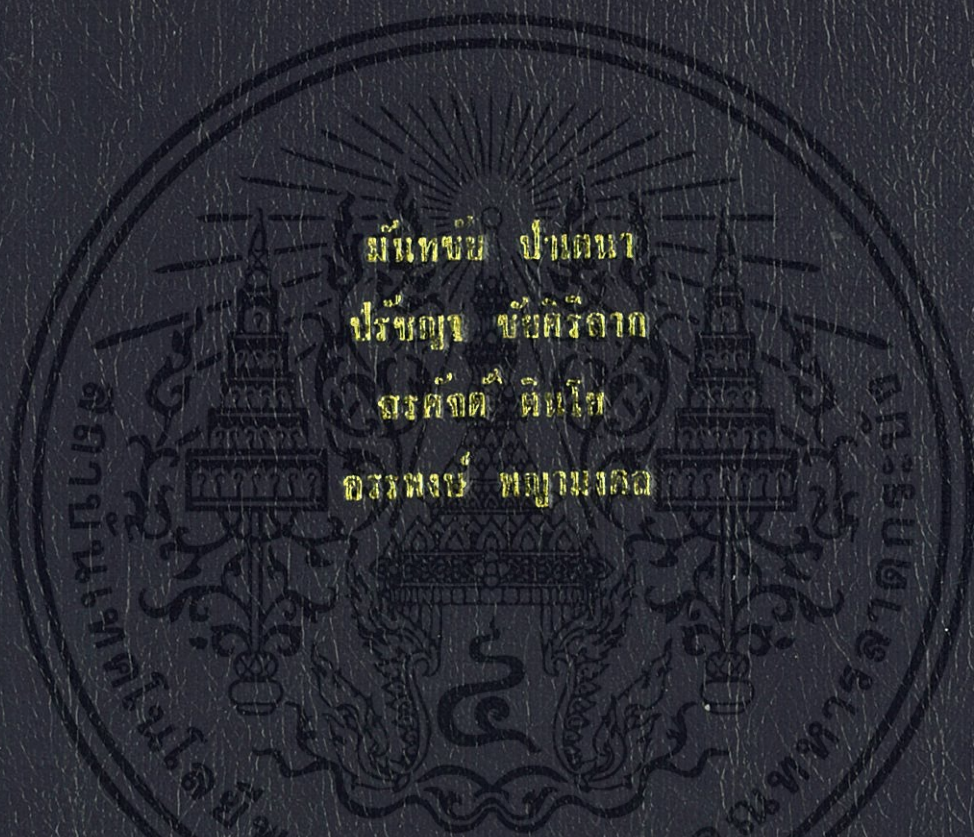


ระบบติดตามวัตถุอัตโนมัติด้วยเรดาร์
AUTOMATIC OBJECT FOLLOWING OF HELICOPTER



ปริญญาโทชั้นเรียนวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมระบบการขนส่ง
สาขาวิศวกรรมระบบการควบคุม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

ระบบติดตามวัตถุอัตโนมัติด้วยเฮลิคอปเตอร์
AUTOMATIC OBJECT FOLLOWING OF HELICOPTER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC OBJECT FOLLOWING OF HELICOPTER



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2555
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบติดตามวัตถุอัตโนมัติด้วยเฮลิคอปเตอร์
AUTOMATIC OBJECT FOLLOWING OF HELICOPTER

นักศึกษาผู้จัดทำ นายนันท์ชัย ปาเตนา รหัสนักศึกษา 52011594
นายปรัชญา ชัยศิริลาภ รหัสนักศึกษา 52010685
นายสรศักดิ์ ตินโย รหัสนักศึกษา 52011250
นายอรรถพงษ์ พญามงคล รหัสนักศึกษา 52011409

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2555

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ดร. พุศศักดิ์ ชิวสุวิทย์	

หัวข้อปริญญาโท	ระบบติดตามวัตถุอัตโนมัติด้วยเฮลิคอปเตอร์			
	AUTOMATIC OBJECT FOLLOWING OF HELICOPTER			
นักศึกษาผู้จัดทำ	นายฉันทชัย	ปาเตนา	รหัสนักศึกษา	52011594
	นายปรัชญา	ชัยศิริลาภ	รหัสนักศึกษา	52010685
	นายสรศักดิ์	ตินโย	รหัสนักศึกษา	52011250
	นายอรรถพงษ์	พญามงคล	รหัสนักศึกษา	52011409
อาจารย์ที่ปรึกษา	รศ.ดร. พุศศักดิ์	ชีวิสุวิทย์		
ปีการศึกษา	2555			

บทคัดย่อ

โครงการนี้เป็นการศึกษาและประยุกต์ใช้งานเฮลิคอปเตอร์บังคับวิทยุในการติดตามวัตถุ โดยใช้กล้องเว็บแคมทำการรับภาพ และส่งข้อมูลให้กับคอมพิวเตอร์ เพื่อทำการประมวลผลภาพและตรวจหาตำแหน่งวัตถุ และใช้ไมโครคอนโทรลเลอร์ตระกูล PIC เป็นตัวควบคุมการทำงาน เพื่อส่งสัญญาณให้กับตัวปรับความต้านทานแบบดิจิตอล เพื่อควบคุมการทำงานของรีโมทคอนโทรลให้สามารถบังคับสั่งการเคลื่อนที่ของเฮลิคอปเตอร์ให้สามารถเคลื่อนที่ติดตามวัตถุเป้าหมายได้



Thesis Title	AUTOMATIC OBJECT FOLLOWING OF HELICOPTER	
Authors	Mr. Nantachai	Patena
	Mr. Prachaya	Chaisirilarp
	Mr. Sorrasak	Tinyo
	Mr. Attapong	Payamongkol
Thesis Advisor	Assoc.Prof. Fusak	Cheevasuvit
Year	2012	

ABSTRACT

This project presents an application of radio helicopter mounted webcam camera. The camera captures an image, then the presented object position data will be calculated by compute using image processing and detected object location. Microtroller (PIC) is used to control and send radio signal to digital potentiometer for controlling the helicopter to track the object.



กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงด้วยดีเพราะได้รับความเมตตาและสนับสนุนจาก รศ.ดร. พุศศักดิ์ ชิวสุวิทย์ ที่ได้ให้คำแนะนำและข้อเสนอแนะแก่ผู้จัดทำตลอดเวลา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆในการทำปริญญานิพนธ์ในครั้งนี้ ผู้จัดทำรู้สึกซาบซึ้งและขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้ให้คำแนะนำรวมทั้งให้สถานที่คือ ภาควิชาอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ครั้งนี้ รวมทั้งขอขอบพระคุณ คุณกุลชาติ ศิริไพศาลกุล เจ้าของร้าน ฮีบบีไทย ที่ช่วยให้คำแนะนำในส่วนของเฮลิคอปเตอร์บังคับวิทยุ และยังแนะนำเกี่ยวกับส่วนของเอาต์พุตที่ออกมาใช้ในการทำงาน

และที่ลืมเสียมิได้ ขอขอบพระคุณพระคุณพ่อแม่ และผู้มีพระคุณทุกท่านที่ช่วยสนับสนุนและคอยเป็นกำลังใจให้ตลอดการทำปริญญานิพนธ์ครั้งนี้

คณะผู้จัดทำ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 ไมโครคอนโทรลเลอร์ (PIC).....	3
2.2 การ Burn PIC18F452 จากโปรแกรม PICkit2.....	5
2.3 การใช้งานโปรแกรม MikroC for PIC.....	7
2.4 การส่งผ่านข้อมูลแบบระบบ USART.....	9
2.5 การส่งผ่านข้อมูลแบบ SPI.....	10
2.6 การส่งผ่านข้อมูลผ่าน RS-232.....	10
2.7 ตัวปรับความต้านทานแบบดิจิตอล (Digital Potentiometer).....	11
2.8 โปรแกรม Hyper Terminal.....	13
2.9 การประมวลผลภาพดิจิตอล (Digital Image Processing).....	14
2.10 ภาพดิจิตอล (Digital Image).....	15
2.11 แบบจำลองสี (Color Model).....	15
2.12 การสร้างภาพไบนารี (Binary Image).....	17
2.13 การแยกภาพ (Image Segmentation).....	17
2.14 การหาจุดศูนย์กลางของวัตถุด้วยโมเมนต์ (To Find The Centroid with Moment).....	18
บทที่ 3 วิธีดำเนินงาน.....	19
3.1 การตรวจจับและติดตามวัตถุ.....	19
3.2 ขั้นตอนวิธีการในส่วนของการควบคุมเฮลิคอปเตอร์.....	26

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การนำเสนอผลของข้อมูลและการวิเคราะห์ข้อมูล.....	28
4.1 การทดลองส่วนโปรแกรมการประมวลผลภาพ.....	28
4.2 การทดลองส่วนของการควบคุมเฮลิคอปเตอร์ด้วยแป้นพิมพ์.....	30
บทที่ 5 สรุปผลการวิจัย และข้อเสนอเชิงทฤษฎีจากการวิจัย.....	31
บรรณานุกรม.....	32
ภาคผนวก.....	33



สารบัญตาราง

ตารางที่

หน้า

2.1 ตารางแสดงรายละเอียดหัวข้อ RS-232.....11



สารบัญญภาพ

รูปที่	หน้า
2.1 แสดงโครงสร้างโดยทั่วไปของ PIC.....	3
2.2 แสดงโครงสร้างภายในของ PIC.....	3
2.3 แสดงหน้าที่ของแต่ละขาภายใน PIC.....	4
2.4 แสดงวงจรกำเนิดสัญญาณนาฬิกาจากคริสตอล.....	5
2.5 แสดงหน้าต่างแรกของโปรแกรม PICKit2.....	5
2.6 แสดงการตรวจสอบเบอร์ IC PIC จากโปรแกรม.....	6
2.7 แสดงโปรแกรม PICKit2 ตรวจสอบอุปกรณ์โดยอัตโนมัติ.....	7
2.8 แสดงการเลือก HEX ไฟล์ จากโปรแกรม PICKit2.....	7
2.9 แสดงหน้าต่างเริ่มโปรแกรม.....	8
2.10 แสดงหน้าต่าง New Project.....	8
2.11 แสดงการส่งผ่านข้อมูล Asynchronous.....	10
2.12 แสดงการเชื่อมต่อระหว่างคอมพิวเตอร์และ RS-232.....	10
2.13 แสดงลักษณะตำแหน่งต่าง ๆ ของหัวต่อ RS-232.....	11
2.14 แสดงรูปแบบการทำงานของตัวปรับความต้านทานแบบดิจิตอล.....	12
2.15 แสดง Single Port Digital Potentiometer.....	12
2.16 แสดง Dual Port Digital Potentiometer.....	13
2.17 แสดงหน้าต่างโปรแกรม Hyper Terminal.....	13
2.18 แสดงการกำหนดค่าเชื่อมต่อ.....	14
2.19 แสดงรูปลูกบาศก์ RGB.....	16
2.20 แสดงไดอะแกรมของแบบจำลองสีแบบ RGB.....	16
2.21 แสดงรูปหกเหลี่ยม HSV.....	17
2.22 แสดงไดอะแกรมของแบบจำลองสีแบบ HSV.....	17
3.1 แสดงโปรแกรมการตรวจจับวัตถุ.....	19
3.2 แสดงภาพที่รับจากกล้อง.....	20
3.3 แสดงภาพ RGB เป็นภาพ HSV.....	22
3.4 แสดงแท็บบาร์สำหรับปรับค่า.....	25
3.5 แสดงค่าตำแหน่งของวัตถุ.....	26
3.6 แสดงการเชื่อมต่อ RS-232 ผ่านคอมพิวเตอร์.....	26
4.1 แสดงการติดกล้องกับเฮลิคอปเตอร์.....	28
4.2 แสดงหลักการทำงานในการบังคับการเคลื่อนที่.....	29

บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาประดิษฐ์

การควบคุมการเคลื่อนที่ของเฮลิคอปเตอร์บังคับวิทยุขนาดเล็กนั้น สามารถกำหนดทิศทางการเคลื่อนที่และควบคุมได้จากรีโมทคอนโทรล เบื้องต้นนั้นเพื่อเป็นการประยุกต์การใช้งานของเฮลิคอปเตอร์บังคับวิทยุให้มีประโยชน์มากขึ้น ในที่นี้คือการทำให้เฮลิคอปเตอร์สามารถควบคุมการบินและการเคลื่อนที่ได้เองโดยอัตโนมัติจากการติดตามวัตถุ ซึ่งสามารถนำไปประยุกต์ใช้ช่วยในการปฏิบัติงานของเจ้าหน้าที่ในการติดตามคนร้าย หรือรถยนต์ต้องสงสัย ในภารกิจปราบปรามอาชญากรรมต่างๆ ดังนั้นการออกแบบระบบการควบคุมดังกล่าวจึงต้องมีการประยุกต์การใช้งานการประมวลผลภาพดิจิทัล (Digital Image Processing) ซึ่งเป็นการใช้ภาพถ่ายจากกล้องถ่ายภาพ หรือกล้องวิดีโอมาทำการประมวลผลบนคอมพิวเตอร์ และส่งสัญญาณการควบคุมออกไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมรีโมทคอนโทรลต่อไป แทนการใช้การตรวจวัดแบบอื่นที่ใช้ตัวเซ็นเซอร์มาทำการตรวจวัดซึ่งจะมีข้อจำกัดในการใช้งานมากกว่า

1.2 วัตถุประสงค์ของปัญญาประดิษฐ์

1. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ (PIC)
2. ศึกษาการส่งผ่านข้อมูลระบบ USART
3. ศึกษาการส่งผ่านข้อมูลผ่าน RS-232
4. ศึกษาการใช้งานตัวต้านทานปรับค่าได้แบบดิจิทัล (Potentiometer)
5. ศึกษาการใช้งาน OpenCV (Open Source Computer Vision) library ในการประมวลผลภาพด้วยคอมพิวเตอร์
6. ศึกษาวิธีการสื่อสารข้อมูลผ่านพอร์ตอนุกรม (Serial port)

1.3 ขอบเขตของปัญญาประดิษฐ์

1. สามารถบังคับเฮลิคอปเตอร์ผ่านคีย์บอร์ดคอมพิวเตอร์ แทนรีโมทคอนโทรลได้
2. สามารถให้ติดตามวัตถุโดยการส่งค่าเสมือนกดแป้นพิมพ์ได้
3. สามารถเขียนโปรแกรมโดยใช้ OpenCV library ในการหาตำแหน่งศูนย์กลางของวัตถุได้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการใช้งานไมโครคอนโทรลเลอร์ (PIC เบอร์ 18F452)
2. ศึกษาการใช้งานโปรแกรม Mikro C for PIC
3. ศึกษาการส่งผ่านข้อมูลระบบ USART
4. ศึกษาการส่งผ่านข้อมูลผ่าน RS-232
5. ศึกษาการส่งผ่านข้อมูลแบบ SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ศึกษาการใช้งานตัวต้านทานปรับค่าได้แบบดิจิตอล (Potentiometer)
7. ศึกษาการใช้งานโปรแกรม OpenCV เพื่อใช้ทำ Image Processing
8. ศึกษาการเขียนโปรแกรมเพื่อการประมวลผลภาพ



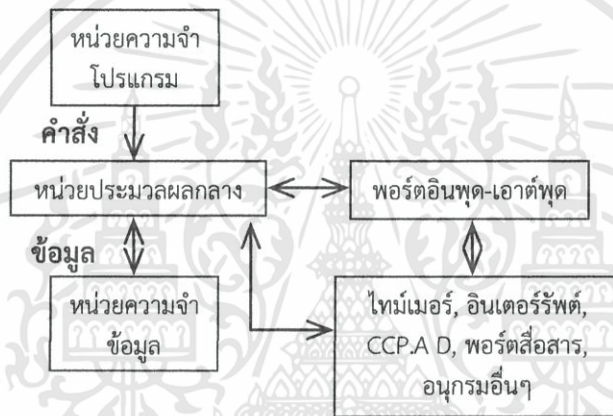
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ (PIC)

PIC เป็นไมโครคอนโทรลเลอร์ซึ่งเป็นการทำงานที่มีจำนวนชุดคำสั่งน้อย โดยแต่ละคำสั่งจะทำงานแบบง่าย ๆ ทำให้ความเร็วในการทำงานของแต่ละชุดคำสั่งสูง หนึ่งคำสั่งของ PIC มีขนาด 14 บิต ซึ่งเป็นผลให้ชุดคำสั่งของ PIC มีน้อย

2.1.1 โครงสร้างของ PIC



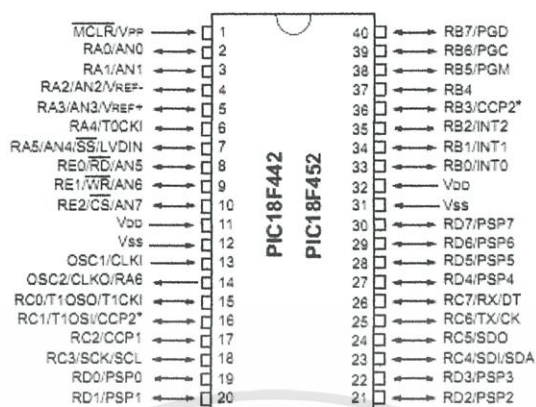
รูปที่ 2.1 โครงสร้างโดยทั่วไปของ PIC

โดย PIC เบอร์ 18F452 นี้ เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา โครงสร้างภายในประกอบด้วย ดังนี้

Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	10
RAM Bytes	1,536
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-AE/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	8 ch, 10-bit
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	2 to 5.5
Pin Count	40

รูปที่ 2.2 โครงสร้างภายในของ PIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 หน้าที่ของแต่ละขาภายใน PIC

2.1.2 โครงสร้างของหน่วยความจำ

หน่วยความจำของ PIC18F452 แบ่งออกเป็น หน่วยความจำโปรแกรม (Program memory) และหน่วยความจำข้อมูล (Data memory) ซึ่งมีรายละเอียดของหน่วย ดังต่อไปนี้

2.1.2.1 หน่วยความจำโปรแกรม

PIC ในตระกูล 18F452 จะมีโปรแกรมเคาน์เตอร์ขนาด 21 บิต ดังนั้น จึงสามารถอ้างแอดเดรสของหน่วยความจำโปรแกรมได้มากที่สุดถึง 2 Mbytes แต่เนื่องจากหน่วยความจำโปรแกรมที่มีอยู่ในชิพ PIC18F452 มีขนาดเพียงแค่ 32 Kbytes ดังนั้น จึงสามารถอ้างแอดเดรสได้ตั้งแต่ 000H จนถึง 7FFFH เท่านั้น ส่วนแอดเดรสตั้งแต่ 8000H จนถึง 1FFFFFFH จะอ่านค่าได้เท่ากับ 0 ส่วนแอดเดรสที่สงวนไว้สำหรับเวกเตอร์การอินเทอร์รัพต์ ก็คือ แอดเดรส 0008H และ 0018H

2.1.2.2 หน่วยความจำข้อมูล

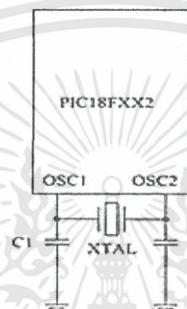
PIV18F452 มีสายสัญญาณข้อมูล (Data Bus) ขนาด 12 บิต ทำให้สามารถอ้างแอดแอดเรสของหน่วยความจำข้อมูลได้ถึง 4Kbytes ซึ่งโดยทั่วไปแล้วจะแบ่งออกเป็น 16 แบนด์ แบนด์ละ 256 Bytes แต่มีเพียงแค่ 6 แบนด์ล่างสุดเท่านั้นที่ถูกใช้งาน จึงทำให้ PIV18F452 มีหน่วยความจำข้อมูลทั้งหมด 1536 Bytes ซึ่งใช้เป็นรีจิสเตอร์ใช้งานทั่วไป ส่วนรีจิสเตอร์ฟังก์ชันพิเศษจะอยู่ภายใน 128 Bytes ของแบนด์บนสุด

2.1.3 การจ่ายไฟเลี้ยง

PIC18F452 สามารถทำงานได้ตามปกติก็ต่อเมื่อ ทำการจ่ายไฟเลี้ยงให้กับชิพที่ขา VDD และ VSS ด้วยแรงดันไฟฟ้าในช่วงตั้งแต่ 4.2 จนถึง 5.5 V ซึ่งจะทำให้ PIC สามารถทำงานที่ความเร็วสัญญาณนาฬิกาสูงสุดที่ 40 MHz แต่มีข้อจำกัดคือ ขณะที่จ่ายไฟเลี้ยง 2.0 V PIC จะสามารถทำงานที่ความเร็วสัญญาณนาฬิกาสูงสุดที่ 4 MHz เท่านั้น และจะเพิ่มขึ้นเรื่อย ๆ จนกระทั่งเท่ากับ 40 MHz ที่ไฟเลี้ยง 4.2 V ขึ้นไป โดยทั่วไปเรามักจ่ายไฟเลี้ยงให้กับ PIC ที่แรงดัน +5 V

2.1.4 สัญญาณนาฬิกา

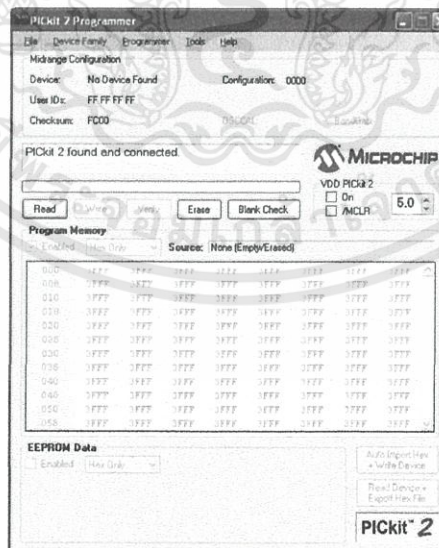
เราสามารถจ่ายสัญญาณนาฬิกาให้กับ PIC18F452 ได้ โดยจ่ายสัญญาณนาฬิกาเข้าไปที่ขา OSC1 และ OSC2 ของ PIC ซึ่งแหล่งกำเนิดสัญญาณนาฬิกานั้นมีหลายรูปแบบ โดยในการศึกษาครั้งนี้ จะใช้แหล่งกำเนิดสัญญาณนาฬิกาจากคริสตอล หรือเซรามิกรีโซเนเตอร์ โดยการใช้งานคริสตอล ออสซิลเลเตอร์นั้นจะแบ่งออกเป็น 3 โหมด โดยจะเรียงลำดับจากคริสตอลความถี่ต่ำจนถึงคริสตอลความถี่สูง คือ LP (low-power), XT, HS (high-speed) ซึ่งในการใช้งานนั้น จำเป็นที่จะต้องต่อตัวเก็บประจุ C ร่วมกับคริสตอลกำเนิดสัญญาณนาฬิกาด้วยอีก 2 ตัว โดยจ่ายสัญญาณนาฬิกาเข้าที่ขา OSC1 และ OSC2 ของ PIC ซึ่งค่าของตัว C ที่ใช้นั้นจะมีความสัมพันธ์กับค่าความถี่ของคริสตอลนั้นๆ



รูปที่ 2.4 วงจรกำเนิดสัญญาณนาฬิกาจากคริสตอล

2.2 การ Burn PIC18F452 จากโปรแกรม PICKit2

ในการสร้างโปรแกรมลง PIC นั้นจะใช้โปรแกรม PICKit2 เป็นตัวช่วยในการนำโปรแกรมที่เขียนขึ้นมาเข้าไปใส่ใน PIC โดยทำการติดตั้งโปรแกรม Pickit2 ก่อน จากนั้นก็ทำการเปิดโปรแกรมขึ้นมา



โปรแกรม PICKIT2

รูปที่ 2.5 หน้าต่างแรกของโปรแกรม PICKit2

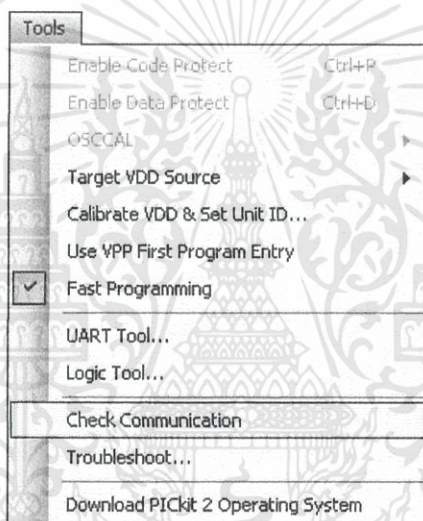
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของปุ่มต่าง ๆ

1. ปุ่ม READ อ่านข้อมูลจาก Device
2. ปุ่ม Erase ลบข้อมูลทั้งหมด
3. ปุ่ม Blank Check ตรวจสอบว่า Device วางอยู่หรือไม่
4. ปุ่ม Auto Import Hex + Write Device ทำการอ่าน Hex ไฟล์แล้วเขียนโดยอัตโนมัติ
5. ปุ่ม Read Device + Export Hex File อ่านข้อมูลจาก Device แล้วบันทึกเป็น Hex ไฟล์

อัตโนมัติ

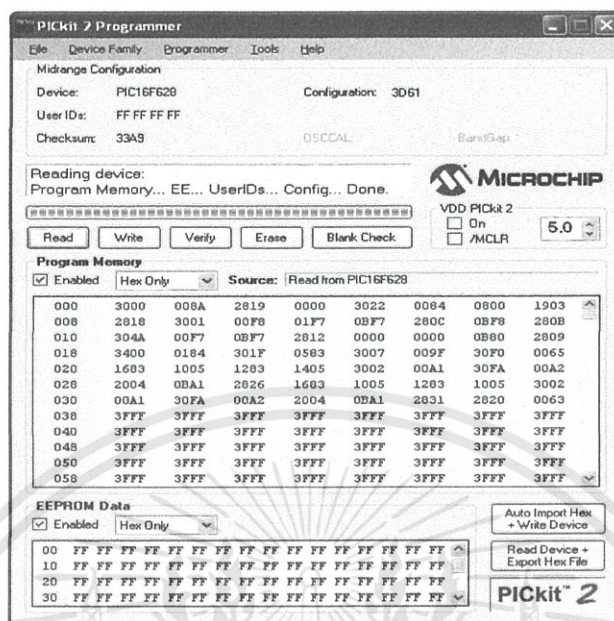
จากนั้นเมื่อตรวจพบการเชื่อมต่อระหว่างตัวบอร์ดโปรแกรม PIC กับคอมพิวเตอร์แล้ว ให้ตรวจสอบเบอร์ IC PIC จากโปรแกรม โดยคลิกที่เมนู Tools > Check Communication



เมนู Tools Check Communication

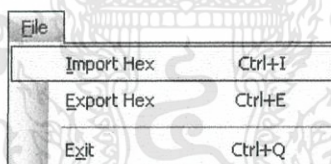
รูปที่ 2.6 การตรวจสอบเบอร์ IC PIC จากโปรแกรม

จากนั้นโปรแกรม PICkit2 จะทำการตรวจสอบอุปกรณ์ที่เชื่อมต่อไว้โดยอัตโนมัติหรือต้องการระบุ Device ตัวอย่างเช่น ต้องการโปรแกรม PIC16F628 โปรแกรม PICkit2 จะทำการตรวจสอบอุปกรณ์ที่ต่อให้โดยอัตโนมัติ



รูปที่ 2.7 โปรแกรม PICKit2 ตรวจสอบอุปกรณ์โดยอัตโนมัติ

เมื่อโปรแกรม PICKit2 ตรวจพบอุปกรณ์ (เบอร์ของ IC PIC) จากนั้นนำ HEX ไฟล์ (ที่ผ่านการคอมไพล์แล้ว) จากเมนู File เลือก Import HEX



เมนู File

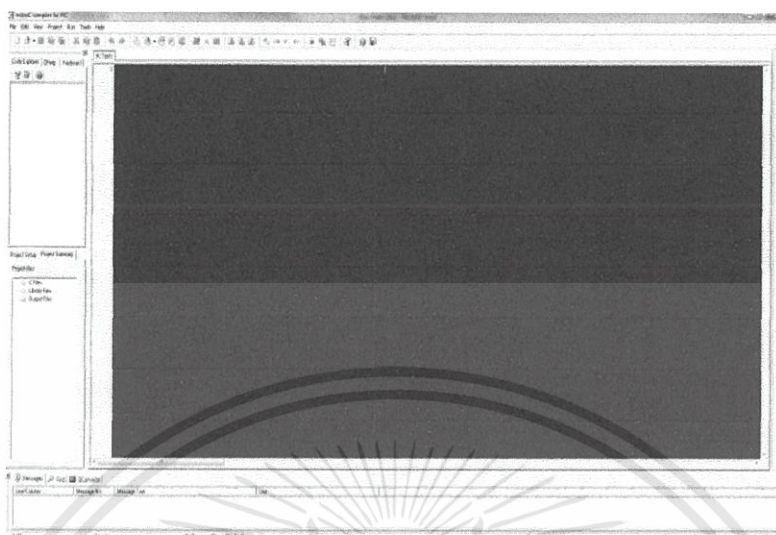
รูปที่ 2.8 เลือก HEX ไฟล์ จากโปรแกรม PICKit2

