หุ่นยนต์ “อิเล็ก 1”
Robot “Elec I”

นรพนธ์ นิรันดร์วิโรจน์
มนัสวุฒิ กลิ่นอุทัย
รพี กิตติชยานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

หุ่นยนต์ “อิเล็ก 1”
Robot “Elec I”

นรพนธ์ นิรันดร์วิโรจน์
มนัสวุฒิ กลิ่นอุทัย
รพี กิตติชยานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

Robot "Elec I"

Norrapon Niranwirote
Manutsawot Klinutai
Raphii Kittichayanon

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN ELECTRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

ปริญญานิพนธ์ปีการศึกษา 2557
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ หุ่นยนต์ “อิเล็ก 1”

นักศึกษาผู้จัดทำ นายนรพนธ์ นิรันดร์วิโรจน์ **รหัสนักศึกษา** 54010669
นายมนัสวุฒิ กลิ่นอุทัย **รหัสนักศึกษา** 54011022
นายรพี กิตติชยานนท์ **รหัสนักศึกษา** 54011061

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

ปีการศึกษา 2557

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์	

หัวข้อปริญญานิพนธ์ หุ่นยนต์ “อิเล็ก 1”
นักศึกษาผู้จัดทำ นายนรพนธ์ นิรันดร์วิโรจน์ รหัสนักศึกษา 54010669
นายมนัสวุฒิ กลิ่นอุทัย รหัสนักศึกษา 54011022
นายรพี กิตติชยานนท์ รหัสนักศึกษา 54011061
อาจารย์ที่ปรึกษา รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์
ปีการศึกษา 2557

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ อธิบายการออกแบบและการควบคุมหุ่นยนต์เพื่อให้สามารถขนส่งสิ่งของต่างๆไป ยังที่หมายได้ถูกต้องตามต้องการ โครงงานชิ้นนี้ประกอบด้วย 2 ส่วนหลักในการทำงานคือ โปรแกรมตรวจจับเส้นทางและไมโครคอนโทรลเลอร์ ส่วนแรกการตรวจจับเส้นทางได้ทำการใช้วิธีประมวลผล ภาพระบบสี HSV และการตรวจจับระบบสี RGB ทำงานร่วมกันเพื่อให้หุ่นยนต์สามารถเคลื่อนที่ได้โดยใช้โปรแกรม MATLAB และในส่วนไมโครคอนโทรลเลอร์ได้ใช้ ARDUINO ควบคุมการทำงานหลักให้หุ่นยนต์เคลื่อนที่ไปถึงยังเป้าหมาย โดยการรับส่งข้อมูลกับคอมพิวเตอร์ผ่านโปรแกรม MATLAB แล้วทำการควบคุมมอเตอร์ให้หุ่นยนต์สามารถเคลื่อนที่ได้

Thesis Title	Robot Elec I	
Authors	Mr. Norrapon	Niranwirote
	Mr. Manutsawot	Klinutai
	Mr. Raphii	Kittichayanon
Thesis Advisor	Assoc. Prof. Dr. Surapan	Airphaiboon
Educational Year	2014	

Abstract

This thesis describes design and control of a mobile robot which delivers things to the target. In this project includes route detection program and microcontroller. The MATLAB is applied to detect routes based on HSV technique and colors detection technique to provide the movement of robot. ARDUINO is chosen as the main controller which communicates to MATLAB and controls the motors for movement.

กิตติกรรมประกาศ

โครงงานหุ่นยนต์ “อิเล็ก 1” นี้ประกอบด้วยชิ้นงาน ปริญญานิพนธ์ และการนำเสนอ ซึ่งโครงงานนี้สามารถดำเนินการอย่างราบรื่นไปได้ด้วยดี โดยได้รับคำแนะนำจากอาจารย์ที่ปรึกษา คือ อาจารย์ สุรพันธุ์ เอื้อไพบูลย์ และอาจารย์ประจำภาควิชาอิเล็กทรอนิกส์ท่านอื่นๆ รวมถึงบุคลากรทุกๆท่านในภาควิชาอิเล็กทรอนิกส์ที่ได้ให้ความช่วยเหลือในการอำนวยความสะดวกด้านต่างๆ ทำให้สามารถดำ เนินงานและแก้ปัญหาต่างๆเพื่อทำให้เกิดชิ้นงานนี้ขึ้นมาได้ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อ I

Abstract II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและลักษณะของโครงงาน 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตโครงงาน 1

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 โครงสร้างของรายงาน 2

บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง 3

2.1 IC (L298) 3

2.2 ARDUINO (Mega 2560) 5

2.3 Ultrasonic Sensor Module (HC-SR04) 6

2.4 Keypad 8

2.5 MATLAB 8

2.6 มาตรฐานสี 9

2.6.1 ระบบสี RGB 10

2.6.2 ระบบสี HSV (Hue Saturation Value) 10

บทที่ 3 หลักการออกแบบและการดำเนินงาน 12

3.1 หลักการออกแบบ 12

3.1.1 การประมวลผลภาพโดยโปรแกรม MATLAB 12

3.1.2 เงื่อนไขการทำงานของบอร์ด ADUINO(MEGA) 12

3.1.3 การออกแบบวงจรขับมอเตอร์เพื่อควบคุมทิศทางและความเร็ว 12

3.1.4 การออกแบบการควบคุม 14

3.2 การดำเนินงาน 16

3.2.1 ศึกษาการใช้งานโปรแกรม MATLAB ในส่วนของการตรวจวิเคราะห์ภาพระบบสี RGB และเทคนิคระบบสี HSV 21

3.2.1.1 ประมวลผลภาพด้วยเทคนิคระบบสี HSV 21

3.2.1.2 ตรวจจับวัตถุน้ำเงิน 21

สารบัญ (ต่อ)

หน้า

3.2.1.3 ตรวจจับวัตถุสีแดง....21
3.2.2 ศึกษาวิธีการใช้งานบอร์ด ARDUINO ในการควบคุมการเคลื่อนที่....22
3.2.3 ออกแบบวงจรขับมอเตอร์กระแสตรง....22
3.2.4 ประกอบรวมทุกส่วนเข้าด้วยกันและทดสอบการเคลื่อนที่....22

บทที่ 4 การทดลองและผลการทดลอง....24
4.1 กล่าวนำ....24
4.2 ขั้นตอนการทดลอง....24
4.2.1 ทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV....24
4.2.2 ทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน....25
4.2.3 ทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่....27
4.2.4 สร้างคำสั่งรับข้อมูลจากอุลตร้าโซนิกเซ็นเซอร์....28
4.2.5 วัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์....29
4.2.6 การทำงานของหุ่นยนต์....29
4.2.7 ทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่....31
4.3 ผลการทดลอง....31
4.3.1 ผลการทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV....31
4.3.2 ผลการทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน....35
4.3.3 ผลการทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่....40
4.3.5 ผลการทดลองวัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์....41
4.3.6 ผลการทดลองการทำงานของหุ่นยนต์....43
4.3.7 ผลการทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่....44

บทที่ 5 สรุปผลการดำเนินงาน....45
5.1 สรุปผลการทดลอง....45
5.1.1 ทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV....45
5.1.2 ทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน....45
5.1.3 ทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่....45
5.1.4 สร้างคำสั่งรับข้อมูลจากอุลตร้าโซนิกเซ็นเซอร์....45
5.1.5 วัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์....45
5.1.6 การทำงานของหุ่นยนต์....46
5.1.7 ทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่....46
5.2 สรุปผลโครงงาน....46
5.3 วิจารณ์ผลการดำเนินงาน....47

สารบัญ (ต่อ)

หน้า

บรรณานุกรม **48**

ภาคผนวก ก **49**

ก.1 แบบวงจรขับมอเตอร์กระแสตรง 50

ก.2 ลายปริ้นท์วงจรขับมอเตอร์กระแสตรง 50

ภาคผนวก ข **51**

ข.1 โค้ดคำสั่งการตรวจจับเส้นสีดำและวัตถุสีน้ำเงินโดยโปรแกรม MATLAB 52

ข.2 โค้ดคำสั่งการควบคุมมอเตอร์ด้วย ADUINO (MEGA) 62

ภาคผนวก ค **70**

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางแสดงรายละเอียดคุณสมบัติของบอร์ด ARDUINO Mega 2560 6
2.2 ตารางแสดงการต่อเพื่อใช้งาน Ultrasonic Sensor Module(HC-SR04).....7
2.3 ตารางแสดงข้อมูลรายละเอียดคุณสมบัติของ Ultrasonic Sensor Module(HC-SR04).....7
3.1 การกำหนดค่าลอจิกต่างๆของL298.....14
4.1 ขอบเขตการวิเคราะห์.....27
4.2 ค่าลอจิกของการเคลื่อนที่รูปแบบต่างๆ.....29
4.3 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 1.....40
4.4 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 2.....40
4.5 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 3.....41
4.6 ผลความคลาดเคลื่อน.....44

สารบัญรูป

รูปที่ หน้า

1.1 โมเดลการทำงานของหุ่นยนต์....2
2.1 ไอซีเบอร์ L298....3
2.2 Circuit Diagram ของ L298....4
2.3 บอร์ด ARDUINO MEGA 2560....6
2.4 Ultrasonic Sensor Module (HC-SR04)....6
2.5 Membrane 4x4 Matrix Keypad....8
2.6 การรวมของแสงสีในระบบ RGB....10
2.7 โมเดลระบบสี RGB....10
2.8 ระบบสี HSV....11
3.1 วงจรขับมอเตอร์....13
3.2 Block diagram แสดงการควบคุมหุ่นยนต์....15
3.3 Flow Chart แสดงการทำงาน....16
3.4 Flow Chart แสดงการทำงาน (ต่อ)....17
3.5 Flow Chart แสดงการทำงาน (ต่อ)....18
3.6 Flow Chart แสดงการทำงาน (ต่อ)....19
3.7 Flow Chart แสดงการทำงาน (ต่อ)....20
3.8 Flow Chart แสดงการทำงานของหุ่นยนต์....23
4.1 ภาพ test1.jpg....24
4.2 ภาพ test2.jpg....25
4.3 ภาพจาก S-Channel ของส่วนหนึ่งบนเส้นทางเมื่อแปลงเป็นภาพไบนารี่....27
4.4 ภาพแสดงขอบเขตที่ใช้วิเคราะห์....28
4.5 Flow Chart แสดงการทำงานของหุ่นยนต์....29
4.6 Flow Chart แสดงการทำงานของหุ่นยนต์ (ต่อ)....30
4.7 ภาพต้นแบบที่ 1....31
4.8 ภาพต้นแบบที่ 2....31
4.9 ภาพจาก H-Channel ต้นแบบที่ 1....32
4.10 ภาพจาก H-Channel ต้นแบบที่ 2....32
4.11 ภาพจาก S-Channel ต้นแบบที่ 1....33
4.12 ภาพจาก S-Channel ต้นแบบที่ 2....33
4.13 ภาพจาก V-Channel ต้นแบบที่ 1....34
4.14 ภาพจาก V-Channel ต้นแบบที่ 2....34
4.15 ภาพต้นแบบที่ 3....35
4.16 ภาพองค์ประกอบสีแดง....35

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.17 ภาพเทาของสีแดง....36
4.18 ภาพผลต่างระหว่างภาพองค์ประกอบสีแดงกับภาพเทาของสีแดง....36
4.19 ภาพไบนารี่ของผลต่าง(สีแดง)....37
4.20 ภาพองค์ประกอบสีน้ำเงิน....37
4.21 ภาพเทาของสีน้ำเงิน....38
4.22 ภาพผลต่างระหว่างภาพองค์ประกอบสีน้ำเงินกับภาพเทาของสีน้ำเงิน....38
4.23 ภาพไบนารี่ของผลต่าง(สีน้ำเงิน)....39
4.24 ภาพหลังการประมวลผลสีต่างๆ....39
4.25 สัญญาณ PWM เมื่อเดินหน้า....41
4.26 สัญญาณ PWM เมื่อเลี้ยวขวา....42
4.27 สัญญาณ PWM เมื่อเลี้ยวซ้าย....42
4.28 หุ่นยนต์ “อิเล็ก 1”....43
4.29 ภาพเส้นทาง....43
4.30 ภาพเส้นทาง (ต่อ)....44
ก.1 วงจรขับมอเตอร์กระแสตรง....50
ก.2 ลายปริ้นท์วงจรขับมอเตอร์กระแสตรง....50

บทที่ 1
บทนำ

1.1 ความเป็นมาและลักษณะของโครงงาน

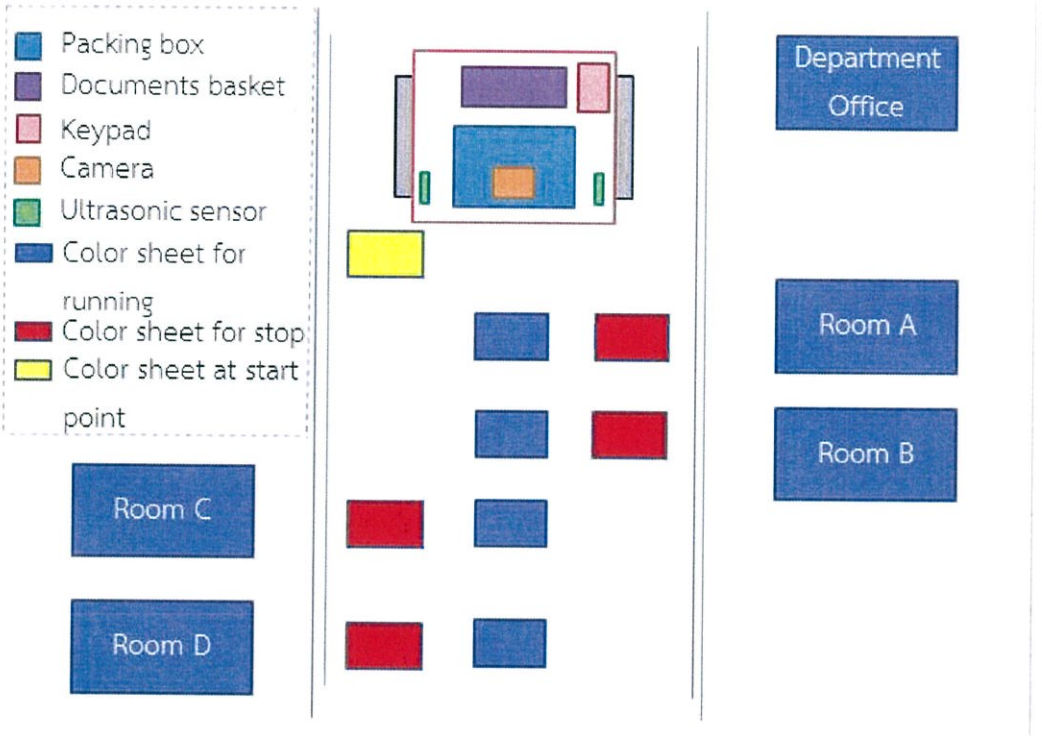
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ถือเป็นสถาบันการศึกษาทางด้านเทคโนโลยีในระดับแนวหน้าของประเทศไทย และเมื่อกล่าวถึงคำว่าเทคโนโลยีแล้วคงหนีไม่พ้นความเข้าใจที่ว่า “เทคโนโลยีก้าวหน้าด้วยอิเล็กทรอนิกส์” ดังนั้น สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ ถือว่าเป็นหน่วยงานทางการศึกษา ที่ผลิตวิศวกรเข้าสู่เส้นทางแห่งเทคโนโลยีโดยตรง ซึ่งวิศวกรที่มีความรู้ความสามารถจะมีความสำคัญต่อทั้งภายในองค์กรและประเทศชาติรวมถึงเป็นผู้มีบทบาทสำคัญต่อการพัฒนาเทคโนโลยีอีกด้วย ฉะนั้นแล้วการที่นักศึกษาจะก้าวไปสู่การเป็นวิศวกรที่มีความสามารถได้ ต้องมีการลงมือปฏิบัติเชิงสร้างสรรค์ ประยุกต์และพัฒนา ดังนั้นจึงเกิดโครงงานหุ่นยนต์“ อิเล็ก 1 ”ขึ้นมาโดยเบื้องต้นได้มีเป้าหมายในการนำไปใช้ประโยชน์ คือการรับส่งสิ่งของภายในตัวอาคารของภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ โดยหุ่นยนต์ “อิเล็ก 1” นี้มีระบบการทำงานแบบอัตโนมัติ

1.2 วัตถุประสงค์

1. เพื่อใช้หุ่นยนต์นี้ในการขนส่งสิ่งของน้ำหนักไม่เกิน 5 กิโลกรัมโดยอัตโนมัติ
2. เพื่อพัฒนาทักษะในการใช้งานการประมวลผลภาพกับการเคลื่อนที่
3. เพื่อสร้างหุ่นยนต์นี้ขึ้นเป็นต้นแบบแก่นักศึกษาสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

1.3 ขอบเขตโครงงาน

สามารถเคลื่อนที่ในเส้นทางตรงและหากมีการเลี้ยวเบน ตัวหุ่นยนต์สามารถกลับสู่เส้นทางตรงได้โดยอัตโนมัติ และสามารถหยุดการเคลื่อนที่ได้เมื่อพบสัญลักษณ์ที่ได้กำหนดเป็นเงื่อนไข



รูปที่ 1.1 โมเดลการทำงานของหุ่นยนต์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับเพิ่มพูนทักษะทางด้านการประมวลผลภาพ
2. ได้ประสบการณ์ทางการวิเคราะห์ปัญหาในขั้นตอนการสร้างการเคลื่อนที่ของหุ่นยนต์
3. ได้พัฒนากระบวนการการวางแผน จัดแจงและประสานงานของสมาชิก
4. ได้ปฏิบัติงานและรับทราบรายละเอียดจริงตลอดทุกขั้นตอนการทำงาน

1.5 โครงสร้างของรายงาน

รายงานฉบับนี้เป็นการรายงานผลจากการศึกษาและทดลองตลอดหนึ่งปีการศึกษา ในหัวข้อโครงงานเรื่อง หุ่นยนต์ “ อิเล็ก 1” ซึ่งได้นำวิธีการประมวลภาพมาใช้เป็นหัวใจของการเคลื่อนที่ของหุ่นยนต์ โดยมีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน ในส่วนของเนื้อหาจะประกอบด้วยส่วนต่างๆ แยกเป็นบท ๆดังนี้

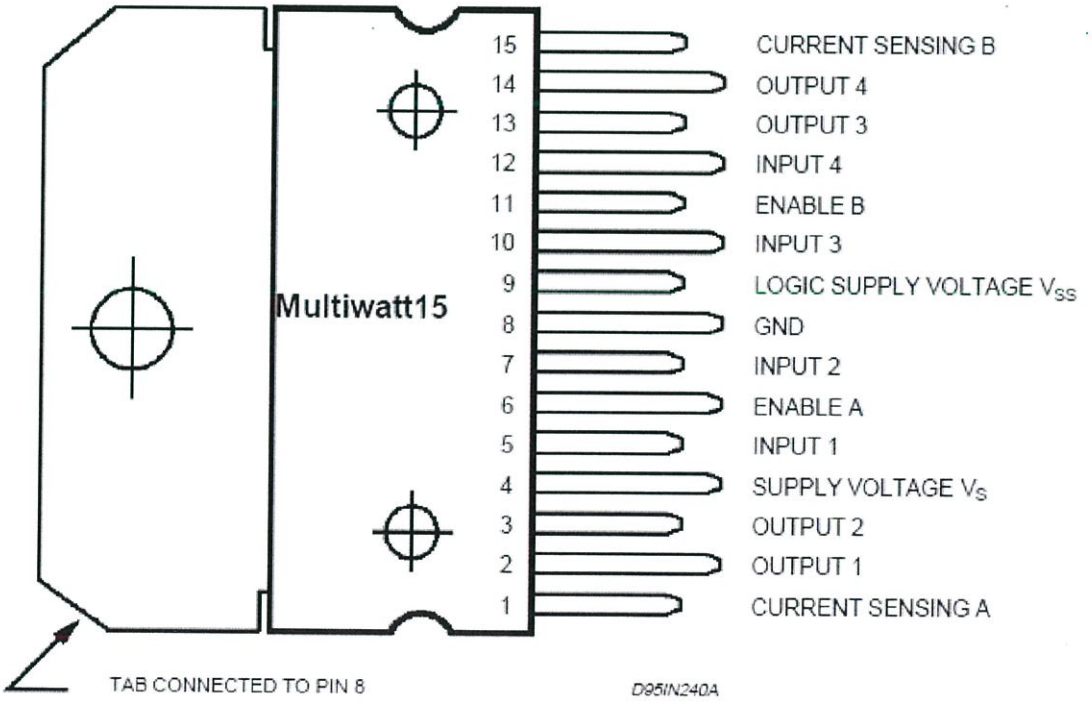
บทที่ 1 บทนำ
บทที่ 2 องค์ประกอบและหลักการทำงาน
บทที่ 3 หลักการออกแบบ
บทที่ 4 ขั้นตอนและวิธีดำเนินการ
บทที่ 5 การทดลองและผลการทดลอง
บทที่ 6 สรุปผลการดำเนินงาน

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 IC (L298)

IC ย่อมาจากคำว่า Integrated Circuit แปลว่าวงจรรวม ไอซีเป็นสิ่งประดิษฐ์ที่รวมชิ้นส่วนทางอิเล็กทรอนิกส์ต่างๆ ที่ประกอบเป็นวงจรหรือส่วนของวงจรที่มีขนาดเล็กไว้ในตัวเดียว ซึ่งชิ้นส่วนต่างๆ เหล่านั้นคือ ทรานซิสเตอร์ ตัวต้านทาน ไดโอด เป็นต้น ดังนั้นไอซีจึงมีหลายขา ไอซีมีหลายชนิดล้วนแต่มีหน้าที่การทำงานและจะมีหมายเลขประจำตัว เช่น หมายเลข L298, หมายเลข 555 เป็นต้น วิธีดูตำแหน่งขาของไอซี ให้สังเกตจุดบนขอบตัวไอซี ขาข้างที่อยู่ใกล้จุดเรียกว่า ขาที่1 แล้วให้นับเรียงต่อกันไปตามลำดับ ซึ่งในโครงงานชิ้นนี้ได้ มีการนำ IC หมายเลข L298 มาใช้ในการทำงานด้วย

วงจรรวม เบอร์ L298 เป็นไอซีที่นิยมใช้กันมากในการควบคุมมอเตอร์กระแสตรงเพื่อสร้างวงจรขับมอเตอร์โดยสามารถที่ใช้ขับมอเตอร์กระแสตรง ได้พร้อมกัน 2 ชุดพร้อมกับรับกระสูงสุดได้ 2A ต่อมอเตอร์ 1 ตัวหรือสามารถต่อแบบ Bridge ให้สามารถทนกระแสได้ถึง 4A



