

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์

Image Sensor Controller by Microprocessor



เลขหมู่.....
เลขทะเบียน..... 72853
วัน,เดือน,ปี 23 ส.ย. 2550

b. 11๙๖๓๓๑๕
i.

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์

Image Sensor Controller by Microprocessor



ปริญญาานิพนธ์นี้สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษาที่ 2549

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์

ผู้จัดทำ

1. นายพหล เจริญศรี รหัส 47015214
2. นายสุภาพ เจริญธรรมสิทธิ์ รหัส 47015229



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์

นายพหล เจริญศรี รหัส 47015214
 นายสุภาพ เจริญธรรมสิทธิ์ รหัส 47015229
 รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
 ปีการศึกษา 2549

บทคัดย่อ

รายงานฉบับนี้อธิบายการออกแบบและควบคุมซีมอสอิมเมจเซนเซอร์(CMOS Image Sensor) โดยการประยุกต์ใช้ไมโครโปรเซสเซอร์ที่เรียกว่า ฟิวซอค(Programmable system on chip, PSoC) อุปกรณ์ที่ใช้ออกแบบประกอบด้วย ซีมอสอิมเมจเซนเซอร์และฟิวซอค ซึ่งเราสามารถทำการโปรแกรมฟิวซอคผ่านทางไอสแควชิ(I²C) เพื่อส่งค่าพารามิเตอร์ต่างๆ เช่น ขนาดภาพ, แสง, โหมดการทำงาน ฯลฯ ซึ่งเป็นพารามิเตอร์เริ่มต้นให้ซีมอสอิมเมจเซนเซอร์ซึ่งจะทำการส่งค่าข้อมูลพิกเซลของภาพออกมา ข้อมูลของภาพจะเป็นแบบขาวดำ(Gray scale) โดยจะมีส่วนติดต่อกับคอมพิวเตอร์เพื่อให้คอมพิวเตอร์เก็บข้อมูลภาพโดยตรง จากนั้นจึงนำภาพไปประมวลผลโดยใช้ซอฟต์แวร์ต่อไป

Image Sensor Controller by Microprocessor

Mr. Pahol Charoensri ID.47015214

Mr. Suphap Charoenthammasit ID.47015229

Assoc.Prof.Dr.Chuchart Pintavirooj Advisor

Educational Year 2006

Abstract

This report describes a design and construction of a CMOS image sensor controller with PSoC (Programmable system on chip). The devices used in this construction consist of CMOS Image Sensor and PSoC. We can program the PSoC with I²C communication and send parameter such as imagesize, light, activemode etc, which is the first packet parameter send to CMOS Image Sensor. After that we acquire the pixel data and send to the computer for collect data finally the computer can apply the image other application.

กิตติกรรมประกาศ

ในการจัดทำปริญญานีพนธ์นี้ ทางผู้เขียนได้รับความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่ายด้วยกันดังนั้นทางผู้เขียนรู้สึกซาบซึ้งเป็นอย่างยิ่งจึงขอขอบพระคุณทุกท่านมา ณ โอกาสนี้

ขอขอบพระคุณ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา ที่เป็นผู้ให้ความรู้ความเข้าใจด้านต่างๆในการศึกษารวมถึงความเอื้อเฟื้อสถานที่ เครื่องมืออุปกรณ์ ตลอดจนคอมพิวเตอร์ที่ใช้ในการทดลองและค้นคว้าเกี่ยวกับโครงการ

ขอขอบพระคุณ รุ่งพี.โท และเพื่อนๆ ผู้ซึ่งให้ความรู้และให้คำแนะนำเกี่ยวกับการเขียนโปรแกรมและแนะนำอุปกรณ์ต่างๆที่น่าสนใจที่ใช้ในการทดลอง

ขอขอบพระคุณ คุณพ่อ คุณแม่และครอบครัวที่อบรมเลี้ยงดูให้กำลังใจและสนับสนุนในการศึกษามาโดยตลอด

ผู้จัดทำ

นายพหล

เจริญศรี

นายสุภาพ

เจริญธรรมสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูปภาพ	V
สารบัญตาราง	VI
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์ของโครงการ	1
1.2 ขอบเขตการดำเนินงาน	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 รายละเอียดโดยย่อของโครงการ	2
บทที่ 2 ทฤษฎี	
2.1 ไมโครคอนโทรลเลอร์ PSoC	3
2.1.1 คุณสมบัติที่สำคัญของ PSoC	4
2.2 โครงสร้างทางฮาร์ดแวร์	5
2.2.1 PSoC Core	6
2.2.1.1 วงจรกำเนิดสัญญาณนาฬิกาประกอบด้วย	7
2.2.2 Digital System	8
2.2.3 Analog System	10
2.2.4 System Resources	12
2.2.5 Port	13
2.2.5.1 ขาสัญญาณต่างๆของ PSoC ไมโครคอนโทรลเลอร์	13
2.2.6 Oscillator	15
2.2.7 Reset	16
2.3 โปรแกรม Visual Basic	16
2.3.1 เป้าหมายการเรียนรู้	17
2.3.2 เริ่มต้นกับ Visual Basic	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.3.3 การสร้าง Project ใหม่	19
2.3.4 การใช้งาน Control ต่างๆ	20
2.3.5 คำสั่งที่ใช้ในการควบคุมลำดับการทำงาน	26
2.3.6 การประยุกต์ใช้ฟังก์ชัน	29
2.3.7 โพรซีเจอร์ (Procedure)	30
2.4 ระบบภาพดิจิทัล	30
2.5 ความรู้เกี่ยวกับ Image Sensor	32
2.6 เปรียบเทียบ CCD กับ CMOS	33
บทที่ 3 หลักการทำงาน	
3.1 หลักการทำงาน	34
3.2 หลักการทำงานของชิปอิมเมจเซนเซอร์ MT9M001	34
3.3 การกำหนดคำสั่งของชิปอิมเมจเซนเซอร์ MT9M001	35
3.4 การใช้งานโมดูล I ² CHW	36
3.4.1 คุณสมบัติของโมดูล I ² CHW	37
3.4.2 การเชื่อมต่ออุปกรณ์แบบ I ² C BUS	38
3.5 การควบคุม CMOS Image Sensor	40
บทที่ 4 ผลการทดลอง	
4.1 สัญญาณที่ได้จาก CMOS Image Sensor	42
4.2 การนำสัญญาณข้อมูลติดต่อคอมพิวเตอร์ผ่านพอร์ตขนานด้วยโปรแกรม Visual Basic	44
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 บล็อกโคอะแกรมไมโครคอนโทรลเลอร์ PSoC	6
รูปที่ 2.2 PSoC Core	7
รูปที่ 2.3 Digital System	9
รูปที่ 2.4 แสดงการใช้งาน Digital Block	10
รูปที่ 2.5 Analog System	11
รูปที่ 2.6 แสดงการใช้งาน Analog Block	11
รูปที่ 2.7 System Resources	12
รูปที่ 2.8 แสดงตัวถังและการวางตำแหน่งขาสัญญาณต่างๆของไอซี CY8C27433	14
รูปที่ 2.9 วงจรกำเนิดความถี่จากภายนอก	15
รูปที่ 2.10 วงจรรีเซต	16
รูปที่ 2.11 ภาพปกติ	31
รูปที่ 2.12 ภาพที่เกิดจากการขยายแล้ว	31
รูปที่ 3.1 Block diagram แสดงหลักการทำงาน	34
รูปที่ 3.2 บล็อกโคอะแกรมการเชื่อมต่อแบบ I ² C	37
รูปที่ 3.3 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I ² C BUS	38
รูปที่ 3.4 แบบการเขียน/อ่านข้อมูลแบบ I ² C BUS	38
รูปที่ 3.5 สถานะเริ่มต้นและสถานะสิ้นสุดของ I ² C BUS	39
รูปที่ 3.6 รหัสควบคุมของ I ² C BUS	39
รูปที่ 3.7 การรับส่งบิตข้อมูลของ I ² C BUS	40
รูปที่ 4.1 สัญญาณ SDATA และ SCLK	42
รูปที่ 4.2 สัญญาณ FRAME_VALID และ LINE_VALID	43
รูปที่ 4.3 สัญญาณ LINE_VALID และ DATAOUT	44
รูปที่ 4.4 แสดงความแตกต่างของภาพที่ให้อิมเมจเซนเซอร์รับแสงกับไม่รับแสง	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ขาสัญญาณต่างๆของ PSoC	13
ตารางที่ 2.2 สัญลักษณ์และความหมายต่างๆของ Flowchart	17
ตารางที่ 2.3 ประเภทของตัวแปรใน Visual Basic ขนาดหน่วยความจำที่ใช้ และรายละเอียดของค่าที่เก็บ	18
ตารางที่ 2.4 เครื่องหมายทางการคำนวณใน Visual Basic	19
ตารางที่ 2.5 เครื่องหมายที่ใช้ในการเปรียบเทียบใน Visual Basic	19
ตารางที่ 2.6 คุณสมบัติและหน้าที่การทำงานของ Object ใน Visual Basic	25
ตารางที่ 3.1 รีจิสเตอร์ต่างๆใน CMOS Image Sensor	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันความเจริญก้าวหน้าของเทคโนโลยี ได้มีการพัฒนาให้ก้าวหน้าไปอย่างรวดเร็ว ชีวิตประจำวันส่วนใหญ่ได้สัมผัสกับระบบดิจิทัลไม่ว่าโดยตรงหรือทางอ้อม ระบบดิจิทัลที่เห็นได้ชัดเจนอย่างง่ายคือคอมพิวเตอร์ที่เราใช้งานกันอยู่ทุกวัน และการสื่อสารแบบไร้สายสมัยใหม่ไม่ว่าเป็นโทรศัพท์และอินเทอร์เน็ตมันก็จะส่งและรับสัญญาณแบบดิจิทัล โดยที่ระบบดิจิทัลในไอซีจะเป็นตัวควบคุมอุปกรณ์อิเล็กทรอนิกส์และไฟฟ้าต่างๆ ดังที่เรามักจะคุ้นกับคำว่าระบบตัวเลขหรือดิจิทัล

ระบบภาพดิจิทัลได้ถูกนำมาใช้อย่างกว้างขวางและเป็นประโยชน์อย่างมากในปัจจุบัน จึงเกิดการพัฒนาระบบภาพดิจิทัลอย่างรวดเร็ว Image Sensor เป็นส่วนสำคัญอย่างมากในระบบการเกิดภาพดิจิทัลที่ได้นำมาใช้กันอย่างกว้างขวางซึ่ง Image Sensor ที่ใช้กันอยู่ปัจจุบันมีอยู่ 2 ชนิดคือ CCD กับ CMOS ซึ่งในการประยุกต์ใช้งาน CMOS Image Sensor จะทำได้ง่ายกว่าและประหยัดการสิ้นเปลืองพลังงานน้อยกว่าและประกอบกับราคาถูกกว่า จึงเป็นที่น่าสนใจในการประยุกต์มาใช้งานร่วมกับ PSoC (Programmable System On Chip) ซึ่งเป็นไมโครคอนโทรลเลอร์ที่ใช้ต้นทุนในการออกแบบต่ำ ลดจำนวนอุปกรณ์ที่ใช้ลงได้มากและสามารถออกแบบควบคุมระบบดิจิทัลได้อย่างมีประสิทธิภาพ โดยสามารถใช้ภาษา C หรือ Assembly เขียนโปรแกรมอธิบายการทำงาน ซึ่ง PSoC สามารถโปรแกรมฟังก์ชันการทำงานที่จำเป็นสำหรับระบบไมโครคอนโทรลเลอร์

1.1 วัตถุประสงค์ของโครงการ

1. สามารถเรียนรู้หลักการการทำงานของ PSoC เพื่อนำไปใช้งาน
2. สามารถประยุกต์ใช้งาน PSoC ร่วมกับ CMOS Image Sensor
3. สามารถศึกษาและเข้าใจระบบการทำงานของ CMOS Image Sensor
4. สามารถควบคุมการออกแบบ PSoC โดยใช้ภาษา C ได้
5. สามารถนำข้อมูลภาพจาก CMOS Image Sensor นำมาจัดเก็บในคอมพิวเตอร์เพื่อใช้

ในการประยุกต์งานทางด้านประมวลผลภาพ (Image Processing)

1.2 ขอบเขตการดำเนินการ

1. ศึกษาหลักการทำงานของ CMOS Image Sensor เพื่อที่ประยุกต์ใช้งาน
2. ศึกษาและออกแบบแผ่นวงจร (PCB) โดยโปรแกรม Altium Designer 6.0
3. ศึกษาหลักการทำงานของ PSoC เพื่อประยุกต์ใช้งานร่วมกับ CMOS Image Sensor
4. ศึกษาและใช้ภาษา C ในการออกแบบบรรยายเชิงพฤติกรรมใช้ร่วมกับ PSoC เพื่อควบคุม CMOS Image Sensor เพื่อดึงภาพสัญญาณออกมา
5. ใช้ภาษา Visual Basic ในการประมวลผลสัญญาณภาพดิจิทัลให้ออกเป็นรูปภาพทางจอคอมพิวเตอร์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงวิธีการออกแบบวงจรดิจิทัลเบื้องต้น
2. เข้าใจในการประยุกต์ใช้งาน CMOS Image Sensor กับการประมวลผลภาพ
3. เข้าใจและสามารถเขียนโปรแกรมออกแบบ PSoC โดยภาษา C
4. เข้าใจและสามารถออกแบบแผ่นวงจร (PCB) โดยใช้โปรแกรม Altium Designer 6.0
5. เข้าใจและสามารถเขียนโปรแกรม Visual Basic เพื่อนำสัญญาณภาพดิจิทัลประมวลผลแสดงบนจอคอมพิวเตอร์

1.4 รายละเอียดโดยย่อของโครงการ

โครงการนี้คือการควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์ โดยใช้ไมโครโปรเซสเซอร์คือ พิชอคไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของอิมเมจเซนเซอร์ โดยการส่งค่ารีจิสเตอร์ต่างๆของอิมเมจเซนเซอร์ผ่านเส้นทางไอสแควซี นอกจากนั้นยังใช้วงจรกำเนิดสัญญาณนาฬิกาภายในพิชอคไมโครคอนโทรลเลอร์ เพื่อกำหนดความเร็วในการทำงานของอิมเมจเซนเซอร์ด้วย โดยความถี่ของสัญญาณนาฬิกาที่ใช้คือ 2 เมกกะเฮิร์ต จากนั้นอิมเมจเซนเซอร์จะส่งสัญญาณข้อมูล 10 บิตออกมา แล้วนำสัญญาณที่ได้ นั้นไปติดต่อกับคอมพิวเตอร์ โดยตรงผ่านพอร์ตขนานด้วยโปรแกรมวิชวลเบสิก

บทที่ 2

ทฤษฎี

2.1 ไมโครคอนโทรลเลอร์ PSoC

PSoC Microcontroller เป็นไมโครคอนโทรลเลอร์ของค่าย Cypress Microsystems โดยเป็นไมโครคอนโทรลเลอร์ที่มีการประมวลผลข้อมูลแบบ 8 บิต เหมือนกับไมโครคอนโทรลเลอร์ 8 บิต ทั่วไป แต่คุณสมบัติเด่นที่แตกต่างไปจากไมโครคอนโทรลเลอร์ตระกูลอื่นๆก็คือ PSoC MCU จะรวมเอาการออกแบบทั้งทางด้าน ดิจิตอลและอนาล็อก มาไว้ด้วยกันทำให้การออกแบบที่ต้องมีการยุ่งเกี่ยวกันระหว่างอุปกรณ์ทาง ดิจิตอลและอนาล็อก สามารถทำได้โดยง่ายและสะดวกสบายยิ่งขึ้น อีกทั้งยังทำให้ขนาดของการออกแบบเล็กลงอีกด้วย โดยเฉพาะวงจรทางด้านอนาล็อกซึ่งมักจะมีขนาดค่อนข้างใหญ่ อีกทั้งการทำงานของ PSoC ยังมีความคล่องตัวสูงทั้งเรื่องของแหล่งกำเนิดสัญญาณนาฬิกาที่หลากหลาย นอกจากนี้ยังมีฟังก์ชัน In-System Serial Programming (ISSP) ที่สามารถทำการโปรแกรมซอร์สโค้ดที่ได้ออกแบบลงไป ในหน่วยความจำโปรแกรม (Flash Memory) ภายในตัวชิปได้ ซึ่งจะช่วยให้การพัฒนาโปรแกรมให้กับไมโครคอนโทรลเลอร์มีความสะดวกสบายมากยิ่งขึ้น

จากรูปแบบของระบบไมโครคอนโทรลเลอร์เดิม ซึ่งสามารถรองรับการทำงานในรูปแบบเฉพาะสัญญาณทางดิจิตอล ตลอดจนการขยายขีดความสามารถที่เพิ่มขึ้นทางด้านอนาล็อกซึ่งจำเป็นต้องอาศัยทักษะความรู้และความชำนาญในการออกแบบจึงได้มีการพัฒนาชิปไมโครคอนโทรลเลอร์ขึ้นเพื่อลดปัญหาและข้อจำกัดของระบบไมโครคอนโทรลเลอร์เดิมตามคอนเซ็ปต์ที่ว่า PSoC หรือ Programmable System on Chip ซึ่งหมายถึงการโปรแกรมชิปให้มีความสามารถที่สอดคล้องกับความต้องการของระบบที่ต้องการได้นั้นเอง หรืออาจจะกล่าวอีกนัยหนึ่งได้ว่าชิป PSoC นี้สามารถโปรแกรมฟังก์ชันการทำงานที่จำเป็นสำหรับระบบไมโครคอนโทรลเลอร์ เช่น วงจรแปลงสัญญาณ ADC, DAC, PWM, Amplifier, Filter และอื่นๆอีกมากมาย ให้อยู่รวมกันภายในชิปเดียวกันได้จึงถือได้ว่าเป็นประโยชน์ต่อการพัฒนาเป็นอย่างยิ่ง เพราะนั่นหมายถึงการลดความยุ่งยากในการจัดทำวงจรอินเทอร์เฟสเพิ่มเติม จึงไม่จำเป็นต้องสิ้นเปลืองสายสัญญาณของไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์อินเทอร์เฟสต่างๆ และที่สำคัญยิ่งไปกว่านั้นก็คือต้นทุนในการผลิตสินค้าที่ลดต่ำลงตามไปด้วยนั่นเอง ณ จุดนี้เชื่อว่าไมโครคอนโทรลเลอร์ PSoC คงเป็นอีกหนึ่งไมโครคอนโทรลเลอร์สำหรับนักพัฒนาที่ควรให้ความสนใจเพื่อการนำไปประยุกต์ใช้งานต่อไปในอนาคต และจากนี้ไปผู้อ่านจะได้รู้จักกับไมโครคอนโทรลเลอร์ PSoC มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติที่สำคัญของ PSoC

จากที่กล่าวมาในช่วงแรกเป็นเพียงเกริ่นนำที่จะให้ผู้่านทุกท่าน ได้สังเกตเห็นถึงคุณสมบัติที่สำคัญของ ไมโครคอนโทรลเลอร์ PSoC ซึ่งมีคุณสมบัติที่สำคัญและโดดเด่นอยู่หลายประการดังนี้

1. มีโครงสร้างของระบบภายในแบบ Harvard Architecture ด้วยหน่วยประมวลผลภายในแบบ M8C และสามารถทำงานได้ที่ความถี่สูงสุดถึง 24 MHz
2. มีวงจรคูณเลขภายในแบบ 8 x 8 Multiply (32 Bit Accumulate)
3. สามารถทำงานได้ที่ระดับแรงดันไฟฟ้าตั้งแต่ 3.0 ถึง 5.25V
4. มีโหมดการทำงานแบบ Switch Mode Pump (SMP) ซึ่งช่วยให้ระบบสามารถทำงานกับแรงดันไฟเลี้ยงที่มีระดับต่ำถึง 1V ได้
5. สามารถนำไปประยุกต์ใช้งานภาคอุตสาหกรรมได้เป็นอย่างดี โดยสามารถทำงานได้ในช่วงอุณหภูมิ -40°C ถึง 85°C
6. วงจรกำเนิดสัญญาณนาฬิกาที่มีความเที่ยงตรงสูง โดยสามารถเลือกใช้ได้ตามความต้องการได้แก่
 - ใช้วงจรกำเนิดสัญญาณนาฬิกาจากภายใน 24/48 MHz โดยมีค่าความผิดพลาดที่ +/- 2.5% โดยไม่จำเป็นต้องใช้อุปกรณ์ภายนอกอื่นๆ เช่น R-C เป็นต้น
 - ทำงานร่วมกับ X-TAL 32.768 KHz ประกอบกับคุณสมบัติ PLL ที่มีอยู่ภายในชิพเพื่อกำเนิดสัญญาณนาฬิกาที่ 24/48 MHz
 - ทำงานร่วมกับ External Oscillator ได้ที่ความถี่สูงสุด 24 MHz
 - มีวงจรกำเนิดสัญญาณภายในอิสระ สำหรับการทำงานในโหมด Sleep และ Watch dog เพื่อเพิ่มเสถียรภาพของระบบ ไมโครคอนโทรลเลอร์
7. หน่วยความจำโปรแกรมที่มีความยืดหยุ่นสูง
 - 16 KB Flash Memory สำหรับเบอร์ CY27-Series และ 32 KB สำหรับเบอร์ CY29-Series ที่มีจำนวนรอบต่อการลบและเขียนข้อมูลใหม่ได้สูงสุด 50,000 ครั้ง
 - หน่วยความจำ RAM ภายในขนาด 256 Byte สำหรับเบอร์ CY27-Series และ 2 KB สำหรับเบอร์ CY29-Series
 - สามารถโปรแกรมข้อมูลลงสู่ชิพด้วยรูปแบบของ In – System Programming ได้โดยไม่ต้องถอดชิพออกจากบอร์ด
 - สามารถโปรแกรมเพื่ออัปเดตข้อมูลบางส่วนได้ (Partial Flash Updates)
 - หลากหลายกับโหมดการป้องกัน (Protection Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถใช้งานพื้นที่ว่างของ Flash Memory เพื่อนำมาประยุกต์ใช้เป็นหน่วยความจำชนิด EEPROM ได้ โดยไม่จำเป็นต้องอินเทอร์เฟสกับ EEPROM จากภายนอก

8. สามารถโปรแกรมฟังก์ชันการทำงานให้กับขาต่างๆของไมโครคอนโทรลเลอร์ได้

- สามารถขับกระแสได้ 25mA ทุกขาในโหมด GPIO
- สามารถเลือกรูปแบบการทำงานของแต่ละขาได้อย่างอิสระ ได้แก่ Pull Up, Pull

Down, High Z, Strong, Open Drain High และ Open Drain Low

- ทุกๆขาสัญญาณสามารถกำหนดให้เป็นแหล่งกำเนิดสัญญาณอินเทอร์รัพต์ได้

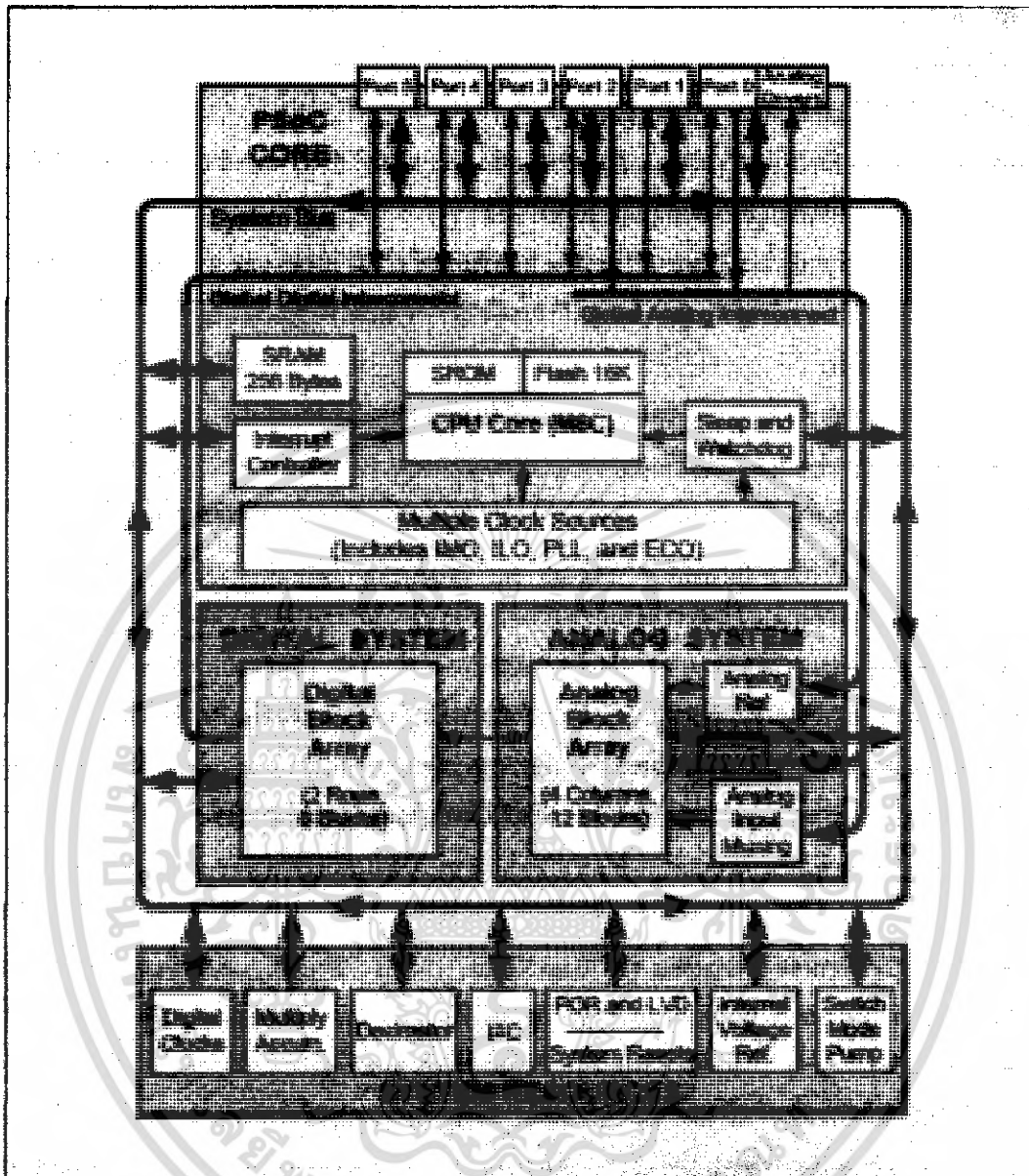
9. ทรัพยากรเพิ่มเติม ที่มีอยู่ในระบบ ได้แก่

- การสื่อสารแบบ I²C Slave, Master และ Multi – Master
- มีวงจรการทำงานของ Watchdog และ Sleep Timer
- ผู้ใช้สามารถกำหนดค่าระดับแรงดันในการทำงานได้
- มีวงจร Supervisory ประกอบอยู่ภายใน
- มีวงจรกำเนิดแรงดันอ้างอิงภายในที่มีความเที่ยงตรงสูง

10. มีซอฟต์แวร์สำหรับการพัฒนาให้ใช้งานได้ทั้งภาษา C และ Assembly ซึ่งสามารถดาวน์โหลดได้ฟรีจาก www.cypressmicro.com หากแต่การใช้งานพรินั้นจะอนุญาตให้ใช้ได้เฉพาะภาษา Assembly เท่านั้น ส่วนภาษา C นั้น จำเป็นต้องเสียค่าใช้จ่ายเพื่อปลดล็อกซอฟต์แวร์ให้สามารถใช้งานภาษา C ได้

2.2 โครงสร้างทางฮาร์ดแวร์

การศึกษาและใช้งานไมโครคอนโทรลเลอร์ให้เกิดประโยชน์และมีประสิทธิภาพสูงสุด ผู้ใช้งานจำเป็นที่จะต้องทราบถึงองค์ประกอบและความสามารถภายในตัวชิพไมโครคอนโทรลเลอร์ เพื่อที่จะสามารถนำไปประยุกต์ใช้งานได้อย่างถูกต้องและเหมาะสมต่อไป สำหรับไมโครคอนโทรลเลอร์ PSoc มีรูปแบบโครงสร้างของระบบภายในดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมไมโครคอนโทรลเลอร์ PSoc

