ระบบติดตามวัตถุด้วยการประมวลผลภาพ

Object tracking system by image procesing

คนูต์ บุรารัตนวงศ์

Kanut Burarattanavong

เจนภูมิ เจนวิริยะกุล

Jenphum Jenviriyakul

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

ระบบติดตามวัตถุด้วยการประมวลผลภาพ

Object tracking system by image processing

โดย

คณุตม์ บุรารัตนวงศ์

เจนภูมิ เจนวิริยะกุล

อาจารย์ที่ปรึกษา

รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบติดตามวัตถุด้วยการประมวลผลภาพ
Object tracking system by image processing

ผู้จัดทำ นาย คณุตม์ บุรารัตนวงศ์ รหัสนักศึกษา 53010164

นาย เจนภูมิ เจนวิริยะกุล รหัสนักศึกษา 53010256

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบติดตามวัตถุด้วยการประมวลผลภาพ
นักศึกษา	นายคณุตม์ บุรารัตนวงศ์ รหัสนักศึกษา 53010164
	นายเจนภูมิ เจนวิริยะกุล รหัสนักศึกษา 53010256
ปริญญา	วิศวกรรมศาสตรบิณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2556
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

บทคัดย่อ

รายงานฉบับนี้อธิบายถึงหลักการทางทฤษฎีเกี่ยวกับการประมวลผลภาพโดยใช้การเขียนโปรแกรม OpenCV ซึ่งนำมาประยุกต์ใช้เป็นเครื่องติดตามวัตถุ โดยใช้ไมโครคอนโทรลเลอร์ส่งสัญญาณไปยังมอเตอร์เพื่อควบคุมกล้องให้ติดตามวัตถุที่กำลังเคลื่อนที่อยู่ โดยการทำงานแบ่งเป็นหลักๆได้สามส่วน ส่วนของการรับภาพโดยใช้กล้อง ส่วนของการประมวลผล และส่วนของอุปกรณ์ที่ใช้ในการติดตามวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	Object tracking system by image processing
Student	Mr. Kanut Burarattanavong Student ID 53010164
	Mr. Jenphum Jenviriyakul Student ID 53010256
Degree	Bachelor of Engineering
Program	Electronis Engineering
Year	2556
Thesis Advisor	Assoc. Prof. Dr.Surapan Airphaiboon

Abstract

This project describes the theory of digital image processing by using OpenCV which is applied for tracking object. Start by using microcontroller to sending all the signal to motor to control the camera while it is tracking the moving object. This project consists of three main parts,capturing image,processing,and tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้จะไม่สามารถสำเร็จลุล่วงได้ถ้าขาดการร่วมแรงร่วมใจของคนในกลุ่มที่ช่วยกันทำโครงงานนี้ขึ้น ประสบการณ์ที่ได้จะนำไปต่อยอดพัฒนาให้เกิดประโยชน์สูงสุด ทั้งนี้ข้าพเจ้าขอขอบคุณท่านอาจารย์ที่ปรึกษา รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์ ที่คอยให้คำปรึกษาให้คำแนะนำวิธีทางแก้ปัญหาต่างๆเกี่ยวกับเรื่องการเขียนโค๊ดในไมโครคอนโทรลเลอร์และเอื้อเฟื้อสถานที่และอุปกรณ์ที่จำเป็นต้องใช้ในการทำโครงงาน ขอขอบคุณคุณพ่อคุณแม่ที่คอยให้กำลังใจและช่วยเหลือทางด้านค่าใช้จ่าย รวมทั้งพี่ๆน้องๆเพื่อนๆที่คอยให้คำแนะนำ ให้คำปรึกษา ให้กำลังใจ ช่วยเหลือทุกๆอย่างที่สามารถช่วยได้

สุดท้ายนี้หากโครงงานนี้มีความผิดพลาดประการใดก็ขออภัยใน ณ ที่นี้ และทางผู้จัดทำหวังว่า โครงงานนี้ อาจจะเป็นประโยชน์กับบุคคลที่ได้มาศึกษาโครงงานของพวกเราต่อไป

นายคณุตม์ บุรารัตนวงศ์
นายเจนภูมิ เจนวิริยะกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ.....I
Abstract.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....V
สารบัญรูป.....VI
บทที่ 1 บทนำ.....1
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3
2.1 ส่วนประกอบพื้นฐาน.....3
2.2 Image Processing.....3
2.3 OpenCV.....6
2.4 บอร์ดArduino.....7
2.5 Webcam Camera.....10
2.6 สเต็ปมอเตอร์.....11
2.7 บอร์ดDrive stepper motor.....16
บทที่ 3 หลักการทำงานและออกแบบ.....19
3.1 หลักการออกแบบ.....19
3.2 หลักการทำงานของระบบ.....19
บทที่ 4 การทดลองและผลการทดลอง.....21
4.1 ส่วนของฮาร์ดแวร์.....21
4.2 ส่วนของซอฟต์แวร์.....23
บทที่ 5 วิเคราะห์และสรุปผลการลอง.....26
หนังสืออ้างอิง.....27
ภาคผนวก.....28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางแสดงการกระตุ้นเฟสแบบต่างๆของสเต็ปปิ้งมอเตอร์.....14
ตารางแสดงผลการทดลอง.....25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1 บล็อกไดอะแกรม....1
รูปที่ 2.1 แสดงภาพที่ได้จากวิธี edge detection....4
รูปที่ 2.2 รูปบอร์ดอาร์ดูโน่(Arduino)....7
รูปที่ 2.3 รูปบอร์ดอาร์ดูโน่(Arduino)....8
รูปที่ 2.4 แสดง pin ของ Arduino Board....9
รูปที่ 2.5 กล้องเว็บแคม OKER รุ่น 088....10
รูปที่ 2.6 สเต็ปมอเตอร์แบบมีสาย 5 เส้น....11
รูปที่ 2.7 สเต็ปมอเตอร์แบบมีสาย 6 เส้น....11
รูปที่ 2.8 สเต็ปมอเตอร์หลายแบบไบโพลาร์....12
รูปที่ 2.9 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์....12
รูปที่ 2.10 ตารางแสดงการกระตุ้นเฟสแบบต่างๆของสเต็ปปิ้งมอเตอร์....14
รูปที่ 2.11 แสดงการต่อวงจรขับสเต็ปปิ้งมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรทรานซิสเตอร์....15
รูปที่ 2.12 การใช้มิเตอร์วัดค่าความต้านทาน....15
รูปที่ 2.13 แสดงการต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับ....16
รูปที่ 2.14 โครงสร้างของบอร์ดDriving Stepper Motor....17
รูปที่ 2.15 สัญลักษณ์และตารางแสดงการใช้งาน....18
รูปที่ 3.1 ขั้นตอนการทำงานของ Image Processing....19
รูปที่ 3.2 โค้ดแสดงการควบคุมกล้องด้วยบอร์ดอาร์ดูโน่....20
รูปที่ 4.1 กล้องWebcamและฐานตั้งกล้อง....21
รูปที่ 4.2 วงจร Driving Stepping Motor....21
รูปที่ 4.3 Stepping Motor....22
รูปที่ 4.4 บอร์ดอาร์ดูโน่....22
รูปที่ 4.5 ข้อมูลรับภาพเริ่มต้น....23
รูปที่ 4.6 HSV Track Bar....23
รูปที่ 4.7 Tracking Object....24
รูปที่ 4.8 แสดงพิกัดพร้อมส่งผ่านพอร์ตอนุกรม....24
รูปที่ 4.9 โปรแกรมอาร์ดูโน่....25
รูปที่ 4.10 ตารางแสดงผลการทดลอง....25

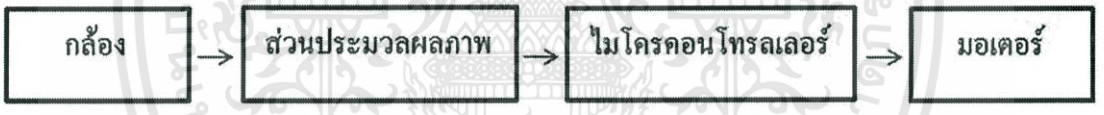
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันมีการประยุกต์ใช้กล้องวิดีโอในหลายๆด้าน เนื่องจากการประมวลผลภาพสามารถทำได้สะดวกโดยอาศัยโปรแกรมต่างๆ คณะผู้จัดทำจึงคิดทำโครงงานระบบติดตามวัตถุโดยใช้โปรแกรม VISUAL C++ ในการระบุสีหรือขนาด ของวัตถุ โดยต้องอาศัยความรู้ในการประมวลผลภาพสำหรับใช้ในการวิเคราะห์ภาพที่ได้จากกล้องวิดีโอ และความรู้ด้าน MICROCONTROLLER และวงจร DRIVE MOTOR สำหรับควบคุมการเคลื่อนที่ ซึ่งเป็นส่วนหนึ่งในการทำงาน เพื่อ ความสะดวกสบาย และความปลอดภัยในชีวิต และทรัพย์สินความสำคัญอาจจะอยู่ตรงภาษาในการใช้ ซึ่งก็คือ โปรแกรม VISUAL C++ ซึ่งเปรียบเสมือนสมอง วงจร DRIVE MOTOR เปรียบได้กับ เส้นสื่อนำประสาทที่จะส่งการสั่งการจากสมองไปยังส่วนต่างๆของร่างกาย และสุดท้าย ดีซีสเต็ปมอเตอร์ เปรียบเหมือนร่างกายซึ่งจะทำตามความต้องการของสมอง และกล้องเปรียบดั่งดวงตา ซึ่งรับสิ่งที่ต้องการจะทำเข้ามา สมองก็จะสั่งการ ไปที่ร่างกาย เมื่อทุกๆส่วนย่อยๆมารวมกันจึงเกิดเป็นระบบขึ้น

รายงานฉบับนี้ เป็นรายงานที่ใช้อธิบายการทำงานของการประมวลผลภาพด้วยกล้องติดตาม จึงจะขอกล่าวเฉพาะเนื้อหาที่ใช้ประกอบการทำงานเท่านั้น



รูปที่1 บล็อกไดอะแกรม

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อศึกษาการการประมวลผลภาพและนำไปประยุกต์ใช้ในการควบคุมระบบต่างๆ
- เพื่อฝึกการเขียนโปรแกรมและใช้งานโปรแกรม VISUAL STUDIO 2008
- เพื่อศึกษาการทำงานของ MICROCONTROLLER
- เพื่อได้รับความรู้เกี่ยวกับการติดต่อและส่งข้อมูลผ่านทางพอร์ตอนุกรม

1.3 ขอบเขตของโครงงาน

- กล้องสามารถบันทึกภาพผ่านโปรแกรมได้
- สามารถควบคุมกล้องไปในทิศทางที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการทำงาน

1) ศึกษาหลักการและวัตถุประสงค์ของโครงงาน
2) ศึกษาการใช้โปรแกรม Visual Studio 2008 และ Arduino
3) ศึกษาคุณสมบัติและหลักการทำงานของอุปกรณ์และโค้ดที่ใช้ในโครงงาน
4) ทดสอบการทำงานของการติดตามวัตถุด้วยกล้องโดยรันผ่านโปรแกรม
5) สรุปผลการทดลองและวิเคราะห์ปัญหา เพื่อเป็นแนวทางในการแก้ไข

1.5 ผลที่คาดว่าจะได้รับ

- สามารถนำวิธีการติดตามวัตถุไปประยุกต์ใช้ในโครงงานด้านอื่นๆได้ เช่น หุ่นยนต์ทำความสะอาด เครื่องแยกความแตกต่างของชิ้นงาน
- สามารถนำโครงงานไปประยุกต์ใช้ในระบบต่างๆได้เช่น ระบบรักษาความปลอดภัย

1.6 รายละเอียดของเนื้อหาในรายงาน

ในรายงานฉบับนี้ แสดงถึงรายละเอียดของโครงงานที่ได้จัดทำขึ้นมา ซึ่งประกอบด้วยหัวข้อต่างๆ รวมทั้งหมด 5 บท แต่ละบทกล่าวถึงดังต่อไปนี้

<u>บทที่ 1</u> เป็นบทนำของรายงาน ได้กล่าวถึงลักษณะโดยรวมของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน ผลที่คาดว่าจะได้รับ และรายละเอียดของโครงงานโดยย่อ

<u>บทที่ 2</u> กล่าวถึงทฤษฎี และแนวความคิดที่ได้ศึกษา เพื่อเป็นข้อมูลที่ใช้ในการจัดทำโครงงาน พื้นฐานของการประมวลผลภาพด้วยกล้องติดตามว่าต้องใช้อุปกรณ์ใดในการทำงาน มีระบบการทำงานเป็นอย่างไร เมื่อผ่านวงจรต่างๆ แล้วผลที่ได้รับจะเป็นอย่างไร

<u>บทที่ 3</u> หลักการทำงานและออกแบบ

<u>บทที่ 4</u> แสดงถึงการทดลอง และผลการทดลอง

<u>บทที่ 5</u> สรุปผล วิเคราะห์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ระบบติดตามวัตถุด้วยการประมวลผลภาพ (Object tracking system by image processing) เป็นการประยุกต์ใช้กล้องเว็บแคมจากคอมพิวเตอร์เพื่อติดตามวัตถุที่มีการเคลื่อนไหว ซึ่งในชีวิตประจำวันแล้วสามารถนำเอาไปใช้ประโยชน์ได้ เพื่อใช้ในระบบรักษาความปลอดภัย เป็นหุ่นยนต์เพื่อติดตามเป้าหมาย และเป็นตัวอย่างในการพัฒนาต่อไปในรูปแบบต่างๆในอนาคตได้

2.1 ส่วนประกอบพื้นฐาน

ระบบติดตามเป้าหมาย หมายถึง ระบบติดตามเป้าหมายสามารถแบ่งได้เป็นของสำคัญ ๆ คือ คำว่าระบบ+ติดตามเป้าหมาย โดยจะให้ความหมายในแต่ละตัวคือ

- ระบบ หมายถึง หน่วยผลรวมของหน่วยย่อยซึ่งทำงานเป็นอิสระต่อกัน แต่มีปฏิสัมพันธ์กันเพื่อช่วยให้งานนั้นบรรลุวัตถุประสงค์ที่ตั้งไว้
- ติดตามเป้าหมาย หมายถึงการเคลื่อนที่จามสิ่งที่เราสนใจ ซึ่งอาจจะเป็นการติดตามได้ในหลายรูปแบบ แล้วแต่โอกาสเรียกใช้งาน

เมื่อนำมารวมกัน คำว่าระบบติดตามเป้าหมาย หมายถึง หน่วยย่อยซึ่งทำงานเป็นอิสระต่อกัน และมีการจัดการตัวเองในการติดตามในสิ่งที่เราสนใจ

สเต็ปมอเตอร์ หมายถึง อุปกรณ์ทางไฟฟ้า ซึ่งใช้พลังงานทางไฟฟ้าแปลงเป็นพลังงานกลโดยใช้ไฟฟ้ากระแสตรง

ไมโครคอนโทรลเลอร์ หมายถึง หน่วยควบคุมและประมวลผลขนาดเล็ก โดยใช้หลักของโปรแกรมในการทำงาน

