รถเข็นสินค้าอัตโนมัติควบคุมการทำงานด้วยการประมวลผลภาพ

โดย

ไกรฤกษ์ เชยชื่น

ปกรณ์ ทู้ไพเราะ

สนับสนุนงบประมาณโดย

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

ประจำปีงบประมาณ 2557

Automatic shopping cart by using image processing

by

Kairoek Choeychuen

Pakon Thuphairo

Granted by

Rajamangala University of Technology Rattanakosin

Fiscal year 2014

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จได้ด้วยความกรุณาของทุกๆ ฝ่าย และทุนวิจัยจากมหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์และคณะวิศวกรรมศาสตร์ในการสนับสนุนการทำผลงานวิจัยในครั้งนี้

ขอขอบพระคุณ นายณัฐภณ สังข์รักษา และนายทัศชัย เชิดเกียรติกุล นักศึกษาสาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์และสถาปัตยกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ สำหรับการสนับสนุนในการดำเนินการงานวิจัยนี้

สุดท้ายนี้ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ พี่ สำหรับทุกสิ่งทุกอย่างที่ข้าพเจ้าได้รับจากท่านซึ่งยากที่จะบรรยายออกมาทั้งหมดในที่นี้ ขอขอบพระคุณทุกๆท่านจากใจ

ไกรฤกษ์ เชยชื่น และคณะ

สิงหาคม 2557

บทคัดย่อ

รหัสโครงการ : Inno 005/2557

ชื่อโครงการ : รถเข็นสินค้าอัตโนมัติควบคุมการทำงานด้วยการประมวลผลภาพ

ชื่อนักวิจัย : ผศ.ดร.ไกรฤกษ์ เชยชื่น และ นายปกรณ์ ทู้ไพเราะ

งานวิจัยนี้แสดงการประยุกต์ใช้เทคนิคการตรวจจับสีในการประมวลผลภาพ เพื่อสร้างรถเข็นสินค้าที่สามารถเคลื่อนที่ตามรูปแบบเป้าหมายที่กำหนด ซึ่งช่วยให้ผู้สูงอายุ หรือผู้ที่ไม่มีความสามารถในการใช้งานรถเข็นได้เหมือนคนปกติทั่วไป ให้สามารถใช้รถเข็นได้ตามลำพังเมื่ออยู่ในห้างสรรพสินค้า โดยการทำงานของรถเข็นสินค้าอัตโนมัตินี้ประกอบด้วยส่วนสำคัญดังนี้ คือกล้องทำหน้าที่รับภาพเป้าหมายซึ่งถูกกำหนดให้เป็นป้ายสีแดง มาประมวลผลด้วยโปรแกรมตรวจจับสีเพื่อติดตามป้ายสีแดง ซึ่งถูกพัฒนาด้วย C# แล้วส่งข้อมูลดิจิตอลต่อไปให้ชุดควบคุมมอเตอร์ซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ทำให้ล้อของรถเข็นทั้งสองล้อเกิดการหมุนในทิศทางตามการเคลื่อนที่ของผู้ใช้ นั่นคือ เดินหน้า, ถอยหลัง, เลี้ยวซ้าย, และเลี้ยวขวา ตามลำดับ

การติดตามป้ายใช้การติดตามป้ายสีซึ่งได้ทำการเรียนรู้ไว้โดยใช้แบบจำลอง Non-stationary Gaussian แบบจำลองนี้ยอมให้สีที่กำลังติดตามมีการเปลี่ยนแปลงได้ซึ่งการเปลี่ยนแปลงอาจเกิดจากสัญญาณรบกวนหรือการเปลี่ยนแปลงสภาพแสงแวดล้อม

จากการทดลองพบว่าการติดตามป้ายสีโดยใช้แบบจำลอง Non-stationary Gaussian สามารถใช้ในการติดตามวัตถุสีได้ดีแต่ถ้ามีการรบกวนจากแสงภาพนอกทำให้เกิดการเปลี่ยนแปลงแสงที่ป้ายสีอย่างทันทีทันใดจะส่งผลให้เกิดการการติดตามที่ผิดพลาดได้ สำหรับการแก้ปัญหาดังกล่าวในอนาคตสามารถทำได้โดยใช้กระบวนการหาคำตอบที่ดีที่สุด (Optimization) ในการติดตามวัตถุสี

คำสำคัญ : รถเข็นสินค้าอัตโนมัติ การประยุกต์การประมวลผลภาพ การตรวจจับสี

E-mail Address : **kairoek.c@rmutr.ac.th**

ระยะเวลาโครงการ : ตุลาคม พ.ศ. 2556 ถึง กันยายน พ.ศ. 2557

Abstract

Code of project : Inno 005/2557

Project Name : Automatic shopping cart by using image processing

Researcher name : Asst. Prof. Kairoek Choeychuen, Ph.D. and Mr.Pakon Thuphiro

This research presents an application of color detection in image processing to make a shopping cart following the user such as elderly or who do not have the ability to use the shopping cart as normal people can use a shopping cart alone in the mall. The operation of automatic pushcart includes the following major sections. The camera gets the image to track the color-target plate and send digital data to each control unit which controls each motor by microcontroller to track the patch at the back of user. The control units make both wheels of the pushcart rotating in different directions. From experiments, pushcart can move in the desired direction that is forward, backward, turn left, or turn right respectively.

Plate tracking uses color-plate tracking that learns color from the plate on user. We use non-stationary Gaussian model for keeping plate color with some image noise including environmental lighting changing.

The results showed that the color-plate tracking using non-stationary Gaussian model can be used to track color plate but we might get error tracking in some cases such as rapid changing of environmental lighting. The rapid lighting changing causes of shifting from the model that can be solved by using an optimization technique for the future work.

Keywords: Automatic shopping cart, Application of image processing, Color detection

---

E-mail Address : **kairoek.c@rmutr.ac.th**

Period of project : October 2013 - September 2014

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
สัญลักษณ์และอักษรย่อ	ฌ
บทที่ 1 บทนำ	**1**
1. ความเป็นมาและความสำคัญของปัญหา	1
2. วัตถุประสงค์การวิจัย	1
3. ขอบเขตการวิจัย	2
4. ประโยชน์ที่คาดว่าจะได้รับ	2
5. นิยามศัพท์	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**4**
1. ภาพรวมของโครงงานวิจัย	3
2. ความรู้เบื้องต้นเกี่ยวกับ Image processing	4
3. การประมวลผลภาพแบบจุด	5
4. Visual C#	5
5. .NET Framework	5
6. ไวยากรณ์ C#	6
7. การกระทำการแบบเทรชโฮลด์ (Threshold)	7
8. Motion Detection	8
9. ระบบสี RGB	9
10. Digital Image Processing	9
11. Microcontroller	11
12 ภาษา C	12

สารบัญ (ต่อ)

		หน้า
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง (ต่อ)		
13	Web Camera	13
14	มอเตอร์ไฟฟ้ากระแสตรง	13
15	TTL (Transistor-Transistor Logic)	15
16	ขั้นตอนการตรวจจับสีและติดตามเป้าหมาย	17
บทที่ 3 ขั้นตอนการดำเนินงาน		**20**
1.	ออกแบบ Flow Chart การทำงานของรถเข็นอัตโนมัติฯ	21
2.	ออกแบบโปรแกรมประมวลผลภาพจากเป้าสี่เหลี่ยมและโปแกรมรับค่าจากการประมวลผลภาพ	22
3.	ออกแบบตัวรถเข็น	24
4.	ออกแบบวงจร MAX232 เพื่อรับข้อมูลจากโปรแกรมประมวลผลภาพ	25
5.	ออกแบบวงจรไมโครคอนโทรลเลอร์เพื่อรับข้อมูลจาก MAX232	25
6.	ออกแบบวงจรขับมอเตอร์เพื่อรับคำสั่งจากไมโครคอนโทรลเลอร์ไปขับมอเตอร์	26
7.	กัดลายวงจรทั้งหมดลงบนแผ่นปริ๊นท์	27
8.	จัดเตรียมอุปกรณ์ทั้งหมด	30
9.	จัดวางวงจรทั้งหมด	31
บทที่ 4 การทดลองและผลการทดลอง		**32**
1.	การทดสอบการทำงานของโปรแกรม Image Processing	32
2.	การทดสอบวงจรขับมอเตอร์	37
3.	การทดสอบความเร็วในการเดิน	38
4.	การทดสอบระยะที่โปรแกรมสามารถจับเป้าสีแดงได้	41
5.	การทดสอบการแบกน้ำหนักของรถเข็นอัตโนมัติ	52
บทที่ 5 สรุป อธิปรายผลและข้อเสนอแนะ		**55**
1.	สรุปผลการวิจัย	55

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุป อธิปรายผลและข้อเสนอแนะ (ต่อ)	
2. การอภิปรายผลและข้อเสนอแนะ	55
3. การนำไปใช้ประโยชน์	55
บรรณานุกรม	**57**
ภาคผนวก	**58**
ประวัติผู้จัดทำ	**72**

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

สารบัญตาราง

ตารางที่		หน้า
2-1	ตารางแสดงตำแหน่งขาและหน้าที่ของขาของ Connector DB9	16
4-1	ผลการทดสอบความเร็วในการเดิน	35

สารบัญภาพ

ภาพที่		หน้า
2-1	ภาพรวมการทำงานของรถเข็นอัตโนมัติ	4
2-2	การกำหนดค่าเทรชโฮลด์จากฮีสโทแกรมของภาพ	7
2-3	แสดงภาพ Frame Differencing	9
2-4	PIC16F877	11
2-5	Stator	13
2-6	Pole Core	14
2-7	ขั้วแม่เหล็ก	14
2-8	Rotor	14
2-9	การติดต่อสื่อสารระหว่าง MCU กับ Computer โดยใช้ MAX232	15
2-10	การต่อเชื่อมต่อใช้งาน MAX232	16
2-11	หัวเชื่อมต่อของ RS232 แบบ DB9	16
2-12	ขั้นตอนการตรวจจับสีและติดตามเป้าหมาย	18
2-13	ขั้นตอนการติดตามป้ายสี	19
3-1	Flow chart การทำงานเครื่องควบคุมรถเข็นอัตโนมัติ	21
3-2	Flow chart แสดงการทำงานชุดควบคุมการสั่งงานของ Software	22
3-3	การออกแบบรถเข็นโดยสังเขป	24
3-4	วงจร MAX232	25
3-5	วงจรไมโครคอนโทรลเลอร์	26
3-6	วงจรขับมอเตอร์	27
3-7	ลายวงจร MAX232	28
3-8	ลายวงจร MAX232 เมื่อลงแผ่นปริ๊นท์	28
3-9	ลายวงจรไมโครคอนโทรลเลอร์	28
3-10	ลายวงจรไมโครคอนโทรลเลอร์ เมื่อลงแผ่นปริ๊นท์	29
3-11	ลายวงจรขับมอเตอร์	29
3-12	ลายวงจรขับมอเตอร์เมื่อลงแผ่นปริ๊นท์	30

3-13 การจัดวางอุปกรณ์ทั้งหมด 31
4-1 การติดตั้งกล้อง 32
4-1 การเชื่อมต่อกล้องกับคอมพิวเตอร์ 33

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4-3 การเชื่อมต่อวงจร MAX232 กับคอมพิวเตอร์ 33
4-4 โปรแกรมประมวลผลภาพ 34
4-5 การนำเป้าสีแดงไว้ใกล้กล้องต่ำกว่าระยะที่กล้องสามารถจับได้ 34
4-6 การนำเป้าสีแดงไว้ไกลกล้องเกินระยะที่กล้องสามารถจับได้ 35
4-7 การนำเป้าสีแดงไว้ในระยะพอดี 35
4-8 การนำเป้าสีแดงไว้ทางด้านซ้ายของกล้อง 36
4-9 การนำเป้าสีแดงไว้ทางด้านขวาของกล้อง 36
4-10 การติดตั้งกล้อง 38
4-11 การเชื่อมต่อกล้องกับคอมพิวเตอร์ 39
4-12 การเชื่อมต่อวงจร MAX232 กับคอมพิวเตอร์ 39
4-13 แสดงการติดของไป LED เมื่อเปิดสวิทช์ 40
4-14 การติดตั้งอุปกรณ์โดยรวม 40
4-15 ไอคอนเพื่อใช้งานไดร์ฟเวอร์กล้อง 42
4-16 หน้าจอแสดงภาพที่ได้จากกล้อง 42
4-17 การเข้าไปตั้งค่าการทำงานของกล้อง 43
4-18 หน้าจอการตั้งค่าการทำงานของกล้องเมื่อปิดการใช้ค่าชดเชยแสงอัตโนมัติ 44
4-19 หน้าจอการตั้งค่าการทำงานของกล้องเมื่อเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 45
4-20 การนำเป้าสีแดงไว้ในระยะ 60 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 46
4-21 การนำเป้าสีแดงไว้ในระยะ 65 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 46
4-22 การนำเป้าสีแดงไว้ในระยะ 72 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 47
4-23 การนำเป้าสีแดงไว้ในระยะ 85 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 47
4-24 การนำเป้าสีแดงไว้ในระยะ 100 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 48
4-25 การนำเป้าสีแดงไว้ในระยะ 150 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 48
4-26 การนำเป้าสีแดงไว้ในระยะ 160 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ 48
4-27 การนำเป้าสีแดงไว้ในระยะ 80 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 49

4-28 การนำเป้าสีแดงไว้ในระยะ 85 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 49
4-29 การนำเป้าสีแดงไว้ในระยะ 105 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 50
4-30 การนำเป้าสีแดงไว้ในระยะ 120 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 50

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4-31 การนำเป้าสีแดงไว้ในระยะ 150 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 50
4-32 การนำเป้าสีแดงไว้ในระยะ 160 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 51
4-33 การนำเป้าสีแดงไว้ในระยะ 180 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 51
4-34 การนำเป้าสีแดงไว้ในระยะ 185 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ 52
4-35 การทดสอบการแบกน้ำหนัก 53
5-1 การทดสอบการทำงานของรถเข็นอัตโนมัติ 56
5-2 การทดสอบการทำงานของรถเข็น 56
5-3 การทดสอบการทำงานของรถเข็น 56

สัญลักษณ์และอักษรย่อ

nm	นาโนเมตร (nano-meter)
V	โวลต์(Volt)
μ	ไมโคร (10^{-6})
F	ฟารัด(Farhad)
k	กิโล (10^{3})
Ω	โอห์ม(Ohm)
p	พิโค (10^{-12})
A	แอมป์(Ampere)
W	วัตต์(Watt)
cm	เซนติเมตร(centi-meter)
USB	Universal Serial Bus
RGB	Red-Green-Blue
CRT	Cathode Ray Tube
CTS	Common Type System
CLR	Common Language Runtime
NTSC	National Television System Committee
CIE	Commission International ‘Eclairage’

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบัน ธุรกิจห้างสรรพสินค้าได้มีการเพิ่มจำนวนขึ้นอย่างมาก ทั้งในเขตเมืองหลวง และในเขตชนบท ซึ่งกว่าร้อยละ 90 เมื่อลูกค้าเข้ามาภายในห้างสรรพสินค้าเพื่อต้องการซื้อสินค้า ทางห้างสรรพสินค้าจะมีบริการรถเข็นใส่สิ้นค้าไว้เพื่อเป็นเครื่องทุ่นแรงเมื่อลูกค้าซื้อสินค้าจำนวนมาก หรือซื้อจำนวนน้อยแต่สินค้านั้นมีขนาดใหญ่ หรือมีน้ำหนักมาก โดยรถเข็นทุกคันจะมีวัตถุแข็งลักษณะเป็นแท่งยาวติดตั้งไว้ด้านหลังรถเข็น เพื่อใช้เป็นอุปกรณ์สำหรับเข็น ส่งผลให้รถเข็นเคลื่อนที่ไปข้างหน้าได้ ซึ่งตัวผู้เข็นก็สามารถทำการบังคับเลี้ยวไปทางซ้ายหรือขวาได้ในขณะเดียวกัน

รถเข็นใส่สินค้านั้นหากผู้ใช้เป็นคนที่อยู่ในวัยรุ่น ไปจนถึงวัยกลางคน และไม่พิการหรือทุพพลภาพ ก็จะสามารถเข็นได้โดยไม่มีปัญหา แต่กับผู้ใช้ที่เป็นเด็กซึ่งมีความสูงไม่มาก หรือผู้ใช้ที่เป็นผู้สูงอายุและพิการไม่สามารถใช้แขนได้ ก็อาจจะเป็นปัญหาใหญ่ได้ อีกทั้งหากลูกค้าผู้ใช้รถเข็นซื้อสินค้าที่มีน้ำหนักมาก หรือตัวสินค้ามีขนาดใหญ่จนทำให้บดบังวิสัยทัศน์ในการเข็น ก็อาจจะทำให้เกิดปัญหาขึ้นมาได้

ดังนั้นจึงเกิดแนวคิดในการประดิษฐ์รถเข็นสินค้าอัตโนมัติ โดยการใช้เทคโนโลยี Image processing ซึ่งจะประมวลผลภาพส่งไปยังมอเตอร์ทำให้รถเข็นสามารถเคลื่อนที่ติดตามลูกค้าผู้ใช้ได้โดยอัตโนมัติ ส่งผลให้ลูกค้าผู้ใช้ไม่จำเป็นต้องมาทำการเข็นด้วยตนเอง ทั้งนี้หากลูกค้าผู้ใช้รถเข็นต้องการที่จะทำการเข็นด้วยตนเอง สามารถทำการปิดระบบอัตโนมัติและเข็นด้วยตนเองได้เช่นเดียวกัน อีกทั้งในอนาคตยังสามารถนำรถเข็นอัตโนมัตินี้ไปทำการพัฒนาต่อเพื่อให้มีประสิทธิภาพและความสามารถมากขึ้นหรือนำเทคโนโลยีบางส่วนไปพัฒนาใช้ในการสร้างหุ่นยนต์เดินตามคน

โดยภาพรวมการทำงานของรถเข็นอัตโนมัติ สามารถอธิบายได้อย่างคร่าวๆ ดังนี้ เมื่อกล้องรับภาพจากเป้าสี่เหลี่ยมโดยในที่นี้ผู้จัดทำได้ใช้แผ่นฟิวเจอร์บอร์ดสีแดงมาตัดเป็นรูปสี่เหลี่ยม ก็จะส่งภาพที่รับได้มาทำการประมวลผลโดยโปรแกรม Visual C# โดยใช้โปรแกรมย่อยที่เรียกว่า OpenCVSharp ซึ่งเป็นกลุ่มของไฟล์ประมวลผลหรือที่เรียกกันทั่วไปว่า Library เป็นไฟล์เฉพาะสำหรับใช้ประมวลผลใน Visual C# โดยการประมวลผล จะทำการเปลี่ยนเป้าสี่เหลี่ยมสีแดงเป็นสีขาวและพื้นหลังเป็นสีดำหรือภาพไบนารี จากนั้นโปรแกรมจะประมวลผลตำแหน่งของสี่เหลี่ยมสีขาวให้เป็นข้อมูลค่าต่างๆกันและส่งข้อมูลเหล่านั้นออกไปทางพอร์ตอนุกรมที่ได้ตั้งค่าไว้ ผ่าน IC MAX232 ซึ่งจะส่งข้อมูลต่อไปให้

ไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะนำข้อมูลเหล่านั้นมาแปลงเป็นค่าไบนารีเพื่อนำไปขับมอเตอร์ต่อไป

2. วัตถุประสงค์การวิจัย

1. เพื่อศึกษาการทำงานรวมกันของโปรแกรม C# และ Image Processing
2. เพื่อศึกษาการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ที่รับค่าจากการประมวลผลทางภาพ
3. เพื่อสร้างรถเข็นอัตโนมัติที่รับค่าจากการประมวลผลจากภาพ
4. เพื่อพัฒนาโมดูลติดตามคนจากการประมวลผลภาพที่มีการประมวลผลไม่ซับซ้อน

3. ขอบเขตการวิจัย

1. รถเข็นสามารถเคลื่อนที่ตามผู้ใช้ได้โดยอัตโนมัติ โดยใช้ระบบ Image processing
2. รถเข็นสามารถเคลื่อนที่ตามผู้ใช้ได้ในระยะไม่เกิน 1 เมตร และไม่น้อยกว่า 30 เซนติเมตร
3. รถเข็นต้องสามารถบรรทุกน้ำหนักได้ใกล้เคียงกับรถเข็นแบบเดิม โดยยังสามารถเคลื่อนที่ไปได้
4. ระบบ Image processing สามารถเปิดและปิดได้ตามต้องการ
5. รถเข็นสามารถใช้พลังงานจากแบตเตอรี่ ที่ติดตั้งอยู่กับรถเข็น

4. ประโยชน์ที่คาดว่าจะได้รับ

1. นวัตกรรมซึ่งสามารถนำไปใช้ประชาสัมพันธ์ห้างสรรพสินค้า
2. เครื่องมือช่วยอำนวยความสะดวกในการซื้อสิ้นค้าให้กับคนปกติและผู้พิการแขนข้างเดียว
3. ต้นแบบในการสร้างหุ่นยนต์ติดตามคน

5. นิยามศัพท์

Non-stationary Gaussian model หมายถึง แบบจำลองชนิดหนึ่งซึ่งพัฒนามาจากแบบจำลอง Gaussian มีลักษณะที่สามารถปรับรูปร่างแบบจำลองได้อย่างไดนามิคตามข้อมูลใหม่ที่รับเข้ามา

Color plate หมายถึงแผ่นป้ายที่อยู่กับผู้ใช้รถเข็นสินค้าซึ่งทำหน้าที่เป็นเป้าสำหรับให้รถติดตามผู้ใช้ตามสีที่ได้โปรแกรมไว้

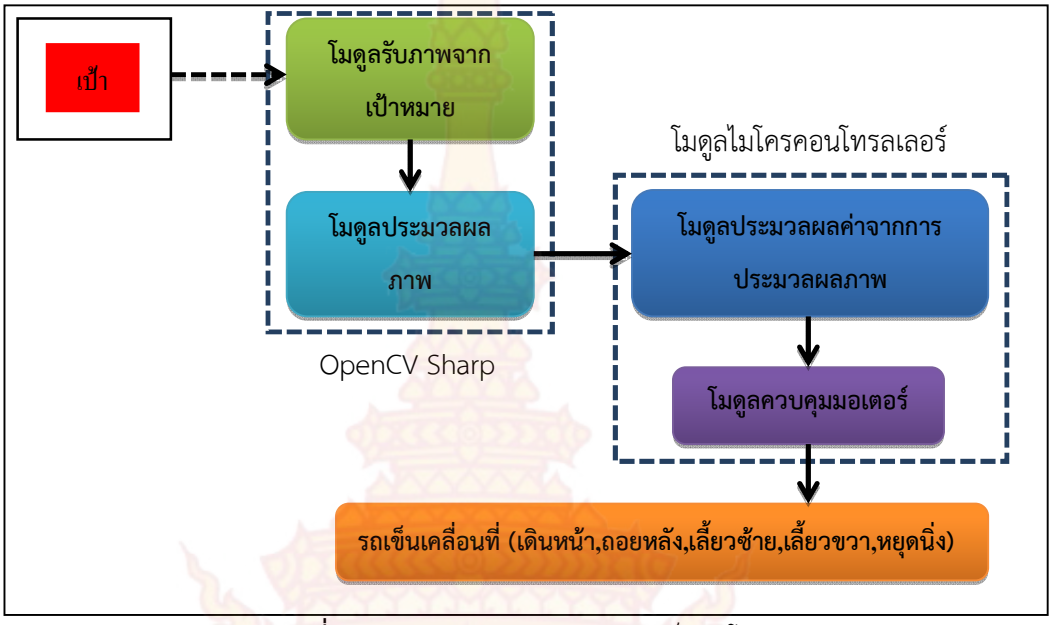
Shopping cart หมายถึงรถเข็นสินค้าที่ใช้ทั่วไปตามห้างสรรพสินค้า

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

โครงงานวิจัยรถเข็นอัตโนมัติพร้อมระบบติดตามวัตถุโดยการประมวลผลด้วยภาพ (Automatic shopping cart with tracking system by image processing) ได้ใช้เทคโนโลยี Image Processing และเทคโนโลยีไมโครคอนโทรลเลอร์ ในการประมวลผลและควบคุม จึงมีหลักการที่เกี่ยวข้องดังนี้

