

ระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ  
SELF – DRIVING CAR MODEL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

ระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ

SELF – DRIVING CAR MODEL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ

SELF – DRIVING CAR MODEL



โดย  
นายทิวต์ นันต์จารุงค์ 56010488  
นางสาวมนชนก สุขพ้อคำ 56010966

อาจารย์ที่ปรึกษา  
ผศ.ดร.กฤษณ์ วงจรูจีระ

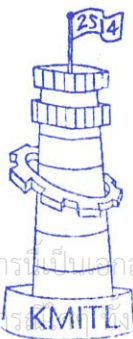
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

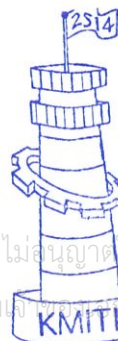
ปีการศึกษา 2559



ผ่านการตรวจรูปเล่มแล้ว

(.....)  
อาจารย์ที่ปรึกษา

19/5/60



ผ่านการตรวจชิ้นงานแล้ว

(.....)  
กรรมการผู้ตรวจชิ้นงาน

19/5/60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าในรูปแบบใด ๆ วิศวกรรมโทรคมนาคม Telecommunications Engineering วิศวกรรมโทรคมนาคม Telecommunications Engineering

ปริญญาโทปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ

SELF – DRIVING CAR MODEL

ผู้จัดทำ

- |                |              |          |
|----------------|--------------|----------|
| 1. นายทิวัดธ์  | นันต์จารุงศ์ | 56010488 |
| 2. นางสาวมนชนก | สุขพ้อคำ     | 56010966 |

(ผศ.ดร.กฤษณ์ วรจรัส)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ สามารถสำเร็จลุล่วงได้ด้วยดี เป็นเพราะได้รับความร่วมมือ ช่วยเหลือ และคำแนะนำที่เป็นประโยชน์ต่อการศึกษาค้นคว้าอย่างยิ่ง จากผู้มีพระคุณหลายท่าน อาทิ

คุณพ่อ คุณแม่ และญาติๆ ที่ให้การสนับสนุนทางด้านทุนทรัพย์ทางการศึกษา คอยให้กำลังใจ ให้ความช่วยเหลือและคอยเป็นแรงผลักดันในการทำงานมาโดยตลอด

ผศ.ดร. กฤษณ์ วรจิริระ ที่ให้ความเชื่อมั่น คำแนะนำแนวทางและวิธีแก้ไขปัญหาที่มีประโยชน์ต่อการพัฒนาปริญญานิพนธ์เล่มนี้

อาจารย์ทุกท่านในคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ช่วยสั่งสอน ให้คำแนะนำในการเรียนมาตลอดเป็นเวลา 4 ปี

เพื่อนๆ ทุกคน ที่คอยให้กำลังใจกัน และช่วยเหลือเกื้อกูลกัน ไม่ว่าจะยามทุกข์หรือว่ายามสุข

สุดท้าย ขอกราบขอบคุณอาจารย์ทุกท่าน ขอขอบคุณเพื่อนๆ ทุกคน และโดยเฉพาะอย่างยิ่ง ขอขอบคุณพ่อ แม่ ญาติพี่น้อง ที่ให้โอกาสให้พวกเรา สามารถเดินทางมาถึงจุดจุดนี้ได้ ขอกราบขอบพระคุณจริงๆ

นายทิวดี นันต์จารุงศ์

นางสาวমনชนก สุขพ่อคำ

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ  
SELF – DRIVING CAR MODEL

โดย นายทิวีร์ นันต์จารุงศ์ 56010488  
นางสาวมนชนก สุขพ้อคำ 56010966

อาจารย์ที่ปรึกษา ผศ.ดร. กฤษณ์ วงจรูระ

### บทคัดย่อ

ปริญญาานิพนธ์นี้ศึกษาเกี่ยวกับการออกแบบระบบของรถบังคับให้สามารถขับเคลื่อนแบบอัตโนมัติได้ โดยการดัดแปลงโครงสร้างของรถบังคับ ติดตั้งโมดูลกล้อง และควบคุมการขับเคลื่อนของรถบังคับ ด้วยการเขียนโปรแกรมประมวลผลภาพลงบนโปรแกรมโอเพนซีวีผ่านบอร์ดไมโครโปรเซสเซอร์ ให้สามารถขับเคลื่อนได้เฉพาะในเส้นทางที่มีการแบ่งเส้นถนนที่ชัดเจน ปฏิบัติตามเครื่องหมายหรือสัญญาณไฟจราจรที่ได้ทำการติดตั้งบนเส้นทางที่ได้จำลองไว้ และควบคุมความเร็วให้สัมพันธ์กับสภาพแวดล้อมได้ ในการทำปริญญาานิพนธ์ชิ้นนี้ จัดทำเพื่อเป็นความรู้ให้กับผู้จัดทำและบุคคลที่สนใจอยากศึกษา และยังหวังลึกๆว่าจะเป็นประโยชน์ในการพัฒนาต่อยอดให้กับอุตสาหกรรมยานยนต์ในประเทศไทยต่อไป

### ABSTRACT

Self-driving car model is a project that we aim to build a car with camera module and embedded controller to let the car to navigate and drive automatically. The Radio-controlled car (RC Car) model was modified to equip with Raspberry Pi board, camera module and ultrasonic sensor. The driving road environment was modeled by painting in curve line and straight line on the road. We also installed a traffic light on the junction of the road. In order to self-driving, an image processing was played as an important role to read the marking sign and process with the program called OpenCV. The result of image processing would control DC motor to vary the speed and fine turn the wheel to the correct direction. With the knowledge that we obtained along with this project, we would like to share and can hopefully benefit to others.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VIII
<b>บทที่ 1</b>	
<b>บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
<b>บทที่ 2</b>	
<b>ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	<b>3</b>
2.1 บอร์ดราสเบอร์รี่พาย	3
2.2 การทำงานของมอเตอร์กระแสตรง	7
2.3 การควบคุมหรือขับมอเตอร์กระแสตรงด้วยไอซี L293D	9
2.4 การทำงานของเซอร์โวมอเตอร์กระแสตรง	10
2.5 อัลตราโซนิก	14
2.6 เทรด	16
2.7 การประมวลผลภาพ	17
<b>บทที่ 3</b>	
<b>การออกแบบและการจัดทำปริญญานิพนธ์</b>	<b>27</b>
3.1 การออกแบบ	27
3.2 เครื่องมือที่ใช้ในการทดลอง	33
3.3 การจัดเก็บผลการทดลอง	35

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 4</b>	
<b>ผลการทดลอง</b>	<b>37</b>
4.1 ผลการทดลอง 1	37
4.2 ผลการทดลอง 2	40
4.3 ผลการทดลอง 3	41
4.4 ผลการทดลอง 4	43
4.5 ผลการทดลอง 5	46
4.6 ผลการทดลอง 6	51
<b>บทที่ 5</b>	
<b>สรุปผลและข้อเสนอแนะ</b>	<b>53</b>
5.1 สรุปผล	53
5.2 ข้อเสนอแนะ	53
<b>บรรณานุกรม</b>	<b>54</b>
<b>ภาคผนวก ก</b>	
<b>โค้ดหลัก</b>	<b>56</b>
<b>ภาคผนวก ข</b>	
<b>โค้ดส่วนวิเคราะห์และประมวลผล</b>	<b>61</b>
<b>ภาคผนวก ค</b>	
<b>โค้ดส่วนควบคุม</b>	<b>68</b>
<b>ภาคผนวก ง</b>	
<b>โค้ดส่วนควบคุมอุปกรณ์เพิ่มเติม</b>	<b>70</b>
<b>ภาคผนวก จ</b>	
<b>คุณลักษณะของอุปกรณ์</b>	<b>74</b>



## สารบัญรูป

รูปที่	หน้า
2.1	3
2.2	5
2.3	5
2.4	7
2.5	8
2.6	9
2.7	11
2.8	11
2.9	12
2.10	12
2.11	13
2.12	14
2.13	17
2.14	19
2.15	20
2.16	21
2.17	22
2.18	23
2.19	25
2.20	26
3.1	27
3.2	28
3.3	29
3.4	30
3.5	31

	หน้า
3.6 แผนภาพการวัดระยะห่างจากสิ่งกีดขวางด้านหน้า	32
3.7 รถบังคับ	33
3.8 บอร์ดราสเบอรี่พาย 3 โมเดลบี	33
3.9 เซอร์โวมอเตอร์ รุ่น SG90	34
3.10 ไอซีขับมอเตอร์ L293D	34
3.11 โมดูลกล้อง	34
3.12 โปรแกรมโอเพนซีวี	35
3.13 อัลตราโซนิกเซ็นเซอร์	35
4.1 สัญญาณพัลส์ที่มีความกว้างของพัลส์ 2.5 เพอร์เซ็นต์	37
4.2 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 2.5 เพอร์เซ็นต์ (-90 องศา)	38
4.3 สัญญาณพัลส์ที่มีความกว้างของพัลส์ 7.5 เพอร์เซ็นต์	38
4.4 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 7.5 เพอร์เซ็นต์ (0 องศา)	39
4.5 สัญญาณพัลส์ที่มีความกว้างของพัลส์ 12.5 เพอร์เซ็นต์	39
4.6 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 12.5 เพอร์เซ็นต์ (+90 องศา)	40
4.7 ผลการคำนวณหามุมและตำแหน่งของเส้นถนนทั้งซ้ายและขวาเมื่อถนนเป็นเส้นตรง	41
4.8 การตรวจจับเลนถนนที่เป็นเส้นตรง	42
4.9 ผลการคำนวณหามุมและตำแหน่งของเส้นถนนทั้งซ้ายและขวาเมื่อเป็นถนนเส้นโค้ง	42
4.10 ผลการตรวจจับเลนถนนที่เลี้ยวโค้งไปทางขวา	43
4.11 โปรแกรมแสดงข้อความ “FORWARD”	43
4.12 โปรแกรมแสดงข้อความ “TURN RIGHT”	44
4.13 โปรแกรมแสดงข้อความ “FORWARD” และ “OK”	44
4.14 โปรแกรมแสดงข้อความ “FORWARD” และ “LEFT”	45
4.15 โปรแกรมแสดงข้อความ “FORWARD” และ “RIGHT”	45
4.16 ผลการตรวจจับเส้นขอบถนนเมื่อมีสิ่งกีดขวาง	46
4.17 ลายวงจรสัญญาณไฟจราจร	46
4.18 สัญญาณไฟจราจรจำลองที่เสร็จแล้ว	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
4.19 การตรวจจับเมื่อเป็นไฟแดง	47
4.20 การตรวจจับเมื่อเป็นไฟเหลือง	48
4.21 การตรวจจับเมื่อเป็นไฟเขียว	48
4.22 ระยะห่าง 14.00 เซนติเมตรจากโคมดูกล้องถึงสัญญาณไฟจราจร	49
4.23 ระยะห่าง 16.00 เซนติเมตรจากโคมดูกล้องถึงสัญญาณไฟจราจร	49
4.24 ระยะห่าง 18.00 เซนติเมตรจากโคมดูกล้องถึงสัญญาณไฟจราจร	49
4.25 ระยะห่าง 20.00 เซนติเมตรจากโคมดูกล้องถึงสัญญาณไฟจราจร	50
4.26 กราฟความสัมพันธ์ระหว่างระยะทางอ้างอิงกับเวลาที่วัดได้	51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่		หน้า
4.1	ค่าความกว้างของพัลส์ที่ป้อนในโปรแกรมเทียบกับองศาของเซอร์โวมอเตอร์	37
4.2	ความสัมพันธ์ระหว่างความกว้างของพัลส์ที่ป้อนเข้าไอซีและความเร็วของมอเตอร์กระแสตรง	40
4.3	ผลการคำนวณระยะห่างระหว่างโมดูลคล็องกับสัญญาณไฟจราจรที่ระยะห่างตั้งแต่ระยะห่างตั้งแต่ 14 CM ถึง 20 CM	50
4.4	ความสัมพันธ์ระหว่างระยะทางอ้างอิง (CM) กับเวลาที่วัดได้ (US) ที่ระยะห่างต่างตั้งแต่ 5 CM ถึง 25 CM	51
4.5	ความสัมพันธ์ระหว่างระยะทางอ้างอิง (CM) กับระยะทางที่วัดได้ (CM) ที่ระยะห่างตั้งแต่ 5 CM ถึง 25 CM	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันรถยนต์เข้ามามีบทบาทในชีวิตประจำวันมากยิ่งขึ้น จึงหลีกเลี่ยงไม่ได้ที่จะเกิดปัญหาจากการจราจรติดขัดรวมถึงอุบัติเหตุที่เกิดขึ้นบนท้องถนน ซึ่งมีจำนวนเพิ่มมากขึ้นทุกวันจากความประมาทของผู้ขับขี่รถยนต์ เช่น เมาแล้วขับ ง่วง หรือแม้กระทั่งโรคประจำตัวกำเริบในขณะที่ขับขี่ ประกอบกับเทคโนโลยีในปัจจุบัน ที่มีความก้าวหน้าและล้ำสมัยมากขึ้น ถ้าหากมีการทำให้รถยนต์ธรรมดากลายเป็นรถยนต์อัจฉริยะ ที่สามารถขับเคลื่อนได้โดยไม่ต้องอาศัยคนขับ ก็อาจจะทำให้ปัญหาต่างๆ เหล่านี้ลดน้อยลงได้

และเนื่องจากรถยนต์ทั่วไปมีต้นทุนที่สูง ด้วยเหตุนี้ข้าพเจ้าจึงมีความสนใจ ที่จะพัฒนาระบบรถบังคับขับเคลื่อนอัตโนมัติ โดยใช้บอร์ดราสเบอร์รี่พายในการควบคุม ร่วมกับใช้หลักการการประมวลผลภาพเข้ามาช่วยในการตรวจจับเส้นทางและสัญญาณจราจรต่างๆ ที่ได้จำลองเอาไว้

### 1.2 วัตถุประสงค์

- 1) เพื่อศึกษาโครงสร้างของรถบังคับ และทำการดัดแปลงโครงสร้าง ให้เหมาะสมกับการนำมาทำเป็นระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ
- 2) เพื่อศึกษาหลักการทำงานของอิมเมจโปรเซสซิง (Image processing) และโปรแกรมโอเพนซีวี (OpenCV)
- 3) เพื่อศึกษาและเขียนซอฟต์แวร์การทำงานของบอร์ดราสเบอร์รี่พาย (Raspberry pi 3 model B)
- 4) เพื่อสามารถเป็นต้นแบบให้กับรถยนต์ขับเคลื่อนอัตโนมัติในประเทศไทย

### 1.3 ขอบเขตของปริญญาโท

1. สร้างระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ โดยการดัดแปลงรถบังคับวิทยุให้สามารถนำมาใช้งานกับบอร์ดราสเบอร์รี่พาย (Raspberry pi 3 model B) ได้
2. เขียนโปรแกรมประมวลผลภาพให้สามารถตรวจจับได้เฉพาะเส้นทางจำลองที่มีการแบ่งเส้นถนนที่ชัดเจน
3. เขียนโปรแกรมประมวลผลภาพให้สามารถตรวจจับเครื่องหมายหรือสัญญาณไฟจราจรที่จำลองไว้ได้
4. ควบคุมความเร็วรถบังคับให้สัมพันธ์กับสภาพแวดล้อมได้



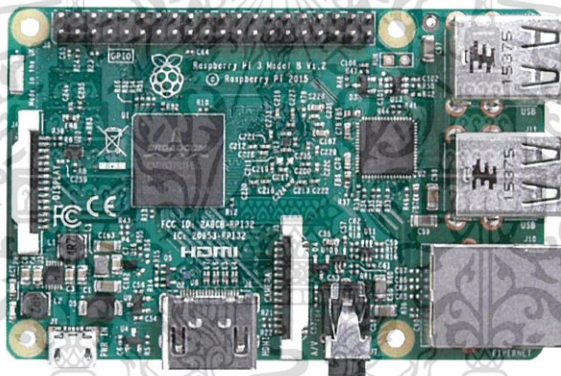
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 บอร์ดราสเบอร์รี่พาย (Raspberry Pi)

บอร์ดราสเบอร์รี่พาย คือบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ บอร์ดราสเบอร์รี่พายสามารถรองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arc Linux เป็นต้น ด้วยการติดตั้งลงบนการ์ดหน่วยความจำ Micro SD บอร์ดราสเบอร์รี่พายนี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน และยังมีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้



รูปที่ 2.1 บอร์ดราสเบอร์รี่พาย 3 โมเดลบี [1]

##### 2.1.1 คุณสมบัติทางเทคนิคของบอร์ด [1]

- มีระบบควบคุมหลักเป็น Broadcom BCM2837 ซึ่งรวมหน่วยประมวลผลกลาง หน่วยประมวลผลกราฟฟิก และหน่วยความจำ SDRAM ไว้ภายในตัวเดียวกัน
- มีหน่วยประมวลผลกลางหรือซีพียู (CPU) เป็นแบบ Quad-core ARM Cortex A53 (ใช้ชุดคำสั่งแบบ ARMv8) ขนาด 64 บิต ที่ใช้ความถี่ 1.2 กิกะเฮิร์ต
- มีหน่วยประมวลผลกราฟฟิกหรือจีพียู (GPU) เป็น Broadcom Video Core IV ที่ใช้ความถี่ 400 เมกะเฮิร์ต
- มีหน่วยความจำ SDRAM ขนาด 1 กิกะบิต

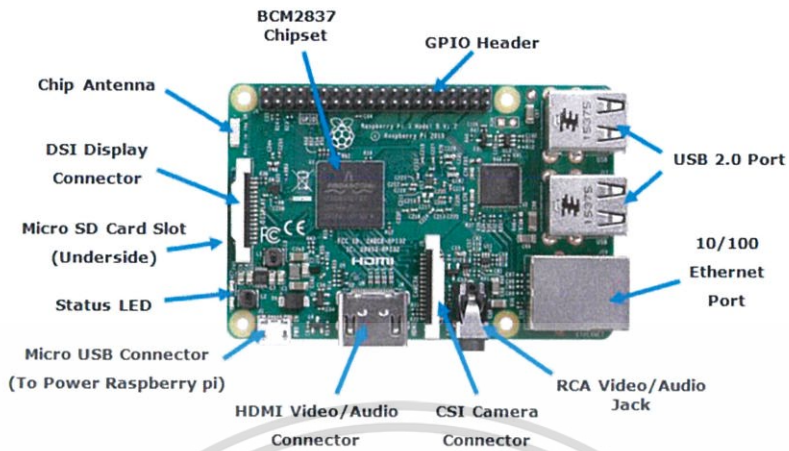
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีจุดเชื่อมต่อ: USB 2.0 (4 พอร์ต), ช่องเอาต์พุตของสัญญาณวิดีโอ, ช่องเอาต์พุตของสัญญาณเสียงขนาด 3.5 มิลลิเมตร, จุดเชื่อมต่อพอร์ต GPIO, พอร์ตอีเทอร์เน็ต (Ethernet) 10/100, ช่องสำหรับเสียบการ์ดหน่วยความจำ Micro SD
- มีชิปสำหรับเชื่อมต่อ Wi-Fi IEEE 802.11n และ Bluetooth 4.1 ที่รองรับ Bluetooth Low Energy
- ใช้ไฟเลี้ยงขนาด +5 โวลต์
- บอร์ดมีขนาด 86 x 56 x 21 มิลลิเมตร

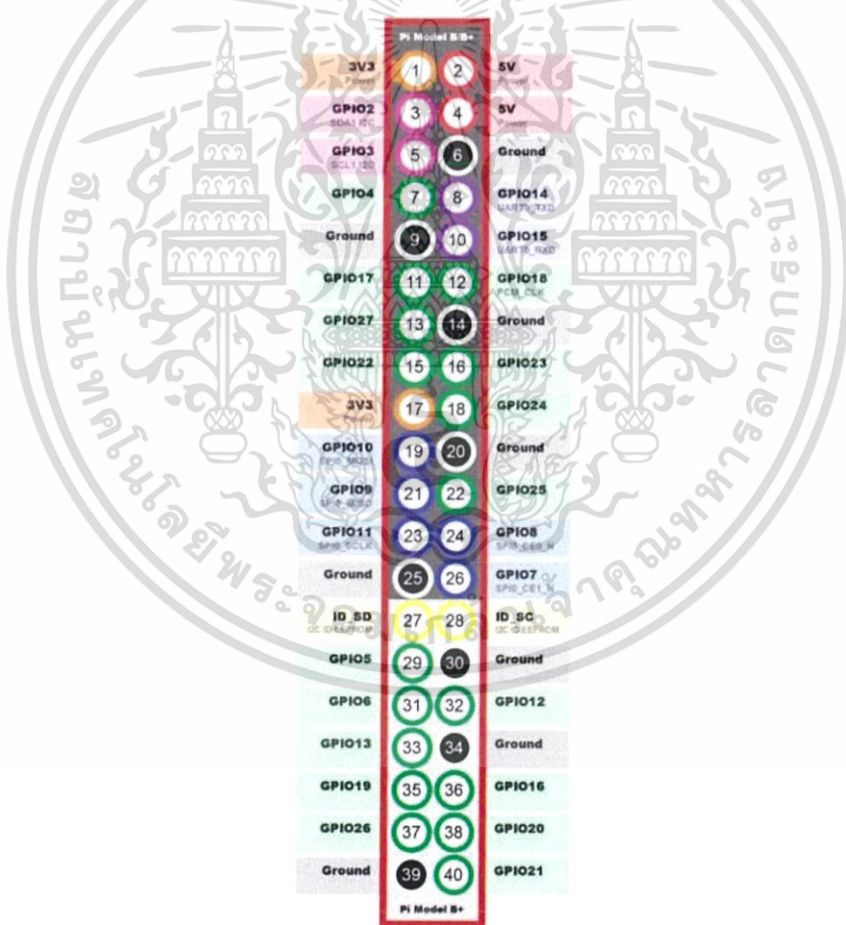
### 2.1.2 ส่วนประกอบของบอร์ด

- Chip Antenna เป็นสายอากาศสำหรับปล่อย Wi-Fi หรือ Bluetooth
- DSI Display Connector เป็นจุดต่อจอภาพแบบ DSI สำหรับเชื่อมต่อกับจอแสดงผลเพิ่มเติม
- Micro SD Card Slot เป็นช่องสำหรับใส่การ์ดหน่วยความจำ Micro SD
- Status LED เป็นจุดบอกสถานะไฟ LED ของบอร์ด
- Micro USB Connector เป็นช่องสำหรับจ่ายไฟ +5 โวลต์ ให้กับบอร์ดด้วย Micro USB
- HDMI Video/Audio Connector เป็นช่อง HDMI สำหรับเชื่อมต่อกับหน้าจอเพื่อแสดงผล
- CSI Camera Connector ใช้สำหรับเชื่อมต่อกับโมดูลกล้องแบบ CSI
- RCA Video/Audio Jack ใช้สำหรับเชื่อมต่อกับอุปกรณ์เสียง
- 10/100 Ethernet Port ใช้สำหรับเชื่อมต่อกับพอร์ต RJ-45 เพื่อเชื่อมต่อกับอินเทอร์เน็ต
- USB 2.0 Port ใช้สำหรับเชื่อมต่อกับพอร์ต USB มีทั้งหมด 4 พอร์ต
- Broadcom BCM2873 Chipset
- GPIO Header (ย่อมาจาก General Purpose Input/Output) มีพิน (Pin) ทั้งหมด 40 พิน มีไว้ใช้สำหรับงานต่างๆ ไปโดยควบคุมผ่านซอฟต์แวร์ สามารถใช้งานได้หลายรูปแบบขึ้นอยู่กับผู้ใช้งาน โดยรูปที่ 2.3 เป็นรูปการใช้งานพินของ GPIO แต่ละพิน





รูปที่ 2.2 ส่วนประกอบของบอร์ดราสเบอร์รี่พาย 3 โมเดลบี [2]



รูปที่ 2.3 พอร์ต GPIO ของบอร์ดราสเบอร์รี่พาย 3 โมเดลบี [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 การใช้งานบอร์ดราสเบอร์รี่พาย [3]

- 1) ดาวน์โหลดไฟล์อิมเมจ Raspbian Jessie จาก [www.raspberrypi.org](http://www.raspberrypi.org) แล้วแตกไฟล์ดังกล่าวด้วยคำสั่ง unzip ไฟล์ที่ได้จะเป็นนามสกุล .img
- 2) ติดตั้งระบบปฏิบัติการ Raspbian Jessie ด้วยการเขียนข้อมูลจากไฟล์อิมเมจ (.img) ลงในการ์ด Micro SD ที่มีขนาดมากกว่า 8GB
- 3) เสียบการ์ดหน่วยความจำ Micro SD เข้าที่ช่องอ่านการ์ดของเครื่องคอมพิวเตอร์
- 4) ใช้คำสั่งเพื่อเขียนข้อมูลจากไฟล์อิมเมจที่แตกไว้ ลงในการ์ดหน่วยความจำ
- 5) เมื่อลงโปรแกรมเสร็จแล้ว ให้นำการ์ดหน่วยความจำ Micro SD เสียบเข้ากับบอร์ดราสเบอร์รี่พาย แล้วต่อกับอุปกรณ์ต่างๆ เช่น เม้าส์ คีย์บอร์ด จอมอนิเตอร์
- 6) เมื่อเริ่มใช้งานบอร์ดราสเบอร์รี่พาย ระบบจะเข้าสู่หน้าจอหลัก (Desktop) ให้อัตโนมัติ

### 2.1.4 การเชื่อมต่อระบบเครือข่ายอินเทอร์เน็ตกับบอร์ด

เมื่อต้องการใช้งานระบบที่เกี่ยวข้องกับเครือข่ายอินเทอร์เน็ต บอร์ดราสเบอร์รี่พาย 3 โมเดลปีสามารถต่อกับเครือข่ายอินเทอร์เน็ตได้ 2 ทาง

- 1) ผ่านทางสาย LAN RJ-45 โดยการต่อสาย LAN ชนิดดังกล่าว เข้ากับพอร์ตLANของบอร์ดราสเบอร์รี่พายได้เลย ซึ่งจะสามารถใช้งานเครือข่ายอินเทอร์เน็ตได้ทันที
- 2) ผ่านทาง Wi-Fi โดยการทำการเลือกเครือข่ายผ่านทางหน้าจอของบอร์ดราสเบอร์รี่พาย เพื่อทำการเชื่อมต่อเข้าสู่อินเทอร์เน็ต

## 2.2 การทำงานของมอเตอร์กระแสตรง

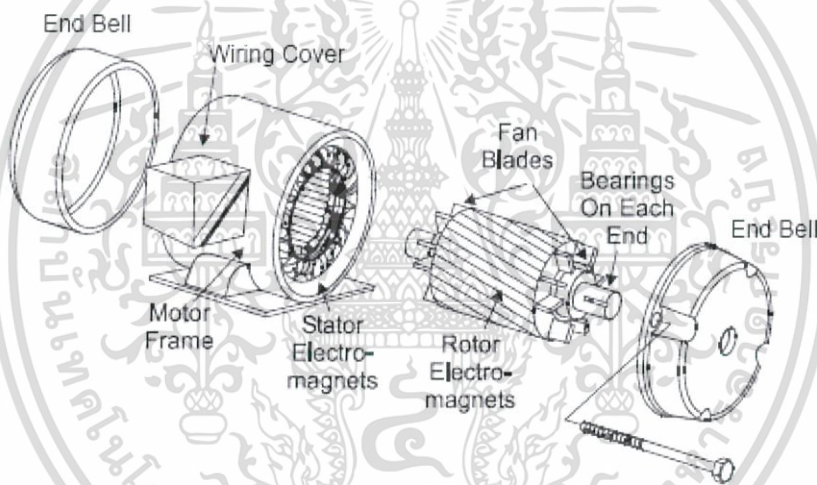
มอเตอร์กระแสตรง มีหน้าที่ในการขับเคลื่อนล้อคู่หลังของรถบังคับให้สามารถหมุนเพื่อเดินหน้าหรือถอยหลังได้

### 2.2.1 ส่วนประกอบภายในของมอเตอร์

มอเตอร์กระแสตรง[4]มีส่วนประกอบที่สำคัญ 2 ส่วน ดังนี้

1) สเตเตอร์ (Stator) คือส่วนที่อยู่กับที่ ซึ่งจะประกอบไปด้วยโครงของมอเตอร์ แกนเหล็กสเตเตอร์ และขดลวด

