สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การติดตามคนที่กำลังเคลื่อนที่ในภาพ

TRACKING OF MOVING PERSONS IN SCENE

นายสิระ รามัญจิต
นายอภิวัฒน์ อำพันธ์ทองศิริ
นางสาวอมรรัตน์ โย้ยนิ่ง
นางสาวภัทรวดี ภักดีชนาภัทร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 119568
วัน,เดือน,ปี - 8 S.A. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRACKING OF MOVING PERSONS IN SCENE

SIRA RAMONCHIT

APHIWAT AMPHANTHONGSIRI

AMORNRAT KHOINUNG

PATTARAWADEE PAKDEECHANAPAT

A THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG

ACADEMIIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	การติดตามคนที่กำลังเคลื่อนที่ในภาพ			
	TRACKING OF MOVING PERSONS IN SCENE			
นักศึกษาผู้จัดทำ	นายสิระ	รามัญจิต	**รหัสนักศึกษา**	50011222
	นายอภิวัฒน์	อำพันธ์ทองศิริ	**รหัสนักศึกษา**	50011869
	นางสาวอมรรัตน์	ไข้ยนึ่ง	**รหัสนักศึกษา**	50011877
	นางสาวภัทรวดี	ภักดีชนาภัทร์	**รหัสนักศึกษา**	50011963
ปริญญา	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมการวัดคุม			
ปีการศึกษา	2553			

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ดร. เกษตร ศิริสันติสัมฤทธิ์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การติดตามคนที่กำลังเคลื่อนที่ในภาพ		
	TRACKING OF MOVING PERSONS IN SCENE		
นักศึกษาผู้จัดทำ	นายสิระ	รามัญจิต	**รหัสนักศึกษา** 50011222
	นายอภิวัฒน์	อำพันทองศิริ	**รหัสนักศึกษา** 50011222
	นางสาวอมรรัตน์	โข้ยนึ่ง	**รหัสนักศึกษา** 50011222
	นางสาวภัทรวดี	ภักดีชนาภัทร์	**รหัสนักศึกษา** 50011963
อาจารย์ที่ปรึกษา	รศ.ดร. เกษตร์	ศิริสันติสัมฤทธิ์	
ปีการศึกษา	2553		

บทคัดย่อ

โครงงานนี้นำเสนอการศึกษาวิธีการติดตามบุคคลต่อจากพรีโปรเจคที่ได้การศึกษาวิจัยและปรับปรุงวิธีการติดตามวัตถุ ซึ่งจากพรีโปรเจคครั้งที่แล้ววิธีการติดตามวัตถุเป็นการถ่ายภาพเข้ามาอย่างต่อเนื่องโดยใช้กล้องเวปแคม จากนั้นนำภาพถ่ายที่ต่อเนื่องกันมาทำการลบกันเพื่อหาความแตกต่างของภาพทั้งสอง ความแตกต่างของภาพที่เกิดขึ้นจะถูกพิจารณาว่าเป็นวัตถุที่กำลังเคลื่อนที่ซึ่งจะถูกตีกรอบล้อมรอบวัตถุนั้นแล้วหาจุดศูนย์กลางของวัตถุ ในกรณีที่จุดศูนย์กลางของวัตถุอยู่ห่างจากจุดศูนย์กลางของภาพตามค่าที่ได้กำหนดไว้ คอมพิวเตอร์จะสั่งให้เซอร์โวมอเตอร์ที่ผูกติดเข้ากับกล้อง หมุนกล้องไปทางด้านซ้ายมือหรือขวามือเพื่อให้จุดศูนย์กลางของวัตถุกลับมาอยู่ใกล้กับจุดศูนย์กลางของภาพ ซึ่งในพรีโปรเจคนั้นโปรแกรมสามารถตีกรอบล้อมรอบวัตถุได้เพียงวัตถุเดียวเท่านั้น และไม่สามารถตีกรอบล้อมรอบวัตถุที่สีใกล้เคียงกับพื้นหลังได้อย่างถูกต้อง ดังนั้นในโครงงาน I จึงได้ทำการทดลองเขียนโปรแกรมเพื่อช่วยในการประมวลผลภาพ ศึกษาโปรแกรมตีกรอบวัตถุหลายวัตถุได้พร้อมทั้งเลือกตีกรอบเฉพาะบุคคลที่อยู่ในภาพ และสามารถตีกรอบล้อมรอบบุคคลที่มีสีใกล้เคียงกับพื้นหลังได้อย่างถูกต้อง โดยได้ทดลองเขียนโปรแกรมหลาย ๆ วิธีเพื่อหาวิธีที่เหมาะสมที่สุด ในส่วนของเซอร์โวมอเตอร์ที่ใช้ในการควบคุมกล้องนั้นได้มีทำการทดลองเขียนโปรแกรมลงบนหน่วยความจำของไมโครคอนโทรลเลอร์เพื่อติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Tracking of moving persons in scene	
Authors	Mr.Sira	Ramonchit
	Mr.Aphiwat	Amphanthongsiri
	Miss Amornrat	Khoinung
Thesis Advisor	Assoc.Prof.. Kaset	Sirisantisamrit
Year	2010	

ABSTRACK

This paper presents continuous study from pre-project (object tracking method) and improvement of it. The object tracking method is to capture pictures continuously by webcam. Then the continuous two frames of pictures are differed to find a group of different pixels of both pictures as which they are considered a moving object (person). Then a border is made around it and determines the centroid of it. In case of the centroid of moving object (person) is far from the center of image as fixed value, the computer will send commands to a servo motor attached with the camera to move a tied camera servo motor turn left or right to arrange the centroid of picture to be near the position of fixed point. The problems of pre-project are program can make only one border and cannot make it correctly when color shades between background and moving object (person) is alike, so we try writing several program to find the best program can make many borders around moving objects (person) correctly. In case of servo motor, we study communication between a microcontroller that controls the servo motor and a computer and try burning program in EEPROM of microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับคำปรึกษาและแนะนำจากอาจารย์ รศ.ดร. เกษตร ศิริสันติสัมฤทธิ์ อาจารย์ที่ปรึกษาและควบคุมปริญญานิพนธ์ตลอดจนการทำปริญญานิพนธ์นี้รวมถึงให้การช่วยเหลือทั้งปัญหาการศึกษาและปัญหาส่วนตัว กลุ่มนักศึกษาทำโครงงานปริญญานิพนธ์รู้สึกซาบซึ้ง จึงใคร่ขอกราบขอบพระคุณอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณอาจารย์ รศ.ดร. เกษตร ศิริสันติสัมฤทธิ์ ที่ให้คำแนะนำในการทำปริญญานิพนธ์ คณาจารย์ภาควิชาวิศวกรรมการวัดคุมและควบคุม สาขาวิชาวิศวกรรมการวัดคุมทุกท่าน และเพื่อน ๆ ทุกคนที่ได้ให้คำแนะนำ และให้การช่วยเหลืออันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้ ขอกราบขอบพระคุณคุณพ่อคุณแม่อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้ คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ กลุ่มนักศึกษาของโครงงานขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... **I**

บทคัดย่อภาษาอังกฤษ ..... **II**

กิตติกรรมประกาศ ..... **III**

สารบัญ ..... **IV**

สารบัญตาราง ..... **VI**

สารบัญภาพ ..... **VIII**

บทที่ 1 บทนำ **1**

1.1 ความสำคัญของปริญญานิพนธ์ 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 2

1.3 ขอบเขตของปริญญานิพนธ์ 2

1.4 ขั้นตอนการศึกษา 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎี **4**

2.1 การมองเห็นของมนุษย์และคอมพิวเตอร์ 4

2.2 การแปลงภาพให้เป็นภาพเชิงดิจิตอล 4

2.3 การประมวลผลภาพดิจิตอลด้วยโปรแกรม 5

2.4 ชนิดของภาพ (Image Type) 6

2.4.1 ภาพความเข้มแสง (Intensity Image) 6

2.4.2 รูปแบบสี (color model) 7

2.5 การแบ่งส่วนภาพ (Image Segmentation) 10

2.6 การกำหนดสัญญาณรบกวนด้วย Median Filter 11

2.7 ทฤษฎีเบื้องต้นทางสถิติ 12

2.7.1 ค่าเฉลี่ยเลขคณิต (Arithmetic Mean) 12

2.7.2 ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation: s.d.) 12

2.8 การ Label วัตถุ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.9 การติดต่อของโปรแกรม MATLAB กับกล้องเว็บแคม 15

บทที่ 3 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการวิจัย 17

3.1 ภาษาซี (C Programming Language).....17

3.1.1 PIC C Compiler 17

3.1.2 โปรแกรม PICkit2 V2.61.....18

3.2 อุปกรณ์และฮาร์ดแวร์.....18

3.2.1 คอมพิวเตอร์ Notebook Lenovo รุ่น G450.....18

3.2.2 ET-PGMPIC USB V 2.0 19

3.2.3 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC18F4620 20

3.2.4 เซอร์โวมอเตอร์ (Servo motor) 21

3.2.5 บอร์ด ET-BASE PIC40 24

3.2.6 ET-USB/RS232 MINI 25

3.2.7 กล้องเว็บแคม(Webcam).....26

3.2.8 ET-SWITCHING ADAPTER 27

3.2.9 หลักการออกแบบชุดขาตั้งกล้อง.....27

3.3 การติดต่อสื่อสารทางพอร์ตอนุกรม (RS-232).....28

3.4 การควบคุมเซอร์โวมอเตอร์โดยผ่านโปรแกรม MATLAB 28

3.4.1 ชนิดข้อมูล.....31

3.4.2 ฟังก์ชันที่ใช้ในการเขียน Source Code บนโปรแกรม PIC C Compiler 31

3.4.3 หลักการทำงานของโปรแกรมควบคุมเซอร์โวมอเตอร์ 32

3.4.4 Source Code ควบคุมเซอร์โวมอเตอร์ 33

3.4.5 ขั้นตอนการเบิร์น hex. file ลงไมโครคอนโทรลเลอร์ 37

3.4.6 โปรแกรม Hyper terminal ใช้ติดต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ 39

3.4.7 โปรแกรมที่ใช้ในการควบคุมเซอร์โวมอเตอร์ผ่านโปรแกรม MATLAB.....40

บทที่ 4 การประมวลผลภาพเพื่อการติดตามคนที่กำลังเคลื่อนที่ในภาพ..... 42

4.1 การเขียนโปรแกรมตีกรอบล้อมรอบวัตถุโดยการใช้ bwlabel.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การตรวจหาสีผิว (Skin Detection).....44
4.2.1 การตัดค่า Threshold แบบอัตโนมัติ.....46
4.2.2 การตัดค่า Threshold จากการทำ Gaussian.....47
4.2.3 การตัดค่า Threshold จากค่า r g และ b ที่ได้จากการ Normalize ภาพสี RGB.....55
4.2.4 การเลือกตัวอย่างสีผิวสำหรับการตัดค่า Threshold ที่เหมาะสม.....63
4.3 การตรวจหาสีของปลอกแขน (Armband's color Detection).....64
4.4 การหาเส้นทางการเคลื่อนที่ของคนที่อยู่ในภาพ.....74
4.5 การหาค่า Hysteresis ของเซอร์โวมอเตอร์.....76
4.6 ขั้นตอนการประมวลผลภาพ.....77
บทที่ 5 ผลการทดลอง.....87
5.1 ผลการทดลองทางด้านฮาร์ดแวร์.....87
5.1.1 การทดลองการติดต่อสื่อสารและการบวกลบจำนวนทางคณิตศาสตร์.....87
5.1.2 การทดลองควบคุมเซอร์โวมอเตอร์ 4 ตัว.....91
5.1.3 การทดลองเพื่อหาคุณลักษณะของเซอร์โวมอเตอร์.....95
5.2 ผลการทดลองทางด้านซอฟต์แวร์ (Software Experiments).....112
5.2.1 การหาสีผิว (Skin Detection).....112
5.2.2 การเลือกตัวอย่างสีผิว.....125
5.3 ทดลองติดตามคนที่เคลื่อนที่อยู่ในภาพ.....127
บทที่ 6 สรุปผลและข้อเสนอแนะ.....128
6.1 สรุปผลการทำโครงงาน.....128
6.2 ข้อเสนอแนะ.....128

บรรณานุกรม.....129

ภาคผนวก.....130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงค่าความน่าเชื่อถือของแต่ละซิกมา 14

3.1 ชนิดข้อมูลภาษา C สำหรับ CCS C คอมไพเลอร์ 31

4.1 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ
เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ r 55

4.2 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ
เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ g 55

4.3 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ
เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ b 56

5.1 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านซ้าย
โดยใช้ความกว้างพัลส์ 1500 ~ 1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1 80

5.2 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านขวา
โดยใช้ความกว้างพัลส์ 1500 ~ 1650 µs โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1 81

5.3 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน X (Pulse/Pixel) ชุดที่ 1 81

5.4 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านบน
โดยใช้ความกว้างพัลส์ 1500 ~ 1650 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1 82

5.5 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านล่าง
โดยใช้ความกว้างพัลส์ 1500 ~ 1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1 83

5.6 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน Y (Pulse/Pixel) ชุดที่ 1 83

5.7 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านซ้าย
โดยใช้ความกว้างพัลส์ 1500 ~ 1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2 84

5.8 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านขวา
โดยใช้ความกว้างพัลส์ 1500 ~ 1650 µs โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2 84

5.9 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน X (Pulse/Pixel) ชุดที่ 2 85

5.10 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านบน
โดยใช้ความกว้างพัลส์ 1500 ~ 1650 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2 85

5.11 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านล่าง
โดยใช้ความกว้างพัลส์ 1500 ~ 1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2 86

5.12 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน Y (Pulse/Pixel) ชุดที่ 2 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

5.13 แสดงค่า Threshold ของ r g และ b ในแต่ละกรณีของการใช้ตัวอย่างสีผิว 87

5.14 สรุปโมเดลที่ใช้ในการตรวจหาสีผิวและกรณีเลือกใช้ตัวอย่างสีผิว 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 การแทนภาพด้วยเมตริกซ์ 5
2.2 หน้าต่างการทำงานและสภาพแวดล้อมต่าง ๆของโปรแกรม MATLAB 6
2.3 ภาพความเข้มของแสง 6
2.4 ภาพสี 7
2.5 แสดง RGB Color Model 7
2.6 แบบจำลองกรวยภาพสีของ HSV 9
2.7 ภาพขาวดำ 10
2.8 การตั้งค่าเทรสโชว์เพื่อแบ่งข้อมูลในภาพ 11
2.9 การนำค่ามัธยฐานที่ได้นั้นแทนค่ากลับลงไปในพิกเซลตำแหน่งตรงกลางของหน้าต่าง 12
2.10 การกระจายเป็นรูประฆังคว่ำ 13
2.11 ภาพที่มีวัตถุ 3 วัตถุ 14
2.12 ภาพผลลัพธ์ภายหลังจากคำสั่ง bwlabel 15
3.1 แสดงหน้าต่างโปรแกรม PIC C Compiler 18
3.2 หน้าต่างโปรแกรม PICKit2 V2.61 18
3.3 คอมพิวเตอร์ Notebook Lenovo รุ่น G450 19
3.4 ET-PGM PIC USB V2.0 19
3.5 โครงสร้างภายนอกของ PIC 18F4620 21
3.6 แสดงโครงสร้าง CPU PIC 18F4620 และการจัดวางขาสัญญาณ 21
3.7 ส่วนประกอบของเซอร์โวมอเตอร์ 22
3.8 แสดงทิศทางการหมุนของเซอร์โวมอเตอร์เมื่อจ่ายพัลส์ขนาดต่างๆ 23
3.9 โครงสร้างบอร์ด ET-BASE PIC40 24
3.10 ET-USB/RS232 MINI 26
3.11 กล้องเว็บแคม 26
3.12 ET-SWITCHING ADAPTER 27
3.13 ชุดขาตั้งกล้อง 28
3.14 การเชื่อมต่อพอร์ตอนุกรมเข้ากับบอร์ดและคอมพิวเตอร์ 29
3.15 พอร์ตอนุกรม RS-232 29
3.16 การทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
3.17 หน้าต่างโปรแกรม PICkit 2 V2.61	37
3.18 โปรแกรม PICkit2 V2.61 ตรวจพบ PIC 18F4620	37
3.19 หน้าต่างโปรแกรม PICkit2 V2.61	38
3.20 หน้าต่างโปรแกรม Hyper terminal	39
3.21 หน้าต่างโปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์	40
4.1 การตีกรอบล้อมรอบวัตถุโดยหาจุด Co-ordinate ของแต่ละ Row และ Column	42
4.2 ตัวอย่างการตีกรอบล้อมรอบเมล็ดข้าว	42
4.3 ตัวอย่างการตีกรอบล้อมรอบคนในภาพ	43
4.4 การตัดตัวอย่างสีผิวแล้วทำการแยกภาพของแสงสีแดง สีเขียวและสีน้ำเงิน	43
4.5 การตัดค่า Threshold โดยปรับค่าลดลงทีละ 0.1	45
4.6 ตัวอย่างการตัดสีผิวของภาพคนที่ถ่ายทั้งภายในและภายนอกห้อง แล้วทำการแยกภาพของแสงสีแดง สีเขียวและสีน้ำเงิน	46
4.7 การเลือกค่า Gaussian เป็นค่า Threshold ต่ำสุดและสูงสุด จากการฉายกลับค่า Gaussian ไปที่แนวแกน R และ G	48
4.8 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RG	49
4.9 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RG	49
4.10 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RG	50
4.11 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RB	50
4.12 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RB	51
4.13 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RB	51
4.14 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล GB	51
4.15 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล GB	52
4.16 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล GB	52
4.17 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RGB	52
4.18 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RGB	53
4.19 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RGB	53
4.21 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rg	56
4.22 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rg	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

4.23 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rg 57
4.24 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rb 58
4.25 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rb 58
4.26 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rb 58
4.27 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล gb 59
4.28 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล gb 59
4.29 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล gb 60
4.30 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rgb 60
4.31 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rgb 61
4.32 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rgb 61
4.33 การตัดตัวอย่างสีผิว 62
4.34 กราฟแสดงการเกิดฮีสเตอรีซีส 63
5.1 เขียน Source Code ลงในโปรแกรม PIC Compiler 67
5.2 ผลการทดลองบวกเลขผ่านโปรแกรม Terminal 68
5.3 เซอร์โวมอเตอร์ให้หมุนไปที่ตำแหน่งกึ่งกลางที่ตำแหน่งกึ่งกลาง 69
5.4 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปที่ตำแหน่งตรงกลางทั้ง 4 ตัว 69
5.5 เซอร์โวมอเตอร์ถูกควบคุมให้หมุนไปทางด้านซ้ายมือสุดทั้ง 4 ตัว 70
5.6 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปทางด้านซ้ายมือสุดทั้ง 4 ตัว 70
5.7 เซอร์โวมอเตอร์ถูกควบคุมให้หมุนไปทางด้านขวามือสุดทั้ง 4 ตัว 71
5.8 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปทางด้านขวามือสุดทั้ง 4 ตัว 71
5.9 การเขียนโปรแกรมประมวลผลโดยใช้ MATLAB 78
5.10 การตีกรอบล้อมรอบวัตถุเพื่อหาจุดศูนย์กลางของวัตถุ 78
5.11 กล้องหมุนในแนวแกน X ทางด้านซ้าย 79
5.12 กล้องหมุนในแนวแกน X ทางด้านขวา 80
5.13 กล้องหมุนในแนวแกน Y ทางด้านบน 82
5.14 กล้องหมุนในแนวแกน Y ทางด้านล่าง 82
5.13 กล้องหมุนในแนวแกน Y ทางด้านบน 82
5.14 กล้องหมุนในแนวแกน Y ทางด้านล่าง 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

5.15 ภาพต้นฉบับ 88

5.16 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rg 88

5.17 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rg 89

5.18 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rg 89

5.19 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rg 90

5.20 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rg 90

5.21 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rg 90

5.22 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rb 91

5.23 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rb 91

5.24 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rb 92

5.25 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rb 92

5.26 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rb 92

5.27 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rb 93

5.28 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล gb 94

5.29 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล gb 94

5.30 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล gb 94

5.31 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล gb 95

5.32 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล gb 95

5.33 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล gb 95

5.34 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rgb 96

5.35 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rgb 97

5.36 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rgb 97

5.37 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rgb 97

5.38 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rgb 98

5.39 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rgb 98

5.40 ตัวอย่างสีผิวในอัตราส่วน 20%:80% 100

5.41 ตัวอย่างสีผิวในอัตราส่วน 50%:50% 101

5.42 ตัวอย่างสีผิวในอัตราส่วน 80%:20% 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ระบบรักษาความปลอดภัยโดยทั่วไปแล้ว มักจะติดตั้งกล้องโทรทัศน์วงจรปิดสำหรับการบันทึกภาพเหตุการณ์ต่าง ๆในแต่ละช่วงเวลา เพื่อตรวจสอบหรือตรวจจับความเคลื่อนไหวของบุคคลในสถานที่ต่าง ๆ แต่ข้อจำกัดของการใช้กล้องโทรทัศน์วงจรปิด (CCTV) ก็คือสามารถตรวจจับการเคลื่อนที่ของบุคคลในมุมที่จำกัด ไม่สามารถตรวจจับบุคคลที่เคลื่อนที่อยู่ภายนอกมุมมองของกล้องได้ ดังนั้นเพื่อให้สามารถตรวจจับบุคคลที่เคลื่อนที่ในมุมมองที่กว้างขึ้น โดยทั่วไปจะต้องติดตั้งกล้องวงจรปิดหลายตัว ทำให้เสียค่าใช้จ่ายเพิ่มมากขึ้น ด้วยเหตุนี้จึงมีแนวคิดที่จะให้กล้องวงจรปิดเพียงตัวเดียวที่สามารถตรวจจับบุคคลที่กำลังเคลื่อนที่ไปที่ตำแหน่งต่างๆได้ โดยใช้อุปกรณ์ประกอบเพื่อหมุนกล้องไปที่ตำแหน่งต่างๆ อุปกรณ์ประกอบที่ใช้ในงานวิจัยนี้ก็คือเซอร์โวมอเตอร์ (Servo Motor)

งานวิจัยนี้เป็นการศึกษาวิธีการติดตามวัตถุ (บุคคล) เป็นการศึกษาต่อจาก pre-project ที่เคยทำการศึกษาวิจัยเอาไว้ แนวความคิดเดิมของการติดตามวัตถุคือการเก็บภาพพื้นหลัง (Background) ไว้ จากนั้นนำภาพในเฟรมถัดมาที่มีวัตถุ (บุคคล) ในภาพมาลบกับภาพพื้นหลังที่ตำแหน่งนั้นๆ เพื่อให้เกิดความแตกต่างและเหลือเฉพาะส่วนที่เป็นวัตถุอยู่ในภาพ จากนั้นตีกรอบล้อมรอบผลต่างของภาพทั้งสองซึ่งคาดว่าเป็นวัตถุ โดยโปรแกรมสามารถตีกรอบล้อมรอบวัตถุได้เพียงวัตถุเดียวเท่านั้น จากนั้นหาจุดศูนย์กลางของวัตถุนั้น และหาระยะทางระหว่างจุดศูนย์กลางของภาพกับจุดศูนย์กลางของวัตถุ ถ้าระยะทางที่หาได้เกินขอบเขตที่ตั้งไว้ คอมพิวเตอร์ก็จะสั่งให้เซอร์โวมอเตอร์หมุนไปทางซ้ายหรือขวามือ เพื่อรักษาให้จุดศูนย์กลางของวัตถุกลับมาอยู่ที่จุดศูนย์กลางของภาพอีกครั้ง จากการศึกษาต่อ pre-project จึงทำการพัฒนาต่อ โดยเพิ่มเซอร์มอเตอร์ 2 ตัว สามารถทำให้กล้องหมุนได้ 2 แกน เพื่อให้มีมุมมองในการตรวจจับบุคคลที่กำลังเคลื่อนไหวในภาพได้เป็นมุมกว้างมากขึ้น และสามารถเลือกติดตามคนได้ด้วยการใช้ปลอกแขนสีแดงมาเป็นตัวกำหนดในการเลือกติดตามบุคคลที่เราสนใจ ทั้งนี้ในงานวิจัยนี้ยังได้แก้ไขปัญหาเรื่องสีเสื้อที่เหมือนกับสีพื้นหลังโดยทำการตรวจจับสีผิวคนทำให้สามารถแยกคนออกจากพื้นหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ติดตามบุคคลที่เคลื่อนที่โดยใช้โปรแกรม MATLAB

2. ออกแบบและปรับปรุงการทำงานของเซอร์โวมอเตอร์ให้สามารถบังคับให้กล้องที่ใช้ในการติดตามบุคคลสามารถเคลื่อนที่ได้ในทิศทาง 4 ทิศทาง คือ ด้านซ้าย ด้านบน และด้านล่าง

3. สามารถตรวจรู้บุคคลหลายคนที่อยู่ในภาพพร้อมทั้งตีกรอบล้อมรอบบุคคลนั้นได้

4. สามารถเลือกบุคคลใดบุคคลหนึ่งที่ต้องการติดตามได้ โดยเลือกจากสีของปลอกแขนที่บุคคลนั้นสวมใส่

5. เมื่อบุคคลเคลื่อนที่ออกนอกบริเวณที่กำหนด กล้องจะสามารถเคลื่อนที่ติดตาม บุคคลนั้น โดยกล้องจะเคลื่อนที่ให้ภาพของบุคคลนั้นกลับมาอยู่ในบริเวณที่กำหนด

1.3 ขอบเขตของปริญญานิพนธ์

1. การติดตามบุคคลที่เคลื่อนที่ สามารถติดตามเฉพาะบุคคลที่เคลื่อนที่ด้วยความเร็วไม่มากนัก

2. กล้องสามารถติดตามบุคคลที่เคลื่อนที่เข้ามาในภาพ เฉพาะบุคคลที่สวมปลอกแขนตามสีที่กำหนดเท่านั้น

1.4 ขั้นตอนการศึกษา

1. ศึกษาการรับส่งข้อมูลระหว่าง PIC กับ Computer ให้ถูกต้อง

2. ศึกษาและทดลองการเขียนโปรแกรมควบคุม Servo Motor 1 แกน และ 2 แกน

3. ศึกษาและทดลองเขียนโปรแกรมการกำจัดสัญญาณรบกวน โดยใช้วิธี Median Filter

4. ศึกษาและทดลองเขียนโปรแกรมการตีกรอบล้อมรอบบุคคลหลายๆ คนที่อยู่ในภาพ

5. ศึกษาเรื่อง Threshold ระหว่างภายในและภายนอกห้อง

6. ศึกษาและทดลองเขียนโปรแกรมการตรวจหาสีผิวโดยวิธี Gaussian และวิธีการ Normalize โดยนำค่า r g และ b มาพิจารณาในการกำหนดเป็นค่า Threshold

7. ศึกษาและทดลองการเขียนโปรแกรมเพื่อเลือกที่จะติดตามเฉพาะบุคคลที่สวมปลอกแขนตามสีที่กำหนดเท่านั้น

8. ศึกษาและเขียนโปรแกรมให้กล้องหมุนติดตามบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถติดตามบุคคลที่เคลื่อนที่โดยใช้โปรแกรม MATLAB ได้
2. สามารถออกแบบและปรับปรุงการทำงานของเซอร์โวมอเตอร์ให้สามารถบังคับให้กล้องที่ใช้ในการติดตามบุคคลสามารถเคลื่อนที่ได้ใน 4 ทิศทาง คือ ด้านซ้าย ด้านขวา ด้านบน และด้านล่าง
3. สามารถตีกรอบล้อมรอบบุคคลที่อยู่ในภาพได้
4. สามารถติดตามบุคคลใดบุคคลหนึ่งที่สวมปลอกแขนที่มีสีที่กำหนดได้
5. กล้องสามารถหมุนติดตามบุคคลได้เมื่อบุคคลนั้นเคลื่อนที่ออกนอกบริเวณที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 การมองเห็นของมนุษย์และคอมพิวเตอร์

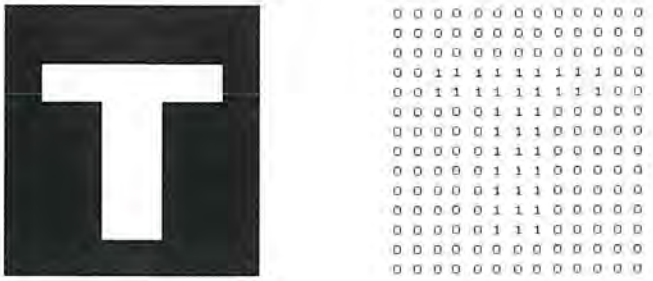
รูปภาพโดยทั่วไปไม่ว่าจะเป็นภาพที่ถ่ายโดยใช้กล้องธรรมดา หรือกล้องแบบดิจิตอลในมุมมองของมนุษย์และคอมพิวเตอร์ย่อมมีความแตกต่างกัน คือมนุษย์สามารถรับรู้ได้ว่าภาพที่ปรากฏอยู่นั้นต้องการจะสื่ออะไร เพราะว่าภาพที่มองเห็นด้วยตาของมนุษย์จะมีทั้งความกว้าง ความยาวและความลึก ส่วนคอมพิวเตอร์จะมองเห็นภาพเพียง 2 มิติเท่านั้น คือภาพมีความกว้างและความยาวและแสดงความหมายต่างๆ โดยการแทนด้วยจุดสีซึ่งเป็นค่าสีต่างๆกันนำมาเรียงต่อๆกันเพื่อที่จะสื่อถึงความหมายของภาพ

2.2 การแปลงภาพให้เป็นภาพเชิงดิจิตอล

ภาพเป็นกระบวนการทางแสง (Optical Process) ซึ่งเกิดจากพลังงานคลื่นแม่เหล็กไฟฟ้าหลายๆ ช่วงความถี่เช่น แสงธรรมดา รังสีเอ็กเรย์ (X-ray) รังสีอินฟาเรด (Infrared) เป็นต้น และพลังงานเสียง เช่น อัลตราซาวด์ (Ultrasound) ตกกระทบวัตถุแล้วสะท้อนกลับมาสู่ประสาทรับรู้ทางตาของมนุษย์หรืออุปกรณ์ตรวจจับ เช่น เซนเซอร์ (Sensor) เป็นต้น

ภาพดิจิตอลคือ ฟังก์ชัน 2 มิติ หรือ f(x,y) ของค่าความเข้มแสง โดยที่ x และ y คือค่าที่บอกถึงตำแหน่งในระบบพิกัดฉาก และค่าฟังก์ชันที่ตำแหน่งใดๆจะเป็นสัดส่วนกับความสว่างของแสงที่ตำแหน่งนั้น ส่วนกระบวนการแปลงภาพให้เป็นภาพในเชิงดิจิตอลเราเรียกว่า Image Digitization มีกระบวนการ 3 ขั้นตอน คือ การนำภาพเข้ามา (Image Acquisition) การสุ่มเลือกจุดตำแหน่ง (Image Sampling) และการประมวลค่าความเข้มของแสง (Image Quantization) ภาพที่ 2.1 แสดงการแทนภาพด้วยเมตริกซ์ โดยที่ภาพที่ 2.1 (ก) จะแสดงเฉพาะจุดสีที่ปรากฏ ส่วนภาพที่ 2.1 (ข) จะแสดงค่าประจำจุดสีที่ตำแหน่งเดียวกันของตัวอักษร T สังเกตเห็นว่าภาพตัวอักษร T ถ้าที่ตำแหน่งพิกเซลใดเป็นสีดำ ที่ตำแหน่งเดียวของภาพที่ 2.1 (ข) ก็จะแทนด้วยเป็น 0 และในทำนองเดียวกันถ้าตำแหน่งใดเป็นพิกเซลสีขาวก็จะถูกแทนด้วย 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) (ข)

ภาพที่ 2.1 การแทนภาพด้วยเมตริกซ์

2.3 การประมวลผลภาพดิจิตอลด้วยโปรแกรม

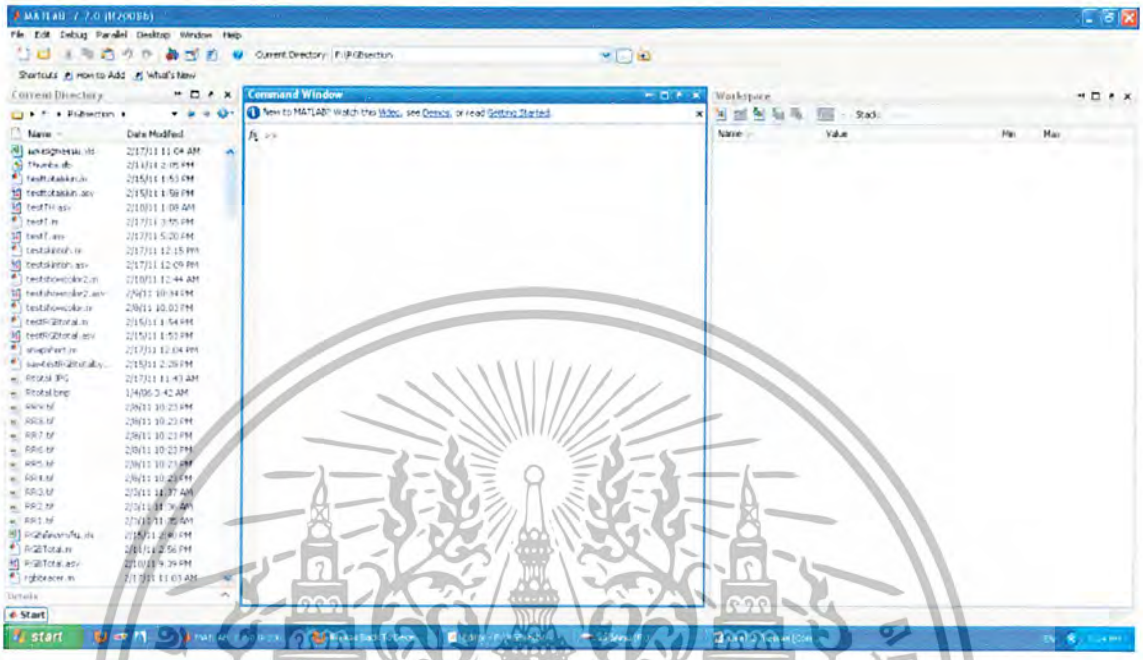
MATLAB เป็นโปรแกรมสำเร็จรูปที่นิยมใช้กันแพร่หลายในแวดวงของวิทยาศาสตร์และวิศวกรรมในปัจจุบัน โดยโปรแกรม MATLAB ได้เริ่มต้นขึ้นเพื่อต้องการให้ผู้ใช้สามารถแก้ปัญหาตัวแปรที่มีลักษณะเป็นเมตริกซ์ได้ง่ายขึ้น ทั้งในด้านการคำนวณทั่วไป การสร้างแบบจำลองและการทดสอบแบบจำลอง การวิเคราะห์ข้อมูล การแสดงผลในรูปกราฟ และทางด้านวิทยาศาสตร์และวิศวกรรม สามารถสร้างโปรแกรมในลักษณะที่ติดต่อกับผู้ใช้ทางกราฟิก

การทำงานของโปรแกรม MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง (Interactive) คือการเขียนคำสั่งเข้าไปทีละคำสั่งเพื่อให้โปรแกรม MATLAB ประมวลผลไปเรื่อยๆหรือสามารถที่จะรวบรวมชุดคำสั่งเหล่านั้นเป็นโปรแกรมได้ ข้อสำคัญอีกอย่างหนึ่งของโปรแกรม MATLAB ก็คือข้อมูลทุกตัวจะเก็บในลักษณะ Array คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆขึ้น (หรือจะได้รับการแบ่งเป็น Element นั่นเอง) ซึ่งการใช้ตัวแปรเป็น Array ในโปรแกรม MATLAB นี้ผู้ใช้ไม่จำเป็นจะต้องจอง Dimension เหมือนกับการเขียนโปรแกรมในภาษาระดับต่ำทั่วไป ซึ่งทำให้สามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะเมตริกซ์และเวคเตอร์ได้โดยง่ายซึ่งทำให้ลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียนโปรแกรมโดยใช้ภาษา C หรือ ภาษา Fortran ภาพที่ 2.2 แสดงหน้าต่างการทำงานและสภาพแวดล้อมต่างๆของโปรแกรม MATLAB จะเห็นว่ามีทั้งส่วนของ Command Window ซึ่งมีไว้เขียนคำสั่งเพื่อให้โปรแกรมทำงานทีละ 1 บรรทัด ส่วนของ Editor สำหรับการเขียนเป็นชุดคำสั่งหลาย ๆบรรทัดเพื่อประมวลผล ส่วนของ Workspace สำหรับแสดงตัวแปรและค่าของตัวแปรที่ใช้งานอยู่ในปัจจุบัน ส่วนของ Current Directory สำหรับแสดงไฟล์ต่างๆที่พร้อมเปิดใช้งาน

เหตุผลที่งานวิจัยนี้เลือกใช้โปรแกรม MATLAB เนื่องจากโปรแกรม MATLAB มีหนังสือให้ศึกษาค้นคว้ามากมายทั้งในด้านการประมวลผลภาพและด้านอื่นๆ ซึ่งแตกต่างจากโปรแกรมภาษาอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ภาษา C ภาษา JAVA ภาษาเบสิก โปรแกรม Labview เป็นต้น ซึ่งมีหนังสือให้ค้นคว้าในหัวข้อการประมวลผลภาพไม่แพร่หลายมากนัก



ภาพที่ 2.2 หน้าต่างการทำงานและสภาพแวดล้อมต่าง ๆของโปรแกรม MATLAB

2.4 ชนิดของภาพ (Image Type)

ใน Toolbox "Digital Image Processing" ของโปรแกรม MATLAB สนับสนุนชนิดของภาพดังต่อไปนี้

2.4.1 ภาพความเข้มแสง (Intensity Image)

ภาพชนิดนี้ในแต่ละพิกเซล (Pixels) จะมีค่าความเข้มของแสงในระดับที่แตกต่างกันไป ตั้งแต่สีขาวจนไปถึงสีดำ โดยสามารถกำหนดระดับความเข้มของแสงเหล่านั้นได้ โดยใช้ค่าระดับสีเทา (Gray Scale) ซึ่งโดยปกติทั่วไปแล้วภาพระดับสีเทาจะมีความละเอียด (Resolution) เท่ากับ 8 บิต ซึ่งภาพจะมีค่าระดับความเข้มแสงสีดำเท่ากับ 0 ส่วนค่าระดับความเข้มแสงของสีขาวจะมีค่าเท่ากับ 255 ภาพความเข้มของแสงแสดงได้ดังภาพที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.3 ภาพความเข้มของแสง

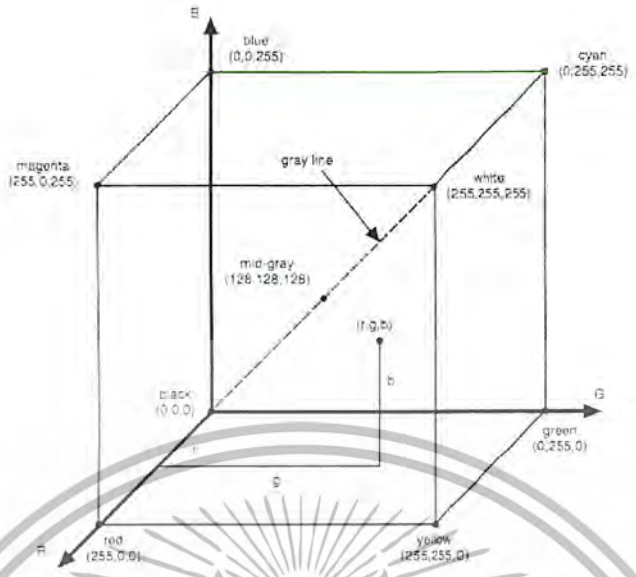
2.4.2 รูปแบบสี (color model)

2.4.2.1 RGB

ภาพชนิดนี้แต่ละพิกเซลของภาพจะเก็บค่าระดับความเข้มของแต่ละแถบแสงของแม่สีหลัก 3 สี ที่เรียงซ้อนกันคือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลนั้นๆ ก็จะแสดงผลของค่าสีแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบสีนั้น แสดงได้ดังภาพที่ 2.4

ภาพที่ 2.4 ภาพสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 แสดง RGB Color Model

RGB Color Model แสดงดังภาพที่ 2.5 ซึ่งแสดงในระบบพิกัดฉากสามมิติ โดยมีสีแดง สีเขียวและสีน้ำเงินเป็นแกนหลักของระบบพิกัดฉากสามมิติที่ตั้งฉากซึ่งกันและกัน จุดกำเนิดของระบบพิกัดฉากสามมิติจะมีค่าเป็นศูนย์ของทั้งสามสี (0,0,0) และจะมีค่าเพิ่มขึ้นไปเรื่อยๆในแต่ละแนวแกนจนกระทั่งมีค่าเท่ากับ 255 ในกรณีของภาพ 8 บิต หรือมีค่าเท่ากับ 1 ในกรณีของภาพสองระดับ ภาพที่ 2.5 แสดง RGB Color Model ในลักษณะของลูกบาศก์ ซึ่งแต่ละมุมของลูกบาศก์แสดงสีที่เป็นแกนหลัก (R, G และ B) และการผสมกันของสีหลักเหล่านั้น ดังเช่น สีม่วง (Magenta) เกิดจากการผสมกันของสีแดงและสีน้ำเงินโดยอยู่ที่พิกัด (255,0,255) และสีเหลืองเป็นการผสมกันของสีแดงและสีเขียวอยู่ที่พิกัด (255,255,0) เป็นต้น สำหรับสีขาวเกิดจากการรวมกันของสีหลักทั้งสามสีในจำนวนที่เท่ากัน และที่จุดกึ่งกลางของลูกบาศก์จะเป็นสีเทาโดยมีพิกัดที่ (128,128,128)

2.4.2.2 YCbCr

YCbCr หรือบางครั้งเรียกว่า Y'CbCr เป็น Color Space ที่ใช้เป็นส่วนหนึ่งของการส่งภาพสีในระบบวิดีโอและระบบการถ่ายภาพดิจิตอล โดยจะแบ่งข้อมูลของจุดภาพตามการรับรู้ของประสาทตาคน ซึ่งจะแยกข้อมูลความสว่างออกจากข้อมูลที่เกี่ยวกับระดับสี นั่นคือใช้องค์ประกอบของ Y เก็บค่าความสว่างของจุดภาพ ส่วน Cb เป็นค่าความแตกต่างระหว่างค่าสีน้ำเงินกับค่าความสว่าง (B-Y) และCr เป็นค่าความแตกต่างระหว่างค่าสีแดงกับค่าความสว่าง (R-Y) อย่างไรก็ตามจุดภาพในระบบ RGB และ YCbCr ก็สามารถแปลงไปเป็นอีกระบบหนึ่งได้เสมอ โดยหากต้องการแปลงจากระบบ RGB เป็นระบบ YCbCr ก็สามารถแปลงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\left.\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= 0.564\,(B-Y) \\ Cr &= 0.713\,(R-Y) \end{aligned}\right\} \quad (1)$$

ในทางกลับกัน การแปลงจากระบบ YCbCr เป็นระบบ RGB ก็สามารถทำได้โดยใช้สมการต่อไปนี้

$$\left.\begin{aligned} R &= Y + 1.402Cr \\ G &= Y - 0.0344Cb - 0.714Cr \\ B &= Y + 1.772Cb \end{aligned}\right\} \quad (2)$$

2.4.2.3 YUV

YUV เป็นโมเดลสีซึ่งมีหลักการเดียวกันกับ YCbCr โดยใช้องค์ประกอบ Y เก็บค่าความสว่างของจุดภาพ องค์ประกอบ U เก็บค่าความแตกต่างระหว่างค่าสีน้ำเงินและค่าความสว่าง ส่วนองค์ประกอบของV เก็บค่าความแตกต่างระหว่างค่าสีแดงและค่าความสว่าง หรืออาจจะกล่าวว่าใช้องค์ประกอบ U และ V แทนองค์ประกอบ Cb และ Cr ตามลำดับ มีหลายครั้งที่เรียกชื่อ YCbCr กับ YUV สลับกันทั้งๆที่ YCbCr ใช้ในระบบอนาล็อก ส่วน YUV ใช้ในระบบดิจิตอล

2.4.2.4 HSV

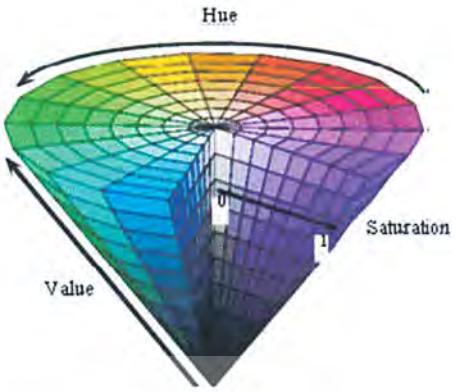
โมเดลสีนี้ประกอบด้วย 3 องค์ประกอบได้แก่ค่า H (Hue) คือค่าโทนสีหรือค่าสีสันซึ่งมีค่าสีสันที่แตกต่างกันไปตามความถี่ของแสง ค่า S (Saturation) คือค่าความอิ่มตัวของสีซึ่งเป็นค่าที่แสดงระดับสีเมื่อเทียบกับค่าโทนสี และค่า V (Value) คือค่าบอกระดับความสว่างของภาพ ซึ่งที่ระดับความสว่างต่ำสุดจะมีสีดำโดยไม่คำนึงถึงว่าค่าโทนสีและค่าความอิ่มตัวของสีจะเป็นเท่าใด และระดับความสว่างสูงสุดจะเป็นสีขาว ซึ่งเป็นสีที่สว่างที่สุดของค่าโทนสีและค่าความอิ่มตัวสี ค่า HSV สามารถหาได้จากการแปลงภาพสี RGB ด้วยสมการดังต่อไปนี้

$$H = arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \quad (1)$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B} \quad (2)$$

$$V = \frac{1}{3}(R+G+B) \quad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 แบบจำลองกรวยภาพสีของ HSV

ภาพที่ 2.6 แสดงแบบจำลองกรวยภาพสีของ HSV จากรูปส่วนที่เป็นฐานของกรวยแสดงค่าโทนสีหรือค่าสีสัน (Hue) ที่แตกต่างกันไปตามความถี่ของแสง ส่วนผิวของกรวยแสดงค่าความสว่างของสี (Value) โดยที่ยอดกรวยจะมีค่าความสว่างน้อยที่สุด และความสว่างจะมากที่สุดที่ฐานของกรวย สำหรับค่าความอิ่มตัวของสี (Saturation) จะเริ่มจากแกนกลางของกรวยจนกระทั่งถึงผิวของกรวย โดยค่าความอิ่มตัวของสีจะมีค่ามากที่สุดที่ผิวของกรวย

2.4.3 ภาพขาวดำ (Binary Image)

ภาพขาวดำ (Binary Image) คือในแต่ละพิกเซลจะแสดงด้วยค่าไบนารี (Binary) มี 2 บิตซึ่งประกอบไปด้วยค่า 1 หรือ 0 โดยที่ 1 หมายถึงจุดภาพสีขาว และ 0 หมายถึงจุดภาพสีดำ โดยที่ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวข้องกับอักษร (Text) ภาพลายนิ้วมือ เป็นต้น ภาพขาวดำแสดงได้ดังภาพที่ 2.7

ภาพที่ 2.7 ภาพขาวดำ

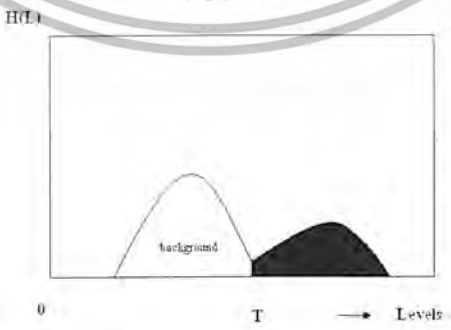
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การแบ่งส่วนภาพ (Image Segmentation)

การแบ่งส่วนภาพเป็นขั้นตอนที่สำคัญก่อนการวิเคราะห์ภาพ เนื่องจากเป็นขั้นตอนที่เราจะแยกวัตถุที่เราสนใจออกจากพื้นหลัง การแบ่งส่วนภาพจะทำการแบ่งภาพออกตามองค์ประกอบหรือวัตถุที่อยู่ในภาพ ระดับของการแบ่งมักขึ้นกับปัญหาที่ศึกษา อย่างไรก็ตามในที่นี้จะขอกล่าวถึงการการแบ่งส่วนภาพโดยการตั้งค่าเทรสโชว์เพียงค่าเดียว

เทรสโชว์ (Threshold) คือการกำหนดค่าระดับความเข้มสีเทาคงที่ที่ค่าหนึ่ง เพื่อทำการแยกสิ่งที่ต้องการ (วัตถุ) ออกจากพื้นหลัง (Background) และยังเป็นอีกขั้นตอนหนึ่งเพื่อใช้ในการสร้างภาพไบนารี ซึ่งในการกำหนดค่าเทรสโชว์ ถ้ากำหนดค่าเทรสโชว์ไม่เหมาะสม เช่น ค่าเทรสโชว์ที่มีค่าน้อยหรือมากเกินไป อาจทำให้รายละเอียดบางส่วนของวัตถุที่ต้องการขาดหายไปหรืออาจมีสิ่งไม่พึงประสงค์ปะปนมาด้วย เช่น สัญญาณรบกวน (Noise) เป็นต้น ดังนั้นการกำหนดค่าเทรสโชว์ที่เหมาะสมจึงเป็นสิ่งสำคัญ ซึ่งในปัจจุบันมีผู้เสนอวิธีในการหาค่าเทรสโชว์หลายวิธีที่เหมาะกับการนำไปใช้ในงานที่มีลักษณะแตกต่างกันไป

การทำเทรสโชว์โกบอลแบบง่าย เป็นวิธีการสำหรับการแบ่งส่วนของวัตถุออกมาจากพื้นหลังของภาพ โดยการพิจารณาจากฮิสโตแกรมของภาพ ซึ่งกลุ่มของข้อมูลของวัตถุและพื้นหลังจะแยกออกเป็นสองกลุ่มตามการกระจายของข้อมูล โดยการใช้ระดับเทรสโชว์เดียว (Threshold) T ดังแสดงในภาพที่ 2.8 การแบ่งส่วนภาพ (Segmentation) เริ่มต้นด้วยการพิจารณาว่าพิกเซลไหนเป็นส่วนของวัตถุ และพิกเซลไหนเป็นส่วนของพื้นหลัง ซึ่งการตัดสินใจขึ้นอยู่กับว่าค่าระดับสีเทาของพิกเซลนั้นๆ ว่ามีค่ามากกว่าหรือน้อยกว่าค่าระดับเทรสโชว์ T ถ้าพิกเซลใดมีค่าระดับสีเทามากกว่าค่าเทรสโชว์ T พิกเซลนั้นจะถูกกำหนดให้เป็นวัตถุในทางตรงกันข้าม ถ้าน้อยกว่าค่าเทรสโชว์ T พิกเซลนั้นจะถูกกำหนดให้เป็นพื้นหลัง อย่างไรก็ตาม ความสำเร็จของวิธีนี้ขึ้นอยู่กับว่าเราสามารถแบ่งส่วนฮิสโตแกรมได้ดีเพียงใด



ภาพที่ 2.8 การตั้งค่าเทรสโชว์เพื่อแบ่งข้อมูลในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.8 จะพบว่าระหว่างส่วนที่เป็นวัตถุกับพื้นหลังจะถูกกั้นด้วยค่า T ซึ่งก็คือค่าเทรสโชว์ที่ถูกกำหนดขึ้น ในรูปที่ 2.8 เป็นการกำหนดค่าเทรสโชว์ที่ 100 ทำให้สามารถแยกวัตถุออกมาจากพื้นหลังได้ การแบ่งส่วนภาพและการแปลงภาพไบนารีสามารถทำได้ดังนี้

$$g(x,y) = \begin{cases} 1 \; if \;\; f(x,y) \geq T \\ 0, \; otherwise \end{cases} \qquad (4)$$

โดยที่ f(x,y) คือภาพอินพุต g (x,y) คือภาพผลลัพธ์หรือภาพไบนารี ซึ่งจะมีค่าเท่ากับหนึ่ง ถ้า f(x,y) มากกว่าหรือเท่ากับ T และจะเท่ากับศูนย์ถ้าไม่เป็นไปตามเงื่อนไข

2.6 การกำจัดสัญญาณรบกวนด้วย Median Filter

Median Filter เป็นตัวกรองความถี่ที่พิจารณาจากข้อมูลทางสถิติ โดยใช้ค่ามัธยฐาน (Median) การหาค่ามัธยฐานทำได้ โดยการนำข้อมูลในมาร์ก (Mark) มาทำการเรียงค่าจากน้อยไปมากตามค่าระดับความเข้มสีเทาในกรณีของภาพระดับสีเทา หรืออาจเรียงตัวเลขศูนย์และหนึ่งในกรณีเป็นภาพไบนารี ซึ่งค่ามัธย-ฐานเป็นค่าตำแหน่งกึ่งกลางของกลุ่มข้อมูลที่พิจารณา จากนั้นนำค่ามัธยฐานที่ได้นั้นแทนค่ากลับลงไปในพิกเซลตำแหน่งตรงกลางของหน้าต่างจะได้ภาพผลลัพธ์ดังภาพที่ 2.9

ภาพที่ 2.9 การนำค่ามัธยฐานที่ได้นั้นแทนค่ากลับลงไปในพิกเซลตำแหน่งตรงกลางของหน้าต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวกรองความถี่แบบ Median Filter สามารถเลือกใช้ได้หลายขนาดขึ้นอยู่กับความเหมาะสมและปริมาณของสัญญาณรบกวน เช่น หน้าต่างขนาด 3x3 5x5 และ 7x7 เป็นต้น

2.7 ทฤษฎีเบื้องต้นทางสถิติ

2.7.1 ค่าเฉลี่ยเลขคณิต (Arithmetic Mean)

ค่าเฉลี่ยเลขคณิตคือผลรวมของข้อมูลทั้งหมดหารด้วยจำนวนข้อมูลที่ได้จากการวัด สมมติข้อมูลที่ได้จากการวัดคือ 7, 13, 22, 9, 11, 4 ผลรวมของข้อมูลชุดนี้คือ 66 ค่าเฉลี่ยจึงมีค่าเท่ากับ 66 หารด้วย 6 เท่ากับ 11 ในกรณีทั่วไป ถ้าข้อมูลที่ได้จากการวัดมี N จำนวน ถูกนำเสนอโดยใช้สัญลักษณ์ $X_1, X_2, ..., X_N$ และการหาค่าเฉลี่ยเลขคณิตสามารถเขียนเป็นสูตรได้ดังนี้

$$\bar{X} = \frac{x_1 + x_2 + x_3 + \ldots + x_N}{N} = \sum_{i=1}^{N} x_i \tag{5}$$

สัญลักษณ์ $\bar{X}$ อ่านว่าเอ็กซ์บาร์ ใช้แทนค่าเฉลี่ยเลขคณิตของข้อมูล อักษรกรีซ Σ อ่านว่า ซิกมา, แทนผลรวมของข้อมูลที่ได้จากการวัด N จำนวน ผลรวมจาก i = 1 ถึง i = N

2.7.2 ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation : s.d.)

ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation: s.d.) ในทางสถิติศาสตร์และความน่าจะเป็นเป็นการวัดการกระจายแบบหนึ่งของกลุ่มข้อมูล สามารถนำไปใช้กับการแจกแจงความน่าจะเป็น ตัวแปรสุ่ม ประชากร หรือมัลติเซต ค่าเบี่ยงเบนมาตรฐานมักเขียนแทนด้วยอักษรกรีกซิกมาตัวเล็ก (σ) นิยามขึ้นจากส่วนเบี่ยงเบนแบบ Root Mean Square (RMS) กับค่าเฉลี่ย หรือนิยามขึ้นจากรากที่สองของความแปรปรวน

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{6}$$

เมื่อ x_i คือค่าของสมาชิกแต่ละตัว

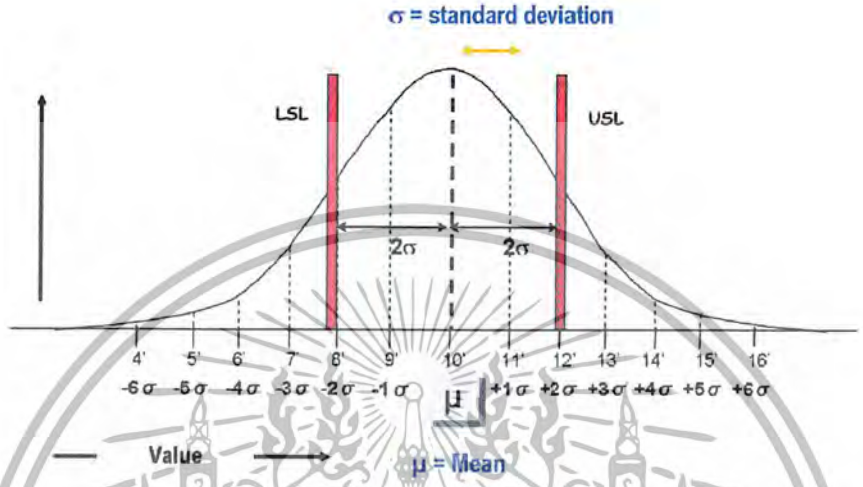
$\bar{x}$ คือค่าเฉลี่ยเลขคณิต

N คือจำนวนสมาชิกทั้งหมด

ค่าเบี่ยงเบนมาตรฐานหรือซิกมามีที่มาจากการประยุกต์ความรู้ทางด้านสถิติ โดยสมมติให้ปรากฏ การณ์ที่เกิดขึ้นในระบบนั้นเป็นการแจกแจงปกติ (Normal Distribution) หรือการกระจายเป็นรูประฆังคว่ำทั้งหมดดังภาพที่ 2.10 ค่าเฉลี่ยที่จุดกึ่งกลางของการกระจายตัวนั้นก็คือค่าที่ต้องการ ส่วนซิกมาคือหนึ่งช่วงของความเบี่ยงเบนมาตรฐานที่วัดจากจุดกึ่งกลางดังกล่าว และจะมีขอบเขตของการยอมรับได้อยู่ 2 ส่วนคือ ขอบเขตจำกัดบน (Upper Specific Limitation) และขอบเขตจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล่าง (Lower Specific Limitation) ซึ่งในนิยามของซิกมานี้ ถ้าขอบเขตบนและล่างอยู่ห่างจากค่าเฉลี่ยเป็นระยะ 3 ซิกมา ก็จะเรียกว่า ระดับ 3 ซิกมา (3 Sigma Level) แต่ถ้าเป็นระยะ 4 ซิกมา ก็จะเรียกว่า ระดับ 4 ซิกมา (4 Sigma Level) ซึ่งในแต่ละระดับจะให้ค่าความน่าเชื่อถือดังตารางที่ 1



ภาพที่ 2.10 การกระจายเป็นรูประฆังคว่ำ

ตารางที่ 2.1 แสดงค่าความน่าเชื่อถือของแต่ละซิกมา

ระดับซิกมา	ค่าความน่าเชื่อถือ (Reliability)
1.0	30.23278734 %
2.0	69.12298322 %
3.0	93.31894011 %
4.0	99.37903157 %
5.0	99.97673709 %
6.0	99.99966023 %

จากตารางแสดงค่าความน่าเชื่อถือของซิกมา จะพบว่าค่าซิกมาที่เพิ่มขึ้นจะมีระดับความน่าเชื่อถือมากขึ้น

2.8 การ label วัตถุ

ถ้าภาพมีหลายๆ วัตถุเราสามารถกำหนดหมายเลขวัตถุโดยใช้วิธีการ Label หรือให้หมายเลขกำกับที่วัตถุแต่ละวัตถุ ซึ่งโปรแกรมจะกำหนดวัตถุแต่ละวัตถุในภาพเป็นวัตถุที่ 1,2,3,...ตามลำดับจนครบหมดทุกวัตถุ ซึ่งการ Label วัตถุแต่ละวัตถุทำได้โดยใช้คำสั่ง bwlabel ซึ่งเป็นคำสั่งใน Image Processing ของ MATLAB คำสั่ง bwlabel เป็นการกำหนดหมายเลขให้กับวัตถุในภาพที่เป็นภาพไบนารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมกรณีใช้คำสั่ง bwlabel ของ MATLAB กับภาพวัตถุจำลอง (BW)

L=bwlabel(BW,n)

BW =

0	0	0	0	1	1	0	0	0	0	1	1	1	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	1	1	1	1	0	0
0	0	0	0	1	1	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ภาพที่ 2.11 ภาพที่มีวัตถุ 3 วัตถุ

L =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	3	3	3	0	0
0	0	0	1	1	1	1	0	0	3	3	3	3	0	0
0	0	0	1	1	1	1	0	0	3	3	3	3	0	0
0	0	0	0	1	1	0	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
0	0	0	0	2	2	2	0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ภาพที่ 2.12 ภาพผลลัพธ์ภายหลังจากคำสั่ง bwlabel

ภาพผลลัพธ์ภายหลังจากใช้คำสั่ง bwlabel แสดงดังภาพที่ 2.12 ซึ่งวัตถุที่อยู่แยกกันถูกกำหนดด้วยหมายเลขที่แตกต่างกัน จากนั้นเราสามารถหาตำแหน่งของแถว(Row) และหลัก (Column) ของวัตถุแต่ละวัตถุในภาพได้ โดยใช้คำสั่ง find

ตัวอย่างการหาตำแหน่งของ แถว(Row) และ หลัก (Column)วัตถุเลข 3

[r,c] = find(L==3); หาค่า แถว(Row) และหลัก (Column)ของวัตถุที่มีกำหนดเลข 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rc = [r c] แสดงค่า แถว (Row) และ หลัก (Column) ของวัตถุที่มีเท่ากับ 3ซึ่งคำตอบที่ได้คือ

r	3	4	2	3	4	5	2	3	4	2	3	4
c	10	10	11	11	11	11	12	12	12	13	13	13

2.9 การติดต่อของโปรแกรม MATLAB กับกล้อง Webcam

การใช้โปรแกรม MATLAB เพื่อติดต่อกับกล้อง Webcam และรับภาพจากกล้องเข้ามาประมวลผลภาพ เนื่องจากกล้อง Webcam จึงเป็นอุปกรณ์สำคัญที่จะนำข้อมูลภาพเข้ามาสู่การประมวลผลภาพ การติดต่อกล้องกับคอมพิวเตอร์จะต้องมีการติดตั้งไดรฟ์เวอร์ที่ให้มากับตัวกล้อง จากนั้นก็ทำการติดต่อตัวกล้องกับคอมพิวเตอร์ทางพอร์ต USB และเมื่อทำการให้กล้องติดต่อกับคอมพิวเตอร์ได้แล้ว จึงสั่งให้โปรแกรม MATLAB รับภาพจากกล้อง Webcam โดยใช้รูปแบบคำสั่งดังต่อไปนี้

vid = videoinput(*adaptorname*,deviceID,*format*)

โดยที่ Videoinput คือคำสั่งในการรับภาพจากกล้อง Webcam , *adaptorname* คือ ชื่อของ Adaptor ที่ถูกใช้เพื่อติดต่อกับกล้อง, deviceID คือค่าตัวเลขที่บ่งชี้ถึงอุปกรณ์เฉพาะที่กำหนดไว้ผ่านทาง Adaptor ที่กำหนด , *format* คือข้อมูล Text ที่บ่งชี้ถึง Video Format ที่ Support กับกล้อง

เพื่อรับภาพจากกล้อง Webcam แต่การรับภาพเข้ามาจะต้องขึ้นอยู่กับคุณสมบัติของกล้องที่ใช้ด้วย ซึ่งในงานโครงงานนี้ใช้กล้องที่สามารถถ่ายภาพได้ขนาด 640x480 แต่ใช้งานที่ขนาดภาพ 320x240 ดังนั้นจึงกำหนดขนาดภาพเป็น 320x240 ใน Video Format จากนั้นใช้คำสั่ง preview (vid) ในการแสดงภาพวิดีโอที่รับเข้ามา อย่างไรก็ตามการเก็บภาพจากกล้องโดยใช้คำสั่ง getsnapsot(vid) แล้วเก็บบันทึกข้อมูลภาพลงเป็น File จะได้ภาพที่มีสีแดงไม่ใช่ภาพสี RGB ดังนั้น เพื่อให้ภาพที่ถูกบันทึกเป็นภาพสี RGB จึงจำเป็นต้องใส่คำสั่ง ReturnedColorspace ก่อนคำสั่ง getsnapshot ดังแสดงในตัวอย่างโปรแกรมข้างล่าง

ตัวอย่างโปรแกรมที่ใช้ในการติดต่อโปรแกรม MATLAB กับกล้อง Webcam

```
clear all;                                          % ล้างตัวแปรทั้งหมดในหน่วยความจำ
close all;                                          % ล้างรูปทั้งหมดในหน่วยความจำ
vid=videoinput('winvideo',1,'YUY2_320x240')         % ใช้คำสั่ง Video input เพื่อรับภาพจากกล้อง
preview(vid)                                        % สั่งแสดงภาพวิดีโอ
set(vid, 'ReturnedColorspace', 'RGB');              % แปลงภาพกลับไปเป็นรูปแบบสี RGB
pause(8)                                            % หยุด 8 วินาที
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


```
pic1=getsnapshot(vid);                  %ถ่ายรูปจากภาพที่รับเข้ามาเก็บในตัวแปรpic1
imwrite(pic1,'61A.tif')                 % เก็บบันทึกในไฟล์ชื่อ 61A.tif
figure(8),imshow(pic1)                  % แสดงภาพที่ถ่าย
delete(vid);                            % ลบวิดีโอที่รับเข้ามา
```

119568

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการวิจัย

การเขียนโปรแกรมด้วยภาษาระดับกลาง หรือสูง เช่น ภาษา C หรือภาษาเบสิกสามารถช่วยลดเวลาในการเขียนโปรแกรมซึ่งมีความสะดวกในการใช้งาน และเมื่อมีความจำเป็นต้องเปลี่ยนตระกูลของไมโครคอนโทรลเลอร์ การเรียนรู้ใหม่ หรือการแก้ไขโปรแกรมเดิมสามารถทำได้ไม่ยากนักและใช้เวลาน้อยกว่า เมื่อเทียบกับการพัฒนาโปรแกรมด้วยภาษาแอสแซมบลี แต่ก็ไม่ได้หมายความว่าภาษาแอสแซมบลีไม่สำคัญหรือไม่ต้องเรียนรู้เลย เพียงแต่เรียนรู้เฉพาะเท่าที่ต้องนำมาใช้งานและเฉพาะแกนหลัก เนื่องจากโปรแกรมภาษาระดับกลาง หรือสูงเองก็มีข้อจำกัดเมื่อต้องลงไปกระทำงานในระดับบิต หรือในงานที่ต้องเกี่ยวข้องกับความเร็ว

3.1 ภาษาซี (C Programming Language)

ภาษาซีเป็นภาษาคอมพิวเตอร์ที่ได้รับความสนใจสูงมากจากผู้เริ่มต้นใช้งาน และผู้ที่ทำงานในระดับมืออาชีพ เพราะนอกจากจะง่ายต่อการเริ่มต้นเรียนรู้ แล้วก็ยังง่ายต่อการพัฒนางาน ต่อไปด้วย โดยเฉพาะอย่างยิ่งกับงานระบบควบคุมอัตโนมัติที่ใช้ในไมโครคอนโทรลเลอร์ ตระกูล PIC

3.1.1 PIC C Compiler

PIC C Compiler เป็นโปรแกรมที่ถูกพัฒนา โดยบริษัท CCS inc ที่ได้รับความนิยมอย่างกว้างขวางในหมู่นักพัฒนาและผู้เริ่มต้นด้วยเนื่องจากเป็นการเขียนด้วยภาษาซี ซึ่งง่ายต่อการเรียนรู้ อีกทั้งยังไม่จำเป็นต้องเข้าไปศึกษารายละเอียดฮาร์ดแวร์ของชิฟไมโครคอนโทรลเลอร์ ก็ยังสามารถใช้งานได้

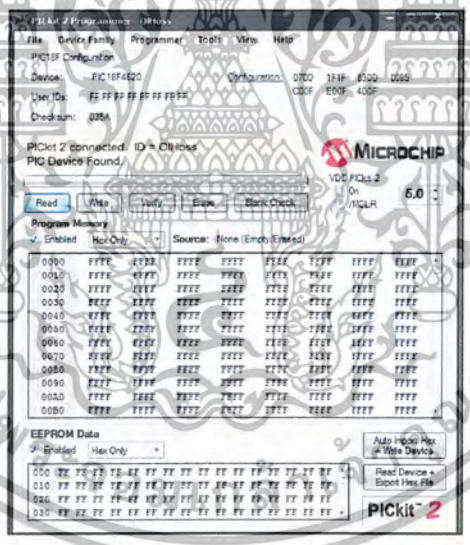
ในการพัฒนาระบบงานของไมโครคอนโทรลเลอร์ PIC ด้วยโปรแกรม โดยใช้ PIC C Compiler ผู้พัฒนางานสามารถสร้างฮาร์ดแวร์ได้เอง จากนั้นทำการเขียนภาษาซี แล้วคอมไพล์เป็น .hex ไฟล์ เพื่อทำการโปรแกรมลงในไมโครคอนโทรลเลอร์ PIC โดยใช้เครื่องโปรแกรมหรือบอร์ดที่สามารถโปรแกรมลงไมโครคอนโทรลเลอร์ได้ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.1 แสดงหน้าต่างโปรแกรม PIC C Compiler

3.1.2 โปรแกรม PICkit2 V2.61

PICkit2 Programming Software เป็นซอฟต์แวร์สำหรับโปรแกรมไมโครคอนโทรลเลอร์ PIC และ dsPIC แบบแฟลชที่ต้องทำงานร่วมกับเครื่องโปรแกรม PICkit2 ของ Microchip หรือเครื่องโปรแกรมอื่นที่มีการทำงานตรงกัน ในที่นี้ใช้บอร์ดโปรแกรมรุ่น ET-PGM PIC USB V2.0



ภาพที่ 3.2 หน้าต่างโปรแกรม PICKit2 V2.61

3.2 อุปกรณ์และฮาร์ดแวร์

3.2.1 คอมพิวเตอร์ Notebook Lenovo รุ่น G450

Notebook Lenovo รุ่น G450 ทำหน้าที่ติดต่อกับกล้องเว็บแคมเก็บภาพที่ได้จากกล้องและทำการประมวลผลด้วยโปรแกรม MATLAB เพื่อคำนวณหาวัตถุที่เคลื่อนที่ในภาพ และตัดสินใจให้เซอร์โวมอเตอร์เคลื่อนที่เมื่อวัตถุเคลื่อนที่เกินระยะขอบเขตที่กำหนดโดยส่งสัญญาณให้ไมโครคอน-โทรลเลอร์ (PIC) เพื่อควบคุมเซอร์โวมอเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.3 คอมพิวเตอร์ Notebook Lenovo รุ่น G450

คุณสมบัติของคอมพิวเตอร์ Notebook Lenovo รุ่น G450

- Intel Core 2 Duo T6600 (2.20 GHz, 2 MB L2 Cache, 800 MT/s FSB)
- Chipset Mobile Intel GL40 Express Chipset
- nVidia GeForce G 105M
- Memory 2 GB DDR3
- Hard Disk 320 GB 5400 RPM

3.2.2 ET-PGMPIC USB V 2.0

ET-PGMPIC USB คือเครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล PIC ซึ่งเป็นไมโครคอนโทรลเลอร์ของ บริษัทไมโครชิพ (microchip) มีคุณสมบัติเทียบเท่ากับเครื่องโปรแกรม PicKit 2 ของไมโครชิพ โดยสามารถโปรแกรมไมโครคอนโทรลเลอร์ PIC ที่มีหน่วยความจำแบบ Flash Memory ได้หลากหลายเบอร์ด้วยกัน (สามารถดูรายการเบอร์ไมโครคอนโทรเลอร์ PIC ที่ ET-PGMPIC USB ได้ในไฟล์ README ของซอร์ฟแวร์โปรแกรม PICKit 2)

ภาพที่ 3.4 ET-PGM PIC USB V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC 18F4620

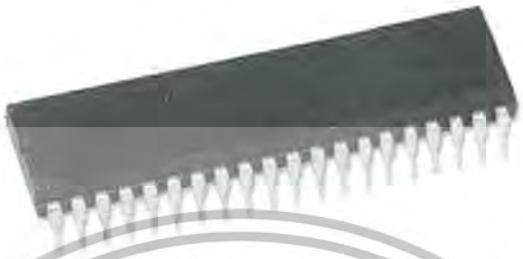
ไมโครคอนโทรลเลอร์ได้มีการพัฒนามาอย่างต่อเนื่อง และในปัจจุบันมีไมโครคอนโทรลเลอร์ตระกูลใหม่ๆ เกิดขึ้นมามากมาย ล้วนแล้วแต่มีศักยภาพในการทำงานสูงด้วยกันทั้งสิ้น แต่ในที่นี้จะกล่าวถึงไมโครคอนโทรลเลอร์ตระกูล PIC ของบริษัท MICROCHIP ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถ และมีทรัพยากรหรือฟังก์ชันการใช้งานต่างๆ มากมายตัวอย่าง เช่น โมดูล Analog to Digital, Timer/Counter, USART, SPI, I^2C, PWM และอื่นๆ ซึ่งส่วนต่าง ๆ เหล่านี้จะถูกสร้างรวมอยู่ภายใน CPU เพียงตัวเดียวทำให้ CPU เพียงตัวเดียวนี้สามารถทำงานได้หลายๆอย่างและสามารถลดภาระในส่วนของฮาร์ดแวร์บางอย่างลง ส่วนในเรื่องของความเร็ว CPU ตระกูลนี้จะใช้เวลาในการกระทำคำสั่งต่าง ๆเพียง 1 หรือ 2 ไซเคิล ต่อคำสั่งเท่านั้น โดยการทำงานนี้จะเป็นลักษณะไปป์ไลน์ (Pipe line) ทำให้มีความเร็วในการทำงานมากกว่า CPU ทั่วไป

คุณสมบัติต่าง ๆ ของไมโครคอนโทรลเลอร์ PIC 18F4620

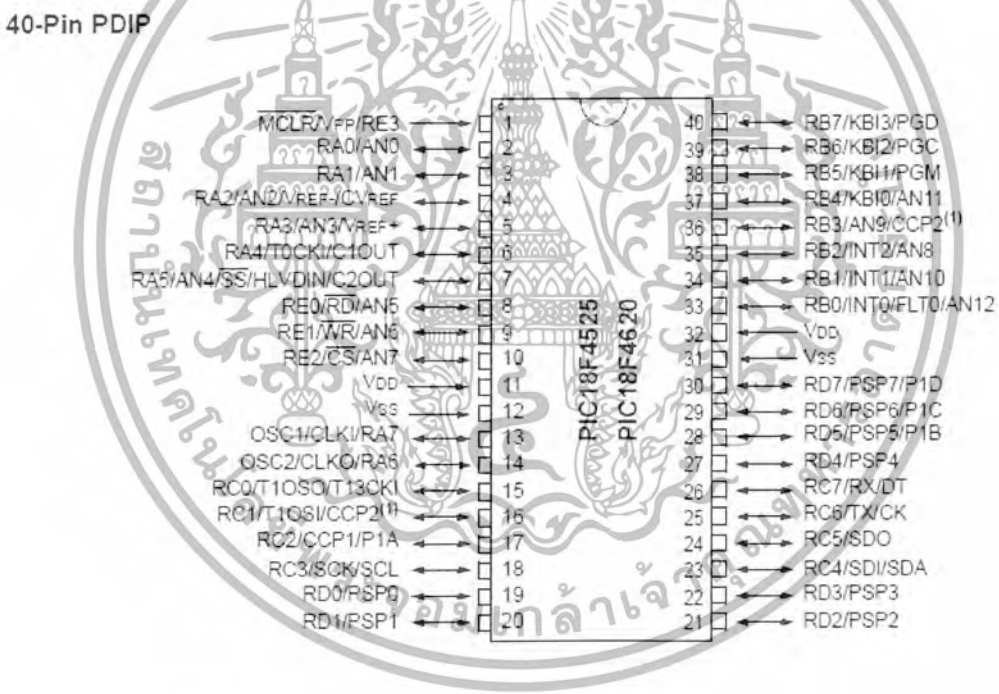
- มีคำสั่งในภาษา Assembly 77 คำสั่ง
- มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM, CAN
- มี Pin ทั้งหมด 40 Pin
- มี A/D ขนาด 10 bits
- ความถี่สูงสุดที่ทำงานได้คือ 10 MHz
- สามารถต่อกับ Program Memory ภายนอก โดยอ้างได้ถึง 64K x 16 Program Memory
- มี Program Memory เป็นแบบ Flash ทำให้สามารถโปรแกรมใหม่ได้หลายครั้ง
- มีหน่วยความจำข้อมูลภายใน (EPROM) 1,024 Byte
- มีหน่วยความจำข้อมูลภายใน (RAM) 3,968 Byte
- สนับสนุน In Circuit Debugging (ICD)
- ทำงานเมื่อมีไฟเลี้ยง 2.0V-5.5V
- Analog to Digital Converter ความละเอียด 10 บิต 13 แชนแนล ภายในตัว
- สามารถโปรแกรมโดยใช้แรงดัน +5V
- Timer/Counter จำนวน 4 ตัว คือ Timer 0 , Timer 1 , Timer 2 ,Timer 3
- โมดูล Capture/Compare/PWM จำนวน 5 ชุด
- มีโมดูล USART
- ทนความร้อน -40 ถึง 125 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวโน้มในอนาคตไมโครคอนโทรลเลอร์ เริ่มจะมี Feature ต่างๆรวมเข้าไปมากขึ้น สำหรับ In Circuit Debugger ต่อไปคงเป็นมาตรฐานของ MCU ทั่วๆไป ต่อไปอาจจะมีการรวม FPGA หรือ DSP เข้าไปด้วย



ภาพที่ 3.5 โครงสร้างภายนอกของ PIC 18F4620



ภาพที่ 3.6 แสดงโครงสร้าง CPU PIC 18F4620 และการจัดวางขาสัญญาณ

3.2.4 เซอร์โวมอเตอร์ (Servo motor)

เซอร์โวมอเตอร์คือมอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ประกอบด้วยชุดเกียร์และส่วนควบคุมต่างๆ ไว้ในโมดูลเดียวกัน หรือภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC, GND และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวา ได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วิดมอด (PWM) แบบ TTL Level +90 องศา -90 องศา (180 องศา) โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

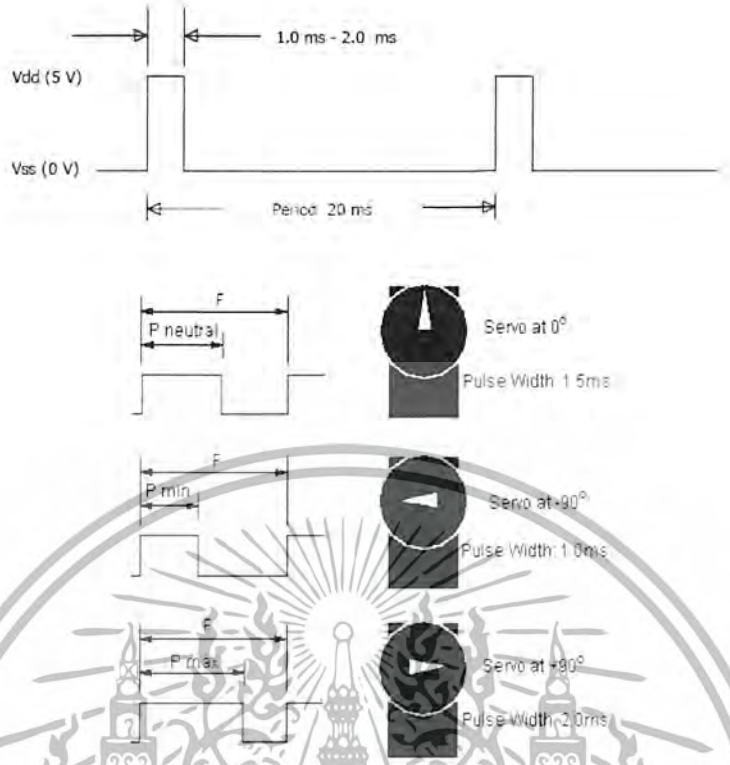
สามารถสั่งงานในการหมุนไปตามองศาต่างๆ ที่ต้องการได้ด้วยตัวของเซอร์โวมอเตอร์เอง ข้อดีของมอเตอร์ชนิดนี้คือมีขนาดเล็กและน้ำหนักเบา ให้แรงบิตสูงกินพลังงานน้อย สามารถควบคุมด้วยแรงดันที่เป็นลอจิก TTL ได้โดยตรง ไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่น ๆ เพราะมอเตอร์ชนิดนี้มีวงจรควบคุมภายในอยู่แล้ว ซึ่งสามารถควบคุมให้หมุน ในตำแหน่งหรือทิศทางที่ต้องการได้โดยอาศัยความกว้างของพัลส์ ที่ป้อนให้มอเตอร์ ในการทดลองครั้งนี้ ได้เลือกใช้ เซอร์โวมอเตอร์ของบริษัท GWS รุ่น S03N/STD/J เป็นรุ่นสำหรับใช้งานทั่วไป นำมาใช้งานในการหมุนกล้องเว็บแคมในแนวแกน x เพราะกล้องมีน้ำหนักเบา และ S03T/2BBMG/J เป็นรุ่นที่มีจุดหมุนเป็น Ball Bearing และเฟืองในเป็นโลหะทองเหลือง ซึ่งมีความทนทานในการใช้งานและมีทอร์กสูงนำมาใช้งานในการหมุนแกน y เนื่องจากชุดขาตั้งกล้องซึ่งค่อนข้างมีน้ำหนักมาก

ภาพที่ 3.7 ส่วนประกอบของเซอร์โวมอเตอร์

หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของเซอร์โวมอเตอร์ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้ จะขึ้นอยู่กับขนาดความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดอ้างอิง 3 จุดดังรูปคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 แสดงทิศทางการหมุนของเซอร์โวมอเตอร์เมื่อจ่ายพัลส์ขนาดต่างๆ

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

การที่จะควบคุมเซอร์โวมอเตอร์หมุนเป็นมุมอื่นๆนั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่าง ๆ โดยอ้างอิงจากจุดทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่าง เช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็ต้องจะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25ms เป็นต้นและสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

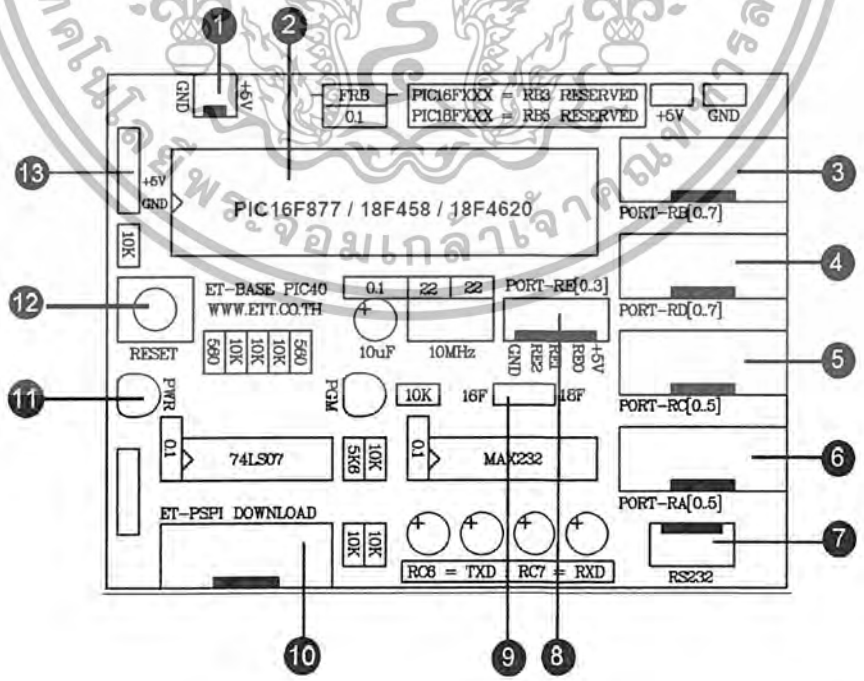
3.2.5 บอร์ด ET-BASE PIC40

ET-BASE PIC40 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล PIC ขนาด 40 PIN ของบริษัท Microchip ซึ่งในเวอร์ชันนี้ได้นำเอา PIC MCU มาจัดวงจรใช้งานให้มีขนาดกะทัดรัดโดยเน้นการใช้งานทรัพยากรของ PIC MCU เป็นหลัก นอกจากนี้ยังออกแบบให้สนับสนุนการนำไปใช้งานร่วมกับบอร์ดทดลอง "ET-BASIC IO" อีกด้วย

คุณสมบัติของบอร์ดที่นำมาใช้ในการวิจัย

- รองรับการใช้งานไมโครคอนโทรเลอร์ขนาด 40 PIN คือ PIC16F877, PIC18F458 และ PIC18F4620
- สัญญาณนาฬิกาคริสตอลออสซิลเลเตอร์ขนาด 10 MHz
- I/O Port ขนาด 10 PIN (จัดเรียงตามมาตรฐานของ อีทีที) จำนวน 4 พอร์ต
- I/O Port ขนาด 5 PIN จำนวน 1 พอร์ต
- ชุดวงจรไดร์ฟเวอร์ RS232 จำนวน 1 พอร์ต
- ชุดวงจรดาวน์โหลดแบบแรงดันต่ำ (Low Voltage Programming)
- ขั้วต่อแรงดันไฟ VCC และ GND

โครงสร้างบอร์ด ET-BASE PIC40



ภาพที่ 3.9 โครงสร้างบอร์ด ET-BASE PIC40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายรายละเอียดตามหมายเลขต่างๆ ดังนี้

หมายเลข 1 ขั้วต่อแหล่งจ่ายไฟ

หมายเลข 2 ไมโครคอนโทรเลอร์

หมายเลข 3 เป็นพอร์ตของขาสัญญาณ I/O PIC MCU PORT-RB

หมายเลข 4 เป็นพอร์ตของขาสัญญาณ I/O PIC MCU PORT-RD

หมายเลข 5 เป็นพอร์ตของขาสัญญาณ I/O PIC MCU PORT-RC

หมายเลข 6 เป็นพอร์ตของขาสัญญาณ I/O PIC MCU PORT-RA

หมายเลข 7 พอร์ต RS232

หมายเลข 8 ขาสัญญาณ PORTE

หมายเลข 9 จัมพ์เปอร์เลือกการใช้งาน

หมายเลข 10 พอร์ตสัญญาณสำหรับดาวน์โหลดโปรแกรม

หมายเลข 11 LED แสดงสถานะของแหล่งจ่ายไฟภายในบอร์ด

หมายเลข 12 สวิตช์ RESET โปรแกรม

หมายเลข 13 ขั้วต่อแหล่งจ่ายไฟ สำหรับใช้ร่วมกับบอร์ด ET - BASIC I/O

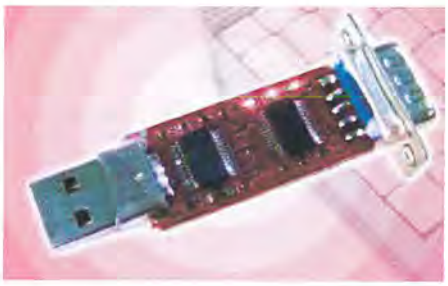
3.2.6 ET-USB/RS232 MINI

ET-USB/RS232 MINI เป็นบอร์ดในรูปแบบ SMD เปลี่ยนสัญญาณจาก RS-232 PORT เป็น USB PORT ขนาดเล็กใช้ต่อเข้าโดยตรงกับขั้ว USB ของคอมพิวเตอร์

คุณสมบัติของ ET-USB/RS232 MINI

- รองรับมาตรฐาน USB 1.1 และ USB 2.0
- อัตราการรับส่งข้อมูล 300 bps ถึง 128 Kbps
- ใช้ไฟเลี้ยงจาก USB Port โดยตรงไม่ต้องต่อเพิ่มภายนอก
- USB Connector แบบ TYPE A, RS232 Connector แบบ DP9 ตัวผู้
- แสดงสถานการณ์ทำงานด้วย LED 3 สี คือ การรับ (RX) สีเขียว, การส่ง (TX) สีเหลือง และ Power (PWR) สีแดง
- Driver รองรับ Windows 98SE/ME/2000/XP
- ขนาดกว้าง 3 cm x ยาว 6.8 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.10 ET-USB/RS232 MINI

3.2.7 กล้องเว็บแคม(Webcam)

กล้องเว็บแคมเป็นอุปกรณ์ที่นำข้อมูลเข้ามาสู่การประมวลผลภาพ โดยในงานวิจัยนี้ได้เลือกใช้กล้องเว็บแคม เนื่องจากสามารถหาซื้อง่าย ราคาถูก และง่ายต่อการติดต่อกับคอมพิวเตอร์ เพียงแค่ทำการติดตั้งไดรฟ์เวอร์ (Driver) ที่ให้มากับตัวกล้อง จากนั้นก็ทำการติดต่อตัวกล้องกับคอมพิวเตอร์ผ่านทางพอร์ต USB ตัวอย่างของกล้องเว็บแคมดังแสดงรูปที่ 3.11

ภาพที่ 3.11 กล้องเว็บแคม

คุณสมบัติของกล้องที่นำมาใช้ในการวิจัย

- ขนาดของภาพที่ใช้ 640x480 พิกเซล
- Frame Rate: 10 เฟรมต่อวินาที
- เลนส์: f=3.8 mm
- มุมมองของกล้อง 60 องศา
- ระยะจับภาพ 50 cm ถึง infinity
- การเชื่อมต่อผ่านทางพอร์ต USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 ET-SWITCHING ADAPTER

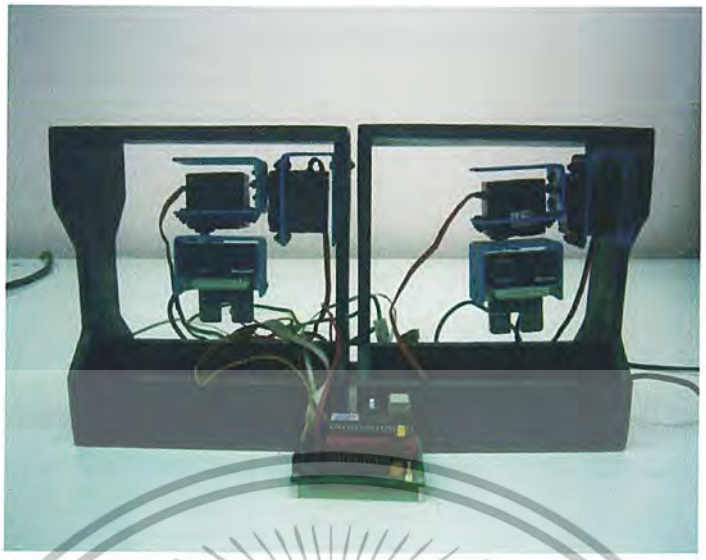
ET-SWITCHING ADAPTER เป็น Power Supply (แหล่งจ่ายพลังงาน) ในแบบของ Switching โดยเลือกใช้แบบขั้วรุ่น 5V 1.2A TYPE H จะเป็นแบบ 2.54 mm. Housing Type ใช้กับบอร์ดต่าง ๆ ในรุ่น ET-BASE ที่ใช้ Power Supply 5 VDC ได้

ภาพที่ 3.12 ET-SWITCHING ADAPTER

3.2.9 หลักการออกแบบชุดขาตั้งกล้อง

เนื่องจากในการทำโครงงานการศึกษาติดตามวัตถุนี้จะต้องมีการติดตามวัตถุใน 4 ทิศทาง คือตามแนวแกน x และ y หรือการเคลื่อนที่ไปทางซ้ายมือ ขวามือ ขึ้นและลง ทำให้ต้องใช้เซอร์โวมอเตอร์ในการควบคุม 2 ตัวคือตัวหนึ่งควบคุมการติดตามวัตถุในแนวแกน x และอีกตัวหนึ่งควบคุมการติดตามวัตถุในแนวแกน y ดังนั้นการออกแบบชุดขาตั้งกล้อง จึงต้องออกแบบให้เซอร์โวมอเตอร์ทั้ง 2 ตัว สามารถทำงานได้สัมพันธ์กันในขณะที่ทำการติดตามการเคลื่อนที่ของวัตถุ ซึ่งในการออกแบบชุดขาตั้งกล้องนั้น ผู้ดำเนินโครงงานมีการออกแบบอย่างง่ายๆ โดยการออกแบบนั้นจะทำการออกแบบให้เซอร์โวมอเตอร์ ในแนวแกน y ยึดติดอยู่บนกรอบ และใช้หมุนกล้องในให้เคลื่อนที่ขึ้นและลง สำหรับเซอร์โวมอเตอร์แกน x ถูกใช้เพื่อให้กล้องหมุน ไปทางซ้ายมือและขวามือ ซึ่งเซอร์โวมอเตอร์แกน x จะถูกยึดติดกับเฟืองของเซอร์โวมอเตอร์แกน y อีกทีหนึ่ง ชุดขาตั้งกล้องเพื่อให้กล้องหมุนในแนวแกน x และ y แสดงดังรูปที่ 3.12 อย่างไรก็ตามใน โครงงานนี้ เราได้สร้างชุดขาตั้งกล้องจำนวน 2 ชุด ดังนั้นจึงมีเซอร์โวมอเตอร์ที่ต้องคบคุมจำนวนทั้งหมด 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.13 ชุดขาตั้งกล้อง

3.3 การติดต่อสื่อสารทางพอร์ตอนุกรม (RS-232)

พอร์ตอนุกรม RS-232 จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 ขาและ 25 ขา และเราเรียกกันว่า COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิพ UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันที่ใช้กันมากที่สุดคือเบอร์ 16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิพ UART นี้จะทำหน้าที่ในการรับส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล (Data Transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่ากระบวนการ Serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต START, STOP และ PARITY เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็มหรือพอร์ตอนุกรม (Baud Rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

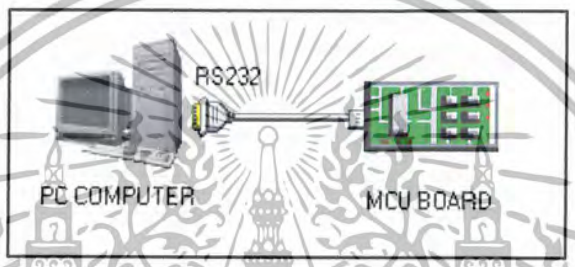
การรับข้อมูล (Data Receiver)

- รับตัวอักษรจากอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตเป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟส

การใช้งานพอร์ตอนุกรม



ภาพที่ 3.14 การเชื่อมต่อพอร์ตอนุกรมเข้ากับบอร์ดและคอมพิวเตอร์

การสื่อสารแบบอนุกรมนับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์และจอภาพของ PC เป็นอินพุตและเอาต์พุต ในการติดต่อหรือควบคุมไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้น คือ

- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- สาย GND

(ก) พอร์ตอนุกรมของ PC DB9 ตัวผู้ (Male)

(ข) พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย (Female)

ภาพที่ 3.15 พอร์ตอนุกรม RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การควบคุมเซอร์โวมอเตอร์โดยผ่านโปรแกรม MATLAB

เนื่องจากคอมพิวเตอร์ถูกใช้เพื่อการติดต่อกับกล้อง และการประมวลผลภาพเพื่อติดตามคนที่อยู่ในภาพ ดังนั้นเพื่อเป็นการลดภาระของคอมพิวเตอร์จึงได้มีการนำไมโครคอนโทรลเลอร์ PIC เข้ามาช่วยในการควบคุมเซอร์โวมอเตอร์ให้หมุนไปตามการควบคุมโดยผ่านทางโปรแกรม MATLAB

ภาพที่ 3.16 การทำงานของโปรแกรมควบคุมไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ชนิดข้อมูล

ในการใช้งานตัวแปร สิ่งสำคัญที่จะต้องคำนึงถึงคือข้อมูลที่เก็บอยู่ภายในตัวแปรนั้น ข้อมูลแต่ละชนิดมีคุณสมบัติแตกต่างกันไป เช่น เป็นเลขจำนวนเต็ม เป็นเลขจำนวนจริง เป็นตัวอักษร เป็นต้น

ตารางที่ 3.1 ชนิดข้อมูลภาษา C สำหรับ CCS C คอมไพเลอร์

ชนิดข้อมูล	ขนาด	ไม่คิดเครื่องหมา (unsigned)	คิดเครื่องหมาย (signed)
int1	1 บิต (ตัวเลข)	0 or 1	-
int8	8 บิต (ตัวเลข)	0 to 255	-128 to 127
int16	16 บิต (ตัวเลข)	0 to 65,535	-32768 to 32768
int32	32 บิต (ตัวเลข)	0 to 4,294,967,295	-2147483648 to 2147483648
float32	32 บิต (ทศนิยม)	-1.5×10^{45} to 3.4×10^{38}	

3.4.2 ฟังก์ชันที่ใช้ในการเขียน Source Code บนโปรแกรม PIC C Compiler

ฟังก์ชัน OUTPUT_LOW( ) : ฟังก์ชันนี้ ใช้ควบคุมขาพอร์ตแบบบิต โดยกำหนดให้ขาพอร์ตมีสถานะ เป็นลอจิก "0" (Ground State)

ฟังก์ชัน OUTPUT_HIGH( ) : ฟังก์ชันนี้ใช้ควบคุมขาพอร์ตแบบบิต โดยกำหนดให้ขาพอร์ตมีสถานะเป็นลอจิก "1" (High State)

ฟังก์ชัน SET_TRIS_X( ) : ฟังก์ชันนี้ใช้กำหนดทิศทางพอร์ตว่าเป็นอินพุตหรือเอาต์พุตพอร์ต โดยกำหนดในรีจีสเตอร์ TRI-State (ติดตั้งการใช้งานพอร์ต A-E เป็นอินพุตหรือเอาต์พุตพอร์ตก่อนการใช้งานพอร์ต) โดยใช้ไดเร็คทีฟ FAST_IO เมื่อต้องการให้เข้าถึงการใช้งานพอร์ตในหน่วยความจำรีจีสเตอร์เมื่อมีการใช้งานไดเร็คทีฟ #BYTE (เมื่อต้องการเข้าถึงรีจีสเตอร์อินพุตเอาต์พุตพอร์ตโดยตรง) และจะใช้ไดเร็คทีฟ #USE STANDARD_IO (Standard I/O) เมื่อต้องการใช้งานฟังก์ชันใน CCS เพื่อเข้าถึงพอร์ตโดยอัตโนมัติ ซึ่งเป็นค่าปกติ (Default) แต่ละบิตในรีจีสเตอร์ TRI-State จะแทนแต่ละขาของพอร์ต เมื่อกำหนดเป็น "0" ขาพอร์ตนั้นจะเป็นเอาต์พุต รูปแบบการใช้งาน : set tris a (value); สำหรับพอร์ต A

ฟังก์ชัน ENABLE_INTERRUPTS( ) : กำหนดให้มีการอินเตอร์รัพท์เกิดขึ้นได้ (เปิดใช้งานอินเตอร์รัพท์)

ฟังก์ชัน DISBLE_INTERRUPTS( ) : กำหนดให้ไม่มีการอินเตอร์รัพท์เกิดขึ้นได้ (ปิดใช้งานอินเตอร์รัพท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน #INT_TIMER1 : การนับในโหมด Timer 1 จะมีได้ถึง 16 บิต โดยการนำเอาข้อมูลในรีจิสเตอร์ TL0 กับ TH0 หรือรีจิสเตอร์ TL1 กับ TH1 ที่มีขนาดตัวละ 8 บิตมารวมกันเพื่อให้ได้ รีจิสเตอร์ขนาด 16 บิต การใช้งานจะให้นับไป จนครบรอบโดยให้ค่าของการนับเปลี่ยนจาก FFFFH เป็น 0000H หรือให้นับเกินค่า 65,536 ก็จะทำให้บิต TFx ซึ่งจะแสดงสถานะค่านับเกิน (Over Flow Flag) ในรีจิสเตอร์ TCON ทำการเซตบิตเป็นลอจิก "1" ไมโครคอนโทรลเลอร์ก็จะรู้ว่ามีการนับครบกำหนดแล้ว แสดงการทำงานของ Timer/Counter ในโหมด Timer1 (16 บิต)

3.4.3 หลักการทำงานของโปรแกรมควบคุมเซอร์โวมอเตอร์

เนื่องจากโครงงานนี้ต้องการควบคุมเซอร์โวมอเตอร์ 4 ตัวพร้อมๆกัน ซึ่งโดยทั่วไปเซอร์โวมอเตอร์ถูกควบคุมตำแหน่งการหมุน โดยค่าสัญญาณความกว้างพัลส์ อย่างไรก็ตามในโครงงานนี้ควบคุมการหมุนของเซอร์โวมอเตอร์จะใช้วิธีการตั้งค่า Timer แทน ซึ่งเซอร์โวมอเตอร์จะหมุนไปทางซ้ายมือสุดเมื่อค่า Timer มีค่าเท่ากับ 16 และทางขวามือสุดเมื่อค่า Timer มีค่าเท่ากับ 32 ดังนั้นเมื่อต้องการสั่งให้เซอร์โวมอเตอร์ทั้ง 4 ตัวหมุน ต้องส่งค่า Timer 4 ค่า เช่น 16161616 เป็นต้น โดยค่า 16 สองบิตทางขวามือเป็นค่า Timer ของเซอร์โวมอเตอร์ตัวที่หนึ่ง ในทำนองเดียวกันทุกๆ 2 บิตถัดไปเป็นการกำหนดค่า Timer ของเซอร์โวมอเตอร์ตัวที่สอง สามและสี่ ตามลำดับ การติดต่อระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ PIC เพื่อการควบคุมเซอร์โวมอเตอร์เมื่อไมโครคอนโทรลเลอร์ PIC รับข้อมูลจากคอมพิวเตอร์ผ่านทาง Port RS232 จะเก็บข้อมูลไว้ในรูปของตัวแปรสตริง แล้วแปลงให้เป็นตัวเลขจำนวนเต็ม (Integer) ชนิด 32 บิต จากนั้นแบ่งข้อมูลที่รับเข้ามาออกเป็น 4 ส่วน สำหรับเซอร์โวมอเตอร์ 4 ตัว โดยในขั้นตอนแรกแบ่งข้อมูล 4 ตัวแรก ด้วยการหารข้อมูลที่รับเข้ามาแบบปกติด้วยค่า 10000 และข้อมูล 4 ตัวหลังด้วยการหารแบบ Modulo (%) ด้วยค่า 10000 เช่นกัน เมื่อแยกข้อมูลออกเป็น 2 ชุดแล้ว ข้อมูล 4 ตัวแรกและ 4 ตัวหลังจะถูกแยกออกเป็นอย่างละ 2 ชุดอีก ด้วยวิธีการหารปกติด้วยค่า 100 และหารแบบ Modulo (%) ด้วยค่า 100 เช่นกัน เมื่อเราทำการแยกข้อมูลออกเป็น 4 ชุด สำหรับเซอร์โวมอเตอร์ 4 ตัวแล้ว หลังจากนั้นโปรแกรมจะพิจารณาเงื่อนไขของเซอร์โวมอเตอร์แต่ละตัวเพื่อให้ Timer1 สร้างสัญญาณความกว้างพัลส์ โดยใน Timer1 จะกำหนด Cycle Servo1 เท่ากับ 325 ซึ่งค่า Cycle Servo1 จะลดค่าลงทีละหนึ่ง เมื่อเราทราบค่า Timer ของเซอร์โวมอเตอร์แต่ละตัวดังที่อธิบายข้างต้นแล้ว ไมโครคอนโทรลเลอร์ PIC จะส่งค่าโลจิก "1" (output_hight(servo_pin1)) ไปที่เซอร์โวมอเตอร์ตามขาพอร์ตที่กำหนด โดยค่า Timer ของเซอร์โวมอเตอร์ตัวนั้นจะลดค่าลงทีหนึ่งจนกระทั่งเป็นศูนย์ จากนั้นไมโครคอนโทรลเลอร์ PIC จะส่งค่าโลจิก "0" (output_low(servo_pin1)) ไปที่เซอร์โวมอเตอร์ เป็นอันสิ้นสุดไซเคิลการทำงานของเซอร์โวมอเตอร์ สำหรับการควบคุมเซอร์โวมอเตอร์ตัวอื่นๆจะมีลักษณะเช่นเดียวกับที่อธิบายไว้ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 Source Code ควบคุมเซอร์โวมอเตอร์

