

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์ติดตามใบหน้าคน

FACE TRACKING ROBOT



โดย
นาย พฤษณ์ นิลสุข
นางสาว ภาวินี รัตนคันหยง

รฟท.
พ.ศ. ๒๕๕๓
๒๕๕๘

เลขหมู่.....**62353**
เลขทะเบียน.....
วันเดือนปี...**16 ส.ค. 2549**

b. **11621418**

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2548

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ติดตามใบหน้าคน

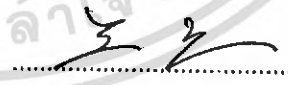
Face Tracking Robot

ผู้จัดทำ

นาย พฤกษ์ นิลสุข 45010514

นางสาว ภาวินี รัตนตันหยง 45010589


.....อาจารย์ที่ปรึกษา
(ผศ. ทวาร เบญจนราษฎร์)


.....อาจารย์ที่ปรึกษา
(รศ. ดร. จงกล งามวิวิทย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ติดตามใบหน้าคน

Face Tracking Robot

โดย

นาย พฤกษ์ นิลสุข

นางสาว ภาวินี รัตนตันหยง

อาจารย์ที่ปรึกษา

ศศ.ถาวร เบญจนาสุสุทธิ

รศ.ดร.จงกล งามวิวิทย์

บทคัดย่อ

ปฏิญญาพันธบัตรฉบับนี้นำเสนอการศึกษาขั้นตอนการสร้าง และออกแบบหุ่นยนต์ติดตามใบหน้าคน โดยใช้กล้องเว็บแคม (Webcam) เป็นอุปกรณ์รับภาพ ซึ่งมีจุดประสงค์ในการศึกษาหลักการของการประมวลผลภาพ (Image Processing) และนำความรู้ทางด้านระบบควบคุมไปใช้งานในการควบคุมการเคลื่อนที่ของหุ่นยนต์เพื่อติดตามใบหน้าคน ขั้นตอนการศึกษาประกอบด้วยการศึกษาทฤษฎีการประมวลผลภาพ และการทำงานของโปรแกรมในการระบุใบหน้าคน จากนั้นทำการออกแบบการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ เพื่อให้สามารถเคลื่อนที่ได้ 3 มิติ ท้ายที่สุดทำการทดลองจริงโดยการให้เคลื่อนที่ตามคนที่เคลื่อนที่ หุ่นยนต์นี้อาจนำไปใช้ประโยชน์ในการระบุตำแหน่ง และเคลื่อนที่ตามใบหน้าคนผ่านทางหน้าจอคอมพิวเตอร์โดยไม่ต้องคอยเฝ้าดูอย่างใกล้ชิด หรือประโยชน์ในการใช้รักษาความปลอดภัยได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Face Tracking Robot

By

Pruk Ninsook

Pavinee Ratanatanyong

Advisor

Asst. Prof. Taworn Benjanarasuth

Assoc. Prof. Dr Jongkol Ngamwiwit

ABSTRACT

This thesis presents the study and implementation procedures of a face tracking robot using a web camera to receive the video stream. The objective is to utilize the knowledge of image processing and control system design for controlling the movement of the robot. First, the theory of image processing and the face tracking programme algorithm have been studied. Secondly, the design of controller is implemented via a microcontroller to control the 3 dimensional movement of the robot. Finally, the experiments were conducted to evaluate the efficiency. The robot can be used to pin point the position of a face and moves as the face moves. Thus, attention can be paid via computer toward a particular area. Also, this can be used as a part of a security system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความเมตตาอย่างสูง จากอาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำ ข้อคิดเห็นที่เป็นประโยชน์ต่อการทำโครงการเป็นอย่างมาก ขอขอบพระคุณ รศ.ดร. เกียรติศักดิ์ คมวัชระ อาจารย์รัชชชัย คำศรี ภาควิชาระบบควบคุม และ ดร.อรัญญา วลัยรัชต์ ภาควิชาคอมพิวเตอร์ ที่ได้ให้คำปรึกษาและความช่วยเหลือแก่คณะผู้จัดทำเป็นอย่างดี

สุดท้ายนี้ ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ของคณะผู้จัดทำ ที่ได้ให้ความสนับสนุนและให้โอกาสผู้จัดทำได้เล่าเรียนจนถึงทุกวันนี้ อีกทั้ง ขอขอบคุณเพื่อนๆ พี่ๆ ที่ให้คำปรึกษาและช่วยเหลืออย่างดีเยี่ยมมาโดยตลอด จนทำให้ปริญญาานิพนธ์ฉบับนี้เสร็จสมบูรณ์

คณะผู้จัดทำ

นาย พฤษณ์ นิลสุข

นางสาว ภาวินี รัตนตันหยง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ระบบการมองเห็น (Vision System)	3
2.2 การแปลงภาพให้เป็นภาพเชิงดิจิทัล (Image Digitization)	3
2.2.1 การรับภาพ (Image Acquisition)	4
2.2.2 การประมวลผลภาพ (Image Processing)	4
2.2.2.1 จุดเดียวกันในรูปภาพเดียวกัน (Point by point in one image)	4
2.2.2.2 จุดเดียวกันในรูปแบบที่ต่างกัน (Corresponding points on different images)	4
2.2.2.3 บริเวณจุดนั้นในภาพนั้น (Using regional points in an image)	4
2.2.3 ผลที่ได้และการแสดงผล (Output and Display)	4
2.3 หลักการเบื้องต้นในการประมวลผลภาพ (Image Processing Fundamentals)	4
2.3.1 การสุ่มเลือกจุดตำแหน่ง (Image Sampling)	5
2.3.2 การประมาณค่าความเข้มของแสง (Intensity Approximation)	6
2.3.3 ฮิสโตแกรม (Histogram)	8
2.3.4 การแปลงระดับสีเทาโดยการทำเทรชโฮล (Threshold)	8
2.4 พื้นฐานและระบบสีของสีโมเดล	8
2.4.1 RGB	9
2.4.2 HSV	9
2.4.3 YCbCr	10
2.5 โลบารีซของโปรแกรมที่ใช้ในการระบุตำแหน่งใบหน้าคน	10
2.5.1 ฟังก์ชันในโลบารีซของ โอเพินซีวี (Functionality of OpenCV Library)	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.1 การวิเคราะห์ภาพ (Image Analysis)	11
2.5.1.2 การระบุตำแหน่งของวัตถุ (Object Recognition)	12
2.5.1.3 การสร้างภาพ 3 มิติ (3D Reconstruction)	13
2.6 การสื่อสารอนุกรมและ RS-232	13
2.6.1 การรับส่งข้อมูลแบบขนาน (Parallel Communication)	13
2.6.2 การรับส่งข้อมูลแบบอนุกรม (Serial Communication)	14
2.6.2.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส	14
2.6.2.2 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232	15
2.6.2.3 ยูอาร์ที (UART)	19
2.6.2.4 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232	20
2.6.2.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232	25
2.6.2.6 แอคเคอเรสของพอร์ตอนุกรม	25
2.7 ไอซี MAX232	26
2.8 ตัวกำเนิดแรงดัน 5V จาก 12V	27
2.9 โครงสร้างและสถาปัตยกรรมของ PIC 16F877	27
2.9.1 คุณสมบัติของ PIC 16F877	27
2.9.2 สถาปัตยกรรมฮาร์ดแวร์ PIC16F877	29
2.10 ดีซีมอเตอร์ (DC Motor)	34
บทที่ 3 การสร้างและการออกแบบ	35
3.1 หลักการทำงานของการทำงานของจับใบหน้าคน	35
3.1.1 การรับภาพ	36
3.1.2 การเทียบสีและแบบแผนของส่วนต่างๆ บนใบหน้าคน	36
3.1.3 การตีกรอบและแสดงผล	36
3.1.4 การหาค่าความผิดพลาด	38
3.1.5 โปรแกรมในกำหนดการเคลื่อนที่ของหุ่นยนต์	40
3.1.5.1 โปรแกรมการเคลื่อนที่ซ้ายขวาและขึ้นลง	40
3.1.5.2 โปรแกรมการกำหนดระยะระหว่างใบหน้าคนกับหุ่นยนต์	41
3.1.6 การส่งค่าเอาต์พุตจากคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรม	41
3.2 การควบคุมตำแหน่งของหุ่นยนต์	41
3.2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)	43
3.2.2 ดีซีมอเตอร์ (DC Motor)	46

3.2.2.1 มอเตอร์ที่ใช้ขับเคลื่อนหุ่นยนต์	46
3.2.2.2 มอเตอร์ที่ใช้ขับเคลื่อนกล้องเว็บแคม	47
3.2.3 วงจรขับมอเตอร์ (Motor Driver)	47
3.3 โครงสร้างทางกลศาสตร์	48
3.3.1 ล้อ	48
3.3.2 ฐานหุ่นยนต์	49
3.3.3 ส่วนบนของหุ่นยนต์	50
บทที่ 4 ผลการทดลอง	51
4.1 ผลการทดลองในส่วนของโปรแกรมการตรวจจับใบหน้าคน	51
4.2 ผลการทดลองการเคลื่อนที่ของหุ่นยนต์	52
4.2.1 การทดลองบนสภาพพื้นผิวต่างๆ	52
4.2.2 การทดลองการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ	52
4.2.2.1 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ถอยหลัง	52
4.2.2.2 ผลการทดลองให้หุ่นยนต์เคลื่อนที่เดินหน้า	54
4.2.2.3 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ไปทางขวา	55
4.2.2.4 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ไปทางซ้าย	57
4.2.2.5 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ในแนวทแยง	58
บทที่ 5 สรุปผลการทดลองและวิจารณ์	60
5.1 สรุปผลการทดลอง	60
5.2 ปัญหาที่พบและแนวทางการแก้ไข	60
5.3 แนวทางในครั้งถัดมาและการพัฒนา	60
ภาคผนวก ก การติดตั้งไลบรารีของโอเพ่นซีวี	61
ภาคผนวก ข การรันโปรแกรมที่ใช้ไลบรารีของโอเพ่นซีวี	63
ภาคผนวก ค โปรแกรมสำหรับการจับใบหน้าคน	67
ภาคผนวก ง โปรแกรมควบคุมไมโครคอนโทรลเลอร์	77
ภาคผนวก จ เอกสารข้อมูลไอซี (Data Sheet)	83
เอกสารอ้างอิง	94

สารบัญรูปภาพ

รูปที่	หน้า
2.1 การแปลงภาพ 2 มิติที่ได้ให้เป็นภาพเชิงดิจิทัล โดยการสุ่มเลือกจุดตำแหน่ง	5
2.2 ความละเอียดของภาพ	5
2.3 ค่าระดับสีเทา (Grey scale) ของจุดใดๆ ในภาพ	6
2.4 จุดของภาพที่ถูกแทนที่ด้วยค่าตัวเลขระดับสีเทา	6
2.5 การไล่สีของภาพเมื่อใช้จำนวนระดับสีต่างกัน	7
2.6 ฮิสโตแกรมของระดับสีเทา	8
2.7 รูปแบบสี RGB	9
2.8 รูปแบบสี HSV	10
2.9 การทำเทรชโฮลด์ (Threshold)	11
2.10 ภาพที่เกิดจากการทำ Erosion และ Dilation	11
2.11 การหาขอบของภาพ	12
2.12 การทำ HMM	13
2.13 การทำ 3D Reconstruction	13
2.14 การส่งข้อมูลแบบขนาน	14
2.15 การส่งข้อมูลแบบอนุกรม	14
2.16 การส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที	14
2.17 การส่งข้อมูลขนาด 8 บิตแบบอนุกรมด้วยความเร็ว 9600 bps	15
2.18 ระดับแรงดันของสัญญาณอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"	16
2.19 คอนเน็คเตอร์แบบ DB-9 และ DB-25	17
2.20 การเชื่อมต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ	17
2.21 ตำแหน่งขาต่างๆของ MAX 232	27
2.22 ขาต่างๆ ของ LM 7805	27
2.23 สถาปัตยกรรมภายในของ PIC	29
2.24 การจัดวางตำแหน่งขาสัญญาณต่างๆ ของ PIC 16F877A และประเภทโถงเสียบ	30
3.1 โพล์วชาร์ตแสดงการระบุตำแหน่งของไบพิน	37
3.2 ตัวอย่างการประมวลผลส่วนต่างๆ ของไบพิน	38
3.3 แสดงการหาค่าความผิดพลาดของตำแหน่งไบพินจากจุดอ้างอิง	38
3.4 การกำหนดช่วงค่าความผิดพลาดในแนวแกน x และ แกน y	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจรรวมของระบบทั้งหมด	44
3.6 ทำงานของโปรแกรมไมโครคอนโทรลเลอร์	46
3.7 การติคมอเตอร์เข้ากับล้อออมนิ	47
3.8 ล้อออมนิ	48
3.9 การทำงานของล้อออมนิเพื่อให้เคลื่อนที่ในทิศทางต่าง	48
3.10 ตัวอย่างการเคลื่อนที่ของล้อออมนิ	49
3.11 หุ่นยนต์ตรวจจับและเคลื่อนที่ตามใบหน้าคน	50
3.12 สัดส่วนของหุ่นยนต์ตรวจจับและเคลื่อนที่ตามใบหน้าคน	50
4.1 ผลการทดลองจับภาพใบหน้าคนในลักษณะต่างๆ	51
4.2 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่ถอยหลัง	53
4.3 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ถอยหลัง	54
4.4 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่เดินหน้า	55
4.5 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่เดินหน้า	55
4.6 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ไปทางขวา	56
4.7 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่ไปทางขวา	56
4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ไปทางซ้าย	57
4.9 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่ไปทางซ้าย	58
4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่แนวทแยง	59
4.11 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่แนวทแยง	59
ข.1 การตั้งค่าในส่วนของ All Configurations Preprocessor	63
ข.2 การตั้งค่าในส่วนของ All Configurations Input	63
ข.3 การตั้งค่าในส่วนของ Debug Input	64
ข.4 การตั้งค่าในส่วนของ Release Input	64
ข.5 การตั้งค่าในส่วนของ Project Dependencies	65

สารบัญตาราง

ตารางที่	หน้า
2.1 การจัดขาของคอนเน็คเตอร์อนุกรมแบบมาตรฐาน RS-232 แบบ DB-9 และ DB-25	18
2.2 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์เอ็นเอเบิลการอินเทอร์รัพต์	20
2.3 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์แสดงโหมดและสถานะการอินเทอร์รัพต์	21
2.4 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์กำหนดรูปแบบของข้อมูล	22
2.5 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์ควบคุมโมเด็ม	23
2.6 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์ตำแหน่ง 05H	23
2.7 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์แสดงสถานะของโมเด็ม	24
2.8 ข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม	26
2.9 หน้าที่การทำงานของขาสัญญาณของ PIC 16F877	31
3.1 ค่าเอาต์พุตในแต่ละกรณีและลักษณะการเคลื่อนที่	42
3.2 การกำหนดค่าเอาต์พุตของไมโครคอนโทรลเลอร์	45

บทที่ 1

บทนำ

ในปัจจุบันถือว่ากล้องเป็นสิ่งจำเป็นมาก โดยขึ้นอยู่กับการใช้งานแต่ละประเภท จึงสามารถนำกล้องมาใช้ให้เกิดประโยชน์กับผู้ใช้งานในด้านต่างๆ ได้หลายประเภท เช่น ถ่ายภาพ ส่งภาพผ่านระบบอินเทอร์เน็ต ใช้ทำกล้องวงจรปิด และสามารถนำมาผสมผสานกับระบบอิเล็กทรอนิกส์เพื่อประยุกต์ใช้ในการควบคุมระยะไกล ซึ่งประกอบด้วยส่วนฮาร์ดแวร์และซอฟต์แวร์ เพียงเท่านี้ก็สามารนำไปใช้ประโยชน์ในที่ต่างๆ ที่ผู้ใช้งานอยู่ห่างไกลจากอุปกรณ์ เช่น ระบบรักษาความปลอดภัย ซึ่งจะก่อให้เกิดประโยชน์เป็นอย่างมาก

โครงการนี้ได้นำเอาแนวความคิดข้างต้นมาพัฒนา เพื่อสร้างหุ่นยนต์ที่สามารถเคลื่อนที่ตามใบหน้าคนได้เพื่อใช้ในการติดตามตัวบุคคล โดยมีหลักการทำงานคร่าวๆ ดังนี้

1. กล้องเว็บแคม (Webcam) ทำหน้าที่รับภาพและแสดงภาพบนจอคอมพิวเตอร์
2. การประมวลผลภาพ (Image Processing) เป็นการนำภาพมาประมวลผลเพื่อให้สามารถระบุบริเวณของใบหน้าคนได้
3. ควบคุมการเคลื่อนที่ของหุ่นยนต์ และรักษาระยะห่างระหว่างใบหน้าคนกับหุ่นยนต์ให้เป็นที่ตามต้องการโดยใช้ไมโครคอนโทรลเลอร์

1.1 วัตถุประสงค์

1. ศึกษาการใช้โปรแกรมคอมพิวเตอร์เพื่อติดตามการเคลื่อนที่ของใบหน้าคน
2. สร้างหุ่นยนต์เพื่อใช้ในการติดตามการเคลื่อนที่ของใบหน้าคน โดยสามารถเคลื่อนที่ได้ทั้ง 3 มิติ คือ เดินหน้า ถอยหลัง ซ้าย ขวา ขึ้นบน และลงล่าง

1.2 ขอบเขตการทำโครงการ

1. ใบหน้าคนต้องอยู่ใกล้กล้องเว็บแคมเพียงพอที่จะเห็นใบหน้าได้ชัดเจน ไม่เอียงมากเกินไป และและไม่เคลื่อนที่เร็วเกินไป
2. สีของพื้นหลังจะต้องมีความแตกต่างจากสีผิวของใบหน้าคนพอสมควร
3. จะต้องมิแสงสว่างที่เพียงพอ ไม่มีมืดและไม่สว่างเกินไป
4. พื้นผิวที่จะให้หุ่นยนต์ทำงานจะต้องเป็นพื้นเรียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เครื่องมือที่ใช้ในการพัฒนา

1. Microsoft Visual C++6
2. Microsoft Visual C++.Net 2003
3. OpenCV Library ของ Intel Corporation
4. DirectX 9.0 SDK (summer 2004)

1.4 วิธีการดำเนินงาน

โครงการนี้เริ่มต้นด้วยการศึกษาทฤษฎีต่างๆ ที่เกี่ยวข้องดังที่จะกล่าวในบทที่ 2 จากนั้นจึงทำการออกแบบตัวโปรแกรมโดยแยกเป็น 2 ส่วนหลัก คือ ส่วนแรก เป็นการประยุกต์ใช้โปรแกรมการตรวจจับใบหน้าคนโดยใช้หลักการประมวลผลภาพบนคอมพิวเตอร์ส่วนบุคคล และส่วนที่สองเป็นการควบคุมหุ่นยนต์ด้วยไมโครคอนโทรลเลอร์ ดังที่จะกล่าวในบทที่ 3 จากนั้นจึงทำการทดลองการทำงานของหุ่นยนต์และแก้ไขข้อผิดพลาดเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ แล้วทำการสรุปผลการทดลองว่ามีข้อดีข้อเสียอย่างไร รวมถึงแนวทางการปรับปรุงและพัฒนาในอนาคต ซึ่งจะกล่าวในบทที่ 4 และบทที่ 5

บทที่ 2

ทฤษฎีและหลักการ

2.1 ระบบการมองเห็น (Vision System)

รูปภาพไม่ว่าจะเป็นภาพที่ถ่ายโดยใช้กล้องธรรมดาหรือแบบดิจิทัล คือ จุดสีหลาย ๆ จุดที่นำมาเรียงต่อกัน จนสามารถมองเห็นเป็นรูปภาพได้ มนุษย์สามารถเข้าใจถึงเนื้อหาของภาพได้ว่าภาพที่ปรากฏนั้นให้ความพึงพอใจ ความน่าสนใจมากน้อยเพียงใด แต่เมื่อรูปภาพถูกนำมาทำเป็นภาพในคอมพิวเตอร์ มันจะรู้และเข้าใจภาพเป็นเพียงแค่ว่าเป็นจุดข้อมูลหลายๆ จุดที่เรียงต่อกันเท่านั้น ดังนั้นการมองภาพของมนุษย์กับคอมพิวเตอร์จึงแตกต่างกันอย่างสิ้นเชิง

2.2 การแปลงภาพให้เป็นภาพเชิงดิจิทัล (Image Digitization)

ภาพเป็นกระบวนการทางแสง (Optical Process) ซึ่งเกิดจากพลังงานคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Spectrum) หลายๆ ช่วงความถี่ เช่น แสงธรรมดา รังสีเอ็กซ์เรย์ (X-Ray) รังสีอินฟราเรด (Infrared) เป็นต้น และพลังงานเสียง เช่น อัลตราซาวนด์ (Ultrasound) ตกกระทบวัตถุแล้วสะท้อนกลับมาสู่ประสาทรับรู้ทางตาของมนุษย์ หรืออุปกรณ์ตรวจจับ เช่น เซนเซอร์ (Sensor) เป็นต้น

ในทางเทคนิคนั้น ภาพดิจิทัลก็คือ ฟังก์ชัน 2 มิติ หรือ $f(x,y)$ ของค่าความเข้มของแสงโดยที่ x และ y คือ ค่าที่บอกถึงตำแหน่งในระบบพิกัดฉาก และค่าของฟังก์ชัน ณ ตำแหน่งใดๆ จะเป็นสัดส่วนกับความสว่างของแสง ณ ตำแหน่งนั้น กระบวนการแปลงภาพให้เป็นภาพในเชิงดิจิทัล มีกระบวนการ 3 ขั้นตอน คือ การรับภาพ (Image Acquisition) กระบวนการประมวลผลภาพ (Image Processing) และผลที่ได้หรือการแสดงผล (Output or Display)

2.2.1 การรับภาพ (Image Acquisition)

ก่อนที่จะได้ภาพมา อันดับแรกเราจะต้องทำการถ่ายภาพเสียก่อน การถ่ายภาพเป็นการแปลงภาพเชิงต่อเนื่อง (Continuous Image) แบบ 3 มิติ ให้เป็นภาพเชิงต่อเนื่อง 2 มิติ โดยใช้อุปกรณ์เชิงแสง (Optical Device) เช่น กล้องถ่ายรูป เพื่อแปลงภาพให้มาเป็นภาพบนฟิล์ม รูปถ่ายบนกระดาษ หรือภาพบนจอคอมพิวเตอร์

โดยปกติแล้ว ภาพที่เรามองเห็นกันอยู่นั้น มีทั้งความกว้าง ความสูง และความลึก ซึ่งเป็นแบบ 3 มิติ นั่นเอง การถ่ายภาพด้วยกล้องจะทำให้เราได้ภาพมา แต่จะเป็นภาพที่มีแต่ความกว้าง และความสูงเท่านั้น ซึ่งเป็นภาพแบบ 2 มิติ เพราะเราไม่อาจจะถ่ายภาพความลึกของสถานที่มาได้ ในส่วนของการบันทึกภาพนี้ เป็นหน้าที่ของกลไกทางแสงในตัวกล้อง ที่จะทำหน้าที่รับภาพเข้ามา ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ คือ การสร้างภาพใหม่โดยการแยกส่วนของข้อมูลที่เรารสนใจออกมาพิจารณา โดยการทำงานพื้นฐานของการประมวลผล ได้แก่ การกำจัดสิ่งรบกวนของภาพ (Noise Elimination) การปรับแต่งขอบภาพให้ดีขึ้น (Edge Enhancement) การกรองภาพ (Filtering) การปรับปรุงเปลี่ยนแปลงค่าระดับสีเทา (Gray Scale Modification) ซึ่งความซับซ้อนของการประมวลผลภาพจะขึ้นอยู่กับลักษณะการใช้งาน และจะมีวิธีประมวลผล 3 อย่าง ดังนี้

2.2.2.1 จุดเดียวกันในรูปภาพเดียวกัน (Point by point in one image)

วิธีนี้เป็นการสร้างภาพใหม่โดยการเปลี่ยนค่าแบบจุดต่อจุด โดยจุดหนึ่งคือจุดในรูปภาพเดิม และอีกจุดหนึ่งจุดในรูปภาพใหม่ที่ได้รับการปรับปรุงเปลี่ยนแปลงจากรูปเดิมแล้ว เช่น การแปลงภาพในระบบเลขฐาน 2 จากจุดที่มีค่าเป็น 0 ถูกเปลี่ยนค่าเป็น 1 และจากค่า 1 ถูกเปลี่ยนเป็น 0

2.2.2.2 จุดเดียวกันในรูปแบบที่ต่างกัน (Corresponding points on different images)

วิธีนี้เป็นการสร้างภาพใหม่โดยการจับคู่แต่ละจุดจากแหล่งภาพ 2 แหล่งหรือมากกว่า จุดของรูปภาพที่แตกต่างกันสองจุดนี้ จะนำมาลบกันเพื่อหาค่าที่เปลี่ยนแปลงไป ข้อมูลของจุดในภาพใหม่จะเกิดจากการรวมกันของตัวเลขที่ต่างกันของภาพ 2 ภาพนั้น

2.2.2.3 บริเวณจุดนั้นในภาพนั้น (Using regional points in one image)

วิธีนี้เป็นการสร้างภาพใหม่โดยการหาค่าเฉลี่ยรอบๆ จุดนั้นในภาพนั้น ค่าของจุดในภาพใหม่ก็คือค่าเฉลี่ยของจุดในภาพเดิมนั่นเอง

2.2.3 ผลที่ได้และการแสดงผล (Output and Display)

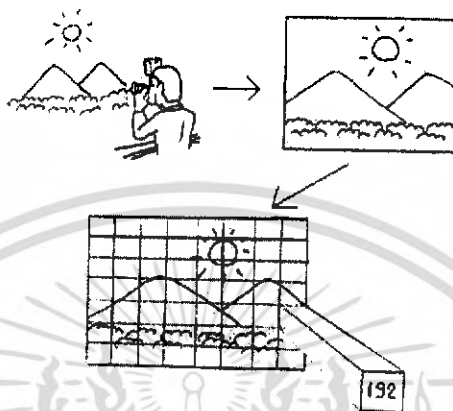
รูปแบบของการแสดงผลนั้นขึ้นอยู่กับว่า จะนำข้อมูลที่ได้ไปใช้ต่อในลักษณะใด หรือแสดงผลอย่างไรจึงจะเหมาะสมที่สุด เช่น แสดงผลโดยการพิมพ์ การแสดงผลผ่านหน้าจอ การแสดงผลในรูปของสัญญาณควบคุม เป็นต้น

2.3 หลักการเบื้องต้นในการประมวลผลภาพ (Image Processing Fundamentals)

ในการประมวลผลภาพขั้นพื้นฐานนั้น จำเป็นจะต้องมีความรู้ทางทฤษฎีหลักๆ ดังนี้

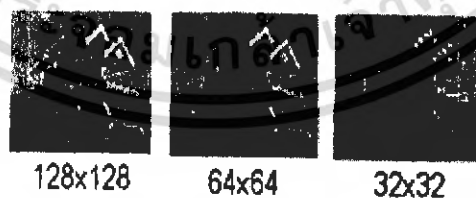
2.3.1 การสุ่มเลือกจุดตำแหน่ง (Image Sampling)

เป็นการแปลงภาพ 2 มิติที่ได้ให้เป็นภาพเชิงดิจิทัล โดยการสุ่มเลือกจุดตำแหน่ง โดยสุ่มเลือกเฉพาะบางตำแหน่งในภาพ ซึ่งถ้าเราสุ่มเลือกมาละเอียด ภาพที่ได้ก็จะมีรายละเอียดสูง หน่วยของการสุ่มเลือกก็คือจุดหรือพิกเซล (Pixel) นั่นเอง



รูปที่ 2.1 การแปลงภาพ 2 มิติที่ได้ให้เป็นภาพเชิงดิจิทัล โดยการสุ่มเลือกจุดตำแหน่ง

ในการแสดงผลของจอภาพคอมพิวเตอร์นั้น หน่วยของความละเอียดในการแสดงผลนั้น เป็นพิกเซล แต่ในโลกแห่งความเป็นจริง ภาพที่เรามองเห็นด้วยตา ไม่ใช่การเรียงกันของจุด แต่มันเป็นภาพเชิงต่อเนื่อง ไม่สามารถแยกออกไปเป็นทีละจุดได้เลย เพราะฉะนั้น เมื่อภาพนั้นมาอยู่ในคอมพิวเตอร์ ภาพจะต้องถูกปรับให้อยู่ในหน้าจอที่ประกอบด้วยจุดสี ดังนั้นจึงจำเป็นที่จะต้องมีการสุ่มเลือกจุดตำแหน่ง ถ้าเราสุ่มเลือกจุดตำแหน่งถี่มากเท่าใด คุณภาพของภาพที่ได้ก็จะดีขึ้น ดังแสดงในรูปที่ 2.2 จะเห็นได้ว่า ทั้ง 3 ภาพ เป็นภาพเดียวกัน แต่เมื่อทำการสุ่มเลือกจุดตำแหน่งมาไม่เท่ากัน คุณภาพของภาพที่ได้จึงต่างกัน และเมื่อขยายภาพออกมาจะได้ภาพที่มีความคมชัดแตกต่างกัน



รูปที่ 2.2 ความละเอียดของภาพ

ภาพที่ 1 สุ่มเลือกมา 128x128 จุด

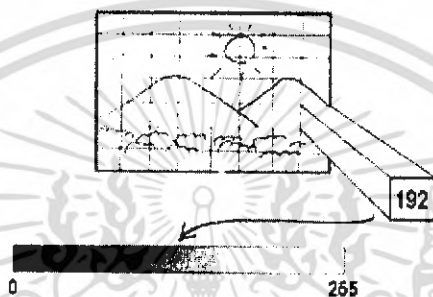
ภาพที่ 2 สุ่มเลือกมา 64x64 จุด

ภาพที่ 3 สุ่มเลือกมา 32x32 จุด

ในการสุ่มเลือกจุดตำแหน่งนี้ ถ้าในระยะความกว้างและความสูงของภาพ เราสุ่มมาละเอียด
 มากๆ เช่น สุ่มมา 128 จุด คุณภาพของภาพก็จะดีขึ้น แต่ก็ต้องใช้เนื้อที่ในการเก็บข้อมูลภาพมากขึ้น
 และขนาดของภาพก็จะมากขึ้นด้วย

2.3.2 การประมาณค่าความเข้มของแสง (Intensity Approximation)

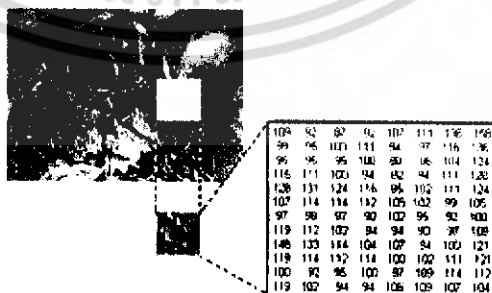
เมื่อเราได้ภาพจากการสุ่มเลือกจุดตำแหน่งมาแล้ว แต่ละจุดในภาพจะถูกแทนด้วยสีในโทน
 สีเทาหรือระดับสีเทา(Grey scale) ซึ่งจะประกอบไปด้วยสีดำ และไล่เฉดสีจางลงไปจนถึงสีขาวดัง
 รูปที่ 2.3