2) โรเตอร์ (Rotor) คือตัวหมุน ทำหน้าที่หมุนเพลลาเพื่อจ่ายพลังงานกล โดยโรเตอร์มักจะมีขดลวดพันอยู่โดยรอบ เมื่อมีกระแสไหลผ่านจะเกิดอำนาจแม่เหล็กที่จะไปทำปฏิกิริยากับสนามแม่เหล็กถาวรของสเตเตอร์ ซึ่งสามารถขับเพลลาให้หมุนได้



รูปที่ 2.4 ส่วนประกอบของมอเตอร์กระแสตรง [5]

### 2.2.2 หลักการทำงานของมอเตอร์กระแสตรง

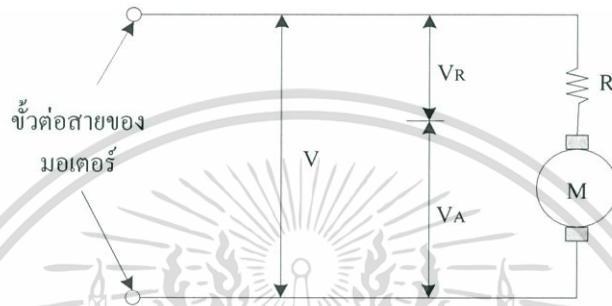
การทำงานของมอเตอร์กระแสตรง[6]เกิดจากการทำงานร่วมกันระหว่างสนามแม่เหล็กของแม่เหล็กในตัวมอเตอร์และสนามแม่เหล็กที่เกิดจากกระแสในขดลวด ทำให้เกิดแรงดูดและแรงผลักของสนามแม่เหล็กทั้งสอง

โดยเมื่อมีการผ่านกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็กจะทำให้เกิดแรงแม่เหล็กซึ่งมีสัดส่วนของแรงขึ้นกับกระแสแรงของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแสและสนามแม่เหล็ก ขณะที่ทิศทางของแรงกลับตรงกันข้ามกัน ถ้าหากกระแสของ

สนามแม่เหล็กไหลย้อนกลับจะทำให้เกิดการเปลี่ยนแปลงของกระแสและสนามแม่เหล็กซึ่งเป็นผลทำให้ทิศทางของแรงเปลี่ยนไป ด้วยคุณสมบัตินี้ทำให้มอเตอร์กระแสตรงกลับทิศทางการทำงานได้

### 2.2.3 คุณสมบัติของมอเตอร์กระแสตรง

เราจะพิจารณาจากแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย ซึ่งวงจรรภายในของมอเตอร์กระแสตรง[6] เป็นดังรูปที่ 2.5



รูปที่ 2.5 วงจรรภายในของมอเตอร์กระแสตรง [6]

โดยสมมติให้ทวนโรเตอร์ไม่มีความต้านทานอยู่เลย อนุกรมกับความต้านทานซึ่งในที่นี้ก็คือ ความต้านทานของขดลวดนั่นเอง แรงดันที่ขั้วต่อสายของมอเตอร์กระแสตรงคือผลบวกระหว่างแรงดันที่ทวนโรเตอร์ ( $V_A$ ) และแรงดันตกคร่อมความต้านทานขดลวด ( $V_R$ )

แรงดัน  $V_A$  ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำป้อนกลับ (BACK EMF) ซึ่งเกิดขึ้นในโรเตอร์ ขณะที่หมุนแรงดันที่เกิดขึ้นนี้เป็นไปตามกฎของการเหนี่ยวนำแม่เหล็กไฟฟ้าจากการเคลื่อนที่ของตัวนำในสนามแม่เหล็กสัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็ก และความเร็วในการเคลื่อนที่ของตัวนำ แรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์กระแสตรงและแปรผันตรงกับความเร็วในการหมุน ผลบวกของแรงดันที่ทวนโรเตอร์ ( $V_A$ ) และแรงดันตกคร่อมขดลวด ( $V_R$ ) ต้องเท่ากับแรงดันที่ป้อนให้กับมอเตอร์กระแสตรง ( $V$ )

$$V = V_A + V_R \quad (\text{โวลต์}) \quad (2.1)$$

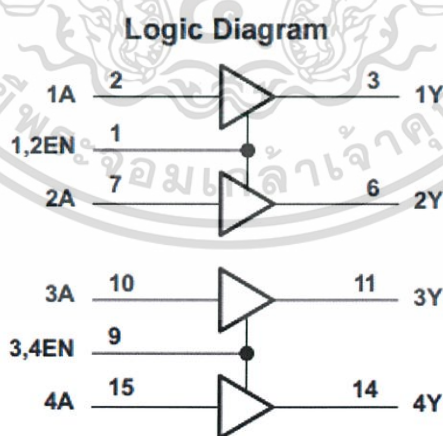
เมื่อพิจารณาตั้งแต่มอเตอร์หยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น  $V_A = 0$ ,  $V_R = V$  กระแสที่ไหลในมอเตอร์กระแสตรงหาได้จาก

$$I = \frac{V_R}{R} \quad (\text{แอมแปร์}) \quad (2.2)$$

เมื่อมอเตอร์กระแสตรงเริ่มหมุนก็จะเริ่มมีความเร็วเกิดขึ้น แรงดันที่ฟუნโรเตอร์จะเพิ่มขึ้นในลักษณะของกราฟเส้นตรงตามความเร็ว แรงดันตกคร่อมขดลวดซึ่งมีค่าเท่ากับความแตกต่างระหว่างแรงดันที่ฟุนโรเตอร์กับแรงดันที่ป้อนให้กับมอเตอร์กระแสตรงนั้นจะเริ่มลดลง กระแสที่ไหล (I) ก็จะมีลดลงด้วยเช่นกัน ขณะที่มอเตอร์กระแสตรงยังมีความเร็วอยู่นั้น ความเร็วจะเพิ่มมากขึ้น แต่แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์กระแสตรงสามารถรับภาระโหลดได้สมดุลพอดี และในขณะที่มอเตอร์กระแสตรงไม่มีโหลดและมีการหมุนอย่างอิสระอยู่นั้น จะมีเพียงค่าความถี่ของแบร์ริงและแรงต้านอากาศที่ทำให้แรงดันที่ฟุนโรเตอร์มีค่าเกือบเท่ากับค่าแรงดันที่ป้อนให้กับมอเตอร์กระแสตรง

### 2.3 การควบคุมหรือขับมอเตอร์กระแสตรงด้วยไอซี L293D

เนื่องมาจากการต่อวงจรมอเตอร์กระแสตรงเข้าโดยตรงกับบอร์ดราสเบอร์รี่พายนั้น จะทำให้บอร์ดราสเบอร์รี่พายได้รับกระแสไฟฟ้าที่ไม่เพียงพอซึ่งอาจจะส่งผลให้บอร์ดราสเบอร์รี่พายเกิดความเสียหายได้ จึงจำเป็นต้องมีวงจรขับมอเตอร์เข้ามาช่วยในการทำหน้าที่ขยายสัญญาณ โดยอุปกรณ์ที่เลือกใช้เป็นไอซี L293D ซึ่งมีความสามารถในการขับมอเตอร์ขนาดเล็กได้ คุณสมบัติคร่าวๆ ของไอซี L293D [7] คือสามารถรับแรงดันได้ตั้งแต่ 4.5 ถึง 36 โวลต์ ขับมอเตอร์กระแสตรงได้ถึง 2 ตัวพร้อมๆกัน จ่ายกระแสได้ 600 มิลลิแอมป์สำหรับการต่อมอเตอร์กระแสตรง 2 ตัว และ 1.2 มิลลิแอมป์สำหรับการต่อมอเตอร์กระแสตรง 1 ตัว อีกทั้งยังสามารถควบคุมทั้งความเร็วและทิศทางการหมุนของมอเตอร์กระแสตรงได้ โลจิกไดอะแกรมของไอซี L293D เป็นดังรูปที่ 2.6



รูปที่ 2.6 โลจิกไดอะแกรมของไอซี L293D [7]

ดังนั้น หากต้องการควบคุมมอเตอร์กระแสตรงให้ได้นั้น จะต้องทำการป้อนสัญญาณการมอดูเลตเชิงความกว้างพัลส์ (Pulse Width Modulation : PWM) จากบอร์ดราสเบอร์รี่พายให้กับไอซีตัวดังกล่าวเข้าที่ขาอินพุต แล้วต่อขาเอาต์พุตเข้ากับมอเตอร์กระแสตรง ซึ่งถ้าหากสัญญาณ PWM มี Duty cycle มาก มอเตอร์กระแสตรงจะมีความเร็วในการหมุนเพิ่มมากขึ้น แต่ถ้าหากสัญญาณ PWM มี Duty cycle น้อย มอเตอร์กระแสตรงจะมีความเร็วในการหมุนลดน้อยลง

## 2.4 การทำงานของเซอร์โวมอเตอร์กระแสตรง

ในการควบคุมทิศทางการเคลื่อนที่ของรถบังคับ เราจะนำเซอร์โวมอเตอร์กระแสตรง (DC Servo Motor) [8] มาใช้ในการควบคุม เซอร์โวมอเตอร์กระแสตรงเป็นมอเตอร์ชนิดหนึ่งที่เราสั่งงานหรือตั้งค่าเข้าไปแล้วนั้น จะทำให้ตัวมอเตอร์ที่อยู่ภายในหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง ด้วยการใช้การควบคุมแบบป้อนกลับ (Feedback Control)

การควบคุมแบบป้อนกลับ คือ ระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบแล้วนำมาเปรียบเทียบกับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบมีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต

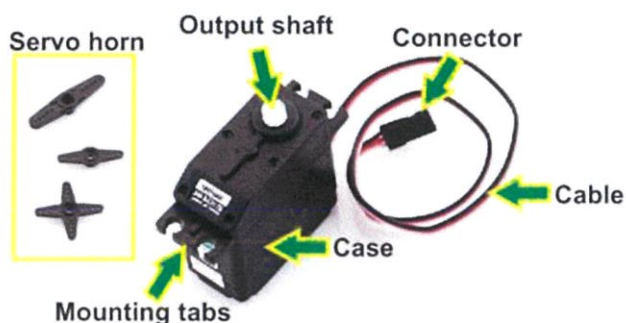
ในที่นี้ ขอเรียกเซอร์โวมอเตอร์กระแสตรงแบบสั้นๆว่า เซอร์โวมอเตอร์

### 2.4.1 ส่วนประกอบภายนอก

- 1) Case คือ ตัวถังหรือกรอบของเซอร์โวมอเตอร์
  - 2) Mounting Tab คือ ส่วนยึดจับระหว่างเซอร์โวมอเตอร์กับชิ้นงาน
  - 3) Output Shaft คือ เฟลาส่งกำลัง
  - 4) Servo Horns คือ ส่วนที่เชื่อมต่อกับเฟลาส่งกำลังเพื่อสร้างกลไก
  - 5) Cable คือ สายเชื่อมต่อเพื่อจ่ายไฟฟ้า และควบคุมเซอร์โวมอเตอร์
- ประกอบไปด้วยสายไฟ 3 เส้น แต่ละเส้นจะมีสีที่แตกต่างกันออกไป ดังนี้

- สายไฟสีแดง คือ ไฟเลี้ยง (4.8 - 6 โวลต์)
- สายสีน้ำตาล คือ กราวนด์
- สายสีเหลือง คือ สายส่งสัญญาณพัลส์ควบคุม (3 - 5 โวลต์)

- 6) Connector คือ จุดเชื่อมต่อสายไฟ



รูปที่ 2.7 ส่วนประกอบภายนอกของเซอร์โวมอเตอร์ [8]

#### 2.4.2 ส่วนประกอบภายใน

- 1) Motor หรือ ตัวมอเตอร์
- 2) Gear Train คือ ชุดเกียร์ทดแรง
- 3) Position sensor คือ เซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
- 4) Electronic control system คือ ส่วนที่ควบคุมและประมวลผล

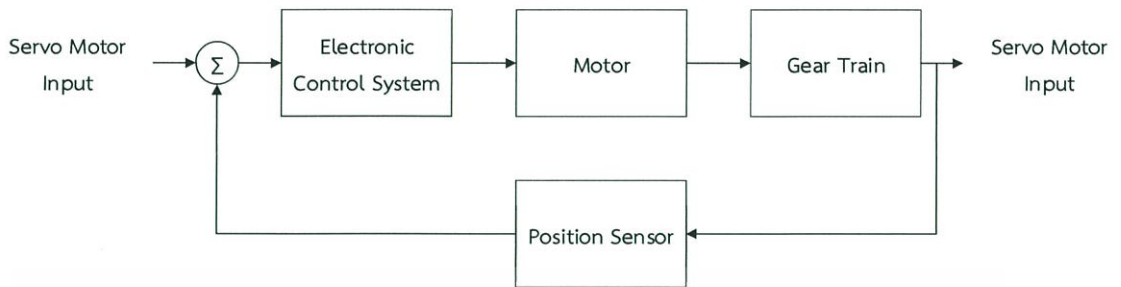
หมุน



รูปที่ 2.8 ส่วนประกอบภายในของเซอร์โวมอเตอร์ [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

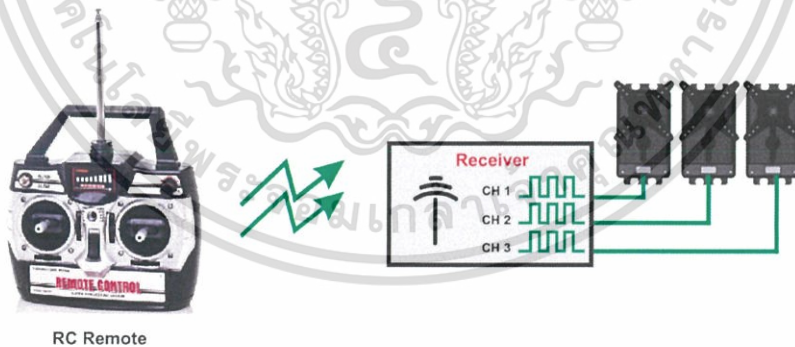
### 2.4.3 หลักการทำงานของเซอร์โวมอเตอร์



รูปที่ 2.9 บล็อกไดอะแกรมการทำงานของเซอร์โวมอเตอร์

การทำงานของเซอร์โวมอเตอร์จะถูกควบคุมโดยสัญญาณพัลส์ที่จ่ายเข้ามาทางสายสัญญาณสี่เส้น โดยที่สัญญาณพัลส์ที่ถูกส่งมานั้น จะเป็นสัญญาณการมอดูเลตเชิงความกว้างพัลส์ (Pulse Width Modulation : PWM) ที่มีความกว้างของพัลส์ที่แตกต่างกัน

เมื่อจ่ายสัญญาณพัลส์เข้ามาที่เซอร์โวมอเตอร์แล้ว วงจรควบคุม (Electronic control system) ภายในเซอร์โวมอเตอร์จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการ แล้วจากนั้นจะส่งคำสั่งไปทำการควบคุมให้ตัวมอเตอร์หมุนเคลื่อนที่ไปยังตำแหน่งนั้นๆ โดยมีเซนเซอร์ตรวจจับตำแหน่งที่คอยวัดค่ามุมของมอเตอร์ที่กำลังหมุน แล้วป้อนกลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องและแม่นยำ

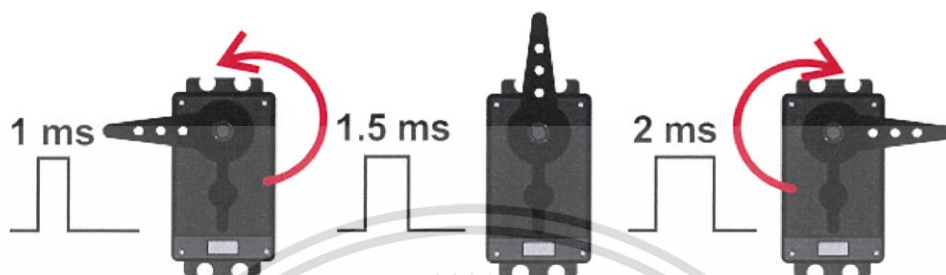


รูปที่ 2.10 การส่งสัญญาณจากรีโมทคอนโทรลไปยังเซอร์โวมอเตอร์ [8]

มุมหรือองศาจะขึ้นอยู่กับค่าความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากค่าความกว้างของสัญญาณพัลส์ที่ใช้ในเซอร์โวมอเตอร์จะอยู่ในช่วง 1 - 2 มิลลิวินาที หรือ 0.5 - 2.5 มิลลิวินาที



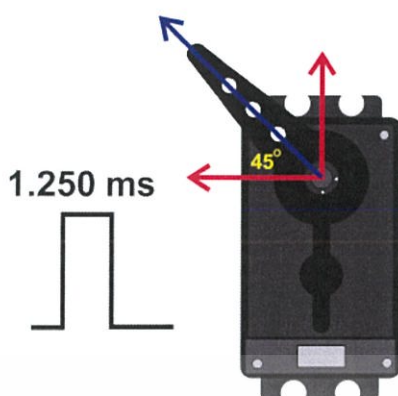
ยกตัวอย่างเช่น หากกำหนดค่าความกว้างของสัญญาณพัลส์ไว้ที่ 1 มิลลิวินาที เซอร์โวมอเตอร์จะหมุนไปทางซ้ายสุด ในทางกลับกันหากกำหนดค่าความกว้างของสัญญาณพัลส์ไว้ที่ 2 มิลลิวินาที เซอร์โวมอเตอร์จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดค่าความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 มิลลิวินาที เซอร์โวมอเตอร์ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกึ่งกลางพอดี



รูปที่ 2.11 มุมต่างๆ ของแขนเซอร์โวมอเตอร์เมื่อเปลี่ยนค่าความกว้างของพัลส์ [8]

ดังนั้นสามารถกำหนดต้องการหมุนของเซอร์โวมอเตอร์ได้โดยการเทียบค่า เช่น เซอร์โวมอเตอร์สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ค่าความกว้างของสัญญาณพัลส์เท่ากับ 1000 ไมโครวินาที ที่ 180 องศาใช้ค่าความกว้างของสัญญาณพัลส์เท่ากับ 2000 ไมโครวินาที เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ค่าความกว้างของสัญญาณพัลส์ต่างกัน  $\frac{(2000 - 1000)}{180}$  เท่ากับ 5.55 ไมโครวินาที

จากการหาค่าความกว้างของสัญญาณพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ เซอร์โวมอเตอร์หมุนไปที่มุม 45 องศาจะหาค่าความกว้างของสัญญาณพัลส์ที่ต้องการได้จาก  $5.55 \times 45$  เท่ากับ 249.75 ไมโครวินาที แต่ที่มุม 0 องศาเราเริ่มที่ค่าความกว้างของสัญญาณพัลส์เท่ากับ 1 มิลลิวินาที หรือ 1000 ไมโครวินาที เพราะฉะนั้นค่าความกว้างของสัญญาณพัลส์ที่ใช้ กำหนดให้เซอร์โวมอเตอร์หมุนไปที่ 45 องศา คือ  $1000 + 249.75$  เท่ากับประมาณ 1250 ไมโครวินาที



รูปที่ 2.12 ภาพด้านบนของเซอร์โวมอเตอร์เมื่อแขนทำมุม 45 องศา [8]

## 2.5 อัลตราโซนิก (Ultrasonic)

อัลตราโซนิก[9] หมายถึง คลื่นเสียงที่มีความถี่สูงเกินกว่าที่หูของมนุษย์จะสามารถได้ยินได้ โดยทั่วไปแล้วหูของมนุษย์โดยเฉลี่ยจะได้ยินเสียงสูงถึงเพียงแค่ประมาณ 15 กิโลเฮิร์ตซ์ ดังนั้นโดยปกติแล้ว คำว่าอัลตราโซนิกจึงมักหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20 กิโลเฮิร์ตซ์ขึ้นไป จะสูงขึ้นจนถึงเท่าใดไม่ได้ระบุจำกัดเอาไว้ สาเหตุที่นำเอาคลื่นย่านอัลตราโซนิกมาใช้เพราะว่าเป็นคลื่นที่มีทิศทาง ทำให้เราสามารถเล็งคลื่นเสียงไปยังเป้าหมายที่ต้องการได้โดยเจาะจง

อัลตราโซนิกเซ็นเซอร์ ทำงานโดยใช้หลักการสะท้อนของคลื่นเสียงที่มีความถี่อยู่ในช่วงอัลตราโซนิก โดยมีปัจจัยที่มีผลต่อการตรวจจับ[10]อยู่ด้วยกัน 4 ปัจจัยคือ

### 2.5.1 ความยาวคลื่นและการแพร่กระจายคลื่น (Wavelength and Radiation)

ความเร็วในการเดินทางของคลื่นนั้นขึ้นอยู่กับความถี่และความยาวคลื่น โดยคลื่นแม่เหล็กไฟฟ้านั้นจะมีความเร็วอยู่ที่  $3 \times 10^8$  เมตรต่อวินาที แต่สำหรับคลื่นเสียงนั้น จะช้ากว่าของคลื่นแม่เหล็กไฟฟ้ามาก โดยอากาศจะมีความเร็วอยู่ที่ประมาณ 344 เมตรต่อวินาที ที่อุณหภูมิ 20 องศาเซลเซียส โดยที่อ้างอิงเรื่องอุณหภูมินั้นเนื่องจากว่าอุณหภูมิมีผลต่อการเดินทางของคลื่นเสียงด้วย โดยที่ความเร็วต่ำๆ นี้ความยาวคลื่นจะสั้น ซึ่งหมายความว่า จะมีความละเอียดในการวัดค่าระยะทางและการกำหนดทิศทางเพราะว่าที่ความละเอียดสูง สามารถวัดค่าได้แม่นยำมากยิ่งขึ้น

### 2.5.2 การสะท้อนของคลื่น (Reflection)

เป็นการพูดถึงการตรวจจับการมีหรือไม่มีของวัตถุที่ถูกคลื่นสะท้อนกลับมา สำหรับวัตถุที่เป็นของแข็ง เช่น เหล็ก ไม้ ปูน แก้ว ยาง หรือกระดาษ จะมีความสามารถในการสะท้อนคลื่น

100% ดังนั้น การตรวจจับวัตถุประเภทนี้จึงเป็นเรื่องง่ายสำหรับการใช้อัลตราโซนิกเซ็นเซอร์ในการตรวจจับ แต่สำหรับ ลำสิ ขนสัตว์ หรือใยแก้วนั้น ยากที่จะใช้เซ็นเซอร์ประเภทนี้ในการตรวจจับ เพราะว่ามี การดุดกกลืนคลื่น ซึ่งจะเกิดขึ้นอยู่บ่อยๆ กับการตรวจจับชิ้นงานที่มีพื้นที่ผิวขนาดใหญ่ และไม่เรียบ

### 2.5.3 ผลกระทบทางด้านอุณหภูมิ (Effects of Temperature)

ในการใช้งานตัวเซ็นเซอร์วัดระยะทางแบบอัลตราโซนิกเซ็นเซอร์ ควรจะต้องพิจารณาเรื่องผลกระทบทางด้านอุณหภูมิที่มีต่อการทำงานของเซ็นเซอร์ โดยอุณหภูมิที่เปลี่ยนแปลงไปจะมีผลทำให้ความเร็วในการเคลื่อนที่ของคลื่นเสียงเปลี่ยนแปลงไปด้วย ซึ่งสามารถอธิบายได้จากสมการด้านล่าง

$$c = 331.5 + 0.607t \quad (\text{เมตรต่อวินาที}) \quad (2.3)$$

โดยที่  $c$  คือ ความเร็วในการเคลื่อนที่ของเสียง

$t$  คือ ค่าอุณหภูมิ ณ ขณะนั้น ( $^{\circ}\text{C}$ )

ความเร็วในการเคลื่อนที่ของเสียงนั้นจะเปลี่ยนแปลงเมื่ออุณหภูมิมีการเปลี่ยนแปลง ซึ่งอัลตราโซนิกเซ็นเซอร์นั้นจะใช้หลักการการสะท้อนของคลื่นเสียงกับวัตถุ แล้วหาค่าเวลา ซึ่งถ้าความเร็วคลื่นเสียงไม่คงที่แล้ว อาจทำให้การวัดค่ามีความคลาดเคลื่อนได้

### 2.5.4 การลดทอนของคลื่น (Attenuation)

สามารถเกิดขึ้นได้ทุกเวลา ขึ้นอยู่กับสภาวะแวดล้อม รวมถึงชนิดและพื้นผิวของวัตถุ ซึ่งการส่งคลื่นอัลตราโซนิกแบบเส้นตรงไปในอากาศ จะมีการลดทอนที่เกิดจากระยะทางในการเคลื่อนที่ของคลื่นเสียง นอกจากนี้โครงสร้างและลักษณะของชิ้นงาน หรือวัตถุที่ต้องการตรวจจับก็มีผล เช่น วัสดุพื้นผิวโค้ง จะทำให้การสะท้อนกลับของคลื่นเป็นแบบกระจัดกระจายยากต่อการทำงานของตัวรับ หรือเกิดการดุดกกลืนคลื่นจากชิ้นงาน นอกจากนี้ความถี่ที่ใช้งานก็มีผลต่อการลดทอนเช่นกัน โดยความถี่สูงอาจทำให้เกิดการลดทอนได้มากกว่าความถี่ต่ำ

## 2.6 เธรด (Thread)

เธรด[11] คือหน่วยการทำงานย่อยที่อยู่ในโปรเซส (Process) มีการแบ่งทรัพยากรภายในโปรเซสนั้น ปกติโปรเซสที่มี 1 เธรดจะเรียกว่าซิงเกิลเธรด (Single Thread) ซึ่งมักพบในระบบปฏิบัติการสมัยก่อน แต่ถ้า 1 โปรเซสมีหลายเธรดจะเรียกว่ามัลติเธรด (Multithread)

เหตุผลที่ต้องมีเธรด คือเพื่อดึงเอาความสามารถของหน่วยประมวลผลกลางออกมาให้เกิดประโยชน์สูงสุด โดยเธรดแต่ละตัวของโปรเซสเดียวกันจะทำงานแตกต่างกัน แต่จะมีความเกี่ยวข้องกันบางอย่างและต้องทำงานอยู่ภายใต้สภาพแวดล้อมเดียวกัน

### 2.6.1 องค์ประกอบภายในเธรด

- Threads ID คือหมายเลขเธรดที่อยู่ในโปรเซส
- Counter คือตัวนับเพื่อติดตามคำสั่งที่ถูกดำเนินการเป็นลำดับถัดไป
- Register คือหน่วยความจำเก็บค่าตัวแปรที่ทำงานอยู่ในปัจจุบัน
- Stack คือการเก็บประวัติการทำงาน

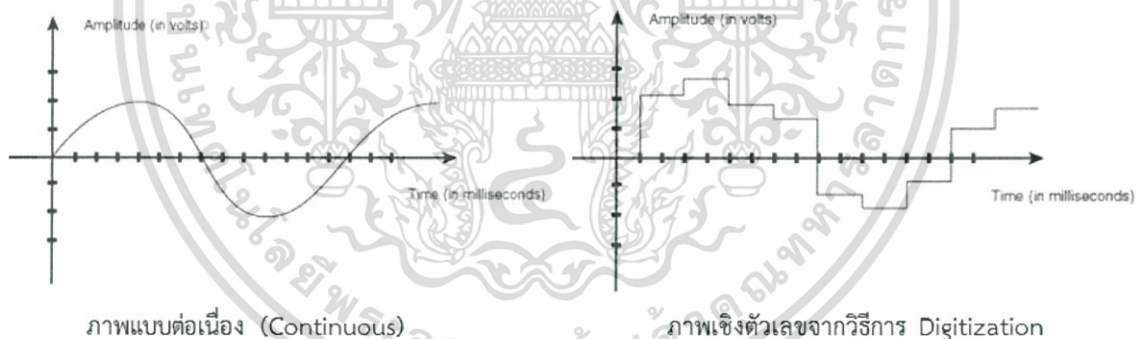
### 2.6.2 ประโยชน์ของมัลติเธรด

- 1) ในเรื่องการทำงานที่มีการตอบสนองกับผู้ใช้ มัลติเธรดช่วยลดและป้องกันการรอกการทำงานจนจบโปรเซส
- 2) ประหยัดการใช้หน่วยความจำในการทำงานของโปรเซส เนื่องจากแต่ละเธรดมีการใช้หน่วยความจำของโปรเซสร่วมกัน
- 3) ประสานจังหวะการทำงานและใช้ทรัพยากรของโปรเซสร่วมกันได้