รูปที่ 2.1 ไอซีเบอร์ L298

หน้าที่ของขาต่างๆ ของไอซีเบอร์ L298

(1) ต่อกราวด์

(2),(3) ต่อเข้าขั้วมอเตอร์

(4) รับแรงดันสำหรับมอเตอร์โดยคำนึงถึงคุณลักษณะมอเตอร์ (Motor Specification) เป็นหลัก

(5),(7) เป็นขาสำหรับรับลอจิก 1(3.3 Volt+) , ลอจิก 0(0Volt) เพื่อควบคุมมอเตอร์ A (มอเตอร์ 1)

(6) Enable A คล้ายกับการเปิด-ปิดมอเตอร์ ถ้าจ่ายแต่ลอจิก เข้า(5),(7) แต่ไม่จ่ายEnable A มอเตอร์ก็จะไม่ทำงาน

(8) กราวด์ ต้องเป็น กราวด์ ร่วม (1),(15) ต้องต่อร่วมกันด้วย

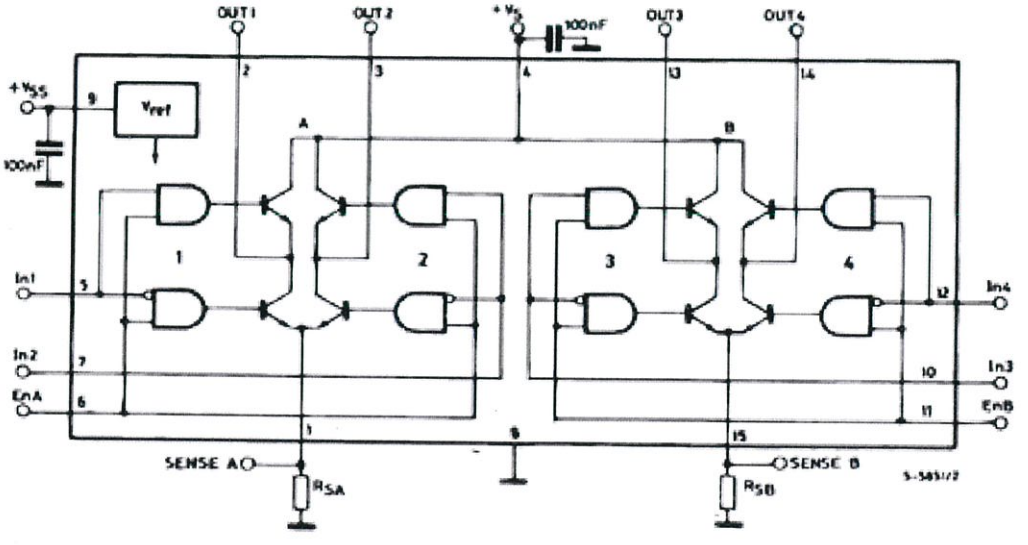
(9) เป็นแรงดันสำหรับ Logic

(10),(12) เป็นขาสำหรับรับ ลอจิก 1(3.3 Volt+),Logic 0(0Volt) เพื่อควบคุมมอเตอร์ B (มอเตอร์ 2)

(11) Enable B คล้ายๆการเปิด-ปิดมอเตอร์ถ้าจ่ายแต่ Logic เข้า(10),(12)แต่ไม่จ่ายEnable B มอเตอร์ก็จะไม่ทำงาน

(13),(14) ต่อเข้าขั้วมอเตอร์

(15) ต่อกราวด์



รูปที่ 2.2 Circuit Diagram ของ L298

2.2 ARDUINO (Mega 2560)

ARDUINO คือเครื่องมือหนึ่งที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกแล้วส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้เครื่อง PC ตั้งโต๊ะโดยตัวบอร์ดนั้นออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยวและมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน

ARDUINO สามารถประยุกต์ทำเครื่องใช้อัจฉริยะ รับสัญญาณจากสวิทช์ หรือ เซนเซอร์, และควบคุม หลอดไฟ, มอเตอร์, หรืออุปกรณ์อื่นๆ โปรเจค ARDUINO เป็นได้ทั้งแบบทำงานอิสระหรือจะทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่อง PC ตัวบอร์ดสามารถประกอบขึ้นใช้เองหรือจะซื้อสำเร็จที่มีขาย ส่วนโปรแกรมพัฒนา ARDUINO สามารถดาวน์โหลดได้ฟรี

บอร์ด ARDUINO เป็นบอร์ดไมโครคอนโทรลเลอร์ ในตระกูล AVR ที่กำลังได้รับความนิยมอย่างสูงทั่วโลกเพราะเป็น Open Source ท่านสามารถดัดแปลง ไปใช้งานได้ทั้ง ฮาร์ดแวร์ และซอฟแวร์ ได้ทันที ภาษาที่ใช้กับบอร์ดนี้จะเป็นลักษณะของ C/C+ โดยจัดให้มี Libraries ต่างๆให้สามารถเรียก ใช้งานได้ทันที ครอบคลุมการติดต่อกับ I/O Port ต่างๆ ได้กว้างมากการใช้งานก็ง่ายเพียงแค่เสียบสาย USB และติดตั้งโปรแกรมก็พร้อมที่จะประยุกต์ใช้ไมโครคอนโทรลเลอร์ได้ทันที

ในตลาดไมโครคอนโทรลเลอร์มีตัวเลือกมากมาย เช่น Parallax Basic Stamp, Netmedia's BX-24, Pidgets, MIT's Handyboard, และอีกหลายเจ้าที่มีคุณสมบัติใกล้เคียงกัน เพื่อที่ให้ใช้งานได้ง่ายและเน้นการเขียนโปรแกรมไมโครคอนโทรลเลอร์เป็นหลัก ARDUINO ก็เช่นเดียวกันแต่มีข้อแตกต่างที่เห็นได้ชัดคือ

- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- เปิดเผยซอร์สโค้ดทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- เปิดเผยวงจร และ นำไปพัฒนาขยาย ฮาร์ดแวร์ ได้
- ราคาไม่แพง

ตารางที่ 2.1 ตารางแสดงรายละเอียดคุณสมบัติของบอร์ด ARDUINO Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output , 4 UART TTL)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

รูปที่ 2.3 บอร์ด ARDUINO MEGA 2560

2.3 Ultrasonic Sensor Module (HC-SR04)

รูปที่ 2.4 Ultrasonic Sensor Module (HC-SR04)

โมดูลอุลตร้าโซนิกนี้เป็นอุปกรณ์ใช้วัดระยะทางโดยไม่ต้องมีการสัมผัสกับตำแหน่งที่ต้องการวัด สามารถวัดได้ตั้งแต่ 2 cm ถึง 400 cm โดยส่งสัญญาณอุลตร้าโซนิกความถี่ 40 kHz ไปที่วัตถุที่ต้องการวัดและรับสัญญาณที่สะท้อนกลับมาพร้อมทั้งจับเวลาเพื่อนำมาใช้ในการคำนวณระยะทาง

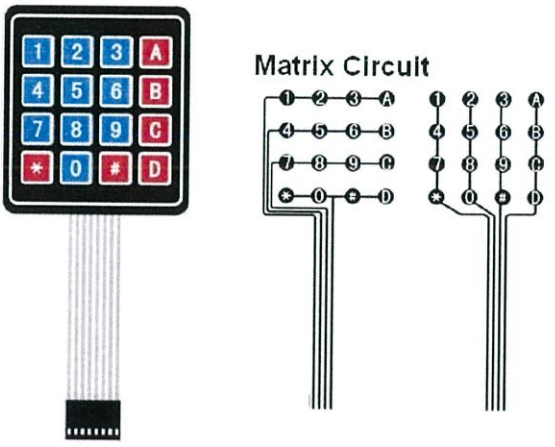
ตารางที่ 2.2 ตารางแสดงการต่อเพื่อใช้งาน Ultrasonic Sensor Module(HC-SR04)

Pin	การต่อเพื่อใช้งาน
1	5V Supply
2	Trigger Pulse Input
3	Echo Pulse Output
4	GND

ตารางที่ 2.3 ตารางแสดงข้อมูลรายละเอียดคุณสมบัติของ Ultrasonic Sensor Module(HC-SR04)

Working voltage	DC 5 V
Static current	3 mA
Working temperature	0 ~ + 70
Output way	GPIO
Induction Angle	Less than 15
Detection range	2 cm to 4 m
Detecting precision	0.3 cm + 1%
Sensor size	Approx. 45 x 20 x 1.6mm
Hole diameter	Approx. 1mm

2.4 Keypad



รูปที่ 2.5 Membrane 4x4 Matrix Keypad

เป็น SWITCH KEYPAD แบบ MEMBRANE SWITCH ขนาน 4x4 (16 KEY) ขั้วต่อใช้งานแบบ 8 PIN 2.54 mm 4x4 Keypad membrane แบบ 4 x 4 ชนิดนี้จะประกอบไปด้วยปุ่ม 16 ปุ่มที่เรียงต่อกันเป็นเมตริกซ์แบบ 4 แถว และ 4 คอลัมน์ ถ้าใครเคยได้สัมผัสแล้วก็คงพอจะเดาออกว่าปุ่มแต่ละปุ่ม เป็นการกดเพื่อให้หน้าสัมผัส ไปแตะกันทำให้เป็นการเชื่อมต่อกันทางไฟฟ้าไปยังอีกด้านหนึ่งของสวิตช์หลักการในการตรวจสอบว่าผู้ใช้กำลังกดปุ่มอะไรอยู่นั้นก็ใช้วิธีการ สแกน ไปทีละคอลัมน์ จนครบทุก คอลัมน์ แล้วนำมาตีความว่ามีการตอบสนองออกมาเป็นแบบใดบ้าง เช่น ถ้ามีการกดเลข 1 อยู่ ในขณะที่เราจ่ายแรงดัน 5 โวลต์ไปที่ คอลัมน์ ที่ 1 จะมีเพียง แถวแรกเท่านั้นที่จะอ่านค่าแรงดันได้ High นอกนั้นจะเป็น Low หรือถ้ามีการกดปุ่ม # อยู่ ขณะที่ สแกน ไปแต่ละคอลัมน์ นั้นจะไม่เจอแรงดัน High ที แถวใดเลย จนกว่าจะ สแกน ไปถึง คอลัมน์ ที่ 3 ซึ่งจะพบว่ามีการตอบสนองกลับมาจาก แถวที่ 4 นั้นเอง ดังนั้นเมื่อพบว่าเป็นการ สแกน คอลัมน์ ที่ 3 และมีแถว4 ตอบสนอง ก็คือปุ่ม '#' นั่นเอง

2.5 MATLAB

MATLAB คือภาษาที่มีประสิทธิภาพสูงใช้คำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์ยุคก่อนๆ เช่นภาษา C, C++ โดยภายในตัว MATLAB ประกอบด้วยภาษาคอมพิวเตอร์ทูลบ็อกซ์(กลุ่มฟังก์ชันเฉพาะสำเร็จรูป) และฟังก์ชันพื้นฐานจำนวนมากเป็นผลทำให้การวิเคราะห์ทำได้หลากหลายวิธีพร้อมกับได้รับคำตอบที่รวดเร็วกว่า

ซึ่งตัวโปรแกรมนี้ได้รวมเอาความสามารถในการคำนวณการดูและติดตามข้อมูลต่างๆรวมทั้งการเขียนโปรแกรมไว้ในรูปแบบที่ง่ายต่อการใช้งานโดยที่ปัญหาและวิธีการหาคำตอบจะแสดงโดยใช้สัญลักษณ์ทางคณิตศาสตร์ที่ใช้ตามปกติการใช้งานทั่วๆไป จะมีดังนี้

- คำนวณทางคณิตศาสตร์
- พัฒนาอัลกอริธึม
- สร้างแบบจำลอง, จำลองการทำงานและสร้างตัวต้นแบบ
- วิเคราะห์ข้อมูล, ตรวจค้นข้อมูลและแสดงผลข้อมูลในรูปแบบต่างๆ

- แสดงภาพกราฟฟิคต่างๆทางวิทยาศาสตร์และวิศวกรรมศาสตร์
- พัฒนาโปรแกรมใช้งานต่างๆรวมถึงการสร้าง Graphic User Interface (GUI) ด้วย

MATLAB สามารถนำไปประยุกต์ใช้งานได้หลายสาขามากทั้งการประมวลผลสัญญาณการสื่อสาร การประมวลผลภาพและวิดีโอ ระบบควมคุมการวัดและควบคุม การคำนวณทางเศรษฐศาสตร์ รวมถึงการคำนวณทางชีววิทยาและอื่นๆ

ข้อดีของ MATLAB

1. โปรแกรมใช้งานง่าย ทำงานได้หลากหลายรูปแบบด้วยสภาพแวดล้อมเชิงโต้ตอบคล้ายเครื่องคิดเลขสามารถตรวจสอบค่าต่างๆซ้ำได้อย่ารวดเร็ว ประยุกต์ใช้ในการออกแบบและแก้ปัญหาทางคณิตศาสตร์ได้ง่ายและมีประสิทธิภาพสูงมาก

2. มีทูลบ็อกซ์หรือไลบรารีฟังก์ชันทางคณิตศาสตร์สำหรับพีชคณิตให้เลือกใช้มากมาย สถิติ การวิเคราะห์ ฟูเรีย ฟัซซีลอจิก การประมวลผลภาพ และวิดีโอ การคำนวณทางเศรษฐศาสตร์และชีววิทยาเป็นต้น

3. มี Simulink เป็นซอฟต์แวร์สนับสนุนการสร้างแบบจำลองทางคณิตศาสตร์บนMATLAB พร้อมด้วยบล็อกเซ็ตหลายสาขาวิชาเพื่อใช้ในการวิเคราะห์ระบบต่างๆและการสร้างแอพ พลิเคชั่นหรือสร้างอุปกรณ์สำเร็จรูป

4. จัดหาเครื่องมือสำหรับสร้างแอพพลิเคชั่นบน MATLAB ด้วย GUI (Graphic User Interface) นั่นคือสามารถสร้างแอพพลิเคชั่นบน MATLAB ด้วยเครื่องมือคล้ายๆ กับ Visual Basic

5. MATLAB สามารถเชื่อมการทำงานร่วมกับภาษาซี จาวา ดอทเน็ต เอ็กเซลหรือฮาร์ดแวร์ภายนอก

ข้อเสียของ MATLAB

1. เครื่องคอมพิวเตอร์ที่ใช้ต้องมีประสิทธิภาพสูง

2. โปรแกรมมีส่วนประกอบเยอะซึ่งมีทั้งส่วนของทูลบ็อกซ์และบล็อกเซ็ตจำนวนมาก ดังนั้นฟังก์ชันพร้อมกับ MATLAB จึงเยอะมาก บางครั้งจึงทำให้การหาฟังก์ชันที่ต้องการทำได้อย่างลำบาก

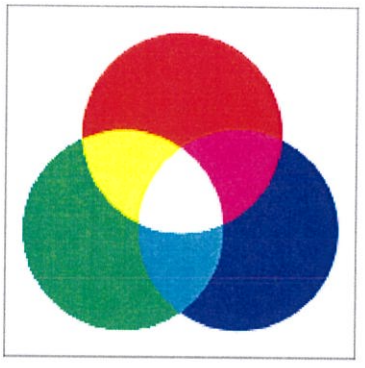
3. ราคาแพง

2.6 มาตรฐานสี

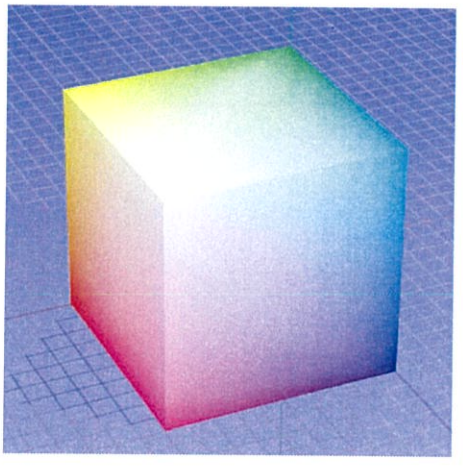
มาตรฐานสีนั้นมีอยู่หลายระบบ เช่น RGB(Red, Green และ Blue), HSV(Hue, Saturation และ Value), CMYK(Cyan, Magenta, Yellow และ Black), CIE(International Commission on Illumination) และ HLS(Hue, Lightness และ Saturation) เป็นต้น โดยในโครงงานชิ้นนี้จะขอกล่าวถึงเฉพาะระบบสี RGB และ HSV

2.6.1 ระบบสี RGB

คือระบบแสงสี มีพื้นฐานจากหลักการของการมองเห็นแสงสีและการผสมของแสงสีในอัตราส่วนต่างๆเป็นระบบสีที่นิยมใช้กันทั่วไป โดยอาศัยหลักการการรวมแม่สีของแสงเข้าด้วยกันได้แก่ แสงสีแดง แสงสีเขียว และแสงสีน้ำเงิน มีค่าตั้งแต่ 0 ถึง 255เมื่อนำมาฉายรวม กันจะทำให้เกิดสีใหม่อีก 3 สี คือสีมาเจนต้า(Magenta) สีฟ้าไซแอน(Cyan) และสีเหลืองเมื่อฉายแสงสีทั้งหมดรวมกันจะได้สีขาวดังรูปที่ 2.6 และโมเดลระบบสี RGB สามารถแสดงได้ดังรูปที่ 2.7 ซึ่งแกน X แทนแสงสีแดง แกน Y แทนแสงสีน้ำเงิน และแกน Z แทนแสงสีเขียว



รูปที่ 2.6 การรวมของแสงสีในระบบ RGB



รูปที่ 2.7 โมเดลระบบสี RGB

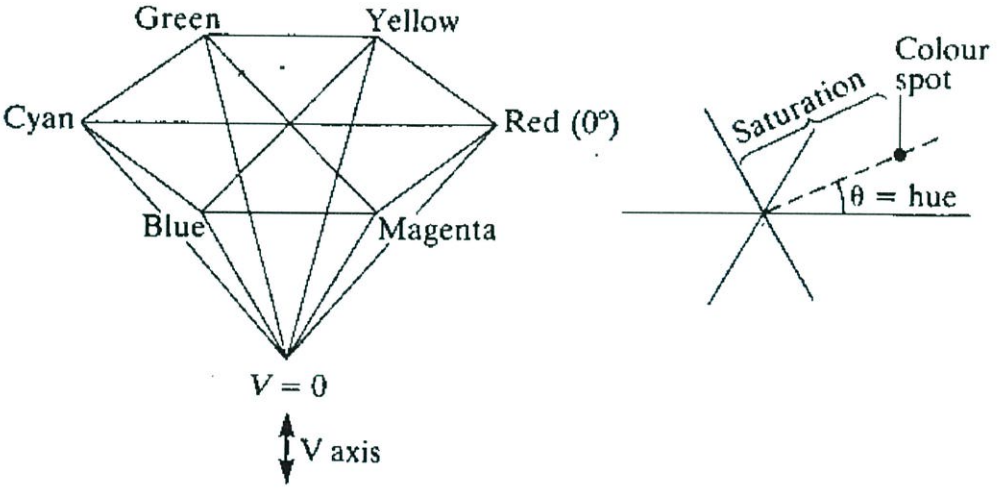
2.6.2 ระบบสี HSV (Hue Saturation Value)

เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก(แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 255 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดงเท่ากับ 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_k = red - \min(red, green, blue)$$

$$green_k = green - \min(red, green, blue)$$

$$blue_k = blue - \min(red, green, blue)$$



รูปที่ 2.8 ระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่าง เช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\frac{(240 x blue_k) + (120 x green_k)}{blue_k + green_k}$$

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังนี้

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)}$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

$$value = \max(red, green, blue)$$

บทที่ 3
หลักการออกแบบและการดำเนินงาน

3.1 หลักการออกแบบ

3.1.1 การประมวลผลภาพโดยโปรแกรม MATLAB

3.1.1.1 ใช้วัตถุสีพื้นฐานต่างๆในการประมวลผลแล้วนำมาใช้ประโยค ซึ่งสีพื้นฐานนั้นได้แก่สีแดง สีเขียว สีน้ำเงิน และสีเหลือง เป็นต้น

3.1.1.2 นำการประมวลผลภาพด้วยเทคนิคระบบสี HSV มาใช้ประโยชน์

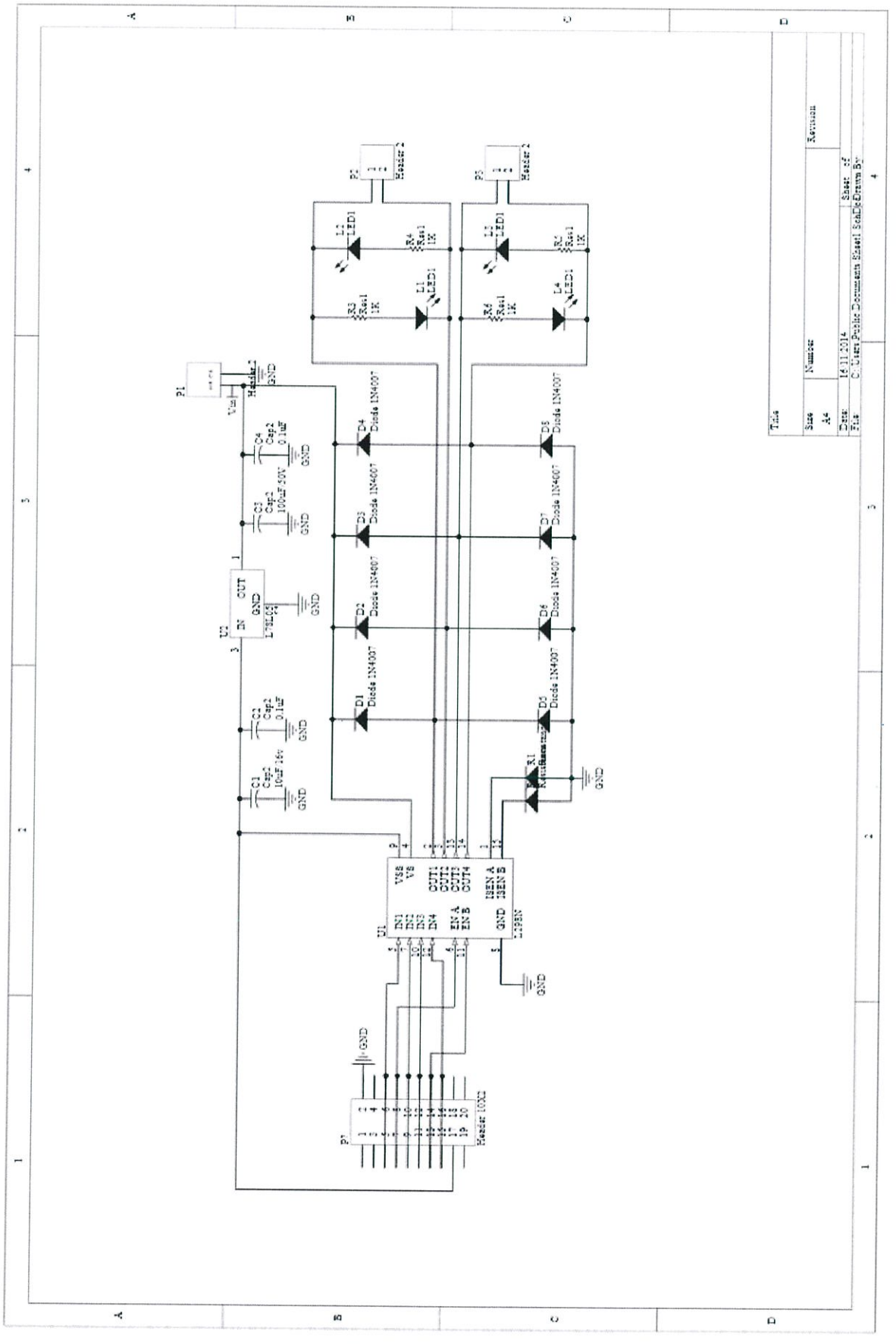
3.1.2 เงื่อนไขการทำงานของบอร์ด ADUINO(MEGA)