```
1. #include <18F4620.h>                    // กำหนดเบอร์ไมโครคอนโทรลเลอร์ PIC ใช้งาน
2. #include <stdlib.h>                     // กำหนดใช้ฟังก์ชันพื้นฐาน
3. #fuses HS,NOLVP,NOWDT,NOPROTECT
                                           // กำหนดการทำงานหลักของไมโครคอนโทรลเลอร์
4. #define TX PIN_C6                      // กำหนดให้ขาที่ 6 เป็นการส่งข้อมูล
5. #define RX PIN_C7                      // กำหนดให้ขาที่ 7 เป็นการรับข้อมูล
6. #use   delay (clock = 10000000)         // Crytal = 10MHz
7. #use   rs232(baud = 9600, xmit = TX, rcv = RX)   // กำหนดการสื่อสารข้อมูลผ่าน RS-232 ที่
                                           อัตราบอร์ดเรต 9600
8. #use   fast_io(A)                       // เข้าถึงรีจิสเตอร์พอร์ต A โดยตรง
9. #use   fast_io(B)                       // เข้าถึงรีจิสเตอร์พอร์ต B โดยตรง
10. #use  fast_io(C)                       // เข้าถึงรีจิสเตอร์พอร์ต C โดยตรง
11. #use  fast_io(D)                       // เข้าถึงรีจิสเตอร์พอร์ต D โดยตรง
12. #define servo_pin1   pin_A0           // ประกาศใช้งานพอร์ตA ใช้ขา A0
13. #define servo_pin2   pin_A1           // ประกาศใช้งานพอร์ตA ใช้ขา A1
14. #define servo_pin3   pin_A2           // ประกาศใช้งานพอร์ตA ใช้ขา A2
15. #define servo_pin4   pin_A3           // ประกาศใช้งานพอร์ตA ใช้ขา A3
//---------- กำหนดการทำงานของไมโครคอนโทรลเลอร์ --------//
16. char McE[10];                          // ให้ McE มีสมาชิก 10 ตัว เป็นแบบ Character
17. int32 Pass=0,Pass1=0,Pass2=0,Pass12=0,Pass11=0,Pass22=0,Pass21=0;
                                           // ประกาศตัวแปรเป็นแบบ interger32 บิต
18. int32 CycleServo1,PulseServo11,PulseServo12,PulseServo21,PulseServo22;
                                           // ประกาศตัวแปรเป็นแบบ interger32 บิต
19. #INT_RDA                               // ใช้อินเตอร์รับเมื่อผ่านพอร์ตอนุกรม
//---------- ประกาศตัวแปรและการใช้งานอินเตอร์รัพท์จากภายนอก---------//
20. void IntRDA_isr(){                     // ฟังก์ชันอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม
21. gets(McE);                             // รับค่า McE จากคีย์บอร์ด
22. Pass = atoi32(McE);                    // แปลงค่าที่ McE เป็นแบบ Integer ให้เท่ากับ Pass
23. Pass1 = Pass / 10000;                  // นำค่าที่รับมาหาร 10000 แบบปัดเศษมาเก็บที่ Pass1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
24. Pass2 = Pass % 10000;          // นำค่าที่รับมาหาร 10000 แบบต้องการเศษมาเก็บที่ Pass2
25. Pass11 = Pass1 / 100;          // นำค่าที่รับมาหาร 100 แบบปัดเศษมาเก็บที่ Pass11
26. Pass12 = Pass1 % 100;          // นำค่าที่รับมาหาร 100 แบบต้องการเศษมาเก็บที่ Pass12
27. Pass21 = Pass2 / 100;          // นำค่าที่รับมาหาร 100 แบบปัดเศษมาเก็บที่ Pass21
28. Pass22 = Pass2 % 100;          // นำค่าที่รับมาหาร 100 แบบต้องการเศษมาเก็บที่ Pass22
30. }
//---------- ใช้งานอินเตอร์รัพท์จากภายนอกและรับข้อมูล ---------//
32. #INT_TIMER1                    // ประกาศใช้งาน Timer 1
33. void IntTimer1_isr( )          // ใช้อินเตอร์รัพท์จากภายนอก
34.{
35.  if(CycleServo1!=0)           // ถ้า CycleServo1 ไม่เท่ากับศูนย์
36.  {
37.    CycleServo1--;             // CycleServo1 ลงลงทีละหนึ่ง
38.     if(PulseServo11!=0)       // ถ้า CycleServo1 ไม่เท่ากับศูนย์
39.     {
40.     output_high(servo_pin1); // ที่ขาพอร์ต A0 ให้มีสถานะเป็นลอจิก 1
41.       PulseServo11--;          // PulseServo11 ลดลงทีละหนึ่ง
42.     }
43.     if(PulseServo11==0)       // ถ้า PulseServo11 เท่ากับศูนย์
44.     {
45.       output_low(servo_pin1); // ที่ขาพอร์ต A0 ให้มีสถานะเป็นลอจิก 0
46.     }
//---------- เงื่อนไขการทำงานของเซอร์มอเตอร์ตัวที่ 1 ---------//
47.      if(PulseServo12!=0)       // ถ้า CycleServo12 ไม่เท่ากับศูนย์
48.     {
49.       output_high(servo_pin2);  // ที่ขาพอร์ต A1 ให้มีสถานะเป็นลอจิก 1
50.       PulseServo12--;           // PulseServo11 ลดลงทีละหนึ่ง
51.     }
55.     if(PulseServo12==0)       // ถ้า PulseServo12 เท่ากับศูนย์
56.     {
57.       output_low(servo_pin2); // ที่ขาพอร์ต A1 ให้มีสถานะเป็นลอจิก 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
58.     }
//---------- เงื่อนไขการทำงานของเซอร์มอเตอร์ตัวที่ 2 ---------//
59.     if(PulseServo21!=0)         // ถ้า CycleServo21 ไม่เท่ากับศูนย์
60.     {
61.       output_high(servo_pin3); // ที่ขาพอร์ต A2 ให้มีสถานะเป็นลอจิก 1
62.       PulseServo21--;           // PulseServo11 ลดลงทีละหนึ่ง
63.     }
64.     if(PulseServo21==0)         // ถ้า PulseServo21 เท่ากับศูนย์
65.     {
66.       output_low(servo_pin3); // ที่ขาพอร์ต A2 ให้มีสถานะเป็นลอจิก 0
67.     }
//---------- เงื่อนไขการทำงานของเซอร์มอเตอร์ตัวที่ 3 ---------//
68.     if(PulseServo22!=0)         // ถ้า CycleServo21 ไม่เท่ากับศูนย์
69.     {
70.       output_high(servo_pin4); // ที่ขาพอร์ต A3 ให้มีสถานะเป็นลอจิก 1
71.       PulseServo22--;           // PulseServo22 ลดลงทีละหนึ่ง
72.     }
73.     if(PulseServo22==0)         // ถ้า PulseServo22 เท่ากับศูนย์
74.     {
75.       output_low(servo_pin4); // ที่ขาพอร์ต A3 ให้มีสถานะเป็นลอจิก 0
76.     }
//---------- เงื่อนไขการทำงานของเซอร์มอเตอร์ตัวที่ 4 ---------//
77.   }
78. if(CycleServo1==0)             // ถ้า CycleServo1 เท่ากับศูนย์
79. {
80.   CycleServo1  = 325;           // ให้ CycleServo1เท่ากับ 325
81.   PulseServo11 = Pass11;        // ให้ PulseServo11 เท่ากับ Pass11
82.   PulseServo12 = Pass12;        // ให้ PulseServo12 เท่ากับ Pass12
83.   PulseServo21 = Pass21;        // ให้ PulseServo21 เท่ากับ Pass21
84.   PulseServo22 = Pass22;        // ให้ PulseServo22 เท่ากับ Pass22
85. }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

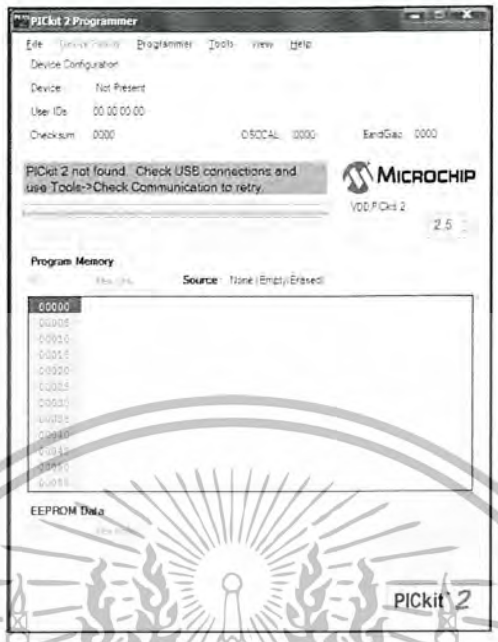
```
86.  set_timer1(65451);              // interrupt 34uS
87.  }
88.  void main(){                    // ฟังก์ชันหลัก
89.  set_tris_c(0b11111111);         // ปิดใช้งานขาพอร์ต C ทุกขา
90.  set_tris_a(0b10000000);         // เปิดใช้งานขาพอร์ต A ทุกขา
91.  set_tris_b(0b11111111);         // ปิดใช้งานขาพอร์ต B ทุกขา
92.  set_tris_d(0b11111111);         // เปิดใช้งานขาพอร์ต D ทุกขา
93.  enable_interrupts(GLOBAL);      // เปิดใช้งานอินเตอร์รัพท์โดยรวม
94.  enable_interrupts(INT_TIMER1); // เปิดใช้งานอินเตอร์รัพท์ของ Timer 1
95.  enable_interrupts(INT_RDA);     // เปิดใช้งานอินเตอร์รัพท์ผ่านทาง RS-232
96.  setup_timer_1(T1_INTERNAL | T1_DIV_BY_1); // ตั้งค่า Timer1 หารด้วย 1
97.  set_timer1(65451);              // interrupt 34uS
98.  Pass = 0;                       // ให้ Pass เท่ากับศูนย์
99.   while(1){                      // ตรวจสอบเงื่อนไข
100. if(Pass!=0){                   // ถ้า Pass ไม่เท่ากับศูนย์
101. PulseServo11 = Pass11;         // ให้ PulseServo11 เท่ากับ Pass11
102. PulseServo12 = Pass12;         // ให้ PulseServo12 เท่ากับ Pass12
103. PulseServo21 = Pass21;         // ให้ PulseServo21 เท่ากับ Pass21
104. PulseServo22 = Pass22;         // ให้ PulseServo22 เท่ากับ Pass22
//---------- ฟังก์ชันหลักเปิดใช้งานอินเตอร์รัพท์และสร้างสัญญาณพัลส์จาก Timer1---------//
105.   }
106.  }
107. }
//---------- จบการทำงานของโปรแกรม---------//
```

3.4.5 ขั้นตอนการเบิร์น hex. file ลงไมโครคอนโทรลเลอร์

หลังจากที่เขียน Source Code เสร็จแล้วให้โปรแกรม PIC CCS ทำการคอมไพล์โปรแกรม จะได้ ไฟล์.hex ออกมาและนำไฟล์ที่ได้มา Burn Program ลงในไมโครคอนโทรลเลอร์ PICโดยใช้โปรแกรม PICkit 2 V2.61 ซึ่งมีขั้นตอนการ Burn Program ดังนี้

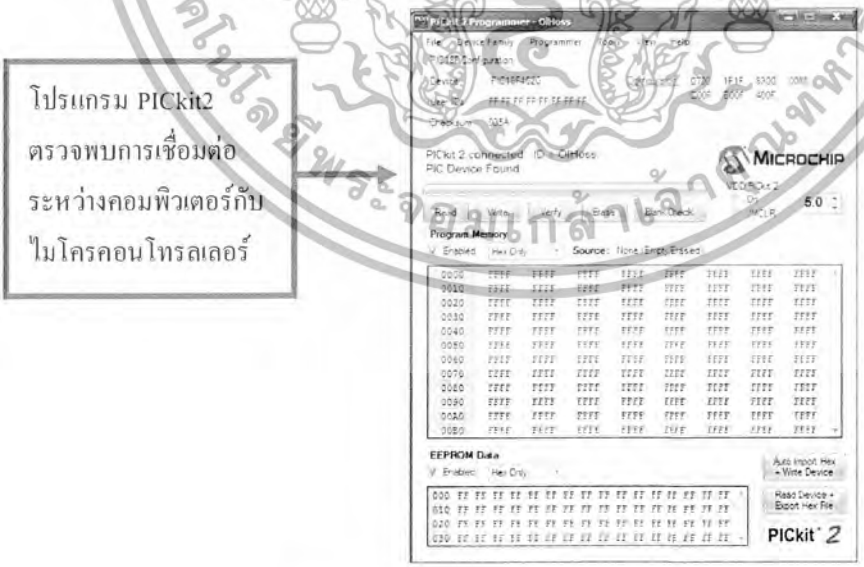
1. เปิดโปรแกรม PICkit 2 V2.61 โดยไป Desktop ดับเบิลคลิกที่ จะปรากฏหน้าต่างโปรแกรมดังภาพที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.17 หน้าต่างโปรแกรม PICkit 2 V2.61

2. โปรแกรม PICkit2 จะตรวจอุปกรณ์โดยอัตโนมัติ (เบอร์ไมโครคอนโทรล- เลอร์ PIC) ในที่นี้ใช้ไมโครคอนโทรลเลอร์ PIC เบอร์18F4620 ซึ่งเมื่อตรวจพบจะแสดงผลที่หน้าต่างของโปรแกรม PICkit2 ว่า PIC Device Found

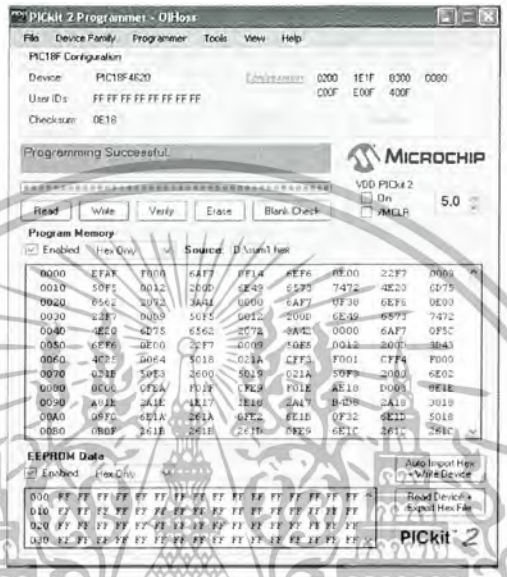


ภาพที่ 3.18 โปรแกรม PICkit2 V2.61 ตรวจพบ PIC 18F4620

3. จากนั้นนำ ไฟล์. hex ที่ได้จากการคอมไพล์โดยโปรแกรม CCS PIC C ไป Burn ลงในไมโครคอนโทรลเลอร์ PIC โดยไปที่เมนู File เลือก Import hex แล้วทำการเลือก hex file ที่ต้องการใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างโปรแกรม PICkit2 จะบอกว่า Hex file successfully imported จากนั้นทำการคลิกที่ปุ่ม Write โปรแกรมจะทำการลบและเขียน hex file โดยอัตโนมัติ เมื่อโปรแกรม PICkit2 ลง hex file ในไมโครคอนโทรลเลอร์เสร็จแล้ว จะปรากฏคำว่า Programming Successful ก็เป็นอันเสร็จสิ้นขั้นตอนการลง hex file บนไมโครคอนโทรลเลอร์ PIC ดังภาพที่ 3.19

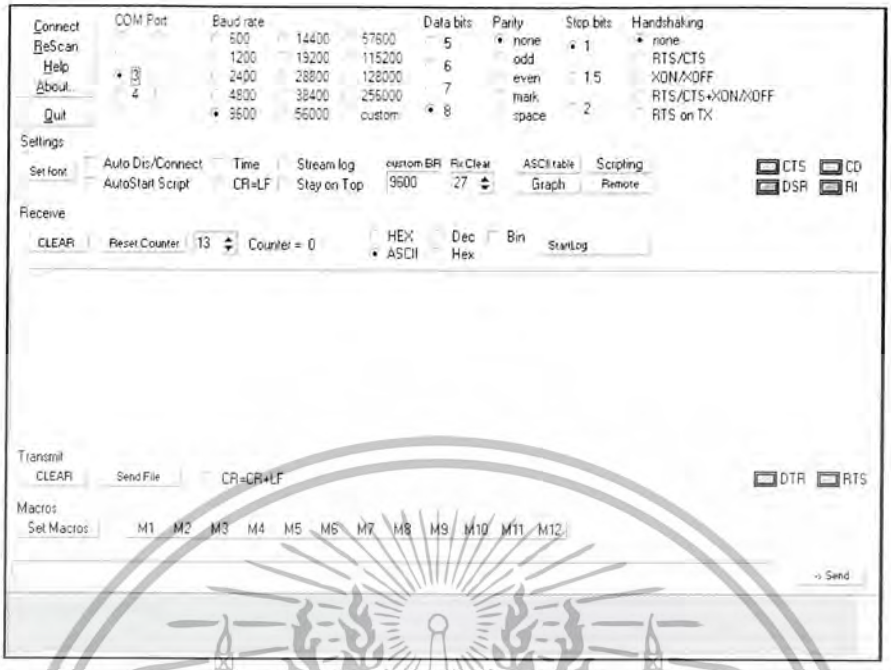


ภาพที่ 3.19 หน้าต่างโปรแกรม PICkit2 V2.61

3.4.6 โปรแกรม Hyper terminal ใช้ติดต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์

การสื่อสารทางพอร์ตอนุกรมเป็นการสื่อสารแบบ Full Duplex คือสามารถรับและส่งได้ในเวลาเดียวกัน ซึ่งจะมีขา TXD ใช้ในการส่งและ RXD ใช้ในการรับ การสื่อสารทางพอร์ตอนุกรมโดยทั่วไปจะใช้ตัว Driver/Receiver มาต่อกับขา RXD/TXD อีกครั้ง เพื่อเพิ่มระยะทางในการสื่อสาร สำหรับการติดต่อกับคอมพิวเตอร์ เราจะต้องใช้ตัว Diver/Receiver เพื่อแปลงสัญญาณลอจิกของไมโครคอนโทรลเลอร์ PIC ให้เป็นมาตรฐาน RS-232 เบอร์ที่นิยมใช้กันมากจะเป็น MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



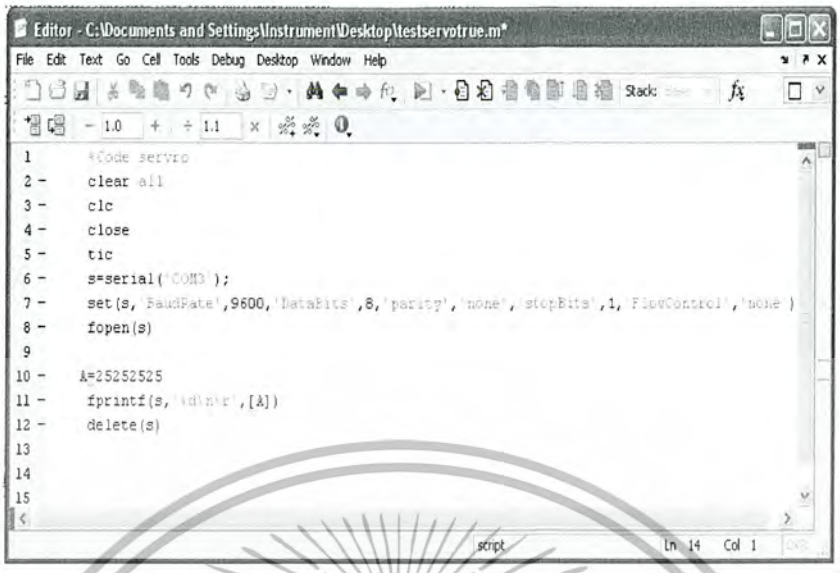
ภาพที่ 3.20 หน้าต่างโปรแกรม Hyper terminal

3.4.7 โปรแกรมที่ใช้ในการควบคุมเซอร์โวมอเตอร์ผ่านโปรแกรม MATLAB

โปรแกรมนี้จะเป็นการสั่งให้เซอร์โวมอเตอร์หมุนไปทางด้านซ้ายมือ ขวามือ ขึ้นข้างบน และลงข้างล่างโดยใช้โปรแกรม MATLAB ส่งค่า Timer ของเซอร์โวมอเตอร์ทั้ง 4 ตัวไปยังไมโครคอนโทรลเลอร์ เพื่อให้เซอร์โวมอเตอร์หมุนตามคำสั่งที่ต้องการ ในขั้นตอนแรกจะกำหนดการสื่อสารแบบอนุกรมที่พอร์ต COM3 จากนั้นกำหนดค่า BaudRate, DataBits, stopBits เป็นต้น แล้วเปิดการทำงานของพอร์ตอนุกรม ส่งค่า Timer ของเซอร์โวมอเตอร์ทั้ง 4 ตัวพร้อมกับด้วยคำสั่ง fprintf และสิ้นสุดการทำงานด้วยคำสั่ง delete

```
1. clear all;                          % ล้างตัวแปรทั้งหมดในหน่วยความจำ
2. clc;                                % ล้างหน้าต่างการทำงาน
3. s = serial('COM3')                  % เปิดการสื่อสารแบบอนุกรมที่พอร์ต
                                         คอมพิวเตอร์ 3
4. set(s,'BaudRate',9600,'DataBits',8,'parity','none','stopBits',1,'FlowControl','none')
                                       % กำหนดค่าให้ตัวแปร s
5. fopen(s)                            % เปิดการทำงานของพอร์ตอนุกรม
7. fprintf(s,'%d\n\r',[25252525]);     % ส่งตัวเลข 25252525 ให้มอเตอร์ทั้ง 4 ตัว
8. delete(s)                           % ลบตัวแปร s เมื่อจบการทำงาน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.21 หน้าต่างโปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

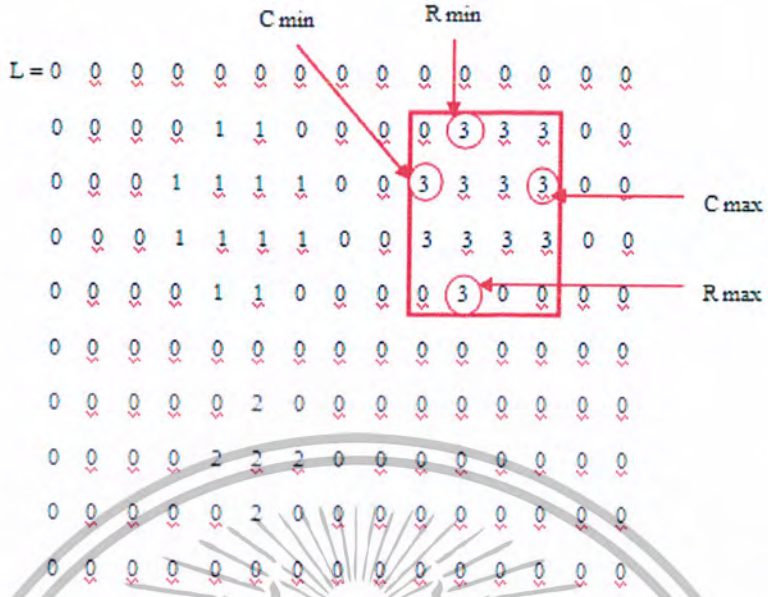
บทที่ 4
การประมวลผลภาพเพื่อการติดตามคนที่กำลังเคลื่อนที่ในภาพ

ในส่วนของซอฟต์แวร์การติดตามคนที่กำลังเคลื่อนที่ในภาพ มีซอฟต์แวร์ที่เกี่ยวข้องในการประมวลผลภาพโดยใช้โปรแกรม MATLAB เริ่มต้นจากโปรแกรมการตีกรอบล้อมรอบวัตถุที่อยู่ในภาพ จากนั้นทำการตรวจหาสีผิวเพื่อระบุว่าเป็นคนได้ถูกต้องมากยิ่งขึ้น และตรวจหาปลอกแขนซึ่งมีด้วยกัน 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน เพื่อกำหนดว่าจะเลือกติดตามคนที่สวมปลอกแขนสีใด เมื่อระบุคนที่ต้องการติดตามได้แล้ว ต่อมาเป็นการคาดการณ์เส้นทางการเคลื่อนที่ของคนที่อยู่ในภาพ และการหาเส้นทางการเคลื่อนที่ของคนที่อยู่ในภาพ ซึ่งรายละเอียดทฤษฎีต่าง ๆ มีดังต่อไปนี้

4.1 การเขียนโปรแกรมตีกรอบล้อมรอบวัตถุโดยการใช้ bwlabel

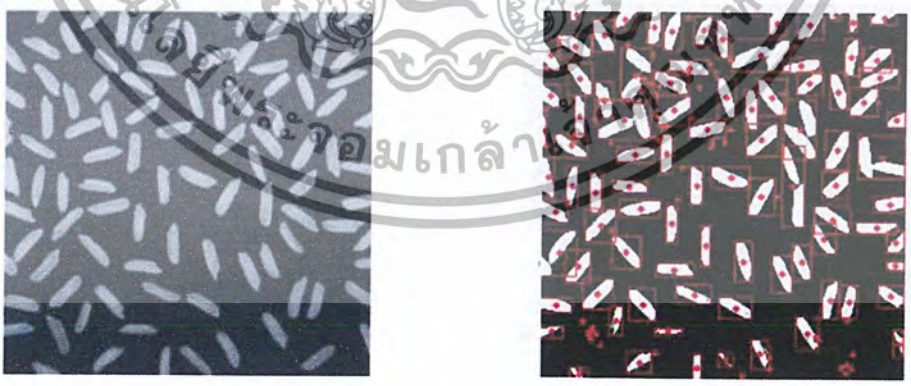
ในการติดตามคนที่กำลังเคลื่อนที่ในภาพ เริ่มจากการตีกรอบล้อมรอบคนทุกคนที่อยู่ในภาพ แต่เมื่อจำนวนคนในภาพเพิ่มมากขึ้น โปรแกรมจะไม่สามารถตีกรอบคนทุกคนในภาพได้ จึงต้องทำการเพิ่มเงื่อนไขโปรแกรมเข้าไปด้วย ดังนั้นเพื่อความสะดวกในการใช้งาน จึงใช้คำสั่งจาก Tool ของ MATLAB คือคำสั่ง bwlabel ที่ได้กล่าวในทฤษฎีพื้นฐานเพื่อช่วยในการ label ให้กับวัตถุ โดยกำหนดหมายเลขให้กับวัตถุแต่ละวัตถุที่ไม่อยู่ติดกันให้มีหมายเลขต่างกัน แล้วใช้คำสั่ง find เพื่อเลือกวัตถุที่ต้องการตีกรอบ เช่น ถ้ามีคน 5 คนอยู่ภาพ สามารถเลือกได้ว่าจะตีกรอบคนที่ 1 2 3 4 หรือ 5 ยกตัวอย่างถ้าต้องการตีกรอบล้อมรอบวัตถุที่ 3 จะกำหนดให้ [r,c] = find(L==3) ซึ่ง r คือ Row และ c คือ Column โดยโปรแกรมจะตรวจหาจุด Co-ordinate ของแถวแรกที่พบกำหนดให้เป็น Row Minimum (R min) และแถวสุดท้ายเป็น Row Maximum (R max) รวมทั้งหาจุด Co-ordinate ของหลักแรกที่พบกำหนดให้เป็น Column Minimum (C min) และหลักสุดท้ายเป็น Column Maximum (C max) จากนั้นทำการตีกรอบตามจุด Co-ordinate ที่ได้ก็จะทำให้สามารถตีกรอบล้อมรอบวัตถุทุกวัตถุที่อยู่ในภาพได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.1 การตีกรอบล้อมรอบวัตถุโดยหาจุด Co-ordinate ของแต่ละ Row และ Column

การตีกรอบล้อมรอบวัตถุโดยใช้คำสั่ง bwlabel เป็นการระบุหมายเลขให้กับวัตถุ กรณีที่วัตถุมีหลายวัตถุในภาพ โปรแกรมจะกำหนดหมายเลขต่าง ๆที่ไม่ซ้ำกันและสามารถตีกรอบล้อมรอบแต่ละวัตถุได้ เช่น ภาพเมล็ดข้าว คำสั่งที่เลือกใช้สามารถตีกรอบเมล็ดข้าวได้ทั้งหมดดังภาพที่ 4.2 ถึงแม้ว่าเมล็ดข้าวจะวางตัวอยู่ในตำแหน่งหรือทิศทางใดก็ตาม ก็สามารถตีกรอบล้อมรอบเมล็ดข้าวได้ทุกเมล็ด

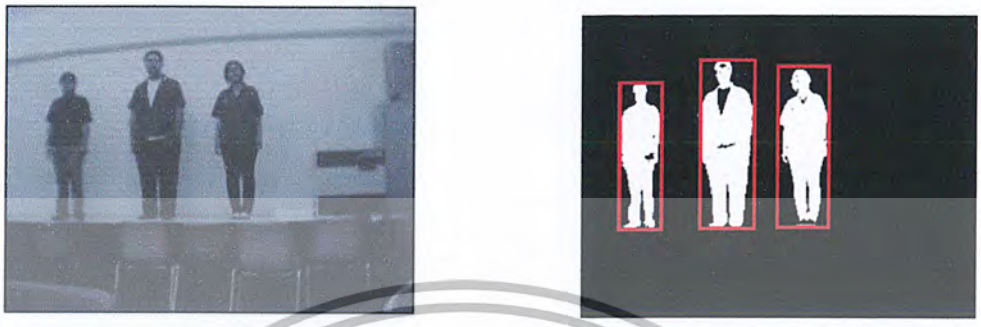


(ก) ภาพต้นฉบับ (ข) ภาพการตีกรอบล้อมรอบเมล็ดข้าว

ภาพที่ 4.2 ตัวอย่างการตีกรอบล้อมรอบเมล็ดข้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทำนองเดียวกัน การตีกรอบล้อมรอบคนหลาย ๆคนในภาพสามารถทำให้สำเร็จได้โดยใช้คำสั่ง bwlabel ดังแสดงในภาพที่ 4.3



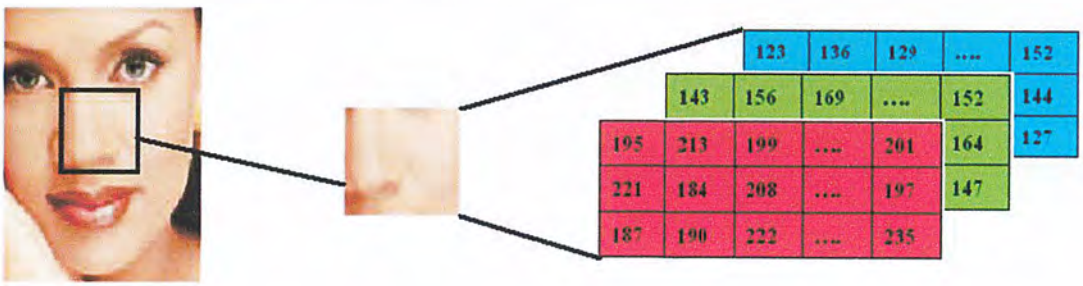
(ก) ภาพถ่ายที่ได้จากกล้อง Webcam (ข) ภาพการตีกรอบคล้อมรอบคน

ภาพที่ 4.3 ตัวอย่างการตีกรอบล้อมรอบคนในภาพ

4.2 การตรวจหาสีผิว (Skin Detection)

ในการลบภาพเมื่อภาพที่ 1 ลบกับภาพที่ 2 ซึ่งมีคนอยู่ในภาพ ผลลัพธ์ก็คือจะปรากฏคนอยู่ในภาพ หลังจากนั้นโปรแกรมก็จะทำการตีกรอบโดยใช้คำสั่ง bwlabel แต่เนื่องจากในกรณีที่คนสวมเสื้อหรือกางเกงที่มีสีใกล้เคียงกับสีพื้นหลัง โปรแกรมก็ไม่สามารถตีกรอบคนได้เต็มตัวแต่จะตีกรอบในส่วนที่เป็นศีรษะ ลำตัวส่วนบน และลำตัวส่วนล่าง ดังนั้นจึงมีแนวคิดเกี่ยวกับการนำสีผิวเข้ามาพิจารณาเพื่อใช้ประกอบการตัดสินใจ โดยถ้าพบว่ามีสีผิวบริเวณใบหน้า แขน หรือขาก็จะทำให้สามารถระบุได้ว่าเป็นคนได้ถูกต้องมากยิ่งขึ้น

เมื่อทำการศึกษาเรื่องสีผิวจากรายงานของต่างประเทศ ชื่อ Skin Detection in Luminance Images using Threshold Technique เป็นการคัดแยกสีผิวโดยลดผลกระทบจากความเข้มแสง ด้วยการทำ Normalize RGB ของภาพซึ่งเริ่มจากการตัดตัวอย่างสีผิวแล้วแยกภาพของแสงสีแดง สีเขียว และสีน้ำเงินดังภาพที่ 4.4 จากนั้นจึงทำการ Normalize ภาพด้วยสมการที่ (4.1)~(4.2)



ภาพที่ 4.4 การตัดตัวอย่างสีผิวแล้วทำการแยกภาพของแสงสีแดง สีเขียวและสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$r = \frac{R}{R+G+B} \tag{4.1}$$

และ

$$g = \frac{G}{R+G+B} \tag{4.2}$$

จากนั้นทำการหาค่าเฉลี่ยของ r และ g (Mean of r and Mean of g) และหาค่า Standard Deviation (σ_r, σ_g) เพื่อนำค่าเหล่านี้มาทำ Gaussian ดังสมการที่ (4.3)

$$f(r,g) = \frac{1}{2\pi} e^{\frac{-(r_i-\bar{r})^2}{2\sigma_r^2}} e^{\frac{-(g_i-\bar{g})^2}{2\sigma_g^2}} \tag{4.3}$$

โดยที่

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{4.4}$$

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i \tag{4.5}$$

$$\sigma_r^2 = \frac{1}{N}(r_i - \bar{r})^2 \tag{4.6}$$

$$\sigma_g^2 = \frac{1}{N}(g_i - \bar{g})^2 \tag{4.7}$$

จากสมการที่ (4.3) เมื่อทำการคูณ Gaussian ของภาพ Normalize r กับ g แล้วก็จะทำการตัดค่า Threshold โดยการปรับด้วยมือซึ่งเริ่มจากค่า 1 แล้วค่อย ๆลดระดับลงทีละ 0.1 จนกระทั่งถึง 0 เพื่อเลือกเป็นค่า Threshold ที่เหมาะสมที่สุด ซึ่งพิจารณาค่าที่เป็นสีผิวให้เป็น TP (True Positive) ส่วนค่าที่ไม่ใช่สีผิวเป็น FN (False Negative) โดยถ้านำ TP+FN≈1

(ก) ภาพต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ข) สีผิวในอุดมคติ (ค) สีผิวที่แท้จริงและส่วนที่ไม่ใช่สีผิว (ง) สีผิวที่แท้จริง

ภาพที่ 4.5 การตัดค่า Threshold โดยปรับค่าลดลงทีละ 0.1

จากภาพที่ 4.5 แสดงการตัดค่า Threshold ด้วยการปรับลดค่าลงทีละ 0.1 โดยลดลงจาก 1 จนถึง 0 จากรูปในส่วนบริเวณที่เป็นสีผิวจะแสดงด้วยสีขาวและบริเวณที่ไม่ใช่สีผิวจะแสดงด้วยสีดำ ซึ่งภาพที่ 4.5(ข) เป็นภาพสีผิวในอุดมคติ ส่วนภาพที่ 4.5(ค) เป็นภาพแสดงสีผิวภายหลังเลือกค่า Threshold ที่ดีที่สุดซึ่งจะพบว่ามีส่วนที่ไม่ใช่สีผิวปรากฏออกมาด้วย (True Positive and False Negative) ภาพที่ 4.5(ง) เป็นภาพแสดงเฉพาะส่วนที่เป็นสีผิวที่แท้จริง (True Positive)

ในการศึกษาข้อมูลจากรายงาน Skin Detection in Luminance Images using Threshold Technique ทำให้มีแนวคิดเกี่ยวกับการตัดค่า Threshold แบ่งออกเป็น 4 แนวคิดด้วยกัน ดังนี้

1. การตัดค่า Threshold แบบอัตโนมัติ
2. การตัดค่า Threshold จากการทำ Gaussian
3. การตัดค่า Threshold จากค่า r g b ที่ได้จากการ Normalize ภาพสี RGB
4. การเลือกตัวอย่างสีผิวสำหรับการตัดค่า Threshold ที่เหมาะสม

4.2.1 การตัดค่า Threshold แบบอัตโนมัติ

เนื่องจากในรายงานข้างต้นใช้วิธีการตัดค่า Threshold แบบปรับด้วยมือ (Manual) ซึ่งจะค่อย ๆปรับลดค่าลงทีละ 0.1 โดยลดลงจาก 1 จนถึง 0 แล้วเลือกค่าที่เหมาะสมที่สุดเป็นค่า Threshold ดังนั้นเพื่อความสะดวกในการนำค่า Threshold ไปใช้ได้ในทุกกรณี โดยที่ไม่ต้องทำการปรับเปลี่ยนค่าทุกครั้ง จึงมีแนวความคิดว่าการตรวจหาสีผิวควรมีการตัดค่า Threshold แบบอัตโนมัติจากการทำ Gaussian ในส่วนที่เหลือจากรายงานข้างต้นและการหาค่า Threshold อัตโนมัติจากค่า r g b ที่ได้จากการ Normalize ภาพสี RGB

อย่างไรก็ตาม สิ่งที่น่าสังเกตจากรายงานการศึกษาข้างต้นคือเพราะเหตุใดในรายงานข้างต้นจึงเลือกพิจารณาเฉพาะภาพ Normalize r และ g เท่านั้น ทั้งๆที่ภาพสี RGB เกิดจากองค์ประกอบของสีสามสีร่วมกัน ดังนั้นเราจึงพิจารณาว่าควรนำภาพ Normalize r g และ b มาจับคู่กันเป็นโมเดล

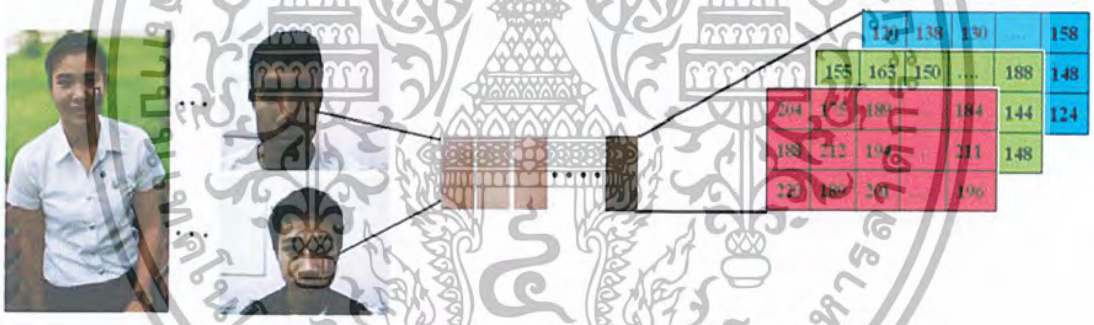
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ 4 โมเดลด้วยกัน นั่นคือ rg , rb , gb และ rgb ทั้งนี้ก็เพื่อที่จะนำผลลัพธ์ที่ได้จากแต่ละโมเดลมาเปรียบเทียบกันว่าโมเดลใดสามารถตรวจหาสีผิวได้ดีที่สุด

4.2.2 การตัดค่า Threshold จากการทำ Gaussian

ดังที่กล่าวไปแล้วว่าเนื่องจากในรายงานการศึกษาข้างต้น ยังเหลือการตัดค่า Threshold อัตโนมัติ ดังนั้น เราจึงศึกษาการตัดค่า Threshold จากการคูณ Gaussian (Gaussian Convolution) ของ rg, rb, bg และ rgb ซึ่งมีทั้งหมด 4 โมเดล ขั้นตอนการหาค่า Threshold มีดังนี้

1. ตัดตัวอย่างสีผิวบนใบหน้าโดยเลือกตัดบริเวณแก้มด้วยขนาด 20x40 พิกเซล ภาพที่ใช้เป็นตัวอย่างเพื่อการหาค่า Threshold อัตโนมัติจะใช้ภาพคนที่ถ่ายในห้องและนอกห้อง เนื่องจากความเข้มแสงมีอิทธิพลต่อสีผิวที่ถ่ายจากกล้อง Webcam และเพื่อให้การติดตามคนในภาพสามารถติดตามได้ทั้งภายในห้องและภายนอกห้อง ซึ่งตัวอย่างสีผิวที่เลือกมี โทนสีอ่อนและสีเข้มในอัตราส่วนที่เท่ากัน ตัวอย่างการตัดสีผิวบริเวณแก้มแสดงดังภาพที่ 4.6 จากนั้นนำภาพตัวอย่างสีผิวของทุกภาพมารวมกันเป็นภาพเดียวแล้วทำการแยกภาพของแสงสีแดง สีเขียวและสีน้ำเงิน



ภาพที่ 4.6 ตัวอย่างการตัดสีผิวของภาพคนที่ถ่ายทั้งภายในและภายนอกห้อง แล้วทำการแยกภาพของแสงสีแดง สีเขียวและสีน้ำเงิน

2. นำภาพตัวอย่างที่เป็นภาพสี RGB มาทำการ Normalize เป็น r g และ b ตามสมการที่ (4.1)~(4.2) และ(4.8) แล้วนำค่าที่ได้มาคูณกันในรูปของ Gaussian (Gaussian Convolution) ซึ่งมีทั้งหมด 4 โมเดล คือ RG , RB , GB และ RGB ดังสมการที่ (4.9)~(4.12) โดยเมื่อเปรียบเทียบสมการที่ (4.3) กับสมการที่ (4.9) จะเห็นว่าสมการที่ (4.3) กำหนดเป็น $f(r,g)$ แต่สมการที่ (4.9) จะกำหนดเป็น $F(R,G)$ เนื่องจาก r และ g เป็นการ Normalize ภาพในส่วนที่เป็นสีแดงและสีเขียวทำให้มีค่าแสดงในรูปของจุดทศนิยม (Floating Point) จึงไม่สามารถอ้างอิงเป็นตำแหน่ง Co-ordinate ได้ ดังนั้น เราจึงใช้วิธีการ Projection ค่า r g R และ G ซึ่งอยู่ในรูป Co-ordinate สองมิติไปเป็น Co-ordinate หนึ่งมิติ r1 g1 R1 และ G1 ดังโปรแกรมที่แสดงไว้ข้างล่างนี้ สาเหตุที่ต้องแปลงข้อมูล r g R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ G ให้เป็นขนาดหนึ่งมิติก็เนื่องจากผลคูณ Gaussian ($F(R,G)$) เป็นข้อมูลสองมิติที่อ้างอิงกับ R และ G จากนั้นทำการฉายกลับผลคูณ Gaussian ($F(R,G)$) ที่มีค่าต่ำสุดและสูงสุดกลับไปที่แกน R และ G เพื่อกำหนดให้เป็นค่า Threshold ต่ำสุดและสูงสุดของสีแดง (R) และสีเขียว (G) ดังแสดงในภาพที่ 4.7

$$b = \frac{B}{R+G+B} \tag{4.8}$$

$$F(R,G) = \frac{1}{2\pi} e^{\frac{-(r_i-\bar{r})^2}{2\sigma_r^2}} e^{\frac{-(g_i-\bar{g})^2}{2\sigma_g^2}} \tag{4.9}$$

$$F(R,B) = \frac{1}{2\pi} e^{\frac{-(r_i-\bar{r})^2}{2\sigma_r^2}} e^{\frac{-(b_i-\bar{b})^2}{2\sigma_b^2}} \tag{4.10}$$

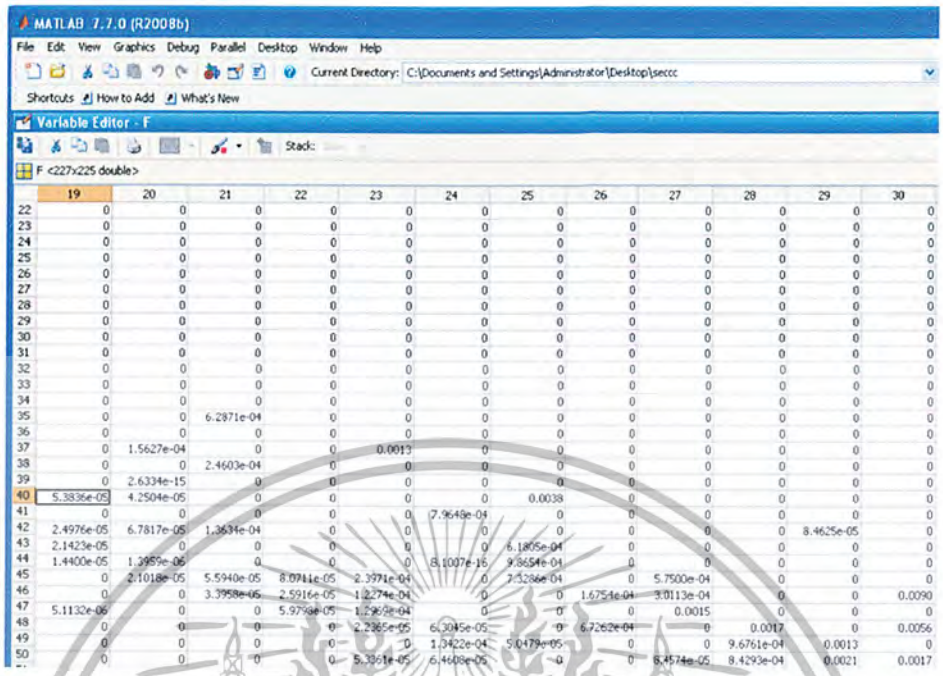
$$F(G,B) = \frac{1}{2\pi} e^{\frac{-(g_i-\bar{g})^2}{2\sigma_g^2}} e^{\frac{-(b_i-\bar{b})^2}{2\sigma_b^2}} \tag{4.11}$$

$$F(R,G,B) = \frac{1}{2\pi} e^{\frac{-(r_i-\bar{r})^2}{2\sigma_r^2}} e^{\frac{-(g_i-\bar{g})^2}{2\sigma_g^2}} e^{\frac{-(b_i-\bar{b})^2}{2\sigma_b^2}} \tag{4.12}$$

โปรแกรมการ Projection ค่า r g R และ G ซึ่งอยู่ในรูป Co-ordinate สองมิติไปเป็น Co-ordinate หนึ่งมิติ r1 g1 R1 และ G1 และการคูณ Gaussian F(R1,G1) จากค่า r1 และ g1