## 2.7 การประมวลผลภาพ (Image Processing)

ดิจิทัลอิมเมจ (Digital Image) [12,13,14] คือภาพที่เก็บอยู่ในรูปแบบของดิจิทัล ภาพที่เรามองเห็นด้วยสายตาทั่วไปนั้นเป็นภาพในลักษณะสามมิติ คือ มีมิติของความกว้าง ความยาว และความลึกหรือความสูง ส่วนภาพถ่ายที่เห็นกันอยู่ในโทรทัศน์หรือเครื่องคอมพิวเตอร์นั้น เป็นการแปลงภาพจากสามมิติมาเป็นสองมิติ โดยการแปลงสัญญาณไฟฟ้าในรูปแบบแอนะล็อก ยกตัวอย่างเช่น ในกล้องวิดีโอ เช่นเซอร์ที่อยู่ในกล้องจะทำการสแกนหรือวัดผลรวมความเข้มแสงที่จุดต่างๆ ไปตามแนวสแกนที่เรียกว่า ราสเตอร์สแกน (Raster Scan) การสแกนแบบนี้จะมีทิศทางจากบนลงล่างและจากซ้ายไปขวา ภาพที่ได้จากการสแกนนั้น จะเป็นภาพแบบต่อเนื่อง (Continuous) ด้วยความเร็วทั่วไปที่ 24 ภาพต่อวินาที เช่นเดียวกันนี้ในเครื่องรับภาพวิดีโอก็จะมีภาพที่ได้มาจากเครื่องถ่ายภาพวิดีโอ และแสดงผลโดยเริ่มจากบนลงล่างและจากซ้ายไปขวาเช่นเดียวกัน

แต่ภาพที่ได้มาจากระบบแอนะล็อกนั้น ยังเป็นภาพแบบต่อเนื่องที่ยังไม่สามารถนำมาใช้ในการประมวลผลได้ ต้องมาทำการแปลงให้เป็นภาพเชิงตัวเลขเสียก่อนด้วยวิธีการดิจิไทเซชัน (Digitization) ซึ่งเป็นการแปลงฟังก์ชันต่อเนื่อง  $f(x, y)$  ให้เป็นฟังก์ชันไม่ต่อเนื่อง  $g(x, y)$  เพื่อนำมาประมวลผลด้วยคอมพิวเตอร์ได้



รูปที่ 2.13 ภาพระบบแอนะล็อกแบบต่อเนื่องและเชิงตัวเลขวิธีการดิจิไทเซชัน [12]

### 2.7.1 ความหมายของการประมวลผลภาพดิจิทัล

การประมวลผลภาพ หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใดๆ มากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพ ให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัด หรือการประหยัดพื้นที่ในการเก็บข้อมูล หรือใช้สำหรับการประมวลผลระดับสูง เช่น การจดจำรูปร่างลักษณะให้ได้อย่างแม่นยำ โดยทั่วไปแล้ววัตถุประสงค์ของการประมวลผลภาพก็คือ

- Image Processing (Image in Image out)

วิธีนี้จะใช้กระบวนการทาง DIP เพื่อให้ได้ภาพออกมา เช่น การตกแต่งภาพด้วยโปรแกรมโฟโต้ช้อป (Photoshop) เป็นต้น

- Image Analysis (Image in Measurements out)

วิธีนี้จะใช้กระบวนการทาง DIP เพื่อให้ได้ค่าการวัดออกมา เช่น การวัดขนาดในงานอุตสาหกรรม เป็นต้น

- Image Understanding (Image in High-Level Description out)

วิธีนี้จะใช้กระบวนการทาง DIP เพื่อให้ได้ผลลัพธ์ออกมาเป็นความหมายตัวอย่างของ High Level Description เช่น การจดจำตัวอักษร (Optical Character Recognition : OCR) เป็นต้น

การประมวลผลภาพด้วยคอมพิวเตอร์ สามารถทำได้โดยนำภาพที่ได้มาจากกล้องหรือแหล่งที่มาของภาพต่างๆ (Image Source) ซึ่งเป็นสัญญาณแอนะล็อก แล้วนำมาแปลงเป็นสัญญาณดิจิทัลที่มีลักษณะเป็นรหัสเชิงตัวเลข 0,1 ที่สามารถใช้รูปแบบทางคณิตศาสตร์เข้ามาช่วยในการคำนวณและการประมวลผลข้อมูลภาพด้วยคอมพิวเตอร์ได้ต่อไป

การประมวลผลภาพแบ่งออกได้เป็น 2 ระดับ คือ

#### 2.7.1.1 การประมวลผลภาพระดับต่ำ (Low Level Image Processing)

เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากที่เราได้ภาพมา ภาพที่ได้จะประกอบไปด้วยองค์ประกอบต่างๆ มากมาย รวมถึงสิ่งที่ไม่ต้องการด้วย ในที่นี้เราจะเรียกว่า สัญญาณรบกวน (Noise) ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี และไม่สามารถนำไปใช้ในการประมวลผลได้ ดังนั้นการประมวลผลภาพระดับต่ำจึงประกอบไปด้วยการกำจัดสัญญาณรบกวน การทำภาพให้ชัด (High Pass Filter) การหาขอบภาพ (Edge Detection) การแปลงภาพไบนารี (Binary Image) การแบ่งแยกรูปร่างวัตถุ (Image Segmentation) เป็นต้น เพื่อหาค่าตัวแปรต่างๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลภาพระดับสูงต่อไป

#### 2.7.1.2 การประมวลผลภาพระดับสูง (High Level Image Processing)

เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูง คือข้อมูลที่น่ามาใช้ในการประมวลผล โดยการประมวลผลระดับต่ำจะใช้ค่าความสว่างหรือความเข้มของแสงโดยตรง

ส่วนการประมวลผลภาพระดับสูง ข้อมูลที่นำมาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ และการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำ มาอธิบายถึงสัญลักษณ์เหล่านี้

### 2.7.2 การแทนภาพด้วยข้อมูลแบบดิจิทัล

จากที่ได้กล่าวไปแล้วในตอนต้น ข้อมูลภาพแบบดิจิทัลเป็นภาพที่ถูกตัดแปลงมาจากภาพแบบต่อเนื่อง ให้อยู่ในรูปตัวเลขด้วยวิธีดิจิทัล โดยภาพแอนะล็อกจะถูกแบ่งให้เป็นพื้นที่สี่เหลี่ยมเล็กๆ ที่เรียกว่า พิกเซล (Pixel) โดยแต่ละพิกเซลจะใช้  $(x, y)$  ในการระบุตำแหน่ง การแสดงข้อมูลภาพดิจิทัลสามารถอธิบายได้ด้วย เมทริกซ์  $(M \times N)$  และให้จุดต่างๆ ที่อยู่ในเมทริกซ์เป็นจุดที่พิกัด  $(x, y)$  ใดๆ เป็นส่วนประกอบของภาพ

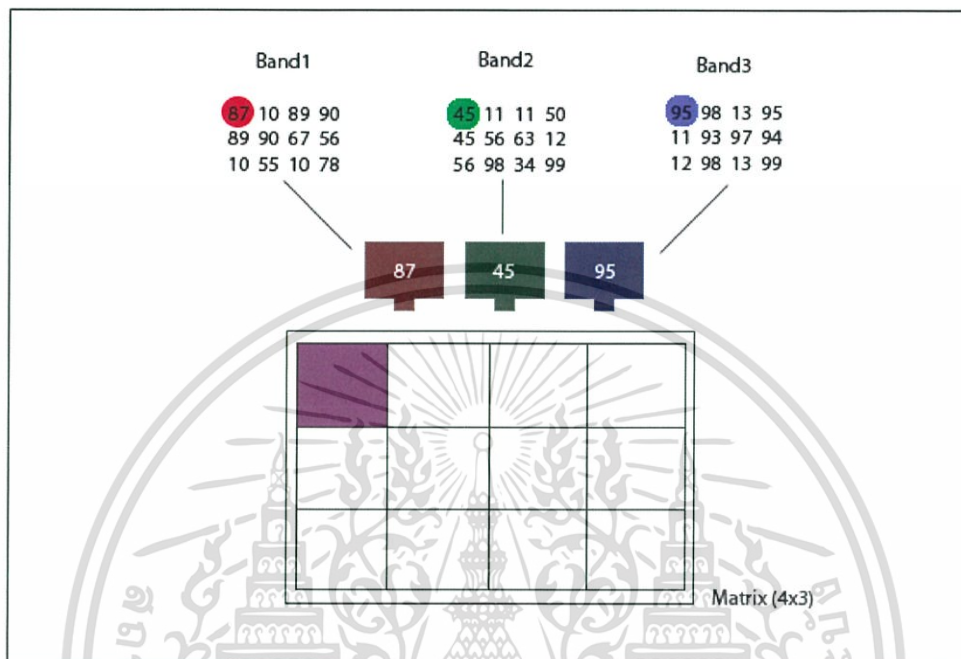


รูปที่ 2.14 ตัวอย่างพิกัดของจุดในภาพแอนะล็อก [12]

ค่าของพิกเซล หรือฟังก์ชัน  $(x, y)$  ณ จุดใดๆ จะแสดงได้ด้วยค่าของความเข้มแสงซึ่งอาจแบ่งได้หลายระดับ ถ้ามี 2 ระดับก็จะเป็น 0 กับ 1 จากรูป จุดต่างๆ ที่แสดงอยู่ในพิกเซลนี้ก็คือความสว่างหรือค่าลูมิแนนซ์ (Luminance :  $L$ ) ของภาพ ถ้าภาพนั้นเป็นภาพขาวดำขนาด 8 บิต จะมีค่า  $L$  เท่ากับ  $2^8$  หรือเท่ากับ 256 คือตั้งแต่ระดับ 0 จนถึง 255 บางครั้งค่าความสว่าง ( $L$ ) อาจหมายถึงระดับความละเอียดของภาพ (Image Resolution) ถ้าพิกเซลเป็นภาพขาวดำจะอ่านค่าภาพดิจิทัลในรูปแบบเมทริกซ์ 2 มิติ ขนาด  $(M \times N)$  ได้ดังนี้

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix}_{(M \times N)} \quad (2.4)$$

โดยที่ค่า  $f(x,y)$  จะอยู่ในช่วง 0 ถึง 255 สมมติว่าอ่านค่าพิกเซลจากภาพหนึ่งได้  $f(x,y)$  เท่ากับ 10 แสดงว่าจุดพิกเซลนั้นมีความสว่างน้อยหรือค่อนข้างจะดำ ถ้าอ่านได้เป็น 255 แสดงว่าจุดพิกเซลนั้นมีความสว่างมากหรือเป็นสีขาว



รูปที่ 2.15 ค่าพิกเซลที่มาจากผลการผสมสี RGB [12]

จากรูปด้านบนจะช่วยให้เข้าใจการแสดงค่าพิกเซลในเมทริกซ์มากขึ้น เริ่มต้นด้วย พิกัด  $f(x,y) = f(0,0)$  ค่าของพิกเซลที่ได้จะเป็นการผสมสีกันระหว่างค่าของแม่สีทั้งสาม ซึ่ง ได้แก่ แดง เขียว น้ำเงิน ซึ่งในที่นี้มีค่าเท่ากับ (87,45,95)

### 2.7.3 ภาพที่นำมาใช้ในการประมวลผล

โดยทั่วไปภาพที่นำมาใช้ในการประมวลผลภาพดิจิทัลนั้นแบ่งออกเป็นสองประเภทหลักๆ คือ

#### 2.7.3.1 ภาพเคลื่อนไหว

ความจริงแล้วภาพเคลื่อนไหวก็คือภาพนิ่งที่นำมาแสดงต่อกันแบบต่อเนื่อง จะต้องใช้รูปภาพอย่างน้อย 24 รูปต่อหนึ่งวินาที เนื่องจากสายตาของคนเราเมื่อนำภาพนิ่งมาฉายติดต่อกันมากกว่า 24 รูปต่อหนึ่งวินาทีแล้วก็จะมองเห็นภาพนั้นเป็นภาพเคลื่อนไหว เพราะสายตารเราแยกไม่ออกเนื่องจากมีความเร็วมากเกินไป แต่หากนำภาพนิ่งมาฉายติดต่อกันน้อยกว่า 24 รูปต่อหนึ่งวินาทีแล้ว เราจะมองเห็นว่าภาพนั้นไม่ต่อเนื่อง



### 2.7.3.2 ภาพสีหรือภาพนิ่ง

ภาพที่นำเข้ามาประมวลผลในคอมพิวเตอร์นั้น ถ้าในระบบ RGB ก็จะใช้ความเข้มแสงสีแดง เขียวและน้ำเงิน ความหมายของภาพนิ่งก็คือมีอยู่เพียงภาพเดียว (ถ้าเป็นภาพเคลื่อนไหวจะประกอบไปด้วยภาพนิ่งหลายๆภาพ) ภาพนิ่งที่นำมาใช้ก็มีอยู่หลายฟอร์แมท (Format) ไม่ว่าจะเป็น .bmp หรือ .jpg เป็นต้น ส่วนการเลือกใช้นั้นก็แล้วแต่ความเหมาะสม ส่วนใหญ่จะเป็น .bmp เพราะไม่ต้องถอดรหัสก่อน เนื่องจากภาพที่เป็น .jpg นั้นมีการบีบอัดภาพให้มีขนาดเล็ก ดังนั้นหากจะนำมาใช้ก็ต้องคลายข้อมูลออกก่อนที่จะนำภาพไปประมวลผลต่อ

### 2.7.4 มาตรฐานสี

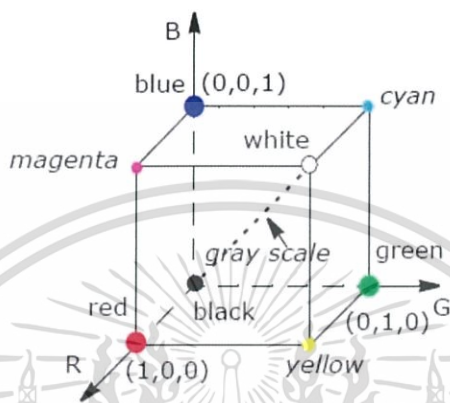
มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีหลายระบบด้วยกันแต่โดยทั่วไปทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ในสเปซ 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปซ ซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่น ในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงิน ในระบบ HLS จะมีแกนเป็นค่าสี (Hue) ความสว่าง (Lightness) และความบริสุทธิ์ของสี (Saturation) โดยระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียว และน้ำเงิน โดยปกติจะนำไปใช้ในจอภาพแบบ CRT และเนื่องจากระบบสี RGB เป็นระบบสีของแสงจึงทำให้ภาพที่ได้ออกมานั้นมีความสมจริงและยังดูสวยงาม



รูปที่ 2.16 ระบบสี RGB [12]

### 2.7.5 โมเดลสี (Color Model)

โมเดลสี ประกอบด้วย 3 แม่สีหลักได้แก่ สีแดง สีเขียว และสีน้ำเงิน ถ้านำแต่ละแม่สีมาพล็อตกราฟในระบบพิกัด โดยแต่ละสีมีค่า 0 ถึง 1 (0 แสดงถึงค่าความมืด และ 1 แสดงถึงค่าความสว่าง) จะได้ภาพการผสมสีทางแสงหรือการบวกแม่สีเข้าด้วยกัน



รูปที่ 2.17 โมเดลสีระบบ RGB [12]

แต่ถ้าแม่สีแต่ละสีมีขนาด 8 บิต รวมทั้งหมดเท่ากับ 24 บิต ซึ่งสามารถสร้างสีใหม่ได้ถึง  $256 \times 256 \times 256$  เท่ากับ 16,777,216 สี ซึ่งในที่นี้จะใช้พิกเซลภาพที่มีแต่ละแม่สีเท่ากับ 8 บิต หรือเรียกว่ามีความลึกเท่ากับ 24 บิตเป็นหลัก ถ้าพิกเซลเป็นภาพสีขนาด 24 บิตจะอ่านค่าภาพดิจิทัลในรูปแบบเมทริกซ์ 2 มิติ ขนาด  $M \times N$  เหมือนกับในสมการข้างต้นที่ได้กล่าวไปแล้ว

### 2.7.6 ภาพระดับเทา (Grayscale Image)

เป็นกระบวนการที่ทำให้ความเข้มของแม่สีในภาพมีระดับเดียวกัน (Intensity Transformation) พูด่ง่ายๆ คือในพิกเซลหนึ่งนั้นจะประกอบไปด้วยสี R, G, B จะเห็นได้ว่ามีถึง 3 ค่า ใน 1 พิกเซล ซึ่งจะทำให้ยากในการประมวลผล ดังนั้นก็ควรทำให้ภาพมีความเข้มสีในระดับเดียวกันทั้งหมดก่อน ด้วยการทำให้เป็น Gray Level (การทำให้แม่สี R, G, B ทั้ง 3 มีค่าเท่ากัน) ส่วนอัลกอริทึมที่ใช้ในการทำภาพระดับเทานั้น ที่นิยมใช้มีอยู่ 2 รูปแบบ คือ วิธีที่ 1 เป็นการเฉลี่ยค่าของแม่สีทั้งสาม เป็นวิธีที่ง่ายที่สุดแต่อาจจะมีความผิดพลาดเพี้ยนของสีได้ จึงมีวิธีที่ 2 ซึ่งคิดตามความสว่างของแต่ละแม่สี โดยมีรูปแบบดังสมการ

$$R_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad (2.5)$$

$$R_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

$$G_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad (2.6)$$

$$G_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

$$B_R = \frac{(R_S + G_S + B_S)}{3} \quad \text{หรือ} \quad (2.7)$$

$$B_R = ((0.299 \times R_S) + (0.587 \times G_S) + (0.114 \times B_S))$$

โดยที่	$R_R$	หมายถึง ค่าเอาต์พุตพิกเซลสีแดง
	$G_R$	หมายถึง ค่าเอาต์พุตพิกเซลสีเขียว
	$B_R$	หมายถึง ค่าเอาต์พุตพิกเซลสีน้ำเงิน
	$R_S$	หมายถึง ค่าอินพุตพิกเซลสีแดง
	$G_S$	หมายถึง ค่าอินพุตพิกเซลสีเขียว
	$B_S$	หมายถึง ค่าอินพุตพิกเซลสีน้ำเงิน

### 2.7.7 การสร้างภาพไบนารีด้วยการใช้เทคนิคการทำ Threshold

การสร้างภาพไบนารี หมายถึง การแปลงข้อมูลภาพที่มีความเข้มหลายระดับให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ คือ หนึ่งจุดภาพมีค่าได้แค่สองค่าเท่านั้นโดยเป็น 0 กับ 1 ซึ่ง 1 จะหมายถึงจุดภาพที่มีสีขาว และ 0 หมายถึงจุดภาพที่มีสีดำ การแปลงข้อมูลภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ มีประโยชน์อีกประการหนึ่งคือการลดเนื้อที่ในการเก็บข้อมูลให้เหลือเพียง 2 บิต



รูปที่ 2.18 ภาพต้นฉบับ (ซ้าย) และภาพที่ถูกทำ Threshold (ขวา) [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างภาพไบนารีนั้น สามารถทำได้โดยการใช้เทคนิคการทำ Threshold โดยพิจารณาว่าจุดใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบกันระหว่างจุดภาพ เริ่มต้นกับค่าคงที่ค่าหนึ่ง เรียกว่า ค่า Threshold ซึ่งจะมีค่าได้ตั้งแต่ 0 – 255 หากค่าของพิกเซลมีค่าน้อยกว่าค่า Threshold ก็ให้พิกเซลนั้นเป็น 0 และหากพิกเซลนั้นมีค่ามากกว่า Threshold ก็ให้พิกเซลมีค่าเป็น 1 ดังสมการ

$$g(x, y) = \begin{cases} 0, & f(x, y) < \text{Threshold}, \\ 1, & f(x, y) \geq \text{Threshold} \end{cases} \quad (2.8)$$

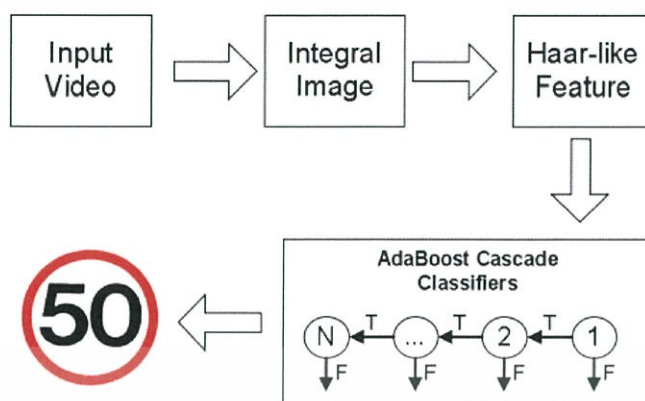
เมื่อ  $g(x, y)$  คือ ระดับสีเทาที่จุด  $(x, y)$  ใดๆ ของภาพ  
 $f(x, y)$  คือ เป็นค่าความสว่างของจุด  $(x, y)$   
 $\text{Threshold}$  คือ ระดับสีเทาที่ใช้อ้างอิงการแปลงข้อมูล

### 2.7.8 การหาคอนทัวร์ของภาพ (Contours Detection)

การหาคอนทัวร์ของภาพ คือการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุจะสามารถคำนวณหาพื้นที่ (ขนาด) หรือรู้จักชนิดของวัตถุนั้นได้ อย่างไรก็ตามการหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เรื่องง่าย โดยเฉพาะอย่างยิ่งการหาขอบภาพที่มีคุณภาพต่ำ มีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วภาพ ขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง หากความต่างนี้มีค่ามากขอบภาพก็จะเห็นได้ชัด ถ้าความแตกต่างมีค่าน้อยขอบภาพก็จะเห็นไม่ชัดเจน

### 2.7.9 การตรวจจับวัตถุด้วยวิธีการของ Viola และ Jones [16, 17]

เราจะใช้วิธีการตรวจจับวัตถุของ Viola และ Jones มาตรวจจับสัญญาณไฟจราจรที่จำลองเอาไว้ ซึ่งการตรวจจับวัตถุด้วยวิธีการของ Viola และ Jones นั้น เป็นการนำเทคนิคการวิเคราะห์ทางสถิติและเทคนิค Machine Learning มาใช้ในการหาคุณลักษณะของรูปภาพที่เป็นสัญญาณจราจรและไม่ได้เป็นสัญญาณจราจร ซึ่งมีขั้นตอนการทำงานดังรูปที่ 2.19



รูปที่ 2.19 แบบแผนการตรวจจับวัตถุด้วยวิธีของ Viola และ Jones [16]

### 2.7.9.1 ภาพอินทิกรัล (Integral Image)

ภาพอินทิกรัลจะทำให้การคำนวณผลรวมของพิกเซลสามารถทำได้ง่ายและรวดเร็วโดยการคำนวณเพียงครั้งเดียว โดยภาพอินทิกรัลที่สร้างได้จะนำไปใช้ในขั้นตอนของ Cascaded Classifier ต่อไป

### 2.7.9.2 วิธีการ Haar-like feature

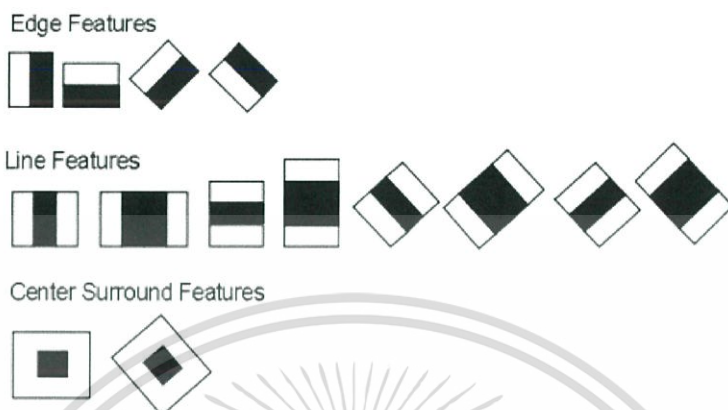
การหารูปร่างของสัญญาณจากราง (Feature) ในภาพจะใช้ตัวกรอง (Filter) ตามลักษณะพื้นฐานของ Haar wavelet จึงเรียกว่า Haar-like feature และรูปร่างแสดงดังรูปที่ 2.20 ตัวกรองจะมีลักษณะเป็นสี่เหลี่ยมและแบ่งออกเป็น 2 ส่วนคือสีขาวและสีดำ ในการทำงานภาพจะถูกกรองด้วยตัวกรองที่มีขนาดต่างๆกันและรูปร่างของป้ายที่ได้จากตัวกรองนี้จะนำไปใช้ในการกำหนดลักษณะของ Weak classifier

### 2.7.9.3 การเรียนรู้แบบ AdaBoost และ Cascade classifiers

ขั้นตอนวิธีการเรียนรู้ด้วยรูปแบบ AdaBoost (AdaBoost learning algorithm) เป็นขั้นตอนวิธีการเรียนรู้เพื่อค้นหาค่าของกลุ่มพิกเซลที่มีลักษณะใกล้เคียงกับภาพอินพุต โดยที่ภาพ Positive คือ ภาพตัวอย่างของวัตถุที่ต้องการตรวจจับ ส่วนภาพ Negative คือ ภาพต่างๆ ไปที่ไม่ใช่วัตถุที่ต้องการตรวจจับ ซึ่งการจำแนกกลุ่มของพิกเซลจะทำภายในส่วนย่อย (sub window) ของภาพ

ส่วนขั้นตอนวิธีการจำแนกรูปร่างที่ต้องการ โดยนำการจำแนกดังที่กล่าวมาแล้วมาทำซ้ำหลายๆรอบ (stage) ซึ่งในแต่ละรอบก็จะตัดพื้นที่ที่เป็นภาพ Negative ออกไปในทุกรอบที่พบ เมื่อจบกระบวนการแล้วจำนวนของ sub window ที่เป็น Negative จะลดลงจนได้

รูปร่างที่ต้องการและในการจำแนกเพื่อหารูปร่างที่ต้องการนี้จะทำกับภาพอินทรีกรัลที่เพิ่งกล่าวมาข้างต้น



รูปที่ 2.20 กลุ่มรูปแบบของ Haar-like Features [16]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

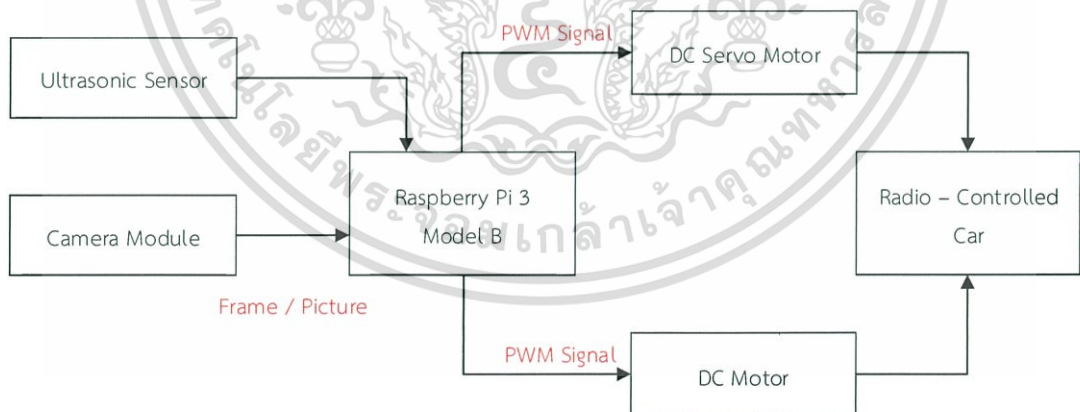
### การออกแบบและการจัดทำปริญญาานิพนธ์

#### 3.1 การออกแบบ

การออกแบบระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ ประกอบไปด้วย 2 ส่วน ส่วนแรกคือ การออกแบบโครงสร้างของรถบังคับ และส่วนที่สองคือการออกแบบการควบคุมการเคลื่อนที่ของรถบังคับ