3.1.2.1 ทำหน้าที่ควบคุมวงจรวงจรขับมอเตอร์และสื่อสารกับโปรแกรมMATLAB จากคอมพิวเตอร์ อุลตร้าโซนิกเซ็นเซอร์ และแป้นพิมพ์

3.1.2.2 ใช้ข้อมูลที่รับได้จากโปรแกรม MATLAB และอุลตร้าโซนิกเซ็นเซอร์ มาเป็นเงื่อนไขในการหมุนของมอเตอร์

3.1.3 การออกแบบวงจรขับมอเตอร์เพื่อควบคุมทิศทางและความเร็ว

การออกแบบวงจรขับมอเตอร์ ได้เลือกใช้ วงจรรวม L298 ให้ทำหน้าที่เป็นตัวควบคุมมอเตอร์กระแสตรง เนื่องจากเหตุผลที่ว่าสามารถขับมอเตอร์ที่แรงดันได้ถึง 50 V และสามารถขับมอเตอร์ได้ 2 ตัวซึ่งในโครงงานนี้ ใช้มอเตอร์ 12 V 2 ตัว สามารถควบคุมความเร็วแบบพัลล์สปีดได้ ซึ่งทำให้บังคับหุ่นยนต์ให้เลี้ยวซ้ายหรือขวาได้ จากการศึกษาข้อมูลใน ดาต้าชีท ของ IC L298 ของบริษัท ST และค้นคว้าหาวงจรในเว็บไซต์ต่างๆของวงจรขับมอเตอร์ที่ใช้ วงจรรวม L298 และได้ยึดแบบวงจรจากแหล่ง ข้อมูลต่างๆในการออกแบบ



รูปที่ 3.1 วงจรขับมอเตอร์

หลักการทำงานของวงจร

เริ่มจากการรับค่าสถานะจากโปรแกรม MATLAB เข้าสู่ ARDUINO โดยขาที่ใช้นั้นขึ้นอยู่กับการเขียนโปรแกรมลงใน ARDUINO จากนั้น L298 ก็จะควบคุมและขับมอเตอร์แต่ละตัวตามคำสั่งจากอินพุทที่ได้รับเข้ามา เช่น สั่งให้เดินหน้า มอเตอร์ทั้งสองตัวก็จะหมุนไปข้างหน้าพร้อมกัน ซึ่งสามารถสรุปการทำงานของวงจรขับมอเตอร์ได้ดังตารางต่อไปนี้

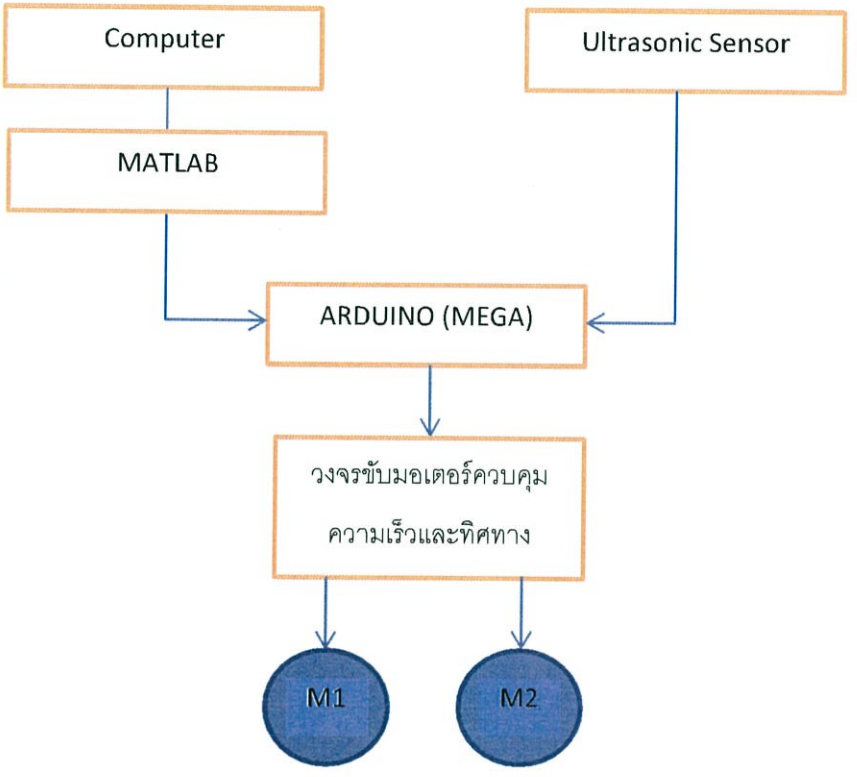
ตารางที่ 3.1 การกำหนดค่าลอจิกต่างๆของL298

Input form		IN1	IN2	IN3	IN4	Function
MATLAB	sensor					
F		0	200	0	165	Forward
R	ระยะด้านซ้ายน้อยกว่า 50 cm	0	255	0	0	Turn Right
L	ระยะด้านขวาน้อยกว่า 50 cm	0	0	0	255	Turn Left
0		0	0	0	0	Stop
		255	0	0	255	Turn Around

หมายเหตุ ค่าอินพุทจาก MATLAB จะเป็นตัวแปรแบบอักษร

3.1.4 การออกแบบการควบคุม

การออกแบบการควบคุม จะกล่าวถึงส่วนที่เป็นการรับส่งข้อมูลและการติดต่อระหว่างคอมพิวเตอร์ เซ็นเซอร์ กับหุ่นยนต์ เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ซึ่งการควบคุมจะประกอบด้วยส่วนต่างๆ ซึ่งเขียนเป็น Block diagram ได้ดังนี้

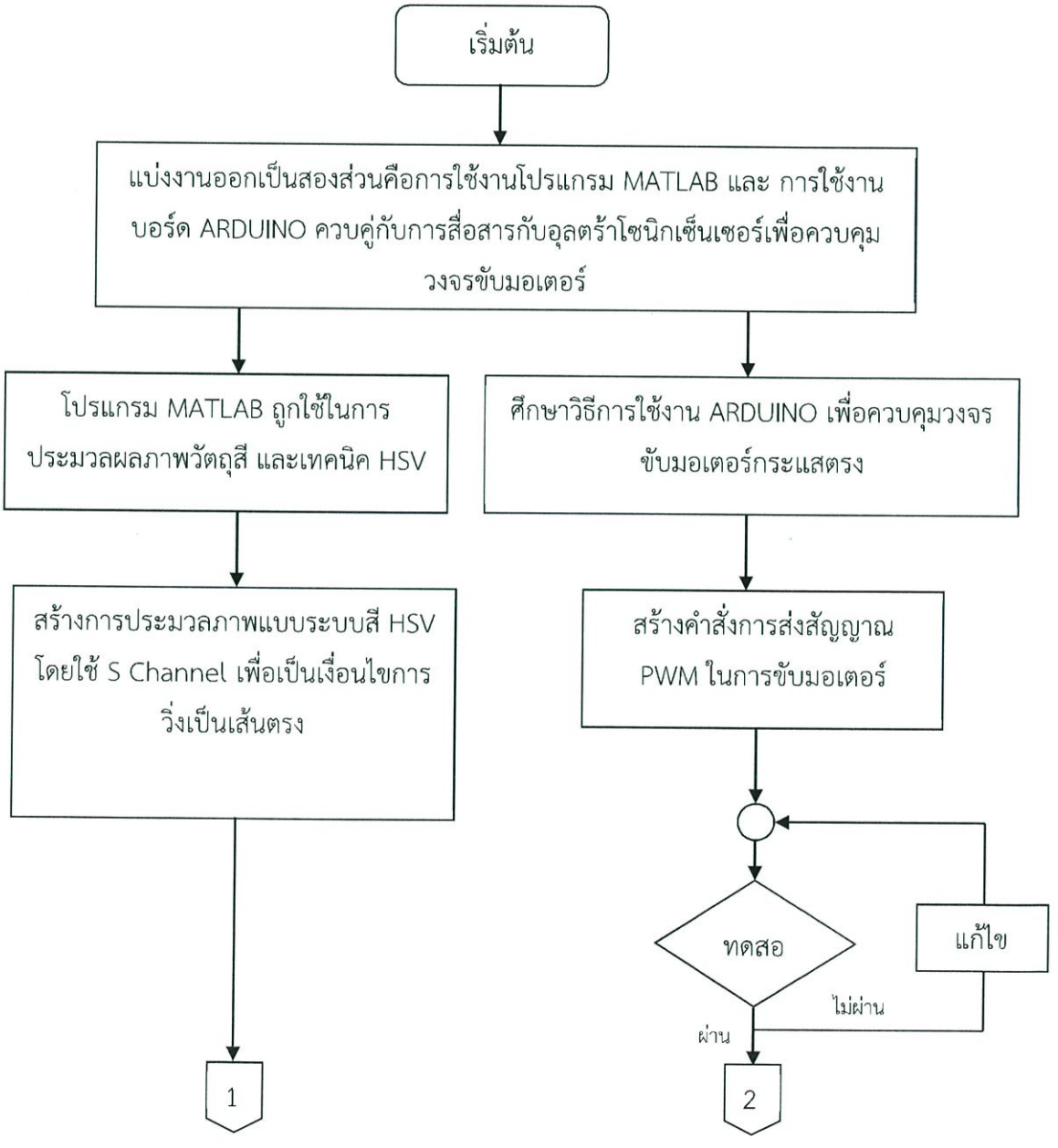


รูปที่ 3.2 Block diagram แสดงการควบคุมหุ่นยนต์

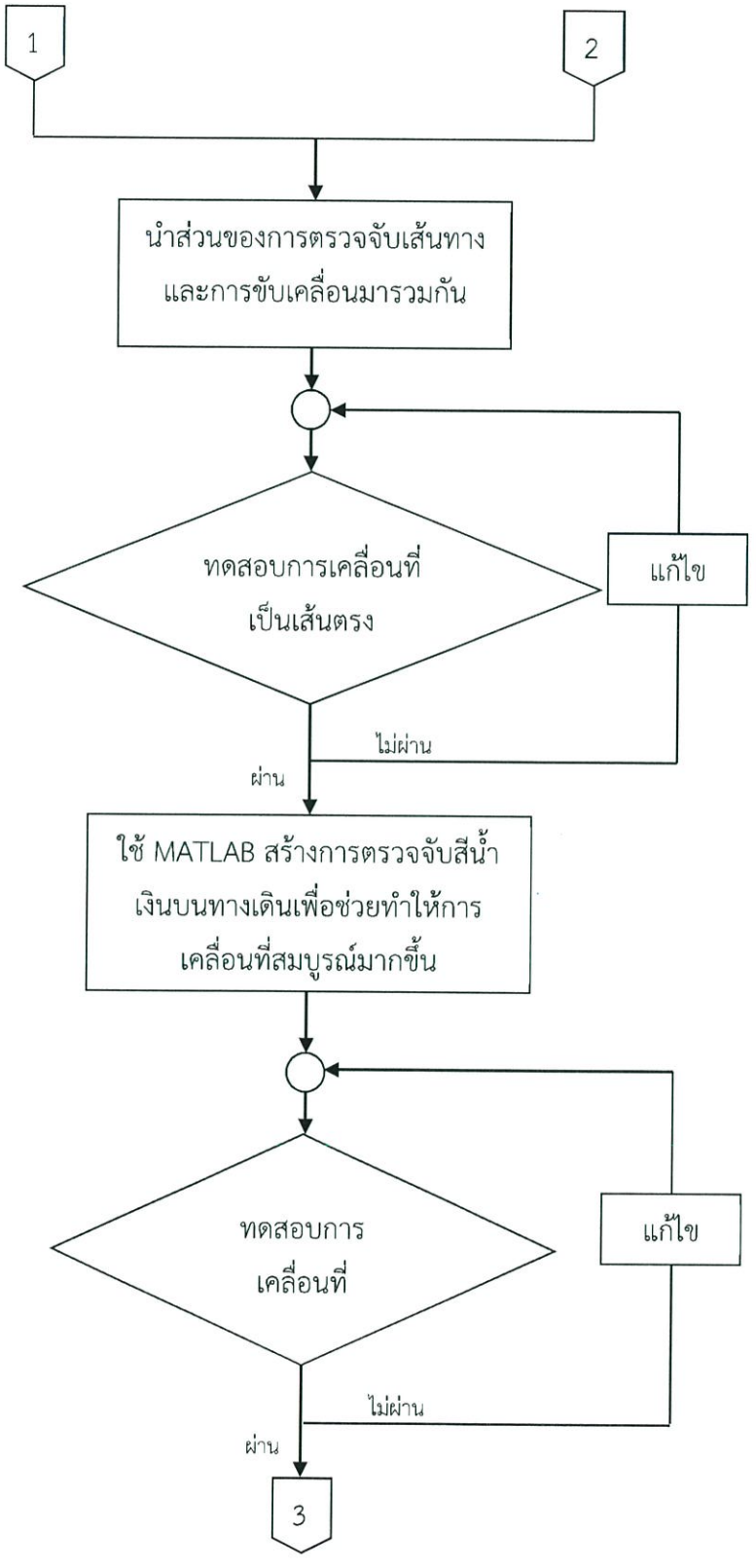
การควบคุมการเคลื่อนที่จะใช้ ARDUINO ในการสั่งการโดยการสั่งการจะรับค่าตัวแปรอินพุทซึ่งได้มาจากประมวลผลภาพจากโปรแกรม MATLAB และอุลตร้าโซนิกเซ็นเซอร์ เพื่อกำหนดค่าลอจิกให้ขา IN1, IN2, IN3 และIN4 ให้แก่วงจรขับมอเตอร์เพื่อควบคุมความเร็วของมอเตอร์แต่ละตัว ซึ่งทำให้เกิดทิศทางในการเคลื่อนที่

3.2 การดำเนินงาน

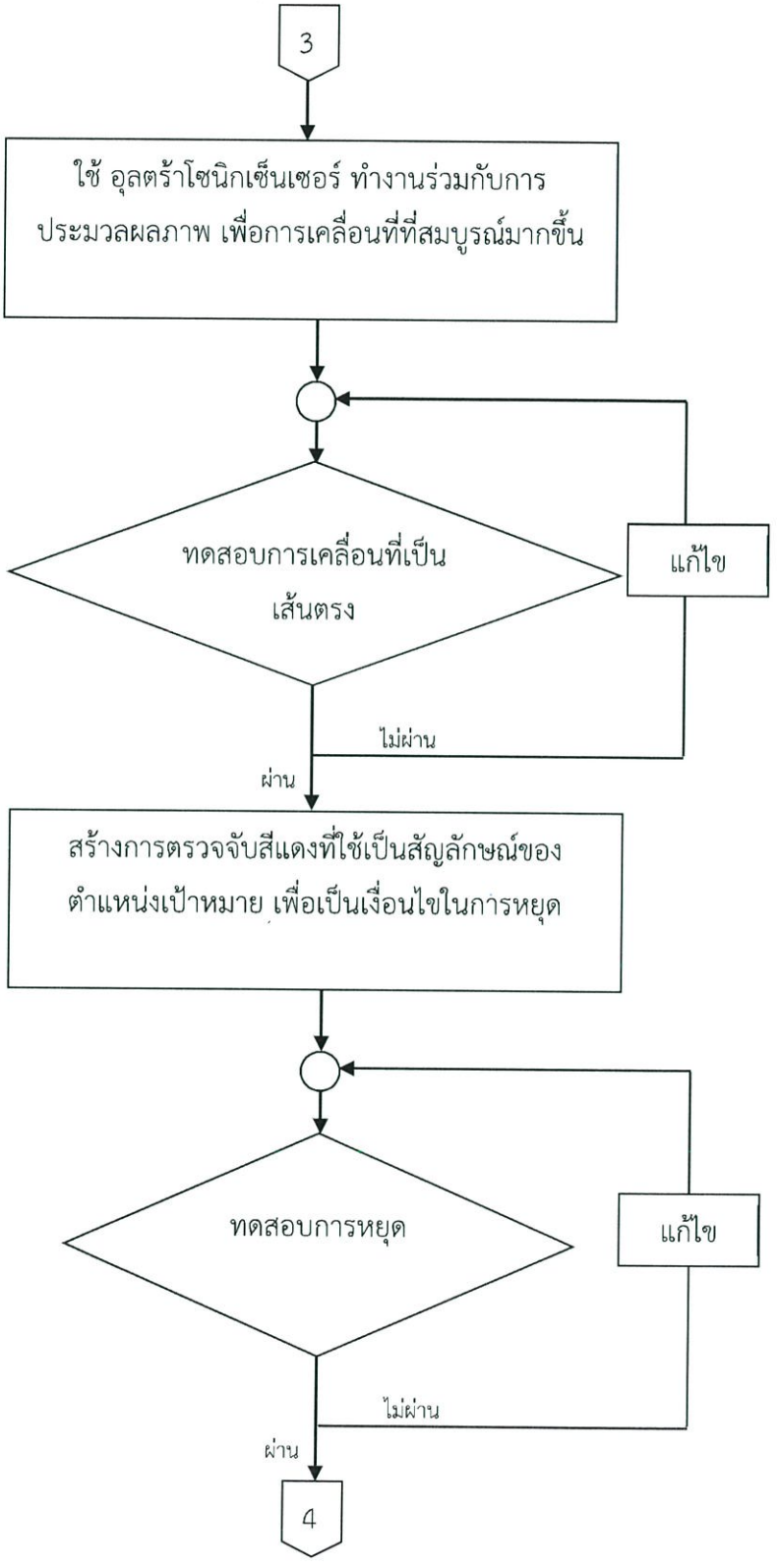
ในการวางแผนการดำเนินงานนั้น จำเป็นที่จะต้องมีการวางแผนการทำงานเพื่อตรวจสอบความเหมาะสมของเวลาในแต่ละช่วงการดำเนินการ การจัดการเวลาที่เหมาะสมจะส่งผลต่องานที่ได้ออกมามีความสมบูรณ์มากที่สุด ในการดำเนินงานของโครงงานนี้มีโฟลว์ชาร์ตดังต่อไปนี้