เมื่อเลือกเสร็จแล้ว ในหน้าต่างโปรแกรม PICKit2 จะบอกว่า HEX file successfully import จากนั้นก็ทำการกดปุ่ม Write โดยโปรแกรมจะทำการเขียนให้โดยอัตโนมัติ

2.3 การใช้งานโปรแกรม MikroC for PIC

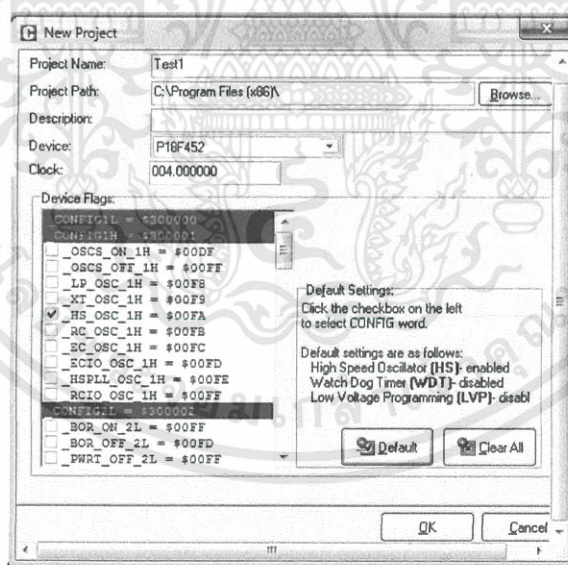
1. ทำการติดตั้งโปรแกรม โดยดาวน์โหลดไฟล์ติดตั้งโปรแกรมจาก www.mikroe.com หรือเปิดไฟล์ติดตั้งโปรแกรมจากแผ่นซีดีที่
2. เปิดใช้งานโปรแกรม mikroC for PIC ได้ โดยไปที่เมนู Start > All Programs > Mikroelektronika > mikroC for PIC หรือดับเบิลคลิกได้ที่ไอคอนของโปรแกรมที่อยู่บนหน้าจอหลักก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 หน้าต่างเริ่มโปรแกรม

3. ทำการสร้างโปรเจกใหม่ โดยให้ไปที่เมนู Project > New Project ก็จะมีปรากฏหน้าต่าง New Project



รูปที่ 2.10 หน้าต่าง New Project

Project Name คือ ชื่อของโปรเจกที่เราจะสร้าง ในที่นี้ชื่อไฟล์ Test1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Path	คือ	ไดเรกทอรีที่ใช้ในการเก็บไฟล์โปรเจค ซึ่งเราสามารถคลิก Browse เพื่อทำการหาตำแหน่งที่ต้องการเก็บไฟล์โปรเจคได้
Description	คือ	ข้อความหรือรายละเอียดเพิ่มเติม เกี่ยวกับโปรเจค ซึ่งเราจะใส่หรือไม่ใส่ก็ได้
Device	คือ	เบอร์ของ PIC ที่จะใช้ในโปรเจคนี้ โดยเลือกใช้ PIC18F452
Clock	คือ	ความถี่ของสัญญาณนาฬิกาที่เราใช้กับ PIC โดยในที่นี้เราเลือกใช้ สัญญาณนาฬิกาที่ 4.000 MHz

4. เมื่อทำการกรอกรายละเอียดจนครบแล้ว ให้คลิกที่ปุ่ม Default และ OK ตามลำดับ ถือว่าเป็นการเสร็จสิ้นการสร้างโปรเจคใหม่ หลังจากคลิก OK แล้ว จะปรากฏหน้าต่าง Code Editor ที่ว่างเปล่า โดยมีชื่อของไฟล์โค้ดโปรแกรมภาษาซี (.C) เป็นชื่อเดียวกับชื่อโปรเจคที่ตั้งไว้ พร้อมสำหรับเขียนโค้ดโปรแกรม

5. เมื่อทำการเขียนโค้ดโปรแกรมเสร็จแล้ว กดเซฟ เพื่อทำการเก็บข้อมูล จากนั้นคลิกที่ Project > Build Project เมื่อคอมไพล์โปรแกรมที่หน้าต่างด้านล่างสุดจะมีข้อความบอกว่า Success แต่ถ้าหากมี Error ก็จะมีขึ้นบอกว่าบรรทัดไหนผิด ก็ต้องกลับไปแก้ไขใหม่อีกรอบหนึ่ง โดยทุกครั้งที่มีการคอมไพล์โปรแกรมนั้น mikroC จะทำการเซฟไฟล์โค้ดของโปรแกรมให้ทุกครั้ง โดย mikroC จะสร้างเป็นไฟล์ .hex ออกมา ซึ่งสามารถนำไปทำการโปรแกรมกับ PIC ต่อไปได้

2.4 การส่งผ่านข้อมูลแบบระบบ USART

Usart ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter หมายถึง อุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัส (Asynchronous) เป็นส่วนหนึ่งในการสื่อสารแบบอนุกรมแบบ Asynchronous การสื่อสารแบบอนุกรมจะแบ่งเป็น 2 รูปแบบ คือ

2.4.1 การสื่อสารผ่านพอร์ตอนุกรมแบบ Synchronous

เป็นรูปแบบที่ใช้วิธีการส่งข้อมูลโดยใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะการรับส่งข้อมูล การรับส่งข้อมูลแบบนี้เป็นการรับส่งที่ค่อนข้างจะมีประสิทธิภาพ มีโอกาสที่ข้อมูลจะสูญหายระหว่างการส่งน้อย แต่มีข้อเสียคือ เป็นการสื่อสารแบบ Half Duplex คือ ตัวส่งและตัวรับไม่สามารถจะสื่อสารกันได้ในเวลาเดียวกัน

2.4.2 การสื่อสารผ่านพอร์ตอนุกรมแบบ Asynchronous

เป็นการส่งข้อมูลที่ไม่ต้องใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะการส่งข้อมูล แต่ใช้วิธีการกำหนดรูปแบบ Format การรับส่งข้อมูลขึ้นมาแทน และอาศัยการกำหนดความเร็วของการรับส่งที่เท่ากันทั้งฝั่งส่งและฝั่งรับ ข้อดีของการสื่อสารแบบ Asynchronous นั่นคือ สามารถสื่อสารแบบ Full Duplex คือสามารถส่งและรับข้อมูลได้ในเวลาเดียวกัน แต่การส่งแบบนี้มีโอกาสที่ข้อมูลจะสูญหายขณะรับส่งข้อมูลได้มากกว่า หรือมีความผิดพลาดได้ง่ายกว่าแบบ Synchronous



รูปที่ 2.11 Asynchronous

2.5 การส่งผ่านข้อมูลแบบ SPI

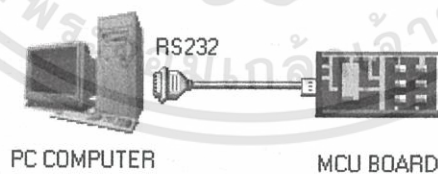
SPI ย่อมาจาก Serial Peripheral Interface เป็นวิธีการสื่อสารรูปแบบหนึ่งที่ใช้ในการสื่อสารกับอุปกรณ์ ตัวอย่างเช่น พรีนเตอร์ กล้องถ่ายรูป เครื่องสแกนเนอร์ และอื่น ๆ อีกมากมาย ถึงแม้ว่าการสื่อสารของ USB ที่มีฟังก์ชันการทำงานที่มากกว่า แต่การสื่อสารในรูปแบบ SPI ก็ยังถูกใช้งานกันอย่างแพร่หลาย

SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น Master และขณะที่อีกตัวหนึ่งทำหน้าที่เป็น Slave และส่งข้อมูลในโหมด Full Duplex ซึ่งหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง Master และ Slave ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ไม่ได้มีมาตรฐานที่กำหนดตายตัว ว่าข้อมูลที่ส่งหากันนั้นจะอยู่ในรูปแบบ หรือว่า format แบบไหน แต่เป็นการคิด Protocol ในการสื่อสารกันเอง โดยในโหมด SPI นี้จะใช้สายอินพุต/เอาต์พุต ทั้งหมด 4 เส้น ประกอบไปด้วย

- SDO (Serial Data Output) เป็นสายในการใช้ส่งข้อมูล
- SDI (Serial Data Input) เป็นสายในการใช้รับข้อมูล
- SCK (Serial Clock) เป็นสายเชื่อมโยงสัญญาณนาฬิกา เพื่อทำการ Synchronize
- SS (Slave Select) ใช้ในการเลือกอุปกรณ์ที่ต้องการและมีการต่อกับ Master

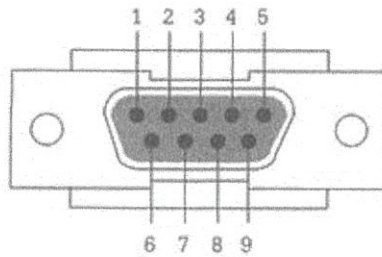
2.6 การส่งผ่านข้อมูลผ่าน RS-232

เป็นการส่งข้อมูลจากคอมพิวเตอร์โดยใช้พอร์ตอนุกรม RS-232 รับส่งข้อมูลผ่านไมโครคอนโทรลเลอร์ ซึ่งโดยปกติพอร์ตอนุกรม RS-232 จะสามารถต่อสายได้ยาว 50 ฟุต โดยประมาณ ขึ้นอยู่กับชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน



รูปที่ 2.12 การเชื่อมต่อระหว่างคอมพิวเตอร์ และ RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ลักษณะตำแหน่งต่าง ๆ ของหัวต่อ RS-232

จากรูป 2.13 จะสามารถอธิบายขาทั้ง 9 ขาของหัวต่อ RS-232 ได้ดังตาราง ต่อไปนี้

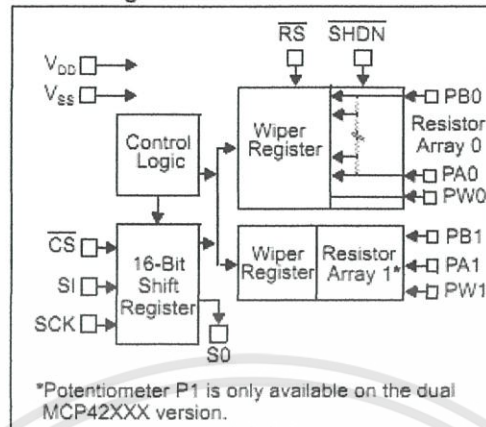
ตารางที่ 2.1 หัวต่อ RS-232

Pin Description	Type
1. Data Carrier Detect (DCD)	Input
2. Receive Data (RXD)	Input
3. Transmitted Data (TXD)	Output
4. Data Terminal Ready (DTR)	Output
5. Signal Ground (GND)	Input
6. Data Set Ready (DSR)	Input
7. Request To Send (RTS)	Output
8. Clear To Send (CTS)	Input
9. Ring Indicator (RI)	Input

2.7 ตัวปรับความต้านทานแบบดิจิตอล (Digital Potentiometer)

ตัวปรับความต้านทานแบบดิจิตอล คือ อุปกรณ์ที่ใช้สัญญาณทางไฟฟ้าสั่งเข้ามา เพื่อปรับค่าความต้านทานภายใน โดยสามารถนำไปต่อใช้กับอุปกรณ์ต่าง ๆ ได้ และสามารถปรับได้ถึง 256 ระดับ และใช้สัญญาณ SPI จาก PIC เป็นตัวควบคุมการปรับค่า

Block Diagram



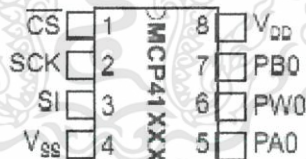
รูปที่ 2.14 รูปแบบการทำงานของตัวปรับความต้านทานแบบดิจิทัล

สำหรับตัวปรับค่าความต้านทานแบบดิจิทัลที่เลือกมาใช้นี้ คือ ตัวปรับค่าความต้านทานแบบดิจิทัลชนิด MCP42XXX ซึ่งมีอยู่ 2 รูปแบบ คือแบบ single port และ dual port

2.7.1 แบบ single port

หมายถึงภายในตัว MCP42XXX จะมีตัวปรับความต้านทานแบบดิจิทัลอยู่ภายใน 1 ตัว และไม่สามารถใช้ร่วมกันหลาย ๆ ตัวได้

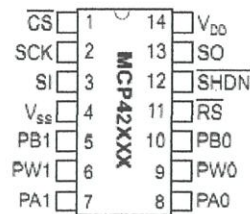
PDIP/SOIC



รูปที่ 2.15 Single Port Digital Potentiometer

2.7.2 แบบ dual port

หมายถึงภายในตัว MCP42XXX จะมีตัวปรับความต้านทานแบบดิจิทัลอยู่ภายใน 2 ตัว และสามารถใช้ร่วมกันได้หลาย ๆ ตัว

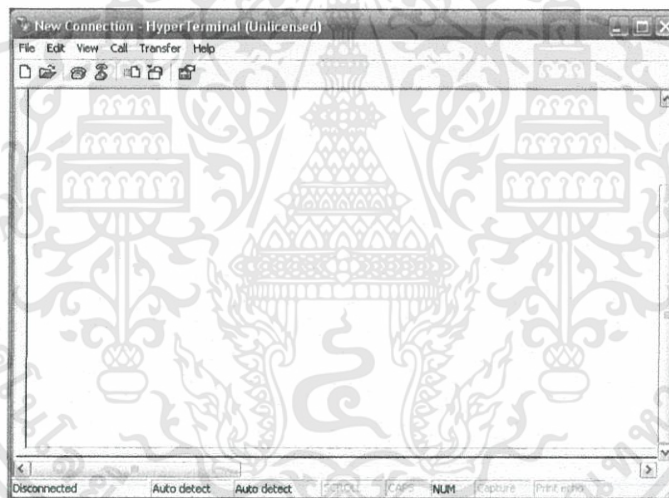


รูปที่ 2.16 Dual Port Digital Potentiometer

2.8 โปรแกรม Hyper Terminal

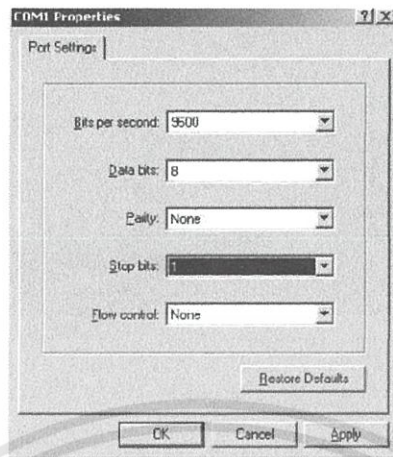
โปรแกรมไฮเปอร์ เทอร์มินอล เป็นโปรแกรมที่ใช้สำหรับรับสัญญาณจากอุปกรณ์ต่าง ๆ ที่ได้ทำขึ้นเพื่อติดต่อกับคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

ในการเปิดใช้งานโปรแกรมนั้น เมื่อเปิดขึ้นมาจะต้องมีการเซตค่าต่าง ๆ ในช่อง Bits per second โดยจะเลือกเป็น 9600 ซึ่งเท่ากับความเร็วในการสื่อสารข้อมูลที่ใช้ นั่นก็คือ 9600 Bits per second



รูปที่ 2.17 หน้าต่างโปรแกรม Hyper Terminal

ถ้าโปรแกรม Hyper Terminal ได้ทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์ (PIC) ผ่านสัญญาณสื่อสารข้อมูลแบบอนุกรม USART เป็นที่เรียบร้อยแล้ว เมื่อจ่ายไฟเลี้ยงให้กับวงจร ข้อความที่กำหนดไว้ในโปรแกรมก็จะแสดงทันที โดยที่ข้อมูลนั้นจะส่งผ่าน PIC และผ่านสายสัญญาณมาแสดงผลบนโปรแกรม Hyper Terminal โดยจะสามารถรับค่าและป้อนค่าต่าง ๆ ได้ตามโปรแกรม



รูปที่ 2.18 แสดงการกำหนดค่าเชื่อมต่อ

2.9 การประมวลผลภาพดิจิทัล (Digital Image Processing)

โดยทั่วไปแล้ว คำว่า Digital Image Processing นั้นหมายถึง การประมวลผลภาพดิจิทัลสองมิติโดยใช้คอมพิวเตอร์ หรืออาจจะหมายถึงการประมวลผลทางดิจิทัลของข้อมูลสองมิติใด ๆ โดยทั่วไปสามารถแบ่งออกเป็นประเภทใหญ่ ๆ ได้สองประเภทคือ การประมวลผลภาพระดับต่ำ (Low-Level Image Processing) และการประมวลผลภาพระดับสูง (High-Level Image Processing) ซึ่งอาจจะสามารถอธิบายถึงความแตกต่างของการประมวลผลทั้งสองแบบได้ ดังนี้

2.9.1 การประมวลผลภาพในระดับต่ำ

จะเป็นการประมวลผลทางตัวเลขเกือบทั้งหมด เพื่อให้ทราบถึงตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ โดยจะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลในระดับสูงต่อไป โดยทั่วไปการประมวลผลในระดับต่ำ จะประกอบไปด้วย การหาขอบเขตของวัตถุ การกำจัดสัญญาณรบกวน ความคมชัดของภาพ เป็นต้น

2.9.2 การประมวลผลภาพในระดับสูง

จะเป็นการนำผลลัพธ์หรือตัวแปรที่ได้จากการประมวลผลในระดับต่ำ มาตีความหรือประมวลผลอีกที เพื่อให้คอมพิวเตอร์รู้จักและเข้าใจภาพมากยิ่งขึ้น ซึ่งความแตกต่างระหว่างการประมวลผลทั้งสองประเภทนี้ คือ การประมวลผลภาพในระดับต่ำนั้นจะใช้ค่าหรือประมวลผลในระดับพิกเซล (ความสว่างของจุดภาพ) ส่วนการประมวลผลภาพในระดับสูงจะประมวลผลในรูปของสัญลักษณ์ เช่น ขนาดของวัตถุ รูปร่าง ความสัมพันธ์ระหว่างวัตถุกับภาพ

ซึ่งในโครงงานนี้อาจจะจัดว่าเป็นทั้งการประมวลผลทั้งในระดับต่ำและสูง โดยในลักษณะการประมวลผลจะทำการแยกวัตถุกับพื้นหลังของภาพ และนำผลจากการแยกวัตถุของภาพไปหาจุดศูนย์กลางของวัตถุ เพื่อทำค่าที่ได้ไปใช้ในการประมวลผลต่อไป

2.10 ภาพดิจิทัล (Digital Image)

ภาพดิจิทัลนั้น เป็นภาพที่ถูกแปลงมาจากอนาล็อก โดยภาพอนาล็อกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่าพิกเซล โดยในแต่ละพิกเซล จะถูกระบุด้วยตำแหน่งคูโคออร์ดิเนต (x, y) และค่าระดับความเข้มแสงของพิกเซลนั้น ๆ ซึ่งจะกล่าวได้ว่า ภาพดิจิทัล หมายถึง ฟังก์ชันความเข้มแสงสองมิติ $f(x, y)$ โดยที่ค่า x และ y แทนพิกัดระยะทาง (Spatial Coordinate) และค่าของ f ที่จุดใด ๆ (x, y) เป็นสัดส่วนกับความสว่างของภาพที่จุดนั้น ๆ

2.10.1 ลักษณะการเก็บข้อมูลของภาพดิจิทัล

โดยทั่วไปแล้วภาพดิจิทัลจะมีความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่นิยมใช้กันมาก คือ ค่าความเข้มของพิกเซลที่เท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าพิกเซลอยู่ในช่วง (0 - 255) โดยใช้เนื้อที่ในการเก็บข้อมูลขนาด 1 ไบท์ หรือ 8 บิต สำหรับข้อมูล 1 พิกเซล (256 ระดับ) แต่ถ้าต้องการภาพที่มีความเข้มขั้นที่สูงกว่านี้ ซึ่งต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต โดยสามารถแยกความแตกต่างของภาพแต่ละประเภทให้เห็นได้อย่างชัดเจนได้ ดังนี้

- 1) ภาพ 2 ระดับ คือ มีพิกเซลสีขาวและสีดำเท่านั้น โดยแต่ละพิกเซลจะมีขนาดข้อมูลเท่ากับ 1 บิต
- 2) ภาพ 16 ระดับ คือ ในแต่ละพิกเซลจะมีขนาดข้อมูลเท่ากับ 4 บิต ซึ่งสามารถแสดงแถบสีได้ 16 สี
- 3) ภาพ 256 ระดับ คือ ในแต่ละพิกเซลจะมีขนาดข้อมูลเท่ากับ 8 บิต ซึ่งทำให้สามารถแสดงความเข้มได้ทั้งหมด 256 ระดับ โดยในโครงนี้จะรับภาพจากกล้องเป็นภาพดิจิทัล ซึ่งมีเนื้อที่เก็บข้อมูลขนาด 8 บิต คือ มีช่วงของค่าพิกเซลทั้งหมด 256 ระดับ

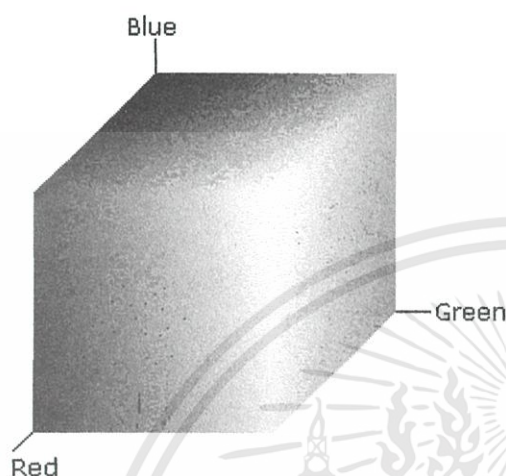
2.11 แบบจำลองสี (Color Model)

แบบจำลองสี (Color model) เป็นสิ่งสำคัญที่ใช้ในการอ้างอิงถึงสีต่าง ๆ สำหรับในคอมพิวเตอร์ เราจะไม่ใช่แบบจำลองที่เป็น Analytical model เหมือนกับที่ใช้ในทางวิทยาศาสตร์ ซึ่งใช้วิธีการวัดอยู่ในรูปของพลังตลอดช่วงสเปกตรัม แต่จะเป็นแบบ Empirical model ที่ได้รับความสัมพันธ์ของค่าที่ใช้อ้างอิงกับสีใด ๆ จากการศึกษาทดลองที่มีการรับรู้ของมนุษย์มาเกี่ยวข้อง ซึ่งแบบจำลองสีมีด้วยกันหลายแบบ โดยในโครงงานนี้จะอธิบายแบบจำลองสีที่ใช้ คือ แบบจำลองสี RGB และแบบจำลองสี HSV

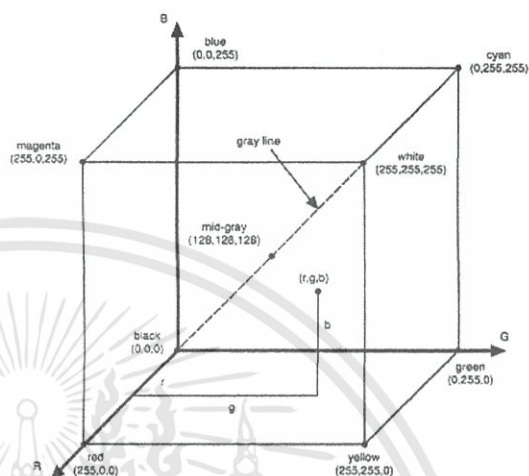
2.11.1 แบบจำลองสี RGB (RGB Color Model)

เป็นแบบจำลองสีที่ใช้เฉพาะเจาะจงกับจอคอมพิวเตอร์ เนื่องจาก RGB Model ได้ทำการสร้างสีต่าง ๆ ขึ้นโดยแหล่งกำเนิดแสงจำนวน 3 สี ได้แก่ Red (สีแดง), Blue (สีน้ำเงิน), Green (สีเขียว) ที่เกิดจากสารเรืองแสงที่มีคุณสมบัติที่แตกต่างกันตามลำดับ โดยสีทั้งสามสีจะไม่เท่ากัน ขึ้นอยู่กับอุปกรณ์ และ RGB อยู่ในระบบพิกัดคาทีเซียน (Cartesian Coordinate System) ซึ่งสามารถแสดงด้วย

แบบจำลองลูกบาศก์ RGB จากทฤษฎีการแทนที่ของ Young ค่าของสีแทนด้วยค่าพิกัดใน RGB space โดยที่จุดยอดของลูกบาศก์คือแม่สี (Red Blue และ Green) และที่ของแสงขั้นที่สอง (Secondary - Cyan Magenta และ Yellow)



รูปที่ 2.19 แสดงรูปลูกบาศก์ RGB



รูปที่ 2.20 ไดอะแกรมของแบบจำลองสีแบบ RGB

2.11.2 แบบจำลองสี HSV (HSV Color Model)

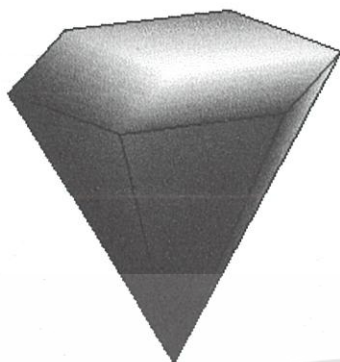
เป็นแบบจำลองหนึ่งในแบบจำลองสี ที่เปรียบเสมือนลูกใช้โดยคน เพื่อเป็นทางเลือก ซึ่ง HSV คือ ระบบโคออดิเนตทรงกระบอก (Cylindrical Coordinate System) โดยเมื่อกล่าวถึงสีต่าง ๆ ในเชิงศิลปะ เช่น เมื่อกล่าวถึงสีเขียวในทางศิลปะจะมีความแตกต่างกัน เมื่อพิจารณาสีเขียวอ่อน สีเขียวแก่ ๆ แต่ละสีนั้นมีความแตกต่างกัน ก็จะพบว่าทุกสีนั้น ก็คือสีเขียวนั่นเอง ที่ต่างกันตรงระดับความเข้ม หรือความอิ่มตัว ดังนั้น สีในแบบจำลองสี HSV จึงให้ความรู้สึกที่เข้าใจมากกว่าสำหรับมนุษย์ ซึ่งประกอบด้วยกันอยู่ 3 ส่วน คือ

H หมายถึง Hue หรือสีที่แตกต่างไปตามความถี่ของแสง เช่น น้ำเงิน เขียว เหลือง ส้ม แดง เป็นต้น

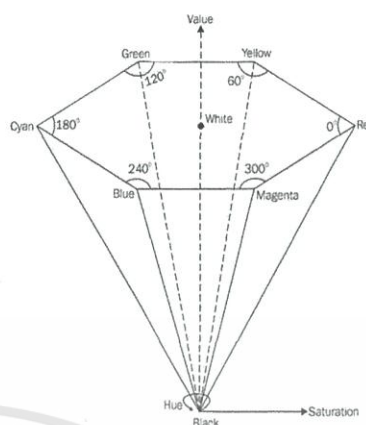
S หมายถึง Saturation หรือความอิ่มตัวของ Hue นั้น เช่น สีแดง และสีส้ม ก็คือสีแดง ซึ่งเพียงแต่ว่าสีส้มมีความอิ่มตัวน้อยกว่า

V หมายถึง Value หรือค่าความสว่างของสี โดยที่ค่า Value ต่ำที่สุดคือ สีดำ ไม่ว่าจะ Hue หรือ Saturation เท่าไรก็ตาม และค่า Value สูงที่สุดคือ สีขาว ซึ่งเป็นค่าที่สว่างที่สุดของ Hue หรือ Saturation นั้น ๆ

โดยในโครงการนี้ เรารับภาพจากกล้องเป็นแบบจำลองสีแบบ RGB และแบบจำลองสีแบบ HSV จะถูกนำไปใช้ในขั้นตอนการแบ่งวัตถุกับพื้นหลังด้วยการแยกสี