รูปที่ 2.3 ค่าระดับสีเทา (Grey scale) ของจุดใดๆ ในภาพ

ภาพในระดับสีเทานี้มีค่าความละเอียด 8 บิต หรือ มีจำนวนระดับสีทั้งสิ้น 2^8 หรือ 256
 ระดับสี (0-255) โดยที่เลข 8 ก็คือ จำนวนบิตในหน่วยความจำที่ใช้ในการเก็บค่าสีหนึ่งค่า เช่น สีดำ
 จะแทนด้วยค่าตัวเลข 0 หรือในเลขฐานสองคือ 00000000 ส่วนสีขาวจะแทนด้วยค่าตัวเลขคือ 255
 หรือในเลขฐานสองคือ 11111111 ส่วนสีที่อยู่ตรงกลางระหว่างสีดำกับสีขาวก็จะไล่ไปตามลำดับ
 การนับของบิตในเลขฐานสอง

ถ้าภาพเป็นแบบโทนสีเทา แต่ละจุดของภาพก็จะถูกแทนที่ด้วยตัวเลขที่บอกถึงค่าสี ตั้งแต่
 0-255 ดังรูปที่ 2.4



รูปที่ 2.4 จุดของภาพที่ถูกแทนที่ด้วยค่าตัวเลขระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าแต่ละจุดจะถูกแทนที่ด้วยตัวเลข ซึ่งตัวเลขเหล่านี้ก็อยู่ระหว่าง 0-255 แต่ถ้าเป็นภาพขาวดำ จะมีอยู่ด้วยกันเพียง 2 สี คือ สีดำ แทนด้วยเลข 0 และสีขาวแทนด้วยเลข 1 เพราะฉะนั้น ถ้าเป็นภาพขาวดำ หนึ่งจุดภาพจะใช้พื้นที่เก็บข้อมูลเพียง 1 บิตเท่านั้น แต่ถ้าเป็นภาพในโทนสีเทา ใน 1 จุดภาพจะใช้พื้นที่ในการเก็บข้อมูล 8 บิตเพราะค่าระดับสีเทาเมื่อเปลี่ยนเป็นเลขฐานสองแล้ว จะได้ 8 บิต เช่น

- ค่าระดับสี 0 ก็คือ 00000000
- ค่าระดับสี 1 ก็คือ 00000001
- ค่าระดับสี 2 ก็คือ 00000010
- ค่าระดับสี 255 ก็คือ 11111111

ถ้าจำนวนระดับความเข้มของสีในภาพ ใช้จำนวนระดับสีน้อย เช่น 256 ระดับ (8 บิต) หรือต่ำกว่า ผลที่ได้ก็คือการไล่สีของภาพนั้นจะไม่แนบ เป็นรอยไม่เรียบเนียนและไม่น่าดู ดังรูปที่ 2.5



รูปที่ 2.5 การไล่สีของภาพเมื่อใช้จำนวนระดับสีต่างกัน

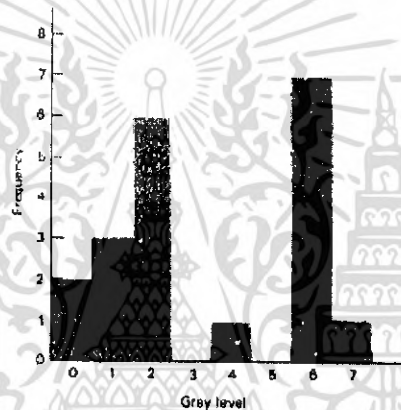
ในความเป็นจริงแล้ว ความละเอียดของภาพจะมีมากกว่านี้ และอัตราในการสุ่มภาพนั้น ถ้าเราไม่คำนึงถึงคุณภาพและรายละเอียดของภาพ แต่คำนึงถึงเนื้อหาของภาพ เราใช้อัตราที่น้อยๆ ก็ได้ เช่น เราต้องการทราบว่ารูปนี้คือรูปอะไรเท่านั้น เพราะฉะนั้น เมื่อเราพูดถึงเนื้อหาของภาพแบบดิจิทัล ก็จะเกี่ยวข้องกับเรื่องของมุมมองเห็นของมนุษย์ด้วย เพราะภาพที่มีความละเอียดต่างกัน ดวงตาของมนุษย์อาจจะแยกแยะไม่ได้ หรือสนใจเพียงแค่เนื้อหา แต่ไม่สนใจในรายละเอียด ดังนั้น ถ้ารูปหนึ่งถูกสุ่มที่ความละเอียดเพียง 1 บิต หรือ 2^1 ระดับสี คือ 0 กับ 1 เราอาจจะดูไม่ออกว่าเป็นรูปอะไร และอาจไม่เพียงพอที่จะนำไปใช้งาน แต่ถ้าเมื่อใดที่ต้องการข้อมูลมากกว่านี้ เช่น เมื่อต้องการทำระบบตรวจจับใบหน้าคนที่ต้องใช้ข้อมูลอื่นๆ ด้วย เมื่อนั้นจึงต้องการความละเอียดของการสุ่มจุดตำแหน่งที่มากกว่า

2.3.3 ฮิสโตแกรม (Histogram)

ฮิสโตแกรมเป็นการแสดงให้เห็นถึงความถี่ของการนับจำนวนพิกเซล ที่มีค่าความเข้มแต่ละค่าหนึ่งๆ ในระดับภาพ แกน x ในกราฟแทนค่าระดับสีเทา และแกน y แทนค่าจำนวนพิกเซลในระดับสีเทานั้นๆ

รูปร่างของฮิสโตแกรมสามารถบอกลักษณะบางประการของภาพได้ เช่น ถ้าฮิสโตแกรมมีลักษณะที่แคบจะหมายถึงการขาดความแตกต่างระหว่างความสว่างและความมืด (Contrast) ในภาพนั้น

ฮิสโตแกรมมีประโยชน์ในการกำหนดระดับเทรชโฮลด์ (Threshold) ซึ่งก็คือการเปลี่ยนภาพระดับสีเทาให้เป็นระบบเลขฐานสอง (Binary) หรือเพื่อปรับปรุงบางส่วนของสเปกตรัม (Spectrum) ของระดับสีเทา



รูปที่ 2.6 ฮิสโตแกรมของระดับสีเทา

2.3.4 การแปลงระดับสีเทาโดยการทำเทรชโฮลด์ (Threshold)

เป็นการเปลี่ยนแปลงหรือหาแนวโน้มของค่าระดับสีเทาในภาพ เพื่อทำให้เป็นแบบดิสครีท (Discrete) มากขึ้น โดยจะนำภาพมาทำฮิสโตแกรมแล้วกำหนดค่าระดับสีเทาที่แน่นอนขึ้นมาเพื่อเป็นจุดตัดทำเทรชโฮลด์ จากนั้นจะทำการตัดหรือปิดบางส่วนของระดับสีเทาที่เราไม่ต้องการออกไป ซึ่งประโยชน์ของเทรชโฮลด์ ได้แก่ การทำให้เป็นภาพแบบไบนารี แต่ทั้งนี้ก็เป็นการทำงานที่กำหนดจุดตัดเทรชโฮลด์ที่ดีที่สุดออกมาได้

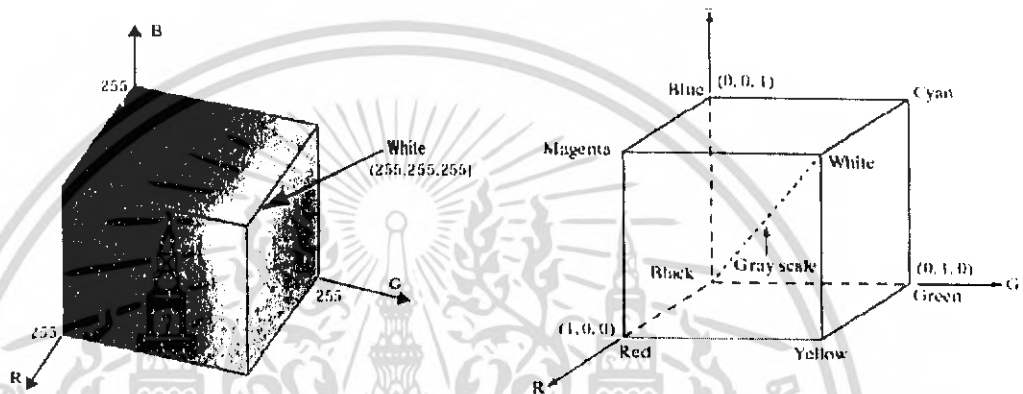
2.4 พื้นฐานและระบบสีของสีโมเดล

จากที่ได้กล่าวไว้ในหัวข้อ 2.1 แล้วว่ารูปภาพ คือ จุดสีหลายๆ จุดที่นำมาเรียงต่อกัน สีที่ว่ามีหลายรูปแบบ โดยแต่ละแบบมีประโยชน์ในการประมวลผลภาพต่างๆ กันไป ซึ่งรูปแบบของสีที่นิยมใช้กันมี 3 รูปแบบ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 RGB

รูปแบบสีแบบ RGB ประกอบไปด้วยสีหลัก 3 สี คือ แดง เขียว และน้ำเงิน ซึ่งรูปแบบสีนี้จะมีโครงสร้างเป็นพิกัดคาร์ทีเซียน (Cartesian Co-ordinate) โดยจะมีลักษณะเป็นทรงลูกบาศก์ มีค่าสีแดง สีเขียว และสีน้ำเงินอยู่ที่มุมทั้งสามที่เป็นแกน และมีสีฟ้าม่วง สีม่วง สีเหลือง อยู่ที่มุมอีกสามมุม สีดำจะอยู่ที่จุดกำเนิด คือถ้าทุกสีมีค่าเป็น 0 คือเป็นสีดำ สีขาวก็คือตรงข้ามกับสีดำอยู่ตรงมุมไกลจากจุดกำเนิดที่สุด ในรูปแบบนี้ค่าระดับสีเทาจะอยู่บนเส้นระหว่างจุดสีดำและจุดสีขาว ส่วนสีอื่นๆ ก็จะมีค่าอยู่ภายในลูกบาศก์นี้

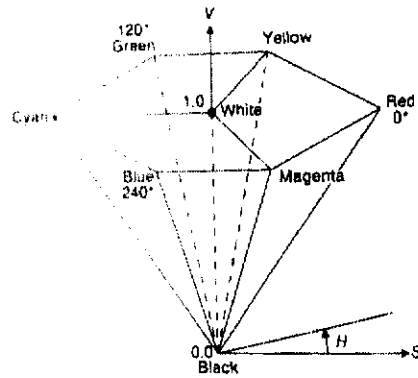


รูปที่ 2.7 รูปแบบสี RGB

รูปแบบสี RGB นี้ ในหนึ่งพิกเซลจะประกอบไปด้วยค่าสี สีละ 8 บิต รวมเป็น 24 บิต คือแต่ละสีจะมีค่าสี 256 ค่า นั่นคือรูปแบบสีจะมีสีได้ถึง 16,777,216 สี รูปแบบสีนี้แต่ละสีจะเป็นอิสระต่อกัน

2.4.2 HSV

รูปแบบสี HSV จะประกอบไปด้วยค่าสี (Hue) คือ การจัดระดับของสี ค่าความอิ่มตัว (Saturation) คือ ค่าความอิ่มตัวของสี ค่าความเข้มหรือความสว่างของสี (Value) ค่าสีจะเป็นมุมรอบแกนตั้ง เริ่มที่สีแดงเปลี่ยนเป็นสีเหลือง สีเขียว สีฟ้าคราม สีน้ำเงิน สีม่วง และวนกลับมาสีแดง ค่าความอิ่มตัวจะมีค่าจาก 0 ไปถึง 1.0 โดยแกนสีเดียวกันในระดับของสีนั้นจะเปลี่ยนค่าจากค่าที่ไม่มีความอิ่มตัวของสี (ระดับสีเทา) ไปยังสีที่มีความอิ่มตัวเต็มที่ ส่วนค่าความเข้มหรือความสว่างก็จะมีค่าจาก 0 ถึง 1.0 เช่นกัน โดยจะเห็นว่าสีนั้นจะมีความสว่างเพิ่มขึ้น



รูปที่ 2.8 รูปแบบสี HSV

2.4.3 YCbCr

รูปแบบสีแบบ YCbCr จะประกอบไปด้วยค่า Y แสดงถึงค่าปริมาณของแสงในการส่องสว่าง (Luminance) และจะมีค่าความแตกต่างของสีคือค่า Cb แสดงถึงค่าความแตกต่างของค่าสีน้ำเงินกับค่าอ้างอิงค่าหนึ่ง และค่า Cr แสดงถึงค่าความแตกต่างของค่าสีแดงกับค่าอ้างอิงอีกค่าหนึ่ง รูปแบบสีแบบนี้ใช้เป็นที่แพร่หลายสำหรับกล้องวิดีโอที่มีการบันทึกภาพแบบดิจิทัล

2.5 ไลบรารีของโปรแกรมที่ใช้ในการระบุตำแหน่งใบหน้าคน

โปรแกรมในการระบุตำแหน่งใบหน้าคนนี้ได้ใช้ไลบรารีที่มีชื่อว่า โอเพ่นซีวี (OpenCV) หรือเรียกเต็มๆว่า Open Computer Vision ซึ่งเป็นไลบรารีที่ประกอบด้วยฟังก์ชันสำเร็จรูปจำนวนมากสำหรับการใช้งานด้านระบบการมองภาพของคอมพิวเตอร์ (Computer Vision) ชั้นสูง การวิเคราะห์ภาพ (Image & Pattern Analysis) และการประมวลผลภาพ (Image Processing) ในการสร้าง Application ชั้นสูง โดยมุ่งเพื่อการใช้งานแบบเรียลไทม์ (Real Time) เป็นหลัก ไลบรารีนี้ถูกพัฒนาขึ้นโดยบริษัทอินเทล ซึ่งมีความสามารถรองรับการใช้งานบนระบบปฏิบัติการ (Operating System) ต่างๆบนเครื่องคอมพิวเตอร์ได้อย่างหลากหลาย เช่น Microsoft Windows 9X, Windows NT, Window XP และสามารถเข้ากับกล้อง Webcam ยี่ห้อใดก็ได้

โอเพ่นซีวีมาพร้อมกับไฮจียูไอ (HighGUI หรือ High Graphic User Interface) ที่ใช้เพื่อการรับภาพจากกล้องวิดีโอและการสร้างหน้าต่างเพื่อแสดงผลภาพ นอกจากนี้ยังสามารถอ่านภาพและจัดเก็บภาพได้หลายรูปแบบ (Format) เช่น Bitmap, Sun Raster, JPEG, TIFF, Portable Network Graphics (.png), Portable image (.pbm, .pgm, .ppm) เป็นต้น

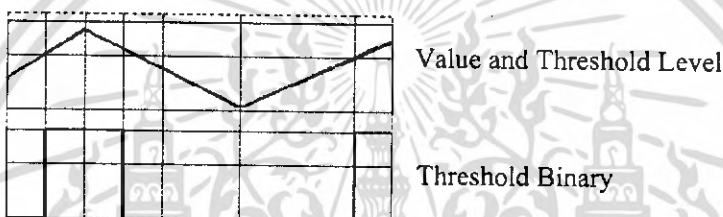
2.5.1 ฟังก์ชันในไลบรารีของโอเพ็นซีวี (Functionality of OpenCV Library)

โอเพ็นซีวีมีฟังก์ชันที่สามารถคำนวณทางคณิตศาสตร์ได้มากกว่า 350 ฟังก์ชัน อีกทั้งยังสามารถกระทำทางเมตริกได้อีกด้วย โดยสามารถแบ่งได้ดังนี้

2.5.1.1 การวิเคราะห์ภาพ (Image Analysis) สามารถทำได้หลายวิธี เช่น

การทำเทรชโฮล (Thresholds)

เป็นการเปลี่ยนแปลงค่าสีเทาในภาพเพื่อทำให้เป็นดิสครีท (Discrete) มากขึ้นหรือทำให้เป็นภาพแบบไบนารี (ขาวดำ) โดยหากค่าสีเทาของจุดใดๆในภาพมีค่ามากกว่าค่าเทรชโฮล จะทำการปัดขึ้นเป็นสีขาว และหากค่าสีเทาของจุดใดๆในภาพมีค่าน้อยกว่าค่าเทรชโฮล จะทำการปัดลงเป็นสีดำ ดังแสดงในรูปที่ 2.9



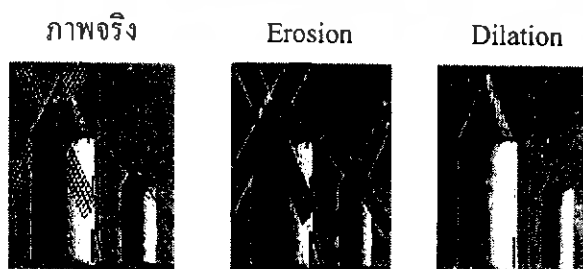
รูปที่ 2.9 การทำเทรชโฮล (Threshold)

การคำนวณทางสถิติ (Statistics)

เป็นการหาค่ามากที่สุด (Max) น้อยที่สุด (Min) ค่าเฉลี่ย (Ave) เป็นต้น

ทฤษฎีมอร์โฟโลยี (Morphology)

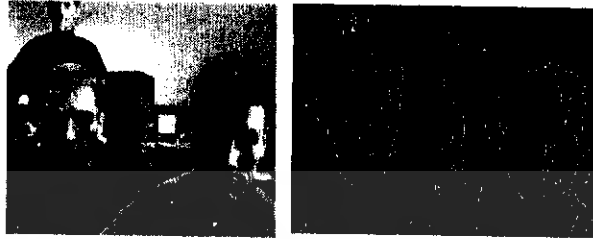
เป็นเทคนิคในการประมวลผลภาพโดยพิจารณาเฉพาะพื้นที่ที่เราต้องการ เช่น การทำอีโรชั่น (Erosion) หรือการหดตัวกลุ่มพิกเซล และไดเลชัน (Dilation) หรือการขยายตัวกลุ่มพิกเซล เป็นต้น



รูปที่ 2.10 ภาพที่เกิดจากการทำ Erosion และ Dilation

4) การหาลักษณะ (Feature Detection)

เป็นการหาลักษณะต่างๆของภาพ เช่น การหาขอบ (Edge) การหาแนวเส้นตรง การหาเค้าโครง (Contour)

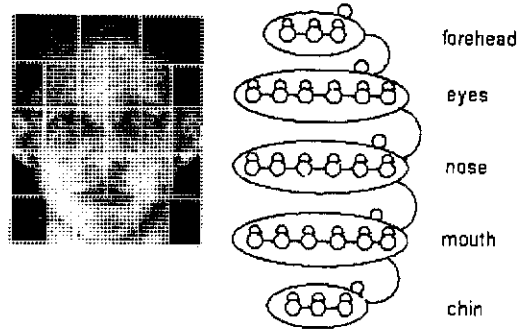


รูปที่ 2.11 การหาขอบของภาพ

2.5.1.2 การระบุตำแหน่งของวัตถุ (Object Recognition)

การระบุตำแหน่งของวัตถุนั้น สามารถใช้ระบุวัตถุใดๆ ก็ได้ หนึ่งในวิธีการระบุตำแหน่งของวัตถุก็คือการทำแบ็คซัพแทรคชัน (Back Subtraction) โดยหลักการทั่วไปก็คือจะต้องให้มีการบันทึกรายละเอียดของวัตถุนั้นๆเสียก่อน จากนั้นจึงทำการเทียบสีของแต่ละพิกเซลของการจับภาพ (Capture) 2 ครั้ง ระหว่างภาพที่ถูกบันทึกไว้ในตอนแรกกับภาพที่รับเข้ามาใหม่ โดยพิกเซลที่มีสีเดียวกันจะปรากฏผลลัพธ์เป็นพิกเซลสีขาว ส่วนพิกเซลที่มีสีต่างกันจะปรากฏผลลัพธ์เป็นพิกเซลสีดำ วิธีนี้เราจะพบว่าส่วนพื้นที่สีขาวก็คือวัตถุที่เราสนใจ การระบุตำแหน่งของวัตถุด้วยวิธีนี้อาจนำไปใช้ระบุใบหน้าคนด้วยก็ได้แต่อาจมีความผิดพลาดมากหากสีของพื้นหลังมีความใกล้เคียงกับสีผิวของใบหน้าคนมาก

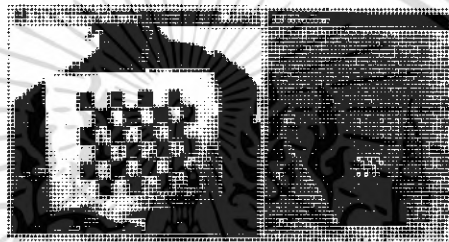
ดังนั้นหากวัตถุที่เราต้องการจะระบุตำแหน่งนั้นคือใบหน้าคน เราสามารถใช้อีกวิธีหนึ่งที่มีความถูกต้องแม่นยำมากกว่าที่เรียกว่า ฮิดเดนมาร์คอฟโมเดล (Hidden Markov Model Algorithms: HMM) ซึ่งวิธีนี้เป็นการเทียบสีและแบบแผน (Pattern) ที่อาจเป็นส่วนของใบหน้าคนกับฐานข้อมูลที่จัดทำโดยบริษัทอินเทล ฐานข้อมูลนี้ได้มีการเก็บรวบรวมรายละเอียดส่วนต่างๆของใบหน้าคนไว้ เช่น หู ตา ปาก จมูก คิ้ว เป็นต้น ทำให้ได้ลักษณะโครงสร้างของอวัยวะต่างๆบนใบหน้าที่แน่นอนไม่ขึ้นกับสีของพื้นหลังเหมือนวิธีแรก และเพื่อเป็นการเพิ่มความเร็วในการประมวลผลส่วนต่างๆจึงใช้วิธีการหาขอบของภาพดังแสดงในรูปที่ 2.12 ร่วมด้วย วิธีการนี้จะทำให้เราสามารถแยกแยะได้ว่าส่วนใดเป็นส่วนประกอบของใบหน้าคนได้ถูกต้องและแม่นยำมากขึ้นโดยไม่จำกัดสีผิว ดังนั้นในการระบุใบหน้าคนนั้นก็สามารถทำได้ง่ายขึ้น เพียงแค่ใช้กลุ่มคำสั่งที่อยู่ในไลบรารีของโอเพ่นซีวีเรียกใช้งานฐานข้อมูลนี้เท่านั้น



รูปที่ 2.12 การทำ HMM

2.5.1.3 การสร้างภาพ 3 มิติ (3D Reconstruction)

เป็นการสร้างภาพ 3 มิติ จากภาพ 2 มิติ



รูปที่ 2.13 การทำ 3D Reconstruction

2.6 การสื่อสารอนุกรมและ RS-232

การรับส่งข้อมูลของคอมพิวเตอร์กับอุปกรณ์ต่อพ่วงอื่นหรือคอมพิวเตอร์นั้นมีอยู่ 2 วิธีด้วยกันคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรม โดยแต่ละวิธีมีรายละเอียดดังนี้

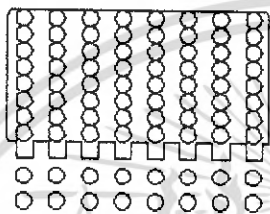
2.6.1 การรับส่งข้อมูลแบบขนาน (Parallel Communication)

การรับส่งข้อมูลแบบขนานจะเป็นการรับหรือส่งข้อมูลคราวละ 4 หรือ 8 บิต หรือสามารถรับส่งข้อมูลได้สูงสุด 8 เครื่องในเวลาเดียวกัน ความเร็วสูงสุดในการรับส่งข้อมูลสูงถึง 1M bps ประกอบกับการรับส่งข้อมูลที่ได้พร้อมๆกันหลายบิต จึงทำให้การรับและส่งข้อมูลทำได้ที่ความเร็วสูง ในระยะทางสูงสุด 3 - 9 เมตร แต่ก็หมายความว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่ส่งด้วย นอกจากนี้ยังจะต้องรวมถึงสายที่ใช้สำหรับการควบคุม และการตรวจสอบการรับส่งข้อมูลอีกด้วย ซึ่งอาจจำเป็นต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตของข้อมูลที่ส่ง ซึ่งถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้ระยะทางของสายมากขึ้น จึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ในการสื่อสาร

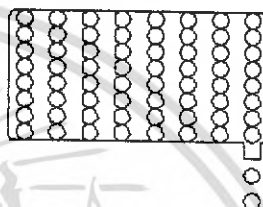
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 การรับส่งข้อมูลแบบอนุกรม (Serial Communication)

การสื่อสารแบบอนุกรมนั้นแบ่งได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาเกี่ยวข้องกับการรับส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา ข้อมูล และกราวด์ การรับส่งข้อมูลแบบอนุกรมนี้นี้มีข้อจำกัดคือสามารถรับส่งข้อมูลได้สูงสุดเพียง 2 เครื่อง แต่รับส่งระยะไกลได้สูงสุด 15-30 เมตร ที่ความเร็วสูงสุด 20 K bps



รูปที่ 2.14 การส่งข้อมูลแบบขนาน

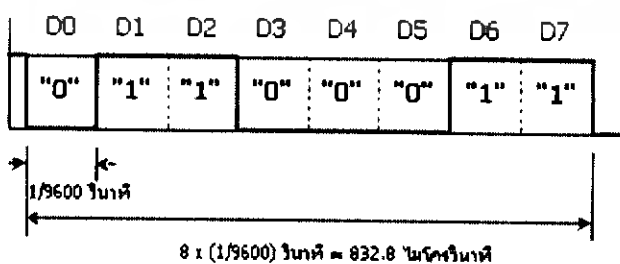


รูปที่ 2.15 การส่งข้อมูลแบบอนุกรม

2.6.2.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส

การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัสไม่จำเป็นต้องมีสัญญาณนาฬิกาเกี่ยวข้องเหมือนการรับส่งแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาหรือความเร็วของทั้งภาครับและภาคส่งให้ค่าเท่ากัน โดยทั่วไปจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที (bps) เช่น ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที ดังแสดงในรูป 2.16 จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ 8 x 104.1 หรือ 832.8 ไมโครวินาที

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



รูปที่ 2.16 การส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

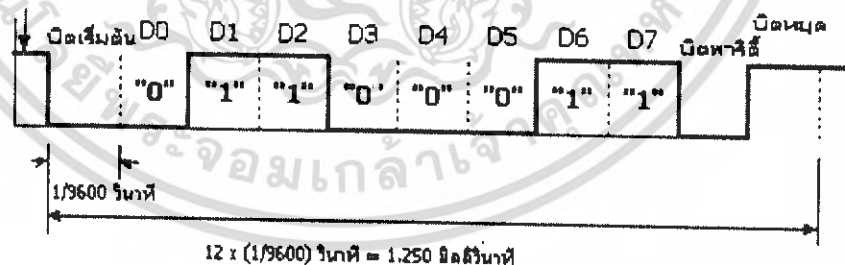
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของการสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส จำเป็นจะต้องเพิ่มบางส่วนเข้าไป เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้นดังแสดงในรูปที่ 2.17 ซึ่งประกอบด้วยกัน 4 ส่วนคือ

1. บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต จะเป็นระดับลอจิกตรงกันข้ามกับระดับลอจิกของสถานะสายสื่อสาร ขณะที่ยังไม่มีการส่งข้อมูล
2. บิตข้อมูล (Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5, 6, 7 หรือ 8 บิตก็ได้
3. บิตแสดงสถานะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิต โดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1" โดยเลือกการส่งข้อมูลเป็นแบบพาริตีคู่หรือ พาริตีคี่ ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตีคู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะทำให้บิตพาริตีนี้เป็น "1" เพื่อจะได้จำนวนเลข "1" เป็นคู่นั่นเอง ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่ได้รับเข้ามาที่เป็น "1" ที่รวมบิตพาริตี 1 บิตด้วย ถ้ามีค่า "1" เป็นจำนวนคู่ แสดงว่าข้อมูลที่ได้รับเข้ามาถูกต้อง อย่างไรก็ตาม สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตีบิตก็ได้
4. บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุดังขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีขนาดมากกว่าหนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0

สถานะของสายสื่อสารเป็น "1"

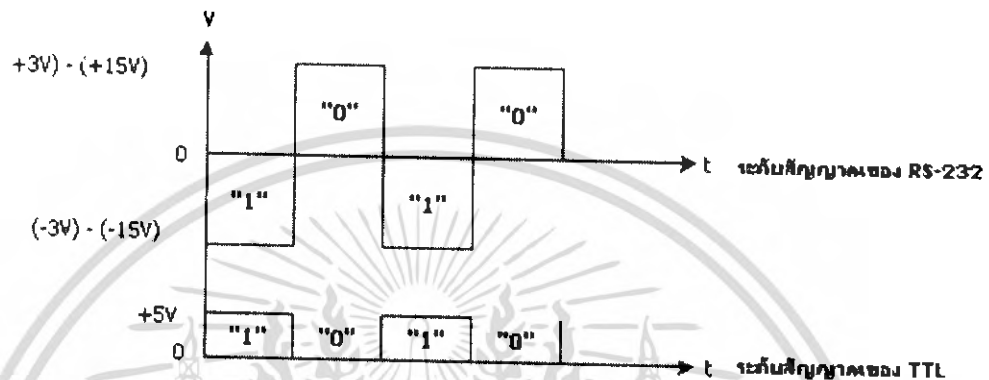


รูปที่ 2.17 การส่งข้อมูลขนาด 8 บิตแบบอนุกรมด้วยความเร็ว 9600 bps

2.6.2.2 การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมและแบบอะซิงโครนัส 2 ทิศทาง โดยกำหนดให้มีระดับสัญญาณตั้งแต่ 3 - 15 V สำหรับลอจิก "0" และ (-3) - (-15) V สำหรับลอจิก "1" ดังนั้นสังเกตได้ว่าจะมีระดับค่าไม่ต่ำกว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิทัลทั่วๆ ไป ซึ่งมีระดับแรงดันอยู่ที่ 5V ดังแสดงในรูปที่ 2.18 การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันไฟฟ้าจาก 0 V – 5 V ของไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า (+3V) หรือต่ำกว่า (-3V) โดยอาจจะต่อวงจรจากทรานซิสเตอร์ได้หรือใช้ไอซีสำเร็จรูปเบอร์ MAX-232 ซึ่งจะขอล่าถึงในหัวข้อ 2.7



รูปที่ 2.18 ระดับแรงดันของสัญญาณอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"