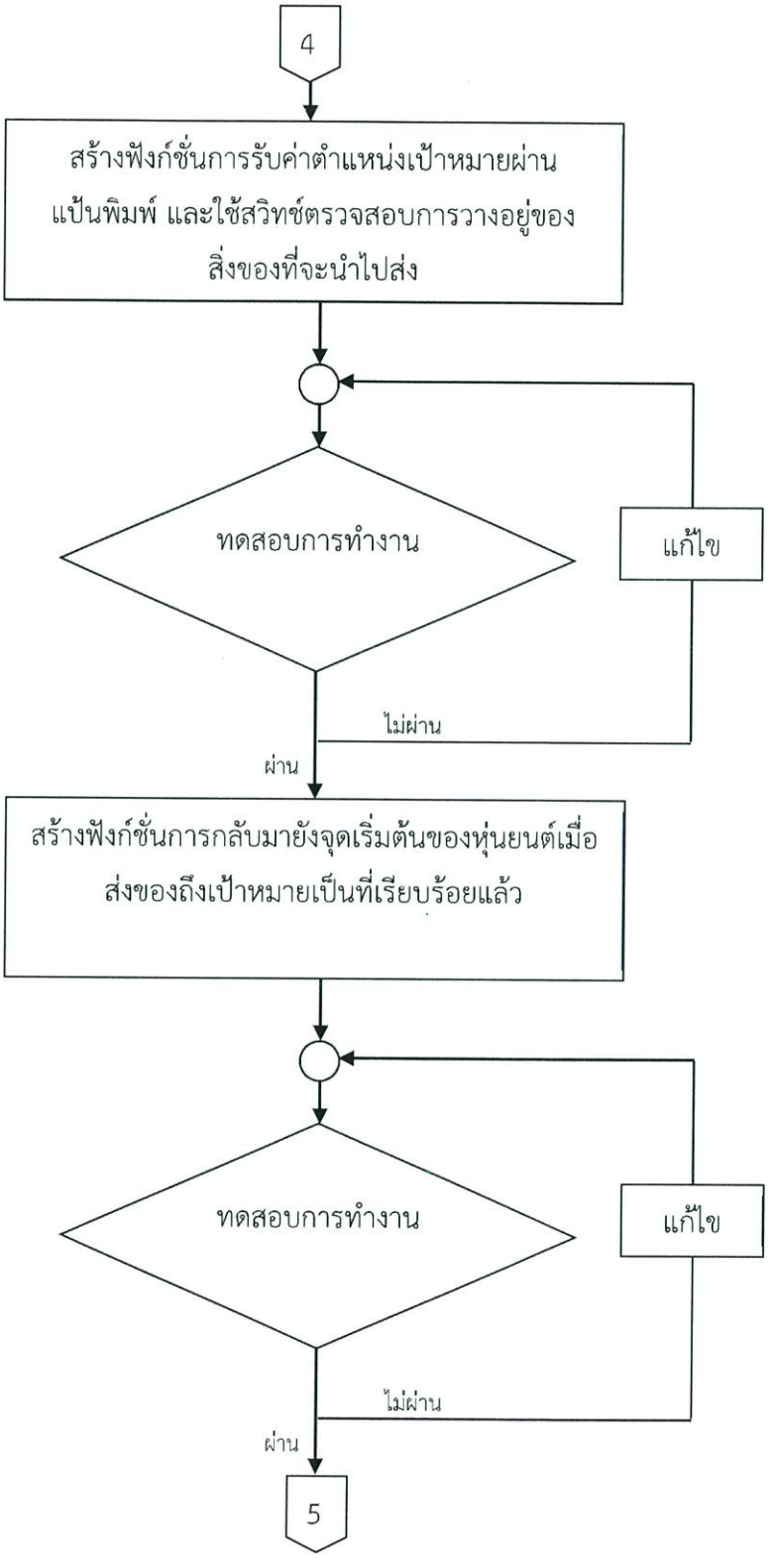
รูปที่ 3.3 Flow Chart แสดงการทำงาน



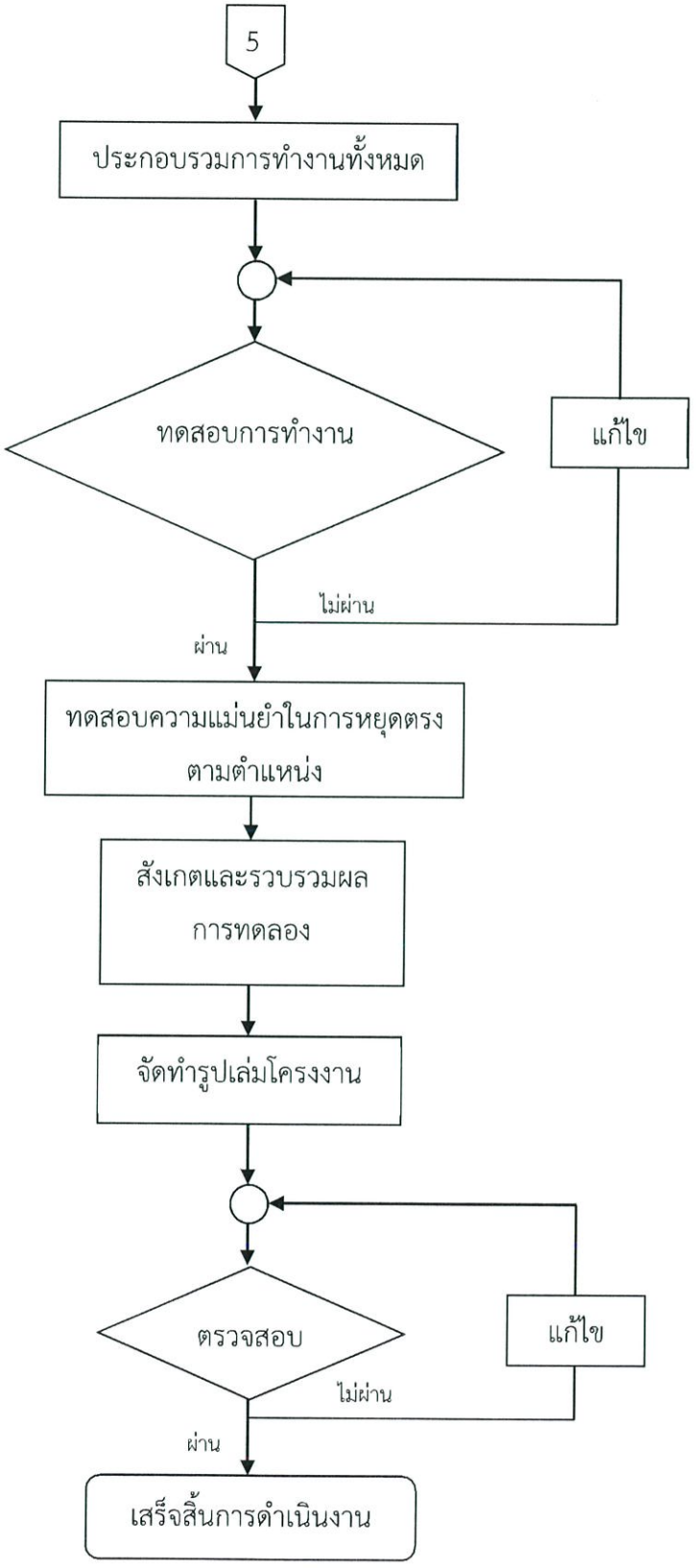
รูปที่ 3.4 Flow Chart แสดงการทำงาน (ต่อ)



รูปที่ 3.5 Flow Chart แสดงการทำงาน (ต่อ)



รูปที่ 3.6 Flow Chart แสดงการทำงาน (ต่อ)



รูปที่ 3.7 Flow Chart แสดงการทำงาน (ต่อ)

ขั้นตอนการดำเนินงาน สามารถแยกเป็นขั้นตอนต่างๆ ได้ดังนี้

3.2.1 ศึกษาการใช้งานโปรแกรม MATLAB ในส่วนของการตรวจวิเคราะห์ภาพระบบสี RGB และเทคนิคระบบสี HSV
3.2.2 ศึกษาวิธีการใช้งานบอร์ด ARDUINO (MEGA) ในการควบคุมการเคลื่อนที่
3.2.3 ออกแบบวงจรขับมอเตอร์กระแสตรง
3.2.4 ประกอบรวมทุกส่วนเข้าด้วยกันและทดสอบการเคลื่อนที่
3.2.5 จัดทำรูปเล่มโครงงาน

3.2.1 ศึกษาการใช้งานโปรแกรม MATLAB ในส่วนของการตรวจวิเคราะห์ภาพระบบสี RGB และเทคนิคระบบสี HSV

3.2.1.1 ประมวลผลภาพด้วยเทคนิคระบบสี HSV

ในการประมวลภาพด้วยเทคนิค HSV ได้จำกัดขอบเขตการใช้เทคนิคเฉพาะส่วนผนังสองข้างของทางเดิน ซึ่งเทคนิคนี้ถูกใช้ในส่วนของ S-Channel คือวิเคราะห์การอิ่มตัวขององค์ประกอบสีของภาพ (S:Saturate) เมื่อระบุได้ว่าผนังอยู่ตำแหน่งใด จึงสามารถนำไปใช้เป็นเงื่อนไขในการสั่งงานให้หุ่นยนต์เคลื่อนที่ตรงไปข้างหน้าได้

3.2.1.2 ตรวจจับวัตถุน้ำเงิน

วัตถุสีน้ำเงินจะถูกวางบนพื้นทางเดินบริเวณกลางเลนเป็นระยะๆ โดยการประมวลผลภาพสีน้ำเงินนี้ จะมีการสร้างพิกัดเพื่อบอกตำแหน่งของสีน้ำเงินแล้วแสดงผลออกทางจอภาพ โดยสีน้ำเงินนี้นำมาใช้เพื่อปรับปรุงการเคลื่อนที่ให้หุ่นยนต์เคลื่อนที่ตรงขึ้น ถือเป็นส่วนช่วยและทำงานร่วมกันกับเทคนิค HSV ซึ่งการที่จะให้หุ่นยนต์เคลื่อนที่ตรงนั้น ได้นั้นต้องใช้พิกัดของสีน้ำเงินมาเป็นเงื่อนไข โดยสร้างขอบเขตที่ว่าสีน้ำเงินควรอยู่บริเวณกลางจอภาพ อาศัยค่าพิกัดแกน X ที่ต้องอยู่ในช่วง (310,330) หากพิกัดแกน X มากกว่า 330 ให้ส่งค่าสถานะให้เลี้ยวขวาแก่ ARDUINO ในทางกลับกันหากค่าพิกัดแกน X น้อยกว่า 310 ให้ส่งค่าสถานะให้เลี้ยวซ้ายแก่ ARDUINO

3.2.1.3 ตรวจจับวัตถุสีแดง

ในการทำให้หุ่นยนต์หยุดเคลื่อนที่นั้นได้ใช้เงื่อนไขที่ว่า ถ้าตรวจพบวัตถุสีแดงซึ่งวางอยู่ที่ตำแหน่งเป้าหมาย ให้โปรแกรมส่งสถานะหยุดแก่ ARDUINO จะถือว่าได้เคลื่อนที่มาถึงยังเป้าหมายแล้ว เป้าหมายมีมากกว่าหนึ่งตำแหน่ง ดังนั้นการระบุว่าเป็นตำแหน่งไหนจึงใช้การนับการพบวัตถุสีแดงที่หุ่นยนต์เคลื่อนที่ผ่าน

3.2.2 ศึกษาวิธีการใช้งานบอร์ด ARDUINO ในการควบคุมการเคลื่อนที่

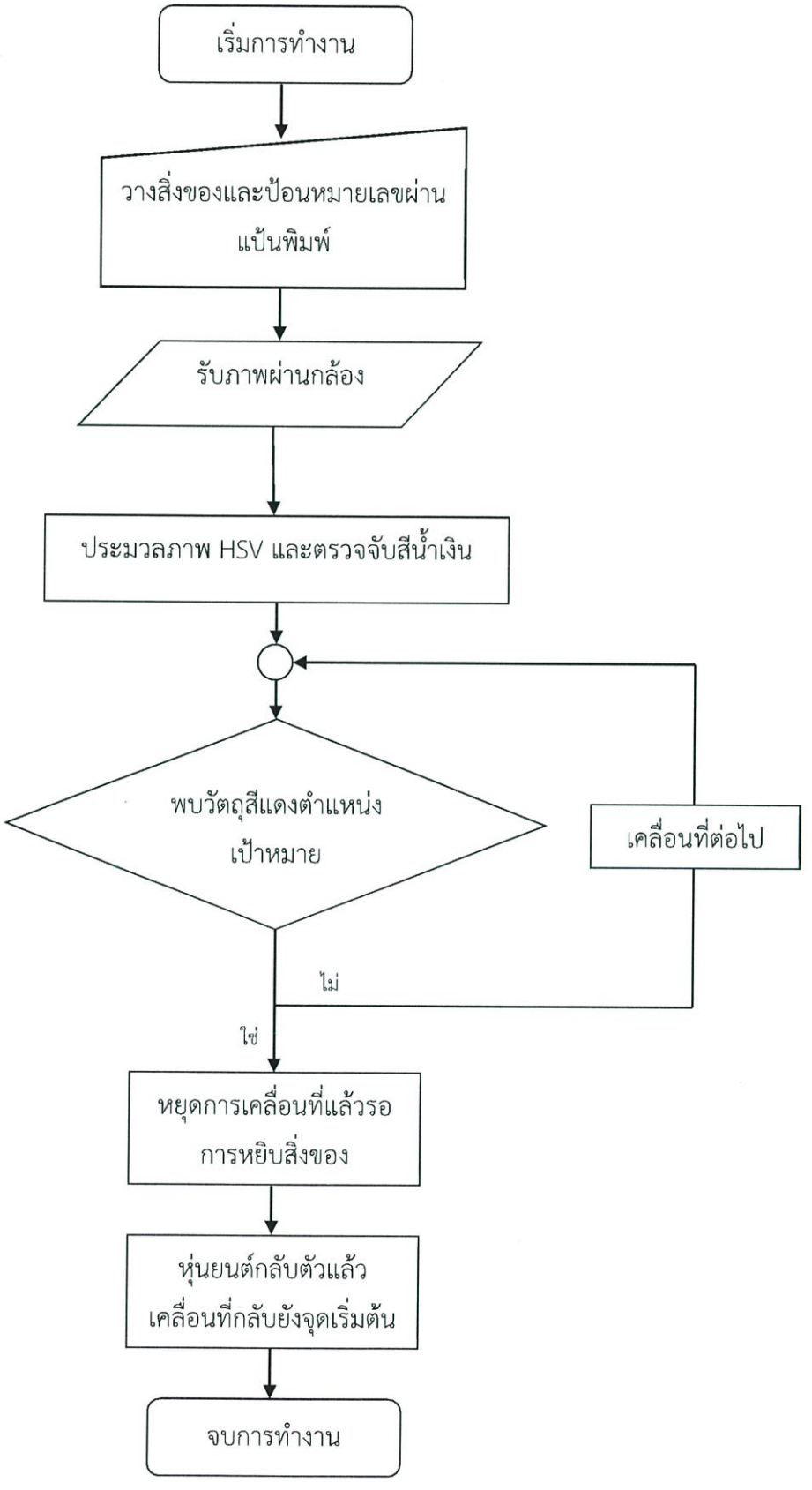
- ศึกษาการใช้งาน Serial Port และการสร้างสัญญาณ PWM
- สร้างคำสั่งเดินหน้า, เลี้ยวซ้าย, เลี้ยวขวาและหยุด
- สร้างคำสั่งการทำงานร่วมกับ อุลตร้าโซนิกเซ็นเซอร์ เพื่อรับค่าระยะห่างระหว่างวัตถุด้านข้างกับตัวหุ่นยนต์
- สร้างคำสั่งการรับค่าหมายเลขเป้าหมายจากแป้นพิมพ์
- สร้างฟังก์ชั่นการกลับไปยังจุดเริ่มต้นเมื่อส่งของเสร็จ

3.2.3 ออกแบบวงจรขับมอเตอร์กระแสตรง

วงจรขับมอเตอร์กระแสตรงนี้ ออกแบบมาเพื่อขับมอเตอร์ได้ 2 ตัว โดยมีหัวใจหลักเป็นไอซีเบอร์ L298 ซึ่งเป็นไอซีที่ใช้สำหรับการขับมอเตอร์ โดยวงจรนี้มีแหล่งจ่ายไฟเป็นแบตเตอรี่ 12V 5.5AH

3.2.4 ประกอบรวมทุกส่วนเข้าด้วยกันและทดสอบการเคลื่อนที่

เมื่อผ่านขั้นตอนต่างๆ และได้ผลการทดสอบในแต่ละขั้นตอนเป็นที่น่าพอใจแล้วจากนั้นจึงนำเอาแต่ละส่วนมาประกอบรวมการทำงานเข้าด้วยกัน คือวางสิ่งของที่จะขนส่งบนที่วางจากนั้นป้อนหมายเลขตำแหน่งที่เป็นเป้าหมาย โดยการเคลื่อนที่จะใช้การรับภาพจากกล้องแล้วให้โปรแกรม MATLAB ประมวลผลภาพด้วยเทคนิค HSV และตรวจจับสีน้ำเงินโดยมีเงื่อนไขที่ว่า หากพิกัดแกน X ของสีน้ำเงินยังอยู่ในช่วง (310,330) และค่าพิกัดแกน Y มากกว่า 240 จะส่งค่าสถานะไปยังบอร์ด ARDUINO ให้ควบคุมวงจรขับมอเตอร์ในการวิ่งเป็นเส้นตรง และหากพิกัดแกน X อยู่นอกขอบเขตพิกัดที่กำหนด ก็จะมีการส่งค่าสถานะให้เลี้ยวซ้ายหรือขวาไป เพื่อให้การเคลื่อนที่กลับมาเป็นแนวตรงอีกครั้ง จากนั้นหากมีการตรวจพบวัตถุสีแดงหุ่นยนต์ก็จะหยุดเคลื่อนที่ หลังจากเคลื่อนที่ถึงเป้าหมายและมีการหยิบสิ่งของเรียบร้อยแล้ว หุ่นยนต์ก็จะกลับตัวและเคลื่อนที่กลับมายังจุดเริ่มต้นได้เองเพื่อพร้อมที่จะรับเป้าหมายใหม่



รูปที่ 3.8 Flow Chart แสดงการทำงานของหุ่นยนต์

บทที่ 4
การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในการทำโครงงานชิ้นนี้ ส่วนที่ต้องมีการทดลองเพื่อหาความเหมาะสมคือการประมวลผลภาพด้วยเทคนิค HSV ซึ่งแบ่งออกเป็น 3 Channel คือ H-Channel S-Channel และ V-Channel ซึ่งการในไปใช้ประโยชน์จำเป็นต้องศึกษาความแตกต่างของแต่ละ channel จึงต้องมีการทดลองหาภาพที่ได้จากแต่ละ channel ออกมา จากนั้นจะกล่าวถึงขั้นตอนการประมวลภาพสีต่างๆ ซึ่งจะถูกนำมาใช้ร่วมกันเพื่อสร้างการทำงานให้แก่หุ่นยนต์เนื่องจากคุณสมบัติทางกลและกายภาพของมอเตอร์ที่ขับเคลื่อนล้อทั้งสองข้างนั้นไม่เหมือนกันโดยสมบูรณ์ จึงทำให้ความเร็วของล้อสองข้างนั้นไม่เท่ากัน ดังนั้นการเคลื่อนที่ของหุ่นยนต์จึงจำเป็นต้องอาศัยการประมวลผลภาพที่จะสร้างเงื่อนไขต่างๆมาเป็นตัวอ้างอิงเพื่อทำให้ตัวหุ่นยนต์สามารถรักษาการเคลื่อนที่เป็นแนวตรงได้ โดยไม่มีการชนกับวัตถุด้านข้างนอกจากนี้ ในสภาวะแวดล้อมจริงบนเส้นทางเดินทำให้ต้องมีการใช้อุลตร้าโซนิกเซ็นเซอร์เข้ามาช่วยปรับปรุงการเคลื่อนที่รวมถึงหุ่นยนต์นี้มีการรับค่าเลขตำแหน่งผ่านแป้นพิมพ์ จึงได้มีการสร้างฟังก์ชั่นการอ่านค่าขึ้นมาด้วย

4.2 ขั้นตอนการทดลอง

4.2.1 ทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV

1. บนหน้าต่าง Command Window ของโปรแกรม MATLAB พิมพ์คำสั่งการอ่านภาพ

```
TestHSV = imread('test1.jpg');
```

รูปที่ 4.1 ภาพ test1.jpg

2. เปลี่ยนภาพ TestHSV ไปเป็นภาพในแต่ละ channel โดยใช้คำสั่งต่อไปนี้ตามลำดับ

[H,~,~] = rgb2hsv(TestHSV);

[~,S,~] = rgb2hsv(TestHSV);

[~,~,V] = rgb2hsv(TestHSV);

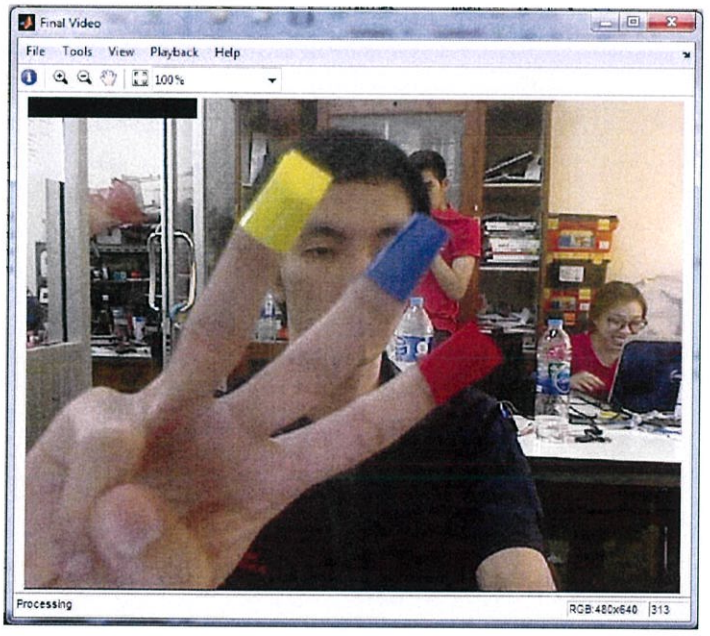
แล้วบันทึกภาพที่ได้

3. ทำการทดลองซ้ำโดยเปลี่ยนภาพต้นแบบเป็นตำแหน่งอื่นๆบนเส้นทาง

4.2.2 ทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน

1. บนหน้าต่าง Command Window ของโปรแกรม MATLAB พิมพ์คำสั่งการอ่านภาพ

TestColors = imread('test2.jpg');



รูปที่ 4.2 ภาพ test2.jpg

2. แยกองค์ประกอบภาพสีแดงและสีน้ำเงิน โดยใช้คำสั่งดังต่อไปนี้

Red = TestColors (: , : , 1);

Blue = TestColors (: , : , 3);

3. เปลี่ยนภาพที่ได้เป็นภาพเทาโดยใช้คำสั่งดังต่อไปนี้

RedGray = rgb2gray(Red);

BlueGray = rgb2gray(Blue);

4. หาผลต่างของภาพองค์ประกอบสีกับภาพเทาโดยใช้คำสั่งดังต่อไปนี้

DifferentRed = imsubtract(Red, RedGray);

DifferentBlue = imsubtract(Blue, BlueGray);

5. เปลี่ยนภาพผลต่างเป็นภาพไบนารี่โดยใช้ค่า Threshold เป็น 0.24(สีแดง) และ 0.1(สีน้ำเงิน) โดยใช้คำสั่งดังต่อไปนี้

```
RedBi = im2bw(DifferentRed, 0.24);
BlueBi = im2bw(DifferentBlue, 0.1);
```

6. ประมวลผลภาพสีเหลืองให้เป็นภาพไบนารี่ด้วยคำสั่งดังต่อไปนี้

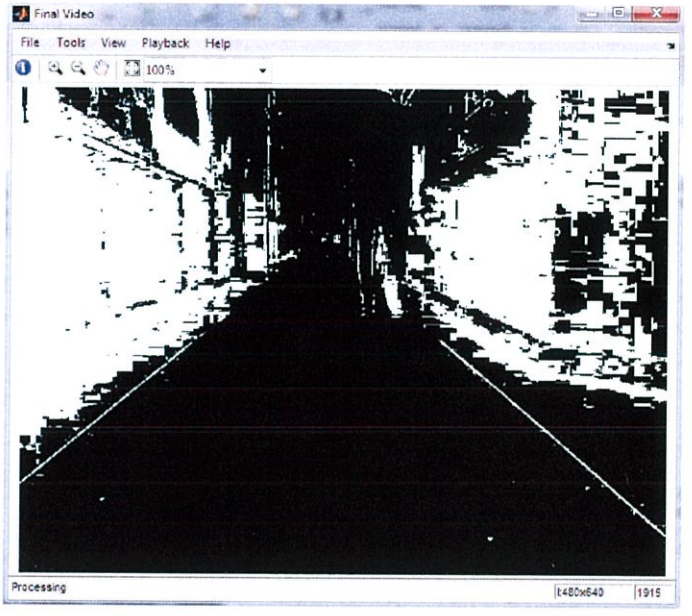
```
Red  = imsubtract( TestColors (: , : , 1), rgb2gray(TestColors) );
Red = im2bw(Red, 0.05);
Green  = imsubtract( TestColors(: , : , 2), rgb2gray(TestColors) );
Green = im2bw(Green, 0.05);
Yellow = imsubtract(Green, Red);
Yellow = imsubtract(Green, Yellow);
```