1. ภาพรวมของโครงงานวิจัย

เมื่อกล้องรับภาพจากเป้าสี่เหลี่ยมโดยในที่นี้ผู้จัดทำได้ใช้แผ่นฟิวเจอร์บอร์ดสีแดงมาตัดเป็นรูปสี่เหลี่ยม ก็จะส่งภาพที่รับได้มาทำการประมวลผลโดยโปรแกรม Visual C# โดยใช้โปรแกรมย่อยที่เรียกว่า OpenCVSharp ซึ่งเป็นกลุ่มของไฟล์ประมวลผลหรือที่เรียกกันทั่วไปว่า Library เป็นไฟล์เฉพาะสำหรับใช้ประมวลผลใน Visual C# โดยการประมวลผล จะทำการเปลี่ยนเป้าสี่เหลี่ยมสีแดงเป็นสีขาวและพื้นหลังเป็นสีดำหรือภาพไบนารี จากนั้นโปรแกรมจะประมวลผลตำแหน่งของสี่เหลี่ยมสีขาวให้เป็นข้อมูลค่าต่างๆกันและส่งข้อมูลเหล่านั้นออกไปทางพอร์ตอนุกรมที่ได้ตั้งค่าไว้ ผ่าน IC MAX232 ซึ่งจะส่งข้อมูลต่อไปให้ไมโครคอนโทรลเลอร์จากนั้นไมโครคอนโทรลเลอร์จะนำข้อมูลเหล่านั้นมาแปลงเป็นค่าไบนารีเพื่อนำไปขับมอเตอร์ต่อไปซึ่งภาพรวมของโครงงานวิจัยสามารถแสดงได้ดังภาพที่ 1



ภาพที่ 2-1 ภาพรวมการทำงานของรถเข็นอัตโนมัติ

2. ความรู้เบื้องต้นเกี่ยวกับ Image Processing

Image Processing คือ การกระทำการอย่างใดอย่างหนึ่งกับภาพต้นฉบับ (Input Image) เพื่อให้ได้ภาพผลลัพธ์ (Output Image) มีลักษณะของภาพเป็นไปตามที่ต้องการ ซึ่งการกระทำการกับภาพที่ใช้ในการประมวลผลภาพดิจิตอลมีอยู่มากมายหลายแบบ ซึ่งสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆคือ การประมวลผลภาพแบบจุด (Point Image Processing) และการประมวลผลภาพแบบบริเวณ (Local Image Processing) โดยในโครงงานวิจัยชิ้นนี้จะใช้การประมวลผลภาพแบบจุดมาวิเคราะห์ตำแหน่งและค่าของแต่ละจุดสีในภาพต้นฉบับ แล้วนำมาประมวลผลเพื่อเปลี่ยนเป็นภาพขาวดำดังที่ได้กล่าวในภาพรวมของโครงงานวิจัย

2.1 ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)

ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซลจะมีค่าความเข้มของแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่ระดับเทาดำไปยังระดับสีขาว ซึ่งสามารถกำหนดระดับความเข้มของแสงนั้นได้โดยใช้ค่าระดับความเข้มเทา (Gray Scale) โดยปกติทั่วไปภาพแบบระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับเมื่อค่าระดับความเข้มเทามีค่าเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำจะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเทาเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงมาก จะทำให้จุดเป็นสีขาว ซึ่งสีขาวจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 255 และสีดำจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 0 ส่วนค่าระหว่าง 0-255 ก็จะมีค่าไล่เฉดสีดำไปหาสีขาว

2.2 ภาพสี (Color Image)

ภาพชนิดนี้ แต่ละจุดภาพหรือพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง สีเขียว และสีน้ำเงิน ซึ่งในแต่ละพิกเซลนั้นๆ ก็จะแสดงผลของค่าสีของแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบแสงสีนั้น

2.3 ภาพไบนารี (Binary Image)

ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบไบนารี คือมี 1 บิต ซึ่งประกอบไปด้วยค่า 1 และ 0 โดยที่ 1 หมายถึงจุดภาพสีขาว และ 0 หมายถึงจุดภาพสีดำ

3. การประมวลผลภาพแบบจุด

การประมวลผลภาพแบบจุดเป็นวิธีการกระทำการกับภาพต้นฉบับที่ค่าระดับความเข้มเทาที่แสดงในแต่ละพิกเซลของภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของพิกเซลในต้นฉบับพิกเซลต่อพิกเซลที่ตำแหน่งสมนัยกันโดยที่ค่าเปลี่ยนแปลงของพิกเซลของภาพผลลัพธ์ไม่ได้ขึ้นอยู่กับค่าพิกเซลที่อยู่บริเวณใกล้เคียงของภาพต้นแบบ

4. Visual C#

ภาษาซีชาร์ป(C# Programming Language) เป็นภาษาโปรแกรมเชิงวัตถุ ที่พัฒนาโดยบริษัทไมโครซอฟท์ และมี Anders Hejlsberg เป็นหัวหน้าโครงการ เพื่อเป็นส่วนหนึ่งของ ดอทเน็ตเฟรมเวิร์ค (. Net Framework) โดยมีรากฐานมาจากภาษา ซีพลัสพลัส (C++) และภาษาจาวา (Java)ซีชาร์ป เป็นภาษาใหม่ที่ทางไมโครซอฟท์ได้พัฒนาขึ้นมาพร้อมกับโปรแกรมชุด วิชวลสตูดิโอดอทเน็ต (Visual Studio .NET) โดยวัตถุประสงค์หลักของการพัฒนาคือ เพื่อให้เป็นภาษาใหม่ที่มีประสิทธิภาพการทำงานเทียบเท่าหรือเหนือกว่าภาษา ซีพลัสพลัส (C++) แต่ในขณะเดียวกันจะต้องไม่ยุ่งยากและซับซ้อนเหมือนกับภาษา ซีพลัสพลัส โดยสามารถใช้งานได้ง่ายๆ เหมือนกับภาษายอดนิยมอย่างวิชวลเบสิก (Visual Basic) และสามารถขยายขีดความสามารถของวิชวลเบสิกได้ ให้สามารถพัฒนาโปรแกรมประยุกต์ในระดับสูงได้ดียิ่งขึ้น ด้วยเหตุผลเหล่านี้ทาให้ซีชาร์ป เป็นภาษาที่มีความลงตัวมากที่สุดเมื่อเทียบกับภาษาอื่นๆ ซึ่งภาษาซีชาร์ปยังเป็นภาษาที่เกิดขึ้นพร้อมกับแนวคิดของการเขียนโปรแกรมในยุค ดอทเน็ต อีกด้วย

5. .NET Framework

ดอทเน็ตเฟรมเวิร์ค (.Net Framework) เป็นแพลตฟอร์มใหม่ และเปลี่ยนแปลงไปอย่างสิ้นเชิง ถูกสร้างขึ้นมาโดย บริษัทไมโครซอฟท์ เพื่อใช้สาหรับ การพัฒนาโปรแกรมประยุกต์ ดอทเน็ต

เฟรมเวิร์คถูกออกแบบมาเพื่อให้สามารถ ถูกใช้จากภาษาใดๆ ก็ได้ รวมถึงซีชาร์ป ด้วย เพื่อให้สิ่งเหล่านี้เป็นไปได้ จึงเกิดภาษาเหล่านี้ขึ้นมาในรูปแบบเฉพาะ สาหรับ ดอทเน็ต อีกด้วย ได้แก่ ภาษาแมนเนจซีพลัสพลัส (Managed C++), วิชวลเบสิกดอทเน็ต (Visual Basic.Net), บอร์แลนด์ซีชาร์ป (Borland C#), เดลไฟล์เวอร์ชันแปด (Delphi8) เป็นต้น และมีอีกมากที่กาลังพัฒนา และปล่อยออกสู่ท้องตลาดอยู่ตลอดเวลา ไม่ใช่เพียงแค่ภาษาทั้งหมดเหล่านี้ จะมีการเข้าถึง ดอทเน็ตเฟรมเวิร์ค (.Net Framework) เท่านั้น แต่มันยังสามารถสื่อสาร กับภาษาอื่น ๆ ได้อีกด้วยดอทเน็ตเฟรมเวิร์ค พื้นฐานประกอบขึ้นด้วยไลบรารี (Library) ของซอร์สโค้ด (Source Code) ขนาดมหึมา ซึ่งเราเรียกใช้จากภาษาไคลเอนท์ (Client) ของเรา เช่น ซีชาร์ปดอทเน็ต, ซีพลัสพลัสดอทเน็ต โดยการใช้เทคนิคเชิงวัตถุ (OOP) ไลบรารีที่ว่านี้ ถูกแบ่งกลุ่มออกเป็นโมดูลต่าง ๆ ดังนั้นเราจึงใช้ส่วนของมันตามผลลัพธ์ที่เราต้องการได้ เช่น วินส์โดว์แอพพลิเคชัน (Windows Application) เป็นต้น จุดมุ่งหมายในที่นี้ก็คือ ระบบปฏิบัติการ ที่แตกต่างกัน อาจจะสนับสนุนโมดูลเหล่านี้ บางโมดูลหรือทั้งหมด ขึ้นอยู่กับคุณลักษณะของมัน ส่วนไลบรารี ดอทเน็ตเฟรมเวิร์ค กำหนดชนิด ข้อมูลพื้นฐานบางอย่างเอาไว้ ชนิดข้อมูลเป็นตัวแทนของข้อมูล และการแบ่งกฎเกณฑ์ทั้งหลายเหล่านี้ ที่จะส่งเสริมความสามารถ ในการสัมพันธ์ระหว่างภาษา โดยใช้ดอทเน็ตเฟรมเวิร์ค สิ่งนี้ถูกเรียกว่าคอมมอนไทป์ซิสเต็ม (CTS) เช่นเดียวกับการจัดให้มีไลบรารี ดอทเน็ต คอมมอนแลงกูเอจรันไทม์ (CLR) ซึ่งรับผิดชอบในการจัดการ กับ ระบบปฏิบัติการ ของแอพพลิเคชันทั้งหมดที่ถูกพัฒนาขึ้นมาด้วย ไลบรารีดอทเน็ตเฟรมเวิร์ค

6. ไวยากรณ์ C#

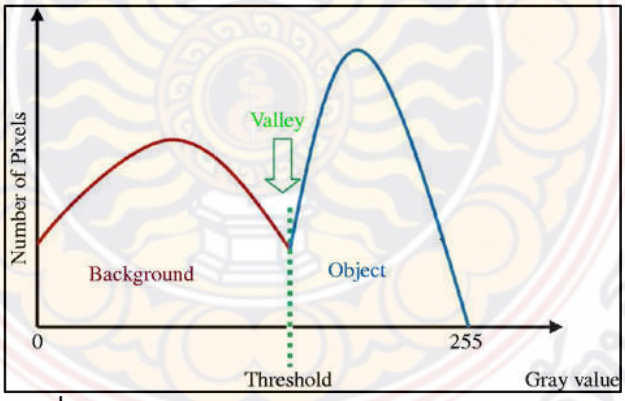
ความคล้ายคลึงกัน ของโค้ด ของซีชาร์ป กับ ซีพลัสพลัส (C++) และ จาวา (Java) นั้นอาจดูค่อนข้างสับสน และมันก็คล้ายกับการเขียนภาษาอังกฤษ น้อยกว่าภาษาอื่น ๆ มาก แต่อย่างไรก็ตามเราจะพบว่า ตัวเราเองอยู่ในโลกของการโปรแกรมซีชาร์ป (C#) ที่มีสไตล์ อ่อนไหวนุ่มนวล และสามารถอ่านโค้ดได้ง่าย และไม่สับสนมากนักซีชาร์ป (C#)นั้นไม่มีการแจ้งเตือน เกี่ยวกับช่องว่าง ที่อยู่ในโค้ดไม่ว่าคุณจะใส่ช่องว่าง จำนวนมากอักขระแคริเอจรีเทิร์น (Carriage return) หรือแทป (Tap) เป็นอักขระที่รู้จักกันในชื่อว่าไวท์เสปซ (White space)นั่นหมายความว่า เรามีอิสระในการที่จะจัดรูปแบบซอร์สโค้ด (Source Code) ของเราได้ ถึงแม้ว่าการทำตามกฎที่แน่นอน สามารถช่วยให้เราทำสิ่งต่าง ๆ ให้อ่านได้ง่ายขึ้นก็ตามโค้ดของซีชาร์ปนั้นสร้างขึ้นจาก คำสั่ง (Statement) ชุดหนึ่ง แต่ละคำสั่งจะจบด้วยเครื่องหมายอัฒภาค (Semicolon ;) เนื่องจากไวท์สเปซ นั้นถูกมองข้ามไป เราจึงสามารถมีหลาย ๆ คำสั่งในบรรทัดเดียวกันได้ แต่เพื่อความง่ายในการอ่าน มันจะมีประโยชน์ถ้าเพิ่มแคริเอจรีเทิร์น (Carriage return) เข้าไปหลังเครื่องหมายอัฒภาค (;) ดังนั้นเราก็จะไม่มีหลายๆ คำสั่งในบรรทัดเดียวกันอย่างไรก็ตามมันเป็นสิ่งที่ยอมรับได้และเป็นสิ่งปกติซีชาร์ป (C#) เป็นภาษา

โครงสร้างบล๊อก (Block structured) หมายความว่าทุกๆคำสั่งถือเป็นส่วนหนึ่งของบล็อกโค้ด (Block Code) และบล็อกเหล่านี้ถูกจำกัดด้วยเครื่องหมาย { } เหมือนกับซีพลัสพลัส (C++) ทุกประการ รูปแบบจึงเหมือนกับซีพลัสพลัส (C++) และมีโครงสร้างเหมือนซีพลัสพลัส(C++) ดังนั้นหากเราได้ศึกษาโครงสร้างของซีพลัสพลัส (C++) แล้วก็จะเข้าใจหากยังไม่เข้าใจให้ไปศึกษาโครงสร้างในซีพลัสพลัส (C++)

7. การกระทำการแบบเทรชโฮลด์ (Threshold)

เทคนิคการกระทำการแบบเทรชโฮลด์เป็นวิธีการหรือเทคนิคการประมวลผลภาพอย่างง่ายๆ เพื่อที่จะแบ่งแยกส่วนพื้นหน้า (foreground) หรือวัตถุ (Object) ออกจากพื้นหลัง (Background) โดยใช้ค่าระดับเทรชโฮลด์(T) หรือค่าระดับความเข้มเทาคงที่ค่าหนึ่งเป็นตัวกำหนดในการแยกแยะส่วนของภาพ เพื่อทำให้ภาพผลลัพธ์ที่ได้เป็นภาพแบบไบนารี (Binary) ที่มีค่าระดับความเข้มเทาเพียง 2 ระดับเท่านั้นคือขาวและดำ ซึ่งในการกำหนดค่าเทรชโฮลด์นั้น ถ้าค่าเทรชโฮลด์ที่เรากำหนดไว้มีค่าไม่เหมาะสม เช่น ค่าเทรชโฮลด์ที่มีค่าน้อยหรือมากเกินไป อาจทำให้รายละเอียดบางส่วนของภาพวัตถุที่ต้องการขาดหายไป หรือภาพอาจจะมีสิ่งไม่พึงประสงค์ปนมาด้วย เช่น สัญญาณรบกวน (Noise) ดังนั้นเราจะต้องมีการกำหนดค่าเทรชโฮลด์ที่เหมาะสม ซึ่งในปัจจุบันมีผู้เสนอวิธีในการหาค่าเทรชโฮลด์หลายวิธี ซึ่งแต่ละวิธีจะถูกนำไปใช้ในงานที่มีลักษณะแตกต่างกันออกไปตามเงื่อนไขของภาพต้นแบบและวิธีการออกแบบ

การกำหนดค่าเทรชโฮลด์หนึ่งระดับ (Single Threshold) เป็นวิธีการแยกข้อมูลระหว่างสิ่งที่ต้องการหรือวัตถุออกจากพื้นหลังของภาพ โดยพิจารณาจากฮีสโทแกรมของภาพ ซึ่งกลุ่มของข้อมูลทั้งสองจะแยกออกเป็นสองกลุ่มตามการกระจายของข้อมูล ดังนั้นการเลือกค่าเทรชโฮลด์จะต้องเลือกค่าที่อยู่ระหว่างกลุ่มข้อมูลทั้งสอง ดังภาพที่ 2 จะเห็นได้ว่าค่าเทรชโฮลด์ที่กำหนดจะเป็นจุดเริ่มต้นแบ่งเขตเพื่อให้ได้วัตถุหรือสิ่งที่ต้องการออกจากพื้นหลังของภาพต้นแบบ



ภาพที่ 2-2 การกำหนดค่าเทรชโฮลด์จากฮีสโทแกรมของภาพ

วิธีการกระทำแบบเทรชโฮลด์นั้น ขั้นตอนแรกต้องทำการกำหนดค่าเทรชโฮลด์ (T) ที่ค่าคงที่ค่าหนึ่ง หรือค่าที่ทำให้เกิดจุดเปลี่ยนแปลงของภาพต้นฉบับ ถ้าหากค่าระดับเทาของจุดภาพต้นฉบับมีค่าต่ำกว่าระดับค่าเทรชโฮลด์ที่กำหนด จุดภาพนั้นจะถูกกำหนดค่าใหม่ให้เป็นศูนย์ (จุดภาพผลลัพธ์จะมีสีดำ) ในทางตรงกันข้าม หากค่าระดับเทาของจุดภาพต้นฉบับมีค่าสูงกว่าระดับค่าเทรชโฮลด์ จุดภาพนั้นจะถูกกำหนดค่าใหม่ให้มีค่าเป็น 255 (จุดภาพผลลัพธ์จะมีสีขาว) ในกรณีที่ต้องทำให้ภาพผลลัพธ์ที่เป็นภาพแบบไบนารี เราสามารถกำหนดค่าใหม่ของภาพผลลัพธ์ให้มีค่าเป็น 1 ซึ่งสามารถเขียนแทนสมการในการทำเทรชโฮลด์ได้ดังนี้

$$y = \begin{cases} 0, & x \leq T \\ 1, & x > T \end{cases} \qquad (2\text{-}1)$$

เมื่อ x คือค่าระดับความเข้มเทาของแต่ละจุดภาพต้นฉบับ
y คือค่าระดับความเข้มเทาที่ผ่านการทำเทรชโฮลด์
T คือค่าเทรชโฮลด์ที่ทำให้เกิดการเปลี่ยนแปลงของภาพ

8. Motion Detection

เป็นการกระทำเพื่อตรวจหาการเคลื่อนไหวระดับทางกายภาพโดยการเคลื่อนที่หรือเคลื่อนไหวของวัตถุสามารถตรวจพบได้เมื่อมีการเปลี่ยนแปลงความเร็วของวัตถุ กล่าวคือ ระบบจะมีการตรวจจับการเคลื่อนไหวตลอดเวลาโดยจะมีการวนรอบการทำงานเรื่อยๆเพื่อนำข้อมูลภาพใหม่ที่ได้รับจากส่วนติดต่อกล้องมาทำการประมวลผล ตรวจหาความแตกต่างของ pixel ระหว่างภาพใหม่กับภาพเก่าว่ามีความแตกต่างกันมากเกินค่าที่ตั้งไว้หรือไม่ ถ้ามีความแตกต่างกันมาก ระบบสามารถรับรู้ได้ว่ามีการเคลื่อนไหวเกิดขึ้น

ซึ่งในการตรวจหาการเคลื่อนที่ จะใช้วิธีการตรวจหาการเคลื่อนที่ของคน ซึ่งจะใช้การหาความแตกต่างระหว่างจุดภาพ โดยมีหลักการ คือ เฟรมภาพแต่ละภาพจะถูกนำไปเปรียบเทียบกับภาพอ้างอิงทุกจุดภาพต่อจุดภาพ ถ้ามีความแตกต่างที่จุดภาพหนึ่งมีค่ามากกว่าค่าขีดเริ่มเปลี่ยน

ในการทำงานจะเริ่มเมื่อทำการรับภาพจากกล้องแล้วแปลงเป็นโมเดลสี RGB ภาพแรกจะถูกเก็บไว้ใช้เป็นภาพอ้างอิง แล้วนำภาพถัดไปมาเปรียบเทียบกับภาพอ้างอิง ถ้าพบว่าค่าผลต่างของภาพที่เปรียบเทียบกันนั้นมากกว่า Threshold จะให้จุดภาพนั้นเป็นสีขาวนั่นก็คือวัตถุ แต่ถ้าผลต่างน้อยกว่า Threshold จะให้จุดภาพนั้นเป็นสีดำซึ่งเป็นฉากหลังนั่นเองทุกภาพที่รับมาจะถูกเก็บไว้เป็นภาพอ้างอิงเพื่อใช้ในการเปรียบเทียบในครั้งถัดไป โดยกระบวนการนี้จะเรียกว่า Frame Differencing ดังภาพที่ 3

สำหรับการเคลื่อนไหวของวัตถุว่ามีการเคลื่อนไหวมากหรือน้อยนั้นสามารถนำขั้นตอนของ FrameDifferencingมาทำการวิเคราะห์จุดพิกเซลโดยการนำจำนวนพิกเซลมาพล็อตเป็นกราฟ กล่าวคือ นำแถวในแต่ละแถวมาบวกกัน และนำคอลัมน์ของแต่ละคอลัมน์มาบวกกัน แล้วนำผลบวกที่ได้มาพล็อตบนกราฟ

ภาพที่ 2-3 แสดงภาพ Frame Differencing

9. ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและ น้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แก่ RGB_{CIE} และ RGB_{NTSC}

9.1 ระบบสีแบบ RGB ของ CIE

เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

9.2 ระบบสีแบบ RGB ของ NTSC

เป็นระบบสีที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพแบบ CRT เป็นมาตรฐานสำหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

10. Digital Image Processing

ภาพโดยทั่วไปที่ได้จากการบันทึกภาพไม่ว่าจะบันทึกด้วยฟิล์ม หรือบันทึกด้วยกล้องดิจิตอลอาจจะได้ ภาพที่มีคุณภาพต่ำ ดังนั้นจึงต้องทำภาพนั้นๆ ให้ให้เป็นภาพดิจิตอล แล้วนำมาทำการประมวลผลภาพ เพื่อทำให้ภาพนั้นได้คุณภาพตามที่ต้องการ การประมวลผลดิจิตอล เป็นการนำภาพดิจิตอลมาปรับปรุงสามารถกระทำได้หลากหลายวิธีการ เช่น การเพิ่มความคมชัดของภาพ การบูรณะภาพ เป็นต้น ซึ่งการกระทำของภาพสามารถกระทำได้สองลักษณะคือ การกระทำเชิงตำแหน่ง และการกระทำเชิงความถี่ การกระทำทั้งสองลักษณะนี้ให้คุณภาพที่ใกล้เคียงกัน แต่แตกต่างกันในเรื่องของเวลาที่ใช้ในการประมวลผล

Digital images หรือภาพที่ถูกเก็บในรูปแบบดิจิตอลนั้นสามารถเก็บได้หลายรูปแบบด้วยกัน หนึ่งในการเก็บที่นิยมที่สุดเราเก็บในรูปแบบของภาพRGBหรือการเก็บเป็นเฉดสี 3เฉด RGB หรือบางทีเรียกกันง่ายๆว่าสีแสงก็คือสีที่เรามองเห็นได้เพราะวัตถุได้ทำการดูดกลืนหรือหักเหแสงบางส่วนออกไปจากแสงขาวที่ส่องกระทบวัตถุนั้นแล้วสะท้อนมาเข้าตาเรา ทำให้เราเห็นแสงในระดับต่างๆออกมาเป็นสีๆ โดยRGBก็จะแยกออกตามพลังงานของแสงแบ่งได้ดังนี้ คลื่นแสงสีแดงจะอยู่ในช่วงคลื่นที่มีพลังงานต่ำสุด คือมีความยาวคลื่นอยู่ระหว่าง 630-760นาโนเมตร(ถ้าคลื่นแสงมีพลังงานน้อยกว่าสีแดงเราเรียกว่า อินฟราเรด)ซึ่งปกติจะพบว่าแสงในช่วงคลื่นอินฟราเรดจะไม่สามารถทะลุผ่านวัตถุได้ คลื่นแสงสีเขียวเป็นคลื่นที่มีความยาวคลื่นอยู่ระหว่างแสงสีแดงและสีน้ำเงินคือระหว่าง 520-570นาโนเมตร คลื่นแสงสีน้ำเงิน(หรือสีม่วง)เป็นแสงที่มีพลังงานสูงที่สุด มีความยาวคลื่นอยู่ระหว่าง 440-490 นาโนเมตร(ถ้าคลื่นแสงที่มีมากกว่าแสงสีม่วงเราเรียกว่าอุลตร้าไวโอเล็ดถ้าสูงกว่าคลื่นแสงอุลตร้าไวโอเล็ดก็จะมีรังสีเอ็กซ์และรังสีแกมม่าตามลำดับซึ่งคลื่นแสงเหล่านี้มีพลังงานมาก มีความสามารถทะลุทะลวงสูง และยังทำอันตรายต่อผู้ที่สัมผัสกับมันอีกด้วย กลับมาที่เรื่องของภาพดิจิตอล ภาพดิจิตอลที่เก็บค่าสีแบบRGBก็แบ่งระดับสีต่างกันตามหน่วยความจำ 32bits(True Color), 16bits(High Color) และ 8bits

11. ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ประเภทสารกึ่งตัวนำที่รวบรวมฟังก์ชั่นการทำงานต่างๆ ไว้ภายในตัวของมันเอง โดยมีโครงสร้างใกล้เคียงกับคอมพิวเตอร์ คือ ภายในประกอบด้วยหน่วยรับข้อมูลและโปรแกรมหน่วยประมวลผล หน่วยความจำ หน่วยแสดงผล ซึ่งส่วนประกอบเหล่านี้มีความสมบูรณ์ในตัวของมันเอง ทำให้มีขนาดเล็ก และสามารถเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ต่างๆ ที่เชื่อมต่อกับตัวมัน ง่ายต่อการนำไปประยุกต์ใช้งานไมโครคอนโทรลเลอร์ มาจากคำ 2 คำ คำหนึ่งคือ ไมโคร (Micro)หมายถึงขนาดเล็กและคำว่าคอนโทรลเลอร์ (controller)หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์ จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็กแต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ ที่คนโดยส่วนใหญ่ค้นเคย กล่าวคือภายใน ไมโครคอนโทรลเลอร์ ได้รวมเอาซีพียู,หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน ความแตกต่างของไมโครคอนโทรลเลอร์ กับ ไมโครคอมพิวเตอร์คือ ไมโครคอนโทรลเลอร์นั้นมีความสมบูรณ์ภายในตัวของมันเอง คือ มีส่วนประกอบต่างๆ ครบถ้วน ส่วนไมโครคอมพิวเตอร์นั้นต้องทำงานร่วมกับอุปกรณ์ข้างเคียงที่เชื่อมต่อจากภายนอก เช่น แป้นพิมพ์ เครื่องอ่านเขียนแผ่นบันทึก หน่วยความจำ I/O ฯลฯโครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

ภาพที่ 2-4 PIC16F877

11.1 หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)ทำหน้าที่ประมวลผลคำสั่งต่างๆ

11.2 หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือหน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory)ใช้เป็นเหมือนกกระดาษทดในการคำนวณของซีพียูและเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความแรมในเครื่องคอมพิวเตอร์ทั่วๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยงและเป็นอีอีพรอม (EEPROM: Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

11.3 ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุตและพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุตส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

11.4 ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล, บัสแอดเดรส และบัสควบคุม

11.5 วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามี

ความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

12. ภาษา C

C Language (ภาษาซี) คือ ซึ่งเป็นภาษาคอมพิวเตอร์ที่ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาโดยเดนนิสริสชี่ (Dennis Ritchie)เมื่อประมาณต้นปีค.ศ.1970 เพื่อใช้งานบนระบบปฏิบัติการยูนิกส์ แทนภาษาแอสเซมบลีซึ่งเป็นภาษาระดับต่ำที่สามารถกระทำในระบบฮาร์ดแวร์ได้ด้วยความรวดเร็ว แต่จุดอ่อนของภาษาแอซเซมบลีก็คือความยุ่งยากในการโปรแกรม ความเป็นเฉพาะตัว และความแตกต่างกันไปในแต่ละเครื่อง ต่อมาถูกนำไปใช้ในระบบปฏิบัติการต่าง ๆ จนถูกใช้เป็นภาษาพื้นฐานสำหรับภาษาอื่น เช่น ภาษาจาวา (Java), ภาษาพีเอชพี (PHP), ภาษาซีชาร์ป(C#), ภาษาซีพลัสพลัส(C++), ภาษาเพิร์ล (Perl), ภาษาไพทอน (Python), หรือภาษารูบี้ (Ruby) เป็นต้น ภาษาซีเป็นภาษาเขียนโปรแกรมระบบเชิงคำสั่ง (หรือเชิงกระบวนงาน) ถูกออกแบบขึ้นเพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง ภาษา C แม้จะเป็นภาษาระดับสูง แต่ก็สามารถใช้เป็นภาษาเครื่องได้เป็นอย่างดี

ข้อดีของ ภาษาซี

1. ภาษาซี สามารถนำไปใช้ได้บนเครื่องทุก platform ไม่ว่าจะเป็น Intel PC ที่วิ่ง Windows 95 หรือ Windows NT หรือ แม้แต่ Linuxทั้งเครื่อง Macintosh และ เครื่องเวอร์คสเตชัน ตลอดจนเมนเฟรม เนื่องจากมี compiler ของภาษาซี อยู่ทั่วไป

2. ภาษาซี เป็นภาษาที่ง่ายๆคือมีแต่ข้อกำหนดในการใช้งานหรือ Syntaxแต่ไม่มีฟังก์ชันสำเร็จรูปใดๆดังนั้นหากผู้ใช้ต้องการทำอะไรก็ตามต้องเขียนทุกอย่างขึ้นเองหรืออาจเรียกLibrary Functions มาใช้งาน โดย ฟังก์ชันที่เป็นงานที่ใช้บ่อยๆ จะถูกรวบรวมไว้ใน Library Functions เช่น การจัดการข้อความ การดำเนินการเกี่ยวกับ Input/output (I/O) การจองหน่วยความจำ (Memory Allocation) แต่ฟังก์ชันที่หรูหรา จะไม่มีใน Standard Library เช่น ฟังก์ชันที่จัดการ Graphics ทั้งนี้จะขึ้นกับระบบที่ใช้ (เช่น เป็นระบบ UNIX หรือ Windows 95) และ สิ่งแวดล้อมในการทำงาน (เช่น GUI เป็น X-Windows หรือ Direct X) การทำเช่นนี้จะทำให้ภาษาซี เป็นภาษาที่เคลื่อนย้ายได้ง่าย (portable)

เมื่อภาษาซี ได้รับความนิยมมากขึ้น จึงมีผู้ผลิต compiler ภาษาซีออกมาแข่งขันกันมากมาย ทำให้เริ่มมีการใส่ลูกเล่นต่างๆ เพื่อดึงดูดใจผู้ซื้อทางAmericanNationalStandardInstitute (ANSI) จึงตั้งข้อกำหนดมาตรฐานของภาษาซีขึ้น เรียกว่า ANSI C เพื่อคงมาตรฐานของภาษาไว้ไม่ให้เปลี่ยนแปลงไป

13. Web Camera

Web Camera หรือที่นิยมเรียกกันว่า Webcam แต่ในบางครั้งก็มีคนเรียกว่า Video Camera หรือ Video Conference เว็บแคมเป็นอุปกรณ์อินพุตที่ สามารถจับภาพเคลื่อนไหวของเราไปปรากฏในหน้าจอมอนิเตอร์ และสามารถส่งภาพเคลื่อนไหวนี้ผ่านระบบเครือข่ายเพื่อให้คนอีกฟากหนึ่งสามารถเห็นตัวเราเคลื่อนไหว ได้เหมือนอยู่ต่อหน้า ถือว่าเป็นอุปกรณ์ที่มีประโยชน์อีกตัวหนึ่งและเริ่มมีความจำเป็นมากขึ้นเรื่อยๆ

14. มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นตัวขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรม เพราะมีคุณสมบัติที่ดีในด้านการปรับความเร็วได้ ตั้งแต่ความเร็วต่ำสุดจนถึงความเร็วสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม และใช้กันอย่างแพร่หลายในงานช่างทั่วไป

14.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง มอเตอร์ไฟฟ้ากระแสตรงมีส่วนประกอบที่สำคัญ 2 ส่วน ดังนี้

14.1.1 ส่วนที่อยู่กับที่ เรียกว่า สเตเตอร์ (Stator) ประกอบด้วย เฟรมหรือโยค (Frame or Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาม้วนเป็นรูปทรงกระบอกโดยขั้วแม่เหล็กประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวด

ภาพที่ 2-5 Stator

ส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกอบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กจากผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์มีกำลังหมุน

ภาพที่ 2-6 Pole Core

ส่วนที่สอง ขดลวดสนามแม่เหล็ก (Field coil) จะพันอยู่รอบๆแกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น

ภาพที่ 2-7 ขั้วแม่เหล็ก

14.1.2 ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์นี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์

ภาพที่ 2-8 Rotors

ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. แกนเพลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลานี้จะวางอยู่บนแบริ่ง เพื่อบังคับให้หมุนอยู่ในแนวนิ่งไม่มีการสั่นสะเทือนได้

2. แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด

3. คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับ

ใส่ปลายสาย ของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่านเพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่าปฏิกิริยามอเตอร์

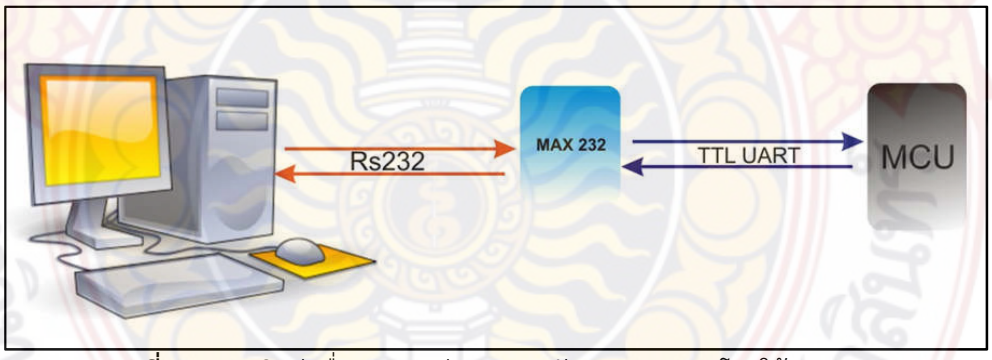
4. ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ

14.2 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง เมื่อเป็นแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะ แปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็กสร้างขั้วเหนือ-ใต้ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรง แม่เหล็ก จะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ซึ่งวางแกนเพลาและแกนเพลานี้ สวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ ซึ่งหมายความว่าตัวหมุน การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกิริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์ หรือโรเตอร์หมุนไปนั้นเป็นไปตามกฎมือซ้ายของเฟลมมิ่ง

15. IC MAX232 สำหรับสื่อสารอนุกรม

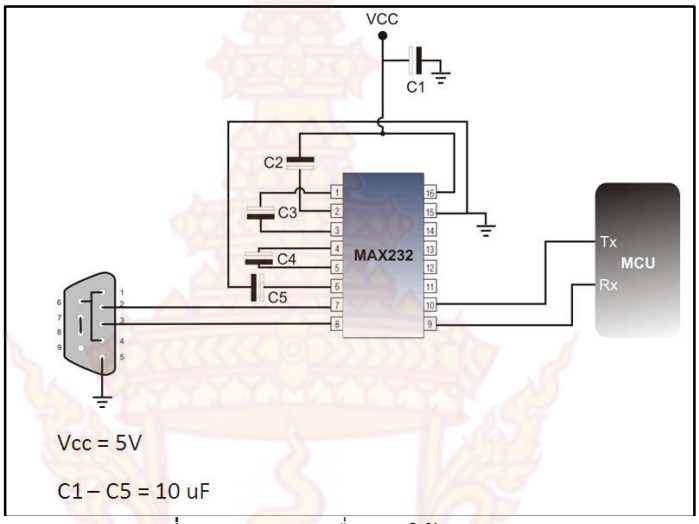
เป็น IC ที่ใช้เปลี่ยน TTL เป็น RS232 ในฝั่งส่ง และ เปลี่ยน RS232 เป็น TTL ในฝั่งรับ ดังภาพที่ 2-9



ภาพที่ 2-9 การติดต่อสื่อสารระหว่าง MCU กับ Computer โดยใช้ MAX232

เนื่องจาก SerialPortของPCเป็นมาตรฐาน RS232แต่MCUเป็นTTLจึงต้องใช้ MAX232 ปรับระดับแรงดันให้อยู่ในระดับเดียวกัน

15.1 วิธีต่อใช้งาน MAX232



ภาพที่ 2-10 การต่อเชื่อมต่อใช้งาน MAX232

ข้อมูลที่ถูกส่งมาจากคอมพิวเตอร์โดยพอร์ตอนุกรม จะถูกรับเข้ามาโดย connector ชนิด DB9 ที่ขา 2 เข้าไปที่ IC MAX232 เพื่อทำการแปลงสัญญาณเป็นแบบ TTLจากนั้นจะส่งออกทางขา TX (ขา 9) เพื่อส่งไปยังไมโครคอนโทรลเลอร์ต่อไป

15.2 ขาของ Connector DB9

ภาพที่ 2-11 หัวเชื่อมต่อของ RS232 แบบ DB9

ตารางที่ 2-1 ตารางแสดงตำแหน่งขาและหน้าที่ของขาของ Connector DB9

ตำแหน่งขา	หน้าที่	คำย่อ
PIN #1	Data Carrier Detect	CD
PIN #2	Receive Data	RX / RD / RXD
PIN #3	Transmitted Data	TX / TD / TXD

PIN #4	Data Terminal Ready	DTR
PIN #5	Signal Ground	GND
PIN #6	Data Set Ready	DSR
PIN #7	Requests to Send	RTS
PIN #8	Clear to Send	CTS
PIN #9	Ring Indicator	RI

CD = ขาสัญญาณตรวจจับคลื่นพาห์ เป็นขาอินพุต

TX = เป็นขาส่งข้อมูล

RX = เป็นขารับข้อมูล

RTS = เป็นขาที่ส่งสถานะไปยังตัวรับ ว่าต้องการส่งข้อมูล เมื่อต้องการส่งข้อมูล จะ ON จนกระทั้งส่ง Data ออกทางขา TX จนเสร็จจึงจะ OFFในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

CTS = ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่

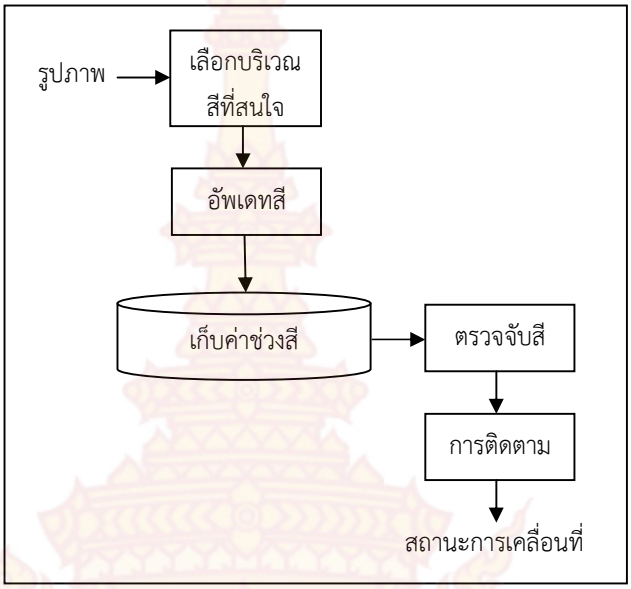
DTR = เป็นขาที่แสดงสถานะว่า Port นั้นเปิดอยู่หรือไม่เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วยในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

DSR = เป็นขาที่ใช้ตรวจเช็ค สถานะ DTR ของอุปกรณ์ที่เชื่อมต่ออยู่ด้วย

GND = Signal Ground

16. ขั้นตอนการตรวจจับสีและติดตามเป้าหมาย

ตามปกติเมื่อรับภาพเข้ามาภาพที่ได้จะมีสัญญาณรบกวนทางแสงทำให้ไม่สามารถตรวจจับสีได้อย่างถูกต้องดังนั้นเราต้องทำการชดเชยการรบกวนของแสงแวดล้อมโดยการกำหนดค่าสีและช่วงสีที่ยอมรับได้ก่อนที่จะทำการตรวจจับและติดตามเป้าหมาย ดังนั้นขั้นตอนการตรวจจับสีและเป้าหมายจึงแบ่งได้เป็น 2 ขั้นตอนย่อยคือการเก็บค่าช่วงสีในบริเวณภาพที่สนใจและการดึงค่าช่วงสีที่เก็บไว้ไปใช้งานดังภาพที่ 12



ภาพที่ 2-12 ขั้นตอนการตรวจจับสีและติดตามเป้าหมาย

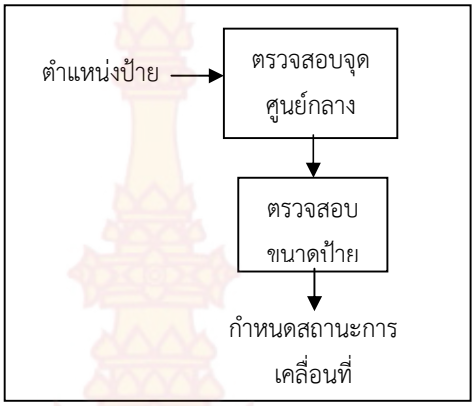
จากภาพที่ 2-12 ส่วนการเก็บค่าสีเราจะดำเนินการทั้งแบบ offline และ online กล่าวคือ แบบ offline ภาพที่กำหนดขอบเขตบริเวณในภาพที่ชัดเจนซึ่งจะเป็นบริเวณที่สีที่สนใจปรากฏจะถูกบันทึกไว้ก่อนและทำการจำลองสีด้วยแบบจำลอง non-stationary Gaussian ดังสมการ

$$\mu_{t+1} = \alpha\mu_{t} + (1-\alpha)z_{t+1} \quad (2\text{-}2)$$

$$\sigma^2_{t+1} = \alpha\left(\sigma^2_t + (\mu_{t+1} - \mu_t)^2\right) + (1-\alpha)(z_{t+1} - \mu_{t+1})^2 \quad (2\text{-}3)$$

เมื่อ μ_{t+1}, μ_t คือค่าเฉลี่ยขององค์ประกอบของสี ณ เฟรมภาพใหม่กับเฟรมภาพปัจจุบัน z_{t+1} คือค่าองค์ประกอบของสี ที่ตรวจจับได้ ณ เฟรมใหม่ σ^2_{t+1} คือค่าวาเรียนท์ขององค์ประกอบของสี ณ เฟรมภาพใหม่และ α คือค่าถ่วงน้ำหนัก จากสมการที่ 1 และ 2 เหมาะกับภาพระดับเทา แต่ในการประมวลผลเรากระทำกับภาพสีดังนั้นเราจะดำเนินการแยกกันระหว่างองค์ประกอบสีในภาพสีคือ สีแดง, สีเขียวและสีน้ำเงิน และเมื่อถึงขั้นตอนการติดตาม การเก็บค่าสีแบบ online จะถูกดำเนินการเพราะระหว่างการติดตามป้าย สีอาจมีการเปลี่ยนแปลงเราจะทำการอัพเดทการเปลี่ยนแปลงสีโดยใช้สมการที่ 2 และ 3 ก่อนเก็บค่าสีไว้ใช้งานต่อไป

ในขั้นตอนการติดตามเราจะทำการติดตามป้ายสีโดยรายละเอียดสามารถแสดงได้ดังภาพที่



ภาพที่ 2-13 ขั้นตอนการติดตามป้ายสี

เมื่อได้ตำแหน่งป้ายจากขั้นตอนการตรวจจับสีตำแหน่งศูนย์กลางป้ายจถูกนำมาเปรียบเทียบกับเส้นศูนย์กลางในขั้นตอนการตรวจสอบศูนย์กลางเพื่อกำหนดว่าจะให้รถเข็นเลี้ยวซ้ายหรือเลี้ยวขวา ทั้งนี้เพื่อให้การเคลื่อนที่ของรถเข็นราบลื่นไม่กระตุกซ้ายขวาเรากำหนดขอบเขตรอบๆ เส้นแบ่งกลางไว้ด้วย ลำดับต่อมาการตรวจสอบขนาดป้ายจะทำการนับจำนวนพิกเซลภาพที่ตรวจจับได้ในขั้นตอนการตรวจจับสีว่ามีขนาดเท่าไร ถ้ามีขนาดเล็กว่าขนาดที่กำหนดแสดงว่าป้ายอยู่ห่างรถเข็นมากให้สั่งรถเข็นเดินหน้าในทางตรงกันข้ามถ้ามีขนาดใหญ่แสดงว่าป้ายอยู่ใกล้รถเข็นมากให้สั่งรถเข็นถอยหลัง ดังนั้นโดยสรุปขั้นตอนการติดตามป้ายสีจะมีข้อมูลส่งไปยังกล่องควบคุมว่าจะให้รถเลี้ยวซ้ายหรือขวาและเดินหน้าหรือถอยหลังหรือหยุด

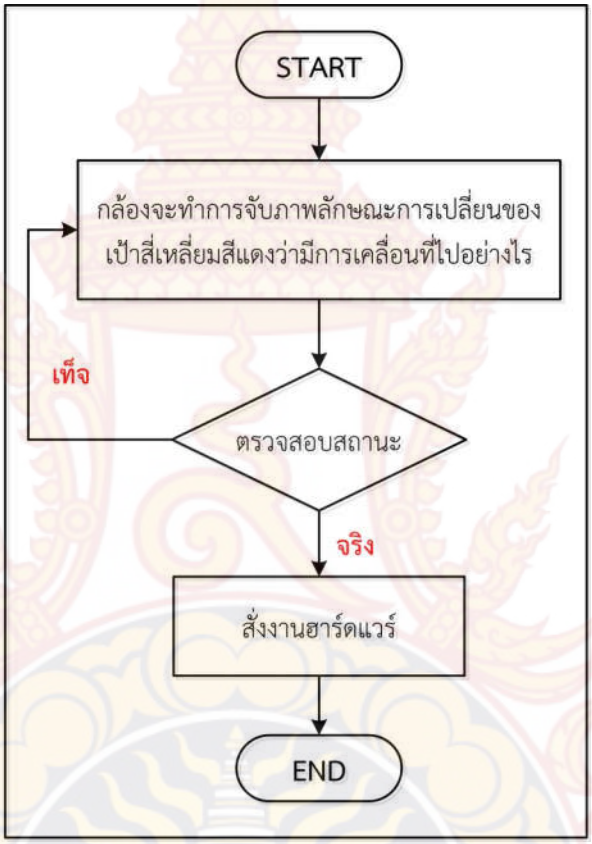
บทที่ 3
ขั้นตอนการดำเนินงาน

การสร้างรถเข็นอัตโนมัติรับค่าจากตัวประมวลผลภาพ (Automatic shopping cart by receiving data from image processing) มีทั้งในส่วนของซอฟท์แวร์และฮาร์ดแวร์โดยสามารถแบ่งขั้นตอนการดำเนินงานออกได้ดังต่อไปนี้

1. ออกแบบ Flow Chart การทำงานของรถเข็นอัตโนมัติฯ
2. ออกแบบโปรแกรมประมวลผลภาพจากเป้าสี่เหลี่ยมและโปแกรมรับค่าจากการประมวลผลภาพ
3. ออกแบบตัวรถเข็น
4. ออกแบบวงจร MAX232 เพื่อรับข้อมูลจากโปรแกรมประมวลผลภาพ
5. ออกแบบวงจรไมโครคอนโทรลเลอร์เพื่อรับข้อมูลจาก MAX232
6. ออกแบบวงจรขับมอเตอร์เพื่อรับคำสั่งจากไมโครคอนโทรลเลอร์ไปขับมอเตอร์
7. สรุปวงจรรวมทั้งหมด
8. กัดลายวงจรทั้งหมดลงบนแผ่นปริ๊นท์
9. จัดเตรียมอุปกรณ์
10. จัดวางวงจรทั้งหมด

1. ออกแบบ Flow Chart การทำงานของรถเข็นอัตโนมัติ

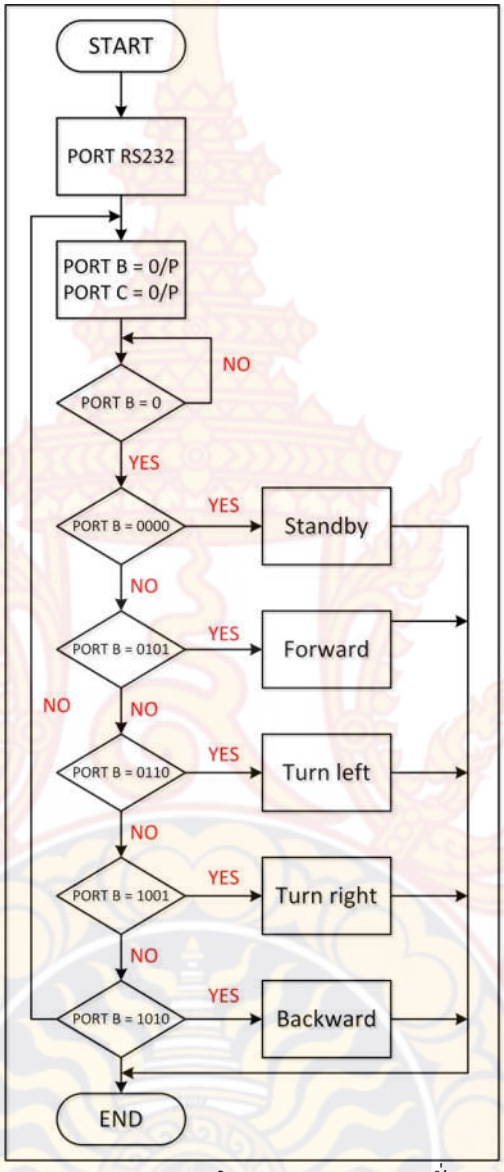
ในการปฏิบัติงานให้สำเร็จลุล่วงตามเป้าหมายภายในเวลาที่กำหนดนั้นจำเป็นจะต้องมีการวางแผนโครงงานวิจัยก่อนเพื่อให้ทราบแนวทางและขั้นตอนการปฏิบัติที่ถูกต้อง จากการศึกษาทฤษฎีในบทที่ 2 สามารถแบ่งขั้นตอนการดำเนินงานของรถเข็นอัตโนมัติซึ่งควบคุมการทำงานโดยใช้ไมโครคอนโทรลเลอร์ได้ดังภาพที่ 3-1



