สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวัดระยะทางอัลตราโซนิคควบคุมโดย PSoC Microcontroller

ULTRASONIC RANGE FINDER

โดย

นายอดิศักดิ์ อรรคไกรสีห์

เลขหมู่.....
เลขทะเบียน 72107
วัน,เดือน,ปี - 8 มิ.ย. 2550

b 11762182
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดระยะทางอัลตราโซนิคควบคุมโดย PSoC Microcontroller

ULTRASONIC RANGE FINDER

โดย

นายอดิศักดิ์ อรรคไกรสีห์ รหัสนักศึกษา 44010576

อาจารย์ที่ปรึกษา

ดร.กสิน วิเชียรชม

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2549

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดระยะทางอัลตราโซนิคควบคุมโดย PSoC Microcontroller

ผู้จัดทำ

1.อดิศักดิ์ อรรคไกรสีห์

.....อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดระยะทางอัลตราโซนิคควบคุมโดย PSoC Microcontroller

นาย อดิศักดิ์ อรรคไกรสีห์ รหัส 44010576
ดร.กสิน วิเชียรชม อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

ปริญญานิพนธ์นี้ อธิบายการออกแบบและการสร้างเครื่องวัดระยะทางโดยอาศัยคลื่นอัลตราโซนิค ระยะทางที่วัดสามารถคำนวณได้จากความเร็วในการเดินทางของคลื่นเสียงผ่านตัวกลางอากาศ และระยะเวลาที่เสียงเดินทางไปและกลับ ทำให้สามารถทราบถึงระยะทางที่เสียงเดินทางผ่านไปได้ โครงงานนี้ยังได้กล่าวถึงการนำ PSoC Microcontroller มาประยุกต์ใช้งาน ในการควบคุมการรับส่งคลื่นอัลตราโซนิค และ คำนวณระยะทาง เพื่อแสดงผลออกทาง LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ultrasonic Range Finder

Mr. Adisak Akkagraisi ID.44010576
Dr. Kasin Vichienchom Advisor
Educational Year 2006

Abstract

This report describes a design of ultrasonic range finder. By using the application of ultrasonic wave, the device can measure the distance from the product of velocity and the flight time of ultrasomic wave in the air. The flight time can be derived from the period of ultrasonic passing through the air and reflecting to its transmitter. In stead of using conventional microcontroller, this range finder was designed and implemented using PSoC, a programmable system-on-a-chip microcontroller. In this design PSoC generates signals to control the ultrasonic transducer, amplifies and filters the receiving ultrasonic wave and finally calculates and displays the distance on LCD.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำปริญญานิพนธ์นี้ ผู้จัดทำขอขอบพระคุณ ดร.กสิน วิเชียรชม ซึ่งท่านเป็นอาจารย์ที่ปรึกษา เป็นผู้ให้คำแนะนำแนวความคิด ช่วยแก้ไขข้อผิดพลาดและอุปสรรคต่างๆ ที่เกิดขึ้นในระหว่างการดำเนินการจัดทำปริญญานิพนธ์ จนสามารถสำเร็จตามเป้าหมายที่วางไว้ สำหรับความสำเร็จของปริญญานิพนธ์นี้ ผู้จัดทำขอยกความดีนี้ให้กับคุณบิดามารดา ครูบาอาจารย์และผู้มีพระคุณทุกๆท่านที่กล่าวมา

นายอดิศักดิ์ อรรคไกรสีห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ

บทที่ 2 ไมโครคอนโทรลเลอร์ PSoC

2.1 แนะนำไมโครคอนโทรลเลอร์ PSoC

2.2 แนวทางในการพัฒนาโปรแกรม

2.3 คุณสมบัติ PSoC MCU ตระกูล CY8C27x43

2.4 โครงสร้างสถาปัตยกรรมไมโครคอนโทรลเลอร์ PSoC

2.4.1 โครงสร้างหลักของ PSoC MCU

2.4.1.1 PSoC Core

2.4.1.2 Digital System

2.4.1.3 Analog System

2.4.1.4 System Resources

2.4.2 หน่วยความจำ (Memory)

2.4.2.1 ROM

2.4.2.2 Register

2.4.3 ขาสัญญาณอินพุต/เอาต์พุต (GPIO : General Purpose IO)

2.4.3.1 Digital IO

2.4.3.2 Global IO

2.4.3.3 Analog IO

2.4.4 ระบบสัญญาณนาฬิกา

2.4.4.1 สัญญาณนาฬิกาหลักภายใน Internal Main Oscillator

2.4.4.2 สัญญาณนาฬิกาแบบความเร็วต่ำภายใน (Internal Low Speed Oscillator (ILO))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.3 สัญญาณนาฬิกา 32 KHz จากคริสตอลภายนอก

2.4.4.4 Phase Locked Loop (PLL)

2.4.4.5 การทำงานในโหมด Sleep และ Watchdog

2.4.4.6 การรีเซตของระบบ (System Resets)

2.5 การอินเตอร์รัพท์ (Interrupt Operation)

2.5.1 โครงสร้างของการอินเทอร์รัพท์

2.5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัพท์

2.5.2.I รีจิสเตอร์ INT_CLRx

2.5.2.2 INT_MSKx Register

2.5.2.3 INT_VC Register

2.5.2.4 CPU_F Register

บทที่ 3 คลื่นอัลตราโซนิค

3.1 คลื่น(Wave)

3.1.1 คลื่นตามยาว (Longitudinal Wave)

3.1.2 คลื่นตามขวาง (Transverse Wave)

3.2 คลื่นเสียง (Sound Wave)

3.2.1 ลักษณะของคลื่นเสียง

3.2.2 คุณสมบัติที่สำคัญของคลื่นเสียง

3.3 คลื่นอัลตราโซนิค

3.4 ความแรงและกำลังของคลื่นอัลตราซนิค

3.5 อัลตราโซนิคทรานสดิวเซอร์

3.5.1 คุณสมบัติทั่วไปของทรานสดิวเซอร์

3.5.2 คุณสมบัติทางความถี่ของทรานสดิวเซอร์

3.5.3 เปียโซทรานสดิวเซอร์

3.5.4 เปียโซอิเล็คทริกเอฟเฟ็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ดิจิตอล และ อนาลอกบล็อกต่างๆที่นำมาใช้งาน

4.1 8-Bit Counter User Module

4.2 Programmable Gain Amplifier (PGA) User Module\

4.3 Digital Inverter User Module

บทที่ 5 หลักการทำงาน และการออกแบบวงจร

5.1 การทำงานโดยรวมของวงจร

5.1.1 วงจรภาคส่งคลื่นอัลตราโซนิค

5.1.2 วงจรภาครับสัญญาณอัลตราโซนิค

5.1.3 วงจรภาคนับระยะเวลาที่คลื่นอัลตราโซนิคใช้ในการเดินทาง

5.2 การออกแบบส่วนต่างๆ ของวงจร

5.2.1 การออกแบบส่วนที่ใช้ในการส่งสัญญาณคลื่นอัลตราโซนิค

5.2.2 การออกแบบส่วนที่ใช้ในการรับสัญญาณคลื่นอัลตราโซนิค

5.2.3 การออกแบบในส่วนของภาคนับระยะเวลาที่คลื่นอัลตราโซนิคใช้ในการเดินทาง

5.2.4 การออกแบบในส่วนของโปรแกรม

5.2.5 แสดงในส่วนของวงจรทั้งหมด

บทที่ 6 ผลการทดลอง

บทที่ 7 สรุปผลการทดลอง

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 แนวคิดในการจัดทำปริญญานิพนธ์

ในชีวิตประจำวัน ถ้าเราสังเกตสิ่งที่อยู่รอบๆตัวเราจะพบเห็นอุปกรณ์อิเลคทรอนิกส์บางอย่าง ซึ่งใช้งานอยู่ในช่วงความถี่อัลตราโซนิค และส่วนใหญ่จะใช้เป็นตัวตรวจจับหรือค้นหา เช่น อุปกรณ์ควบคุมระยะไกล , ตัวตรวจจับการเคลื่อนที่ , เครื่องค้นหาตำแหน่งปลา , เครื่องทำความสะอาดโดยคลื่นอัลตราโซนิค ฯลฯ เป็นต้น และความถี่ที่ใช้งานจะอยู่ในช่วง 19-100 KHz แต่ก็มีที่ใช้ความถี่สูงกว่านี้

ดังเช่นในโครงงานนี้ได้นำคลื่นอัลตราโซนิคมาประยุกต์ใช้ในการทำเครื่องวัดระยะทางด้วยคลื่นอัลตราโซนิค ซึ่งเราจะส่งคลื่นอัลตราโซนิคตรงออกไปยังวัตถุ จากนั้นก็อาศัยการสะท้อนกลับของคลื่นที่กลับมายังตัวรับทางทรานสดิวเซอร์ และคำนวณระยะทางจากคลื่น , ความถี่ที่ใช้ในการส่ง , ความเร็วของเสียงในอากาศ

1.2 ขอบเขตของโครงงาน

ลักษณะของเครื่องวัดระยะทางด้วยคลื่นอัลตราโซนิค และแสดงผลออกมาทางจอแอลซีดี วิธีที่ใช้ในการวัดระยะทาง คือ การปล่อยพัลส์จากวงจรกำเนิดสัญญาณ และรอรับการสะท้อนกลับของพัลสัญญาณ จึงหยุดวงจรนับ จากนั้นคำนวณระยะเวลาทั้งหมดและหารด้วยสอง เพราะเวลาเป็นเวลาทั้งไปและกลับ เมื่อได้ผลลัพธ์ของระยะทางแล้ว ก็จะนำไปแสดงผลออกทางจอแอลซีดีเพื่อแสดงผลให้ผู้ใช้ทราบถึงระยะทางที่ทำการวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ไมโครคอนโทรลเลอร์ PSoC

2.1 แนะนำไมโครคอนโทรลเลอร์ PSoC

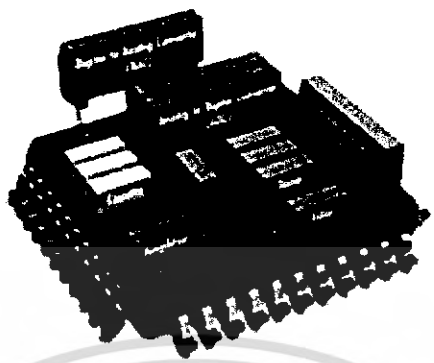
PSoC Microcontroller เป็นไมโครคอนโทลเลอร์ของค่าย Cypress Microsystems โดยเป็นไมโครคอนโทลเลอร์ 8 บิต ทั่วๆไป แต่คุณสมบัติที่แตกต่างไปจากไมโครคอนโทลเลอร์ตระกูลอื่นๆ ก็คือ PSoC MCU จะรวมเอาการออกแบบทั้งทางด้านดิจิตอล และ อนาลอก มาไว้ด้วยกันทำให้การออกแบบที่ต้องมีการใช้งานร่วมกันระหว่าง อุปกรณ์ทางดิจิตอล และ อนาลอก สามารถทำได้โดยง่าย และ สะดวกสบายยิ่งขึ้น อีกทั้งยังทำให้ขนาดของการออกแบบเล็กลงอีกด้วย โดยเฉพาะวงจรทางด้านอนาลอก ซึ่งมักจะมีขนาดค่อนข้างใหญ่ อีกทั้งการทำงานของ PSoC ยังมีความคล่องตัวสูงทั้งเรื่องของแหล่งกำเนิดสัญญาณนาฬิกาที่หลากหลาย นอกจากนี้ยังมีฟังก์ชัน In-System Serial Programming(ISSP) ที่สามารถทำการโปรแกรมซอร์สโค้ดที่ได้ออกแบบลงไปในหน่วยความจำโปรแกรม(Flash Memory) ภายในตัวชิพได้ ซึ่งจะช่วยให้การพัฒนาโปรแกรมให้กับไมโครคอนโทลเลอร์มีความสะดวกมากยิ่งขึ้น

จุดเด่นของ PSoC MCU เมื่อเทียบกับไมโครคอนโทลเลอร์รุ่นอื่นๆ

1. User Module : สามารถเลือกใช้ทรัพยากรของระบบได้ตามต้องการทั้ง อนาลอก และดิจิตอล จะไม่ถูกจำกัดด้วยโครงสร้างฮาร์ดแวร์เหมือนกับไมโครคอนโทลเลอร์อื่นๆ
2. API (Application Programming Interface) : สนับสนุนการพัฒนาโปรแกรมด้วยฟังก์ชัน API ช่วยผู้พัฒนาโปรแกรมสามารถเขียนออกแบบโปรแกรมได้โดยง่าย
3. ISRs(Interrupt Service Routines) : สนับสนุนและรองรับการทำงานแบบอินเตอร์รัพท์
4. Interconnect device interface : สามารถทำการเชื่อมต่อสัญญาณต่างๆได้อย่างอิสระ ไม่ถูกกำหนดตายตัวตามฮาร์ดแวร์

จากที่กล่าวมาเป็นเพียงคุณสมบัติคร่าวๆเท่านั้น ซึ่งคุณสมบัติอื่นๆที่เหลือจะคล้ายๆกับไมโครคอนโทลเลอร์อื่นๆ เช่น Sleep , watchdog , Power on Reset(POR) , SPI , UART , และ I2C เป็นต้น เรียกได้ว่าไม่ได้ด้วยกว่าไมโครคอนโทลเลอร์รุ่นอื่นๆเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ภาพจำลองการวางโมดูลต่างๆลงในไอซี PSoC

ไมโครคอนโทลเลอร์ PSoC จะสนับสนุนระบบการทำงานทั้งด้าน ดิจิตอลและอนาลอก โดยในระบบของดิจิตอล (Digital System) และ อนาลอก (Analog System) จะถูกออกแบบให้เป็นบล็อกโมดูล ซึ่งจะเรียกว่าบล็อกดิจิตอล(Digital Block) และ บล็อกอนาลอก(Analog Block) โดยบล็อกเหล่านี้จะรองรับการนำเอาโมดูลต่างๆมาใช้งาน เปรียบเสมือนเป็นพื้นที่ว่างๆสำหรับต่อจิ๊กซอ โดยชิ้นส่วนของจิ๊กซอก็คือโมดูลต่างๆเช่น ADC , DAC , I2C , PWM , UART และ SPI เป็นต้น ดังเช่นในรูปที่ 2.1

ผู้ใช้งาน(User) จะเป็นผู้กำหนดเองว่าจะเอาโมดูลใดมาวางใช้งานได้ นอกจากนี้ผู้ใช้งานยังสามารถกำหนดการเชื่อมต่อสัญญาณต่างๆ(Programmable Interconnect) ภายในได้เองอีกด้วย เสมือนกับว่าผู้ใช้งานสามารถทำการออกแบบเอง ได้ตั้งแต่ ฮาร์ดแวร์ ไปจนถึง ซอฟแวร์ ซึ่งถือได้ว่าเป็นความสามารถ หนึ่งที่เหนือกว่าไมโครคอนโทลเลอร์ตระกูลอื่นๆ ที่ทรัพยากรทุกอย่างถูกกำหนดไว้ตายตัว ไม่สามารถเปลี่ยนแปลงได้

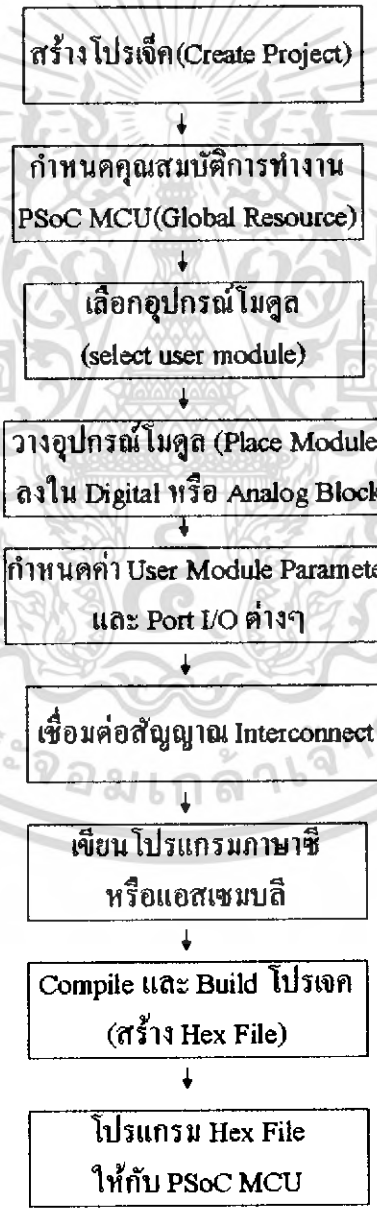
2.2 แนวทางในการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรม ภาษาโปรแกรมที่ใช้ในการออกแบบพัฒนาการทำงานของ PSoC MCU ปัจจุบันจะมีอยู่ 2 ภาษาด้วยกัน คือ ภาษาแอสเซมบลี และ ภาษาซี โดยภาษาที่เป็นพื้นฐานในการออกแบบการเขียนโปรแกรมให้กับ PSoC MCU คือ ภาษาแอสเซมบลี ซึ่งเป็นภาษาซึ่งทางบริษัทได้ผลิตได้สร้างขึ้นมาเพื่อสนับสนุนการเขียนโปรแกรมคำสั่งให้กับ PSoC MCU เป็นซอร์ฟแวร์ฟรี ไม่เสียค่าใช้จ่ายใดๆ ส่วนภาษาซี เป็นภาษาที่มีเพิ่มเติมขึ้นมาเพื่อสนับสนุนให้มีการพัฒนาโดยใช้ภาษาระดับสูง โดยเป็นการร่วมมือกันกับบริษัทผลิตชิพ PSoC และบริษัทผู้ผลิตซอร์ฟแวร์(IMAGECRAFT) โดยทั้งตัวแปลภาษาแอสเซมบลี และ ภาษาซี จะถูกรวมอยู่ในซอร์ฟแวร์โปรแกรมชื่อ PSoC Designer สามารถดาวน์โหลดได้ฟรีที่เวปไซต์ของ Cypress

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsystems (www.cypressmicro.com) โดยการใช้งานภาษาซี จะต้องมีค่าใช้จ่ายในการนำ License number มาเปิดใช้งาน ส่วนภาษาแอสเซมบลีนั้นสามารถใช้งานได้ฟรีทันที

เนื่องจากการพัฒนาโปรแกรมของ PSoC MCU ส่วนใหญ่จะทำโดยการเรียกใช้งานฟังก์ชัน API และ การกำหนดคุณสมบัติต่างๆ เช่น ความถี่สัญญาณนาฬิกา , Sleep , Watchdog , Supply Voltage และ อื่นๆยังทำได้จากหน้าต่าง Device Editor ของซอร์ฟแวร์ PSoC Designer ทำให้เราไม่จำเป็นต้องทราบรายละเอียดของรีจิสเตอร์ต่างๆมากนัก โดยสามารถสรุปขั้นตอนการพัฒนาโปรแกรมให้กับ PSoC MCU คร่าวๆได้ตามลำดับขั้นในรูปที่ 2.2



รูปที่ 2.2 ลำดับขั้นตอนการพัฒนาโปรแกรมให้กับ PSoC MCU ด้วย PSoC Designer แบบคร่าวๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 คุณสมบัติ PSoC MCU ตระกูล CY8C27x43

PSoC MCU จะมีอยู่หลายตระกูลด้วยกัน โดยตระกูลที่เราจะนำมาใช้ในโครงงานนี้ก็คือตระกูล Cy8327x43 ซึ่งในตระกูลนี้สามารถมีพอร์ต I/O ได้สูงสุดถึง 5 พอร์ต โดยจะมีคุณสมบัติคร่าวๆ ดังต่อไปนี้

คุณสมบัติของ CY8C27443

- สถาปัตยกรรมแบบ Harvard Architecture Processor
- ความเร็วของ M8C Processor สูงสุด 24 MHz
- 8x8 Multiply , 32-Bit Accumulate
- Low Power at High Speed
- ทำงานในช่วงแรงดัน 3.0 ถึง 5.25 V
- สามารถทำงานที่แรงดันต่ำสุดที่ 1.0 V โดยใช้วงจร Switch Mode Pump (SMP) ภายในชิพ
- 12 Analog PSoC Blocks รองรับการใช้งานทางด้านอนาลอก เช่น
  - -ADCs ความละเอียดสูงสุดถึง 14 บิต
  - -DACs ความละเอียดสูงสุดถึง 9 บิต
  - -วงจรเกณฑ์การขยาย(Programmable Gain Amplifiers)
  - -วงจรฟิลเตอร์ และ วงจรคอมพาราเตอร์ (Programmable Filters and Comparators)
- 8 Digital PSoC Blocks รองรับการใช้งานทางดิจิตอล
  - -Timers , Counter และ PWMs ขนาด 8 ถึง 32 บิต
  - -CRC และ PRs โมดูล
  - -UARTs แบบ Full-Duplex สูงสุด 2 ช่อง
  - -SPI โมดูล เป็นได้ทั้งแบบ Master และ Slave
  - -ดิจิตอลบล็อกต่างๆสามารถเชื่อมต่อไปยังขาสัญญาณ GPIO ได้ทุกขาสัญญาณ
- สามารถกำหนดขนาดความถี่ของสัญญาณนาฬิกาภายในได้หลายระดับ
- สัญญาณนาฬิกาภายในขนาด 24/48 MHz ค่าความคลาดเคลื่อน +/- 2.5 %
- สามารถเลือกแหล่งกำเนิดสัญญาณ 24/48 MHz จากออสซิลเลเตอร์ 32 KHz ภายใน หรือภายนอกได้
- สามารถรับสัญญาณออสซิลเลเตอร์จากภายนอกได้สูงสุด 24MHz
- มีแหล่งกำเนิดสัญญาณนาฬิกาภายในให้กับ Watchdog และ Sleep

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

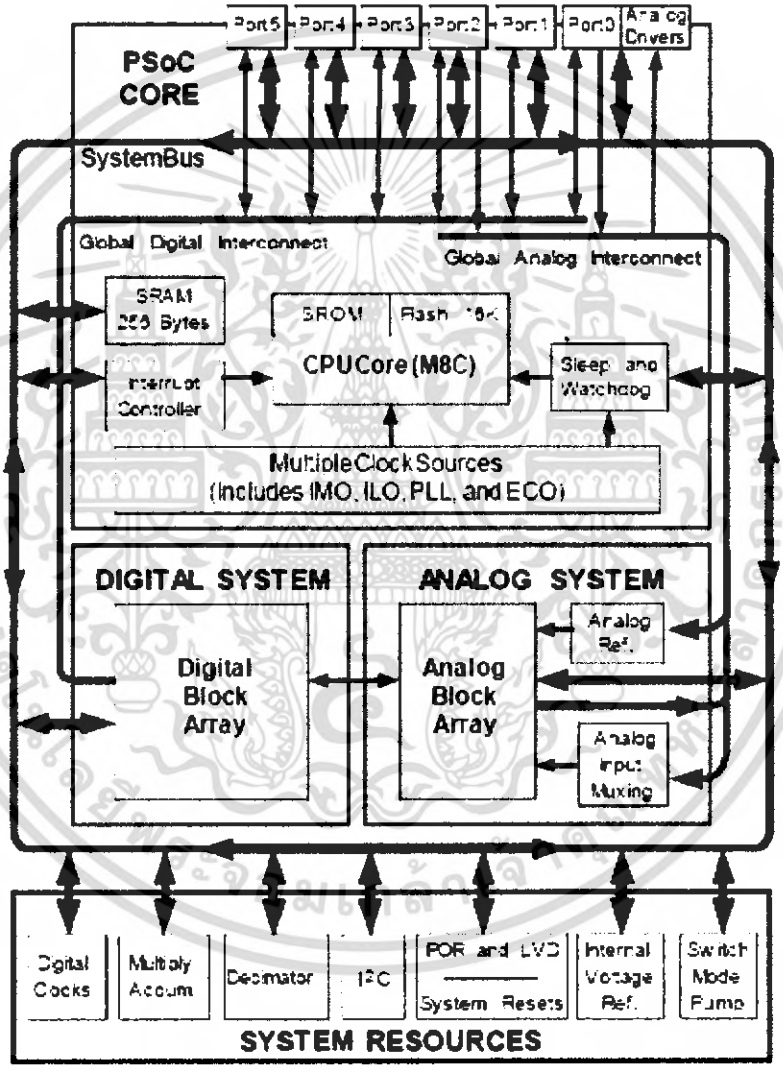
- หน่วยความจำโปรแกรมแบบ Flash ขนาด 16 K Byte สามารถ Erase/Write ได้ถึง 50,000 ครั้ง
- หน่วยความจำข้อมูล SRAM ขนาด 256 Byte
- ฟังก์ชันโปรแกรมภายใน ISSP (In-System Serial Programming)
- สามารถเปลี่ยนแปลงข้อมูล Flash Memory เฉพาะบางส่วนได้
- สามารถตั้งค่าระบบป้องกันข้อมูลได้(Flash Security)
- หน่วยความจำ EEPROM (ใช้จาก Flash Memory)
- ขาสัญญาณ GPIO จำนวน 24 ขาสัญญาณ
- สามารถกำหนดคุณสมบัติของขาสัญญาณต่างๆได้ (GPIO Pin Configuration)
- GPIO สามารถจ่ายกระแสได้สูงสุดถึง 25 mA
- สามารถกำหนด Device Mode ของสัญญาณ GPIO ได้ เช่น Pull Up , Pull Down , High Z , Strong หรือ Open Drain
- ขาสัญญาณอินพุทอนาลอกสูงสุด 12 ขาสัญญาณ (จากขาสัญญาณของ GPIO)
- ขาสัญญาณอนาลอกเอาพุตสูงสุด 4ขาสัญญาณ (จากขาสัญญาณ GPIO) จ่ายกระแสได้ถึง 40 mA
- สามารถกำหนดการอินเตอร์รัพท์ได้ทุกขาสัญญาณ GPIO
- I2C โหมด Slave , Master และ Multi-Master ความเร็วสูงสุด 400 KHz
- Watchdog และ sleep timer
- สามารถตั้งค่าการตรวจจับระดับแรงดันต่ำได้(Low Voltage Detection)
- Integrated Supervisory Circuit
- On-Chip Precision Voltage Reference

จะเห็นได้ว่า PSoC MCU เป็นไมโครคอนโทรลเลอร์ที่มีคุณสมบัติที่ถือว่าครบถ้วนมาก ไม่ได้ด้วยไปกว่า ไมโครคอนโทรลเลอร์ตระกูลอื่นๆเลย บางทีอาจถูกมองว่าการที่มีคุณสมบัติหลายๆ อย่างเช่นนี้จะทำให้การออกแบบเป็นไปได้ด้วยความยุ่งยาก แต่สำหรับ PSoC MCU แล้ว กลับทำได้ไม่ยุ่งยากนักเนื่องจากใช้ซอร์ฟแวร์ PSoC Designer ในการออกแบบ ซึ่งซอร์ฟแวร์ดังกล่าวนี้จะมีการออกแบบในลักษณะของกราฟิก หรือแบบวิสชวล (Visual) เช่นการเลือกใช้โมดูลต่างๆหรือการกำหนดค่า Configuration ต่างๆ ส่วนในการเขียนโปรแกรมก็จะเป็นลักษณะการใช้งานฟังก์ชัน (API : Application Programming Interface) ที่ PSoC Designer ได้จัดเตรียมไว้ให้แล้ว ทำให้ลดความยุ่งยากไปได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างสถาปัตยกรรมไมโครคอนโทรลเลอร์ PSoC

ในบทนี้จะกล่าวถึงโครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PSoC โดยจะอธิบายโครงสร้างภายใน และ ทรัพยากรต่างๆของระบบ เช่น โครงสร้างภายใน , หน่วยความจำ , ขาสัญญาณ GPIO , ระบบสัญญาณนาฬิกา , Sleep , Watchdog การรีเซตของระบบ(System Reset) เป็นต้น



รูปที่2.3 แสดงบล็อกไดอะแกรมของ PSoC

2.4.1 โครงสร้างหลักของ PSoC MCU

ไมโครคอนโทรลเลอร์ PSoC โครงสร้างภายในจะประกอบไปกอบไปด้วยส่วนต่างๆมากมาย โดยแบ่งออกเป็น 4 ส่วนหลักๆคือ PSoC CORE ระบบของดิจิตอล (Digital System) และระบบของอนาลอก (Analog System) และ System Resource โดยแต่ละส่วนจะเชื่อมต่อกันด้วยระบบบัส ดังรูปที่ 2.3

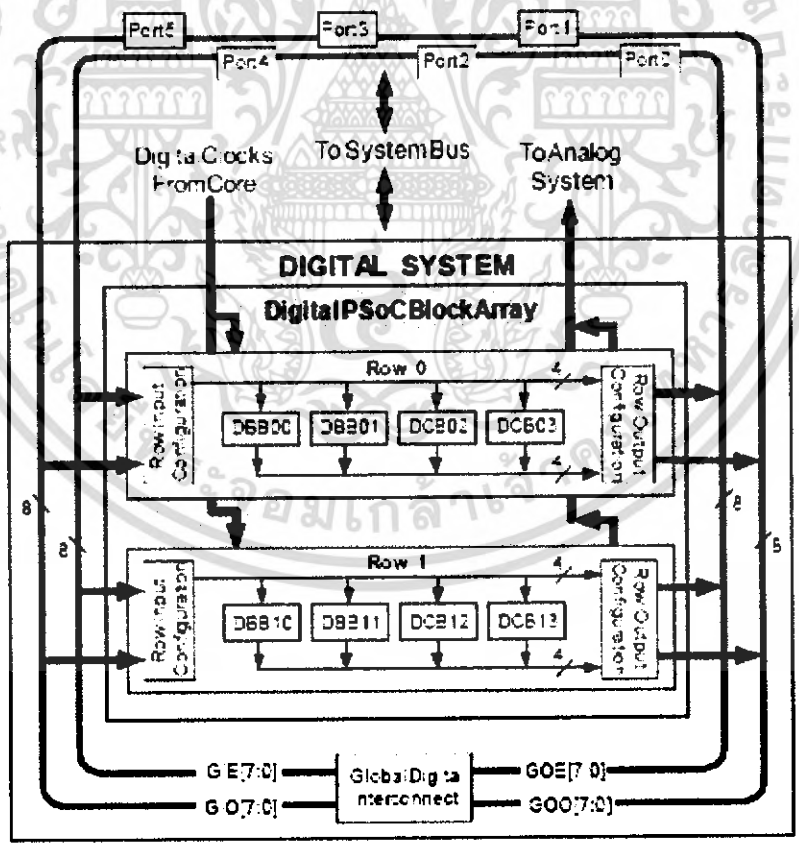
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 PSoC Core

เป็นส่วนหลักของการประมวลผล ซึ่งจะดูแลส่วนต่างๆ เช่น การการประมวลผลคำสั่ง , การจัดเก็บข้อมูลในหน่วยความจำ SRAM , ควบคุมการอินเตอร์รัพท์ , Sleep , Watchdog timers และ การเลือกแหล่งสัญญาณนาฬิกา (Clock Source) เป็นต้น ซึ่งเราจะเรียกส่วนที่จัดการส่วนต่างๆ เหล่านี้ว่า M8C ซึ่งเป็นสถาปัตยกรรมไมโครโปรเซสเซอร์ 8 บิต แบบ Harvard นอกจากนี้ภายใน PSoC Core ยังมีหน่วยความจำ SROM และ Flash Memory สำหรับใช้เก็บโปรแกรมคำสั่งอีกด้วย

2.4.1.2 Digital System

เป็นระบบของดิจิตอลบล็อกที่วางอยู่ในรูปแบบของบล็อกอาเรย์ (Array Block) ในส่วนของ Digital System การวางของบล็อกอาเรย์จะวางแถวละ 4 บล็อกสำหรับ CY8C27443 จะมีอาเรย์บล็อกจำนวน 2 แถว รวมแล้วมีทั้งหมด 8 ดิจิตอลบล็อก ดังรูปที่ 2.4 ส่วน PSoC MCU เบอร์อื่นๆจะมีขนาดแตกต่างกันออกไป โดยบล็อกเหล่านี้ก็จะรองรับการใช้งาน ในส่วนของโมดูลดิจิตอลต่างๆ



