เครื่องนับก้าวเดิน

PEDOMETER

โดย

นายประสิทธิ์ ประภาสัย
นายสมหมาย ไชยราษฎร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เครื่องนับก้าวเดิน

PEDOMETER

โดย

นายประสิทธิ์ ประภาสัย

นายสมหมาย ไชยราษฎร์

อาจารย์ที่ปรึกษา

ดร. กิติพล ชิตสกุล

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องนับก้าวเดิน

ผู้จัดทำ

1. นายประสิทธิ์ ประภาสัย
2. นายสมหมาย ไชยราษฎร์

....อาจารย์ที่ปรึกษา

(ดร. กิติพล ชิตสกุล)

เครื่องนับก้าวเดิน
(PEDOMETER)

นายประสิทธ์ ประภาสัย รหัส 47015252
นายสมหมาย ไชยราษฎร์ รหัส 47015265
ดร.กิติพล ชิตสกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาเครื่องนับก้าวเดิน (PEDOMETER) ซึ่งเป็นอุปกรณ์ที่ช่วยแสดงปริมาณการออกกำลังกายโดยวิธีการเดิน ซึ่งเป็นการออกกำลังกายที่สามารถทำได้ง่ายเป็นกิจวัตรประจำวันอยู่แล้ว เหมาะอย่างยิ่งสำหรับผู้ป่วยที่เป็นโรคหัวใจและโรคเบาหวาน และยังเหมาะสำหรับผู้ที่ต้องการควบคุมน้ำหนักตัวเองแต่ไม่มีเวลาออกกำลังกาย ดังนั้นการออกกำลังกายเพื่อเผาผลาญพลังงานในร่างกายโดยวิธีการเดินจึงเป็นอีกวิธีหนึ่งที่เหมาะสมที่สุด โดยในโครงงานนี้ได้มีการประยุกต์ใช้งานเซนเซอร์ที่ใช้เทคโนโลยี MEMs และสวิตซ์ปรอท (Mercury switch) เป็นตัวตรวจวัดการก้าวเดิน และใช้ไมโครคอนโทรลเลอร์ PSoC เป็นตัวควบคุมและประมวลผลการทำงาน โดยจะแสดงผลเป็นระยะทางที่เดิน จำนวนก้าวและปริมาณแคลลอรี่ที่ถูกเผาผลาญไป ออกทางหน้าจอกราฟฟิกของโทรศัพท์ โนเกียร์ 3310 ซึ่งหาซื้อง่าย และมีราคาถูกกว่าจอกราฟฟิกทั่วไป อีกทั้งยังออกแบบวงจรโดยใช้ใช้อุปกรณ์แบบ surface mounted (SMD) ทำให้มีขนาดเล็กกระทัดรัด สะดวกต่อการพกพาและการใช้งาน

PEDOMETER

Mr.Prasit Prapasai ID 47015252
Mr.Sommai Chairast ID 47015265
Dr.Kitiphol Chitsakul (Advisor)
Educational Year 2006

Abstract

This project is concerned a development of PERDOMETER which is a device used as a monitor for daily exercises by walking. The device is not only suitable for one who has heart trouble, diabetes but also for anybody who need to burn energy from his/her body without doing the sports. The device is based on a MEMs technology sensor and Mercury switch, accelerometer used as footstep detection and a microcontroller type PSoC as central processing unit. A cheap mobile series N3310 screen graphic monitor is employed as functional display such as counted footsteps and burned calories. To give a small size, surface mounted device (SMD) is used to construct the project.

กิตติกรรมประกาศ

ในการจัดทำปริญญานิพนธ์ฉบับนี้จะไม่สามารถประสบความสำเร็จและลุล่วงได้ด้วยดี หากไม่มีท่านอาจารย์ที่ปรึกษา ดร.กิติพล ชิตสกุล และท่านอาจารย์หลาย ๆ ท่านที่ได้คอยให้ความรู้และให้คำแนะนำแนวการคิดและการแก้ปัญหาในการทำงานในแต่ละขั้นตอน รวมทั้งพี่ ๆ เพื่อนๆ ทั้งหลายที่คอยให้คำแนะนำและอำนวยความสะดวกต่างๆ จนผลักดันให้ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี

ลงชื่อ ประสิทธิ์ ประภาสัย

(นายประสิทธิ์ ประภาสัย)

ลงชื่อ สมหมาย ไชยราษฎร์

(นายสมหมาย ไชยราษฎร์)

สารบัญ

สารบัญ (ต่อ)

สารบัญรูป

หน้า

สารบัญรูป (ต่อ)

สารบัญตาราง

บทที่ 1

บทนำ

1.1 ความเป็นมาและวัตถุประสงค์ของโครงงาน

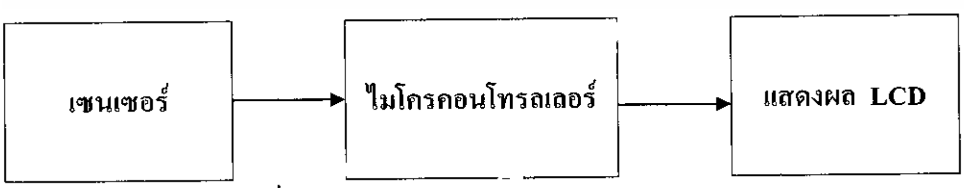
มนุษย์เราทุกวันนี้ต่างใช้ชีวิตกับการทำงานเป็นส่วนมาก เนื่องจากต้องทำงานแข่งขันกับเวลาจนลืมดูแลสุขภาพของตัวเองด้วยการออกกำลังกาย รวมทั้งในทุกๆวันนี้อาหารต่างๆ ก็มีสารอาหารที่เป็นประโยชน์น้อยรวมทั้งไขมันและคอลแลสโตรอลสูง เป็นผลทำให้คนส่วนมากมีน้ำหนักที่เกินมาตรฐานโดยมักจะเป็นโรคอ้วนโดยไม่รู้ตัว เนื่องจากการขาดการออกกำลังกายอย่างสม่ำเสมอนั่นเอง การออกกำลังกายนั้นมีหลายวิธีด้วยกันขึ้นอยู่กับความสะดวกของผู้ออกกำลังกายเอง การเดินก็ถือเป็นการออกกำลังกายวิธีหนึ่งเหมือนกันโดยเฉพาะอย่างยิ่งกับผู้ที่ป่วยเป็นโรคหัวใจ โรค เบาหวาน เป็นต้น จากการวิจัยพบว่าโดยปกติสำหรับบุคคลทั่วไป และผู้ที่ป่วยเป็นโรคหัวใจจะต้องเดินออกกำลังกายในแต่ละวันให้ได้อย่างน้อย 10,000 ก้าว จะช่วยทำให้สุขภาพของผู้ที่ปฏิบัติมีสุขภาพร่างกายที่ดีและลดอัตราการเกิดเป็นโรคหัวใจซึ่งเป็นสาเหตุต้นๆ ของการเสียชีวิตของคนในปัจจุบันนี้

เครื่องนับก้าวเดิน (PEDOMETER) ได้มีการสร้างขึ้นมาใช้งานก่อนหน้านี้แล้วโดยเป็นปริญญานิพนธ์หลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547 ซึ่งตัวเครื่องนับก้าวเดิน ในขณะนั้นยังคงมีขนาดที่ใหญ่มากและการใช้งานไม่สะดวกมากนัก ประกอบกับหน้าจอแสดงผลที่มีขนาดที่ใหญ่และฟังก์ชั่นการใช้งานยังมีให้ใช้งานน้อย จึงส่งผลให้ตัวเครื่องมีขนาดที่ใหญ่เพิ่มขึ้นอีกด้วย

รายงานนี้เป็นการออกแบบปรับปรุงใช้งานเครื่องนับก้าวเดิน (PEDOMETER) เพื่อให้มีขนาดที่เล็กลงสามารถพกพาและใช้งานได้สะดวกมากยิ่งขึ้น พร้อมทั้งมีฟังก์ชั่นการใช้งานเพิ่มเติมจากเครื่องนับก้าวเดินรุ่นเดิมมากยิ่งขึ้นอีกด้วยรวมทั้งหน้าจอแสดงผลที่ใช้หน้าจอ LCD ของ NOKIA รุ่น 3310 ซึ่งเป็นหน้าจอกราฟฟิกทำให้สามารถมองเห็นได้ชัดเจนและมีขนาดที่เล็กเป็นผลให้ตัวเครื่องนับก้าวเดินมีขนาดเล็กลงอีกด้วย รวมทั้งมีการใช้ตัวไมโครคอนโทรลเลอร์ PSoC MCU ที่มีลักษณะโดดเด่นมากมายมาเป็นตัวควบคุมและประมวลผลในการทำงานจึงสามารถใช้งานได้หลากหลายมากยิ่งขึ้นอีกด้วย

1.2 หลักการทำงานของเครื่องนับก้าวเดิน (PEDOMETER)

เครื่องนับก้าวเดินนี้เป็นการประยุกต์การใช้งาน ไมโครคอนโทรลเลอร์กับอุปกรณ์วงจรภายนอกที่เป็นอุปกรณ์เซ็นเซอร์และหน้าจอแสดงผลโดยมีลักษณะการทำงานดังรูป



รูปที่ 1 บล็อกไดอะแกรมของโครงงาน

ส่วนที่1 เซ็นเซอร์ ได้มีการออกแบบให้สามารถใช้อุปกรณ์เซนเซอร์ได้สองแบบคือ

1. ใช้เทคโนโลยีของ MEMS (Mechanical Machine System) รุ่น ADXL202 มาประยุกต์ใช้เป็นตัวเซ็นเซอร์ในการตรวจจับการก้าวเดินของผู้ใช้ ซึ่งอาศัยหลักการของการเปลี่ยนแปลงความเร่งในทิศทางการเคลื่อนที่เนื่องจากการก้าวเดิน และผลจากการก้าวเดินนั้นจะทำให้ ADXL202 ให้สัญญาณเอาต์พุตออกมาเป็นค่าการเปลี่ยนแปลงของ ค่าดิวตี้ไซเคิลที่เป็นสัดส่วนกับค่าความเร่งซึ่งจะทำให้เอาต์พุตที่มีการเปลี่ยนแปลงค่าเป็นเปอร์เซ็นของดิวตี้ไซเคิล

2. ใช้สวิตซ์ปรอทเป็นตัวเซนเซอร์การก้าวเดิน เนื่องจากหาซื้อง่าย และการใช้งานก็เหมือนสวิตซ์ทั่วๆ ไป เพียงแต่ใช้แรงที่มากระทำ หรือการเคลื่อนที่ของสวิตซ์เป็นตัวเปิด-ปิดการทำงานของสวิตซ์ปรอท

ส่วนที่ 2 ไมโครคอนโทรลเลอร์ ทำหน้าที่ในการควบคุมและประมวลผลทั้งหมดของระบบ โดยใช้ไมโครคอนโทรลเลอร์ PSoC MUC ในการรับสัญญาณเข้ามาจากอุปกรณ์เซ็นเซอร์ แล้วทำการประมวลผลซึ่งใช้การเขียนโปรแกรมในการคำนวณหาค่าการก้าวเดินโดยใช้การตรวจจับค่าการเปลี่ยนแปลงของค่าเปอร์เซ็นดิวตี้ไซเคิลจากสัญญาณที่ตัวเซ็นเซอร์ส่งออกมา แล้วนำไปประมวลผลและทำการแสดงค่าออกมาทางหน้าจอ LCD

ส่วนที่ 3 ส่วนแสดงผล LCD โครงงานนี้ใช้หน้าจอแสดงผลที่เป็นหน้าจอกราฟฟิก LCD ของ Nokia 3310 ซึ่งใช้ชิป รุ่น PCD 8544 ในการควบคุมการส่งข้อมูลเชื่อมต่อหน้าจอ LCD ทำให้การแสดงผลมีความหลากหลายมากขึ้นชัดเจนมากขึ้นเพื่อใช้ในการแสดงค่าของการก้าวเดิน และค่าปริมาณการเผาพลาญพลังงานที่สูญเสียในการเดิน

1.3 โครงสร้างของรายงาน

ปริญญานิพนธ์นี้ได้รวบรวมรายละเอียดความเป็นมาของรายงานและการเลือกใช้อุปกรณ์ต่างๆ เพื่อใช้ในการออกแบบวงจรในการใช้งานให้มีขนาดที่เล็ก พกพาง่ายและการใช้งานสะดวกมากยิ่งขึ้น ในการทำโครงงานนี้จะทำการแบ่งรายละเอียดออกเป็นบทต่างๆ ตามลำดับเพื่อให้ง่ายต่อการศึกษาทำความเข้าใจดังนี้

บทที่ 1 บทนำเกี่ยวกับความเป็นมาและลักษณะของโครงงานนี้

บทที่ 2 เป็นการกล่าวถึงการนำไปประยุกต์ใช้ในการออกกำลังกายด้วยการเดิน ซึ่งเป็นจุดประสงค์หลักในการออกแบบเครื่องนับก้าวเดินนี้ขึ้นมา

บทที่ 3 เป็นการอธิบายถึงหลักการและการทำงานของอุปกรณ์ที่ใช้ในการสร้างเครื่องนับก้าวเดิน ซึ่งมีอุปกรณ์หลักๆอยู่ 3 ส่วนด้วยกัน คือ อุปกรณ์เซ็นเซอร์ ไมโครคอนโทรลเลอร์ และหน้าจอแสดงผล LCD

บทที่ 4 เป็นส่วนของวงจรและการทดลองการใช้งาน

บทที่ 5 สรุปและวิจารณ์การทดลอง

บทที่ 2
การออกกำลังกายและการเดินเพื่อสุขภาพ

2.1 The power of walking การเดิน...เพื่อสุขภาพที่ดี

การเดินนอกจากจะเป็นส่วนหนึ่งในชีวิตประจำวันแล้ว ยังมีประโยชน์ในแง่ของการออกกำลังกาย แต่ภาวะการณ์ปัจจุบันทำให้คนเรา มักจะอ้างว่าไม่มีเวลาออกกำลังกาย แต่ที่จริงแล้วร่างกายมนุษย์ไม่ได้ถูกออกแบบมาเพื่อเคลื่อนไหวในแต่ละวันเพียงอย่างเดียวเท่านั้น แต่ผลของการเดินยังมีความสำคัญต่อหน้าที่ทางกายภาพของร่างกาย และยังเป็นการส่งเสริมสุขภาพและเป็นจุดเริ่มต้นของการมีสุขภาพชีวิตที่ดี เราจะมาพิจารณาถึงความเข้าใจ และประโยชน์ที่จะได้รับจากการเดินดังต่อไปนี้

2.2 ประโยชน์ของการเดินออกกำลังกาย

1. **Metabolism(การเผาผลาญ)** การเดินเป็นรูปแบบการเคลื่อนไหวที่มากที่สุดของมนุษย์ในอดีตเพราะการเดินมีความสำคัญต่อการยังชีพ ต่อมาได้มีการศึกษาว่า การเดินมีความจำเป็นต่อสุขภาพ การเดินเป็นการประสานงานของอวัยวะหลายส่วนร่วมกัน คือ แขน ขา และลำตัว การเดินยังสามารถทำให้เราใช้พลังงานไปถึง 100 แคลอรี อย่างง่ายดาย การเดินทุกวันไม่เพียงแต่มีผลในการเผาผลาญแคลอรี แต่ยังสามารถเพิ่มหน้าที่ของ enzyme และ metabolism อีกด้วย ซึ่งจะมีผลให้เพิ่มการเผาผลาญแคลอรี มากถึง 12 ชั่วโมง ภายหลังจากการเดินเพียงแค่ 2 ไมล์ อย่างไรก็ตามการเดินทำให้ metabolic effect ของการเดินเพิ่มขึ้น

การเพิ่ม metabolism หมายความว่าจะมีการเผาผลาญแคลอรีมากขึ้น และยังส่งผลให้ร่างกายใช้พลังงานจากอาหารและอาหารเสริมที่เราทานเข้าไปด้วย ทำให้เราอยากดื่มน้ำเพิ่มมากขึ้น และน้ำที่เราดื่มยังช่วยสนับสนุนการขับพิษ การขับถ่ายและการย่อยอาหารในร่างกายอีกด้วย

2. **Circulation(ระบบไหลเวียนเลือด)** การเดินมีผลต่อจังหวะการบีบและคลายตัวของกล้ามเนื้อ และมีการเปลี่ยนแปลงความดันในร่างกาย ทำให้เพิ่มการไหลเวียนของเลือด น้ำเหลือง และ Synovial fluid (ของเหลวหล่อลื่นที่ผลิตจากในข้อ) จะเห็นว่าการเดินทุกๆวันจะทำให้ระบบภูมิคุ้มกันดีขึ้นเนื่องจากมีการไหลเวียนของเลือดและน้ำเหลืองมากขึ้น

3. **Mobilization of internal organs (การเคลื่อนไหวของอวัยวะภายใน)** การเดินทำให้เกิด แรงดันในช่องว่างของร่างกาย จังหวะการเคลื่อนไหวของโครงสร้าง และการหดตัวของกล้ามเนื้อจะเปลี่ยนแปลงไป กระบังลมเป็นกล้ามเนื้อส่วนแรกที่ใช้ในการหายใจ และต่อมาจะมีการหดตัวของกล้ามเนื้อในช่องท้อง ทำให้เกิดสมดุลในร่างกาย ผลจากการที่กล้ามเนื้อกระบังลมหดตัวเป็นการบริหารกล้ามเนื้อภายในของตัวมนุษย์ การเพิ่มการหายใจจากการเดินทำให้ อวัยวะภายในเคลื่อนไหวมากขึ้นซึ่งเป็นผลดีต่อสุขภาพ ตัวอย่างจากประเทศสหรัฐอเมริกา เช่น คนที่ท้องผูกบ่อย การรักษาโดยการเดิน อาหารจะทำให้ลำไส้มีการเคลื่อนไหวมากขึ้น และสิ่งที่จำเป็นต่อมา คืออาหารที่มีประโยชน์ และน้ำรวมถึงอุปนิสัยการรับประทาน การใช้ยาระบายเป็นทางเลือกสุดท้ายที่จะใช้

4. **The P-M-E-S connection** มนุษย์เราเป็นสิ่งมีชีวิตที่มีองค์ประกอบ คือ ร่างกาย Physical [P] สติปัญญา Mental [M] อารมณ์ Emotional [E] จิตใจ Spiritual [S] หากลองสังเกตเพียงเล็กน้อยและ

พบว่าร่างกายของมนุษย์เฉื่อยชาหรือทำหน้าที่ได้ไม่เต็มประสิทธิภาพ รวมถึงด้านสติ ปัญญา อารมณ์และจิตใจ ทำให้เราเป็นทุกข์ได้ เราจะประหลาดใจมากที่สุดเมื่อพบว่าเราเดินเพียง 2 ไมล์ต่อวัน จะทำให้ P-M-E-S ของเราดีขึ้น

2.3 การเดินสามารถช่วยรักษาโรคและอาการต่างๆได้ดังนี้

ซึ่งในปัจจุบันทางการแพทย์พบว่าการออกกำลังกายที่ดีที่สุดคือการเดินเร็ว (ประมาณ 5.0 - 7.0 กิโลเมตรต่อชั่วโมง) นาน 30 นาที สัปดาห์ละ 5-6 วัน จากรายงานของ Dr. JoAnn Manson แห่ง Harvard's Brigham and Women's Hospital รายงานว่า การออกกำลังกายโดยการเดินเร็วนี้ สามารถช่วยลดอัตราการเกิดภาวะเรื้อรังของโรคต่างๆ ได้มากถึง 30-40 % แน่นอนในระยะแรกเราอาจไม่รู้ถึงความเปลี่ยนแปลงใดๆ เลย แต่ในระยะยาวแล้ว จะพบว่าร่างกายแข็งแรงขึ้น สามารถป้องกันโรคภัยไข้เจ็บต่างๆ ได้ เช่น

- ภาวะความดันโลหิตสูง โรคหัวใจ และ ภาวะสมองขาดเลือดไปเลี้ยง การออกกำลังกายโดยการเดินอยู่เป็นประจำนี้ จะกระตุ้นให้เลือดไปเลี้ยงส่วนต่างๆของร่างกายมากขึ้นทำให้กล้ามเนื้อหัวใจมีสภาพแข็งแรงขึ้น นอกจากนี้ยังช่วยลดระดับความดันโลหิต ช่วยกระตุ้นการสร้าง HDL (High Density Lipoprotein) มากขึ้น ช่วยลดความหนืดของเลือด ป้องกันภาวะเลือดเกาะตัวกันเป็นก้อน ทำให้เลือดไหลเวียนในร่างกายได้ดีขึ้น ลดความเสี่ยงต่อภาวะสมองขาดเลือด
- อาการเบาหวาน การเดินเร็วเป็นเวลา 30 นาทีต่อวันนี้สามารถป้องกันการเกิด หรือบรร เทาอาการเบาหวาน (Type II Diabetes Mellitus) โดยเฉพาะในผู้ที่น้ำหนักตัวมากกว่าปกติซึ่งการเดินสามารถช่วยได้
- ภาวะกระดูกพรุน และ ภาวะไขข้ออักเสบ การเดินเร็วนี้ นอกจากจะเพิ่มความแข็งแรงของกล้ามเนื้อแล้ว ยังช่วยให้ร่างกายดูดซึม Calcium ได้มากขึ้น ป้องกันภาวะกระดูกพรุน นอก จากนี้การเดินจะช่วยลดแรงกระแทกที่อาจเกิดขึ้นที่ข้อต่อได้ (ซึ่งการวิ่งจะทำให้ข้อต่อต้องรับการกระแทกที่มากกว่า) การที่กล้ามเนื้อรอบข้อต่อแข็งแรงขึ้น ก็สามารถช่วยบรรเทาอาการปวดจากโรคไขข้ออักเสบได้ด้วย การเดินในสระน้ำก็เป็นอีกหนทางหนึ่งที่ช่วยบรรเทาอาการไขข้ออักเสบได้ (ควรเดินสัปดาห์ละ 5-6 วัน เพื่อให้เวลาพักแก่ไขข้อบ้าง)
- โรคอ้วนหรือน้ำหนักเกินมาตรฐาน การเดินเร็วเป็นเวลาอย่างน้อย 30 นาที นอกจากจะช่วยเผาผลาญพลังงานได้ 200-300 แคลอรี่แล้ว ยังสามารถกระตุ้นอัตราการเผาผลาญพลังงานในภาวะปกติ (Metabolic Rate)ได้มากขึ้นอีกด้วย และหากทำการเดินมากกว่า 30 นาที ร่างกายก็เริ่มจะมีการเผาผลาญไขมันที่สะสมอยู่ให้เป็นพลังงาน สามารถลดน้ำหนักได้
- ภาวะซึมเศร้า การเดินเร็วนี้เป็นสันทนาการอย่างหนึ่ง ช่วยผ่อนคลายความตึงเครียดได้ดี จากรายงานนี้พบว่าผู้ป่วยอาการซึมเศร้าที่ไม่ได้รับยารักษา หลังจากออกกำลังกายโดยการเดินเร็วอยู่เป็นประจำนาน 10 เดือนแล้ว จะมีอัตราการเกิดภาวะซึมเศร้านี้น้อยกว่าผู้ป่วยที่ได้รับยารักษาเพียงอย่างเดียว

- อาการท้องผูก การเดินเร็วนี้จะเป็นการกระตุ้นให้ระบบทางเดินอาหารโดยเฉพาะบริเวณลำไส้มีการเคลื่อนไหวตัวมากขึ้น ทำให้สามารถขับถ่ายของเสียออกจากร่างกายได้ง่ายขึ้นรวมถึงลดการสะสมของสารพิษตามเซลล์ไขมันตับและไตได้

เราจะเห็นได้ว่าการเดินนั้นมีประโยชน์มากมายหลายประการจนไม่น่าเชื่อว่า สิ่งที่เราสามารถทำได้ง่ายๆด้วยตัวเราเอง และสามารถทำได้ตลอดเวลาโดยไม่ต้องเตรียมตัวหรือใช้สถานที่ใหญ่โตอะไร แต่การเดินก็มีข้อเสียด้วยเหมือนกันดังนี้

ข้อเสียของการเดิน

- หากเดินช้าเกินไป หัวใจจะเต้นไม่ถึงกำหนดทำให้หัวใจไม่แข็งแรงได้
- การเดินอาจต้องใช้ระยะทางที่ยาว ความอดทน และหาสถานที่ ที่ยาวมาก ๆ นั้นยาก
- การเดินเร็วอาจเป็นสิ่งที่น่าเบื่อสำหรับบางคน
- หากรองเท้าหรือเท้าของเราผิดรูปหรือมีลักษณะผิดปกติ อาจทำให้เดินลำบากและปวดเท้าได้

ข้อแนะนำในการเดินเร็ว

- หากร่างกายมีอาการอ่อนเพลีย เหนื่อยเวียนศีรษะ ฯลฯ ให้เดินช้าลง หรือถ้าเป็นหนักควรหยุดเดินทันที
- เมื่อเริ่มต้นเดินควรเดินช้า ๆ ก่อนเพื่อเป็นการปรับร่างกาย ไม่ควรเดินเร็วโดยทันทีเนื่องจากว่าอาจทำให้ร่างกายปวดเมื่อยได้
- สำหรับผู้ที่เป็นโรคปวดเมื่อยตามข้อกระดูกเท้า ควรปรึกษาแพทย์ก่อน อออกกำลังกายด้วยการเดิน โดยเฉพาะผู้ที่ป่วยเป็นโรคความดันโลหิตสูง หรือโรคหัวใจ

การออกกำลังกายโดยการเดินนี้ไม่ได้ช่วยรักษาโรคภัยไข้เจ็บได้ทุกโรคและไม่ได้เกิดประโยชน์เร็วในช่วงข้ามคืนหรือภายในสัปดาห์เท่านั้นแต่เราสามารถรักษาสุขภาพโดยรวมของเราให้แข็งแรงได้แม้ในยามที่เราแก่ตัวลงไปแล้ว ดังนั้นเราไม่จำเป็นต้องแข่งขันกับใครเลยในการมีสุขภาพที่ดี แต่ความสำคัญอยู่ที่ความสม่ำเสมอในการออกกำลังกายเท่านั้นเอง

บทที่ 3

ทฤษฎีและหลักการทำงานของส่วนต่างๆในรายงาน

3.1 อุปกรณ์เซนเซอร์

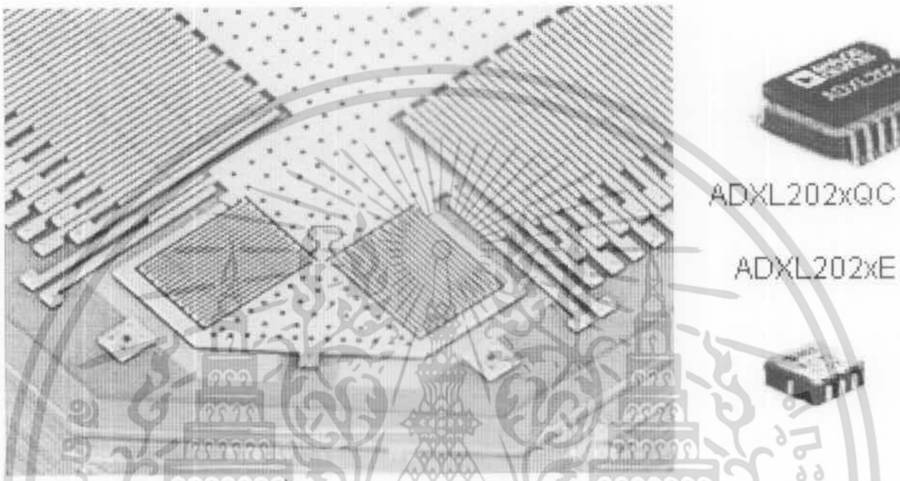
3.1.1 เมมส์ (iM3.1.1EMS ADXL202)

อุปกรณ์เซนเซอร์ที่ใช้ในการตรวจจับการก้าวเดินของโครงงานนี้ใช้ iMEMS Accelerometers เบอร์ ADXL202 เมมส์ เป็นอุปกรณ์ชนิดหนึ่งที่เรียกว่า "เครื่องกลจิ๋ว" เนื่องจากใช้ในการตรวจจับการเคลื่อนไหวหรืออัตราการเร่งโดยใช้กลไกภายในตัวชิฟไอซีแล้วจะให้สัญญาณเป็นรูปคลื่นพัลส์ออกมา โดยจะมีสัญญาณออกมาเป็นสองแกนคือในแนวแกน X และ แกน Y ด้วยประสิทธิภาพในการให้ความแม่นยำสูง และมีรูปแบบการใช้งานได้หลากหลายกว่า โดยสามารถทำงานประสานได้สอดคล้องกับวงจรไฟฟ้าทั่วไปที่ถูกย่อส่วนเล็กลงเช่นเดียวกับความ ก้าวหน้าของเทคโนโลยีวงจรรวมเป็นผลให้ เมมส์ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน ในการนำมาใช้เป็นอุปกรณ์เซนเซอร์

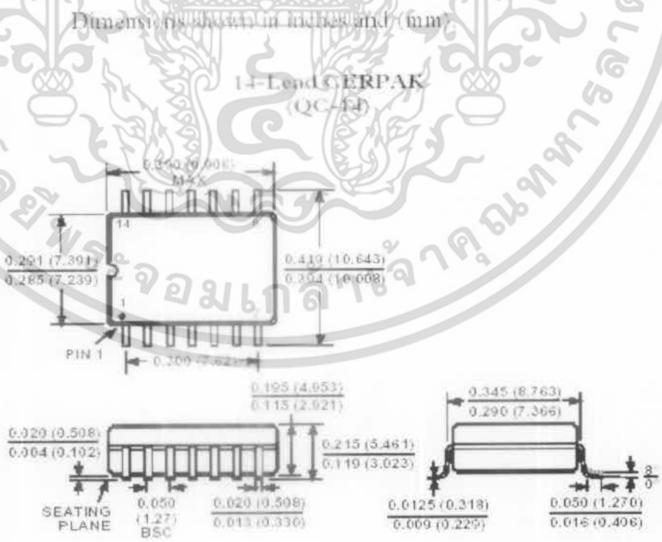
เทคโนโลยีของเมมส์ (MEMS) ในช่วงยุคหลังๆนั้นจะมีการพัฒนาจากที่เป็นอุปกรณ์เซนเซอร์ที่ไม่มีการเคลื่อนไหวมาเป็นแบบการเคลื่อนไหว โดยใช้การขยับของแขนหรือเฟืองถ้าเป็นเกียร์ก็มีเพลาที่หมุนได้ เทคโนโลยีของ เมมส์ (MEMS) นั้นเป็นชุดอุปกรณ์ขนาดเล็กที่รวมระบบกลไกเข้ากับระบบวงจรรวมทางอิเล็กทรอนิกส์ โดยทั้งสองระบบนี้วางอยู่บนฐานเดียวกัน ซึ่งโดยปกติก็คือ ซิลิกอนที่ใช้เป็นฐานสำหรับวงจรรวมอยู่แล้ว นอกจากนี้ในกระบวนการผลิตเมมส์ (MEMS) ยังมีกระบวนการในการผลิตที่ร่วมกับการผลิตวงจรรวมอื่นอีกด้วยโดยภาพรวมแล้วเมมส์(MEMS)มักนิยมใช้เป็นเซนเซอร์ขนาดเล็กหรือมินิเซนเซอร์ ตัวอย่างเช่นอุปกรณ์เซนเซอร์ที่ใช้ในการวัดแรงกระแทกขนาดเล็กในถุงลมนิรภัย ซึ่งทำให้ต้นทุนในการผลิตมีราคาที่ถูกลงกว่าแบบสมัยก่อนมากและยังมีเซนเซอร์อีกหลายแบบคือ เซนเซอร์วัดแรงดัน เซนเซอร์วัดกำลังแม่เหล็ก เซนเซอร์ตรวจจับสารเคมี นอกจากนั้น เมมส์ (MEMS) ยังถูกนำมาใช้งานด้านอื่นๆด้วย เช่น ใช้เป็นวาวล์ขนาดจิ๋ว กระจกสะท้อนขนาดจิ๋ว ปั้มขนาดจิ๋ว เป็นต้น

1. ลักษณะทางกายภาพของ เมมส์ (iMEMS ADXL202)

iMEMs Accelerometer ADXL202 ในส่วนของโครงงานนี้ใช้ชิฟไอซีแบบ Surface Mount เป็นชิฟไอซีขนาดเล็ก 14 ขา ADXL202 เป็น IC วัดความเร่งบนชิปเดี่ยวแบบ monolithic IC โดยตัวมันจะประกอบไปด้วย polysilicon surface-micromachined sensor และสัญญาณเอาต์พุตในแต่ละแกนถูกเปลี่ยนจากสัญญาณอนาลอกมาเป็นสัญญาณดิจิตอลที่สามารถใช้ระบบ timer / counter ของระบบ microprocessor ในการตรวจสอบได้โดย ADXL202 สามรถวัดความเร่งได้ทั้งแบบความเร่ง (positive) และความหน่วง (negative)ซึ่งมีลักษณะรูปร่างตัวถังดังรูป

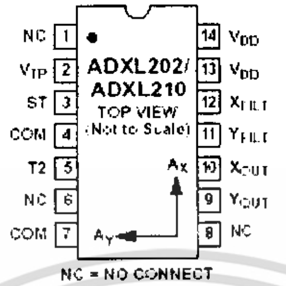


รูปที่ 2 ลักษณะภายในตัว ADXL202



รูปที่ 3 แสดงรูปร่างและตัวถังของ ADXL202

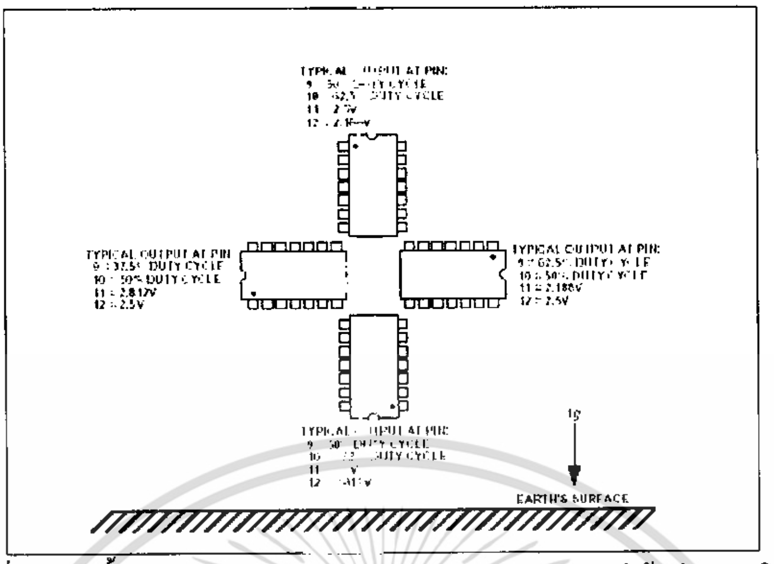
การใช้งานขาต่างๆ ของ ADXL202



รูปที่4 ลักษณะตำแหน่งขาของADXL202

Pin	Name	Description
1	NC	No Connect
2	Vtp	Test Point , Do Not Connect
3	ST	Self Test
4	COM	Common
5	T2	Connect Rset to Set T2 Period
6	NC	No Connect
7	COM	Common
8	NC	No Connect
9	Y_{OUT}	Y Axis Duty Cycle Output
10	X_{OUT}	X Axis Duty Cycle Output
11	Y_{FILT}	Connect capacitor for Y Filter
12	X_{FILT}	Connect capacitor for X Filter
13	VDD	+3V to +5.25V , Connect to 14
14	VDD	+3V to +5.25V , Connect to 13

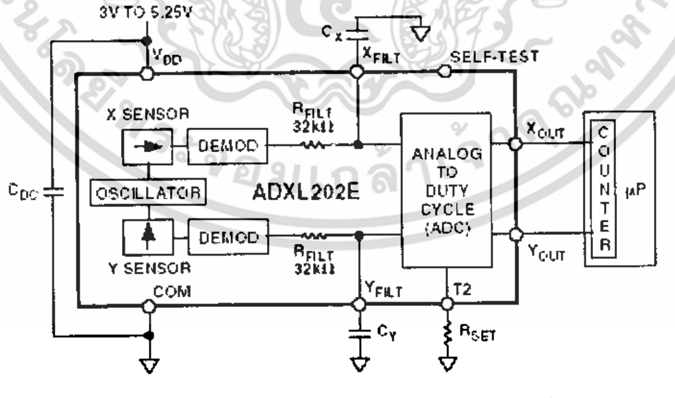
ตารางที่ 1 แสดงหน้าที่ขาการใช้งาน ของ ADXL202



รูปที่ 5 ค่าดิวตี้ไซเคิลต่อการตอบสนองของ ADXL202 กับแรงโน้มถ่วงของโลก

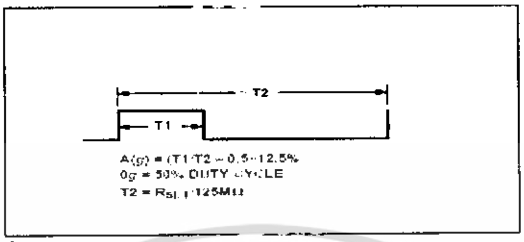
2. หลักการทำงานของ iMEMS ADXL202

ADXL202 เป็น accelerometer sensor ที่อยู่บนชิพไอซีเดี่ยว ที่มีต้นทุนต่ำ ใช้พลังงานในการทำงานต่ำ ซึ่งสามารถวัดความเร่งในการเคลื่อนที่ได้ 2 แกน โดยสามารถวัดความเร่งได้ในคปีนช่วงค่าคาบเวลา duty cycle และสามารถวัดความเร่งได้ทั้งแบบ static (e.g.,vibration) คือ แรงโน้มถ่วงของโลก และแบบ dynamic (e.g.,gravity) คือ แรงจากการเคลื่อนที่ การสั่นสะเทือน แสดงบล็อกการทำงานของ ADXL202 ดังรูป



รูปที่ 6 ฟังก์ชั่นบล็อกภายใน iMEMs Accelerometer ADXL 202

สัญญาณเอาต์พุตที่ได้ จาก ADXL202 จะให้ค่าออกมาเป็นสัญญาณดิจิตอล ทั้งในแนวแกน x และในแนวแกน y โดยสัญญาณที่ได้จะเป็นค่าดิวตี้ไซเคิล (duty cycle) ที่มีการเปลี่ยนแปลงค่าเป็นสัดส่วนกับอัตราเร่ง (acceleration) โดยสัญญาณที่ได้จะออกมาในแต่ละแกนจะเป็นค่าตามการเคลื่อนที่ในแนวแกนนั้น ๆ ดังรูปจะเป็นค่าสัญญาณเอาต์พุตที่สภาวะปกติ คือจะให้ค่าดิวตี้ไซเคิล (duty cycle) ออกมาเป็น 50 %



รูปที่ 7 แสดงค่าคาบเวลาเอาต์พุตของสัญญาณ ADXL202

T1 คือความยาวของช่วงเวลา " on " ของสัญญาณ

T2 คือความยาวของคาบสัญญาณ

Duty Cycle คือ T1/T2

เซ็นเซอร์เป็นโครงสร้างของตัวคาปาซิเตอร์ที่ถูกสร้างอยู่บนแผ่น silicon wafer โดยสปริงซิลิกอนถูกยึดติดอยู่บนแผ่นเวเฟอร์ ที่จัดไว้ต้านแรงจากความเร่งและ การเปลี่ยนแปลงของโครงสร้างจะถูกวัดโดยใช้ differential คาปาซิเตอร์ที่ประกอบด้วยแผ่นเพลททีอยู่กับที่และแผ่นเพลททีเคลื่อนที่โดยแผ่นเพลททีถูกยึดติดไว้สามารถหมุนได้ 180 องศาทำให้เอาต์พุตออกมาเป็นสัญญาณสี่เหลี่ยมที่เป็นอัตราส่วนของการเปลี่ยนแปลงความเร่งโดยความเร่งที่ 0 g จะให้เอาต์พุต Duty cycle 50% และในการใช้งานต้องต่อคาปาซิเตอร์ (Cdc) เพื่อช่วยในการลด noise ที่มาจากเพาว์เวอร์ซัพพลาย เราสามารถปรับค่าแบนด์วิดท์โดยใช้ค่า Cx และ Cy

$$F_{-3dB} = 1/(2\pi\,(32k\Omega) \times C_{(x,y)})$$

$$F_{-3dB} = 5\ \mu F / C_{(x,y)}$$

Bandwidth	Capacitor Value
10 Hz	0.47 μF
50 Hz	0.10 μF
100 Hz	0.05 μF
200 Hz	0.027 μF
500 Hz	0.01 μF
5 kHz	0.001 μF

ตารางที่ 2 แสดงการเลือกค่าตัวเก็บประจุ Cx และ Cy

การกำหนดค่าคาบเวลา (T2) ด้วยค่า Rset จากสมการ

$$T2 = Rset(\Omega) / 125\ M\Omega$$

โดยถ้าใช้ค่า Rset 125 kΩ จะทำให้ได้ค่าความถี่สัญญาณประมาณ 1 kHz ได้ค่าคาบเวลาเป็น 1 ms ดังจากตาราง

T2	Rset
1 ms	125 kΩ
2 ms	250 kΩ
5 ms	625 kΩ
10 ms	1.25 MΩ

ตารางที่ 3 แสดงการเลือกค่าความต้านทาน Rset

3. การประยุกต์ใช้งาน iMEMs Accelerometer

iMEMs Accelerometer ได้ถูกนำมาใช้ในการเซนเซอร์ตำแหน่งในระบบ navigation มานานแล้ว โดยในช่วงแรกๆนั้นจะเป็นการนำเอา Accelerometer มาใช้ร่วมกับ Gyroscopes เพื่อหาตำแหน่ง (Dead reckoning) โดยใช้สมการทางคณิตศาสตร์คือ

$$Position = Starting\ Position + (A \times t^2)/2$$

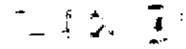
แต่สำหรับการเคลื่อนที่ความเร็วต่ำๆ แท้จริงของระบบนั้น ความยาวที่เกิดขึ้นจะไม่เหมาะสมกับเวลา เนื่องจากว่ามีสัญญาณไฟดีซีที่ไม่เหมาะสมเกิดขึ้นและในท้ายที่สุดแล้วความผิดพลาดที่เกิดก็จะมากขึ้นเรื่อยๆ เพื่อให้เข้าใจได้ง่ายขึ้นจะแสดงดังตัวอย่างคือ ถ้าหากว่าคนเราเดินที่ความเร็ว 5 km/h (1.39m/s) มากกว่า 5 นาที ค่าเฉลี่ยความเร็วที่ได้เป็น 416 m จะสามารถหาค่าได้

$$A_{avg} = \frac{2 \times Displacement}{t^2} = \frac{833}{300^2} =$$
$$0.00926\ m/s^2 = 0.944\ mg$$

ค่าสัมประสิทธิ์อุณภูมิของ ADXL 202 นั้นมีค่าอยู่ที่ ประมาณ 2 mg/°c และค่าอุณภูมิเบี่ยงเบนจะมีค่าอยู่ที่ 0.5 °c ในทุกๆ 5 นาทีและจะมีความคลาดเคลื่อนเพิ่มขึ้น 1 mg ในอนาคตมีแนวโน้มที่จะมีการพัฒนาเพื่อทำให้ค่าสัมประสิทธิ์อุณภูมิของ ADXL 202 มีค่าเป็น 0.06 °c ซึ่งจะทำให้ค่าของความคลาดเคลื่อนน้อยกว่า 1 mg

3.1.2 สวิตซ์ปรอท (Mercury switch)