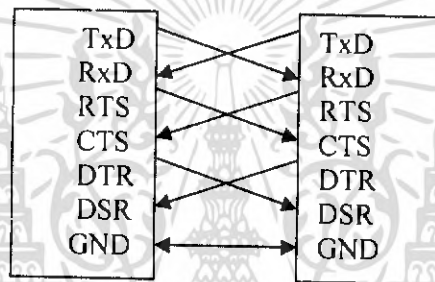
มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 ถูกออกแบบมาเพื่อใช้ในงานที่มีการเชื่อมต่อข้อมูลที่สอดคล้องกันระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ โดยได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232 ข้อแตกต่างของอุปกรณ์ DTE และ DCE อย่างหนึ่งที่ได้ชัดคือ คอนเน็คเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็คเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็คเตอร์ของโมเด็มจะเป็นแบบ DCE มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 จะใช้คอนเน็คเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ โดยคอนเน็คเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่นที่เคยใช้งานในอดีตปัจจุบันมีการใช้งานไม่มากนัก โดยแสดงรูปร่างและตำแหน่งดังรูปที่ 2.19 (ก) และ (ข)



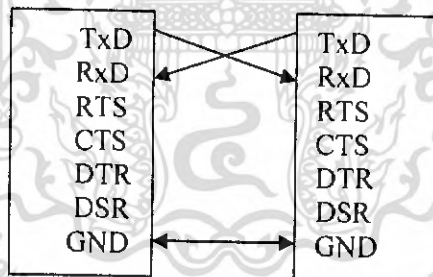
(ก) คอนเน็คเตอร์อนุกรม 9 ขา (DB-9) (ข) คอนเน็คเตอร์อนุกรม 25 ขา (DB-25)

รูปที่ 2.19 คอนเน็คเตอร์แบบ DB-9 และ DB-25

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกในรูปที่ 2.20 นั้น ลูกศรแสดงถึงทิศทางการส่งข้อมูล ในรูปที่ 2.20 (ก) เป็นการเชื่อมต่อแบบ Null Modem หรือ การเชื่อมต่อโดยตรงโดยไม่ต้องผ่าน โมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนรูปที่ 2.20 (ข) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งรับข้อมูลและเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดดังนี้



(ก) การเชื่อมต่ออุปกรณ์ภายนอกแบบ Null Modem



(ข) การเชื่อมต่อแบบ Null Modem โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 2.20 การเชื่อมต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

ถึงแม้ว่ารูปแบบของคอนเน็คเตอร์จะมี 2 แบบ แต่การใช้งานก็มีลักษณะคล้ายกัน ต่างกันเพียงที่ตำแหน่งของสายสัญญาณ ดังแสดงตารางที่ 2.1

62353

คอนเน็คเตอร์แบบ	คอนเน็คเตอร์แบบ	ชื่อของสายสัญญาณ	ชนิดของ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB-9	DB-25		สายสัญญาณ
1	8	Data Carrier Detect : DCD	Input
2	3	Received Data : RxD	Input
3	2	Transmitted Data : TxD	Output
4	20	Data Terminal Ready : DTR	Output
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	Input
7	4	Request To Send : RTS	Output
8	5	Clear To Send : CTS	Input
9	22	Ring Indicator : RI	Input

ตารางที่ 2.1 การจัดขาของคอนเน็คเตอร์อนุกรมแบบมาตรฐาน RS-232 แบบ DB-9 และ DB-25

Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็มสำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

Receive Data : RD หรือ **RxD** ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลทีอ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

Transmitted Data : TD หรือ **TxD** ใช้เพื่อส่งข้อมูลจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลเป็นตัวส่งข้อมูลออกไป

Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น จะต้องเชื่อมต่อกับขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้มีการตรวจจับพาห้จะต้องต่อกับขา DCD ด้วย

Signal Ground : GND กราวด์ของระบบ

Data Set Ready : DSR ใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DSR

Request To Send : RTS ใช้สำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS แต่ในกรณีที่ใช้ในการเชื่อมต่อแบบ Null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับส่งข้อมูลเกิดขึ้นได้ตลอดเวลา

Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่รับข้อมูลหรือไม่

Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารทั่วไปจะไม่ถูกใช้งาน จะใช้ก็ต่อเมื่อมีการเชื่อมต่อโมเด็มกับ โปรแกรมที่มีการตรวจสอบสัญญาณนี้เท่านั้น

2.6.2.3 ยูอาร์ที (UART)

UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับส่งข้อมูลแบบอะซิงโครนัสนั่นเอง UART ถือเป็นหัวใจสำคัญสำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์

หน้าที่หลักของ UART คือ ทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัสแล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนกลับเข้ามายังคอมพิวเตอร์เป็นแบบขนาน ก่อนส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้วยังแจ้งข้อมูลอื่นให้คอมพิวเตอร์ทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (Baud Rate) รูปแบบการส่งข้อมูล ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี เฟรมข้อมูล โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอเดอเรตแบบโปรแกรมได้ (Programmable Baud Rate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต สามารถส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex) และฟูลดูเพล็กซ์ (Full Duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งได้พร้อมๆกัน

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 5706 กิโลบิตต่อวินาที แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ด้วยความเร็ว 115200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิพรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์

TL 16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นประโยชน์ในการนำเอกสารนี้ไปใช้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมคประหยัดพลังงานสามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มาจากของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.6.2.4 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM1 COM2 COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

การทำงานภายในของพอร์ตอนุกรมประกอบด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอแดปเตอร์ของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม เช่น พอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

รีจิสเตอร์ตำแหน่ง 00H : รีจิสเตอร์บัพเฟอร์

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่ได้รับเข้ามาและส่งออกไป โดยการติดต่อกับรีจิสเตอร์นี้ทำเพื่อเก็บข้อมูลที่ต้องการจะส่งออกไปจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูลและข้อมูลจะส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อรับข้อมูลเข้ามาเรียบร้อยแล้วและแปลงเป็นขนาานแล้ว ข้อมูลจะถูกส่งมายังรีจิสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจิสเตอร์ออกไป รีจิสเตอร์นี้จะถูกเคลียร์และเตรียมพร้อมสำหรับการรับข้อมูลในไปต์ต่อไป

รีจิสเตอร์ตำแหน่ง 01H : รีจิสเตอร์เอ็นเอเบิลการอินเทอร์รัพต์

เป็นรีจิสเตอร์สำหรับการเอ็นเอเบิลการอินเทอร์รัพต์ ซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเทอร์รัพต์ขึ้นมา ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์นี้มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	SINP	ERBK	TBE	RxD

ตารางที่ 2.2 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์เอ็นเอเบิลการอินเทอร์รัพต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 4 – บิต 7	ไม่ถูกใช้งาน
SINP	เอ็นเอเบิลการอินเทอร์รัพต์เนื่องจากเกิดการเปลี่ยนแปลงสถานะที่ขาอินพุต (CTS, DSR, DCD, RI) โดย “1” คือเอ็นเอเบิลการอินเทอร์รัพต์ และ “0” คือดีสเอเบิลการอินเทอร์รัพต์
ERBK	เอ็นเอเบิลการอินเทอร์รัพต์ เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจากพาริตี, โอเวอร์รัน, เฟรมข้อมูลหรือเบรกข้อมูล โดย “1” คือเอ็นเอเบิลการอินเทอร์รัพต์ และ “0” คือดีสเอเบิลการอินเทอร์รัพต์
TBE	เอ็นเอเบิลการอินเทอร์รัพต์ เมื่อบัฟเฟอร์สำหรับส่งข้อมูลว่าง โดย “1” คือ เอ็นเอเบิลการอินเทอร์รัพต์ และ “0” คือดีสเอเบิลการอินเทอร์รัพต์
RxD	เอ็นเอเบิลการอินเทอร์รัพต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ได้รับข้อมูลเรียบร้อยแล้ว โดย “1” คือเอ็นเอเบิลการอินเทอร์รัพต์ และ “0” คือดีสเอเบิลการอินเทอร์รัพต์

รีจิสเตอร์ตำแหน่ง 02H : รีจิสเตอร์แสดงโหมดและสถานะการอินเทอร์รัพต์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	ID1	ID0	PND

ตารางที่ 2.3 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์แสดงโหมดและสถานะการอินเทอร์รัพต์

บิต 3 – บิต 7	ไม่ถูกใช้งาน
ID0, ID1	ใช้งานร่วมกันเพื่อแจ้งสาเหตุการเกิดอินเทอร์รัพต์ “00” เกิดอินเทอร์รัพต์ เนื่องจากการเปลี่ยนแปลงของขาอินพุต ซึ่งมีนัยสำคัญเป็นอันดับ 4 “01” เกิดอินเทอร์รัพต์ เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่าง ซึ่งมีนัยสำคัญเป็นอันดับ 3 “10” เกิดอินเทอร์รัพต์ เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเรียบร้อยแล้ว ซึ่งมีนัยสำคัญเป็นอันดับ 2 “11” เกิดอินเทอร์รัพต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรก ซึ่งมีนัยสำคัญเป็นสูงสุดเป็นอันดับ 1
PND	ใช้แสดงสถานะของการเกิดอินเทอร์รัพต์ โดย “1” คือเอ็นเอเบิลการอินเทอร์รัพต์ และ “0” คือดีสเอเบิลการอินเทอร์รัพต์

เมื่อมีการสร้างสัญญาณอินเทอร์รัพต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดการอินเทอร์รัพต์ในครั้งต่อไปโดยสามารถทำได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- * ถ้าเกิดอินเทอร์รัพต์ เนื่องจากการเปลี่ยนแปลงของขาอินพุต จะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเทอร์รัพต์
- * ถ้าเกิดอินเทอร์รัพต์ เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่าง จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ส่งข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเทอร์รัพต์
- * ถ้าเกิดอินเทอร์รัพต์ เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเรียบร้อยแล้ว จะต้องเคลียร์ค่าอินเทอร์รัพต์โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์
- * ถ้าเกิดอินเทอร์รัพต์ เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรกจะต้องเคลียร์ค่าอินเทอร์รัพต์โดยการอ่านค่าจากรีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม

รีจิสเตอร์ตำแหน่ง 03H : รีจิสเตอร์กำหนดรูปแบบของข้อมูล

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0

ตารางที่ 2.4 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์กำหนดรูปแบบของข้อมูล

DLAB	ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัฟเฟอร์ (00H) “1” เป็นการเข้าสู่โหมดการหารค่าบอดเรต “0” เป็นการเข้าถึงรีจิสเตอร์บัฟเฟอร์ (รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับเอ็นเอเบิลการอินเทอร์รัพต์ (รีจิสเตอร์ตำแหน่ง 01H) เมื่อบิต DLAB เป็น “1” รีจิสเตอร์บัฟเฟอร์ (00H) และรีจิสเตอร์เอ็นเอเบิลการอินเทอร์รัพต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนดค่าบอดเรตโดยรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหารค่าบอดเรตสามารถเขียนเป็นสมการได้ว่า บอดเรต = $115200 / \text{ค่าตัวหาร}$ 16 บิต โดยตัวเลข 115200 คือ ความถี่ของคริสตอลในวงจร UART ซึ่งทำการหารค่าความถี่ภายในเครื่องคอมพิวเตอร์ที่ใช้คริสตอลความถี่ 1.8432 MHz ด้วย 16 ทำให้ได้ความถี่ 115200 Hz และค่าตัวหาร 16 บิต คือ ข้อมูลในรีจิสเตอร์ 00H + (256 x ข้อมูลในรีจิสเตอร์ 01H) สมมติว่าต้องการค่าบอดเรตสูงสุดที่ 9600 บิตต่อวินาที ค่าตัวหารที่ต้องใช้ต้องมีค่าเป็น 12 ซึ่งค่านี้จะต้องถูกโหลดลงในรีจิสเตอร์ 00H และโหลดค่า 0 ลงไปในรีจิสเตอร์ 01H
BRK	ใช้ควบคุมการหยุดถ่ายทอดข้อมูล โดย “1” สามารถหยุดหรือเบรกได้ และ “0” ไม่มีการหยุดหรือเบรกได้
PAR0 - PAR2	ใช้เพื่อกำหนดพาริตี “000” ไม่ใช่พาริตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	“001” กำหนดพาริตีคู่
	“011” กำหนดพาริตีคู่
	“101” มาร์ก (Mark)
	“111” ช่องว่าง (Space)
STOP	กำหนดจำนวนบิตปิดท้าย โดย “1” มีบิตปิดท้าย 2 บิต และ “0” มีบิตปิดท้าย 1 บิต
DAB1, DAB0	ใช้ร่วมกันในการกำหนดจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด
	“00” จำนวนบิตข้อมูลเท่ากับ 5 บิต
	“01” จำนวนบิตข้อมูลเท่ากับ 6 บิต
	“10” จำนวนบิตข้อมูลเท่ากับ 7 บิต
	“11” จำนวนบิตข้อมูลเท่ากับ 8 บิต

รีจิสเตอร์ตำแหน่ง 04H : รีจิสเตอร์ควบคุมโมเด็ม

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

ตารางที่ 2.5 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์ควบคุมโมเด็ม

บิต 5 – บิต 7	ไม่ถูกใช้งาน
LOOP	“1” เอ็นเอเบิลการส่งค่ากลับ และ “0” ดิสเอเบิล
OUT1, OUT2	“1” เอ็นเอเบิลการใช้งานภายใน และ “0” ดิสเอเบิล
RTS	ใช้ควบคุมการทำงานของขา RTS โดย “1” เอ็นเอเบิลและ “0” ดิสเอเบิล
DTR	ใช้ควบคุมการทำงานของขา DTR โดย “1” เอ็นเอเบิลและ “0” ดิสเอเบิล

รีจิสเตอร์ตำแหน่ง 05H : รีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลอนุกรมของ UART

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเทอร์รัพต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อแสดงสาเหตุของการเกิดการอินเทอร์รัพต์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	TXE	TBE	BREK	FRME	PARE	OVFE	RxD

ตารางที่ 2.6 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์ตำแหน่ง 05H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TXE (Transmitter Empty)	“1” รีจิสเตอร์สำหรับส่งข้อมูลว่าง “0” ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์สำหรับส่งข้อมูล
TBE (Transmitter Buffer Empty)	“1” รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง “0” ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล
BREK (Break)	“1” UART ตรวจพบการเบรก “0” ไม่มีการเบรก
FRME (Frame Error)	“1” UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล “0” ไม่พบความผิดพลาดด้านเฟรมข้อมูล
PARE (Parity Error)	“1” UART ตรวจพบความผิดพลาดทางพาริตี “0” ไม่พบความผิดพลาดทางพาริตี
OVRE (Overrun Error)	“1” UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน “0” ไม่พบความผิดพลาดแบบโอเวอร์รัน
RxD (Received Data Ready)	“1” มีการรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์ “0” ไม่มีข้อมูล

รีจิสเตอร์ตำแหน่ง 06H : รีจิสเตอร์แสดงสถานะของโมเด็ม

ใช้เพื่อแสดงสถานะสัญญาณอินพุตของพอร์ตอนุกรม RS-232 ซึ่งได้แก่ สัญญาณ DCD, DSR, CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบเอนกประสงค์ มีรายละเอียดดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

ตารางที่ 2.7 ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์แสดงสถานะของโมเด็ม

DCD (Data Carrier Detect)	ใช้แสดงสถานะของขา DCD
RI (Ringing Indicator)	ใช้แสดงสถานะของขา RI
DSR (Data Set Ready)	ใช้แสดงสถานะของขา DSR
CTS Delta Clear To Send)	ใช้แสดงสถานะของขา CTS
DDCD (Delta Data Carrier Detect)	ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DCD “1” แสดงว่าบิต DCD เกิดการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว “0” ไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว
DRI (Delta Ring Indicator)	ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“1” แสดงว่าบิต RI เกิดการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

“0” ไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDSR (Delta Data Set Ready) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DSR

“1” แสดงว่าบิต DSR เกิดการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

“0” ไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DCTS (Delta Clear To Send) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

“1” แสดงว่าบิต CTS เกิดการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

“0” ไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

รีจิสเตอร์ตำแหน่ง 07H : รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว

ทำหน้าที่เป็นหน่วยความจำ RAM ขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์นี้ไม่ส่งผลใดๆ ต่อการทำงาน UART

2.6.2.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบ TTL เท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับ เพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับการเชื่อมต่อปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณระดับเดียวกัน แฉ่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

2.6.2.6 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 3E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM1 = 0000:0400H - 0000:0401H

COM2 = 0000:0402H - 0000:0403H

COM3 = 0000:0404H - 0000:0405H

COM4 = 0000:0406H - 0000:0408H

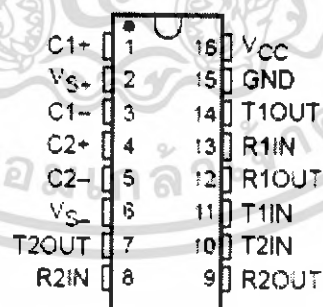
นอกจากนี้ที่ 0000:0411H ยังใช้แสดงจำนวนของพอร์ตอนุกรมที่ใช้อยู่ในคอมพิวเตอร์ ดังนี้

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

ตารางที่ 2.8 ข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

2.7 ไอซี MAX232

ไอซี MAX232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้ ตำแหน่งของขาต่างๆ ของ MAX 232 แสดงในรูปที่ 2.21

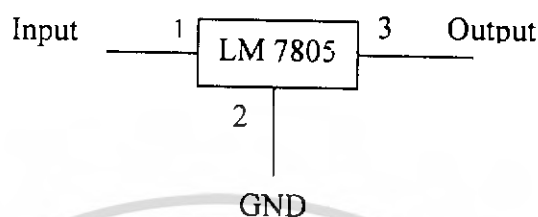


รูปที่ 2.21 ตำแหน่งขาต่างๆ ของ MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ตัวกำเนิดแรงดัน 5V จาก 12V

เนื่องจากแหล่งจ่ายแรงดันไฟที่ใช้มีระดับ 12V แต่ IC ใช้แรงดัน 5V จึงต้องมีการแปลงแรงดันจาก 12V เป็น 5V โดยใช้ IC เบอร์ LM 7805 ซึ่งสามารถใช้งานได้ตามรูปที่ 2.22



รูปที่ 2.22 ขาต่างๆ ของ LM 7805

2.9 โครงสร้างและสถาปัตยกรรมของ PIC 16F877A

การคิดค้นและพัฒนาไมโครคอนโทรลเลอร์ได้มีมาอย่างต่อเนื่อง จนกระทั่งในปัจจุบันนี้ ไมโครคอนโทรลเลอร์ตระกูลใหม่เกิดขึ้นมากมาย ซึ่งล้วนแล้วแต่มีศักยภาพในการทำงานสูง โดยในโครงงานนี้ได้ทำการศึกษาการใช้งานไมโครคอนโทรลเลอร์ตระกูล PIC ผลิตโดยบริษัทไมโครชิพ ซึ่งได้รับความนิยมเป็นอย่างมากในประเทศไทย เนื่องจากเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถและเพียบพร้อมไปด้วยทรัพยากรหรือฟังก์ชันการใช้งานต่างๆ มากมาย เช่น โมดูล Analog to Digital, Timer/Counter, USART, SPI, I²C, PWM และอื่นๆ ซึ่งส่วนต่างๆ เหล่านี้จะถูกสร้างรวมอยู่ใน CPU เพียงตัวเดียว ทำให้ CPU ตัวนี้สามารถทำงานได้หลายๆ อย่างและสามารถลดในส่วนของฮาร์ดแวร์บางอย่างลง ส่วนในเรื่องของความเร็วนั้น CPU ตระกูลนี้จะใช้เวลาในการกระทำคำสั่งต่างๆ เพียง 1 หรือ 2 ไชเคลต ต่อคำสั่งเท่านั้น โดยการทำงานนี้จะเป็นลักษณะไปป์ไลน์ (Pipe Line) ทำให้มีความเร็วในการทำงานมากกว่า CPU ทั่วไปที่ความถี่เดียวกัน

2.9.1 คุณสมบัติของ PIC 16F877A

คุณสมบัติต่างๆ ของไมโครคอนโทรลเลอร์ PIC 16F877 สามารถสรุปคร่าวๆ ได้ดังนี้

- ในการปฏิบัติงานคำสั่งต่างๆ จะใช้ 1 ไชเคลตเดียว และ 2 ไชเคลตในคำสั่งที่เป็นการกระโดด
- ความถี่สูงสุดที่ทำงานได้คือ 20MHz (16F877-20/P)
- การทำงานจะเป็นลักษณะไปป์ไลน์ทำให้มีการทำงานที่เร็วขึ้น
- หน่วยความจำข้อมูล (RAM) 368 Bytes
- หน่วยความจำข้อมูล (EEPROM) 256 Bytes
- สามารถตอบสนองการอินเทอร์รัพได้ถึง 14 แหล่ง

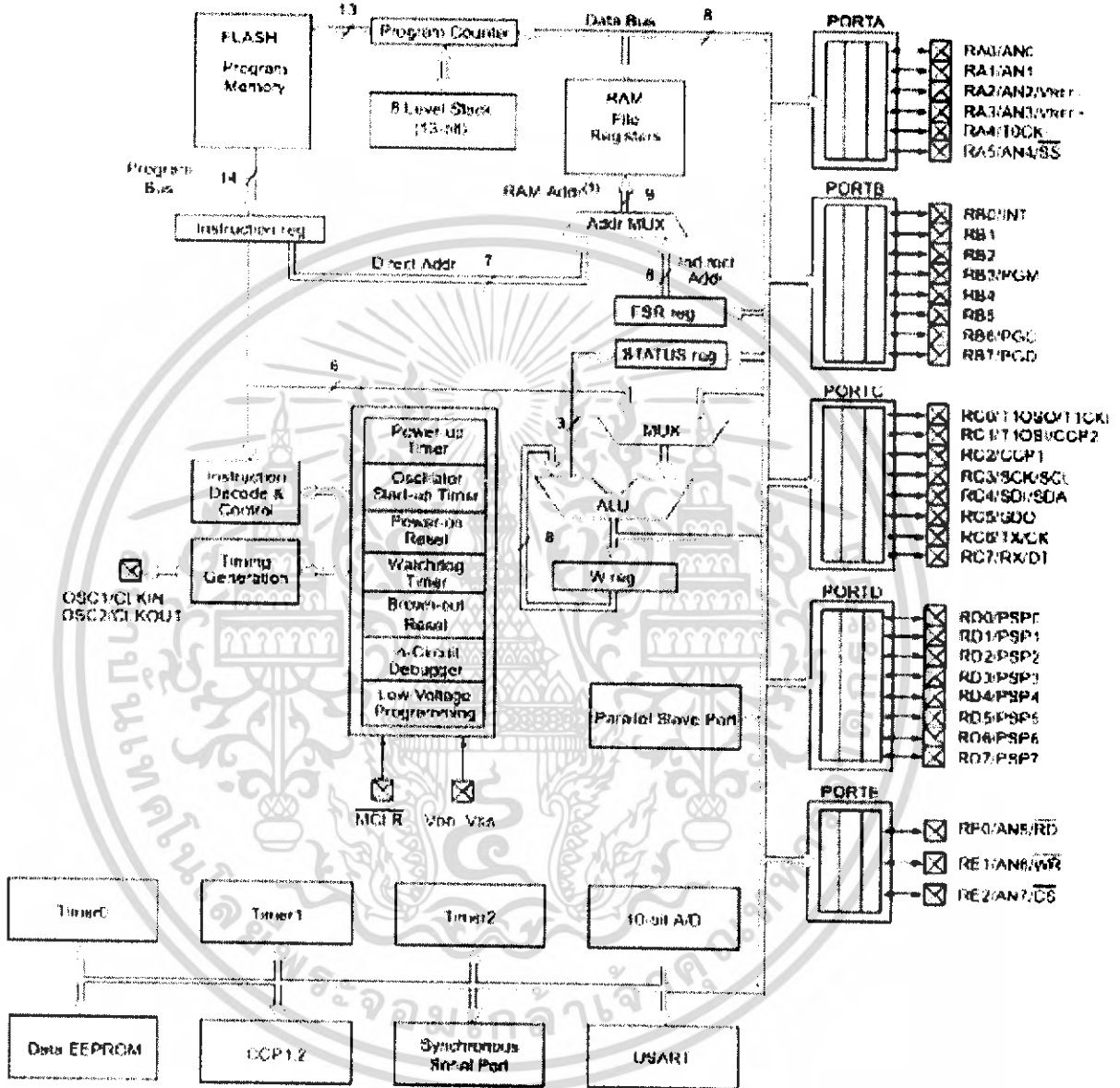
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Stack 8 ระดับ
- เพาเวอร์อนรีเซต (POR), เพาเวอร์อัปไทมเมอร์ (PWRT) และ Oscillator Start-Up Timer
- Watchdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- โหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของสัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- ฟังก์ชันการ โปรแกรมแบบ ICSP (In-Circuit Serial Programming)
- ทำงานที่ไฟเลี้ยง 2V ถึง 5.5V
- กระแสซิงก์และซอร์สของพอร์ตคือ 25mA
- Timer/Counter จำนวน 3 ตัว คือ Timer0, Timer1 และ Timer2
- โมดูล Capture/Compare/PWM จำนวน 2 ชุด
- Analog to Digital Converter ความละเอียด 10 บิต 8 แชนแนล ภายในตัว
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับแรงดันไฟเลี้ยง Brown – out reset (BOR)
- มีพอร์ต I/O 5 พอร์ต ประกอบด้วย A, B, C, D และ E แต่ละพอร์ตจะมีจำนวนบิตไม่เท่ากันซึ่งรวมแล้วจะมี I/O จำนวน 33 บิต
 - PORTA = RA5 – RA0 จำนวน 6 บิต
 - PORTB = RB7 – RB0 จำนวน 8 บิต
 - PORTC = RC7 – RC0 จำนวน 8 บิต
 - PORTD = RD7 – RD0 จำนวน 8 บิต
 - PORTE = RE2 – RE0 จำนวน 3 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 สถาปัตยกรรมฮาร์ดแวร์ PIC16F877A

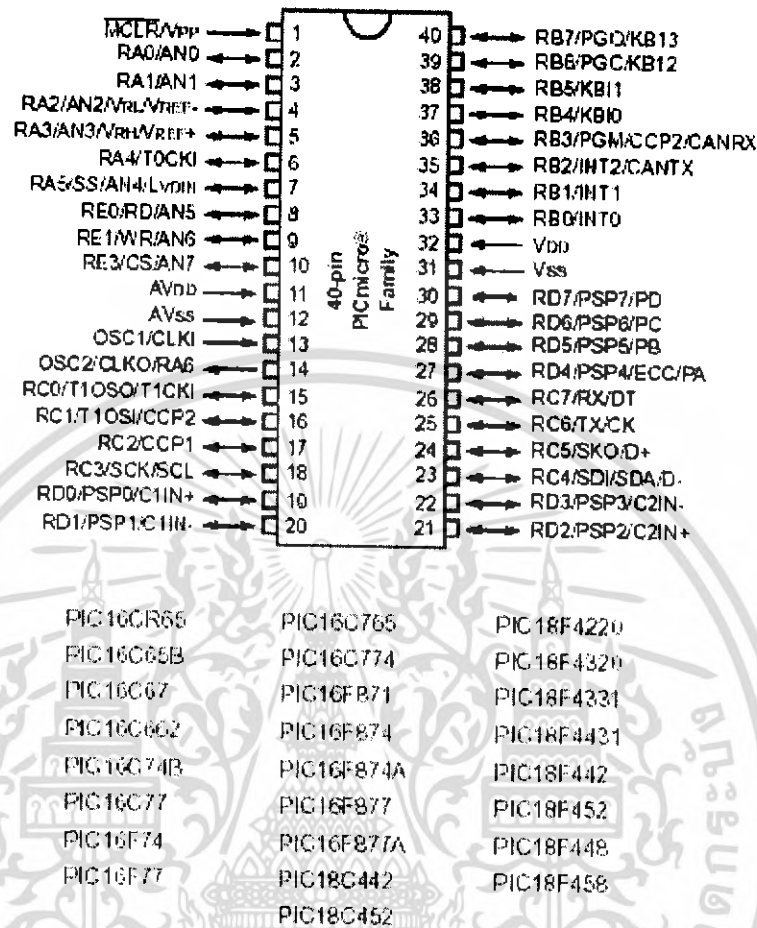
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: High-order bits are from the STATUS register.

