อภินันทนาการ

สำนักหอสมุด

กล้องเคลื่อนที่หมุนตามการตรวจจับใบหน้า

CAMERA TRACKING BASED ON FACE DETECTION

นายวิศัลย์ สีสมุด รหัส 54361138
นายณัฐพงศ์ พันธุ์จง รหัส 54363736
นายธีระยศ กองเพชร รหัส 54363859

i7194870

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 10 มิ.ย. 2558
เลขทะเบียน
เลขเรียกหนังสือ

ป
วทรก
ส557

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2557

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	กล้องเคลื่อนที่หมุนตามการตรวจจับใบหน้า			
ผู้ดำเนินโครงงาน	นายวิศัลย์	สีสมุด	รหัส	54361138
	นายณัฐพงศ์	พันธุ์จง	รหัส	54363736
	นายธีระยศ	กองเพชร	รหัส	54363859
ที่ปรึกษาโครงงาน	รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง			
สาขาวิชา	วิศวกรรมไฟฟ้า			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2557			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(รองศาสตราจารย์ ดร. ไพศาล มุณีสว่าง)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พรพิศุทธิ์ วรจิรันตน์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนัส นัถฤทธิ์)

ชื่อหัวข้อโครงงาน	กล้องเคลื่อนที่หมุนตามการตรวจจับใบหน้า			
ผู้ดำเนินโครงงาน	นายวิศัลย์	สีสมุด	รหัส	54361138
	นายณัฐพงศ์	พันธุ์จง	รหัส	54363736
	นายธีระยศ	กองเพชร	รหัส	54363859
ที่ปรึกษาโครงงาน	รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง			
สาขาวิชา	วิศวกรรมไฟฟ้า			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2557			

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมควบคุมกล้องเว็บแคมโดยระบบคอมพิวเตอร์ ที่ใช้สำหรับการตรวจจับใบหน้าของบุคคลที่กำลังเคลื่อนไหวทั้งในแนวราบและแนวดิ่งได้โดยอัตโนมัติ โดยใช้ทฤษฎีการประมวลผลภาพของ Open CV คือ ทฤษฎีการรวมภาพเพื่อตรวจจับลักษณะเด่นอย่างรวดเร็ว ในการตรวจจับใบหน้า และตีกรอบ เพื่อแสดงตำแหน่งใบหน้า ตามวิธีของพอล วิโอลา และไมเคิล โจนส์ โปรแกรมที่ใช้ในการพัฒนา คือ โปรแกรมภาษา Processing อุปกรณ์ที่ใช้ประกอบด้วย กล้องเว็บแคม บอร์ด Aduino (Atmega 328) เซอร์โวมอเตอร์ และคอมพิวเตอร์

ผลของการศึกษาและพัฒนา พบว่า ได้กล้องเคลื่อนที่หมุนตามการตรวจจับใบหน้าที่สามารถตรวจจับใบหน้า เคลื่อนที่ตามใบหน้า และสามารถแยกใบหน้าคนออกจากส่วนประกอบที่อยู่ในวีดีโอได้คือ สามารถตรวจจับใบหน้าที่หันเข้ามาหากล้อง หรือใบหน้าที่มีการเอียงซ้ายเอียงขวาหรือเงยหน้าขึ้นลงที่ไม่มากนัก ได้ในระยะ 3 เมตร สามารถตรวจจับใบหน้าที่เคลื่อนไหวในแนวราบได้ด้วยความเร็วสูงสุด 0.38 เมตรต่อวินาที สามารถตรวจจับใบหน้าที่เคลื่อนไหวในแนวดิ่งได้ด้วยความเร็วสูงสุด 0.80 เมตรต่อวินาที

Project title	Camera Tracking based on Face Detection		
Name	Mr. Wisun	Seesamoot	ID. 54361138
	Mr. Natthaphong	Phanchong	ID. 54363736
	Mr. Teerayot	Kongphet	ID. 54363859
Project advisor	Assc. Prof. Dr. Paisarn	Muneesawang	
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2014		

Abstract

The proposes of this project were to find out and develop the computer program for camera tracking based on face detection which can detect people's face automatically in vertical and horizontal. The program was designed by using the processing theory of Open CV which created by Paul Viola and Michael Jones. The materials were a web camera, an Arduino board, a servo motor and a computer.

The results revealed that the camera tracking based on face detection can detect people's face automatically as following; The camera tracking based on face detection can detect people's face which be front of it in 3 meters, The camera tracking based on face detection can detect people's face which move in horizontal limit in 0.38 meters/a second and The camera tracking based on face detection can detect people's face which move in vertical limit in 0.80 meters/a second.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้เนื่องจาก รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง ได้เสียสละเวลาให้คำแนะนาปรึกษา และแนวทางในการแก้ไขปัญหา ตลอดจนตรวจสอบแก้ไขปริญญานิพนธ์ฉบับนี้ให้มีความถูกต้องสมบูรณ์ ผู้จัดทาขอกราบขอบพระคุณ ณ โอกาสนี้ด้วย

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พรพิศุทธิ์ วรจิรันตน์ และ ผู้ช่วยศาสตราจารย์ ดร.พนัส นัถฤทธิ์ ที่ให้ความอนุเคราะห์เสียสละเวลามาเป็นกรรมการตรวจสอบโครงงาน

สุดท้ายขอขอบคุณเพื่อนๆ ทุกคนที่คอยให้ความร่วมมือด้วยดีมาโดยตลอด ให้คำแนะนำและคำปรึกษาจนทาให้ปริญญานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดี

นายวิศัลย์ สีสมุด
นายณัฐพงศ์ พันธุ์จง
นายธีระยศ กองเพชร

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อ ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่ 1 บทนำ
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 แผนการดำเนินงานตลอดโครงงาน 2
1.5 ผลที่คาดว่าจะได้รับ 3
1.6 งบประมาณของโครงงาน 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง
2.1 การเตรียมข้อมูลภาพ (Preprocessing) 4
2.2 ภาพเคลื่อนไหว 5
2.3 รูปภาพขาว - ดำ (Binary Image) 6
2.4 รูปดิจิตอล 6
2.5 กล้องวีดิโอ 7
2.6 Processing 8
2.7 ภาษาจาวา (Java) 9
2.8 OpenCV (Open Source Computer Vision Library) 9
2.9 ระบบตรวจจับใบหน้า ของ OpenCV 10
2.10 การสื่อสารอนุกรม 15
2.11 เซอร์โวมอเตอร์(Servo Motor) 18

สารบัญ (ต่อ)

หน้า

บทที่ 3 ขั้นตอนการดำเนินงาน

3.1 การออกแบบโปรแกรม 20

3.2 การสร้างโปรแกรม 29

3.3 อุปกรณ์ 30

บทที่ 4 ผลการทดลอง

4.1 ขั้นตอนการทดสอบการเชื่อมต่อและทำการของระบบ 33

4.2 ทดสอบขอบเขตของการใช้งาน 37

บทที่ 5 สรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลองโครงงาน 41

5.2 ข้อเสนอแนะ 41

เอกสารอ้างอิง 42

ภาคผนวก 43

ประวัติผู้ดำเนินโครงงาน 52

สารบัญตาราง

ตารางที่ หน้า

1.1 ตารางการดำเนินการ 2
4.1 ผลการทดลองตรวจจับใบหน้าในรูปแบบต่างๆ 35
4.2 ตารางผลการทดลองตรวจจับและหมุนตามใบหน้าในระยะห่างต่างๆ 38
4.3 ตารางทดสอบความเร็วที่กล้องสามารถทำงานได้ในแนวราบ 39
4.4 ทดสอบความเร็วที่กล้องสามารถทำงานได้ในแนวดิ่ง 40

สารบัญรูป

รูปที่ หน้า

2.1 แสดงแต่ละพิกเซล (Pixel) ที่ถูกกำหนด 6
2.2 ระบบพิกัด Space 7
2.3 ตัวอย่างการจำลองรูปแบบ Haar-like 11
2.4 ตัวอย่างภาพการรวมภาพ (Integral image) 12
2.5 ตัวอย่างการคำนวณผลรวมความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธี Integral image 12
2.6 การทำงานของ Adaboost 13
2.7 แสดงตัวอย่างของตัวจำแนกแบบแข็งแรงที่ได้จากกระบวนการเรียนรู้ด้วย Adaboost 14
2.8 แสดงขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง 15
2.9 การถ่ายโอนข้อมูลแบบอนุกรม 15
2.10 USB1.0 17
2.11 USB1.1 17
2.12 หัวต่อและเต้ารับ 17
2.13 รูปแบบการจัดเรียง 17
2. 14 ชิ้นส่วนภายในของเซอร์โวมอเตอร์ (Servo Motor) 18
2.15 การควบคุมเซอร์โวมอเตอร์ (Servo Motor) 19
3.1 ระบบโดยรวม 20
3.2 ระบบตรวจจับใบหน้า (Detection System) 21
3.3 ขอบเขตควบคุมตำแหน่งใบหน้า 22
3.4 การทำงานของระบบควบคุมมอเตอร์ 25
3.5 แสดงการตรวจพบใบหน้าทางด้านซ้ายมือของกรอบวีดีโอ 26
3.6 แสดงส่วนของเงื่อนไขคำสั่งในโปรแกรม Processing 27
3.7 แสดงเส้นแบ่งขอบเขตของกรอบวีดีโอ 27
3.8 แสดงคำสั่งควบคุมเซอร์โวมอเตอร์ (Servo Motor) 29
3.9 ขั้นตอนการสร้างโปรแกรม 29

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.10 บอร์ด Aduino (Atmega 328) 30
3.11 เซอร์โวมอเตอร์ (Servo Motor) 30
3.12 กล้องเว็บแคม 30
3.13 คอมพิวเตอร์ 30
3.14 ชิ้นงานของโครงงาน 31
3.15 วงจรในโครงงาน 31
4.1 แสดงผลการติดต่อกับกล้อง 33
4.2 แสดงการหาตำแหน่งของใบหน้า 34
4.3 แสดงตำแหน่งของใบหน้า 37
4.4 แสงตำแหน่งของใบหน้า 37

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

กล้องคืออุปกรณ์ที่ใช้ส่งสัญญาณภาพไปยังอุปกรณ์ปลายทาง เช่น กล้องถ่ายรูป กล้องวีดีโอ กล้องวงจรปิด และมอนิเตอร์ ในที่นี้เราจะสนใจการทำงานของกล้องวีดีโอที่ใช้สำหรับการจับภาพและบันทึกภาพในลักษณะแบบทันทีทันใด (Real time) คล้ายกับกล้องวงจรปิด แต่จะใช้กล้องวีดีโอธรรมดาเช่น กล้องเว็บแคมเพื่อนำมาใช้ในการจับภาพใบหน้าและหมุนตามการเคลื่อนไหวได้อัตโนมัติ

ซึ่งกล้องวีดีโอมีขีดจำกัดความสามารถในการใช้งานอยู่ คือ จะจับภาพได้เฉพาะจุดและการที่จะบันทึกภาพวัตถุที่เคลื่อนไหวจำเป็นจะต้องมีคนคอยหมุนกล้อง ให้กล้องหันไปในทิศทางที่วัตถุนั้นเคลื่อนที่ รวมถึงการจะจับภาพที่มีลักษณะเฉพาะเช่น ใบหน้าของคนที่เคลื่อนที่นั้นถ้าจะให้คนมาคอยหันกล้องวีดีโอคอยจับใบหน้าตลอดเวลา ซึ่งเป็นเรื่องที่เกินจำเป็นอย่างมากที่จะต้องมีคนเฝ้ากล้องวีดีโอนั้น

ดังนั้นโครงการนี้จึงทำการศึกษาเกี่ยวกับกล้องเคลื่อนที่หมุนตามการเคลื่อนไหวของใบหน้าที่เคลื่อนที่ได้ โดยใช้หลักการประมวลผลภาพ (Image Processing) มาเขียนโปรแกรมประมวลผลภาพเพื่อตรวจสอบความเคลื่อนไหวแล้วนำภาพส่งไปประมวลผลเพื่อบันทึกและแสดงผลออกทาง

หน้าจอคอมพิวเตอร์ โดยใช้กล้องหมุนตามเป้าหมายทั้งแนวราบและแนวดิ่งโดยผ่านไมโครคอนโทลเลอร์ (Microcontroller) ไปควบคุมเซอร์โวมอเตอร์ (Servo motor)ให้กล้องเคลื่อนที่และประโยชน์ของกล้องตรวจจับใบหน้านี้ เช่น การล็อคใบหน้าอาจารย์ที่กำลังบรรยายในห้องเรียนขนาดใหญ่ เพื่อแสดงในจอแสดงผลที่อยู่อีกที่หนึ่งได้

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาลักษณะและการทำงานของการตรวจจับความเคลื่อนไหว
2. ศึกษาและออกแบบวงจรที่ใช้ควบคุมกล้องผ่านคอมพิวเตอร์
3. ศึกษาการทำงานของโปรแกรม Processing และ OpenCV
4. ศึกษาลักษณะการส่งข้อมูลแบบอนุกรม (Serial Port)
5. ศึกษาการทำงานของไมโครคอนโทลเลอร์ (Microcontroller)
6. ศึกษาการทำงานของเซอร์โวมอเตอร์ (Servo motor)
7. ศึกษาการเปรียบเทียบภาพของการประมวลผลภาพ (Image Processing)

1.3 ขอบเขตของโครงงาน

1. เขียนโปรแกรม Processing, OpenCV เพื่อวิเคราะห์การเคลื่อนไหวของใบหน้า
2. เขียนโปรแกรมเพื่อควบคุมการหมุนของกล้องในแนวราบ (180°) และแนวดิ่ง (180°)
3. สร้างกล้องที่หมุนตามการเคลื่อนไหวของใบหน้าคนได้ 1 คน
4. ความเร็วสูงสุดในการหันตาม 0.38 เมตรต่อวินาที
5. ระยะห่างจากกล้องถึงบุคคล 2 เมตร

1.4 แผนการดำเนินงานตลอดโครงงาน

ลำดับ		ปี 2557					ปี 2558		
		ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	ม.ค.
1.	ศึกษาการทำงานของโปรแกรม Processing			←	—	—	→		
2.	ศึกษาการทำงานของโปรแกรม OpenCV			←	—	—	→		
3.	ศึกษาและออกแบบวงจรควบคุมกล้องผ่านคอมพิวเตอร์				←	—	→		
4.	ศึกษาและเขียนโปรแกรมควบคุมการหมุนของกล้อง				←	—	—	—	→
5.	ศึกษาการทำงานของไมโครคอนโทลเลอร์				←	—	—	→	
6.	ศึกษาการทำงานของการส่งข้อมูลแบบอนุกรม				←	—	→		
7.	ศึกษาการเปรียบเทียบภาพของการประมวลผลภาพ					←	—	→	
8.	สร้างกล้องที่หมุนตามความเคลื่อนไหวของวัตถุและใบหน้า					←	—	—	→

ตารางที่ 1.1 ตารางการดำเนินงาน

1.5 ผลที่คาดว่าจะได้รับ

1.ได้เรียนรู้หลักการทำงานของโปรแกรมควบคุมการหมุนของกล้อง

2.เขียนโปรแกรม Processing , OpenCV เพื่ออ่านค่าจากกล้อง ไปแสดงผลในคอมพิวเตอร์

3.ได้เรียนรู้หลักการทำงานของการเชื่อมต่อแบบอนุกรม (Serial Port)

4.ได้เรียนรู้กระบวนการเปรียบเทียบภาพของการประมวลผลภาพ (Image Processing)

5.ได้เรียนรู้หลักการทำงานของไมโครคอนโทลเลอร์ (Microcontroller)

6.ได้เรียนรู้หลักการทำงานของเซอร์โวมอเตอร์ (Servo motor)

7.สามารถที่จะนำกล้องไปใช้เพื่อเกิดประโยชน์ต่อสังคมได้

1.6 งบประมาณของโครงงาน

1.ค่าวัสดุอิเล็กทรอนิกส์ 1,300 บาท

2.ค่าวัสดุคอมพิวเตอร์ 250 บาท

รวม 1,550 บาท

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 การเตรียมข้อมูลภาพ (Preprocessing)

การประมวลผลภาพเรียกอีกอย่างหนึ่งว่าการประมวลผลภาพ (Image Processing) [5] หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใดๆมากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัด หรือการประหยัดพื้นที่ในการเก็บข้อมูลหรือใช้สำหรับการประมวลผลระดับสูง เช่น การจดจำรูปร่างลักษณะให้ได้อย่างแม่นยำ โดยทั่วไปแล้ววัตถุประสงค์ของการประมวลผลภาพ (Image Processing) ก็คือ

การประมวลผลภาพ (Image Processing): Image in ----Image out : วิธีนี้จะใช้กระบวนการทางการประมวลผลภาพดิจิตทัลเพื่อทำให้ได้ภาพออกมา เช่น การตกแต่งภาพด้วยโปรแกรม Photoshop

การวิเคราะห์ภาพ (Image Analysis): Image in----Measurement out : วิธีนี้จะใช้กระบวนการทางการประมวลผลภาพดิจิตทัลเพื่อทำให้ได้ค่าการวัดออกมา เช่น การวัดขนาดในงาอุตสาหกรรม

การเข้าใจและแปลความหมายภาพ (Image Understanding): Image in---- High-level Description out : วิธีนี้จะใช้กระบวนการทางการประมวลผลภาพดิจิตทัลเพื่อให้ได้ผลลัพธ์ออกมาเป็นความหมาย ตัวอย่างเช่น การจดจำตัวอักษร