รูปที่ 2.21 รูปหกเหลี่ยม HSV



รูปที่ 2.22 ไดอะแกรมของแบบจำลองสีแบบ HSV

2.12 การสร้างภาพไบนารี (Binary Image)

การสร้างภาพไบนารีนั้นสามารถทำได้ โดยใช้เทคนิคการทำเทรชโฮล (Threshold Technique) โดยพิจารณาว่าพิกเซลใดเป็นสีขาว พิกเซลใดเป็นสีดำ และจะกระทำโดยการเปรียบเทียบระหว่างพิกเซลของภาพเริ่มต้นกับค่าคงที่หนึ่งๆที่เรียกว่า “ค่าเทรชโฮล” โดยในเทคนิคนี้จะใช้กันมากในกรณีที่ข้อมูลของภาพนั้นมีลักษณะที่แตกต่างกัน ระหว่างวัตถุ (Object) และพื้นหลัง (Background) ซึ่งค่าพิกเซลของภาพใด ๆ ที่มีค่ามากกว่าหรือเท่ากับค่าเทรชโฮล จะถูกเปลี่ยนเป็นสีดำ หรือจุดดำ

ในการสร้างภาพไบนารีโดยใช้เทคนิคการทำเทรชโฮล เพื่อให้ได้ผลลัพธ์ที่เหมาะสมและมีความคมชัด สิ่งที่สำคัญที่สุด คือ ค่าเทรชโฮล เนื่องจากถ้าค่าเทรชโฮลไม่มีความเหมาะสม (ค่าเทรชโฮลที่มีค่ามากเกินไปหรือน้อยเกินไป) ภาพที่ได้ออกมาอาจจะสว่างเกินไป หรือมืดเกินไป หรืออาจจะเกิด noise (สิ่งรบกวน) เกิดขึ้น อันเป็นผลที่ทำให้ภาพไบนารีที่ได้ออกมาไม่สวยงาม ดังนั้น ปัญหาการสร้างภาพไบนารีคือ วิธีกำหนดค่าเทรชโฮลให้เหมาะสม สำหรับภาพที่ต้องการจะนำมาทำภาพไบนารี ซึ่งมีวิธีที่ต่างกันไป โดยในโครงการนี้ได้ใช้วิธีการหาค่าเทรชโฮล คือ การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) ซึ่ง การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) นั้น เป็นการหาค่าเทรชโฮลโดยอัตโนมัติ โดยไม่ต้องให้ผู้ใช้เป็นผู้กำหนด โดยการหาค่าเทรชโฮลวิธีนี้ เป็นการหาค่าเทรชโฮลด้วยกระบวนการทางสถิติในเรื่องการค่าเฉลี่ย หรือค่ากลาง (Mean) คือ ค่าเทรชโฮลที่คำนวณได้จากค่ากึ่งกลางที่อยู่ระหว่างค่าระดับความเข้มสูงสุด (Maximum Level) กับค่าระดับความเข้มต่ำสุด (Minimum Level) ของภาพ เมื่อทำการคำนวณได้แล้วก็สามารถสร้างภาพไบนารีได้

2.13 การแยกภาพ (Image Segmentation)

การแยกภาพนั้น จะสามารถแยกภาพออกเป็นส่วน ๆ โดยสามารถแยกภาพส่วนหนึ่งจากส่วนที่ไม่ต้องการได้ ซึ่งทั่วไปแล้วสามารถแบ่งการแยกภาพออกเป็นส่วน ๆ ได้เป็น 3 กลุ่ม คือ Amplitude Segmentation Method Region Segmentation Method และ Edge Segmentation Method โดยจะกล่าวถึง Amplitude Segmentation Method เป็นหลัก

การแยกภาพแบบ Amplitude Segmentation Method เป็นวิธีการการแยกองค์ประกอบของภาพ โดยดูจากความเหมือนกันของคุณสมบัติพิกเซลภายในพื้นที่เพียงอย่างเดียว ซึ่งข้อดีของวิธีการนี้ คือ มีขั้นตอนในการทำงานไม่ยุ่งยาก และไม่ซับซ้อน ทำงานได้รวดเร็ว แต่ข้อเสีย คือ ไม่สามารถใช้กับภาพที่มีสัญญาณรบกวนมาก หรือภาพที่มีความสว่างไม่สม่ำเสมอ

โดยในโครงการนี้ ใช้วิธีแบบ Amplitude Segmentation Method ในการแบ่งแยกภาพ โดยใช้สี คือ แยกภาพของวัตถุออกจากพื้นหลัง โดยวัตถุที่ต้องการกำหนดให้เป็นสีขาว และพื้นหลังทั้งหมดกำหนดให้เป็นสีดำ ซึ่งเป็นการแปลงภาพแบบไบนารี

2.14 การหาจุดศูนย์กลางของวัตถุด้วยโมเมนต์ (To Find The Centroid with Moment)

2.14.1 นิยามของโมเมนต์

โมเมนต์ หรือ Moment เป็นนิยามของภาพ $f(x, y)$ คือ

$$M_n(m, n) = \sum_{x=1}^J \sum_{y=1}^K (x)^m (y)^n F(x, y)$$

ถ้าพิจารณา Zero-Order Moment เราจะได้ว่า Zero-Order Moment คือ ผลรวมของค่าพิกเซลของภาพ ถ้าภาพเป็นภาพขาวดำ (Binary) Zero-Order Moment คือ พื้นที่ First-Order row Moment คือ

$$M(1,0) = \frac{1}{K} \sum_{x=1}^J \sum_{y=1}^K x F(x, y)$$

และ First-Order Column Moment คือ

$$M(0,1) = \frac{1}{J} \sum_{x=1}^J \sum_{y=1}^K y F(x, y)$$

2.14.2 Centroid ของภาพ

จากอัตราส่วนของ First-Order Moment กับ Zero-Order Moment จะถูกนิยามว่าเป็น Centroid ของภาพ ซึ่ง Centroid คือ จุดศูนย์กลางของแรงโน้มถ่วง (Center of gravity) ของภาพ $f(x, y)$ หรือจุดซึ่งมวลของ $f(x, y)$ ที่อยู่เหนือ ได้ ซ้าย ขวาของภาพที่ค่าเท่ากัน คือ

$$x = M(1,0)/M(0,0) \quad \text{และ} \quad y = M(0,1)/M(0,0)$$

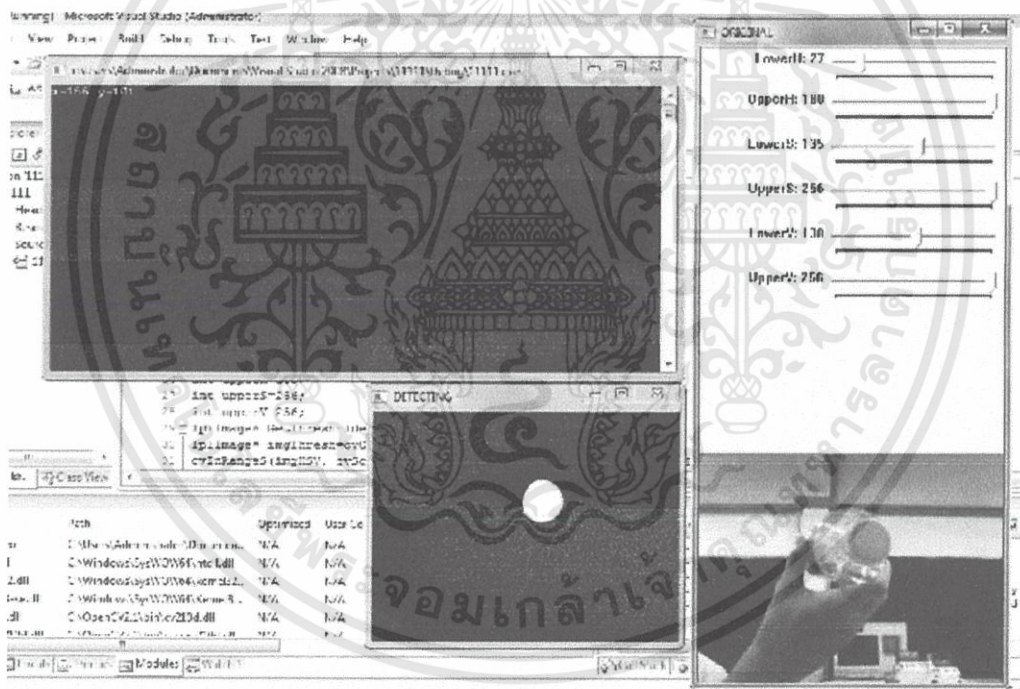
บทที่ 3

วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการดำเนินงาน และการปฏิบัติงานในการทำโครงงาน ทั้งในส่วนของการตรวจจับและติดตามวัตถุ และการติดต่อระหว่างคอมพิวเตอร์และรีโมทคอนโทรล เพื่อควบคุมเฮลิคอปเตอร์

3.1 การตรวจจับและติดตามวัตถุ

การประมวลผลภาพ (Image Processing) เพื่อทำการการตรวจจับและติดตามวัตถุนั้น ได้ใช้ Intel OpenCV Library ซึ่งเป็นไลบรารีสำหรับงานด้านคอมพิวเตอร์วิชั่น (Computer Vision) ซึ่งพัฒนาขึ้นโดยบริษัท Intel และเผยแพร่ให้ใช้งานได้ฟรี ใช้ในการประมวลผลสัญญาณภาพที่รับเข้ามาจากกล้อง และนำผลที่ได้จากกระบวนการไปทำการควบคุมเฮลิคอปเตอร์ โดยผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องคอมพิวเตอร์อีกต่อหนึ่ง โดยได้มีการใช้งานในการเขียนโปรแกรม ดังนี้



รูปที่ 3.1 โปรแกรมการตรวจจับวัตถุ

3.1.1 การรับภาพจากกล้อง

ในการทำโครงงานนี้ได้ใช้กล้องเว็บแคม (Web CAM) ในการรับภาพเข้าสู่คอมพิวเตอร์ เพื่อทำการประมวลผล คำสั่งที่ใช้ในการรับภาพจากกล้องเว็บแคม คือ cvCapture (เป็นคำสั่งของ OpenCVLibrary) โดยภาพที่รับเข้ามานี้จะถูกจัดเก็บในรูปแบบสี RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 3.1 การรับภาพจากกล้อง

```

#include<cv.h>
#include<highgui.h>
int main()
{
    CvCapture* capture = cvCaptureFromCAM(0);
    if(!capture)return 1;
    cvNamedWindow("Webcam");
    while(1){
        IplImage* frame = cvQueryFrame(capture);
        if(!frame)break;
        cvShowImage("Webcam", frame);
        int c = cvWaitKey(40);
        if((char)c==27)break;
    }
    cvDestroyWindow("Webcam");
    cvReleaseCapture(&capture);
    return 0;
}

```



รูปที่ 3.2 ภาพที่รับจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การทำงานของโปรแกรม

หลังจากทำการติดต่อกับกล้องเพื่อทำการรับภาพแล้ว โปรแกรมก็จะเริ่มกระบวนการในการประมวลผลภาพ เพื่อทำการตรวจจับและหาตำแหน่งวัตถุ เพื่อทำการติดตามต่อไป ซึ่งสามารถแบ่งกระบวนการทำงานออกเป็นส่วนต่าง ๆ ได้ดังนี้

การแปลงระบบภาพ RGB (Red, Green, Blue) เป็น HSV (Hue, Saturation, Value Space) โปรแกรมจะทำการแปลงภาพในระบบ RGB ให้เป็นภาพในระบบ HSV เพื่อใช้ในกระบวนการการแบ่งส่วนภาพต่อไป เนื่องจากภาพในระบบ HSV นั้นจะมีความเหมาะสมในการนำมาใช้งานในการทำโครงงานนี้มากกว่าภาพในระบบ RGB เนื่องจากจะทำให้สามารถแยกความแตกต่างระหว่างวัตถุและสิ่งแวดล้อมบริเวณพื้นหลังได้ชัดเจน คำสั่งที่ใช้ในการทำการแปลง คือ

```
cvCvtColor (src image, dst image, CV_BGR2HSV );
```

โปรแกรม 3.2 แปลงภาพ RGB เป็นภาพ HSV

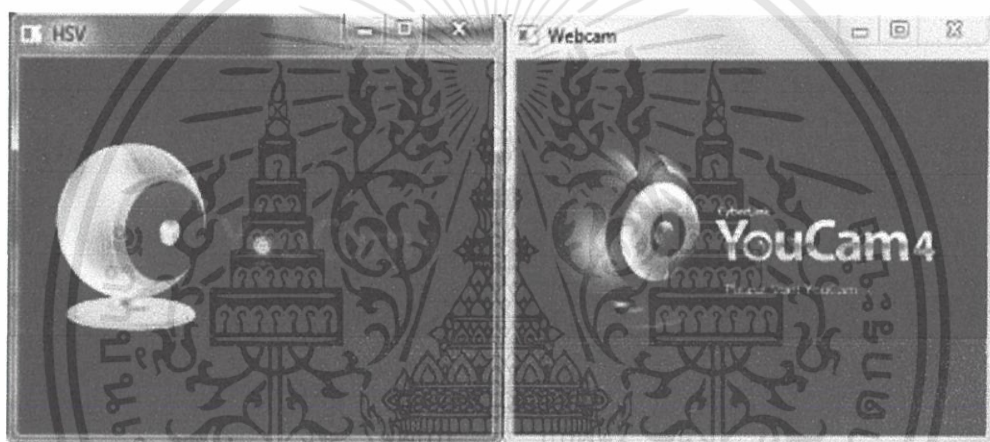
```
#include<cv.h>
#include<highgui.h>
int main ()
{
    CvCapture* capture = cvCaptureFromCAM(0) ;
    if(!capture) return 1 ;
    cvNamedWindow("Webcam") ;
    cvNamedWindow("HSV") ;
    while(1) {
        IplImage* frame = cvQueryFrame(capture) ;
        if(!frame) break ;
        frame = cvCloneImage(frame) ;
        IplImage* imgHSV = cvCreateImage(cvGetSize(frame),
        IPL_DEPTH_8U, 3) ;
        cvCvtColor(frame,imgHSV,CV_BGR2HSV) ;
        cvShowImage("Webcam", frame) ;
        cvShowImage("HSV",imgHSV) ;
        int c = cvWaitKey(40) ;
        if((char)c==27) break ;
        cvReleaseImage(&imgHSV);
        cvReleaseImage(&frame) ;
    }
}
```

โปรแกรม 3.2 (ต่อ)

```

cvDestroyWindow("Webcam");
cvDestroyWindow("HSV");
cvReleaseCapture(&capture);
return 0;
}

```



รูปที่ 3.3 ภาพ RGB เป็นภาพ HSV

3.1.3 การแบ่งส่วนภาพ (Segmentation)

หลังจากที่ได้ทำการแปลงภาพจาก RGB เป็นภาพ HSV แล้ว ก็ทำการแบ่งส่วนภาพเพื่อกำหนดขอบเขตของวัตถุที่ต้องการตรวจจับภายในบริเวณภาพ ให้สามารถเห็นความแตกต่างที่ชัดเจนเพื่อนำไปประมวลผลในขั้นต่อไป ในที่นี่ได้ใช้วิธีการในการปรับค่าเทรชโฮลด์ (Thresholding technique) ซึ่งการปรับค่าเทรชโฮลด์ของภาพ HSV นั้น ต้องทำการปรับค่าเทรชโฮลด์ด้วยกันทั้งหมด 3 ค่า ได้แก่ Hue (โทนสี), Saturation (ความเข้ม), Value (ความมืด/สว่าง) ในการทดลองนั้นจะได้ทำการสร้างแท็กบาร์ (Trackbar) เพื่อใช้ในการปรับค่าพารามิเตอร์ทั้งสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.3 การปรับค่าเทรชโฮลด์ของภาพ HSV

```

#include<cv.h>
#include<highgui.h>
int lowerH=0,lowerS=0,lowerV=0,upperH=256,upperS=256,upperV=256 ;
IplImage* GetThresholdedImage(IplImage* imgHSV) {
    IplImage*
imgThresh=cvCreateImage(cvGetSize(imgHSV),IPL_DEPTH_8U, 1);
    cvInRangeS(imgHSV,cvScalar(lowerH,lowerS,lowerV),cvScalar(upperH,
upperS
        ,upperV), imgThresh);
    return imgThresh;
}
void setwindowsetting(){
    cvNamedWindow("Webcam");
    cvNamedWindow("ThresholdHSV");
    cvNamedWindow("Trackbar");
    cvCreateTrackbar("LowerH", "Trackbar", &lowerH, 256, NULL);
    cvCreateTrackbar("UpperH", "Trackbar", &upperH, 256, NULL);
    cvCreateTrackbar("LowerS", "Trackbar", &lowerS, 256, NULL);
    cvCreateTrackbar("UpperS", "Trackbar", &upperS, 256, NULL);
    cvCreateTrackbar("LowerV", "Trackbar", &lowerV, 256, NULL);
    cvCreateTrackbar("UpperV", "Trackbar", &upperV, 256, NULL);
}
int main() {
    CvCapture* capture = cvCaptureFromCAM(0);
    if(!capture) return -1 ;
    setwindowsetting();
    while(1){
        IplImage* frame = cvQueryFrame(capture);
        if(!frame) break ;
        frame = cvCloneImage(frame);
        cvSmooth(frame, frame, CV_GAUSSIAN,3,3);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

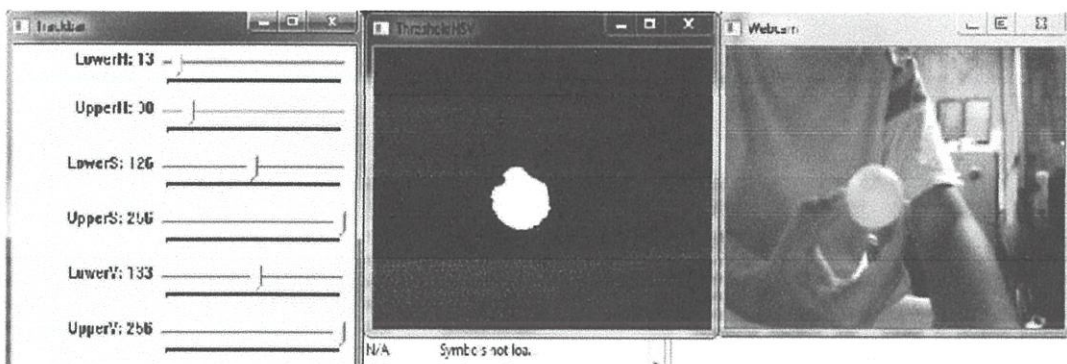
โปรแกรม 3.3 (ต่อ)

```

        IplImage* img = cvCreateImage(cvGetSize(frame),
IPL_DEPTH_8U, 1);
        IplImage* imgHSV = cvCreateImage(cvGetSize(frame),
IPL_DEPTH_8U,3);
        cvCvtColor(frame,imgHSV,CV_BGR2HSV);
        IplImage* imgThresh = GetThresholdedImage(imgHSV);
        cvSmooth(imgThresh, imgThresh, CV_GAUSSIAN,3,3);
        cvShowImage("Webcam", frame);
        cvShowImage("ThresholdHSV",imgThresh);
        cvShowImage("Trackbar",img);
        int c = cvWaitKey(40);
        if((char)c==27){
            break;
        }
        cvReleaseImage(&imgHSV);
        cvReleaseImage(&imgThresh);
        cvReleaseImage(&frame);
        cvReleaseImage(&img);
    }
    cvDestroyWindow("Webcam");
    cvDestroyWindow("ThresholdHSV");
    cvDestroyWindow("Trackbar");
    cvReleaseCapture(&capture);
    return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แท็บสำหรับปรับค่า

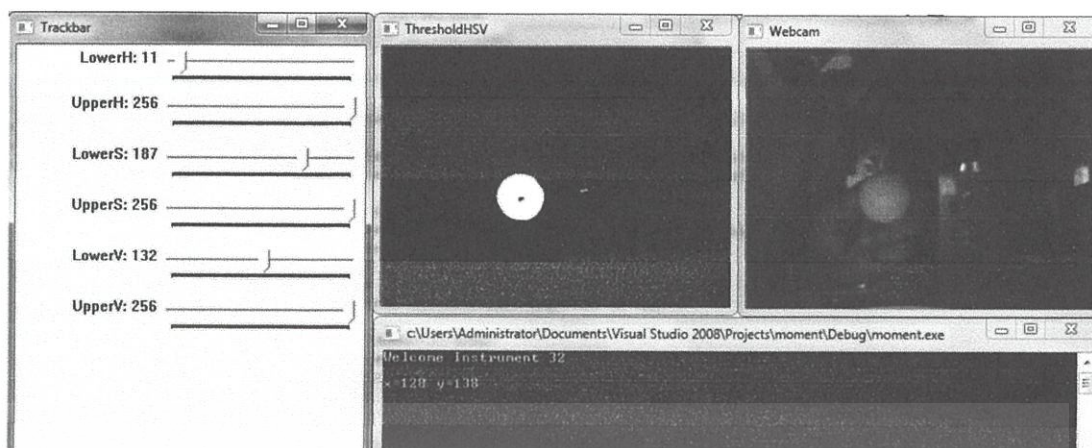
3.1.4 การจดจำวัตถุด้วยโมเมนต์ (Object Recognition with Moment)

ในส่วนนี้จะเป็นการหาตำแหน่งจุดศูนย์กลางของวัตถุ เพื่อทำการติดตามโดยในส่วนของโปรแกรมจะต้องทำการเพิ่มฟังก์ชันย่อย ซึ่งใช้ในการคำนวณจุดศูนย์กลางเข้าไปจากโปรแกรมที่ 3.3

โปรแกรมที่ 3.4 การคำนวณหาค่าโมเมนต์

```
void trackObject(IplImage* imgThresh) {
    CvMoments *moments = (CvMoments*)malloc(sizeof(CvMoments));
    cvMoments(imgThresh, moments, 1);
    double moment10 = cvGetSpatialMoment(moments, 1, 0);
    double moment01 = cvGetSpatialMoment(moments, 0, 1);
    double area = cvGetCentralMoment(moments, 0, 0);
    if(area>1000){
        int posX = moment10/area;
        int posY = moment01/area;
        printf("x=%d y=%d\n",posX,posY);
    }
    free(moments);
}
```

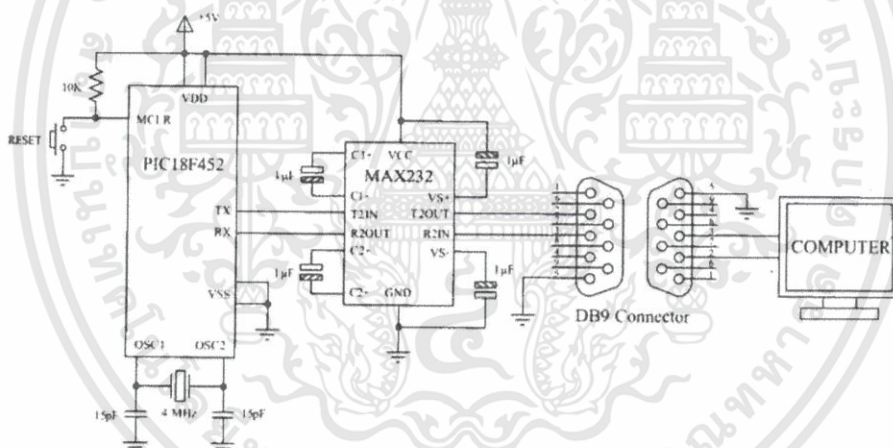
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงค่าตำแหน่งของวัตถุ

3.2 ขั้นตอนวิธีการในส่วนของการควบคุมเฮลิคอปเตอร์

1. ทำการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับคอมพิวเตอร์ โดยใช้การเชื่อมต่อแบบ USART Library function ของโปรแกรม mikro C และทำการเชื่อมต่อวงจร ดังรูป



รูปที่ 3.6 การเชื่อมต่อ RS-232 ผ่านคอมพิวเตอร์

2. เปิดโปรแกรม Hyperterminal ในคอมพิวเตอร์ จากนั้นทำการตั้งค่าในโปรแกรม Hyperterminal ตามที่ได้กล่าวไว้ในบทที่ 2 แล้วทำการทดลองรับส่งข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางโปรแกรม โดยทดลองพิมพ์อักขระและสังเกตการณ์รับข้อมูลเข้ามาที่โปรแกรม Hyperterminal โดยปกติเมื่อเชื่อมต่อกันเรียบร้อยแล้ว และทำการจ่ายไฟเลี้ยงเข้าบอร์ดทดลอง หน้าโปรแกรมจะขึ้นคำว่า “Start Program by INS32”

3. เมื่อโปรแกรมขึ้นแสดงข้อความดังกล่าวแล้ว แสดงว่าโปรแกรมจากไมโครคอนโทรลเลอร์ได้มีการติดต่อกับ RS-232 แล้ว จากนั้นก็ทำการสั่งการทำงานของเฮลิคอปเตอร์ โดยใช้การกดจากแป้นพิมพ์คอมพิวเตอร์ โดยได้แสดงฟังก์ชันการใช้งานของแต่ละปุ่ม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

W = บินขึ้น	O = เอียงหน้า
S = บินลง	L = เอียงหลัง
Q = เอียงซ้าย	[= เลี้ยวซ้าย
E = เอียงขวา] = เลี้ยวขวา
C = รีเซตสถานะ	B = ค้างสถานะของการบินให้บินที่ความสูงคงที่

4. ในการกดปุ่มคำสั่งจากแป้นพิมพ์นั้น เมื่อกดแต่ละครั้ง สามารถกำหนดค่าได้จากโปรแกรม ไมโครโปรแกรม MikroC ซึ่งจะเป็ค่าที่ทำการปรับเทียบกับรีโมตจริงแล้ว โดยการเปรียบเทียบค่าจากรีโมต จะใช้การวัดค่าความต้านทานของรีโมตแล้วทำการบันทึกผลการทดลองไว้ จากนั้นค่าที่เราสามารถปรับระดับได้ทั้งหมด 256 ค่า โดยจะเริ่มต้นที่ค่า 0-255 แล้วนำไปเปรียบเทียบกับค่าความต้านทานของตัวต้านทานปรับค่าได้ (MCP42100) ที่ 100 กิโลโอห์ม โดยค่าที่ 0 จะหมายถึงค่าความต้านทาน 100 กิโลโอห์ม และค่าที่ 255 จะหมายถึงค่าที่ 0 กิโลโอห์มนั่นเอง