7.จากหลักการเพื่อให้ได้ภาพไบนารี่ของสีต่างๆ ในการใช้งานจริงภาพจะถูกรับมาจากกล้องแล้วจะมีการประมวลผลภาพตลอดเวลา ภาพไบนารี่ที่ได้นั้นจะถูกสร้างกรอบให้กับวัตถุสีนั้นๆ พร้อมทั้งกำหนดพิกัดระบุตำแหน่งวัตถุสีอีกด้วย โดยใช้คำสั่งต่อไปนี้

```
hblob = vision.BlobAnalysis('AreaOutputPort', false, ...
                    'CentroidOutputPort', true, 'BoundingBoxOutputPort',
                    true','MinimumBlobArea', 30, 'MaximumBlobArea', 5000,
                    'MaximumCount', 5);

hshapeinsRedBox = vision.ShapeInserter('BorderColor', 'Custom',
                                'CustomBorderColor', [1 0 0], 'Fill', true, ...
                                'FillColor', 'Custom', 'CustomFillColor',
                                 [1 0 0], 'Opacity', 0.4);

htextinsCent = vision.TextInserter('Text', '+ X:%4d, Y:%4d', ...
                            'LocationSource', 'Input port', 'Color', [1 1 0], ...
                            'FontSize', 14);
[centroidRed, bboxRed] = step(hblob, RedBi);
centroidRed = uint16(centroidRed);
vidIn = step(hshapeinsRedBox, vidIn, bboxRed);
for object = 1:1:length(bboxRed(:,1))
centRedX = centroidRed(object,1); centRedY = centroidRed(object,2);
vidIn = step(htextinsCent, vidIn, [centRedX centRedY], [centRedX-6
centRedY-9]);
end
```

4.2.3 ทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่

1. ใช้กล้องรับภาพเข้ามาแล้วทำการแปลงภาพจาก S-Channel ให้เป็นภาพไบนารี่ จะได้ภาพต้นแบบในการนำมาวิเคราะห์ดังรูป

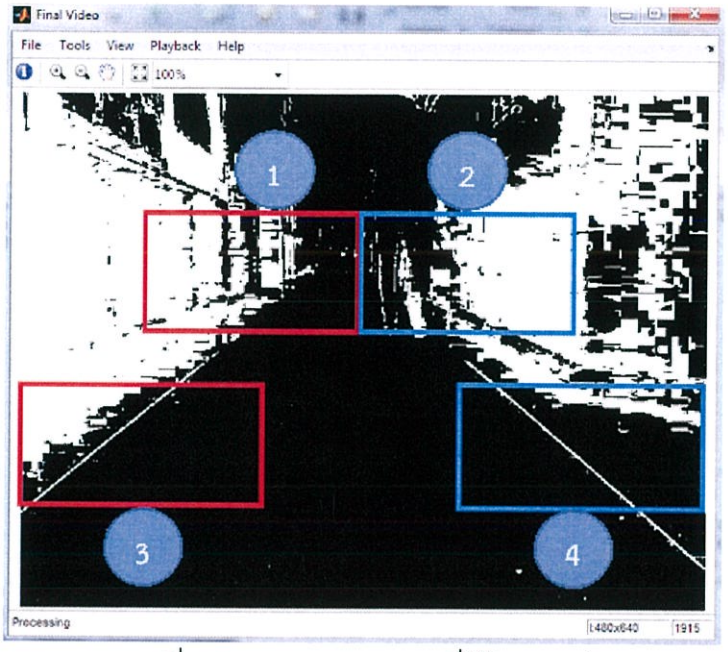


รูปที่ 4.3 ภาพจาก S-Channel ของส่วนหนึ่งบนเส้นทางเมื่อแปลงเป็นภาพไบนารี่

2. จากรูปที่ 4.3 เป็นภาพขนาด 640X480 โดยสีขาวแทนด้วยเลข 1 ส่วนสีดำแทนด้วยเลข 0 ในเมทริกซ์ mxn (480X640) ดังนั้นจึงวิเคราะห์การเปลี่ยนแปลงของภาพเมื่อหุ่นยนต์มีการเคลื่อนที่ไม่ตรงโดยการบวกค่าในเมทริกซ์ซึ่งมีขอบเขตดังนี้

ตารางที่ 4.1 ขอบเขตการวิเคราะห์

ตำแหน่งขอบเขตที่วิเคราะห์	ช่วงแถว	ช่วงคอลัมน์
1	120 - 240	120 - 320
2		320 - 520
3	270 - 390	1 - 240
4		400 - 640



รูปที่ 4.4 ภาพแสดงขอบเขตที่ใช้วิเคราะห์

4.2.4 สร้างคำสั่งรับข้อมูลจากอุลตร้าโซนิกเซ็นเซอร์

ใช้โปรแกรม ARDUINO ในการเขียนโปรแกรมคำสั่งรับข้อมูลจากอุลตร้าโซนิกเซ็นเซอร์ให้แก่บอร์ด โดยมีรายละเอียดดังนี้

```
const int pingPin1 = A3;
const int inPin1 = A2;
unsigned long Sonic1()
{
        long duration1, cm1;
        pinMode(pingPin1, OUTPUT);
        digitalWrite(pingPin1, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin1, HIGH);
        delayMicroseconds(5);
        digitalWrite(pingPin1, LOW);
        pinMode(inPin1, INPUT);
        duration1 = pulseIn(inPin1, HIGH);
        cm1 = duration1/ 29 / 2;
        return cm1;
}
```

4.2.5 วัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์

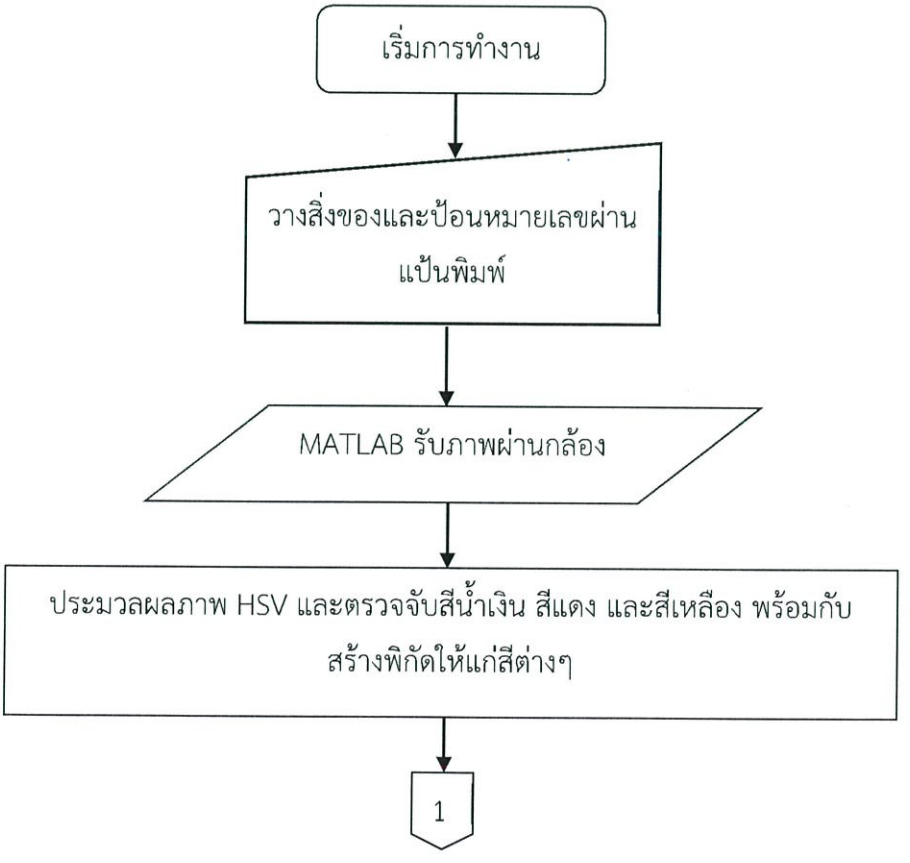
ทดลองวัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์เมื่อสั่งการทำงานดังนี้

ตารางที่ 4.2 ค่าลอจิกของการเคลื่อนที่รูปแบบต่างๆ

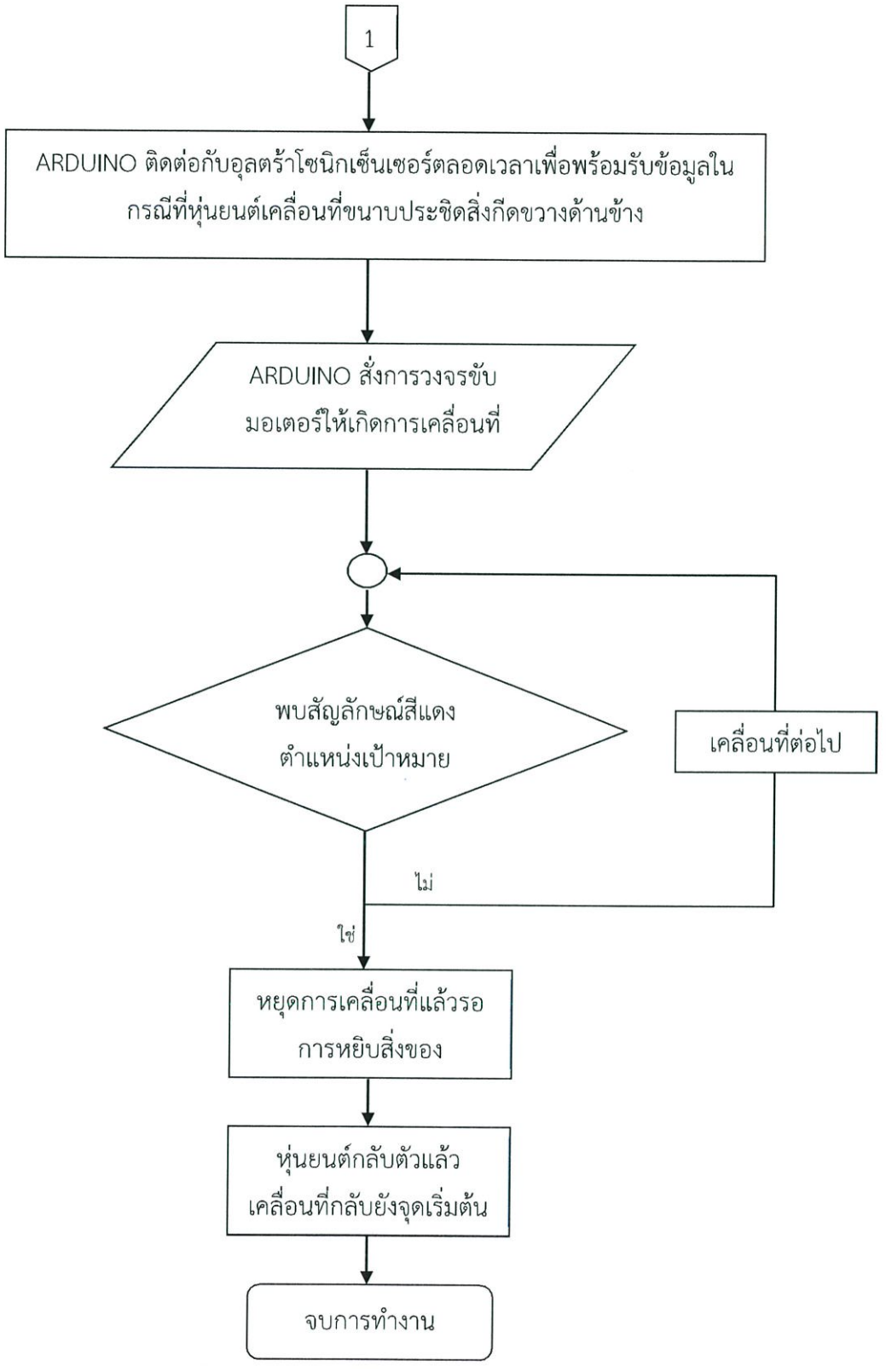
ฟังก์ชั่น	ลอจิกจาก ARDUINO			
	IN1	IN2	IN3	IN4
เดินหน้า	0	200	0	165
เลี้ยวขวา	0	255	0	0
เลี้ยวซ้าย	0	0	0	255

4.2.6 การทำงานของหุ่นยนต์

การทำงานของหุ่นยนต์โดยรวมเอาทุกส่วนมาประกอบกันสามารถเขียนเป็น Flow Chart ได้ดังต่อไปนี้



รูปที่ 4.5 Flow Chart แสดงการทำงานของหุ่นยนต์



รูปที่ 4.6 Flow Chart แสดงการทำงานของหุ่นยนต์ (ต่อ)

4.2.7 ทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่

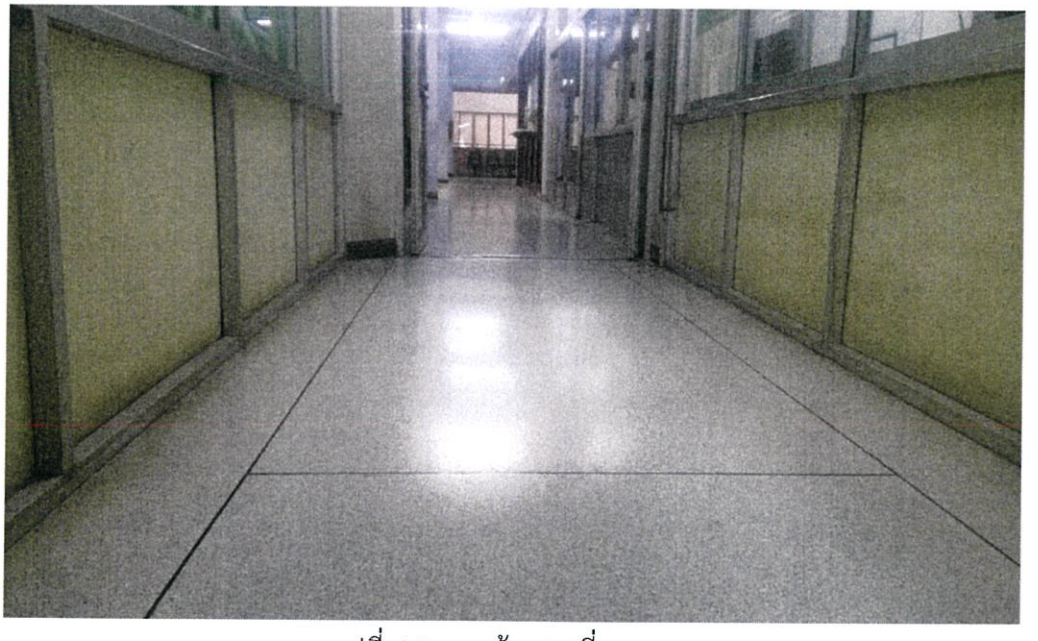
1. วัดระยะความคลาดเคลื่อนจากเส้นอ้างอิงของการหยุดเคลื่อนที่เมื่อหุ่นยนต์ตรวจจับสีแดงได้

2. ทำการทดลองซ้ำเพื่อพิจารณาค่าระยะความคลาดเคลื่อน

4.3 ผลการทดลอง

4.3.1 ผลการทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV

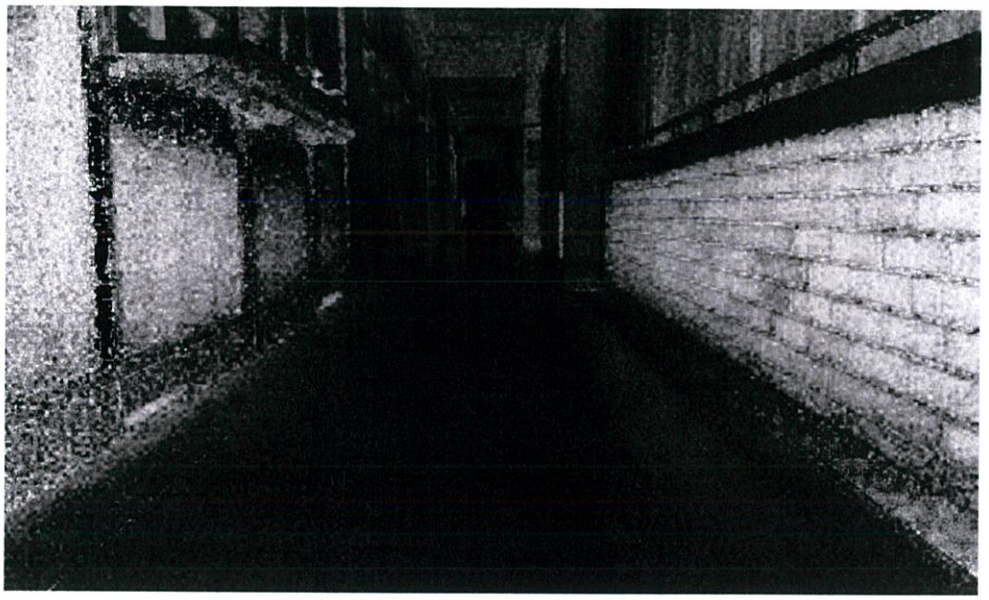
รูปที่ 4.7 ภาพต้นแบบที่ 1



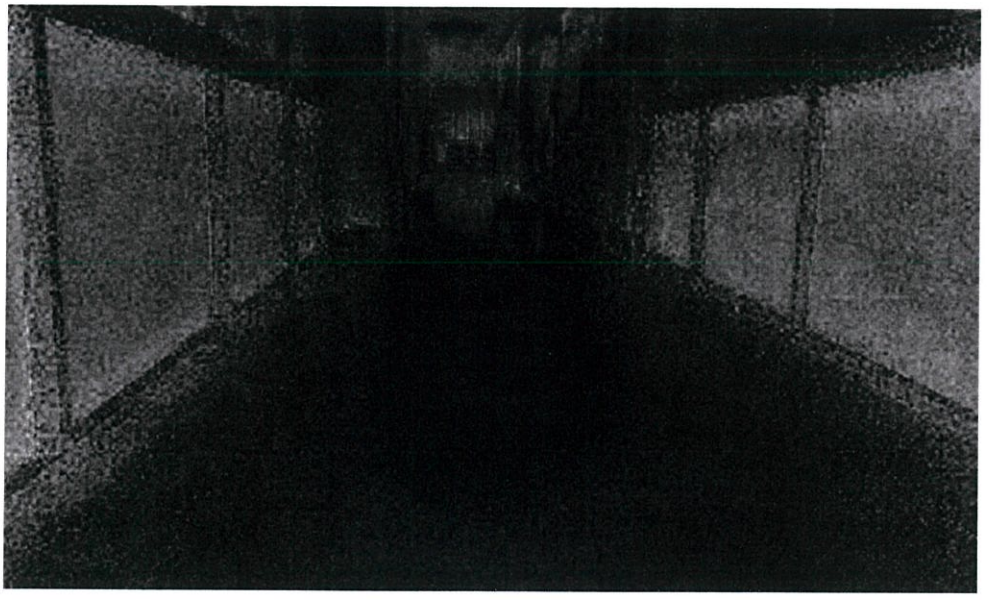
รูปที่ 4.8 ภาพต้นแบบที่ 2

รูปที่ 4.9 ภาพจาก H-Channel ต้นแบบที่ 1

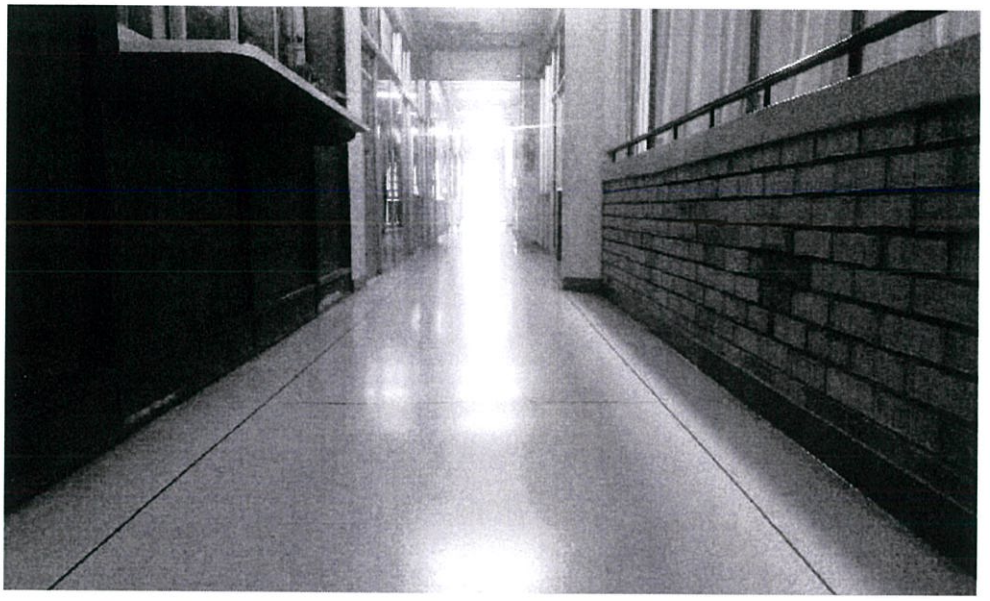
รูปที่ 4.10 ภาพจาก H-Channel ต้นแบบที่ 2



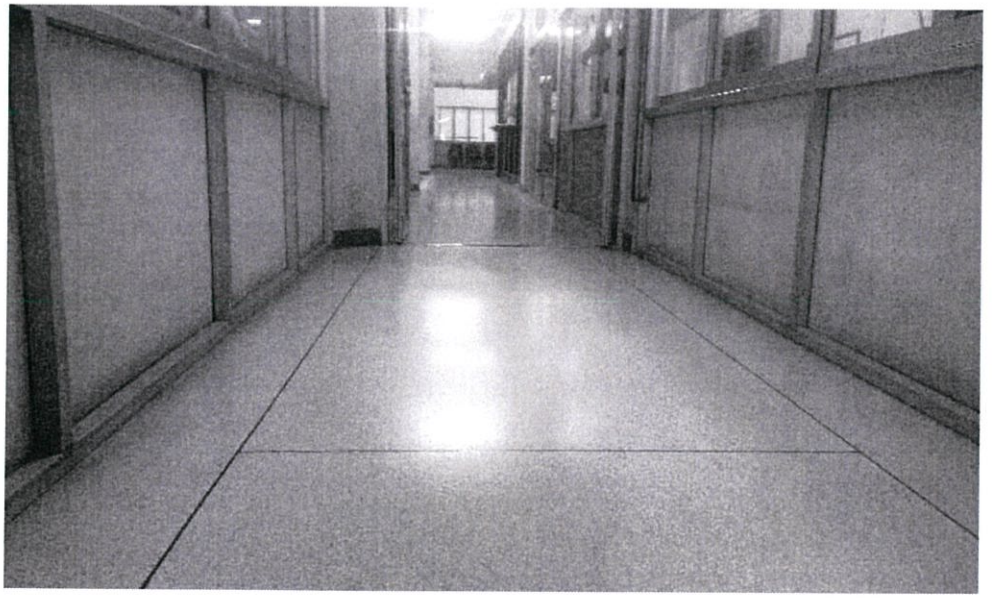
รูปที่ 4.11 ภาพจาก S-Channel ต้นแบบที่ 1