การประมวลผลด้วยคอมพิวเตอร์ สามารถทำได้โดยนำภาพที่ได้มาจากกล้องหรือ Image Source ต่างๆ ซึ่งเป็นสัญญาณ Analog แล้วนำมาแปลงเป็นสัญญาณดิจิตอลที่มีลักษณะ เป็นรหัสเชิงตัวเลข 0,1 ที่สามารถใช้รูปแบบทางคณิตศาสตร์เข้ามาช่วยในการคำนวณและการประมวลผลข้อมูลภาพด้วยคอมพิวเตอร์ต่อไป การประมวลผลภาพแบ่งออกได้เป็น 2 ระดับ คือ

2.1.1 การประมวลผลภาพระดับต่ำ

เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากได้ภาพมา ภาพที่ได้ก็จะประกอบไปด้วยองค์ประกอบต่างๆ มากมาย รวมถึงสิ่งที่เราไม่ต้องการด้วย ในที่นี้เราจะเรียกว่า Noise ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี ยังไม่สามารถที่นำไปใช้ในการประมวลผลได้ ดังนั้น การประมวลผลภาพระดับต่ำจึงจะประกอบไปด้วยการกำจัดสัญญาณรบกวน การทำภาพให้ชัด การหาขอบภาพ การแปลงภาพขาว-ดำ (Binary Image) การแบ่งแยกรูปร่างวัตถุ เพื่อหาค่าตัวแปรต่างๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลภาพระดับสูงต่อไป

2.1.2 การประมวลผลภาพระดับสูง

เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผล โดยการประมวลผลระดับต่ำจะใช้ค่าความสว่างหรือความเข้มของแสงโดยตรง ส่วนกลางประมวลผลระดับสูง ข้อมูลที่นำมาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ และการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำ มาอธิบายถึงสัญลักษณ์เหล่านี้ การประมวลผลภาพระดับสูงนั้นส่วนใหญ่มักจะใช้ทฤษฎีต่างๆ เข้ามาใช้เป็นตัวช่วยในการทำงาน หรือเป็นหัวใจของโปรแกรม

2.1.3 การแทนภาพด้วยข้อมูลแบบดิจิตอล

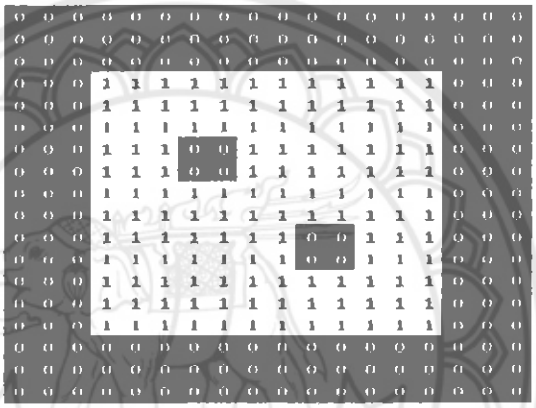
ข้อมูลภาพแบบดิจิตอลเป็นภาพที่ถูกดัดแปลงมาจากภาพแบบต่อเนื่องหรือ Analog image ให้อยู่ในรูปตัวเลข หรือ Digital Image [5] ด้วยวิธีการ Digitization โดยภาพ Analog จะถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (Pixels) โดยในแต่ละพิกเซลจะใช้ (x,y) ในการระบุตำแหน่ง การแสดงข้อมูลภาพดิจิตอลสามารถอธิบายได้ด้วยเมตริกซ์ (Matrix) M*N และให้จุดต่างๆ ที่อยู่ในเมตริกซ์เป็นจุดพิกัด (x,y) ใดๆ เป็นส่วนประกอบของภาพ ค่าของพิกเซล หรือฟังก์ชัน (x,y) ณ จุดใดๆ จะแสดงได้ด้วยค่าของความเข้มของแสง ซึ่งอาจแบ่งได้หลายระดับ ถ้ามี 2 ระดับก็จะเป็นแค่ 0 กับ 1 จุดต่างๆ ที่แสดงอยู่ในพิกัดก็ คือ พิกเซล หรือ Picture Element ซึ่งก็คือ ความสว่างหรือค่า Luminance ของภาพ ถ้าภาพนั้นเป็นภาพขาวดำ ขนาด8 บิตจะมีค่า L เท่ากับ หรือเท่ากับ 256 คือ ตั้งแต่ระดับ 0 ถึง 255 บางครั้งค่าความสว่างอาจหมายถึงระดับความละเอียดของภาพ

2.2 ภาพเคลื่อนไหว

ความจริงแล้วภาพเคลื่อนไหว [5] ก็คือภาพนิ่งที่นำมาแสดงต่อๆ กันแบบต่อเนื่อง จะต้องใช้รูปภาพอย่างน้อย 24 รูปต่อหนึ่งวินาที เนื่องจากสายตาของคนเราเมื่อนำภาพนิ่งมาฉายติดต่อกันมากกว่า 24 รูปต่อหนึ่งวินาทีแล้วก็จะมองภาพนั้นว่าเป็นภาพเคลื่อนไหว เพราะสายตาเราแยกไม่ออกเนื่องจากมีความเร็วมากเกินไปแต่หากนำภาพนิ่งมาฉายติดต่อกันน้อยกว่า 24 รูปต่อหนึ่งวินาทีแล้ว เราจะมองเห็นเป็นภาพไม่ต่อเนื่อง

2.3 รูปภาพขาว - ดำ (Binary Image)

รูปภาพขาว-ดำ คือ รูปภาพที่ประกอบด้วยสีเฉพาะสีขาวและสีดำ ในรูปดิจิตอล (Digital images) ภาพจะอยู่ในรูปของแผ่นตาราง โดยแต่ละช่องจะเป็นส่วนหนึ่งของภาพหรืออักษร เรียกแต่ละจุดหรือช่องนั้นว่า pixel แต่ละ pixel จะถูกกำหนดให้มีระดับของความเข้ม (สีดำ สีขาว สีเทาหรือสีอื่น ๆ) ซึ่งแสดงให้อยู่ในรูปของรหัสเลขฐานสอง (0 และ 1) แต่ละพิกเซล ก็จะแทนด้วย Binary digital (bits) จะถูกเก็บเป็นลาดับในคอมพิวเตอร์ โดยส่วนใหญ่แล้วคอมพิวเตอร์สามารถวิเคราะห์ข้อมูลจากรูปภาพขาวดำได้ง่ายกว่ารูปภาพแบบอื่น ๆ เนื่องจากรูปภาพขาวดำมีค่าข้อมูลของสีเพียง 2 ค่า ซึ่งแทนด้วยค่า 0, 1 หรือ 0, 255 โดยที่ 0 แทนสีดำ และ1 หรือ 255 แทนสีขาว



รูปที่ 2.1 แสดงแต่ละ Pixel ที่ถูกกำหนด

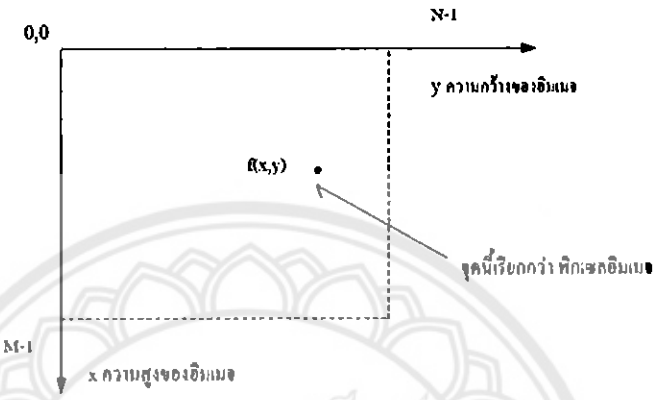
จากรูป 2.1 เป็นกระบวนการแปลงภาพสีให้มีการแสดงผลได้แค่ 2 ระดับ คือ ขาว และดำ โดยจะแปลงข้อมูลภาพให้เป็นภาพขาว-ดำ (Binary Image) มีกระบวนการแปลงภาพที่มีความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต (bit) คือ 0 และ โดย 0 แทนด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสีดำ

2.4 รูปดิจิตอล

รูปดิจิตอล (Digital Image) ก็คือภาพที่เก็บอยู่ในรูปแบบดิจิตอล โดยปกติแล้วสายตาคนเราจะมองเห็นภาพทิวทัศน์ต่างๆเป็นลักษณะแบบอนาล็อก ซึ่งสามารถอธิบายได้ด้วยคณิตศาสตร์ที่มีตัวแปรแบบนับได้อย่างต่อเนื่อง แต่เครื่องคอมพิวเตอร์จะใช้เลขฐาน 2 เป็นหลักในการคำนวณ เมื่อนำภาพอิมเมจมาแปลงเข้าสู่เครื่องคอมพิวเตอร์ อิมเมจนั้นก็จะกลายเป็น อิมเมจดิจิตอล

2.4.1 ระบบพิกัด Space

อิมเมจดิจิตอลเป็นผลมาจากการสุ่มค่าในระบบพิกัด Space หรือ (Spatial Coordinate) ระบบพิกัด Space นี้ จะใช้กับการแสดงอิมเมจดิจิตอล ซึ่งจะมีความกว้างและความสูงของอิมเมจแสดงในแกน Y และ X ตามลาดับ ส่วนจุดใดๆที่วางบนระนาบ X,Y จะเป็นฟังก์ชัน f(x,y) และเรียกว่า พิกเซล (Pixel)



รูปที่ 2.2 ระบบพิกัด Space

จากรูปที่ 2.2 จุดที่วางอยู่ในพิกัด Space นี้คือ พิกเซล (Pixel) ซึ่งก็คือความสว่างหรือ Luminance (L) ของอิมเมจถ้าอิมเมจนั้นเป็นภาพขาวดำขนาด 8 บิต จะมีค่า L เท่ากับ 2^8 หรือ เท่ากับ 256 ระดับคือตั้งแต่ ระดับ 0 (พิกเซลเป็นสีดำ) จนถึง 255 (พิกเซลเป็นสีขาว)

2.5 กล้องวีดิโอ

กล้องวีดีโอ เป็นกล้องที่ทำการบันทึกข้อมูลภาพและเสียงด้วยการแปลสัญญาณคลื่นให้เป็นตัวเลข 0 กับ 1 คุณภาพของวีดีโอที่ได้จะมีความใกล้เคียงกับต้นฉบับมาก ทำให้สามารถบันทึกข้อมูลลงบนฮาร์ดดิสก์ ซีดีรอม ดีวีดี หรืออุปกรณ์บันทึกข้อมูลอื่นๆ และสามารถแสดงผลบนคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ ในการผลิตมัลติมีเดียบนคอมพิวเตอร์สามารถเปลี่ยนรูปแบบของสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลได้ หากผู้ใช้มีทรัพยากรทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่เหมาะสม

2.5.1 คุณสมบัติของวิดีโอ

วิดีโอมีคุณสมบัติที่สำคัญ 3 อย่างได้แก่ ภาพ (Image) ,เสียง (Audio) และวีดีโอ (Video)

ภาพ (Image) ประกอบด้วยส่วนสำคัญ 2 อย่างคือ

- Width คือความกว้างของภาพวิดีโอ (pixels)

- Height คือความสูงของภาพวิดีโอ (pixels)

เสียง(Audio) ประกอบด้วยส่วนสำคัญ 3 อย่าง คือ

- Duration คือช่วงเวลาของเสียง (00.00.00)

- Bit Rate คืออัตราการบีบอัดข้อมูลเสียง (มีหน่วยเป็น kbps)

- Audio Format คือรูปแบบการเข้ารหัสไฟล์เสียง เช่น (mp3 , .wma , wav)

วีดีโอ(Video) ประกอบด้วยส่วนสำคัญ 4 อย่าง คือ

- Frame Rate คือ ความเร็วในการแสดงผลภาพเคลื่อนไหว โดยมีหน่วยเป็นเฟรมต่อวินาที (.Fps)

- Data rate คือ กรบีบอัดข้อมูลเสียงและภาพวิดีโอ โดยมีตัวเลขบอกเป็นกิโลบิตต่อวินาที (Kpbs) หากผู้ใช้งานกำหนดค่านี้สูง จะทำให้คุณภาพของเสียงและภาพมีความคมชัดยิ่งขึ้น แต่ขนาดไฟล์ก็จะมีขนาดใหญ่ขึ้นด้วย

- Video Sample Size การแสดงผลความละเอียดต่อพิกเซลโดยมีหน่วยเป็นบิต (bit)

- Video compression เป็นเทคโนโลยีการเข้ารหัสข้อมูลซึ่งมีผลโดยตรงต่อคุณภาพของวีดีโอ และเป็นตัวกำหนดว่าวีดีโอนั้นจะใช้ฟอร์แมตใด

2.6 Processing

Processing [6] เป็นซอฟท์แวร์ระบบเปิดเหมาะสำหรับผู้ต้องการพัฒนาโปรแกรมเกี่ยวกับการสร้างภาพเคลื่อนไหวและการมีปฏิสัมพันธ์สำหรับผู้ที่เคยใช้ชุดพัฒนาโปรแกรม Arduino เมื่อเห็นรูปร่างหน้าตาการอินเตอร์เฟสของ Processing จะเห็นว่ามีความคล้ายคลึงกัน อีกทั้ง Processing และ Arduino ใช้หลักการในการเขียนโปรแกรมเหมือนกันโดยมีพื้นฐานมาจากภาษา Java

2.6.1 คุณสมบัติ Processing

- ใช้ได้กับระบบปฏิบัติการ Windows, Linux, Mac OS X

- เป็นซอฟต์แวร์ประเภท เปิดเผยโค้ดต้นฉบับ (Opensource)

- สามารถใช้สร้างรูปกราฟิกแบบ 2 และ 3 มิติ (2Dและ3D) หรือแบบมีปฏิสัมพันธ์ (Interactive) กับผู้ใช้ได้ เช่น ในการเรียนรู้การสร้างเกมส์คอมพิวเตอร์เบื้องต้น หรือ การนำเสนอข้อมูลในรูปแบบต่างๆ (Data Visualization)

- ทำให้ผู้เรียนได้เห็นความเชื่อมโยงระหว่างการเขียนโค้ดและสิ่งที่ปรากฏเห็นได้อันเป็นผลมาจากการทำงานของโปรแกรม

- รองรับการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)

- มีตัวอย่างการใช้งานมากมายลองดูได้จาก (http://openprocessing.org/) และแหล่งข้อมูลอ้างอิง รวมทั้งหนังสือให้ศึกษาได้

- มีความเชื่อมโยงกับภาษา Java

2.6.2 การติดตั้ง

1. ไปที่ http://processing.org/ เพื่อดาวน์โหลดโปรแกรม Processing ซึ่งมีให้เลือกทั้งระบบปฏิบัติการ Linux, Mac, OSX และ Windows สำหรับ Windows มีให้เลือกทั้งแบบ มี Java และไม่มี Java (สำหรับเบราว์เซอร์ที่มี Java อยู่แล้ว) ในที่นี้ได้ดาวน์โหลด processing-2.2.1-windows-expert.zip เป็น รุ่น 2.2.1 แบบไม่มี Java ถ้าต้องการความสะดวกสบายควรใช้รุ่นที่มี Java แต่ก็อาจไม่ได้ Java รุ่นล่าสุด

2. ให้แตกไฟล์จะได้โฟลเดอร์ Processing-2.2.1 จะย้ายไปไว้ที่ไหนก็ได้ เช่นย้ายไปเก็บที่ C:/Program File เวลาจะเปิดโปรแกรมให้เข้าไปดับเบิลคลิกไฟล์ Processing.exe ในโฟลเดอร์ Processing-1.5.1 หรือจะสร้างช็อตคัต (Shortcut) แล้วดับเบิลคลิกที่ช็อตคัต หน้าตา Windows ของ Processing จะปรากฏขึ้น

2.7 ภาษาจาวา (Java)

จาวา (Java) หรือ Java programming language [1] คือภาษาโปรแกรมเชิงวัตถุพัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆที่บริษัท ซันไมโครซิสเต็มส์ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของเจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทนจุดเด่นของภาษาจาวา (Java) อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) มาพัฒนาโปรแกรมของตนด้วย Java ได้

ภาษาจาวา (Java) เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP:Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บ เมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

2.8 OpenCV (Open Source Computer Vision Library)

OpenCV [2-3] เป็นไลบรารีสำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) ซึ่งบางความสามารถของ OpenCV สามารถทำภาพเบลอ หา threshold หา Histogram ของภาพได้แต่

ความสามารถโดยส่วนใหญ่แล้วจะทำการค้นหาขอบของภาพการตรวจสอบการเคลื่อนไหวและทำการแยกบริเวณรูปภาพ (Image segmentation) นอกจากนี้ OpenCV สามารถจัดการกับข้อมูลแบบวิดีโอได้ด้วย เนื่องจาก OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม เมื่อต้องการเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งเหล่านั้นซึ่งภาษาที่นิยมเขียนคือภาษา C, C++ และภาษาไพธอน (Phyton) ซึ่ง OpenCV จะประกอบด้วยสองส่วนคือ โครงสร้างข้อมูล (Data structure) ที่ใช้ในการเก็บข้อมูลต่างๆอาทิ เช่น รูปภาพ เมตริกซ์และพิกัดสำหรับอีกส่วนคือ อัลกอริทึ่ม (Algorithm) ซึ่งจะใช้ในการประมวลผลต่างๆ โดยเฉพาะการประมวลทางรูปภาพสำหรับใน OpenCV จะประกอบด้วยไลบรารีอยู่ 4 ส่วน ได้แก่