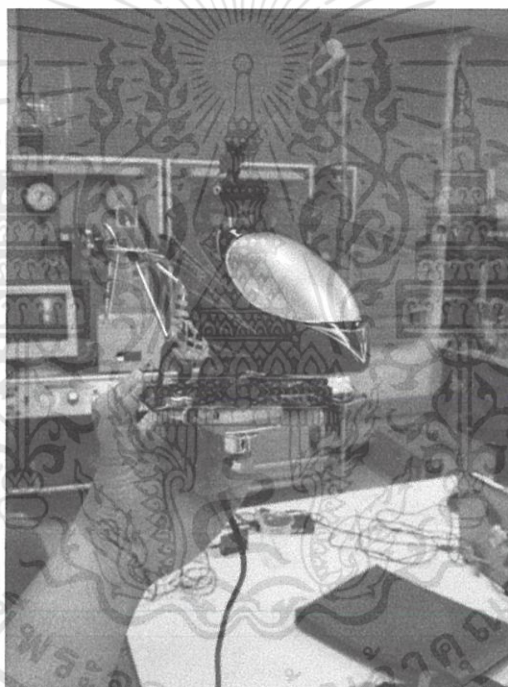


บทที่ 4

การนำเสนอผลของข้อมูลและการวิเคราะห์ข้อมูล

4.1 การทดลองส่วนโปรแกรมการประมวลผลภาพ

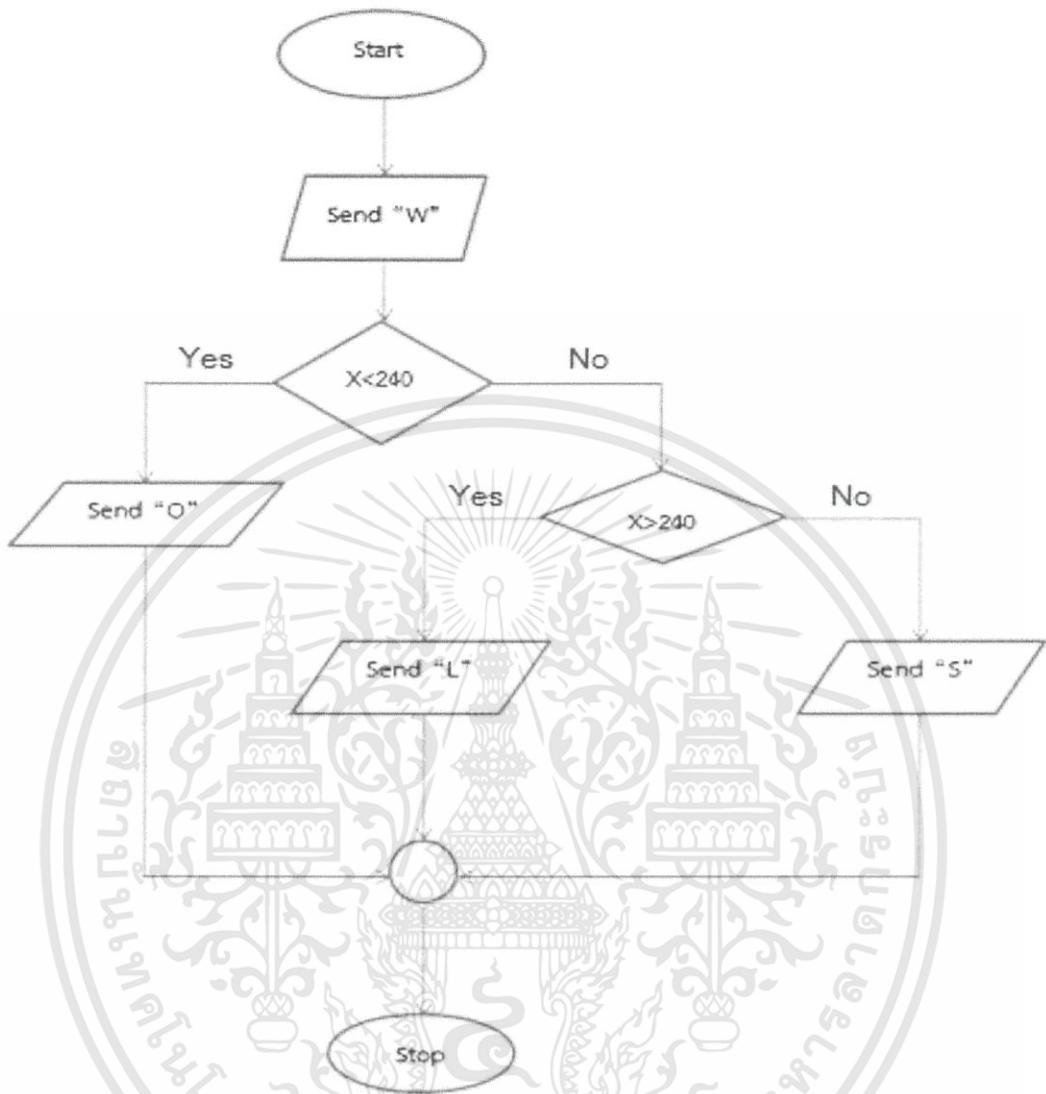
โปรแกรมสามารถทำการติดตามวัตถุเป้าหมายได้ โดยทำการรับภาพจากกล้องเว็บแคมที่ติดตั้งไว้บริเวณฐานของเฮลิคอปเตอร์บังคับที่ใช้ในการทดลอง และนำภาพที่ได้มาทำการสร้างภาพขาวดำ (Binary Image) โดยทำการปรับขอบเขตของค่าเทรชโฮลด์ (Threshold) ของพารามิเตอร์ทั้งสาม ได้แก่ H (Hue) คือ ค่าโทนสี, S (Saturation) คือ ค่าความอิ่มตัวของสี และ V (Value) คือ ค่าความสว่างของเพื่อให้สามารถทราบความแตกต่างระหว่างวัตถุที่ต้องการติดตามและพื้นหลังของวัตถุได้ โดยวัตถุที่ต้องการติดตามจะมีสีขาวบนพื้นหลังที่เป็นสีดำ จากนั้นโปรแกรมจะทำการคำนวณหาตำแหน่งพิกัดของวัตถุที่ต้องการติดตามและแสดงค่าออกมา



รูปที่ 4.1 แสดงการติดกล้องกับเฮลิคอปเตอร์

ซึ่งค่าพิกัดของวัตถุที่ได้มานั้นจะถูกนำมาทำการเปรียบเทียบ และส่งค่าเอาต์พุตไปยังพอร์ตอนุกรม เพื่อทำการบ่งการเคลื่อนที่ของเฮลิคอปเตอร์ต่อไป

ในการทดลองนี้ได้กำหนดลักษณะการเคลื่อนที่ของเฮลิคอปเตอร์ไว้ ดังแสดงในผังการทำงานรูปที่ 4.2 โดยกำหนดให้เฮลิคอปเตอร์ทำการเคลื่อนที่ติดตามวัตถุตามแนวแกน “X” ในระบบพิกัดภาพเท่านั้น กล่าวคือการเคลื่อนที่เดินหน้าและถอยหลัง



รูปที่ 4.2 ผังการทำงานในการบังคับการเคลื่อนที่

จากภาพผังการทำงานดังกล่าวเมื่อเริ่มการทำงาน โปรแกรมจะส่ง “w” ไปยังไมโครคอนโทรลเลอร์ เพื่อบังคับให้เฮลิคอปเตอร์เปิดเครื่องและทำการยกตัวให้ลอยขึ้น

จากนั้นโปรแกรมจะทำการตรวจสอบว่าเป็นไปตามเงื่อนไขการทำงานใด ถ้า X น้อยกว่า 240 (ขนาดภาพคือ 640×480) โปรแกรมจะส่ง “o” คือคำสั่งให้เฮลิคอปเตอร์เคลื่อนที่ไปด้านหน้า แต่ถ้าหาก X ไม่น้อยกว่า 240 โปรแกรมจะทำการตรวจสอบในเงื่อนไขถัดมาคือ ถ้า X มากกว่า 240 โปรแกรมจะส่ง “l” คือคำสั่งให้เฮลิคอปเตอร์เคลื่อนที่ถอยหลัง แต่ถ้าหากค่าพิกัด X ไม่เข้ากับเงื่อนไขการทำงานทั้งสอง คือ $X = 240$ ซึ่งคือจุดศูนย์กลางของภาพ ในที่นี้คือจุดศูนย์กลางของภาพตรงกับตำแหน่งจุดศูนย์กลางของวัตถุที่ทำการติดตาม โปรแกรมจะส่ง “s” คือคำสั่งให้เฮลิคอปเตอร์หยุดการทำงาน

4.2 การทดลองส่วนของการควบคุมเฮลิคอปเตอร์ด้วยแป้นพิมพ์

โดยการทดลองจะเป็นการบังคับเฮลิคอปเตอร์ด้วยคอมพิวเตอร์ โดยมีไมโครคอนโทรลเลอร์ เป็นสื่อกลางในการควบคุมรีโมตบังคับการบิน และเป็นตัวประมวลผลการบังคับแต่ละทิศทางให้ออกมาเป็นข้อมูล ซึ่งจะนำข้อมูลนี้ไปอ่านค่าและนำไปพิจารณาการบังคับที่เป็นข้อมูลที่เป็นลายลักษณ์อักษร ส่วนการส่งข้อมูลผ่านพอร์ตอนุกรม จะเป็นการส่งผ่านโหมด USART อีกรี่ ครั้งละ 8 บิต ดังนั้นการที่จะส่งข้อมูลผ่านโหมดนี้ให้ได้ข้อมูลจำนวนมากๆต่อกันจะต้องใช้คำสั่งในภาษา C ในการวนส่งข้อมูล ดังนั้นเราจึงไม่สามารถกดปุ่มแป้นพิมพ์บังคับหลายปุ่มเพื่อให้ microcontroller ประมวลผลได้พร้อมกันหลายปุ่มในเวลาเดียวกัน

ส่วนในเรื่องของการบังคับรีโมตบังคับจะเป็นการแบ่งแรงดัน เหมือนกรณีการใช้ความต้านทานปรับค่าได้ สำหรับปุ่มบังคับรีโมต ดังนั้นเราต้องใช้ IC MCP42100 ที่สามารถปรับค่าความต้านทานได้อย่างเป็นเส้นตรง (linear) เราจึงสามารถปรับความละเอียดการบังคับทุกทิศทางได้อย่างแม่นยำ และสุดท้ายการแสดงผลข้อมูลการบังคับจะถูกส่งผ่านคอมพิวเตอร์ในรูปของเลขฐานสิบ แต่ว่าการประมวลผลของ microcontroller จะส่งข้อมูลเลขฐานสิบไปเป็นค่าตำแหน่งของรหัสแอสกีมาเข้าโปรแกรม Hyperterminal ที่คอม ดังนั้นเราจึงต้องแบ่งเลขฐานสิบใน microcontroller จาก 3 หลักมาเป็น 3 ส่วน จากนั้นจึงนำค่า 30 บวกเข้าไป ซึ่งเลขนี้คือ ตำแหน่งเริ่มต้นของเลขฐานสิบของตารางแอสกี เมื่อตัวเลข 3 ชุดที่ถูกแปลงออกมาจะถูกนำไปแสดงเรียงต่อกันออกมาให้ได้เลขฐานสิบที่ต้องการ



บทที่ 5

สรุปผลการวิจัย และข้อเสนอเชิงทฤษฎีจากการวิจัย

ในส่วนของการใช้โปรแกรมประมวลผลภาพนั้น เมื่อทำการเปิดโปรแกรม visual studio แล้วทำการรันโปรแกรม จะมีการส่งค่าผ่านพอร์ตอนุกรม RS-232 ไปยังตัวไมโครคอนโทรลเลอร์ (PIC18F452) เพื่อใช้ในการประมวลผลในขั้นตอนของการควบคุมรีโมตคอนโทรล โดยการปรับค่าความต้านทานจากตัวปรับความต้านแบบดิจิตอล (MCP42100) ซึ่งเมื่อทำการรันโปรแกรมเพื่อจะจับภาพของวัตถุ โปรแกรมจะทำการส่ง “w” ออกไปเพื่อให้เฮลิคอปเตอร์ยกตัวขึ้นก่อน หลังจากนั้นก็จะพิจารณาตามเงื่อนไขที่ได้ตั้งไว้ (ค่า “x” เป็นค่าในแนวแกน x ในระบบพิกัดภาพ) ซึ่งได้อธิบายไว้ในผลการทดลองว่าจะส่งค่า “o” เดินหน้า “l” ถอยหลัง หรือ “s” หยุดการทำงาน จากการทดลองได้ทำการทดสอบว่าค่าจากโปรแกรมประมวลผลภาพสามารถส่งผ่านพอร์ตอนุกรม RS-232 ได้โดยทำการจำลองพอร์ตขึ้นมาแล้วทดสอบโดยให้ค่าไปแสดงยังโปรแกรม Hyper Terminal ได้ จากนั้นทดสอบโดยทำการส่งค่าไปยังไมโครคอนโทรลเลอร์ พบว่า ค่าที่ส่งออกไปนั้นมันเปลี่ยนแปลงแบบ Real time คือมีการเปลี่ยนแปลงค่าตลอดเวลา ซึ่งค่าที่ส่งออกไปก็จะมีการส่งไปเรื่อย ๆ ในอัตราความเร็วที่สูงพอสมควร ทำให้เกิดปัญหาที่ว่า การประมวลผลภาพ และการรับส่งข้อมูลไม่สัมพันธ์ คือพอร์ตอนุกรม RS-232 ไม่สามารถรับส่งข้อมูลจากการประมวลผลภาพได้ทันหรือช้า ทำให้เกิดการความผิดพลาดในการติดตามวัตถุต่อไป

ในโครงการครั้งนี้ ในส่วนของการควบคุมเฮลิคอปเตอร์ผ่านคอมพิวเตอร์นั้น สามารถส่งผ่านคำสั่งไปสั่งเฮลิคอปเตอร์ได้เหมือนรีโมตคอนโทรล แต่ในส่วนของการบังคับนั้นไม่สามารถบังคับได้เสถียรเหมือนรีโมตคอนโทรลจริงๆ เนื่องจากการควบคุมผ่านคอมพิวเตอร์นั้นการส่งข้อมูลผ่านพอร์ตอนุกรม สามารถส่งได้ครั้งละ 8 บิต ซึ่งหมายความว่า ไม่สามารถกดปุ่มคีย์บอร์ดได้พร้อมกันในหลายๆ ปุ่ม แต่ในส่วนของรีโมตคอนโทรลจริงๆ สามารถจะรักษาสภาพของเฮลิคอปเตอร์ในการควบคุมได้นิ่งและเสถียรมากกว่า อีกทั้งการควบคุมเฮลิคอปเตอร์นั้นต้องใช้จังหวะในการควบคุมที่ต้องผ่านการฝึกฝนมาเป็นเวลานานพอสมควรถึงจะสามารถบังคับให้ได้เป็นดังที่ต้องการได้ เฮลิคอปเตอร์บังคับวิทยุที่ใช้ก็ต้องการความนิ่งมากๆ ในการทำงานประเภทติดตามวัตถุโดยอัตโนมัติ เพราะจะสามารถสั่งจากคอมพิวเตอร์ได้แม่นยำ โดยใช้เทคโนโลยีใหม่ๆ ในการสร้างสมดุลงของเฮลิคอปเตอร์ โดยอาจใช้เฮลิคอปเตอร์ 4 ใบพัด โดยบังคับผ่าน Wi-fi หรือตรวจจับตำแหน่งจากดาวเทียมหรือ GPS ได้

บรรณานุกรม

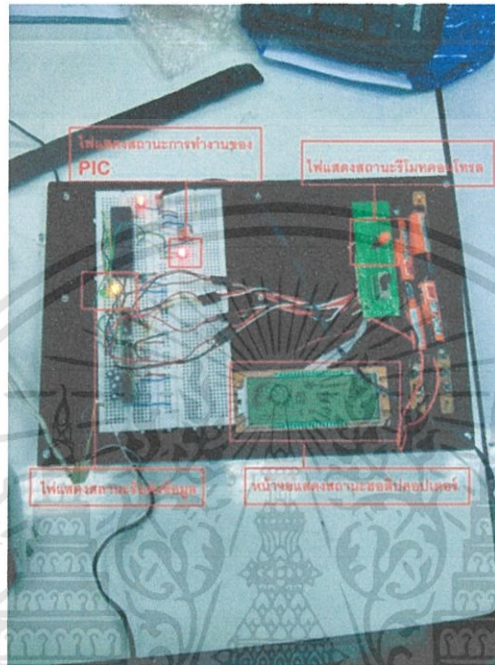
1. ชูชาติ ปิณฑวิรุจน์. “การประมวลผลภาพดิจิทัลด้วย C++ Digital image processing with C++”. พิมพ์ครั้งที่ 1. กรุงเทพฯ : ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2552.
2. ไพศาล สีลาธรกิจ, เมธี ไทยอุทิศ, อวิรุทธ์ วิจิตรเมฆทอง. “หุ่นยนต์เคลื่อนที่ตามวัตถุ Object tracking robot”. ปรียญานิพนธ์วิศวกรรมศาสตร์บัณฑิต. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2550.
3. ทีมงานสมาร์ทเลิร์นนิ่ง , Advance PIC Microcontroller in C ประยุกต์ใช้งาน PIC ขั้นสูงด้วย ภาษา C



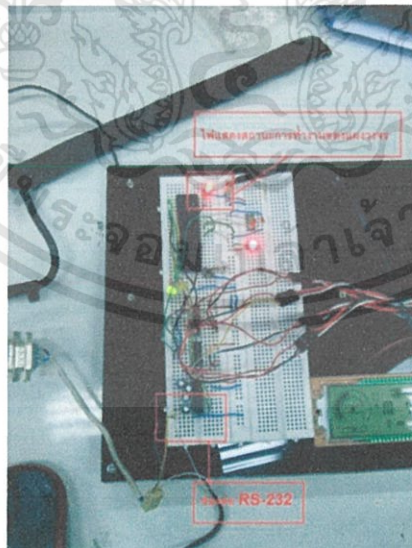


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพแสดงส่วนประกอบชิ้นงาน

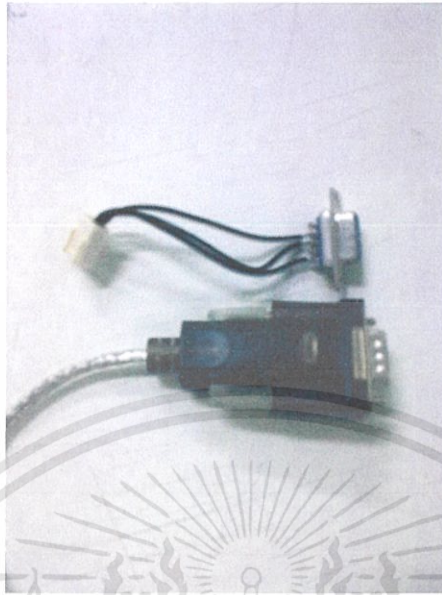


รูปที่ 1 ไฟแสดงสถานะการทำงานต่างๆ

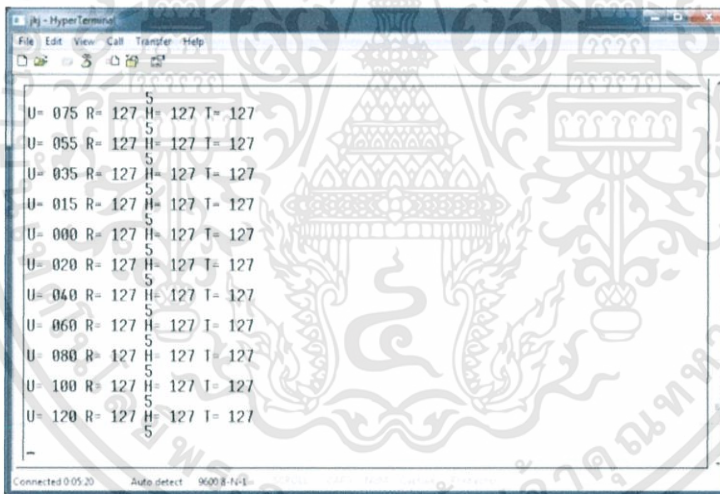


รูปที่ 2 ไฟแสดงสถานะและช่องต่อ RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 หัวต่อ RS-232



รูปที่ 4 หน้าจอแสดงการควบคุมผ่าน Hyperterminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคแสดงการบินของฮอลิคอปเตอร์



รูปที่ 5 การบังคับขึ้นลง

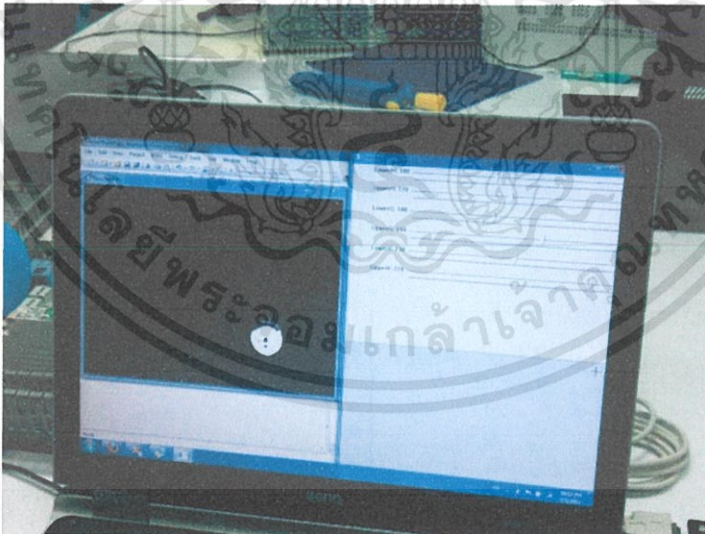


รูปที่ 6 การบังคับเลี้ยวซ้ายและเลี้ยวขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 การนำกล้องมาติดในตัวฮอบลีคอปเตอร์



รูปที่ 8 การทำImage prossing จากภาพที่ได้จากฮอบลีคอปเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void main()
{

    unsigned char t[]="Start Program by INS 32";
    unsigned char i;
    unsigned char receive;
    short clearbuff;
    int upsignal=0,rotate=125,head=130,tilt=116;

    int k;
    int runu[3],runr[3],runh[3],runt[3];
    int recu,recr,rech,rect;
    //////////////////////////////////////
    Usart_Init(9600);
    Spi_Init();
    TRISB=0; //open port b to output
    TRISD=0; //open port c to test receive and transimtion
    PORTB=0; //clear port B
    PORTD=0; //clear status for receive and transimtion,set Bit RD1 for run
    TRISE=0; // Standard configuration for USE CS at PORT E
    PORTE=0; ///
    ADCON1=0x0F; ///change to digital
    //////////////////////////////////////
    for(i=0;i<23;i++)Usart_Write(t[i]);
    Usart_Write('\n');
    Usart_Write('\n');
    Usart_Write('\r');

    clearbuff=Usart_Read();

    while(1)
    {

        PORTD.F0=1; // PORT RD0 SET When Program run (RED LED)

    ////////////////////////////////////// USART FORM RS232 //////////////////////////////////////
        if(Usart_Data_Ready())
        {
            PORTD.F3=1; // if usart data ready , Green LED at RD3 'on'
            delay_ms(100);
            PORTD.F2=0; // Yellow LED to off
            delay_ms(100);
            receive=Usart_Read();
            if((receive=='w')||(receive=='s')||(receive=='[')||(receive==''])

```

```

||(receive=='o')||(receive=='l')||(receive=='e')||(receive=='q')
||(receive=='c')||(receive=='b')
{
if(receive=='w')    // up
{
        // increase signal to control

        upsignal+=4;
        PORTD.F4=1; // LED 1 on
    }
if(receive=='s')    // down
{
        // decrease signal to control

        upsignal-=4;
        PORTD.F5=1; // LED 2 on
        PORTD.F4=0; // LED 1 off

    }

if(receive=='[')    // left
{
        rotate+=112;
}
if(receive==']')    //right
{
        rotate-=112;
}

if(receive=='o')    //forward
{
        head+=121;
        PORTD.F6=1; // LED 3 on
        PORTD.F5=0; // LED 2 off

    }

if(receive=='l')    //back
{
        head-=121;
        PORTD.F7=1; // LED 4 on
        PORTD.F6=0; // LED 3 off

    }

if(receive=='e')    //>>>
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    tilt+=123;

}
if(receive=='q')    //<<<
{
    tilt=123;

}
if(receive=='c')    // stop all and clear
{
    upsignal=0;
    rotate=125;
    head=130;
    tilt=116;
    PORTD.F7=0;    /// LED 4 off
}
if(receive=='b')    // HOLD Helicopter when Helicopter is flying
{
    rotate=0;
    head=255;
    tilt=10;
}
////////////////////////////////////
//////////////////////////////////// set range at control signal ///////////////////////////////////
////////////////////////////////////

if(upsignal>255)upsignal=255;
if(upsignal<0)upsignal=0;
if(rotate>237)rotate=237;
if(rotate<13)rotate=13;
if(head>251)head=251;
if(head<9)head=9;
if(tilt>239)tilt=239;
if(tilt<7)tilt=7;

///Convert Integral to Hyperterminal    //////////////////////////////////

recu=upsignal; // Convert at upsignal

runu[3]=(recu%10)+48; //for Unit
runu[2]=(((recu%100)-(recu%10))/10)+48; //for Ten
runu[1]=(((recu%1000)-(recu%100))/100)+48; //for Hundred
////////////////////////////////////
recr=rotate; // Convert at rotate

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

runr[3]=(recr%10)+48; //for Unit
runr[2]=(((recr%100)-(recr%10))/10)+48; //for Ten
runr[1]=(((recr%1000)-(recr%100))/100)+48; //for Hundred
////////////////////////////////////

rech=head; // Convert at head

runh[3]=(rech%10)+48; //for Unit
runh[2]=(((rech%100)-(rech%10))/10)+48; //for Ten
runh[1]=(((rech%1000)-(rech%100))/100)+48; //for Hundred
//////// ///////////////////////////////////////////////////
rect=tilt; // Convert at tilt

runt[3]=(rect%10)+48; //for Unit
runt[2]=(((rect%100)-(rect%10))/10)+48; //for Ten
runt[1]=(((rect%1000)-(rect%100))/100)+48; //for Hundred
//////Show output to Hyperterminal's display ///////////////////////////////////////////////////

Usart_Write('U');
Usart_Write('=');
Usart_Write(' ');
for(k=1;k<=3;k++)Usart_Write(runu[k]);
Usart_Write(' ');
Usart_Write('R');
Usart_Write('=');
Usart_Write(' ');
for(k=1;k<=3;k++)Usart_Write(runr[k]);
Usart_Write(' ');
Usart_Write('H');
Usart_Write('=');
Usart_Write(' ');
for(k=1;k<=3;k++)Usart_Write(runh[k]);
Usart_Write(' ');
Usart_Write('T');
Usart_Write('=');
Usart_Write(' ');
for(k=1;k<=3;k++)Usart_Write(runt[k]);
Usart_Write(' ');
for(k=1;k<=7;k++)Usart_Write('\t'); // repair Bug
for(k=1;k<=15;k++)Usart_Write(' '); // repair Bug
Usart_Write((rect%10)+48);

Usart_Write('\n');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Usart_Write('\r');

////////////////////////////////////
//////// Sent to Potentiometer //////////
////////////////////////////////////

PORTE.F2 = 0 ;// select CS to low

Spi_Write(0x12); //select poten 1 for IC 1
Spi_Write(tilt); //Write to data register
Spi_Write(0x12); //select poten 1 for IC 2
Spi_Write(rotate);

PORTE.F2 = 1; // select CS to high
delay_ms(50);
PORTE.F2 = 0; // deselect CS to low
Spi_Write(0x11); //select poten 0 for IC 1
Spi_Write(head);
Spi_Write(0x11); //select poten 0 for IC 2
Spi_Write(upsignal);

PORTE.F2 = 1 ;// deselect CS to low
}
}

//////////////////////////////// Nothing USART FORM RS 232 //////////

if(!Usart_Data_Ready())
{
    PORTD.F3=0; // if nothing usart data ready , Green LED at RD3 [OFF ]
    PORTD.F2=1; // yellow [ON]at RD2