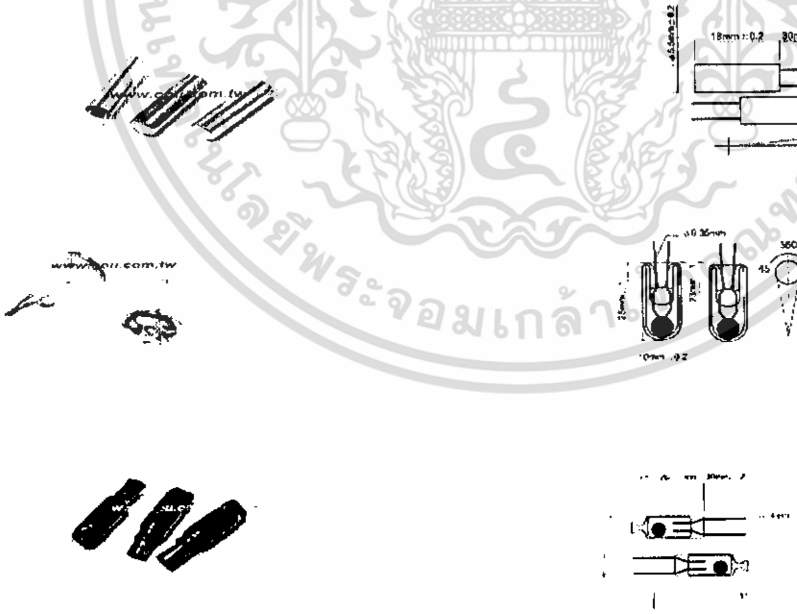
สวิตซ์ปรอทเป็นสวิตซ์ชนิดหนึ่งที่ทำงานเป็นอุปกรณ์เซนเซอร์การเคลื่อนที่หรือการเอียง โดยมีการใช้งานกันอย่างแพร่หลายและเป็นที่นิยมในการใช้เป็นตัววัดการเคลื่อนที่ เนื่องจากการใช้งานที่ง่ายสะดวกและไม่ต้องต่อไฟเลี้ยงในการทำงานโดยสามารถทำงานได้เหมือนสวิตซ์ทั่วๆ



รูปที่ 8 แสดงลักษณะสวิตซ์ปรอท

1. ส่วนประกอบของสวิตซ์ปรอท

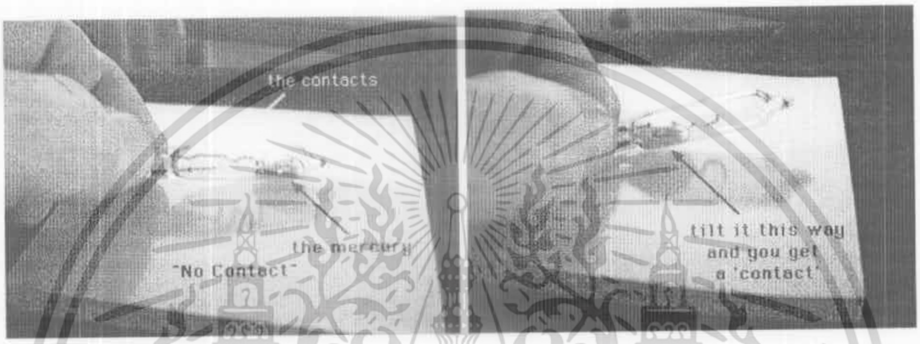
สวิตซ์ปรอท (Mercury switch) ประกอบด้วยหลอดแก้วซึ่งภายในบรรจุของเหลวที่นำไฟฟ้าหรือปรอทนั่นเองซึ่งปรอทที่บรรจุอยู่ข้างในนั้นมีขนาดเพียงเล็กน้อย และที่ปลายของหลอดแก้วจะมีขั้วไปไฟฟ้าที่ไม่ต่อถึงกันอยู่สองขั้ว แต่ในปัจจุบันนั้นสวิตซ์ปรอทมีลักษณะที่แตกต่างกันออกไป ขึ้นอยู่กับการนำไปใช้งาน ซึ่งสวิตซ์ปรอทนั้นสามารถออกแบบให้มีลักษณะที่พอเหมาะกับการใช้งานได้หลายแบบ ดังรูป เป็นลักษณะของสวิตซ์ปรอทแบบต่างๆ



รูปที่ 9 แสดงสวิตซ์ปรอทแบบต่างๆ

2. การทำงานของสวิตซ์ปรอท

เมื่อก่อนนั้นสวิตซ์ปรอทเคยมีการห้ามนำมาใช้งาน เนื่องจากมีความเสี่ยงต่ออันตรายที่อาจจะเกิดจากไอของสารปรอทที่อาจจะรั่วไหลมาได้ แต่ปัจจุบันนี้สารปรอทนั้นได้ถูกผลิตภายใต้การควบ คุมความปลอดภัยในการสัมผัสกับสารทางผิวหนัง จึงเป็นที่นิยมใช้งานกันทั่วไป ในการทำงานของสวิตซ์ปรอทนั้น จะอาศัยการเคลื่อนที่ของสารเหลวภายในหรือปรอทซึ่งเป็นสารที่นำไฟฟ้า เคลื่อนที่ไปยังขั้วสัมผัสของตัวนำทั้งสองจึงทำให้กระแสไฟฟ้าสามารถไหลผ่านได้ โดยการเคลื่อนที่ของสารเหลวนั้นจะเคลื่อนที่อยู่เพียงภายในหลอดแก้วเท่านั้น ดังรูป



รูปที่ 10 แสดงการทำงานของสวิตซ์ปรอท

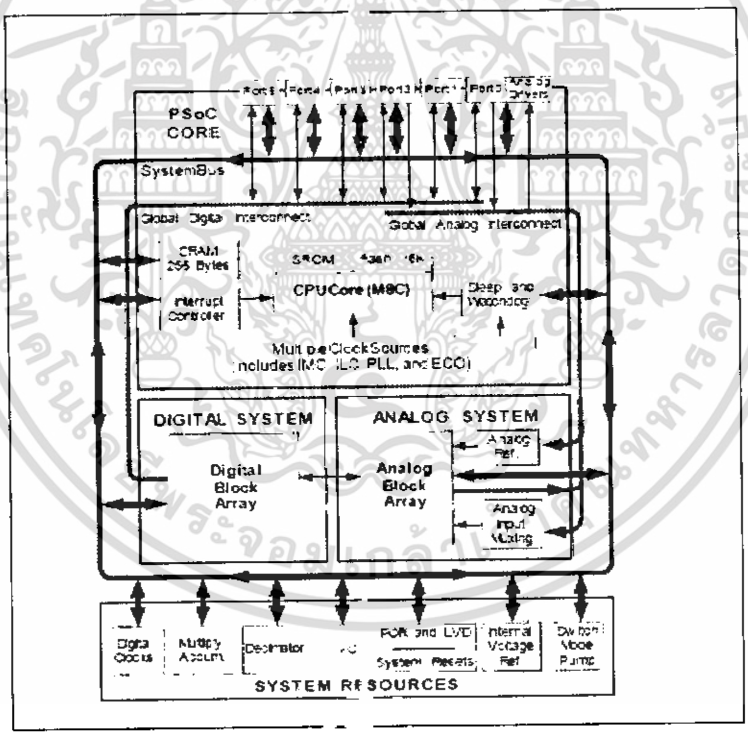
3. ข้อควรระวังในการใช้งานสวิตซ์ปรอท

เนื่องจากภายในของสวิตซ์ปรอทนั้นมีสารปรอทอยู่ซึ่งสารปรอทนี้เป็นพิษต่อร่างกาย ดังนั้นเมื่อสวิตซ์ปรอทนั้นชำรุดหรือพังเสียหายไม่ควรไปสัมผัสโดยตรง และควรนำไปทิ้งในที่ที่เหมาะสมเพื่อป้องกันความเป็นพิษของสารปรอทต่อบุคคลอื่นด้วย

3.2 ไมโครคอนโทรลเลอร์

3.2.1 PSoC MCU

โครงงานนี้เป็นการประยุกต์ใช้อุปกรณ์ควบคุมไมโครคอนโทรลเลอร์ PSoC(Programmable system on chip)ซึ่งเป็นไมโครคอนโทรลเลอร์ของ CypressMicrosystems โดยเป็นไมโครคอน โทรลเลอร์ที่มีการประมวลผลข้อมูลแบบ 8 บิต เหมือนไมโครคอนโทรลเลอร์ทั่วๆ ไป แต่ PSoc มีจุดเด่นหลายประการที่แตกต่างจากอุปกรณ์ไมโครคอนโทรลเลอร์ตระกูลอื่นๆคือ PSoC MCU จะทำการรวมเอาการออกแบบทั้งทางดิจิตอล และด้าน อนาลอก มาไว้ด้วยกันภายในตัว PSoC MCU ทำให้การออกแบบในการใช้งานทางด้านดิจิตอลและทางด้านอนาลอกสามารถทำได้ง่ายขึ้นและสะดวก สบายยิ่งขึ้นอีกทั้งยังทำให้การออกแบบในด้านต่างๆ มีขนาดที่เล็กลงโดยเฉพาะทางด้านอนาลอก ซึ่งมักจะมีขนาดค่อนข้างใหญ่ แต่เมื่อถูกรวมอยู่ภายใน PSoC MCU แล้วทำให้ไม่ต้องทำการต่อวงจรภายนอกเพิ่มขึ้นอีกจึงทำให้วงจรมีขนาดที่เล็กลง นอกจากนี้ยังมีฟังก์ชั่น In-System Serial Programming (ISSP) ที่สามารถทำการโปรแกรมซอร์สโค้ดที่ได้ออกแบบลงไปในหน่วยความจำโปรแกรม (Flash Memory) ภายในตัวชิพได้ ซึ่งช่วยให้การพัฒนาโปรแกรมโดยใช้ PSoC MCU มีความสะดวกสบายยิ่งขึ้น



รูปที่ 11 แสดงบล็อกการทำงานภายในของ PSoC MCU

จุดเด่นของ PSoC MCU เมื่อเทียบกับไมโครคอนโทรลเลอร์ชนิดอื่นๆมีดังนี้

1. **User Modules** สามารถเลือกใช้ทรัพยากรของระบบได้ตามที่ต้องการทั้งอนาลอก และดิจิตอล ซึ่งจะไม่ถูกจำกัดด้วยโครงสร้างทางฮาร์ดแวร์เหมือนกับไมโครคอนโทรลเลอร์ชนิดอื่นๆ

2. **API (Application Programming Interface)** สนับสนุนการพัฒนาโปรแกรมด้วยฟังก์ชั่น API ซึ่งช่วยให้ผู้พัฒนาโปรแกรมสามารถเขียนออกแบบโปรแกรมได้โดยง่าย

3. **ISRs(Interrupt Service Routines)**รองรับการทำงานแบบอินเตอร์ รัพท์

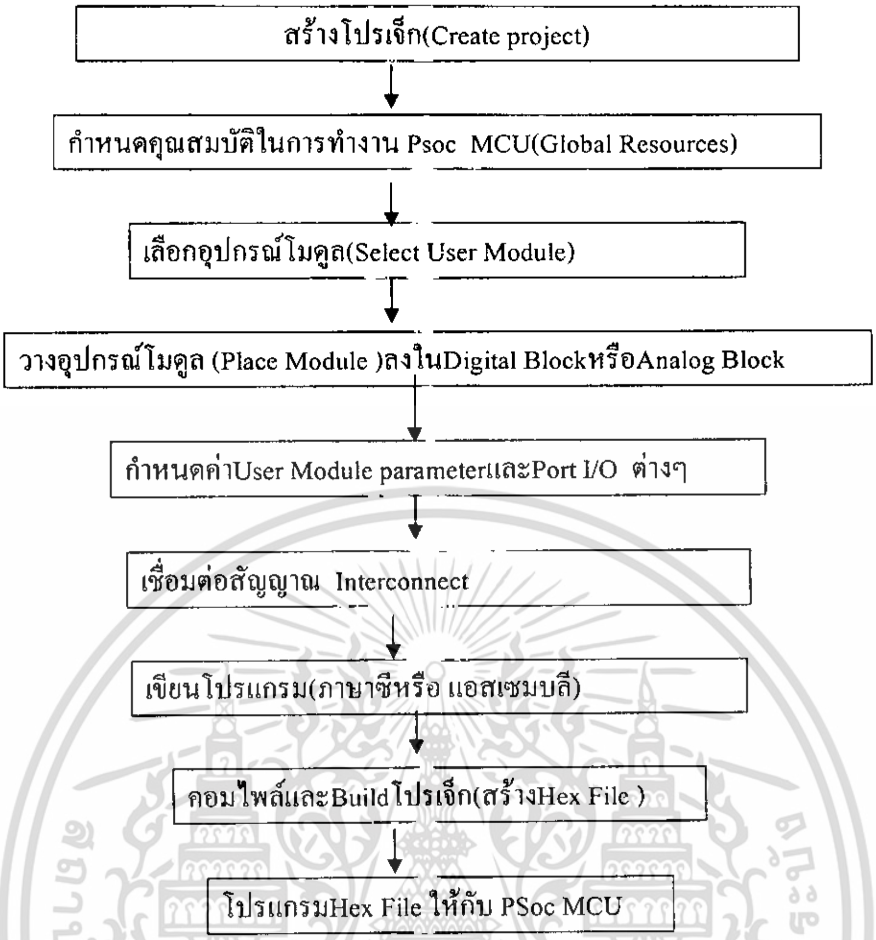
4. **Interconnect device interface** สามารถทำการเชื่อมต่อสัญญาณต่างๆ ได้อย่างอิสระ ไม่ถูกกำหนดตายตัวตามฮาร์ด์แวร์ เหมือนกับไมโครคอนโทรลเลอร์ชนิด อื่นๆ

จากที่กล่าวมาเบื้องต้นทั้งหมดเป็นเพียงคุณสมบัติคร่าวๆเท่านั้น ซึ่งคุณสมบัติอื่นๆที่เหลือก็จะคล้ายๆกับไมโครคอนโทรลเลอร์ชนิดอื่นๆ เช่น Sleep , watchdog, Power on Reset(POR) , SPI , UART และI^2C เป็นต้น จะเห็นได้ว่า PSoC MCU นั้นไม่ด้อยไปกว่า ไมโครคอนโทรลเลอร์ อื่น ๆ เลย

3.2.2 รูปแบบการใช้งานและการพัฒนา PSoC MCU

ไมโครคอนโทรลเลอร์ PSoC นั้นสนับสนุนระบบการทำงานทั้งทางด้าน ดิจิตอล และ อนา ลอก โดยในระบบของดิจิตอล (Digital System) และ อนาลอก (Analog System) ได้ถูกออกแบบเป็นบล็อกโมดูลซึ่งจะเรียกว่า บล็อกดิจิตอล (Digital Blocks) และ บล็อกอนาลอก (Analog Blocks) โดยบล็อกเหล่านี้จะรองรับการนำเอาโมดูลต่างๆ มาใช้งานเปรียบเสมือนกับเป็นพื้นที่ว่างๆสำหรับต่อจิ๊กซอ โดยชิ้นส่วนของจิ๊กซอก็คือโมดูลต่างๆ เช่น ADC , DAC , I2C , PWM , UART , SPI เป็นต้น โดยผู้ใช้สามารถกำหนดได้เองว่าจะนำเอาโมดูลใดมาใช้งานและนอกจากนี้ผู้ใช้งานยังสามารถกำหนด การเชื่อมต่อสัญญาณต่างๆ (Programmable Interconnect) ภายในได้เองอีกด้วย เสมือนกับว่าผู้ใช้งานสามารถทำการออกแบบได้เองตั้งแต่ ฮารด์แวร์ ไปจนถึง ซอฟแวร์ ซึ่งถือได้ว่าเป็นความสามารถหนึ่งที่เหนือกว่าไมโครคอนโทรลเลอร์ชนิดอื่นๆที่ทรัพยากรทุกอย่างถูกกำหนดไว้ตายตัวไม่สามารถเปลี่ยน แปลงได้

ภาษาที่ใช้ในการออกแบบพัฒนาการทำงานของ PSoC MCUปัจจุบันจะมีอยู่ด้วยกัน 2 ภาษา คือ ภาษาแอสเซมบลีและ ภาษาซี เนื่องจากการพัฒนาโปรแกรมของ PSoC MCU ส่วนใหญ่จะทำโดยการเรียกใช้งานฟังก์ชั่นAPIและการกำหนดคุณสมบัติต่างๆเช่นความถี่สัญญาณนาฬิกาSleep , Watch dog , Supply Voltage และอื่นๆรวมทั้งยังสามารถทำได้จากหน้าต่าง Device Editor ของซอร์ฟแวร์ Psoc Designer ทำให้เราไม่จำเป็นต้องทราบรีจิสเตอร์ต่างๆ มากนักจึงจะไม่ขอกล่าวถึงรายละเอียดในการใช้งานรีจิสเตอร์มากนัก แต่จะกล่าวถึงเฉพาะในส่วนที่เกี่ยวข้องกับ โครงงานนี้เท่านั้น และจะอธิบายการใช้งานในการพัฒนาโปรแกรมเป็นขั้นตอนดังต่อไปนี้ คือ



รูปที่ 12 แสดงลำดับขั้นตอนการพัฒนาโปรแกรมให้กับ PSoc MCU ด้วย PSoc Designer แบบคร่าวๆ

3.2.3 คุณสมบัติของ PSoc MCU ตระกูล CY8C27x43

1. สถาปัตยกรรมแบบ Harvard Architecture Processor
2. ความเร็วของ M8C Processor สูงสุด 24 MHz
3. 8x8 Multiply, 32Bit Accumulate
4. Low Power at High Speed
5. ทำงานในช่วงแรงดัน 3.0 Vถึง 5.25 V
6. สามารถทำงานที่แรงดันต่ำสุดที่ 1.0V โดยใช้วงจร Switch Mode Pump (SMP)
7. 12 Analog PSoc Blocks รองรับการใช้งานทางด้านอนาลอก เช่น
    - ADCs ความละเอียดสูงสุดถึง 14 บิต
    - DACs ความละเอียดสูงสุดถึง 9 บิต
    - วงจรเกณฑ์การขยาย (Programmable Gain Amplifier)

- วงจรฟิลเตอร์และวงจรคอมพาราเตอร์Programmable Filters and Comparater

8. 8 Digital PSoc Blocks รองรับการใช้งานทางดิจิตอล
   - Timers, Counters และ PWMs ขนาด 8 ถึง 32 บิต
   - CRC และ PRS โมดูล
   - UARTs แบบ Full-Duplex สูงสุด 2 ช่อง
   - SPI โมดูลเป็นได้ทั้งแบบ Master และ Slave
   - ดิจิตอลบล็อกต่างๆสามารถเชื่อมต่อไปยังขาสัญญาณ GPIO ได้ทุกขาสัญญาณ
9. สามารถกำหนดขนาดความถี่ของสัญญาณนาฬิกาภายในได้หลายระดับ
10. สัญญาณนาฬิกาภายในขนาด 24/48 MHz ค่าความคลาดเคลื่อน +/- 2.5 %
11. สามารถเลือกแหล่งกำเนิดสัญญาณ 24/48 MHz จากออสซิลเลเตอร์ 32 kHz ภายใน หรือภายนอกได้
12. สามารถรับสัญญาณออสซิลเลเตอร์จากภายนอกได้สูงสุด 24 MHz
13. มีแหล่งกำเนิดสัญญาณนาฬิกาภายในให้กับ Watchdog และ Sleep
14. หน่วยความจำโปรแกรมแบบ Flash ขนาด 16K Byte สามารถ Erase/ Write ได้ถึง 50,00 ครั้ง
15. หน่วยความจำข้อมูล SRAM ขนาด 256 Byte
16. ฟังก์ชั่นการโปรแกรมภายใน ISSP (In-System Serial Programming)
17. สามารถเปลี่ยนแปลงข้อมูล Flash Memory เฉพาะบางส่วนได้
18. สามารถตั้งค่าระบบป้องกันข้อมูลได้ (Flash Security)
19. หน่วยความจำ EEPROM (ใช้จาก Flash Memory)
20. ขาสัญญาณ GPIO จำนวน 24 ขาสัญญาณ
21. สามารถกำหนดคุณสมบัติของขาสัญญาณต่างๆ ได้ (GPIO Pin Configurations)
22. GPIO สามารถจ่ายกระแสสูงสุดได้ถึง 24 mA
23. สามารถกำหนด Drive Mode ของสัญญาณ GPIO ได้ เช่น Pull up, Pull down, HighZ,
24. ขาสัญญาณอินพุตอนาลอกสูงสุด 12 สัญญาณ (จากขาสัญญาณของ GPIO)
25. ขาสัญญาณอนาลอกเอาท์พุตสูงสุด 4 ขาสัญญาณ(จากขาสัญญาณGPIO)จ่ายกระแสได้ถึง 40 mA
26. สามารถกำหนดการอินเตอร์รัพท์ได้ทุกขาสัญญาณของ GPIO
27. I2C โหมด Slave, Master และ Multi-Master ความเร็วสูงสุด 400 kHz
28. Watchdog และ Sleep Timer
29. สามารถตั้งค่าระดับการตรวจจับแรงดันต่ำได้ (Low Voltage Detection)
30. Integrated Supervisory Circuit
31. On-Chip Precision Voltage Reference

จะเห็นได้ว่า PSoc MCU เป็นไมโครคอนโทรลเลอร์ที่มีคุณสมบัติที่ถือได้ว่าครบถ้วนมาก ไม่ได้ด้อยไปกว่าไมโครคอนโทรลเลอร์ตระกูลอื่นๆเลย ในการที่มีคุณสมบัติมากมายอาจถูกมองว่าการออกแบบยุ่งยาก แต่ในความจริงแล้ว PSoc MCU ไม่ได้ออกแบบยุ่งยากอย่างที่คิดเนื่องจากสา มารถใช้ซอฟแวร์ PSoc Designerในการออกแบบได้ซึ่งซอฟแวร์ดังกล่าวนี้จะมีการออกแบบเป็นลักษณะของกราฟิกหรือเป็นแบบวิสชวล(Visual)ส่วนในการเขียนโปรแกรมก็จะเป็นลักษณะการใช้งานฟังก์ชั่น API (Application Programming Interfce) ที่ PSoc Designer ได้จัดเตรียมไว้ให้แล้ว ทำให้ลดความยุ่งยากไปได้มากเลยทีเดียว

3.2.4 ฟังก์ชั่นและโครงสร้างของไมโครคอนโทรลเลอร์ PSoc

ไมโครคอนโทรลเลอร์ PSoc นั้นมีฟังก์ชั่นการใช้งานมากมายหลายฟังก์ชั่นด้วยกันในหัวข้อนี้จะขอกล่าวถึงเฉพาะฟังก์ชั่นและโครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PSoc ที่เกี่ยวข้องกับโครงงานนี้เท่านั้น โดยในส่วนที่โครงงานนี้จะขอกล่าวถึงโครงสร้างภายในเพียงบางส่วนของไมโคร คอนโทรลเลอร์ PSoc เท่านั้น คือ

PSoc Core

เป็นส่วนหลักของการประมวลผล ซึ่งจะดูแลส่วนต่างๆ เช่น การประมวลผลคำสั่ง , การจัดเก็บข้อมูลในหน่วยความจำ SRAM , ควบคุมการอินเตอร์รัพท์ , Sleep, Watchdog timers และการเลือกแหล่งสัญญาณนาฬิกา (Clock sources) เป็นต้นซึ่งจะเรียกส่วนที่จัดการส่วนต่างๆ เหล่านี้ว่า M8C ซึ่งเป็นสถาปัตยกรรมไมโครโปรเซสเซอร์ 8 บิต แบบ Harvard นอกจากนี้ภายใน Psoc Core ยังมีหน่วยความจำ SROM และ Flash Memory สำหรับใช้เก็บโปรแกรมคำสั่งอีกด้วย

Digital System

เป็นระบบของดิจิตอลบล็อกที่วางอยู่ในรูปแบบของบล็อกอาเรย์ (array block) ในส่วนของ DIGITAL SYSTEM การวางของบล็อกอาเรย์จะวางแถวละ 4 บล็อก สำหรับ CY8C27443 จะมีอาเรย์บล็อกจำนวน 2 แถวรวมแล้วจะมีทั้งหมด 8 ดิจิตอลบล็อก ส่วน PSoc MCU เบอร์อื่น ๆ จะมีขนาดแตกต่างกันออกไปโดยบล็อกเหล่านี้ก็จะรองรับการใช้งานในส่วนของ โมดูลดิจิตอลต่างๆ

Analog System

ระบบของอนาลอกจะประกอบไปด้วยส่วนต่างๆ คือ อนาลอกบล็อก (Analog Blocks), แรงดันอ้างอิงอนาลอก (Analog Ref) และวงจรเลือกสัญญาณอินพุตอนาลอก (Analog Input Muxing) โดยอนาลอกบล็อกจะมีอยู่ 2 ประเภทด้วยกัน คือ CT (Continuous Time) และ SC (Switched Capacitor) ซึ่งอนาลอกบล็อกจะถูกจัดเรียงเป็น 4 คอร์ลัม (Columm) ในหนึ่งคอร์ลัมก็จะประกอบด้วย CT หนึ่งบล็อกและ SC อีกสองบล็อก ซึ่งบล็อกเหล่านี้จะมีไว้สำหรับรองรับการทำงานของโมดูลอนาลอกโดยการจัดวางโมดูลลงในบล็อกต่างๆนั้น ก็ขึ้นอยู่กับความเหมาะสมของแต่ละโมดูล

System Resources

System Resources ก็คือ ฮาร์ดแวร์ต่างๆ ของระบบ เช่น Digital clocks , Multiply accumulate (MAC), Decimator , I^{2}C , System reset , Internal voltage reference , Switch mode pump (SPM) เป็นต้น

หน่วยความจำ (Memory)

หน่วยความจำเป็นองค์ประกอบหนึ่งที่มีความจำเป็นต่อการทำงานของไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ โดยจะมีการแบ่งหน่วยความจำออกเป็นประเภทต่างๆ ตามคุณสมบัติและการใช้งาน ซึ่งในส่วนของสถาปัตยกรรม M8C ก็จะแบ่งหน่วยความจำออกเป็น 3 ส่วน คือ ROM, RAM และ Register

ขาสัญญาณอินพุต/เอาต์พุต (GPIO : General Purpose IO)

ในหัวข้อนี้จะกล่าวถึงคุณสมบัติทางด้านขาสัญญาณ I/O (Input/Output) ของ PSoc MCU ซึ่งขาสัญญาณของ PSoc MCU นั้นสามารถทำงานได้ทั้งในส่วนของขาสัญญาณดิจิตอล และขาสัญญาณอนาลอกซึ่งในการใช้งานจะผ่านทางรีจีสเตอร์ต่างๆ

Address	Name	Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
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 4 แสดงขาสัญญาณอินพุต/เอาต์พุต GPIO

- หมายเหตุ เนื่องจากในการพัฒนาจริง จะอาศัยซอร์ฟแวร์ PSoC Designer ในการออกแบบ ทำให้เราไม่จำเป็นต้องทราบรายละเอียดของรีจีสเตอร์ต่างๆ เหล่านี้มากนัก

ขาสัญญาณอินพุต/เอาต์พุต หรือ ขาสัญญาณ GPIO ของไมโครคอนโทรลเลอร์ PSoC แต่ละขาสัญญาณนอกจากการทำงานในโหมด I/O ปกติแล้ว บางขาสัญญาณยังสามารถทำงานในหน้าที่อื่นๆ ได้อีกด้วย ดังเช่น

ชื่อขาสัญญาณ	คำอธิบาย	Input / Output
SMP	Switch Mode Pump	Power
Vdd	Supply Voltage	Power
Vss	Ground	Power
XRES	External Reset (Active High)	Input / Output
P0[0]-P0[1]	Port 0 [0],Port 0[1], Analog Input	Input / Output
P0[2]-P0[5]	Port 0[2],[3],[4],[5],Analog Input/Output	Input / Output
P0[6]-P0[7]	Port 0[6],[7], Analog Input	Input / Output
P1[0]	Port 1[0], XTAL Out/SDATA /I2C SDA	Input / Output
P1[1]	Port 1[1],XTAL In/SCLK/ I2C SCL	Input / Output
P1[2]	Port 1[2]	Input / Output
P1[3]	Port 1[3]	Input / Output
P1[4]	Port 1[4], EXTCLK	Input / Output
P1[5]	Port 1[5], I2C SDA	Input / Output
P1[6]	Port 1[6]	Input / Output
P1[7]	Port 1[7], I2C SCL	Input / Output
P2[0]- P2[3]	Port2[0],[1],[2],[3],Non-Multiplexed AnalogInput (Switch capacitor)	Input / Output
P2[4]	Port2[4], External AGND	Input / Output
P2[5]	Port2[5]	Input / Output
P2[6]	Port2[6],External VREF	Input / Output
P2[7]	Port2[7]	Input / Output
P3[0] –P3[7]	Port3[0],[1],[2],[3],[4],[5],[6],[7]	Input / Output
P4[0]- P4[7]	Port4[0],[1],[2],[3],[4],[5],[6],[7]	Input / Output
P5[0]- P5[3]	Port5[0],[1],[2],[3]	Input / Output

ตารางที่ 5 แสดงหน้าที่ขาสัญญาณต่างๆ ของ Cy8C27443

จะเห็นว่าโครสร้างขาสัญญาณ GPIO ของ PSoc MCU มีความซับซ้อนมากทั้งนี้เพราะ PSoc MCU ได้ออกแบบโครสร้างขาสัญญาณ ให้สามารถทำงานได้ในหลากหลายคุณสมบัติ โดยผู้ใช้งาน (User) สามารถกำหนดความต้องการในการใช้งานได้ โดยเราสามารถกำหนดคุณสมบัติการทำงานของขาสัญญาณในโหมดต่างๆ ได้ดังนี้

- Pull Down (Resistive pull down) โหมดนี้ขาสัญญาณ I/O จะถูกต่อผ่านตัวต้านทานลงกราวด์
- Strong (Strong Drive) โหมดนี้เหมาะสำหรับใช้งานเป็นเอาต์พุตดิจิตอล
- High Z (High Impedance) โหมดนี้ที่ขาสัญญาณ I/O จะมีความต้านทานสูงเหมาะสำหรับการใช้งานเป็นอินพุต
- Pull Up (Resistive Pull Up) โหมดนี้ขาสัญญาณ I/O จะถูกต่อผ่านตัวต้านทานไปยัง Vcc
- Open Drain High สถานะของสัญญาณ I/O เป็นแบบ Open Drain High
- Strong Slow (Slow strong drive) สถานะสัญญาณ I/O เป็นแบบ Strong Slow
- High Z Analog เป็นสถานะความต้านทานสูงแบบอนาลอก ซึ่งจะเป็นค่าสถานะเริ่มต้น (Default) หลังจากเกิดการ รีเซต (reset state)

DM2	DM1	DM0	Drive Mode	Data = 0	Data = 1
0	0	0	Resistive Pull Down	Resistive	Strong
0	0	1	Strong Drive	Strong	Strong
0	1	0	High Impedance	HI-Z	HI-Z
0	1	1	Resistive Pull Up	Strong	Resistive
1	0	0	Open Drain, Drive High	HI-Z	Strong(Slow)
1	0	1	Slow Strong Drive	Strong(Slow)	Strong(Slow)
1	1	0	High Impedance Analog	HI-Z	HI-Z
1	1	1	Open Drain, Drive Low	Strong(Slow)	HI-Z

ตารางที่ 6 การกำหนด Drive Mode ของ GPIO โดยผ่านรีจีสเตอร์ PRTxDMx[2:0]

การกำหนดคุณสมบัติของขาสัญญาณเหล่านี้ สามารถทำได้สองวิธี คือ การกำหนดที่ค่ารีจีสเตอร์ (PRTxDMx) และอีกวิธีก็คือการกำหนดโดยใช้โปรแกรม PSoC Designer ตรงส่วนของ Device Editor Configuration ซึ่งจะเรียกว่าการกำหนด Configure I/O Pins ซึ่งจะช่วยลดความยุ่งยากในการเขียนโปรแกรมลงได้เป็นอย่างดี

ขาสัญญาณ I/O ของ PSoc MCU จะประกอบไปด้วย บัฟเฟอร์อินพุต และ วงจรขับทางด้านเอาต์พุตโดยขาสัญญาณ I/O เหล่านี้จะถูกจัดไว้เป็นพอร์ต ซึ่งปกติ 1 พอร์ต จะมีทั้งหมด 8 บิต แต่จะมีบางกรณีที่พอร์ตนั้นมีขาสัญญาณไม่ถึง 8 บิต ขาสัญญาณ I/O ต่างๆ เหล่านี้สามารถทำงานในลักษณะต่างๆ ดังนี้

* Digital IO : เป็นขาสัญญาณดิจิตอล อินพุต/ เอาต์พุต สามารถควบคุมการทำงานได้โดยการผ่านค่าให้กับรีจีสเตอร์ PRTxDR

* Global IO : เป็นขาสัญญาณดิจิตอล อินพุต/ เอาต์พุต ที่เชื่อมโยงระหว่าง Digital PSoC block

* Analog IO : เป็นขาสัญญาณ อินพุต/ เอาต์พุต ที่เชื่อมโยงระหว่าง Analog PSoC block

3.2.4 การใช้งาน SPIM (Serial Peripheral Interconnect Master)

ในโครงงานนี้เป็นการนำโมดูลในส่วนของ SPIM มาใช้ในการติดต่อกับการแสดงผลบนหน้าจอ LCD NOKIA 3310 ที่เป็นหน้าจอกราฟฟิกขนาดความละเอียด 48x 84 จุดซึ่งจะกล่าวถึงในหัวข้อต่อไป ในส่วนของโมดูล SPIM (Serial Peripheral Interconnect Master) จะขออธิบายรายละเอียดดังต่อไปนี้

โมดูล SPIM คือ โมดูลการสื่อสารอนุกรมที่มีโปรโตคอลการสื่อสารเป็นแบบ SPI โดยจะเป็นการส่งแบบเข้าจังหวะ 8 บิตข้อมูลฟูลดูเพล็กซ์ (Full-duplex) โมดูลนี้จะทำงานเป็นแบบมาสเตอร์ซึ่ง สามารถที่จะนำไปต่อเข้ากับอุปกรณ์ SPI ที่เป็นสลาฟมากกว่าหนึ่งตัว ขึ้นอยู่กับการนำไปใช้งานโดยภายในโมดูล SPIM จะประกอบไปด้วย Tx Buffer, Rx Buffer, Control, และ Shift registers การใช้งานโมดูลจะเรียกใช้งานผ่านฟังก์ชั่น API โดยค่าเริ่มต้นของโมดูล SPIM จะถูกเซตมีการส่งข้อมูลแบบส่งบิตนัยสำคัญต่ำสุดก่อน (LSB First) และจะสนับสนุนการทำงานของสัญญาณในโหมดต่างๆ ตั้งแต่ โหมด 0,1,2 และ 3 ซึ่งควรจะมีการเซตโหมดของสัญญาณนาฬิการะหว่างอุปกรณ์ SPI Master และอุปกรณ์ SPI Slave ให้มีสัญญาณนาฬิกาเป็นโหมดเดียวกัน โดยโหมดการทำงานของ SPI จะเป็นดังตารางที่

SPI Mode			
Mode	**SCLK Edge Performig Data Latch**	**Clock Polarity**	**Notes**
0	Leading	Non-inverting	Leading edge latches data . Data changes on trailing edge of clock.
1	Leading	Inverted	
2	Trailing	Non-inverting	Trailing edge latches data . Data changes on leading edge .
3	Trailing	Inverted	

ตารางที่ 7 แสดงโหมดการทำงานของ ฟังก์ชั่น SPI

ขาสัญญาณการเชื่อมต่อ

- **SCLK** คือ ขาสัญญาณนาฬิกาที่ให้จังหวะในการรับส่งข้อมูลของ SPI
- **MOSI** (Master-Out-Slave-In data) คือ ขาสัญญาณเอาต์พุตข้อมูลของตัวอุปกรณ์มาสเตอร์ ซึ่งจะไปเป็นขาสัญญาณอินพุตของอุปกรณ์สลาฟ

คุณสมบัติของโมดูล SPIM (Serial Peripheral Interconnect Master)

- สนับสนุนการเชื่อมต่อในโหมดมาสเตอร์ ด้วยโปรโตคอลการสื่อสารแบบ SPIM (Serial Peripheral Interconnect Master)
- สนับสนุนโหมดสัญญาณนาฬิกาของ SPI คือ โหมด 0,1,2 และ 3
- สามารถกำหนดแหล่งสัญญาณนาฬิกาได้
- สามารถเลือกการเชื่อมต่อเอาต์พุตของสัญญาณ MOSI และ SCLK ได้
- สามารถสร้างสัญญาณอินเตอร์รัพท์ได้
- ความเร็วในการรับส่งข้อมูลสูงสุด (Maximum bit rate) 12 MHz

การใช้งานฟังก์ชั่น API ของโมดูล SPIM

การใช้งานโมดูลจะกระทำผ่านการเรียกใช้ฟังก์ชั่น API ซึ่งการใช้งานฟังก์ชั่นนี้อาจมีผลทำให้ค่าของรีจีสเตอร์ A และ X มีการเปลี่ยนแปลงไปด้วยโดยจะมีฟังก์ชั่นต่างๆ ให้ใช้งานดังต่อไปนี้

SPIM_Start

ฟังก์ชั่นการกำหนดโหมดการทำงานของ SPI พร้อมทั้งเปิดการทำงานของโมดูล โดยการกำหนดโหมดการทำงานจะต้องผ่านค่าให้กับฟังก์ชั่นผ่านทางรีจีสเตอร์ A ซึ่งจะมีโหมดต่างๆ ตามตารางต่อไปนี้

Symbolic Name	Value	ความหมาย
SPIM_MODE_0	0x00	การทำงานของ SPI ในโหมด 0
SPIM_MODE_1	0x02	การทำงานของ SPI ในโหมด 1
SPIM_MODE_2	0x04	การทำงานของ SPI ในโหมด 2
SPIM_MODE_3	0x06	การทำงานของ SPI ในโหมด 3
SPIM_LSB_FIRST	0x80	ส่งบิตข้อมูลทางด้านบิตต่ำก่อน
SPIM_MSB_FIRST	0x00	ส่งบิตข้อมูลทางด้านบิตสูงก่อน

ตารางที่ 8 แสดงค่าของโหมดการทำงานต่างๆ ของ SPI

ตัวอย่าง

Assembly :

```
Mov A,SPIM_MODE_2|SPIM_LSB_FIRST   ; เลือกโหมด 2 ส่งข้อมูลทางด้านบิตต่ำก่อน
```

C Prototype :

```
Void    SPIM_Start (BYTE bConfiguration )
```

- หมายเหตุเราสามารถกำหนดโหมดการทำงานร่วมกับการเลือกรูปแบบการส่ง (SPIM_LSB_FIRST) *หรือ*(SPIM_MSB FIRST) โดยใช้การ OR โดยใช้เครื่องหมาย " | " ขั้นกลางดังตัวอย่าง

SPIM_STOP

ฟังก์ชั่นการปิดการทำงานของโมดูล SPIM ฟังก์ชั่นนี้จะไปเคลียร์ค่าในบิตEnableของรีจีสเตอร์ Control ทำให้โมดูล SPIM หยุดการทำงาน

ตัวอย่าง

Assembly :

```
Call SPIM_Stop
```

C Prototype :

```
Void SPIM_Stop(void)
```

SPIM_EnableInt

ฟังก์ชั่นเปิดการอินเตอร์รพท์ของ SPIM ตามเงื่อนไขของการอินเตอร์รัพท์

ตัวอย่าง

Assembly :

```
Call SPIM_EnableInt
```

C Prototype :

```
Void SPIM_EnableInt(void)
```

SPIM_DisableInt

ปิดการอิเตอร์รัพท์ของ SPIM

ตัวอย่าง

Assembly :

```
Call SPIM_DisableInt
```

C Prototype :

```
Void SPIM_DisableInt(void)
```

SPIM_SendTxData

ส่งข้อมูลแบบ SPI ไปยังอุปกรณ์สลาฟที่ต่อร่วมด้วย

ตัวอย่าง

Assembly :

```
Mov A,bSPIMData
Call SPIM_SendTxData
```

C Prototype :

```
BOOL SPIM_SendTxData(BYTE bSPIMData)
```

SPIM_bReadRxData

ฟังก์ชั่นการอ่านรับค่าจากไบต์ข้อมูลจากอุปกรณ์สลาฟ โดยควรทำการเช็คสถานะของบัฟเฟอร์ (Rx Buffer) ก่อนว่าว่างหรือไม่ ก่อนที่จะทำการเรียกใช้ฟังก์ชั่นนี้

ตัวอย่าง

Assembly :

```
Call SPIM_bReadRxData
Mov bRxData, A
```

C Prototype :

```
BOOL SPIM_bReadRxData(void)
```

SPIM_bReadStatus

เป็นคำสั่งอ่านค่าสถานะจากรีจิสเตอร์ควบคุม (ControlRegister) ของโมดูล SPIM รีจิสเตอร์ จะประกอบด้วยสถานการณ์ทำงานต่างๆของโมดูลโดยค่าสถานะต่างๆ จะรีเทิร์นค่าออกมาผ่านทางรีจิสเตอร์ A ซึ่งค่าต่างๆ จะเป็นดังตารางต่อไปนี้

โหมดการทำงานของ SPI

Symbolic Name	Value	ความหมาย
SPIM_DONE	0x20	การทำงานของ SPIM เสร็จสิ้น
SPIM_RX_OVERRUN	0x40	เกิดการ Overrun Error ของการรับข้อมูล
SPIM_TX_BUFFER_EMPTY	0x10	บัฟเฟอร์ของการส่งว่าง
SPIM_RX_BUFFER_FULL	0x08	บัฟเฟอร์การรับข้อมูลเต็ม

ตารางที่ 9 แสดงค่าสถานะรีจิสเตอร์ของการข้อมูล ของฟังก์ชั่น SPI

สามารถใช้การ OR โดยใช้เครื่องหมาย ‘ ’ เพื่อเพิ่มเงื่อนไขการตรวจสอบได้ ดังเช่น ตัวอย่างต่อไปนี้

ตัวอย่าง

Assembly :

```
Call SPIM_bReadStatus
And A, SPIM_DONE | SPIM_RX_BUFFER_FULL
Jnz SpimCompleteGetRxData
```

C Prototype :

BYTE SPIM_bReadStatus(void)

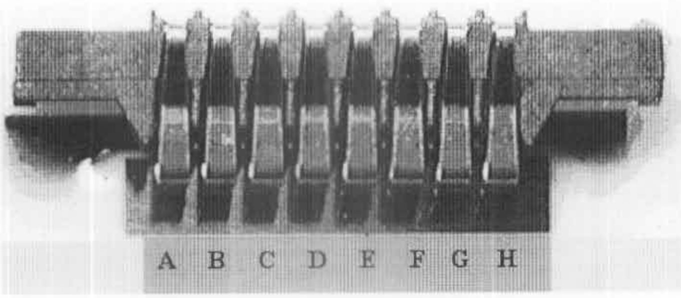
- หมายเหตุ บิตสถานะต่างๆ ในรีจิสเตอร์ควบคุม (Control register) จะถูกเคลียร์ หลังเสร็จสิ้นฟังก์ชั่นนี้

3.3 หน้าจอแสดงผล LCD Nokia 3310

3.3.1 ลักษณะทางกายภาพของ หน้าจอ LCD

โครงงานนี้ใช้หน้าจอแสดงผลแบบกราฟฟิกที่มีขนาดเล็กซึ่งเป็นหน้าจอ LCD ของ Nokia 3310 ซึ่งใช้ชิปไอซีในการเชื่อมต่อเบอร์ PCD8544 ซึ่งชิฟไอซีนี้จะประกอบติดมากับหน้าจอโดยตรง PCD 8544 ออกแบบมาเพื่อควบคุมการแสดงผลข้อมูลบนจอ LCD Display ขนาด 48 row x84 column โดยจัดเตรียมฟังก์ชันที่จำเป็นในการทำงานไว้ในชิพเดี่ยวต้องการอุปกรณ์ในการต่อเพิ่มน้อยประหยัดพลังงานมีขาใช้งานทั้งหมด 8 ขา ซึ่งมีลักษณะเป็นแบบ Surface Mount

รูปที่ 13 แสดงลักษณะหน้าจอแสดงผล LCD ของ Nokia 3310



รูปที่ 14 แสดงตำแหน่งขา LCD ของ Nokia 3310

ตำแหน่งขา	สัญลักษณ์	รายละเอียด
A	VDD	ขารับแรงดัน 2.7 - 3.3 V
B	SCLK	ขารับสัญญาณนาฬิกา
C	SDIN	ขารับข้อมูลอนุกรม
D	D/C'	ขาควบคุมการเขียนคำสั่งหรือข้อมูล
E	SCE'	ขาเริ่มการทำงานของ LCD
F	GND	กราวน์
G	VOUT	ขาปรับความสว่างของจอ
H	RES'	ขารีเซ็ต

ตารางที่ 10 แสดงหน้าที่ขาต่างๆ ของ LCD

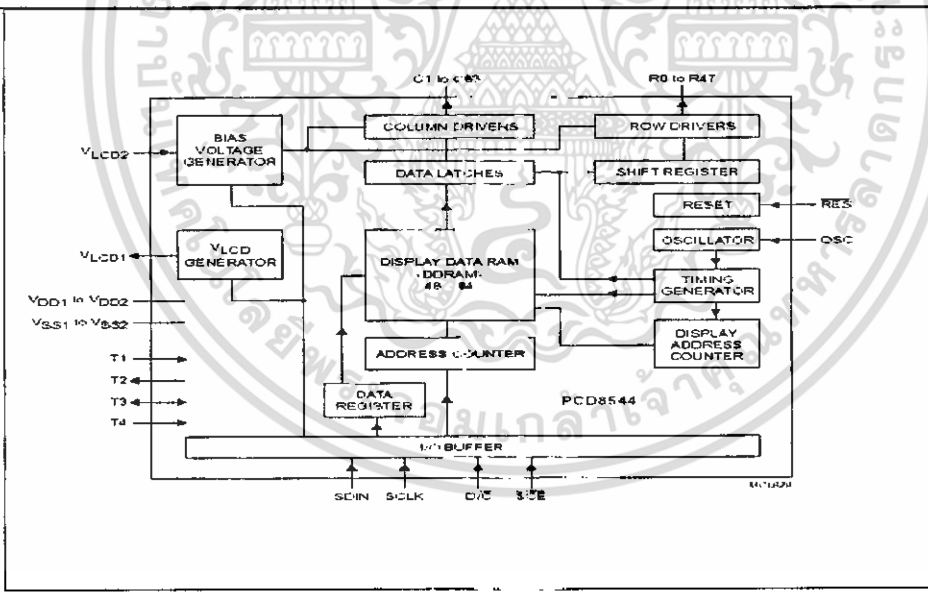
3.3.2 คุณสมบัติของหน้าจอ PCD 8544

* ใช้ชิปเดี่ยวเป็นตัว controller/driver
* มีขนาดความละเอียดของหน้าจอ 48 แถว 84 หลัก
* หน่วยความจำในการแสดงผล 48x84 บิต
* ใช้ชิปในการควบคุมการทำงาน
  - จ่ายแรงดันให้หน้าจอจากภายนอก
  - จ่ายแรงดันไบอัสให้หน้าจอแบบปานกลาง
  - ไม่จำเป็นต้องต่อสัญญาณคล็อกภายนอก เนื่องจากมีสัญญาณคล็อกบนชิพอยู่แล้ว
* รีเซตผ่านขา RES
* เชื่อมต่อแบบ Serial ส่งข้อมูลด้วยความเร็วสูงสุด 4.0 Mbits/s
* วงจรภาครับแบบ CMOS

* Mux rate : 48
* แรงดัน V_{DD} to V_{SS} ประมาณ 2.7 ถึง 3.3 โวลท์
* แรงดันไฟเลี้ยงหน้าจอ จาก V_{LCD} to V_{SS} 2 แบบ
  - 6.0 ถึง 8.5 โวลท์ กรณีที่ใช้แรงดันภายใน
  - 6.0 ถึง 9.0 โวลท์ กรณีที่ใช้แรงดันจากภายนอก
* กินไฟน้อย เหมาะกับการใช้งานที่แรงดันไฟต่ำ
* ค่าอุณภูมิชดเชยของ V_{LCD}
* ช่วงการทำงานในย่านอุณภูมิ -25 ถึง 70 °C

3.3.3 โครงสร้างของ PCD 8544

หน้าจอแสดงผลแบบกราฟฟิกที่ใช้ชิป PCD 8544 นั้นเป็นอุปกรณ์ที่ใช้กำลังงานต่ำและใช้ CMOS เป็นตัวควบคุมและเป็นตัว Driver โดยจะมีการแสดงผลออกหน้าจอที่มีขนาด 48 แถว 84 หลัก ในการใช้งานนั้นจะทำการส่งฟังก์ชั่นการใช้งานผ่านไปยังชิปซึ่งเป็นชิปเดี่ยว หลังจากนั้นชิปจะประ มวล ผลออกทางหน้าจอ LCD ส่วนการใช้งานเชื่อมต่อกับไมโครคอนโทรลเลอร์นั้น จะทำการเชื่อมต่อแบบ Serial Bus ดังรูปแสดงบล็อกการทำงานภายในของ LCD



Block diagram

รูปที่ 15 แสดงบล็อกภายในชิปของ PCD 8544

SYMBOL	DESCRIPTION
R0 to R47	สัญญาณเอาต์พุตทางด้าน แถว ของ LCD
C0 to C83	สัญญาณเอาต์พุตทางด้าน หลัก ของ LCD
V_{SS1}, V_{SS2}	Ground
V_{DD1}, V_{DD2}	ต่อไฟเลี้ยงของชิป
V_{LCD1}, V_{LCD2}	ต่อไฟเลี้ยงของ LCD
T1	Test 1 input (จะต่อขา V_{SS})
T2	Test 2 output (noconnect)
T3	Test 3 input/output (จะต่อขา V_{SS})
T4	Test 4 input (จะต่อขา V_{SS})
SDIN	Serial data in put (ขาสัญญาณข้อมูล อินพุต)
SCLK	Serial clock input (ขาสัญญาณนาฬิกา 0.0: to 4.0 Mbits/s)
D/C	Data/ command (ขาเลือกส่ง Data/ command)
SCE	Chip enable (เลือกให้ชิปทำงาน)
OSC	Oscillator
RES	External reset input
Dummy1,2,3,4	Not connected

ตารางที่ 11 การทำงานของขาต่างๆภายในชิป PCD 8544

3.3.4 ฟังก์ชั่นในการทำงาน

1. ตัวกำเนิดสัญญาณความถี่ (Oscillator) ไม่จำเป็นต้องต่อสัญญาณ Clock ภายนอก เนื่องจากมีสัญญาณ Clock บนชิพอยู่แล้วแต่ถ้าหากต้องการต่อClock จากภายนอกก็สามารถต่อกับขานี้ได้เช่นกัน

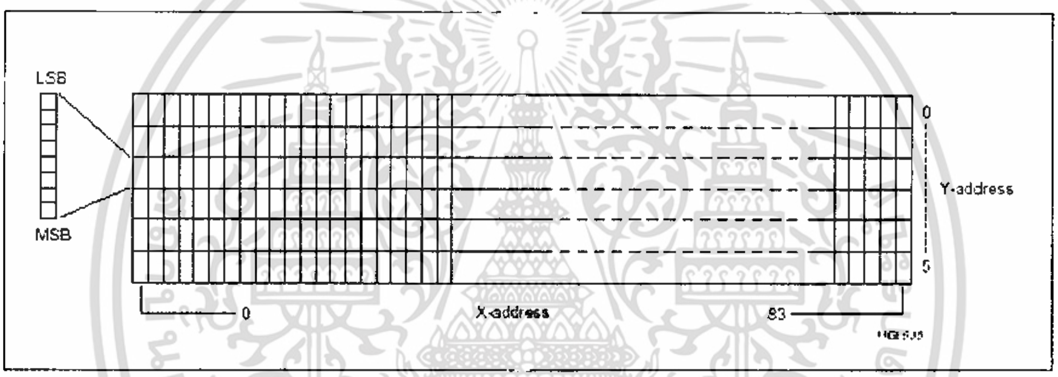
2. ตัวนับตำแหน่ง (Address Counter: AC) ตัวนับตำแหน่งเป็นตัวกำหนดค่าที่จะใช้เขียนข้อมูลจากหน่วยความจำข้อมูล โดยจะประกอบไปด้วยตำแหน่งในแกน X (X- address) X6 ถึง X0 และตำแหน่งในแกน Y (Y-address) Y2 ถึง Y0 ซึ่งสามารถแยกเซ็ตเป็นอิสระต่อกันได้ หลังจากทำการเขียนข้อมูลเสร็จแล้วตัวนับตำแหน่งจะเพิ่มค่าขึ้น 1 อัตโนมัติตามค่า V flag

3. หน่วยความจำข้อมูล (Display Data RAM: DDRAM) DDRAM เป็นแบบ static RAM ขนาด 48X84 bits ใช้สำหรับเก็บค่าข้อมูลที่จะทำการแสดง โดยแรมจะถูกแบ่งเป็น bank ขนาด 84 bit(6x8x84 bits) จำนวน 6 bankในขณะที่แรมรับข้อมูลเข้ามา ข้อมูลจะถูกส่งเข้ามาทาง serial interfaceโดยค่าตำ แหน่งในแกน X จะมีความสัมพันธ์โดยตรงกับตำแหน่งของ output column

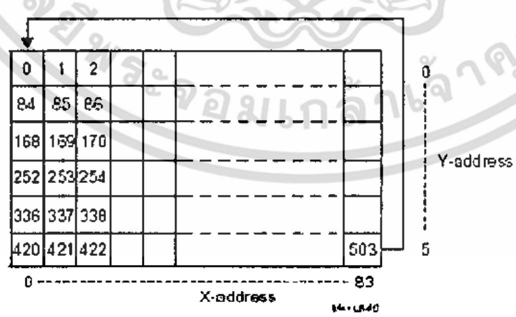
4. ตัวกำเนิดเวลา (Timing generator) ตัวกำเนิดเวลาเป็นตัวสร้างสัญญาณที่จำเป็นในการขับวงจรภายในชิพ โดยกระบวนการทำงานภายในชิพจะไม่ได้รับผลกระทบจากกระบวนการใดๆ ที่เกิดในบัสข้อมูลเลย

5. ตัวนับตำแหน่งแสดงผล (Display Address counter) ข้อมูลที่แสดงบนจอเกิดจากการเลื่อนข้อมูลในหน่วยความจำไปยังเอาท์พุทคอลัมน์แบบจุดบนหน้าจอ LCD

6. ตำแหน่งแสดงผล (Addressing) ข้อมูลจะถูกโหลดทีละไบต์เข้าไปในหน่วยความจำโดยจะใส่ทีละคอลัมน์จนเต็ม โดยคอลัมน์จะถูกระบุตำแหน่งโดย address pointer ซึ่งจะมีช่วงตำแหน่งแกน X อยู่ระหว่าง 0 ถึง 83 ส่วน Y อยู่ระหว่าง 0 ถึง 5 โดยการโหลดข้อมูลอาจจะเป็นไปโดย Y address จะเพิ่มขึ้นทุกๆไบต์ที่มีการโหลดเข้ามาเมื่อเต็มทั้งคอลัมน์แล้ว และก็จะเพิ่มค่าตำแหน่งแกน X หรืออาจจะทำการเพิ่มค่าในแกน X ก่อนเมื่อเต็มลัว ค่อยเพิ่มค่าในแกน Y ต่อไป จำนวนของแรมและการส่งข้อมูลออกหน้าจอเป็นดังรูป

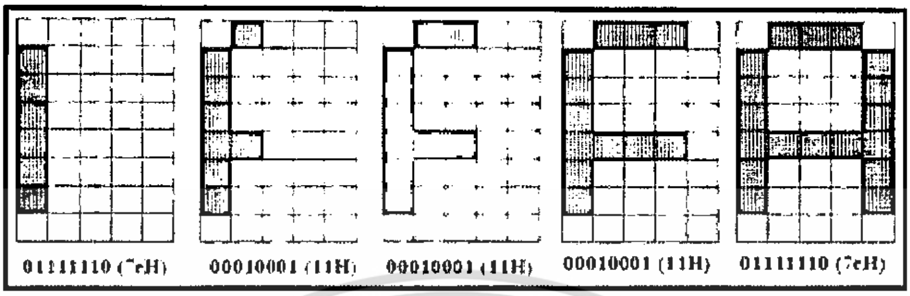


รูปที่ 16 แสดงรูปแบบของแรมและตำแหน่งของข้อมูล



รูปที่ 17 แสดงลำดับในการเขียนข้อมูลลงในหน่วยความจำแบบ horizontal addressing

7. การเก็บข้อมูลเพื่อใช้แสดงผลในการส่งข้อมูลไปแสดงเป็นอักขระบนจอเราจะทำการส่ง ไป 5 ไบต์โดยจะทำการส่งบิต most ไปก่อน เพื่อเก็บในตำแหน่งบิตล่างสุดในแต่ละไบต์ ตัวอย่างการส่งข้อมูลไปแสดงผลอักขระดังแสดงในรูป

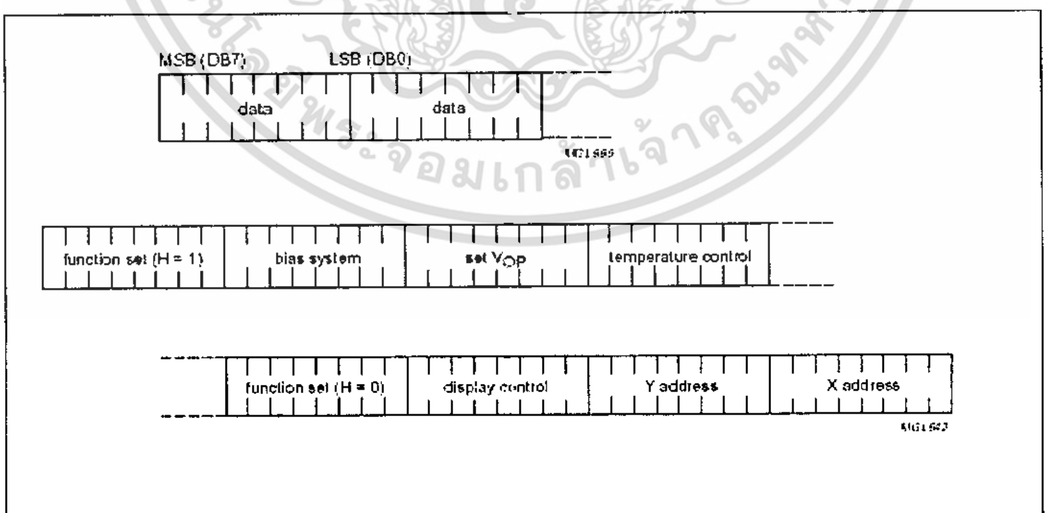


รูปที่ 18 แสดงลำดับขั้นการส่งข้อมูลไปแสดงผลอักขระ

3.3.5 หลักการทำงานของหน้าจอ LCD

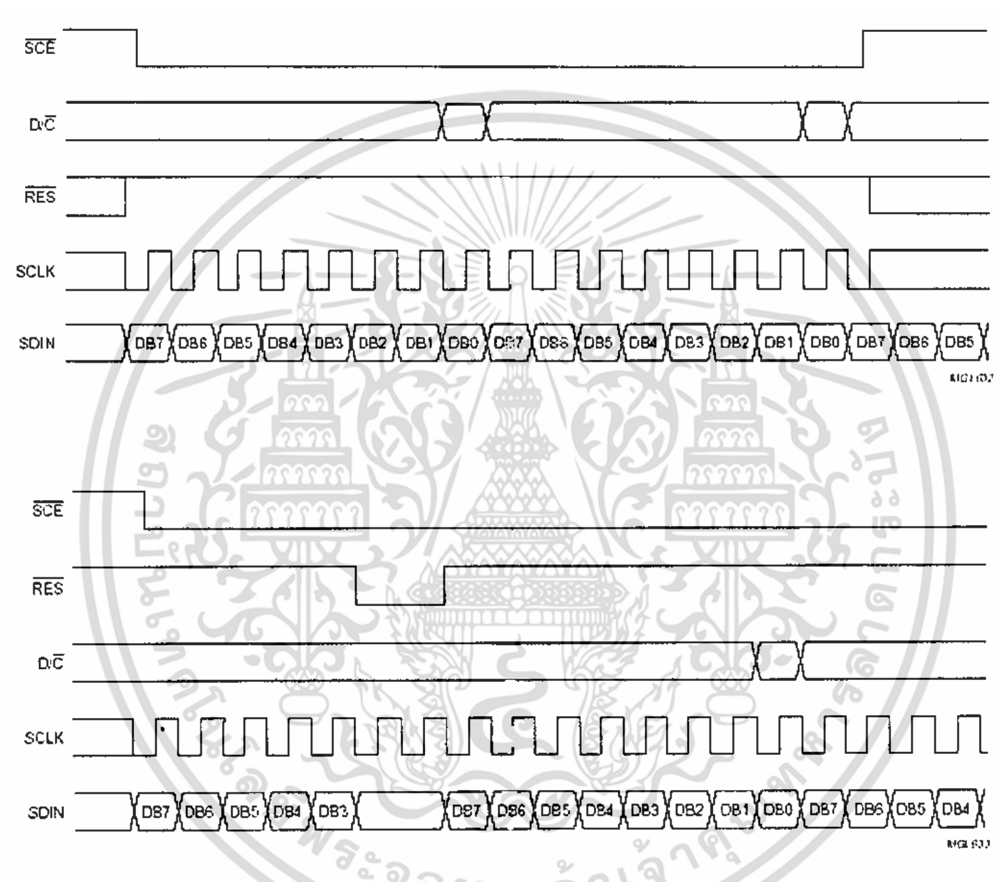
โครงสร้างรูปแบบการทำงานจะถูกแบ่งออกเป็น สองโหมด คือ ถ้ามีต้องการส่งค่า LOW มาจะถือว่าเป็นการทำงานในโหมด Command แต่ถ้ามีการส่ง HIGH มาก็จะเป็นการทำงานในโหมด ส่งข้อมูล DATA จาก RAM โดยจะมีการนับตำแหน่งเพิ่มขึ้นอย่างอัตโนมัติ

ในการส่งค่า $D/\overline{C}$ นั้นจะส่งระหว่างบิตสุดท้ายของข้อมูลในแต่ละไบต์ โดยในการส่งข้อมูลนั้น PCD8544 จะทำการส่งค่าด้าน MSB ก่อนแล้วค่อยส่งค่า ด้าน LSB และในการส่งแบบ Serial นั้นเริ่มต้นค่าของ -SCE จะเป็น HIGH สัญญาณนาฬิกาจะไม่มีผล และจะไม่มีการสูญเสียพลังงาน แต่เมื่อขา SCE ถูกเซตเป็น LOW จะเป็นการส่งค่าของข้อมูลได้ดังรูป



รูปที่ 19 แสดงหลักการส่งข้อมูลของ PCD 8544

เมื่อจะทำการส่งข้อมูลจะต้องทำการส่งค่า LOW ให้กับขา SCE และป้อนสัญญาณ SCLK และป้อนสัญญาณ ขา D/$\overline{C}$ ว่าจะส่งข้อมูลหรือส่งค่าคำสั่งในการทำงาน (0 = Command , 1 = DATA) หลังจากนั้นก็ทำการส่งข้อมูล โดยสามารถส่งข้อมูลผ่านขา SDIN ในการทำงานนั้นจะทำงานที่ขอบขาลง ยกเว้นการส่งข้อมูลที่จะทำงานที่ขอบขาขึ้นของ SCLK และเมื่อมีการรีเซตข้อมูลจะเป็นผลให้ข้อมูลในช่วงนั้นไม่ถูกส่งไป ดังรูป



รูปที่ 20 แสดงการส่งข้อมูลพร้อมสัญญาณ Clock

3.3.6 รูปแบบคำสั่ง COMMAND ในการทำงานของ LCD

INSTRUCTION	D/$\overline{C}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DESCRIPTION
		COMMAND BYTE								
(H=0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	No operation
Function set	0	0	0	1	0	0	PD	V	H	Power down control ; ; extended instruction set control(H)
Write data	1	D7	D6	D5	D4	D3	D2	D1	D0	Write data to display RAM
(H=0)										
Reserved	0	0	0	0	0	0	1	X	X	Do not use
Display control										Set display configuration
Reserved	0	0	0	0	1	X	X	X	X	Do not use
Set Y address of RAM										set Y address of RAM $0 \leq Y \leq 5$
Set X address of RAM	0	1	X6	X5	X4	X3	X2	X1	X0	Set X address part of RAM $0 \leq X \leq 83$
(H=1)										
Reserved	0	0	0	0	0	0	0	0	1	Do not use
	0	0	0	0	0	0	0	1	X	Do not use
Temperature Control	0	0	0	0	0	0	1	TC_1	TC_0	Set temparture coefficient(TC_X)
Reserved	0	0	0	0	0	1	X	X	X	Do not use
Bias system	0	0	0	0	1	0	BS_2	BS_1	BS_0	Set bias system (BS_X)
Reserved	0	0	1	X	X	X	X	X	X	Do not use
Set Vop	0	1	Vop6	Vop5	Vop4	Vop3	Vop2	Vop1	Vop0	Write Vop to register

ตารางที่ 12 แสดงรูปแบบคำสั่งต่างๆ ใน PCD 8544

BIT	0	1
PD	Chip is active	Chip is in Power-Down mode
V	Horizontal addressing	Vertical addressing
H	Use basic instruction set	Use extended instruction set
D and E 00 10 01 11	 Display blank Normal mode All display segment on Inverse video mode	
TC_1 and TC_1 00 01 10 11	 Vlcd tempareture coefficient0 Vlcd tempareture coefficient1 Vlcd tempareture coefficient2 Vlcd tempareture coefficient3	

ตารางที่ 13 แสดงความหมายและค่าของตัวแปร ในคำสั่ง COMMAND ของ PCD 8544

บทที่ 4

วงจรและการทดลอง

4.1 การทดลองและตรวจสอบการทำงานของเซนเซอร์

วัตถุประสงค์

1. เพื่อเป็นการตรวจสอบว่าเซนเซอร์ทำงานได้หรือไม่
2. เพื่อเป็นการตรวจสอบว่าเซนเซอร์มีความสามารถในการตรวจจับการเคลื่อนที่ได้หรือไม่
3. เพื่อตรวจสอบดูว่าเซอเซอร์สามารถผลิตสัญญาณพัลส์ ที่อถมีความถี่ตามที่ต้องการได้ โดยในโปรเจ็คนี้ต้องการสัญญาณพัลส์ที่มีความถี่ 1 กิโลเฮิร์ส
4. ทดสอบคุณสมบัติของตัวเซนเซอร์ว่ามีการเปลี่ยนแปลงค่าของ ดิวตี้ไซเคิลที่สัมพันธ์กับการเปลี่ยนแปลงความเร่งได้
5. หาค่าของดิวตี้ไซเคิลในสภาวะปกติ(ไม่มีความเร่งเกิดขึ้น) ในตำแหน่งการวางเซนเซอร์ในตำแหน่งต่างๆ

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

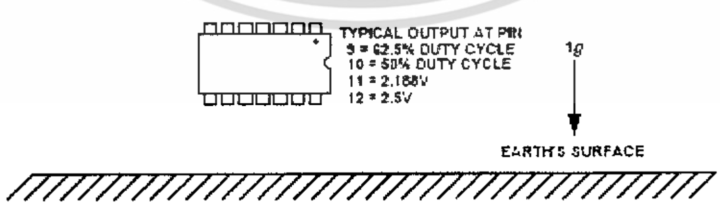
1. ออสซิลโลสโคป
2. แหล่งจ่ายไป 3.3 โวลต์
3. ตัวเซนเซอร์

ขั้นตอนการทดลอง

การทดลองหาค่าดิวตี้ไซเคิลปกติ(ขณะที่ยังไม่มีความเร่งมากระทำ) ที่ตอบสนองแรงโน้มถ่วงเมื่อมีการเปลี่ยนแปลงตำแหน่งของการวางเซนเซอร์ สามารถคำนวณหาค่าของดิวตี้ไซเคิล ได้จากสูตร

$$\text{Duty Cycle} = (T_1 / T_2) \times 100\ \%$$

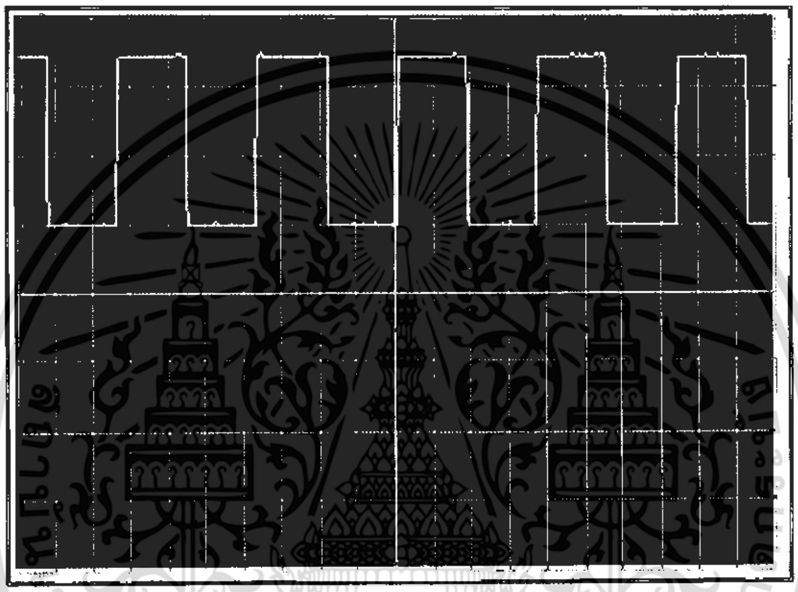
1. การจัดวางเซนเซอร์ให้อยู่ในตำแหน่ง ดังรูป และบันทึกค่าของดิวตี้ไซเคิล



รูปที่ 21 ใช้ทดลองวัดความเร่งเมื่อวางในตำแหน่งตะแคงขวาขนานกับพื้น

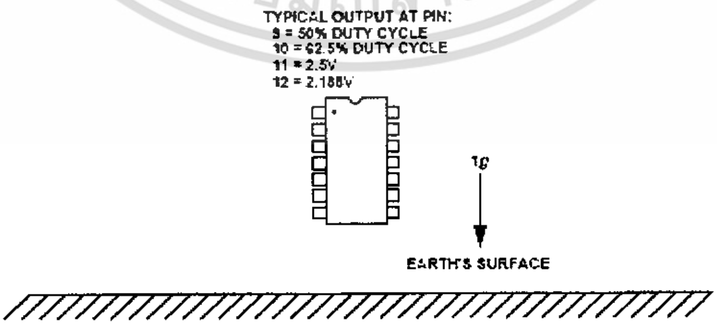
ดิวตี้ไซเคิล	ค่าที่ได้จากการทดลอง	ค่าทที่ได้จากการคำนวณ	หน่วย
X out	55.12	50	%
Y out	60	62	%

ตารางที่ 14 แสดงค่าดิวตี้ไซเคิลที่จากการวางในตำแหน่งตะแคงขวาขนานกับพื้น



รูปที่ 22 แสดงสัญญาณเอาท์พุทที่ได้จากการทดลองวางตะแคงขวาขนานกับพื้นดิน

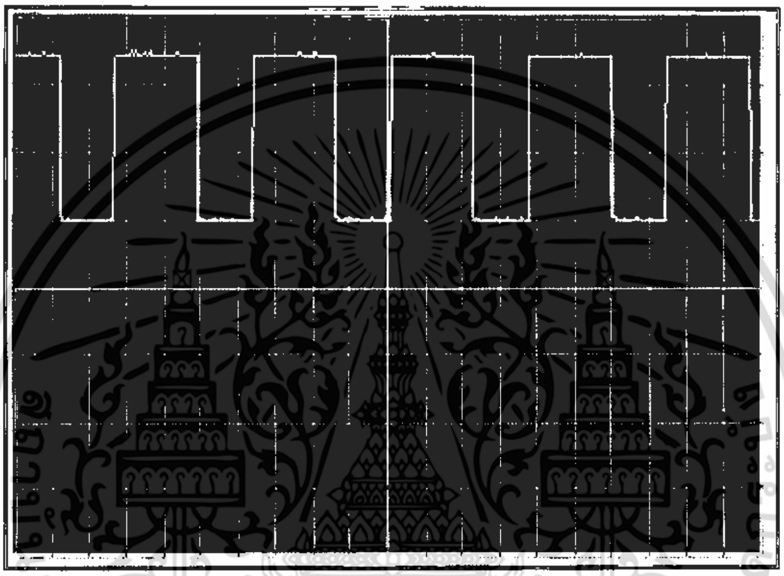
2. จัดวางเซนเซอร์ให้อยู่ในตำแหน่งดังรูป และบันทึกค่าของดิวตี้ไซเคิล



รูปที่ 23 ใช้ทดลองวัดความเร่งเมื่อวางในตำแหน่งตั้งฉากกับพื้น

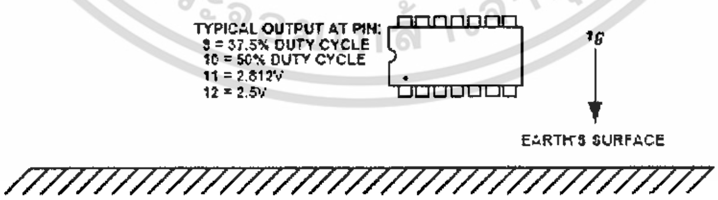
ดิวตี้ไซเคิล	ค่าที่ได้จากการทดลอง	ค่าที่ได้จากการคำนวณ	หน่วย
X out	61	62	%
Y out	50	50	%

ตารางที่ 15 แสดงค่าดิวตี้ไซเคิลที่จากวางในตำแหน่งตั้งฉากกับพื้นดิน



รูปที่ 24 แสดงสัญญาณเอาท์พุทที่ได้จากการวางในตำแหน่งตั้งฉากกับพื้นดิน

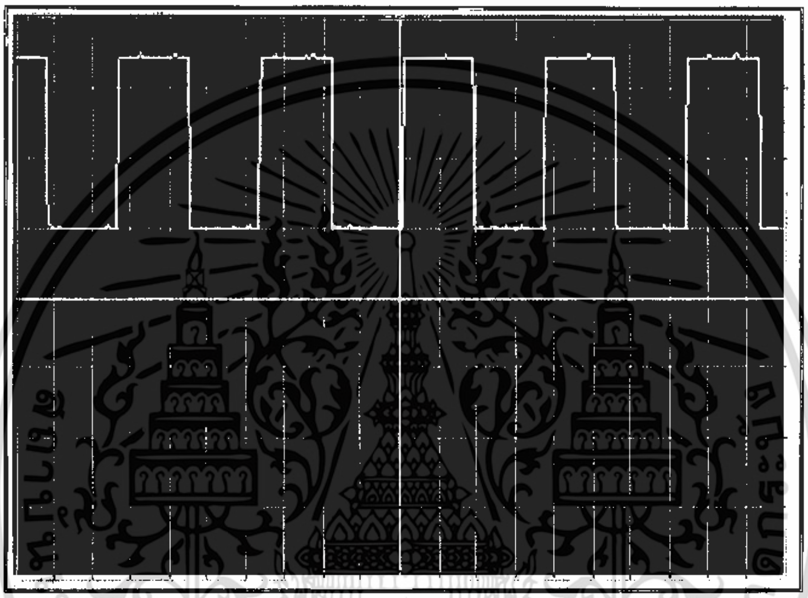
3. จัดวางเซนเซอร์ให้อยู่ในตำแหน่งดังรูป และบันทึกค่าของดิวตี้ไซเคิล



รูปที่ 25 ใช้ทดลองวัดความเร่งเมื่อวางในตำแหน่งตะแคงซ้ายตั้งขนานกับพื้น

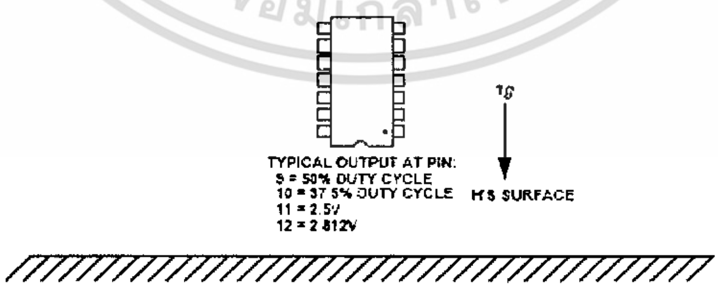
ดิวตี้ไซเคิล	ค่าที่ได้จากการทดลอง	ค่าทที่ได้จากการคำนวณ	หน่วย
X out	51.5	50	%
Y out	37.8	38	%

ตารางที่ 16 แสดงค่าดิวตี้ไซเคิลที่ได้จากการวางในตำแหน่งตะแคงซ้ายขนานกับพื้นดิน



รูปที่ 26 แสดงสัญญาณเอาท์พุทที่ได้ได้จากการวางในตำแหน่งตะแคงซ้ายขนานกับพื้นดิน

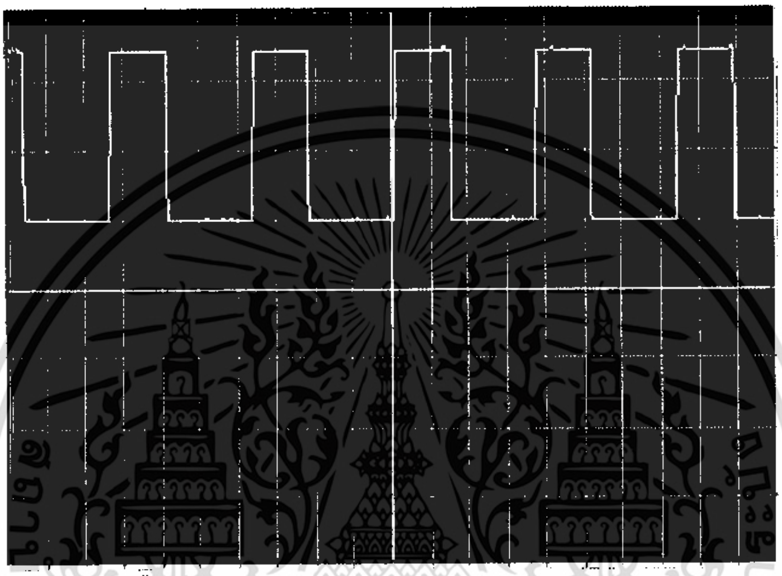
4. จัดวางเซนเซอร์ให้อยู่ในตำแหน่งดังรูป และบันทึกค่าของดิวตี้ไซเคิล



รูปที่ 27 ใช้ทดลองวัดความเร่งเมื่อวางในตำแหน่งหันหัวลงตั้งฉากกับพื้น

ดิวตี้ไซเคิล	ค่าที่ได้จากการทดลอง	ค่าทที่ได้จากการคำนวณ	หน่วย
X out	37.8	38	%
Y out	50	50	%

ตารางที่ 17 แสดงค่าดิวตี้ไซเคิลที่ได้จากการวางในตำแหน่งหันหัวลงตั้งฉากกับพื้น



รูปที่ 28 แสดงสัญญาณเอาท์พุททที่ได้จากการวางในตำแหน่งหันหัวลงตั้งฉากกับพื้น

5. ทำการทดลองโดยให้เซนเซอร์เคลื่อนที่ แล้วสังเกตค่าของดิวตี้ไซเคิล

พบว่าเมื่อตัวเซนเซอร์ได้มีการเคลื่อนที่ จะทำให้ค่าของดิวตี้ไซเคิลที่ถูกผลิตออกมาจากตัวเซนเซอร์มีการเปลี่ยนแปลง และค่าของดิวตี้ไซเคิลจะกลับมาอยู่ที่ตำแหน่งเดิมเมื่อมีการหยุดการเคลื่อนไหว หรือ ไม่มีการเปลี่ยนแปลงความเร่งนั่นเอง

รูปที่ 29 เซนเซอร์วัดความเร่งเมื่อประกอบลงบนปรินท์ลายวงจง

4.2 การทดลองในส่วนของหน้าจอกราฟฟิกแอลซีดี ของ Nokia 3310

วัตถุประสงค์

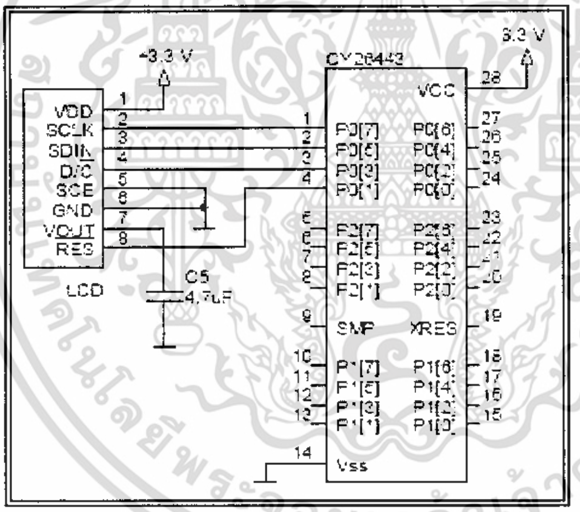
1. เพื่อเป็นการตรวจสอบดูว่าหน้าจอกราฟฟิกแอลซีดีสามารถตอบสนองการทำงานได้หรือไม่
2. เพื่อศึกษาการใช้งานหน้าจอกราฟฟิกแอลซีดี
3. เพื่อศึกษาการเขียนโปรแกรมโดยใช้ไมโครคอนโทรลเลอร์ PSoC มาควบคุมการแสดงผลหน้าจอกราฟฟิกแอลซีดี

เครื่องมือ และอุปกรณ์ที่ใช้ในการทดลอง

1. หน้าจอกราฟฟิกแอลซีดีของโทรศัพท์โนเกียร์3310
2. ไมโครคอนโทรลเลอร์ PSoC พร้อมชุดโปรแกรม
3. แหล่งจ่ายไป 3.3 โวลต์

ขั้นตอนการทดลอง

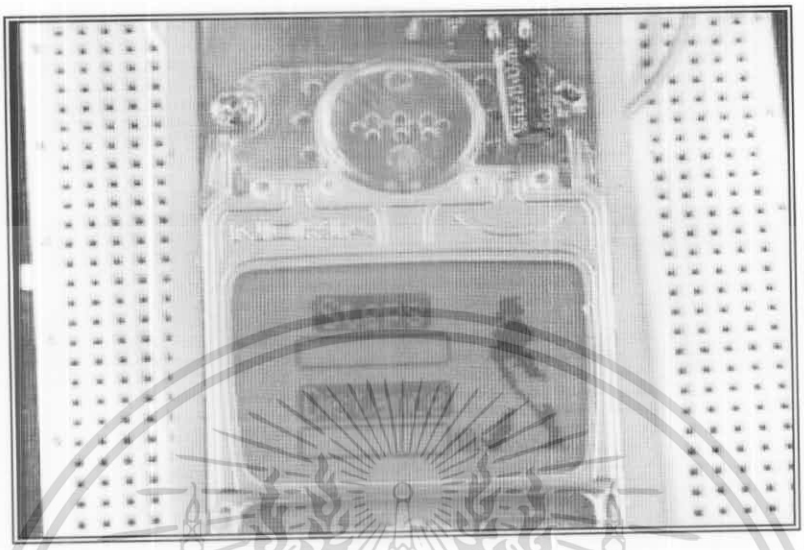
1. ต่อวงจรของจอกราฟฟิกแอลซีดี เข้ากับไมโครคอนโทรลเลอร์ PSoC ดังรูป



รูปที่ 30 วงจรการเชื่อมต่อหน้าจอกราฟฟิกเข้ากับ ไมโครคอนโทรลเลอร์PSoC

2. ทำการเขียนโปรแกรมเพื่อเขียนคำสั่งไปควบคุม โดยขั้นแรกเราต้องทำการกำหนดค่าเริ่มต้นต่างๆ ของหน้าจอ หรือที่เราเรียกว่า การ Set Initial function หลังจากนั้นก็จะทำการส่ง Command ไปก่อน แล้วตามด้วยการส่ง Data

3. สังเกตุที่หน้าจอกราฟฟิกแอลซีดี จะมีการตอบสนองเกิดขึ้นคือ เมื่อเราเขียนคำสั่งเพื่อที่ให้จอแสดงผล เป็นสกรีนแบ็กกราวนด์ ก็จะปรากฏแบ็กกราวนด์ขึ้นที่หน้าจอตามที่เราส่งไป



รูปที่ 31 การแสดงผลของกราฟฟิกแอลซีดีเมื่อเขียนคำสั่งควบคุม

4.3 การทดลองการทำงานของเครื่องนับก้าวเดิน

วัตถุประสงค์

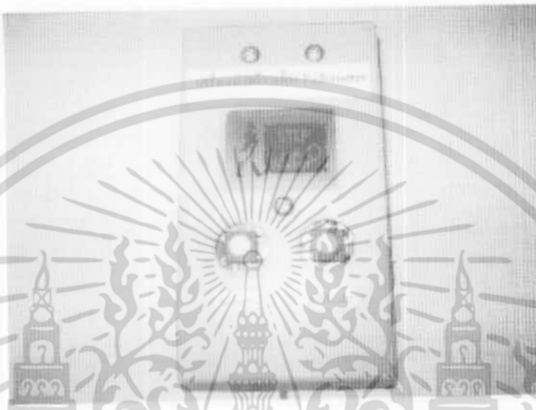
1. เพื่อเป็นการตรวจสอบการทำงานเครื่องนับก้าวเดินเมื่อใช้งานจริง

เครื่องมือ และอุปกรณ์ที่ใช้ในการทดลอง

1. เครื่องนับก้าวเดิน Pedometer

ขั้นตอนการทดลอง

1. ทำการเปิดเครื่องนับก้าวเดินเพื่อพร้อมสำหรับการทำงาน



รูปที่ แสดงเครื่องนับก้าวเดินขณะเปิดเครื่อง

2. ทำการตั้งค่าการใช้งานของเครื่องนับก้าวเดิน

รูปที่ แสดงการตั้งค่าการทำงานของเครื่องนับก้าวเดิน

3. ติดตั้งเครื่องนับก้าวเดินที่บริเวณเอว

รูปที่ แสดงการติดตั้งเครื่องนับก้าวเดิน

4. ทำการทดสอบโดยการเดินเป็นจำนวน 100 ก้าว เพื่อหาความผิดพลาดในการทำงานของเครื่องนับก้าวเดิน เทียบกับระยะจริงว่าเกิดความคลาดเคลื่อนไปเท่าใด

ครั้งที่	ค่าที่อ่านได้จากเครื่องนับก้าวเดิน	ค่าความผิดพลาด(%)
1	110	10
2	112	12
3	120	20
4	130	30
5	123	23
6	140	40
7	135	35
8	120	20
9	140	140
10	102	2

ตารางที่ 18 บันทึกค่าจากการทดลองของเครื่องนับก้าวเดิน กับการนับจริง 100 ก้าว

จากการทดลองจะได้ค่าเฉลี่ยจากการอ่านค่าของการนับก้าวเดินเป็น 123 ก้าว ค่าความผิดพลาดคิดเป็นเปอร์เซ็นได้ 23 %

บทที่ 5
สรุปและวิจารณ์

เซนเซอร์วัดความเร่ง

โดยปกติค่าของสัญญาณพัลส์ที่เกิดขึ้นจะมีอยู่ทั้งสองด้านคือทั้งทางด้านแกน X และแกน Y ซึ่งเป็นผลจากวงจรพัลส์วิดท์มอดูเลเตอร์ดิวตี้ไซเคิลภายในสภาวะเมื่อแรงโน้มถ่วงปกติหรือค่า g เป็นศูนย์ เอาท์พุทที่ได้จะถูกเซตให้มีค่าสัญญาณพัลส์ให้มีสัดส่วนระหว่าง T1/T2 เท่ากับ 50% แต่เมื่อใดที่มีแรงมากระทำให้เกิดการเปลี่ยนแปลง ค่า Duty Cycle ก็จะเปลี่ยนแปลงไปตามสเกลค่าเท่ากับ 12.5% ต่อ ค่า g หนึ่งค่า และจะได้ค่าความเร่งเป็นหน่วย g สามารถกำหนดคาบเวลาได้จากการเลือกใช้ค่าของตัวต้านทาน (Rset) ในการทดลองนั้นผู้จัดทำได้เลือกใช้ค่าของ Rset เป็น 125 k จะทำให้ได้ค่าความถี่สัญญาณประมาณ 1 kHz ได้ค่าคาบเวลาเป็น 1 ms

การวางตำแหน่งของตัวเซนเซอร์ยังมีผลทำให้ค่าของดิวตี้ไซเคิลของแต่ละแกนมีค่าแตกต่างกัน เมื่อมีความเร่งมากระทำจะทำให้ค่าของดิวตี้ไซเคิลที่ถูกผลิตออกมาจากตัวเซนเซอร์ในแต่ละแกนมีการเปลี่ยนแปลงที่แตกต่างกัน และค่าของดิวตี้ไซเคิลจะกลับมาอยู่ที่ตำแหน่งเดิมเมื่อมีการหยุดการเคลื่อนไหว หรือ ไม่มีการเปลี่ยนแปลงความเร่ง

การแสดงผลกราฟฟิก LCD NOKIA 3310

ข้อมูลจะถูกโหลดทีละไบต์เข้าไปในหน่วยความจำโดยจะใส่ทีละคอลัมน์จนเต็มโดยคอลัมน์จะถูกระบุตำแหน่งโดย address pointer ซึ่งจะมีช่วงตำแหน่งแกน X อยู่ระหว่าง 0 ถึง 83 ส่วน Y 0 ถึง 5 โดยการโหลดข้อมูลอาจจะเป็นไปโดย Y addressจะเพิ่มขึ้นทุกๆไบต์ที่มีการโหลดเข้ามาเมื่อเต็มทั้งคอลัมน์แล้วก็จะเพิ่มค่าตำแหน่งแกน X หรืออาจจะทำการเพิ่มค่าในแกน X ก่อน ค่อยเพิ่มค่าในแกน Yในการส่งข้อมูลไป

การแสดงเป็นอักขระบนจอทำได้โดยการส่งไป 5 ไบต์ โดยส่งบิต most ไปก่อน เพื่อเก็บในตำแหน่งบิตล่างสุดในแต่ละไบต์ ซึ่งในการติดต่อกับหน้าจอดังกล่าว จะใช้โครคอนโทรลเลอร์ PSoC ในการควบคุมการแสดงผล โดยมีการเรียกใช้โมดูล SPIM ซึ่งเป็นโมดูลการสื่อสารอนุกรมที่มีโปรโตคอลการสื่อสารเป็นแบบ SPI โดยจะเป็นการส่งแบบเข้าจังหวะ 8 บิตข้อมูลฟูลดูเพล็กซ์ (Full-duplex)โมดูล SPIM จะถูกเซตมีการส่งข้อมูลแบบส่งบิตนัยสำคัญต่ำสุดก่อน (LSB First)

การทดสอบการทำงานของเครื่องนับก้าวเดิน Pedometer

จากการทดสอบการใช้งานเครื่องนับก้าวเดิน Pedometer ในการใช้งานจริงนั้นได้ทำการเลือกใช้สวิตซ์ปรอทเป็นตัวเซนเซอร์การก้าวเดินจะเห็นได้ว่าจะมีค่าความผิดพลาดโดยเฉลี่ย 23 % จากการอ่านค่าของเครื่องนับก้าวเดินเทียบกับจำนวนก้าวจริง แสดงว่าการเลือกใช้สวิตซ์ปรอทนั้นมีความผิดพลาดสูงซึ่งอาจเกิดจาก ตำแหน่งที่ติดตั้ง ระดับความสั่นสะเทือนและการเคลื่อนที่ก้าวเดินเนื่องจากว่าสวิตซ์ปรอทนั้นมีความเป็น Stability ไม่ค่อยดีเลยทำให้ค่าที่อ่านได้มีความผิดพลาดค่อนข้างสูง

ปัญหาและแนวทางพัฒนาแก้ไข

จากการทดลองการใช้ไมโครคอนโทรลเลอร์ PSoC ในการรับอินพุตนั้นยังไม่สามารถทำการรับอินพุตจากเซนเซอร์ เมมส์ DXL 202 ได้ ซึ่งเกิดจากปัญหาในการใช้โมดูลของ PSoC ในการรับอินพุตที่เป็นดิวตี้ไซเคิลซึ่งยังไม่สามารถทำได้เหมือนกับไมโครคอนโทรลเลอร์แบบอื่นๆ (เนื่องจากเป็นไมโครคอนโทรลเลอร์ชนิดใหม่) ทำให้ฟังก์ชั่นบางฟังก์ชั่นยังไม่สามารถนำมาใช้งานได้อย่างถูกต้อง

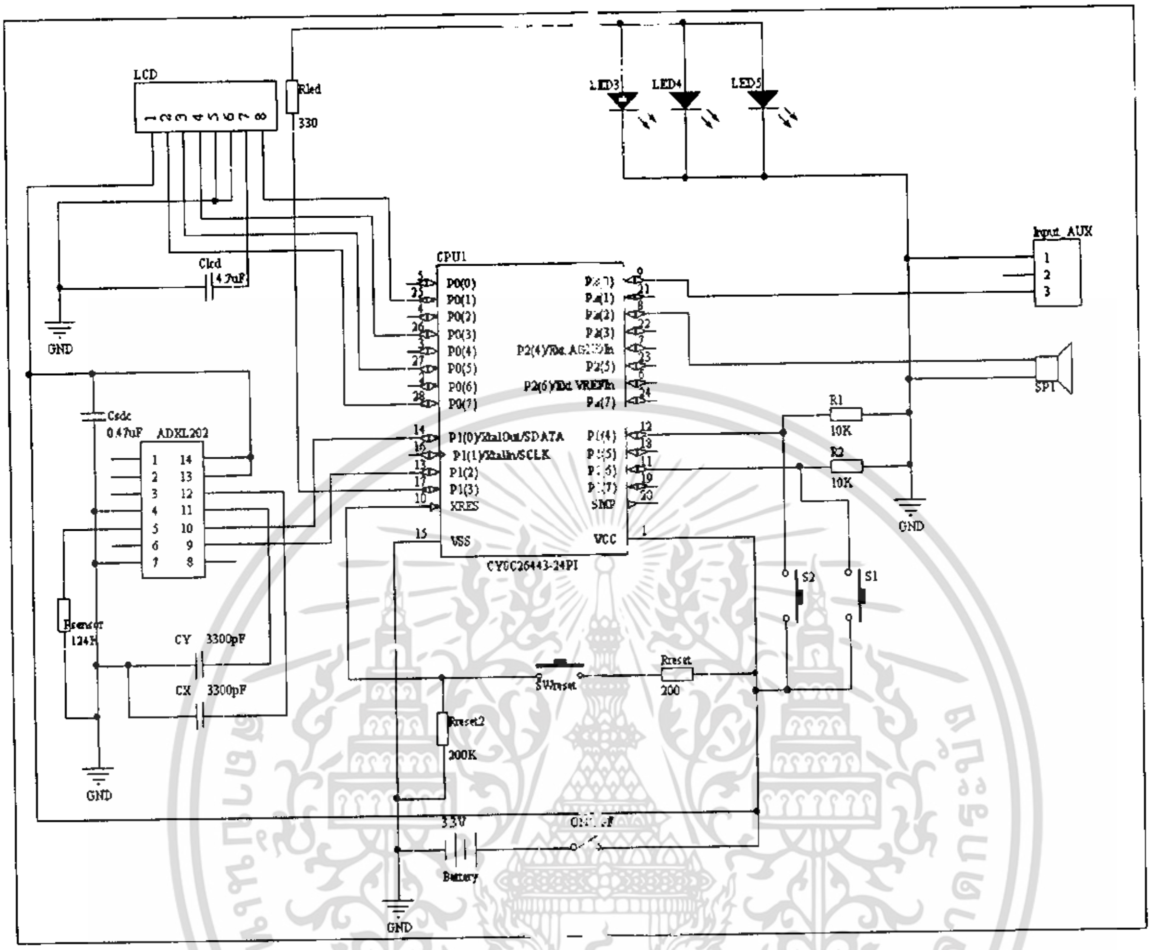
แนวทางแก้ไขอาจจะต้องมีการเปลี่ยนอุปกรณ์เซนเซอร์แบบใหม่ที่ให้เอาท์พุตแบบเป็นสวิตซ์ซึ่งจะใช้งานได้ง่ายกว่า หรือต้องทำการศึกษาการทำงานของโมดูลไมโครคอนโทรลเลอร์ PSoC ต่อไปให้ละเอียดลึกซึ้งมากยิ่งขึ้น เนื่องจากการใช้ไมโครคอนโทรลเลอร์ PSoC รับอินพุตแบบดิวตี้ไซเคิลนั้นยังไม่มีการทดลองหรือใช้งานมาก่อนจึงทำให้หาข้อมูลในการใช้งานทำได้ยาก

บรรณานุกรม

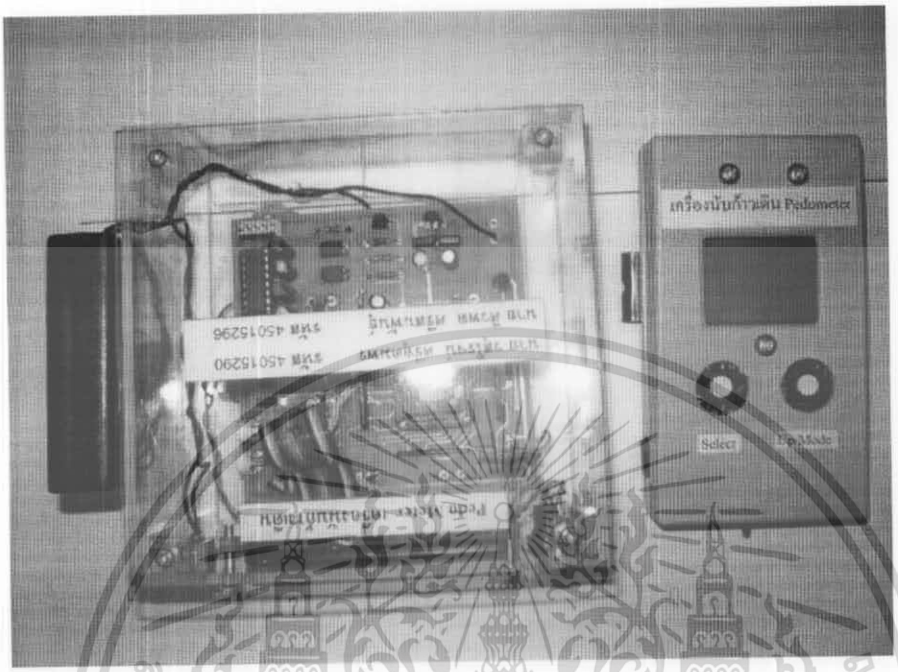
1. www.inventiondb.com , "ADXL 202 MEMS Accelerometer "
2. www.analog.com , "ADXL 202 MENS and Sensors "
3. www.shed.com/tutor/images/ "Basic Switches and the Powerflas "
4. www.en.wikipedia.org/wiki/Image:Mercuryswitch2.jpg " Mercury switch"
5. www.wisegeek.com/is-mercury-dangerous.htm "What is a Mercury Switch?"
6. www.cpu.com.tw/kh/sw/mer/mer.html "Mercury Switch "
7. www.semiconductors.philips.com/acrobat/datasheet/PCD8544_1.pdf , "PCD8544"
8. www.cypress.com/Aplication note/an2152.pdf ,"Aplication note Graphics lcd and PSoC interface"
9. www.100acre.org/elec/nokia_lcd/nokia_lcd.jal , "Nokia 3310 LCD Driver Library"
10. อกฤษฎ์ ตันทสุทธานนท์ ," การเขียนโปรแกรมไมโครคอนโทรลเลอร์ PSoC ด้วยภาษาซี " , MRT Design for Quality, 220 หน้า , 2548
11. วัชรินทร์ เคารพ , " เรียนรู้และเข้าใจ PSoC Microcontroller ด้วยภาษา Assembly และภาษา C" , บริษัท อีทีที จำกัด , 338 หน้า , 2548
12. NOkia 3310 LCD Driver Library

ภาคผนวก

วงจรเครื่องนับก้าวเดิน Pedometer



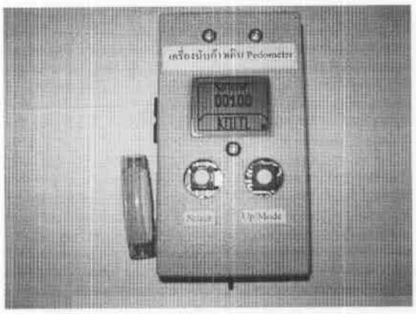
การใช้งานเครื่องนับก้าวเดิน

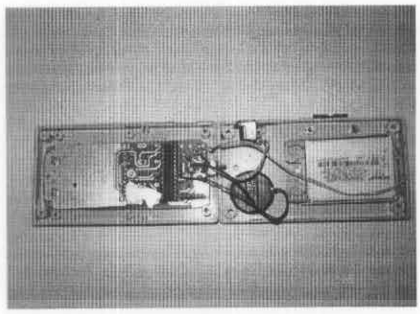


รูป A

รูป A แสดงการเปรียบเทียบเครื่องนับก้าวเดินรุ่นก่อนกับเครื่องนับก้าวเดินรุ่นใหม่ที่ใช้หน้าจอ LCD ของ Nokia 3310 ซึ่งจะเห็นได้ว่ามีขนาดที่เล็กลงกว่าเดิมเกือบ 3 เท่า และทำให้สามารถพกพาได้ง่ายกว่าดังรูป B

รูป B





รูป C

จากรูป C แสดงเครื่องนับก้าวเดินทั้งภายในและภายนอกจะเห็นได้ว่าเราสามารถเขียนโปรแกรมเข้าไปใหม่ได้เพียงแค่ถอด ไมโครคอนโทรลเลอร์ออกมา Set โปรแกรมลงไปใหม่แล้วนำไปใส่ไว้ที่ socket ดังเดิมเท่านั้นเอง

Source Code ของโปรแกรม

MainProgram

```
#include "m8c.h"
#include "PSoCAPI.h"
#include "string.h"
#include "math.h"
#include "delay.h"
#include "ioport.h"
#include "display_chen.h"
#include "form2.h" //Background 1
#include "form12.h" //Background 2
#include "formstep.h"//First Page
#include "page01.h"
#include "page02.h"
#include "page03.h"
#include "step2.h"
#include "step3.h"
#include "select1.h"
#include "select2.h"
//Include Number
#include "nn0.h"
#include "nn1.h"
#include "nn2.h"
#include "nn3.h"
#include "nn4.h"
#include "nn5.h"
#include "nn6.h"
#include "nn7.h"
#include "nn8.h"
#include "nn9.h"

void num_step(unsigned long);
void num_distance(unsigned long);
void num_calory(unsigned long);
void Beep(void);

const WORD PeriodValue = 1000; //Set Period of Counter

unsigned char select,select_1,select_2;
unsigned char mode,mode1,mode2,mode3,mode4;
static char page_setting = 1;
unsigned char v_sex,v_weight,v_height;

void main()
{
        unsigned int li[3] = { 2 , 12 , 8 };
        unsigned int cc[3] = { 1 , 1 , 1 };
        unsigned int temp,i,x;
        unsigned int count_step;
```

```
	unsigned long v_distance , v_energy;
	unsigned int c_stride;
	unsigned char voice;


	Counter16_1_WritePeriod(PeriodValue);
	Counter16_1_Start();

	ClrBit1_4;
	ClrBit1_6;

	mode2 = 1;
	select = 1;

//Motion Text is 'R' for Initial Display
	NokiaStart(form2);
	for(x=0;x<65;x++)
	{
		SetBit1_3;//LED Display
		Clrscr();
		GotoXY(x+2,3); NokiaString("R");
		//Beep();
		Delay1mS(50 );

	}
	Beep();
	Delay1mS(500);
	//ClrBit1_3;//Close LED
   while(1)
   {
	if(page_setting == 2)
	{
		//Display Step
		NokiaStart(formstep);
		count_step = PeriodValue - Counter16_1_wReadCounter();
		num_step(count_step); //Number of Steps

			//if(Bit0_0 == 0)
			//{
			//	Beep();
			//}
			SetBit1_3;
			if(count_step >= 5) ClrBit1_3;
			if(v_sex == 1)
			{
				c_stride = fround(v_height * 0.415);
				v_distance = count_step * c_stride;
				v_energy = (v_weight * v_distance)/100;
			}
			else if(v_sex == 2)
```

```
                    {
                            c_stride = fround(v_height * 0.413);
                            v_distance = count_step * c_stride;
                            v_energy = (v_weight * v_distance)/100;
                    }

            if(Bit1_4 == 1)
            {
                    Delay1mS(50);
                    if(Bit1_4 == 1)
                    {
//*******************Display Distance***************************
                            NokiaStart(page01);
                                    num_distance(v_distance);
                            Delay1mS(1500);
//*******************Display Energy*******************************

                            NokiaStart(page02);
                            num_calory(v_energy);
                            Delay1mS(1500);
                    }
            }
    }

    else if(page_setting == 1)
    {
            NokiaStart(page03);
            GotoXY(4,1); NokiaString("Sex");GotoXY(33,1);
//NokiaString("Female");
                    GotoXY(4,2); NokiaString("Weight");GotoXY(41,2);
//NokiaString("61-65");
                    GotoXY(4,3); NokiaString("Height");GotoXY(41,3);
//NokiaString("150-155");

                    if(Bit1_6 == 1 ) //push is Enter
                    {
                            Delay1mS(50);
                            if(Bit1_6 == 1)
                            {
                                    mode2 += 1;
                                    if(mode2 > 3)
                                    {
                                            page_setting = 2;
                                    }
                            }
                    }
                    else if(Bit1_4 == 1)//push is Select
                    {
                            Delay1mS(50);
                            if(Bit1_4 == 1)
```

```
                        {
                                cc[mode2-1] += 1;
                                if(cc[mode2-1] > li[mode2-1])
                                {
                                        cc[mode2-1] = 1;
                                }
                        }

                }
//*****************Select Sex*****************************

                select = cc[mode2-1];
                if(mode2 == 1)
                {
                        NokiaImage(1,1,3,1,select1);
                        NokiaImage(1,2,3,1,select2);
                        NokiaImage(1,3,3,1,select2);
                        GotoXY(4,1); NokiaString("Sex");GotoXY(33,1);
//NokiaString("Female");
                GotoXY(4,2); NokiaString("Weight");GotoXY(41,2);
//NokiaString("61-65");
                GotoXY(4,3); NokiaString("Height");GotoXY(41,3);
//NokiaString("150-155");
                        if(select == 1)
                        {
                                GotoXY(33,1); NokiaString("      ");
                                GotoXY(33,1); NokiaString("Male");
                                v_sex = 1;

                        }
                        if(select == 2)
                        {
                                GotoXY(33,1); NokiaString("      ");
                                GotoXY(33,1); NokiaString("Female");
                                v_sex = 2;
                        }
                }
//********************Select  Weight******************************

                else if(mode2 == 2)
                {
                        NokiaImage(1,1,3,1,select2);
                        NokiaImage(1,2,3,1,select1);
                        NokiaImage(1,3,3,1,select2);
                        GotoXY(4,1); NokiaString("Sex");GotoXY(33,1);
//NokiaString("Female");
                        GotoXY(4,2); NokiaString("Weight");GotoXY(41,2);
//NokiaString("61-65");
```

```
                        GotoXY(4,3); NokiaString("Height");GotoXY(41,3);
//NokiaString("150-155");

                        if(select == 1)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("40-45");
                                v_weight = 43;
                        }
                        if(select == 2)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("46-50");
                                v_weight = 48;
                        }
                        if(select == 3)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("51-55");
                                v_weight = 53;
                        }
                        if(select == 4)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("56-60");
                                v_weight = 58;
                        }
                        if(select == 5)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("61-65");
                                v_weight = 63;
                        }
                        if(select == 6)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("66-70");
                                v_weight = 68;
                        }
                        if(select == 7)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("71-75");
                                v_weight = 73;
                        }
                        if(select == 8)
                        {
                                GotoXY(41,2); NokiaString("     ");
                                GotoXY(41,2); NokiaString("76-80");
                                v_weight = 78;
```

```
                }
                if(select == 9)
                {
                        GotoXY(41,2); NokiaString("     ");
                        GotoXY(41,2); NokiaString("81-85");
                        v_weight = 83;
                }
                if(select == 10)
                {
                        GotoXY(41,2); NokiaString("     ");
                        GotoXY(41,2); NokiaString("86-90");
                        v_weight = 88;
                }
                if(select == 11)
                {
                        GotoXY(41,2); NokiaString("     ");
                        GotoXY(41,2); NokiaString("91-95");
                        v_weight = 93;
                }
                if(select == 12)
                {
                        GotoXY(41,2); NokiaString("     ");
                        GotoXY(41,2); NokiaString("96-100");
                        v_weight = 98;
                }
        }
//***************************Select   Height************************
        else if(mode2 == 3)
        {

                NokiaImage(1,1,3,1,select2);
                NokiaImage(1,2,3,1,select2);
                NokiaImage(1,3,3,1,select1);
                GotoXY(4,1); NokiaString("Sex");GotoXY(33,1);
//NokiaString("Female");
        GotoXY(4,2); NokiaString("Weight");GotoXY(41,2);
//NokiaString("61-65");
        GotoXY(4,3); NokiaString("Height");GotoXY(41,3);
//NokiaString("150-155");

                if(select == 1)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("150-155");
                        v_height = 153;
                }
                if(select == 2)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("156-160");
```

```
                        v_height = 158;
                }
                if(select == 3)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("161-165");
                        v_height = 163;
                }
                if(select == 4)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("166-170");
                        v_height = 168;
                }
                if(select == 5)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("171-175");
                        v_height = 173;
                }
                if(select == 6)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("176-180");
                        v_height = 178;
                }
                if(select == 7)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("181-185");
                        v_height = 183;
                }
                if(select == 8)
                {
                        GotoXY(41,3); NokiaString("      ");
                        GotoXY(41,3); NokiaString("186-190");
                        v_height = 188;
                }
        }

//**********************Display  Sex****************************
        for(i = 0;i < 3;i++)
        {
                temp = cc[i];

                if(i == 0)
                {
                        if(temp == 1)
                        {
```

```
                        GotoXY(33,1); NokiaString("Male   ");
                        v_sex = 1;
                }
                else if(temp == 2)
                {
                        GotoXY(33,1); NokiaString("Female ");
                        v_sex = 2;
                }
        }
//***********************Display  Weight************************
        else if(i == 1)
        {
                if(temp == 1)
                {
                        GotoXY(41,2); NokiaString("40-45");
                        v_weight = 43;
                }
                else if(temp == 2)
                {
                        GotoXY(41,2); NokiaString("46-50");
                        v_weight = 48;
                }
                else if(temp == 3)
                {
                        GotoXY(41,2); NokiaString("51-55");
                        v_weight = 53;
                }
                        else if(temp == 4)
                {
                        GotoXY(41,2); NokiaString("56-60");
                        v_weight = 58;
                }
                else if(temp == 5)
                {
                        GotoXY(41,2); NokiaString("61-65");
                        v_weight = 63;
                }
                        else if(temp == 6)
                {
                        GotoXY(41,2); NokiaString("66-70");
                        v_weight = 68;
                }
                else if(temp == 7)
                {
                        GotoXY(41,2); NokiaString("71-75");
                        v_weight = 73;
                }
                        else if(temp == 8)
                {
                        GotoXY(41,2); NokiaString("76-80");
```

```
                v_weight = 78;
        }
        else if(temp == 9)
        {
                GotoXY(41,2); NokiaString("81-85");
                v_weight = 83;
        }
                else if(temp == 10)
        {
                GotoXY(41,2); NokiaString("86-90");
                v_weight = 93;
        }
        else if(temp == 11)
        {
                GotoXY(41,2); NokiaString("91-95");
                v_weight = 93;
        }
        else if(temp == 12)
        {
                GotoXY(41,2); NokiaString("96-100");
                v_weight = 98;
        }
    }
//*************************Display  height***************************
    else if(i == 2)
    {
        if(temp == 1)
        {
                GotoXY(41,3); NokiaString("150-155");
                v_height = 153;
        }
        else if(temp == 2)
        {
                GotoXY(41,3); NokiaString("156-160");
                v_height = 158;
        }
        else if(temp == 3)
        {
                GotoXY(41,3); NokiaString("161-165");
                v_height = 163;
        }
        else if(temp == 4)
        {
                GotoXY(41,3); NokiaString("166-170");
                v_height = 168;
        }
        else if(temp == 5)
        {
                GotoXY(41,3); NokiaString("171-175");
                v_height = 173;
```

```
                }
                else if(temp == 6)
                {
                        GotoXY(41,3); NokiaString("176-180");
                        v_height = 178;
                }
                else if(temp == 7)
                {
                        GotoXY(41,3); NokiaString("181-185");
                        v_height = 183;
                }
                else if(temp == 8)
                {
                        GotoXY(41,3); NokiaString("186-190");
                        v_height = 188;
                }
            }

          }
        //page_setting == 2
        }

      //NokiaStart(page4);
    }
}
void Beep(void)
{
        PWM16_1_WritePeriod(511);
        PWM16_1_WritePulseWidth(255);
        PWM16_1_DisableInt();
        PWM16_1_Start();
        Delay1mS(100);
        PWM16_1_Stop();
        Delay1mS(100);
        PWM16_1_Start();
        Delay1mS(100);
        PWM16_1_Stop();
        Delay1mS(100);
}
```

Display

```
#include "m8c.h"
#include "spim_1.h"
#include "display_chen.h"

int Background_cursor;

//Send Data
void SendData ( char data )
{
```

```
  PRT0DR |= DC_PIN;
  SPIM_1_SendTxData(data);

  while ( ! (bSPIM_1_ReadStatus() & SPIM_1_SPIM_TX_BUFFER_EMPTY) );
 }

 //Send Command
 void SendCommand ( char command )
 {
  PRT0DR &= ~DC_PIN;
  SPIM_1_SendTxData(command);
  while ( ! (bSPIM_1_ReadStatus() & SPIM_1_SPIM_TX_BUFFER_EMPTY) );
 }

 static void wait ( void )
 {
  int time;
  for (time= -32000; time< 32000; time++ );
 }



 #ifdef CHEN
 void NokiaStart (const char * background)
 #else
 void NokiaStart ( void )
 #endif
 {
  int i;
  wait();
  PRT0DR = RST_PIN;
  SPIM_1_Start(SPIM_1_SPIM_MODE_0 | SPIM_1_SPIM_MSB_FIRST);
  SendCommand(0x21); // LCD Extended Commands.
  SendCommand(0x88); // Set LCD Vop (Contrast).
  SendCommand(0x06); // Set Temp coefficent.
  SendCommand(0x13); // LCD bias mode 1:48.
  SendCommand(0x20); // LCD Standard Commands, Horizontal addressing mode.
  SendCommand(0x0C); // LCD in normal mode.
  SendCommand(0x40); //clear display
  SendCommand(0x80); //clear display
 #ifdef CHEN
  Background = background;
  for(i=0; i<504; i++)
  {
        SendData(Background[i]);
  }
  Background_cursor = 0;
 #else
  for(i=0; i<504; i++)
  {
```

```
        SendData(0x00);
  }
 #endif
 }


// SetBackground

#ifdef CHEN
void SetBackground ( const char *dataPtr )
{
 Background = dataPtr;
}
#endif


//Contrast

void Contrast ( char contrast )
{
 SendCommand(0x21);
 SendCommand(0x80 | contrast);
 SendCommand(0x20);
}


//Clear Screen

void Clrscr( void )
{
 int i;
#ifdef CHEN
 SendCommand(0x40);
 SendCommand(0x80);
 for(i=0; i<504; i++)
 {
       SendData(Background[i]);
 }
#else
 for(i=0; i<504; i++)
 {
       SendData(0x00);
 }
#endif
 SendCommand(0x40);
 SendCommand(0x80);
}

//Goto XY
```

```
void GotoXY ( char x, char y )
{
// SendCommand(0x40 | y);
// SendCommand(0x80 | x);
 SendCommand(0x40|(y&0x07));
 SendCommand(0x80|(x&0x7f));
#ifdef CHEN
 Background_cursor = y*84+x;
#endif
}

//String

#ifdef CHEN
void String ( char *dataPtr )
{
 int i;
 while ( *dataPtr )
 {
  char ch = *dataPtr++;

   for(i=0; i<5; i++, Background_cursor++)
   {
       SendData(Font[ch-32][i] ^ Background[Background_cursor]);
   }
  SendData(Background[Background_cursor++]);
 }
}
#else
void String ( char *dataPtr )
{
 int i;
 while ( *dataPtr )
 {
  char ch = *dataPtr++;
  for(i=0; i<5; i++)
  {
       SendData(Font[ch-32][i]);
  }
  SendData(0x00);
 }
}
#endif

//CString

#ifdef CHEN
void NokiaString ( const char *dataPtr)
{
 int i;
```

```
  while ( *dataPtr )
  {
   char ch = *dataPtr++;

     for(i=0; i<5; i++, Background_cursor++)
     {
          SendData(Font[ch-32][i] ^ Background[Background_cursor]);
     }
   SendData(Background[Background_cursor++]);
  }
}
#else
void NokiaString ( const char *dataPtr )
{
 int i;
 while ( *dataPtr )
 {
  char ch = *dataPtr++;
  for(i=0; i<5; i++)
  {
        SendData(Font[ch-32][i]);
  }
  SendData(0x00);
 }
}
#endif
/*Character*/
#ifdef CHEN
void NokiaChar ( char ch )
{
 int i;

   for(i=0; i<5; i++, Background_cursor++) SendData(Font[ch-32][i] |
Background[Background_cursor]);
  SendData(Background[Background_cursor]); // empty space after character
}
#else //DRAW_OVER_BACKGROUND
void NokiaChar ( char ch )
{
 int i;
 for(i=0; i<5; i++) SendData(Font[ch-32][i]);
 // empty space after character
 SendData(0x00);
}
#endif //DRAW_OVER_BACKGROUND


void NokiaImage (char x, char y, char xsize, char ysize, const char * dataPtr)
{
 int i, j;
```

```
  for (j=0; j<ysize; j++)
  {
   // set start address
   SendCommand(0x40 | y+j);
   SendCommand(0x80 | x);
   for (i=0; i<xsize; i++) SendData(dataPtr[j*xsize+i]);
  }
#ifdef CHEN
 Background_cursor = (y+ysize)*84+x+xsize;
#else //DRAW_OVER_BACKGROUND
#endif //DRAW_OVER_BACKGROUND
}
```

Number

```
#include "m8c.h"
#include "string.h"
#include "display_chen.h"
#include "nn0.h"
#include "nn1.h"
#include "nn2.h"
#include "nn3.h"
#include "nn4.h"
#include "nn5.h"
#include "nn6.h"
#include "nn7.h"
#include "nn8.h"
#include "nn9.h"



/****************Step*********************/

void num_step(unsigned long ns)
{
	//static const char step_digit1[10] = {'0','1','2','3','4','5','6','7','8','9'};
	//static const char step_digit2[10] = {'0','1','2','3','4','5','6','7','8','9'};
	//static const char step_digit3[10] = {'0','1','2','3','4','5','6','7','8','9'};
	//static const char step_digit4[10] = {'0','1','2','3','4','5','6','7','8','9'};
	//static const char step_digit5[10] = {'0','1','2','3','4','5','6','7','8','9'};
	unsigned long sn,sd1,sd2,sd3,sd4,sd5;
	unsigned char l = 20;
	unsigned char k = 1;
	/*Step Number*/
			sn = ns;
		sd1 = sn % 10;
		sd2 = (sn / 10) % 10 ;
```

```
sd3 = (sn / 100) % 10 ;
sd4 = (sn / 1000) % 10 ;
sd5 = sn / 10000;

/*Step_Digit1*/
//GotoXY(l+24,1);     NokiaChar(step_digit1[sd1]);
/*Step_Digit2*/
//GotoXY(l+18,1);     NokiaChar(step_digit2[sd2]);
/*Step_digit3*/
//GotoXY(l+12,1);     NokiaChar(step_digit3[sd3]);
/*Step_Digit4*/
//GotoXY(l+6,1);      NokiaChar(step_digit4[sd4]);
/*Step_Digit5*/
//GotoXY(l,1);NokiaChar(step_digit5[sd5]);

switch(sd5)
{
        case 0 :NokiaImage(l,k,8,2,nn0);
                                break;
        case 1 :        NokiaImage(l,k,8,2,nn1);
                                break;
        case 2 :NokiaImage(l,k,8,2,nn2);
                                break;
        case 3 :        NokiaImage(l,k,8,2,nn3);
                                break;
        case 4 :NokiaImage(l,k,8,2,nn4);
                                break;
        case 5 :NokiaImage(l,k,8,2,nn5);
                                break;
        case 6 :NokiaImage(l,k,8,2,nn6);
                                break;
        case 7 :        NokiaImage(l,k,8,2,nn7);
                                break;
        case 8 :NokiaImage(l,k,8,2,nn8);
                                break;
        case 9 :        NokiaImage(l,k,8,2,nn9);
                                break;
}
switch(sd4)
{
        case 0 :NokiaImage(l+9,k,8,2,nn0);
                                break;
        case 1 :        NokiaImage(l+9,k,8,2,nn1);
                                break;
        case 2 :NokiaImage(l+9,k,8,2,nn2);
                                break;
        case 3 :        NokiaImage(l+9,k,8,2,nn3);
                                break;
        case 4 :NokiaImage(l+9,k,8,2,nn4);
                                break;
```

```
        case 5 :NokiaImage(l+9,k,8,2,nn5);
                        break;
        case 6 :NokiaImage(l+9,k,8,2,nn6);
                        break;
        case 7 :        NokiaImage(l+9,k,8,2,nn7);
                        break;
        case 8 :NokiaImage(l+9,k,8,2,nn8);
                        break;
        case 9 :        NokiaImage(l+9,k,8,2,nn9);
                        break;
}
switch(sd3)
{
        case 0 :NokiaImage(l+18,k,8,2,nn0);
                        break;
        case 1 :        NokiaImage(l+18,k,8,2,nn1);
                        break;
        case 2 :NokiaImage(l+18,k,8,2,nn2);
                        break;
        case 3 :        NokiaImage(l+18,k,8,2,nn3);
                        break;
        case 4 :NokiaImage(l+18,k,8,2,nn4);
                        break;
        case 5 :NokiaImage(l+18,k,8,2,nn5);
                        break;
        case 6 :NokiaImage(l+18,k,8,2,nn6);
                        break;
        case 7 :        NokiaImage(l+18,k,8,2,nn7);
                        break;
        case 8 :NokiaImage(l+18,k,8,2,nn8);
                        break;
        case 9 :        NokiaImage(l+18,k,8,2,nn9);
                        break;
}
switch(sd2)
{
        case 0 :NokiaImage(l+27,k,8,2,nn0);
                        break;
        case 1 :        NokiaImage(l+27,k,8,2,nn1);
                        break;
        case 2 :NokiaImage(l+27,k,8,2,nn2);
                        break;
        case 3 :        NokiaImage(l+27,k,8,2,nn3);
                        break;
        case 4 :NokiaImage(l+27,k,8,2,nn4);
                        break;
        case 5 :NokiaImage(l+27,k,8,2,nn5);
                        break;
        case 6 :NokiaImage(l+27,k,8,2,nn6);
                        break;
```

```
                case 7 :        NokiaImage(l+27,k,8,2,nn7);
                                break;
                case 8 :NokiaImage(l+27,k,8,2,nn8);
                                break;
                case 9 :        NokiaImage(l+27,k,8,2,nn9);
                                break;
        }
        switch(sd1)
        {
                case 0 :NokiaImage(l+36,k,8,2,nn0);
                                break;
                case 1 :        NokiaImage(l+36,k,8,2,nn1);
                                break;
                case 2 :NokiaImage(l+36,k,8,2,nn2);
                                break;
                case 3 :        NokiaImage(l+36,k,8,2,nn3);
                                break;
                case 4 :NokiaImage(l+36,k,8,2,nn4);
                                break;
                case 5 :NokiaImage(l+36,k,8,2,nn5);
                                break;
                case 6 :NokiaImage(l+36,k,8,2,nn6);
                                break;
                case 7 :        NokiaImage(l+36,k,8,2,nn7);
                                break;
                case 8 :NokiaImage(l+36,k,8,2,nn8);
                                break;
                case 9 :        NokiaImage(l+36,k,8,2,nn9);
                                break;
        }
}

/****************Distance*****************/

void num_distance(unsigned long nd)
{
        /*Distance Number*/
        static const char dist_digit1[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit2[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit3[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit4[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit5[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit6[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char dist_digit7[10] = {'0','1','2','3','4','5','6','7','8','9'};
        unsigned long dn,dd1,dd2,dd3,dd4,dd5,dd6,dd7;
        unsigned char m = 18;
                dn = nd;
                dd1 = dn % 10 ;
                dd2 = (dn / 10) % 10 ;
                   dd3 = (dn / 100) % 10 ;
```

```
                dd4 = (dn / 1000) % 10 ;
                dd5 = (dn / 10000) % 10 ;
                dd6 = (dn / 100000) % 10 ;
                dd7 = dn / 1000000;
                /*Distance_Digit1*/
                GotoXY(m+42,3);     NokiaChar(dist_digit1[dd1]);
                /*Distance_Digit2*/
                GotoXY(m+36,3);     NokiaChar(dist_digit2[dd2]);
                /*Distance_Digit3*/
                GotoXY(m+30,3);     NokiaChar('.');
                /*Distance_Digit4*/
                GotoXY(m+24,3);     NokiaChar(dist_digit4[dd3]);
                /*Distance_Digit5*/
                GotoXY(m+18,3);     NokiaChar(dist_digit5[dd4]);
                /*Distance_Digit6*/
                        GotoXY(m+12,3);     NokiaChar(dist_digit3[dd5]);
                        /*Distance_Digit7*/
                        GotoXY(m+6,3);      NokiaChar(dist_digit3[dd6]);
                        /*Distance_Digit8*/
                        GotoXY(m,3);NokiaChar(dist_digit3[dd7]);

}
/***************Calory*****************/

void num_calory(unsigned long nc)
{
        /*Calory Number*/
        static const char cal_digit1[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char cal_digit2[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char cal_digit3[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char cal_digit4[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char cal_digit5[10] = {'0','1','2','3','4','5','6','7','8','9'};
        static const char cal_digit6[10] = {'0','1','2','3','4','5','6','7','8','9'};
        unsigned long cn,cd1,cd2,cd3,cd4,cd5,cd6;
        unsigned char n = 14;
                cn = nc;
                cd1 = cn % 10;
                cd2 = (cn / 10) % 10 ;
                cd3 = (cn / 100) % 10 ;
                cd4 = (cn / 1000) % 10 ;
                cd5 = (cn / 10000) % 10 ;
                cd6 = (cn / 100000);
                /*Calory_Digit1*/
                GotoXY(n+36,3);     NokiaChar(cal_digit1[cd1]);
                /*Calory_Digit2*/
                GotoXY(n+30,3);     NokiaChar('.');
                /*Calory_Digit3*/
                GotoXY(n+24,3);     NokiaChar(cal_digit2[cd2]);
                /*Calory_Digit4*/
                GotoXY(n+18,3);     NokiaChar(cal_digit3[cd3]);
```

```
        /*Calory_Digit5*/
        GotoXY(n+12,3);      NokiaChar(cal_digit4[cd4]);
        /*Calory_Digit6*/
        GotoXY(n+6,3);       NokiaChar(cal_digit5[cd5]);
        /*Calory_Digit7*/
        GotoXY(n,3); NokiaChar(cal_digit6[cd6]);
}
/***************Heart Rate*****************/

void num_HeartRate(unsigned int nh)
{
    /*Calory Number*/
    static const char hr_digit1[10] = {'0','1','2','3','4','5','6','7','8','9'};
    static const char hr_digit2[10] = {'0','1','2','3','4','5','6','7','8','9'};
    static const char hr_digit3[10] = {'0','1','2','3','4','5','6','7','8','9'};
    unsigned int hn,hd1,hd2,hd3;
    unsigned char n = 22;
        hn = nh;
        hd1 = hn % 10;
        hd2 = (hn / 10) % 10 ;
        hd3 = (hn / 100);
        /*Calory_Digit1*/
        GotoXY(n+12,4);      NokiaChar(hr_digit1[hd1]);
        /*Calory_Digit2*/
        GotoXY(n+6,4);       NokiaChar(hr_digit2[hd2]);
        /*Calory_Digit3*/
        GotoXY(n,4); NokiaChar(hr_digit3[hd3]);
}
```

ANALOG DEVICES

Low Cost ±2 *g*/±10 *g* Dual Axis *i*MEMS® Accelerometers with Digital Output

ADXL202/ADXL210

FEATURES

2-Axis Acceleration Sensor on a Single IC Chip
Measures Static Acceleration as Well as Dynamic Acceleration
Duty Cycle Output with User Adjustable Period
Low Power <0.6 mA
Faster Response than Electrolytic, Mercury or Thermal Tilt Sensors
Bandwidth Adjustment with a Single Capacitor Per Axis
5 m*g* Resolution at 60 Hz Bandwidth
+3 V to +5.25 V Single Supply Operation
1000 *g* Shock Survival

APPLICATIONS

2-Axis Tilt Sensing
Computer Peripherals
Inertial Navigation
Seismic Monitoring
Vehicle Security Systems
Battery Powered Motion Sensing

GENERAL DESCRIPTION

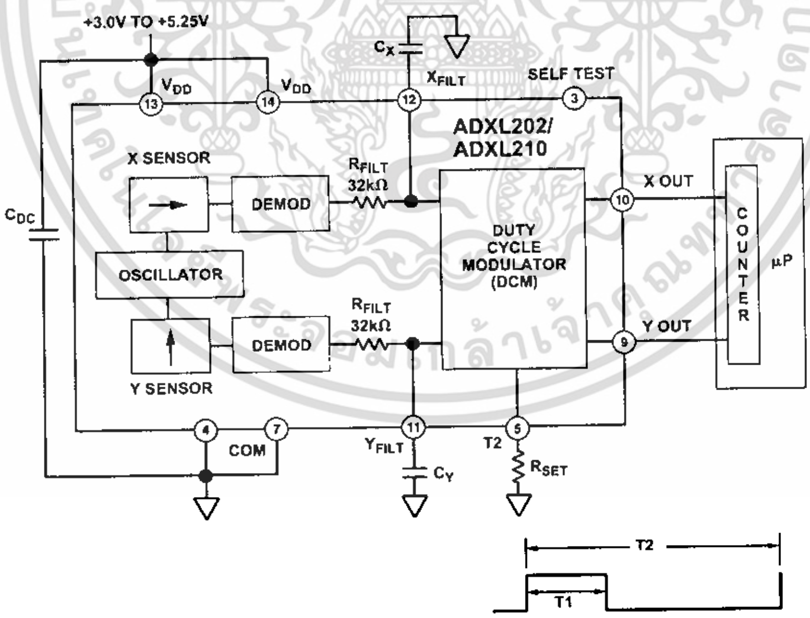
The ADXL202/ADXL210 are low cost, low power, complete 2-axis accelerometers with a measurement range of either ±2 *g*/±10 *g*. The ADXL202/ADXL210 can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are digital signals whose duty cycles (ratio of pulse-width to period) are proportional to the acceleration in each of the 2 sensitive axes. These outputs may be measured directly with a microprocessor counter, requiring no A/D converter or glue logic. The output period is adjustable from 0.5 ms to 10 ms via a single resistor (R_{SET}). If a voltage output is desired, a voltage output proportional to acceleration is available from the X_{FILT} and Y_{FILT} pins, or may be reconstructed by filtering the duty cycle outputs.

The bandwidth of the ADXL202/ADXL210 may be set from 0.01 Hz to 5 kHz via capacitors C_X and C_Y. The typical noise floor is 500 $\mu g/\sqrt{Hz}$ allowing signals below 5 m*g* to be resolved for bandwidths below 60 Hz.

The ADXL202/ADXL210 is available in a hermetic 14-lead Surface Mount CERPAK, specified over the 0°C to +70°C commercial or –40°C to +85°C industrial temperature range.

FUNCTIONAL BLOCK DIAGRAM



$A(g) = (T1/T2 - 0.5)/12.5\%$
$0g$ = 50% DUTY CYCLE
$T2 = R_{SET}/125M\Omega$

*i*MEMS is a registered trademark of Analog Devices, Inc.

REV. B

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: http://www.analog.com
Fax: 781/326-8703

XL202/ADXL210–SPECIFICATIONS ($T_A = T_{MIN}$ to T_{MAX}, T_A = +25°C for J Grade only, V_{DD} = +5 V, R_{SET} = 125 kΩ, Acceleration = 0 *g*, unless otherwise noted)

ıeter	Conditions	ADXL202/JQC/AQC Min	Typ	Max	ADXL210/JQC/AQC Min	Typ	Max	Units
OR INPUT	Each Axis							
ısurement Range[1]		±1.5	±2		±8	±10		*g*
ılinearity	Best Fit Straight Line		0.2			0.2		% of FS
nment Error[2]			±1			±1		Degrees
nment Error	X Sensor to Y Sensor		±0.01			±0.01		Degrees
ısverse Sensitivity[3]			±2			±2		%
ITIVITY	Each Axis							
y Cycle per *g*	T1/T2 @ +25°C	10	12.5	15	3.2	4.0	4.8	%/*g*
ıitivity, Analog Output	At Pins X_{FILT}, Y_{FILT}		312			100		mV/*g*
ıperature Drift[4]	Δ from +25°C		±0.5			±0.5		% Rdg
g BIAS LEVEL	Each Axis							
Duty Cycle	T1/T2	25	50	75	42	50	58	%
al Offset			±2			±2		*g*
Duty Cycle vs. Supply			1.0	4.0		1.0	4.0	%/V
Offset vs. Temperature[4]	Δ from +25°C		2.0			2.0		m*g*/°C
E PERFORMANCE								
se Density[5]	@ +25°C		500	1000		500	1000	μ*g*/√Hz
UENCY RESPONSE								
3 Bandwidth	Duty Cycle Output		500			500		Hz
3 Bandwidth	At Pins X_{FILT}, Y_{FILT}		5			5		kHz
sor Resonant Frequency			10			14		kHz
ER								
T Tolerance	32 kΩ Nominal		±15			±15		%
imum Capacitance	At X_{FILT}, Y_{FILT}	1000			1000			pF
TEST								
y Cycle Change	Self-Test "0" to "1"		10			10		%
CYCLE OUTPUT STAGE								
			125 MΩ/R_{SET}			125 MΩ/R_{SET}		
Tolerance	R_{SET} = 125 kΩ	0.7		1.3	0.7		1.3	kHz
put High Voltage	I = 25 μA	V_S – 200 mV			V_S – 200 mV			mV
put Low Voltage	I = 25 μA			200			200	mV
Drift vs. Temperature			35			35		ppm/°C
/Fall Time			200			200		ns
ER SUPPLY								
rating Voltage Range		3.0		5.25	2.7		5.25	V
cified Performance		4.75		5.25	4.75		5.25	V
escent Supply Current			0.6	1.0		0.6	1.0	mA
n-On Time[6]	To 99%		160 C_{FILT} + 0.3			160 C_{FILT} + 0.3		ms
ERATURE RANGE								
rating Range	JQC	0		+70	0		+70	°C
cified Performance	AQC	–40		+85	–40		+85	°C

s
combinations of offset and sensitivity variation.
ent error is specified as the angle between the true and indicated axis of sensitivity.
erse sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.
cation refers to the maximum change in parameter from its initial at +25°C to its worst case value at T_{MIN} to T_{MAX}.
density (μ*g*/√Hz) is the average noise at any frequency in the bandwidth of the part.
n μF. Addition of filter capacitor will increase turn on time. Please see the Application section on power cycling.
and max specifications are guaranteed. Typical specifications are not tested or guaranteed.
ations subject to change without notice.

BSOLUTE MAXIMUM RATINGS*

cceleration (Any Axis, Unpowered for 0.5 ms) . . . . . . 1000 *g*
cceleration (Any Axis, Powered for 0.5 ms) . . . . 500 *g*
V_S . . . . –0.3 V to +7.0 V
utput Short Circuit Duration
(Any Pin to Common) . . . . Indefinite
perating Temperature . . . . –55°C to +125°C
torage Temperature . . . . –65°C to +150°C

tresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; the functional operation of he device at these or any other conditions above those indicated in the operational ections of this specification is not implied. Exposure to absolute maximum rating onditions for extended periods may affect device reliability.

)rops onto hard surfaces can cause shocks of greater than 1000 *g* nd exceed the absolute maximum rating of the device. Care hould be exercised in handling to avoid damage.

PIN CONFIGURATION

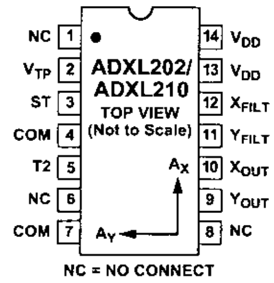


Figure 1 shows the response of the ADXL202 to the Earth's gravitational field. The output values shown are nominal. They are presented to show the user what type of response to expect from each of the output pins due to changes in orientation with respect to the Earth. The ADXL210 reacts similarly with output changes appropriate to its scale.

PIN FUNCTION DESCRIPTIONS

'in	Name	Description
	NC	No Connect
	V_{TP}	Test Point, Do Not Connect
	ST	Self Test
	COM	Common
	T2	Connect R_{SET} to Set T2 Period
	NC	No Connect
	COM	Common
	NC	No Connect
	Y_{OUT}	Y Axis Duty Cycle Output
0	X_{OUT}	X Axis Duty Cycle Output
1	Y_{FILT}	Connect Capacitor for Y Filter
2	X_{FILT}	Connect Capacitor for X Filter
3	V_{DD}	+3 V to +5.25 V, Connect to 14
4	V_{DD}	+3 V to +5.25 V, Connect to 13

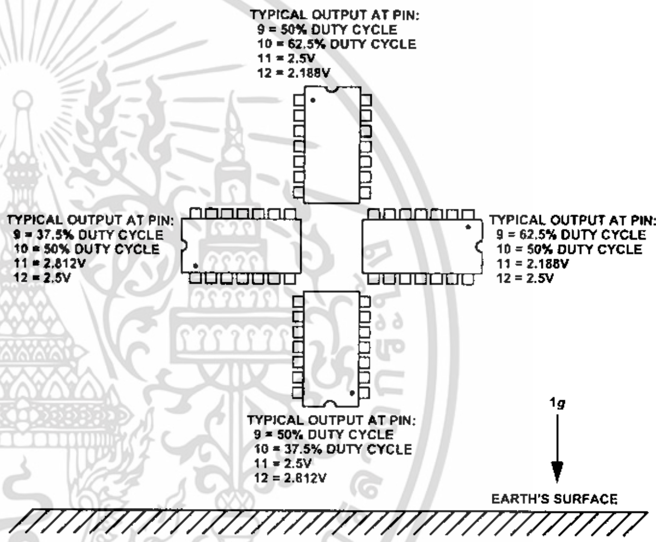


Figure 1. ADXL202/ADXL210 Nominal Response Due to Gravity

ACKAGE CHARACTERISTICS

ackage	θ_{JA}	θ_{JC}	Device Weight
-Lead CERPAK	110°C/W	30°C/W	5 Grams

ORDERING GUIDE

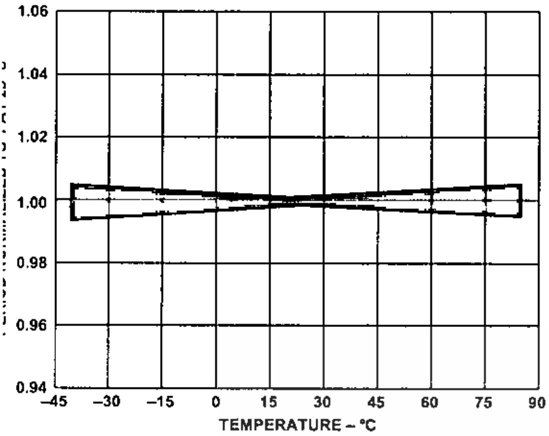
Model	*g* Range	Temperature Range	Package Description	Package Option
ADXL202JQC	±2	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL202AQC	±2	–40°C to +85°C	14-Lead CERPAK	QC-14
ADXL210JQC	±10	0°C to +70°C	14-Lead CERPAK	QC-14
ADXL210AQC	±10	–40°C to +85°C	14-Lead CERPAK	QC-14

AUTION

D (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily cumulate on the human body and test equipment and can discharge without detection. though the ADXL202/ADXL210 features proprietary ESD protection circuitry, permanent mage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper D precautions are recommended to avoid performance degradation or loss of functionality.

WARNING!
ESD SENSITIVE DEVICE

ICAL CHARACTERISTICS (@ +25°C R_{SET} = 125 kΩ, V_{DD} = +5 V, unless otherwise noted)



re 2. Normalized DCM Period (T2) vs. Temperature

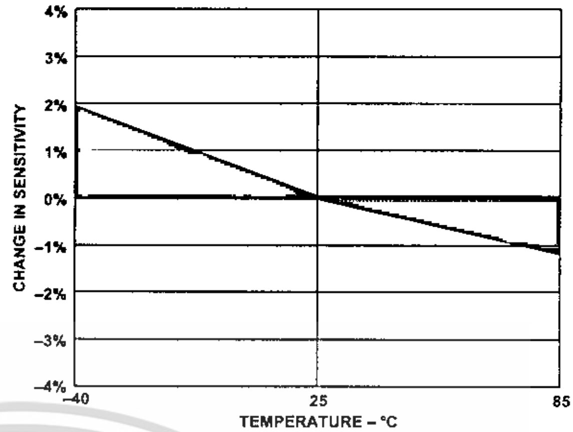
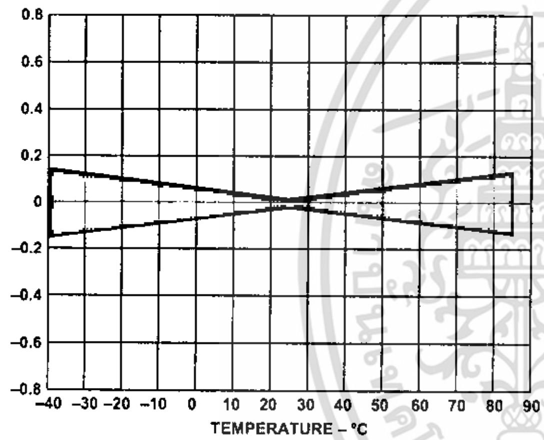


Figure 5. Typical X Axis Sensitivity Drift Due to Temperature



igure 3. Typical Zero g Offset vs. Temperature

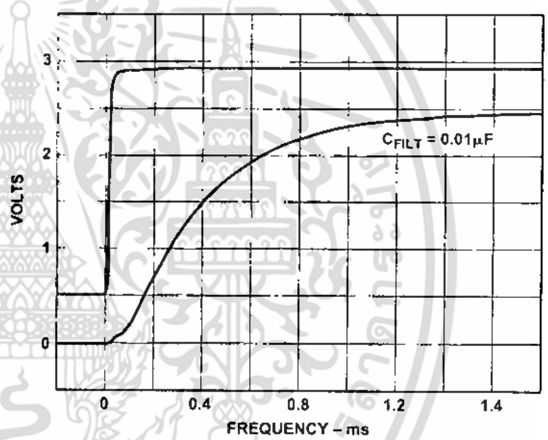
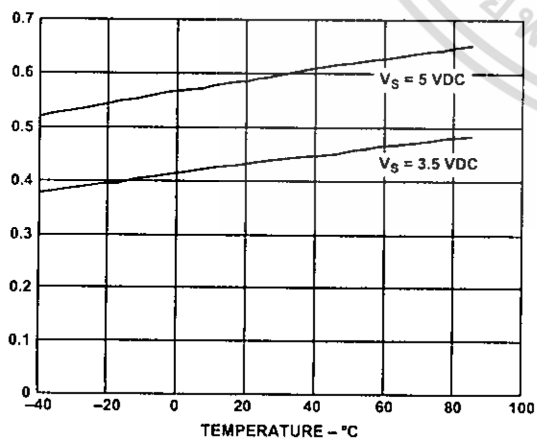


Figure 6. Typical Turn-On Time



gure 4. Typical Supply Current vs. Temperature

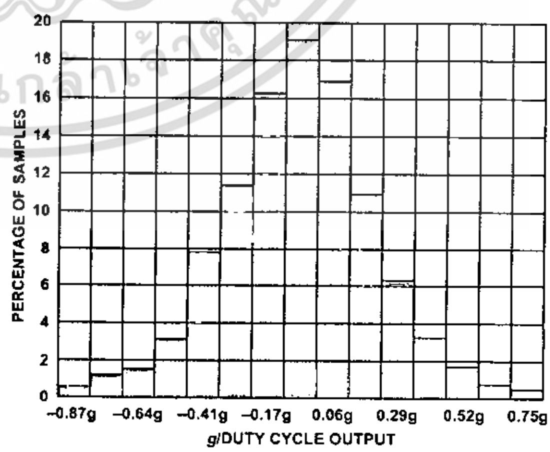


Figure 7. Typical Zero g Distribution at +25°C

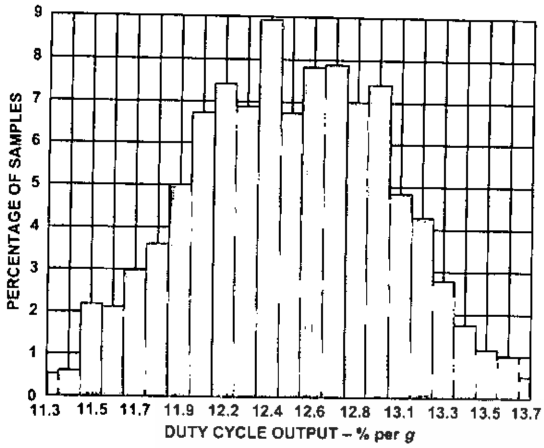


Figure 8. Typical Sensitivity per g at +25°C

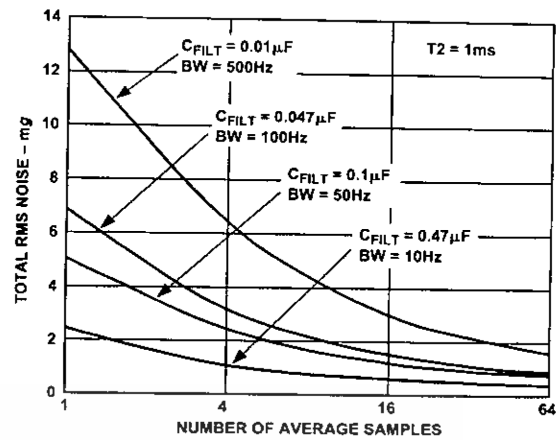


Figure 10. Typical Noise at Digital Outputs

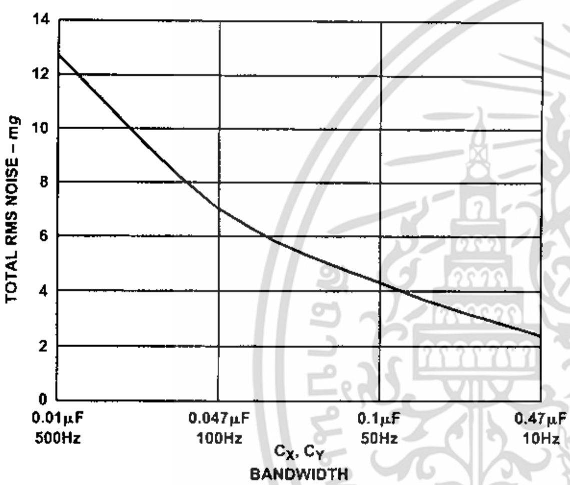


Figure 9. Typical Noise at X_{FILT} Output

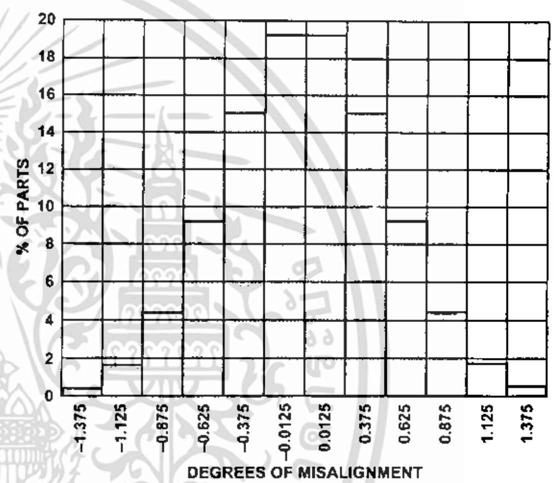


Figure 11. Rotational Die Alignment

:INITIONS

	Length of the "on" portion of the cycle.
	Length of the total cycle.
y Cycle	Ratio of the "on" time (T1) of the cycle to the total cycle (T2). Defined as T1/T2 for the ADXL202/ADXL210.
ewidth	Time period of the "on" pulse. Defined as T1 for the ADXL202/ADXL210.

:ORY OF OPERATION

ADXL202/ADXL210 are complete dual axis acceleration surement systems on a single monolithic IC. They contain a silicon surface-micromachined sensor and signal condition- :ircuitry to implement an open loop acceleration measure- t architecture. For each axis, an output circuit converts the og signal to a duty cycle modulated (DCM) digital signal can be decoded with a counter/timer port on a micropro- or. The ADXL202/ADXL210 are capable of measuring . positive and negative accelerations to a maximum level of ' or ±10 *g*. The accelerometer measures static acceleration :s such as gravity, allowing it to be used as a tilt sensor.

sensor is a surface micromachined polysilicon structure : on top of the silicon wafer. Polysilicon springs suspend the :ture over the surface of the wafer and provide a resistance 1st acceleration forces. Deflection of the structure is mea- d using a differential capacitor that consists of independent l plates and central plates attached to the moving mass. The l plates are driven by 180° out of phase square waves. An leration will deflect the beam and unbalance the differential citor, resulting in an output square wave whose amplitude oportional to acceleration. Phase sensitive demodulation niques are then used to rectify the signal and determine the :tion of the acceleration.

output of the demodulator drives a duty cycle modulator M) stage through a 32 kΩ resistor. At this point a pin is able on each channel to allow the user to set the signal lwidth of the device by adding a capacitor. This filtering oves measurement resolution and helps prevent aliasing.

: being low-pass filtered, the analog signal is converted to a cycle modulated signal by the DCM stage. A single resistor the period for a complete cycle (T2), which can be set be- n 0.5 ms and 10 ms (see Figure 12). A 0 *g* acceleration uces a nominally 50% duty cycle. The acceleration signal oe determined by measuring the length of the T1 and T2 :s with a counter/timer or with a polling loop using a low microcontroller.

nalog output voltage can be obtained either by buffering the ıl from the X_{FILT} and Y_{FILT} pin, or by passing the duty cycle :l through an RC filter to reconstruct the dc value.

ADXL202/ADXL210 will operate with supply voltages as ıs 3.0 V or as high as 5.25 V.

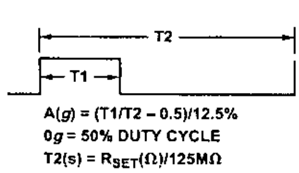


Figure 12. Typical Output Duty Cycle

APPLICATIONS

POWER SUPPLY DECOUPLING

For most applications a single 0.1 µF capacitor, C_{DC}, will adequately decouple the accelerometer from signal and noise on the power supply. However, in some cases, especially where digital devices such as microcontrollers share the same power supply, digital noise on the supply may cause interference on the ADXL202/ADXL210 output. This is often observed as a slowly undulating fluctuation of voltage at X_{FILT} and Y_{FILT}. If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite beads, may be inserted in the ADXL202/ADXL210's supply line.

DESIGN PROCEDURE FOR THE ADXL202/ADXL210

The design procedure for using the ADXL202/ADXL210 with a duty cycle output involves selecting a duty cycle period and a filter capacitor. A proper design will take into account the application requirements for bandwidth, signal resolution and acquisition time, as discussed in the following sections.

V_{DD}

The ADXL202/ADXL210 have two power supply (V_{DD}) Pins: 13 and 14. These two pins should be connected directly together.

COM

The ADXL202/ADXL210 have two commons, Pins 4 and 7. These two pins should be connected directly together and Pin 7 grounded.

V_{TP}

This pin is to be left open; make no connections of any kind to this pin.

Decoupling Capacitor C_{DC}

A 0.1 µF capacitor is recommended from V_{DD} to COM for power supply decoupling.

ST

The ST pin controls the self-test feature. When this pin is set to V_{DD}, an electrostatic force is exerted on the beam of the accelerometer. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output will be 10% at the duty cycle outputs (corresponding to 800 m*g*). This pin may be left open circuit or connected to common in normal use.

Duty Cycle Decoding

The ADXL202/ADXL210's digital output is a duty cycle modulator. Acceleration is proportional to the ratio T1/T2. The nominal output of the ADXL202 is:

0 *g* = 50% *Duty Cycle*

Scale factor is 12.5% Duty Cycle Change per *g*

The nominal output of the ADXL210 is:

0 *g* = 50% *Duty Cycle*

Scale factor is 4% Duty Cycle Change per *g*

These nominal values are affected by the initial tolerance of the device including zero *g* offset error and sensitivity error.

T2 does not have to be measured for every measurement cycle. It need only be updated to account for changes due to temperature, (a relatively slow process). Since the T2 time period is shared by both X and Y channels, it is necessary only to measure it on one channel of the ADXL202/ADXL210. Decoding algorithms for various microcontrollers have been developed. Consult the appropriate Application Note.

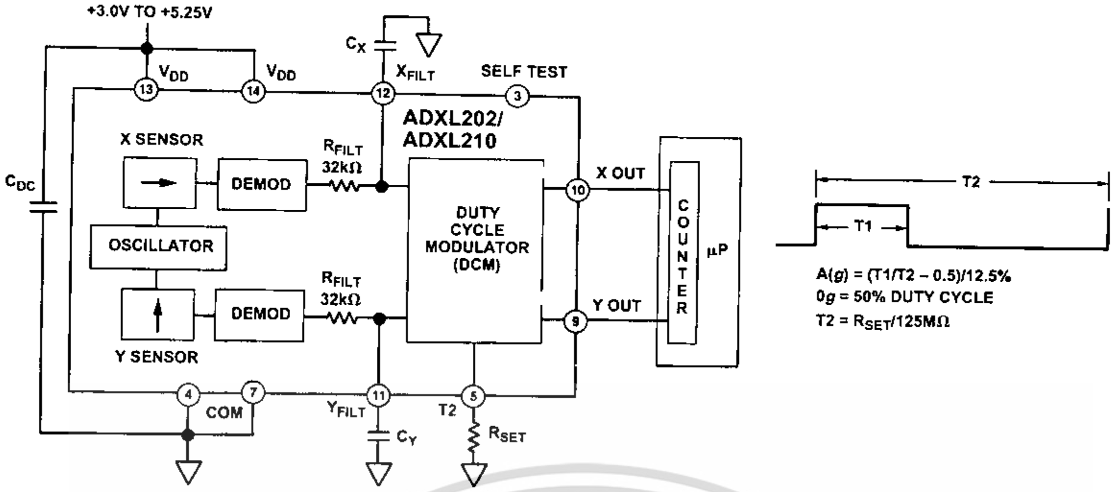


Figure 13. Block Diagram

Setting the Bandwidth Using C_X and C_Y

The ADXL202/ADXL210 have provisions for bandlimiting the X_{FILT} and Y_{FILT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is:

$$F_{-3\,dB} = \frac{1}{\left(2\,\pi\,(32\ k\Omega) \times C(x,y)\right)}$$

or, more simply, $F_{-3\,dB} = \dfrac{5\,\mu F}{C_{(X,Y)}}$

The tolerance of the internal resistor (R_{FILT}), can vary as much as ±25% of its nominal value of 32 kΩ; so the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for $C_{(X,\,Y)}$ is required in all cases.

Table I. Filter Capacitor Selection, C_X and C_Y

Bandwidth	Capacitor Value
10 Hz	0.47 µF
50 Hz	0.10 µF
100 Hz	0.05 µF
200 Hz	0.027 µF
500 Hz	0.01 µF
5 kHz	0.001 µF

Setting the DCM Period with R_{SET}

The period of the DCM output is set for both channels by a single resistor from R_{SET} to ground. The equation for the period is:

$$T2 = \frac{R_{SET}\ (\Omega)}{125\ M\Omega}$$

A 125 kΩ resistor will set the duty cycle repetition rate to approximately 1 kHz, or 1 ms. The device is designed to operate at duty cycle periods between 0.5 ms and 10 ms.

Table II. Resistor Values to Set T2

T2	R_{SET}
1 ms	125 kΩ
2 ms	250 kΩ
5 ms	625 kΩ
10 ms	1.25 MΩ

Note that the R_{SET} should always be included, even if only an analog output is desired. Use an R_{SET} value between 500 kΩ and 2 MΩ when taking the output from X_{FILT} or Y_{FILT}. The R_{SET} resistor should be place close to the T2 Pin to minimize parasitic capacitance at this node.

Selecting the Right Accelerometer

For most tilt sensing applications the ADXL202 is the most appropriate accelerometer. Its higher sensitivity (12.5%/*g* allows the user to use a lower speed counter for PWM decoding while maintaining high resolution. The ADXL210 should be used in applications where accelerations of greater than ±2 *g* are expected.

MICROCOMPUTER INTERFACES

The ADXL202/ADXL210 were specifically designed to work with low cost microcontrollers. Specific code sets, reference designs, and application notes are available from the factory. This section will outline a general design procedure and discuss the various trade-offs that need to be considered.

The designer should have some idea of the required performance of the system in terms of:

Resolution: the smallest signal change that needs to be detected.

Bandwidth: the highest frequency that needs to be detected.

Acquisition Time: the time that will be available to acquire the signal on each axis.

These requirements will help to determine the accelerometer bandwidth, the speed of the microcontroller clock and the length of the T2 period.

When selecting a microcontroller it is helpful to have a counter timer port available. The microcontroller should have provisions for software calibration. While the ADXL202/ADXL210 are highly accurate accelerometers, they have a wide tolerance for al offset. The easiest way to null this offset is with a calibra- factor saved on the microcontroller or by a user calibration ero g. In the case where the offset is calibrated during manu- ıre, there are several options, including external EEPROM microcontrollers with "one-time programmable" features.

ıIGN TRADE-OFFS FOR SELECTING FILTER ıRACTERISTICS: THE NOISE/BW TRADE-OFF

accelerometer bandwidth selected will determine the mea- ment resolution (smallest detectable acceleration). Filtering ɔe used to lower the noise floor and improve the resolution .e accelerometer. Resolution is dependent on both the ana- ilter bandwidth at X_{FILT} and Y_{FILT} and on the speed of the ocontroller counter.

analog output of the ADXL202/ADXL210 has a typical lwidth of 5 kHz, much higher than the duty cycle stage is ble of converting. The user must filter the signal at this t to limit aliasing errors. To minimize DCM errors the ɔg bandwidth should be less than 1/10 the DCM frequency. og bandwidth may be increased to up to 1/2 the DCM ıency in many applications. This will result in greater dy- ic error generated at the DCM.

analog bandwidth may be further decreased to reduce noise improve resolution. The ADXL202/ADXL210 noise has :haracteristics of white Gaussian noise that contributes lly at all frequencies and is described in terms of µg per root ..e., the noise is proportional to the square root of the band- ı of the accelerometer. It is recommended that the user limit width to the lowest frequency needed by the application, to mize the resolution and dynamic range of the accelerometer.

the single pole roll-off characteristic, the typical noise of ıDXL202/ADXL210 is determined by the following equation:

$$Noise\,(rms) = \left(500\ \mu g/\sqrt{Hz}\right) \times \left(\sqrt{BW \times 1.5}\right)$$

)0 Hz the noise will be:

$$oise\,(rms) = \left(500\ \mu g/\sqrt{Hz}\right) \times \left(\sqrt{100 \times (1.5)}\right) = 6.12\ mg$$

ı the peak value of the noise is desired. Peak-to-peak noise ınly be estimated by statistical methods. Table III is useful stimating the probabilities of exceeding various peak values, ı the rms value.

Table III. Estimation of Peak-to-Peak Noise

Nominal Peak-to-Peak Value	% of Time that Noise Will Exceed Nominal Peak-to-Peak Value
2.0 × rms	32%
4.0 × rms	4.6%
6.0 × rms	0.27%
8.0 × rms	0.006%

ɔeak-to-peak noise value will give the best estimate of the :tainty in a single measurement.

Table IV gives typical noise output of the ADXL202/ADXL210 for various C_X and C_Y values.

Table IV. Filter Capacitor Selection, C_X and C_Y

Bandwidth	C_X, C_Y	rms Noise	Peak-to-Peak Noise Estimate 95% Probability (rms × 4)
10 Hz	0.47 µF	1.9 mg	7.6 mg
50 Hz	0.10 µF	4.3 mg	17.2 mg
100 Hz	0.05 µF	6.1 mg	24.4 mg
200 Hz	0.027 µF	8.7 mg	35.8 mg
500 Hz	0.01 µF	13.7 mg	54.8 mg

CHOOSING T2 AND COUNTER FREQUENCY: DESIGN TRADE-OFFS

The noise level is one determinant of accelerometer resolution. The second relates to the measurement resolution of the counter when decoding the duty cycle output.

The ADXL202/ADXL210's duty cycle converter has a resolution of approximately 14 bits; better resolution than the accelerometer itself. The actual resolution of the acceleration signal is, however, limited by the time resolution of the counting devices used to decode the duty cycle. The faster the counter clock, the higher the resolution of the duty cycle and the shorter the T2 period can be for a given resolution. The following table shows some of the trade-offs. It is important to note that this is the resolution due to the microprocessors's counter. It is probable that the accelerometer's noise floor may set the lower limit on the resolution, as discussed in the previous section.

Table V. Trade-Offs Between Microcontroller Counter Rate, T2 Period and Resolution of Duty Cycle Modulator

T2 (ms)	R_{SET} (kΩ)	ADXL202/ ADXL210 Sample Rate	Counter-Clock Rate (MHz)	Counts per T2 Cycle	Counts per g	Resolution (mg)
1.0	124	1000	2.0	2000	250	4.0
1.0	124	1000	1.0	1000	125	8.0
1.0	124	1000	0.5	500	62.5	16.0
5.0	625	200	2.0	10000	1250	0.8
5.0	625	200	1.0	5000	625	1.6
5.0	625	200	0.5	2500	312.5	3.2
10.0	1250	100	2.0	20000	2500	0.4
10.0	1250	100	1.0	10000	1250	0.8
10.0	1250	100	0.5	5000	625	1.6

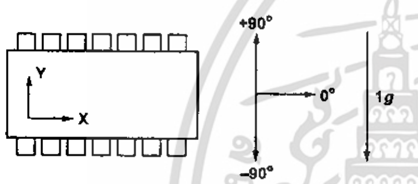
STRATEGIES FOR USING THE DUTY CYCLE OUTPUT WITH MICROCONTROLLERS

Application notes outlining various strategies for using the duty cycle output with low cost microcontrollers are available from the factory.

USING THE ADXL202/ADXL210 AS A DUAL AXIS TILT SENSOR

One of the most popular applications of the ADXL202/ADXL210 is tilt measurement. An accelerometer uses the force of gravity as an input vector to determine orientation of an object in space.

An accelerometer is most sensitive to tilt when its sensitive axis is perpendicular to the force of gravity, i.e., parallel to the earth's surface. At this orientation its sensitivity to changes in tilt is highest. When the accelerometer is oriented on axis to gravity, i.e., near its +1 *g* or –1 *g* reading, the change in output acceleration per degree of tilt is negligible. When the accelerometer is perpendicular to gravity, its output will change nearly 17.5 m*g* per degree of tilt, but at 45° degrees it is changing only at 12.2 m*g* per degree and resolution declines. The following table illustrates the changes in the X and Y axes as the device is tilted ±90° through gravity.



X AXIS ORIENTATION TO HORIZON (°)	X OUTPUT X OUTPUT (g)	X OUTPUT Δ PER DEGREE OF TILT (mg)	Y OUTPUT (g) Y OUTPUT (g)	Y OUTPUT (g) Δ PER DEGREE OF TILT (mg)
–90	–1.000	–0.2	0.000	17.5
–75	–0.966	4.4	0.259	16.9
–60	–0.866	8.6	0.500	15.2
–45	–0.707	12.2	0.707	12.4
–30	–0.500	15.0	0.866	8.9
–15	–0.259	16.8	0.966	4.7
0	0.000	17.5	1.000	0.2
15	0.259	16.9	0.966	–4.4
30	0.500	15.2	0.866	–8.6
45	0.707	12.4	0.707	–12.2
60	0.866	8.9	0.500	–15.0
75	0.966	4.7	0.259	–16.8
90	1.000	0.2	0.000	–17.5

Figure 14. How the X and Y Axes Respond to Changes in Tilt

A DUAL AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT

When the accelerometer is oriented so both its X and Y axes are parallel to the earth's surface it can be used as a two axis tilt sensor with a roll and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between –1 *g* and +1 *g*, the output tilt in degrees is calculated as follows:

$$Pitch = ASIN\ (Ax/1\ g)$$
$$Roll = ASIN\ (Ay/1\ g)$$

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than ±1 *g* due to vibration, shock or other accelerations.

MEASURING 360° OF TILT

It is possible to measure a full 360° of orientation through gravity by using two accelerometers oriented perpendicular to one another (see Figure 15). When one sensor is reading a maximum change in output per degree, the other is at its minimum.

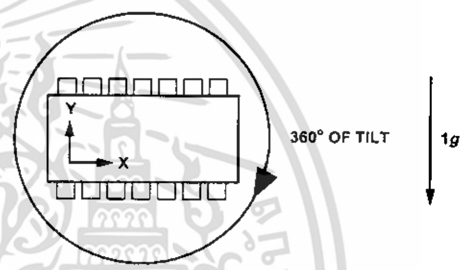


Figure 15. Using a Two-Axis Accelerometer to Measure 360° of Tilt

NG THE ANALOG OUTPUT

ADXL202/ADXL210 was specifically designed for use with igital outputs, but has provisions to provide analog outputs ell.

y Cycle Filtering

ınalog output can be reconstructed by filtering the duty : output. This technique requires only passive components. duty cycle period (T2) should be set to 1 ms. An RC filter a 3 dB point at least a factor of 10 less than the duty cycle ıency is connected to the duty cycle output. The filter resis- hould be no less than 100 kΩ to prevent loading of the ut stage. The analog output signal will be ratiometric to the ıly voltage. The advantage of this method is an output scale ır of approximately double the analog output. Its disadvan- is that the frequency response will be lower than when ; the X_{FILT}, Y_{FILT} output.

r, Y_{FILT} Output

second method is to use the analog output present at the T and Y_{FILT} pin. Unfortunately, these pins have a 32 kΩ ut impedance and are not designed to drive a load directly. ıp amp follower may be required to buffer this pin. The ntage of this method is that the full 5 kHz bandwidth of the lerometer is available to the user. A capacitor still must be d at this point for filtering. The duty cycle converter should ept running by using R_{SET} <10 MΩ. Note that the acceler- ter offset and sensitivity are ratiometric to the supply volt- The offset and sensitivity are nominally:

)ffset = $V_{DD}/2$	2.5 V at +5 V
L202 Sensitivity = (60 mV × V_S)/g	300 mV/g at +5 V, V_{DD}
L210 Sensitivity = (20 mV × V_S)/g	100 mV/g at +5 V, V_{DD}

NG THE ADXL202/ADXL210 IN VERY LOW POWER LICATIONS

pplication note outlining low power strategies for the L202/ADXL210 is available. Some key points are pre- d here. It is possible to reduce the ADXL202/ADXL210's ıge current from 0.6 mA to less than 20 µA by using the wing techniques:

ower Cycle the accelerometer.

un the accelerometer at a Lower Voltage, (Down to 3 V).

er Cycling with an External A/D

nding on the value of the X_{FILT} capacitor, the ADXL202/ L210 is capable of turning on and giving a good reading in ıs. Most microcontroller based A/Ds can acquire a reading other 25 µs. Thus it is possible to turn on the ADXL202/ L210 and take a reading in <2 ms. If we assume that a z sample rate is sufficient, the total current required to 20 samples is 2 ms × 20 samples/s × 0.6 mA = 24 µA aver- urrent. Running the part at 3 V will reduce the supply nt from 0.6 mA to 0.4 mA, bringing the average current ı to 16 µA.

A/D should read the analog output of the ADXL202/ L210 at the X_{FILT} and Y_{FILT} pins. A buffer amplifier is nmended, and may be required in any case to amplify the g output to give enough resolution with an 8-bit to 10-bit erter.

Power Cycling When Using the Digital Output

An alternative is to run the microcontroller at a higher clock rate and put it into shutdown between readings, allowing the use of the digital output. In this approach the ADXL202/ ADXL210 should be set at its fastest sample rate (T2 = 0.5 ms), with a 500 Hz filter at X_{FILT} and Y_{FILT}. The concept is to acquire a reading as quickly as possible and then shut down the ADXL202/ADXL210 and the microcontroller until the next sample is needed.

In either of the above approaches, the ADXL202/ADXL210 can be turned on and off directly using a digital port pin on the microcontroller to power the accelerometer without additional components. The port should be used to switch the common pin of the accelerometer so the port pin is "pulling down."

CALIBRATING THE ADXL202/ADXL210

The initial value of the offset and scale factor for the ADXL202/ ADXL210 will require calibration for applications such as tilt measurement. The ADXL202/ADXL210 architecture has been designed so that these calibrations take place in the software of the microcontroller used to decode the duty cycle signal. Calibration factors can be stored in EEPROM or determined at turn-on and saved in dynamic memory.

For low *g* applications, the force of gravity is the most stable, accurate and convenient acceleration reference available. A reading of the 0 *g* point can be determined by orientating the device parallel to the earth's surface and then reading the output.

A more accurate calibration method is to make a measurements at +1 *g* and –1 *g*. The sensitivity can be determined by the two measurements.

To calibrate, the accelerometer's measurement axis is pointed directly at the earth. The 1 *g* reading is saved and the sensor is turned 180° to measure –1 *g*. Using the two readings, the sensitivity is:

Let A = Accelerometer output with axis oriented to +1 *g*
Let B = Accelerometer output with axis oriented to –1 *g then*:

$$Sensitivity = [A - B]/2\,g$$

For example, if the +1 *g* reading (A) is 55% duty cycle and the –1 *g* reading (B) is 32% duty cycle, then:

$$Sensitivity = [55\% - 32\%]/2\,g = 11.5\%/g$$

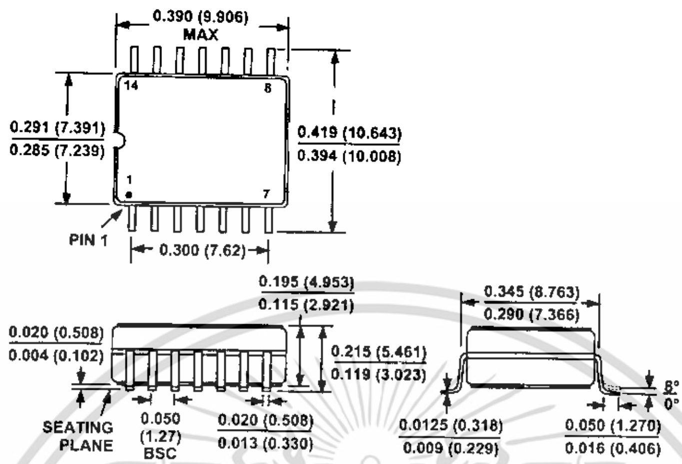
These equations apply whether the output is analog, or duty cycle.

Application notes outlining algorithms for calculating acceleration from duty cycle and automated calibration routines are available from the factory.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

14-Lead CERPAK (QC-14)



PRINTED IN U.S.A.

ANALOG DEVICES

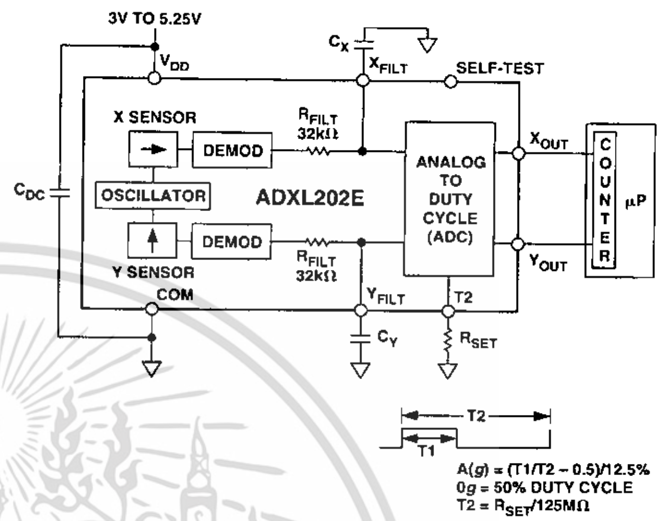
Low-Cost ±2 *g* Dual-Axis Accelerometer with Duty Cycle Output

ADXL202E*

FEATURES
- 2-Axis Acceleration Sensor on a Single IC Chip
- 5 mm × 5 mm × 2 mm Ultrasmall Chip Scale Package
- 2 m*g* Resolution at 60 Hz
- Low-Power < 0.6 mA
- Direct Interface to Low-Cost Microcontrollers via Duty Cycle Output
- BW Adjustment with a Single Capacitor
- 3 V to 5.25 V Single Supply Operation
- 1000 *g* Shock Survival

APPLICATIONS
- 2-Axis Tilt Sensing with Faster Response than Electrolytic, Mercury, or Thermal Sensors
- Computer Peripherals
- Information Appliances
- Alarms and Motion Detectors
- Disk Drives
- Vehicle Security

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The ADXL202E is a low-cost, low-power, complete 2-axis accelerometer with a digital output, all on a single monolithic IC. It is an improved version of the ADXL202AQC/JQC. The ADXL202E will measure accelerations with a full-scale range of ±2 *g*. The ADXL202E can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are analog voltage or digital signals whose duty cycles (ratio of pulsewidth to period) are proportional to acceleration. The duty cycle outputs can be directly measured by a microprocessor counter, without an A/D converter or glue logic. The duty cycle period is adjustable from 0.5 ms to 10 ms via a single resistor (R_{SET}).

The typical noise floor is 200 $\mu g/\sqrt{Hz}$, allowing signals below 2 m*g* (at 60 Hz bandwidth) to be resolved.

The bandwidth of the accelerometer is set with capacitors C_X and C_Y at the X_{FILT} and Y_{FILT} pins. An analog output can be reconstructed by filtering the duty cycle output.

The ADXL202E is available in 5 mm × 5 mm × 2 mm 8-lead hermetic LCC package.

*Patents Pending

REV. A

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: http://www.analog.com
Fax: 781/326-8703

DXL202E–SPECIFICATIONS ($T_A = T_{MIN}$ to T_{MAX}, $T_A = 25°C$ for J Grade only, $V_{DD} = 5$ V, $R_{SET} = 125$ kΩ, Acceleration = 0 *g*, unless otherwise noted.)

:ameter	Conditions	TPC[1] Graph	ADXL202JE Min	Typ	Max	ADXL202AE Min	Typ	Max	Unit
NSOR INPUT	Each Axis								
Measurement Range[2]			±2			±2			*g*
Nonlinearity	Best Fit Straight Line			0.2			0.2		% of FS
Alignment Error[3]		X		±1			±1		Degrees
Alignment Error	X Sensor to Y Sensor			0.01			0.01		Degrees
Cross-Axis Sensitivity[4]		X		±2			±2		%
NSITIVITY	Each Axis								
Duty Cycle per *g*	T1/T2, V_{DD} = 5 V	X	10.5	12.5	14.5	10	12.5	15	%/*g*
Duty Cycle per *g*	T1/T2, V_{DD} = 3 V	X	9.0	11	13.0	8.5	11	13.5	%/*g*
ensitivity X_{FILT}, Y_{FILT}	V_{DD} = 5 V	X	265	312	360	250	312	375	mV/*g*
ensitivity X_{FILT}, Y_{FILT}	V_{DD} = 3 V	X	140	167	195	140	167	200	mV/*g*
emperature Drift[5]	Delta from 25°C	X		±0.5			±0.5		%
RO *g* BIAS LEVEL	Each Axis								
g Duty Cycle	T1/T2, V_{DD} = 5 V	X	34	50	66	30	50	70	%
g Duty Cycle	T1/T2, V_{DD} = 3 V	X	31	50	69	31	50	69	%
g Voltage X_{FILT}, Y_{FILT}	V_{DD} = 5 V	X	2.1	2.5	2.9	2.0	2.5	3.0	V
g Voltage X_{FILT}, Y_{FILT}	V_{DD} = 3 V	X	1.2	1.5	1.8	1.2	1.5	1.8	V
g Duty Cycle vs. Supply		X		1.0	4.0		1.0	4.0	%/V
g Offset vs. Temperature[5]	Delta from 25°C	X		2.0			2.0		m*g*/°C
ISE PERFORMANCE									
Noise Density	@ 25°C	X		200			200	1000	μ*g*√Hz rms
EQUENCY RESPONSE									
dB Bandwidth	At Pins X_{FILT}, Y_{FILT}			6			6		kHz
ensor Resonant Frequency				10			10		kHz
TER									
$_{FILT}$ Tolerance	32 kΩ Nominal			±15			±15		%
Minimum Capacitance	At Pins X_{FILT}, Y_{FILT}		1000			1000			pF
F-TEST									
Duty Cycle Change	Self-Test "0" to "1"			10			10		%
TY CYCLE OUTPUT STAGE									
$_{SET}$	R_{SET} = 125 kΩ		0.7		1.3	0.7		1.3	kHz
Output High Voltage	I = 25 μA		V_S – 200 mV			V_S – 200 mV			V
Output Low Voltage	I = 25 μA				200			200	mV
2 Drift vs. Temperature				50			50		ppm/°C
Rise/Fall Time				200			200		ns
WER SUPPLY									
Operating Voltage Range			3		5.25	3.0		5.25	V
Quiescent Supply Current				0.6	1.0		0.6	1.0	mA
Turn-On Time	C_{FILT} in μF			160 × C_{FILT} + 0.3			160 × C_{FILT} + 0.3		ms
MPERATURE RANGE									
pecified Performance AE						–40		+85	°C
Operating Range			0		70	–40		+85	°C

TES
pical Performance Characteristics.
ıaranteed by measurement of initial offset and sensitivity.
gnment error is specified as the angle between the true and indicated axis of sensitivity (see TPC 15).
oss-axis sensitivity is the algebraic sum of the alignment and the inherent sensitivity errors.
fined as the output change from ambient to maximum temperature or ambient to minimum temperature.

cifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

Acceleration (Any Axis, Unpowered for 0.5 ms) . . . . . . 1000 *g*
Acceleration (Any Axis, Powered for 0.5 ms) . . . . 500 *g*
$+V_S$ . . . . –0.3 V to +6.0 V
Output Short Circuit Duration, (Any Pin to Common) . . . . Indefinite
Operating Temperature . . . . –55°C to +125°C
Storage Temperature . . . . –65°C to +150°C

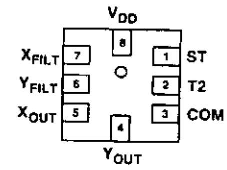
*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicate in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Drops onto hard surfaces can cause shocks of greater than 1000 *g* and exceed the absolute maximum rating of the device. Care should be exercised in handling to avoid damage.

Package Characteristics

Package Weight	θ_{JA}	θ_{JC}	Device
8-Lead LCC	120°C/W	tbd°C/W	<1.0 grams

PIN CONFIGURATION



BOTTOM VIEW

PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Description
1	ST	Self-Test
2	T2	Connect R_{SET} to Set T2 Period
3	COM	Common
4	Y_{OUT}	Y-Channel Duty Cycle Output
5	X_{OUT}	X-Channel Duty Cycle Output
6	Y_{FILT}	Y-Channel Filter Pin
7	X_{FILT}	X-Channel Filter Pin
8	V_{DD}	3 V to 5.25 V

ORDERING GUIDE

Model	No. of Axes	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL202JE	2	3 V to 5 V	0 to 70°C	8-Lead LCC	E-8
ADXL202AE	2	3 V to 5 V	–40°C to +85°C	8-Lead LCC	E-8

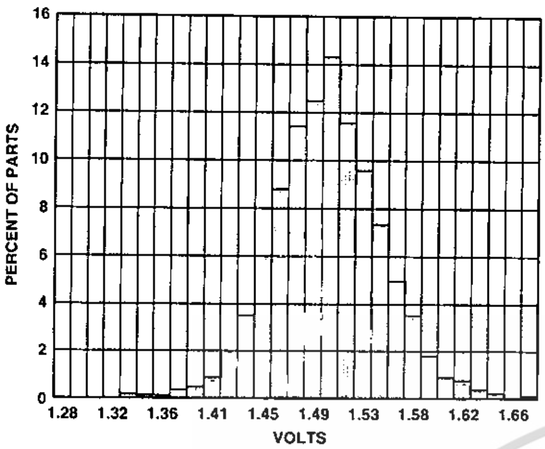
CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADXL202E features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high-energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

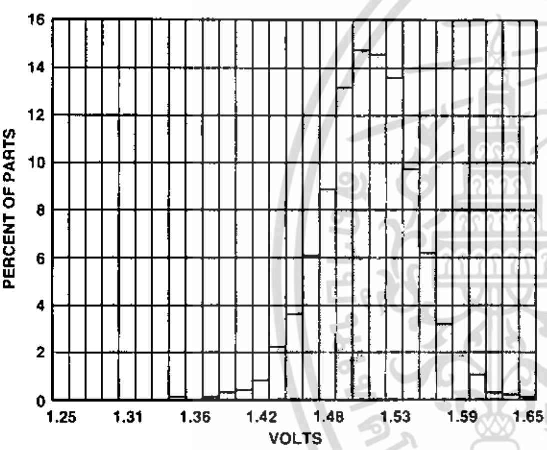
DXL202E—Typical Performance Characteristics*

$V_{DD} = 3\ V$

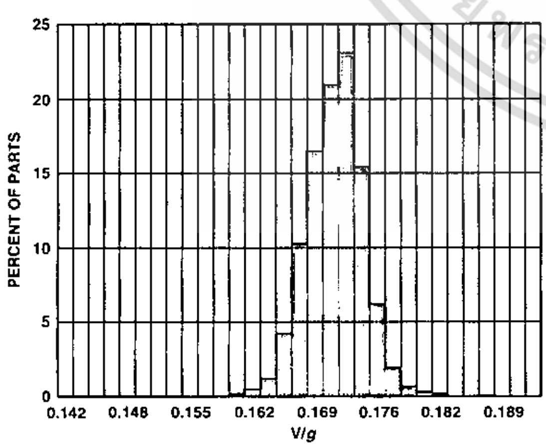
$V_{DD} = 5\ V$



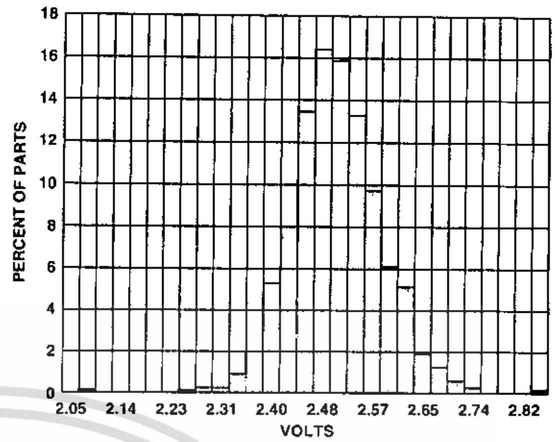
TPC 1. X-Axis Zero g Bias Distribution at X_{FILT}, $V_{DD} = 3\ V$



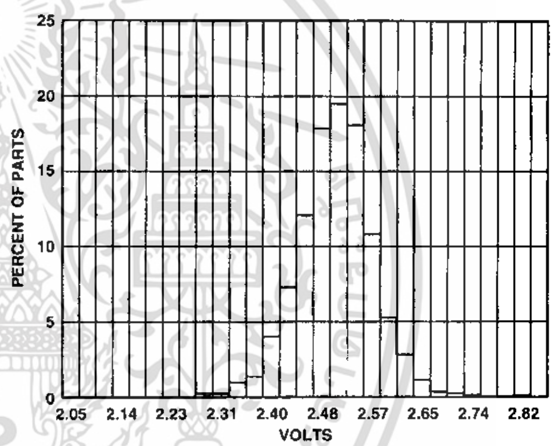
TPC 2. Y-Axis Zero g Bias Distribution at Y_{FILT}, $V_{DD} = 3\ V$



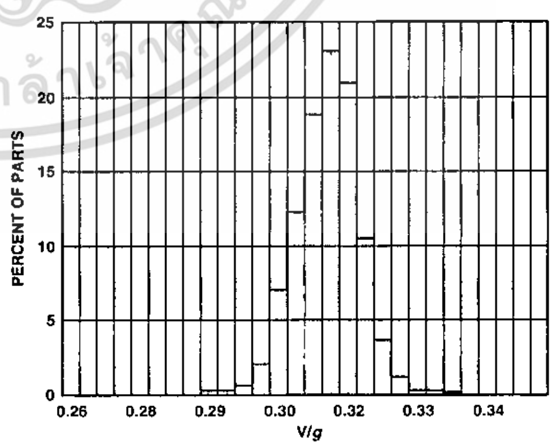
TPC 3. X-Axis Sensitivity Distribution at X_{FILT}, $V_{DD} = 3\ V$



TPC 4. X-Axis Zero g Bias Distribution at X_{FILT}, $V_{DD} = 5\ V$



TPC 5. Y-Axis Zero g Bias Distribution at Y_{FILT}, $V_{DD} = 5\ V$

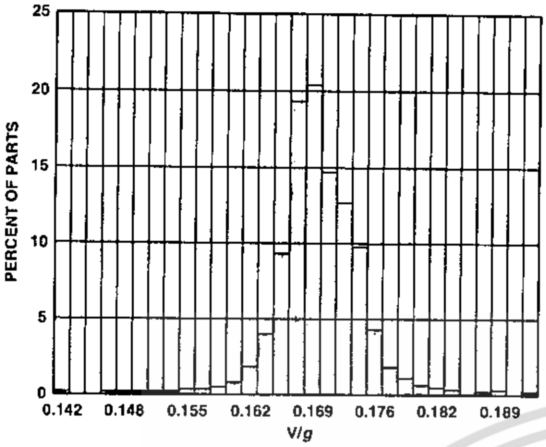


TPC 6. X-Axis Sensitivity Distribution at X_{FILT}, $V_{DD} = 5\ V$

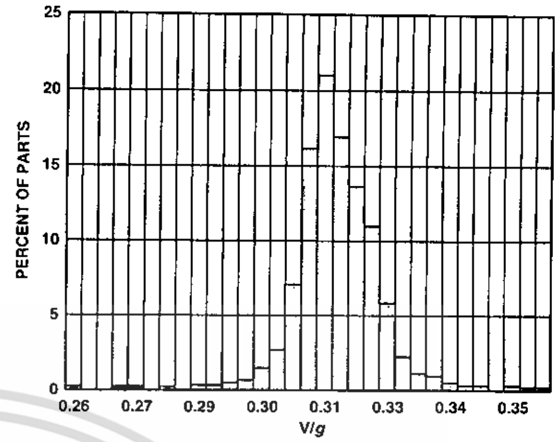
*Data taken from 4500 parts over 3 lots minimum.

$V_{DD} = 3$ V

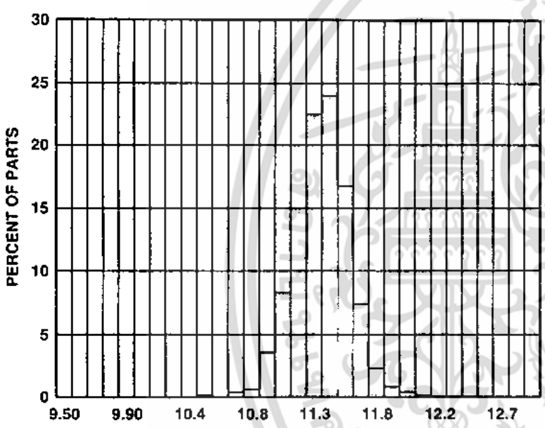
$V_{DD} = 5$ V



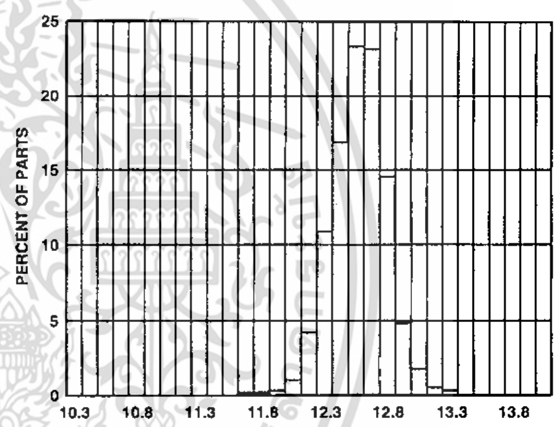
TPC 7. Y-Axis Sensitivity Distribution at Y_{FILT}, $V_{DD} = 3$ V



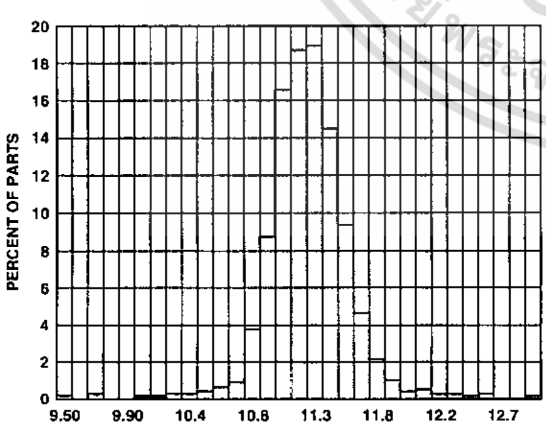
TPC 10. Y-Axis Sensitivity Distribution at Y_{FILT}, $V_{DD} = 5$ V



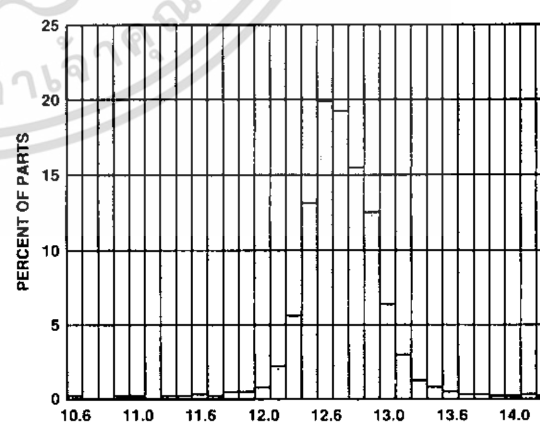
TPC 8. X-Axis Sensitivity at X_{OUT}, $V_{DD} = 3$ V



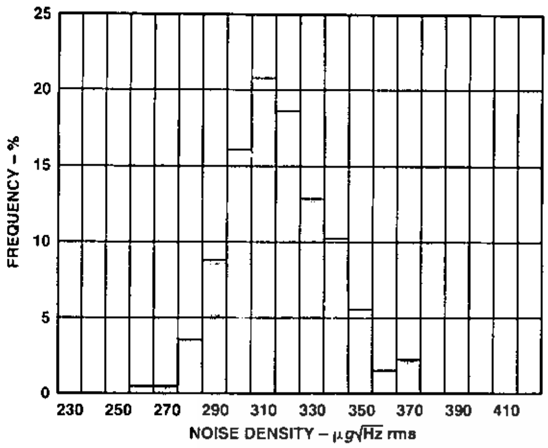
TPC 11. X-Axis Sensitivity at X_{OUT}, $V_{DD} = 5$ V



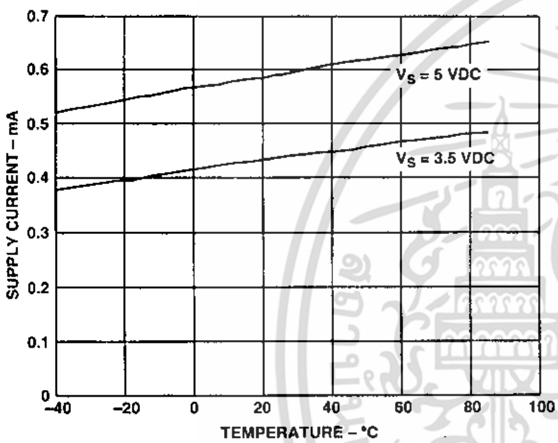
TPC 9. Y-Axis Sensitivity at Y_{OUT}, $V_{DD} = 3$ V



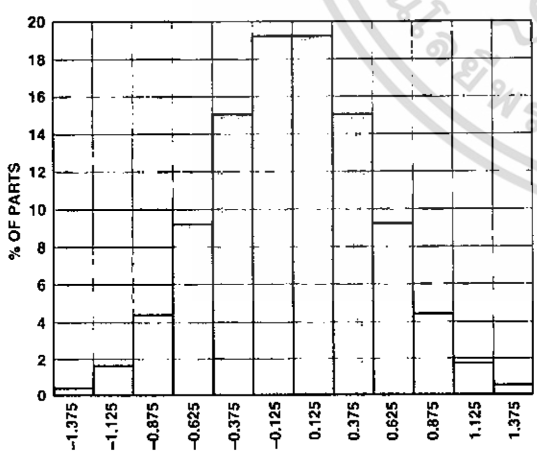
TPC 12. Y-Axis Sensitivity at Y_{OUT}, $V_{DD} = 5$ V



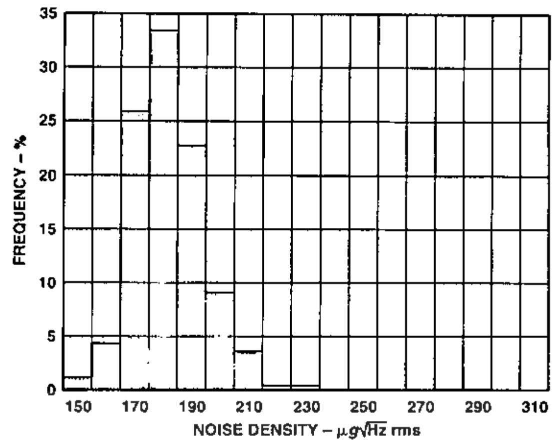
TPC 13. Noise Density Distribution, V_{DD} = 3 V



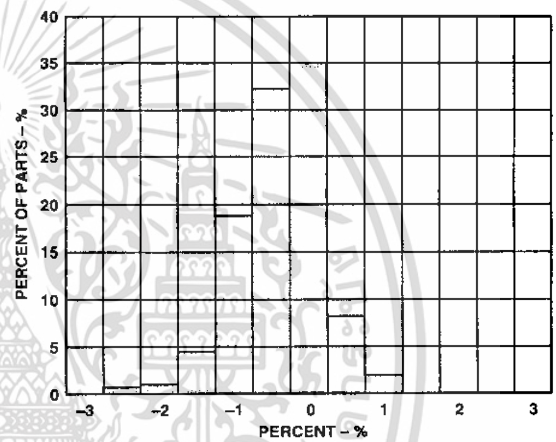
TPC 14. Typical Supply Current vs. Temperature



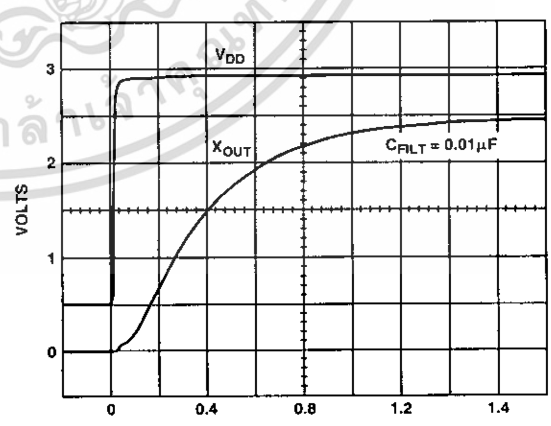
TPC 15. Rotational Die Alignment



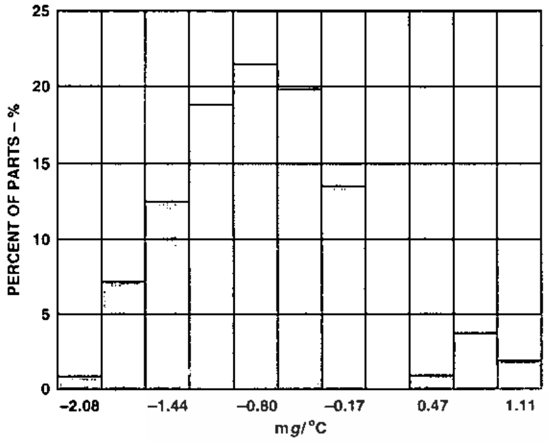
TPC 16. Noise Density Distribution, V_{DD} = 5 V



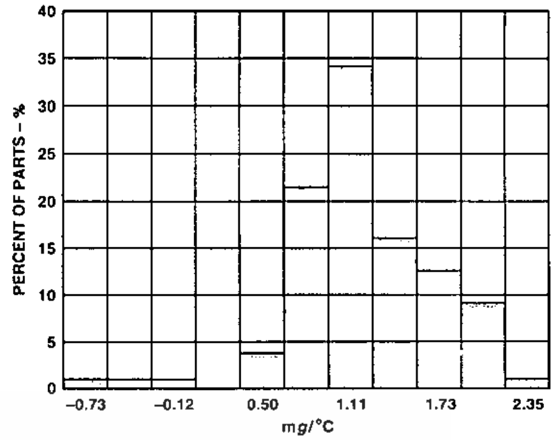
TPC 17. Cross-Axis Sensitivity Distribution



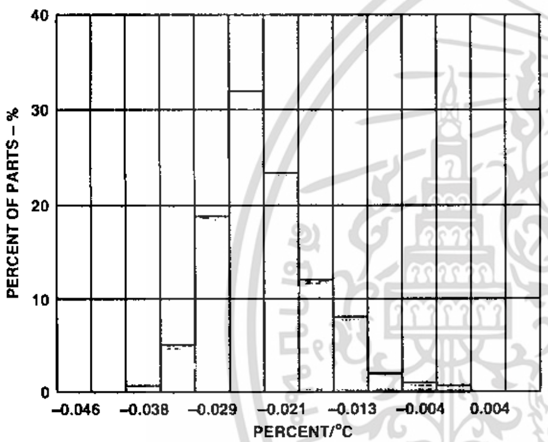
TPC 18. Typical Turn-On Time



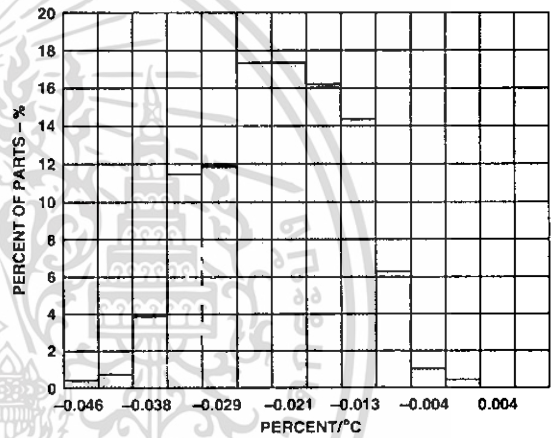
TPC 19. X-Axis Zero g Drift Due to Temperature Distribution, –40°C to +85°C



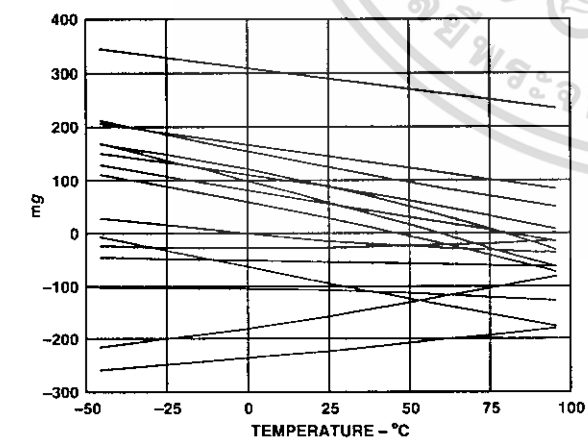
TPC 22. Y-Axis Zero g Drift Due to Temperature Distribution, –40°C to +85°C



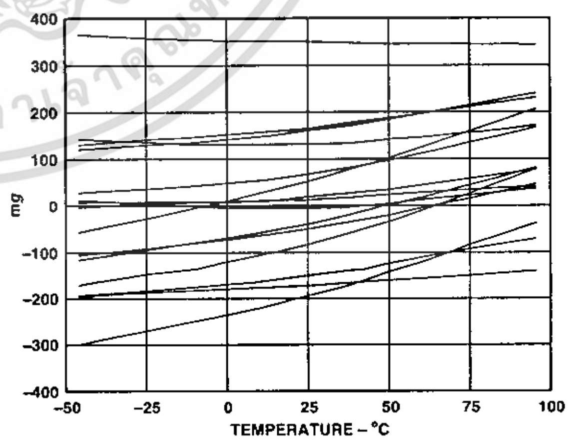
TPC 20. X-Axis Sensitivity Drift at X_{FILT} Due to Temperature Distribution, –40°C to +85°C



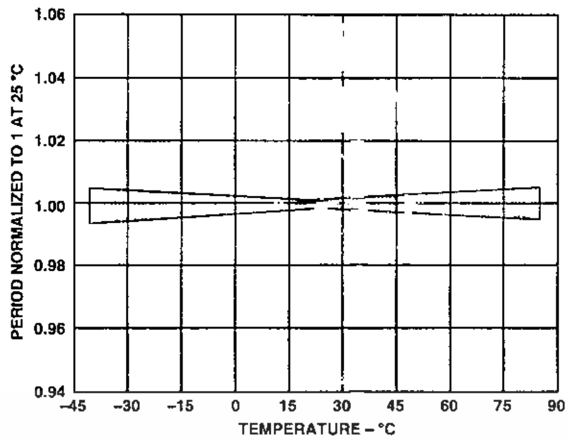
TPC 23. Y-Axis Sensitivity Drift at Y_{FILT} Due to Temperature Distribution, –40°C to +85°C



TPC 21. Typical X-Axis Zero g vs. Output for 16 Parts



TPC 24. Typical Y-Axis Zero g vs. Output for 16 Parts



TPC 25. Normalized DCM Period (T2) vs. Temperature

FINITIONS

Length of the "on" portion of the cycle.

Length of the total cycle.

ty Cycle — Ratio of the "on" time (T1) of the cycle to the total cycle (T2). Defined as T1/T2 for the ADXL202E/ADXL210.

sewidth — Time period of the "on" pulse. Defined as T1 for the ADXL202E/ADXL210.

EORY OF OPERATION

: ADXL202E is a complete, dual-axis acceleration measurement :em on a single monolithic IC. It contains a polysilicon surface- :romachined sensor and signal conditioning circuitry to imple- nt an open loop acceleration measurement architecture. For h axis, an output circuit converts the analog signal to a duty le modulated (DCM) digital signal that can be decoded with ounter/timer port on a microprocessor. The ADXL202E is able of measuring both positive and negative accelerations to east ±2 *g*. The accelerometer can measure static acceleration :es such as gravity, allowing it to be used as a tilt sensor.

e sensor is a surface micromachined polysilicon structure lt on top of the silicon wafer. Polysilicon springs suspend structure over the surface of the wafer and provide a resistance inst acceleration forces. Deflection of the structure is measured ıg a differential capacitor that consists of independent fixed tes and central plates attached to the moving mass. The fixed :es are driven by 180° out of phase square waves. An accelera- ı will deflect the beam and unbalance the differential capacitor, ılting in an output square wave whose amplitude is propor- ıal to acceleration. Phase sensitive demodulation techniques are n used to rectify the signal and determine the direction of acceleration.

e output of the demodulator drives a duty cycle modulator CM) stage through a 32 kΩ resistor. At this point a pin is ilable on each channel to allow the user to set the signal band- th of the device by adding a capacitor. This filtering improves asurement resolution and helps prevent aliasing.

er being low-pass filtered, the analog signal is converted to a y cycle modulated signal by the DCM stage. A single resistor the period for a complete cycle (T2), which can be set between ms and 10 ms (see Figure 12). A 0 *g* acceleration produces a nominally 50% duty cycle. The acceleration signal can be determined by measuring the length of the T1 and T2 pulses with a counter/timer or with a polling loop using a low cost microcontroller.

An analog output voltage can be obtained either by buffering the signal from the X_{FILT} and Y_{FILT} pin, or by passing the duty cycle signal through an RC filter to reconstruct the dc value.

The ADXL202E will operate with supply voltages as low as 3.0 V or as high as 5.25 V.

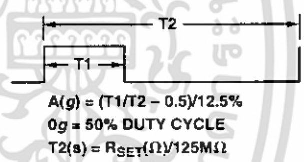


Figure 1. Typical Output Duty Cycle

APPLICATIONS

POWER SUPPLY DECOUPLING

For most applications a single 0.1 µF capacitor, C_{DC}, will adequately decouple the accelerometer from signal and noise on the power supply. However, in some cases, especially where digital devices such as microcontrollers share the same power supply, digital noise on the supply may cause interference on the ADXL202E output. This may be observed as a slowly undulating fluctuation of voltage at X_{FILT} and Y_{FILT}. If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite beads, may be inserted in the supply line of the ADXL202E.

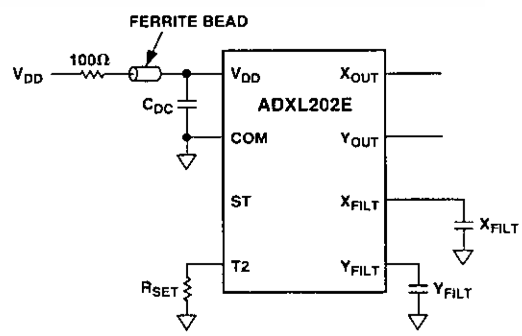


Figure 2.

)ESIGN PROCEDURE FOR THE ADXL202E

'he design procedure for using the ADXL202E with a duty cycle utput involves selecting a duty cycle period and a filter capacitor. , proper design will take into account the application requirements)r bandwidth, signal resolution and acquisition time, as discussed ı the following sections.

)ecoupling Capacitor C_{DC}

, 0.1 µF capacitor is recommended from V_{DD} to COM for power ɹpply decoupling.

T

'he ST pin controls the self-test feature. When this pin is set to 'DD, an electrostatic force is exerted on the beam of the accelerom-ter. The resulting movement of the beam allows the user to test if ıe accelerometer is functional. The typical change in output will e 10% at the duty cycle outputs (corresponding to 800 m*g*). 'his pin may be left open circuit or connected to common in ormal use.

)uty Cycle Decoding

'he ADXL202E's digital output is a duty cycle modulator. ıcceleration is proportional to the ratio T1/T2. The nominal utput of the ADXL202E is:

0 *g* = 50% *Duty Cycle*

Scale factor is 12.5% Duty Cycle Change per *g*

'hese nominal values are affected by the initial tolerance of the ,evice including zero *g* offset error and sensitivity error.

'2 does not have to be measured for every measurement cycle. t need only be updated to account for changes due to tempera-ure, (a relatively slow process). Since the T2 time period is shared ıy both X and Y channels, it is necessary only to measure it on ne channel of the ADXL202E. Decoding algorithms for various ıicrocontrollers have been developed. Consult the appropriate ıpplication Note.

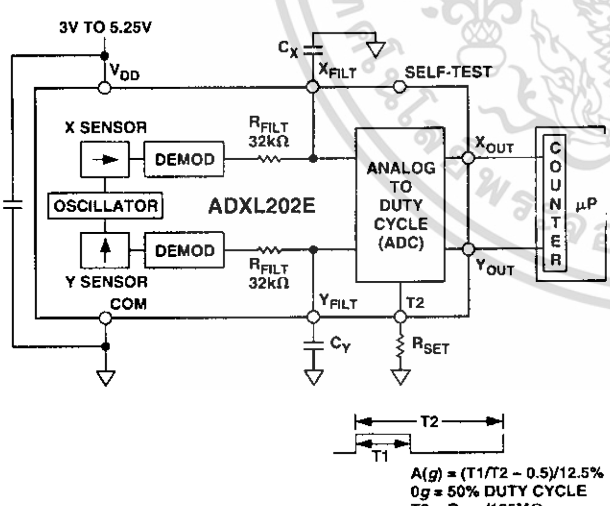


Figure 3. Block Diagram

Setting the Bandwidth Using C_X and C_Y

The ADXL202E has provisions for bandlimiting the X_{FILT} and Y_{FILT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is:

$$F_{-3\,dB} = \frac{1}{\left(2\,\pi\,(32\;k\Omega) \times C(x,y)\right)}$$

or, more simply, $F_{-3\,dB} = \dfrac{5\,\mu F}{C_{(X,Y)}}$

The tolerance of the internal resistor (R_{FILT}), can vary typically as much as ±15% of its nominal value of 32 kΩ; so the bandwidth will vary accordingly. A minimum capacitance of 1000 pF for $C_{(X,\,Y)}$ is required in all cases.

Table I. Filter Capacitor Selection, C_X and C_Y

Bandwidth	Capacitor Value
10 Hz	0.47 µF
50 Hz	0.10 µF
100 Hz	0.05 µF
200 Hz	0.027 µF
500 Hz	0.01 µF
5 kHz	0.001 µF

Setting the DCM Period with R_{SET}

The period of the DCM output is set for both channels by a single resistor from R_{SET} to ground. The equation for the period is:

$$T2 = \frac{R_{SET}\;(\Omega)}{125\;M\Omega}$$

A 125 kΩ resistor will set the duty cycle repetition rate to approximately 1 kHz, or 1 ms. The device is designed to operate at duty cycle periods between 0.5 ms and 10 ms.

Table II. Resistor Values to Set T2

T2	R_{SET}
1 ms	125 kΩ
2 ms	250 kΩ
5 ms	625 kΩ
10 ms	1.25 MΩ

Note that the R_{SET} should always be included, even if only an analog output is desired. Use an R_{SET} value between 500 kΩ and 2 MΩ when taking the output from X_{FILT} or Y_{FILT}. The R_{SET} resistor should be place close to the T2 Pin to minimize parasitic capacitance at this node.

Selecting the Right Accelerometer

For most tilt sensing applications the ADXL202E is the most appropriate accelerometer. Its higher sensitivity (12.5%/*g*) allows the user to use a lower speed counter for PWM decoding while maintaining high resolution. The ADXL210 should be used in applications where accelerations of greater than ±2 *g* are expected.

OCOMPUTER INTERFACES

DXL202E is specifically designed to work with low-cost controllers. Specific code sets, reference designs, and applica-otes are available from the factory. This section will outline a al design procedure and discuss the various trade-offs that o be considered.

esigner should have some idea of the required performance system in terms of:

tion: the smallest signal change that needs to be detected.

vidth: the highest frequency that needs to be detected.

ition Time: the time that will be available to acquire the signal h axis.

requirements will help to determine the accelerometer band-, the speed of the microcontroller clock and the length of 2 period.

selecting a microcontroller it is helpful to have a counter port available. The microcontroller should have provisions tware calibration. While the ADXL202E is a highly accurate rometer, it has a wide tolerance for initial offset. The easi-y to null this offset is with a calibration factor saved on the controller or by a user calibration for zero *g*. In the case the offset is calibrated during manufacture, there are several is, including external EEPROM and microcontrollers with time programmable" features.

GN TRADE-OFFS FOR SELECTING FILTER ACTERISTICS: THE NOISE/BW TRADE-OFF

ccelerometer bandwidth selected will determine the measure-esolution (smallest detectable acceleration). Filtering can be o lower the noise floor and improve the resolution of the rometer. Resolution is dependent on both the analog filter idth at X_{FILT} and Y_{FILT} and on the speed of the micro-ller counter.

nalog output of the ADXL202E has a typical bandwidth Hz, while the duty cycle modulators' bandwidth is 500 Hz. ser must filter the signal at this point to limit aliasing . To minimize DCM errors the analog bandwidth should be an 1/10 the DCM frequency. Analog bandwidth may be sed to up to 1/2 the DCM frequency in many applications. ill result in greater dynamic error generated at the DCM.

nalog bandwidth may be further decreased to reduce noise nprove resolution. The ADXL202E noise has the character-of white Gaussian noise that contributes equally at all ncies and is described in terms of μg per root Hz; i.e., the is proportional to the square root of the bandwidth of the rometer. It is recommended that the user limit bandwidth to west frequency needed by the application, to maximize the ion and dynamic range of the accelerometer.

With the single pole roll-off characteristic, the typical noise of the ADXL202E is determined by the following equation:

$$Noise\,(rms) = \left(200\ \mu g/\sqrt{Hz}\right) \times \left(\sqrt{BW \times 1.6}\right)$$

At 100 Hz the noise will be:

$$Noise\,(rms) = \left(200\ \mu g/\sqrt{Hz}\right) \times \left(\sqrt{100 \times (1.6)}\right) = 2.53\ mg$$

Often the peak value of the noise is desired. Peak-to-peak noise can only be estimated by statistical methods. Table III is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table III. Estimation of Peak-to-Peak Noise

Nominal Peak-to-Peak Value	% of Time that Noise Will Exceed Nominal Peak-to-Peak Value
2.0 × rms	32%
4.0 × rms	4.6%
6.0 × rms	0.27%
8.0 × rms	0.006%

The peak-to-peak noise value will give the best estimate of the uncertainty in a single measurement.

Table IV gives typical noise output of the ADXL202E for various C_X and C_Y values.

Table IV. Filter Capacitor Selection, C_X and C_Y

Bandwidth	C_X, C_Y	rms Noise	Peak-to-Peak Noise Estimate 95% Probability (rms × 4)
10 Hz	0.47 μF	0.8 m*g*	3.2 m*g*
50 Hz	0.10 μF	1.8 m*g*	7.2 m*g*
100 Hz	0.05 μF	2.5 m*g*	10.1 m*g*
200 Hz	0.027 μF	3.6 m*g*	14.3 m*g*
500 Hz	0.01 μF	5.7 m*g*	22.6 m*g*

CHOOSING T2 AND COUNTER FREQUENCY: DESIGN TRADE-OFFS

The noise level is one determinant of accelerometer resolution. The second relates to the measurement resolution of the counter when decoding the duty cycle output.

The ADXL202E's duty cycle converter has a resolution of approximately 14 bits; better resolution than the accelerometer itself. The actual resolution of the acceleration signal is, however, limited by the time resolution of the counting devices used to decode the duty cycle. The faster the counter clock, the higher the resolution of the duty cycle and the shorter the T2 period can be for a given resolution. The following table shows some of the trade-offs. It is important to note that this is the resolution due to the microprocessors' counter. It is probable that the accelerometer's noise floor may set the lower limit on the resolution, as discussed in the previous section.

able V. Trade-Offs Between Microcontroller Counter Rate, '2 Period, and Resolution of Duty Cycle Modulator

2 (ms)	R_{SET} (kΩ)	ADXL202E Sample Rate	Counter-Clock Rate (MHz)	Counts per T2 Cycle	Counts per *g*	Resolution (m*g*)
0	124	1000	2.0	2000	250	4.0
0	124	1000	1.0	1000	125	8.0
0	124	1000	0.5	500	62.5	16.0
0	625	200	2.0	10000	1250	0.8
0	625	200	1.0	5000	625	1.6
0	625	200	0.5	2500	312.5	3.2
).0	1250	100	2.0	20000	2500	0.4
).0	1250	100	1.0	10000	1250	0.8
).0	1250	100	0.5	5000	625	1.6

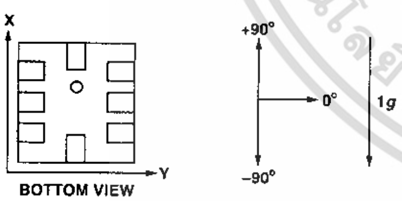
TRATEGIES FOR USING THE DUTY CYCLE OUTPUT 'ITH MICROCONTROLLERS

pplication notes outlining various strategies for using the duty 'cle output with low cost microcontrollers are available from ie factory.

SING THE ADXL202E AS A DUAL-AXIS TILT SENSOR

ne of the most popular applications of the ADXL202E is tilt easurement. An accelerometer uses the force of gravity as an put vector to determine orientation of an object in space.

n accelerometer is most sensitive to tilt when its sensitive axis perpendicular to the force of gravity, i.e., parallel to the earth's rface. At this orientation its sensitivity to changes in tilt is hight. When the accelerometer is oriented on axis to gravity, i.e., :ar its +1 *g* or –1 *g* reading, the change in output acceleration :r degree of tilt is negligible. When the accelerometer is perpencular to gravity, its output will change nearly 17.5 m*g* per degree ' tilt, but at 45° degrees it is changing only at 12.2 m*g* per :gree and resolution declines. The following table illustrates e changes in the X and Y axes as the device is tilted ±90° rough gravity.



Axis rientation Horizon (°)	X Output (*g*)	Δ per Degree of Tilt (m*g*)	Y Output (*g*)	Δ per Degree of Tilt (m*g*)
–90	–1.000	–0.2	0.000	17.5
–75	–0.966	4.4	0.259	16.9
–60	–0.866	8.6	0.500	15.2
–45	–0.707	12.2	0.707	12.4
–30	–0.500	15.0	0.866	8.9
–15	–0.259	16.8	0.966	4.7
0	0.000	17.5	1.000	0.2
15	0.259	16.9	0.966	–4.4
30	0.500	15.2	0.866	–8.6
45	0.707	12.4	0.707	–12.2
60	0.866	8.9	0.500	–15.0
75	0.966	4.7	0.259	–16.8
90	1.000	0.2	0.000	–17.5

Figure 4. How the X and Y Axes Respond to Changes in Tilt

A DUAL AXIS TILT SENSOR: CONVERTING ACCELERATION TO TILT

When the accelerometer is oriented so both its X and Y axes are parallel to the earth's surface it can be used as a two axis tilt sensor with a roll and a pitch axis. Once the output signal from the accelerometer has been converted to an acceleration that varies between –1 *g* and +1 *g*, the output tilt in degrees is calculated as follows:

$$Pitch = ASIN\ (Ax/1\ g)$$
$$Roll = ASIN\ (Ay/1\ g)$$

Be sure to account for overranges. It is possible for the accelerometers to output a signal greater than ±1 *g* due to vibration, shock or other accelerations.

MEASURING 360° OF TILT

It is possible to measure a full 360° of orientation through gravity by using two accelerometers oriented perpendicular to one another (see Figure 5). When one sensor is reading a maximum change in output per degree, the other is at its minimum.

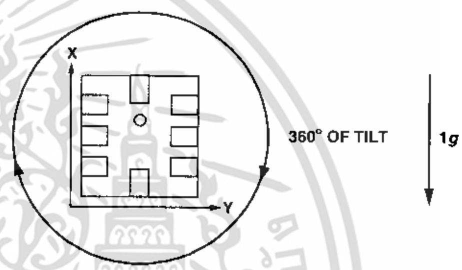


Figure 5. Using a Two-Axis Accelerometer to Measure 360° of Tilt

USING THE ANALOG OUTPUT

The ADXL202E was specifically designed for use with its digital outputs, but has provisions to provide analog outputs as well.

Duty Cycle Filtering

An analog output can be reconstructed by filtering the duty cycle output. This technique requires only passive components. The duty cycle period (T2) should be set to <1 ms. An RC filter with a 3 dB point at least a factor of >10 less than the duty cycle frequency is connected to the duty cycle output. The filter resistor should be no less than 100 kΩ to prevent loading of the output stage. The analog output signal will be ratiometric to the supply voltage. The advantage of this method is an output scale factor of approximately double the analog output. Its disadvantage is that the frequency response will be lower than when using the X_{FILT}, Y_{FILT} output.

X_{FILT}, Y_{FILT} Output

The second method is to use the analog output present at the X_{FILT} and Y_{FILT} pin. Unfortunately, these pins have a 32 kΩ output impedance and are not designed to drive a load directly. An op amp follower may be required to buffer this pin. The advantage of this method is that the full 5 kHz bandwidth of the accelerometer is available to the user. A capacitor still must be added at this point for filtering. The duty cycle converter should be kept running by using R_{SET} <10 MΩ. Note that the accelerometer offset and sensitivity are ratiometric to the supply voltage. The offset and sensitivity are nominally:

0 *g* Offset = $V_{DD}/2$

ADXL202E Sensitivity = (60 mV × V_S)/*g*

3 THE ADXL202E IN VERY LOW POWER ICATIONS

plication note outlining low power strategies for the .202E is available. Some key points are presented here. ›ssible to reduce the ADXL202E's average current from \ to less than 20 µA by using the following techniques:

ver Cycle the accelerometer.

ı the accelerometer at a Lower Voltage, (Down to 3 V).

Cycling with an External A/D

ding on the value of the X_{FILT} capacitor, the ADXL202E ble of turning on and giving a good reading in 1.6 ms. Most ontroller based A/Ds can acquire a reading in another 25 µs. t is possible to turn on the ADXL202E and take a reading ms. If we assume that a 20 Hz sample rate is sufficient, al current required to take 20 samples is 2 ms × 20 samples/s nA = 24 µA average current. Running the part at 3 V will the supply current from 0.6 mA to 0.4 mA, bringing the e current down to 16 µA.

/D should read the analog output of the ADXL202E at $_{LT}$ and Y_{FILT} pins. A buffer amplifier is recommended, and : required in any case to amplify the analog output to give ı resolution with an 8-bit to 10-bit converter.

Cycling When Using the Digital Output

ernative is to run the microcontroller at a higher clock rate t it into shutdown between readings, allowing the use of the output. In this approach the ADXL202E should be set at est sample rate (T2 = 0.5 ms), with a 500 Hz filter at X_{FILT} $_{ILT}$. The concept is to acquire a reading as quickly as posıd then shut down the ADXL202E and the microcontroller ıe next sample is needed.

er of the above approaches, the ADXL202E can be turned off directly using a digital port pin on the microcontroller to the accelerometer without additional components.

CALIBRATING THE ADXL202E/ADXL210

The initial value of the offset and scale factor for the ADXL202E will require calibration for applications such as tilt measurement. The ADXL202E architecture has been designed so that these calibrations take place in the software of the microcontroller used to decode the duty cycle signal. Calibration factors can be stored in EEPROM or determined at turn-on and saved in dynamic memory.

For low *g* applications, the force of gravity is the most stable, accurate and convenient acceleration reference available. A reading of the 0 *g* point can be determined by orientating the device parallel to the earth's surface and then reading the output.

A more accurate calibration method is to make measurements at +1 *g* and –1 *g*. The sensitivity can be determined by the two measurements.

To calibrate, the accelerometer's measurement axis is pointed directly at the earth. The 1 *g* reading is saved and the sensor is turned 180° to measure –1 *g*. Using the two readings, the sensitivity is:

Let A = Accelerometer output with axis oriented to +1 g
Let B = Accelerometer output with axis oriented to –1 g then:

$$Sensitivity = [A - B]/2\,g$$

For example, if the +1 *g* reading (A) is 55% duty cycle and the –1 *g* reading (B) is 32% duty cycle, then:

$$Sensitivity = [55\% - 32\%]/2\,g = 11.5\%/g$$

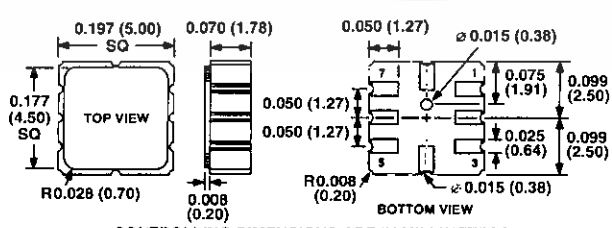
These equations apply whether the output is analog or duty cycle.

Application notes outlining algorithms for calculating acceleration from duty cycle and automated calibration routines are available from the factory.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

8-Terminal Ceramic Leadless Chip Carrier (E-8)



CONTROLLING DIMENSIONS ARE IN MILLIMETERS

C02064–2.5–10/00 (rev. A)
PRINTED IN U.S.A.

CYPRESS MICROSYSTEMS

Application Note

AN2152

Graphics LCD and PSoC Interface

By: Svyatoslav Paliy
Associated Project: Yes
Associated Part Family: CY8C27443
PSoC Designer Version: 4.0

Summary
This Application Note describes how to control a PCD8544-based graphics LCD in a PSoC project.

Introduction
In most applications there is a need to display information to the user. A graphics LCD is a powerful, easy-to-control solution. It can provide both text and graphical illustration to an application. This Application Note shows how to control a graphics LCD using a PSoC device. The project has a software library for write text and a graphics drawing on LCD and PC software to build a font generator and bitmap-to-C-array converter.

Graphics LCD
In this application, I used a 48x84 graphics LCD with Philips PCD8544 controller/driver. You may find the data sheet for this controller under Reference [1]. Devices similar to the PCD8544 are listed at Reference [2].

Many manufacturers make displays based on the PCD8544 or compatible controllers. This very low-cost LCD controller has memory bits, each of which represent one pixel on the LCD. This memory allows only writes. It is not possible to read from this memory, which can create some difficulties with building routines for the smaller memory versions of PSoC.

Data are downloaded in bytes into the 48x84-bit RAM data display matrix of PCD8544, as indicated in Figure 1. The columns are indicated by the address pointer. The address ranges are: X 0 to 83 (1010011), Y 0 to 5 (101). Addresses outside these ranges are not allowed. The X addresses increment after each byte. After the last X address (X = 83), X wraps around to 0 and Y increments to the address in the next row. After the very last address (X = 83 and Y = 5), the address pointers wrap around to address (X =0 and Y =0).

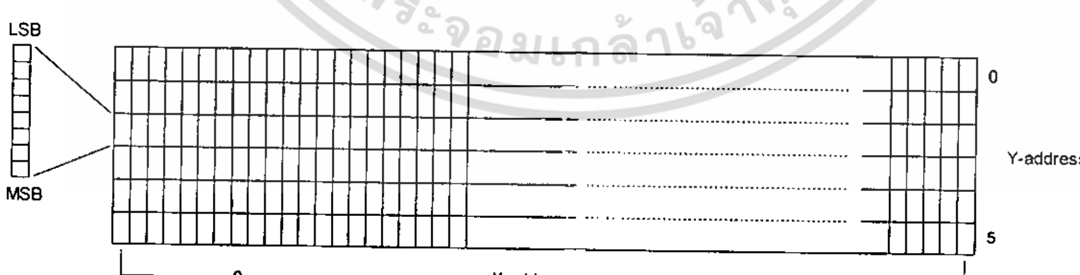


Figure 1. LCD RAM Format, Addressing

Circuit Schematic

The LCD connects to the PSoC by four wires. Two wires are for one-direction SPI, one wire is for data/control switching, and one wire is for the reset signal. The PCD8544 needs one external capacitor for an internal bias voltage generator.

Figure 2 shows the schematic when PSoC is powered by 3.3V. Figure 3 shows the schematic when PSoC is powered by 5V. Some applications only allow use of 3.3V supply. Some require 5V and therefore need an additional level translator to be added.

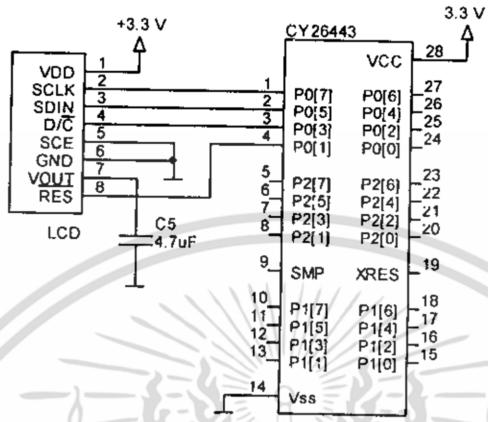


Figure 2. Circuit Schematic for 3.3V Powered PSoC

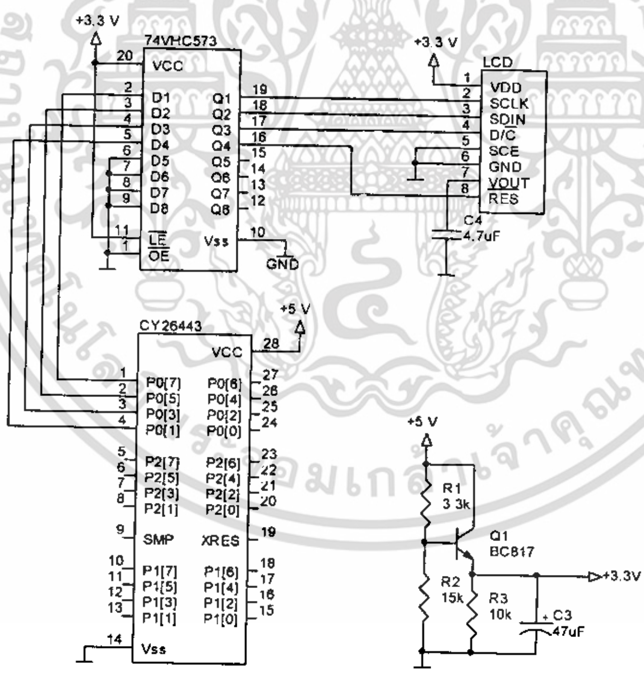


Figure 3. Circuit Schematic for 5V Powered PSoC

Software Library

For the simplified drawing on this LCD, I wrote a software library that can work in two modes: drawing over background (when C-compiler directive, DRAW_OVER_BACKGROUND, is defined) and drawing without background (in other cases).

Because the LCD internal memory is write only and some PSoC devices have too little memory to build a cache, all drawing routines output immediately on the LCD.

The software library has two low-level functions that are hardware dependent (see Table1). If you want to port this library onto a similar LCD controller with another physical connection (for example, use of BF9864AFPH with I^2C interface), you must rewrite only these two functions.

Table 1. LCD Controller Low-Level, Hardware-Dependent Functions

`LcdSendData(char data)`	Send byte of data to LCD. For more information, see the LCD driver data sheet.
`LcdSendCommand(char data)`	Send command byte to LCD. For more information, see the LCD driver data sheet.

High-level functions that may be used with DRAW_OVER_BACKGROUND are listed in Table 2. They differ from the functions that may be used without DRAW_OVER_BACKGROUND (see Table 3) by **dt** parameter, which can take the following values:

- DRAW_OR – Text or graphics draw over background with using Logical OR operator under drawn and background pixels.
- DRAW_XOR – Similar to DRAW_OR but uses XOR instead OR operator.
- DRAW_CLEAR – Does not draw pixels, only restores background. Erases drawn pixels.

Table 2. High-Level Functions used when DRAW_OVER_BACKGROUND is Defined

`LcdInit(const char * dataPtr)`	Performs LCD initialization, draws background. Parameters: dataPtr – pointer to array in Flash memory that contains background.
`LcdSetBackground ( const char * dataPtr)`	Allows pointer to change to current background. Does not perform repaint. Only the pointer changes. Parameters: dataPtr – pointer to array that contains background.
`LcdClear()`	Clears display and only shows background.
`LcdContrast(char contrast)`	Allows contrast change. No visible result at ambient temperature. High temperature allows decrease contrast. Low temperature allows increase contrast. Parameters: contrast – byte describes contrast (higher value means higher contrast).
`LcdGoTo(char x, char y)`	Changes current text position. Parameters: x – X- coordinant of text position. y – Y- coordinant of text position. Y- coordinant means not quite a pixel, but an 8-pixel bank (e.g., display has 6 bank by height).
`LcdImage ( char x, char y, char xsize, char ysize, const char * dataPtr)`	Draws image. Parameters: x,y – coordinants of image top-left corner. xsize, ysize – image width and height. dataPrt – pointer to the array that contains image.
`LcdChr ( char ch, draw_type dt)`	Draws single character (by the small font) starting from current text position (see `LcdGoto` function above). Parameters: ch – character. dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdStr (
     char *dataPtr,
     draw_type dt)
```

Writes string (by the small font) starting from current text position from data memory.
Parameters: dataPtr – pointer to the string in the data memory.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdCStr (
     const char *dataPtr,
     draw_type dt)
```

Writes string (by the small font) starting from current text position from program memory.
Parameters: dataPtr – pointer to string in the program memory.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdBigChr (
     char x,
     char y,
     char ch,
     draw_type dt)
```

Draws single character by the big font.
Parameters: x,y – coordinants of character.
ch – character.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdBigStr (
     char x,
     char y,
     char *dataPtr,
     draw_type dt)
```

Draws string from data memory by the big font.
Parameters: x,y – coordinants of string begin.
dataPtr – pointer to the string in data memory.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdBigCStr (
     char x,
     char y,
     const char *dataPtr,
     draw_type dt)
```

Draws string from program memory by the big font.
Parameters: x,y – coordinants of string begin.
dataPtr – pointer to the string in program memory.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdVBargraph (
     char x,
     char ystart,
     char yend,
     char yposition,
     draw_type dt)
```

Draws vertical bar graph.
Parameters: x – coordinate of left bar graph.
ystart – coordinant of top bar graph (8-pixel bank).
yend – coordinant of bottom bar graph (8-pixel bank).
yposition – current bar graph position, by pixels. (yposition <=(yend-begin)*8).
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
LcdHBargraph (
     char y,
     char xstart,
     char xend,
     char xposition,
     draw_type dt)
```

Draws horizontal bar graph.
Parameters: y – coordinant of the top bar graph (8-pixel bank).
xstart – coordinant of the left bar graph.
xend – coordinant of the right bar graph.
xposition – current bar graph position, by pixels. (xposition <=xyend-xbegin).
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

```
void LcdLine (
     char xb,
     char yb,
     char xe,
     char ye,
     draw_type dt);
```

Draws line.
Parameters: xb,yb – coordinants of where the line begins.
xe,ye – coordinants of where the line ends.
dt – (DRAW_OR, DRAW_XOR or DRAW_CLEAR).

Table 3. High-Level Functions used when DRAW_OVER_BACKGROUND is Undefined

`LcdInit()`	Performs LCD initialization.
`LcdClear()`	Clears display and shows blank.
`LcdContrast(char contrast)`	Allows contrast change. No visible result is observed at ambient temperature. Parameters: contrast – byte describes contrast (higher value means higher contrast).
`LcdGoTo(char x, char y)`	Change current text position. Parameters: x – X-coordinant of text position. y – Y-coordinant of text position. Y- coordinant means not quite a pixel, but an 8-pixel bank (e.g., display has 6 bank by height).
`LcdImage ( char x, char y, char xsize, char ysize, const char * dataPtr)`	Draws image. Parameters: x,y – coordinants of image top-left corner. xsize, ysize – image width and height. dataPrt – pointer to the array that contains image.
`LcdChr (char ch)`	Draws single character (by the small font) starting from current text position (see `LcdGoto` function above). Parameters: ch – character.
`LcdStr (char *dataPtr)`	Writes string (by the small font) starting from current text position. Parameters: dataPtr – pointer to the string in the data memory.
`LcdCStr (const char *dataPtr)`	Writes string (by the small font) starting from current text position. Parameters: dataPtr – pointer to the string in the program memory.
`LcdBigChr ( char x, char y, char ch)`	Draws single character by the big font. Parameters: x,y – coordinants of character. ch – character.
`LcdBigStr ( char x, char y, char *dataPtr)`	Draws string from data memory by the big font. Parameters: x,y – coordinants where string begins. dataPtr – pointer to the string in data memory.
`LcdBigCStr ( char x, char y, const char *dataPtr)`	Draw string from program memory by the big font. Parameters: x,y – coordinants where string begins. dataPtr – pointer to the string in program memory.
`LcdVBargraph ( char x, char ystart, char yend, char yposition)`	Draws vertical bar graph. Parameters: x – coordinant of left bar graph. ystart – coordinant of top bar graph (8-pixel bank). yend – coordinant of bottom bar graph (8-pixel bank). yposition – coordinate of current bar graph position, by pixel. (yposition <=(yend-ybegin)*8).
`LcdHBargraph ( char y, char xstart, char xend, char xposition)`	Draws horizontal bar graph. Parameters: y – coordinant of the top bar graph (8-pixel bank). xstart – coordinant of the left bar graph. xend – coordinant of the right bar graph. xposition - current bar graph position, by pixels. (xposition <=xyend-xstart).
`void LcdLine ( char xb, char yb, char xe, char ye);`	Draws line. Parameters: xb,yb – coordinants of where the line begins. xe,ye – coordinants of where the line ends.

PC Utilities

The software library contains two fonts. Both big and small fonts have been written as separate header files (*big_font.h* and *small_font.h*). To simplify font building, I wrote a utility for the PC that facilitates the font building process (see Figure 4).

In the left side of the form, you may draw a character and give its hexadecimal representation in the text editor. You may also write hexadecimal code and get a character picture.

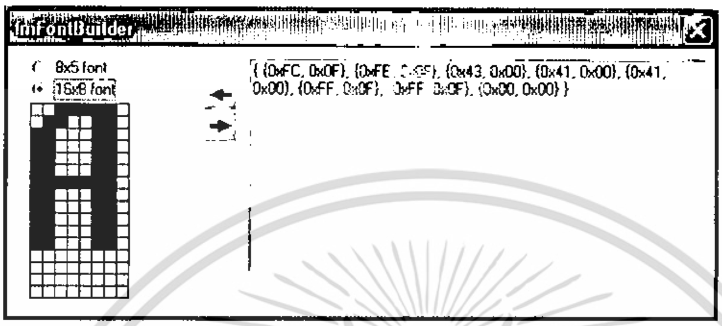


Figure 4. Font Building Utility

Another utility (Figure 5) converts bitmaps to C-language header files. Users must choose the path to the bitmap. Only black-and-white bitmaps with a height divisible by eight are supported (which is a consequence of using LCD controller page organization).

Also, users must choose a target file. If a target file exists, the utility rewrites it. The name for the hexadecimal array will be built from the file name but can be changed. By pressing the Convert button, the bitmap converts to a constant array of hexadecimal values. A file with conversion results is also generated.

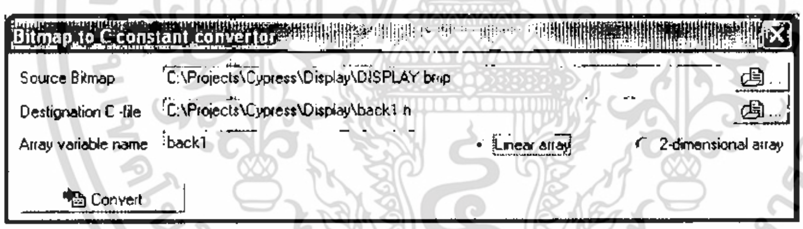


Figure 5. Utility for Bitmap-to-C-Array Conversion

Demonstration Applications

The demonstration application consists of a few screens that show display output possibilities. The first demonstration screen shows big and small text writing on the LCD. The second screen shows a text drawing using the DRAW_XOR parameter.

The third and fourth screens show horizontal and vertical drawings of bar graphs, respectively. The fifth screen shows a bitmap drawing. And the sixth screen is an example of an analog gauge showing line drawings with DRAW_OR and DRAW_CLEAR parameters.

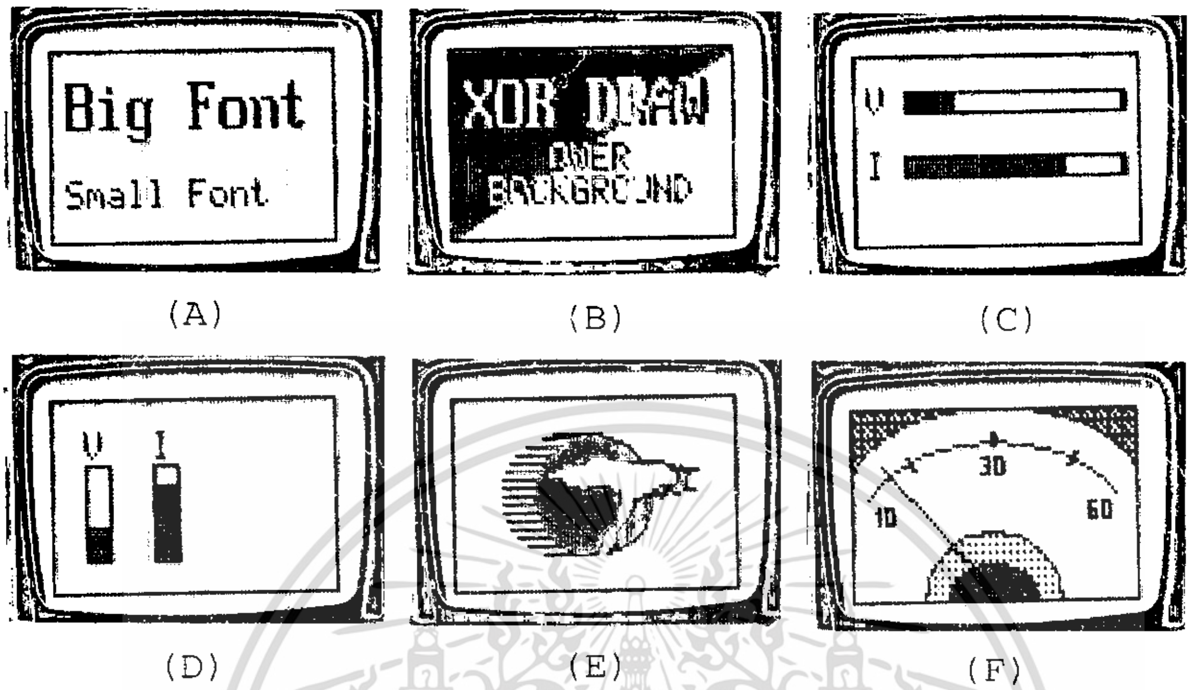


Figure 6. Screenshots from Demo Application

References

[1] Datasheet for PCD8544 can be downloaded from:
www.semiconductors.philips.com/acrobat/datasheets/PCD8544_1.pdf

[2] List similar to PCD8544 devices can be found at:
www.semiconductors.philips.com/similar/PCD8544U_2_F1.html

About the Author

Name: Svyatoslav Paliy

Title: Application Engineer

Background: Svyatoslav earned his Master of Science diploma in 2000 from National University "Lviv Polytechnic" (Ukraine). His interests include programming for embedded systems and Windows and Linux.

Contact: You may reach him at svt@isto.lviv.ua.

Cypress MicroSystems, Inc.
2700 162nd St. SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641

http://www.cypress.com/ / http://www.cypress.com/aboutus/sales_locations.cfm

DATA SHEET

PCD8544
48 × 84 pixels matrix LCD controller/driver

Product specification
File under Integrated Circuits, IC17

1999 Apr 12

Philips Semiconductors

PHILIPS

CONTENTS

1 FEATURES

- Single chip LCD controller/driver
- 48 row, 84 column outputs
- Display data RAM 48 × 84 bits
- On-chip:
  - Generation of LCD supply voltage (external supply also possible)
  - Generation of intermediate LCD bias voltages
  - Oscillator requires no external components (external clock also possible).
- External $\overline{RES}$ (reset) input pin
- Serial interface maximum 4.0 Mbits/s
- CMOS compatible inputs
- Mux rate: 48
- Logic supply voltage range V_{DD} to V_{SS}: 2.7 to 3.3 V
- Display supply voltage range V_{LCD} to V_{SS}
  - 6.0 to 8.5 V with LCD voltage internally generated (voltage generator enabled)
  - 6.0 to 9.0 V with LCD voltage externally supplied (voltage generator switched-off).
- Low power consumption, suitable for battery operated systems
- Temperature compensation of V_{LCD}
- Temperature range: −25 to +70 °C.

2 GENERAL DESCRIPTION

The PCD8544 is a low power CMOS LCD controller/driver, designed to drive a graphic display of 48 rows and 84 columns. All necessary functions for the display are provided in a single chip, including on-chip generation of LCD supply and bias voltages, resulting in a minimum of external components and low power consumption.

The PCD8544 interfaces to microcontrollers through a serial bus interface.

The PCD8544 is manufactured in n-well CMOS technology.

3 APPLICATIONS

- Telecommunications equipment.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCD8544U	–	chip with bumps in tray; 168 bonding pads + 4 dummy pads	–

5 BLOCK DIAGRAM

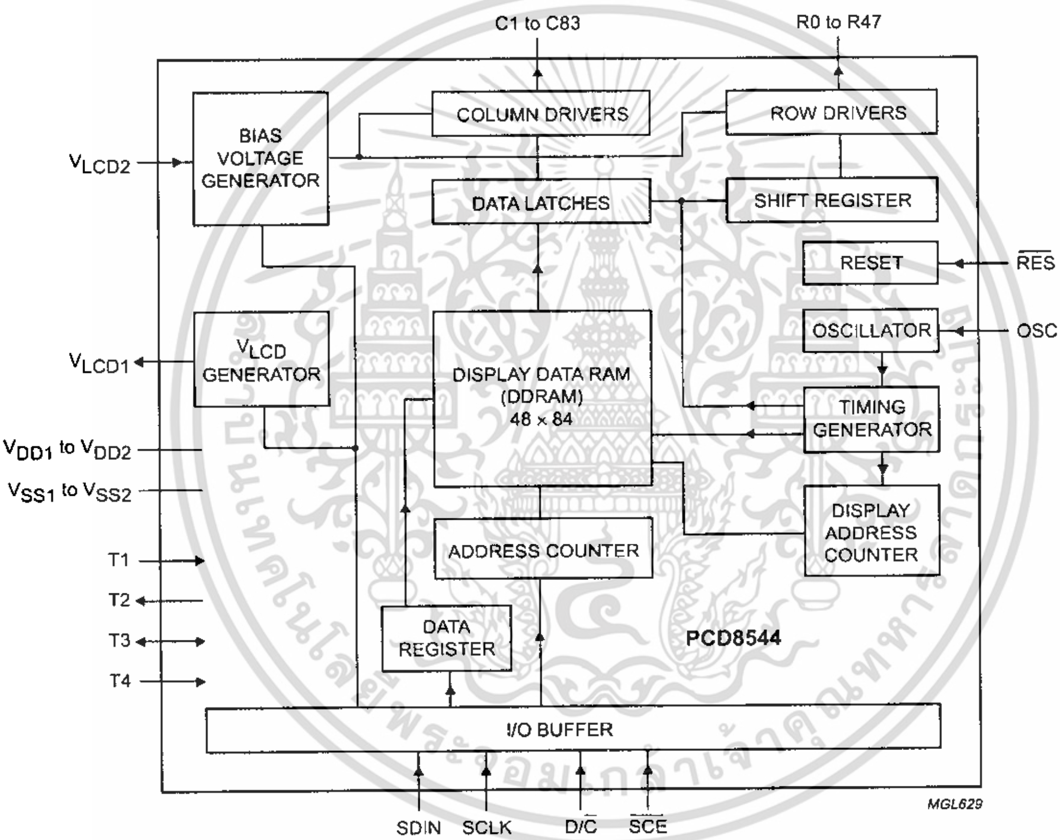


Fig.1 Block diagram.

6 PINNING

SYMBOL	DESCRIPTION
R0 to R47	LCD row driver outputs
C0 to C83	LCD column driver outputs
V_{SS1}, V_{SS2}	ground
V_{DD1}, V_{DD2}	supply voltage
V_{LCD1}, V_{LCD2}	LCD supply voltage
T1	test 1 input
T2	test 2 output
T3	test 3 input/output
T4	test 4 input
SDIN	serial data input
SCLK	serial clock input
D/$\overline{\text{C}}$	data/$\overline{\text{command}}$
$\overline{\text{SCE}}$	chip enable
OSC	oscillator
$\overline{\text{RES}}$	external reset input
dummy1, 2, 3, 4	not connected

Note

1. For further details, see Fig.18 and Table 7.

6.1 Pin functions

6.1.1 R0 TO R47 ROW DRIVER OUTPUTS

These pads output the row signals.

6.1.2 C0 TO C83 COLUMN DRIVER OUTPUTS

These pads output the column signals.

6.1.3 V_{SS1}, V_{SS2}: NEGATIVE POWER SUPPLY RAILS

Supply rails V_{SS1} and V_{SS2} must be connected together.

6.1.4 V_{DD1}, V_{DD2}: POSITIVE POWER SUPPLY RAILS

Supply rails V_{DD1} and V_{DD2} must be connected together.

6.1.5 V_{LCD1}, V_{LCD2}: LCD POWER SUPPLY

Positive power supply for the liquid crystal display. Supply rails V_{LCD1} and V_{LCD2} must be connected together.

6.1.6 T1, T2, T3 AND T4: TEST PADS

T1, T3 and T4 must be connected to V_{SS}, T2 is to be left open. Not accessible to user.

6.1.7 SDIN: SERIAL DATA LINE

Input for the data line.

6.1.8 SCLK: SERIAL CLOCK LINE

Input for the clock signal: 0.0 to 4.0 Mbits/s.

6.1.9 D/$\overline{\text{C}}$: MODE SELECT

Input to select either command/address or data input.

6.1.10 $\overline{\text{SCE}}$: CHIP ENABLE

The enable pin allows data to be clocked in. The signal is active LOW.

6.1.11 OSC: OSCILLATOR

When the on-chip oscillator is used, this input must be connected to V_{DD}. An external clock signal, if used, is connected to this input. If the oscillator and external clock are both inhibited by connecting the OSC pin to V_{SS}, the display is not clocked and may be left in a DC state. To avoid this, the chip should always be put into Power-down mode before stopping the clock.

6.1.12 $\overline{\text{RES}}$: RESET

This signal will reset the device and must be applied to properly initialize the chip. The signal is active LOW.

7 FUNCTIONAL DESCRIPTION

7.1 Oscillator

The on-chip oscillator provides the clock signal for the display system. No external components are required and the OSC input must be connected to V_{DD}. An external clock signal, if used, is connected to this input.

7.2 Address Counter (AC)

The address counter assigns addresses to the display data RAM for writing. The X-address X_6 to X_0 and the Y-address Y_2 to Y_0 are set separately. After a write operation, the address counter is automatically incremented by 1, according to the V flag.

7.3 Display Data RAM (DDRAM)

The DDRAM is a 48 × 84 bit static RAM which stores the display data. The RAM is divided into six banks of 84 bytes (6 × 8 × 84 bits). During RAM access, data is transferred to the RAM through the serial interface. There is a direct correspondence between the X-address and the column output number.

7.4 Timing generator

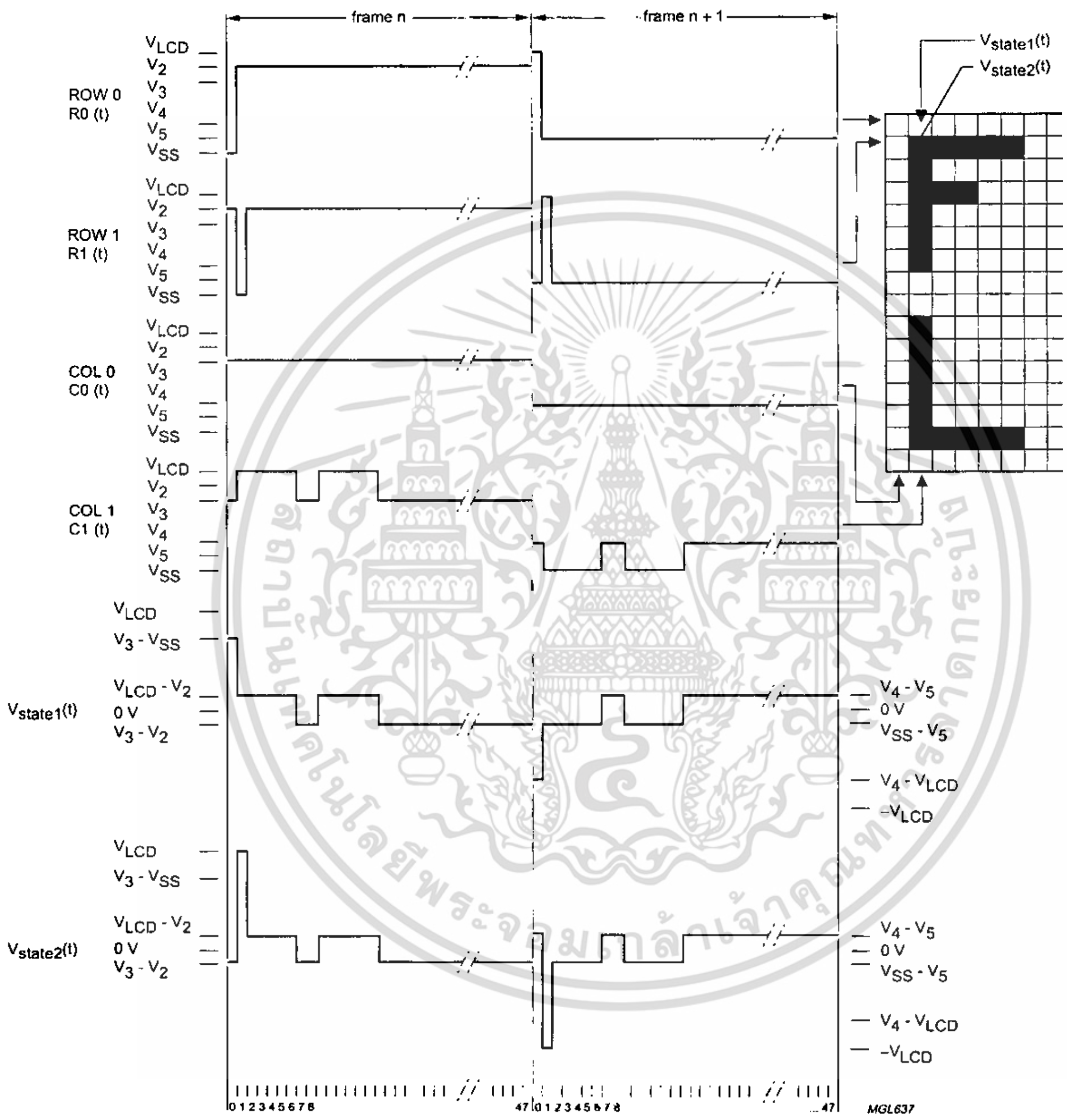
The timing generator produces the various signals required to drive the internal circuits. Internal chip operation is not affected by operations on the data buses.

7.5 Display address counter

The display is generated by continuously shifting rows of RAM data to the dot matrix LCD through the column outputs. The display status (all dots on/off and normal/inverse video) is set by bits E and D in the 'display control' command.

7.6 LCD row and column drivers

The PCD8544 contains 48 row and 84 column drivers, which connect the appropriate LCD bias voltages in sequence to the display in accordance with the data to be displayed. Figure 2 shows typical waveforms. Unused outputs should be left unconnected.



$V_{state1}(t) = C1(t) - R0(t)$.
$V_{state2}(t) = C1(t) - R1(t)$.

Fig.2 Typical LCD driver waveforms.

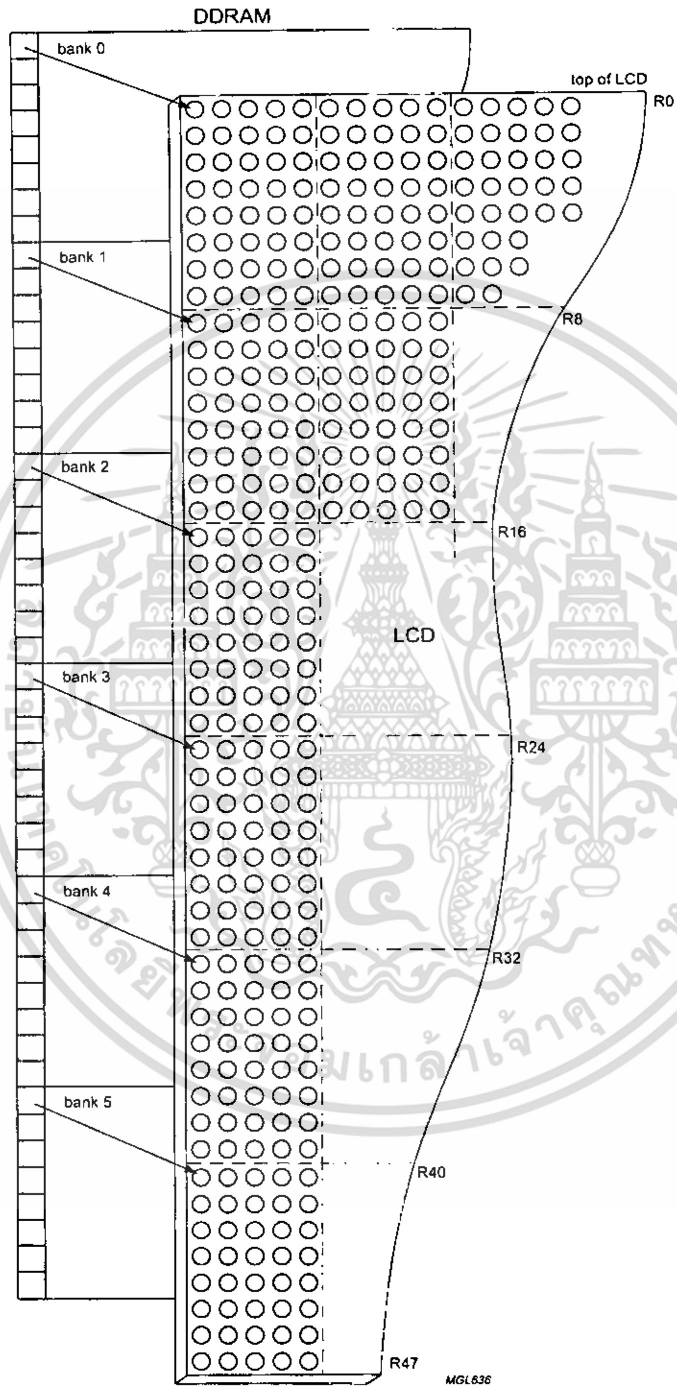


Fig.3 DDRAM to display mapping.

7.7 Addressing

Data is downloaded in bytes into the 48 by 84 bits RAM data display matrix of PCD8544, as indicated in Figs. 3, 4, 5 and 6. The columns are addressed by the address pointer. The address ranges are: X 0 to 83 (1010011), Y 0 to 5 (101). Addresses outside these ranges are not allowed. In the vertical addressing mode (V = 1), the Y address increments after each byte (see Fig.5). After the last Y address (Y = 5), Y wraps around to 0 and X increments to address the next column. In the horizontal addressing mode (V = 0), the X address increments after each byte (see Fig.6). After the last X address (X = 83), X wraps around to 0 and Y increments to address the next row. After the very last address (X = 83 and Y = 5), the address pointers wrap around to address (X = 0 and Y = 0).

7.7.1 DATA STRUCTURE

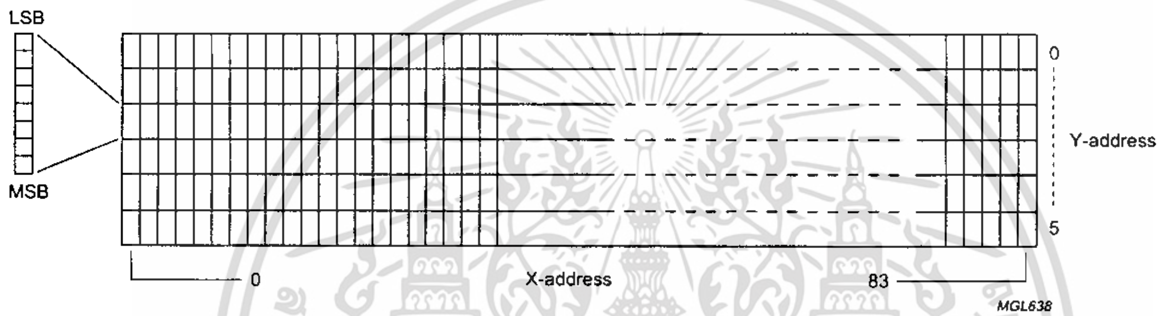


Fig.4 RAM format, addressing.

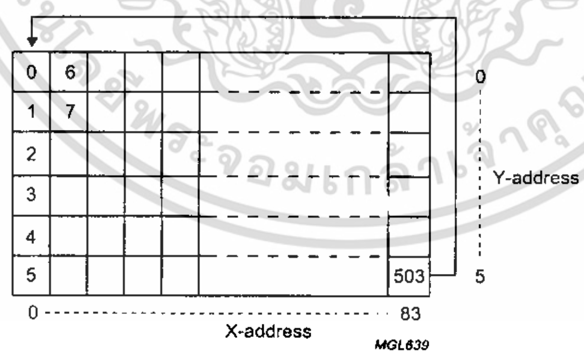


Fig.5 Sequence of writing data bytes into RAM with vertical addressing (V = 1).

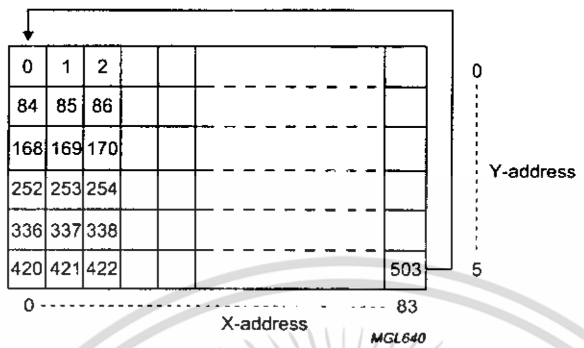
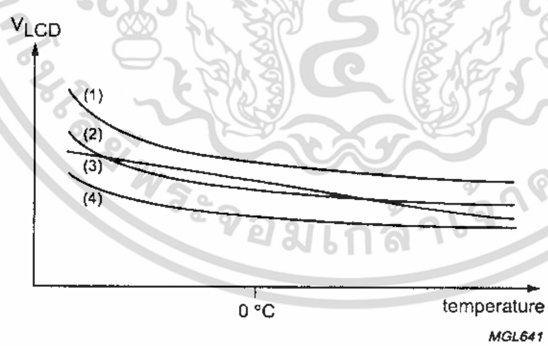


Fig.6 Sequence of writing data bytes into RAM with horizontal addressing (V = 0).

7.8 Temperature compensation

Due to the temperature dependency of the liquid crystals' viscosity, the LCD controlling voltage V_{LCD} must be increased at lower temperatures to maintain optimum contrast. Figure 7 shows V_{LCD} for high multiplex rates. In the PCD8544, the temperature coefficient of V_{LCD}, can be selected from four values (see Table 2) by setting bits TC_1 and TC_0.



(1) Upper limit.
(2) Typical curve.
(3) Temperature coefficient of IC.
(4) Lower limit.

Fig.7 V_{LCD} as function of liquid crystal temperature (typical values).

8 INSTRUCTIONS

The instruction format is divided into two modes: If D/$\overline{C}$ (mode select) is set LOW, the current byte is interpreted as command byte (see Table 1). Figure 8 shows an example of a serial data stream for initializing the chip. If D/$\overline{C}$ is set HIGH, the following bytes are stored in the display data RAM. After every data byte, the address counter is incremented automatically.

The level of the D/$\overline{C}$ signal is read during the last bit of data byte.

Each instruction can be sent in any order to the PCD8544. The MSB of a byte is transmitted first. Figure 9 shows one possible command stream, used to set up the LCD driver.

The serial interface is initialized when $\overline{SCE}$ is HIGH. In this state, SCLK clock pulses have no effect and no power is consumed by the serial interface. A negative edge on $\overline{SCE}$ enables the serial interface and indicates the start of a data transmission.

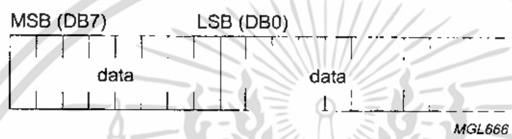


Fig.8 General format of data stream.

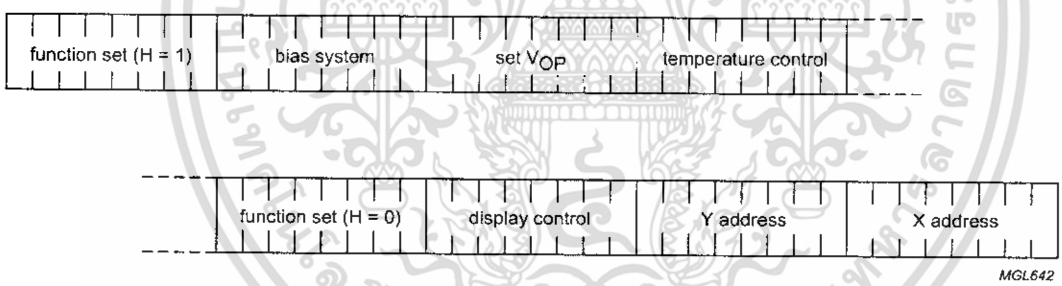


Fig.9 Serial data stream, example.

Figures 10 and 11 show the serial bus protocol.

- When $\overline{SCE}$ is HIGH, SCLK clock signals are ignored; during the HIGH time of $\overline{SCE}$, the serial interface is initialized (see Fig.12)
- SDIN is sampled at the positive edge of SCLK
- D/$\overline{C}$ indicates whether the byte is a command (D/$\overline{C}$ = 0) or RAM data (D/$\overline{C}$ = 1); it is read with the eighth SCLK pulse
- If $\overline{SCE}$ stays LOW after the last bit of a command/data byte, the serial interface expects bit 7 of the next byte at the next positive edge of SCLK (see Fig.12)
- A reset pulse with $\overline{RES}$ interrupts the transmission. No data is written into the RAM. The registers are cleared. If $\overline{SCE}$ is LOW after the positive edge of $\overline{RES}$, the serial interface is ready to receive bit 7 of a command/data byte (see Fig.13).

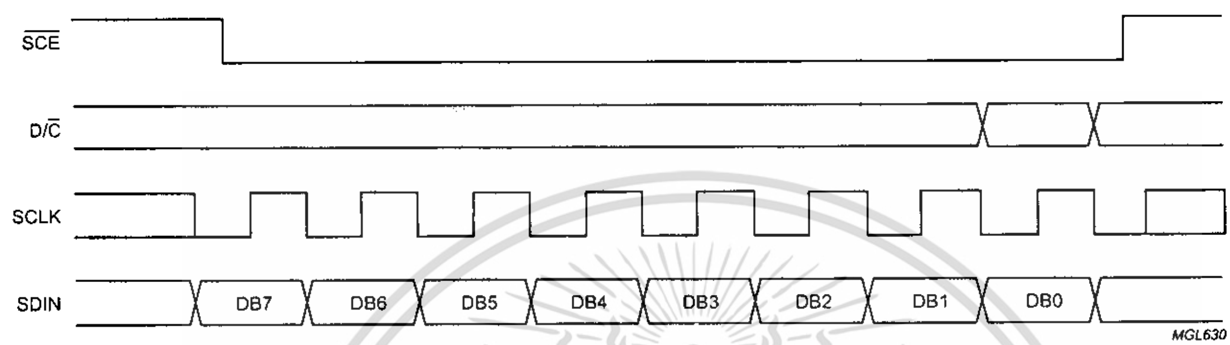


Fig.10 Serial bus protocol - transmission of one byte.

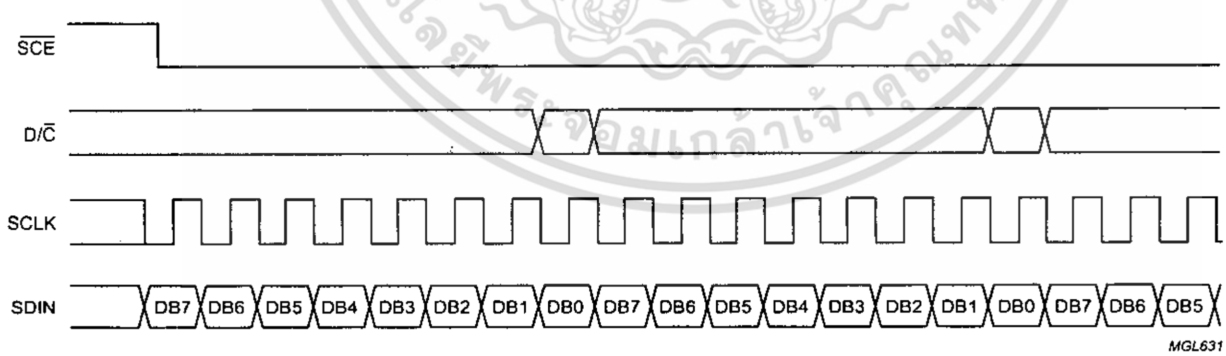


Fig.11 Serial bus protocol - transmission of several bytes.

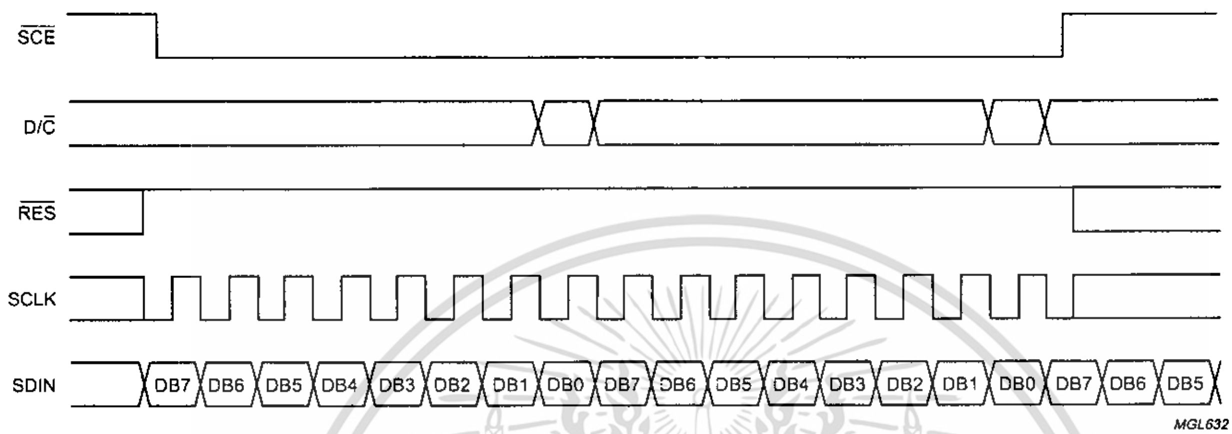


Fig.12 Serial bus reset function ($\overline{SCE}$).

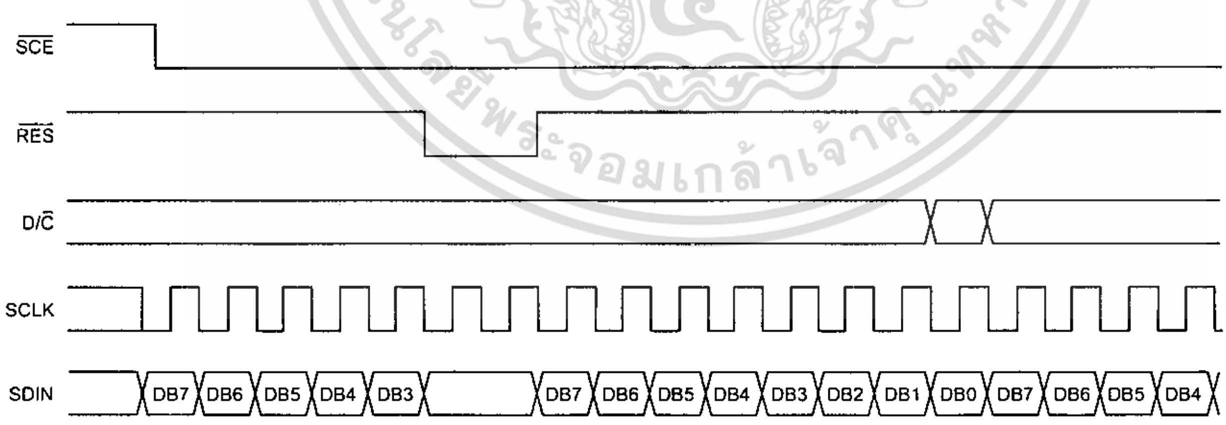


Fig.13 Serial bus reset function ($\overline{RES}$).

Table 1 Instruction set

INSTRUCTION	D/$\overline{C}$	COMMAND BYTE								DESCRIPTION
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H = 0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	no operation
Function set	0	0	0	1	0	0	PD	V	H	power down control; entry mode; extended instruction set control (H)
Write data	1	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	writes data to display RAM
(H = 0)										
Reserved	0	0	0	0	0	0	1	X	X	do not use
Display control	0	0	0	0	0	1	D	0	E	sets display configuration
Reserved	0	0	0	0	1	X	X	X	X	do not use
Set Y address of RAM	0	0	1	0	0	0	Y_2	Y_1	Y_0	sets Y-address of RAM; $0 \leq Y \leq 5$
Set X address of RAM	0	1	X_6	X_5	X_4	X_3	X_2	X_1	X_0	sets X-address part of RAM; $0 \leq X \leq 83$
(H = 1)										
Reserved	0	0	0	0	0	0	0	0	1	do not use
	0	0	0	0	0	0	0	1	X	do not use
Temperature control	0	0	0	0	0	0	1	TC_1	TC_0	set Temperature Coefficient (TC_x)
Reserved	0	0	0	0	0	1	X	X	X	do not use
Bias system	0	0	0	0	1	0	BS_2	BS_1	BS_0	set Bias System (BS_x)
Reserved	0	0	1	X	X	X	X	X	X	do not use
Set V_{OP}	0	1	V_{OP6}	V_{OP5}	V_{OP4}	V_{OP3}	V_{OP2}	V_{OP1}	V_{OP0}	write V_{OP} to register

Table 2 Explanations of symbols in Table 1

BIT	0	1
PD	chip is active	chip is in Power-down mode
V	horizontal addressing	vertical addressing
H	use basic instruction set	use extended instruction set
D and E		
00	display blank	
10	normal mode	
01	all display segments on	
11	inverse video mode	
TC_1 and TC_0		
00	V_{LCD} temperature coefficient 0	
01	V_{LCD} temperature coefficient 1	
10	V_{LCD} temperature coefficient 2	
11	V_{LCD} temperature coefficient 3	

8.1 Initialization

Immediately following power-on, the contents of all internal registers and of the RAM are undefined. **A $\overline{\text{RES}}$ pulse must be applied**. Attention should be paid to the possibility that the **device may be damaged** if not properly reset.

All internal registers are reset by applying an external $\overline{\text{RES}}$ pulse (active LOW) at pad 31, within the specified time. However, the RAM contents are still undefined. The state after reset is described in Section 8.2.

The $\overline{\text{RES}}$ input must be $\leq 0.3V_{DD}$ when V_{DD} reaches V_{DDmin} (or higher) within a maximum time of 100 ms after V_{DD} goes HIGH (see Fig.16).

8.2 Reset function

After reset, the LCD driver has the following state:

- Power-down mode (bit PD = 1)
- Horizontal addressing (bit V = 0) normal instruction set (bit H = 0)
- Display blank (bit E = D = 0)
- Address counter X_6 to X_0 = 0; Y_2 to Y_0 = 0
- Temperature control mode (TC_1 TC_0 = 0)
- Bias system (BS_2 to BS_0 = 0)
- V_{LCD} is equal to 0, the HV generator is switched off (V_{OP6} to V_{OP0} = 0)
- After power-on, the RAM contents are undefined.

8.3 Function set

8.3.1 Bit PD

- All LCD outputs at V_{SS} (display off)
- Bias generator and V_{LCD} generator off, V_{LCD} can be disconnected
- Oscillator off (external clock possible)
- Serial bus, command, etc. function
- Before entering Power-down mode, the RAM needs to be filled with '0's to ensure the specified current consumption.

8.3.2 Bit V

When V = 0, the horizontal addressing is selected. The data is written into the DDRAM as shown in Fig.6. When V = 1, the vertical addressing is selected. The data is written into the DDRAM, as shown in Fig.5.

8.3.3 Bit H

When H = 0 the commands 'display control', 'set Y address' and 'set X address' can be performed; when H = 1, the others can be executed. The 'write data' and 'function set' commands can be executed in both cases.

8.4 Display control

8.4.1 Bits D and E

Bits D and E select the display mode (see Table 2).

8.5 Set Y address of RAM

Y_n defines the Y vector addressing of the display RAM.

Table 3 Y vector addressing

Y_2	Y_1	Y_0	BANK
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

8.6 Set X address of RAM

The X address points to the columns. The range of X is 0 to 83 (53H).

8.7 Temperature control

The temperature coefficient of V_{LCD} is selected by bits TC_1 and TC_0.

8.8 Bias value

The bias voltage levels are set in the ratio of R - R - nR - R - R, giving a 1/(n + 4) bias system. Different multiplex rates require different factors n (see Table 4). This is programmed by BS_2 to BS_0. For Mux 1 : 48, the optimum bias value n, resulting in 1/8 bias, is given by:

$$n = \sqrt{48} - 3 = 3.928 = 4 \quad (1)$$

Table 4 Programming the required bias system

BS_2	BS_1	BS_0	n	RECOMMENDED MUX RATE
0	0	0	7	1 : 100
0	0	1	6	1 : 80
0	1	0	5	1 : 65/1 : 65
0	1	1	4	1 : 48
1	0	0	3	1 : 40/1 : 34
1	0	1	2	1 : 24
1	1	0	1	1 : 18/1 : 16
1	1	1	0	1 : 10/1 : 9/1 : 8

Table 5 LCD bias voltage

SYMBOL	BIAS VOLTAGES	BIAS VOLTAGE FOR 1/8 BIAS
V1	V_{LCD}	V_{LCD}
V2	(n + 3)/(n + 4)	$\frac{7}{8} \times V_{LCD}$
V3	(n + 2)/(n + 4)	$\frac{6}{8} \times V_{LCD}$
V4	2/(n + 4)	$\frac{2}{8} \times V_{LCD}$
V5	1/(n + 4)	$\frac{1}{8} \times V_{LCD}$
V6	V_{SS}	V_{SS}

8.9 Set V_{OP} value

The operation voltage V_{LCD} can be set by software. The values are dependent on the liquid crystal selected. $V_{LCD} = a + (V_{OP6}$ to $V_{OP0}) \times b$ [V]. In the PCD8544, a = 3.06 and b = 0.06 giving a program range of 3.00 to 10.68 at room temperature.

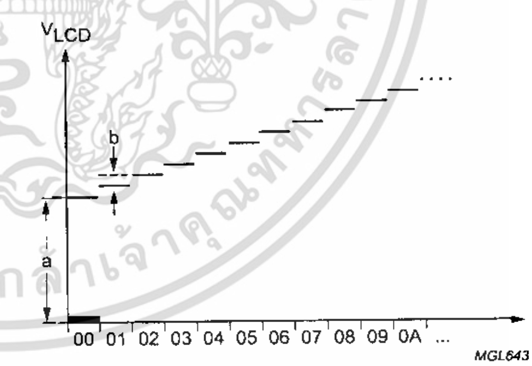
Note that the charge pump is turned off if V_{OP6} to V_{OP0} is set to zero.

For Mux 1 : 48, the optimum operation voltage of the liquid can be calculated as:

$$V_{LCD} = \frac{1+\sqrt{48}}{\sqrt{2 \cdot \left(1 - \frac{1}{\sqrt{48}}\right)}} \cdot V_{th} = 6.06 \cdot V_{th} \qquad (2)$$

where V_{th} is the threshold voltage of the liquid crystal material used.

Caution, as V_{OP} increases with lower temperatures, care must be taken not to set a V_{OP} that will exceed the maximum of 8.5 V when operating at −25 °C.



a = 3.06.
b = 0.06.
V_{OP6} to V_{OP0} (programmed) [00 to 7FH].

Fig.14 V_{OP} programming.

9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134); see notes 1 and 2.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DD}	supply voltage	note 3	−0.5	+7	V
V_{LCD}	supply voltage LCD	note 4	−0.5	+10	V
V_i	all input voltages		−0.5	V_{DD} + 0.5	V
I_{SS}	ground supply current		−50	+50	mA
I_I, I_O	DC input or output current		−10	+10	mA
P_{tot}	total power dissipation		–	300	mW
P_O	power dissipation per output		–	30	mW
T_{amb}	operating ambient temperature		−25	+70	°C
T_j	operating junction temperature		−65	+150	°C
T_{stg}	storage temperature		−65	+150	°C

Notes

1. Stresses above those listed under limiting values may cause permanent damage to the device.
2. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.
3. With external LCD supply voltage externally supplied (voltage generator disabled). V_{DDmax} = 5 V if LCD supply voltage is internally generated (voltage generator enabled).
4. When setting V_{LCD} by software, take care not to set a V_{OP} that will exceed the maximum of 8.5 V when operating at −25 °C, see Caution in Section 8.9.

10 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

11 DC CHARACTERISTICS

V_{DD} = 2.7 to 3.3 V; V_{SS} = 0 V; V_{LCD} = 6.0 to 9.0 V; T_{amb} = −25 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{DD1}	supply voltage 1	LCD voltage externally supplied (voltage generator disabled)	2.7	–	3.3	V
V_{DD2}	supply voltage 2	LCD voltage internally generated (voltage generator enabled)	2.7	–	3.3	V
V_{LCD1}	LCD supply voltage	LCD voltage externally supplied (voltage generator disabled)	6.0	–	9.0	V
V_{LCD2}	LCD supply voltage	LCD voltage internally generated (voltage generator enabled); note 1	6.0	–	8.5	V
I_{DD1}	supply current 1 (normal mode) for internal V_{LCD}	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; T_{amb} = 25 °C; display load = 10 μA; note 2	–	240	300	μA
I_{DD2}	supply current 2 (normal mode) for internal V_{LCD}	V_{DD} = 2.70 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; T_{amb} = 25 °C; display load = 10 μA; note 2	–	–	320	μA
I_{DD3}	supply current 3 (Power-down mode)	with internal or external LCD supply voltage; note 3	–	1.5	–	μA
I_{DD4}	supply current external V_{LCD}	V_{DD} = 2.85 V; V_{LCD} = 9.0 V; f_{SCLK} = 0; notes 2 and 4	–	25	–	μA
I_{LCD}	supply current external V_{LCD}	V_{DD} = 2.7 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; T = 25 °C; display load = 10 μA; notes 2 and 4	–	42	–	μA
Logic						
V_{IL}	LOW level input voltage		V_{SS}	–	$0.3V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7V_{DD}$	–	V_{DD}	V
I_L	leakage current	V_I = V_{DD} or V_{SS}	−1	–	+1	μA
Column and row outputs						
$R_{o(C)}$	column output resistance C0 to C83		–	12	20	kΩ
$R_{o(R)}$	row output resistance R0 to R47		–	12	20	kΩ
$V_{bias(tol)}$	bias voltage tolerance on C0 to C83 and R0 to R47		−100	0	+100	mV

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
LCD supply voltage generator						
V_{LCD}	V_{LCD} tolerance internally generated	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; display load = 10 μA; note 5	–	0	300	mV
TC0	V_{LCD} temperature coefficient 0	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; display load = 10 μA	–	1	–	mV/K
TC1	V_{LCD} temperature coefficient 1	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; display load = 10 μA	–	9	–	mV/K
TC2	V_{LCD} temperature coefficient 2	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; display load = 10 μA	–	17	–	mV/K
TC3	V_{LCD} temperature coefficient 3	V_{DD} = 2.85 V; V_{LCD} = 7.0 V; f_{SCLK} = 0; display load = 10 μA	–	24	–	mV/K

Notes

1. The maximum possible V_{LCD} voltage that may be generated is dependent on voltage, temperature and (display) load.
2. Internal clock.
3. RAM contents equal '0'. During power-down, all static currents are switched off.
4. If external V_{LCD}, the display load current is not transmitted to I_{DD}.
5. Tolerance depends on the temperature (typically zero at 27 °C, maximum tolerance values are measured at the temperate range limit).

12 AC CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
f_{OSC}	oscillator frequency		20	34	65	kHz
$f_{clk(ext)}$	external clock frequency		10	32	100	kHz
f_{frame}	frame frequency	f_{OSC} or $f_{clk(ext)}$ = 32 kHz; note 1	–	67	–	Hz
t_{VHRL}	V_{DD} to $\overline{RES}$ LOW	Fig.16	0(2)	–	30	ms
$t_{WL(RES)}$	$\overline{RES}$ LOW pulse width	Fig.16	100	–	–	ns
Serial bus timing characteristics						
f_{SCLK}	clock frequency	V_{DD} = 3.0 V ±10%	0	–	4.00	MHz
T_{cy}	clock cycle SCLK	All signal timing is based on 20% to 80% of V_{DD} and maximum rise and fall times of 10 ns	250	–	–	ns
t_{WH1}	SCLK pulse width HIGH		100	–	–	ns
t_{WL1}	SCLK pulse width LOW		100	–	–	ns
t_{su2}	$\overline{SCE}$ set-up time		60	–	–	ns
t_{h2}	$\overline{SCE}$ hold time		100	–	–	ns
t_{WH2}	$\overline{SCE}$ min. HIGH time		100	–	–	ns
t_{h5}	$\overline{SCE}$ start hold time; note 3		100	–	–	ns
t_{su3}	$D/\overline{C}$ set-up time		100	–	–	ns
t_{h3}	$D/\overline{C}$ hold time		100	–	–	ns
t_{su4}	SDIN set-up time		100	–	–	ns
t_{h4}	SDIN hold time		100	–	–	ns

Notes

1. $T_{frame} = \frac{f_{clk\,(ext)}}{480}$
2. $\overline{RES}$ may be LOW before V_{DD} goes HIGH.
3. t_{h5} is the time from the previous SCLK positive edge (irrespective of the state of $\overline{SCE}$) to the negative edge of $\overline{SCE}$ (see Fig.15).

12.1 Serial interface

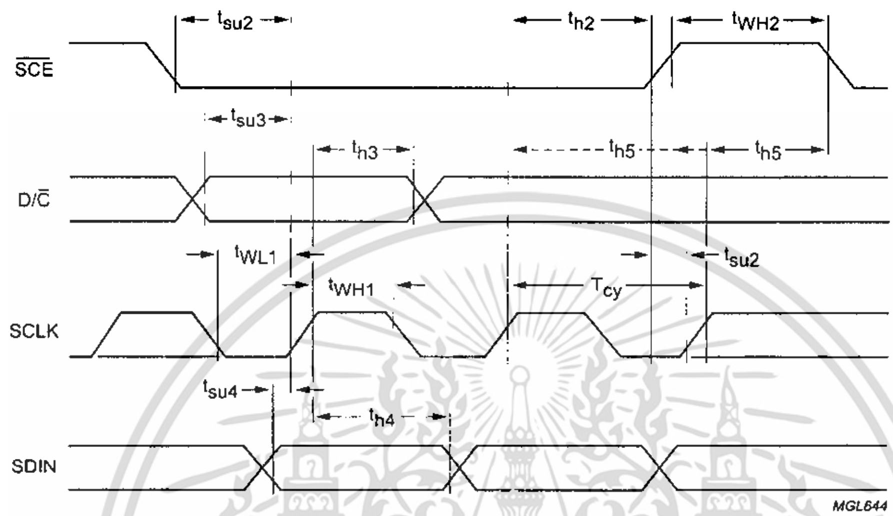


Fig.15 Serial interface timing.

12.2 Reset

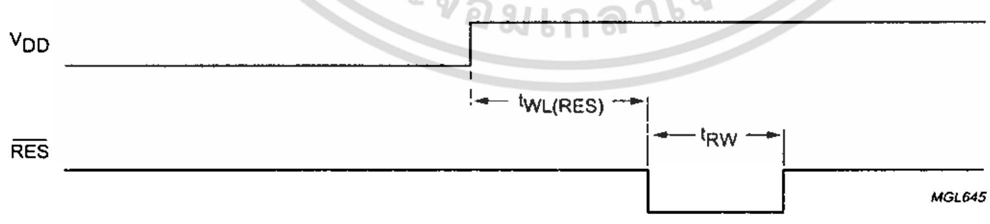
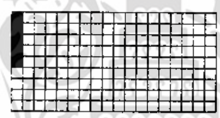
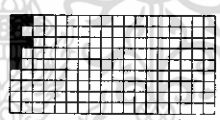
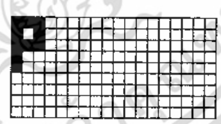
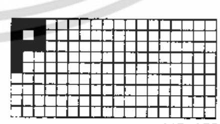
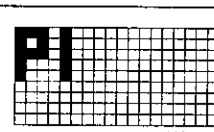
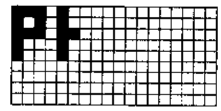
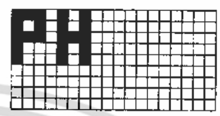


Fig.16 Reset timing.

13 APPLICATION INFORMATION

Table 6 Programming example

STEP	SERIAL BUS BYTE									DISPLAY	OPERATION
	D/$\overline{C}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	start										$\overline{SCE}$ is going LOW
2	0	0	0	1	0	0	0	0	1		function set PD = 0 and V = 0, select extended instruction set (H = 1 mode)
3	0	1	0	0	1	0	0	0	0		set V_{OP}; V_{OP} is set to a +16 × b [V]
4	0	0	0	1	0	0	0	0	0		function set PD = 0 and V = 0, select normal instruction set (H = 0 mode)
5	0	0	0	0	0	1	1	0	0		display control set normal mode (D = 1 and E = 0)
6	1	0	0	0	1	1	1	1	1	 MGL673	data write Y and X are initialized to 0 by default, so they are not set here
7	1	0	0	0	0	0	1	0	1	 MGL674	data write
8	1	0	0	0	0	0	1	1	1	 MGL675	data write
9	1	0	0	0	0	0	0	0	0	 MGL675	data write
10	1	0	0	0	1	1	1	1	1	 MGL676	data write

STEP	SERIAL BUS BYTE									DISPLAY	OPERATION
	D/$\overline{C}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
11	1	0	0	0	0	0	1	0	0		data write
12	1	0	0	0	1	1	1	1	1		data write
13	0	0	0	0	0	1	1	0	1		display control; set inverse video mode (D = 1 and E = 1)
14	0	1	0	0	0	0	0	0	0		set X address of RAM; set address to '0000000'
15	1	0	0	0	0	0	0	0	0		data write

'he pinning is optimized for single plane wiring e.g. for chip-on-glass display modules. Display size: 48 × 84 pixels.

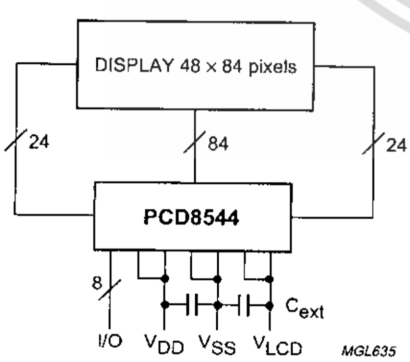


Fig.17 Application diagram.

The required minimum value for the external capacitors is: C_{ext} = 1.0 μF.

Higher capacitor values are recommended for ripple reduction.

14 BONDING PAD LOCATIONS

14.1 Bonding pad information (see Fig.18)

PARAMETER	SIZE
Pad pitch	min. 100 μm
Pad size, aluminium	80 × 100 μm
Bump dimensions	59 × 89 × 17.5 (±5) μm
Wafer thickness	max. 380 μm

14.2 Bonding pad location

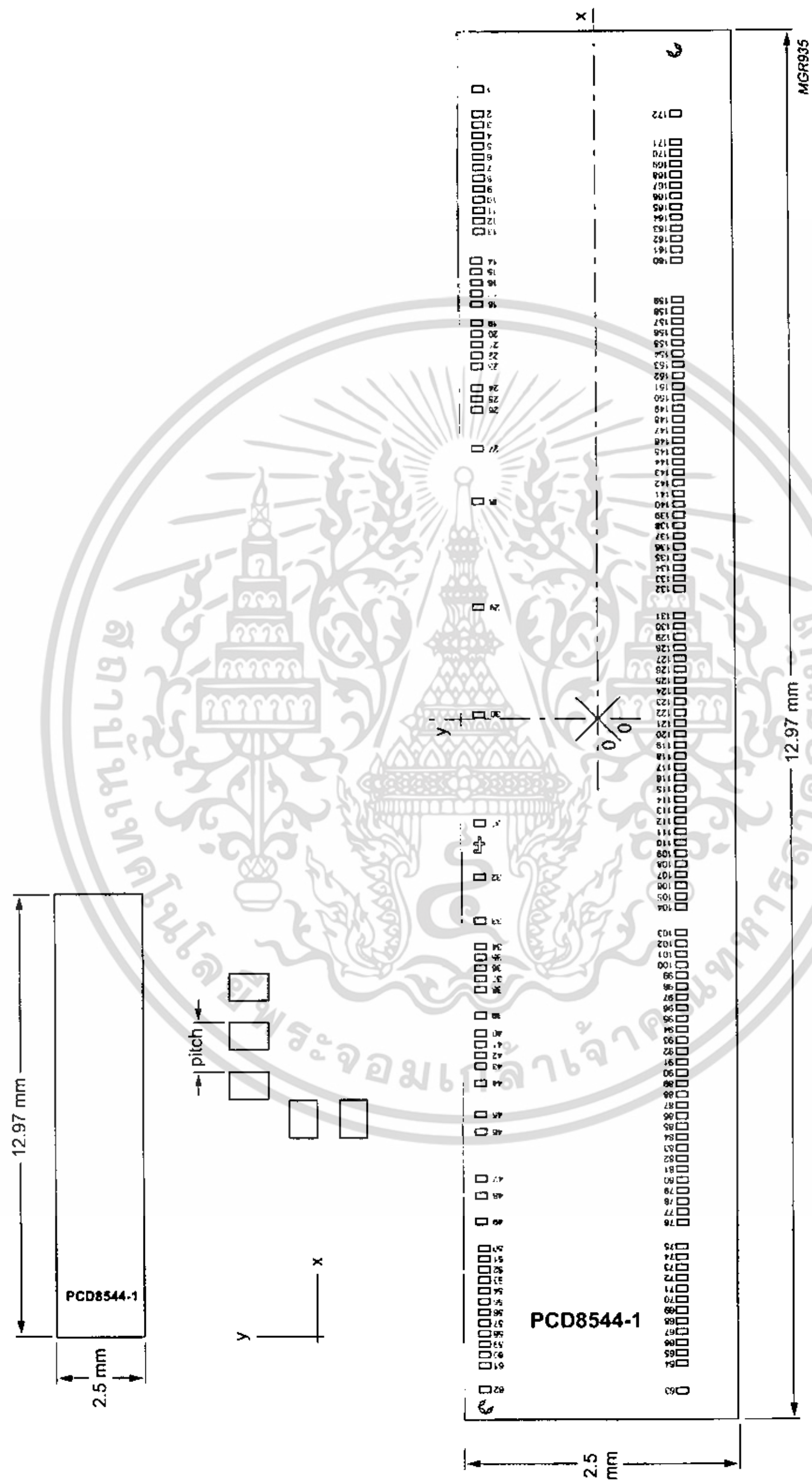


Fig.18 Bonding pad locations.

Table 7 Bonding pad locations (dimensions in μm). All X/Y coordinates are referenced to the centre of chip (see Fig.18)

PAD	PAD NAME	x	y
1	dummy1	+5932	+1060
2	R36	+5704	+1060
3	R37	+5604	+1060
4	R38	+5504	+1060
5	R39	+5404	+1060
6	R40	+5304	+1060
7	R41	+5204	+1060
8	R42	+5104	+1060
9	R43	+5004	+1060
10	R44	+4904	+1060
11	R45	+4804	+1060
12	R46	+4704	+1060
13	R47	+4604	+1060
14	V_{DD1}	+4330	+1085
15	V_{DD1}	+4230	+1085
16	V_{DD1}	+4130	+1085
17	V_{DD1}	+4030	+1085
18	V_{DD1}	+3930	+1085
19	V_{DD2}	+3750	+1085
20	V_{DD2}	+3650	+1085
21	V_{DD2}	+3550	+1085
22	V_{DD2}	+3450	+1085
23	V_{DD2}	+3350	+1085
24	V_{DD2}	+3250	+1085
25	V_{DD2}	+3150	+1085
26	V_{DD2}	+3050	+1085
27	SCLK	+2590	+1085
28	SDIN	+2090	+1085
29	D/$\overline{C}$	+1090	+1085
30	$\overline{SCE}$	+90	+1085
31	$\overline{RES}$	−910	+1085
32	OSC	−1410	+1085
33	T3	−1826	+1085
34	V_{SS2}	−2068	+1085
35	V_{SS2}	−2168	+1085
36	V_{SS2}	−2268	+1085
37	V_{SS2}	−2368	+1085
38	V_{SS2}	−2468	+1085
39	T4	−2709	+1085
40	V_{SS1}	−2876	+1085
41	V_{SS1}	−2976	+1085
42	V_{SS1}	−3076	+1085
43	V_{SS1}	−3176	+1085
44	T1	−3337	+1085
45	V_{LCD2}	−3629	+1085
46	V_{LCD2}	−3789	+1085
47	V_{LCD1}	−4231	+1085
48	V_{LCD1}	−4391	+1085
49	T2	−4633	+1085
50	R23	−4894	+1060
51	R22	−4994	+1060
52	R21	−5094	+1060
53	R20	−5194	+1060
54	R19	−5294	+1060
55	R18	−5394	+1060
56	R17	−5494	+1060
57	R16	−5594	+1060
58	R15	−5694	+1060
59	R14	−5794	+1060
60	R13	−5894	+1060
61	R12	−5994	+1060
62	dummy2	−6222	+1060
63	dummy3	−6238	−738
64	R0	−5979	−738
65	R1	−5879	−738
66	R2	−5779	−738
67	R3	−5679	−738
68	R4	−5579	−738
69	R5	−5479	−738
70	R6	−5379	−738
71	R7	−5279	−738
72	R8	−5179	−738
73	R9	−5079	−738
74	R10	−4979	−738
75	R11	−4879	−738
76	C0	−4646	−746

48 × 84 pixels matrix LCD controller/driver

PCD8544

PAD	PAD NAME	x	y
77	C1	−4546	−746
78	C2	−4446	−746
79	C3	−4346	−746
80	C4	−4246	−746
81	C5	−4146	−746
82	C6	−4046	−746
83	C7	−3946	−746
84	C8	−3846	−746
85	C9	−3746	−746
86	C10	−3646	−746
87	C11	−3546	−746
88	C12	−3446	−746
89	C13	−3346	−746
90	C14	−3246	−746
91	C15	−3146	−746
92	C16	−3046	−746
93	C17	−2946	−746
94	C18	−2846	−746
95	C19	−2746	−746
96	C20	−2646	−746
97	C21	−2546	−746
98	C22	−2446	−746
99	C23	−2346	−746
100	C24	−2246	−746
101	C25	−2146	−746
102	C26	−2046	−746
103	C27	−1946	−746
104	C28	−1696	−746
105	C29	−1596	−746
106	C30	−1496	−746
107	C31	−1396	−746
108	C32	−1296	−746
109	C33	−1196	−746
110	C34	−1096	−746
111	C35	−996	−746
112	C36	−896	−746
113	C37	−796	−746
114	C38	−696	−746
115	C39	−596	−746
116	C40	−496	−746
117	C41	−396	−746

PAD	PAD NAME	x	y
118	C42	−296	−746
119	C43	−196	−746
120	C44	−96	−746
121	C45	+4	−746
122	C46	+104	−746
123	C47	+204	−746
124	C48	+304	−746
125	C49	+404	−746
126	C50	+504	−746
127	C51	+604	−746
128	C52	+704	−746
139	C53	+804	−746
130	C54	+904	−746
131	C55	+1004	−746
132	C56	+1254	−746
133	C57	+1354	−746
134	C58	+1454	−746
135	C59	+1554	−746
136	C60	+1654	−746
137	C61	+1754	−746
138	C62	+1854	−746
139	C63	+1954	−746
140	C64	+2054	−746
141	C65	+2154	−746
142	C66	+2254	−746
143	C67	+2354	−746
144	C68	+2454	−746
145	C69	+2554	−746
146	C70	+2654	−746
147	C71	+2754	−746
148	C72	+2854	−746
149	C73	+2954	−746
150	C74	+3054	−746
151	C75	+3154	−746
152	C76	+3254	−746
153	C77	+3354	−746
154	C78	+3454	−746
155	C79	+3554	−746
156	C80	+3654	−746
157	C81	+3754	−746
158	C82	+3854	−746

PAD	PAD NAME	x	y
159	C83	+3954	−746
160	R35	+4328	−738
161	R34	+4428	−738
162	R33	+4528	−738
163	R32	+4628	−738
164	R31	+4728	−738
165	R30	+4828	−738
166	R29	+4928	−738
167	R28	+5028	−738
168	R27	+5128	−738
169	R26	+5228	−738
170	R25	+5328	−738
171	R24	+5428	−738
172	dummy4	+5694	−738

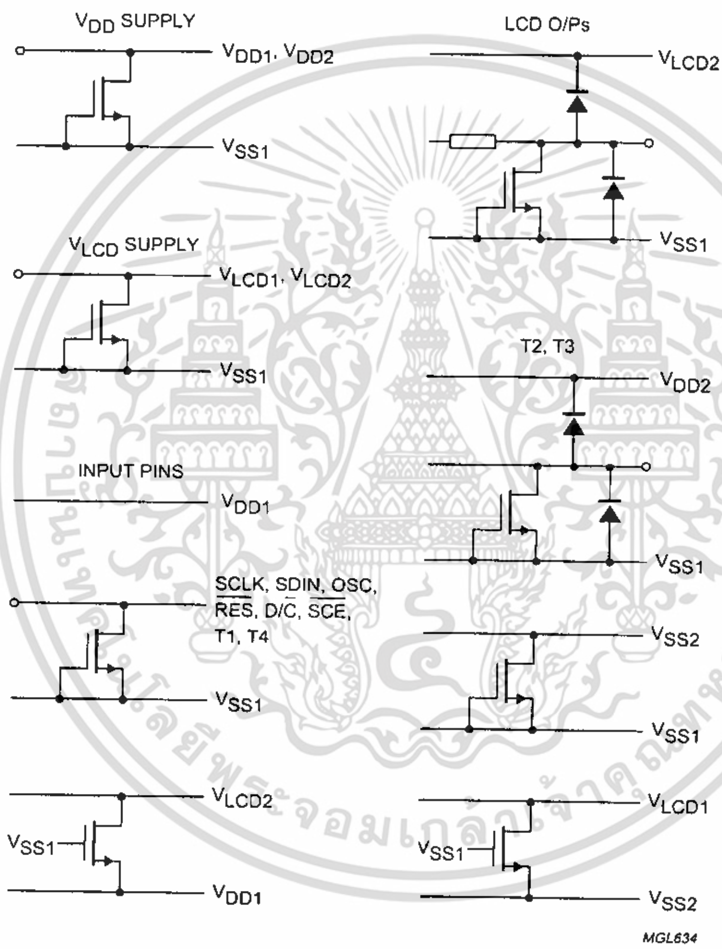
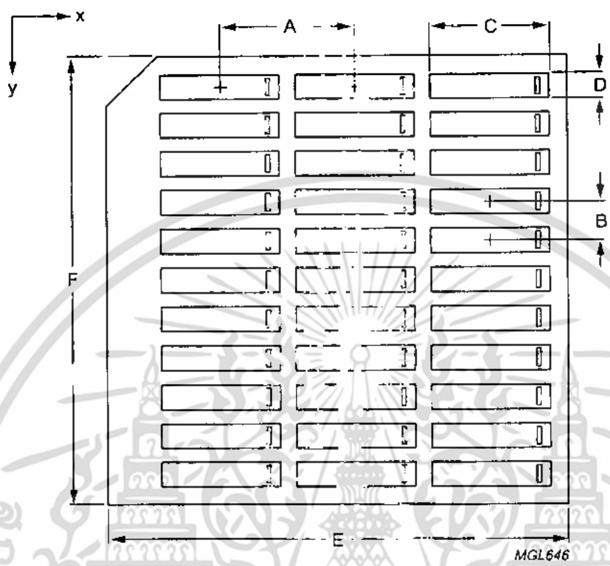


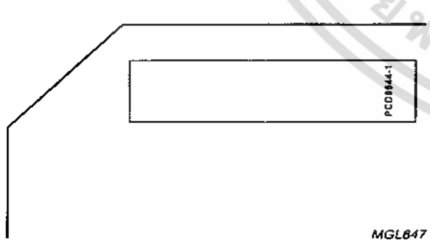
Fig.19 Device protection diagram.

15 TRAY INFORMATION



For the dimensions of x, y and A to F, see Table 8.

Fig.20 Tray details.



The orientation of the IC in a pocket is indicated by the position of the IC type name on the die surface with respect to the chamfer on the upper left corner of the tray. Refer to the bonding pad location diagram for the orientation and position of the type name on the die surface.

Fig.21 Tray alignment.

Table 8 Dimensions

DIM.	DESCRIPTION	VALUE
A	pocket pitch, in the x direction	14.82 mm
B	pocket pitch, in the y direction	4.39 mm
C	pocket width, in the x direction	13.27 mm
D	pocket width, in the y direction	2.8 mm
E	tray width, in the x direction	50.67 mm
F	tray width, in the y direction	50.67 mm
x	no. of pockets in the x direction	3
y	no. of pockets in the y direction	11

16 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

17 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

NOTES

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113, Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213, Tel. +43 1 60 101 1248, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6, 220050 MINSK, Tel. +375 172 20 0733, Fax. +375 172 20 0773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor, 51 James Bourchier Blvd., 1407 SOFIA, Tel. +359 2 68 9211, Fax. +359 2 68 9102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS, Tel. +1 800 234 7381, Fax. +1 800 943 0087

China/Hong Kong: 501 Hong Kong Industrial Technology Centre, 72 Tat Chee Avenue, Kowloon Tong, HONG KONG, Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Sydhavnsgade 23, 1780 COPENHAGEN V, Tel. +45 33 29 3333, Fax. +45 33 29 3905

Finland: Sinikalliontie 3, FIN-02630 ESPOO, Tel. +358 9 615 800, Fax. +358 9 6158 0920

France: 51 Rue Carnot, BP317, 92156 SURESNES Cedex, Tel. +33 1 4099 6161, Fax. +33 1 4099 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG, Tel. +49 40 2353 60, Fax. +49 40 2353 6300

Hungary: see Austria

India: Philips INDIA Ltd, Band Box Building, 2nd floor, 254-D, Dr. Annie Besant Road, Worli, MUMBAI 400 025, Tel. +91 22 493 8541, Fax. +91 22 493 0966

Indonesia: PT Philips Development Corporation, Semiconductors Division, Gedung Philips, Jl. Buncit Raya Kav.99-100, JAKARTA 12510, Tel. +62 21 794 0040 ext. 2501, Fax. +62 21 794 0080

Ireland: Newstead, Clonskeagh, DUBLIN 14, Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, PO Box 18053, TEL AVIV 61180, Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3, 20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108-8507, Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL, Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR, Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905, Tel. +9-5 800 234 7381, Fax +9-5 800 943 0087

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB, Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND, Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO, Tel. +47 22 74 8000, Fax. +47 22 74 8341

Pakistan: see Singapore

Philippines: Philips Semiconductors Philippines Inc., 106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI, Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA, Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW, Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 319762, Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale, 2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000, Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Al. Vicente Pinzon, 173, 6th floor, 04547-130 SÃO PAULO, SP, Brazil, Tel. +55 11 821 2333, Fax. +55 11 821 2382

Spain: Balmes 22, 08007 BARCELONA, Tel. +34 93 301 6312, Fax. +34 93 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM, Tel. +46 8 5985 2000, Fax. +46 8 5985 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH, Tel. +41 1 488 2741 Fax. +41 1 488 3263

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1, TAIPEI, Taiwan Tel. +886 2 2134 2886, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd., 209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260, Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL, Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7, 252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes, MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409, Tel. +1 800 234 7381, Fax. +1 800 943 0087

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD, Tel. +381 11 62 5344, Fax.+381 11 63 5777

For all other countries apply to: Philips Semiconductors, International Marketing & Sales Communications, Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com

SCA63

Printed in The Netherlands 465008/750/01/pp32 Date of release: 1999 Apr 12 Document order number: 9397 750 05024

Let's make things better.

Philips Semiconductors

PHILIPS