CXCORE เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุดขนาด อาเรย์ (Array) หน่วยความจำคำสั่งในการวาดภาพ การประกาศตัวแปรภาพ เป็นต้น ตัวอย่างคำสั่งในการประกาศรูปภาพ คือ IplImage, CvMat, CvMatND

CV ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับพิกเซลที่เป็นอาเรย์ 2 มิติ หรือที่เราเรียกว่าภาพนั่นเองเช่นการหาขอบหรือมุมการทำฮิสโตแกรม (Histogram) และการทำออพติคอลโฟลว์ (Optical Flow) เป็นต้น

การเรียนรู้ของเครื่อง (Machine Learning) เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติการแยกคลาสและการแบ่งกลุ่มข้อมูลอัลกอริทึม (Algorithm) ที่ใช้จะเขียนด้วยภาษา C++ แต่ละอัลกอริทึมจะมีคุณลักษณะเด่นแตกต่างกันไปแต่ทั้งหมดจะใช้คลาส CvStatModel ร่วมกัน

HighGUI เป็นไลบรารีที่ใช้ในการ โหลด (Load) และบันทึกภาพติดต่อกับกล้องวีดีโอ (VDO) การสร้างหน้าต่างเพื่อแสดงภาพและทำลายภาพการเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่างรวมไปถึงการตรวจสอบเมาส์ (Mouse) และ แป้นพิมพ์ (Keyboard)

2.9 ระบบตรวจจับใบหน้า ของ OpenCV

การตรวจจับใบหน้าของ OpenCV ใช้วิธีของ Paul Viola และ Michael Jones [4] ซึ่งถูกตีพิมพ์ในปี 2001 ถูกเรียกง่ายๆว่า "Viola-Jones method" โดยใช้การตรวจจับวัตถุในภาพรวมกับแนวคิดหลัก 4 แนวคิดดังนี้

1. ใช้ Haar features
2. ใช้การรวมภาพ (Integral Image) เพื่อตรวจจับลักษณะเด่นอย่างรวดเร็ว (Rapid Feature Detection)
3. ใช้ The Adaboost machine-learning method
4. ใช้ Cascaded classifier เพื่อรวมลักษณะเด่นต่างๆเข้าด้วยกันอย่างมีประสิทธิภาพ

การตรวจสอบหาลักษณะเด่นของ Haar ในแต่ละส่วนของภาพอย่างมีประสิทธิภาพนั้น Paul Viola และMichael Jones ใช้เทคนิคที่เรียกว่าการรวมภาพ (Integral Image) โดยการรวม

จุดภาพเล็กๆเข้าด้วยกัน ในการเลือกลักษณะเด่นของ Haar และการตั้งระดับค่าขีดแบ่ง Viola และ Jones ใช้ machine-learning method ที่เรียกว่า "Adaboost" ซึ่งรวมตัวจำแนกแบบอ่อนแอ (Weak classifier) ได้รับคำตอบที่ถูกต้องมากกว่าการเดาสุ่มเพียงเล็กน้อยจำนวนมากเข้าด้วยกัน เพื่อสร้างตัวจำแนกแบบแข็ง(Strong classifier) โดยใช้หลักที่ว่า ตัวจำแนกแบบอ่อนแอ (Weak classifier) แต่ละตัวค่อยๆผลักดันคำตอบสุดท้ายด้วยการสร้างการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

2.9.1 Haar –like

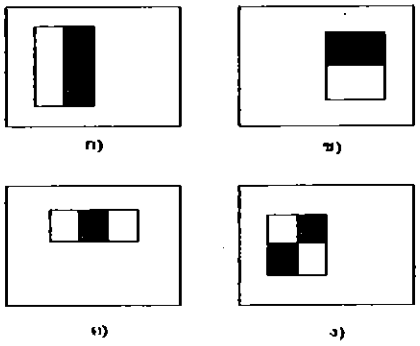
รูปแบบจำลอง Haar-like [7]ใช้กฎเกณฑ์พื้นฐานความรู้ของมนุษย์ของส่วนประกอบสำคัญบนใบหน้าโดยที่สังเกตจากโปรไฟล์ความเข้มของพิกเซลภาพทั้งในแนวตั้ง และแนวนอน ซึ่งในรูปที่มีใบหน้าจะต้องประกอบด้วยลักษณะของดวงตา 2 ดวง ซึ่งอยู่ในตำแหน่งที่สมมาตรกันในแนวระนาบทั้งสองด้านจะต้องมีหนึ่งจมูกและหนึ่งปาก ซึ่งจุดสำคัญดังกล่าวจะมีความสัมพันธ์กันโดยใช้ระยะทางและตำแหน่งตามกฎเกณฑ์ที่ได้ตั้งไว้

การหาค่ารูปแบบจำลอง Haar-like คือการหาค่าความเข้มที่แตกต่างระหว่างส่วนที่แรงเงากับส่วนที่ไม่ได้แรงเงาจากนั้นนำผลที่ได้ไปเปรียบเทียบกับค่าขีดแบ่ง (Threshold) กับขั้ว (Polarity) ที่ใช้ในการตัดสินใจว่าภาพที่รับมาควรถูกจะถูกจัดให้เป็นบวก (ภาพใบหน้า) หรือเป็นลบ (ไม่ใช่ภาพใบหน้า) ผลลัพธ์ที่ได้จะถูกนำไปพิจารณาในขั้นต่อไปการจำลองรูปแบบ Harr-like ที่แสดงในรูปที่ 2.3 มี 3 แบบคือ

1. การจำลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสองภาพหรือ Two-rectangle feature ดังรูปที่ 2.3ก และ 2.3ข

2. การจำลองรูปแบบ Haarlikeที่ประกอบด้วยพื้นที่สี่เหลี่ยมสามภาพหรือ Three-rectangle feature ดังรูปที่ 2.3ค

3.การจำลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสี่ภาพหรือ Four-rectangle feature ดังรูปที่ 2.3ง



รูปที่ 2.3 ตัวอย่างการจำลองรูปแบบ Haar-like

2.9.2 การรวมภาพ (Integral image)

ในการเพิ่มประสิทธิภาพ Viola และ Jones ได้นำเสนอให้ใช้เทคนิคที่เรียกว่า Integral Image [7] ในการคำนวณการจำลองรูปแบบ Haar-like เทคนิค Integral image คือการรวมความเข้ม (Intensity) ของแต่ละพิกเซลเข้าด้วยกัน รูปที่ 2.4 ก แสดง Integral image ของรูป 2.4ข การคำนวณหา Integral image ที่จุด(x, y)สามารถเขียนแทนได้ด้วยสมการที่ 1

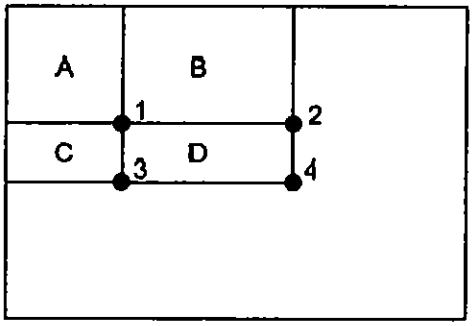
$$ii(x,y) = \sum x' \leq x, y \leq y' \, i(x',y') \qquad (1)$$

เมื่อ $ii(x,y)$ คือ ค่าของ Integral image ที่ตำแหน่งจุดที่ตำแหน่งจุด (x,y)และ $i(x',y')$ คือ ค่าความเข้มในแต่ละพิกเซลของภาพต้นฉบับ

ก) ข)

รูปที่ 2.4 ตัวอย่างภาพ Integral image

การคำนวณหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยม D โดยใช้ค่า Integral image จำนวน 4 จุด ผลลัพธ์ที่ได้คือ 4+1-(2+3) ดังแสดงในรูปที่ 2.5 ซึ่งจะเห็นได้ว่าไม่ว่าสี่เหลี่ยมจะมีขนาดเท่าใดก็จะใช้เวลาในการคำนวณค่าผลรวมความเข้มคงที่



รูปที่ 2.5 ตัวอย่างการคำนวณผลรวมความเข้มในพื้นที่สี่เหลี่ยม D โดยวิธี Integral image

กำหนดให้การจำลองรูปแบบ Haar-like ที่ได้หลังจากทำการเปรียบเทียบกับค่าขีดแบ่ง เรียกว่าตัวจำแนกแบบอ่อนแอ (Weak Classifier) เนื่องจากการจำลองรูปแบบ Haar-like ที่ได้รับจะให้คำตอบที่ถูกต้องมากกว่าการเดาสุ่มเพียงเล็กน้อย ซึ่งการหาตัวจำแนกแบบอ่อนแอสามารถเขียนแทนด้วยสมการที่ 2

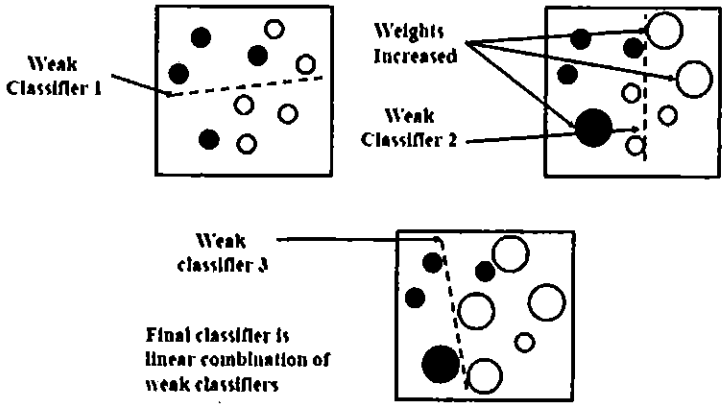
$$h(x_i, f, p, y_2, \theta) = \begin{cases} 1 \; if \; pf(x_i) < p\theta \\ 0 \qquad otherwise \end{cases} \tag{2}$$

เมื่อ x คือ ภาพย่อยได้กำหนดให้มีขนาดเท่ากับ 24 x 24 พิกเซลส่วน $h(x_i, f, p, y_2, \theta)$ คือตัวจำแนกแบบอ่อนแอที่พิจารณาจากการจำลองรูปแบบ Haar-like (f) เทียบกับค่าขีดแบ่ง (θ) และ (p) คือ ขั้วซึ่งเป็นตัวกำหนดทิศของสมการ

2.9.3 Adaboost

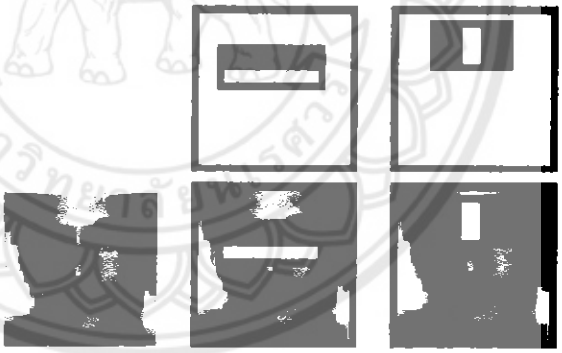
Adaboost [4] คือเทคนิคการตรวจจับใบหน้าของ Viola-Jones ในขั้นตอนนี้จะนำการจำลองรูปแบบ Haar-like ที่ได้จากขั้นตอนแรกมาเข้ากระบวนการเรียนรู้ของเครื่อง (Machine Learning) ที่เรียกว่า "Adaptive Boost" หรือ"Adaboost" ซึ่งกระบวนการนี้ใช้การเร่งหาตัวจำแนกแบบอ่อนแอ (Weak Classifier) ดังรูป 2.6 และกำหนดค่าถ่วงน้ำหนักที่ทำให้ค่าผิดพลาดน้อยที่สุดในแต่ละรอบของกระบวนการเพื่อสร้างตัวจำแนกแบบแข็งแรง (Strong Classifier) ขั้นตอนการเรียนรู้ด้วย Adaboost มีดังต่อไปนี้

- เริ่มแรกกำหนด ค่าน้ำหนักให้กับ Feature ที่วิ่งหาภายในภาพตัวอย่าง
- หาบริเวณที่ประกอบด้วย ส่วนที่เราต้องการ
- เพิ่มค่า น้ำหนักให้กับส่วนที่เหลือ เฉพาะลักษณะที่เราต้องการ ที่ยังไม่ได้ แบ่งลักษณะไว้
- ทำวนเช่นนี้ซ้ำไปเรื่อยๆ จนสุดท้าย นำบริเวณที่ได้ทั้งหมดมารวมกัน จะได้บริเวณของ Object ที่เราต้องการหา และลักษณะในส่วนต่างๆภายใน Object นั้น



รูปที่ 2.6 การทำงานของ Adaboost

ตัวอย่างของการจำลองรูปแบบ Haar-like สองลำดับแรกที่ได้จากกระบวนการเรียนรู้ด้วยวิธี Adaboost ในรูปที่ 2.7 สังเกตเห็นว่าการจำลองรูปแบบ Haar-like แรกเป็นการจำลองรูปแบบ Haarlikeที่ได้จากความแตกต่างบริเวณดวงตาซึ่งจะมืดกว่าบริเวณส่วนบนของโหนกแก้ม สำหรับการจำลองรูปแบบ Haar-like ตัวที่สองได้จากความแตกต่างบริเวณดวงตาสองข้างซึ่งจะมืดกว่าบริเวณสันจมูก

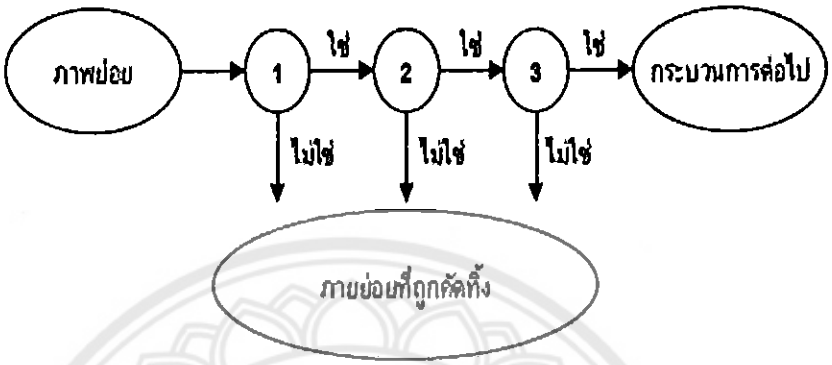


รูปที่ 2.7 แสดงตัวอย่างของตัวจำแนกแบบแข็งแรงที่ได้จากกระบวนการเรียนรู้ด้วย Adaboost

2.9.4 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

ขั้นตอนนี้ ที่ Viola-Jones นำเสนอเป็นเทคนิคที่ใช้ในการเพิ่มประสิทธิภาพในการตรวจจับให้มีความถูกต้องและยังใช้เวลาในการคำนวณลดลงด้วยโดยแนวความคิดของขั้นตอนนี้คือการสร้างการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier) [7] โดยเทคนิคนี้จะนำตัวจำแนก (Classifier) หลายๆตัวต่อกันเป็นลำดับ ดังแสดงในรูปที่ 2.8 โดยที่ตัวจำแนกในลำดับต้นๆมักจะมีความซับซ้อนที่น้อยกว่าอาจสร้างมาจากตัวจำแนกแบบอ่อนแอ (Weak Classifier)ไม่กี่ตัวเพื่อลดปริมาณการคำนวณและลดระดับของค่าแบ่งขีด (Threshold) ลงเพื่อเพิ่มลดการความผิดพลาด

ในการตรวจจับไม่พบใบหน้าซึ่งภาพย่อยที่ถูกจัดประเภทเป็นไม่ใช่ใบหน้า (Non-face) จะถูกทิ้งไปเป็นจำนวนมากในลำดับต้นๆในทางตรงข้ามถ้าภาพย่อยนั้นถูกจำแนกว่ามีโอกาสเป็นภาพใบหน้าจะถูกส่งต่อไปยังตัวจำแนกตัวถัดไปซึ่งมีความซับซ้อนสูงขึ้นตามลำดับและกล่าวได้ว่ายิ่งมีจำนวนชั้นของตัวจำแนกมากเท่าใดโอกาสที่ภาพย่อยจะเป็นใบหน้าจะยิ่งมีมากขึ้น



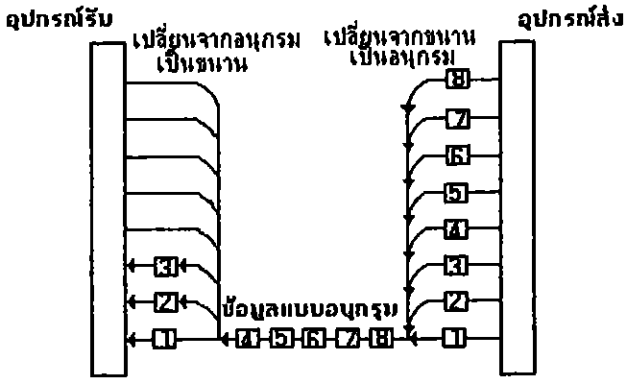
รูปที่ 2.8 แสดงขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

2.10 การสื่อสารอนุกรม

การเชื่อมต่อระบบไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกส่วนใหญ่จะใช้การเชื่อมต่อแบบขนาน (Parallel transmission) กับแบบอนุกรม (Serial transmission) [8] สำหรับการเชื่อมต่อแบบอนุกรมนิยมใช้กันมาก เช่น การเคลื่อนย้ายกันระหว่างไมโครคอมพิวเตอร์ หรือ อุปกรณ์เสริมต่างๆ เช่น mouse เป็นต้น

