

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตรวจจับความเร็ววัตถุ

OBJECT VELOCITY DETECTOR



T111793



นายพงศกร ประพันธ์ไพโรจน์
นายภักดีไพท จอกน้อย

เลขหมู่.....
เลขทะเบียน **111793**
วัน,เดือน,ปี. **23 S.ค. 2553**

b.....
i.....

ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2552

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBJECT VELOCITY DETECTOR



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2009

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ เครื่องตรวจจับความเร็ววัตถุ

OBJECT VELOCITY DETECTOR


นักศึกษาผู้จัดทำ นายพงศกร ประพันธ์ไพโรจน์ รหัสนักศึกษา 49010589

นายกัทธิไศพ จอกน้อย รหัสนักศึกษา 49010691

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2552

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
อาจารย์นรินทร์ ธรรมารักษ์วัฒน์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องตรวจจับความเร็ววัตถุ

OBJECT VELOCITY DETECTOR

นักศึกษาผู้จัดทำ นายพงศกร ประพันธ์ไพโรจน์ รหัสนักศึกษา 49010589

นายภักดีไผท จอกน้อย รหัสนักศึกษา 49010691

อาจารย์ที่ปรึกษา อาจารย์ นรินทร์ ธรรมารักษ์วัฒน์

ปีการศึกษา 2552

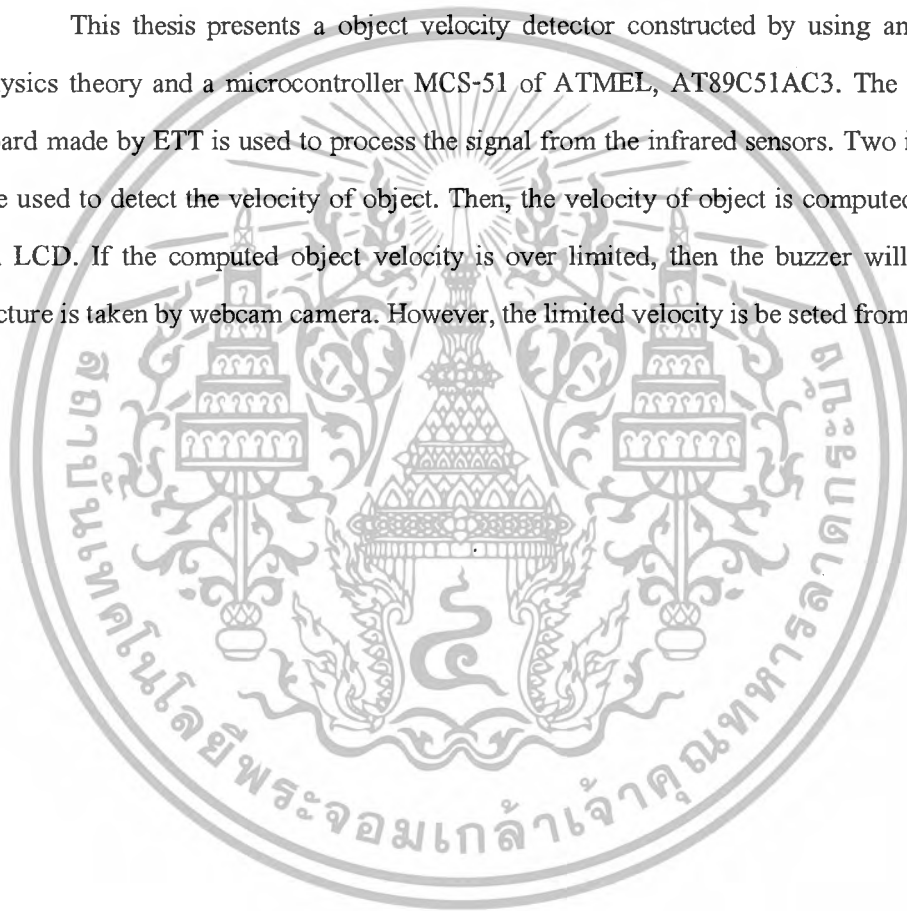
บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเสนอเครื่องตรวจจับความเร็ววัตถุ โดยอาศัยหลักการทางฟิสิกส์ มาอ้างอิงในการทำเครื่องตรวจจับความเร็ววัตถุ และใช้ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ของ ATMEL เบอร์ AT89C51AC3 ที่มีอยู่ในบอร์ดสำเร็จรูปของ ETT (ET-BASE51 AC3) มาใช้ในการประมวลผล ซึ่งมีอินฟราเรดเซนเซอร์มาใช้ในการตรวจจับวัตถุ เพื่อเปรียบเทียบความแตกต่างของเวลาระหว่างเซนเซอร์ตัวที่ 1 และเซนเซอร์ตัวที่ 2 ไปคำนวณค่าความเร็วด้วยไมโครคอนโทรลเลอร์ และแสดงผลผ่านทางหน้าจอ LCD หากมีความเร็วเกินที่กำหนดก็จะมีเสียงเตือน และมีกล้องเวปแคมจับภาพได้ โดยสามารถจำกัดค่าความเร็วผ่านสวิตซ์จากคีย์เพด

Thesis Title	Object Velocity Detector	
Authors	Mr.Pongsakorn	Parphanpairod
	Mr.Pakphatai	Choknoy
Thesis Advisor	Mr.Narin	Thummarukwattana
Year	2009	

ABSTRACT

This thesis presents a object velocity detector constructed by using an application of physics theory and a microcontroller MCS-51 of ATMEL, AT89C51AC3. The microcontroller board made by ETT is used to process the signal from the infrared sensors. Two infrared sensors are used to detect the velocity of object. Then, the velocity of object is computed and displayed on LCD. If the computed object velocity is over limited, then the buzzer will alarm and the picture is taken by webcam camera. However, the limited velocity is be seted from the keypad.



กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ เรื่อง เครื่องตรวจจับความเร็ววัตถุ (OBJECT VELOCITY DETECTOR) คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่จะต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้โครงการนี้สำเร็จลงได้ก็คือ

ขอขอบคุณ อาจารย์นรินทร์ ธรรมารักษ์วัฒนะ อาจารย์ที่ปรึกษาโครงการ ผู้ที่ได้ช่วยเสนอแนะแนวคิดแลกเปลี่ยนทัศนะกันและคอยให้ความช่วยเหลือ เอาใจใส่มาโดยตลอด จนได้โครงการออกมาเป็นรูปเป็นร่างได้อย่างในปัจจุบัน

บุคคลอันเป็นที่รักยิ่ง คือ คุณพ่อ คุณแม่ ที่ได้ให้การเลี้ยงดูเรามาเป็นอย่างดีพร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ทำให้พวกเรามีความอดทนได้ต่อทุกสภาวะกาลจนสามารถอดทนต่อสิ่งต่าง ๆ ให้ผ่านพ้นปัญหามาได้โดยไม่มี ความย่อท้อ

เทคโนโลยีในปัจจุบันและอนาคตที่ทำให้พวกเราได้แนวคิดดีๆ มาสร้างสรรค์โครงการที่ดี และที่ขาดไม่ได้ เพื่อนที่ร่วมทำโครงการนี้ที่คอยช่วยเหลือดูแลซึ่งกัน และกันมาโดยตลอด ทำให้การพัฒนาโครงการดำเนินไปได้ด้วยดี

ทั้งนี้ผู้จัดทำโครงการนี้ขอระลึกถึงพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	1
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎี.....	3
2.1 กล่าวนำ.....	3
2.2 หลักการเคลื่อนที่ในแนวเส้นตรง.....	3
2.2.1 ความหมายการเคลื่อนที่ในแนวเส้นตรง.....	3
2.2.2 การกระจัด เวลา และความเร็วเฉลี่ย.....	3
2.2.3 ความเร็วขณะหนึ่ง.....	5
2.3 วงจรเปรียบเทียบกับสัญญาณ.....	6
2.4 หลักการและความหมายของแสงอินฟราเรด.....	9
2.4.1 ประเภทของแสงอินฟราเรด.....	9
2.4.2 การปล่อยคลื่นรังสีอินฟราเรดออกจากพื้นผิววัตถุ.....	12
2.4.3 หลักการทำงานของโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด.....	13
2.4.4 เซนเซอร์ตรวจจับ GP2Y0A02Y2K.....	14
2.5 หลักการของไมโครคอนโทรลเลอร์.....	17
2.5.1 ความหมายของไมโครคอนโทรลเลอร์.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5.2 ไมโครคอนโทรลเลอร์ MCS-51.....	18
2.5.3 การเขียนโปรแกรมไมโครคอนโทรลเลอร์.....	18
2.5.4 การคอมไพล์เลอร์โดยใช้โปรแกรมเคลต์ นิวเวอร์ชัน 3 (Keil uVision 3)...	18
2.5.5 การรับค่าจาก Keypad.....	18
2.5.6 LCD Module.....	19
2.5.7 พอร์ตอนุกรม.....	23
2.5.8 บอร์ดไมโครคอนโทรลเลอร์ ET-BASE51 AC3.....	26
2.6 ความรู้เบื้องต้นเกี่ยวกับโปรแกรมวิซวลเบสิก (Visual Basic).....	35
บทที่ 3 วิธีการดำเนินงาน.....	36
3.1 กล่าวนำ.....	36
3.2 การออกแบบในส่วนของหน่วยประมวลผล.....	36
3.3 การออกแบบวงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์.....	37
3.3.1 การต่อวงจรกับอินฟราเรดเซนเซอร์.....	38
3.4 การออกแบบวงจรเปรียบเทียบแรงดัน.....	39
3.4.1 การต่อวงจร Keypad.....	41
3.4.2 การต่อวงจร Buzzer.....	42
3.4.3 การต่อจอ LCD.....	43
3.4.4 การออกแบบแผ่นวงจร PCB.....	44
3.4.5 การเขียนโปรแกรมใช้กับไมโครคอนโทรลเลอร์.....	44
3.4.6 การออกแบบซอฟต์แวร์.....	44
บทที่ 4 การทดลองและการแสดงผล.....	62
4.1 กล่าวนำ.....	62
4.2 การทดลองความสอดคล้องกับทฤษฎีของวงจร.....	62
4.2.1 วงจรจ่ายกระแสตรง 5 โวลต์.....	62
4.2.2 วงจรเปรียบเทียบแรงดัน.....	63
4.2.3 วงจร Buzzer.....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.3 การทดลองเครื่องตรวจจับความเร็ววัตถุโดยใช้กล้องที่เหลื่อมวางบนสายพาน.....	65
4.3.1 สรุปผลการทดลองเครื่องตรวจจับความเร็ววัตถุโดยใช้กล้องที่เหลื่อม วางบนสายพาน.....	80
4.4 ผลการทดลองจากโปรแกรมวิซวลเบสิก (Visual Basic).....	81
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	83
5.1 สรุปผลการทดลอง.....	83
5.2 ปัญหาที่เกิดขึ้นขณะปฏิบัติงานและการแก้ไข.....	83
5.3 แนวทางในการพัฒนา.....	84
บรรณานุกรม.....	85
ภาคผนวก.....	86



สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างค่า Emissivity ของพื้นผิววัสดุบางชนิด.....	13
2.2 ตำแหน่งขาและหน้าที่ของ LCD Module.....	20
2.3 คำสั่งของ โมดูล LCD มาตรฐานแบบตัวอักษร.....	22
4.1 ผลการทดลองความถี่จากสายพานที่ 4 Hz.....	69
4.2 ผลการทดลองความถี่จากสายพานที่ 100 Hz.....	70
4.3 ผลการทดลองความถี่จากสายพานที่ 200 Hz.....	71
4.4 ผลการทดลองความถี่จากสายพานที่ 300 Hz.....	72
4.5 ผลการทดลองความถี่จากสายพานที่ 400 Hz.....	73
4.6 ผลการทดลองความถี่จากสายพานที่ 500 Hz.....	74
4.7 ผลการทดลองความถี่จากสายพานที่ 600 Hz.....	75
4.8 ผลการทดลองความถี่จากสายพานที่ 700 Hz.....	76
4.9 ผลการทดลองความถี่จากสายพานที่ 800 Hz.....	77
4.10 ผลการทดลองความถี่จากสายพานที่ 900 Hz.....	78
4.11 ผลการทดลองความถี่จากสายพานที่ 999 Hz.....	79

สารบัญรูป

รูปที่	หน้า
2.1 ตำแหน่งของรถ ณ เวลาสองเวลาในระหว่างการเคลื่อนที่จากจุด P_1 ไป P_2 ทางขวา.....	4
2.2 ตำแหน่งของรถ ณ เวลาสองเวลาในระหว่างการเคลื่อนที่จากจุด P_1 ไป P_2 ทางซ้าย.....	5
2.3 วงจรเปรียบเทียบสัญญาณอย่างง่าย.....	6
2.4 วงจรตรวจจับผ่านศูนย์.....	7
2.5 วงจรดีฟเฟอร์เรนเชียล คอมพาราเตอร์.....	8
2.6 วงจรเปรียบเทียบแบบซิงเกิลเอ็นที่มีระดับสัญญาณแรงดันอ้างอิงคงที่.....	8
2.7 การทดลองของ Sir William Herschel.....	10
2.8 สเปกตรัมของคลื่นแม่เหล็กไฟฟ้าเนื่องจากอุณหภูมิของวัตถุ.....	11
2.9 สเปกตรัมของคลื่นแม่เหล็กไฟฟ้าเนื่องจากอุณหภูมิของวัตถุ.....	11
2.10 การทำงานของเซนเซอร์ในการวัดระยะทางถึงวัตถุตกกระทบ.....	14
2.11 เซนเซอร์ตรวจจับ GP2Y0A02Y2K.....	15
2.12 บล็อกไดอะแกรมภายในของ GP2Y0A02Y2K.....	15
2.13 ความสัมพันธ์ระหว่างแรงดันเอาต์พุตกับระยะทาง.....	16
2.14 การทำงานของไมโครคอนโทรลเลอร์.....	17
2.15 ลักษณะโครงสร้างของ Keypad.....	19
2.16 LCD Module 16 ตัวอักษร 2 บรรทัด.....	19
2.17 ตำแหน่งขาและหน้าที่ของ LCD Module.....	20
2.18 ตำแหน่งแอดเดรสของ DDRAM ของโมดูล LCD.....	23
2.19 ตำแหน่งขาใช้งานของหัวต่อ DB9.....	24
2.20 ระดับสัญญาณของ RS-232C และระดับสัญญาณของ TTL.....	24
2.21 การสื่อสารแบบอะซิงโครนัส.....	25
2.22 การรับ-ส่งข้อมูลแบบอะซิงโครนัส.....	26
2.23 โครงสร้างบอร์ด ET-BASE51 AC3 (AT89C51AC3).....	28
2.24 พอร์ต P0 มีขนาด 8 บิต.....	29
2.25 พอร์ต P1 มีขนาด 8 บิต.....	30
2.26 พอร์ต P2 มีขนาด 8 บิต.....	30
2.27 พอร์ต P3 มีขนาด 6 บิต.....	31
2.28 พอร์ต P4 มีขนาด 5 บิต.....	31

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.29 พอร์ต CLCD.....	32
2.30 การจัดเรียงขาสัญญาณของ Character LCD มาตรฐาน.....	33
2.31 พอร์ต RS-232.....	33
2.32 พอร์ต ET-DOWNLOAD.....	34
2.33 วงจรส่วนที่เชื่อมต่อกับ RS-232 และ ET-DOWNLOAD.....	34
3.1 บอร์ดสำเร็จรูปของ ETT (ET-BASE51 AC3).....	37
3.2 วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์.....	37
3.3 วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ ออกแบบด้วยโปรแกรม PCB Wizard.....	38
3.4 อินฟราเรดเซนเซอร์.....	39
3.5 วงจรเปรียบเทียบแรงดันที่ใช้กับอินฟราเรดเซนเซอร์.....	40
3.6 วงจรเปรียบเทียบแรงดันออกแบบด้วยโปรแกรม PCB Wizard ที่ใช้กับอินฟราเรดเซนเซอร์.....	40
3.7 วงจร Keypad และพอร์ต P0.....	41
3.8 Keypad ที่ต่อเรียบร้อยแล้ว.....	42
3.9 วงจร Buzzer และพอร์ต P1.....	42
3.10 วงจร Buzzer ออกแบบด้วยโปรแกรม PCB wizard.....	43
3.11 จอ LCD และอุปกรณ์เชื่อมต่อ.....	43
3.12 วงจรที่สมบูรณ์.....	44
3.13 Flowchart แสดงการทำงาน.....	45
3.14 หน้าจอ LCD ตอนเปิดเครื่อง โดยยังไม่มีวัตถุผ่าน.....	48
3.15 เครื่องตรวจความเร็วขณะวัตถุผ่านเซนเซอร์ตัวที่ 1.....	49
3.16 เครื่องตรวจความเร็วขณะวัตถุผ่านเซนเซอร์ตัวที่ 2.....	50
3.17 หน้าจอ LCD แสดงผลที่ได้จากการตรวจจับความเร็ววัตถุ.....	52
3.18 การเขียนโปรแกรมภาษา C ให้ไมโครคอนโทรลเลอร์ MCS-51.....	54
3.19 Flowchart แสดงการทำงานของโปรแกรมวิซวลเบสิก.....	55
3.20 หน้าต่างของโปรแกรมวิซวลเบสิกตอนเริ่มต้น.....	56
3.21 หน้าต่างโปรแกรมวิซวลเบสิกแสดงภาพจากกล้องเว็บแคม.....	57
3.22 โปรแกรมวิซวลเบสิกแสดงภาพและค่าที่ได้จากกล้องเว็บแคม.....	58
3.23 หน้าต่างของโปรแกรมวิซวลเบสิกตอน Start Debug	60

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.24 ส่วนประกอบของเครื่องตรวจจับความเร็ววัตถุ.....	61
3.25 เครื่องตรวจจับความเร็ววัตถุและอุปกรณ์ที่ใช้ร่วมกัน.....	61
4.1 การทดลองวงจรจ่ายกระแสตรง 5 โวลต์.....	62
4.2 การทดลองวงจรเปรียบเทียบแรงดันในสถานะ High.....	63
4.3 การทดลองวงจรเปรียบเทียบแรงดันในสถานะ Low.....	63
4.4 วงจรจับ Buzzer.....	64
4.5 แผ่นสะท้อนแสงที่ใช้ในการทดลองกับสายพาน.....	65
4.6 การวัดความเร็วรอบของสายพาน โดยใช้เครื่องเทค โคเจน.....	66
4.7 การแสดงผลออกทางหน้าจอของเครื่องออสมิด โคลส โคป.....	67
4.8 อุปกรณ์ต่างๆที่ใช้ร่วมกันในการทดลองกับสายพาน.....	68
4.9 การทดลองของเครื่องตรวจจับความเร็ววัตถุ.....	81
4.10 ผลที่ได้จากการทดลองค่าความเร็วของวัตถุบนสายพาน.....	82



บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาภิพนธ์

ปัจจุบันนี้จำนวนโรงงานอุตสาหกรรมของประเทศไทยมีจำนวนเพิ่มมากขึ้นเรื่อยๆ ทำให้มีการพัฒนาในระบบอุตสาหกรรมต่างๆ เพิ่มมากขึ้น ส่งผลให้เกิดความก้าวหน้าทางเทคโนโลยีต่างๆ ซึ่งมีผลต่อความมีมาตรฐานที่ดีในระบบอุตสาหกรรม การที่จะมีเครื่องที่มำำนวยความสะดวก อันทันสมัยได้ถูกนำมาใช้มากขึ้น ดังนั้นเพื่อเป็นการแบ่งเบาภาระของบุคลากรในการทำงาน ในส่วนของระบบสายพานลำเลียง จึงมีความจำเป็นที่จะต้องมีการตรวจสอบความเร็ววัตถุมาใช้ในการตรวจสอบความเร็วของวัตถุบนสายพานลำเลียง แต่เนื่องจากมีข้อจำกัดในการทำงานหลายประการ ในส่วนของระบบสายพานลำเลียง ซึ่งมีอยู่ในโรงงานอุตสาหกรรมที่มีขนาดใหญ่ อาจจะมี ความยากลำบากในการทำงานได้ ด้วยเหตุนี้เพื่อที่จะได้ง่ายต่อการทำงานในส่วนการตรวจสอบความเร็วของวัตถุบนสายพานลำเลียงนั้น ทางผู้จัดทำได้คิดค้นเครื่องตรวจสอบความเร็ววัตถุ เพื่อให้เกิดความเรียบร้อย รวดเร็ว ในการลำเลียง และความถูกต้องในค่าความเร็วที่สามารถจับค่าได้จากของที่อยู่บนสายพานลำเลียง ซึ่งนำมาเป็นพื้นฐานในการประยุกต์ใช้กับวัตถุอื่นๆ ที่มีความจำเป็นในอนาคตได้

1.2 วัตถุประสงค์ของปัญญาภิพนธ์

1. เพื่อศึกษาและออกแบบเครื่องตรวจสอบความเร็ววัตถุ
2. สามารถตรวจสอบความเร็ววัตถุได้
3. สามารถส่งเสียงเตือนเมื่อความเร็วของวัตถุเกินกำหนดได้
4. สามารถจับภาพของวัตถุได้ เมื่อความเร็วเกินกำหนดให้เป็นไปอย่างอัตโนมัติได้

1.3 ขอบเขตของปัญญาภิพนธ์

1. ออกแบบเครื่องตรวจสอบความเร็ววัตถุได้
2. เขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี
3. เขียนโปรแกรมวิซวลเบสิก เพื่อควบคุมการทำงานของกล้องเว็บแคมได้

1.4 ขั้นตอนการศึกษา

1. ศึกษาหลักการทำงานของอุปกรณ์ตรวจจับความเร็ว
2. ศึกษาหลักการทำงานของเซนเซอร์ที่ใช้ในการตรวจจับความเร็ว
3. ศึกษาหลักการทำงานของไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุมและแสดงผล
4. ศึกษาหลักการทำงานของวงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์
5. ศึกษาหลักการทำงานของวงจรเปรียบเทียบเทียบแรงดัน
6. ศึกษาหลักการทำงานของวงจร Buzzer
7. ศึกษาการเขียนโปรแกรม PCB Wizard 3.50 Pro Unlimited ได้
8. ศึกษาการเขียนโปรแกรมภาษาซี
9. ศึกษาการเขียนโปรแกรมภาษาซี ในไมโครคอนโทรลเลอร์ MCS-51 ได้
10. ศึกษาการเขียนโปรแกรมวิซวลเบสิก
11. ศึกษาการเขียนโปรแกรมวิซวลเบสิก ติดต่อกับกล้องเวปแคมได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำมาใช้ตรวจจับความเร็วของวัตถุได้
2. สามารถนำไปใช้ในการตรวจจับความเร็วของวัตถุบนสายพานลำเลียงของได้
3. สามารถนำมาจับภาพได้จากการทำงานของเครื่องโดยใช้กล้องเวปแคม
4. สามารถนำมาเก็บเป็นฐานข้อมูลจากภาพเมื่อมีความเร็วในระดับต่างๆ
5. สามารถนำมาใช้ในงานอุตสาหกรรม ประเภทที่ต้องการทราบค่าความเร็วในการตรวจจับความเร็วของผลิตภัณฑ์ในระบบสายพานลำเลียง

บทที่ 2 ทฤษฎี

2.1 กล่าวนำ

ในบทนี้จะกล่าวถึงหลักการทางฟิสิกส์ เรื่องความเร็วเฉลี่ย ทฤษฎีเบื้องต้นของวงจรรูปทรงที่นำมาใช้ในโครงการเช่น อินฟราเรดเซนเซอร์ ไมโครคอนโทรลเลอร์และหลักการทางงาน ตลอดจนโปรแกรมต่างๆ เช่น โปรแกรมภาษาซีที่ใช้ควบคุมไมโครคอนโทรลเลอร์ โปรแกรมวิซวลเบสิกที่ใช้ควบคุมการทำงานในส่วนของกล้อง และโปรแกรมต่างๆ ที่ต้องใช้ในการทำโครงการ

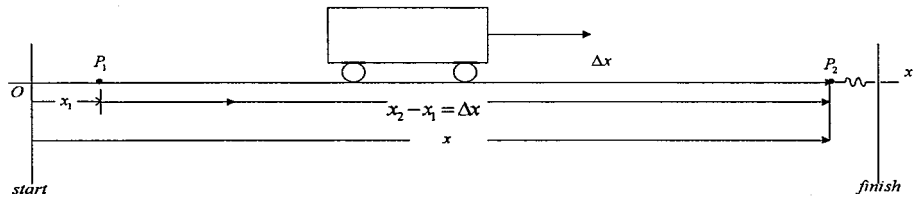
2.2 หลักการเคลื่อนที่ในแนวเส้นตรง

2.2.1 ความหมายการเคลื่อนที่ในแนวเส้นตรง

การเคลื่อนที่แบบง่ายที่สุดคือ อนุภาคเดี่ยวเคลื่อนที่ในแนวเส้นตรง เรามักจะใช้อนุภาคเป็นแบบจำลองแทนวัตถุเมื่อผล เช่น การหมุนหรือการเปลี่ยนรูปร่างไม่สำคัญ เราจะใช้ปริมาณความเร็วและความเร่งอธิบายการเคลื่อนที่ของอนุภาค ปริมาณเหล่านี้มีนิยามง่ายๆ ในฟิสิกส์ อย่่างไรก็ตามนิยามเหล่านั้นเป็นนิยามที่แม่นยำมากกว่า และแตกต่างไม่มากจากนิยามความเร็ว และความเร่งก็คือปริมาณเหล่านี้เป็นเวกเตอร์ นั่นหมายความว่าปริมาณเหล่านี้มีทั้งขนาดและทิศทาง การเคลื่อนที่ในแนวตรง กรณีพิเศษที่สำคัญคือกรณีที่มีความเร่งมีค่าคงตัว ตัวอย่างหนึ่งคือการเคลื่อนที่ของวัตถุที่ตกอย่างเสรี เราจะพัฒนาสมการง่าย ๆ เพื่อใช้อธิบายการเคลื่อนที่ด้วยความเร่งคงตัว เราจะพิจารณาสถานการณ์ที่ความเร่งมีค่าไม่คงตัวในระหว่างการเคลื่อนที่ด้วย ในกรณีนี้เราต้องใช้การอินทิเกรตเพื่ออธิบายการเคลื่อนที่

2.2.2 การกระจัด เวลา และความเร็วเฉลี่ย

สมมุตินักขับรถแข่งคันหนึ่งขับรถแข่งไปตามเส้นทางตรง เพื่อที่จะศึกษาการเคลื่อนที่นี้ เราต้องมีระบบพิกัดเพื่อใช้อธิบายตำแหน่งรถแข่ง เราเลือกแกน x ของระบบพิกัดให้อยู่ในแนวเส้นทางตรงของรถแข่งโดยมีจุดกำเนิด 0 อยู่ที่เส้นเริ่มแข่ง เราจะอธิบายตำแหน่งของรถแข่งในรูปของตำแหน่งของจุดตัวแทน เช่น จุดปลายหน้ารถ โดยการทำเช่นนี้เราแทนรถแข่งทั้งคันด้วยจุดนั้น และดังนั้นพิจารณาการแข่งเป็นอนุภาค



รูปที่ 2.1 ตำแหน่งของรถ ณ เวลาสองเวลาในระหว่างการเคลื่อนที่จากจุด P_1 ไป P_2 ทางขวา

เราบอกตำแหน่งจุดหน้ารถด้วยพิกัด x ซึ่งแปรตามเวลาในขณะที่รถเคลื่อนที่ วิธีอธิบายการเคลื่อนที่ของรถที่มีประโยชน์วิธีหนึ่งใช้การเปลี่ยนตำแหน่ง x ในช่วงเวลาหนึ่ง สมมติว่า 1.0 วินาที หลังจากเริ่มต้นจุดหน้ารถอยู่ที่ P_1 ซึ่งอยู่ห่างจากจุดกำเนิด 19 เมตร และ 4.0 วินาที หลังจากเริ่มต้นรถอยู่ที่ P_2 ซึ่งห่างจากจุดกำเนิด 227 เมตร ดังนั้นรถเคลื่อนที่ได้ $(227 \text{ เมตร} - 19 \text{ เมตร}) = 258 \text{ เมตร}$ ในช่วงเวลา $(4.0 \text{ วินาที} - 1.0 \text{ วินาที}) = 3.0 \text{ วินาที}$ เรานิยามความเร็วเฉลี่ยของรถแข่งในช่วงเวลานี้ให้เป็นปริมาณเวกเตอร์ ซึ่งองค์ประกอบตามแกน x คือการเปลี่ยนตำแหน่ง x หารด้วยช่วงเวลา $(258 \text{ เมตร}) / (3.0 \text{ วินาที}) = 86 \text{ เมตร/วินาที}$ ความเร็วเฉลี่ยมีค่าขึ้นอยู่กับช่วงเวลา สำหรับช่วงเวลา 3.0 วินาที ก่อนการแข่งขัน ความเร็วเฉลี่ยเป็นศูนย์เพราะรถหยุดนิ่งที่เส้นตั้งต้นและมีการกระจัดเท่ากับศูนย์

ขยายความคิดของความเร็วเฉลี่ยให้กว้างขึ้นที่ t_1 รถแข่งอยู่ที่ P_1 ซึ่งมีพิกัด x_1 และที่ t_2 รถแข่งจะอยู่ที่ P_2 ซึ่งมีพิกัด x_2 การกระจัดของรถแข่งในช่วงเวลาจาก t_1 ไป t_2 คือเวกเตอร์จาก P_1 ไป P_2 ที่มีองค์ประกอบตามแกน x เท่ากับ $(x_2 - x_1)$ และมีองค์ประกอบตามแกน y และแกน z เป็นศูนย์ องค์ประกอบตามแกน x ของการกระจัดของรถเป็นเพียงการเปลี่ยนแปลงพิกัด x ซึ่งเราเขียนให้กระชับมากขึ้นเป็น

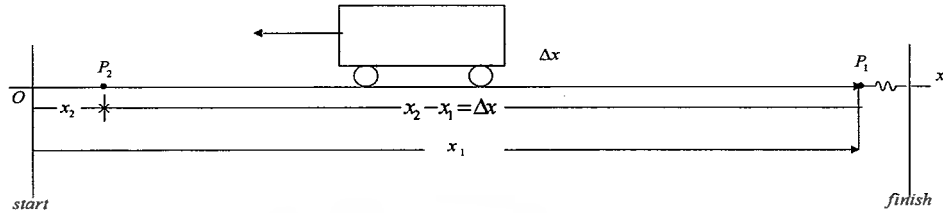
$$\Delta x = x_2 - x_1 \quad (1)$$

Δ (เดลต้า) แทนการเปลี่ยนแปลงของปริมาณหนึ่ง ซึ่งหาได้จากการลบค่าเดิมออกจากค่าสุดท้าย Δx ไม่ใช่ผลคูณระหว่าง Δ กับ x แต่เป็นสัญลักษณ์เดียวซึ่งหมายถึง การเปลี่ยนแปลงของปริมาณ x ในทำนองเดียวกันเราเขียนช่วงเวลาจาก t_1 ไป t_2 ด้วย $\Delta t = t_2 - t_1$ สังเกตว่า Δx หรือ Δt หมายถึงค่าสุดท้ายลบด้วย ค่าเดิมเสมอ ไม่ใช่กลับกัน

ที่นี้เราให้นิยามองค์ประกอบตามแกน x ของความเร็วเฉลี่ยได้แม่นยำมากขึ้น ค่านี้คือองค์ประกอบตามแกน x ของการกระจัด Δx หารด้วยช่วงเวลา Δt ที่การกระจัดนั้นเกิดขึ้นเราแทนปริมาณนี้ด้วยอักษร v พร้อมสัญลักษณ์ห้อย av เพื่อบอกว่าเป็นค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v_{av} = \frac{x_2 - x_1}{t_2 - t_1} = \frac{\Delta x}{\Delta t} \quad (2)$$



รูปที่ 2.2 ตำแหน่งของรถ ณ เวลาสองเวลาในระหว่างการเคลื่อนที่จากจุด P_1 ไป P_2 ทางซ้าย

ในที่นี้อ้างถึงการเคลื่อนที่ของรถ และดังนั้นต่างจากรูปที่ 2.1 องค์ประกอบตามแกน x ของการกระจัดของรถเป็นลบและดังนั้น v_{av} มีค่าเป็นลบ

2.2.3 ความเร็วขณะหนึ่ง

ความเร็วเฉลี่ยของอนุภาคไม่สามารถบอกได้ว่าอนุภาคกำลังเคลื่อนที่เร็วแค่ไหน หรือกำลังไปในทิศทางใด ณ เวลาขณะหนึ่ง เพื่อที่จะอธิบายการเคลื่อนที่ให้ละเอียดมากขึ้น เราต้องนิยามความเร็วขณะใด ๆ บนเส้นทางการเคลื่อนที่ เราเรียกความเร็วเช่นนั้นว่าความเร็วขณะหนึ่ง

จะสังเกตว่าคำว่าขณะหนึ่งมีความหมายในฟิสิกส์ที่ค่อนข้างจะแตกต่างกัน จากภาษาประจำวันอาจใช้วลีว่า มันใช้เวลาานเพียงขณะหนึ่ง เพื่อพูดถึงบางอย่างที่คงอยู่ในช่วงเวลาสั้นมาก แต่ในฟิสิกส์คำว่าขณะหนึ่งไม่มีช่วงเวลาใดเลย มันหมายถึงค่า ๆ หนึ่งของเวลา

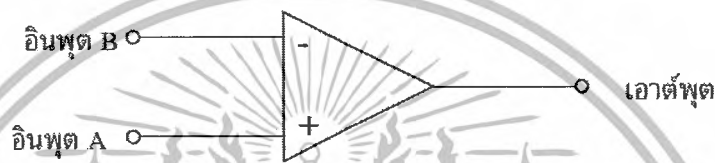
เพื่อหาความเร็วขณะหนึ่งของรถในรูปที่ 2.1 ที่จุด P_1 เราจินตนาการเคลื่อนจุด P_2 ให้เข้าใกล้จุด P_1 มากขึ้นเรื่อย ๆ เราคำนวณความเร็วเฉลี่ย $v_{av} = \frac{\Delta x}{\Delta t}$ ในช่วงการกระจัดและช่วงเวลาที่สั้นมากขึ้น ๆ ทั้ง Δx และ Δt จะมีค่าเล็กลงมากขึ้นทั้งคู่ แต่อัตราส่วนระหว่างปริมาณทั้งสองไม่จำเป็นต้องมีค่าเล็กลงด้วย ในภาษาของแคลคูลัสเราเรียกลิมิตของ $\frac{\Delta x}{\Delta t}$ ในขณะที่ Δt เข้าใกล้ศูนย์ว่าอนุพันธ์ของ x เทียบกับ t และเขียนว่า $\frac{dx}{dt}$ ความเร็วขณะหนึ่งคือลิมิตของความเร็วเฉลี่ยเมื่อช่วงเวลาเข้าใกล้ศูนย์ ค่านี้มีค่าเท่ากับอัตราการเปลี่ยนแปลงตำแหน่งเทียบกับเวลาขณะหนึ่ง เราใช้สัญลักษณ์ v โดยไม่มีตัวห้อยแทนความเร็วขณะหนึ่ง

$$v = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} = \frac{dx}{dt} \quad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วงจรเปรียบเทียบสัญญาณ (Comparator Circuit)

วงจรเปรียบเทียบสัญญาณ (Comparator) เป็นวงจรที่มีบทบาทสำคัญในการควบคุมการทำงานของระบบในงานวิจัยนี้ ดังนั้นในบทนี้จึงได้นำวงจรเปรียบเทียบสัญญาณมาอธิบาย ในการเปรียบเทียบสัญญาณแรงดันนี้ โดยมากจะใช้โอปแอมป์ (Op-Amp) มาเป็นตัวแสดงผลซึ่งสภาวะเอาต์พุตของวงจรเปรียบเทียบสัญญาณจะสามารถบอกได้ว่าสัญญาณอินพุตสองสัญญาณ สัญญาณใดมีค่ามากกว่าหรือน้อยกว่ากัน ดังนั้นวงจรเปรียบเทียบสัญญาณจึงเป็นวงจรที่มีอินพุตเป็นสัญญาณอนาล็อก (Analog) และมีเอาต์พุตเป็นสัญญาณดิจิทัล (Digital) ลักษณะของวงจรเปรียบเทียบสัญญาณอย่างง่ายแสดงดังรูปที่ 2.3



รูปที่ 2.3 วงจรเปรียบเทียบสัญญาณอย่างง่าย

จากรูปที่ 2.3 จะได้ว่าแรงดันเอาต์พุตของวงจร (V_0) จะมีค่าเป็น

$$V_0 = (V_{inA} - V_{inB}) A_{vol} \quad (4)$$

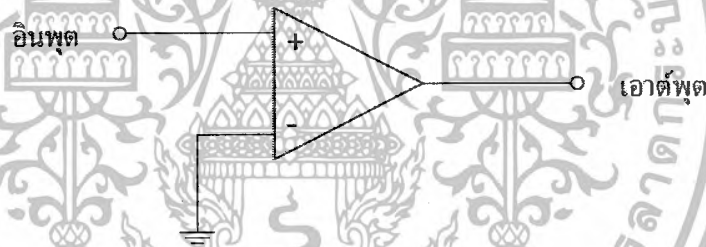
โดยที่ A_{vol} คืออัตราขยายแรงดันรูปเปิด (Open-loop voltage gain) ของโอปแอมป์ส่วน V_{inA} และ V_{inB} เป็นค่าแรงดันจากอินพุต A และ B ตามลำดับ เมื่อเทียบกับจุดกราวด์จะได้ว่า ถ้า V_{inA} มีค่ามากกว่า V_{inB} เทอมในวงเล็บดังสมการที่ (4) จะมีค่าเป็นบวกและโอปแอมป์ในอุดมคติจะให้แรงดันเอาต์พุตเป็นค่าบวกอนันต์ (Infinity) V_{inA} มีค่าน้อยกว่า V_{inB} แล้ว เทอมในวงเล็บของสมการนี้จะมีค่าเป็นลบ และโอปแอมป์ในอุดมคติจะให้แรงดันเอาต์พุตเป็นค่าลบอนันต์ แต่ในทางปฏิบัติแรงดันเอาต์พุตนี้ถูกจำกัดโดยแหล่งจ่ายไฟเลี้ยง และวงจรภายในของโอปแอมป์ซึ่งค่าแรงดันเอาต์พุตที่ได้ จะมีค่าน้อยกว่าแหล่งจ่ายไฟเลี้ยงประมาณ 1 โวลต์ และเรียกค่าแรงดันนี้ว่า แรงดันอิ่มตัว (Saturation voltage : V_{SAT}) ดังนั้นถ้าสมมติว่า แหล่งจ่ายไฟเลี้ยงมีค่าเป็น ± 15 โวลต์ แล้ว

V_{SAT+} ที่ได้จะมีค่าเท่ากับ ± 14 โวลต์ และ V_{SAT-} จะมีค่าเท่ากับ -14 โวลต์ แต่ในกรณีที่แหล่งจ่ายไฟเลี้ยงมีขั้วเดียว (Unipolar) เช่น $+30$ โวลต์ จะได้ค่า V_{SAT+} ประมาณ $+29$ โวลต์ และจะมีค่า V_{SAT-}

ประมาณ +1 โวลต์ ดังนั้นจากที่กล่าวมาข้างต้นจะพบว่าค่าแรงดันเอาต์พุตของวงจรเปรียบเทียบสัญญาณจึงมีสองสถานะคือ

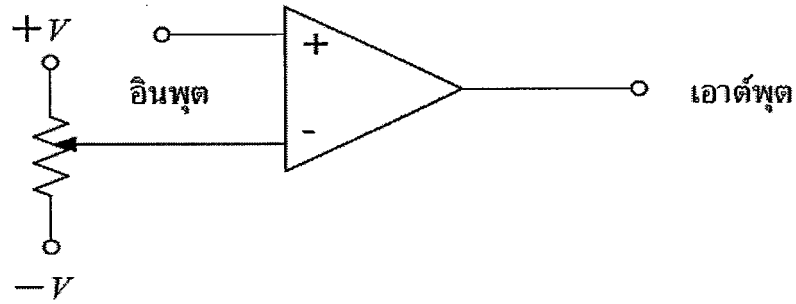
$$V_{out} = \begin{cases} V_{SAT+} & ; V_{inA} > V_{inB} \\ V_{SAT-} & ; V_{inB} < V_{inA} \end{cases} \quad (5)$$

การประยุกต์ใช้วงจรเปรียบเทียบสัญญาณที่พบบ่อยๆ เช่น นำมาใช้เพื่อแสดงว่าสัญญาณอินพุตที่เข้ามามีค่ามากหรือน้อยกว่าศูนย์ โดยวงจรเปรียบเทียบสัญญาณที่ทำหน้าที่เช่นนี้เรียกว่า วงจรตรวจจับผ่านศูนย์ (Zero Crossing Detector) ซึ่งการต่อวงจรมีลักษณะดังแสดงในรูปที่ 2.4 นั่นคือจะมีขั้วอินพุตขั้วใดขั้วหนึ่งถูกป้อนอินพุตให้มีค่าระดับสัญญาณเป็นศูนย์ หรือต่อลงกราวด์นั่นเอง เพื่อใช้เป็นระดับสัญญาณแรงดันอ้างอิง (Voltage Reference : ref V) ให้กับวงจรเปรียบเทียบนี้



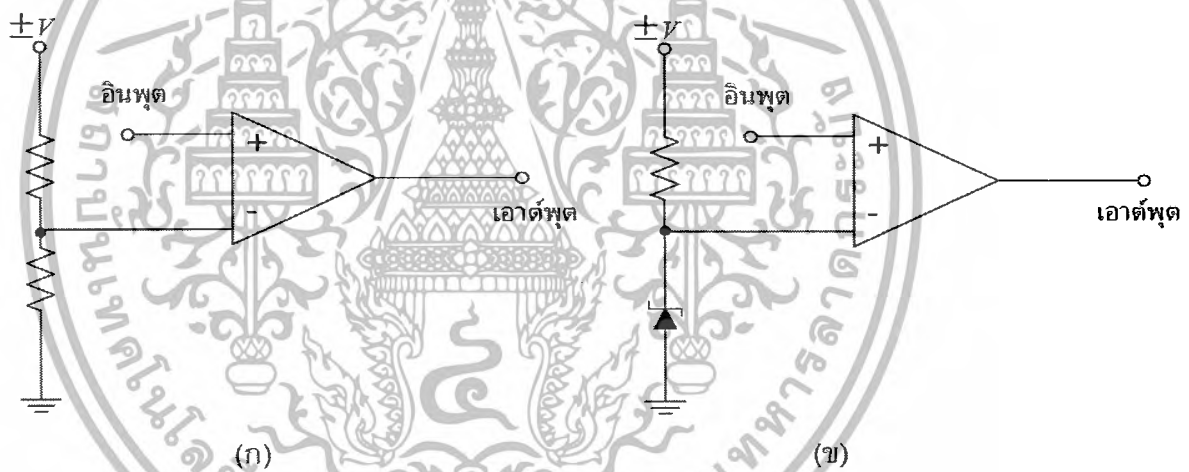
รูปที่ 2.4 วงจรตรวจจับผ่านศูนย์

ปกติโดยทั่วไปแล้วค่าระดับแรงดันอินพุตที่ใช้สำหรับอ้างอิง ไม่จำเป็นจะต้องมีค่าเท่ากับ 0 โวลต์เสมอไป ทั้งนี้อาจจะมีค่ามากกว่าหรือน้อยกว่าศูนย์ก็ได้ และอาจจะเป็นระดับแรงดันอ้างอิงที่มีค่าคงที่ หรือมีค่าเปลี่ยนแปลงตามเวลาก็ได้ สำหรับวงจรเปรียบเทียบที่มีแรงดันอ้างอิงเป็นสัญญาณที่มีค่าเปลี่ยนแปลงตามเวลาได้นี้เรียกว่า วงจรเปรียบเทียบสัญญาณแบบผลต่างหรือดิฟเฟอเรนเชียล คอมพาราเตอร์ (Differential Comparator) ซึ่งวงจรมีลักษณะดังรูปที่ 2.5



รูปที่ 2.5 วงจรดิฟเฟอเรนเชียล คอมพาราเตอร์

ส่วนในกรณีที่สัญญาณแรงดันมีค่าเป็นศูนย์ หรือเป็นระดับใดๆที่มีค่าคงที่เรียกว่า วงจรเปรียบเทียบแบบซิงเกิลเอนด์ (Single-Ended Comparator) ตัวอย่างวงจรในกรณีนี้มีลักษณะดังรูปที่ 2.6



วงจรเปรียบเทียบแรงดันด้วยความต้านทาน วงจรเปรียบเทียบแรงดันที่ตกคร่อมซีเนอร์ไดโอด

รูปที่ 2.6 วงจรเปรียบเทียบแบบซิงเกิลเอนด์ที่มีระดับสัญญาณแรงดันอ้างอิงคงที่

สำหรับวงจรเปรียบเทียบสัญญาณในรูปที่ 2.6 (ก) จะนำความต้านทานมาแบ่งแรงดันเพื่อใช้เป็นระดับสัญญาณอ้างอิงส่วนวงจรตามรูปที่ 2.6 (ข) ระดับสัญญาณแรงดันอ้างอิงได้มาจากแรงดันที่ตกคร่อมซีเนอร์ไดโอด (Zener Diode) นั้นเอง

ในการเปรียบเทียบสัญญาณของวงจรเปรียบเทียบสามารถพิจารณาได้ 2 ลักษณะที่แตกต่างกันดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การเปรียบเทียบสัญญาณด้วยจุดสัญญาณเข้าที่ขั้วบวก
2. การเปรียบเทียบสัญญาณด้วยจุดสัญญาณเข้าที่ขั้วลบ

2.4 หลักการและความหมายของแสงอินฟราเรด

หลักการของวงจรอินฟราเรด ตัวส่งแบบใช้กับโมดูลรับสำเร็จรูป (3ขา) จะส่งด้วยความถี่ 40 KHz โดยประมาณ ประโยชน์เพื่อเป็นความถี่หลักในการตรวจจับว่าเป็นสัญญาณตัวจริงไม่ใช่สัญญาณรบกวน ตัวรับแบบโมดูล (3ขา) โมดูลจะรับสัญญาณที่กระทบด้วยความถี่ประมาณ 40 KHz ถ้าตรงก็จะให้เอาต์พุตที่ขาเอาต์พุตเป็น 0 หลักการของมันก็มีแค่ส่งแสงอินฟราเรดไปยังวัตถุที่ต้องการตรวจจับ ถ้าพบวัตถุนั้นก็สะท้อนแสงกลับมายังตัวรับ สิที่สะท้อนได้ดีที่สุดก็คือสีขาว ถ้าเป็นสีดำจะดูดกลืนได้มากกว่าทั่วไป

2.4.1 ประเภทของแสงอินฟราเรด

เซนเซอร์ อะไรก็แล้วแต่มีหลักการเหมือนกัน คือต้องมีตัวส่งสัญญาณ แล้วก็ตัวรับสัญญาณ แล้วก็เอาสัญญาณที่ได้จากตัวรับมาประมวลผล คราวนี้ก็ขึ้นอยู่กับว่าจะเอาเซนเซอร์นั้นไปใช้งานทำอะไร อย่างเช่นคำว่า Infrared Sensor ก็คือบอกว่าเป็นเซนเซอร์ที่ใช้คือสัญญาณ เป็น Infrared Light บอกแค่นั้น ยังไม่ได้บอกว่าจะเอาไปใช้ทำอะไร ตัวอย่างเช่น เอา Infrared Sensor ไปใช้ตรวจจับวัตถุเมื่อมีวัตถุผ่านทางวิถีของ Infrared Light หลักการทำงานของวงจรก็คือ ทางด้านฝั่งส่งสัญญาณ Infrared ก็ทำหน้าที่คือขับ Infrared Light ให้ออกจาก Infrared LED จะให้มีความแรง แล้วไปตั้งตัวรับจะใช้เป็น Photo Transistor ก็ได้ แต่ก็ตั้งค่า Set Up ค่าในสถานะปกติให้ดีกว่าก่อนคือ ให้วงจรไบอัสแชนให้ได้ว่า สภาวะนี้มีแสง Infrared มากเกินไปหรือไม่ ถ้าไม่มีก็ไม่สามารถทำได้

เซนเซอร์แบบอินฟราเรดเป็นเซนเซอร์ที่อยู่ในแบบใช้แสง (Optic Sensor) การจะใช้อินฟราเรดเป็นตัวเซนเซอร์เรา ต้องคำนึงถึงว่าเราจะนำอินฟราเรดไปตรวจจับหรือเซนเซอร์อะไร ควรต้องคิดถึงหลักความจริงว่า เซนเซอร์นั้นเป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่ง ที่อาจจะมีผลกระทบต่างๆได้ เช่น แสงที่กระทบต่อผลการตรวจจับทำให้ข้อมูลที่ได้อาจผิดพลาด ขึ้นอยู่กับการติดตั้งเซนเซอร์มากกว่า และการออกแบบระบบหรือวงจรมันเอง ซึ่งเอาต์พุตของเซนเซอร์แต่ละตัวจะออกมาเป็นสัญญาณซึ่งเราต้องนำเอาต์พุตนั้น ไปประมวลผลหรือแปลงสัญญาณ เพื่อจะให้ได้สัญญาณที่เหมาะสม เพื่อจะทำการเอาสัญญาณนั้นไปใช้ส่วนมากเอาต์พุตที่ออกมาจะเป็นแรงดันหรือไม่มีกระแส ซึ่งก็นำเอากระแสหรือแรงดันนั้นไปไบอัสหรือเป็นตัวทริกให้กับอุปกรณ์อิเล็กทรอนิกส์ที่ต้องการควบคุมเพียงแค่นั้น ก็จะทำได้สามารถควบคุมวงจรหรืออุปกรณ์อิเล็กทรอนิกส์ต่างๆได้ หลักการทำงานของวงจรก็คือ จะมีตัวอินฟราเรดเป็นตัวส่งสัญญาณและมีตัวรับหรือไม่มีตัวส่งสัญญาณแล้วกระทบแล้วสะท้อนแสงมายังตัวรับ ซึ่งตัวรับจะเป็นตัวส่งสัญญาณ