รูปที่ 2.23 สถาปัตยกรรมภายในของ PIC 16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

40-pin PICmicro[®] MCU Family

รูปที่ 2.24 การจัดวางตำแหน่งขาสัญญาณต่างๆ ของ PIC 16F877A และประเภทใกล้เคียง

รูปที่ 2.24 แสดงการจัดวางตำแหน่งขาสัญญาณต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล PIC ซึ่งมีเบอร์ต่างๆ ให้เลือกใช้หลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์มีคุณสมบัติของแต่ละขาสัญญาณใกล้เคียงกัน

ขาสัญญาณของ PIC 16F877 นี้จะมีทั้งหมด 40 ขา ประกอบไปด้วยขาที่ทำหน้าที่ต่างๆ โดยจะมีขาสัญญาณ I/O Ports ทั้งหมดจำนวน 33 ขา สามารถนำไปใช้ป็นอินพุต/เอาต์พุตได้ทั้งหมดทุกขา ยกเว้นขา RA4 ซึ่งโครงสร้างภายในเป็นแบบโอเพินเดรน (Open Drain) ดังนั้นหากต้องการนำไปใช้ป็นขาสัญญาณเอาต์พุต จะต้องต่อตัวต้านทานพูลอัพ (Pull - up) ไว้ด้วย ส่วนขาที่เหลือสามารถใช้งานได้ตามปกติ นอกจากขาสัญญาณ I/O แล้ว ยังประกอบด้วยขาสัญญาณอื่นๆ อีก คือ ขาไฟเลี้ยง ขากราวด์ ขารีเซ็ต และ ขาออสซิลเลเตอร์ ซึ่งสามารถสรุปหน้าที่การทำงานของขาสัญญาณในแต่ละขาได้ดังตารางที่ 2.9

ขาสัญญาณ	DIP PIN	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	I	ST/CMOS	ขาสัญญาณอินพุตสัญญาณนาฬิกาของ CPU
OSC2/CLKOUT	14	O	-	เป็นขาเอาต์พุตของสัญญาณนาฬิกา (1/4 ของ CLKIN) , ใช้คู่ร่วมกับขาสัญญาณ OSC1 เพื่อกำเนิดสัญญาณนาฬิกาในกรณีที่ใช้คริสตอลแบบเรโซเนเตอร์ หรือ วงจร RC ภายนอก
MCLR/VPP	1	I/P	ST	ขาสัญญาณรีเซ็ตแอกทีฟ "0" , ขารับแรงดันสำหรับการโปรแกรม
RA0/AN0	2	I/O	TTL	ขาสัญญาณอินพุต/เอาต์พุต PORTA -ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN0
RA1/AN1	3	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN1
RA2/AN2/VREF-	4	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN2 และขาสัญญาณแรงดันอ้างอิงลบ (-)
RA2/AN2/VREF+	5	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN3 และขาสัญญาณแรงดันอ้างอิงบวก (+)
RA4/T0CKI	6	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN4 และขาสัญญาณ Slave Select ในโหมดการสื่อสารแบบ Synchronous Serial Port
RA5/SS/AN4	7	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณนาฬิกาของไทม์เมอร์ 0 -ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณอนาล็อก AN4 และขาสัญญาณ Slave Select ในโหมดการสื่อสารแบบ Synchronous Serial Port
RB0/INT	33	I/O	TTL/ST	ขาสัญญาณอินพุต/เอาต์พุต PORTB -ขาสัญญาณอินพุต/เอาต์พุต, ขาอินพุตสัญญาณอินเทอร์รัพท์ภายนอก
RB1	34	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต
RB2	35	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต
RB3/PGM	36	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาระดับมัธยมศึกษาตอนต้นเท่านั้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RB4	37	I/O	TTL	โปรแกรมแบบแรงดันต่ำ -ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินเทอร์รัพท์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ
RB5	38	I/O	TTL	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินเทอร์รัพท์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ
RB6/PGC	39	I/O	TTL/ST ⁽²⁾	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินเทอร์รัพท์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ
RB7/PGD	40	I/O	TTL/ST ⁽²⁾	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินเทอร์รัพท์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ และเป็นขาสัญญาณนาฬิกาในโหมดการโปรแกรม -ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินเทอร์รัพท์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ และเป็นขาสัญญาณข้อมูลในโหมดการโปรแกรม
RC0/TIOSO/TICK 1	33	I/O	ST	ขาสัญญาณอินพุต/เอาต์พุต PORTC -ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณออสซิลเลเตอร์เอาต์พุต และขาสัญญาณนาฬิกาของไทม์เมอร์ 1
RC1/TIOS1/CCP2	16	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตออสซิลเลเตอร์ไทม์เมอร์ 1 และเป็นขาสัญญาณ Capture 2 input/Compare 2 output/PWM2 output
RC2/CCP1	17	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต และขาสัญญาณ Capture1 input/Compare 1 output/PWM1 output
RC3/SCK/SCL	18	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณอินพุตสัญญาณนาฬิกาในการสื่อสาร Synchronous, ขาสัญญาณนาฬิกาในโหมด I ² C และ SPI
RC4/SD1/SDA	23	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณข้อมูลในโหมด SPI, ขาอินพุต/เอาต์พุตสัญญาณข้อมูลในโหมด I ² C
RC5/SDO	24	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณข้อมูลในโหมด SPI
RC6/TX/CK	25	I/O	ST	-ขาสัญญาณอินพุต/เอาต์พุต, ขาเอาต์พุตสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RC7/RX/DT	26	I/O	ST	ด้านส่งของการสื่อสาร USART, ขาสัญญาณนาฬิกา ในโหมดการสื่อสาร Synchronous - ขาสัญญาณอินพุต/เอาต์พุต, ขาเอาต์พุตสัญญาณ ด้านรับของการสื่อสาร USART, ขาสัญญาณนาฬิกา ในโหมดการสื่อสาร Synchronous
RD0/PSP0	19	I/O	ST/TTL ⁽³⁾	ขาสัญญาณอินพุต/เอาต์พุต PORTD มาสารณใช้เป็น Slave Port กรณีติดต่อกับระบบบัสของ ไมโครโปรเซสเซอร์ - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต - ขาสัญญาณอินพุต/เอาต์พุต
RD1/PSP1	20	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	I/O	ST/TTL ⁽³⁾	
RE0/RD/AN5	8	I/O	ST/TTL ⁽³⁾	ขาสัญญาณอินพุต/เอาต์พุต PORTE - ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณควบคุมการ อ่านในโหมด Parallel Slave Port และเป็น ขาสัญญาณอินพุตสัญญาณอนาล็อก AN5 - ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณควบคุมการ เขียนในการติดต่อแบบ Parallel Slave Port และเป็น ขาสัญญาณอินพุตสัญญาณอนาล็อก AN6 - ขาสัญญาณอินพุต/เอาต์พุต, ขาสัญญาณควบคุม Control Select ในการติดต่อแบบ Parallel Slave Port และเป็นขาสัญญาณอินพุตสัญญาณอนาล็อก AN7
RE1 WR/AN6	9	I/O	ST/TTL ⁽³⁾	
RE2/CS/AN7	10	I/O	ST/TTL ⁽³⁾	
V _{SS}	12,31	P	-	ขาสัญญาณกราวด์
V _{DD}	11,32	P	-	ขาแรงดันไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I = อินพุต O = เอาต์พุต I/O = อินพุต/เอาต์พุต P = Power
 - = ไม่ใช่ TTL = ระดับสัญญาณ TTL ST = วงจรขมิตริกเกอร์ อินพุต

Note

- 1 : Buffer จะเป็นแบบขมิตริกเกอร์ เมื่อมีการใช้งานอินเทอร์รัพท์ภายนอก
- 2 : Buffer จะเป็นแบบขมิตริกเกอร์ เมื่อทำงานในโหมดการ โปรแกรม
- 3 : Buffer จะเป็นแบบขมิตริกเกอร์ เมื่อใช้งานทั่วๆ ไปและจะเป็นแบบ TTL เมื่ออยู่ในโหมด Parallel Slave Port ในการติดต่อกับระบบบัสของไมโคร โปรเซสเซอร์
- 4 : Buffer จะเป็นแบบขมิตริกเกอร์ เมื่อใช้งานในโหมด RC Oscillator และจะเป็นแบบ CMOS เมื่อใช้งานอื่นๆ

ตารางที่ 2.9 หน้าที่การทำงานของขาสัญญาณของ PIC 16F877

2.10 ดีซีมอเตอร์ (DC Motor)

มอเตอร์ คือ อุปกรณ์ซึ่งทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล และมอเตอร์ซึ่งขับเคลื่อนด้วยกระแสไฟฟ้าดีซีนี้เรียกว่า ดีซีมอเตอร์ ทิศทางการหมุนของมอเตอร์ดีซีจะตรงกันข้ามกับกรณีเหมือนกันทุกประการ จึงสามารถนำเครื่องกำเนิดไฟฟ้าดีซีมาใช้ทำหน้าที่เครื่องกำเนิดไฟฟ้าดีซีโดยสิ้นเชิง แต่สำหรับ โครงสร้างแล้วจะของมอเตอร์ดีซีได้

เมื่อมอเตอร์ดีซีหมุน ลวดตัวนำแต่ละเส้นซึ่งมีกระแสไหลผ่านจะตัดกับเส้นแรงแม่เหล็กจากขั้วแม่เหล็กใหญ่ ทำให้เกิดแรงเคลื่อนไฟฟ้าขึ้นมาจำนวนหนึ่งในทิศทางที่สวนกับทิศทางของกระแสที่ป้อนให้กับมอเตอร์ เรียกว่า แรงเคลื่อนไฟฟ้าย้อนกลับ

ในการเลือกให้มอเตอร์ ควรคำนึงถึงแรงที่พยายามขับเคลื่อนแกนเหล็กของมอเตอร์ด้วย หรือ ทอร์ก ซึ่งจะพบว่าหากมอเตอร์ยิ่งหมุนเร็ว ทอร์กจะยิ่งน้อย ในทางกลับกันหากมอเตอร์ยิ่งหมุนช้าจะยิ่งให้ทอร์กมากหรือขับโหลดได้มากนั่นเอง นอกจากนี้ดีซีมอเตอร์ยังมีความสามารถในการปรับกระแสได้เองตามขนาดของโหลด กล่าวคือ เมื่อป้อนแรงดัน V คงที่ ให้กับมอเตอร์และให้ทำงานที่โหลดต่ำๆ มอเตอร์จะหมุนที่ความเร็วรอบค่าหนึ่ง และจะดึงกระแสไฟดีซีค่าหนึ่ง (I) เนื่องจากด้านเอาต์พุตต้องการกำลังต่ำ ($P = IV$) แหล่งจ่ายจึงจ่าย I ออกมาต่ำไปด้วย เมื่อใดก็ตามที่ให้มอเตอร์ทำงานที่โหลดมากๆ กำลัง (P) ก็จะต้องมากขึ้นตามไปด้วย แต่หากแหล่งจ่ายของเรามีค่าคงที่ นั่นคือ V คงที่ แหล่งจ่ายจำเป็นจะต้องจ่ายกระแสให้มากขึ้นเพื่อให้ได้กำลังมากพอที่จะขับโหลด นั่นคือเมื่อโหลดน้อยมอเตอร์จะดึงกระแสจากแหล่งจ่ายไฟดีซีน้อย และเมื่อโหลดมากมอเตอร์จะดึงกระแสจากแหล่งจ่ายไฟดีซีมากขึ้นโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การสร้างและการออกแบบ

การออกแบบหุ่นยนต์ ได้แบ่งออกเป็น 3 ส่วนใหญ่ คือ การออกแบบในส่วนของโปรแกรม การตรวจจับใบหน้าคน โปรแกรมในไมโครคอนโทรลเลอร์เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ และโครงสร้างของหุ่นยนต์ โดยมีรายละเอียดดังนี้

3.1 หลักการทำงานของ การตรวจจับใบหน้าคน

ในการตรวจจับใบหน้าคน ได้ใช้โปรแกรมที่ใช้ไลบรารีของโอเพ่นซีวี ซึ่งมีกลุ่มคำสั่งที่สำคัญ ดังนี้

1) cvCapture

การสั่งให้รับภาพจากกล้องเว็บแคม

2) cvCreateMemStorage

การสร้างหน่วยความจำ โดยดีฟอลท์ (Default) แล้วเป็นขนาด 64K

3) cvGrabFrame

การจับภาพแต่ละเฟรม (Frame) จากกล้องเว็บแคม แล้วเก็บในรูปแบบที่ตัวกล้องกำหนด

4) cvRetrieveFrame

การเรียกใช้เฟรมที่เก็บไว้จากข้อที่ 3)

5) cvReleaseCapture

การแสดงภาพสู่หน้าจอ

6) cvHaarClassifierCascade

การเรียกใช้ฐานข้อมูล haarcascade_frontalface_alt.xml

7) cvHaarDetectObjects

การหาส่วนต่างๆของใบหน้าคน เช่น หู ตา จมูก ปาก เป็นต้น

8) CvSeq* cvHaarDetectObjects

การรวมส่วนต่างๆของใบหน้าคนที่ซ้อนทับกันอยู่มารวมกันเป็นพื้นที่ใหญ่เพียงพื้นที่เดียว

9) cvRectangle

การสั่งให้วาดกรอบสี่เหลี่ยมใหญ่จากข้อ 8) ซึ่งกรอบนี้เองคือกรอบล้อมรอบใบหน้าคน

การทำงานของโปรแกรมในการตรวจจับใบหน้าคนแสดงในโฟลว์ชาร์ต (Flow Chart) ดัง

รูปที่ 3.1 ซึ่งสามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การรับภาพ

ใช้กล้องเว็บแคมยี่ห้อโลจitech (Logitech) ในการรับภาพเข้าสู่คอมพิวเตอร์ ซึ่งสามารถปรับความละเอียดของภาพได้ โดยขนาดของภาพอยู่ที่ 360 x 280 พิกเซล คำสั่งที่ใช้ในการรับภาพจากกล้องเว็บแคมคือ cvCapture โดยภาพที่รับเข้ามานี้จะถูกจัดเก็บในรูปแบบสี RGB

3.1.2 การเทียบสีและแบบแผนของส่วนต่างๆ บนใบหน้าคน

หลังจากที่มีการจัดเก็บภาพที่รับเข้ามาจากกล้องเว็บแคมเรียบร้อยแล้ว เราจะต้องนำภาพนั้นไปประมวลผลหาส่วนประกอบของใบหน้าคน ซึ่งจะต้องเรียกใช้ฐานข้อมูลที่ได้มีการเก็บรวบรวมรายละเอียดในแต่ละส่วนของใบหน้าคน เช่น หู ตา ปาก จมูก คิ้ว เป็นต้น ทำให้เราสามารถแยกแยะได้ว่าส่วนใดเป็นส่วนประกอบของใบหน้าคน โดยไม่จำกัดสีผิว ฐานข้อมูลนี้จัดเก็บอยู่ในไฟล์ชื่อ haarcascade_frontalface_alt.xml ที่จะต้องคัดลอก (Copy) จากแผ่นของโอเพ่นซีวี มาไว้ในแฟ้มโปรเจกของเรา การเรียกใช้ฐานข้อมูลนี้จะใช้คำสั่ง cvHaarClassifierCascade จากนั้นจะต้องสั่งให้โปรแกรมทำการเทียบสีและแบบแผนที่อาจเป็นส่วนประกอบของใบหน้าคนกับฐานข้อมูลนี้ โดยใช้คำสั่ง cvHaarDetectObjects คำสั่งนี้จะทำการสแกนภาพเพื่อหาส่วนประกอบของใบหน้าคนและจดจำไว้ในรูปของสี่เหลี่ยมเล็กๆ หากมีการซ้อนทับกันของสี่เหลี่ยมเหล่านี้ ก็จะนำมารวมกันเพื่อให้ได้เป็นพื้นที่ใหญ่เพียงพื้นที่เดียว โดยวิธีการเทียบสีและแบบแผนนี้จะใช้การแปลงภาพเป็นสีระดับเกรย์ แล้วทำเทรศ โชลเพื่อหาขอบในภาพ

ตัวอย่าง ส่วนประกอบของใบหน้าคนที่รวบรวมอยู่ใน haarcascade_frontalface_alt.xml



ส่วนนี้อาจเป็นคิ้วหรือปาก เนื่องจากมีความเข้มมากกว่าบริเวณ โดยรอบ



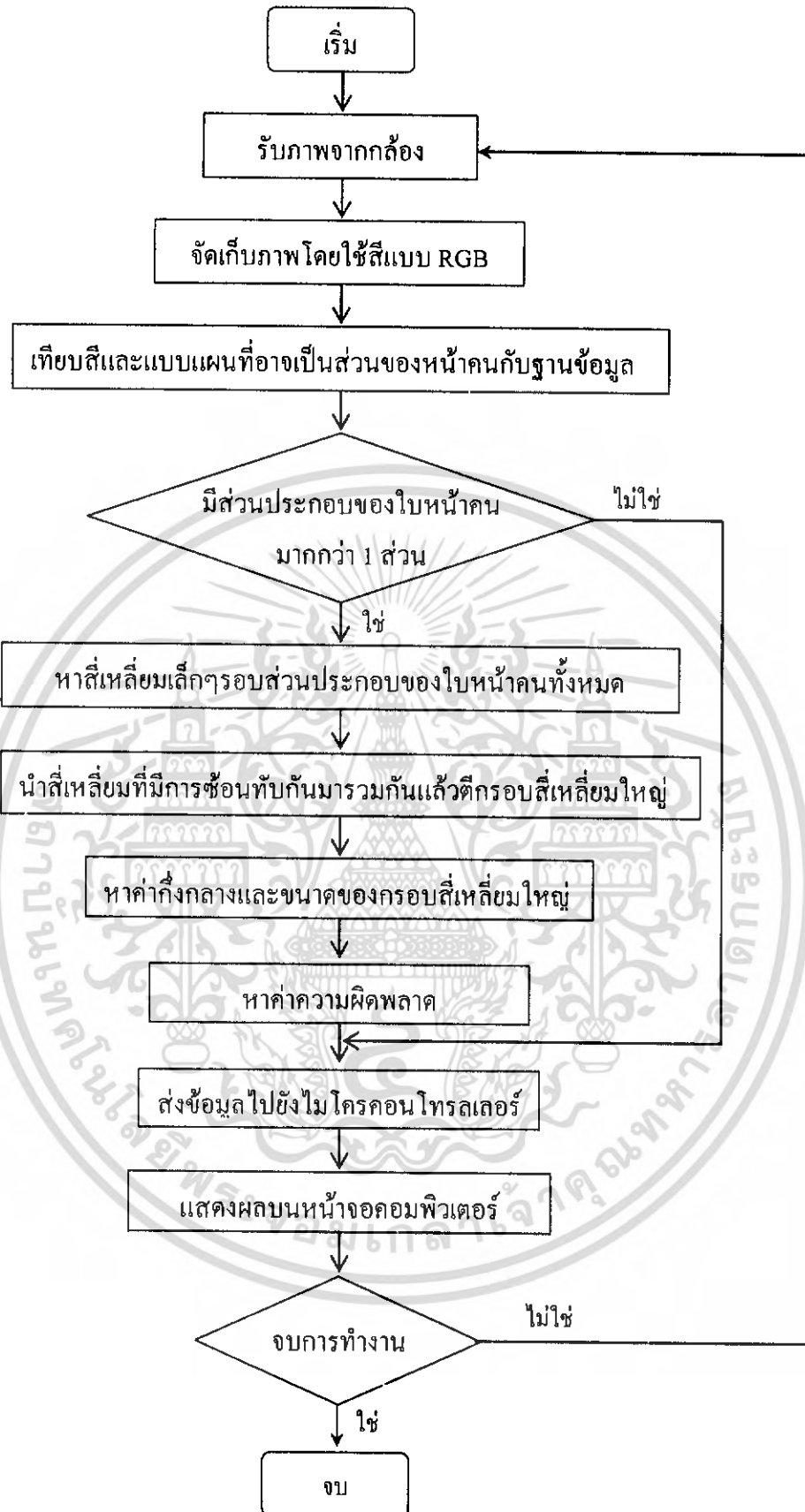
ส่วนนี้อาจเป็นตา เนื่องจากมันมีความเข้มมากกว่าลูกตา



ส่วนนี้อาจเป็นจมูกเนื่องจากสันจมูกมีความเข้มมากกว่าบริเวณ โดยรอบ

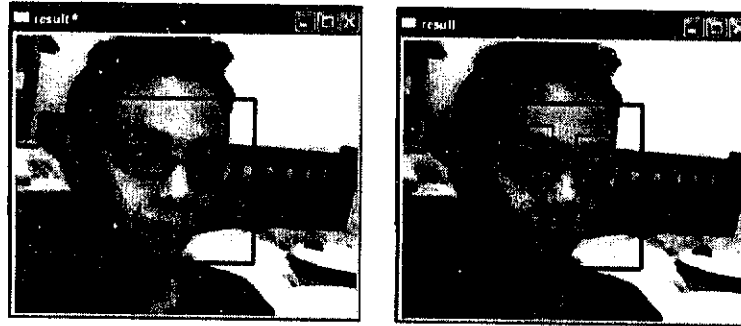
3.1.3 การติกรอบและแสดงผล

ติกรอบสี่เหลี่ยมรอบพื้นที่ที่ได้จากข้อ 3.1.2 โดยใช้คำสั่ง cvRectangle ซึ่งภายในกรอบสี่เหลี่ยมนี้ก็คือตำแหน่งของใบหน้าคนนั่นเอง จากนั้นก็นำภาพที่ได้ไปแสดงผลบนหน้าจอ ไม่ว่าจะจับภาพหน้าคนได้หรือไม่ก็ตาม



รูปที่ 3.1 โฟลว์ชาร์ตแสดงการระบุตำแหน่งของใบหน้าคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ตัวอย่างการประมวลผลส่วนต่างๆ ของใบหน้า

3.1.4 การหาค่าความผิดพลาด

ในการหาค่าความผิดพลาดของตำแหน่งใบหน้าคนขณะใดๆ สามารถทำได้โดยการหาผลต่างของพิกัดของจุดกึ่งกลางของกรอบสี่เหลี่ยมที่ถูกสร้างขึ้นในข้อ 3.1.4 กับพิกัดของจุดกึ่งกลางของหน้าตาหรือเรียกว่าพิกัดอ้างอิง (reference point) ในที่นี้ได้กำหนดตัวแปรต่างๆ ดังนี้

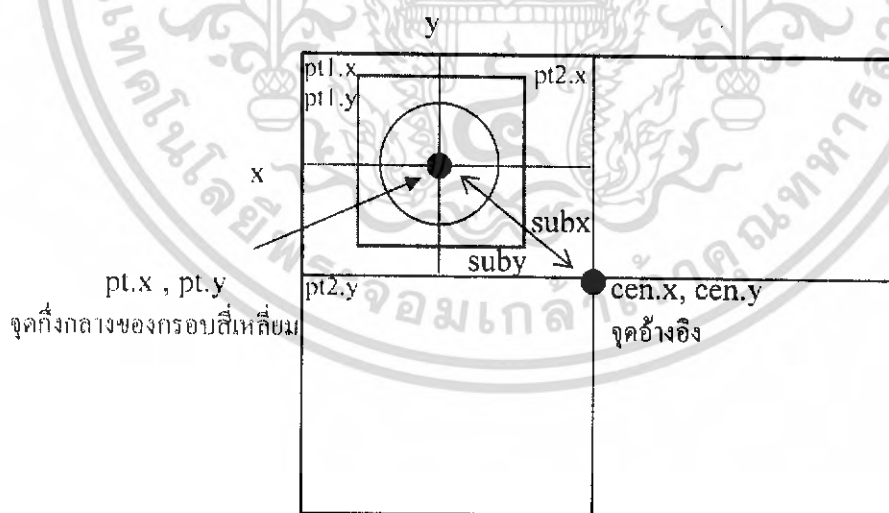
ค่าความผิดพลาดในแนวแกน x ให้แทนด้วย subx

ค่าความผิดพลาดในแนวแกน y ให้แทนด้วย suby

จุดกึ่งกลางของหน้าตาหรือจุดอ้างอิง ให้แทนด้วย (cen.x, cen.y)

จุดกึ่งกลางของกรอบสี่เหลี่ยมให้แทนด้วย (pt.x, pt.y)

ทั้งนี้ พิกัด (cen.x, cen.y) ก็คือ จุดที่ความกว้างและความยาวของหน้าตาหารด้วยสองพอดี ซึ่งในที่นี้อยู่ที่พิกัด (180,140) ส่วนพิกัด (pt.x, pt.y) นั้น ก็คือ จุดที่ความกว้างและความยาวของเส้นกรอบสี่เหลี่ยมหารสองด้วยเช่นกัน



รูปที่ 3.3 แสดงการหาค่าความผิดพลาดของตำแหน่งใบหน้าคนจากจุดอ้างอิง

$$pt.x = (pt1.x + pt2.x) / 2 ;$$

$$pt.y = (pt2.y + pt2.y) / 2 ;$$

กำหนดจุดอ้างอิง : $cen.x = 180 ;$

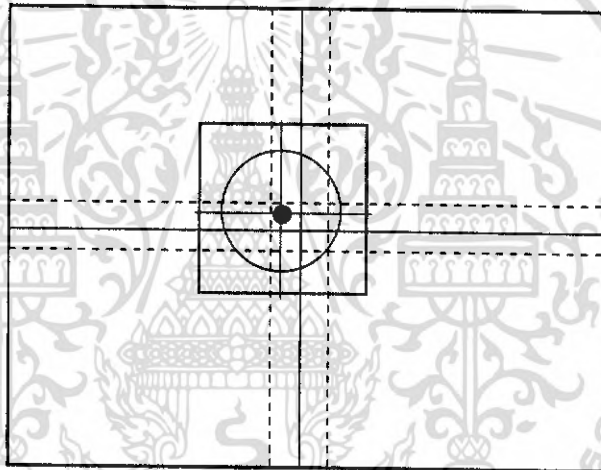
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{cen.y} = 140 ;$$

$$\text{ค่าความผิดพลาด: } \text{subx} = \text{cen.x-pt.x} ;$$

$$\text{suby} = \text{cen.y-pt.y} ;$$

อย่างไรก็ตาม การที่จะทำให้จุดกึ่งกลางของกรอบสี่เหลี่ยมมาอยู่ในตำแหน่งที่ตรงกับจุดอ้างอิงพอดีนั้นเป็นไปได้ยากมาก เนื่องจากใบหน้าคนมีการเคลื่อนที่อยู่ตลอดเวลา ทำให้เกิดการกระตุกของหุ่นยนต์ขึ้น จึงจำเป็นต้องกำหนดเป็นช่วงค่าความผิดพลาดที่ยอมรับได้ขึ้นมาค่าหนึ่ง ซึ่งค่านี้ได้มาจากการทดลองจริง เพื่อหาค่าที่เหมาะสมของแต่ละแกน ที่จะทำให้หุ่นยนต์หยุดการเคลื่อนที่พอดี จากการทดลองพบว่าช่วงค่าความผิดพลาดที่เหมาะสมในแนวแกน x และ แกน y คือ 40 พิกเซล ซึ่งหมายความว่า หากค่า subx และ suby ไม่เกิน ± 40 หุ่นยนต์จะไม่มีอาการเคลื่อนที่ซ้ายขวา หรือขึ้นลง ตามลำดับ การกำหนดช่วงค่าความผิดพลาดนี้สามารถอธิบายได้ดังรูปที่ 3.4 การทำเช่นนี้ก็เพื่อลดปัญหาการเคลื่อนที่กระตุกของหุ่นยนต์



รูปที่ 3.4 การกำหนดช่วงค่าความผิดพลาดในแนวแกน x และ แกน y

ในแนวแกน z เป็นการเคลื่อนที่เดินหน้าและถอยหลัง เพื่อปรับระยะระหว่างใบหน้าคนกับตัวหุ่นยนต์ให้เหมาะสม โดยการหาขนาดของใบหน้าคน ซึ่งขนาดของใบหน้าคน ($z1$) ก็คือขนาดของกรอบสี่เหลี่ยมที่ล้อมรอบอยู่ เนื่องจากกรอบสี่เหลี่ยมล้อมรอบใบหน้าคนนี้เป็นสี่เหลี่ยมจัตุรัส ความกว้างจึงเท่ากับความยาว ซึ่งมีค่าเท่ากับ $z1 = \text{pt2.x} - \text{pt1.x}$

ในที่นี้ เราต้องการให้ระยะห่างระหว่างใบหน้าคนกับตัวหุ่นยนต์มีขนาดประมาณ 1 เมตร ซึ่งจากการทดลองพบว่าตรงกับค่า $z1$ ประมาณ 100 พิกเซล และจากการทดลองยังพบว่าช่วงค่าความผิดพลาดที่เหมาะสมในแนวแกน z คือ 10 พิกเซล ซึ่งหมายความว่าหากค่า $z1$ ไม่เกิน ± 10 หรือกรอบสี่เหลี่ยมมีขนาด 90- 110 พิกเซล หุ่นยนต์จะไม่มีอาการเคลื่อนที่เดินหน้าหรือถอยหลัง การทำเช่นนี้ก็เพื่อลดปัญหาการเคลื่อนที่กระตุกของหุ่นยนต์ เช่นเดียวกับการกำหนดช่วงค่าความผิดพลาด

ในแนวแกน x และ แกน y ที่ได้กล่าวไว้ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 โปรแกรมในการกำหนดการเคลื่อนที่ของหุ่นยนต์

จากการหาค่า $subx$, $suby$ และ $z1$ ในข้อ 3.1.5 ทำให้เราทราบว่าขณะนั้นใบหน้าคนอยู่บริเวณใดของหน้าต่าง มีขนาดเท่าใด และกล้องควรจะเคลื่อนที่ไปในทิศทางใด จึงจะสามารถทำให้ใบหน้าคนมาอยู่บริเวณกึ่งกลางของหน้าต่างได้ โดยค่า $subx$ และ $suby$ บอกถึงควอดแรนท์ที่พบจุดกึ่งกลางของกรอบสี่เหลี่ยม ส่วนค่า $z1$ บอกระยะของใบหน้าคนกับหุ่นยนต์ ค่าเหล่านี้จะถูกนำมาคำนวณอีกครั้ง เพื่อกำหนดค่าที่จะส่งออกไปทางพอร์ตอนุกรม จากนั้นให้ไมโครคอนโทรลเลอร์ทำการประมวลผลสั่งให้มอเตอร์หมุนไปในทิศทางที่ต้องการ เพื่อให้การเคลื่อนที่เป็นไปได้ 3 มิติ คือ ซ้าย ขวา ขึ้นบน ลงล่าง เดินหน้า ถอยหลัง

3.1.5.1 โปรแกรมการเคลื่อนที่ซ้ายขวาและขึ้นลง

จากการหาค่า $z1$ ในหัวข้อ 3.1.5 ทำให้เราทราบขนาดของใบหน้าคนจากการที่เรากำหนดให้จุดอ้างอิงในแนวแกน x อยู่ที่ $cen.x = 180$ และในแนวแกน y อยู่ที่ $cen.y = 140$ เราจะสามารถหาค่าความผิดพลาดของตำแหน่งของกรอบสี่เหลี่ยมล้อมรอบใบหน้าคนได้ โดยนำค่าพิกัดของจุดอ้างอิงมาลบกับค่าพิกัดของจุดกึ่งกลางของกรอบสี่เหลี่ยม ทั้งในแนวแกน x และแกน y จากนั้นก็กำหนดให้ตัวแปร $xout$ และ $yout$ มารับค่าที่อาจเป็นไปได้ทั้งหมด ได้แก่ ในแกน x หากลบกันแล้วมีค่ามากกว่า 40 แสดงว่าหุ่นยนต์จะต้องทำการเคลื่อนที่ไปทางซ้าย ถ้าหากมีค่าน้อยกว่า -40 แสดงว่าหุ่นยนต์จะต้องเคลื่อนที่ไปทางขวา แต่หากค่าที่ได้นั้นอยู่ระหว่าง -40 และ 40 หุ่นยนต์จะหยุดเคลื่อนที่ตามแนวแกน x และในทำนองเดียวกันกับแกน y คือ หากได้ผลลัพธ์มากกว่า 40 กล้องเว็บแคมจะต้องเคลื่อนที่ขึ้นบน แต่หากได้ผลลัพธ์น้อยกว่า 40 กล้องเว็บแคมจะต้องเคลื่อนที่ลงล่าง และถ้าค่าที่ได้อยู่ระหว่าง -40 กับ 40 หุ่นยนต์จะหยุดเคลื่อนที่ตามแนวแกน y เพื่อเป็นการลดปัญหาการเคลื่อนที่กระตุกของหุ่นยนต์ จึงกำหนดค่าความผิดพลาดที่เหมาะสมในแนวแกน x และ แกน y เท่ากับ 40 พิกเซล ซึ่งหมายความว่าหากค่า $subx$ และ $suby$ ไม่เกิน 40 หุ่นยนต์จะไม่มี การเคลื่อนที่ซ้ายขวา และขึ้นลง ตามลำดับ โดยการเขียนโปรแกรมมีดังนี้