รูปที่ 4.12 ภาพจาก S-Channel ต้นแบบที่ 2

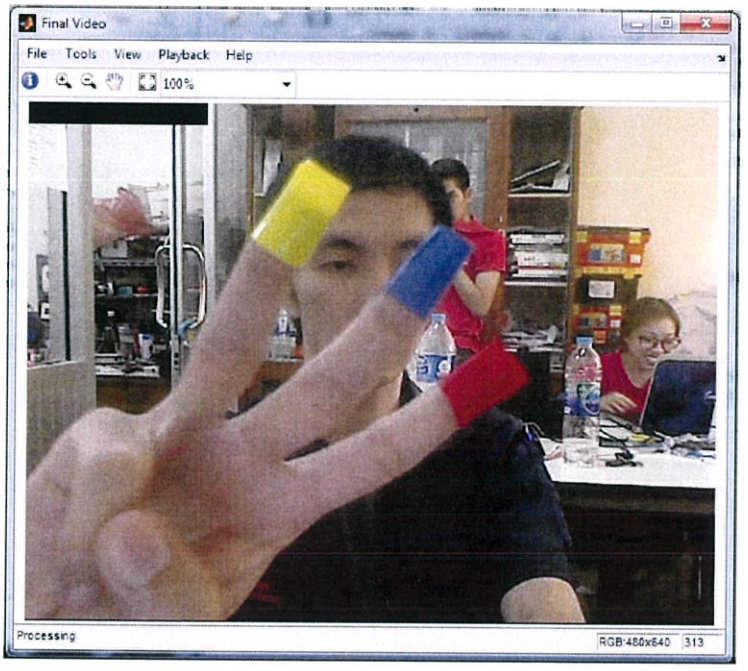


รูปที่ 4.13 ภาพจาก V-Channel ต้นแบบที่ 1



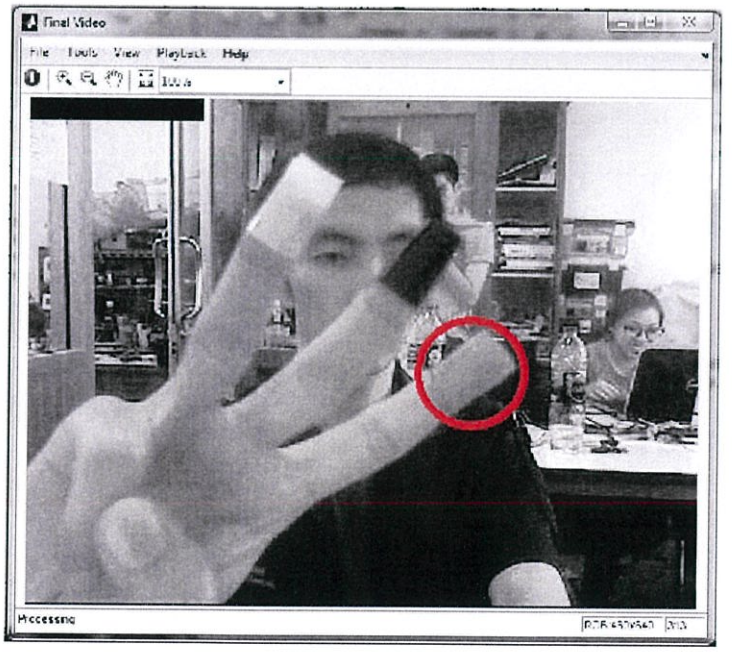
รูปที่ 4.14 ภาพจาก V-Channel ต้นแบบที่ 2

4.3.2 ผลการทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน

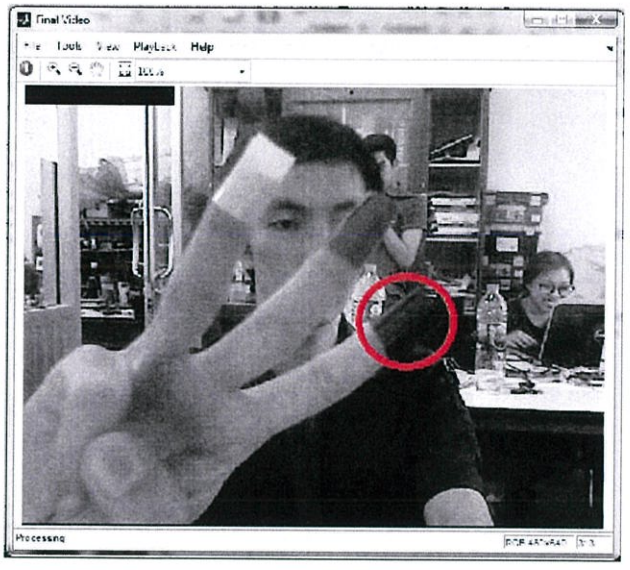


รูปที่ 4.15 ภาพต้นแบบที่ 3

4.3.2.1 ประมวลผลภาพสีแดง

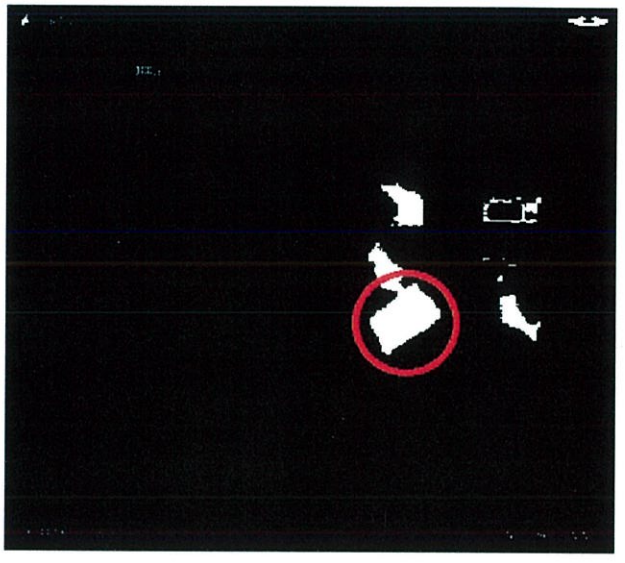


รูปที่ 4.16 ภาพองค์ประกอบสีแดง



รูปที่ 4.17 ภาพเทาของสีแดง

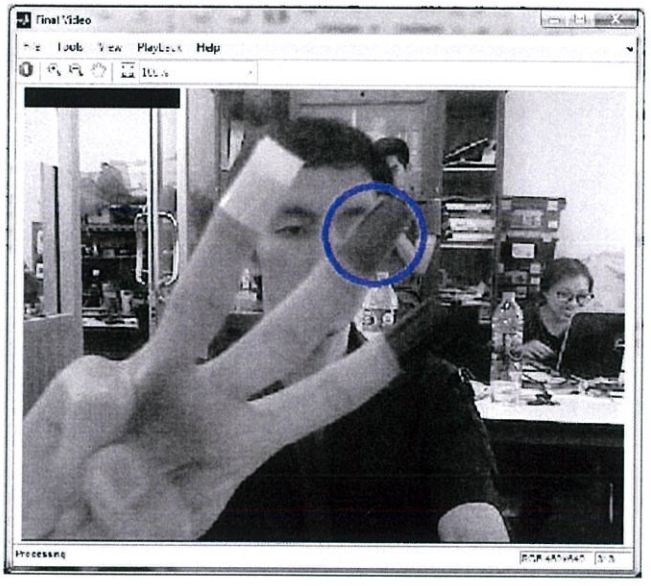
รูปที่ 4.18 ภาพผลต่างระหว่างภาพองค์ประกอบสีแดงกับภาพเทาของสีแดง



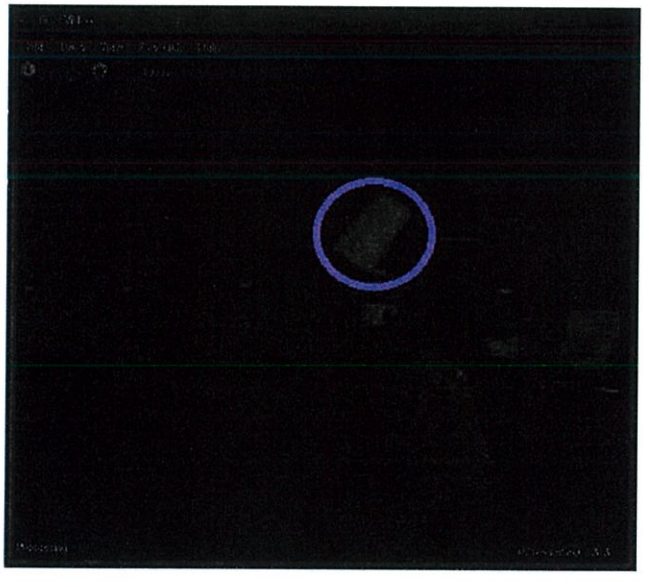
รูปที่ 4.19 ภาพไบนารี่ของผลต่าง(สีแดง)

4.3.2.2 ประมวลผลภาพสีน้ำเงิน

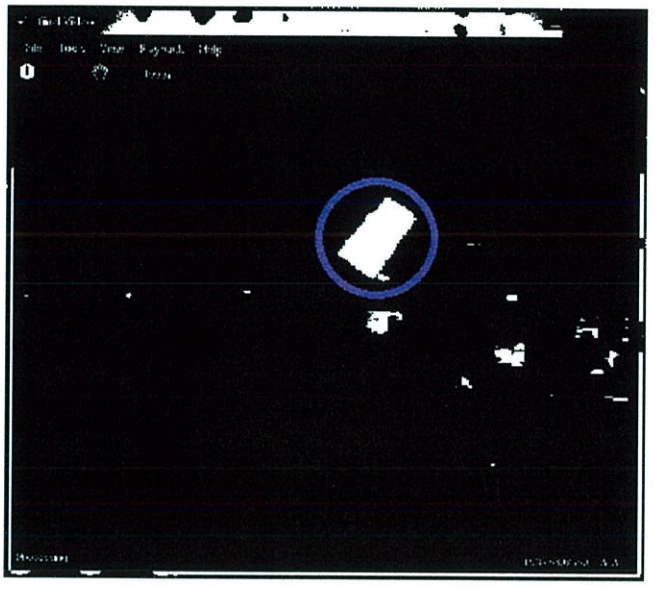
รูปที่ 4.20 ภาพองค์ประกอบสีน้ำเงิน



รูปที่ 4.21 ภาพเทาของสีน้ำเงิน



รูปที่ 4.22 ภาพผลต่างระหว่างภาพองค์ประกอบสีน้ำเงินกับภาพเทาของสีน้ำเงิน

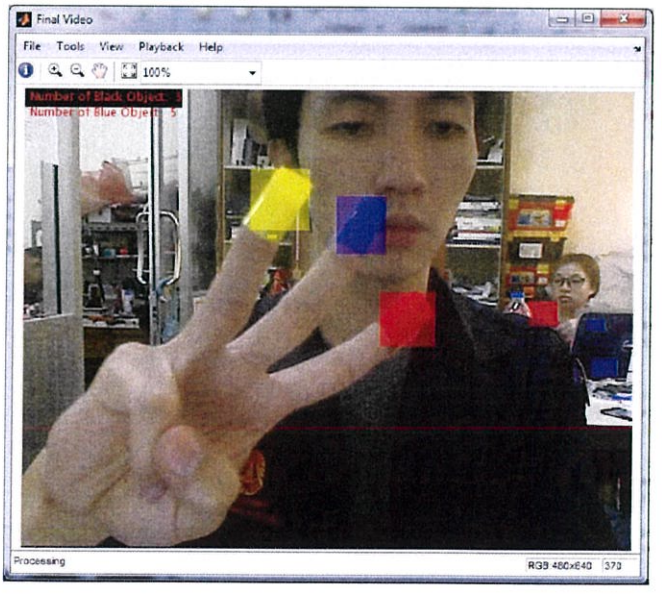


รูปที่ 4.23 ภาพไบนารี่ของผลต่าง(สีน้ำเงิน)

4.3.2.3 ประมวลผลภาพสีเหลือง

ภาพสีเหลืองนี้ใช้วิธีการเดียวกันกับการประมวลผลภาพสีแดงและสีน้ำเงิน แต่การประมวลผลจะใช้ภาพไบนารี่ของผลต่าง(สีแดง) กับ ภาพไบนารีของผลต่าง(สีเขียว) มาทำการหาผลต่างของภาพไบนารี่ของทั้งสอง ซึ่งขั้นตอนเป็นไปตามบรรทัดคำสั่งที่แสดงไว้ในการทดลอง

4.3.2.4 ประมวลผลภาพจากกล้อง



รูปที่ 4.24 ภาพหลังการประมวลผลสีต่างๆ

4.3.3 ผลการทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่

ตารางที่ 4.3 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 1

ตำแหน่งขอบเขตที่วิเคราะห์	ช่วงแถว	ช่วงคอลัมน์	ผลรวม		
			วิ่งตรง	เบี่ยงซ้าย	เบี่ยงขวา
1	120 - 240	120 - 320	13522	24061	7974
2		320 - 520	16088	5331	22812
3	270 - 390	1 - 240	5413	15785	1086
4		400 - 640	2462	702	8614

ตารางที่ 4.4 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 2

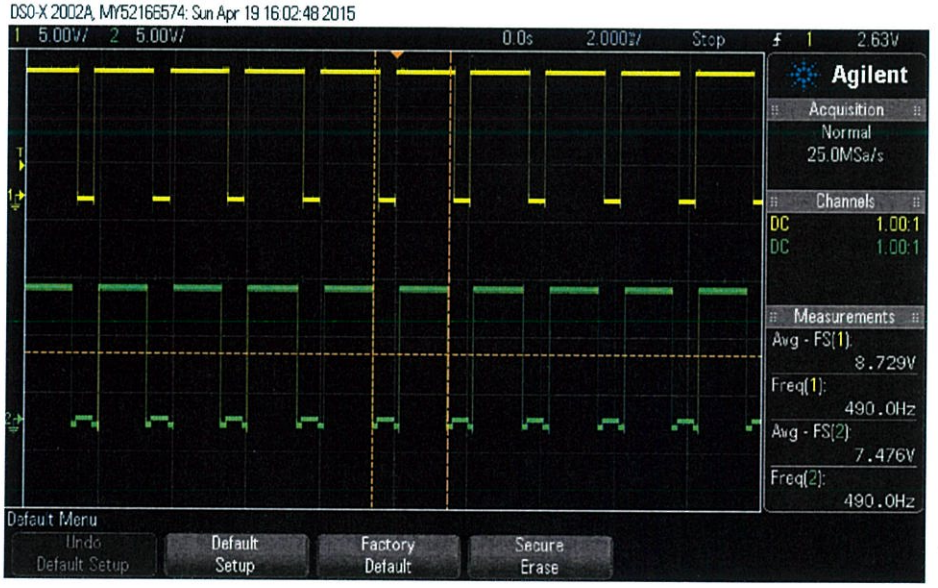
ผลรวม		
วิ่งตรง	เบี่ยงซ้าย	เบี่ยงขวา
14463	22107	8462
17015	6447	23258
731	9580	52
3419	129	10202

ตารางที่ 4.5 ตารางผลรวมค่าเชิงตัวเลขของภาพครั้งที่ 3

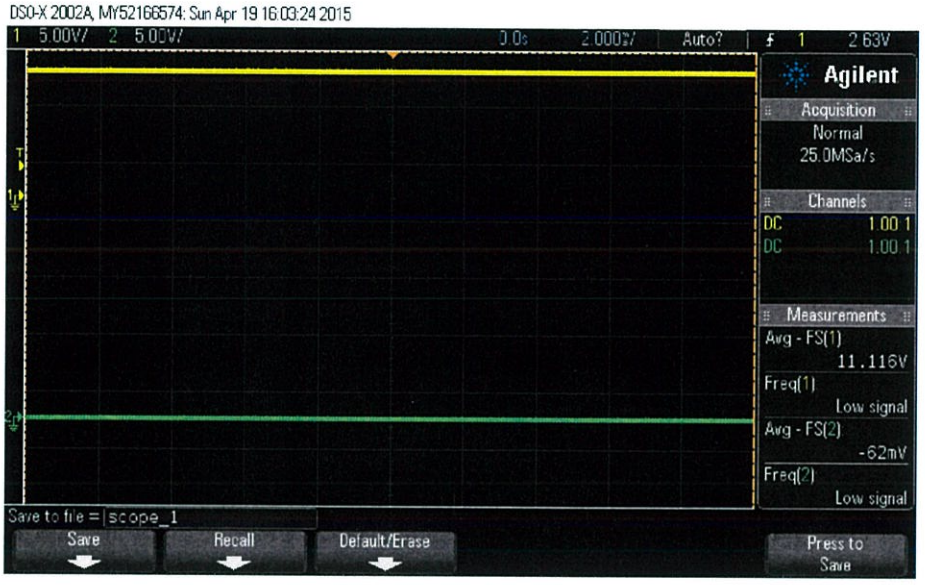
ผลรวม		
วิ่งตรง	เบี่ยงซ้าย	เบี่ยงขวา
16910	23414	7623
16188	8799	24511
1022	9003	133
422	31	5044

4.3.5 ผลการทดลองวัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์

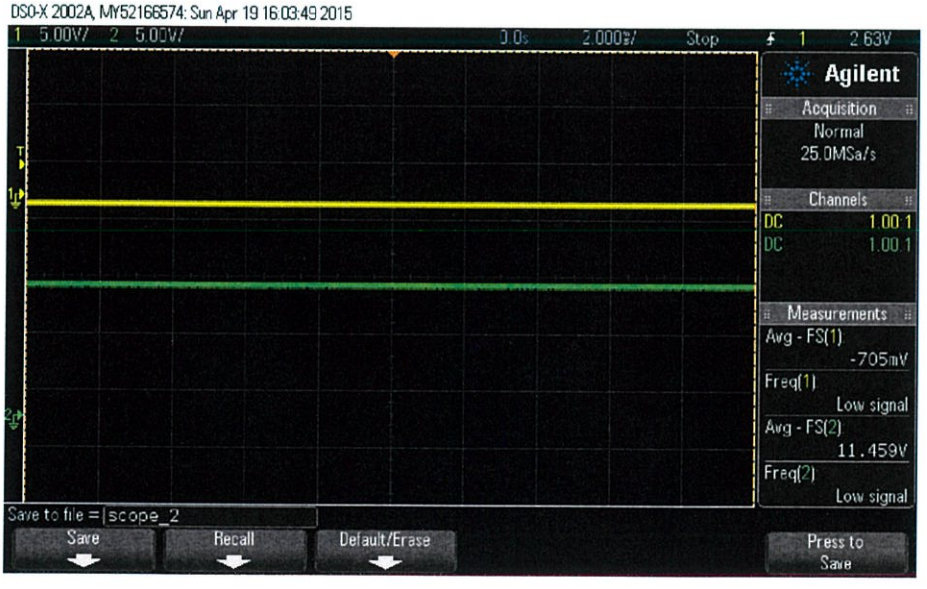
CH1: มอเตอร์ฝั่งซ้าย CH2: มอเตอร์ฝั่งขวา



รูปที่ 4.25 สัญญาณ PWM เมื่อเดินหน้า

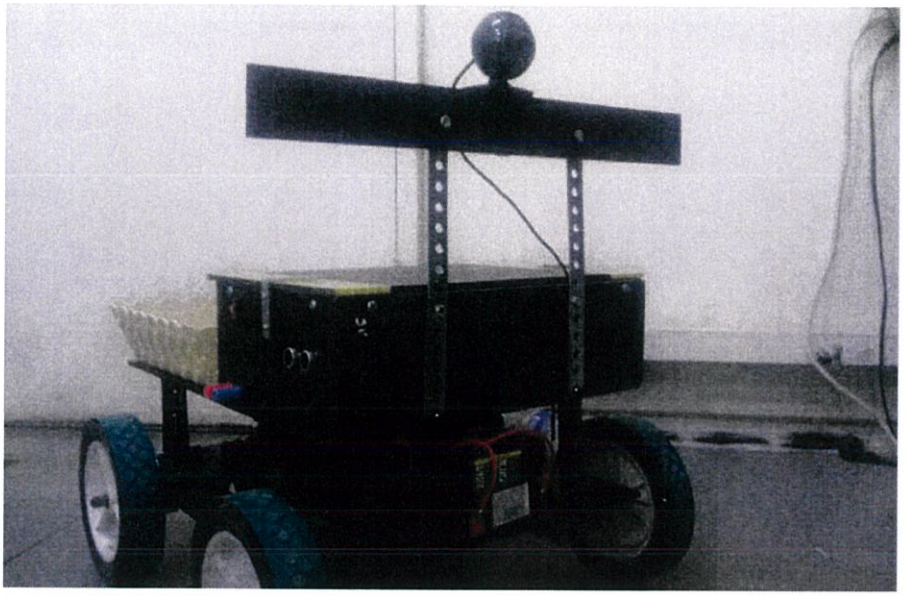


รูปที่ 4.26 สัญญาณ PWM เมื่อเลี้ยวขวา

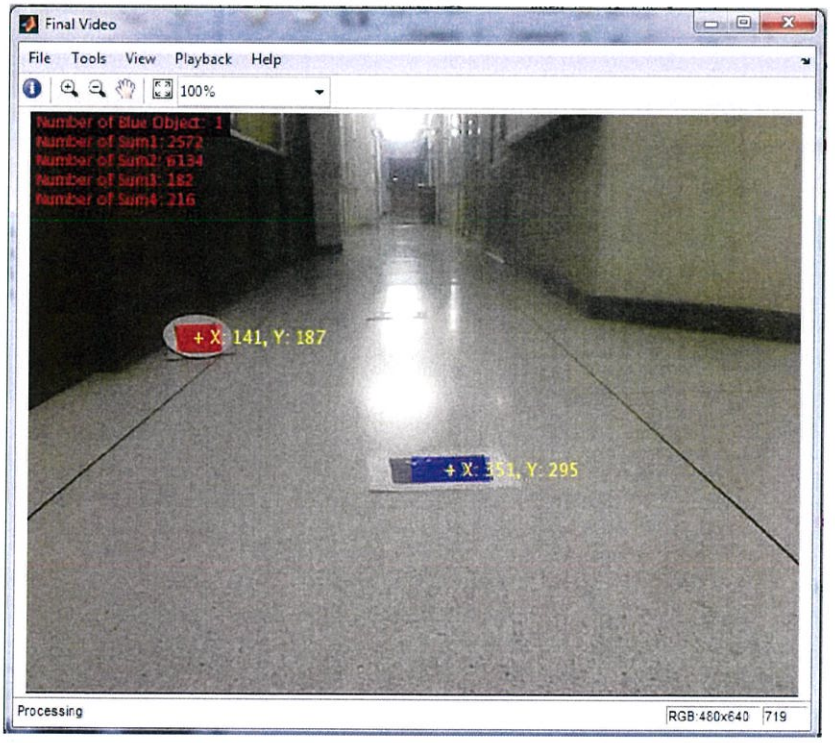


รูปที่ 4.27 สัญญาณ PWM เมื่อเลี้ยวซ้าย

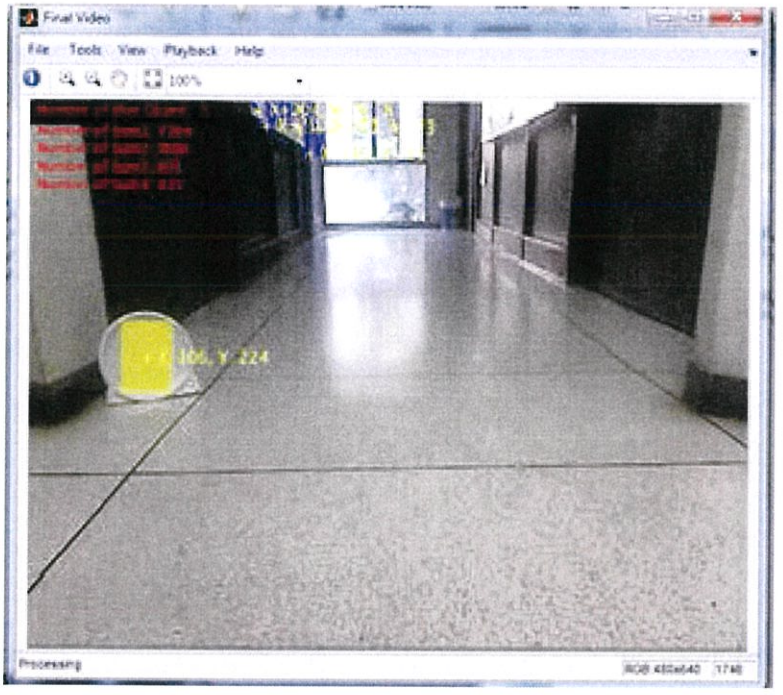
4.3.6 ผลการทดลองการทำงานของหุ่นยนต์



รูปที่ 4.28 หุ่นยนต์ “อิเล็ก 1”