}
}
}

```

Single/Dual Digital Potentiometer with SPI™ Interface

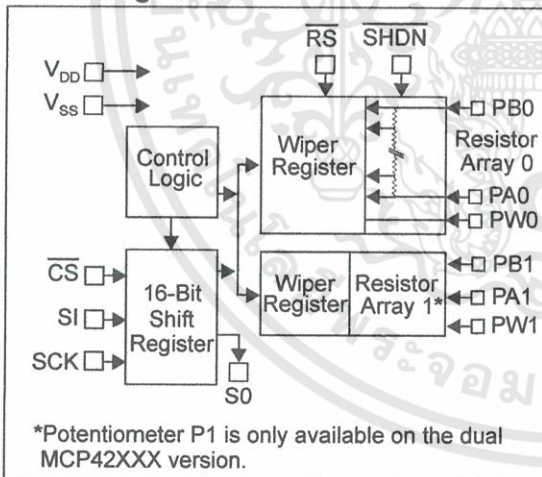
Features

- 256 taps for each potentiometer
- Potentiometer values for 10 kΩ, 50 kΩ and 100 kΩ
- Single and dual versions
- SPI™ serial interface (mode 0,0 and 1,1)
- ±1 LSB max INL & DNL
- Low power CMOS technology
- 1 μA maximum supply current in static operation
- Multiple devices can be daisy-chained together (MCP42XXX only)
- Shutdown feature open circuits of all resistors for maximum power savings
- Hardware shutdown pin available on MCP42XXX only
- Single supply operation (2.7V - 5.5V)
- Industrial temperature range: -40°C to +85°C
- Extended temperature range: -40°C to +125°C

Description

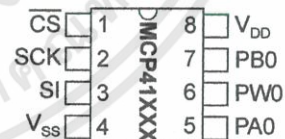
The MCP41XXX and MCP42XXX devices are 256-position, digital potentiometers available in 10 kΩ, 50 kΩ and 100 kΩ resistance versions. The MCP41XXX is a single-channel device and is offered in an 8-pin PDIP or SOIC package. The MCP42XXX contains two independent channels in a 14-pin PDIP, SOIC or TSSOP package. The wiper position of the MCP41XXX/42XXX varies linearly and is controlled via an industry-standard SPI interface. The devices consume <1 μA during static operation. A software shutdown feature is provided that disconnects the "A" terminal from the resistor stack and simultaneously connects the wiper to the "B" terminal. In addition, the dual MCP42XXX has a SHDN pin that performs the same function in hardware. During shutdown mode, the contents of the wiper register can be changed and the potentiometer returns from shutdown to the new value. The wiper is reset to the mid-scale position (80h) upon power-up. The RS (reset) pin implements a hardware reset and also returns the wiper to mid-scale. The MCP42XXX SPI interface includes both the SI and SO pins, allowing daisy-chaining of multiple devices. Channel-to-channel resistance matching on the MCP42XXX varies by less than 1%. These devices operate from a single 2.7 - 5.5V supply and are specified over the extended and industrial temperature ranges.

Block Diagram

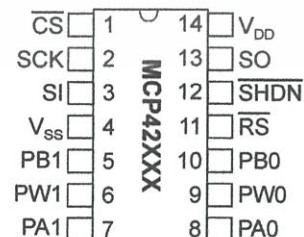


Package Types

PDIP/SOIC



PDIP/SOIC/TSSOP



MCP41XXX/42XXX

1.0 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS: 10 kΩ VERSION

Electrical Characteristics: Unless otherwise indicated, $V_{DD} = +2.7V$ to $5.5V$, $T_A = -40^\circ C$ to $+85^\circ C$ (TSSOP devices are only specified at $+25^\circ C$ and $+85^\circ C$). Typical specifications represent values for $V_{DD} = 5V$, $V_{SS} = 0V$, $V_B = 0V$, $T_A = +25^\circ C$.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Rheostat Mode						
Nominal Resistance	R	8	10	12	kΩ	$T_A = +25^\circ C$ (Note 1)
Rheostat Differential Non Linearity	R-DNL	-1	±1/4	+1	LSB	Note 2
Rheostat Integral Non Linearity	R-INL	-1	±1/4	+1	LSB	Note 2
Rheostat Tempco	$\Delta R_{AB}/\Delta T$	—	800	—	ppm/°C	
Wiper Resistance	R_W	—	52	100	Ω	$V_{DD} = 5.5V$, $I_W = 1$ mA, code 00h
	R_W	—	73	125	Ω	$V_{DD} = 2.7V$, $I_W = 1$ mA, code 00h
Wiper Current	I_W	-1	—	+1	mA	
Nominal Resistance Match	$\Delta R/R$	—	0.2	1	%	MCP42010 only, P0 to P1; $T_A = +25^\circ C$
Potentiometer Divider						
Resolution	N	8	—	—	Bits	
Monotonicity	N	8	—	—	Bits	
Differential Non-Linearity	DNL	-1	±1/4	+1	LSB	Note 3
Integral Non-Linearity	INL	-1	±1/4	+1	LSB	Note 3
Voltage Divider Tempco	$\Delta V_W/\Delta T$	—	1	—	ppm/°C	Code 80h
Full Scale Error	V_{WFSE}	-2	-0.7	0	LSB	Code FFh, $V_{DD} = 5V$, see Figure 2-25
	V_{WFSE}	-2	-0.7	0	LSB	Code FFh, $V_{DD} = 3V$, see Figure 2-25
Zero Scale Error	V_{WZSE}	0	+0.7	+2	LSB	Code 00h, $V_{DD} = 5V$, see Figure 2-25
	V_{WZSE}	0	+0.7	+2	LSB	Code 00h, $V_{DD} = 3V$, see Figure 2-25
Resistor Terminals						
Voltage Range	$V_{A,B,W}$	0	—	V_{DD}		Note 4
Capacitance (C_A or C_B)		—	15	—	pF	$f = 1$ MHz, Code = 80h, see Figure 2-30
Capacitance	C_W	—	5.6	—	pF	$f = 1$ MHz, Code = 80h, see Figure 2-30
Dynamic Characteristics (All dynamic characteristics use $V_{DD} = 5V$)						
Bandwidth -3dB	BW	—	1	—	MHz	$V_B = 0V$, Measured at Code 80h, Output Load = 30 pF
Settling Time	t_s	—	2	—	μs	$V_A = V_{DD}$, $V_B = 0V$, ±1% Error Band, Transition from Code 00h to Code 80h, Output Load = 30 pF
Resistor Noise Voltage	e_{NWB}	—	9	—	nV/√Hz	$V_A = \text{Open}$, Code 80h, $f = 1$ kHz
Crosstalk	C_T	—	-95	—	dB	$V_A = V_{DD}$, $V_B = 0V$ (Note 5)
Digital Inputs/Outputs (CS, SCK, SI, SO) See Figure 2-12 for RS and SHDN pin operation						
Schmitt Trigger High-Level Input Voltage	V_{IH}	$0.7V_{DD}$	—	—	V	
Schmitt Trigger Low-Level Input Voltage	V_{IL}	—	—	$0.3V_{DD}$	V	
Hysteresis of Schmitt Trigger Inputs	V_{HYS}	—	$0.05V_{DD}$	—		
Low-Level Output Voltage	V_{OL}	—	—	0.40	V	$I_{OL} = 2.1$ mA, $V_{DD} = 5V$
High-Level Output Voltage	V_{OH}	$V_{DD} - 0.5$	—	—	V	$I_{OH} = -400$ μA, $V_{DD} = 5V$
Input Leakage Current	I_{LI}	-1	—	+1	μA	$\overline{CS} = V_{DD}$, $V_{IN} = V_{SS}$ or V_{DD} , includes V_A SHDN=0
Pin Capacitance (All inputs/outputs)	C_{IN}, C_{OUT}	—	10	—	pF	$V_{DD} = 5.0V$, $T_A = +25^\circ C$, $f_c = 1$ MHz
Power Requirements						
Operating Voltage Range	V_{DD}	2.7	—	5.5	V	
Supply Current, Active	I_{DDA}	—	340	500	μA	$V_{DD} = 5.5V$, $\overline{CS} = V_{SS}$, $f_{SCK} = 10$ MHz, SO = Open, Code FFh (Note 6)
Supply Current, Static	I_{DDs}	—	0.01	1	μA	\overline{CS} , SHDN, $\overline{RS} = V_{DD} = 5.5V$, SO = Open (Note 6)
Power Supply Sensitivity	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 4.5V - 5.5V$, $V_A = 4.5V$, Code 80h
	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 2.7V - 3.3V$, $V_A = 2.7V$, Code 80h

- Note**
- $V_{AB} = V_{DD}$, no connection on wiper.
 - Rheostat position non-linearity R-INL is the deviation from an ideal value measured between the maximum resistance and the minimum resistance wiper positions. R-DNL measures the relative step change from the ideal between successive tap positions. $I_W = 50$ μA for $V_{DD} = 3V$ and $I_W = 400$ μA for $V_{DD} = 5V$ for 10 kΩ version. See Figure 2-26 for test circuit.
 - INL and DNL are measured at V_W with the device configured in the voltage divider or potentiometer mode. $V_A = V_{DD}$ and $V_B = 0V$. DNL specification limits of ±1 LSB max are specified monotonic operating conditions. See Figure 2-25 for test circuit.
 - Resistor terminals A,B and W have no restrictions on polarity with respect to each other. Full-scale and zero-scale error were measured using Figure 2-25.
 - Measured at V_W pin where the voltage on the adjacent V_W pin is swinging full-scale.
 - Supply current is independent of current through the potentiometers.

DC CHARACTERISTICS: 50 kΩ VERSION

Electrical Characteristics: Unless otherwise indicated, $V_{DD} = +2.7V$ to $5.5V$, $T_A = -40^\circ C$ to $+85^\circ C$ (TSSOP devices are only specified at $+25^\circ C$ and $+85^\circ C$). Typical specifications represent values for $V_{DD} = 5V$, $V_{SS} = 0V$, $V_B = 0V$, $T_A = +25^\circ C$.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Rheostat Mode						
Nominal Resistance	R	35	50	65	kΩ	$T_A = +25^\circ C$ (Note 1)
Rheostat Differential Non-Linearity	R-DNL	-1	$\pm 1/4$	+1	LSB	Note 2
Rheostat Integral Non-Linearity	R-INL	-1	$\pm 1/4$	+1	LSB	Note 2
Rheostat Tempo	$\Delta R_{AB}/\Delta T$	—	800	—	ppm/ $^\circ C$	
Wiper Resistance	R_W	—	125	175	Ω	$V_{DD} = 5.5V$, $I_W = 1 mA$, code 00h
	R_W	—	175	250	Ω	$V_{DD} = 2.7V$, $I_W = 1 mA$, code 00h
Wiper Current	I_W	-1	—	+1	mA	
Nominal Resistance Match	$\Delta R/R$	—	0.2	1	%	MCP42050 only, P0 to P1; $T_A = +25^\circ C$
Potentiometer Divider						
Resolution	N	8	—	—	Bits	
Monotonicity	N	8	—	—	Bits	
Differential Non-Linearity	DNL	-1	$\pm 1/4$	+1	LSB	Note 3
Integral Non-Linearity	INL	-1	$\pm 1/4$	+1	LSB	Note 3
Voltage Divider Tempo	$\Delta V_W/\Delta T$	—	1	—	ppm/ $^\circ C$	Code 80h
Full-Scale Error	V_{WFSE}	-1	-0.25	0	LSB	Code FFh, $V_{DD} = 5V$, see Figure 2-25
	V_{WFSE}	-1	-0.35	0	LSB	Code FFh, $V_{DD} = 3V$, see Figure 2-25
Zero-Scale Error	V_{WZSE}	0	+0.25	+1	LSB	Code 00h, $V_{DD} = 5V$, see Figure 2-25
	V_{WZSE}	0	+0.35	+1	LSB	Code 00h, $V_{DD} = 3V$, see Figure 2-25
Resistor Terminals						
Voltage Range	$V_{A,B,W}$	0	—	V_{DD}		Note 4
Capacitance (C_A or C_B)		—	11	—	pF	$f = 1 MHz$, Code = 80h, see Figure 2-30
Capacitance	C_W	—	5.6	—	pF	$f = 1 MHz$, Code = 80h, see Figure 2-30
Dynamic Characteristics (All dynamic characteristics use $V_{DD} = 5V$)						
Bandwidth -3dB	BW	—	280	—	MHz	$V_B = 0V$, Measured at Code 80h, Output Load = 30 pF
Settling Time	t_s	—	8	—	μs	$V_A = V_{DD}$, $V_B = 0V$, $\pm 1\%$ Error Band, Transition from Code 00h to Code 80h, Output Load = 30 pF
Resistor Noise Voltage	e_{NWB}	—	20	—	nV/ \sqrt{Hz}	$V_A = \text{Open}$, Code 80h, $f = 1 kHz$
Crosstalk	C_T	—	-95	—	dB	$V_A = V_{DD}$, $V_B = 0V$ (Note 5)
Digital Inputs/Outputs (CS, SCK, SI, SO) See Figure 2-12 for RS and SHDN pin operation.						
Schmitt Trigger High-Level Input Voltage	V_{IH}	$0.7V_{DD}$	—	—	V	
Schmitt Trigger Low-Level Input Voltage	V_{IL}	—	—	$0.3V_{DD}$	V	
Hysteresis of Schmitt Trigger Inputs	V_{HYS}	—	$0.05V_{DD}$	—		
Low-Level Output Voltage	V_{OL}	—	—	0.40	V	$I_{OL} = 2.1 mA$, $V_{DD} = 5V$
High-Level Output Voltage	V_{OH}	$V_{DD} - 0.5$	—	—	V	$I_{OH} = -400 \mu A$, $V_{DD} = 5V$
Input Leakage Current	I_{LI}	-1	—	+1	μA	$\overline{CS} = V_{DD}$, $V_{IN} = V_{SS}$ or V_{DD} , includes V_A $\overline{SHDN} = 0$
Pin Capacitance (All inputs/outputs)	C_{IN}, C_{OUT}	—	10	—	pF	$V_{DD} = 5.0V$, $T_A = +25^\circ C$, $f_c = 1 MHz$
Power Requirements						
Operating Voltage Range	V_{DD}	2.7	—	5.5	V	
Supply Current, Active	I_{DDA}	—	340	500	μA	$V_{DD} = 5.5V$, $\overline{CS} = V_{SS}$, $f_{SCK} = 10 MHz$, $SO = \text{Open}$, Code FFh (Note 6)
Supply Current, Static	I_{DDS}	—	0.01	1	μA	\overline{CS} , \overline{SHDN} , $RS = V_{DD} = 5.5V$, $SO = \text{Open}$ (Note 6)
Power Supply Sensitivity	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 4.5V - 5.5V$, $V_A = 4.5V$, Code 80h
	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 2.7V - 3.3V$, $V_A = 2.7V$, Code 80h

- Note**
- $V_{AB} = V_{DD}$, no connection on wiper.
 - Rheostat position non-linearity R-INL is the deviation from an ideal value measured between the maximum resistance and the minimum resistance wiper positions. R-DNL measures the relative step change from the ideal between successive tap positions. $I_W = V_{DD}/R$ for +3V or +5V for 50 kΩ version. See Figure 2-26 for test circuit.
 - INL and DNL are measured at V_W with the device configured in the voltage divider or potentiometer mode. $V_A = V_{DD}$ and $V_B = 0V$. DNL specification limits of ± 1 LSB max are specified monotonic operating conditions. See Figure 2-25 for test circuit.
 - Resistor terminals A,B and W have no restrictions on polarity with respect to each other. Full-scale and zero-scale error were measured using Figure 2-25.
 - Measured at V_W pin where the voltage on the adjacent V_W pin is swinging full scale.
 - Supply current is independent of current through the potentiometers.

MCP41XXX/42XXX

DC CHARACTERISTICS: 100 kΩ VERSION

Electrical Characteristics: Unless otherwise indicated, $V_{DD} = +2.7V$ to $5.5V$, $T_A = -40^\circ C$ to $+85^\circ C$ (TSSOP devices are only specified at $+25^\circ C$ and $+85^\circ C$). Typical specifications represent values for $V_{DD} = 5V$, $V_{SS} = 0V$, $V_B = 0V$, $T_A = +25^\circ C$.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Rheostat Mode						
Nominal Resistance	R	70	100	130	kΩ	$T_A = +25^\circ C$ (Note 1)
Rheostat Differential Non-Linearity	R-DNL	-1	±1/4	+1	LSB	Note 2
Rheostat Integral Non-Linearity	R-INL	-1	±1/4	+1	LSB	Note 2
Rheostat Tempco	$\Delta R_{AB}/\Delta T$	—	800	—	ppm/°C	
Wiper Resistance	R_W	—	125	175	Ω	$V_{DD} = 5.5V$, $I_W = 1$ mA, code 00h
	R_W	—	175	250	Ω	$V_{DD} = 2.7V$, $I_W = 1$ mA, code 00h
Wiper Current	I_W	-1	—	+1	mA	
Nominal Resistance Match	$\Delta R/R$	—	0.2	1	%	MCP42010 only, P0 to P1; $T_A = +25^\circ C$
Potentiometer Divider						
Resolution	N	8	—	—	Bits	
Monotonicity	N	8	—	—	Bits	
Differential Non-Linearity	DNL	-1	±1/4	+1	LSB	Note 3
Integral Non-Linearity	INL	-1	±1/4	+1	LSB	Note 3
Voltage Divider Tempco	$\Delta V_W/\Delta T$	—	1	—	ppm/°C	Code 80h
Full-Scale Error	V_{WFSE}	-1	-0.25	0	LSB	Code FFh, $V_{DD} = 5V$, see Figure 2-25
	V_{WFSE}	-1	-0.35	0	LSB	Code FFh, $V_{DD} = 3V$, see Figure 2-25
Zero-Scale Error	V_{WZSE}	0	+0.25	+1	LSB	Code 00h, $V_{DD} = 5V$, see Figure 2-25
	V_{WZSE}	0	+0.35	+1	LSB	Code 00h, $V_{DD} = 3V$, see Figure 2-25
Resistor Terminals						
Voltage Range	$V_{A,B,W}$	0	—	V_{DD}		Note 4
Capacitance (CA or CB)		—	11	—	pF	f = 1 MHz, Code = 80h, see Figure 2-30
Capacitance	C_W	—	5.6	—	pF	f = 1 MHz, Code = 80h, see Figure 2-30
Dynamic Characteristics (All dynamic characteristics use $V_{DD} = 5V$.)						
Bandwidth -3dB	BW	—	145	—	MHz	$V_B = 0V$, Measured at Code 80h, Output Load = 30 pF
Settling Time	t_s	—	18	—	μs	$V_A = V_{DD}$, $V_B = 0V$, ±1% Error Band, Transition from Code 00h to Code 80h, Output Load = 30 pF
Resistor Noise Voltage	e_{NWB}	—	29	—	nV/√Hz	$V_A = \text{Open}$, Code 80h, f = 1 kHz
Crosstalk	C_T	—	-95	—	dB	$V_A = V_{DD}$, $V_B = 0V$ (Note 5)
Digital Inputs/Outputs (CS, SCK, SI, SO) See Figure 2-12 for RS and SHDN pin operation.						
Schmitt Trigger High-Level Input Voltage	V_{IH}	$0.7V_{DD}$	—	—	V	
Schmitt Trigger Low-Level Input Voltage	V_{IL}	—	—	$0.3V_{DD}$	V	
Hysteresis of Schmitt Trigger Inputs	V_{HYS}	—	$0.05V_{DD}$	—		
Low-Level Output Voltage	V_{OL}	—	—	0.40	V	$I_{OL} = 2.1$ mA, $V_{DD} = 5V$
High-Level Output Voltage	V_{OH}	$V_{DD} - 0.5$	—	—	V	$I_{OH} = -400$ μA, $V_{DD} = 5V$
Input Leakage Current	I_{IJ}	-1	—	+1	μA	$\overline{CS} = V_{DD}$, $V_{IN} = V_{SS}$ or V_{DD} , includes V_A SHDN=0
Pin Capacitance (All inputs/outputs)	C_{IN}, C_{OUT}	—	10	—	pF	$V_{DD} = 5.0V$, $T_A = +25^\circ C$, $f_c = 1$ MHz
Power Requirements						
Operating Voltage Range	V_{DD}	2.7	—	5.5	V	