จากภาพแสดงให้เห็นถึงองค์ประกอบโดยรวมของไมโครคอนโทรลเลอร์ PSoc ในตระกูล CY8C27XXX ซึ่งสามารถอธิบายการทำงานภายในได้ดังนี้

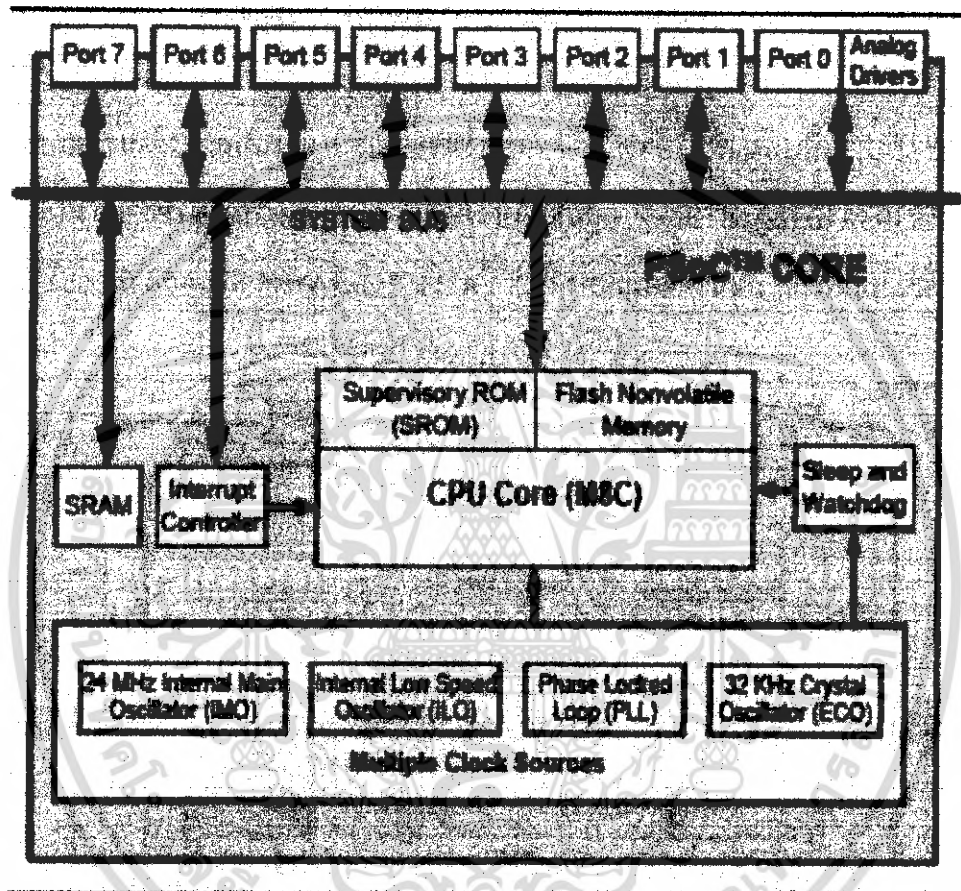
2.2.1 PSoc Core

PSoc Core เป็นส่วนของแกนหลักในการประมวลผลและควบคุมการทำงานภายในทั้งหมด อันประกอบด้วย

- หน่วยประมวลผลแบบ M8C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Flash Memory หรือหน่วยความจำโปรแกรม
- Supervisory ROM (SROM) หน่วยความจำโปรแกรมที่ทำหน้าที่ควบคุมระบบการทำงานภายในของไมโครคอนโทรลเลอร์ เช่น การ Boot ระบบ, การ Calibrate วงจรภายใน และ การจัดการข้อมูลกับ Flash Memory



รูปที่ 2.2 PSoC Core

- SRAM หน่วยความจำ RAM ภายใน สำหรับใช้ในการเก็บข้อมูลระหว่างการทำงาน ของโปรแกรมนอกจากนี้พื้นที่ของ Stack Pointer ก็เก็บไว้ในหน่วยความจำส่วนนี้ด้วยเช่นกัน

2.2.1.1 วงจรกำเนิดสัญญาณนาฬิกา ประกอบด้วย

- 24 MHz Internal Main Oscillator หรือ IMO เป็นวงจรถูกกำเนิดสัญญาณนาฬิกาที่มีความถี่ 24 MHz โดยไม่จำเป็นต้องใช้สัญญาณนาฬิกาหรือ X-TAL จากภายนอก ซึ่งถือได้ว่าเป็น ส่วนของวงจรถูกกำเนิดสัญญาณนาฬิกาที่มีบทบาทและความสำคัญมาก เพราะนอกจากการใช้งาน ต่างๆ ของไมโครคอนโทรลเลอร์ได้อย่างครบถ้วน โดยไม่จำเป็นต้องต่ออุปกรณ์ในการกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณนาฬิกาจากภายนอกแล้ว ยังสามารถกำเนิดความถี่ในการทำงาน ได้สูงถึง 24 MHz ประกอบกับค่า Error เพียง 2.5% จึงสามารถใช้งานได้อย่างหลากหลายเช่นการสื่อสารข้อมูลอนุกรม (UART) ที่ baud rate 57600 bps ได้เป็นต้น

- Internal Low Speed Oscillator หรือ ILO เป็นวงจรถูกกำเนิดสัญญาณนาฬิกาความถี่ต่ำอีกชุดหนึ่งซึ่งแยกออกมาเป็นอิสระเพื่อใช้เป็นวงจรถูกกำเนิดสัญญาณนาฬิกาให้กับการทำงานของ Watchdog และ Sleep Timer การแยกวงจรถูกกำเนิดสัญญาณนาฬิกาส่วนนี้ออกมาก็เพื่อต้องการให้มั่นใจว่าระบบจะมีเสถียรภาพที่คตินั่นเอง

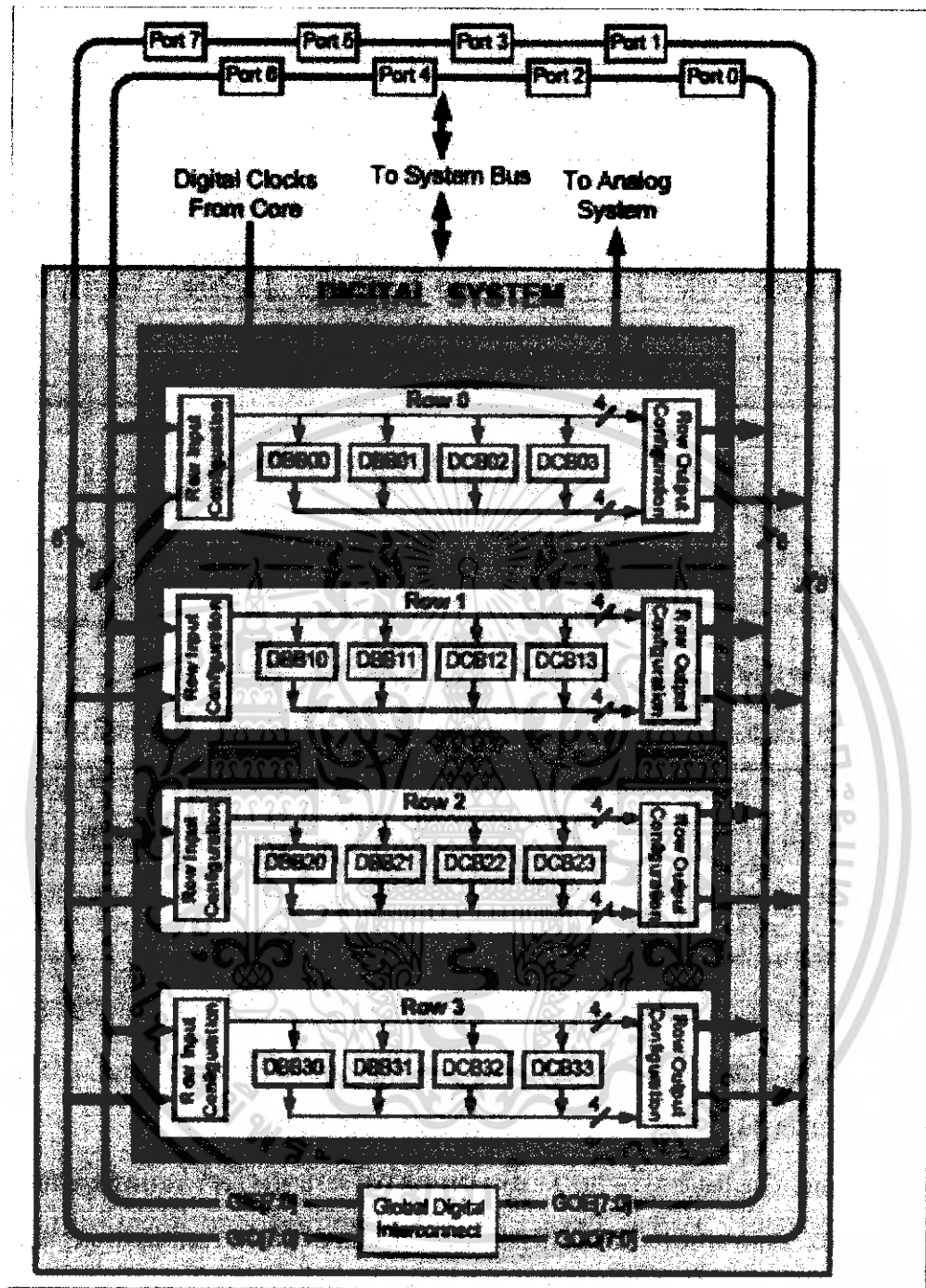
- 32 KHz Crystal Oscillator หรือ ECO เป็นโหมดการทำงานของวงจรถูกกำเนิดสัญญาณนาฬิกาซึ่งจะต้องทำงานร่วมกับ X-TAL 32.768 KHz จากภายนอก การใช้งานรูปแบบนี้ก็เพื่อต้องการให้ความถี่ของสัญญาณภายในมีค่าความเที่ยงตรงสูงสุดที่ 24 MHz ซึ่งผู้อ่านหลายท่านอาจจะสงสัยว่าเหตุใด X-TAL 32.768 KHz จึงสามารถสร้างความถี่สัญญาณนาฬิกาภายในได้สูงถึง 24 MHz เหตุที่สามารถทำงานเช่นนั้นได้ก็เพราะมีวงจร PLL (Phase Lock Loop) ประกอบอยู่ภายใน

- Sleep and Watchdog เป็นโหมดควบคุมการทำงานในแบบ Sleep และระบบตรวจสอบระบบการทำงานของระบบไมโครคอนโทรลเลอร์ด้วย Watchdog

- Interrupt Controller หน่วยควบคุมการอินเทอร์รัพต์ โดยทำหน้าที่ควบคุมการตอบสนองต่ออินเทอร์รัพต์จากแหล่งต่างๆและการจัดลำดับความสำคัญของการอินเทอร์รัพต์

2.2.2 Digital System

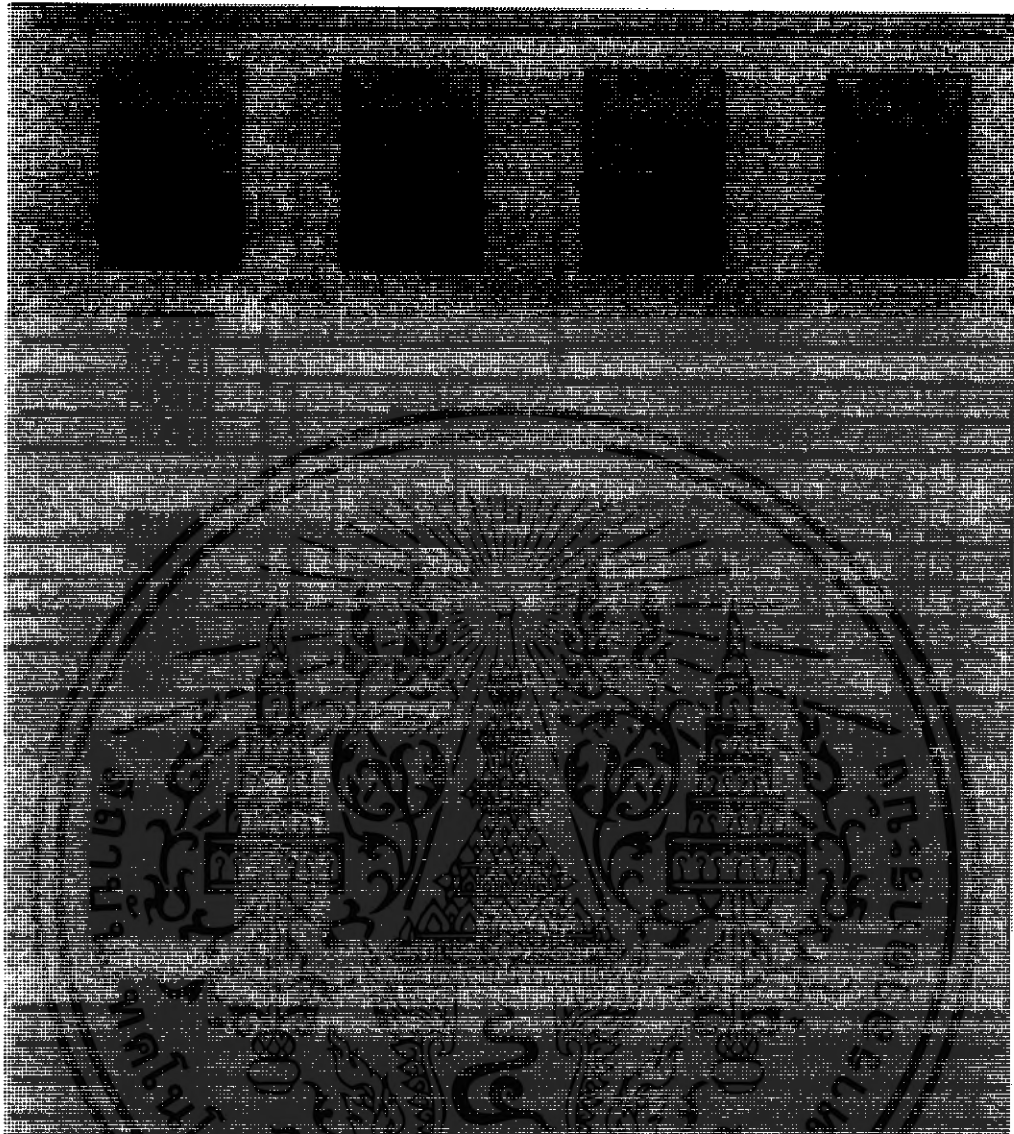
เป็นพื้นที่การทำงานของระบบดิจิทัลโดยเป็นส่วนการทำงานทาง Hardware ที่แยกเป็นอิสระจาก PSoC Core โครงสร้างส่วนนี้เองที่ผู้ใช้สามารถกำหนดคุณสมบัติฟังก์ชันทางด้านดิจิทัลลงให้แก่ชิพไมโครคอนโทรลเลอร์ได้ เช่น Timer, Counter, PWM, PC และ UART เป็นต้น เพื่อให้ชิพไมโครคอนโทรลเลอร์มีคุณสมบัติทางด้านดิจิทัลตามต้องการ สำหรับชิพรุ่น CY27-Series มีให้ใช้งานได้ 8 Digital Block และชิพรุ่น CY29-Series สามารถใช้งานได้ถึง 16 Digital Block สำหรับคุณสมบัติพื้นฐานของแต่ละบล็อก คือ มีขนาดข้อมูลเท่ากับ 8 บิตเช่น Timer ขนาด 8/16 24 บิต จะใช้พื้นที่จำนวน 1, 2 และ 3 บล็อกตามลำดับ



รูปที่ 2.3 Digital System

ดังรูปที่ 2.4 แสดงตัวอย่างการกำหนดฟังก์ชัน Timer ขนาด 8 บิต จำนวน 2 ชุด, PWM ขนาด 8 บิต จำนวน 2ชุด, Timer ขนาด 16 บิต จำนวน 1 ชุด และ Counter ขนาด 16 บิต จำนวน 1 ชุด ซึ่งแสดงให้เห็นถึงการใชพื้นที่ของ Digital Block ตามฟังก์ชันการทำงานที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

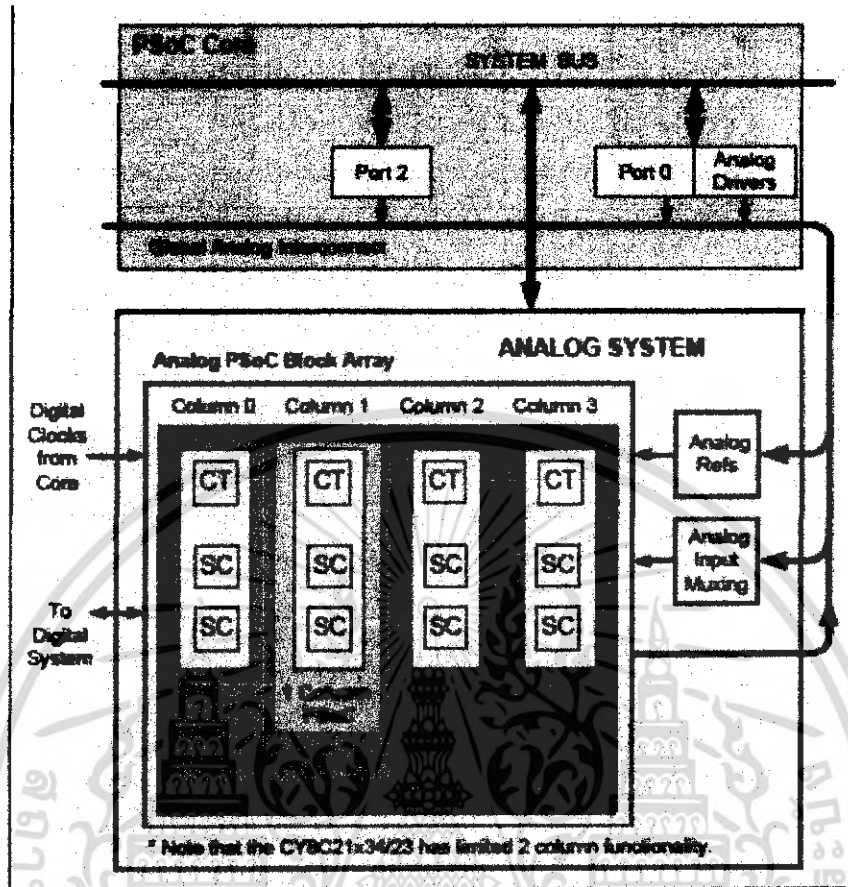


รูปที่ 2.4 แสดงการใช้รวม Digital Block.

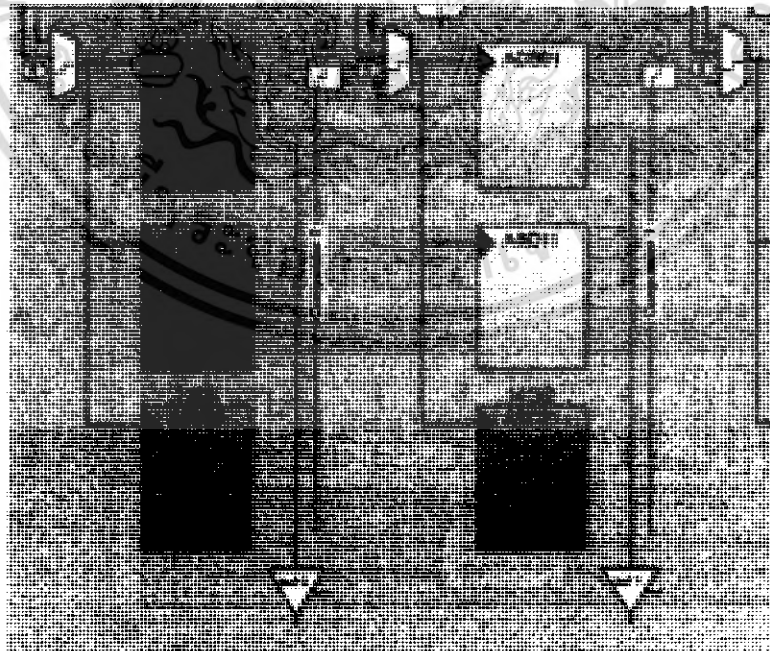
2.2.3 Analog System

เป็นพื้นที่การทำงานของระบบอนาล็อกโดยมีลักษณะเป็นส่วนการทำงานทาง Hardware ที่แยกเป็นอิสระจาก PSoC Core เช่นเดียวกับ Digital System โดย โครงสร้างในส่วนนี้ผู้ใช้สามารถกำหนดคุณสมบัติฟังก์ชันทางด้านอนาล็อกให้แก่ชิพไมโครคอนโทรลเลอร์ได้ เช่น Amplifier, ADC และ DAC เป็นต้น สำหรับชิพรุ่น CY27-Series และ CY29-Series มีพื้นที่ให้ใช้งานได้เท่ากับ 12 Analog Block, ดังรูปที่ 2.6 แสดงการกำหนดฟังก์ชันการทำงานของ Amplifier, ADC และ DAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 Analog System



รูปที่ 2.6 แสดงการใช้งาน Analog Block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 System Resources

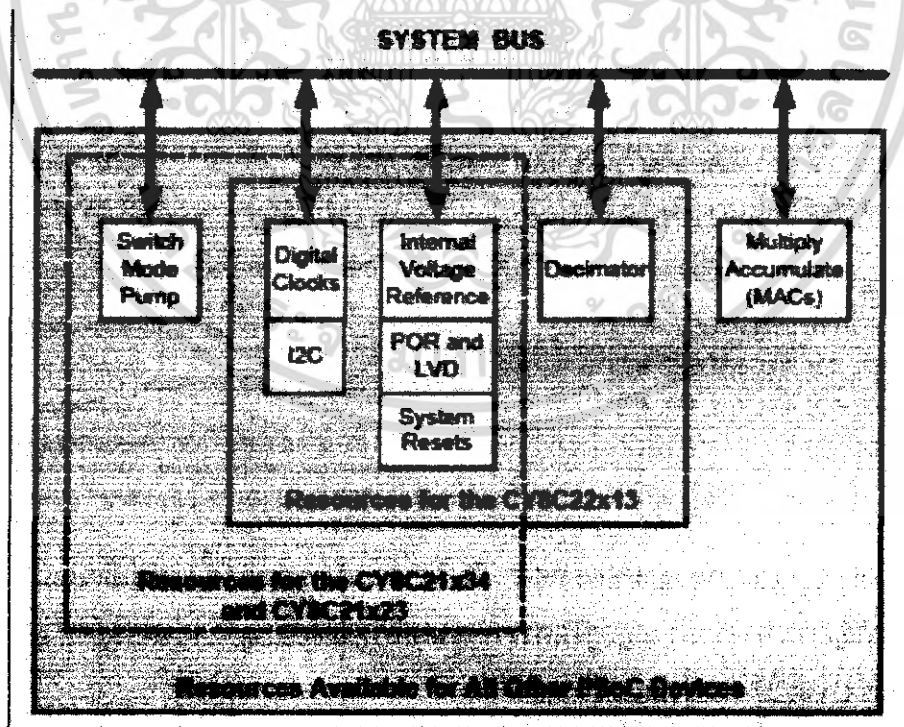
เป็นส่วนของทรัพยากรรวมภายในระบบ ซึ่งแต่ละส่วนของระบบไมโครคอนโทรลเลอร์สามารถติดต่อถึงกันได้ผ่านซิสเต็มบัส (System Bus) อันประกอบด้วย

- Digital Clocks สำหรับควบคุมการหารความถี่สัญญาณนาฬิกา
- Multiply Accumulate (MAC)
- Decimator
- I2C สำหรับการสื่อสารด้วยรูปแบบ I2C
- POR and LVD สำหรับควบคุมระบบ Reset และระบบตรวจสอบแรงดันไฟเลี้ยงต่ำกว่ากำหนด

กว่ากำหนด

- Internal Voltage Reference แรงดันอ้างอิงภายใน สำหรับกำหนดใช้งานเป็นค่าแรงดันอ้างอิงภายในให้แก่วงจร ADC หรือส่งค่าแรงดันอ้างอิงออกมาสู่ขาสัญญาณเพื่อนำออกไปใช้งานภายนอกได้

- Switch Mode Pump โหมคควบคุมการทำงานสำหรับบูตค่าแรงดันไฟเลี้ยงระดับต่ำให้มีค่าแรงดันที่สูงขึ้นและเพียงพอสำหรับการทำงานของระบบไมโครคอนโทรลเลอร์ได้สำหรับการประยุกต์ใช้งานแรงดันไฟฟ้าจากแบตเตอรี่



รูปที่ 2.7 System Resources

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 Port

เป็นขาสัญญาณต่างๆที่ใช้สำหรับการอินเตอร์เฟสไปยังวงจรต่างๆโดยจำนวนของพอร์ต จึงขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์ที่เลือกนำมาใช้งาน โดยมีขนาดตั้งแต่ 8 / 20 / 28 / 44 และ 48 ขา ให้เลือกใช้งานตามความเหมาะสมของการใช้งาน

2.2.5.1 ขาสัญญาณต่างๆของ PSoC ไมโครคอนโทรลเลอร์

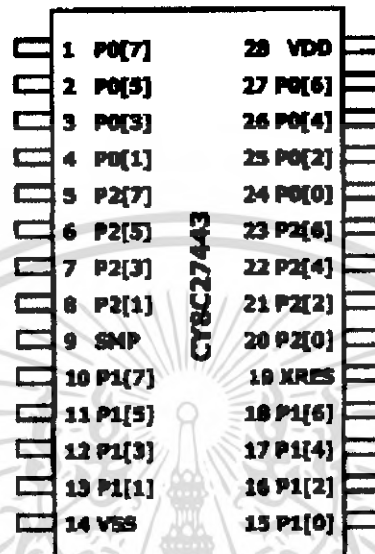
ขาสัญญาณของ PSoC มีลักษณะคล้ายกับไมโครคอนโทรลเลอร์เบอร์อื่นๆก็จะมีทั้งขาสัญญาณอินพุต เอาท์พุต ซึ่งในบางขาอาจทำหน้าที่มากกว่าหนึ่งหน้าที่ หน้าที่การทำงานของขาสัญญาณต่างๆของ PSoC สามารถสรุปได้ดังตารางต่อไปนี้

Pin Name	Description	Input/Output
SMP	Switch Mode Pump	Power
Vdd	Supply Voltage	Power
Vss	Ground	Power
XRES	External Reset (Active High)	Input
P0[0] – P0[1]	Port 0[0], 0[1], Analog Input	Input/Output
P0[2] – P0[5]	Port 0[2], 0[3], 0[4], 0[5], Analog Input/Output	Input/Output
P0[6] – P0[7]	Port 0[6], 0[7], Analog Input	Input/Output
P1[0]	Port 1[0], XTALOut#SDATA / I ² C SDA	Input/Output
P1[1]	Port 1[1], XTALIn/SCLK / I ² C SCL	Input/Output
P1[2]	Port 1[2]	Input/Output
P1[3]	Port 1[3]	Input/Output
P1[4]	Port 1[4], EXTCLK	Input/Output
P1[5]	Port 1[5], I ² C SDA	Input/Output
P1[6]	Port 1[6]	Input/Output
P1[7]	Port 1[7], I ² C SCL	Input/Output
P2[0] – P2[3]	Port 2[0], 2[1], 2[2], 2[3], Non-Multiplexed Analog Input (Switched Capacitor)	Input/Output
P2[4]	Port 2[4], External AGND	Input/Output
P2[5]	Port 2[5]	Input/Output
P2[6]	Port 2[6], External VREF	Input/Output
P2[7]	Port 2[7]	Input/Output
P3[0]-P3[7]	Port 3[0], 3[1], 3[2], 3[3], 3[4], 3[5], 3[6], 3[7]	Input/Output
P4[0]-P4[7]	Port 4[0], 4[1], 4[2], 4[3], 4[4], 4[5], 4[6], 4[7]	Input/Output
P5[0]-P5[3]	Port 5[0], 5[1], 5[2], 5[3]	Input/Output

ตารางที่ 2.1 ขาสัญญาณต่างๆของ PSoC

จากตารางข้างต้นเป็นการกำหนดขาสัญญาณมาตรฐานของ PSoC ไมโครคอนโทรลเลอร์ ทั้งเบอร์เล็กและเบอร์ใหญ่ นั้นหมายความว่าบางเบอร์อาจไม่มีขาสัญญาณบางขาปรากฏอยู่ในตาราง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเบอร์ CY8C27433 ไม่มีขาพอร์ต P3, P4 และ P5 เป็นต้น ในหนังสือเล่มนี้จะอ้างอิงเบอร์ CY8C27443 ซึ่งมีการจัดขาตั้งต่อไปนี้



รูปที่ 2.8 แสดงตัวถังและการวางตำแหน่งขาสัญญาณต่างๆของไอซี CY8C27433

ขาพอร์ตของ PSoC สามารถเป็นขาอินพุต/เอาต์พุตได้อย่างอิสระ และเป็นขาสัญญาณที่มีสถานะการทำงานได้หลายรูปแบบ เช่น pull-up, pull down, open drain เป็นต้น ซึ่งการทำงานต่างๆเหล่านี้สามารถเลือกได้ด้วยกระบวนการทางซอฟต์แวร์ นอกจากนี้การใช้งานขาพอร์ตต่างๆเป็นพอร์ตอินพุต/เอาต์พุตทั่วไปแล้ว ขาพอร์ตบางขายังมีหน้าที่เฉพาะอย่าง ดังต่อไปนี้

- VDD เป็นขาสำหรับรับไฟเลี้ยง ค่อกับไฟ 5 โวลต์
- GND เป็นขากาวด์ ค่อกับไฟ 0 โวลต์
- XRES เป็นขาจับสัญญาณรีเซ็ต เมื่อนานี้เป็นลอจิก “1” CPU จะถูกรีเซ็ต
- P0[2]-P0[5] เป็นขาสำหรับรับสัญญาณทางอนาล็อกเข้ามาภายในเพื่อทำการประมวลผล นอกจากนี้แล้วยังสามารถส่งสัญญาณอนาล็อกออกไปทางขาเหล่านี้ได้ด้วย
- P0[6]-P0[7] เป็นขาสำหรับรับสัญญาณทางอนาล็อกเข้ามาภายในเพื่อทำการประมวลผล ไม่สามารถส่งสัญญาณอนาล็อกออกไปได้
- P1[0] เป็นขา X-TAL out ใช้สำหรับค่อกับ X-TAL เพื่อสร้างสัญญาณนาฬิกาให้กับ PSoC (ใช้งานร่วมกับ P1[1]) นอกจากนี้แล้วยังถูกใช้เป็นขาสัญญาณในการรับ/ส่งข้อมูลแบบอนุกรมด้วย (ทั้ง SPI และ PC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

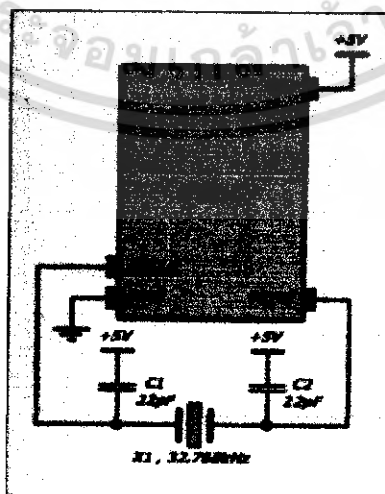
- P1[1] เป็นขา X-TAL in ใช้สำหรับต่อกับ X-TAL เพื่อสร้างสัญญาณนาฬิกาให้กับ PSoC (ใช้งานร่วมกับ P1[0]) นอกจากนี้แล้วขานี้ยังถูกใช้เป็นขาสัญญาณในการรับสัญญาณนาฬิกา สำหรับรับ/ส่งข้อมูลแบบอนุกรม (ทั้ง SPI และ I²C)

- P1[4] เป็นขารับสัญญาณนาฬิกาจากภายนอก
- P1[5] เป็นขารับ/ส่งข้อมูลของ I²C ซึ่งเรียกว่าขา SDA (Series Data)
- P1[7] เป็นขารับสัญญาณนาฬิกาในการรับ/ส่งข้อมูลของ I²C เพื่อให้ด้านส่งและด้านรับทำการรับ/ส่งข้อมูลได้อย่างสอดคล้องกัน ซึ่งเรียกขานี้ว่าขา SCL (Serial Clock)
- P2[0]-P2[3] เป็นขารับสัญญาณอนาล็อกแบบ Non-Multiplexed
- P2[6] เป็นขารับสัญญาณอ้างอิงจากภายนอก

2.2.6 Oscillator

ไมโครคอนโทรลเลอร์ PSoC สามารถทำงานได้ด้วยวงจรถูกกำเนิดความถี่ภายในได้สูงถึง 24 MHz แต่อย่างไรก็ดี การใช้วงจรมีค่าความผิดพลาด โดยปกติจะมีค่าไม่เกิน 2.5 % ดังนั้น สำหรับงานที่ต้องการค่าความเที่ยงตรงสูงกว่านี้ สามารถใช้งานวงจรถูกกำเนิดความถี่จากภายนอกซึ่งสามารถรองรับการใช้งานได้ 2 ลักษณะคือ

- Crystal Oscillator 32KHz เป็นคริสตอลชนิดเดียวกับที่ใช้กับชิพนาฬิกาหรือ RTC แม้ว่าความถี่ในการทำงานจะมีค่าเป็น 32 KHz แต่ไมโครคอนโทรลเลอร์ PSoC สามารถใช้เป็นความถี่อ้างอิงสำหรับผลิตความถี่ 24MHz ภายในไมโครคอนโทรลเลอร์ได้ โดยใช้คุณสมบัติของ PLL (Phase Lock Loop) การเลือกใช้งานวงจรถูกกำเนิดความถี่ประเภทนี้ จะต้องต่อผ่านขาของไมโครคอนโทรลเลอร์ P1[0] และ P1[1] ประกอบกับค่าคาปาซิเตอร์จากภายนอกอีก 2 ตัว



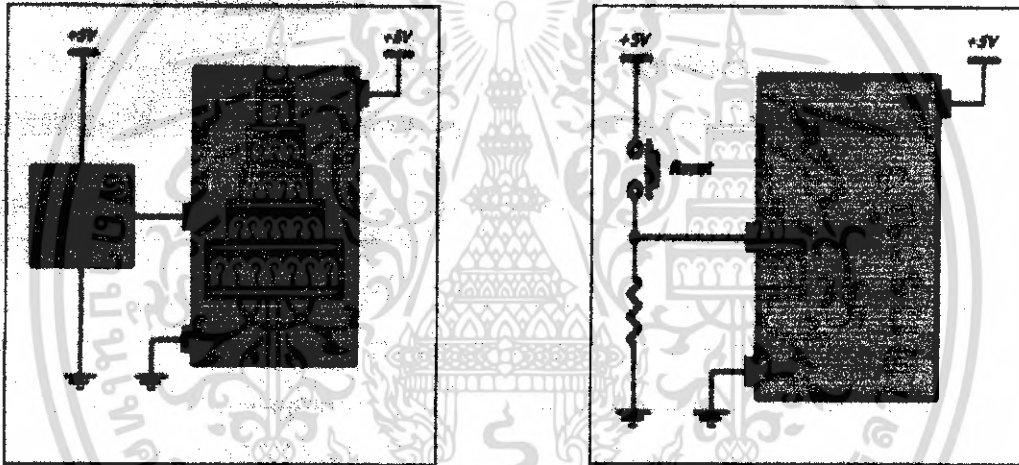
รูปที่ 2.9 วงจรถูกกำเนิดความถี่จากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Oscillator Module เป็น โมดูล กำเนิดความถี่ สามารถทำงานได้ที่ความถี่ 1MHz - 24 MHz โดยต่อผ่านทางขา P1[4] ดังแสดงในรูป ไมโครคอนโทรลเลอร์จะใช้ความถี่ที่ป้อนเข้ามาจากภายนอกนี้ เป็นความถี่อ้างอิงในการทำงาน แทนความถี่กำเนิดขึ้นจากภายใน ดังนั้นความถี่โดยตรงของค่าความถี่ในการทำงานจึงขึ้นตรงกับความถี่ของ Oscillator Module นั้นเอง

2.2.7 Reset

ขาปรับสัญญาณรีเซ็ตของไมโครคอนโทรลเลอร์ PSoC หรือขา XRES จะมีอยู่ในคอนโทรลเลอร์ทุกรุ่น ยกเว้นรุ่นที่มีขนาด 8 ขา ซึ่งจะไม่ขาสัญญาณนี้ การรีเซ็ตจะเกิดขึ้นเมื่อได้รับลอจิก "1" หรือแรงดันไฟ +5V ดังนั้นวงจรรีเซ็ตจึงสามารถแสดงได้ดังรูป



รูปที่ 2.10 วงจรรีเซ็ต

2.3 โปรแกรม Visual Basic

โปรแกรม Visual Basic คือ โปรแกรมที่ใช้หลักของภาพและการมองเห็นเป็นหลัก แต่เดิมการเขียนโปรแกรม จะเขียนคำสั่งแบบเป็นลำดับโครงสร้าง เรียกว่า Structure Program แต่ในปัจจุบัน การเขียนโปรแกรมจะเป็นแบบ OOP (Object Oriented Programming) คือการทำงานขึ้นอยู่กับวัตถุและเหตุการณ์ที่เกิดขึ้น (Event Driven) เนื้อหาจะเป็นหลักสูตรเริ่มต้นในการฝึกเขียนโปรแกรมคอมพิวเตอร์ และมีตัวอย่างโปรแกรมคำนวณงานต่าง ๆ เช่น การหาค่าเฉลี่ย หาค่าครน. การเขียนเกมส์ OX และเขียนโปรแกรมจัดการงานแต่ละชิ้น ซึ่งจะช่วยให้ผู้ฝึกหัด มีกระบวนการคิดงาน อย่างเป็นขั้นเป็นตอน และสามารถนำ โปรแกรม Visual Basic ไปประยุกต์ใช้กับงานอื่น ๆ ต่อไปได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3.1 เป้าหมายการเรียนรู้

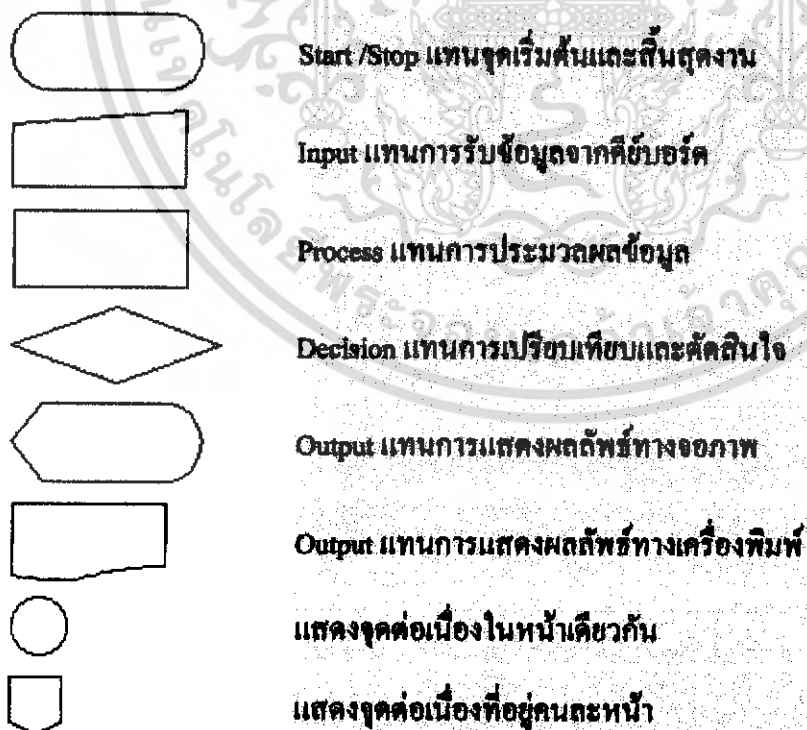
1. สามารถคิดและวางแผนล่วงหน้าเกี่ยวกับการทำงานต่าง ๆ ได้
2. สามารถนำแนวคิดนั้นมาเขียนเป็น Flowchart ได้
3. สามารถนำ Flowchart มาเขียนเป็น โปรแกรมคอมพิวเตอร์ได้
4. เพื่อให้เรียนรู้หลักของการเขียน โปรแกรมแบบ Object
5. เพื่อให้เรียนรู้ control ต่าง ๆ เพื่อใช้ในการเขียน โปรแกรมได้
6. เพื่อให้เรียนรู้คำสั่งต่าง ๆ เพื่อใช้ในการเขียน โปรแกรมได้
7. เพื่อให้เขียน โปรแกรมคำนวณหาค่า ต่าง ๆ โดยใช้โปรแกรม Visual Basic ได้

- Algorithms

Algorithms คือ แนวคิดหรือแผนการทำงานสิ่งใดสิ่งหนึ่ง หรือแผนการสั่งงานให้คอมพิวเตอร์ทำงาน ตามที่เราต้องการ โดยการเขียนแผนงานออกมาเป็นลำดับขั้นตอน

- Flowchart

Flowchart หรือผังงาน คือแผนภาพใช้สำหรับอธิบายขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์ โดยอาศัย สัญลักษณ์และข้อความสั้น ๆ สื่อความหมาย



ตารางที่ 2.2 สัญลักษณ์และความหมายต่างๆของ Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก 72853 การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 เริ่มต้นกับ Visual Basic

โปรแกรม Visual Basic คือ โปรแกรมที่ใช้หลักของภาพและการมองเห็นเป็นหลัก แต่เดิมการเขียนโปรแกรม จะเขียนคำสั่งแบบเป็นลำดับโครงสร้าง เรียกว่า Structure Programming แต่ในปัจจุบัน การเขียนโปรแกรมจะเป็นแบบ OOP (Object Oriented Programming) คือการทำงานขึ้นอยู่กับวัตถุและเหตุการณ์ที่เกิดขึ้น (Event Driven) การเริ่มเขียนโปรแกรม เริ่มจากออกแบบ Windows ย่อย ที่เรียกว่า ฟอรัม ในฟอรัมจะประกอบด้วยสิ่งต่าง ๆ ที่เราจะทำงานด้วยที่เรียกว่า Object เช่น Text , Label เมื่อกำหนดสิ่งเหล่านี้ครบแล้ว จึงระบุว่าจะประกอบแต่ละอย่างจะทำงานอย่างไร โดยการเขียนโปรแกรมย่อย ๆ ปะเข้าไปกับ Object เหล่านี้

การกำหนดตัวแปรคือ การแจ้งให้โปรแกรมทราบว่าใช้ตัวแปรชื่อว่าอะไรเป็นตัวแทน ในการคำนวณค่าต่าง ๆ และเป็นตัวแปรประเภทใดๆมีกฎเกณฑ์ดังนี้

1. ต้องขึ้นต้นด้วยตัวอักษรเสมอ
2. ห้ามเว้นวรรค ถ้าต้องการเว้นวรรคให้ใช้เครื่องหมาย _ (Under Score) เชื่อม
3. สามารถใช้ตัวอักษรตัวเล็กตัวใหญ่ผสมกันได้
4. จะต้องไม่ซ้ำกับคำสั่งหรือฟังก์ชันต่าง ๆ เช่น คำว่า IF , Case , Do เป็นต้น

- รูปแบบ Dim ชื่อตัวแปร As ประเภท
เช่น Dim Number As Integer

ประเภท	ขนาด	ขอบเขตตัวแปร
Integer	2 ไบต์	-32,768 ถึง 32,767
Long	4 ไบต์	-2,147,483,648 ถึง 2,147,483,647
Single	4 ไบต์	-3.4E38 ถึง 3.4E38
Double	8 ไบต์	-1.7E308 ถึง 1.7E308
String	1 ไบต์	0 ถึง 65535
Variant	ตามเหมาะสม	อาจจะเป็นDate/Time หรือ Integer ,String

ตารางที่ 2.3 ประเภทของตัวแปรใน Visual Basic ขนาดหน่วยความจำที่ใช้และรายละเอียดของค่าที่เก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องหมายทางการคำนวณ

เครื่องหมาย	การใช้งาน
=	กำหนดค่าให้กับตัวแปร
+	ใช้ในการบวก
-	ใช้ในการลบ
*	ใช้ในการคูณ
/	ใช้ในการหาร
\	ใช้ในการหารพิเศษทั้ง
Mod	หารได้ค่าเศษ

ตารางที่ 2.4 เครื่องหมายทางการคำนวณใน Visual Basic

- เครื่องหมายที่ใช้ในการเปรียบเทียบ

=	เท่ากับ
>	ไม่เท่ากับ
<	น้อยกว่า
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ

ตารางที่ 2.5 เครื่องหมายที่ใช้ในการเปรียบเทียบใน Visual Basic

2.3.3 การสร้าง Project ใหม่

1. คลิกคำสั่ง File / New Project

- Standard EXE ใช้สำหรับสร้างฟอร์มแบบธรรมดาทั่วไป
- ActiveX EXE ใช้สำหรับสร้างโปรแกรมติดต่อกับโปรแกรมอื่นในรูปแบบของ OLE Automation Server
- ActiveX DLL เหมือนกับ ActiveX EXE แต่จะเก็บไว้ในรูปแบบไฟล์นามสกุล .dll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ActiveX Control ไว้สำหรับสร้าง Control ใหม่ขึ้นใช้งานเอง
 - VB Application Wizard สำหรับสร้างฟอร์มอัตโนมัติ ตาม Database Structure
2. เลือก Standard EXE
 3. คลิกปุ่ม OK

2.3.4 การใช้งาน Control ต่างๆ

- Pointer ใช้เลือกวัตถุที่ต้องการ
- Label Control ใช้แสดงข้อความบนจอภาพหรือฟอร์ม

วิธีใช้

1. คลิกที่ Label Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties
 - Name ให้ชื่อวัตถุว่าอะไร
 - Caption ข้อความที่จะแสดงบน Form คำว่าอะไร
 - Font กำหนด Font และขนาดของข้อความ
 - ForeColor กำหนดสีตัวอักษร

- TextBox Control ใช้รับข้อความทางแป้นพิมพ์

วิธีใช้

1. คลิกที่ TextBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties
 - Name ให้ชื่อวัตถุว่าอะไร
 - Text ข้อความที่จะปรากฏบนวัตถุ Textbox
 - Font กำหนด Font และขนาดของข้อความ
 - ForeColor กำหนดสีตัวอักษร
 - TabIndex ลำดับการกดปุ่ม Tab
 - Alignment กำหนดให้ตำแหน่งของข้อความอยู่ด้านไหน ซ้าย , กลาง , ขวา
 - Multiline กำหนดแสดงข้อความได้หลาย ๆ บรรทัด ต้องให้เป็น True
จึงจัดตำแหน่งของข้อความได้
 - PasswordChar ใช้แสดงข้อความตามตัวอักษรที่กำหนดจากที่นี่

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- **CommandButton Control** ใช้รับคำสั่งจากผู้ใช้

วิธีใช้

1. คลิกที่ **CommandButton Control**
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด **Properties**

Name ให้ชื่อวัตถุว่าจะไร

Caption ข้อความที่จะปรากฏบนวัตถุ **CommandButton**

Font กำหนด Font และขนาดของข้อความ

TabIndex ลำดับการกดปุ่ม Tab

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- **ListBox Control** ใช้แสดงรายการต่าง ๆ ที่มีหลาย ๆ รายการ เช่น แสดงรายการสินค้า

วิธีใช้

1. คลิกที่ **ListBox Control**
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด **Properties**

Name ให้ชื่อวัตถุว่าจะไร

List รายการต่าง ๆ ที่จะปรากฏบนวัตถุ **ListBox**

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- **PictureBox Control** ใช้แสดงรูปภาพบนจอภาพ หรือ ฟอรัม

วิธีใช้

1. คลิกที่ **PictureBox Control**
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด **Properties**

Name ให้ชื่อวัตถุว่าจะไร

Picture กำหนดชื่อ ไฟล์รูปภาพ

MouseIcon สัญลักษณ์ของเมาส์เมื่อเลื่อนไปที่วัตถุนั้น ๆ

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Image Control ใช้แสดงรูปภาพบนจอภาพ หรือ ฟอรัม
วิธีใช้

1. คลิกที่ Image Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Picture กำหนดชื่อไฟล์รูปภาพ

Stretch กำหนดการยืดขยายรูปภาพ ตามกรอบของวัตถุ

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- Frame Control ใช้สร้างกรอบพื้นที่ เพื่อกำหนดรายการย่อยให้เลือก
วิธีใช้

1. คลิกที่ Frame Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Caption ข้อความอธิบายกลุ่มวัตถุ จะปรากฏบน Frame

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

ToolTipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- Option Button Control ใช้กำหนดปุ่มขึ้นมาสำหรับเลือกเอาอย่างใดอย่างหนึ่ง
วิธีใช้

1. คลิกที่ OptionButton Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Caption ข้อความอธิบายการเลือกวัตถุ จะปรากฏใกล้ ๆ Option Button

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

Value กำหนดให้เลือกวัตถุหรือไม่

- CheckBox Control ใช้กำหนดปุ่มขึ้นมาสำหรับเลือกได้หลาย ๆ รายการ หรือไม่เลือกรายการใด ๆ เลย

วิธีใช้

1. คลิกที่ CheckBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Caption ข้อความอธิบายการเลือกวัตถุ จะปรากฏใกล้ ๆ CheckBox

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

Value กำหนดให้เลือกวัตถุหรือไม่

- ComboBox Control ใช้แสดงรายการต่าง ๆ ที่มีหลายรายการ แล้วเลือกเอารายการใดรายการหนึ่ง

วิธีใช้

1. คลิกที่ ComboBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Text ข้อความที่จะปรากฏบนวัตถุ ComboBox

List รายการต่าง ๆ ที่จะมีให้เลือก

Font กำหนด Font และขนาดของข้อความ

- DriveListBox Control ใช้แสดงไดรฟ์ต่าง ๆ เพื่อให้ผู้ใช้เลือก

วิธีใช้

1. คลิกที่ DriveListBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

TabIndex ลำดับการกดปุ่ม Tab

TooltipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- DirListBox Control ใช้แสดงโฟลเดอร์ต่าง ๆ เพื่อให้ผู้ใช้เลือก

วิธีใช้

1. คลิกที่ DirListBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

TabIndex ลำดับการกดปุ่ม Tab

TooltipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- FileListBox Control ใช้แสดงไฟล์ต่าง ๆ เพื่อให้ผู้ใช้เลือก

วิธีใช้

1. คลิกที่ FileListBox Control
2. นำมาวางบน Form ปรับขนาดตามความต้องการ
3. กำหนด Properties

Name ให้ชื่อวัตถุว่าอะไร

Font กำหนด Font และขนาดของข้อความ

ForeColor กำหนดสีตัวอักษร

Pattern กำหนดชนิดของไฟล์ที่จะแสดง เช่น *.jpg ; *.bmp แสดง Jpeg File

และ Bitmap

TabIndex ลำดับการกดปุ่ม Tab

TooltipText ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ

- คุณสมบัติ (Property) เป็นชื่อของแอตทริบิวต์ที่ใช้ในการกำหนดคุณลักษณะของ Object

Property	หน้าที่
Name	กำหนดชื่อของ Control
Alignment	กำหนดตำแหน่งของข้อความอยู่ด้านไหน ซ้าย , กลาง , ขวา
Multiline	กำหนดแสดงข้อความได้หลาย ๆ บรรทัด ต้องให้เป็น True จึงจัดตำแหน่งของข้อความได้
BackColor	กำหนดสีพื้น
Caption	กำหนดข้อความสำหรับแสดงผลบนคอนโทรล
Enabled	กำหนดให้คอนโทรล สามารถตอบสนองหรือไม่ตอบสนองต่อเหตุการณ์
ForeColor	กำหนดสีตัวอักษร
Icon	กำหนดรูป Icon ให้กับ Project
Font	กำหนด Font ของตัวอักษร
Picture	กำหนดการแสดงรูปภาพ
Height,Width,Left,Top	กำหนดความสูง,กว้างและตำแหน่ง (Left,Top) การวางคอนโทรล
TabIndex	กำหนดลำดับการรับโฟกัสของคอนโทรล
TooltipText	ให้ปรากฏข้อความเมื่อเลื่อนเมาส์ไปที่วัตถุนั้น ๆ
Visible	กำหนดให้มองเห็นหรือถูกซ่อน
WindowState	กำหนดสถานะการแสดงผล Windows

ตารางที่ 2.6 คุณสมบัติและหน้าที่การทำงานของ Object ใน Visual Basic

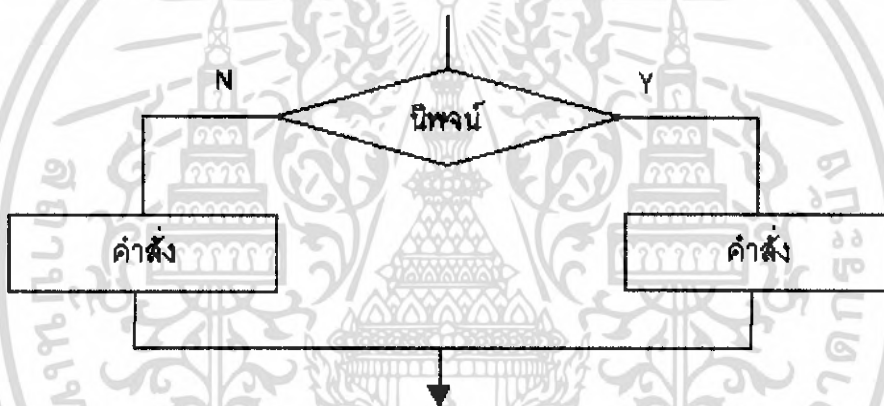
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 คำสั่งที่ใช้ในการควบคุมลำดับการทำงาน

- คำสั่ง IF Then Else คือ รูปแบบการตัดสินใจ เพื่อเลือกทำงานอย่างใดอย่างหนึ่ง จากสองตัวเลือก

รูปแบบ

```
if (นิพจน์) then
คำสั่ง
Else
คำสั่ง
Endif
```



โดยที่นิพจน์ : เป็นเงื่อนไขในการตรวจสอบ ถ้าผลลัพธ์เป็น True ก็จะทำคำสั่งหลัง Then ถ้าเป็น False ก็จะทำคำสั่งหลัง else