```
a=1;
for i=1:(m
  for j=1:(n)
     R1(a) = R(i,j);
     G1(a) = G(i,j);
     r1(a) = r(i,j);
     g1(a) = g(i,j);
     a=a+1;
  end
end
[o,p]=size(R)
for i=1:(o*p)
   if R1(i)>0 && G1(i)>0
     F(R1(i),G1(i))=(1/(2*pi))*(exp(-(((r1(i)-rbar)^2)/(2*yr))))*(exp(-(((g1(i)-gbar)^2)/(2*yg))));
   end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



	19	20	21	22	23	24	25	26	27	28	29	30
22	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	6.2871e-04	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0
37	0	1.5627e-04	0	0	0.0013	0	0	0	0	0	0	0
38	0	0	2.4603e-04	0	0	0	0	0	0	0	0	0
39	0	2.6334e-15	0	0	0	0	0	0	0	0	0	0
40	5.3836e-05	4.2504e-05	0	0	0	0	0.0038	0	0	0	0	0
41	0	0	0	0	0	7.9648e-04	0	0	0	0	0	0
42	2.4976e-05	6.7817e-05	1.3634e-04	0	0	0	0	0	0	0	8.4625e-05	0
43	2.1423e-05	0	0	0	0	0	6.1805e-04	0	0	0	0	0
44	1.4400e-05	1.3959e-06	0	0	0	8.1007e-16	9.8654e-04	0	0	0	0	0
45	0	2.1018e-05	5.5940e-05	8.0711e-05	2.3971e-04	0	2.5286e-04	0	5.7500e-04	0	0	0
46	0	0	3.3958e-05	2.5916e-05	1.2274e-04	0	0	1.6754e-04	3.0113e-04	0	0	0.0090
47	5.1132e-06	0	0	5.9798e-05	1.2969e-04	0	0	0	0.0015	0	0	0
48	0	0	0	0	2.2365e-05	6.3045e-05	0	6.7262e-04	0	0.0017	0	0.0056
49	0	0	0	0	0	1.3422e-04	5.0479e-05	0	0	9.6761e-04	0.0013	0
50	0	0	0	0	5.3361e-05	6.4608e-05	0	0	8.4574e-05	8.4293e-04	0.0021	0.0017

(ก) การเลือกค่า Threshold ต่ำสุด

	216	217	218	219	220	221	222	223	224	225	226	227
219	0	0	0	0	0	0	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0	0	0	0
221	0	0	0	0	0	0	0	0	0	0	0	0
222	0	0	0	0	0	0	0	0	0	0	0	0
223	0	0	0	0	0	0	0	0	0	0	0	0
224	0	0	0	0	0	0	0	0	0	0	0	0
225	0	0	0	0	0	0	0	0	0	0	0	0
226	0	0	0	0	0	0	0	0	0	0	0	0
227	0	0	0	0	0	0	0	0	0	0	0	0
228	0.0216	0	0	0	0	0	0	0	0	0	0	0
229	0	0.0216	0	0	0	0	0	0	0	0	0	0
230	0	0.0190	0.0215	0	0	0	0	0	0	0	0	0
231	0.0228	0	0.0190	0.0215	0	0	0	0	0	0	0	0
232	0.0264	0.0228	0	0	0.0215	0	0	0	0	0	0	0
233	0.0221	0.0264	0.0228	0	0	0.0214	0	0	0	0	0	0
234	0	0.0222	0.0263	0.0228	0	0	0	0	0	0	0	0
235	0.0467	0	0	0.0263	0	0	0	0	0	0	0	0
236	0	0.0469	0	0	0	0	0	0	0	0	0	0
237	0	0	0.0461	0	0	0	0	0	0	0	0	0
238	0.0579	0	0	0	0.0432	0	0	0	0	0	0	0
239	0	0	0	0.0496	0	0	0	0	0	0	0	0
240	0	0	0	0	0.0493	0	0	0	0	0	0	0
241	0	0	0.0538	0	0	0	0.0455	0	0	0	0	0
242	0	0	0	0.0535	0	0	0	0	0	0	0	0
243	0	0	0	0	0.0533	0.0561	0	0.0472	0	0	0	0
244	0	0	0	0	0	0	0.0545	0	0	0	0	0
245	0	0	0	0	0	0	0	0	0	0	0	0
246	0	0	0	0	0	0	0	0	0	0	0	0
247	0	0	0	0	0	0	0	0	0	0.0535	0	0.0473
248	0	0	0	0	0	0	0	0	0	0	0	0
249	0	0	0	0	0	0	0	0	0	0	0	0.0529

(ข) การเลือกค่า Threshold สูงสุด

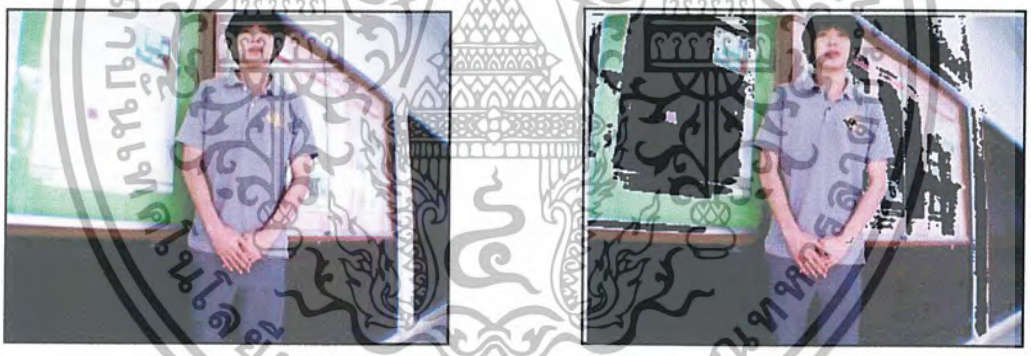
ภาพที่ 4.7 การเลือกค่า Gaussian เป็นค่า Threshold ต่ำสุดและสูงสุด จากการฉายกลับค่า Gaussian ไปที่แนวแกน R และ G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการเลือกค่า Threshold ตามที่ได้อธิบายข้างต้น ทำให้เราได้ขอบเขตค่า Threshold ของสีแดง (R) และสีเขียว (G) อยู่ระหว่าง 40~249 และ 19~227 ตามลำดับ จากนั้นนำค่า Threshold ที่ได้มาทดลองใช้กับภาพถ่ายคนทั้งที่อยู่ภายในและภายนอกห้อง รวมทั้งภาพจากแหล่งอื่น ซึ่งภาพผลลัพธ์แสดงดังภาพที่ 4.8~4.10

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.8 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RG



(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.9 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก)ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.10 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RG

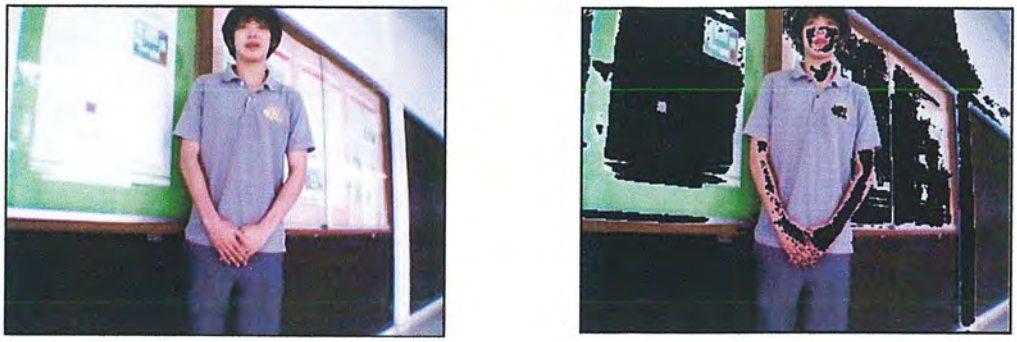
จากภาพที่ 4.8 จะพบว่าวิธีการตัด Threshold อัตโนมัติจากการทำ Gaussian สำหรับโมเดล RG ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้โดยเสื้อสีขาวและพื้นหลังสีเทายังคงปรากฏชัดอยู่ รวมทั้งบริเวณที่เป็นสีผิวก็มีสีดำปะปนอยู่ที่บริเวณคาง ส่วนภาพที่ 4.9 ภาพผลลัพธ์ที่ได้ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้โดยสีพื้นหลังที่เป็นสีเขียว สีแดง และสีเทาก็ยังคงปรากฏอยู่ และภาพที่ 4.10 ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้เช่นเดียวกัน บริเวณที่เป็นสีผิวยังปรากฏสีดำให้เห็นอีกด้วย

จากนั้น เราได้ทดลองใช้โมเดล RB GB และ RGB เพื่อหาค่า Threshold อัตโนมัติตามหลักการที่ได้อธิบายข้างต้น และได้ทำการทดลองตัดค่า Threshold เพื่อตรวจหาสีผิว ซึ่งผลการทดลองแสดงดังภาพที่ 4.11~4.19

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.11 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.12 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RB

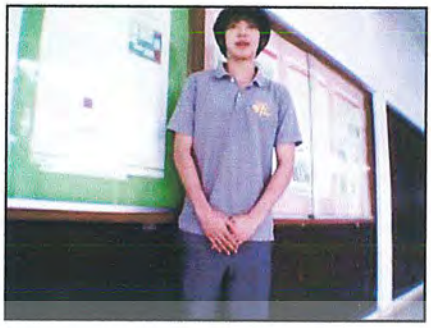
(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.13 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RB

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.14 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ภาพต้นฉบับ

(ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.15 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล GB

(ก) ภาพต้นฉบับ

(ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

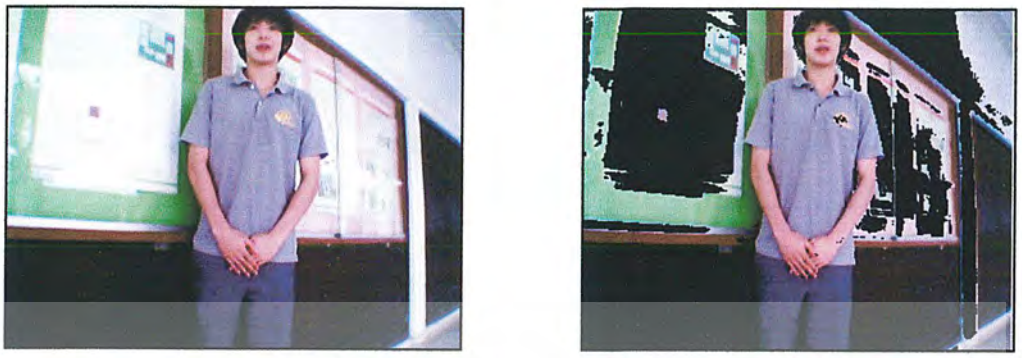
ภาพที่ 4.16 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล GB

(ก) ภาพต้นฉบับ

(ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.17 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.18 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล RGB

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.19 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล RGB

สรุปจากผลการทดลองกับโมเดล RB GB และ RGB ในการตรวจหาสีผิวไม่สามารถทำได้เช่นเดียวกับโมเดล RG ไม่ว่าภาพถ่ายคนจะอยู่ภายในหรือภายนอกห้อง รวมทั้งภาพคนจากแหล่งอื่นด้วย แสดงว่าการตัดค่า Threshold จากการทำ Gaussian ไม่เหมาะสมที่จะนำมาตรวจหาสีผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การตัดค่า Threshold จากค่า r g และ b ที่ได้จากการ Normalize ภาพสี RGB

ขั้นตอนการตัดค่า Threshold จากค่า r g และ b ที่ได้จากการ Normalize ภาพสี RGB มีดังนี้

1. เริ่มจากการนำตัวอย่างสีผิวมารวมแล้วทำการแยกภาพของแสงสีต่างๆ เช่นเดียวกับวิธีการตัดค่า Threshold จากการทำ Gaussian แล้วทำการ Normalize ภาพตามสมการที่ (4.1)~(4.2) และ(4.8) แล้วนำค่า r g และ b มาหาค่าสูงสุด (Maximum) ค่าต่ำสุด (Minimum) ค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ดังสมการที่ (4.4)~(4.7) และ (4.13)~(4.14) ซึ่งการตัดค่า Threshold จากค่า r g และ b เป็นการคำนวณในลักษณะของหนึ่งมิติแทนที่จะเป็นการคำนวณสองมิติเหมือนกับการตัดค่า Threshold จากการทำ Gaussian

2. เนื่องจากค่าต่ำสุดและสูงสุดของ r g และ b เป็นตัวแทนของสีผิว ดังนั้นเพื่อการหาค่า Threshold ต่ำสุดและสูงสุดของ r g และ b เราจะนำค่าเบี่ยงเบนมาตรฐานคูณกับค่าคงที่ตั้งแต่ 0.5~6 โดยเพิ่มขึ้นทีละ 0.5 นั่นคือเราจะมีค่าเบี่ยงเบนมาตรฐานอันใหม่เป็น 0.5 σ~6 σ จากนั้นนำค่าเบี่ยงเบนมาตรฐานอันใหม่แต่ละตัวมาบวกและลบออกค่าเฉลี่ย ซึ่งค่าผลลัพธ์ที่ได้จะนำมาเปรียบเทียบกับค่าต่ำสุดและสูงสุดของ r g และ b ถ้าค่าผลลัพธ์ที่ได้ค่าใดมีค่าใกล้เคียงกับค่าต่ำสุดหรือสูงสุด จะเลือกค่าผลลัพธ์ค่านั้นเป็นค่า Threshold ต่ำสุดและสูงสุดต่อไป

$$\bar{b} = \frac{1}{N}\sum_{i=1}^{N} b_i \tag{4.13}$$

$$\sigma_b^2 = \frac{1}{N}(b_i - \bar{b})^2 \tag{4.14}$$

ตัวอย่างการหาค่า Threshold ต่ำสุดและสูงสุดของ r จากตารางที่ 4.1 ค่าต่ำสุดของ r (r min) และค่าสูงสุดของ r (r max) ของตัวอย่างสีผิวในข้อที่ 1 มีค่าเท่ากับ 0.3682 และ 0.6806 ตามลำดับ จากนั้นนำค่าเบี่ยงเบนมาตรฐานอันใหม่ตั้งแต่ 0.5 σ~6 σ แต่ละตัวไปบวกและลบออกจากค่าเฉลี่ย (r bar) แล้วนำค่าผลลัพธ์ที่ได้ไปเปรียบเทียบกับค่าต่ำสุดและสูงสุดของ r ซึ่งเราจะพบว่าที่ค่า 2.5 σ มีค่าผลลัพธ์เท่ากับ 0.3595 และที่ค่า 6 σ มีค่าผลลัพธ์เท่ากับ 0.6854 ซึ่งค่าดังกล่าวเป็นค่าที่ใกล้เคียงกับค่าต่ำสุดและสูงสุดของ r มากที่สุด ดังนั้น เราจึงเลือกค่าผลลัพธ์เหล่านี้เป็นขอบเขตค่า Threshold ของ r สำหรับการหาค่า Threshold ต่ำสุดและสูงสุดของ g และ b ก็มีหลักการคิดเช่นเดียวกับของ r ทุกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ r

max	r min	r bar	จำนวนเท่า σ	σ_r	r left	r right
0.6806	0.3682	0.4553	0.5	0.0157	0.4362	0.4745
0.6806	0.3682	0.4553	1	0.0314	0.4170	0.4937
0.6806	0.3682	0.4553	1.5	0.047	0.3978	0.5129
0.6806	0.3682	0.4553	2	0.0627	0.3786	0.532
0.6806	0.3682	0.4553	2.5	0.0784	0.3595	0.5512
0.6806	0.3682	0.4553	3	0.0941	0.3403	0.5704
0.6806	0.3682	0.4553	3.5	0.1097	0.3211	0.5896
0.6806	0.3682	0.4553	4	0.1254	0.302	0.6087
0.6806	0.3682	0.4553	4.5	0.1411	0.2828	0.6279
0.6806	0.3682	0.4553	5	0.1568	0.2636	0.6471
0.6806	0.3682	0.4553	5.5	0.1724	0.2444	0.6662
0.6806	0.3682	0.4553	6	0.1881	0.2253	0.6854

ตารางที่ 4.2 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ g

g max	g min	g bar	จำนวนเท่า σ	σg	g left	g right
0.3768	0.2718	0.3234	0.5	0.0057	0.3177	0.3291
0.3768	0.2718	0.3234	1	0.0113	0.312	0.3348
0.3768	0.2718	0.3234	1.5	0.017	0.3063	0.3405
0.3768	0.2718	0.3234	2	0.0227	0.3005	0.3462
0.3768	0.2718	0.3234	2.5	0.0283	0.2948	0.3519
0.3768	0.2718	0.3234	3	0.034	0.2891	0.3576
0.3768	0.2718	0.3234	3.5	0.0397	0.2834	0.3633
0.3768	0.2718	0.3234	4	0.0453	0.2777	0.369
0.3768	0.2718	0.3234	4.5	0.051	0.2720	0.3747
0.3768	0.2718	0.3234	5	0.0567	0.2663	0.3804
0.3768	0.2718	0.3234	5.5	0.0623	0.2606	0.3861
0.3768	0.2718	0.3234	6	0.068	0.2549	0.3918

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ b

b max	b min	b bar	จำนวนเท่า σ	σb	b left	b right
0.2997	0	0.2214	0.5	0.0159	0.2031	0.2396
0.2997	0	0.2214	1	0.0317	0.1849	0.2579
0.2997	0	0.2214	1.5	0.0476	0.1666	0.2761
0.2997	0	0.2214	2	0.0635	0.1483	0.2944
0.2997	0	0.2214	2.5	0.0793	0.1301	0.3126
0.2997	0	0.2214	3	0.0952	0.1118	0.3309
0.2997	0	0.2214	3.5	0.1111	0.0936	0.3492
0.2997	0	0.2214	4	0.1269	0.0753	0.3674
0.2997	0	0.2214	4.5	0.1428	0.0571	0.3857
0.2997	0	0.2214	5	0.1587	0.0388	0.4039
0.2997	0	0.2214	5.5	0.1745	0.0206	0.4222
0.2997	0	0.2214	6	0.1904	0.0023	0.4404

ในการกำหนดค่า Threshold ของ r g และ b ตามวิธีที่ได้อธิบายข้างต้น เราจะได้ขอบเขต Threshold ของ r ตั้งแต่ 0.3595~0.6854 ของ g ตั้งแต่ 0.2944~0.0023 และของ b ตั้งแต่ 0.272~0.3747 เมื่อได้ค่า Threshold ของ r g b มาแล้วนำมาใช้กับโมเดลทั้ง 4 โมเดลคือ rg, rb, gb และ rgb เพื่อเปรียบเทียบดูว่าระหว่างแนวความคิดการเลือกค่า Threshold ซึ่งได้จากการทำ Gaussian กับแนวความคิดการหาค่า Threshold จากการเลือกค่า r g b ที่ทำการ Normalize ภาพแนวความคิดใดที่เหมาะสมที่สุดในการหาค่า Threshold โดยแสดงในแต่ละกรณีได้ดังนี้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.21 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.22 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rg

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.23 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rg

จากภาพที่ 4.21 จะพบว่าวิธีการตัดค่า Threshold อัตโนมัติจากค่า r g b ที่ได้จากการ Normalize ภาพสี RGB โดยใช้โมเดล rg สามารถแยกสีผิวออกจากสีพื้นหลังได้ชัดเจน แม้ว่าจะมีสีดำปะปนอยู่บริเวณใบหน้าบ้างเล็กน้อย ส่วนภาพที่ 4.22 ภาพผลลัพธ์ที่ได้สามารถแยกสีผิวและสีน้ำตาลของเก้าอี้ออกจากสีพื้นหลังได้แต่ไม่ทั้งหมด เพราะจะเห็นว่าสีเทาบางส่วนของพื้นหลังยังปรากฏอยู่ และภาพที่ 4.23 สามารถแยกสีผิวออกจากสีพื้นหลังได้เช่นเดียวกันโดยจะเห็นว่าเสื้อซึ่งมีสีขาวเปลี่ยนเป็นสีดำและบริเวณปากก็มีสีดำปะปนอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.24 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rb

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.25 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rb

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.26 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 4.24 พบว่าภาพที่ได้จากการตัดค่า Threshold โดยใช้โมเดล rb สามารถแยกสีผิวออกจากสีพื้นหลังได้เช่นเดียวกันและยังปรากฏสีผิวชัดเจนกว่าโมเดล rg ด้วยส่วนภาพที่ 4.25 ภาพผลลัพธ์ที่ได้สามารถแยกสีผิวและสีน้ำตาลของเก้าอี้ออกจากสีพื้นหลัง รวมทั้งสีขาวของเสื้อเป็นสีดำ แต่ว่าสีเขียวและสีเทาของพื้นหลังปรากฏขึ้นมาเมื่อเปรียบเทียบกับโมเดล rg ซึ่งสามารถแยกสีเขียวของพื้นหลังออกจากสีผิวได้ และภาพที่ 4.26 สามารถแยกสีผิวออกจากสีพื้นหลังได้เช่นเดียวกันและมีความใกล้เคียงกับโมเดล rg แตกต่างกันตรงที่บริเวณปากไม่ปรากฏสีดำให้เห็นเลย

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.27 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล gb

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.28 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล gb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.29 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล gb

จากภาพที่ 4.27 พบว่าภาพที่ได้จากการตัดค่า Threshold โดยใช้โมเดล gb ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้ทั้งหมด โดยจะปรากฏสีขาวขึ้นมาในส่วนของพื้นหลังซึ่งเดิมเป็นสีเทา ส่วนภาพที่ 4.28 ภาพผลลัพธ์ที่ได้สามารถแยกสีผิวและสีน้ำตาลของเก้าอี้ออกจากสีพื้นหลังแต่สีขาวของพื้นหลังและเสื้อบางส่วนไม่สามารถแยกออกจากกันได้ทั้งหมด รวมทั้งบริเวณปากก็มีสีดำปะปนอยู่บ้างเล็กน้อย และภาพที่ 4.29 จะพบว่าไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้เลยต่างจากโมเดล rg และ rb ที่สามารถแยกออกจากกันได้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.30 ภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rgb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.31 ภาพถ่ายคนที่อยู่ภายนอกห้องเมื่อใช้โมเดล rgb

(ก) ภาพต้นฉบับ (ข) ภาพผลลัพธ์หลังจากตัดค่า Threshold

ภาพที่ 4.32 ภาพถ่ายคนที่มาจากแหล่งอื่นเมื่อใช้โมเดล rgb

จากภาพที่ 4.30-4.32 พบว่าภาพที่ได้จากการตัดค่า Threshold โดยใช้โมเดล rgb สามารถแยกสีผิวออกจากสีพื้นหลังได้และคุณภาพของภาพมีความใกล้เคียงกับโมเดล rg ดังที่ได้แสดงไว้ข้างต้น

การตัดค่า Threshold จากค่า r g b ที่ได้จากการ Normalize ภาพสี RGB พบว่าแนวความคิดนี้สามารถตรวจหาสีผิวของคนที่อยู่ในภาพได้ไม่ว่าจะใช้โมเดล rg, rb, gb หรือ rgb เราจึงเลือกแนวความคิดนี้เป็นวิธีการตัดค่า Threshold เพียงแต่คุณภาพในการหาสีผิวของแต่ละโมเดลมีความแตกต่างกัน ดังนั้นเพื่อที่จะสามารถเลือกโมเดลที่เหมาะสมที่สุดในการตรวจหาสีผิว จึงทำการเปรียบเทียบแต่ละโมเดลโดยใช้ภาพคนที่ถ่ายภายในห้อง (Indoor) ภายนอกห้อง (Outdoor) และภาพจากแหล่งอื่นมาพิจารณาเป็นจำนวนทั้งหมด 100 ภาพ เพราะว่าสีผิวมนุษย์นั้นมีความหลากหลายแตกต่างกันไปขึ้นอยู่กับลักษณะพันธุกรรมและเชื้อชาติ อย่างไรก็ตามความเข้มแสง (Intensity) ก็มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิทธิพลต่อสีผิวด้วย เช่น คนๆ เดียวกันเมื่ออยู่ในห้องจะมีสีผิวที่แตกต่างกับตอนอยู่นอกห้อง เพราะถ้าแสงภายนอกห้องมีความสว่างมากกว่าภายในห้องก็จะทำให้มีสีผิวขณะที่อยู่นอกห้องมีสีอ่อนกว่าเมื่ออยู่ภายในห้อง สิ่งที่น่าสังเกตก็คือโมเดลทั้ง 4 โมเดล แม้ว่าจะสามารถแยกสีผิวได้ แต่ก็น่าจะตรวจสอบว่าโมเดลใดสามารถหาสีผิวได้ถูกต้องมากที่สุด นอกจากนี้ยังมีอีกประเด็นหนึ่งที่น่าสนใจและควรตรวจสอบคืออัตราส่วนในการตัดตัวอย่างสีผิวระหว่างโทนอ่อนต่อโทนเข้ม เนื่องจากในตอนต้นเราคัดเลือกตัวอย่างสีผิวโทนอ่อนต่อโทนเข้มในอัตราส่วนที่เท่ากัน แต่ถ้าเรากำหนดให้อัตราส่วนสีผิวโทนอ่อนมากกว่าโทนเข้ม หรือสีผิวโทนอ่อนน้อยกว่าโทนเข้มจะมีผลอย่างไรต่อการกำหนดค่า Threshold

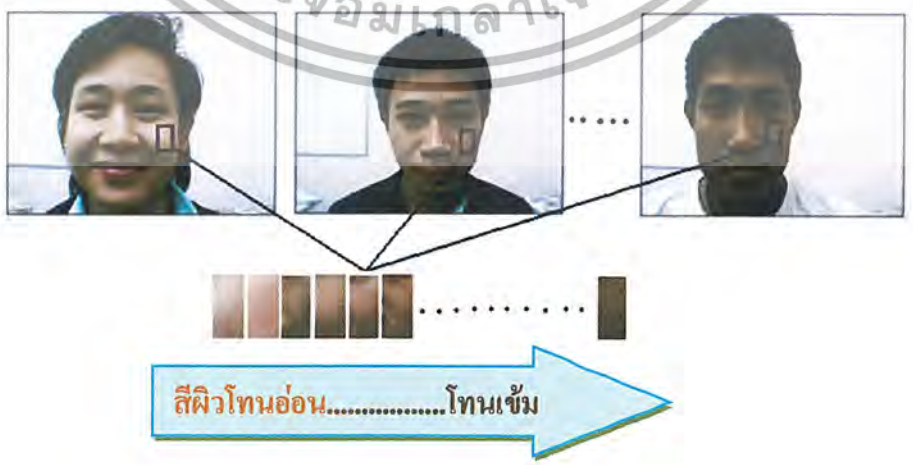
4.2.4 การเลือกตัวอย่างสีผิวสำหรับการตัดค่า Threshold ที่เหมาะสม

ในการเลือกตัวอย่างสีผิวสำหรับการตัดค่า Threshold ที่เหมาะสมมีประเด็นที่น่าสนใจอีกประเด็นหนึ่งก็คือโทนสีผิวเพราะว่าคนเรามีสีผิวที่แตกต่างกันออกไป บางคนมีสีผิวขาว บางคนสีผิวคล้ำ ถ้าเรากำหนดตัวอย่างสีผิวได้ไม่ครอบคลุมก็จะทำให้โปรแกรมตรวจหาคนที่อยู่ในภาพผิดเพี้ยนไปหรือไม่สมบูรณ์ ซึ่งการกำหนดอัตราส่วนของโทนสีผิวจึงมีความสำคัญต่อการกำหนดค่า Threshold ดังนั้นจึงกำหนดเลือกโทนสีผิวโทนอ่อนและโทนเข้มเป็น 3 กรณีดังนี้ โดยในแต่ละกรณีการตัดตัวอย่างสีผิวจะมีขนาด 20x40 พิกเซล ซึ่งจะมีสีผิวของคนที่ถ่ายภายในห้องจำนวน 16 คน แบ่งเป็นสีผิวโทนอ่อน 8 คน สีผิวโทนเข้ม 8 คน และสีผิวของคนที่ถ่ายภายนอกห้องอีกจำนวน 16 คน โดยมีสีผิวโทนอ่อนและโทนเข้มในจำนวนที่เท่ากัน

กรณีที่ 1 ตัวอย่างสีผิวเป็น 20% : 80% คือสีผิวโทนอ่อน 20% และสีผิวโทนเข้ม 80%

กรณีที่ 2 ตัวอย่างสีผิวเป็น 50% : 50% คือสีผิวโทนอ่อน 50% และสีผิวโทนเข้ม 50%

กรณีที่ 3 ตัวอย่างสีผิวเป็น 80% : 20% คือสีผิวโทนอ่อน 80% และสีผิวโทนเข้ม 20%



ภาพที่ 4.33 การตัดตัวอย่างสีผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การตรวจหาสีของปลอกแขน (Armband's color Detection)

การติดตามคนที่อยู่ในภาพนั้น ในกรณีที่มีคนอยู่ในภาพหลายคนจะทำให้โปรแกรมไม่สามารถเลือกได้ว่าจะติดตามคนใดคนหนึ่งในภาพ ดังนั้นเพื่อให้ง่ายต่อติดตามคนในภาพ เราจึงใช้วิธีติดตามคนที่อยู่ในภาพที่ใส่ปลอกแขนสีแดง สีเขียว และสีน้ำเงิน โดยผู้ควบคุมต้องระบุว่าต้องการติดตามคนที่ใส่ปลอกแขนสีอะไร ภาพที่ 4.34 แสดงภาพคนที่ใส่ปลอกแขนสีต่างๆ

ภาพที่ 4.34 ภาพคนที่ใส่ปลอกแขนสีต่างๆ

วิธีการหาสีของปลอกแขน

เนื่องจากปลอกแขนสีแดง สีเขียว และสีน้ำเงิน ต่างก็มีค่าระดับสีของ RGB เป็นลักษณะเฉพาะของแต่ละสี ดังนั้นเราจะทำการหาค่า Threshold ของปลอกแขนแต่ละสีเพื่อให้โปรแกรมสามารถแยกแยะได้ว่าปลอกแขนที่สวมใส่เป็นสีอะไร ซึ่งขั้นตอนการหาค่า Threshold ของปลอกแขนแต่ละสีมีดังต่อไปนี้

1. เก็บภาพคนที่สวมปลอกแขนแต่ละสี ในขั้นตอนนี้เพื่อให้โปรแกรมสามารถตรวจหาสีของปลอกแขนได้ทั้งภาพที่อยู่ในห้องและนอกห้อง ดังนั้น เราได้ทำการเก็บภาพคนที่สวมปลอกแขนสีต่างๆที่ถ่ายภายในห้อง (Indoor) จำนวน 5 รูป และถ่ายภายนอกห้อง (Outdoor) จำนวน 5 รูป

2. นำรูปที่ได้มาตัดเฉพาะส่วนที่เป็นปลอกแขน โดยใช้รูปส่วนที่เป็นปลอกแขนภายในห้อง(Indoor) และภายนอกห้อง (Outdoor) ตัดรูปละ 4 ส่วน ขนาด 10 x 10 พิกเซล โดยเลือกตัดให้ทั่วทั้งปลอกแขนเพื่อให้ได้ค่าระดับโทนสีครอบคลุมมากที่สุด ดังนั้นจะได้ส่วนของปลอกแขนสีแดง สีเขียว และสีน้ำเงิน อย่างละ 40 ส่วน ดังรูปที่ 4.35-4.37 แสดงตัวอย่างส่วนของปลอกแขนสีแดง สีเขียว และสีน้ำเงิน ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.35 ตัวอย่างส่วนของปลอกแขนสีแดง

ภาพที่ 4.36 ตัวอย่างส่วนของปลอกแขนสีเขียว

ภาพที่ 4.37 ตัวอย่างส่วนของปลอกแขนสีน้ำเงิน

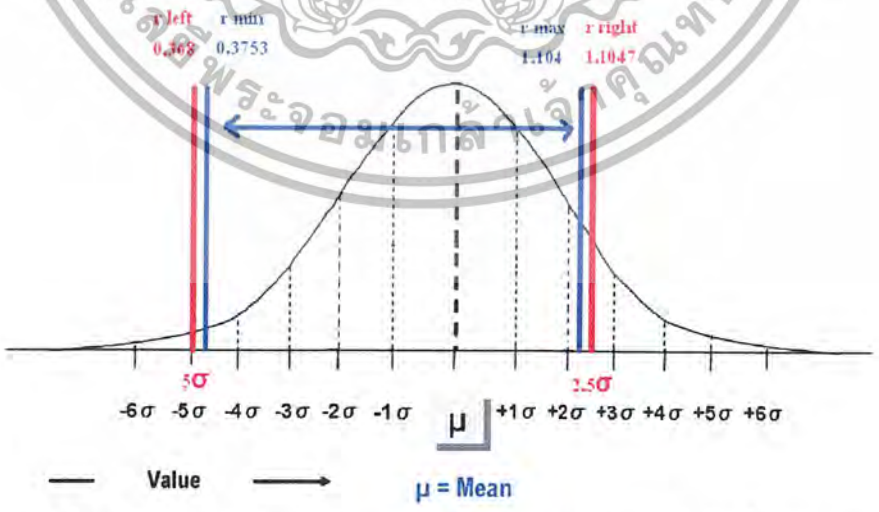
3. นำส่วนของปลอกแขนแต่ละสีที่ได้มารวมกัน โดยใช้โปรแกรม MATLAB ภาพที่ 4.38 แสดงภาพที่เกิดจากการนำส่วนของปลอกแขนแต่ละสีที่ถ่ายภายในห้องและนอกห้องมารวมกัน เราจะสังเกตุเห็นได้ว่าโทนสีที่อยู่ข้างซ้ายมือจะมีสีเข้มกว่าด้านขวามือเนื่องจากเป็นส่วนของปลอกแขนที่ถ่ายภายในห้องจึงมีโทนสีเข้มกว่าปลอกแขนที่ถ่ายนอกห้อง จากนั้นนำภาพมาทำ Normalize ตามสมการที่ (4.1) , (4.2) และ (4.3) เพื่อหาค่า r, g และ b

ภาพที่ 4.38 ภาพที่เกิดจากการนำส่วนของปลอกแขนแต่ละสีมารวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นำค่า r, g และ b ที่ได้มาหาค่าสูงสุด (Maximum) ค่าต่ำสุด (Minimum) ค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ของตัวอย่างปลอกแขนแต่ละสี ดังสมการที่ (4.4)~(4.7) และ (4.13)~(4.14)

5. นำค่าที่ได้ทั้งหมดมาหาค่า Threshold โดยใช้วิธีเดียวกันกับการหาค่า Threshold ของสีผิวโดยนำค่าเบี่ยงเบนมาตรฐานอันใหม่ที่มีค่าตั้งแต่ 0.5 σ~6 σ ไปบวกและลบออกจากค่าเฉลี่ยจะได้ค่าด้านซ้ายมือสุดและขวามือสุด (left และ right) ของ r g และ b จากนั้นนำค่าผลลัพธ์ที่ได้ไปเปรียบเทียบกับค่าต่ำสุดและสูงสุดของตัวอย่างปลอกแขนแต่ละสี ตัวอย่างเช่นตารางที่ 4.4 ค่าสูงสุด (r max) และค่าต่ำสุด (r min) ของปลอกแขนสีแดง มีค่าเท่ากับ 0.3753 และ 1.104 ตามลำดับ ดังนั้น เราจะเลือกค่า Threshold ที่มีค่าใกล้เคียงกับค่าสูงสุดและต่ำสุดของตัวอย่างปลอกแขนมากที่สุด นั่นคือเราจะพบว่า r left และ r right ที่มีค่าเท่ากับ 5σ และ 2.5σ (0.368 และ 1.104) ตามลำดับ จะมีค่าใกล้เคียงกับค่าสูงสุดและค่าต่ำสุดของปลอกแขนสีแดงมากที่สุด จากรูปที่ 4.39 เส้นสีน้ำเงินแสดงค่าสูงสุด (r max) และค่าต่ำสุด (r min) ของปลอกแขนสีแดง ส่วนเส้นสีแดงแสดงค่า Threshold ทางด้านซ้าย (r left) และค่า Threshold ทางด้านขวา (r right) โดยจากรูปเราจะเห็นได้ว่าเราจะเลือกค่าที่ใกล้เคียงกับค่าสูงสุด (r max) และค่าต่ำสุด (r min) ของปลอกแขนสีแดงให้มากที่สุด โดยค่า Threshold ทางด้านซ้าย (r left) จะต้องมีค่าน้อยกว่าค่าต่ำสุด (r min) และค่า Threshold ทางด้านขวา (r right) จะต้องมีค่ามากกว่าค่าสูงสุด (r max) เสมอ จึงจะได้ค่า Threshold ที่ครอบคลุมทั่วทั้งย่าน ส่วนปลอกแขนสีอื่นก็ทำในลักษณะเดียวกัน จะได้ผลลัพธ์ดังตารางที่ 4.4-4.12



ภาพที่ 4.39 การเลือกค่า Threshold ของค่า r ของปลอกแขนสีแดงที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ค่าของปลอกแขนสีแดงแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ r

r max	r min	r bar	จำนวนเท่า σ	σ_r	r left	r right
1.104	0.3753	0.8174	6	0.84	0.2697	1.4486
1.104	0.3753	0.8174	5.5	0.77	0.3188	1.3994
1.104	0.3753	0.8174	5	0.70	0.368	1.3503
1.104	0.3753	0.8174	4.5	0.63	0.4171	1.3012
1.104	0.3753	0.8174	4	0.56	0.4662	1.2521
1.104	0.3753	0.8174	3.5	0.49	0.5153	1.203
1.104	0.3753	0.8174	3	0.42	0.5644	1.1538
1.104	0.3753	0.8174	2.5	0.35	0.6135	1.1047
1.104	0.3753	0.8174	2	0.28	0.6627	1.0556
1.104	0.3753	0.8174	1.5	0.21	0.7609	0.9574
1.104	0.3753	0.8174	1	0.14	0.7609	0.9574
1.104	0.3753	0.8174	0.5	0.07	0.81	0.9083

ตารางที่ 4.5 ค่าของปลอกแขนสีแดงแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ g

g max	g min	g bar	จำนวนเท่า σ	σ_g	g left	g right
0.312	-0.003	0.1418	6	0.84	-0.1213	0.3307
0.312	-0.003	0.1418	5.5	0.77	-0.1025	0.3118
0.312	-0.003	0.1418	5	0.70	-0.0837	0.293
0.312	-0.003	0.1418	4.5	0.63	-0.0648	0.2742
0.312	-0.003	0.1418	4	0.56	-0.046	0.2553
0.312	-0.003	0.1418	3.5	0.49	-0.0272	0.2365
0.312	-0.003	0.1418	3	0.42	-0.0083	0.2177
0.312	-0.003	0.1418	2.5	0.35	0.0105	0.1988
0.312	-0.003	0.1418	2	0.28	0.0293	0.18
0.312	-0.003	0.1418	1.5	0.21	0.067	0.1423
0.312	-0.003	0.1418	1	0.14	0.0006	0.1423
0.312	-0.003	0.1418	0.5	0.07	0.0085	0.1235

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ค่าของปลอกแขนสีแดงแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ b

b max	b min	b bar	จำนวนเท่า σ	σ_b	b left	b right
0.313	-0.009	0.1409	6	0.84	-0.2234	0.322
0.313	-0.009	0.1409	5.5	0.77	-0.2006	0.3
0.313	-0.009	0.1409	5	0.70	-0.1779	0.277
0.313	-0.009	0.1409	4.5	0.63	-0.1552	0.254
0.313	-0.009	0.1409	4	0.56	-0.1324	0.231
0.313	-0.009	0.1409	3.5	0.49	-0.1097	0.209
0.313	-0.009	0.1409	3	0.42	-0.087	0.186
0.313	-0.009	0.1409	2.5	0.35	-0.0642	0.163
0.313	-0.009	0.1409	2	0.28	-0.0415	0.14
0.313	-0.009	0.1409	1.5	0.21	0.004	0.095
0.313	-0.009	0.1409	1	0.14	0.0004	0.761
0.313	-0.009	0.1409	0.5	0.07	0.0267	0.072

ตารางที่ 4.7 ค่าของปลอกแขนสีเขียวแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ r

r max	r min	r bar	จำนวนเท่า σ	σ_r	r left	r right
1.104	-0.099	0.1504	6	0.828	-0.5248	0.4292
1.104	-0.099	0.1504	5.5	0.759	-0.4681	0.4013
1.104	-0.099	0.1504	5	0.690	-0.4115	0.3734
1.104	-0.099	0.1504	4.5	0.621	-0.3549	0.3455
1.104	-0.099	0.1504	4	0.552	-0.2983	0.3175
1.104	-0.099	0.1504	3.5	0.483	-0.2416	0.2896
1.104	-0.099	0.1504	3	0.414	-0.1989	0.2617
1.104	-0.099	0.1504	2.5	0.345	-0.1284	0.2337
1.104	-0.099	0.1504	2	0.277	-0.0825	0.2058
1.104	-0.099	0.1504	1.5	0.207	-0.0242	0.1779
1.104	-0.099	0.1504	1	0.138	0.034	0.15
1.104	-0.099	0.1504	0.5	0.069	0.0982	0.122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 ค่าของปลอกแขนสีเขียวแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ g

g max	g min	g bar	จำนวนเท่า σ	σ_g	g left	g right
0.312	-0.003	0.1418	6	0.828	-0.8001	-0.8
0.312	-0.003	0.1418	5.5	0.759	-0.684	1.87
0.312	-0.003	0.1418	5	0.690	-0.568	1.7539
0.312	-0.003	0.1418	4.5	0.621	-0.4519	1.6378
0.312	-0.003	0.1418	4	0.552	-0.3358	1.5217
0.312	-0.003	0.1418	3.5	0.483	-0.2197	1.4056
0.312	-0.003	0.1418	3	0.414	-0.1036	1.2895
0.312	-0.003	0.1418	2.5	0.345	0.0125	1.1734
0.312	-0.003	0.1418	2	0.277	0.1286	1.0574
0.312	-0.003	0.1418	1.5	0.207	0.2447	0.9413
0.312	-0.003	0.1418	1	0.138	0.3608	0.8252
0.312	-0.003	0.1418	0.5	0.069	0.4769	0.7091

ตารางที่ 4.9 ค่าของปลอกแขนสีเขียวแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ b

b max	b min	b bar	จำนวนเท่า σ	σ_b	b left	b right
0.313	-0.009	0.1409	6	0.828	1.9861	1.048
0.313	-0.009	0.1409	5.5	0.759	-0.3608	0.987
0.313	-0.009	0.1409	5	0.690	-0.2995	0.925
0.313	-0.009	0.1409	4.5	0.621	-0.2383	0.864
0.313	-0.009	0.1409	4	0.552	-0.177	0.803
0.313	-0.009	0.1409	3.5	0.483	-0.1158	0.742
0.313	-0.009	0.1409	3	0.414	-0.0546	0.68
0.313	-0.009	0.1409	2.5	0.345	0.0067	0.619
0.313	-0.009	0.1409	2	0.277	0.0679	0.558
0.313	-0.009	0.1409	1.5	0.207	0.1292	0.497
0.313	-0.009	0.1409	1	0.138	0.1904	0.435
0.313	-0.009	0.1409	0.5	0.069	0.2517	0.374

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ค่าของปลอกแขนสีน้ำเงินแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ r

r max	r min	r bar	จำนวนเท่า σ	σ_r	r left	r right
1.104	-0.108	0.3897	6	0.348	-0.3787	0.5984
1.104	-0.108	0.3897	5.5	0.319	-0.338	0.5577
1.104	-0.108	0.3897	5	0.29	-0.2973	0.517
1.104	-0.108	0.3897	4.5	0.261	-0.2566	0.4763
1.104	-0.108	0.3897	4	0.232	-0.2159	0.4356
1.104	-0.108	0.3897	3.5	0.203	-0.1752	0.3948
1.104	-0.108	0.3897	3	0.174	-0.1344	0.3541
1.104	-0.108	0.3897	2.5	0.145	-0.0937	0.3134
1.104	-0.108	0.3897	2	0.116	-0.053	0.2727
1.104	-0.108	0.3897	1.5	0.087	-0.0123	0.232
1.104	-0.108	0.3897	1	0.058	0.0284	0.1913
1.104	-0.108	0.3897	0.5	0.029	0.0691	0.1506

ตารางที่ 4.11 ค่าของปลอกแขนสีน้ำเงินแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ g

g max	g min	g bar	จำนวนเท่า σ	σ_g	g left	g right
-0.088	1.018	0.4599	6	0.348	-0.5832	1.0838
-0.088	1.018	0.4599	5.5	0.319	-0.5137	1.0143
-0.088	1.018	0.4599	5	0.29	-0.4443	0.9449
-0.088	1.018	0.4599	4.5	0.261	-0.3748	0.8754
-0.088	1.018	0.4599	4	0.232	-0.3053	0.806
-0.088	1.018	0.4599	3.5	0.203	-0.2359	0.7365
-0.088	1.018	0.4599	3	0.174	-0.1664	0.667
-0.088	1.018	0.4599	2.5	0.145	-0.097	0.5976
-0.088	1.018	0.4599	2	0.116	-0.0275	0.5281
-0.088	1.018	0.4599	1.5	0.087	0.0419	0.4587
-0.088	1.018	0.4599	1	0.058	0.1114	0.3892
-0.088	1.018	0.4599	0.5	0.029	0.1808	0.3198

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 ค่าของปลอกแขนสีน้ำเงินแสดงค่าต่ำสุด สูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และจำนวนเท่าของ σ เพื่อใช้ในการหาค่า Threshold ต่ำสุดและสูงสุดของ b

b max	b min	b bar	จำนวนเท่า σ	σ_b	b left	b right
1.08	0.4264	0.3897	6	0.348	-1.5021	2.782
1.08	0.4264	0.3897	5.5	0.319	-1.3236	2.603
1.08	0.4264	0.3897	5	0.29	-1.1451	2.425
1.08	0.4264	0.3897	4.5	0.261	-0.9666	2.246
1.08	0.4264	0.3897	4	0.232	-0.7881	2.068
1.08	0.4264	0.3897	3.5	0.203	-0.6096	1.889
1.08	0.4264	0.3897	3	0.174	-0.4311	1.711
1.08	0.4264	0.3897	2.5	0.145	-0.2526	1.532
1.08	0.4264	0.3897	2	0.116	-0.0741	1.354
1.08	0.4264	0.3897	1.5	0.087	0.1044	1.175
1.08	0.4264	0.3897	1	0.058	0.2829	0.997
1.08	0.4264	0.3897	0.5	0.029	0.4614	0.818

สรุปการเลือกค่า Threshold ที่เหมาะสมของปลอกแขนแต่ละสี จากวิธีที่ได้อธิบายข้างต้นมีดังนี้

1) ปลอกแขนสีแดง

จากตารางที่ 4.4~4.6 เราจะพบว่าค่า Threshold ที่เหมาะสมสำหรับปลอกแขนสีแดง แยกตามค่า r g และ b ได้ดังนี้