รูปที่ 4.29 ภาพเส้นทาง



รูปที่ 4.30 ภาพเส้นทาง (ต่อ)

4.3.7 ผลการทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่

ตารางที่ 4.6 ผลความคลาดเคลื่อน

ครั้งที่	ระยะการคลาดเคลื่อนจากเส้นอ้างอิง(เซนติเมตร)
1	10 (หลังเส้นอ้างอิง)
2	7 (หลังเส้นอ้างอิง)
3	2 (หลังเส้นอ้างอิง)
4	5 (หน้าเส้นอ้างอิง)
5	8 (หน้าเส้นอ้างอิง)

บทที่ 5
สรุปผลการดำเนินงาน

5.1 สรุปผลการทดลอง

5.1.1 ทดลองหาภาพที่ได้จากแต่ละ channel ของเทคนิคระบบสี HSV

จากผลการทดลองภาพที่ได้ออกมาจากทั้งสาม channel นั้น ภาพจาก S-Channel มีลักษณะเด่นที่สุด คือผนังด้านข้างของทางเดินจะมีปริมาณสีขาวมากและสังเกตได้ชัดเจน ฉะนั้นในโครงงานชิ้นนี้ ภาพจาก S-Channel จึงถูกนำมาใช้ในการวิเคราะห์

5.1.2 ทดลองประมวลผลภาพสีแดง สีเหลือง และสีน้ำเงิน

จากผลการทดลองภาพที่ได้ออกมา สามารถนำไปใช้ประโยชน์ในควบคุมการเคลื่อนที่ของหุ่นยนต์ได้ เนื่องจากวัตถุสีต่างๆถูกตรวจจับได้พร้อมกันกับวัตถุสีได้ถูกกำหนดพิกัดบอกตำแหน่งเอาไว้อีกด้วย

5.1.3 ทดลองนำภาพจาก S-Channel มาวิเคราะห์เมื่อหุ่นยนต์เคลื่อนที่

จากตารางผลการทดลองจะเห็นได้ว่ามีเพียงตำแหน่งขอบเขตการวิเคราะห์ที่ 1 และ 2 ที่มีการเปลี่ยนแปลงผลรวมอย่างชัดเจน เมื่อการเคลื่อนที่เปลี่ยนไป คือเมื่อเคลื่อนที่ตรงผลรวมที่ตำแหน่งที่ 1 และ 2 จะมีค่าอยู่ระหว่าง 10000 ถึง 20000 จากนั้นเมื่อเคลื่อนที่เบี่ยงซ้ายผลรวมที่ตำแหน่งที่ 1 จะมีค่ามากกว่า 20000 ส่วนตำแหน่งที่ 2 จะมีค่าน้อยกว่า 10000 และเมื่อมีการเบี่ยงขวาผลรวมตำแหน่งที่ 1 จะมีค่าน้อยกว่า 10000 ส่วนตำแหน่งที่2 จะมีค่ามากกว่า 20000 ส่วนทางด้านตำแหน่ง ที่ 3 กับ 4 ค่าผลรวมที่ได้มายังไม่เหมาะสมที่จะนำไปใช้งานเนื่องจากค่าที่ได้มาไม่มีขอบเขตที่ชัดเจน

5.1.4 สร้างคำสั่งรับข้อมูลจากอุลตร้าโซนิกเซ็นเซอร์

ข้อมูลที่ได้จากอุลตร้าโซนิกเซ็นเซอร์คือค่าเวลาจากการส่งแล้วรับคลื่นเสียงมีหน่วยเป็นไมโครวินาที เมื่อเข้าสูตรคำนวณหาระยะทางจะได้ค่าออกมาในหน่วยเซนติเมตร

5.1.5 วัดสัญญาณ PWM ที่ได้จากวงจรขับมอเตอร์

เมื่อกำหนด Duty Ratio เป็น 255 สัญญาณที่วงจรขับมอเตอร์ให้ออกมาจะเป็นแรงดันไฟตรง แต่เมื่อเปลี่ยนค่า Duty Ratio สัญญาณที่วงจรขับมอเตอร์ให้ออกมาจะเป็น PWM ที่มีความถี่ 490 Hz โดย Duty Ratio มีค่าอยู่ในช่วง 0 - 255

5.1.6 การทำงานของหุ่นยนต์

โปรแกรม MATLAB สามารถตรวจจับสีต่างๆพร้อมกับกำหนดพิกัดให้แก่วัตถุสีนั้นๆได้ โดยหุ่นยนต์สามารถขนส่งสิ่งของไปยังเป้าหมายแล้วกลับมายังจุดเริ่มต้นได้

5.1.7 ทดลองความคลาดเคลื่อนของการหยุดเคลื่อนที่

จากการทดลองการหยุดเคลื่อนที่ นั้นผลที่ได้มาจะสังเกตได้ว่าส่วนใหญ่หุ่นยนต์จะหยุดหลังจุดอ้างอิงซึ่งอาจเป็นผลเนื่องจากการตรวจจับของกล้องที่ใช้และพลังงานของแบตเตอรี่ ทำให้เมื่อทดลองครั้งแรกนั้นแบตเตอรี่มีประจุเต็มจึงทำให้การเคลื่อนที่ของหุ่นยนต์นั้นมีความเร็วตามที่ต้องการได้ แต่เมื่อทำการทดลองซ้ำไปเรื่อยๆ ประจุในแบตเตอรี่เริ่มที่จะลดน้อยลงจึงทำให้หุ่นยนต์เคลื่อนที่ช้าลงซึ่งอาจเป็นผลให้การหยุดการเคลื่อนที่นั้นหยุดก่อนถึงเส้นอ้างอิงและจากปัญหาของตัวกล้องที่ใช้นั้นมีการตรวจจับภาพได้ไม่ชัดเจน จึงทำให้การตรวจจับสีนั้นช้าซึ่งเมื่อตรวจจับช้าก็จะส่งผลให้การหยุดการเคลื่อนไหวของหุ่นยนต์นั้นช้าลงไปด้วย

5.2 สรุปผลโครงงาน

จากการดำเนินงานในทุกๆขั้นตอนที่ได้กล่าวมาแล้วนั้น สามารถสรุปการทำงานของหุ่นยนต์ได้ว่า หุ่นยนต์จะรับค่าตำแหน่งเป้าหมายผ่าน keypad และจะมีปุ่มสวิทช์ตรวจสอบการวางสิ่งของ โดยการเคลื่อนที่เกิดจากการประมวลผลภาพและอุลตร้าโซนิกเซ็นเซอร์ การประมวลผลภาพนั้นใช้โปรแกรม MATLAB ซึ่งรับภาพมาจากกล้องแล้วนำไปประมวลผลภาพด้วยเทคนิคระบบสี HSV เฉพาะส่วน S-Channel และการประมวลผลภาพสีต่างๆ โดยสีน้ำเงินจะถูกใช้ในการควบคุมให้หุ่นยนต์เคลื่อนที่อยู่ในเส้นทางตรง โดยค่าพิกัดบอกตำแหน่งของวัตถุสีน้ำเงินที่ประมวลผลได้จะถูกนำมาใช้งาน ซึ่งมีขอบเขตอยู่ว่าหากค่าแกน X อยู่ในช่วง 310 ถึง 330 จะถือว่าหุ่นยนต์ยังเคลื่อนที่ตรง แต่หากค่าแกน X น้อยกว่า 310 แสดงว่าหุ่นยนต์เคลื่อนที่เบี่ยงขวา ส่วนถ้าค่าแกน X มากกว่า 330 แสดงว่าหุ่นยนต์เคลื่อนที่เบี่ยงซ้าย ซึ่งในทุกกรณีโปรแกรม MATLAB จะส่งค่าสถานะให้แก่บอร์ด ARDUINO เพื่อบังคับวงจรขับมอเตอร์ให้เกิดการเคลื่อนที่ ในส่วนของการหยุดนั้นจะใช้สีแดงเป็นเงื่อนไข โดยหากตรวจจับวัตถุสีแดงได้โปรแกรมจะส่งค่าสถานะให้แก่ ARDUINO เพื่อสั่งการวงจรขับมอเตอร์ให้เกิดการหยุดเคลื่อนที่ นอกจากนี้ยังใช้อุลตร้าโซนิกเซ็นเซอร์ช่วยควบคุมให้หุ่นยนต์เคลื่อนที่ห่างจากสิ่งกีดขวางด้านข้างอย่างน้อย 50 cm เมื่อหุ่นยนต์เคลื่อนที่ส่งของถึงจุดหมายเรียบร้อยแล้ว หุ่นยนต์จะกลับตัวแล้วเคลื่อนที่กลับสู่จุดเริ่มต้นดังเดิม

หุ่นยนต์ “อิเล็ก 1” สามารถเคลื่อนที่เพื่อขนส่งสิ่งของน้ำหนักเบา เช่น เอกสาร เป็นต้น ไปยังตำแหน่งเป้าหมายตามที่กำหนดได้อย่างถูกต้องโดยอัตโนมัติ และสามารถกลับมายังจุดเริ่มต้นได้อีกด้วย เนื่องด้วยประสิทธิภาพของโครงสร้างหุ่นยนต์และผลกระทบจากสิ่งแวดล้อมรอบตัวมีผลต่อการเคลื่อนที่ดังจะได้อธิบายในหัวข้อ 5.3 ต่อไป จึงเป็นข้อจำกัดแก่หุ่นยนต์ที่จะขนส่งได้เพียงสิ่งของที่น้ำหนักไม่มากเช่นเอกสาร หนังสือ หรืออื่นๆที่น้ำหนักอยู่ในข่ายเดียวกันเท่านั้น ทั้งนี้ผู้จัดทำขอจำกัดน้ำหนักไว้ที่ไม่เกิน 5 กิโลกรัม

หุ่นยนต์ “อิเล็ก 1” เป็นหุ่นยนต์ที่ถูกคิดค้นการทำงานขึ้นมาในปีการศึกษา 2557 ถือเป็นหุ่นยนต์ต้นแบบแก่นักศึกษาสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ในปีการศึกษาต่อไปที่สนใจได้ เนื่องจากหุ่นยนต์นี้ยังสามารถนำไปปรับปรุงและพัฒนาเพื่อจุดประสงค์อื่นๆได้อีกมากมาย ทั้งนี้โดยมีพื้นฐานวิธีการเคลื่อนที่และการหยุดจากโครงงานชิ้นนี้ที่สามารถนำไปต่อยอดได้

5.3 วิจารณ์ผลการดำเนินงาน

จากการดำเนินงานทั้งหมดของโครงงานนี้ได้พบว่าปัญหาจากสิ่งแวดล้อมซึ่งควบคุมได้ยากนั้นมีผลต่อการทำงานของหุ่นยนต์อยู่ไม่น้อย เช่นแสงจากหลอดไฟและแสงจากดวงอาทิตย์ที่ส่งผลให้การประมวลผลภาพไม่มีประสิทธิภาพ เช่นไม่สามารถตรวจจับสีน้ำเงินได้ หรือการตรวจจับสีส้มเป็นสีแดงเป็นต้น นอกจากนี้วัตถุที่ถูกวางไว้ชั่วคราวหรือถาวรบนเส้นทางเดินซึ่งทำให้ความกว้างของทางเดินแคบลง ได้ส่งผลต่อการประมวลผลภาพและค่าระยะที่ได้จากเซ็นเซอร์ ส่งผลทำให้การเคลื่อนที่มีปัญหาอีกด้วย อีกทั้งผลจากประสิทธิภาพของแบตเตอรี่ยังมีผลต่อการเคลื่อนที่อีกด้วยโดยเมื่อพลังงานจากแบตเตอรี่ลดลง กำลังในการขับเคลื่อนก็น้อยลงตามไปด้วย รวมถึงชิ้นส่วนประกอบของโครงสร้างหุ่นยนต์ยังมีผลต่อการเคลื่อนที่อีกด้วยเช่นกัน เช่นประสิทธิภาพของมอเตอร์สองตัวไม่เท่ากัน รวมถึงส่วนของล้อและชิ้นส่วนเกียร์ต่างๆอาจส่งผลให้เกิดปัญหาได้เช่นเดียวกัน

บรรณานุกรม

เอกสารอ้างอิง

[1] รศ.ดร.ชูชาติ ปิณฑวิรุจน์,"การประมวลผลภาพดิจิตอลด้วย MATLAB",คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เว็บอ้างอิง

[2] arindambose.com/blog/

[3] http://www.mathworks.com/help/vision/examples.html

[4] http://www.arduitronics.com/article/arduino-and-motor-control-part-2

[5] http://arduino.cc/en/Tutorial/SecretsOfArduinoPWM

[6] http://matabthai.blogspot.com/2013/04/matlab.html

[7] http://sk-mce.blogspot.com/2012/05/dc-motor-control-with-l298-l298.html

[8] http:// www.arduitronics.com/product/8/arduino-uno-r3-free-usb-cable

[9] http://digi.library.tu.ac.th/thesis/st/0251/03CHAPTER2.pdf

[10] http://www.arduino.cc/en/Main/arduinoBoardMega2560

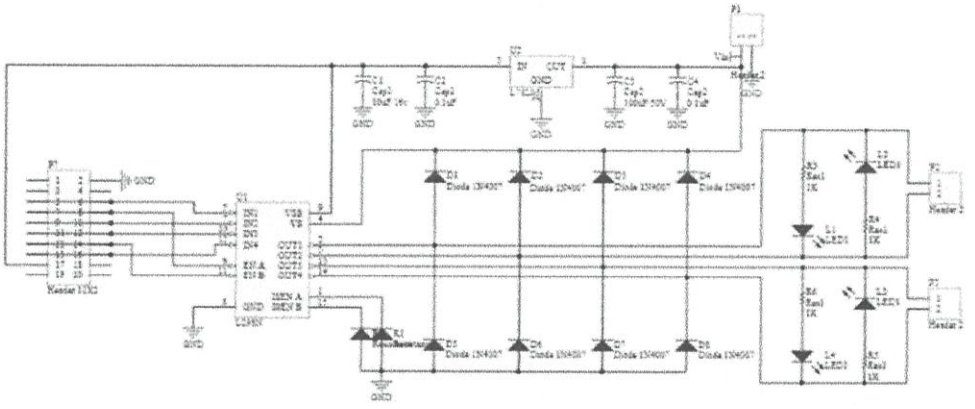
[11] http://www.elecfreaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf

[12] http://cpre.kmutnb.ac.th/esl/learning/index.php?article=4x4-keypad

[13] http://www.lib.ku.ac.th/KUthesis/2551/SarawutSUP/SarawutSUPAll.pdf

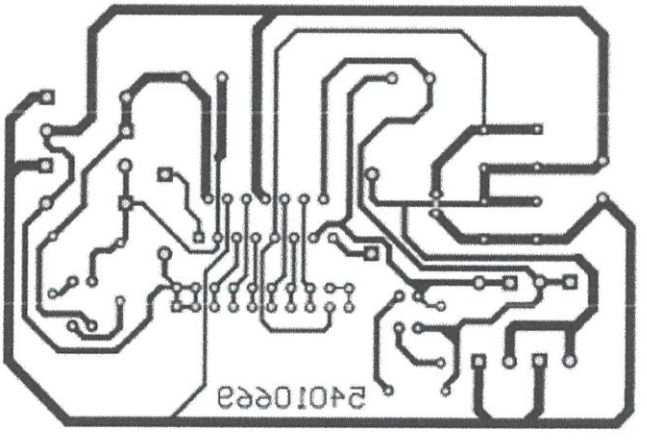
ภาคผนวก ก
วงจรขับมอเตอร์กระแสตรง

ก.1 แบบวงจรขับมอเตอร์กระแสตรง



รูปที่ ก.1 วงจรขับมอเตอร์กระแสตรง

ก.2 ลายปริ้นท์วงจรขับมอเตอร์กระแสตรง



รูปที่ ก.2 ลายปริ้นท์วงจรขับมอเตอร์กระแสตรง

ภาคผนวก ข
โค้ดโปรแกรมการทำงาน

ข.1 โค้ดคำสั่งการตรวจจับเส้นสีดำและวัตถุสีน้ำเงินโดยโปรแกรม MATLAB

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program Name : HSV and Colors Detection
% Description : Route Direction of Robot in Live Video
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initialization
clc;
clear all;
close all;

blueThresh = 0.054;     % Threshold of blue
blackThresh = 0.048;    % Threshold of black
redthresh = 0.03;       % Red Threshold of yellow
greenthresh = 0.03;     % Green Threshold of yellow
greenThresh= 0.075;     % Threshold of green
redThresh=0.15;         % Threshold of red

vidDevice = imaq.VideoDevice('winvideo', 2,'MJPG_640x480', ...
            'ROI', [1 1 640 480], ...
            'ReturnedColorSpace', 'rgb');
vidInfo = imaqhwinfo(vidDevice);       % Acquire input video property
hblob = vision.BlobAnalysis('AreaOutputPort', false, ... % Set blob analysis handling
                    'CentroidOutputPort', true, ...
                    'BoundingBoxOutputPort', true', ...
                    'MinimumBlobArea', 30, ...
                    'MaximumBlobArea', 5000, ...
                    'MaximumCount', 5);
hblob2 = vision.BlobAnalysis('AreaOutputPort', false, ... % Set blob analysis handling
                    'CentroidOutputPort', true, ...
                    'BoundingBoxOutputPort', true', ...
                    'MinimumBlobArea', 300, ...
                    'MaximumBlobArea', 5000, ...
                    'MaximumCount', 5);
hblob3 = vision.BlobAnalysis('AreaOutputPort', false, ... % Set blob analysis handling
                    'CentroidOutputPort', true, ...
                    'BoundingBoxOutputPort', true', ...
                    'MinimumBlobArea', 300, ...
```

```
                                'MaximumBlobArea', 3000, ...
                                'MaximumCount', 5);
% Set Blue box handling
hshapeinsBlueBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                        'CustomBorderColor', [0 0 1], ...
                                        'Fill', true, ...
                                        'FillColor', 'Custom', ...
                                        'CustomFillColor', [0 0 1], ...
                                        'Opacity', 0.4);
% Set Yellow box handling
hshapeinsYellowBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                          'CustomBorderColor', [1 1 0], ...
                                          'Fill', true, ...
                                          'FillColor', 'Custom', ...
                                          'CustomFillColor', [1 1 0], ...
                                          'Opacity', 0.4);
% Set Red box handling
hshapeinsRedBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                       'CustomBorderColor', [1 0 0], ...
                                       'Fill', true, ...
                                       'FillColor', 'Custom', ...
                                       'CustomFillColor', [1 0 0], ...
                                       'Opacity', 0.4);
% Set Green box handling
hshapeinsGreenBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                         'CustomBorderColor', [0 1 0], ...
                                         'Fill', true, ...
                                         'FillColor', 'Custom', ...
                                         'CustomFillColor', [0 1 0], ...
                                         'Opacity', 0.4);
% Set Black box handling
hshapeinsBlackBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                         'CustomBorderColor', [0 0 0], ...
                                         'Fill', true, ...
                                         'FillColor', 'Custom', ...
                                         'CustomFillColor', [0 0 0], ...
                                         'Opacity', 0.4);
```

```
htextinsBlue = vision.TextInserter('Text', 'Number of Blue Object: %2d', ...
                                   'Location', [7 2], ...
                                   'Color', [1 0 0], ...
                                   'FontSize', 12);
htextinsSum1 = vision.TextInserter('Text', 'Number of Sum1: %2d', ...
                                   'Location', [7 18], ...
                                   'Color', [1 0 0], ...
                                   'FontSize', 12);
htextinsSum2 = vision.TextInserter('Text', 'Number of Sum2: %2d', ...
                                   'Location', [7 34], ...
                                   'Color', [1 0 0], ...
                                   'FontSize', 12);
htextinsSum3 = vision.TextInserter('Text', 'Number of Sum3: %2d', ...
                                   'Location', [7 50], ...
                                   'Color', [1 0 0], ...
                                   'FontSize', 12);
htextinsSum4 = vision.TextInserter('Text', 'Number of Sum4: %2d', ...
                                   'Location', [7 66], ...
                                   'Color', [1 0 0], ...
                                   'FontSize', 12);
htextinsCent = vision.TextInserter('Text', '+ X:%4d, Y:%4d', ... % set text for centroid
                                   'LocationSource', 'Input port', ...
                                   'Color', [1 1 0], ...
                                   'FontSize', 14);
hVideoIn = vision.VideoPlayer('Name', 'Final Video', ... % Output video player
        'Position', [100 100 vidInfo.MaxWidth+20 vidInfo.MaxHeight+30]);

nFrame = 1; % Frame number initialization
out=0; %out of loop variable
%%%%%%%%%%% Other Variable %%%%%%%%%%
FB=0;
a=0;
b=0;
c=0;
cc=0;
d=0;
dd=0;
```

```
e=0;
f=0;
ff=0;
centXBlack=0;
centYBlack=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s = serial('COM6');
s.InputBufferSize = 50000;
s.BaudRate = 9600;
s.DataBits=8;
fopen(s);

%% Processing Loop
while(out<50)
rgbFrame = step(vidDevice); % Acquire single frame