##### 3.1.1 การทำงานของระบบ

การทำงานของระบบรถบังคับจำลองขับเคลื่อนอัตโนมัติ เริ่มจากการเชื่อมต่อโมดูลกล้องเข้ากับบอร์ดราสเบอร์รี่พาย โมดูลกล้องจะส่งเฟรมภาพที่ได้ให้กับบอร์ดราสเบอร์รี่พายเพื่อนำไปประมวลผลด้วยการใช้โปรแกรมโอเพ่นซีวี (OpenCV) แล้วออกคำสั่งโดยใช้สัญญาณพัลส์ (Pulse Width Modulation : PWM) ในการควบคุมการขับเคลื่อนของเซอร์โวมอเตอร์และมอเตอร์กระแสตรงตามเงื่อนไขที่เขียนไว้ในโปรแกรม โดยที่เซอร์โวมอเตอร์จะเป็นตัวควบคุมทิศทางการเลี้ยวของล้อคู่หน้าให้กับรถบังคับ ในขณะที่มอเตอร์กระแสตรงจะเป็นตัวควบคุมการขับเคลื่อนของล้อคู่หลังให้กับรถบังคับ ส่วนของอัลตราโซนิกเซ็นเซอร์นั้นจะใช้ในการตรวจจับระยะห่างระหว่างรถบังคับกับสิ่งกีดขวางที่อยู่ด้านหน้า เพื่อควบคุมการหยุดของรถบังคับ ดังรูปที่ 3.1

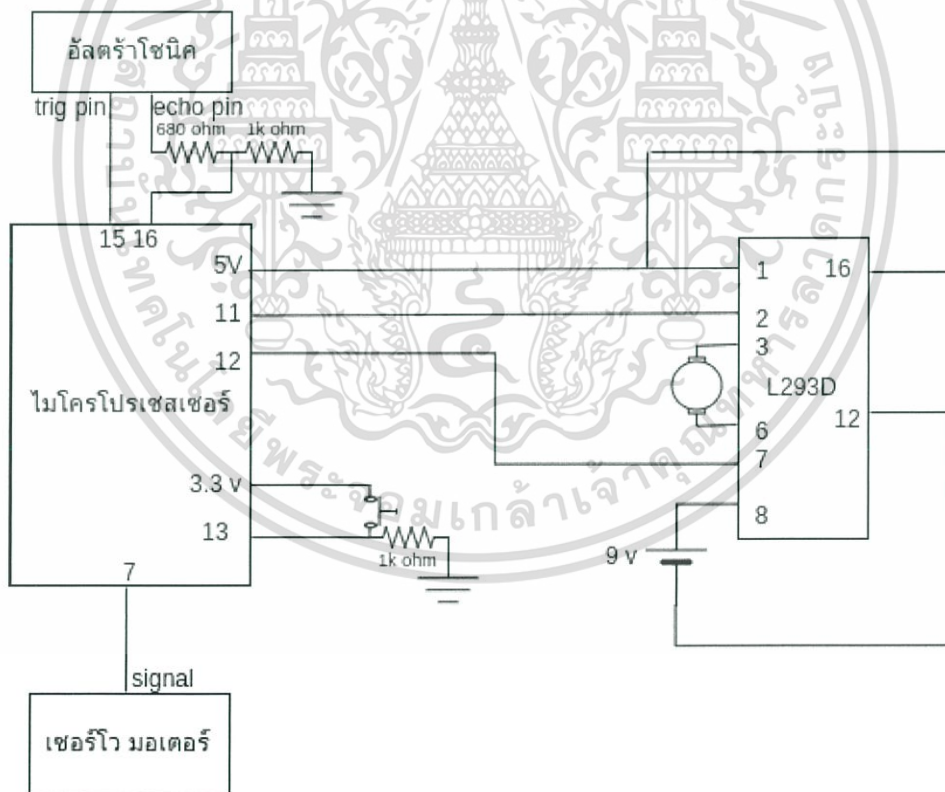


รูปที่ 3.1 บล็อกไดอะแกรมรวมของระบบ

### 3.1.2 การออกแบบโครงสร้างรถบังคับ

เพื่อให้รถบังคับทำงานสัมพันธ์กับคำสั่งที่ส่งมาจากบอร์ดราสเบอร์รี่พายได้นั้น จำเป็นต้องมีการตัดแปลงองค์ประกอบของรถบังคับใหม่ โดยการตัดแผงวงจรและสายอากาศเดิมออก จากนั้นเพิ่มแผงวงจรขับเคลื่อนมอเตอร์ (Motor Driver Circuit) เซอร์โวมอเตอร์ โมดูลกลิ้งบอร์ดราสเบอร์รี่พาย และอัลตราโซนิกเซ็นเซอร์เข้าไป

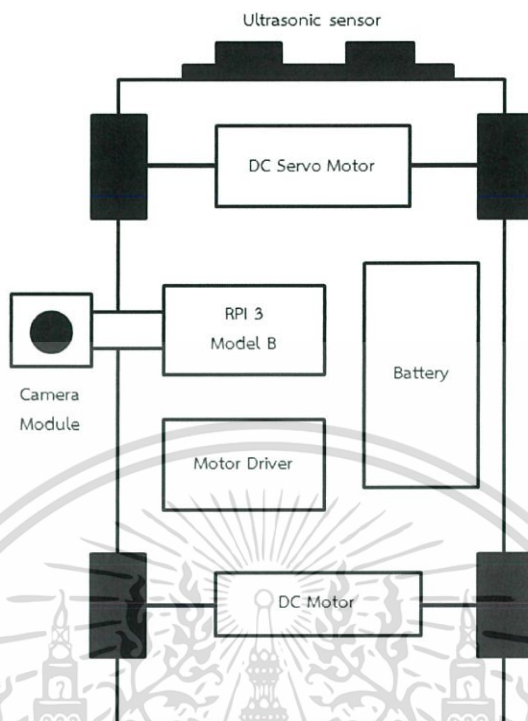
โดยที่แผงวงจรขับเคลื่อนมอเตอร์ จะประกอบไปด้วยไอซี L293D ซึ่งมีหน้าที่ในการขับเคลื่อนมอเตอร์ ทำการต่อขาอินพุตของไอซี L293D เข้ากับพอร์ต GPIO ของบอร์ดราสเบอร์รี่พาย แล้วต่อขาเอาต์พุตของไอซี L293D เข้ากับขามอเตอร์กระแสตรง ในขณะที่อัลตราโซนิกเซ็นเซอร์ที่จะนำมาติดตั้งบริเวณด้านหน้าของรถบังคับและเซอร์โวมอเตอร์นั้น จะนำมาต่อเข้ากับบอร์ดราสเบอร์รี่พายโดยตรงที่พอร์ต GPIO ส่วนโมดูลกลิ้งต่อเข้ากับบอร์ดราสเบอร์รี่พายที่ช่องสำหรับต่อโมดูลกลิ้งโดยตรง โดยแผนภาพวงจรภายในรถบังคับเป็นดังรูปที่ 3.2 และโครงสร้างของรถบังคับคร่าวๆ เป็นดังรูปที่ 3.3



รูปที่ 3.2 แผนภาพวงจรภายในรถบังคับ (Schematic Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 3.3 การดัดแปลงโครงสร้างรถบังคับ

### 3.1.3 การออกแบบการควบคุมการเคลื่อนที่ของรถบังคับ

เมื่อออกแบบส่วนโครงสร้างของรถบังคับเสร็จเรียบร้อยแล้ว จะเริ่มต้นออกแบบในส่วนที่สอง นั่นคือส่วนของการออกแบบการควบคุมการเคลื่อนที่ของรถบังคับ

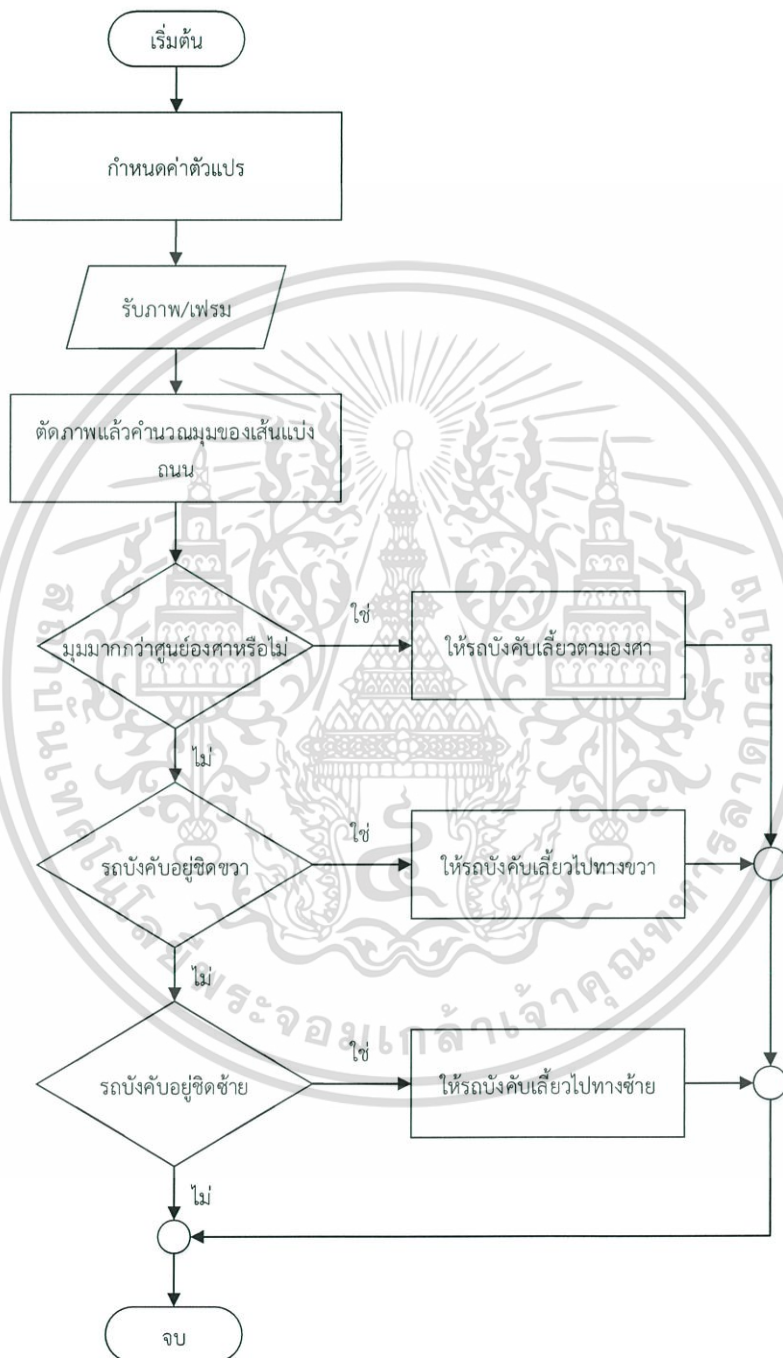
#### 3.1.3.1 การออกแบบการควบคุมให้รถบังคับเคลื่อนที่หรือหยุด

การควบคุมให้รถบังคับสามารถเคลื่อนที่หรือหยุดได้นั้น ทำได้โดยการต่อมอเตอร์กระแสตรงเข้าที่ขาเอาต์พุตของ L293D จากนั้นต่อขาอินพุตของ L293D ไปยังพอร์ต GPIO ของบอร์ดราสเบอร์รี่พาย โดยให้บอร์ดราสเบอร์รี่พายจ่ายสัญญาณออกมาเป็นสัญญาณ PWM เพื่อควบคุมความเร็วรอบของมอเตอร์กระแสตรง ซึ่งถ้าหากสัญญาณพัลส์มี Duty cycle มาก มอเตอร์กระแสตรงจะมีความเร็วรอบในการหมุนเพิ่มมากขึ้น แต่ถ้าหากสัญญาณพัลส์มี Duty cycle น้อย มอเตอร์กระแสตรงก็จะมีความเร็วรอบในการหมุนลดน้อยลง

#### 3.1.3.2 การออกแบบการควบคุมทิศทางของรถบังคับ

การควบคุมทิศทางของรถบังคับจะเริ่มทำงานจากการเรียกใช้โมดูลกล้องเพื่อรับภาพและตัดภาพให้เหลือเฉพาะส่วนที่เป็นพื้นที่ถนนจำลอง จากนั้นจะเข้าสู่กระบวนการวิเคราะห์

ทิศทางของรถบังคับ โดยการนำโปรแกรมโอเพนซีวีเข้ามาช่วยในการวิเคราะห์และประมวลผลหาเส้นแบ่งถนนด้วยวิธีการคอนทัวร์ (Contour) และหามุมของถนนด้วยหลักการทางตรีโกณมิติ ซึ่งแผนภาพการออกแบบการควบคุมทิศทางของรถบังคับ เป็นไปดังรูปที่ 3.4

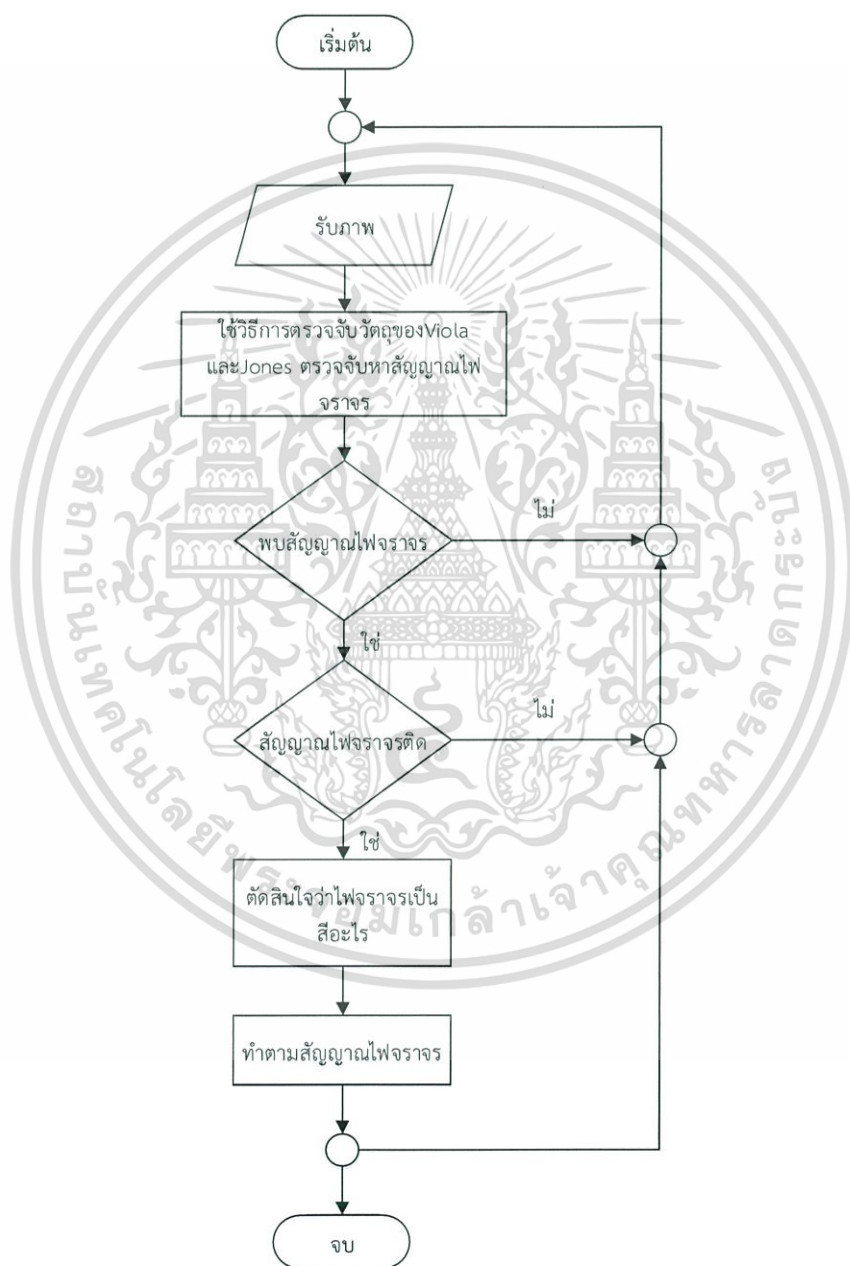


รูปที่ 3.4 แผนภาพการควบคุมทิศทางรถบังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3.3 การออกแบบการตรวจจับสัญญาณจราจร

สัญญาณจราจรที่นำมาใช้ในการทดลองนี้ คือ สัญญาณไฟจราจร ซึ่งขั้นตอนการทำงานของกระบวนการการตรวจจับสัญญาณไฟจราจรนั้น เป็นดังรูปที่ 3.5 โดยใช้วิธีการของ Viola และ Jones เข้ามาช่วยในการตรวจจับหาสัญญาณไฟจราจร เพื่อควบคุมรถบังคับให้สามารถปฏิบัติตามสัญญาณไฟจราจรที่แสดงให้เห็นได้

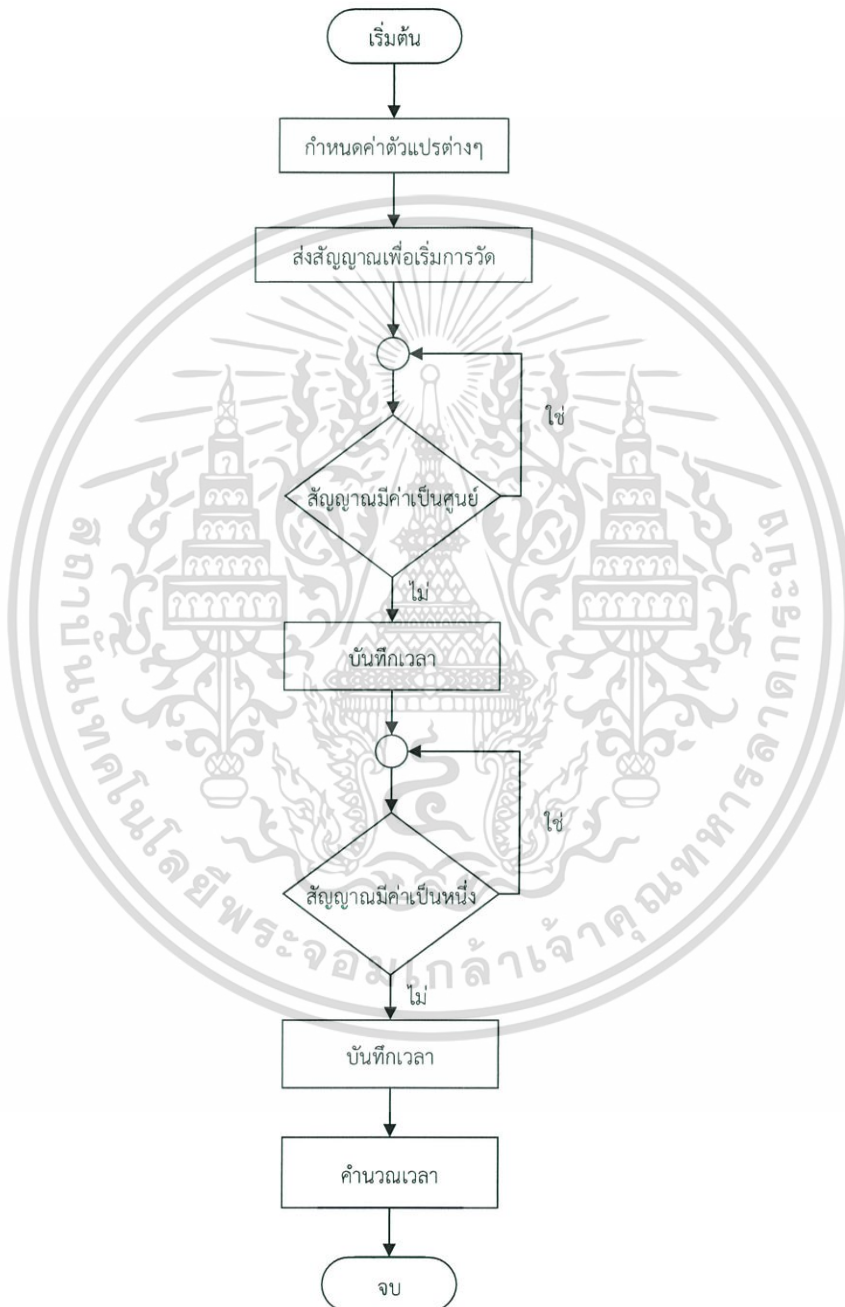


รูปที่ 3.5 แผนภาพการตรวจจับสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3.4 การออกแบบการวัดระยะห่างจากสิ่งกีดขวางด้านหน้าโดยใช้อัลตราโซนิกเซ็นเซอร์

ขั้นตอนการทำงานในการวัดระยะห่างจากสิ่งกีดขวางที่อยู่บริเวณด้านหน้าของรถบังคับนั้น เป็นดังรูปที่ 3.6 โดยนำเอาอัลตราโซนิกเซ็นเซอร์มาช่วยในการวัดระยะห่าง



รูปที่ 3.6 แผนภาพการวัดระยะห่างจากสิ่งกีดขวางด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

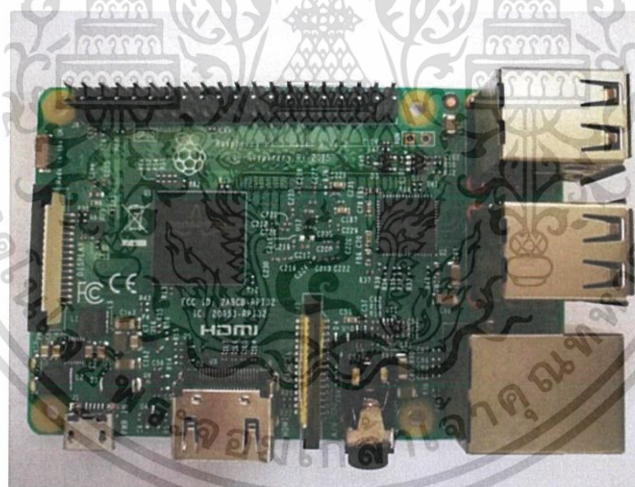
## 3.2 เครื่องมือที่ใช้ในการทดลอง

### 3.2.1 รถบังคับ (Radio - Controlled Car : RC Car)



รูปที่ 3.7 รถบังคับ

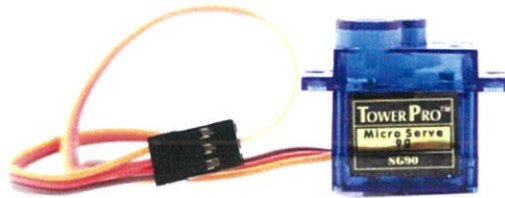
### 3.2.2 บอร์ดราสเบอร์รี่พาย (Raspberry Pi 3 Model B)



รูปที่ 3.8 บอร์ดราสเบอร์รี่พาย 3 โมเดลบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 เซอร์โวมอเตอร์ (Servo Motor)



รูปที่ 3.9 เซอร์โวมอเตอร์ รุ่น SG90 [18]

### 3.2.4 ไอซีขับมอเตอร์ (IC Driver)



รูปที่ 3.10 ไอซีขับมอเตอร์ L293D [19]

### 3.2.5 โมดูลกล้อง (Camera module)



รูปที่ 3.11 โมดูลกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6 โปรแกรมโอเพนซีวี (OpenCV Program)



รูปที่ 3.12 โปรแกรมโอเพนซีวี [20]

### 3.2.7 อัลตราโซนิกเซ็นเซอร์ (Ultrasonic Sensor)



รูปที่ 3.13 อัลตราโซนิกเซ็นเซอร์ [21]

## 3.3 การจัดเก็บผลการทดลอง

เป็นจัดเก็บผลการทดสอบของระบบ นั่นคือการทดสอบใช้งานโปรแกรมโอเพนซีวี สำหรับวิเคราะห์การตรวจจับภาพที่รับมาจากโมดูลกล้อง และทดสอบการรับ – ส่งคำสั่งจากบอร์ด ราวสเบอร์รี่ไปยังส่วนย่อยต่างๆ ของรถบังคับ เช่น วงจรขับมอเตอร์, มอเตอร์กระแสตรง, เซอร์โว มอเตอร์, โมดูลกล้อง เป็นต้น เพื่อควบคุมทิศทางและการเคลื่อนที่ของรถบังคับ

### 3.3.1 ทดสอบการรับคำสั่งจากบอร์ดราสเบอร์รี่พายไปยังเซอร์โวมอเตอร์

ทดสอบการทำงานโดยการเชื่อมต่อพอร์ต GPIO (พอร์ตที่ 7) ของบอร์ดราสเบอร์รี่พายเข้ากับเซอร์โวมอเตอร์ แล้วเขียนโปรแกรมป้อนคำสั่งให้พอร์ต GPIO ควบคุมองศาการหมุนของเซอร์โวมอเตอร์ด้วยการใช้สัญญาณ PWM

### 3.3.2 ทดสอบการรับคำสั่งจากบอร์ดราสเบอร์รี่พายไปยังมอเตอร์กระแสตรง

ทดสอบการทำงานโดยการเชื่อมต่อพอร์ต GPIO (พอร์ตที่ 11 และ 12) ของบอร์ดราสเบอร์รี่พาย เข้ากับวงจรขับเคลื่อนมอเตอร์กระแสตรง แล้วเขียนโปรแกรมป้อนคำสั่งให้บอร์ดราสเบอร์รี่พายจ่ายสัญญาณออกมาเป็นสัญญาณ PWM เพื่อใช้ควบคุมความเร็วรอบในการหมุนของมอเตอร์

### 3.3.3 ทดสอบการตรวจจับเส้นทาง

ทดสอบการทำงานโดยการเชื่อมต่อโมดูลกล้องเข้ากับบอร์ดราสเบอร์รี่พายเพื่อรับสัญญาณภาพ แล้วใช้โปรแกรมโอเพนซีวีมาช่วยในการประมวลผลภาพที่ได้ด้วยวิธีคอนทัวร์ (Contour) หาตำแหน่งของรถด้วยพิกัดที่ได้รับมาจากผลลัพธ์ของการประมวลผลภาพด้วยวิธีคอนทัวร์ และการหามุมถนนด้วยหลักการทางตรีโกณมิติ

### 3.3.4 ทดสอบการตรวจจับสัญญาณจราจร

ทดสอบการทำงานโดยการเชื่อมต่อโมดูลกล้องเข้ากับบอร์ดราสเบอร์รี่พายเพื่อรับสัญญาณภาพ แล้วใช้โปรแกรมโอเพนซีวีมาช่วยในการประมวลผลภาพที่ได้ด้วยวิธีการตรวจจับของ Viola กับ Joans

### 3.3.5 ทดสอบการทำงานของอัลตราโซนิกเซ็นเซอร์

ทดสอบการทำงานโดยการนำอัลตราโซนิกเซ็นเซอร์ไปติดตั้งบริเวณหน้ารถบังคับ แล้วคำนวณหาความสัมพันธ์ระหว่างระยะห่างอ้างอิงกับเวลาที่วัดได้ และคำนวณหาความสัมพันธ์ระหว่างระยะห่างอ้างอิงกับระยะห่างที่วัดได้ จากนั้นนำไปคำนวณหาความสัมพันธ์ระหว่างระยะห่างอ้างอิงและเวลาที่วัดได้ เพื่อนำไปใช้ในการวัดระยะห่างระหว่างสิ่งกีดขวางที่อยู่ข้างหน้า



## บทที่ 4

### ผลการทดลอง

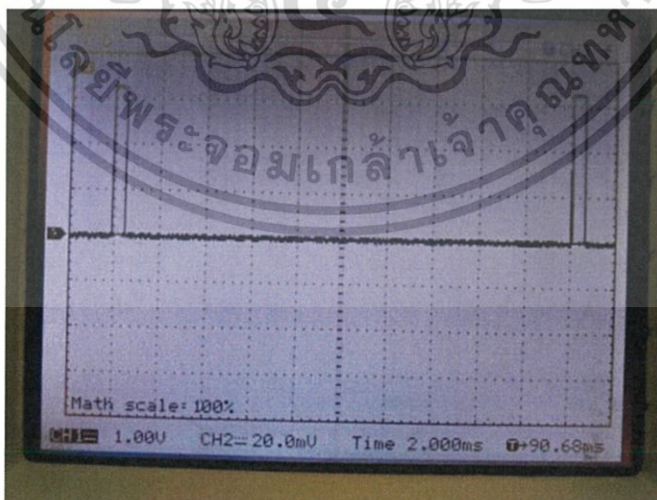
#### 4.1 ผลการทดลอง 1

ทดสอบการควบคุมเซอร์โวมอเตอร์ด้วยบอร์ดราสเบอร์รี่พาย โดยทำการป้อนสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation : PWM) ที่มีความกว้างของพัลส์ที่แตกต่างกัน มาควบคุมตัวเซอร์โวมอเตอร์ให้หมุนไปในองศาที่แตกต่างกัน โดยได้ผลลัพธ์เป็นไปดังตารางที่ 4.1