ค่า Threshold ด้านซ้ายมือ (r left) ถึงด้านขวามือ (r right) ของภาพสี r อยู่ระหว่าง 0.2947 ถึง 1.1401

ค่า Threshold ด้านซ้ายมือ (g left) ถึงด้านขวามือ (g right) ของภาพสี g อยู่ระหว่าง 0.0005 ถึง 0.283

ค่า Threshold ด้านซ้ายมือ (b left) ถึงด้านขวามือ (b right) ของภาพสี b อยู่ระหว่าง 0.0007 ถึง 0.282

2) ปลอกแขนสีเขียว

ในทำนองเดียวกัน จากตารางที่ 4.7~4.9 เราจะพบว่าค่า Threshold ที่เหมาะสมสำหรับปลอกแขนสีแดง แยกค่า r g และ b ได้ดังนี้

ค่า Threshold ด้านซ้ายมือ (r left) ถึงด้านขวามือ (r right) ของภาพสี r อยู่ระหว่าง 0.5248 ถึง 0.8343

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Threshold ด้านซ้ายมือ (g left) ถึงด้านขวามือ (g right) ของภาพสี g อยู่ระหว่าง 0.3448 ถึง 0.575

ค่า Threshold ด้านซ้ายมือ (b left) ถึงด้านขวามือ (b right) ของภาพสี b อยู่ระหว่าง 0.1806 ถึง 0.594

3) ปลอกแขนสีน้ำเงิน

ในทำนองเดียวกัน จากตารางที่ 4.10~4.12 เราจะพบว่าค่า Threshold ที่เหมาะสมสำหรับปลอกแขนสีแดง แยกค่า r g และ b ได้ดังนี้

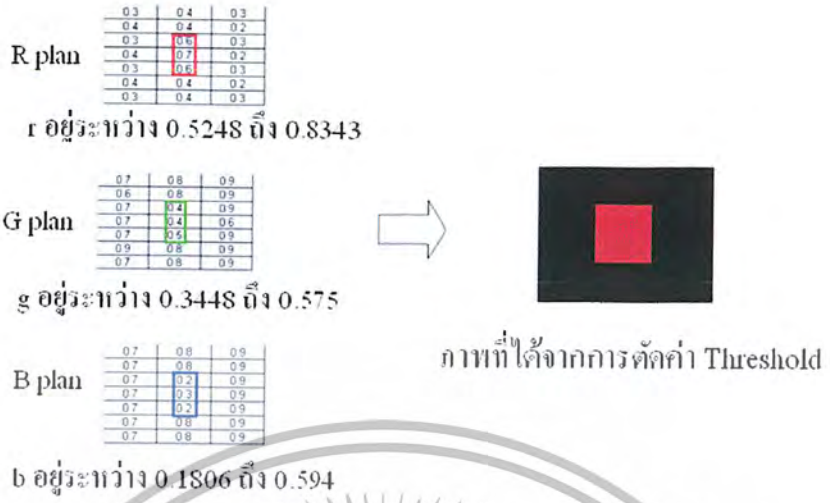
ค่า Threshold ด้านซ้ายมือ (r left) ถึงด้านขวามือ (r right) ของภาพสี r อยู่ระหว่าง 0.388 ถึง 0.7334

ค่า Threshold ด้านซ้ายมือ (g left) ถึงด้านขวามือ (g right) ของภาพสี g อยู่ระหว่าง 0.4966 ถึง 0.9582

ค่า Threshold ด้านซ้ายมือ (b left) ถึงด้านขวามือ (b right) ของภาพสี b อยู่ระหว่าง 0.4031 ถึง 1.078

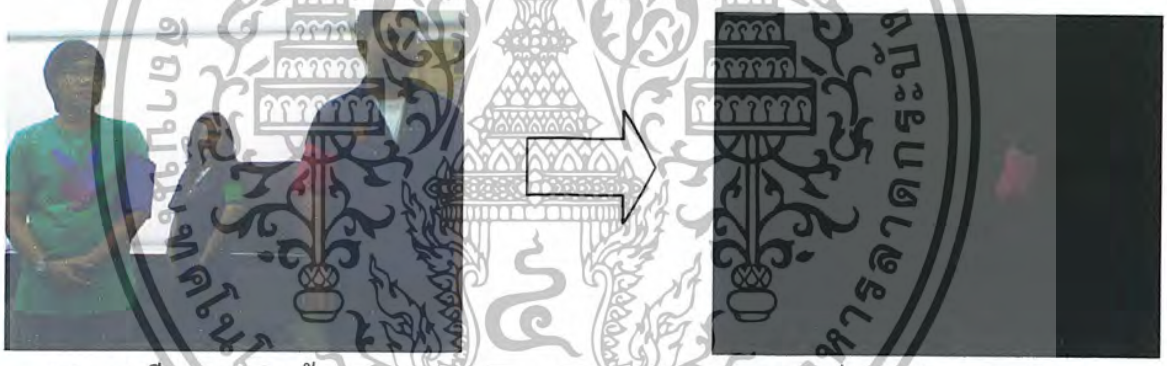
หลังจากที่เราได้ค่า Threshold ที่เหมาะสมกับปลอกแขนตัวอย่างแล้ว เราจะนำค่า Threshold ที่ได้นี้ไปใช้เป็นเกณฑ์ในการเลือกแสดงสีปลอกแขนที่ต้องการ ในที่นี้ผู้ใช้จะสามารถเลือกสีของปลอกแขนได้ว่าจะเป็นสีแดง สีเขียว หรือสีน้ำเงิน โดยเริ่มแรกเราจะหาค่า r g และ b จากการทำ Normalize ภาพสี RGB ก่อน หลังจากนั้นนำค่า r g และ b ที่ได้ไปตรวจเช็คหาสีของปลอกแขนที่กำหนด ตัวอย่างเช่น ถ้าผู้ใช้ต้องการแสดงปลอกแขนสีแดง โปรแกรมจะตรวจสอบค่า r g และ b ทั้ง 3 ค่า ณ ตำแหน่งใดตำแหน่งหนึ่งว่ามีค่าอยู่ระหว่างค่า Threshold ของปลอกแขนสีแดง คือ r อยู่ระหว่าง 0.5248 ถึง 0.8343 g อยู่ระหว่าง 0.3448 ถึง 0.575 และ b อยู่ระหว่าง 0.1806 ถึง 0.594 หรือไม่ ถ้าค่า r g และ b ทั้ง 3 ค่าที่ตำแหน่งนั้น มีค่าอยู่ระหว่างค่า Threshold ของปลอกแขนสีแดงนั้น โปรแกรมจะแสดงค่า RGB ออกมาซึ่งก็คือสีแดง แต่ถ้าไม่อยู่ระหว่างค่า Threshold ที่กำหนด จะแสดงเป็นสีดำ ดังรูปที่ 40 แสดงตัวอย่างการตัดค่า Threshold ของปลอกแขนสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.40 ตัวอย่างในการตัดค่า Threshold ของปลอกแขนสีแดง

การทดลองกับภาพจริง

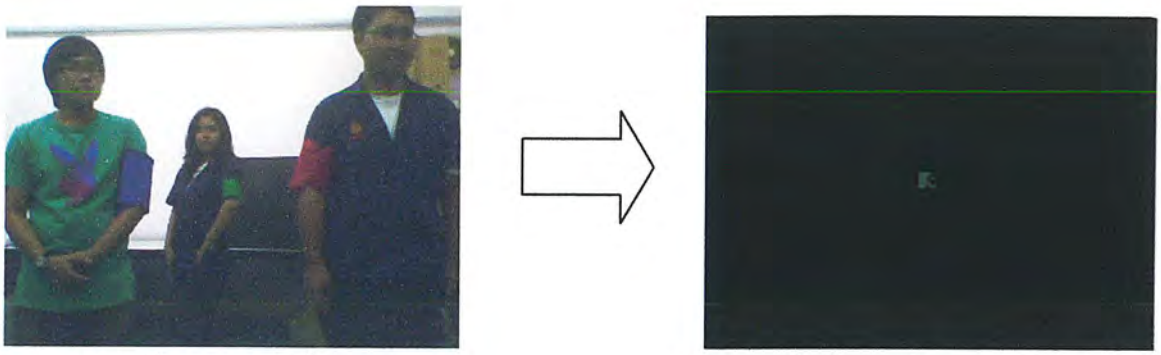


(ก) ภาพสี RGB จากกล้อง (ข) ภาพผลลัพธ์แสดงปลอกแขนสีแดง

ภาพที่ 4.41 ภาพที่ผ่านการตัดค่า Threshold โดยใช้ค่าปลอกแขนสีแดง

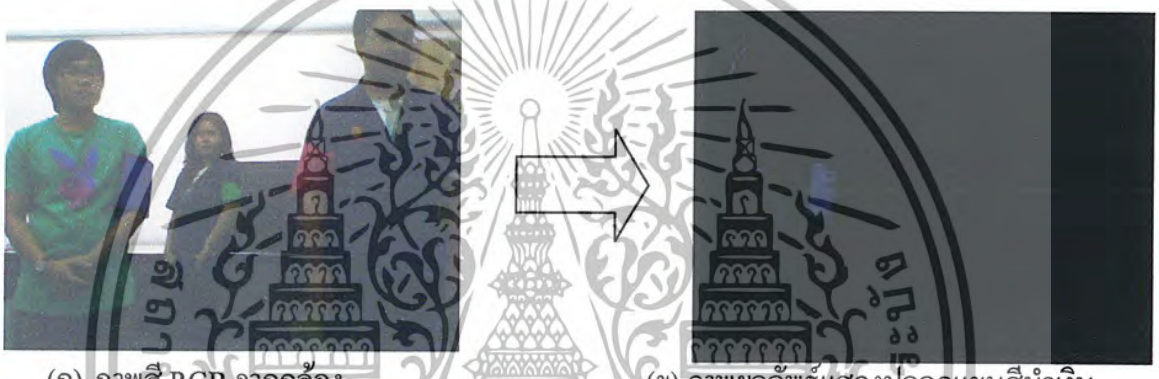
จากภาพที่ 4.41 เป็นการทดลองเลือกแสดงปลอกแขนสีแดง จากภาพ RGB ที่ถ่ายจากกล้อง เมื่อเรานำค่า Threshold ของปลอกแขนสีแดงที่ได้จากการทดลองข้างต้นมาใช้ในการเลือกแสดงปลอกแขนสีแดงจะได้ผลลัพธ์ดังภาพ 4.41(ข) จะเห็นได้ว่าสามารถมองเห็นปลอกแขนสีแดงได้อย่างชัดเจนในขณะที่คนที่สวมปลอกแขนสีแดงยืนอยู่ในตำแหน่งที่ใกล้กับกล้อง โดยอาจมีบางส่วนของปลอกแขนสีแดงขาดหายไปบ้างเนื่องจากในขั้นตอนการหาค่า Threshold เราได้ตัดปลอกแขนเพียง 4 ส่วนเท่านั้นและตัดมาเพียง 40 ส่วนดังที่ได้อธิบายไว้ข้างต้น ดังนั้น ข้อมูลอาจจะไม่เพียงพอ จึงได้ค่าสีของปลอกแขนที่ยังไม่ครอบคลุมทั่วทั้งปลอกแขน แต่ก็ยังสามารถมองเห็นปลอกแขนสีแดงได้เกือบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ภาพสี RGB จากกล้อง (ข) ภาพผลลัพธ์แสดงปลอกแขนสีเขียว

ภาพที่ 4.42 ภาพที่ผ่านการตัดค่า Threshold โดยใช้ค่าปลอกแขนสีเขียว



(ก) ภาพสี RGB จากกล้อง (ข) ภาพผลลัพธ์แสดงปลอกแขนสีน้ำเงิน

ภาพที่ 4.43 ภาพที่ผ่านการตัดค่า Threshold โดยใช้ค่าปลอกแขนสีน้ำเงิน

ในทำนองเดียวกัน เมื่อผู้ใช้ต้องการแสดงปลอกแขนสีเขียวและสีน้ำเงิน ก็จะสามารถเลือกแสดงได้ดังภาพที่ 4.42 (ข) และ 4.43 (ข) ซึ่งแสดงให้เห็นว่าไม่ว่าปลอกแขนจะมีขนาดใหญ่หรือเล็กก็ตาม โปรแกรมก็สามารถแสดงตำแหน่งและสีของปลอกแขนได้อย่างถูกต้อง แม้ว่าบางส่วนของปลอกแขนอาจมองไม่เห็น ก็เป็นเพราะเหตุผลที่ได้อธิบายไว้ข้างต้น

4.4 การหาเส้นทางการเคลื่อนที่ของคนที่อยู่ในภาพ

ในการติดตามคนที่อยู่ในภาพนั้น เราจะมีการเก็บตำแหน่งของคนที่อยู่ในภาพโดยเก็บอยู่ในรูปของจุดศูนย์กลางของคนๆนั้น พร้อมทั้งลากเส้นตรงเชื่อมต่อระหว่างจุดศูนย์กลางของคนที่กำลังเคลื่อนที่อยู่ในภาพกับจุดศูนย์กลางของตำแหน่งเดิม เพื่อเป็นการบันทึกข้อมูลเก็บไว้ว่าคนเคลื่อนที่ไปทิศทางใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการหาเส้นทางการเคลื่อนที่ของคนที่อยู่ในภาพ

เริ่มแรกเราจะทำการหาจุดศูนย์กลางของคนที่อยู่ในภาพก่อน โดยการตีกรอบล้อมรอบคนที่อยู่ในภาพแล้วหาจุดศูนย์กลางของคนที่อยู่ในภาพดังหัวข้อที่ 4.4 (เรื่องการหาจุดศูนย์กลางของคนที่อยู่ในภาพ) จากนั้นเราจะทำการเก็บข้อมูลจุดศูนย์กลางของภาพไว้ในตัวแปร ดังภาพที่ 4.44 (ข) แสดงการตีกรอบล้อมรอบคนที่อยู่ในภาพพร้อมทั้งหาจุดศูนย์กลาง เมื่อมีคนเดินเข้ามาในภาพ

(ก) ภาพสี RGB จากกล้อง (ข) ภาพการหาจุดศูนย์กลางของคน

ภาพที่ 4.44 การหาจุดศูนย์กลางของคนที่อยู่ในภาพ

(ก) ภาพสี RGB จากกล้อง (ข) ภาพการหาจุดศูนย์กลางของคน

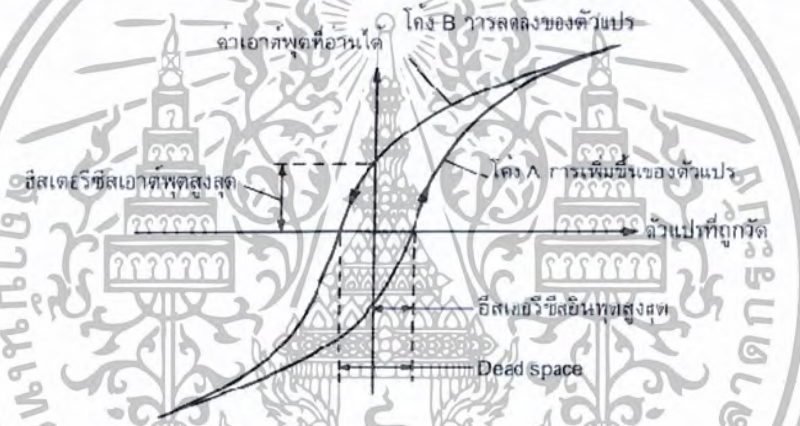
(ค) ภาพแสดงการลากเส้นตรงเชื่อมต่อระหว่างจุดศูนย์กลางใหม่กับจุดศูนย์กลางเดิม

ภาพที่ 4.45 การหาจุดศูนย์กลางในตำแหน่งใหม่พร้อมทั้งลากเส้นตรงเชื่อมต่อระหว่างจุดศูนย์กลางใหม่กับจุดศูนย์กลางเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การหาค่า Hysteresis ของเซอร์โวมอเตอร์

ฮีสเตอรีซีส (Hysteresis) เป็นคุณลักษณะเฉพาะตัวของอุปกรณ์ทางกลทุกชนิด (Instrument) ที่ไม่สามารถให้ค่าเอาท์พุทซ้ำที่ค่าเดิมได้ เมื่อเปลี่ยนแปลงค่าอินพุทในทิศทางเพิ่มค่าขึ้นและลดค่าลง จากรูปที่ 4.5 เป็นกราฟแสดงความสัมพันธ์ระหว่างตัวแปรที่ถูกวัดและค่าเอาท์พุทที่อ่านได้ ซึ่งค่าเอาท์พุทที่อ่านได้เป็นสัดส่วนโดยตรงกับตัวแปรที่ถูกวัด กล่าวคือเมื่อตัวแปรที่ถูกวัดมีค่าเพิ่มขึ้น ค่าเอาท์พุทที่อ่านได้จะมีค่าเพิ่มขึ้นด้วย อย่างไรก็ตาม ในภาพที่ 4.46 จะพบว่าเมื่อเพิ่มค่าตัวแปรที่ถูกวัดจากค่าต่ำสุดจนกระทั่งถึงค่าสูงสุดจะได้เส้นโค้ง A และเมื่อลดค่าตัวแปรที่ถูกวัดลงจากค่าสูงสุดกลับไปที่ค่าต่ำสุดจะได้เส้นโค้ง B ซึ่งเส้นโค้งทั้งสองไม่สามารถทับกันได้สนิทเกิดเป็นช่องว่างขึ้นที่เรียกว่า "Dead Space" การเกิดขึ้นในลักษณะเช่นนี้ถูกเรียกว่า "ฮีสเตอรีซีส" การเกิดฮีสเตอรีซีสนี้จะไม่เกิดเฉพาะอุปกรณ์ทางกลเท่านั้น แต่จะเกิดกับอุปกรณ์ทางไฟฟ้าอีกด้วยเช่น มอเตอร์ไฟฟ้า เป็นต้น



ภาพที่ 4.46 กราฟแสดงการเกิดฮีสเตอรีซีส

ในการติดตามคนที่กำลังเคลื่อนที่ในภาพ เราจะใช้เซอร์โวมอเตอร์เป็นตัวควบคุมการหมุนของกล้องไปที่ตำแหน่งต่างๆ เริ่มต้นด้วยการส่งค่าความกว้างของสัญญาณพัลส์ที่ 1500 µs เพื่อให้เซอร์โวมอเตอร์หมุนไปที่ตำแหน่งตรงกลาง จากนั้นหมุนเซอร์โวมอเตอร์ที่ไปตำแหน่งต่างๆ ไม่ว่าจะเป็นทางด้านซ้ายมือหรือขวามือ แล้วหมุนเซอร์โวมอเตอร์กลับที่ตำแหน่งตรงกลางอีกครั้งจะเห็นได้ว่าทุกครั้งที่เซอร์โวมอเตอร์หมุนกลับมาอยู่ที่ตำแหน่งตรงกลาง จุดศูนย์กลางของวัตถุจะมีค่าเปลี่ยนแปลงไปไม่ทับที่ตำแหน่งเดิม ซึ่งการหาคุณลักษณะของเซอร์โวมอเตอร์ที่ใช้ในการทำโครงงานนี้ สิ่งที่ต้องการตรวจสอบก็คือค่าฮีสเตอรีซีสของเซอร์โวมอเตอร์เพื่อใช้ในการวิเคราะห์ว่าฮิสเตอรีซีสที่เกิดขึ้นมีค่าอยู่ในขอบเขตที่ยอมรับได้หรือไม่ ควรมีวิธีการแก้ไขอย่างไร นอกจากนี้การหาค่าจำนวนพัลส์ต่อพิกเซลจะทำให้ทราบคุณลักษณะของเซอร์โวมอเตอร์มอเตอร์ว่ามีความไวในการหมุนกล้องเป็นค่าเท่าใดบ้าง

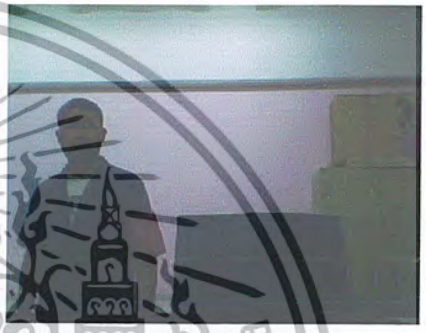
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ขั้นตอนการประมวลผลภาพ

ขั้นตอนที่ 1 ถ่ายภาพเพื่อนำไปประมวลผล

เปิดการใช้งานวีดีโอโดยใช้คำสั่ง vid=videoinput('winvideo',1,'YUY2_320x240') โดยกำหนดขนาดของวีดีโอเท่ากับ 320 x 240 แสดงภาพวีดีโอที่ได้โดยใช้คำสั่ง preview(vid) หลังจากนั้นถ่ายภาพครั้งที่ 1 และ 2 โดยใช้คำสั่ง pic12=getsnapshot(vid); และpic2=getsnapshot(vid) โดยภาพที่ได้เราจะใช้คำสั่ง set(vid, 'ReturnedColorspace', 'RGB'); เพื่อให้ได้ภาพสีชนิด RGB ดังภาพที่ 4.1 และ 4.2 ตามลำดับ

ภาพที่ 4.1 ภาพที่ถ่ายครั้งที่ 1 ภาพพื้นหลัง



ภาพที่ 4.2 ภาพที่ถ่ายครั้งที่ 2 มีคนอยู่ในภาพ

```
1  clear all;                                  % ล้างตัวแปรทั้งหมดในหน่วยความจำ
2  close all;                                  % ล้างรูปทั้งหมดในหน่วยความจำ
%%%%%%%%%%%%ส่วนของการถ่ายภาพเพื่อนำไปประมวลผล%%%%%%%%%%%%
3  vid=videoinput('winvideo',1,'YUY2_320x240') % นำภาพวิดีโอเข้ามากำหนดเป็น 320 x 240
                                               พิกเซล
4  preview(vid)                                % แสดงภาพวีดีโอ
5  set(vid, 'ReturnedColorspace', 'RGB');      % ทำให้ภาพที่ได้เป็นภาพชนิด RGB
6  a=50;g=1;h=1;cgx=0;cgy=0;                   % กำหนดค่าตัวแปรให้มีค่าต่างๆ
7  while a>0                                   % while loop
8  pic1=getsnapshot(vid);                      % ถ่ายรูปแล้วนำมาเก็บไว้ในตัวแปร pic1
9  set(vid, 'ReturnedColorspace', 'RGB');      % ทำให้ภาพที่ได้เป็นภาพชนิด RGB
10 f1=rgb2gray(pic1);                          % แปลงภาพ pic12 เป็นภาพระดับสีเทาแล้ว
                                               เก็บไว้ในตัวแปร f1
11 pic2=getsnapshot(vid);                      % ถ่ายรูปแล้วนำมาเก็บไว้ในตัวแปร pic2
12 f2=rgb2gray(pic2);                          % แปลงภาพ pic12 เป็นภาพระดับสีเทาแล้ว
                                               เก็บไว้ในตัวแปร f1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 การลบภาพเพื่อหาคนที่อยู่ในภาพ

นำภาพสี RGB ที่ได้มาแปลงเป็นภาพระดับสีเทาโดยใช้คำสั่ง f1=rgb2gray(pic1) และ f2=rgb2gray(pic2) ดังภาพที่ 4.3 และ 4.4 ตามลำดับ หลังจากนั้นนำภาพทั้งสองมาลบกันจะได้บุคคลที่อยู่ในภาพดังรูปที่ 4.5

ภาพที่ 4.3 ภาพพื้นหลัง (ระดับสีเทา)

ภาพที่ 4.4 ภาพที่มีคนอยู่ในภาพ (ระดับสีเทา)

ภาพที่ 4.5 ภาพที่ได้จากการลบกัน

แปลงภาพผลลัพธ์ที่ได้เป็นภาพขาวดำโดยใช้ฟังก์ชันของโปรแกรม MATLAB คือ level = graythresh(f3); f5 = im2bw(f3,level); แล้วทำการตีกรอบโดยใช้ bwlabel จะได้ดังภาพที่ 4.6

ภาพที่ 4.6 ภาพที่แปลงเป็นภาพขาวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%%%%%%%%%%%%ส่วนของการลบภาพเพื่อหาบุคคลในภาพ%%%%%%%%%%%%%%%
13  f3=(f1-f2);                    % นำค่าที่อยู่ในตัวแปร f1 ลบกับ f2 แล้วเก็บไว้ใน f3
14  [m n]=size(pic);               % กำหนดขนาดภาพ pic1
15  level = graythresh(f3);        % หาค่า threshold โดยใช้ฟังก์ชัน graythresh
16  f5 = im2bw(f3,level);          % แปลงภาพ f3 เป็นภาพขาวดำ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ขั้นตอนที่ 3 การหาปลอกแขนสีต่างๆ

จากโปรแกรมที่เขียน กำหนดให้ค่าของตัวแปร a1 ดังนี้ สีแดงมีค่าเท่ากับ 1 สีเขียวมีค่าเท่ากับ 2 และสีน้ำเงินมีค่าเท่ากับ 3 ทำการดึงค่าของเพลนสีแดง เขียวและน้ำเงินจากภาพสี RGB ที่ถ่ายครั้งที่ 2 โดยใช้คำสั่ง R=pic2(:,:,1) G=pic2(:,:,2) และ B=pic2(:,:,3) ตามลำดับ นำค่าของแต่ละเพลนที่ได้มาทำ Normalize แล้วนำมาแยกเก็บไว้ในแต่ละตัวแปร โดยเพลนสีแดงเก็บไว้ในตัวแปร r สีเขียวเก็บไว้ในตัวแปร g และสีน้ำเงินเก็บไว้ในตัวแปร b ตามลำดับ ใช้คำสั่ง r(i,j)=R(i,j)/(R(i,j)+G(i,j)+B(i,j)); g(i,j)=G(i,j)/(R(i,j)+G(i,j)+B(i,j)); b(i,j)=B(i,j)/(R(i,j)+G(i,j)+B(i,j)); หลังจากนั้นนำค่าที่ได้มาเข้าโปรแกรมคัดแยกสีปลอกแขน เช่น เลือกที่จะติดตามคนที่ใส่ปลอกแขนสีแดง เราจะกำหนดให้ค่า a1 มีค่าเท่ากับ 1 โปรแกรมจะทำการคัดแยกเฉพาะปลอกแขนสีแดง (บรรทัดที่ 17-53) แล้วแปลงเป็นภาพขาวดำ ก็จะได้ส่วนที่เป็นเฉพาะปลอกแขนสีแดงออกมาเป็นภาพสีขาวดังรูปที่ 4.8

ภาพที่ 4.7 ภาพที่ถ่ายครั้งที่ 2 มีคนอยู่ในภาพ

ภาพที่ 4.8 ภาพที่ผ่านโปรแกรมหาปลอกแขน

```
%%%%%%%%%%%%%%%%%%ส่วนของการหาปลอกแขน%%%%%%%%%%%%%%%%%%%%
17  a1=1; %red                     % เลือกสีของปลอกแขนที่จะติดตามโดยมี
                                     ทั้งหมด 3 สี
18  R=pic2(:,:,1);                 % ดึงเพลนสีแดงแล้วไปเก็บไว้ที่ R
19  G=pic2(:,:,2);                 % ดึงเพลนสีเขียวแล้วไปเก็บไว้ที่ G
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

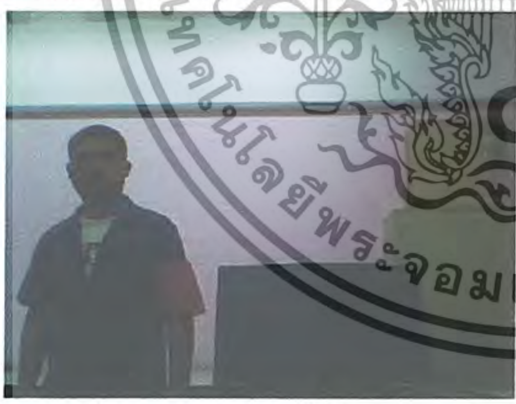
```
B=pic2(:,:,3);                          % ดึงเพลนสีน้ำเงินแล้วไปเก็บไว้ที่ B
R=double(R);                            % แปลงค่าในตัวแปร R ให้เป็นชนิด double
G=double(G);                            % แปลงค่าในตัวแปร G ให้เป็นชนิด double
B=double(B);                            % แปลงค่าในตัวแปร B ให้เป็นชนิด double
[m,n]=size(R);                          % กำหนดค่า m และ n ให้มีค่าเท่ากับ R
for i=1:m                               % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง m
 for j=1:n                              % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง n
    r(i,j)=R(i,j)/(R(i,j)+G(i,j)+B(i,j));   % ทำการ Normalize ค่าของระดับสีแดง
    g(i,j)=G(i,j)/(R(i,j)+G(i,j)+B(i,j));   % ทำการ Normalize ค่าของระดับสีเขียว
    b(i,j)=B(i,j)/(R(i,j)+G(i,j)+B(i,j));   % ทำการ Normalize ค่าของระดับสีน้ำเงิน
 end                                    % จบการทำงานแบบ for loop
end                                     % จบการทำงานแบบ for loop
[o,p]=size(R);                          % กำหนดค่า o และ p ให้มีค่าเท่ากับขนาดของ
R
pic1(o,p)=0;                            % กำหนดค่า pic1 ตำแหน่งที่ (o,p) เท่ากับ 0
for i=1:m                               % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง m
  for j=1:n                             % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง n
    if a1==1                            % กำหนดเงื่อนไขให้ a1 เท่ากับ 1
if r(i,j)>= 0.2947 && r(i,j)<=1.1401 && g(i,j)>= 0.0005 && g(i,j)<=0.283 && b(i,j)>= -0.0007 &&
b(i,j)<=0.282
                                        % กำหนดเงื่อนไขโดยใช้ค่า Threshold
      pic1(i,j)=1;                      % ให้ค่า pic1 ตำแหน่งที่ (i,j) มีค่าเท่ากับ 1
    end                                 % จบการทำงานคำสั่ง if
    end                                 % จบการทำงานคำสั่ง if
    if a1==2                            % กำหนดเงื่อนไขให้ a1 เท่ากับ 2
if r(i,j)>=-0.0825 && r(i,j)<=0.3834 && g(i,j)>=0.3448 && g(i,j)<=1.0218 && b(i,j)>=0.3056 &&
b(i,j)<=0.429
                                        % กำหนดเงื่อนไขโดยใช้ค่า Threshold ของ
                                        ปลอกแขนสีเขียว
      pic1(i,j)=1;                      % ให้ค่า pic1 ตำแหน่งที่ (i,j) มีค่าเท่ากับ 1
    end                                 % จบการทำงานคำสั่ง if
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
45      if a1==3                                % กำหนดเงื่อนไขให้ a1 เท่ากับ 3
46 if r(i,j)>=-0.0167 && r(i,j)<=0.7334 && g(i,j)>=-0.4966 && g(i,j)<=0.8976 && b(i,j)>=0.4031 &&
b(i,j)<=1.078
                                                % กำหนดเงื่อนไข โดยใช้ค่า Threshold ของปลอกแขนสีน้ำเงิน
47          pic1(i,j)=1;                        % ให้ค่า pic1 ตำแหน่งที่ (i,j) มีค่าเท่ากับ 1
48      end                                     % จบการทำงานคำสั่ง if
49      end                                     % จบการทำงานคำสั่ง if
50   end                                        % จบการทำงานแบบ for loop
51  end                                         % จบการทำงานแบบ for loop
52  pic1 = medfilt2(pic1 ,[9 9]);               % ใช้ตัวกรองสัญญาณรบกวนแบบ median filter ขนาด 9x9
53  end                                         % จบการทำงานแบบ for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ขั้นตอนที่ 4 การหาสีผิวของคนที่อยู่ในภาพ

จากการที่เรากำหนดโปรแกรมหาสีผิวของคนที่อยู่ในภาพ (บรรทัดที่ 54 – 64) เมื่อมีคนผ่านเข้ามาโปรแกรมจะทำการตรวจจับเฉพาะสีผิวของคนตามที่กำหนดจะได้ภาพดังรูปที่ 4.10



ภาพที่ 4.9 ภาพที่ถ่ายครั้งที่ 2 มีคนอยู่ในภาพ

ภาพที่ 4.10 ภาพที่ผ่านโปรแกรมหาสีผิว

```
%%%%%%%%%%%%%%%การหาสีผิวของบุคคลที่อยู่ในภาพ%%%%%%%%%%%%%%%
54 pic(m,n)=0;                                  % กำหนดค่า pic ตำแหน่งที่ (m,n) มีค่าเท่ากับ 0
55 for i=1:m                                    % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง m
56    for j=1:n                                 % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง n
57    if r(i,j)> 0.3464 && r(i,j)< 0.6857 && g(i,j)> 0.2659 && g(i,j)<0.3768
                                                % กำหนดเงื่อนไขโดยใช้ค่า Threshold ของสีผิว
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
58          pic(i,j)=1;              % กำหนดค่า pic ตำแหน่งที่ (m,n) มีค่าเท่ากับ 1
59       end                         % จบการทำงานคำสั่ง if
60    end                            % จบการทำงานแบบ for loop
61 end                               % จบการทำงานแบบ for loop
62 f5=f5+pic;                        % นำค่า f5 บวกกับ pic แล้วเก็บไว้ในตัวแปร f5
63 f5 = medfilt2(f5,[9 9]);          % ใช้ตัวกรองสัญญาณรบกวนแบบ median filter ขนาด
                                     9x9
64 imshow(pic2)                      % แสดงรูปภาพ (pic2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ขั้นตอนที่ 5 การเลือกตีกรอบบุคคลที่อยู่ในภาพ

เราจะสั่งให้โปรแกรมตีกรอบบุคคลที่อยู่ในภาพโดยมีเงื่อนไขดังต่อไปนี้

1. วัตถุที่ตีกรอบนั้นต้องใส่ปลอกแขนตามสีที่กำหนด
2. วัตถุที่จะตีกรอบนั้นต้องมีสีผิว

ดังนั้นเราจะนำภาพที่ 4.8 ภาพที่ผ่านโปรแกรมหาปลอกแขนมาเข้าโปรแกรมตีกรอบ และนำรูปที่ 4.10 ภาพที่ผ่านโปรแกรมหาสีผิวมาเข้าโปรแกรมตีกรอบตามบรรทัดที่ 65-107 เพื่อหาตำแหน่งของจุดศูนย์กลางของปลอกแขนและสีผิวดังภาพที่ 4.11 และ 4.12 หลังจากนั้นจะทำการตรวจสอบว่าจุดศูนย์กลางของปลอกแขนและจุดศูนย์กลางของสีผิวอยู่ภายในขอบเขตของคนที่อยู่ในภาพตามภาพที่ 4.6 ถ้าอยู่จะสั่งให้โปรแกรมตีกรอบล้อมรอบคนพร้อมทั้งหาจุดศูนย์กลางของคนดังภาพที่ 4.13

ภาพที่ 4.11 ภาพการตีกรอบปลอกแขน

ภาพที่ 4.12 ภาพการตีกรอบสีผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.13 ตีกรอบบุคคลที่ใส่ปลอกแขนสีแดงและมีผิวหนัง

```
%%%%%%%%%%%%%%%%การเลือกตีกรอบบุคคลที่อยู่ในภาพ%%%%%%%%%%%%%%%%
65    cgy11=0;                          % กำหนดให้ cgy11 มีค่าเท่ากับ 0
66    cgx11=0;                          % กำหนดให้ cgx11 มีค่าเท่ากับ 0
67  [L,num]=bwlabel(pic1);              % ใช้ฟังก์ชันคำสั่ง bwlabel โดยที่ตรวจสอบ
                                        pixel ข้างเคียง 4 ทิศทาง
68  for i=1:num                         % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง num
69      [r,c]=find(L==i);               % ค้นหาค่า row และ column ของวัตถุ
70      max(i)=0; min(i)=255; max1(i)=0; min1(i)=255;  % เก็บตัวแปร max และmin เพื่อใช้ใน
                                                       การหาหาค่า max และ min ของ ค่า
                                                       row และ column
71      u=size(r);                      % กำหนด u เท่ากับขนาดของ r
72      for j=1:u                       % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
73      if c(j)>max(i)                  % กำหนดเงื่อนไขให้ a1 เท่ากับ 1
74        max(i)=c(j);                  % กำหนดให้ max(i) มีค่าเท่ากับ c(j)
75      end                             % จบการทำงานคำสั่ง if
76      if c(j)<min(i)                  % กำหนดเงื่อนไขให้ c(j) น้อยกว่า min(i)
77         min(i)=c(j);                 % กำหนดให้ min(i) มีค่าเท่ากับ c(j)
78      end                             % จบการทำงานคำสั่ง if
79    end                               % จบการทำงานแบบ for loop
80    for j=1:u                         % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
81      if r(j)>max1(i)                 % กำหนดเงื่อนไขให้ r เท่ากับ max1
82         max1(i)=r(j);                % กำหนดให้ max1 เท่ากับ r
83      end                             % จบการทำงาน if
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
84      if r(j)<min1(i)                    % กำหนดเงื่อนไขให้ r น้อยกว่า min1
85          min1(i)=r(j);                  % ให้ min1 เท่ากับ r
86      end                                % จบการทำงาน if
87    end                                  % จบการทำงานแบบ for loop
88    x11=[ min1(i) min1(i) max1(i) max1(i) min1(i)];     % กำหนดค่า x11
89    y11=[ min(i)  max(i)  max(i)  min(i)  min(i)];      % กำหนดค่า y11
90    xmin11 = min(i);                   % กำหนดให้ xmin11 เท่ากับ min
91    xmax11 = max(i);                   % กำหนดให้ xmax11 เท่ากับ min
92    ymin11 = min1(i) ;                 % กำหนดให้ ymin11 เท่ากับ min1
93    ymax11 = max1(i);                  % กำหนดให้ ymax11 เท่ากับ min1
94    cgy11=((ymax11-ymin11)/2)+ymin11;  % หาจุด centroid y
95    cgx11=((xmax11-xmin11)/2)+xmin11;  % หาจุด centroid x
96  end                                  % จบการทำงาน for loop
97    cgy12=0;                           % กำหนดให้ cgy12 เท่ากับ 0
98    cgx12=0;                           % กำหนดให้ cgx12 เท่ากับ 0
99    [L,num]=bwlabel(f5);               % ใช้ bwlabel กับ f5
100   for i=1:num                        % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง num
101     [r,c]=find(L==i);                % ใช้คำสั่ง find
102     max(i)=0; min(i)=255; max1(i)=0; min1(i)=255;      % กำหนดค่าต่างๆ
103     u=size(r);                       % ให้ u เท่ากับขนาดของ r
104     for j=1:u                        % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
105       if c(j)>max(i)                 % กำหนดเงื่อนไขให้ c มากกว่า max
106         max(i)=c(j);                 % ให้ max เท่ากับ c
107       end                            % จบการทำงาน for loop
108       if c(j)<min(i)                 % กำหนดเงื่อนไขให้ c น้อยกว่า min
109         min(i)=c(j);                 % ให้ min เท่ากับ c
110       end                            % จบการทำงาน for loop
111     end                              % จบการทำงาน for loop
112     for j=1:u                        % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
113       if r(j)>max1(i)                % กำหนดเงื่อนไขให้ r มากกว่า max1
114         max1(i)=r(j);                % ให้ max1 เท่ากับ r
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
115     end                                          % จบการทำงานแบบ for loop
116     if r(j)<min1(i)                              % กำหนดเงื่อนไขให้ r น้อยกว่า min1
117        min1(i)=r(j);                             % ให้ min1 เท่ากับ r
118     end                                          % จบการทำงาน for loop
119  end                                             % จบการทำงาน for loop
120  x11=[ min1(i) min1(i) max1(i) max1(i) min1(i)];    % กำหนดค่าตัวแปรต่างๆ
121  y11=[ min(i)  max(i)  max(i)  min(i)  min(i)];     % กำหนดค่าตัวแปรต่างๆ
122  xmin11 = min(i);                                % ให้ xmin11 เท่ากับ min
123  xmax11 = max(i);                                % ให้ xmax11 เท่ากับ max
124  ymin11 = min1(i);                               % ให้ ymin11 เท่ากับ min1
125  ymax11 = max1(i);                               % ให้ ymax11 เท่ากับ max1
126  if cgy11>=ymin11 && cgy11<=ymax11 && cgx11>=xmin11&&cgx11<=xmax11
                                                     % กำหนดค่าตัวแปรต่างๆ
127  hold on                                         % ค้างค่าไว้
128  plot (y11,x11,'-r')                             % พล็อตกราฟ
129  end                                             % จบการทำงาน for loop
130  cgy12=((ymax11-ymin11)/2)+ymin11; %centroid y   % หาค่าจุด cgy
131  cgx12=((xmax11-xmin11)/2)+xmin11; %centroid x   % หาค่าจุด cgx
132  if cgy11>=ymin11 && cgy11<=ymax11 && cgx11>=xmin11&&cgx11<=xmax11
                                                     % กำหนดค่าตัวแปรต่างๆ
133  plot(cgx12,cgy12,'-r.');                        % พล็อตกราฟ
134  end                                             % จบการทำงาน for loop
135 end                                              % จบการทำงาน for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ขั้นตอนที่ 6 การส่งค่าไปควบคุมเซอร์โวมอเตอร์ให้หมุนติดตามคนที่อยู่ในภาพ

การหาผลต่างระหว่างจุดศูนย์กลางของคนและจุดศูนย์กลางของภาพ เมื่อเราได้จุดศูนย์กลางของคนจากขั้นตอนที่ 5 เราจะนำมาลบกับจุดศูนย์กลางของภาพ เพื่อหาระยะห่างระหว่างจุดศูนย์กลางของคนและจุดศูนย์กลางของภาพ โดยจุดศูนย์กลางของภาพอยู่ตำแหน่งที่ (160,120) ดังนั้นเราจะนำค่าจุดศูนย์กลางของคนมาลบด้วยจุดศูนย์กลางของภาพทั้งแกน x และ แกน y เพื่อที่จะนำไปสั่งให้เซอร์โวมอเตอร์หมุนติดตามบุคคลได้ดังภาพที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.14 การหาผลต่างระหว่างจุดศูนย์กลางของคนและจุดศูนย์กลางของภาพ

```
%%%%%%%%หาผลต่างระหว่างจุดศูนย์กลางของคนและจุดศูนย์กลางของภาพ%%%%%%%%
136  if cgy11>0&&cgx11>0                    % กำหนดเงื่อนไข cgy มากกว่า 0
137  y2=cgy11-160;                          % หาผลต่างจุด cg
138  x2=cgx11-120;                          % หาผลต่างจุด cg
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ขั้นตอนที่ 7 การแปลงระยะห่างของคนเป็นจำนวนพัลส์แล้วส่งไปให้เซอร์โวมอเตอร์หมุนติดตามคนที่อยู่ในภาพ