ซึ่งสัญญาณที่ได้เป็นกระแสและแรงดัน ซึ่งเรานำเอาสัญญาณซึ่งเป็นกระแส และแรงดันนั้นไปใช้ในงานต่างๆ

เซนเซอร์แสง สามารถทำได้หลายๆแบบ แต่ละแบบทำงานในลักษณะเดียวกันคือมีตัวส่งแสงอินฟราเรดไปกระทบวัตถุ ถ้าวัตถุนั้นสามารถสะท้อนแสงได้ดี แสงดังกล่าวก็จะมาเป็นอินพุตให้อุปกรณ์รับแสงซึ่งอาจจะเป็นโฟโตไดโอด โฟโตทรานซิสเตอร์หรือแอลดีอาร์ก็ได้ หรืออาจใช้อุปกรณ์เซนเซอร์แสงแบบสำเร็จก็มีขาย คือมีทั้งตัวส่งและตัวรับในตัวถึงเดียวกัน แต่ถ้าวัตถุไม่สะท้อนแสงหรือสะท้อนแสงได้น้อย เช่น วัตถุสีดำ ตัวรับก็จะไม่มีนำกระแสหรือนำกระแสเพียงเล็กน้อย ผลจากการเปลี่ยนแปลงของเซนเซอร์สามารถนำไปเข้าวงจรเปรียบเทียบแรงดันให้เป็นค่าลอจิก ซึ่งจะง่ายกับการนำไปควบคุมงานที่ได้ออกแบบ

สิ่งที่ควรคำนึงมากอีกสิ่งหนึ่งสำหรับการใช้อินฟราเรดเซนเซอร์คือ แสงอินฟราเรดจากภายนอกซึ่งมีอยู่ทั่วไปและมีมากในตอนกลางวันและวันที่ฟ้าโปร่งโล่ง ทางที่ดีควรจะใช้การ Mod เข้าไป อาจจะเป็น PCM ก็ได้ซึ่งจะช่วยได้ในระดับหนึ่ง

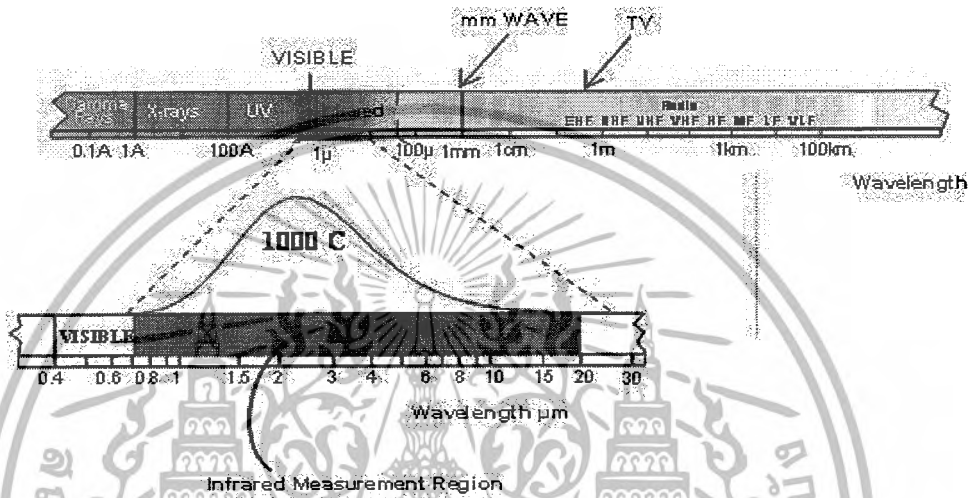
คลื่นรังสีอินฟราเรดหรือคลื่นรังสีความร้อนนั้น ถูกค้นพบโดยนักวิทยาศาสตร์ชาวอังกฤษ คือ Sir William Herschel ซึ่งได้ค้นพบรังสีอินฟราเรดสเปกตรัม (Infrared Ray) ในปี ค.ศ.1800 โดยเขาได้ทำการทดลองวัดอุณหภูมิของแถบสีต่างๆ ที่เปล่งออกมาเป็นสีรุ้งจากปริซึม และพบว่าอุณหภูมิความร้อนจะเพิ่มขึ้นตามลำดับและสูงสุดที่แถบสีสีแดง ในความเป็นจริงนั้นการที่เขาเลื่อนเทอร์โมมิเตอร์จากแถบสีที่ไม่สว่างไปยังแถบสีแดง ซึ่งเป็นแถบสีที่สิ้นสุดของสเปกตรัม และอุณหภูมิสูงขึ้นเป็นลำดับ ซึ่งขอบเขตดังกล่าวนี้เรียกว่า อินฟราเรด (ขอบเขตที่ต่ำกว่าแถบสีแดง) แสดงดังรูปที่ 2.8



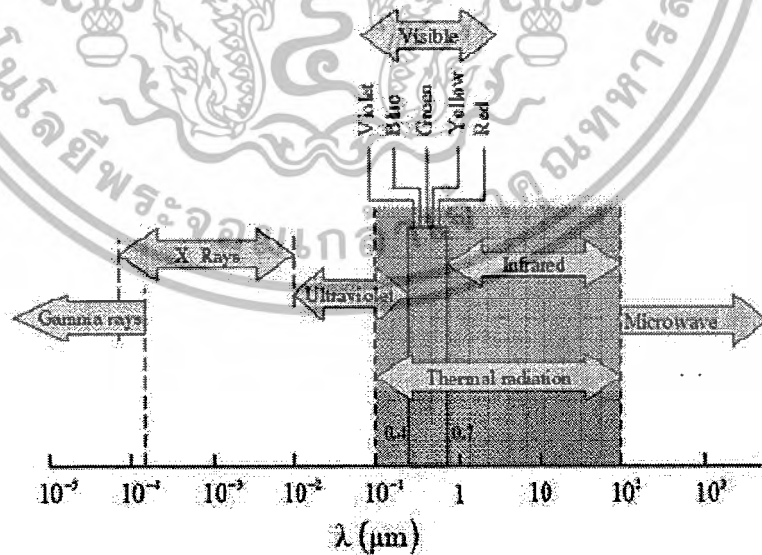
รูปที่ 2.7 การทดลองของ Sir William Herschel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายตาของมนุษย์นั้นธรรมชาติได้ออกแบบ ให้สามารถมองเห็นคลื่นได้ในระดับหนึ่งหรือเรียกว่า รังสีการมองเห็น (Visible Light) โดยจะอยู่ในรูปของความสว่าง จากรูปที่ 2.8 จะเห็นได้ว่าเราสามารถมองเห็นรังสีในย่านแคบๆ เท่านั้น หรือเรียกว่าย่านสเปกตรัมแม่เหล็กไฟฟ้า ซึ่งถ้ามองด้านซ้ายของสเปกตรัมแม่เหล็กไฟฟ้าก็จะติดกับย่านรังสีอัลตราไวโอเล็ต ซึ่งมนุษย์ไม่สามารถมองเห็นในย่านนี้และส่วนด้านขวาของสเปกตรัมแม่เหล็กไฟฟ้าก็จะติดกับย่านรังสีอินฟราเรดซึ่งเราก็ไม่สามารถมองเห็นเช่นกัน



รูปที่ 2.8 สเปกตรัมของคลื่นแม่เหล็กไฟฟ้านื่องจากอุณหภูมิของวัตถุ



รูปที่ 2.9 สเปกตรัมของคลื่นแม่เหล็กไฟฟ้านื่องจากอุณหภูมิของวัตถุ
หน่วยของความยาวคลื่น โดยปกติเป็นไมโครเมตร (μm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งกำเนิดของรังสีอินฟราเรดก็คือ ความร้อนหรือการแผ่รังสีความร้อน ไม่ว่าจะวัตถุใดก็ตามที่มีอุณหภูมิสูงกว่าค่า Absolute Zero (-273.15 องศาเซลเซียส หรือ 0 องศาเคลวิน) จะมีการแผ่รังสีในย่านอินฟราเรดหรือรังสีความร้อนออกมา แม้ในโตรเจนที่อุณหภูมิต่ำ -196 องศาเซลเซียส ก็ยังมีการแผ่รังสีอินฟราเรดออกมาได้ถึงแม้จะมีค่าน้อยก็ตาม ซึ่งในความเป็นจริงและในชีวิตประจำวัน เราจะสัมผัสและเกี่ยวข้องกับรังสีอินฟราเรดอยู่ตลอดเวลา เช่น เราารู้สึกร้อนเมื่ออยู่กลางแจ้งหรือใกล้ไฟ ถึงแม้ว่าตามนุษย์จะไม่สามารถมองเห็นรังสีอินฟราเรดได้ แต่ผิวหนังก็สามารถรับรู้รังสีจากความร้อนได้ วัตถุที่ร้อนย่อมแผ่พลังงานรังสีออกมามากกว่าวัตถุที่เย็น

2.4.2 การปล่อยคลื่นรังสีอินฟราเรดออกจากพื้นผิววัตถุ

จากแนวคิดเกี่ยวกับวัตถุดำ เพื่อเป็นการอธิบายพฤติกรรมของผิวอุดมคติ ต่อจากนี้เราจะพิจารณาผิวของวัตถุจริงบ้าง วัตถุดำเป็นตัวอย่างตัวปล่อยรังสีออกในอุดมคติที่ไม่มีผิววัตถุใดสามารถส่งออกรังสีได้มากกว่าวัตถุดำในอุณหภูมิเดียวกัน เพราะฉะนั้นจึงเลือกวัตถุดำมาเป็นมาตรฐานในการอธิบายของการแผ่หรือการปล่อยรังสีออกจริงนั้น คุณสมบัติการปล่อยรังสีออกเรียกว่า Emissivity (e) กำหนดเป็นอัตราส่วนของการปล่อยรังสีของวัตถุใดๆต่อรังสีของวัตถุดำที่อุณหภูมิเดียวกัน จะเห็นได้ว่าการปล่อยออกมีค่าสูงสุดได้ไม่เกินหนึ่งหมายความว่าพื้นผิวใดๆ สามารถแผ่รังสีได้สูงสุดเท่ากับวัตถุดำนั่นเอง

สำหรับการคำนวณทางวิศวกรรมจะถือว่าคุณสมบัติของผิววัตถุ จะสามารถปล่อยออกเฉลี่ยเท่ากันทุกทิศทางเป็นผลให้ไม่ขึ้นกับทิศทาง ค่า Emissivity (e) จะขึ้นอยู่กับลักษณะพื้นผิวของวัตถุ การเป็นตัวนำ (Conductor) หรือไม่เป็นตัวนำ (Non-Conductor) โดยทั่วไปค่า Emissivity ของผิวโลหะจะมีค่าน้อย และถ้าเป็นโลหะผิวมันเงาก็ยิ่งจะมีค่าน้อยกว่า 0.02 แต่ถ้าทาสีบนแผ่นโลหะนั้นค่าอาจเพิ่มขึ้น และถ้าโลหะมีสนิมก็เพิ่มขึ้นได้เช่นกัน ยกตัวอย่างเช่น $e = 0.01$ สำหรับเหล็กสแตนเลสมีสนิมเล็กน้อย แต่ถ้ามีสนิมมากจะมีค่าประมาณ 0.5 ส่วนค่าปกติเป็นตัวนำจะมีค่ามากกว่า 0.6 ขึ้นไป และเพิ่มขึ้นหรือลดถ้าอุณหภูมิมีการเปลี่ยนแปลง

ตารางที่ 2.1 ตัวอย่างค่า Emissivity ของพื้นผิววัสดุบางชนิด

Surface	อุณหภูมิ, °C	Emissivity
Polished silver	225-625	0.0198-0.0324
Platinum filament	25-1225	0.036-0.192
Polished nickel	23	0.045
Aluminum foil	100	0.087
Concrete	21	0.63
Roofing paper	20	0.91
Plaster	10-88	0.91
Rough red brick	21	0.93
Asbestos paper	38-371	0.93-0.945
Smooth glass	22	0.937
Water	0-100	0.95-0.963
Blackbody	-	1.00

2.4.3 หลักการทำงานของโมดูลตรวจวัดทางด้วยแสงอินฟราเรด

โมดูลวัดระยะทางด้วยแสงอินฟราเรด จะส่งแสงอินฟราเรดจากตัวส่ง ไปกระทบวัตถุผ่านเลนส์นูนเพื่อโฟกัสแสงให้มีความเข้มแสงไปยังจุดใดจุดหนึ่ง เมื่อแสงไปกระทบวัตถุจะเกิดการกระเจิงของแสงไปในทิศทางต่างๆ แสงส่วนหนึ่งจะกระเจิงไปยังภาครับ โดยมีเลนส์ภาครับทำหน้าที่รวมแสงและกำหนดจุดตกกระทบ แสงจะถูกส่งผ่านไปยังโฟโตทรานซิสเตอร์จำนวนมากที่ต่อเรียงตัวกันเป็นส่วนรับแสงหรืออะเรย์รับแสง ตำแหน่งที่แสงตกกระทบนี้สามารถนำมาคำนวณหาระยะทาง (L) จากภาคส่งไปยังวัตถุได้ โดยใช้สูตรหาสามเหลี่ยมคล้ายดังนี้

$$\frac{L}{A} = \frac{F}{X} \quad (6)$$

ดังนั้นค่า L จะมีค่าเท่ากับ

$$L = \frac{F \times A}{X} \quad (7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดให้

L = ระยะตั้งฉากจากจุดศูนย์กลางเลนส์ LED อินฟราเรดตัวส่งถึงวัตถุ

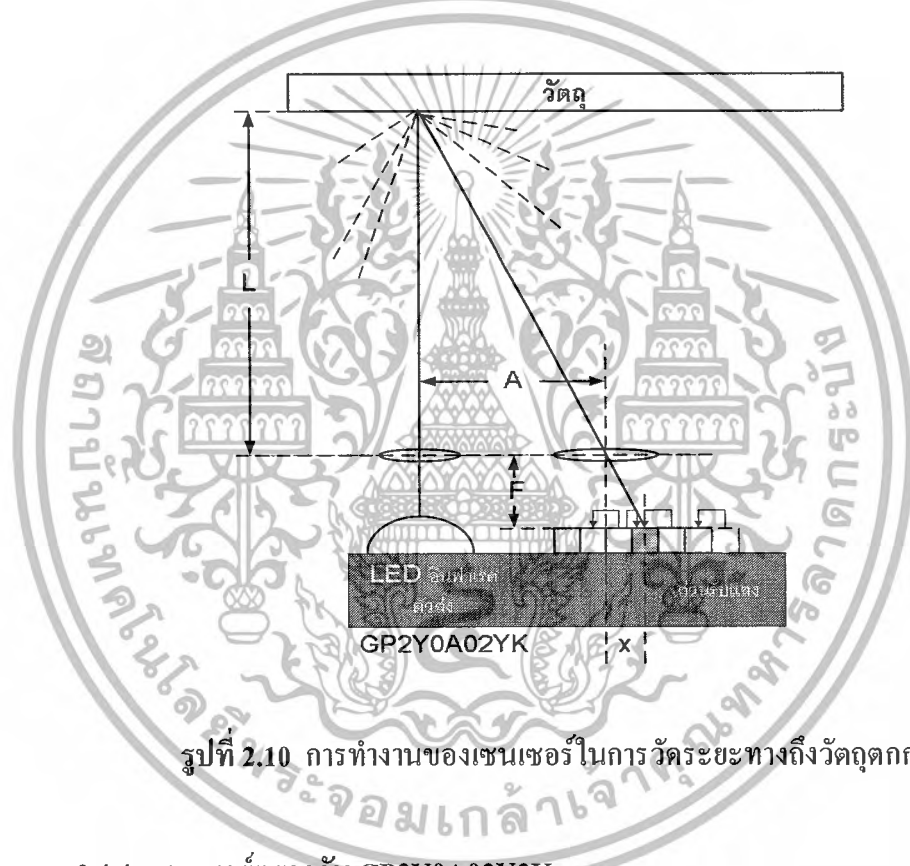
A = ระยะตั้งฉากระหว่างจุดศูนย์กลางเลนส์ LED อินฟราเรดตัวส่งถึงจุดศูนย์กลางเลนส์

ส่วนรับแสง

F = ระยะห่างระหว่างจุดศูนย์กลางเลนส์ส่วนรับแสงที่ตั้งฉากถึงส่วนรับแสง

X = ระยะห่างระหว่างจุดศูนย์กลางเลนส์ถึงจุดที่แสงตกกระทบบ

โดยค่าที่วัดได้จากโฟโตทรานซิสเตอร์จะถูกส่งไปยังวงจรประมวลผลสัญญาณ ก่อนจะเปลี่ยนค่าเป็นระดับแรงดัน ซึ่งให้ผลการเปลี่ยนแปลงแรงดันตามระยะทางที่ตรวจวัดได้

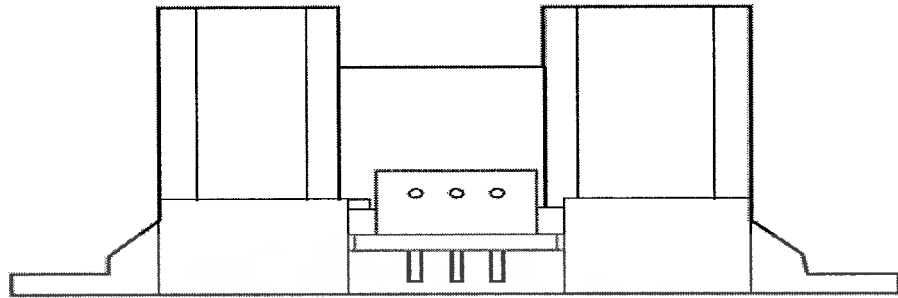


รูปที่ 2.10 การทำงานของเซนเซอร์ในการวัดระยะทางถึงวัตถุตกกระทบบ

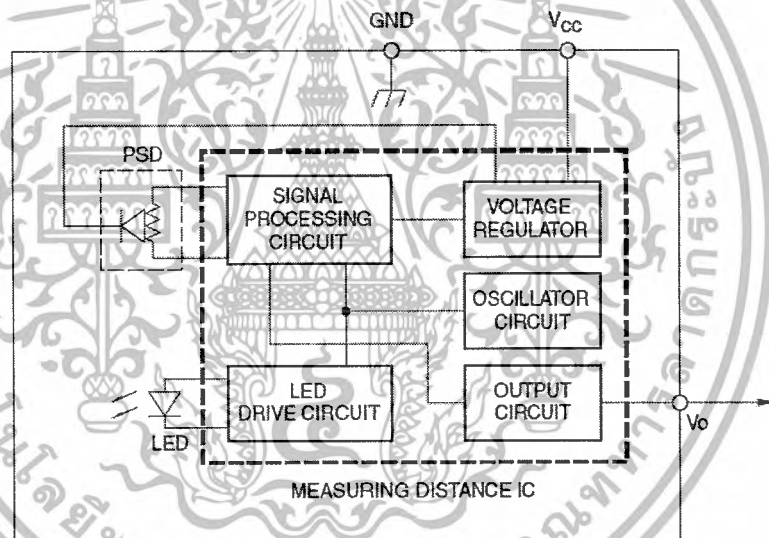
2.4.4 เซนเซอร์ตรวจจับ GP2Y0A02Y2K

คุณสมบัติเซนเซอร์ตรวจจับวัตถุ GP2Y0A02Y2K ของบริษัท SHARP สี่ผิวของวัตถุต่างๆ จะมีผลกระทบน้อยมากต่อการสะท้อนสัญญาณของตัวเซนเซอร์แต่จะมีผลบ้างในวัตถุที่มีสีดำ ซึ่งเป็นสีที่ทำให้การสะท้อนของสัญญาณอินฟราเรดทำได้ไม่ดีนัก ระยะทางในการตรวจจับวัตถุ GP2Y0A02Y2K ให้เอาต์พุตออกมาเป็นแรงดัน ซึ่งเป็นอัตราส่วนกับระยะทาง โดยมีระยะทางในการตรวจจับมีความยาว 20 ถึง 150 ซม. โดยไม่จำเป็นต้องมีวงจรควบคุมภายนอกเพียงแค่อำยไฟเลี้ยง (VCC , GND) ก็สามารถนำเอาต์พุตของตัวเซนเซอร์ไปใช้งานได้เลย ซึ่งแรงดัน VCC ที่ใช้

จะอยู่ในช่วง 4.5 ถึง 5.5 V โดยปกติจะใช้ที่ 5 V แต่สามารถทนแรงดันต่ำสุด และสูงสุดได้ -0.3 ถึง +7 V



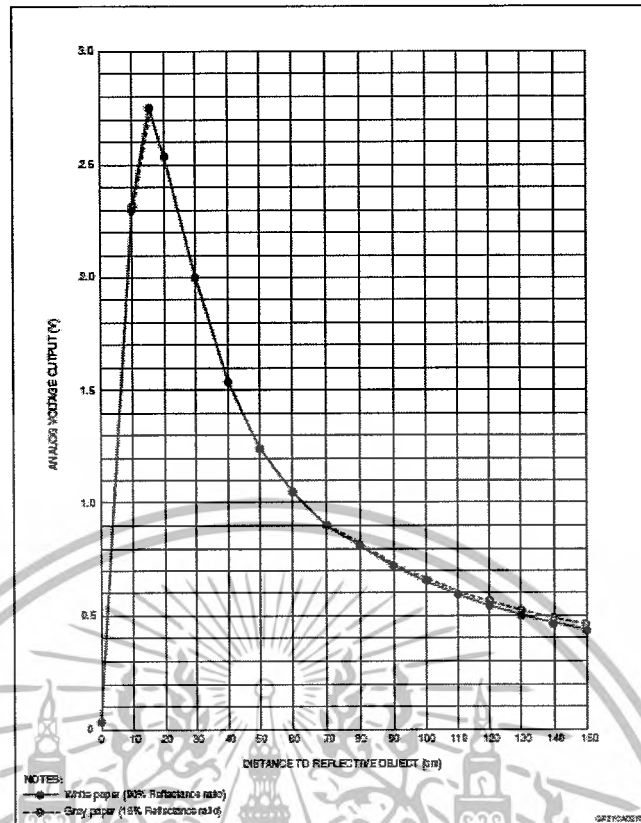
รูปที่ 2.11 เซนเซอร์ตรวจจับ GP2Y0A02Y2K



รูปที่ 2.12 บล็อกไดอะแกรมภายในของ GP2Y0A02Y2K

จากรูปที่ 2.12 เป็นการอธิบายหลักการทำงานของเซนเซอร์ ภายในบล็อกไดอะแกรม GP2Y0A02Y2K เมื่อทำการจ่ายไฟแรงดันกระแสตรง +5 V ให้กับเซนเซอร์ก็จะทำการแปลงสัญญาณความถี่ให้อยู่ในช่วงความถี่อินฟราเรด โดยมีตัวกำเนิดความถี่ภายในอุปกรณ์เซนเซอร์และส่งความถี่นั้นด้วย LED เป็นภาคส่ง และทำการรับความถี่ที่ตกกระทบเพื่อทำการเปรียบเทียบความถี่ว่ามีความถี่ตรงกับช่วงที่ส่งไปหรือไม่ โดยจะให้เอาต์พุตที่ออกมาเป็นแรงดันกระแสตรง 0 - 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ความสัมพันธ์ระหว่างแรงดันเอาต์พุตกับระยะทาง

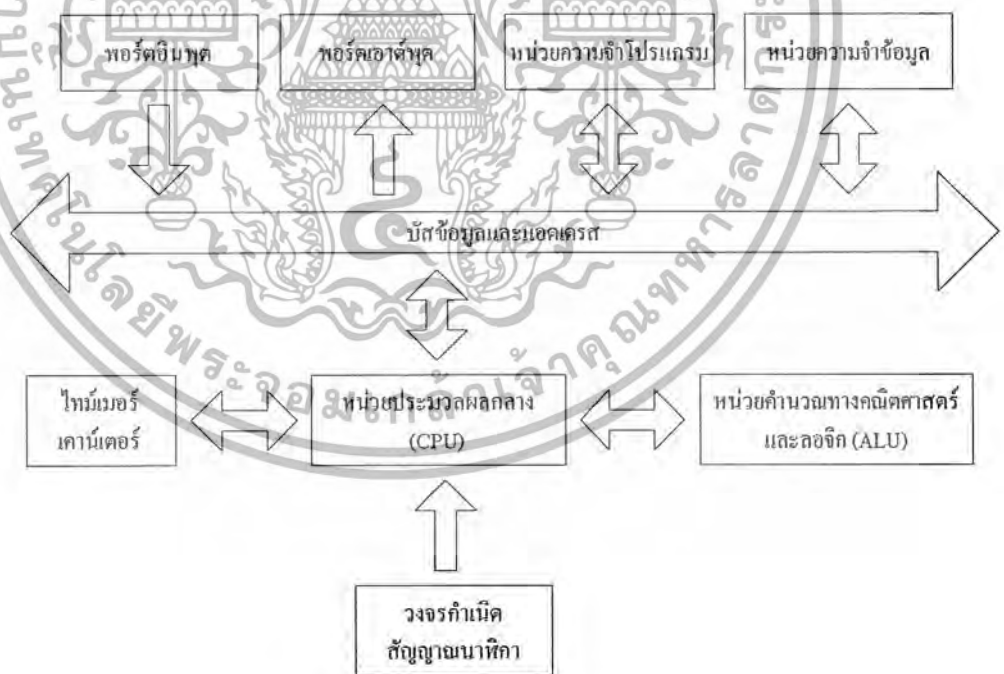
จากรูปที่ 2.13 เป็นการแสดงให้เห็นถึงความสัมพันธ์ระหว่างแรงดันเอาต์พุตกับระยะทางของเซนเซอร์ จะเห็นได้ว่าเซนเซอร์สามารถตรวจจับวัตถุในระยะทางตั้งแต่ 15 - 150 เซนติเมตร โดยจะให้เอาต์พุตที่ออกมาเป็นแรงดันกระแสตรงตั้งแต่ 0.5 - 2.75 โวลต์

2.5 หลักการของไมโครคอนโทรลเลอร์

2.5.1 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คำหนึ่งคือ ไมโคร (Micro) หมายถึงขนาดเล็กและคำว่าคอนโทรลเลอร์ (Controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ได้ทำการบรรจุความสามารถที่ทำงานเสมือนกับระบบคอมพิวเตอร์ที่คนส่วนใหญ่คุ้นเคย นั่นคือภายในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู หน่วยความจำและพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกันโดยที่สามารถเขียน โปรแกรมเพื่อกำหนดรูปแบบการทำงาน และการควบคุมได้อย่างอิสระตามความต้องการ

การควบคุมไมโครคอนโทรลเลอร์นั้นกระทำโดยผ่านกระบวนการควบคุม โดยโปรแกรมที่เขียนขึ้น เพื่อบอกถึงการทำงานของไมโครคอนโทรลเลอร์ โดยลักษณะที่ถูกระบุขึ้น โดยที่ผู้เขียนโปรแกรมควบคุม ซึ่งควบคุมการทำงานทั้งหมดของไมโครคอนโทรลเลอร์ในการกำหนดพอร์ตให้เป็นอินพุตหรือเอาต์พุต และยังสามารถกำหนดหน่วยความจำภายในซึ่งเป็นที่เก็บข้อมูล และเป็นที่พักข้อมูลตามความต้องการ โดยในการทำงานของไมโครคอนโทรลเลอร์แต่ละคำสั่งจะอ้างอิงเวลาจากสัญญาณนาฬิกาที่ส่งให้กับไมโครคอนโทรลเลอร์เป็นหลัก



รูปที่ 2.14 การทำงานของไมโครคอนโทรลเลอร์

2.5.2 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 หรือ Intel 8051 เป็นไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นโดยบริษัท Intel เมื่อปี ค.ศ. 1980 และได้รับความนิยมอย่างสูงในขณะนั้นจนถึงปัจจุบัน ก็ยังคงมีใช้กันอยู่อย่างแพร่หลาย MCS-51 เป็นไมโครคอนโทรลเลอร์ที่มีสถาปัตยกรรม และฟังก์ชันการใช้งานไม่ซับซ้อนเท่าใดนัก และยังเป็นไมโครคอนโทรลเลอร์ที่มีความเร็วในการทำงานต่ำ ทำให้ในปัจจุบันความนิยมใช้ MCS-51 จึงลดลง แต่อย่างไรก็ตาม MCS-51 ก็เหมาะที่จะเป็นไมโครคอนโทรลเลอร์ตัวแรกๆ ที่จะช่วยให้เราศึกษาเรียนรู้พื้นฐานการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์ได้เป็นอย่างดี

2.5.3 การเขียนโปรแกรมไมโครคอนโทรลเลอร์ MCS-51 โดยใช้ภาษาซี

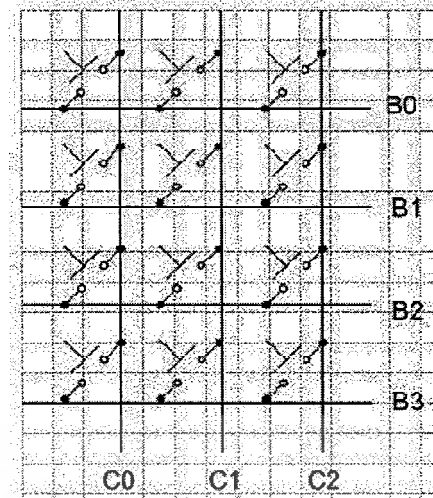
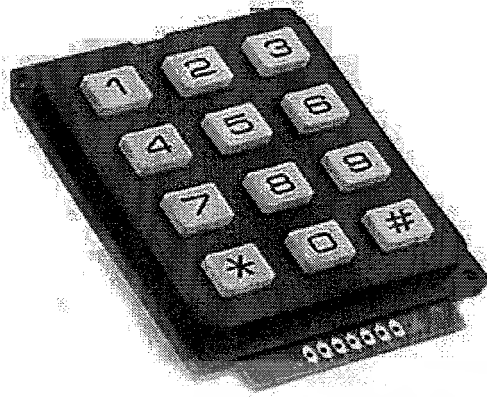
โปรแกรมภาษาซีนั้นเป็นภาษาระดับสูงที่มีความใกล้เคียงกับภาษามนุษย์ ทำให้มีการทำความเข้าใจได้ง่าย นอกจากนั้นแล้วในการเขียนโปรแกรมภาษาซีก็ไม่ต้องจำเป็นเข้าใจโครงสร้างภายในของไมโครคอนโทรลเลอร์อย่างละเอียด เพียงแต่เข้าใจในการเขียนโปรแกรมแบบโครงสร้างก็เพียงพอแล้ว โปรแกรมภาษาซีสามารถใช้ในการเข้าถึงโครงสร้างภายในของไมโครคอนโทรลเลอร์ได้โดยตรง ทำให้โปรแกรมที่เขียนขึ้นทำงานได้รวดเร็ว ดังนั้นโปรแกรมภาษาซีจึงเป็นที่นิยมแพร่หลายในการเขียนโปรแกรมไมโครคอนโทรลเลอร์ คอมพิวเตอร์ที่ใช้ในการแปลภาษาซี เป็นภาษาเครื่องมีอยู่มากมาย แต่ได้ทำการเลือกใช้คอมพิวเตอร์ที่ใช้โปรแกรมเคลล์ นิวเวอร์ชัน 3 (Keil uVision 3)

2.5.4 การคอมไพล์เลอร์โดยใช้โปรแกรมเคลล์ นิวเวอร์ชัน 3 (Keil uVision 3)

โปรแกรมเคลล์ นิวเวอร์ชัน 3 (Keil uVision 3) ถูกสร้างขึ้นโดยบริษัท เคลล์ แอนอาร์ม (Keil TM, an ARM Company) เป็นโปรแกรมคอมไพล์เลอร์ภาษาซี สำหรับ MCS-51 โดยสามารถเขียนและทำการคอมไพล์ไฟล์ .Hex ได้ในตัว ซึ่งในเวอร์ชันนี้เป็นเวอร์ชันทดลองใช้งาน สามารถเขียนโปรแกรมที่รองรับความจุ 2 Kbyte ซึ่งเพียงพอกับโปรแกรมที่ไม่ยุ่งยากซับซ้อนจนเกินไป เราสามารถดาวน์โหลดไฟล์การติดตั้งโปรแกรมเคลล์ นิวเวอร์ชัน 3 (Keil uVision 3) ได้จาก www.keil.com

2.5.5 การรับค่าจาก Keypad

Keypad คืออุปกรณ์ที่นำเอาสวิตช์หลายๆ ตัวมาต่อกันแบบเมตริกซ์ ซึ่งจะช่วยลดจำนวนสายสัญญาณของข้อมูลให้น้อยลง ในขณะใช้สวิตช์จำนวนมากๆ แถบยังลดขนาดของวงจรให้มีขนาดเล็กลงด้วย Keypad จึงเปรียบเสมือนคีย์บอร์ดเล็กๆ นั่นเอง ลักษณะโครงสร้างของ Keypad แสดงดังรูปที่ 2.15



รูปที่ 2.15 ลักษณะโครงสร้างของ Keypad

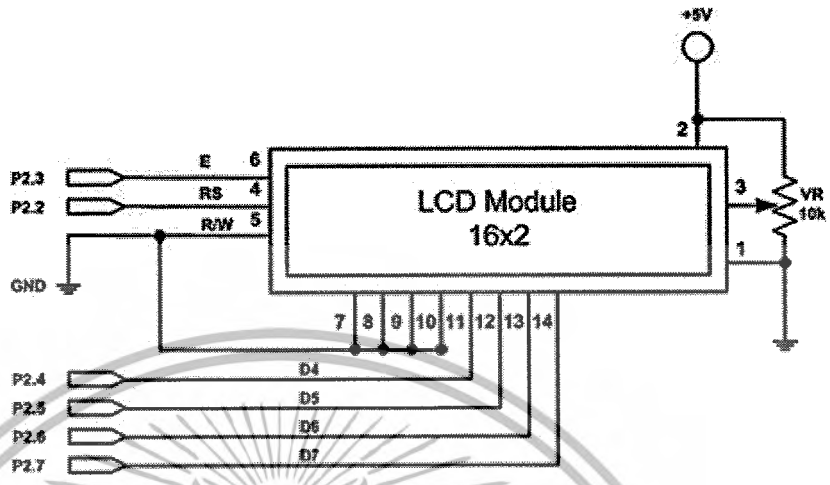
2.5.6 LCD Module

LCD ย่อมาจาก Liquid Crystal Display เป็นอุปกรณ์แสดงผลที่นิยมใช้กันอย่างแพร่หลาย เช่น เครื่องคิดเลข โทรศัพท์ และเครื่องมือวัดต่าง ๆ เพราะ LCD สามารถที่แสดงผลได้ทั้งตัวเลข ตัวอักษร และภาพกราฟฟิก เหตุผลที่เรียกอุปกรณ์นี้ว่า LCD Module นั่นก็คือ ในการใช้งาน LCD นั้นจำเป็นต้องมีวงจรควบคุมการทำงานอีกส่วนหนึ่ง ซึ่งโดยปกติแล้วก็จะรวมอยู่ด้วยกันกับจอ LCD จึงเรียกรวมกันว่า LCD Module ลักษณะของ LCD Module แสดงดังรูปที่ 2.16



รูปที่ 2.16 LCD Module 16 ตัวอักษร 2 บรรทัด

ในการเชื่อมต่อ LCD Module เข้ากับไมโครคอนโทรลเลอร์ จะใช้การส่งงานผ่านขาทั้ง 16 ขา ของ LCD Module ซึ่งแสดงรายละเอียดหน้าที่ของแต่ละขาได้ดังตารางที่ 2.2



รูปที่ 2.17 ตำแหน่งขาและหน้าที่ของ LCD Module

ตารางที่ 2.2 ตำแหน่งขาและหน้าที่ของ LCD Module

ตำแหน่งขา	รายละเอียดการทำงาน
1	ขา GND ต่อกับราวน์ด์ 0 volt
2	ขา VCC ต่อไฟเลี้ยง +5 V
3	ขาสัญญาณอินพุตเพื่อรับแรงดันในการปรับความเข้มของจอ LCD
4	ขาสัญญาณ RS หากเป็น 1 คือการรับข้อมูลเพื่อแสดงผล หากเป็น 0 คือการรับข้อมูลเพื่อควบคุมการทำงานของ LCD
5	ขาสัญญาณ R/W หากเป็น 1 จะเป็นการอ่านข้อมูลจาก LCD หากเป็น 0 จะเป็นการเขียนข้อมูลลงบนจอ LCD
6	ขาสัญญาณ E ให้ LCD ทำงานโดยสัญญาณไม่น้อยกว่า 2 ms
7 – 14	ขาสัญญาณ DB0-DB7 ทำหน้าที่เป็นตัวกลางในการส่งและรับข้อมูล
15	ขาไฟ (+) สำหรับแสง Backlight
16	ขาไฟ (-) สำหรับแสง Backlight

นอกจากนั้นแล้ว ยังต้องรู้คำสั่งที่ส่งไปควบคุม LCD Module อีกด้วย ซึ่งรายละเอียดคำสั่งทั้งหมดแสดงดังตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโมดูล LCD มีโหมดการทำงานหลัก 2 โหมดคือ โหมด 8 บิต ที่ใช้สัญญาณข้อมูล D0 ถึง D7 ในการรับคำสั่งหรือข้อมูลจากไมโครคอนโทรลเลอร์ และโหมด 4 บิต ซึ่งจะใช้สัญญาณข้อมูล D4 ถึง D7 ในการรับคำสั่งหรือข้อมูล

การทำงานในโหมด 4 บิต จะช่วยให้สามารถประหยัดขาพอร์ตของไมโครคอนโทรลเลอร์ที่ใช้ในการติดต่อ แต่จะทำงานได้ช้ากว่าเนื่องจากต้องแบ่งข้อมูล 8 บิต ออกเป็น 4 บิต 2 ชุดแล้วส่งสองครั้ง ในการทดลองนี้เป็นการต่อให้โมดูล LCD ทำงานในโหมด 4 บิต โดยมีการจัดวงจรการต่อกับโมดูล POP-168 ได้

วงจรได้ต่อขา R/W ลงกราวด์ จึงทำให้สามารถเขียนค่าให้กับโมดูล LCD เท่านั้น ไม่สามารถอ่านค่าสถานะโมดูล LCD กลับมาได้

ในการต่อใช้งานถ้าไม่สามารถหาตัวต้านทานปรับค่าได้ ให้ต่อขา V_0 ของโมดูล LCD ลงกราวด์ ซึ่งในกรณีนี้คือ ปรับค่าความคมชัดของจอแสดงผลให้อยู่ในตำแหน่งสูงสุดเท่านั้น ไม่สามารถปรับค่าความคมชัดได้

ในการควบคุมให้โมดูล LCD แสดงตัวอักษร เริ่มต้นด้วยการเขียนคำสั่งควบคุมการทำงานก่อน เพื่อกำหนดรูปแบบการแสดงผล จากนั้นก็ส่งคำสั่งรหัสตัวอักษรตามรหัส ASCII ให้กับโมดูล LCD เพื่อนำไปแสดงบนจอแสดงผล โดยมีคำสั่งควบคุมแสดงในตารางที่ 2.3 ส่วนลำดับขั้นตอนการติดต่อสามารถอธิบายเป็นขั้นตอนได้ดังนี้

1. ในการเขียนคำสั่งควบคุมจอแสดงผล LCD จะต้องให้ขา RS และ R/W ค่าลอจิกเป็น 0 ทั้งคู่
2. ในการส่งรหัสตัวอักษรให้จอแสดงผล LCD จะต้องให้ขา RS มีค่าลอจิกเป็น 1 และขา R/W มีค่าลอจิกเป็น 0
3. เมื่อเขียนข้อมูลไปแล้ว ต้องตามด้วยการส่งสัญญาณพัลส์เอ็นเอเบิลไปยังขา E ด้วย เพื่อกระตุ้นให้โมดูล LCD รับรู้ถึงคำสั่งและข้อมูลแสดงผลที่ส่งไปให้

ตารางที่ 2.3 คำสั่งของโมดูล LCD มาตรฐานแบบตัวอักษร

ข้อมูลคำสั่ง	การทำงาน
0x01	ล้างหน่วยแสดงผล
0x02	ย้ายเคอร์เซอร์กลับตำแหน่งซ้ายสุด
0x04	แสดงผลโดยเลื่อนเคอร์เซอร์ไปทางซ้าย
0x05	เลื่อนไปทางขวา
0x06	แสดงผลโดยเลื่อนเคอร์เซอร์ไปทางขวา
0x07	เลื่อนไปทางซ้าย
0x08	ปิดการแสดงผล ไม่แสดงเคอร์เซอร์
0x0A	ปิดการแสดงผล แต่แสดงเคอร์เซอร์
0x0C	แสดงผลแต่ไม่แสดงเคอร์เซอร์
0x0E	แสดงผล และแสดงเคอร์เซอร์
0x0F	แสดงผล และแสดงเคอร์เซอร์กระพริบ
0x10	เลื่อนเคอร์เซอร์ไปทางซ้าย
0x14	เลื่อนเคอร์เซอร์ไปทางขวา
0x18	เลื่อนตัวอักษรใหม่ไปทางซ้าย
0x1C	เลื่อนตัวอักษรใหม่ไปทางขวา
0x80	ตำแหน่งเริ่มต้นบรรทัดแรก
0x28	โหมด 4 บิต 2 บรรทัด ขนาดตัวอักษร 5 x 7 จุด
0x38	โหมด 8 บิต 2 บรรทัด ขนาดตัวอักษร 5 x 7 จุด
0xC0	ตำแหน่งเริ่มต้นบรรทัดที่สอง

4. เมื่อเริ่มจ่ายไฟให้โมดูล LCD วงจรภายในจะกำหนดค่าการติดต่อเป็นแบบ 8 บิต ทางผู้ผลิตโมดูล LCD กำหนดไว้ว่า หากต้องการติดต่อแบบ 4 บิตจะต้องส่งข้อมูล 0x32 ตามด้วย 0x33 จากนั้นส่งข้อมูลคำสั่ง 0x28 เพื่อกำหนดให้โมดูล LCD ทำงานในโหมด 4 บิต แสดงผล 2 บรรทัด และมีความละเอียดของอักษรแต่ละตัวเท่ากับ 5x7 จุด โดยในการส่งข้อมูลคำสั่งแต่ละค่านั้น หลังจากส่งข้อมูลแล้วต้องมีการหน่วงเวลาเล็กน้อย

5. หลังจากนั้นผู้ใช้งาน สามารถเขียนข้อมูลคำสั่ง เพื่อกำหนดรูปแบบการทำงานได้ตามต้องการ ไม่ว่าจะเป็นการกำหนดให้โมดูล LCD แสดงเคอร์เซอร์ (ตัวแถบดำ) หรือไม่ และถ้าให้แสดงจะให้กระพริบหรือไม่ กำหนดรูปแบบของการเลื่อนข้อมูล เปิดหรือปิดจอแสดงผล เป็นต้น

6. เมื่อกำหนดการทำงานของโมดูล LCD เสร็จ ขั้นตอนถัดมาเป็นการกำหนดตำแหน่งบนหน้าจอที่จะแสดงผล โดยตำแหน่งที่จะแสดงผลนี้จะเรียกว่า DDRAM (Display Data RAM) หรือหน่วยความจำแรม สำหรับการแสดงผลแอดเดรส DDRAM เพื่อกำหนดตำแหน่งที่จะแสดงข้อความ จะเริ่มต้นที่มุมบนซ้ายมือคือตำแหน่งที่ 0 (0x00) นับเพิ่มค่าไปทางขวามือโดยมีค่าสูงสุดไม่เกิน 39 คอลัมน์ (0x27) สำหรับบรรทัดที่ 2 จะเริ่มที่แอดเดรส 0x40 จนถึงค่าสูงสุดที่ 0x67

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53

แอดเดรสของ DDRAM (ฐานสิบหก) บรรทัดที่ 1 (บน)

แอดเดรสของ DDRAM (ฐานสิบหก) บรรทัดที่ 2 (ล่าง)

*กรณี LCD 16 ตัวอักษร 2 บรรทัด จะมีแอดเดรส DDRAM ถึง 0x0F สำหรับบรรทัดที่ 1 (บน)

และ 0x4F สำหรับบรรทัดที่ 2 (ล่าง)

รูปที่ 2.18 ตำแหน่งแอดเดรสของ DDRAM ของโมดูล LCD

โดยที่โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด จะมี DDRAM ให้ใช้ 32 ตำแหน่ง แบ่งเป็น 2 ช่วง คือ 00 ถึง 0F และ 40 ถึง 4F

2.5.7 พอร์ตอนุกรม

พอร์ตอนุกรม คือ พอร์ตที่ใช้สำหรับการติดต่อกันระหว่างไมโครคอนโทรลเลอร์กับไมโครคอนโทรลเลอร์ หรือระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ ซึ่งใช้วิธีการสื่อสารกันด้วยข้อมูลแบบอนุกรม การรับและส่งข้อมูลแบบอนุกรมยังแบ่งออกเป็นลักษณะการใช้งานได้ 3 แบบ คือ

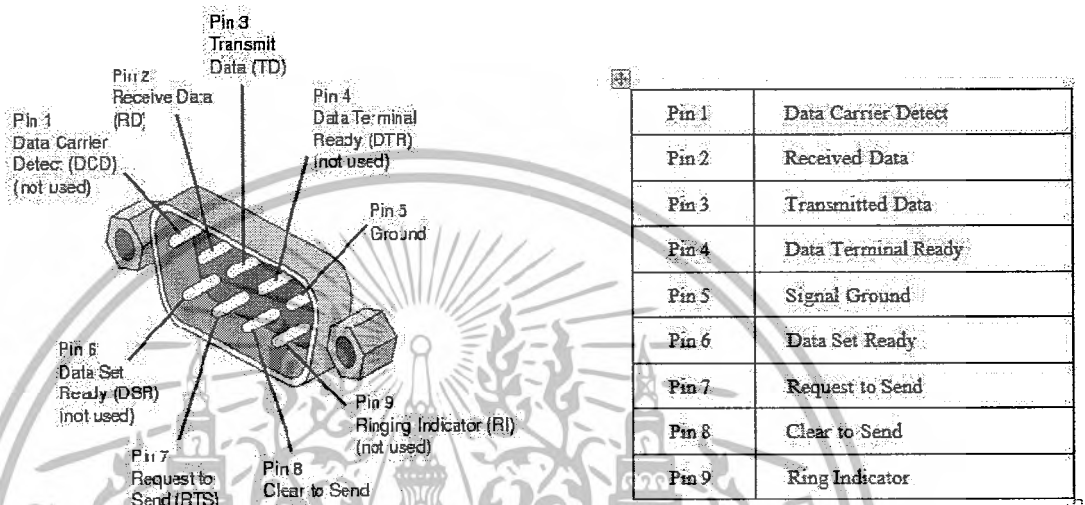
1. แบบซิมเพลกซ์ (Simplex) เป็นการส่งหรือรับข้อมูลแบบทิศทางเดียวเท่านั้น

2. แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลแบบสลับกัน

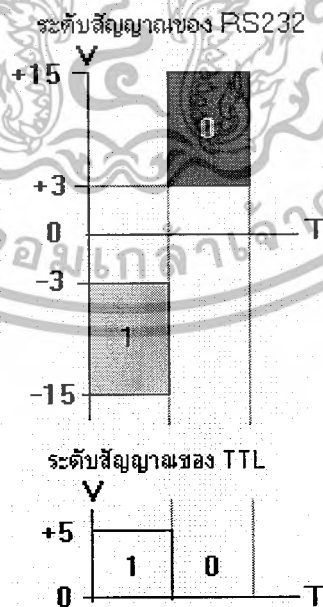
คือเมื่อด้านหนึ่งส่ง อีกด้านหนึ่ง เป็นฝ่ายรับ สลับกัน ไม่สามารถรับ-ส่ง ในเวลาเดียวกันได้

3. แบบฟูลดูเพลกซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

ในคอมพิวเตอร์หนึ่งเครื่องจะมีพอร์ตอนุกรม (Serial Port) หรือที่เรียกกันติดปากว่า Comport ด้วยกันหลายชนิด เช่น DB9, DB25, DB15 ซึ่งการนำไปใช้งานก็จะแตกต่างกันไปตามแต่ลักษณะอุปกรณ์ที่ต่อพ่วง วิธีการเชื่อมต่อสายสัญญาณจะใช้มาตรฐานการรับ-ส่งแบบ RS-232C โดยใช้วิธีการไขว้สาย เพื่อให้เกิดการรับ-ส่งข้อมูลระหว่างกัน ซึ่งสายที่ใช้ในการไขว้ก็คือ สาย pin2 (RXD) กับ pin3 (TXD) นั่นเอง ตำแหน่งการใช้งานของหัวต่อ DB9 แสดงดังรูป



รูปที่ 2.19 ตำแหน่งการใช้งานของหัวต่อ DB9



รูปที่ 2.20 ระดับสัญญาณของ RS-232C และระดับสัญญาณของ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณรบกวนที่เกิดขึ้น ในสายนำสัญญาณมักจะมีแรงดันเป็นบวก เมื่อเทียบกับกราวด์ เพื่อป้องกันสัญญาณรบกวนนี้จึงออกแบบแรงดันของลอจิก 1 เป็นลบคือ อยู่ในช่วง -3V ถึง -15V ส่วนแรงดันของลอจิก 0 อยู่ในช่วง +3V ถึง +15V และเหตุที่ระดับสัญญาณของ RS-232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น ดังนั้นจึงจำเป็นต้องมีวงจรเปลี่ยนระดับแรงดันของ RS-232 มาเป็นระดับแรงดันของ TTL