2.10.1 วิธีการถ่ายโอนข้อมูล

ในการถ่ายโอนข้อมูลแบบอนุกรมนั้น ข้อมูลจะได้รับการส่งออกมาครั้งละ 1 บิตระหว่างจุดรับและจุดส่ง จะเห็นว่ามีการส่งข้อมูลแบบอนุกรมนี้จะช้ากว่าการส่งข้อมูลแบบขนาน แต่ยังคงใช้อยู่ก็เพราะ ตัวกลางการสื่อสารต้องการช่องเดียวหรือมีสายเพียงคู่เดียวซึ่งจะประหยัดค่าใช้จ่าย ในการใช้ตัวกลางมากกว่าแบบขนานซึ่งถ้าเป็นระยะทางไกลจะดีเพราะเรามีระบบการสื่อสารทางโทรศัพท์อยู่แล้ว จึงสามารถนำมาใช้ในการส่งข้อมูลแบบอนุกรมนี้ได้ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรมมีหน่วยวัดเป็น บิตต่อวินาที หรือที่เรียกว่า บีพีเอส (bps)



รูปที่ 2.9 การถ่ายโอนข้อมูลแบบอนุกรม

จากรูป 2.9 การส่งข้อมูลแบบขนานข้อมูลจะถูกเปลี่ยนให้เป็นแบบอนุกรมเสียก่อนแล้วค่อยทยอยส่งครั้งละ 1 บิต ไปยังที่จะรับ ณ จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาครั้งละบิตให้เป็นสัญญาณแบบขนานซึ่งลงตัวพอดีนั้น คือ บิตที่ 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดีการที่จะทำให้การแปลงสัญญาณจากแบบอนุกรม ครั้งละบิตให้ลงพอดีนั้น จำเป็นต้องมีกลไกที่เหมาะสมเพื่อป้องกันการผิดพลาดจากการรับกลไกที่ว่าแบ่ง ออกเป็น 2 แบบ คือ แบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

รูปแบบของการติดต่อสื่อสารแบบอนุกรม

1. แบบทิศทางเดียว (Simplex) สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว ซึ่งโครงงานนี้ได้ใช้รูปแบบการส่งข้อมูลแบบทิศทางเดียว

2. แบบกึ่งสองทิศทาง (Half – Duplex) สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน

3. แบบสองทิศทาง (Full – Duplex) สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

2.10.2 มาตรฐาน USB (Universal Serial Bus)

USB หรือ Universal Serial Bus ถูกวางมาตรฐานโดยบริษัทยักษ์ใหญ่ผู้นำทางด้านอุปกรณ์ไฟฟ้าอิเล็กทรอนิกส์และคอมพิวเตอร์ โดยในยุคเริ่มแรกนั้น ก็มี COMPAQ, IBM, DEC, Intel, Microsoft, NEC และ Northern Telecom มาตรฐานของ USB นั้นออกสู่สาธารณะชนเป็นครั้งแรกเมื่อวันที่ 11 พฤศจิกายน ปี พ.ศ.2537 ได้ปรับปรุงแก้ไขเรื่อยมาจนกระทั่งเมื่อ วันที่ 15 มกราคม พ.ศ. 2539 ออกมาเป็น Revision 1.0 (USB1.0) รูปที่ 2.10 ได้สำเร็จและยังได้ปรับปรุงแก้ไขปัญหาต่างๆจนเมื่อวันที่ 23 กันยายน พ .ศ.2541 ได้เป็น Revision 1.1 (USB1.1) รูปที่ 2.11 เมื่อความเร็วที่ได้ยังไม่เพียงพอกับความต้องการดังนั้นทางกลุ่มผู้พัฒนาหรือ USB-IF (USB Implementers Forum, Inc.) ได้ร่างมาตรฐาน USB รุ่นใหม่ความเร็วในการรับ-ส่งข้อมูลนั้น USB1.1 จะมีความเร็วอยู่ที่ 12Mbps ส่วน USB 2.0 นั้นรองรับระดับการรับส่งข้อมูลได้ถึง 3 ระดับ

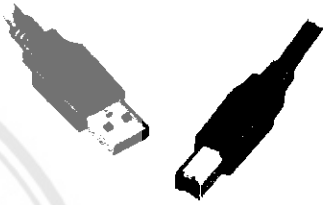
-ความเร็ว 1.5 Mbps (Low Speed) สำหรับการเชื่อมต่อกับอุปกรณ์ที่ไม่จำเป็นต้องส่งข้อมูลคราวละมากๆ

-ความเร็ว 12 Mbps (Full Speed) สำหรับการเชื่อมต่อกับ USB 1.1

-ความเร็ว 480 Mbps (Hi-Speed) สำหรับการเชื่อมต่อกับ USB 2.0

โครงงานนี้ได้ใช้ USB 2.0 ซึ่งมีความเร็ว 480 Mbps

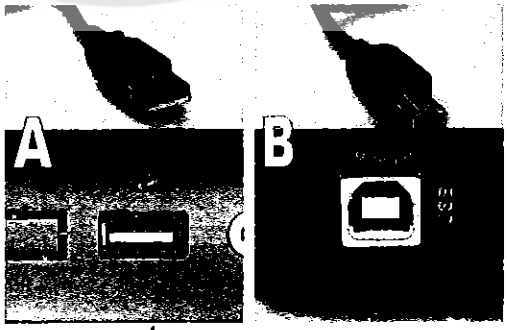
รูปที่ 2.10 USB1.0



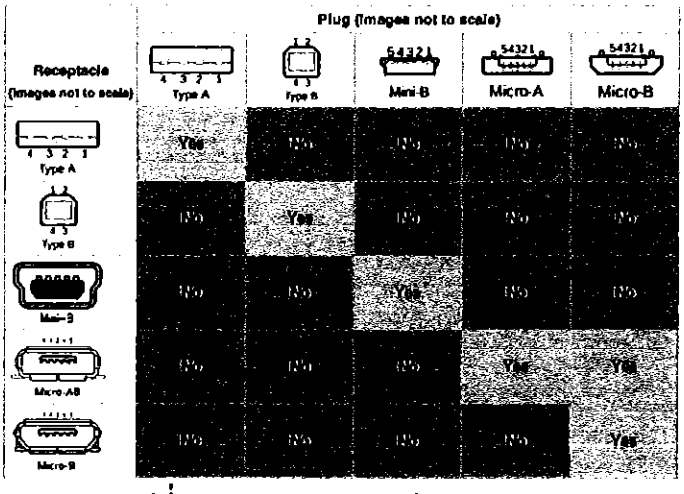
รูปที่ 2.11 USB1.1

2.10.3 ลักษณะของคอนเน็คเตอร์และเต้ารับ

หัวต่อแบบที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้น จะมีอยู่ 2 คือแบบ A และแบบ B (รูปที่ 2.12) ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน (รูปที่ 2.13)



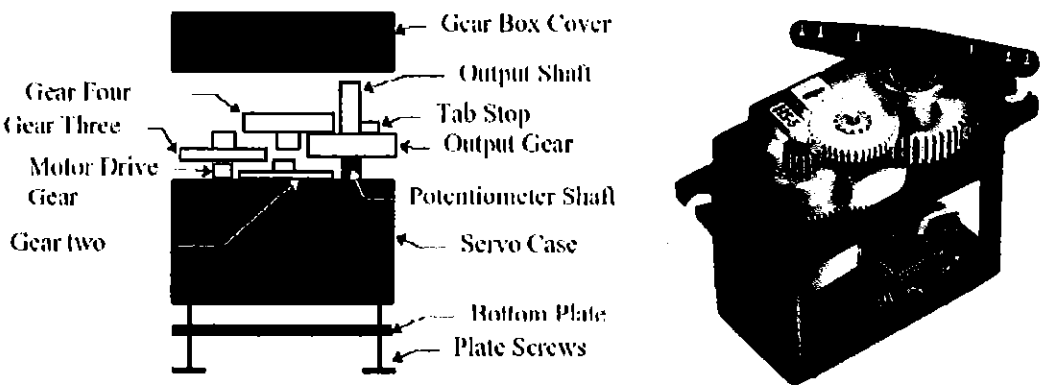
รูปที่ 2.12 หัวต่อและเต้ารับ



รูปที่ 2.13 รูปแบบการจัดเรียง

2.11 เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ (Servo motor) คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับชุดเกียร์และส่วนควบคุมต่างๆ ไว้ในมอดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกันโดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ ไฟเลี้ยง, กราวด์ และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวาได้จากสายสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณ Pulse Width Modulation (PWM) มอเตอร์ชนิดนี้จะมีวงจรควบคุมให้หมุนไปในตำแหน่งหรือทิศทางองศาที่ต้องการได้โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180° หรือ ครึ่งรอบเท่านั้นหรือบางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้เนื่องจากโครงสร้างภายในจะประกอบด้วยตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์และตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งส่วนภายในเซอร์โวมอเตอร์ (Servo motor) โดยทั่วไปเป็นไปตามรูปที่ 2.14



รูปที่ 2.14 ชิ้นส่วนภายในของ Servo Motor

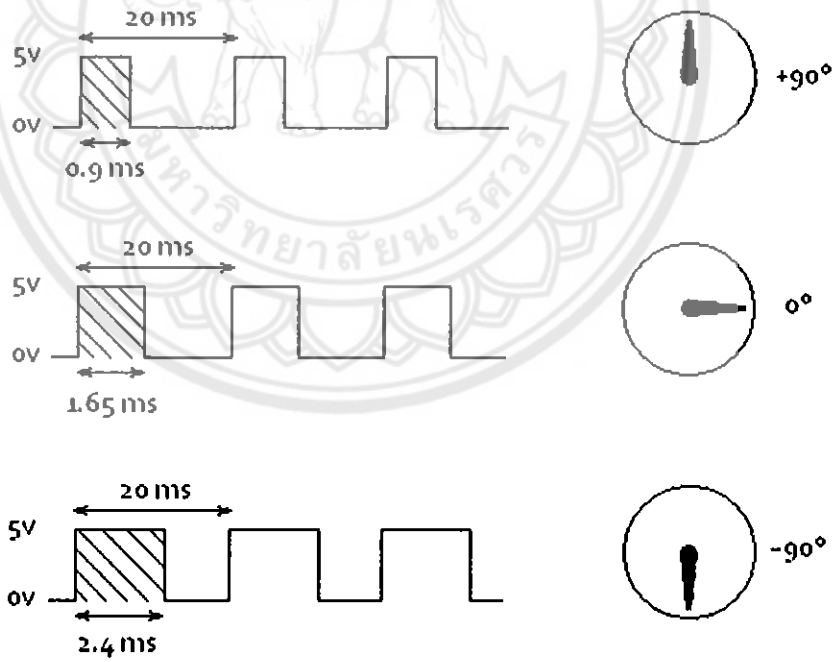
ข้อดีของมอเตอร์ชนิดนี้คือ

มีขนาดเล็ก น้ำหนักเบา ให้แรงบิดสูง กินพลังงานน้อยและสามารถควบคุมด้วยแรงดันลอจิกที่เป็นTTL ได้โดยตรงโดยไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่น ๆ

2.11.1 หลักการทางานของเซอร์โวมอเตอร์ (Servo motor)

การควบคุมการทำงานของเซอร์โวมอเตอร์ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูป 2.15 คือ

- สัญญาณความกว้างพัลส์ขนาด 1.65 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศาหรือจุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 2.4 ms จะควบคุมเซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศาหรือ ในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 0.9 ms จะควบคุมเซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือ ในทิศทางตามเข็มนาฬิกา



รูปที่ 2.15 การควบคุม servo motor

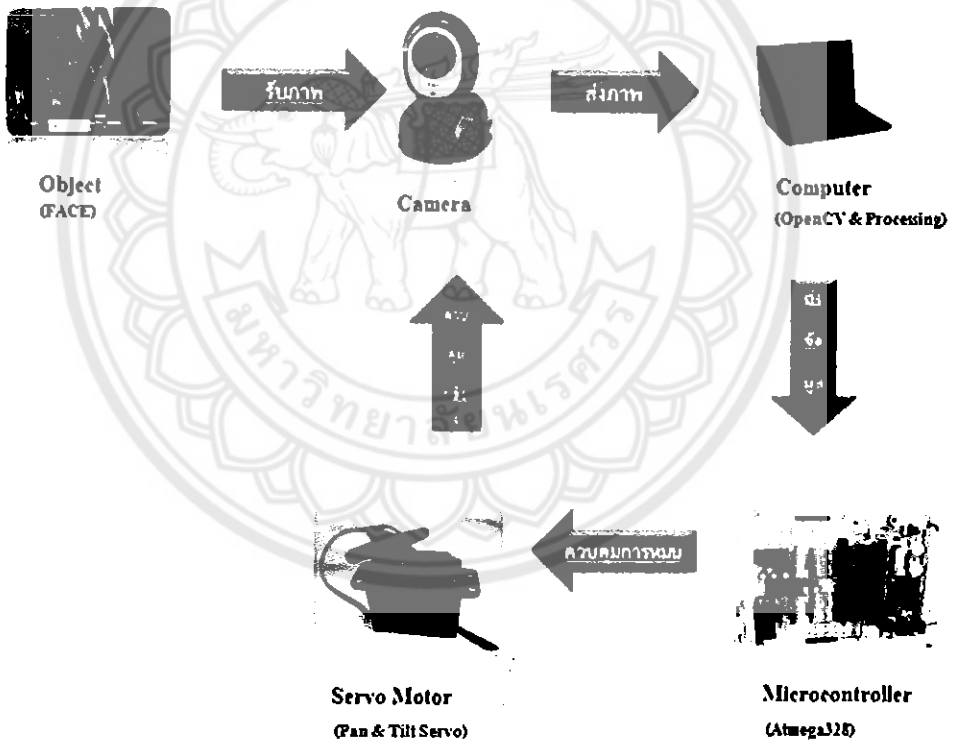
หมายเหตุ ค่าความกว้างพัลส์และระยะองศาการหมุนของมอเตอร์ที่อธิบายด้านบนนั้น เป็นเพียงค่าประมาณเท่านั้นทั้งนี้ระยะการหมุนและขนาดของพัลส์ที่ควบคุมการทางานของมอเตอร์ในแต่ละยี่ห้ออาจจะไม่เท่ากันดังนั้นในการใช้งานจึงควรศึกษารายละเอียดของมอเตอร์ในแต่ละรุ่น

บทที่ 3

ขั้นตอนการดำเนินงาน

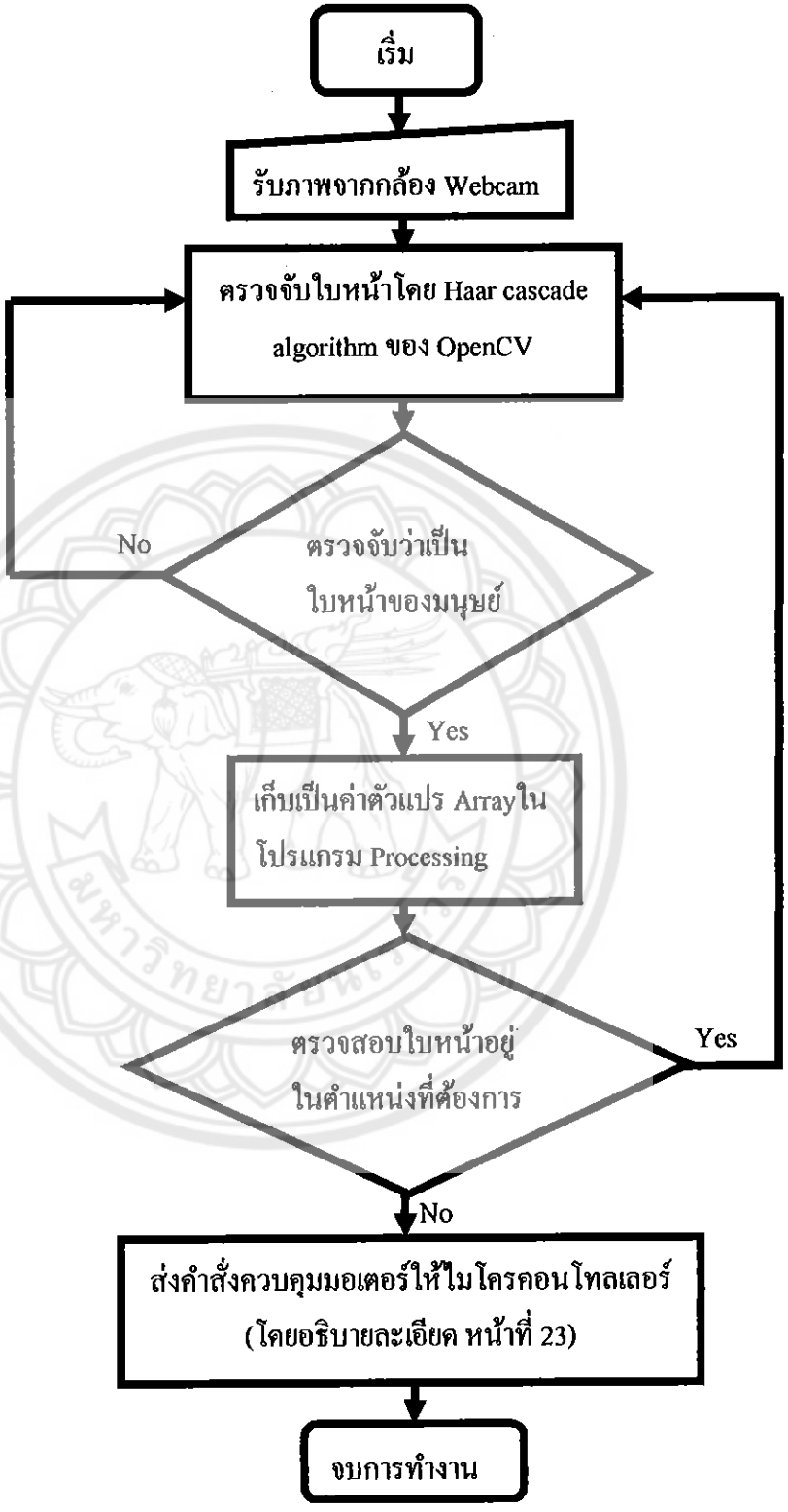
3.1 การออกแบบโปรแกรม

จากรูปที่ 3.1 โปรแกรมทำงานติดตามใบหน้าโดยใช้กล้องเว็บแคมทำหน้าที่รับภาพ แล้วทำการส่งต่อไปที่โปรแกรม Processing ที่ติดตั้งอยู่ในคอมพิวเตอร์ ซึ่งโปรแกรม Processing จะทำการวิเคราะห์ตรวจจับใบหน้าและหาตำแหน่งที่พิกัดใบหน้าจากอัลกอริทึมของ OpenCV แล้วส่งผ่าน Serial port ไปที่ไมโครคอนโทลเลอร์ ไมโครคอลโทลเลอร์จะควบคุมให้เซอร์โวมอเตอร์ หมุนไปตามทิศทางที่กล้องตรวจพบใบหน้า



