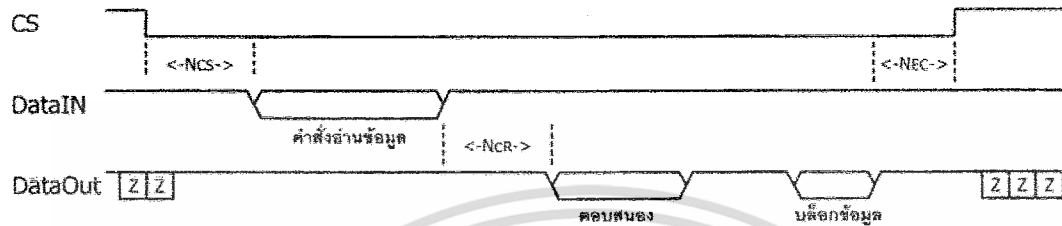
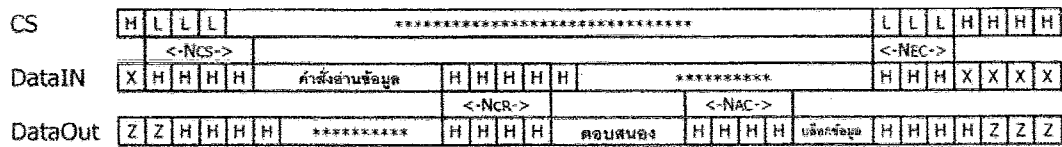


(1.2) โฮสต์ส่งคำสั่งมายัง SD การ์ด แล้วการ์ดไม่ว่าง มีไคอะแกรมเวลาดังรูป



(1.3) SD การ์ดส่งข้อมูลตอบสนองกลับมายังโฮสต์ มีไคอะแกรมเวลาดังรูป





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- N_{CS} มีค่าได้ตั้งแต่ 0 และไม่มีจำกัดค่าสูงสุด ขึ้นอยู่กับเวลาในการติดต่อกับ SD การ์ด
- N_{DS} มีค่าได้ตั้งแต่ 0 และไม่มีจำกัดค่าสูงสุด ขึ้นอยู่กับเวลาในการติดต่อกับ SD การ์ด
- N_{EC} มีค่าได้ตั้งแต่ 0 และไม่มีจำกัดค่าสูงสุด ขึ้นอยู่กับเวลาในการติดต่อกับ SD การ์ด
- N_{RC} มีค่าได้ตั้งแต่ 8 ไชเกิลสัญญาณนาฬิกาและไม่มีจำกัดค่าเวลาสูงสุด ขึ้นอยู่กับเวลาในการติดต่อกับ SD การ์ด
- N_{WR} มีค่าได้ตั้งแต่ 8 ไชเกิลสัญญาณนาฬิกาและไม่มีจำกัดค่าเวลาสูงสุด ขึ้นอยู่กับเวลาในการติดต่อกับ SD การ์ด

3.3) FAT (File Allocation Table)

3.3.1 ความหมายของ FAT

FAT ย่อมาจากคำว่า File Allocation Table ซึ่งคือตารางที่ใช้เก็บตำแหน่งของข้อมูลต่างๆ ที่อยู่บนฮาร์ดดิสก์ เพื่อใช้ในการจัดสรรและติดตามการใช้เนื้อที่ในฮาร์ดดิสก์

3.3.2 ประเภทของ FAT

FAT เป็นระบบไฟล์ที่ถูกกำหนดโดยซอฟต์แวร์ระบบปฏิบัติการ (operating system) จึงแตกต่างกันไปได้หลายแบบ FAT ที่เป็นที่รู้จัก ได้แก่

FAT (File Allocation Table)

เป็นระบบไฟล์มาตรฐานสำหรับ DOS และ Windows และด้วยการที่ FAT เป็นที่นิยมใช้อย่างกว้างขวาง จึงสามารถใช้ร่วมกับระบบปฏิบัติการอื่น เช่น Linux, OS/2 และระบบปฏิบัติการอื่นๆอีกด้วย

FAT32 (32-bit File Allocation System)

ระบบไฟล์แบบนี้จะอยู่ใน Windows95 OSR2 ในรุ่นที่มีการติดตั้งจากผู้ผลิต และ Windows98, FAT32 กำจัดข้อจำกัดของ FAT หลายประการออกไป แต่ระบบไฟล์นี้จะไม่สามารถใช้กับระบบปฏิบัติการอื่นนอกจาก Windows95 OSR2, Windows98

VFAT (Virtual File Allocation Table)

เป็นระบบไฟล์ FAT เวอร์ชันที่มีลักษณะเป็น Protected Mode ซึ่งจะถูกใช้โดย Windows 9x ระบบไฟล์นี้จะคล้ายๆกับ FAT ต่างกันตรงที่สามารถรับชื่อไฟล์ยาวๆได้

NTFS (NT Files System)

เป็นระบบไฟล์ที่ถูกออกแบบมาเพื่อใช้กับ Windows NT โดยเฉพาะ แม้ว่าคุณสามารถติดตั้ง Windows NT ในระบบไฟล์ FAT32 ได้ แต่ว่า NTFS จะให้ประสิทธิภาพที่ดีกว่าในด้านระบบความปลอดภัยในการเข้าถึงไฟล์มากกว่า และเสียเนื้อที่น้อยกว่า

HPFS (High Performance File System)

เป็นระบบไฟล์ที่ออกแบบมาเพื่อใช้กับ OS/2 ซึ่ง HPFS ก็เหมือนกับ NTFS ที่จะมีระบบรักษาความปลอดภัยที่ดี, มีความเชื่อถือได้ของข้อมูล มีประสิทธิภาพและความเร็วสูงกว่า FAT

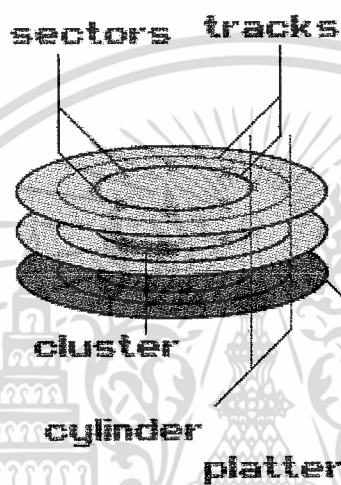
ตารางที่ 3.9 แสดงตัวอย่างระบบปฏิบัติการกับระบบไฟล์ที่ใช้

ระบบปฏิบัติการ	ระบบไฟล์ที่สนับสนุน
MS DOS 6.22	FAT 16
Windows 3.1	FAT 16
Windows 95 และ 95A	FAT 16
Windows 95B และ 95C	FAT 16 , FAT 32
Windows 98 และ Me	FAT 16 , FAT 32
Windows NT4	FAT 16 , NTFS (SP4)
Windows 2000	FAT 16 , FAT32 , NTFS
Windows XP	FAT 32 , NTFS

ปกติเมื่อผู้ใช้ซื้อฮาร์ดดิสก์มาใหม่ 1 ตัว ผู้ใช้ต้องทำการฟอร์แมต (Format) ฮาร์ดดิสก์ก่อนที่จะนำไปบรรจุข้อมูล การฟอร์แมตฮาร์ดดิสก์เป็นการแบ่งฮาร์ดดิสก์ออกเป็นส่วนๆ เพื่อให้คอมพิวเตอร์รู้ว่าตำแหน่งของ เซ็กเตอร์ (Sector) และแทร็ก (Track)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการจัดสรรเนื้อที่บนดิสก์ข้อมูลในฮาร์ดดิสก์ เพื่อให้การอ่าน-เขียนกระทำได้อย่างรวดเร็ว จะใช้วิธีกำหนดเป็นไซลินเดอร์ (cylinder) หรือเพื่อให้แทร็กที่ตรงกันทุกหน้าของฮาร์ดดิสก์ ซึ่งการกำหนดพาร์ติชันของฮาร์ดดิสก์ก็เป็นการแบ่งตามกลุ่มไซลินเดอร์ที่อยู่ต่อเนื่องกัน ทำให้มีลักษณะเป็นโลจิคอลไดรฟ์ และในแต่ละพาร์ติชันก็จะมีการจัดทำตาราง FAT สำหรับพาร์ติชันนั้น



รูปที่ 3.14 แสดงส่วนประกอบของฮาร์ดดิสก์

FAT กำหนดเนื้อที่สำหรับไฟล์โดยใช้หน่วย คลัสเตอร์ (cluster) คือเป็นกลุ่มของเซ็กเตอร์ ซึ่งอาจประกอบด้วย 4 ถึง 64 เซ็กเตอร์ หรือ ขนาดตั้งแต่ 2K ถึง 32K

ระบบปฏิบัติการดอส และวินโดวส์รุ่นแรกๆ ใช้ FAT16 (16 บิต) สามารถอ้างอิงหรือชี้ตำแหน่ง คลัสเตอร์ได้สูงสุด 65,536 คลัสเตอร์ (2^{16}) ในระบบ FAT16 สามารถมีขนาดของคลัสเตอร์ใหญ่ที่สุด 32 KB (Kilobyte) ดังนั้นจึงสามารถอ้างอิงข้อมูลในหนึ่งพาร์ติชัน (partition) ได้สูงสุดที่ 2 GB (gigabyte) ($32 \text{ KB} \times 65,536 \text{ คลัสเตอร์} = 2,097,152 \text{ KB}$ หรือ 2,048 MB หรือ 2 GB)

ดังนั้นถ้าผู้ใช้ต้องการใช้ฮาร์ดดิสก์ขนาด 2 GB หรือฮาร์ดดิสก์ที่มีพาร์ติชันเท่ากับ 2 GB หมายความว่าขนาดของคลัสเตอร์ที่เล็กที่สุดเท่ากับ 32 KB ซึ่งหมายความว่าไม่ว่าไฟล์ที่ต้องการเก็บในฮาร์ดดิสก์จะมีขนาดเล็กแค่ไหนก็ตาม ฮาร์ดดิสก์ก็ต้องจองพื้นที่ให้ไฟล์นี้ไม่ต่ำกว่า 32 KB ตัวอย่างเช่น ถ้าต้องการเก็บไฟล์ขนาด 1 KB ลงในฮาร์ดดิสก์ที่เป็น FAT 16 ฮาร์ดดิสก์จะต้องจองพื้นที่เพื่อเก็บไฟล์นี้ 32 KB ซึ่งหมายความว่าผู้ใช้จะต้องสูญเสียพื้นที่ในฮาร์ดดิสก์ไปโดยไม่สามารถ

ใช้ได้ถึง 31 KB ดังนั้นยังมีไฟล์ขนาดเล็กกว่า 32 KB มากเท่าไร หมายความว่า จะสูญเสียเนื้อที่ว่างบนฮาร์ดดิสก์ไปโดยไม่ได้ใช้ประโยชน์มากขึ้นเท่านั้น

ในระบบ FAT32 (32 บิต) จำนวนคลัสเตอร์ที่จะอ้างอิงถึงได้เท่ากับ 2^{28} หรือเท่ากับ 268,436,456 คลัสเตอร์ ดังนั้นเมื่อใช้ขนาดของคลัสเตอร์ 4 KB ขนาดของพาร์ทิชันสูงสุดที่จะมีได้จะเท่ากับ 8 GB และถ้าขนาดของคลัสเตอร์สูงสุดที่ 32 KB จะทำให้ฮาร์ดดิสก์สามารถมีพาร์ทิชันได้สูงที่สุดที่ 2 TB (1 Teta Byte เท่ากับ 1,024 GB)

FAT เป็นระบบไฟล์ชนิดหนึ่งที่ถูกกำหนดโดยซอฟต์แวร์ระบบปฏิบัติการ (operating system) ซึ่งระบบไฟล์ที่นิยมใช้กันอยู่ในปัจจุบันมี

- FAT ของระบบปฏิบัติการดอส (DOS) และวินโดวส์ (Windows)
- NTFS ของระบบปฏิบัติการวินโดวส์เอ็นที (Windows NT)
- HPFS ของระบบปฏิบัติการโอเอสทู (OS2)

FAT ที่นิยมใช้กันอยู่ใน ของระบบปฏิบัติการดอส และวินโดวส์ คือ FAT16 โดย FAT จะทำหน้าที่จัดการข้อมูลหลายๆ เซกเตอร์โดยในแต่ละเซกเตอร์จะแบ่งย่อยออกเป็นอีกหลายๆ คลัสเตอร์ (Cluster) ซึ่งในระบบ FAT16 (16 บิต) นั้นสามารถอ้างอิงหรือชี้ตำแหน่งคลัสเตอร์ได้สูงสุด 65,536 คลัสเตอร์ (ข้อมูลทางดิจิทัลจำนวน 1 bit สามารถเป็นได้เพียง 2 สถานะคือ 0 และ 1 ดังนั้นถ้าเป็น 16 บิต สามารถเป็นได้เท่ากับ 2 ยกกำลัง 16 ซึ่งเท่ากับ 65,536) แต่ในระบบ FAT16 นั้นสามารถมีขนาดของคลัสเตอร์ใหญ่ที่สุด 32 KB (Kilobyte) ดังนั้นในระบบ FAT16 จึงสามารถอ้างข้อมูลในหนึ่งพาร์ทิชัน (partition) ได้สูงที่สุดที่ 2 GB (gigabyte) (32 KB คูณ 65,536 คลัสเตอร์เท่ากับ 2,097,152 KB หรือ 2,048 MB หรือ 2 GB)