$cen.x=180$; พิกัดจุดอ้างอิงในแนวแกน x

$cen.y=140$; พิกัดจุดอ้างอิงในแนวแกน y

$subx = cen.x-pt.x$; ค่าความผิดพลาดในแนวแกน x

$suby = cen.y-pt.y$; ค่าความผิดพลาดในแนวแกน y

if ($subx > 40$) $xout=1$; หุ่นยนต์จะต้องเคลื่อนที่ไปทางขวา

if ($subx < -40$) $xout=2$; หุ่นยนต์จะต้องเคลื่อนที่ไปทางซ้าย

if ($subx > -40 \ \&\& \ subx < 40$) $xout=0$; หุ่นยนต์ไม่เคลื่อนที่ไปทางซ้ายและขวา

if ($suby > 40$) $yout=1$; หุ่นยนต์จะต้องเคลื่อนที่ขึ้นบน

if ($suby < -40$) $yout=2$; หุ่นยนต์จะต้องเคลื่อนที่ลงล่าง

if (suby>-40 && suby<40) yout=0; หุ่นยนต์จะไม่มี การเคลื่อนที่ขึ้นและลง

3.1.5.2 โปรแกรมการกำหนดระยะระหว่างใบหน้าคนกับหุ่นยนต์

การกำหนดระยะของใบหน้าคนกับตัวหุ่นยนต์ คือ การกำหนดขนาดของกรอบสี่เหลี่ยมที่ล้อมรอบใบหน้าคนนั่นเอง ซึ่งสามารถปรับขนาดได้โดยการเคลื่อนที่เดินหน้าและถอยหลังของหุ่นยนต์ กล่าวคือ หากกรอบสี่เหลี่ยมมีขนาดเล็กกว่าที่กำหนดไว้ หุ่นยนต์จะต้องทำการเดินหน้าให้เข้าใกล้ใบหน้าคนมากขึ้น ในทางกลับกัน หากกรอบสี่เหลี่ยมมีขนาดใหญ่กว่าที่กำหนดไว้ หุ่นยนต์จะต้องเคลื่อนที่ถอยหลังเพื่อให้ใบหน้าคนมีขนาดพอดี

จากที่ได้กล่าวไว้แล้วในหัวข้อ 3.1.5 ถึงการกำหนดช่วงค่าความผิดพลาดเป็น 10 พิกเซล สามารถเขียนโปรแกรมได้ดังนี้

$z1 = pt2.x - pt1.x$; ขนาดของกรอบสี่เหลี่ยม

if (z1<90) zout=1; กรอบสี่เหลี่ยมมีขนาดเล็กเกินไป หุ่นยนต์ต้องเดินหน้า

if (z1>110) zout=2; กรอบสี่เหลี่ยมมีขนาดใหญ่เกินไป หุ่นยนต์ต้องถอยหลัง

if (z1>90 && z1<110) zout=0; หุ่นยนต์ไม่เคลื่อนที่เดินหน้าและถอยหลัง

3.1.6 การส่งค่าเอาต์พุตจากคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรม

จากค่า xout yout และ zout ที่ได้จากข้อ 3.1.6.1 และ 3.1.6.2 สามารถนำมาเขียนเป็นกรณีต่างๆ เพื่อทำการส่งไปยังไมโครคอนโทรลเลอร์ ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ได้ทั้งหมด 27 กรณี โดยกำหนดให้ตัวแปร out เป็นตัวเก็บค่าของแต่ละกรณี โดยให้บิตเริ่มต้น (Start bit) เป็น a และบิตสุดท้าย (Stop bit) เป็น b ดังแสดงในตารางที่ 3.1

3.2 การควบคุมตำแหน่งของหุ่นยนต์

ในการที่จะทำให้หุ่นยนต์เคลื่อนที่ได้อย่างถูกต้องนั้น จะต้องประกอบด้วยองค์ประกอบหลายๆ ส่วน คือ ไมโครคอนโทรลเลอร์ คีชีมอเตอร์ และวงจรมอเตอร์

Xout	yout	Zout	out	ลักษณะการเคลื่อนที่
0	0	0	a01b	ไม่มีการเคลื่อนที่
0	0	1	a02b	เดินหน้า
0	0	2	a03b	ถอยหลัง
0	1	0	a04b	ขึ้นบน
0	1	1	a05b	ขึ้นบน เดินหน้า
0	1	2	a06b	ขึ้นบน ถอยหลัง
0	2	0	a07b	ลงล่าง
0	2	1	a08b	ลงล่าง เดินหน้า
0	2	2	a09b	ลงล่าง ถอยหลัง
1	0	0	a10b	ซ้าย
1	0	1	a11b	ซ้าย เดินหน้า
1	0	2	a12b	ซ้าย ถอยหลัง
1	1	0	a13b	ซ้าย ขึ้นบน
1	1	1	a14b	ซ้าย ขึ้นบน เดินหน้า
1	1	2	a15b	ซ้าย ขึ้นบน ถอยหลัง
1	2	0	a16b	ซ้าย ลงล่าง
1	2	1	a17b	ซ้าย ลงล่าง เดินหน้า
1	2	2	a18b	ซ้าย ลงล่าง ถอยหลัง
2	0	0	a19b	ขวา
2	0	1	a20b	ขวา เดินหน้า
2	0	2	a21b	ขวา ถอยหลัง
2	1	0	a22b	ขวา ขึ้นบน
2	1	1	a23b	ขวา ขึ้นบน เดินหน้า
2	1	2	a24b	ขวา ขึ้นบน ถอยหลัง
2	2	0	a25b	ขวา ลงล่าง
2	2	1	a26b	ขวา ลงล่าง เดินหน้า
2	2	2	a27b	ขวา ลงล่าง ถอยหลัง

ตารางที่ 3.1 ค่าเอาต์พุตในแต่ละกรณีและลักษณะการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ที่ใช้ในที่นี้คือ PIC 16F877A หลังจากทีโปรแกรมตรวจจับใบหน้าคนทำการระบุตำแหน่งใบหน้าคนเรียบร้อยแล้วจะต้องทำการส่งข้อมูลออกมาทางพอร์ตอนุกรมเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ โดยข้อมูลเหล่านี้อยู่ในตัวแปร out รวมทั้งหมด 27 กรณี ดังตารางที่ 3.1 จากนั้นนำข้อมูลที่ได้ไปกำหนดค่าเอาต์พุตของไมโครคอนโทรลเลอร์ดังตารางที่ 3.2 ซึ่งในที่นี้ได้เลือกใช้พอร์ต B โดยได้กำหนดค่าให้บิตแต่ละบิต ทำหน้าที่ดังนี้

RB0	ทำหน้าที่ควบคุมการเคลื่อนที่ถอยหลัง
RB1	ทำหน้าที่ควบคุมการเคลื่อนที่เดินหน้า
RB2	ทำหน้าที่ควบคุมการเคลื่อนที่ไปทางขวา
RB3	ทำหน้าที่ควบคุมการเคลื่อนที่ไปทางซ้าย
RB4	ทำหน้าที่ควบคุมการเคลื่อนที่ลงล่าง
RB5	ทำหน้าที่ควบคุมการเคลื่อนที่ขึ้นบน
RB6	ไม่ใช้
RB7	ไม่ใช้

ในการที่จะส่งข้อมูลจากพอร์ตอนุกรมของคอมพิวเตอร์นั้น จะต้องมีการแปลงสัญญาณให้เป็นแบบ TTL เพื่อส่งเข้าสู่ไมโครคอนโทรลเลอร์ ตัวแปลงสัญญาณนี้ก็คือ MAX 232 ดังที่ได้กล่าวไว้ในหัวข้อ 2.6 จากนั้นจึงส่งสัญญาณจาก MAX 232 ไปที่ไมโครคอนโทรลเลอร์ PIC 16F877A เพื่อควบคุมการทำงานของมอเตอร์ต่อไป โดยรูปร่างของระบบทั้งหมดแสดงในรูปที่ 3.5

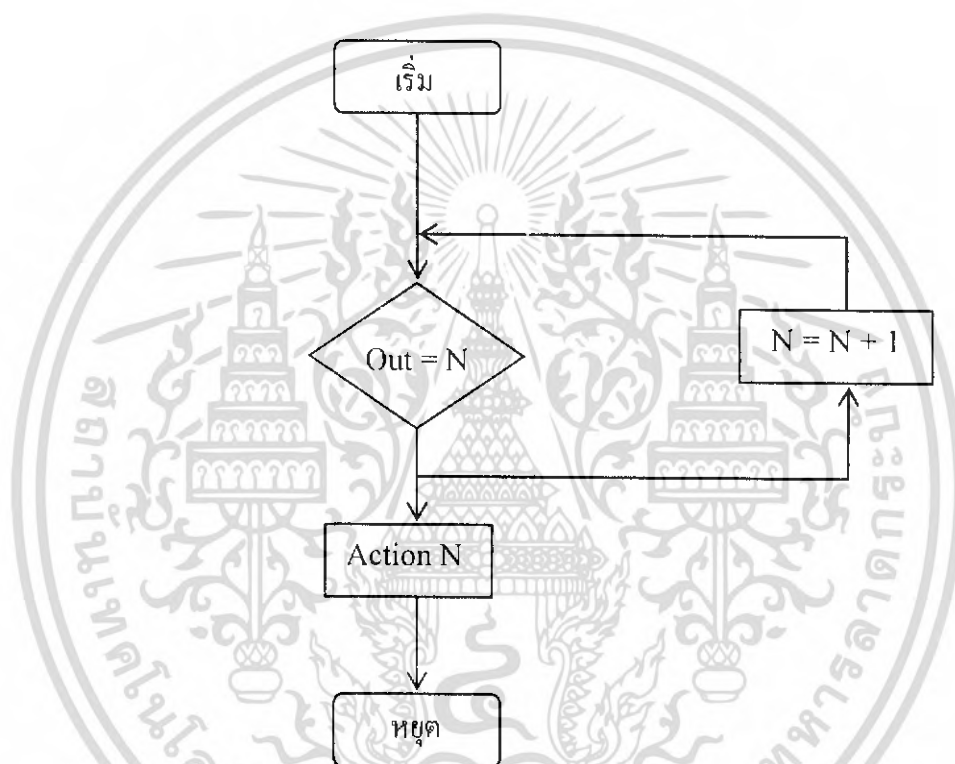
ตัวอย่างการส่งสัญญาณจากคอมพิวเตอร์มายังไมโครคอนโทรลเลอร์ เพื่อไปทำการขับมอเตอร์ให้เดินหน้า สามารถอธิบายได้ด้วยตารางที่ 3.1 หากต้องการให้หุ่นยนต์เดินหน้า จะต้องกำหนดค่า xout และ yout เป็น 0 ส่วนค่า zout เป็น 1 จากนั้น คอมพิวเตอร์จะส่งค่า a02b ให้กับไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูล จะทำการตรวจสอบบิตเริ่มต้น เมื่อพบว่าเป็น a จะทำการรับค่า 2 บิตถัดมา เพื่อกำหนดค่าเอาต์พุตที่พอร์ต B ซึ่งจากตารางที่ 3.2 จะพบว่า เมื่อไมโครคอนโทรลเลอร์รับค่า 02 เอาต์พุตของพอร์ต B จะถูกกำหนดให้เป็น 00000010 ซึ่งหมายความว่า ให้ตั้งสถานะบิต RB1 เป็น High เท่านั้น ส่วนบิตอื่นๆ จะมีสถานะเป็น Low สถานะที่ขาของพอร์ต B นี้ จะถูกส่งไปให้ L298 โดยสถานะของบิต RB0 จะถูกส่งไปให้ขา 10 และ ขา 5 ของ L298 ส่วนสถานะของบิต RB1 จะถูกส่งไปให้ขา 7 และ ขา 12 ของ L298 เพื่อนำไปขับมอเตอร์ทั้งสองตัวที่ต่ออยู่กับขาเอาต์พุตของ L298 ให้เคลื่อนที่เดินหน้า (มอเตอร์ทั้ง 2 ตัวที่ใช้ L298 ในการขับจะหมุนไปในทิศทางเดียวกัน)

ตัวแปร out	ไมใช่ (RB7)	ไมใช่ (RB6)	ขึ้นบน (RB5)	ลงล่าง (RB4)	ซ้าย (RB3)	ขวา (RB2)	เดินหน้า (RB1)	ถอยหลัง (RB0)
01	0	0	0	0	0	0	0	0
02	0	0	0	0	0	0	1	0
03	0	0	0	0	0	0	0	1
04	0	0	1	0	0	0	0	0
05	0	0	1	0	0	0	1	0
06	0	0	1	0	0	0	0	1
07	0	0	0	1	0	0	0	0
08	0	0	0	1	0	0	1	0
09	0	0	0	1	0	0	0	1
10	0	0	0	0	1	0	0	0
11	0	0	0	0	1	0	1	0
12	0	0	0	0	1	0	0	1
13	0	0	1	0	1	0	0	0
14	0	0	1	0	1	0	1	0
15	0	0	1	0	1	0	0	1
16	0	0	0	1	1	0	0	0
17	0	0	0	1	1	0	1	0
18	0	0	0	1	1	0	0	1
19	0	0	0	0	0	1	0	0
20	0	0	0	0	0	1	1	0
21	0	0	0	0	0	1	0	1
22	0	0	1	0	0	1	0	0
23	0	0	1	0	0	1	1	0
24	0	0	1	0	0	1	0	1
25	0	0	0	1	0	1	0	0
26	0	0	0	1	0	1	1	0
27	0	0	0	1	0	1	0	1

ตารางที่ 3.2 การกำหนดค่าเอาต์พุตของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมไมโครคอนโทรลเลอร์แสดงดังรูปที่ 3.6 ใช้หลักการตรวจสอบกรณีทั้งหมด 27 กรณี ซึ่งก็คือตัวแปร Out ที่ได้รับจากพอร์ตอนุกรมของคอมพิวเตอร์นั่นเอง โดยไมโครคอนโทรลเลอร์จะตรวจสอบค่า Out เทียบกับค่าเปรียบเทียบ (N) โดยเริ่มที่ $N = 01$ หากไม่ตรงกับค่า Out จากพอร์ตอนุกรมของคอมพิวเตอร์ ไมโครคอนโทรลเลอร์จะทำการบวกค่า N เพิ่มไปอีก 1 เป็น $N = 02$ แล้วตรวจสอบว่าตรงกับค่า Out จากพอร์ตอนุกรมของคอมพิวเตอร์หรือไม่ หากไม่ใช่ ไมโครคอนโทรลเลอร์จะทำการเพิ่มค่า N ไปเรื่อยๆ ทีละ 1 หรือ $N = N + 1$ จนกว่าจะได้ค่า N ที่ตรงกับค่า Out ที่รับเข้ามาจากพอร์ตอนุกรมของคอมพิวเตอร์ แล้วจึงส่งสถานะไปควบคุมทิศทางการหมุนของมอเตอร์ตามตารางที่ 3.2 ต่อไป



รูปที่ 3.6 การทำงานของ โปรแกรมไมโครคอนโทรลเลอร์

3.2.2 ดีซีมอเตอร์ (DC Motor)

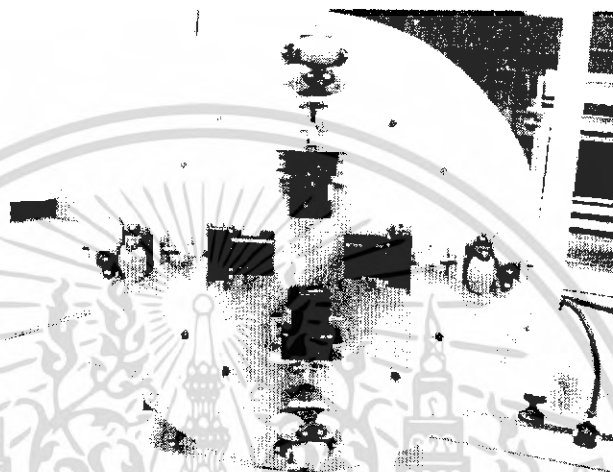
ในโครงการนี้ ได้เลือกใช้ดีซีมอเตอร์ในการขับเคลื่อนฐานหุ่นยนต์และขับเคลื่อนล้อวงเวียนแคม ซึ่งมีรายละเอียดดังนี้

3.2.2.1 มอเตอร์ที่ใช้ขับเคลื่อนหุ่นยนต์

ในส่วนขับเคลื่อนหุ่นยนต์ ได้ใช้มอเตอร์ทั้งหมด 4 ตัว ซึ่งติดอยู่กับส่วนล้อทั้งสิ้น เพื่อการเคลื่อนที่ซ้าย ขวา เดินหน้า และถอยหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากหุ่นยนต์ต้องมีขนาดใหญ่ เพื่อรับน้ำหนักของคอมพิวเตอรืโน้ตบุ๊ก ซึ่งหนักถึง 3 กิโลกรัม และน้ำหนักของโครงสร้างที่มีขนาดใหญ่ จึงจำเป็นต้องใช้มอเตอร์ขนาดใหญ่เพื่อให้มีแรงบิดเพียงพอที่จะขับเคลื่อนหุ่นยนต์ได้ โดยมอเตอร์แต่ละตัวมีพิกัดกระแส 30mA แรงดัน 12V และความเร็ว 95 rpm สังเกตว่าพิกัดความเร็วของมอเตอร์ที่เลือกใช้ในการเคลื่อนที่ของหุ่นยนต์มีค่าไม่สูงนัก เพราะหุ่นยนต์ไม่สามารถเคลื่อนที่เร็วได้เนื่องจากข้อจำกัดในการจับภาพของกล้องยังไม่เร็วพอ หากตัวกล้องเคลื่อนที่เร็วเกินไปจะทำให้ไม่สามารถตรวจจับใบหน้าคนได้



รูปที่ 3.7 การติมอเตอร์เข้ากับล้ออ้อมนิ

3.2.2.2 มอเตอร์ที่ใช้ขับเคลื่อนกล้องเว็บแคม

ในส่วนของการขับเคลื่อนกล้องเว็บแคม อาศัยมอเตอร์ 1 ตัว เพื่อใช้ในการขับเคลื่อนกล้องเว็บแคมให้เคลื่อนที่ขึ้นลง เนื่องจากกล้องเว็บแคมมีขนาดเล็กและน้ำหนักเบา อีกทั้งการออกแบบส่วนขับเคลื่อนด้านบนของหุ่นยนต์มีน้ำหนักเบา ทำให้มอเตอร์ในส่วนนี้ไม่ต้องรับภาระมาก จึงไม่จำเป็นต้องใช้มอเตอร์ที่ให้แรงบิดมากเหมือนกับมอเตอร์สำหรับขับเคลื่อนหุ่นยนต์ในข้อ 3.2.1.1 ทั้งนี้มอเตอร์ที่ใช้จะต้องมีความเร็วรอบต่ำที่ 10 rpm ด้วยเหตุผลของความสามารถในการจับภาพของกล้องเช่นเดียวกับในข้อ 3.2.1.1

3.2.3 วงจรขับมอเตอร์ (Motor Driver)

ในการที่มอเตอร์จะทำงานได้จะต้องผ่านวงจรขับเสียก่อน ซึ่งการขับมอเตอร์อาจทำได้ 2 วิธี คือ การขับแบบลิเนียร์ และ PWM ในโครงงานนี้ได้กำหนดการทำงานของมอเตอร์เพียงทำงานหรือไม่ทำงานเท่านั้น โดยไม่จำเป็นต้องปรับความเร็วของมอเตอร์แต่ใช้การจ่ายแรงดันเต็มพิกัด 12V ให้กับมอเตอร์ทั้งสองส่วนที่กล่าวไว้ในข้อ 3.2.2.1 และ 3.2.2.2 ดังนั้นการใช้วงจรขับดังกล่าวอาจทำได้โดยการต่ออุปกรณ์ทรานซิสเตอร์หรือใช้ IC สำเร็จรูปก็ได้ จากคุณสมบัติของมอเตอร์ที่

เราเลือกใช้ข้างต้น พบว่าสามารถใช้ IC เบอร์ L298 ในการขับเคลื่อนหุ่นยนต์ ซึ่งเป็นวงจรขับแบบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

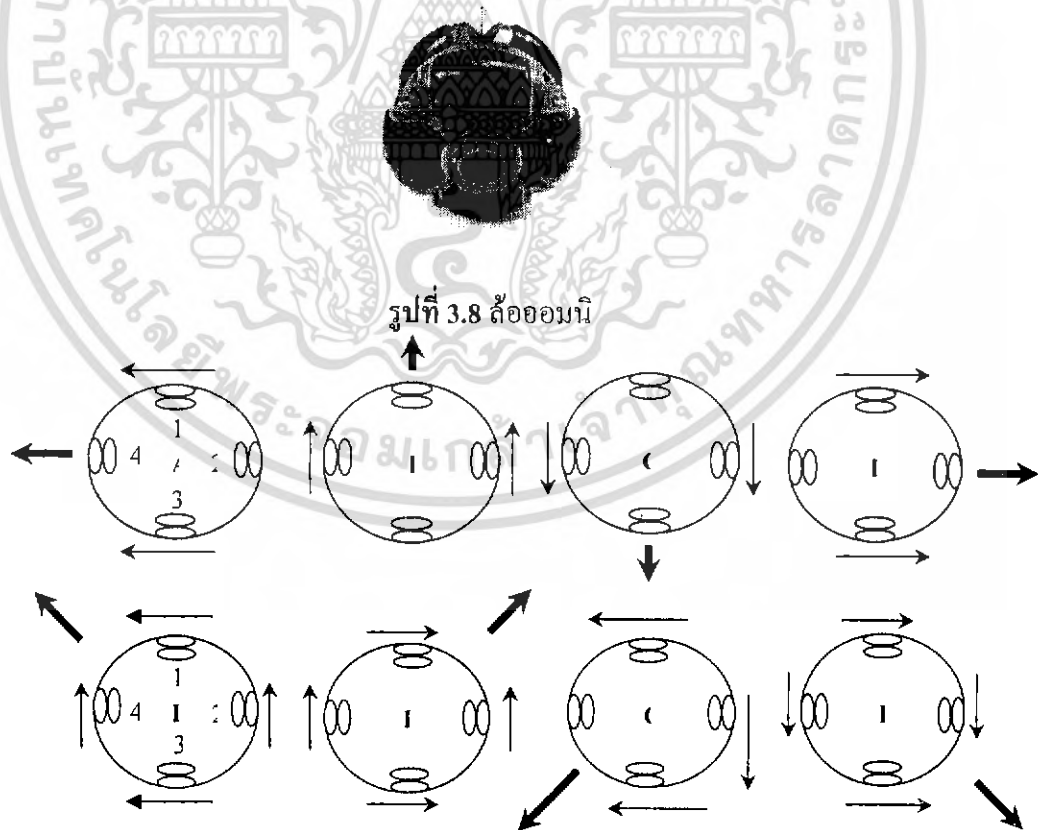
ฟูลบริดจ์ ที่มีพิคัดกระแสสูงสุดถึง 4A แรงดัน 46V และใช้ IC เบอร์ L293D ในการขับเคลื่อนล้อวง เว็บบแคม ซึ่งมีพิคัดกระแสสูงสุด 400mA แรงดัน 36V รายละเอียดของ IC เบอร์ L298 และ L293 นี้ สามารถดูได้ในภาคผนวก

3.3 โครงสร้างทางกลศาสตร์

โครงสร้างทางกลศาสตร์ของหุ่นยนต์ สามารถแยกอธิบายได้เป็น 3 ส่วนด้วยกัน คือ ล้อ ฐานหุ่นยนต์ และส่วนบนของหุ่นยนต์ ซึ่งสัดส่วนแสดงในรูปที่ 3.12

3.3.1 ล้อ

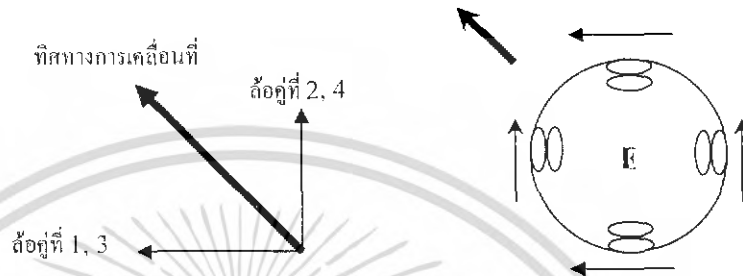
เนื่องจากคุณสมบัติของหุ่นยนต์ คือ จะต้องสามารถติดตามการเคลื่อนไหวของใบหน้าคน ได้ทั้ง 3 มิติ คือ เดินหน้า ถอยหลัง ซ้าย ขวา ขึ้นบน และลงล่าง โดยด้านหน้าของหุ่นยนต์จะต้องหันไปทางด้านหน้าเสมอ เนื่องจากถ้าหากล้อทั้งหมักรวมกับใบหน้าคนมากเกินไป จะทำให้ไม่สามารถจับใบหน้าคนได้ จึงจำเป็นจะต้องใช้ล้อที่มีคุณสมบัติดังกล่าวที่เรียกว่า ล้อแบบเคลื่อนที่ได้หลายทิศทาง (Omni-directional Wheel หรือ Omni Wheel) หรือนิยมเรียกสั้นๆ ว่า ล้อออมนิ มาใช้งาน เมื่อนำล้อมาประกบกัน 2 ล้อ จะสามารถเคลื่อนที่ได้ 2 แกน ดังแสดงในรูปที่ 3.8 ดังนั้นสำหรับการขับเคลื่อนล้อทั้ง 4 ชุด จะต้องใช้ล้อออมนิทั้งหมด 8 ล้อ ดังรูปที่ 3.9



รูปที่ 3.9 การทำงานของล้อออมนิเพื่อให้เคลื่อนที่ในทิศทางต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการพิจารณาว่าจะต้องให้ล้อยอมนิคูใดหมุนบ้าง อาศัยหลักการของเวกเตอร์ในการอธิบาย เช่น การเคลื่อนที่ในแบบ A ตามรูปที่ 3.9 หากเราต้องการให้การเคลื่อนที่เป็นไปทางซ้าย ล้อยอมนิคูที่ 1 และ 3 จะทำการหมุนในทิศทางไปทางซ้าย ต่อมาสังเกตการเคลื่อนที่ในแบบ E ต้องการให้หุ่นยนต์เคลื่อนที่ไปในทิศทางทแยงหน้าซ้าย จะต้องให้ล้อยูที่ 1 และ 3 หมุนในทิศทางไปทางซ้าย และจะต้องให้ล้อยูที่ 2 และ 4 หมุนในทิศทางไปด้านหน้า ซึ่งจะทำให้ได้เวกเตอร์ลัพธ์ในแนวทแยงขึ้น ดังรูปเวกเตอร์ในรูปที่ 3.10



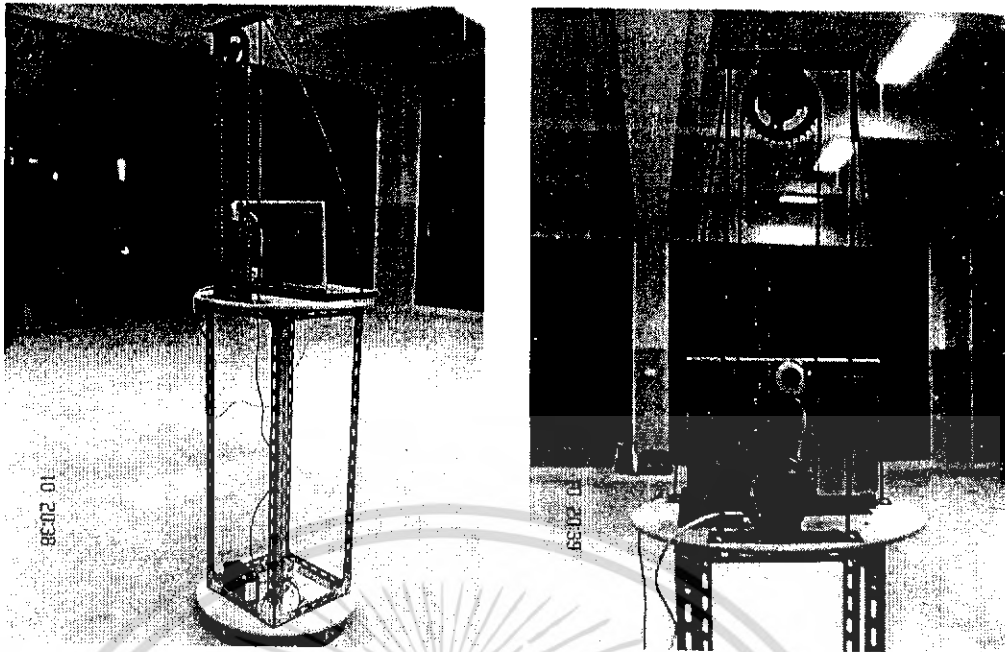
รูปที่ 3.10 ตัวอย่างการเคลื่อนที่ของล้อยอมนิ

3.3.2 ฐานหุ่นยนต์

เนื่องจากการส่งภาพจากกล้องเว็บแคมไปสู่คอมพิวเตอร์ และการเชื่อมต่อพอร์ตอนุกรม ยังมีข้อจำกัดในเรื่องของสายไฟที่ใช้ จึงจำเป็นต้องนำคอมพิวเตอร์โน้ตบุ๊กวางบนตัวหุ่นยนต์ด้วยการออกแบบฐานจึงต้องเน้นความมั่นคง เพื่อให้รองรับน้ำหนักของคอมพิวเตอร์โน้ตบุ๊กได้ ส่งผลให้หุ่นยนต์มีรูปร่างใหญ่และมีน้ำหนักมาก ส่วนฐานหุ่นยนต์นี้สามารถแสดงได้ดังรูปที่ 3.11 (ก)

3.3.3 ส่วนบนของหุ่นยนต์

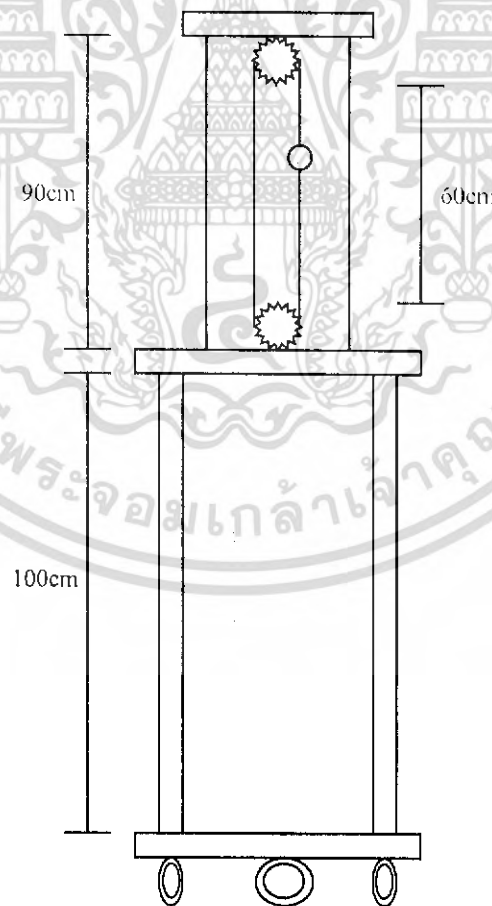
ถูกออกแบบมาเพื่อให้กล้องเว็บแคมสามารถเคลื่อนที่ขึ้นลงได้ในช่วง 60 เซนติเมตร โดยอาศัยโซ่และเฟลาต่อกับมอเตอร์ที่ได้กล่าวไว้ในหัวข้อ 3.2 ซึ่งสามารถใช้ได้กับคนที่มีความสูงประมาณ 2 เมตร เนื่องจากในใช้งานจริง คนไม่ได้ยืนอยู่ติดกับด้านหน้าของหุ่นยนต์พอดี ส่วนบนของหุ่นยนต์นี้ แสดงได้ดังรูป ที่ 3.11 (ข)