รูปที่ 3.1 ระบบโดยรวม

3.1.1 ส่วนขยายของระบบตรวจจับใบหน้า (Detection)



รูปที่ 3.2 ระบบตรวจจับใบหน้า (Detection System)

จากรูป 3.2 สามารถอธิบายในส่วนขยายระบบตรวจจับใบหน้าได้ดังนี้

1. เริ่มจากคอมพิวเตอร์รับภาพจากกล้องเว็บแคมแบบ Real Time ส่งไปประมวลผลในโปรแกรม Processing โดยใช้ Library ของ Open CV

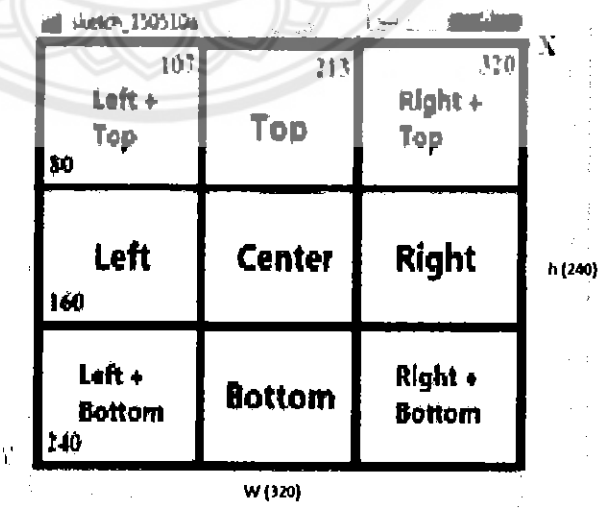
2. นำภาพแบบ Real Time มาวิเคราะห์หาใบหน้าโดยใช้อัลกอริทึม Haar-like โดยจะรับภาพจากกล้องวีดีโอเข้ามาวิเคราะห์ระบุตำแหน่งใบหน้า แล้วนำภาพที่ได้มาคำนวณหาส่วนของใบหน้า โดยใช้อัลกอริธึม Haar-Like ในการค้นหาสำหรับฟังก์ชันที่ใช้ในการค้นหาใบหน้าคือ CvHaarDetectObjects ซึ่งอยู่ในไลบรารีของ OpenCV และได้มีรูปแบบไฟล์ xml ที่ใช้ในการระบุส่วนของใบหน้าอยู่ 3 รูปแบบด้วยกัน ได้แก่

- Haarcascade_frontalface_alt.xml
- Haarcascade_frontalface_alt_tree.xml
- Haarcascade_frontalface_default.xml

โดยในการเขียนโปรแกรมในส่วนนี้ จะเลือกรูปแบบไฟล์ xml มาเพียงรูปแบบเดียว นั่นคือแบบ haarcascade_frontalface_alt.xml เพราะองศาของใบหน้าบุคคลที่เราตรวจสอบอยู่ในลักษณะแนวตั้ง ซึ่งในแต่ละรูปแบบนั้นสามารถนำมาระบุตำแหน่งใบหน้าได้เหมือนกันแต่อาจจะค้นหาได้ในองศาที่แตกต่างกันไปบ้าง เมื่อค้นหาใบหน้าเจอจะทำการเลือกบริเวณของใบหน้าโดยการวาดกรอบสี่เหลี่ยมในส่วนของใบหน้านั้น

3. นำค่าตำแหน่งของใบหน้ามาเปรียบเทียบกับขอบเขตที่เรากำหนดไว้

3.1 วิธีหาขอบเขตการทำงานโดยกำหนดให้ความกว้างของ (w) และความยาว (h) ของพิกเซลจากรูปที่ 3.3



รูปที่ 3.3 ขอบเขตควบคุมตำแหน่งใบหน้า

1. หาเส้นที่ 1 แกน X จาก $\frac{w}{3}$ จะได้ $\frac{320}{3} = 106.666 \approx 107$
2. หาเส้นที่ 2 แกน X จาก $\frac{w}{3} \times 2$ จะได้ $\frac{320}{3} \times 2 = 213.333 \approx 213$
3. หาเส้นที่ 3 แกน Y จาก $\frac{h}{3}$ จะได้ $\frac{320}{3} = 80$
4. หาเส้นที่ 4 แกน Y จาก $\frac{h}{3} \times 2$ จะได้ $\frac{320}{3} \times 2 = 160$

3.2 โดยช่วงพิกัดการแบ่งพิกเซลออกเป็น 9 ช่องดังรูป 3.3 มีค่าดังนี้

โดยกำหนดให้ความกว้างของ (w) และความยาว (h) ของพิกเซล ค่าตำแหน่งของพิกเซลบนแกนแนวระดับ(X)และค่าตำแหน่งของพิกเซลบนแกนแนวระดิ่ง(Y)

1. Center โดยที่ใบหน้าต้องอยู่ในช่วง $\frac{w}{3} < X < \frac{w}{3} \times 2$ และ $\frac{h}{3} < Y < \frac{h}{3} \times 2$
2. Botom โดยที่ใบหน้าต้องอยู่ในช่วง $\frac{X}{3} < X < \frac{X}{3} \times 2$ และ $0 < Y < \frac{h}{3}$
3. Top โดยที่ใบหน้าต้องอยู่ในช่วง $\frac{w}{3} < X < \frac{w}{3} \times 2$ และ $\frac{h}{3} \times 2 < Y < h$
4. Left โดยที่ใบหน้าต้องอยู่ในช่วง $0 < X < 107$ และ $\frac{h}{3} < Y < \frac{h}{3} \times 2$
5. Left + Top โดยที่ใบหน้าต้องอยู่ในช่วง $0 < X < \frac{w}{3}$ และ $0 < Y < \frac{h}{3}$
6. Left + Bottom โดยที่ใบหน้าต้องอยู่ในช่วง $0 < X < \frac{w}{3}$ และ $\frac{h}{3} \times 2 < Y < h$
7. Right โดยที่ใบหน้าต้องอยู่ในช่วง $\frac{w}{3} \times 2 < X < w$ และ $\frac{h}{3} < Y < \frac{h}{3} \times 2$
8. Right + Top โดยที่ $0 < X < \frac{X}{3}$ และ $0 < Y < \frac{h}{3}$
9. Right + Bottom โดยที่ $0 < X < \frac{X}{3}$ และ $\frac{h}{3} \times 2 < Y < h$

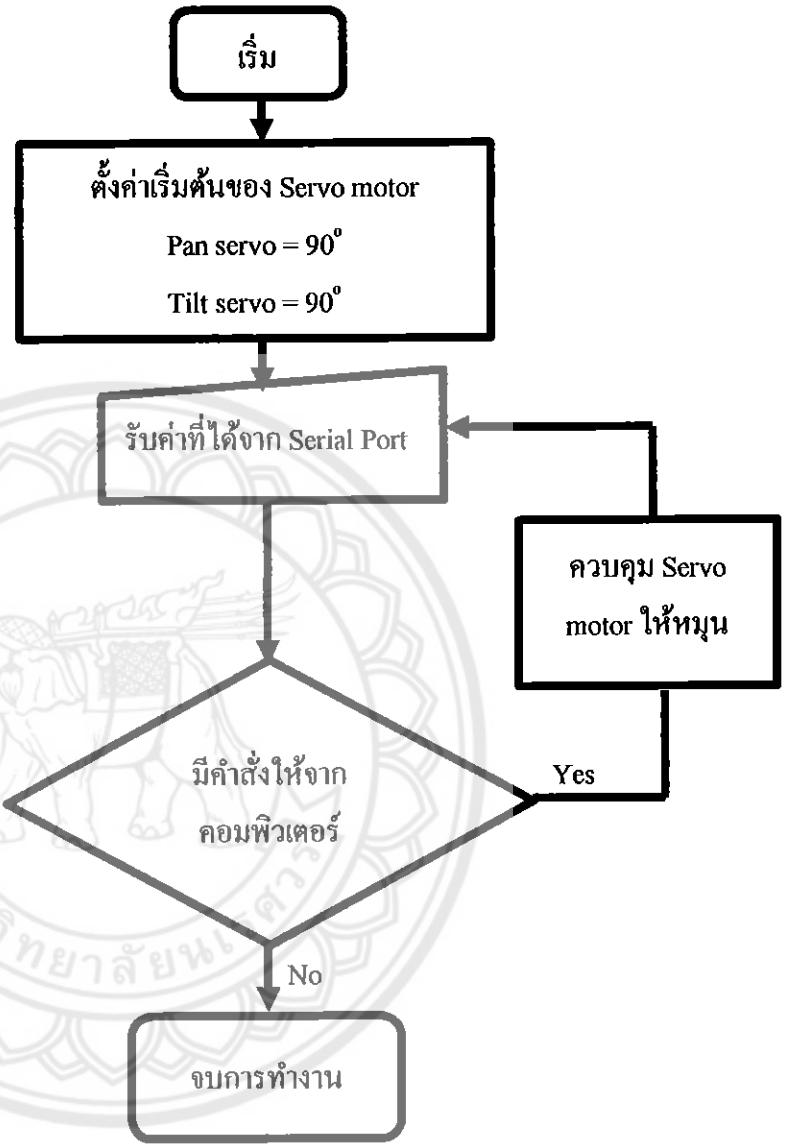
เมื่อพิกัดของใบหน้าอยู่ในตำแหน่งที่ไม่ใช่ Center โปรแกรม Processing ส่งคำสั่งไปยัง Microcontroller ทำงานตาม case ต่างๆตามรูปที่ 3.3 ที่เราตั้งไว้ดังนี้

1. ภาพอยู่ที่ตำแหน่ง Left + Top จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนซ้าย ตัวที่ 2 หมุนขึ้นตามลำดับ จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
2. ภาพอยู่ที่ตำแหน่ง Top จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 2 หมุนขึ้น จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center

3. ภาพอยู่ที่ตำแหน่ง Right + Top จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนขวา ตัวที่ 2 หมุนขึ้นตามลำดับ จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
4. ภาพอยู่ที่ตำแหน่ง Left จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนซ้าย จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
5. ภาพอยู่ที่ตำแหน่ง Right จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนขวา จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
6. ภาพอยู่ที่ตำแหน่ง Left + Bottom จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนซ้าย ตัวที่ 2 หมุนลงตามลำดับ จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
7. ภาพอยู่ที่ตำแหน่ง Bottom จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 2 หมุนลง จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center
8. ภาพอยู่ที่ตำแหน่ง Right + Bottom จะส่งคำสั่งให้ไมโครคอลโทลเลอร์ ทำการสั่งให้เซอร์โวมอเตอร์ตัวที่ 1 หมุนขวา ตัวที่ 2 หมุนลงตามลำดับ จนพิกัดภาพมาอยู่ที่ตำแหน่ง Center

1719 4870 ป
วิทยก
2557
สำนักหอสมุด
10 ต.ค. 2560

3.1.2 ส่วนขยายการทำงานของระบบควบคุมมอเตอร์



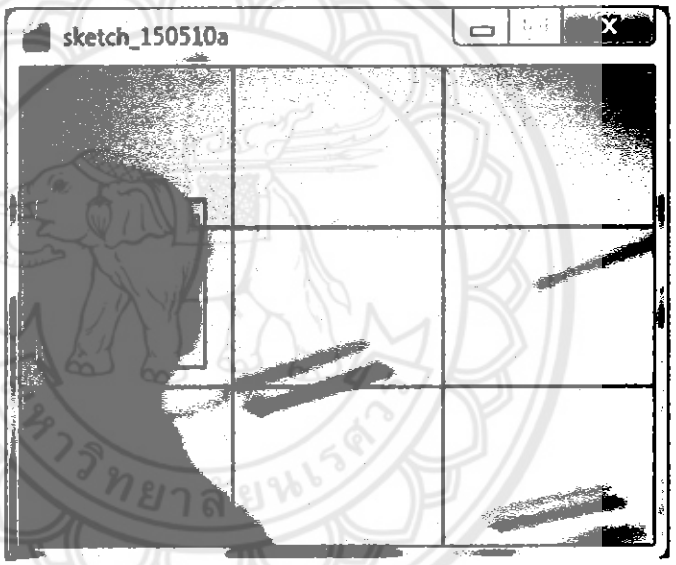
รูปที่ 3.4 การทำงานของระบบควบคุมมอเตอร์

จากรูปที่ 3.4 สามารถอธิบายส่วนขยายของระบบควบคุมมอเตอร์ได้ดังนี้

1. เริ่มโปรแกรมการทำงานของไมโครคอนโทลเลอร์
2. ไมโครคอลโทลเลอร์ทำการควบคุมเซอร์โวมอเตอร์ให้องศาการหมุนไปยังตำแหน่งของค่าตั้งต้นคือเซอร์โวมอเตอร์ในแนวดิ่งตั้งค่าเริ่มต้นไว้ที่มุม 90 องศา เซอร์โวมอเตอร์ในแนวราบตั้งค่าเริ่มต้นไว้ที่มุม 90 องศา

3. ไมโครคอลโทลเลอร์ รับค่าคำสั่งจากคอมพิวเตอร์ที่ผ่านการประมวลผลแล้ว ผ่านทาง Serial Port ที่เชื่อมต่ออยู่กับคอมพิวเตอร์
4. ไมโครคอลโทลเลอร์ทำการควบคุมเซอร์โวมอเตอร์ให้หมุนไปตามองศาที่ต้องการจากการป้อนสัญญาณพัลซ์ด้วยความกว้างที่เหมาะสม

ตัวอย่างคำสั่งการทำงาน ตรวจพบใบหน้าอยู่บริเวณด้านซ้ายของจอภาพวีดีโอดังรูปที่ 3.5 โปรแกรม Processing จะทำการตีกรอบรอบใบหน้าและเก็บค่าของความกว้างความยาวของกรอบและค่าพิกัดทั้งแกน x และแกน y เก็บไว้เป็นตัวแปร Array และนำมาเปรียบเทียบกับเงื่อนไข if , else ในโปรแกรม Processing



รูปที่ 3.5 แสดงการตรวจพบใบหน้าทางด้านซ้ายมือของกรอบวีดีโอ

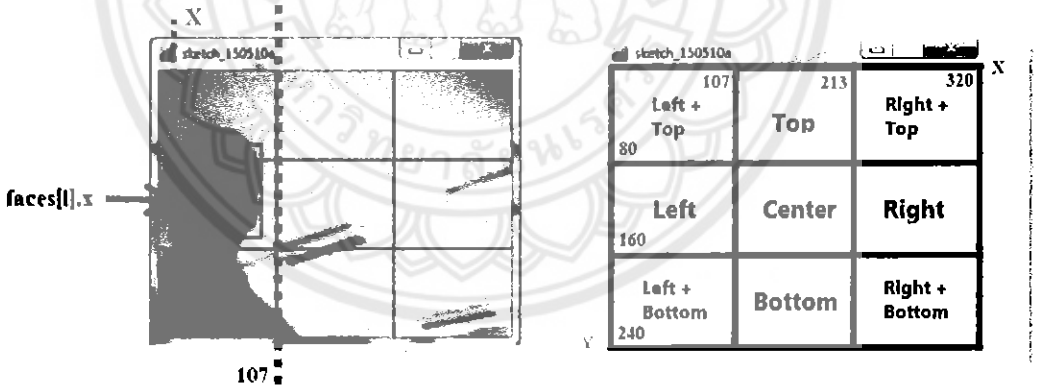
การตรวจจับใบหน้าและการสร้างกรอบสีแดงนี้จะใช้ Algorithm ของ OpenCV ในการประมวลผล แต่การเก็บค่าตัวแปรและเปรียบเทียบเงื่อนไขการทำงาน จะใช้งานในส่วนโปรแกรม Processing ดังนั้นเงื่อนไขที่ใบหน้าอยู่ด้านซ้ายของกรอบวีดีโอ จึงเป็นเงื่อนไขในโปรแกรม ดังรูปที่ 3.6

```
    for(int i=0;i<faces.length;i++){
  rect(faces[i].x,faces[i].y,faces[i].width,faces[i].height);

 if(faces[i].x<107){
   if(faces[i].x<50){
                println("Much Left");
                arduinoPort.write("a");
                delay(20);
              }
            else{
                println("Left");
                arduinoPort.write("b");
                delay(20);
    }
                }
```