ภาพที่ 3-1 Flow chart การทำงานเครื่องควบคุมรถเข็นอัตโนมัติ

เมื่อเริ่มการทำงานของโปรแกรม Image processing ซึ่งจะทำการรับภาพจากกล้อง webcamซึ่งภาพที่รับมาได้นั้น โปรแกรมจะทำหน้าที่ประมวลผลภาพคือ เลือกเป้าที่เป็นลักษณะรูปสี่เหลี่ยมสีแดงเมื่อได้รูปตามกำหนด ก็จะทำการประมวลผลการเคลื่อนที่ของเป้าสี่เหลี่ยมสีแดงว่ามีลักษณะการเคลื่อนที่อย่างไรหลังจากนั้นโปรแกรม Image processing จะส่งค่าการเคลื่อนที่ ที่ได้ไปยัง ชุดควบคุมไมโครคอนโทรลเลอร์โดยผ่านวงจร RS232 หลังจากนั้นชุดควบคุมมอเตอร์นี้จะตอบสนองการเคลื่อนที่ของเป้าสี่เหลี่ยมอย่างอัตโนมัติอาทิเช่นการเคลื่อนที่ไปข้างหน้าเมื่อเป้าสี่เหลี่ยมสีแดงมีลักษณะเล็กลงโปรแกรมก็จะทำการสั่งการให้ชุดขับเคลื่อนมอเตอร์เดินหน้าเพื่อรักษา

ระยะให้เป้าสี่เหลี่ยมสีแดงมีขนาดปกติที่กำหนดไว้ หรือการควบคุมการเลี้ยวซ้ายและขวาต่างๆ แบบอัตโนมัติดังแสดงการทำงานได้ดังภาพที่ 15



ภาพที่ 3-2 Flow chart แสดงการทำงานชุดควบคุมการสั่งงานของ Software

เริ่มต้นการทำงานของ Software จะกำหนดพอร์ตอินพุตและเอาต์พุตและกำหนดลักษณะการทำงานของเอาต์พุตไว้ 4 พอร์ตโดยทั้ง 4 พอร์ตมีหน้าที่ควบคุมการทำงานของมอเตอร์ให้มีการขับเคลื่อนตามที่กำหนดไว้คือ

PortB = 0x00 = 0000 คือ โหมดหยุด

PortB = 0x05 = 0101 คือ โหมดเดินหน้า

PortB=0x06 = 0110คือ โหมดเลี้ยวซ้าย

PortB=0x09 = 1001 คือ โหมดเลี้ยวขวา

PortB=0x0A = 1010 คือ โหมดถอยหลัง

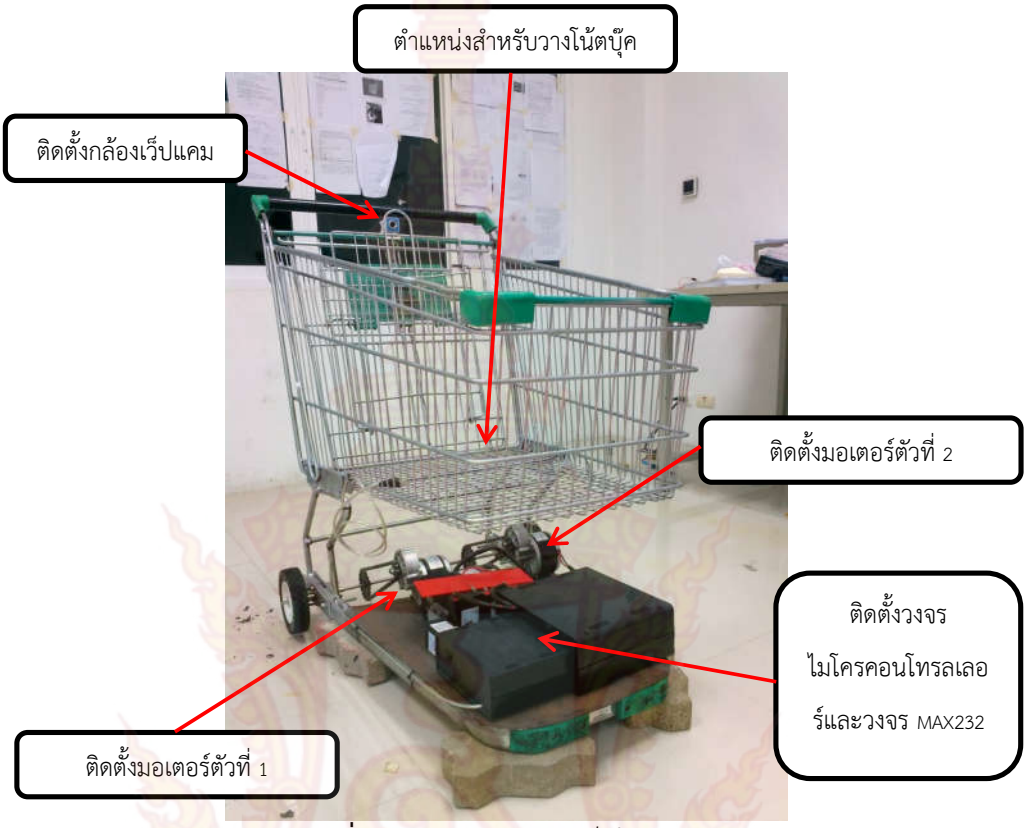
สำหรับคำสั่ง (Code) สามารถดูได้จากภาคผนวก

2. ออกแบบโปรแกรมประมวลผลภาพจากเป้าสี่เหลี่ยมและโปรแกรมรับค่าจากการประมวลผลภาพ

ในส่วนของโปรแกรม Image processing จะรับภาพจากกล้อง ซึ่งจะจับเฉพาะภาพเป้าสีแดงเท่านั้นถ้าไม่ใช่เป้าสีแดงโปรแกรมจะไม่ทำงาน เมื่อสามารถจับเป้าสีแดงได้แล้วโปรแกรมจะเริ่มประมวลผลทันที โดยจะเปลี่ยนเป้าสีแดงเป็นสีขาวและสีอื่นๆเป็นสีดำ ในที่นี้เราจะกำหนดเฟรมของเป้าไว้ที่ขนด 640 x 480 พิกเซลหลังจากจับภาพได้แล้ว โปรแกรมมีหน้าที่ตรวจสถานะตลอดเวลาของภาพว่าในลักษณะตอนนั้นภาพมีการเปลี่ยนแปลงอย่างไร และกำหนดลักษณะรูปแบบของการเคลื่อนที่เพื่อจะส่งไปยังชุดควบคุมไมโครคอนโทรลเลอร์และขับมอเตอร์ต่อไปซึ่งโค้ดโปรแกรมประมวลผลภาพและโปรแกรมรับค่าจากการประมวลผลภาพสามารถดูได้ที่ภาคผนวก

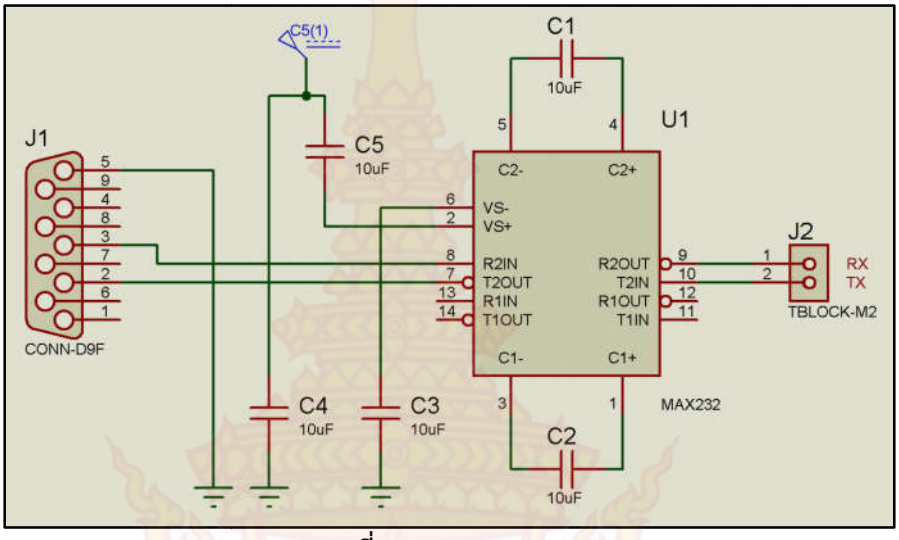
3. ออกแบบตัวรถเข็น

การออกแบบโครงสร้างของรถเข็นจะเป็นรถเข็นสินค้าขนาดมาตรฐานที่ใช้ตามห้างสรรพสินค้าทั่วไปจะเป็นการขับเคลื่อนด้วยมอเตอร์ 2 ตัว เพื่อควบคุมการเคลื่อนที่ของรถเข็น ซึ่งตัวโครงสร้าง จะประกอบด้วย รถเข็น 1 คัน มีมอเตอร์ 2 ตัว กล้อง webcam และโน้ตบุ๊ค1 เครื่องเพื่อทำหน้าที่ประมวลผลภาพที่รับได้จากกล้องซึ่งแสดงดังภาพที่ 16 อย่างคร่าวๆ



ภาพที่ 3-3 การออกแบบรถเข็นโดยสังเขป

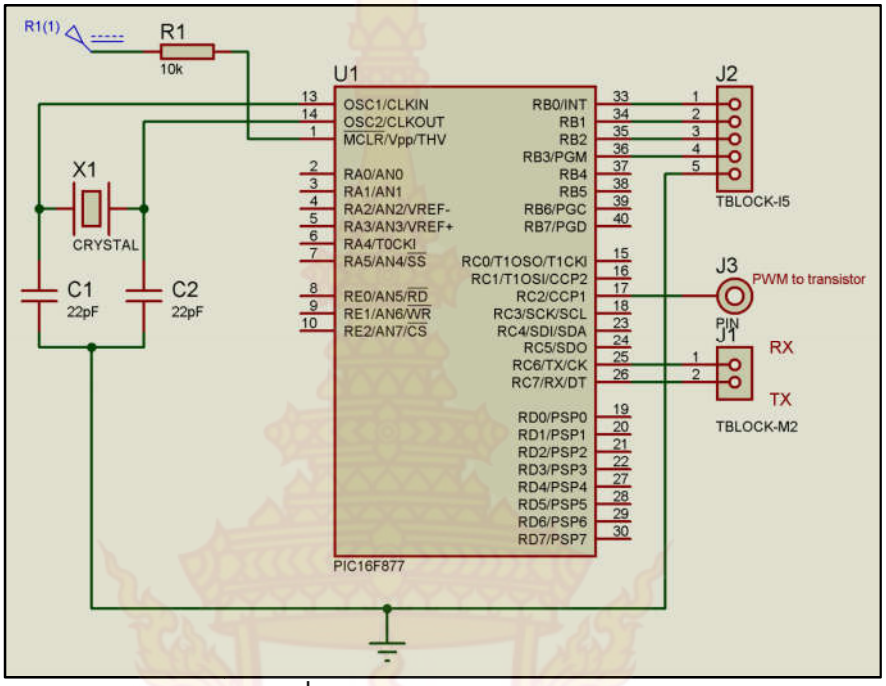
4. ออกแบบวงจร MAX232 เพื่อรับข้อมูลจากโปรแกรมประมวลผลภาพ



ภาพที่ 3-4 วงจร MAX232

เมื่อโปรแกรมประมวลผลภาพส่งข้อมูลออกมาทางพอร์ตอนุกรมข้อมูลจะถูกรับโดย MAX232 และจะถูกส่งต่อไปยังไมโครคอนโทรลเลอร์อีกทอดหนึ่ง เนื่องจากคอมพิวเตอร์จะรับส่งข้อมูลเป็นแบบ RS232แต่ไมโครคอนโทรลเลอร์จะรับส่งข้อมูลเป็นแบบ TTL-UART จึงต้องมี MAX232 เป็นตัวส่งผ่านข้อมูล

5. ออกแบบวงจรไมโครคอนโทรลเลอร์เพื่อรับข้อมูลจาก MAX232

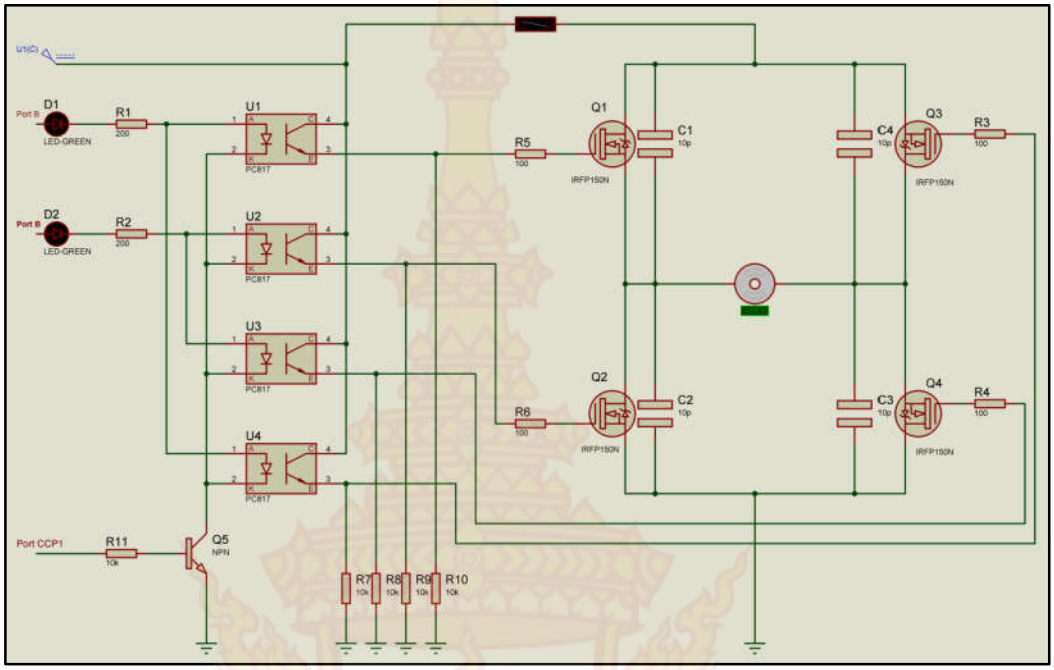


ภาพที่ 3-5 วงจรไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เมื่อรับข้อมูลจาก MAX232 มาแล้วจะนำข้อมูลนั้นมาเข้าไปพิจารณาในเงื่อนไขของโปรแกรมซึ่งอยู่ในรูปแบบของภาษซีที่ผู้จัดทำได้เขียนและบรรจุลงไว้ในไมโครคอนโทรลเลอร์ก่อนแล้ว เมื่อทำการเข้าเงื่อนไขแล้วก็จะทำการส่งข้อมูลไบนารีออกทางพอร์ต B เพื่อส่งให้วงจรขับมอเตอร์ต่อไป

6. ออกแบบวงจรขับมอเตอร์เพื่อรับคำสั่งจากไมโครคอนโทรลเลอร์ไปขับมอเตอร์

ในส่วนของภาควงจรขับมอเตอร์นี้ทำหน้าที่เป็นตัวรับค่าจากภาคไมโครคอนโทรลเลอร์หน้าที่ของวงจรเมื่อได้รับค่ามาแล้วจะทำหน้าที่เป็นตัวนำค่าที่ได้ไปใช้ ควบคุมมอเตอร์ที่ทำการติดตั้งไว้ทั้งสองตัว ซึ่งจากวงจรภายในจะใช้ มอสเฟสจำนวน 4 ตัว ต่อมอเตอร์ 1 ตัว ต่อหนึ่งวงจร เพื่อใช้ในการควบคุมการหมุนสลับข้างกันของมอเตอร์คือเดินหน้าและถอยหลัง ซึ่งสามารถแสดงได้ดังภาพที่ 19



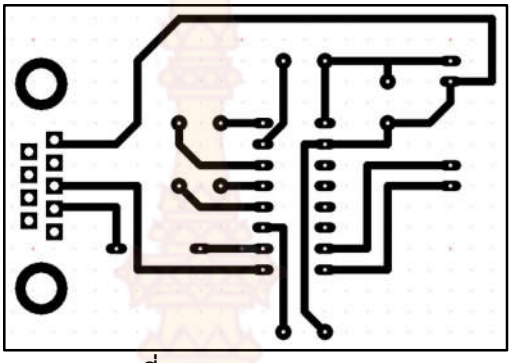
ภาพที่ 3-6 วงจรขับมอเตอร์

เมื่อไมโครคอนโทรลเลอร์ส่งข้อมูลออกมาทางพอร์ต B0, B1, B2 และ B3 สองพอร์ตจะมาเข้ากับวงจรขับมอเตอร์วงจรหนึ่ง ส่วนอีกสองพอร์ตก็จะไปเข้ากับวงจรขับมอเตอร์อีกวงจรหนึ่ง ซึ่งข้อมูลไบนารีที่แตกต่างกันที่ส่งมาจากไมโครคอนโทรลเลอร์จะส่งผลให้MOSFETทั้งแปดตัวในวงจรทำงานในลักษณะที่ต่างกัน ทำให้เกิดแรงดันไปขับมอเตอร์ในลักษณะที่ต่างกันไป

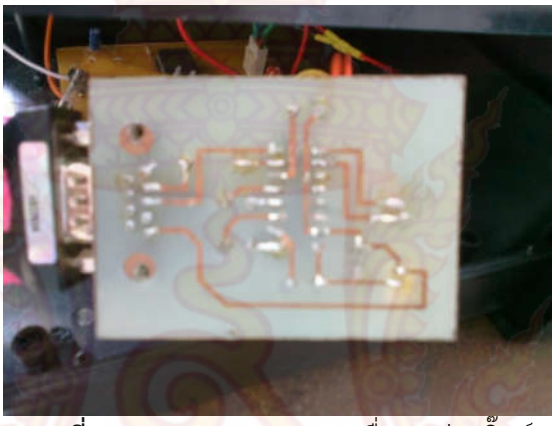
7. กัดลายวงจรทั้งหมดลงบนแผ่นปริ๊นท์

การออกแบบลายวงจรต่างๆนั้น ทางคณะผู้จัดทำได้เลือกใช้โปรแกรม PCBWizard ในการออกแบบ เนื่องจากใช้งานง่ายและชัดเจน โดยในการออกแบบนั้นต้องทำการเลือกอุปกรณ์ที่ใช้มาจัดวางให้อยู่ในตำแหน่งที่เหมาะสม ซึ่งอาจจะต้องอาศัยประสบการณ์ในการออกแบบลายวงจรด้วย จากนั้นจึงทำการลากเส้นเชื่อมระหว่างอุปกรณ์ต่างๆให้ถูกต้อง เมื่อออกแบบเสร็จแล้วทำการพิมพ์ลายที่ออกแบบไว้ลงบนแผ่นใส แล้วทำการกัดลายวงจรลงแผ่นปริ๊นท์โดยผู้จัดทำได้เลือกใช้วิธีการฉายแสงเพราะให้ลายวงจรที่คมสวยงาม

7.1 ลายวงจร MAX232

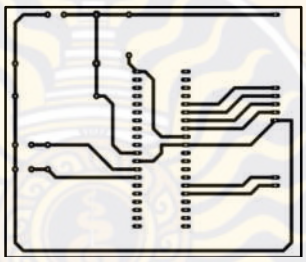


ภาพที่ 3-7 ลายวงจร MAX232



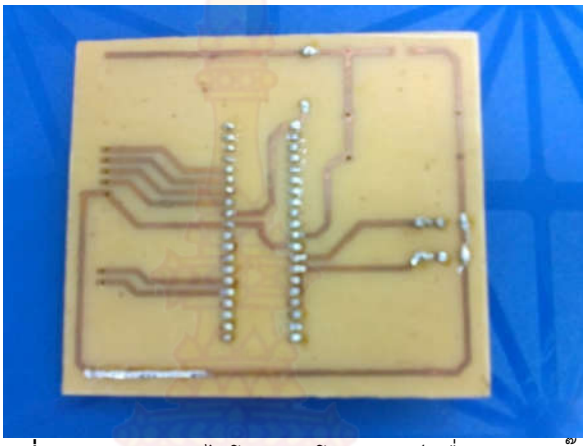
ภาพที่ 3-8 ลายวงจร MAX232 เมื่อลงแผ่นปริ๊นท์