Supply Current, Active	I_{DDA}	—	340	500	μA	$V_{DD} = 5.5V$, $\overline{CS} = V_{SS}$, $f_{SCK} = 10$ MHz, SO = Open, Code FFh (Note 6)
Supply Current, Static	I_{DDS}	—	0.01	1	μA	\overline{CS} , SHDN, $\overline{RS} = V_{DD} = 5.5V$, SO = Open (Note 6)
Power Supply Sensitivity	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 4.5V - 5.5V$, $V_A = 4.5V$, Code 80h
	PSS	—	0.0015	0.0035	%/%	$V_{DD} = 2.7V - 3.3V$, $V_A = 2.7V$, Code 80h

- Note**
- $V_{AB} = V_{DD}$, no connection on wiper.
 - Rheostat position non-linearity R-INL is the deviation from an ideal value measured between the maximum resistance and the minimum resistance wiper positions. R-DNL measures the relative step change from the ideal between successive tap positions. $I_W = 50$ μA for $V_{DD} = 3V$ and $I_W = 400$ μA for $V_{DD} = 5V$ for 10 kΩ version. See Figure 2-26 for test circuit.
 - INL and DNL are measured at V_W with the device configured in the voltage divider or potentiometer mode. $V_A = V_{DD}$ and $V_B = 0V$. DNL specification limits of ±1 LSB max are specified monotonic operating conditions. See Figure 2-25 for test circuit.
 - Resistor terminals A, B and W have no restrictions on polarity with respect to each other. Full-scale and zero-scale error were measured using Figure 2-25.
 - Measured at V_W pin where the voltage on the adjacent V_W pin is swinging full-scale.
 - Supply current is independent of current through the potentiometers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCP41XXX/42XXX

Absolute Maximum Ratings †

V _{DD}	7.0V
All inputs and outputs w.r.t. V _{SS}	-0.6V to V _{DD} +1.0V
Storage temperature.....	-60°C to +150°C
Ambient temp. with power applied.....	-60°C to +125°C
ESD protection on all pins.....	≥ 2 kV

† **Notice:** Stresses above those listed under "maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

AC TIMING CHARACTERISTICS

Electrical Characteristics: Unless otherwise indicated, V _{DD} = +2.7V to 5.5V, T _A = -40°C to +85°C.						
Parameter	Sym	Min.	Typ.	Max.	Units	Conditions
Clock Frequency	F _{CLK}	—	—	10	MHz	V _{DD} = 5V (Note 1)
Clock High Time	t _{HI}	40	—	—	ns	
Clock Low Time	t _{LO}	40	—	—	ns	
CS Fall to First Rising CLK Edge	t _{CSSR}	40	—	—	ns	
Data Input Setup Time	t _{SU}	40	—	—	ns	
Data Input Hold Time	t _{HD}	10	—	—	ns	
SCK Fall to SO Valid Propagation Delay	t _{DO}	—	—	80	ns	C _L = 30 pF (Note 2)
SCK Rise to CS Rise Hold Time	t _{CHS}	30	—	—	ns	
SCK Rise to CS Fall Delay	t _{CS0}	10	—	—	ns	
CS Rise to CLK Rise Hold	t _{CS1}	100	—	—	ns	
CS High Time	t _{CSH}	40	—	—	ns	
Reset Pulse Width	t _{RS}	150	—	—	ns	Note 2
RS Rising to CS Falling Delay Time	t _{RSCS}	150	—	—	ns	Note 2
CS rising to RS or SHDN falling delay time	t _{SE}	40	—	—	ns	Note 3
CS low time	t _{CSL}	100	—	—	ns	Note 3
Shutdown Pulse Width	t _{SH}	150	—	—	ns	Note 3

- Note 1:** When using the device in the daisy-chain configuration, maximum clock frequency is determined by a combination of propagation delay time (t_{DO}) and data input setup time (t_{SU}). Max. clock frequency is therefore ~ 5.8 MHz based on SCK rise and fall times of 5 ns, t_{HI} = 40 ns, t_{DO} = 80 ns and t_{SU} = 40 ns.
- Note 2:** Applies only to the MCP42XXX devices.
- Note 3:** Applies only when using hardware pins to exit software shutdown mode, MCP42XXX only.

MCP41XXX/42XXX

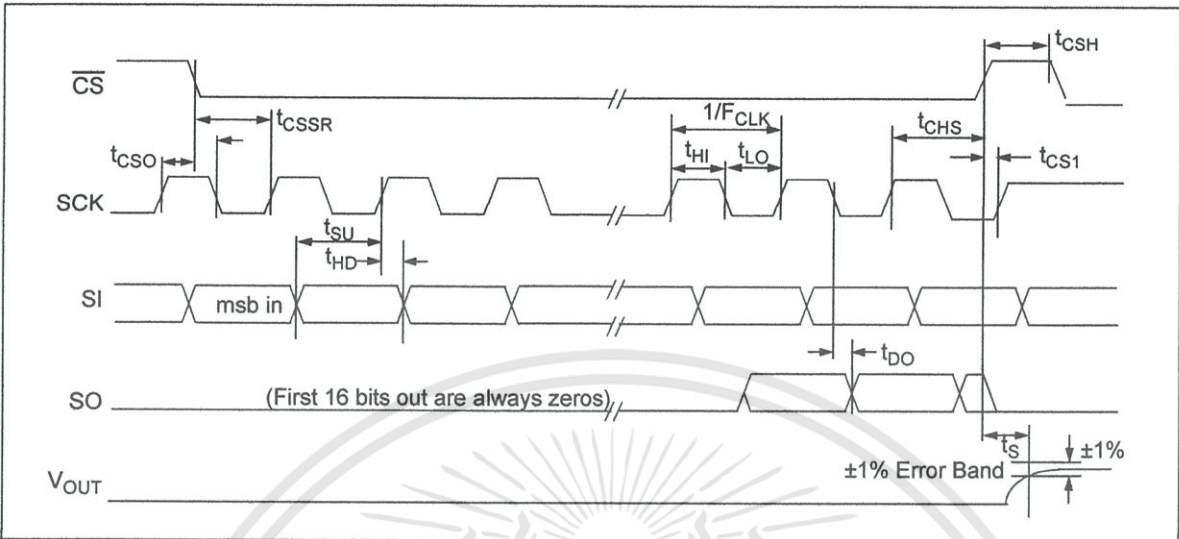


FIGURE 1-1: Detailed Serial interface Timing.

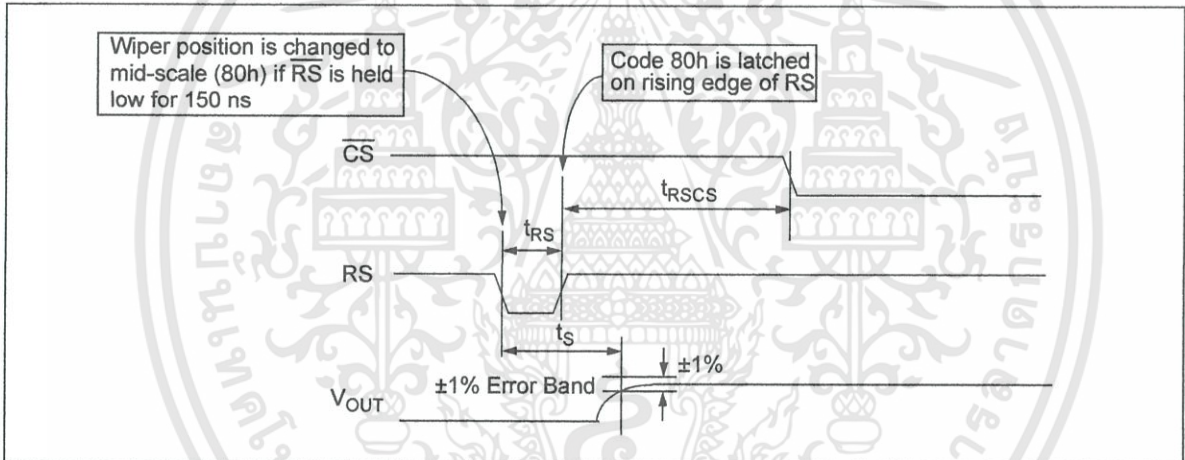


FIGURE 1-2: Reset Timing.

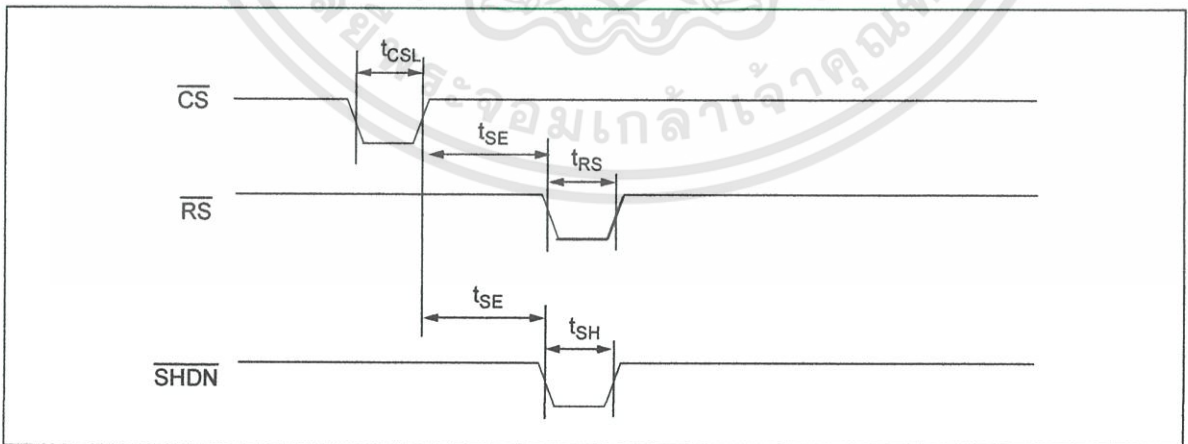


FIGURE 1-3: Software Shutdown Exit Timing.

2.0 TYPICAL PERFORMANCE CURVES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore outside the warranted range.

Note: Unless otherwise indicated, curve represents 10 k Ω , 50 k Ω and 100 k Ω devices, $V_{DD} = 5V$, $V_{SS} = 0V$, $T_A = +25^\circ C$, $V_B = 0V$.

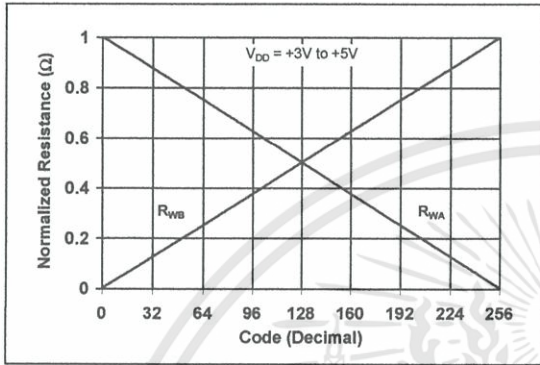


FIGURE 2-1: Normalized Wiper to End Terminal Resistance vs. Code.

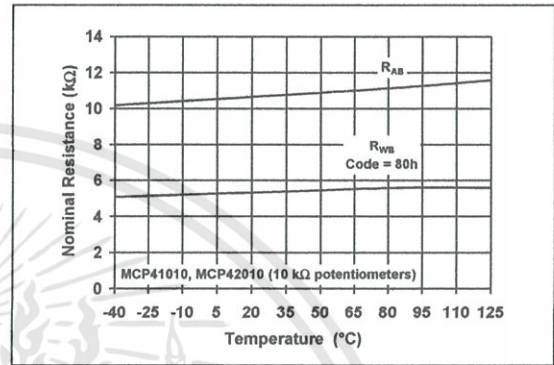


FIGURE 2-4: Nominal Resistance 10 k Ω vs. Temperature.

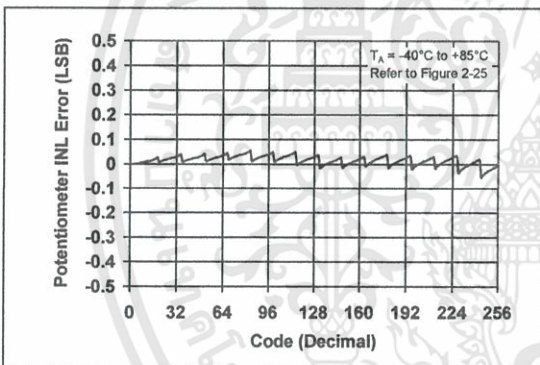


FIGURE 2-2: Potentiometer INL Error vs. Code.

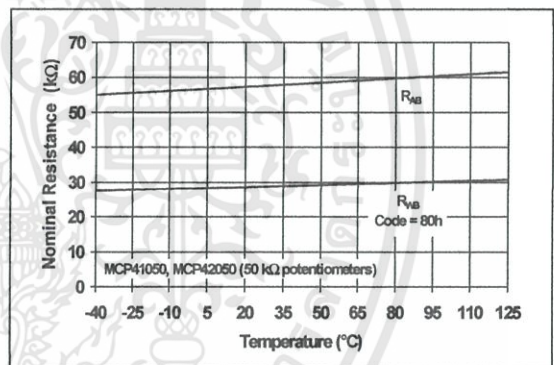


FIGURE 2-5: Nominal Resistance 50 k Ω vs. Temperature.

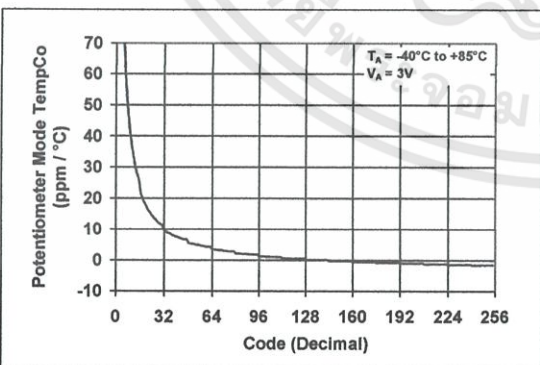


FIGURE 2-3: Potentiometer Mode Tempco vs. Code.

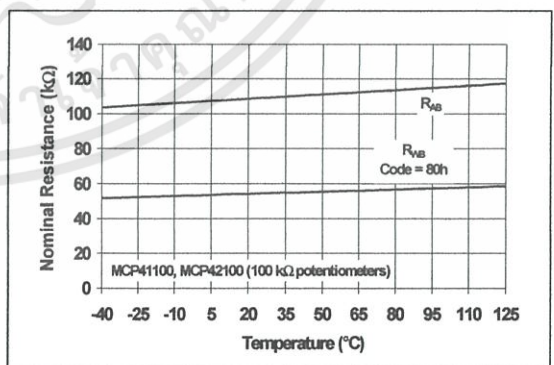


FIGURE 2-6: Nominal Resistance 100 k Ω vs. Temperature.

MCP41XXX/42XXX

Note: Unless otherwise indicated, curve represents 10 kΩ, 50 kΩ and 100 kΩ devices, $V_{DD} = 5V$, $V_{SS} = 0V$, $T_A = +25^\circ C$, $V_B = 0V$.

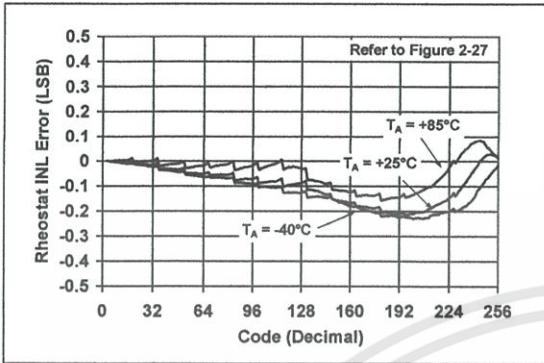


FIGURE 2-7: Rheostat INL Error vs. Code.

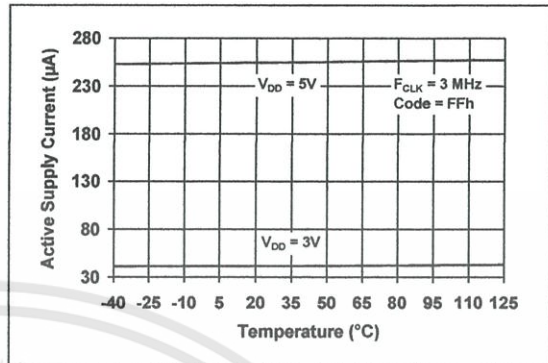


FIGURE 2-10: Active Supply Current vs. Temperature.

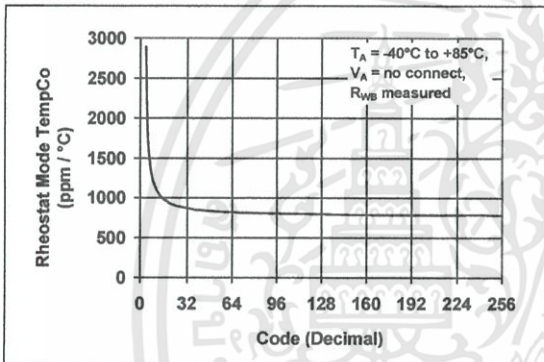


FIGURE 2-8: Rheostat Mode Tempco vs. Code.

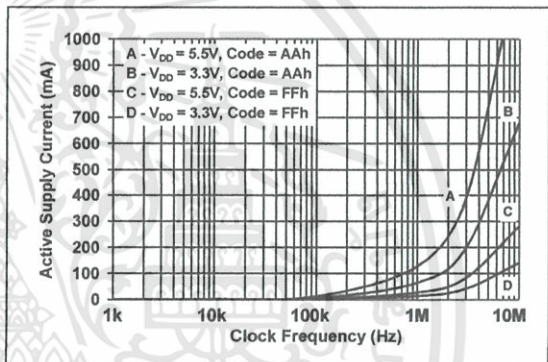


FIGURE 2-11: Active Supply Current vs. Clock Frequency.

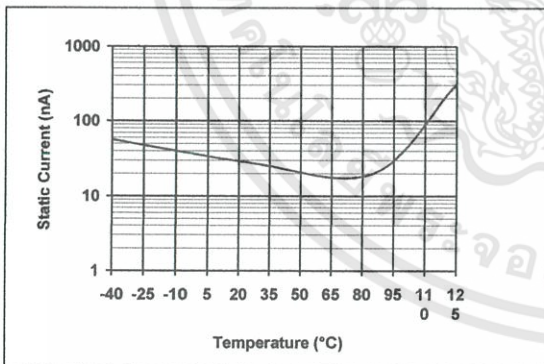


FIGURE 2-9: Static Current vs. Temperature.

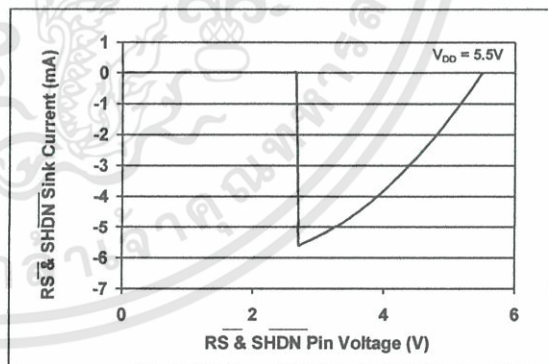


FIGURE 2-12: Reset & Shutdown Pins Current vs. Voltage.

MCP41XXX/42XXX

Note: Unless otherwise indicated, curve represents 10 k Ω , 50 k Ω and 100 k Ω devices, $V_{DD} = 5V$, $V_{SS} = 0V$, $T_A = +25^\circ C$, $V_B = 0V$.

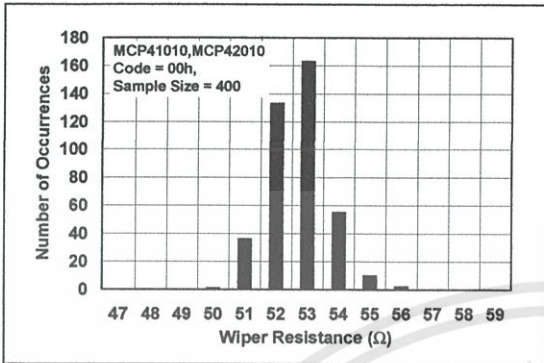


FIGURE 2-13: 10 k Ω Device Wiper Resistance Histogram.

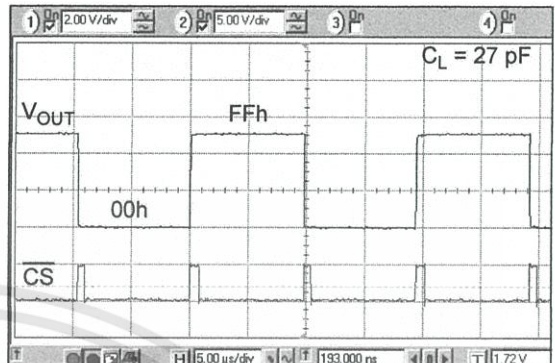


FIGURE 2-16: Full-Scale Settling Time.

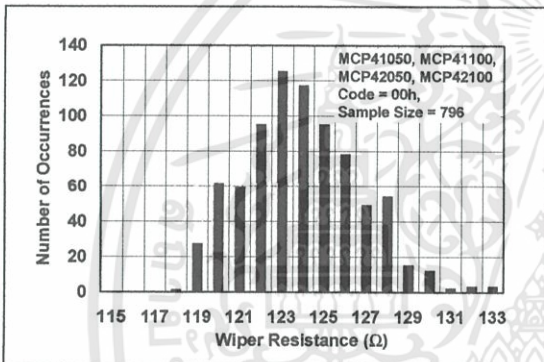


FIGURE 2-14: 50 k Ω , 100 k Ω Device Wiper Resistance Histogram.

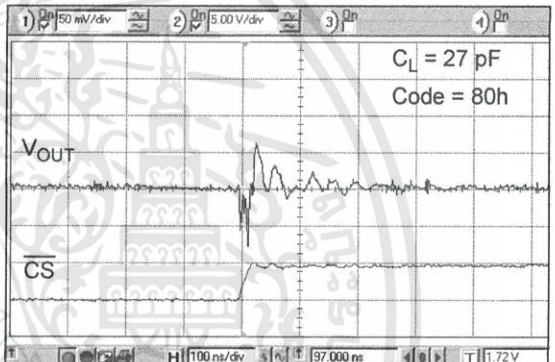


FIGURE 2-17: Digital Feed through vs. Time.

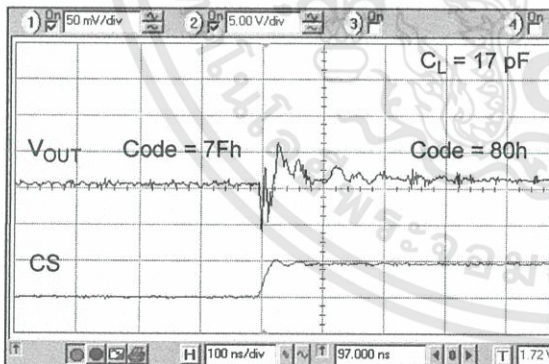


FIGURE 2-15: One Position Settling Time.

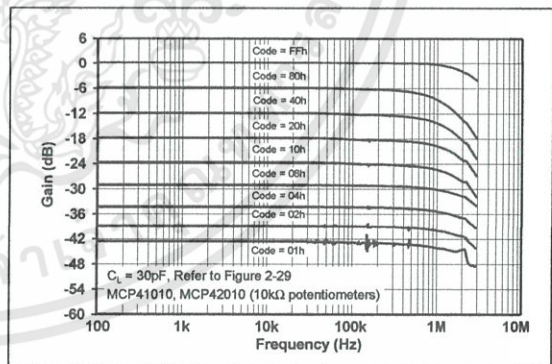


FIGURE 2-18: Gain vs. Frequency for 10 k Ω Potentiometer.

MCP41XXX/42XXX

Note: Unless otherwise indicated, curve represents 10 k Ω , 50 k Ω and 100 k Ω devices, $V_{DD} = 5V$, $V_{SS} = 0V$, $T_A = +25^\circ C$, $V_B = 0V$.

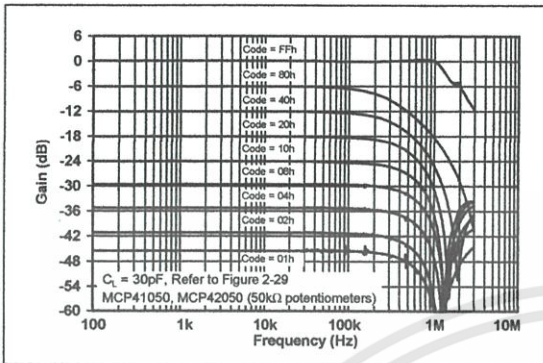


FIGURE 2-19: Gain vs. Frequency for 50k Ω Potentiometer.

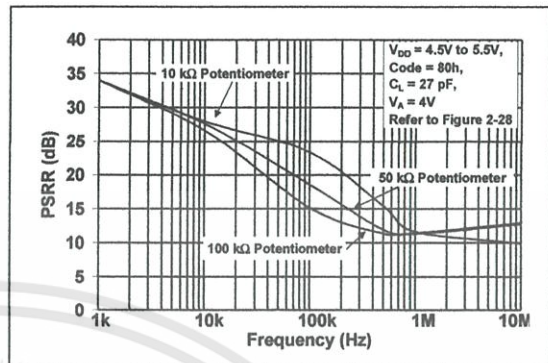


FIGURE 2-22: Power Supply Rejection Ratio vs. Frequency.

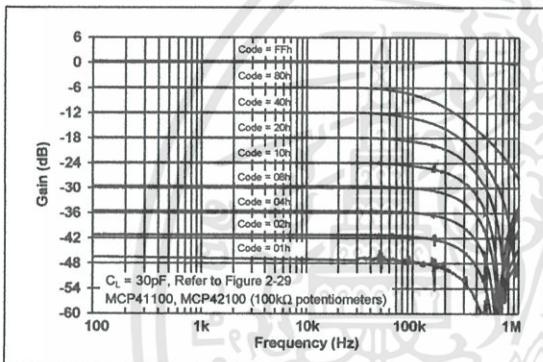


FIGURE 2-20: Gain vs. Frequency for 100k Ω Potentiometer.

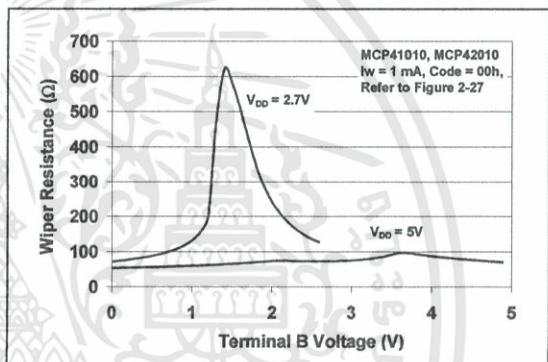


FIGURE 2-23: 10 k Ω Wiper Resistance vs. Voltage.

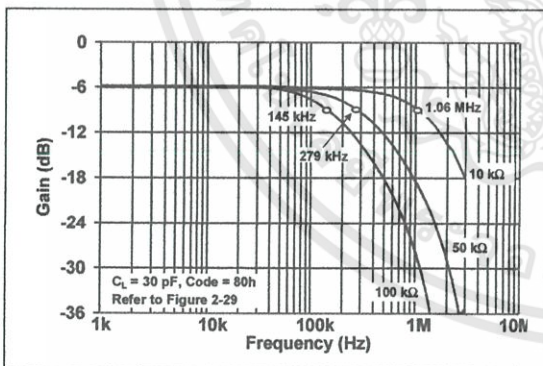


FIGURE 2-21: -3 dB Bandwidths.

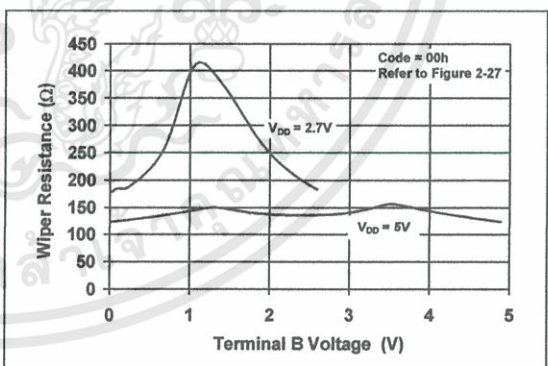


FIGURE 2-24: 50 k Ω & 100 k Ω Wiper Resistance vs. Voltage.

2.1 Parametric Test Circuits

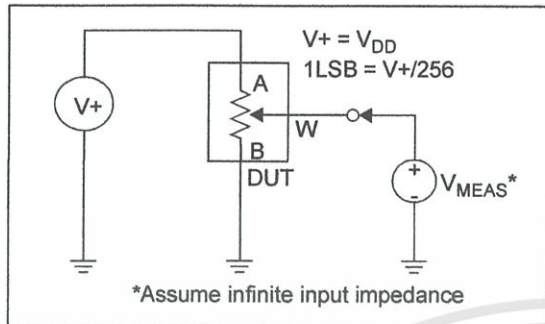


FIGURE 2-25: Potentiometer Divider Non-Linearity Error Test Circuit (DNL, INL).

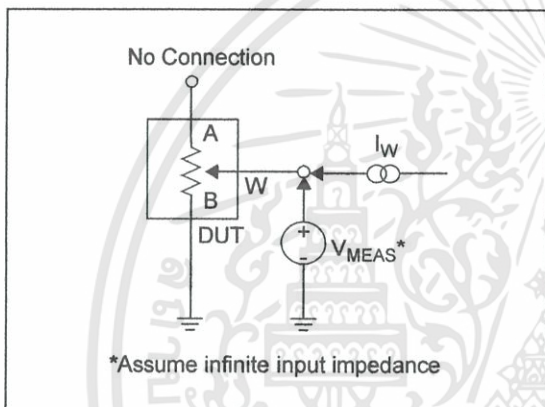


FIGURE 2-26: Resistor Position Non-Linearity Error Test Circuit (Rheostat operation DNL, INL).

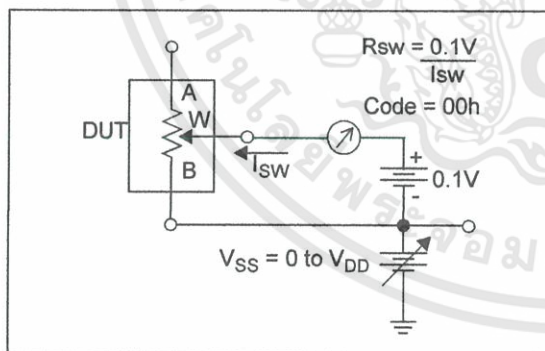


FIGURE 2-27: Wiper Resistance Test Circuit.

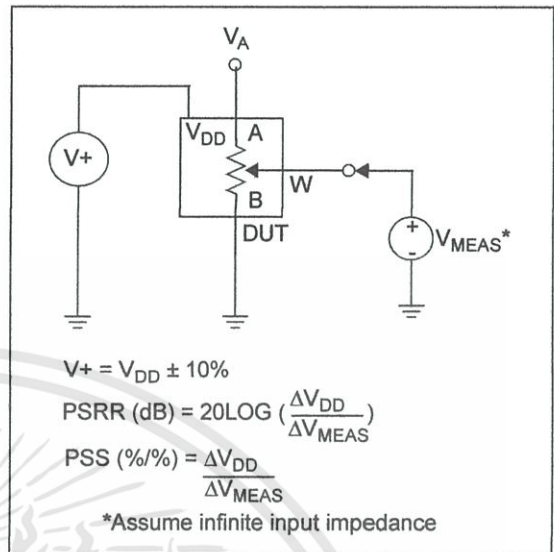


FIGURE 2-28: Power Supply Sensitivity Test Circuit (PSS, PSRR).

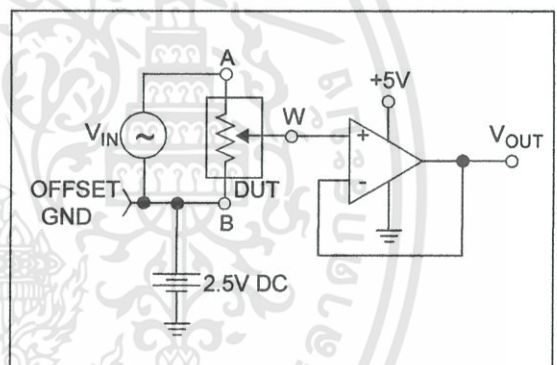


FIGURE 2-29: Gain vs. Frequency Test Circuit.

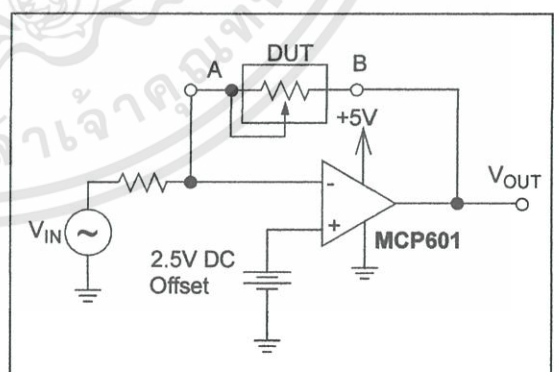


FIGURE 2-30: Capacitance Test Circuit.

MCP41XXX/42XXX

3.0 PIN DESCRIPTIONS

3.1 PA0, PA1

Potentiometer Terminal A Connection.

3.2 PB0, PB1

Potentiometer Terminal B Connection.

3.3 PW0, PW1

Potentiometer Wiper Connection.

3.4 Chip Select (\overline{CS})

This is the SPI port chip select pin and is used to execute a new command after it has been loaded into the shift register. This pin has a Schmitt Trigger input.

3.5 Serial Clock (SCK)

This is the SPI port clock pin and is used to clock-in new register data. Data is clocked into the SI pin on the rising edge of the clock and out the SO pin on the falling edge of the clock. This pin is gated to the \overline{CS} pin (i.e., the device will not draw any more current if the SCK pin is toggling when the \overline{CS} pin is high). This pin has a Schmitt Trigger input.

3.6 Serial Data Input (SI)

This is the SPI port serial data input pin. The command and data bytes are clocked into the shift register using this pin. This pin is gated to the \overline{CS} pin (i.e., the device will not draw any more current if the SI pin is toggling when the \overline{CS} pin is high). This pin has a Schmitt Trigger input.

3.7 Serial Data Output (SO) (MCP42XXX devices only)

This is the SPI port serial data output pin used for daisy-chaining more than one device. Data is clocked out of the SO pin on the falling edge of clock. This is a push-pull output and **does not** go to a high-impedance state when \overline{CS} is high. It will drive a logic-low when \overline{CS} is high.

3.8 Reset (\overline{RS}) (MCP42XXX devices only)

The Reset pin will set all potentiometers to mid-scale (Code 80h) if this pin is brought low for at least 150 ns. This pin should not be toggled low when the \overline{CS} pin is low. It is possible to toggle this pin when the \overline{SHDN} pin is low. In order to minimize power consumption, this pin has an active pull-up circuit. The performance of this circuit is shown in Figure 2-12. This pin will draw negligible current at logic level '0' and logic level '1'. Do not leave this pin floating.

3.9 Shutdown (\overline{SHDN}) (MCP42XXX devices only)

The Shutdown pin has a Schmitt Trigger input. Pulling this pin low will put the device in a power-saving mode where A terminal is opened and the B and W terminals are connected for all potentiometers. This pin should not be toggled low when the \overline{CS} pin is low. In order to minimize power consumption, this pin has an active pull-up circuit. The performance of this circuit is shown in Figure 2-12. This pin will draw negligible current at logic level '0' and logic level '1'. Do not leave this pin floating.

TABLE 3-1: MCP41XXX Pins

Pin #	Name	Function
1	\overline{CS}	Chip Select
2	SCK	Serial Clock
3	SI	Serial Data Input
4	V_{SS}	Ground
5	PA0	Terminal A Connection For Pot 0
6	PW0	Wiper Connection For Pot 0
7	PB0	Terminal B Connection For Pot 0
8	V_{DD}	Power

TABLE 3-2: MCP42XXX Pins

Pin #	Name	Function
1	\overline{CS}	Chip Select
2	SCK	Serial Clock
3	SI	Serial Data Input
4	V_{SS}	Ground
5	PB1	Terminal B Connection For Pot 1
6	PW1	Wiper Connection For Pot 1
7	PA1	Terminal A Connection For Pot 1
8	PA0	Terminal A Connection For Pot 0
9	PW0	Wiper Connection For Pot 0
10	PB0	Terminal B Connection For Pot 0
11	\overline{RS}	Reset Input
12	\overline{SHDN}	Shutdown Input
13	SO	Data Out for Daisy-Chaining
14	V_{DD}	Power

4.0 APPLICATIONS INFORMATION

The MCP41XXX/42XXX devices are 256 position single and dual digital potentiometers that can be used in place of standard mechanical pots. Resistance values of 10 k Ω , 50 k Ω and 100 k Ω are available. As shown in Figure 4-1, each potentiometer is made up of a variable resistor and an 8-bit (256 position) data register that determines the wiper position. There is a nominal wiper resistance of 52 Ω for the 10 k Ω version, 125 Ω for the 50 k Ω and 100 k Ω versions. For the dual devices, the channel-to-channel matching variation is less than 1%. The resistance between the wiper and either of the resistor endpoints varies linearly according to the value stored in the data register. Code 00h effectively connects the wiper to the B terminal. At

power-up, all data registers will automatically be loaded with the mid-scale value (80h). The serial interface provides the means for loading data into the shift register, which is then transferred to the data registers. The serial interface also provides the means to place individual potentiometers in the shutdown mode for maximum power savings. The SHDN pin can also be used to put all potentiometers in shutdown mode and the RS pin is provided to set all potentiometers to mid-scale (80h).

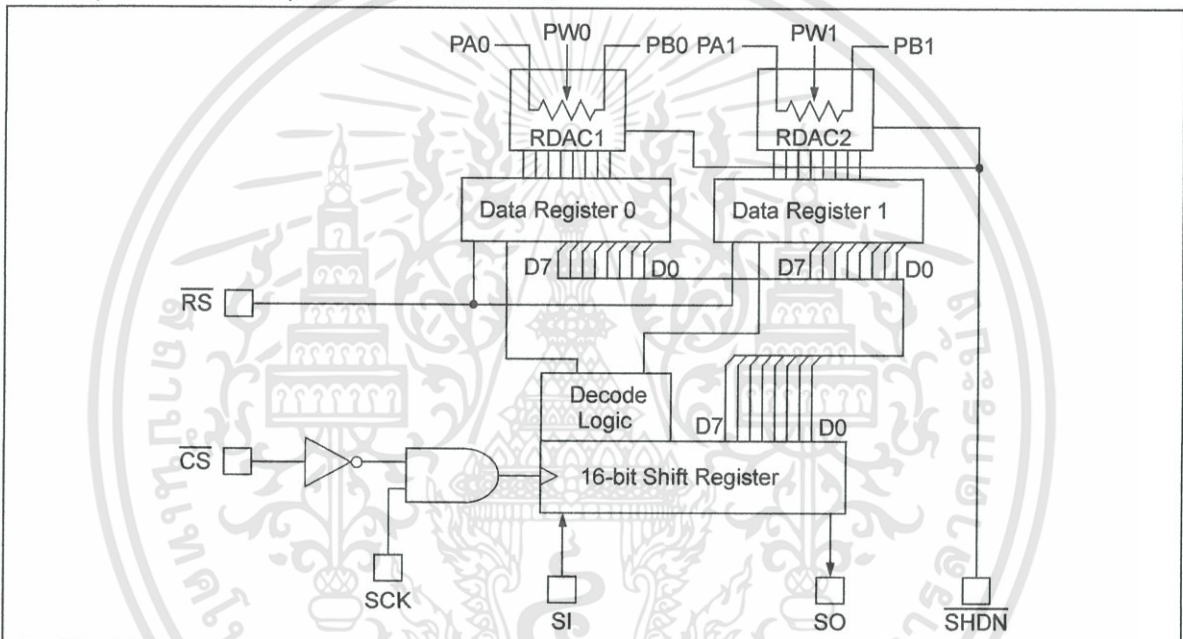
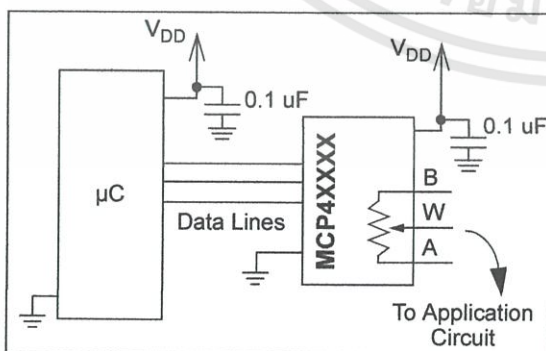


FIGURE 4-1: Block diagram showing the MCP42XXX dual digital potentiometer. Data register 0 and data register 1 are 8-bit registers allowing 256 positions for each wiper. Standard SPI pins are used with the addition of the Shutdown (SHDN) and Reset (RS) pins. As shown, reset affects the data register and wipers, bringing them to mid-scale. Shutdown disconnects the A terminal and connects the wiper to B, without changing the state of the data registers.



When laying out the circuit for your digital potentiometer, bypass capacitors should be used. These capacitors should be placed as close as possible to the device pin. A bypass capacitor value of 0.1 μF is recommended. Digital and analog traces should be separated as much as possible on the board, with no traces running underneath the device or the bypass capacitor. Extra precautions should be taken to keep traces with high-frequency signals (such as clock lines) as far as possible from analog traces. Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board.

MCP41XXX/42XXX

4.1 Modes of Operation

Digital potentiometer applications can be divided into two categories: rheostat mode and potentiometer, or voltage divider, mode.

4.1.1 RHEOSTAT MODE

In the rheostat mode, the potentiometer is used as a two-terminal resistive element. The unused terminal should be tied to the wiper, as shown in Figure 4-2. Note that reversing the polarity of the A and B terminals will not affect operation.

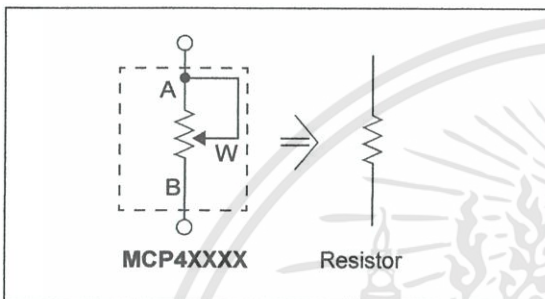


FIGURE 4-2: Two-terminal or rheostat configuration for the digital potentiometer. Acting as a resistive element in the circuit, resistance is controlled by changing the wiper setting.