จากการทดลองทำให้ทราบว่า แกน x ได้ค่า 0.84 พิกเซล/พัลส์ และแกน y ได้ค่า 0.9 พิกเซล/พัลส์ แล้วแปลงค่าส่งไปยังเซอร์โวมอเตอร์เพื่อหมุนติดตามคนที่ต้องการติดตาม

```
%%%%%%%%%%การแปลงระยะห่างของคนเป็นจำนวนพัลส์%%%%%%%%%%%%%%%%%%%%%%%%
139 pulsex1=x2/0.84;                      %เปลี่ยนเป็นค่าพัลส์
140 pulsey1=y2/0.9;                       %เปลี่ยนเป็นค่าพัลส์
141 pulsex2=int32(pulsex1)                %เปลี่ยนเป็นค่าพัลส์
142 pulsey2=int32(pulsey1)                %เปลี่ยนเป็นค่าพัลส์
143 pulsex3=pulsex2+1500                  %เปลี่ยนเป็นค่าพัลส์
144 pulsey3=pulsey2+1500                  %เปลี่ยนเป็นค่าพัลส์
145 x3=pulsex3*10000;                     %เปลี่ยนเป็นค่าพัลส์
146 sendpulse=x3+pulsey3;                 % ส่งค่าพัลส์
147 h=double(sendpulse)                   % แปลงเป็นตัวแปรชนิด double
148  s=serial('COM3');                    % กำหนดใช้คอมพอร์ต 3
149 set(s,'BaudRate',9600,'DataBits',8,'parity','none','stopBits',1,'FlowControl','none')
                                          % กำหนดค่าต่างๆในการส่ง
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการทดลอง

5.1 ผลการทดลองทางด้านฮาร์ดแวร์

การทดลองทางด้านฮาร์ดแวร์นั้นจะประกอบไปด้วย 2 การทดลองคือ การทดลองการติดต่อสื่อสารและการบวกลบจำนวนทางคณิตศาสตร์ เพื่อทดลองการติดต่อสื่อสารระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ PIC และเพื่อตรวจคำตอบจากการส่งเลขสองจำนวนไปบวกกัน ซึ่งไมโครคอนโทรลเลอร์ PIC จะส่งผลลัพธ์กลับมาแสดงที่โปรแกรม Terminal ว่าผลลัพธ์ถูกต้องหรือไม่ และการทดลองควบคุมเซอร์โวมอเตอร์ 4 ตัว เพื่อควบคุมกล้องเว็บแคม 2 ตัว โดยใช้ไมโครคอนโทรลเลอร์ PIC เป็นตัวสร้างสัญญาณพัลส์แบบ Timer ซึ่งการควบคุมเซอร์โวมอเตอร์จะผ่านทางโปรแกรม MATLAB

5.1.1 การทดลองการติดต่อสื่อสารและการบวกลบจำนวนทางคณิตศาสตร์

ในการทดลองนี้เป็นการทดลองติดต่อสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ PIC พร้อมทั้งมีการบวกลบจำนวนทางคณิตศาสตร์กับจำนวนตัวเลขที่ส่งไปให้กับไมโครคอนโทรลเลอร์ PIC จากนั้นส่งค่าผลลัพธ์กลับมาแสดงผลที่คอมพิวเตอร์อีกครั้ง การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ PIC18F4620 จะผ่านทางพอร์ตอนุกรม ส่วนโปรแกรมบนคอมพิวเตอร์ที่ใช้ติดต่อกับไมโครคอนโทรลเลอร์ PIC18F4620 จะเป็นโปรแกรม Terminal จุดมุ่งหมายของการทดลองนี้ก็เพื่อตรวจสอบดูว่าไมโครคอนโทรลเลอร์ PIC18F4620 สามารถรับส่งข้อความหรือตัวเลขที่ส่งจากคอมพิวเตอร์ได้ถูกต้องหรือไม่ และคำนวณทางคณิตศาสตร์ได้ถูกต้องหรือไม่ ในการเขียนโปรแกรมที่ไมโครคอนโทรลเลอร์ PIC18F4620 มีขั้นตอนดังนี้ ขั้นตอนแรกกำหนดค่า Configuration เลือกการสื่อสารผ่านทางพอร์ต RS-232 และกำหนดให้ความเร็วในการสื่อสารข้อมูลเท่ากับ 9600 bit/second โดยการเก็บข้อมูลที่เป็นตัวอักษรในรูปของ ASCII Code ชุดแรกไว้ที่ตัวแปร A ที่มีจำนวนสมาชิก 5 ตัว ชุดที่สองไว้ที่ตัวแปร B ที่มีจำนวนสมาชิก 5 ตัวเช่นกัน จากนั้นนำสมาชิกของ A โดยสมาชิกตัวที่ศูนย์นำมาคูณ 10000 เพื่อให้สมาชิกตัวที่ศูนย์เป็นหลักหมื่น สมาชิกตัวที่หนึ่งคูณ 1000 เพื่อให้สมาชิกตัวที่หนึ่งเป็นหลักพัน สมาชิกตัวที่สองคูณ 100 เพื่อให้สมาชิกตัวที่สองเป็นหลักร้อย สมาชิกตัวที่สามคูณ 10 เพื่อให้สมาชิกตัวที่สามเป็นหลักสิบ สมาชิกตัวที่สี่คูณด้วย 1 เพื่อให้สมาชิกตัวที่สี่เป็นหลักหน่วย แล้วนำมารวมกันโดยการบวกกัน ส่วนตัวแปร B ก็ทำเช่นเดียวกับตัวแปร A หลังจากนั้นนำค่า A+B โดยให้เท่ากับ C แล้วแสดงค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์กับมาที่คอมพิวเตอร์โดยใช้ฟังก์ชัน printf ซึ่งการป้อนค่าและแสดงผลจะทำผ่านทางโปรแกรม Terminal

Source code การติดต่อสื่อสารโดยวิธีบวกเลข

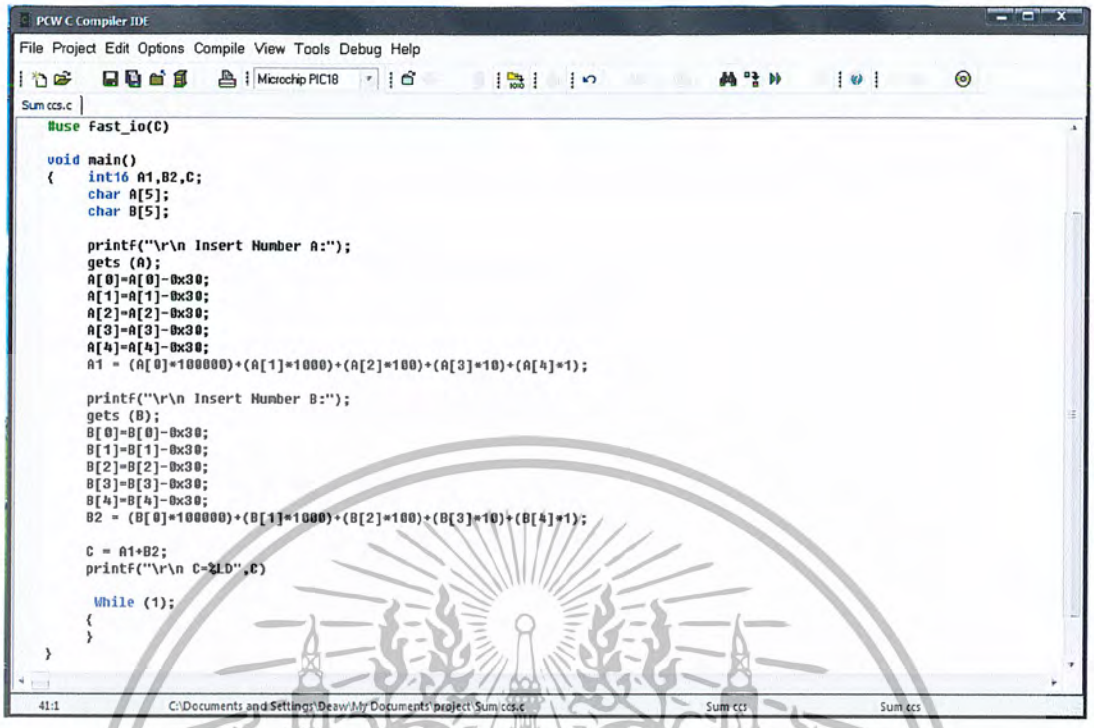
```
1. #include <18F4620.h>                    // รับข้อมูลจาก Header File ของ PIC18F4620
2. #define TX1   PIN_C6                    // กำหนดให้ขาที่ 6 เป็นการส่งข้อมูล
3. #define RX1   PIN_C7                    // กำหนดให้ขาที่ 7 เป็นการรับข้อมูล
4. #fuses   HS,NOLVP,NOWDT,NOPROTECT
                                           // กำหนดค่า Configuration
5. #use delay (clock = 10000000)           // กำหนดความถี่ออสซิเลเตอร์เท่ากับ 10 MHz
6. #use rs232(baud = 9600, xmit = TX1, rcv = RX1)
                                           // ใช้พอร์ตอนุกรมกำหนดอัตราการรับส่งข้อมูล 9600
7. void main()                             // ใช้ฟังก์ชันหลัก
8. {
9.      int16 A1,B2,C;                     // ประกาศตัวแปร A1,B2,C เป็นชนิด Integer
10.     char A[5],B[5];                    // ประกาศตัวแปร A[5],B[5] เป็นชนิด character
11.     While (1)                          // ให้โปรแกรมทำซ้ำใน Statement ในวงเล็บปีกกา
12.  {
13.      printf("\r Insert Number A:");    // แสดงผล Insert Number A แล้วเริ่มตำแหน่งแรกของ
                                           บรรทัดใหม่
14.      gets(A);                          // รับข้อมูลที่ป้อนเข้ามาใน A
15.      A[0]=A[0]-0x30;                   // นำค่า A array ที่ศูนย์ลบกับ 30 และนำไปเก็บไว้ใน
                                           array ที่ศูนย์
16.      A[1]=A[1]-0x30;                   // นำค่า A array ที่หนึ่งลบกับ 30 และนำไปเก็บไว้ใน
                                           array ที่หนึ่ง
17.      A[2]=A[2]-0x30;                   // นำค่า A array ที่สองลบกับ 30 และนำไปเก็บไว้ใน
                                           array ที่สอง
18.      A[3]=A[3]-0x30;                   // นำค่า A array ที่สามลบกับ 30 และนำไปเก็บไว้ใน
                                           array ที่สาม
19.      A[4]=A[4]-0x30;                   // นำค่า A array ที่สี่ลบกับ 30 และนำไปเก็บไว้ใน array
                                           ที่สี่
20.      A1=(A[0]*10000)+(A[1]*1000)+(A[2]*100)+(A[3]*10)+(A[4]*1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    // นำ array (A[0]x1000 ) + (A[1]x1000) + (A[2]x100)
                                       + (A[3]x10)+(A[4]x1) ให้เท่ากับ A1
21.   printf("%Ld",A1);             // แสดงค่าของ A1 เป็นแบบ Long Decimal
22.   printf("\r  Insert Number B:"); // แสดงผล Insert Number B แล้วเริ่มตำแหน่งแรกของ
                                       บรรทัดใหม่
23.   gets(B);                     // รับข้อมูลที่ป้อนเข้ามาใน B
24.   B[0]=B[0]-0x30;              // นำค่า B array ที่ศูนย์ลบกับ 30 และนำไปเก็บไว้ใน
                                       array ที่ศูนย์
25.   B[1]=B[1]-0x30;              // นำค่า B array ที่หนึ่งลบกับ 30 และนำไปเก็บไว้ใน
                                       array ที่หนึ่ง
26.   B[2]=B[2]-0x30;              // นำค่า B array ที่สองลบกับ 30 และนำไปเก็บไว้ใน
                                       array ที่สอง
27.   B[3]=B[3]-0x30;              // นำค่า B array ที่สามลบกับ 30 และนำไปเก็บไว้ใน
                                       array ที่สาม
28.   B[4]=B[4]-0x30;              // นำค่า B array ที่สี่ลบกับ 30 และนำไปเก็บไว้ใน array ที่สี่
29.   B2=(B[0]*10000)+(B[1]*1000)+(B[2]*100)+(B[3]*10)+(B[4]*1);
                                    // นำ array (B[0]x1000 ) + (B[1]x1000) + (B[2]x100) +
                                       (B[3]x10)+(B[4]x1) ให้เท่ากับ B2
30.   printf("%Ld",B2);            // แสดงค่าของ B2 เป็นแบบ Long Decimal
31.   C=A1+B2;                     // นำ A1+B2 ให้เท่ากับ C
32.   printf("\r  C=%Ld",C);       // แสดงผลรวม C เป็นแบบ Long Decimal แล้วเริ่ม
                                       ตำแหน่งแรกของบรรทัดใหม่
32.       }
34.   }
```

จบการทำงานของโปรแกรม

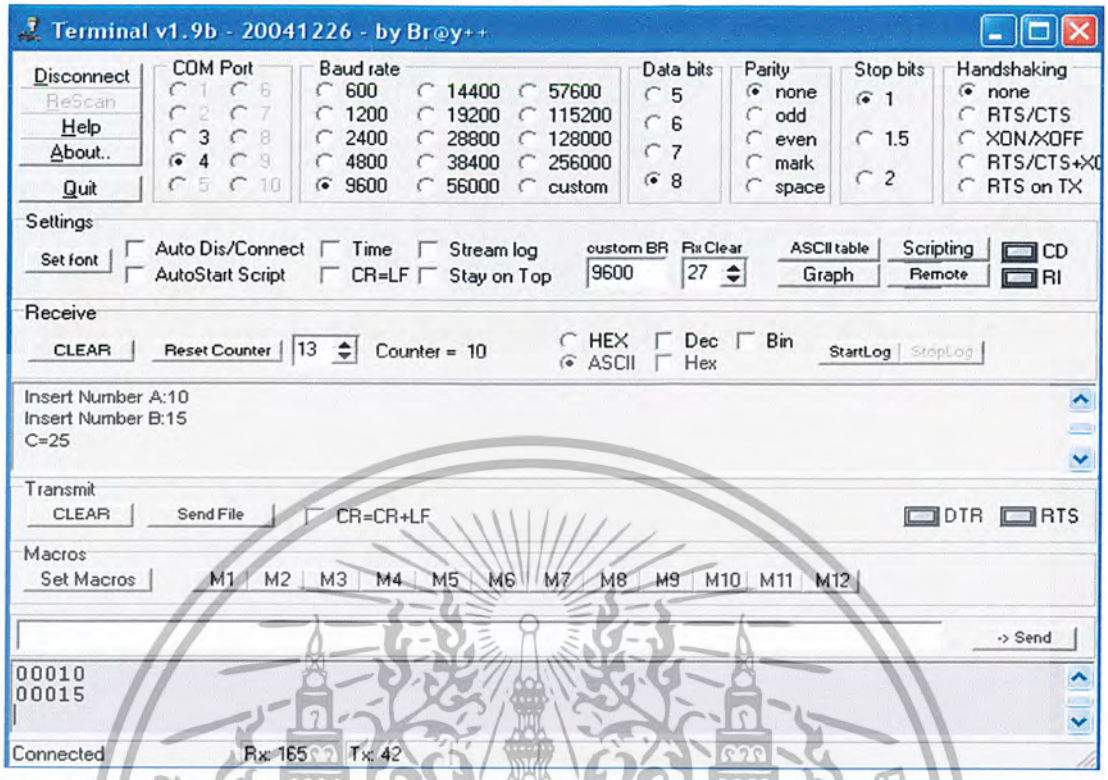
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.1 เขียน Source Code ลงในโปรแกรม PIC Compiler

ในการทดลองรับส่งข้อมูลการบวกลบจำนวนทางคณิตศาสตร์ผ่านทางพอร์ต RS-232 บนโปรแกรม Terminal และมีการตั้งค่าความเร็วในการรับส่งข้อมูลให้ตรงกับความเร็วในการรับส่งข้อมูลในโค้ดโปรแกรมและการเลือก Com Port ให้ตรงตามอุปกรณ์ ในการทดลองนี้จะใช้ Com Port 4 จากการทดลองมีการรับส่งข้อมูล โดยส่งข้อมูลผ่านทางคีย์บอร์ดเช่น 00010 เข้าไปจะปรากฏบนหน้าโปรแกรม Terminal ว่า "Insert Number A:10" แล้วทำการกดปุ่ม Enter เพื่อบันทึกค่าที่ PIC แล้วจะปรากฏคำว่า "Insert Number B:" แล้วจึงใส่เลขเข้าไป 00015 จะปรากฏ "Insert Number B:15" ขึ้นมาหลังจากนั้นไมโครคอนโทรลเลอร์ PIC18F4620 ก็จะส่งผลลัพธ์ของการบวกกันระหว่าง 10 กับ 15 จะปรากฏคำตอบขึ้นมาคือ C = 25 จะเห็นได้ว่าการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ PIC18F4620 มีการรับส่งข้อมูลและคำนวณได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.2 ผลการทดลองบวกเลขผ่านโปรแกรม Terminal

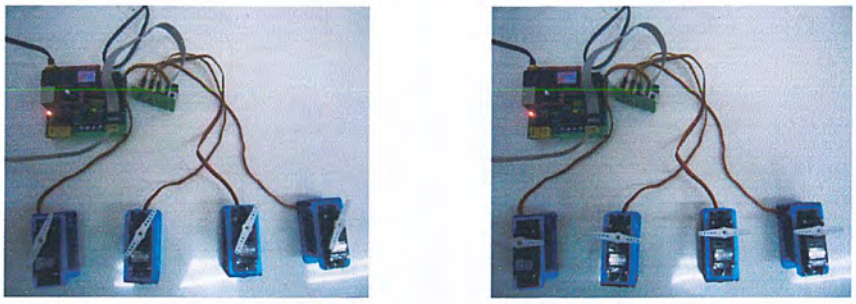
5.1.2 การทดลองควบคุมเซอร์โวมอเตอร์ 4 ตัว

การควบคุมเซอร์โวมอเตอร์ทั้ง 4 ตัวนั้น จะควบคุมด้วยโปรแกรม MATLAB ซึ่งการส่งข้อมูลจากคอมพิวเตอร์จะผ่านทางพอร์ต RS-232 ไปที่ไมโครคอนโทรลเลอร์ PIC18F4620 ที่เสียบอยู่บนบอร์ดอเนกประสงค์ ET-BASE PIC 40 โดยมีเซอร์โวมอเตอร์ทั้ง 4 ตัวเชื่อมต่ออยู่ที่พอร์ต A ของบอร์ดอเนกประสงค์ ET-BASE PIC 40 และกำหนดให้เป็นเอาต์พุตพอร์ต (Output Port) ผลการทดลองมีดังนี้

มอเตอร์อยู่ตรงกลางทั้ง 4 ตัว

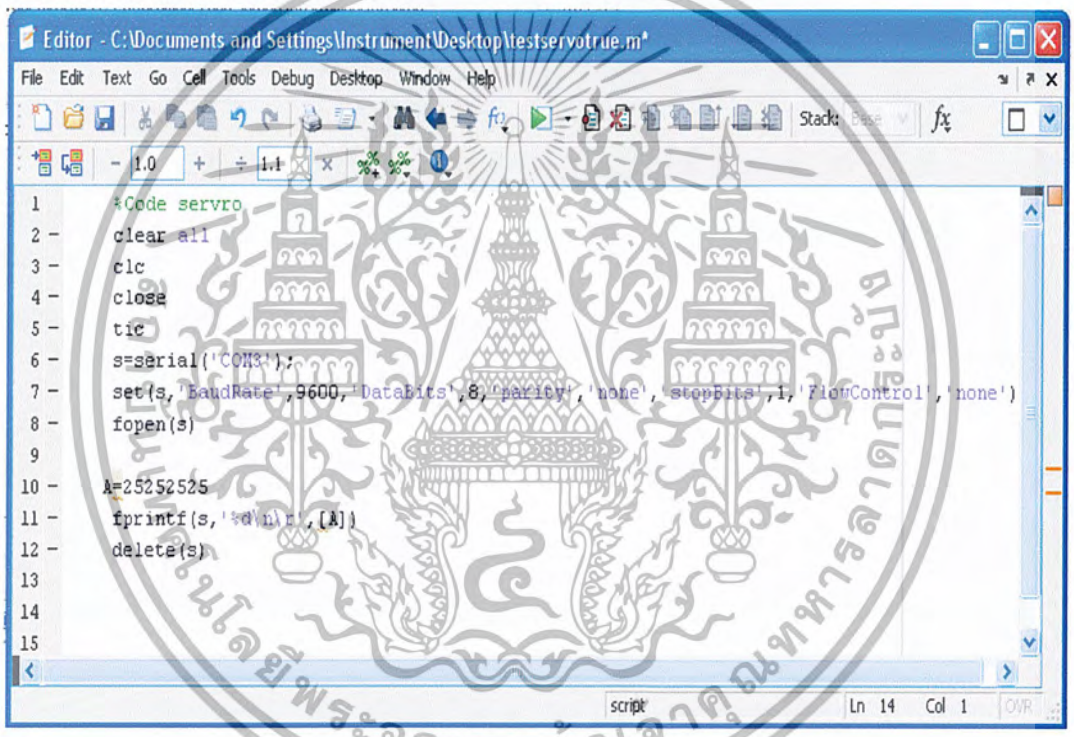
เนื่องจากไมโครคอนโทรลเลอร์ PIC18F4620 ควบคุมเซอร์โวมอเตอร์ด้วยการกำหนดค่า Timer ที่อยู่ในช่วง 16~32 ซึ่งเมื่อค่า Timer เท่ากับ 16 เซอร์โวมอเตอร์จะหมุนทางด้านซ้ายสุด และถ้าค่า Timer เท่ากับ 32 เซอร์โวมอเตอร์จะหมุนทางด้านขวาสุด สอดคล้องกับกรณีที่ควบคุมเซอร์โวมอเตอร์แบบความกว้างของสัญญาณพัลส์ในช่วง 1ms~2ms สำหรับการควบคุมเซอร์โวมอเตอร์ให้หมุนไปที่ตำแหน่งกึ่งกลางนั้นค่า Timer จะมีค่าเท่ากับ 25 ซึ่งโปรแกรมใน MATLAB จะกำหนดให้เป็น fprintf(s,'%d\n\r',[25252525]); เพื่อควบคุมเซอร์โวมอเตอร์ทั้ง 4 ตัวจะหมุนมาอยู่ที่จุดกึ่งกลาง ดังแสดงรูปที่ 5.3 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์แสดงดังภาพที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ค่า Timer ที่ 32 (ข) ค่า Timer ที่ 25

ภาพที่ 5.3 เซอร์โวมอเตอร์ให้หมุนไปที่ตำแหน่งกึ่งกลางที่ตำแหน่งกึ่งกลาง

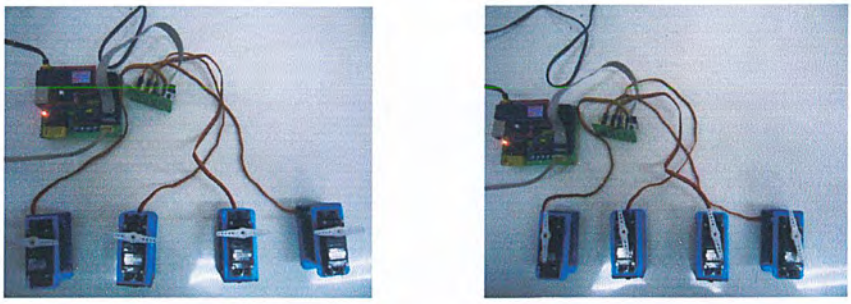


ภาพที่ 5.4 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปที่ตำแหน่งตรงกลางทั้ง 4 ตัว

มอเตอร์หมุนไปทางซ้ายมือสุดทั้ง 4 ตัว

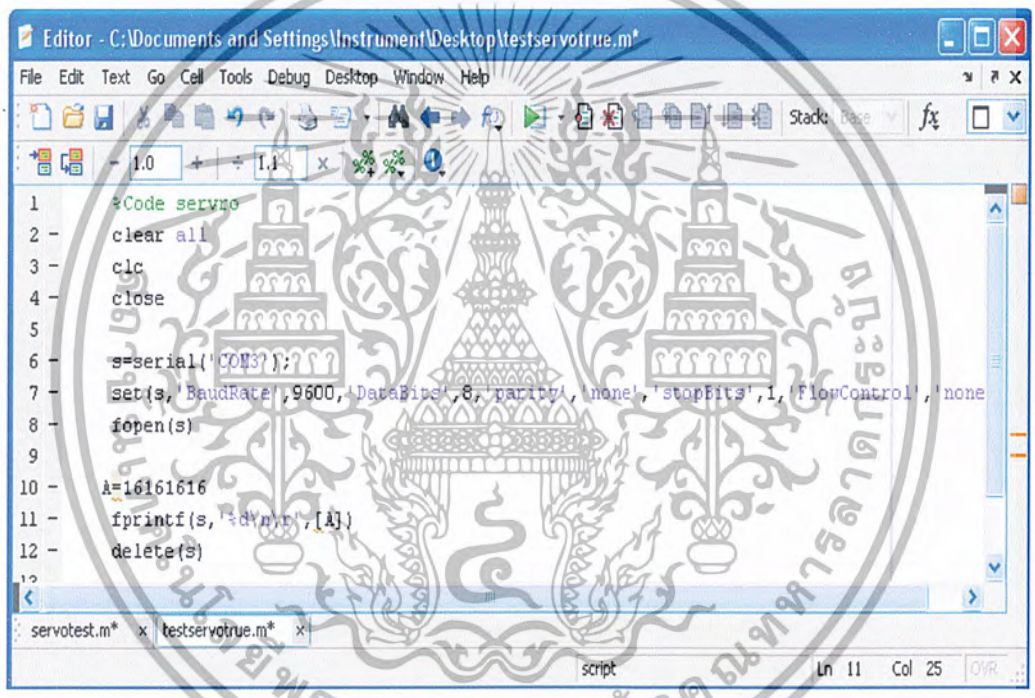
หลังจากที่เซอร์โวมอเตอร์หมุนมาที่ตำแหน่งกึ่งกลางแล้ว จากนั้นทดลองป้อนค่าผ่านทางโปรแกรม MATLAB กำหนดค่า Timer เท่ากับ 16 เพื่อให้เซอร์โวมอเตอร์หมุนไปที่ด้านซ้ายสุดด้วยคำสั่ง fprintf(s,'%d\n\r',[16161616]); เซอร์โวมอเตอร์ทั้ง 4 ตัวจะหมุนมาอยู่ที่จุดซ้ายสุด ดังภาพที่ 5.5 และโปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์แสดงดังภาพที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ค่า Timer ที่ 25 (ข) ค่า Timer ที่ 16

ภาพที่ 5.5 เซอร์โวมอเตอร์ถูกควบคุมให้หมุนไปทางด้านซ้ายมือสุดทั้ง 4 ตัว

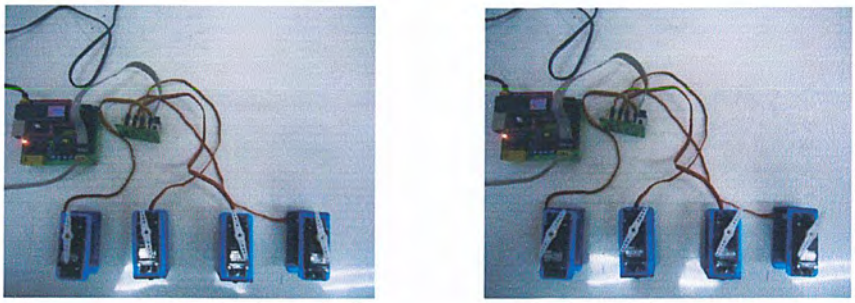


ภาพที่ 5.6 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปทางด้านซ้ายมือสุดทั้ง 4 ตัว

มอเตอร์หมุนไปทางขวามือสุดทั้ง 4 ตัว

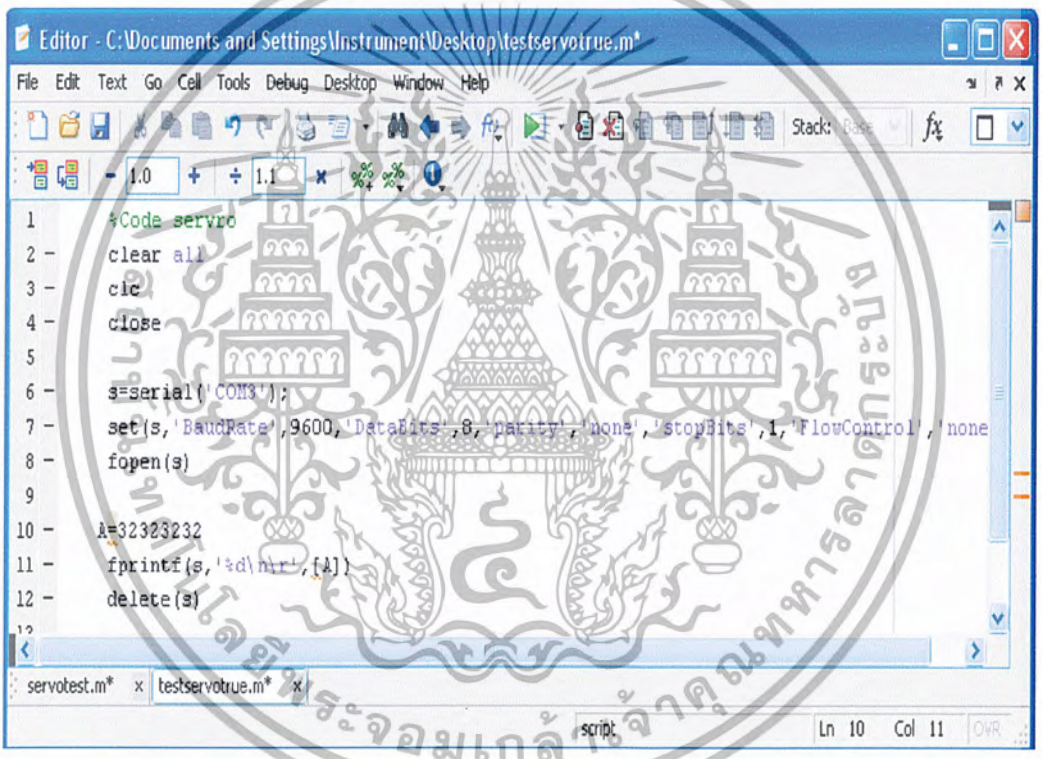
การควบคุมเซอร์โวมอเตอร์ทั้ง 4 ตัวเพื่อให้หมุนไปทางด้านขวามือสุดก็กระทำในลักษณะเดียวกัน แต่จะป้อนค่า Timer เป็น 32 แทน รูปแสดงการหมุนของเซอร์โวมอเตอร์ทั้ง 4 ตัว และโปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ แสดงดังภาพที่ 5.7 และ 5.8 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ค่า Timer ที่ 16 (ข) ค่า Timer ที่ 32

ภาพที่ 5.7 เซอร์โวมอเตอร์ถูกควบคุมให้หมุนไปทางด้านขวามือสุดทั้ง 4 ตัว



รูปที่ 5.8 โปรแกรม MATLAB ควบคุมเซอร์โวมอเตอร์ให้หมุนไปทางด้านขวามือสุดทั้ง 4 ตัว

จากการทดลองควบคุมเซอร์โวมอเตอร์ทั้ง 4 ตัว สรุปได้ว่าเราสามารถควบคุมเซอร์โวมอเตอร์ให้หมุนไปทางซ้ายมือ ตรงกลาง และขวามือ ด้วยการกำหนดค่า Timer ในไมโครคอนโทรลเลอร์ PIC18F4620 ให้อยู่ในช่วงระหว่าง 16 ถึง 32 ซึ่งเราจะนำไปประยุกต์ใช้เพื่อการติดตามคนที่อยู่ในภาพต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 การทดลองเพื่อหาคุณลักษณะของเซอร์โวมอเตอร์

ในการทดลองนี้เป็นการหาคุณลักษณะของเซอร์โวมอเตอร์ทั้งหมดที่ใช้ในการทำโครงงานนี้ ซึ่งคุณลักษณะที่ต้องการตรวจสอบก็คือค่าฮีสเตอรีซีสของเซอร์โวมอเตอร์แต่ละตัว ดังที่ได้กล่าวไว้ในทฤษฎีบทที่ 4 และการหาค่าจำนวนพัลส์ต่อพิกเซลที่ต้องส่งให้เซอร์โวมอเตอร์แต่ละตัว เนื่องจากเซอร์โวมอเตอร์ทำหน้าที่ในการหมุนกล้องไปที่ตำแหน่งต่างๆ ดังนั้นแนวความคิดในการหาค่าฮีสเตอรีซีสของเซอร์โวมอเตอร์คือการส่งค่าความกว้างของสัญญาณพัลส์ที่ 1500 µs เพื่อให้เซอร์โวมอเตอร์หมุนไปที่ตำแหน่งตรงกลาง จากนั้นหมุนเซอร์โวมอเตอร์ที่ไปตำแหน่งต่างๆ ไม่ว่าจะเป็นทางด้านซ้ายมือหรือขวามือ แล้วหมุนเซอร์โวมอเตอร์กลับที่ตำแหน่งตรงกลางอีกครั้ง อย่างไรก็ดีทุกๆครั้งที่เซอร์โวมอเตอร์หมุนไป เราจะให้กล้องถ่ายภาพวัตถุ และคำนวณหาจุดศูนย์กลางของวัตถุ จากนั้นนำค่าจุดศูนย์กลางวัตถุเมื่อเซอร์โวมอเตอร์หมุนไปที่ตำแหน่งอื่นลบด้วยค่าจุดศูนย์กลางวัตถุเมื่อเซอร์โวมอเตอร์อยู่ที่ตำแหน่งตรงกลาง จะทำให้เราสามารถหาค่าจำนวนพัลส์ต่อพิกเซลได้ อย่างไรก็ตามทุกๆ ครั้งที่เซอร์โวมอเตอร์หมุนกลับมาอยู่ที่ตำแหน่งตรงกลาง จุดศูนย์กลางของวัตถุจะมีค่าเปลี่ยนแปลงไปไม่ทับที่ตำแหน่งเดิม เป็นผลมาจากค่าฮีสเตอรีซีสของเซอร์โวมอเตอร์ การหมุนของเซอร์โวมอเตอร์เพื่อให้สามารถติดตามคนที่กำลังเคลื่อนที่ในภาพได้ เราจะใช้กล้อง 2 ชุดโดยมีเซอร์โวมอเตอร์ชุดละ 2 ตัว ซึ่งเซอร์โวมอเตอร์ 1 ตัวสามารถหมุนได้ 1 แกน ดังนั้นเพื่อให้กล้องสามารถหมุนได้ 2 แกนคือแกน X และแกน Y เราจึงติดตั้งกล้องกับเซอร์โวมอเตอร์ 2 ตัว สำหรับความกว้างของสัญญาณพัลส์ที่ใช้ในการทดลองอยู่ในช่วง 1350 – 1650 µs เพราะเป็นช่วงที่วัตถุยังอยู่ในขอบเขตของภาพซึ่งมีขนาด 320x240 พิกเซล ซึ่งในการทดลองจะมีส่วนของโปรแกรมการควบคุมเซอร์โวมอเตอร์แบ่งออกเป็น 2 ส่วน คือ ในส่วนของโปรแกรมภาษาซีที่ใช้เบิร์นลงในไมโครคอนโทรลเลอร์ PIC และโปรแกรม MATLAB ที่ใช้ควบคุมเซอร์โวมอเตอร์ดังรายละเอียดต่อไปนี้

หลักการทำงานของโปรแกรมควบคุมเซอร์โวมอเตอร์

ในขั้นตอนนี้ไมโครคอนโทรลเลอร์ PIC18F4620 จะควบคุมการทำงานของเซอร์โวมอเตอร์ทั้ง 4 ตัว ในรูปแบบของการควบคุมความกว้างของสัญญาณพัลส์ในช่วง 1000 µsec ~ 2000 µsec นั่นคือเซอร์โวมอเตอร์จะหมุนไปทางซ้ายมือสุด เมื่อได้รับความกว้างของสัญญาณพัลส์เป็น 1000 µsec ในทำนองเดียวกันจะหมุนไปทางขวามือสุด เมื่อได้รับความกว้างของสัญญาณพัลส์เป็น 2000 µsec การควบคุมเซอร์โวมอเตอร์ในขั้นตอนนี้ก็เพื่อหาค่าฮีสเตอรีซีส (Hysteresis) ของเซอร์โวมอเตอร์และหาค่าจำนวนพัลส์ต่อพิกเซลที่ต้องส่งให้เซอร์โวมอเตอร์ทั้ง 4 ตัว โดยขั้นตอนการทำงานของโปรแกรมมีดังนี้ กำหนดค่า Configuration เลือกการสื่อสารผ่านทางพอร์ต RS-232 และกำหนดให้ความเร็วในการสื่อสารข้อมูลเท่ากับ 9600 bit/second สำหรับการรับข้อมูล (Data) จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MATLAB เพื่อให้เป็นไปตามมาตรฐานจะกำหนดให้วงเล็บเปิด "(หมายถึงบิตเริ่มต้น (Start Bit) ตามด้วยข้อมูล (Data) และให้วงเล็บปิด)" เป็นบิตสิ้นสุด (Stop Bit) ตัวอย่างของการส่งข้อมูลจากโปรแกรม MATLAB ไปที่ไมโครคอนโทรลเลอร์ PIC18F4620 มีดังนี้ (A1500B1000C2000D1500) เป็นต้น จากตัวอย่างข้อมูลความกว้างของสัญญาณพัลส์ D1500 จะถูกเก็บไว้ในตัวแปร A_BUF สำหรับควบคุมเซอร์โวมอเตอร์ตัวที่ 1 ในทำนองเดียวกัน ข้อมูลความกว้างของสัญญาณพัลส์ C2000, B1000 และ A1500 จะถูกเก็บไว้ในตัวแปร B_BUF, C_BUF และ D_BUF สำหรับควบคุมเซอร์โวมอเตอร์ตัวที่ 2, 3 และ 4 ตามลำดับ โดยจะเก็บเข้ามาทีละบิตซึ่งค่า 1500, 2000, 1000 และ 1500 คือความกว้างของสัญญาณพัลส์ที่ต้องการควบคุมให้เซอร์โวมอเตอร์หมุนไปที่ตำแหน่งต่างๆ ตามที่กำหนด ค่าที่อยู่ใน A_BUF, B_BUF, C_BUF และ D_BUF ซึ่งเป็นชนิด unsigned int จะถูกเปลี่ยนให้เป็น long int และถูกส่งให้เป็น Output ของไมโครคอนโทรลเลอร์ PIC ผ่านทาง Port C0 ถึง C3 โดยการควบคุมเป็นแบบ Delay ให้คงค่าสถานะโลจิก "High" ตามขนาดความกว้างของสัญญาณพัลส์ที่ส่งมาจากโปรแกรม MATLAB และตามด้วยโลจิก "Low"

Source code ในการหาค่าฮีสเตอรีซีส (Hysteresis) ของโปรแกรมที่ใช้ภาษาซี

```
#include <18F4620.h>                    // กำหนดเบอร์ไมโครคอนโทรลเลอร์ PIC ที่ใช้งาน
#priority RDA,TIMER1                    // จัดลำดับความสำคัญของอินเตอร์รัปต์
#define TX PIN_C6                       // กำหนดให้ขาที่ 6 เป็นการส่งข้อมูล
#define RX PIN_C7                       // กำหนดให้ขาที่ 7 เป็นการรับข้อมูล
#define Servo_0  PIN_C0                 // ประกาศใช้งานพอร์ตซี ใช้ขา C0
#define Servo_1  PIN_C1                 // ประกาศใช้งานพอร์ตซี ใช้ขา C1
#define Servo_2  PIN_C2                 // ประกาศใช้งานพอร์ตซี ใช้ขา C2
#define Servo_3  PIN_C3                 // ประกาศใช้งานพอร์ตซี ใช้ขา C3
#fuses H4,NOLVP,NOWDT,NOPROTECT         // กำหนดการทำงานหลักของไมโครคอนโทรลเลอร์
#use delay (clock = 40000000)           // กำหนดสัญญาณนาฬิกาที่ต้องการใช้งาน
#use rs232(baud = 9600, xmit = TX, rcv = RX,ERRORS) // กำหนดการสื่อสารข้อมูลผ่าน RS-232
                                                    ที่อัตราบอร์ดเรต 9600
#use fast_io(A)                         // เข้าถึงพอร์ต A โดยตรง
#use fast_io(B)                         // เข้าถึงพอร์ต B โดยตรง
#use fast_io(C)                         // เข้าถึงพอร์ต C โดยตรง
#use fast_io(D)                         // เข้าถึงพอร์ต D โดยตรง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#use fast_io(E)                                  // เข้าถึงพอร์ต E โดยตรง
#include <string.h>                             // กำหนดใช้ฟังก์ชัน String
#include <stdlib.h>                             // กำหนดใช้ฟังก์ชันพื้นฐาน
unsigned int16 pulse_Servo0,pulse_Servo1,pulse_Servo2,pulse_Servo3;  // กำหนดตัวแปรเป็น
                                                    จำนวนเต็มแบบ16 บิตไม่มี เครื่องหมาย
unsigned int   dat;                              // กำหนดตัวแปร dat เป็นแบบจำนวนเต็มไม่มี
                                                    เครื่องหมาย
unsigned int   A_BUF[20],B_BUF[10],C_BUF[10],D_BUF[10];
                                                 // กำหนดตัวแปรเป็นจำนวนเต็มไม่มีเครื่องหมาย
unsigned int   Rec_packet[30];                   // กำหนดตัวแปร Rec_packet เป็นจำนวนเต็มไม่มี
                                                    เครื่องหมายมีจำนวนสมาชิก 30 ตัว
unsigned int  pkt_full=0;                        // กำหนดตัวแปร pkt_full=0เป็นจำนวนเต็มไม่มี
                                                    เครื่องหมาย
unsigned      rx_cnt=0;                          // กำหนดให้ตัวแปร rx_cnt=0 ไม่มีเครื่องหมาย
#INT_RDA                                         // ใช้อินเตอร์รับเมื่อผ่านพอร์ตอนุกรม
void IntRDA_isr()                                // ฟังก์ชันอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม
{
  disable_interrupts(INT_RDA);                   // ไม่ใช้ฟังก์ชันอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม

  dat = getc();                                  // รับค่า dat จากคีย์บอร์ด
  if (dat=='(')rx_cnt=0;                         // ถ้า dat มีค่าเท่ากับ " ( " จะให้ rx_cnt=0
  if (dat==')')pkt_full=1;                       // ถ้า dat มีค่าเท่ากับ " ) " จะให้ pkt_full=1
  Rec_packet[rx_cnt]=dat;                        // นำค่า rx_cnt มาเก็บไว้ที่ Rec_packetและให้
                                                    เท่ากับ dat
  rx_cnt++;                                      // rx_cnt เพิ่มทีละหนึ่ง
  enable_interrupts(INT_RDA);                    // เปิดใช้งานอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม
}
void main()                                      // ฟังก์ชันหลัก
{
   enable_interrupts(INT_RDA);                   // เปิดใช้งานอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม
   enable_interrupts(GLOBAL);                    // เปิดใช้งานอินเตอร์รัพท์ทั้งหมด
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
set_tris_a(0B11111111);                 // กำหนดการใช้ขาสัญญาณของพอร์ต A
set_tris_b(0B11111111);                 // กำหนดการใช้ขาสัญญาณของพอร์ต B
set_tris_c(0B11110000);                 // กำหนดการใช้ขาสัญญาณของพอร์ต C
set_tris_d(0B11111111);                 // กำหนดการใช้ขาสัญญาณของพอร์ต D
output_low(Servo_0);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขาพอร์ต
                                           C0 มีสถานะเป็นลอจิก 0
output_low(Servo_1);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขาพอร์ต
                                           C1 มีสถานะเป็นลอจิก 0
output_low(Servo_2);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขาพอร์ต
                                           C2 มีสถานะเป็นลอจิก 0
output_low(Servo_3);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขาพอร์ต
                                           C3 มีสถานะเป็นลอจิก 0
                  //---- Start value of servo motor about 1.5 ms ----//
pulse_Servo0=1500;                      // กำหนดให้ค่าเริ่มต้น pulse_Servo0=1500
pulse_Servo1=1500;                      // กำหนดให้ค่าเริ่มต้น pulse_Servo1=1500
pulse_Servo2=1500;                      // กำหนดให้ค่าเริ่มต้น pulse_Servo2=1500
pulse_Servo3=1500;                      // กำหนดให้ค่าเริ่มต้น pulse_Servo3=1500
                  //-------------------------------------------//
while(1)
{
                  //---------- Check RS232 Command ----------//
 if (pkt_full==1)                       // ถ้า pkt_full==1 ให้ปิดการใช้งานอินเตอร์รัพท์
 {
  disable_interrupts(INT_RDA);
                     // Get Servo 0 value index as 2,3,4,5 of packet//
  A_BUF[0]=Rec_packet[2];               // เก็บค่า A_BUF 0 ไว้ในRec_packet ลำดับที่ 2
  A_BUF[1]=Rec_packet[3];               // เก็บค่า A_BUF 1 ไว้ในRec_packet ลำดับที่ 3
  A_BUF[2]=Rec_packet[4];               // เก็บค่า B_BUF 2 ไว้ในRec_packet ลำดับที่ 4
  A_BUF[3]=Rec_packet[5];               // เก็บค่า B_BUF 3 ไว้ในRec_packet ลำดับที่ 5
  A_BUF[4]=0; // end                    // เก็บค่า A_BUF 4 ให้เท่ากับศูนย์
  pulse_Servo0= atol(A_BUF);            // แปลงค่า A ที่รับมาเป็นค่า string ให้เป็น
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        ตัวเลขจำนวนเต็มชนิด long integer
        // Get Servo 1 value index as 7,8,9,10 of packet//
B_BUF[0]=Rec_packet[7];          // เก็บค่า B_BUF 0 ไว้ในRec_packet ลำดับที่ 7
B_BUF[1]=Rec_packet[8];          // เก็บค่า B_BUF 1 ไว้ในRec_packet ลำดับที่ 8
B_BUF[2]=Rec_packet[9];          // เก็บค่า B_BUF 2 ไว้ในRec_packet ลำดับที่ 9
B_BUF[3]=Rec_packet[10];         // เก็บค่า B_BUF 3 ไว้ในRec_packet ลำดับที่ 10
B_BUF[4]=0; // end               // เก็บค่า B_BUF 4 ให้เท่ากับศูนย์
pulse_Servo1= atol(B_BUF);       // แปลงค่า B ที่รับมาเป็นค่า string ให้เป็น
                                        ตัวเลขจำนวนเต็มชนิด long integer
        // Get Servo 2 value index as 12,13,14,15 of packet//
C_BUF[0]=Rec_packet[12];                 // เก็บค่า C_BUF 0 ไว้ในRec_packet ลำดับที่ 12
C_BUF[1]=Rec_packet[13];                 // เก็บค่า C_BUF 1 ไว้ในRec_packet ลำดับที่ 13
C_BUF[2]=Rec_packet[14];                 // เก็บค่า C_BUF 2 ไว้ในRec_packet ลำดับที่ 14
C_BUF[3]=Rec_packet[15];                 // เก็บค่า C_BUF3 ไว้ในRec_packetลำดับที่15
C_BUF[4]=0; // end                      // เก็บค่า C_BUF 4 ให้เท่ากับศูนย์
pulse_Servo2= atol(C_BUF);               // แปลงค่า A ที่รับมาเป็นค่า string ให้เป็น
                                        ตัวเลขจำนวนเต็มชนิด long integer
        // Get Servo 3 value index as 17,18,19,20 of packet//
D_BUF[0]=Rec_packet[17];          // เก็บค่า D_BUF 0 ไว้ในRec_packet ลำดับที่ 17
D_BUF[1]=Rec_packet[18];          // เก็บค่า D_BUF 1 ไว้ในRec_packet ลำดับที่ 18
D_BUF[2]=Rec_packet[19];          // เก็บค่า D_BUF 2 ไว้ในRec_packet ลำดับที่ 19
D_BUF[3]=Rec_packet[20];          // เก็บค่า D_BUF 3 ไว้ในRec_packet ลำดับที่ 20
D_BUF[4]=0;                       // เก็บค่า D_BUF 4 ให้เท่าศูนย์
pulse_Servo3= atol(D_BUF);        // แปลงค่า A ที่รับมาเป็นค่า string ให้เป็น
                                        ตัวเลขจำนวนเต็มชนิด long integer
rx_cnt=0;                         // กำหนด rx_cnt=0
pkt_full=0;                       // กำหนด pkt_full=0
enable_interrupts(INT_RDA);       // เปิดใช้งานอินเตอร์รัพท์เมื่อใช้พอร์ตอนุกรม
}
        //---------- End of RS232 Command----------//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//---------- Servo Motor Control loop---------//
output_high(Servo_0);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C0 มีสถานะเป็นลอจิก 1
delay_us(pulse_Servo0);                  // ดีเลย์การทำงานให้มีค่าเท่ากับ Pulse_Servo 0
output_low(Servo_0);                     // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C0 มีสถานะเป็นลอจิก 0
output_high(Servo_1);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C1 มีสถานะเป็นลอจิก 1
delay_us(pulse_Servo1);                  // ดีเลย์การทำงานให้มีค่าเท่ากับ Pulse_Servo 1
output_low(Servo_1);                     // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C1 มีสถานะเป็นลอจิก 0
output_high(Servo_2);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้
                                           ขาพอร์ต C2 มีสถานะเป็นลอจิก 1
delay_us(pulse_Servo2);                  // ดีเลย์การทำงานให้มีค่าเท่ากับ Pulse_Servo 2
output_low(Servo_2);                     // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C2 มีสถานะเป็นลอจิก 0
output_high(Servo_3);                    // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C3 มีสถานะเป็นลอจิก 1
delay_us(pulse_Servo3);                  // ดีเลย์การทำงานให้มีค่าเท่ากับ Pulse_Servo 3
output_low(Servo_3);                     // ควบคุมขาพอร์ตแบบบิต ใช้กำหนดให้ขา
                                           พอร์ต C3 มีสถานะเป็นลอจิก 0
 delay_ms(15);                           // delay 15 ms
                        //---------- End of Servo Control ----------//
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