7.2 ลายวงจรไมโครคอนโทรลเลอร์



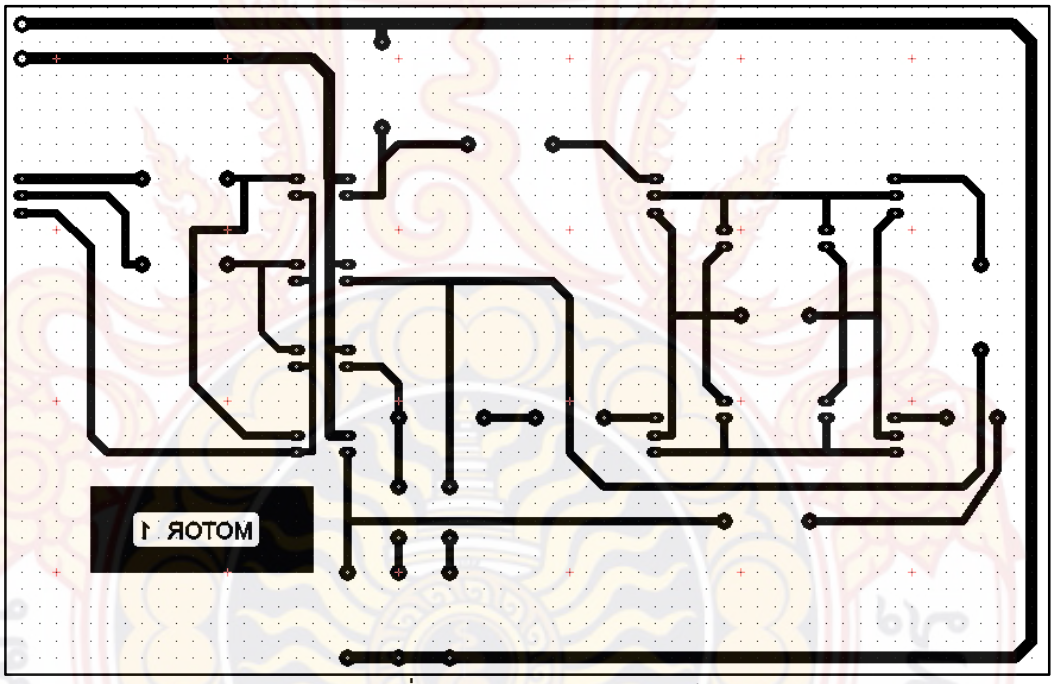
ภาพที่ 3-9 ลายวงจรไมโครคอนโทรลเลอร์

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

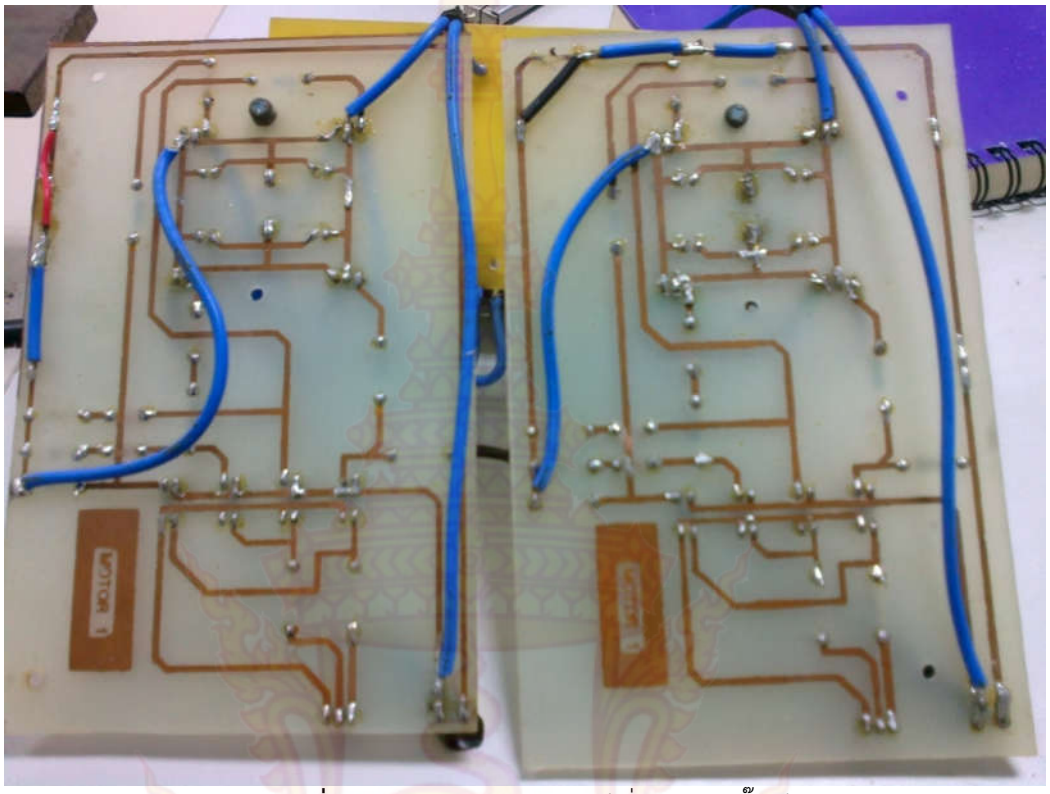


ภาพที่ 3-10 ลายวงจรไมโครคอนโทรลเลอร์ เมื่อลงแผ่นปริ๊นท์

7.3 ลายวงจรขับมอเตอร์



ภาพที่ 3-11 ลายวงจรขับมอเตอร์



ภาพที่ 3-12 ลายวงจรขับมอเตอร์เมื่อลงแผ่นปริ๊นท์

8. จัดเตรียมอุปกรณ์ทั้งหมด

8.1 วงจร MAX232

1. MAX232	1	ตัว
2. ตัวเก็บประจุ (Capacitor) 10uF	5	ตัว
3. หัว Connector ชนิด DB9	1	ตัว

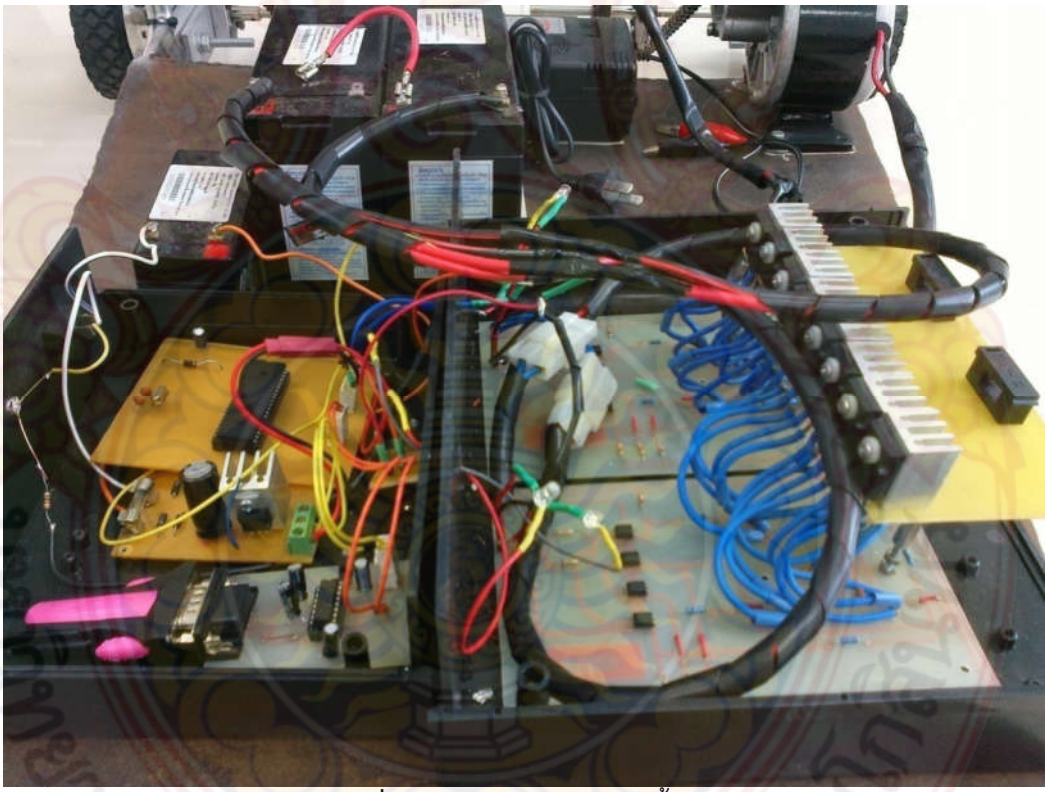
8.2 วงจรไมโครคอนโทรลเลอร์

1. PIC16F877	1	ตัว
2. ตัวต้านทาน (Resistor) 10kΩ	1	ตัว
3. ตัวเก็บประจุ (Capacitor) 100uF	1	ตัว
4. ตัวเก็บประจุ (Capacitor) 22pF	2	ตัว
5. ไดโอด (Diode) 1N4007	1	ตัว
6. Crystal 20MHz	1	ตัว
7. แบตเตอรี่ (Battery) 12V 2A	1	ก้อน

8.3 วงจรขับมอเตอร์ (2 วงจร)

1. ออปโต้ (OPTO) PC817	8	ตัว
2. มอสเฟ็ท (MOSFET) IRFP150N	8	ตัว
3. ตัวต้านทาน (Resistor) 100Ω	8	ตัว
4. ตัวต้านทาน (Resistor) 1kΩ	8	ตัว
5. ตัวต้านทาน (Resistor) 200Ω	4	ตัว
6. ตัวต้านทาน (Resistor) 10kΩ	2	ตัว
7. ตัวเก็บประจุ (Capacitor) 10pF	8	ตัว
8. ทรานซิสเตอร์ (Transistor) C1815L	2	ตัว
9. ฟิวส์ (Fuse) 10A	2	ตัว
10. LED สีเขียว	4	หลอด
11. แบตเตอรี่ (Battery) 12V 7.8A	2	ก้อน

9. จัดวางอุปกรณ์ทั้งหมด



ภาพที่ 3-13 การจัดวางอุปกรณ์ทั้งหมด

บทที่ 4
การทดสอบและผลการทดสอบ

การทดลองทั้งหมดแบ่งออกเป็น 4 การทดลอง ได้แก่ การทดสอบการทำงานของโปรแกรม Image processing, การทดสอบวงจรขับมอเตอร์, การทดสอบความเร็วในการเดิน และการทดสอบระยะที่โปรแกรมสามารถจับเป้าสีแดงได้

1. การทดสอบการทำงานของโปรแกรม Image Processing

1.1 วัตถุประสงค์

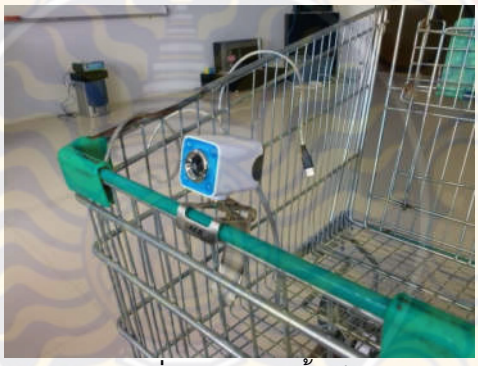
เพื่อทดสอบการจับภาพเป้าสี่เหลี่ยมสีแดงและการแสดงผลของโปรแกรม Image Processing

1.2 วัสดุอุปกรณ์ที่ใช้

1. โมดูลประมวลผลภาพ (Image Processing) 1 ชุด
2. กล้องเว็ปแคม 1 ตัว
3. เป้าสี่เหลี่ยมสีแดงขนาด 12.5 X 11 cm 1 อัน

1.3 ขั้นตอนการทดสอบ

1. ติดตั้งกล้องไว้ที่ด้านหน้ารถเข็น

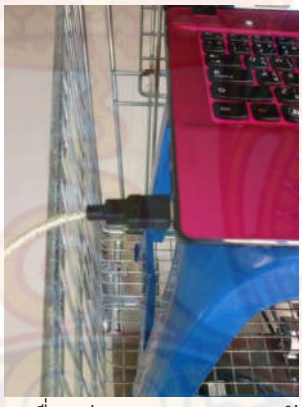


ภาพที่ 4-1 การติดตั้งกล้อง

2. เชื่อมต่อสาย USB ของกล้องเข้ากับคอมพิวเตอร์

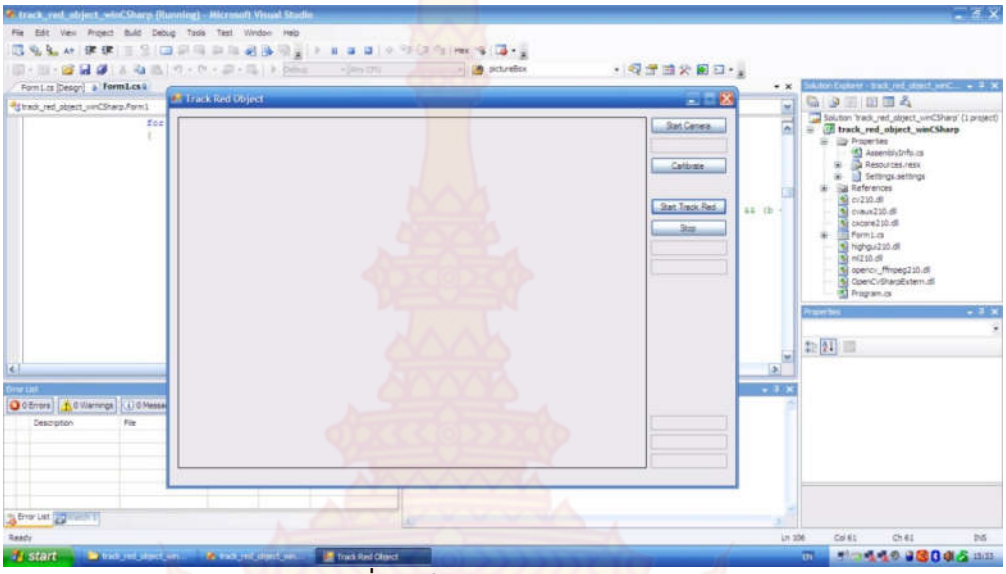
ภาพที่ 4-2 การเชื่อมต่อกล้องกับคอมพิวเตอร์

3. เชื่อมต่อสาย USB to RS232 เพื่อให้สามารถใช้งานพอร์ตได้



ภาพที่ 4-3 การเชื่อมต่อวงจร MAX232 กับคอมพิวเตอร์

4. เปิดใช้งานโมดูลประมวลผลภาพ



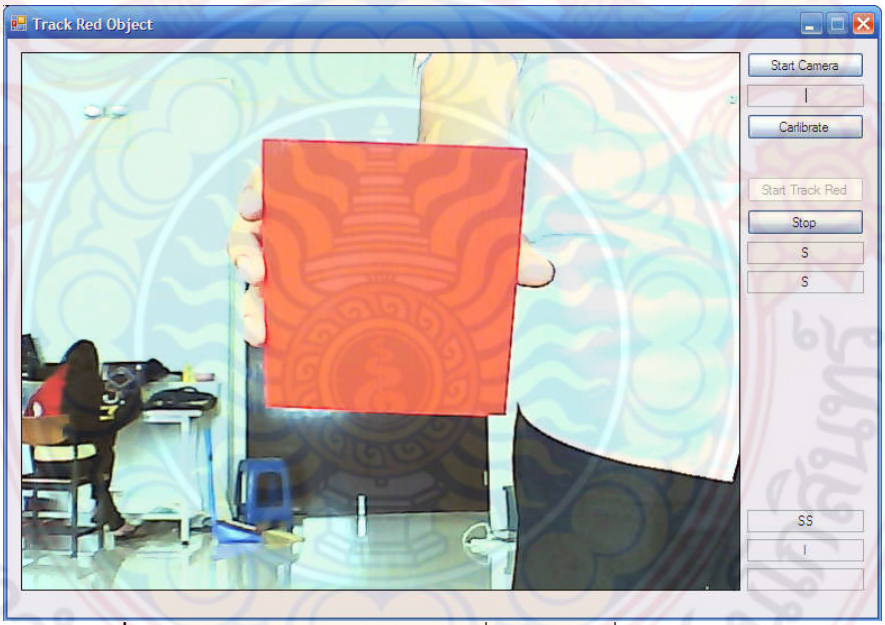
ภาพที่ 4-4 โปรแกรมประมวลผลภาพ

5. คลิกปุ่ม Start Track Red

6. นำเป้าสีแดงมาไว้ด้านหน้ากล้อง และขยับเป้าสีแดงไปที่ตำแหน่งซ้าย, ขวา, ใกล้ และไกล สังเกตการณ์เปลี่ยนแปลงค่าใน textbox ด้านขวา

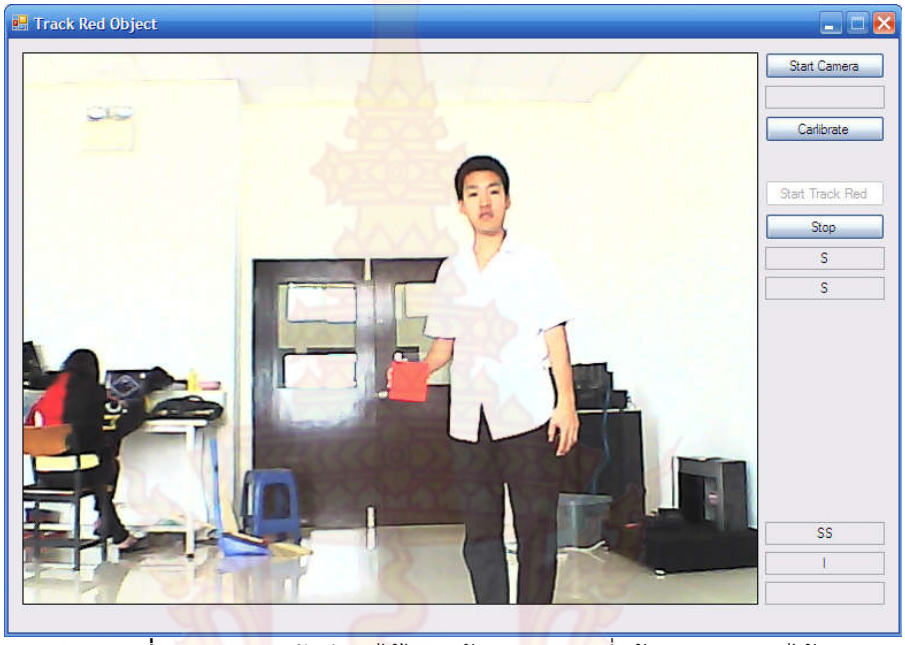
1.4 ผลการทดสอบ

1. เมื่อนำเป้าสีแดงไว้ใกล้กล้องได้ผลดังรูป ซึ่งเป็นการแสดงผลที่ถูกต้อง



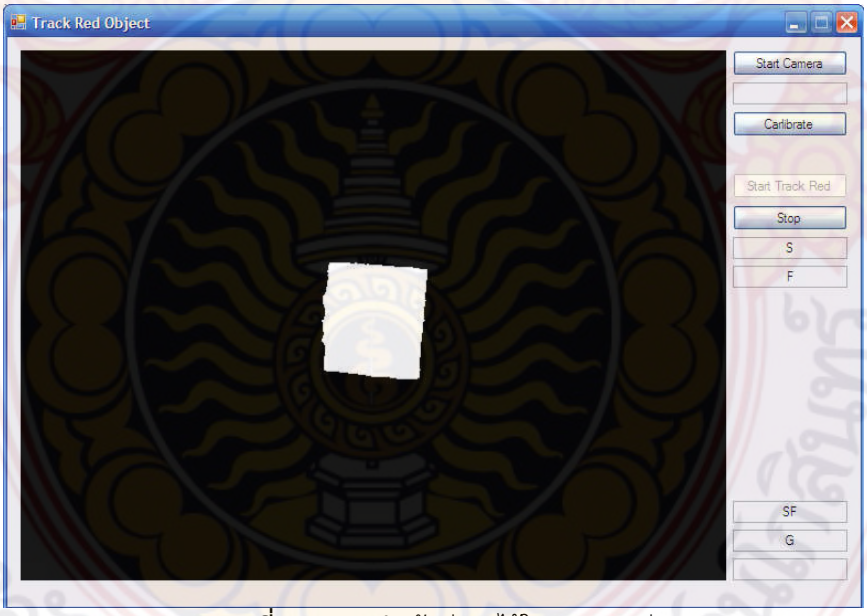
ภาพที่ 4-5 การนำเป้าสีแดงไว้ใกล้กล้องต่ำกว่าระยะที่กล้องสามารถจับได้

2. เมื่อนำเป้าสีแดงไว้ไกลกล้อง ได้ผลดังรูป ซึ่งเป็นการแสดงผลที่ถูกต้อง



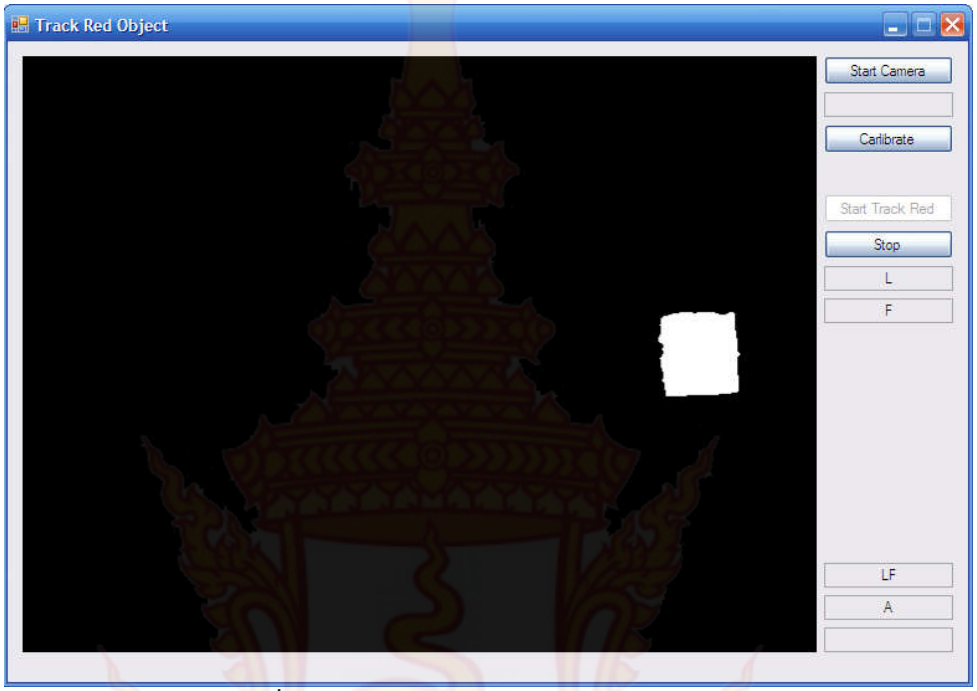
ภาพที่ 4-6 การนำเป้าสีแดงไว้ไกลกล้องเกินระยะที่กล้องสามารถจับได้

3. เมื่อนำเป้าสีแดงไว้ตรงกลาง ได้ผลดังรูป ซึ่งเป็นการแสดงผลที่ถูกต้อง



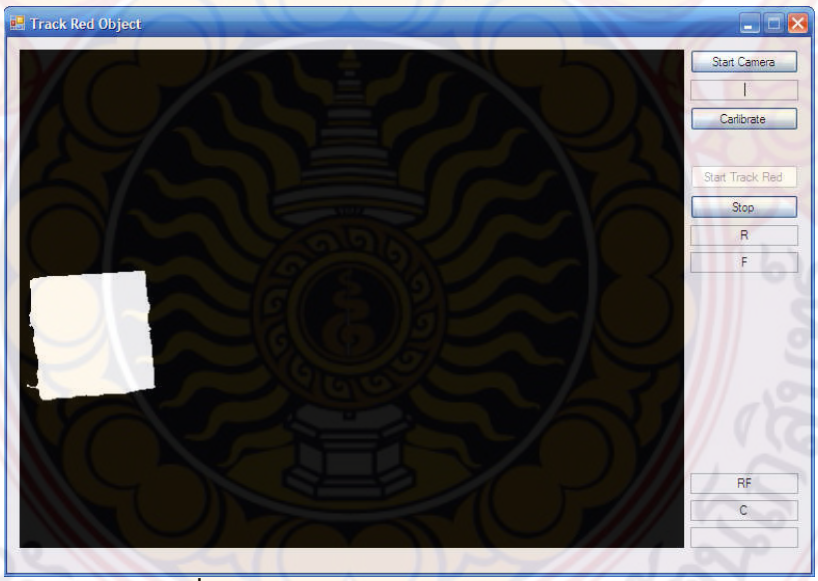
ภาพที่ 4-7 การนำเป้าสีแดงไว้ในระยะพอดี

4. เมื่อนำเป้าสีแดงไว้ด้านซ้ายของกล้อง ได้ผลดังรูป ซึ่งเป็นการแสดงผลที่ถูกต้อง



ภาพที่ 4-8 การนำเป้าสีแดงไว้ทางด้านซ้ายของกล้อง

5. เมื่อนำเป้าสีแดงไว้ด้านขวาของกล้อง ได้ผลดังรูป ซึ่งเป็นการแสดงผลที่ถูกต้อง



ภาพที่ 4-9 การนำเป้าสีแดงไว้ทางด้านขวาของกล้อง

2. การทดสอบวงจรขับมอเตอร์

2.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของวงจรขับมอเตอร์ทั้งสองวงจรว่าสามารถทำงานได้ตามที่ต้องการหรือไม่

2.2 วัสดุอุปกรณ์ที่ใช้

1. วงจรขับมอเตอร์ 2 ชุด
2. วงจรไมโครคอนโทรลเลอร์ 1 ชุด
3. แบตเตอรี่ขนาด 12V 2 ก้อน
4. มอเตอร์ไฟฟ้ากระแสตรงขนาด 24V 2 ตัว

2.3 ขั้นตอนการทดสอบ

1. ทดสอบการขับมอเตอร์เพื่อควบคุมทิศทางของรถเข็นฯโดยเขียนโค้ดภาษาซีอย่างง่ายในการทดสอบดังนี้

```
#include <16F877.h>
#FUSES NOWDT,NOPROTECT,NOLVP,HS
#USE delay(clock=20M)
#USE fast_io(b)
void main()
{
set_tris_b(0x00);
while(1)
{
 output_b(0x05);          //ให้รถเข็นเดินหน้า
 delay_ms(3000); //หน่วงเวลา 3 วินาที
 output_b(0x00); //ให้รถเข็นหยุดนิ่ง
 delay_ms(2000); //หน่วงเวลา 2 วินาที
 output_b(0x0A);          //ให้รถเข็นถอยหลัง
 delay_ms(3000); //หน่วงเวลา 3 วินาที
 output_b(0x00); //ให้รถเข็นหยุดนิ่ง
 delay_ms(2000); //หน่วงเวลา 2 วินาที
 output_b(0x09);          //ให้รถเข็นเลี้ยวขวา
 delay_ms(3000); //หน่วงเวลา 3 วินาที
```

```
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
            output_b(0x06);          //ให้รถเข็นเลี้ยวซ้าย
            delay_ms(3000); //หน่วงเวลา 3 วินาที
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
          }
    }
```

2.4 ผลการทดสอบ

เมื่อป้อนโค้ดภาษาซีตามด้านบนเข้าไปใน PIC แล้วลองเปิดสวิทช์ปรากฏว่ารถเข็นฯ สามารถเดินหน้า, ถอยหลัง, เลี้ยวซ้าย และเลี้ยวขวาได้

3. การทดสอบความเร็วในการเดิน

3.1 วัตถุประสงค์

เพื่อทดสอบความเร็วในการเดินที่ผู้ใช้สามารถใช้ได้โดยรถเข็นสามารถวิ่งตามได้อย่างไม่ติดขัด