Using the device in this mode allows control of the total resistance between the two nodes. The total measured resistance would be the least at code 00h, where the wiper is tied to the B terminal. The resistance at this code is equal to the wiper resistance, typically 52Ω for the 10 kΩ MCP4X010 devices, 125Ω for the 50 kΩ (MCP4X050), and 100 kΩ (MCP4X100) devices. For the 10 kΩ device, the LSB size would be 39.0625Ω (assuming 10 kΩ total resistance). The resistance would then increase with this LSB size until the total measured resistance at code FFh would be 9985.94Ω. The wiper will never directly connect to the A terminal of the resistor stack.

In the 00h state, the total resistance is the wiper resistance. To avoid damage to the internal wiper circuitry in this configuration, care should be taken to ensure the current flow never exceeds 1 mA.

For dual devices, the variation of channel-to-channel matching of the total resistance from A to B is less than 1%. The device-to-device matching, however, can vary up to 30%. In the rheostat mode, the resistance has a positive temperature coefficient. The change in wiper-to-end terminal resistance over temperature is shown in Figure 2-8. The most variation over temperature will occur in the first 6% of codes (code 00h to 0Fh) due to the wiper resistance coefficient affecting the total resistance. The remaining codes are dominated by the total resistance tempco R_{AB} , typically 800 ppm/°C.

4.1.2 POTENTIOMETER MODE

In the potentiometer mode, all three terminals of the device are tied to different nodes in the circuit. This allows the potentiometer to output a voltage proportional to the input voltage. This mode is sometimes called voltage divider mode. The potentiometer is used to provide a variable voltage by adjusting the wiper position between the two endpoints as shown in Figure 4-3. Note that reversing the polarity of the A and B terminals will not affect operation.

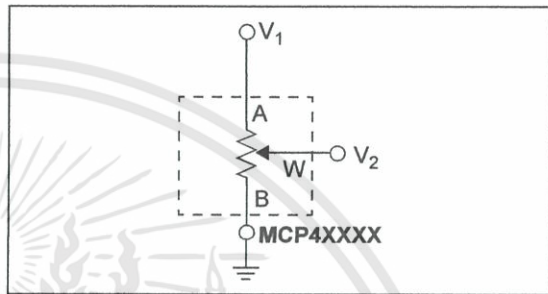


FIGURE 4-3: Three terminal or voltage divider mode.

In this configuration, the ratio of the internal resistance defines the temperature coefficient of the device. The resistor matching of the R_{WB} resistor to the R_{AB} resistor performs with a typical temperature coefficient of 1 ppm/°C (measured at code 80h). At lower codes, the wiper resistance temperature coefficient will dominate. Figure 2-3 shows the effect of the wiper. Above the lower codes, this figure shows that 70% of the states will typically have a temperature coefficient of less than 5 ppm/°C. 30% of the states will typically have a ppm/°C of less than 1.

4.2 Typical Applications

4.2.1 PROGRAMMABLE SINGLE-ENDED AMPLIFIERS

Potentiometers are often used to adjust system reference levels or gain. Programmable gain circuits using digital potentiometers can be realized in a number of different ways. An example of a single-supply, inverting gain amplifier is shown in Figure 4-4. Due to the high input impedance of the amplifier, the wiper resistance is not included in the transfer function. For a single-supply, non-inverting gain configuration, the circuit in Figure 4-5 can be used.

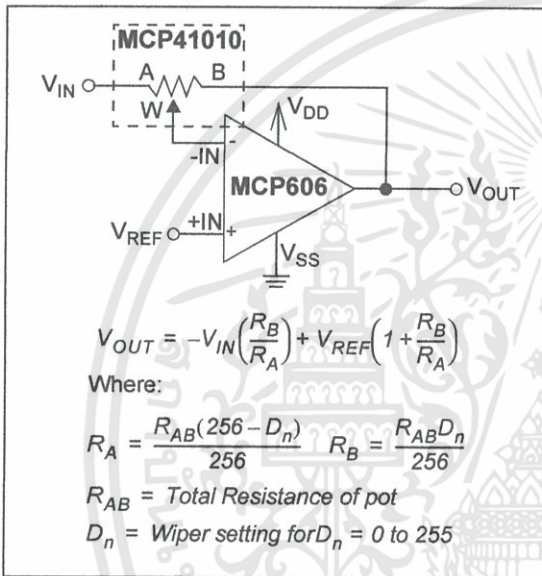


FIGURE 4-4: Single-supply, programmable, inverting gain amplifier using a digital potentiometer.

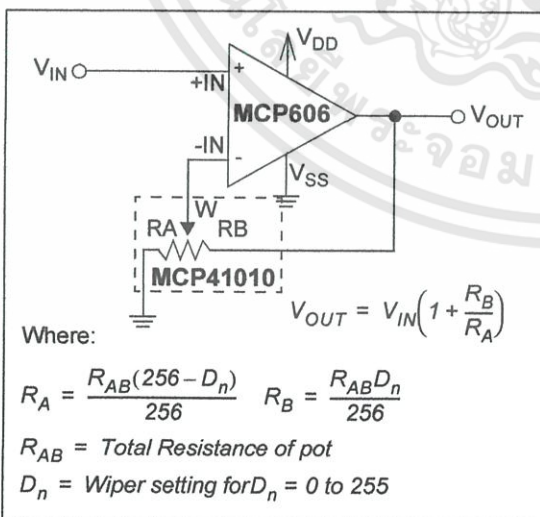


FIGURE 4-5: Single-supply, programmable, non-inverting gain amplifier.

In order for these circuits to work properly, care must be taken in a few areas. For linear operation, the analog input and output signals must be in the range of V_{SS} to V_{DD} for the potentiometer and input and output rails of the op-amp. The circuit in Figure 4-4 requires a virtual ground or reference input to the non-inverting input of the amplifier. Refer to Application Note 682, "Using Single-Supply Operational Amplifiers in Embedded Systems" (DS00682), for more details. At power-up or reset (\overline{RS}), the resistance is set to mid-scale, with R_A and R_B matching. Based on the transfer function for the circuit, the gain is -1 V/V. As the code is increased and the wiper moves towards the A terminal, the gain increases. Conversely, when the wiper is moved towards the B terminal, the gain decreases. Figure 4-6 shows this relationship. Notice the pseudo-logarithmic gain around decimal code 128. As the wiper approaches either terminal, the step size in the gain calculation increases dramatically. Due to the mismatched ratio of R_A and R_B at the extreme high and low codes, small increments in wiper position can dramatically affect the gain. As shown in Figure 4-3, recommended gains lie between 0.1 and 10 V/V.

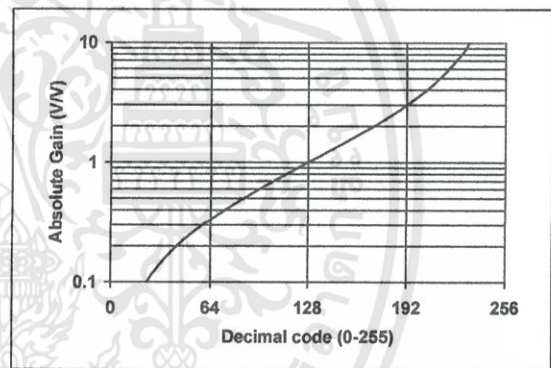


FIGURE 4-6: Gain vs. Code for inverting and differential amplifier circuits.

4.2.2 PROGRAMMABLE DIFFERENTIAL AMPLIFIER

An example of a differential input amplifier using digital potentiometers is shown in Figure 4-7. For the transfer function to hold, both pots must be programmed to the same code. The resistor-matching from channel-to-channel within a dual device can be used as an advantage in this circuit. This circuit will also show stable operation over temperature due to the low potentiometer temperature coefficient. Figure 4-6 also shows the relationship between gain and code for this circuit. As the wiper approaches either terminal, the step size in the gain calculation increases dramatically. This circuit is recommended for gains between 0.1 and 10 V/V.

MCP41XXX/42XXX

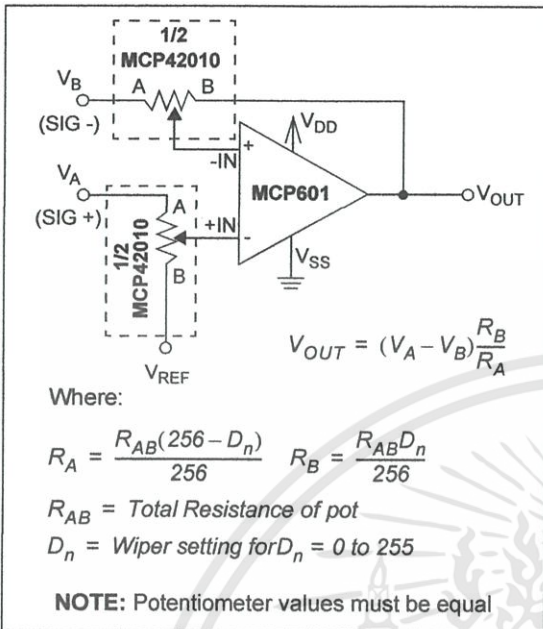


FIGURE 4-7: Single Supply programmable differential amplifier using digital potentiometers.

4.2.3 PROGRAMMABLE OFFSET TRIM

For applications requiring only a programmable voltage reference, the circuit in Figure 4-8 can be used. This circuit shows the device used in the potentiometer mode along with two resistors and a buffered output. This creates a circuit with a linear relationship between voltage-out and programmed code. Resistors R_1 and R_2 can be used to increase or decrease the output voltage step size. The potentiometer in this mode is stable over temperature. The operation of this circuit over temperature is shown in Figure 2-3. The worst performance over temperature will occur at the lower codes due to the dominating wiper resistance. R_1 and R_2 can also be used to affect the boundary voltages, thereby eliminating the use of these lower codes.

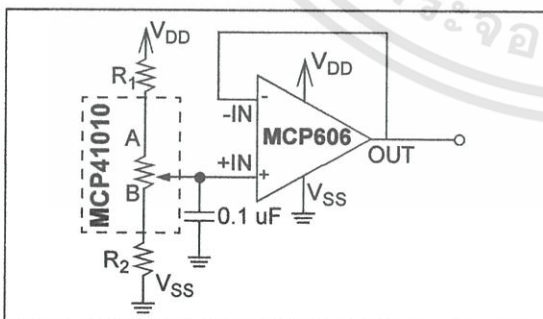


FIGURE 4-8: By changing the values of R_1 and R_2 , the voltage output resolution of this programmable voltage reference circuit is affected.

4.3 Calculating Resistances

When programming the digital potentiometer settings, the following equations can be used to calculate the resistances. Programming code 00h effectively brings the wiper to the B terminal, leaving only the wiper resistance. Programming higher codes will bring the wiper closer to the A terminal of the potentiometer. The equations in Figure 4-9 can be used to calculate the terminal resistances. Figure 4-10 shows an example calculation using a 10 k Ω potentiometer.

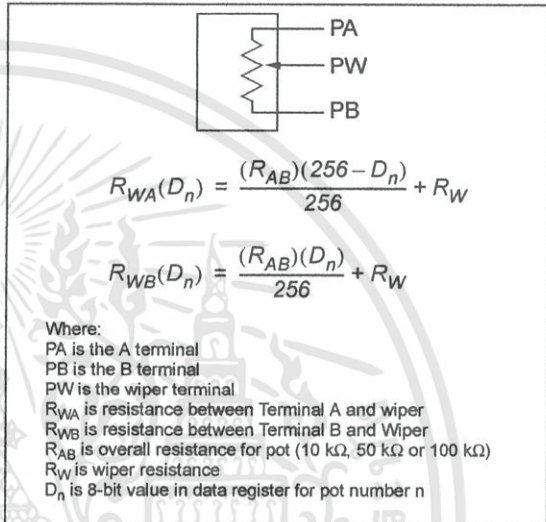


FIGURE 4-9: Potentiometer resistances are a function of code. It should be noted that, when using these equations for most feedback amplifier circuits (see Figure 4-4 and Figure 4-5), the wiper resistance can be omitted due to the high impedance input of the amplifier.

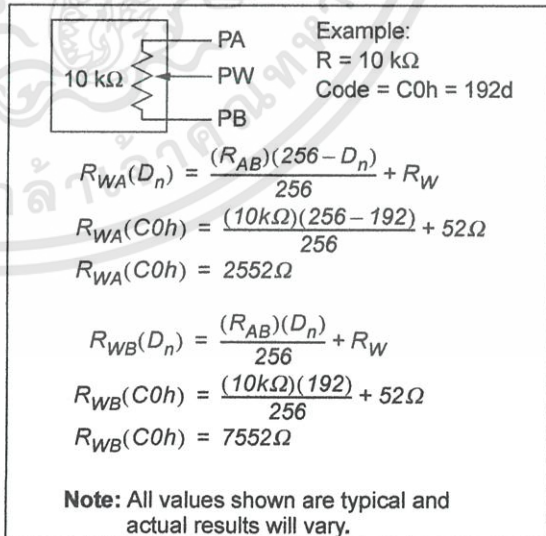


FIGURE 4-10: Example Resistance calculations.

5.0 SERIAL INTERFACE

Communications from the controller to the MCP41XXX/42XXX digital potentiometers is accomplished using the SPI serial interface. This interface allows three commands:

1. Write a new value to the potentiometer data register(s).
2. Cause a channel to enter low power shutdown mode.
3. NOP (No Operation) command.

Executing any command is accomplished by setting \overline{CS} low and then clocking-in a command byte followed by a data byte into the 16-bit shift register. The command is executed when \overline{CS} is raised. Data is clocked-in on the rising edge of clock and out the \overline{SO} pin on the falling edge of the clock (see Figure 5-1). The device will track the number of clocks (rising edges) while \overline{CS} is low and will abort all commands if the number of clocks is not a multiple of 16.

5.1 Command Byte

The first byte sent is always the command byte, followed by the data byte. The command byte contains two command select bits and two potentiometer select bits. Unused bits are 'don't care' bits. The command select bits are summarized in Figure 5-2. The command select bits C1 and C0 (bits 4:5) of the command byte determine which command will be executed. If the command bits are both 0's or 1's, then a NOP command will be executed once all 16 bits have been loaded. This command is useful when using the daisy-chain configuration. When the command bits are 0,1, a write command will be executed with the 8 bits sent in the data byte. The data will be written to the potentiometer(s) determined by the potentiometer select bits. If the command bits are 1,0, then a shutdown command will be executed on the potentiometers determined by the potentiometer select bits.

For the MCP42XXX devices, the potentiometer select bits P1 and P0 (bits 0:1) determine which potentiometers are to be acted upon by the command. A corresponding '1' in the position signifies that the command for that potentiometer will get executed, while a '0' signifies that the command will not effect that potentiometer (see Figure 5-2).

5.2 Writing Data Into Data Registers

When new data is written into one or more of the potentiometer data registers, the write command is followed by the data byte for the new value. The command select bits C1, C0 are set to 0,1. The potentiometer selection bits P1 and P0 allow new values to be written to potentiometer 0, potentiometer 1 (or both) with a single command. A '1' for either P1 or P0 will cause the data to be written to the respective data register and a '0' for P1 or P0 will cause no change. See Figure 5-2 for the command format summary.

5.3 Using The Shutdown Command

The shutdown command allows the user to put the application circuit into a power-saving mode. In this mode, the A terminal is open-circuited and the B and W terminals are shorted together. The command select bits C1, C0 are set to 1,0. The potentiometer selection bits P1 and P0 allow each potentiometer to be shutdown independently. If either P1 or P0 are high, the respective potentiometer will enter shutdown mode. A '0' for P1 or P0 will have no effect. The eight data bits following the command byte still need to be transmitted for the shutdown command, but they are 'don't care' bits. See Figure 5-2 for command format summary. Once a particular potentiometer has entered the shutdown mode, it will remain in this mode until:

- A new value is written to the potentiometer data register, provided that the \overline{SHDN} pin is high. The device will remain in the shutdown mode until the rising edge of the \overline{CS} is detected, at which time the device will come out of shutdown mode and the new value will be written to the data register(s). If the \overline{SHDN} pin is low when the new value is received, the registers will still be set to the new value, but the device will remain in shutdown mode. This scenario assumes that a valid command was received. If an invalid command was received, the command will be ignored and the device will remain in the shutdown mode.

It is also possible to use the hardware shutdown pin and reset pin to remove a device from software shutdown. To do this, a low pulse on the chip select line must first be sent. For multiple devices, sharing a single \overline{SHDN} or \overline{RESET} line allows you to pick an individual device on that chain to remove from software shutdown mode. See Figure 1-3 for timing. With a preceding chip select pulse, either of these situations will also remove a device from software shutdown:

- A falling edge is seen on the \overline{RS} pin and held low for at least 150 ns, provided that the \overline{SHDN} pin is high. If the \overline{SHDN} pin is low, the registers will still be set to mid-scale, but the device will remain in shutdown mode. This condition assumes that \overline{CS} is high, as bringing the \overline{RS} pin low while \overline{CS} is low is an invalid state and results are indeterminate.
- A rising edge on the \overline{SHDN} pin is seen after being low for at least 100 ns, provided that the \overline{CS} pin is high. Toggling the \overline{SHDN} pin low while \overline{CS} is low is an invalid state and results are indeterminate.
- The device is powered-down and back up.

Note: The hardware \overline{SHDN} pin will always put the device in shutdown regardless of whether a potentiometer has already been put in the shutdown mode using the software command.

MCP41XXX/42XXX

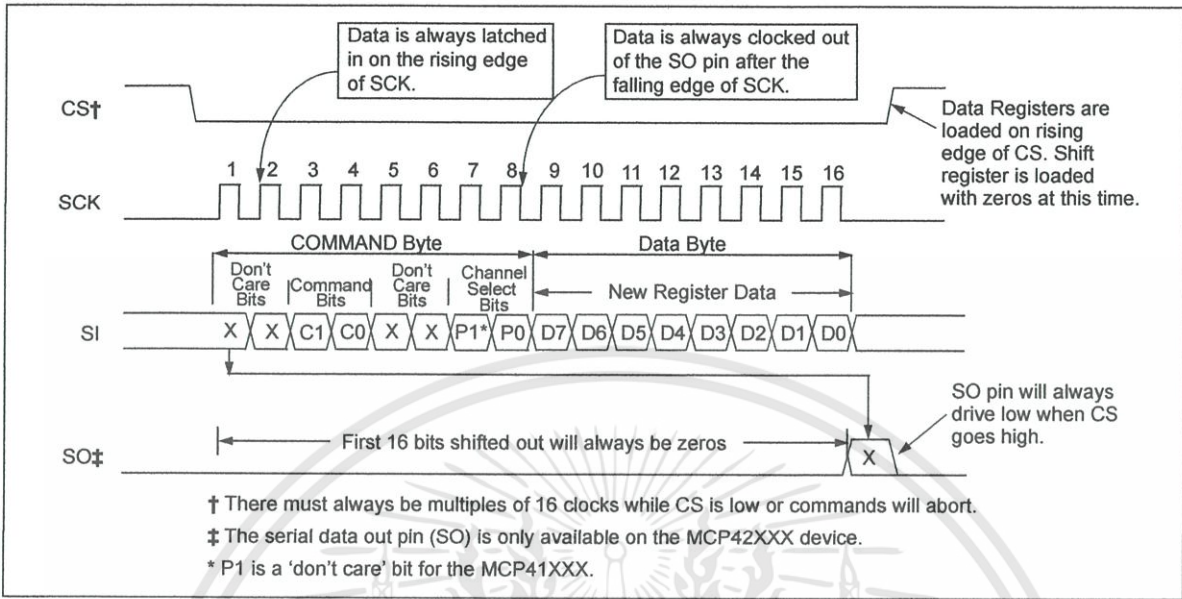


FIGURE 5-1: Timing Diagram for Writing Instructions or Data to a Digital Potentiometer.

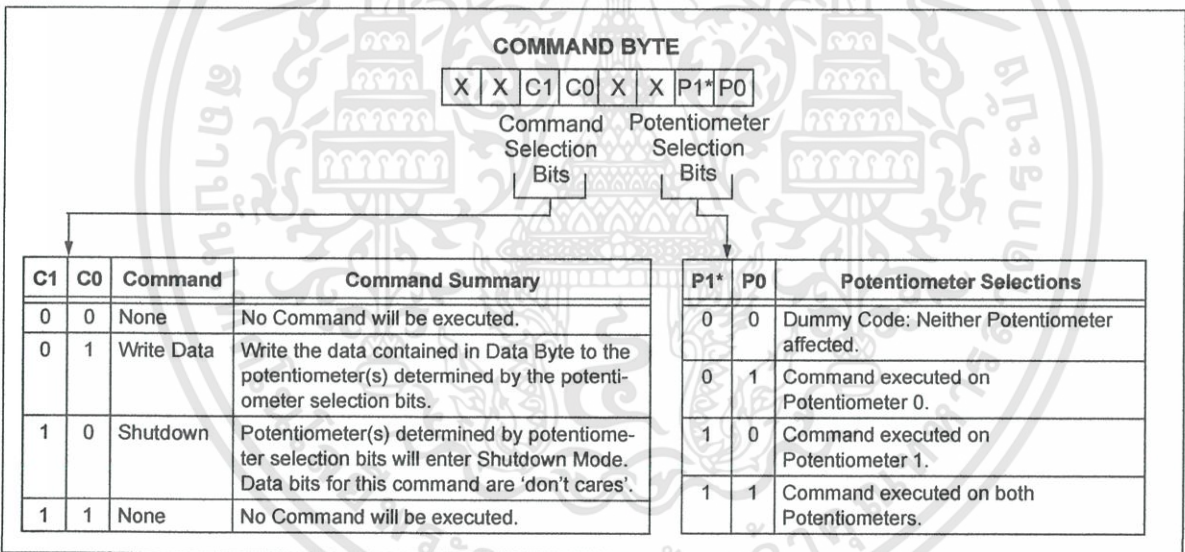


FIGURE 5-2: Command Byte Format.

5.4 Daisy-Chain Configuration

Multiple MCP42XXX devices can be connected in a daisy-chain configuration, as shown in Figure 5-4, by connecting the SO pin from one device to the SI pin on the next device. The data on the SO pin is the output of the 16-bit shift register. The daisy-chain configuration allows the system designer to communicate with several devices without using a separate CS line for each device. The example shows a daisy-chain configuration with three devices, although any number of devices (with or without the same resistor values) can be configured this way. While it is not possible to use a MCP41XXX at the beginning or middle of a daisy-chain (because it does not provide the serial data out (SO) pin), it is possible to use the device at the end of a chain. As shown in the timing diagram in Figure 5-3, data will be clocked-out of the SO pin on the falling edge of the clock. The SO pin has a CMOS push-pull output and will drive low when CS goes high. SO will not go to a high-impedance state when CS is held high.