ตารางที่ 4.1 ค่าความกว้างของพัลส์ที่ป้อนลงในโปรแกรมเทียบกับองศาของเซอร์โวมอเตอร์

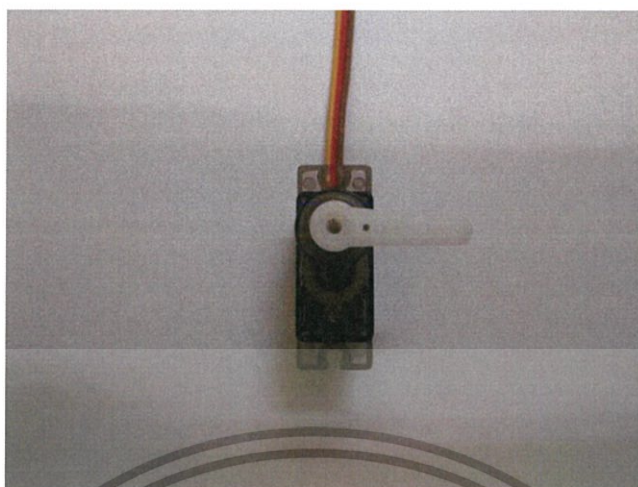
ความกว้างของพัลส์	องศาของเซอร์โวมอเตอร์
2.5 %	-90 องศา
7.5 %	0 องศา
12.5 %	+90 องศา

ซึ่งเมื่อความกว้างของพัลส์มีค่า 2.5 เปอร์เซ็นต์ แขนของเซอร์โวมอเตอร์จะหมุนไปทางขวา 90 องศา หากป้อนความกว้างของพัลส์ให้มีค่า 7.5 เปอร์เซ็นต์ แขนของเซอร์โวมอเตอร์จะอยู่ที่ตำแหน่ง 0 องศา แต่ถ้าหากป้อนความกว้างของพัลส์ให้มีค่า 12.5 เปอร์เซ็นต์ แขนของเซอร์โวมอเตอร์จะหมุนไปทางซ้าย 90 องศา ซึ่งผลการทดสอบที่ได้ เป็นไปดังรูปต่อไปนี้



รูปที่ 4.1 สัญญาณพัลส์ที่มีความกว้างของพัลส์เป็น 2.5 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

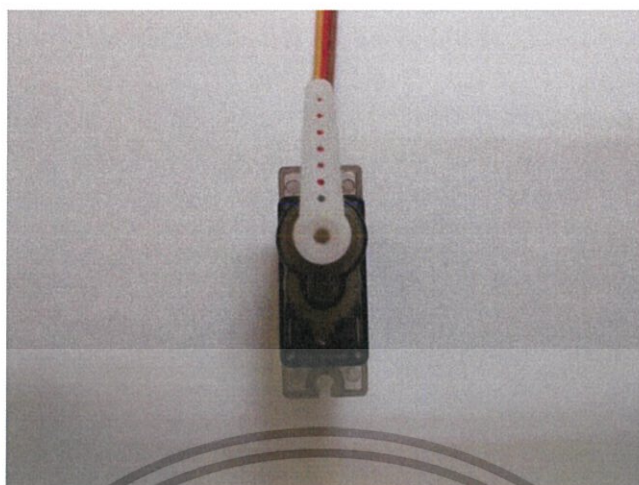


รูปที่ 4.2 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 2.5 เปอร์เซ็นต์ (-90 องศา)



รูปที่ 4.3 สัญญาณพัลส์ที่มีความกว้างของพัลส์เป็น 7.5 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

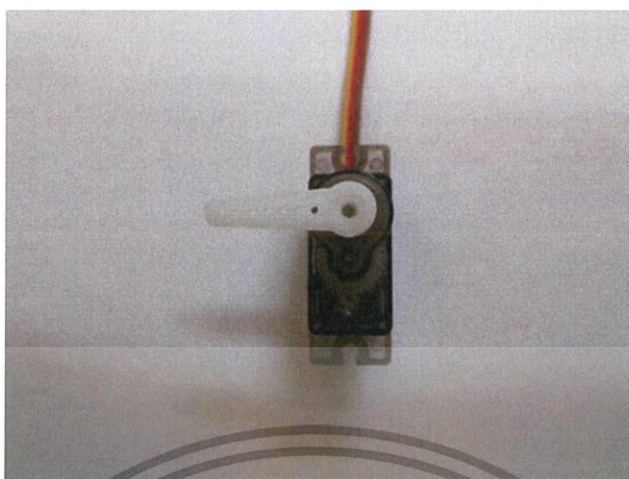


รูปที่ 4.4 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 7.5 เปอร์เซ็นต์ (0 องศา)



รูปที่ 4.5 สัญญาณพัลส์ที่มีความกว้างของพัลส์เป็น 12.5 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การหมุนของเซอร์โวมอเตอร์เมื่อป้อนพัลส์ 12.5 เปอร์เซ็นต์ (+90 องศา)

## 4.2 ผลการทดลอง 2

ทดสอบการควบคุมมอเตอร์กระแสตรงด้วยไอซี L293D โดยทำการป้อนสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation : PWM) ที่มีความกว้างของพัลส์ที่แตกต่างกันเพื่อควบคุมความเร็วของมอเตอร์ ได้ผลการทดสอบ 2 ตามตารางที่ 4.2

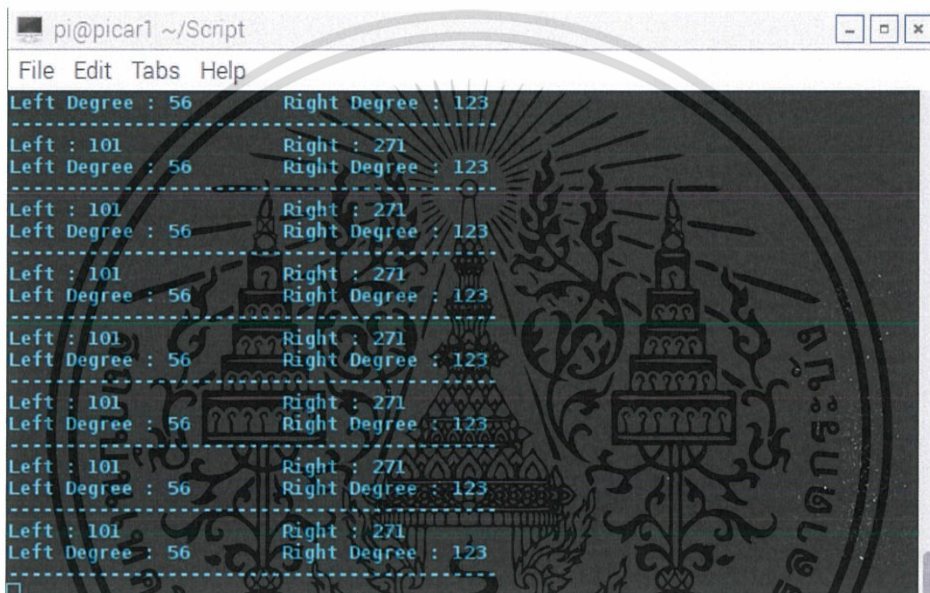
ตารางที่ 4.2 ความสัมพันธ์ระหว่างความกว้างของพัลส์ที่ป้อนเข้าไอซีและความเร็วของมอเตอร์กระแสตรง

ความกว้างของพัลส์ที่ป้อนเข้าที่ขาสัญญาณขาเข้า 1 (หน่วยเปอร์เซ็นต์)	ความกว้างของพัลส์ที่ป้อนเข้าที่ขาสัญญาณขาเข้า 2 (หน่วยเปอร์เซ็นต์)	ทิศทางการหมุนของมอเตอร์กระแสตรง	ความเร็วการหมุนของมอเตอร์กระแสตรง
0	0	ไม่มีการหมุน	ไม่มีการหมุน
50	0	หมุนตามเข็มนาฬิกา	ปานกลาง
100	0	หมุนตามเข็มนาฬิกา	เร็ว
0	50	หมุนทวนเข็มนาฬิกา	ปานกลาง
0	100	หมุนทวนเข็มนาฬิกา	เร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ผลการทดลอง 3

ทำการถ่ายวิดีโอด้วยโมดูลกล้องที่เชื่อมต่อเข้ากับบอร์ดราสเบอร์รี่พาย เพื่อนำเฟรมของภาพที่ได้ไปประมวลผลหาเส้นขอบของถนนที่จำลองไว้ โดยใช้เฉพาะเส้นด้านในของเลนถนนทั้งซ้ายและขวาเท่านั้นด้วยวิธีคอนทัวร์ จากนั้นนำไปคำนวณหามุมของเส้นถนนที่ทำกับแนวแกน  $x$  เพื่อระบุว่าตัวรถบังคับอยู่ในตำแหน่งกึ่งกลางเส้นแบ่งของถนนที่จำลองไว้หรือไม่ อีกทั้งยังคำนวณองศาของถนนจำลองเพื่อตัดสินใจในการควบคุมการเลี้ยวของรถบังคับ ซึ่งจะได้ผลการทดลองดังรูปต่อไปนี



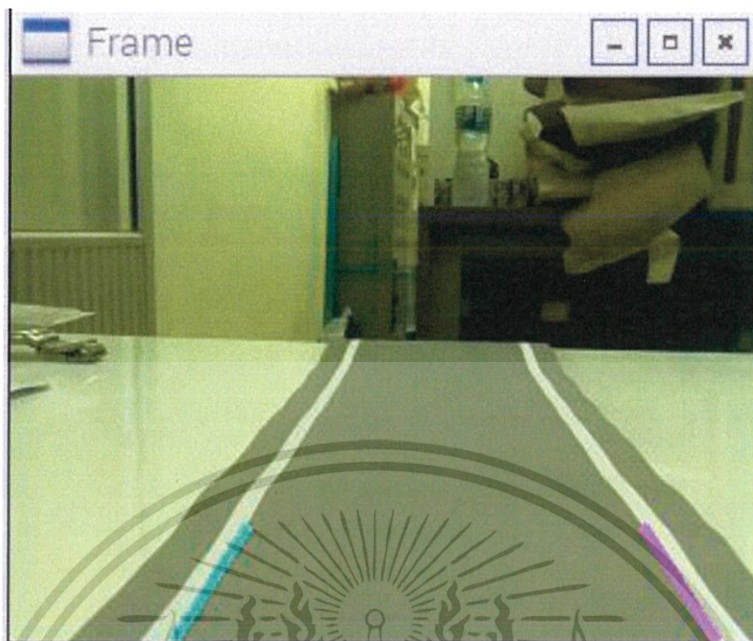
```

pi@picar1 ~/Script
File Edit Tabs Help
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123
Left : 101          Right : 271
Left Degree : 56      Right Degree : 123

```

รูปที่ 4.7 ผลการคำนวณหามุมและตำแหน่งของเส้นถนนทั้งซ้ายและขวาเมื่อถนนเป็นเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

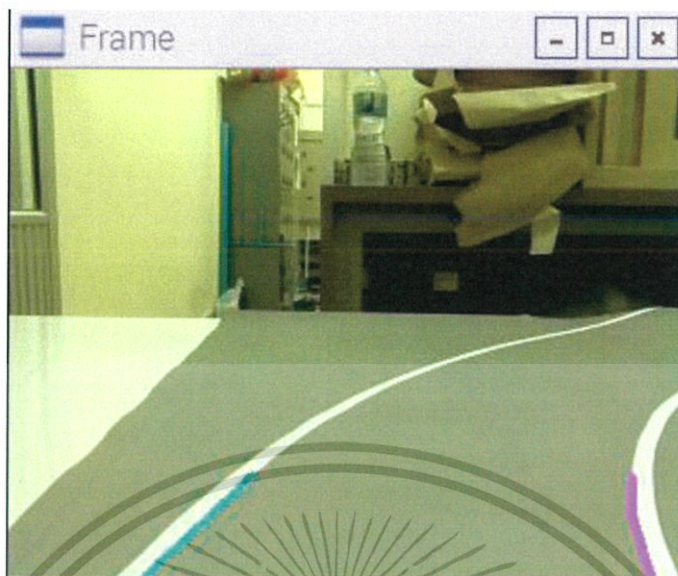


รูปที่ 4.8 การตรวจจับเลนถนนที่เป็นเส้นตรง



รูปที่ 4.9 ผลการคำนวณหามุมและตำแหน่งของเส้นถนนทั้งซ้ายและขวาเมื่อเป็นถนนเส้นโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลการตรวจจับเลนถนนที่เลี้ยวโค้งไปทางขวา

#### 4.4 ผลการทดลอง 4

เพิ่มฟังก์ชันระบุทิศทาง การเคลื่อนที่ และตำแหน่งของรถบังคับ โดยเมื่อโปรแกรมประมวลผลได้ว่าเส้นเลนถนนเป็นแบบเส้นตรง โปรแกรมจะสั่งให้แสดงข้อความ “forward” ดังรูปที่ 4.11



รูปที่ 4.11 โปรแกรมแสดงข้อความ “forward”

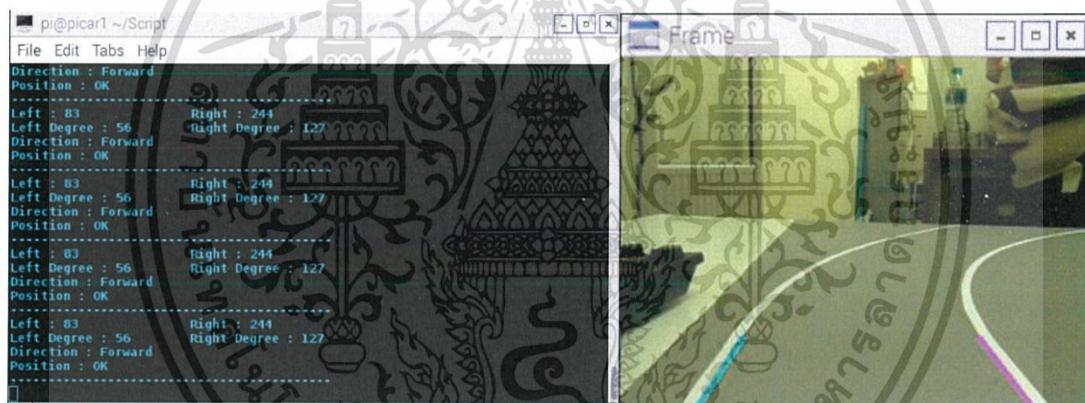
เมื่อโปรแกรมประมวลผลได้ว่าเส้นเลนถนนเป็นแบบเส้นโค้ง ในที่นี้เป็นเส้นโค้งไปทางขวา โปรแกรมจะสั่งให้แสดงข้อความ “turn right” ดังรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 โปรแกรมแสดงข้อความ “turn right”

เมื่อโปรแกรมประมวลผลได้วาระบังคับมีทิศทางการเคลื่อนที่ไปข้างหน้า ในตำแหน่งกึ่งกลางระหว่างเส้นเลนถนน โปรแกรมจะสั่งให้แสดง “Direction: forward” และ “Position: OK” ตามลำดับ ดังรูปที่ 4.13



รูปที่ 4.13 โปรแกรมแสดงข้อความ “Forward” และ “OK”

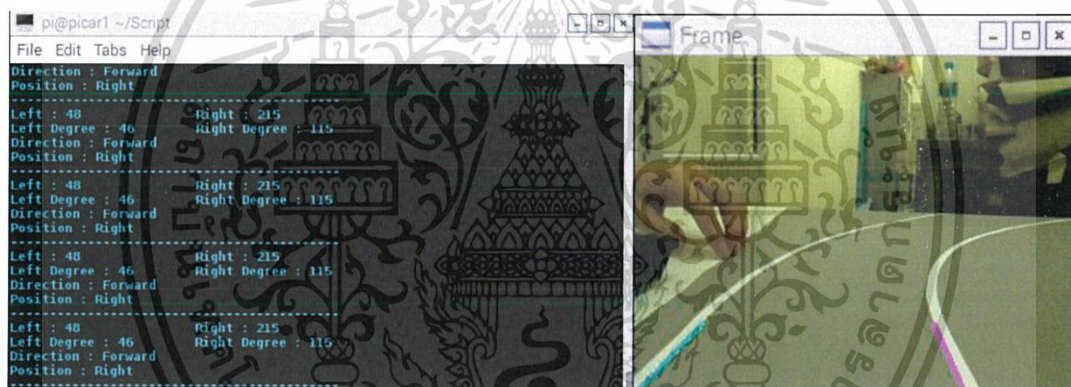
เมื่อโปรแกรมประมวลผลได้วาระบังคับมีทิศทางการเคลื่อนที่ไปข้างหน้า และอยู่ในตำแหน่งชิดซ้ายขอบถนน โปรแกรมจะสั่งให้แสดง “Direction: Forward” และ “Position: Left” ตามลำดับ ดังรูปที่ 4.14





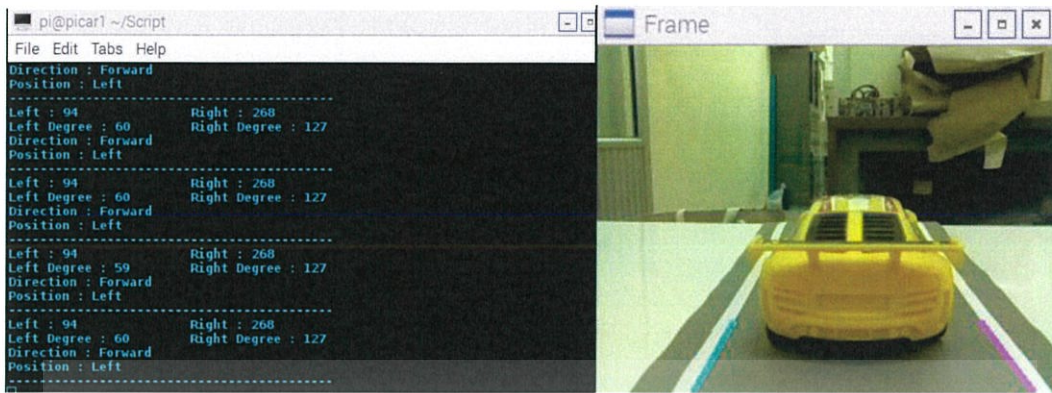
รูปที่ 4.14 โปรแกรมแสดงข้อความ “Forward” และ “Left”

เมื่อโปรแกรมประมวลผลได้วาระบังคับมีทิศทางเคลื่อนที่ไปข้างหน้า และอยู่ในตำแหน่งซิดขวาขอบถนน โปรแกรมจะสั่งให้แสดง “Direction: Forward” และ “Position: Right” ตามลำดับ ดังรูปที่ 4.15



รูปที่ 4.15 โปรแกรมแสดงข้อความ “Forward” และ “Right”

เมื่อมีสิ่งกีดขวางอยู่ข้างหน้า ระบบยังสามารถตรวจจับเส้นขอบถนนได้อยู่ ดังรูปที่ 4.16

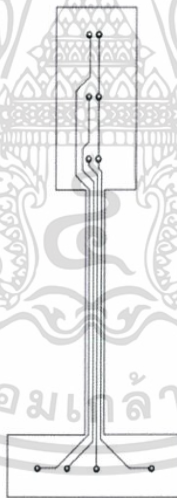


รูปที่ 4.16 ผลการตรวจจับเส้นขอบถนนเมื่อมีสิ่งกีดขวาง

## 4.5 ผลการทดลอง 5

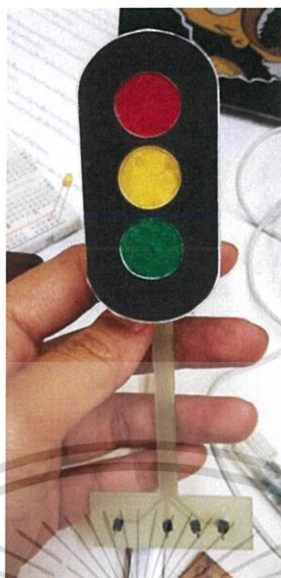
### 4.5.1 จำลองสัญญาณไฟจราจรขนาดเล็ก

จำลองสัญญาณไฟจราจรให้มีขนาดที่เหมาะสมสำหรับตำแหน่งโมดูลกล้องในการตรวจจับเพื่อประมวลผลภาพด้วยส่วนสูงประมาณ 5 นิ้ว



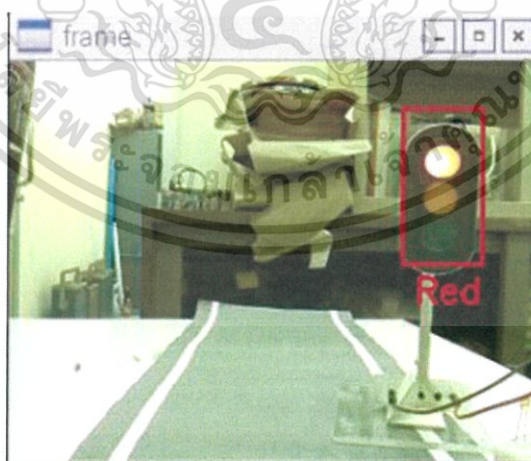
รูปที่ 4.17 ลายวงจรสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



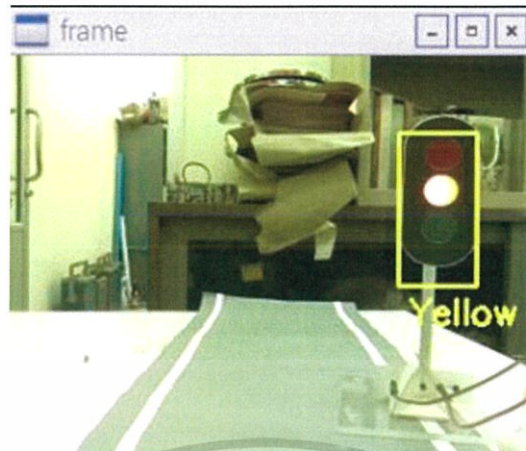
รูปที่ 4.18 สัญญาณไฟจราจรจำลองที่เสร็จแล้ว

4.5.2 ออกแบบโปรแกรมประมวลผลภาพให้สามารถตรวจจับสัญญาณไฟจราจรได้  
 ออกแบบโปรแกรมประมวลผลภาพให้สามารถตรวจจับสัญญาณไฟจราจรได้ โดยใช้  
 กระบวนการที่ให้ระบบเรียนรู้ผ่านตัวอย่างของกลุ่มรูปที่ถูกต้อง ซึ่งคือรูปของสิ่งที่เราต้องการ  
 ตรวจจับแต่ต้องมีการตัดเอาส่วนอื่นออกให้เหลือเพียงส่วนของสิ่งที่เราต้องการเท่านั้นและตัวอย่าง  
 ของกลุ่มรูปทั่วไปที่ไม่มีสิ่งที่เราต้องการตรวจจับอยู่ในรูปเลยเพื่อที่จะสามารถนำข้อมูลที่ได้ไป  
 ตรวจจับในรูปอื่นๆ ได้ผลดังรูป



รูปที่ 4.19 การตรวจจับเมื่อเป็นไฟแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 การตรวจจับเมื่อเป็นไฟเหลือง

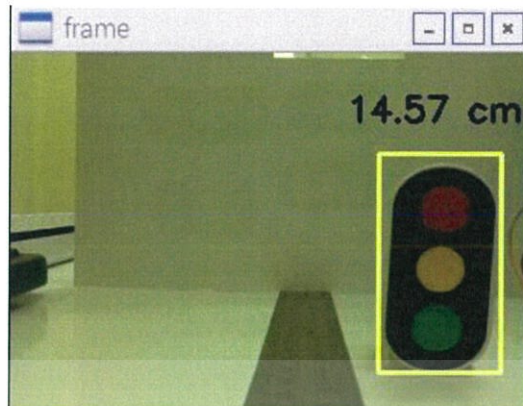


รูปที่ 4.21 การตรวจจับเมื่อเป็นไฟเขียว

#### 4.5.3 เขียนโปรแกรมคำนวณระยะห่างระหว่างโมดูลกล้องกับสัญญาณไฟจราจร

โดยการหาความสัมพันธ์ระหว่างระยะห่างกับความกว้างของสัญญาณไฟที่ตรวจจับได้ เพื่อนำมาใช้ในการคำนวณหาระยะห่าง ให้รถบังคับสามารถหยุดในระยะที่เหมาะสมได้ ผลการทดลองเป็นดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ระยะห่าง 14.00 เซนติเมตรจากโมดูลกล้องถึงสัญญาณไฟจราจร

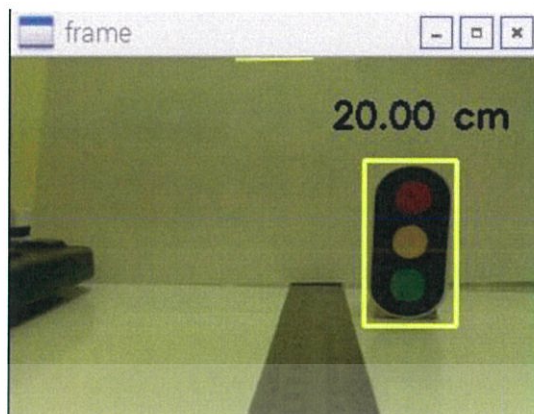


รูปที่ 4.23 ระยะห่าง 16.00 เซนติเมตรจากโมดูลกล้องถึงสัญญาณไฟจราจร



รูปที่ 4.24 ระยะห่าง 18.00 เซนติเมตรจากโมดูลกล้องถึงสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ระยะห่าง 20.00 เซนติเมตรจากโมดูลกล้องถึงสัญญาณไฟจราจร

ตารางที่ 4.3 ผลการคำนวณระยะห่างระหว่างโมดูลกล้องกับสัญญาณไฟจราจรที่ระยะห่างตั้งแต่ 14 cm ถึง 20 cm

ระยะ(cm)	14	16	18	20
ครั้งที่				
1	13.71	16.57	17.43	20.00
2	14.29	16.29	18.29	20.29
3	14.00	16.00	17.71	19.71
4	13.71	16.57	18.00	19.43
5	14.57	16.29	18.29	20.00
ค่าเฉลี่ย	14.06	16.34	17.94	19.89
เปอร์เซ็นต์ความผิดพลาด (%)	-0.43	-2.13	0.33	0.55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.6 ผลการทดลอง 6

ได้ทดลองนำอัลตราโซนิกเซ็นเซอร์มาติดตั้งไว้บริเวณด้านหน้าของรถบังคับ โดยมีจุดประสงค์เพื่อต้องการให้สามารถตรวจจับสิ่งกีดขวางที่อยู่บริเวณด้านหน้าของรถ

### 4.6.1 ผลการวัดความสัมพันธ์ระหว่างระยะทางอ้างอิง (cm) กับเวลาที่วัดได้ (us)

ตารางที่ 4.4 ความสัมพันธ์ระหว่างระยะทางอ้างอิง (cm) กับเวลาที่วัดได้ (us) ที่ระยะห่างตั้งแต่ 5 cm ถึง 25 cm

ระยะทาง อ้างอิง (cm)	ครั้งที่ 1 (us)	ครั้งที่ 2 (us)	ครั้งที่ 3 (us)	ครั้งที่ 4 (us)	ครั้งที่ 5 (us)	ค่าเฉลี่ย (us)
5 cm	283.956	284.910	286.102	287.056	283.957	285.196
10 cm	579.834	560.999	559.091	597.000	585.089	576.401
15 cm	882.864	869.036	864.029	865.936	878.096	871.992
20 cm	1126.051	1132.965	1157.999	1131.058	1158.953	1141.405
25 cm	1428.127	1427.174	1427.889	1432.120	1420.975	1425.457

นำค่าที่คำนวณได้มาพล็อตกราฟ โดยที่แกน x คือระยะทางอ้างอิง (cm) ในขณะที่แกน y คือค่าเวลาที่วัดได้ (us) ได้กราฟดังนี้



รูปที่ 4.26 กราฟความสัมพันธ์ระหว่างระยะทางอ้างอิงกับเวลาที่วัดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6.1 ผลการวัดความสัมพันธ์ระหว่างระยะห่างอ้างอิง (cm) กับระยะห่างที่วัดได้ (cm)