ดังนั้นถ้าผู้ใช้ต้องการใช้ฮาร์ดดิสก์ขนาด 2 GB หรือฮาร์ดดิสก์ที่มีพาร์ทิชันเท่ากับ 2GB หมายความว่าขนาดของคลัสเตอร์ที่เล็กที่สุดเท่ากับ 32 KB ซึ่งหมายความว่าไม่ว่าไฟล์ที่ต้องการเก็บในฮาร์ดดิสก์จะมีขนาดเล็กแค่ไหนก็ตาม ฮาร์ดดิสก์ก็จะต้องจองพื้นที่ให้ไฟล์นี้ไม่ต่ำกว่า 32 KB ตัวอย่างเช่น ถ้าต้องการเก็บไฟล์ขนาด 1 KB ลงในฮาร์ดดิสก์ที่เป็น FAT 16 ฮาร์ดดิสก์ก็จะต้องจองพื้นที่เพื่อเก็บไฟล์นี้ 32 KB ซึ่งหมายความว่าผู้ใช้จะต้องสูญเสียพื้นที่ในฮาร์ดดิสก์ไปโดยไม่สามารถใช้ได้ถึง 31 KB ดังนั้นยังมีไฟล์ขนาดเล็กกว่า 32 KB มากเท่าไร หมายความว่า จะสูญเสียเนื้อที่ว่างบนฮาร์ดดิสก์ไปโดยไม่ได้ใช้ประโยชน์มากขึ้นเท่านั้น วิธีแก้ไขปัญหาคือการสูญเสียเนื้อที่นี้อาจจะทำได้ 2 กรณีคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ถ้าผู้ใช้ยังต้องการใช้ฮาร์ดดิสก์ที่เป็น FAT16 อยู่ ผู้ใช้ต้องแบ่งพาร์ติชันฮาร์ดดิสก์ให้มีขนาดเล็กลง เพื่อให้ฮาร์ดดิสก์มีขนาดคลัสเตอร์เล็กลง จะได้ทำให้เนื้อที่ว่างที่ไม่สามารถใช้งานเหลือน้อยลง ดังตารางข้างล่างนี้ ถ้าผู้ใช้แบ่งพาร์ติชันฮาร์ดดิสก์ไว้ที่ขนาด 512 MB ต่อพาร์ติชันขนาดของคลัสเตอร์จะลดลงเหลือแค่ 8 KB ซึ่งจะทำให้ความสูญเสียเนื้อที่บนฮาร์ดดิสก์โดยเปล่าประโยชน์ลดน้อยลงถึง 3 เท่าเมื่อเทียบกับการแบ่งพาร์ติชันไว้ที่ขนาด 2 GB แต่วิธีการแบ่งฮาร์ดดิสก์ออกเป็นหลายๆ พาร์ติชันนี้อาจทำให้เกิดความยุ่งยากเช่น มีไคร์ฟฮาร์ดดิสก์หลายๆ ไคร์ฟ อาจทำให้สับสนเวลาใช้งาน

2. ให้ใช้ฮาร์ดดิสก์ที่เป็น FAT32 ซึ่งเป็นระบบ FAT แบบใหม่ 32 บิต ซึ่งมีในระบบปฏิบัติการวินโดวส์ 95 OSR2 (Windows 95 OEM Service Release 2) หรือในวินโดวส์ 98 หรือใช้โปรแกรมที่ช่วยแปลง FAT16 ให้เป็น FAT32 อย่างเช่น Partition-It, Partition Magic เป็นต้น สำหรับ FAT32 นี้ จำนวนคลัสเตอร์ที่จะอ้างอิงถึงได้เท่ากับ 2 ยกกำลัง 28 หรือเท่ากับ 268,436,456 คลัสเตอร์ ดังนั้นเมื่อใช้ขนาดของคลัสเตอร์ 4 KB ขนาดของพาร์ติชันสูงสุดที่จะมีได้จะเท่ากับ 8 GB และถ้าขนาดของคลัสเตอร์สูงสุดที่ 32 KB จะทำให้ฮาร์ดดิสก์สามารถมีพาร์ติชันได้สูงที่สุดที่ 2 TB (1 Teta Byte เท่ากับ 1,024 GB)

ตารางที่ 3.10 แสดงการเปรียบเทียบขนาดระหว่างคลัสเตอร์ FAT32 และ FAT16

ขนาดของพาร์ติชัน	ขนาดของคลัสเตอร์ FAT 32	ขนาดของคลัสเตอร์ FAT 16
น้อยกว่า 260 Megabyte	512 byte	4 kilobyte
260-510 Megabyte	4 kilobyte	8 kilobyte
512-1023 Megabyte	4 kilobyte	8 kilobyte
1024-2048 Megabyte	4 kilobyte	16 kilobyte
2-8 Gigabyte	4 kilobyte	32 kilobyte
8-16 Gigabyte	8 kilobyte	32 kilobyte
16-32 Gigabyte	16 kilobyte	32 kilobyte
มากกว่า 32 Gigabyte	32 kilobyte	32 kilobyte

ในเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ 95 นั้น ถ้าผู้ใช้สำรวจดูในโฟลเดอร์ (Folder) ต่างๆ จะพบว่ามีไฟล์ขนาดเล็กๆ เป็นจำนวนหลายร้อยไฟล์ ตัวอย่างเช่น ไฟล์ที่มีนามสกุล .dll ซึ่งไฟล์เหล่านี้จะเป็นต้นตอของการทำให้เกิดการสูญเสียเนื้อที่ในฮาร์ดดิสก์อย่างไม่มีประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไม่ว่าฮาร์ดดิสก์จะมีคลัสเตอร์เป็นแบบ FAT32 หรือ FAT 16 ก็ตาม จากตารางข้างล่างต่อไปนี้ จะแสดงถึงไฟล์ต่างๆ ในไฟล์เคอร์ของวินโดวส์ 95 โดยทั่วไป เมื่อใช้ฮาร์ดดิสก์มีคลัสเตอร์เป็น 4 Kilobyte (FAT32) เนื้อที่ในฮาร์ดดิสก์ 83.5 เปอร์เซ็นต์เป็นเนื้อที่ที่เสียไปโดยไม่มีประโยชน์ และเมื่อใช้ฮาร์ดดิสก์มีคลัสเตอร์เป็น 32 Kilobyte (FAT16) เนื้อที่ในฮาร์ดดิสก์ 97.9 เปอร์เซ็นต์เป็นเนื้อที่ที่เสียไปโดยไม่มีประโยชน์

3.3.3 ข้อเสียของ FAT 32

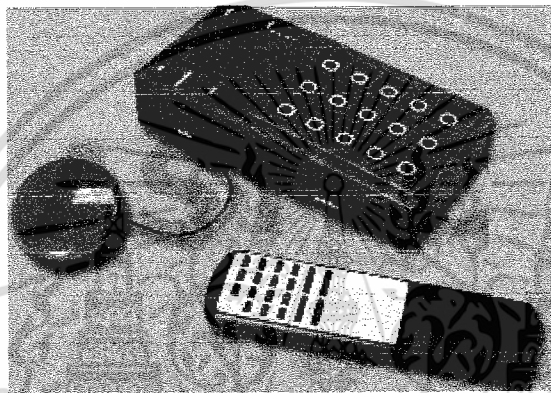
ไม่สนับสนุนการทำงานร่วมกับระบบปฏิบัติการอื่น รวมถึงวินโดวส์ 95 เวอร์ชันเดม และ วินโดวส์เอ็นที 4.0

ในการใช้โปรแกรม Utility ที่จัดการกับดิสก์ ผู้ใช้ต้องตรวจสอบ โปรแกรมนั้นว่าสนับสนุน ระบบ FAT32 หรือไม่ ถ้าไม่สนับสนุนหรือไม่ได้ตรวจสอบแล้วผู้ใช้ใช้โปรแกรมนั้นกับฮาร์ดดิสก์ ที่เป็น FAT32 จะทำให้เกิดความเสียหายกับข้อมูลในฮาร์ดดิสก์ โดยไม่สามารถกู้ข้อมูลกลับคืนมา สำหรับโปรแกรมวินโดวส์ 95 OSR2 นั้นจะมีซอฟต์แวร์ Fdisk, Format, Scandisk และ Defrag ที่สนับสนุน FAT32 แต่มีซอฟต์แวร์ 1 ตัวในวินโดวส์เวอร์ชันนี้ที่ไม่สนับสนุน FAT32 คือ DriveSpace3 ดังนั้นจึงห้ามใช้ซอฟต์แวร์นี้กับฮาร์ดดิสก์ที่เป็น FAT32 เป็นอันขาด

บทที่ 4

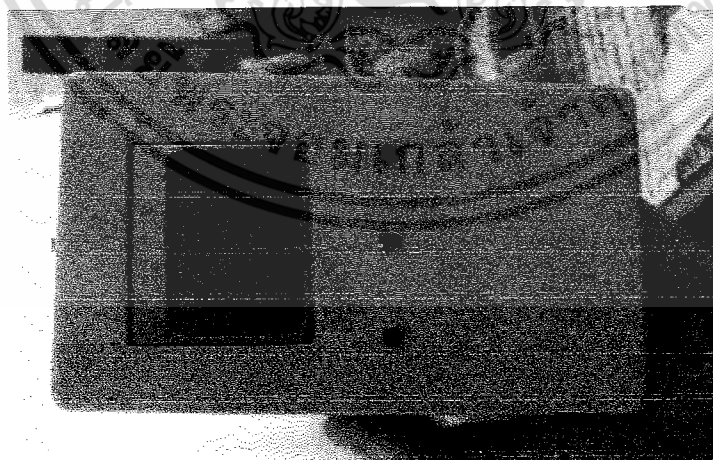
การออกแบบและหลักการทำงาน

เนื่องจากโครงการนี้เป็นการพัฒนาต่อจากโครงการอุปกรณ์เครื่องช่วยในการสื่อสาร
”โอบา” อุปกรณ์พกพาซึ่งมีขนาดที่ใหญ่ดังรูปที่ 4.1



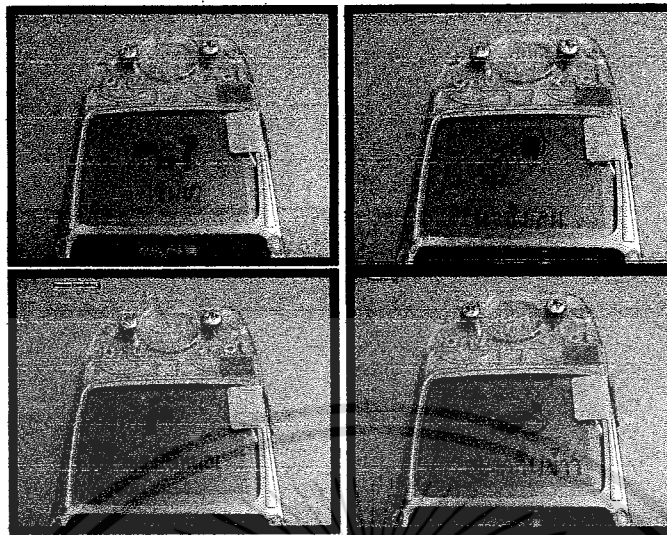
รูปที่ 4.1 แสดงอุปกรณ์เครื่องช่วยในการสื่อสารแบบพกพา

ดังนั้นจึงได้พัฒนาเพื่อขนาดที่เล็กลง วัตถุประสงค์การดำเนินงาน คือสร้างต้นแบบที่ใช้
งานได้ ราคาประหยัด ใช้งานง่ายและกินกำลังงานต่ำ ระบบที่ออกแบบเป็นดังรูปที่ 4.2



รูปที่ 4.2 แสดงอุปกรณ์เครื่องช่วยในการสื่อสารแบบพกพาที่ได้พัฒนาแล้ว
แต่เนื่องจากการแสดงผลยังเป็นการใช้จอ LCD แบบจอขาวดำ และในส่วนของเสียงใช้ไอซี
บันทึกเสียง ที่บันทึกข้อมูลได้จำนวนน้อย ดังแสดงในรูปที่ 4.3