When using the daisy-chain configuration, the maximum clock speed possible is reduced to ~5.8 MHz, because of the propagation delay of the data coming out of the SO pin.

When using the daisy-chain configuration, keep in mind that the shift register of each device is automatically loaded with zeros whenever a command is executed (\overline{CS} = high). Because of this, the first 16 bits that come out of the SO pin once the CS line goes low will always be zeros. This means that when the first command is being loaded into a device, it will always shift a NOP command into the next device on the chain because the command bits (and all the other bits) will be zeros. This feature makes it necessary only to send command and data bytes to the device farthest down the chain that needs a new command. For example, if there were three devices on the chain and it was desired to send a command to the device in the middle, only 32 bytes of data need to be transmitted. The last device on the chain will have a NOP loaded from the previous device so no registers will be affected when the CS pin is raised to execute the command. **The user must always ensure that multiples of 16 clocks are always provided (while CS is low), as all commands will abort if the number of clocks provided is not a multiple of 16.**

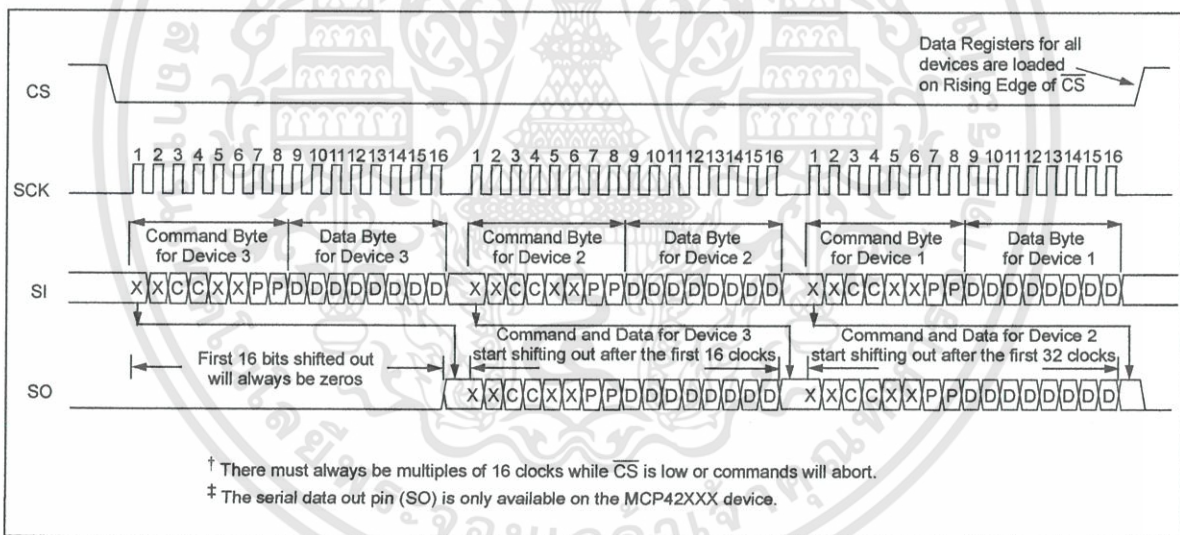


FIGURE 5-3: Timing Diagram for Daisy-Chain Configuration.

MCP41XXX/42XXX

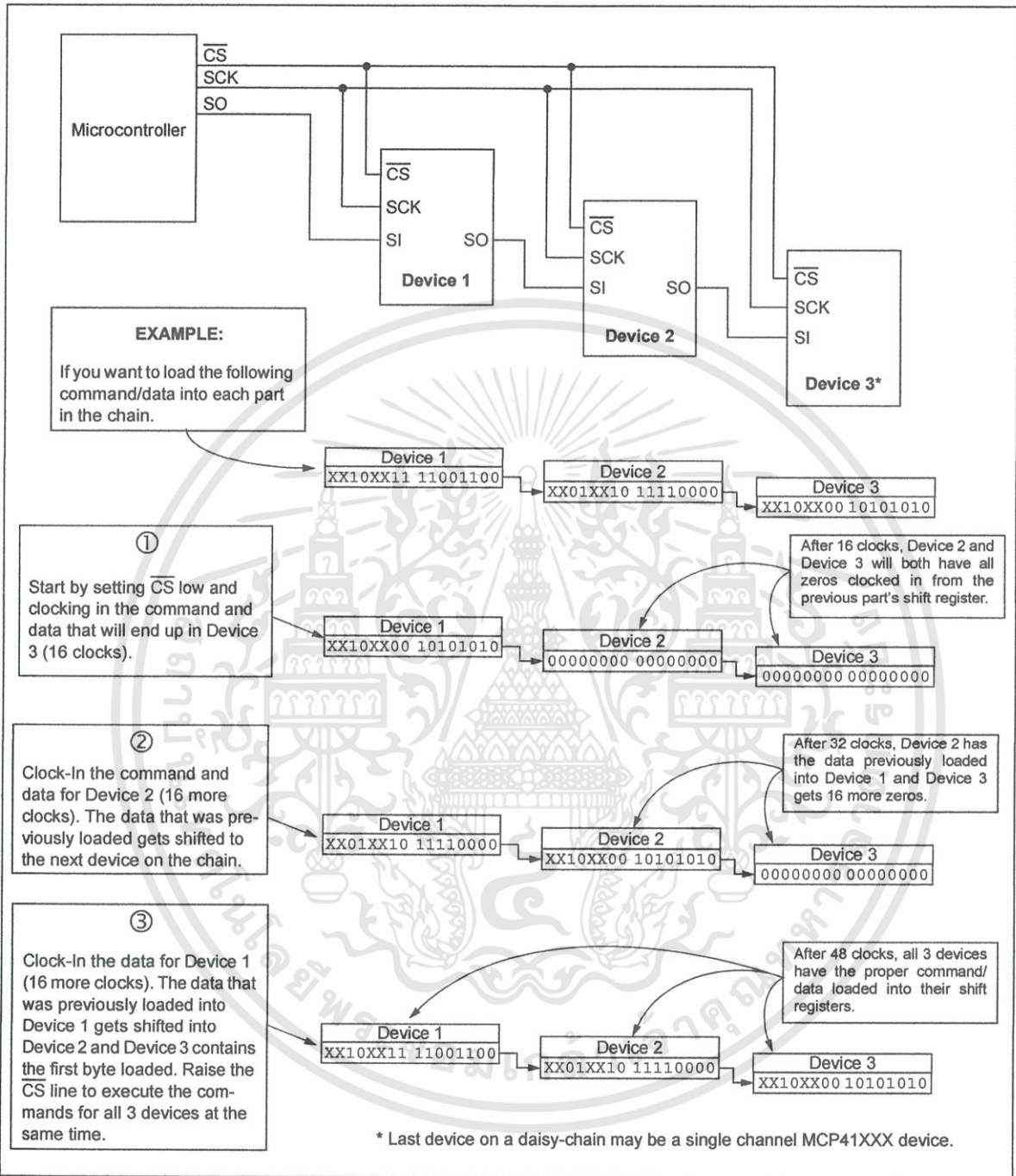


FIGURE 5-4: Daisy-Chain Configuration.

5.5 Reset (\overline{RS}) Pin Operation

The Reset pin (\overline{RS}) will automatically set all potentiometer data latches to mid-scale (Code 80h) when pulled low (provided that the pin is held low at least 150 ns and \overline{CS} is high). The reset will execute regardless of the position of the SCK, \overline{SHDN} and SI pins. It is possible to toggle \overline{RS} low and back high while \overline{SHDN} is low. In this case, the potentiometer registers will reset to mid-scale, but the potentiometer will remain in shutdown mode until the \overline{SHDN} pin is raised.

Note: Bringing the \overline{RS} pin low while the \overline{CS} pin is low constitutes an invalid operating state and will result in indeterminate results when \overline{RS} and/or \overline{CS} are brought high.

5.6 Shutdown (\overline{SHDN}) Pin Operation

When held low, the shutdown pin causes the application circuit to go into a power-saving mode by open-circuiting the A terminal and shorting the B and W terminals for all potentiometers. Data register contents are not affected by entering shutdown mode (i.e., when the \overline{SHDN} pin is raised, the data register contents are the same as before the shutdown mode was entered).

While in shutdown mode, it is still possible to clock in new values for the data registers, as well as toggling the \overline{RS} pin to cause all data registers to go to mid-scale. The new values will take affect when the \overline{SHDN} pin is raised.

If the device is powered-up with the \overline{SHDN} pin held low, it will power-up in the shutdown mode with the data registers set to mid-scale.

Note: Bringing the \overline{SHDN} pin low while the \overline{CS} pin is low constitutes an invalid operating state and will result in indeterminate results when \overline{SHDN} and/or \overline{CS} are brought high.

5.7 Power-up Considerations

When the device is powered on, the data registers will be set to mid-scale (80h). A power-on reset circuit is utilized to ensure that the device powers up in this known state.

TABLE 5-1: TRUTH TABLE FOR LOGIC INPUTS

SCK	\overline{CS}	\overline{RS}	\overline{SHDN}	Action
X	\emptyset	H	H	Communication is initiated with device. Device comes out of standby mode.
L	L	H	H	No action. Device is waiting for data to be clocked into shift register or \overline{CS} to go high to execute command.
1	L	H	X	Shift one bit into shift register. The shift register can be loaded while the \overline{SHDN} pin is low.
\emptyset	L	H	X	Shift one bit out of shift register on the SO pin. The SO pin is active while the \overline{SHDN} pin is low.
X	1	H	H	Based on command bits, either load data from shift register into data latches or execute shutdown command. Neither command executed unless multiples of 16 clocks have been entered while \overline{CS} is low. SO pin goes to a logic low.
X	H	H	H	Static Operation.
X	H	\emptyset	H	All data registers set and latched to code 80h.
X	H	\emptyset	L	All data registers set and latched to code 80h. Device is in hardware shutdown mode and will remain in this mode.
X	H	H	\emptyset	All potentiometers put into hardware shutdown mode; terminal A is open and W is shorted to B.
X	H	H	1	All potentiometers exit hardware shutdown mode. Potentiometers will also exit software shutdown mode if this rising edge occurs after a low pulse on \overline{CS} . Contents of data latches are restored.

MCP41XXX/42XXX

5.8 Using the MCP41XXX/42XXX in SPI Mode 1,1

It is possible to operate the devices in SPI modes 0,0 and 1,1. The only difference between these two modes is that, when using mode 1,1, the clock idles in the high state, while in mode 0,0, the clock idles in the low state. In both modes, data is clocked into the devices on the rising edge of SCK and data is clocked out the SO pin once the falling edge of SCK. Operations using mode 0,0 are shown in Figure 5-1. The example in Figure 5-5 shows mode 1,1.

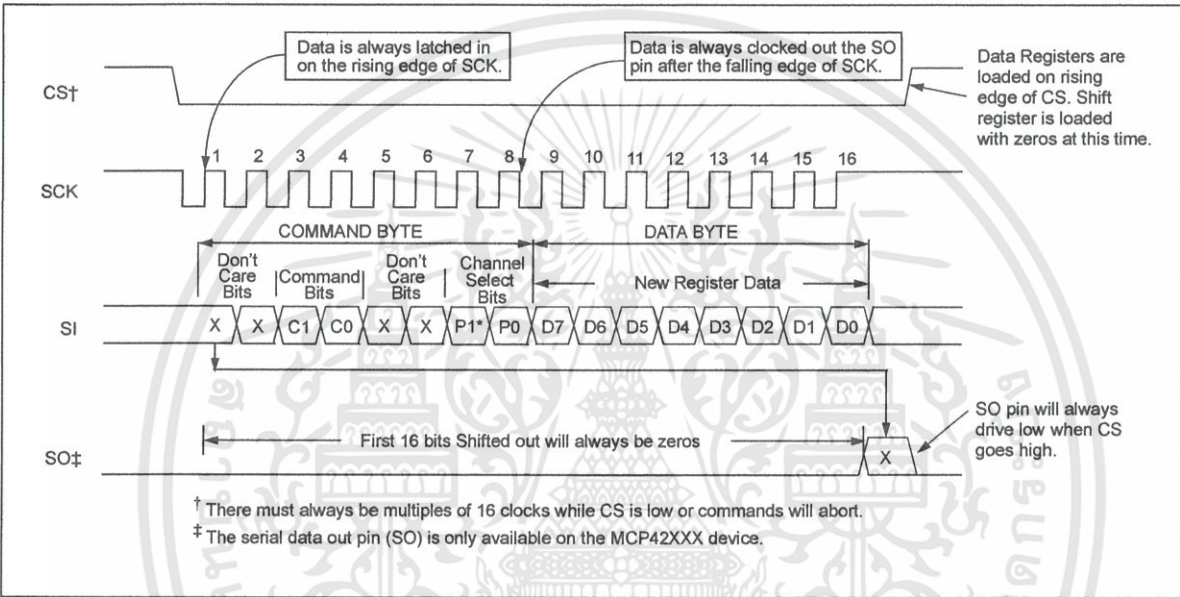
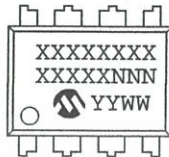


FIGURE 5-5: Timing Diagram for SPI Mode 1,1 Operation.

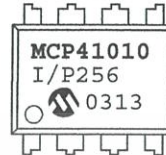
6.0 PACKAGING INFORMATION

6.1 Package Marking Information

8-Lead PDIP (300 mil)



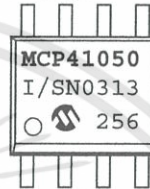
Example:



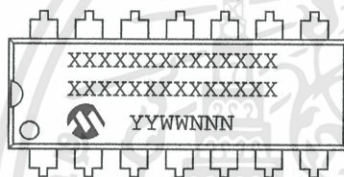
8-Lead SOIC (150 mil)



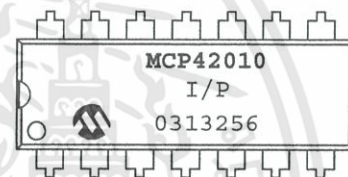
Example:



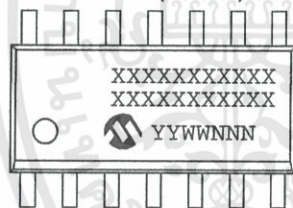
14-Lead PDIP (300 mil)



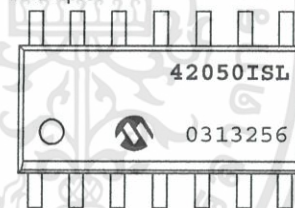
Example:



14-Lead SOIC (150 mil)



Example:



14-Lead TSSOP (4.4mm) *



Example:



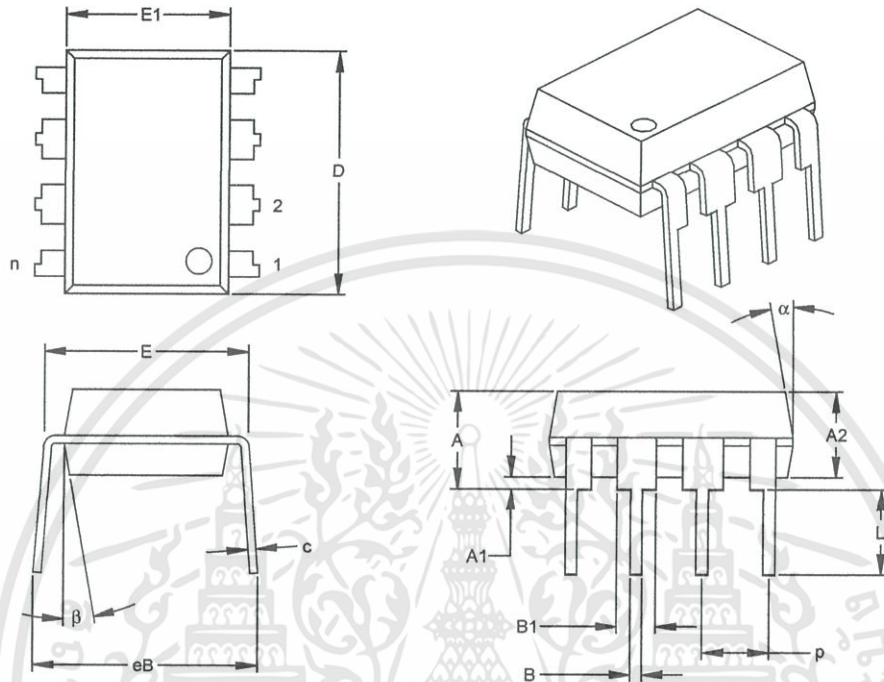
Legend: XX...X Customer specific information*
 YY Year code (last 2 digits of calendar year)
 WW Week code (week of January 1 is week '01')
 NNN Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code.

MCP41XXX/42XXX

8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



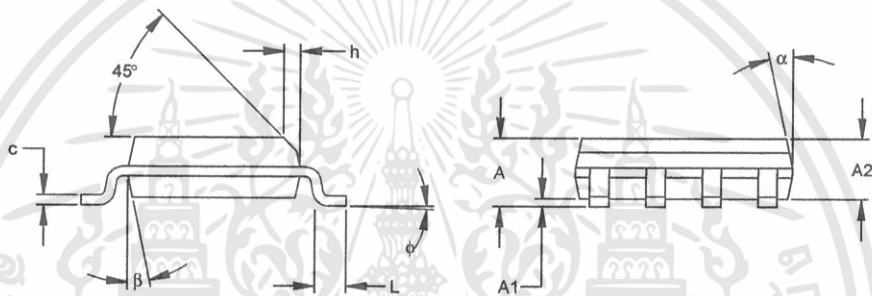
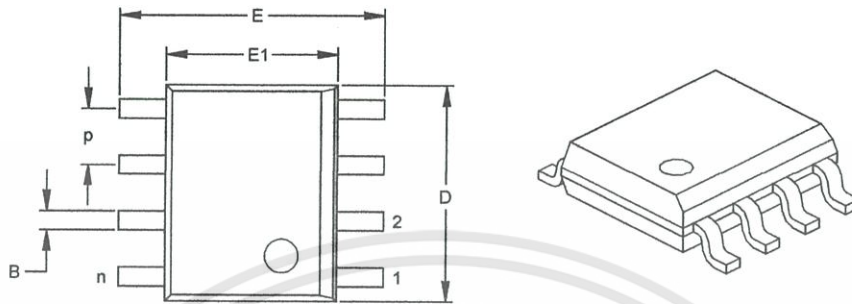
Dimension	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
 § Significant Characteristic

Notes:
 Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
 JEDEC Equivalent: MS-001
 Drawing No. C04-018

MCP41XXX/42XXX

8-Lead Plastic Small Outline (SN) – Narrow, 150 mil (SOIC)



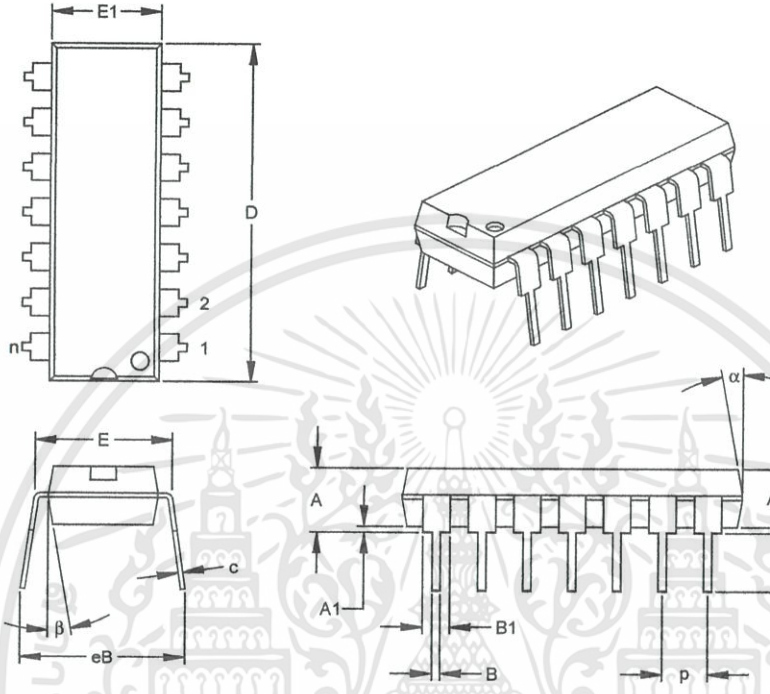
Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	P		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	0.10	0.18	0.25
Overall Width	E	.228	.237	.244	5.79	6.02	6.20
Molded Package Width	E1	.146	.154	.157	3.71	3.91	3.99
Overall Length	D	.189	.193	.197	4.80	4.90	5.00
Chamfer Distance	h	.010	.015	.020	0.25	0.38	0.51
Foot Length	L	.019	.025	.030	0.48	0.62	0.76
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.013	.017	.020	0.33	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MS-012
Drawing No. C04-057

MCP41XXX/42XXX

14-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.740	.750	.760	18.80	19.05	19.30
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter

§ Significant Characteristic

Notes:

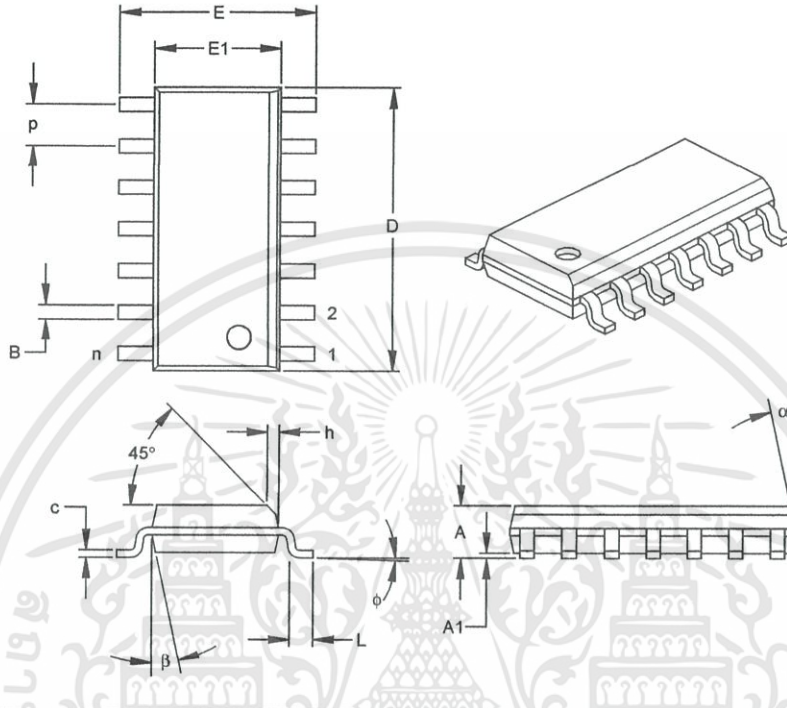
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-005

MCP41XXX/42XXX

14-Lead Plastic Small Outline (SL) – Narrow, 150 mil (SOIC)



Dimension	Units	INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	0.10	0.18	0.25
Overall Width	E	.228	.236	.244	5.79	5.99	6.20
Molded Package Width	E1	.150	.154	.157	3.81	3.90	3.99
Overall Length	D	.337	.342	.347	8.56	8.69	8.81
Chamfer Distance	h	.010	.015	.020	0.25	0.38	0.51
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
 § Significant Characteristic

Notes:

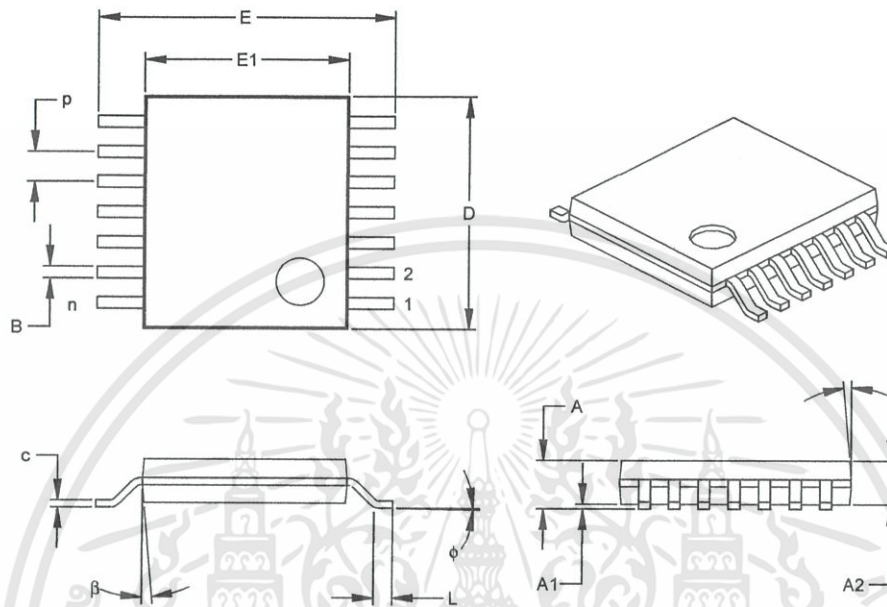
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-012

Drawing No. C04-065

MCP41XXX/42XXX

14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP)



Dimension	Units	INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		14			14	
Pitch	p		.026			0.65	
Overall Height	A			.043			1.10
Molded Package Thickness	A2	.033	.035	.037	0.85	0.90	0.95
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Overall Width	E	.246	.251	.256	6.25	6.38	6.50
Molded Package Width	E1	.169	.173	.177	4.30	4.40	4.50
Molded Package Length	D	.193	.197	.201	4.90	5.00	5.10
Foot Length	L	.020	.024	.028	0.50	0.60	0.70
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.007	.010	.012	0.19	0.25	0.30
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005" (0.127mm) per side.

JEDEC Equivalent: MO-153

Drawing No. C04-087

MCP41XXX/42XXX

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX		
Device	Temperature Range	Package		
<p>Device:</p> <p>MCP41010: Single Digital Potentiometer (10 kΩ) MCP41010T: Single Digital Potentiometer (10 kΩ) (Tape and Reel) MCP41050: Single Digital Potentiometer (50 kΩ) (Tape and Reel) MCP41050T: Single Digital Potentiometer (50 kΩ) (Tape and Reel) MCP41100: Single Digital Potentiometer (100 kΩ) (Tape and Reel) MCP41100T: Single Digital Potentiometer (100 kΩ) (Tape and Reel)</p> <p>MCP42010: Dual Digital Potentiometer (10 kΩ) MCP42010T: Dual Digital Potentiometer (10 kΩ) (Tape and Reel) MCP42050: Dual Digital Potentiometer (50 kΩ) (Tape and Reel) MCP42050T: Dual Digital Potentiometer (50 kΩ) (Tape and Reel) MCP42100: Dual Digital Potentiometer (100 kΩ) (Tape and Reel) MCP42100T: Dual Digital Potentiometer (100 kΩ) (Tape and Reel)</p> <p>Temperature Range:</p> <p>I = -40°C to +85°C E = -40°C to +125°C</p> <p>Package:</p> <p>P = Plastic DIP (300 mil Body), 8-lead, 14-lead SN = Plastic SOIC (150 mil Body), 8-lead SL = Plastic SOIC (150 mil Body), 14-lead ST = TSSOP (4.4mm Body), 14-lead</p>				
			Examples:	
			a) MCP41010-I/SN: I-Temp., 8LD SOIC pkg.	
			b) MCP41010-E/P: E-Temp., 8LD PDIP pkg.	
			c) MCP41010T-I/SN: Tape and Reel, I-Temp., 8LD SOIC pkg.	
			d) MCP41050-E/SN: E-Temp., 8LD SOIC pkg.	
			e) MCP41050-I/P: I-Temp., 8LD PDIP pkg.	
			f) MCP41050-E/SN: E-Temp., 8LD SOIC pkg.	
			g) MCP41100-I/SN: I-Temp., 8LD SOIC package.	
			h) MCP41100-E/P: E-Temp., 8LD PDIP pkg.	
			i) MCP41100T-I/SN: I-Temp., 8LD SOIC pkg.	
			a) MCP42010-E/P: E-Temp., 14LD PDIP pkg.	
			b) MCP42010-I/SL: I-Temp., 14LD SOIC pkg.	
			c) MCP42010-E/ST: E-Temp., 14LD TSSOP pkg.	
			d) MCP42010T-I/ST: Tape and Reel, I-Temp., 14LD TSSOP pkg.	
			e) MCP42050-E/P: E-Temp., 14LD PDIP pkg.	
			f) MCP42050T-I/SL: Tape and Reel, I-Temp., 14LD SOIC pkg.	
			g) MCP42050-E/SL: E-Temp., 14LD SOIC pkg.	
			h) MCP42050-I/ST: I-Temp., 14LD TSSOP pkg.	
			i) MCP42050T-I/SL: Tape and Reel, I-Temp., 14LD SOIC pkg.	
			j) MCP42050T-I/ST: Tape and Reel, I-Temp., 14LD TSSOP pkg.	
			k) MCP42100-E/P: E-Temp., 14LD PDIP pkg.	
			l) MCP42100-I/SL: I-Temp., 14LD SOIC pkg.	
			m) MCP42100-E/ST: E-Temp., 14LD TSSOP pkg.	
			n) MCP42100T-I/SL: Tape and Reel, I-Temp., 14LD SOIC pkg.	
			o) MCP42100T-I/ST: Tape and Reel, I-Temp., 14LD TSSOP pkg.	

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCP41XXX/42XXX

NOTES:



Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELoQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

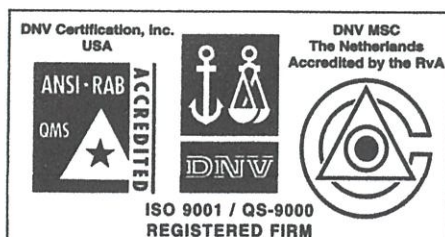
Accuron, Application Maestro, dsPICDEM, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICKit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: <http://www.microchip.com>

Atlanta

3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

Phoenix

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966
Fax: 480-792-4338

San Jose

2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950
Fax: 408-436-7955

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia

Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu

Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou

Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR

Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai

Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen

Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde

Room 401, Hongjian Building
No. 2 Fengxiangnan Road, Ronggui Town
Shunde City, Guangdong 528303, China
Tel: 86-765-8395507 Fax: 86-765-8395571

China - Qingdao

Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India

Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan

Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea

168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore

200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan

Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan

Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria

Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark

Regus Business Centre
Lautrup høj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France

Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - 1er Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany

Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy

Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands

P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom

505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/28/03

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้