รูปที่ 3.6 แสดงส่วนของเงื่อนไขคำสั่งในโปรแกรม Processing

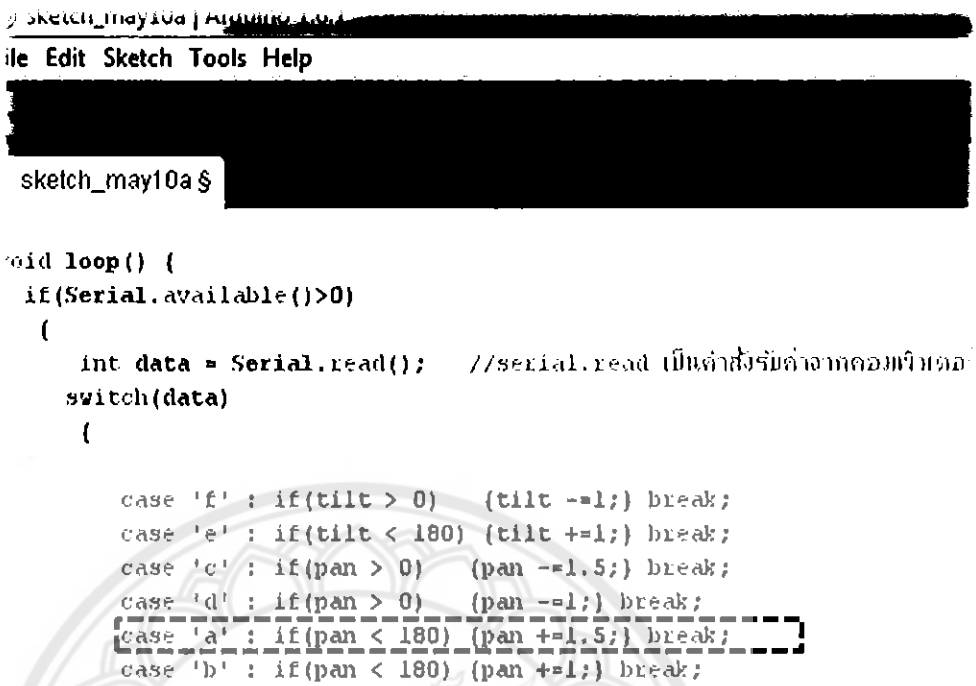
ค่าของตัวแปร faces[i].x คือค่ากรอบสีแดงรอบใบหน้าว่ามีค่าในแนวแกน x เป็นเท่าไรและค่าที่นำมาเปรียบเทียบกับตัวแปร array นี้คือค่าความยาวของกรอบวีดีโอที่แบ่งขอบเขตเป็นทั้งหมด 9 ช่องดังรูปที่ 3.7



รูปที่ 3.7 แสดงเส้นแบ่งขอบเขตของกรอบวีดีโอ

จากนั้น Processing จะส่งคำสั่ง write ไปที่ไมโครคอลโทลเลอร์ เพื่อให้เซอร์โวมอเตอร์ทำตามเงื่อนไขคำสั่งภายใน Arduino board อีกครั้ง และจากตัวอย่างนี้เป็นการสั่งให้ write เป็น case ของ ('a')

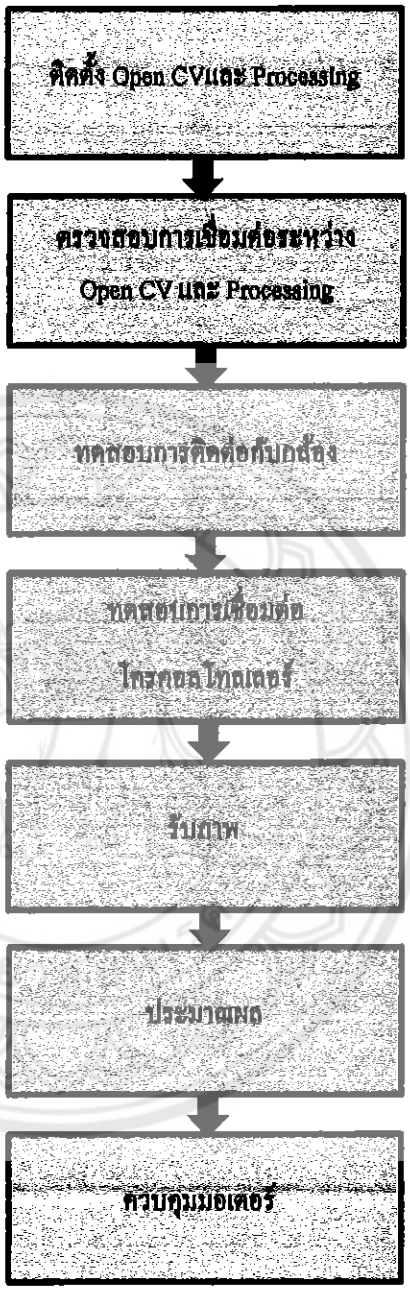
ภายใน Sketch หรือ คำสั่งการทำงานของ Arduino board ถูกตั้งให้อ่านค่าจากคอมพิวเตอร์ไว้คือส่วนของ Serial.read(); คำสั่งที่ถูกส่งมาจากโปรแกรม Processing จึงไปเข้าเงื่อนไขภายในส่วนของ Sketch ดังรูปที่ 3.8



รูปที่ 3.8 แสดงคำสั่งควบคุม Servo motor

ภายในส่วนของ case 'a' ภายในจะเป็นคำสั่งให้เซอร์โวมอเตอร์ ทำงานตามที่กำหนดไว้ ดังตัวอย่างนี้จะเป็นการสั่งให้ pan servo หรือ เซอร์โวมอเตอร์ที่ควบคุมทิศทางในแนวราบทำการหมุนเพิ่มไป 1.5 องศา และทำงานวนซ้ำจนกว่าจะไม่มีคำสั่งจากคอมพิวเตอร์หรือ อีกความหมายหนึ่งคือใบหน้านั้นอยู่ในส่วนกลางของกรอบวีดีโอ

3.2 การสร้างโปรแกรม



รูปที่ 3.9 ขั้นตอนการสร้างโปรแกรม

จากรูปที่ 3.9 หลังจากวิเคราะห์การทำงานของโปรแกรมและออกแบบโปรแกรม ขั้นตอนต่อไปคือขั้นตอนของการสร้างโปรแกรม โดยแบ่งเป็นขั้นตอนดังนี้

1. ติดตั้ง OpenCV มาช่วยในการประมวลผลภาพ 1.0 และติดตั้งโปรแกรม Processing ในการเขียนชุดคำสั่งโปรแกรมและเชื่อมต่อกับไมโครคอนโทรลเลอร์
2. เชื่อมต่อ OpenCV และ Processing โดยใช้ JavaCV ในการช่วยในการเชื่อมต่อ
3. สร้างโปรแกรมที่ติดต่อกับกล้องเว็บแคม เพื่อตรวจสอบว่าสามารถติดต่อกับกล้องได้หรือไม่
4. สร้างโปรแกรมทดสอบใน Processing ส่งคำสั่งไปยัง ไมโครคอลโทลเลอร์ผ่านทาง Serial Port
5. สร้างโปรแกรมรับภาพแบบ Real Time จากกล้องเว็บแคม
6. สร้างโปรแกรมเรียกใช้ OpenCV และการประมวลต่างๆ ของใบหน้า
7. สร้างโปรแกรมชุดคำสั่งลงในไมโครคอลโทลเลอร์เพื่อ ควบคุมเซอร์โวมอเตอร์

3.3 อุปกรณ์

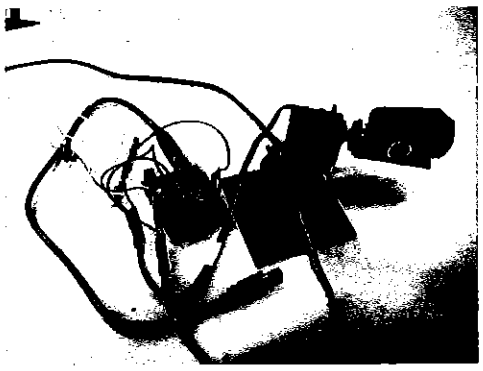
อุปกรณ์ที่ใช้ประกอบด้วย กล้องเว็บแคม , บอร์ด Aduino (Atmega 328) , เซอร์โวมอเตอร์, คอมพิวเตอร์ (windows 7 32 bit)

รูปที่ 3.10 บอร์ด Aduino (Atmega 328)

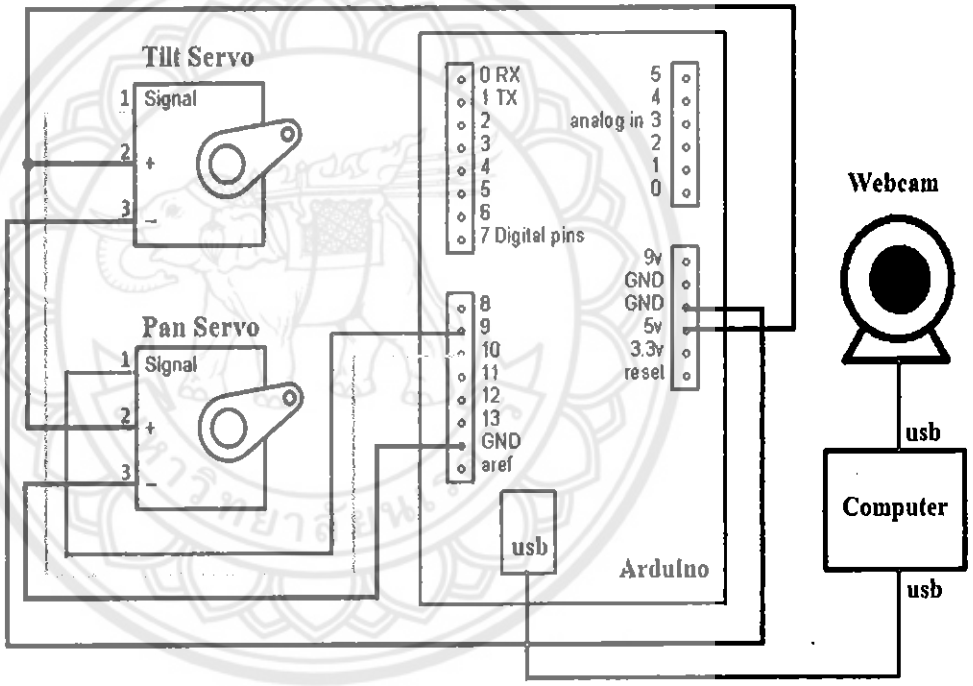
รูปที่ 3.11 เซอร์โวมอเตอร์

รูปที่ 3.12 กล้องเว็บแคม

รูปที่ 3.13 คอมพิวเตอร์



รูปที่ 3.14 ชิ้นงานของโครงงาน



รูปที่ 3.15 วงจรในโครงงาน

จากรูปที่ 3.15 เป็นการแสดงการเชื่อมต่อของอุปกรณ์ไมโครคอนโทรลเลอร์ (Arduino) ซึ่งเป็นบอร์ดที่ใช้ในการควบคุมเซอร์โวมอเตอร์ทั้งสองตัว เซอร์โวมอเตอร์ตัวแรกคือเซอร์โวมอเตอร์ที่ทำหน้าที่หมุนในแนวระดับ (Tilt Servo) ซึ่งจะใช้สายรับสัญญาณเชื่อมต่อกับบอร์ดทั้งหมด 3 เส้น คือ

1. สายรับสัญญาณ (Signal) จะเชื่อมต่อกับ Pin ขาของบอร์ดที่ทำหน้าที่ส่งสัญญาณพัลส์ (PWM) ออกมา ที่มีให้เลือกทั้งหมด 14 ขาด้วยกัน และจากตัวอย่างเลือกใช้ขาที่ 9

2. สายที่ใช้รับแรงดันไฟฟ้า (VCC) ต่อกับขาที่ 5V ของบอร์ด

3. สายกราวด์ ต่อกับขา GND ของบอร์ด

ส่วนเซอร์โวมอเตอร์ที่ทำหน้าที่หมุนในแนวราบ (Pan Servo) เลือกใช้สายรับสัญญาณต่อเข้ากับบอร์ดควบคุมขาที่ 10

การเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับคอมพิวเตอร์นี้จะใช้การเชื่อมต่อโดยผ่านสายยูเอสบี (USB) เช่นกันกับการเชื่อมต่อระหว่างกล้องวีดีโอกับคอมพิวเตอร์

บทที่ 4

ผลการทดลอง

ผลการทดลองนี้เริ่มต้นจากการทดสอบการทำงานของอุปกรณ์ต่างๆของระบบ ทดสอบฟังก์ชั่นการทำงานของโปรแกรมที่เขียนและทดสอบการทำการจริงในสภาวะที่ต่างกันเพื่อจะวิเคราะห์ว่าโปรแกรมกล้องเคลื่อนที่หมุนติดตามใบหน้าโดยใช้กล้องเว็บแคมมีขอบเขตความสามารถในการทำงานอย่างไรเพื่อที่จะได้ประสิทธิภาพในการใช้งานสูงสุด

4.1 ขั้นตอนการทดสอบการเชื่อมต่อและทำการของระบบ

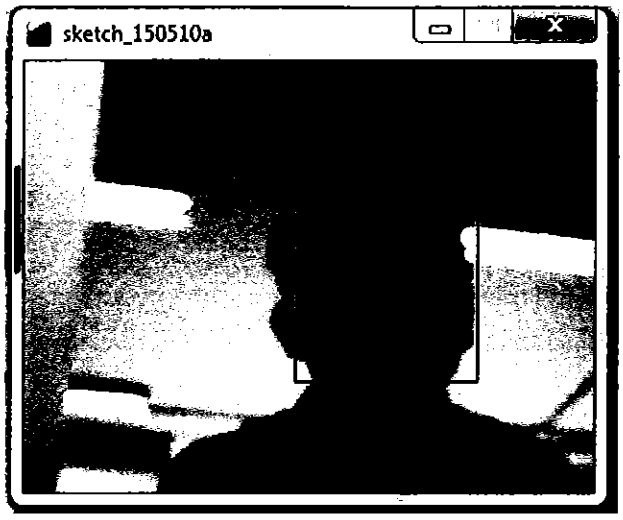
4.1.1 ทดสอบการเชื่อมกล้อง

การเชื่อมกล้องเป็นรูปแบบการทำงานของโปรแกรม Processing ที่ใช้ในลักษณะของวิดีโอทำให้เราสามารถรับภาพจากกล้องมาแสดงผลในหน้าจอคอมพิวเตอร์ได้ตามรูปที่ 4.1

รูปที่ 4.1 แสดงผลการติดต่อกับกล้อง

4.1.2 การประมาณและตรวจจับใบหน้า

จากนั้นนำวิดีโอที่ได้จากกล้องเว็บแคมเอาไปประมวลผลโดยใช้ Library ของ OpenCV เพื่อตรวจจับใบหน้าและหาตำแหน่งทำให้เราสามารถตรวจจับพบใบหน้าบนวีดีโอที่รับมาจากกล้องเว็บแคมดังรูป 4.2



รูปที่ 4.2 แสดงการหาตำแหน่งของใบหน้า

ส่วนของคำสั่งในการทดลองเปิดกล้องวีดีโอและตรวจจับใบหน้า

```
import java.awt.Rectangle;   // library ที่เกี่ยวข้องในการเชื่อมต่อ processing กับ OpenCV
import processing.video.*;
import gab.opencv.*;
import java.awt.*;
import cc.arduino.*;
import processing.serial.*;
OpenCV opencv;
Capture video;
Serial arduinoPort;
int vidWidth = ความยาว;  //กำหนดขนาดของกรอบหน้าต่างวีดีโอ
int vidLength = ความกว้าง;
void setup(){ // ทำงาน 1 ครั้ง ซึ่งเป็นฟังก์ชันสำหรับกำหนดค่าทั่วไปเช่น กำหนดค่า port
ของ arduino และเรียกใช้ algorithm ของ OpenCV
size(vidWidth, vidLength);
arduinoPort = new Serial(this, "Port  ของ microcontroller", arduino baud rate);
video = new Capture(this, vidWidth, vidLength);
 opencv = new OpenCV(this, vidWidth, vidLength);
 opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
 video.start(); //เปิดกล้องวีดีโอ
```

```
}
void draw(){    //ทำงานต่อจาก void setup และทำวนซ้ำไปเรื่อยๆภายใน void draw
จนกว่าจะสั่งหยุด
….. //เงื่อนไขในการสั่งงาน arduino microcontroller
}
void captureEvent(Capture c) {c.read();
}
```

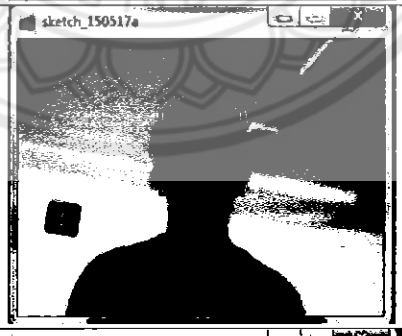
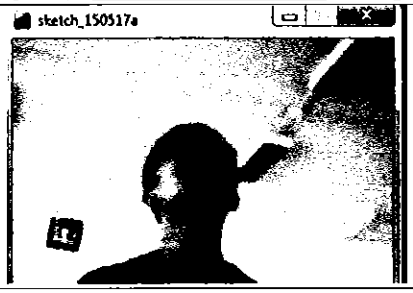
4.1.3 ทดสอบการตรวจจับใบหน้าในรูปแบบต่างๆ

เป็นการทดสอบการตรวจจับใบหน้าในรูปแบบต่างๆที่เราทำการสุ่มมาเพื่อทดลองว่าโปรแกรมเราสามารถทำงานครอบคลุมรูปแบบใบหน้าได้มาน้อยเพียงใด

ตารางที่ 4.1 ผลการทดลองตรวจจับใบหน้าในรูปแบบต่างๆ

ลำดับที่	รูปแบบใบหน้า	ผลการตรวจจับใบหน้า
1		ตรวจจับได้
2		ตรวจจับได้
3		ตรวจจับไม่ได้ เพราะองค์ประกอบของใบหน้าที่ใช้ในการตรวจจับไม่สมบูรณ์

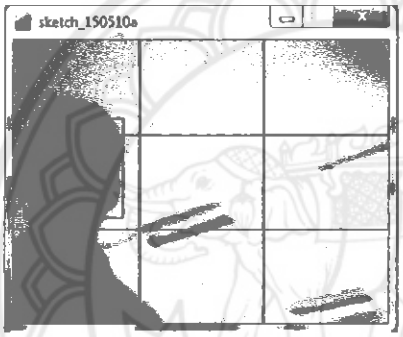
ตารางที่ 4.1(ต่อ) ผลการทดลองตรวจจับใบหน้าในรูปแบบต่างๆ

ลำดับที่	รูปแบบใบหน้า	ผลการตรวจจับ
4		ตรวจจับไม่ได้ เพราะองค์ประกอบของใบหน้าที่ใช้ในการตรวจจับไม่สมบูรณ์
5		ตรวจจับได้
6		ตรวจจับไม่ได้ เพราะองค์ประกอบของใบหน้าที่ใช้ในการตรวจจับไม่สมบูรณ์
7		ตรวจจับได้
8		ตรวจจับไม่ได้ เพราะองค์ประกอบของใบหน้าที่ใช้ในการตรวจจับไม่สมบูรณ์

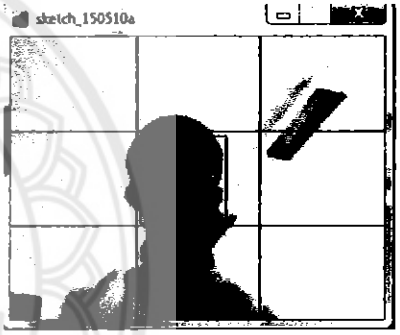
จากตารางที่ 4.1 หลังจากทดสอบสอบระบบตรวจจับใบหน้าด้วยใบหน้าที่มีรูปแบบที่แตกต่างกันแล้วพบ ว่าใบหน้าที่หันในทิศทางตรงเข้ามาหากล้อง ระบบสามารถตรวจจับได้ หรือใบหน้าที่มีการเอียงซ้ายเอียงขวาหรือเงยหน้าขึ้นลงที่ไม่มากนัก ระบบก็สามารถตรวจสอบพบใบหน้าได้ ส่วนใบหน้าที่มีองศาเอียงขึ้น ลง ซ้าย และขวา มากเกินไปนั้น จะทำให้ภาพใบหน้าที่ได้รับมา มีองค์ประกอบของใบหน้าที่ไม่สมบูรณ์ เช่น ควงตาหายไป โปรแกรมจึงไม่สามารถทราบได้ว่าเป็นใบหน้า

4.1.4 ทดสอบฟังก์ชั่นการทำงานของโปรแกรม

หลังจากได้ผลการตรวจจับแหละตำแหน่งใบหน้าแล้วเราจะนำค่าที่ได้ไปใส่ในฟังก์ชั่นการทำงานที่ได้เขียนไว้เพื่อควบคุมกล้องต่อไป



รูปที่ 4.3 แสดงตำแหน่งของใบหน้า



รูปที่ 4.4 แสดงตำแหน่งของใบหน้า

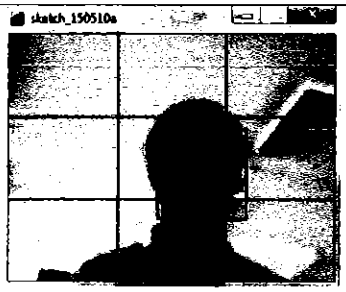
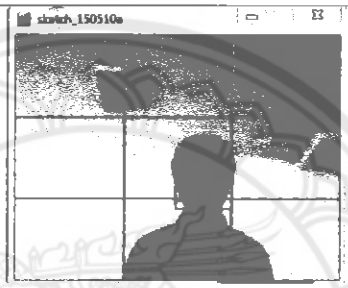
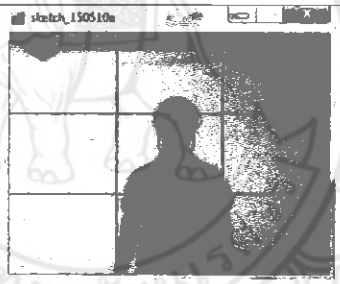
ผลการทดลอง เมื่อพบว่าใบหน้าไม่ได้อยู่ที่ตำแหน่งศูนย์ของภาพกลางรูปที่ 4.3 กล้องจะเคลื่อนไหวทำให้ใบหน้ามาอยู่ที่ตำแหน่งศูนย์กลางของภาพรูปที่ 4.4

4.2 ทดสอบขอบเขตของการใช้งาน

4.2.1 ระยะที่กล้องสามารถตรวจจับใบหน้า

การทดสอบนี้เป็นการทดสอบหาระยะการจับใบหน้าที่เหมาะสม โดยการจับภาพที่บริเวณลานกว้างโดยมีคนที่ถูกตรวจ 1 คนตามระยะที่กำหนดไว้ตั้งแต่ 1- 4 เมตร

ตารางที่ 4.2 ผลการทดลองตรวจจับและหมุนตามใบหน้าในระยะทางต่างๆ

ลำดับที่	ระยะ	ผลการตรวจจับใบหน้า	สถานะกล้องเมื่อมีการเคลื่อนไหว
1	1 เมตร		ติดตาม
2	2 เมตร		ติดตาม
3	3 เมตร		ติดตาม
4	4 เมตร		ไม่ติดตามเพราะไม่สามารถประมวลผลใบหน้าได้

จากตารางที่ 4.2 หลังจากการทดสอบหาระยะห่างของกล้องกับใบหน้าตามที่บริเวณลานกว้างสังเกตเมื่อใบหน้าอยู่ไกลขึ้นจะทำให้ความระเอียดของใบหน้าลดลงโปรแกรมไม่สามารถตรวจพบใบหน้าได้และกล้องจึงไม่สามารถเคลื่อนที่ติดตามใบหน้าได้

4.2.2 ทดสอบความเร็วใบหน้าที่เคลื่อนไหวในแนวราบ

เป็นการทดสอบว่ากล้องสามารถทำงานได้หรือไม่ในสถานะของใบหน้าที่เคลื่อนที่ด้วยความเร็วต่างๆจากทางด้านขวาไปด้านซ้ายด้วยระยะทาง 2 เมตรและบุคคลอยู่ห่างจากกล้อง 2 เมตร

ตารางที่ 4.3 ทดสอบความเร็วที่กล้องสามารถทำงานได้ในแนวราบ

ลำดับ	ระยะการเดิน	เวลา	การตรวจใบหน้า	ความเร็ว
1	2 เมตร	9.0 วินาที	ได้	0.22 เมตรต่อวินาที
2	2 เมตร	7.0 วินาที	ได้	0.28 เมตรต่อวินาที
3	2 เมตร	5.7 วินาที	ได้	0.35 เมตรต่อวินาที
4	2 เมตร	5.4 วินาที	ได้	0.37 เมตรต่อวินาที
5	2 เมตร	5.2 วินาที	ได้	0.38 เมตรต่อวินาที
6	2 เมตร	4.9 วินาที	ไม่ได้เพราะประมวลผลไม่ทัน	0.40 เมตรต่อวินาที
7	2 เมตร	4.5 วินาที	ไม่ได้เพราะประมวลผลไม่ทัน	0.44 เมตรต่อวินาที

จากตารางที่ 4.3 สามารถวิเคราะห์ผลได้ว่า ความเร็วในการเคลื่อนที่ของใบหน้านั้น มีผลต่อการตรวจจับใบหน้าเพราะ ถ้าวัตถุเคลื่อนที่เร็วเกินไป โปรแกรมยังสามารถประมวลผลตรวจจับใบหน้าได้ แต่ในส่วนของคำสั่งที่ใช้ควบคุมมอเตอร์นั้น ต้องอาศัยเวลาช่วงเวลาประมวลเล็กน้อยในการทำงาน ทำให้กล้องนั้นไม่สามารถหมุนติดตามใบหน้าได้ที่เคลื่อนที่เร็วเกินไปได้

4.2.3 ทดสอบความเร็วใบหน้าที่เคลื่อนไหวในแนวดิ่ง

เป็นการทดสอบว่ากล้องสามารถทำงานได้หรือไม่ในสาถานะของใบหน้าที่เคลื่อนที่ด้วยความเร็วต่างๆจากทางด้านล่างขึ้นด้านบนของกล้องด้วยการเดินจากยะระห่างจากกล้อง 3 เมตรเข้ามาหากล้องจนเหลือระยะ 1 เมตร

ตารางที่ 4.4 ทดสอบความเร็วที่กล้องสามารถทำงานได้ในแนวดิ่ง

ลำดับ	ระยะการเดิน	เวลา	การตรวจจับใบหน้า	ความเร็ว
1	2 เมตร	5.5 วินาที	ได้	0.36 เมตรต่อวินาที
2	2 เมตร	5.2 วินาที	ได้	0.38 เมตรต่อวินาที
3	2 เมตร	4.8 วินาที	ได้	0.41 เมตรต่อวินาที
4	2 เมตร	4.6 วินาที	ได้	0.43 เมตรต่อวินาที
5	2 เมตร	4.1 วินาที	ได้	0.48 เมตรต่อวินาที
6	2 เมตร	3.7 วินาที	ได้	0.54 เมตรต่อวินาที
7	2 เมตร	3.7 วินาที	ได้	0.54 เมตรต่อวินาที
8	2 เมตร	2.5 วินาที	ได้	0.80 เมตรต่อวินาที
9	2 เมตร	2.3 วินาที	ไม่ได้เพราะประมวลผลไม่ทัน	0.86 เมตรต่อวินาที
10	2 เมตร	2.2 วินาที	ไม่ได้เพราะประมวลผลไม่ทัน	0.90 เมตรต่อวินาที

จากตารางที่ 4.4 สามารถวิเคราะห์ผลได้ว่า ความเร็วในการเคลื่อนที่ของใบหน้านั้น มีผลต่อการตรวจจับใบหน้าเพราะ ถ้าวัตถุเคลื่อนที่เร็วเกินไป โปรแกรมยังสามารถประมวลผลตรวจจับใบหน้าได้ แต่ในส่วนของคำสั่งที่ใช้ควบคุมมอเตอร์นั้น ต้องอาศัยเวลาช่วงเวลาประมวลเล็กน้อยในการทำงาน ทำให้กล้องนั้นไม่สามารถหมุนติดตามใบหน้าได้ที่เคลื่อนที่เร็วเกินไปได้

บทที่ 5

สรุปและข้อเสนอแนะ

ในการทำโครงงาน กล้องเคลื่อนที่หมุนตามการตรวจจับใบหน้าโดยใช้กล้องเว็บแคมครั้งนี้ เป็นการนำเอาทฤษฎีทางการประมวลผลภาพดิจิตอลมาประยุกต์ใช้กับกล้องเว็บแคมที่มีราคาถูกเพื่อตรวจจับใบหน้า และใช้ไมโครคอนโทรลเลอร์เพื่อควบคุมมอเตอร์ทำให้สามารถควบคุมกล้องเว็บแคมได้อีกด้วย

เมื่อทำการทดลองแล้วทำให้ทราบถึงปัญหาบางประการ ทั้งในระหว่างการพัฒนาและการทดลองใช้งานโดยสามารถสรุปได้เป็นหัวข้อได้ดังต่อไปนี้

5.1 สรุปผลการทดลองโครงงาน

สามารถเขียนโปรแกรมตรวจจับใบหน้าและเคลื่อนที่หมุนตามใบหน้าได้ถือว่าโปรแกรมนี้เป็นประโยชน์ต่อการใช้งานในด้านการประยุกต์ใช้ในการถ่ายทอดวิดีโอที่ต้องการโฟกัสให้ความสำคัญของบุคคลหนึ่งได้ เช่น อาจารย์ที่กำลังสอน หรือ บุคคลที่กำลังนำเสนอข้อมูล

5.2 ข้อเสนอแนะ

5.2.1 จากการทดลองกล้องที่ใช้เป็นกล้องเว็บแคมที่มีความละเอียดต่ำ เมื่อวัตถุอยู่ไกลจะทำให้ภาพที่ได้มาไม่มีความชัดเจน เมื่อนำภาพมาประมวลผลจะไม่สามารถตรวจพบใบหน้า สิ่งที่จำเป็นต้องการพัฒนาคือกล้องเว็บแคมที่มีประสิทธิภาพสูง

5.2.2 ในการทดลองจำเป็นต้องใช้กับสถานที่ที่เหมาะสม คือ มีแสงสว่างมากเพียงพอ และจะต้องไม่มีหน้าของบุคคลที่สองเข้ามาเพิ่ม

5.2.3 เมื่อใบหน้าอยู่ใกล้เกินไปทำให้ใบหน้ามีขนาดใหญ่ จึงทำให้โปรแกรมอ่านค่าของใบหน้าเข้าขอบเขตในการควบคุมกล้องหลายขอบเขต จะทำให้โปรแกรมสับสน ดังนั้นจึงต้องมีระห่างในการใช้งานที่เหมาะสม

5.2.4 การเชื่อมต่อ OpenCV เข้ากับ Processing ทำได้ยากมีขั้นตอนที่ละเอียดอ่อน เช่น ต้องใช้รุ่นที่สามารถเข้ากันได้ ต้องมีโปรแกรมช่วยในการเชื่อมต่อ ต้องทำตามขั้นตอนการเชื่อมต่อทุกขั้นตอนอย่างเคร่งครัด ดังนั้นผู้ศึกษาจึงต้องหาข้อมูลเพิ่มเติม

5.2.5 การพัฒนาโปรแกรมตรวจจับใบหน้าผู้พัฒนาจำเป็นต้องมีความรู้ด้าน Image Processing และความถนัดทางภาษาที่ใช้เขียน ในส่วนของการควบคุมมอเตอร์นั้นจำเป็นต้องศึกษาเกี่ยวกับการส่งสัญญาณพัลส์ ของไมโครคอนโทรลเลอร์

เอกสารอ้างอิง

[1] Anon kaewart. **Java programming language** สืบค้นเมื่อ 18 เมษายน 2558, จาก https://aywkornayw.wordpress.com/

[2] Gary R. Bradski, Adrian Kaeble. **Learning OpenCV Computer Vision with the OpenCV Library,USA.** สืบค้นเมื่อ 24 มีนาคม 2558, จาก www.cse.iitk.ac.in/users/vision/.../OReilly%20Learning%20OpenCV.pdf

[3] โซฟีนา ยะขุ. **Face Localization, Tracking and Identification** สืบค้นเมื่อ 24 มีนาคม 2558, จาก http://fivedots.coe.psu.ac.th/~kom/wp-content/uploads/2009/07/48101671.pdf

[4] P. Viola, M. Jones. **Fast Multi-view FaceDetection, Technical Report TR-2003-96.** สืบค้นเมื่อ 30 เมษายน 2558

[5] กิตติศักดิ์ ผาภู, ปิยะวัฒน์ ภูมิศาสตร์.โครงการกล้องหมุนตามการเคลื่อนที่, คณะวิศวกรรมศาสตร์สาขา วิศวกรรมไฟฟ้า, มหาวิทยาลัยขอนแก่น, **2550.** สืบค้นเมื่อ 28 มีนาคม 2558, จาก https://app.enit.kku.ac.th/mis/administrator/doc.../20100308170015.pdf

[6] Ben Fry, Casey Reas. **Processing Handbook, 2nd Edition** สืบค้นเมื่อ 30 มีนาคม 2558, จาก https://processing.org/

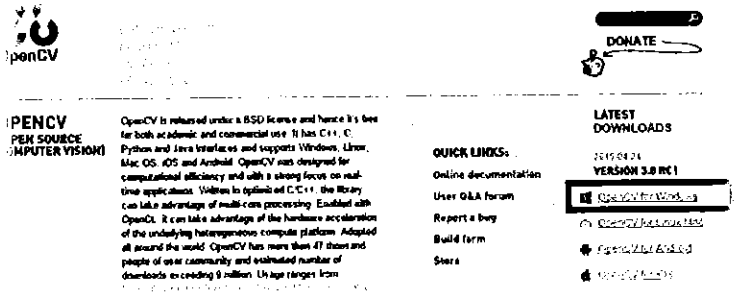
[7] รุสลี สุทธวีร์,กูล วิไลพร แซ่ลี้. **Face Detection based-on Haar-like Features** สืบค้นเมื่อ 19 เมษายน 2558, จาก http://ejournals.swu.ac.th/index.php/SwuENGj/article/viewFile/2306/2348