2.2 Image Processing

รูปภาพที่เราเห็นกันอยู่ไม่ว่าจะเป็นภาพที่ถ่ายโดยใช้กล้องธรรมดาหรือแบบดิจิตอล ถ้าเรามองกันในแบบของคอมพิวเตอร์ มันก็คือ จุดสีหลายๆ จุดที่นำมาเรียงต่อๆ กันจนสามารถบอกได้ว่าเรียงกันเป็นรูปอะไร เนื้อหาของรูปภาพเป็นอย่างไร การมองเพื่อทำความเข้าใจรูปภาพหนึ่งๆไม่ว่าจะเป็นภาพถ่าย หรือภาพที่เป็นแบบดิจิตอลในคอมพิวเตอร์ก็ตาม ในมุมมองของมนุษย์กับรูปภาพ หรือมุมมองของคอมพิวเตอร์กับรูปภาพเป็นคนละมุมกันและแตกต่างกันอย่างสิ้นเชิง มนุษย์สามารถเข้าใจถึงเนื้อหาของภาพได้ว่าภาพที่ปรากฏนั้นให้ความพึงพอใจ ความน่าสนใจมากน้อยแค่ไหน และภาพนี้บอกอะไร สามารถสื่อถึงความรู้สึกอะไรบางอย่างได้หรือไม่ และอีกหลาย ๆ ความรู้สึกที่ได้จากการมองภาพ

แต่เมื่อรูปภาพถูกนำมาทำเป็นภาพในคอมพิวเตอร์ มันจะรู้และเข้าใจภาพเป็นเพียงแค่เป็นจุดสีหลาย ๆ จุดที่เรียงต่อกันในความสัมพันธ์ระหว่างจุดภาพที่เหมาะสม ภาพดิจิตอลถึงแม้จะเก็บอยู่ในรูปของไฟล์ในดิสก์ของคอมพิวเตอร์เองหรือแม้แต่เราจะนำเอาภาพสวยๆ มาเป็นวอลเปเปอร์พื้นหลังของ Desktop ใน Windows ก็ตาม มันก็ไม่อาจจะรู้และเข้าใจถึงเนื้อหาของภาพที่ปรากฏนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกเว้นมนุษย์ที่เป็นผู้ใช้คอมพิวเตอร์นั้น ๆ จะเป็นผู้เห็นภาพๆ นั้นจึงจะบอกได้ว่านี่คือภาพที่สวยหรือไม่สวย

2.2.1 Feature Extraction

เป็นการดึงเอาลักษณะเฉพาะของแต่ละรูปภาพออกมา เป็นเวกเตอร์เพื่อนำไปใช้ในการฝึกฝนระบบและทดสอบระบบ บางครั้งการใช้สีเพื่อทำการแยกแยะวัตถุนั้น มักไม่มีความทนทานเพียงพอต่อการเปลี่ยนแปลงที่เกิดขึ้น เช่นความเปลี่ยนแปลงของสภาพแสง มุมมองของภาพ และการบดบังจากวัตถุอื่นๆก็สามารถทำให้การแยกแยะวัตถุโดยใช้สีผิดพลาดได้ การดึงจุดเด่นในภาพออกมา และทำการ Matching กับจุดเด่นของภาพใน Database ที่ได้กำหนดไว้ จุดเด่นของภาพนั้นสามารถกำหนดได้จากหลายวิธีด้วยกัน เช่นการหาขอบภาพ หามุมในภาพจากความต่างของค่าสีใน pixel ที่พิจารณาและ pixel รอบข้าง เมื่อทำการแปลงภาพให้แสดงค่าจุดเด่นของภาพ เช่น edge value, corner value แล้ว ค่าจุดเด่นนั้นจะนำมาเทียบกับค่าจุดเด่นของภาพตัวอย่างที่เก็บไว้ใน database เพื่อหาค่าความเหมือนซึ่งจะได้นำไปใช้ในการแยกแยะต่อไป

2.2.2 Image Segmentation

เป็นกระบวนการแยกภาพออกเป็นส่วนย่อยๆ จะทำให้สามารถแยกภาพส่วนที่ต้องการออกจากส่วนอื่นๆ โดยการตัดแยกวัตถุนั้นสามารถแบ่งออกได้ดังนี้

2.2.2.1 Edge Detection

รูปที่ 2.1 แสดงภาพที่ได้จากวิธี edge detection

จากรูปที่ 2.1 การหาขอบภาพ (Edge Detection) คือ การตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว หรืออาจกล่าวได้ว่าเป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุ เราจะสามารถคำนวณหาพื้นที่(ขนาด) หรือรู้จำ ชนิดของวัตถุนั้นได้ อย่างไรก็ตามการหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เป็นเรื่องง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีคุณภาพต่ำ มีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วภาพ ขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง หากความต่างนี้มีค่ามาก ขอบภาพก็จะเห็นได้ชัด ถ้าความแตกต่างมีค่าน้อย ขอบภาพก็จะไม่ชัดเจน ซึ่งวิธีการหาขอบนั้นมีด้วยกัน ซึ่งวิธีการหาขอบนั้นมีด้วยหลายวิธี สามารถแบ่งได้เป็น 2 กลุ่มหลักคือ Gradient method และ Laplacian method สำหรับ Gradient method จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

เช่น Sobel, Roberts, Prewitt, Canny เป็นต้น ส่วน Laplacian method จะหาขอบโดยใช้อนุพันธ์อับดับ 2 โดยใช้จุดที่ค่า y เป็น 0 (Zerocrossing) ในภาพ เช่น Laplacian of Gussian และMarrs-Hildreth เป็นต้น

2.2.3 Matching

กระบวนการจับคู่ภาพที่เหมือนกัน(Matching)[8] เป็นเหมือนการสื่อสารกันระหว่างหว่างภาพและขั้นตอนการพัฒนาการจับคู่รูปภาพนั้นขึ้นอยู่กับการตอบสนองหลังจากการทำ Localization ถ้าหากผู้ใช้แสดงให้เห็นว่าการทำ localization นั้นไม่ถูกต้อง การเรียนรู้จะเกิดขึ้นโดยการใช้คุณสมบัติหรือลักษณะเด่นของภาพทั้งหมดที่มีอยู่สำทำ localization และกำหนดตำแหน่งโดยผู้ใช้ การเรียนรู้การจับคู่ภาพเป็นกระบวนการเรียนรู้ที่มีการสืบทอด คือ การสร้างคลังคำศัพท์หรือ dictionary และการรวบรวมข้อมูลสำหรับการจำแนกจะเพิ่มมากขึ้น คือ จำนวนของคำที่ได้จากการเรียนรู้จะมีเพิ่มมากขึ้นเพียงเวลาแค่ไม่นาน

การสร้าง Dictionary จะขึ้นอยู่กับการเพิ่มของ nearest neighbor classifier สำหรับคำหรือคุณสมบัติใหม่ที่ไม่มีอยู่ใน dictionary ถ้าถูกนำเข้ามาในระบบ แล้วมีการจำคู่ภาพและพบว่ามีส่วนที่คล้ายกับข้อมูลอื่นที่อยู่ใน dictionary ในขั้นตอนนี้หากระยะห่างระหว่างคำกับลักษณะเด่นของภาพอยู่ต่ำกว่า ค่าเกณฑ์ที่กำหนด จะทำให้มีการรู้จำคำนั้นเกิดขึ้น คือระบบสามารถระบุได้ว่าภาพรถที่มีการจับคู่เป็นภาพรถยี่ห้อหรือรุ่นไหน แต่ถ้าค่าระยะห่างสูงกว่าเกณฑ์ที่กำหนด(ภาพที่จับคู่กันมีส่วนที่เหมือนกันน้อยอาจไม่ใช่ภาพเดียวกัน) คำหรือชิ้นส่วนของภาพนั้น จะถูกเพิ่มเข้าไปเป็นคุณสมบัติและกำหนดเป็นค่าเริ่มต้นใหม่ของข้อมูลหรือ dictionary และนำไปใช้ในการ match ภาพในครั้งต่อไปที่มี ภาพแบบนี้เข้ามา

การหาจุดเด่นและการ Matching ภาพนั้นจะต้องใช้เวลานานขึ้นมาก และอาจมีประสิทธิภาพที่ไม่ดี ดังนั้นภาพที่จะนำมาหาจุดเด่นจึงต้องมีการ ปรับขนาดซึ่งอาจผ่านการแยกแยะรูปร่างมาก่อนแล้ว จะถูกนำมาแปลงขนาดให้เท่ากับขนาดของ template ที่เก็บไว้ใน database หลังจากนั้นจึงทำหมุนภาพให้มี orientation ที่สอดคล้องกับภาพใน template โดยอ้างอิงจากรูปร่างของรถยนต์เป็นหลัก

2.2.4 Scale Invariant Feature Transforms (SIFT)

Scale invariant feature transform หรือ SIFT [3] เป็นอัลกอริทึมหนึ่งในคอมพิวเตอร์วิชันสำหรับคำนวณหาจุดสนใจ (Key point) ในรูปภาพหนึ่ง ๆ และคำนวณหาคุณลักษณะของ Key point หนึ่งๆ ที่หาได้ SIFT คือการเอาจุดเด่นในรูปที่ไม่ขึ้นอยู่กับสเกล การกำหนดทิศทาง ตำแหน่ง มุมการมอง แสงสว่าง เงา ซึ่งจะทำให้สามารถนำมาใช้ในการเปรียบเทียบจุดเด่นในรูปอื่นๆได้ง่ายและถูกต้องแม่นยำมากยิ่งขึ้นอัลกอริทึมนี้ถูกคิดค้นโดย ศร.เดวิด โวลว์ (David Lowe) แห่งมหาวิทยาลัย British Columbia ในปี 1999 กาประยุกต์ใช้งาน SIFT มีหลากหลาย ดังเช่น การรู้จกวัตถุ (Object recognition), การสร้างแผนที่สำหรับนำทางหุ่นยนต์ (Robotic mapping and navigation, image stitching), การสร้างโมเดล 3 มิติของวัตถุหรือทัศนียภาพ (3D modeling of object and scenes)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปแล้ว Key point จะหมายถึงจุดพิกเซลในภาพที่มีการเปลี่ยนแปลงเชิงสองมิติ (Two -Dimensional) ของระดับความสว่าง (Pixel Intensity) รอบๆ Key point นั้นๆ

2.2.4.1 ส่วนประกอบของอัลกอริทึมในการหา SIFT Key point ในภาพหนึ่ง คือ

ส่วนของการตรวจหา Key point ออกมาจากภาพอินพุต (Key point Detection) ซึ่งในส่วนนี้เราจะได้ชุดของพิกัด x, y ของจุดที่จะเป็น Key point ซึ่งจะใช้ในการให้คำอธิบายของ Key point นั้นๆในขั้นตอนถัดไป

ส่วนของการให้คำอธิยายแก่ Key point หนึ่งๆ (Key point Description) ในส่วนนี้อัลกอริทึมจะคำนวณหาเวกเตอร์อธิยาย (Descriptor Vector) ซึ่งคำนวณมาจากค่าความสว่างของพิกเซลในอาณาบริเวณรอบๆ Keypoint เวกเตอร์อธิบายเหล่านี้จะใช้ในการเปรียนเทียบเพื่อวัตถุประสงค์ของการรู้จำอัตลักษณ์ (Identity) เมื่อมันไปปรากฏอยู่ในภาพอื่น

2.2.4.2 การนำ SIFT ไปประยุกต์ใช้ในการรู้จำวัตถุ

ทำการหา SIFT Key point และ Descriptor Vector จากภาพต้นแบบของวัตถุ (Template Image) ขั้นตอนนี้โดยทั่วไปนิยมเรียกว่า ขั้นตอนการเรียนรู้ (Learning Phase) จากขั้นตอนการทำ SIFT Key point เมื่อเราต้องการจะรู้จำวัตถุที่ปรากฏอยู่ในภาพอินพุตหนึ่ง ๆ ที่ไม่ใช้ต้นแบบ เราสามารถทำได้โดยหา Key point ในภาพอินพุตนั้นๆ ไปเปรียบเทียบ Key point หนึ่ง ๆ ใน โมเดลของวัตถุ โดยการเปรียบเทียบจะใช้การคำนวณหา Euclidean Distance ระหว่างเวกเตอร์อธิบายที่ประกอบกับ Key point โดยที่ถ้าค่าระยะทางยิ่งน้อยแสดงว่า Key point ทั้งสองที่นำมาเปรียบเทียบกันยิ่งเหมือนกัน

จากชุดของการจับคู่ (Matches) ระหว่าง Key point ในภาพอินพุตกับ Key point ของโมเดลของวัตถุ เราสามารถประยุกต์ใช้อัลกอริทึมที่ใช้ในการพิสูจน์ว่าเป็นวัตถุนั้นจริงได้

2.3 OpenCV (Open Source Computer Vision Library)

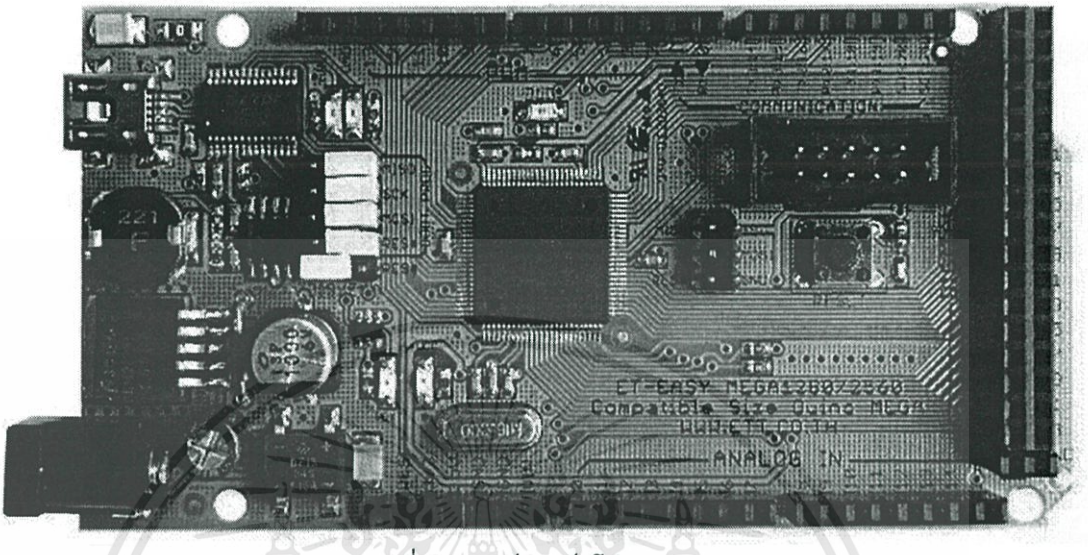
OpenCV หรือ Open Source Computer Vision Library [2] ใช้ในการประมวลผลภาพและงานทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) Library นี้ถูกพัฒนาขึ้นด้วยภาษา C และ C++ และยังมี interface ที่ไว้เชื่อมต่อกับ tool อื่นด้วยเช่น Python, Ruby, Mathlab เป็นต้น นอกจานี้ OpenCV เป็น library ที่สร้างขึ้นเพื่อให้ผู้ใช้หรือนักพัฒนาสามารถใช้ฟังก์ชันใน library มาพัฒนาชิ้นงานที่มีความซับซ้อนโดยใช้เวลาเพียงไม่นาน OpenCV ประกอบด้วย Data Structure และ Algorithm