3.2 วัสดุอุปกรณ์ที่ใช้

1. โมดูลประมวลผลภาพ (Image Processing) 1 ชุด
2. โมดูลรับข้อมูลจาก Image Processing 1 ชุด
3. โมดูลขับมอเตอร์กระแสตรง 2 ชุด
4. มอเตอร์กระแสตรงขนาด 24V 250W 2 ตัว

3.3 ขั้นตอนการทดลอง

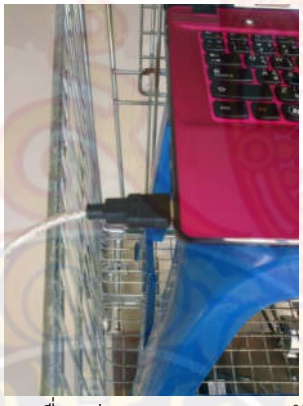
1. ติดตั้งอุปกรณ์ดังรูป
   - ติดตั้งกล้องไว้ที่ด้านหน้ารถเข็น

ภาพที่ 4-10 การติดตั้งกล้อง

- เชื่อมต่อสาย USB ของกล้องเข้ากับคอมพิวเตอร์

ภาพที่ 4-11 การเชื่อมต่อกล้องกับคอมพิวเตอร์

- เชื่อมต่อสาย USB ของโมดูลรับข้อมูลจาก Image Processing



ภาพที่ 4-12 การเชื่อมต่อวงจร MAX232 กับคอมพิวเตอร์

- เปิดสวิตช์ที่แผงวงจรด้านล่าง

ภาพที่ 4-13 แสดงการติดของไป LED เมื่อเปิดสวิทช์

- การติดตั้งอุปกรณ์โดยรวม

ภาพที่ 4-14 การติดตั้งอุปกรณ์โดยรวม

2. นำเป้าสีแดงรัดติดไว้ที่ด้านหลัง ดังรูป
3. ค่อยๆเดินไปข้างหน้า โดยให้เป้าสีแดงอยู่ในระยะด้านข้างที่กล้องสามารถมองเห็นได้

3.4 ผลการทดสอบ

เมื่อทำการทดลองทั้งหมด 10 ครั้ง โดยแต่ละครั้งใช้เวลา 1 นาที สามารถแสดงเป็นตารางผลการทดลองได้ดังตารางที่ 2

ตารางที่ 4-1 ผลการทดสอบความเร็วในการเดิน

ครั้งที่	ระยะทาง (เมตร)	ความเร็วในหน่วย เมตร/วินาที
1	9.1	0.152
2	8.9	0.148
3	9.2	0.153
4	9.5	0.158
5	9.0	0.15
6	9.2	0.153
7	9.4	0.157
8	9.7	0.162
9	9.6	0.16
10	9.3	0.155
เฉลี่ย	**9.3**	**0.155**

เมื่อปรับแรงดันไฟฟ้าในการขับมอเตอร์ไป 12V ซึ่งเป็นแรงดันต่ำสุดที่ปรับได้ ความเร็วมากที่สุดในการเดินของผู้ใช้งาน ที่สามารถใช้งานได้อย่างไม่ติดขัด คือ 9.3เมตรต่อนาทีหรือ0.155 เมตร/วินาทีและสามารถหาค่าความแปรปรวนของข้อมูลได้จาก $\sigma^2 = \frac{\Sigma(X-\mu)^2}{N_p}$ ซึ่งมีค่าเท่ากับ 0.061 ทั้งนี้ถ้าหากปรับแรงดันไฟฟ้าในการขับมอเตอร์มากขึ้น ก็จะทำให้สามารถใช้ความเร็วในการเดินได้มากขึ้น ซึ่งการปรับแรงดันในการขับมอเตอร์สามารถปรับได้มากที่สุด 24V

4. การทดสอบระยะที่โปรแกรมสามารถจับเป้าสีแดงได้

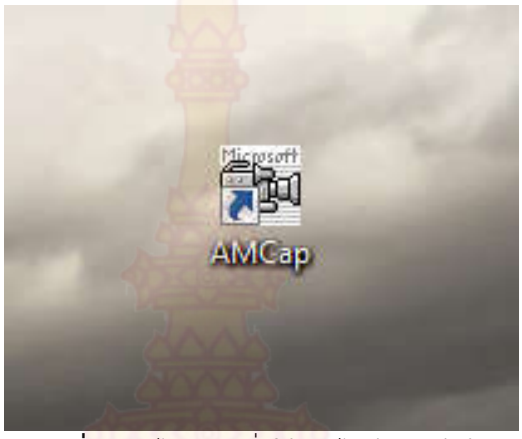
4.1 วัตถุประสงค์

เพื่อหาระยะที่โปรแกรม Image Processing สามารถจับภาพเป้าสี่เหลี่ยมสีแดงและนำมาประมวลผลได้

4.2 ค่าชดเชยแสงอัตโนมัติ (Auto Exposure)

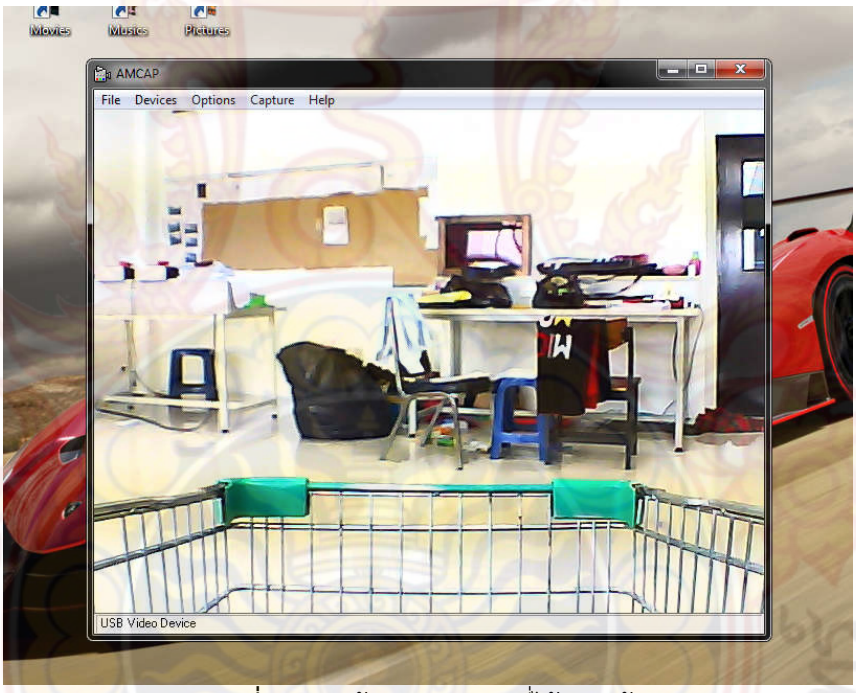
ในการทดสอบนี้จะแบ่งออกเป็น 2 ส่วนคือ ส่วนที่เปิดการใช้ค่าชดเชยแสงอัตโนมัติ และส่วนที่ปิดการใช้ค่าชดเชยแสงอัตโนมัติโดยค่าชดเชยแสงอัตโนมัติคือค่าแสงสีขาวที่กล้องสามารถเพิ่มหรือลดให้กับภาพได้ ส่งผลให้ภาพสว่างขึ้นหรือมืดลงได้โดยอัตโนมัติ ซึ่งค่านี้ผู้ใช้กล้องสามารถเปิดและปิดได้ตามที่ต้องการ โดยการเข้าไปตั้งค่าที่ตัวไดร์ฟเวอร์ที่ให้มากับกล้องตามขั้นตอนดังนี้ ซึ่งในที่นี้ผู้จัดทำได้ใช้กล้อง OKER 16M

1. เข้าใช้งานไดร์ฟเวอร์ของกล้องโดยคลิกที่ไอคอนดังภาพ



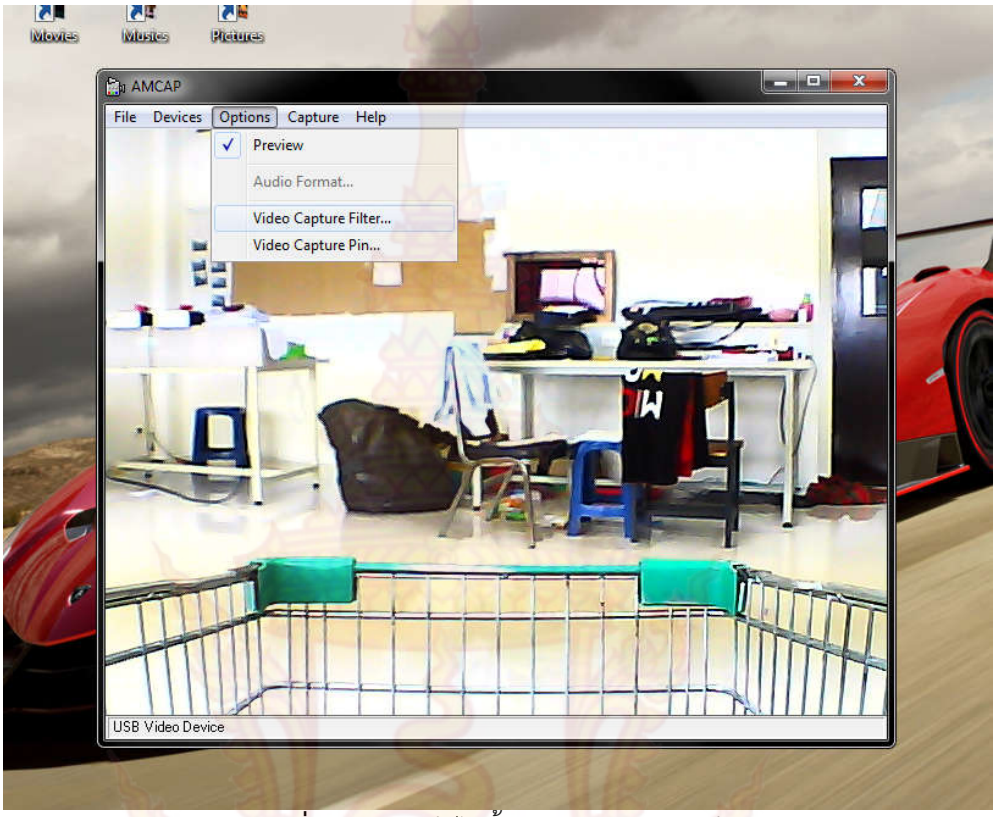
ภาพที่ 4-15 ไอคอนเพื่อใช้งานไดร์ฟเวอร์กล้อง

2. จะเข้าสู่หน้าจอตามภาพ



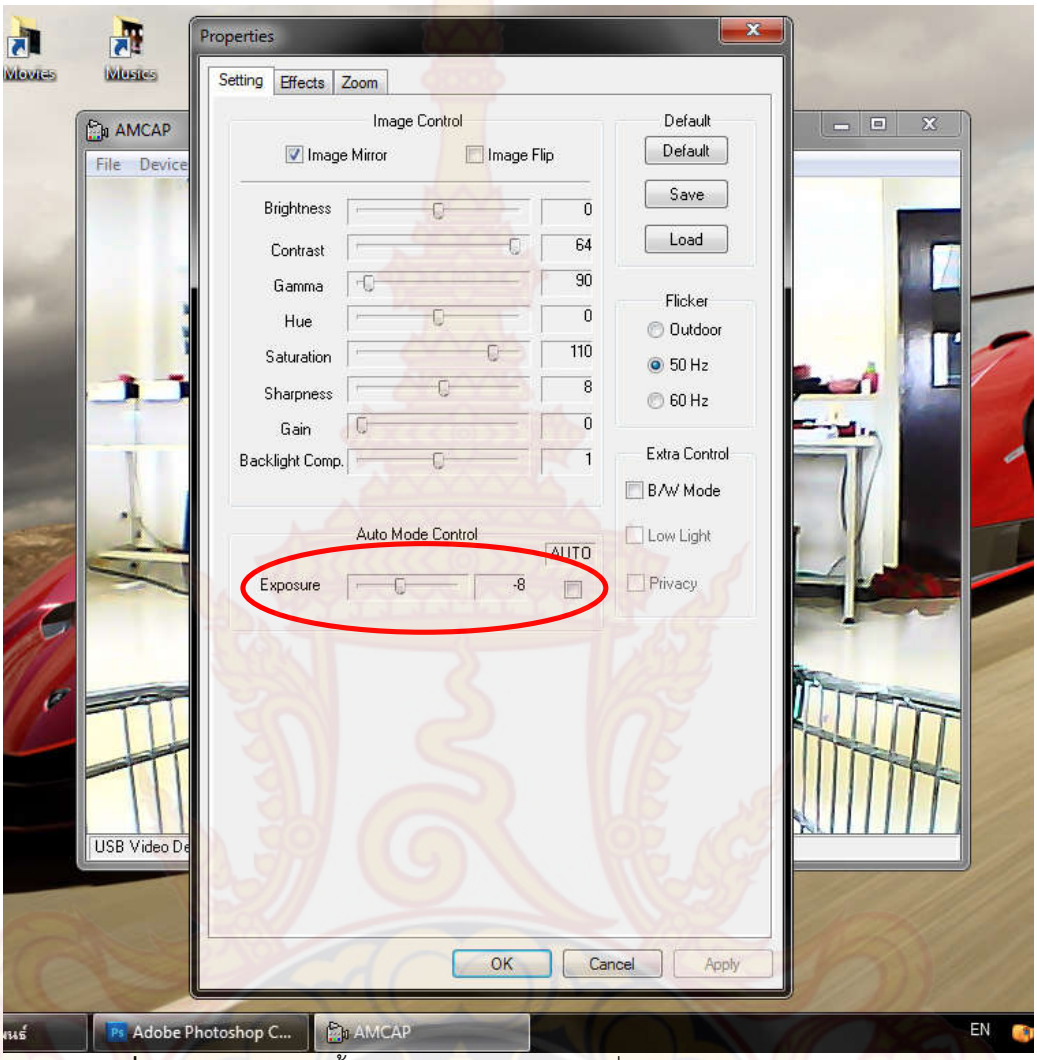
ภาพที่ 4-16 หน้าจอแสดงภาพที่ได้จากกล้อง

3. เลือกที่เมนู Option > Video Capture Filter...



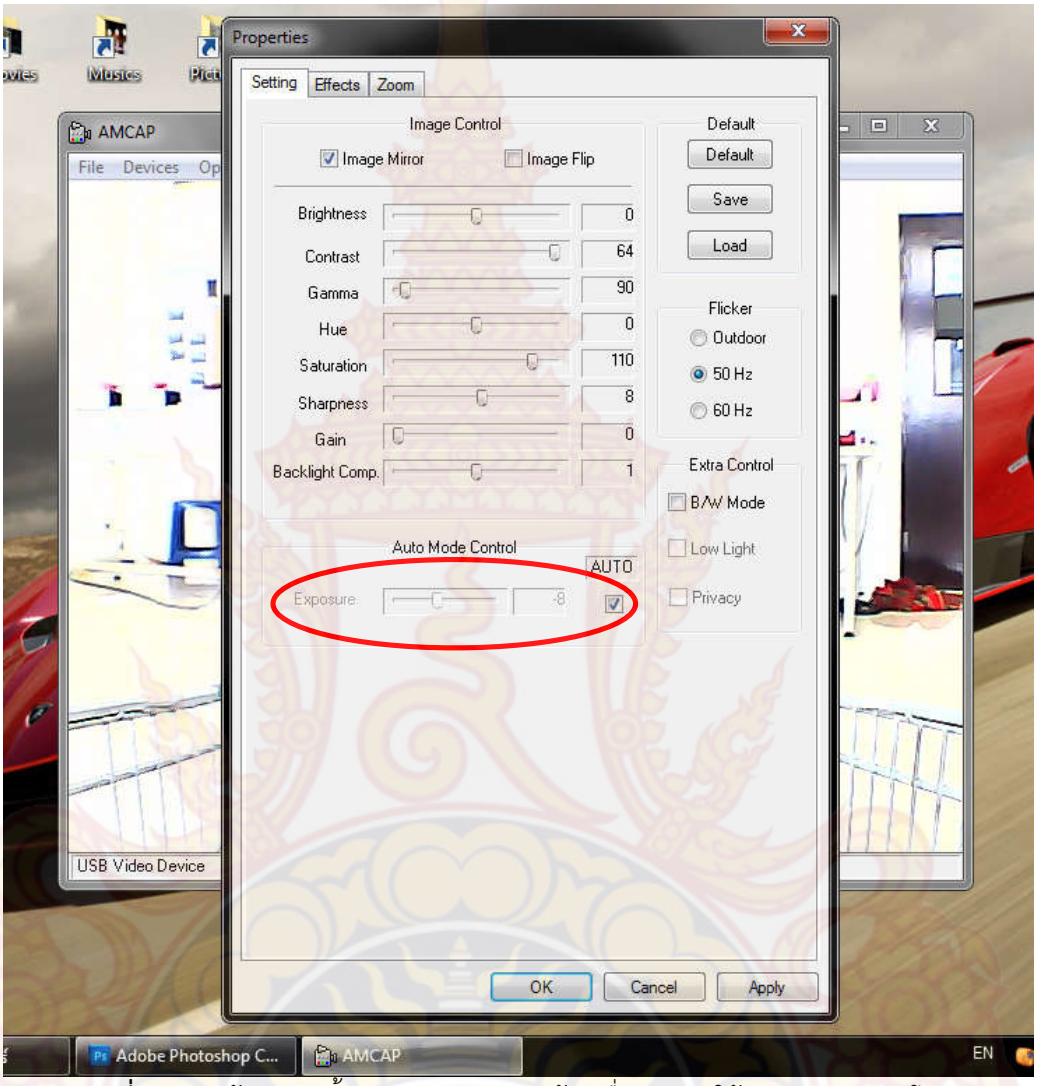
ภาพที่ 4-17 การเข้าไปตั้งค่าการทำงานของกล้อง

4. จะขึ้นหน้าจอตามภาพ ในวงกลมสีแดงจะเป็นส่วนที่ใช้ตั้งค่าชดเชยแสงอัตโนมัติ ซึ่งในภาพเป็นการปิดการใช้ค่าชดเชยแสงอัตโนมัติ ทำให้สามารถเลือกระดับค่าชดเชยแสงอัตโนมัติด้วยตนเองได้



ภาพที่ 4-18 หน้าจอการตั้งค่าการทำงานของกล้องเมื่อปิดการใช้ค่าชดเชยแสงอัตโนมัติ

5. ภาพนี้จะเป็นการเปิดใช้ค่าชดเชยแสงอัตโนมัติ



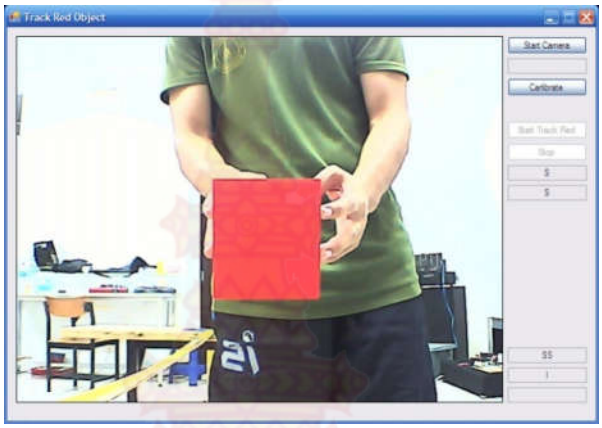
ภาพที่ 4-19 หน้าจอการตั้งค่าการทำงานของกล้องเมื่อเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

4.2 วัสดุอุปกรณ์ที่ใช้

1. กล้องเว็ปแคม 1 ตัว
2. โมดูลประมวลผลภาพ (Image Processing) 1 ชุด
3. ตลับเมตร 1 อัน
4. เป้าสี่เหลี่ยมสีแดงขนาด12.5 X 11 cm 1 อัน

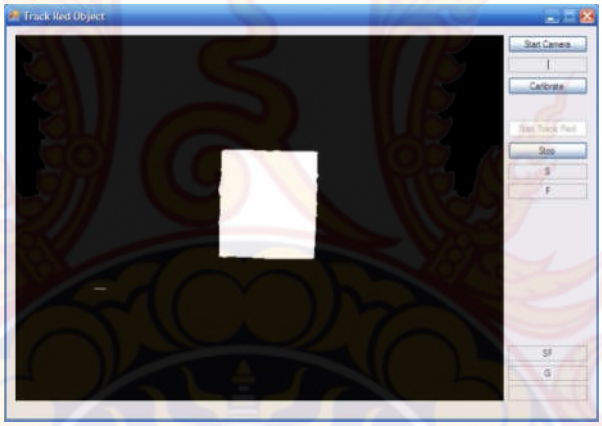
4.3 การทดสอบเมื่อปิดการใช้ค่าชดเชยแสงอัตโนมัติ (Non-Auto Exposure)

1. ระยะ 60 cm จากกล้อง



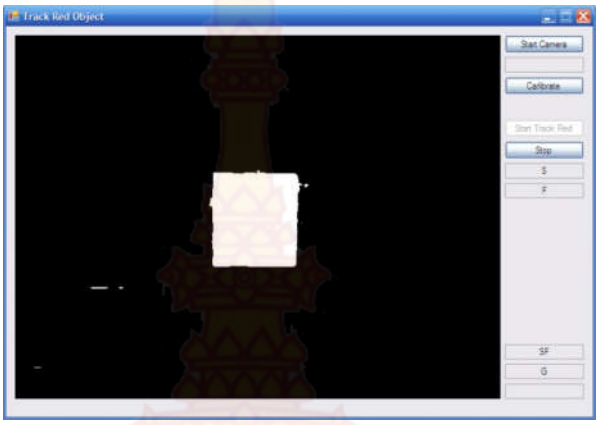
ภาพที่ 4-20 การนำเป้าสีแดงไว้ในระยะ 60 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

2. ระยะ 65 cm จากกล้อง



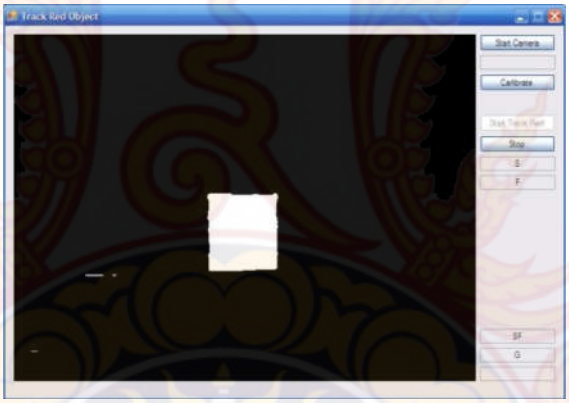
ภาพที่ 4-21 การนำเป้าสีแดงไว้ในระยะ 65 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

3. ระยะ 72 cm จากกล้อง



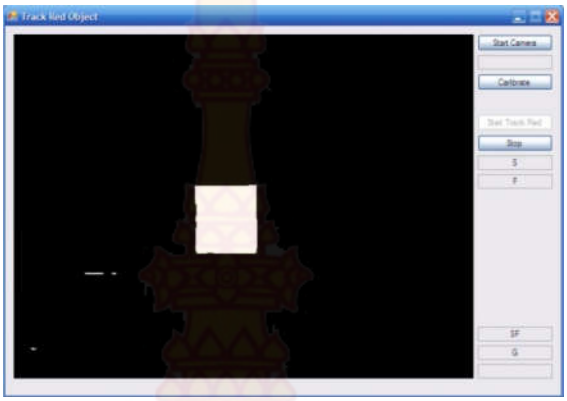
ภาพที่ 4-22 การนำเป้าสีแดงไว้ในระยะ 72 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

4. ระยะ 85 cm จากกล้อง



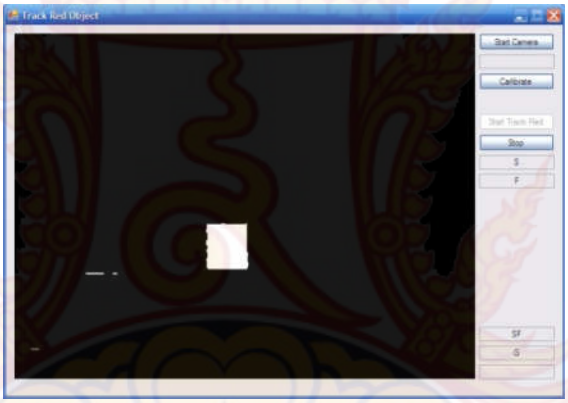
ภาพที่ 4-23 การนำเป้าสีแดงไว้ในระยะ 85 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

5. ระยะ 100 cm จากกล้อง



ภาพที่ 4-24 การนำเป้าสีแดงไว้ในระยะ 100 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

6. ระยะ 150 cm จากกล้อง



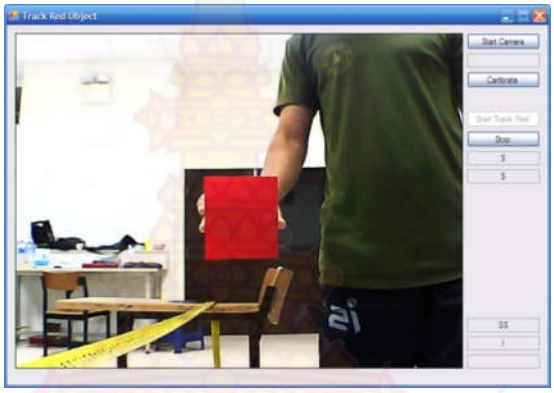
ภาพที่ 4-25 การนำเป้าสีแดงไว้ในระยะ 150 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

7. ระยะ 160 cm จากกล้อง

ภาพที่ 4-26 การนำเป้าสีแดงไว้ในระยะ 160 cm จากกล้องโดยปิดการใช้ค่าชดเชยแสงอัตโนมัติ

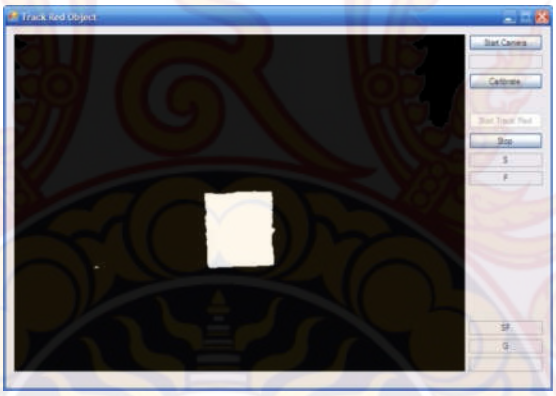
4.4 การทดสอบเมื่อเปิดใช้การใช้ค่าชดเชยแสงอัตโนมัติ (Auto Exposure)

1. ระยะ 80 cm จากกล้อง



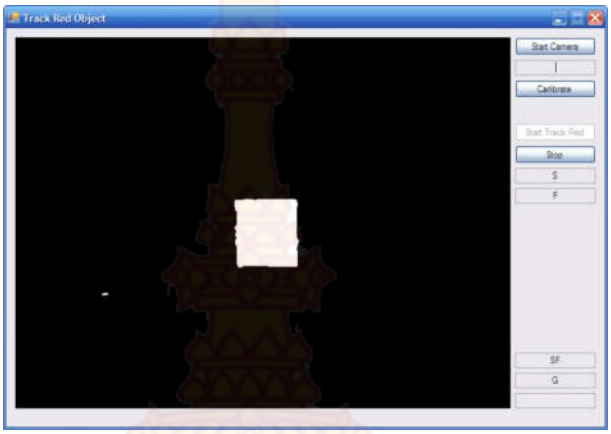
ภาพที่ 4-27 การนำเป้าสีแดงไว้ในระยะ 80 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

2. ระยะ 85 cm จากกล้อง



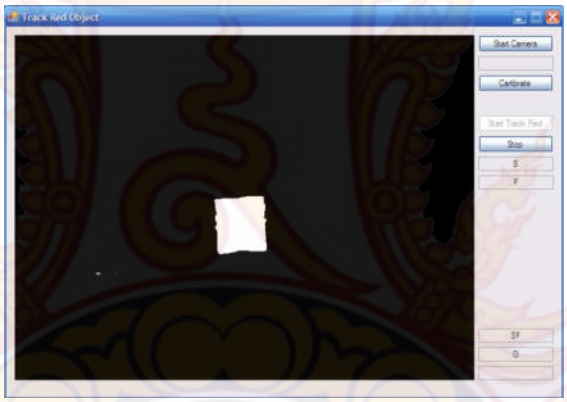
ภาพที่ 4-28 การนำเป้าสีแดงไว้ในระยะ 85 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

3. ระยะ 105 cm จากกล้อง



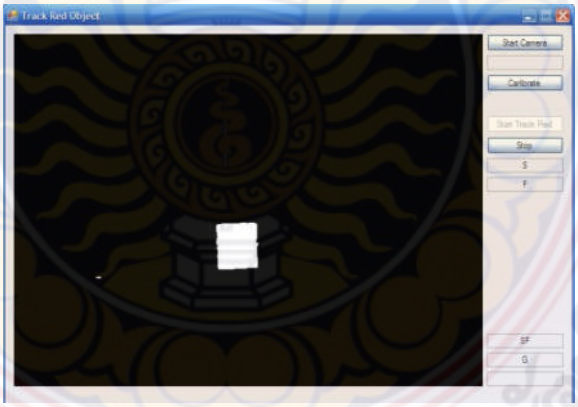
ภาพที่ 4-29 การนำเป้าสีแดงไว้ในระยะ 105 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

4. ระยะ 120 cm จากกล้อง



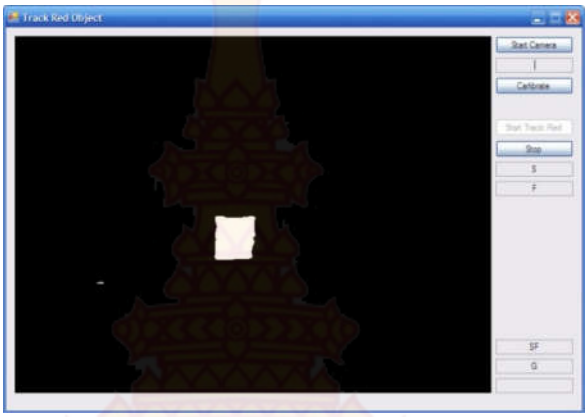
ภาพที่ 4-30 การนำเป้าสีแดงไว้ในระยะ 120 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

5. ระยะ 150 cm จากกล้อง



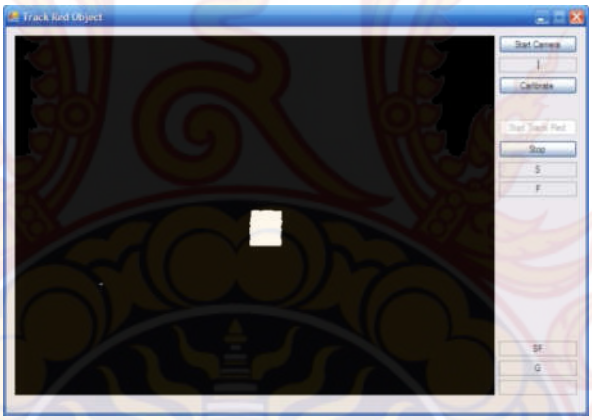
ภาพที่ 4-31 การนำเป้าสีแดงไว้ในระยะ 150 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

6. ระยะ 160 cm จากกล้อง



ภาพที่ 4-32 การนำเป้าสีแดงไว้ในระยะ 160 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

7. ระยะ 180 cm จากกล้อง



ภาพที่ 4-33 การนำเป้าสีแดงไว้ในระยะ 180 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

8. ระยะ 185 cm จากกล้อง

ภาพที่ 4-34 การนำเป้าสีแดงไว้ในระยะ 185 cm จากกล้องโดยเปิดการใช้ค่าชดเชยแสงอัตโนมัติ

4.5 ผลการทดสอบ

เมื่อปิดการใช้ค่าชดเชยแสงอัตโนมัติ โปรแกรมและกล้องสามารถจับและประมวลผลภาพเป้าสีแดงได้ตั้งแต่ระยะ 65 cm ไปจนถึงระยะ 150 cm แต่เมื่อเปิดการใช้ค่าชดเชยแสงอัตโนมัติ โปรแกรมและกล้องสามารถจับและประมวลผลภาพเป้าสีแดงได้ไกลขึ้นเล็กน้อยคือตั้งแต่ระยะ 80 cm ไปจนถึงระยะ 180 cm

สรุปคือถ้าปิดการใช้ค่าชดเชยแสงอัตโนมัติควรให้เป้าสีแดงอยู่ห่างจากกล้องไม่น้อยกว่า 65 cm และไม่ควรเกิน 150 cm แต่ถ้าเปิดการใช้ค่าชดเชยแสงอัตโนมัติควรให้เป้าสีแดงอยู่ห่างจากกล้องไม่น้อยกว่า 65 cm และไม่ควรเกิน 180 cm

5. การทดสอบการแบกน้ำหนักของรถเข็นอัตโนมัติ

5.1 วัตถุประสงค์

เพื่อทดสอบว่ารถเข็นอัตโนมัติสามารถบรรทุกน้ำหนักได้เท่าใด โดยที่มอเตอร์ยังสามารถทำงานได้

5.2 วัสดุอุปกรณ์ที่ใช้

1. วงจรไมโครคอนโทรลเลอร์	1	ชุด	
2. วงจรขับมอเตอร์	2	ชุด	
3. แบตเตอรี่ขนาด 12V		2	ก้อน
4. มอเตอร์ไฟฟ้ากระแสตรงขนาด 24V		2	ตัว

5.3 ขั้นตอนการทดสอบ

1. สำรวจการเลือกสินค้าใส่รถเข็นของคนทั่วไปในห้างสรรพสินค้า โดยการสังเกตุจากสายตา เป็นระยะเวลา 1 วัน (9.00 น. –17.00 น.) พบว่าคนทั่วไปเลือกสินค้าใส่รถเข็น น้อยกว่าหรือเท่ากับ 30 กิโลกรัม

2. ทำการทดสอบโดยการใส่ขวดน้ำขนาด 1 ลิตร ที่มีน้ำบรรจุอยู่เต็ม ทั้งหมด 6 แพ็ค แพ็คละ 6 ขวด รวมเป็นจำนวนทั้งหมด 36 ขวด แต่ละขวดมีน้ำหนัก 1 กิโลกรัม ดังนั้นน้ำหนักรวมที่ใส่ไว้ในรถเข็นทั้งหมด 36 กิโลกรัม

ภาพที่ 4-35 การทดสอบการแบกน้ำหนัก

3. ทดสอบการขับมอเตอร์โดยเขียนโค้ดภาษาซีอย่างง่ายในการทดสอบดังนี้

```
#include <16F877.h>
#FUSES NOWDT,NOPROTECT,NOLVP,HS
#USE delay(clock=20M)
#USE fast_io(b)
void main()
{
set_tris_b(0x00);
while(1)
{
 output_b(0x05);          //ให้รถเข็นเดินหน้า
```

```
            delay_ms(3000); //หน่วงเวลา 3 วินาที
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
            output_b(0x0A);          //ให้รถเข็นถอยหลัง
            delay_ms(3000); //หน่วงเวลา 3 วินาที
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
            output_b(0x09);          //ให้รถเข็นเลี้ยวขวา
            delay_ms(3000); //หน่วงเวลา 3 วินาที
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
            output_b(0x06);          //ให้รถเข็นเลี้ยวซ้าย
            delay_ms(3000); //หน่วงเวลา 3 วินาที
            output_b(0x00); //ให้รถเข็นหยุดนิ่ง
            delay_ms(2000); //หน่วงเวลา 2 วินาที
        }
}
```

5.4 ผลการทดสอบ

เมื่อป้อนโค้ดภาษาซีตามด้านบนเข้าไปใน PIC แล้วลองเปิดสวิทซ์ ปรากฏว่ารถเข็นสามารถเดินหน้า, ถอยหลัง, เลี้ยวซ้าย และเลี้ยวขวาได้ โดยบรรทุกน้ำหนัก 36 กิโลกรัม ซึ่งเกินกว่าน้ำหนักโดยทั่วไปในการเลือกสินค้า จึงสรุปว่าในการใช้งานทั่วไป คือใส่สินค้าในรถเข็นไม่เกิน 30 กิโลกรัม รถเข็นสามารถทำงานได้ตามปกติ

บทที่ 5
สรุป อธิปรายผลและข้อเสนอแนะ

1. สรุปผลการวิจัย

โครงงานนี้เป็นการสร้างรถเข็นอัตโนมัติที่ที่ควบคุมโดยใช้การประมวลผลภาพหรือImage Processing เพื่อให้สามารถแล่นตามเป้าสีแดงซึ่งติดอยู่กับตัวผู้ใช้ได้ การทำงานของรถเข็นอัตโนมัติฯ นั้น เริ่มการรับภาพเป้าสีแดงโดยกล้อง ส่งมาประมวลผลภาพที่โปรแกรม Visual C# จากนั้นส่งข้อมูลที่ประมวลผลออกมายังภาครับ ซึ่งหมายถึงโมดูลไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะส่งข้อมูลไปให้ชุดวงจรขับมอเตอร์ทั้งสองชุดเพื่อขับมอเตอร์ตามการเคลื่อนที่ของเป้าสีแดง ซึ่งในการทดสอบ โมดูลประมวลผลภาพจะยังมีปัญหาในด้านของการประมวลผลเพื่อส่งค่าถอยหลัง ซึ่งทางคณะผู้จัดทำได้ตั้งสมมติฐานว่า มีสาเหตุมาจากคุณภาพของกล้อง ซึ่งกล้องที่คณะผู้จัดทำใช้ทดสอบนั้น เป็นเพียงกล้องเว็ปแคมธรรมดาที่มีขายตามท้องตลาดทั่วไป เนื่องมาจากกำลังทรัพย์ที่มีจำกัดและสภาพแสงแวดล้อม ส่วนโมดูลอื่นๆรวมทั้งชุดวงจรขับมอเตอร์นั้น สามารถทำงานได้อย่างถูกต้องเป็นไปตามวัตถุประสงค์และขอบเขตที่ตั้งไว้นอกจากนี้ผู้จัดทำยังได้นำรถเข็นอัตโนมัติไปทดลองใช้จริงโดยให้ผู้ใช้ในห้างสรรพสินค้าทำแบบสอบถามความพึงพอใจในการใช้งานจำนวน 5 คนซึ่งแบบสอบถามสามารถดูได้ในภาคผนวก ข

2. การอภิปรายผลและข้อเสนอแนะ

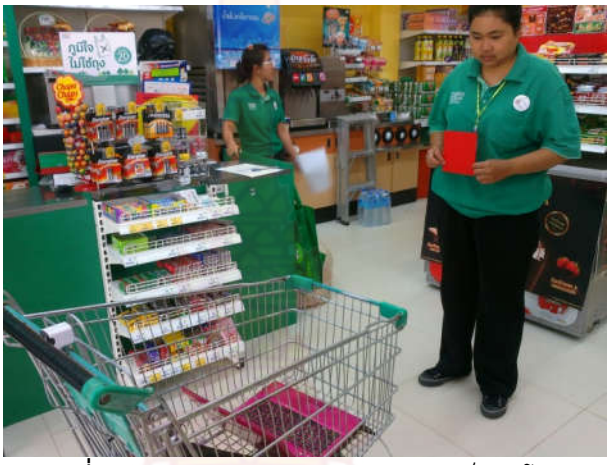
2.1 กล้องที่ใช้เป็นกล้องเว็ปแคมธรรมดาที่มีขายตามท้องตลาดทั่วไป คุณภาพและประสิทธิภาพในการจับภาพอาจจะไม่ดีเท่าที่ควร ผู้ใช้สามารถนำกล้องที่มีประสิทธิภาพมากกว่านี้ ซึ่งอาจมีราคาสูงมาก มาใช้แทนได้

2.2 เมื่อใช้งานไประยะเวลาหนึ่ง แบตเตอรี่ที่ติดอยู่กับตัวรถเข็นฯอาจจะหมด ทางคณะผู้จัดทำได้ทำการติดตั้งอุปกรณ์ชาร์จไว้ที่ด้านบนของแบตเตอรี่ ผู้ใช้สามารถชาร์จไฟให้กับแบตเตอรี่ได้โดยใช้ไฟ AC 220V ทั่วไป

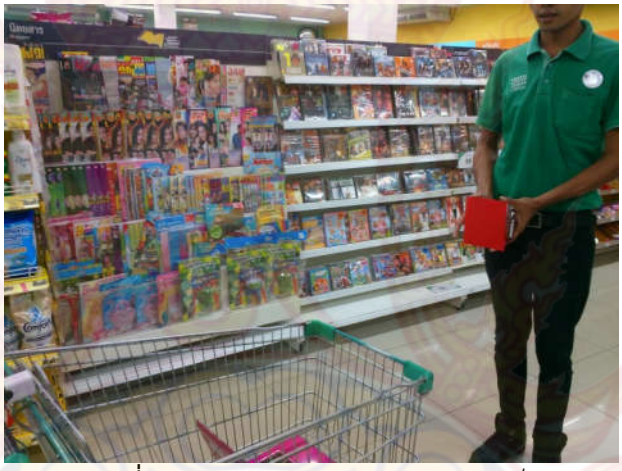
3. การนำไปใช้ประโยชน์

3.1 กลุ่มเป้าหมายคือพนักงานเทสโก้ โลตัสเอ็กซ์เพลส

3.2 ทดลองใช้โดยพนักงานเทสโก้ โลตัสเอ็กซ์เพลส จำนวน 5 คน และทดสอบด้วยตัวเอง ดังภาพที่ 5-1 ถึง 5-3



ภาพที่ 5-1 การทดสอบการทำงานของรถเข็นอัตโนมัติ



ภาพที่ 5-2 การทดสอบการทำงานของรถเข็น

ภาพที่ 5-3 การทดสอบการทำงานของรถเข็น

บรรณานุกรม

1. รศ.ดร.สมเกียรติ อุดมหรรษากุล, **การประมวลผลภาพดิจิตอลเบื้องต้น**, กรุงเทพฯ :สำนักพิมพ์ท้อป, 2554.
2. ธนากร สุนทรวัฒน์, **ปฏิบัติการไมโครโปรเซสเซอร์**, นครปฐม : มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์, 2552.
3. สาธิต วานิชชัยกิต, การใช้งาน OpenCV Library เบื้องต้นเพื่อการประมวลผลภาพจากกล้อง, ออนไลน์ [23 กันยายน 2552], สาระสังเขป สืบค้นจาก :http://fibo.kmutt.ac.th/blog/.
4. opencvsharp - OpenCV wrapper for .NET Framework, ออนไลน์ [20 มกราคม 2556], สาระสังเขป สืบค้นจาก : http://code.google.com/p/opencvsharp/.
5. Image processing &Computervision, ออนไลน์[25 มีนาคม 2556], สาระสังเขป สืบค้นจาก :http://www.shadowwares.com/.
6. ไมโครคอนโทรลเลอร์ PIC, ออนไลน์[11 มกราคม 2556], สาระสังเขป สืบค้นจาก :http://ไมโครคอนโทรลเลอร์.blogspot.com/.
7. Jan Slama, String Format for Double [C#], online [2013, February 10], Available from :http://www.csharp-examples.net/string-format-double/.
8. Jubpas, การแปลง ASCII to Char และ Char to ASCII, ออนไลน์ [15 มกราคม 2554], สาระสังเขป สืบค้นจาก :http://www.it-devzone.com/how-to-ascii-to-char-and-char-to-ascii/.
9. Mazcedure, C# เขียนติดต่อ SerialPort (RS232) อย่างง่าย, ออนไลน์ [21 ตุลาคม 2555], สาระสังเขป สืบค้นจาก : http://codetoday.net/archives/901.
10. สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี, คู่มืออบรมการเขียนโปรแกรมด้วยภาษา C#, ออนไลน์ [5 พฤษภาคม 2556], สาระสังเขป สืบค้นจาก :http://www.scribd.com/.
11. การเชื่อมต่อพอร์ตอนุกรมด้วย C#, ออนไลน์ [14 สิงหาคม 2553], สาระสังเขป สืบค้นจาก : http://www.nawattakam.com/talk/index.php?topic=220.0.
12. บัญชา ปะสีละเตสัง, **พัฒนาแอปพลิเคชั่นด้วย Visual C# 2010**, กรุงเทพฯ :สำนักพิมพ์ ซีเอ็ดยูเคชั่น, 2554.
13. S. McKenna, S. Jabri, Z. Doric, H. Wechsler and A. Rosenfeldf, “Tracking Groups of People,” **Journal of Computer Vision and Image Understanding**, Vol. 80, No.1, pp. 42-56, 2000.

ภาคผนวก

ชุดคำสั่งงาน

ชุดคำสั่งที่ใช้ในการประมวลผลภาพ

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
```

เรียกใช้ Library ของ Visual C#

```
using OpenCvSharp;
using OpenCvSharp.Blob;
using OpenCvSharp.MachineLearning;
using OpenCvSharp.DebuggerVisualizers;
using OpenCvSharp.UserInterface;
```

เรียกใช้ Library ของ OpenCV Sharp

```
namespace track_red_object_winCSharp //กำหนดชื่อโปรแกรม
{
publicpartialclassForm1 : Form
   {
privateBitmap bitmap = null;        //กำหนดตัวแปร bitmap
privateIplImage img = null;                 //กำหนดตัวแปร img
privateIplImage img1 = null;        //กำหนดตัวแปร img1
privateIplImage segment = null;     //กำหนดตัวแปร segment
privateIplImage imgLabel = null;    //กำหนดตัวแปร imgLabel
privateCvCapture capture = null;    //กำหนดตัวแปร capture
privateint CntObj = 0;              //กำหนดตัวแปร Cntobj
privateuint Max_Area = 10000;       //กำหนดตัวแปร Max_Area
privateuint Min_Area = 800;                 //กำหนดตัวแปร Min_Area
privateuint Area1 = 0;              //กำหนดตัวแปร Area1
privateuint Area2 = 0;              //กำหนดตัวแปร Area2
privatedouble MemlengthLine = 0;            //กำหนดตัวแปร MemlengthLine
privatedouble AreaPERWidth = 0; //กำหนดตัวแปร AreaPERWidth
privatedouble ValTurn = 0;                  //กำหนดตัวแปร ValTurn
```

```
privatedouble r1 = 0, g1 = 0, b1 = 0;          //กำหนดตัวแปร r1, g1, b1
privatedouble ra = 0, ga = 0, ba = 0;          //กำหนดตัวแปร ra, ga, ba
privatedouble rc, gc, bc;  //กำหนดตัวแปร rc, gc, bc
public Form1()
        {
            InitializeComponent();
        }

privatevoid From_Load(object sender, EventArgs e) //กำหนดการทำงานเมื่อโหลดโปรแกรมขึ้นมา
        {
// set default size mode of picture box
            pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage; //ตั้งค่าโหมดรูปภาพใน
picturebox
this.MaximizeBox = false; //ตั้งค่า maximize ของฟอร์มเป็น false
            img = newIplImage(640, 480, BitDepth.U8, 3); //กำหนดการทำงานของตัวแปร img
            img1 = newIplImage(img.Size, BitDepth.U8, 3); //กำหนดการทำงานของตัวแปร img1
            segment = newIplImage(img.Size, BitDepth.U8, 1); //กำหนดการทำงานของตัวแปร segment
            imgLabel = newIplImage(img.Size, BitDepth.F32, 1); //กำหนดการทำงานของตัวแปร imgLabel
        }

privatevoid StartCam_Click(object sender, EventArgs e)//กำหนดการทำงานเมื่อกดปุ่ม StartCam
        {
try
            {
                capture = CvCapture.FromCamera(1); //รับภาพจากกล้องเก็บไว้ที่ตัวแปร capture
                StartCam.Enabled = false;//กล้องเริ่มทำงาน
                timer3.Start(); //เริ่มการทำงานของ timer3
            }
catch
            {
MessageBox.Show("Error", "No cameras found", MessageBoxButtons.OK, MessageBoxIcon.Error); //
แสดงกล่องข้อความ error กรณีไม่ได้เชื่อมต่อกล้อง
            }
        }

privatevoid Carlibrate_Click(object sender, EventArgs e)//กำหนดการทำงานเมื่อกดปุ่ม Carlibrate
```

```
        {
            Carlibrate.Enabled = false;//ปุ่ม Carlibrate ไม่สามารถใช้งานได้
            rc = ra;//กำหนดตัวแปร rc=ra
            gc = ga; //กำหนดตัวแปร gc=ga
            bc = ba; //กำหนดตัวแปร bc=ba
            timer3.Stop(); //หยุดการทำงานของ timer3
        }

    privatevoid Start_Click(object sender, EventArgs e)//กำหนดการทำงานเมื่อกดปุ่ม Start
        {
    try
            {
                capture = CvCapture.FromCamera(1); //รับภาพจากกล้องเก็บไว้ที่ตัวแปร capture
                Start.Enabled = false; //ไม่อนุญาตให้ใช้ปุ่ม Start
                Stop.Enabled = true;//อนุญาตให้ใช้ปุ่ม Stop
                timer1.Start(); //เริ่มการทำงานของ timer1
                timer2.Start(); //เริ่มการทำงานของ timer2
                serialPort1.Open(); //เริ่มการทำงานของ SerialPort1
            }
    catch
            {
    MessageBox.Show("Error", "No cameras found", MessageBoxButtons.OK, MessageBoxIcon.Error); //
    แสดงกล่องข้อความ error กรณีไม่ได้เชื่อมต่อกล้อง
            }
        }

    privatevoid From_Close(object sender, FormClosingEventArgs e) //กำหนดการทำงานเมื่อปิดโปรแกรม
        {
            timer1.Stop();//หยุดการทำงานของ timer1
            timer2.Stop(); //หยุดการทำงานของ timer2
            timer3.Stop(); //หยุดการทำงานของ timer3
        }

    privatevoid timer1_Tick(object sender, EventArgs e) //กำหนดการทำงานเมื่อ timer1 ทำงาน
        {
            img1 = capture.RetrieveFrame();//ตัวแปร img1 เก็บค่าจากตัวแปร capture ทีละหนึ่งเฟรม
```