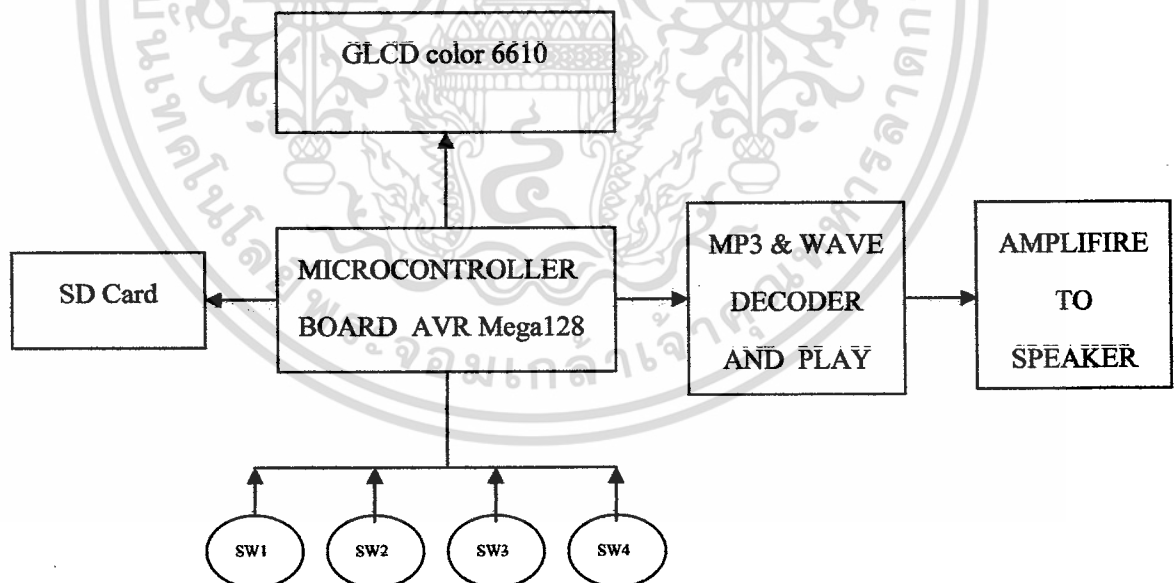
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงอุปกรณ์เครื่องช่วยในการสื่อสารแบบพกพาที่ใช้จอ LCD แบบจอขาวดำ

จึงได้มีการพัฒนาต่อ โดยการเปลี่ยนจากจอแสดงผลขาวดำ เป็นจอแสดงผลแบบสีที่มีความละเอียดมากกว่า และให้การแสดงผลที่ดูคมชัดและทันสมัยกับเทคโนโลยีปัจจุบัน และเพิ่มหน่วยความจำได้มากขึ้น โดยใช้ SD Card เป็นตัวเก็บข้อมูลทั้งภาพและเสียง โดยมีการออกแบบการทำงานดังรูปที่

4.4



SW1=ปุ่มเลือกเมนูแสดงหมวดภาพ SW2=ปุ่มเลือกภาพหรือเมนูก่อนหน้า
 SW3=ปุ่มเลือกภาพหรือเมนูถัดไป SW4=ปุ่มเล่นเสียง

รูปที่ 4.4 แสดงแผนภาพการออกแบบวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการออกแบบแบ่งเป็น 3 ส่วนหลักๆคือ

- 1) ส่วนของการแสดงภาพ
- 2) ส่วนของการเล่นเสียง
- 3) ส่วนของ SD Card

4.1) การออกแบบและหลักการทำงานของวงจร

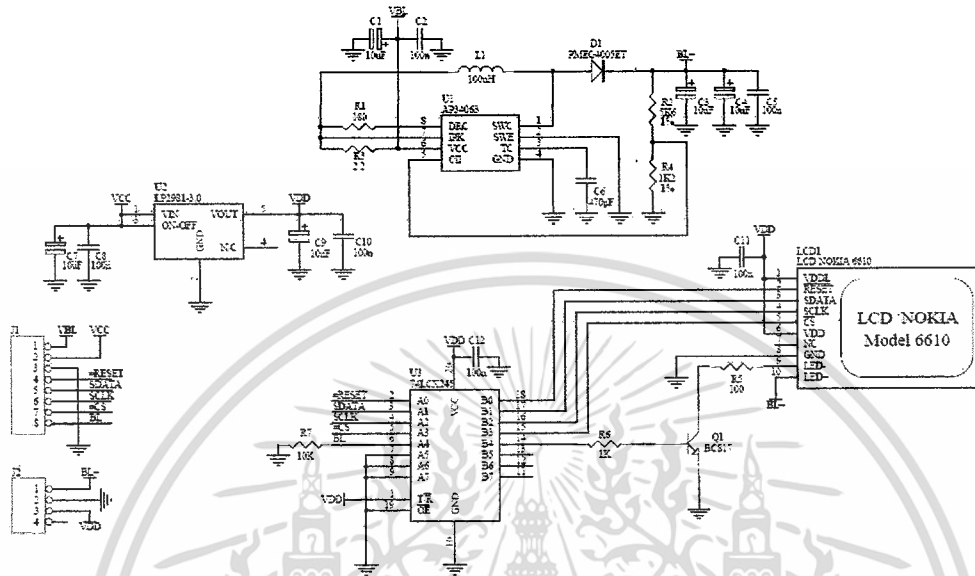
หลักการและจุดประสงค์ในการออกแบบ

- 1 เพื่อติดต่อกันระหว่าง AVR Mega 128 กับ GLCD Color 6610
- 2 เพื่อติดต่อกันระหว่าง AVR Mega 128 กับ VS1002D
- 3 เพื่อติดต่อกันระหว่าง AVR Mega 128 กับ SD Card
- 4 การแสดงผลเป็นภาพสีด้วยจอ GLCD Color 6610 แทนจอ LCD 3310

4.2) อุปกรณ์ที่นำมาใช้งาน

- 1 GLCD Color 6610
- 2 ไมโครคอนโทรลเลอร์ AVR เบอร์ ATmega 128
- 3 SD Card มีหน่วยความจำขนาด 32 M
- 4 Switch ที่ใช้ในการควบคุม
- 5 ไอซีถอดรหัสเสียง ไฟล์ MP3 หรือ WAVE (IC VLSI เบอร์ VS1002D)

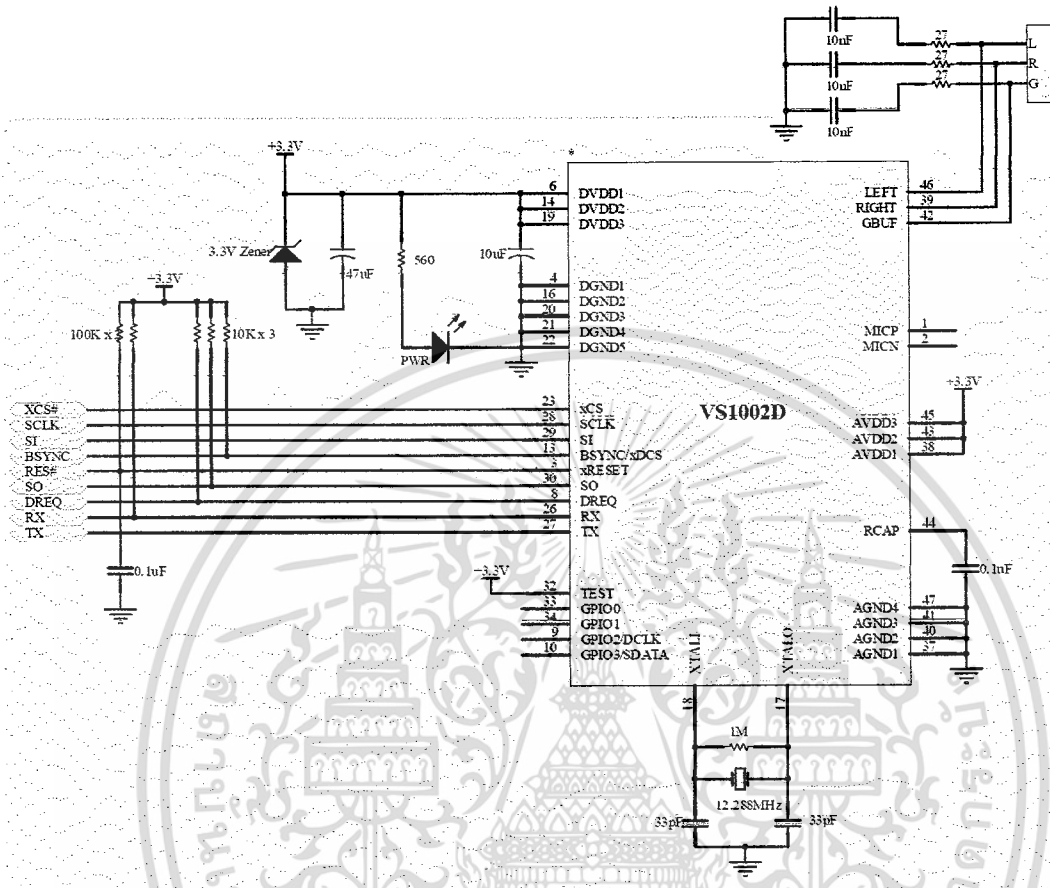
4.3) การออกแบบวงจรเชื่อมต่อกับ GLCD 6610



รูปที่ 4.5 วงจรเชื่อมต่อกับ GLCD 6610

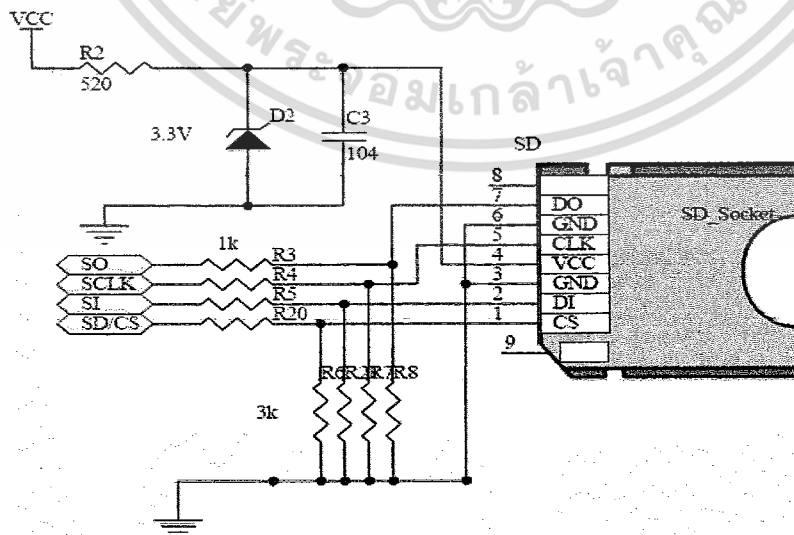
ออกแบบให้ไฟเลี้ยงจอ GLCD 6610 เป็น 3.0 Volte และตัวไมโครคอนโทรลเลอร์ AVR ใช้แรงดันได้ตั้งแต่ 1.8-5 Volte ในการทำงาน (ขึ้นอยู่กับเบอร์ที่เลือกใช้งาน) ดังนั้นจึงต้องใช้ตัวควบคุมแรงดันให้เป็น 3.0 Volte ในที่นี้เลือก IC เบอร์ LP2981-3.0 ในการควบคุมแรงดัน

4.4) การออกแบบวงจรถอดรหัสไฟล์ MP3 และ WAVE



รูปที่4.6 วงจรถอดรหัสไฟล์ MP3 และ WAVE

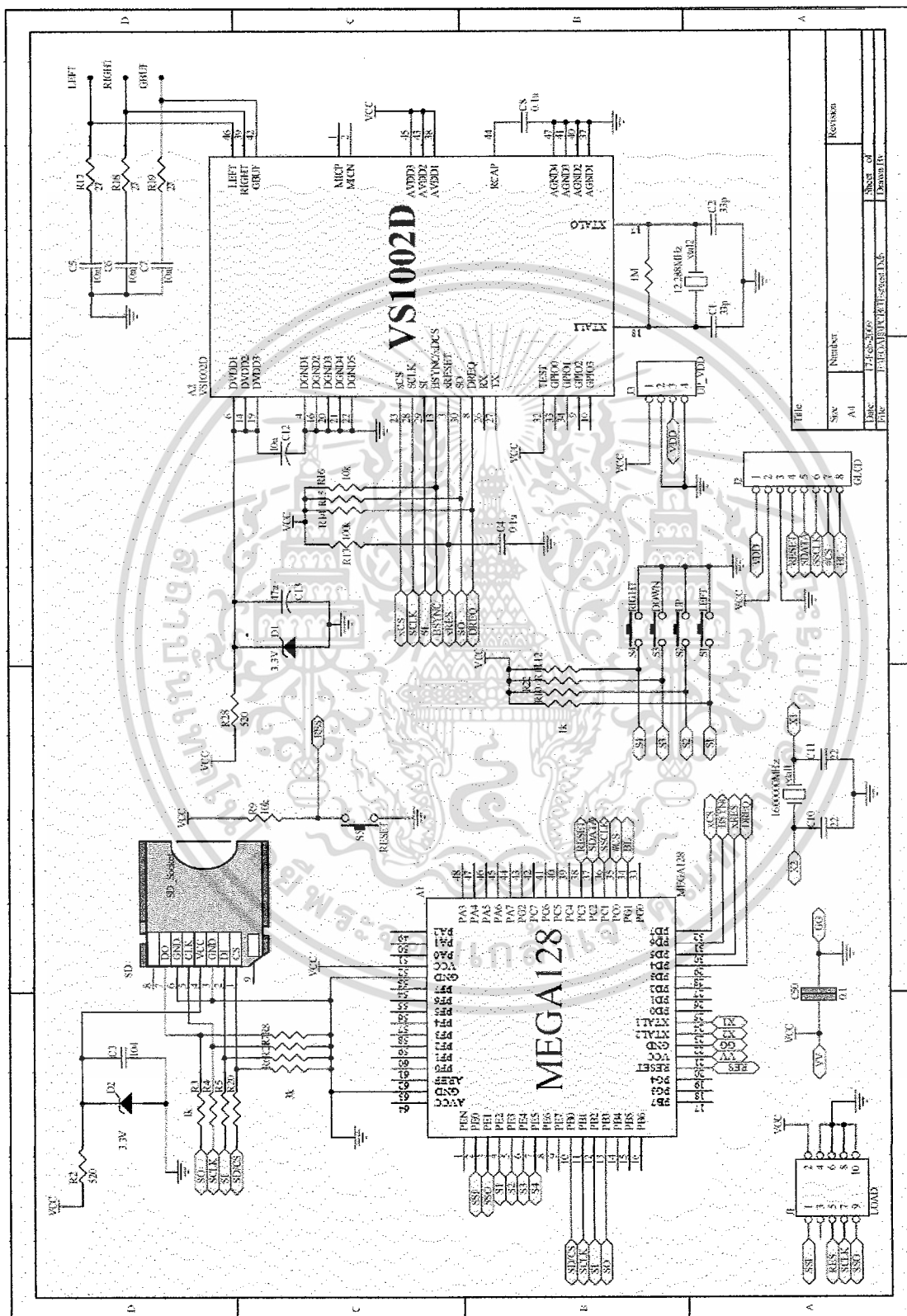
4.5) การออกแบบวงจร SD Card



รูปที่4.7 วงจร SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6) การออกแบบวงจรรวม



4.7) การออกแบบโปรแกรมเพื่อทำงานร่วมกับอุปกรณ์

4.7.1 จุดประสงค์ในการออกแบบ

- 1 เพื่อให้สามารถติดต่ออุปกรณ์ผ่านทางพอร์ตอนุกรมได้
- 2 เพื่อให้สามารถอ่านภาพและเสียงจาก SD Card ได้
- 3 ผลภาพกราฟิกบนทาง GLCD 6610 ได้
- 4 เพื่อให้สามารถถอดรหัสไฟล์เสียง MP3 และ WAVE ได้
- 5 เพื่อให้สามารถเล่นเสียงได้เมื่อต้องการอธิบายภาพที่ปรากฏ

4.7.2 ซอฟต์แวร์ (Software) ที่ใช้ในการออกแบบ

- 1 CodeVisionAVR V2.03.4 ซึ่งใช้ในการเขียนภาษาซีในการสั่งงาน AVR Mega128 ให้ติดต่อและควบคุมอุปกรณ์
- 2 AVR Studio ซึ่งใช้ในการเบิร์น hex ไฟล์สู่ AVR Mega128
- 3 Photo Shop ใช้ในการตกแต่งภาพที่ใช้แสดงผลทางหน้าจอ GLCD
- 4 Paint ซึ่งใช้ในการปรับแต่งและบีบอัดภาพให้ได้ตามขนาดของหน้าจอ GLCD
- 5 Jet Audio ใช้บันทึกไฟล์เสียง เป็น MP3 หรือ WAVE

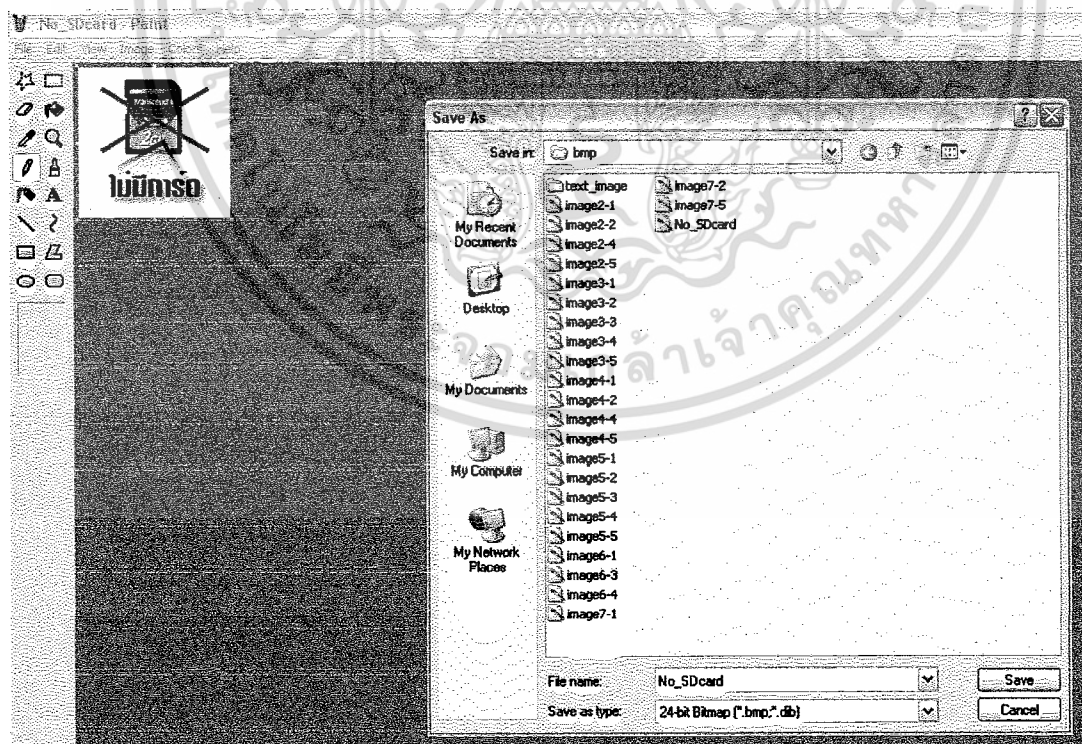
4.7.3 การออกแบบ

การทำงานของตัวโปรแกรมเมื่อต้องการโหลดภาพจากคอมพิวเตอร์เข้าสู่ AVR Mega128 เพื่อส่งต่อไปยัง GLCD color 6610 สามารถทำได้โดยกำหนดขนาดภาพ ให้มีขนาด 132x132 pixel โดยใช้โปรแกรม Photo Shop หลังจากนั้นกดโหลดภาพเข้าสู่โปรแกรม Paint และเนื่องจากจอ GLCD color 6610 เป็นชนิดจอสีจึงต้องเปลี่ยนชนิดของภาพเป็นชนิด 24-bit Bitmap ซึ่งแสดงเป็นภาพสี

การทำงานของโปรแกรมเมื่อต้องการโหลดเสียงเข้าสู่ SD card ทำได้โดยการบันทึกเสียงโดยใช้โปรแกรม Jet Audio บันทึกเป็นไฟล์ MP3 หรือ WAVE บันทึกเสียงจากคอมพิวเตอร์เข้าสู่หน่วยความจำของ SD card



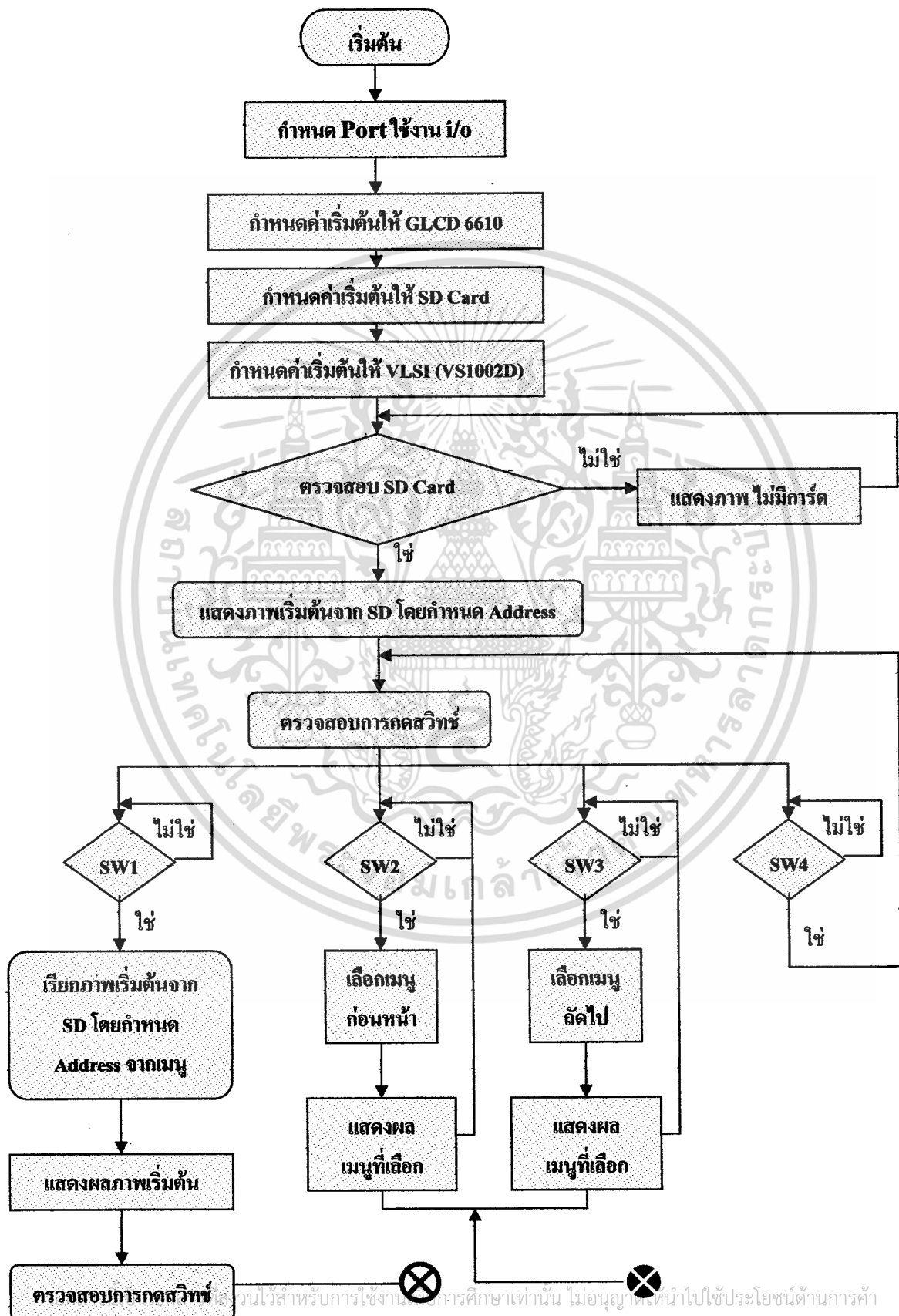
รูปที่4.10 แสดงการปรับภาพขนาด 132x132 pixel โดยใช้โปรแกรม Photo shop



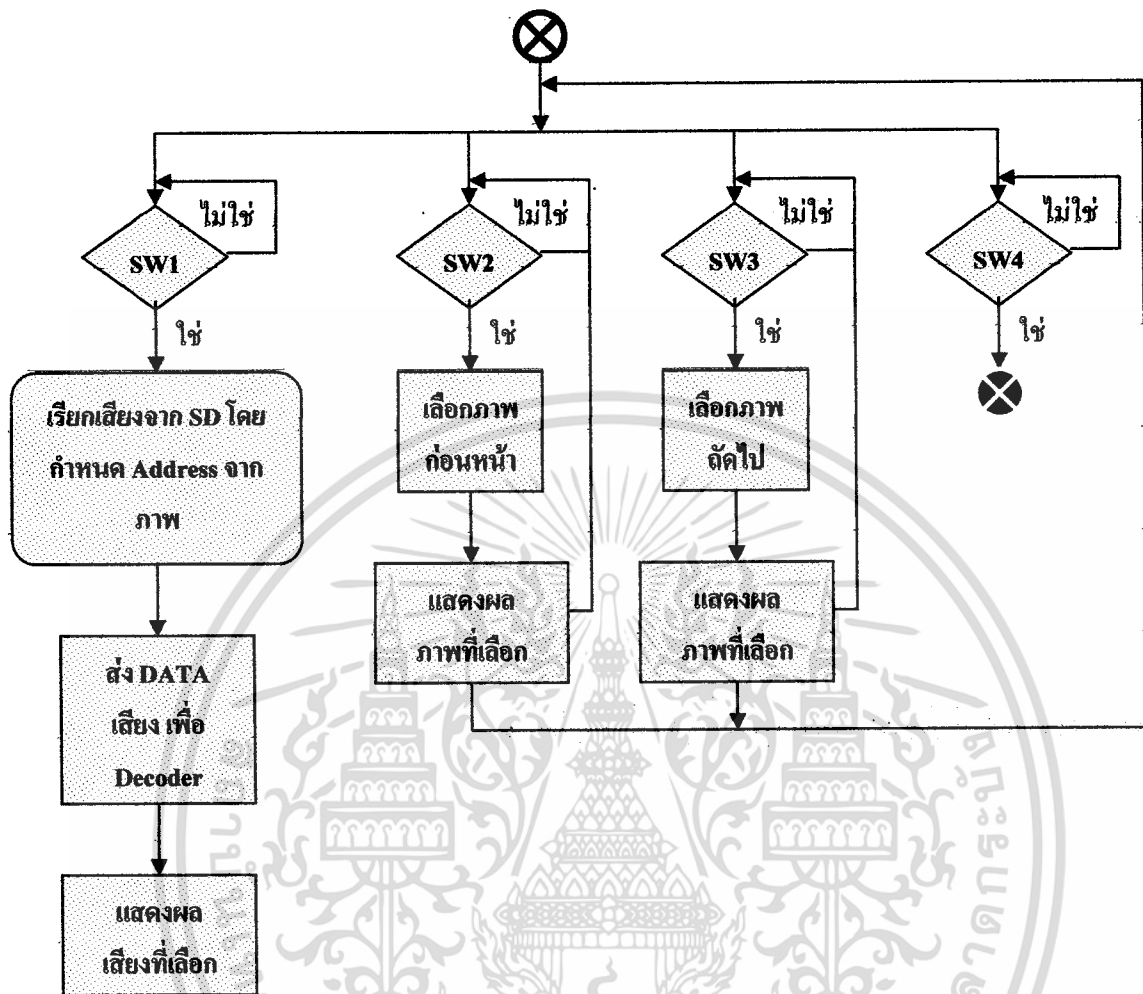
รูปที่4.11 แสดงการปรับเปลี่ยนชนิดของภาพเป็นชนิด 24-bit Bitmap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8) แผนผังการทำงานของโปรแกรม



ส่วนไว้สำหรับการใช้งานในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

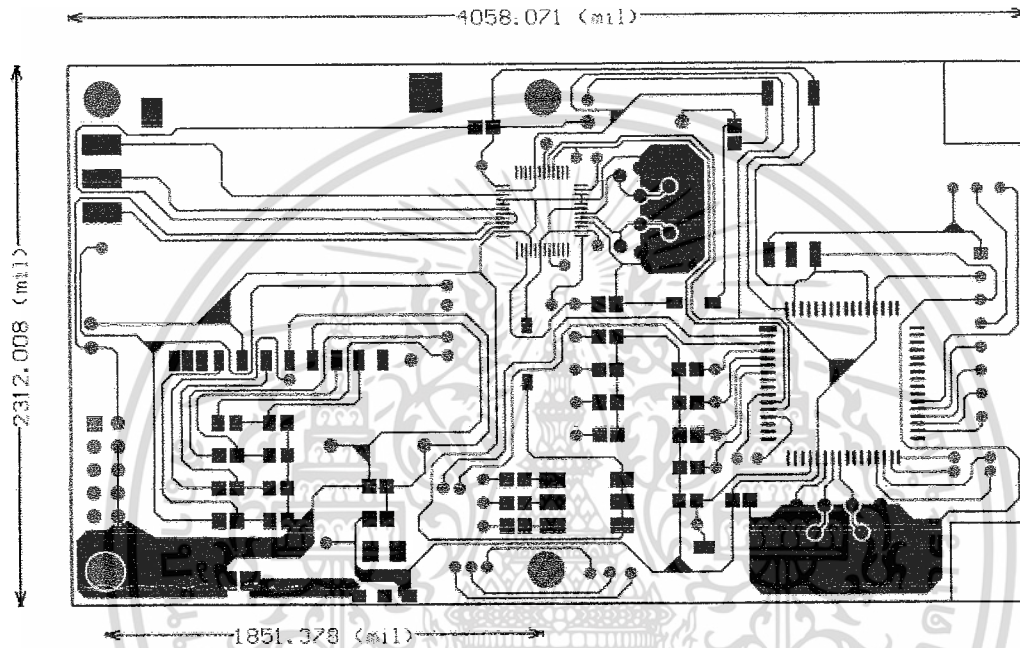


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

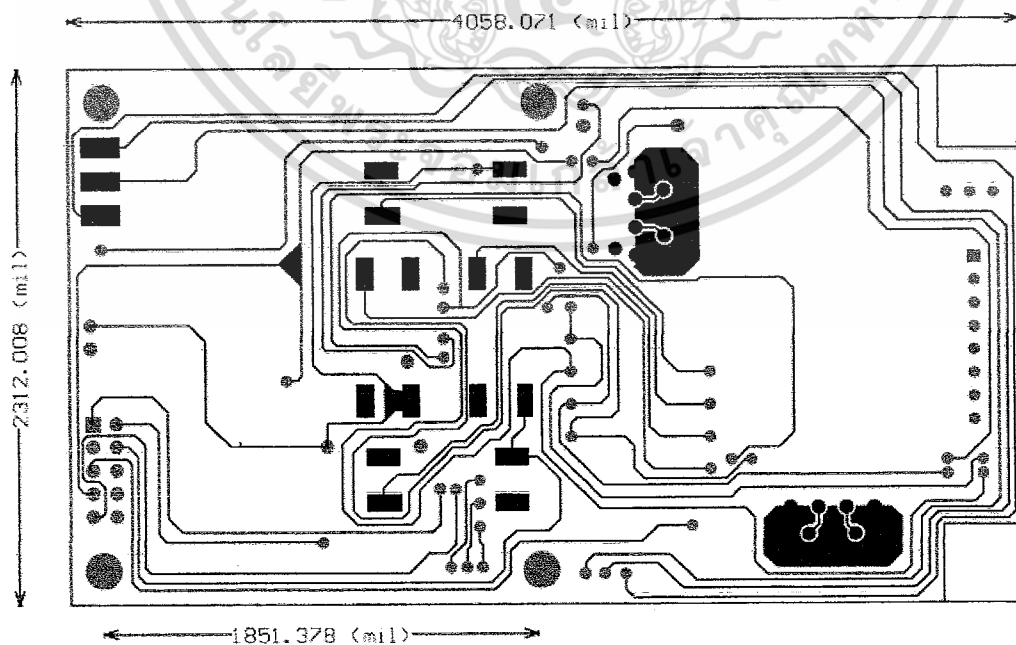
บทที่ 5

โครงสร้างทางฮาร์ดแวร์และซอฟต์แวร์ของเครื่อง

5.1) การออกแบบ PCB



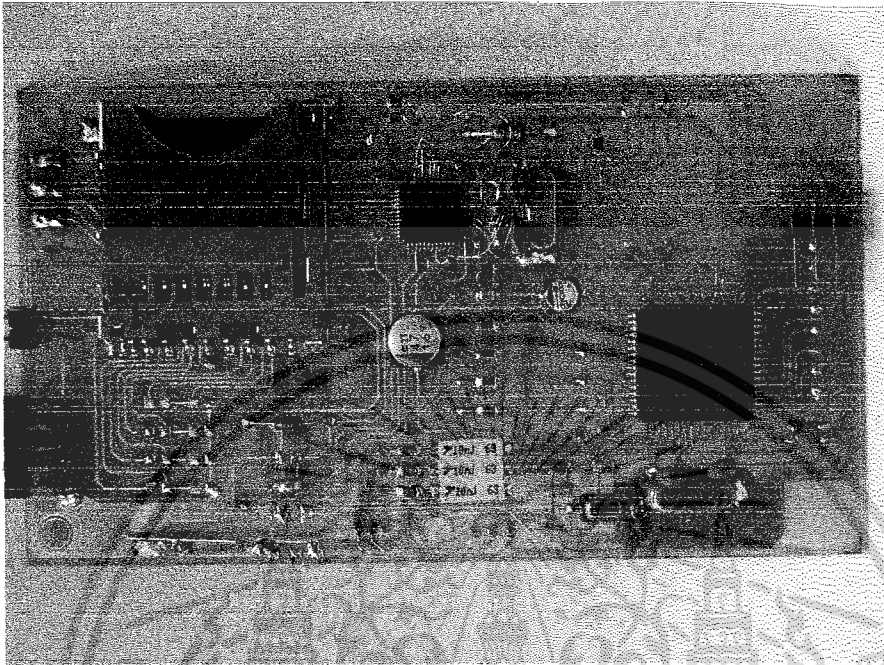
รูปที่ 5.1 การออกแบบ PCB ด้าน TOP LAYER



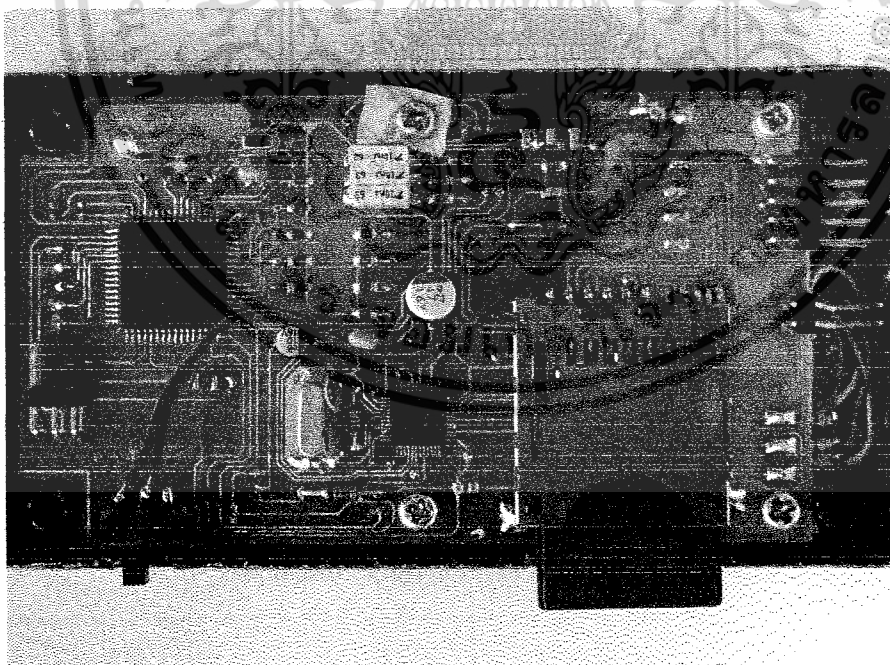
รูปที่ 5.2 การออกแบบ PCB ด้าน BUTTOM LAYER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2) การลงอุปกรณ์

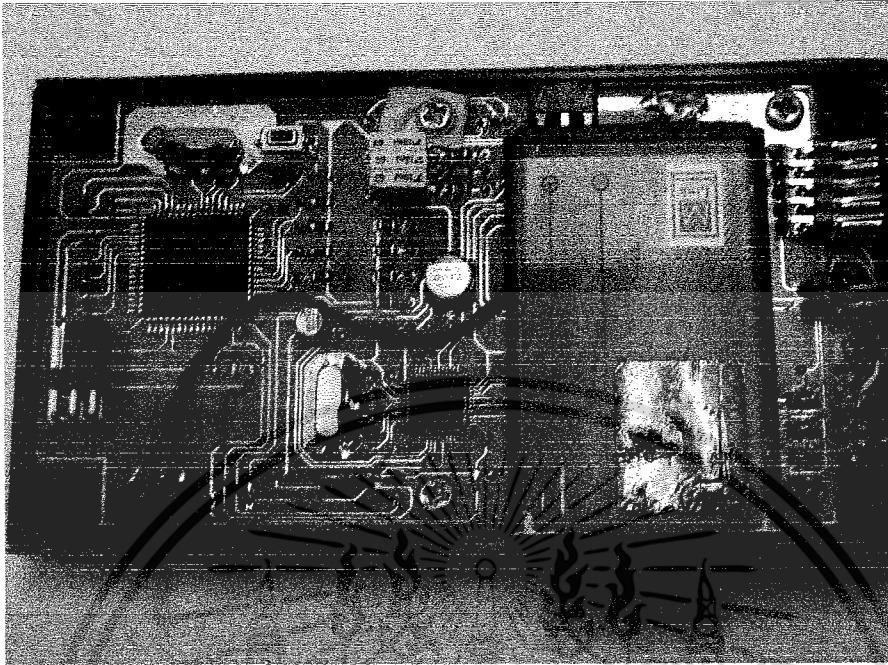


รูปที่ 5.5 ลายปริ๊นท์พร้อมลงอุปกรณ์



รูปที่ 5.6 แผ่นปริ๊นท์ที่ลงอุปกรณ์แล้วใส่กล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 การใส่แบตเตอรี่

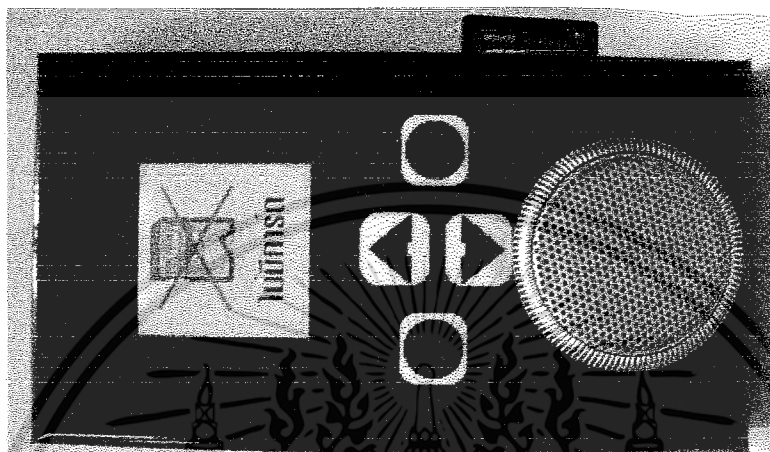


รูปที่ 5.8 ตัวเครื่องที่ประกอบเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3) การทดลองและการใช้งาน

เมื่อทำการประกอบตัวเครื่องเรียบร้อยแล้วทำการเปิดสวิทช์โดยที่ยังไม่ใส่ SD Card ที่หน้าจอ GLCD จะแสดงภาพพร้อมข้อความ “ไม่มีการ์ด” ดังแสดงในรูปด้านล่าง



รูปที่ 5.9 แสดงภาพพร้อมข้อความ “ไม่มีการ์ด” โดยที่ยังไม่ใส่ SD Card

ภาพที่แสดงบนจอ GLCD เป็นภาพที่เก็บไว้ในหน่วยความจำของไมโครคอนโทรลเลอร์ จะแสดงเมื่อไมโครคอนโทรลเลอร์หาไฟล์ภาพเริ่มต้นจาก SD Card ไม่พบ และจะแสดงตลอดไม่ว่าจะมีการกดสวิทช์ใดๆก็ตาม จนกว่าจะมีการใส่ SD Card เข้าไปในเครื่อง



รูปที่ 5.9 แสดงภาพพร้อมข้อความ “ไม่มีการ์ด”

เมื่อใส่SD Card แล้วทำการกดปุ่มใดๆ ไมโครคอนโทรลเลอร์จะค้นหาไฟล์ภาพเริ่มต้นจาก SD Card และแสดงบนจอ GLCD ภาพที่แสดงจะเป็นภาพที่ไว้สำหรับการเลือกใช้เสียงผู้ชาย หรือ เสียงผู้หญิง โดยการเลือกกดสวิทช์ ด้านซ้าย หรือขวา ดังแสดงในภาพด้านล่าง



รูปที่5.10 แสดงภาพเริ่มต้นพร้อมข้อความ เพื่อเลือกการใช้เสียง

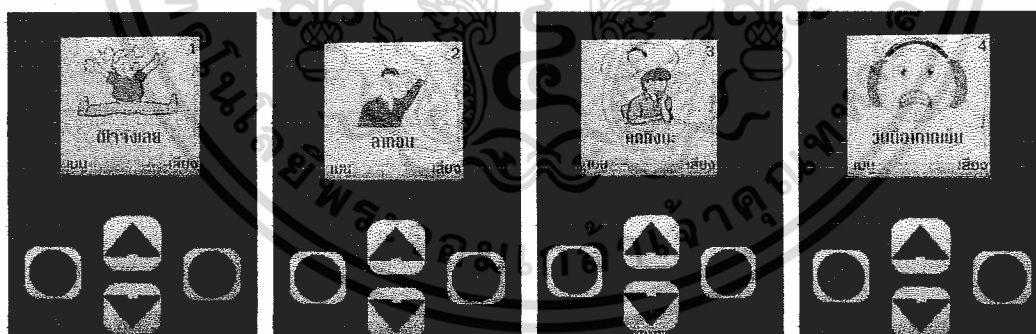


รูปที่5.11 แสดงภาพการเปลี่ยนภาพเมื่อมีการกดสวิทช์

เมื่อมีการกดสวิทช์เลือกการใช้เสียงที่หน้าจอ GLCD จะแสดงภาพเมนูเริ่มต้น โดยได้ทำการออกแบบ เมนูไว้ใช้งาน 7 เมนู สามารถเลือกเมนูได้โดยกดเลือกสวิทช์ ขึ้น ลง โดยแต่ละเมนู จะแสดงภาพได้ 5 ภาพ รวมทั้งหมด 35 ภาพ โดยการกดเลือกสวิทช์ด้านซ้าย ส่วนสวิทช์ด้านขวาไว้สำหรับออกจากหมวดเมนูกลับไปภาพเริ่มต้น



รูปที่ 5.10 แสดงภาพเมนูเริ่มต้น



รูปที่ 5.11 แสดงภาพตัวอย่างของแต่ละเมนู

เมื่อเข้าสู่หมวดภาพต่างๆจะมีภาพแสดงดังรูป และภาพแต่ละภาพจะมรเสียงที่สื่อถึงภาพนั้นๆ โดยสามารถเลือกเล่นเสียงได้โดยกรเลือกกดสวิทช์ ทางด้านขวา ส่วนสวิทช์ด้านซ้ายจะกลับไปเมนูที่เราเลือกเข้ามา

5.4) การแสดงภาพ

ในโปรแกรมจะมี Function ส่งข้อมูลภาพ ที่เป็น Format BMP มีขนาด 132 x 132 pixel หรือ $132 \times 132 \times 3 = 52272$ byte ~ 52k byte ต่อ 1 ภาพ โดยไมโครคอนโทรลเลอร์จะอ่านข้อมูลภาพ จาก SD Card ทีละ 1 Sector (512 byte) โดย Sector แรกจะเริ่มอ่านที่ตำแหน่ง byte = 55 มาเก็บไว้ในตัวแปร SBuf และใช้วนลูปส่งข้อมูล 512 รอบ โดยแต่ละbyte ไมโครคอนโทรลเลอร์จะนำไปเขียนลงใน หน่วยความจำของจอ GLCD จนครบ 52272 byte

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	42	4D	66	CC	00	00	00	00	00	00	36	00	00	00	28	00	BMF.....6...(.
00000010	00	00	84	00	00	00	84	00	00	00	01	00	18	00	00	00	..0I.Ä...Ä.....
00000020	00	00	30	CC	00	00	C4	0E	00	00	C4	0E	00	00	00	00by)by.py.b
00000030	00	00	00	00	00	00	FE	79	03	FE	79	03	FE	79	03	FE	y.py.py.ow.ót.it
00000040	79	03	FE	79	03	FE	79	03	F8	77	03	F3	74	04	EE	74	.ál.Ög.Ög.Ög.Ü
00000050	07	E1	6C	05	D6	67	04	D6	67	04	D6	67	04	CD	63	05	í.í.Ög.Ög.Úu.Ú
00000060	CD	63	05	CD	63	05	D6	67	04	D6	67	04	D9	75	1B	D9	u.Đ16Đ16'z='z= g
	61	1C	AD	5C	12	BE	5D	07	CD	63	05	E1	6C	05			.-a.-\.*].í.ál.
	04	F8	77	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	ót.ow.py.py.py.p
	79	03	FE	79	03	FE	79	03									.py.py.py.py.py
	03	FE	79	03	FE	79	03	FE									py.py.py.py.py.
	FE	79	03	FE	79	03	FE	79									y.py.py.py.py.p
	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	y.py.py.py.py.py
	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	.py.py.py.py.py.
	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	py.py.py.py.py.p
	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	y.py.py.py.py.py
	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	.py.py.py.py.py.
	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	py.py.py.py.py.p
	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	y.py.py.py.py.py
	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	.py.py.py.py.py.
	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	79	03	FE	py.py.py.py.py.p
	79	03	FE	79	03	FE	79	03	F8	77	03	F3	74	04	E1	6C	y.py.ow.ót.ót.ál
	05	E1	6C	05	D6	67	04	D6	67	04	D6	67	04	D6	67	04	.ál.Ög.Ög.Ög.Ög.
	D6	67	04	D6	67	04	D6	67	04	D6	67	04	D6	67	04	D6	Ög.Ög.Ög.Ög.Ög.Ö
	67	04	D6	67	04	CD	63	05	CD	63	05	CD	63	05	D6	67	g.Ög.í.í.í.Ög
	04	D6	67	04	E1	6C	05	E1	6C	05	E1	6C	05	D6	67	04	.Ög.ál.ál.ál.Ög.

รูปที่ 5.11 แสดงข้อมูลไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน ส่งข้อมูลภาพ

```

void SendBMP(unsigned long data)
{ unsigned long i,j;
  for(i=0;i<((FileSize/512)+1);i++)          // loop Sector Count ; 1 Sector = 512 byte
    { Read_SD(data+i);                       // Read Data from File Sector
      j=0;
      if(i==0)
        { j =55;
          while(j<512)                       // loop for send data 512 byte
            { bmp1 = SBuf[j]/0x10;
              bmp1 = bmp1*0x10;
              j++;
              bmp2 = SBuf[j]/0x10;
              sum = bmp1+bmp2;
              sendData(sum);
              j++;
            }
          } else
            { while(j<512)                   // loop for send data 512 byte
              { bmp1 = SBuf[j]/0x10;
                bmp1 = bmp1*0x10;
                j++;
                bmp2 = SBuf[j]/0x10;
                sum = bmp1+bmp2;
                sendData(sum);
                j++;
              }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง คือ สามารถเขียน โปรแกรมเพื่อควบคุม MCU ให้ติดต่อกับ จอ GLCD และส่งข้อมูลภาพ โดยการอ่านข้อมูลจาก SD Card

5.5) การแสดงเสียง

ในโปรแกรมจะมี Function ส่งข้อมูลเสียง ที่เป็น Format MP3 มีขนาดไม่แน่นอนขึ้นอยู่กับ การบันทึก โดยไมโครคอนโทรลเลอร์จะอ่านข้อมูลภาพ จาก SD Card ทีละ 1 Sector (512 byte) มา เก็บไว้ในตัวแปร SBuf และใช้วนลูปส่งข้อมูล 512 รอบ โดยแต่ละbyte ไมโครคอนโทรลเลอร์จะส่ง ข้อมูลให้ ไอซีถอดรหัส MP3 ทำการแปลงข้อมูลเป็นสัญญาณเสียง

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FF	FA	92	80	64	0F	00	00	02	74	07	45	61	29	30	A0	yú'ld...t.Ea)0
00000010	51	41	08	AC	25	83	24	07	78	23	29	87	A4	64	81	0D	QA.-%IS.x#)I'dl.
00000020	04	E5	30	F4	8C	90	05	C8	D9	08	0D	4D	C8	48	00	03	.á0ó .ÈÜ..MÈH..
00000030	A0	5A	20	08	0C	08	C3	6C	83	A6	61	00	89	F0	F8	60	Z...Ã a.l8ø`
00000040	B8	7C	8B	33	81	82	7F	38	7C	F9	41	AF	F2	85	0B	DD	. 13 8 úA`òl.Y
00000050	F0	1A	8E	1F	11	94	01	BC	E0	62	D3	82	41	38	82	93	ä...l..kábÓIAB
00000060	E4	1A	4D	CA	D4	50	BF	F8	E0	18	10	C6	B7	D9	CE	13	ä.MÉ0Pçóà..Æ·ÚÍ.
00000070	10	26	F3	82	70	5D	8D										.Sólp]l'.aóIeÀ .
00000080	44	01	05	85	73	37	1B										D..ls7.^À.(...ç
00000090	06	6B	1C	70	40	5C	70										.k.p@`pEç.yrIzQÀ
000000A0	99	C4	73	E7	F3	E1	95	04	47	4A	24	98	60	3C	0F	88	lÀsçóá .GJS `<.l
000000B0	24	CF	90	27	A9	6F	A1	3F	D0	16	01	94	FA	59	A4	81	\$Íl'çol?B..lúY*I
000000C0	71	01	38	58	E0	9C	25	AC	90	B5	57	54	81	DB	05	96	q.8Xà %-lµWT Ü.l
000000D0	48	D3	85	41	84	07	8C	34	2D	C3	A8	43	CC	5C	8B	31	HÓIAI.l4-Ã`Cì\l1
000000E0	5A	12	42	E3	F7	3E	7E	7E	F9	C0	FB	77	13	A6	B5	17	Z.Bã->~`úÁúw. µ.
000000F0	B1	4B	D2	51	62	AD	45	02	B7	9F	45	0B	EB	94	ED	46	±KÓQb-E..lE.èlIF
00000100	2A	A0	AE	D6	C6	95	55	52	0A	A8	2B	32	25	95	61	11	* @0ÆIUR..+2%la.
00000110	D4	8C	18	28	38	AD	AA	82	94	88	5C	64	2B	18	5D	A8	Ól.(8 @ l\d+.]`
00000120	85	5D	26	C7	1F	52	C8	3D	8D	74	73	5D	73	6B	13	9E	l &ç.RE=Its]sk.l
00000130	A5	AC	62	CC	AC	8A	16	C3	A2	44	A0	81	EB	48	48	2E	¥-bí-l.ÃçD lèHH.
00000140	83	FF	F6	8A	AB	0E	6F	ED	96	49	24	8D	A0	29	D3	74	lÿó «.oillS)Ót
00000150	78	50	D8	A8	04	80	68	1B	0A	45	D4	02	2E	1F	17	22	xP0`.lh..Eó...."
00000160	C0	92	D0	5D	53	B6	28	61	B5	84	1C	20	32	71	88	40	À`ð]S¶(aµl.2q @
00000170	BE	68	08	A0	C0	A1	4B	B0	2D	99	95	B6	97	36	64	8C	¾h. ÀIK°-l ¶l6d
00000180	57	67	CA	2D	88	AE	80	E7	FB	6D	24	92	4A	DA	05	AF	พฐÈ-l@ çúm\$`JÚ.-
00000190	D5	A6	8B	23	8B	54	1B	76	03	22	40	75	F4	4E	71	DB	Ö;l#IT.v."@úNqÚ
000001A0	35	B3	FF	FA	92	80	B1	0E	2E	80	02	0C	07	CB	69	A8	5`yú`l±..l...Èi"

รูปที่ 5.11 แสดงข้อมูลไฟล์เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน ส่งข้อมูลเสียง

```

void Send_Sound(unsigned long data)
{ unsigned int Sound_Pointer;          // MP3 File Read Data Pointer
  unsigned int i,j;
  for(j=0;j<((FileSize/512)+1);j++)    // loop Sector Count ; 1 Sector = 512 byte
  { VS1002D_XDCS_HIGH();
    SPI_CS = 0;                        // Enable SD SPI
    Read_SD(data+j); // Read Data from File Sector
    SPI_CS = 1;                        // Disable SD SPI
    VS1002D_XDCS_LOW();
    Sound_Pointer = 0;
    while(Sound_Pointer < 512)
  { while(VS1002D_Wait_DREQ_Ready());  // Wait VS1002D Ready
    for (i=0; i<1; i++)
      { VS1002D_Write_SDI(SBuf[Sound_Pointer]); // Write Data to SPI0(VS1002D)
        Sound_Pointer++;
      }
    }
  }
}

```

ผลการทดลอง คือ สามารถเขียนโปรแกรมเพื่อควบคุม MCU ให้ติดต่อสื่อสารกับ IC VLSI ซึ่งเป็น IC ถอดรหัสไฟล์ MP3 และส่งข้อมูลเสียงได้โดยการอ่านข้อมูลจาก SD Card และสามารถแสดงเสียงที่บันทึกออกสู่ลำโพงได้

5.6) การแสดงภาพและเสียงที่บันทึกโดยใช้ สวิตช์ควบคุม

ผลการทดลอง คือ สามารถสามารถเขียนโปรแกรมเพื่อควบคุม MCU ให้ติดต่อกับ IC VLSI ซึ่งเป็น IC ถอดรหัสไฟล์ MP3 และติดต่อกับจอแสดงผล GLCD Color 6610 ได้พร้อมกัน โดยสามารถเลือกแสดงภาพได้เมื่อกดปุ่ม และสามารถเล่นเสียงได้ เสียงที่ออกมาจะสอดคล้องกับข้อความภาพที่ปรากฏบนจอ GLCD 6610



บทที่ 6

บทสรุป

อุปกรณ์สื่อสารสำหรับผู้พิการทางหู

เครื่องสื่อสารสำหรับผู้พิการทางหู เครื่องนี้ได้ทำการพัฒนามาจากเครื่องรุ่นเก่าโดยอาศัยหลักการทำงานที่คล้ายๆกัน โดยเครื่องนี้สามารถพัฒนาได้ดีกว่าเครื่องรุ่นเก่ามาก เช่น

- มีการใช้งานในส่วนของ การแสดงผลจอภาพ จากเครื่องรุ่นเก่าที่เป็นจอขาวดำ 3310 เปลี่ยนมาเป็นจอสี GLCD 6100 ที่มีความละเอียด 132 x 132 pixel ซึ่งให้ความคมชัดและแสดงสีได้ดี
- มีการใช้งานไอซีสำหรับถอดรหัสไฟล์ MP3 ของ VLSI เบอร์ VS1002D ซึ่ง VS1002D จัดว่าเป็นไอซีถอดรหัสไฟล์ MP3 ที่มีคุณภาพดี และ ใช้งานง่ายมากที่สุดตัวหนึ่ง โดย VS1002D สามารถถอดรหัสไฟล์ MP3 และ WAVE ได้ทันที พร้อมทั้งมีภาค Output แบบ Analog เป็นสัญญาณเสียงแบบ Stereo
- สามารถเก็บภาพและเสียงได้เป็นจำนวนมาก เพราะมีการจัดเก็บข้อมูลไว้ในหน่วยความจำ SD Card ซึ่งไม่จำกัดความจุได้ตั้งแต่ 1Gbyte ขึ้นไป ในรูปแบบ FAT 32
- มีสวิทช์ใช้งานด้วยกัน 4 ปุ่ม ทำให้สามารถใช้งานได้สะดวกขึ้น จากเดิมซึ่งมี 3 ปุ่ม
- มีขนาดเล็กกะทัดรัดเหมาะแก่การพกพา

คุณสมบัติของอุปกรณ์สื่อสารสำหรับผู้พิการทางหู

- ใช้ไฟ DC 3.7 V จากแบตเตอรี่โทรศัพท์มือถือแต่ใช้ไฟ 3.0 V และ 5.0V ในการเลี้ยงวงจรแสดงภาพและเสียง โดยมี IC เบอร์ LP2981-3.0 จำกัดแรงดัน
- สามารถแสดงได้ 35 ภาพและ 35 เสียง (7 หมวดเมนู)
- แสดงผลโดยใช้จอสี GLCD Color6100 และลำโพง
- เก็บข้อมูลภาพและเสียงโดยใช้ SD Card
- ใช้งานได้นานติดต่อกันถึง 6 ชั่วโมง โดยไม่ปิดเครื่อง

บรรณานุกรม

1. ประจัน พลังสันติกุล การประยุกต์ใช้งานภาษา C สำหรับไมโครคอนโทรลเลอร์ AVR I
2. ประจัน พลังสันติกุล การประยุกต์ใช้งานภาษา C สำหรับไมโครคอนโทรลเลอร์ AVR II
3. Paul Stoffregen, “Mp3 Player & Fat”,
<http://www.pirc.com/tech/mp3>
4. Thaeasyelec.com, “Nokia 6100”,
<http://www.thaeasyelec.com/index.php?lay=show&ac=article&Id=526712&Ntype=1>
5. mikroe.com, “Schematic”,
http://mikroe.com/pdf/mmc_sd_board_schematic.pdf



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 SPI Buses

7.1 General

The SPI Bus - that was originally used in some Motorola devices - has been used for both VS1002d's Serial Data Interface SDI (Chapters 7.4 and 8.4) and Serial Control Interface SCI (Chapters 7.5 and 8.5).

7.2 SPI Bus Pin Descriptions

7.2.1 VS1002 Native Modes (New Mode)

These modes are active when SM_SDINEW is set to 1 (default at startup). DCLK, SDATA and BSYNC are replaced with GPIO2, GPIO3 and XDCS, respectively.

SDI Pin	SCI Pin	Description
XDCS	XCS	Active low chip select input. A high level forces the serial interface into standby mode, ending the current operation. A high level also forces serial output (SO) to high impedance state. If SM_SDISHARE is 1, pin XDCS is not used, but the signal is generated internally by inverting XCS.
SCK		Serial clock input. The serial clock is also used internally as the master clock for the register interface. SCK can be gated or continuous. In either case, the first rising clock edge after XCS has gone low marks the first bit to be written.
SI		Serial input. If a chip select is active, SI is sampled on the rising CLK edge.
-	SO	Serial output. In reads, data is shifted out on the falling SCK edge. In writes SO is at a high impedance state.

7.2.2 VS1001 Compatibility Mode

This mode is active when SM_SDINEW is set to 0. In this mode, DCLK, SDATA and BSYNC are active.

SDI Pin	SCI Pin	Description
-	XCS	Active low chip select input. A high level forces the serial interface into standby mode, ending the current operation. A high level also forces serial output (SO) to high impedance state. There is no chip select for SDI, which is always active.
BSYNC	-	SDI data is synchronized with a rising edge of BSYNC.
DCLK	SCK	Serial clock input. The serial clock is also used internally as the master clock for the register interface. SCK can be gated or continuous. In either case, the first rising clock edge after XCS has gone low marks the first bit to be written.
SDATA	SI	Serial input. SI is sampled on the rising SCK edge, if XCS is low.
-	SO	Serial output. In reads, data is shifted out on the falling SCK edge. In writes SO is at a high impedance state.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

7.3 Data Request Pin DREQ

The DREQ pin/signal is used to signal if VS1002d's FIFO is capable of receiving data. If DREQ is high, VS1002d can take at least 32 bytes of SDI data or one SCI command. When these criteria are not met, DREQ is turned low, and the sender should stop transferring new data.

Because of the 32-byte safety area, the sender may send upto 32 bytes of SDI data at a time without checking the status of DREQ, making controlling VS1002d easier for low-speed microcontrollers.

Note: DREQ may turn low or high at any time, even during a byte transmission. Thus, DREQ should only be used to decide whether to send more bytes. It should not abort a transmission that has already started.

7.4 Serial Protocol for Serial Data Interface (SDI)

7.4.1 General

The serial data interface operates in slave mode so the DCLK signal must be generated by an external circuit.

Data (SDATA signal) can be clocked in at either the rising or falling edge of DCLK (Chapter 8.6).

VS1002d assumes its data input to be byte-synchronized. SDI bytes may be transmitted either MSb or LSb first, depending of contents of SCI_MODE (Chapter 8.6.1).

The firmware is able to accept the maximum bitrate the SDI supports.

7.4.2 SDI in VS1002 Native Modes (New Mode)

In VS1002 native modes, byte synchronization is achieved by XDCS (or XCS if SM_SDISHARE is 1). The state of XDCS (or XCS) may not change while a data byte transfer is in progress. To always maintain data synchronization even if there may be glitches in the boards using VS1002d, it is recommended to turn XDCS (or XCS) every now and then, for instance once after every flash data block or a few kilobytes, just to keep sure the host and VS1002d are in sync.

If SM_SDISHARE is 1, the XDCS signal is internally generated by inverting the XCS input.

For new designs, using VS1002 native modes are recommended.

7.4.3 SDI in VS1001 Compatibility Mode

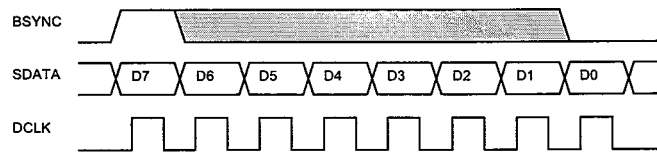


Figure 4: BSYNC Signal - one byte transfer.

When VS1002d is running in VS1001 compatibility mode, a BSYNC signal must be generated to ensure correct bit-alignment of the input bitstream. The first DCLK sampling edge (rising or falling, depending on selected polarity), during which the BSYNC is high, marks the first bit of a byte (LSB, if LSB-first order is used, MSB, if MSB-first order is used). If BSYNC is '1' when the last bit is received, the receiver stays active and next 8 bits are also received.

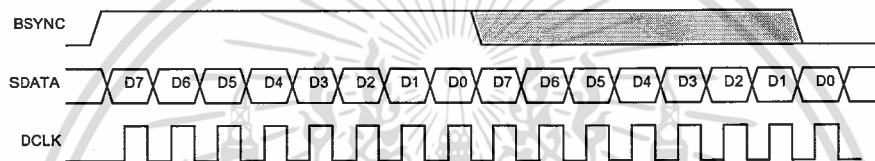


Figure 5: BSYNC Signal - two byte transfer.

7.5 Serial Protocol for Serial Command Interface (SCI)

7.5.1 General

The serial bus protocol for the Serial Command Interface SCI (Chapter 8.5) consists of an instruction byte, address byte and one 16-bit data word. Each read or write operation can read or write a single register. Data bits are read at the rising edge, so the user should update data at the falling edge. Bytes are always send MSb first.