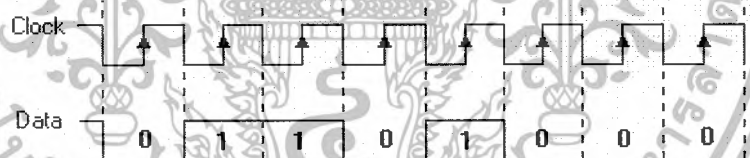
อัตราการส่งข้อมูล (Baud rate) คือ ความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาที เช่น 300 1,200 2,400 4,800 9,600 14,400 19,200 38,400 และ 56,000 เป็นต้น การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทาง และปริมาณสัญญาณรบกวน

รูปแบบการสื่อสารแบบอนุกรม มีด้วยกันอยู่ 2 แบบ คือ

1. แบบซิงโครนัส (Synchronous)
2. แบบอะซิงโครนัส (Asynchronous)

การสื่อสารแบบซิงโครนัส (Synchronous)

การรับ-ส่งข้อมูล จะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนด จังหวะเวลา การส่งข้อมูล ร่วมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด

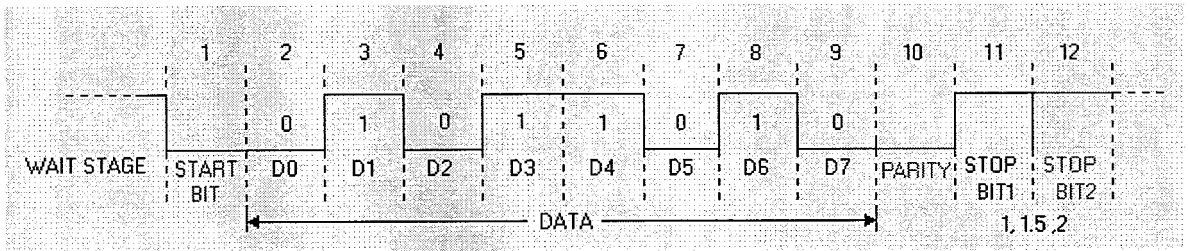


รูปที่ 2.21 การสื่อสารแบบอะซิงโครนัส (Asynchronous)

การสื่อสารแบบอะซิงโครนัส (Asynchronous)

การรับ-ส่งข้อมูล โดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิกา ร่วมด้วย แต่จะใช้ให้ตัวส่งและตัวรับ มี อัตราส่งข้อมูลที่เท่ากัน รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูล (Data) มีขนาด 5 บิต 6 บิต 7 บิต หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity bit) มีขนาด 1 บิต หรือ ไม่มี
4. บิตหยุด (Stop bit) มีขนาด 1 บิต 1.5 บิต 2 บิต



รูปที่ 2.22 การรับ-ส่งข้อมูลแบบอะซิงโครนัส

เมื่อไม่มีการส่งข้อมูล ขาดata จะมีสถานะเป็นลอจิก 1 หรือ สถานะหยุดรอ (Waiting stage) เมื่อเริ่มต้นส่งข้อมูลจะให้ขาดata เป็นลอจิก 0 เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น(Start bit) จากนั้นก็จะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำไปก่อน (LSB) แล้วตามด้วยพริตตี้บิต (จะมีหรือไม่ มีก็ได้ ขึ้นอยู่กับการติดตั้งค่า ของทั้งสองฝ่าย) สุดท้ายตามด้วยลอจิก 1 อย่างน้อย 1 บิต (มีขนาด 1 บิต 1.5 บิตหรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

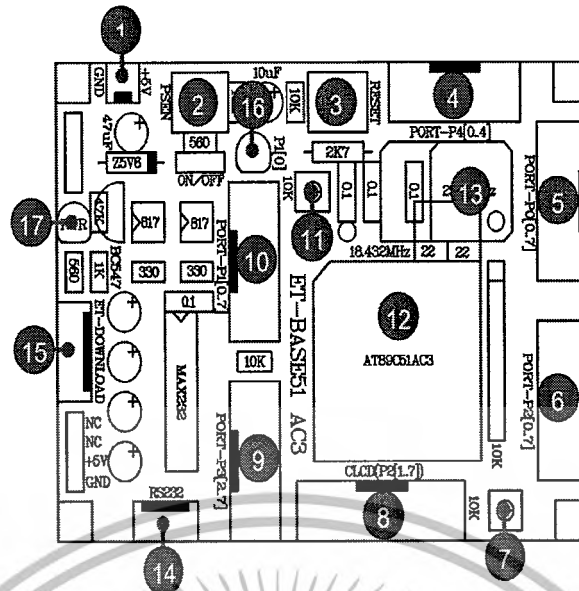
2.5.8 บอร์ดไมโครคอนโทรลเลอร์ ET-BASE51 AC3

ET-BASE51 AC3 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ขนาด 52 Pin ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C51AC3 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 52 Pin PLCC โดย MCU ตัวนี้จะมีจุดเด่น คือ เรื่องของความเร็วในการประมวลผลซึ่งสามารถทำงานได้ด้วยความถี่สูงสุด 60 MHz ที่ 12 Clock / 1 Machine Cycle นอกจากนี้ยังมีความเข้ากันได้กับอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 2 KByte หรือหน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 2304 Byte (2048+256) ในส่วนทางด้านของอุปกรณ์ที่อยู่บริเวณภายนอกนั้น ก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้ง SPI, UART, Watchdog, Timer/Counter, PWM และ ADC โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม

คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์ ET-BASE51 AC3

เลือกใช้ MCU ตระกูล MCS-51 เบอร์ AT89C51AC3 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งสามารถกำหนดการทำงานของ MCU ให้ทำงานในโหมดความเร็ว 2 เท่า (X2 Mode) ได้ ทำให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 58.9824 MHz โดยคุณสมบัติเด่นๆของ MCU ได้แก่

- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรมขนาด 64 Kbyte
- มี EEPROM ขนาด 2 KByte สำหรับเก็บข้อมูล และสามารถเขียนซ้ำได้กว่า 1 ล้านครั้ง
- มีพอร์ต I/O ขนาด 8 บิต จำนวน 5 พอร์ต (P0, P1, P2, P3 และ P4(5Bit))
- มี RAM ใช้งาน 2304 Byte (ERAM 2048 Byte + IRAM 256 Byte)
- มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
- มีวงจร Timer/Counter ขนาด 16 บิต จำนวน 3 ชุด
- มีวงจร ADC ขนาด 10 บิต จำนวน 8 ช่อง (ใช้ Port-P1 โดยกำหนดจากโปรแกรม)
- มีวงจร Watchdog, Power-ON Reset, Capture/Compare, PWM
- มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 5 ชุด (P0, P1, P2, P3 และ P4)
- มีขั้วต่อ LCD แบบ Header 2x7 รองรับการเชื่อมต่อกับ LCD Character (เชื่อมต่อแบบ 4 บิต)
- มีขั้วต่อใช้งาน RS-232 สำหรับใช้งาน และ ET-DOWNLOAD สำหรับ Download ผ่าน RS-232
- มี LED แสดงสถานะแหล่งจ่าย Power และ Self-Test สำหรับใช้ทดสอบการทำงานของบอร์ด
- ใช้แหล่งจ่ายไฟขนาด +5V DC
- ขนาด PCB Size เล็กเพียง 8 x 6 cm.
- โครงสร้างบอร์ด ET-BASE51 AC3 (AT89C51AC3)

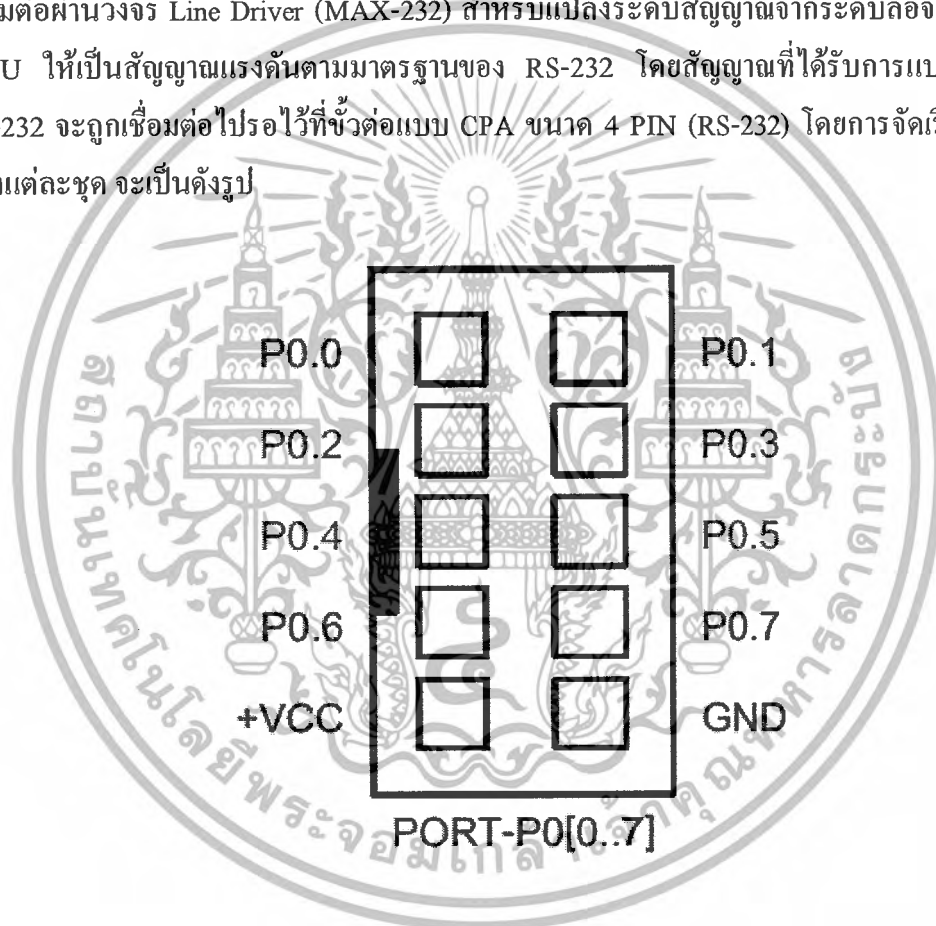


รูปที่ 2.23 โครงสร้างบอร์ด ET-BASE51 AC3 (AT89C51AC3)

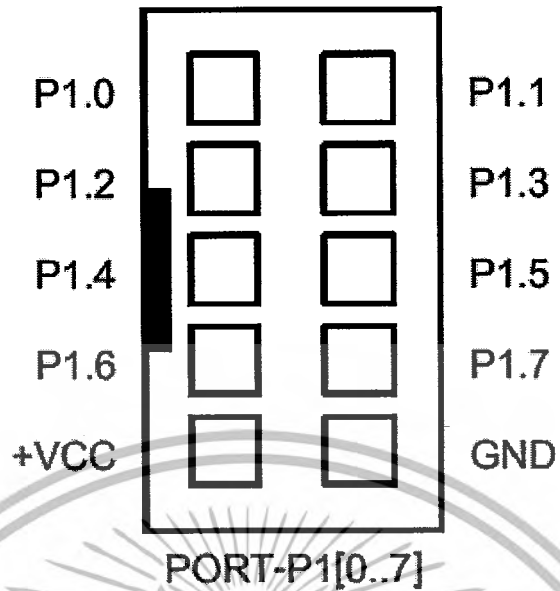
- หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟตรง +5V DC
- หมายเลข 2 เป็น Switch PSEN ใช้ร่วมกับ RESET สำหรับ Download แบบ Manual
- หมายเลข 3 เป็น Switch RESET ใช้สำหรับ Reset การทำงานของ CPU
- หมายเลข 4 เป็น Port-P4 มีขนาด 5 บิต คือ P4.0-P4.4
- หมายเลข 5 เป็น Port-P0 มีขนาด 8 บิต
- หมายเลข 6 เป็น Port-P2 มีขนาด 8 บิต
- หมายเลข 7 เป็นตัวต้านทานสำหรับปรับค่าความสว่างให้ LCD
- หมายเลข 8 เป็น Port-LCD ชนิด Character Type ใช้การเชื่อมต่อแบบ 4 บิต ผ่าน Port-P2
- หมายเลข 9 เป็น Port-P3 มีขนาด 6 บิต (P3.2-P3.7)
- หมายเลข 10 เป็น Port-P1 มีขนาด 8 บิต
- หมายเลข 11 เป็นตัวต้านทานสำหรับปรับค่าแรงดันอ้างอิงของ ADC (3V)
- หมายเลข 12 คือ MCU เบอร์ AT89C51AC3 ซึ่งเป็น MCU ตระกูล MCS-51 จาก ATMEL
- หมายเลข 13 คือ Oscillator Module ค่า 29.4912 MHz
- หมายเลข 14 คือ ขั้วต่อ RS-232 สำหรับใช้งานทั่วไป และ Download แบบ Manual
- หมายเลข 15 คือ ขั้วต่อ ET-DOWNLOAD ใช้สำหรับ Download แบบ Auto
- หมายเลข 16 เป็น LED Self Test (P1.0) ใช้สำหรับทดสอบการทำงานของบอร์ด
- หมายเลข 17 เป็น LED Power ใช้แสดงสถานะของแหล่งจ่ายไฟ +5V DC

ขั้วต่อสัญญาณต่างๆ ของพอร์ต I/O จาก MCU

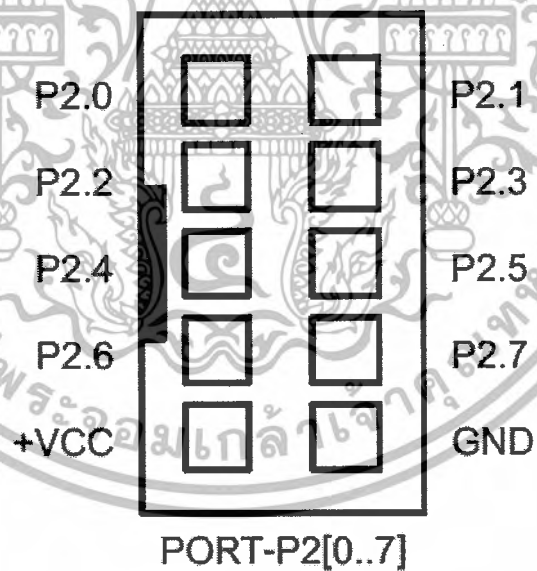
สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้น จะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDE Header ขนาด 10 Pin (2x5) จำนวน 5 ชุด คือ PORT-P0, PORT-P1, PORT-P2, PORT-P3 และ PORT-P4 ตามลำดับ โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยแต่ละพอร์ตจะมีสัญญาณพอร์ตละ 8 บิต ยกเว้น PORT-P3 และ PORT-P4 โดย PORT-P4 มีขนาด 5 บิต (P4.0-P4.4) และ PORT-P3 ซึ่งมีเพียง 6 บิตเท่านั้นคือ P3.2-P3.7 ส่วน P3.0 และ P3.1 จะถูกสงวนไว้ใช้งานเป็นขาสัญญาณ RXD และ TXD สำหรับรับ-ส่งข้อมูลของ RS-232 ซึ่งสัญญาณทั้ง 2 เส้น (P3.0 และ P3.1) จะถูกเชื่อมต่อผ่านวงจร Line Driver (MAX-232) สำหรับแปลงระดับสัญญาณจากระดับลอจิก TTL ของ MCU ให้เป็นสัญญาณแรงดันตามมาตรฐานของ RS-232 โดยสัญญาณที่ได้รับการแปลงเป็นแบบ RS-232 จะถูกเชื่อมต่อไปรอไว้ที่ขั้วต่อแบบ CPA ขนาด 4 PIN (RS-232) โดยการจัดเรียงสัญญาณของแต่ละชุด จะเป็นดังรูป



รูปที่ 2.24 พอร์ต P0 มีขนาด 8 บิต

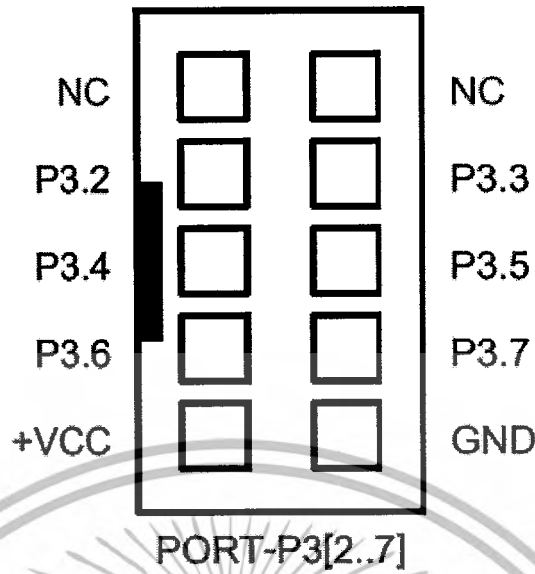


รูปที่ 2.25 พอร์ต P1 มีขนาด 8 บิต

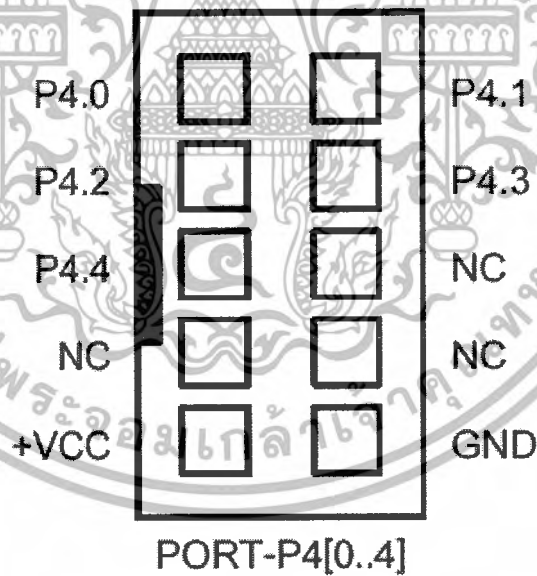


รูปที่ 2.26 พอร์ต P2 มีขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



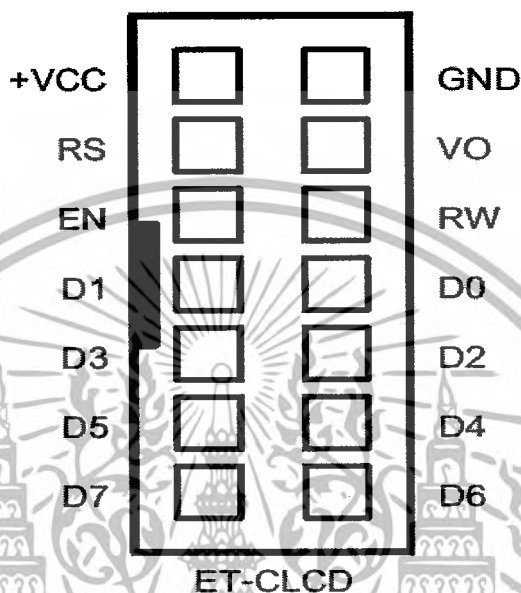
รูปที่ 2.27 พอร์ต P3 มีขนาด 6 บิต



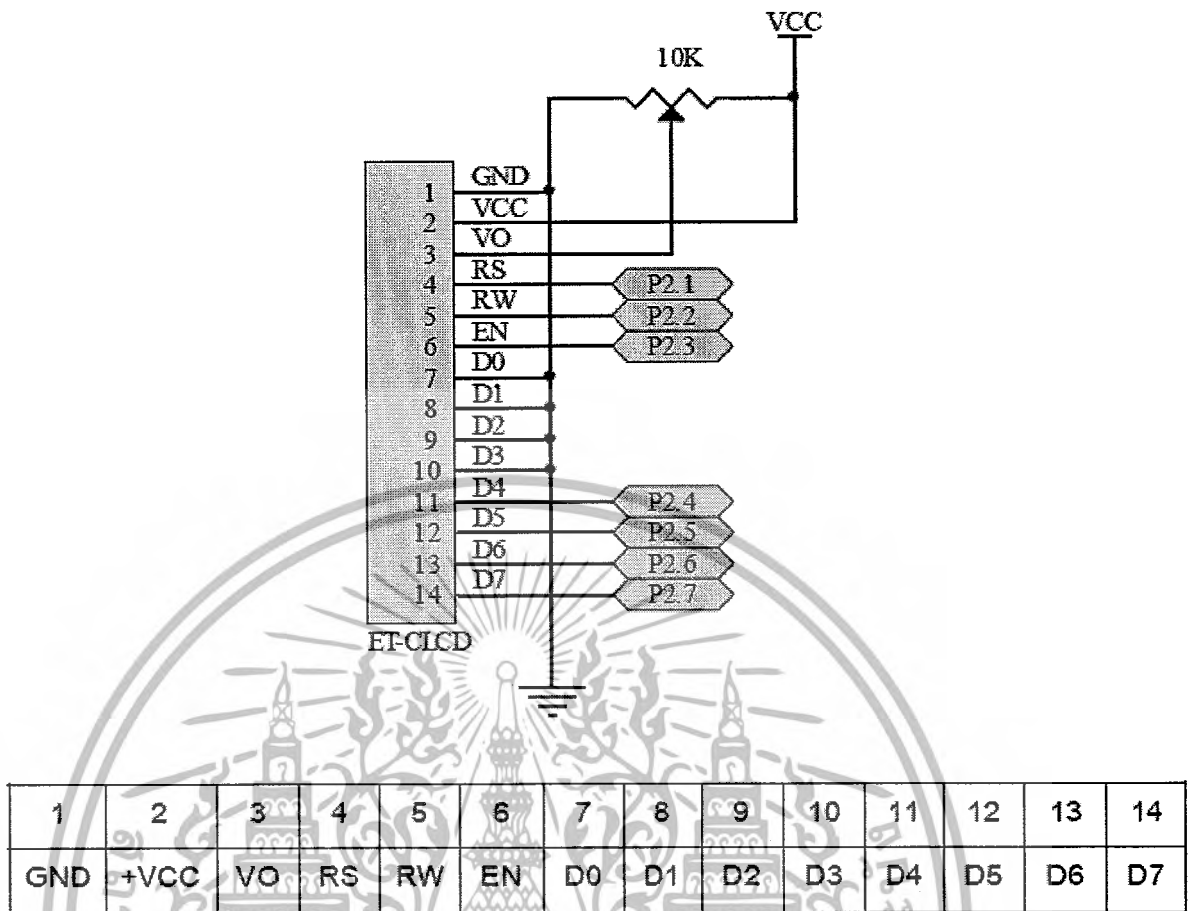
รูปที่ 2.28 พอร์ต P4 มีขนาด 5 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต CLCD ใช้กับ Character LCD โดยเชื่อมต่อแบบ 4 บิต โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะเป็นสัญญาณชุดเดียวกับที่ต่อไปยังขั้วต่อของ PORT-P2 โดยในการเชื่อมต่อสายสัญญาณจากขั้วต่อของพอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น

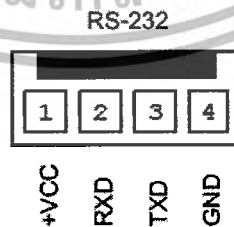


รูปที่ 2.29 พอร์ต CLCD



รูปที่ 2.30 การจัดเรียงขาสัญญาณของ Character LCD มาตรฐาน

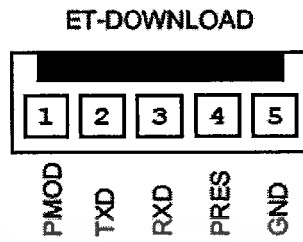
พอร์ต RS-232 เป็นสัญญาณ RS-232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX-232 เรียบร้อยแล้วสามารถใช้เชื่อมต่อกับสัญญาณ RS-232 เพื่อรับ-ส่งข้อมูล อีกทั้งแล้วยังสามารถใช้งานร่วมกับ Switch PSEN และ Switch RESET เพื่อทำการ Download แบบ Manual ได้ด้วย



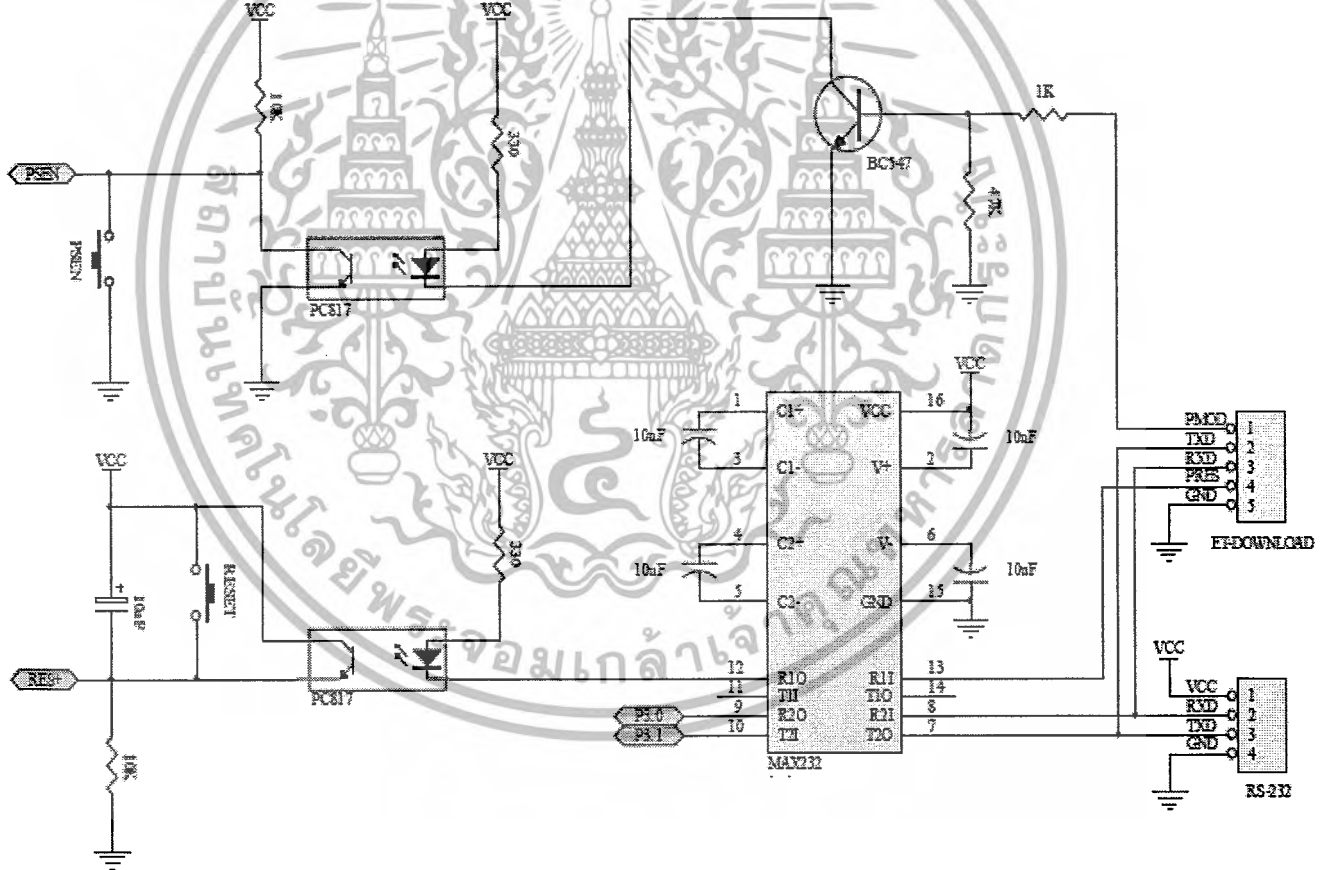
รูปที่ 2.31 พอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต ET-DOWNLOAD เป็นขั้วต่อสำหรับใช้ Download Hex File ให้กับ MCU แบบ
อัตโนมัติโดยใช้งานร่วมกับโปรแกรม FLIP V2.4.4 ของ ATMEL



รูปที่ 2.32 พอร์ต ET-DOWNLOAD



รูปที่ 2.33 วงจรส่วนที่เชื่อมต่อกับ RS-232 และ ET-DOWNLOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Download Hex file ให้กับ MCU ของบอร์ดไมโครคอนโทรลเลอร์ ET-BASE51 AC3

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรม ชื่อ “FLIP” ของ ATMEL ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดได้จาก WWW.ATMEL.COM โดยไม่เสียค่าใช้จ่ายใดๆ แต่สำหรับในกรณีที่ซื้อบอร์ดจาก ETT นั้น โปรแกรมดังกล่าวจะจัดเตรียมไว้ให้ในแผ่น CD ROM อยู่แล้ว โปรแกรม FLIP (Flexible In-system Programmer) เป็นโปรแกรมสำหรับพัฒนาระบบของไมโครคอนโทรลเลอร์ของ ATMEL โดยสามารถใช้สนับสนุนการพัฒนา โปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ในกลุ่มที่ใช้การพัฒนาแบบ ISP ซึ่งรวมถึงเบอร์ AT89C51AC3 ด้วย โดยโปรแกรมจะทำงานภายใต้ระบบปฏิบัติการของ Windows9X/Me/NT/2000 และ Windows XP โดยสนับสนุนการเชื่อมต่อกับระบบฮาร์ดแวร์ที่ใช้การเชื่อมต่อแบบ RS-232 หรือ CAN หรือ USB ซึ่งวิธีการเชื่อมต่อของโปรแกรม FLIP กับระบบฮาร์ดแวร์ของไมโครคอนโทรลเลอร์นั้น จะขึ้นอยู่กับความสามารถของตัวไมโครคอนโทรลเลอร์ที่จะนำมาทำการพัฒนาว่าสามารถใช้การติดต่อสื่อสารด้วยวิธีใดได้บ้าง แต่สำหรับไมโครคอนโทรลเลอร์เบอร์ AT89C51AC3 นั้นจะสามารถใช้การเชื่อมต่อผ่านทางพอร์ตอนุกรม RS-232 เท่านั้นไม่สามารถเชื่อมต่อผ่านระบบการสื่อสารของ CAN หรือ USB ได้ โดยโปรแกรม FLIP จะใช้สำหรับ Download ข้อมูลให้กับหน่วยความจำของไมโครคอนโทรลเลอร์ที่ทำงานใน Monitor Mode เพื่อให้ผู้ใช้สั่งจัดการกับหน่วยความจำภายในตัว CPU ไม่ว่าจะเป็นการส่งลบข้อมูล (Erase) ตรวจสอบข้อมูลในหน่วยความจำ (Blank Check) ส่งโปรแกรมข้อมูลให้กับหน่วยความจำ โปรแกรมของ CPU

2.6 ความรู้เบื้องต้นเกี่ยวกับโปรแกรมวิซวลเบสิก (Visual Basic)

โปรแกรมวิซวลเบสิก (Visual Basic) เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน โปรแกรมวิซวลเบสิกเป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล (Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้น จะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้คลิกปุ่มบนคีย์บอร์ด ผู้ใช้คลิกปุ่มเมาส์ เป็นต้น

เครื่องมือหรือคอนโทรลต่างๆ ที่โปรแกรมวิซวลเบสิกได้เตรียมไว้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่าออบเจกต์) นั้นหมายความว่า ไม่ว่าจะ เป็นเครื่องมือใดๆ ในโปรแกรมวิซวลเบสิกจะเป็นออบเจกต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แกไขคุณสมบัติของออบเจกต์นั้นได้โดยตรง ในทุกๆ ออบเจกต์จะมีคุณสมบัติ (Properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจกต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

3.1 กล่าวนำ

ในการจัดทำเครื่องตรวจจับความเร็ววัดภูมิมีส่วนประกอบหลักอยู่ 2 ส่วนด้วยกันคือ ส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ ซึ่งส่วนของฮาร์ดแวร์นั้นจะประกอบด้วยกล่องเวปแคมและตัวเครื่องซึ่งประกอบด้วยอินฟราเรดเซนเซอร์ Keypad จอ LCD โดยมีวงจรต่างๆ ดังนี้ คือ วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ วงจรเปรียบเทียบแรงดัน วงจรบอร์คสำเร็จรูปของไมโครคอนโทรลเลอร์ วงจรเสียงเตือน และส่วนของซอฟต์แวร์ซึ่งประกอบไปด้วยโปรแกรมควบคุมการทำงานของอุปกรณ์ไมโครคอนโทรลเลอร์และนำอุปกรณ์ต่างๆ มาจัดทำเป็นเครื่องตรวจจับความเร็ววัดภูมิ

เครื่องตรวจจับความเร็ววัดภูมิได้ถูกออกแบบมาโดยการอ้างอิงทฤษฎีทางฟิสิกส์ โดยทำการวางเซนเซอร์ตัวที่ 1 และเซนเซอร์ตัวที่ 2 ห่างกัน 10 เซนติเมตร ทำการหาเวลาความแตกต่างระหว่างเซนเซอร์ทั้ง 2 ตัว โดยให้ไมโครคอนโทรลเลอร์คำนวณหาเวลา โดยคำนวณค่าความเร็วจากการหารระยะห่างระหว่างเซนเซอร์ตัวที่ 1 และเซนเซอร์ตัวที่ 2 นำมาหารด้วยเวลา แล้วทำการแสดงผลผ่านจอ LCD โดยสามารถกำหนดค่าความเร็วเพื่อที่จะเตือน เมื่อมีค่าความเร็วเกินจากที่กำหนด และส่งค่าความเร็วนั้นผ่านพอร์ตอนุกรมไปยังคอมพิวเตอร์ โดยใช้โปรแกรมวิซวลเบสิก ในการรับค่าความเร็วและจับภาพเมื่อมีการส่งค่าความเร็วผ่านทางพอร์ตอนุกรม

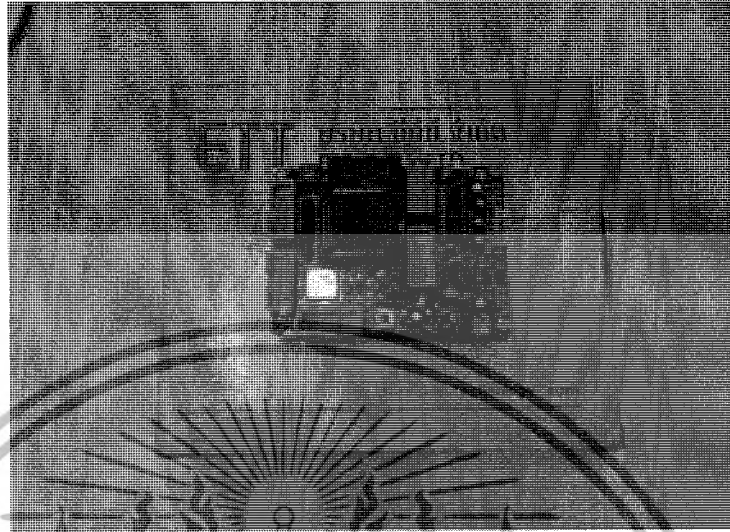
การออกแบบการทำเครื่องตรวจจับความเร็ววัดภูมินั้นจะสำเร็จไม่ได้เลย ถ้าหากขาดการวางแผนและการดำเนินงานที่ดี เพราะเปรียบเสมือนตัวขับเคลื่อนเป็นหัวใจให้การสร้างเครื่องตรวจจับความเร็ววัดภูมิออกมาให้ได้สำเร็จ โดยมีส่วนประกอบต่างๆ ที่สำคัญดังนี้

3.2 การออกแบบในส่วน of หน่วยประมวลผล

ส่วน of หน่วยประมวลผล เราได้นำเอาไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ของ ATMEL เบอร์ AT89C51AC3 ซึ่งมีอยู่ในบอร์คสำเร็จรูปของ ETT (ET-BASE51 AC3) ซึ่งเป็นอุปกรณ์ควบคุมขนาดเล็ก เพื่อใช้ในการประมวลผลค่าของการทำงานที่ได้จากการควบคุมโดยใช้ภาษาซีในการเขียนโปรแกรม เพื่อกำหนดรูปแบบการทำงานและการควบคุมตามความต้องการ และมีพอร์ตในการเชื่อมต่อที่เพียงพอ โดยมีความเร็วในการประมวลผลที่เหมาะสม โดยไมโครคอนโทรลเลอร์ MCS-51 ทำงาน 1 เมกเฮิรตซ์ใช้สัญญาณนาฬิกา 12 ลูก บอร์คสำเร็จรูปของ ETT (ET-BASE51 AC3) ใช้สัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งการทำงานของ MCU ในโหมดความเร็ว 2 เท่า (x2 mode) ดังนั้น 1 เมกเฮิรตซ์ของ MCS-51 จะใช้เวลาเท่ากับ $12/29.4912 \times 2 = 12/58.9824$ ซึ่งมีค่าเท่ากับ 0.2

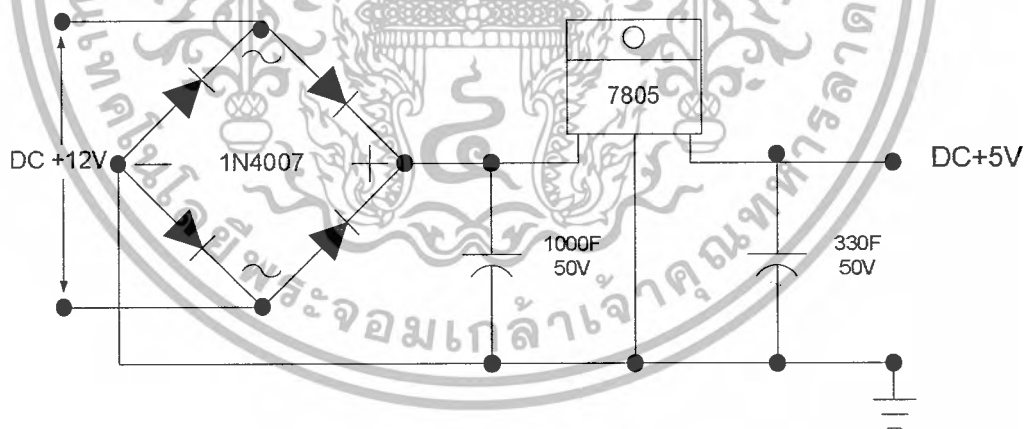
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครวินาที ($0.2\ \mu s$) ดังนั้นเมื่อต้องการสร้างสัญญาณพัลส์ขนาด 1 มิลลิวินาที (1 ms) จะต้องใช้เวลาเท่ากับ $1\ ms / 0.2\ \mu s$ เท่ากับ 5000 แมกซีนไซเคิล



รูปที่ 3.1 บอร์ดสำเร็จรูปของ ETT (ET-BASE51 AC3)

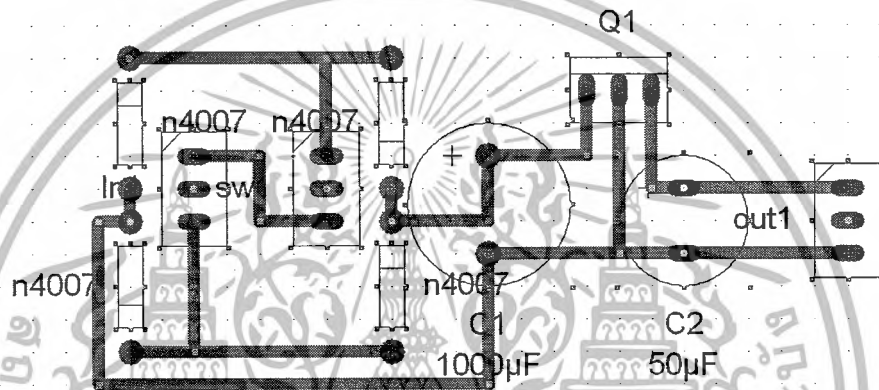
3.3 การออกแบบวงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์



รูปที่ 3.2 วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

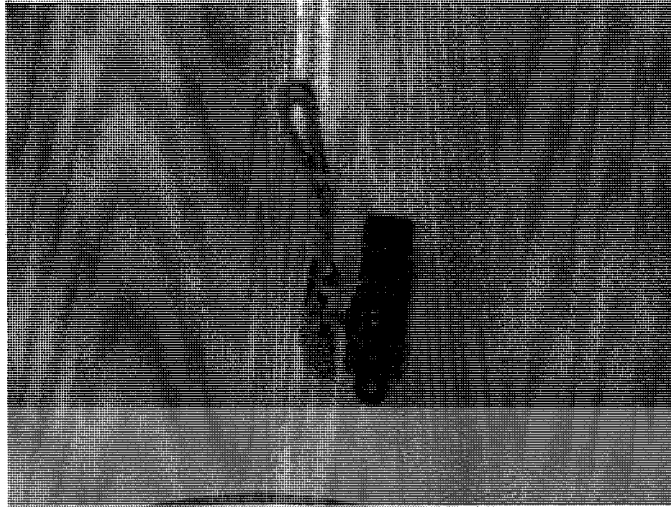
วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ เป็นการแปลงแรงดันกระแสตรง +12V DC จากแบตเตอรี่ให้เป็นแรงดันกระแสตรง +5V DC โดยใช้ไอซี 7805 เพื่อจ่ายแรงดัน +5V DC ให้กับอุปกรณ์ โดยเราทำการจ่ายไฟแรงดันกระแสตรงจากแบตเตอรี่แรงดันกระแสตรง +12V DC เข้ากับวงจรวิทสโตนบริดจ์ เพื่อให้สามารถจ่ายกระแสไฟฟ้าจากแบตเตอรี่สลับขั้วกันได้ แต่ถ้าหากไม่ใช่วงจรวิทสโตนบริดจ์ ก็จะต้องต่อที่ขาบวกเข้ากับขาอินพุตไอซี 7805 และต่อที่ขาลบเข้ากับขากราวด์ของไอซี 7805 โดยมีคาปาซิเตอร์ $100 \mu F$ (เราใช้คาปาซิเตอร์เพื่อให้กระแสมีความคงที่และสม่ำเสมอ) แล้วทำการต่อขาเอาต์พุตของไอซี 7805 โดยมีคาปาซิเตอร์ $50 \mu F$ ไปใช้งานโดยใช้ต่อเป็นแหล่งจ่ายให้กับบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.3 วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ ออกแบบด้วยโปรแกรม PCB Wizard

3.3.1 การต่อวงจรกับอินฟราเรดเซนเซอร์

โดยใช้อินฟราเรดเซนเซอร์ของ SHARP รุ่น GP2Y0A02Y2K ใช้เป็นเซนเซอร์โดยไม่จำเป็นต้องมีวงจรควบคุมภายนอก เพียงแค่จ่ายไฟเลี้ยง (VCC, GND) ก็สามารถนำเอาต์พุตของตัวเซนเซอร์ไปใช้งานได้เลย ซึ่งแรงดัน VCC ที่ใช้จะอยู่ในช่วง 4.5 ถึง 5.5V โดยปกติจะใช้ที่ 5V แต่สามารถทนแรงดันต่ำสุดและสูงสุดได้ -0.3 ถึง +7V เนื่องจากเอาต์พุตของตัวเซนเซอร์จะได้ออกมาเป็นแรงดันอนาล็อก ดังนั้นการจะนำมาต่อกับตัวไมโครคอนโทรลเลอร์นั้นจึงต้องผ่านกระบวนการแปลงสัญญาณเสียก่อนเพื่อให้ได้สัญญาณในรูปแบบดิจิทัลที่ MCU สามารถประมวลผลได้ โดยใช้อุปกรณ์แปลงสัญญาณ Analog to Digital Converter หรืออาจนำเอาเอาต์พุตของตัวเซนเซอร์มาต่อเข้ากับวงจรเปรียบเทียบแรงดัน (Comparator) แล้วให้มีสถานะเอาต์พุตออกมาเป็น 0 (0V) หรือ 1 (+5V) ก็ได้เช่นกัน โดยนำเซนเซอร์ตัวที่ 1 ต่อกับวงจรเปรียบเทียบแรงดันวงจรที่ 1 และให้เซนเซอร์ตัวที่ 2 ต่อกับวงจรเปรียบเทียบแรงดันวงจรที่ 2



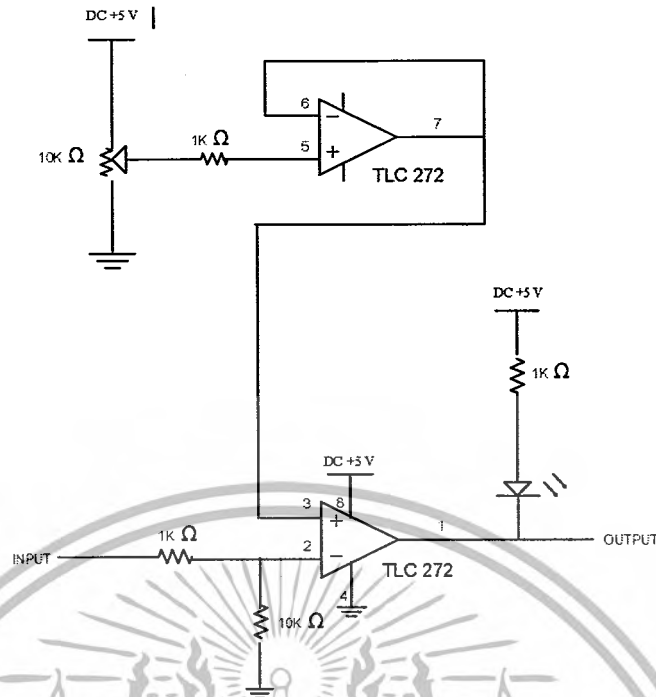
รูปที่ 3.4 อินฟราเรดเซนเซอร์

3.4 การออกแบบวงจรเปรียบเทียบแรงดัน

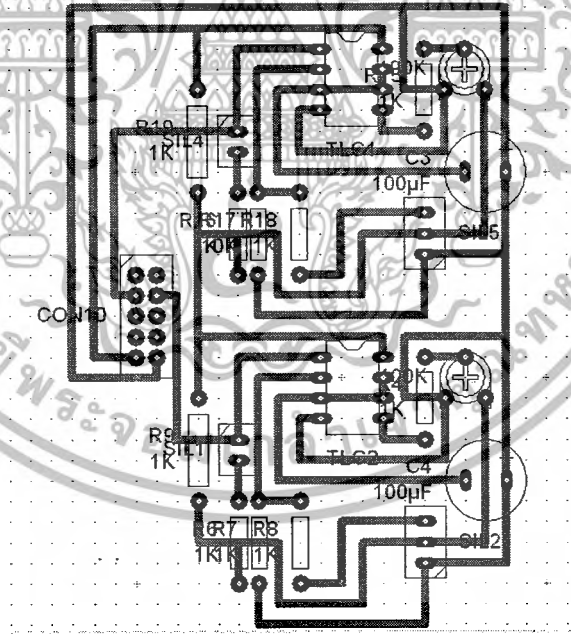
วงจรเปรียบเทียบแรงดัน โดยให้สภาวะทางเอาต์พุตเป็นแบบ Logic TTL โดยสามารถกำหนดค่าแรงดันอ้างอิงได้เองตามต้องการภายในบอร์ด โดยการปรับตั้งค่าแรงดันจากตัวต้านทานปรับค่าได้ 10 k Ω ต่อเข้ากับขาที่ 5 ของไอซี TLC 272 โดยใช้ไอซีออปแอมป์ทำหน้าที่เป็นบัฟเฟอร์ โดยมีอัตราขยายของวงจรเท่ากับ 1 เพื่อใช้เป็นจุดอ้างอิงในการเปรียบเทียบแรงดันของ ไอซี ออปแอมป์ทำหน้าที่เป็นบัฟเฟอร์ต่อเข้ากับขาที่ 3 ของ ไอซี TLC 272 ซึ่งใช้เปรียบเทียบแรงดันกับขาที่ 2 ของ ไอซี TLC 272 เพื่อทำการเปรียบเทียบแรงดัน โดยมีอัตราขยายเป็นอนันต์

โดยในการใช้งานนั้นจะป้อนแรงดันเข้ามายังจุดอินพุต เพื่อทำการเปรียบเทียบค่าระดับแรงดันกับค่าแรงดันอ้างอิงที่กำหนดไว้กับแรงดันที่ได้จากตัวต้านทานปรับค่าได้ ซึ่งเมื่อค่าแรงดัน อินพุตมีค่าต่ำกว่าแรงดันอ้างอิงที่กำหนดไว้ภายในบอร์ด จะได้สถานะของสัญญาณเอาต์พุตเป็นลอจิก 1 (LED สถานะจะดับ) แต่เมื่อใดก็ตามที่ค่าแรงดันของอินพุตมีค่ามากกว่าค่าแรงดันอ้างอิงที่กำหนดไว้จะทำให้ได้สถานะของเอาต์พุตเป็นลอจิก 0 ซึ่ง LED แสดงสถานะจะติดสว่างให้เห็นโดยนำวงจรเปรียบเทียบแรงดันตัววงจรที่ 1 ต่อกับบอร์ดไมโครคอนโทรลเลอร์พอร์ต P3.2 และให้วงจรเปรียบเทียบแรงดันวงจรที่ 2 ต่อกับในบอร์ดไมโครคอนโทรลเลอร์พอร์ต P3.3 จะเห็นได้ว่าใช้วงจรเปรียบเทียบแรงดันจำนวน 2 วงจร เพราะต้องต่อกับเซนเซอร์ทั้งสองตัวเพื่อแปลงสัญญาณอนาล็อกที่ได้จากเซนเซอร์ให้เป็นสัญญาณดิจิทัลให้กับไมโครคอนโทรลเลอร์ ทั้งนี้เราทำการปรับตัวต้านทานปรับค่าได้ของวงจรเปรียบเทียบแรงดันทั้งสอง เพื่อให้เซนเซอร์ตรวจจับวัตถุให้มีระยะห่างเท่าๆ กัน โดยทางผู้จัดทำพบว่าการปรับตัวต้านทานของเครื่องตรวจจับความเร็ว สามารถตรวจจับวัตถุที่มีระยะห่าง 10 ถึง 50 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