(ก) หุ่นยนต์แบบเต็มตัว

(ข) ส่วนบนของหุ่นยนต์

รูปที่ 3.11 หุ่นยนต์ตรวจจับและเคลื่อนที่ตามใบหน้าคน



รูปที่ 3.12 สัดส่วนของหุ่นยนต์ตรวจจับและเคลื่อนที่ตามใบหน้าคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

จากแนวคิดในการออกแบบโครงสร้างหุ่นยนต์และโปรแกรมตามที่ได้กล่าวไว้ในบทที่ 3 ได้ดำเนินการสร้างหุ่นยนต์ขึ้นจริง และได้เขียนโปรแกรมในการตรวจจับใบหน้าคนและควบคุมหุ่นยนต์ตามแนวคิดในรูปที่ 3.1 พบว่าค่าคาบการทำงาน (Sampling Time) มีค่าประมาณ 0.375 วินาที ซึ่งค่านี้จะเป็นขีดจำกัดความเร็วในการเคลื่อนที่ของใบหน้าคนที่ต้องการตรวจจับ

ในโครงการนี้ ได้แบ่งการทดลองออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของโปรแกรมในการตรวจจับใบหน้าคน และส่วนของการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ ในการทดลองแต่ละครั้ง จะต้องมีการปรับความสามารถในการรับแสงของกล้องให้เข้ากับสภาวะของแสง ณ เวลานั้นๆ หากแสงสว่างในสถานที่ทำการทดลองเปลี่ยนไป จะต้องทำการปรับความสามารถในการรับแสงของกล้องใหม่ ทั้งนี้ ความแม่นยำในการระบุตำแหน่งใบหน้าคนขึ้นอยู่กับประสิทธิภาพของกล้องด้วย ในกรณีนี้ ได้ทำการทดลองโดยใช้ไฟนีออน ได้ผลการทดลองดังนี้

4.1 ผลการทดลองการตรวจจับใบหน้าคน

การทดลองจับภาพใบหน้าคนในลักษณะต่างๆ โดยใช้โปรแกรมที่เขียนด้วยไลบรารีของโอเพินซีวี ได้ผลดังนี้



(ก) จับภาพหน้าตรง



(ข) จับภาพหน้าเอียงเล็กน้อย



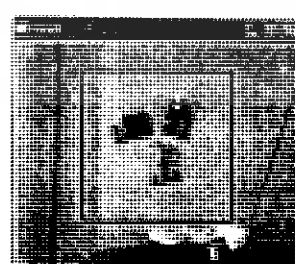
(ค) จับหน้าระยะไกล



(ง) ใบหน้าเอียงเกินไป



(จ) ใบหน้าอยู่ไกลเกินไป



(ฉ) ภาพที่คล้ายใบหน้าคน

รูปที่ 4.1 ผลการทดลองจับภาพใบหน้าคนในลักษณะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 (ก) (ข) และ (ค) พบว่าโปรแกรมสามารถตรวจจับใบหน้าคนได้ถูกต้อง เมื่อใบหน้าคนอยู่ในระยะที่เหมาะสม คือ กล้องสามารถจับใบหน้าได้เต็มและระยะห่างไม่เกิน 3.7 เมตร และไม่ทำมุมเอียงมากเกินไป (ประมาณ 45 องศา) ในรูปที่ 4.1 (ง) และ (จ) ไม่สามารถตรวจจับใบหน้าคนได้ เนื่องจากใบหน้าคนทำมุมเอียงกับกล้องมากเกินไป และอยู่ไกลเกินไป ตามลำดับ ส่วนรูปที่ 4.1 (ฉ) เกิดการผิดพลาดในการจับใบหน้าคน เนื่องจากบริเวณนั้นมีลักษณะที่คล้ายโครงสร้างของใบหน้าคน

4.2 ผลการทดลองการเคลื่อนที่ของหุ่นยนต์

การทดลองการเคลื่อนที่ของหุ่นยนต์นี้ ได้แบ่งการทดลองออกเป็น 2 ส่วนด้วยกัน คือ การทดลองบนสภาพพื้นผิวต่างๆ และการทดลองการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ

4.2.1 การทดลองบนสภาพพื้นผิวต่างๆ

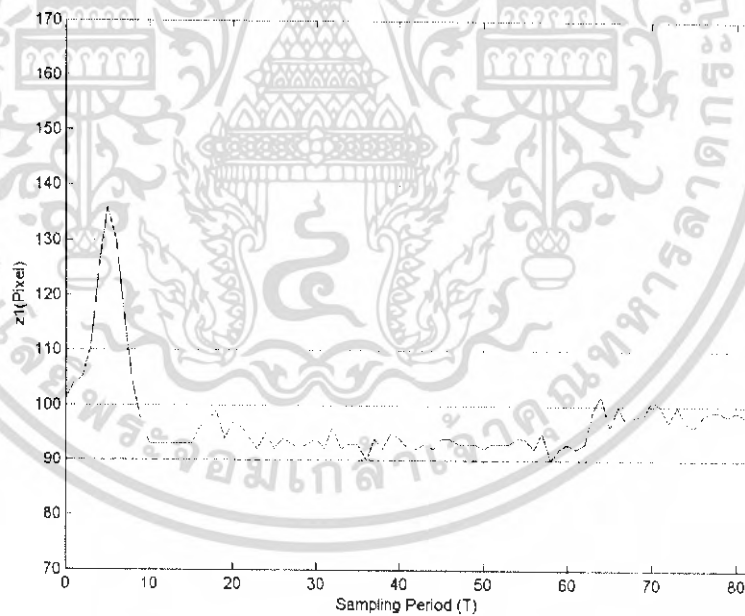
การทดลองในส่วนนี้ เพื่อความสามารถในการเคลื่อนที่ของหุ่นยนต์ บนสภาพพื้นผิวต่างๆ กัน คือ พื้นปูนซีเมนต์ พรม พื้นกระเบื้อง และพื้นปูนขัด พบว่าหุ่นยนต์สามารถเคลื่อนที่ได้เป็นอย่างดีบนพื้นเรียบ เช่น พื้นปูนขัด และมีการกระตุกเพียงเล็กน้อยเท่านั้น ส่วนการเคลื่อนที่บนสภาพพื้นอื่นๆ จะมีการติดขัดมาก เนื่องจากพื้นที่ทำการทดลองไม่เรียบพอ

4.2.2 การทดลองการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ

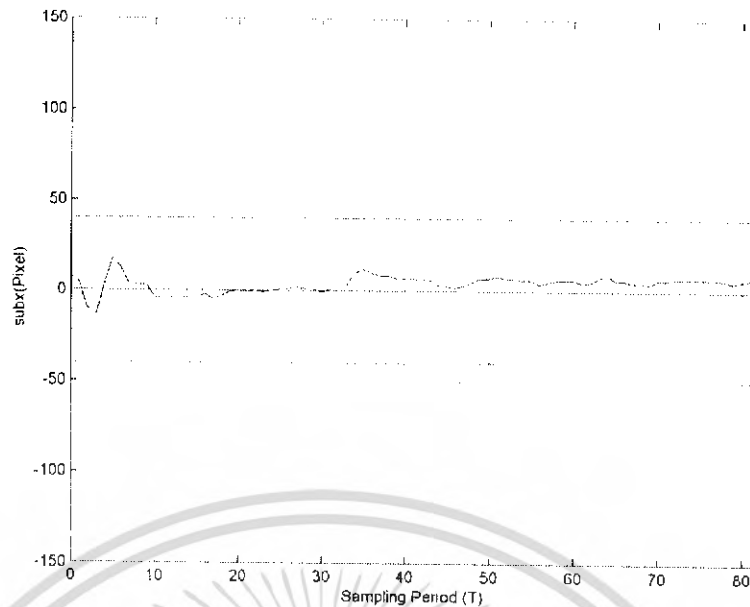
ในที่นี้ได้ออกแบบการทดลองให้หุ่นยนต์เคลื่อนที่เดินหน้า ถอยหลัง ซ้าย ขวา และ แนวทแยงมุม โดยพิจารณาผลตอบสนองในระยะเวลา 30 วินาที ในแต่ละกรณี เพื่อดูการเคลื่อนที่ของหุ่นยนต์ โดยที่หุ่นยนต์จะต้องปรับระยะห่างระหว่างคนกับหุ่นยนต์ให้ได้ตามที่กำหนด คือ ทำให้อาคารรอบสี่เหลี่ยมล้อมรอบหน้าคน (z_1) อยู่ในช่วง 90 - 110 พิกเซล หรือประมาณ 70 เซนติเมตร หรือ และจะต้องปรับให้จุดกึ่งกลางของใบหน้าคนอยู่ ณ จุดอ้างอิง พิกัด (180,140) โดยค่าความผิดพลาดตามแนวแกน x หรือ $subx$ มีค่าที่ยอมรับได้ในช่วง ± 40 พิกเซล จากจุดอ้างอิงที่ตำแหน่ง 180 พิกเซล ส่วนค่าความผิดพลาดตามแนวแกน y หรือ $suby$ ได้กำหนดให้มีขนาดเท่ากัน แต่เนื่องจากการเคลื่อนที่ของหุ่นยนต์เป็นการติดตามการเคลื่อนที่ของคนคนเดียวกัน ค่า $suby$ ซึ่งปรับตามความสูงของบุคคล จึงเปลี่ยนแปลงไม่มากนัก การเคลื่อนที่ในลักษณะต่างๆ มีรายละเอียดดังนี้

4.2.2.1 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ถอยหลัง

จากการที่ให้คนเดินหน้าเข้าหาหุ่นยนต์ ทำให้หุ่นยนต์ต้องถอยหลัง เพื่อปรับขนาดของกรอบสี่เหลี่ยมล้อมรอบใบหน้าคน ($z1$) ให้เป็นไปตามที่กำหนด คือ 90 – 110 พิกเซล ลักษณะการเคลื่อนที่ของหุ่นยนต์สามารถอธิบายได้ดังรูปที่ 4.2 คือ คนได้มีการก้าวเข้าหาหุ่นยนต์ ทำให้ค่า $z1$ เพิ่มขึ้น จากที่เวลาเริ่มต้น $z1$ มีค่าเท่ากับ 100 พิกเซล เปลี่ยนไปเป็น 136 พิกเซล โดยใช้เวลาในการก้าว 6 รอบการทำงาน หรือประมาณ 2.25 วินาที แล้วหยุดนิ่ง กล่าวคือ ใบหน้าคนมีขนาดใหญ่ขึ้น ทำให้หุ่นยนต์เริ่มเคลื่อนที่ถอยหลัง เพื่อลดค่าของ $z1$ ลงอย่างต่อเนื่อง จนมาถึงช่วงการทำงานที่ 9 ค่า $z1$ ณ จุดนี้ เป็น 104 ซึ่งถือว่าตรงตามข้อกำหนดแล้ว ต่อมาพิจารณาค่า $subx$ ในรูปที่ 4.3 ซึ่งพบว่า มีค่าอยู่ในช่วงที่กำหนดตลอดช่วงระยะเวลาการทำงาน เพราะเป็นการเคลื่อนที่ถอยหลังอย่างเดียว จึงทำให้ค่า $subx$ เปลี่ยนเพียงเล็กน้อยเท่านั้นเมื่อพิจารณาทั้งสองรูปพบว่าค่า $z1$ และ $subx$ อยู่ในช่วงที่กำหนดในช่วงการทำงานที่ 9 (ประมาณ 3.375 วินาที) จึงทำให้หุ่นยนต์หยุดการเคลื่อนที่ อย่างไรก็ตาม จะเห็นว่าทั้งค่า $z1$ และ $subx$ เปลี่ยนแปลงอยู่ตลอดเวลา เนื่องจากการที่ใบหน้าคนจะอยู่นิ่งกับที่ตลอดเวลานั้นเป็นไปได้ยาก แต่ทราบได้ที่ใบหน้าคนมีการเคลื่อนที่แล้วยังรักษาค่า $z1$ และ $subx$ ให้อยู่ในช่วงที่กำหนดได้ หุ่นยนต์ก็จะไม่มีการเคลื่อนที่



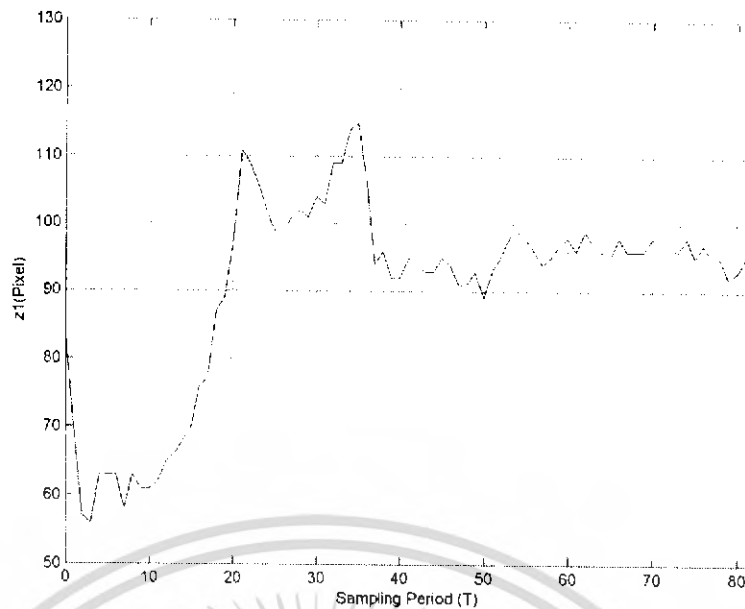
รูปที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่ถอยหลัง



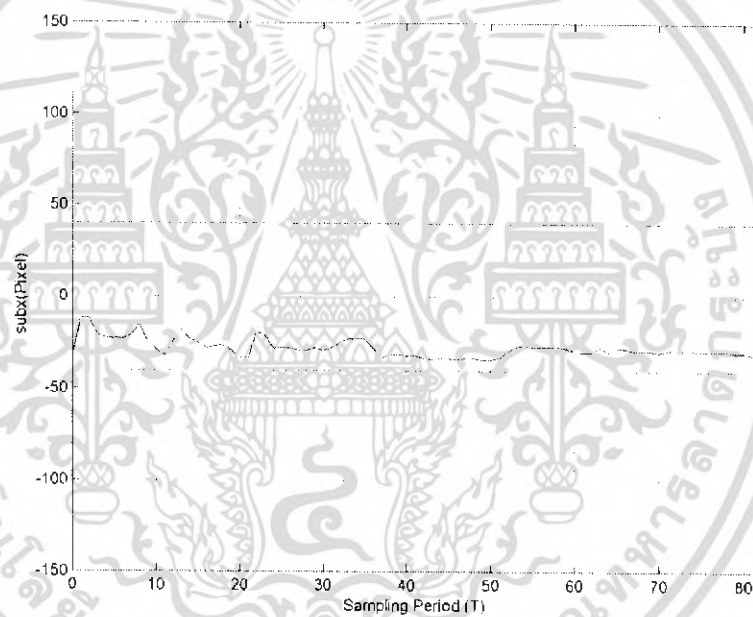
รูปที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ถอยหลัง

4.2.2.2 ผลการทดลองให้หุ่นยนต์เคลื่อนที่เดินหน้า

จากการที่ให้คนถอยออกจากหุ่นยนต์ ทำให้หุ่นยนต์ต้องเดินหน้า เพื่อปรับขนาดของกรอบสี่เหลี่ยมล้อมรอบใบหน้าคน ($z1$) ให้เป็นไปตามที่กำหนด คือ 90 – 110 พิกเซล ลักษณะการเคลื่อนที่ของหุ่นยนต์สามารถอธิบายได้ดังรูปที่ 4.4 คือ คนได้มีการก้าวถอยออกจากหุ่นยนต์ ทำให้ค่า $z1$ ลดลง จากที่เวลาเริ่มต้น $z1$ มีค่าเท่ากับ 84 พิกเซล เปลี่ยนไปเป็น 56 พิกเซล โดยใช้เวลาในการก้าว 4 รอบการทำงาน หรือประมาณ 1.5 วินาที แล้วหยุดนิ่ง กล่าวคือ ใบหน้าคนมีขนาดเล็กลง ทำให้หุ่นยนต์เริ่มเคลื่อนที่เดินหน้า เพื่อเพิ่มค่าของ $z1$ ขึ้นอย่างต่อเนื่อง จนมาถึงช่วงการทำงานที่ 19 ค่า $z1$ ณ จุดนี้ เป็น 90 ซึ่งถือว่าตรงตามข้อกำหนดแล้ว ซึ่งหุ่นยนต์ควรจะหยุดการเคลื่อนที่ แต่เกิดการกระตุกขึ้นในช่วงการทำงานที่ 21 35 และ 50 ซึ่งน่าจะมีสาเหตุมาจากเกิดการสั่นของกล้องเว็บแคมมากขณะหุ่นยนต์เคลื่อนที่ เพราะค่าที่เกินออกมาจากช่วงความผิดพลาดที่ยอมรับได้นั้นน้อยมาก ต่อมาพิจารณาค่า $subx$ ในรูปที่ 4.5 ซึ่งพบว่าตลอดช่วงการทำงาน $subx$ มีค่าเป็นไปตามข้อกำหนดตลอดช่วงเวลาการทำงาน เพราะเป็นการเคลื่อนที่เดินหน้าอย่างเดียว จึงทำให้ค่า $subx$ เปลี่ยนเพียงเล็กน้อยเท่านั้น เมื่อพิจารณาทั้งสองรูปพบว่าค่า $z1$ และ $subx$ อยู่ในช่วงที่กำหนดในช่วงการทำงานที่ 50 (ประมาณ 18.75 วินาที) จึงทำให้หุ่นยนต์หยุดการเคลื่อนที่



รูปที่ 4.4 กราฟแสดงความสัมพันธ์ระหว่างค่า z_1 กับเวลาสุ่มของการเคลื่อนที่เดินหน้า

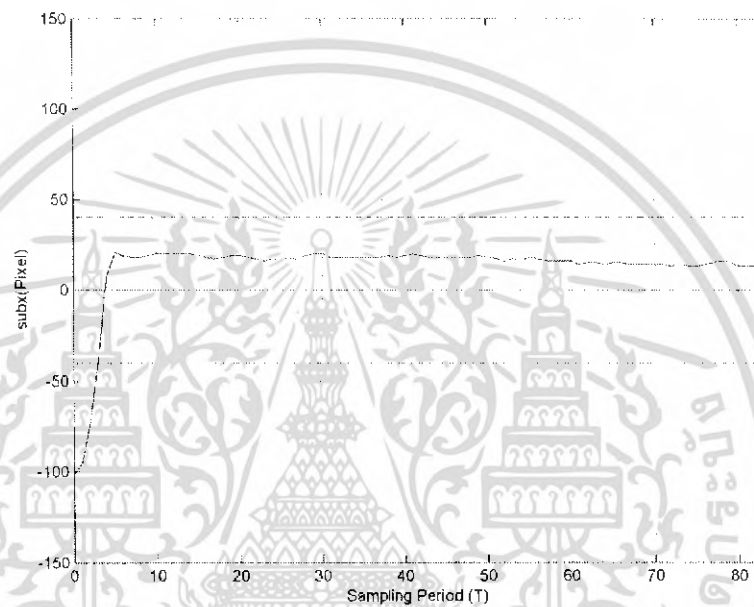


รูปที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่เดินหน้า

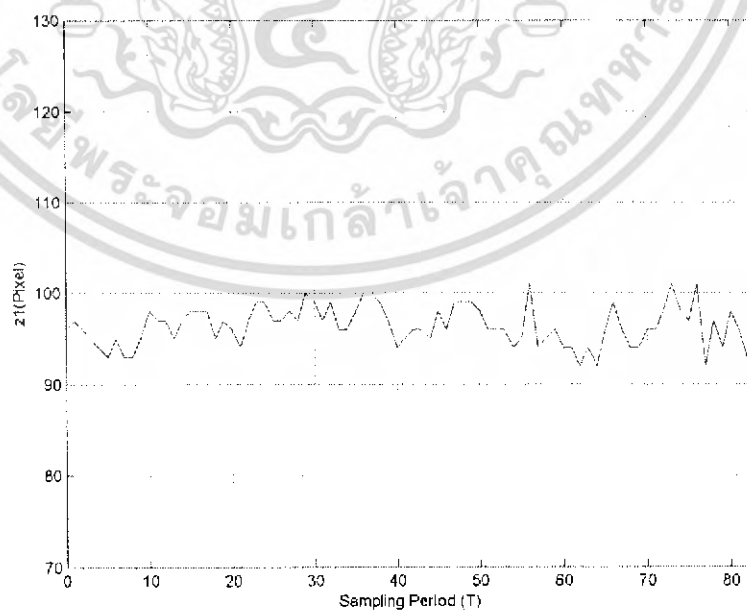
4.2.2.3 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ไปทางขวา

จากการที่ให้คนเดินไปทางซ้าย ทำให้หุ่นยนต์ต้องเคลื่อนที่ไปทางขวา เพื่อปรับให้จุดกึ่งกลางของกรอบสี่เหลี่ยมล้อมรอบใบหน้าคนมาอยู่ที่จุดอ้างอิง หรือ $subx$ มีค่าอยู่ในช่วง ± 40 พิกเซล ลักษณะการเคลื่อนที่ของหุ่นยนต์สามารถอธิบายได้ดังรูปที่ 4.6 คือ คนได้มีการเคลื่อนที่ไปทางซ้ายแล้วหยุดนิ่ง ณ ตำแหน่งที่ทำให้ $subx$ มีค่าเท่ากับ -100 พิกเซล ทำให้หุ่นยนต์เริ่มเคลื่อนที่ไปทางขวา เพื่อปรับให้ค่า $subx$ มาอยู่ในช่วง ± 40 พิกเซลตามที่กำหนดไว้ จนมาถึงประมาณช่วงการทำงานที่ 5 ที่ $subx$ มีค่าเป็น -39 ซึ่งอยู่ในช่วงที่กำหนด ต่อมาพิจารณาค่า z_1 ในรูปที่ 4.7 พบว่ามีค่าอยู่ในช่วงที่กำหนดตลอดการทำงาน เนื่องจากการเคลื่อนที่ในแนวซ้ายขวาอย่างเดียว จึงทำให้ออกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า $z1$ เปลี่ยนเพียงเล็กน้อยเท่านั้น เมื่อพิจารณาทั้งสองรูปพบว่าค่า $z1$ และ $subx$ อยู่ในช่วงที่กำหนด ในช่วงการทำงานที่ 5 (ประมาณ 1.875 วินาที) จึงทำให้หุ่นยนต์หยุดการเคลื่อนที่ อย่างไรก็ตาม การที่ค่า $subx$ และ $z1$ ในรูปที่ 4.6 และ 4.7 ไม่คงที่นั้น มิได้หมายความว่า หุ่นยนต์มีการเคลื่อนที่ หากแต่เป็นการที่ใบหน้าคนไม่สามารถอยู่นิ่งสนิทได้ตลอดเวลา ซึ่งมีได้ทำให้ค่า $subx$ และ $z1$ ผิดไปจากข้อกำหนดแต่อย่างใด สังเกตได้ว่า การเคลื่อนที่ของหุ่นยนต์ในครั้งนี้ไม่มีความผิดพลาดใดๆ ทั้งสิ้น กล่าวคือหุ่นยนต์ได้เคลื่อนที่ไปในทิศทางที่ถูกต้องแม่นยำโดยไม่มีการกระตุก อาจเนื่องจาก พื้นบริเวณนั้นมีความเรียบสนิท และเกิดการสั่นของกล้องเว็บแคมเพียงเล็กน้อยเท่านั้น



รูปที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ไปทางขวา

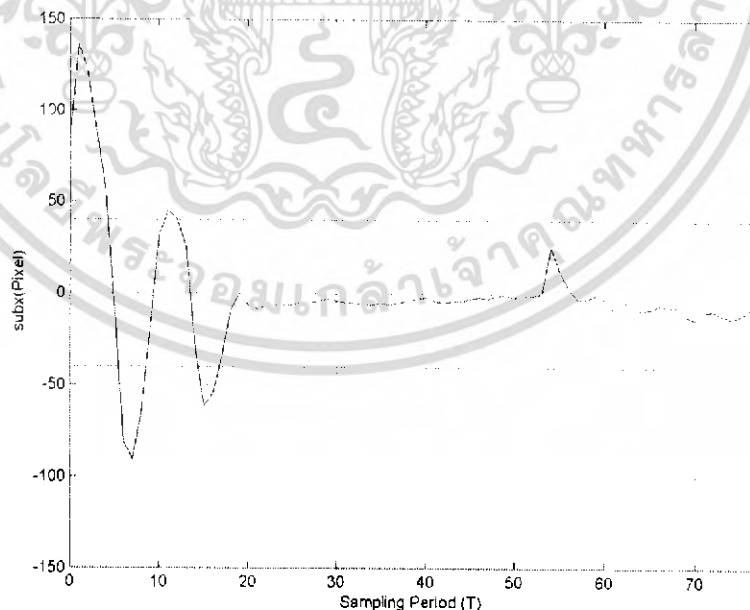


รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่ไปทางขวา

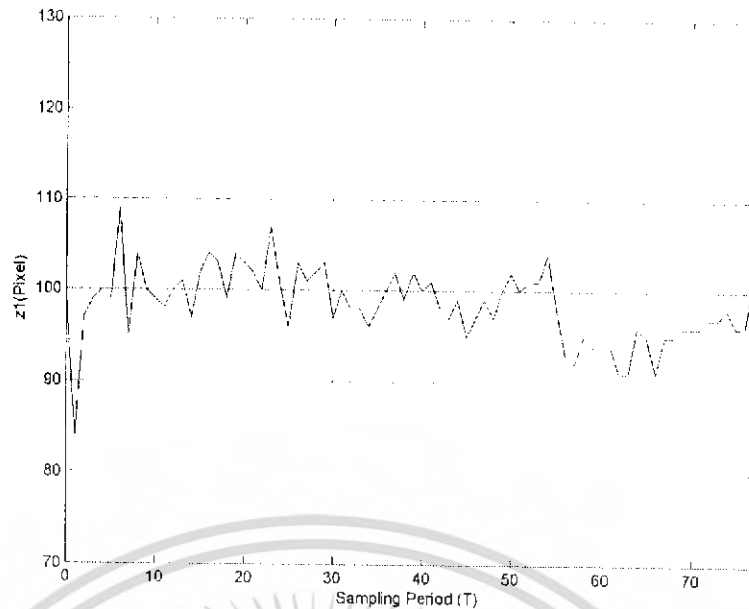
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.4 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ไปทางซ้าย

จากการที่ให้คนเคลื่อนที่ไปทางขวา ทำให้หุ่นยนต์ต้องเคลื่อนที่ไปทางซ้าย เพื่อปรับให้จุดกึ่งกลางของกรอบสี่เหลี่ยมล้อมรอบใบหน้าคนมาอยู่ที่จุดอ้างอิง หรือ $subx$ มีค่าอยู่ในช่วง ± 40 พิกเซล ลักษณะการเคลื่อนที่ของหุ่นยนต์สามารถอธิบายได้ดังรูปที่ 4.8 คือ คนได้มีการเคลื่อนที่ไปทางขวา จนทำให้จากที่เวลาเริ่มต้น $subx$ มีค่าเท่ากับ 82 พิกเซล เปลี่ยนเป็น 136 โดยใช้เวลาในการก้าว 1 รอบการทำงาน หรือประมาณ 0.375 วินาที แล้วหยุดนิ่ง ทำให้หุ่นยนต์เริ่มเคลื่อนที่ไปทางซ้าย เพื่อปรับให้ค่า $subx$ มาอยู่ในช่วง ± 40 พิกเซลตามที่กำหนดไว้ แต่พบว่าเกิดการกระตุกในช่วงการทำงานที่ 5 ถึง 15 ซึ่งอาจมีสาเหตุมาจากสภาพพื้นผิวบริเวณนั้นไม่เรียบพอ หรืออาจเกิดจากการที่ L298 ถูกใช้งานเป็นเวลานานติดต่อกัน ทำให้ร้อนมาก จึงส่งผลกระทบต่อการทำงานของมอเตอร์ นอกจากนี้ ยังอาจเกิดได้จากการติดตั้งล้อของหุ่นยนต์ของผู้จัดทำโครงงาน ที่ขาดความชำนาญทางด้านกลศาสตร์ด้วย แต่หากพิจารณา ณ ช่วงเวลาการทำงานที่ 11 พบว่าค่า $subx$ เกินจากค่าที่ยอมรับได้มาเพียงเล็กน้อยเท่านั้น ดังนั้น การกระตุกในช่วงนี้ น่าจะมีสาเหตุจากการสั่นของล้อเว็บแคมขณะเคลื่อนที่มากกว่า หุ่นยนต์สามารถปรับให้ค่า $subx$ จะเป็นไปตามได้ในช่วงเวลาที่ 18 ทำให้หุ่นยนต์หยุดการเคลื่อนที่ในแนวซ้ายขวา ต่อมาพิจารณาค่า $z1$ ในรูปที่ 4.9 พบว่าหลังจากคาบการทำงานที่ 2 แล้ว ค่า $z1$ มีค่าอยู่ในช่วงที่กำหนดตลอดการทำงาน เนื่องจากการเคลื่อนที่ในแนวซ้ายขวาวงเดียว จึงทำให้ค่า $z1$ เปลี่ยนเพียงเล็กน้อยเท่านั้น ดังนั้นเมื่อพิจารณาทั้งสองรูปพบว่าค่า $z1$ และ $subx$ อยู่ในช่วงที่กำหนดในช่วงการทำงานที่ 18 (ประมาณ 6.75 วินาที) จึงทำให้หุ่นยนต์หยุดนิ่ง



รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่ไปทางซ้าย



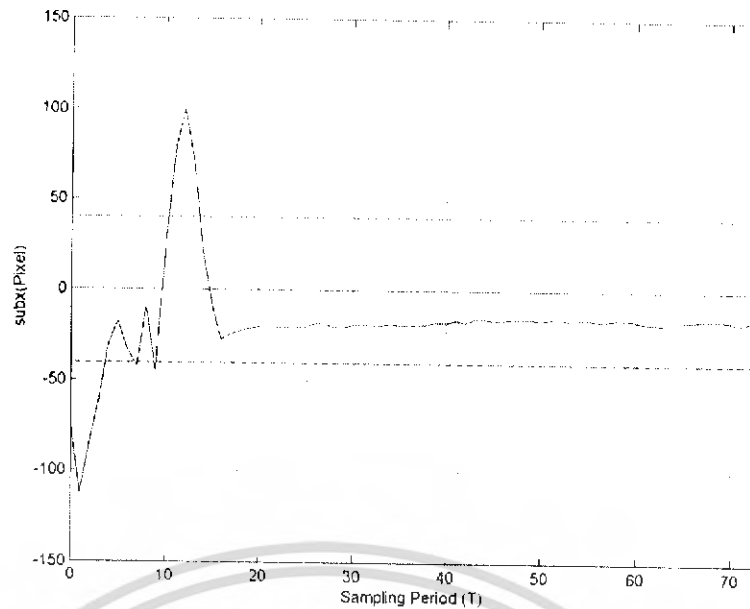
รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างค่า z_1 กับเวลาสุ่มของการเคลื่อนที่ไปทางซ้าย