```
if (img1.CvPtr != null)//กำหนดเงื่อนไขเช็คค่า img1 เมื่อเข้าฟังก์ชัน CvPtr ไม่เท่ากับ null
        {
Cv.Smooth(img1, img, SmoothType.Gaussian, 3);//นำตัวแปร img1 มาขจัดสัญญาณรบกวนในภาพ แล้วเก็บค่าไว้ในตัวแปร img
for (int y = 0; y < img.Height; y++) //กำหนดตัวแปร y เช็คค่าในแนวตั้งของภาพทีละหลักไปเรื่อยๆ
for (int x = 0; x < img.Width; x++)//กำหนดตัวแปร x เช็คค่าในแนวนอนของภาพทีละแถวไปเรื่อยๆ
                {
CvScalar c = img[y, x];//กำหนดตัวแปร c แสดงค่าตัวแปร img
double b = ((double)c.Val0) / 255.0; //กำหนดตัวแปร b แทนสีน้ำเงินมีค่า 0-255
double g = ((double)c.Val1) / 255.0; //กำหนดตัวแปร g แทนสีเขียวมีค่า 0-255
double r = ((double)c.Val2) / 255.0;//กำหนดตัวแปร r แทนสีแดงมีค่า 0-255
if ((r > 0.34 + b) && (r > 0.34 + g))//กำหนดเงื่อนไขเช็คค่า b,g,r
//if (((r > ra - 0.2) && (r < ra + 0.2)) && ((g > ga - 0.2) && (g < ga + 0.2)) && ((b > ba - 0.2) && (b < ba + 0.2)))//กำหนดเงื่อนไขเช็คค่า b,g,r สามารถสลับใช้ได้กับเงื่อนไขบน
                    {
                        img[y, x] = newCvColor//เปลี่ยนแปลงค่าสีของ img ใหม่
                        {
                            B = 255, //ให้สีน้ำเงินมีค่า 255
                            G = 255, //ให้สีเขียวมีค่า 255
                            R = 255, //ให้สีแดงมีค่า 255
                        };
                    }
else
                    {
                        img[y, x] = newCvColor//เปลี่ยนแปลงค่าสีของ img ใหม่
                        {
                            B = 0, //ให้สีน้ำเงินมีค่า 0
                            G = 0, //ให้สีเขียวมีค่า 0
                            R = 0, //ให้สีแดงมีค่า 0
                        };
                    }
                }
Cv.CvtColor(img, segment, ColorConversion.BgrToGray); //แปลงค่าสีในตัวแปร img เป็นสีขาวดำ แล้วเก็บค่าในตัวแปร segment
IplImage tmpImg = segment.Clone(); //คัดลอกค่าตัวแปร segment เก็บไว้ในตัวแปร tmpImg
```

```
IplImage dstImgOpening = segment.Clone(); //คัดลอกค่าตัวแปร segment เก็บไว้ในตัวแปร dstImgOpening
IplConvKernel element = Cv.CreateStructuringElementEx(3, 3, 1, 1, ElementShape.Rect, null); //ใช้ฟังก์ชัน IplConvKernel สร้างสี่เหลี่ยมล้อมรอบพื้นที่สีขาว
Cv.MorphologyEx(segment, dstImgOpening, tmpImg, element, MorphologyOperation.Open, 1); //ใช้ฟังก์ชัน Morphology เพื่อปรับขอบรูปสี่เหลี่ยมให้เรียบ
CvBlobs blobs = newCvBlobs(); //สร้างตัวแปร blobs เพื่อเก็บค่าแบบ Blobs
uint result = blobs.Label(dstImgOpening, imgLabel); //กำหนดตัวแปร result เพื่อเก็บค่าตัวแปร blobs ซึ่งได้จากการใช้ฟังก์ชัน Label กับตัวแปร dstImgOpening กับ imgLabel
            tmpImg.ReleaseData(); //ลบข้อมูลของ tmpImg
  dstImgOpening.ReleaseData(); //ลบข้อมูลของ dstImgOpening
            blobs.FilterByArea(Min_Area, Max_Area); //นำค่าตัวแปร blobs มาประมวลผลโดยฟังก์ชัน FilterByArea แล้วเก็บค่าที่ตัวแปร Min_Area และ Max_Area
int turn = 0; //สร้างตัวแปร turn=0
Cv.CvtColor(segment, img, ColorConversion.GrayToBgr); //ใช้ฟังก์ชัน CvtColor แปลงค่าสี่ของตัวแปร segment เป็น B,G,R แล้วเก็บค่าที่ตัวแปร img
        CntObj = blobs.Count; //
double x1 = 0, y1 = 0, x2 = 0, y2 = 0; //สร้างตัวแปร x1,y1,x2,y2
double lengthLine = 0; //สร้างตัวแปร lengthLine
foreach (var item in blobs)
            {
CvBlob b = item.Value; //สร้างตัวแปร b เก็บค่า item
                Area1 = b.Area; //ใช้ฟังก์ชัน Area กับตัวแปร b แล้วเก็บค่าที่ Area1
                ValTurn = b.Centroid.X; //ใช้ฟังก์ชัน Centroid เช็คค่าในแกน X กับตัวแปร b แล้วเก็บค่าในตัวแปร ValTurn
if (item.Value.Centroid.X > 500) //กำหนดเงื่อนไขเช็คค่า item ในแกน X ว่ามากกว่า 500 หรือไม่
                    turn = 1; //ให้ตัวแปร turn=1
elseif (item.Value.Centroid.X < 140 && item.Value.Centroid.X >= 0) //กำหนดเงื่อนไขเช็คค่า item ในแกน X ว่าน้อยกว่า 140 และมากกว่าหรือเท่ากับ 0 หรือไม่
                    turn = 2; //ให้ตัวแปร turn=2
else
                    turn = 0;//ให้ตัวแปร turn=0
CvBlobLib.CentralMoments(b, imgLabel); //ใช้ฟังก์ชัน CentralMoments กับตัวแปร b แล้วเก็บค่าใน imgLabel
double angle = CvBlobLib.Angle(b); //ใช้ฟังก์ชัน Angle กับตัวแปร b แล้วเก็บค่าใน angle
```

```
                lengthLine = Math.Max(b.MaxX - b.MinX, b.MaxY - b.MinY) / 2.0; //คำนวณค่าใน
แนวแกน X,Y แล้วเก็บค่าในตัวแปร lengthLine
                x1 = b.Centroid.X - lengthLine * Math.Cos(angle);
                y1 = b.Centroid.Y - lengthLine * Math.Sin(angle);
                x2 = b.Centroid.X + lengthLine * Math.Cos(angle);
                y2 = b.Centroid.Y + lengthLine * Math.Sin(angle);
                x1 = x2 - x1; //ให้ x1=x2-x1
                y1 = y2 - y1; //ให้ y1=y2-y1
                lengthLine = Math.Sqrt(Math.Pow(x1, 2.0) + Math.Pow(y1, 2.0)); //นำค่า x1,y1 มา
คำนวณแล้วเก็บค่าในตัวแปร lengthLine
            }
if (bitmap == null) //กำหนดเงื่อนไขเช็คค่า bitmap=null
                bitmap = img1.ToBitmap(); //แปลงค่า img1 ให้เป็นภาพแบบ Bitmap แล้วเก็บค่าใน bitmap
            blobs.RenderBlobs(imgLabel, img1, img); //ใช้ฟังก์ชัน RenderBlobs กับตัวแปร imgLabel,
img1, img
            blobs.Release(); //ลบข้อมูลตัวแปร blobs

if (CntObj == 1) //กำหนดเงื่อนไขเช็คค่า CntObj=1
            {
BitmapConverter.ToBitmap(img, bitmap); //แปลงค่า img ให้เป็นภาพแบบ Bitmap แล้วเก็บค่าใน bitmap
if (Start.Enabled == false&& Analyze.Enabled == false) //กำหนดเงื่อนไขเช็คค่าปุ่ม Start และ Analyze
                {
                    Area2 = Area1; //กำหนด Area2=Area1
                    MemlengthLine = lengthLine; //กำหนด MemlengthLine=lengthLine
if (Area2 < 10000) //กำหนดเงื่อนไขเช็คค่า Area2<10000
                    {
                        Min_Area = 0; //กำหนด Min_Area=0
                    }
else
                    {
                        Min_Area = Area2 - 10000; //กำหนด Min_Area=Area2-10000
                    }
                    Max_Area = Area2 + 10000; //กำหนด Max_Area=Area2+10000
                    Analyze.Enabled = true; //อนุญาตให้ใช้งานปุ่ม Analyze ได้
                    AreaPERWidth = Area2 / MemlengthLine; //กำหนด AreaPERWidth เท่ากับ Area2
หารด้วย MemlengthLine
```

การคำนวณค่า x1,y1,x2,y2

```
                }
            }
else
            {
BitmapConverter.ToBitmap(img1, bitmap);//แปลงค่า img1 ให้เป็นภาพแบบ Bitmap แล้วเก็บค่าใน bitmap
            }
if (CntObj == 1 && (MemlengthLine - 100) < lengthLine && (MemlengthLine + 100) > lengthLine)
//กำหนดเงื่อนไขเช็คค่า CntObj , MemlengthLine , lengthLine
            {
if (turn == 1) //กำหนดเงื่อนไขเช็คค่า turn=1
                textBox1.Text = "L"; //textbox1 แสดงค่า L
elseif (turn == 2) //กำหนดเงื่อนไขเช็คค่า turn=2
                textBox1.Text = "R"; //textbox1 แสดงค่า R
else
                textBox1.Text = "S"; //textbox1 แสดงค่า S

if (Area1 > Area2 && Area2 > 0) //กำหนดเงื่อนไขเช็คค่า Area1>Area2 และArea>0
                {
                textBox2.Text = "B"; //textbox2 แสดงค่า B
                }
else
                {
                textBox2.Text = "F"; //textbox2 แสดงค่า F
                }
            }
else
            {
                textBox1.Text = "S"; //textbox1 แสดงค่า S
                textBox2.Text = "S"; //textbox2 แสดงค่า S
            }
            pictureBox1.Image = (Bitmap)bitmap; //แสดงภาพใน picturebox1
        }
    }

privatevoid Stop_Click(object sender, EventArgs e) //กำหนดการทำงานเมื่อกดปุ่ม Stop
    {
```

```
            Stop.Enabled = false; //ไม่อนุญาตให้ใชปุ่ม Stop
            Start.Enabled = true; //อนุญาตให้ใช้ปุ่ม Start
            StartCam.Enabled = true; //อนุญาตให้ใช้ปุ่ม StartCam
            Carlibrate.Enabled = true; //อนุญาตให้ใช้ปุ่ม Carlibrate
            Analyze.Enabled = true; //อนุญาตให้ใช้ปุ่ม Analyze
            timer1.Stop(); //หยุดการทำงานของ timer1
            timer2.Stop(); //หยุดการทำงานของ timer2
            timer3.Stop();//หยุดการทำงานของ timer3
if (serialPort1.IsOpen) //กำหนดเงื่อนไขเช็คค่า serialPort1
            {
                serialPort1.Close(); //ปิดการส่งข้อมูลผ่าน serialPort1
            }
        }

privatevoid timer2_Tick(object sender, EventArgs e)//กำหนดการทำงานเมื่อ timer2 ทำงาน
        {
string rx1 = textBox1.Text;//สร้างตัวแปร rx1 เก็บค่าใน textbox1
string rx2 = textBox2.Text; //สร้างตัวแปร rx2 เก็บค่าใน textbox2
            textBox3.Text = rx1 + rx2; //นำ rx1+rx2 แล้วแสดงใน textbox3

if (textBox3.Text == "LF") //กำหนดเงื่อนไขเช็คค่าใน textbox3=LF
            {
                textBox4.Text = "A"; //กำหนด textbox4 แสดงค่า A
            }
elseif (textBox3.Text == "LB") //กำหนดเงื่อนไขเช็คค่าใน textbox3=LB
            {
                textBox4.Text = "B"; //กำหนด textbox4 แสดงค่า B
            }
elseif (textBox3.Text == "RF") //กำหนดเงื่อนไขเช็คค่าใน textbox3=RF
            {
                textBox4.Text = "C"; //กำหนด textbox4 แสดงค่า C
            }
elseif (textBox3.Text == "RB") //กำหนดเงื่อนไขเช็คค่าใน textbox3=RB
            {
                textBox4.Text = "D"; //กำหนด textbox4 แสดงค่า D
            }
```

```
elseif (textBox3.Text == "LS") //กำหนดเงื่อนไขเช็คค่าใน textbox3=LS
        {
            textBox4.Text = "E"; //กำหนด textbox4 แสดงค่า E
        }
elseif (textBox3.Text == "RS") //กำหนดเงื่อนไขเช็คค่าใน textbox3=RS
        {
            textBox4.Text = "F"; //กำหนด textbox4 แสดงค่า F
        }
elseif (textBox3.Text == "SF") //กำหนดเงื่อนไขเช็คค่าใน textbox3=SF
        {
            textBox4.Text = "G"; //กำหนด textbox4 แสดงค่า G
        }
elseif (textBox3.Text == "SB") //กำหนดเงื่อนไขเช็คค่าใน textbox3=SB
        {
            textBox4.Text = "H"; //กำหนด textbox4 แสดงค่า H
        }
elseif (textBox3.Text == "SS") //กำหนดเงื่อนไขเช็คค่าใน textbox3=SS
        {
            textBox4.Text = "I"; //กำหนด textbox4 แสดงค่า I
        }
else
        {
            textBox4.Text = "S"; //กำหนด textbox4 แสดงค่า S
        }

if (serialPort1.IsOpen) //กำหนดเงื่อนไขเช็คค่า serialPort1
        {
            serialPort1.Write(textBox4.Text.ToString() + "\n\r");//นำค่าที่แสดงใน textbox4 แปลงเป็น
ข้อมูล string แล้วส่งออกทาง serialPort1
        }
    }

privatevoid timer3_Tick(object sender, EventArgs e)//กำหนดการทำงานเมื่อ timer3 ทำงาน
    {
        img1 = capture.RetrieveFrame(); //ตัวแปร img1 เก็บค่าจากตัวแปร capture ทีละหนึ่งเฟรม
        r1 = 0;//กำหนด r1=0
```

```
            g1 = 0;//กำหนด g1=0
            b1 = 0;//กำหนด b1=0
if (img1.CvPtr != null) //กำหนดเงื่อนไขเช็คค่า img1 เมื่อเข้าฟังก์ชัน CvPtr ไม่เท่ากับ null
            {
Cv.Smooth(img1, img, SmoothType.Gaussian, 3); //นำตัวแปร img1 มาขจัดสัญญาณรบกวนในภาพ แล้วเก็บค่าไว้ในตัวแปร img
for (int y = img.Height / 3; y < img.Height * 2 / 3; y++) //กำหนดตัวแปร y=จำนวนพิกเซลแนวตั้งของภาพแบ่งเป็นสามส่วน แล้วเช็คค่าที่ส่วนกึ่งกลางของภาพ
for (int x = img.Width / 3; x < img.Width * 2 / 3; x++) //กำหนดตัวแปร x=จำนวนพิกเซลแนวนอนของภาพแบ่งเป็นสามส่วน แล้วเช็คค่าที่ส่วนกึ่งกลางของภาพ
                    {
CvScalar c = img[y, x];//กำหนดตัวแปร c แสดงค่าตัวแปร img
double b = ((double)c.Val0) / 255.0;//กำหนดตัวแปร b แทนสีน้ำเงินมีค่า 0-255
double g = ((double)c.Val1) / 255.0;//กำหนดตัวแปร g แทนสีเขียวมีค่า 0-255
double r = ((double)c.Val2) / 255.0;//กำหนดตัวแปร r แทนสีแดงมีค่า 0-255
                        r1 = r1 + r; //กำหนด r1=r1+r
                        g1 = g1 + g; //กำหนด g1=g1+g
                        b1 = b1 + b; //กำหนด b1=b1+b
                    }
            }
            ra = r1 * 9 / (img.Width * img.Height); //กำหนด ra=r1คูณด้วย9 แล้วหารด้วยความกว้างคูณกับความยาวของภาพ
            ga = g1 * 9 / (img.Width * img.Height);//กำหนด ga=g1คูณด้วย9 แล้วหารด้วยความกว้างคูณกับความยาวของภาพ
            ba = b1 * 9 / (img.Width * img.Height);//กำหนด ba=b1คูณด้วย9 แล้วหารด้วยความกว้างคูณกับความยาวของภาพ
            textBox6.Text = ra.ToString("0.00"); //แปลงค่า ra เป็น string แล้วแสดงใน textbox6
            bitmap = img.ToBitmap();//แปลงค่า img ให้เป็นภาพแบบ Bitmap แล้วเก็บค่าใน bitmap
            pictureBox1.Image = (Bitmap)bitmap; //แสดงภาพใน picturebox1
        }
privatevoid Analyze_Click(object sender, EventArgs e)//กำหนดการทำงานเมื่อกดปุ่ม Analyze
        {
            Analyze.Enabled = false; //ไม่อนุญาตให้ใช้ปุ่ม Analyze
        }
    }
}
```

ชุดคำสั่งที่ใช้รับค่าจากการประมวลผลภาพและขับมอเตอร์

```
#include <16f877.h>                          //กำหนดใช้ library สำหรับ PIC16F877
#fuses HS,NOLVP,NOWDT,NOPROTECT              //ให้ไอซีทำงานความเร็วสูง,ไมโปรแกรมแรงดันต่ำ,ไม่มี
                                             watchdog timer,ป้องกันโค๊ด
#use delay (clock = 20M)                     //Crystal = 20MHz
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)  //ตั้งค่าPIC ให้ baud rate = 9600, ขา C6 เป็นขาส่ง
                                               ข้อมูล, ขา C7 เป็นขารับข้อมูล
char temp;                                   //ประกาศตัวแปร temp ชนิด char

#INT_RDA
voidIntRDA_isr()                             //กำหนดการทำงานให้ Interrupt
{
temp=getc();                                 //เช็คข้อมูลที่เก็บไว้โดยตัวแปร temp ที่ส่งมาทีละตัวอักษร
}

void main()                                  //กำหนดการทำงานของฟังก์ชันหลัก
{
int16 du=300;                                //กำหนดตัวแปร du = 500 ชนิด int16
set_tris_b(0b00000000);                      //set port B to output
enable_interrupts(GLOBAL);                   //เปิดใช้งานฟังก์ชัน interrupt
enable_interrupts(INT_RDA);          //interrupts check data received from rs232

  setup_ccp1(CCP_PWM);                       //เปิดใช้งานฟังก์ชัน pwmที่ขา ccp1
  setup_timer_2(T2_DIV_BY_16,255,1);         //ตั้งค่า timer2
  set_timer2(0);                             //ให้ timer2 เริ่มนับจากศูนย์
du=du+100;                                   //ให้ du เพิ่มค่าทีละ 100
  set_pwm1_duty(du);                         //ตั้งค่า duty cycle เท่ากับ du
delay_ms(200);                               //หน่วงเวลา 200 ms

while(true)                                  //วนลูปเช็คข้อมูลไปเรื่อยๆ
  {
while(temp=='A')                     //ถ้า ตัวแปร temp = A
    {
output_b(0x06);                      //รถเข็นเลี้ยวซ้าย
```

```
delay_ms(600);                  //หน่วงเวลา 600 ms
output_b(0x05);                 //ให้รถเข็นเดินหน้า
delay_ms(500);                  //หน่วงเวลา 500 ms
    }

while(temp=='B')                //ถ้า ตัวแปร temp = B
    {
output_b(0x06);                 //ให้รถเข็นเลี้ยวซ้าย
delay_ms(600);                  //หน่วงเวลา 600 ms
output_b(0x0A);                 //ให้รถเข็นถอยหลัง
delay_ms(500);                  //หน่วงเวลา 500 ms
    }

while(temp=='C')                //ถ้า ตัวแปร temp = C
    {
output_b(0x09);                 //รถเข็นเลี้ยวขวา
delay_ms(600);                  //หน่วงเวลา 600 ms
output_b(0x05);                 //ให้รถเข็นเดินหน้า
delay_ms(500);                  //หน่วงเวลา 500 ms
    }

while(temp=='D')                //ถ้า ตัวแปร temp = D
    {
output_b(0x09);                 //ให้รถเข็นเลี้ยวขวา
delay_ms(600);                  //หน่วงเวลา 600 ms
output_b(0x0A);                 //ให้รถเข็นถอยหลัง
delay_ms(500);                  //หน่วงเวลา 500 ms
    }

while(temp=='E')                //ถ้า ตัวแปร temp = E
    {
output_b(0x06);                 //ให้รถเข็นเลี้ยวซ้าย
    }

while(temp=='F')                //ถ้า ตัวแปร temp = F
    {
```

```
output_b(0x09);                 //ให้รถเข็นเลี้ยวขวา
    }

while(temp=='G')                //ถ้า ตัวแปร temp = G
    {
output_b(0x05);                 //ให้รถเข็นเดินหน้า
    }

while(temp=='H')                //ถ้า ตัวแปร temp = H
    {
output_b(0x0A);                 //ให้รถเข็นถอยหลัง
    }

while(temp=='I')                //ถ้า ตัวแปร temp = I
    {
output_b(0x00);                 //ให้รถเข็นหยุดนิ่ง
    }
  }
}
```

ประวัตินักวิจัย

ประวัติคณะผู้วิจัยและผู้ร่วมวิจัย

หัวหน้าโครงการวิจัย

1. **ชื่อ – นามสกุล** ดร.ไกรฤกษ์ เชยชื่น

2. **ตำแหน่งงานปัจจุบัน** อาจารย์/รองคณบดีฝ่ายวิจัยและบริการวิชาการ

3. **หน่วยงานและสถานที่อยู่ที่ติดต่อได้**

สาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ ศาลายา มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ โทร. 02-889-4585-7 ต่อ 2640

E-mail: kairoek2448@hotmail.com

4. **ประวัติการศึกษา**

ปริญญาเอก	มจธ.	สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์	2553
ปริญญาโท	มจธ.	สาขาวิศวกรรมไฟฟ้า	2544
ปริญญาตรี	มจธ.	สาขาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม	2541

5. **สาขาวิชาการที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)**

Computer programming

Digital image processing

Pattern recognition

6. **ประสบการณ์ที่เกี่ยวข้องกับการบริหารงานวิจัย**

ปีที่สำเร็จ	ชื่อเรื่อง	แหล่งทุน (ปี)
2555	ระบบควบคุมและโปรแกรมรูปแบบแสงไฟเวทีแบบไร้สาย	งบประมาณแผ่นดิน (วช.) ปี 2554
2555	วงจรขับสัญญาณแบบไม่เป็นเชิงเส้นสำหรับโซลินอยด์ซึ่งถูกประยุกต์ใช้ในการเชื่อมต่อทางกลและซอร์ฟแวร์สำหรับพิมพ์ไฟล์อักษรจากเครื่องคอมพิวเตอร์ ไปยังเครื่องพิมพ์เบรลล์ทางกล	งบประมาณ สกอ. ปี 2555

ประวัติผู้ร่วมวิจัย

1. ชื่อ – นามสกุล นายปกรณ์ ทู้ไพเราะ

2. ตำแหน่งงานปัจจุบัน อาจารย์สาขาวิชาวิศวกรรมคอมพิวเตอร์

3. หน่วยงานและสถานที่อยู่ที่ติดต่อได้

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ศาลายา มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ โทร. 02-889-4585-7 ต่อ 2680

E-mail: pakon.thuphairo@gmail.com

4. ประวัติการศึกษา

ปริญญาโท สาขาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

5. สาขาวิชาการที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา) ระบุสาขาวิชาการ-

Asynchronous digital logic design

6. ประสบการณ์ที่เกี่ยวข้องกับการบริหารงานวิจัย

International conference:

1. Pakon Thuphairo and Arthit Thongtak, “A design of Asynchronous Double Grain Reconfigurable Computing Array”, 13th ANSCSE, Bangkok, Thailand, 2008.