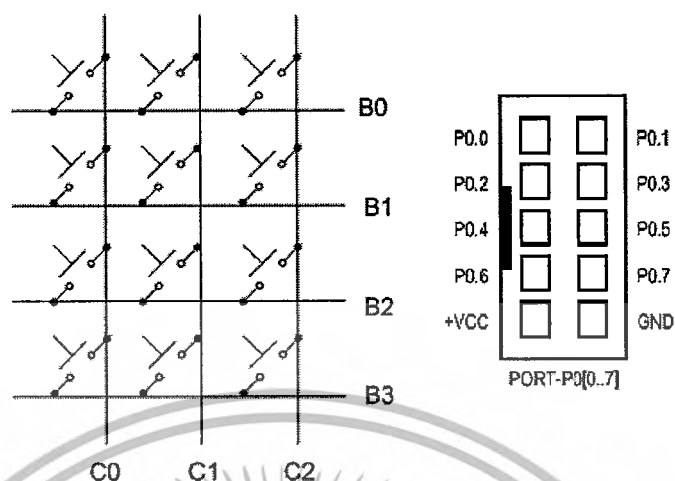
รูปที่ 3.5 วงจรเปรียบเทียบแรงดันที่ใช้กับอินฟราเรดเซนเซอร์



รูปที่ 3.6 วงจรเปรียบเทียบแรงดันออกแบบด้วยโปรแกรม PCB Wizard ที่ใช้กับอินฟราเรดเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การต่อวงจร Keypad

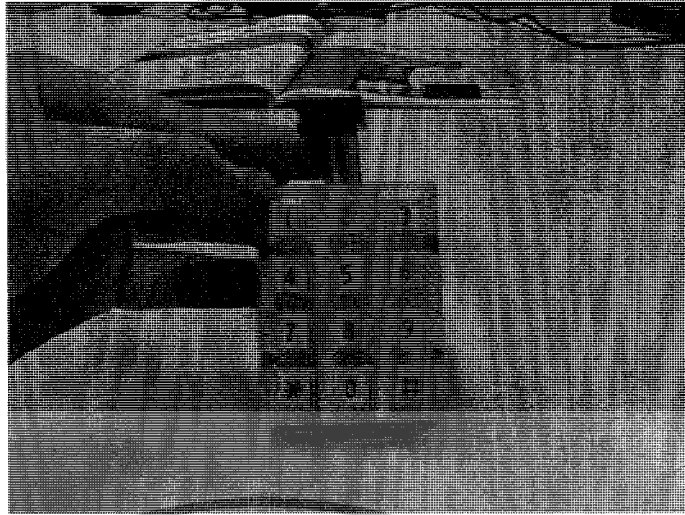


รูปที่ 3.7 วงจร Keypad และพอร์ต P0

โดยเป็นอุปกรณ์อินพุตชนิดหนึ่งใช้ในการรับค่าในการจำกัดค่าความเร็วสูงสุด และทำการตั้งค่าให้กับไมโครคอนโทรลเลอร์ ใช้ในการเปรียบเทียบกับค่าและทำตามเงื่อนไขต่อไป โดยทำการติดต่อกันดังนี้

ชื่อพอร์ต	การต่อเข้ากับ	พอร์ตของคีย์แพด
P0.0	=	B 3
P0.1	=	B 2
P0.2	=	B 1
P0.3	=	B 0
P0.4	=	C 2
P0.5	=	C 1
P0.6	=	C 0

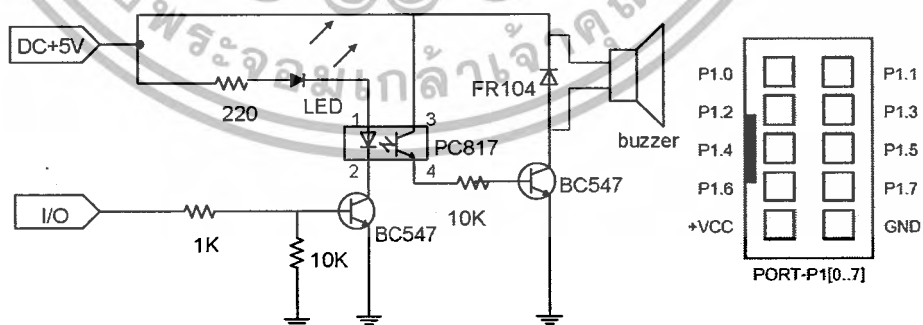
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Keypad ที่ต่อเรียบร้อยแล้ว

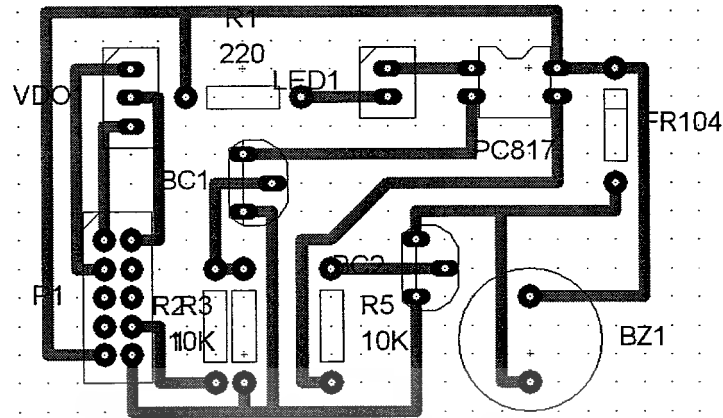
3.4.2 การต่อวงจร Buzzer

วงจร Buzzer ใช้สำหรับควบคุม Buzzer โดยจะมีไอซี Photocoupler เบอร์ PC817 ในการแยกวงจรทางไฟฟ้าออกจากกัน โดยมีหลักการทำงานดังนี้ ตัวต้านทาน 1 k Ω และ 10 k Ω ที่ขา B ของทรานซิสเตอร์ BC547 เป็นวงจรแบ่งแรงดันเพื่อใช้เป็นแรงดันไบอัสทางอินพุตของทรานซิสเตอร์ BC547 ซึ่งเมื่อทรานซิสเตอร์ทำงานจะทำให้ LED สว่างและไอซี Photocoupler เบอร์ PC817 จะนำกระแสทางเอาต์พุตส่งผลให้ทรานซิสเตอร์ BC547 ตัวที่ 2 นำกระแสเพื่อขับให้ Buzzer ทำงาน โดยมีอัตราขยายของ BC547 ตัวที่ 1 และตัวที่ 2 เป็น 10 เท่า การใช้งาน เมื่อให้ลอจิก 1 ทางขาอินพุต/เอาต์พุต จะทำให้ LED ติด และ Buzzer เกิดเสียงดังขึ้นและทำการต่อขาอินพุต/เอาต์พุต เข้ากับพอร์ต P1.7 ในบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.9 วงจร Buzzer และพอร์ต P1

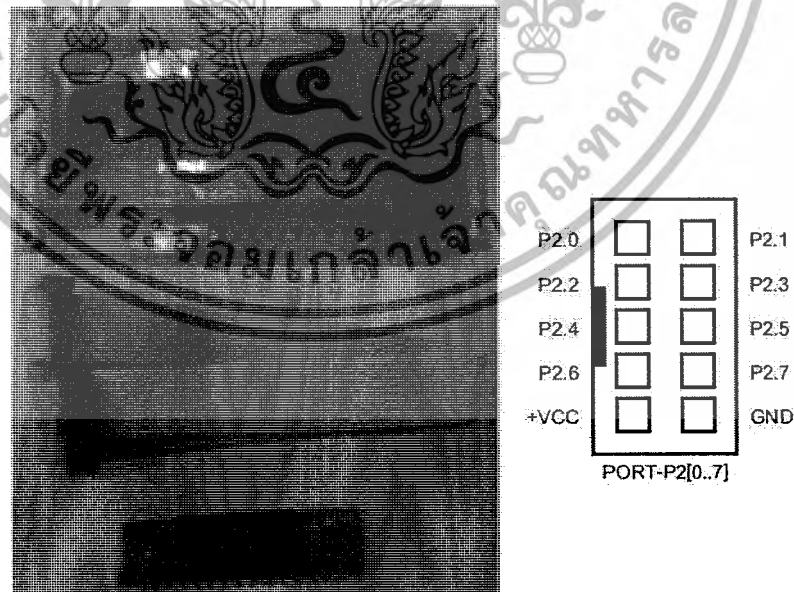
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจร Buzzer ออกแบบด้วยโปรแกรม PCB wizard

3.4.3 การต่อจอ LCD

ในการติดต่อระหว่างไมโครคอนโทรลเลอร์กับโมดูล LCD เราจะไม่ใช่พอร์ต LCD โดยตรงแต่เราแปลงจาก พอร์ต 16 pin เป็น Con 10 pin โดยใช้สายแพรเพื่อที่จะให้ใช้งานได้สะดวก โดยทำการต่อกับ พอร์ต P2 ในบอร์ดไมโครคอนโทรลเลอร์ ดังรูปที่ 3.11 โดย LCD ที่ใช้จะเป็น LCD แบบ 16x2 คือมีการแสดงตัวอักษรหรือตัวเลขจำนวน 2 แถว แถวละ 16 ตัวอักษร แสดงตัวอักษรและตัวเลขเป็นสีเขียว ในการแสดงค่าบนหน้าจอ LCD

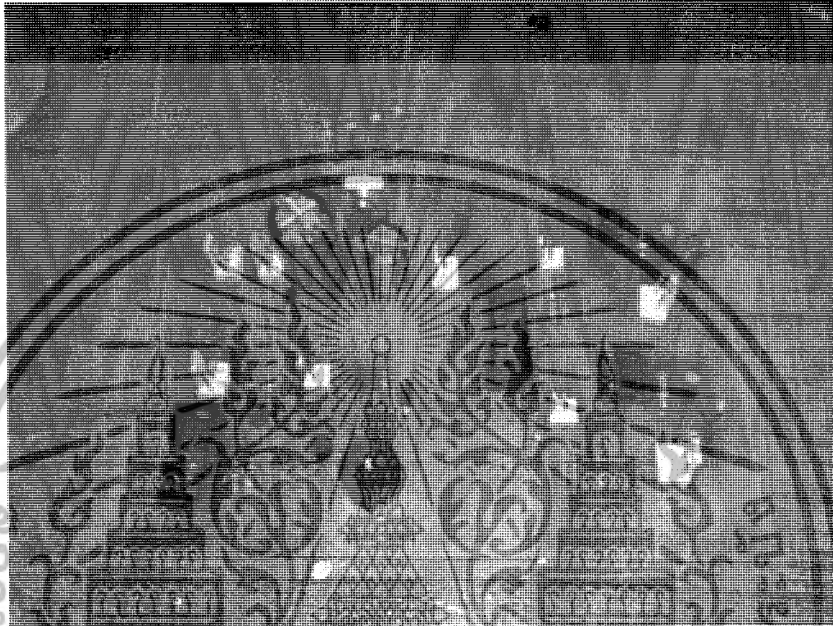


รูปที่ 3.11 จอ LCD และอุปกรณ์เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การออกแบบแผ่นวงจร PCB

การออกแบบแผ่นวงจรด้วย PCB Wizard ได้ทำการออกแบบวงจรต่างๆ ดังนี้ วงจรจ่ายแรงดัน DC +5V วงจรเปรียบเทียบแรงดันสำหรับเซนเซอร์ทั้ง 2 ตัวและวงจรเสียงเตือน Buzzer เพื่อสะดวกในการใช้งานก็ให้วงจรทั้ง 3 วงจรรวมอยู่ในบอร์ดแผ่นเดียวกัน เมื่อออกแบบเสร็จเรียบร้อยแล้ว ก็ทำการกัดแผ่นปริ้นซ์ และนำอุปกรณ์ลงแผ่น PCB เป็นอันเสร็จสมบูรณ์ดังรูปที่ 3.12



รูปที่ 3.12 วงจรที่สมบูรณ์

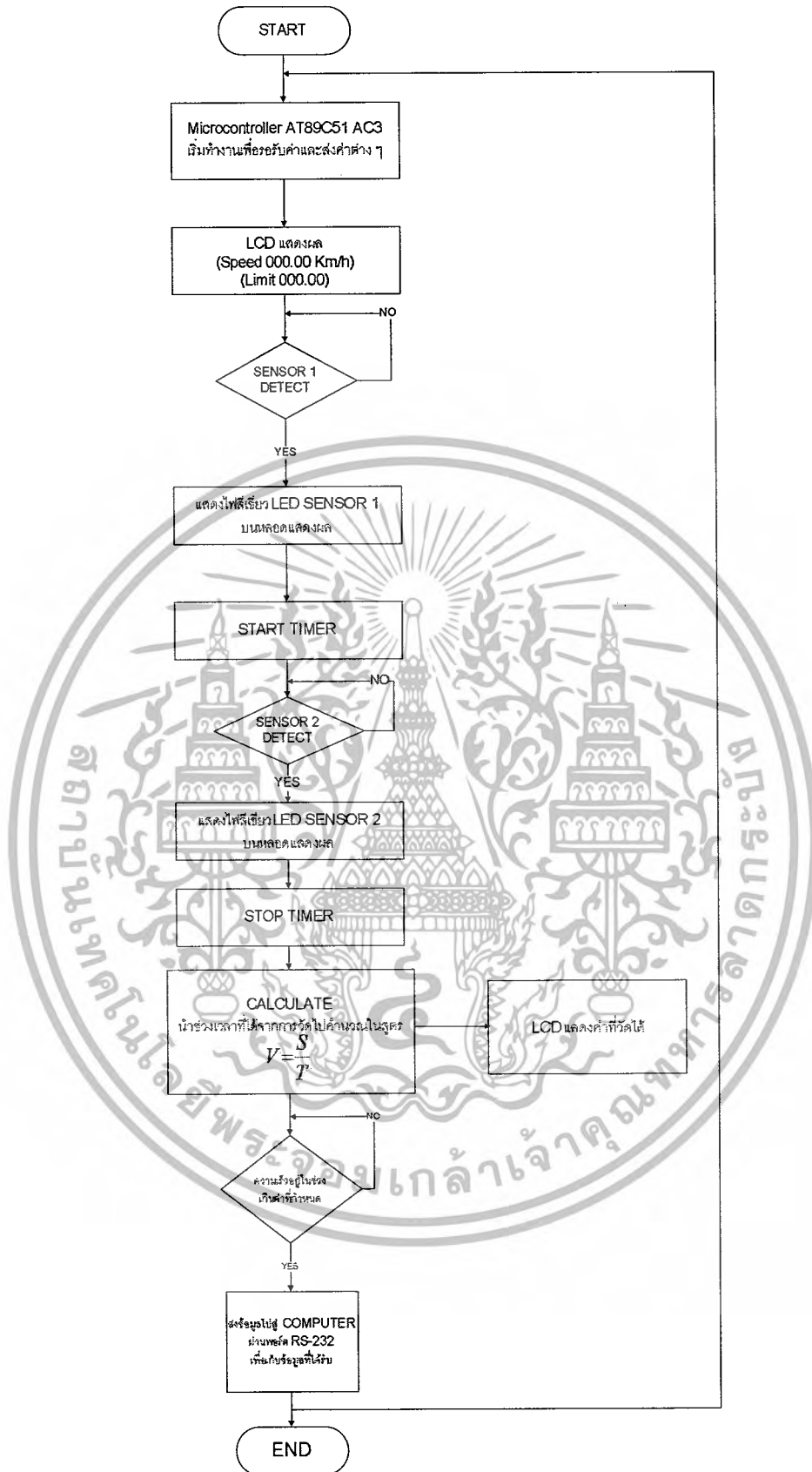
3.4.5 การเขียนโปรแกรมใช้กับไมโครคอนโทรลเลอร์

ทำการเขียนโปรแกรมเพื่อสั่งให้ไมโครคอนโทรลเลอร์ไปควบคุมส่วนต่างๆของอุปกรณ์ได้ นั้นต้องมีความรู้ทางด้านฮาร์ดแวร์ประกอบในการเขียนโปรแกรม และเมื่อเขียนเสร็จแล้วต้องมีการตรวจสอบโปรแกรมว่ามีความผิดพลาด ในการเขียน โปรแกรม และอาจต้องนำเงื่อนไขบางอย่างเพิ่มเติมลงในโปรแกรม เพื่อให้เครื่องตรวจจับความเร็วทำงาน ได้อย่างสมบูรณ์

3.4.6 การออกแบบซอฟต์แวร์

1. การเขียนโปรแกรมภาษา C ให้ไมโครคอนโทรลเลอร์ MCS-51 จากการกำหนด ขาสัญญาณต่างๆ ในการออกแบบวงจร เราจะต้องนำพอร์ต้นั้นๆ นำมาใช้ในการเขียนโปรแกรมตามเงื่อนไขในการตรวจจับความเร็ววัตถุ ซึ่งสามารถอธิบายเป็น Flowchart ได้ดังต่อไปนี้ (โปรแกรมอยู่ในภาคผนวก ก.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Flowchart แสดงการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.13 เป็นการอธิบายในส่วนของการทำงานของโปรแกรมที่ควบคุมการทำงานของไมโครคอนโทรลเลอร์มีดังนี้

1.1 ไมโครคอนโทรลเลอร์ จะเริ่มการทำงานโดยการติดต่อกับพอร์ตต่างๆ เพื่อรอรับค่าและให้ไมโครคอนโทรลเลอร์พร้อมที่จะทำงาน โดยการกำหนดหน้าที่ของพอร์ตแต่ละพอร์ตไว้ให้ทำหน้าที่นั้นๆดังโปรแกรมที่จะแสดงดังต่อไปนี้

```
#include <reg52.h> // ประกาศเรียกใช้ไลบรารีฟังก์ชันของ MCS-51
#include "stdio.h" // ประกาศเรียกใช้ไลบรารีฟังก์ชันสแตนด์ออลไอโอ
/* AT89C51AC3 SFR */ // ไมโครคอนโทรลเลอร์ MCS-51 ที่ใช้
sfr CKCON = 0x8F; // กำหนดค่า Clock Control
char lcdbuf[16+1]; // กำหนดค่า บัฟเฟอร์ LCD
unsigned char num[]={\n','1','2','3','4','5','6','7','8','9','*','0','#'}; // กำหนดตัวแปร num
char numbuf[5]; // กำหนดค่า numbuf เท่ากับ 5
/*Key Switch*/
#define KeySwitch P0 // กำหนด P0 เป็น Key Switch
sbit sw1 = P0^4; // กำหนด SW1 ที่ P0.4
sbit sw2 = P0^5; // กำหนด SW2 ที่ P0.5
sbit sw3 = P0^6; // กำหนด SW3 ที่ P0.6
/*Sensor Input*/
sbit Sensor_R = P3^2; // กำหนด Sensor_R ที่ P3.2
sbit Sensor_L = P3^3; // กำหนด Sensor_L ที่ P3.3
sbit beep = P1^7; // กำหนด beep ที่ P1.7
sbit LED = P1^0; // กำหนด LED ที่ P1.0
sbit SHUTTER = P1^1; // กำหนด SHUTTER ที่ P1.1
sbit LED_AMR = P1^2; // กำหนด LED_AMR ที่ P1.2
float time_100u_sec=0; // กำหนด time_100u_sec เท่ากับ 0
float Km_hour=0; // กำหนด Km_hour เท่ากับ 0
float numSET=0; // กำหนด numSET เท่ากับ 0
char setlinelcd=0x46; // กำหนด setlinelcd เท่ากับ 0
char menu=0; // กำหนด menu เท่ากับ 0
char on=0; // กำหนด on เท่ากับ 0
char i=0; // กำหนด i เท่ากับ 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char number_time=0;           // กำหนด number_time เท่ากับ 0
sbit BK = P2^0;                        // ให้ตัวแปร BK แทนบิต P2.0
sbit RS = P2^1;                        // กำหนด RS LCD (0 = คำสั่ง,1= ข้อมูล)
sbit RW = P2^2;                        // กำหนด RW LCD (0 = การเขียน,1= การอ่าน)
sbit E = P2^3;                         // เปิดใช้งาน LCD(กำหนดทำงานที่ = "1")
//char lcdbuf[16+1];                  // กำหนดค่าบัฟเฟอร์แสดงผล LCD
void serial_init (void) {              // เริ่มต้น โปรแกรม Serial Initial
    TMOD &= 0x0F;                       // รีเซ็ตค่าไทม์เมอร์อันเก่า โหมดตั้งค่า
    TMOD |= 0x20;                       // ใส่ค่าไทม์เมอร์ 1 เป็น 8 บิต อัตโนมติ
    SCON = 0x50;                        // พอร์ตอนุกรม โหมด 1 (N,8,1)
    ES = 0;                             // หยุดการเชื่อมต่ออินเตอร์รัพต์อนุกรม
    ET1 = 0;                            // หยุดการเชื่อมต่ออินเตอร์รัพต์ไทม์เมอร์1
    PCON &= 0x7F;                       // หยุดการเชื่อมต่อคัมเบิลบอร์ดเรต
    TH1 = 0xF0;                         // ตั้งค่าไทม์เมอร์ 1 บอร์ดเรต 9600 บิตต่อวินาที / 58.9824 เมกกะเฮิร์ตซ์
    TL1 = 0xF0;                         // ตั้งค่าไทม์เมอร์ 1 บอร์ดเรต 9600 บิตต่อวินาที / 58.9824 เมกกะเฮิร์ตซ์
    TR1 = 1;                            // เริ่มต้นการทำงานไทม์เมอร์ 1
    TI = 1;                             // เซตค่า TI เท่ากับ 1 เพื่อที่จะทำงานส่งข้อมูล
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ทำการเคลียร์หน้าจอ LCD เพื่อให้ LCD แสดงผลเป็น Speed 000.00 km/h Limit 000.00 เพื่อรอรับค่าความเร็วจากการประมวลผลไมโครคอนโทรลเลอร์มาแสดงผลต่อไป

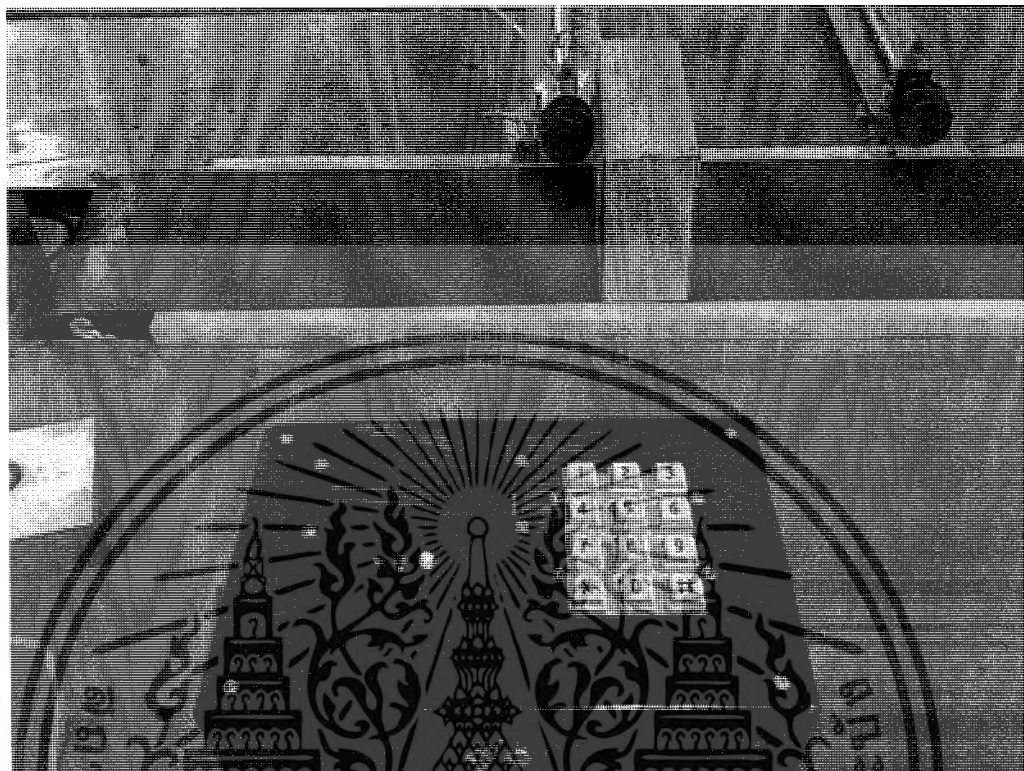


รูปที่ 3.14 หน้าจอ LCD ตอนเปิดเครื่องโดยยังไม่มีวัตถุผ่าน

```
void main(void){ // รายละเอียดของฟังก์ชันเมน
CKCON = 0x01; // เชื่อมต่อ โหมด 2 เท่า (ความถี่สัญญาณนาฬิกา = 58.9824 เมกกะเฮิร์ตซ์
Init_Init(); // ทำการเชื่อมต่อ
serial_init(); // ทำการเชื่อมต่อพอร์ตอนุกรม
init_lcd(); // ทำการเชื่อมต่อ LCD
time_1mS_sec=0; // เวลา 1 มิลลิวินาที เท่ากับ 0
clrscr_lcd(); // เคลียร์หน้าจอ LCD
beepk1(); // เรียกใช้ฟังก์ชัน beepk1
Delay // เรียกใช้ Delay
uS(10000); // 10,000 ไมโครวินาที
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เมื่อมีวัตถุผ่านเซนเซอร์ตัวที่ 1 ไฟแสดงสถานะของเซนเซอร์ตัวที่ 1 ก็จะติด ไทม์เมอร์ในไมโครคอนโทรลเลอร์จะเริ่มการทำงาน



รูปที่ 3.15 เครื่องตรวจความเร็วขณะที่วัตถุผ่านเซนเซอร์ตัวที่ 1

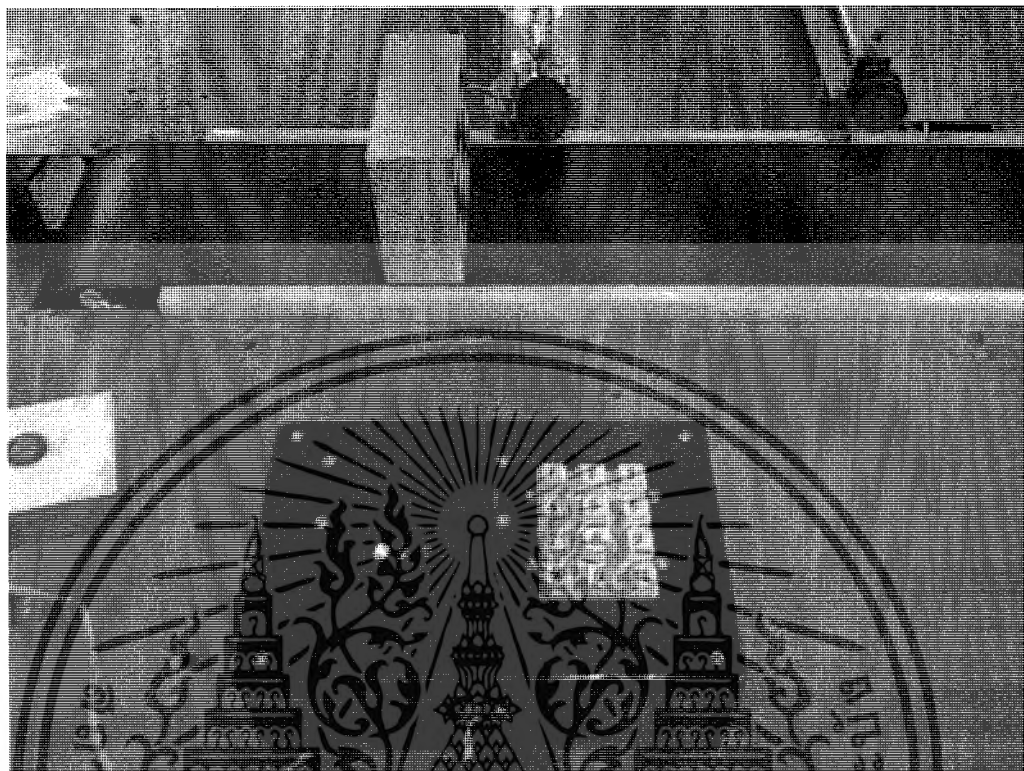
```

}
void StartTimer(void) { // เริ่มต้นทำงานไทม์เมอร์
time_1mS_sec=0; // เซตเวลา เวลา 1 มิลลิวินาที เท่ากับ 0
on=1; // เซต on เท่ากับ 1
LED=0; // สั่งให้ LED ดับ
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ไทม์เมอร์จะหยุดทำงานก็ต่อเมื่อมีวัตถุผ่านเซนเซอร์ตัวที่ 2 ไฟแสดงสถานะของเซนเซอร์ตัวที่ 2 ก็จะติด เป็นการหยุดนับค่าไทม์เมอร์



รูปที่ 3.16 เครื่องตรวจความเร็วขณะที่วัตถุผ่านเซนเซอร์ตัวที่ 2

```

}
void StopTimer(void) { // หยุดทำงาน ไทม์เมอร์
on=0; // เซต on เท่ากับ 0
if(time_1mS_sec<=200) // กำหนดเงื่อนไข เวลา 1 มิลลิวินาที น้อยกว่าหรือเท่ากับ 200
Km_hour=((0.10/time_1mS_sec)*3600); // กำหนดค่ากิโลเมตรต่อชั่วโมง
else Km_hour=0; // กรณีค่ากิโลเมตรต่อชั่วโมง เท่ากับ 0
LED=1; // สั่ง LED ให้ทำงาน
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ไมโครคอนโทรเลอร์ MCS-51 ทำงาน 1 แมกซ์ซินไซเคิล ใช้สัญญาณนาฬิกา 12 ลูก ใน AT89C51AC3 ใช้สัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งการทำงานของ MCU ในโหมดความเร็ว 2 เท่า (x 2 mode) ดังนั้น 1 แมกซ์ซินไซเคิลของ MCS-51 จะใช้เวลาเท่ากับ $12/29.4912 \times 2 = 12/58.9824$ ซึ่งมีค่าเท่ากับ 0.2 ไมโครวินาที (0.2 μs) ดังนั้นเมื่อต้องการสร้างสัญญาณพัลส์ขนาด 1 มิลลิวินาที (1 ms) จะต้องใช้เวลาเท่ากับ $1 \text{ ms}/0.2 \mu s$ เท่ากับ 5000 แมกซ์ซินไซเคิล ค่าที่ไทม์เมอร์สามารถนับสูงสุดก่อนเกิดการโอเวอร์โฟลว์ คือ FFFFH เมื่อเลือกใช้ไทม์เมอร์ขนาด 16 บิต ซึ่ง FFFFH ซึ่งมีค่าเท่ากับ 65535 - 5000 = 60535 ซึ่งมีค่าเท่ากับ EC77H เป็นค่าเริ่มต้น จะกำหนดให้กับไทม์เมอร์ โดยแบ่งกำหนดให้ TH (ไบต์สูง) = ECH และ TL (ไบต์ต่ำ) = 77H แล้วเรานำจำนวนครั้งที่เกิดจากการโอเวอร์โฟลว์ของไทม์เมอร์ที่ได้มา คำนวณความเร็วจากสูตร $v = \frac{S}{T}$ โดยที่กำหนดระยะห่างระหว่างเซนเซอร์ตัวที่ 1 และเซนเซอร์ตัวที่ 2 คือ 0.1 เมตร และเรานำค่าระยะทางเวลามาทำการหารด้วยเวลา ซึ่งจะออกมาเป็นความเร็วโดยเวลา

```

}
void Timer_0_service(void) interrupt 1 using 1 { //รายละเอียดฟังก์ชันไทม์เมอร์ 0 อินเทอร์รัพต์ 1
//service routine Timer interrupt T0 (#1) register bank 1 (using 1) //เก็บค่าไทม์เมอร์ อินเทอร์รัพต์ T0
//Delay time 1_ms //หน่วยเวลา 1 มิลลิวินาที
    TH0 = 0xEC; //เพิ่มค่า TH0 = 5000H = 1 มิลลิวินาที
    TL0 = 0x77; //เพิ่มค่า TL0 = 5000H = 1 มิลลิวินาที
    if(on==1) //กำหนดเงื่อนไข on เท่ากับ 1
        time_1mS_sec+=0.0015; //เวลา 1 มิลลิวินาที += 0.0015
}

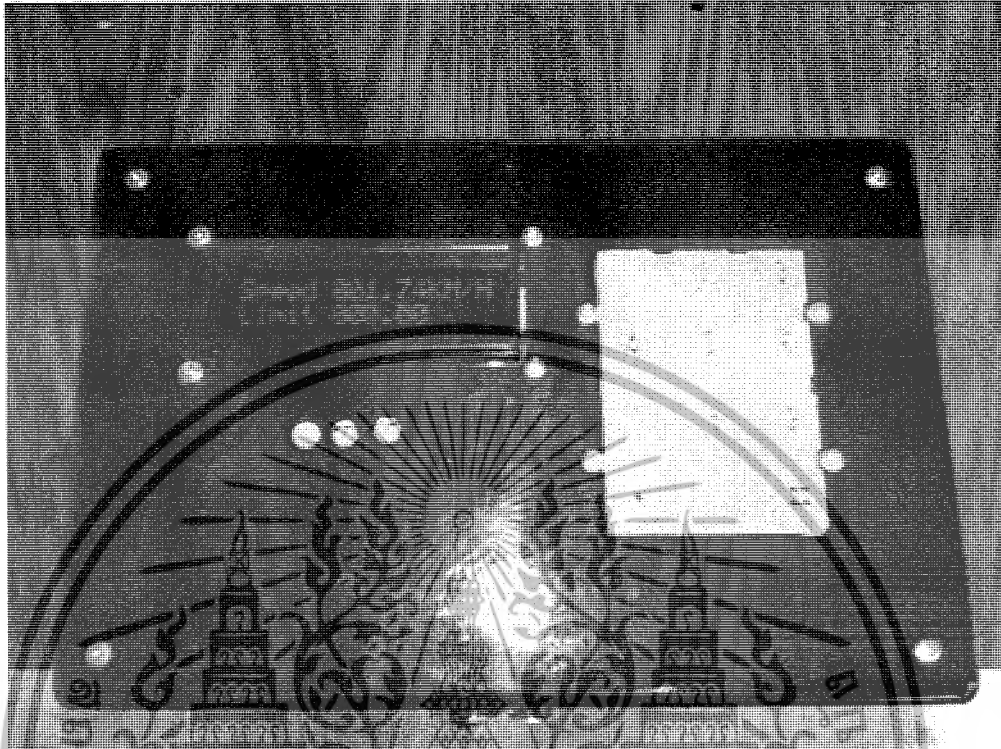
if(time_1mS_sec<=200) //กำหนดเงื่อนไขเวลา 1 มิลลิวินาที น้อยกว่าหรือเท่ากับ 200
    Km_hour=((0.10/time_1mS_sec)*3600); //กำหนดค่ากิโลเมตรต่อชั่วโมง
else Km_hour=0; //กรณีค่ากิโลเมตรต่อชั่วโมง เท่ากับ 0
LED=1; //สั่ง LED ให้ทำงาน
}

void SeviceIninterrupt_1(void) interrupt 0 using 2 { //รายละเอียดฟังก์ชันของเซอร์วิตอินเทอร์รัพต์ 1
if(menu==0&& i==0){ //เงื่อนไขเมนูเท่ากับ 0 และอินเดกซ์เท่ากับ 0
    StartTimer(); //เริ่มต้นการทำงานไทม์เมอร์
    number_time=0; //กำหนดค่า number_time เท่ากับ 0
    i=1; //ค่าอินเดกซ์เท่ากับ 1
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 แสดงผลค่าความเร็วที่คำนวณได้จากไมโครคอนโทรลเลอร์ทาง LCD มีหน่วยเป็น กิโลเมตรต่อชั่วโมง



รูปที่ 3.17 หน้าจอ LCD แสดงผลที่ได้จากการตรวจจับความเร็ววัตถุ

```
void printlcd(char *str){ // รายละเอียดฟังก์ชัน printlcd
while(*str){ // รับค่า ASCII และเขียนลง LCD จนกระทั่งหมดค่า
write_data(*str); // เขียนค่า ASCII ลง LCD
str++; // ค่า ASCII ตัวถัดไป
}
return; // กลับไปที่ฟังก์ชัน
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 และถ้ามีการจำกัดค่าความเร็วสูงสุดไว้ก็จะมีพิจารณาว่า ความเร็วที่ตรวจจับได้นั้นเกินค่าที่กำหนดหรือไม่ ถ้าเกินก็จะมีการส่งเสียงเตือน

```
if(Km_hour/1000>numSET&&numSET>0&&number_time<30&&i==0){ // กำหนดเงื่อนไขค่า
                                                    ความเร็ว

number_time++; // ทำการบวกค่าเพิ่มขึ้นทีละ 1 ค่า
beeper(); // ฟังก์ชันเสียง beeper
LED_AMR=0; // LED เมื่อเท่ากับ 0
if(number_time==1){SHUTTER=0; // เงื่อนไขค่า number_time เท่ากับ 1 ชัตเตอร์ จะไม่ทำงาน
printf("\rSPEED %f KM/H Limit %f ",Km_hour/1000,numSET); // แสดงค่าที่ได้บนหน้าจอ
}
} else LED_AMR=1; // กรณีที่ LED เมื่อเท่ากับ 1
```

1.8 และตั้งค่าความเร็วไปยังพอร์ตอนุกรม เพื่อติดต่อกับคอมพิวเตอร์ โปรแกรมก็จะทำการรวนเพื่อไปรอรับค่าจากเซนเซอร์ตัวที่ 1 ใหม่ และทำการวนอย่างนี้เรื่อยไปจนกว่าจะมีการปิดเครื่อง

```
printf("\rSPEED %f KM-H Limit %f ",Km_hour/1000,numSET); // ทำการส่งค่าความเร็ว
                                                    ออกไปยังพอร์ตอนุกรม
```

1.9 เมื่อเรากำหนดเงื่อนไขในการเขียนโปรแกรมดัง Flowchart ที่กล่าวมาแล้ว เราก็ทำการเขียนโปรแกรม โดยใช้ Keil uVision 3 ซึ่งเป็นซอฟต์แวร์สำเร็จรูปที่ทางบริษัทผู้ผลิตบอร์ดให้มา เมื่อเขียนเสร็จต้องทำการ Build target ถ้าหากเขียนถูกต้องจะไม่มี ERROR (ต้องดูการเงื่อนไขให้รอบคอบ)

```

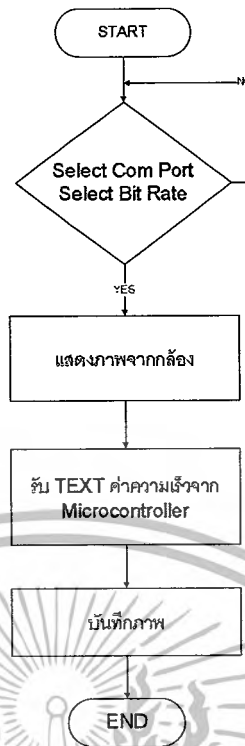
016 E2 |= 0x0F; // Clear old LCD Data (Bit[7..4])
017 E2 |= 0x30; // DS:DS4 = 1:1
018 enable_lcd(); // Enable Pulse
019 for (i=0;i<2500;i++); // Delay 1.2ms
020 E2 |= 0x0F; // Clear old LCD Data (Bit[7..4])
021 E2 |= 0x30; // DS:DS4 = 1:1
022 enable_lcd(); // Enable Pulse
023 for (i=0;i<100;i++); // Delay 100us
024 E2 |= 0x0F; // Clear old LCD Data (Bit[7..4])
025 E2 |= 0x95; // DS:DS4 = 1:1
026 enable_lcd(); // Enable Pulse
027 while(busy_lcd()); // Wait LCD Execute Complete
028 E2 |= 0x0F; // Clear old LCD Data (Bit[7..4])
029 E2 |= 0x26; // DS:DS4 = 1:0
030 enable_lcd(); // Enable Pulse
031 while(busy_lcd()); // Wait LCD Execute Complete
032
033 write_ins(0x28); // Function Set (DL=0,1-Bit,Mode 2 Idle,F=0 SK7)
034 write_ins(0x0C); // Display on/off Control (Entry Display, Cursor off, Cursor not Blink)
035 write_ins(0x06); // Entry Mode Set (I/D=1 Increment, S=0 Cursor Shift)
036 write_ins(0x01); // Clear Display (Clear Display, Set DD RAM Address=0)
037
038 void clrscr_lcd(void) {
039     write_ins(0x01);

```

Build target 'AT89C51AC3'
linking...
Program Size: data=118.1 xdata=0 const=0 code=4796
creating hex file from "Project"...
"Project" - 0 Error(s), 0 Warning(s).

รูปที่ 3.18 การเขียนโปรแกรมภาษา C ให้ไมโครคอนโทรลเลอร์ MCS-51

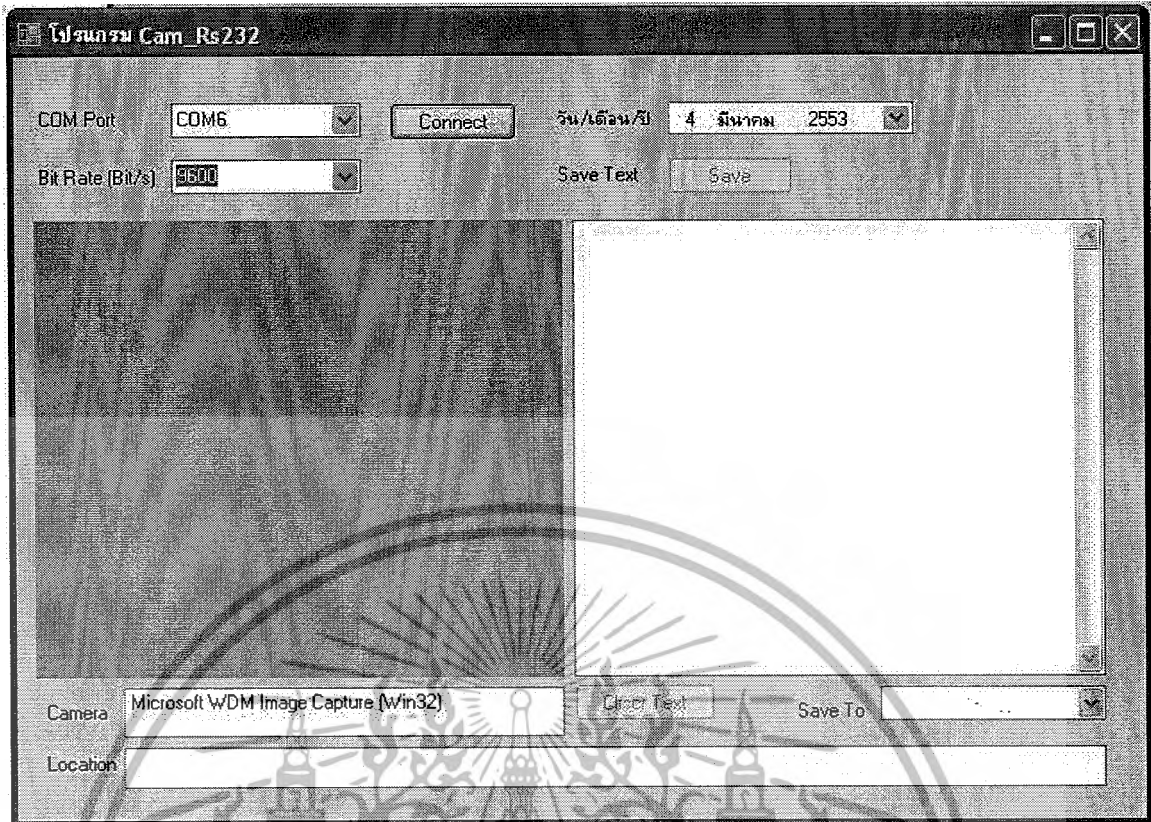
2. การออกแบบโปรแกรมวิซวลเบสิก ที่ทำการออกแบบมา เพื่อใช้การจับภาพจากกล้องเว็บแคมโดยมีการออกแบบ ด้วยการสร้างหน้าต่างรับค่าความเร็วจากพอร์ตอนุกรมที่ติดต่อกับเครื่องตรวจจับความเร็ว และอีกหน้าต่างแสดงผลจากภาพจากกล้องเว็บแคมโดยมีเงื่อนไขว่าต้องมีการส่งข้อมูลความเร็วจากไมโครคอนโทรลเลอร์มาสู่คอมพิวเตอร์ โปรแกรมวิซวลเบสิกก็จะทำการบันทึกภาพจากกล้องเว็บแคมโดยบันทึกค่าความเร็วเป็นตัวอักษรสีแดงบริเวณมุมซ้ายด้านบน ตามที่ระบุไว้ซึ่งสามารถอธิบายเป็น Flowchart ได้ดังต่อไปนี้ (โปรแกรมอยู่ในภาคผนวก ข.)



รูปที่ 3.19 Flowchart แสดงการทำงานของโปรแกรมวิซวลเบสิก

จากรูปที่ 3.19 เป็นการอธิบายถึงหลักการทำงานของโปรแกรมวิซวลเบสิกซึ่งเป็นโปรแกรมที่ช่วยจัดการกับภาพที่จับได้จากกล้องเว็บแคมมีดังนี้

2.1 จะต้องเปิดโปรแกรมที่เขียนด้วยวิซวลเบสิกขึ้นมา แล้วเราก็ทำการเลือก Com Port ในการเลือก Com Port ต้องเลือก Com Port ที่ต่ออยู่กับพอร์ตสื่อสารที่ต่อมาจากเครื่องตรวจจับความเร็ววัตถุ และทำการเลือก Bit Rate 9600 Bit/sec ซึ่งเป็นความเร็วในการส่งถ่ายข้อมูล แต่หากไม่เลือก Bit Rate ก็ได้ เพราะทางผู้จัดทำทำการกำหนด Bit Rate ไว้ที่ 9600 Bit/sec ซึ่งความสัมพันธ์กับการส่งถ่ายข้อมูลจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ เมื่อทำการเลือกแล้วโปรแกรมจะทำการเชื่อมต่อและรอรับข้อมูลจากไมโครคอนโทรลเลอร์



รูปที่ 3.20 หน้าต่างของโปรแกรมวิซวลเบสิกตอนเริ่มต้น

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click // เป็นการกำหนดฟังก์ชันปุ่มกด
    If Button1.Text = "Connect" Then // เป็นการตรวจสอบว่าทำการเชื่อมต่อเข้ากับกล้องหรือไม่
        If comport.Text <> "" Then // ถ้ามีข้อมูลไม่เท่ากับว่า
            If BaudRateBox1.Text <> "" Then // ถ้าไม่เท่ากับว่างก็มีข้อมูลเป็นข้อความ
                SerialPort1.BaudRate = CInt(BaudRateBox1.Text) // ถ้าเป็นตามเงื่อนไขเพื่อจะรอเปิด
            Else // กรณี
                SerialPort1.BaudRate = 9600 // ถ้ากำหนดบิตเรตที่ 9600
                BaudRateBox1.Text = SerialPort1.BaudRate.ToString // ทำการแปลงข้อมูลเป็นสตริง
            End If // ถ้า
        End If // ถ้าไม่ได้กำหนดว่าจะทำการบันทึกภาพที่ใด
        If ComboBox1.Text = "" Then // กำหนดเซฟในไดฟซี
            ComboBox1.Text = "C:\\"
        End If // ถ้า
        InitializePictureBox1() // เตรียมแสดงภาพ
        OpenPreviewWindow() // ทำการเปิดภาพ
    End If
End Sub

```

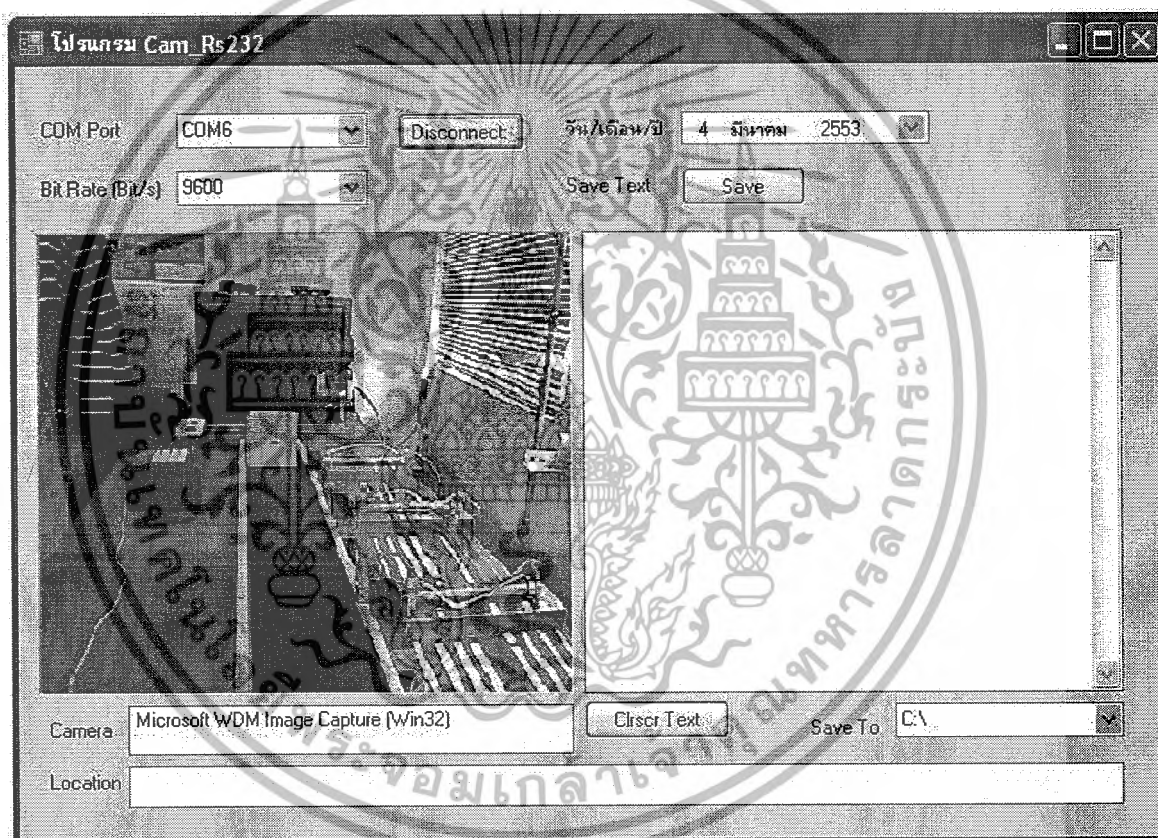
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

With SerialPort1 // จากพอร์ตที่ต่อจากกล้อง
.PortName = comport.Text // ให้ใส่ชื่อพอร์ตที่จะทำการเชื่อมต่อ
.DataBits = 8 // กำหนดบิตในการเชื่อมต่อเท่ากับ 8
.StopBits = IO.Ports.StopBits.One // กำหนดคให้หยุดการเชื่อมต่อ เมื่อ ไม่มีข้อมูลในการเชื่อมต่อ
.Parity = IO.Ports.Parity.None // จะแสดงชื่อพอร์ตในการเชื่อมต่อในระหว่างการเชื่อมต่อ
End With // จบ

```

2.2 เมื่อทำการติดต่อกับกล้องเรียบร้อยแล้ว โปรแกรมก็จะทำการแสดงภาพที่ถ่ายจากกล้องเว็บแคมบนหน้าต่างของโปรแกรกดังรูป



รูปที่ 3.21 หน้าต่างโปรแกรมวิศวกรรมแสดงภาพจากกล้องเว็บแคม

```

hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0, 0, 640, _
480, PictureBox1.Handle.ToInt32, 0) // เปิดแสดงผลที่ PictureBox1 สร้าง window
ถูก ขึ้นมาด้วย ฟังก์ชัน capCreateCaptureWindowA ซึ่งคุณสามารถเห็นใน picturebox.