4.2.2.5 ผลการทดลองให้หุ่นยนต์เคลื่อนที่ในแนวทแยง

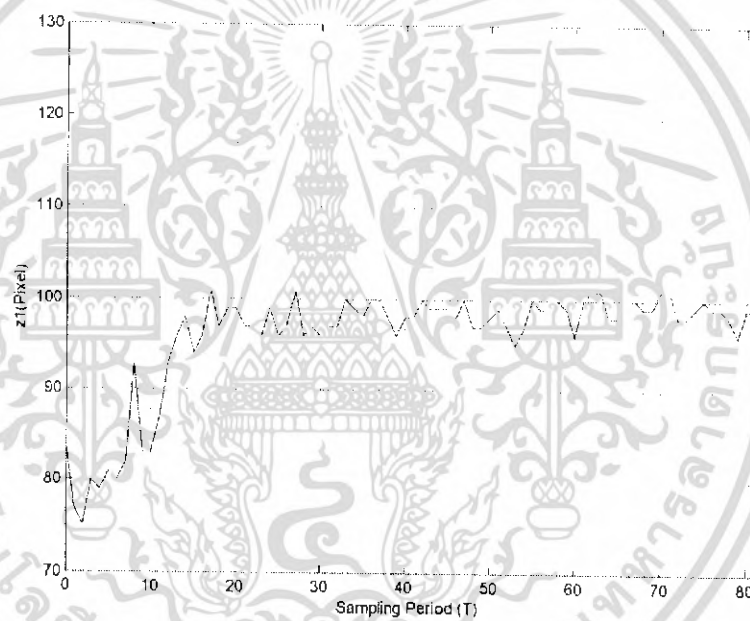
จากรูปที่ 4.10 จะสังเกตได้ว่า ณ เวลาเริ่มต้น $subx$ มีค่าเป็น -73 พิกเซล แสดงว่าหุ่นยนต์จะต้องเคลื่อนที่ไปทางขวาเพื่อปรับค่า $subx$ ให้อยู่ในช่วง ± 40 พิกเซล และจากรูปที่ 4.11 ณ เวลาเริ่มต้น z_1 มีค่าเท่ากับ 84 แสดงว่าหุ่นยนต์จะต้องเดินหน้าเพื่อปรับให้ z_1 มีค่าอยู่ในช่วง 90 - 110 พิกเซล กล่าวคือ ในกรณีนี้ หุ่นยนต์จะต้องเคลื่อนที่เดินหน้าเฉียงไปทางขวานั้นเอง

การเคลื่อนที่ของหุ่นยนต์สามารถอธิบายได้โดยพิจารณาจากรูปที่ 4.11 พบว่าในรอบการทำงานที่ 4 z_1 มีค่าเป็น 79 ซึ่งหมายความว่าระยะระหว่างคนกับหุ่นยนต์ยังไกลเกินไป หุ่นยนต์จึงเคลื่อนที่เดินหน้าต่อไป เพื่อที่จะทำให้ค่า z_1 จะเป็นไปตามข้อกำหนด ในขณะที่เดียวกันเมื่อพิจารณาจากรูปที่ 4.10 พบว่าค่า $subx$ มีค่าเป็น -30 ส่วน หุ่นจึงต้องเคลื่อนที่ไปทางขวา เพื่อที่จะทำให้ค่า $subx$ อยู่ในช่วงที่กำหนด แต่พบว่าขณะที่หุ่นยนต์เดินหน้า เกิดการกระตุกขึ้นในช่วงการทำงานที่ 9 ถึง 10 ซึ่งจะเห็นได้จากรูปที่ 4.10 และเมื่อพิจารณารูปที่ 4.11 การกระตุกนี้ ส่งผลกระทบต่อค่า $subx$ ด้วย คือ ทำให้ค่าเปลี่ยนไปเป็น 100 พิกเซล หุ่นยนต์จึงเคลื่อนที่มาทางซ้ายเล็กน้อย ก่อนที่จะหยุดนิ่งในช่วงการทำงานที่ 13 (ประมาณ 4.875 วินาที) เพราะทั้งค่า $subx$ และ z_1 ได้เป็นไปตามข้อกำหนดแล้ว

สังเกตว่า ในการทดลองครั้งนี้ หุ่นยนต์มีการกระตุกซ้ายขวาเล็กน้อยในช่วงการทำงานที่ 8 ถึง 10 ซึ่งอาจเนื่องมาเกิดการสั่นของกล้องเว็บแคมอย่างมากขณะที่หุ่นยนต์เคลื่อนที่ และเป็นการเคลื่อนที่พร้อมกันทั้ง 2 ทิศทางนั่นเอง



รูปที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า $subx$ กับเวลาสุ่มของการเคลื่อนที่แนวทแยง



รูปที่ 4.11 กราฟแสดงความสัมพันธ์ระหว่างค่า $z1$ กับเวลาสุ่มของการเคลื่อนที่แนวทแยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลอง พบว่าการระบุตำแหน่งของใบหน้าคนสามารถทำได้ดีในบริเวณที่มีแสงสว่างเพียงพอ เช่น แสงจากไฟนีออน แต่ยังมีข้อผิดพลาดอยู่เล็กน้อย หากในบริเวณนั้นมีส่วนที่มีโครงสร้างคล้ายกับใบหน้าคนอยู่ด้วย นอกจากนี้หากคนมีการเคลื่อนที่เร็วเกินไป หรือทำมุมเอียงกับกล้องมากเกินไป จะทำให้กล้องเว็บแคมไม่สามารถจับภาพใบหน้าคนได้ ในส่วนของการเคลื่อนที่ พบว่าหุ่นยนต์สามารถทำงานได้เป็นที่น่าพอใจ แต่มีการกระตุกบ้าง เนื่องจากสภาพพื้นผิวที่ให้หุ่นยนต์เคลื่อนที่ไม่เรียบพอ อีกทั้ง กล้องเว็บแคมมีการสั่นมากขณะที่หุ่นยนต์เคลื่อนที่ จึงทำให้การระบุตำแหน่งฟิกเชลผิดพลาดไป

5.2 ปัญหาที่พบและแนวทางการแก้ไข

1. การติดตั้งไลบรารีของโอเพ่นซีวีนั้นซับซ้อนมาก และเนื่องจากการใช้งานต้องใช้ร่วมกับโปรแกรมอื่นๆที่ผู้ทำโครงการไม่มีความรู้มาก่อน เช่น Visual C++ 6, Visual C++.NET 2003 อีกทั้งคู่มือการติดตั้งที่ให้มาไม่มีความละเอียด ทำให้ต้องไปปรึกษาอาจารย์หลายท่าน การติดตั้งให้สมบูรณ์จึงต้องใช้เวลานาน
2. ล้อของหุ่นยนต์ไม่เสมอกับพื้น เนื่องจากผู้ทำโครงการขาดความชำนาญในการออกแบบโครงสร้างทางกล ทำให้ต้องเสียเวลาในการปรับให้ล้อทั้งหมดเสมอกับพื้น
3. กล้องเว็บแคมจับภาพได้ช้า ทำให้บางครั้งไม่สามารถจับหน้าคนที่กำลังเคลื่อนไหวย่างรวดเร็วได้

5.3 แนวทางในคันคว้าและการพัฒนา

1. เพื่อแก้ปัญหาที่ล้อของหุ่น ไม่สัมผัสพื้น จึงควรที่จะใช้ล้อออมนิ 3 คู่ วางทำมุมกัน 120° ล้อทุกล้อจึงจะสามารถสัมผัสพื้นได้ แต่การควบคุมการเคลื่อนที่จะยากขึ้น คือต้องควบคุมความเร็วของแต่ละล้อในการเคลื่อนที่ไปยังทิศทางต่างๆ
2. การส่งข้อมูลจากแบบไร้สาย (Wireless) จะทำให้การใช้งานสะดวกขึ้น ไม่ต้องนำคอมพิวเตอร์โน้ตบุคไปวางไว้บนหุ่นยนต์ และยังเป็นลดขนาดและน้ำหนักของหุ่นยนต์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ประจัน พงษ์สันติกุล. เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุม ไมโครคอนโทรลเลอร์ PIC. พระราชบัญญัติลิขสิทธิ์ พ.ศ. 2521. กรุงเทพมหานคร: บริษัท อินโนติฟ เอ็กเพอริเมนต์ จำกัด.
- [2] นรุตม์ วิทยภัฏวดีพงษ์, นพพงษ์ เมธาประสิทธิ์. “ยานสำรวจใต้น้ำควบคุมด้วยเครื่องควบคุมระยะไกล.” ปรินญาณิพนธ์ปริญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546.
- [3] ศัทธิตักดี มณีแสง. “ระบบควบคุมตำแหน่งลูกบอลบนคาน.” ปรินญาณิพนธ์ปริญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2544.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้งไลบรารีของโอเพินซีวี

1. ลงโปรแกรม Microsoft Visual C++ 6.0
2. ลงโปรแกรม Microsoft DirectX 9.0 SDK (Summer 2004) โดยมีขั้นตอนดังนี้
 - 2.1 ดาวน์โหลดได้จาก www.msdn.microsoft.com/directx/
 - 2.2 เลือกลงพร้อมกับตัวอย่าง

Together with samples

Note ในการใช้ OpenCV Beta 5 จะต้องทำการ build library 2 ตัว คือ strmbased.lib และ strmbas.lib โดย library 2 ตัวนี้เป็น library ของ DirectXSDK ดังนั้นเราจึงต้องทำการติดตั้ง DirectXSDK ตามข้อ 2 ที่ได้กล่าวข้างต้น
3. ลงโปรแกรม Microsoft Visual C++.NET 2003

Note ในกรณีที่ท่านใช้ DirectXSDK เวอร์ชันที่สามารถ build library 2 ตัวดังกล่าวได้ โดยใช้ Microsoft Visual C++ 6.0 ก็ไม่จำเป็นต้องลง Microsoft Visual.NET 2003 อีก
4. หลังจากลงโปรแกรม Microsoft DirectX 9.0 SDK (Summer 2004) เรียบร้อยแล้วให้ทำตามขั้นตอนดังนี้
 - 4.1 ในกรณีที่ใช้ Microsoft Visual C++.NET 2003 ให้เปิด file ที่ชื่อ


```
<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses.sln
```

 หรือ


```
<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses.dsw
```

 ในกรณีที่ใช้ Microsoft Visual C++ 6.0 ให้เปิด file ที่ชื่อ


```
<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses.dsw
```
 - 4.2 ไปที่ Build -> Batch Build -> เลือก Debug และ Release Configurations -> Build
 - 4.3 Copy strmbased.lib จาก


```
<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses\debug
```

 และ strmbas.lib จาก


```
<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses\release
```

 ไปไว้ที่


```
<DirectXSDK>\lib
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 เปิด file

<DirectXSDK>\samples\C++\DirectShow\BaseClasses\baseclasses.{sln/dsw}

4.5 ไปที่ Tools -> Options -> Directories -> Include files แล้วเพิ่ม

<DirectXSDKInstallFolder>\include และ

<DirectXSDKInstallFolder>\samples\C++\DirectShow\BaseClasses เข้าไป

4.6 ไปที่ Tools -> Options -> Directories -> Library files แล้วเพิ่ม

<DirectXSDKInstallFolder>\lib เข้าไป

Note การทำข้อ 4.5 และ 4.6 ให้เอา file ที่เราเพิ่มเข้าไปไว้บนสุด

5. ลงโปรแกรม OpenCV Beta 5 ซึ่งดาวน์โหลดได้จาก www.sourceforge.net จากนั้นต้องทำการติดตั้งดังนี้

5.1 เพิ่ม c:\opencv\bin ใน System Path ซึ่งบน Windows2000/XP สามารถทำได้ดังนี้

MyComputer -> [click ขวา] -> Properties -> Advanced -> Environment Variables.

5.2 ใช้ Microsoft Visual C++ 6.0 เปิด file ชื่อ <opencv>_make\opencv.dsw เพื่อทำการ build

ตัวโปรแกรมหลักของ OpenCV

5.3 ไปที่ Menu Build -> Batch Build -> เลือก เฉพาะ **Debug** กับ **Release Configurations**

เท่านั้น -> Build ในขั้นตอนนี้อาจต้องแยกทำ 2 ครั้ง ทำทีละครั้งเนื่องจาก โปรแกรม batch build ที่เดียวทั้งหมดไม่ได้

5.4 ปิด file นี้แล้วเปิด file ชื่อ <opencv>_make\opencv_directshow.dsw

5.5 ไปที่ menu Build -> Batch Build -> เลือก เฉพาะ **Debug** และ **Release Configurations** เท่านั้น -> Build

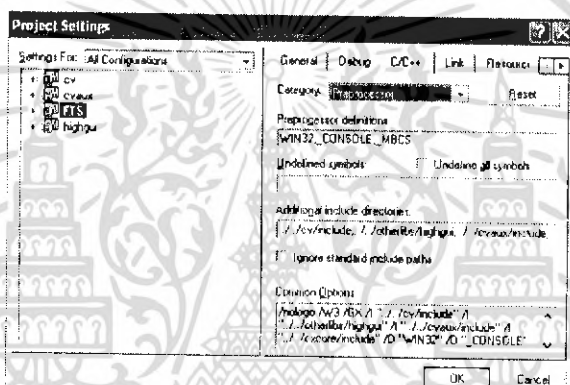
Note ทุกครั้งที่มีการลงโปรแกรม OpenCV ใหม่จะต้องทำตามขั้นตอนที่ 5 นี้ใหม่ทั้งหมด

ภาคผนวก ข

การรันโปรแกรมที่ใช้ไลบรารีของโอเพ่นซีวี

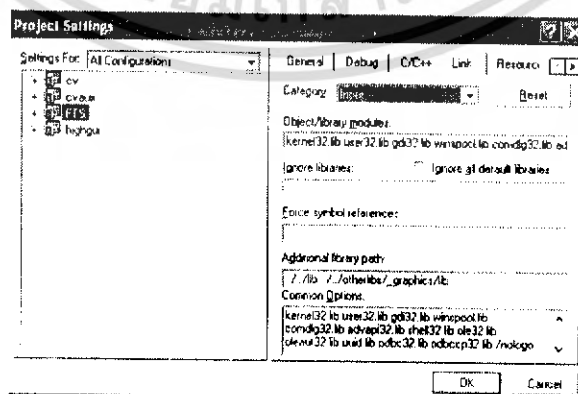
ทุกครั้งที่เริ่มโปรเจกใหม่ที่ใช้ไลบรารีของโอเพ่นซีวี จะต้องทำการเพิ่มพาร์ (Path) และตั้งค่าต่างๆ ตามขั้นตอนต่อไปนี้

1. ไปที่ Menu Project -> Settings -> Setting For (หน้าต่างซ้าย) -> All Configurations -> เลือกที่ชื่อ project -> C/C++ (หน้าต่างขวา) -> Category -> Preprocessor -> Additional include directories: ใส่ `../../cv/include,../../otherlibs/highgui,../../cvaux/include,../../cxcore/include` (แยกด้วยเครื่องหมาย ,)



รูปที่ ข.1 การตั้งค่าในส่วนของ All Configurations Preprocessor

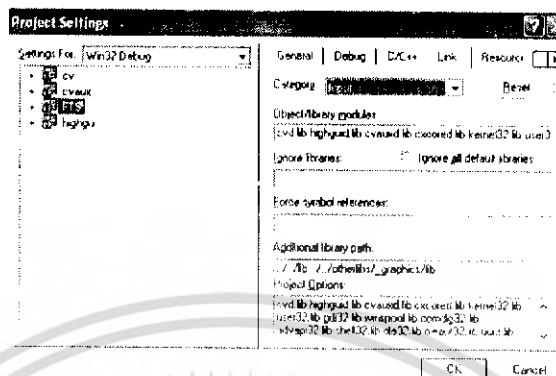
2. ไปที่ Menu Project -> Settings -> Setting For (หน้าต่างซ้าย) -> All Configurations -> เลือกที่ชื่อ project -> Link (หน้าต่างขวา) -> Category -> Input -> Additional library path: ใส่ `../../lib ../../otherlibs/_graphics/lib` (แยกด้วย ช่องว่าง)



รูปที่ ข.2 การตั้งค่าในส่วนของ All Configurations Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไปที่ Menu Project -> Settings -> Setting For (หน้าต่างซ้าย) -> Win32 Debug -> เลือกที่ชื่อ project -> Link (หน้าต่างขวา) -> Category -> General -> Object/library modules: เพิ่ม cvd.lib highguid.lib cvauxd.lib excored.lib (แยกด้วย ช่องว่าง)



รูปที่ ข.3 การตั้งค่าในส่วนของ Debug Input

4. ไปที่ Menu Project -> Settings -> Setting For (หน้าต่างซ้าย) -> Win32 Release -> เลือกที่ชื่อ project -> Link (หน้าต่างขวา) -> Category -> General -> Object/library modules: เพิ่ม cv.lib highgui.lib cvaux.lib excore.lib (แยกด้วย ช่องว่าง)



รูปที่ ข.4 การตั้งค่าในส่วนของ Release Input

5. Click "OK"

6. ต่อมาต้องทำการเพิ่มโปรเจกต์ลงในงานของเรา โดย

6.1 Menu -> Project -> Insert Project Into Workspace

6.2 เลือกไฟล์ ดังนี้ c:\opencv\cv\src\cv.dsp

c:\opencv\cvaux\src\cvaux.dsp

c:\opencv\otherlibs\highgui\highgui.dsp

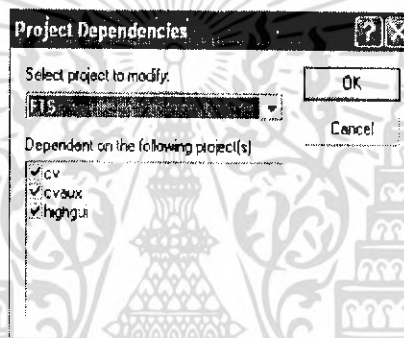
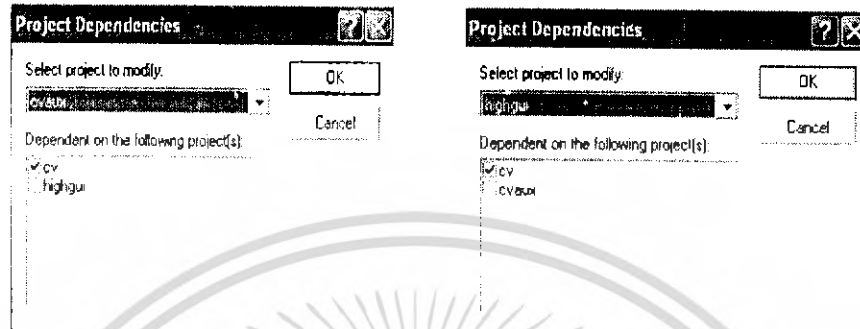
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ไปที่ Menu -> Project -> Dependencies

สำหรับ "cvaux" เลือก "cv",

สำหรับ "highgui" เลือก "cv",

สำหรับชื่อ project ที่เราตั้งเลือก "cv", "cvaux", "highgui"



รูปที่ ข.5 การตั้งค่าในส่วนของ Project Dependencies


8. กด "OK" ซึ่งถือว่าจบการเซ็ทค่าต่างๆแล้ว

9. ต่อมาในการ Build โปรเจ็คของเรา ไปที่ Menu -> Build -> Batch Build แล้ว เลือก build เฉพาะ Debug และ Release Configurations เท่านั้น จากนั้น

10. เมื่อไม่พบ error ในการ build หมายความว่า โปรแกรมได้ทำการสร้างโปรเจ็คของเราแล้ว ต่อมาต้องทำการ execute ซึ่งจะพบว่า เมื่อกด execute จะมี window ขึ้นมาให้ใส่ชื่อไฟล์ที่ต้องการทำการ execute ตรงนี้ให้ไปที่ Browse -> แล้วไปที่เพิ่มโปรเจ็คของเรา -> เพิ่ม Debug -> เลือก execute ไฟล์เป็นชื่อเดียวกับโปรเจ็ค ซึ่งทั้งเพิ่มนั้นก็จะมีอยู่เพียงไฟล์เดียวเท่านั้น

11. เมื่อกด execute แล้ว จะมีหน้าต่างต่างๆกระพริบขึ้นมาแล้วหายไป นั้นหมายความว่าโปรแกรมได้ทำการ execute เรียบร้อยแล้ว แต่การ run โปรเจ็คของเรายังไม่สามารถทำได้ เนื่องจากเราต้องทำ

การเพิ่มฐานข้อมูลที่เป็นของใบหน้าคนลงไปเสียก่อน ซึ่งทำได้ในข้อถัดไป
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้ณาไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. คัดลอก (copy) ไฟล์ชื่อ haarcascade_frontalface_alt.xml ที่มีลักษณะ  จากแฟ้ม OpenCV/apps/Haartraining ไปไว้ในแฟ้มเดียวกับ execute file ของเรา

13. คลิกขวาที่ execute file -> Edit ใน Notepad หรือ Editor ตัวใดก็ได้ -> พิมพ์ ชื่อ project แล้วตามด้วย --cascade="haarcascade_frontalface_alt.xml" เช่น หาก project ชื่อ facedetect ก็ ให้พิมพ์ ดังนี้ facedetect --cascade="haarcascade_frontalface_alt.xml"

14. save as เป็นชื่อ project.cmd เช่น facedetect.cmd เอาไว้ใน แฟ้มเดียวกับ execute file ของเรา
สังเกตจะเห็นว่า icon จะเป็นรูปเกียร์ 

15. ในการเรียกใช้ project จะต้องทำการต่อกล้องและ port serial ให้เรียบร้อย แล้ว double click ที่ .cmd icon (icon รูปเกียร์) ที่สร้างไว้ในข้อ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมสำหรับการจับใบหน้าคน

```

#ifndef _EiC
#include "cv.h"
#include "highgui.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <math.h>
#include <float.h>
#include <limits.h>
#include <time.h>
#include <ctype.h>
#include <windows.h>
#endif
#ifdef _EiC
#define WIN32
#endif

static CvMemStorage* storage = 0;
static CvHaarClassifierCascade* cascade = 0;

void detect_and_draw( IplImage* image );

const char* cascade_name = "haarcascade_frontalface_alt.xml";

#define BUFSIZE 1200
char *pcCommPort = "COM2";
unsigned char buffer[100];

DWORD inbuffer=100;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DWORD outbuffer=100;
DWORD dwBytesRead, dwBytesWritten, dwBufSize=BUFSIZE;
HANDLE hCom;
BOOL fSuccess;

```

```

int conf_port(void)

```

```

{
DCB dcb;
hCom = CreateFile( pcCommPort,
GENERIC_READ | GENERIC_WRITE,
0, // must be opened with exclusive-access
NULL, // no security attributes
OPEN_EXISTING, // must use OPEN_EXISTING
FILE_ATTRIBUTE_NORMAL, // normal file
NULL // hTemplate must be NULL for comm devices
);
if (hCom == INVALID_HANDLE_VALUE)
{
// Handle the error.
printf("CreateFile failed with error %d.\n", GetLastError());
return (1);
}

```

```

// Build on the current configuration, and skip setting the size
// of the input and output buffers with SetupComm.

```

```

fSuccess = GetCommState(hCom, &dcb);

```

```

if (!fSuccess)

```

```

{

```

```

// Handle the error.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("GetCommState failed with error %d.\n", GetLastError());
return (2);
}
```

```
# Fill in DCB: 57,600 bps, 8 data bits, no parity, and 1 stop bit.
```

```
deb.BaudRate = CBR_9600; // set the baud rate
```

```
deb.ByteSize = 8; // data size, xmit, and rev
```

```
deb.Parity = NOPARITY; // no parity bit
```

```
deb.StopBits = ONESTOPBIT; // one stop bit
```

```
fSuccess = SetCommState(hCom, &deb);
```

```
if (!fSuccess)
```

```
{
```

```
    // Handle the error.
```

```
    printf("SetCommState failed with error %d.\n", GetLastError());
```

```
    return (3);
```

```
}
```

```
printf("Serial port %s successfully reconfigured.\n", pcCommPort);
```

```
return(0);
```

```
}
```

```
int main( int argc, char** argv )
```

```
{
```

```
    CvCapture* capture = 0;
```

```
    IplImage *frame, *frame_copy = 0;
```

```
    int optlen = strlen("--cascade=");
```

```
    const char* input_name;
```

```
    int k;
```

```
    k=conf_port();
```

```
    fSuccess=SetupComm(hCom, inbuffer,outbuffer);
```

```
    if( argc > 1 && strncmp( argv[1], "--cascade=", optlen ) == 0 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    cascade_name = argv[1] + optlen;
    input_name = argc > 2 ? argv[2] : 0;
}
else
{
    fprintf( stderr,
        "Usage: facedetect --cascade=\"<cascade_path>\" [filename|camera_index]\n" );
    return -1;
    /*input_name = argc > 1 ? argv[1] : 0;*/
}
cascade = (CvHaarClassifierCascade*)cvLoad( cascade_name, 0, 0, 0 );
if( !cascade )
{
    fprintf( stderr, "ERROR: Could not load classifier cascade\n" );
    return -1;
}
storage = cvCreateMemStorage(0);
if( !input_name || (isdigit(input_name[0]) && input_name[1] == '\0') )
    capture = cvCaptureFromCAM( !input_name ? 0 : input_name[0] - '0' );
cvNamedWindow( "F1S", 1 );
if( capture )
{
    for(;;)
    {
        if( !cvGrabFrame( capture ) )
            break;
        frame = cvRetrieveFrame( capture );
        if( !frame )
            break;
        if( !frame_copy )
            frame_copy = cvCreateImage( cvSize(frame->width,frame->height),

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IPL_DEPTH_8U, frame->nChannels );
    if( frame->origin == IPL_ORIGIN_TL )
        cvCopy( frame, frame_copy, 0 );
    else
        cvFlip( frame, frame_copy, 0 );
        detect_and_draw( frame_copy );
    if( cvWaitKey( 10 ) >= 0 )
        break;
}
cvReleaseImage( &frame_copy );
cvReleaseCapture( &capture );
}
else
{
    const char* filename = input_name ? input_name : (char*)"lena.jpg";
    IplImage* image = cvLoadImage( filename, 1 );
    if( image )
    {
        detect_and_draw( image );
        cvWaitKey(0);
        cvReleaseImage( &image );
    }
}
else
{
    /* assume it is a text file containing the
       list of the image filenames to be processed - one per line */
    FILE* f = fopen( filename, "rt" );
    if( f )
    {
        char buf[1000+1];
        while( fgets( buf, 1000, f ) )
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int len = (int)strlen(buf);
while( len > 0 && isspace(buf[len-1]) )
    len--;
buf[len] = '\0';
image = cvLoadImage( buf, 1 );
if( image )
{
    detect_and_draw( image );
    cvWaitKey(0);
    cvReleaseImage( &image );
}
}
fclose(f);
}
}
cvDestroyWindow("FTS");
return 0;
}

void detect_and_draw( IplImage* img )
{
    int scale = 1;
    IplImage* temp = cvCreateImage( cvSize(img->width/scale,img->height/scale), 8, 3 );
    CvPoint pt1, pt2, pt, cen;
    int i, subx, suby, xout, yout, zout ,z1 ,z2;
    char out[4];
    cvClearMemStorage( storage );
    if( cascade )
    {
        CvSeq* faces = cvHaarDetectObjects( img, cascade, storage,

```

1.1, 2, CV_HAAR_DO_CANNY_PRUNING,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cvSize(40, 40) );
for( i = 0; i < (faces ? faces->total : 0); i++)
{
    CvRect* r = (CvRect*)cvGetSeqElem( faces, i );
    pt1.x = r->x*scale;
    pt2.x = (r->x+r->width)*scale;
    pt1.y = r->y*scale;
    pt2.y = (r->y+r->height)*scale;

    pt.x=(pt1.x+pt2.x)/2;
    pt.y=(pt1.y+pt2.y)/2;
    cen.x=180;
    cen.y=140;
    subx = cen.x-pt.x;
    suby = cen.y-pt.y;
    if (subx>40) xout=1;
    if (subx<-40) xout=2;
    if (subx>-40 && subx<40) xout=0;
    if (suby>40) yout=1;
    if (suby<-40) yout=2;
    if (suby>-40 && suby<40) yout=0;
    z1=pt1.x-pt2.x;
    if (z1<0)
        z1=-z1;
    if (z1<90) zout=1;
    if (z1>110) zout=2;
    if (z1>90 && z1<110) zout=0;
    printf("zout = %d", zout);

    // F=forward,B=backward,U=up,D=down,L=left,R=right

    if (xout==0)
    {
        if (yout==0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (zout==0) {out[0]='a';out[1]='0';out[2]='1';out[3]='b';}; //no moving
        if (zout==1) {out[0]='a';out[1]='0';out[2]='2';out[3]='b';}; //F
        if (zout==2) {out[0]='a';out[1]='0';out[2]='3';out[3]='b';}; //B
    }
if (yout==1)
    {
        if (zout==0) {out[0]='a';out[1]='0';out[2]='4';out[3]='b';}; //U
        if (zout==1) {out[0]='a';out[1]='0';out[2]='5';out[3]='b';}; //U,F
        if (zout==2) {out[0]='a';out[1]='0';out[2]='6';out[3]='b';}; //U,B
    }
if (yout==2)
    {
        if (zout==0) {out[0]='a';out[1]='0';out[2]='7';out[3]='b';}; //D
        if (zout==1) {out[0]='a';out[1]='0';out[2]='8';out[3]='b';}; //D,F
        if (zout==2) {out[0]='a';out[1]='0';out[2]='9';out[3]='b';}; //D,B
    }
}
if (xout==1)
    {
        if (yout==0)
            {
                if (zout==0) {out[0]='a';out[1]='1';out[2]='0';out[3]='b';}; //L
                if (zout==1) {out[0]='a';out[1]='1';out[2]='1';out[3]='b';}; //L,F
                if (zout==2) {out[0]='a';out[1]='1';out[2]='2';out[3]='b';}; //L,B
            }
        if (yout==1)
            {
                if (zout==0) {out[0]='a';out[1]='1';out[2]='3';out[3]='b';}; //L,U
                if (zout==1) {out[0]='a';out[1]='1';out[2]='4';out[3]='b';}; //L,U,F
                if (zout==2) {out[0]='a';out[1]='1';out[2]='5';out[3]='b';}; //L,U,B
            }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (yout==2)
    {
        if (zout==0) {out[0]='a';out[1]='1';out[2]='6';out[3]='b';}; //L,D
        if (zout==1) {out[0]='a';out[1]='1';out[2]='7';out[3]='b';}; //L,D,F
        if (zout==2) {out[0]='a';out[1]='1';out[2]='8';out[3]='b';}; //L,D,B
    }
}
if (xout==2)
{
    if (yout==0)
    {
        if (zout==0) {out[0]='a';out[1]='1';out[2]='9';out[3]='b';}; //R
        if (zout==1) {out[0]='a';out[1]='2';out[2]='0';out[3]='b';}; //R,F
        if (zout==2) {out[0]='a';out[1]='2';out[2]='1';out[3]='b';}; //R,B
    }
    if (yout==1)
    {
        if (zout==0) {out[0]='a';out[1]='2';out[2]='2';out[3]='b';}; //R,U
        if (zout==1) {out[0]='a';out[1]='2';out[2]='3';out[3]='b';}; //R,U,F
        if (zout==2) {out[0]='a';out[1]='2';out[2]='4';out[3]='b';}; //R,U,B
    }
    if (yout==2)
    {
        if (zout==0) {out[0]='a';out[1]='2';out[2]='5';out[3]='b';}; //R,D
        if (zout==1) {out[0]='a';out[1]='2';out[2]='6';out[3]='b';}; //R,D,F
        if (zout==2) {out[0]='a';out[1]='2';out[2]='7';out[3]='b';}; //R,D,B
    }
}

printf("pt.x = %d pt.y = %d cen.x = %d cen.y = %d \n",pt.x,pt.y,cen.x,cen.y);
cvRectangle( img, pt1, pt2, CV_RGB(255,0,0), 3, 8, 0 );
cvRectangle( img, pt, pt, CV_RGB(255,0,0), 3, 8, 0 );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cvRectangle( img, cen, cen, CV_RGB(255,0,0), 3, 8, 0 );
        printf("out = %s \n",out);
        fSuccess=WriteFile(hCom,&out,4,&dwBytesWritten,NULL);
    }
}

cvShowImage( "FTS", img );
cvReleaseImage( &temp );
}

#ifdef _EiC
main(1,"facedetect.c");
#endif

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปรแกรมควบคุมไมโครคอนโทรลเลอร์