- Data Structure ใช้เก็บข้อมูลต่าง ๆ อาทิ เช่น รูปภาพ เมทริกซ์ พิกัด
- Algorithm เพื่อการประมวลผลต่าง โดยเฉพาะการประมวลผลทางรูปภาพ

ข้อจำกัดของ OpenCV คือ สามารถใช้งานได้เฉพาะเครื่องคอมพิวเตอร์ที่ใช้หน่วยประมวลผล (CPU) จาก Intel Corporation แต่ข้อจำกัดนี้ทำให้เกิดจุดเด่นเช่นกัน กล่าวคือ การประมวลผลต่าง ๆ จะใช้ความสามารถของหน่วยประมวลผลอย่างเต็มที่ ทำให้โปรแกรมที่พัฒนาโดยการใช้ OpenCV นี้มีประสทธิภาพในการประมวลผลที่สูงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 บอร์ดอาร์ดูโน่(Arduino)

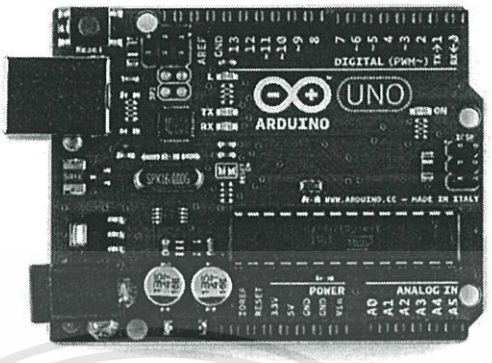


รูปที่ 2.2 บอร์ดอาร์ดูโน่

Arduino คือบอร์ดสำเร็จรูปชนิดหนึ่งที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไป ควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้คอมพิวเตอร์ทั่วไป โดยตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน

Arduino สามารถประยุกต์ทำเครื่องใช้อัจฉริยะ รับสัญญาณจากสวิทช์ หรือเซนเซอร์ และควบคุม หลอดไฟ มอเตอร์ หรืออุปกรณ์อื่นๆ ดังนั้นในการศึกษาและจัดทำโครงงานโดยเลือกใช้บอร์ด Arduino จะเป็นได้ทั้งแบบที่ทำงานได้อิสระ หรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ โดยที่บอร์ด Arduino ใช้ไมโครคอนโทรลเลอร์หลักเป็น AVR ตระกูล 8 บิต โดยถูกออกแบบให้มีการต่อใช้งานเป็นขา Digital I/O จำนวน 13 ขา และเป็นขารับสัญญาณ Analog จำนวน 6 ขา ซึ่งการกำหนดจำนวนขาแบบนี้ ได้กลายเป็นมาตรฐานของบอร์ด Arduino ไปแล้ว ถึงว่า Arduino เองจะถูกผลิตออกมาหลายแบบ แต่ที่เหมือนกัน ก็ยังคงเป็นขาที่เอาไว้ใช้งาน แต่แตกต่างกันตรงที่แต่ละรุ่น ที่ทำออกมามี memory ที่ไม่เท่ากันนั่นเอง (แต่ปัจจุบัน บอร์ด Arduino มีจำนวนขาให้ใช้ มากกว่าที่ได้กล่าวไว้แล้ว แตกต่างกันออกไปในแต่ละรุ่น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 บอร์ดอาร์ดูโน่

ส่วนภาษาในการเขียนโปรแกรมลงบน Arduino นั้นจะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) อื่นๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง ซึ่งจากการที่ได้ทำการศึกษาค้นคว้าทดลองการใช้งานภาษาซีของ Arduino มาในระยะเวลาหนึ่ง จะพบว่าในความเป็นจริงแล้ว Arduino นั้นไม่ใช่ C-Compiler โดยตรง แต่ Arduino จะมีลักษณะการทำงานเช่นเดียวกันกับ Text Editor เป็นฉากหน้าในการติดต่อสื่อสารกับผู้ใช้เท่านั้น ส่วนเบื้องหลังจริงๆนั้น Arduino จะไปเรียกใช้ตัวแปลภาษาซีและ Utility อื่นๆ ที่ใช้เป็นเครื่องมือพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล AVR อีกทีหนึ่ง โดย Arduino จะเลือกใช้ C-Compiler ของ “GNU AVRGCC Toolchain” ร่วมกับ Library Function ของ “avr-libc” ส่วน Utility ที่ใช้ในการ Upload Code ให้กับ AVR นั้นก็จะใช้ของ “AVRDude” ดังนั้นผู้ที่เขียนภาษาซีของ AVR เป็นอยู่แล้ว และต้องการประยุกต์ใช้งาน Arduino ให้ได้ประสิทธิภาพการท างานมากยิ่งขึ้นไปอีก ก็สามารถศึกษาข้อกำาหนด และหน้าที่ในการใช้งาน Library และคำสั่งอื่นๆที่บรรจุไว้ใน Library ต่างๆ ทั้งจากของ “GNU AVR-GCC Toolchain” และ “avr-libc” เพิ่มเติมอีก เพื่อใช้เป็นแนวทางในการปรับปรุงและประยุกต์ใช้งาน Arduino ในรูปแบบที่สลับซับซ้อนมากๆ ขึ้นไปได้อีก

- Chip และ IC ภายใน Arduino Board ที่สำคัญ

ATmega328P-PU ตัวบอร์ด Arduino สถาปัตยกรรมของ AVR ขนาด 8 บิต โดยในสถาปัตยกรรม AVR ซึ่งออกแบบโดย ATMEL เมื่อปี 1996 เป็นซีพียูแบบ RISC (Reduced Instruction Set Computer) มีสถาปัตยกรรมการต่อหน่วยความจำแบบ Harvard ซึ่งแยกหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลออกจากกันโดยเด็ดขาด โดยใช้หน่วยความจำแบบ Flash สาหรับเป็นหน่วยความจำโปรแกรม และใช้หน่วยความจำแบบ SRAM สาหรับหน่วยความจำข้อมูล และนอกจากนี้ยังมีหน่วยความจำแบบ EEPROM ซึ่งสามารถเก็บข้อมูลเอาไว้ได้โดยไม่จำเป็นต้องมีไฟเลี้ยงอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ Harvard จะแยกหน่วยความจาสาหรับเก็บข้อมูลออกจาก โปรแกรมอย่างชัดเจน สถาปัตยกรรม AVR และ MCS-51 จะใช้รูปแบบนี้ในการจัดการหน่วยความจา ส่วนสถาปัตยกรรมแบบ Von-neumann การตัดสินใจว่าจะเก็บโปรแกรมหรือข้อมูลจะแบ่งเก็บ อย่างไรจะทาได้อย่างอิสระ โดยขึ้นอยู่กับโปรแกรมเมอร์ หรืออาจจะเป็นระบบ ปฏิบัติการเป็นผู้ดาเนินการให้ลักษณะเด่นของสถาปัตยกรรม AVR คือ คาสั่งส่วนใหญ่สามารถทางาน ได้เสร็จภายใน 1 clock cycle ตัวซีพียู AVR ขนาด 8 บิตจะแบ่งออกเป็นประเภทการใช้งานได้ 5 กลุ่ม ได้แก่

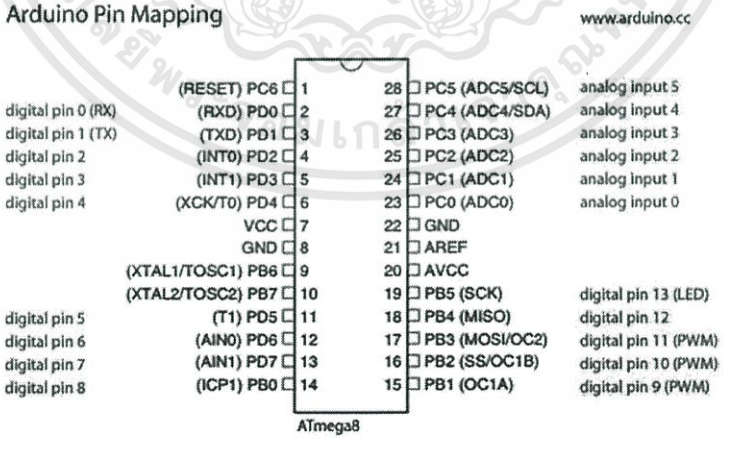
- tinyAVR เป็นซีพียูในรุ่นเล็ก ซึ่งต้องการความเล็กกะทัดรัดของวงจร โดยเหมาะกับระบบ ควบคุมขนาดเล็กๆ ที่ต้องการหน่วยความจาและวงจรสนับสนุนไม่มากนัก ซีพียูในรุ่นนี้จะมีราคาถูก กว่ากลุ่มอื่น

- megaAVR จะมีชื่ออีกอย่างว่า ATmega โดยมีวงจรสนับสนุนภายในเพิ่มเติมตลอดจนเพิ่ม ขนาดของหน่วยความจำให้ใช้งานมากกว่าตระกูล Tiny เหมาะกับงานควบคุมทั่วๆ ไป

- XMEGA เพิ่มความละเอียดของวงจร A/D จากปกติมีความละเอียด 10 บิตในรุ่นเล็กกว่า เป็น 12 บิต และวงจร DMA controller ซึ่งช่วยลดภาระของซีพียูในการควบคุมการรับส่งข้อมูล ระหว่าง
อุปกรณ์ I/O กับหน่วยความจำ

- FPSLIC (AVR core with FPGA) สาหรับงานที่ต้องการควบคุมที่ต้องการความยืดหยุ่นใน ขั้นตอนการออกแบบและพัฒนา โดยผู้ออกแบบสามารถออกแบบวงจรในระดับฮาร์ดแวร์เพิ่มเติมด้วย ภาษาบรรยายฮาร์ดแวร์ (HDL: Hardware Description Language) เช่น ภาษา VHDL หรือภาษา Verilog และให้วงจรที่ออกแบบทางานร่วมกับซีพียู AVR core

- Application Specific AVR เป็นซีพียูที่ออกแบบมาโดยเพิ่มวงจรควบคุมเฉพาะด้านเข้าไป ซึ่งไม่พบในซีพียูกลุ่มอื่นๆ เช่นวงจร USB controller หรือวงจร CAN bus เป็นต้น



รูปที่ 2.4 แสดง Pin ของ Arduino Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซีพียู AVR มีให้เลือกใช้งานหลายเบอร์ แต่ละเบอร์จะมีขนาด ราคา ความสามารถ และขนาดหน่วยความจาตลอดจนถึงวงจรสนับสนุนภายในที่แตกต่างกันออกไป ในโครงงานนี้จะเลือกใช้ซีพียูรุ่น ATmega328P ซึ่งมีคุณสมบัติดังนี้

- หน่วยความจาโปรแกรมแบบ FLASH ขนาด 32 กิโลไบต์
- หน่วยความจาข้อมูลแบบ SRAM ขนาด 2 กิโลไบต์
- หน่วยความจาข้อมูลแบบ EEPROM ขนาด 1 กิโลไบต์
- สนับสนุนการเชื่อมต่อแบบ I2C bus
- พอร์ตอินพุตเอาต์พุตจานวน 23 บิต
- วงจรสื่อสารอนุกรม
- วงจรนับ/จับเวลาขนาด 8 บิต จานวน 2 ตัว และน3586 .นาด 16 บิตจานวน 1 ตัว
- สนับสนุนช่องสัญญาณสาหรับสร้าง Pulse Width Modulation (PWM) จานวน 6 ช่อง
- วงจรแปลงอนาลอกเป็นดิจิตอลขนาด 10 บิตในตัวจานวน 8 ช่อง
- ทางานได้ตั้งแต่ย่านแรงดัน 1.8-5.5 Volts
- ความถี่ใช้งานสูงสุด 20 MHz

2.5 WEBCAM CAMERA

รูปที่ 2.5 กล้องเว็บแคม OKER รุ่น 088

กล้องเว็บแคม OKER รุ่น 088 รุ่นนี้มีความละเอียดสามารถปรับได้ตั้งแต่2 ล้านพิกเซล จนถึง 10 ล้านพิกเซลเลยทีเดียว อัตราการถ่าย VDO ให้ความละเอียดอยู่ที่ 30fps Interface ส่วนพอตร์เชื่อมต่อจะเป็นแบบ USB2.0

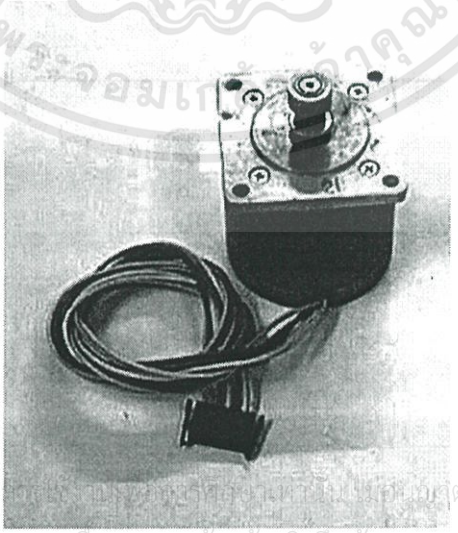
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 สเต็ปมอเตอร์ (STEPPING MOTOR)

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเครื่องด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1,1.5,1.8 หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์ลักษณะที่ นำมอเตอร์ไปใช้ จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER)ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเหล็ก ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTER เป็นต้น

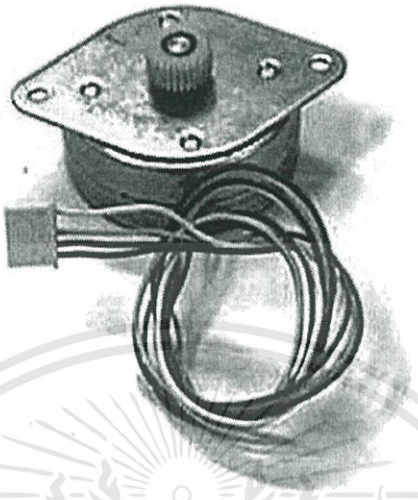
ชนิดและโครงสร้างของสเต็ปมอเตอร์

รูปที่ 2.6 สเต็ปมอเตอร์แบบมีสาย 5 เส้น

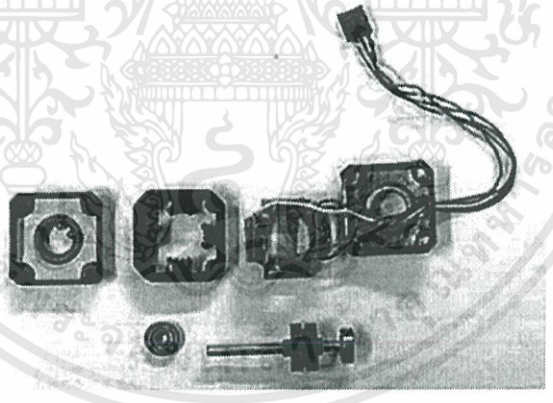


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 สเต็ปมอเตอร์แบบมีสาย 6 เส้น



รูปที่ 2.8 สเต็ปมอเตอร์หลายแบบไบโพลาร์



รูป 2.9 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็ปมอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

1. แบบแม่เหล็กถาวร(PERMANENT MAGNET_PM)

สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง ให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงฉุดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR)