[8] น้ำทิพย์ รัตนาวงษ์ไชยา, สุมาลี โสน้ำเที่ยง, ธรรมนูญ ดิษเจริญ. รูปแบบในการส่งข้อมูลแบบอนุกรม สืบค้นเมื่อ 26 มีนาคม 2558, จาก http://www.il.mahidol.ac.th/e-media/computer/network/net_datacom.3htm

ภาคผนวก

1.การใช้งาน OpenCV เพื่อการประมวลผลภาพ

OpenCV เป็น library ซึ่งสามารถดาวน์โหลดได้ที่นี่ http://opencv.org/ โดยไม่เสียบริการ

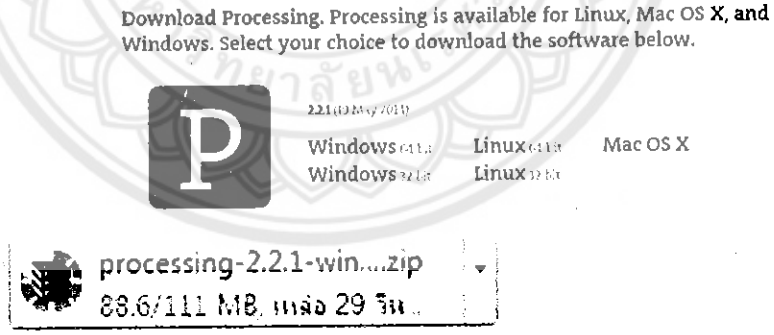


รูปที่ 1.1 แสดงส่วนของการดาวน์โหลด OpenCV

หลังจากติดตั้งเสร็จแล้ว การเรียกใช้งาน OpenCV จำเป็นจะต้องมีโปรแกรมสำหรับเขียนแบบ Opensource ยกตัวอย่างเช่น Python, Visual C++, Processing เป็นต้น

2. การใช้งาน โปรแกรมภาษา Processing

Processing สามารถดาวน์โหลดได้จาก https://processing.org/download/?processing



รูปที่ 2.1 แสดงส่วนของการดาวน์โหลด Processing

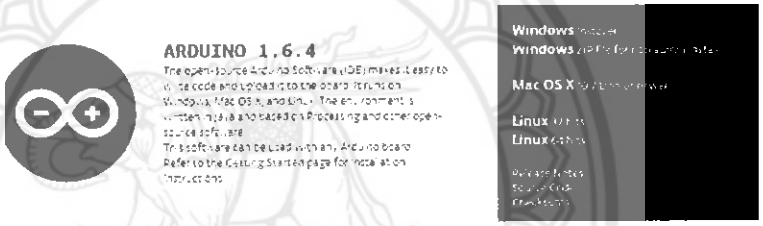
3. การเชื่อมต่อระหว่าง Processing และ OpenCV สำหรับ Window 7

การเชื่อมต่อนี้จำเป็นจะต้องมีตัวช่วยในการเชื่อมต่อที่เรียกว่า JavaCvPro ภายใน JavaCvPro จะประกอบไปด้วย Javacv และ Javacpp

สามารถดาวน์โหลดได้จาก http://www.mon-club-elec.fr/mes_downloads/javacvPro-0.3.zip แตกไฟล์และ copy ทั้งหมดลงไปใน Processing library directory: Processing /Mode/java/libraries

4. การใช้งานโปรแกรม Arduino

Arduino เป็นโปรแกรมที่ใช้สำหรับเขียน Code เพื่อใช้ควบคุม Board Arduino สามารถดาวน์โหลดได้จาก http://www.arduino.cc/en/Main/Software



รูปที่ 4.1 แสดงส่วนของการดาวน์โหลด Arduino

การใช้งาน Arduino UNO R3 เบื้องต้นต่อบอร์ด Arduino UNO R3 เข้ากับคอมพิวเตอร์ผ่านทาง USB port จากนั้นหลังจากนี้ Windows จะหาไดร์เวอร์แต่จะไม่พบ ให้เข้าไปที่ Device Manager แล้วคลิกขวาที่ Unknown แล้วเลือก Update Driver เลือกเมนู Browse my computer เลือก Brown ไปที่ไดร์เวอร์ Arduino โดยจะอยู่ที่ X:xxx\arduino-x.x.x\drivers และเอาเครื่องหมายที่ช่อง Include subfolders ออก เลือก Install this driver software anyware เมื่อสิ้นสุดขั้นตอนนี้ Windows ก็จะรู้จักบอร์ด Arduino UNO R3

5. ส่วนของ code processing ของโครงงาน

รูปที่ 5.1 ส่วนของ void setup ของภาษา Processing

โครงสร้างส่วนแรกของภาษา Processing คือการตั้งค่าต่างๆ

import เป็นการเรียก library ที่เกี่ยวข้องเข้ามามีส่วนในการใช้งาน

int เป็นการกำหนดตัวแปรทั้งหลายทั้งขนาดความกว้างความยาวของวีดีโอ

void setup() เป็นการตั้งค่าของ processing และสั่งทำงาน 1 ครั้ง

arduinoPort คือการสร้าง Serail ติดต่อกับ arduino โดยตั้งค่า baud rate ที่ 57600 เป็นค่า บิตต่อวินาที

opencv.loadCascade เป็นการเรียกใช้ library ในการตรวจจับใบหน้าของ OpenCV ที่พัฒนาโดย Paul Viola และ Michael Jones

```
void draw(){
opencv.loadImage(video);
image(video, 0, 0 );
  noFill();
 stroke(255,0,0);
 Rectangle[] faces = opencv.detect();
 println(faces);
if (faces.length ==0)
    {
if( t < 720){
println("There is no one here  >>>>>> Scan");
 if(millis() - time >= wait){
  println("tick");//if it is, do something
    time = millis();
   if (flip == false) {
 println("false");
arduinoPort.write("s");
arduinoPort.write("z");
i--;
time = millis();
}
else {
println("ture");
arduinoPort.write("x");
arduinoPort.write("y");
i++;
time = millis();
}
}
if (i>180) {
flip = false;
}
else if (i<0) {
flip = true;
}
}
}
    t=t+1;

if (faces.length ==1) {
t=720;
println("Center");
 }
```

รูปที่ 5.2 ส่วนของ void draw ของภาษา Processing

```
   for(int i=0;i<faces.length;i++){
 rect(faces[i].x,faces[i].y,faces[i].width,faces[i].height);

if(faces[i].x<107){
if(faces[i].x<50){
println("Much Left");
arduinoPort.write("a");
delay(20);
}
else{
println("Left");
arduinoPort.write("b");
 delay(20);
}
}
if(faces[i].x+faces[i].width>213){
  if(faces[i].x+faces[i].width>270){
println("Very Right");
arduinoPort.write("c");
delay(20);
}
else
{
 println("Right");
arduinoPort.write("d");
delay(20);
}
}
if(faces[i].y+faces[i].height>190){
println("Bottom");
arduinoPort.write("f");
delay(20);
}
if(faces[i].y<60){
println("Top");
arduinoPort.write("e");
delay(20);
}
}

line(0,80,320,80);
line(0,160,320,160);
line(107,240,107,0);
line(213,0,213,240);
}
```

รูปที่ 5.3 ส่วนของ void draw ของภาษา Processing

โครงสร้างของคำสั่ง Void draw นี้เป็นการทำงานแบบวนลูปจนกว่าจะมีคำสั่งให้หยุดการทำงานโดยภายในนี้จะกำหนดการรับค่าจากการ detect ใบหน้าและเก็บค่าเป็นตัวแปร array และนำค่าที่ได้มาเปรียบเทียบกับคำสั่ง if , else ที่ภายในคำสั่ง if else นี้จะเป็นตัวกำหนดว่าจะส่งคำสั่งให้ arduino ว่าอย่างไร และคำสั่ง line คือการวาดเส้นแบ่งภายในวีดีโอเป็น 9 ช่อง

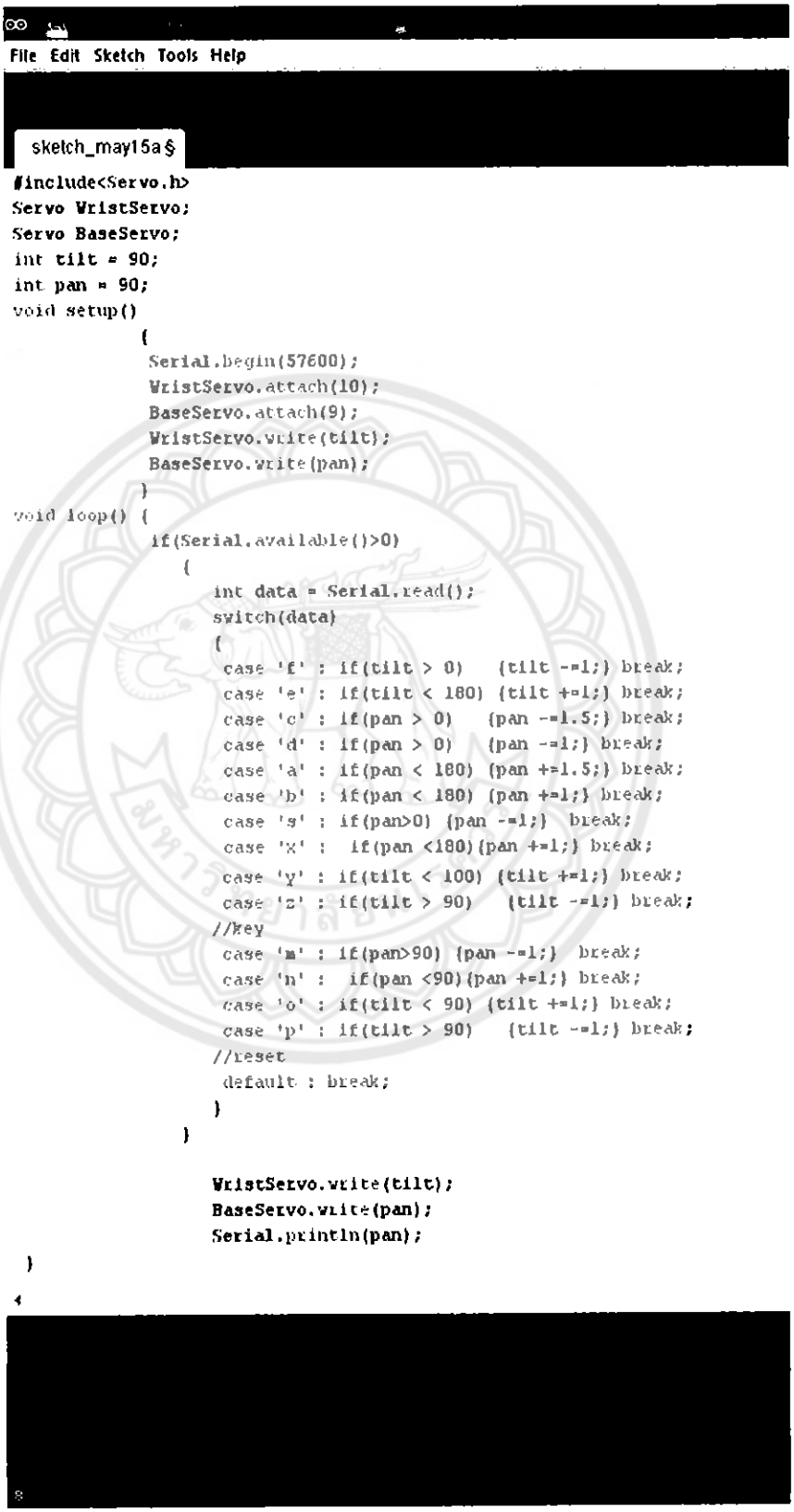
```
void captureEvent(Capture c)
 {
 c.read();
}
void keyPressed() {
 for(int o =1;o<3;o++){
 println("RESET");
arduinoPort.write("m");
arduinoPort.write("n");
arduinoPort.write("p");
arduinoPort.write("o");
 return ;
 }
}
```

133

รูปที่ 5.4 แสดงส่วนของ void keyPressed

void keyPressed() เป็นการทำงานเสริมของโปรแกรมที่จะนำมาใช้ทำงานโดยการกด Key ตัวใดตัวหนึ่งของคีย์บอร์ดเพื่อที่จะสั่งให้ Processing ทำงานในคำสั่ง Void keyPressed() แทนการทำงานภายใน Void draw()

6. การเขียน Sketch Arduino เพื่อสั่งให้เซอร์มอเตอร์ทำงาน



File Edit Sketch Tools Help

sketch_may15a§

```
#include<Servo.h>
Servo WristServo;
Servo BaseServo;
int tilt = 90;
int pan = 90;
void setup()
            {
             Serial.begin(57600);
             WristServo.attach(10);
             BaseServo.attach(9);
             WristServo.write(tilt);
             BaseServo.write(pan);
            }
void loop() {
             if(Serial.available()>0)
                {
                   int data = Serial.read();
                   switch(data)
                   {
                    case 'f' : if(tilt > 0)   {tilt -=1;} break;
                    case 'e' : if(tilt < 180) {tilt +=1;} break;
                    case 'c' : if(pan > 0)    {pan -=1.5;} break;
                    case 'd' : if(pan > 0)    {pan -=1;} break;
                    case 'a' : if(pan < 180)  {pan +=1.5;} break;
                    case 'b' : if(pan < 180)  {pan +=1;} break;
                    case 's' : if(pan>0)  {pan -=1;}  break;
                    case 'x' :  if(pan <180){pan +=1;} break;
                    case 'y' : if(tilt < 100) {tilt +=1;} break;
                    case 'z' : if(tilt > 90)   {tilt -=1;} break;
                   //key
                    case 'm' : if(pan>90) {pan -=1;}  break;
                    case 'n' :  if(pan <90){pan +=1;} break;
                    case 'o' : if(tilt < 90) {tilt +=1;} break;
                    case 'p' : if(tilt > 90)   {tilt -=1;} break;
                   //reset
                    default : break;
                   }
                }

                   WristServo.write(tilt);
                   BaseServo.write(pan);
                   Serial.println(pan);
 }
```

รูปที่ 6.1 แสดงตัวอย่างคำสั่งที่ใช้ควบคุมเซอร์โวมอเตอร์

#include<Servo.h> คือ เป็นการอ้างอิงถึงไฟล์ที่อยู่ใน include path ที่เราตั้งค่าให้ compiler ไปหา (เรียกใช้ library ของ เซอร์โวมอเตอร์)

Servo WristServo; คือ แทนการคำว่า Servo ในฟังก์ชั่นต่างๆ เป็น WristServo เพื่อง่ายต่อความเข้าใจ เช่น เรียกใช้ฟังก์แบบเดิม Servo.attach(pin) ก็สามารถใช้ WristServo.attach(pin) แทนได้เลย

Servo BaseServo; คือ แทนการคำว่า Servo ในฟังก์ชั่นต่างๆ เป็น BaseServo เพื่อง่ายต่อความเข้าใจ เช่น เรียกใช้ฟังก์แบบเดิม Servo.attach(pin) ก็สามารถใช้ BaseServo.attach(pin) แทนได้เลย

int tilt = 90; ประกาศตัวแปรชนิดจำนวนเต็ม(integer) ชื่อ tilt มีค่าเท่ากับ 90

int pan = 90; ประกาศตัวแปรชนิดจำนวนเต็ม(integer) ชื่อ pan มีค่าเท่ากับ 90

void setup(){}8คือ ทำงาน 1 ครั้ง ซึ่งเป็นฟังก์ชันสำหรับกำหนดค่าทั่วไป

Serial.begin(57600); คือ กำหนดอัตราบอดของการรับส่งข้อมูลเป็น 57600 บิตต่อวินาที (bits per second :bps baud rate)

WristServo.attach(10); คือ ฟังก์ชั่นที่ใช้ในการกำหนดขาสัญญาณที่เซอร์โวมอเตอร์ต่อกับ Arduino ที่ pin 10

BaseServo.attach(9); คือ ฟังก์ชั่นที่ใช้ในการกำหนดขาสัญญาณที่ เซอร์โวมอเตอร์ต่อกับ Arduino ที่ pin 9

WristServo.write(tilt); คือฟังก์ชั่นที่ใช้ควบคุมตำแหน่งที่ต้องการให้เซอร์โวมอเตอร์หมุนไปยังองศาที่กำหนดได้เลย คือ tilt(90) หมุนไปที่ 90 องศา

BaseServo.write(pan); คือฟังก์ชั่นที่ใช้ควบคุมตำแหน่งที่ต้องการให้ เซอร์โวมอเตอร์หมุนไปยังองศาที่กำหนดได้เลย คือ pan(90) หมุนไปที่ 90 องศา

if(Serial.available(0>0) คือ การตรวจสอบว่ามีการส่งค่าจาก Serial Port มาแล้วหรือยัง

int data = Serial.read(); คือ การประกาศตัวแปรชนิดจำนวนเต็ม (integer) ชื่อ data จะเป็นตัวแปรที่รับค่ามาจาก Processing โดยการอ่านค่าจาก Serial port case ไหน

case 'f' : if(tilt > 0) {tilt -=1;} break; คือ ค่าที่รับมามีค่าเท่ากับ f if กำหนดให้ ถ้าค่าตำแหน่งองศาของมอเตอร์มีค่ามากกว่า 0 มอเตอร์จะหมุนลดลง 1 องศา แล้วหยุดทำงาน

default : break; คือ ถ้าไม่มีคำสั่งตรงตาม case ไหน ให้หยุดทำงาน

Setial.println(pan) คือ ฟังก์ชั่นที่ใช้พิมพ์ค่าของpan ออกทางพอร์ตอนุกรม แต่ขึ้นบรรทัดใหม่ด้วย