The operation is specified by an 8-bit instruction opcode. The supported instructions are read and write. See table below.

Instruction		
Name	Opcode	Operation
READ	0b0000 0011	Read data
WRITE	0b0000 0010	Write data

Note: After sending an SCI command, it is not allowed to send SCI or SDI data for 5 microseconds.

7.5.2 SCI Read

VS1002d registers are read by the following sequence, as shown in Figure 6. First, XCS line is pulled low to select the device. Then the READ opcode (0x3) is transmitted via the SI line followed by an 8-bit word address. After the address has been read in, any further data on SI is ignored. The 16-bit data corresponding to the received address will be shifted out onto the SO line.

XCS should be driven high after data has been shifted out.

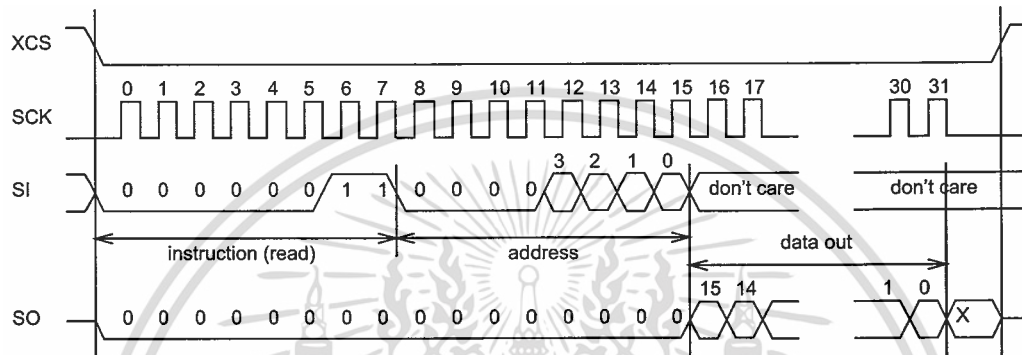


Figure 6: SCI Word Read

7.5.3 SCI Write

VS1002d registers are written to using the following sequence, as shown in Figure 7. First, XCS line is pulled low to select the device. Then the WRITE opcode (0x2) is transmitted via the SI line followed by an 8-bit word address.

After the word has been shifted in and the last clock has been sent, XCS should be pulled high to end the WRITE sequence.

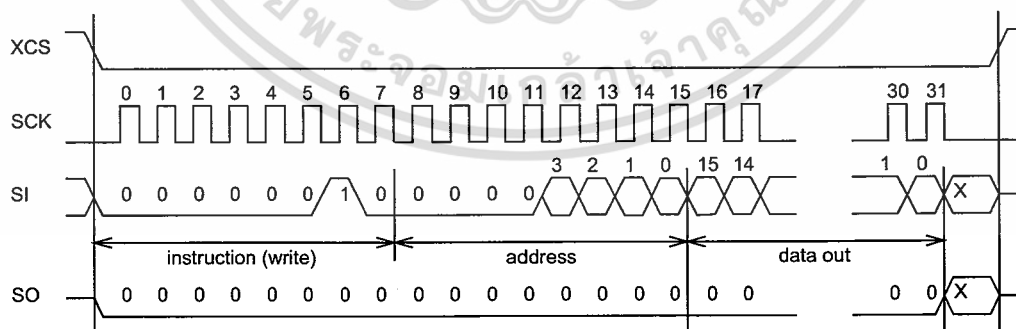


Figure 7: SCI Word Write

7.6 SPI Timing Diagram

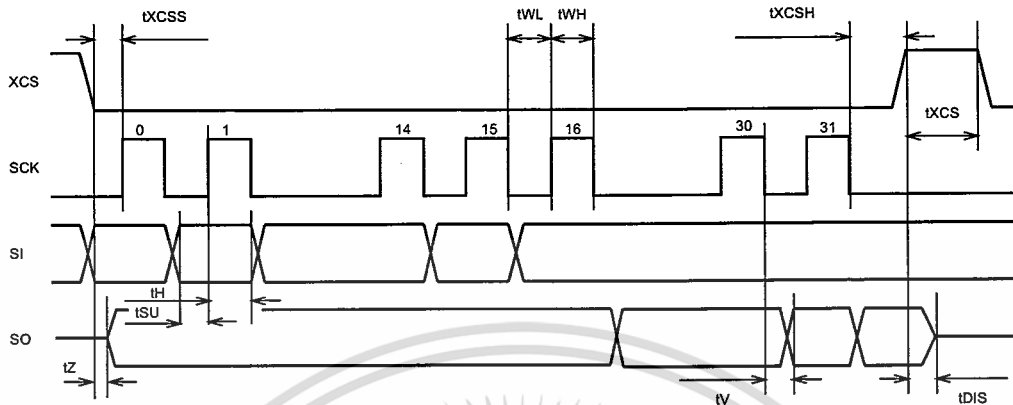


Figure 8: SPI Timing Diagram.

Symbol	Min	Max	Unit
tXCSS	5		ns
tSU	-26		ns
tH	2		XTALI cycles
tZ	0		ns
tWL	2		XTALI cycles
tWH	2		XTALI cycles
tV		2 (+ 25ns ¹)	XTALI cycles
tXCSH	-26		ns
tXCS	2		XTALI cycles
tDIS		10	ns

¹ 25ns is when pin loaded with 100pF capacitance. The time is shorter with lower capacitance.

Note: As tWL and tWH, as well as tH require at least 2 clock cycles, the maximum speed for the SPI bus that can easily be used is 1/6 of VS1011's external clock speed XTALI. Slightly higher speed can be achieved with very careful timing tuning. For details, see Application Notes for VS10XX.

Note: Negative numbers mean that the signal can change in different order from what is shown in the diagram.

7.7 SPI Examples with SM_SDINEW and SM_SDISHARED set

7.7.1 Two SCI Writes

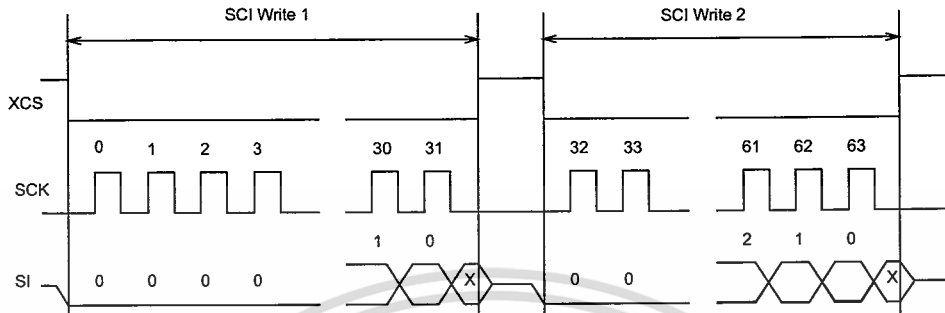


Figure 9: Two SCI Operations.

Figure 9 shows two consecutive SCI operations. Note that xCS *must* be raised to inactive state between the writes.

7.7.2 Two SDI Bytes

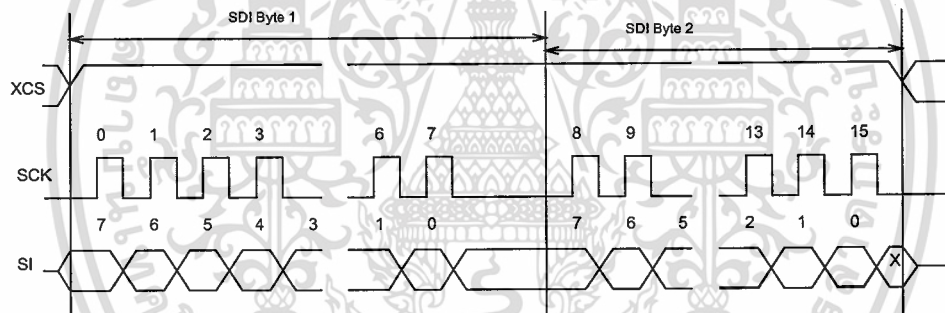


Figure 10: Two SDI Bytes.

SDI data is synchronized with a raising edge of xCS as shown in Figure 10. However, every byte doesn't need separate synchronization.

7.7.3 SCI Operation in Middle of Two SDI Bytes

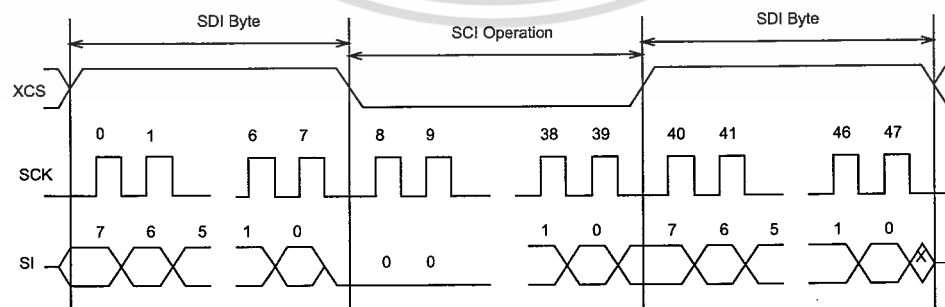


Figure 11: Two SDI Bytes Separated By an SCI Operation.

Figure 11 shows how an SCI operation is embedded in between SDI operations. The changes in xCS are used to synchronize both SDI and SCI.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

1. DESCRIPTION

S1D15G00 series are the LCD drivers equipped with the liquid crystal drive power circuit to realize color display with one chip.

S1D15G00 can be directly connected to the MPU bus to store parallel or serial gray-scale display data from MPU on the built-in RAM and to generate liquid crystal drive signals independent from MPU. S1D15G00 generates 396 segment outputs and 160^{*1} common outputs for driving liquid crystal. It incorporates the display RAM with capacity of 396 × 168 × 4 (16 gray-scale). A single dot of pixel on the liquid crystal panel corresponds to 4 bits of the built-in RAM, enabling to display 132 (RGB) × 160 pixels with one chip.

Read or write operations from MPU to the display RAM can be performed without resorting to external actuating clock signals. S1D15G00 allows you to run the display system of high performance and handy equipment at the minimum power consumption thanks to its low-power liquid crystal drive power circuit and oscillation circuit.

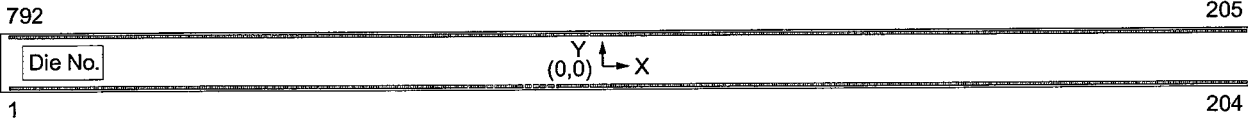
*1: The S1D15G00D10*100 generates 300 segment outputs and 120 common outputs. It incorporates the display RAM with 300 × 168 × 4 capacity and displays 100 (RGB) × 120 pixels.

2. FEATURES

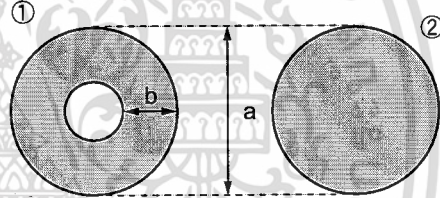
- Number of liquid crystal-drive outputs:
396 segment outputs and 160 common outputs.
- Low cross talk by frame rate modulation.
- 256 color from 4096-color display or full 4096-color display.
When 256 color from 4096-color display is selected: 8 gray-scale for red and green and 4 gray-scale for blue (intermediate tone is selected with the command).
When 4096-color display is selected: 16 gray-scale for red, green and blue.
- Direct data display with display RAM
(When the LCD is set to normally black)
RAM bit Data "0000" ... OFF (Black)
"1111" ... ON (Maximum RGB display)
(Normally black LCD, using "inverse display" command)
- Partial display function: You can save power by limiting the display space. This function is most suited for handy equipment in the standby mode.
- Display RAM : 396 × 168 × 4 = 266,112 bits.*1
*1: The S1D15G00D10*000 has RAM of 300 × 120 × 4 = 144,000 bits.
- MPU interface: S1D15G00 can be directly connected to both of the 8/16-bit parallel 80 and 68 series MPU. Two type serial interface are also available.
 - 3 pins serial : \overline{CS} , SCL and SI (D/C + 8-bit data)
 - 4 pins serial : \overline{CS} , SCL, SI and A0
- Abundant command functions: Area scroll function, automatic page & column increment function, display direction switching function and power circuit control function.
- Built-in liquid crystal drive power circuit: S1D15G00 is equipped the charge pump booster circuit, voltage follower circuit and electric volume control circuit.
- Oscillation circuit with built-in high precision CR (external clock signals acceptable)
- EEPROM interface functions
- Low current consumption
500 μ A (Conditions: S1D15G00D01B100, V_{DD} = V_{DDI} = 3.0V, frame frequency 130Hz, V₂ = 6.0V, all display RAM data is "0")
- Supply voltage
Power for input/output system power:
V_{DDI}-GND=1.7V to 3.6V
Power for internal circuit operation:
V_{DD}-GND=2.6V to 3.6V
Reference power for booster circuit:
V_{DD2}-GND=2.6V to 3.6V
Power for liquid crystal drive:
V₃-MV₃=12.0V to 21.0V
- Wider operational range: -40°C to 85°C.
- Shipping from: Chip with gold bump. COF.
- Note that the radiation resistant design or light resistance design in strict sense is not employed for S1D15G00.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

4. PIN LAYOUT



- Chip size 25.04 mm × 2.70 mm
- Chip thickness 725 μm ± 25 μm (for reference)
- Die No. See Section 5 “List of Device Models.”
- Potential on board GND
- Bump size Tolerance: bump of the shorter side ± 3 μm, bump of the longer side ± 4 μm (reference)
 Driver output side: 30 μm × 137 μm
 Driver input side: 82 μm × 109 μm
- Bump pitch Driver output side: 42 μm
 I/O signal line side: 100 μm min.
- Bump height 22.5 μm ± 4 μm (for reference) : S1D15G00D0*B0
- Alignment coordinate
 ① (-11974.2, -639.2)
 ② (12091.8, -730.4)
- Mark size a = 80 μm
 b = 20 μm



5. LIST OF DEVICE MODELS

Model name	Die No.	Output count	V ₂ voltage control resistor External/Internal	Access to EEPROM	MPU RAM read	Frame frequency /built-in oscillation frequency
S1D15G00D00*100 (#)	D15G0D0B	Segment: 396 Common: 160	Internal only (voltage electronically controlled via electronic volume)	○	Unable to read	130 Hz /41.6 kHz
S1D15G00D05*100	D15G0D5B				Read enabled	
S1D15G00D01*100 (#)	D15G0D1B		External only (voltage controlled via VR pin resistance)	×	Unable to read	180 Hz /57.6 kHz
S1D15G00D06*100	D15G0D6B				Read enabled	
S1D15G00D03*100 (#)	D15G0D3B		External only (voltage controlled via VR pin resistance)	×	Unable to read	130 Hz /31.2 kHz
S1D15G00D08*100	D15G0D8B				Read enabled	
S1D15G00D10*100 (#)	D15G0DAB	Segment: 300 Common: 120	External only (voltage via VR pin resistance)	×	Unable to read	130 Hz /31.2 kHz

(Note)
 For “unable to read” models in the above diagram, the MPU cannot read the RAM. If the RAM must be read, use “read enabled” models.
 (#) : These models will be discontinued.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า