สเต็ปมอเตอร์แบบVR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนท์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างขดกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม(HYBRID-H)

สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ

หลักการทำงานของ stepping motor

เชื่อว่านักอิเล็กทรอนิกส์ทุกท่านคงเคยได้ยินหรือได้เคยใช้งาน stepping motor กันมาบ้างแล้ว ชื่อของเจ้า stepping motor นั้นก็ค่อนข้างที่จะบ่งบอกถึงคุณสมบัติของตัวมันได้ดีอยู่แล้ว ว่าเป็น motor ที่สามารถเคลื่อนที่ในลักษณะที่เป็นทีละ step ซึ่งต่างจาก motor ทั่วไปที่การเคลื่อนที่ของมันจะเป็นแบบต่อเนื่อง และ stepping motor ยังสามารถควบคุมการทำงานที่ง่ายด้วยการต่อร่วมกับอุปกรณ์ภายนอกเพียงเล็กน้อย ดั้งนั้นเราจะเห็นว่า stepping motor ถูกนำไปใช้งานอย่างกว้างขวางไม่ว่าจะเป็นงานด้าน robot,แขนกล,plotter หรืองานใดๆที่ต้องใช้การเคลื่อนที่ในลักษณะของแบบ step การเรียนรู้หลักการทำงานของ stepping motor ก็ย่อมก่อให้เกิดประโยชน์แก่เราให้สามารถนำ stepping motor ไปประยุกต์ใช้งานได้ ลักษณะของ stepping motor ภายนอกก็จะประกอบไปด้วยสายไฟที่เราจะต้องป้อนสัญญาณ pluse เข้าไปควบคุมมัน ถ้าหาก steppping motor ของคุณมี 3 ขดลวด ก็จะได้ลักษณะการต่อดังรูป ซึ่งจะมีสายหนึ่งที่ต่อเป็น ground ร่วมของทั้ง 3 สายนั้น ลักษณะแบบนี้คือเป็นแบบ 3 เฟส โดยที่ stepping motor ที่ขายทั่วๆไปก็อาจจะมีหลายแบบเช่นแบบ 4 เฟส 5 เฟส ซึ่งเมื่อเราทราบเฟสแล้วสิ่งที่จะต้องทำต่อไปคือการหาว่าเฟสใดเป็นเฟส 1 ,2,3 ซึ่งที่ตัวของ stepping motor อาจจะมีบอกอยู่แล้ว หรือถ้าไม่มีวิธีง่ายๆก็คือการใช้ไฟขนาดที่ motor ตัวนั้นใช้ป้อนเข้าไปที่เฟสแต่ละเฟส ถ้าหากเราป้อนเป็นเฟส 1-2-3 แล้ว stepping motor จะต้องหมุนไปในทิศทางเดียวกันหรือถ้าหากป้อนเป็น 3-2-1 จะต้องหมุนกับทางกันซึ่งเป็นวิธีง่ายๆในการ check เฟสของ stepping motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของสเต็ปปิ้งมอเตอร์

ในการควบคุมสเต็ปปิ้งมอเตอร์เพื่อที่จะให้ทำการหมุน มีวิธีการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์ (Stator) ในแต่ละเฟสของสเต็ปปิ้งมอเตอร์ อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆเป็นสถานะลอจิก "1" และในการกระตุ้นเฟสของของสเต็ปปิ้งมีอยู่ด้วยกัน 2 แบบคือ

1 การกระตุ้นเฟส แบบฟูลสเต็ปมอเตอร์ (Full Step Motor) ยังสามารถแบ่งการกระตุ้นเฟสออกได้เป็นอีก 2 วิธีด้วยกันคือ

1.1 การกระตุ้นเฟสแบบฟูลสเต็ป 1 เฟส (Single-Phase Driver) หรือแบบเวฟ แสดงดังตารางรูป 6.15.ก จะเป็นการป้อนกระแสไฟให้กับขดลวด ของสเต็ปปิ้งมอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไป ดังนั้นกระแส ที่ไหลในขดลวด จะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเต็ปปิ้งมอเตอร์มีน้อย

1.2 การกระตุ้นเฟสแบบฟูลสเต็ป 2 เฟส (Two-Phase Driver) แสดงดังตารางรูป 6.15 ข เป็นการป้อนกระแสให้กับขดลวด 2 ขด ของสเต็ปปิ้งมอเตอร์พร้อมๆกันไป และจะกระตุ้นเรียงถัดกับไปเช่นเดียวกับแบบหนึ่งเฟส ดังนั้นการกระตุ้นแบบนี้จึงต้องใช้กำลังไฟมากขึ้น และจะทำให้มีแรงบิดของมอเตอร์มากกว่าการกระตุ้นแบบ 1 เฟส

2 การกระตุ้นเฟส แบบฮาลฟ์สเตป (Half Step Motor) หรือ one-two phase Driver คือ การกระตุ้นเฟสแบบ ฟูล สเต็ป 1 เฟส และ 2 เฟส เรียงลำดับกันไป แสดงดังตารางรูป 6.15.ค แรงบิดที่ได้จากการกระตุ้นเฟสแบบนี้จะมีเพิ่มมากขึ้น เพราะช่วงของสเต็ปมีระยะสั้นลง ในการกระตุ้นแบบนี้ เราจะต้องมีการกระตุ้นที่เฟสถึง 2 ครั้ง จึงจะได้ระยะของ สเต็ปเท่ากับการกระตุ้นเพียงครั้งเดียว ของแบบฟูลสเต็ป 2 แบบแรก ความละเอียดของการหมุนตำแหน่งองศาต่อสเต็ป ก็เป็นสองเท่าของแบบแรก ความถูกต้องของตำแหน่งที่กำหนดจึงมีมากขึ้น

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

รูป 6.15 ก แบบฟูลสเต็ป 1 เฟส

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

รูป 6.15 ข แบบฟูลสเต็ป 2 เฟส

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

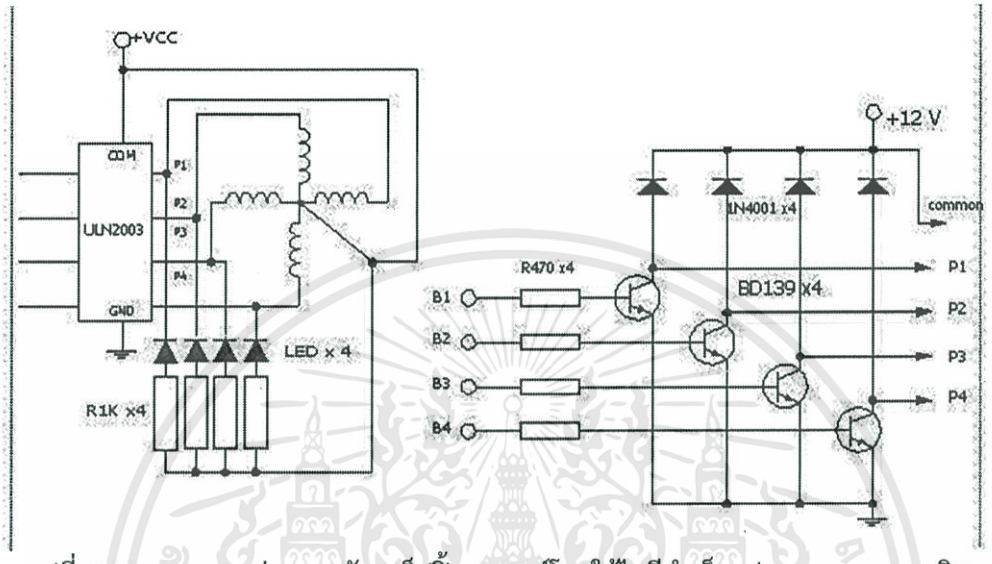
รูป 6.15 ค แบบฮาลฟ์สเต็ป 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.10 ตารางแสดงการกระตุ้นเฟสแบบต่างๆของสเต็ปปิ้งมอเตอร์

วงจรที่ใช้ในการขับสเต็ปปิ้งมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรจากทรานซิสเตอร์ แสดงได้ในรูป 6.16 โดย ไอซีสำเร็จรูปเบอร์ ULN2003 จะมีคุณสมบัติเป็นไอซีไดรเวอร์กระแสสูงแบบคอลเล็ค

เตอร์เปิด สามารถเลือกแรงดันได้กว้าง 5-30 โวลท์ จ่ายกระแสได้สูงถึง 500 mA. ต่อขา และมีไดโอดที่ป้องกันกระแสย้อนกลับอยู่ภายในไอซี ส่วนแอลอีดีที่ต่อในวงจรเราจะต่อไว้เพื่อแสดงการกระตุ้นแต่ละเฟส ของแต่ละแบบ

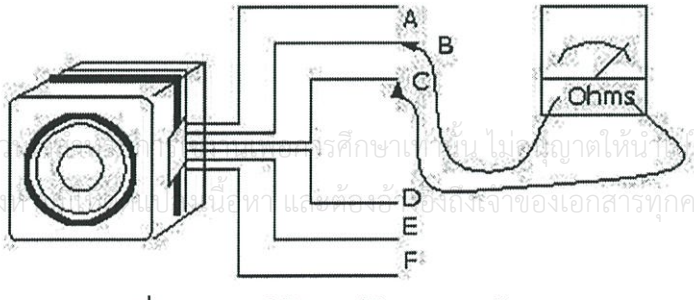


รูปที่ 2.11 แสดงการต่อวงจรขับสเต็ปปิ้งมอเตอร์โดยใช้ไอซีสำเร็จรูป และวงจรทรานซิสเตอร์

วิธีการตรวจสอบหาเฟสของขดลวดสเต็ปปิ้งมอเตอร์

ในขั้นตอนที่ 1 ให้สังเกตว่าสเต็ปปิ้งมอเตอร์ที่นำมาทดลองที่เป็นแบบยูนิโพล่าร์ (Uni-polar stepper motor) จะมีจำนวนสาย 5เส้นหรือ 6 เส้น (ถอดจากดิสก์ไดร์ฟเก่าขนาด 5 นิ้วนำมาใช้งานได้)

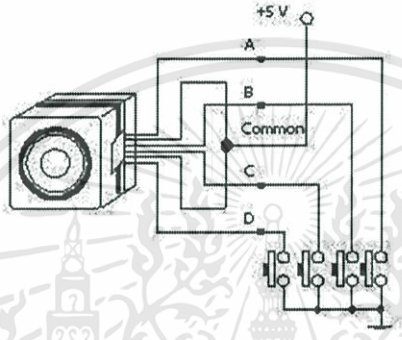
ในขั้นตอนที่ 2 ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขดดังรูป 6.17 ขั้นตอนการวัด ให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน(common) โดยให้ใช้ มัลติมิเตอร์ตั้งค่าไว้สำหรับการวัดค่าความต้านทาน แต่ละเส้น สังเกตที่ค่าความต้านทาน ถ้าหากเราไม่ได้วัดระหว่าง จุดต่อร่วม (common) กับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่างเช่น ถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 60 Ohm แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 120 Ohm หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน ทำนองเดียวกันหากเป็นแบบที่มีสาย 5 เส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 การใช้มิเตอร์วัดค่าความต้านทาน

ในขั้นตอนที่ 3 หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดร่วมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรตาม รูปหลังจากนั้นให้ทดลองกดสวิตช์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่ จุด A จุด B จุด C และจุด D แล้วให้สังเกตการหมุนของสเต็ปปิ้งมอเตอร์ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดดข้ามสเต็ปก็ให้ทดลองโดยเรียงลำดับการกดสวิตช์ใหม่ จนหาลำดับของสายได้ถูกต้องคือมอเตอร์เดินตามที่ละสเต็ป อย่างเป็นลำดับ



รูปที่ 2.13 แสดงการต่อวงจรเพื่อทดสอบโดยการสวิตช์เพื่อหาลำดับ

2.7 บอร์ดDrive stepper motor (MSTEP-2U)

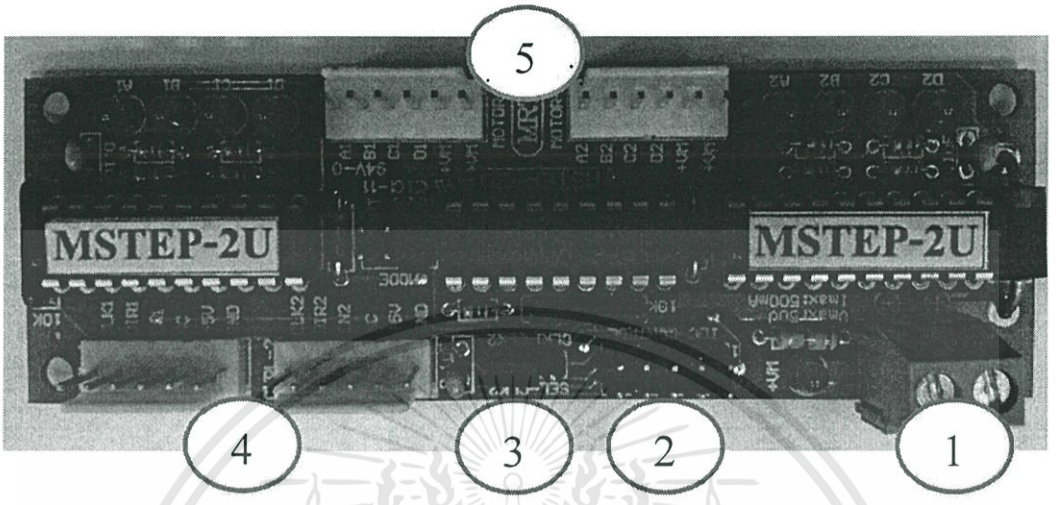
บอร์ด MSTEP-2U เป็นบอร์ดdrive stepper motor เพื่อช่วยลดการประมวลผลของไมโครคอนโทรลเลอร์ลง โดยสามารถขับมอเตอร์แบบ Unipolar (5 สาย/ 6สาย) ได้ถึง 2 ตัว และแต่ละชุดจะใช้สายสัญญาณควบคุมเพียง 3 เส้น คือ CLK, EN และ DIR และยังสามารถกำหนดรูปแบบการทำงานได้ทั้ง 1 Phase, 2 Phase และ Half Step ได้ตามรุ่นที่ต้องการ

2.7.1 Technical Specification

1. ชนิดของมอเตอร์ : Unipolar Stepping Motor
2. พิกัดการขับโหลด : 50 Vmax, 500mA (ULN2803)
3. จำนวนมอเตอร์ : 2 ตัว
4. สัญญาณการควบคุม : TTL (CLK, EN, DIR)
5. รูปแบบการทำงาน : 1 Phase/ 2 Phase รุ่น MStep-2U, Half Step/2 Phase รุ่น MStep- 2U-H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 โครงสร้างและส่วนประกอบ



รูปที่ 2.14 โครงสร้างของบอร์ดDriving Stepper Motor

1. V-MOTOR : พอร์ตสำหรับป้อนแรงดันไฟเลี้ยงให้กับสเต็ปปิ้งมอเตอร์
2. IDC-Control : พอร์ตสัญญาณควบคุมการทำงาน 10ขา โดยจัดเรียงพอร์ตตามมาตรฐานของ MRT จึงสามารถอินเตอร์เฟสร่วมกับบอร์ดไมโครคอนโทรลเลอร์ต่างๆ ของทางบริษัทได้ทันทีและมีการจัดเรียงขาพอร์ตดังต่อไปนี้

CLK1 = สัญญาณนาฬิกา (clock) ควบคุมความเร็วของมอเตอร์ชุดที่ 1
DIR1 = สัญญาณควบคุมทิศทางการหมุนของมอเตอร์ชุดที่ 1
EN1 = สัญญาณ Enable เพื่อควบคุมการทำงานของมอเตอร์ชุดที่ 1 ทำงานที่ลอจิก “0”
CLK2 = สัญญาณนาฬิกา (clock) ควบคุมความเร็วของมอเตอร์ชุดที่ 2
DIR2 = สัญญาณควบคุมทิศทางการหมุนของมอเตอร์ชุดที่ 2
EN2 = สัญญาณ Enable เพื่อควบคุมการทำงานของมอเตอร์ชุดที่ 2 ทำงานที่ลอจิก “0”
+5V = แรงดันไฟเลี้ยงวงจรภายในบอร์ด
0V = กราวด์
NC = No-connect ไม่เชื่อมต่อกับสัญญาณใดๆ

3. SEL-SCK : จั๊มเปอร์เลือกสัญญาณ Clock เพื่อควบคุมวงจรขับชุดที่ 2 โดยสามารถเลือกเป็น CLK1 หรือ CLK ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

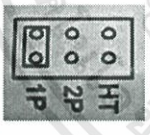
- เลือกCLK1 เมื่อต้องการให้วงจรขับมอเตอร์ทั้งสองชุดใช้ สัญญาณ Clock ร่วมกันจากสัญญาณที่ขา CLK1 ในขณะนี้สัญญาณนาฬิกาที่ขา CLK2 จะไม่มีผลต่อการทำงานของมอเตอร์ชุดที่2

- เลือกCLK2 เมื่อต้องการให้วงจรขับมอเตอร์ทั้งสองชุดใช้สัญญาณ Clock อิสระต่อกัน

4. CTRL-1, CTRL-2 : พอร์ตสัญญาณควบคุมการทำงานจัดวางขาในด้วยคอนเน็กเตอร์ขนาด 8 ขา ดังนี้ โดยมีหน้าที่และการทำงานของแต่ละขาดังที่ได้อธิบายในส่วนของ IDC-Control.