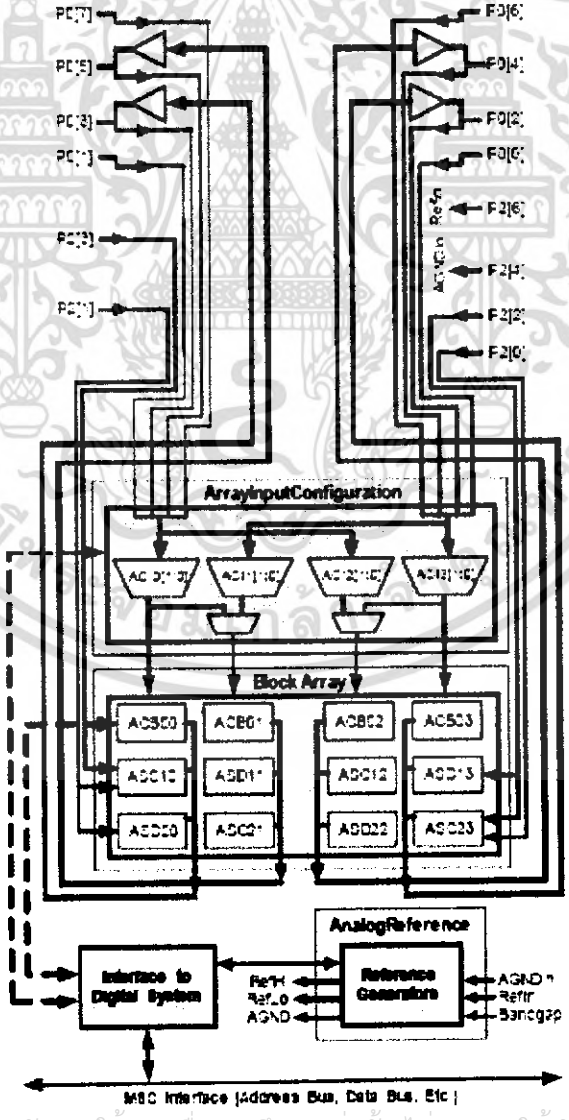
รูปที่ 2.4 โครงสร้างบล็อกของ Digital System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.3 Analog System

ระบบอนาลอก (Analog System) จะประกอบไปด้วยส่วนต่างๆคือ อนาลอกบล็อก (Analog Block) , แรงดันอ้างอิง (Analog Ref) ,และ วงจรเลือกสัญญาณอินพุตอนาลอก (Analog Input Muxing) ดังรูปที่ 2.5

โดยอนาลอกบล็อกจะมีอยู่สองประเภทด้วยกันคือ CT (Continuous Time) และ SC (Switched Capacitor) ซึ่งอนาลอกบล็อกจะถูกจัดเรียงเป็น 4 คอร์ลัม (Column) ในหนึ่งคอร์ลัมก็จะประกอบด้วย CT หนึ่งบล็อก และ SC อีก สองบล็อก ซึ่งบล็อกเหล่านี้จะมีไว้สำหรับรองรับการทำงานของโมดูลอนาลอกโดยการจัดวางโมดูลลงในบล็อกต่างๆนั้นก็ขึ้นอยู่กับความเหมาะสมของแต่ละโมดูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 โครงสร้างบล็อกของ Analog System

2.4.1.4 System Resources

System Resource ก็คือ ฮาร์ดแวร์ต่างๆของระบบ เช่น Digital Clock , Multiply accumulate(MAC) , Decimator , I2C , System resets , Internal voltage reference , Switch mode pump(SMP) เป็นต้น ดังรูปที่ 2.6

รูปที่ 2.6 แสดงส่วนต่างๆของ System Resources

2.4.2 หน่วยความจำ (Memory)

หน่วยความจำเป็นองค์ประกอบหนึ่งที่มีความจำเป็นต่อการทำงานของไมโครโปรเซสเซอร์ หรือ ไมโครคอนโทรลเลอร์ โดยจะมีการแบ่งหน่วยความจำออกเป็นประเภทต่างๆ ตามคุณสมบัติและการใช้งาน ซึ่งในส่วนของสถาปัตยกรรม M8C ก็จะแบ่งหน่วยความจำออกเป็น 3 ส่วน ROM , RAM และ Register

2.4.2.1 ROM จะมีการเข้าถึงโดยผ่านทาง Address Bus และ Data Bus โดยจากรูปภายใน ROM จะประกอบไปด้วย SROM (Supervisory ROM) และ หน่วยความจำชนิด Flash จะแยกออกเป็นบล็อกๆละ 64 ไบต์ โดยผู้ใช้งานไม่จำเป็นต้องมาดูแลในเรื่องของเขตแดนรอยต่อของหน่วยความจำในแต่ละบล็อก เพราะ M8C จะมีการจัดการให้เราโดยอัตโนมัติ รีจิสเตอร์ที่ทำหน้าที่เก็บตำแหน่งของการประมวลผลก็คือ รีจิสเตอร์ PC จะมีการเพิ่มค่าขึ้นทุกๆ instruction ไซเคิลคำสั่ง โดยจะเป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งแบ่งออกเป็นสองส่วน คือ 8 บิตบน (PCH) และ 8 บิตล่าง (PCL) ซึ่งใน 8 บิตบนจะมีการเพิ่มค่าหนึ่งค่าเมื่อ PC 8 บิตล่างนับถึงค่าที่ 256 (ยกเว้นคำสั่งที่เป็นการกระโดด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Description
0x0000	Reset Vector
0x0004	Supply Monitor Interrupt Vector
0x0008	DBA 00 PSoC Block Interrupt Vector
0x000C	DBA 01 PSoC Block Interrupt Vector
0x0010	DBA 02 PSoC Block Interrupt Vector
0x0014	DBA 03 PSoC Block Interrupt Vector
0x0018	DCA 04 PSoC Block Interrupt Vector
0x001C	DCA 05 PSoC Block Interrupt Vector
0x0020	DCA 06 PSoC Block Interrupt Vector
0x0024	DCA 07 PSoC Block Interrupt Vector
0x0028	Analog Column 0 Interrupt Vector
0x002C	Analog Column 1 Interrupt Vector
0x0030	Analog Column 2 Interrupt Vector
0x0034	Analog Column 3 Interrupt Vector
0x0038	GPIO Interrupt Vector
0x003C	Sleep Timer Interrupt Vector
0x0040	On-Chip User Program Memory Start Here
…	***
…	***
…	***
0x3FFF	16K Flash Maximum Depending on Version

ตารางที่ 2.1 Flash Program Memory Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.2 Register รีจิสเตอร์เป็นหน่วยความจำหนึ่งที่มีความสำคัญมากต่อการทำงานของระบบซึ่ง การเขียนโปรแกรม เราจำเป็นจะต้องอาศัยรีจิสเตอร์เหล่านี้ไว้เก็บ หรือพักข้อมูล รวมถึงการทำงาน บางอย่างที่ต้องอาศัยรีจิสเตอร์ โดย M8C จะจัดพื้นที่รีจิสเตอร์เป็นแบบแบงก์(Bank) ซึ่ง M8C จะมี รีจิสเตอร์อยู่ 2 แบงก์ด้วยกัน คือ พื้นที่แบงก์ของ User Space และ Configuration Space มีขนาด แบงก์ละ 256 ไบต์ เนื่องจากรีจิสเตอร์มีการจัดหน่วยความจำเป็นแบงก์แยกกันอยู่ดังนั้นในการ เข้าถึงข้อมูลแต่ละส่วนจึงต้องมีการเลือกแบงก์ข้อมูลที่เราจะเข้าถึงซึ่งสามารถทำการเลือกแบงก์ได้ โดยการกำหนดค่าให้กับบิต XIO ด้วยการเซต หรือ เคลียร์(เซตเป็น 1 คือ เลือกพื้นที่ของ Configuration Space และหากเคลียร์เป็น 0 คือการเลือก User Space)

สถาปัตยกรรม M8C จะมีรีจิสเตอร์พิเศษภายใน 5 ตัวที่ใช้เกี่ยวกับการประมวลผลโปรแกรม ดังต่อไปนี้

-Accumulator(A) : เป็นรีจิสเตอร์หลักที่ทำงานในส่วนของการกระทำทางคณิตศาสตร์ และลอจิก

-Index(X) : เป็นรีจิสเตอร์ชี้ทางอ้อมสำหรับการเข้าถึงข้อมูลแบบทางอ้อม

-Program Counter(PC) : เป็นรีจิสเตอร์ที่เก็บคำสั่งที่ CPU จะทำการตีความ หรือ ประมวลผล

-Stack Pointer(SP) : เป็นรีจิสเตอร์ที่เก็บตำแหน่งของหน่วยความจำ stack

-Flag(F) : เป็นรีจิสเตอร์สถานะ

รีจิสเตอร์ทั้งหมดจะเป็น 8 บิต ยกเว้นรีจิสเตอร์ PC ซึ่งมีขนาดเป็น 16 บิต เมื่อเกิดการรีเซตของระบบ รีจิสเตอร์ A,X,PC,SP จะมีสถานะเป็น 0x000 ส่วนรีจิสเตอร์ F จะมีค่าเป็น 0x02 ซึ่งจะเห็นว่า Zero Flag มีการเซตเป็น 1

- **รีจิสเตอร์ A (Accumulator)**
  เป็นรีจิสเตอร์เก็บข้อมูลทั่วไปขนาด 8 บิต นอกจากนี้ยังเป็นรีจิสเตอร์หลัก ที่ทำงานในส่วนของคำสั่งการกระทำทางคณิตศาสตร์ลอจิกอีกด้วย
- **รีจิสเตอร์ X (Index Register)**
  เป็นรีจิสเตอร์เก็บข้อมูลขนาด 8 บิต และยังทำหน้าที่เป็นรีจิสเตอร์ตัวชี้ตำแหน่ง (Index) ในกรณีคำสั่งที่มีการเข้าถึงแอดเดรสแบบทางอ้อมโดยใช้ตัวชี้ (Index Addressing Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **รีจิสเตอร์ PC (Program Counter Register)**
  เป็นรีจิสเตอร์ขนาด 16 บิต ที่ทำหน้าที่เก็บตำแหน่งหน่วยความจำโปรแกรมที่ CPU กำลังประมวลผลอยู่โดยจะประกอบด้วย 8 บิต ไบต์สูง (PCH) และ 8 บิต ไบต์ต่ำ (PCL) ซึ่งค่าของรีจิสเตอร์ PC จะมีการเปลี่ยนแปลงค่าเองโดยอัตโนมัติ เมื่อ CPU ทำการแอกซิคิวซ์คำสั่งต่างๆ หรือการกระทำคำสั่ง การกระโดด และการเกิด อินเตอร์รัพท์ต่างๆ เป็นต้น
- **รีจิสเตอร์ SP (Stack Pointer Register)**
  เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของหน่วยความจำสแต็ก โดยรีจิสเตอร์ stack จะมีการเพิ่มค่าขึ้นโดยอัตโนมัติเมื่อถูกใช้งาน เช่น กรใช้คำสั่ง PUSH หรือ CALL เป็นต้น โดยจะเป็นการเพิ่มค่าการชี้ตำแหน่งในหน่วยความจำแรมซึ่งถ้าหากมีการชี้ตำแหน่งจนถึงค่าตำแหน่ง 0xFF ค่าของสแต็ก จะกลับไปที่ 0x00 อีกครั้ง ก็คือ จะเป็น stack แบบวน ดังนั้นผู้พัฒนาโปรแกรมจะต้องดูแลไม่ให้เกิดการทับกัน (Overlap) ของแสต็ก กับตัวแปรที่ได้ Define ไว้ใน RAM
- **รีจิสเตอร์ F (Flags)**
  เป็นรีจิสเตอร์สถานะ มีทั้งหมด 8 บิต ซึ่งในแต่ละบิตมีการใช้งานดังนี้
  -PgMode และ XIO ใช้สำหรับการเลือกแบงก์ของรีจิสเตอร์
  -C เป็น Carry Flag จะมีผลต่อการกระทำคำสั่งทางคณิตศาสตร์ ลอจิก การเลื่อนหรือหมุนบิต
  -Z เป็น Zero Flag จะมีผลต่อการกระทำคำสั่งทางคณิตศาสตร์ ลอจิก
  -GIE เป็นบิตควบคุมการเกิดอินเตอร์รัพท์ โดยเป็นบิตที่มีนัยสำคัญสูงสุด หาเซตบิตนี้ก็จะเป็นการอนุญาตให้การอินเตอร์รัพท์สามารถเกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ F(Flags)

Bit#	7	6	5	4	3	2	1	0
Access POR				RL:0		RL:0	RL:0	RL:0
Bit Name	Reserved			XIO	Reserved	Carry	Zero	GIE

Bit Name **Description**

Bit7 : Reserved เป็นบิตที่สงวนไว้ไม่ใช้งาน

Bit6 : Reserved เป็นบิตที่สงวนไว้ไม่ใช้งาน

Bit5 : Reserved เป็นบิตที่สงวนไว้ไม่ใช้งาน

Bit4 : XIO เป็นบิตเลือกการเข้าถึงแบงก์รีจิสเตอร์ มีด้วยกันสองแบงก์คือ Bank 0

และ Bank 1 0 = Bank0 , 1 = Bank1

Bit3 : Reserved เป็นบิตที่สงวนไว้ไม่ใช้งาน

Bit2 : Carry บิตนี้จะมีการเปลี่ยนแปลงโดยอัตโนมัติ จากผลลัพธ์การกระทำทาง

คณิตศาสตร์ลอจิก

0 = No Carry

-เมื่อผลลัพธ์จากการกระทำคำสั่งลบไม่มีการยืม

-เมื่อผลลัพธ์จากการกระทำคำสั่งบวกไม่มีการยืม

1 = Carry

-เมื่อผลลัพธ์จากการกระทำคำสั่งลบมีการยืม

-เมื่อผลลัพธ์จากการกระทำคำสั่งบวกมีการยืม

Bit1 : Zero บิตนี้จะมีการเปลี่ยนแปลงโดยอัตโนมัติ จากผลลัพธ์การกระทำทาง

คณิตศาสตร์ลอจิก

0 = Not Equal to Zero : เมื่อผลลัพธ์จากการกระทำทาง

คณิตศาสตร์ไม่เท่ากับ 0

1 = Equal to Zero : เมื่อผลลัพธ์จากการกระทำทาง

คณิตศาสตร์เท่ากับ 0

Bit0 : Global IE บิตนัยความสำคัญสูงสุดในการควบคุมการเกิดอินเตอร์รัพท์

0 = Disable : ปิดการอินเตอร์รัพท์ทั้งหมด

1 = Enable : เปิดการอินเตอร์รัพท์

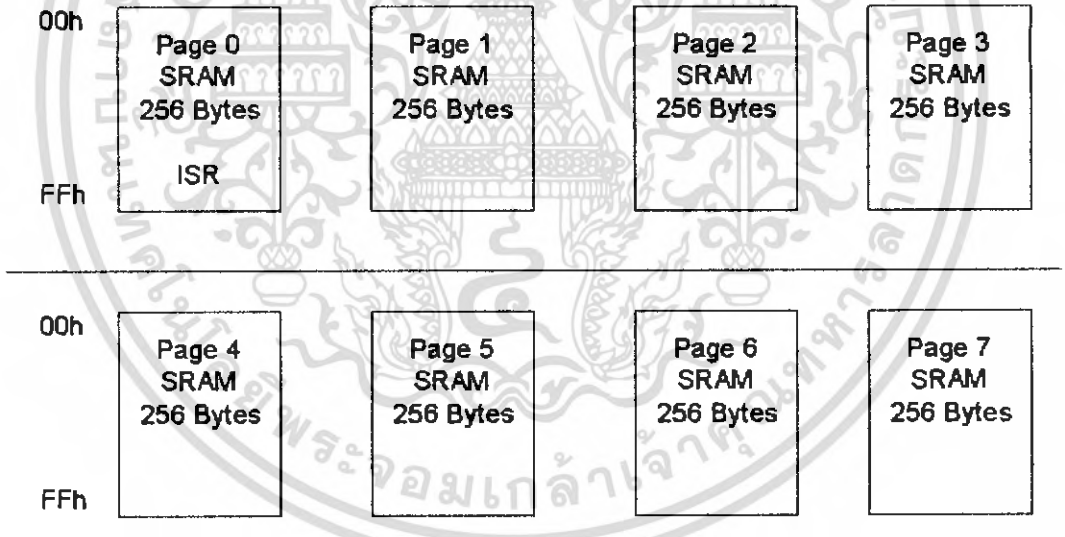
72107

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.3 RAM (Random Access Memory) หน่วยความจำแรมเป็นหน่วยความจำที่ใช้เก็บข้อมูลชั่วคราว ซึ่งข้อมูลไม่สามารถคงอยู่ได้หากไม่มีไฟเลี้ยงป้อนให้ โดยหน่วยความจำแรมของ PSoC MCU จะมีโครงสร้างเป็นแบบ Page ซึ่งจะมีขนาด Page ละ 256 Byte ซึ่งสำหรับ PSoC ในตระกูล CY8C27x43 จะมี RAM ขนาด 1 Page คือ 256 byte ส่วน PSoC ในตระกูลอื่นๆก็จะมีขนาดต่างๆ ตามตารางที่ 2.2

PSoC Device	ขนาดของ RAM	
CY8C29x66	2 KB	8 Pages
CY8C27x66	2 KB	8 Pages
CY8C27x43	256 Bytes	1 Pages
CY8C24x23	256 Bytes	1 Pages
CY8C22x13	256 Bytes	1 Pages

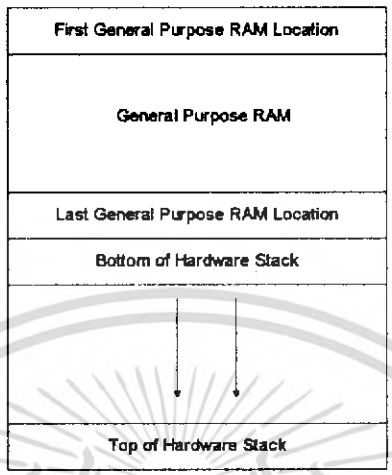
ตารางที่ 2.2 ขนาดหน่วยความจำ RAM ของ PSoC



รูปที่ 2.7 แสดงลักษณะของหน่วยความจำ RAM

หน่วยความจำส่วนหนึ่งภายใน RAM จะถูกใช้งานเป็นหน่วยความจำ สแต็ก (stack) โดยจากจากรูปที่ 2.8 จะเห็นว่าตำแหน่งแรกของหน่วยความจำสแต็ก (Bottom of Hardware Stack) จะเริ่มต้นที่ตำแหน่งต่อจากตำแหน่งสุดท้ายของพื้นที่ General Purpose RAM และจะไปสิ้นสุดที่ตำแหน่งสุดท้ายคือ 0xFF แต่ถ้ามีการเพิ่มตำแหน่งของสแต็กอีก จาก 0xFF ตำแหน่งต่อไปจะเป็น 0x00 ซึ่งจะเรียกเหตุการณ์นี้ว่า Stack Overflow โดยไม่ควรจะเกิดขึ้นเนื่องจาก อาจทำให้ตำแหน่งของสแต็กไปทับกับตำแหน่งของข้อมูลที่เรากำลังใช้งานอยู่ ซึ่งอาจจะทำให้เกิดการผิดพลาดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงโครงสร้างหน่วยความจำ RAM

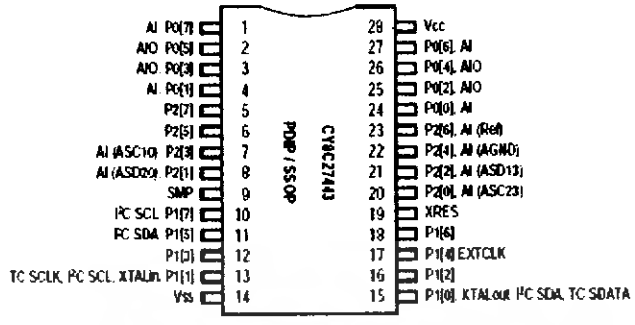
2.4.3 ขาสัญญาณอินพุต/เอาต์พุต (GPIO : General Purpose IO)

ในหัวข้อนี้จะกล่าวถึง คุณสมบัติทางด้านขาสัญญาณ I/O (Input/Output) ของ PSoC MCU ซึ่งขาสัญญาณของ PSoC MCU สามารถทำงานได้ทั้งส่วนของ ขาสัญญาณดิจิตอล และขาสัญญาณอนาลอก ซึ่งในการใช้งานจะผ่านทางรีจิสเตอร์ต่างๆ ดังตารางที่ 2.3

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Acccess
0,xxh	PRTxDR	Data Register								RW : 00
0,xxh	PRTxIE	Bit Interrupt Enables								RW : 00
0,xxh	PRTxGS	Global Select								RW : 00
0,xxh	PRTxDM2	Drive Mode 2								RW : FF
1,xxh	PRTxDM0	Drive Mode 0								RW : 00
1,xxh	PRTxDM1	Drive Mode 1								RW : FF
1,xxh	PRTxIC0	Interrupt Control 0								RW : 00
1,xxh	PRTxIC1	Interrupt Control 1								RW : 00

ตารางที่ 2.3 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งาน I/O (GPIO Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

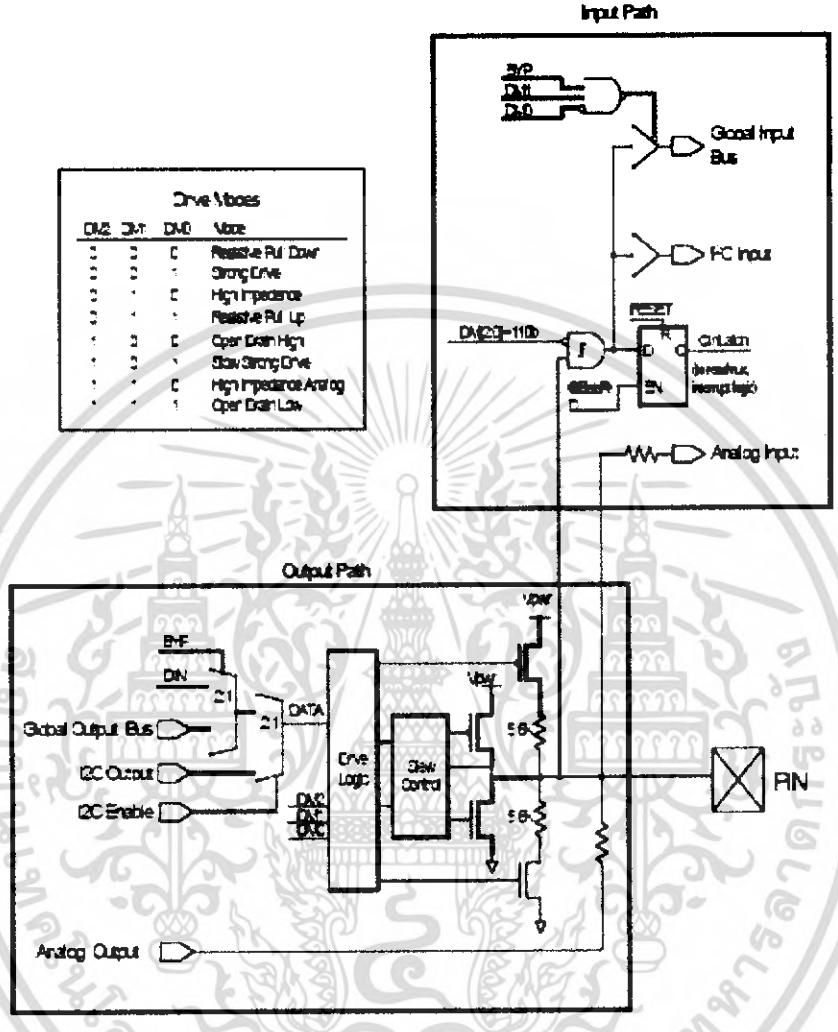


รูปที่ 2.9 แสดงตัวถังและ การวางตำแหน่งขาสัญญาณต่างๆของไอซี CY8C27443

Pin No.	Type		Pin Name	Description
	Digital	Analog		
1	IO	I	P0[7]	Analog column mux input.
2	IO	IO	P0[5]	Analog column mux input and column output.
3	IO	IO	P0[3]	Analog column mux input and column output.
4	IO	I	P0[1]	Analog column mux input.
5	IO		P2[7]	
6	IO		P2[5]	
7	IO	I	P2[3]	Direct switched capacitor block input.
8	IO	I	P2[1]	Direct switched capacitor block input.
9	Power		SMP	Switch Mode Pump (SMP) connection to external components required.
10	IO		P1[7]	I2C Serial Clock (SCL).
11	IO		P1[5]	I2C Serial Data (SDA).
12	IO		P1[3]	
13	IO		P1[1]	Crystal Input (XTALin), I2C Serial Clock (SCL), ISSP-SCLK
14	Power		Vss	Ground connection.
15	IO		P1[0]	Crystal Output (XTALout), I2C Serial Data (SDA), ISSP-SDATA
16	IO		P1[2]	
17	IO		P1[4]	Optional External Clock Input (EXTCLK).
18	IO		P1[6]	
19	Input		XRES	Active high external reset with internal pull down.
20	IO	I	P2[0]	Direct switched capacitor block input.
21	IO	I	P2[2]	Direct switched capacitor block input.
22	IO		P2[4]	External Analog Ground (AGND).
23	IO		P2[6]	External Voltage Reference (VRef).
24	IO	I	P0[0]	Analog column mux input.
25	IO	IO	P0[2]	Analog column mux input and column output.
26	IO	IO	P0[4]	Analog column mux input and column output.
27	IO	I	P0[6]	Analog column mux input.
28	Power		Vdd	Supply voltage.

รูปที่ 2.10 หน้าที่ขาสัญญาณต่างๆของ CY8C27443

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 บล็อกไดอะแกรมของขาสัญญาณ GPIO (GPIO – Block Diagram)

จากรูปที่ 2.11 จะเห็นว่าโครงสร้างขาสัญญาณ GPIO ของ PSoC MCU มีความซับซ้อนมาก ทั้งนี้เพราะ PSoC MCU ได้ออกแบบโครงสร้างขาสัญญาณ ให้สามารถทำงานได้ในหลากหลายคุณสมบัติ โดยผู้ใช้งาน (USER) สามารถกำหนดความต้องการในการใช้งานได้ โดยเราสามารถกำหนดคุณสมบัติการทำงานของขาสัญญาณในโหมดต่าง ได้ดังนี้

- Pull Down (Resistive pull down) โหมดนี้ขาสัญญาณ I/O จะถูกต่อผ่านตัวต้านทานลงกราวด์
- Strong (Strong Drive) โหมดนี้เหมาะสำหรับใช้งานเป็นเอาต์พุตดิจิตอล
- High Z (High impedance) โหลดนี้ทีขาสัญญาณ I/O จะมีความต้านทานสูงเหมาะสำหรับการใช้งานเป็นอินพุต
- Pull Up (Resistive pull up) โหมดนี้ขาสัญญาณ I/O จะถูกต่อผ่านความต้านทานไปยัง Vcc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Open Drain High สถานะของ I/O เป็นแบบ Open Drain High
- Strong Slow (Strong Slow Drive) สถานะสัญญาณ I/O เป็นแบบ Strong Slow
- High Z Analog เป็นสถานะความต้านทานสูงแบบอนาลอก ซึ่งจะเป็นค่าสภาวะเริ่มต้น (Default) หลังจากเกิดรีเซ็ต (reset rate)

Drive Mode DM[2:0]	Pin State	Description
000b	Resistive pull down	Strong high, resistive low
001b	Strong drive	Strong high, strong low
010b	High impedance	Hi-Z high and low, digital input enabled
011b	Resistive pull up	Resistive high, strong low
100b	Open drain high	Slow strong high, Hi-Z low
101b	Slow strong drive	Slow strong high, slow strong low
110b	High impedance, analog (reset state)	Hi-Z high and low, digital input disabled (for zero power) (reset state)
111b	Open drain low	Slow strong low, Hi-Z high

ตารางที่ 2.4 การกำหนด Drive Modes ของ GPIO โดยผ่านรีจิสเตอร์ PRTxDMx[2:0]

การกำหนดคุณสมบัติของขาสัญญาณเหล่านี้ สามารถทำได้สองวิธีคือ การกำหนดที่ค่ารีจิสเตอร์ (PRTxDMx) ซึ่งวิธีนี้จะเป็นการเขียนโปรแกรมเข้าไปเซตค่ารีจิสเตอร์ ดังนั้นวิธีการนี้จึงทำให้เราสามารถเปลี่ยนแปลงคุณสมบัติ Drive Mode ของขาสัญญาณได้ตลอดเวลาขณะที่ CPU กำลังประมวลผลอยู่ และอีกวิธีก็คือ การกำหนดโดยใช้โปรแกรม PSoC Designer ตรงส่วนของ Device Editor Configuration ซึ่งจะเรียกว่าการกำหนด Configure I/O Pins ซึ่งจะช่วยลดความยุ่งยากในการเขียนโปรแกรมลงได้เป็นอย่างดี

ขาสัญญาณ I/O ของ PSoC MCU จะประกอบด้วย บัฟเฟอร์อินพุต และ วงจรขับทางด้านเอาต์พุต โดยขาสัญญาณ I/O เหล่านี้จะถูกจัดไว้เป็นพอร์ต ซึ่งปกติ 1 พอร์ต จะมีทั้งหมด 8 บิต แต่จะมีบางกรณีที่พอร์ตนั้นๆ มีขาสัญญาณไม่ถึง 8 บิต ขาสัญญาณ I/O ต่างๆ เหล่านี้สามารถทำงานในลักษณะต่างๆดังนี้

- Digital IO : เป็นขาสัญญาณดิจิตอล อินพุต/เอาต์พุต สามารถควบคุมการทำงานได้โดยการผ่านค่าให้กับรีจิสเตอร์ PRTxDR
- Global IO : เป็นขาสัญญาณดิจิตอล อินพุต/เอาต์พุต ที่เชื่อมโยงระหว่าง Digital PSoC Block
- Analog IO : เป็นขาสัญญาณ อินพุต/เอาต์พุต ที่เชื่อมโยงระหว่าง Analog PSoC block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.1 Digital IO

เป็นโหมดการทำงานหนึ่งของขาสัญญาณ I/O Port ที่หน่วยประมวลผล M8C สามารถติดต่อเข้าถึงขาสัญญาณพอร์ตได้โดยตรง โดยไม่ต้องอาศัย PSoC Block ไม่ว่าจะเป็นการอ่าน หรือ การเขียนค่าไปที่พอร์ต ซึ่ง M8C จะติดต่อผ่านทางรีจิสเตอร์ PRTxDR ซึ่งสถานะของรีจิสเตอร์นี้จะส่งผลโดยตรงกับขาสัญญาณ I/O Pin ของ PSoC MCU เช่น ในการเขียนข้อมูลลอจิกไปยังพอร์ดผ่านรีจิสเตอร์ PRTxDR ข้อมูลดังกล่าวจะถูกเปลี่ยนให้เป็นสถานะทางไฟฟ้าไปปรากฏที่ขาสัญญาณต่างๆ ส่วนในกรณีการอ่านข้อมูลจากพอร์ตหน่วยประมวลผลสามารถอ่านค่าสถานะทางไฟฟ้าจากขาสัญญาณ I/O ได้จากรีจิสเตอร์ PRTxDR ด้วยเช่นกัน ซึ่งสถานะทางไฟฟ้าจะถูกเปลี่ยนเป็นลอจิกแล้วนำกลับมาเก็บที่รีจิสเตอร์ PRTxDR