```

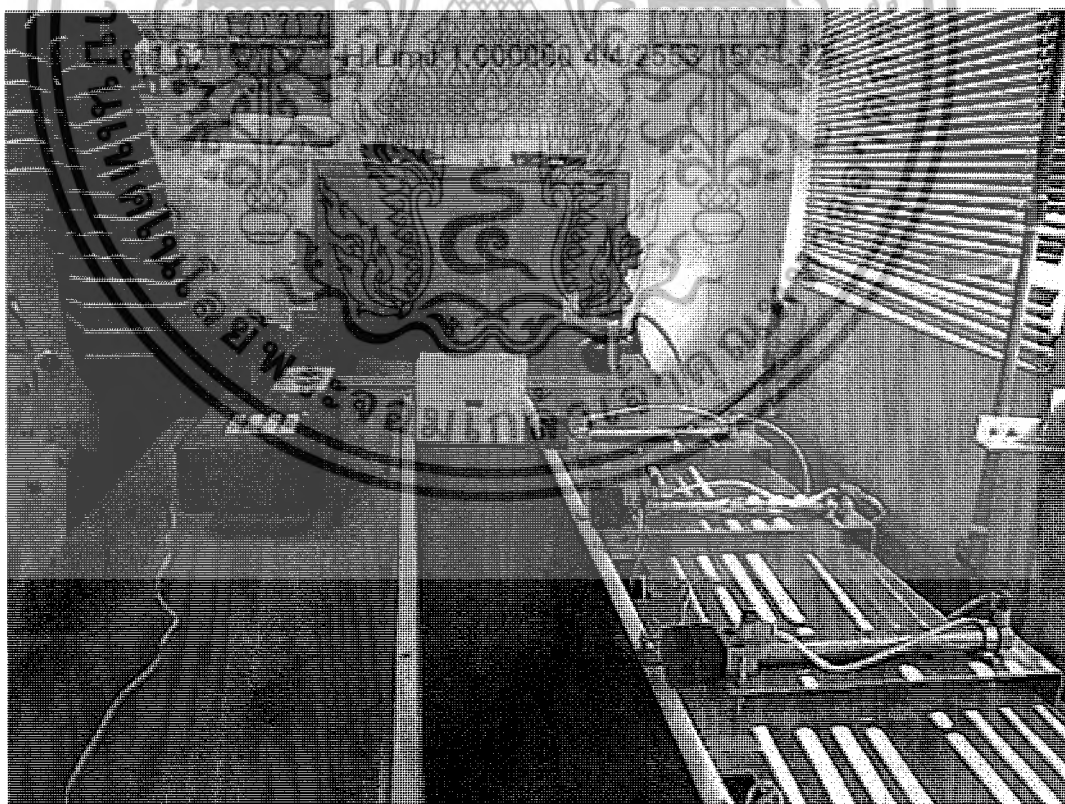
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then // ติดต่อกับ
device
    SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0) // ตั้งค่า
preview scale
    SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 66, 0) // ตั้งค่า
preview rate ในระดับ milliseconds
    SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0) // เริ่มต้นการแสดงผล จาก
กล้อง

```

2.3 เมื่อมีการส่งค่าจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ โปรแกรมวิซวลเบสิก ก็จะทำการจับภาพและขึ้นข้อความ และรายละเอียดความเร็วด้านมุมบนด้านซ้ายมือด้วยตัวอักษรสีแดง แล้วทำการเก็บภาพที่ระบุข้อมูลไว้ให้อยู่ในโปรแกรม โดยมีการทำงานเป็นอย่างไรเรื่อยไป จนกว่าจะจบการทำงาน of โปรแกรม ในการเซฟรูปภาพ เมื่อเราได้รูปภาพที่จับภาพ เราต้องทำการมาเก็บในฟังก์ชันอิมเมจเพื่อที่จะทำการเขียนข้อความที่ได้จากพอร์ตอนุกรม RS-232 แล้วทำการจัดเก็บรูปภาพ



รูปที่ 3.22 โปรแกรมวิซวลเบสิกแสดงผลภาพและค่าที่ได้จากกล้องเว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Savepicture(ByVal str As String) // ประกาศตัวแปรที่ใช้ในการเซฟรูปภาพ
    Dim saveFileDialog1 As New SaveFileDialog() // กำหนดตัวแปรในการเซฟรูปภาพ
    Dim data As IDataObject // กำหนดข้อมูลให้เป็นไอค้ำด้าออปปเจค
    Dim bmp As Image // กำหนดบิตแมพให้เป็นรูปภาพ
    Dim iHeight As Integer = PictureBox1.Height // กำหนดความสูงของรูปอ้ด โนม้ด
    Dim iWidth As Integer = PictureBox1.Width // กำหนดความกว้างของรูปอ้ด โนม้ด
    Dim dir As String // กำหนดให้รับค่าเป็นสตริง
    Dim formatfile As String = ".jpg" // กำหนดฟอร์แมทให้เป็นสตริง โดยให้เป็น
    คอทเจพีจ
    StrPic = "IMG" // ให้สตริงพิคเป็นรูปภาพ
    StrPic = StrPic + CStr(numPic) // ให้สตริงพิคบวกน้มพิคได้
    numPic = numPic + 1 // ให้น้มพิคเป็นน้มพิคบวกหนึ่ง
    dir = ComboBox1.Text + Dir_Location + "\" + (CStr(DateTimePicker1.Value.Day) + "." +
    CStr(DateTimePicker1.Value.Month) + "." + CStr(DateTimePicker1.Value.Year)) +
    "IMAGES" // ให้รับค่าจากข้อมูลตามด้วยค่าเวลา วันที่ เดือน ปี ในไฟล์อ้มเมจ
    My.Computer.FileSystem.CreateDirectory(dir)
    SendMessage(hWnd, WM_CAP_EDIT_COPY, 0, 0) // นำ ภาพ จาก clipboard และ
    convert มัน ไปเป็น bitmap
    data = Clipboard.GetDataObject() // กำหนดค้ำด้าคริปบอ้ดเป็นคอทเจค
    ค้ำด้าออปปเจค
    If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then // ถ้าข้อมูลคอทเจคค้ำด้า
    แสดงเป็นรูปภาพ
        bmp = CType(data.GetData(GetType(System.Drawing.Bitmap)), Image) // กำหนดบิต
    แมพเป็นไฟล์รูปภาพ
        Dim g As Graphics = Graphics.FromImage(bmp) // กำหนดบิตแมพให้เป็นรูปภาพ
        g.DrawString(strs + " " + DateTime.Now.ToString(), New Font("Aral", 14), New
        SolidBrush(Color.Red), New PointF(15.0F, 15.0F)) // ทำการเริ่มวาดสตริงในพิคที่ (15.0,
        15.0) ของรูปด้วยสีแดง กำหนดรูปแบบตัวอักษรชนิด Aral ขนาด 14 โดยแสดงข้อความที่รับมาจาก
        Serial Port ตามด้วยวันที่และเวลา
        PictureBox1.Image = bmp // ให้รูปในช่อง 1 เป็นบิตแมพ
    
```

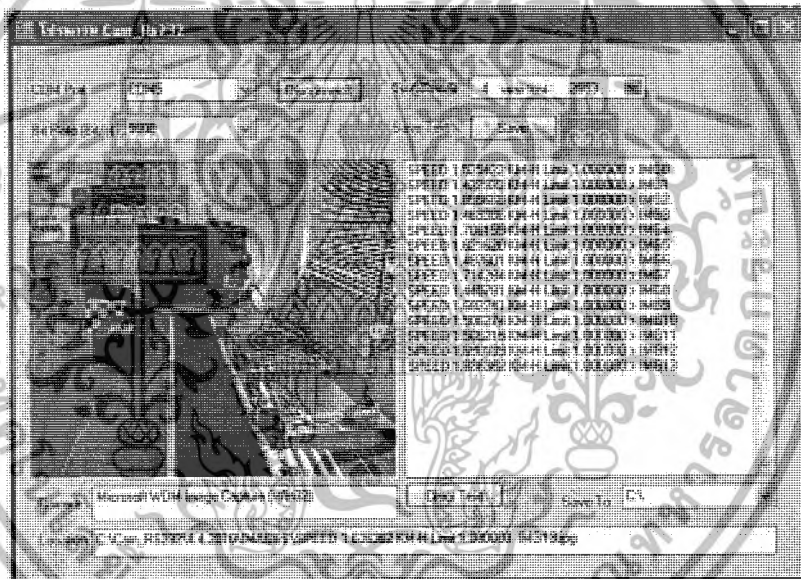
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

saveFileDialog1.CreatePrompt = True // ถ้าไฟล์ใดจะลอกที่ 1 ถูกสร้างเป็น
จริง
saveFileDialog1.OverwritePrompt = True // ถ้าไฟล์ใดจะลอกที่ 1 ถูกแทนที่เป็น
จริง
TextBox1.Text = dir + "\" + str + " " + StrPic + formatfile // กำหนดชื่อไฟล์เป็นค่าที่รับ
ได้จาก Serial Port
bmap.Save(dir + "\" + str + " " + StrPic + formatfile) // ทำการเซฟปิดเมพ
End If // ถ้าจบ
End Sub // เมื่อจบทำการเลิกประกาศตัวแปร

```

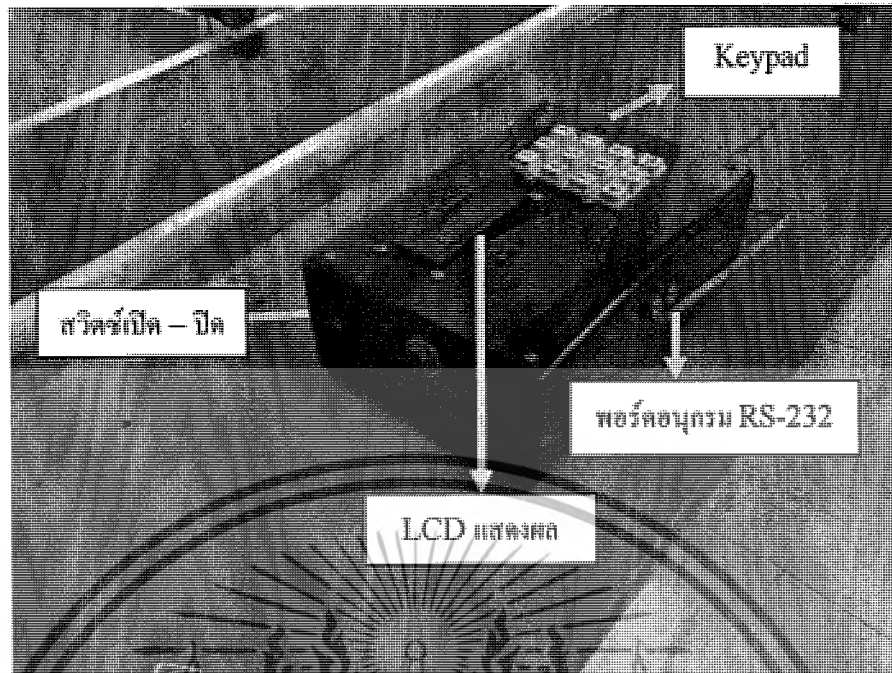
ทำการเขียน โปรแกรม Start Debug ถ้าผ่านก็จะใช้ได้ โดยที่ออกแบบหน้าต่างดังนี้ ดังรูปที่ 3.23



รูปที่ 3.23 หน้าต่างของ โปรแกรมวิซวลเบสิกตอน Start Debug

3. ขั้นตอนการทำอุปกรณ์ส่งกล่อง จากการที่เราได้เลือกออกแบบทั้งทางด้านฮาร์ดแวร์ และซอฟต์แวร์ต่างๆ เรียบร้อยแล้ว เราได้ทำแผ่น PCB และลงอุปกรณ์เรียบร้อยแล้ว เราก็ทำการเจาะรูต่างๆ ที่กล่องเพื่อติดตั้งอุปกรณ์ต่างๆที่จำเป็น โดยในเซนเซอร์ตัวที่ 1 และตัวที่ 2 ห่างกัน 10 เซนติเมตร ตามโปรแกรมที่ระบุความห่างของเซนเซอร์ ควรใช้เวอร์เนียร์วัดเพื่อให้ความละเอียด เพราะฉะนั้นมีผลต่อความผิดพลาดของค่าที่ได้จากการวัด แล้วทำการต่อสายต่อพ่วงต่างๆ แล้วทำการเบิร์นโปรแกรม โดยใช้โปรแกรม FLIP V2.4.4 ทำการปรับค่าที่วงจรเปรียบเทียบกับแรงดัน 5 โวลต์ ให้เรียบร้อยแล้วทำการประกอบกล่องเป็นอันเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 ส่วนประกอบของเครื่องตรวจจับความเร็ววัตถุ



รูปที่ 3.25 เครื่องตรวจจับความเร็ววัตถุและอุปกรณ์ที่ใช้ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและการแสดงผล

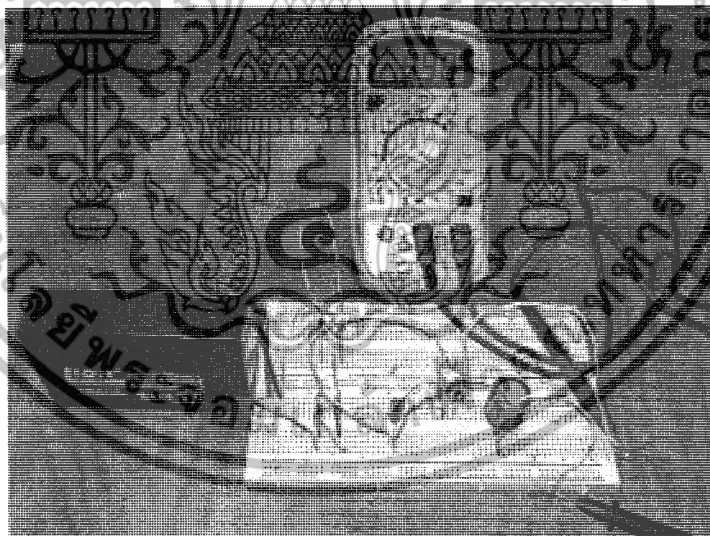
4.1 กล่าวนำ

ในบทนี้เราจะกล่าวถึงการทดลองรวมทั้งผลการทดลอง ตลอดจนแนวคิดซึ่งประกอบด้วยการทดสอบความสอดคล้องกับทฤษฎีของวงจร การทดลองเครื่องตรวจจับความเร็ววัตถุ และการทดลองในส่วนของโปรแกรมวิชาการเบสิก ซึ่งมีรายละเอียดดังต่อไปนี้

4.2 การทดลองความสอดคล้องกับทฤษฎีของวงจร

เป็นการทดลองเพื่อทดสอบว่า วงจรที่เรานำมาใช้งานเป็นไปตามทฤษฎีหรือไม่ โดยจะสามารถแสดงได้ดังนี้

4.2.1 วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์



รูปที่ 4.1 การทดลองวงจรจ่ายกระแสตรง 5 โวลต์

จากวงจรดังรูปที่ 4.1 เป็นวงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ โดยนำมาต่อวงจรลงในบอร์ดทดลองและทำการทดลองโดยการจ่ายไฟกระแสตรง +12V DC จากแบตเตอรี่รีเข้าที่วงจรแปลงแรงดันจ่ายกระแสตรง 5 โวลต์ โดยใช้ไอซี 7805 เป็นตัวแปลงแรงดัน แล้วทำการวัด ผลออกมาได้ DC 4.98 V ซึ่งมีค่าใกล้เคียงกับค่าทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

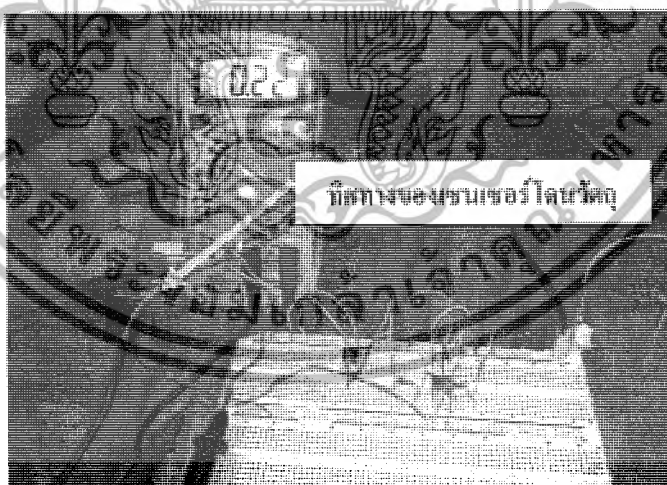
4.2.2 วงจรเปรียบเทียบแรงดัน

นำวงจรเปรียบเทียบแรงดันมาทดลองลงในบอร์ดทดลองโดยใช้ TLC272 ซึ่งเป็นไอซีออปแอมป์ โดยใช้แหล่งจ่ายเพื่อทำการทดลองและทำการวัดค่าจะได้ผลดังนี้



รูปที่ 4.2 การทดลองวงจรเปรียบเทียบแรงดันในสถานะ High

เมื่อไม่มีสิ่งกีดขวางเซนเซอร์ เอาต์พุตที่ได้จากวงจรจะได้ค่า 4.32 โวลต์ ซึ่งจะเป็นการทำงานในสถานะ High ในอุปกรณ์ไมโครคอนโทรลเลอร์ซึ่งมีค่าใกล้เคียงกับค่าทางทฤษฎี



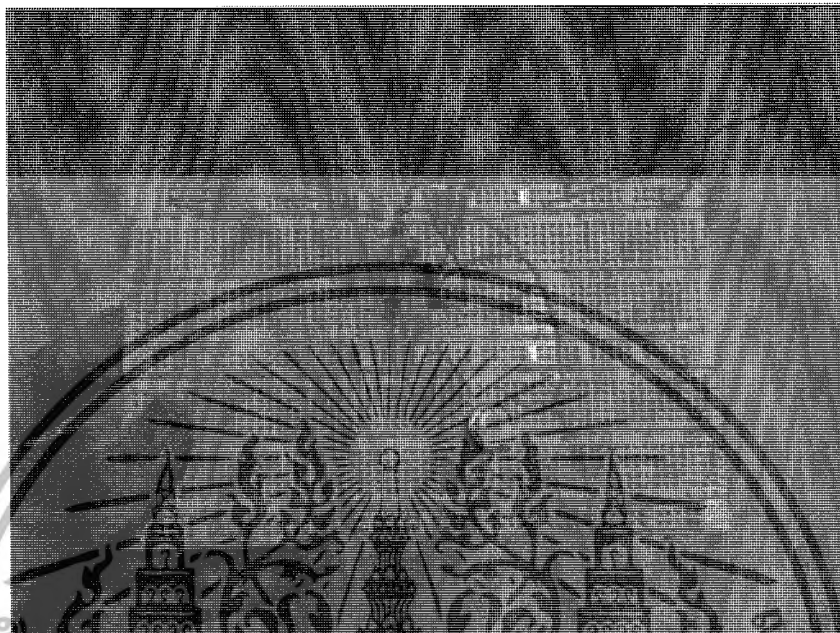
รูปที่ 4.3 การทดลองวงจรเปรียบเทียบแรงดันในสถานะ Low

เมื่อมีสิ่งกีดขวางเซนเซอร์ เอาต์พุตที่ได้จากวงจรจะได้ค่า 0.22 โวลต์ ซึ่งจะเป็นการทำงานในสถานะ Low ในอุปกรณ์ไมโครคอนโทรลเลอร์ซึ่งมีค่าใกล้เคียงกับค่าทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 วงจร Buzzer

วงจร Buzzer เพื่อแสดงสัญญาณเตือน โดยใช้ขา B ของทรานซิสเตอร์ BC547 เป็นวงจรแบ่งแรงดัน และได้ทำการทดลองในบอร์ดทดลองดังรูปที่ 4.4



รูปที่ 4.4 วงจรขับ Buzzer

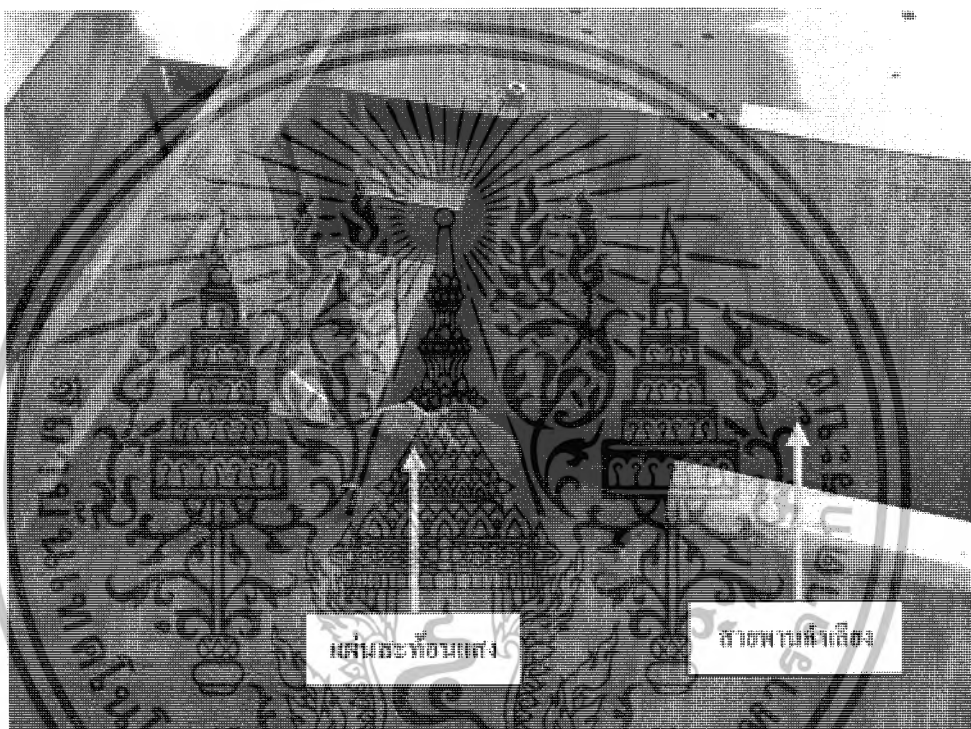
จากการทดลองถ้าเราทำการจ่ายกระแสไฟ +5 โวลต์ ให้กับขาอินพุต หลอด LED จะติด และมีเสียงของ Buzzer ดังขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองเครื่องตรวจจับความเร็ววัตถุโดยใช้กล้องที่เหลื่อมวางบนสายพาน

- ทำการทดลองโดยการการนำกล้องวางบนสายพาน แล้วให้กล้องผ่านเครื่องตรวจจับความเร็วโดยอินเวอร์เตอร์ (Invertor) ซึ่งทำหน้าที่ควบคุมความเร็วรอบในการหมุนของมอเตอร์ที่ใช้ในการขับสายพานด้วยความถี่ต่างๆ ดังนี้คือ 4 Hz, 100 Hz, 200 Hz, 300 Hz, 400 Hz, 500 Hz, 600 Hz, 700 Hz, 800 Hz, 900 Hz และ 999 Hz เพื่อทำการประมวลผลโดยจะเปรียบเทียบกับเครื่องเทคโคเจน โดยทำการส่งเครื่องเทคโคเจนที่เพลาขับโดยทำดังนี้

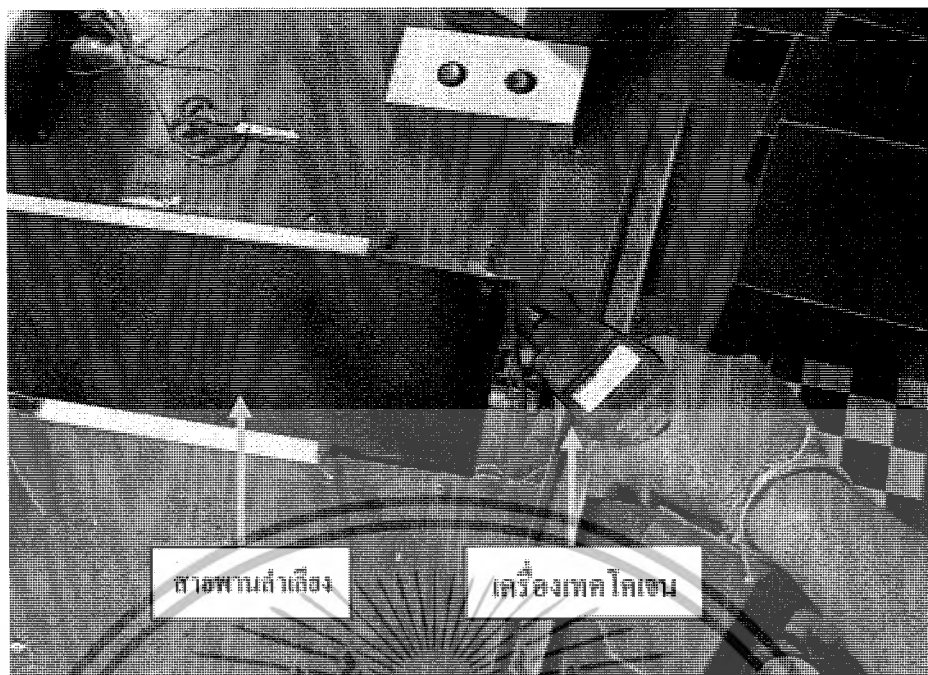
1. ทำการติดแผ่นสะท้อนแสงไว้ 1 แผ่น



รูปที่ 4.5 แผ่นสะท้อนแสงที่ใช้ในการทดลองกับสายพาน

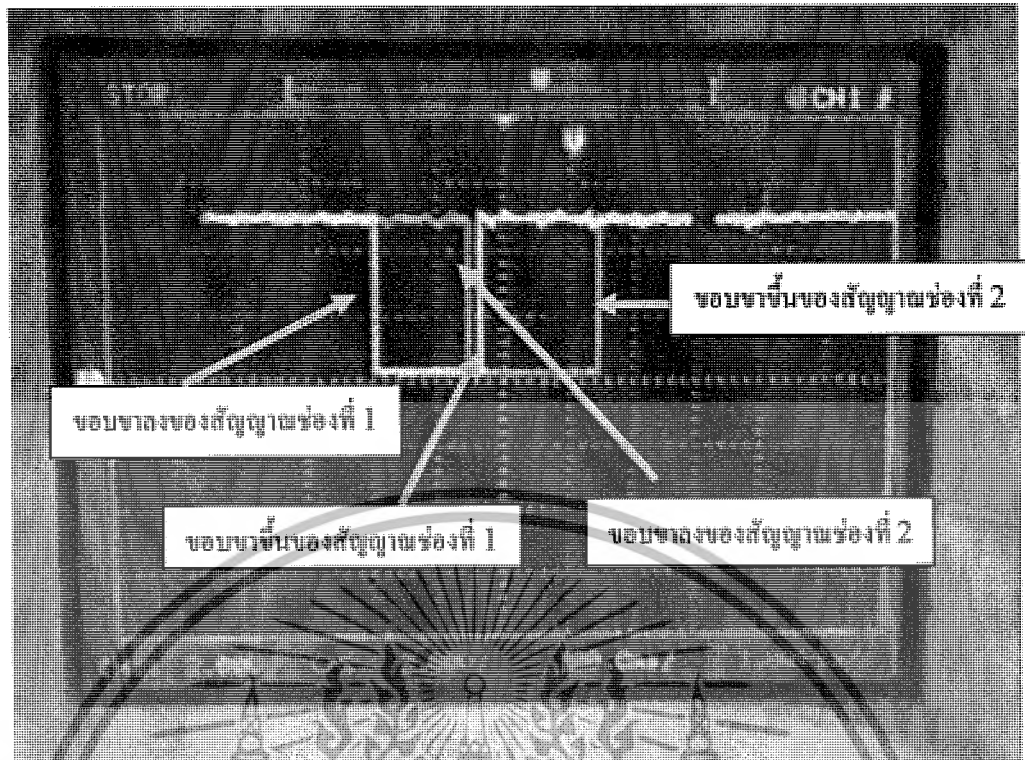
2. เราใช้เครื่องเทคโคเจน ของ Lutron รุ่น DT - 2234A ซึ่งมี Display 5 digit, 10 มิลลิเมตร, 0.4 นิ้ว LCD โดยมีความแม่นยำ (Accuracy) $\pm (0.05 \% + 1 \text{ digit})$ มาทำการวัดความเร็วของเพลาขับสายพานซึ่งมีเส้นรอบวงยาว 30 เซนติเมตร เมื่อได้ค่าความเร็วจากเครื่องเทคโคเจนซึ่งมีหน่วยเป็นรอบต่อนาที เราต้องนำค่าที่ได้มาคำนวณโดยการคูณด้วย 0.3 เมตร และคูณด้วย 60 (นาทึ) และก็จะได้ค่าความเร็วในหน่วย เมตรต่อชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การวัดความเร็วรอบของสายพาด โดยใช้เครื่องเทคโคเจน

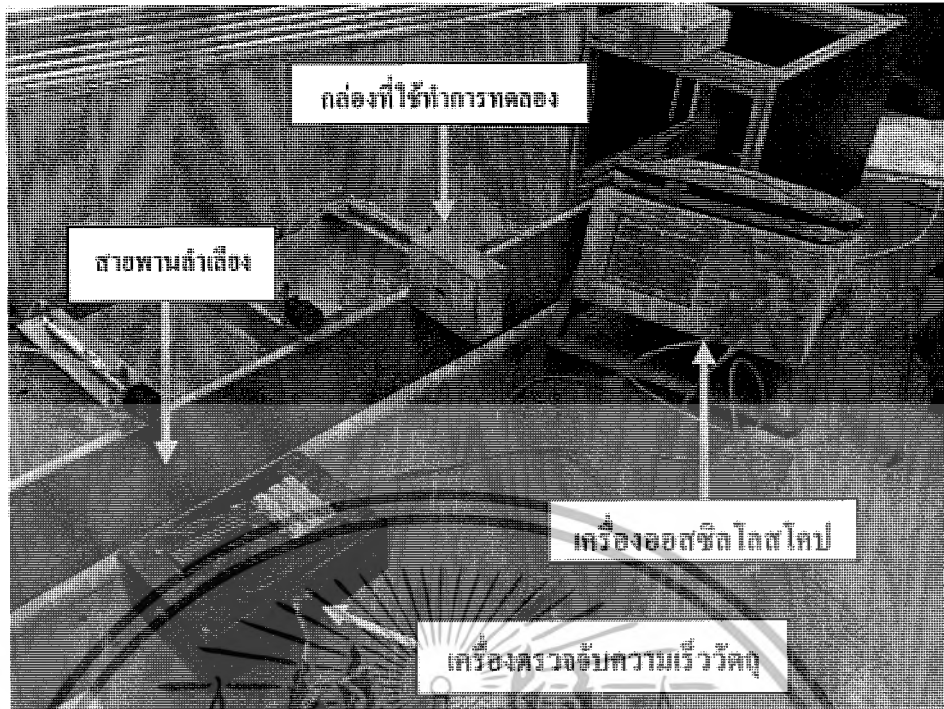
3. ทำการเปรียบเทียบสัญญาณอินพุตของเซนเซอร์ ทั้งสองตัว ด้วยเครื่องออสซิลโลสโคปของ Agillan รุ่น DSO - 3062A โดยทำการต่อสายสโคปช่องที่ 1 เข้ากับอินพุตของเซนเซอร์ตัวที่ 1 ในไมโครคอนโทรลเลอร์ และช่องที่ 2 เข้ากับอินพุตของเซนเซอร์ตัวที่ 2 ในไมโครคอนโทรลเลอร์ และนำค่าความแตกต่างของเวลาในขอบขาลงระหว่างเซนเซอร์ตัวที่ 1 และเซนเซอร์ตัวที่ 2 มาคำนวณหาค่าความเร็วที่ได้หารด้วยระยะทาง 0.1 เมตร แล้วทำการคูณด้วย 3600 จะได้ค่าความเร็วออกมาเป็นหน่วย เมตรต่อชั่วโมง แล้วเราทำการบันทึกค่าลงในผลการทดลอง



รูปที่ 4.7 การแสดงผลออกทางหน้าจอของเครื่องออสซิลโลสโคป

4. เราต้องเปลี่ยนหน่วยเป็น เมตรต่อชั่วโมง โดยการนำค่าที่ได้คูณด้วย 1,000 จะได้เป็นหน่วยเป็น เมตรต่อชั่วโมง แล้วนำค่าที่คำนวณได้มาบันทึกค่าลงในตารางแสดงผลการทดลองซึ่งมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 อุปกรณ์ต่างๆที่ใช้ร่วมกันในการทดลองกับสายพาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดลองความถี่จากสายพานที่ 4 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (sec)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	2.75	130.909090	7.2	129.6	130
2	2.75	130.909090	7.2	129.6	130
3	2.76	130.434780	7.2	129.6	130
4	2.76	130.434780	7.2	129.6	130
5	2.74	131.380000	7.2	129.6	130
6	2.74	131.380000	7.2	129.6	130
7	2.75	130.909090	7.2	129.6	130
8	2.76	130.434780	7.2	129.6	130
9	2.76	130.434780	7.2	129.6	130
10	2.76	130.434780	7.2	129.6	130
			ค่าเฉลี่ย	129.6	130

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\ &= \left[\frac{(129.6 - 130)}{129.6} \right] \times 100 \\ &= 0.3086419753 \% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\ &= \frac{1}{10-1} [(130 - 130) + (130 - 130) + (130 - 130) + \\ &\quad (130 - 130) + (130 - 130) + (130 - 130) + \\ &\quad (130 - 130) + (130 - 130) + (130 - 130) + \\ &\quad (130 - 130)] \end{aligned}$$

$$= 0.0000$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดลองความถี่จากสายพานที่ 100 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	247	1457.489879	79.8	1436.4	1460
2	248	1451.612903	79.8	1436.4	1450
3	248	1451.612903	79.8	1436.4	1450
4	246	1592.920354	79.8	1436.4	1460
5	248	1451.612903	79.8	1436.4	1450
6	251	1434.262948	79.8	1436.4	1440
7	251	1434.262948	79.8	1436.4	1440
8	250	1440.000000	79.8	1436.4	1440
9	250	1440.000000	79.8	1436.4	1440
10	249	1445.78313	79.8	1436.4	1450
			ค่าเฉลี่ย	1436.4	1448

$$\begin{aligned}
 \text{ค่าความแม่นยำ} &= \frac{[\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}]}{\text{ค่าที่จริงตามทฤษฎี}} \times 100 \\
 &= \frac{[(1436.4 - 1448)]}{1436.4} \times 100 \\
 &= 0.8075744918 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q})^2 \\
 &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q})^2 \\
 &= \frac{1}{10-1} [(1460 - 1448)^2 + (1450 - 1448)^2 + (1450 - 1448)^2 + \\
 &\quad (1460 - 1448)^2 + (1450 - 1448)^2 + (1440 - 1448)^2 + \\
 &\quad (1440 - 1448)^2 + (1440 - 1448)^2 + (1440 - 1448)^2 + \\
 &\quad (1450 - 1448)^2]
 \end{aligned}$$

$$= 7.1111$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการทดลองความถี่จากสายพานที่ 200 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	214	1682.242991	93.2	1677.6	1680
2	215	1674.418605	93.2	1677.6	1680
3	215	1674.418605	93.2	1677.6	1680
4	213	1690.140845	93.2	1677.6	1690
5	217	1658.986175	93.2	1677.6	1670
6	214	1682.242991	93.2	1677.6	1680
7	214	1682.242991	93.2	1677.6	1680
8	214	1682.242991	93.2	1677.6	1680
9	213	1690.140845	93.2	1677.6	1690
10	213	1690.140845	93.2	1677.6	1690
			ค่าเฉลี่ย	1677.6	1682

$$\begin{aligned}
 \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\
 &= \left[\frac{(1677.6 - 1682)}{1677.6} \right] \times 100 \\
 &= 0.2622794468 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q})^2 \\
 &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q})^2 \\
 &= \frac{1}{10-1} [(1680 - 1682) + (1680 - 1682) + (1680 - 1682) + \\
 &\quad (1690 - 1682) + (1670 - 1682) + (1680 - 1682) + \\
 &\quad (1680 - 1682) + (1680 - 1682) + (1690 - 1682) + \\
 &\quad (1690 - 1682)]
 \end{aligned}$$

$$= 5.3333$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลการทดลองความถี่จากสายพานที่ 300 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	164	2195.121951	123.5	2223	2220
2	164	2195.121951	123.5	2223	2220
3	162	2222.222222	123.5	2223	2220
4	164	2195.121951	123.5	2223	2220
5	164	2195.121951	123.5	2223	2220
6	162	2222.222222	123.5	2223	2220
7	163	2195.121951	123.5	2223	2220
8	164	2222.222222	123.5	2223	2220
9	162	2222.222222	123.5	2223	2220
10	164	2195.121951	123.5	2223	2220
			ค่าเฉลี่ย	2223	2220

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\ &= \left[\frac{(2223 - 2220)}{2223} \right] \times 100 \\ &= 0.1349527665 \% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} [(2220 - 2220)^2 + (2220 - 2220)^2 + (2220 - 2220)^2 + \\ &\quad (2220 - 2220)^2 + (2220 - 2220)^2 + (2220 - 2220)^2 + \\ &\quad (2220 - 2220)^2 + (2220 - 2220)^2 + (2220 - 2220)^2 + \\ &\quad (2220 - 2220)^2] \end{aligned}$$

$$= 0.0000$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ผลการทดลองความถี่จากสายพานที่ 400 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทค โจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	149	2416.100000	133.5	2403	2390
2	149	2416.100000	133.5	2403	2410
3	149	2416.100000	133.5	2403	2390
4	150	2400.000000	133.5	2403	2410
5	150	2400.000000	133.5	2403	2410
6	150	2400.000000	133.5	2403	2390
7	149	2416.100000	133.5	2403	2390
8	150	2400.000000	133.5	2403	2410
9	150	2400.000000	133.5	2403	2410
10	150	2400.000000	133.5	2403	2410
			ค่าเฉลี่ย	2403	2402

$$\begin{aligned}
 \text{ค่าความแม่นยำ} &= \frac{(\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้})}{\text{ค่าที่จริงตามทฤษฎี}} \times 100 \\
 &= \frac{(2403 - 2402)}{2403} \times 100 \\
 &= 0.04161464836 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\
 &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\
 &= \frac{1}{10-1} [(2390 - 2402) + (2410 - 2402) + (2390 - 2402) + \\
 &\quad (2410 - 2402) + (2410 - 2402) + (2390 - 2402) + \\
 &\quad (2390 - 2402) + (2410 - 2402) + (2410 - 2402) + \\
 &\quad (2410 - 2402)]
 \end{aligned}$$

$$= 10.6666$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ผลการทดลองความถี่จากสายพานที่ 500 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	142	2535.211268	141	2538	2530
2	141	2553.190000	141	2538	2540
3	142	2535.211268	141	2538	2530
4	141	2553.190000	141	2538	2540
5	141	2553.190000	141	2538	2540
6	142	2535.211268	141	2538	2530
7	142	2535.211268	141	2538	2530
8	142	2535.211268	141	2538	2530
9	141	2553.190000	141	2538	2540
10	142	2535.211268	141	2538	2530
			ค่าเฉลี่ย	2538	2534

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \frac{[\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}]}{\text{ค่าที่จริงตามทฤษฎี}} \times 100 \\ &= \frac{[(2538 - 2534) / 2538] \times 100}{=} \\ &= 0.1576044129 \% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\ &= \frac{1}{10-1} [(2530 - 2534) + (2540 - 2534) + (2530 - 2534) + \\ &\quad (2540 - 2534) + (2540 - 2534) + (2530 - 2534) + \\ &\quad (2530 - 2534) + (2530 - 2534) + (2540 - 2534) + \\ &\quad (2530 - 2534)] \end{aligned}$$

$$= 5.3333$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ผลการทดลองความถี่จากสายพานที่ 600 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	132	2727.270000	151	2718	2720
2	132	2727.270000	151	2718	2710
3	133	2706.766917	151	2718	2710
4	132	2727.270000	151	2718	2720
5	132	2727.270000	151	2718	2720
6	132	2727.270000	151	2718	2730
7	133	2706.766917	151	2718	2720
8	132	2727.270000	151	2718	2710
9	132	2727.270000	151	2718	2720
10	132	2727.270000	151	2718	2720
			ค่าเฉลี่ย	2718	2718

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\ &= \left[\frac{(2718 - 2718)}{2718} \right] \times 100 \\ &= 0\% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Deviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} [(2720 - 2718)^2 + (2710 - 2718)^2 + (2710 - 2718)^2 + \\ &\quad (2720 - 2718)^2 + (2720 - 2718)^2 + (2730 - 2718)^2 + \\ &\quad (2720 - 2718)^2 + (2710 - 2718)^2 + (2720 - 2718)^2 + \\ &\quad (2720 - 2718)^2] \end{aligned}$$

$$= 5.3333$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 ผลการทดลองความถี่จากสายพานที่ 700 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	128	2812.500000	156	2808	2810
2	129	2790.690000	156	2808	2800
3	128	2812.500000	156	2808	2810
4	128	2812.500000	156	2808	2810
5	129	2790.690000	156	2808	2800
6	128	2812.500000	156	2808	2810
7	127	2834.6456690	156	2808	2800
8	129	2790.6976740	156	2808	2820
9	128	2812.5000000	156	2808	2810
10	128	2812.5000000	156	2808	2810
			ค่าเฉลี่ย	2808	2808

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\ &= \left[\frac{2808 - 2808}{2808} \right] \times 100 \\ &= 0\% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\ &= \frac{1}{10-1} [(2810 - 2808) + (2800 - 2808) + (2810 - 2808) + \\ &\quad (2810 - 2808) + (2800 - 2808) + (2808 - 2810) + \\ &\quad (2800 - 2808) + (2820 - 2808) + (2810 - 2808) + \\ &\quad (2810 - 2808)] \end{aligned}$$

$$= 5.3333$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 ผลการทดลองความถี่จากสายพานที่ 800 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโนโลยี		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	123	2926.820000	161	2898	2890
2	123	2926.820000	161	2898	2900
3	124	2903.000000	161	2898	2900
4	124	2903.000000	161	2898	2900
5	124	2903.000000	161	2898	2900
6	123	2926.820000	161	2898	2890
7	124	2903.000000	161	2898	2900
8	124	2903.000000	161	2898	2890
9	123	2926.820000	161	2898	2900
10	123	2926.820000	161	2898	2890
			ค่าเฉลี่ย	2898	2896

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \left[\frac{\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}}{\text{ค่าที่จริงตามทฤษฎี}} \right] \times 100 \\ &= \left[\frac{(2898 - 2896)}{2898} \right] \times 100 \\ &= 0.06901311249 \% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\ &= \frac{1}{10-1} [(2890 - 2896) + (2900 - 2896) + (2900 - 2896) + \\ &\quad (2900 - 2896) + (2900 - 2896) + (2890 - 2896) + \\ &\quad (2900 - 2896) + (2890 - 2896) + (2900 - 2896) + \\ &\quad (2890 - 2896)] \end{aligned}$$

$$= 5.3333$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ผลการทดลองความถี่จากสายพานที่ 900 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	113	3185.840708	170	3060	3060
2	114	3157.894737	170	3060	3070
3	113	3185.840708	170	3060	3060
4	112	3214.285714	170	3060	3040
5	112	3214.285714	170	3060	3040
6	113	3185.840708	170	3060	3060
7	113	3185.840708	170	3060	3060
8	113	3185.840708	170	3060	3060
9	113	3185.840708	170	3060	3060
10	114	3157.894737	170	3060	3070
			ค่าเฉลี่ย	3060	3058

$$\begin{aligned} \text{ค่าความแม่นยำ} &= \frac{[\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้} / \text{ค่าที่จริงตามทฤษฎี}] \times 100}{1} \\ &= \frac{[(3060 - 3058) / 3060] \times 100}{1} \\ &= 0.06535947712\% \end{aligned}$$

$$\begin{aligned} \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q})^2 \\ &= \frac{1}{10-1} [(3060 - 3058)^2 + (3070 - 3058)^2 + (3060 - 3058)^2 + \\ &\quad (3040 - 3058)^2 + (3040 - 3058)^2 + (3060 - 3058)^2 + \\ &\quad (3060 - 3058)^2 + (3060 - 3058)^2 + (3060 - 3058)^2 + \\ &\quad (3070 - 3058)^2] \end{aligned}$$

$$= 8.0000$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 ผลการทดลองความถี่จากสายพานที่ 999 Hz

ครั้งที่	ค่าที่วัดได้จากสโคป		ค่าที่วัดได้จากเทคโคเจน		ค่าความเร็วที่ได้จากเครื่อง ตรวจจับความเร็ว (m / hr)
	เวลา (ms)	ค่าความเร็ว (m / hr)	รอบ / นาที	(m / hr)	
1	114	3364.894000	174	3132	3130
2	114	3364.894000	174	3132	3120
3	115	3130.430000	174	3132	3130
4	115	3130.430000	174	3132	3130
5	115	3130.430000	174	3132	3120
6	115	3130.430000	174	3132	3110
7	114	3364.894000	174	3132	3130
8	114	3364.894000	174	3132	3130
9	115	3130.430000	174	3132	3120
10	115	3130.430000	174	3132	3130
			ค่าเฉลี่ย	3132	3125

$$\begin{aligned}
 \text{ค่าความแม่นยำ} &= \frac{[\text{ค่าที่จริงตามทฤษฎี} - \text{ค่าที่วัดได้}]}{\text{ค่าที่จริงตามทฤษฎี}} \times 100 \\
 &= \frac{[(3132 - 3125) / 3132] \times 100}{=} \\
 &= 0.2234993614 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{ค่า Standard Diviation } S(q_i) &= \frac{1}{n-1} \sum_{i=1}^n (q_i - \bar{q}) \\
 &= \frac{1}{10-1} \sum_{i=1}^{10} (q_i - \bar{q}) \\
 &= \frac{1}{10-1} [(3130 - 3125) + (3120 - 3125) + (3130 - 3125) + \\
 &\quad (3130 - 3125) + (3120 - 3125) + (3110 - 3125) + \\
 &\quad (3130 - 3125) + (3130 - 3125) + (3120 - 3125) + \\
 &\quad (3130 - 3125)] \\
 &= 6.6666
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 สรุปผลการทดลองเครื่องตรวจจับความเร็ววัตถุโดยใช้กล้องที่เหลื่อมวางบนสายพาน
สรุปผลการทดลองเครื่องตรวจจับความเร็ววัตถุ โดยใช้กล้องที่เหลื่อมวางบนสายพานเป็น
การทดลอง โดยการนำค่าความเร็วที่ได้จากเครื่องเทคโนโลยีมาเปรียบเทียบกับค่าความเร็วที่ได้จาก
เครื่องตรวจจับความเร็ววัตถุ โดยใช้เครื่องออสซิลโลสโคปในการตรวจจับสัญญาณที่ได้จาก
เซนเซอร์ จะสังเกตเห็นได้ว่าที่ความถี่อินเวอร์เตอร์เดียวกัน ค่าความเร็วที่วัดค่าได้จากเครื่อง
ตรวจจับความเร็ววัตถุ ค่าที่ได้นั้นจะมีการแกว่งเนื่องมาจากสาเหตุหลักๆ ดังต่อไปนี้

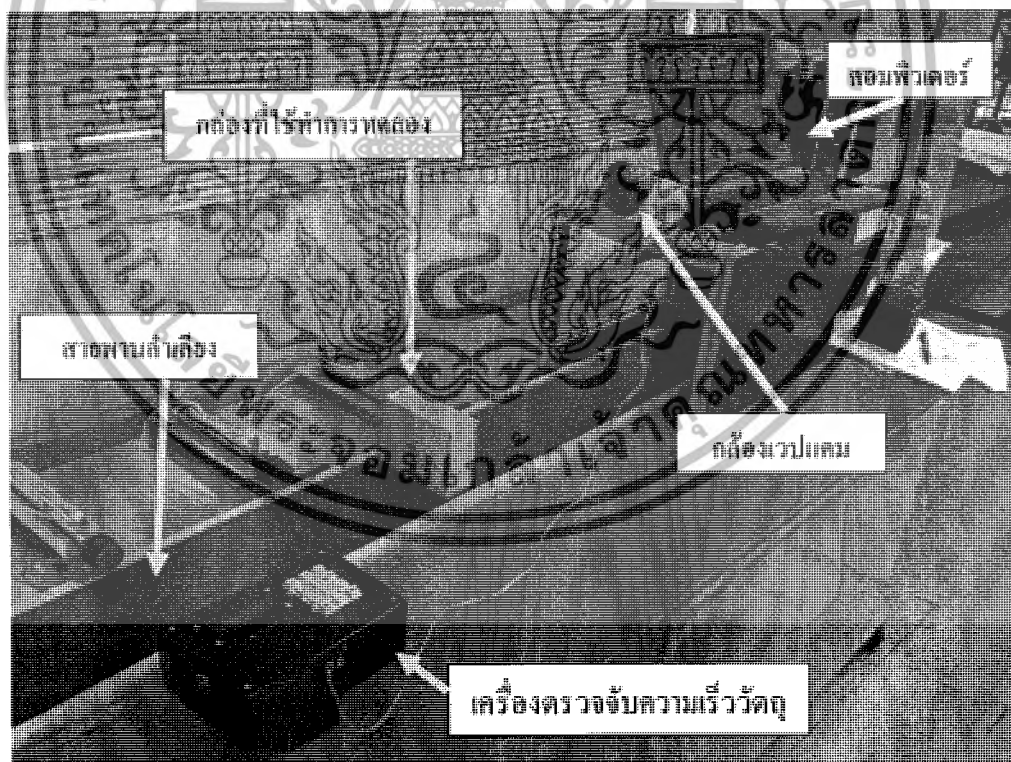
1. เป็นผลอันเนื่องมาจาก ช่วงเวลาในการตอบสนองของเซนเซอร์ หากเราต้องการทำค่าให้
มีความเที่ยงตรงมากขึ้น เราต้องใช้เซนเซอร์ที่มีประสิทธิภาพที่ดีมากกว่านี้
2. จากวงจรเปรียบเทียบแรงดัน จะเห็นได้ว่าจะมีช่วงเวลาในการตอบสนอง ถ้าหากเราต้องการ
ผลการทดลองที่ดีกว่านี้เราต้องทำการเปรียบเทียบวงจร เพื่อที่จะหาวงจรที่มีอัตรา การตอบสนองที่ดีขึ้น
กว่านี้
3. ค่าที่ได้มาจากความละเอียดของเวลา จากการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์จะ
ทำการแก้ไขโดยการเขียนโปรแกรมให้มีความละเอียดมากขึ้น



4.4 ผลการทดลองจากโปรแกรมวิซวลเบสิก (Visual Basic)

ทำการทดลองในส่วนของโปรแกรมวิซวลเบสิก โดยการนำเอากล่องสี่เหลี่ยมมาวางบนสายพานลำเลียง ทำการต่อเครื่องตรวจจับความเร็ว ด้วยพอร์ตอนุกรมกับคอมพิวเตอร์ โดยมีกล่องเวปแคมวางอยู่ด้านบนของสายพานลำเลียง เพื่อคอยจับภาพวัตถุที่มีความเร็วเกินค่าที่กำหนดจากเครื่องตรวจจับความเร็ว โดยมีโปรแกรมวิซวลเบสิก คอยควบคุมการจับภาพจากกล้องเวปแคม เราจะทำการทดลองดังนี้

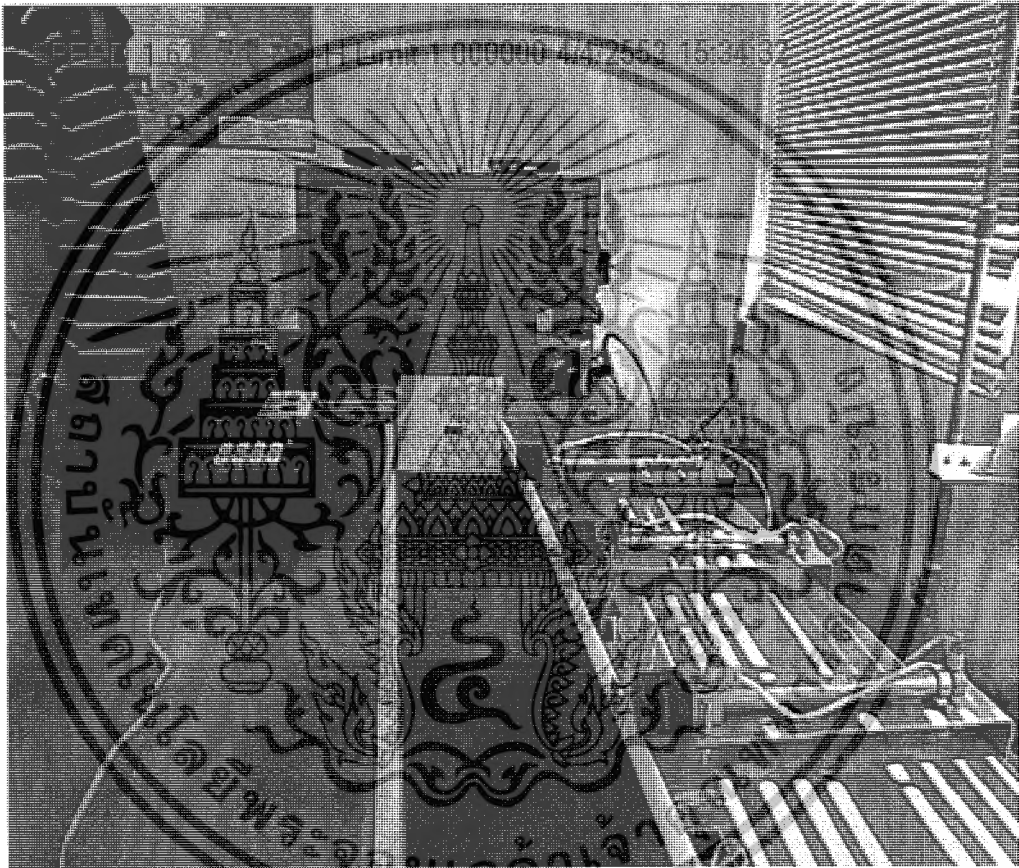
1. ทำการเปิดเครื่องลำเลียงสายพานโดยกำหนดความถี่ที่ 150 Hz
2. การเปิดโปรแกรมวิซวลเบสิก เปิดเครื่องตรวจจับความเร็ว ทำการกำหนดค่าความเร็วสูงสุดไว้ที่ 1.00 กิโลเมตรต่อชั่วโมง โดยทำการกดเครื่องหมาย# (สี่เหลี่ยม) จำนวน 1 ครั้ง เพื่อเข้าสู่เมนูจำกัดความเร็ว สังเกตเห็นได้จากจะมีคำว่าLimit 000.00 km/h แล้วทำการกด * (ดอกจัน) จำนวน 3 ครั้ง เพื่อเลื่อนตัวที่จะเปลี่ยนไปยังหลักหน่วย จากนั้นทำการกดเลข1 แล้วกดเครื่องหมาย# (สี่เหลี่ยม) เพื่อออกจากเมนูจำกัดความเร็ว
3. ทำการวางกล่องสี่เหลี่ยมบนสายพาน ถ้ากล่องนั้นเคลื่อนที่เกินความเร็วที่กำหนด กล้องเวปแคมจะทำการบันทึกภาพดังผลการทดลอง



รูปที่ 4.9 การทดลองของเครื่องตรวจจับความเร็ววัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองทำการจำกัดความเร็วโดยการจำกัดความเร็วกำหนดค่าที่คีย์เพด เครื่องตรวจจับความเร็ว เมื่อวัตถุมีความเกินค่าที่กำหนด โปรแกรมจะทำการจับภาพจากกล้องเวปแคมซึ่งจะถูกควบคุมโดยโปรแกรมวิซวลเบสิก และทำการบันทึกค่าความเร็ว วันที่ และเวลา ด้วยตัวอักษรสีแดงบริเวณด้านบนซ้ายของรูปที่ 4.10 จะเห็นได้ว่าเมื่อวัตถุเคลื่อนที่ด้วยความเร็ว เกินจากค่าที่กำหนดไว้คือ 1.00 km/h โปรแกรมวิซวลเบสิกก็จะทำการส่งกล้องเวปแคมให้จับภาพและเขียนค่าความเร็วที่กล้องนั้นคือ 1.621620 km/h โดยระบุวันที่และเวลาดังนี้ 4/4/2553 15.34.32 ด้วยตัวอักษรสีแดงแล้วทำการบันทึกภาพลงในไฟล์ข้อมูล



รูปที่ 4.10 ผลที่ได้จากการทดลองค่าความเร็วของวัตถุบนสายพาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการจัดทำเครื่องตรวจจับความเร็ววัตถุ เราก็สามารถทำได้สำเร็จตามวัตถุประสงค์ที่กำหนด คือ สามารถตรวจจับความเร็วของวัตถุในแนวระนาบ โดยแสดงผลที่ได้จากการวัดผ่านจอแอลซีดี โดยสามารถจำกัดค่าความเร็วผ่านคีย์เพด เพื่อให้แสดงสัญญาณเตือนเมื่อมีความเร็วเกินค่าที่กำหนดและสื่อสารข้อมูล ส่งค่าความเร็วที่เกินผ่านทางพอร์ตอนุกรมไปติดต่อกับคอมพิวเตอร์ เพื่อให้กล้องเวปแคมจับภาพ โดยให้โปรแกรมวิซวลเบสิก เป็นตัวควบคุมการทำงานของกล้องเวปแคม และแสดงข้อมูล ความเร็วที่เกินและบันทึกภาพผ่านทางโปรแกรมวิซวลเบสิกได้

5.2 ปัญหาที่เกิดขึ้นขณะปฏิบัติงานและการแก้ไข

1. การต่อบอร์ดทดลองในบางครั้งพบว่าบอร์ดหลวมเกินไป
2. ในการออกแบบวงจรด้วยโปรแกรมต้องศึกษาข้อมูลจำเพาะให้ดีมิฉะนั้นจะออกแบบวงจรผิดพลาดได้
3. ในการนำอุปกรณ์ลงแผ่น PCB ต้องตรวจดูโปรแกรมควบคุมไปมิฉะนั้นจะเกิดการผิดพลาดได้
4. ในการต่อสายไฟกับอุปกรณ์ต้องตรวจสอบดูให้แน่ใจว่ายังสามารถใช้งานได้คืออยู่หรือไม่
5. ในการนำอุปกรณ์ลงกล่องต้องคอยตรวจสอบดูอย่าให้เกิดการลัดวงจรขึ้นได้
6. ในการปรับจูนวงจรเปรียบเทียบแรงดันต้องใช้ความละเอียด เพื่อให้จะให้เซนเซอร์ 2 ตัววัดค่าได้ใกล้เคียงกัน
7. การเปลี่ยนแปลงของอุณหภูมิมีผลต่อการตอบสนองของเซนเซอร์
8. มุมที่เซนเซอร์กระทำต่อวัตถุมีผลต่อการตอบสนองของเซนเซอร์
9. ลักษณะพื้นผิวของวัตถุมีผลต่อการตอบสนองของเซนเซอร์
10. ในการเขียนโปรแกรมต้องทำการปรึกษาอาจารย์เพื่อให้คำแนะนำ โดยเฉพาะเรื่องของเงื่อนไขต่างๆ
11. ในการเขียนโปรแกรมต้องทดสอบและปรับใช้เพื่อความเหมาะสม
12. ไม่มีสถานที่และอุปกรณ์ในการปฏิบัติงานที่เพียงพอจึงทำให้เสียเวลาในการทำงาน
13. ใช้งบประมาณในการทำโครงการมากกว่าแผนที่กำหนดไว้ เพราะทำอุปกรณ์ชำรุด

เนื่องจากขาดประสบการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. การศึกษาหลักการของเครื่องตรวจจับความเร็วนี้ทำได้ล้ำซ้ำ เนื่องจากข้อมูลดังกล่าวมีอยู่น้อยมาก

5.3 แนวทางในการพัฒนา

ในการนำไปใช้จริง ควรมีการเพิ่มระยะห่างระหว่างเซนเซอร์ให้เพิ่มมากขึ้นและทำการจับเวลาโดยให้มีความละเอียดมากที่สุด ซึ่งจะกำหนดการนับจำนวนพัลส์ในไทม์เมอร์กับค่าเวลาที่จะเกิดโอเวอร์โฟลว์ ทั้งนี้จะได้ค่าที่มีความละเอียดและแม่นยำมากขึ้น

แนวทางในการพัฒนานั้นสามารถที่จะทำการพัฒนาได้หลายรูปแบบ เช่นในส่วนของแมคคานิกส์สามารถที่จะทำให้มีขนาดกะทัดรัดลงได้อีก คือการเปลี่ยนอุปกรณ์อิเล็กทรอนิกส์ให้เป็นแบบเซอร์เฟสให้หมด และในส่วนของโปรแกรมวิชวลเบสิกก็ยังสามารถเพิ่มแอปพลิเคชันอื่นๆลงไปได้อีกเพื่อความสวยงามให้เหมาะสมกับการใช้งาน

ถ้ามีการนำไปใช้จริง เราสามารถสั่งชิ้นส่วนอุปกรณ์ครั้งละหลายๆ ได้ในราคาที่ถูกลง โดยที่ไม่ต้องนำเข้าอุปกรณ์สำเร็จรูปจากต่างประเทศที่มีราคาแพง และไม่ทำให้ประเทศชาติเสียดุลทางการค้า

บรรณานุกรม

1. Hugh D. Young and Roger A. Freedman เขียน ปิยพงษ์ สิทธิคง แปลและเรียบเรียง, ฟิสิกส์ระดับอุดมศึกษา เล่ม 1 กลศาสตร์, บริษัท เพียร์สัน เอ็ดดูเคชัน อินโดไชน่า, 2547
2. กิตินันท์ พลสวัสดิ์, เริ่มต้น Visual Basic 2008 ฉบับโปรแกรมเมอร์, บริษัท ไอดีซี อินโฟคิสทรีบิวเตอร์ เซ็นเตอร์ จำกัด, 2552
3. ชัยวัฒน์ ทวีจันทร์, Advanced Visual Basic ควบคุมอุปกรณ์ฮาร์ดแวร์, บริษัท ชัคเซส มีเดีย จำกัด, 2551
4. ทีมงานสมาร์ทเลิร์นนิ่ง, เรียนรู้ไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C พร้อมโครงการ ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง, 2552
5. ทีมงานสมาร์ทเลิร์นนิ่ง, ออกแบบลายวงจรพิมพ์ด้วย Protel 99SE, ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง, 2551
6. วิสรุต ศรีรัตนะ, เซนเซอร์และทรานสดิวเซอร์ในงานอุตสาหกรรม, บริษัท ซีเอ็ดดูเคชัน จำกัด (มหาชน), 2550
7. รองศาสตราจารย์ สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51, จัดพิมพ์โดยภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สจล., 2550
8. <http://www.altiumthai.com>
9. <http://www.c51c51.com>
10. <http://writer.dek-d.com>
11. <http://www.electfree.com>
12. <http://www.kkw.ac.th>
13. <http://www.scribd.com>
14. <http://www.thaimicrotron.com>
15. <http://www.vec thai.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมภาษา C ให้ไมโครคอนโทรลเลอร์ MCS-51