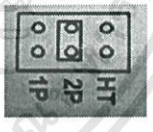
5. MODE : จั๊มเปอร์เลือกโหมดหรือรูปแบบการขับสเต็ปปิ้งมอเตอร์ ขึ้นอยู่กับรุ่นของบอร์ดดังนี้

- รุ่นMSTEP-2U : สามารถกำหนดรูปแบบการขับสเต็ปปิ้งมอเตอร์ได้ 2 แบบคือ 1 Phase และ 2 Phase ดังแสดงในรูปต่อไปนี้



1 Phase Drive

หรือ



2 Phase Drive

2.7.3 ตารางแสดงการขับสเต็ปปิ้งมอเตอร์ในแบบต่างๆ

2 Phase Drive

STEP	PHASE			
	A	B	C	D
1	ON	ON	—	—
2	—	ON	ON	—
3	—	—	ON	ON
4	ON	—	—	ON

1 Phase Drive

STEP	PHASE			
	A	B	C	D
1	ON	—	—	—
2	—	ON	—	—
3	—	—	ON	—
4	—	—	—	ON

Half Step Drive

STEP	PHASE			
	A	B	C	D
1	ON	—	—	—
2	ON	ON	—	—
3	—	ON	—	—
4	—	ON	ON	—
5	—	—	ON	—
6	—	—	ON	ON
7	—	—	—	ON
8	ON	—	—	ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 สัญลักษณ์และตารางแสดงการใช้งาน

บทที่ 3
หลักการทำงานและออกแบบ

3.1 หลักการออกแบบ

ออกแบบและพัฒนาโปรแกรมบนภาษา C++ โดยใช้ Microsoft Visual Studio 2008 เนื่องจาก เป็นโปรแกรมกลุ่ม IDE(integrated development environment)เพื่อพัฒนาโปรแกรมบนภาษา C++โดยเฉพาะเนื่องจากง่ายและสะดวกต่อการใช้งาน นอกจากนั้นยังอยู่ในโครงการซอฟแวร์เพื่อการศึกษาจากไมโครซอฟต์ ทางผู้จัดทำโครงงานจึงสามารถใช้งานได้อย่างถูกต้องตามลิขสิทธิ์โดยทางผู้จัดทำโครงงานได้เลือกใช้ Library ที่มีชื่อว่า OpenCV (Open Source Computer Vision) ซึ่งเป็น Library ที่ใช้เขียนโปรแกรมเกี่ยวกับ Computer Vision เป็นหลัก เหตุผลที่ทางผู้จัดทำ โครงการเลือกใช้ OpenCV เนื่องจากเป็น Libraryที่สามารถนำไปใช้ได้เลยโดยมี Algorithm พื้นฐานสามารถเรียกใช้งานและสามารถนำ Algorithm เหล่านี้ไปประยุกต์ใช้งานในรูป แบบที่ต้องการได้ซึ่งเหมาะสมกับโครงงานเป็นอย่างมาก

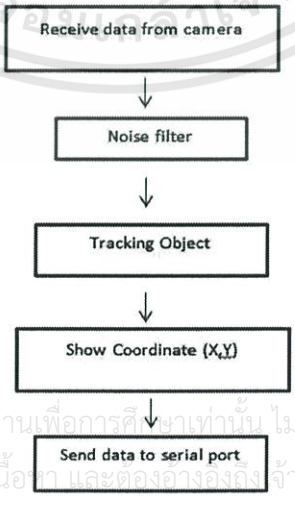
3.2 หลักการทำงานของระบบ

ขั้นตอนการพัฒนา

แบ่งออกเป็น 2 ขั้นตอนหลักๆ คือ

3.2.1 ประมวลผลด้วย Image Processing

ขบวนการประมวลผลภาพขั้นต้น (Pre-Processing) สามารถแบ่งการทำงานย่อยๆ ได้ดังนี้ การกรองข้อมูลแทรกซ้อน (Noise Filtering) เป็นวิธีทีการที่ทำหน้าที่หน้าที่กำจัดสิ่งไม่พึงประสงค์ออกที่ปรากฏอยู่บนภาพที่ต้องการจะนำมาประมวลผลและติดตามวัตถุ (Tracking) เป็นการจับรูปภาพที่เคลื่อนไหวโดยใช้หลักการจับสีที่ต้องการการและแสดงพิกัดภาพในแนวระนาบแกน(x,y) แล้วส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port)

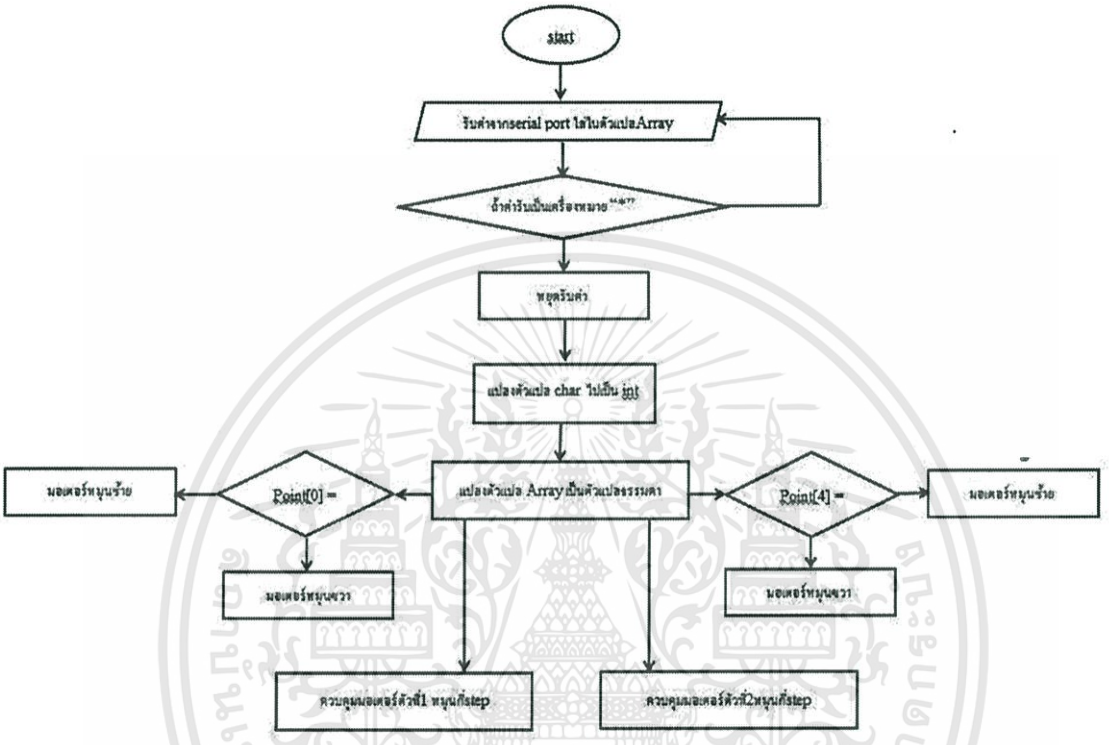


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 ขั้นตอนการทำงานของ Image Processing

3.2.2 ประมวลผลด้วยไมโครคอนโทรลเลอร์

แสดงโครงสร้างโดยรวมของการควบคุมกล้องด้วยไมโครคอนโทรลเลอร์



รูปที่ 3.2 โค้ดแสดงการควบคุมกล้องด้วยบอร์ดอาร์ดูโน่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

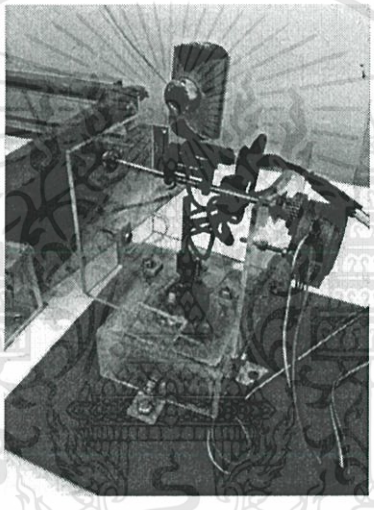
บทที่ 4
การทดลองและผลการทดลอง

การทดลองจะแบ่งออกเป็น 2 ส่วน คือ

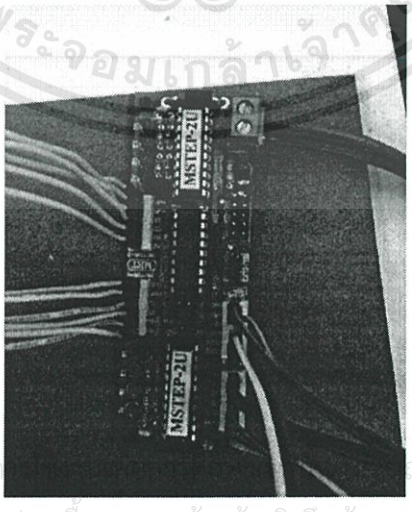
- ฮาร์ดแวร์
- ซอฟต์แวร์

ในการทดลองส่วนแรก คือ

4.1 ส่วนของฮาร์ดแวร์

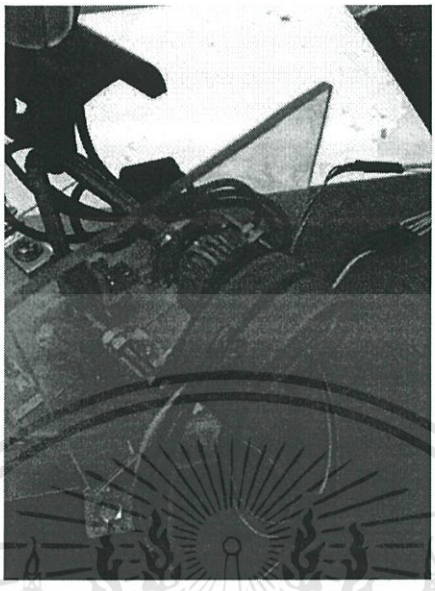


รูปที่ 4.1 กล้องWebcamและฐานตั้งกล้อง

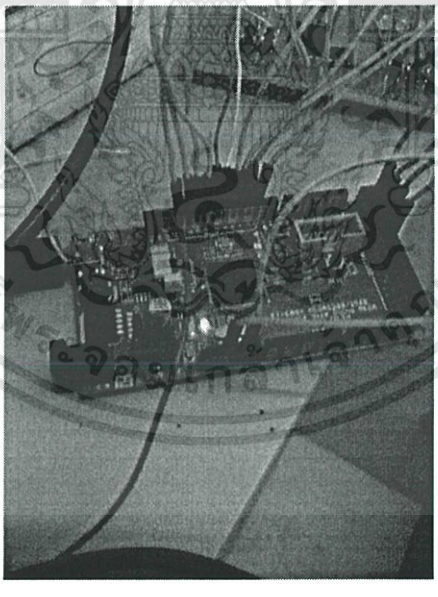


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก ญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 วงจร Driving Stepping Motor



รูปที่ 4.3 Stepping Motor



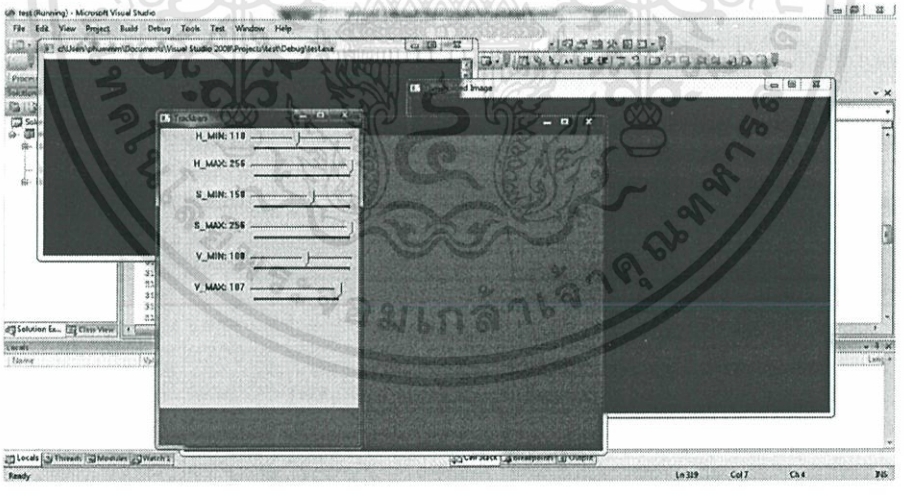
รูปที่ 4.4 บอร์ดอาร์ดูโน่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของซอฟต์แวร์