- คำสั่ง Select Case คือ การตัดสินใจเลือกตัวเลือกเดียวจากหลายๆ ตัวเลือก

รูปแบบ

```
Select Case (นิพจน์)
```

```
Case ค่านิพจน์
```

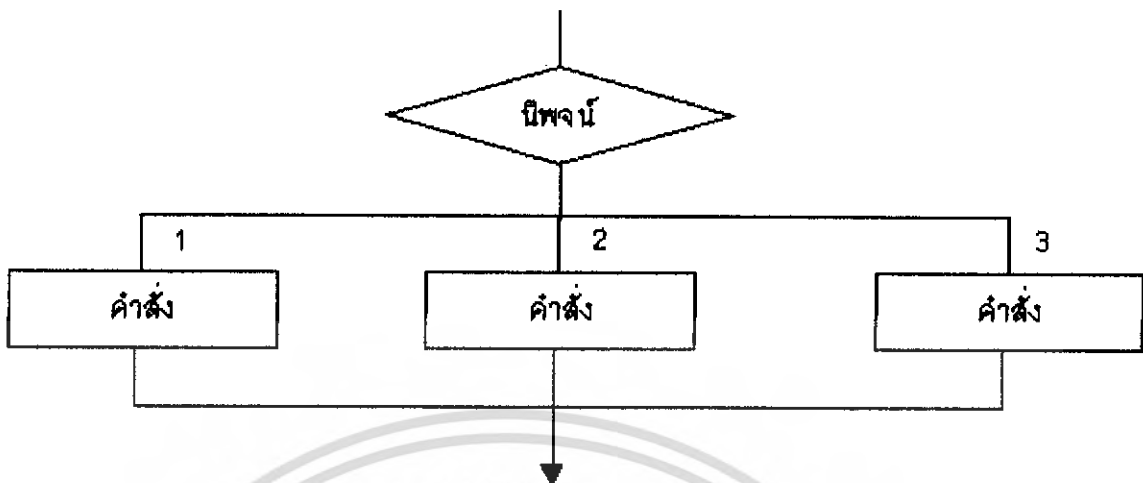
```
คำสั่ง ...
```

```
Case Else
```

```
คำสั่ง
```

```
End Select
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- คำสั่ง Do While เป็นคำสั่งวนซ้ำเมื่อเงื่อนไขเป็นจริง (ทำเมื่อเป็นจริง)

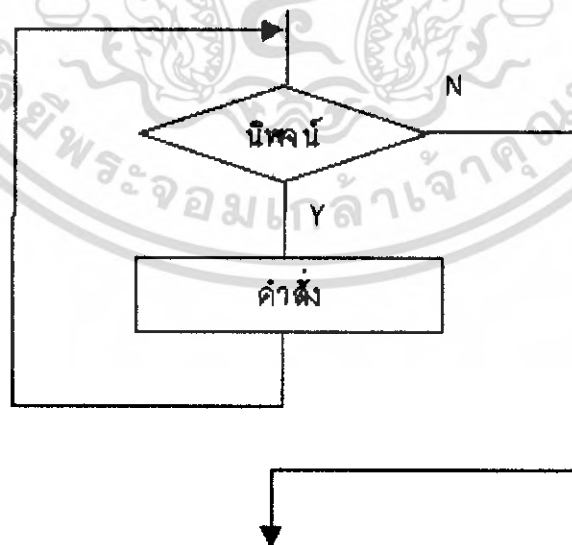
รูปแบบ

Do While (นิพจน์)

คำสั่ง

:

Loop



โดยที่นิพจน์ คือ ค่าที่จะตัดสินใจในการวนloop ถ้าผลลัพธ์เป็น True ก็จะทำคำสั่งในloop

While ถ้าเป็น False ก็จะจบคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่ง For Next เป็นรูปแบบการทำงานซ้ำ ๆ กันไปโดยมีจำนวนครั้งที่แน่นอน โดยอาศัยตัวแปร ตัวหนึ่ง เป็นตัวกำหนดตำแหน่งสิ้นสุดของการวนซ้ำว่าหมดหรือยัง
- รูปแบบ

```
For counter = start To end [Step step]
```

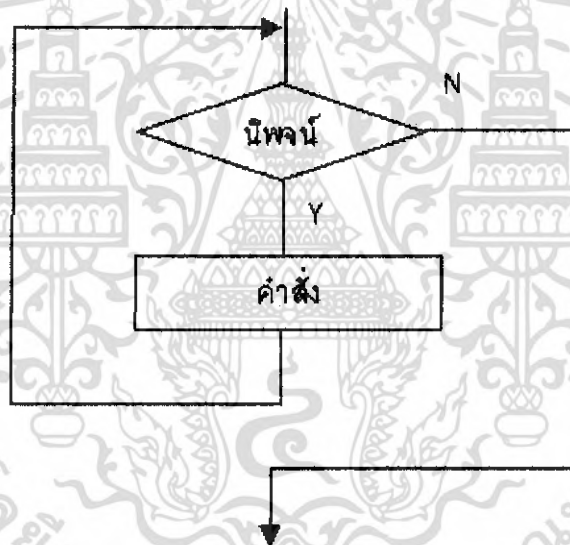
```
    คำสั่ง
```

```
Next [counter]
```

<counter> คือ ตัวแปรที่เก็บค่าตัวเลข

<start> ค่าเริ่มต้นของตัวแปร ก่อนจะเข้าไปในลูป (loop)

<end> ค่าสุดท้ายที่จะจบการทำงานออกจากลูป



- คำสั่ง Do Until เป็นคำสั่งวนซ้ำเมื่อเงื่อนไขเป็นเท็จ (ทำงานกระทั่งเป็นจริง)

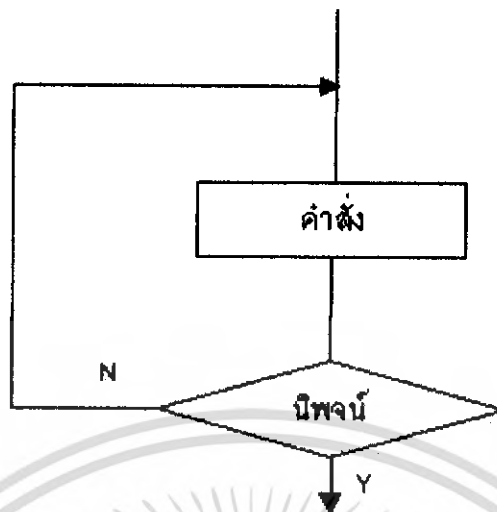
รูปแบบ

```
Do
```

```
    คำสั่ง
```

```
Loop Until (นิพจน์)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โดยที่นิพจน์ คือ เงื่อนไขซึ่งจะใช้ในการตัดสินใจว่าจะทำงานต่อหรือไม่

2.3.6 การประยุกต์ใช้ฟังก์ชัน

- **InputBox** คือ ฟังก์ชันที่ Prompt ขึ้นมา ให้ผู้ใช้พิมพ์ข้อความเข้าไป
รูปแบบ

`InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])`

เช่น

`Label1.Caption = Inputbox("Name ", "Please Enter Name ", "Srilai Ruangchai")`

- **MsgBox Statement** คือ ฟังก์ชันที่ใช้แสดง Message Prompt ขึ้นมา แจ้งข่าวสารต่าง ๆ
รูปแบบ

`MsgBox prompt[, buttons] [, title] [, helpfile, context]`

เช่น

`Msgbox "ขอกล่าวคำว่าสวัสดี", vbokcancel, "สวัสดี "`

- **MsgBox Function** คือ การรับค่าจากฟังก์ชันที่ใช้แสดง Message Prompt ขึ้นมา
รูปแบบ

`ตัวแปรรับค่า = MsgBox(prompt[, buttons] [, title] [, helpfile, context])`

เช่น

`Ans = MsgBox("ขอกล่าวคำว่าสวัสดี", vbokcancel, "สวัสดี ")`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 โพรซีเจอร์(Procedure)

โพรซีเจอร์หมายถึงโปรแกรมย่อยซึ่งส่วนอื่นๆสามารถเรียกใช้ได้
รูปแบบ

```
Sub <ชื่อโปรแกรมย่อย> (<รายการของตัวรับส่งค่า>)  
    คำสั่ง  
end Sub
```

- ขอบเขตการใช้งานตัวแปร

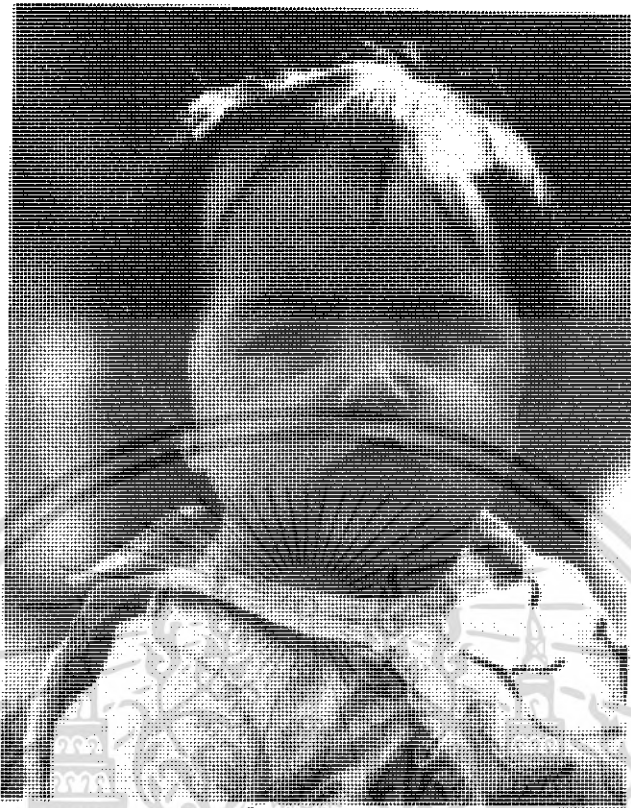
ตัวแปรโลคอล (Local) มีขอบเขตอยู่ในระดับ Procedure หรือ function ที่ประกาศใน procedure นั้น ๆ

ตัวแปรโมดูล (Module) มีขอบเขตใช้ได้หลาย Procedure หรือ function ในฟอร์ม นั้น ให้ประกาศในส่วน Declaration ของโมดูล

ตัวแปรโกลบอล (Global) ใช้ในส่วนใด ๆ ของโปรเจกต์ก็ได้ให้ประกาศในส่วน Module เช่น public n as integer

2.4 ระบบภาพดิจิทัล

กล้องดิจิทัลใช้ตัวรับภาพในการบันทึกภาพแทนฟิล์ม ซึ่งบนตัวรับภาพจะประกอบด้วย ตารางสี่เหลี่ยมเล็กๆหลายแสนล้านชิ้นเรียกว่า พิกเซล(pixel) หนึ่งตารางสี่เหลี่ยมจะมีสีเพียงสีเดียวเท่านั้น ตารางสี่เหลี่ยมที่ประกอบรวมกันเป็นภาพดิจิทัลภาพหนึ่งนั้นมีขนาดเล็กมากๆ เราจึงดูไม่ออกว่าภาพเหล่านั้นที่จริงแล้วคือการเรียงต่อกันของตารางสี หากเราเปิดภาพดิจิทัลในเครื่องคอมพิวเตอร์แล้วใช้แว่นขยายส่องดูจะเห็นตารางสี่เหลี่ยมได้อย่างชัดเจน การเรียงตัวกันอย่างเป็นระเบียบของตารางสีเหล่านี้เป็นเสมือนแผนที่ของตารางสี ซึ่งเป็นที่มาของชื่อภาพประเภทนี้ซึ่งเรียกว่าเป็นภาพ บิตแมพ(Bit Map)



รูปที่ 2.11 ภาพปกนี้



รูปที่ 2.12 ภาพที่เกิดจากการขยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณภาพของภาพดิจิทัลไม่ว่าจะเป็นภาพที่นำไปอัด หรือแสดงบนจอมอนิเตอร์จะขึ้นอยู่กับข้อมูลของภาพที่มีอยู่ หรืออีกนัยหนึ่งก็คือขึ้นอยู่กับจำนวนของพิกเซลที่มีอยู่บนตัวรับภาพ จำนวนพิกเซลที่มากขึ้นหมายถึงรายละเอียดของข้อมูลภาพที่สูงขึ้น ซึ่งจะมีส่วนช่วยให้ได้ภาพถ่ายที่มีคุณภาพดีขึ้นไปด้วย (เมื่อนำไปอัด - ขยาย) เราวัดจำนวนของพิกเซลในภาพดิจิทัล โดยการนำเอาจำนวนของพิกเซลแนวตั้งคูณกับจำนวนของพิกเซลแนวนอน ซึ่งค่าที่ได้ก็คือค่าความละเอียดของกล้องดิจิทัลที่เราู้จักกันว่า กล้องตัวนี้ 3 ล้าน กล้องตัวนี้ 5 ล้าน หรือกล้องตัวนี้ 6 ล้าน เป็นต้น

ภาพขนาด $1600 \times 1200 = 1,920,000$ พิกเซล หรือเป็นความละเอียดสูงที่สุดที่กล้องดิจิทัล 2 ล้านพิกเซลสามารถบันทึกได้ ความละเอียดของกล้องดิจิทัลที่ระบุไว้จะดูที่ความละเอียดสูงที่สุดที่กล้องสามารถบันทึกได้เท่านั้น (นอกจากกล้องบางรุ่นที่ระบุความละเอียดโดยใช้หลักการของการประมวลผลและเพิ่มข้อมูลให้กับภาพ)

2.5 ความรู้เกี่ยวกับ Image Sensor

ตัว Sensor ในการรับภาพถือเป็นปัจจัยสำคัญที่กำหนดประเภทของกล้องดิจิทัลในปัจจุบัน นอกจากจะเป็นตัวกำหนดประเภทแล้วยังเป็นตัวบ่งบอกถึงคุณภาพของตัวกล้องอีกด้วย นั่นเพราะว่ากล้องที่ใช้เทคโนโลยี CMOS นั้น เป็นกล้องที่จัดอยู่ในระดับ Low end หรือกล้องราคาต่ำสำหรับมือใหม่ อีกทั้งยังพบมากในกล้องเก่าๆที่ตกุ่นไปแล้ว ส่วนกล้องที่ใช้เทคโนโลยี CCD นั้น จะเป็นกล้องที่ออกมาสู่ตลาดในช่วงปัจจุบันและเป็นกล้องที่มีคุณภาพของภาพถ่ายที่ดีกว่า แต่ต้องไม่ลืมด้วยว่าคุณก็จะต้องแลกมาด้วยจำนวนเงินที่เพิ่มสูงขึ้นเช่นเดียวกัน

ทั้ง CCD และ CMOS นั้น เริ่มต้นจากจุดเดียวกันนั่นคือ อาศัยหลักการทำงานในการอาศัย Photosite ให้เปลี่ยนแสงที่ตกกระทบให้กลายเป็นอิเล็กตรอน เพื่อบ่งบอกค่าของแสงสีนั้นๆ ภายในตัว Sensor ทั้งสองชนิดนี้จึงประกอบไปด้วย Photosite ขนาดเล็กๆนับล้านๆชิ้นเพื่อทำหน้าที่ในการรับแสง ข้อแตกต่างของทั้งสองเทคโนโลยีเกิดขึ้นเมื่อเข้าสู่ขั้นตอนของการคำนวณค่าของแสงนั้นๆจากแต่ละ Photosite ในอุปกรณ์ที่ใช้ CCD นั้น จะทำการประจุก้านั้นๆโดยตรงในแต่ละ Photosite จากนั้นจะแปลงค่าอนุภาคของแสงที่ตกกระทบให้กลายเป็นค่าดิจิทัล กระบวนการทั้งหมดนี้สามารถเกิดขึ้นได้อย่างรวดเร็วแตกต่างจาก CMOS ที่ถึงแม้ว่าแต่ละ Photosite จะสามารถประจุก้าได้โดยตรงเช่นกัน แต่กระบวนการส่งผ่านข้อมูลเหล่านั้นยังต้องอาศัยสายขนาดเล็กมากๆเพื่อทำการส่งผ่านข้อมูลเหล่านั้นเพื่อประมวลผลอีกต่อหนึ่ง

ในกระบวนการผลิตนั้น CCD จะใช้กรรมวิธีพิเศษในการสร้างความสามารถในการส่งผ่านประจุกโดยตรงไปยังตัว Chip โดยไม่มีปัญหาในการคัดทอนสัญญาณ ซึ่งด้วยวิธีการนี้จึง

ต้องอาศัยขบวนการผลิตที่มีคุณภาพสูงมากเป็นพิเศษ เพื่อให้ตัว Sensor ที่มีคุณภาพและมีความไวต่อแสงอย่างยิ่งยวดในขณะที่ CMOS นั้นยังอาศัยเทคโนโลยีการผลิตในรูปแบบเดิมๆที่ผ่านมา และแทบไม่มีการเปลี่ยนแปลงอะไรมากนัก ซึ่งเทคนิคเช่นเดียวกับการผลิต Microprocessor ดังนั้นด้วยความแตกต่างในเทคโนโลยีและกระบวนการการผลิตนี้เองทำให้ทั้ง CCD และ CMOS นี้มีต้นทุนในการผลิตและมีคุณภาพของการประมวลผลที่แตกต่างกันไปด้วย

2.6 เปรียบเทียบ CCD กับ CMOS

- CCD Sensors ถูกสร้างด้วยเทคโนโลยีและกรรมวิธีการผลิตขั้นสูงทำให้ไม่มีปัญหาในการรบกวนของสัญญาณภาพ ในขณะที่ CMOS นั้นอาศัยกระบวนการต่างๆในการผลิตอีกทั้งยังเต็มไปด้วยสัญญาณรบกวนที่จะลดทอนคุณภาพของภาพที่ได้

- แต่ละ Photosite ของ CMOS นั้นจะประกอบไปด้วยทรานซิสเตอร์จำนวนมากที่วางเรียงติดกันอยู่ ทำให้ปัญหาในการส่งผ่านข้อมูลมีความผิดพลาดสูงอีกทั้งที่ Cell รับแสงนั้นยังมีความไวต่อแสงน้อยกว่าเทคโนโลยี CCD

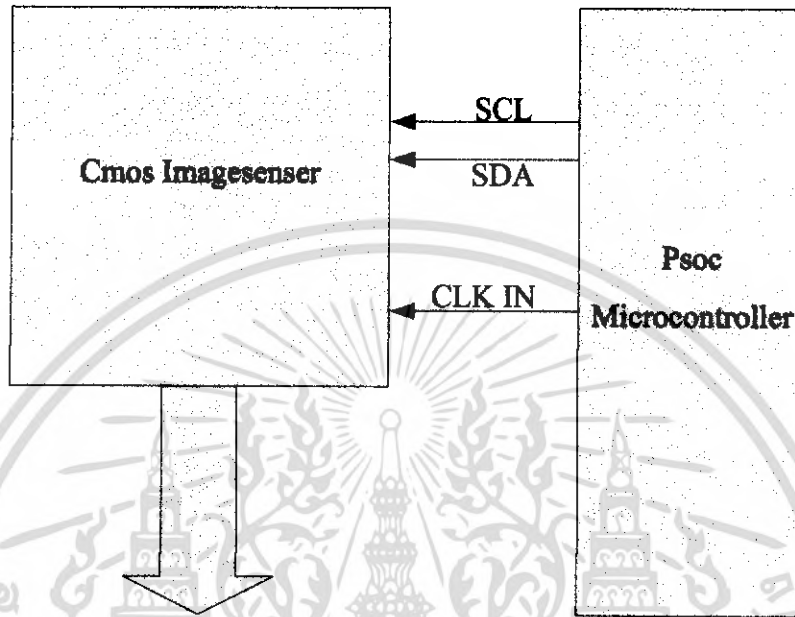
- อุปกรณ์ CCD นั้นมีอัตราปรีโกลทพลังงานสูงกว่า CMOS นั้นเพราะว่ามันต้องอาศัยการประจุอิเล็กตรอนโดยตรง เพื่อทำให้เกิดสัญญาณและค่าดิจิตอลถ่ายทอดออกไปได้ ซึ่งเมื่อเปรียบเทียบแล้วมีอัตราสูงกว่า 100 เท่าเลยทีเดียวเมื่อเทียบกับ CMOS

- การผลิต CMOS sensor นั้นสามารถพิมพ์ลายวงจรได้ลงบนแผ่นเวเฟอร์ซิลิคอนมาตรฐานแทบทุกชนิด ในขณะที่ CCD นั้นจะต้องอาศัยแผ่นซิลิคอนที่ถูกผลิตมาเป็นพิเศษ เพื่อทำการผลิตและนี่คืออีกสาเหตุหนึ่งที่ทำให้ต้นทุนการผลิตถึงได้แตกต่างกัน

ด้วยความแตกต่างกันในหลายๆด้านนี้เองทำให้คุณจะได้เห็น CCD sensor นี้อยู่ในเฉพาะกล้องดิจิตอลรุ่นใหม่ๆระดับมืออาชีพ ซึ่งให้ความสมจริงของภาพและมีความละเอียดสูง ในขณะที่ CMOS sensor นั้นส่วนใหญ่มักพบในกล้องดิจิตอลมือสมัครเล่นหรือในระดับ Low end ซึ่งมีราคาไม่แพงนัก แต่ก็ต้องไม่ลืมด้วยว่าไม่อาจคาดหวังถึงความละเอียดและคุณภาพของภาพถ่ายในระดับมืออาชีพได้ อย่างไรก็ตามถึงตอนนี้ผู้ผลิต CMOS sensor รายใหญ่ก็กำลังพัฒนาและปรับปรุงเพื่อให้คุณภาพของ CMOS ก้าวขึ้นมาเทียบเคียงกับ CCD ได้

บทที่ 3

หลักการทํางาน



รูปที่ 3.1 Block diagram แสดงหลักการทํางาน

3.1 หลักการทํางาน

โครงการนี้คือการควบคุมอิมเมจเซนเซอร์ด้วยไมโครโปรเซสเซอร์ โดยใช้ไมโครโปรเซสเซอร์คือ พิคอคไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทํางานของซิมอสอิมเมจเซนเซอร์ โดยการส่งค่ารีจิสเตอร์ต่างๆของซิมอสอิมเมจเซนเซอร์ผ่านทางไอสแควซี นอกจากนี้ยังใช้วงจรมินิซิสัญญาณนาฬิกาภายในพิคอคไมโครคอนโทรลเลอร์ เพื่อกําหนดความเร็วในการทํางานของซิมอสอิมเมจเซนเซอร์ด้วย โดยความถี่ของสัญญาณนาฬิกาที่ใช้คือ 2 เมกกะเฮิร์ต จากนั้นซิมอสอิมเมจเซนเซอร์จะส่งสัญญาณข้อมูล 10 บิตออกมา แล้วนำสัญญาณที่ได้ นั้น ไปติดต่อกับคอมพิวเตอร์โดยตรงผ่านพอร์ตขนานด้วยโปรแกรมวิชวลเบซิก

3.2 หลักการทํางานของซิมอสอิมเมจเซนเซอร์ MT9M001

การทํางานจะเริ่มจากการเขียนและอ่านข้อมูลจากตัวซิมอสอิมเมจเซนเซอร์ผ่านทางบัสสองเส้นและเป็นการควบคุมโดย SCLK ซึ่งจะเคลื่อนไปทางมาสเตอร์ข้อมูลจะถูกส่งไปทั้งภายในและภายนอกของตัวเซนเซอร์ผ่านทาง SDATA โดยเส้น SDATA ถูกพูดถึงที่ 3.3 โวลต์และ โหลด

1,500 โอห์ม การอินเตอร์เฟสแต่ละอุปกรณ์จะยินยอมให้ผ่านลงมาทาง SDATA เมื่อมีการกำหนดให้ทุกเมื่อ

ชนิดของการอ่านและเขียนจะเรียงจากมาสเตอร์ส่งไปสแตร์ทบิทหลังจากสแตร์ทบิท มาสเตอร์จะส่งไปยังสเลฟ 8 บิต ข้อมูลบิตสุดท้ายของแอดเดรสจะกำหนดให้ถ้ามีการขอให้อ่าน หรือเขียน ที่ "0" แสดงถึงการเขียนส่วน "1" แสดงถึงการอ่านเมื่อสเลฟตอบรับแล้วมันจะส่งบิต กลับไปยังมาสเตอร์

ถ้ามีการบอกให้มีการเขียนมาสเตอร์จะส่งรีจิสเตอร์ขนาด 8 บิต ซึ่งการเขียนจะถูกนำไป วางไว้ในตำแหน่ง เมื่อสเลฟมีการตอบรับการส่งบิตแล้วจะแสดงถึงแอดเดรสของรีจิสเตอร์ที่ถูก ขอมรับ เมื่อมาสเตอร์มีการส่งข้อมูลขนาด 8 บิต แล้วสเลฟจะทำการส่งและมีการตอบรับหลังจาก ครบทีละ 8 บิต ตัวซีมอสอิมเมจเซนเซอร์จะใช้ 16 บิต ข้อมูลสำหรับรีจิสเตอร์ภายในดังนั้นจึง ต้องบังคับให้มีการส่งข้อมูลที่ละ 8 บิต 2 ครั้ง สำหรับการเขียนทีละหนึ่งรีจิสเตอร์หลังจากส่ง ครบ 16 บิต รีจิสเตอร์แอดเดรสจะเพิ่มขึ้นโดยอัตโนมัติและ 16 บิต ต่อไปจะถูกเขียนไปยัง รีจิสเตอร์แอดเดรสถัดไปมาสเตอร์จะหยุดเขียนเมื่อมีการส่งสต็อบบิต

ชนิดของการอ่านจะดำเนินการตามข้อมูลที่มีความสำคัญที่มีผลต่อการนำข้อมูลมาให้ ซึ่ง เป็นสิ่งที่สำคัญมาก ๆ ต่อการส่งผ่านข้อมูลจากตัวซีมอสอิมเมจเซนเซอร์ ผ่านทางบัสสองเส้นและ เป็นการควบคุมโดย SCLK ซึ่งจะเคลื่อนไปทางมาสเตอร์ข้อมูลจะถูกส่งไปทั้งภายในและภายนอก ของตัวเซนเซอร์ผ่านทาง SDATA ซึ่งอันดับแรกจะเป็นมาสเตอร์ที่ทำการเขียนโหมด สเลฟ แอดเดรสและ 8 บิต รีจิสเตอร์แอดเดรสจะเขียนเมื่อมีการขอ เมื่อมาสเตอร์ส่งสแตร์ทบิตแล้วจะทำการอ่านโหมดสเลฟแอดเดรสแล้วมาสเตอร์จะมีสัญญาณคือออกมาที่รีจิสเตอร์ ซึ่งมีขนาด 8 บิต มาสเตอร์จะส่งบิตตอบรับหลังจากการส่งแต่ละ 8 บิต รีจิสเตอร์แอดเดรสจะเพิ่มขึ้นโดย อัตโนมัติหลังจากทุกๆ 16 บิต ที่มีการส่งแล้วข้อมูลจะหยุดส่งเมื่อมาสเตอร์ส่งออกมาว่า No-Acknowledge bit

3.3 การกำหนดคำสั่งของซีมอสอิมเมจเซนเซอร์ MT9M001

- Start bit

คือการกำหนดจาก High เป็น Low เป็นการส่งผ่านของสายข้อมูลขณะที่ clock เป็น High

- Stop bit

คือการกำหนดจาก Low เป็น High เป็นการส่งผ่านของสายข้อมูลขณะที่ clock เป็น High

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Slave address

แอดเดรส 8 บิตของบัสไอทีแควจะประกอบขึ้นด้วย 7 บิตของแอดเดรสและ 1 บิตของคำสั่ง ที่ "0" (0xBA) ใน LSB ของแอดเดรสแสดงถึงโหมดการเขียนและ "1" (0xBB) แสดงถึงโหมดการอ่าน

- Data bit transfer

1 บิตข้อมูลจะถูกส่งในระหว่างแต่ละสัญญาณนาฬิกาเส้นทางการอินเตอร์เฟสของสัญญาณนาฬิกาจะมีเงื่อนไขจากมาสเตอร์คือข้อมูลจะต้องคงที่ระหว่างคาบที่เป็น High ของการอินเตอร์เฟส โดยสามารถเปลี่ยนเฉพาะตอนที่ clock เป็น Low ข้อมูลจะถูกส่งทีละ 8 บิตตามด้วยการตอบรับของบิต

- Acknowledge bit

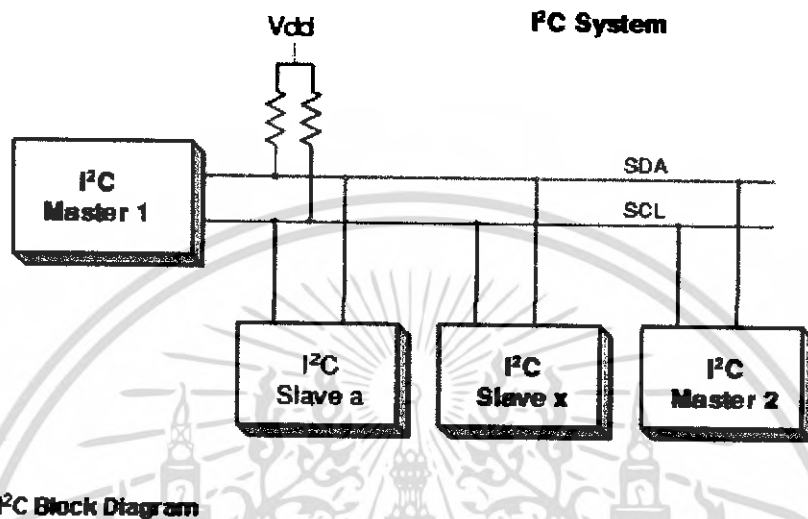
มาสเตอร์จะสร้างการตอบรับของสัญญาณนาฬิกา ตัวส่งสัญญาณ(คือมาสเตอร์เมื่อทำการเขียนหรือสเลฟเมื่อทำการอ่าน)จะถูกปลดออกจากสายข้อมูลและจะถูกยอมรับแสดงถึงการตอบรับของบิต โดยการดึงสายข้อมูลให้เป็น Low ระหว่างการตอบรับของสัญญาณนาฬิกา

- No-Acknowledge bit

เกิดขึ้นเมื่อเส้นข้อมูลไม่พุดคาวน์ โดยการยอมรับระหว่างการตอบรับของสัญญาณนาฬิกาจะใช้เมื่อสิ้นสุดการเรียงลำดับการอ่าน

3.4 การใช้งานโมดูล I²C HW

I²C = I²C Bus ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆว่า I²C BUS (ไอ-แอสคว-ซี-บัส) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อดสื่อสารระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้นคือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆ ตัวเข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น ในการใช้งาน I²C Bus นั้นจะใช้ขาสัญญาณเพียงสองเส้นเท่านั้นคือ ขา สัญญาข้อมูล (SDATA) และขาสัญญาณนาฬิกา (SCLK) โดยสามารถต่อเชื่อมอุปกรณ์ต่างๆเข้าด้วยกันเป็นระบบบัสได้ ซึ่งกรณีที่มีการต่อเป็นระบบบัสจะต้องมีอุปกรณ์หนึ่งที่ทำหน้าที่เป็นตัวมาสเตอร์(Master) เพื่อควบคุมการสื่อสาร ส่วนอุปกรณ์ต่อพ่วงตัวอื่นๆในระบบบัสเป็นสเลฟ (Slave) โดยอุปกรณ์ที่นำมาต่อเชื่อมกันภายในบัสจะต้องมีหมายเลข ID ไม่ตรงกัน ซึ่งลักษณะการเชื่อมต่อจะเป็นคิงบัสสื่อไออะแกรมรูปที่ 3.2



รูปที่ 3.2 บล็อกไดอะแกรมการเชื่อมต่อแบบ I²C

โมดูล I²C HW จะสนับสนุนความเร็วในการสื่อสารข้อมูลตั้งแต่ 50 จนถึง 400kbits/s โดยจะเป็นฮาร์ดแวร์ I²C ภายใน จะไม่มีการใช้จิกจิลออสติก หรือ อนุภาคบิตออสติกเลข การใช้งานจะเรียกใช้งานผ่านฟังก์ชัน API ซึ่งการเรียกใช้เพียงฟังก์ชันเดียวของ API ก็สามารถรับและส่งข้อมูลหลายๆ ไบต์ต่อกันได้ นอกจากนี้ยังสนับสนุนการใช้งานแบบ multi-master และโหมดแอดเดรสทั้งแบบ 7-bit addressing และ 10-bit addressing ในการต่อใช้งานจำเป็นต้องต่อตัวต้านทานพูลอัพให้กับขาสัญญาณ (SDATA) และขาสัญญาณนาฬิกา (SCLK) ด้วย โดยค่าความต้านทานควรมีค่าความต้านทานไม่ต่ำกว่า 1,500 โอห์ม แนะนำควรใช้ประมาณ 4,700 โอห์ม สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ I²C Bus สามารถดูได้จากเว็บไซต์ของ Philips (www.philips.com)

3.4.1 คุณสมบัติของโมดูล I²C HW

- ใช้การสื่อสารแบบ I²C Bus ตามมาตรฐานของ Philips
- สามารถทำงานเป็นมาสเตอร์ สเลฟหรือแบบมัลติโหมดได้
- ใช้ขาสัญญาณเพียงสองขาสัญญาณ (SDATA และ SCLK) ในการเชื่อมต่ออุปกรณ์ต่างๆ
- สนับสนุนความเร็วข้อมูลมาตรฐานที่ 50 จนถึง 400kbits/s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

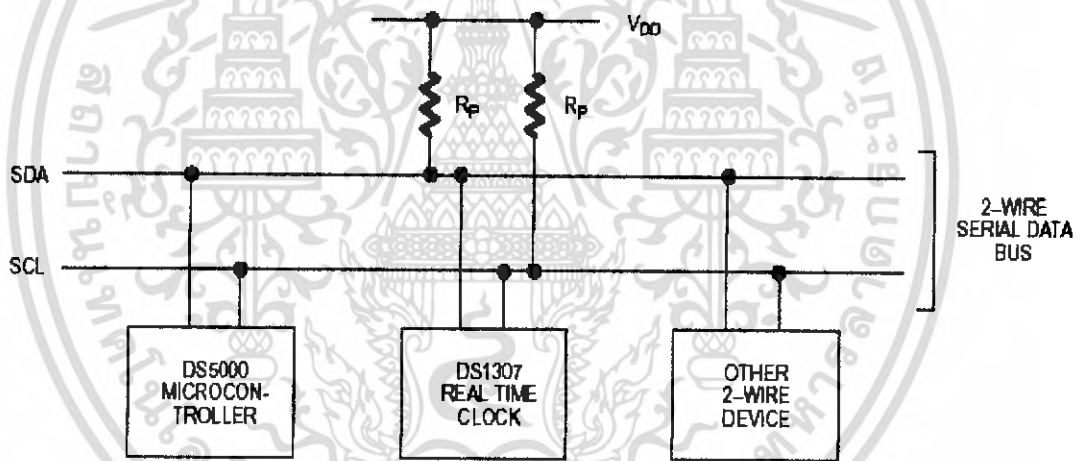
- มี API (Application Programming Interface) ระดับ High level เป็นฟังก์ชันที่สร้างไว้ให้พร้อมอยู่แล้วสามารถเรียกใช้งานได้เลย

- มี API (Application Programming Interface) ระดับ Low level เป็นฟังก์ชันขนาดเล็กๆที่เตรียมไว้ให้ผู้เขียนโปรแกรม โดยสามารถปรับเปลี่ยนการทำงานของโปรแกรมได้ละเอียดยิ่งขึ้น

- สนับสนุนโหมดแอดเดรสแบบ 7 บิตและ 10 บิต

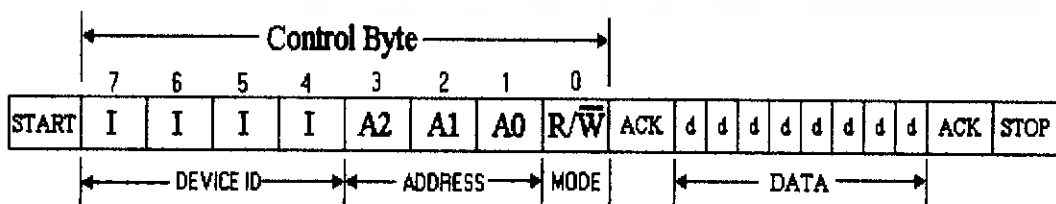
3.4.2 การเชื่อมต่ออุปกรณ์แบบ I²C BUS

I²C BUS ใช้สายสัญญาณ 2 เส้น คือ SCL ,SDA สำหรับติดกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับตัวต้านทานแบบ pull up 2-10K เนื่องจากเอาต์พุตมีลักษณะเป็นแบบ Open Darin หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัว



รูปที่ 3.3 ลักษณะการการเชื่อมต่ออุปกรณ์แบบ I²C BUS

- การเขียนและอ่านข้อมูลกับอุปกรณ์แบบ I²C BUS

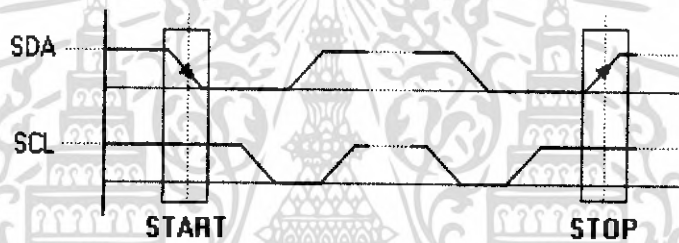


รูปที่ 3.4 แบบการเขียน/อ่านข้อมูลแบบ I²C BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับ-ส่งข้อมูลแบบ I²C BUS MCU จะเริ่มต้นการส่งข้อมูลด้วยการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัสแล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัส ประจำตัวอุปกรณ์ Device ID ,Device Address ,และ Mode ในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์ รับทราบว่า MCU ต้องการ จะติดต่อกับก็ต่อส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส สถานะบัสว่าง คือเมื่อบัสไม่ได้ถูกใช้งาน ทั้ง SCL และ SDA จะเป็น 1 ทั้งคู่

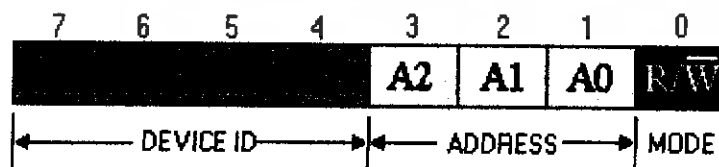
- การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I²C BUS (START and STOP Conditions)



รูปที่ 3.5 สถานะเริ่มต้นและสถานะสิ้นสุดของ I²C BUS

ลักษณะการกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I²C BUS เมื่อต้องการส่งข้อมูล MCU จะต้องส่งสถานะเริ่มต้น (START Conditions) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1 เมื่อสิ้นสุดการการใช้บัส MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) คือให้ SDA เปลี่ยนจาก 0 มาเป็น 1 ในขณะที่ SCL มีค่าเป็น 1

- รหัสควบคุมของ I²C BUS (Control Byte)

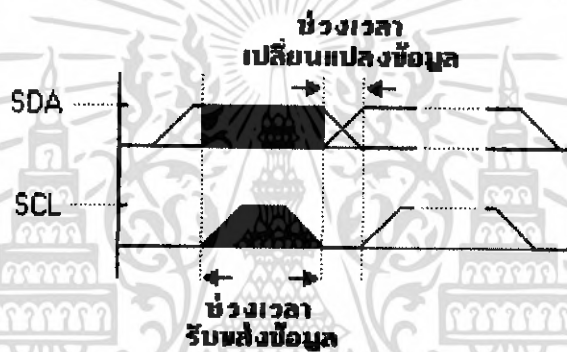


รูปที่ 3.6 รหัสควบคุมของ I²C BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสควบคุมของ I²C BUS ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิต 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่าน รหัสประจำตัวของอุปกรณ์ ประกอบด้วยรหัสประจำตัวจากผู้ผลิต Product ID 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้ สามารถ กำหนด เองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุตัวอุปกรณ์ ที่ต่ออยู่บนบัส จะมีค่าซ้ำกัน ไม่ได้ บิตควบคุมการเขียนอ่าน (Mode) บิต 0 เมื่อ MCU ต้องการเขียนข้อมูลไปยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการ อ่านข้อมูล จากอุปกรณ์ ก็กำหนดให้บิตนี้เป็น 1

- ช่วงเวลารับส่งบิตข้อมูลของ I²C BUS



รูปที่ 3.7 การรับส่งบิตข้อมูลของ I²C BUS

ช่วงเวลารับส่งบิตข้อมูลของ I²C BUS ถ้าสถานะการรับ-ส่งข้อมูล จะกระทำในขณะที่ขา SCL เป็น 1 และถ้าสถานะการเปลี่ยนแปลงข้อมูล จะกระทำในขณะที่ขา SCL เป็น 0

3.5 การควบคุม CMOS Image Sensor

โดยการควบคุม CMOS Image Sensor เราจะใช้การเขียนโปรแกรมลงใน PSoc และเลือกใช้โมดูล I²C HW ในการใช้งาน ซึ่งตัว CMOS Image Sensor ก็จะมีขาสัญญาณข้อมูล (SDATA) และขาสัญญาณนาฬิกา (SCLK) ทำให้สามารถควบคุมการสื่อสารข้อมูลแบบ I²C Bus ได้ โดยในการเขียนโปรแกรมเราจะต้องกำหนดค่ารีจิสเตอร์ต่างๆให้ตรงกับค่ารีจิสเตอร์ที่เราต้องการจากตัว CMOS Image Sensor ซึ่งค่ารีจิสเตอร์ต่างๆได้จากตารางที่ 3.1

Address (Hex)	Register	Default (Binary)	Default Value (Hex)
0x00	Chip Version	1000 0100 0001 0001	0x8431
0x01	Row Start	0000 0ddd dddd dddd	0x000C
0x02	Column Start	0000 0ddd dddd dddd	0x0014
0x03	Row Size (Window Height)	0000 0ddd dddd dddd	0x03FF
0x04	Col Size (Window Width)	0000 0ddd dddd dddd	0x04FF
0x05	Horizontal Blanking	0000 0ddd dddd dddd	0x0009
0x06	Vertical Blanking	0000 0ddd dddd dddd	0x0019
0x07	Output Control	0000 0000 0d00 00dd	0x0002
0x09	Shutter Width	00dd dddd dddd dddd	0x0419
0x0B	Restart	0000 0000 0000 000d	0x0000
0x0C	Shutter Delay	0000 0ddd dddd dddd	0x0000
0x0D	Reset	0000 0000 0000 000d	0x0000
0x1E	Read Options 1	1000 dddd 00dd d000	0x8000
0x20	Read Options 2	d0d1 0dd1 d00d d10d	0x1104
0x2B	Even Row, Even Column	0000 0000 0ddd dddd	0x0008
0x2C	Odd Row, Even Column	0000 0000 0ddd dddd	0x0008
0x2D	Even Row, Odd Column	0000 0000 0ddd dddd	0x0008
0x2E	Odd Row, Odd Column	0000 0000 0ddd dddd	0x0008
0x35	Global Gain	0000 0000 0ddd dddd	0x0008
0x5F	Cal Threshold	dddd dddd d0dd dddd	0x0904
0x60	Even Row, Even Column	0000 000d dddd dddd	0x0000
0x61	Odd Row, Odd Column	0000 000d dddd dddd	0x0000
0x62	Cal Ctrl	d00d d100 1001 1ddd	0x0498
0x63	Even Row, Odd Column	0000 000d dddd dddd	0x0000
0x64	Odd Row, Even Column	0000 000d dddd dddd	0x0000
0xF1	Chip Enable	0000 0000 0000 00dd	0x0001

Note: 1 = always 1
0 = always 0
d = programmable

ตารางที่ 3.1 รีจิสเตอร์ต่างๆของ CMOS Image Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

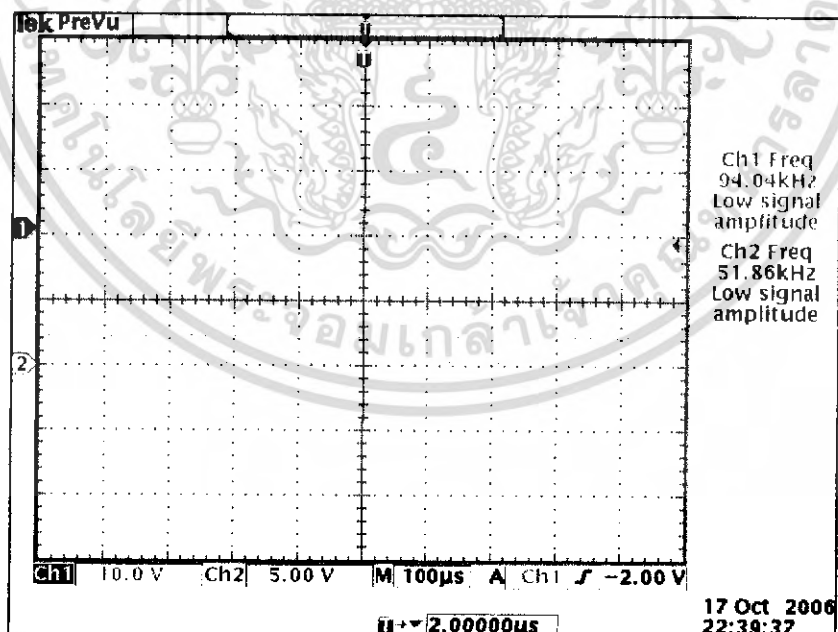
บทที่ 4

การทดลองและผลการทดลอง

4.1 สัญญาณที่ได้จาก CMOS Image Sensor

การทดลองนี้คือการเขียนค่าให้ซีมอสอิมเมจเซนเซอร์ผ่านเส้นทางไอสแควซี โดยจะมีสองสายคือสายสัญญาณนาฬิกา (SCLK) ซึ่งจะใช้การป้อนสัญญาณที่ 100 กิโลเฮิร์ตซ์ โดยเราสามารถเลือกได้จากโปรแกรมพีชอกไมโครคอนโทรลเลอร์ ที่มีเครื่องกำเนิดสัญญาณนาฬิกาภายในตัวเอง และต่อมาคือสายสัญญาณข้อมูลซึ่งเราสามารถเขียนโปรแกรมป้อนค่าข้อมูลต่างๆ ได้จาก โปรแกรมพีชอกไมโครคอนโทรลเลอร์ตามที่ซีมอสอิมเมจเซนเซอร์ต้องการได้ โดยเราสามารถดูค่าข้อมูลต่างๆ ได้จากค่าดัชนีของซีมอสอิมเมจเซนเซอร์เอง

การทำงานเริ่มจากการเขียนค่าแอดเดรสของซีมอสอิมเมจเซนเซอร์คือค่า 0xBA แล้วจึงเขียนค่ารีจิสเตอร์แอดเดรสต่างๆ แล้วจึงเขียนค่าข้อมูลอีก 16 บิต เข้าไปเก็บไว้ในแต่ละรีจิสเตอร์แอดเดรส โดยการเขียนแต่ละครั้งจะให้มีการเขียนได้ทีละ 8 บิต ดังนั้นการเขียนค่าข้อมูลจึงต้องเขียนทีละ 8 บิต 2 ครั้ง เมื่อเขียนค่าข้อมูลครบแล้ว จึงนำสไปวาล์วสัญญาณที่ขาสัญญาณทั้ง 2 เส้น แล้วดูว่าผลที่ได้ตรงกับค่าที่เราเขียนส่งไปให้หรือไม่



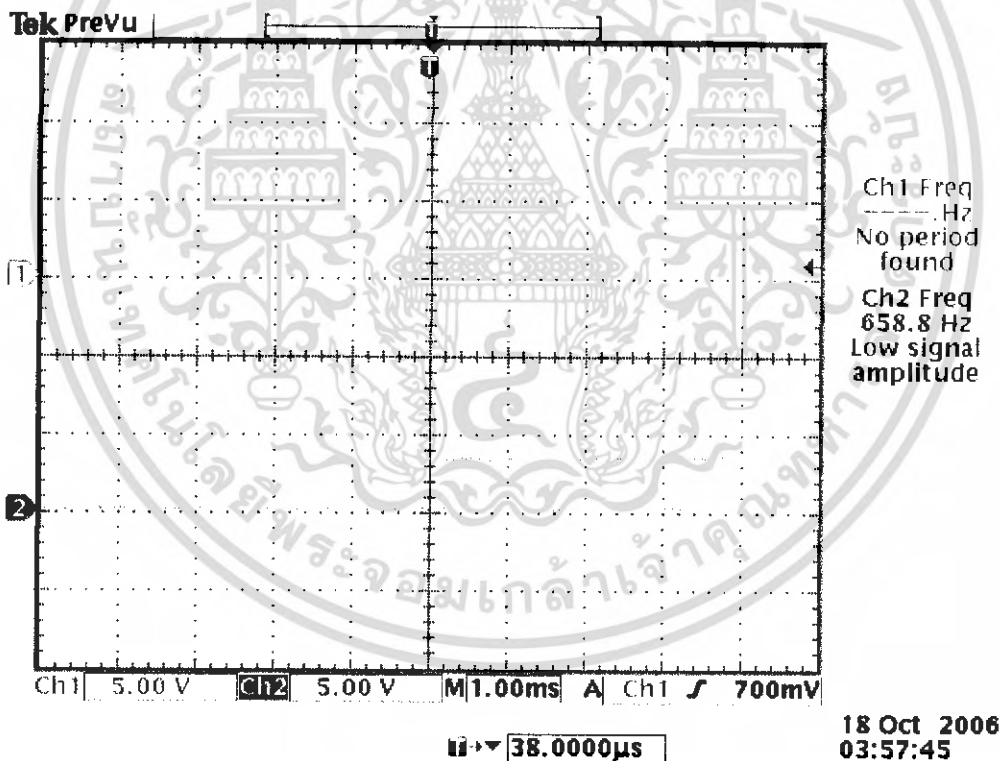
รูปที่ 4.1 สัญญาณ SDATA และ SCLK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 ค่าสัญญาณ SDATA และ SCLK ที่ได้มีความถูกต้องตรงกับค่าที่ส่งไป โดยในสัญญาณช่วงแรกจะเป็นค่าแอดเดรสของซีมอสิมเมจเซนเซอร์คือ 0xBA ช่วงที่สองคือค่ารีจิสเตอร์แอดเดรส ในที่นี้เราจะอ้างถึงที่ตำแหน่ง 0x09 ส่วนช่วงที่สามและช่วงที่สี่คือค่าข้อมูลที่เรากำหนดให้อีก 16 บิต

เมื่อสามารถได้สัญญาณที่ถูกต้องแล้ว เราจึงเริ่มทำการโปรแกรมค่ารีจิสเตอร์ต่างๆ ส่งให้กับซีมอสิมเมจเซนเซอร์ โดยค่ารีจิสเตอร์ต่างๆ สามารถดูได้จากตารางที่ 3.2 โดยจะมีรีจิสเตอร์ที่สำคัญอยู่บางรีจิสเตอร์เท่านั้นคือ รีจิสเตอร์ reset กับรีจิสเตอร์ restart ที่ต้องทำการโปรแกรมก่อนเพื่อให้ซีมอสิมเมจเซนเซอร์เริ่มทำงานได้อย่างถูกต้อง ส่วนรีจิสเตอร์อื่นๆ ต้องทำการเลือกโปรแกรมเอง โดยดูรายละเอียดต่างๆ ได้จากคาส์ชิทของซีมอสิมเมจเซนเซอร์

หลังจากทำการโปรแกรมค่ารีจิสเตอร์ต่างๆ แล้ว จึงนำสโคปมาวัดสัญญาณที่เอาต์พุตของซีมอสิมเมจเซนเซอร์

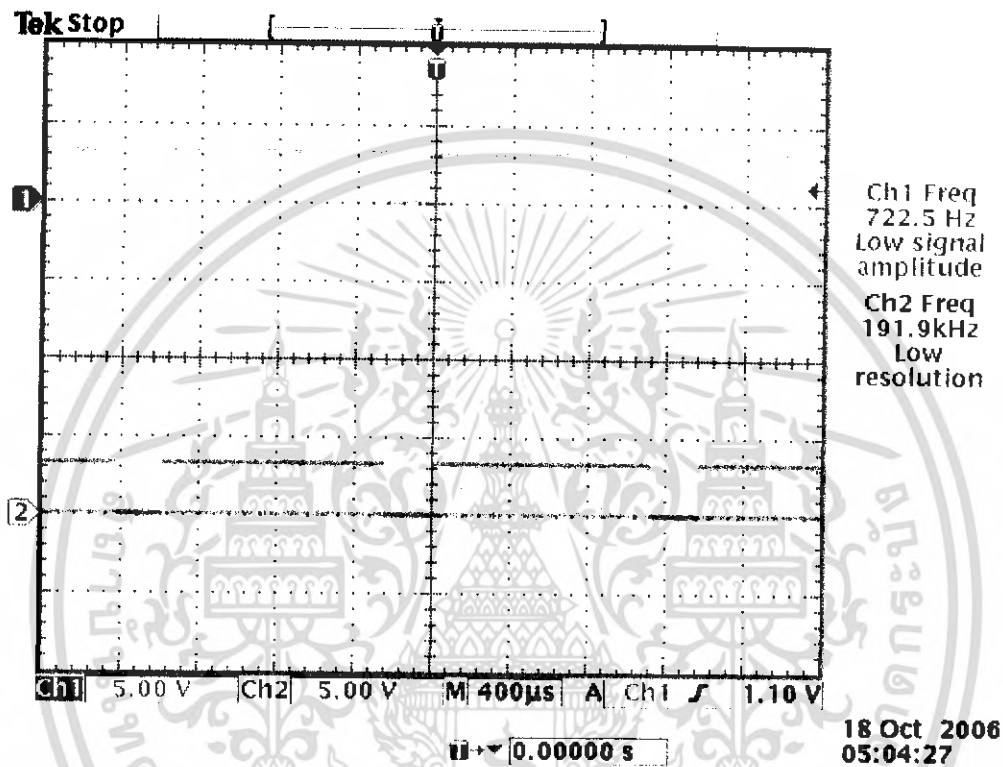


รูปที่ 4.2 สัญญาณ FRAME_VALID และ LINE_VALID

รูปที่ 4.2 เป็นสัญญาณ FRAME_VALID และ LINE_VALID โดย 1 เฟรมก็คือภาพทั้งหมดของซีมอสิมเมจเซนเซอร์ 1 ภาพ ในสัญญาณ 1 เฟรมก็จะมีสัญญาณ LINE_VALID 1024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูกสัญญาณ ซึ่งเป็นค่าพิกเซลในแนวตั้งของซิมอสติมเมจเซนเซอร์ และใน 1 ลูกสัญญาณของ LINE_VALID ก็คือค่าพิกเซลในแนวนอน 1280 พิกเซล
 ในสัญญาณ LINE_VALID 1 ลูกสัญญาณ ก็จะมีสัญญาณข้อมูล 10 บิตอยู่ ดังรูปที่ 4.3 ซึ่งเราจะนำสัญญาณข้อมูลนั้นมาประมวลผลภาพโดยใช้ซอฟต์แวร์ต่อไป

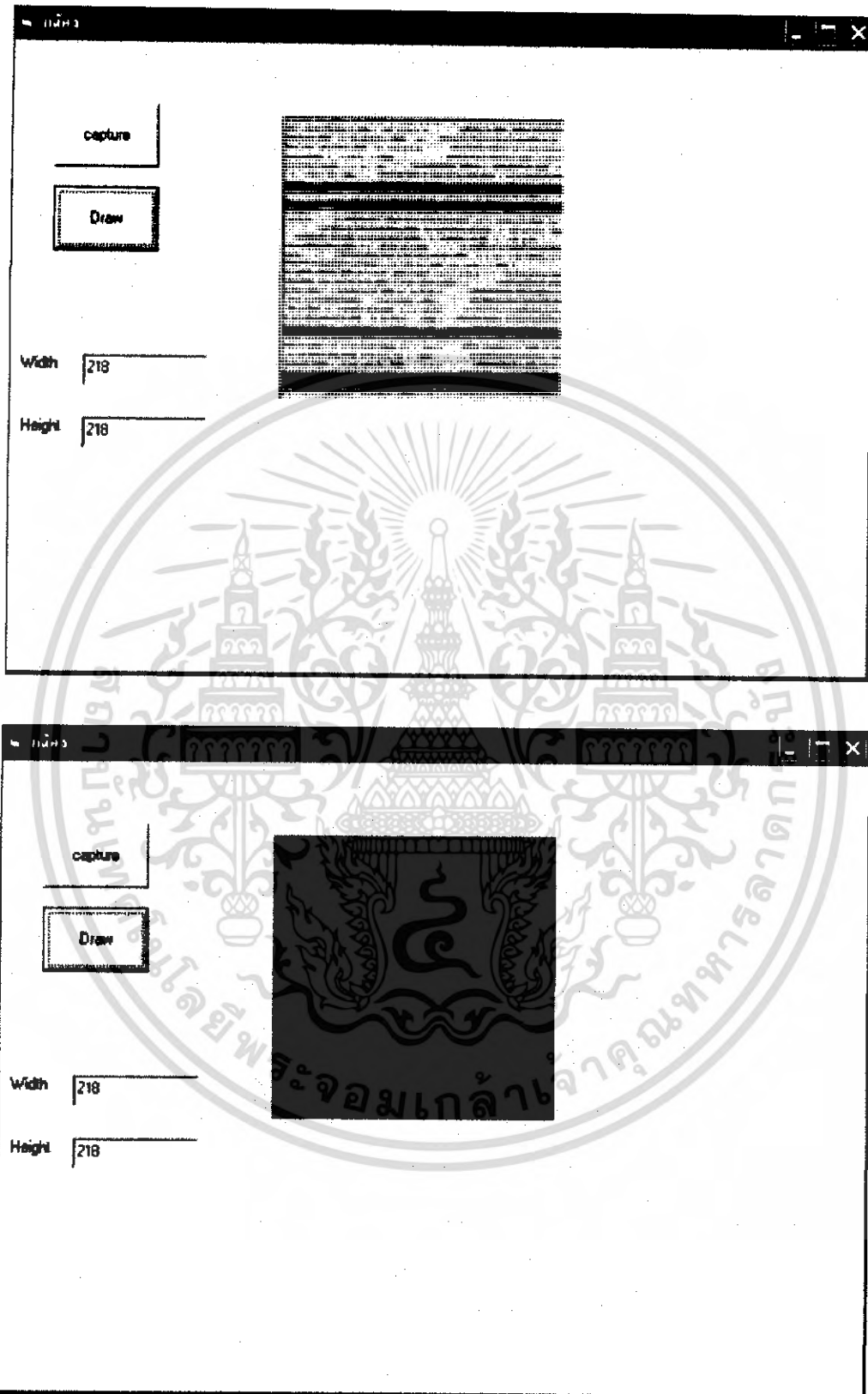


รูปที่ 4.3 สัญญาณ LINE_VALID และ DATAOUT

4.2 การนำสัญญาณข้อมูลติดต่อคอมพิวเตอร์ผ่านพอร์ตขนานด้วยโปรแกรม Visual Basic

เมื่อได้สัญญาณข้อมูลออกมา 10 บิตแล้ว จะเลือกมาใช้ติดต่อพอร์ตขนานแค่ 8 บิต เท่านั้น โดยจะตัดบิตที่ต่ำสุดออก แล้วจึงเขียน โปรแกรมติดต่อพอร์ตขนาน

หลังจากที่ได้เขียนโปรแกรมติดต่อพอร์ตขนานแล้ว จะเห็นได้ว่าไม่สามารถนำภาพที่ชัดเจนออกมาได้เลยเพราะความเร็วในการติดต่อสัญญาณผ่านพอร์ตขนานนั้นไม่เพียงพอที่จะรับความเร็วของสัญญาณข้อมูลที่ออกมาจากซิมอสติมเมจเซนเซอร์ได้ จึงทำให้เห็นได้แค่เพียงความแตกต่างของภาพที่ให้อิมเมจเซนเซอร์รับแสงกับไม่รับแสงได้เท่านั้น



รูปที่ 4.4 แสดงความแตกต่างของภาพที่ให้อิมเมจเซนเซอร์รับแสงกับไม่รับแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

จากการที่ได้ทดลองและศึกษาการทำงานของ โครงการนี้เป็นการเขียนค่าข้อมูลต่างๆจาก ไมโครโปรเซสเซอร์ผ่านทางสายสัญญาณบนระบบบัสไอส์แควซีให้ซิมอสติมเมจเซนเซอร์ทำงาน และส่งค่าสัญญาณต่างๆออกมา เพื่อที่จะนำไปใช้ในการประมวลผลภาพ ซึ่งในโครงการนี้ผู้ทดลอง ได้เลือกใช้การติดต่อสัญญาณ โดยตรงเข้าสู่คอมพิวเตอร์ผ่านทางพอร์ตขนาน ซึ่งอาจนับได้ว่าเป็น ข้อผิดพลาด เนื่องจากความเร็วในการรับข้อมูลของพอร์ตขนาน ไม่สามารถรับสัญญาณข้อมูลที่ ส่งออกมาจากซิมอสติมเมจเซนเซอร์ได้ทัน จึงทำให้ไม่สามารถนำข้อมูลออกมาเรียงเป็นภาพได้ โดยผู้ทดลองคิดว่าควรที่จะใช้วิธีการติดต่อแบบอื่นที่มีความเร็วในการรับส่งข้อมูลที่เร็วกว่านี้ เช่น ยูเอสบีพอร์ต หรือ การ์ด I/O พอร์ต แบบ PCI BUS INTERFACE ซึ่งน่าจะมีความเร็วพอที่จะรับ สัญญาณข้อมูลที่ออกมาจากซิมอสติมเมจเซนเซอร์ได้

หนังสืออ้างอิง

1. วันชัย ลิเลิศ , วิฑูร ศรีวิชัย “ ปริชญานิพนธ์ ” สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. บริษัท อีทีที จำกัด “ เรียนรู้และเข้าใจ PSoC Microcontroller คิว ภาษา Assembly และ ภาษา C ”
3. ฉัททวุฒิ พิษผล , พิชิต สันติกุลานนท์ , พร้อมเลิศ หล่อวิจิตร “ คู่มือเรียน Visual Basic 6 ”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

```
//-----  
// C main line  
//-----  
  
#include <m8c.h>    // part specific constants and macros  
#include "PSoC_API.h" // PSoC API definitions for all User Modules  
  
#define SLAVE_ADDRESS 0x5D  
  
#define Reset_Add 0x0D  
#define Reset_ValH1 0x00  
#define Reset_ValL1 0x01  
#define Reset_ValH2 0x00  
#define Reset_ValL2 0x00  
  
#define Read2_Add 0x20  
#define Read2_ValH 0x11  
#define Read2_ValL 0x04  
  
#define Restart_Add 0x0B  
#define Restart_ValH 0x00  
#define Restart_ValL 0x00  
  
// Prototypes function  
void I2C_WriteByte(unsigned char Add, unsigned char ValH, unsigned char ValL);  
void delay(long int del);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void main()
{

    PWM16_1_Start();

    I2CHW_1_Start();
    I2CHW_1_EnableMstr();

    delay(1000);
    I2C_WriteByte(Reset_Add,Reset_ValH1,Reset_ValL1);
    delay(10);
    I2C_WriteByte(Reset_Add,Reset_ValH2,Reset_ValL2);
    delay(10);
    I2C_WriteByte(Read2_Add,Read2_ValH,Read2_ValL);
    delay(10);
    I2C_WriteByte(Restart_Add,Restart_ValH,Restart_ValL);
    delay(10);

    while(1)
    {
    }
}

void I2C_WriteByte(unsigned char Add, unsigned char ValH, unsigned char ValL)
{
    I2CHW_1_fSendStart(SLAVE_ADDRESS,I2CHW_1_WRITE);
    I2CHW_1_fWrite(Add);
    I2CHW_1_fWrite(ValH);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
I2CHW_1_fWrite(Vall);  
I2CHW_1_SendStop();  
}  
  
void delay(long int del)  
{  
    long int j;  
    for (j = 0; j < del; ++j);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress As Integer) As Integer

'Public N As Integer

Public V As Integer

Public W As Integer

Public x As Integer

Public y As Integer

Public k As Integer

Public x1 As Integer

Public y1 As Integer

Private m(1280, 1024) As Integer

Private Sub Command1_Click()

Dim a As Integer

Dim b As Integer

Dim c As Integer

Dim d As Integer

Dim e As Integer

Dim f As Integer

Dim g As Integer

Dim h As Integer

Dim i As Integer

Dim j As Integer

Dim p As Integer

Dim l As Integer

'Out &H378, 0



```

'b = Inp(&H37A)
'V = Not (1 And Inp(&H37A))
'If V = 1 Then
For y = 1 To y1
For x = 1 To x1
    a = Inp(&H379)
    b = Inp(&H37A)

    c = 120 And a
    d = Not a
    e = 128 And d
    f = e + c

    g = 4 And b
    h = 3 And b
    i = Not h
    j = g + i
    k = f + j
    V = Not (1 And b)
    m(x, y) = k
Next x
For p = 1 To 400
Next p
Next y
'End If

```

End Sub

Private Sub Command2_Click()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For y = 1 To y1
  For x = 1 To x1
    Picture1.PSet (x - 1, y - 1), RGB(m(x, y), m(x, y), m(x, y))
  Next x
Next y
End Sub

```

```
Private Sub Form_Load()
```

```
  x1 = Text1.Text
```

```
  y1 = Text2.Text
```

```
  Picture1.Width = x1
```

```
  Picture1.Height = y1
```

```
  'Out &H378, 0
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

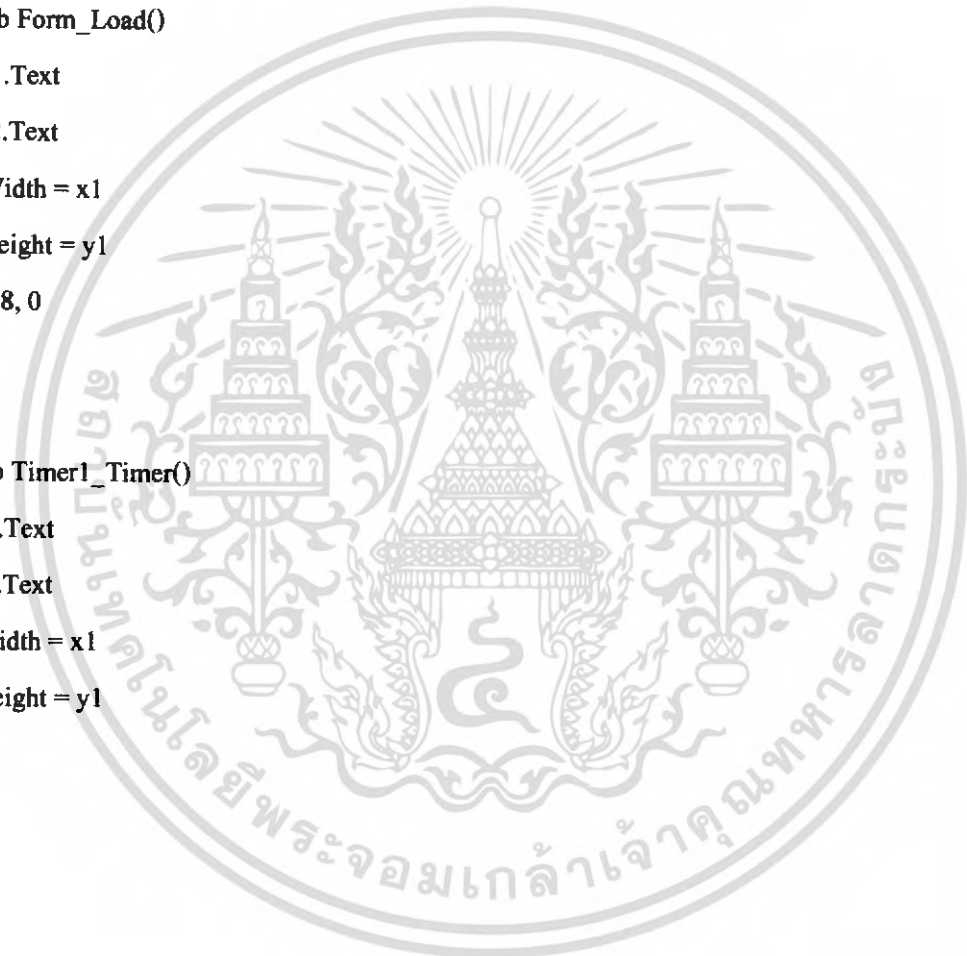
```
  x1 = Text1.Text
```

```
  y1 = Text2.Text
```

```
  Picture1.Width = x1
```

```
  Picture1.Height = y1
```

```
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1/2-Inch Megapixel CMOS Digital Image Sensor

MT9M001C12STM (Monochrome)

For the latest data sheet, refer to Micron's Web site: www.micron.com/imaging

Features

- Micron® DigitalClarity™ CMOS imaging technology
- Array Format (5:4): 1,280H x 1,024V (1,310,720 active pixels). Total (incl. dark pixels): 1,312H x 1,048V (1,374,976 pixels)
- Frame Rate: 30 fps progressive scan; programmable
- Shutter: Electronic Rolling Shutter (ERS)
- Window Size: SXGA; programmable to any smaller format (VGA, QVGA, CIF, QCIF, etc.)
- Programmable Controls: Gain, frame rate, frame size

Applications

- Digital still cameras
- Digital video cameras
- PC cameras

General Description

The Micron® Imaging MT9M001 is an SXGA-format with a 1/2-inch CMOS active-pixel digital image sensor. The active imaging pixel array of 1,280H x 1,024V. It incorporates sophisticated camera functions on-chip such as windowing, column and row skip mode, and snapshot mode. It is programmable through a simple two-wire serial interface.

This megapixel CMOS image sensor features DigitalClarity—Micron's breakthrough low-noise CMOS imaging technology that achieves CCD image quality (based on signal-to-noise ratio and low-light sensitivity) while maintaining the inherent size, cost, and integration advantages of CMOS.

Table 1: Key Performance Parameters

Parameter	Value
Optical format	1/2-inch (5:4)
Active imager size	6.66mm(H) x 5.32mm(V)
Active pixels	1,280H x 1,024V
Pixel size	5.2µm x 5.2µm
Shutter type	Electronic rolling shutter (ERS)
Maximum data rate/ master clock	48 MPS/48 MHz
Frame rate	SXGA (1280 x 1024) 30 fps progressive scan; programmable
ADC resolution	10-bit, on-chip
Responsivity	2.1 V/lux-sec
Dynamic range	68.2dB
SNR _{MAX}	45dB
Supply voltage	3.0V--3.6V, 3.3V nominal
Power consumption	363mW at 3.3V (operating); 294mW (standby)
Operating temperature	0°C to +70°C
Packaging	48-pin CLCC

The sensor can be operated in its default mode or programmed by the user for frame size, exposure, gain setting, and other parameters. The default mode outputs an SXGA-size image at 30 frames per second (fps). An on-chip analog-to-digital converter (ADC) provides 10 bits per pixel. FRAME_VALID and LINE_VALID signals are output on dedicated pins, along with a pixel clock that is synchronous with valid data.

Ordering Information

Table 2: Available Part Numbers

Part Number	Description
MT9M001C12STM	48-pin CLCC (lead free)



Table of Contents

Applications 1

General Description 1

 Ordering Information 3

List of Tables 4

List of Figures 5

General Description 7

Pixel Data Format 7

 Pixel Array Structure 8

 Output Data Format 8

 Output Data Timing 10

 Frame Timing Formulas 11

Serial Bus Description 11

 Protocol 11

 Sequence 11

 Bus Idle State 11

 Start Bit 12

 Stop Bit 12

 Slave Address 12

 Data Bit Transfer 12

 Acknowledge Bit 12

 No-Acknowledge Bit 12

Two-Wire Serial Interface Sample Write and Read Sequences 13

 16-Bit Write Sequence 13

 16-Bit Read Sequence 14

Registers 14

 Register Map 21

Feature Description 21

 Signal Path 21

 Programmable Gain Stage 22

 Programmable Analog Offset Stage 22

 Column and Row Mirror Image 22

 Column and Row Skip 22

 Black Level Calibration 23

Registers 25

 Still Image Capture with External Synchronization 27

 LINE_VALID Signal 28

Electrical Specifications 28

 Data Output and Propagation Delays 30

 Two-wire Serial Bus Timing 32

 Quantum Efficiency 32

 Image Center Offset and Orientation 32

Revision History 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



List of Tables

Table 1:	Key Performance Parameters.....	1
Table 2:	Available Part Numbers.....	1
Table 3:	Pin Descriptions.....	6
Table 4:	Frame Timing.....	10
Table 5:	Frame Time—Long Integration Time.....	10
Table 6:	Register List and Default Values.....	14
Table 7:	Register Description.....	15
Table 8:	Recommended Gain Settings at 48 MHz.....	21
Table 9:	Black Level Registers.....	25
Table 10:	DC Electrical Characteristics.....	28
Table 11:	AC Electrical Characteristics.....	29
Table 12:	Absolute Maximum Ratings.....	29
Table 13:	Optical Area Dimensions.....	33



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



List of Figures

Figure 1:	48-Pin CLCC Package Pinout Diagram	.5
Figure 2:	Block Diagram	.5
Figure 3:	Pixel Array Description	.7
Figure 4:	Pixel Pattern Detail (Top Right Corner)	.7
Figure 5:	Spatial Illustration of Image Readout	.8
Figure 6:	Timing Example of Pixel Data	.8
Figure 7:	Row Timing and FRAME_VALID/LINE_VALID Signals	.9
Figure 8:	Timing Diagram Showing a Write to Reg0x09 with the Value 0x0284	.13
Figure 9:	Timing Diagram Showing a Read from Reg0x09; Returned Value 0x0284	.13
Figure 10:	Signal Path	.21
Figure 11:	Readout of Six Columns in Normal and Column Mirror Output Mode	.22
Figure 12:	Readout of Six Rows in Normal and Row Mirror Output Mode	.22
Figure 13:	Readout of Eight Pixels in Normal and Column Skip Output Mode	.23
Figure 14:	Black Level Calibration Flow Chart	.24
Figure 15:	General Timing for Snapshot Mode	.27
Figure 16:	Different LINE_VALID Formats	.27
Figure 17:	Data Output Timing Diagram	.28
Figure 18:	Serial Host Interface Start Condition Timing	.30
Figure 19:	Serial Host Interface Stop Condition Timing	.30
Figure 20:	Serial Host Interface Data Timing for Write	.30
Figure 21:	Serial Host Interface Data Timing for Read	.30
Figure 22:	Acknowledge Signal Timing After an 8-Bit Write to the Sensor	.31
Figure 23:	Acknowledge Signal Timing After an 8-Bit Read from the Sensor	.31
Figure 24:	Quantum Efficiency—Monochrome	.32
Figure 25:	Image Center Offset	.32
Figure 26:	Optical Orientation	.33
Figure 27:	48-pin CLCC Package Outline Drawing	.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MT9M001: 1/2-Inch Megapixel Digital Image Sensor General Description

General Description

Figure 1: 48-Pin CLCC Package Pinout Diagram

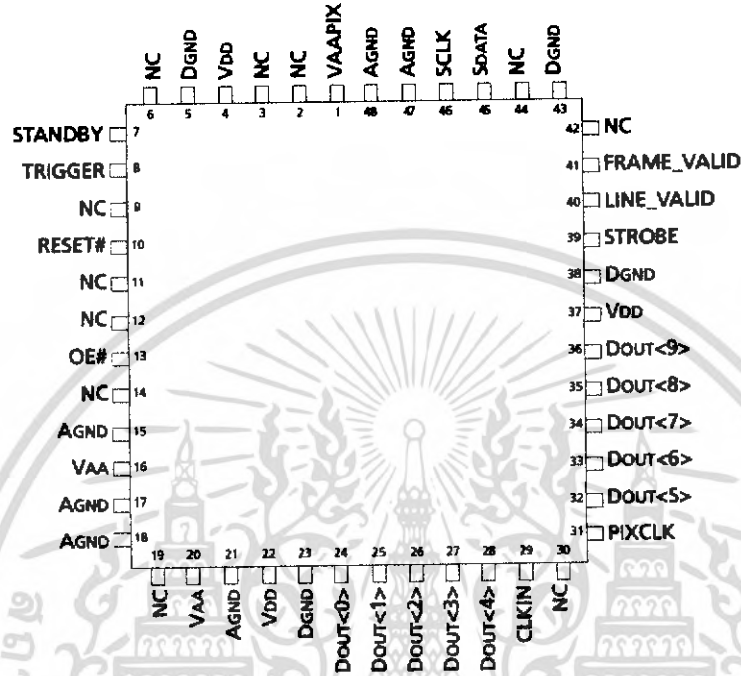
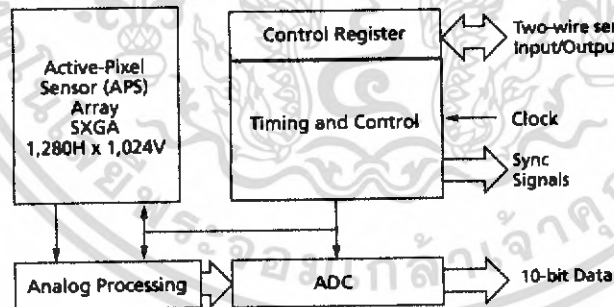


Figure 2: Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MT9M001: 1/2-Inch Megapixel Digital Image Sensor General Description

Table 3: Pin Descriptions

Pin Numbers	Symbol	Type	Description
29	CLKIN	Input	Clock in. Master clock into sensor (48 MHz maximum).
13	OE#	Input	Output enable. OE# when HIGH places outputs DOUT<0:9>, FRAME_VALID, LINE_VALID, PIXCLK, and STROBE into a tri-state configuration.
10	RESET#	Input	Reset. Activates (LOW) asynchronous reset of sensor. All registers assume factory defaults.
46	SCLK	Input	Serial clock. Clock for serial interface.
7	STANDBY	Input	Standby. Activates (HIGH) standby mode, disables analog bias circuitry for power saving mode.
8	TRIGGER	Input	Trigger. Activates (HIGH) snapshot sequence.
45	SDATA	Input/Output	Serial data. Serial data bus, requires 1.5K Ω resistor to 3.3V for pull-up.
24-28, 32-36	DOUT<0-9>	Output	Data out. Pixel data output bits 0:9, DOUT<9> (MSB), DOUT<0> (LSB).
41	FRAME_VALID	Output	Frame valid. Output is pulsed HIGH during frame of valid pixel data.
40	LINE_VALID	Output	Line valid. Output is pulsed HIGH during line of selectable valid pixel data (see Reg0x20 for options).
31	PIXCLK	Output	Pixel clock. Pixel data outputs are valid during falling edge of this clock. Frequency = (master clock).
39	STROBE	Output	Strobe. Output is pulsed HIGH to indicate sensor reset operation of pixel array has completed.
15,17,18,21, 47, 48	AGND	Supply	Analog ground. Provide isolated ground for analog block and pixel array.
5, 23, 38, 43	DGND	Supply	Digital ground. Provide isolated ground for digital block.
16, 20	VAA	Supply	Analog power. Provide power supply for analog block, 3.3V \pm 0.3V.
1	VAAPIX	Supply	Analog pixel power. Provide power supply for pixel array, 3.3V \pm 0.3V (3.3V).
4, 22, 37	VDD	Supply	Digital power. Provide power supply for digital block, 3.3V \pm 0.3V.
2, 3, 6, 9, 11, 12, 14, 19, 30, 42, 44	NC	—	No connect. These pins must be left unconnected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pixel Data Format

Pixel Array Structure

The MT9M001 pixel array is configured as 1,312 columns by 1,048 rows (shown in Figure 3). The first 16 columns and the first eight rows of pixels are optically black, and can be used to monitor the black level. The last seven columns and the last seven rows of pixels are also optically black. The black row data is used internally for the automatic black level adjustment. However, the black rows can also be read out by setting the sensor to raw data output mode (Reg0x20, bit 11 = 1). There are 1,289 columns by 1,033 rows of optically active pixels, which provides a four-pixel boundary around the SXGA (1,280 x 1,024) image.

Figure 3: Pixel Array Description

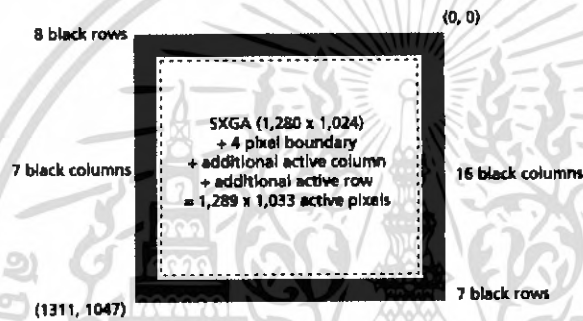
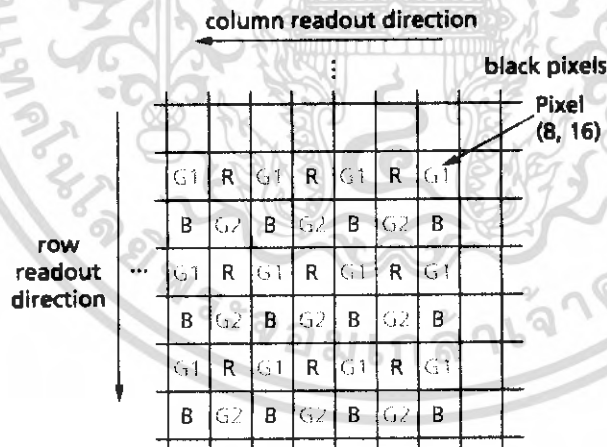


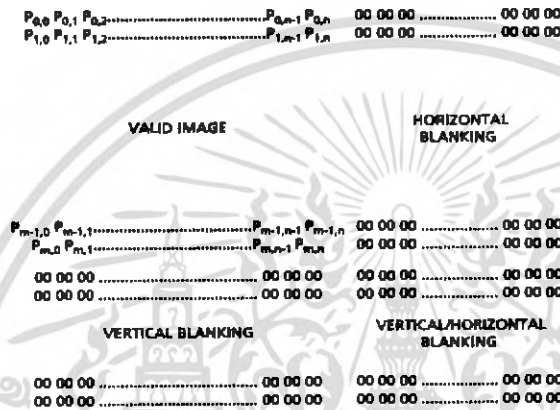
Figure 4: Pixel Pattern Detail (Top Right Corner)



Output Data Format

The MT9M001 image data is read out in a progressive scan. Valid image data is surrounded by horizontal blanking and vertical blanking, as shown in Figure 5. The amount of horizontal blanking and vertical blanking is programmable through Reg0x05 and Reg0x06, respectively. LINE_VALID is HIGH during the shaded region of the figure. FRAME_VALID timing is described in “Output Data Timing” on page 8.

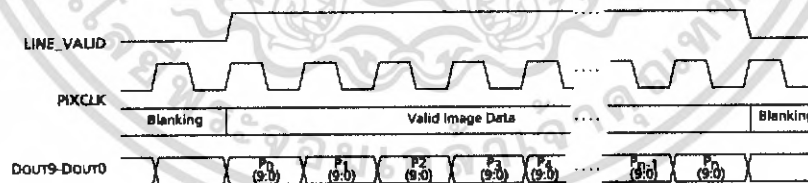
Figure 5: Spatial Illustration of Image Readout



Output Data Timing

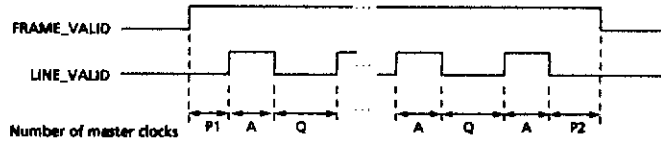
The data output of the MT9M001 is synchronized with the PIXCLK output. When LINE_VALID is HIGH, one 10-bit pixel datum is output every PIXCLK period.

Figure 6: Timing Example of Pixel Data



The rising edges of the PIXCLK signal are nominally timed to occur on the rising DOUT edges. This allows PIXCLK to be used as a clock to latch the data. DOUT data is valid on the falling edge of PIXCLK. The PIXCLK is HIGH while master clock is HIGH and then LOW while master clock is LOW. It is continuously enabled, even during the blanking period. The parameters P1, A P2, and Q in Figure 7 are defined in Table 4.

Figure 7: Row Timing and FRAME_VALID/LINE_VALID Signals





**Frame Timing Formulas
Table 4: Frame Timing**

Parameter	Name	Equation (MASTER CLOCK)	Default Timing	Notes
A	Active Data Time	(Reg0x04 + 1)	1,280 pixel clocks = 26.7µs	1
P ₁	Frame Start Blanking	(242)	242 pixel clocks = 5.04µs	
P ₂	Frame End Blanking	(2 + Reg0x05 - 19) (MIN Reg0x05 value = 19)	2 pixel clocks = 0.042µs	2
Q = P ₁ + P ₂	Horizontal Blanking	(244 + Reg0x05 - 19) (MIN Reg0x05 value = 19)	244 pixel clocks = 5.08µs	2
A + Q	Row Time	((Reg0x04 + 1) + (244 + Reg0x05 - 19))	1,524 pixel clocks = 31.75µs	
V	Vertical Blanking	(Reg0x06 + 1) x (A + Q) (MIN Reg0x06 value = 15)	39,624 pixel clocks = 825.5µs	
N _{ROWS} x (A + Q)	Frame Valid Time	(Reg0x03 + 1) x (A + Q)	1,560,576 pixel clocks = 32.51ms	
F	Total Frame Time	(Reg0x03 + 1 + Reg0x06 + 1) x (A + Q)	1,600,200 pixel clocks = 33.34ms	

- Notes:
1. Row skip mode should have no effect on the integration time. Column skip mode changes the effective value of Column Size (Reg0x04) as follows:
 Column Skip 2 => R4eff = (int(R4 / 4) x 2) + 1
 Column Skip 4 => R4eff = (int(R4 / 8) x 2) + 1
 Column Skip 8 => R4eff = (int(R4 / 16) x 2) + 1
 where the int() function truncates to the next lowest integer. Now use R4eff in the equation for row time instead of R4
 2. Default for Reg0x05 = 9. However, sensor ignores any value for Reg0x05 less than 19.

Sensor timing is shown above in terms of pixel clock and master clock cycles (please refer to Figure 6). The recommended master clock frequency is 48 MHz. The vertical blank and total frame time equations assume that the number of integration rows (bits 13 through 0 of Reg0x09) is less than the number of active plus blanking rows (Reg0x03 + 1 + Reg0x06 + 1). If this is not the case, the number of integration rows must be used instead to determine the frame time, as shown in Table 5.

Table 5: Frame Time—Long Integration Time

Parameter	Name	Equation (master clock)	Default Timing
V'	Vertical Blanking (long integration time)	(Reg0x09 - Reg0x03) x (A + Q)	39,624 pixel clocks = 82.5µs
F'	Total Frame Time (long integration time)	(Reg0x09 + 1) x (A + Q)	1,600,200 pixel clocks = 33.34ms

Serial Bus Description

Registers are written to and read from the MT9M001 through the two-wire serial interface bus. The sensor is a two-wire serial interface slave and is controlled by the serial clock (SCLK), which is driven by the serial interface master. Data is transferred into and out of the MT9M001 through the serial data (SDATA) line. The SDATA line is pulled up to 3.3V off-chip by a 1.5K Ω resistor. Either the slave or master device can pull the SDATA line down—the serial interface protocol determines which device is allowed to pull the SDATA line down at any given time.

Protocol

The two-wire serial interface defines several different transmission codes, as follows:

- a start bit
- the slave device eight-bit address
- a(an) (no) acknowledge bit
- an 8-bit message
- a stop bit

Sequence

A typical read or write sequence begins by the master sending a start bit. After the start bit, the master sends the slave device's eight-bit address. The last bit of the address determines if the request will be a read or a write, where a "0" indicates a write and a "1" indicates a read. The slave device acknowledges its address by sending an acknowledge bit back to the master.

If the request was a write, the master then transfers the 8-bit register address to which a write should take place. The slave sends an acknowledge bit to indicate that the register address has been received. The master then transfers the data eight bits at a time, with the slave sending an acknowledge bit after each eight bits. The MT9M001 uses 16-bit data for its internal registers, thus requiring two 8-bit transfers to write to one register. After 16 bits are transferred, the register address is automatically incremented, so that the next 16 bits are written to the next register address. The master stops writing by sending a start or stop bit.

A typical read sequence is executed as follows. First the master sends the write-mode slave address and 8-bit register address, just as in the write request. The master then sends a start bit and the read-mode slave address. The master then clocks out the register data eight bits at a time. The master sends an acknowledge bit after each 8-bit transfer. The register address is auto-incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Bus Idle State

The bus is idle when both the data and clock lines are HIGH. Control of the bus is initiated with a start bit, and the bus is released with a stop bit. Only the master can generate the start and stop bits.

Start Bit

The start bit is defined as a HIGH-to-LOW transition of the data line while the clock line is HIGH.



Stop Bit

The stop bit is defined as a LOW-to-HIGH transition of the data line while the clock line is HIGH.

Slave Address

The 8-bit address of a two-wire serial interface device consists of seven bits of address and 1 bit of direction. A "0" (0xBA) in the LSB (least significant bit) of the address indicates the write mode, and a "1" (0xBB) indicates read mode.

Data Bit Transfer

One data bit is transferred during each clock pulse. The serial interface clock pulse is provided by the master. The data must be stable during the HIGH period of the two-wire serial interface clock—it can only change when the serial clock is LOW. Data is transferred eight bits at a time, followed by an acknowledge bit.

Acknowledge Bit

The master generates the acknowledge clock pulse. The transmitter (which is the master when writing, or the slave when reading) releases the data line, and the receiver indicates an acknowledge bit by pulling the data line LOW during the acknowledge clock pulse.

No-Acknowledge Bit

The no-acknowledge bit is generated when the data line is not pulled down by the receiver during the acknowledge clock pulse. A no-acknowledge bit is used to terminate a read sequence.

Two-Wire Serial Interface Sample Write and Read Sequences

16-Bit Write Sequence

A typical write sequence for writing 16 bits to a register is shown in Figure 8. A start bit given by the master, followed by the write address, starts the sequence. The image sensor will then give an acknowledge bit and expects the register address to come first, followed by the 16-bit data. After each eight-bit transfer, the image sensor will give an acknowledge bit. All 16 bits must be written before the register will be updated. After 16 bits are transferred, the register address is automatically incremented so that the next 16 bits are written to the next register. The master stops writing by sending a start or stop bit.

Figure 8: Timing Diagram Showing a Write to Reg0x09 with the Value 0x0284



16-Bit Read Sequence

A typical read sequence is shown in Figure 9. First the master has to write the register address, as in a write sequence. Then a start bit and the read address specifies that a read is about to happen from the register. The master then clocks out the register data eight bits at a time. The master sends an acknowledge bit after each eight-bit transfer. The register address should be incremented after every 16 bits is transferred. The data transfer is stopped when the master sends a no-acknowledge bit.

Figure 9: Timing Diagram Showing a Read from Reg0x09; Returned Value 0x0284





Registers

Register Map

Table 6: Register List and Default Values

Register # (Hex)	Description	Data Format (Binary)	Default Value (Hex)
0x00	Chip Version	1000 0100 0001 0001	0x8431
0x01	Row Start	0000 0ddd dddd dddd	0x000C
0x02	Column Start	0000 0ddd dddd dddd	0x0014
0x03	Row Size (Window Height)	0000 0ddd dddd dddd	0x03FF
0x04	Col Size (Window Width)	0000 0ddd dddd dddd	0x04FF
0x05	Horizontal Blanking	0000 0ddd dddd dddd	0x0009
0x06	Vertical Blanking	0000 0ddd dddd dddd	0x0019
0x07	Output Control	0000 0000 0d00 00dd	0x0002
0x09	Shutter Width	00dd dddd dddd dddd	0x0419
0x0B	Restart	0000 0000 0000 000d	0x0000
0x0C	Shutter Delay	0000 0ddd dddd dddd	0x0000
0x0D	Reset	0000 0000 0000 000d	0x0000
0x1E	Read Options 1	1000 dddd 00dd dd00	0x8000
0x20	Read Options 2	dd01 0dd1 d00d d10d	0x1104
0x2B	Even Row, Even Column	0000 0000 0ddd dddd	0x0008
0x2C	Odd Row, Even Column	0000 0000 0ddd dddd	0x0008
0x2D	Even Row, Odd Column	0000 0000 0ddd dddd	0x0008
0x2E	Odd Row, Odd Column	0000 0000 0ddd dddd	0x0008
0x35	Global Gain	0000 0000 0ddd dddd	0x0008
0x5F	Cal Threshold	dddd dddd d0dd dddd	0x0904
0x60	Even Row, Even Column	0000 000d dddd dddd	0x0000
0x61	Odd Row, Odd Column	0000 000d dddd dddd	0x0000
0x62	Cal Ctrl	d00d d100 1001 1ddd	0x0498
0x63	Even Row, Odd Column	0000 000d dddd dddd	0x0000
0x64	Odd Row, Even Column	0000 000d dddd dddd	0x0000
0xF1	Chip Enable	0000 0000 0000 00dd	0x0001

Note: 1 = always 1
0 = always 0
d = programmable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MT9M001: 1/2-Inch Megapixel Digital Image Sensor Registers

Table 7: Register Description

Register	Bit	Description
Chip ID		
0x00	15:0	This register is read-only and gives the chip identification number: 0x8431.
Window Control These registers control the size of the window.		
0x01	10:0	First row to be read out—default = 0x000C (12).
0x02	10:0	First column to be read out—default = 0x0014 (20). Register value must be an even number.
0x03	10:0	Window height (number of rows - 1)—default = 0x03FF (1023). Minimum value for 0x03 = 0x0002.
0x04	10:0	Window width (number of columns - 1)—default = 0x04FF (1279). Register value must be an odd number. Minimum value for 0x04 = 0x0003.
Blanking Control These registers control the blanking time in a row (called column fill-in or horizontal blanking) and between frames (vertical blanking). Horizontal blanking is specified in terms of pixel clocks. Vertical blanking is specified in terms of row readout times. The actual imager timing can be calculated using Table 4, "Frame Timing," on page 10.		
0x05	10:0	Horizontal Blanking—default = 0x0009 (9 pixels).
0x06	10:0	Vertical Blanking—default = 0x0019 (25 rows).
Output Control This register controls various features of the output format for the sensor.		
0x07	0	Synchronize changes (copied to Reg0xF1, bit1). 0 = normal operation. Update changes to registers that affect image brightness (integration time, integration delay, gain, horizontal blanking and vertical blanking, window size, row/column skip or row mirror) at the next frame boundary. The "frame boundary" is 8 row_times before the rising edge of FRAME_VALID. (If "Show Dark Rows" is set, it will be coincident with the rising edge of FRAME_VALID.) 1 = do not update any changes to these settings until this bit is returned to "0."
	1	Chip Enable (copied to Reg0xF1, bit0). 1 = normal operation. 0 = stop sensor readout. When this is returned to "1," sensor readout restarts at the starting row in a new frame. The digital power consumption can then also be reduced to less than 5uA by turning off the master clock.
	2	Reserved—default is 0; set to zero at all times.
	3	Reserved—default is 0; set to zero at all times.
	6	Use Test Data. When set, a test pattern will be output instead of the sampled image from the sensor array. The value sent to the Dout[9:0] pins will alternate between the Test Data register (Reg0x32) in even columns and the inverse of the Test Data register for odd columns. The output "image" will have the same width, height, and frame rate as it would otherwise have. No digital processing (gain or offset) is applied to the data. When clear (the default), sampled pixel values are output normally.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table 7: Register Description (continued)

Register	Bit	Description
Pixel Integration Control		
<p>These registers (along with the window sizing and blanking registers) control the integration time for the pixels. The actual total integration time (t_{INT}) is:</p> <p>$t_{INT} = \text{Reg0x09} \times \text{row time} - \text{overhead time} - \text{reset delay}$, where:</p> <p>Row time = $((\text{Reg0x04} + 1) + 244 + \text{Reg0x05} - 19)$ pixel clock periods</p> <p>Overhead time = 180 pixel clock periods</p> <p>Reset delay = $4 \times \text{Reg0x0C}$ pixel clock periods</p> <p>If the value in Reg0x0C exceeds $(\text{row time} - 548)/4$ pixel clock cycles, the row time will be extended by $(4 \times \text{Reg0x0C} - (\text{row time} - 548))$ pixel clock cycles.</p> <p>In this expression, the row time term, $\text{Reg0x09} \times ((\text{number of columns}) + 244 + \text{horizontal blanking register} - 19)$, corresponds to the number of rows integrated. The overhead time (180 pixel clocks) is the overhead time between the READ cycle and the RESET cycle, and the final term is the effect of the reset delay.</p> <p>Typically, the value of Reg0x09 is limited to the number of rows per frame (which includes vertical blanking rows) such that the frame rate is not affected by the integration time. If Reg0x09 is increased beyond the total number of rows per frame, the MT9M001 will add additional blanking rows as needed. A second constraint is that t_{INT} must be adjusted to avoid banding in the image from light flicker. Under 60Hz flicker, this means t_{INT} must be a multiple of 1/120 of a second. Under 50Hz flicker, t_{INT} must be a multiple of 1/100 of a second.</p>		
0x09	13:0	Number of rows of integration—default = 0x0419 (1049).
0x0C	10:0	Shutter delay—default = 0x0000 (0). This is the number of master clocks times four that the timing and control logic waits before asserting the reset for a given row.
Frame Restart		
0x0B	0	Setting bit 0 to "1" of Reg0x0B will cause the sensor to abandon the readout of the current frame and restart from the first row. This register automatically resets itself to 0x0000 after the frame restart. The first frame after this event is considered to be a "bad frame" (see description for Reg0x20, bit0).
Reset		
0x0D	0	This register is used to reset the sensor to its default, power-up state. To put the MT9M001 in reset mode first write a "1" into bit 0 of this register, then write a "0" into bit 0 to resume operation.

Table 7: Register Description (continued)

Register	Bit	Description
Read Mode 1		
In read mode 1, this register is used to control many aspects of the readout of the sensor.		
0x1E	0	Reserved—default is 0; set to zero at all times.
	1	Reserved—default is 0; set to zero at all times.
	2	Column Skip 4—default is 0 (disable). 1 = enable.
	3	Row Skip 4—default is 0 (disable). 1 = enable.
	4	Column Skip 8—default is 0 (disable). 1 = enable.
	5	Row Skip 8—default is 0 (disable). 1 = enable.
	6	Reserved—default is 0; do not change.
	7	Reserved—default is 0; do not change.
	8	Snapshot Mode—default is 0 (continuous mode). 1 = enable (wait for TRIGGER; TRIGGER can come from outside signal (TRIGGER pin on the sensor) or from serial interface register restart, i.e. programming a “1” to bit 0 of Reg0x0B).
	9	STROBE Enable—default is 0 (no STROBE signal). 1 = enable STROBE (signal output from the sensor during the time all rows are integrating. See STROBE width for more information).
	10	STROBE Width—default is 0 (STROBE signal width at minimum length, 1 row of integration time, prior to line valid going HIGH). 1 = extend STROBE width (STROBE signal width extends to entire time all rows are integrating).
	11	Strobe Override—default is 0 (STROBE signal created by digital logic). 1 = override STROBE signal (STROBE signal is set HIGH when this bit is set, LOW when this bit is set LOW. It is assumed that STROBE enable is set to “0” if STROBE override is being used).
	12	Reserved—default is 0; do not change.
	13	Reserved—default is 0; do not change.
	14	Reserved—default is 0; do not change.
15	Reserved—default is 1; do not change.	
Gain Settings		
The gain is individually controllable for each of the four groups of pixels that lie in odd rows and columns, even rows and columns, odd rows and even columns, and even rows and odd columns. This is shown in the register chart.		
Formula for gain setting:		
Gain ≤ 8 Gain = (bit[6] + 1) × (bit[5-0] × 0.125)		
Gain > 8 (bit[6] = 1 and bit[5] = 1) Gain = 8.0 + bit[2-0]		
Since bit[6] of the gain registers are multiplicative factors for the gain settings, there are alternative ways of achieving certain gains. Some settings offer superior noise performance to others, despite the same overall gain. The following lists the recommended gain settings:		
Gain	Increments	Recommended Settings
1.000 to 4.000	0.125	0x08 to 0x20
4.25 to 8.00	0.25	0x51 to 0x60
9.0 to 15.0	1.0	0x61 to 0x67
0x2B	6:0	Even row, even column—default = 0x08 (8) = 1x gain.
0x2C	6:0	Odd row, even column—default = 0x08 (8) = 1x gain.

Table 7: Register Description (continued)

Register	Bit	Description
0x2D	6:0	Even row, odd column—default = 0x08 (8) = 1x gain.
0x2E	6:0	Odd row, odd column—default = 0x08 (8) = 1x gain.
0x35	6:0	Global gain—default = 0x08 (8) = 1x gain. This register can be used to set all four gains at once.
Test Data		
0x32	11:2	Test Data. The value used to produce a test pattern in "Use Test Data" mode (Reg0x07 bit 6).
Read Mode 2		
This register is used to control many aspects of the readout of the sensor.		
0x20	0	No bad frames—1 = output all frames (including bad frames). 0 (default) = only output good frames. A bad frame is defined as the first frame following a change to: window size or position, horizontal blanking, row or column skip, or mirroring.
	1	Reserved—default is 0; do not change.
	2	Reserved—default is 1; set to "1" at all times.
	3	Column skip—1 = read out two columns, and then skip two columns (for example, col 0, col 1, col 4, col 5...), 0 = normal readout (default).
	4	Row skip—1 = read out two rows, and then skip two rows (for example, row 0, row 1, row 4, row 5...), 0 = normal readout (default).
	5	Reserved—default is 0; do not change.
	6	Reserved—default is 0; set to zero at all times.
	7	Flip Row—1 = readout starting 1 row later (alternate color pair). 0 (default) = normal readout.
	8	Reserved—default is 1; set to "1" at all times.
	9	1 = "Continuous" LINE_VALID (continue producing LINE_VALID during vertical blanking). 0 = normal LINE_VALID (default, no LINE_VALID during vertical blanking).
	10	1 = LINE_VALID = "Continuous" LINE_VALID XOR FRAME_VALID. 0 = LINE_VALID determined by bit 9.
	11	Reserved—default is 0; do not change.
	12	Reserved—default is 1; do not change.
	13	Reserved—default is 0; do not change.
	14	Reserved—default is 0; do not change.
	15	Mirror Row—1 = read out from bottom to top (upside down). 0 (default) = normal readout (top to bottom).
Test Data		
0x32	11:2	Test Data. The value used to produce a test pattern in "Use Test Data" mode (Reg0x07 bit 6).
Black Level Calibration		
These registers are used in the black level calibration. Their functionality is described in detail in the next section.		

Table 7: Register Description (continued)

Register	Bit	Description
0x5F	5:0	Thres_lo—Lower threshold for black level in ADC LSBs—default = 000100.
	7	1 = override automatic Thres_hi and Thres_lo adjust (Thres_hi always = bits 14:8; Thres_lo always = bits 5:0). Default = 0 = Automatic Thres_hi and Thres_lo adjustment.
	14:8	Thres_hi—Maximum allowed black level in ADC LSBs (default = Thres_lo + 5). Black level maximum is set to this value when bit 7 = 1; black level maximum is reset to this value after every black level average restart if bit 15 = 1 and bit 7 = 0.
	15	No gain dependence. 1 = Thres_lo is set by the programmed value of bits 5:0, Thres_hi is reset to the programmed value (bits 14:8) after every black level average restart. 0 = Thres_lo and Thres_hi are set automatically, as described above.
0x60	8:0	Even row, even column—analog offset correction value for even row, even column, bits 0:7 sets magnitude, bit 8 set sign. 0 = positive; 1 = negative. two's complement, if bit 8 = 1, Offset = bits [0:7] - 256.
0x61	8:0	Odd row, odd column—analog offset correction value for odd row, odd column, bits 0:7 sets magnitude, bit 8 set sign. 0 = positive; 1 = negative. two's complement, if bit 8 = 1, Offset = bits [0:7] - 256.
0x62	0	Manual override of black level correction. 1 = override automatic black level correction with programmed values. 0 = normal operation (default).
	2:1	Force/disable black level calibration. 00 = apply black level calibration during ADC operation only (default). 10 = apply black level calibration continuously. X1 = disable black level correction (Offset Correction Voltage = 0.0V). (In this case, no black level correction is possible).
	4:3	Reserved—default is 1; do not change.
	6:5	Reserved—default is 0; do not change.
	7	Reserved—default is 1; do not change.
	9:8	Reserved—default is 0; do not change.
	10	Reserved—default is 1; do not change.
	11	1 = do not reset the upper threshold after a black level recalculation sweep. 0 = reset the upper threshold after a black level recalculation sweep (default).
	12	1 = start a new running digitally filtered average for the black level (this is internally reset to "0" immediately), and do a rapid sweep to find the new starting point. 0 = normal operation (default).
	14:3	Reserved—default is 0; set to zero at all times.
0x63	15	1 = do not perform the rapid black level sweep on new gain settings. 0 = normal operation.
	8:0	Even row, odd column—analog offset correction value for even row, odd column, bits 0:7 sets magnitude, bit 8 set sign. 0 = positive; 1 = negative. two's complement, if bit 8 = 1, Offset = bits [0:7] - 256.
0x64	8:0	Odd row, even column—analog offset correction value for odd row, even column, bits 0:7 sets magnitude, bit 8 set sign. 0 = positive; 1 = negative. two's complement, if bit 8 = 1, Offset = bits [0:7] - 256.

Table 7: Register Description (continued)

Register	Bit	Description
Chip Enable and Two-Wire Serial Interface Write Synchronize.		
0xF1	0	Mirrors the functionality of Reg0x07 bit1 (Chip Enable). 1 = normal operation. 0 = stop sensor readout; when this is returned to "1," sensor readout restarts at the starting row in a new frame.



Feature Description

Signal Path

The MT9M001 signal path consists of two stages, a programmable gain stage and a programmable analog offset stage.

Programmable Gain Stage

A total programmable gain of 15 is available and can be calculated using the following formula:

$$\text{Gain 1 to 8: Gain} = (\text{bit}[6] + 1) \times (\text{bit}[5:0] \times 0.125)$$

For gain higher than eight, the user would need to set bit[6:5] = 11 and use the lower 3 LSB's bit[2:0] to set the higher gain values. The formula for obtaining gain greater than eight is as follows:

$$\text{Total gain} = 8 + \text{bit}[2:0]$$

For example, for total gain = 12, the value to program is bit[6:0] = 1100100.

The maximum total gain = 15, i.e. bit[6:0] = 1100111.

The gain circuitry in the MT9M001 is designed to offer signal gains from one to 15. The minimum gain of one corresponds to the lowest setting where the pixel signal is guaranteed to saturate the ADC under all specified operating conditions. Any reduction of the gain below this value may cause the sensor to saturate at ADC output values less than the maximum, under certain conditions. It is recommended that this guideline be followed at all times.

Since bit[6] of the gain registers are multiplicative factors for the gain settings, there are alternative ways of achieving certain gains. Some settings offer superior noise performance to others, despite the same overall gain. Recommended gain settings are listed in Table 8.

Figure 10: Signal Path

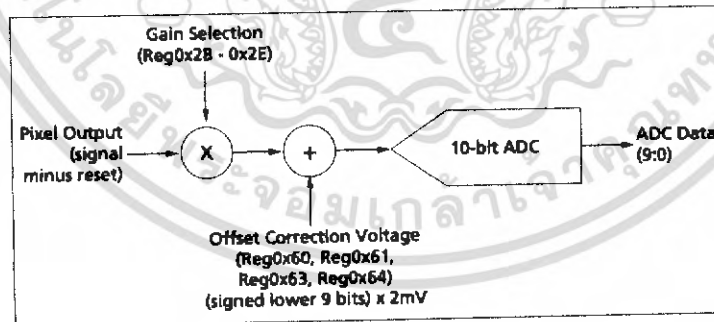
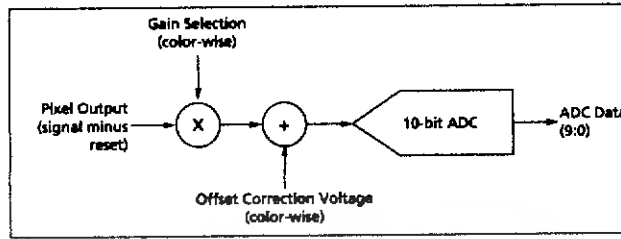


Table 8: Recommended Gain Settings at 48 MHz

Nominal Gain	Increments	Recommended Settings
1 to 4.000	0.125	0x08 to 0x20
4.25 to 8.00	0.25	0x51 to 0x60
9 to 15	1.0	0x61 to 0x67

Figure 14: Black Level Calibration Flow Chart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Registers

Table 9: Black Level Registers

Register	bit	Description
Reg0x5F This register controls the operation of the black level calibration thresholds.		
	15	No gain dependence. 1 = Thres_lo is set by the programmed value of bits 5:0, Thres_hi is reset to the programmed value (bits 14:8) after every black level average restart. 0 = Thres_lo and Thres_hi are set automatically as described below.
	14:8	Thres_hi—maximum allowed black level in ADC LSBs (default = Thres_lo + 5). Black level maximum is set to this value when bit 7 = 1, black level maximum is reset to this value after every black level average restart if bit 15 = 1 and bit 7 = 0.
	7	1 = override automatic Thres_hi and Thres_lo adjust (Thres_hi always = bits 14:8, Thres_lo always = bits 5:0). 0 = automatic Thres_hi and Thres_lo adjustment.
	5:0	Thres_lo—Lower threshold for black level in ADC LSBs. Under default automatic operation (bit 7 = 0, bit 15 = 0), $Thres_lo = RegGain_{max}/4 \times (RegGain_{max}, bit\ 6 + 1) \times (RegGain_{max}, bit\ 7 + 1)$, where $RegGain_{max}$ is the maximum of the four independent gain register settings. Whenever a jump in the calibration causes the black level data to change from below Thres_lo to above Thres_hi, Thres_hi is adjusted according to the following: If new black level < 64: $Thres_hi = Thres_lo + 2 + (2 \times Delta)$, where $Delta = new\ black\ level - Thres_lo$ If new black level > 63 and < 119: $Thres_hi = new\ black\ level + 4$ If new black level > 119: $Thres_hi = 123$ After any recalculation of the black level and average restart, Thres_hi is reset to either Thres_lo + 5 (automatic, default mode), Thres_hi (bit 7 = 1). Reg0x62, bit 11 will override this.
Reg0x62 This register is used to control the automatic black level calibration circuitry.		
	15	1 = do not perform the rapid black level sweep on new gain settings. 0 = normal operation.
	14	Reserved—default is 0; do not change.
	13	Reserved—default is 0; do not change.
	12	1 = start a new running digitally filtered average for the black level (this is internally reset to "0" immediately), and do a rapid sweep to find the new starting point.
	11	1 = do not reset the upper threshold after a black level recalculation sweep. 0 = reset the upper threshold after a black level recalculation sweep (default).
	10:3	Reserved—default is 1; do not change.
	2:1	Force/disable black level calibration. 00 = apply black level calibration during ADC operation only (default). 10 = apply black level calibration continuously. X1 = disable black level correction (Offset Correction Voltage = Skew Voltage = 0.0V). (In this case, no black level correction is possible).
	0	Manual override of black level correction. 1 = override automatic black level correction with programmed values. 0 = normal operation (default).

Registers

Table 9: Black Level Registers (continued)

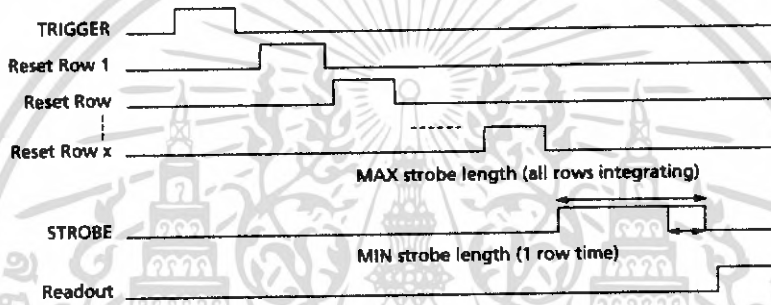
Register	bit	Description
Reg0x60, Reg0x61, Reg0x63, Reg0x64		<p>These registers contain the 9-bit signed black level calibration values. In normal operation, these values are calculated at the beginning of each frame. However, if Reg0x62, bit 0 is set to "1," these registers can be written to, overriding the automatic black level calculation. This feature can be used in conjunction with readout of the black rows (Reg0x20, bit 11) if the user would like to use an external black level calibration circuit. The offset correction voltage is generated according to the following formula:</p> <p style="text-align: center;"> $\text{Offset Correction Voltage} = (9\text{-bit signed calibration value, } -256 \text{ to } 255) \times (2\text{mV} \times \text{Enable bit})$ $\text{two's complement, if bit 8} = 1, \text{ Offset} = \text{bits [0:7]} - 256$ </p> <p style="text-align: center;"> $\text{ADC input voltage} = \text{Pixel Output Voltage} \times \text{Analog Gain} - \text{Offset Correction Voltage}$ </p>



Still Image Capture with External Synchronization

In continuous mode video image capture, the TRIGGER signal should be held LOW or “0.” To capture a still image, the sensor must first be put into snapshot mode by programming a “1” in register 0x1E, bit 8. In snapshot mode, the sensor waits for a TRIGGER signal (FRAME_VALID, LINE_VALID signals are LOW, pixel clock signal continues). When the TRIGGER signal is received (active HIGH), one frame is read out (a TRIGGER signal can also be achieved by programming a restart—for example, program a “1” to bit 0 of Reg0x0B). The reset, readout timing for that frame will be the same as for a continuous frame with similar register settings; the only difference is that only one frame is read out. General timing for the snapshot mode is shown in Figure 15.

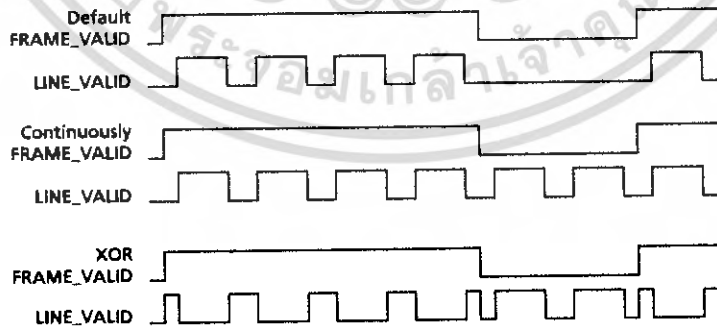
Figure 15: General Timing for Snapshot Mode



LINE_VALID Signal

By setting bit 9 and 10 of Reg0x20 the line valid signal can get three different output formats. The formats are shown when reading out four rows and two vertical blanking rows (Figure 16). In the last format, the LINE_VALID signal is the XOR between the continuously LINE_VALID signal and the FRAME_VALID signal.

Figure 16: Different LINE_VALID Formats





Electrical Specifications

Data Output and Propagation Delays

By default, the MT9M001 launches pixel data, FRAME_VALID and LINE_VALID with the rising edge of PIXCLK. The expectation is that the user captures DOUT[7:0], FRAME_VALID and LINE_VALID using the falling edge of PIXCLK.

Figure 17: Data Output Timing Diagram

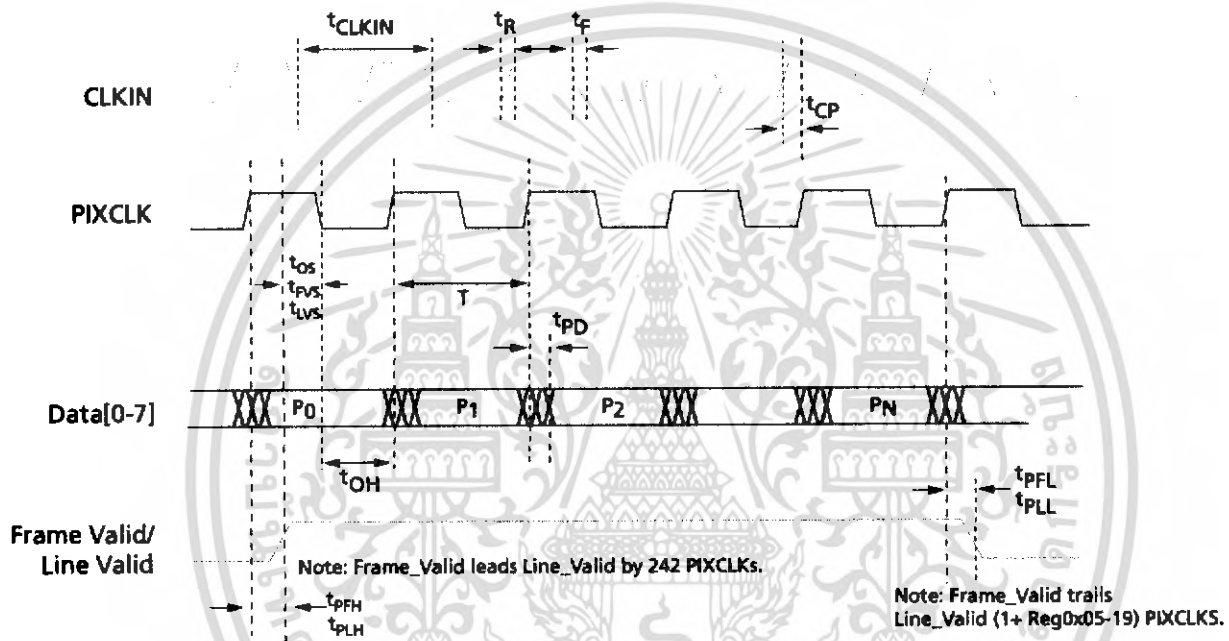


Table 10: DC Electrical Characteristics

(DC Setup Conditions: $f_{CLKIN} = 48 \text{ MHz}$, $V_{DD} = 3.3\text{V}$, $V_{AA} = 3.3\text{V}$, $V_{AAPIX} = 3.3\text{V}$, $T_A = 25^\circ\text{C}$)

Symbol	Definition	Condition	Min	Typ	Max	Units
VDD	Core digital voltage		3	3.3	3.6	V
VAA	Analog voltage		3	3.3	3.6	V
VAAPIX	Pixel supply voltage		3	3.3	3.6	V
VIH	Input high voltage		$V_{PWR} - 0.3$		$V_{PWR} + 0.3$	V
VIL	Input low voltage		-0.3		0.8	V
IIN	Input leakage current	No Pull-up Resistor; $V_{IN} = V_{DD}$ or DGND	-15		15	μA
VOH	Output high voltage		$V_{PWR} - 0.2$		—	V
VOL	Output low voltage				0.2	V
IOZ	Tri-state output leakage current		—		15	μA
IDD	Digital operating current		—	20	24	mA
IAA	Analog operating current		—	85	110	mA



MT9M001: 1/2-Inch Megapixel Digital Image Sensor Electrical Specifications

Table 10: DC Electrical Characteristics (continued)

(DC Setup Conditions: $f_{CLKIN} = 48$ MHz, $V_{DD} = 3.3V$, $V_{AA} = 3.3V$, $V_{AAPIX} = 3.3V$, $T_A = 25^\circ C$)

Symbol	Definition	Condition	Min	Typ	Max	Units
IAAPIX	Pixel supply current		—	5	10	mA
ISTDBYD	Digital standby current	STDBY = V_{DD} , CLKIN = 0 MHz	—	9	20	mA
ISTDBYD w/CLK	Digital standby current	STDBY = V_{DD} , CLKIN = 48 MHz	—	55	125	μA
ISTDBYDA	Analog standby current	STDBY = V_{DD}	—	80	100	μA

Table 11: AC Electrical Characteristics

(AC Setup Conditions: $f_{CLKIN} = 48$ MHz, $V_{DD} = 3.3V$, $V_{AA} = 3.3V$, $V_{AAPIX} = 3.3V$, Output Load = 30pF, $T_A = 25^\circ C$)

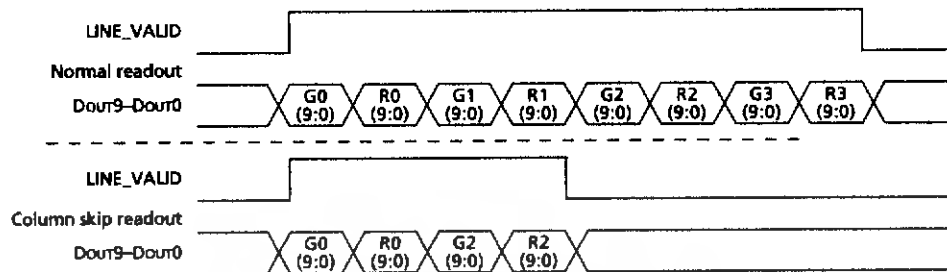
Symbol	Definition	Condition	Min	Typ	Max	Unit
f_{CLKIN}	Input clock frequency		1	—	48	MHz
T_{CLKIN}	Input clock period		1000	—	20.83	ns
T	PIXCLK period		1000	—	20.83	ns
t_R	Input clock rise time		—	4	—	V/ns
t_F	Input clock fall time		—	4	—	V/ns
	Clock duty cycle		45	50	55	%
t_{CP}	CLKIN to PIXCLK propagation delay		—	10	—	ns
t_{PD}	PIXCLK to data valid		—	—	1	ns
t_{PFH}	PIXCLK to FV high		—	—	7	ns
t_{PLH}	PIXCLK to LV high		—	—	7	ns
t_{PFL}	PIXCLK to FV low		—	—	13	ns
t_{PLL}	PIXCLK to LV low		—	—	13	ns
t_{OS}	Setup time for data before falling edge of PIXCLK		$T/2 - 1$	$T/2$	$T/2 + 1$	ns
t_{OH}	Hold time for data after falling edge of PIXCLK		$T/2 - 1$	$T/2$	$T/2 + 1$	ns
t_{FVS}	Setup time for FV before falling edge of PIXCLK		2	3	—	ns
t_{LVS}	Setup time for LV before falling edge of PIXCLK		2	3	—	ns
CLOAD	Load capacitance		—	—	30	pF

Table 12: Absolute Maximum Ratings

Symbol	Parameter	Rating		Unit
		MIN	MAX	
TOP	Operating temperature	0	70	$^\circ C$
TSTG ¹	Storage temperature	-40	125	$^\circ C$

Note: ¹Stresses greater than those listed may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Figure 13: Readout of Eight Pixels in Normal and Column Skip Output Mode



Black Level Calibration

The MT9M001 has automatic black level calibration on-chip which can be overridden by the user, as described below and shown in Figure 14.

The automatic black level calibration measures the average value of 256 pixels from two dark rows of the chip for each of the four colors. The pixels are averaged as if they were light-sensitive and passed through the appropriate color gain. This average is then digitally filtered over many frames.

For each color, the new filtered average is compared to a minimum acceptable level (to screen for too low a black level) and a maximum acceptable level. If the average is lower than the minimum acceptable level, the offset correction voltage for that color is increased by one offset LSB (offset LSBs do not match ADC LSBs; typically, one offset LSB is approximately 2mV). If it is above the maximum level, the level is decreased by 1 LSB (2mV). The upper threshold is automatically adjusted upwards whenever an upward shift in the black level from below the minimum results in a new black level above the maximum. This prevents black level oscillation from below the minimum to above the maximum. The lower threshold is increased with the maximum gain setting according to the formula described under Reg0x5F. This prevents clipping of the black level.

Whenever the gain or any of the readout timing registers is changed (shutter width, vertical blanking, number of rows or columns, or the shutter delay) or if the black level recalculation bit, reset bit or restart bit is set, the running digitally filtered average is reset to the first average of the dark pixels. The digital filtering over many frames is then restarted. Whenever the gain or the readout timing registers are changed, the upper threshold is restored to its default value.

After changes to the sensor configuration, large shifts in the black level calibration can result. To quickly adapt to this shift, a rapid sweep of the black level during the dark-row readout is performed on the first frame after certain changes to the sensor registers. Any changes to the registers listed above will cause this recalculation. The data from this sweep allows the sensor to choose an accurate new starting point for the running average. This procedure can be disabled as described under Reg0x5F.

Programmable Analog Offset Stage

The programmable analog offset stage corrects for analog offset that might be present in the analog signal. The user would need to program register 0x62 appropriately to enable the analog offset correction.

The lower eight bits (bit[7:0]) determines the absolute value of the analog offset to be corrected and bit[8] determines the sign of the correction. When bit[8] is "1", the sign of the correction is negative and vice versa. The analog value of the correction relative to the analog gain stage can be determined from the following formula:

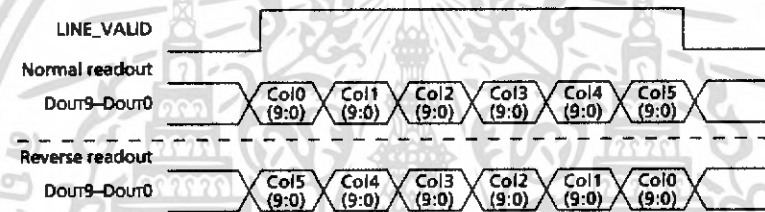
$$\text{Analog offset (bit[8] = 0)} = \text{bit[7:0]} \times 2\text{mV}$$

$$\text{Analog offset (bit[8] = 1)} = -(\text{bit[7:0]} \times 2\text{mV})$$

Column and Row Mirror Image

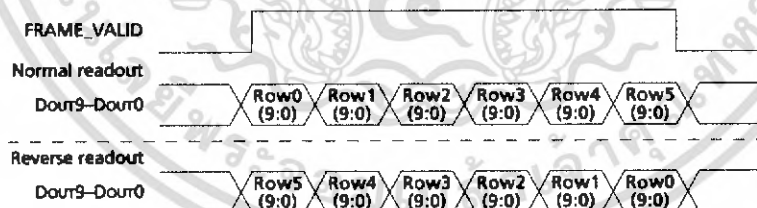
By setting bit 14 of Reg0x20, the readout order of the columns will be reversed, as shown in Figure 11.

Figure 11: Readout of Six Columns in Normal and Column Mirror Output Mode



By setting bits 15 of Reg0x20 the readout order of the rows will be reversed, as shown in Figure 12.

Figure 12: Readout of Six Rows in Normal and Row Mirror Output Mode



Column and Row Skip

By setting bit 3 of Reg0x20, only half of the columns set will be read out. An example is shown in Figure 13. Only columns with bit 1 equal to "0" will be read out (xxxxxxx0x). The row skip works in the same way and will only read out rows with bit 1 equal to "0." Row skip mode is enabled by setting bit 4 of Reg0x20. For both row and column skips, the number of rows or columns read out will be half of what is set in Reg0x03 or Reg0x04, respectively.

Two-wire Serial Bus Timing

The two-wire serial bus operation requires certain minimum master clock cycles between transitions. These are specified in the following diagrams in master clock cycles.

Figure 18: Serial Host Interface Start Condition Timing

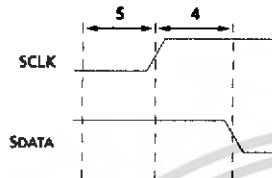
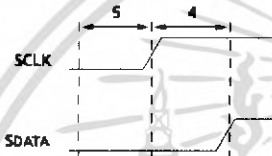
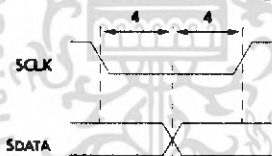


Figure 19: Serial Host Interface Stop Condition Timing



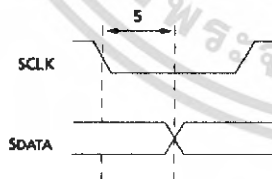
Note: All timing are in units of master clock cycle.

Figure 20: Serial Host Interface Data Timing for Write



Note: SDATA is driven by an off-chip transmitter.

Figure 21: Serial Host Interface Data Timing for Read



Note: SDATA is pulled LOW by the sensor, or allowed to be pulled HIGH by a pull-up resistor off-chip.

Figure 22: Acknowledge Signal Timing After an 8-Bit Write to the Sensor

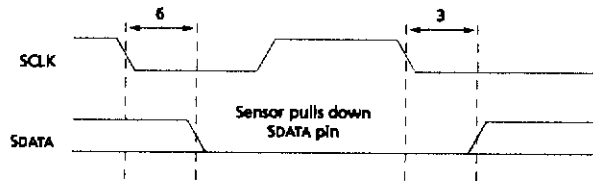
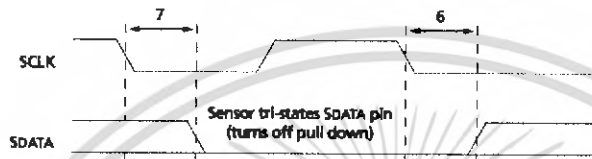
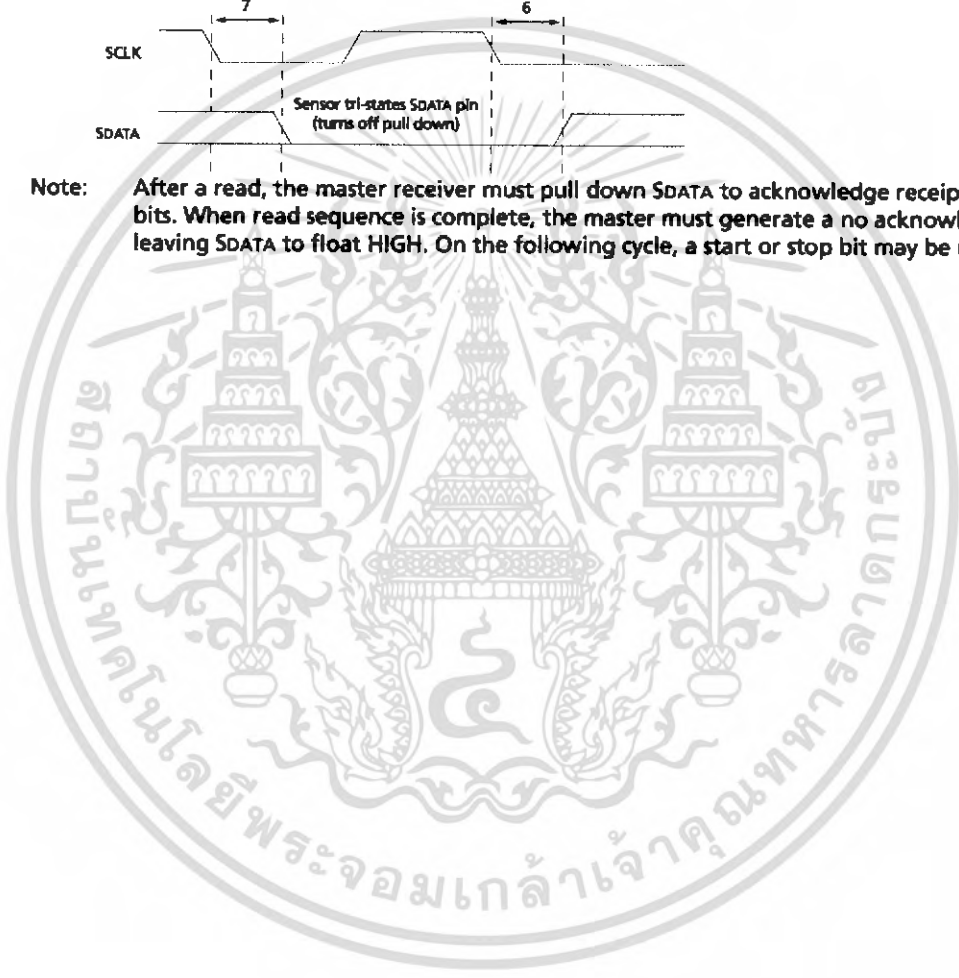


Figure 23: Acknowledge Signal Timing After an 8-Bit Read from the Sensor



Note: After a read, the master receiver must pull down SDATA to acknowledge receipt of data bits. When read sequence is complete, the master must generate a no acknowledge by leaving SDATA to float HIGH. On the following cycle, a start or stop bit may be used.



Quantum Efficiency

Figure 24: Quantum Efficiency—Monochrome

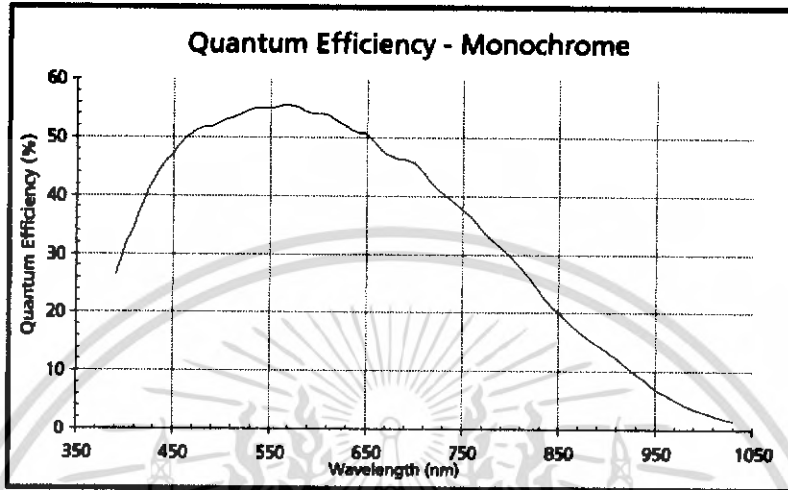
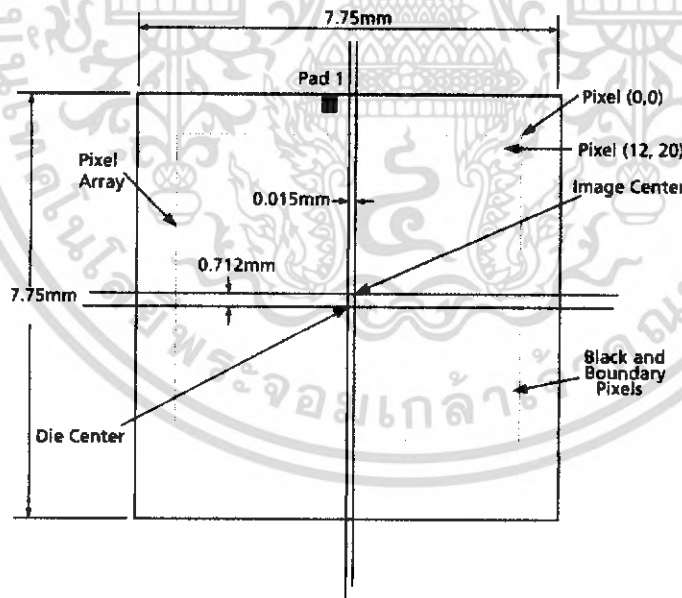


Image Center Offset and Orientation

Figure 25: Image Center Offset



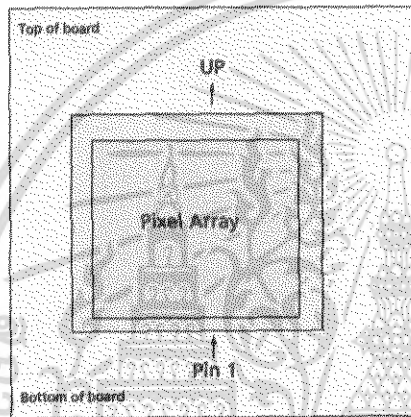
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 13: Optical Area Dimensions

Optical Area	Pixel	X-Dimension	Y-dimension
SXGA	Center of pixel (20, 12)	3,340.70 μ m	3,372.45 μ m
	Center of Pixel (1299, 1035)	-3,315.2 μ m	-1,952.35 μ m
Chip Size, mm	(including Seal Ring)	7.75mm	7.75mm

- Notes:
1. X and Y coordinates referenced to center of die.
 2. Die center = package center.
 3. Image center offset from package center (x = 0.015mm, y = 0.712mm).

Figure 26: Optical Orientation

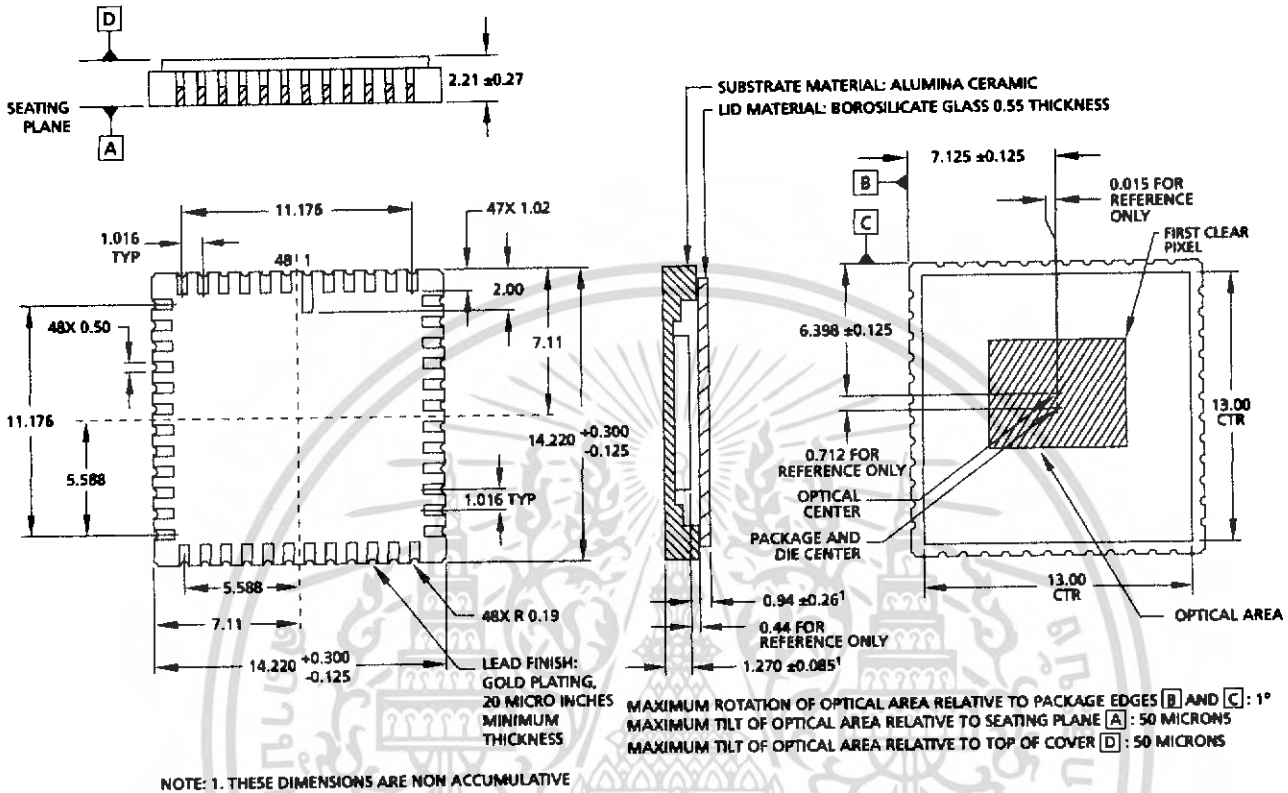


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MT9M001: 1/2-Inch Megapixel Digital Image Sensor Electrical Specifications

Figure 27: 48-pin CLCC Package Outline Drawing



8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
 prodmktg@micron.com www.micron.com Customer Comment Line: 800-932-4992
 Micron, the M logo, and the Micron logo are trademarks of Micron Technology, Inc.
 All other trademarks are the property of their respective owners.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Revision History

Rev F 5/06

- Updated Table 2, “Available Part Numbers,” on page 1

Rev E 4/06

- Updated “Data Output and Propagation Delays” on page 28
- Updated Table 11, “AC Electrical Characteristics,” on page 29

Rev D 2/06

- Updated Table 1, “Key Performance Parameters,” on page 1
- Update Table 2, “Available Part Numbers,” on page 1
- Updated Figure 4: “Pixel Pattern Detail (Top Right Corner),” on page 7

Rev C 6/05

- Remove color information
- Updated Table 1, “Key Performance Parameters,” on page 1
- Updated Table 6, “Register List and Default Values,” on page 14
- Updated Table 7, “Register Description,” on page 15
- Updated Figure 12, Readout of Six Rows in Normal and Row Mirror Output Mode, on page 22
- Deleted Figure 13, Readout of Eight Pixels in Normal and Column Skip Output Mode, on page 23
- Updated Table 11, “AC Electrical Characteristics,” on page 29
- Updated Figure 25, Image Center Offset, on page 32
- Updated Figure 27, 48-pin CLCC Package Outline Drawing, on page 34

Rev B 5/05

- Page 1, remove PRELIMINARY disclaimer
- Page 1, add Key Performance Parameters table, add APPLICATIONS
- Page 2, add Table of Contents
- Page 6, update Pin Description table
- Page 11, update Serial Bus Description
- Page 12, update Timing Diagram Showing a Read from Reg0x09; Returned Value 0x0284 figure
- Page 13, update Register List and Default Values table
- Page 14, update Register Description Table (add Test Data-Reg0x32[11:2], update Output Control-Reg0x07[6])
- Page 28, update AC and DC Electrical Characteristics table
- Page 29, add Figure 17, Data Output Timing Diagram, and Absolute Maximum Ratings, Table 11
- Page 30, update Propagation Delay for Frame_Valid and Line_Valid Signals (Data Output and Propagation Delays)
- Page 32, delete Quantum Efficiency figure (Color)
- Page 33, update Figure 27, 48-pin CLCC Package Outline Drawing

Rev A, Preliminary 11/03

- Initial Release of document

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้