```

#include <reg52.h>
#include "stdio.h"

/* AT89C51AC3 SFR */
sfr CKCON = 0x8F; // Clock Control
char lcdbuf[16+1]; // Buff LCD
unsigned char num[]={ 'n','1','2','3','4','5','6','7','8','9','*','0','#' }; //กำหนดคิ้วแปร num
char numbuf[5]; //กำหนด numbuf เท่ากับ 5

/*Key Switch*/
#define KeySwitch P0 //กำหนด P0 เป็น Key Switch
sbit sw1 = P0^4; //กำหนด SW1 ที่ P0.4
sbit sw2 = P0^5; //กำหนด SW2 ที่ P0.5
sbit sw3 = P0^6; //กำหนด SW3 ที่ P0.6

/*Sensor Input*/
sbit Sensor_R = P3^2; //กำหนด Sensor_R ที่ P3.2
sbit Sensor_L = P3^3; //กำหนด Sensor_L ที่ P3.3
sbit beep = P1^7; //กำหนด beep ที่ P1.7
sbit LED = P1^0; //กำหนด LED ที่ P1.0
sbit SHUTTER = P1^1; //กำหนด SHUTTER ที่ P1.1
sbit LED_AMR = P1^2; //กำหนด LED_AMR ที่ P1.2

float time_100u_sec=0; //กำหนด time_100u_sec เท่ากับ 0
float Km_hour=0; //กำหนด Km_hour เท่ากับ 0
float numSET=0; //กำหนด numSET เท่ากับ 0
char setlinelcd=0x46; //กำหนด setlinelcd เท่ากับ 0
char menu=0; //กำหนด menu เท่ากับ 0
char on=0; //กำหนด on เท่ากับ 0
char i=0; //กำหนด i เท่ากับ 0
unsigned char number_time=0; //กำหนด number_time เท่ากับ 0

sbit BK = P2^0;
sbit RS = P2^1; // RS LCD
(0=Instruction,1=Data)
sbit RW = P2^2; // RW LCD (0=Write,1=Read)
sbit E = P2^3; // Enable LCD(Active = "1")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//char lcdbuf[16+1]; // LCD Display Buffer
//*****Function*****
**/
void init_lcd(void); // Initial Character LCD(4-Bit
Interface)
void gotolcd(unsigned char); // Set Cursor LCD
void write_ins(unsigned char); // Write Instruction LCD
void write_data(unsigned char); // Write Data LCD
void enable_lcd(void); // Enable Pulse
char busy_lcd(void); // Read Busy LCD Status
void printlcd(char *str); // Display Message LCD

void clrscr_lcd(void); //Clrscr LCD
void DispCounter(float count); //Stop Counter
unsigned char Hex2ascLO(unsigned char dat); // Read Hex2ascLO
unsigned char Asc2hex(unsigned char dat); // Read Asc2hex
//*****//

/*****
***
Function
*****
**/
//unsigned char Key4X3Switch(void);
void DelayuS(unsigned long timer); // Delay Time
Function(1.4294967295)
void StartTimer(void); // StartTimer Function
void StopTimer(void); // StopTimer Function
void beepk1(void); // beepk1 Function
void beepk2(void); // beepk2 Function
void beeper(void); // beeper Function
void SetspeedLimit(void); // SetspeedLimit Function

/*****
**/
/* serial_init: initialize serial interface */
/*****
**/
void serial_init (void) {
// Initial MCS51 Serial Port
TMOD &= 0x0F; // Reset old Timer1 Mode Config

TMOD |= 0x20; // Update Timer1 = 8 Bit Auto Reload

SCON = 0x50; // Serial Port Mode 1 (N,8,1)

ES = 0; // Disable Serial Interupt

ET1 = 0; // Disable Timer1 Interrupt

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PCON &= 0x7F; // SMOD1 = 0 (Disable Double
Baudrate)

TH1 = 0xF0; // Setup Timer1 Baudrate 9600BPS / 58.9824
MHz

TL1 = 0xF0;
TR1 = 1; // Start Timer1 Generate Baudrate
TI = 1; // Set TI to send First char of UART
}
void Init_Initer(void){
TMOD= 0x01; //(TMOD & 0xF0) | 0x01; //Interrupt Timer 0 Mode 1
TH0 = 0xEC;
TL0 = 0x77; //Interval 0.1mS
IT0 = 1; //Set detect change logic Hi to
Low at INTO
EX0 = 1; // Enable EX0 Interrupt
EX1 = 1; // Enable EX1 Interrupt
IT0 = 1; // Enable IT0
IT1 = 1; // Enable IT1
ET0 = 1; // Enable Timer 0
TR0 = 1; // Start Timer 0
EA = 1; // Enable Interrupt all
}
void Timer_0_service(void) interrupt 1 using 1 {
//service routine Timer interrupt T0 (#1) register bank 1 (using 1)
//delaytime 1 msec

TH0 = 0xEC; //reload TH0, TL0 = 5000H =
1mS
TL0 = 0x77;
if(on==1) //กำหนดเงื่อนไข on เท่ากับ 1
time_1mS_sec+=0.0015; // time_1mS_sec+ = 0.0015
}
void StartTimer(void) { // StartTimer
time_1mS_sec=0; // Set time_1mS_sec เท่ากับ 0
on=1; // Set on เท่ากับ 1
LED=0; // Set LED เท่ากับ 0
}
void StopTimer(void){ //StopTimer
on=0; // Set on เท่ากับ 0
if(time_1mS_sec<=200) //กำหนดเงื่อนไข time_1mS_sec น้อยกว่าหรือเท่ากับ 200
Km_hour=((0.10/time_1mS_sec)*3600); // 1M/10Cm
else Km_hour=0;
LED=1; // Set LED เท่ากับ 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void SeviceIninterrupt_1(void) interrupt 0 using 2 {           //ใช้งาน SeviceIninterrupt_1

    if(menu==0&& i==0){

        StartTimer();
        number_time=0;
        i=1;
    }
}
void SeviceIninterrupt_2(void) interrupt 2 using 3 {           //ใช้งาน SeviceIninterrupt_2
    if(menu==0&& i==1){
        StopTimer();
        i=0;

    } else DelayuS(100000);
}
/*****
***
Function Main
*****/
void main(void){
    CKCON = 0x01;           // Initial X2 Mode (BUS Clock = 58.9824 MHz)

    Init_Initer();
    serial_init();
    init_lcd();           // Initial LCD
    beep=0;
    time_1mS_sec=0;
    clrscr_lcd();           //เคลียร์หน้าจอ LCD
    beepk1();           //beepk1
    Delay
    uS(10000);

    while(1){
        gotolcd(0x00);           // Set Cursor Line-1
        printlcd("Speed");
        gotolcd(0x06);           // Set Cursor Line-1
        DispCounter(Km_hour/1000);
        gotolcd(0x0C);           // Set Cursor Line-1
        printlcd("KM/H");           // Display Line-1
        gotolcd(0x40);
        printlcd("Limit");           //แสดงผล LCD

        gotolcd(setlinelcd);           // Set Cursor Line-1
        DispCounter(numSET);           //Show number speed
        SetspeedLimit();           //SetspeedLimit
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(Km_hour/1000>numSET&&numSET>0&&number_time<30&&i==0){
            number_time++;
            beeper();
            LED_AMR=0;
            if(number_time==1){SHUTTER=0;
            printf("\rSPEED %fKM/H Limit %f
",Km_hour/1000,numSET);
            }
            else SHUTTER=1;

        }else LED_AMR=1;
    }
}

/*****
*
*           Key Switch 4X3
*****
/
unsigned char Key4X3Switch(void){
    unsigned char clo , value=0;
    unsigned char row=0x08;
    for(clo=0;clo<=12;clo+=3){
        KeySwitch=~(row);
        if(sw3==0) value =1+clo;
        else if(sw2==0) value =2+clo;
        else if(sw1==0) value =3+clo;
        row=row/2;
        KeySwitch=~0x00;
    }
    return num[value];
}
/*****
***
***           Select Sound Frequency
*****
/

void sound(unsigned char freq,int time){
    unsigned char i;
    while(time>0){
        beep=1;
        for(i=1;i<=freq;i++)
            time--;
        beep=0;
        for(i=1;i<=freq;i++)
            time--;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void beepk1(void){ sound(100,3000);}
void beepk2(void){ sound(50,1000);}
void beeper(void){ sound(200,30000);}
/*****
*
*           SetspeedLimit to LCD
*****/
/

void SetspeedLimit(void){
char a=0;

if(Key4X3Switch()=='#'){
while(Key4X3Switch()=='#');
gotolcd(setlinelcd+7);
printlcd("SET");
menu=1;
while(Key4X3Switch()!='#'){
if(Key4X3Switch()>='0'&&Key4X3Switch()<='9'){
numbuf[a]=Asc2hex(Key4X3Switch());
}else{
numSET=((long)numbuf[0]*100)
+((long)numbuf[1]*10)
+((long)numbuf[2])
+((long)numbuf[3]*0.1)
+((long)numbuf[4]*0.01);
}

if(a<3)
gotolcd(setlinelcd+a); //Cursor on Blink
else
gotolcd(setlinelcd+a+1);
printlcd("_");
DelayuS(5000);
gotolcd(setlinelcd); // Set Cursor

Line
speed
DispCounter(numSET); //Show number

DelayuS(5000);

if(Key4X3Switch()=='*'){
while(Key4X3Switch()=='*');
if(a<4&&a>=0)a++;
else a=0;
}

}while(Key4X3Switch()=='#');
gotolcd(setlinelcd+7);
printlcd(" ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Km_hour=0;
        menu=0;
        beepk2();
    }
}
//=====//
void DelayuS(unsigned long timer){ //read Delay
while(timer--);
}
/*****/
/* Initial LCD 4-Bit Interface */
/*****/
void init_lcd(void)
{
    unsigned int i; // Delay Count

    E = 0; // Start LCD Control
    (Disable)
    RS = 0; // Default Instruction
    RW = 0; // Default = Write
    Direction
    for (i=0;i<10000;i++); // Power-On Delay (15
    mS)

    P2 &= 0x0F; // Clear old LCD Data
    (Bit[7..4])
    P2 |= 0x30; // DB5:DB4 = 1:1
    enable_lcd(); // Enable Pulse
    for (i=0;i<2500;i++); // Delay 4.1mS

    P2 &= 0x0F; // Clear old LCD Data
    (Bit[7..4])
    P2 |= 0x30; // DB5:DB4 = 1:1
    enable_lcd(); // Enable Pulse
    for (i=0;i<100;i++); // delay 100uS

    P2 &= 0x0F; // Clear old LCD Data
    (Bit[7..4])
    P2 |= 0x30; // DB5:DB4 = 1:1
    enable_lcd(); // Enable Pulse
    while(busy_lcd()); // Wait LCD Execute
    Complete

    P2 &= 0x0F; // Clear old LCD Data
    (Bit[7..4])
    P2 |= 0x20; // DB5:DB4 = 1:0
    enable_lcd(); // Enable Pulse

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(busy_lcd()); // Wait LCD Execute Complete

write_ins(0x28); // Function Set (DL=0 4-
Bit,N=1 2 Line,F=0 5X7)
write_ins(0x0C); // Display on/off Control (Entry Display,Cursor
off,Cursor not Blink)
write_ins(0x06); // Entry Mode Set (I/D=1 Increment,S=0 Cursor Shift)
write_ins(0x01); // Clear Display (Clear Display,Set DD RAM
Address=0)
}
void clrscr_lcd(void){
write_ins(0x01);
}
/*****/
/* Set LCD Cursor */
/*****/
void gotolcd(unsigned char i)
{
i |= 0x80; // Set DD-RAM Address
Command
write_ins(i);
}

/*****/
/* Write Instruction to LCD */
/*****/
void write_ins(unsigned char i)
{
RS = 0; // Instruction Select
RW = 0; // Write Select

P2 &= 0x0F; // Clear old LCD Data (Bit[7..4])
P2 |= i & 0xF0; // Strobe High Nibble Command
enable_lcd(); // Enable Pulse

P2 &= 0x0F; // Clear old LCD Data (Bit[7..4])
P2 |= (i << 4) & 0xF0; // Strobe Low Nibble Command
enable_lcd(); // Enable Pulse

while(busy_lcd()); // Wait LCD Execute Complete
}

/*****/
/* Write Data(ASCII) to LCD */
/*****/
void write_data(unsigned char i)
{
RS = 1; // Data Select
RW = 0; // Write Select

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P2 &= 0x0F; // Clear old LCD Data (Bit[7..4])
P2 |= i & 0xF0; // Strobe High Nibble Data
enable_lcd(); // Enable Pulse

P2 &= 0x0F; // Clear old LCD Data (Bit[7..4])
P2 |= (i << 4) & 0xF0; // Strobe Low Nibble Data
enable_lcd(); // Enable Pulse

while(busy_lcd()); // Wait LCD Execute Complete
}

/*****/
/* Enable Pulse to LCD */
/*****/
void enable_lcd(void) // Enable Pulse
{
    unsigned int i; // Delay Count
    E = 1; // Enable ON
    for (i=0;i<500;i++);
    E = 0; // Enable OFF
}

/*****/
/* Wait LCD Ready */
/*****/
char busy_lcd(void)
{
    unsigned char busy_status; // Busy Status Read

    RS = 0; // Instruction Select
    RW = 1; // Read Direction
    E = 1; // Start Read Busy
    busy_status = P2; // Read LCD Data

    if(busy_status & 0x80) // Read & Check Busy Flag
    {
        E = 0; // Disable Read
        RW = 0; // Default = Write Direction
        return 1; // LCD Busy Status
    }
    else
    {
        E = 0; // Disable Read
        RW = 0; // Default = Write Direction
        return 0; // LCD Ready Status
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/* Print Data(ASCII) to LCD */
/*****/
void printlcd(char *str){
    while(*str){
        // Get ASCII & Write to LCD
        // Write ASCII to LCD
        // Next ASCII
        // End of ASCII (null)
        write_data(*str);
        str++;
    }

    return;
}

unsigned char Hex2ascLO(unsigned char dat){
    dat=dat &0x0f;
    if(dat>9)
        dat=(dat-10)|0x40;
    else
        dat=dat|0x30;
    return dat ;
}

unsigned char Asc2hex(unsigned char dat){
    dat=dat &0x0F;
    return dat ;
}

unsigned char LCD_BUF[10];
void DispCounter(float count){
    unsigned long a;
    a=(long)(count*100);
    LCD_BUF[5]=Hex2ascLO(a%10);
    a/=10;
    LCD_BUF[4]=Hex2ascLO(a%10);
    a/=10;
    LCD_BUF[3]='.';
    LCD_BUF[2]=Hex2ascLO(a%10);
    a/=10;
    LCD_BUF[1]=Hex2ascLO(a%10);
    a/=10;
    LCD_BUF[0]=Hex2ascLO(a%10);
    //a/=10;
    printlcd(LCD_BUF);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรม วิชาเว็บติก

```

Imports System
Imports System.IO.Ports
Imports System.Threading.Thread
Public Class Camery
    Const WM_CAP As Short = &H400S
    Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    Const WS_CHILD As Integer = &H40000000
    Const WS_VISIBLE As Integer = &H10000000
    Const SWP_NOMOVE As Short = &H2S
    Const SWP_NOSIZE As Short = 1
    Const SWP_NOZORDER As Short = &H4S
    Const HWND_BOTTOM As Short = 1
    Dim iDevice As Integer = 0 ' Normal device ID
    Dim hHwnd As Integer ' Handle value to preview window
    ' Declare function from AVI capture DLL.
    Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
        (ByVal hwnd As Integer, ByVal wParam As Integer, ByVal lParam As Integer, _
        ByVal lParam As Object) As Integer
    Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal hwnd
As Integer, _
        ByVal hWndInsertAfter As Integer, ByVal x As Integer, ByVal y As Integer, _
        ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer) As Integer
    Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Integer) As
Boolean
    Declare Function capCreateCaptureWindowA Lib "avicap32.dll" _
        (ByVal lpszWindowName As String, ByVal dwStyle As Integer, _
        ByVal x As Integer, ByVal y As Integer, ByVal nWidth As Integer, _
        ByVal nHeight As Short, ByVal hWndParent As Integer,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ByVal nID As Integer) As Integer

```

Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal wDriver As
Short, _
    ByVal lpszName As String, ByVal cbName As Integer, ByVal lpszVer As
String, _
    ByVal cbVer As Integer) As Boolean
Public Delegate Sub StringSubPointer(ByVal Buffer As String)
Dim StrPic As String = "IMG"
Dim Dir_Location As String = "Cam_RS232"
Dim numPic As Integer = 0
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    LoadDeviceList()
    If ListBox1.Text <> "" Then
        ListBox1.Items.Add("ไม่พบ WebCam")
    End If
    For Each COMString As String In My.Computer.Ports.SerialPortNames ' Load
all available COM ports.
        comport.Items.Add(COMString)
    Next
    BaudRateBox1.Items.Add("110")
    BaudRateBox1.Items.Add("300")
    BaudRateBox1.Items.Add("600")
    BaudRateBox1.Items.Add("1200")
    BaudRateBox1.Items.Add("1800")
    BaudRateBox1.Items.Add("2400")
    BaudRateBox1.Items.Add("4800")
    BaudRateBox1.Items.Add("7200")
    BaudRateBox1.Items.Add("9600")
    BaudRateBox1.Items.Add("14400")
    BaudRateBox1.Items.Add("19200")
    BaudRateBox1.Items.Add("38400")
    BaudRateBox1.Items.Add("57600")
    BaudRateBox1.Items.Add("115200")
    ComboBox1.Items.Add("C:\")
    ComboBox1.Items.Add("D:\")

    Button1.Text = "Connect"
    Timer1.Stop()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TextBox2.Multiline = True
TextBox2.WordWrap = True
save.Enabled = False
Button2.Enabled = False

```

```

End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If Button1.Text = "Connect" Then
        If comport.Text <> "" Then
            If BaudRateBox1.Text <> "" Then
                SerialPort1.BaudRate = CInt(BaudRateBox1.Text)
            Else
                SerialPort1.BaudRate = 9600
                BaudRateBox1.Text = SerialPort1.BaudRate.ToString
            End If
            If ComboBox1.Text = "" Then
                ComboBox1.Text = "C:\\"
            End If
            InitializePictureBox1()
            OpenPreviewWindow()
            With SerialPort1
                .PortName = comport.Text
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Parity = IO.Ports.Parity.None
            End With

            SerialPort1.Open()

            Button1.Text = "Disconnect:"
            Timer1.Interval = 350
            Timer1.Start()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

save.Enabled = True
Button2.Enabled = True
Else
MsgBox("No COM Port Selected")
End If

```

```

Else
SerialPort1.Close()
Timer1.Stop()
Button1.Text = "Connect"
save.Enabled = False
Button2.Enabled = False
ClosePreviewWindow()
End If
End Sub
Private Sub InitializePictureBox1()

Me.PictureBox1.TabStop = False
' Set the SizeMode property to the StretchImage value. This
' will shrink or enlarge the image as needed to fit into
' the PictureBox.
Me.PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
' Set the border style to a three-dimensional border.
Me.PictureBox1.BorderStyle = BorderStyle.Fixed3D
' Add the PictureBox to the form.
Me.Controls.Add(Me.PictureBox1)

End Sub

Private Sub LoadDeviceList()
Dim strName As String = Space(100)
Dim strVer As String = Space(100)
Dim bReturn As Boolean
Dim x As Integer = 0

' เรียกรายชื่อของ device กล้องทั้งหมดในเครื่องคอมของเรา และโชว์ใน List Box ที่ชื่อ lstDevices .

Do
' เรียกชื่อ Driver และ version
bReturn = capGetDriverDescriptionA(x, strName, 100, strVer, 100)
' ถ้าใช่ device ให้เพิ่มชื่อเข้าไปใน list box

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If bReturn Then ListBox1.Items.Add(strName.Trim)
x += 1
Loop Until bReturn = False
End Sub

Private Sub OpenPreviewWindow()
Dim iHeight As Integer = PictureBox1.Height
Dim iWidth As Integer = PictureBox1.Width

' เปิดแสดงผลที่ PictureBox1
' สร้าง window ลูก ขึ้นมาด้วย ฟังก์ชัน capCreateCaptureWindowA ซึ่งคุณสามารถเห็นใน
picturebox.

hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD,
0, 0, 640, _
480, PictureBox1.Handle.ToInt32, 0)

' ติดต่อกับ device
If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then

' ตั้งค่า preview scale
SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)

' ตั้งค่า preview rate ในระดับ milliseconds
SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 66, 0)

' เริ่มต้นการแสดงผล จากกล้อง
SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)

' ปรับขนาด window ให้เท่ากับใน PictureBox1
SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, PictureBox1.Width,
PictureBox1.Height, _
SWP_NOMOVE Or SWP_NOZORDER)
' save.Enabled = True

Else
' การติดต่อ device Error ให้ ปิด window
DestroyWindow(hHwnd)
' btnSave.Enabled = False
End If
End Sub
Private Sub ClosePreviewWindow()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

' เลิกติดต่อกับ device
SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)

' ปิด window
DestroyWindow(hHwnd)
End Sub
Private Sub Savepicture()
    Dim saveFileDialog1 As New SaveFileDialog()
    Dim data As IDataObject
    Dim bmap As Image
    Dim iHeight As Integer = PictureBox1.Height
    Dim iWidth As Integer = PictureBox1.Width
    Dim dir As String
    Dim formatfile As String = ".jpg"
    StrPic = "IMG"
    StrPic = StrPic + CStr(numPic)
    numPic = numPic + 1
    dir = ComboBox1.Text + Dir_Location + "\" +
(CStr(DateTimePicker1.Value.Day) + "." + CStr(DateTimePicker1.Value.Month) _
+ "." + CStr(DateTimePicker1.Value.Year)) + "\IMAGES"
    My.Computer.FileSystem.CreateDirectory(dir)
    SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
    ' นำ ภาพ จาก clipboard และ convert มันไปเป็น bitmap
    data = Clipboard.GetDataObject()
    If data.GetDataPresent(GetType(System.Drawing.Bitmap)) Then
        bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)), Image)
        PictureBox1.Image = bmap
        saveFileDialog1.CreatePrompt = True
        saveFileDialog1.OverwritePrompt = True
        TextBox1.Text = dir + "\" + StrPic + formatfile
        bmap.Save(dir + "\" + StrPic + formatfile)
    End If
End Sub

Private Sub save_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles save.Click
    Dim SaveFileDialog1 As New SaveFileDialog()
    Dim dir As String
    dir = ComboBox1.Text + Dir_Location + "\" +
(CStr(DateTimePicker1.Value.Day) + "." + CStr(DateTimePicker1.Value.Month) _

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

+ "." + CStr(DateTimePicker1.Value.Year)) + "\Text"
SaveFileDialog1.Filter = "Text Files (*.txt)|*.txt"
SaveFileDialog1.Title = "Save Received As"
My.Computer.FileSystem.CreateDirectory(dir)
SaveFileDialog1.InitialDirectory = dir
If SaveFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK _
    And SaveFileDialog1.FileName.Length > 0 Then
    My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName,
    TextBox2.Text, False) ' Overwrite file
End If

```

```
End Sub
```

```
Public Delegate Sub mydelegate()
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

```

```

Dim str As String
str = SerialPort1.ReadExisting
If str <> "" Then
    Savepicture()
    TextBox2.AppendText(str + ">" + StrPic + vbCrLf)

```

```
End If
```

```
End Sub
```

' This subroutine is activated when the form is closed. It closes the COM port.
Without such a close command,
' the garbage collector may close the COM port while it is still in use!

```

Private Sub MaxiTesterClosing(ByVal sender As Object, ByVal e As
ComponentModel.CancelEventArgs) Handles MyBase.Closing
    If MessageBox.Show("Do you really want to close the window", "ออกจากโปรแกรม",
    MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
    Windows.Forms.DialogResult.No Then
        e.Cancel = True

```

```

Else
    ClosePreviewWindow()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If SerialPort1.IsOpen Then SerialPort1.Close() ' Close COM port when the
form is terminated with [X]
```

```
End If
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
```

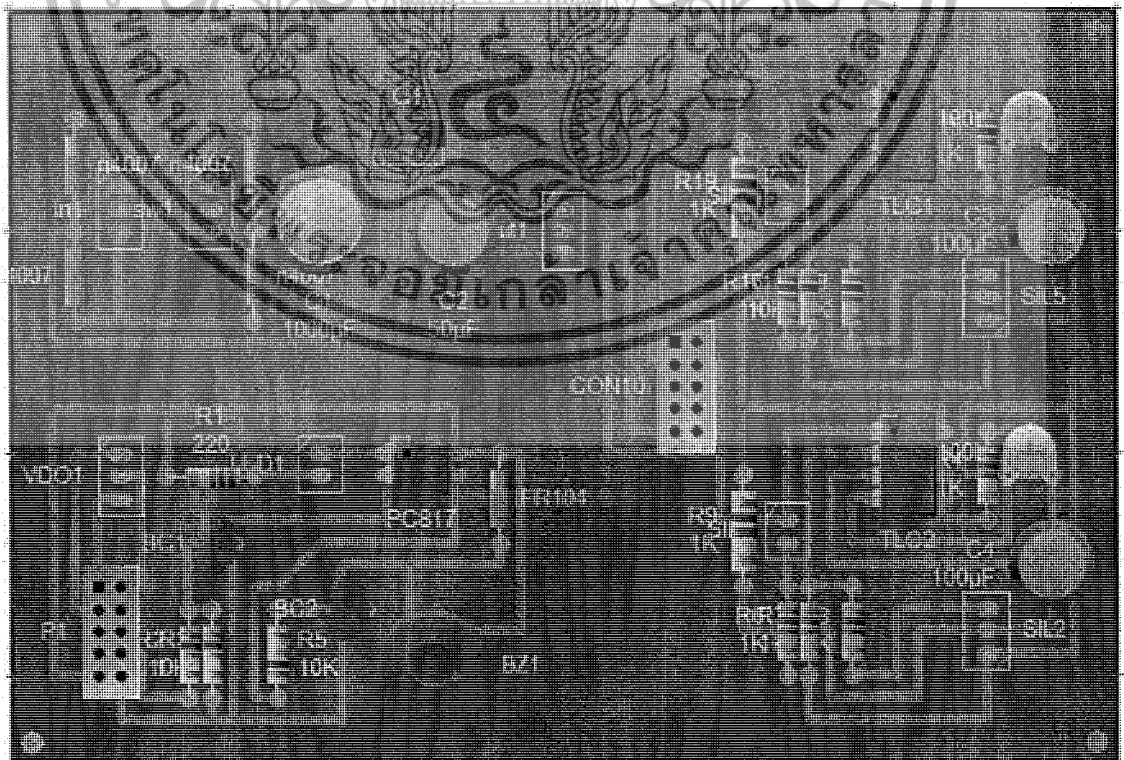
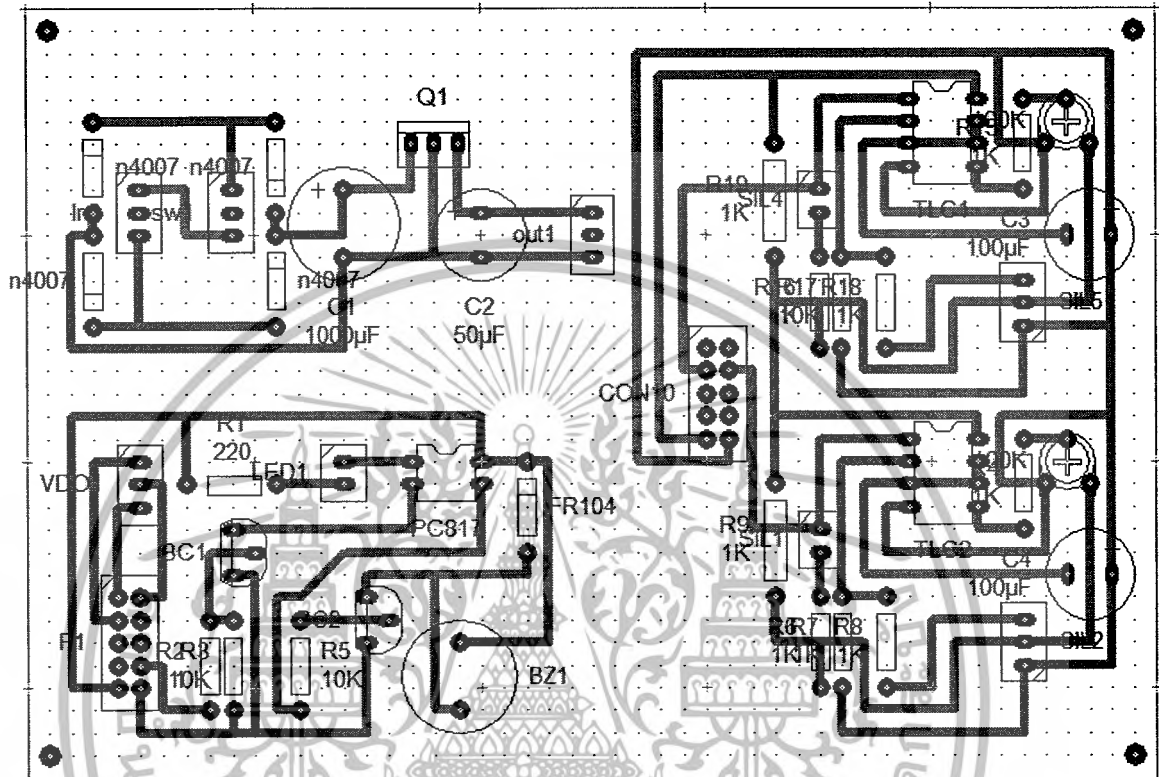
```
TextBox2.Text = ""
End Sub
```

```
End Class
```

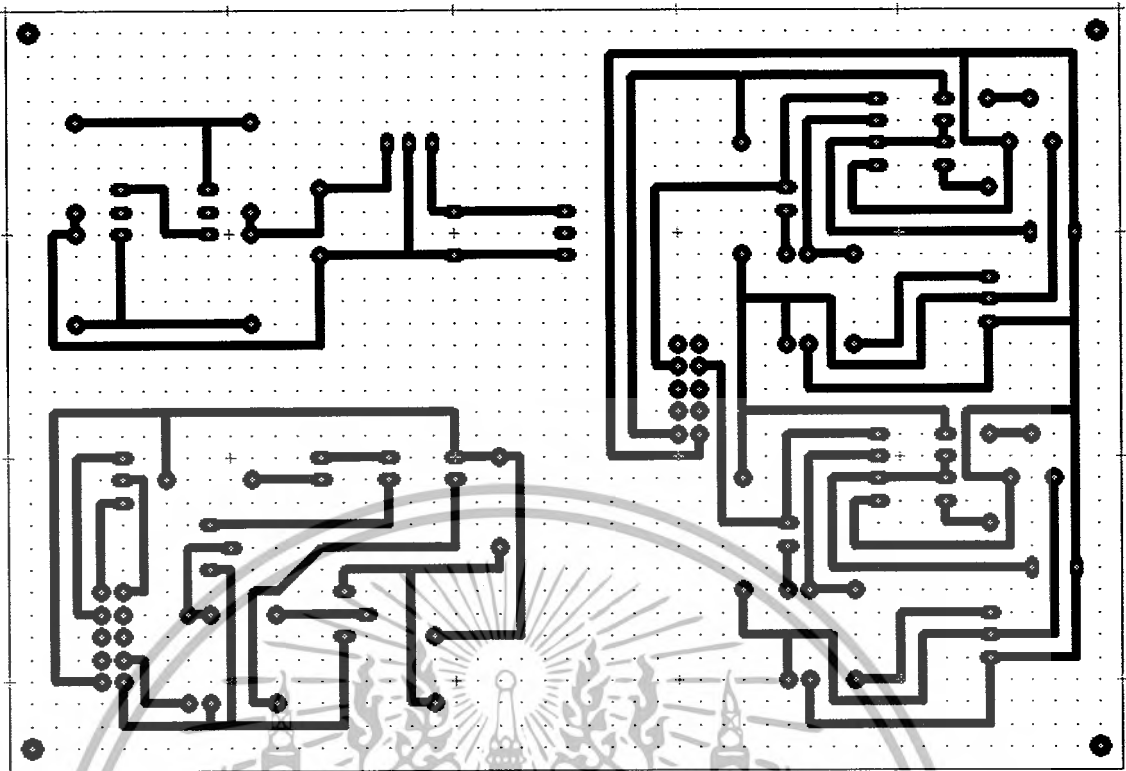


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
วงจรแผ่น PCB



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 2048 Bytes of On-chip ERAM
- 64K Bytes of On-chip Flash Memory
 - Data Retention: 10 Years at 85°C
 - Read/Write Cycle: 100K
- Boot Code Section with Independent Lock Bits
- 2K Bytes of On-chip Flash for Bootloader
- In-System Programming by On-Chip UART Boot Program and IAP Capability
- 2K Bytes of On-chip EEPROM
 - Read/Write Cycle: 100K
- Integrated Power Monitor (POR: PFD) To Supervise Internal Power Supply
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- High-speed Architecture
 - In Standard Mode:
 - 40 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
 - 60 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
 - In X2 mode (6 Clocks/machine cycle)
 - 20 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
 - 30 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
- Five Ports: 32 + 4 Digital I/O Lines
- Five-channel 16-bit PCA with
 - PWM (8-bit)
 - High-speed Output
 - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- SPI Interface (PLCC52 and VFP64 packages only)
- On-chip Emulation Logic (Enhanced Hook System)
- Power Saving Modes
 - Idle Mode
 - Power-down Mode
- Power Supply: 3 volts to 5.5 volts
- Temperature Range: Industrial (-40° to +85°C)
- Packages: VQFP44, PLCC44, VQFP64, PLCC52

Description

The AT89C51AC3 is a high performance Flash version of the 80C51 single chip 8-bit microcontrollers.

In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.

Besides the AT89C51AC3 provides 64K Bytes of Flash memory including In-System Programming (ISP) and IAP, 2K Bytes Boot Flash Memory, 2K Bytes EEPROM and 2048 byte ERAM.

Primary attention is paid to the reduction of the electro-magnetic emission of AT89C51AC3.



Enhanced 8-bit
Microcontroller
with 64KB Flash
Memory

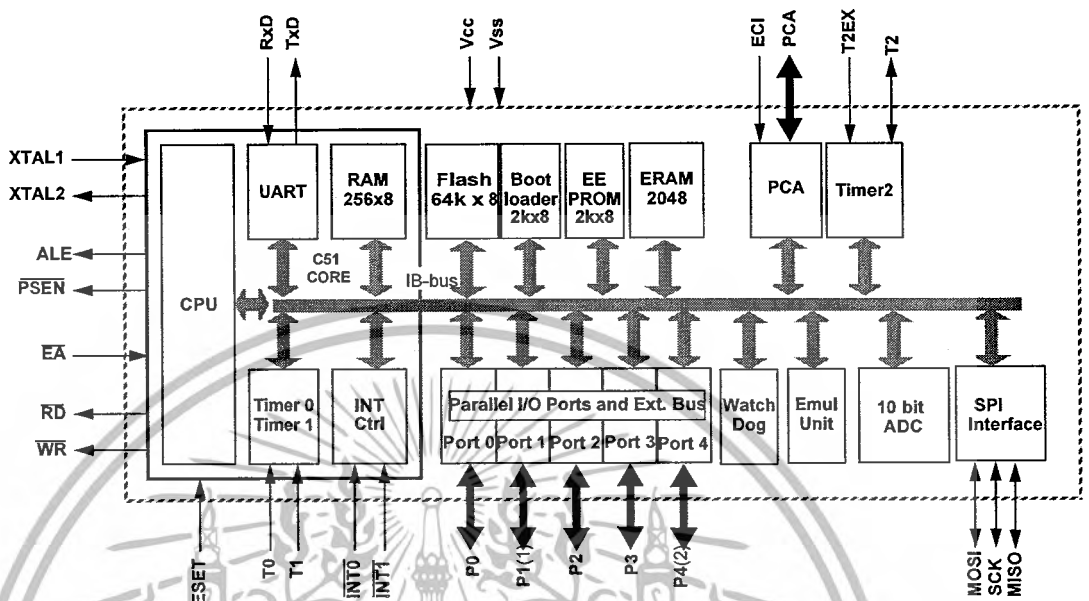
AT89C51AC3

Rev. 4383B-8051-01/05



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

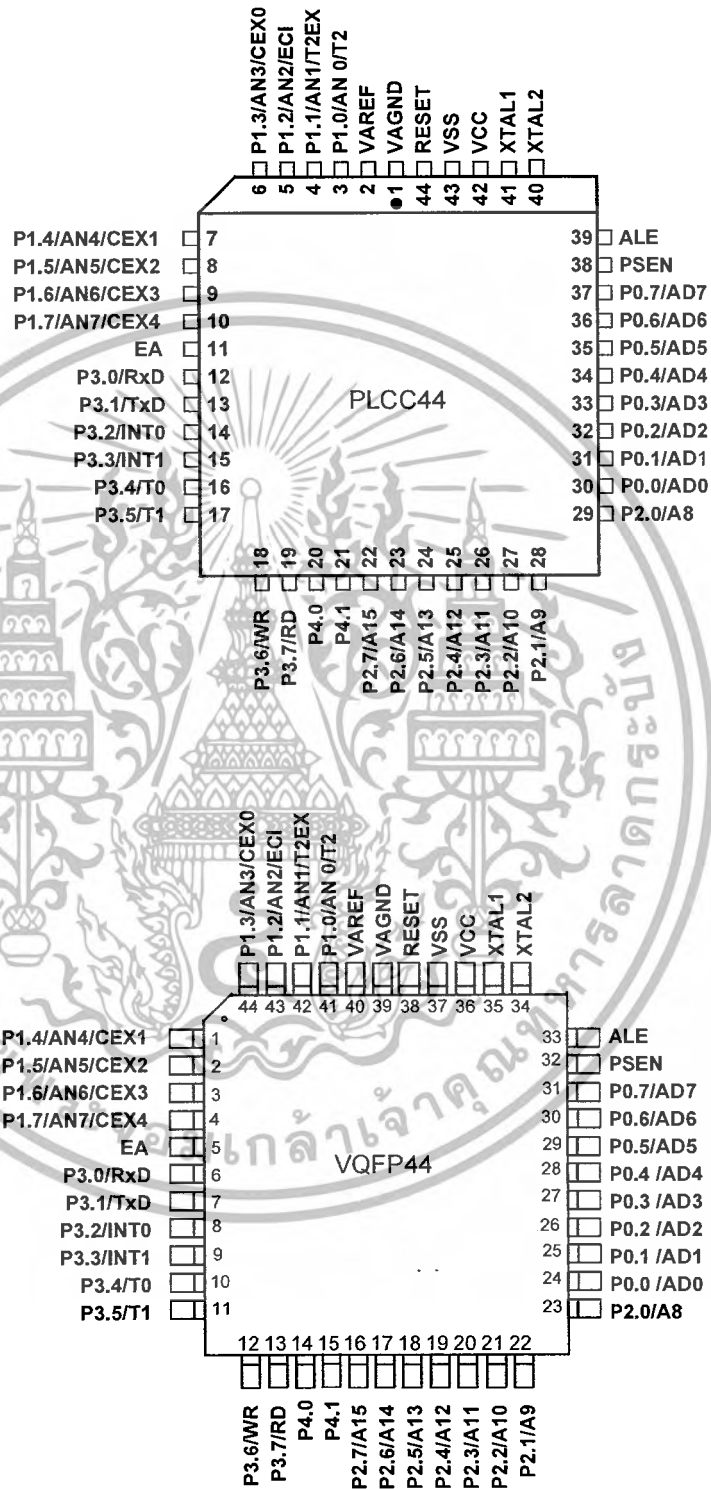
Block Diagram

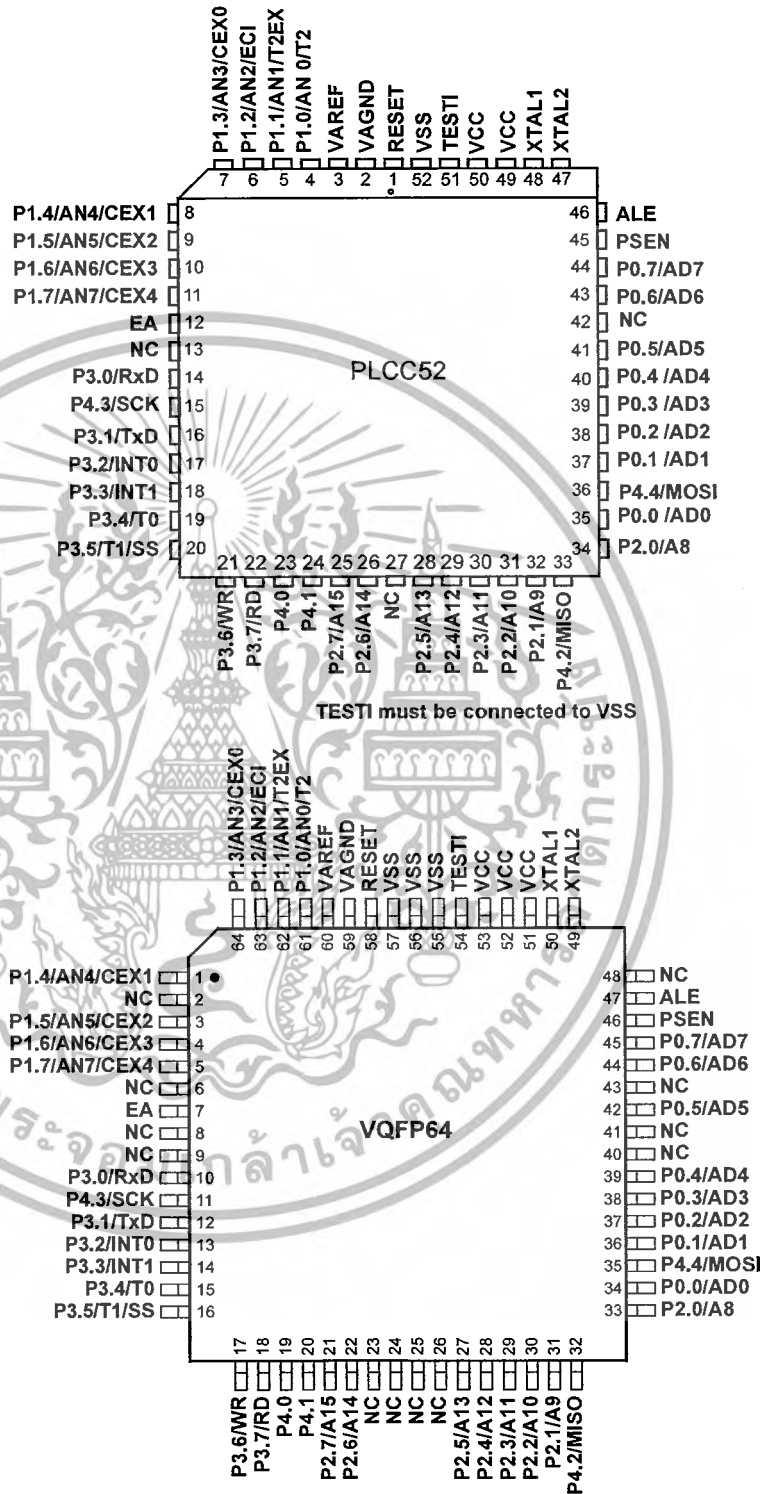


Notes: 1. 8 analog Inputs/8 Digital I/O
 2. 5-Bit I/O Port

AT89C51AC3

Pin Configuration





AT89C51AC3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4383B-8051-01/05

Pin Name	Type	Description
VSS	GND	Circuit ground
TEST1	I	Must be connected to VSS
VCC		Supply Voltage
VAREF		Reference Voltage for ADC
VAGND		Reference Ground for ADC
P0.0:7	I/O	<p>Port 0: Is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pull-ups when emitting 1's. Port 0 also outputs the code Bytes during program validation. External pull-ups are required during program verification.</p>
P1.0:7	I/O	<p>Port 1: Is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC). Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O.</p> <p>P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2.</p> <p>P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2.</p> <p>P1.2/AN2/EC1 Analog input channel 2, PCA external clock input.</p> <p>P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output.</p> <p>P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output.</p> <p>P1.5/AN5/CEX2 Analog input channel 5, PCA module 2 Entry of input/PWM output.</p> <p>P1.6/AN6/CEX3 Analog input channel 6, PCA module 3 Entry of input/PWM output.</p> <p>P1.7/AN7/CEX4 Analog input channel 7, PCA module 4 Entry of input/PWM output.</p> <p>Port 1 receives the low-order address byte during EPROM programming and program verification. It can drive CMOS inputs without external pull-ups.</p>
P2.0:7	I/O	<p>Port 2: Is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 2 emits the high-order address byte during accesses to the external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1's. During accesses to external Data Memory that use 8 bit addresses (MOVX @Ri), Port 2 transmits the contents of the P2 special function register. It also receives high-order addresses and control signals during program validation. It can drive CMOS inputs without external pull-ups.</p>



Pin Name	Type	Description
RESET	I/O	Reset: A high level on this pin during two machine cycles while the oscillator is running resets the device. An internal pull-down resistor to VSS permits power-on reset using only an external capacitor to VCC.
ALE	O	ALE: An Address Latch Enable output for latching the low byte of the address during accesses to the external memory. The ALE is activated every 1/6 oscillator periods (1/3 in X2 mode) except during an external data memory access. When instructions are executed from an internal Flash (EA = 1), ALE generation can be disabled by the software.
PSEN	O	PSEN: The Program Store Enable output is a control signal that enables the external program memory of the bus during external fetch operations. It is activated twice each machine cycle during fetches from the external program memory. However, when executing from the external program memory two activations of PSEN are skipped during each access to the external Data memory. The PSEN is not activated for internal fetches.
EA	I	EA: When External Access is held at the high level, instructions are fetched from the internal Flash. When held at the low level, AT89C51AC3 fetches all instructions from the external program memory.
XTAL1	I	XTAL1: Input of the inverting oscillator amplifier and input of the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. To operate above a frequency of 16 MHz, a duty cycle of 50% should be maintained.
XTAL2	O	XTAL2: Output from the inverting oscillator amplifier.

I/O Configurations

Each Port SFR operates via type-D latches, as illustrated in Figure 1 for Ports 3 and 4. A CPU "write to latch" signal initiates transfer of internal bus data into the type-D latch. A CPU "read latch" signal transfers the latched Q output onto the internal bus. Similarly, a "read pin" signal transfers the logical level of the Port pin. Some Port data instructions activate the "read latch" signal while others activate the "read pin" signal. Latch instructions are referred to as Read-Modify-Write instructions. Each I/O line may be independently programmed as input or output.

Port 1, Port 3 and Port 4

Figure 1 shows the structure of Ports 1 and 3, which have internal pull-ups. An external source can pull the pin low. Each Port pin can be configured either for general-purpose I/O or for its alternate input output function.

To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 1, 3 or 4). To use a pin for general-purpose input, set the bit in the Px register. This turns off the output FET drive.

To configure a pin for its alternate function, set the bit in the Px register. When the latch is set, the "alternate output function" signal controls the output level (see Figure 1). The operation of Ports 1, 3 and 4 is discussed further in the "quasi-Bidirectional Port Operation" section.





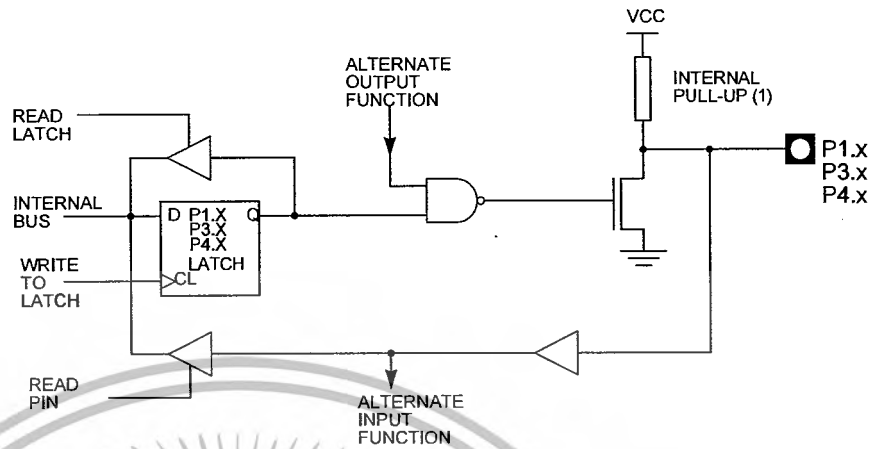
Pin Name	Type	Description
P3.0:7	I/O	<p>Port 3: Is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 3 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. The output latch corresponding to a secondary function must be programmed to one for that function to operate (except for TxD and \overline{WR}). The secondary functions are assigned to the pins of port 3 as follows:</p> <p>P3.0/\overline{RxD}: Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface</p> <p>P3.1/TxD: Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface</p> <p>P3.2/$\overline{INT0}$: External interrupt 0 input/timer 0 gate control input</p> <p>P3.3/$\overline{INT1}$: External interrupt 1 input/timer 1 gate control input</p> <p>P3.4/$T0$: Timer 0 counter input</p> <p>P3.5/$T1/\overline{SS}$: Timer 1 counter input SPI Slave Select</p> <p>P3.6/\overline{WR}: External Data Memory write strobe; latches the data byte from port 0 into the external data memory</p> <p>P3.7/\overline{RD}: External Data Memory read strobe; Enables the external data memory. It can drive CMOS inputs without external pull-ups.</p>
P4.0:4	I/O	<p>Port 4: Is an 2-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 4 pins that are being pulled low externally will be a source of current (I_{IL}, on the datasheet) because of the internal pull-up transistor. The secondary functions are assigned to the 5 pins of port 4 as follows:</p> <p>P4.0: Regular Port I/O</p> <p>P4.1: Regular Port I/O</p> <p>P4.2/\overline{MISO}: Master Input Slave Output of SPI controller</p> <p>P4.3/\overline{SCK}: Serial Clock of SPI controller</p> <p>P4.4/\overline{MOSI}: Master Output Slave Input of SPI controller</p> <p>It can drive CMOS inputs without external pull-ups.</p>

AT89C51AC3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4383B-8051-01/05

Figure 1. Port 1, Port 3 and Port 4 Structure



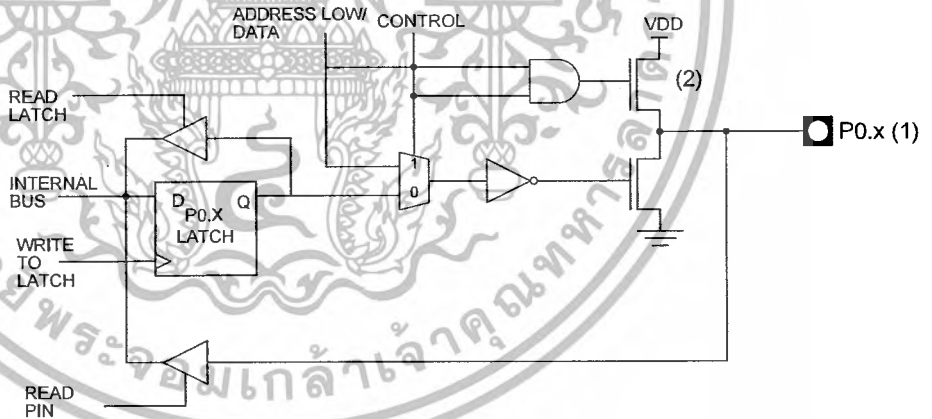
Note: The internal pull-up can be disabled on P1 when analog function is selected.

Port 0 and Port 2

Ports 0 and 2 are used for general-purpose I/O or as the external address/data bus. Port 0, shown in Figure 3, differs from the other Ports in not having internal pull-ups. Figure 3 shows the structure of Port 2. An external source can pull a Port 2 pin low.

To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 0 or 2). To use a pin for general-purpose input, set the bit in the Px register to turn off the output driver FET.

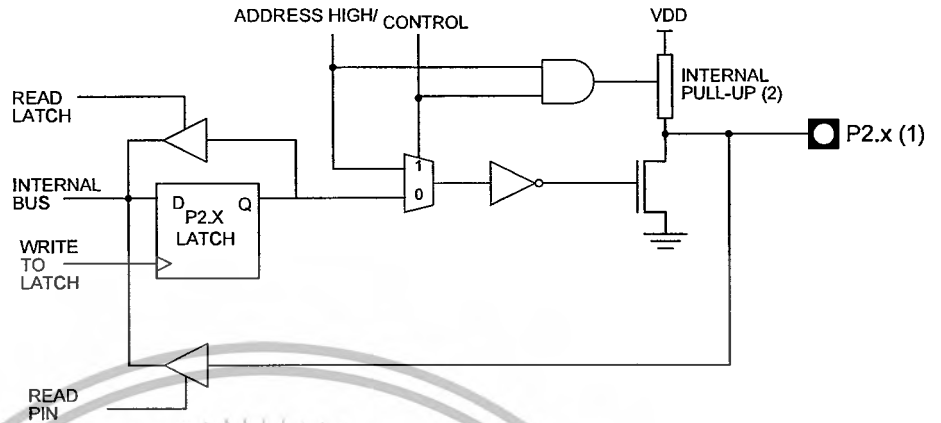
Figure 2. Port 0 Structure



- Notes:
1. Port 0 is precluded from use as general-purpose I/O Ports when used as address/data bus drivers.
 2. Port 0 internal strong pull-ups assist the logic-one output for memory bus cycles only. Except for these bus cycles, the pull-up FET is off, Port 0 outputs are open-drain.

AT89C51AC3

Figure 3. Port 2 Structure



- Notes:
1. Port 2 is precluded from use as general-purpose I/O Ports when as address/data bus drivers.
 2. Port 2 internal strong pull-ups FET (P1 in FiGURE) assist the logic-one output for memory bus cycle.

When Port 0 and Port 2 are used for an external memory cycle, an internal control signal switches the output-driver input from the latch output to the internal address/data line.

Read-Modify-Write Instructions

Some instructions read the latch data rather than the pin data. The latch based instructions read the data, modify the data and then rewrite the latch. These are called "Read-Modify-Write" instructions. Below is a complete list of these special instructions (see Table). When the destination operand is a Port or a Port bit, these instructions read the latch rather than the pin:

Instruction	Description	Example
ANL	logical AND	ANL P1, A
ORL	logical OR	ORL P2, A
XRL	logical EX-OR	XRL P3, A
JBC	jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	complement bit	CPL P3.0
INC	increment	INC P2
DEC	decrement	DEC P2
DJNZ	decrement and jump if not zero	DJNZ P3, LABEL
MOV Px.y, C	move carry bit to bit y of Port x	MOV P1.5, C
CLR Px.y	clear bit y of Port x	CLR P2.4
SET Px.y	set bit y of Port x	SET P3.3

It is not obvious the last three instructions in this list are Read-Modify-Write instructions. These instructions read the port (all 8 bits), modify the specifically addressed bit and



write the new byte back to the latch. These Read-Modify-Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor can not rise above the transistor's base-emitter junction voltage (a value lower than VIL). With a logic one written to the bit, attempts by the CPU to read the Port at the pin are misinterpreted as logic zero. A read of the latch rather than the pins returns the correct logic-one value.

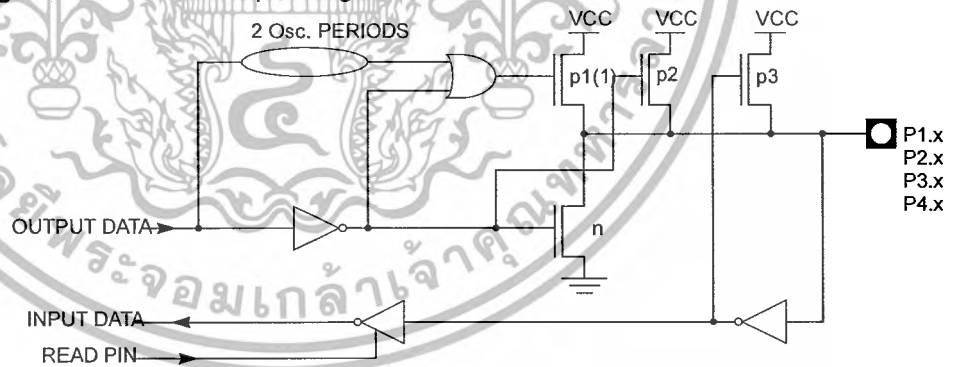
Quasi-Bidirectional Port Operation

Port 1, Port 2, Port 3 and Port 4 have fixed internal pull-ups and are referred to as "quasi-bidirectional" Ports. When configured as an input, the pin impedance appears as logic one and sources current in response to an external logic zero condition. Port 0 is a "true bidirectional" pin. The pins float when configured as input. Resets write logic one to all Port latches. If logical zero is subsequently written to a Port latch, it can be returned to input conditions by a logical one written to the latch.

Note: Port latch values change near the end of Read-Modify-Write instruction cycles. Output buffers (and therefore the pin state) update early in the instruction after Read-Modify-Write instruction cycle.

Logical zero-to-one transitions in Port 1, Port 2, Port 3 and Port 4 use an additional pull-up (p1) to aid this logic transition (see Figure 4.). This increases switch speed. This extra pull-up sources 100 times normal internal circuit current during 2 oscillator clock periods. The internal pull-ups are field-effect transistors rather than linear resistors. Pull-ups consist of three p-channel FET (pFET) devices. A pFET is on when the gate senses logical zero and off when the gate senses logical one. pFET #1 is turned on for two oscillator periods immediately after a zero-to-one transition in the Port latch. A logical one at the Port pin turns on pFET #3 (a weak pull-up) through the inverter. This inverter and pFET pair form a latch to drive logical one. pFET #2 is a very weak pull-up switched on whenever the associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3.

Figure 4. Internal Pull-Up Configurations



Note: Port 2 p1 assists the logic-one output for memory bus cycles.

FEATURES

- Analog output
- Detection Accuracy @ 80 cm: ± 10 cm
- Range: 20 to 150 cm
- Typical response time: 39 ms
- Typical start up delay: 44 ms
- Average Current Consumption: 33 mA

DESCRIPTION

The GP2Y0A02YK is a wide angle sensor.

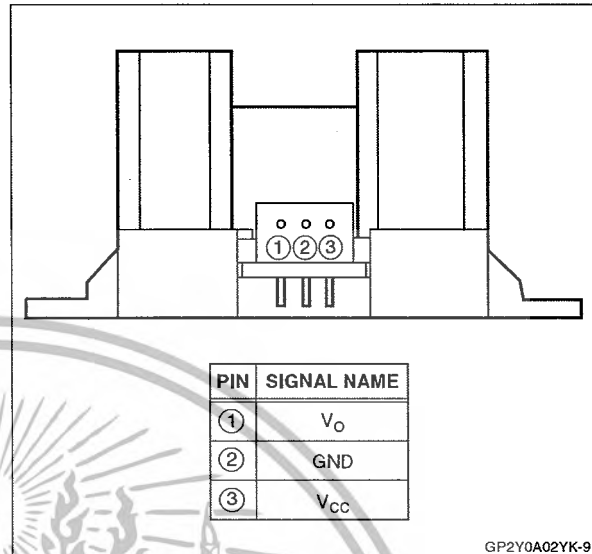


Figure 1. Pinout

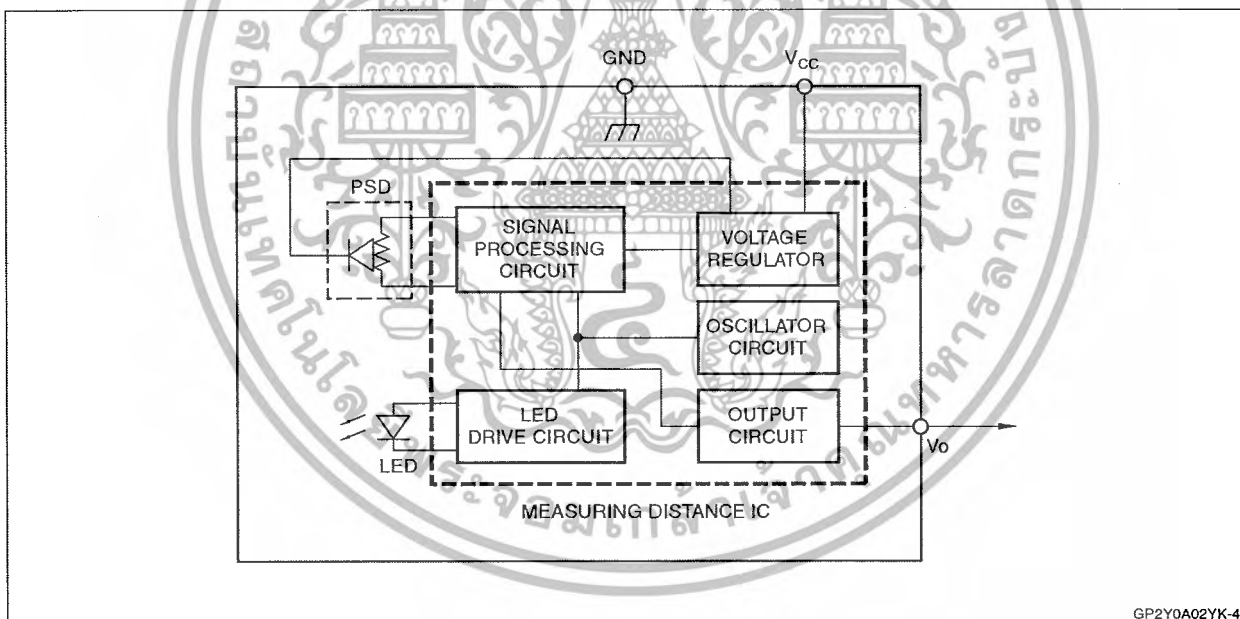


Figure 2. Block Diagram

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

Ta = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V _{CC}	-0.3 to +7.0	V
Output Terminal Voltage	V _O	-0.3 to (V _{CC} + 0.3)	V
Operating Temperature	T _{opr}	-10 to +60	°C
Storage Temperature	T _{stg}	-40 to +70	°C

Operating Supply Voltage

PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	V _{CC}	4.5 to 5.5	V

Electro-optical Characteristics

Ta = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTES
Measuring Distance Range	ΔL		20	—	150	cm	1
Output Terminal Voltage	V _O	L = 150 cm	0.25	0.4	0.55	V	1
Output Voltage Difference	ΔV _O	Output change at L change (150 cm – 20 cm)	1.8	2.05	2.3	V	1
Average Supply Current	I _{CC}	L = 150 cm	—	33	50	mA	1, 2

NOTES:

1. Measurements made with Kodak R-27 Gray Card, using the white side, (90% reflectivity).
2. L = Distance to reflective object.

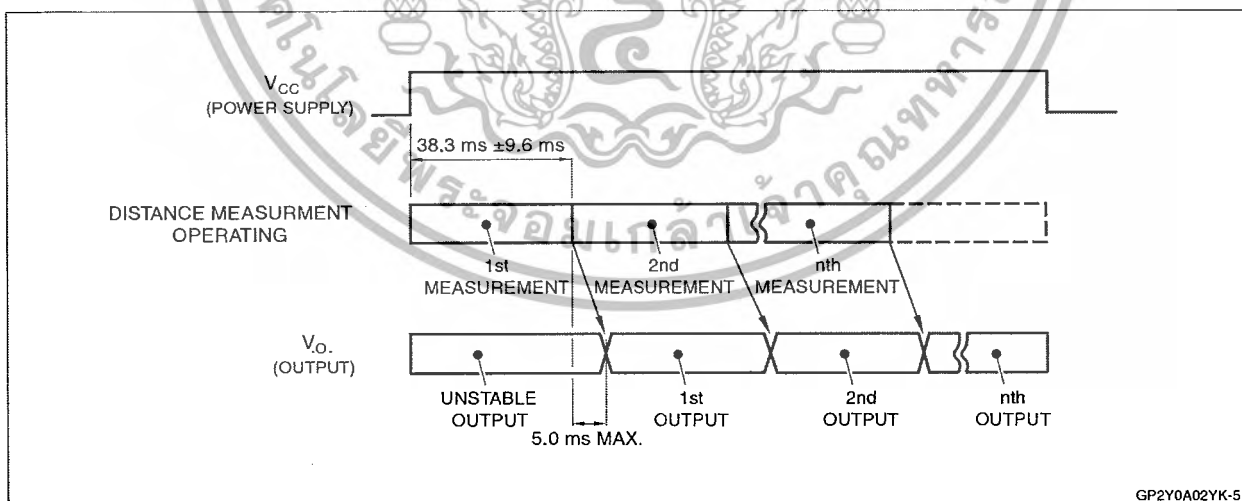


Figure 3. Timing Diagram

GP2Y0A02YK-5

RELIABILITY

The reliability requirements of this device are listed in Table 1.

Table 1. Reliability

TEST ITEMS	TEST CONDITIONS	FAILURE JUDGEMENT CRITERIA	SAMPLES (n), DEFECTIVE (C)
Temperature Cycling	One cycle -40°C (30 min.) to +70°C in 30 minutes, repeated 25 times	Initial $\times 0.8 > V_O$ $V_O > \text{Initial} \times 1.2$	n = 11, C = 0
High Temperature and High Humidity Storage	+40°C, 90% RH, 500h		n = 11, C = 0
High Temperature Storage	+70°C, 500h		n = 11, C = 0
Low Temperature Storage	-40°C, 500h		n = 11, C = 0
Operation Life (High Temperature)	+60°C, $V_{CC} = 5\text{ V}$, 500h		n = 11, C = 0
Mechanical Shock	100 m/s ² , 6.0 ms 3 times/ $\pm X$, $\pm Y$, $\pm Z$ direction		n = 6, C = 0
Variable Frequency Vibration	10-to-55-to-10 Hz in 1 minute Amplitude: 1.5 mm 2h in each X, Y, Z direction		n = 6, C = 0

NOTES:

1. Test conditions are according to Electro-optical Characteristics, shown on page 2.
2. At completion of the test, allow device to remain at nominal room temperature and humidity (non-condensing) for two hours.
3. Confidence level: 90%, Lot Tolerance Percent Defect (LTPD): 20%/40%.

MANUFACTURER'S INSPECTION

Inspection Lot

Inspection shall be carried out per each delivery lot.

Inspection Method

A single sampling plan, normal inspection level II based on ISO 2859 shall be adopted.

Table 2. Quality Level

DEFECT	INSPECTION ITEM and TEST METHOD	AQL (%)
Major Defect	Electro-optical characteristics defect	0.4
Minor Defect	Defect on appearance and dimension (crack, split, chip, scratch, stain)*	1.0

NOTE: *Any one of these that affects the Electro-optical Characteristics shall be considered a defect.

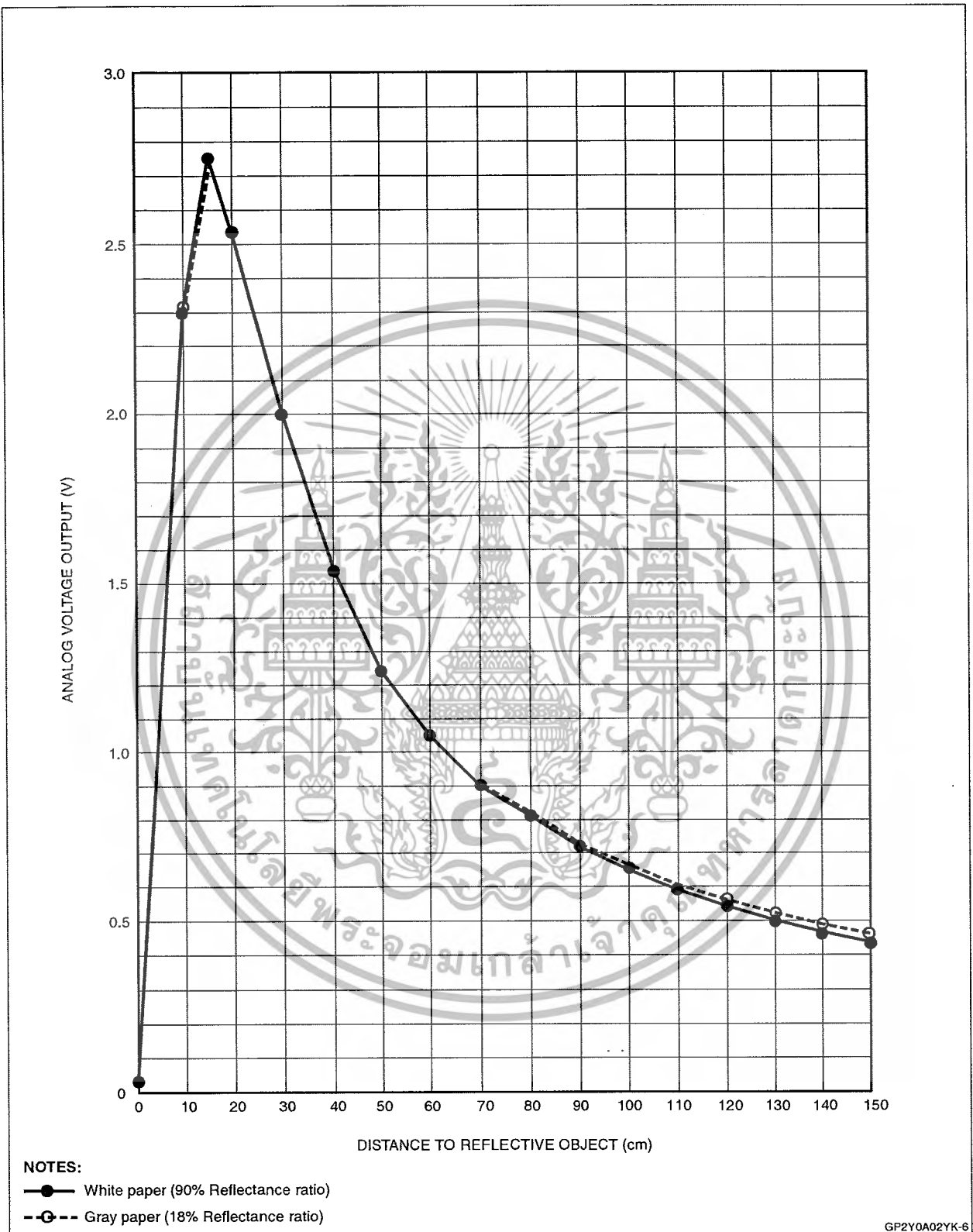
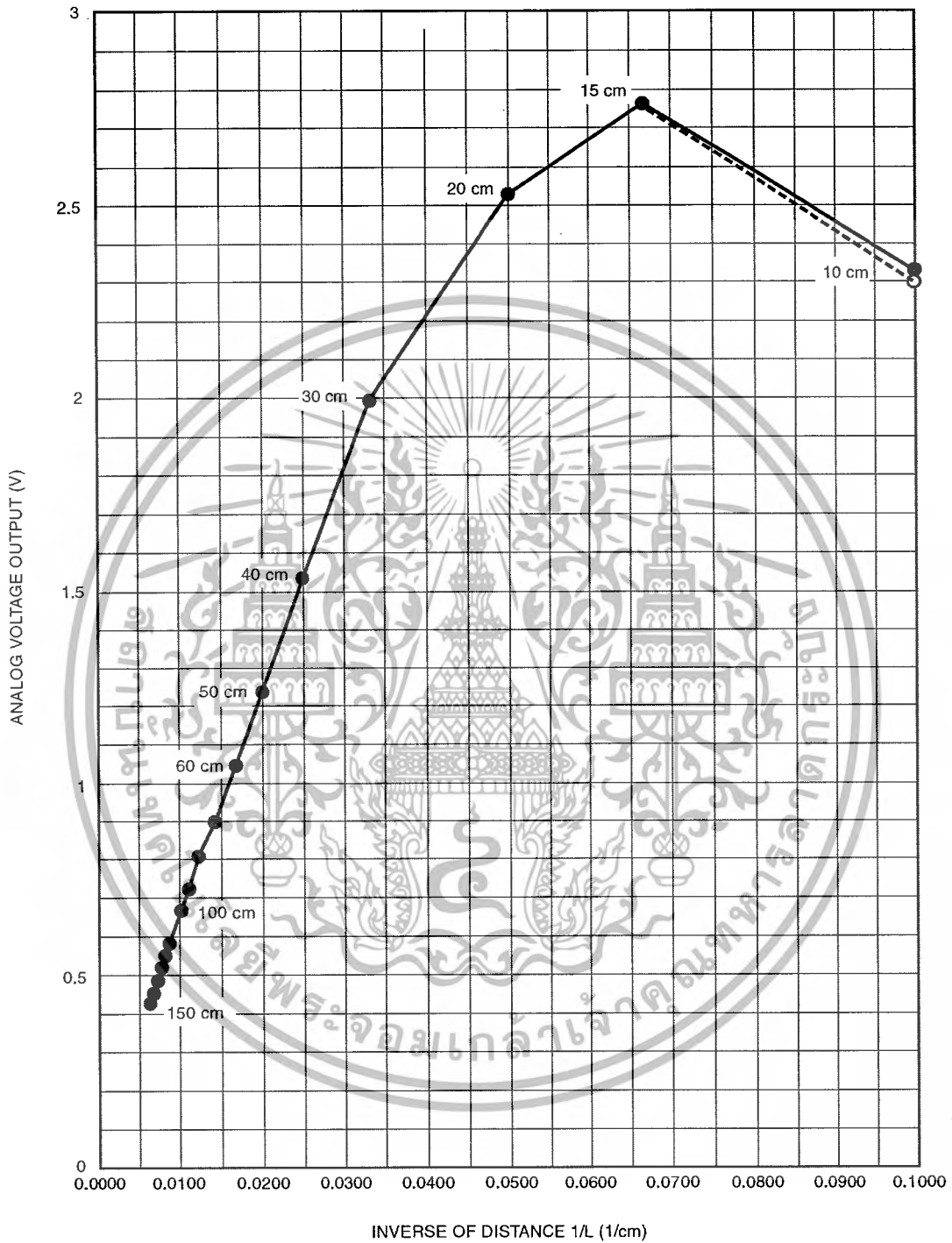


Figure 4. GP2Y0A02YK Example of Output Distance Characteristics



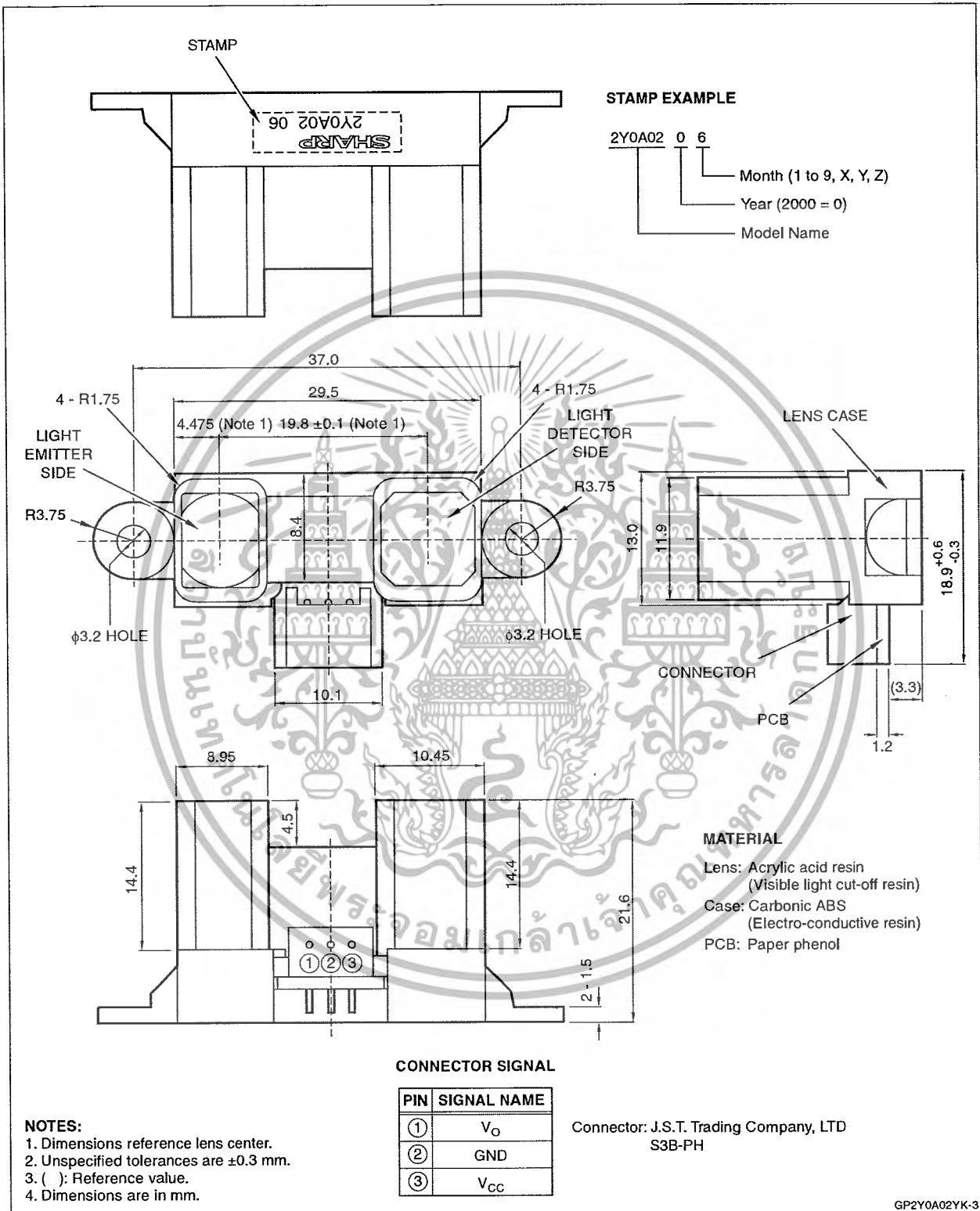
NOTES:

- White paper (90% Reflectance ratio)
- Gray paper (18% Reflectance ratio)

GP2Y0A02YK-7

Figure 5. GP2Y0A02YK Example of Output Characteristics with Inverse Number of Distance

PACKAGE SPECIFICATIONS



GP2Y0A02YK-3

PACKING SPECIFICATION

PART NAME	MATERIAL
Packing case	Corrugated cardboard
Pad	Corrugated cardboard
Tray	Polystyrene

PACKING METHOD

- Each tray holds 50 pieces. Packing methods are shown in (A).
- Each box holds 10 trays. Pads are added to top and bottom, and between layers, as in (B).
top and bottom. Put pads between each tray (9 pads total) see above drawing (B).
- The box is sealed with packing tape. (C) shows the location of the Model number, Quantity, and Inspection date.
- Package weight: Approximately 4 kg.

The diagram illustrates the following components and steps:

- (A)** Tray: A polystyrene tray with a grid pattern, used to hold individual products.
- (B)** Packing Case: A corrugated cardboard box containing 10 trays. It is padded with corrugated cardboard sheets (2 sheets on top and bottom, 9 sheets between trays) and sealed with packing tape.
- (C)** Packing Case: The final sealed box with labels for Model Number, Quantity, and Date.

GP2Y0A02YK-8

NOTES

- Keep the sensor lens clean. Dust, water, oil, and other contaminants can deteriorate the characteristics of this device. Applications should be designed to eliminate sources of lens contamination.
- When using a protective cover over the emitter and detector, ensure the cover efficiently transmits light throughout the wavelength range of the LED ($\lambda = 850 \text{ nm} \pm 70 \text{ nm}$). Both sides of the protective cover should be highly polished. Use of a protective cover may decrease the effective distance over which the sensor operates. Ensure that any cover does not negatively affect the operation over the intended application range.
- Objects in proximity to the sensor may cause reflections that can affect the operation of the sensor.
- Sources of high ambient light (the sun or strong artificial light) may affect measurement. For best results, the application should be designed to prevent interference from direct sunlight or artificial light.
- Using the sensor with a mirror can induce measurement errors. Often, changing the incident angle on the mirror can correct this problem.
- If a prominent boundary line exists in the surface being measured, it should be aligned vertically to avoid measurement error. See Figure 5 for further details.
- When measuring the distance to objects in motion, align the sensor so that the motion is in the horizontal direction instead of vertical. Figure 6 illustrates the preferred alignment.
- A 10 μF (or larger) bypass capacitor between V_{CC} and GND near the sensor is recommended.
- To clean the sensor, use a dry cloth. Use of any liquid to clean the device may result in decreased sensitivity or complete failure.
- Excessive mechanical stress can damage the internal sensor or lens.

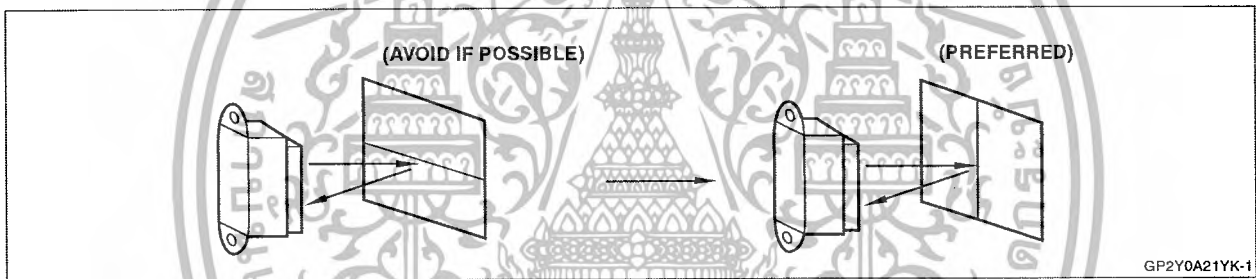


Figure 6. Proper Alignment to Surface Being Measured

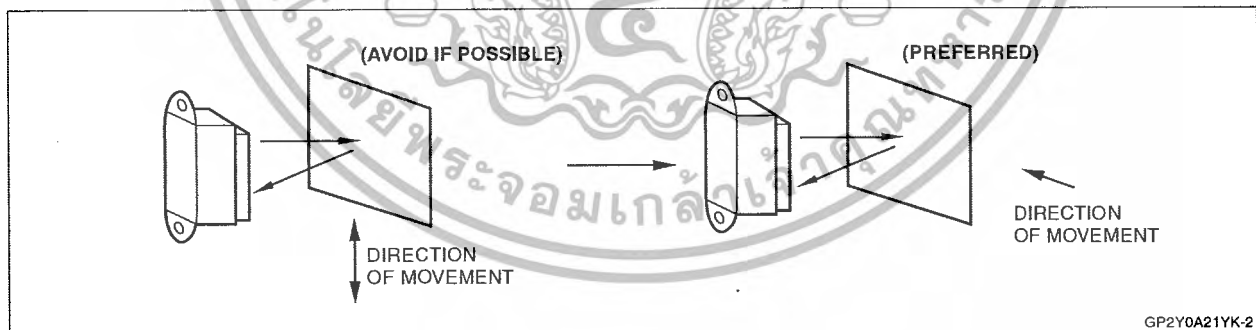


Figure 7. Proper Alignment to Moving Surfaces

NOTICE

The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property right. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP devices.

SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structures and other contents described herein at any time without notice in order to improve design or reliability.

Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. Manufacturing locations are also subject to change without notice.

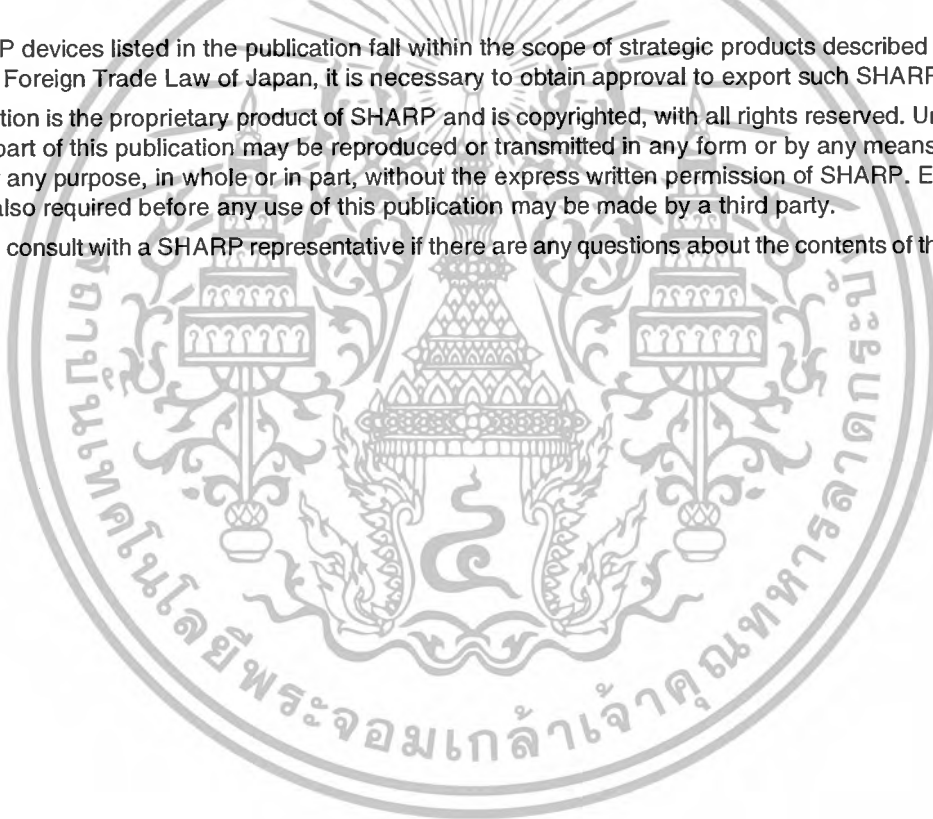
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any SHARP devices shown in catalogs, data books, etc.

The devices listed in this publication are designed for standard applications for use in general electronic equipment. SHARP's devices shall not be used for or in connection with equipment that requires an extremely high level of reliability, such as military and aerospace applications, telecommunication equipment (trunk lines), nuclear power control equipment and medical or other life support equipment (e.g. Scuba). SHARP takes no responsibility for damage caused by improper use of device, which does not meet the conditions for use specified in the relevant specification sheet.

If the SHARP devices listed in the publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

Contact and consult with a SHARP representative if there are any questions about the contents of this publication.



SHARP

SHARP CORPORATION
SALES & MARKETING GROUP
ELECTRONIC COMPONENTS & DEVICES
22-22 NAGAIKE-CHO, ABENO-KU, OSAKA 545-8522, JAPAN
PHONE: (81) 6-6621-1221
FAX: (81) 6117-725300, 6117-725301, 6117-725302
<http://sharp-world.com/products/device>

Specifications are subject to change without notice.

Countries and Areas

U.S.A.

SHARP MICROELECTRONICS
OF THE AMERICAS

North American Head Office
5700 North West Pacific Rim Boulevard
Camas, Washington 98607 USA
PHONE: (1)360-834-2500
FAX: (1)360-834-8903
<http://www.sharpsma.com>

Western Area

1980 Zanker Road, San Jose, CA 95112
PHONE: (1)408-436-4900
FAX: (1)408-436-0924

5901 Bolsa Ave.
Huntington Beach, CA 92647-2053
PHONE: (1)714-903-4600
FAX: (1)714-903-0295

6390 Greenwich Drive, S uite 175
San Diego, CA 92122
PHONE: (1)858-597-0982
FAX: (1)858-597-8701

Central Area

85 W. Algonquin Road, Suite 280
Arlington Heights, IL 60005
PHONE: (1)847-258-2750
FAX: (1)847-439-2479

6303 Commerce Drive, Suite 175 Irving, TX 75063
PHONE: (1)972-582-1710
FAX: (1)972-580-7537

8911 Capitol of Texas Hwy. Suite 3130
Austin, TX 78759
PHONE: (1)512-349-7262
FAX: (1)512-349-7002

20333 State Hwy. 249, Suite 200 Houston, TX 77070
PHONE: (1)281-378-1520
FAX: (1)281-378-1521

W129 S 9647 Tony Lema Lane Muskego, WI 53150
PHONE: (1)414-529-9568
FAX: (1)414-529-9569

3001 West Big Beaver Road, Suite 722
Troy, MI 48084
PHONE: (1)248-458-1527
FAX: (1)248-458-6255

Eastern Area

1070 N. Kimbles Road, Yardley, PA 19067
PHONE: (1)215-321-5530
FAX: (1)215-321-5534

200 Wheeler Rd., Burlington, MA 01803
PHONE: (1)781-270-7979; (1)781-229-5100
FAX: (1)781-229-9117

8000 Regency Parkway, Suite 280 Cary, NC 27511
PHONE: (1)919-460-0695
FAX: (1)919-460-0795

2321 Sidney St. Pittsburgh, PA 15203
PHONE: (1)412-381-1191
FAX: (1)412-381-1192

4875 North Federal Highway, Third Floor
Ft. Lauderdale, FL 33318
PHONE: (1)954-267-8883
FAX: (1)954-267-0254

EUROPE

SHARP MICROELECTRONICS EUROPE
A division of Sharp Electronics (Europe) GmbH

Head Office
Sonnenstrasse 3, 20097, Hamburg, Germany
PHONE: (49)180-5073507
FAX: (49)40-2376-2232
<http://www.sharpsme.com/>

Germany

SME München Office
Fuerstenniederstrasse 5, 80687 München, Germany
PHONE: (49)89-5468420
FAX: (49)89-54 684250

France

SME Paris Office
1 Rue Raoul Follereau Bussy Saint Georges
77608 Marne la Vallée Cedex 3
PHONE: (33)1 6476 22 22
FAX: (33)1 6476 22 23

Italy

SME Milano Office
Centro Direzionale Colleoni
Palazzo Taurus Ingresso 2
20041 Agrate Brianza, Milano, Italy
PHONE: (39)39-68 99 946
FAX: (39)39-68 99 948

U.K.

SME London Office
Centennial Court, Easthampstead Road,
Bracknell, Berkshire R G12 1YQ, United Kingdom
PHONE: (44)1344-86 99 22
FAX: (44)1344-36 09 03

Ireland

SME Dublin Office
First Floor, Block 1, St. Johns Court, Santry,
Dublin 9, Ireland
PHONE: (353)1-842 87 05
FAX: (353)1-842 84 55

ASIA

SHARP ELECTRONICS (SHANGHAI) CO., LTD.
Microelectronics Sales & Marketing Division
16F, King Tower, 28 Xin Jin Qiao Road,
Pudong DIST, Shanghai 201206 P.R. China
PHONE: (86)21-5854-7710/21-5834-6056
FAX: (86)21-5854-4340/21-5834-6057

Registered Address

No. 11, De Bao Road, Xin Development BLDG
46 Wai Gao Qiao Free Trade Zone, Shanghai
200131, P.R. China

Beijing Office

Room 1062, Beijing Jing An Center No. 8 East
Bei San Huan Road, Chao Yang DIST, Beijing
100028 P.R. China
PHONE: (86) 10-6466-7543/10-6466-6561
FAX: (86) 10-6468-8920
<http://sharp-world.com/products/devicechina/index.html>

SHARP-ROXY (HONG KONG) LTD.

Device Sales Division, 17/F, Admiralty Centre,
Tower 1, 18 Harcourt Road, Hong Kong
PHONE: (852)28229311
FAX: (852)28660779
<http://www.sharp.com.hk>

Shenzhen Representative Office

Room 13B1, Tower C, Electronics Science &
Technology Building, Shen Nan Zhong Road,
Shenzhen, P.R. China
PHONE: (86)755-83273731
FAX: (86)755-83273735

SHARP ELECTRONIC COMPONENTS

(TAIWAN) CORPORATION
8F-A, No. 16, Sec. 4, Nanking E. R. d., Taipei, Taiwan
PHONE: (886)2-2577-7341
FAX: (886)2-2577-7326/2-2577-7328

SHARPELECTRONICS (SINGAPORE) PTE., LTD.

396 Alexandra Road #07-00
BP Tower Singapore 119954
PHONE: (65) 62713566
FAX: (65) 62713855
<http://www.seel-sharp.com>

SHARP MICROELECTRONICS

TECHNOLOGY (M) SDN BHD.
Suite E 408, 4th Floor, East Tower,
Wisma Consplant 1, No. 2 J In. SS 16/4,
Subng Jaya, 47500, Selangor Darul Ehsan, Malaysia
PHONE: (60) 3-5637-8964
FAX: (60) 3-5638-4029

SHARP ELECTRONIC COMPONENTS

(KOREA) CORPORATION
RM 501 iLsin B/D. 541, Dohwa-dong,
Mapo-ku, Seoul, Korea, 121-701
PHONE: (82)2-711-5813 ~ 8
FAX: (82)2-711-5819

Distributed By

©2006 by SHARP Corporation

Reference Code SMA06009

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Microtips Technology Inc.

台北縣汐止鎮康寧街169巷31號12樓
12F, No 31, Lane 169, Kang Ning St.
Hsi-Chih, Taipei Hsien, Taiwan, R.O.C.

WEB : <http://www.microtips.com.tw>
E-mail : Macros@microtips.com.tw
TEL: 886-2-2695-3133 FAX: 886-2-2695-8625

LCD MODULE SPECIFICATION FOR CUSTOMER'S APPROVAL

CUSTOMER : Standard

MODULE TYPE : MTC-S16205DEFGHSAY

APPROVED BY: (FOR CUSTOMER USE ONLY)



Approved By	Checked By	Prepared By	MT File No	Date Issued

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTENTS

ITEM	PAGE
FEATURES	3
LCD MODULE DRAWING	4
GENERAL SPECIFICATION	5
ABSOLUTE MAXIMUM RATING	5
ELECTRICAL CHARACTERISTICS	6
OPTICAL CHARACTERISTICS	6
MECHANICAL SPECIFICATION	7
INTERFACE PIN ASSIGNMENT	8
BLOCK DIAGRAM	11
POWER SUPPLY DIAGRAM	11
TIMING CHARACTERISTICS	12
READ/WRITE TIMING CHART	13
COMMANDS	14
COMMANDS DESCRIPTION	15
DD RAM ADDRESSING	19
CG RAM MAPPING	20
CHARACTER FONT TABLE	21
RELIABILITY TEST	22
APPEARANCE CHECK	22
HANDLING PRECAUTIONS	23
LCD PRODUCT QUALITY STANDARD	24
REVISION HISTORY	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES of

LIQUID CRYSTAL DISPLAY MODULE

MODEL NO. : MTC-S16205DFGHSAY

Character Type Dot Matrix LCD Module

Easy interface of 4-bit or 8-bit

Display character pattern : 5 × 7 font with cursor(208 kinds)

The special character pattern can be programmable by Character Generator RAM directly.

A customer character pattern can be programmable by mask option.

Automatically power on reset.

Internal Memory:

- Character Generator ROM (CGROM) : 10,080 bits (204 characters × 5 × 8 dot)
- Character Generator RAM (CGRAM) : 64 × 8 bits (8 characters × 5 × 8 dot)

Low power operation:

- Power supply voltage range : 2.7 ~ 5.5 V (V_{DD})
- LCD drive voltage range : 3.0 ~ 13.0 V ($V_{DD} - V_o$)

Low power consumption

CMOS process

Duty-cycle : 1/16

View Direction	<input checked="" type="checkbox"/> 6 O'clock		<input type="checkbox"/> 12 O'clock	
LCD Type	<input type="checkbox"/> FSTN Positive		<input type="checkbox"/> FSTN Negative	
	<input checked="" type="checkbox"/> STN Gray	<input type="checkbox"/> STN Yellow Green	<input type="checkbox"/> STN Blue	
Rear Polarizer	<input type="checkbox"/> Reflective	<input checked="" type="checkbox"/> Transflective	<input type="checkbox"/> Transmissive	
Backlight Type	<input checked="" type="checkbox"/> LED	<input checked="" type="checkbox"/> Internal Power	<input type="checkbox"/> EL	<input type="checkbox"/> 5V input
		<input type="checkbox"/> External Power	<input type="checkbox"/> CCFL	<input type="checkbox"/> 12V input
Backlight Color	<input type="checkbox"/> White	<input type="checkbox"/> Amber	<input type="checkbox"/> Blue Green	<input checked="" type="checkbox"/> Yellow Green
			<input type="checkbox"/> Other	
Temperature Range	<input type="checkbox"/> Normal		<input checked="" type="checkbox"/> Wide	<input type="checkbox"/> Super Wide
CCFL Inverter	<input type="checkbox"/> Build-in		<input checked="" type="checkbox"/> Not Build-in	
EL Driver IC	<input type="checkbox"/> Build-in		<input checked="" type="checkbox"/> Not Build-in	
Touch Screen	<input type="checkbox"/> With		<input checked="" type="checkbox"/> Without	

TO BE VERY CAREFUL !

The LCD driver ICs are made of CMOS process, which is very easy to be damaged by static charge, make sure the user is grounded when handling the LCM.

GENERAL SPECIFICATION

Item	Content
Display Resolution	16 Charactersx 2 lines
Dimensional Outline(mm)	80.0(W)x 36.0(H)x 12.9 max(D)
Display mode	Transflective/ Positive Type
Circuit	Controller IC, Segment-driver IC
Interface	Data (DB0~DB7), RS, R/W, E

ABSOLUTE MAXIMUM RATING

(1) Electrical Absolute Ratings

Item	Symbol	Min.	Max.	Unit	Note
Power Supply for Logic	$V_{DD}-V_{SS}$	-0.3	7.0	Volt	
Power Supply for LCD	$V_{DD}-V_O$	0	15.0	Volt	
Input Voltage	V_{IN}	-0.3	$V_{DD}+0.3$	Volt	
Supply Current for LED Backlight	I_{LED}	-	240	MA	

Note: Operator should be grounded during handling LCM.

(2) Environmental Absolute Maximum Ratings

Item	Normal Temperature				Wide Temperature			
	Operating		Storage		Operating		Storage	
	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
Ambient Temperature	0°C	+50°C	-20°C	+70°C	-20°C	+70°C	-30°C	+80°C
Humidity(without condensation)	Note 2,4		Note 3,5		Note 4,5		Note 4,6	

Note 2 $T_a \leq 50^\circ\text{C}$: 80% RH max

$T_a > 50^\circ\text{C}$: Absolute humidity must be lower than the humidity of 85%RH at 50°C

Note 3 T_a at -20°C will be <48hrs at 70°C will be <120hrs when humidity is higher than 70%.

Note 4 Background color changes slightly depending on ambient temperature. This phenomenon is reversible.

Note 5 $T_a \leq 70^\circ\text{C}$: 75RH max

$T_a > 70^\circ\text{C}$: absolute humidity must be lower than the humidity of 75%RH at 70°C

Note 6 T_a at -30°C will be <48hrs, at 80°C will be <120hrs when humidity is higher than 70%.

ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	note
Power Supply for Logic	$V_{DD}-V_{SS}$	-	2.7	4.5	5.5	Volt	
Input Voltage	V_{IL}	L level	V_{SS}	$0.2 V_{DD}$	-	Volt	
	V_{IH}	H level	$0.8 V_{DD}$	V_{DD}	-	Volt	
LCM Recommend LCD Module Driving Voltage	$V_{DD}=5.0V$	$T_a=-20^{\circ}C$	4.1	4.4	4.7	Volt	
		$T_a=25^{\circ}C$	3.9	4.25	4.6		
		$T_a=70^{\circ}C$	3.65	3.95	4.2		
Power Supply Current for LCM	I_{DD} (B/L OFF)	$V_{DD}=5.0V$ $V_{DD}-V_o=4.2V$ $T_a=25^{\circ}C$	-	1.5	3.0	mA	
	I_{LED}	$V_{LED}=4.2V$	-	110	180		

OPTICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	note
Viewing angle range	Φ_f (12 o'clock)	When $Cr \geq 2$	-	15	-	Degree	9,10
	Φ_b (6 o'clock)		-	30	-		
	Φ_l (9 o'clock)		-	30	-		
	Φ_r (3 o'clock)		-	30	-		
Rise Time	T_r	$V_{DD}-V_o=4.2V$ $T_a=25^{\circ}C$	-	155	310	mS	
Fall Time	T_f		-	140	280		
Contrast	Cr		-	5	-		7

MECHANICAL SPECIFICATION

Product No.		MTC-S16205DFGHSAY
Module Size		85.0(W)mmx 36.0(H)mmx 12.9(D)mm Max
Display Format		16 charactersx 2 lines
Character Size		3.0 mmx 5.23 mm
Character Pitch		3.51 mmx 5.75 mm
Dot Size		0.56(W)mmx 0.61(H)mm
Dot Pitch		0.61(W)mmx 0.66(H)mm
Duty Ratio		1/16 Duty
Character Font Table		<input checked="" type="checkbox"/> English <input type="checkbox"/> European <input type="checkbox"/> Other_____
LCD Display Mode	STN	<input checked="" type="checkbox"/> Gray Mode <input type="checkbox"/> Yellow Mode <input type="checkbox"/> Blue Mode
	FSTN	<input type="checkbox"/> Black & White(Normally White/Positive Image) <input type="checkbox"/> Black & White(Normally White/Negative Image)
	Rear Polarizer:	<input type="checkbox"/> Reflective <input checked="" type="checkbox"/> Transflective <input type="checkbox"/> Transmissive
Viewing Direction		<input checked="" type="checkbox"/> 6 O'clock <input type="checkbox"/> 12 O'clock <input type="checkbox"/> 3 O'clock <input type="checkbox"/> 9 O'clock
Backlight		<input type="checkbox"/> Without <input type="checkbox"/> CCFL <input type="checkbox"/> EL <input checked="" type="checkbox"/> LED
Controller		KS0066 or compatible
DC/DC Converter		Without
CCFL Inverter		Without
EL Driver		Without

INTERFACE PIN ASSIGNMENT

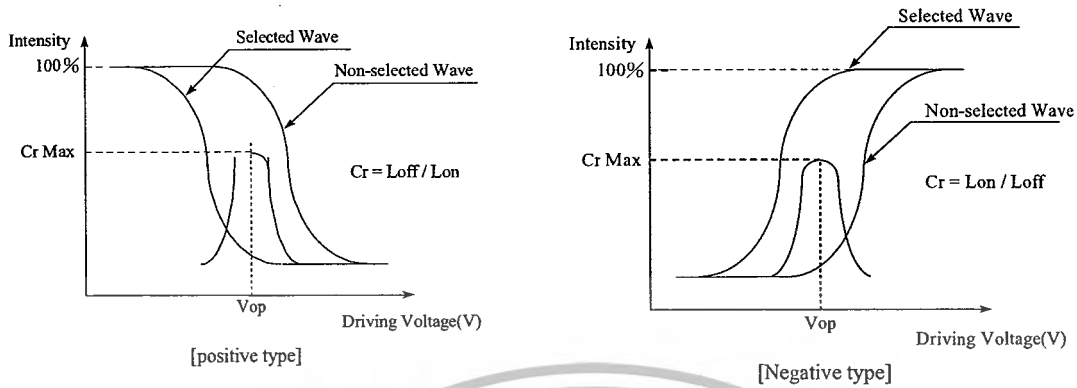
PIN NO.	PIN OUT	DESCRIPTION
1	V _{SS}	Ground
2	V _{DD}	Logic Circuit Power Supply
3	V _O	Power Supply For LCD Panel
4	RS	Data/ Instruction Register Select
5	R/W	Read/ Write Select
6	E	Enable Signal
7	DB0	3-State I/O Data Bus
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	BKL1	Power Supply for Backlight. See JUMPER EXPLANATION below. 100V/400Hz AC for EL, 4.2V or 120~180mA DC for LED backlight
16	BKL2	Don't care if no backlight

JUMPER EXPLANATION

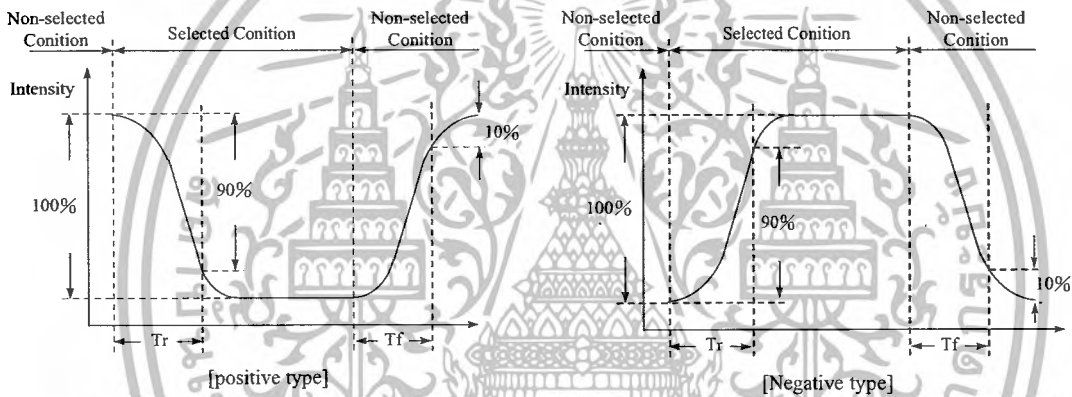
1. JP2 and JP3 are both short: Pin15 is short with pin2 and pin16 is short with pin1. Backlight powered up via V_{DD} (pin2) and V_{SS} (pin1). Customer does not need to supply power to pin15 and pin16. This is the Standard setting.
2. JP2 short and JP3 open: Pin 15 is short with Pin2 and pin16 is NOT short with pin1. Backlight is powered with V_{DD} (same as logic circuit) and customer should keep pin 15 floating or connected to V_{DD}. Customer could control the backlight independently ON or OFF with pin16 L or H.
3. JP2 open and JP3 short: Pin 15 is NOT short with pin 2 and pin16 is short with pin1. Backlight is common VSS with logic and customer should keep pin16 floating or connected to GND (V_{SS}). Customer could control the backlight ON or OFF with pin15 H or L
4. JP2 and JP3 are both open: The backlight is fully independent with the logic, control the backlight via pin15/pin16 or A/K. Keep the current consumption for LED backlight follow the spec.
5. J1 short: Bezel and screw holes connected to GND. J1 open: Bezel and screw holes floating
6. Never change the J2 and J3, it may burn off your system.

Note: J1, JP2 and JP3 are #0805 0 ohm resistors on the rear side of the PCB.

[Note 7] Definition of Operation Voltage (Vop)



[Note 8] Definition of Response Time (Tr, Tf)



Conditions:

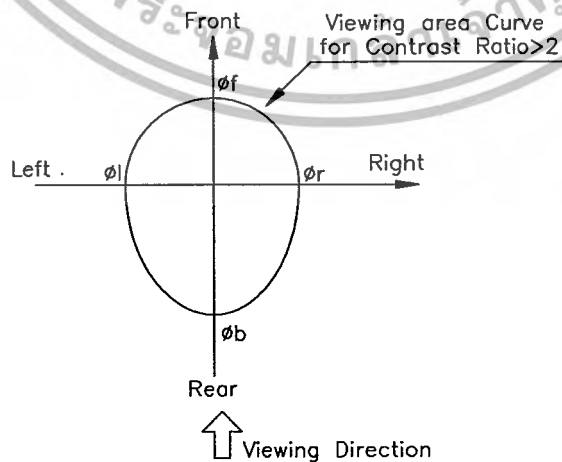
Operating Voltage : Vop

Viewing Angle (θ, φ): $0^\circ, 0^\circ$

Frame Frequency : 64 Hz

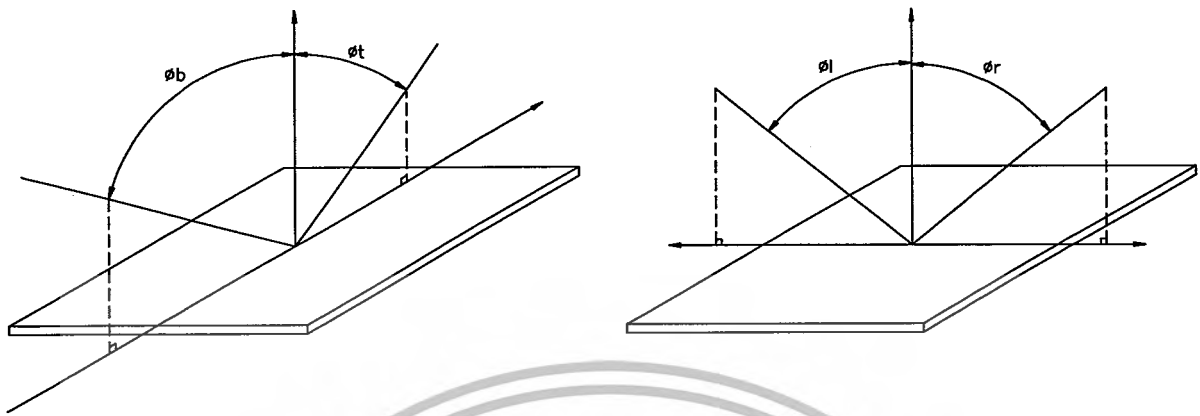
Driving Wave form : 1/N duty, 1/a bias

[Note 9] Definition of Viewing Direction

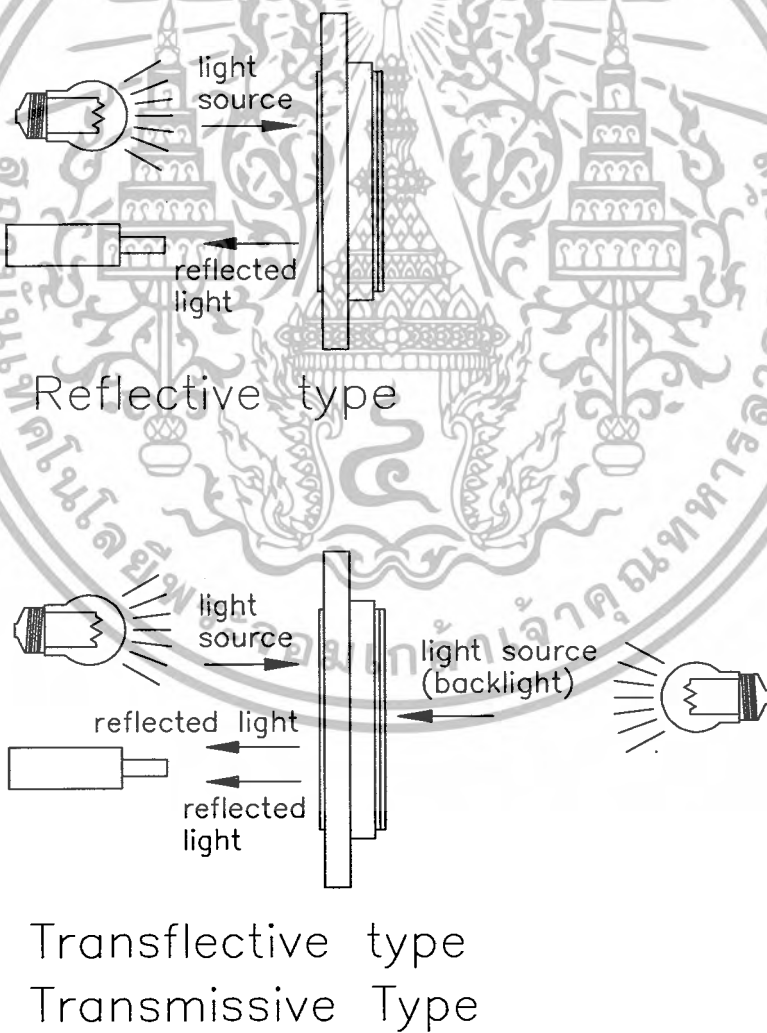


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Note 10] Definition of viewing angle

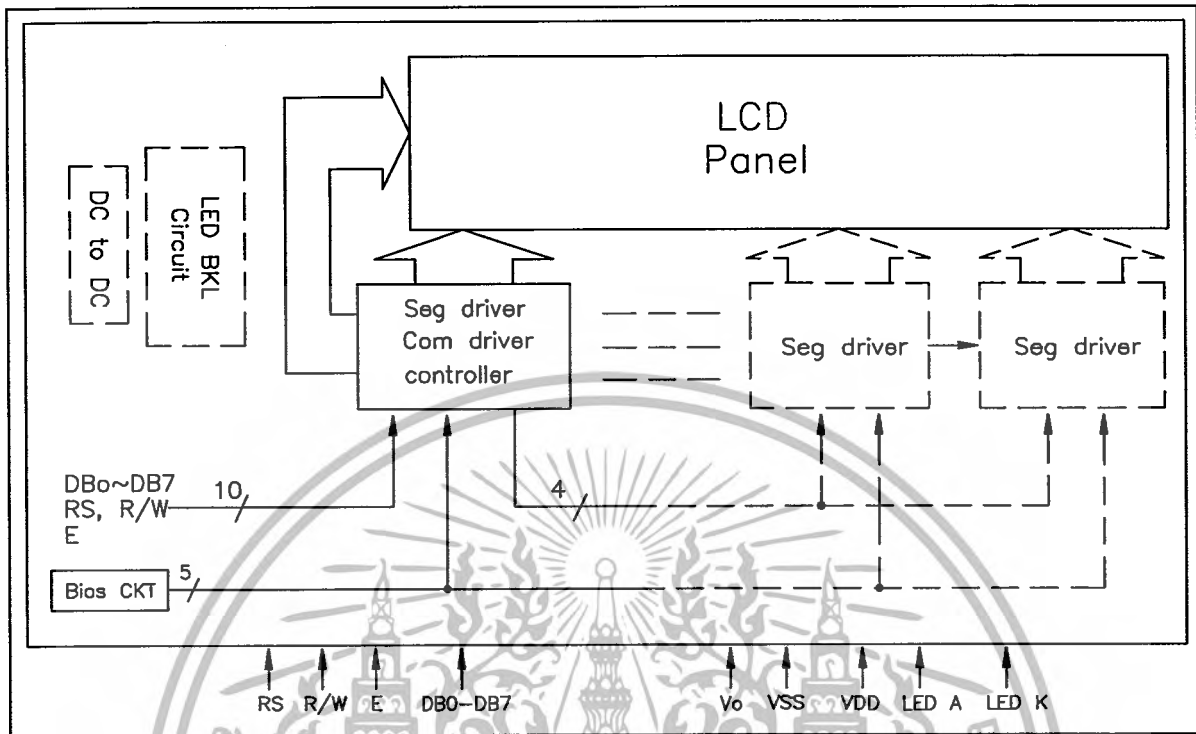


[Note 11] Description of Measuring Equipment

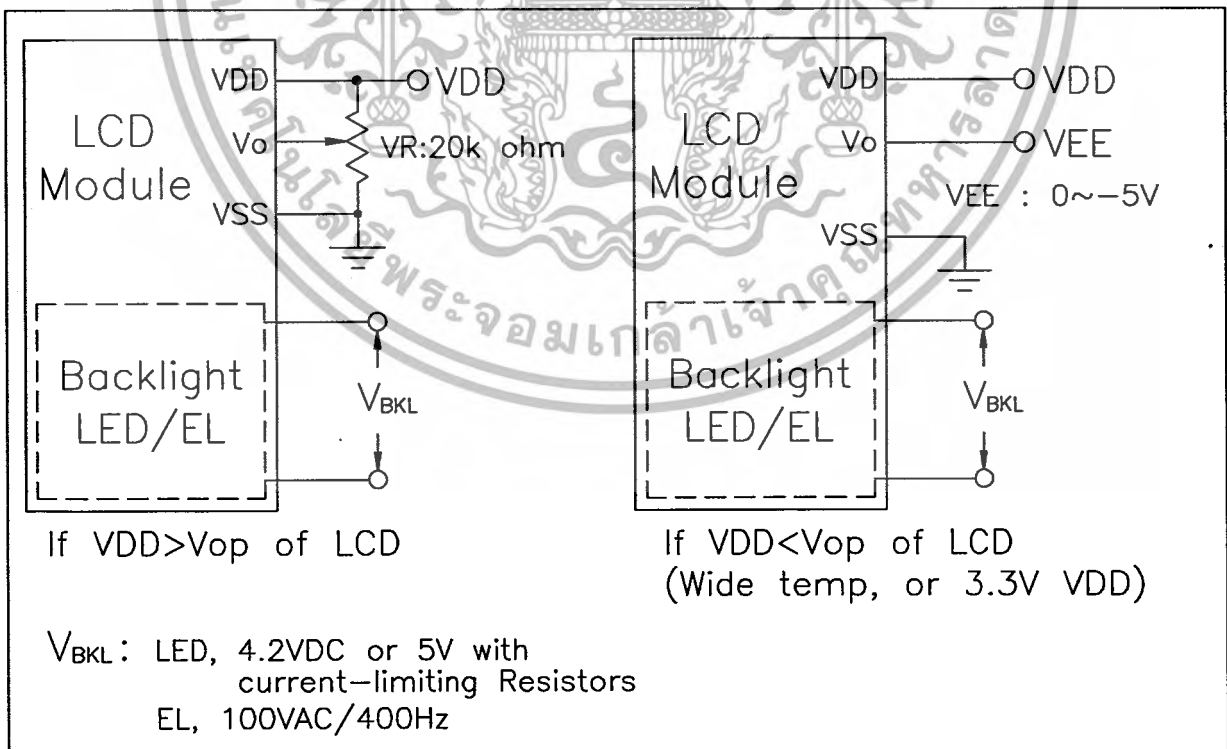


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM



POWER SUPPLY



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING CHARACTERISTICS

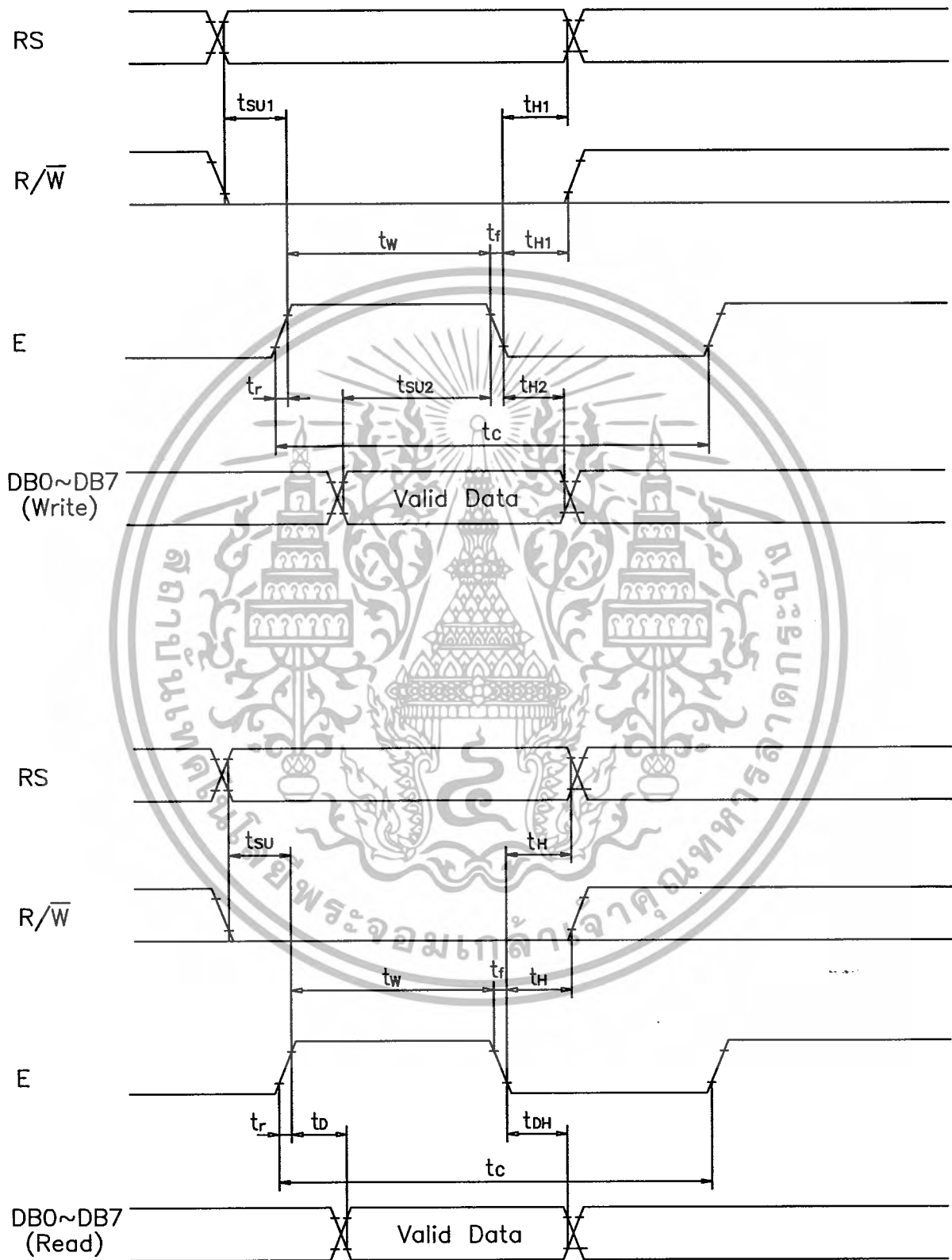
AC Characteristics ($V_{SS}=0V$, $V_{DD}=4.5V$ to $5.0V$, $T_a=0$ to $50^\circ C$)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode	E Cycle Time	t_c	500	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	20	ns
	E Pulse Width(High, Low)	t_w	230	-	-	ns
	R/W And RS Setup Time	t_{SU1}	40	-	-	ns
	R/W And RS Hold Time	t_{H1}	10	-	-	ns
	Data Setup Time	t_{SU2}	80	-	-	ns
	Data Hold Time	t_{H2}	10	-	-	ns
Read Mode	E Cycle Time	t_c	500	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	20	ns
	E Pulse Width(High, Low)	t_w	230	-	-	ns
	R/W And RS Setup Time	t_{SU}	40	-	-	ns
	R/W And RS Hold Time	t_H	10	-	-	ns
	Data Setup Time	t_D	-	-	120	ns
	Data Hold Time	t_{DH}	5	-	-	ns

AC Characteristics ($V_{SS}=0V$, $V_{DD}=2.7V$ to $4.5V$, $T_a=0$ to $50^\circ C$)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode	E Cycle Time	t_c	1000	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	25	ns
	E Pulse Width(High, Low)	t_w	450	-	-	ns
	R/W And RS Setup Time	t_{SU1}	60	-	-	ns
	R/W And RS Hold Time	t_{H1}	20	-	-	ns
	Data Setup Time	t_{SU2}	195	-	-	ns
	Data Hold Time	t_{H2}	10	-	-	ns
Read Mode	E Cycle Time	t_c	1000	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	25	ns
	E Pulse Width(High, Low)	t_w	450	-	-	ns
	R/W And RS Setup Time	t_{SU}	60	-	-	ns
	R/W And RS Hold Time	t_H	20	-	-	ns
	Data Setup Time	t_D	-	-	360	ns
	Data Hold Time	t_{DH}	5	-	-	ns

Read/Write Timing Chart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Commands

Instruction	Instruction Code										Description	Execution time(f_{osc} is 270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.53mS
Return Home	0	0	0	0	0	0	0	0	0	*	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53mS
Entry Mode	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39 μ S
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 μ S
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39 μ S
Function Set	0	0	0	0	1	DL	N	F	*	*	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39 μ S
Set CG RAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μ S
Set DD RAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μ S
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 μ S
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μ S
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μ S

* means don't care

COMMANDS DESCRIPTION

Clear Display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter). Return cursor to the original status. namely, bring the cursor to the left edge on first line of the display. Make entry mode increment (I/D = "1").

Return Home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

Return Home is cursor return home instruction. Set DDRAM address to "00H" into the address counter. Return cursor to its original site and return display to its original status, if shifted. Content of DDRAM is not changed.

Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Set the moving direction of cursor and display.

I/D : Increment / decrement of DDRAM address (cursor or blink)

When I/D = "High", cursor/blink moves to right and DDRAM address is increased by 1.

When I/D = "Low", cursor/blink moves to left and DDRAM address is decreased by 1.

* CGRAM operates the same as DDRAM, when read from or write to CGRAM.

SH: Shift of entire display

When DDRAM read (CGRAM read/write) operation or SH = "Low", shift of entire display is not performed. If SH = "High" and DDRAM write operation, shift of entire display is performed according to I/D value (I/D = "1", shift left, I/D = "0" : shift right).

Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Control display/cursor/blink ON/OFF 1 bit register.

D : Display ON/OFF control bit

When D = "High", entire display is turned on.

When D = "Low", display is turned off, but display data is remained in DDRAM.

C : Cursor ON/OFF control bit

When C = "High", cursor is turned on.

When C = "Low", cursor is disappeared in current display, but I/D register remains its data.

B : Cursor Blink ON/OFF control bit

When B = "High", cursor blink is on, that performs alternate between all the high data and display character at the cursor position.

When B = "Low", blink is off.

Cursor or Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

Without writing or reading of display data, shift right/left cursor position or display.

This instruction is used to correct or search display data. (Refer to Table 4)

During 2-line mode display, cursor moves to the 2nd line after 40th digit of 1st line.

Note that display shift is performed simultaneously in all the line.

When displayed data is shifted repeatedly, each line shifted individually.

When display shift is performed, the contents of address counter are not changed.

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1.
0	1	Shift cursor to the right, AC is increased by 1.
1	0	Shift all of the display to the left, cursor moves according to the display.
1	1	Shift all of the display to the right, cursor moves according to the display.

Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
00	0	0	0	1	DL	N	F	*	*

DL : Interface data length control bit

When DL = "High", it means 8-bit bus mode with MPU.

When DL = "Low", it means 4-bit bus mode with MPU. So to speak, DL is a signal to select 8-bit or 4-bit bus mode. When 4-bit bus mode, it needs to transfer 4-bit data by two times.

N : Display line number control bit

When N = "Low", it means 1-line display mode.

When N = "High", 2-line display mode is set.

F : Display font type control bit

When F = "Low", it means 5 X 8 dots format display mode

When F = "High", 5 x11 dots format display mode.

Set CG RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.

This instruction makes CGRAM data available from MPU.

Set DD RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.

This instruction makes DDRAM data available from MPU.

When 1-line display mode (N = 0), DDRAM address is from "00H" to "4FH".

In 2-line display mode (N = 1), DDRAM address in the 1st line is from "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

Read Busy Flag and Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether KS0066U is in internal operation or not. If the resultant BF is High, it means the internal operation is in progress and you have to wait until BF to be Low, and then the next instruction can be performed. In this instruction you can read also the value of address counter.

Write Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.

The selection of RAM from DDRAM, CGRAM, is set by the previous address set instruction : DDRAM address set, CGRAM address set. RAM set instruction can also determine the AC direction to RAM. After write operation, the address is automatically increased/decreased by 1, according to the entry mode.

Read Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.

The selection of RAM is set by the previous address set instruction. If address set instruction of RAM is not performed before this instruction, the data that read first is invalid, because the direction of AC is not determined. If you read RAM data several times without RAM address set instruction before read operation, you can get correct RAM data from the second, but the first data would be incorrect, because there is no time margin to transfer RAM data.

In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set instruction : it also transfers RAM data to output data register. After read operation address counter is automatically increased/decreased by 1 according to the entry mode. After CGRAM read operation, display shift may not be executed correctly.

NOTE: In case of RAM write operation, after this AC is increased/decreased by 1 like read operation. In this time, AC indicates the next address position, but you can read only the previous data by read instruction.

DD RAM ADDRESSING

For 16×1 or 8×1 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

For 16×2 or 8×2 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

For 16×4 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

For 20×2 Display

Character	1	2	3	4	5	6	7	8	9	10	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	50	51	52	53

For 20×4 Display

Character	1	2	3	4	5	6	7	8	9	10	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	50	51	52	53
	14	15	16	17	18	19	1A	1B	1C	1D	24	25	26	27
	54	55	56	57	58	59	5A	5B	5C	5D	64	65	66	67

For 40×2 Display

Character	1	2	3	4	5	6	7	8	9	10	37	38	39	40
DD RAM Address	00	01	02	03	04	05	06	07	08	09	24	25	26	27
	40	41	42	43	44	45	46	47	48	49	64	65	66	67

For 40×4 Display

Character	E	1	2	3	4	5	6	7	8	9	10	37	38	39	40
DD RAM Address	E1	00	01	02	03	04	05	06	07	08	09	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	64	65	66	67
	E2	00	01	02	03	04	05	06	07	08	09	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	64	65	66	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CG RAM MAPPING

Character Code (DD RAM data)								CG RAM Address						Character Patterns (CG RAM data)																																				
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0																													
High				Low				High			Low			High				Low																																
0 0 0 0 * 0 0 0								0 0 0			0	0	0	0	0	0	*	*	*	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	←Character Pattern
											1	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	←Cursor								
0 0 0 0 * 0 0 1								0 0 1			0	0	0	0	0	1	*	*	*	1	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	0	1	1	1	←Character Pattern										
											0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	0	0	1	1	1	1	1	1	←Cursor								
⋮								⋮			⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮														
0 0 0 0 * 1 1 1								1 1 1			0	0	0	0	0	1	*	*	*	1	1	1	1	1	1	0	0	0	1	1	1	1	0	1	1	0	0	0	1	←Character Pattern										
											0	1	1	1	0	0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	1	1	1	←Cursor													

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHARACTER FONT TABLE

Upper 4 bits Lower 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)			0	o	P	'	F				—	๑	๒	๓	๔
0001	CG RAM (2)		!	1	A	Q	a	๑			๑	๒	๓	๔	๕	๖
0010	CG RAM (3)		"	2	B	R	b	r			๑	๒	๓	๔	๕	๖
0011	CG RAM (4)		#	3	C	S	c	s			๑	๒	๓	๔	๕	๖
0100	CG RAM (5)		\$	4	D	T	d	t			๑	๒	๓	๔	๕	๖
0101	CG RAM (6)		%	5	E	U	e	u			๑	๒	๓	๔	๕	๖
0110	CG RAM (7)		&	6	F	V	f	v			๑	๒	๓	๔	๕	๖
0111	CG RAM (8)		'	7	G	W	g	w			๑	๒	๓	๔	๕	๖
1000	CG RAM (1)		(8	H	X	h	x			๑	๒	๓	๔	๕	๖
1001	CG RAM (2)		,	9	I	Y	i	y			๑	๒	๓	๔	๕	๖
1010	CG RAM (3)		*	๐	J	Z	j	z			๑	๒	๓	๔	๕	๖
1011	CG RAM (4)		+	๑	K	[k	[๑	๒	๓	๔	๕	๖
1100	CG RAM (5)		,	๒	L]	l]			๑	๒	๓	๔	๕	๖
1101	CG RAM (6)		—	๓	M	^	m	^			๑	๒	๓	๔	๕	๖
1110	CG RAM (7)		.	๔	N	_	n	_			๑	๒	๓	๔	๕	๖
1111	CG RAM (8)		/	๕	O	`	o	`			๑	๒	๓	๔	๕	๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

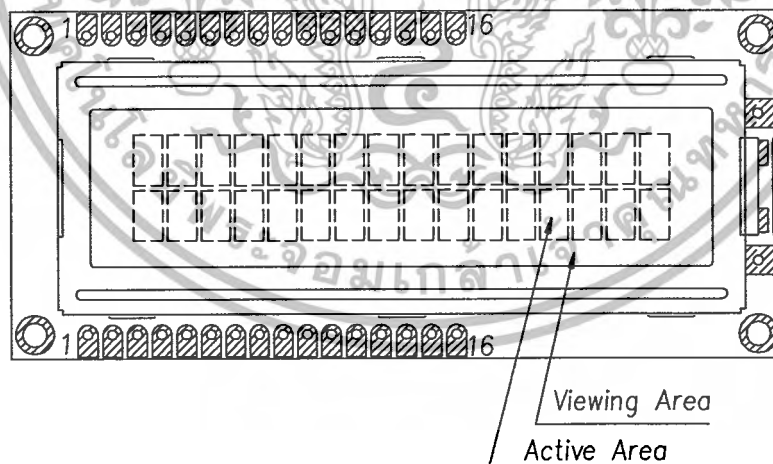
RELIABILITY TEST

No	Item	Conditions		Note
1	High Temp. Operation	70°C	240HR	
2	High Temp. Storage	80°C	240HR	
3	Low Temp. Operation	-20°C	240HR	
4	Low Temp. Storage	-30°C	240HR	
5	High Temp./Humid Storage	60°C 90%RH	240HR	
6	Thermal Shock	-00°C ,30min +50°C ,30min	10 cycles	
7	Vibration Test (IEC-68-2-6)	Frequency : 10~55 Hz Duration : 20 times, 6 min/time Amplitude : 0.75 mm	-	
8	Shock (IEC 68-2-27)	Duration : 11 mS Acceleration : 100g	-	X, Y, Z direction

APPEARANCE CHECK

CONDIITON OF APPEARANCE CHECK:

- (1) Specimen shall be checked by eyes in distance of 30cm under 40w-fluorescence lamp.
- (2) Checking direction shall be in 45 degree from perpendicular line op specimen surface.



HANDLING PRECAUTIONS

- (1) Treat polarizer very carefully since it is easy to be damaged.
- (2) When cleaning the display surface, use soft cloth (e.g. gauss) with a solvent (recommended below) and wipe lightly.
 - ◆ ethyl alcohol
 - ◆ iso-procolol

Do not wipe the display surface with dry or hard materials that will damage the polarizer surface.

Do not use the following solvents:

- ◆ water
- ◆ ketone
- ◆ aromatics

- (3) Direct current causes electro-chemical reaction with remarkable degradation of the display quality. Give careful consideration to prevent direct current at ON/OFF timing and during operation.
- (4) Avoid strong shock and drop from the height.
- (5) To prevent LCD panels from degradation, do not operate or store them exposed directly to sunshine or high temperature/humidity.
- (6) Give careful consideration to avoid electrical static discharge with causes uneven contrast.
- (7) Even a small condensation on the contact pads (terminals) causes electro-chemical reaction which makes missing row and column. Give careful attention to avoid condensation. When assembling with zebra connector, clean the surface of the pads with alcohol and keep the air very clean.

LCD PRODUCT QUALITY STANDARD

DISPLAY APPEARANCE

No	Item	Criteria
1	inclusions (black spot, white spot, dust)	<p>(1)round type diameter mm(a*) no of defect* a ≤ 0.20 neglect 0.20 < a ≤ 0.35 5max 0.35 < a none</p> <p>(2)linear type length mm(l) width mm(W) no. of defect na W ≤ 0.03 neglect 1 ≤ 3 0.03 < W ≤ 0.08 6 3 < 1 0.08 < W none</p>
2	scratch	<p>1.scratch on protective film is permitted. 2.scratch on polarizer shall be as follow: (1)round type diameter mm(a*) no of defect a ≤ 0.15 neglect 0.15 < a ≤ 0.20 2 max 0.20 < a none</p> <p>(2)linear type be judged by 1.-(2) linear type</p>
3	dent	diameter < 1.5mm
4	bubble	not exceeding 0.5mm average diameter is acceptable between glass and polarizing film
5	pin hole	$(a+b)/2 \leq 0.15\text{mm}$ maximum number: ignored $0.15 < (a+b)/2 \leq 0.20\text{mm}$ maximum number:10
6	dot defect	$(a+b)/2 \leq 0.20\text{mm}$ maximum number: ignored $0.20 < (a+b)/2 \leq 0.30\text{mm}$ maximum number:5 x=width
7	contrast irregularity(spot)	<p>diameter spec no of defect a ≤ 0.50mm neglect 0.50 < a ≤ 0.75 5 0.75 < a ≤ 1.00 3 1.00 < a none</p>
8	dot width	design width ±15%
9	color tone and uniformity	obvious uneven color is not permitted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