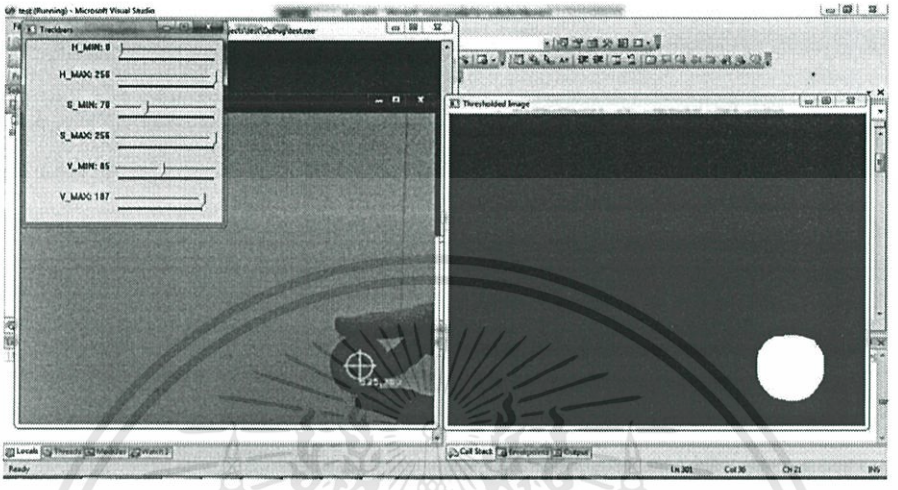
4.2.2 Visual studio 2008

รูปที่ 4.5 ข้อมูลรับภาพเริ่มต้น

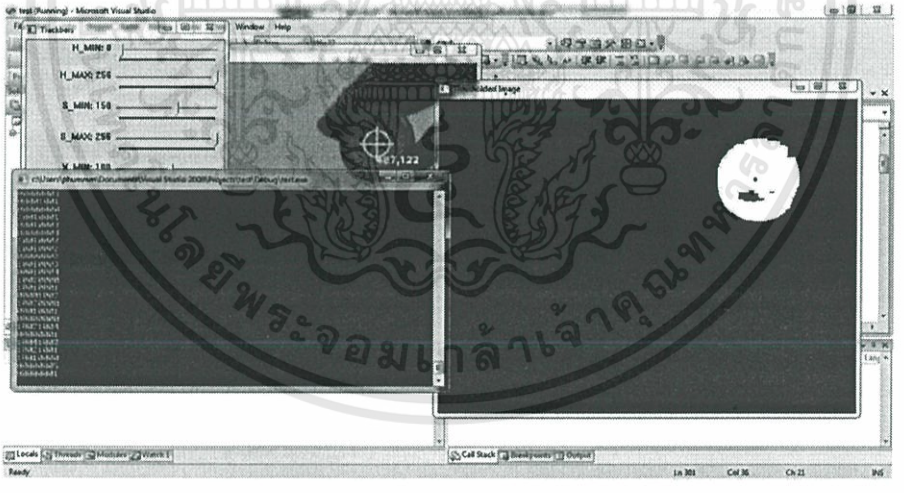


รูปที่ 4.6 HSV Track Bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 Tracking Object



รูปที่ 4.8 แสดงพิกัดพร้อมส่งผ่านพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 Arduino

รูปที่ 4.9 โปรแกรมอาร์ดูโน่

ตำแหน่ง	ระยะerrorจากกึ่งกลาง		เปอร์เซ็นต์ความคาดเคลื่อน	
	x	y	x	y
ซ้ายบน (20,20)	0	-20	0	8.33
ซ้ายกลาง (20,240)	20	0	8.33	0
ซ้ายล่าง (20,460)	0	20	0	8.33
กลางล่าง (320,460)	20	0	8.33	0
ขวาล่าง (620,460)	20	-60	8.33	25
ขวากลาง (620,240)	40	0	16.67	0
ขวาบน (620,20)	-20	20	8.33	8.33
กลางบน (320,20)	0	-20	0	8.33

รูปที่ 4.10 ตารางแสดงผลการทดลอง

จุดกึ่งกลางของภาพอยู่ที่ (320,240)

จากตารางดังกล่าวแสดงถึงการวางวัตถุไว้ที่ขอบของภาพ แล้วดูค่าความคลาดเคลื่อนที่ออกมา โดยผ่านการคอมไพล์ y -20 คือ 240-20=220

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
วิเคราะห์และสรุปผลการทดลอง

จากการทดลองพบว่าในการควบคุมทิศทางกล้องในแนวแกน Y นั้น พบว่าเฟืองของมอเตอร์ในบางครั้งจะไม่สัมผัสกับเฟืองที่ทำให้กล้องหมุน ซึ่งทำให้กล้องไม่หมุนได้ตามที่ระยะ และส่งผลทำให้เกิดข้อผิดพลาดได้ และในตอนที่ควบคุมกล้องนั้น ไม่สามารถทำให้มอเตอร์หมุนพร้อมกันสองตัวพร้อมกันไม่ได้โดยโปรแกรมจะต้องสั่งให้มอเตอร์ตัวแรกหมุนให้เสร็จก่อนจึงจะทำให้มอเตอร์อีกตัวหมุนตาม จึงทำให้เกิดดีเลย์ในการใช้จริงโดยจะแก้ไขได้โดยเพิ่มความเร็วของมอเตอร์ให้เร็วขึ้นจะช่วยลดดีเลย์ลงได้ และจากผลการทดลองนั้นพบว่าค่าความผิดพลาดจากมอเตอร์แกน Y จะพบมากกว่ามอเตอร์แกน X อาจเกิดจากการรับน้ำหนักในแนวดิ่ง ในส่วนของโปรแกรม Visual Studio จากการพล็อตข้อมูลถี่เกินไปจึงทำให้มีการส่งข้อมูลเป็นจำนวนมากจึงทำให้เกิดการส่งค่าได้เฉพาะค่าแรกค่าเดียว ความผิดพลาดขึ้นบ้างซึ่งทำปัญหาที่เกิดขึ้นหลักๆจะพบระหว่างการส่งข้อมูลจากโปรแกรม Visual Studio ถึง Arduino ผ่าน Serial Port

จากการทำโครงงานนี้เมื่อมองจากภาพรวมของโครงงานพบว่าโครงงานนี้ทำให้ผู้จัดทำได้รับความรู้ทางด้านการเขียนโปรแกรม อีกทั้งยังได้พบปัญหา ที่นำไปสู่แนวทางการแก้ไขปัญหา เพื่อที่จะนำมาปรับปรุงให้ดีขึ้นในภายภาคหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. หนังสือ “เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino”
2. เรียนรู้ไมโครคอนโทรลเลอร์ด้วยโปรแกรมภาษา C/C++ กับ Arduino และโมดูล POP-MCU
3. หนังสือ Image Processing by OpenCV อ.ชูชาติ ปิณฑวิรุจน์
4. http://playground.arduino.cc/Interfacing/CPPWindows
5. http://www.shadowwares.com/forum/index.php?board=12.0
6. http://stackoverflow.com/questions/6838199/c-serial-interfacing-with-arduino?rq=1
7. http://arduino.cc/en/Tutorial/HomePage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน OpenCV Library เบื้องต้นเพื่อการประมวลผลภาพจากกล้อง (สำหรับ MS Windows)

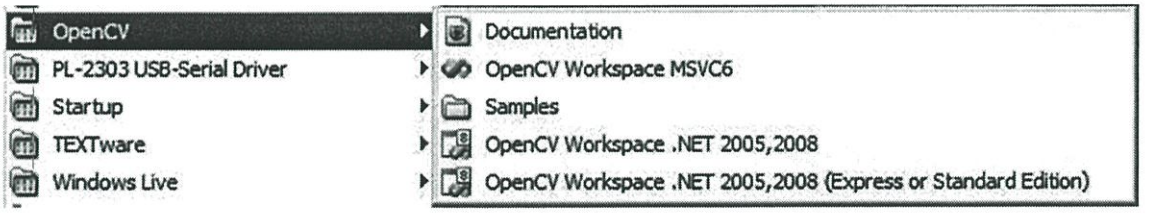
September 23rd, 2009 by Sathit Wanitchaikit

1. การติดตั้ง

สามารถ download ตัว installer ได้ที่ http://sourceforge.net/projects/opencvlibrary

โดยสำหรับ MS Windows platform นั้นให้เลือก download จากหัวข้อ opencv-win ซึ่งจะมีอยู่หลาย version ด้วยกัน แต่หลายตัวยังเป็น beta version อยู่ ดังนั้น แนะนำให้เลือก donwload ตัว stable version ล่าสุด ซึ่งก็คือ version 1.1pre1 โดยใน installer package จะประกอบไปด้วยชุด library พร้อมทั้ง API Reference ของฟังก์ชัน และ data structure ต่างๆ ที่เราสามารถเรียกใช้ได้

หลังจากที่ download เรียบร้อยแล้ว ให้ double click เพื่อเริ่มทำการติดตั้งตามลำดับของ installation wizard จนเสร็จสิ้น ซึ่งหลังจากการติดตั้งเราสามารถเรียกดูคู่มือการใช้งานและตัวอย่างโปรแกรมได้จาก link ที่ถูกสร้างไว้ใน start menu



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับแหล่งข้อมูลเพิ่มเติมของการใช้งานนอกเหนือจากที่แถมมากับ installer สามารถหาอ่านได้ที่

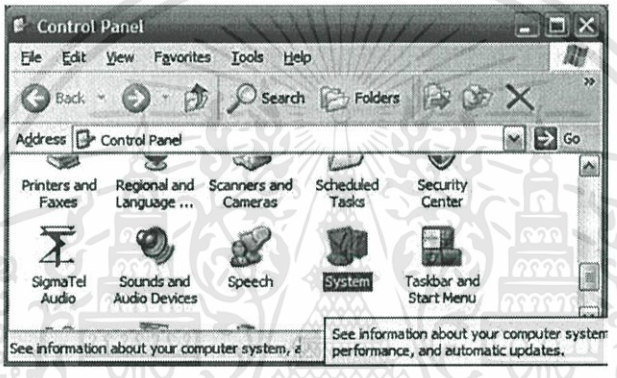
http://opencv.willowgarage.com/wiki/FullOpenCVWiki

2. การตั้งค่า Environment สำหรับการใช้งานกับ Visual C++ 2008 Express Edition

เพื่อให้ Visual C++ เรียกใช้ OpenCV ได้นั้น จำเป็นต้องตั้งค่า Environment Variables ของระบบเสียก่อน จึงจะทำให้สามารถค้นหาคำสั่งและไฟล์ต่างๆ ที่เราเรียกใช้งานได้ โดยมีขั้นตอนการตั้งค่าตามลำดับดังนี้

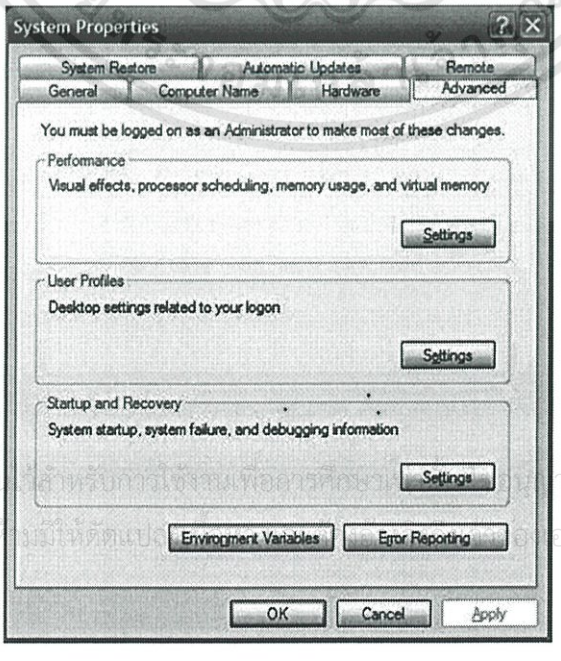
2.1 การตั้งค่าระบบเพื่อค้นหา DLL file

ให้ทำการเปลี่ยนแปลงค่า environment variable ที่ชื่อ PATH โดยให้เพิ่มตำแหน่ง path ที่เก็บ DLL files (โดยปกติเก็บอยู่ที่ C:\Program Files\OpenCV\bin)ของ OpenCV เข้าไปต่อข้างท้ายค่าของตัวแปรเดิม ซึ่งการแก้ไขค่านั้น ทำโดยเรียก environment variable dialog ขึ้นมา ซึ่งเรียกผ่าน system ใน control panel



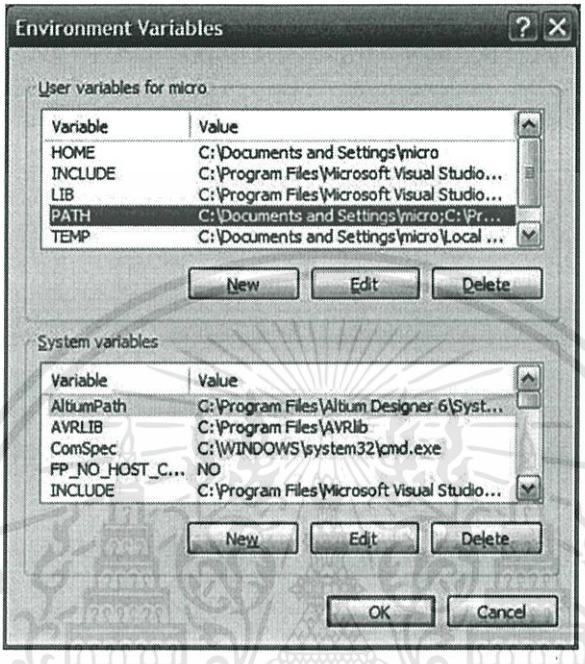
จากนั้นให้เลือกที่ tab ชื่อ advanced แล้วมองหาปุ่ม environment varibles

แล้วให้ click เพื่อเรียก dialog สำหรับการตั้งค่าขึ้นมา

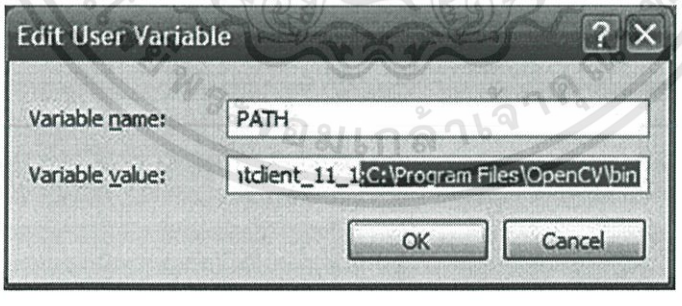


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้น ให้ click เลือกตัวแปรที่ชื่อ PATH แล้วทำการ click ที่ปุ่ม edit เพื่อทำการแก้ไขค่า



ทำการนำค่า path ที่เก็บ DLL files ของ OpenCV ไว้ที่ข้างท้ายของค่าเดิม โดยมีเครื่องหมาย ";" นำหน้า จากนั้นให้กดปุ่ม OK เพื่อยืนยันการแก้ไขค่า