จากการทดสอบจะพบว่าการวัดระยะห่างนั้นมีความคลาดเคลื่อนเล็กน้อย แต่ไม่มีผลกระทบที่รุนแรง

ตารางที่ 4.5 ความสัมพันธ์ระหว่างระยะทางอ้างอิง (cm) กับระยะทางที่วัดได้ (cm) ที่ระยะห่างตั้งแต่ 5 cm ถึง 25 cm

ระยะห่างอ้างอิง (cm)	ครั้งที่ 1 (cm)	ครั้งที่ 2 (cm)	ครั้งที่ 3 (cm)	ครั้งที่ 4 (cm)	ครั้งที่ 5 (cm)	ค่าเฉลี่ย (cm)	เปอร์เซ็นต์ความผิดพลาด %
5 cm	5.08	5.08	5.14	5.11	5.13	5.11	-2.2
10 cm	9.91	10.04	10.01	9.57	10.03	9.91	0.9
15 cm	15.01	15.10	15.17	15.10	15.15	15.11	-0.73
20 cm	19.66	19.55	19.77	19.81	19.84	19.73	1.35
25 cm	24.77	24.87	24.89	24.89	24.93	24.87	0.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

การควบคุมองศาของเซอร์โวมอเตอร์ สามารถทำได้โดยการป้อนสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation : PWM) ที่มีความกว้างของพัลส์ที่แตกต่างกัน

การควบคุมความเร็วและทิศทางการหมุนของมอเตอร์กระแสตรง สามารถทำได้โดยการป้อนสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation : PWM) ที่มีความกว้างของพัลส์ที่แตกต่างกันให้กับตัวไอซี L293D โดยที่ยิ่งความกว้างของพัลส์มากขึ้นจะยิ่งทำให้มอเตอร์กระแสตรงมีความเร็วรอบของการหมุนที่มากขึ้นด้วยเช่นกัน

ในขณะที่การตรวจวัดระยะห่างระหว่างตัวรถบังคับกับวัตถุข้างหน้าสามารถใช้อัลตราโซนิกเซ็นเซอร์ได้ โดยระยะเวลาของสัญญาณที่วัดได้จะแปรผันตรงกับระยะห่าง

ส่วนการควบคุมทิศทางของรถบังคับสามารถทำได้โดยการวิเคราะห์เฟรมภาพที่ถ่ายด้วยโมดูลกล้องที่เชื่อมต่อกับบอร์ดราสเบอร์รี่พายเพื่อวิเคราะห์ตำแหน่งของตัวรถบังคับและวิเคราะห์ความโค้งของถนนจำลอง

และสุดท้ายการตรวจจับวัตถุหรือสัญญาณจราจร สามารถตรวจจับได้จากการใช้วิธีการตรวจจับของ Viola และ Jones ที่อาศัยกระบวนการการเรียนรู้จากเฟรมภาพที่เราต้องการกับเฟรมภาพที่เราไม่ต้องการ

#### 5.2 ข้อเสนอแนะ

การทดสอบในบางขั้นตอนนั้นอาจมีข้อแม้บางประการ อาทิ การตรวจจับเส้นแบ่งถนนจำลองที่หากเส้นแบ่งถนนจำลองมีความเข้มไม่มากพอ หรือสภาพถนนจำลองมีความเสียหายมากหรือรถบังคับอยู่ในสภาวะแวดล้อมที่รบกวนต่อการประมวลผล เช่น ฝุ่น น้ำฝน หรือความร้อน ฯลฯ รวมถึงตัวบอร์ดราสเบอร์รี่พายที่มีความเร็วในการประมวลผลที่ต่ำจนเกินไปนั้น ข้อแม้ดังกล่าวอาจมีผลทำให้ไม่สามารถวิเคราะห์ข้อมูลและก่อให้เกิดปัญหาในการควบคุมรถบังคับได้

## บรรณานุกรม

- [1] Brian Benchoff .“INTRODUCING THE RASPBERRY PI 3” .  
<http://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/> .
- [2] “RASPBERRY PI 3 MODEL B” . <https://www.raspberrypi.org> .
- [3] KMUTNB. “การใช้งานบอร์ด Raspberry Pi 3 เบื้องต้น” .  
[http://cpre.kmutnb.ac.th/esl/learning/index.php?article=rpi3\\_quickstart](http://cpre.kmutnb.ac.th/esl/learning/index.php?article=rpi3_quickstart) .
- [4] Wikipedia. “มอเตอร์” . <https://th.wikipedia.org/wiki/มอเตอร์> .
- [5] “Chapter 3: AC and DC Motors - AC Motors: Magnetic Field (Rotor)” .  
<http://www.globalspec.com/reference/10792/179909/chapter-3-ac-and-dc-motors-ac-motors-magnetic-field-rotor> .
- [6] “บทที่3 มอเตอร์กระแสตรง” . <https://fivedots.coe.psu.ac.th/Software.coe/240-380/PDF/report.doc> .
- [7] “L293x Quadruple Half-H Drivers” . <http://www.ti.com/lit/ds/symlink/l293.pdf> .
- [8] “บทความตัวอย่างการควบคุม RC Servo Motor ด้วย Arduino” .  
<http://www.thaieasyelec.com/article-wiki/review-product-article/บทความตัวอย่างการควบคุม-rc-servo-motor-ด้วย-arduino.html>.
- [9] “Ultrasonic Theory” . <http://www.compomax.co.th/product/ultrasonic-theory/>.
- [10] “คลื่นเสียง Ultrasonic Sensor มีหลักการตรวจจับยังไง?” .  
<https://www.factomart.com/th/factomartblog/structure-and-principle-of-ultrasonic-sensor/>.
- [11] “Thead คืออะไร” . <http://www.mindphp.com/คู่มือ/73-คืออะไร/2264-thread-คืออะไร.html>.
- [12] สิทธิโชค ยอดระยับ. *การเขียนโปรแกรม Digital Image Processing ด้วย Visual Basic*. พิมพ์ครั้งที่ 1. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2550.
- [13] รศ.ดร.กิตติ โพธิ์ชัยวัฒน์กิจ. *การประมวลผลภาพดิจิทัล*. พิมพ์ครั้งที่ 1. กรุงเทพฯ : วศ.สจล, 2549.
- [14] ผศ.ดร.สมเกียรติ อุดมธรรษากุล. *การประมวลผลภาพเบื้องต้น*. พิมพ์ครั้งที่ 1. กรุงเทพฯ : วศ.สจล, 2550.

- [15] Gonzalez, Rafael C. and Woods, Richard E. Digital Image Processing. 3rd ed. Upper Saddle River, NJ. : Prentice Hall, 2008.
- [16] ตะวัน ขุนอาสา. “ระบบตรวจจับและตีความป้ายจำกัดความเร็วในเวลากลางคืนด้วยเทคนิค Haar-like Feature Detection.” รายงานการวิจัยในวิชาโครงการ, หลักสูตร วิทยาศาสตร์บัณฑิต, มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ ศูนย์นนทบุรี.
- [17] Viola, Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features.”, 2001.
- [18] “Tower Pro SG90 9g Gear Steering Servo”.  
<https://www.fasttech.com/product/1170304-tower-pro-sg90-9g-gear-steering-servo>.
- [19] “APERTURA DE GARAJE CON MANDO A DISTANCIA (L293D)”.  
<http://www.prometec.net/apertura-mando-distancia/>.
- [20] “Welcome to opencv documentation!”. <http://docs.opencv.org/3.0-beta/index.html>.
- [21] “HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi”.  
<https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from picamera.array import PiRGBArray
from picamera import PiCamera
import numpy as np
import constants
import cv2
import analyze
from threading import Thread
import device
import control
import time

fourcc = cv2.VideoWriter_fourcc(*'MJPG')
out = cv2.VideoWriter('output.avi', fourcc, constants.frame_rate,
(constants.width, constants.height))

#set camera initial
try:
    camera = PiCamera()
    camera.resolution = (constants.width, constants.height)
    camera.framerate = constants.frame_rate
    rawCapture = PiRGBArray(camera, size=(constants.width,
constants.height))
    #device.car_drive(60)
    for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
        #change object to array
        image = frame.array
        #find Light Traffic
        "findLight = Thread(target = analyze.findLight, args=(image,))
        findLight.start()
        if constants.light != None and constants.light_distance != None:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#if the light traffic is red and near
if constants.light == "red" and constants.light_distance < 25:
    constants.check_red += 1
    if constants.check_red == 1:
        print "Stop!!!!!!!!!!!!!!!!!!!!\n\n-----\n"
        constants.stop = True
#if the light traffic is red and far
elif constants.light == "red" and constants.light_distance < 30:
    constants.check_red_slow = 1
    if constants.check_red_slow == 1:
        print "Slow!!!!!!!!!!!!!!!!!!!!\n\n-----\n"
        constants.speed = 20
        constants.check_red = 0
        constants.check_green = True
        constants.check_slow = True
#if the light traffic is green
elif constants.light == "green" and constants.check_green:
    print "Go!!!!!!!!!!!!!!!!!!!!\n\n-----\n"
    constants.check_green = False
    constants.stop = False
    constants.check_red_slow = 0
    constants.check_red = 0
    constants.direction_deg = 100
else:
    constants.check_red = 0
    constants.stop = False
    constants.check_red_slow = 0
if not constants.stop:
    if constants.light != "green":

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        print "Light : {light} \t Distance : {distance} cm
".format(light=constants.light, distance=constants.light_distance)
'''

#detect the lane
analyze.findLine(image)
if constants.contour_left != None and constants.contour_right !=
None:

    cv2.drawContours(image, [constants.contour_left+[0,
constants.frame_crop_top]], -1, (184, 243, 46), 4)
    cv2.drawContours(image, [constants.contour_right+[0,
constants.frame_crop_top]], -1, (252, 99, 201), 4)
    #if constants.old_time != int(time.time()%60):
    #if not constants.stop:
        #measureDistance = Thread(target=device.measureDistance,
args=())
        #measureDistance.start()
        #constants.old_time = int(time.time()%60)
    if constants.check_red_slow == 1:
        constants.speed = 30
    if constants.left_deg != None and constants.right_deg != None:
    if constants.stop or not constants.front_check:
        print 'stop'
        control.car_drive(0)
        constants.old_speed = 0
        time.sleep(0.07)
    elif not constants.stop:
        print "Direction : {0}\tPosition : {1}".format(constants.direction,
constants.position)

        print '\n-----\n'
        if constants.old_direction_deg != constants.direction_deg:
            constants.old_direction_deg = constants.direction_deg

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        control.car_turn(constants.direction_deg)
        time.sleep(0.07)
    if constants.old_speed != constants.speed:
        constants.old_speed = constants.speed
        control.car_drive(constants.speed)
        time.sleep(0.07)

    #cv2.imshow("image", image)
    out.write(image)
    #wait for esc key
    key = cv2.waitKey(1) & 0xFF

    #clear data in buffer
    rawCapture.truncate(0)

    #if press ESC quit
    if key == 27:
        break
    device.clean()
    control.clean()
    out.release()
except KeyboardInterrupt:
    #close all windows
    device.clean()
    control.clean()
    cv2.destroyAllWindows()
    out.release()
except:
    device.clean()
    control.clean()
    out.release()
    print "Finish !!!!!"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import cv2
import constants
import math
import sys
```

```
light_cascade = cv2.CascadeClassifier('/home/pi/opencv-
3.0.0/data/haarcascades/traffic_light.xml')
```

```
def findLight(frame):
    #crop only top of the frame reduce cpu work
    frame = frame[0:220, 0:320]
    #convert frame from BGR to Gray
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #return array of object that similar to traffic light
    traffic_light = light_cascade.detectMultiScale(gray, 1.1, 3)
    #analyze object in array
    for x, y, w, h in traffic_light:
        #analyze only near object
        if w > 20:
            #crop image only traffic light
            roi = gray[y:y+h, x:x+w]
            mask = cv2.GaussianBlur(roi, (15, 15), 0)
            minVal, maxVal, minXy, maxXy = cv2.minMaxLoc(mask)
            constants.light_pos = [x, y, x+w, y+h]
            #find if light is on
            if (maxVal - minVal)>160:
                #calculate distance between car and traffic light
                distance = float("{0:.2f}".format((77-w)/3.5+14))
                #Red light detect
                if h/5.5 < maxXy[1] < h/2.56 and w*2.0/6.0 < maxXy[0] <
                    w*4.0/6.0:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        constants.light = "red"
        constants.light_distance = distance
        constants.light_color = (40, 40, 255)
#Green light detect
elif h/1.62 < maxXy[1] < h/1.44 and w*2.0/6.0 < maxXy[0] <
w*4.0/6.0:

```

```

        constants.light = "green"
        constants.light_distance = distance
        constants.light_color = (40, 255, 40)
#light detected is not red or green
else:
        constants.light = None
        constants.light_distance = None
        constants.light_color = (255, 255, 255)
else:
        constants.light = None
        constants.light_distance = None
        constants.light_color = (255, 255, 255)
else:
        constants.light = None
        constants.light_distance = None
        constants.light_pos = None
        constants.light_color = (255, 255, 255)

```

```

def findAngle(x1, y1, x2, y2):
    #use for find angle
    dx = x2 - x1
    dy = y2 - y1
    deg = 180 - math.degrees(math.atan2(dy, dx))
    return int(deg)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def findLine(frame):
    left = []
    right = []
    left_contour = []
    right_contour = []
    #crop image only bottom for reduce cpu
    image =
frame[constants.frame_crop_top:constants.frame_crop_bottom,
constants.frame_crop_left:constants.frame_crop_right]
    edges = cv2.Canny(image, 50, 400)
    #return list of contour
    _, cnts, _ = cv2.findContours(edges, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    for c in cnts:
        peri = cv2.arcLength(c, True)
        #find angle of object
        approx = cv2.approxPolyDP(c, 0.02 * peri, True)
        #if line position is less than center
        if approx[0][0][0] < 150:
            try:
                left.append([approx[0][0][0], approx[0][0][1], approx[1][0][0],
approx[1][0][1]])
                left_contour.append(approx)
            except:
                pass
        #if line position is more than center
        elif approx[0][0][0] > 170:
            try:
                right.append([approx[0][0][0], approx[0][0][1], approx[1][0][0],
approx[1][0][1]])
                right_contour.append(approx)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

except:
    pass
try:
    #find the left lane line
    max_left = max(left)
    index_left = left.index(max_left)
    contour_left = left_contour[index_left]

    #find the right lane line
    min_right = min(right)
    index_right = right.index(min_right)
    contour_right = right_contour[index_right]
    #find left and right degree
    left_deg = findAngle(max_left[0], max_left[1], max_left[2],
max_left[3])
    right_deg = findAngle(min_right[0], min_right[1], min_right[2],
min_right[3])
    constants.left_deg = left_deg
    constants.right_deg = right_deg

print max_left[0], min_right[0], left_deg, right_deg
#determine curve of the lane
#if the lane is forward lane
if 52 < left_deg < 72 and 110 < right_deg < 130:
    constants.direction = "Forward"
    constants.speed = 58
#check position of the car
if max_left[0] < 65 and min_right[0] < 225:
    constants.position = "Right"
    constants.direction_deg += 2
#print "Position : Right"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

elif max_left[0] > 75 and 250 > min_right[0] > 230:
    #print "Position : Left"
    constants.position = "Left"
    constants.direction_deg -= 2
else:
    constants.position = "OK"
    #constants.direction_deg = 90
if constants.old_position != constants.position:
    constants.old_position = constants.position
    constants.direction_deg = 100
#if the lane is curve right lane
elif 30 < left_deg < 55 and 70 < right_deg < 120:
    constants.direction = "Right"
    constants.speed = 58
    constants.direction_deg = 85
    if constants.direction_deg <= 55:
        constants.direction_deg = 55
        constants.speed = 150
    if max_left[0] < 106 and min_right[0] < 245:
        #print "Position : Right"
        constants.position = "Right"
    elif max_left[0] > 127 and min_right[0] > 270:
        #print "Position : Left"
        constants.position = "Left"
else:
    #print "Position : OK"
    constants.position = "OK"
if max_left[0] > 120 and min_right[0] > 300:
    constants.direction_deg = 75
    print "75"*100
elif max_left[0] > 135 and min_right[0] > 304:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        constants.direction_deg = 77
        print "70"*100
    elif max_left[0] > 100 and min_right[0] > 265:
        constants.direction_deg = 87
        print "90"*100
    else:
        constants.direction_deg = 87
    #if the lane is curve left lane
    #else can not detect
    else:
        constants.left_deg = None
        constants.right_deg = None
    #return value for main process
    constants.contour_left = contour_left
    constants.contour_right = contour_right
except:
    #print sys.exc_info()
    constants.contour_left = None
    constants.contour_right = None
    constants.left_deg = None
    constants.right_deg = None

```

```

if __name__ == '__main__':
    pass

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
import serial
import os
import time

list_dev = os.popen('ls /dev/ttyUSB*').read().split('\n')[0]

ser = serial.Serial(list_dev, 9600)

def car_turn(degree):
    ser.write('s'+str(degree))

def car_drive(speed):
    ser.write('m'+str(speed))

def clean():
    ser.write('s100')
    time.sleep(0.07)
    ser.write('m0')
    ser.close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ง

ได้ส่วนควบคุมอุปกรณ์เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import time
import RPi.GPIO as GPIO
import constants

#set GPIO mode
GPIO.setmode(GPIO.BOARD)

#set GPIO type I/O
GPIO.setup(constants.decode_pin, GPIO.IN)
GPIO.setup(constants.trig_pin, GPIO.OUT)
GPIO.setup(constants.echo_pin, GPIO.IN)

#function for measure Speed
def measureSpeed():

    while GPIO.input(constants.decode_pin) == 0:
        pass
    while GPIO.input(constants.decode_pin) == 1:
        pass
    #start time for calculate the speed
    start_time = time.time()
    time.sleep(0.1)
    #wait for sensor
    while GPIO.input(constants.decode_pin) == 0:
        pass
    #if sensor detect
    while GPIO.input(constants.decode_pin) == 1:
        pass
    #stop time for calculate the speed
    stop_time = time.time()
    #calculate the speed by using 7.54 cm of wheel's circumference
    speed = "{0:.2f}".format(7.54/(stop_time-start_time))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
print "speed : {} cm/s".format(speed)
time.sleep(0.1)
```

```
#function for measure Distance
```

```
def measureDistance():
```

```
    #trig the ultrasonic
```

```
    GPIO.output(constants.trig_pin, 1)
```

```
    time.sleep(0.00001)
```

```
    GPIO.output(constants.trig_pin, 0)
```

```
    #wait for output setup to HIGH
```

```
    while GPIO.input(constants.echo_pin) == 0:
```

```
        pass
```

```
    #start time for calculate the distance
```

```
    start_time = time.time()
```

```
    #wait for output back to LOW
```

```
    while GPIO.input(constants.echo_pin) == 1:
```

```
        pass
```

```
    #stop time for calculate the distance
```

```
    stop_time = time.time()
```

```
    #calculate the distance by using average speed value (17539.85) from the
```

graph's plot in page 51

```
    distance = "{0:.0f}".format((stop_time-start_time)*17539.85)
```

```
    if int(distance) <= 20:
```

```
        constants.front_check = False
```

```
    else:
```

```
        constants.front_check = True
```

```
#function clean the GPIO
```

```
def clean():
```

```
    GPIO.cleanup()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if __name__ == '__main__':  
    try:  
        measureDistance()  
    except KeyboardInterrupt:  
        GPIO.cleanup()  
    print "stop"
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## L293x Quadruple Half-H Drivers

### 1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

### 2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

### 3 Description

The L293 and L293D devices are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

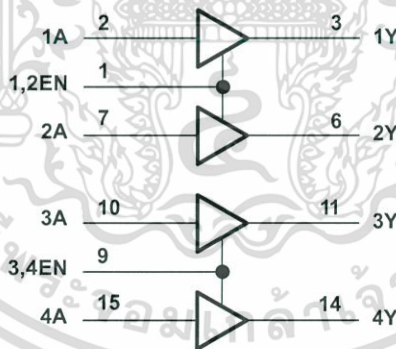
The L293 and L293D are characterized for operation from 0°C to 70°C.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm × 6.35 mm
L293DNE	PDIP (16)	19.80 mm × 6.35 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

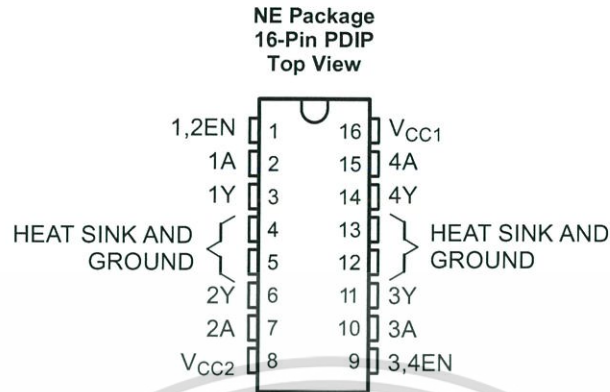
#### Logic Diagram







## 5 Pin Configuration and Functions



**Pin Functions**

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V <sub>CC1</sub>	16	—	5-V supply for internal logic translation
V <sub>CC2</sub>	8	—	Power VCC for drivers 4.5 V to 36 V

## 6 Specifications

### 6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)<sup>(1)</sup>

	MIN	MAX	UNIT
Supply voltage, $V_{CC1}$ <sup>(2)</sup>		36	V
Output supply voltage, $V_{CC2}$		36	V
Input voltage, $V_I$		7	V
Output voltage, $V_O$	-3	$V_{CC2} + 3$	V
Peak output current, $I_O$ (nonrepetitive, $t \leq 5$ ms): L293	-2	2	A
Peak output current, $I_O$ (nonrepetitive, $t \leq 100$ $\mu$ s): L293D	-1.2	1.2	A
Continuous output current, $I_O$ : L293	-1	1	A
Continuous output current, $I_O$ : L293D	-600	600	mA
Maximum junction temperature, $T_J$		150	°C
Storage temperature, $T_{stg}$	-65	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values are with respect to the network ground terminal.

### 6.2 ESD Ratings

		VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge		
	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	$\pm 2000$	V
Charged-device model (CDM), per JEDEC specification JESD22-C101 <sup>(2)</sup>	$\pm 1000$		

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
Supply voltage	$V_{CC1}$	-4.5		7	V
	$V_{CC2}$	$V_{CC1}$		36	
$V_{IH}$	High-level input voltage	$V_{CC1} \leq 7$ V	2.3	$V_{CC1}$	V
		$V_{CC1} \geq 7$ V	2.3	7	V
$V_{IL}$	Low-level output voltage	-0.3 <sup>(1)</sup>		1.5	V
$T_A$	Operating free-air temperature	0		70	°C

- (1) The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

### 6.4 Thermal Information

THERMAL METRIC <sup>(1)</sup>		L293, L293D	UNIT
		NE (PDIP)	
		16 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance <sup>(2)</sup>	36.4	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	16.5	°C/W
$\Psi_{JT}$	Junction-to-top characterization parameter	7.1	°C/W
$\Psi_{JB}$	Junction-to-board characterization parameter	16.3	°C/W

- (1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.
- (2) The package thermal impedance is calculated in accordance with JESD 51-7.

### 6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	L293: I <sub>OH</sub> = -1 A	V <sub>CC2</sub> - 1.8	V <sub>CC2</sub> - 1.4			V
		L293D: I <sub>OH</sub> = -0.6 A					
V <sub>OL</sub>	Low-level output voltage	L293: I <sub>OL</sub> = 1 A			1.2	1.8	V
		L293D: I <sub>OL</sub> = 0.6 A					
V <sub>OKH</sub>	High-level output clamp voltage	L293D: I <sub>OK</sub> = -0.6 A		V <sub>CC2</sub> + 1.3			V
V <sub>OKL</sub>	Low-level output clamp voltage	L293D: I <sub>OK</sub> = 0.6 A			1.3		V
I <sub>IH</sub>	High-level input current	A	V <sub>I</sub> = 7 V		0.2	100	μA
		EN					
I <sub>IL</sub>	Low-level input current	A	V <sub>I</sub> = 0		-3	-10	μA
		EN					
I <sub>CC1</sub>	Logic supply current	I <sub>O</sub> = 0	All outputs at high level		13	22	mA
			All outputs at low level		35	60	
			All outputs at high impedance		8	24	
I <sub>CC2</sub>	Output supply current	I <sub>O</sub> = 0	All outputs at high level		14	24	mA
			All outputs at low level		2	6	
			All outputs at high impedance		2	4	

### 6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V<sub>CC1</sub> = 5 V, V<sub>CC2</sub> = 24 V, T<sub>A</sub> = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t <sub>PLH</sub>	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE	C <sub>L</sub> = 30 pF, See Figure 2		800		ns
		L293DWP, L293N L293DN					
t <sub>PHL</sub>	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE	C <sub>L</sub> = 30 pF, See Figure 2		400		ns
		L293DWP, L293N L293DN					
t <sub>TLH</sub>	Transition time, low-to-high-level output	L293NE, L293DNE	C <sub>L</sub> = 30 pF, See Figure 2		300		ns
		L293DWP, L293N L293DN					
t <sub>THL</sub>	Transition time, high-to-low-level output	L293NE, L293DNE	C <sub>L</sub> = 30 pF, See Figure 2		300		ns
		L293DWP, L293N L293DN					

### 6.7 Typical Characteristics

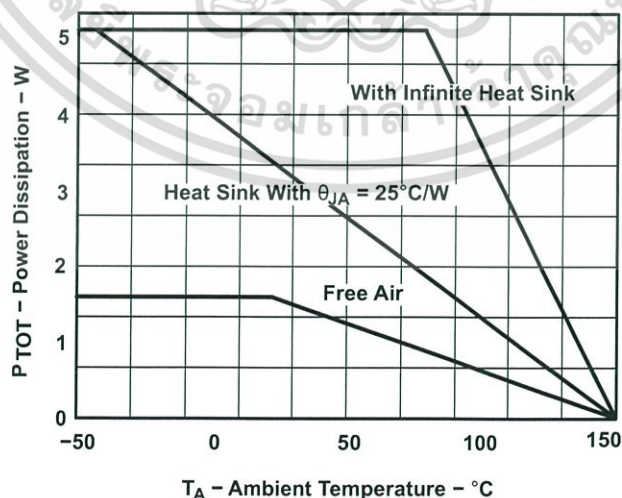
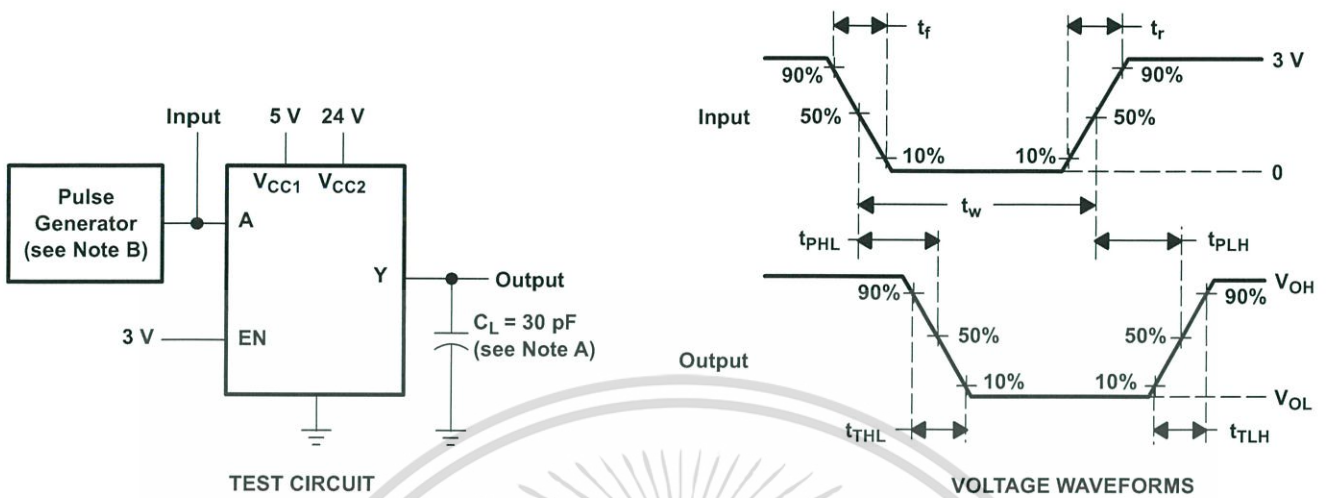


Figure 1. Maximum Power Dissipation vs Ambient Temperature

## 7 Parameter Measurement Information



- NOTES: A.  $C_L$  includes probe and jig capacitance.  
 B. The pulse generator has the following characteristics:  $t_r \leq 10$  ns,  $t_f \leq 10$  ns,  $t_w = 10$   $\mu$ s, PRR = 5 kHz,  $Z_O = 50$   $\Omega$ .

Figure 2. Test Circuit and Voltage Waveforms

## 8 Detailed Description

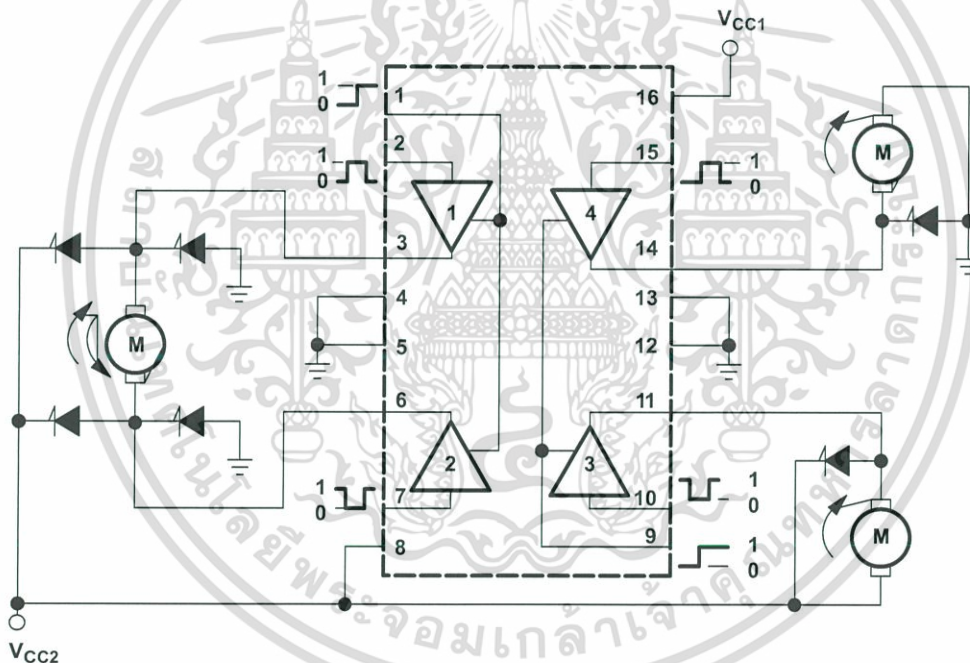
### 8.1 Overview

The L293 and L293D are quadruple high-current half-H drivers. These devices are designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. On the L293D, these diodes are integrated to reduce system complexity and overall system size. A  $V_{CC1}$  terminal, separate from  $V_{CC2}$ , is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C.

### 8.2 Functional Block Diagram



Output diodes are internal in L293D.

### 8.3 Feature Description

The L293x has TTL-compatible inputs and high voltage outputs for inductive load driving. Current outputs can get up to 2 A using the L293.

### 8.4 Device Functional Modes

Table 1 lists the functional modes of the L293x.

Table 1. Function Table (Each Driver)<sup>(1)</sup>

INPUTS <sup>(2)</sup>		OUTPUT (Y)
A	EN	
H	H	H
L	H	L
X	L	Z

- (1) H = high level, L = low level, X = irrelevant, Z = high impedance (off)
- (2) In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

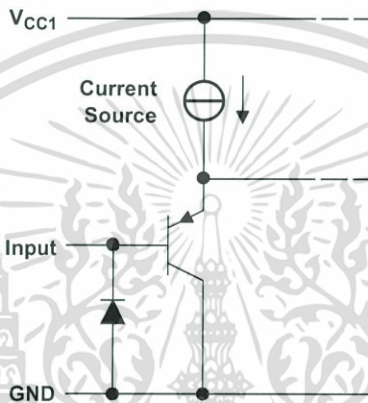


Figure 3. Schematic of Inputs for the L293x

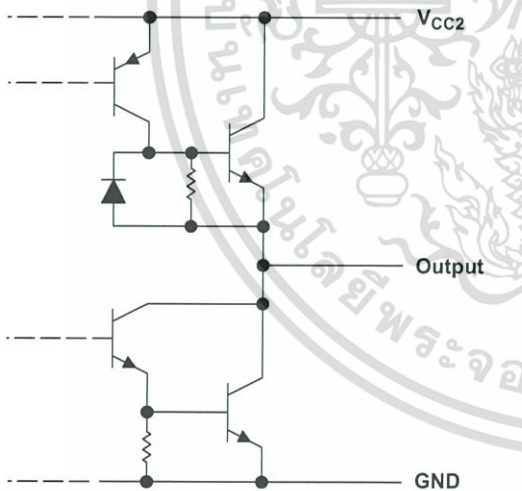


Figure 4. Schematic of Outputs for the L293

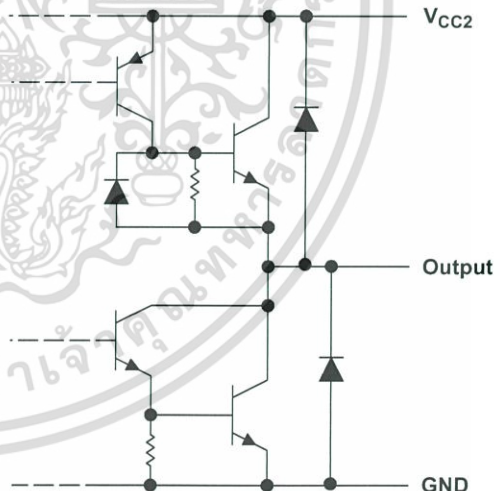


Figure 5. Schematic of Outputs for the L293D

## 9 Application and Implementation

### NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

### 9.1 Application Information

A typical application for the L293 device is driving a two-phase motor. Below is an example schematic displaying how to properly connect a two-phase motor to the L293 device.

Provide a 5-V supply to  $V_{CC1}$  and valid logic input levels to data and enable inputs.  $V_{CC2}$  must be connected to a power supply capable of supplying the needed current and voltage demand for the loads connected to the outputs.

### 9.2 Typical Application

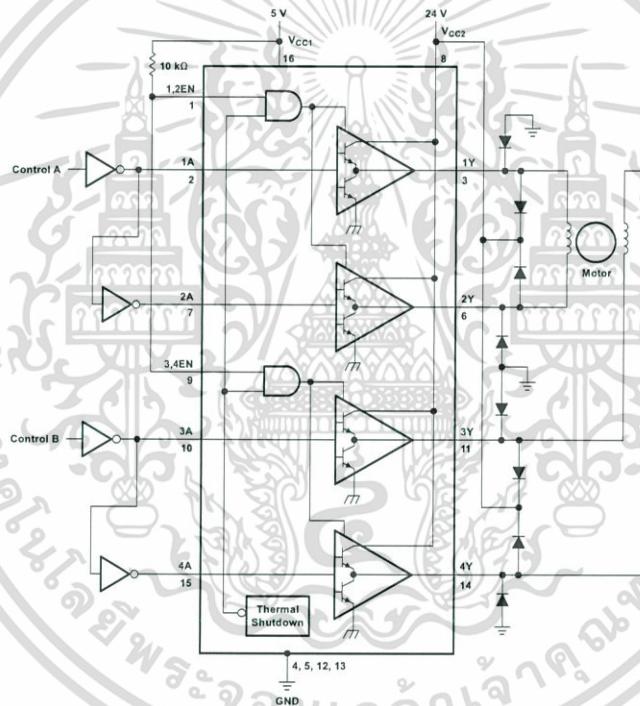


Figure 6. Two-Phase Motor Driver (L293)

#### 9.2.1 Design Requirements

The design techniques in the application above as well as the applications below should fall within the following design requirements.

1.  $V_{CC1}$  should fall within the limits described in the *Recommended Operating Conditions*.
2.  $V_{CC2}$  should fall within the limits described in the *Recommended Operating Conditions*.
3. The current per channel should not exceed 1 A for the L293 (600mA for the L293D).

#### 9.2.2 Detailed Design Procedure

When designing with the L293 or L293D, careful consideration should be made to ensure the device does not exceed the operating temperature of the device. Proper heatsinking will allow for operation over a larger range of current per channel. Refer to the *Power Supply Recommendations* as well as the *Layout Example*.

Typical Application (continued)

9.2.3 Application Curve

Refer to *Power Supply Recommendations* for additional information with regards to appropriate power dissipation. Figure 7 describes thermal dissipation based on Figure 14.

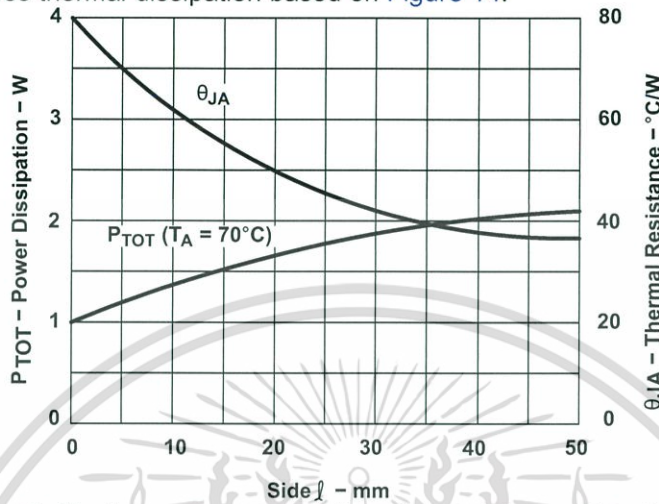


Figure 7. Maximum Power and Junction vs Thermal Resistance

9.3 System Examples

9.3.1 L293D as a Two-Phase Motor Driver

Figure 8 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.

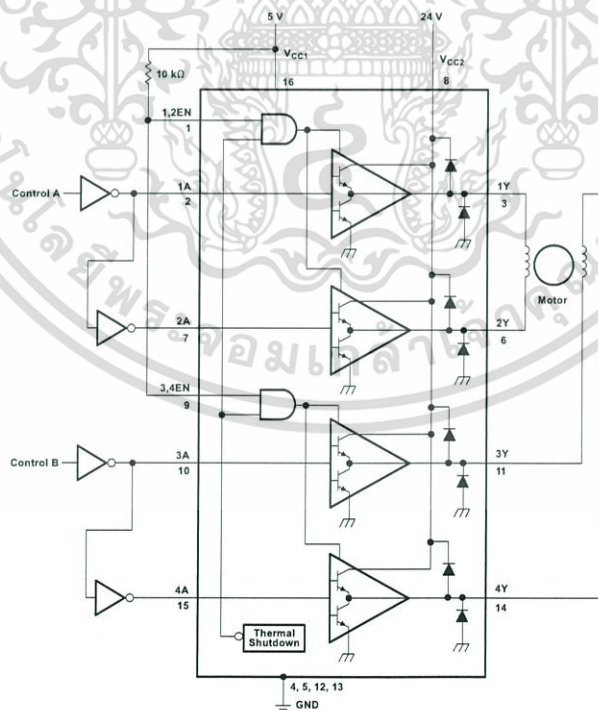


Figure 8. Two-Phase Motor Driver (L293D)



### System Examples (continued)

#### 9.3.2 DC Motor Controls

Figure 9 and Figure 10 below depict a typical setup for using the L293 device as a controller for DC motors. Note that the L293 device can be used as a simple driver for a motor to turn on and off in one direction, and can also be used to drive a motor in both directions. Refer to the function tables below to understand unidirectional vs bidirectional motor control. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.

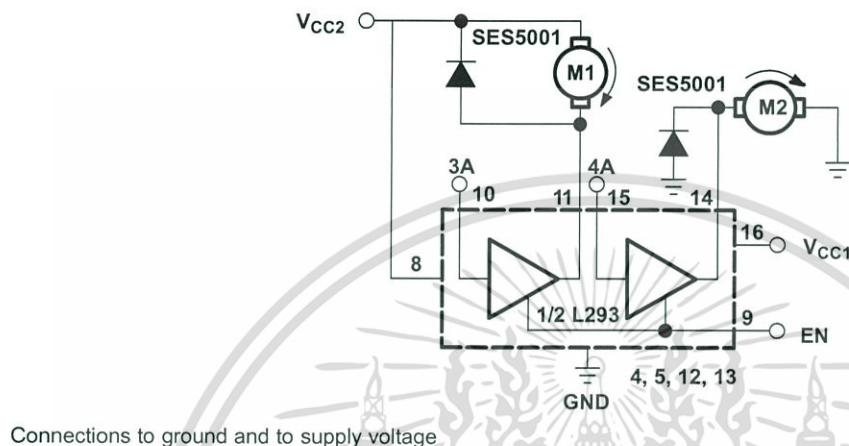


Figure 9. DC Motor Controls

Table 2. Unidirectional DC Motor Control

EN	3A	M1 <sup>(1)</sup>	4A	M2
H	H	Fast motor stop	H	Run
H	L	run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

(1) L = low, H = high, X = don't care

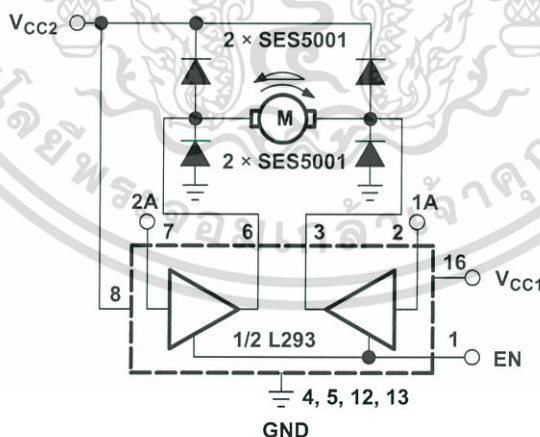


Figure 10. Bidirectional DC Motor Control

Table 3. Bidirectional DC Motor Control

EN	1A	2A	FUNCTION <sup>(1)</sup>
H	L	H	Turn right
H	H	L	Turn left

(1) L = low, H = high, X = don't care

Table 3. Bidirectional DC Motor Control (continued)

EN	1A	2A	FUNCTION <sup>(1)</sup>
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

9.3.3 Bipolar Stepping-Motor Control

Figure 11 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.

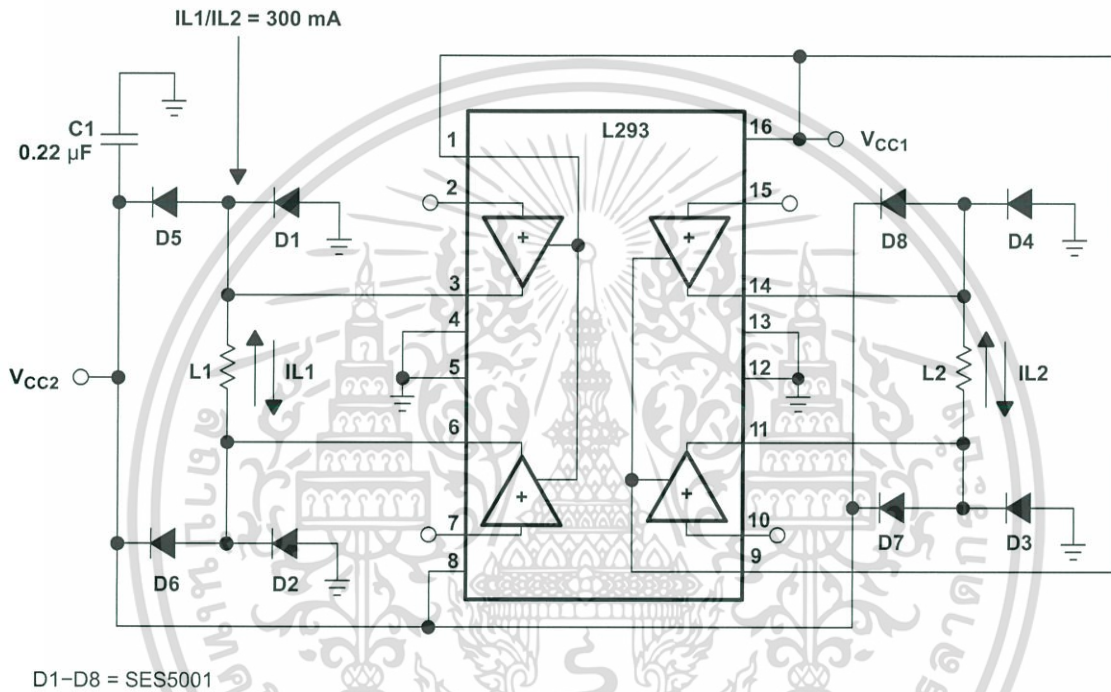


Figure 11. Bipolar Stepping-Motor Control

## 10 Power Supply Recommendations

$V_{CC1}$  is  $5\text{ V} \pm 0.5\text{ V}$  and  $V_{CC2}$  can be same supply as  $V_{CC1}$  or a higher voltage supply with peak voltage up to 36 V. Bypass capacitors of 0.1  $\mu\text{F}$  or greater should be used at  $V_{CC1}$  and  $V_{CC2}$  pins. There are no power up or power down supply sequence order requirements.

Properly heatsinking the L293 when driving high-current is critical to design. The  $R_{thj-amp}$  of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heat sink.

Figure 14 shows the maximum package power  $PTOT$  and the  $\theta_{JA}$  as a function of the side of two equal square copper areas having a thickness of 35  $\mu\text{m}$  (see Figure 14). In addition, an external heat sink can be used (see Figure 12).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not exceed 12 seconds.

The external heatsink or printed circuit board copper area must be connected to electrical ground.

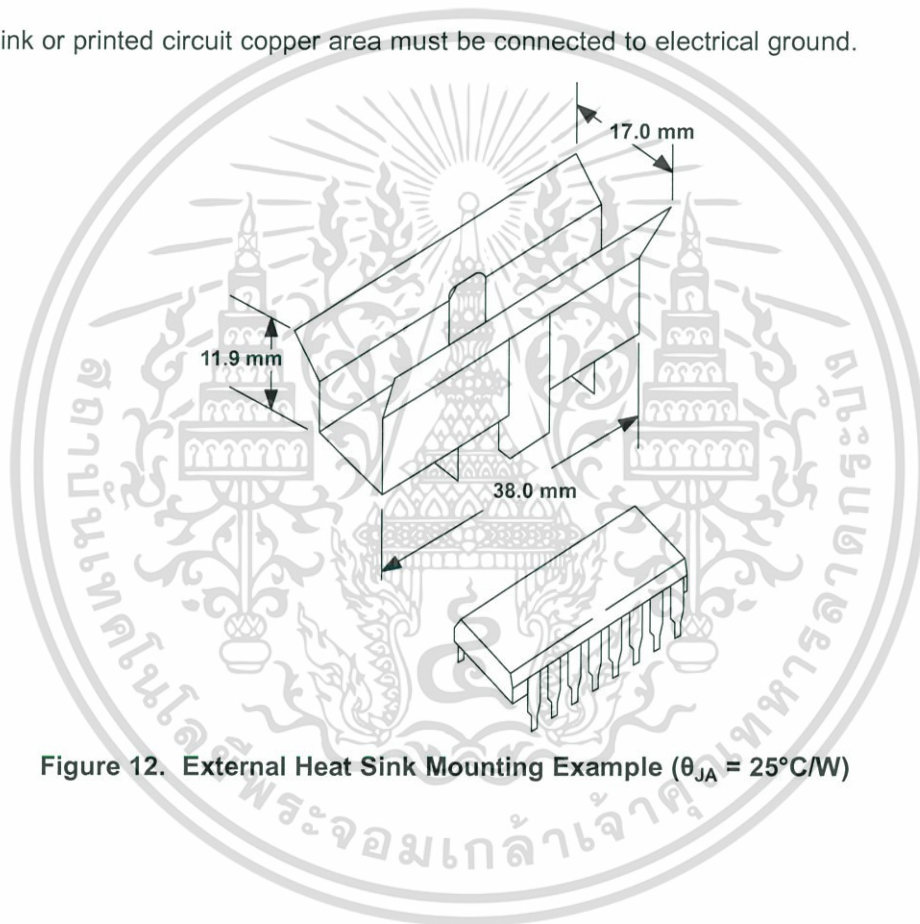


Figure 12. External Heat Sink Mounting Example ( $\theta_{JA} = 25^\circ\text{C/W}$ )

## 11 Layout

### 11.1 Layout Guidelines

Place the device near the load to keep output traces short to reduce EMI. Use solid vias to transfer heat from ground pins to ground plane of the printed-circuit-board.

### 11.2 Layout Example

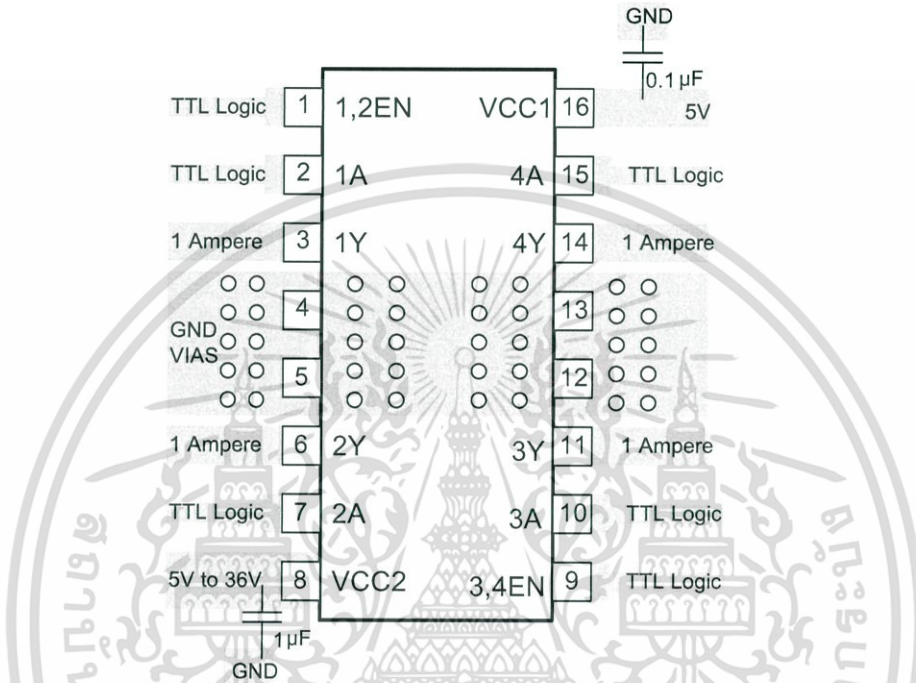


Figure 13. Layout Diagram

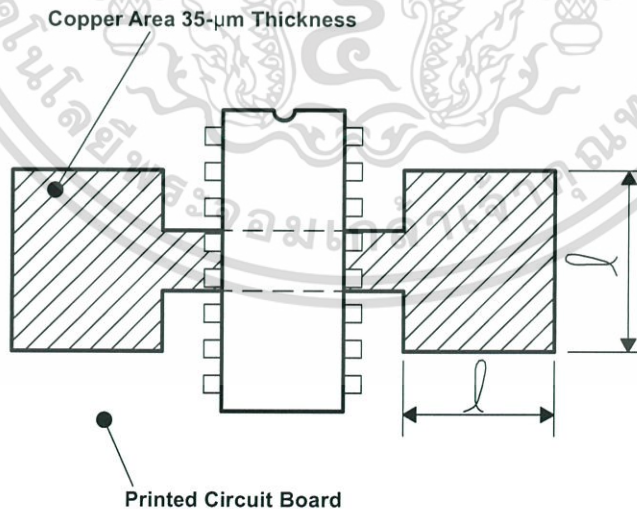


Figure 14. Example of Printed-Circuit-Board Copper Area (Used as Heat Sink)

## 12 Device and Documentation Support

### 12.1 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

**Table 4. Related Links**

PARTS	PRODUCT FOLDER	SAMPLE & BUY	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
L293	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>
L293D	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>	<a href="#">Click here</a>

### 12.2 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's [Terms of Use](#).

**TI E2E™ Online Community** *TI's Engineer-to-Engineer (E2E) Community*. Created to foster collaboration among engineers. At [e2e.ti.com](http://e2e.ti.com), you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

**Design Support** *TI's Design Support* Quickly find helpful E2E forums along with design support tools and contact information for technical support.

### 12.3 Trademarks

E2E is a trademark of Texas Instruments. All other trademarks are the property of their respective owners.

### 12.4 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

### 12.5 Glossary

**SLYZ022** — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

## 13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

## PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
L293DNE	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	<a href="#">Samples</a>
L293DNEE4	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	<a href="#">Samples</a>
L293NE	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type	0 to 70	L293NE	<a href="#">Samples</a>
L293NEE4	ACTIVE	PDIP	NE	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type	0 to 70	L293NE	<a href="#">Samples</a>

(1) The marketing status values are defined as follows:

**ACTIVE:** Product device recommended for new designs.

**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.

**OBSOLETE:** TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

**TBD:** The Pb-Free/Green conversion plan has not been defined.

**Pb-Free (RoHS):** TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

**Pb-Free (RoHS Exempt):** This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

**Green (RoHS & no Sb/Br):** TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

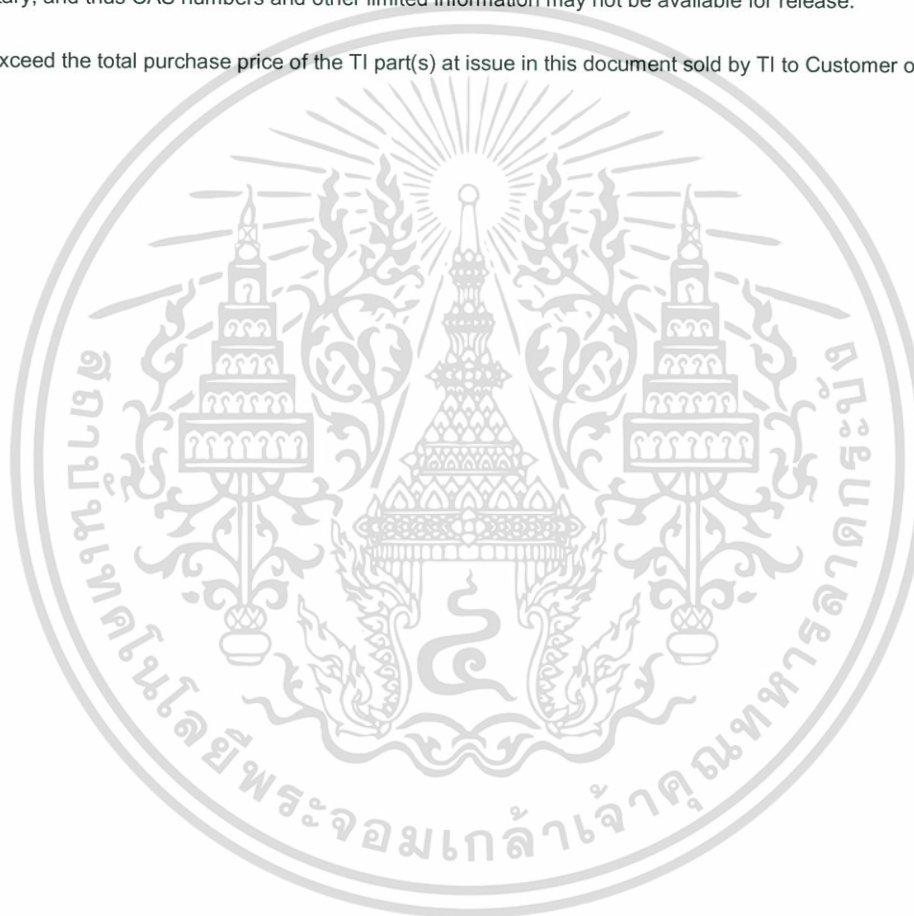
(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

**Important Information and Disclaimer:** The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.



## IMPORTANT NOTICE

Texas Instruments Incorporated (TI) reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

TI's published terms of sale for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>) apply to the sale of packaged integrated circuit products that TI has qualified and released to market. Additional terms may apply to the use or sale of other types of TI products and services.