```

#include <16f877.h>
#define TxD    PIN_C6
#define RxD    PIN_C7
#define CLOCK SP 20000000

#fuses hs
#fuses NOLVP, NOWDT
#fuses NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=TxD, rev=RxD)
#bit          cren=0x18.4
int i;
int8 k,pwm[4],pw[3],control,control1,control2;

char data[4],strz;

void main(void)
{
    set_tris_b(0x00);
    set_tris_d(0x00);
    output_b(0x00) ;

    for(k=0; k<=3; k++)
    {
        data[k] = 0;
    }

    while(true)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(true)
{
    cren=0;
    delay_ms(0.01);
    cren=1;
    strz=getch();
    if (strz=='a')
        break;
}

for(k=1; k<=3; k++)
{
    data[0]='a';
    data[k]=getch();
}

if (data[0]=='a')
{
    pw[1]= data[1];
    pw[2]= data[2];
    pwm[1]= pw[1] - 0x30;
    pwm[2]= pw[2] - 0x30;

    if ((pwm[1]==0)&(pwm[2]==1))
    {
        output_b(0x00);
    }

    if ((pwm[1]==0)&(pwm[2]==2))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
    output_b(0x02);
    }

    if ((pwm[1]==0)&(pwm[2]==3))
    {
    output_b(0x01);
    }

    if ((pwm[1]==0)&(pwm[2]==4))
    {
    output_b(0x20);
    }
    if ((pwm[1]==0)&(pwm[2]==5))
    {
    output_b(0x22);
    }
    if ((pwm[1]==0)&(pwm[2]==6))
    {
    output_b(0x21);
    }

    if ((pwm[1]==0)&(pwm[2]==7))
    {
    output_b(0x10);
    }

    if ((pwm[1]==0)&(pwm[2]==8))
    {
    output_b(0x12);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((pwm[1]==0)&(pwm[2]==9))
    {
    output_b(0x11) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==0))
    {
    output_b(0x08) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==1))
    {
    output_b(0x0A) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==2))
    {
    output_b(0x09) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==3))
    {
    output_b(0x28) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==4))
    {
    output_b(0x2A) ;
    }

```

```

if ((pwm[1]==1)&(pwm[2]==5))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output_b(0x29);
}

if ((pwm[1]==1)&(pwm[2]==6))
{
output_b(0x18);
}

if ((pwm[1]==1)&(pwm[2]==7))
{
output_b(0x1A);
}

if ((pwm[1]==1)&(pwm[2]==8))
{
output_b(0x19);
}

if ((pwm[1]==1)&(pwm[2]==9))
{
output_b(0x04);
}

if ((pwm[1]==2)&(pwm[2]==0))
{
output_b(0x06);
}

if ((pwm[1]==2)&(pwm[2]==1))
{
output_b(0x05);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((pwm[1]==2)&(pwm[2]==2))
    {
    output_b(0x24);
    }

if ((pwm[1]==2)&(pwm[2]==3))
    {
    output_b(0x26);
    }

if ((pwm[1]==2)&(pwm[2]==4))
    {
    output_b(0x25);
    }

if ((pwm[1]==2)&(pwm[2]==5))
    {
    output_b(0x14);
    }

if ((pwm[1]==2)&(pwm[2]==6))
    {
    output_b(0x16);
    }

if ((pwm[1]==2)&(pwm[2]==7))
    {
    output_b(0x15);
    }
} //this one is for if loop

```

```

} //this one is for while loop

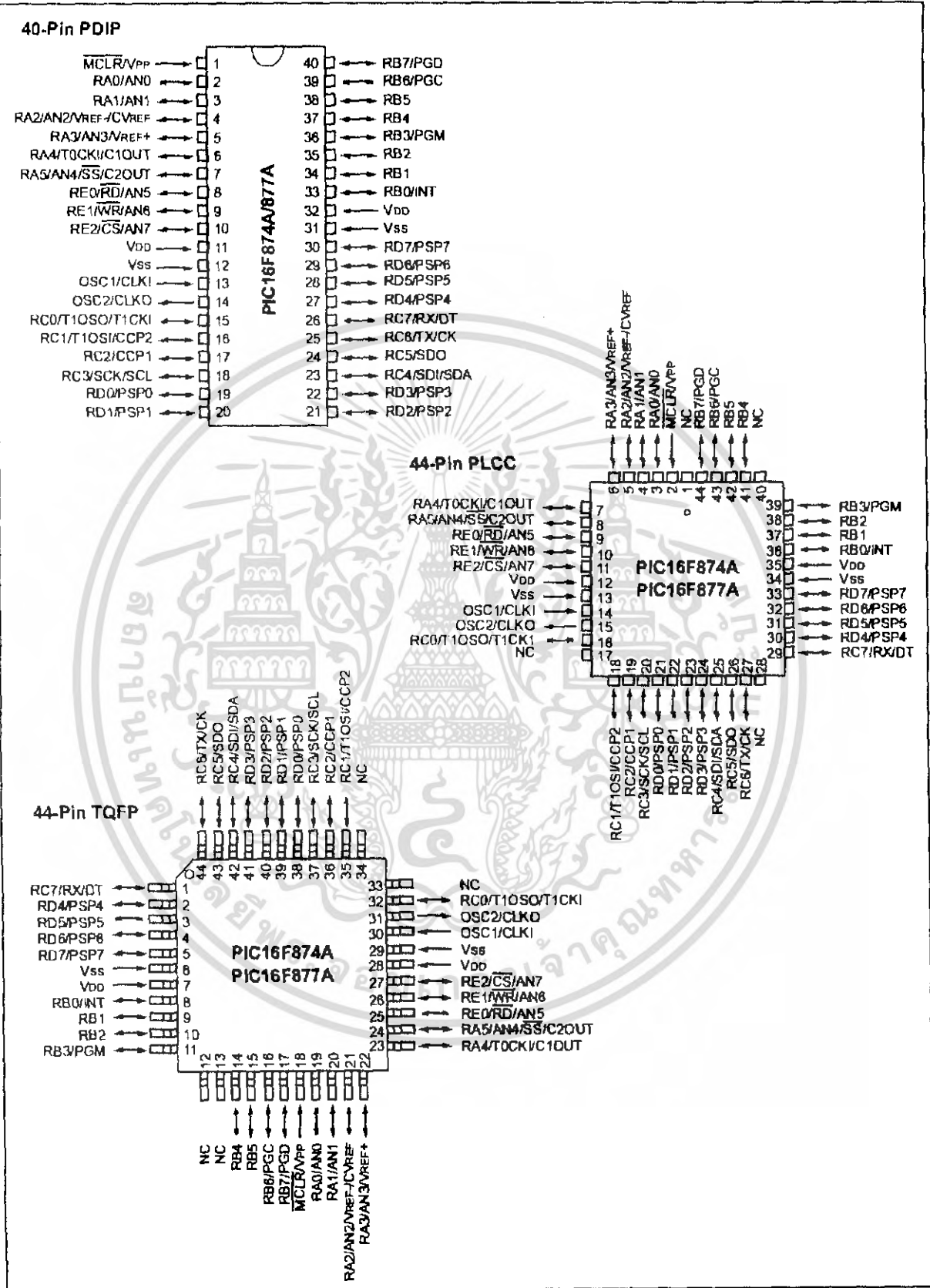
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagrams (Continued)





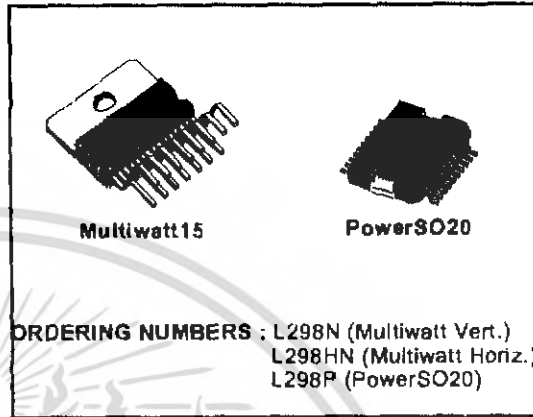
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

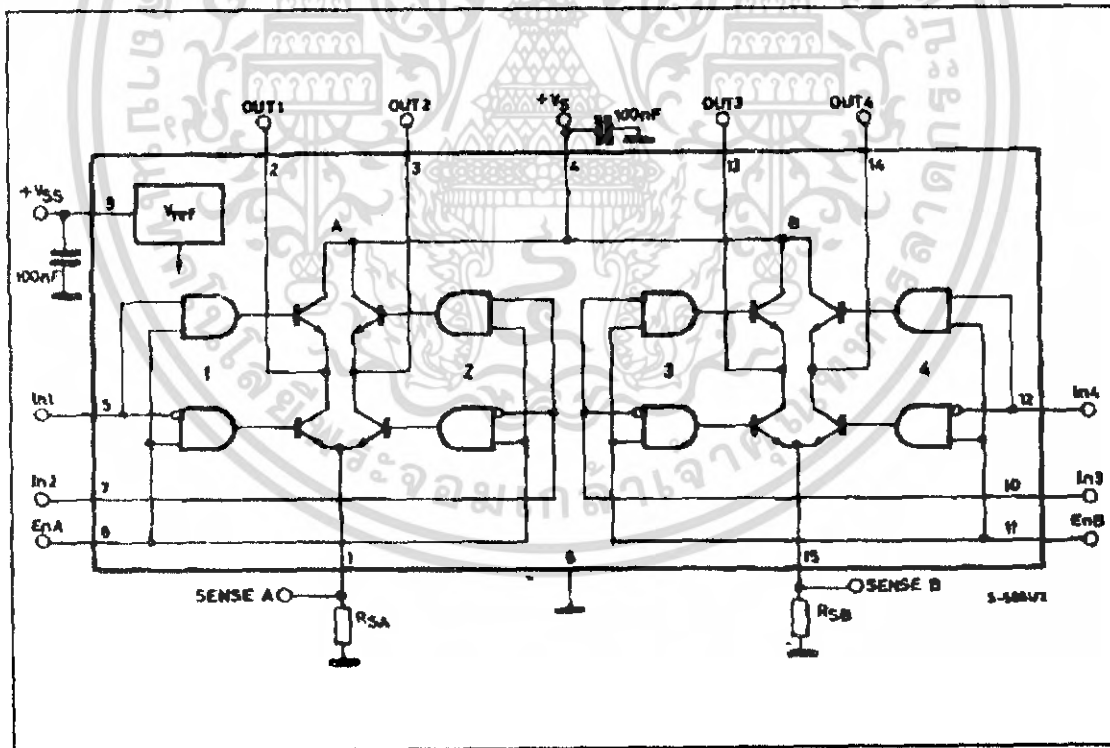
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

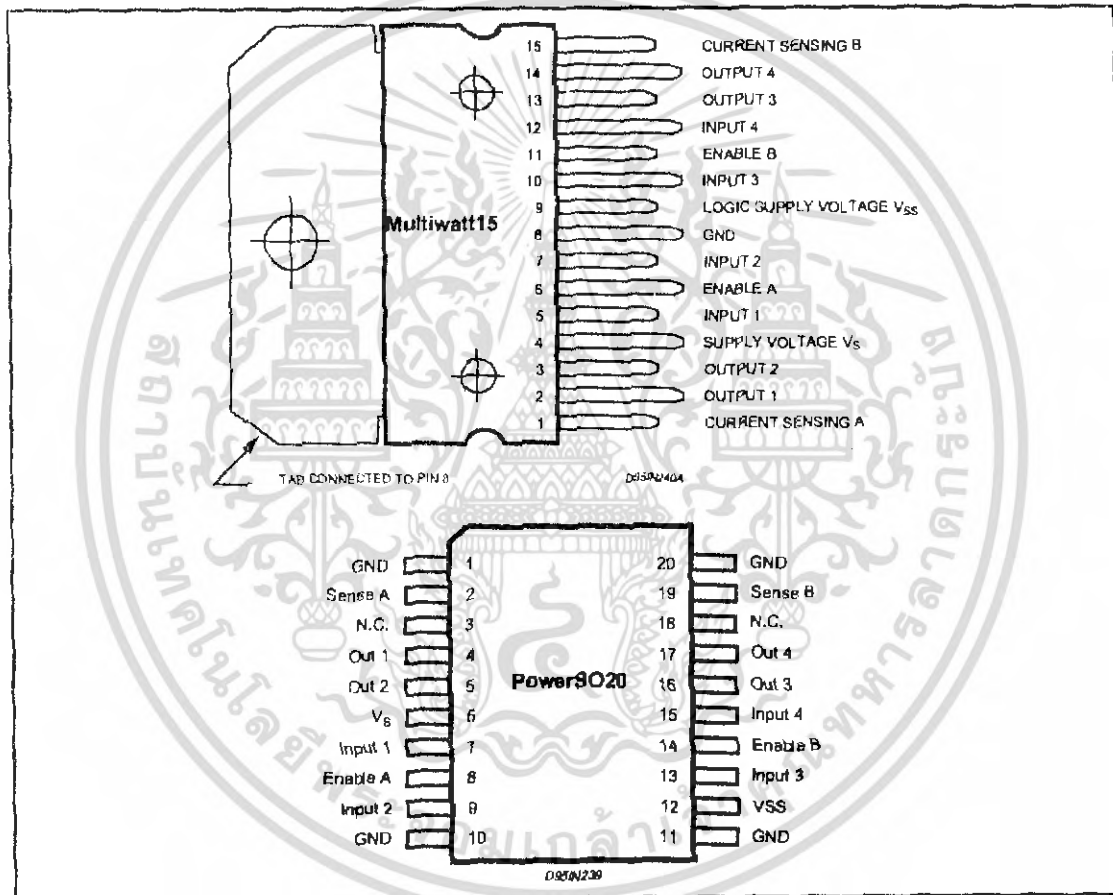


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Power Supply	50	V
V _{SS}	Logic Supply Voltage	7	V
V _I , V _{en}	Input and Enable Voltage	-0.3 to 7	V
I _O	Peak Output Current (each Channel) - Non Repetitive (t = 100µs) - Repetitive (80% on -20% off; I _{on} = 10ms) - DC Operation	3 2.5 2	A A A
V _{sense}	Sensing Voltage	-1 to 2.3	V
P _{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T _{op}	Junction Operating Temperature	-25 to 130	°C
T _{stg} , T _j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
R _{th(j-c)}	Thermal Resistance Junction-case	Max. -	3	°C/W
R _{th(j-amb)}	Thermal Resistance Junction-ambient	Max. 13 (*)	35	°C/W

(*) Mounted on aluminum substrate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS008C - SEPTEMBER 1986 - REVISED NOVEMBER 2004

- Featuring Unijunction L293 and L293D Products Now From Texas Instruments
- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

description/ordering information

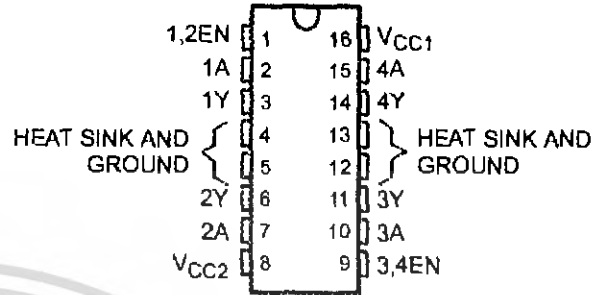
The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

L293 ... N OR NE PACKAGE

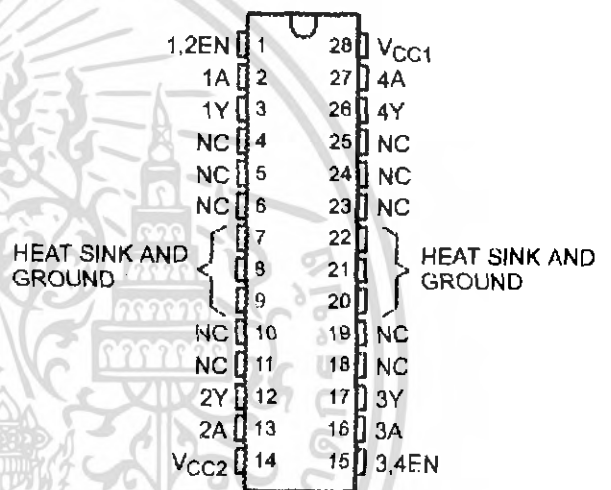
L293D ... NE PACKAGE

(TOP VIEW)



L293 ... DWP PACKAGE

(TOP VIEW)



ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



TEXAS INSTRUMENTS

Copyright © 2004, Texas Instruments Incorporated

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงหรือทำซ้ำโดยไม่ได้รับอนุญาตของเอ็กซาร์ทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

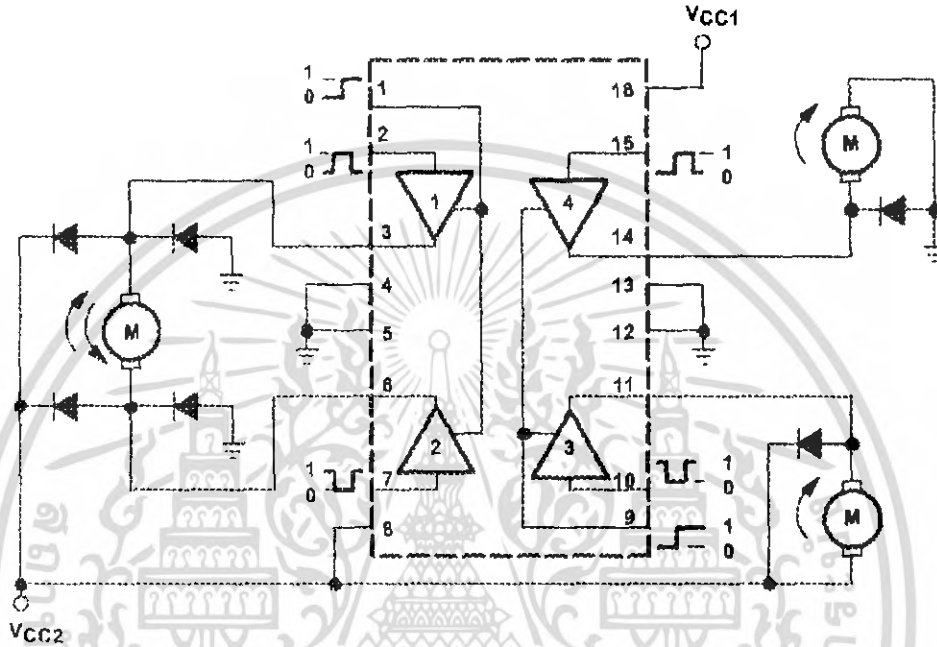
L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS009C - SEPTEMBER 1988 - REVISED NOVEMBER 2004

description/ordering information (continued)

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. A V_{CC1} terminal, separate from V_{CC2} , is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C.

block diagram



NOTE: Output diodes are internal in L293D.

FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant, Z = high impedance (off)

† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.



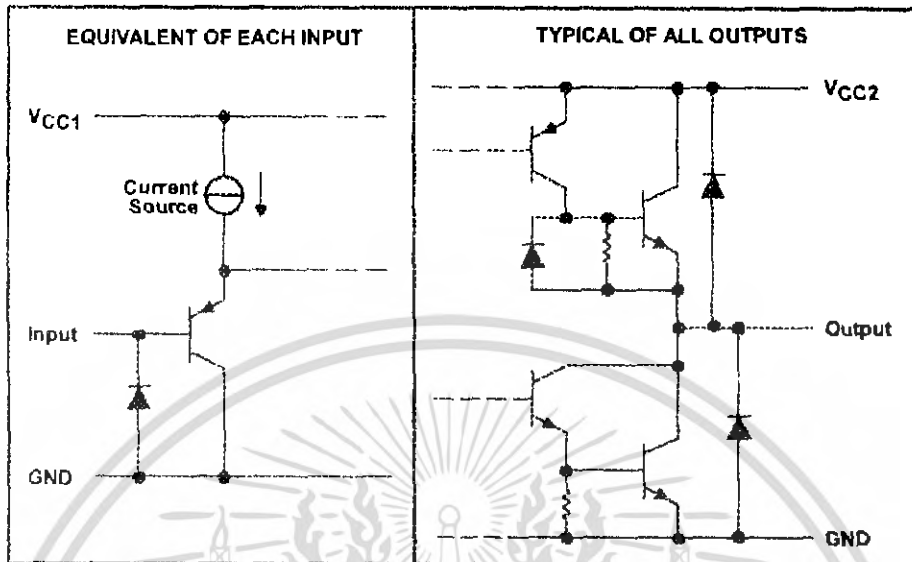
2 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... POST OFFICE BOX 655303 • DALLAS, TEXAS 75265... อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS002C - SEPTEMBER 1988 - REVISED NOVEMBER 2004

schematics of inputs and outputs (L293D)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC1} (see Note 1)	36 V
Output supply voltage, V_{CC2}	36 V
Input voltage, V_I	7 V
Output voltage range, V_O	-3 V to $V_{CC2} + 3$ V
Peak output current, I_O (nonrepetitive, $t \leq 5$ ms): L293	± 2 A
Peak output current, I_O (nonrepetitive, $t \leq 100$ μ s): L293D	± 1.2 A
Continuous output current, I_O : L293	± 1 A
Continuous output current, I_O : L293D	± 600 mA
Package thermal impedance, θ_{JA} (see Notes 2 and 3): DWP package	TBD°C/W
N package	67°C/W
NE package	TBD°C/W
Maximum junction temperature, T_J	150°C
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to the network ground terminal.

2. Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\text{max}) - T_A) / \theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

3. The package thermal impedance is calculated in accordance with JESD 51-7.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ
ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS008C - SEPTEMBER 1986 - REVISED NOVEMBER 2004

recommended operating conditions

		MIN	MAX	UNIT
Supply voltage	VCC1	4.5	7	V
	VCC2	VCC1	36	
V _{IH} High-level input voltage	VCC1 ≤ 7 V	2.3	VCC1	V
	VCC1 ≥ 7 V	2.3	7	V
V _{IL} Low-level output voltage		-0.3†	1.5	V
T _A Operating free-air temperature		0	70	°C

† The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

electrical characteristics, VCC1 = 5 V, VCC2 = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V _{OH}	High-level output voltage	L293: I _{OH} = -1 A L293D: I _{OH} = -0.6 A		VCC2 - 1.8	VCC2 - 1.4		V
V _{OL}	Low-level output voltage	L293: I _{OL} = 1 A L293D: I _{OL} = 0.6 A			1.2	1.8	V
V _{OKH}	High-level output clamp voltage	L293D: I _{OK} = -0.6 A			VCC2 + 1.3		V
V _{OKL}	Low-level output clamp voltage	L293D: I _{OK} = 0.6 A			1.3		V
I _{IH}	High-level input current	A	V _I = 7 V		0.2	100	μA
		EN			0.2	10	
I _{IL}	Low-level input current	A	V _I = 0		-3	-10	μA
		EN			-2	-100	
I _{CC1}	Logic supply current	I _O = 0	All outputs at high level		13	22	mA
			All outputs at low level		35	60	
			All outputs at high impedance		8	24	
I _{CC2}	Output supply current	I _O = 0	All outputs at high level		14	24	mA
			All outputs at low level		2	6	
			All outputs at high impedance		2	4	

switching characteristics, VCC1 = 5 V, VCC2 = 24 V, T_A = 25°C

PARAMETER	TEST CONDITIONS	L293NE, L293DNE			UNIT
		MIN	TYP	MAX	
t _{PLH}	Propagation delay time, low-to-high-level output from A input		800		ns
t _{PHL}	Propagation delay time, high-to-low-level output from A input		400		ns
t _{TLH}	Transition time, low-to-high-level output		300		ns
t _{THL}	Transition time, high-to-low-level output		300		ns

C_L = 30 pF, See Figure 1

switching characteristics, VCC1 = 5 V, VCC2 = 24 V, T_A = 25°C

PARAMETER	TEST CONDITIONS	L293DWP, L293N L293DN			UNIT
		MIN	TYP	MAX	
t _{PLH}	Propagation delay time, low-to-high-level output from A input		750		ns
t _{PHL}	Propagation delay time, high-to-low-level output from A input		200		ns
t _{TLH}	Transition time, low-to-high-level output		100		ns
t _{THL}	Transition time, high-to-low-level output		350		ns

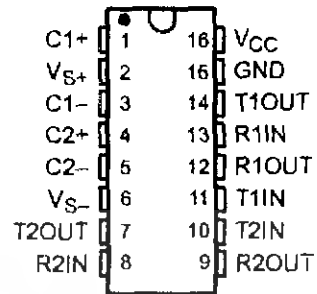
C_L = 30 pF, See Figure 1

MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

- Operates With Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Designed to be Interchangeable With Maxim MAX232
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers
- ESD Protection Exceeds 2000 V Per MIL-STD-883, Method 3015
- Package Options Include Plastic Small-Outline (D, DW) Packages and Standard Plastic (N) DIPs

D, DW, OR N PACKAGE
(TOP VIEW)

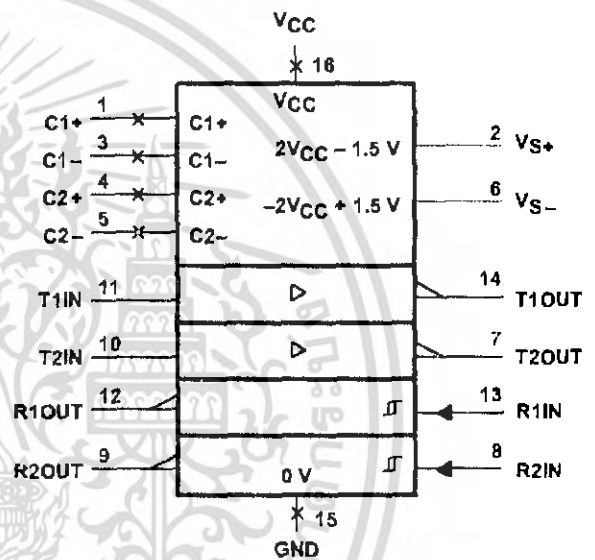


description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

The MAX232 is characterized for operation from 0°C to 70°C. The MAX232I is characterized for operation from -40°C to 85°C.

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

AVAILABLE OPTIONS

T _A	PACKAGED DEVICES		
	SMALL OUTLINE (D)	SMALL OUTLINE (DW)	PLASTIC DIP (N)
0°C to 70°C	MAX232D‡	MAX232DW‡	MAX232N
-40°C to 85°C	MAX232ID‡	MAX232IDW‡	MAX232IN

‡ This device is available taped and reeled by adding an R to the part number (i.e., MAX232DR).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.

Copyright © 1998, Texas Instruments Incorporated

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655393 • DALLAS, TEXAS 75265

MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	-0.3 V to -15 V
Input voltage range, V_I : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Note 2): D package	113°C/W
DW package	105°C/W
N package	78°C/W
Storage temperature range, T_{stg}	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.

2: The package thermal impedance is calculated in accordance with JESD 51, except for through-hole packages, which use a trace length of zero.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	V
High-level input voltage, V_{IH} (T1IN, T2IN)	2			V
Low-level input voltage, V_{IL} (T1IN, T2IN)			0.8	V
Receiver input voltage, R1IN, R2IN			± 30	V
Operating free-air temperature, T_A	MAX232	0	70	°C
	MAX232I	-40	85	



MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

SLLS047G – FEBRUARY 1989 – REVISED AUGUST 1998

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	T1OUT, T2OUT	R _L = 3 kΩ to GND	5	7		V
		R1OUT, R2OUT	I _{OH} = -1 mA	3.5			
V _{OL}	Low-level output voltage‡	T1OUT, T2OUT	R _L = 3 kΩ to GND		-7	-5	V
		R1OUT, R2OUT	I _{OL} = 3.2 mA			0.4	
V _{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V _{CC} = 5 V, T _A = 25°C		1.7	2.4	V
V _{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V _{CC} = 5 V, T _A = 25°C	0.8	1.2		V
V _{Hys}	Input hysteresis voltage	R1IN, R2IN	V _{CC} = 5 V	0.2	0.5	1	V
r _i	Receiver input resistance	R1IN, R2IN	V _{CC} = 5, T _A = 25°C	3	5	7	kΩ
r _o	Output resistance	T1OUT, T2OUT	V _{S+} = V _{S-} = 0, V _O = ±2 V	300			Ω
I _{OS} §	Short-circuit output current	T1OUT, T2OUT	V _{CC} = 5.5 V, V _O = 0		±10		mA
I _{IS}	Short-circuit input current	T1IN, T2IN	V _I = 0			200	μA
I _{CC}	Supply current		V _{CC} = 5.5 V, T _A = 25°C, All outputs open,		8	10	mA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH(R)}	Receiver propagation delay time, low- to high-level output	See Figure 1		500		ns
t _{PHL(R)}	Receiver propagation delay time, high- to low-level output	See Figure 1		500		ns
SR	Driver slew rate	R _L = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(tr)	Driver transition region slew rate	See Figure 3		3		V/μs



TEXAS
INSTRUMENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในสหรัฐอเมริกาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องข้อมูลเชิงลึกของเอกสารทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265