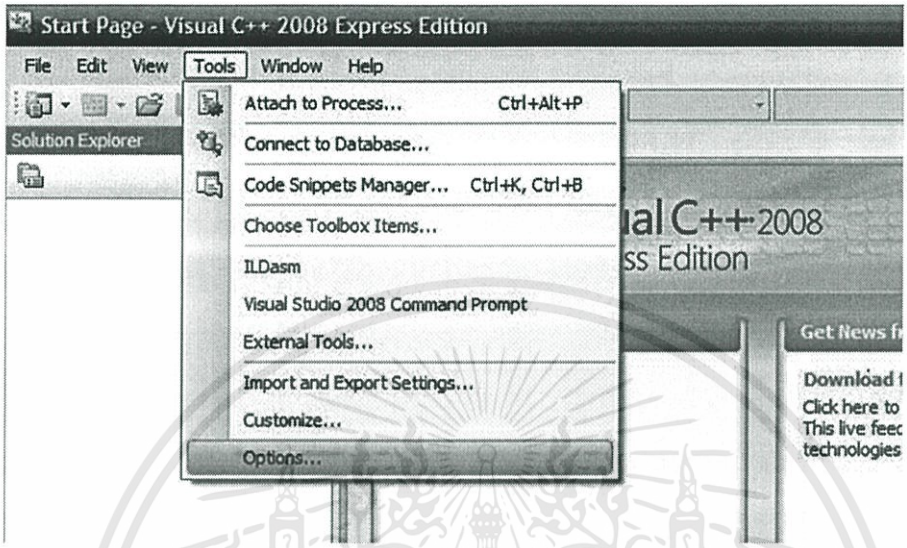
เมื่อทำเสร็จแล้ว ระบบจะทำการเรียกใช้ DLL files จาก path ดังกล่าว เมื่อโปรแกรมที่เราเขียนขึ้นเรียกใช้ DLL files ของ OpenCV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

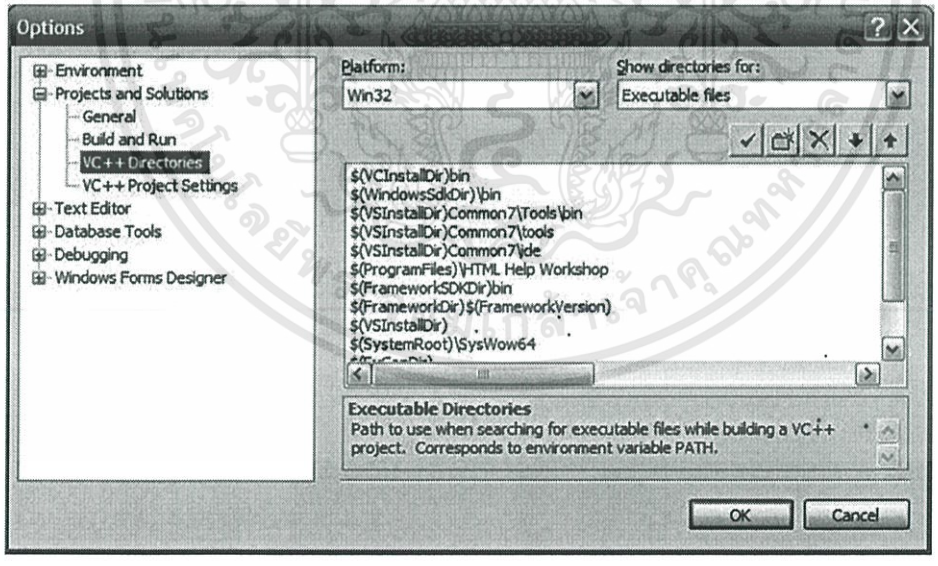
2.2. การตั้งค่าระบบเพื่อการค้นหา header file และ static library file

ในการ compile โปรแกรมที่เรียกใช้ function ของ OpenCV จำเป็นที่จะต้องอ้างอิงถึงข้อมูลใน static library files, include files และ source files ดังนั้นแล้วจึงต้องตั้งค่าต่างๆ เหล่านี้ให้ระบบของ Visual C++ รู้จักเสียก่อน จึงจะสามารถที่จะใช้งาน

ได้ ซึ่งการตั้งค่าทำโดยการเปิดโปรแกรม Visual C++ ขึ้นมา แล้วเรียก Option dialog เพื่อทำการแก้ไขข้อมูลของ VC++ directories

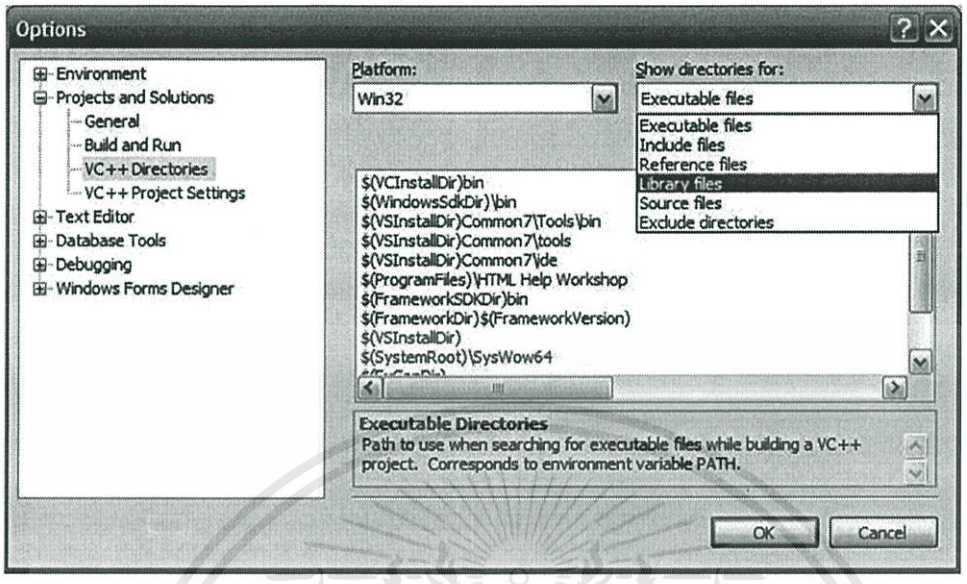


ทำการเลือกรายการที่ชื่อ VC++ Directories ซึ่งเป็น sub-entry ภายใต้ Project and Solutions ซึ่งอยู่ทางซ้ายมือของหน้าต่าง



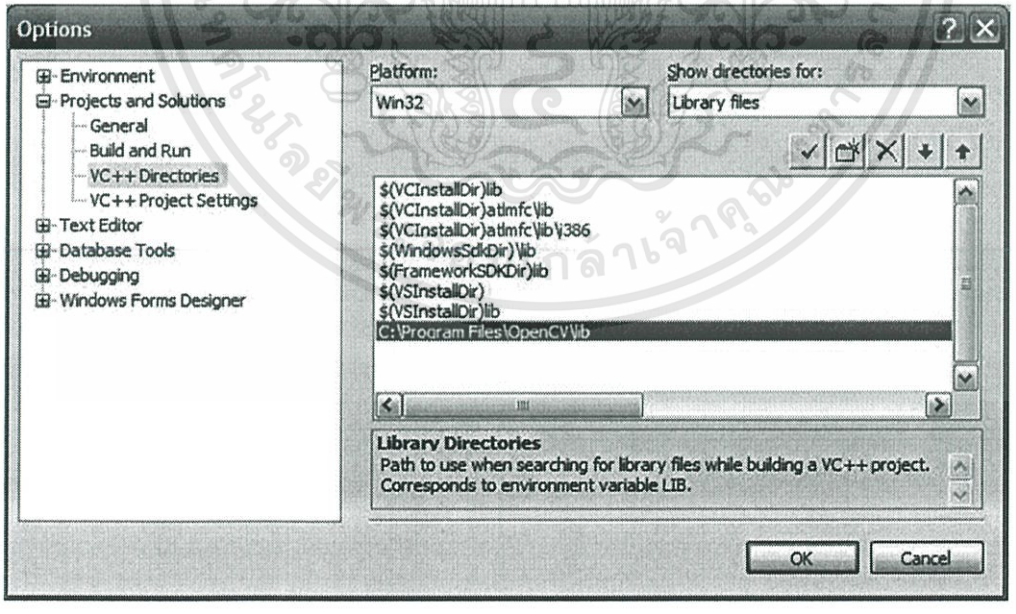
ใน drop down list ที่ชื่อ Show directories for: ให้เลือกเป็น Library files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากนั้นทำการเพิ่ม path ต่อไปนี้ที่รายการสุดท้าย

C:\Program Files\OpenCV\lib



เมื่อเสร็จสิ้นการเพิ่มpath ของ library files แล้ว ให้เปลี่ยนหัวข้อของ Show directories for: เป็น Include files แล้วให้เพิ่ม path ต่อไปนี้ที่ข้างท้าย

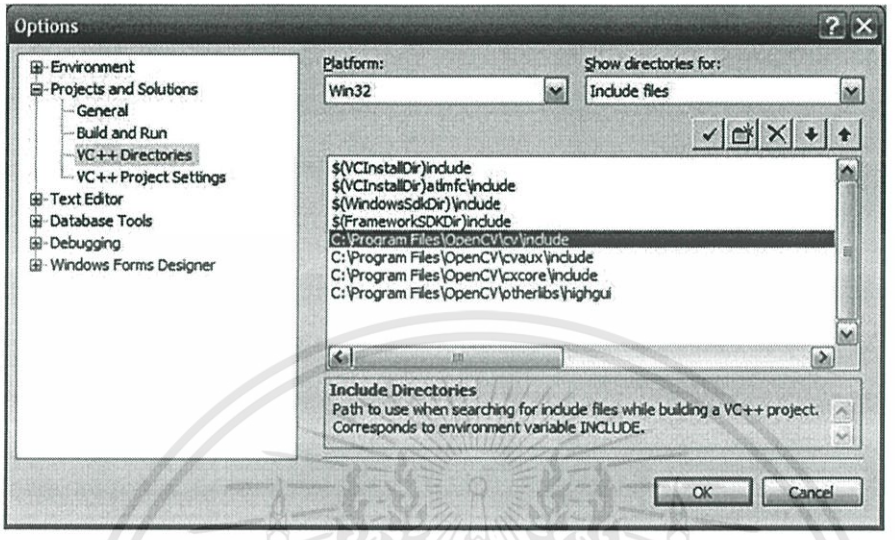
เอกสารนี้ถือเป็นลิขสิทธิ์ของมหาวิทยาลัยเจ้าคุณสมเด็จฯ ผู้ใดจะนำไปใช้ประโยชน์ด้านการค้า
เสียค่าธรรมเนียมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Program Files\OpenCV\cv\include

C:\Program Files\OpenCV\cvaux\include

C:\Program Files\OpenCV\cxcore\include

C:\Program Files\OpenCV\otherlibs\highgui



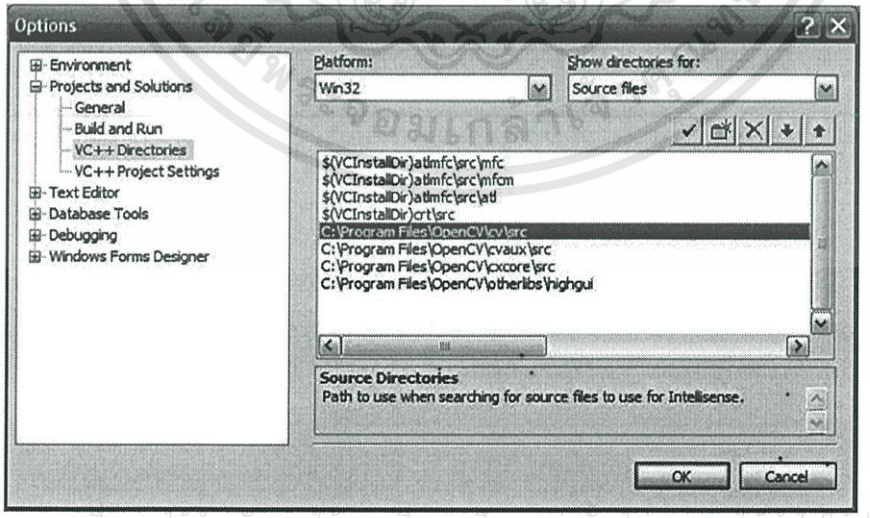
จากนั้นให้เปลี่ยนหัวข้อของ Show directories for: เป็น Source files แล้วให้เพิ่ม path ต่อไปนี้ที่ข้างท้าย

C:\Program Files\OpenCV\cv\src

C:\Program Files\OpenCV\cvaux\src

C:\Program Files\OpenCV\cxcore\src

C:\Program Files\OpenCV\otherlibs\highgui



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเสร็จสิ้นการเพิ่มpath ข้างต้นทั้งหมดแล้ว ให้ click ปุ่ม OK เพื่อยืนยันการเพิ่มค่า

2.3. การตั้งค่าระบบเพื่อการใช้งาน library files

เนื่องจาก library files ต่างๆ ที่มาพร้อมกับ installer เป็น library ที่ถูกสร้างให้ใช้กับ application ที่สร้างโดย Visual C++ 2005 ดังนั้นแล้ว เพื่อให้สามารถทำงานเข้ากันได้กับ Visual C++ 2008 จึงจำเป็นต้องมีการแก้ไขบางอย่าง ซึ่งทำได้ 2 วิธี ด้วยกันได้แก่

1. ทำการ recompile ตัว library ใหม่ทั้งหมดให้เข้ากันได้กับ Visual C++ 2008

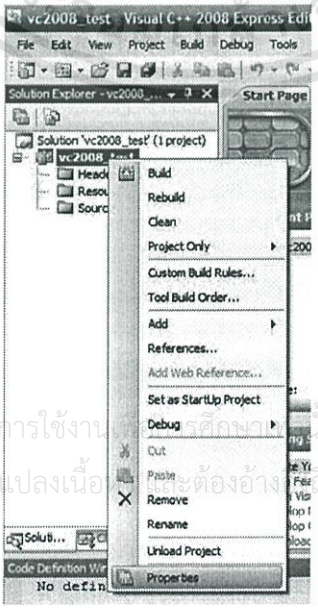
หรือ

2. ให้ทำการติดตั้ง package เสริมจาก Microsoft ที่ชื่อ "Microsoft Visual C++ 2005 Redistributable Package" ซึ่งเป็น runtime component ที่ช่วยให้ระบบที่ไม่ได้ติดตั้ง Visual C++ 2005 สามารถใช้งานโปรแกรมหรือ library ที่สร้างจาก Visual C++ ได้ ซึ่งสามารถ download ตัว package ดังกล่าวได้
ที่ http://www.microsoft.com/downloads/details.aspx?familyid=200B2FD9-AE1A-4A14-984D-389C36F85647&displaylang=en

โดยให้เลือกวิธีการใดวิธีการหนึ่งเท่านั้น ซึ่งผมแนะนำให้เลือกวิธีที่ 2 ซึ่งยุ่งยากน้อยกว่าการ recompile ตัว library ใหม่ทั้งหมดมากครับ

2.4. การตั้งค่าสำหรับการ build executable file

เมื่อทำตาม 3 ขั้นตอนข้างต้นเรียบร้อยแล้ว Visual C++ ก็พร้อมที่จะสร้าง project สำหรับสร้างโปรแกรมที่เรียกใช้งาน OpenCV library โดย project ต่างๆ ที่ต้องการเรียกใช้งานนั้น ต้องเพิ่มรายการของ static library ของ OpenCV เข้าในส่วนของขั้นตอนการทำ linking เมื่อทำการ build executable file ซึ่งสามารถเพิ่มรายการของ library ดังกล่าวได้จาก properties dialog ของ project โดยเรียกผ่านจาก panel ที่ชื่อ solution explorer (ปกติแล้วจะอยู่ทางด้านขวาของหน้าต่าง Visual C++ ยกเว้นว่าจะมีคนไปย้ายตำแหน่งมัน) โดยให้ right-click ที่ชื่อ project แล้วเลือกคำสั่ง properties จาก list ที่ปรากฏขึ้นมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