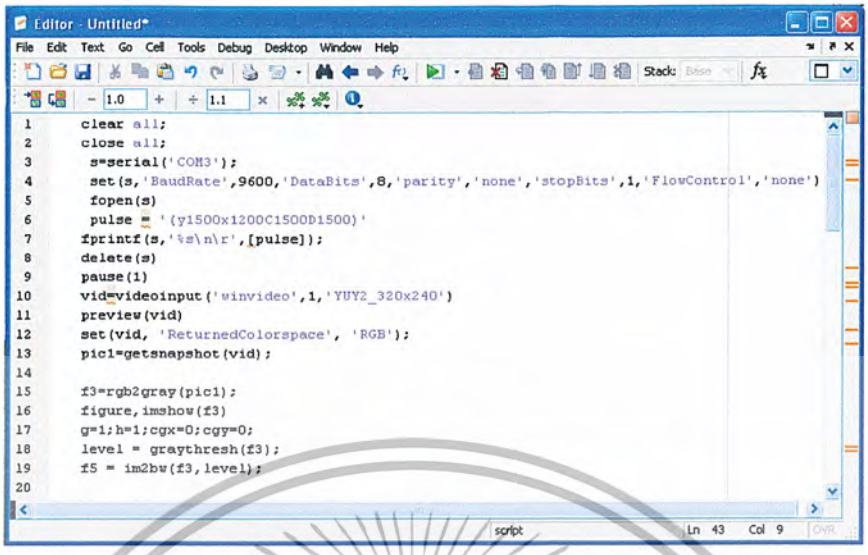
Source code ในการหาค่าฮีสเตอรีซีส (Hysteresis) ของโปรแกรม MATLAB ที่ใช้ควบคุมเซอร์โวมอเตอร์

```
clear all;                                          % ล้างข้อมูลทั้งหมด
close all                                           % ปิดหน้าต่างทั้งหมด
 s=serial('COM3');                                  % กำหนดใช้พอร์ตที่ 3
set(s,'BaudRate',9600,'DataBits',8,'parity','none','stopBits',1,'FlowControl','none')
                                                    % กำหนดค่าต่างๆในการส่งข้อมูล
fopen(s)                                            % เปิดการใช้พอร์ต s
 pulse = '(Y1500X1200C1500D1500)'                   % กำหนดค่า pulse
fprintf(s,'%s\n\r',[pulse]);                        % ส่งค่าไปยังเซอร์โวมอเตอร์
delete(s)                                           % ลบตัวแปร s
pause(1)                                            % พักการประมวลผล 1 วินาที
vid=videoinput('winvideo',1,'YUY2_320x240')         % รับวีดีโอมาใส่ในตัวแปร vid
preview(vid)                                        % แสดงวีดีโอ
set(vid, 'ReturnedColorspace', 'RGB');              % กำหนดให้เป็นภาพ RGB
pic1=getsnapshot(vid);                              % เก็บภาพไว้ในตัวแปร pic1
f3=rgb2gray(pic1);                                  % แปลงเป็นภาพระดับสีเทา
figure,imshow(f3)                                   % แสดงภาพ f3
g=1;h=1;cgx=0;cgy=0;                                % กำหนดค่าตัวแปรต่างๆ
level = graythresh(f3);                             % หาค่าเทรสโชว์
f5 = im2bw(f3,level);                               % แปลงเป็นภาพขาวดำ
imshow(f5)                                          % แสดงภาพ f5
   cgy11=0;                                         % กำหนดค่า cgy11 เท่ากับ 0
   cgx11=0;                                         % กำหนดค่า cgx11 เท่ากับ 0
[L,num]=bwlabel(f5);                                % ใช้ bwlabel กับ f5
 for i=1:num                                        % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง num
     [r,c]=find(L==i);                              % ใช้คำสั่ง find
     max(i)=0; min(i)=255; max1(i)=0; min1(i)=255;      % กำหนดค่าต่างๆ
    u=size(r);                                      % ให้ u เท่ากับขนาดของ r
    for j=1:u                                       % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
     if c(j)>max(i)                                 % กำหนดเงื่อนไขให้ c มากกว่า max
        max(i)=c(j);                                % ให้ max เท่ากับ c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end                                          % จบการทำงาน for loop
    if c(j)<min(i)                           % กำหนดเงื่อนไขให้ c น้อยกว่า min
      min(i)=c(j);                           % ให้ min เท่ากับ c
    end                                      % จบการทำงาน for loop
  end                                        % จบการทำงาน for loop
  for j=1:u                                  % วนรอบแบบ for-loop ตั้งแต่ 1 ถึง u
    if r(j)>max1(i)                          % กำหนดเงื่อนไขให้ r มากกว่า max1
      max1(i)=r(j);                          % ให้ max1 เท่ากับ r
    end                                      % จบการทำงานแบบ for loop
    if r(j)<min1(i)                          % กำหนดเงื่อนไขให้ r น้อยกว่า min1
      min1(i)=r(j);                          % ให้ min1 เท่ากับ r
    end                                      % จบการทำงาน for loop
  end                                        % จบการทำงาน for loop
   if max(i)-min(i)>10&&max1(i)-min1(i)>10&&max(i)-min(i)<100
                                             % กำหนดค่าตัวแปรต่างๆ
  x11=[ min1(i) min1(i) max1(i) max1(i) min1(i)];
                                             % กำหนดค่าตัวแปรต่างๆ
  y11=[ min(i)  max(i)  max(i)  min(i)  min(i)];
                                             % กำหนดค่าตัวแปรต่างๆ
  hold on                                    % ค้างค่าไว้
  plot (y11,x11,'-r')                        % พล็อตกราฟ
  xmin11 = min(i);                           % เก็บค่าไว้ในตัวแปร xmin11
  xmax11 = max(i);                           % เก็บค่าไว้ในตัวแปร xmax11
  ymin11 = min1(i) ;                         % เก็บค่าไว้ในตัวแปร ymin11
  ymax11 = max1(i);                          % เก็บค่าไว้ในตัวแปร ymax11
  cgy11=((ymax11-ymin11)/2)+ymin11;          % หาค่าจุด centroid y
  cgx11=((xmax11-xmin11)/2)+xmin11;          % หาค่าจุด centroid x
  plot(cgx11,cgy11,'-r.');                   % พล็อตค่าจุด centroid
   end                                       % จบการทำงาน for loop
end                                          % จบการทำงาน for loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.9 การเขียนโปรแกรมประมวลผลโดยใช้ MATLAB

ขั้นตอนการทดลอง

1. เนื่องจากเราต้องการหาค่าฮิสเตอรีซีสและหาค่าจำนวนพัลส์ต่อพิกเซลของเซอร์โวมอเตอร์ทุกตัวซึ่งค่าที่ได้ออกมาจะอยู่ในรูปความไวของเซอร์โวมอเตอร์ ดังนั้นเราจะสั่งให้เซอร์โวมอเตอร์หมุนไปทางด้านซ้าย ด้านขวา ด้านบน และด้านล่าง โดยการเพิ่มและลดความกว้างของสัญญาณพัลส์ลงทีละ 50 µs

2. หมุนเซอร์โวมอเตอร์ไปที่ตำแหน่งตรงกลางโดยใช้ความกว้างของสัญญาณพัลส์มีค่าเป็น 1500 µs จากนั้นถ่ายภาพพื้นหลัง (Background) ซึ่งเป็นสีดำและเก็บไว้เป็นภาพที่ 1 ต่อมานำกระดาษรูปสี่เหลี่ยมจัตุรัสสีขาวมาติดบนฉากสีดำซึ่งเปรียบเสมือนวัตถุมาวางให้อยู่ในตำแหน่งใกล้กับจุดศูนย์กลางของภาพมากที่สุดคือที่จุด Coordinate เท่ากับ (160,120) พิกเซล เนื่องจากภาพที่ใช้ในการประมวลผลมีขนาด 320x240 พิกเซล จากนั้นถ่ายภาพวัตถุเก็บไว้เป็นภาพที่ 2 แล้วนำภาพที่ 1 กับภาพที่ 2 มาลบกันและสั่งให้โปรแกรมตีกรอบล้อมรอบวัตถุเพื่อหาจุดศูนย์กลางของวัตถุซึ่งจะได้เป็นจุด Coordinate (x,y) ดังภาพที่ 5.10

(ก) ภาพที่ 1 ภาพพื้นหลัง (Background) (ข) ภาพที่ 2 วัตถุอยู่ตรงกลางของภาพ

ภาพที่ 5.10 การตีกรอบล้อมรอบวัตถุเพื่อหาจุดศูนย์กลางของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการลดค่าความกว้างของสัญญาณพัลส์จาก 1500 µs เป็น 1450 µs แล้วป้อนค่าให้ไมโครคอนโทรลเลอร์ส่งสัญญาณพัลส์ไปยังเซอร์โวมอเตอร์ในแนวแกน X ทางด้านซ้ายมือ เพื่อทำให้กล้องหมุน โดยถ่ายภาพเก็บไว้เป็นภาพที่ 3 แล้วนำมาลบกับภาพที่ 2 ซึ่งเราจะเลือกเฉพาะเฟรมของภาพที่ 3 มาแสดงและประมวลผลโดยสั่งให้โปรแกรมตีกรอบล้อมรอบวัตถุ เพื่อหาจุด Coordinate (x,y) ของจุดศูนย์กลางของวัตถุดังแสดงในภาพที่ 5.11 (ก) แล้วบันทึกค่าลงในตารางที่ 5.1 จากนั้นป้อนค่าความกว้างของสัญญาณพัลส์กลับไปที่ 1500 µs เพื่อหาจุด Coordinate (x,y) ของจุดศูนย์กลางของวัตถุและตรวจสอบว่าเซอร์โวมอเตอร์ในแนวแกน X หมุนกลับมาที่ตำแหน่งเดิมหรือคลาดเคลื่อนไปจากเดิมเท่าไร ซึ่งเราจะทำการทดลองในลักษณะเช่นนี้เป็นจำนวนทั้งหมด 3 ครั้ง แล้วหาค่าเฉลี่ยของจุดศูนย์กลางของวัตถุพร้อมทั้งหาค่าฮิสเตอรีซีสด้วยการเลือกค่าจุด Coordinate (x,y) จุดศูนย์กลางของวัตถุสูงสุดลบกับค่าต่ำสุดเพื่อหาว่าระยะห่างระหว่างพิกเซลที่เกิดขึ้นว่ามีความคลาดเคลื่อนไปจากเดิมเท่าไร ต่อมาเราจะลดค่าความกว้างของสัญญาณพัลส์ลงทีละ 50 µs จะได้เป็น 1400 และ 1350 µs แล้วทำการทดลองเพื่อหาค่าเฉลี่ยของจุดศูนย์กลางของวัตถุและค่าฮิสเตอรีซีสของเซอร์โวมอเตอร์เหมือนที่ได้อธิบายไว้ข้างต้น ซึ่งภาพที่ 5.11 (ข) และ (ค) แสดงการหมุนของเซอร์โวมอเตอร์ในแนวแกน X ทางด้านซ้ายมือไปยังตำแหน่งที่มีค่าความกว้างของสัญญาณพัลส์เป็น 1400 และ 1350 µs ตามลำดับ เมื่อเราทำการทดลองครบเป็นจำนวน 3 ครั้งแล้ว จากนั้นทำการหาค่าเฉลี่ยการเคลื่อนที่ในแนวแกน X ที่ความกว้างของสัญญาณพัลส์ค่าต่างๆ และหาค่าผลต่างของการเคลื่อนที่ในแต่ละช่วง ดังแสดงในตารางที่ 5.1

(ก) ความกว้างพัลส์ = 1450 µs (ข) ความกว้างพัลส์ = 1400 µs (ค) ความกว้างพัลส์ = 1350 µs

ภาพที่ 5.11 กล้องหมุนในแนวแกน X ทางด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 การหาค่าฮิสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านซ้าย โดยใช้ความกว้างพัลส์ 1500~1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน X (X_{max}-X_{min})	X เฉลี่ย	ผลต่างแนวแกน X (Pixels)
1500	154.5, 122	154, 123	153, 121	1.5	153.8	$x = \|153.8 - 198.5\|$ **44.7**
1450	198,121	200, 121	197.5, 121	2.5	198.5	
1500	154, 118	155, 117	151, 117	3	153.3	$x = \|153.3 - 233.2\|$ **79.9**
1400	233.5,110.5	235.5, 111	230.5, 118	5	233.2	
1500	155, 120	155, 126	156, 121	1	155.3	$x = \|155.3 - 265\|$ **109.7**
1350	265, 124.5	265,124.5	265, 124.5	0	265	

4. หมุนเซอร์โวมอเตอร์ในแนวแกน X ไปทางด้านขวามือ โดยการเพิ่มค่าความกว้างของสัญญาณพัลส์จาก 1500 µs เป็น 1550, 1600 และ 1650 ตามลำดับ แล้วหาจุด Coordinate (x,y) ของจุดศูนย์กลางของวัตถุดังภาพที่ 5.12 แล้วหาค่าเฉลี่ยของจุดศูนย์กลางของวัตถุและค่าฮิสเตอรีซีสดังตารางที่ 5.2 ซึ่งวิธีการทดลองเหมือนกับข้อที่ 3 ต่อมาจะหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน X ทางด้านซ้ายมือและขวามือ ซึ่งพิจารณาจากค่าความชันของกราฟ โดยเราจะให้ผลต่างความกว้างของสัญญาณพัลส์ (50, 100, 150 µs) เป็นแกน Y และผลต่างของพิกเซลซึ่งได้จากตารางที่ 5.1 เป็นแกน X จากตารางที่ 5.3 ความไวในแนวแกน X ทางด้านซ้ายมีเท่ากับ 1.5 พัลส์ต่อพิกเซล ส่วนความไวในแนวแกน X ทางด้านขวาก็มีวิธีการคิดเช่นเดียวกับทางด้านซ้าย ซึ่งมีค่าเท่ากับ -1.8 พัลส์ต่อพิกเซล

(ก) ความกว้างพัลส์ = 1550 µs (ข) ความกว้างพัลส์ = 1600 µs (ค) ความกว้างพัลส์ = 1650 µs

ภาพที่ 5.12 กล้องหมุนในแนวแกน X ทางด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านขวา โดยใช้ความกว้างพัลส์ 1500~1650 µs โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1

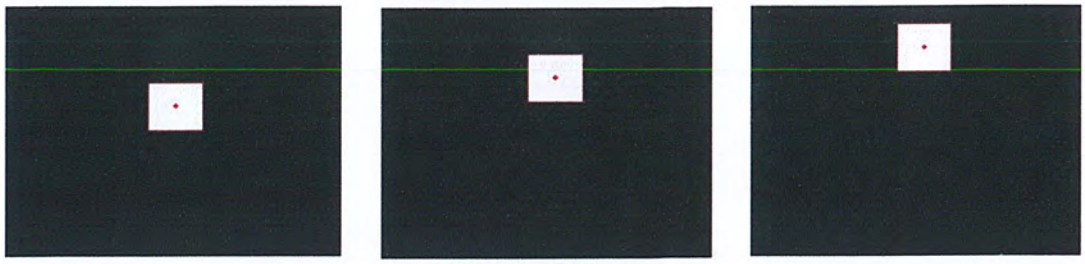
ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน X (X_{max}-X_{min})	X เฉลี่ย	ผลต่างแนวแกน X (Pixels)
1500	160.5, 95	160,94	162, 95	2	160.8	$x = \|160.8 - 64.6\|$ **96.2**
1650	64, 89.5	67, 91	63, 94.5	4	64.6	
1500	162, 107	166, 102	163.5, 96	4	163.8	$x = \|163.8 - 93.5\|$ **70.3**
1600	94.5, 106.5	93, 98.5	93, 94	1.5	93.5	
1500	162.5, 97	162, 96	161, 96	1.5	161.8	$x = \|161.8 - 120.3\|$ **41.5**
1550	121, 94	120, 94.5	120, 94.5	1	120.3	

ตารางที่ 5.3 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน X (Pulse/Pixel) ชุดที่ 1

ทิศทางการหมุนของเซอร์โวมอเตอร์	ความกว้างของสัญญาณพัลส์ ที่ต่างกัน	ความต่างของ pixels	ความไวของเซอร์โวมอเตอร์(Pulse/Pixel)
ด้านขวา	50	96.2	-1.8
	100	70.3	
	150	41.4	
ด้านซ้าย	50	44.7	1.5
	100	79.9	
	150	109.7	

5. ทำการทดลองเช่นเดียวกับเซอร์โวมอเตอร์ในแนวแกน X ด้วยการหมุนเซอร์โวมอเตอร์ในแนวแกน Y ขึ้นด้านบนและลงด้านล่างโดยการลดและเพิ่มความกว้างพัลส์ทีละ 50 µs จะได้เป็น 1350, 1400, 1450 µs (ด้านบน) และ 1550, 1600, 1650 µs (ด้านล่าง)จากนั้นหาจุด Coordinate (x,y) ของจุดศูนย์กลางของวัตถุดังภาพที่ 5.13~5.14 พร้อมทั้งหาค่าเฉลี่ยของจุดศูนย์กลางของวัตถุและค่าฮิสเตอรีซีสดังตารางที่ 5.4~5.5 ซึ่งวิธีการทดลองเหมือนกับเซอร์โวมอเตอร์ในแนวแกน X รวมทั้งหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน Y ทางด้านบนและด้านล่างดังแสดงในตารางที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ความกว้างพัลส์ = 1450 µs (ข) ความกว้างพัลส์ = 1400 µs (ค) ความกว้างพัลส์ = 1350 µs

ภาพที่ 5.13 กล้องหมุนในแนวแกน Y ทางด้านบน

(ก) ความกว้างพัลส์ = 1550 µs (ข) ความกว้างพัลส์ = 1600 µs (ค) ความกว้างพัลส์ = 1650 µs

ภาพที่ 5.14 กล้องหมุนในแนวแกน Y ทางด้านล่าง

ตารางที่ 5.4 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านบน โดยใช้ ความกว้างพัลส์ 1500 –1650 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 1

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน Y (Y_{max}-Y_{min})	Y เฉลี่ย	ผลต่างแนวแกน Y (Pixels)
1500	171.5, 120	170.5, 115	165.5, 117	5	117.3	$x = \|117.3 - 224.3\|$ **107**
1650	175, 224.5	172, 224.5	167, 224	0.5	224.3	
1500	169, 117.5	169.5,118.5	169.5, 119	1.5	118.3	$x = \|118.3 - 195.6\|$ **77.3**
1600	171.5,196.5	172.5,195.5	171.5, 195	1.5	195.6	
1500	169.5,119.5	167.5, 119	166.5,119.5	0.5	119.3	$x = \|119.3 - 168\|$ **48.7**
1550	171, 168	167, 168	167, 168	0	168	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.5 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านล่าง โดยใช้ความกว้างพัลส์ 1500~1350 โดยใช้กล้องและเซอร์โวชุดที่ 1

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน Y (Y_{max}-Y_{min})	Yเฉลี่ย	ผลต่างแนวแกน Y (Pixels)
1500	166.5, 140	165, 141.5	164, 141.5	1.5	141	$x = \|141 - 90.2\|$ **50.2**
1450	165.5, 90	164.5, 90	169.5, 90.5	0.5	90.2	
1500	170.5,142.5	169.5,141.5	167.5, 141	1.5	141.6	$x = \|141.6 - 58.3\|$ **83.3**
1400	169.5, 58	168.5, 58	164.5, 59	1	58.3	
1500	163.5,140.5	169,138.5	169, 137.5	3	138.8	$x = \|138.8 - 27.2\|$ **111.6**
1350	162, 26	168, 27	168.5, 28.5	2.5	27.2	

ตารางที่ 5.6 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน Y (Pulse/Pixel) ชุดที่ 1

ทิศทางการหมุนของเซอร์โวมอเตอร์	ความกว้างของสัญญาณพัลส์ ที่ต่างกัน	ความต่างของ pixels	ความไวของเซอร์โวมอเตอร์(Pulse/Pixel)
ด้านบน	50	107	-1.7
	100	77.3	
	150	48.7	
ด้านล่าง	50	50.9	1.6
	100	83.3	
	150	111.7	

เนื่องจากในโครงงานนี้เราใช้กล้องด้วยกันทั้งหมด 2 ชุดซึ่งจะมีเซอร์โวมอเตอร์จำนวน 4 ตัว ดังนั้นการทดลองคุณลักษณะของเซอร์โซมอเตอร์ในชุดที่ 2 ก็มีวิธีการในการทดลองและหาค่าต่างๆ เช่นเดียวกับชุดที่ 1 ทุกประการ โดยรายละเอียดแสดงดังตารางที่ 5.7~5.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.7 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านซ้าย โดยใช้ความกว้างพัลส์ 1500~1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน X (X_{max}-X_{min})	X เฉลี่ย	ผลต่างแนวแกน X (Pixels)
1500	165.5, 121	154, 122	153, 111	12.5	157.5	$x = \|157.5 - 199.2\|$ **41.7**
1450	199,121	200, 111	198.5, 111	1.5	199.2	
1500	154, 111	154, 113	151, 119	3	153	$x = \|153 - 232.2\|$ **79.2**
1400	232.5,111.5	231.5, 112	232.5, 119	1	232.2	
1500	154, 119	154, 126	154, 126	0	154	$x = \|154 - 267\|$ **113**
1350	267, 126.5	267,126.5	267, 125.5	0	267	

ตารางที่ 5.8 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน X ทางด้านขวา โดยใช้ความกว้างพัลส์ 1500~1650 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน X (X_{max}-X_{min})	X เฉลี่ย	ผลต่างแนวแกน X (Pixels)
1500	166.5, 96	166,94	167, 95	1	166.5	$x = \|166.5 - 55.3\|$ **111.2**
1650	54, 89.5	57, 90	55, 96.5	2	55.3	
1500	167, 108	168, 101	165.5, 95	2.5	166.8	$x = \|166.8 - 90.8\|$ **76**
1600	92.5, 105.5	90, 97.5	90, 93	2.5	90.8	
1500	167.5, 96	167, 96	166, 96	1.5	166.8	$x = \|166.8 - 124.3\|$ **42.5**
1550	125, 94	124, 94.5	124, 94.5	1	124.3	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน X (Pulse/Pixel) ชุดที่ 2

ทิศทางการหมุนของเซอร์โวมอเตอร์	ความกว้างของสัญญาณพัลส์ ที่ต่างกัน	ความต่างของ pixels	ความไวของเซอร์โวมอเตอร์(Pulse/Pixel)
ด้านขวา	50	111.2	-1.4
	100	76	
	150	42.5	
ด้านซ้าย	50	41.6	1.4
	100	79.1	
	150	113	

ตารางที่ 5.10 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านบน โดยใช้ความกว้างพัลส์ 1500 ~1650 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่2 Coordinate (x,y)	ค่าcgครั้งที่3 Coordinate (x,y)	ฮิสเตอรีซีส แนวแกน Y (Y_{max}-Y_{min})	Y เฉลี่ย	ผลต่างแนวแกน Y (Pixels)
1500	153, 124	160, 110.5	157, 111.5	13.5	115.3	$x = \|115.3 - 152\|$ **36.7**
1650	164, 152	160.5, 151	157.5, 153	2	152	
1500	153,109	159, 108	160, 107	2	108	$x = \|108 - 180.8\|$ **72.8**
1600	155.5, 181	163.5, 180	161.5,181.5	1.5	180.8	
1500	157, 107	154, 107.5	151, 106	1.5	106.8	$x = \|106.8 - 210.6\|$ **103.8**
1550	158, 211	155.5, 211	163, 210	1	210.6	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.11 การหาค่าฮีสเตอร์รีซีสและการหาจำนวนพัลส์ต่อพิกเซลในแนวแกน Y ทางด้านล่าง โดยใช้ความกว้างพัลส์ 1500~1350 โดยใช้กล้องและเซอร์โวมอเตอร์ชุดที่ 2

ความกว้างพัลส์ (µs)	ค่าcgครั้งที่ 1 Coordinate (x,y)	ค่าcgครั้งที่ 2 Coordinate (x,y)	ค่าcgครั้งที่ 3 Coordinate (x,y)	ฮิสเตอรีซีสแนวแกน Y (Y_{max}-Y_{min})	Y เฉลี่ย	ผลต่างแนวแกน Y (Pixels)
1500	167.5, 100	163, 105	159.5, 105	5	103.3	$x = \|103.3 - 62.8\|$ **40.5**
1450	163.5, 62.5	160.5, 62.5	156.5, 63.5	1	62.8	
1500	158, 106	167, 107	164, 110	4	107.6	$x = \|107.6 - 42.2\|$ **65.4**
1400	163, 36.5	161.5, 47.5	160, 42.5	11	42.2	
1500	161, 118.5	155, 119	160, 121	2.5	119.5	$x = \|119.5 - 17.8\|$ **101.7**
1350	154.5, 17.5	154, 18	121, 18	1.5	17.8	

ตารางที่ 5.12 การหาค่าความไวของเซอร์โวมอเตอร์ในแนวแกน Y (Pulse/Pixel) ชุดที่ 2

ทิศทางการหมุนของเซอร์โวมอเตอร์	ความกว้างของสัญญาณพัลส์ ที่ต่างกัน	ความต่างของ pixels	ความไวของเซอร์โวมอเตอร์(Pulse/Pixel)
ด้านบน	50	36.7	1.5
	100	72.8	
	150	103.8	
ด้านล่าง	50	40.5	1.6
	100	65.5	
	150	101.7	

สรุปการหาคุณลักษณะของเซอร์โวมอเตอร์ด้วยการตรวจสอบค่าความไวและค่าฮิสเตอรีซีสของเซอร์โวมอเตอร์แต่ละตัว เราจะพบว่าแม้ว่าเซอร์โวมอเตอร์ที่ใช้จะเป็นยี่ห้อเดียวกัน ขนาดเท่ากัน แต่ก็มีค่าความไวและค่าฮิสเตอรีซีสที่แตกต่างกัน ดังนั้นการหาค่าความไวและค่าฮิสเตอรีซีสจึงเป็นขั้นตอนหนึ่งที่สำคัญในการติดตามคนในภาพ สำหรับค่าความไวของเซอร์โวมอเตอร์แต่ละตัวจะถูกนำไปคูณกับค่าผลต่างของพิกเซลในแนวแกน X และแกน Y เพื่อให้คนที่กำลังถูกติดตามกลับมาอยู่ใกล้เคียงกับจุดศูนย์กลางของภาพ ส่วนค่าฮีสเตอรีซีสที่ได้จากการทดลองนั้นเราจะไม่นำมาใช้เพราะว่าเราได้กำหนดขอบเขตของวัตถุไว้ กล่าวคือแม้ว่าคนในภาพจะมีการเคลื่อนที่ แต่ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คนที่กำลังติดตามยังอยู่ในขอบเขตที่กำหนด เซอร์โวมอเตอร์ก็จะไม่หมุนแต่กล้องยังคงถ่ายภาพและติดตามต่อไปจนกว่าคนจะอยู่นอกขอบเขตที่กำหนด เซอร์โวมอเตอร์ก็จะหมุนโดยทำให้คนกลับมาอยู่ใกล้กับจุดศูนย์กลางอีกครั้ง

5.2 ผลการทดลองทางด้านซอฟต์แวร์ (Software Experiments)

5.2.1 การหาสีผิว (Skin Detection)

จากทฤษฎีที่ได้กล่าวไว้ในบทที่ 4 การตรวจหาสีผิวโดยการตัดค่า Threshold จากค่า r g b ที่ได้จากการ Normalize ภาพสี RGB ซึ่งจะนำภาพถ่ายคนที่อยู่ภายในห้อง ภายนอกห้องและภาพถ่ายคนจากแหล่งอื่นมาประมวลผลด้วยการเลือกใช้ค่า Threshold ดังแสดงในตารางที่ 5.13 ซึ่งได้จากการกำหนดอัตราส่วนโทนสีผิวโดยแบ่งออกเป็นสามกรณีคือกรณีโทนสีผิวอ่อนต่อสีผิวเข้มเป็น 20%:80% 50%:50% และ 80%:20% ตามลำดับ โดยทดลองกับทั้ง 4 โมเดลคือ rg, rb, gb และ rgb

ตารางที่ 5.13 แสดงค่า Threshold ของ r g และ b ในแต่ละกรณีของการใช้ตัวอย่างสีผิว

กรณีตัวอย่างสีผิว	ค่า r	ค่า g	ค่า b
20%:80%	0.3464~0.6857	0.2659~0.3768	0.0122~0.3374
50%:50%	0.3595~0.6854	0.2720~0.3747	0.0023~0.2944
80%:20%	0.3463~0.6617	0.2967~0.3686	0.0200~0.3670

ภาพถ่ายคนคนที่อยู่ภายในห้อง (Indoor)

(ก) ภาพที่ 1 (ข) ภาพที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพถ่ายคนคนที่อยู่ภายนอกห้อง (Outdoor)

(ค) ภาพที่ 3

(ง) ภาพที่ 4

ภาพถ่ายคนคนจากแหล่งอื่น

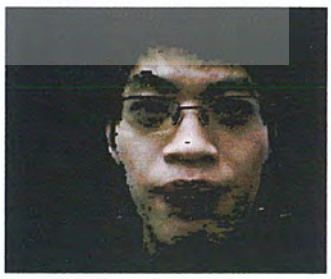
(จ) ภาพที่ 5

(ฉ) ภาพที่ 6

ภาพที่ 5.15 ภาพต้นฉบับ

(ก) กรณี 20%:80%

(ข) กรณี 50%:50%



(ค) กรณี 80%:20%

ภาพที่ 5.16 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.17 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rg

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.18 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rg

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.19 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.20 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rg

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.21 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rg

ตัวอย่างภาพต้นฉบับที่ใช้ในการทดลอง แสดงดังภาพที่ 5.15 ซึ่งเราจะใช้ภาพต้นฉบับทั้งภาพคนที่ถ่ายในห้อง และนอกห้อง รวมทั้งภาพคนที่มาจากแหล่งอื่นด้วย ทั้งนี้ก็เพื่อทดสอบว่าโมเดลใดให้ความถูกต้องในการตรวจหาสีผิวได้มากที่สุด และตัวอย่างสีผิวในอัตราส่วนโทนผิวสีอ่อนต่อผิวสีเข้มในอัตราส่วนเท่าใดที่เหมาะสมกับการตรวจหาสีผิว

ภาพที่ 5.16 แสดงภาพถ่ายคนที่อยู่ภายในห้องเมื่อใช้โมเดล rg จากผลการทดลองจะพบว่าในกรณีที่ใช้ตัวอย่างสีผิว 20%:80% สามารถหาสีผิวได้ใกล้เคียงกับกรณีที่ใช้ตัวอย่างสีผิว 80%:20% ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50% นั้นบริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่มากกว่ากรณีอื่นภาพที่ 5.17 จะพบว่ากรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีผิวออกจากสีพื้นหลังได้ทั้งหมด แม้ว่าบริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่บ้าง โดยบริเวณที่เป็นสีผิวจะมีความใกล้เคียงกับกรณีที่ใช้ตัวอย่างสีผิว 20%:80%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับภาพถ่ายคนที่อยู่ภายนอกห้องดังแสดงในภาพที่ 5.18 และภาพที่ 5.19 พบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% สามารถตรวจหาสีผิวได้ดีที่สุดเพราะว่ามีสีดำปะปนอยู่บริเวณปากและคางเพียงเล็กน้อย ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50% และ 80%:20% จะมีสีดำปะปนอยู่บริเวณจมูกและตาตามลำดับ อย่างไรก็ตาม กรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีผิวกับสีพื้นหลังได้ชัดเจนกว่ากรณีอื่นๆซึ่งมีสีเทาของพื้นหลังปรากฏอยู่ด้วย

ภาพที่ 5.20 และภาพที่ 5.21 เป็นผลลัพธ์ของภาพถ่ายที่นำมาจากแหล่งอื่น ซึ่งจะพบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% สามารถหาสีผิวได้ดีกว่าทุกกรณี ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50% จะมีสีดำปะปนอยู่บ้างแต่เมื่อเปรียบเทียบกับกรณีที่ใช้ตัวอย่างสีผิว 80%:20% แล้วจะพบว่าบริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่มากกว่า สรุปผลการทดลองของภาพคนที่ถ่ายภายในและภายนอกห้อง รวมทั้งภาพที่มาจากแหล่งอื่น สำหรับโมเดล rg กรณีที่ใช้ตัวอย่างสีผิว 20%:80% สามารถตรวจหาสีผิวได้ดีที่สุด

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.22 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.23 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.24 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.25 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.26 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.27 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rb

ภาพที่ 5.22 แสดงผลลัพธ์ของภาพถ่ายคนที่อยู่ภายในห้อง เมื่อนำมาตรวจหาสีผิวโดยใช้โมเดล rb จะพบว่าทั้งกรณีที่ใช้ตัวอย่างสีผิว 20%:80% และ 80%:20% มีความสามารถในการหาสีผิวได้ใกล้เคียงกันและมีสีพื้นหลังปรากฏอยู่บ้างเล็กน้อย ส่วนภาพที่ 5.23 จะเห็นว่าทั้งสามกรณีมีความสามารถในการหาสีผิวได้ใกล้เคียงกันเพราะบริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่เท่าๆกัน รวมทั้งบริเวณที่เป็นพื้นหลังของกรณีที่ใช้ตัวอย่างสีผิว 20%:80% และ 80%:20% ก็ปรากฏขึ้นมาในจำนวนที่เท่ากันด้วย ส่วนในกรณีที่ใช้ตัวอย่างสีผิว 50%:50% พบว่าบริเวณที่เป็นสีผิวสามารถแยกออกจากสีพื้นหลังได้ทั้งหมด

ภาพที่ 5.24 และภาพที่ 5.25 แสดงผลลัพธ์ของภาพถ่ายคนที่อยู่ภายนอกห้องพบว่าในกรณีที่ใช้ตัวอย่างสีผิว 20%:80% และ 80%:20% สามารถตรวจหาสีผิวได้ดีที่สุดซึ่งคุณภาพของภาพที่ได้มีความใกล้เคียงกัน แต่จากภาพพบว่าทั้งสามกรณีจะปรากฏสีเขียวของพื้นหลังโดยกรณีที่ใช้ตัวอย่างสีผิว 50%:50% จะเห็นสีพื้นหลังปรากฏน้อยกว่ากรณีอื่น