Reproduction of significant portions of TI information in TI data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions. Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyers and others who are developing systems that incorporate TI products (collectively, "Designers") understand and agree that Designers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Designers have full and exclusive responsibility to assure the safety of Designers' applications and compliance of their applications (and of all TI products used in or for Designers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Designer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Designer agrees that prior to using or distributing any applications that include TI products, Designer will thoroughly test such applications and the functionality of such TI products as used in such applications.

TI's provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using TI Resources in any way, Designer (individually or, if Designer is acting on behalf of a company, Designer's company) agrees to use any particular TI Resource solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

Designer is authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY DESIGNER AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Unless TI has explicitly designated an individual product as meeting the requirements of a particular industry standard (e.g., ISO/TS 16949 and ISO 26262), TI is not responsible for any failure to meet such industry standard requirements.

Where TI specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Designers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any TI products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death (e.g., life support, pacemakers, defibrillators, heart pumps, neurostimulators, and implantables). Such equipment includes, without limitation, all medical devices identified by the U.S. Food and Drug Administration as Class III devices and equivalent classifications outside the U.S.

TI may expressly designate certain products as completing a particular qualification (e.g., Q100, Military Grade, or Enhanced Product). Designers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Designers' own risk. Designers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Designer will fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2017, Texas Instruments Incorporated

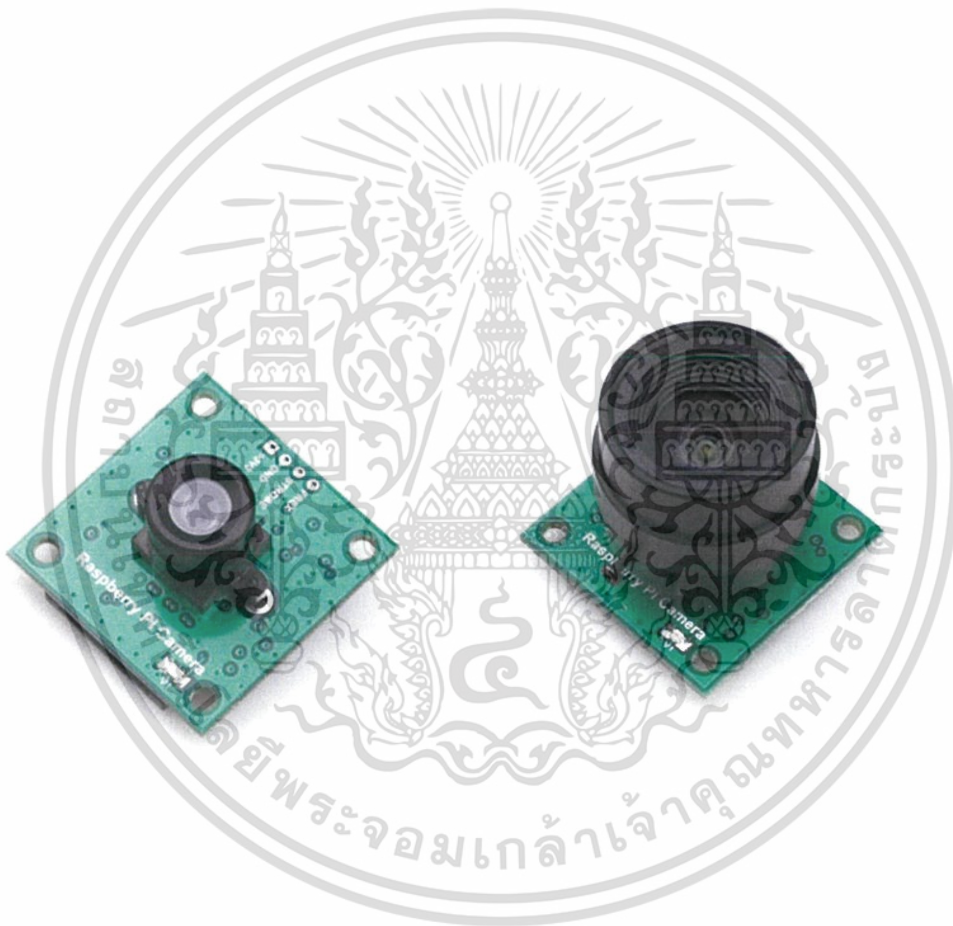
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# ArduCam

## Raspberry Pi Camera Module 1/4-Inch 5-Megapixel Module Datasheet

Rev 1.0, Feb 2015



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Table of Contents

1	Introduction.....	2
2	Block Diagram.....	2
3	Features.....	3
4	Key Specifications.....	3
5	Application.....	4
6	Pin Definition.....	4
7	Lens Options.....	5
8	Mechanical Dimension.....	6



# 1 Introduction

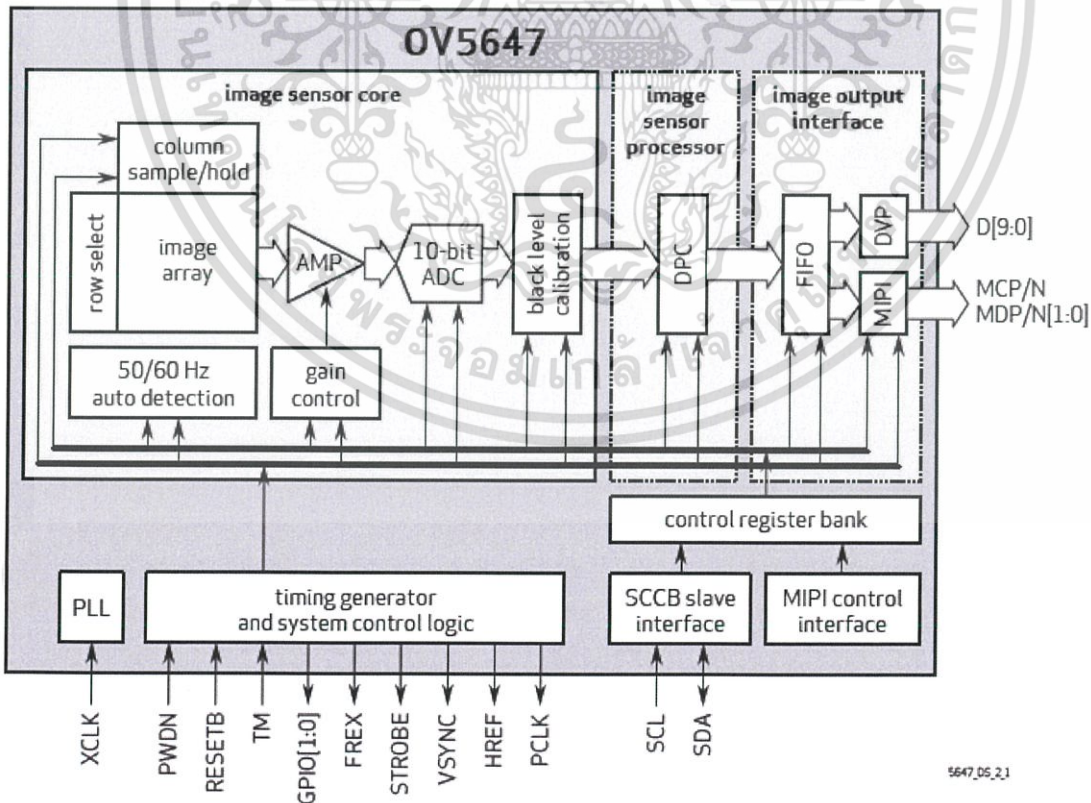
In order to meet the increasing need of Raspberry Pi compatible camera modules. The ArduCAM team now released a revision C add-on camera module for Raspberry Pi which is fully compatible with official one. It optimizes the optical performance than the previous Pi cameras, and give user a much clear and sharp image. Also it provides the FREX and STROBE signals which can be used for multi-camera synchronize capture with proper camera driver firmware.

It attaches to Raspberry Pi by way of one of the two small sockets on the board upper surface. This interface uses the dedicated CSI interface, which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

The board itself is tiny, at around 36mm x 36mm. The highlight of our module is that the Lens is replaceable compared to official one, making it perfect for mobile or other applications where size and image quality are important. It connects to Raspberry Pi by way of a short ribbon cable. The camera is connected to the BCM2835/BCM2836 processor on the Pi via the CSI bus, a higher bandwidth link which carries pixel data from the camera back to the processor. This bus travels along the ribbon cable that attaches the camera board to the Pi.

The sensor itself has a native resolution of 5 megapixel, and has a fixed focus lens onboard. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video.

# 2 Block Diagram



Note: Raspberry Pi camera module only support MIPI interface, it doesn't support DVP interface..

### 3 Features

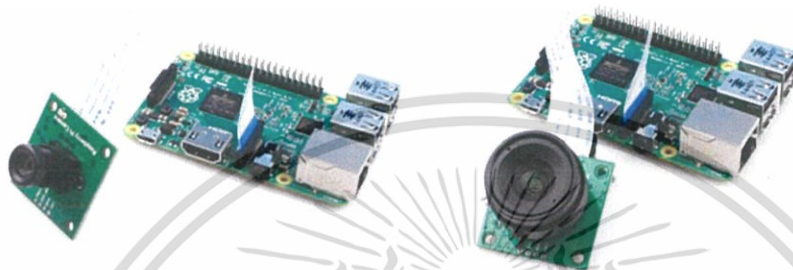
- High-Definition video camera for Raspberry Pi Model A/B/B+ and Raspberry Pi 2
- Omnivision OV5647 sensor in a fixed-focus module with replaceable Lens
- Lens holder: M12x0.5 , CS mount or C mount
- 5MPixel sensor
- Integral IR filter
- Still picture resolution: 2592 x 1944
- Max video resolution: 1080p
- Max frame rate: 30fps
- Support FREX/ STROBE feature
- Size: 36 x 36 mm
- 15 cm flat ribbon cable to 15-pin MIPI Camera Serial Interface (CSI) connector

### 4 Key Specifications

- active array size: 2592 x 1944
- power supply:
  - core: 1.5V  $\pm$  5% (with embedded 1.5V regulator)
  - analog: 2.6 ~ 3.0V (2.8V typical)
  - I/O: 1.7V ~ 3.0V
- power requirements:
  - active: TBD
  - standby: TBD
- temperature range:
  - operating: -30°C to 70°C (see [table 8-2](#))
  - stable image: 0°C to 50°C (see [table 8-2](#))
- output formats: 8-/10-bit RGB RAW output
- lens size: 1/4"
- lens chief ray angle: 24° (see [figure 10-2](#))
- input clock frequency: 6~27 MHz
- S/N ratio: TBD
- dynamic range: TBD
- maximum image transfer rate:
  - QSXGA (2592 x 1944): 15 fps
  - 1080p: 30 fps
  - 960p: 45 fps
  - 720p: 60 fps
  - VGA (640 x 480): 90 fps
  - QVGA (320 x 240): 120 fps
- sensitivity: TBD
- shutter: rolling shutter / global shutter
- maximum exposure interval: 1968 x t<sub>ROW</sub>
- pixel size: 1.4  $\mu$ m x 1.4  $\mu$ m
- well capacity: TBD
- dark current: TBD
- fixed pattern noise (FPN): TBD
- image area: 3673.6  $\mu$ m x 2738.4  $\mu$ m
- die dimensions: 5520  $\mu$ m x 4700  $\mu$ m

## 5 Application

- Cellular phones
- PDAs
- Toys
- Other battery-powered products
- Can be used in Raspberry Pi, ARM, DSP, FPGA platforms



## 6 Pin Definition

Table 1 P1 Connector Pin Definition

Pin No.	PIN NAME	TYPE	DESCRIPTION
1	DGND	Ground	Power ground
2	CAM_D0_N	Output	MIPI data lane0 negative output
3	CAM_D0_P	Output	MIPI data lane0 positive output
4	DGND	Ground	Power ground
5	CAM_D1_N	Output	MIPI data lane1 negative output
6	CAM_D1_P	Output	MIPI data lane1 positive output
7	DGND	Ground	Power ground
8	CAM_C_N	Output	MIPI clock negative output
9	CAM_C_P	Output	MIPI clock positive output
10	DGND	Ground	Power ground
11	POWER_EN	Input	Camera module power enable active high
12	LED_EN	Input	Reserved
13	SCL	Input	Two-Wire Serial Interface Clock
14	SDA	Bi-directional	Two-Wire Serial Interface Data I/O
15	+3.3V	POWER	3.3v Power supply

Table 2 P2 Connector Pin Definition

Pin No.	PIN NAME	TYPE	DESCRIPTION
1	+3.3V	POWER	3.3v Power supply
2	DGND	Ground	Power ground
3	STROBE	Output	Strobe output
4	FREX	input	Frame exposure control

## 7 Lens Options

The Raspberry Pi camera shipped with default LS-40136 (M12x0.5 mount) and LS-6018 (CS mount), Lenses specification list as follows. Please contact us [admin@arducam.com](mailto:admin@arducam.com) for more lens options.

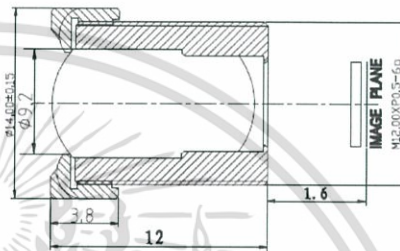
### LS-40136 Lens Specification

A. Specification: **LS-40136**

1. sensor size: 1/4"
2. focal length(EFL): **3.2 mm**
3. F/NO(inifiniton): **2.0**
4. back focal length: **1.6 mm**

B. Layout

6. Field of view:
  - Diagonal, **85°**;
  - Horzongtal, **63.7°**;
  - Vertical, **70°**;
7. Thread size: **M12\*P0.5**
8. Element: **5B+IR**

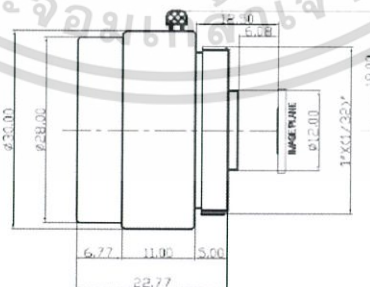


### LS-6018 Lens Specification

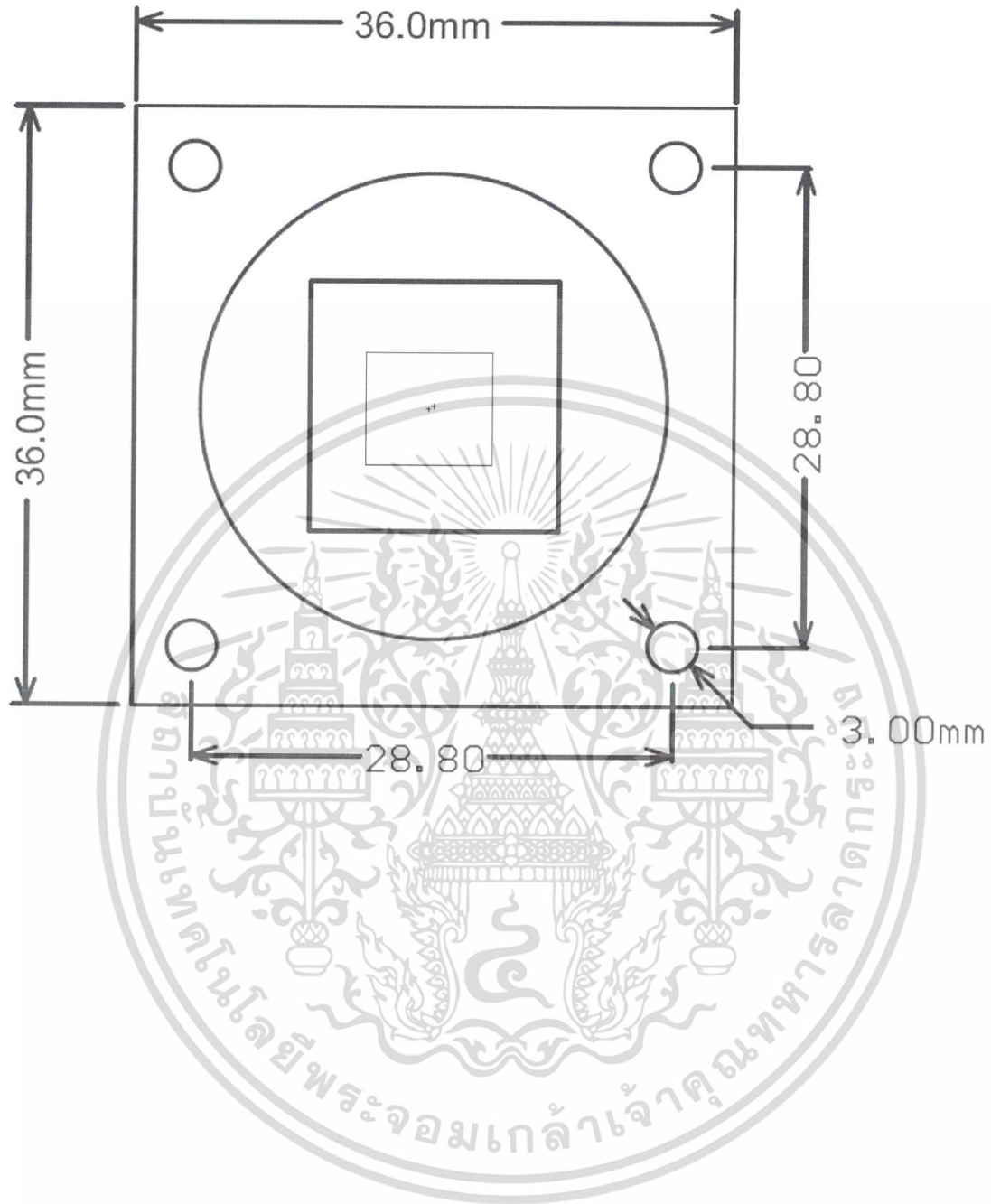
### 技术参数

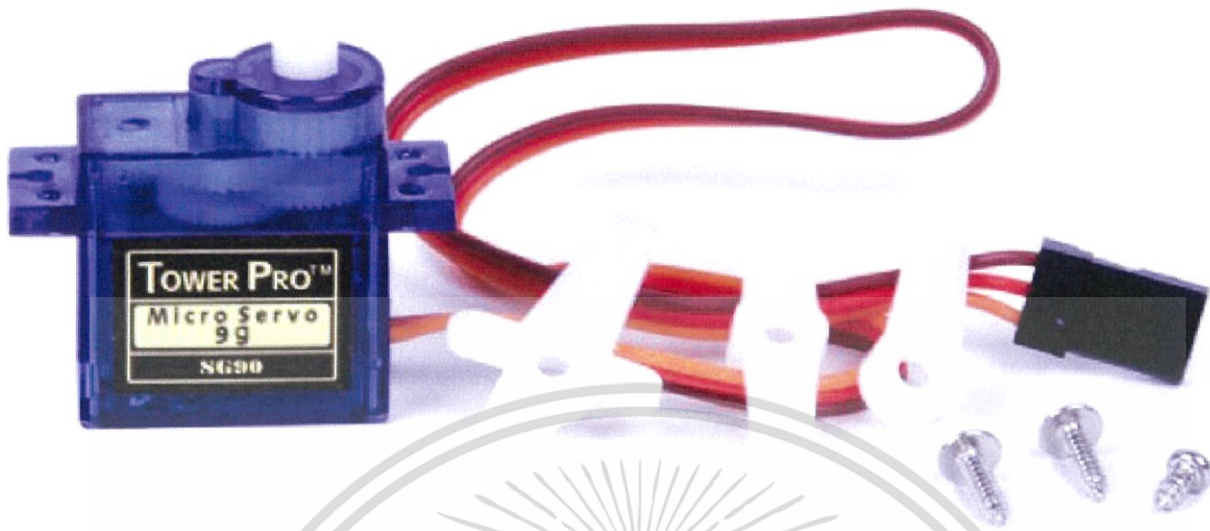
Technical parameters

型号 Model No.	LS-6018CS	视场角 Field of View	68°
焦距 Focal Length	6.0MM	外型尺寸 Dimensions	Φ28*24.2mm
通光口径 Aperture(F)	1.4	近摄距离 M.O.D(m)	0.1
接口 Mount	CS	净重 Weight(g)	29.0
靶面尺寸 Format	1/2.7"	备注 Remarks	<b>Metal</b>

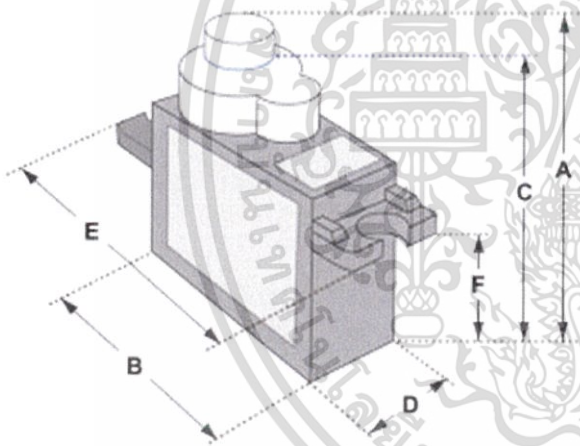


## 8 Mechanical Dimension





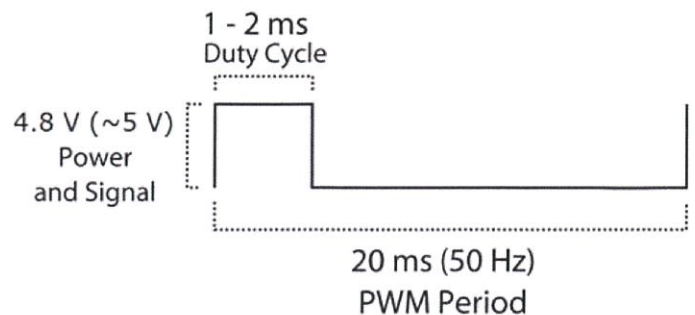
Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

PWM = Orange (⌈⌋)  
 Vcc = Red (+)  
 Ground = Brown (-)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# HC-SR04 User Guide

## 1. Ultrasonic Distance Measurement Principles

The transmitter emits a 8 bursts of an directional 40KHz ultrasonic wave when triggered and starts a timer. Ultrasonic pulses travel outward until they encounter an object, The object causes the the wave to be reflected back towards the unit. The ultrasonic receiver would detect the reflected wave and stop the stop timer. The velocity of the ultrasonic burst is 340m/sec. in air. Based on the number of counts by the timer, the distance can be calculated between the object and transmitter The TRD Measurement formula is expressed as:  $D = C \times T$  which is know as the time/rate/distance measurement formula where D is the measured distance, and R is the propagation velocity (Rate) in air (speed of sound) and T represents time. In this application T is divided by 2 as T is double the time value from transmitter to object back to receiver.

## 2. Product Features

### Features

- Stable performance (Xtal.)
- Accurate distance measurement
- High-density SMD Board
- Close Range (2cm)

### Uses

- Robotics barrier
- Object distance measurement
- Level detection
- Security systems
- Vehicle detection/avoidance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. Product Views



### 4. Module Pin Assignments

	Pin Symbol	Pin Function Description
1	VCC	5V power supply
2	Trig	Trigger Input pin
3	Echo	Receiver Output pin
4	GND	Power ground

### 5. Electrical Specifications

#### WARARNING

Do Not connect Module with Power Applied! Always apply power after connecting Connect "GND" Terminal first

Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	5VDC
Operating Current	15mA
Operating Frequency	40KHz
Max. Range	4m
Nearest Range	2cm
Measuring Angle	15 Degrees
Input Trigger Signal	10us min. TTL pulse
Output Echo Signal	TTL level signal, proportional to distance
Board Dimensions	1-13/16" X 13/16" X 5/8"
Board Connections	4 X 0.1" Pitch Right Angle Header Pins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

## 6. Module Operation

Set Trig and Echo Low to initialize module. Place a minimum 10us High level pulse to "Trigger" (module will automatically send eight 40KHz acoustic bursts). At the same time, Gate the microcontroller timer to start timing.

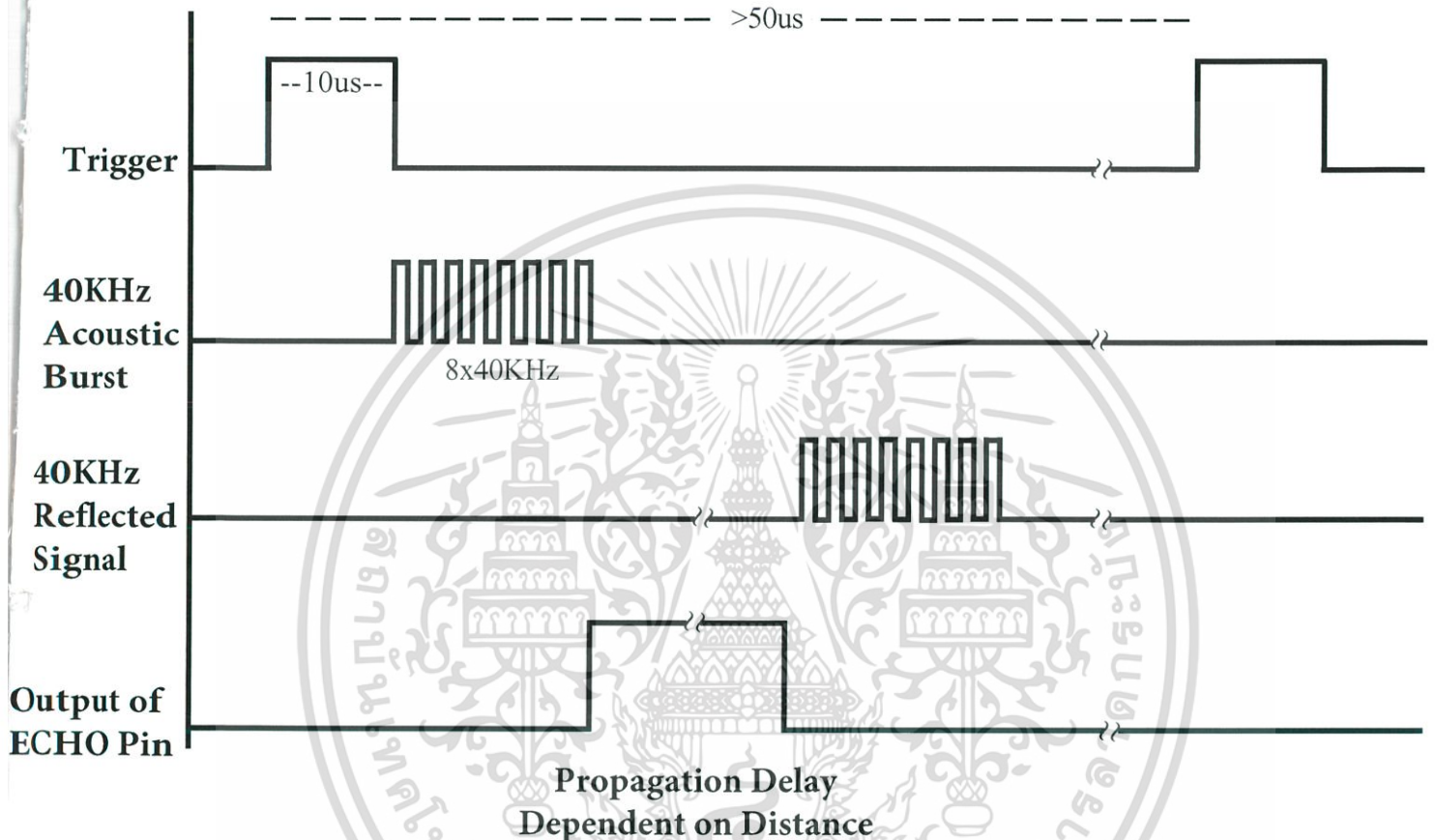
Wait to capture the rising edge output of ECHO port to stop the timer. Now read the time of the counter, which is the ultrasonic propagation time in the air. According to the formula: Distance = (ECHO high level time X ultrasonic velocity (Speed of Sound in air 340m/sec) / 2, you can calculate the distance to the obstacle.

For best results and maximum range, the Object should be larger than 0.5M<sup>2</sup> the nearer the target object, the smaller it may be



## 7. Module Timing

### HC-SR04 ULTRASONIC MODULE



**Trigger** 10µs min. start measurement from microcontroller.

Max Rep. Rate: 50µs

**ECHO** Output pulse to microcontroller, width is the time from last of 8 40KHz bursts to detected reflected signal (microcontroller Timer gate signal)

Distance in cm = echo pulse width in µs/58

Distance in inch = echo pulse width in µs/148

Information obtained from or supplied by Mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้