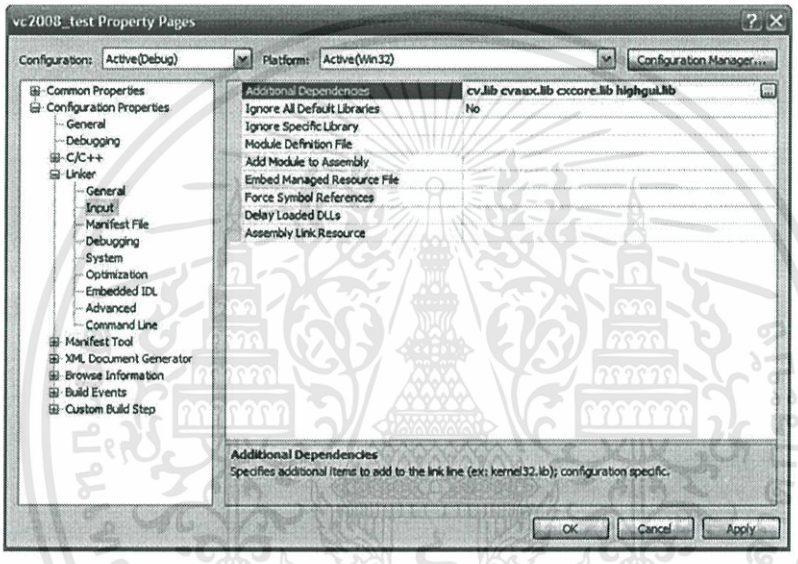
ให้เปิด Configuration Properties->Linker->Input จากรายการที่อยู่ทางด้านซ้ายมือ แล้วทำการเพิ่มรายชื่อ library ต่อไปนี้ในช่อง Additional Dependecies แล้วทำการกดปุ่ม OK เพื่อยืนยันการตั้งค่าดังกล่าว

cv.lib

cvaux.lib

cxcore.lib

highgui.lib



จากนั้นเมื่อทำการ build executable file เราก็จะได้โปรแกรมที่เรียกใช้งาน function ต่างๆ จาก OpenCV library ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code Image Processing

```
#include <opencv\highgui.h>
#include <opencv\cv.h>
#include "SerialClass.h"
#include <iostream>
#include <stdio.h>
#include <tchar.h>
#include <string>
#include <sstream>

using namespace std;
using namespace cv;

Serial::Serial(char *portName)
{

    this->connected = false;
      this->hSerial = CreateFile(portName,
        GENERIC_READ | GENERIC_WRITE,
        0,
        NULL,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    //Check if the connection was successful.
    if (this->hSerial==INVALID_HANDLE_VALUE)
    {
        //If not success full display an Error.
        if (GetLastError()==ERROR_FILE_NOT_FOUND){

            //Print Error if neccessary
            printf("ERROR: Handle was not attached. Reason: %s not
available.\n", portName);

        }
        else
        {
            printf("ERROR!!!");
        }
    }
    else
    {
        //If connected we try to set the comm parameters
        DCB dcbSerialParams = {0};

        //Try to get the current
        if (!GetCommState(this->hSerial, &dcbSerialParams))
        {
            //If impossible, show an error
            printf("failed to get current serial parameters!");
        }
        else
        {
            //Define serial connection parameters for the arduino board
            dcbSerialParams.BaudRate=CBR_9600;
            dcbSerialParams.ByteSize=8;
            dcbSerialParams.StopBits=ONESTOPBIT;
            dcbSerialParams.Parity=NOPARITY;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            //Set the parameters and check for their proper application
            if (!SetCommState(hSerial, &dcbSerialParams))
            {
               printf("ALERT: Could not set Serial Port parameters");
            }
            else
            {
                //If everything went fine we're connected
                this->connected = true;
                //We wait 2s as the arduino board will be reseting
                Sleep(ARDUINO_WAIT_TIME);
            }
        }
    }
}

Serial::~Serial()
{
    //Check if we are connected before trying to disconnect
    if (this->connected)
    {
        //We're no longer connected
        this->connected = false;
        //Close the serial handler
        CloseHandle(this->hSerial);
    }
}

int Serial::ReadData(char *buffer, unsigned int nbChar)
{
    //Number of bytes we'll have read
    DWORD bytesRead;
    //Number of bytes we'll really ask to read
    unsigned int toRead;

    //Use the ClearCommError function to get status info on the Serial port
    ClearCommError(this->hSerial, &this->errors, &this->status);

    //Check if there is something to read
    if (this->status.cbInQue>0)
    {
        //If there is we check if there is enough data to read the required
number
        //of characters, if not we'll read only the available characters to
prevent
        //locking of the application.
        if (this->status.cbInQue>nbChar)
        {
            toRead = nbChar;
        }
        else
        {
            toRead = this->status.cbInQue;
        }

        //Try to read the require number of chars, and return the number of
read bytes on success
        if (ReadFile(this->hSerial, buffer, toRead, &bytesRead, NULL) &&
bytesRead != 0)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return bytesRead;
        }

    }

    //If nothing has been read, or that an error was detected return -1
    return -1;
}


bool Serial::WriteData(char *buffer , unsigned int nbChar)
{
    DWORD bytesSend ;

    //Try to write the buffer on the Serial port
    if (!WriteFile(this->hSerial, (void *)buffer, nbChar, &bytesSend, 0))
    {
        //In case it don't work get comm error and return false
        ClearCommError(this->hSerial, &this->errors, &this->status);

        return false;
    }
    else
        return true;
}

bool Serial::IsConnected()
{
    //Simply return the connection status
    return this->connected;
}


int H_MIN = 0;
int H_MAX = 256;
int S_MIN = 0;
int S_MAX = 256;
int V_MIN = 0;
int V_MAX = 256;

const int FRAME_WIDTH = 640;
const int FRAME_HEIGHT = 480;

const int MAX_NUM_OBJECTS=50;

const int MIN_OBJECT_AREA = 20*20;
const int MAX_OBJECT_AREA = FRAME_HEIGHT*FRAME_WIDTH/1.5;

const string windowName = "Original Image";
const string windowName2 = "Thresholded Image";
const string trackbarWindowName = "Trackbars";
void on_trackbar( int, void* )
{//This function gets called whenever a
      // trackbar position is changed
}
string intToString(int number){

      std::stringstream ss;
      ss << number;
      return ss.str();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void createTrackbars(){


      namedWindow(trackbarWindowName,0);

      char TrackbarName[50];
      sprintf( TrackbarName, "H_MIN", H_MIN);
      sprintf( TrackbarName, "H_MAX", H_MAX);
      sprintf( TrackbarName, "S_MIN", S_MIN);
      sprintf( TrackbarName, "S_MAX", S_MAX);
      sprintf( TrackbarName, "V_MIN", V_MIN);
      sprintf( TrackbarName, "V_MAX", V_MAX);


    createTrackbar( "H_MIN", trackbarWindowName, &H_MIN, H_MAX, on_trackbar
);
    createTrackbar( "H_MAX", trackbarWindowName, &H_MAX, H_MAX, on_trackbar
);
    createTrackbar( "S_MIN", trackbarWindowName, &S_MIN, S_MAX, on_trackbar
);
    createTrackbar( "S_MAX", trackbarWindowName, &S_MAX, S_MAX, on_trackbar
);
    createTrackbar( "V_MIN", trackbarWindowName, &V_MIN, V_MAX, on_trackbar
);
    createTrackbar( "V_MAX", trackbarWindowName, &V_MAX, V_MAX, on_trackbar
);


}
void drawObject(int x, int y,Mat &frame){

      circle(frame,Point(x,y),20,Scalar(0,255,0),2);
    if(y-25>0)
    line(frame,Point(x,y),Point(x,y-25),Scalar(0,255,0),2);
    else line(frame,Point(x,y),Point(x,0),Scalar(0,255,0),2);
    if(y+25<FRAME_HEIGHT)
    line(frame,Point(x,y),Point(x,y+25),Scalar(0,255,0),2);
    else line(frame,Point(x,y),Point(x,FRAME_HEIGHT),Scalar(0,255,0),2);
    if(x-25>0)
    line(frame,Point(x,y),Point(x-25,y),Scalar(0,255,0),2);
    else line(frame,Point(x,y),Point(0,y),Scalar(0,255,0),2);
    if(x+25<FRAME_WIDTH)
    line(frame,Point(x,y),Point(x+25,y),Scalar(0,255,0),2);
    else line(frame,Point(x,y),Point(FRAME_WIDTH,y),Scalar(0,255,0),2);

      putText(frame,intToString(x)+","+intToString(y),Point(x,y+30),1,1,Sca
lar(0,255,0),2);

                              int old_x = 320 ;
                              int old_y = 240 ;
                              int z=0;

                              int dis_x = x - old_x;
                              int dis_y = y - old_y;
                              if(dis_x<0){
                                    z+=10000000;
                                    dis_x=dis_x*(-1);}
                              if(dis_y<0){
                                    z+=1000;
                                    dis_y=dis_y*(-1);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        int w=z+((dis_x*10000)+dis_y);

                        char data[9];
                        char data1[2]="*";

                        int datasize = sizeof(data);
                        int datasize1 = sizeof(data1);

                        sprintf(data, "%08d",w);

                        Serial* SP = new Serial("COM4");
                        if (SP->IsConnected())
                              printf("We're connected\n");


                        SP->Serial::WriteData(data,datasize);
                        SP->Serial::WriteData(data1,datasize1);

                        cout<<data<<" "<<data1<<endl;



}
void morphOps(Mat &thresh){



      Mat erodeElement = getStructuringElement( MORPH_RECT,Size(3,3));
    Mat dilateElement = getStructuringElement( MORPH_RECT,Size(8,8));

      erode(thresh,thresh,erodeElement);
      erode(thresh,thresh,erodeElement);


      dilate(thresh,thresh,dilateElement);
      dilate(thresh,thresh,dilateElement);

}
void trackFilteredObject(int &x, int &y, Mat threshold, Mat &cameraFeed){

      Mat temp;
      threshold.copyTo(temp);
      vector< vector<Point> > contours;
      vector<Vec4i> hierarchy;
      findContours(temp,contours,hierarchy,CV_RETR_CCOMP,CV_CHAIN_APPROX_SI
MPLE );

      double refArea = 0;
      bool objectFound = false;
      if (hierarchy.size() > 0)
      {
            int numObjects = hierarchy.size();
        if(numObjects<MAX_NUM_OBJECTS)
            {
                  for (int index = 0; index >= 0; index =
hierarchy[index][0])
                  {

                        Moments moment = moments((cv::Mat)contours[index]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
                        double area = moment.m00;

                        if(area>MIN_OBJECT_AREA && area<MAX_OBJECT_AREA &&
area>refArea)
                        {
                              x = moment.m10/area;
                              y = moment.m01/area;
                              objectFound = true;
                              refArea = area;


                        }else objectFound = false;
                  }

                  if(objectFound ==true)
                  {
                        putText(cameraFeed,"Tracking
Object",Point(0,50),2,1,Scalar(0,255,0),2);
                        drawObject(x,y,cameraFeed);
                  }

            }else putText(cameraFeed,"TOO MUCH NOISE! ADJUST
FILTER",Point(0,50),1,2,Scalar(0,0,255),2);
      }

}
int main(int argc, _TCHAR* argv[])
{

    bool trackObjects = true;
    bool useMorphOps = true;
      Mat cameraFeed;
      Mat HSV;
      Mat threshold;
      int x=0, y=0;
      createTrackbars();
      VideoCapture capture;
      capture.open(0);
      capture.set(CV_CAP_PROP_FRAME_WIDTH,FRAME_WIDTH);
      capture.set(CV_CAP_PROP_FRAME_HEIGHT,FRAME_HEIGHT);
      while(1){

            capture.read(cameraFeed);
            cvtColor(cameraFeed,HSV,COLOR_BGR2HSV);

      inRange(HSV,Scalar(H_MIN,S_MIN,V_MIN),Scalar(H_MAX,S_MAX,V_MAX),thres
hold);
            if(useMorphOps)
            morphOps(threshold);
            if(trackObjects)
            trackFilteredObject(x,y,threshold,cameraFeed);

            imshow(windowName2,threshold);
            imshow(windowName,cameraFeed);

            waitKey(30);
      }
      return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code Arduino

```
unsigned long currentMillis;

int signx;

int x;

int y;

int signy;

int ledPin = 11;   //control motor x axis

int ledPin2 = 8;   //control motor y axis

char point;

int i = 0;

int j = 0;

int k = 0;

int pointint[9];

int pin = 12;    //motor(x) turn left=1, turn right=0

int pin2 = 9;    //motor(y) up = 1, down = 0

int ledState = LOW;             // ledState used to set the LED

long previousMillis = 0;

long interval = 20;  //control speed motor

void setup()

{

 Serial.begin(9600);

 pinMode(ledPin, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode(ledPin2, OUTPUT);
  pinMode(pin, OUTPUT);
  pinMode(pin2, OUTPUT);
}

void loop()
{
 if(Serial.available()){
   point = Serial.read();
   if(point != '*'){
     pointint[j] = point - '0';
     Serial.print(pointint[j]);
     Serial.println(j);
     j++;
   }else{
     Serial.println("End");
     j = 0;
     x= (pointint[1]*100 + pointint[2]*10 + pointint[3])/10;
     y= (pointint[5]*100 + pointint[6]*10 + pointint[7])/10;
     signx= pointint[0]*1;
     signy= pointint[4]*1;
     if(x > 0){
```

เอกสารนี้จัดทำขึ้นเพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print("x = ");
Serial.println(x);
//detect sign
if(signx == 1){
  digitalWrite(pin, HIGH);
}else{
  digitalWrite(pin, LOW);
}
//Blink
i = 0;

unsigned long currentMillis = millis();
 while(i<x)
 {
   currentMillis = millis();
   if(currentMillis - previousMillis > interval)
   {
     previousMillis = currentMillis;
     if (ledState == LOW)
       ledState = HIGH;
     else
       ledState = LOW;
   digitalWrite(ledPin, ledState);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      //digitalWrite(9, HIGH);
       i++;
     }
    }


  }
  if(y > 0)  {
  Serial.print("y = ");
  Serial.println(y);
  //detect sign
  if(signy == 1){
   digitalWrite(pin2, HIGH);
  }else{
   digitalWrite(pin2, LOW);
  }
  //Blink
 k = 0;


 currentMillis = 0;
 previousMillis = 0;
 unsigned long currentMillis = millis();
      Serial.println(currentMillis);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  while(k<y)
  {
    currentMillis = millis();
    Serial.println(currentMillis);
    Serial.print("previousMillis is ");
    Serial.println(previousMillis);
    if(currentMillis - previousMillis > interval)
    {
     previousMillis = currentMillis;
     if (ledState == LOW)
       ledState = HIGH;
     else
       ledState = LOW;
     digitalWrite(ledPin2, ledState);
     k++;
     Serial.print("k = ");
    Serial.println(k);
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pointint[0] = 0;
pointint[1] = 0;
```

```
        pointint[2] = 0;
        pointint[3] = 0;
        pointint[4] = 0;
        pointint[5] = 0;
        pointint[6] = 0;
        pointint[7] = 0;
        pointint[8] = 0;
    }
  }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้