2.4.3.2 Global IO

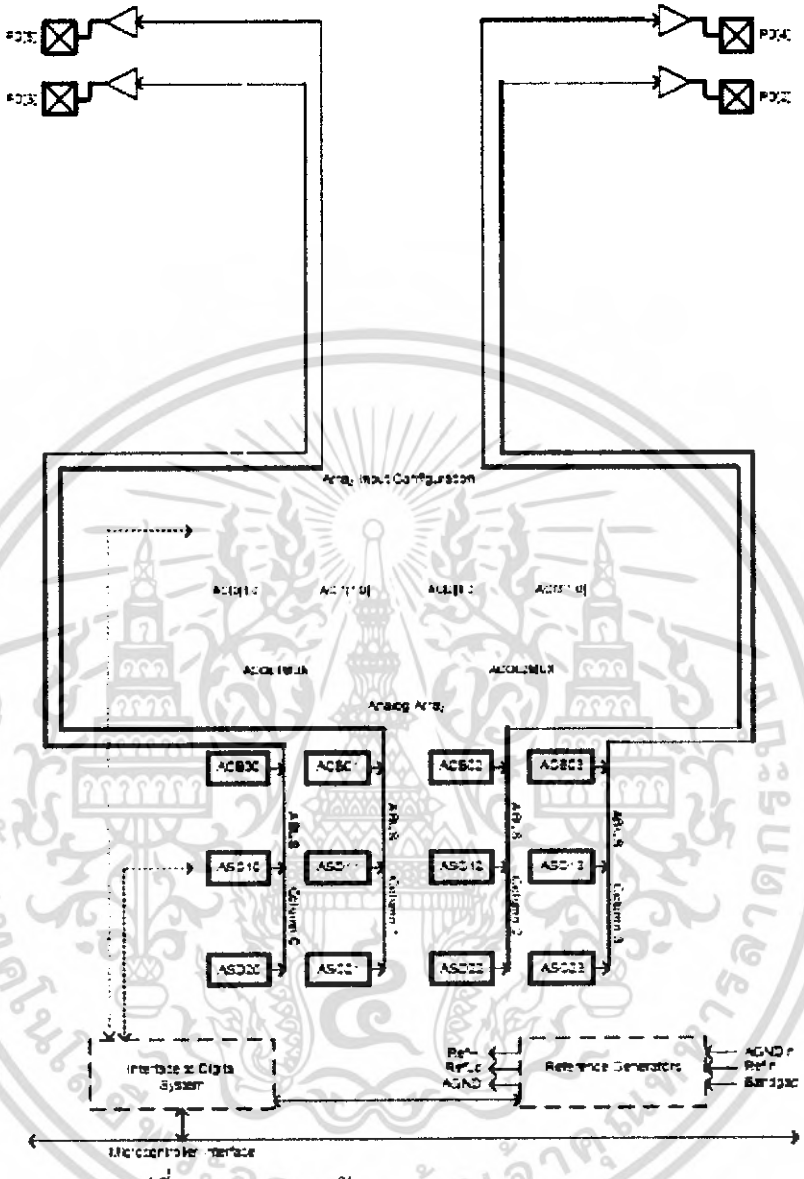
Global IO จะถูกใช้เป็นสัญญาณเชื่อมต่อ (interconnect) ระหว่าง PSoC Block กับขาสัญญาณภายนอกโดยจะเชื่อมต่อผ่าน Global In และ Global Out ซึ่งเราสามารถทำการเชื่อมโยงสัญญาณจาก GPIO เข้ามายัง Global In หรือ Global Out แล้วนำไปเชื่อมต่อกับสัญญาณใน PSoC Block ต่างๆ ได้ตามต้องการ ซึ่งในการออกแบบสามารถทำได้โดยใช้โปรแกรม PSoC Designer ซึ่งในการออกแบบจะเป็นลักษณะวิสชวล หรือ แบบกราฟิก ทำให้ง่ายต่อการออกแบบ และ โดยผู้ออกแบบไม่จำเป็นต้องเข้าไปเซตค่ารีจิสเตอร์ใดๆเลย

2.4.3.3 Analog IO

เป็นขาสัญญาณ Input/Output อนาลอก ซึ่งจะเชื่อมต่อขาสัญญาณภายนอกกับสัญญาณภายในอนาลอกบล็อกผ่านขาสัญญาณ AOUT โดยจะต่อผ่านตัวต้านทานภายในประมาณ 300 โอห์ม สำหรับในโหมดของอนาลอก ชนิดของGPIO จะกำหนดเป็น High Impedance Analog Drive mode (Hi-Z)

ขาสัญญาณ GPIO ที่ทำหน้าที่เป็นขาสัญญาณ Analog Input/Output ซึ่งสามารถเชื่อมต่อกับ Analog Block ได้ จะมีทั้งหมด 4 ขาสัญญาณ คือ ขาสัญญาณ P0[2] , P0[3] , P0[4] และ P0[5] ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

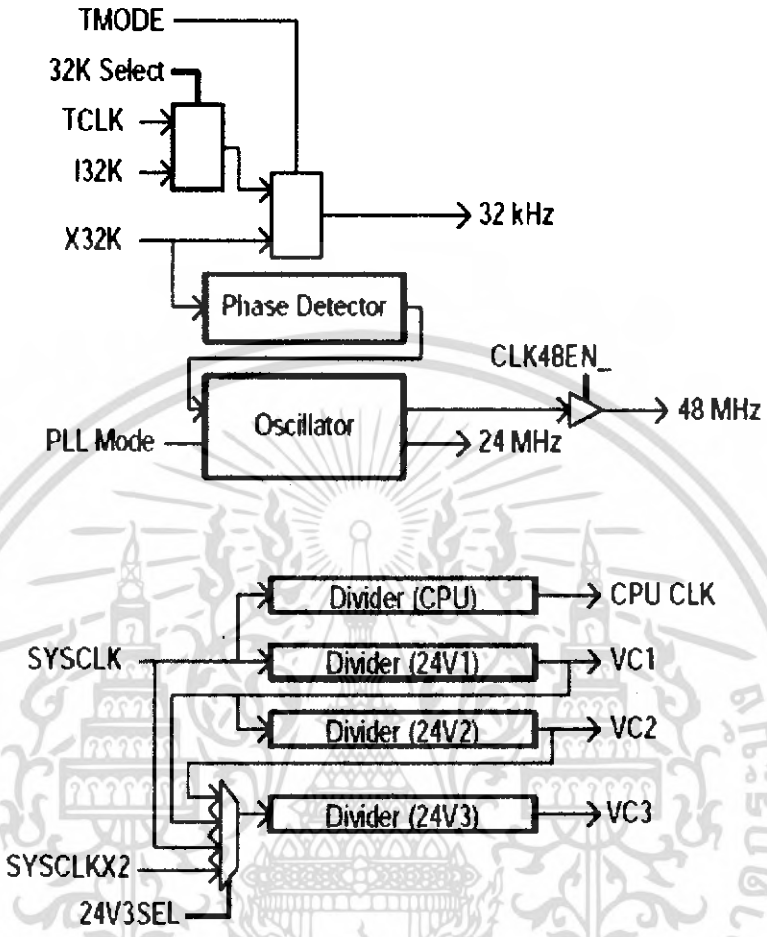


รูปที่ 2.12 แสดงขาสัญญาณ Analog Input/Output

2.4.4 ระบบสัญญาณนาฬิกา

ระบบสัญญาณนาฬิกาของ PSoC MCU เป็นระบบที่มีความหลากหลาย ค่อนข้างซับซ้อนและมีรายละเอียดการทำงานต่างๆมากมาย หากศึกษาให้เข้าใจแล้วจะทำให้เราสามารถใช้งานสัญญาณนาฬิกาต่างๆของ PSoC MCU ได้อย่างมีประสิทธิภาพ และจะพบว่าเป็นระบบโครงสร้างสัญญาณนาฬิกาที่ดีมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ระบบสัญญาณนาฬิกาของ PSoC MCU

โดยเราสามารถแบ่งแหล่งกำเนิดสัญญาณนาฬิกาออกเป็นส่วนๆดังนี้

- Internal main oscillation (IMO)
- Internal low speed oscillator (ILO)
- 32 KHz Crystal oscillator (ECO)
- Phase Lock Loop (PLL)

2.4.4.1 สัญญาณนาฬิกาหลักภายใน Internal Main Oscillator (IMO)

เป็นแหล่งกำเนิดสัญญาณนาฬิกาหลัก ซึ่งเป็นแหล่งกำเนิดสัญญาณนาฬิกาภายใน PSoC MCU โดยสามารถผลิตสัญญาณนาฬิกาได้ ขนาด 24 MHz เรียกว่า SYSCLOCK และยังมีวงจรคูณความถี่เป็นสองเท่า (Clock Double) ซึ่งจะทำให้ความถี่เป็น 48 MHz ซึ่งจะเรียกว่า SYSCLOCK2X โดยเราสามารถเลือกใช้งานได้

นอกจากนี้ยังสามารถปรับแต่งความถี่ (Trim) ได้อีกด้วย โดยรีจิสเตอร์ IMO_TR ที่ใช้สำหรับปรับแต่งค่าความถี่ของสัญญาณนาฬิกาเพื่อให้มีความเที่ยงตรงมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit#	7	6	5	4	3	2	1	0
Access POR	W:0							
Bit Name	Trim [7:0]							

ตารางที่ 2.5 แสดงรีจิสเตอร์ IMO_TR

ในการผลิตสัญญาณความถี่นี้จะได้สัญญาณอินพุตมาจากแหล่งกำเนิดสัญญาณออสซิลเลเตอร์ขนาดความถี่ 32 KHz จากภายในตัว PSoC MCU ซึ่งค่าความคลาดเคลื่อนของสัญญาณนาฬิกาทั้งจาก 24 MHz และ 48 MHz จะอยู่ที่ +/- 2.5% ซึ่งมีผลมาจากสภาวะของอุณหภูมิ และ แรงดัน โดยการใช้ออสซิลเลเตอร์แบบภายในนี้มีข้อดีคือ ไม่จำเป็นต้องใช้อุปกรณ์ภายนอกใดๆ ต่อร่วมเลย แต่อาจจะมีปัญหาในเรื่องของความคลาดเคลื่อนอยู่บ้าง ดังนั้นหากต้องการความเที่ยงตรงของสัญญาณความถี่สูงๆ ควรจะใช้แหล่งสัญญาณนาฬิกา คริสตอลออสซิลเลเตอร์จากภายนอก (ECO) ซึ่งโมดูลของ IMO จะสนับสนุนการใช้งานสัญญาณนาฬิกาจากภายนอกด้วย โดยการอาศัยสัญญาณจาก Phase Lock Loop (PLL) ซึ่งอินพุตของวงจร PLL นี้จะได้จากสัญญาณ 32KHz ของคริสตอลออสซิลเลเตอร์ภายนอก (ECO)

ความเร็วในการทำงานของ CPU นั้นสามารถเลือกได้หลายระดับ โดยการเอาสัญญาณนาฬิกาของระบบ (SYSCLOCK) 24 MHz มาหารด้วยอัตราส่วนต่างๆ ซึ่งกำหนดได้ด้วยรีจิสเตอร์ OSC_CR0 ในบิต 2:0 ตามตารางที่ 2.6 หรือ สามารถกำหนดได้จากหน้าต่าง Global Resource ของซอฟแวร์ PSoC Designer

OSC_CR0[2:0]	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/8
001b	6 MHz	EXTCLK/4
010b	12 MHZ	EXTCLK/2
011b	24 MHz	EXTCLK/1
100b	1.5 MHz	EXTCLK/16
I01b	750 KHz	EXTCLK/32
110b	187.5 KHz	EXTCLK/128
111b	93.7 KHz	EXTCLK/256

ตารางที่ 2.6 การกำหนดความเร็วการทำงานของ CPU กำหนดโดยรีจิสเตอร์ OSC_CR0[2:0]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\

2.4.4.2 สัญญาณนาฬิกาแบบความเร็วต่ำภายใน (Internal Low Speed Oscillator (ILO))

เป็นสัญญาณนาฬิกาแบบความเร็วต่ำ (32KHz) ที่อยู่ภายใน PSoC MCU โดยปกติแล้วจะใช้กำเนิดสัญญาณให้กับ การทำงานในส่วนประหยัดพลังงาน (Sleep) และ watchdog นอกจากนี้ก็ยังสามารถนำเอาไปใช้เป็นสัญญาณนาฬิกาให้กับ Digital PSoC blocks ได้อีกด้วย

เราสามารถกำหนดระดับการใช้พลังงานของสัญญาณออสซิลเลเตอร์ ILO ได้ 3 โหมดด้วยกัน โดยการกำหนดค่าให้กับรีจิสเตอร์ ILO_TR ในบิต Bias Trim [5:4] ดังนี้ คือ

- โหมดปกติ (normal power mode) เหมาะสำหรับการใช้งานทั่วไปที่ต้องการความถูกต้องของสัญญาณสูง
- โหมดพลังงานต่ำ (low power mode) เหมาะสำหรับการทำงานในโหมดประหยัดพลังงาน (Sleep Mode) แต่ในโหมดนี้จะทำให้ความถูกต้องของสัญญาณต่ำลงด้วย

นอกจากการกำหนดระดับพลังงานของออสซิลเลเตอร์แล้ว ยังสามารถทำการปรับค่าความถี่สัญญาณ (Freq Trim) ได้อีกด้วยโดยการปรับค่ารีจิสเตอร์ ILO_TR ในบิต Trim[3:0] ดังต่อไปนี้

รีจิสเตอร์ ILO_TR

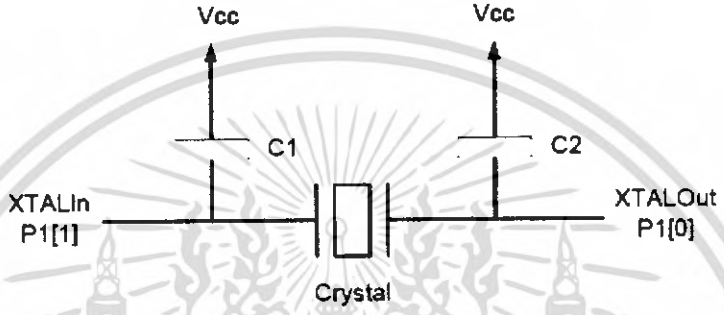
Bit#	7	6	5	4	3	2	1	0
Access POR			W:0		W:00			
Bit Name	Reserved		Bias Trim[5:4]		Freq Trim[3:0]			

Bit	Name	Description
Bits 7:6	Reserve	สงวนไว้ไม่ใช้งาน
Bits 5:4	Bias Trim	ปรับระดับพลังงานของออสซิลเลเตอร์ (ILO)
		00b Medium Bias
		01b Maximum Bias
		I0b Minimum Bias
		11b Reserved
Bits 3:0	Freq Trim	ปรับความถี่ของออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.3 สัญญาณนาฬิกา 32 KHz จากคริสตอลภายนอก (32KHz Crystal Oscillator(ECO))

เป็นแหล่งกำเนิดสัญญาณนาฬิกาที่ได้จากภายนอกโดยวงจรคริสตอลออสซิลเลเตอร์ 32 KHz โดยวงจรการต่อนั้นจะต่อคริสตอลออสซิลเลเตอร์เข้ากับขาสัญญาณ XTAL In และ XTAL Out นอกจากนี้จะต้องต่อตัวเก็บประจุด้วยอีก 2 ตัว ดังรูปที่ 2.14



รูปที่ 2.14 วงจรคริสตอลออสซิลเลเตอร์ภายนอก

สัญญาณนาฬิกาจาก ECO นี้สามารถใช้เป็นฐานเวลาให้กับแหล่งกำเนิดสัญญาณนาฬิกาหลัก (IMO) ได้โดยการใช้งานในโหมด PLL ซึ่งจะได้สัญญาณนาฬิกาหลัก (IMO) ที่มีความเที่ยวตรงสูง ขนาดความถี่ 24 MHz ซึ่งจะเป็นสัญญาณนาฬิกาของระบบ (System Clock) นอกจากนี้การต่อสัญญาณนาฬิกาแบบ ECO ยังสนับสนุนการใช้งานคริสตอลที่ความถี่ 32.768 KHz ซึ่งเป็นความถี่ที่ใช้กับไอซีนาฬิกา (Real Time Clock(RTC)) เหมาะกับการนำไปประยุกต์ใช้งานเป็นนาฬิกา เพราะจะให้เวลาที่เที่ยงตรงเช่นเดียวกับนาฬิกาทั่วๆไป ซึ่งการใช้งานออสซิลเลเตอร์ในโหมด ECO นี้ ในบิต 7 ของรีจิสเตอร์ OSC_CR0 จะต้องเซตเป็น 1 ด้วย (ตามค่า default บิตนี้จะเป็น 0)

-การใช้งานออสซิลเลเตอร์แบบ ECO

โดยการใช้งานสัญญาณนาฬิกา ECO นั้น จะแบ่งออกเป็น 2 ลักษณะคือ

- ใช้เป็นแหล่งกำเนิดสัญญาณ CLK32K หรือ ใช้เป็นฐานเวลาให้กับ Sleep Timer
- ใช้เป็นฐานเวลาให้กับสัญญาณนาฬิกาหลัก (IMO) ซึ่งในกรณีนี้ จะต้องมีการใช้งาน PLL (Phase Lock Loop) โดยจะได้สัญญาณนาฬิกาของระบบ (SYSCLK) ขนาด 24 MHz ที่มีความเที่ยงตรงสูงมาก

จากการใช้งานในสองลักษณะดังกล่าวจะมีการใช้งานค่าตัวเก็บประจุที่มีความแตกต่างกัน โดยกรณีของการใช้งาน ECO แบบไม่ใช้ PLL จะใช้ค่าตัวเก็บประจุ C1 และ C2 ที่มีค่าเท่ากันดังตารางที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package	C1	C2
8 PDIP	22 pF	22 pF
20 PDIP	22 pF	22 pF
20 SOIC	22 pF	22 pF
20 SSOP	22 pF	22 pF
28 PDIP	22 pF	22 pF
28 SOIC	22 pF	22 pF
28 SSOP	22 pF	22 pF
44 TQFP	22 pF	22 pF
48 PDIP	20 pF	20 pF
48 SSOP	22 pF	22 pF

ตารางที่ 2.7 ค่าตัวเก็บประจุที่ใช้กับวงจรออสซิลเลเตอร์ภายนอก ECO แบบไม่ใช้ PLL

ส่วนค่าตัวเก็บประจุของวงจรคริสตอลในกรณีที่ใช้งาน PLL จะต้องใช้ค่าตัวเก็บประจุ C1 และ C2 ที่มีค่าแตกต่างกัน ดังตารางที่ 2.8

Package	C1	C2
8 PDIP	12 pF	100 pF
20 PDIP	12 pF	100 pF
20 SOIC	12 pF	100 pF
20 SSOP	12 pF	100 pF
28 PDIP	12 pF	100 pF
28 SOIC	12 pF	100 pF
28 SSOP	12 pF	100 pF
44 TQFP	12 pF	100 pF
48 PDIP	9 pF	100 pF
48 SSOP	12 pF	100 pF

ตารางที่ 2.8 ค่าตัวเก็บประจุที่ใช้กับวงจรคริสตอลภายนอก ECO แบบใช้ PLL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-การกำหนดการใช้งานสัญญาณนาฬิกาจาก ECO

การกำหนดการใช้งานสัญญาณนาฬิกาแบบ ECO นั้นสามารถทำการกำหนดค่าต่างๆ ในหน้าต่าง Device Editor ของซอร์ฟแวร์ PSoC Designer โดยกำหนด 32K_Select เป็น External และหากต้องการให้สัญญาณนาฬิกาจาก ECO นี้เป็นฐานเวลาให้กับสัญญาณนาฬิกาหลัก IMO ก็สามารถทำได้โดยเลือก PLL Mode เป็น Ext Lock นอกจากนี้จะต้องกำหนดให้ขาสัญญาณ P1[0] เป็น XTAL Out และ P1[1] เป็น XTAL In อีกด้วย

เมื่อเรากำหนดการทำงาโดยเลือกสัญญาณนาฬิกาจากภายนอก (ECO) เมื่อ CPU เริ่มต้นทำงานหรือหลังจากเกิดการรีเซต การทำงานของระบบจะอาศัยสัญญาณนาฬิกาจากภายในไประยะหนึ่งเพื่อรอให้สัญญาณจากคริสตอลภายนอก (ECO) และ PLL มีเสถียรภาพมากพอ ก่อนจึงจะสวิตซ์การทำงานมาใช้สัญญาณนาฬิกาจากคริสตอลออสซิลเลเตอร์ภายนอก แต่ถ้าหากพบว่าสัญญาณนาฬิกาจากภายนอกไม่ทำงาน มันจะสวิตซ์สัญญาณนาฬิกากลับไปทำงานจากสัญญาณนาฬิกาภายใน

การใช้สัญญาณจากคริสตอลออสซิลเลเตอร์ภายนอกนั้น จะให้ความถูกต้องเที่ยงตรงกว่าการใช้สัญญาณนาฬิกาจากภายใน ดังนั้นการใช้สัญญาณนาฬิกาจาก ECO จึงเหมาะกับการทำงานที่ต้องใช้ความแม่นยำของสัญญาณนาฬิกาสูง เช่น การใช้งาน DTMF หรือ RS232 เป็นต้น

2.4.4.4 Phase Locked Loop (PLL)

Phase-Locked Loop (PLL) เป็นฟังก์ชันการทำงานในการสร้างสัญญาณนาฬิกาของระบบ โดยอาศัย อินพุตสัญญานาฬิกาจากวงจรคริสตอลออสซิลเลเตอร์ภายนอก ซึ่งจะให้ความเที่ยงตรงสูงกว่าการใช้สัญญาณนาฬิกาภายนอก โดยเอาพุตสัญญาณของ PLL จะมีค่าความถี่เท่ากับ 23.986 MHz เมื่อใช้ความถี่คริสตอลเท่ากับ 32.768 KHz โดยสัญญาณความถี่เอาพุตของ PLL จะถูกใช้เป็นสัญญาณนาฬิกาของระบบ (SYSCLK) โดยการใช้งาน PLL สามารถทำได้โดยซอร์ฟแวร์ PSoC Designer ดังขั้นตอนต่อไปนี้

- เลือกสัญญาณนาฬิกา 32K_Select เป็น External
- เลือก PLL_Mode เป็น Ext Lock
- กำหนดให้ขาสัญญาณ P1[0] เป็น XTAL Out และ P1[1] เป็น XTAL In

2.4.4.5 การทำงานในโหมด Sleep และ Watchdog

Sleep Mode หรือ โหมดการประหยัดพลังงาน จุดประสงค์ของการทำงานในโหมดนี้ก็เพื่อที่จะประหยัดพลังงานของ CPU ให้มากที่สุดเท่าที่จะเป็นไปได้ ซึ่งเราสามารถควบคุมการทำงานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sleep Mode ได้โดยการเขียนโปรแกรมควบคุม ซึ่งเมื่อเราสั่งให้ CPU เข้าสู่การทำงานในโหมด Sleep จะมีผลทำให้ CPU หยุดทำงานและทำให้ แหล่งกำเนิดสัญญาณนาฬิกา 24/48MHz , หน่วยความจำ Flash และ Bandgap-Voltage อยู่ในโหมดพลังงานต่ำ (Power-Down)

Watchdog Timer คือส่วนที่ทำหน้าที่ในการเฝ้าระวังการทำงานของ CPU โดยจุดประสงค์ในการทำงานของ Watchdog Timer เพื่อป้องกันการทำงานที่ผิดพลาดของ CPU ในที่นี้หมายถึง CPU ไม่สามารถทำในสิ่งที่ควรจะทำได้ หรือ CPU เกิดการค้าง (halt stage) ไม่สามารถทำงานต่อได้โดย Watchdog Timer จะทำการรีเซต CPU ในทันทีที่ค่าการนับของมันสิ้นสุดลง เพื่อให้ CPU กลับไปตั้งต้นการทำงานใหม่ได้

Sleep Timer

Sleep Timer คือค่าของการทำงานอยู่ใน Sleep Mode ซึ่งจะเป็นไทเมอร์ขนาด 15 บิต แบบนับขึ้นซึ่งจะนับสัญญาณจากแหล่งกำเนิดสัญญาณ 32 KHz โดยสามารถเลือกได้ว่าจะใช้จากภายใน (IMO) หรือจากภายนอก (EMO) โดยสามารถเลือกช่วงเวลา Sleep Timer ได้ดังในตารางที่ 2.9

Sleep Interval	Sleep Period	Watchdog Period
(Default) 00	1.95 mS (512 Hz)	6 mS
01	15.6 mS (64 Hz)	47 mS
10	125 mS (8 Hz)	375 mS
11	1 mS (1 Hz)	3 sec

ตารางที่ 2.9 คาบเวลาของ Sleep Timer และ Watchdog Timer

จากตารางที่ 2.9 จะเห็นว่าคาบเวลาของ Watchdog Timer จะมีค่าเวลาเป็น 3 เท่าของค่าเวลาของ Sleep Timer โดยการเลือกคาบเวลาดังกล่าว สามารถทำได้โดยการกำหนดค่าที่หน้าต่าง Global resource ในโปรแกรม PSoC Designer

การเปิดการทำงานของ sleep timer สามารถทำได้โดยการเซตบิตที่ [3] ของรีจิสเตอร์ CPU_SCR0 โดยตรง หรือเซตค่าผ่านมาโครฟังก์ชัน M8C Sleep ของ M8C ดังตารางที่ 2.10

ฟังก์ชั่น M8C	ความหมาย
M8C_EnableWatchDog	;Enable การทำงานของ Watchdog
M8C_ClearWDTAndSleep	;เคลียร์ค่าการทำงานของ Watchdog Timer และ Sleep Timer
M8C_ClearWDT	;เคลียร์ค่า Watchdog Timer
M8C_EnableGInt	;Enable Global Interrupt
M8C_DisableGInt	;Disable Global Interrupt
M8C_Sleep	;เข้าสู่ Sleep Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 ตัวอย่างฟังก์ชันการใช้งานของ M8C ที่เกี่ยวข้องกับ Sleep Timer และ Watchdog

การออกจาก Sleep Mode (Wake Up from Sleep)

เมื่อเข้าสู่การทำงานในสภาวะ Sleep ระบบต่างๆจะทำงานด้วยพลังงานต่ำ และในการออกจากสภาวะการทำงานในสภาวะ Sleep Mode จะเรียกว่า Wake Up โดยสามารถเกิดขึ้นจากการเกิดอินเตอร์รัพท์ และ จากการรีเซตต่างๆ เช่น Sleep Timer Interrupt , GPIO(pin) interrupts และ Timer Clocked Externally เป็นต้น

2.4.4.6 การรีเซตของระบบ (System Resets)

การรีเซตของระบบจะมีผลให้ CPU กลับไปทำงานในสถานะเริ่มต้น และ ทำให้รีจิสเตอร์ต่างๆ มีค่าสถานะ กลับไปตามค่าของ default states โดยการเกิดการรีเซตสามารถเกิดขึ้นได้จากสาเหตุต่างๆดังนี้

- Power-on Reset (POR) เป็นการรีเซตที่เกิดจากแหล่งจ่ายแรงดันของ CPU มีแรงดันต่ำกว่าระดับที่กำหนด (ต่ำกว่าแรงดัน Trip Voltage ที่ตั้งไว้)
- External Reset (XRES) เป็นการรีเซ็ตจากขาสัญญาณ XRES ของ PSoC MCU โดยจะเกิดการรีเซตเมื่อขาสัญญาณ XRES ได้รับสภาวะลอจิกสูง
- Watchdog Reset (WDR) เป็นการรีเซ็ตที่เกิดจากการทำงานของ Watchdog Timer
- Internal Reset(IRES) การรีเซตที่เกิดขึ้นจากกระบวนการภายใน

1.Power on Reset (POR)

เมื่อเริ่มต้นจ่ายพลังงานให้กับ CPU วงจร POR จะทำการรีเซต CPU พร้อมทั้งตรวจสอบแรงดันไฟเลี้ยงที่ขา Vdd กับแรงดัน Trip Voltage (LVD) ซึ่งจะทำให้ CPU อยู่ในสถานะรีเซตจนกว่าแรงดันของ Vdd จะมีค่า สูงกว่าแรงดันที่จุด Trip Voltage ได้จากการกำหนดค่าในบิต VM[2:0] (ในรีจิสเตอร์ VLC_TR) หรือกำหนดจากหน้าต่าง Global Resource ในซอร์ฟแวร์ PSoC Designer

2.External Reset (XRES)

การรีเซตจากขาสัญญาณ XRES จะเกิดการรีเซตขึ้นได้ก็ต่อเมื่อขาสัญญาณ XRES ได้รับสัญญาณสภาวะลอจิก 1 ซึ่งโดยปกติแล้วจะต้องทำการต่อตัวต้านทานแบบ Pull Down ให้กับขาสัญญาณนี้

3.Watchdog Timer Reset(WDR)

จุดประสงค์ในการทำงานของ Watchdog Timer ก็เพื่อป้องกันการทำงานที่ผิดพลาดของ CPU ในที่นี้หมายถึง CPU ไม่สามารถทำงานในสิ่งที่ควรจะเป็นได้ หรือ CPU เกิดค้าง (halt state)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

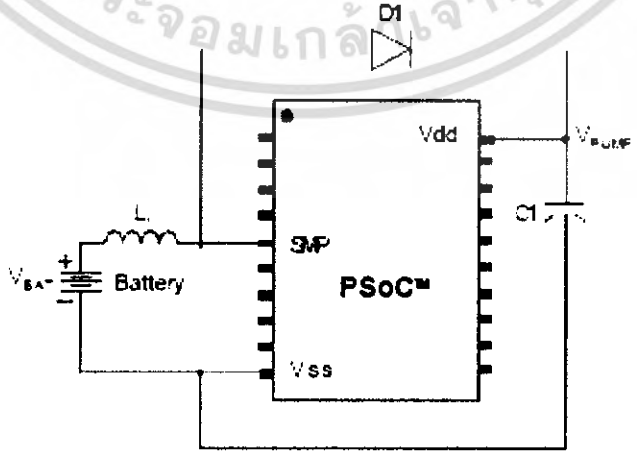
ไม่สามารถทำงานต่อได้ โดย Watchdog Timer จะทำการรีเซต CPU ในทันทีที่ค่าการนับของมันสิ้นสุดลงเพื่อให้ CPU กลับไปตั้งต้นการทำงานใหม่ เราสามารถ Enable หรือ Disable การทำงานของ Watchdog Timer ได้โดยการเคลียร์บิต POR (บิต 4 ของรีจิสเตอร์ CPU_SCR(0)) หลังจากที่เราได้เปิดการทำงานของ WDR แล้ว เราจะไม่สามารถปิดการทำงานของ WDR ได้ด้วยการเขียนคำสั่งไปเซตค่า ให้กับบิต PORS การปิดการทำงานของ WDR จะสามารถเกิดขึ้นได้จากการรีเซตของ POR และ XRES เท่านั้น

นอกจากการการเปิดการทำงานของ WDR โดยการเคลียร์บิต PORS แล้วยังสามารถทำการเปิดการทำงาน WDR โดยการเปิดค่าในซอร์ฟแวร์ PSoC Designer

ค่าเวลาของ Watchdog Timer จะได้มาจากแหล่งกำเนิดสัญญาณนาฬิกา 32KHz ซึ่งสามารถเลือกได้ว่าจะใช้สัญญาณขนาด 32 KHz จากคริสตอลภายนอก (ECO) หรือ จากภายใน (ILO) โดยค่า default จะถูกกำหนดไว้เป็น 32 KHz ภายใน (ILO) และเมื่อต้องการเปลี่ยนมาใช้สัญญาณจากภายนอก (ECO) ก็ให้ทากรเซตบิตที่ 7 ของรีจิสเตอร์ OSC_CR0 ซึ่งสัญญาณ 32 KHz ที่ถูกเลือกนี้สามารถนำไปใช้ ทั้งกับ Watchdog Timer และ การทำงานในโหมดประหยัดพลังงาน (Sleep Mode)

4. Switch Mode Pump (SMP)

SMP คือวงจรที่ใช้ในการสวิตซ์แรงดันจากแบตเตอรี่ และ เซลล์พลังงาน จากระดับแรงดันต่ำๆ ให้มีระดับแรงดันที่สูงขึ้นจนสามารถเลี้ยงการทำงานของ CPU ได้ โดยภายในไอซี PSoC MCU จะมีฮาร์ดแวร์ภายในที่รองรับการทำงานในส่วนนี้ แต่ก็ต้องมีการต่อวงจรภายนอกในบางส่วน คือ ต่อตัว อินดัคเตอร์ (Inductor) ระหว่างขั้วแรงดันแบตเตอร์รี่กับขาสัญญาณ SMP ของ PSoC MCU และควรจะมีค่าคาปาร์ซิเตอร์ต่อบายพาสจากขาสัญญาณดังกล่าวลงกราวด์ Vss ด้วย โดยค่าเก็บประจุของคาปาร์ซิเตอร์อย่างต่ำควรจะประมาณ 0.1uF ดังรูปที่ 2.15



รูปที่ 2.15 วงจร Switch Mode Pump (SMP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากออกแบบวงจร และกำหนดค่าอุปกรณ์ที่เหมาะสมจะทำให้ได้แรงดันสูงพอที่จะเลี้ยงการทำงานของ PSOC MCU ถึงแม้จะใช้แบตเตอรี่ขนาด 1.5 V เพียงก้อนเดียว MCU ก็สามารถทำงานได้โดยการใช้งานโหมด SMP สามารถทำได้โดยการกำหนดที่หน้าต่าง Global Resource ของโปรแกรม PSoC Designer

2.5 การอินเตอร์รัพท์ (Interrupt Operation)

ในบทนี้จะกล่าวถึงการทำงานในส่วนของราชการ อินเตอร์รัพท์ โดยปกติการประมวลผลของโปรเซสเซอร์จะมีการประมวลผลคำสั่งแบบเป็นลำดับเรียงต่อ ๆ กันไป ซึ่งการอินเตอร์รัพท์ ก็คือการขัดจังหวะการทำงานที่เป็นลำดับนั้นเพื่อให้ โปรเซสเซอร์มาประมวลผลในส่วนของโปรแกรมอินเตอร์รัพท์ เมื่อทำงานในส่วนของโปรแกรมอินเตอร์รัพท์เสร็จเรียบร้อยแล้ว โปรเซสเซอร์จะกลับไปยังโปรแกรมหลักเพื่อทำงานปกติต่อไป

การอินเตอร์รัพท์ของ PSoC Microcontroller สามารถเกิดขึ้นได้จากแหล่งกำเนิดอินเตอร์รัพท์หลายแหล่งด้วยกัน เช่น จาก Digital Block, Analog block, supply voltage, sleep, variable clocks และ GPIO (pin) โดยผู้ใช้งาน (User) สามารถควบคุมการเกิดหรือ ไม่เกิด อินเตอร์รัพท์เหล่านี้ได้ด้วยการเขียนโปรแกรมกำหนดค่าให้กับรีจิสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการอินเตอร์รัพท์ซึ่งจะได้กล่าวในรายละเอียดต่อไป

ลำดับความสำคัญของการอินเตอร์รัพท์	แอดเดรสการอินเตอร์รัพท์	ชื่ออินเตอร์รัพท์
0 (highest)	0000h	Rest
1	0004h	Supply voltage monitor
2	0008h	Analog column 0
3	000Ch	Analog column 1
4	0010h	Analog column 2
5	0014h	Analog column 3
6	0018h	VC3
7	001Ch	GPIO
8	0020h	PSoC block DBB00
9	0024h	PSoC block DBB01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10	0028h	PSoC block DCB02
11	002Ch	PSoC block DCB03
12	0030h	PSoC block DBB10
13	0034h	PSoC block DBB11
14	0038h	PSoC block DCB12
15	003Ch	PSoC block DCB13
24	0060h	12C
25 (lowest)	0064h	Sleep timer

ตารางที่ 2.11 ตำแหน่งเวกเตอร์การอินเตอร์รัพท์ต่าง ๆ ของ CY8C27xxx

จากตารางที่ 2.11 จะแสดงตำแหน่งแอดเดรสและลำดับความสำคัญของการอินเตอร์รัพท์ โดยจะเรียงจากลำดับการอินเตอร์รัพท์ที่มีนัยความสำคัญสูงสุดไปยังลำดับที่มีนัยความสำคัญต่ำสุด ทั้งนี้เนื่องจาก PSoC MCU สามารถเกิดอินเตอร์รัพท์ได้จากแหล่งกำเนิดสัญญาณอินเตอร์รัพท์หลาย ๆ แหล่ง ดังนั้นเพื่อจำแนกชนิดของสัญญาณอินเตอร์รัพท์ จึงต้องมีการแยกแอดเดรสเวกเตอร์ของการอินเตอร์รัพท์ออกเป็นส่วน ๆ เรียงตามลำดับความสำคัญ เพื่อให้สามารถทำการตอบสนองต่อสัญญาการอินเตอร์รัพท์ได้อย่างถูกต้อง

2.5.1 โครงสร้างของการอินเตอร์รัพท์

การอินเตอร์รัพท์จะเป็นการทำงานในส่วนของฮาร์ดแวร์ภายในไมโครคอนโทรเลอร์ วงจรการทำงานเหล่านี้ผู้ใช้ (User) จะสามารถกำหนดลักษณะความต้องการของการทำงานได้ด้วยการเขียนโปรแกรมควบคุมผ่านการกำหนดค่าให้กับรีจิสเตอร์ต่าง ๆการทำงานของการอินเตอร์รัพท์ เริ่มตั้งแต่บล็อก Interput Source ซึ่งเป็นแหล่งกำเนิดของการอินเตอร์รัพท์ เช่น Time, GPIO และอื่น ๆ ดังที่ได้กล่าวมาแล้ว ซึ่งโมดูลต่าง ๆ เหล่านี้ หากเข้าเงื่อนไขของการเกิดอินเตอร์รัพท์ก็จะส่งสภาวะร้องขออินเตอร์รัพท์โดยจะส่งสัญญาณลอจิก “1” ไปแลตช์ค้างไว้ที่ Posted Interrupt ดังรูปที่ 3.1 (สภาวะนี้จะแลตช์ค้างไว้จนกว่าจะมีการส่งสัญญาณมาเคลียร์ โดยการเคลียร์บิต INT_CLRx) ซึ่งหากบิต INT_MSKx ของอินเตอร์รัพท์นั้น ๆ ไม่อยู่ในสภาวะเซต (Disable) การอินเตอร์รัพท์ในส่วนนั้น ๆ ก็จะไม่สามารถเกิดขึ้นได้ แต่ถ้าหากมีการ เซตบิต INT_MSKx (มีการ Enbie) สัญญาณร้องขอการอินเตอร์รัพท์ที่แลตช์ค้างไว้ตรง Posted Interrupt ก็จะถูกส่งเข้าไปยังส่วนของ Priority Encoder ซึ่งเป็นส่วนของการเข้ารหัสเพื่อจัดลำดับความสำคัญของการอินเตอร์รัพท์ ทั้งนี้ก็เพราะว่าการอินเตอร์รัพท์สามารถเกิดขึ้นได้จากหลาย ๆ แหล่ง จึงต้องมีการจัดลำดับความสำคัญ ซึ่งอินเตอร์รัพท์ที่มีความสำคัญสูงสุดกว่าจะถูกจัดให้ตอบสนองการทำงานก่อน จะเห็นว่าต่อจากส่วน

เอกสารนี้มีลิขสิทธิ์ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ในทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Encoder จะมีสัญญาณบิต GIE ต่อกับอินพุตข้างหนึ่งของ And Gate ซึ่งบิตนี้จะเป็นบิตที่มีนัยสำคัญสูงสุดที่จะควบคุมให้เกิด หรือไม่ เกิด อินเตอร์รัพท์ทั้งหมดได้ โดยถ้าเซตเป็น "1" (Enable) จะหมายถึงการอนุญาตให้เกิดอินเตอร์รัพท์ใด ๆ ได้ทั้งหมด แต่ถ้าเคลียร์เป็น "0" จะไม่อนุญาตให้มีการอินเตอร์รัพท์ใด ๆ เกิดขึ้นเลย โดยบิต GIE จะอยู่ที่รีจิสเตอร์ F ในตำแหน่งบิตที่ 0

- ลำดับของการเกิดอินเตอร์รัพท์

1. เมื่อมีสัญญาณอินเตอร์รัพท์เกิดขึ้น ไม่ว่าจะจากกรณีต่าง ๆ คือ เกิดเงื่อนไขอินเตอร์รัพท์ เช่น การเกิดไทม์เอาต์ของ Timer โดยบิต INT_MSKx และบิต GIE จะต้องถูกเซตไว้ด้วย

2. เมื่อเกิดเหตุการณ์ตามข้อที่ 1 โปรเซสเซอร์จะยังคงทำการเอ็กซิคิวซ์ คำสั่งปัจจุบันในขณะนั้นให้เสร็จเสียก่อนจึงจะเข้าไปตอบสนองการอินเตอร์รัพท์

3. จากนั้น โปรแกรมรองรับการอินเตอร์รัพท์ภายในจะถูกเอ็กซิคิวซ์ ซึ่งใช้เวลาไป 13 ไซเคิล โดยในระหว่างช่วงเวลานี้จะมีสิ่งเกิดขึ้นดังนี้

   - ค่าของรีจิสเตอร์ PCH, PCL และ F จะถูกเก็บลงในหน่วยความจำสแต็ก ตามลำดับ
   - รีจิสเตอร์ F จะถูกเคลียร์ทำให้บิต GIE ถูกเคลียร์เป็น "0" ซึ่งจะทำให้อินเตอร์รัพท์ทั้งหมดถูกปิดการทำงาน (Disable)
   - รีจิสเตอร์ PCH (PC[5:8]) ถูกเคลียร์เป็นศูนย์
   - ตำแหน่งแอดเดรสเตอร์ของการอินเตอร์รัพท์จะถูกอ่านออกมาจากส่วนควบคุมอินเตอร์รัพท์ แล้วนำมาเก็บที่รีจิสเตอร์ (PC[7:0]) มีผลทำให้ตำแหน่งแอดเดรสของรีจิสเตอร์ PC เปลี่ยนแปลงไปในตำแหน่งเวกเตอร์ของอินเตอร์รัพท์ดังกล่าว เช่น ถ้าเป็นการอินเตอร์รัพท์จาก GPIO ตำแหน่งของ PC ก็จะเป็น 001Ch เป็นต้น

4. โปรเซสเซอร์จะเอ็กซิคิวซ์คำสั่ง ที่อยู่ในตำแหน่งเวกเตอร์อินเตอร์รัพท์ ซึ่งก็คือ คำสั่ง LIMP ซึ่งคำสั่งนี้จะสั่งให้โปรเซสเซอร์กระโดดไปทำงานยังตำแหน่งของโปรแกรมตอบสนองการอินเตอร์รัพท์ (Interrupt Service Routine (ISR) ตามแต่ละประเภทของการอินเตอร์รัพท์

5. ในระหว่างที่โปรเซสเซอร์กำลังประมวลผลในโปรแกรมตอบสนองการอินเตอร์รัพท์อยู่นี้ บิต GIE จะยังคงเป็นศูนย์อยู่ทำให้ไม่มีการอินเตอร์รัพท์ซ้อนเกิดขึ้น แต่ถ้าเราต้องการเปิดอินเตอร์รัพท์ก็สามารถทำได้โดยการเซตบิต GIE ในรีจิสเตอร์ F แต่จะต้องคอยจัดการเรื่องของ Stack เอง เพราะอาจจะเกิด Stack overflow ขึ้นได้ ดังนั้นจึงไม่แนะนำ

6. โปรแกรมบริการอินเตอร์รัพท์จะถูกปิดท้ายด้วยคำสั่ง RETI ซึ่งเป็นคำสั่งที่ทำให้โปรเซสเซอร์รีเทิร์นกลับไปทำงานยังตำแหน่งของโปรแกรมหลักเพื่อทำคำสั่งที่ถูกขัดจังหวะมาก่อนหน้านี้ โดยเมื่อโปรเซสเซอร์เอ็กซิคิวซ์คำสั่ง RETI สิ่งที่เกิดขึ้นคือ ค่ารีจิสเตอร์ต่าง ๆ ที่ถูกเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ใน Stack เมื่อครั้งที่เกิดอินเตอร์รัพท์ ก็จะถูกคืนค่ากลับ (POP) ดังนี้ คือ รีจิสเตอร์ F, PCL และ PCH นอกจากนี้ บิต GIE ยังถูกเซตกลับไปเป็น "I" ด้วย ทำให้การอินเตอร์รัพท์ถูกเปิด (Enable Interrupt) ตามเดิม

7. หลงจากทำคำสั่ง RETI โปรเซสเซอร์จะกลับมาทำคำสั่งถัดจากคำสั่งก่อนหน้าที่ถูกขัดจังหวะจากการอินเตอร์รัพท์ อย่างไรก็ตาม ในการอินเตอร์รัพท์ย่อมจะมีเวลาที่สูญเสียไปก่อนที่โปรเซสเซอร์จะได้ประมวลผลในโปรแกรม รองรับการอินเตอร์รัพท์ที่ผู้ใช้งานได้ออกแบบไว้ ซึ่งค่าเวลาที่ศูนย์เสียไปนี้เราจะเรียกมันว่า ค่า **"Interrupt Latency"** โดยค่าเวลานี้จะเริ่มนับตั้งแต่เมื่อมีเงื่อนไขของอินเตอร์รัพท์ใด ๆ เกิดขึ้น ไปจนถึงเวลาที่โปรเซสเซอร์ได้เอ็กซ์ซิคิวซ์คำสั่งแรกของโปรแกรมบริการอินเตอร์รัพท์ ซึ่งสามารถคำนวณได้ตามสมการดังต่อไปนี้

Latency = [เวลาที่โปรเซสเซอร์ต้องเอ็กซ์วิคิวซ์คำสั่งปัจจุบันให้เสร็จสิ้น] + [เวลาที่โปรเซสเซอร์เตรียมการก่อนอินเตอร์รัพท์ (เช่น การเก็บค่า PCH, PCL, รีจิสเตอร์ F ลงหน่วยความจำแสต๊ก และเคลียร์บิต GIE)] + [เวลาของการเอ็กซ์ซิคิวซ์คำสั่ง LJMP เพื่อกระโดดไปยังตำแหน่งของโปรแกรมบริการอินเตอร์รัพท์]

ตัวอย่างเช่น

โปรเซสเซอร์กำลังเอ็กซ์ซิคิวซ์คำสั่ง JMP ซึ่งใช้เวลา 5 ไซเคิล แล้วมีเหตุการณ์อินเตอรรัพท์เกิดขึ้นโปรเซสเซอร์จะทำคำสั่งที่ทำอยู่ในปัจจุบันให้เสร็จก่อน จากนั้นจะตอบสนองการอินเตอร์รัพท์ด้วยการเก็บค่ารีจิสเตอร์ต่าง ๆ คือ PCH และรีจิสเตอร์ F ลงในหน่วยความจำแสต๊ก ซึ่งจะใช้เวลา 13 ไซเคิล จากนั้นจะทำการเอ็กซ์ซิคิวซ์คำสั่ง LJMP ใช้เวลา 7 ไซเคิล โดยสามารถคำนวณได้ดังสมการต่อไปนี้

Latency = (1 to 5 cycles for JMP to finish) + (13 cycles for interruptroutine) + (7 cycles for LMP)

= 21 to 25 cycles.

* ในตัวอย่างนี้หาก CPU ทำงานที่ความถี่ 24 MHz เวลาที่ใช้ในการทำงาน 25 ไซเคิล จะเท่ากับ

1/24 MHz x 25 = 1.042 uS

นี่คือค่าเวลาที่สูญเสียไปในกระบวนเกิดอินเตอร์รัพ ก่อนที่คำสั่งในโปรแกรมบริการอินเตอร์รัพท์จะถูกทำการเอ็กซ์ซิคิวซ์จริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัพท์

ในการควบคุมการอินเตอร์รัพท์เราจะทำการควบคุมและกำหนดค่าการทำงานให้กับรีจิสเตอร์ต่างๆ ดังนั้นเราจำเป็นที่จะต้องทราบรายละเอียดของรีจิสเตอร์ต่างๆ เหล่านั้นด้วย ซึ่งรีจิสเตอร์ต่างๆ ที่เกี่ยวข้องกับการอินเตอร์รัพท์จะแสดงไว้ในตารางที่ 2.12

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh	INT_CLR0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	VMonitor	RW:00
0,DBh	INT_CLR1	DCB13	DCB12	DBB11	DBB10	DCB03	DCB02	DBB01	DBB00	RW:00
0,DDh	INT_CLR3								12C	RW:00
0,DEh	INT_MSK3	ENSWINT							12C	RW:00
0,E0h	INT_MSK0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	VMonitor	RW:00
0,E1h	INT_MSK1	DCB13	DCB12	DDB11	DBB10	DCB03	DCB02	DBB01	DBB00	RW:00
0,E2h	INT_VC									RW:00
X,F7h	CPU_F				XOI		Carry	Zero	GIE	RW:00

ตารางที่ 2.12 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัพท์

*** หมายเหตุ**

ตัวเลขที่อยู่หน้าเครื่องหมายคอมม่า (,) ในช่อง Address เป็นหมายเลขแบงก์ของรีจิสเตอร์ ส่วน x ที่อยู่หน้าเครื่องหมายคอมม่า (,) หมายถึงแบงก์ใด ๆ ก็สามารถเข้าถึงรีจิสเตอร์นี้ได้ การแสดงความสัมพันธ์ในแต่ละบิตของรีจิสเตอร์ INT_CLRx,INT_MSKx และบิต GIE โดยแต่ละบิตจะมีความสัมพันธ์กันเปรียบได้ดังกับการนำเอาสวิตซ์มาต่ออนุกรมกัน ถ้าหากตัวใดตัวหนึ่งเปิด (Open Circuit) ก็ไม่สามารถส่งสัญญาณไปร้องขอการอินเตอร์รัพท์ได้ ซึ่งการปิดหรือเปิดสวิตซ์เหล่านี้ก็เปรียบได้กับการ เซต หรือ เคลียร์บิตต่างๆ นั่นเอง โดยจะเห็นว่าบิต GIE จะมีน้อยความสำคัญสูงสุด โดยจะเป็นตัวที่ควบคุมการเกิดอินเตอร์รัพท์ทั้งหมด โดยรายละเอียดของรีจิสเตอร์ต่างๆจะเป็นดังนี้

2.5.2.1 รีจิสเตอร์ INT_CLRx

รีจิสเตอร์นี้ จะมีอยู่ด้วยกันสามตัวคือ INT_CLR0,INT_CLR1และINT_CLR3 ซึ่งจะใช้สำหรับเคลียร์สถานะของการร้องขออินเตอร์รัพท์ที่เกิดขึ้นจากเงื่อนไขการอินเตอร์รัพท์ต่างๆซึ่งจะมีความสอดคล้องกับรีจิสเตอร์INT_MSKx แบบบิตต่อบิต กล่าวคือ การอินเตอร์รัพท์ก็จะไม่สามารถเกิดขึ้นได้เลย ถ้าหากว่าบิตต่างๆในรีจิสเตอร์ INT_CLRx เป็น “0” และแม้ในบิตรีจิสเตอร์ INT_MSKsจะเป็น “1” อยู่ก็ตาม นอกจากนี้ยังมีความสัมพันธ์กับบิต ENSWINT (ในรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INT_MSK3[7]) ซึ่งถ้าบิตนี้เป็น "0" การเขียนค่าใดๆในรีจิสเตอร์ INT_CLRx ก็จะไม่ส่งผลใดๆ ดังนั้นหากต้องการใช้งานอินเตอร์รัพท์ได้ โดยรายละเอียดของรีจิสเตอร์ในแต่ละบิตจะเป็นดังต่อไปนี้

INT_CLR0 : 0,DAh

	7	6	5	4	3	2	1	0
Access:POR	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0
BitName	VC0	Cleep	GPIO	Analog3	Analog2	Analog1	Analog0	V.monitor

Bit [7] VC3

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Variable Clock 3 เกิดขึ้น

1 = มีสัญญาณ Posted interrupt สำหรับ Variable Clock 3 เกิดขึ้น

- กรณีการเขียนค่า

เขียนค่าด้วย 0 และ บิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนค่าด้วย 1 และ บิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนค่าด้วย 0 และ บิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนค่าด้วย 1 และ บิต ENSWINT = 1 สัญญาณ Posted interrupt ของ VC3 จะเกิดขึ้น

Bit [6] Sleep

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Sleep timer เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Sleep timer เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = I ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิต ENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt ของ Sleep timer ขึ้น

Bit [5] GPIO

- กรณีการอ่านค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 = ไม่มีสัญญา Posted interrupt ของ GPIO เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ GPIO เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิตENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt ของ GPIO

Bit[4:1] Analog3,Analog2,Analog0

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Analog columns เกิดขึ้น

1 = สัญญาณ Posted interrupt ของ Analog columns เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1ทำให้เกิดสัญญาณ Posted interruptของAnalog columns

Bit[0] V.Monitor

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interruptของVoltage monitor

INT_CLR1 : 0,DBh

	7	6	5	4	3	2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Access:POR	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0
Bit Name	DCB13	DCB12	DCB11	DCB10	DCB03	DCB02	DCB01	DCB00

[7] DCB13 Digital Communications Block type B,row 1,position 3

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[6] DCB Digital Communication Block type B,row 1 ,position 2

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[5] DBB11 Digital Block typeB,row1,position 1

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[4] DBB10 Digital Block typeB,row1,position 0

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[3] DCB03 Digital Communication Block type B,row 0 ,position 3

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[2] DCB02 Digital Communication Block type B,row 0 ,position 2

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[1] DBB02 Digital Communication Block type B,row 0 ,position 2

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

[0] DBB00 Digital Basic Block type B,row 0 ,position 0

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

1 = มีสัญญาณ Posted interrupt ของ Voltage monitor เกิดขึ้น

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt

	7	6	5	4	3	2	1	0
Access:POR								RW:0
Bit Name								12C

Bit [7:1] Reserved สงวนไว้ไม่ใช้งาน

Bit [0] 12C

- กรณีการอ่านค่า

0 = ไม่มีสัญญาณ Posted interrupt เกิดขึ้น

1 = มีสัญญาณ Posted interrupt เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีการเขียนค่า

เขียนด้วยค่า 0 และบิต ENSWINT = 0 สัญญาณ Posted interrupt ของ 12C จะถูกเคลียร์

เขียนด้วยค่า 1 และบิต ENSWINT = 0 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 0 และบิต ENSWINT = 1 ไม่มีผลกระทบใดๆ

เขียนด้วยค่า 1 และบิตENSWINT = 1 ทำให้เกิดสัญญาณ Posted interrupt ของ 12 C

2.5.2.2 INT_MSKx Register

เป็นรีจิสเตอร์ที่ควบคุมการเกิดอินเตอร์รัพท์โดยจะควบคุมสัญญาณร้องขออินเตอร์รัพท์ที่เข้ามาที่ Posted interrupt ว่าจะให้ผ่านเข้าไปในส่วนของ Priority Encoder หรือไม่ ซึ่งจากรูปที่ 3.1 จะเห็นว่าถ้าบิตของรีจิสเตอร์ INT_MSKx เป็น "0" สัญญาณร้องขอการอินเตอร์รัพท์ของแหล่งสัญญาณอินเตอร์รัพท์นั้น จะไม่สามารถผ่านไปได้ซึ่งสัญญาณร้องขอการอินเตอร์รัพท์นี้จะยังคงอยู่และสามารถเคลียร์ค่าสถานการณ์ร้องขอนี้ด้วยการเคลียร์ค่าในรีจิสเตอร์ INT_CLRx ซึ่งจะมีความสัมพันธ์กันแบบบิตต่อบิต แต่ถ้าในกรณีที่บิต INT_MSKx เซตเป็น "1" สัญญาณร้องขอการอินเตอร์รัพท์ก็จะสามารรถส่งผ่านเข้าไปยัง Priority Encoder ได้ และหากบิต GIE อยู่ในสภาวะ (GIE=1) สัญญาณอินเตอร์รัพท์ก็จะสามารถผ่านเข้าไปแจ้งให้กับโปรเซสเซอร์ (M8CX\) เพื่อให้โปรเซสเซอร์เข้ามาตอบสนองกับสัญญาณอินเตอร์รัพท์นั้นๆได้ โดยรายละเอียดของรีจิสเตอร์แต่ละตัวจะเป็นดังนี้

INT_MSK : 0,E0h

	7	6	5	4	3	2	1	0
Access:POR	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0
BitName	VC0	Cleep	GPIO	Analog3	Analog2	Analog1	Analog0	V.monitor

Bit [7] VC3 :

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก VC3 interrupt

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก VC3 interrupt

Bit[6] Sleep

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Sleep interrupt

I = อนุญาตให้มีการอินเตอร์รัพท์จาก Sleep interrupt

Bit[5] GPIO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก GPIO interrupt

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก GPIO interrupt

Bit[4] Analog 3

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column3

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column3

Bit[3] Analog 2

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column2

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column2

Bit[2] Analog 1

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column1

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column1

Bit[1] Analog 0

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column0

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Analog interrupt,column0

Bit[0] V.Monitor

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จาก Voltage monitor interrupt

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Voltage monitor interrupt

INT_MSK1: 0,E1h

	7	6	5	4	3	2	1	0
Access:POR	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0
Bit Name	DCB13	DCB12	DCB11	DCB10	DCB03	DCB02	DCB01	DCB00

Bit[7] DCB13

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigitalCommunicationsBlock,row1,position 3

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalCommunicationsBlock,row1,position 3

Bit[6] DCB12

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigitalCommunicationsBlock,row1,position 2

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalCommunicationsBlock,row1,position 2

Bit[5] DBB11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigital Basic Block,row1,position 1

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Digital Basic BlockB,row1,position 1

Bit[4] DBB10

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigital Basic Block,row1,position 0

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก Digital Basic BlockB,row1,position 0

Bit[3] DCB03

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigitalCommunicationsBlock,row0,position3

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalCommunicationsBlock,row0,position 3

Bit[2] DCB02

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigitalCommunicationsBlock,row0,position2

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalCommunicationsBlock,row0,position 2

Bit[1] DBB01

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigital Basic Block,row0,position1

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalBasic Block,row0,position 1

Bit[0] DBB00

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์จากDigital Basic Block,row0,position0

1 = อนุญาตให้มีการอินเตอร์รัพท์จาก DigitalBasic Block,row0,position 0

INT_MSK3 : 0,DEh

	7	6	5	4	3	2	1	0
Access:POR	RW:0							RW:0
Bit Name	ENSWINT							12C

Bit[7] ENSWIT

0 = ไม่อนุญาตกระบวนการทำงานของซอร์ฟแวร์อินเตอร์รัพท์

1 = อนุญาตกระบวนการทำงานของซอร์ฟแวร์อินเตอร์รัพท์

Bit[6:1] Reserved สงวนไว้ไม่ใช้งาน

Bit[0] 12C

0 = ไม่อนุญาตให้มีการอินเตอร์รัพท์ จาก 12 C

1 = อนุญาตให้มีการอินเตอร์รัพท์ จาก 12 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.3 INT_VC Register

เป็นรีจิสเตอร์สำหรับเคลียร์เวกเตอร์ของอินเตอร์รัพท์ (Interrupt Clear register (INT_VC) โดยจะมีการทำงานอยู่สองลักษณะด้วยกันคือ การอ่านค่า และการเขียนค่า โดยเมื่อรีจิสเตอร์นี้ถูกอ่านมันจะให้ค่าไบต์ข้อมูลไบต์ต่ำ (LSB) ของเวกเตอร์อินเตอร์รัพท์ที่มีนัยสำคัญสูงสุดที่ค้างอยู่ในส่วนของการจัดลำดับความสำคัญของการอินเตอร์รัพท์ (Priority Interrupt) ในขณะนั้นออกมา ตัวอย่างเช่น มีการอินเตอร์รัพท์เข้ามาพร้อมกันระหว่าง GPIO กับ 12C ซึ่งจะเข้ามายังส่วนจัดลำดับความสำคัญของการอินเตอร์รัพท์ เมื่อเราทำการอ่านค่ารีจิสเตอร์ INT_VC นี้ เราจะได้ค่าเป็น 1Ch (เป็นค่าเวกเตอร์ของอินเตอร์รัพท์แบบGPIO) ทั้งนี้เพราะGPIO มีนัยความสำคัญสูงกว่า 12C แต่ถ้าหากไม่มีการอินเตอร์รัพท์ใดๆเราจะอ่านค่ารีจิสเตอร์ได้เป็น 00h ซึ่งเป็นค่าของรีเซ็ตเวกเตอร์ ส่วนกรณีของการเขียนค่าให้กับรีจิสเตอร์ INT_VC นี้หากมีค่าใดๆก็ตามถูกเขียนให้กับรีจิสเตอร์ INT_VC จะทำให้ค่าเวกเตอร์ของอินเตอร์รัพท์ต่างๆที่รออยู่ในส่วนจัดลำดับความสำคัญของการอินเตอร์รัพท์ถูกเคลียร์

INT_VC:0,E2h

	7	6	5	4	3	2	1	0
Access:POR	RC:0							
Bit Name	Pending Interrupt [7:0]							

Bit [7:0] Pending Interrupt [7:0]

- กรณีของการอ่าน

จะให้ค่าเวกเตอร์ของการอินเตอร์รัพท์ที่มีนัยสำคัญสูงสุดที่ค้างอยู่ในส่วนเข้ารหัสลำดับความสำคัญของการอินเตอร์รัพท์ (Priority Encoder) ออกมา

- กรณีของการเขียน

เมื่อเขียนค่าใด ๆ ให้กับรีจิสเตอร์นี้จะทำให้ ค่าเวกเตอร์ต่างๆ ที่ค้างอยู่ในส่วนเข้ารหัสลำดับความสำคัญของการอินเตอร์รัพท์ (Priority Encoder) ถูกเคลียร์ทั้งหมด

2.5.2.4 CPU_F Register

รีจิสเตอร์ CPU_F จะมีส่วนเกี่ยวข้องกับการอินเตอร์รัพท์ในบิต CiE ซึ่งเป็นบิตที่มีนัยความสำคัญสูงสุดในการควบคุมการเกิดอินเตอร์รัพท์ ซึ่งถ้าบิตนี้เซต (GIE= "1") อินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆ ก็จะสามารถเกิดขึ้นได้แต่ถ้าหากบิตนี้ถูกเคลียร์ (GIE= "0") การอินเตอร์รัพท์ใด ๆ ก็จะไม่สามารถเกิดขึ้นได้เลย แต่ถ้าเราต้องการใช้งานอินเตอร์รัพท์เราก็สามารถทำได้โดยการเขียนคำสั่งไปเซตบิดดังกล่าว การเซตหรือเคลียร์บิตนั้นเนื่องจาก M8C ไม่มีคำสั่งสำหรับการ เซต หรือ เคลียร์บิต ดังนั้นแนะนำให้ใช้คำสั่ง AND F,expr (สำหรับการเคลียร์บิต) และ OR R,expr (สำหรับการเซตบิต) ซึ่งรายละเอียดต่างๆของรีจิสเตอร์ F ได้กล่าวไว้แล้ว

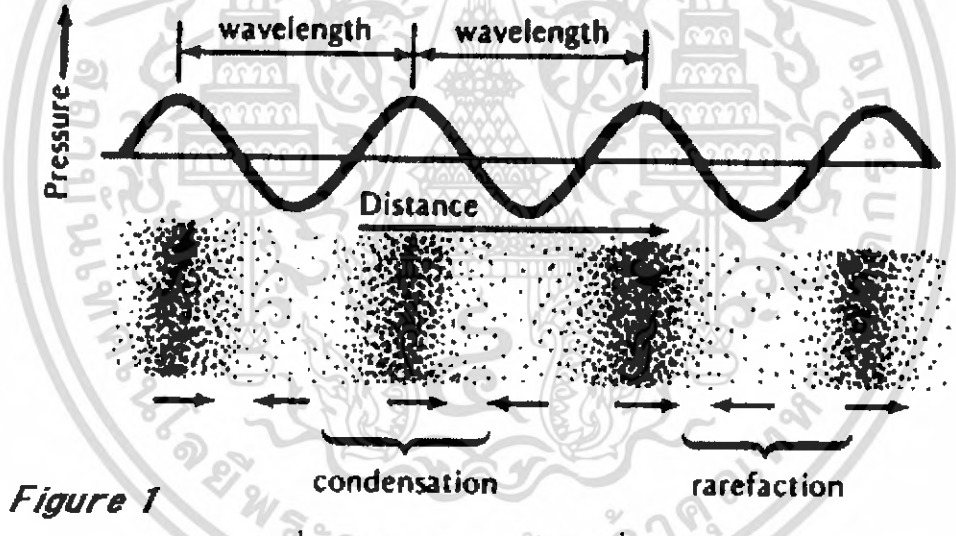
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
คลื่นอัลตราโซนิค

3.1 คลื่น (Wave)

คลื่นเกิดจากการแกว่งหรือสั่น (Oscillated) ของวัตถุ ซึ่งจะเกิดเป็นพลังงานและถ่ายทอดออกจากแหล่งกำเนิดคลื่นไปยังบริเวณโดยรอบ โดยคลื่นทั่วไปสามารถจำแนกตามความสัมพันธ์ระหว่างทิศทางการสั่นของอนุภาคตัวกลางกับทิศทางการเคลื่อนที่ได้ 2 ชนิด คือ

3.1.1 คลื่นตามยาว (Longitudinal Wave) คือคลื่นที่อนุภาคของตัวกลางมีการเคลื่อนไปในทิศทางเดียวกันกับการเคลื่อนที่ของคลื่น เช่น คลื่นเสียง เป็นต้น ซึ่งแสดงลักษณะของคลื่นยาวได้ดังรูปที่ 3.1

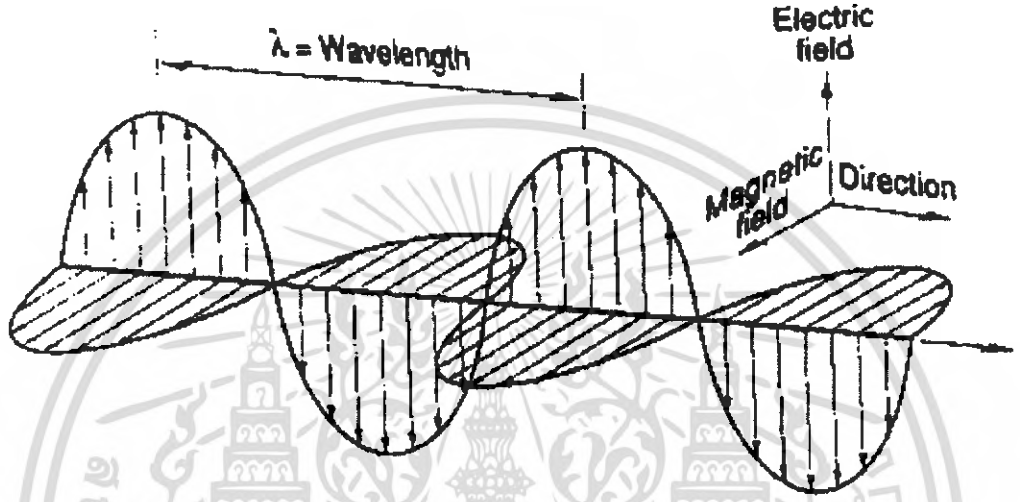


รูปที่ 3.1 แสดงลักษณะการเกิดคลื่นตามยาว

จากรูปจะเห็นได้ว่ามีส่วนที่เป็นคลื่นอัด (Compression) คลื่นช่วงที่อนุภาคของตัวกลางมีความดันสูง และคลื่นขยาย (Reaction) คือ คลื่นช่วงที่อนุภาคของตัวกลางมีความดันต่ำ เมื่อนำค่าของความดัน (P) ที่เปลี่ยนไปตามระยะทางมาเขียนกราฟจะได้เป็นรูป Sine Wave โดยยอดคลื่นจะตรงกับส่วนอัด ท้องคลื่นจะตรงกับส่วนขยายระยะทางระหว่างส่วนอัดถึงส่วนอัด หรือส่วนขยายถึงส่วนขยายคือ 1 ความยาวคลื่น และมีคาบเวลาเป็น T ซึ่งมีค่าเท่ากับ 1/f โดยจุดที่เป็นเส้นแกนนั้นมีค่าความดัน 1 บรรยากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 คลื่นตามขวาง (Transverse Wave) เป็นคลื่นที่อนุภาคตัวกลางมีการเคลื่อนที่ไปในทิศทางตั้งฉากกับทิศทางการเคลื่อนที่ของคลื่น เช่น คลื่นแม่เหล็กไฟฟ้า, คลื่นน้ำ เป็นต้น โดยรูปที่ 3.2 เป็นการแสดงถึงลักษณะของคลื่นแม่เหล็กไฟฟ้าซึ่งเกิดจากสนามแม่เหล็ก (Magnetic Field) และสนามไฟฟ้า (Electric Field) มีการเคลื่อนที่ตั้งฉากกับทิศทางของคลื่นนั้นเอง



รูปที่ 3.2 แสดงลักษณะการเกิดคลื่นตามขวาง

3.2 คลื่นเสียง (Sound Wave)

คลื่นเสียงนั้นจัดอยู่ในรูปคลื่นตามยาว ซึ่งสามารถเดินทางผ่านวัตถุตัวกลางในลักษณะอัดและขยายทำให้เกิดเสียงขึ้นมา โดยแหล่งกำเนิดเสียงแบ่งได้ตามวัตถุกำเนิดเสียงได้ 3 ประเภท คือ

1. เกิดจากการสั่นของสายหรือแท่ง ได้แก่ เครื่องสาย ๆ , ส้อมเสียง ฯลฯ
2. เกิดจากการสั่นของแผ่นผิว ได้แก่ ไดอะเฟรมของลำโพง , หนังกลอง ฯลฯ
3. เกิดจากการสั่นของลำอากาศ ได้แก่ เครื่องเป่าต่าง ๆ ฯลฯ

เนื่องจากคลื่นเสียงนั้นต้องอาศัยตัวกลางในการเดินทาง จึงทำให้คลื่นเสียงไม่สามารถผ่านสูญญากาศได้

3.2.1 ลักษณะของคลื่นเสียง

"ความถี่" ความถี่ของคลื่นเสียงในตัวกลางใด ๆ คือ ความถี่ของการสั่นของแหล่งกำเนิดเสียงนั้น ความถี่ที่มนุษย์สามารถได้ยินมีค่าอยู่ในช่วง 20 Hz – 20 kHz เรียกว่า "Audible Frequency" ความถี่ต่ำกว่าที่มนุษย์ได้ยินเรียกว่า "Inflared Frequency" และความถี่ที่สูงกว่าที่มนุษย์ได้ยินเรียกว่า "Ultrasonic Frequency"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

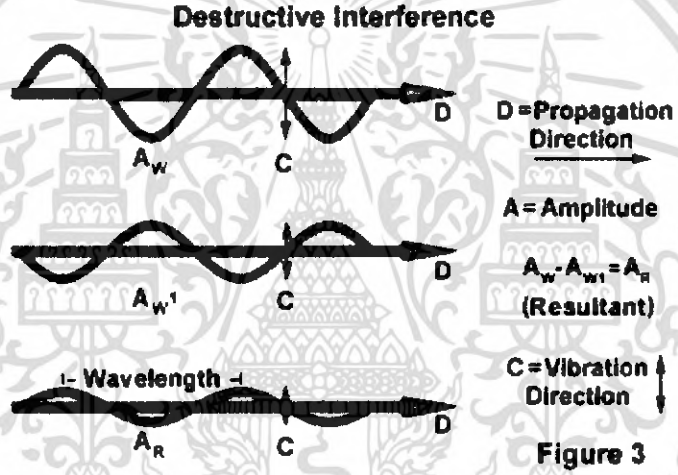
"อัตราเร็วของเสียง" อัตราเร็วของคลื่นเสียงนั้นย่อมแปรผันตามคุณสมบัติของตัวกลางที่มีเสียงเดินผ่าน ซึ่งตัวแปรที่จำกัดอัตราเร็วของเสียง คือ ความหนาแน่น (Density), อุณหภูมิ (Temperature) และความยืดหยุ่น (Elasticity)

3.2.2 คุณสมบัติที่สำคัญของคลื่นเสียง

คุณสมบัติที่สำคัญของคลื่นเสียง แบ่งได้เป็น 4 แบบ คือ

1. การแทรกสอด (Interference) เกิดจากการรวมกันของคลื่น 2 คลื่นขึ้นไป ขณะเมื่อพบกันในตัวกลาง (Medium) เดียวกัน ซึ่งทำให้เกิดผลได้หลายลักษณะ เช่น

การบีท (Beat) ของคลื่นเป็นปรากฏการณ์ที่เกิดจากการรวมคลื่นที่มีความถี่ต่างกัน หรือต่างเฟสกันเคลื่อนที่ไปในตัวกลางเดียวกันแล้วรวมเป็นคลื่นใหม่ ซึ่งทำให้แอมปลิจูดเปลี่ยนไป

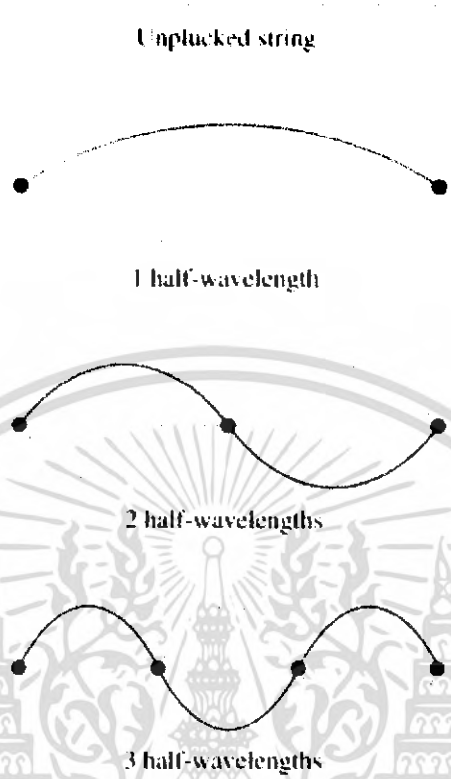


รูปที่ 3.3 แสดงการบีทของคลื่น

จากรูปที่ 3.3 เป็นการแสดงการบีทของคลื่น 2 คลื่น ซึ่งได้คลื่นที่มีแอมปลิจูดเปลี่ยนไป ซึ่งประโยชน์ของการบีทของคลื่น จะนำไปใช้ในการเปรียบเทียบความถี่ของคลื่นให้แสดงผลออกมาในลักษณะของแอมปลิจูดที่แตกต่างกัน

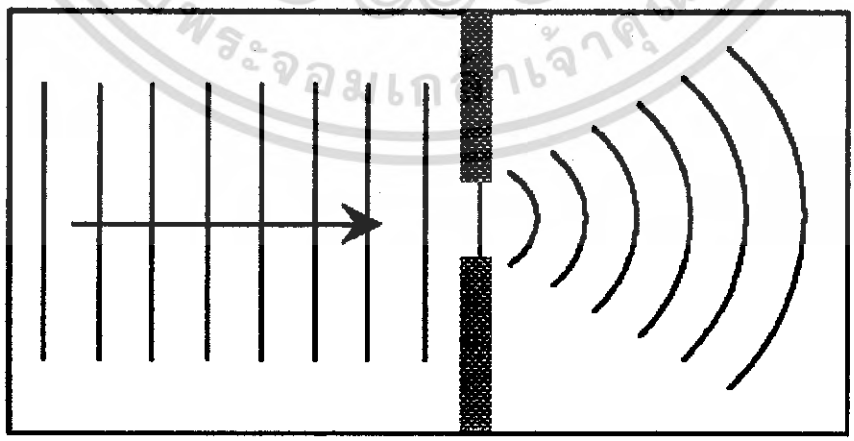
การเกิดคลื่นนิ่ง (Standing Wave) เกิดจากแทรกสอดของคลื่นชัดที่มีแอมปลิจูดเท่ากันและความถี่เดียวกัน แต่มีทิศทางการเคลื่อนที่ตรงกันข้าม หรือมีเฟสตรงข้ามกัน ซึ่งจะทำให้เกิดคลื่นนิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงลักษณะการเกิดคลื่นนิ่ง

2. การเลี้ยวเบนของคลื่นเสียง (Diffraction) คลื่นเสียงจะเลี้ยวเบนอ้อมสิ่งกีดขวางที่มีลักษณะเป็นมุมหรือช่องแคบ ซึ่งปรากฏการณ์เช่นนี้เราพบอยู่ในชีวิตประจำวันอยู่ตลอดเวลา เช่น ในกรณีที่เราได้ยินเสียงแตรรถที่แล่นอยู่คนละถนนของมุมตึก หรือการได้ยินเสียงที่ลอดผ่านช่องเล็ก ๆ จากอีกห้องหนึ่ง ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงลักษณะของคลื่นเสียงที่ผ่านช่องแคบ ๆ จนเสมือนเป็นแหล่งกำเนิดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การสะท้อนของคลื่นเสียง (Reflection) คลื่นเสียงสามารถสะท้อนได้เมื่อตกกระทบตัวกลางโดยที่มุมตกกระทบเท่ากับมุมสะท้อน และจะทำให้เกิดเสียงก้อง (Echo) ซึ่งเสียงก้องที่สะท้อนกลับมาในเวลาที่มากกว่า 50 ms จะทำให้ได้ยินเสียงนั้นเป็นครั้งที่ 2

4. การหักเหของคลื่นเสียง (Refraction) คลื่นเสียงเมื่อเดินทางผ่านตัวกลางที่มีความหนาแน่นต่างกันจะเกิดการหักเหของคลื่น ซึ่งทำให้ความเร็วของคลื่นเสียงเปลี่ยนไป โดยที่ความถี่ยังคงที่อยู่ ดังรูปที่ 3.6

รูปที่ 3.6 แสดงการหักเหของคลื่นเสียงในตัวกลางที่มีความหนาแน่นต่างกัน

3.3 คลื่นอัลตราโซนิค

คลื่นอัลตราโซนิค (Ultrasonic Wave) เป็นคลื่นเสียงที่มีความถี่สูงเกินกว่าที่หูมนุษย์จะได้ยิน ดังนั้นปกติแล้วคำว่า "อัลตราโซนิค" จึงหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20 kHz ขึ้นไป สาเหตุที่มีการนำเอาคลื่นย่านอัลตราโซนิคมาใช้ก็เพราะว่าเป็นคลื่นที่มีทิศทาง ทำให้เราสามารถเล็งคลื่นเสียงไปยังเป้าหมายที่ต้องการได้โดยเจาะจง เรื่องนี้เป็นคุณสมบัติของคลื่นอย่างหนึ่ง การมีทิศทางของคลื่นเสียงย่านอัลตราโซนิค (Ultrasonic) ทำให้เรานำไปใช้งานได้หลายอย่าง

คลื่นอัลตราโซนิค คือคลื่นเสียงที่มีความถี่สูงเกินกว่าที่หูมนุษย์จะสามารถได้ยิน ซึ่งมีความถี่ตั้งแต่ 20 kHz – 10^4 GHz โดยคลื่นที่มีความถี่มากกว่า 10^4 GHz และมีค่าแอมปลิจูดสูง ๆ จะเรียกว่า "ไฮเปอร์ซาวด์" (Hypersound) โดยคุณสมบัติของคลื่นที่อัลตราโซนิคจะมีคุณสมบัติเช่นเดียวกับคลื่นเสียงที่กล่าวมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นอัลตราโซนิคส่วนใหญ่จะกำเนิดขึ้นจากการออสซิเลชั่น (Oscillation) ของคลื่นแม่เหล็กไฟฟ้าที่เป็นคลื่นซายน์ (Sinusoidal) ซึ่งจะทำให้เกิดคลื่นอัลตราโซนิค ซึ่งมีความถี่ต่าง ๆ ซึ่งความยาวคลื่นของคลื่นในแต่ละความถี่จะแตกต่างไปด้วย โดยความยาวคลื่นจะสัมพันธ์กับค่าความเร็ว (c) และค่าความถี่ (f) ของคลื่นอัลตราโซนิค ซึ่งการเปรียบเทียบความยาวคลื่นที่ความถี่และความเร็วของคลื่นที่มีค่าต่างกันซึ่งแสดงในตารางที่ 3.1

Frequency	Wavelength (λ)		
	(For C = 1000 m/s)	(For C = 3000 m/s)	For electromagnetic radiation
20 kHz = 2 x 10^4 Hz	5 cm	15 cm.	1.5 x 10^4 m
100 kHz = 10^5 Hz	1 cm	3 cm.	3 x 10^3 m
1 MHz = 10^6 Hz	1x10^{-3} cm	3x10^{-3} cm	300 m
50 MHz = 5 x 10^7 Hz	20 μm	60 μm	6 m
1 GHz = 10^9 Hz	1 μm	3 μm	30 cm.

ตารางที่ 3.1 แสดงการเปรียบเทียบความยาวคลื่นที่ความถี่และความเร็วของคลื่นที่มีค่าต่างกัน

จากตาราง แสดงค่าที่ c = 1000 m / s และ 3000 m / s ซึ่งอยู่ในตัวกลางที่เป็นของเหลว และเปรียบเทียบกับการแผ่รังสีของคลื่นแม่เหล็กไฟฟ้าที่ความถี่เดียวกัน จะเห็นว่ามีความแตกต่างกันมากโดยความเร็วของคลื่นอัลตราโซนิค จะมีค่าต่ำว่าความเร็วในการแผ่รังสีของคลื่นแม่เหล็กไฟฟ้ามาก (ในสูญญากาศมีค่าความเร็วประมาณ 3 x 10^8 m / s) จากหลักการนี้เป็นประโยชน์มากในตัวอย่างของระบบโทรทัศน์และระบบดิจิตอลคอมพิวเตอร์ที่ใช้หน่วง (Delay) คลื่นแม่เหล็กไฟฟ้าการเปลี่ยนคลื่นแม่เหล็กไฟฟ้านี้ไปเป็นคลื่นอัลตราโซนิค โดยใช้ทรานสดิวเซอร์ (Transducer) หลักการคือ คลื่นแม่เหล็กไฟฟ้าใน 1 ms สามารถเดินทางได้ระยะทาง 3 x 10^8 x 10^{-3} = 300 km ซึ่งมีค่ามาก เราจึงสามารถหน่วงสัญญาณนี้ได้ โดยเปลี่ยนคลื่นแม่เหล็กไฟฟ้าเป็นคลื่นอัลตราโซนิค โดยใช้อุปกรณ์ "Delay Line" ซึ่งแสดงในรูป 3.8 ซึ่งประกอบด้วยตัวส่งและตัวรับคลื่นอัลตราโซนิคโดยคลื่นจะเคลื่อนที่ใน Delay Line ในระยะทาง d ซึ่งในระยะ d นี้อาจจะเป็นตัวกลางที่เป็นก๊าซ, ของเหลว, หรือของแข็งก็ได้ ซึ่งจะมีความเร็วที่แตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Material	Temperature (^{0}C)	Density (Kg/m^3)	Sound Velocity (m/s)	Characteristic Impedance (ns/m^3)
Gases				
Argon	0	1.783	319	569
Helium	0	0.178	965	172
Oxygen	0	1.429	316	452
Hydrogen	0	0.090	1284	116
Carbon dioxide	0	1.977	259	512
Air	0	1.293	331.45	429
Liquids				(10^6 Ns/m^3)
Water	20	998	1483	1.48
Mercury	20	13500	1451	19.6
Glycerine	20	1228	1895	2.33
Ethyl alcohol	20	790	1159	0.92
Methyl acohol	20	792	1120	0.89
Carbon disulphide	20	1264	1157	1.46

ตารางที่ 3.2 แสดงค่าความเร็วและคุณสมบัติทางด้านอิมพีแดนซ์ของคลื่นอัลตราโซนิคในตัวกลางที่เป็นแก๊สและของเหลว

3.4 ความแรงและกำลังของคลื่นอัลตราซนิค

เมื่อทรานสดิวเซอร์ของอัลตราโซนิคปล่อยคลื่นอัลตราโซนิคออกไป จะทำให้อนุภาพของตัวกลางมีการสั่นสะเทือน อนุภาคนั้น ๆ จึงมีพลังงานที่เนื่องมาจากการสั่นสะเทือนอัตราอัตราการไหลของพลังงานจึงเรียกได้ว่าเป็นความแรงของลำอัลตราโซนิค อาจกล่าวให้ชัดเจนไปว่า ความแรงอัลตราโซนิคที่จุด ๆ หนึ่งเป็นอัตราการไหลของพลังงานผ่านพื้นที่ 1 ตารางเซนติเมตร ซึ่งวางอยู่ตั้งฉากกับบลำอัลตราโซนิคที่จุดนั้น เมื่อความแรงของอัลตราโซนิคเพิ่มมากขึ้นจะมีการเพิ่มใน Pressure amplitude, Particle- displacement amplitude และ Particle – velocity amplitude

กำลังของอัลตราโซนิค เป็นอัตราการไหลของพลังงานที่ผ่านพื้นที่หน้าตัดทั้งหมดของลำอัลตราโซนิค จะได้ความสัมพันธ์ดังนี้

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ultrasonic poiwer	=	Ultrasonic intensity	x	Cross-sections	(2-1)
(W)		(W / m^2)			

ในการใช้งานคลื่นอัลตราโซนิคนั้น เนื่องจากเป็นคลื่นที่มีทิศทาง ทำให้เราสามารถส่งคลื่นไปยังจุดเป้าหมายที่ต้อการได้อย่างเจาะจง ยิ่งถ้าเกิดคลื่นมีความถี่สูงความยาวคลื่นจะยิ่งสั้นลง โดยถ้าความยาวของคลื่นมากกว่าช่องเปิดของทรานสดิวเซอร์ เช่น คลื่นความถี่ 300 Hz ในอากาศจะมีความยาวคลื่นประมาณ 1 เมตรเศษซึ่งยาวกว่าช่องเปิดของตัวทรานสดิวเซอร์ ทำให้คลื่นเสียงที่ออกจากตัวกำเนิดเสียงนี้เกิดการหักเหและเลี้ยงเบนที่ขอบด้านนอกของตัวกำเนิดเสียง ทำให้เกิดการกระจายของคลื่นเสียง (Scatter) ไม่มีทิศทางที่แน่นอน แต่ถ้าใช้คลื่นเสียงที่มีความถี่สูงขึ้นมาในย่านอัลตราโซนิค เช่นที่ความถี่ 40 kHz จะมีความยาวคลื่นในอากาศเพียง 8 mm. เท่านั้น ซึ่งจะเล็กกว่าช่องเปิดของตัวกำเนิดเสียงมากคลื่นเสียงที่ได้จะไม่เลี้ยวเบนที่ขอบจึงพุ่งเป็นลำแคบ ๆ (Beam) ซึ่งมีทิศทางที่แน่นอน

คลื่นอัลตราโซนิคนั้นเมื่อต้องเดินทางในอากาศแล้วนั้น ความถี่ที่ใช้งานมักจะถูกจำกัดอยู่เพียงไม่เกิน 50 kHz เพราะถ้าความถี่สูงขึ้นกว่านี้ อากาศจะเกิดดูดกลืนคลื่นเสียงเพิ่มมากขึ้น ทำให้ระดับความแรงของคลื่นอัลตราโซนิคที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานในด้านการแพทย์นั้นจะใช้รัศมีทำการสั้น ๆ จึงใช้ความถี่ในช่วง 1 MHz – 10 MHz ส่วนความถี่ที่เป็น GHz จะมีใช้งานในตัวกลางที่คลื่นเสียงเดินทางนั้นไม่ใช่อากาศแต่เป็นตัวกลางชนิดอื่น เช่น น้ำ เป็นต้น

จากการมีทิศทางของคลื่นอัลตราโซนิคทำให้เรานำไปใช้งานได้หลายอย่างเช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic Remote Control) เครื่องล้างอุปกรณ์ (Ultrasonic Cleaner) โดยให้น้ำสั่นที่ความถี่สูง เครื่องวัดความหนาของวัตถุ โดยสังเกตตำแหน่งอวัยวะบางส่วนในร่างกายใช้ทดสอบการรั่วของท่อ เป็นต้น โดยความถี่ที่ใช้ขึ้นกับการใช้งาน เช่น ถ้าคลื่นเสียงต้องเดินทางผ่านอากาศแล้ว ความถี่ที่ใช้ก็มักจำกัดอยู่เพียงไม่เกิน 50 kHz เพราะที่ความถี่สูงกว่านี้อากาศจะดูดกลืนเสียงเพิ่มขึ้นมาก ทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว

3.5 อัลตราโซนิคทรานสดิวเซอร์

อัลตราโซนิคทรานสดิวเซอร์ คือ อุปกรณ์ซึ่งสามารถแปลงพลังงานกลโดยการสั่นไปมา ซึ่งทำให้เกิดคลื่นอัลตราโซนิคกระจายไปในวัตถุตัวกลาง หรือจากพลังงานของคลื่นอัลตราโซนิคแปลงไปเป็นพลังงานกลซึ่งสามารถทำให้เกิดแรงดันไฟฟ้าขึ้นได้จากตัวทรานสดิวเซอร์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานสดิวเซอร์ที่ให้กำเนิดคลื่นอัลตราโซนิคมี 2 ชนิดใหญ่ ๆ คือ

1. เมคคานิคทรานสดิวเซอร์ (Mechanical Transducers)
2. อิเล็กโตรเมคคานิคัลทรานสดิวเซอร์ (Electromechanical Transducers)

อิเล็คโตรแมคคานิคเคิลทรานสดิวเซอร์ จะเปลี่ยนพลังงานไฟฟ้าไปเป็นพลังงานเสียงที่ใช้กันอยู่มี 4 ชนิดคือ

1. แบบมูฟวิ่งคอลย์ (Moving Coil)
2. แบบอีเล็คโตรสแตทริก (Electrostatic)
3. แบบเปียโซอิเล็คทริก (Piezoelectric)
4. แบบเฟอร์โรอิเล็คทริก (Ferroelectric)

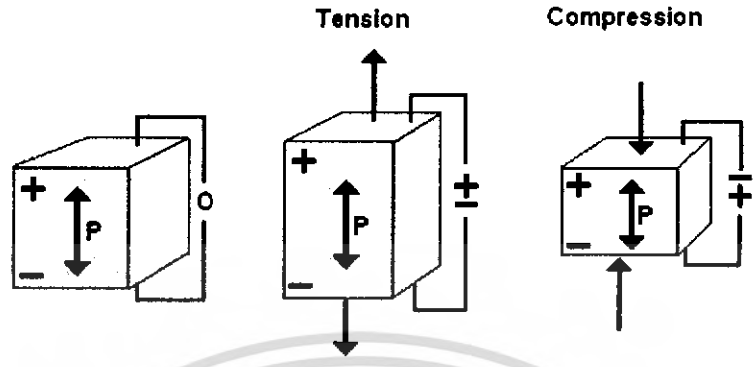
1. แบบมูฟวิ่งคอลย์ นั้นคล้ายกับมูฟวิ่งคอลย์ของลำโพง มีประโยชน์มากในการเป็นตัวสั่น (vibrator) ของทดสอบความเร่งของอุปกรณ์แบมูฟวิ่งคอยล์ ชนิดริบบอน (Ribbon) ทำด้วยอะลูมิเนียมบาง ๆ วางไว้ระหว่างขั้วทั้งสองของแม่เหล็ก มันทำตัวเสมือนเป็นขดลวดของมูฟวิ่งคอลย์ใช้ทำให้เกิดคลื่นอัลตราโซนิคในการศึกษาคุณสมบัติในการดูดคลื่นอัลตราโซนิคของก๊าซ

2. แบบอีเล็คโตรสแตทริก ใช้หลักการที่ว่า ในระหว่างแผ่นขนานของ Parallel Plate Capaciter เมื่อเราให้ความต่างศักย์ระหว่างแผ่นทั้งสองจะทำให้เกิดแรงขึ้น หลักการใหญ่ของมันจะใช้ทำเป็นลำโพงเสียงที่มีคุณภาพสูง แต่ก็ใช้ทำให้เกิดการสั่นแบบอัลตราโซนิคในของแข็งแบบที่ 1 และ 2 นั้นไม่ค่อยใช้กันเป็นแหล่งกำเนิดคลื่นอัลตราโซนิค ส่วนแบบที่ 3 และ 4 ใช้กันมากและแพร่หลาย ดังนั้นจะกล่าวละเอียดเฉพาะแบบที่ 3 และ 4 เท่านั้น

3. แบบเปียโซอิเล็คทริก และ

4. แบบเฟอร์โรอิเล็คทริก ในผลึกที่ประกอบด้วยโพล่าโมเลกุล (Polar Molecule) ซึ่งอยู่จุดศูนย์กลางของความสมมาตร (Center of symmbtry) ถ้าตัดเอาเพลท (Plate) อันเนื่องมาจากผลึกนี้แล้วทำให้เกิดการบิดงอ (Deform) ด้วยเครื่องมือกลจะทำให้เกิดความต่างศักย์ขึ้นระหว่างผิวหน้าทั้งสองเพลท ปรากฏการณ์นี้เราเรียกว่า *ไดเร็กเปียโซอิเล็กทริกเอฟเฟ็ค* (direct Piezoelectric Effect) ศักระหว่างผิวหน้าทั้งสองของเพลทเพลทนี้จะเกิดบิดเบี้ยวไปได้เอง เรียกปรากฏการณ์นี้ว่า *อินเวอร์สปีย โซอิเล็คทริคเอฟเฟ็ค* (Inverse Piezoelectric Effect) ตัวอย่างเช่น ผลึกควอทซ์ ดังรูปที่ 3.9 แสดงหนึ่งหน่วยเซลซึ่งอยู่ระหว่างเพลน (Plane) ที่ขนานกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 (ก) เซลที่ยังไม่ถูกแรงใด ๆ (ข) เซลถูกอัด (ค) เซลถูกยืด

รูปที่ 3.7 (ก) จะอยู่ในสภาวะปกติจุดศูนย์ถ่วงของประจุบวกของซิลิคอนกับประจุลบของออกซิเจนอยู่ที่เดียวกันและจำนวนประจุไฟฟ้าทั้งสองเท่ากันจะไม่เกิดไดโพลโมเมนต์ขึ้น ดังนั้นผลเนื่องจากการกระจายของประจุตกคร่อมระนาบทั้งสองที่ขนานกันจึงเป็นศูนย์

รูปที่ 3.7 (ข) ถ้าทำทำให้มันเปลี่ยนรูปร่างไปจากเดิมโดยการจัดให้หดหรือดึงได้ดังตามรูปที่ 3.7 (ค) จุดศูนย์ถ่วงของประจุทั้งสองจะไม่อยู่ที่เดียวกัน ทำให้เกิดการกระจายของประจุขึ้นระหว่างระนาบทั้งสอง

การทำให้มันเปลี่ยนรูปร่างด้วยการอัดหรือการหดตามรูปนั้น จะเป็นการแยกจุดศูนย์ถ่วงผลจะทำให้เกิดประจุปรากฏบนระนาบทั้งสอง ในรูปแบบนี้เรียกว่า *ไดเร็กเปียโซอิเล็กทริกเอฟเฟ็ค*

สำหรับแบบ *อินเวอร์สเปียโซอิเล็กทริกเอฟเฟ็ค* จะเกิดโดยให้สนามไฟฟ้าแก่ระหว่างเพลนทั้งสองทำให้จุดศูนย์ถ่วงของทั้งสองประจุเคลื่อนออกจากกันในทิศทางตรงกันข้าม ทำให้เซลบิดเบี้ยวไปถ้ามีแผ่นควอทซ์ซึ่งตัดมาจากผลึกควอทซ์ ในทิศทางที่จะทำให้เกิดสั่นสะเทือนขนานกับสนามไฟฟ้าที่ใช้เมื่อเราให้สนามไฟฟ้าสลับ เพลนนี้จะขยายและหดตัวด้วยความถี่เดียวกับสนามไฟฟ้าซึ่งจะเหมือนกับลูกสูบ ดังนั้นมันจะเป็นตัวกำเนิดคลื่นเสียงได้

ปัจจุบันมีสารไม่กี่ชนิดที่จะให้ปรากฏการณ์เปียโซอิเล็คทริก สารไดอิเล็คทริกทุกชนิดเป็นสารอิเล็กโทรสแตทริก แต่ปกติแล้วผล (Effect) นี้จะน้อยมาก แต่สารที่ให้ผลอย่างมากได้แก่ สารไททาเนท (Titanates) ซีคอเนท (Ziecoates) และในกรณีเช่นนี้เราเรียกว่า มันเป็นสารเฟอร์โรอิเล็คทริก ปกติสารเหล่านี้จะอยู่ในรูปของส่วนผสมเซอร์รามิค ซึ่งประกอบไปด้วยโดเมน (Domain) ที่มีขั้วอิสระรวมตัวกันมาก ๆ ถ้าเรานำสารแบบนี้มาสักหนึ่งชนิด เอาไปเผาให้สูงกว่าอุณหภูมิเตอริก (Curic Temperature) ของมันแล้วทำให้เย็นภายในสนามไฟฟ้าสูง ๆ ทิศทางของขั้วโดเมนต่าง ๆ จะวางตัวไปแนวเดียวกับสนาม และจะยังคงอยู่ต่อไปเช่นนั้น แม้ว่าจะเอาสนามไฟฟ้าออกไปนั่นแสดงว่ามันแสดงคุณสมบัติ เช่น เพอร์มาเนนท์อิเล็คทริก บางทีมันจะเป็นสารเปียโซอิเล็คทริก ถ้าหากว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันประพฤติตัวไปตามสมการของสารเปียโซอิเล็คทริกตามธรรมชาติแล้ว เราจะใช้สมการเหล่านี้อธิบายคุณสมบัติของสารเปียโซอิเล็คทริกแท้ ๆ

3.5.1 คุณสมบัติทั่วไปของทรานสดิวเซอร์

ทรานสดิวเซอร์ที่ใช้ในทางอัลตราโซนิคจะมีคุณสมบัติพิเศษคือ มีเปียโซอิเล็คทริกเอฟเฟ็ค และที่ใช้กันมากคือผลึกแร่ควอทซ์ (Quartz), แบเรียมติเตเนท (Barium titanate) และลีดเซอโคเนท (Lead circonate titanate) สารเหล่านี้เมื่อนำมาทำเป็นทรานสดิวเซอร์ ยังแสดงคุณสมบัติเป็นคาปาซิเตอร์ทางไฟฟ้าอีกด้วย ค่าคาปาซิแตนซ์เป็นไมโครฟารัดจะได้สัดส่วนกับค่าที่ไดเลคติด (dielectric constant, K) ของวัสดุคูณด้วยพื้นที่และหารด้วยความหนา โครงสร้างของทรานสดิวเซอร์มีความคล้ายคาปาซิเตอร์ที่ว่าอีเล็คโตรดที่อยู่ 2 ข้างถูกแยกโดยช่องที่มีสารเซรามิค ตารางที่ 3.3 แสดงคุณสมบัติบางประการของสารเซรามิคได้รับความร้อน จะทำให้ไดโพล (dipole) เคลื่อนไหวได้โดยเสรีและที่จุดนี้เมื่อได้รับสักไฟฟ้าเข้าไป ก็จะมีการจัดเรียงตัวและมีคุณสมบัติเปียโซอิเล็คทริค ถ้าภายหลังเซรามิคที่มีคุณสมบัตินี้ถูกเผาให้ร้อนถึงอุณหภูมิดังกล่าวอีก คุณสมบัติเปียโซอิเล็คทริกจะหมดไป ดังนั้นความร้อนจะสามารถทำลายทรานสดิวเซอร์ได้ โดยเฉพาะถ้าอุณหภูมิสูงเกินเคอริก

ค่าคงที่ไดเลคติดเป็นค่าทรานสดิวเซอร์ต้องการพลังงานไฟฟ้าเท่าใด จึงจะเปลี่ยนเป็นพลังงานกลได้ทั้ง แบเรียมติตาเนท และลีดเซอโคเนท มีประสิทธิภาพมากกว่าผลึกแร่ควอทซ์ 5-6 เท่า เนื่องจากค่าคงที่ไดเลคติดต่ำมากและมีค่าคงที่ (Coupling constant) ต่ำมากจึงไม่ค่อยได้นำควอทซ์มาใช้ในเครื่องอัลตราโซนิคในปัจจุบัน

3.5.2 คุณสมบัติทางความถี่ของทรานสดิวเซอร์

ความถี่ของทรานสดิวเซอร์ขึ้นอยู่กับคุณสมบัติทางเชิงกลและลักษณะรูปร่างตัวอย่างเช่น เซรามิคแท่งเหลี่ยมแบนจะมีความถี่เรโซแนนซ์ได้ 3 อย่าง คือ มีการยืดหดตามความหนาและตามรัศมี อย่างไรก็ดีเมื่อป้อนศักไฟฟ้าเข้าไปสามารถทำให้ทรานสดิวเซอร์ยืดหดในแนวไหนก็ได้ ซึ่งขึ้นอยู่กับขั้วไฟฟ้าที่ใช้บริษัทที่ทำทรานสดิวเซอร์จะกำหนดความถี่คงที่ สำหรับแต่ละแบบของการยืดหด

$$\text{Resonant frequency} = \frac{\text{constant}}{\text{Thickness (inch)}} \qquad (3\text{-}1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Material	Curic Tempera Ture (0 C)	Q	Dielectric Constant (K)	Coupling Coefficient (KT)
Quartz	550	Very High	4.5	0.09
Barium titanate	100	400 – 700	1700	0.38
Lead zirconate titanate	300	90 – 500	1450	0.51

ตารางที่ 3.3 แสดงคุณสมบัติบางอย่างของสารเซรามิค

3.5.3 เปียโซทรานสดิวเซอร์

โครงสร้างภายในจะประกอบด้วย "ชิ้นสารเปียโซอิเล็คทริค" (Piezoelectric element) ซึ่งสามารถทำได้จากสารหลายชนิด เช่น ผลึกแร่ควอทซ์ (quartz), แบเรียมติตาเนท (barium titanate), เกลือร็อกเชล (Rochelle salt) ฯลฯ แต่นิยมใช้กันแพร่หลายคือชิ้นสารเปียโซอิเล็คทริคที่ทำมาจากแบเรียมติตาเนท ซึ่งได้จากการผสมสารติตาเนียมแบเรียมคาบอเนท ที่อุณหภูมิประมาณ 1300-1400 C^0 และผลิตออกมาในรูปแบบของชิ้นสาร "เซรามิค" ซึ่งจะเป็นรูปสี่เหลี่ยมโดยส่วนใหญ่จากชิ้นเซรามิคนี้จะถูกประกบด้วยแผ่นเพลท 2 แผ่นซึ่งส่วนใหญ่ทำมาจากแผ่นเงิน เพื่อต่อเป็นขาของอิเล็คโทรดต่อไปนั่นเอง ด้านข้างหนึ่งของแผ่นเพลทจะถูกยึดเพื่อให้ไม่เกิดการสั่น ส่วนอีกด้านจะยึดกับแผ่นไดอะเฟรม ซึ่งสามารถเคลื่อนไหวไปมาได้โดยแผ่นไดอะเฟรมนี้จะถูกสร้างให้มีลักษณะคล้ายเลนซ์ เพื่อรวมคลื่นให้มีความเร็วสูงขึ้น ซึ่งจะกล่าวถึงลักษณะโครงสร้างต่อไป

3.5.4 เปียโซอิเล็คทริกเอฟเฟ็ค

ปรากฏการณ์แบบเปียโซอิเล็คทริคสามารถอธิบายได้คือ เมื่อสภาวะปกติยังไม่มีความดันหรือคลื่นมาตกกระทบกับชิ้นสารเซรามิค ประจุไฟฟ้าซึ่งอยู่ภายในชิ้นสารจะเกิดการสมดุล ทำให้แผ่นเพลทเงินที่ประกบอยู่มีศักไฟฟ้าที่เท่ากัน ซึ่งผลรวมของประจุไฟฟ้าบนแผ่นเพลทมีค่าเท่ากับศูนย์ แต่เมื่อมีความดันหรือคลื่นเสียงมากระทบชิ้นสารเซรามิค จะทำให้เกิดการเปลี่ยนแปลงทางกายภาพของชิ้นสารคือมีการโก่งงอไปมาตามลักษณะคลื่นที่ตกกระทบ ซึ่งในกรณีนี้จะทำให้เกิดการจัดเรียงประจุคือ เกิดมีประจุตรงข้ามขึ้นบนแผ่นเพลทที่ประกบชิ้นสารเซรามิคอยู่ ซึ่งผลของประจุไฟฟ้าตรงข้ามนี้ทำให้เกิดเป็นแรงดันหรือคลื่นที่มาตกกระทบชิ้นสารเซรามิคที่โก่งงอไปมา

ทรานสดิวเซอร์แบบเปียโซอิเล็คทริค สามารถที่จะเปลี่ยนแปลงไปมาระหว่างพลังงานกลกับพลังงานไฟฟ้า ในกรณีการเปลี่ยนจากพลังงานไฟฟ้าเป็นพลังงานกลคือส่วนตัวส่งคลื่นอัลตราโซนิคจะทำงาน เมื่อเราป้อนแรงดันไฟฟ้ากระแสสลับซึ่งอาจมีความถี่ในช่วง 20 – 100 kHz แรงดันที่ป้อนนี้จะทำให้ชิ้นสารเซรามิคเกิดการเปลี่ยนแปลงรูปร่าง โดยที่ด้านหนึ่งของชิ้นสารเซรามิคจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กดกับแผ่นไดอะแฟรม เมื่อชิ้นสารเซรามิคเกิดการบิดตัวไปมาจะทำให้แผ่นไดอะแฟรมนี้อัดกับอากาศโดยรอบเกิดเป็นคลื่นความถี่กับสัญญาณที่ป้อนเข้ามา ซึ่งคลื่นที่ผลิตได้จะมีกำลังเอาท์พุตสูงสุดเมื่อป้อนความถี่ที่ป้อนเข้ามาตรงกับความถี่รีโซแนนซ์ ซึ่งเป็นความถี่ทางกลตามธรรมชาติของชิ้นสารเซรามิคชิ้นนั้น จะทำให้ความถี่อื่น ๆ นอกเหนือจากนี้จะมีค่าลดลงกว่าความถี่รีโวแนนซ์มาก

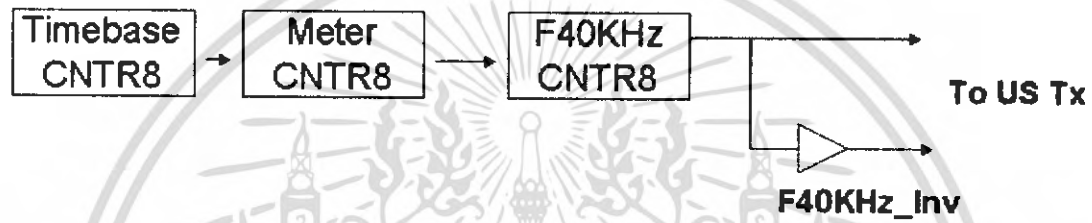
ในทางกลับกันการเปลี่ยนจากพลังงานกลเป็นพลังงานไฟฟ้านี้คือ ตัวรับคลื่นอัลตราโซนิค เราสามารถที่จะมองได้ว่าเป็นแหล่งจ่ายหนึ่ง ๆ เพราะมันไม่ต้องการแหล่งจ่ายอื่นๆ มาป้อนก็สามารถสร้างแรงดันไม่สูงมากนัก โดยการทำงานของตัวรับคลื่นนี้เมื่อคลื่นซึ่งมีความดันมาตกกระทบบนแผ่นไดอะแฟรม ซึ่งค่าความถี่ของคลื่นทำให้ชิ้นสารเซรามิคถูกบีบไปมา ซึ่งจะทำให้เกิดปรากฏการณ์เปียโซอิเล็คทริคบนชิ้นสารทำให้สามารถสร้างแรงดันไฟฟ้ากระแสสลับที่มีความถี่เดียวกับคลื่นที่มาตกกระทบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

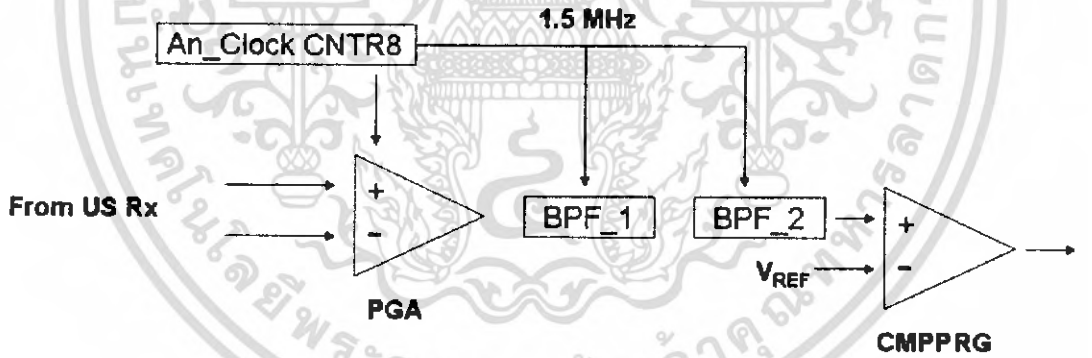
บทที่ 4

ดิจิตอล และ อนาลอกบล็อกต่างๆ ที่นำมาใช้งาน

ในปริญญานิพนธ์นี้ จะมีการกล่าวถึงการออกแบบ เครื่องวัดระยะทางอัลตราโซนิค โดยใช้ไมโครคอลโทรลเลอร์แบบ PSoC (Programmable System On Chip) ซึ่งในบทนี้จะกล่าวถึงทฤษฎีต่างที่เกี่ยวของกับบล็อกที่ได้นำมาใช้ในการออกแบบโครงงานนี้



รูปที่ 4.1 แสดงโครงสร้างบล็อกต่างๆที่ใช้ในการสร้างภาคส่ง



รูปที่ 4.2 แสดงโครงสร้างบล็อกต่างๆที่ใช้ในการสร้างภาครับ

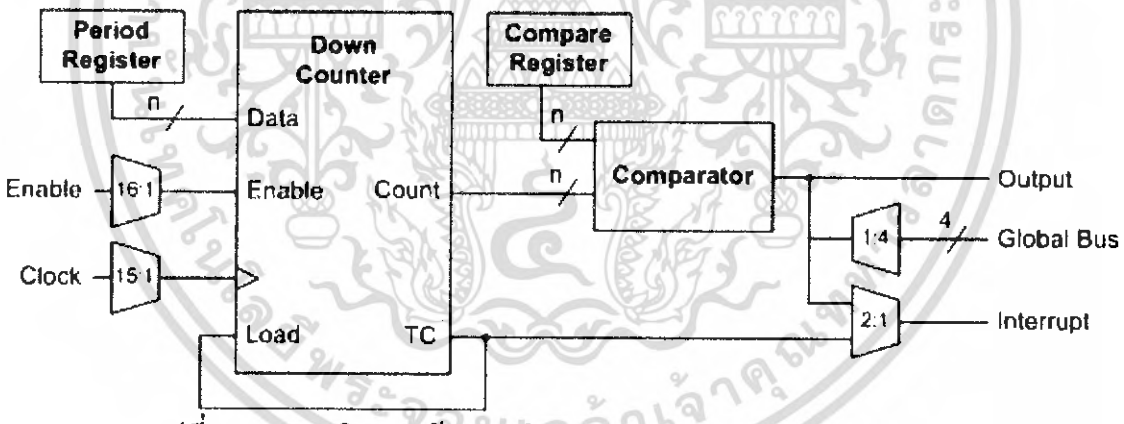
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างของภาคส่ง และ ภาครับ คลื่นอัลตราโซนิค ในรูปที่ 3.1 และ 3.2 จะเห็นว่าจะต้องมีการนำบล็อกต่างๆทั้ง ดิจิตอล และ อนาลอก บล็อก มาใช้เพื่อสร้างเป็นวงจรขึ้นมา ซึ่งบล็กต่างๆที่ต้องนำมาใช้งานได้แก่

1. 8-Bit Counter User Module
2. Programmable Gain Amplifier (PGA) User Module
3. Digital Inverter User Module
4. Bandpass Filter User Module
5. Programmable Threshold Comparator User Module

4.1 8-Bit Counter User Module

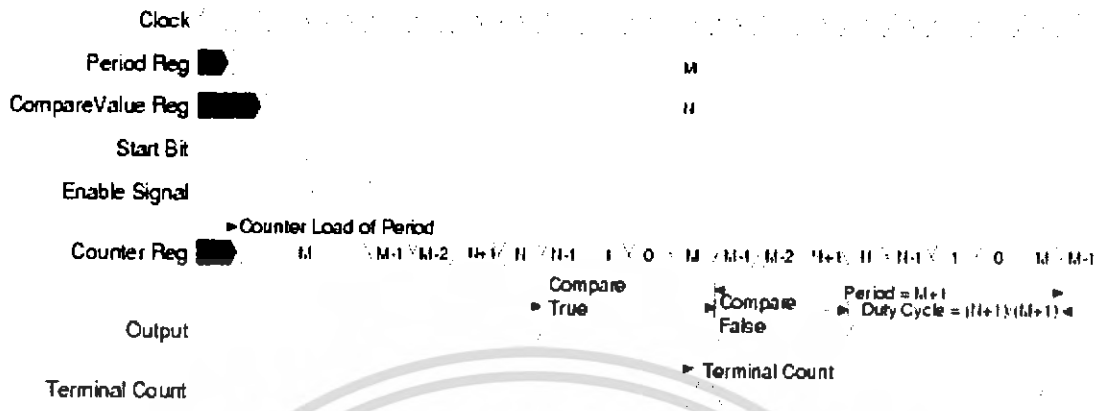
Counter User Module ของ PSoC Microcontroller มีขนาดตั้งแต่ 8 ,16 , 32 , 64 บิตให้เลือกใช้ตามความเหมาะสม Counter User Module มีลักษณะคล้ายกันกับ Timer User Module มาก ทั้งรูปแบบการทำงานและ API ฟังก์ชัน ต่างกันตรงที่ Timer User Module จะมี Capture Mode ในขณะที่ Counter User Module ไม่มี แต่จะมี Enable แทน



รูปที่ 4.3 แสดงโครงสร้างภายในของ 8-Bit Counter User Module

Counter User Module ใช้ทรัพยากรเป็นดิจิตอลบล็อกจำนวน 1 , 2 , 3 , 4 ตามขนาดของ Counter 8 Counter 16 และ Counter 32 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงการทำงานของ 8-Bit Counter User Module

Counter Module ทำงานแบบนับลง (Count Down) สามารถเลือก clock ได้ทั้งสัญญาณภายในและสัญญาณภายนอก กำหนดค่า Period และ Capture ได้จาก Hardware (ใน Device Manager) กับ Terminal Count Out

พารามิเตอร์ของ Counter User Module

Clock

Clock ของ Counter สามารถเลือกได้จาก VC1 , VC2 , VC3 , Sysclk , Sysclk*2 และอีกหลายๆแหล่ง

Period

ค่า Period รีจิสเตอร์ โดยในการทำงานของ Timer จะโหลดค่า Period รีจิสเตอร์ให้กับ Counter รีจิสเตอร์ อัตโนมัติเมื่อ Counter รีจิสเตอร์ = 0 (Terminal Count) หรือมีการ Enable Timers ขณะ Disable อยู่

Enable

Hardware Enable สามารถเลือกได้จาก 16 แหล่งสัญญาณ ถ้ากำหนดให้เป็น High Counter จะทำงานแบบนับต่อเนื่อง

CompareValue

ค่า Compare รีจิสเตอร์ใช้งานเมื่อกำหนดการทำงานแบบ Compare Event

CompareType

ลักษณะการเปรียบเทียบระหว่าง Counter Register กับ Compare Register

InterruptType

เลือกแหล่งที่มาของ interrupt สามารถกำหนดให้เป็น Terminal Count หรือ Compare Event

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ClockSync

ClockSync เป็นพารามิเตอร์ ที่ต้องกำหนดให้กับ User Module ที่มีการใช้ Digital Block การเลือก ClockSync ขึ้นอยู่กับ Clock ที่เลือกใช้ตามตารางที่ 4.1

ClockSync	Clock
Sync to Sysclk	ใช้ในกรณีที่clockมีแหล่งกำเนิดมาจาก Sysclk เช่น VC1 , VC2 , VC3 (เมื่อ VC3 หารมาจาก Sysclk) , 32 KHz และจาก Digital PSoC Block ที่ใช้ Sysclk
Sync to Sysclk*2	ใช้ในกรณีที่clockมีแหล่งกำเนิดมาจาก Sysclk*2ยกเว้นใช้ Sysclk*2 โดยตรง
Use Sysclk Direct	ใช้ในกรณีที่clockมีแหล่งกำเนิดมาจาก Sysclk โดยตรง
Unsynchronize	ใช้ในกรณีที่clockมีแหล่งกำเนิดมาจาก Sysclk*2 โดยตรง และใช้เมื่อไม่ต้องการ Synchronize สัญญาณ

ตารางที่ 4.1 แสดงการเลือกใช้ ClockSync

API ฟังก์ชัน ของ Counter User Module

Counter8_Start

ฟังก์ชันเริ่มให้ Counter User Module ทำงาน

Counter8_Stop

ฟังก์ชันหยุดการทำงานของ Counter User Module

Counter8_EnableInt

สำหรับเปิดการใช้งาน Interrupt ของ Counter User Module

Counter8_DiableInt

สำหรับปิดการใช้งาน Interrupt ของ Counter User Moduler

Counter8_WritePeriod

ใช้กำหนดค่า Period ให้กับ Counter User Module

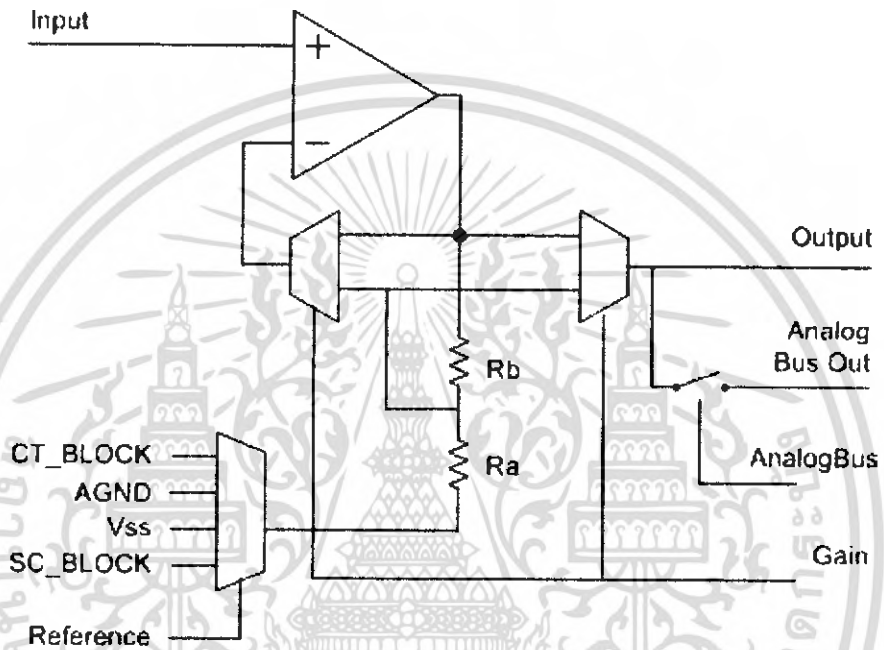
Counter8_WriteCompareValue

ใช้กำหนดค่า Compare Value ให้กับ Counter User Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Programmable Gain Amplifier (PGA) User Module

PGA User Module เป็นวงจรขยายแบบ Non-Inverting สามารถกำหนดค่า Gain ได้ทั้งจาก Device Editor และจาก API ฟังก์ชัน มี Input Impedance สูง และสามารถกำหนดแรงดัน Reference ได้อย่างอิสระ



รูปที่ 4.5 แสดงโครงสร้างของ PGA Block

พารามิเตอร์ของ PGA Module

Input

Input ของ PGA กำหนดได้ใน Device Editor โดยสามารถกำหนดให้เป็น analog column input multiplexer จาก PSoC Block อื่นๆ

Reference

Reference Input สามารถกำหนดให้เป็น AGND และ PSoC Block ใกล้เคียง

AnalogBus

เป็นทางออกของสัญญาณเอาพุต อีกทางหนึ่งสามารถกำหนดให้เชื่อมต่อกับ AnalogBus ใกล้เคียงกับบล็อกที่วางโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API ฟังก์ชัน ของ PGA Module

PGA_Start

ใช้กำหนดค่าเริ่มต้นต่างๆที่จำเป็นสำหรับ PGA Module และทำหน้าที่กำหนดค่า Power Setting ตามตารางที่ 4.2

Symbolic Name	Value
PGA_OFF	0
PGA_LOWPOWER	1
PGA_MEDPOWER	2
PGA_HIGHPOWER	3

ตารางที่ 4.2 แสดงค่าValueที่ใช้ในการตั้งค่าระดับพลังงานของ PGA

PGA_SetGain

ฟังก์ชันที่ใช้กำหนดค่า Gain ให้กับ PGA Module โดยค่าที่ต้องส่งให้กับฟังก์ชันที่ค่า Gain ต่างๆนั้นสามารถดูได้จากตารางที่ 4.3

Symbolic Name	Value	Symbolic Name	Value
PGA_G48_0	0Ch	PGA_G1_00	F8h
PGA_G24_0	1Ch	PGA_G0_93	E0h
PGA_G16_0	O8h	PGA_G0_87	D0h
PGA_G8_00	18h	PGA_G0_81	C0h
PGA_G5_33	28h	PGA_G0_75	B0h
PGA_G4_00	38h	PGA_G0_68	A0h
PGA_G3_20	48h	PGA_G0_62	90h
PGA_G2_67	58h	PGA_G0_56	80h
PGA_G2_27	68h	PGA_G0_50	70h
PGA_G2_00	78h	PGA_G0_43	60h
PGA_G1_78	88h	PGA_G0_37	50h
PGA_G1_60	98h	PGA_G0_31	40h
PGA_G1_46	A8h	PGA_G0_25	30h
PGA_G1_33	B8h	PGA_G0_18	20h
PGA_G1_23	C8h	PGA_G0_12	10h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PGA_G1_14	D8h	PGA_G0_06	00h
PGA_G1_06	E8h		

ตารางที่ 4.3 แสดงการตั้ง Gain ให้กับ PGA

PGA_Stop

หยุดการทำงานของ PGA Module

4.3 Digital Inverter User Module

Digital Inverter User Module เป็นดิจิตอลบล็อกซึ่งมี API ฟังชัน เพื่อเริ่มและหยุดการทำงานของ Digital Inverter User Module และมี API ฟังก์ชันสำหรับการเปิดและปิดการอินเตอร์รัพท์อีกด้วย

รูปที่ 4.6 แสดงDigital Inverter User Module

พารามิเตอร์ของ Digital Inverter User Module

Input

สามารถเลือกแหล่งสัญญาณอินพุตได้ 16 แหล่งสัญญาณ รวมไปถึงสัญญาณนาฬิกา 48 MHz , 24V1 , 24V2 ซึ่งถูกหารความถี่มาจาก Sysclk ด้วย

Output

สัญญาณเอาพุตสามารถออกผ่าน Global Output ทั้งสี่ได้

ClockSync

ในกรณีที่เอาพุตจาก Digital Inverter User Module ส่งผ่านไปยังบล็อกตัวอื่นๆเพื่อเป็นสัญญาณอินพุตแล้ว ควรจะทำการ Synchronize กับสัญญาณนาฬิกาของระบบ

Interrupt General Control

สำหรับการ Interrupt API และ IntDispatchMode สามารถเข้าถึงได้โดยการ Enable การอินเตอร์รัพท์ผ่านซอร์ฟแวร์ PSoC Designer

InterruptAPI

การเปิดใช้การอินเตอร์รัพท์สำหรับ Digital Inverter User Module แล้วสามารถทำได้โดยการ เลือก Enable การทำงานของอินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API ฟังก์ชันของ Digital Inverter User Module

DigInv_EnableInt

เพื่อเปิดการทำงานของการอินเตอร์รัพท์

DigInv_DisableInt

เพื่อปิดการทำงานของการอินเตอร์รัพท์

DigInv_Start

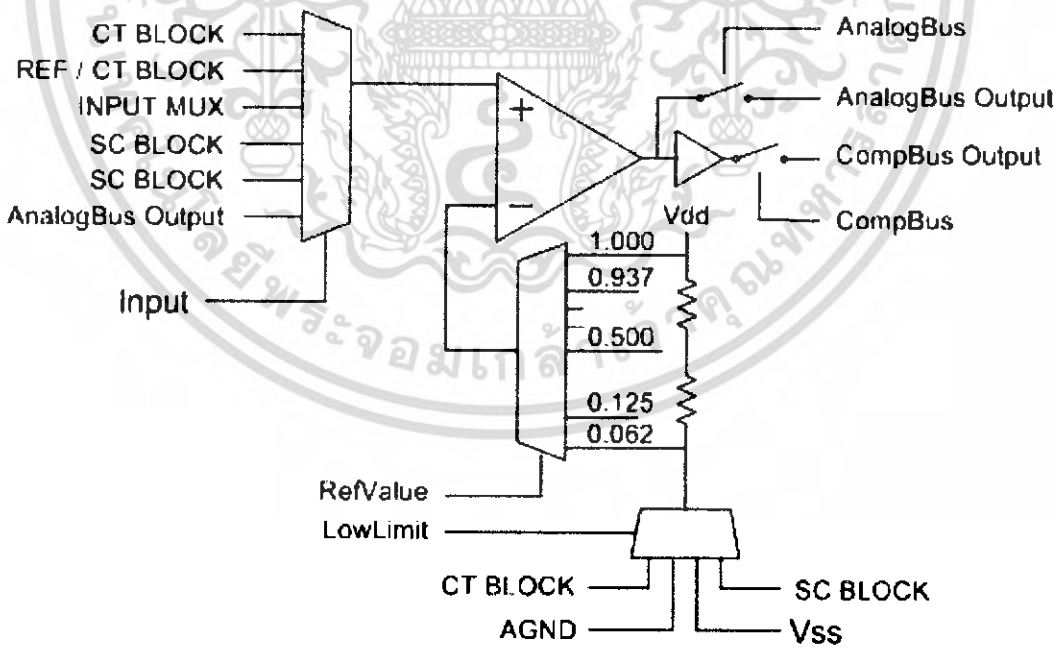
เพื่อเริ่มการทำงานของ Digital Inverter User Module

DigInv_Stop

เพื่อหยุดการทำงานของ Digital Inverter User Module สัญญาณเอาพุตจะเป็น LOW

4.5 Programmable Threshold Comparator User Module

สามารถใช้ในการเปรียบเทียบสัญญาณอินพุตกับสัญญาณอ้างอิงที่ตั้งไว้ได้ และสามารถเลือกแหล่งสัญญาณอินพุตได้ถึง 6 แหล่งสัญญาณ



รูปที่ 4.7 แสดงถึงโครงสร้างของ Programmable Threshold Comparator User Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ของ Programmable Threshold Comparator User Module

Input

สัญญาณอินพุตสามารถเลือกได้จาก 6 แหล่งสัญญาณโดยอาศัย Input Multiplexer

LowLimit

สามารถเลือกได้จาก AGND หรือ ขา Vss ซึ่งเป็นขาหนึ่งของไอซี PSoC Microcontroller

RefValue

ขนาดของสัญญาณ Reference สามารถโปรแกรมได้ตั้งแต่ 6.25 % ของแรงดันระหว่าง LowLimit กับ ขนาดของสัญญาณ Vdd

ComBus

สัญญาณเอาพุตสามารถส่งต่อไปเป็นอินพุตให้กับบล็อกต่างๆภายใน PSoC ได้โดยผ่านพารามิเตอร์ตัวนี้

AnalogBus

สัญญาณเอาพุตจากคอมพาราเตอร์ภายในสามารถส่งผ่านออกทางขาของ PSoC MCU ได้โดยผ่านทางพารามิเตอร์ตัวนี้

API ฟังก์ชันของ Programmable Threshold Comparator User Module

CMPPRG_Start

ใช้สำหรับเริ่มการทำงานของ Programmable Threshold Comparator User Module และเซตค่าระดับพลังงานให้กับ Programmable Threshold Comparator User Module

Symbolic Name	Value
CMPPRG_OFF	0
CMPPRG_LOWPOWER	1
CMPPRG_MEDPOWER	2
CMPPRG_HIGHPOWER	3

ตารางที่ 4.4 แสดงการตั้งค่าระดับพลังงาน

CMPPRG_Setpower

ใช้สำหรับการตั้งค่าระดับพลังงานให้กับ Programmable Threshold Comparator User Module โดยใช้ค่าระดับพลังงานตามตารางที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CMPPRG_SetRef

ใช้สำหรับตั้งค่าระดับแรงดันอ้างอิงให้กับคอมพาราเตอร์ สามาเซตค่าระดับแรงดันได้จาก ตารางที่ 4.5 โดย VThreshold = LowLimit + (Vdd – LowLimit)*RefValue

Symbolic Name	Value	Symbolic Name	Value
CMPPRG_REF1_000	F0h	CMPPRG_REF0_437	60h
CMPPRG_REF0_937	E0h	CMPPRG_REF0_375	50h
CMPPRG_REF0_875	D0h	CMPPRG_REF0_312	40h
CMPPRG_REF0_812	C0h	CMPPRG_REF0_250	30h
CMPPRG_REF0_750	B0h	CMPPRG_REF0_188	20h
CMPPRG_REF0_688	A0h	CMPPRG_REF0_125	10h
CMPPRG_REF0_625	90h	CMPPRG_REF0_062	00h
CMPPRG_REF0_562	80h	CMPPRG_REF0_042	14h
CMPPRG_REF0_500	70h	CMPPRG_REF0_021	04h

ตารางที่ 4.5 แสดงค่าที่ใช้ในการตั้งค่าแรงดันอ้างอิง

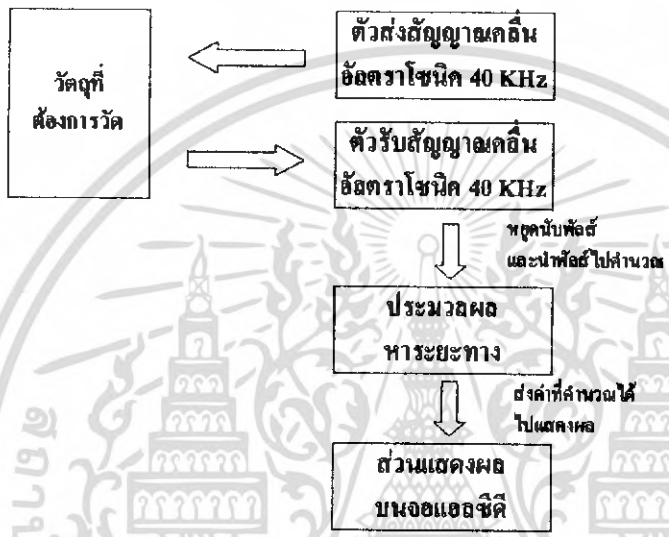
CMPPRG_Stop

หยุดการทำงานของ Programmable Threshold Comparator User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
หลักการทำงานและการออกแบบของวงจร

5.1 การทำงานโดยรวมของวงจร



รูปที่ 5.1 แสดงโครงสร้างโดยรวมของทั้งระบบ

คุณสมบัติของเครื่องวัดระยะทางด้วยอัลตราโซนิคในโครงงานนี้

ระยะทางต่ำสุดที่สามารถวัดระยะทางได้ = 35 cm.

ระยะทางสูงสุดที่สามารถวัดระยะทางได้ = 150 cm.

% ความผิดพลาดในการวัดระยะทาง = 0.1%

5.1 การออกแบบวงจรสามารถแบ่งได้เป็น 3 ส่วนใหญ่ๆก็คือ

5.1.1 วงจรภาคส่งคลื่นอัลตราโซนิค

วงจรในส่วนนี้จะทำการสร้างพัลส์ความถี่ 40 KHz ส่งออกไปยังตัวส่งสัญญาณอัลตราโซนิค ซึ่งประกอบด้วย Block

1. Counter8 ทำหน้าที่สร้าง สัญญาณพัลส์สี่เหลี่ยมความถี่ 40 KHz
2. DigInv ทำหน้าที่ในการสร้างสัญญาณพัลส์สี่เหลี่ยมความถี่ 40 KHz แต่เป็นสัญญาณในลักษณะกลับเฟสกับสัญญาณเอาท์พุทจาก Counter8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 วงจรภาครับสัญญาณอัลตราโซนิค

วงจรในส่วนนี้ จะทำหน้าที่ในการรับสัญญาณและแจ้งให้ภาคส่งทราบว่าได้รับสัญญาณจากภาคส่งแล้วเพื่อทำการหยุดนับ ซึ่งประกอบไปด้วย Block

1. PGA (Programmable Gain Amplifier) ทำหน้าที่ในการขยายสัญญาณอัลตราโซนิคที่รับเข้ามาจากอัลตราโซนิคทรานซ์ดิวซ์เซอร์
2. Band Pass Filter ทำหน้าที่ในการกรองสัญญาณที่ไม่ได้อยู่ในย่านความถี่ 40 KHz ออกไปจากสัญญาณที่นำเข้ามาประมวลผล เนื่องจากสัญญาณที่ได้รับเข้ามาจากทรานดิวซ์เซอร์นั้นมีความถี่อื่นๆปะปนมาด้วย
3. CMPPRG (Programmable Threshold Comparator) ส่วนนี้ทำหนี้ที่ในการแยกแยะความแตกต่างระหว่างสัญญาณที่เข้ามาว่าเป็นอินพุทจากรับคลื่นอัลตราโซนิคความถี่ 40 KHz หรือไม่ หากเป็นความถี่ 40 KHz และมีขนาดสัญญาณมากกว่าขนาดของแรงดันอ้างอิง Comparator จะส่งเอาท์พุตออกมาเป็นลอจิก 1 เพื่อให้ระบบทราบว่าสัญญาณที่ไดส่งมานั้นได้ถูกรับเรียบร้อยแล้ว
4. An_Clock เป็นส่วนที่ทำหน้าที่ในการสร้างสัญญาณพัลส์ความถี่ 1.5 MHz เพื่อป้อนให้กับ Block ต่างๆ ในส่วนของ Analog Block เพื่อใช้ในการควบคุม Switched Capacitor ที่เป็นโครงสร้างภายใน Block ของ Analog Block ชนิด switch
5. Refmux (Reference Multiplexer) ใช้สำหรับสร้างแรงดันอ้างอิงให้กับระบบ โดยการเลือกจาก Block ส่วนนี้เพื่อกำหนดให้เป็น Analog Ground ให้กับระบบในส่วนของภาครับสัญญาณอัลตราโซนิค

5.1.3 วงจรภาคนับระยะเวลาที่คลื่นอัลตราโซนิคใช้ในการเดินทาง

วงจรส่วนนี้จะทำการนับระยะเวลาที่คลื่นอัลตราโซนิคใช้ในการเดินทางเพื่อนคำนวณระยะทางจากความเร็วของคลื่นอัลตราโซนิค โดยประกอบด้วย Block ต่างๆ ดังนี้

1Timebase เป็น Block ที่นำมาใช้ในการสร้างฐานเวลาที่คลื่นอัลตราโซนิคใช้ในการดินทางไปได้ระยะทาง 1 เซนติเมตร โดยเมื่อถึงเวลาดังกล่าวเอาท์พุตจาก Timebase จะส่งออกไปเป็นสัญญาณนาฬิกา 1 ลูก ให้กับ Meter เพื่อแจ้งให้ระบบทราบว่า คลื่นอัลตรา

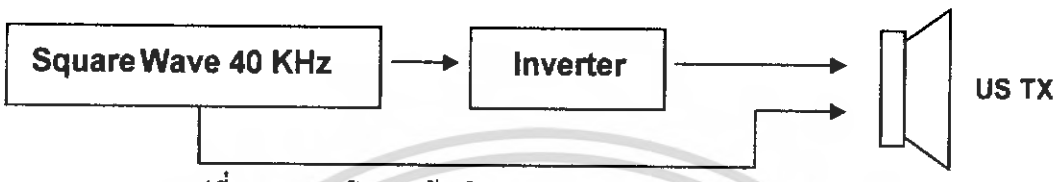
โซนิคที่ได้ส่งออกไปขณะนี้ได้เดินทางครบ 1 เซนติเมตรแล้ว

2. Meter เป็น Block ที่นำมาใช้ในการนับจำนวนสัญญาณนาฬิกาที่ส่งมาจาก Timebase เพื่อทำหน้าที่ในการนับว่าสัญญาณอัลตราโซนิคได้เคลื่อนที่ไปแล้วทั้งหมดเป็นระยะทางกี่เซนติเมตรแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การออกแบบส่วนต่างๆ ของวงจร

5.2.1 การออกแบบส่วนที่ใช้ในการส่งสัญญาณคลื่นอัลตราโซนิค



รูปที่ 5.2 แสดงโครงสร้างโดยรวมของ ภาคส่งสัญญาณอัลตราโซนิค

ในส่วนภาคส่งคลื่นอัลตราโซนิคนี้ เนื่องจากทรานดิวซ์เซอร์ ที่นำมาใช้งานในส่วนของภาคส่งนั้นทำงานที่ความถี่ 40 KHz ดังนั้นจึงต้องมีการสร้างคลื่นความถี่ 40 KHz ให้กับทรานดิวซ์เซอร์โดยการใช้ Counter 8 บิต นับสัญญาณนาฬิกา 2 MHz ซึ่งได้มาจากสัญญาณนาฬิกา 24 MHz ของระบบหารด้วย 12 แล้วทำการนับให้ได้เอาต์พุทออกมา 40 KHz ซึ่งก็คือการนำเอาความถี่ 2 MHz มาหาร 50 อีกครั้งหนึ่งซึ่งก็คือการกำหนดค่า

$$\text{Period Value} = 2M / 40\ KHz$$

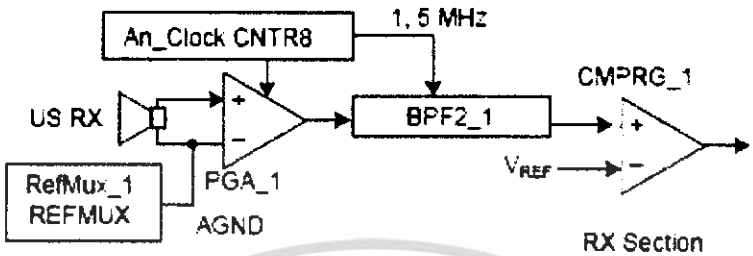
$$= 50$$

และทำการกำหนดค่า Compare Value ไว้ที่ครึ่งหนึ่งของ 50 ซึ่งก็คือ 25 เพื่อให้ได้สัญญาณที่มีค่า Duty Cycle = 50% เนื่องจากต้องการให้เอาต์พุทมีค่าลอจิก 0 หลังจากที่เป็น 1 ผ่านมาเป็นระยะเวลาครึ่งคาบ ตามที่กำหนดไว้จากความถี่ที่ต้องการซึ่งก็คือ 25 μS

และจากเอาต์พุทที่ได้จาก Counter8 เพียงอย่างเดียวนั้นไม่เพียงพอต่อการขับให้อัลตราโซนิคทรานดิวเซอร์ทำงานได้ในระยะไกลเท่าที่ควร จึงทำการเพิ่มแรงดันให้กับทรานดิวซ์เซอร์อีก โดยการนำสัญญาณที่กลับเฟสโดย DigInv มาจ่ายให้กับอีกขั้วหนึ่งของอัลตราโซนิคทรานดิวเซอร์ ดังนั้นจึงทำให้แรงดันที่ตกคร่อมอัลตราโซนิคทรานดิวเซอร์ขนาด 10 Vp-p ความถี่ 40 KHz เพื่อให้ได้ขนาดของสัญญาณคลื่นอัลตราโซนิคที่แรงมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การออกแบบส่วนที่ใช้ในการรับสัญญาณคลื่นอัลตราโซนิค



รูปที่ 5.3 แสดงบล็อกไดอะแกรมของภาครับสัญญาณ

จากระยะทำการไกลสุดของเครื่องวัดระยะทางที่ได้กำหนดไว้คือ 1.5 m ซึ่งก็คือ คลื่นอัลตราโซนิคจะต้องเดินทางไปและกลับเป็นระยะทางรวม 3 เมตร สามารถ หาแรงดันเอาต์พุตที่ออกมาจากอัลตราโซนิคทรานดิวเซอร์ของภาครับได้โดย

จาก SPL(Sound Presser Level) ของทรานซ์ดิวเซอร์คือ 119 dB 40 KHz โดย 0 dB =0.0002 μbar ที่สัญญาณ 10 V และเดินทางเป็นระยะทาง 30 เซนติเมตร ดังนั้น10 V ที่ระยะทาง 300 เซนติเมตรจะได้ SPL ที่เอาต์พุตของทรานดิวเซอร์ จะได้เท่ากับ

SPL ที่ถูกลดทอนลงที่ระยะทาง 3 เมตร = 20log (30/300)

= -20 dB

SPL ที่ถูลดทอนที่แรงดัน 10 V = 20log(20/20)

= 0 dB

ดังนั้น SPL คงเหลือ = 119 – 20 dB

= 99 dB

จาก 0 dB คิดเป็น 0.0002 ดังนั้นที่ 99 dB สัญญาณเอาต์พุตของคลื่นอัลตราโซนิคจะมีขนาด

99 = 20log(SPL/0.0002)

SPL = 17.825 μbar

และจากทรานดิวเซอร์ตัวรับคลื่นอัลตราโซนิคมีความไว -65 dB ดังนั้น

-65 = 20log(output in volt/μbar)

output in volt/μbar = 0.0005623

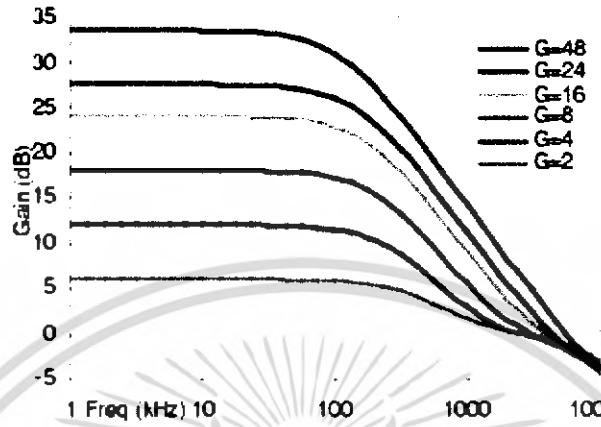
จากขนาด SPL ที่ส่งมาถึงมีค่าเท่ากับ 17.825 μbar ดังนั้น

V_{out} = 17.825 X 0.0005623 V

= 10 mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสัญญาณที่ได้จะเห็นว่ามีขนาดสัญญาณที่น้อยมากดังนั้นจึงโปรแกรมให้ PGA ทำการขยายที่อัตราขยาย 48 เท่า

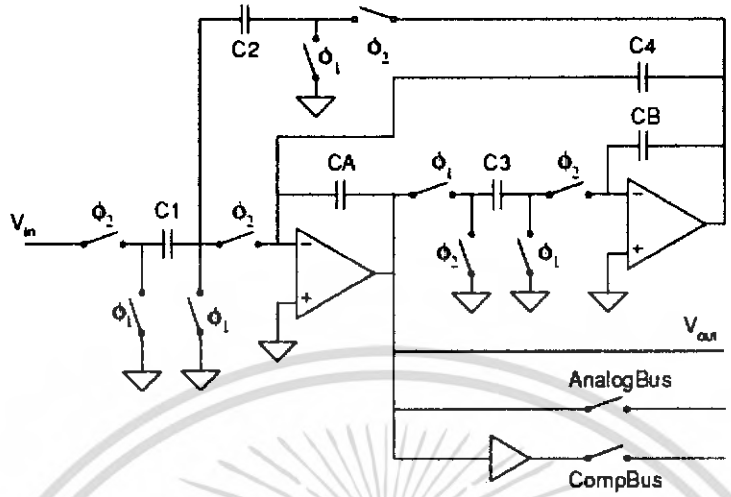


รูปที่ 5.3 แสดงผลของความถี่ต่างๆที่มีต่อกำลังขยายของ PGA

แต่จากกราฟข้อมูลของ PGA ระบุไว้ว่าที่การทำงานในช่วงความถี่ 40 KHz จะทำให้ PGA สามารถขยายสัญญาณ ได้เพียง 34 เท่าเท่านั้น ดังนั้นจึงทำให้ขนาดสัญญาณเอาต์พุตที่ได้มีขนาด 340 mVp-p หลังจากที่มีคลื่นสะท้อนผ่านระยะทาง 3 เมตรมาตกกระทบตัวทรานดิวเซอร์

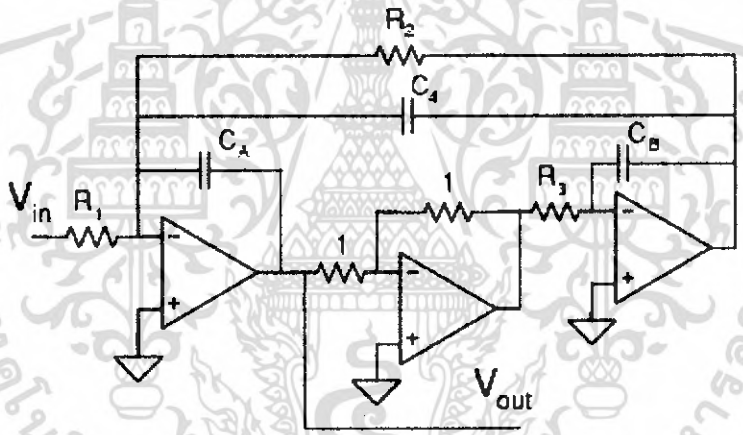
หลังจากที่ได้ทำการขยายสัญญาณจากเอาต์พุททรานดิวเซอร์ตัวรับให้มีขนาดสัญญาณแรงขึ้นแล้ว แต่สัญญาณที่ได้นั้นยังมีลักษณะของสัญญาณความถี่อื่นๆรบกวนอยู่ตลอดจึงอาจทำให้ไม่สามารถรับคลื่นสัญญาณน้อยๆที่มีขนาดสัญญาณเล็กกว่าสัญญาณรบกวนดังกล่าว เข้ามาได้ จึงได้ทำการออกแบบให้มีส่วนของวงจรกรองแถบความถี่ผ่านโดย Block ของวงจรกรองแถบความถี่ผ่านภายใน PSoC นั้นได้ออกแบบไว้โดยเป็นวงจรกรองแบบใช้ switched capacitor ภายในซึ่งมีโครงสร้างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 โครงสร้างภายในของวงจรกรองภายใน PSoC

ซึ่งสามารถเขียนวงจรเสมือนได้โดย



รูปที่ 5.5 แสดงวงจรเสมือนของวงจรกรองความถี่ภายใน PSoC

จากรูปสามารถขา Transfer Function ได้เป็น

$$\frac{V_{OUT}}{V_{IN}} = \frac{-\frac{R_2}{R_1 R_2 C_A}\frac{1}{s}}{s^2 + s\frac{C_4}{R_3 C_A C_B} + \frac{1}{R_2 R_3 C_A C_B}} \qquad \text{.....(1)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแปลงสมการให้อยู่ในรูปของ Switched Capacitor ได้เป็น

$$\frac{V_{OUT}}{V_{IN}} = \frac{-\frac{C_1C_B}{C_2C_3}\frac{s\left(1+\frac{s}{2f_s}\right)f_s}{\left(\frac{C_AC_B}{C_2C_3}-\frac{1}{4}-\frac{1}{2}\frac{C_4}{C_2}\right)}}{s^2+\frac{C_4}{C_2}\frac{sf_s}{\left(\frac{C_AC_B}{C_2C_3}-\frac{1}{4}-\frac{1}{2}\frac{C_4}{C_2}\right)}+\frac{f_s^2}{\left(\frac{C_AC_B}{C_2C_3}-\frac{1}{4}-\frac{1}{2}\frac{C_4}{C_2}\right)}}$$

.....(2)

แลจาก Transfer Function ของวงจรกรองโดยทั่วๆไปมีรูปแบบดังนี้

$$\frac{V_{OUT}}{V_{IN}} = \frac{\frac{G}{Q}\omega_0 s}{s^2+\frac{s\omega_0}{Q}+\omega_0^2}$$

.....(3)

ดังนั้นเมื่อเทียบสมการที่ 2 กับ 3 แล้วสามาหาค่าต่างๆซึ่งใช้ในการออกแบบวงจรกรองคือ

$$Q = \frac{C_2}{C_4}\left(\frac{C_AC_B}{C_2C_3}-\frac{1}{4}-\frac{1}{2}\frac{C_4}{C_2}\right)^{\frac{1}{2}}$$

....(4)

$$f_c = \frac{1}{2\pi}\frac{f_s}{\left(\frac{C_AC_B}{C_2C_3}-\frac{1}{4}-\frac{1}{2}\frac{C_4}{C_2}\right)^{\frac{1}{2}}}$$

....(5)

$$G = -\frac{C_1C_B}{C_4C_3}$$

....(6)

จากการใช้งานในการกรองความถี่สัญญาณอินพุตจาก PGA กำหนดให้ กรองความถี่ที่

f_c = 40 KHz , G = 33 , และ Bandwidth = 1.6 KHz

ทำให้สามาหาค่าตัวแปรต่างๆภายใน Block ของ วงจรกรอง ได้ดังนี้

$C_1 = 10$, $C_2 = 31$, $C_3 = 1$, $C_4 = 7$, $C_A = 32$, $C_B = 32$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่สัญญาณผ่านวงจรกรองแถบความถี่ผ่านแล้ว จะทำให้เหลือเฉพาะความถี่ที่ใช้ในการแยกแยะว่าได้รับ หรือ ไม่ได้รับสัญญาณจากอัลตราโซนิคทรานดิวเซอร์ และจากการทดลองสามารถระบุได้ว่า ขนาดของสัญญาณอินพุทที่ส่งเข้าไปยัง CMPPRG นั้นหากเป็นสัญญาณเงียบที่ไม่มีการสะท้อนของคลื่นกลับมาสู่ทรานดิวเซอร์จะมีลักษณะเป็นสัญญาณเรียบๆซึ่งมีระดับเท่ากับระดับของ AGND (Analog Ground) แต่หากมีสัญญาณสะท้อกลับเข้ามาสู่ทรานดิวเซอร์ จะทำให้สัญญาณที่ได้มีลักษณะเป็นสัญญาณไซน์ 40 KHz แกว่งตัวอยู่บนระดับของสัญญาณ AGND และมีระดับต่ำสุดที่ 4 V

และจาก

$$V_{Threshold} = LowLimit + (Vdd - LowLimit) \cdot RefValue$$

ดังนั้นจึงตั้งค่า RefValue ไว้ที่ 0.812 เพื่อสร้างแรงดันอ้างอิง 4.06 V ให้กับ Comparator หลังจากนั้น หากมีสัญญาณคลื่นอัลตราโซนิคสะท้อนกลับมา จะทำให้ CMPPRG มีลอจิกเอาท์พุตเป็น 1 เพื่อแจ้งให้กับส่วนของวงจรนับทำการหยุดนับและแสดงผลออกมาเป็น เซนติเมตร

5.2.3 การออกแบบในส่วนของภาคนับระยะเวลาที่คลื่นอัลตราโซนิคใช้ในการเดินทาง

รูปที่ 5.6 แสดงส่วนของภาคนับระยะเวลาที่คลื่นใช้ในการเดินทาง

จากการออกแบบ จะต้องการแบบให้ Timer ทำการนับขึ้นและส่งสัญญาณไปยัง Meter ทำการรับรู้ว่าระยะเวลาได้ผ่านไปเป็นระยะเวลาครบรอบตามระยะเวลาที่คลื่นอัลตราโซนิคจะเดินทางได้ 1 เซนติเมตรแล้ว ดังนั้นสามารถหาค่า Period Value ได้โดย

ความเร็วของเสียง = 331 + 0.6 T

= 331 + 0.6 (27)

= 347.2 m/s

ระยะทาง 1 cm เสียงจะใช้เวลา = 1/34720

= 28.8 μs

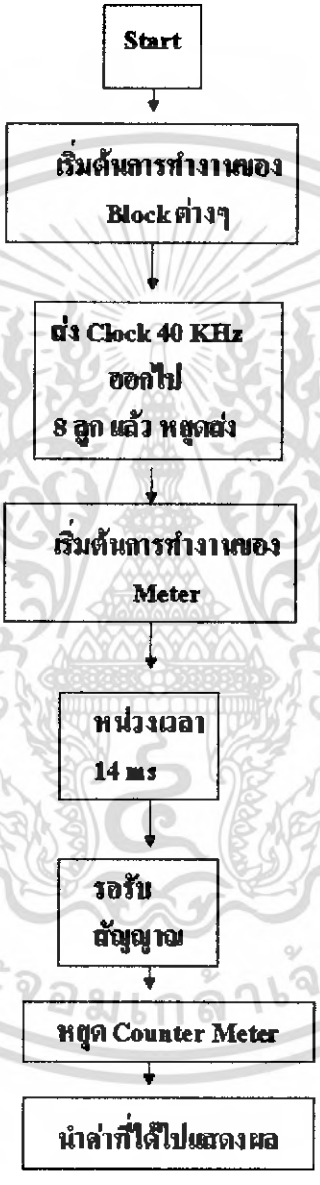
แต่การวัดระยะทาง 1 cm หมายความว่าเสียงเดินทาง 2 cm

ดังนั้นการวัด 1cm จะต้องใช้เวลา = 57.6 μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นหากต้องการให้ clock 2 MHz นับครบรอบทุกๆ 57.6 µs จะต้องกำหนดค่าให้กับ Timebase โดยมีค่า Period Value = 115 โดยมีค่า Compare Value = 57

5.2.4 การออกแบบในส่วนของโปรแกรม



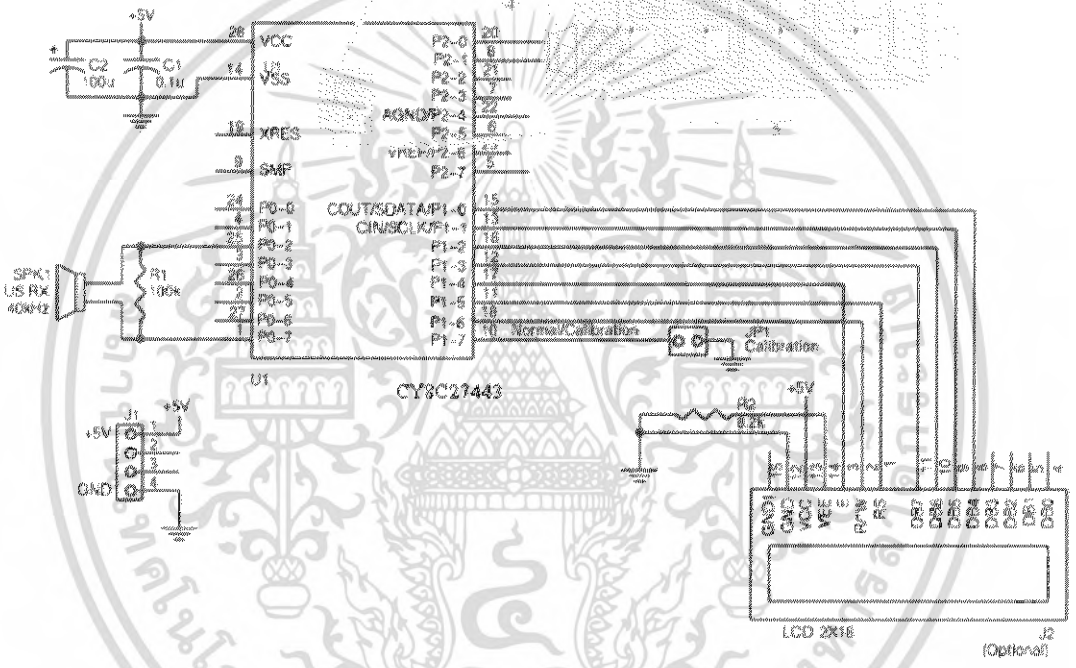
รูปที่ 5.7 แสดงการทำงานของโปรแกรมที่ใช้ในการควบคุม PSoC Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมเริ่มจาก การตั้งค่าต่างๆ ของบล็อกต่างๆให้เริ่มทำงาน และหลังจากนั้นจึงทำการส่งความถี่ 40 KHz ออกไปยัง อัลตราโซนิคทรานดิวเซอร์ โดยการส่งจะต้องทำการส่งโดยส่งออกไปครั้งละ 8 ลูก แล้วหยุดนับ เนื่องจากหากทำการส่งตลอดเวลา จะทำให้คลื่นมีการส่งผ่านไปยังตัวรับโดยตรงได้ ดังนั้นจึงทำการหน่วงเวลาไว้ที่ 2 μs แล้วทำการหยุดส่งความถี่ 40 KHz ออกไป หลังจากนั้นจึงเริ่มการทำงานของ Meter ซึ่งจะเริ่มนับระยะเวลาทันทีหลังจากที่หยุดส่งความถี่ 40 KHz ออกไป หลังจากนั้นจะให้ภาครับรอสัญญาณที่ส่งและสะท้อนกลับมา หลังจากที่ทำการหน่วงเวลาเอาไว้ 14ms เนื่องจากในตอนเริ่มต้นของการส่งหากทำการรับค่าสัญญาณอัลตราโซนิคเข้ามาเลยจะทำให้ได้รับสัญญาณที่ถูกส่งเข้ามาโดยตรงเท่านั้นจึงทำการหยุดรับเอาไว้ก่อน เมื่อภาครับได้รับสัญญาณแล้ว จะทำการหยุดนับตัวนับระยะเวลา และทำการนำค่าที่ได้ไปแสดงผลออกทาง LCD และ Buzzer

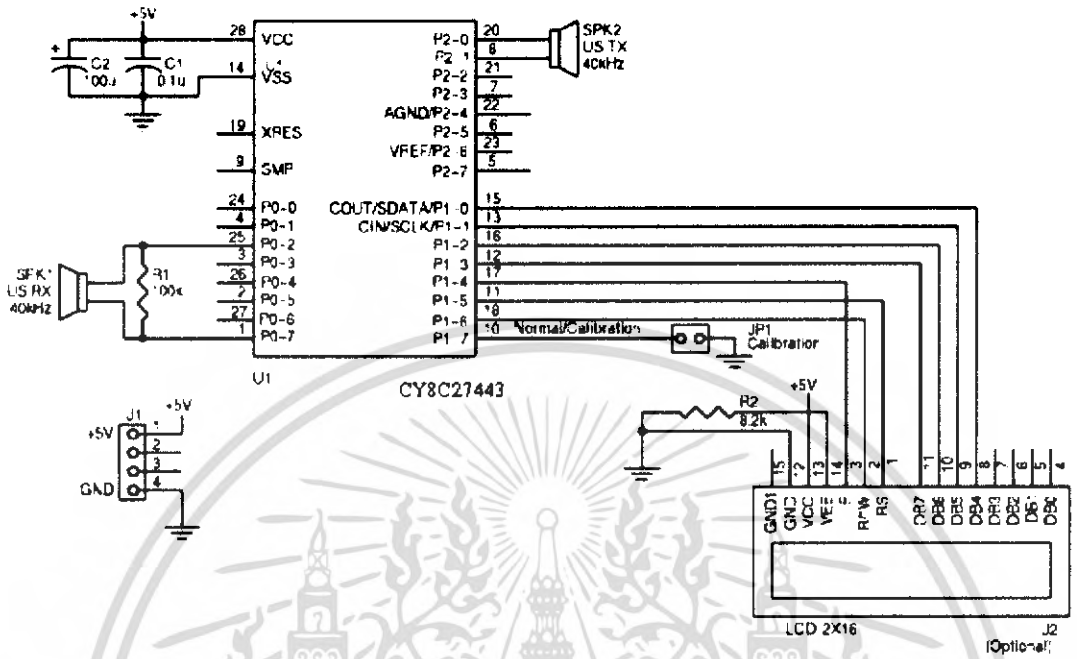
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 แสดงในส่วนของวงจรทั้งหมด



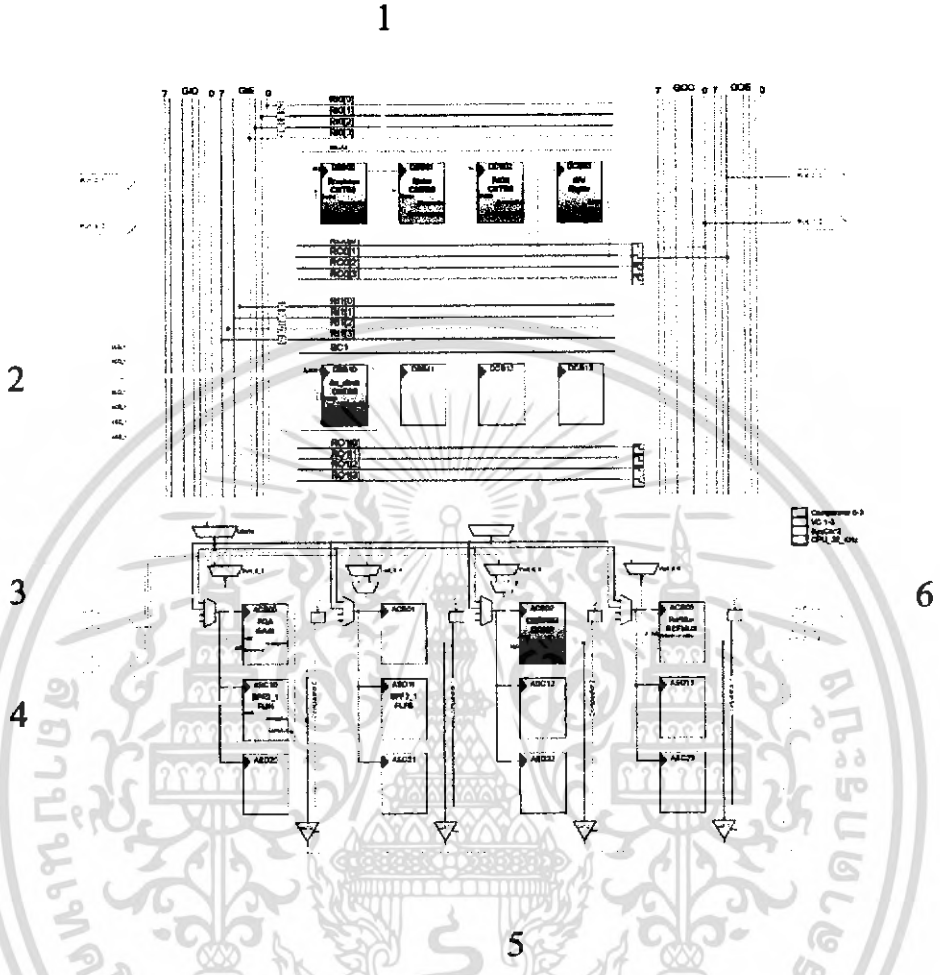
รูปที่ 5.8 แสดงวงจรทั้งหมด เปรียบเทียบกับ block diagram ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แสดงการเชื่อมต่อในส่วนของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดง block ภายในของ PSOC Microcontroller

1.แสดงการเชื่อมต่อของ Block ในส่วนของภาคกำเนิดสัญญาณ 40 KHz เพื่อส่งออกไปขับ Ultrasonic Transducer

2.แสดงส่วนของ Analog Clock เพื่อใช้เป็นสัญญาณนาฬิกาให้กับ Analog Block

3.แสดงส่วนของ Block PGA สำหรับขยายสัญญาณ Ultrasonic ที่สะท้อนกลับมายัง Ultrasonic Transducer ภาครับ

4.แสดงการเชื่อมต่อของ Block เพื่อสร้างภาคกรองสัญญาณแถบความถี่ผ่าน

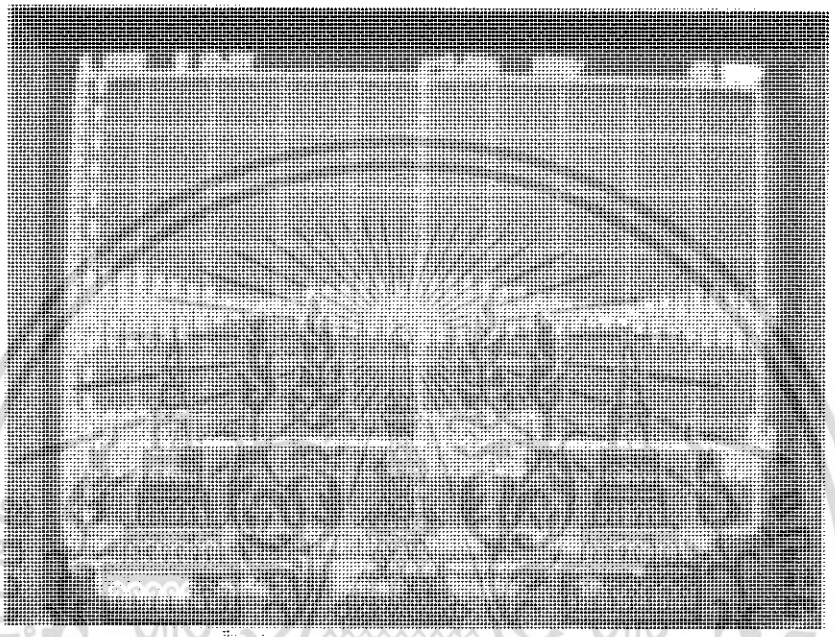
5.แสดงสัญญาณในส่วนของ Block comparator เพื่อใช้ในการตรวจจับสัญญาณสะท้อนกลับมา

6.แสดงส่วนของ block ที่ทำหน้าที่ในการสร้างแรงดันอ้างอิงให้กับBlock ต่างๆ

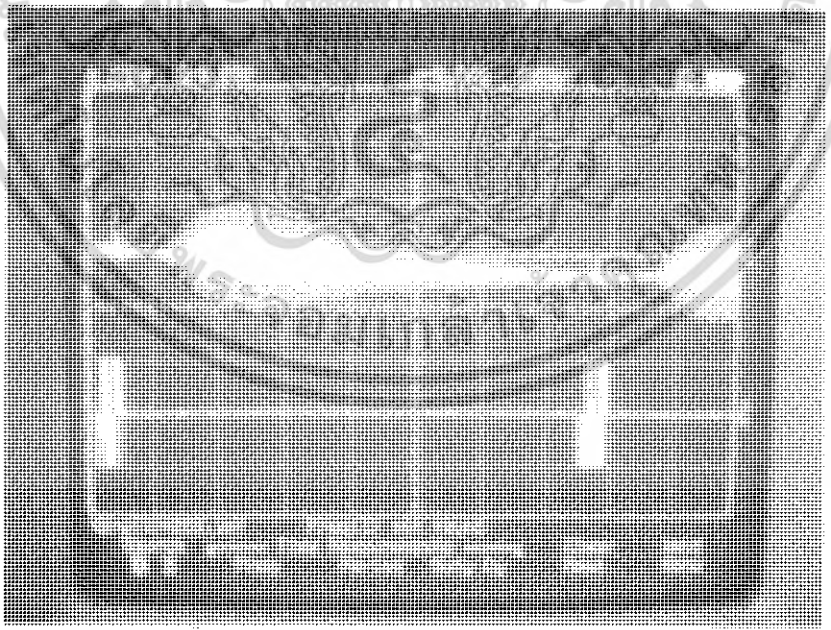
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ผลการทดลอง

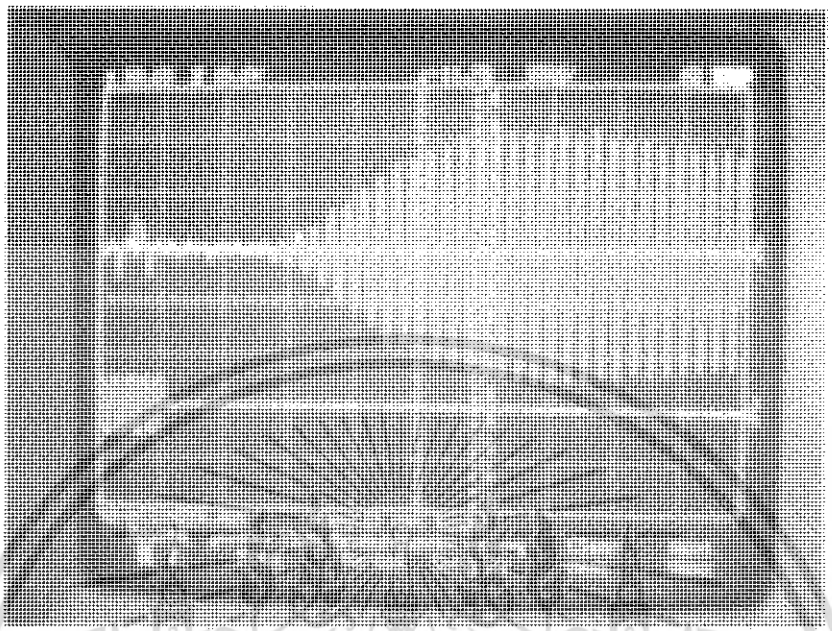


รูปที่ 6.1 แสดงสัญญาณที่ได้จากการส่ง และรับสัญญาณโดยไม่ใช้ฟังก์ชันหน่วงเวลา

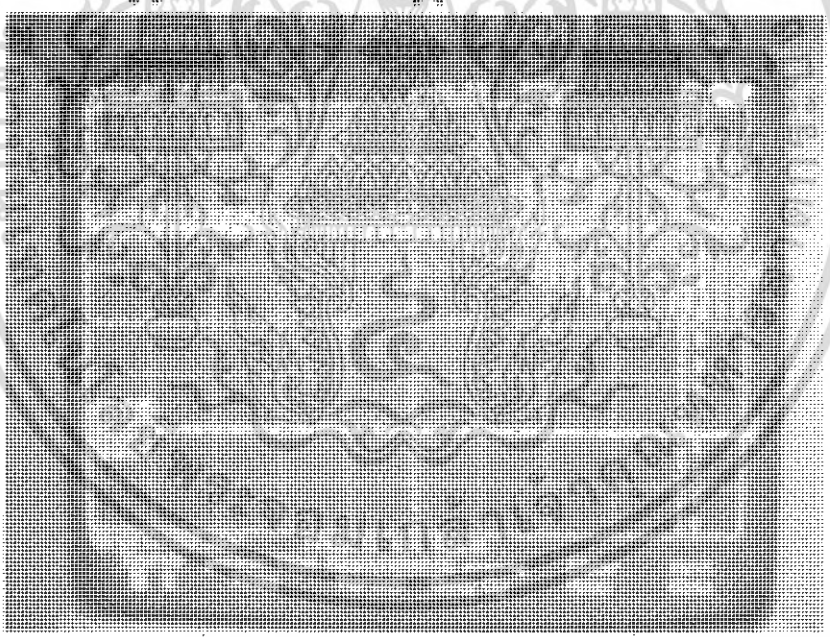


รูปที่ 6.2 แสดงสัญญาณที่ได้จากการหน่วงเวลาเฉลี่ยสำหรับการเข้ารหัส Ethernet ในตัวรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

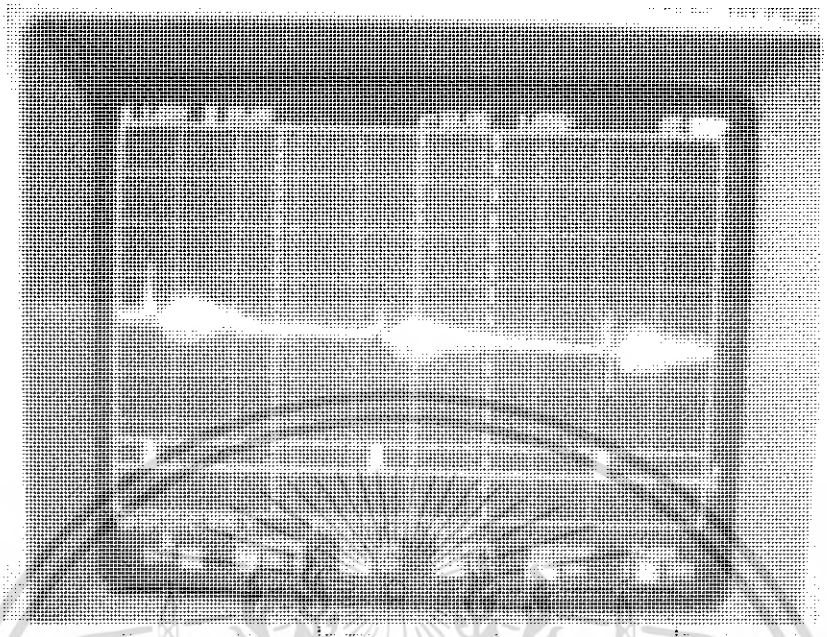


รูปที่ 6.3 แสดงเซ็นทรอยด์ที่คำนวณได้จากเซ็นทรอยด์ของวัตถุในภาพที่ระยะทาง 5 เซนติเมตร

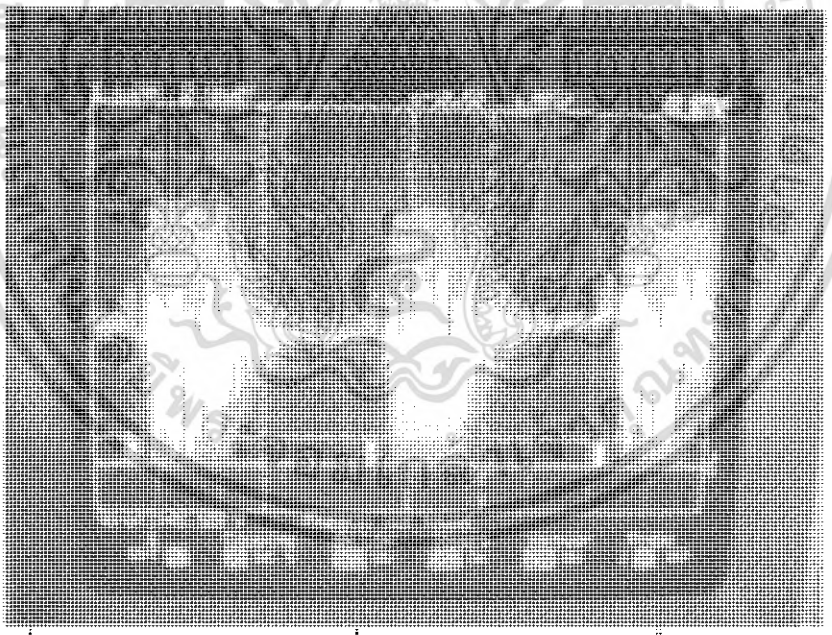


รูปที่ 6.4 แสดงเซ็นทรอยด์ที่คำนวณได้จากเซ็นทรอยด์ของวัตถุในภาพที่ระยะทาง 10 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

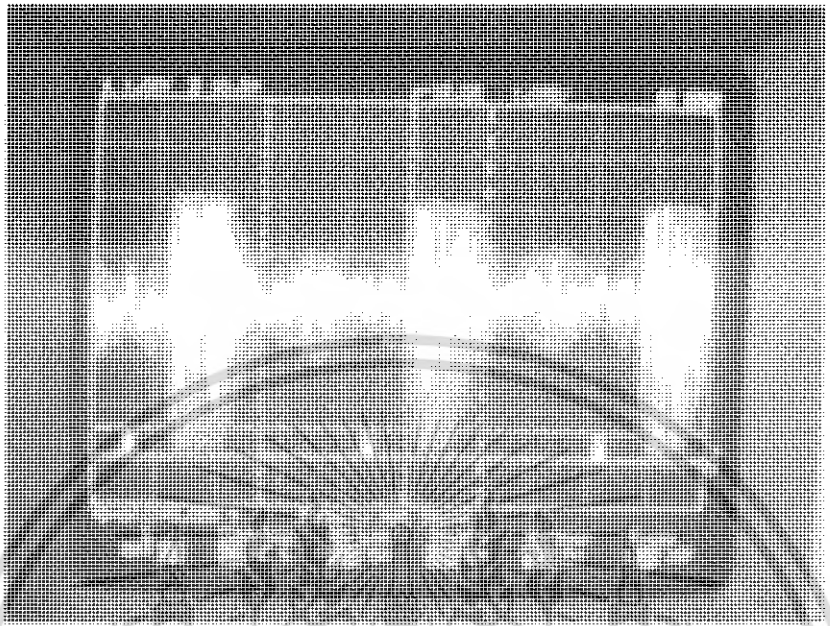


รูปที่ ๘.๖ แสดงเขื่อนลูกตะเข้ขวางที่ได้รับการจ่ายไฟจาก ควบ. และที่ไม่มีไฟฟ้าจ่าย



รูปที่ ๘.๗ แสดงเขื่อนลูกตะเข้ขวางที่ได้รับการจ่ายไฟจาก ควบ. ที่มีไฟฟ้าใช้ในเวลาต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

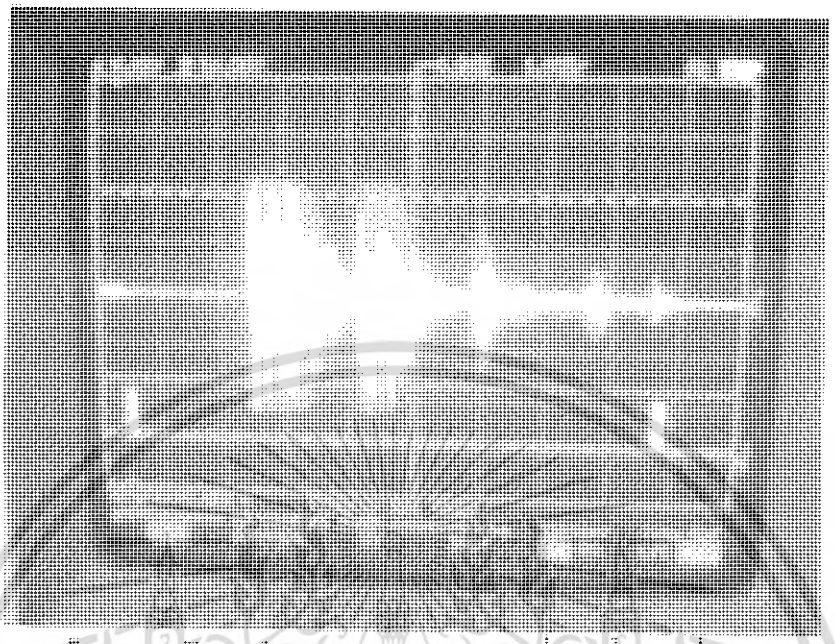


รูปที่ 6.7 แสดงสัญญาณแม่เหล็กที่ได้รับการขยายโดย PGA ที่มีวัตถุที่ระยะ 10 เซนติเมตร

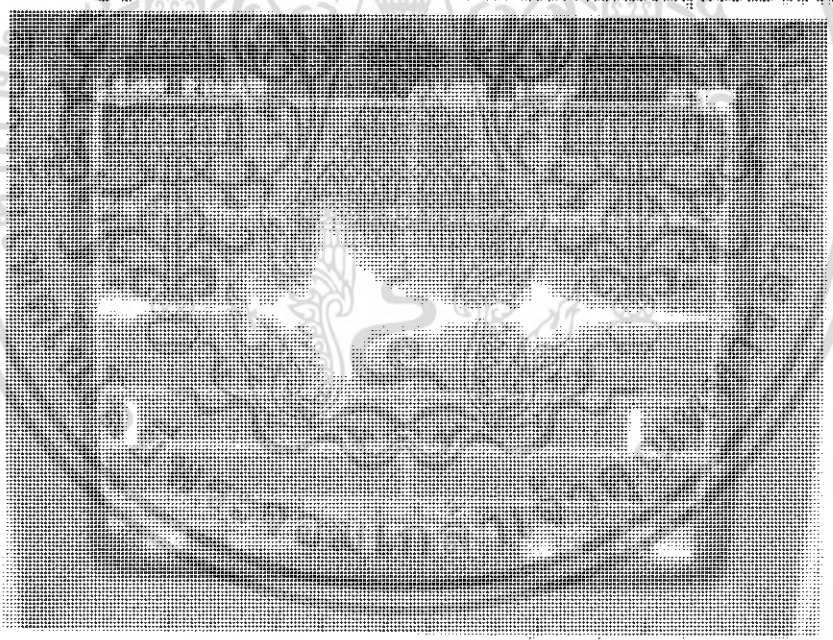
รูปที่ 6.8 แสดงสัญญาณแม่เหล็กที่ได้รับการขยายโดย PGA ที่มีวัตถุที่ระยะ 30 เซนติเมตร

จากรูปที่ 6.7 และ 6.8 สัญญาณที่สะท้อนกลับมาและอ่านค่าได้จะมีระยะเวลาที่ใช้ในการสะทอนนานมากขึ้นแปรผันตามระยะเวลาที่คลื่นใช้ในการเดินทาง และมีสัญญาณรบกวนปะปนมาด้วย
ทำให้ไม่สามารถรับสัญญาณที่สะท้อนมาจากระยะทางไกลๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



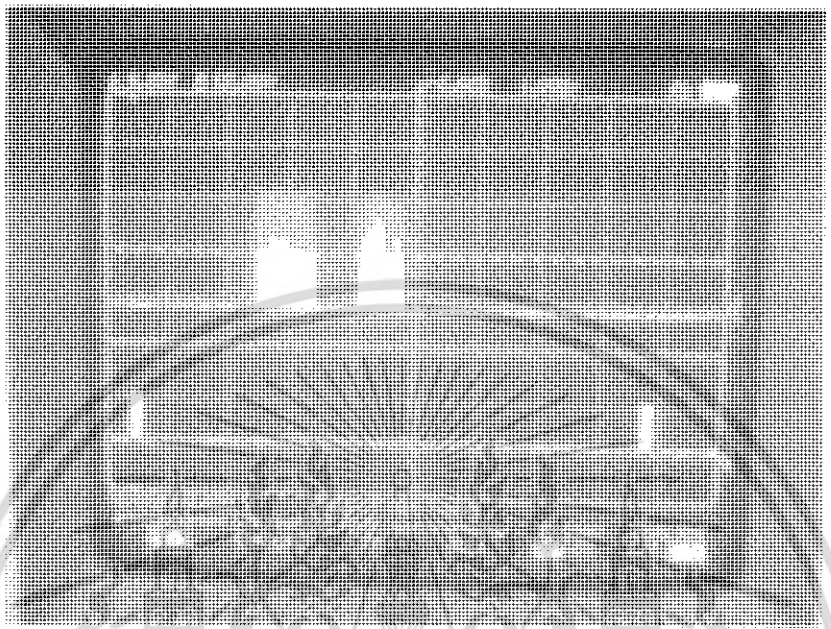
รูปที่ 6.9 แสดงสัญญาณสะท้อนกลับจากตำแหน่งของแนวความเสียหายที่ผ่านที่มีขนาดระยะ 30 เซนติเมตร



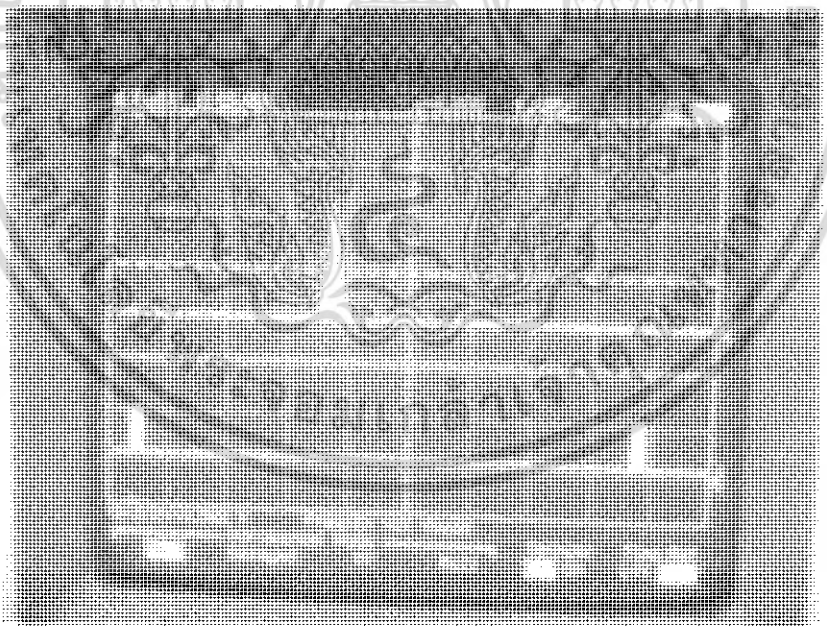
ที่ 6.9 แสดงสัญญาณสะท้อนกลับจากตำแหน่งของแนวความเสียหายที่ผ่านที่มีขนาดระยะ 60 เซนติเมตร

จากรูปที่ 6.8 และ 6.9 สัญญาณที่สะท้อนกลับมาและอ่านค่าได้จะมีระยะเวลาที่ใช้ในการสะท้อนนานมากขึ้นแปรผันตามระยะเวลาที่คลื่นใช้ในการเดินทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ที่ 6.10 แสดงสัญญาณหลังจากผ่านวงจรเปรียบเทียบแรงดันที่มีวัตถุที่ระยะ 30 เซนติเมตร

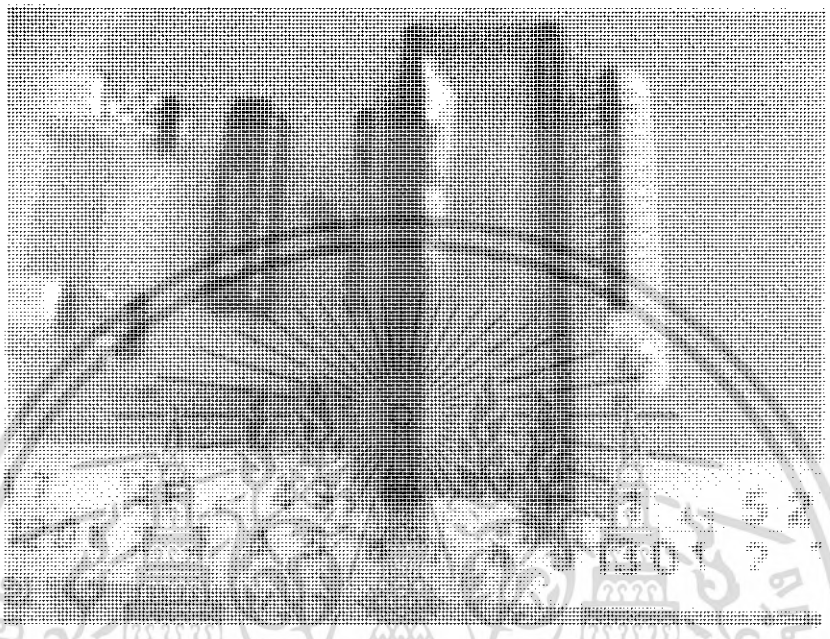


ที่ 6.11 แสดงสัญญาณหลังจากผ่านวงจรเปรียบเทียบแรงดันที่มีวัตถุที่ระยะ 60 เซนติเมตร

จากรูปที่ 6.10 และ 6.11 สัญญาณที่สะท้อนกลับมาและอ่านค่าได้จะมีระยะเวลาที่ใช้ในการสะทอนนานมากขึ้นแปรผันตามระยะเวลาที่คลื่นใช้ในการเดินทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบทางการวัดระยะทางจริง กับ การวัดโดยใช้เครื่องวัดโดยอัลตราโซนิค



การวัดจะทำโดยการเปรียบเทียบการวัดโดยเครื่องวัดระยะทางอัลตราโซนิคกับการวัดโดยใช้อุปกวัดจริง เพื่อทำการเปรียบเทียบค่าที่ได้ว่าสามารถวัดได้แม่นยำหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองจากการวัด 3 ครั้งเปรียบเทียบกับการวัดจริง

ระยะทาง (cm)	วัดครั้งที่ 1 (cm)	วัดครั้งที่ 2 (cm)	วัดครั้งที่ 3 (cm)	ค่าเฉลี่ย (cm)	%ความผิดพลาด
40	41	49	37	42.3	5.75
50	53	48	48	49.7	0.6
60	59	62	61	60.6	1
70	72	69	71	70.6	0.85
80	78	80	82	80	0
90	90	92	94	92	2.2
100	101	99	101	100.3	0.3
110	110	110	112	110.6	0.54
120	121	117	122	120	0
130	128	134	130	130.6	0.46
140	137	141	142	140	0
150	153	145	151	149.7	0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการทดลอง

จากผลการทดลองจะเห็นว่าค่าที่ได้จากการวัดมีความผิดพลาดอยู่บ้าง แต่ก็มีค่าใกล้เคียงกับค่าที่วัดระยะทางจริงๆ เนื่องมาจากคลื่นสัญญาณที่ได้จากการสะท้อนกลับของคลื่นอัลตราโซนิคนั้นมีขนาดสัญญาณที่เล็กมาก แต่ตัว PGA ภายใน PSoC นั้นมีกำลังขยายสัญญาณสูงสุดเพียง 48 เท่านั้น ทำให้ไม่สามารถขยายสัญญาณที่รับมาให้มีขนาดใหญ่เพียงพอต่อการตรวจจับได้ จึงทำให้เกิดความผิดพลาดในการวัดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมที่ใช้ในการควบคุม PSoC Micrcontroller

```
//-------------------------------------------------------------------
// C main line
//-------------------------------------------------------------------

#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Modules
#include <ioport.h>
#include <delay.h>
char TextBuff[5];
const WORD PeriodValue=255;
void main()
{char A,B;

 BPF2_1_SetC1( 10 );
 BPF2_1_SetC2( 31 );
 BPF2_1_SetC3( 1 );
 BPF2_1_SetC4( 7 );
 BPF2_1_SetCA(1);
 BPF2_1_SetCB(1);
 BPF2_1_Start(3);
 BPF2_1_SetPolarity(0);
 PGA_1_SetGain(12);
 CMPPRG_1_SetRef(192);      // Set RefValue
 while(1)
 {
 Timebase_WritePeriod(115);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timebase_WriteCompareValue(57);
Meter_WritePeriod(255);
Meter_WriteCompareValue(0);
Timebase_Start();
A=0;
B=0;
ClrBit0_3;
An_clock_Start();
PGA_1_Stop();
RefMux_1_Start(2);
Counter8_Start();
DigInv_Start();
Delay100uS(2);
Meter_Start();
Counter8_Stop();
DigInv_Stop();
PGA_1_Stop();

Delay100uS(42);
PGA_1_Start(3);
CMPPRG_1_Start(3);

while(A<140)
{                                                                    //
Begin Check Here
        if(Bit1_2==0)
                {
                Delay100uS(1);
                A=A+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                B=0;
                }
                else
                {
                A=141;
                B=1;
                }
}
        //wait until A=140 so display Below "No it 0"
if (B==0)
{
Meter_Stop();
LCD_1_Start();
LCD_1_Position(0,4);
LCD_1_PrCString("Out of Range");
}
else
{
if((PeriodValue-6-Meter_bReadCounter())<35)
{
Meter_Stop();
LCD_1_Start();
LCD_1_Position(0,1);
LCD_1_PrCString("Less Than 35 CM");
ClrBit1_4;
Delay1mS(100);

SetBit1_4;
Delay1mS(50);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ClrBit1_4;
Delay1mS(100);

SetBit1_4;
Delay1mS(50);
ClrBit1_4;
Delay1mS(100);

SetBit1_4;
Delay1mS(50);
ClrBit1_4;
Delay1mS(100);

SetBit1_4;
Delay1mS(50);
PGA_1_Stop();

ClrBit1_4;
Delay1mS(100);



}
else
{
Meter_Stop();
itoa(TextBuff,(PeriodValue-6-Meter_bReadCounter()),10);
LCD_1_Start();
LCD_1_Position(0,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD_1_PrCString("The Range is");
LCD_1_Position(1,4);
LCD_1_PrString(TextBuff);
LCD_1_Position(1,8);
LCD_1_PrCString("CM.");



ClrBit1_4;
Delay1mS(10*(PeriodValue-6-Meter_bReadCounter()));

SetBit1_4;
Delay1mS(5*(PeriodValue-6-Meter_bReadCounter()));

}
PGA_1_Stop();

}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้