ภาพที่ 5.26 เป็นผลลัพธ์ของภาพถ่ายที่นำมาจากแหล่งอื่น ซึ่งจะพบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% และ 80%:20% สามารถหาสีผิวได้ใกล้เคียงกัน ส่วนกรณีที่ใช้สีผิว 50%:50% จะมีสีดำปะปนอยู่มากกว่าโดยบริเวณที่ขาไม่มีสีผิวปรากฏอยู่เลย และเมื่อเปรียบเทียบสีพื้นหลังกับกรณีอื่นจะเห็นว่าเป็นสีดำทั้งหมดแต่สองกรณีแรกนั้นมีสีซึ่งใกล้เคียงกับสีผิวปรากฏอยู่ ส่วนภาพที่ 5.27 ก็มีลักษณะเช่นเดียวกันกับภาพที่ 5.20 เพียงแต่ว่าสีพื้นหลังซึ่งเป็นสีเหลืองและสีเขียวปรากฏอยู่ทั้งสามกรณี ดังนั้นโมเดล rb ที่สามารถตรวจหาสีผิวได้เหมาะสมที่สุดคือกรณีที่ใช้ตัวอย่างสีผิว 20%:80% อย่างไรก็ตาม กรณีที่ใช้ตัวอย่างสีผิว 50%:50% นั้นสามารถแยกสีพื้นหลังออกจากสีผิวได้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.28 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล gb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.29 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล gb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.30 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล gb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.31 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล gb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.32 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล gb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.33 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล gb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 5.28 และภาพที่ 5.29 แสดงผลลัพธ์ของภาพถ่ายคนที่อยู่ภายในห้อง เมื่อนำมาตรวจหาสีผิวโดยใช้โมเดล gb จะพบว่าในกรณีที่ใช้ตัวอย่างสีผิว 20%:80% ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้โดยสีพื้นหลังจะเปลี่ยนเป็นสีขาว ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีผิวออกจากสีพื้นหลังได้แต่ในบริเวณที่เป็นสีผิวมีสีดำปะปนอยู่มากกว่ากรณีอื่น และกรณีที่ใช้กรณีที่ใช้ตัวอย่างสีผิว 80%:20% สามารถแยกสีผิวได้เช่นเดียวกันแต่บริเวณสีผิวจะมีสีดำปะปนน้อยกว่ากรณีอื่น

จากภาพที่ 5.30 และภาพที่ 5.31 เป็นผลลัพธ์จากภาพถ่ายคนที่อยู่ภายนอกห้อง ซึ่งพบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้ เพราะจะเห็นว่าสีขาวของเสื้อและพื้นหลังยังคงปรากฏอยู่ ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50%จะพบว่าสีเขียวของพื้นหลังปรากฏอยู่บ้างเล็กน้อยแต่บริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่จำนวนมาก ส่วนกรณีที่ใช้ตัวอย่างสีผิว 80%:20% มีลักษณะเช่นเดียวกันเพียงแต่บริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่น้อยกว่ากรณีที่ใช้ตัวอย่างสีผิว 50%:50%

จากภาพที่ 5.32 เป็นผลลัพธ์ของภาพถ่ายที่นำมาจากแหล่งอื่น ซึ่งพบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% ไม่สามารถแยกสีผิวออกจากสีพื้นหลังได้ทั้งหมด เพราะว่ามีสีขาวของกางเกงปรากฏขึ้นมาเช่นเดียวกับภาพที่ 5.33 ที่ปรากฏสีขาวของเก้าอี้มาด้วย ส่วนกรณีที่ใช้ตัวอย่างสีผิว 50%:50% จะพบว่าสามารถแยกสีผิวออกจากสีพื้นหลังได้ชัดเจน เพียงแต่บริเวณที่เป็นสีผิวจะมีสีดำปะปนอยู่เช่นเดียวกับกรณีที่ใช้ตัวอย่างสีผิว 80%:20% แต่กรณีนี้สีผิวมีสีดำปะปนมากกว่า ดังนั้น โมเดล gb เมื่อเปรียบเทียบกันแล้วกรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถตรวจหาสีผิวได้เหมาะสมที่สุดในโมเดลนี้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

รูปที่ 5.34 ภาพผลลัพธ์ของภาพที่ 1 เมื่อใช้โมเดล rgb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.35 ภาพผลลัพธ์ของภาพที่ 2 เมื่อใช้โมเดล rgb

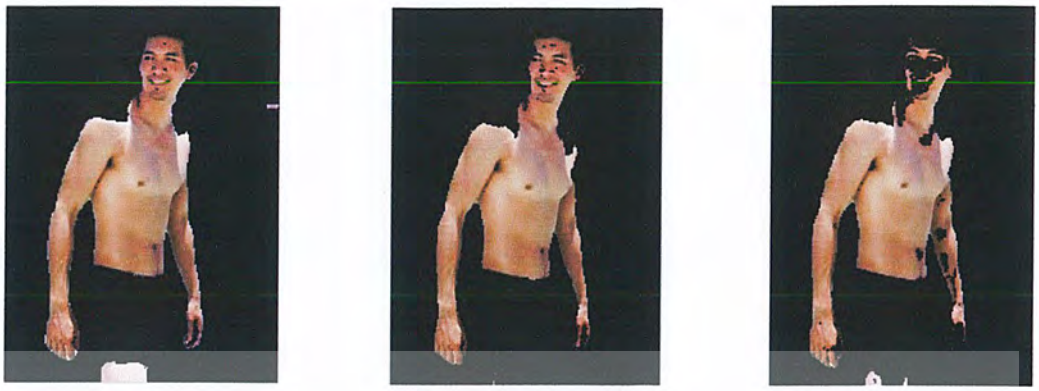
(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.36 ภาพผลลัพธ์ของภาพที่ 3 เมื่อใช้โมเดล rgb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.37 ภาพผลลัพธ์ของภาพที่ 4 เมื่อใช้โมเดล rgb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.38 ภาพผลลัพธ์ของภาพที่ 5 เมื่อใช้โมเดล rgb

(ก) กรณี 20%:80% (ข) กรณี 50%:50% (ค) กรณี 80%:20%

ภาพที่ 5.39 ภาพผลลัพธ์ของภาพที่ 6 เมื่อใช้โมเดล rgb

จากภาพที่ 5.34 แสดงผลลัพธ์ของภาพถ่ายคนที่อยู่ภายในห้องเมื่อนำมาตรวจหาสีผิวโดยใช้โมเดล rgb จะพบว่าทุกกรณีสามารถแยกสีผิวออกจากสีพื้นหลังได้ แต่มีความแตกต่างกันตรงที่ความสมบูรณ์ของสีผิว โดยกรณีที่ใช้ตัวอย่างสีผิว 20%:80% จะมีความสมบูรณ์ของสีผิวมากที่สุด ส่วนภาพที่ 5.35 กรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีผิวออกจากสีพื้นหลังได้ดีที่สุดและสีผิวมีความสมบูรณ์เทียบเท่ากับกรณีที่ใช้ตัวอย่างสีผิว 20%:80%

จากภาพที่ 5.36 และ 5.37 เป็นผลลัพธ์ของภาพถ่ายคนที่อยู่ภายนอกห้อง ซึ่งจะพบว่ากรณีที่ใช้ตัวอย่างสีผิว 20%:80% ให้สีผิวที่สมบูรณ์ที่สุด แต่สีพื้นหลังที่เป็นสีเทายังคงปรากฏให้เห็นอยู่บ้างซึ่งต่างกับกรณีที่ใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีพื้นหลังออกได้หมด จะมีก็แต่สีของเก้าอี้และต้นไม้ที่ปรากฏให้เห็นเช่นเดียวกับกรณีที่ใช้ตัวอย่างสีผิว 80%:20% นอกจากนี้บริเวณที่เป็นสีผิวก็มีสีดำปะปนอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 5.38 และ 5.39 เป็นผลลัพธ์ของภาพถ่ายที่นำมาจากแหล่งอื่น ซึ่งจะพบว่าทั้งรูปที่ 5.32 และ 5.33 กรณีที่ใช้ตัวอย่างสีผิว 20%:80% แสดงสีผิวได้สมบูรณ์ที่สุด แต่ในภาพที่ 5.38 (ก) จะเห็นว่ามีสีพื้นหลังซึ่งใกล้เคียงกับสีผิวปรากฏขึ้นมาด้วย และในภาพที่ 5.39 (ก) บริเวณที่เป็นสีเสื้อก็มีบางส่วนที่ไม่เปลี่ยนเป็นสีดำซึ่งต่างจากกรณีที่ใช้ตัวอย่างสีผิว 50%:50% และ 80%:20% อย่างไรก็ตาม กรณีที่ใช้ตัวอย่างสีผิว 50%:50% และ 80%:20% ไม่สามารถตรวจหาสีผิวได้สมบูรณ์ มีสีดำปะปนอยู่บริเวณที่เป็นสีผิวของทั้งสองกรณี

สรุปผลการทดลองในกรณีของโมเดล rgb เมื่อใช้ตัวอย่างสีผิวโทนอ่อนต่อโทนเข้มเป็นอัตราส่วน 20%:80% สามารถตรวจหาสีผิวได้สมบูรณ์กว่ากรณีที่ใช้ตัวอย่างสีผิว 50%:50% และ 80%:20 % แต่ในการแยกสีผิวกับสีพื้นหลัง กรณีที่ใช้ตัวอย่างสีผิว 50%:50% บริเวณที่เป็นพื้นหลังสามารถแยกออกมาได้ความสมบูรณ์มากกว่ากรณีอื่น

จากการทดลองทำให้เราทราบว่าในโมเดลเดียวกันแต่ภาพต่างกันก็ให้ผลที่แตกต่างกันด้วย ดังนั้นจึงทำการทดลองกับภาพจำนวน 100 ภาพ ซึ่งเป็นภาพถ่ายคนภายในห้องจำนวน 39 ภาพ ภายนอกห้องจำนวน 40 ภาพและภาพคนจากแหล่งอื่นอีกจำนวน 21 ภาพ โดยนำมาเปรียบเทียบในเรื่องของสีผิวและสีพื้นหลังว่ามีความสมบูรณ์ในแต่ละกรณีคิดเป็นกี่เปอร์เซ็นต์ จากผลการทดลองสามารถสรุปได้ดังตารางที่ 5.2

<u>หมายเหตุ</u> ภาพคนที่ถ่ายทั้งภายในห้องและนอกห้อง รวมทั้งภาพคนจากแหล่งอื่นจำนวน 100 ภาพ ที่ใช้ในการทดลองแสดงไว้ในภาคผนวก (เฉพาะในแผ่น CD เท่านั้น)

ตารางที่ 5.14 สรุปโมเดลที่ใช้ในการตรวจหาสีผิวและกรณีเลือกใช้ตัวอย่างสีผิว

โมเดล	ตัวอย่างสีผิว 20%:80%		ตัวอย่างสีผิว 50%:50%		ตัวอย่างสีผิว 80%:20%	
	สีผิว	สีพื้นหลัง	สีผิว	สีพื้นหลัง	สีผิว	สีพื้นหลัง
rg	80%	15%	33%	41%	30%	5%
rb	63%	5%	30%	65%	48%	4%
rgb	75%	10%	34%	36%	25%	23%
gb	70%	2%	36%	48%	24%	46%

จากตารางที่ 5.2 โมเดลที่สามารถตรวจหาสีผิวได้ดีที่สุดคือโมเดล rg และเลือกกรณีที่ใช้ตัวอย่างสีผิวเป็น 20%:80% เพราะมีเปอร์เซ็นต์สูงสุดเป็น 80% แต่กรณีการเลือกใช้ตัวอย่างสีผิว 50%:50% สามารถแยกสีผิวกับสีพื้นหลังได้ดีที่สุดเป็น 65% ซึ่งค่า Threshold ที่เหมาะสมสำหรับโมเดล rg มีค่า Threshold ของ $r = 0.3464 \sim 0.6857$ และ $g = 0.2659 \sim 0.3768$ จากผลการทดลองที่แสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้างต้นจะเห็นได้ว่าสอดคล้องกับรายงานการศึกษาที่กล่าวไว้ในบทที่ 4 ที่เลือกใช้โมเดล rg ในการตรวจหาสีผิว

5.2.2 การเลือกตัวอย่างสีผิว

จากทฤษฎีในบทที่ 4 ในการเลือกตัวอย่างโทนสีผิวมีผลต่อการกำหนดค่า Threshold ที่เหมาะสม ดังนั้นในการเลือกตัวอย่างสีผิวจึงเลือกสีผิวโทนอ่อนต่อสีผิวโทนเข้มจากภาพถ่ายคนที่อยู่ภายในห้อง (Indoor) และภายนอกห้อง (Outdoor) โดยในแต่ละกรณีใช้ตัวอย่างสีผิวเป็นจำนวน 20 คน ดังภาพที่ 5.40~5.42

กรณีที่ 1 ตัวอย่างสีผิวเป็น 20% : 80% ซึ่งโทนผิวสีอ่อนเป็น 20% โดยเลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 2 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 2 คน ส่วนโทนผิวสีเข้มเป็น 80% เลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 8 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 8 คน

ภาพที่ 5.40 ตัวอย่างสีผิวในอัตราส่วน 20%:80%

กรณีที่ 2 ตัวอย่างสีผิวเป็น 50% : 50% ซึ่งโทนผิวสีอ่อนเป็น 50% โดยเลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 5 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 5 คน ส่วนโทนผิวสีเข้มเป็น 50% เลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 5 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 5 คน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 5.41 ตัวอย่างสีผิวในอัตราส่วน 50%:50%

กรณีที่ 3 ตัวอย่างสีผิวเป็น 80% : 20% ซึ่งโทนผิวสีอ่อนเป็น 80% โดยเลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 8 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 8 คน ส่วนโทนผิวสีเข้มเป็น 20% เลือกจากภาพถ่ายคนที่อยู่ภายในห้อง 2 คน และภาพถ่ายคนที่อยู่ภายนอกห้อง 2 คน

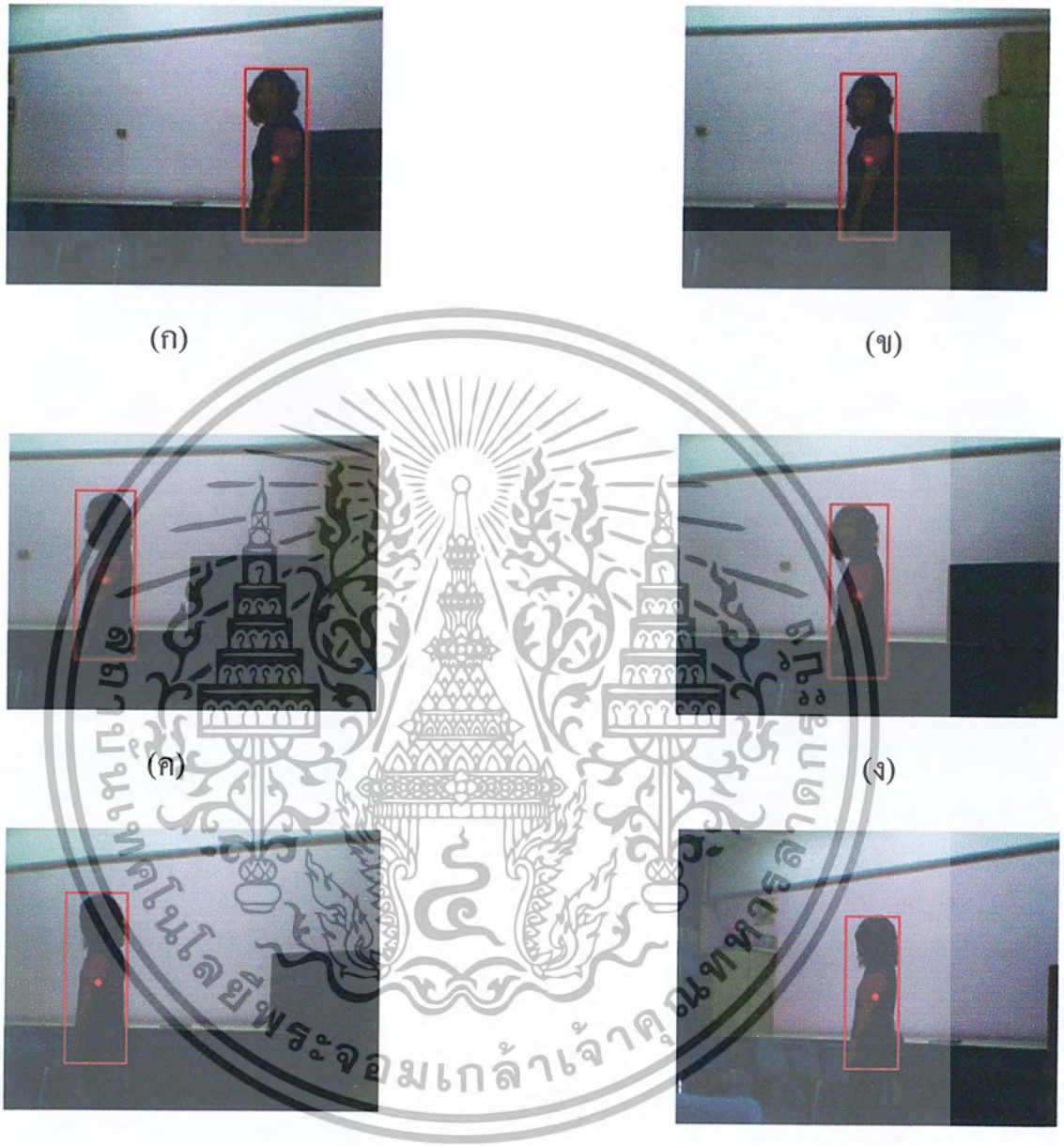
ภาพที่ 5.42 ตัวอย่างสีผิวในอัตราส่วน 80%:20%

จากภาพที่ 5.34~5.36 แสดงตัวอย่างการเลือกสีผิวทั้ง 3 กรณีคือ กรณีที่เลือกใช้ตัวอย่างสีผิวในอัตราส่วน 20%:80% 50%:50% และ80%:20% เพื่อนำมาวิเคราะห์ว่ากรณีใดที่เหมาะสมในการกำหนดเป็นค่า Threshold โดยการทดลองได้แสดงผลไว้ในข้อที่ 5.2.1 ซึ่งสามารถสรุปได้ว่าการเลือกตัวอย่างสีผิวในอัตราส่วน 20%:80% เหมาะสมที่สุดในการตรวจหาสีผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ทดลองติดตามคนที่เคลื่อนที่อยู่ในภาพ

ติดตามคนที่ใส่ปลอกแขนสีแดง



(ก) (ข)

(ค) (ง)

(จ) (ฉ)

ภาพที่ 5.43 แสดงการติดตามคนที่ใส่ปลอกแขนสีแดง

จากภาพที่ 5.43 (ก) เมื่อมีคนเข้ามาอยู่ในภาพกล้องจะทำการตีกรอบพร้อมทั้งหาจุดศูนย์กลางของคนแล้วนำไปลบกับจุดศูนย์กลางของภาพเพื่อหาผลต่างแล้วส่งค่าไปยังมอเตอร์เพื่อหมุนกล้องให้จุดศูนย์กลางของคนมาอยู่ตรงกับจุดศูนย์กลางของภาพดังภาพที่ 5.43 (ข) หลังจากนั้นเมื่อคนเคลื่อนที่ไปทางด้านซ้ายดังภาพที่ 5.43 (ค) และภาพที่ 5.43 (จ) โปรแกรมก็จะทำตามขั้นตอนเดียวกันกับภาพที่ 5.43 (ก) และภาพที่ 5.43 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและข้อเสนอแนะ

6.1 สรุปผลการทำโครงงาน

จากการศึกษาและทำการทดลองการติดตามคนที่กำลังเคลื่อนที่อยู่ในภาพ โดยเราจะใช้กล้องเวปแคมถ่ายภาพคน แล้วใช้โปรแกรม MATLAB ในการประมวลผลภาพเพื่อหาคนที่อยู่ในภาพ หลังจากนั้นโปรแกรมก็จะการตีกรอบคนเพื่อหาจุดศูนย์กลางเพื่อนำจุดศูนย์กลางของคนไปเปรียบเทียบกับจุดศูนย์กลางของภาพ แล้วแปลงเป็นค่าพัลซ์ส่งไปยังเซอร์โวมอเตอร์เพื่อให้เซอร์โวมอเตอร์หมุนกล้องเวปแคมติดตามคนได้

6.2 ข้อเสนอแนะ

เนื่องจากตอนนี้ระบบการทำงานของโปรแกรมยังต้องใช้เวลาในการประมวลผลนาน เนื่องจากโปรแกรมที่เขียนมานี้มีฟังก์ชันที่ใช้ในการประมวลผลเยอะและผู้ทำโครงงานยังขาดทักษะในการเขียนโปรแกรม อีกทั้งยังมีสัญญาณรบกวนในสภาพแวดล้อมจริงอยู่มาก

ดังนั้น จึงขอเสนอให้เลือกใช้เพียงฟังก์ชันใดฟังก์ชันหนึ่งในการประมวลผลและเขียนโปรแกรมให้รัดกุมมากกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษดา ใจยิ่น และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เขียนโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ PIC ด้วย mikro C คอมไพเลอร์และการทดลองเบื้องต้น.** กรุงเทพฯ: อินโนเวทีฟ เอ็กเพอริเมนต์, 2545.

ทีมงานสมาร์ทเลิร์นนิ่ง. **Advance PIC Microcontroller in C ประยุกต์ใช้งาน PIC ขั้นสูงด้วยภาษา C.** กรุงเทพฯ: สมาร์ทเลิร์นนิ่ง, 2553.

ทีมงานสมาร์ทเลิร์นนิ่ง. **PIC Microcontroller learning by doing ด้วยภาษา C.** กรุงเทพฯ: สมาร์ทเลิร์นนิ่ง, 2550.

ประจิน พลังสันติกุล. **เรียนรู้และการใช้งาน CCS C คอมไพเลอร์เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC.** กรุงเทพฯ: อินโนเวทีฟ เอ็กเพอริเมนต์, 2548.

ประจิน พลังสันติกุล. **PIC Works Example and C Source Code.** กรุงเทพฯ: แอพซอฟต์เทค, 2548.

ฤดี มาสุจันท์. **การควบคุมคุณภาพ.** พิมพ์ครั้งที่ 2. กรุงเทพฯ: แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2550.

วัชรินทร์ เคารพ. **คู่มือการใช้งาน SERVO MOTOR.** พิมพ์ครั้งที่ 1. กรุงเทพฯ: อีทีที, 2546.

D. Marius, S. Pennathur and K. Rose, **Face Detection Using Color Thresholding and Eigenimage Template Matching.** 2005.

H. K. Almohair, A. R. Ramli, A. M. Elsadig and S. J. Hashim, **Skin Detection in Luminance Images using Threshold Technique.** Int. Jour. of The Computer, the Internet and Management Vol.15, 2007.

R. C. Gonzalez and R. E. Woods. **Digital Image Processing.** Addison-Wesley Publishing Company, 1992.

V. Vezhnevets, V. Sazonov and A. Andreeva. **A Survey on Pixel-Based Skin Color Detection Techniques.** Moscow: 2003.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

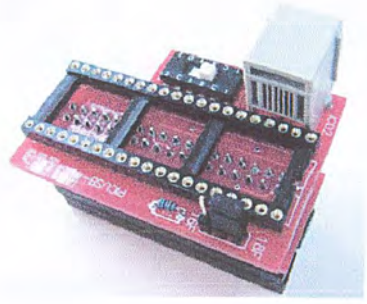
คู่มือการใช้งานอุปกรณ์ในการวิจัย

คุณสมบัติของ ET-PGMPIC USB V2.0

- รองรับการใช้งานกับไมโครคอนโทรเลอร์ตระกูล PIC
- เชื่อมต่อกับคอมพิวเตอร์ผ่าน USB Port
- ใช้ไฟเลี้ยงจาก USB Port (เฉพาะบอร์ด ET-PGMPIC USB เท่านั้น)
- มีพอร์ต ICSP สำหรับนำไปต่อโปรแกรมแบบ In-Circuit Serial Programming
- มีไฟแสดงสภาวะต่างๆ
- สามารถโปรแกรมโดยการกดสวิตช์ PROGRAM บนเครื่องโปรแกรม
- สามารถโปรแกรมผ่านโมดูล Adaptor Module ต่างๆ ได้

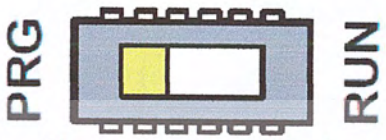
ชุดอุปกรณ์ Adaptor Module

ชุดอุปกรณ์เสริมของเครื่องโปรแกรม ET-PGM PIC USB V2.0 จุดประสงค์เพื่อรองรับการโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์ (TARGET Board) โดยไม่ต้องถอดไอซีเข้าออก เพิ่มความสะดวกในการพัฒนาโปรแกรม และ ช่วยป้องกันการหักงอของขาไอซี ที่มักเกิดขึ้นจากการถอดไอซีเข้า-ออกเครื่องโปรแกรม เป็นต้น ชุดโมดูล Adaptor คือ 40-PIN ดังรูป

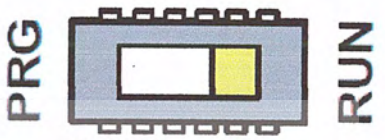


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโมดูลจะมีสวิตช์เลือกโหมด การโปรแกรม (PRG) และ โหมดการรัน (RUN) โดยเมื่อต้องการทำการโปรแกรมก็ให้เลือกสวิตช์มาที่ตำแหน่ง PRG และ เมื่อต้องการรัน ให้เลื่อนสวิตช์ไปที่ตำแหน่ง RUN ดังรูปต่อไปนี้



Programming Mode



Running Mode

ในบางโมดูลจะมีจัมพ์เปอร์ 18F/16F สำหรับเลือกเบอร์ของไมโครคอนโทรเลอร์ PIC จะต้องทำการเซตจัมพ์เปอร์ให้ตรงกับเบอร์ที่เราใช้งานดังรูปต่อไปนี้

สำหรับไมโครคอนโทรลเลอร์ตระกูล PIC1

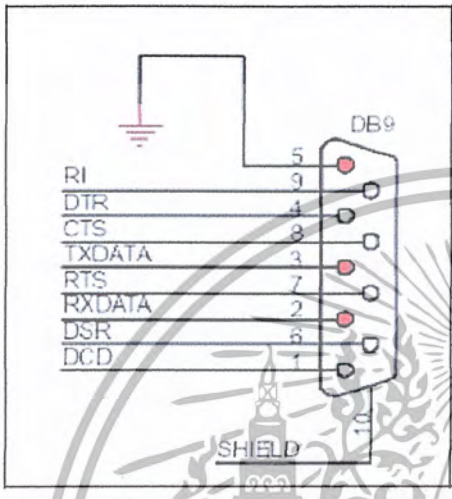
สำหรับไมโครคอนโทรลเลอร์ตระกูล PIC16FXXX (ด้านบน)

การใช้งาน ET-USB/RS232 MINI

การต่อใช้งาน ET-USB/RS232 MINI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เสียบ ET-USB/RS232 MINI เข้ากับ Port USB ของคอมพิวเตอร์ ขณะนี้จะสังเกตเห็นว่าไฟแสดงสถานะ PWR จะยังไม่ติดสว่าง
2. ติดตั้ง Driver สำหรับ ET-USB/RS232 MINI โดยจากคู่มือการติดตั้ง Driver เมื่อติดตั้งเรียบร้อยแล้ว ไฟแสดงสถานะ PWR จะติดสว่างแสดงว่าพร้อมที่จะใช้งานแล้ว
3. ต่อสายพอร์ตอนุกรม เพื่อใช้งานโดยตำแหน่งขาใช้งาน จะเป็นดังรูป



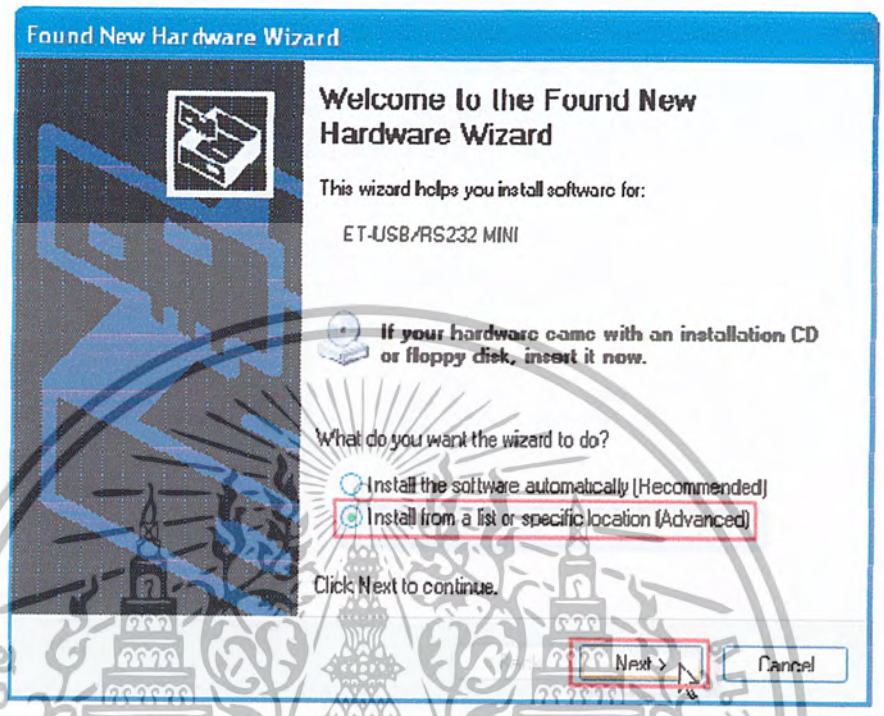
Pin	Name	Dir	Description
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmi. Data
4	DTR	→	Data Terminal Ready
5	GND	—	Signal Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

การติดตั้ง Driver ของ ET- USB/RS232 MINI

1. ทำการเชื่อมต่อ ET-USB/RS232 MINI เข้ากับคอมพิวเตอร์ทางพอร์ต USB จากนั้นวินโดวส์จะตรวจพบฮาร์ดแวร์ใหม่ดังรูป ตอนนี้จะสังเกตว่าไฟแสดงสถานะ PWR จะยังไม่ติดสว่าง

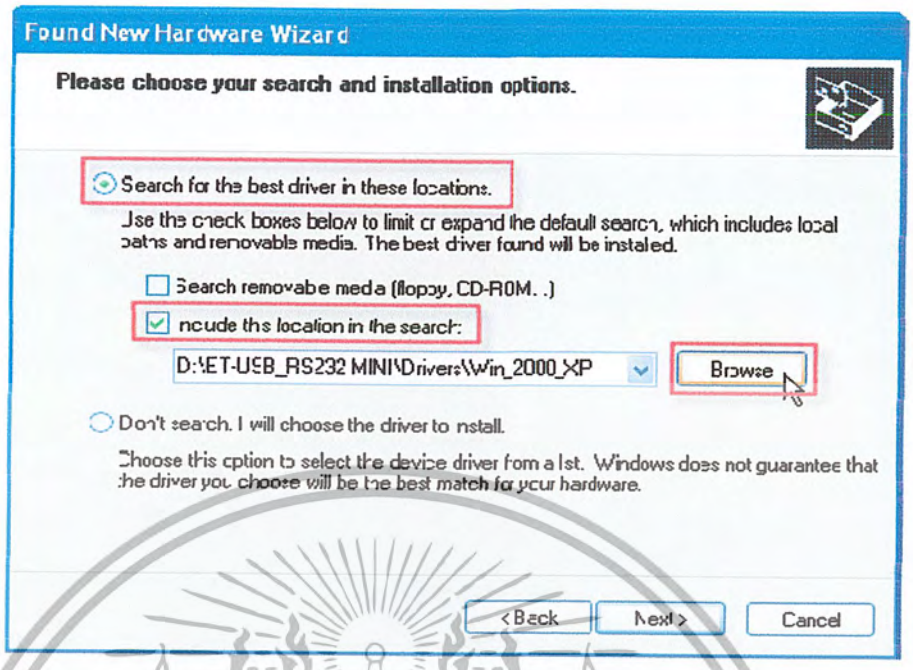
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

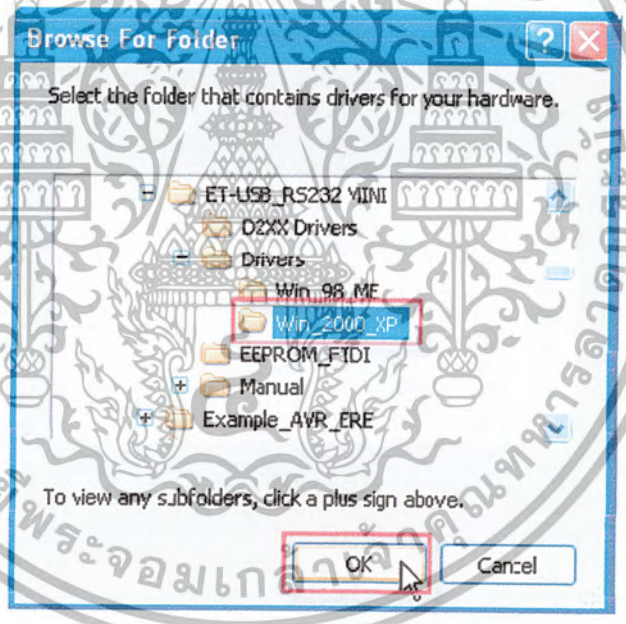
2. จากนั้นจะปรากฏหน้าต่าง Found New Hardware Wizard ให้เลือกที่ Install from a list or specific location (Advanced) และคลิก Next



3. ทำการเลือกดังรูป และคลิกปุ่ม Browse เพื่อระบุตำแหน่งที่เก็บ Driver จากนั้นคลิก OK

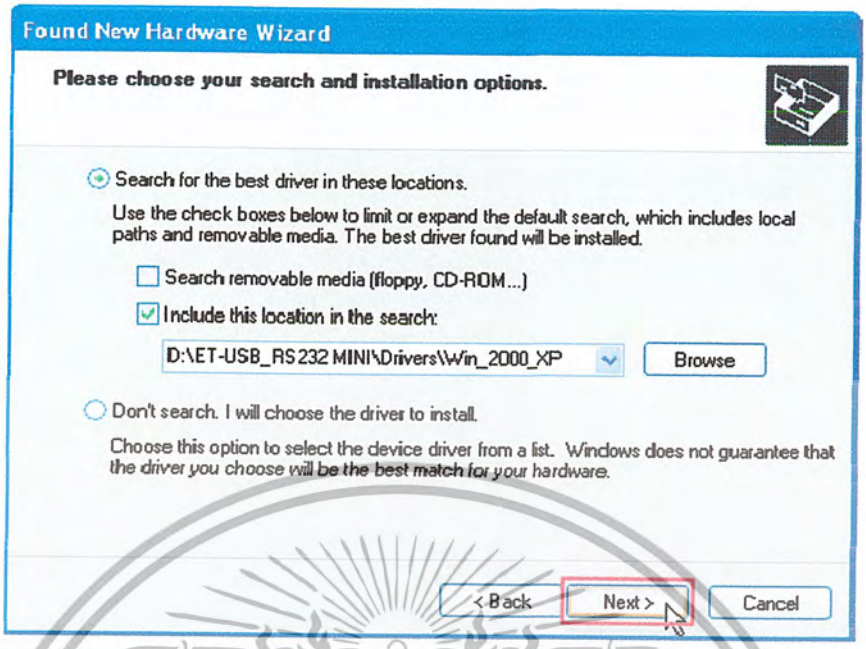
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



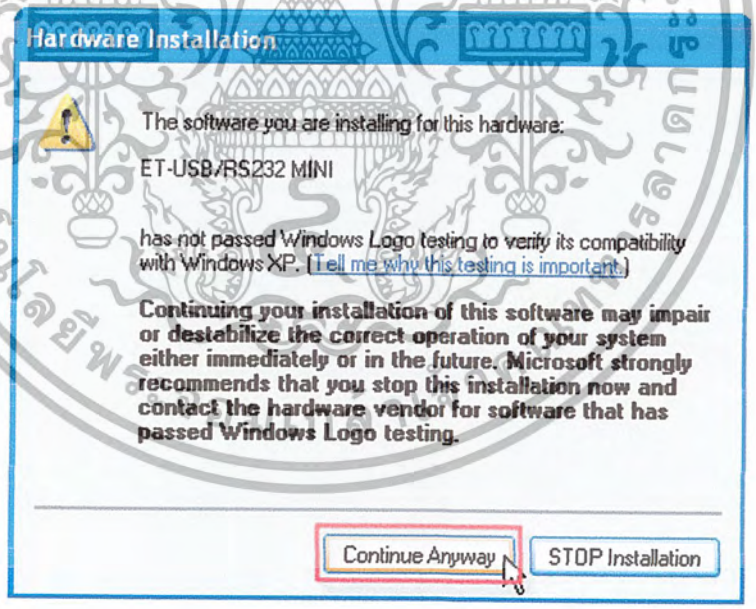


4. เมื่อทำการเลือกเรียบร้อยแล้วให้คลิกปุ่ม Next จากนั้นวินโดวส์จะทำการหาฮาร์ดแวร์เพื่อที่จะทำการติดตั้ง Driver

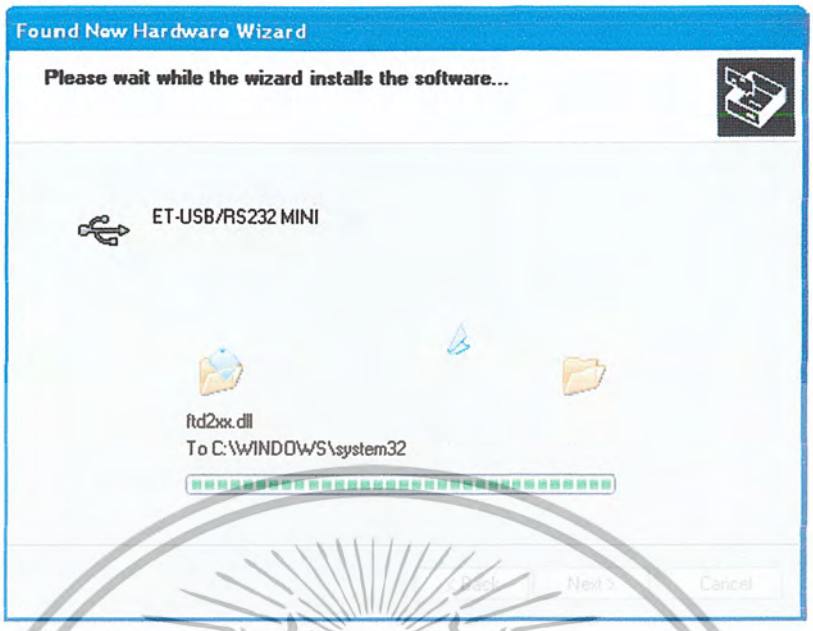
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. จากนั้นจะปรากฏหน้าต่าง Hardware Installation บอกว่าฮาร์ดแวร์ไม่ได้ผ่านการทดสอบของวินโดวส์ ให้ทำการคลิกที่ Continue Anyway ซึ่งวินโดวส์จะทำการติดตั้ง Driver ของ ET- USB/RS232 MINI



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

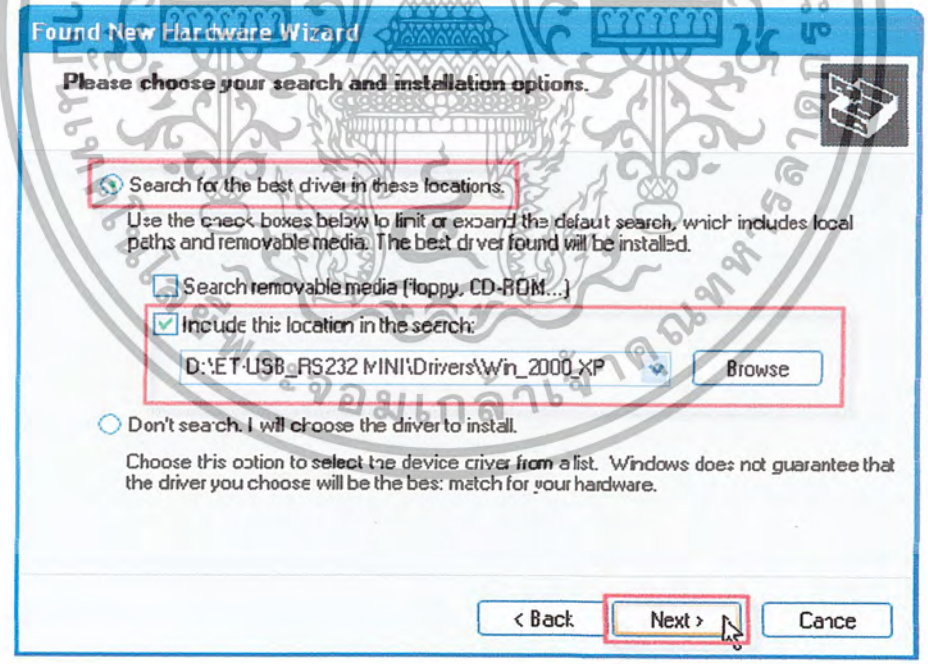


6. เมื่อปรากฏหน้าต่างว่าได้ทำการติดตั้งเรียบร้อยแล้วให้คลิก Finish จะสังเกตเห็นว่าไฟแสดงสถานะ

7. จากนั้นไม่นานตัว Driver จะมีการสร้างพอร์ตอนุกรมเสมือนขึ้นมาและมีหน้าต่างให้ติดตั้ง Driver ของ USB Serial Port ดังรูป ให้ทำการเลือกเหมือนข้อที่ผ่านมาจากนั้นคลิก Next

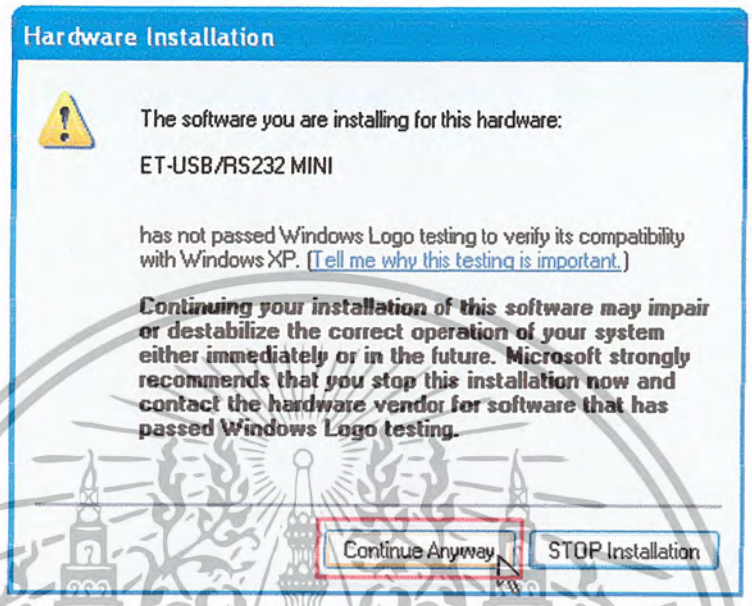
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เลือกตำแหน่งที่ตั้งของ Driver ซึ่งปกติจะจำค่าเดิมไว้อยู่แล้วให้คลิก Next ได้เลย จากนั้นวินโดวส์จะทำการหาฮาร์ดแวร์เพื่อที่จะทำการติดตั้ง Driver



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

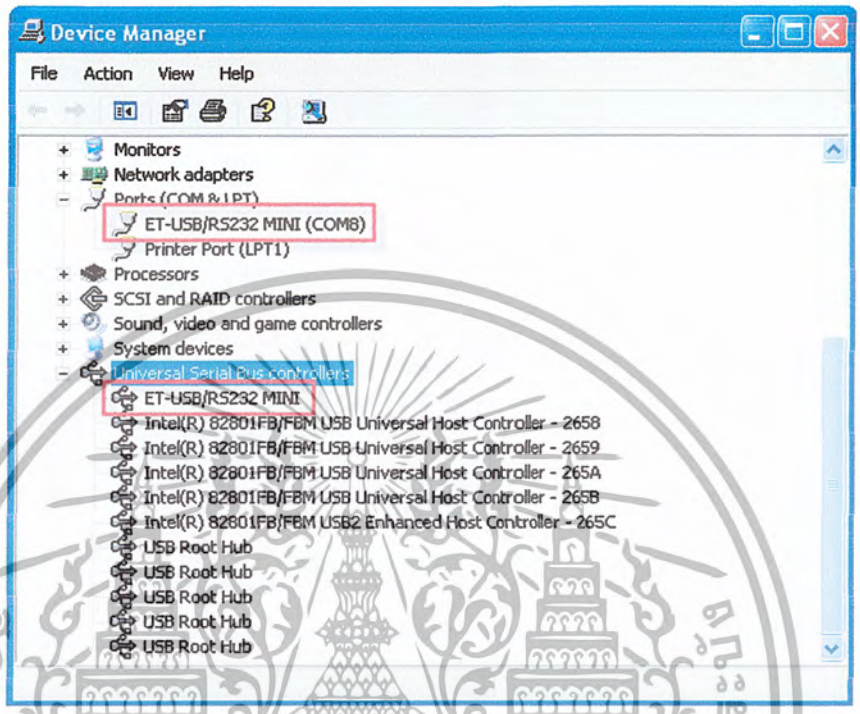
9. จากนั้นจะปรากฏหน้าต่าง Hardware Installation บอกว่าฮาร์ดแวร์ไม่ได้ผ่านการทดสอบของวินโดวส์ให้ทำการคลิกที่ Continue Anyway ซึ่งวินโดวส์จะทำการติดตั้ง Driver ของ ET-USB/RS232 MINI



10. เมื่อปรากฏหน้าต่างว่าได้ทำการติดตั้งเรียบร้อยแล้วให้คลิก Finish

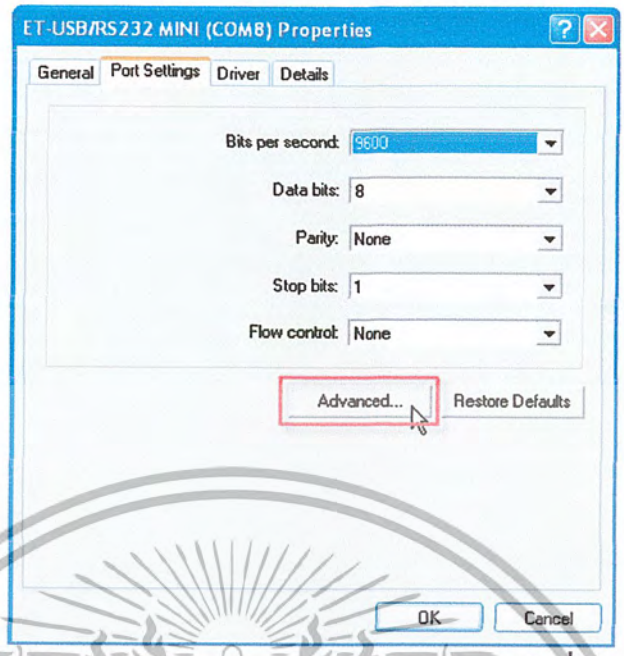
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. เราสามารถที่จะตรวจสอบว่าได้ติดตั้ง Driver ของ ET-USB/RS232 MINI เสร็จสมบูรณ์หรือไม่โดยดูที่ Control Panel > System เลือก Hardware และเลือกที่ Device Manager ซึ่งจะเห็นรายการฮาร์ดแวร์ ET-USB/RS232 MINI ดังรูป

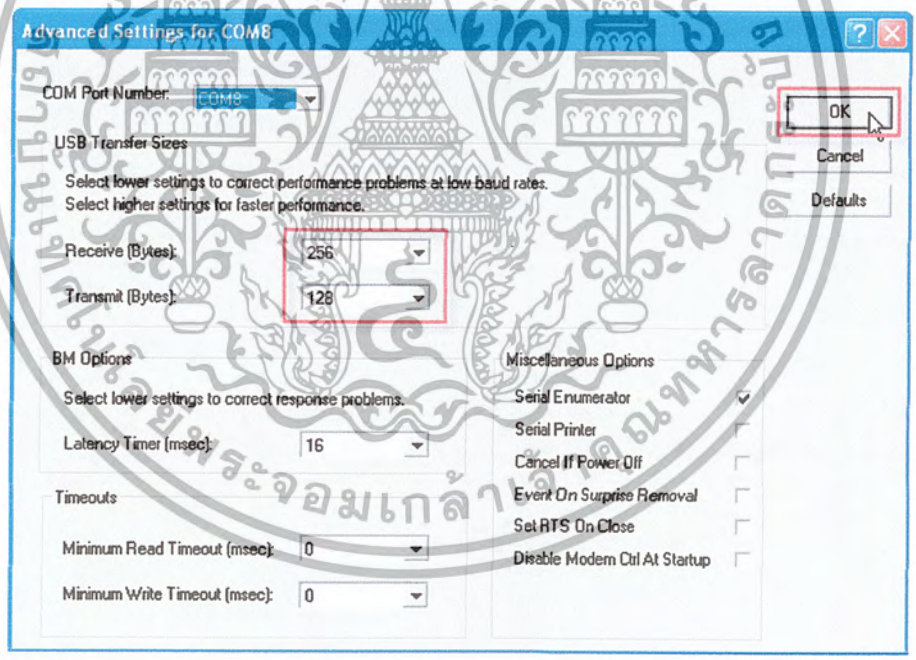


12. ดับเบิลคลิกที่ ET- USB/RS232 MINI ตรงส่วนของ Ports (COM&LPT) จากนั้นจะปรากฏหน้าต่าง Properties ดังรูป ให้เลือกมาที่ Port Setting และทำการคลิกที่ปุ่ม Advanced...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

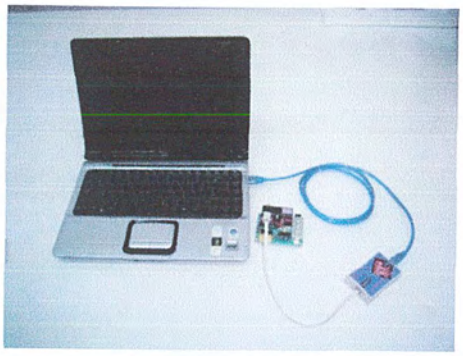


13. กำหนดค่า Receive (Bytes) และ Transmit (Bytes) ดังรูป และคลิกที่ OK เพื่อยืนยันการเปลี่ยนแปลง จากนั้นให้ทำการรีสตาร์ทเครื่องคอมพิวเตอร์หรือสแกนหาฮาร์ดแวร์ใหม่



การเชื่อมต่ออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

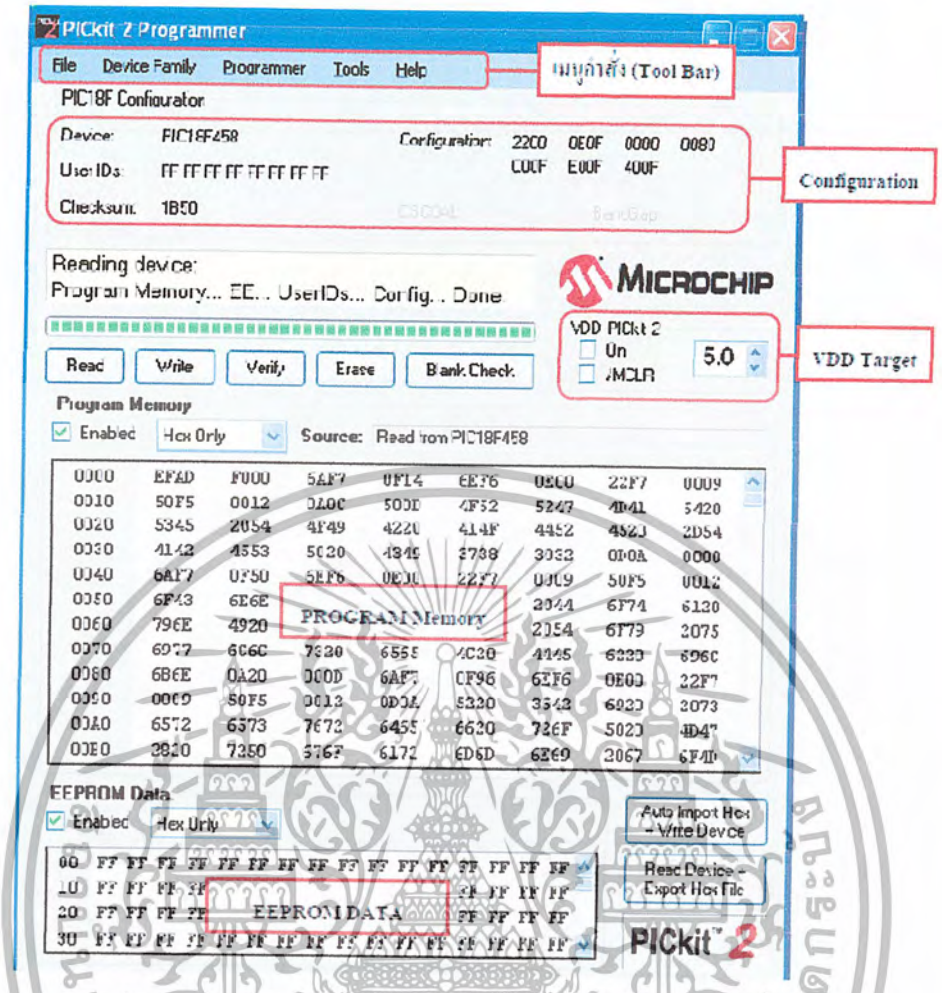


ซอร์ฟแวร์ที่ใช้กับเครื่องโปรแกรม ET-PGMPIC USB V2.0

ในส่วนของซอร์ฟแวร์ บอร์ด ET-PGMPIC USB V2.0 จะใช้ซอร์ฟแวร์ชื่อ PICkit 2 Programming ซึ่งเป็นของทางบริษัทไมโครชิฟ โดยก่อนใช้งานโปรแกรมจะต้องทำการติดตั้งโปรแกรมให้เรียบร้อยก่อน โดยจะต้องติดตั้งโปรแกรม .NET Framework (dotntfx) ก่อนตามด้วยโปรแกรม PICkit 2 Setup ดังต่อไปนี้

การใช้งานซอร์ฟแวร์โปรแกรม PICkit 2 Programmer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

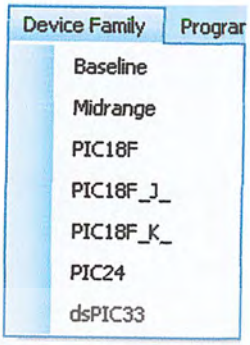


เมนูคำสั่งเกี่ยวกับการจัดการไฟล์ (File)

- **Import Hex** – โหลด hex file ที่ต้องการทำการโปรแกรมเข้ามาในโปรแกรม PICkit2
- **Export Hex** – Export hex file ที่อ่านได้จากตัวไมโครคอนโทรลเลอร์ เพื่อบันทึกเป็น hex file
- **Exit** – ออกจากโปรแกรม

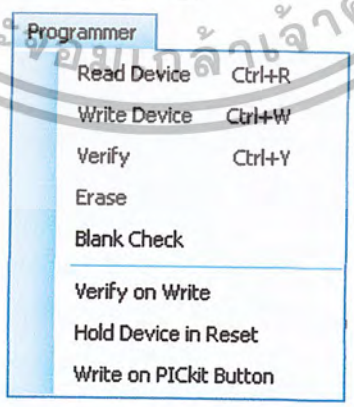
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูคำสั่งสำหรับเลือกตระกูลของไมโครคอนโทรลเลอร์ (DEVICE FAMILY)



- **Baseline (12-bit Core)** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์แบบ 12-bit Core Flash devices
- **Mid-range (14-bit Core)** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์แบบ 14-bit Core Flash devices
- **PIC18F** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์ตระกูล PIC18F Flash devices
- **PIC18F_J** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์ตระกูล PIC18FXXJXX Flash devices
- **PIC18F_K** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์ตระกูล PIC18FXXKXX Flash devices
- **PIC24** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์ตระกูล PIC24 Flash device
- **dsPIC33** เลือกใช้งานโปรแกรมกับไมโครคอนโทรลเลอร์ตระกูล dsPIC33 Flash devices

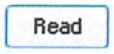
เมนูคำสั่งสำหรับฟังก์ชันการโปรแกรม (PROGRAMMER)



- **Read Device** อ่านข้อมูลจากหน่วยความจำโปรแกรม (Program memory), data EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

memory, IDlocations, และ Configuration bits.

มีคุณสมบัติเดียวกับปุ่ม 

- **Write Device** เขียนข้อมูลลงหน่วยความจำ Program memory, data EEPROM, ID locations, และ Configuration bits.

มีคุณสมบัติเดียวกับปุ่ม

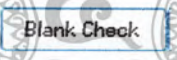
- **Verify** ตรวจสอบข้อมูลใน Program memory, Data EEPROM , ID locations และ Configuration bits ของตัวไมโครคอนโทรลเลอร์กับโค้ด (HEX File) ที่อยู่ในบัฟเฟอร์ของโปรแกรม PICkti2

มีคุณสมบัติเดียวกับปุ่ม

- **Erase** ลบข้อมูลในหน่วยความจำของไมโครคอนโทรลเลอร์

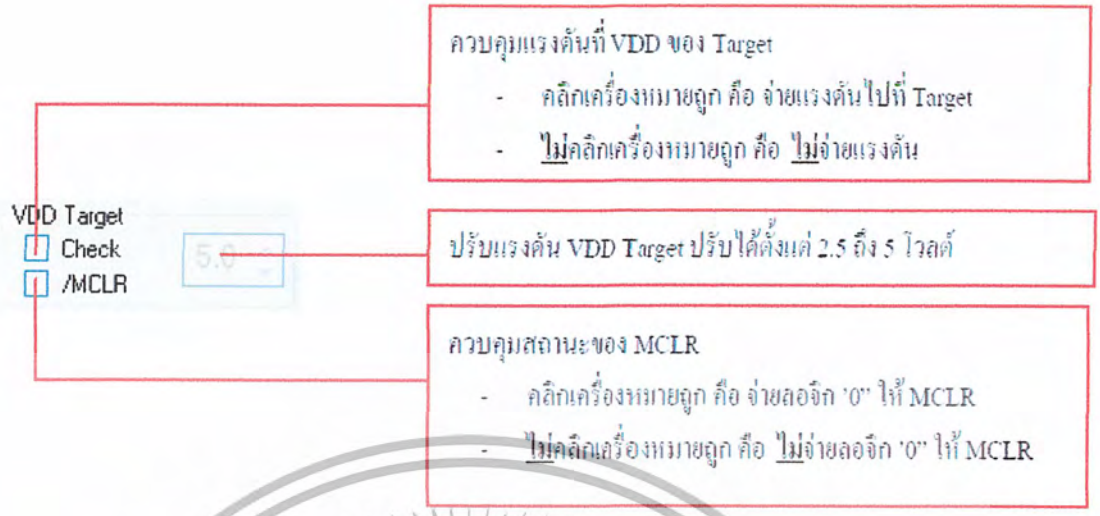
มีคุณสมบัติเดียวกับปุ่ม

- **Blank Check** ตรวจสอบพื้นที่หน่วยความจำ Program memory, data EEPROM , ID locations และConfiguration bits ว่าอยู่ในสถานะ ว่างเปล่า (Blank) หรือไม่

มีคุณสมบัติเดียวกับปุ่ม 

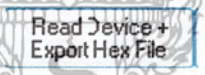
- **Verify on Write** ฟังก์ชันการตรวจสอบข้อมูลในหน่วยความจำ Program memory, data EEPROM, IDlocations, และ Configuration bits ในขณะทำการ Write ข้อมูล
- **Hold Device in Reset** ค้างสถานะที่ขาสัญญาณรีเซตไว้เป็นลอจิก "0" (MCLR =0)
- **Write on PICkit Button** ฟังก์ชันการโปรแกรมจากการกดสวิตช์ (PROGRAM) บนบอร์ด ET-PGM USB
- **VDD Target** เป็นฟังก์ชันในการจ่ายไฟเลี้ยงให้อุปกรณ์ Target และ ควบคุมสัญญาณรีเซต (MCLR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

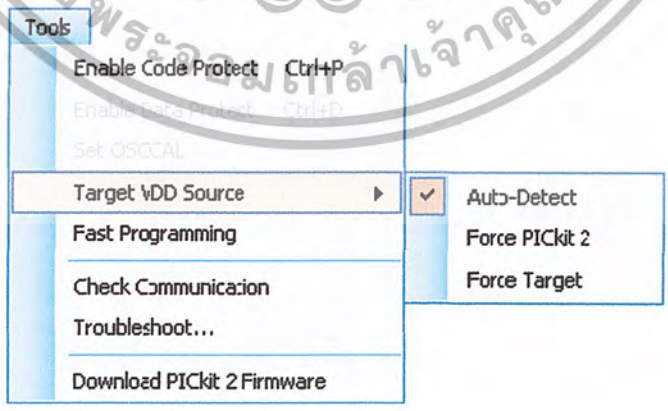


- **Auto Import Hex + Write Device** คือ ปุ่มคำสั่งที่ทำหน้าที่ทั้ง Import Hex File และ ทำการ Write ข้อมูล

- **Read Device + Export Hex File** คือ ปุ่มคำสั่งที่ทำหน้าที่ทั้งอ่านข้อมูลจากหน่วยความจำของไมโครคอนโทรลเลอร์และทำการ Export เป็น Hex File



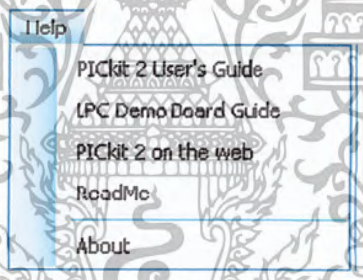
เมนูคำสั่งเครื่องมือสำหรับการโปรแกรม (Tools)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Enable Code Protect (Ctrl+P)** ฟังก์ชันปกป้องหน่วยความจำโค้ดโปรแกรม
- **Enable Data Protect (Ctrl+D)** ฟังก์ชันปกป้องหน่วยความจำข้อมูล EEPROM
- **Set OSCCAL** ใช้ค่าจากรีจิสเตอร์ OSCCAL เพื่อการปรับแต่งค่าความถี่ OSC ภายใน PIC
  - **Auto-Detect** ตรวจสอบแรงดันของอุปกรณ์ปลายทางโดยอัตโนมัติ
  - **Force PICkit 2** กำหนดให้แรงดัน VDD ที่จ่ายให้กับ Target มาจากบอร์ด PICkit2
  - **Force Target** กำหนดให้แรงดัน VDD ที่จ่ายให้กับ Target เป็นแรงดันของ Target เอง
- **Fast Programming** การโปรแกรมแบบรวดเร็ว
- **Check Communication** ตรวจสอบการเชื่อมต่อสื่อสารระหว่าง ET-PGMPIC กับ Computer
- **Troubleshoot...**เป็นฟังก์ชันของการให้ข้อมูลในการช่วยเหลือเมื่อเกิดปัญหาต่างๆ
- **Download PICkit 2 Firmware** คือ ฟังก์ชันที่ใช้สำหรับดาวโหลด Firmware ใหม่ๆ ของ PICkit2 ลงไปในบอร์ด ET-PGMPIC USB ใช้สำหรับการ Update Firmware

เมนูคำสั่งสำหรับการช่วยเหลือ (Help)



- **PICkit 2 User's Guide** คู่มือการใช้งาน PicKit2 เป็น PDF File
- **LPC Demo Board Guide** คู่มือบอร์ด Low Pin Count Demo Board ของ MICROCHIP
- **PICkit 2 on the web** ข้อมูลต่างๆของ PICkit2 บนเว็ปไซต์ของ MICROCHIP
- **ReadMe** ไฟล์ ReadMe ของโปรแกรม PICkit 2 แสดงรายละเอียด และ เบอร์ต่างๆ ของ PIC MCUที่ PICkit 2 สนับสนุนการใช้งาน
- **About** ข้อมูลรายละเอียดของตัวซอร์ฟแวร์ PICkit 2

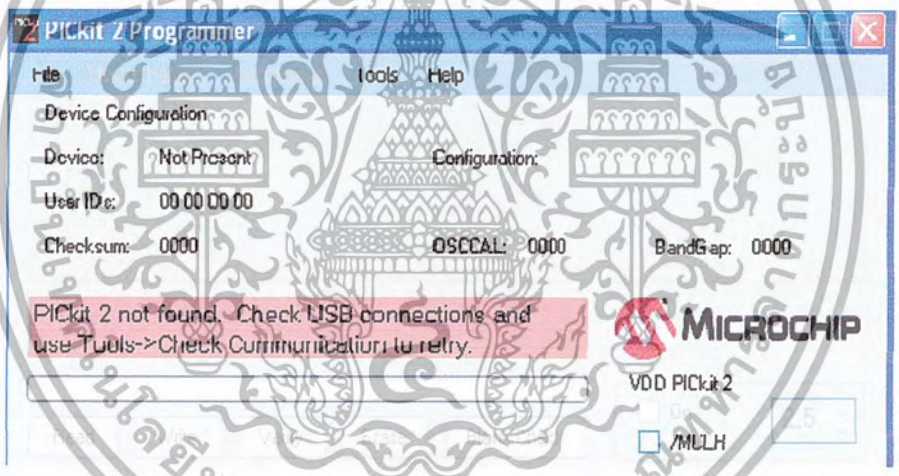
หน่วยความจำ EEPROM Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม PICkit 2 สามารถทำการแก้ไขข้อมูลในหน่วยความจำโปรแกรม EEPROM ของ PIC Micro ได้โดยจะมีหน้าต่างในการแก้ไขข้อมูล วิธีการแก้ไขก็เพียงแค่นำเมาส์ ไปคลิกแก้ไขในตำแหน่งข้อมูลที่เราต้องการ ซึ่งเมื่อการ Write ข้อมูลลงไป หน่วยความจำ EEPROM ของ PIC Micro ก็จะเปลี่ยนแปลงตามข้อมูลดังที่เรากำหนดดังรูปต่อไปนี้

ข้อผิดพลาดและแนวทางการแก้ไขปัญหา

1. การผิดพลาดจากการเชื่อมต่อระหว่างคอมพิวเตอร์กับ บอร์ด PICkit2 จะฟ้องข้อความดังรูปต่อไปนี้



แนวทางการแก้ไข

- ตรวจสอบการเชื่อมต่อของสาย USB ระหว่างคอมพิวเตอร์ กับ บอร์ด ET-PGMPIC USB
- คลิก Tools -> Check Communication เพื่อทำการตรวจสอบอีกครั้ง

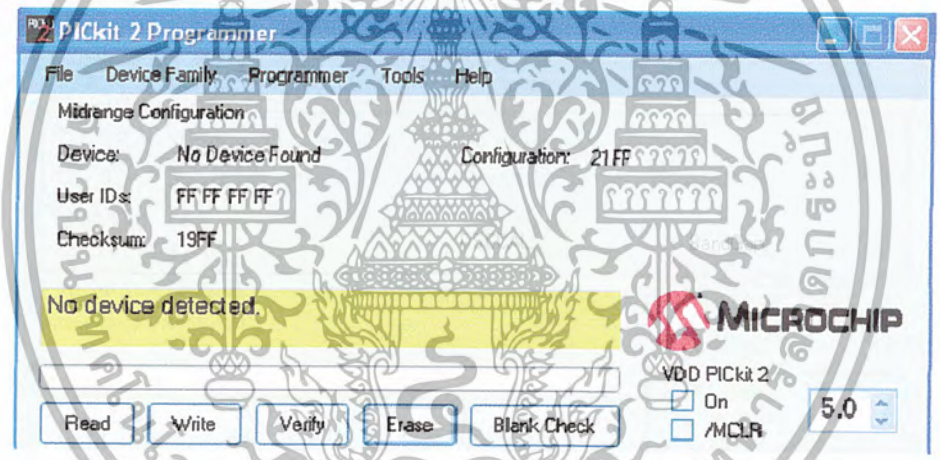
2. ความผิดพลาดจากการตรวจสอบแรงดันที่ Target Board โดยจะมี Error Massage ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการแก้ไข

- กรณีการใช้งานเครื่องโปแกรมโดยผ่าน Text Tool ให้ตรวจสอบจัมพ์เปอร์ T/B ว่าอยู่ในตำแหน่ง T หรือไม่
- กรณีการใช้งานผ่านโมดูล Emulator ให้ตรวจสอบจัมพ์เปอร์ T/B ว่าอยู่ในตำแหน่ง B หรือไม่ และตรวจสอบไฟเลี้ยงของ Target Board ว่ามีไฟเลี้ยงหรือไม่ ถ้าไม่มีให้ทำการจ่ายไฟเลี้ยงที่บอร์ดปลายทาง (Target Board) ให้เรียบร้อย

3. ปัญหาจากการตรวจไม่พบไมโครคอนโทรลเลอร์

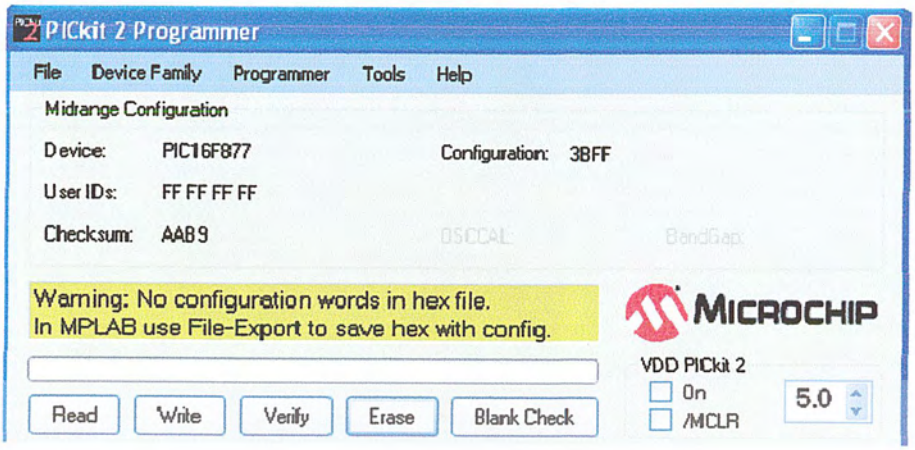


แนวทางการแก้ไข

- ตรวจสอบการใส่ไอซี ใน Text Tool ว่าใส่ถูกต้องหรือไม่ ขา 1 ของไอซีใส่ในตำแหน่งที่ถูกต้องหรือไม่
- กรณีการโปแกรมด้วยโมดูล Emulator ให้เช็คสายสัญญาณที่เชื่อมต่อว่าอยู่ในสภาพดีหรือไม่ และเช็คแรงดันที่ Target Board ว่ามีการจ่ายแรงดันหรือไม่

4. ปัญหาจาก Hex File ที่ Import เข้าไม่มีค่า Configuration รวมอยู่ด้วย ซึ่งปัญหานี้เกิดขึ้นในขั้นตอนของการออกแบบ และ คอมไพล์โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แนวทางการแก้ไข

- ทำการกำหนดค่า Configuration ให้เรียบร้อยในขั้นตอนของการออกแบบและสร้างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้