%%%%%%%%%%%%%% Blue %%%%%%%%%%%%%%%%
diffFrameBlue = imsubtract(rgbFrame(:,:,3), rgb2gray(rgbFrame)); % Get red
component of the image
diffFrameBlue = medfilt2(diffFrameBlue, [3 3]); % Filter out the noise by using median
filter
binFrameBlue = im2bw(diffFrameBlue, blueThresh); % Convert the image into binary
image with the red objects as white
[centroidBlue, bboxBlue] = step(hblob, binFrameBlue); % Get the centroids and
bounding boxes of the blobs
centroidBlue = uint16(centroidBlue); % Convert the centroids into Integer for further
steps

%%%%%%%%%%%%%% Yellow %%%%%%%%%%%%%%%
red  = imsubtract(rgbFrame(:,:,1), rgb2gray(rgbFrame));
red = im2bw(red, redthresh);
red = im2uint8(red);
green  = imsubtract(rgbFrame(:,:,2), rgb2gray(rgbFrame));
green = im2bw(green, greenthresh);
green = im2uint8(green);
Yellow = imsubtract(green, red);
Yellow = imsubtract(green, Yellow);
```

```
Yellow = logical(Yellow);
[centroidYellow, bboxYellow] = step(hblob2,Yellow);
centroidYellow = uInt16(centroidYellow);

%%%%%%%%%%%% Green %%%%%%%%%%%%%
diffFrameGreen = imsubtract(rgbFrame(:,:,2), rgb2gray(rgbFrame));
diffFrameGreen = medfilt2(diffFrameGreen, [3 3]);
binFrameGreen = im2bw(diffFrameGreen, greenThresh);
[centroidGreen, bboxGreen] = step(hblob2, binFrameGreen); % Get the centroids and
bounding boxes of the blobs
centroidGreen = uint16(centroidGreen);

%%%%%%%%%%%% Red %%%%%%%%%%%%%
diffFrameRed = imsubtract(rgbFrame(:,:,1), rgb2gray(rgbFrame));
diffFrameRed = medfilt2(diffFrameRed, [3 3]);
binFrameRed = im2bw(diffFrameRed, redThresh);
[centroidRed, bboxRed] = step(hblob2, binFrameRed);
centroidRed = uint16(centroidRed);

%%%%%%%%%%%% Black %%%%%%%%%%%%%
bwredFrame=im2bw(rgbFrame(:,:,1),blackThresh);
bwgreenFrame=im2bw(rgbFrame(:,:,2),blackThresh);
bwblueFrame=im2bw(rgbFrame(:,:,3),blackThresh);
binFrameBlack=bwredFrame&bwgreenFrame&bwblueFrame;
binFrameBlack=~binFrameBlack;
[centroidBlack, bboxBlack] = step(hblob, binFrameBlack);
centroidBlack = uint16(centroidBlack); % Convert the centroids into Integer for further
steps

%%%%%%%%% HSV : Saturation Channel %%%%%%%%%%
[~,s1,~] = rgb2hsv(rgbFrame);
b1=im2bw(s1,0.25);
a1=b1(220:340,1:240);
a2=b1(100:220,120:320);
a3=b1(310:430,1:240);
a4=b1(310:430,400:640);
a5=b1(220:340,400:640);
```

```
a6=b1(100:220,340:520);

rgbFrame(1:20,1:165,:) = 0; % put a black region on the output stream
vidIn = step(hshapeinsBlueBox, rgbFrame, bboxBlue); % Instert the blue box
vidIn = step(hshapeinsYellowBox, vidIn, bboxYellow);% Instert the yellow box
vidIn = step(hshapeinsGreenBox, vidIn, bboxGreen);% Instert the green box
vidIn = step(hshapeinsRedBox, vidIn, bboxRed);% Instert the red box

for object = 1:1:length(bboxRed(:,1)) % Write the corresponding centroids for red
centRedX = centroidRed(object,1); centRedY = centroidRed(object,2);
vidIn = step(htextinsCent, vidIn, [centRedX centRedY], [centRedX-6 centRedY-9]);
end

for object = 1:1:length(bboxGreen(:,1)) % Write the corresponding centroids for green
centGreenX = centroidGreen(object,1); centGreenY = centroidGreen(object,2);
vidIn = step(htextinsCent, vidIn, [centGreenX centGreenY], [centGreenX-6 centGreenY-
9]);
end

for object = 1:1:length(bboxBlue(:,1)) % Write the corresponding centroids for blue
centBlueX = centroidBlue(object,1); centBlueY = centroidBlue(object,2);
vidIn = step(htextinsCent, vidIn, [centBlueX centBlueY], [centBlueX-6 centBlueY-9]);
end
for object = 1:1:length(bboxBlack(:,1)) % Write the corresponding centroids for black
centXBlack = centroidBlack(object,1); centYBlack = centroidBlack(object,2);

end
for object = 1:1:length( bboxYellow(:,1)) % Write the corresponding centroids for
yellow
centXYellow = centroidYellow(object,1); centYYellow = centroidYellow(object,2);
vidIn = step(htextinsCent, vidIn, [centXYellow centYYellow], [centXYellow-6
centYYellow-9]);
end
```

```
%%%%%%%% Polling the direction of robot %%%%%%%%

If
((uint8(length(bboxRed(:,1)))~=0)&&((((0<centRedX)&&(centRedX<200)&&(centRedY>26
0))...
        ||((440<centRedX)&&(centRedX<640)&&(centRedY>260))))

        b=0;
        e=0;
        c=c+1;
        d=1;


elseif (d==1)
    f=f+1;

else
     b=1;
     e=1;

end

%%%% Condition to count the passing red object %%%%
if ((f>(2*c))&&(d==1))
    fwrite(s,'r','uchar');
    c=0;
    f=0;
    d=0;

end
```

```
%%%%%%%% Green %%%%%%%%

If
((uint8(length(bboxGreen(:,1)))~=0)&&(((0<centGreenX)&&(centGreenX<250)&&(centGre
enY>260))...
        ||((390<centGreenX)&&(centGreenX<640)&&(centGreenY>260))))
        b=0;
        e=0;
        cc=cc+1;
        dd=1;


elseif (dd==1)
    ff=ff+1;

else
     b=1;
     e=1;

end

if ((ff>(2*cc))&&(dd==1))
    fwrite(s,'y','uchar');
    cc=0;
    ff=0;
    dd=0;

end

%%%%%%%%%% Blue %%%%%%%%%%

if ((uint8(length(bboxBlue(:,1)))~=0)&&(e==1))

    if ((310<centBlueX)&&(centBlueX<330)&&(centBlueY>240)) % Go ahead
        fwrite(s,'FF','uchar');
       b=0;
```

```
    elseif ((330<centBlueX)&&(centBlueY>240)) % Go right
        fwrite(s,'RR','uchar');
        b=0;
    elseif ((310>centBlueX)&&(centBlueY>240)) % Go left
        fwrite(s,'LL','uchar');
        b=0;
    end
else
    b=1;

end

if ((b==1)&&(mod(nFrame,10)==0)&&(e==1))


if ((uint16(sum(sum(a1)))<10000)||(uint16(sum(sum(a5)))>20000))% Go left
        fwrite(s,'L','uchar');

elseif
((uint16(sum(sum(a1)))>20000)||((uint16(sum(sum(a1)))<10000)&&(uint16(sum(sum(a2)))
<3000))||((uint16(sum(sum(a1)))>10000)&&(uint16(sum(sum(a3)))>10000))...

||((uint16(sum(sum(a2)))<10000)&&(uint16(sum(sum(a3)))>15000))||((uint16(sum(sum(a1)
))<10000)&&(uint16(sum(sum(a2)))<10000)&&(uint16(sum(sum(a3)))<10000)...
       &&(uint16(sum(sum(a4)))<10000)&&(uint16(sum(sum(a5)))>5000)))% Go right
        fwrite(s,'R','uchar');

elseif((uint16(sum(sum(a1)))<10000)&&(uint16(sum(sum(a2)))<10000)&&(uint16(sum(su
m(a3)))<10000)&&(uint16(sum(sum(a4)))<10000))% Go ahead
        fwrite(s,'F','uchar');
else
        fwrite(s,'F','uchar');
end

end
```

```
%%%%%%%%%%% Count the number of blobs %%%%%%%%%%%
vidIn = step(htextinsBlue, vidIn, uint8(length(bboxBlue(:,1))));
vidIn = step(htextInsSum1, vidIn, uint16(sum(sum(a1))));
vidIn = step(htextinsSum2, vidIn, uint16(sum(sum(a2))));
vidIn = step(htextinsSum3, vidIn, uint16(sum(sum(a3))));
vidIn = step(htextinsSum4, vidIn, uint16(sum(sum(a4))));

step(hVideoIn, vidIn); % Output video stream
if nFrame==50
    b=1;
end

nFrame = nFrame+1;
if uint8(length(bboxBlack(:,1)))==0;
    out=out+1;
else
    out=0;
end
end
fwrite(s,'0','uchar');
fwrite(s,'0','uchar');
fclose(s);
delete(s);
clear s;

%% Clearing Memory
release(hVideoIn); % Release all memory and buffer used
release(vidDevice);
clear all;
clc;
```

ข.2 โค้ดคำสั่งการควบคุมมอเตอร์ด้วย ADUINO (MEGA)

```
/*
 /* Brushed_H_Bridge_simple2 sketch
 * commands from serial port control motor direction
 * F,R or L set the direction, any other key stops the motors
 */
//  H-Bridge input pins:
#include <Keypad.h>
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Four columns
// Define the Keymap
char keys[ROWS][COLS] = {
                                {'1','2','3','A'},
                                {'4','5','6','B'},
                                {'7','8','9','C'},
                                {'*','0','#','D'}
                              };
byte rowPins[ROWS] = {13, 12, 11, 10};
byte colPins[COLS] = {9, 8, 7, 2};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
const int in1Pin = 6;  // digital pin 6 has a in1Pin attached to it. Give it a name
const int in2Pin = 5;  // digital pin 5 has a in2Pin attached to it. Give it a name
// H-Bridge pins for second motor
const int in3Pin = 4;  // digital pin 4 has a in3Pin attached to it. Give it a name
const int in4Pin = 3;; // digital pin 3 has a in4Pin attached to it. Give it a name
const int VPin1 = A0;
const int VPin2 = A1;
const int pingPin1 = A3;
     int inPin1 = A2;
const int pingPin2 = A4;
     int inPin2 = A5;
int val = 0;
int B=0 ;
static int A=0 ;
static int C=0 ;
static int D=1;
static int r=1 ;
```

```
static int gb=0 ;
static int t=0 ;
statlc Int tt=0 ;
static int count=0;
static int red=0;
static int f=1;
static int rou=0;
char k;
static int key;
int x1 = 0;
int x2 = 0;
int i=0;
int sw;
static int swc=1;
void setup()
{
        Serial.begin(9600);// initialize serial communication at 9600 bit/s
        pinMode(in1Pin, OUTPUT); // initialize digital pin 6 as an output.
        pinMode(in2Pin, OUTPUT); // initialize digital pin 5 as an output.
        pinMode(in3Pin, OUTPUT); // initialize digital pin 4 as an output.
        pinMode(in4Pin, OUTPUT); // initialize digital pin 3 as an output.
        pinMode(VPin1, OUTPUT);
        pinMode(VPin2, OUTPUT);
        //Switch
        pinMode(52, INPUT);
        pinMode(53, OUTPUT);
        digitalWrite(53,HIGH);
        digitalWrite(VPin1,HIGH);
        digitalWrite(VPin2,HIGH);
}
unsigned long Sonic1()
   {
        long duration1, cm1;
        pinMode(pingPin1, OUTPUT);
        digitalWrite(pingPin1, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin1, HIGH);
```

```
        delayMicroseconds(5);
         digitalWrite(pingPin1, LOW);
         pinMode(InPin1, INPUT);
         duration1 = pulseIn(inPin1, HIGH);
         cm1 = duration1/ 29 / 2;
         return cm1;
    }
unsigned long Sonic2()
    {
        long duration2, cm2;
        pinMode(pingPin2, OUTPUT);
        digitalWrite(pingPin2, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin2, HIGH);
        delayMicroseconds(5);
        digitalWrite(pingPin2, LOW);
        pinMode(inPin2, INPUT);
        duration2 = pulseIn(inPin2, HIGH);
        cm2 = duration2/ 29 / 2;
        return cm2;
    }
void loop()
{
 if(swc==1)
 {
    sw=digitalRead(52);

 }
    k = keypad.getKey();

 if (k)
 {
   key=k-'0';
   B=1;
 }
 if (key==2)
 {
```

```
  C=1;
}

if((B==1)&&(sw==0))
{
  swc=0;
  if ( Serial.available())// check if data has been sent from the computer
    {

     char ch = Serial.read(); // read the most recent byte
      i++;
      if(i==3)
      {
        x1 = Sonic1();
        x2 = Sonic2();
        if (x1<55)
       {
         //first motor
         analogWrite(in1Pin,0);
         analogWrite(in2Pin,255);
           //second motor
         analogWrite(in3Pin,0);
         analogWrite(in4Pin,0);
         D=0;
       }
       if(x2<55)
       {
         // first motor
         analogWrite(in1Pin,0);
         analogWrite(in2Pin,0);
           //second motor
         analogWrite(in3Pin,0);
         analogWrite(in4Pin,255);
         D=0;
       }
       if((x1>60)&&(x2>60))
       {
```

```
    D=1;
  }
  i=0;
}
if((D==1)||(ch=='F')||(ch=='FF')||(ch=='r')||(ch=='0'))
{
  if ((ch == 'F')||(ch == 'FF')||(ch == 'FFF'))// give the motor turn forward
   {
            Serial.println("CW");// print out the word that is “CW”
   // first motor
            analogWrite(in1Pin,0);
            analogWrite(in2Pin,200);
  //second motor
            analogWrite(in3Pin,0);
            analogWrite(in4Pin,165);
            r=1;
   }

  else if ((ch == 'R')||(ch == 'RR')||(ch == 'RRR'))//give the motor turn right
    {
          Serial.println("Right");// print out the word that is “Right”
    // first motor
          analogWrite(in1Pin,0);
          analogWrite(in2Pin,255);
    //second motor
          analogWrite(in3Pin,0);
          analogWrite(in4Pin,0);
          r=1;
     }

  else if ((ch == 'L')||(ch == 'LL')||(ch == 'LLL'))//give the motor turn left
     {
          Serial.println("Left");// print out the word that is “Left”
    // first motor
          analogWrite(in1Pin,0);
           analogWrite(in2Pin,0);
    //second motor
```

```
            analogWrite(in3Pin,0);
            analogWrite(in4Pin,225);
            r=1;
        }

    else if (ch=='0')
      {
            Serial.print("Stop motors");
            digitalWrite(in1Pin,LOW);
            digitalWrite(in2Pin,LOW);
            digitalWrite(in3Pin,LOW);
            digitalWrite(in4Pin,LOW);
            B = 0;
      }
if (ch=='r')
{
   red++;
}
if (red==1)
{
  rou++;
  Serial.end();
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,LOW);
  digitalWrite(in3Pin,LOW);
  digitalWrite(in4Pin,LOW);
  delay(3000);
  red=0;
   if (rou==key)
  {
    digitalWrite(in1Pin,LOW);
    digitalWrite(in2Pin,LOW);
    digitalWrite(in3Pin,LOW);
    digitalWrite(in4Pin,LOW);
    while(digitalRead(52)==0);
    delay(3000);
    t=1;
```

```
          B=0;
        }
        Serial.begin(9600);
      }
    if(ch=='y')
    {
      Serial.end();
      digitalWrite(in1Pin,LOW);
      digitalWrite(in2Pin,LOW);
      digitalWrite(in3Pin,LOW);
      digitalWrite(in4Pin,LOW);
      delay(3000);
      tt=1;
      B=0;
      Serial.begin(9600);
    }
  }
 }
}
if (t==1)
{
  Serial.end();
  analogWrite(in1Pin,255);
  analogWrite(in2Pin,0);

  analogWrite(in3Pin,0);
  analogWrite(in4Pin,255);
  delay(4300);
  t=0;
  B=1;
  Serial.begin(9600);
}
if (tt==1)
{
  Serial.end();
  analogWrite(in1Pin,255);
  analogWrite(in2Pin,0);
```

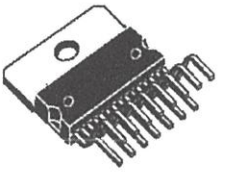
```
  analogWrite(in3Pin,0);
  analogWrite(in4Pin,255);
  delay(4000);
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,LOW);
  digitalWrite(in3Pin,LOW);
  digitalWrite(in4Pin,LOW);
  tt=0;
  B=0;
  swc=1;
  Serial.begin(9600);
 }
}

void pwmSetup(){//just run once at setup
  TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM20); //phase correct pwm
31250hz
  TCCR2B = _BV(CS20);//change this as datasheet says to mainly get different
pwm frequencies
  OCR2A = 0;
  OCR2B = 0;
 }
long microsecondsToCentimeters(long microseconds)
   {
     // The speed of sound is 340 m/s or 29 microseconds per centimeter.
     // The ping travels out and back, so to find the distance of the
     // object we take half of the distance travelled.
     return microseconds / 29 / 2;
   }
```

ภาคผนวก ค
Datasheet

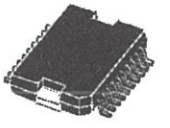
L298

DUAL FULL-BRIDGE DRIVER

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



Multiwatt15



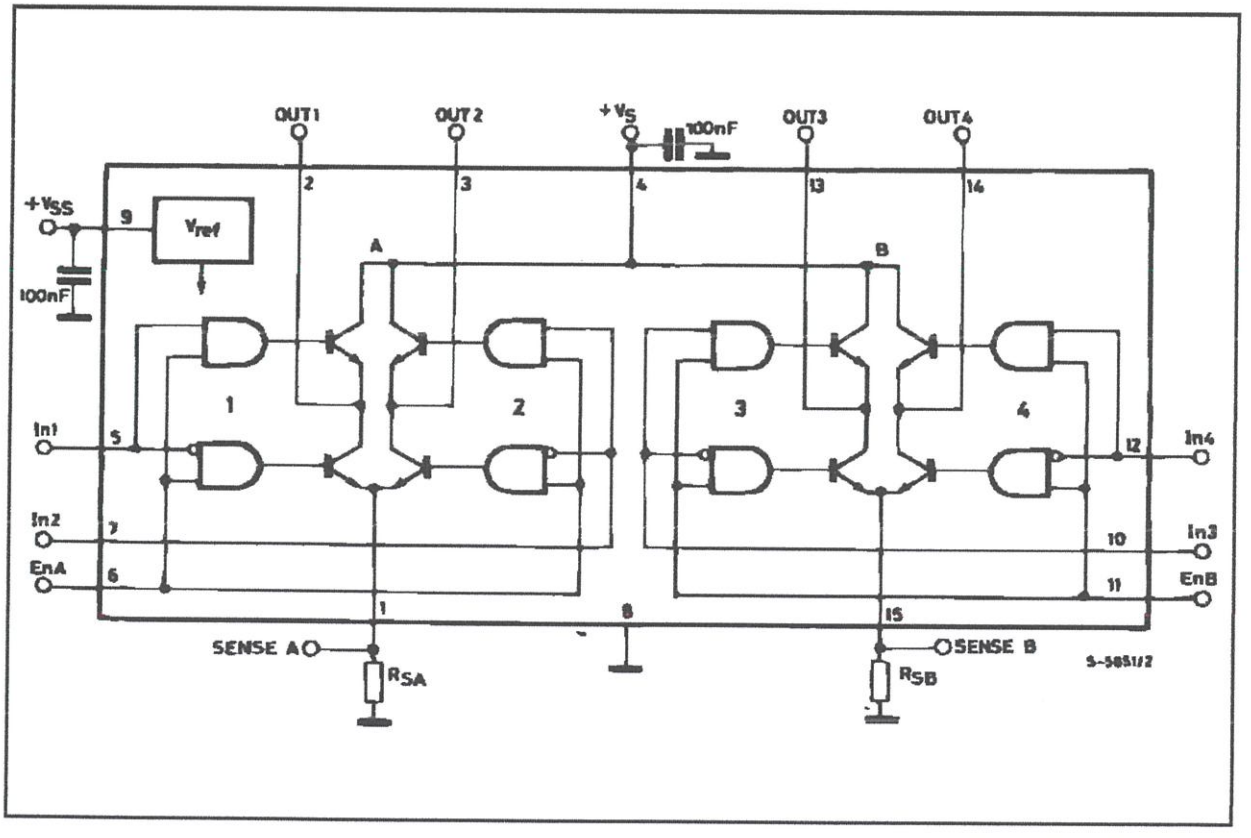
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

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

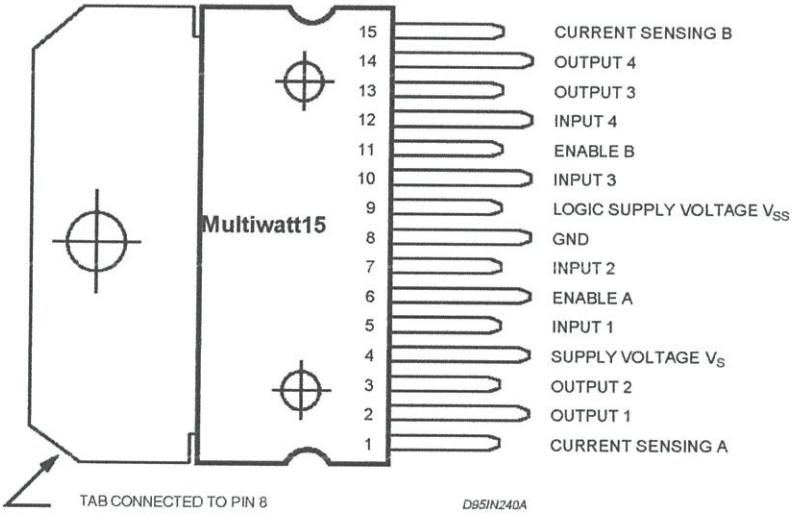
BLOCK DIAGRAM

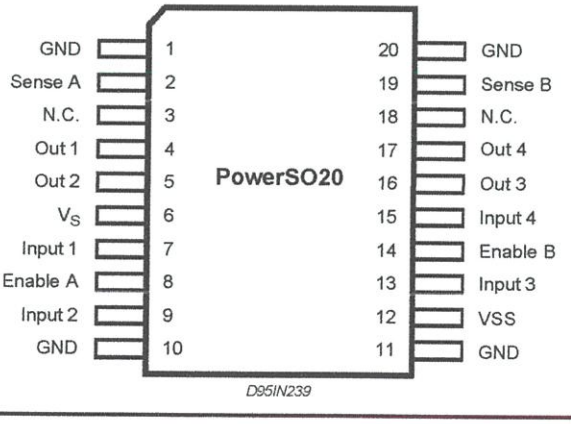


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate