

รถจำลองขับเคลื่อนอัตโนมัติ

Automatically Car



ณัฐภูมิภัทร์ เขมะทัตส์

ยุทธนา พิมเสน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2561

แบบจำลองรถไร้คนขับ ขับเคลื่อนอัตโนมัติ

Automatic Car

ณัฐภูมิภัทร์ เขมะทัสสี

ยุทธนา พิมพ์เสน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

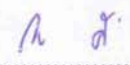
ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2561


..... ประธานกรรมการสอบปริญญานิพนธ์
(อาจารย์ตะวัน ภูรัต)


..... กรรมการ
(ดร.กาญจนา ศิลาวราเวช)


..... กรรมการผู้ทรงคุณวุฒิ
(พล.อ.ต.ผศ.ดร.พารณ์ สงวนโกศล)


..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ดร.กาญจนา ศิลาวราเวช)


..... คณบดีคณะวิศวกรรมศาสตร์
(ผู้ช่วยศาสตราจารย์สราวุธ วรสุมนต์)

หัวข้อปริญญานิพนธ์	รถจำลองขับเคลื่อนอัตโนมัติ	
หน่วยกิตของปริญญานิพนธ์	2 หน่วยกิต	
โดย	นายณัฐภูมิ นิมะทัสสี	5804000018
	นายยุทธนา พิมพ์เสน	5804000025
อาจารย์ที่ปรึกษา	อาจารย์ตะวัน ภูรัต	
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2561	

บทคัดย่อ

บทความนี้นำเสนอการพัฒนาการพัฒนารถจำลองควบคุมการเคลื่อนที่ด้วยกล้องโดยใช้เส้นขอบทางสองเส้น กล้องเว็บแคมที่ 2 จับภาพวิดีโอจากส่วนด้านหน้าของรถและใช้ชุดคำสั่งจากคลังคำสั่งโอเพ่นซีวีเช่นการเขียนอัลกอริทึม แคนดีและฮอทไลน์ในการแปลงภาพให้เป็นระดับสีเทา การจับขอบภาพและเส้นตรงตามลำดับ ค่าความแตกต่างระหว่างค่าเฉลี่ยของจุดบนเส้นสองเส้นเทียบกับจุดคงที่จะทำให้สามารถคำนวณได้ว่ารถจะบังคับรถให้เดินหน้า เลี้ยวซ้าย เลี้ยวขวา หรือหยุด รถจำลองขนาดเล็กนี้ถูกนำไปทดสอบบนพื้นที่สนามขนาด 2x2 ตารางเมตร ผลการทดสอบแสดงให้เห็นว่าอัลกอริทึมที่ใช้สามารถขับเคลื่อนรถได้ แต่เนื่องจากมีความผิดพลาดเล็กน้อยหลายประการที่ทำให้รถตกขอบทางจึงมีความจำเป็นต้องมีเซนเซอร์อื่นๆเช่นอินฟราเรดหรือเลเซอร์นำทางติดตั้งเพิ่มเติมเพื่อป้องกันรถวิ่งออกนอกถนน

คำสำคัญ การมองเห็นของเครื่องจักร / การเรียนรู้ของเครื่องจักร/ ยานยนต์ไร้คนขับ / กล้องถ่ายภาพดิจิทัล / ภาษาไพทอน

อาจารย์ที่ปรึกษา.....

(อาจารย์ตะวัน ภูรัต)

26.../.../...62...

Project Title	AUTOMATICALLY CAR	
Project Credit	2 Credits	
By	Mr. Natthaphumin	Khemathussee
	Mr. Yuttana	Pimsean
Project Advisor	Mr. Tawan	Phurat
Degree	Bachelor of Engineering	
Academic Year	2018	

Abstract

A development of a small autonomous car model which is driven by camera on two of continuous edge lines has been presented in this paper in order to study some suitable empirical driving algorithm. In addition, Camera PI v2 captures the video from the front of car and uses some libraries from OpenCV such Gaussian algorithm, Candy and Hough line to transform picture to Grayscale, edged-detection, and straight line respectively. The different value of averages of two straight line and constant point can determine the action to drive the car such go straight, turn left, turn right or stop. The small autonomous car was tested on 2x2 square meter circuit and the results illustrated that the empirical algorithm can drive it. Unfortunately, a little mistake has been found. To prevent this problem, it requires more sensor such infra-red, or laser guild-line to avoid the car run out of the road.

Keywords : Machine Vision / Machine learning / Driverless Vehicle / Digital Camera / Python Language

Approved by

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ตะวัน ภูริต อาจารย์ที่ปรึกษา ซึ่งกรุณาสละเวลา ให้ความรู้และคำแนะนำตลอดการทำโครงการ

ขอขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสยาม ตลอดจนผู้เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามมา ณ ที่นี้

ท้ายที่สุด ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ผู้เป็นที่รัก ผู้ให้กำลังใจและให้โอกาส การศึกษาอันมีค่ายิ่ง



คณะผู้จัดทำโครงการ

สารบัญ

	หน้า
บทคัดย่อ (ภาษาไทย)	๗
บทคัดย่อ (ภาษาอังกฤษ)	๘
กิตติกรรมประกาศ	๙
สารบัญตาราง	๑๑
สารบัญรูป	๑๓
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 คำสำคัญ (Keyword) สำหรับโครงงาน	1
1.3 วัตถุประสงค์ในการทำโครงงาน	1
1.4 ขอบเขตและความสามารถของโครงงาน	2
1.5 ประโยชน์คาดว่าจะได้รับ	2
1.6 เครื่องมือ เครื่องวัด และอุปกรณ์ที่ใช้	2
1.7 แผนการดำเนินงาน	3
บทที่ 2 การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง	4
2.1 การออกแบบโครงสร้างระบบ	4
2.2 OpenCV	7
2.3 Raspberry Pi	8
2.4 โมดูลควบคุมมอเตอร์	11
2.5 เกียร์มอเตอร์และล้อรถ สำหรับ Smart Robot Car	14

สารบัญ (ต่อ)

	หน้า
2.6 Raspbian OS	15
2.7 Python	16
2.8 การควบคุมมอเตอร์ (Motor Control)	17
บทที่ 3 การวิเคราะห์และการออกแบบ	19
3.1 การออกแบบระบบ	19
3.2 ภาพรวมการทำงานของโครงงาน	21
3.3 หลักการทำงานของระบบ	22
3.4 การสร้างรถจำลองอัตโนมัติ	23
3.5 การออกแบบระบบไฟฟ้าของรถจำลองอัตโนมัติ	25
3.6 การหาขอบ (Edge Detection)	25
บทที่ 4 การทดลอง	27
4.1 การทดลองส่งคำสั่งบอร์ดราสเบอรี่พายเพื่อควบคุมมอเตอร์ของล้อรถให้เคลื่อนที่	27
4.2 การกำหนดไฟฟ้าภายใน	28
4.3 การทดลองการเคลื่อนที่ทางตรงของรถจำลองอัตโนมัติ	28
4.4 การทดลองการเคลื่อนที่เลี้ยวซ้ายของรถจำลองอัตโนมัติ	30
4.5 การทดลองการเคลื่อนที่เลี้ยวขวาของรถจำลองอัตโนมัติ	31
4.6 การทดลองถ่ายภาพจากกล้องคาเมราพาย (Camera Pi)	32

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะ	37
5.1 สรุปผลการดำเนินโครงการ	37
5.2 ปัญหาการดำเนินโครงการ	37
5.3 ข้อเสนอแนะ	37
บรรณานุกรม	38
ภาคผนวก	39
ประวัติผู้จัดทำ	44



สารบัญตาราง

	หน้า
ตารางที่ 1.1 การวางแผนการดำเนินงาน	3
ตารางที่ 2.1 การทำงานมอเตอร์ A	13
ตารางที่ 2.2 การทำงานมอเตอร์ B	13
ตารางที่ 3.1 คุณสมบัติของรถจำลองอัตโนมัติ	23
ตารางที่ 3.2 คุณสมบัติของรถจำลองอัตโนมัติ	24
ตารางที่ 4.1 การกำหนด Output สำหรับการเชื่อมต่อโมดูลควบคุมมอเตอร์ L298N	27
ตารางที่ 4.2 ทดลองการเคลื่อนที่ทางตรงแต่ละครั้ง	29
ตารางที่ 4.3 ทดลองการเคลื่อนที่เลี้ยวซ้ายแต่ละครั้ง	31
ตารางที่ 4.4 ทดลองการเคลื่อนที่เลี้ยวขวาแต่ละครั้ง	32

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 ภาพแสดงขั้นตอนการทำงาน	6
รูปที่ 2.2 ไอโฟนซีวี	7
รูปที่ 2.3 บอร์ด Raspberry Pi	8
รูปที่ 2.4 แสดงรายละเอียดความแตกต่างของโมดูล A และ B	9
รูปที่ 2.5 รูปลักษณะโมเดล A และ B	9
รูปที่ 2.6 พอร์ต GPIO Raspberry Pi Model A & B (Revision 1)	10
รูปที่ 2.7 พอร์ต GPIO Raspberry Pi Model B (Revision 2)	10
รูปที่ 2.8 โมดูลควบคุมมอเตอร์ (L298N)	11
รูปที่ 2.9 เกียร์มอเตอร์และล้อรถ	14
รูปที่ 2.10 ระบบปฏิบัติการราสเบียน	15
รูปที่ 2.11 ภาษาไพทอน	16
รูปที่ 2.12 ดิวิตี่เซเคิล	18
รูปที่ 3.1 ตัวอย่างกราฟเส้นแกน X,Y	20
รูปที่ 3.2 ภาพรวมการทำงานของระบบฮาร์ดแวร์	21
รูปที่ 3.3 บล็อกไดอะแกรมการทำงานของรถจำลองอัตโนมัติ	22
รูปที่ 3.4 รถจำลองอัตโนมัติ	24
รูปที่ 3.5 ไดอะแกรมการต่อแหล่งจ่ายไฟฟ้าแบตเตอรี่กับบอร์ด	25
รูปที่ 3.6 การทำรูปให้เป็นสี่เหลี่ยม	26
รูปที่ 3.7 การทำรูปให้เป็นเส้นขอบ	26

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.1 การกำหนดพอร์ตของบอร์ด Raspberry Pi	28
รูปที่ 4.2 ถ่ายภาพจากกล้องคาเมราพาย (Camera Pi)	33
รูปที่ 4.3 ปรับภาพให้เป็นสีเทา	34
รูปที่ 4.4 ภาพที่ลดความละเอียดของภาพสีเทา	34
รูปที่ 4.5 ภาพที่ทำให้เป็นขอบเส้นในของรูปขาว-ดำ	35
รูปที่ 4.6 กราฟรูปภาพแกน X แกน Y	35
รูปที่ 4.7 รูปภาพที่เลือกขอบเส้นส่วนจำเป็น	36
รูปที่ 4.8 รูปภาพที่ทำการจับขอบเส้น	36

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อุบัติเหตุที่เกิดขึ้นกับการขับขี่รถยนต์นั้นเกิดขึ้นได้จากสาเหตุหลายประการ ทั้งจากตัวยานพาหนะ ถนน หรือแม้แต่จากตัวผู้ขับขี่เองก็อาจเป็นหนึ่งในสาเหตุสำคัญประการหนึ่งเพราะผู้ขับขี่รถยนต์แต่ละคนมีความแตกต่างกัน ไม่ว่าจะเป็นสภาพร่างกาย สภาพจิตใจ ความรวดเร็วในการตัดสินใจ ตลอดจนความประมาทขาดสติขาดความระมัดระวังก็มีส่วนทำให้เกิดอุบัติเหตุได้เช่นกัน การควบคุมรถด้วยระบบคอมพิวเตอร์แทนมนุษย์จะช่วยลดปัจจัยเสี่ยงเหล่านี้ทำให้ลดอุบัติเหตุลงได้

ปัญญาประดิษฐ์ (AI : Artificial Intelligence) เป็นความพยายามของมนุษย์ในการสร้างระบบที่มีความฉลาด ถูกนำมาใช้ประโยชน์ในหลายด้าน ทั้งด้านการเกษตร อุตสาหกรรม การแพทย์ รวมทั้งด้านธุรกิจการค้าขายโดยเฉพาะอย่างยิ่งในอุตสาหกรรมยานยนต์ไฟฟ้า ตัวอย่างเช่นบริษัทเทสลามอเตอร์ผู้ผลิตรถยนต์ไฟฟ้าได้นำระบบปัญญาประดิษฐ์มาใช้เพื่อช่วยให้การขับขี่รถยนต์มีความปลอดภัยมากขึ้น ลดความผิดพลาดจากการควบคุมของผู้ขับขี่ แต่เนื่องจากเทคโนโลยีนี้มีความซับซ้อนและยังไม่แพร่หลายในประเทศไทย ดังนั้นทางผู้จัดทำโครงการจึงมีแนวคิดพัฒนาประยุกต์ใช้กับรถจำลองขนาดเล็กติดกล้องดิจิทัลควบคุมด้วยระบบสมองกลฝังตัวเพื่อให้ขับเคลื่อนรถจำลองได้ด้วยโปรแกรมโดยปราศจากคนขับบนถนนจำลองเพื่อเป็นแนวทางในการพัฒนาระบบขับอัตโนมัติทดแทนการขับขี่โดยมนุษย์

1.2 คำสำคัญ (Keyword) สำหรับโครงการ

การมองเห็นของเครื่องจักร / การเรียนรู้ของเครื่องจักร/ ยานยนต์ไร้คนขับ / กล้องถ่ายภาพดิจิทัล / ภาษาไพทอน

Machine Vision / Machine learning / Driverless Vehicle / Digital Camera / Python Language

1.3 วัตถุประสงค์ในการทำโครงการ

- 1.3.1 เพื่อเป็นการศึกษาการใช้ภาษา python บน raspberry pi
- 1.3.2 เพื่อเป็นการศึกษาการใช้โปรแกรม Open

1.4 ขอบเขตและความสามารถของโครงการ

- 1.4.1 รถจำลองขนาดเล็กมี 3 ล้อ ควบคุมด้วยสมองกลฝังตัว
- 1.4.2 ถนนจำลองอยู่ภายในพื้นที่ขนาดไม่เกิน 4 ตารางเมตร
- 1.4.3 พัฒนาโดยใช้โปรแกรม Python, OpenCV, บนระบบปฏิบัติการ Raspbian
- 1.4.4 ความเร็วไม่เกิน 5 กิโลเมตรต่อชั่วโมง

1.5 ประโยชน์คาดว่าจะได้รับ

- 1.5.1 สามารถพัฒนารถจำลองที่ขับเคลื่อนโดยไร้คนขับได้ และช่วยลดอุบัติเหตุจากการขับที่
โดยมนุษย์
- 1.5.2 ได้รับรู้ระบบที่ตรวจจับถนนและสิ่งกีดขวางในช่องจราจรด้วยภาพถ่ายได้
- 1.5.3 ได้ศึกษาการทำงานของบอร์ด Raspberry Pi

1.6 เครื่องมือ เครื่องวัด และอุปกรณ์ที่ใช้

- | | | |
|---|---|---------|
| 1.6.1 ชุดบอร์ดพัฒนาราสเบอรี่พายพร้อมกล้องดิจิทัลขนาด 8 ล้านจุดภาพ | 1 | ชุด |
| 1.6.2 โมเดลแบบจำลองรถ | 1 | ชุด |
| 1.6.3 แบตเตอรี่ ลิเทียมโพลิเมอร์ | 1 | ชุด |
| 1.6.4 แผ่นกระดาษทำสนาม | 6 | ชุด |
| 1.6.5 เครื่องคอมพิวเตอร์ PC | 1 | เครื่อง |

1.7 แผนการดำเนินงาน

ตารางที่ 1.1 การวางแผนการดำเนินงาน

เลขที่	หัวข้องาน	%	2562									
			ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	
1	ศึกษางานข้อมูลเกี่ยวกับโครงการ	8.00										
		8.33										
2	นำเสนองานและรูปแบบ โครงงาน	8.00										
		8.33										
3	ศึกษาโครงสร้างและหลักการทำงาน Raspbian	8.00										
		8.33										
4	ศึกษาโครงสร้างและหลักการทำงาน OpenCV	4.00										
		4.17										
5	ออกแบบการทำงาน	16.00										
		12.51										
6	ศึกษาการควบคุมมอเตอร์และติดตั้งอุปกรณ์	12.00										
		12.51										
7	ติดตั้ง Open CV และเขียน โปรแกรมบน Python	12.00										
		8.33										
8	ทดสอบการใช้งาน	12.00										
		12.51										
9	ปรับปรุงโครงงานและบันทึกผล	12.00										
		12.51										
10	จัดทำปฏิญานินพนธ์	8.00										
		12.51										

ตามแผนงาน



ตามงานจริง



บทที่ 2

การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง

เนื้อหาบทนี้เป็นเอกสาร ทฤษฎีที่เกี่ยวข้องเพื่อใช้ในการปฏิบัติงานในโครงงานสร้างแบบจำลองรถไร้คนขับเคลื่อนอัตโนมัติ (Automatic Car) ได้ศึกษาค้นหาเอกสารการออกแบบโปรแกรม ขั้นตอนการสร้างโปรแกรมความรู้เกี่ยวกับ ตัวโปรแกรม Linux

2.1 การออกแบบโครงสร้างระบบ

การออกแบบ เป็นการแสดงลำดับของการทำงานของระบบ เพื่อให้เห็นภาพรวมของการทำงานของขั้นตอนทั้งหมด และเป็นการวางแผนการทำงานไว้

โดยทั่วไปผู้เขียนโปรแกรมจะแสดงการทำงานของโปรแกรมโดยใช้ผังงาน (Flowchart) ซึ่งสามารถอ่านและเข้าใจการทำงานได้ง่าย ช่วยให้ออกแบบระบบ การตรวจสอบ และแก้ไขสะดวก รวดเร็วขึ้น

ผังงาน (Flowchart) คือ รูปภาพหรือสัญลักษณ์ ที่ใช้เขียนแทนคำอธิบาย ข้อความ หรือ คำพูดที่ใช้ในอัลกอริทึม เพราะการที่จะเข้าใจขั้นตอนได้ง่ายและตรงกันนั้น การใช้คำพูดหรือข้อความอาจทำได้ยากกว่าการใช้รูปภาพหรือสัญลักษณ์ ผังงานสามารถแบ่งได้เป็น 2 ประเภท ใหญ่ ๆ คือ

1. ผังงานระบบ (System Flowchart) เป็นผังงานที่แสดงขั้นตอนการทำงานในระบบงานหนึ่ง ๆ ในลักษณะของภาพกว้าง ๆ แต่จะไม่เจาะลึกลงไปว่าในระบบงานย่อย ๆ นั้นจะมีการทำงานหรือวิธีการทำงานอย่างไร ผังงานจะแสดงทิศทางการทำงานในระบบ ตั้งแต่เริ่มต้นว่าข้อมูลเกิดขึ้นครั้งแรกที่ใด เก็บอยู่ในรูปแบบใด และผ่านขั้นตอนการประมวลผลอย่างไร อะไรบ้าง (แต่จะไม่เน้นถึงวิธีการประมวลผล) จนสุดท้ายผลลัพธ์ที่ได้เก็บอยู่ในรูปแบบใด ตัวอย่างเช่น ผังงานระบบบริหารโรงเรียนแห่งหนึ่ง ข้อมูลทะเบียนประวัติของนักเรียนจะเริ่มขึ้นครั้งแรกเมื่อมีการรับสมัครนักศึกษาใหม่ จากแผนกรับสมัคร และถือว่าเป็นข้อมูลพื้นฐานไปยังแผนกต่าง ๆ ในโรงเรียน เช่น แผนกปกครอง แผนกวัดผล หรือแผนกทะเบียน ซึ่งในส่วน of แผนกทะเบียนอาจจะมีการแก้ไขข้อมูลบางอย่าง เช่น มีการแก้ไขชื่อ ที่อยู่ของนักศึกษา ก็ได้

2. ผังงานโปรแกรม (Program Flowchart) เป็นผังงานที่แสดงถึงขั้นตอนในการทำงานของโปรแกรม ซึ่งจะแสดงการทำงานตั้งแต่เริ่มต้น โดยผู้เขียนผังงานอาจดึงเอาแต่ละจุดที่เกี่ยวข้องกับการทำงานของคอมพิวเตอร์เพื่อนำมาวิเคราะห์ว่า ซึ่งการเขียนผังงานนี้จะช่วยเพิ่มความสะดวกในการเขียนโปรแกรมของผู้เขียนโปรแกรมได้มาก เพราะสามารถดูได้ง่ายว่าในแต่ละขั้นตอนการทำงานควรใช้คำสั่งอย่างไร

ประโยชน์ของผังงานโปรแกรม

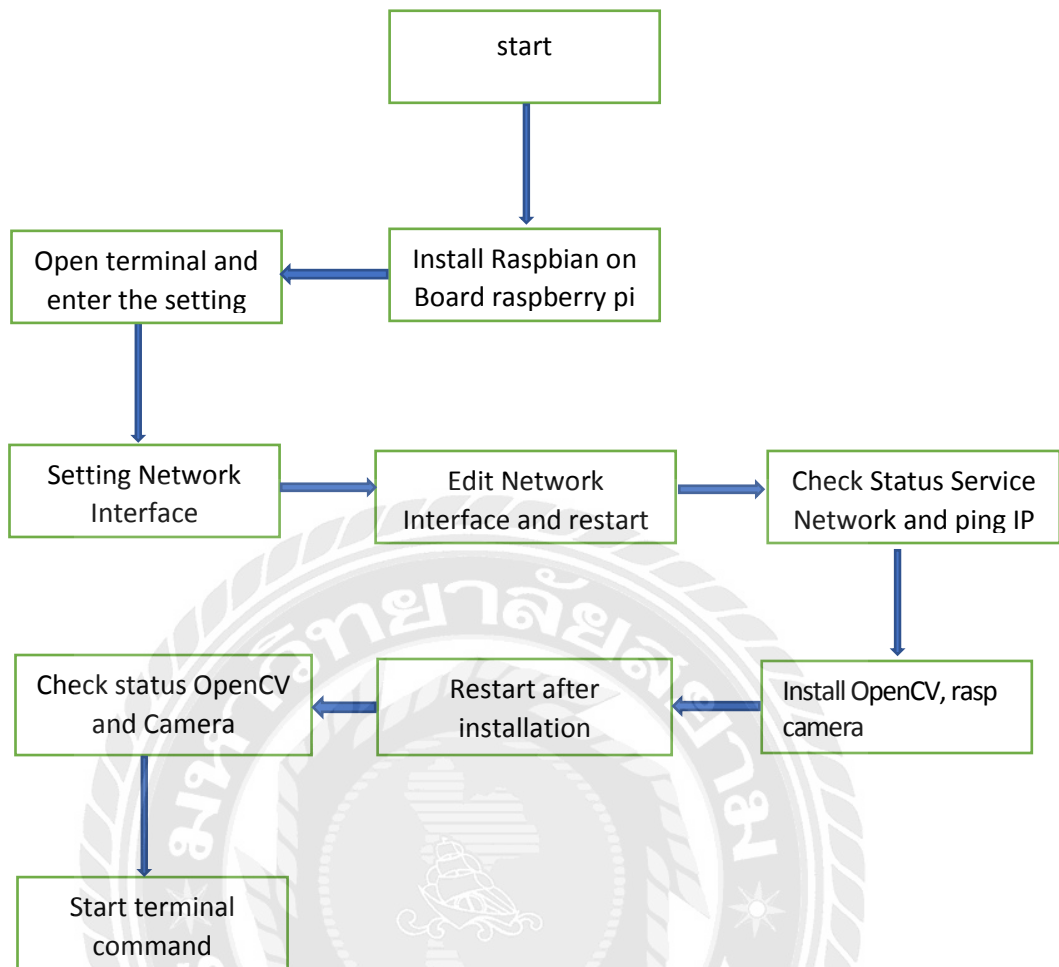
- ทำให้เข้าใจและแยกแยะปัญหาต่าง ๆ ได้ง่ายขึ้น
- ผู้เขียนโปรแกรมมองเห็นลำดับการทำงาน รู้ว่าสิ่งใดควรทำก่อน สิ่งใดควรทำหลัง
- สามารถหาข้อผิดพลาดของโปรแกรมได้ง่าย
- ทำให้ผู้อื่นเข้าใจการทำงานได้ง่ายกว่าการดูจาก source code
- ไม่ขึ้นกับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ผู้อื่นสามารถเรียนรู้และเข้าใจได้

ง่าย

ข้อจำกัดของผังงาน ผู้เขียนโปรแกรมบางคนไม่นิยมเขียนผังงานก่อนการเขียนโปรแกรม เพราะเห็นว่าเสียเวลา นอกจากนี้แล้ว ยังมีข้อจำกัดอื่น ๆ อีก คือ

- ผังงานเป็นการสื่อความหมายระหว่างบุคคลกับบุคคลมากกว่าที่สื่อความหมายระหว่างบุคคลกับเครื่อง เพราะผังงานไม่ขึ้นกับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ทำให้เครื่องไม่สามารถรับและเข้าใจได้ว่าในผังงานนั้นต้องการให้ทำอะไร
- ในบางครั้ง เมื่อพิจารณาจากผังงาน จะไม่สามารถทราบได้ว่า ขั้นตอนการทำงานใดสำคัญกว่ากัน เพราะทุก ๆ ขั้นตอนจะใช้รูปภาพหรือสัญลักษณ์ในลักษณะเดียวกัน

การเขียนผังงานเป็นการสิ้นเปลือง เพราะจะต้องใช้กระดาษและอุปกรณ์อื่น ๆ เพื่อประกอบกรเขียนภาพ ซึ่งไม่สามารถเขียนด้วยมืออย่างเดียวได้ และในบางครั้ง การเขียนผังงานอาจจะต้องใช้กระดาษมากกว่า 1 แผ่น หรือ 1 หน้า ซึ่งถ้าเป็นข้อความอธิบายอาจจะใช้เพียง 2-3 บรรทัดเท่านั้น



รูปที่ 2.1 ภาพแสดงขั้นตอนการทำงาน



รูปที่ 2.2 โอเพ่นซีวี

2.2 OpenCV 4.1.1

ปัจจุบัน OpenCV เป็น Open Source Library ที่กำลังเป็นที่นิยมสำหรับการพัฒนาโปรแกรมทาง Computer Vision ทั้งนี้ OpenCV นั้นไม่ได้เพิ่งเริ่มเกิดขึ้นในระยะหลังๆ หากแต่เริ่มต้นมาตั้งแต่ ปี 1999 ที่บริษัท Intel โดย Gary Bradski มีความประสงค์ที่จะสนับสนุนการวิจัยทาง Computer Vision ให้สามารถสร้างโปรแกรมประยุกต์ที่สามารถนำไปใช้ทางการค้าได้ ทาง Intel ทำเพื่อสร้างความต้องการ computer ที่ทรงพลังมากขึ้นกว่าเดิม ต่อมา Vadim Pisarevsky ได้เข้าร่วมกับ Gary บริหารทีม OpenCV รัสเซีย ของ Intel หลังจากนั้น ทีมงาน OpenCV ได้ย้ายไปบริษัทและทีมวิจัยต่างๆ เช่น Willow Garage ที่ทำการวิจัยทางหุ่นยนต์

ในปี 2008 ทาง Willow Garage ได้เล็งเห็นว่าหุ่นยนต์ต้องการความสามารถในการมองเห็น และความต้องการนี้เพิ่มขึ้นอย่างรวดเร็ว ทำให้กลับมาสนับสนุน OpenCV อีกครั้งหนึ่ง ทำให้ ระยะ 2-3 ปีหลังมานี้ เราได้พบชื่อ OpenCV ในการพัฒนาโปรแกรมต่าง ๆ มากขึ้น

OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License)

2.3 Raspberry Pi



รูปที่ 2.3 บอร์ด Raspberry Pi

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ เขียนโปรแกรม หรือเป็นคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ทำได้ทั้งการทำงาน Spreadsheet Word Processing ฟอรัมอินเทอร์เน็ต ส่งอีเมล หรือแม้กระทั่งเล่นเกม อีกทั้งยังสามารถเล่นไฟล์วิดีโอที่มีความละเอียดสูง (High-Definition) ได้

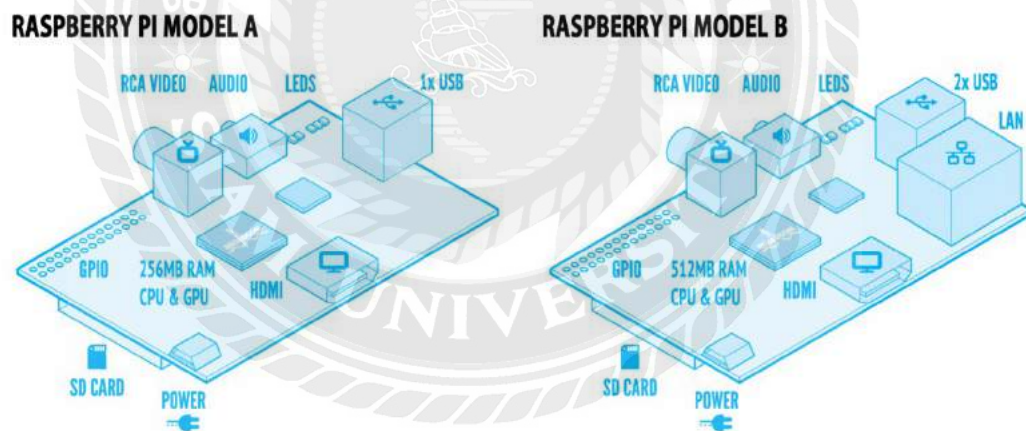
บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ได้

คุณสมบัติทางเทคนิคของบอร์ด

บอร์ด Raspberry Pi ปัจจุบันมีด้วยกัน 2 โมเดล คือ โมเดล A และ โมเดล B ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกันแตกต่างกันเพียงบางส่วน รายละเอียดดังตาราง

	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GIOPs) MPEG-2 and VC-1, 1080p 30 Hz/264/MPEG-4 AVC high-profile decoder and encoder	
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)
USB 2.0 Ports	1 (direct from BCM2835)	2 (via the build in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of RPi designed camera module (ออกแบบมาให้อัปเกรดกับ Raspberry Pi Camera Module โดยตรง)	
Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งจอคอม หรือ จอ TV RCA หรือ HDMI)	
Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I2S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripheral/low-level peripherals	8 x GPIO, UART, I2C Bus, SPI Bus with two chip selects, I2S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	
Size	85.60 mm x 53.3 mm (3.370 inch x 2.125 inch)	
Weight	45 g. (1.6 oz.)	

รูปที่ 2.4 แสดงรายละเอียดความแตกต่างของโมเดล A และ B



รูปที่ 2.5 รูปลักษณะโมเดล A และ B

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

รูปที่ 2.6 พอร์ต GPIO Raspberry Pi Model A & B (Revision 1)

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND		10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

รูปที่ 2.7 พอร์ต GPIO Raspberry Pi Model B (Revision 2)

2.4 โมดูลควบคุมมอเตอร์ (L298N)



รูปที่ 2.8 โมดูลควบคุมมอเตอร์ (L298N)

L298N เป็นชุดควบคุมมอเตอร์ชนิด H-Bridge ซึ่งถูกนำไปใช้ในการควบคุมทิศทาง และความเร็วของมอเตอร์ ซึ่งสามารถควบคุมมอเตอร์ได้ทั้งหมด 2 Channel

2.4.1 หลักการทำงานของ L298N

วงจร H-Bridge ของ L298N ขับกระแสเข้ามอเตอร์ตามหัวที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นถูกควบคุมด้วย สัญญาณ PWM (Pulse Width Modulation) ซึ่งเป็นการปรับความถี่ให้เหมาะสมกับมอเตอร์ที่ใช้

2.4.2 ลักษณะขาต่าง ๆ

- Out 1: ช่องต่อหัวไฟของมอเตอร์ A
- Out 2: ช่องต่อหัวไฟของมอเตอร์ A
- Out 3: ช่องต่อหัวไฟของมอเตอร์ B
- Out 4: ช่องต่อหัวไฟของมอเตอร์ B

- 12V : ช่องจ่ายไฟเลี้ยงมอเตอร์ 12V (ต่อได้ตั้งแต่ 5V ถึง 35V)
- GND : ช่องต่อไฟลบ (Ground)
- 5V : ช่องจ่ายไฟเลี้ยงมอเตอร์ 5V หากมีการต่อไฟเลี้ยงที่ช่อง 12V ทำหน้าที่จ่ายไฟ
ออก เป็น

5V Output สามารถต่อไฟจากช่องนี้ไปเลี้ยงบอร์ด Raspberry Pi หรือ
Arduino ได้

- ENA : ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ A
- IN1 : ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A
- IN2 : ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ A
- IN3 : ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B
- IN4 : ช่องต่อสัญญาณลอจิกเพื่อควบคุมทิศทางของมอเตอร์ B
- ENB : ช่องต่อสัญญาณ PWM สำหรับมอเตอร์ B

2.4.3 สเปกทางเทคนิค

- Dual H bridge Drive Chip : L298N
- แรงดันสัญญาณลอจิก : 5V Driver voltage 5V - 35V
- กระแสของสัญญาณลอจิก : 0 – 36 mA
- กระแสขับมอเตอร์ : สูงสุด 2A (เมื่อใช้มอเตอร์เดียว)
- กำลังไฟฟ้าสูงสุด : 25W
- ขนาด : 43 x 43 x 26 มิลลิเมตร
- น้ำหนัก : 26 กรัม

2.4.4 การสั่งงานมอเตอร์เบื้องต้น

การใช้ค่าในการควบคุมมอเตอร์ต่าง ๆ ตามตาราง 2.1 และ 2.2 ดังนี้

ตารางการทำงาน มอเตอร์ A

ENA	IN1	IN2	คำอธิบาย
0	N/A	N/A	มอเตอร์ A ไม่ทำงาน
0	LOW	LOW	มอเตอร์ A หยุดการทำงานแบบหมุนอิสระ (แบบไม่เบรค)
0 - 255	LOW	HIGH	มอเตอร์ A ทำงานและหมุนกลับหลังด้วยความเร็วที่ ENA
0 - 255	HIGH	LOW	มอเตอร์ A ทำงานและหมุนเดินหน้าด้วยความเร็วที่ ENA
0	HIGH	HIGH	มอเตอร์ A หยุดการทำงานแบบทันที (แบบเบรค)

ตารางที่ 2.1 การทำงานมอเตอร์ A

ตารางการทำงาน มอเตอร์ B

ENA	IN1	IN2	คำอธิบาย
0	N/A	N/A	มอเตอร์ B ไม่ทำงาน
0	LOW	LOW	มอเตอร์ B หยุดการทำงานแบบหมุนอิสระ (แบบไม่เบรค)
0 - 255	LOW	HIGH	มอเตอร์ B ทำงานและหมุนกลับหลังด้วยความเร็วที่ ENA
0 - 255	HIGH	LOW	มอเตอร์ B ทำงานและหมุนเดินหน้าด้วยความเร็วที่ ENA
0	HIGH	HIGH	มอเตอร์ B หยุดการทำงานแบบทันที (แบบเบรค)

ตารางที่ 2.2 การทำงานมอเตอร์ B

2.5 เกียร์มอเตอร์และล้อรถ สำหรับ Smart Robot Car Gear Motor with Tire

เกียร์มอเตอร์และล้อรถ สำหรับ Smart Robot Car Gear Motor with Tire เฟือง

1 : 120 ทดรอบสูงมอเตอร์มีแรงมากขึ้น จังหวะมอเตอร์ได้ละเอียดมากขึ้น



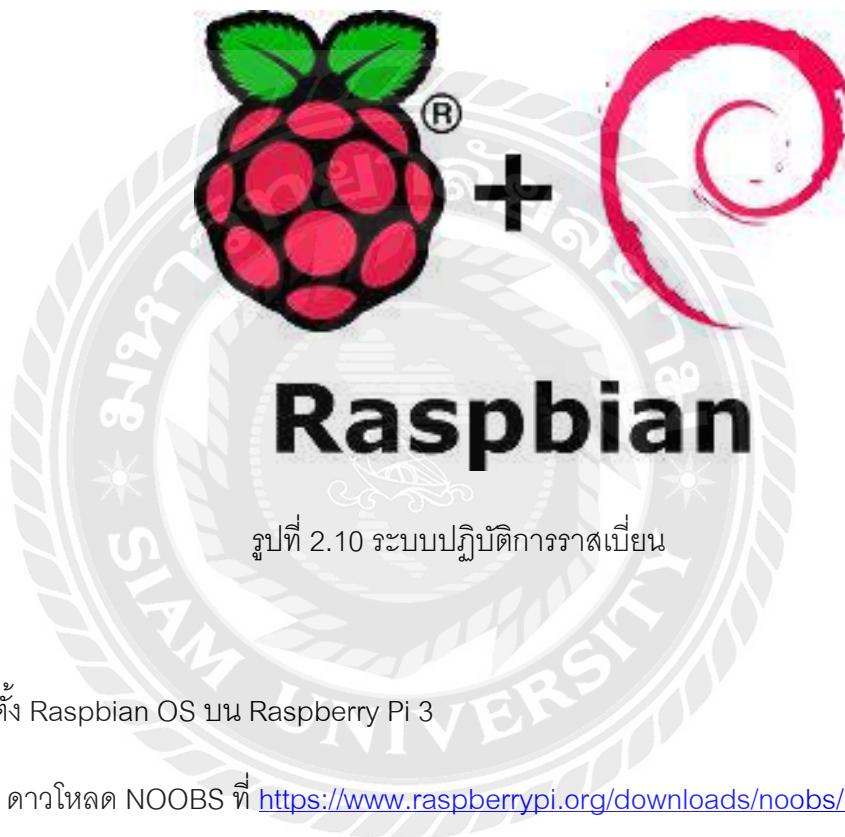
รูปที่ 2.9 เกียร์มอเตอร์และล้อรถ

อุปกรณ์ประกอบด้วย

- ล้อรถ
- เกียร์มอเตอร์ เฟือง 1 : 120

2.6 Raspbian OS

Raspbian คือ เป็นระบบปฏิบัติการสำหรับติดตั้งใช้งานบนบอร์ดขนาดเล็กนาม Raspberry Pi พัฒนามาจากระบบ Debian Linux เหมาะสำหรับนำมาใช้ทำแล็ปและงานวิจัยเกี่ยวกับระบบคอมพิวเตอร์แบบฝังตัว (Embedded System) โดยที่ Raspbian มีแพ็คเกจให้ใช้งาน 35,000 แพ็คเกจ กล่าวได้ว่าสามารถติดตั้งแพ็คเกจที่ใช้งานใน Debian Linux และ Ubuntu Linux ได้เกือบทุกแพ็คเกจ



การติดตั้ง Raspbian OS บน Raspberry Pi 3

- ดาวโหลด NOOBS ที่ <https://www.raspberrypi.org/downloads/noobs/>
- Format Micro SD Card โดยมี File System เป็น FAT32
- แยก Zip ไฟล์ไปที่ Micro SD Card ที่เราใช้ลง OS
- นำ Micro SD Card มาใส่เครื่อง Raspberry Pi 3
- เปิดเครื่อง Raspberry Pi
- จะพบหน้าจอการติดตั้ง ให้เลือกระบบปฏิบัติการ Raspbian แล้วกด Install
- รอจนเสร็จ โดยระบบมี Default คือ

Username: pi

Password: raspberry



รูปที่ 2.11 ภาษาไพทอน

2.7 Python

Python มีคุณสมบัติเป็นภาษาเขียนโปรแกรมไดนามิกส์และมีระบบการจัดหน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบ ที่ประกอบไปด้วย การเขียนโปรแกรมเชิงวัตถุ Imperative การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมที่มีขั้นตอน มีไลบรารีครอบคลุมการทำงานหลายรูปแบบ

ตัวแปรที่ใช้ในภาษา Python นั้นมีใช้ได้หลายระบบปฏิบัติการ ทำให้ได้ของ ภาษา Python สามารถรันในระบบอื่นได้อย่างกว้างขวาง CPython นั้น เป็นการพัฒนาระบบขั้นต้นของ Python ซึ่งเป็นโปรแกรมแบบ Open Source และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากมันได้มีการนำไปพัฒนากระจายไปหลากหลาย CPython นั้นจึงถูกจัดตั้งโดยองค์กรไม่แสวงหาผลกำไรอย่าง Python Software Foundation

ประวัติความเป็นมาของภาษา Python

ภาษา Python นั้นกำเนิดขึ้นในปลายปี 1980 และการพัฒนาของมันนั้นเริ่มต้นใน December 1989 โดย Guido van Rossum ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ เนื่องในผู้ประสบความสำเร็จในการสร้างภาษา ABC ที่มีความสามารถสำหรับการ Exception Handling และ การติดต่อผสานกับระบบปฏิบัติการ Amoeba ซึ่ง Van Rossum นั้นเป็นผู้เขียนหลักการของภาษา Python และทำหน้าที่ตัดสินใจทิศทางการพัฒนาภาษา Python

ไวยากรณ์ของภาษาของภาษา Python

ภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจจะให้มันเป็นภาษาที่อ่านง่ายมันถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษา C และ Pascal

Python Interpreter

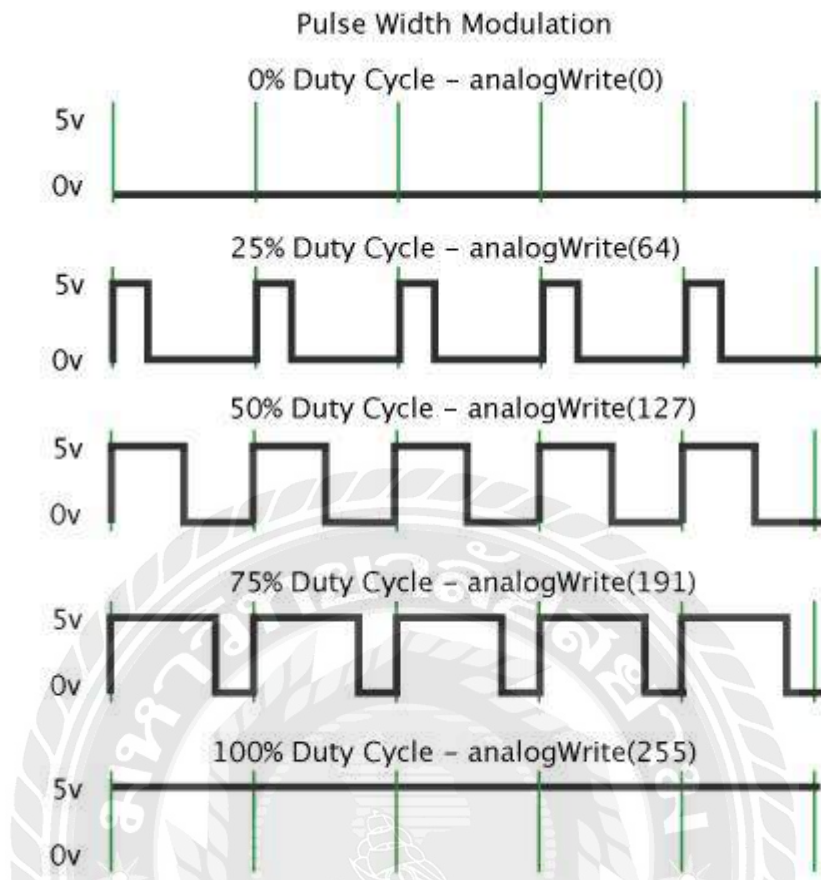
Python Interpreter นั้นเป็นตัวแปลภาษาของภาษา Python เพื่อให้สามารถรัน โค้ด Python ได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ <https://www.python.org/> ซึ่งเป็นโปรแกรมแบบ Source และ Binary สำหรับแพลตฟอร์มที่ได้รับ ความนิยม นอกจากนี้ Interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive Shell ซึ่งเป็นการเขียนโค้ดของ ภาษา Python ลงไปและเห็นผลลัพธ์การ

ทำงานของคำสั่ง Python Interpreter นั้นยังสามารถนำเพิ่มความสามารถกับฟังก์ชันใหม่ที่ถูก พัฒนาจาก ภาษา C และ C++ Python นั้นเหมาะสำหรับเป็นภาษาในการสร้าง Extension และ แอปพลิเคชันที่ปรับแต่งได้

2.8 การควบคุมมอเตอร์ (Motor Control)

2.7.1 สัญญาณ PWM (Pulse Width Modulation)

ถูกนำมาใช้ในการควบคุมความเร็วของมอเตอร์เพื่อสร้างสัญญาณเอาต์พุตแบบ ดิจิตอลส่งค่าให้มีสัญญาณสูง (1) สลับกับต่ำ (0) ในหนึ่งคาบเวลาด้วยความถี่ที่คงที่ ซึ่ง สามารถควบคุมระยะเวลาการจ่ายกระแสให้ขึ้นไปสูงสุดและต่ำสุดให้ได้สัดส่วนที่ต่างกัน ความเร็วของมอเตอร์จะขึ้นอยู่กับความถี่ที่ถูกใช้ ซึ่งมอเตอร์จะทำงานได้อย่างมีประสิทธิภาพ ต้องมีค่าความถี่ที่เหมาะสม เช่นความถี่เกินไปหมุนของมอเตอร์จะมีอาการสะดุด หรือถ้า ความถี่มากขึ้นเกินไปจะทำให้สูญเสียพลังงานเกินความจำเป็น



รูปที่ 2.12 ดิวตี้ไซเคิล

จากรูปที่ 2.10 เห็นได้ว่าความกว้างของพัลส์มีความแตกต่างกันเกิดจากแรงดันเฉลี่ยของสัญญาณ ซึ่งเปอร์เซ็นต์ความต่างในหนึ่งคาบเวลาเรียกว่า ดิวตี้ไซเคิล (Duty Cycle) หาค่าความสัมพันธ์ได้ดังนี้ $(\text{ค่าของสัญญาณพัลส์} / \text{คาบเวลาทั้งหมด ของสัญญาณ}) \times 100\%$ ซึ่งถ้าดิวตี้ไซเคิลมากนั้นหมายถึงค่าแรงดันเฉลี่ยมากจะทำให้มอเตอร์หมุนเร็วขึ้น แต่ถ้าดิวตี้ไซเคิลน้อยนั้นหมายถึงค่าแรงดันเฉลี่ยน้อยจะทำให้มอเตอร์หมุนช้าลง

บทที่ 3

การวิเคราะห์และการออกแบบ

ในบทนี้เป็นส่วนหนึ่งของการวิเคราะห์และออกแบบ โดยจะอธิบายถึงการออกแบบโครงสร้างรถจำลองอัตโนมัติ ระบบควบคุมซอฟต์แวร์ ของโครงการที่ได้ทำ โดยมีรายละเอียดในการวิเคราะห์และออกแบบดังต่อไปนี้

3.1 การออกแบบระบบ

ในโครงการนี้ทำการสร้างระบบที่ใช้กล้องคาเมราพาย (Camera Pi) ตรวจจับขอบเส้นสนาม ได้อย่างถูกต้อง โดยใช้คำสั่งผ่านราสเบอร์รี่พาย (Raspberry Pi) กำหนดการคำนวณค่าเฉลี่ยของการเลี้ยวซ้าย เลี้ยวขวา การป้องกันออกนอกเส้นทาง และขับเคลื่อนอัตโนมัติ

3.1.1 การออกแบบระบบฮาร์ดแวร์

ใช้บอร์ดราสเบอร์รี่พาย (Raspberry Pi) เป็นตัวหลักที่เรียกใช้ Library แต่ละแพ็คเกจต่าง ๆ ในการเปิดใช้ในการเขียนคำสั่งควบคุม โมดูลควบคุมมอเตอร์ (L298N) และกล้องคาเมราพาย (Camera Pi) กล้องทำการถ่ายภาพแล้วหาค่าเฉลี่ยของแกน X และการ Y คำนวณการเลี้ยวทิศทางของรถจำลองอัตโนมัติได้ มอเตอร์จะถูกกำหนดการปรับความเร็วแต่ละล้อ เพื่อการเข้าโค้งโดยใช้ PWM ทำงาน

3.1.2 การออกแบบระบบซอฟต์แวร์

CV2 : หรือ Opencv เป็นไลบรารีสำหรับการจัดการเกี่ยวกับรูปภาพ สามารถเรียกใช้จากภาษาไพธอน

- cv2.imread ทำการอ่านรูปขึ้นมา
- cv2.cvtColor คำสั่งเปลี่ยนรูปแบบสีในที่เลือกโหมด
cv2.COLOR_BGR2GRAY เปลี่ยนเป็นแบบขาวดำหรือสีเทา

- cv2.equalizeHist ทำให้ความเข้มภายในภาพมีความเปรียบเทียบต่างชัดเจน ดีขึ้นทำให้ตรวจจับภาพได้ง่ายขึ้น ภาพเปรียบเทียบระหว่างภาพเดิม กับ หลังเปลี่ยนไว้สี แล้วหลังจากใช้ cv2.equalizeHist
- cv2.GaussianBlur ทำให้ภาพที่คมชัดเป็นภาพเบลอ ลดรายละเอียดของภาพถ่าย
- cv2.Canny เป็นอัลกอริทึมการหาขอบตัดของรูปภาพ กระบวนการเบื้องต้นสำหรับการหาเส้นขอบ

Numpy : เป็นโมดูลส่วนเสริมของ Python ที่มีฟังก์ชันเกี่ยวกับคณิตศาสตร์และการคำนวณต่าง ๆ มาให้ใช้งาน โดยทั่วไปจะเกี่ยวกับการจัดการข้อมูลชุด (Array) ขนาดใหญ่และเมทริกซ์

- np.array อาเรย์ที่สร้างขึ้นมาจากลิสต์ หรือ เรนจ์
- np.zeros สร้างอาเรย์ที่มีแต่เลข 0 ตามขนาดที่กำหนด

Matplotlib : เป็นโมดูลสำหรับสร้างกราฟเส้นโดยอาศัยการประมวลผลจากโมดูล NumPy และสามารถเข้ากันกับ MATLAB ได้

```
import matplotlib.pyplot as plt
```

```
>>> X = [1,2,3,4]
```

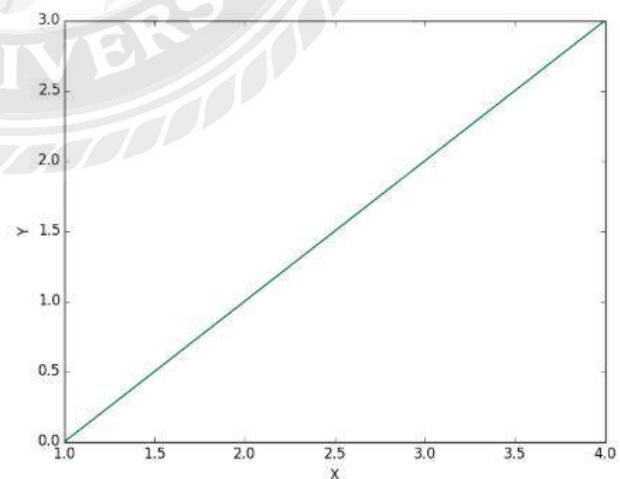
```
>>> Y = [0,1,2,3]
```

```
>>> plt.plot(x,y)
```

```
>>> plt.xlabel('X')
```

```
>>> plt.ylabel('Y')
```

```
>>> plt.show()
```

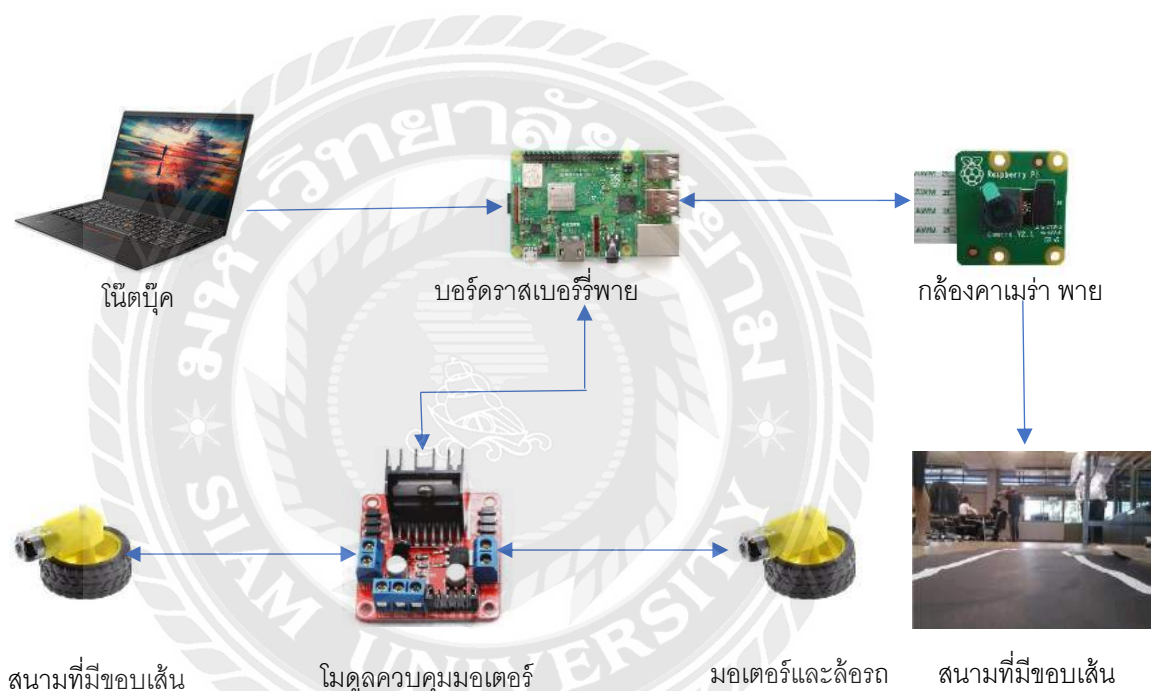


รูปที่ 3.1 ตัวอย่างกราฟเส้นแกน X,Y

GPIO : เป็นพอร์ตเอนกประสงค์ที่เป็น อินพุต และ เอาต์พุต สามารถควบคุมคอนโทรลให้เป็นลอจิก “1” หรือ ลอจิก “0” โดยสามารถควบคุมได้แต่ละ Pin เหมือนไมโครคอนโทรลเลอร์ทั่วไป

3.2 ภาพรวมการทำงานของโครงการ

ส่วนของภาพรวมการทำงานของรถจำลองอัตโนมัติเคลื่อนที่โดยใช้กล้องคำนวณค่าระยะห่างขอบเส้นทั้งสองฝั่ง โดยจะแบ่งการทำงานออกไว้ 2 ส่วนคือ ส่วนของฮาร์ดแวร์ และส่วนทางด้านซอฟต์แวร์

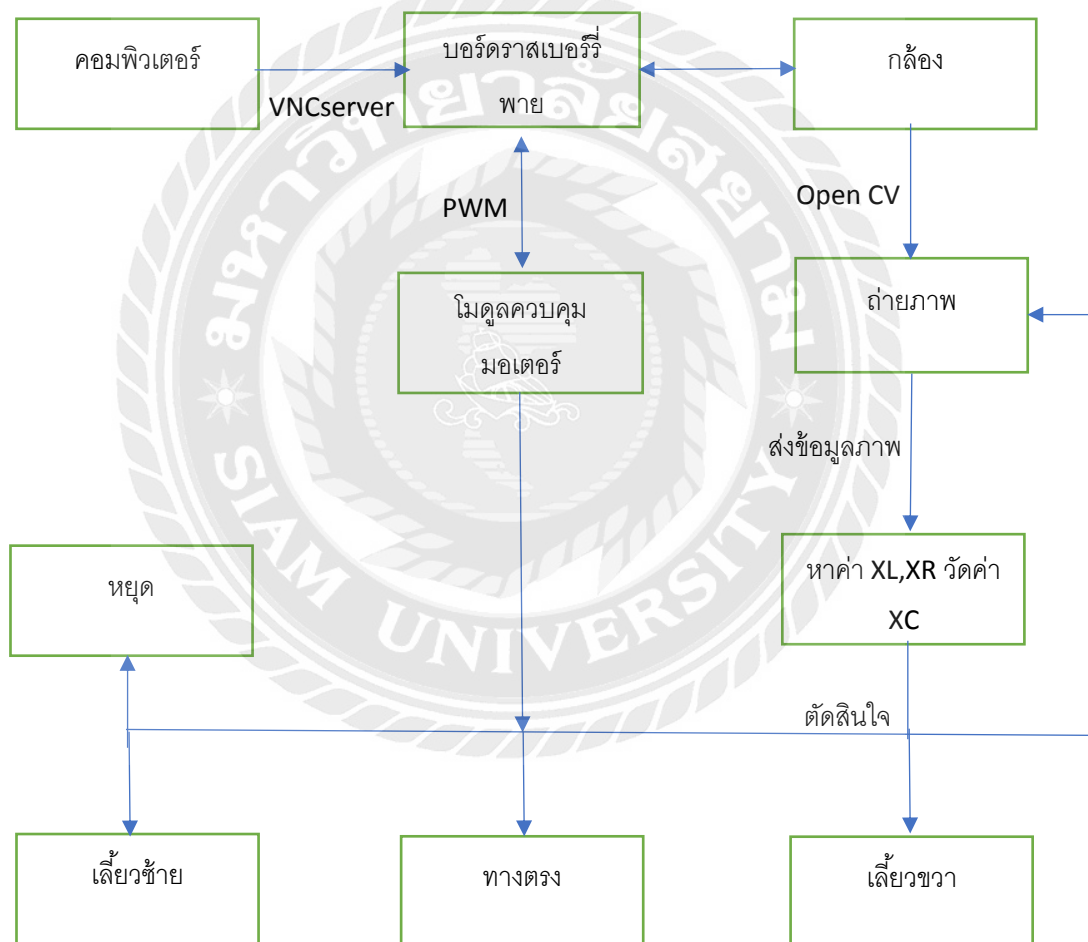


รูปที่ 3.2 ภาพรวมการทำงานของระบบฮาร์ดแวร์

จากรูปที่ 3.1 การทำงานของระบบฮาร์ดแวร์ มีลักษณะการทำงานโดยใช้คอมพิวเตอร์เชื่อมต่อผ่านบอร์ด Raspberry Pi โดยใช้โปรแกรม VNC Server เพื่อสะดวกในการใช้งานหรือใช้บอร์ดเป็นตัวหลักการทำงานทั้งหมดได้เช่นกัน การทำงานของบอร์ดประกอบไปด้วย กล้อง Camera Pi เพื่อทำการถ่ายภาพ และโมดูลควบคุมมอเตอร์ในการขับเคลื่อน

3.3 หลักการทำงานของระบบ

จากรูปที่ 3.1 สามารถอธิบายการทำงานของระบบที่ใช้ในการออกแบบในการสร้างและออกแบบรถจำลองอัตโนมัติ เมื่อคอมพิวเตอร์ทำการเชื่อมต่อกับบอร์ดราสเบอรี่พาย โดยผ่าน VNC Server จากนั้นป้อนคำสั่งการควบคุมมอเตอร์และควบคุมกล้องลงในราสเบอรี่พาย เพื่อให้ล้อสองข้างหมุนไปตามความเร็วที่กำหนดไว้ ผ่านการควบคุมของบอร์ด L298N ส่วนกล้องใช้ Open CV ทำงานในหน้าที่ตรวจจับขอบเส้นในขณะล้อหมุน ดังรูปที่ 3.2



รูปที่ 3.3 บล็อกไดอะแกรมการทำงานของรถจำลองอัตโนมัติ

3.4 การสร้างรถจำลองอัตโนมัติ

การออกแบบรถจำลองอัตโนมัติสำหรับโครงการนี้จำเป็นต้องคำนึงถึงปัจจัยด้านต่าง ๆ เช่น ใช้ต้นทุนที่ต่ำ ขนาดไม่ใหญ่ โครงสร้างไม่ซับซ้อน ประกอบง่าย และใช้แบตเตอรี่เพื่อประหยัดพลังงาน มีคุณสมบัติดังนี้

ตารางที่ 3.1 คุณสมบัติของรถจำลองอัตโนมัติ

องค์ประกอบ	คุณสมบัติ
ราสเบอร์รี่พาย (Raspberry Pi)	Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็ก เขียนโปรแกรม มีราคาถูก
คาเมร่าพาย (Camera Pi)	ราคาถูก ประสิทธิภาพสามารถถ่ายรูปและถ่ายวิดีโอ
โมดูลควบคุมมอเตอร์ (L298N)	ราคาถูก หาซื้อได้ง่าย มีขนาดเล็ก แข็งแรง ทนทาน
มอเตอร์และล้อรถ	ใช้พลังงานกลสร้างการเคลื่อนที่ สามารถติดตั้งและประกอบง่าย
สนามการทดลอง	จำลองสนามที่มีเส้นขอบ สามารถกำหนดทิศทางการวิ่งของรถจำลองอัตโนมัติ

จากข้อมูลตารางที่ 3.1 นำมาสร้างรถจำลองอัตโนมัติ โดยโครงรถจำลองอัตโนมัติใช้พลาสติกซึ่งมีน้ำหนักเบา ติดตั้งบอร์ดราสเบอร์รี่และบอร์ดควบคุมมอเตอร์ ในการควบคุมการใช้มอเตอร์และล้อรถ โดยทำอัลกรอริทึมผ่านการใช้กล้องคาเมร่าพายกำหนดค่าเฉลี่ย แต่ละทาง 1 วินาที/เซนติเมตร



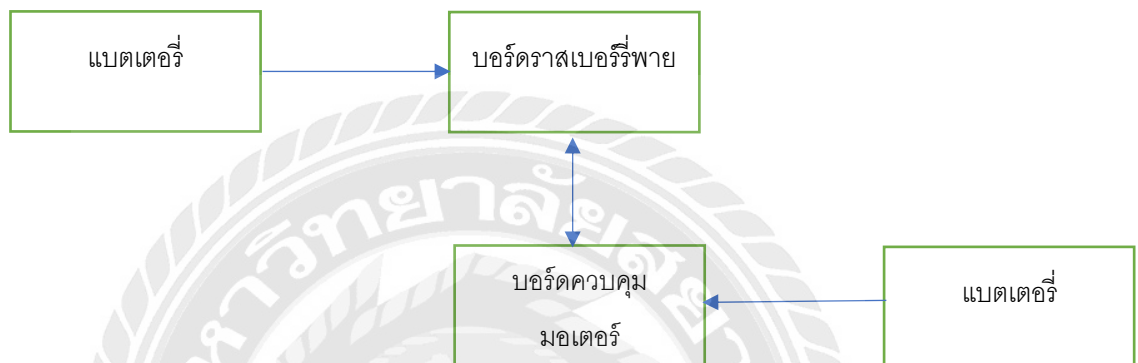
รูปที่ 3.4 รถจำลองอัตโนมัติ

ตารางที่ 3.2 คุณลักษณะของรถจำลองอัตโนมัติ

คุณลักษณะ	รายละเอียด
คอมพิวเตอร์	Raspberry Pi 3 B+
แรงดันไฟฟ้าที่ใช้	12 โวลต์
ขนาด (กว้าง x ยาว x สูง)	15 x 24 x 6 เซนติเมตร
เซนเซอร์	Camera Pi
กลไกการขับ	บอร์ดควบคุม L298N
น้ำหนัก	2.4 กิโลกรัม

3.5 การออกแบบระบบไฟฟ้าของรถจำลองอัตโนมัติ

การออกแบบระบบไฟฟ้าของรถจำลองอัตโนมัติทำการต่อแหล่งจ่ายไฟฟ้าโดยใช้แบตเตอรี่เป็นตัวจ่ายไฟฟ้าให้กับบอร์ดราสเบอร์รี่พาย รางเบอร์รี่พายเชื่อมต่อเข้ากับบอร์ด L298N บอร์ดควบคุมมอเตอร์ บอร์ดควบคุมมอเตอร์ใช้พลังงานไฟฟ้าจากแบตเตอรี่อีกตัว เพื่อให้มอเตอร์ทำงาน

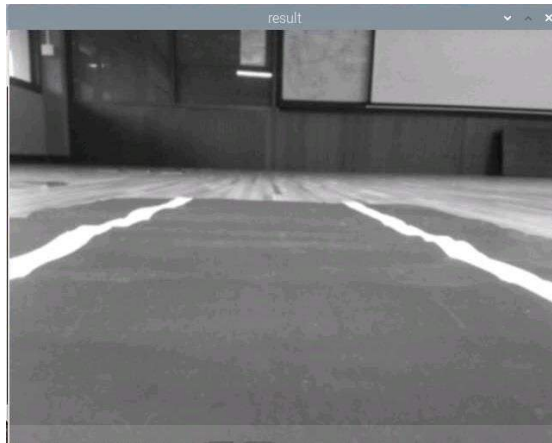


รูปที่ 3.5 ไดอะแกรมการต่อแหล่งจ่ายไฟฟ้าแบตเตอรี่กับบอร์ด

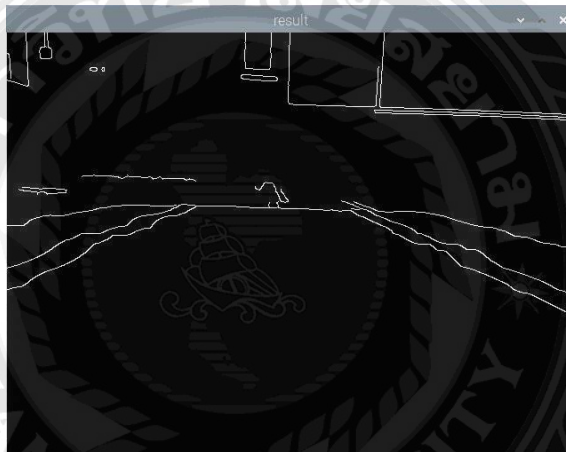
3.6 การหาขอบ (Edge Detection)

การหาขอบเส้นของภาพ ถูกนำมาใช้เพื่อหลายวัตถุประสงค์ เช่น เพื่อ Segment ภาพนั้น ออกเป็นส่วน ๆ หรือ เพื่อเป็นกระบวนการเบื้องต้นสำหรับการหาเส้นตรง หรือวงกลมในภาพ ฯลฯ ในที่นี่ใช้คำสั่งสำเร็จรูปของ OpenCV เพื่อหาขอบภาพ

<code>import cv2</code>	#เรียกแพ็คเกจจาก Library
<code>import numpy as np</code>	open cv และ Numpy
<code>img = cv2.imread("image.jpg")</code>	#เรียกรูปภาพ
<code>gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)</code>	#กำหนดภาพเป็นสีเทา
<code>blur = cv2.GaussianBlur(gray, (5, 5), 0)</code>	#ลดความละเอียดของภาพ
<code>canny = cv2.Canny(blur, 50, 150)</code>	#กำหนดทำเส้นขอบ
<code>cv2.imshow("edge" , gray)</code>	#แสดงผลภาพสีเทา
<code>cv2.imshow("canny" , canny)</code>	#แสดงผลภาพเส้นขอบ
<code>cv2.waitKey()</code>	#ออกจากการรัน



รูปที่ 3.6 การทำรูปให้เป็นสีเทา



รูปที่ 3.7 การทำรูปให้เป็นเส้นขอบ

บทที่ 4

การทดลอง

เพื่อให้ทราบถึงการทำงานของรถจำลองขับเคลื่อนอัตโนมัติ ใช้กล้องในการวัดค่าเฉลี่ยของระยะขอบเส้นถนน จึงทำการทดลองการทำงานของอุปกรณ์ และโปรแกรมว่าตรงตามที่ได้ออกแบบไว้ หรือไม่ เพื่อให้ทราบถึงข้อผิดพลาดที่เกิดขึ้น และนำข้อผิดพลาดนั้นมาทำการแก้ไข ปรับปรุงโครงงานให้สามารถทำงานได้อย่างมีประสิทธิภาพ

4.1 การทดลองส่งคำสั่งบอร์ดราสเบอร์รี่พายเพื่อไปควบคุมมอเตอร์ของล้อรถให้เคลื่อนที่

4.1.1 กำหนด Output ขาของ GPIO 1

4.1.2 กำหนด Output ขาของ GPIO 10

4.1.3 กำหนด Output ขาของ GPIO 22

4.1.4 กำหนด Output ขาของ GPIO 23

4.1.5 กำหนด Output ขาของ GPIO 24

4.1.6 กำหนด Output ขาของ GPIO 25

4.1.7 กำหนด Output ขาของ GPIO 27

ตารางที่ 4.1 การกำหนด Output สำหรับการเชื่อมต่อโมดูลควบคุมมอเตอร์ L298N

Port GPIO	เชื่อมต่อ	ใช้งาน
GPIO 1	Temp 1	Power
GPIO 24	IN 1	LOW
GPIO 23	IN 2	LOW
GPIO 27	IN 3	LOW
GPIO 22	IN 4	LOW
GPIO 25	EN 1	PWM
GPIO 10	EN 2	PWM



รูปที่ 4.1 การกำหนดพอร์ตของบอร์ด Raspberry Pi

4.2 การกำหนดไฟฟ้าภายใน

4.2.1 ใช้ไฟฟ้ากระแสตรง 5 V

4.2.2 ใช้ไฟฟ้ากระแสสลับ 220 V

4.2.3 การใช้ไฟฟ้าในอุปกรณ์ต่าง ๆ

4.2.3.1 ใช้พลังงานแบตเตอรี่จำนวน 10000 A เลี้ยงบอร์ด Raspberry

Pi

4.2.3.2 ใช้พลังงานแบตเตอรี่จำนวน 10000 A เลี้ยงโมดูลควบคุม

มอเตอร์ L298N

4.3 การทดลองการเคลื่อนที่ทางตรงของรถจำลองอัตโนมัติ

วัตถุประสงค์

4.3.1 เพื่อกำหนดความเร็วในการเคลื่อนที่ของการเลี้ยวซ้าย เลี้ยวขวา และหยุด
เมื่อไม่มีเส้นขอบ

4.3.2 ทำให้เคลื่อนที่ได้แบบอัตโนมัติ

อุปกรณ์ที่ใช้ในการทดลอง

4.3.3	โครงรถจำลอง	จำนวน 1 ชุด
4.3.4	คอมพิวเตอร์โน้ตบุ๊ก	จำนวน 1 เครื่อง
4.3.5	บอร์ด Raspberry Pi	จำนวน 1 เครื่อง
4.3.6	โมดูลควบคุมมอเตอร์ L298N	จำนวน 1 ชุด
4.3.7	แบตเตอรี่	จำนวน 2 ชุด

ขั้นตอนการทดลอง

เปิดการเชื่อมต่อ VNC server ระหว่างโน้ตบุ๊กผ่านตู้บอร์ดราสเบอร์รี่ พิมพ์ใน Terminal ภาษาไพทอน เพื่อทำการรันของมอเตอร์ หาค่าเฉลี่ยของการเคลื่อนที่ทางตรงของ 1 วินาที/เซนติเมตร โดยใช้คำสั่ง Run_Motor.py

ตารางผลการทดลองที่ได้

- ผลการทดลองจำนวน 5 ครั้งของการเคลื่อนที่ทางตรง 1วินาที/เซนติเมตร

ตารางที่ 4.2 ทดลองการเคลื่อนที่ทางตรงแต่ละครั้ง

จำนวนครั้ง	การทดสอบทางตรง	1 วินาที/เซนติเมตร
1	รันคำสั่ง Run_Motor.py	18.2
2	รันคำสั่ง Run_Motor.py	16.7
3	รันคำสั่ง Run_Motor.py	15.2
4	รันคำสั่ง Run_Motor.py	14.5
5	รันคำสั่ง Run_Motor.py	13.6

4.4 การทดลองการเคลื่อนที่เลี้ยวซ้ายของรถจำลองอัตโนมัติ

วัตถุประสงค์

- 4.4.1 เพื่อกำหนดความเร็วในการเคลื่อนที่ของการเลี้ยวซ้าย เลี้ยวขวา และหยุดเมื่อไม่มีเส้นขอบ
- 4.4.2 ทำให้เคลื่อนที่ได้แบบอัตโนมัติ

อุปกรณ์ที่ใช้ในการทดลอง

- | | | |
|-------|--------------------------|-----------------|
| 4.4.3 | โครงรถจำลอง | จำนวน 1 ชุด |
| 4.4.4 | คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.4.5 | บอร์ด Raspberry Pi | จำนวน 1 เครื่อง |
| 4.4.6 | โมดูลควบคุมมอเตอร์ L298N | จำนวน 1 ชุด |
| 4.4.7 | แบตเตอรี่ | จำนวน 2 ชุด |

ขั้นตอนการทดลอง

เปิดการเชื่อมต่อ VNC server ระหว่างโน้ตบุ๊กผ่านตู้บอร์ดราสเบอร์รี่พาย พิมพ์ใน Terminal ภาษาไพทอน เพื่อทำการรันของมอเตอร์ หาค่าเฉลี่ยของการเคลื่อนที่เลี้ยวซ้ายของ 1 วินาที/เซนติเมตร โดยใช้คำสั่ง Run_Motor.py

ตารางผลการทดลองที่ได้

- ผลการทดลองจำนวน 5 ครั้งของการเคลื่อนที่เลี้ยวซ้าย 1วินาที/เซนติเมตร

ตารางที่ 4.3 ทดลองการเคลื่อนที่เลี้ยวซ้ายแต่ละครั้ง

จำนวนครั้ง	การทดสอบเลี้ยวซ้าย	1 วินาที/เซนติเมตร
1	รันคำสั่ง Run_Motor.py	16.2
2	รันคำสั่ง Run_Motor.py	15.5
3	รันคำสั่ง Run_Motor.py	14.8
4	รันคำสั่ง Run_Motor.py	13.6
5	รันคำสั่ง Run_Motor.py	11.0

4.5 การทดลองการเคลื่อนที่เลี้ยวขวาของรถจำลองอัตโนมัติ

วัตถุประสงค์

- 4.5.1 เพื่อกำหนดความเร็วในการเคลื่อนที่ของการเลี้ยวซ้าย เลี้ยวขวา และหยุดเมื่อไม่มีเส้นขอบ
- 4.5.2 ทำให้เคลื่อนที่ได้แบบอัตโนมัติ

อุปกรณ์ที่ใช้ในการทดลอง

- | | | |
|-------|--------------------------|-----------------|
| 4.5.3 | โครงรถจำลอง | จำนวน 1 ชุด |
| 4.5.4 | คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.5.5 | บอร์ด Raspberry Pi | จำนวน 1 เครื่อง |
| 4.5.6 | โมดูลควบคุมมอเตอร์ L298N | จำนวน 1 ชุด |
| 4.5.7 | แบตเตอรี่ | จำนวน 2 ชุด |

ขั้นตอนการทดลอง

เปิดการเชื่อมต่อ VNC server ระหว่างโน้ตบุ๊กผ่านตู้บอร์ดราสเบอร์รี่พาย พิมพ์ใน Terminal ภาษาไพทอน เพื่อทำการรันของมอเตอร์ หาค่าเฉลี่ยของการเคลื่อนที่เฉลี่ยของ 1 วินาที/เซนติเมตร โดยใช้คำสั่ง Run_Motor.py

ตารางผลการทดลองที่ได้

- ผลการทดลองจำนวน 5 ครั้งของการเคลื่อนที่เฉลี่ยของ 1 วินาที/เซนติเมตร

ตารางที่ 4.4 ทดลองการเคลื่อนที่เฉลี่ยของ 1 วินาที/เซนติเมตร

จำนวนครั้ง	การทดสอบเฉลี่ยของ 1 วินาที/เซนติเมตร	1 วินาที/เซนติเมตร
1	รันคำสั่ง Run_Motor.py	15.0
2	รันคำสั่ง Run_Motor.py	14.2
3	รันคำสั่ง Run_Motor.py	12.8
4	รันคำสั่ง Run_Motor.py	11.1
5	รันคำสั่ง Run_Motor.py	10.5

4.6 การทดลองถ่ายภาพจากกล้องแคมเร่าพาย (Camera Pi)

วัตถุประสงค์

- 4.6.1 เพื่อแปลงรูป RGB ให้เป็นรูปเส้นขอบตามลำดับขั้นตอน และสามารถจับขอบเส้นได้

อุปกรณ์ที่ใช้ในการทดลอง

- | | | |
|-------|--------------------------|-----------------|
| 4.6.3 | โครงรถจำลอง | จำนวน 1 ชุด |
| 4.6.4 | คอมพิวเตอร์โน้ตบุ๊ก | จำนวน 1 เครื่อง |
| 4.6.5 | บอร์ด Raspberry Pi | จำนวน 1 เครื่อง |
| 4.6.6 | โมดูลควบคุมมอเตอร์ L298N | จำนวน 1 ชุด |
| 4.6.7 | แบตเตอรี่ | จำนวน 2 ชุด |

ขั้นตอนการทดลอง

เป็นขั้นตอนการทำรูปถ่ายหรือวิดีโอ นำรูป Original ที่ถ่าย เปลี่ยนเป็นรูปภาพสีเทาเพื่อลดความละเอียดของสี RGB ถึงการทำเส้นขอบและสามารถจับขอบเส้นได้ตามลำดับขั้นตอนดังต่อไปนี้

- ถ่ายรูปจากกล้องคาเมราพาย (Camera Pi)
- นำรูปจากภาพถ่ายมาปรับเป็นรูปสีเทา
- ทำการลดละเอียดของภาพสีเทาให้น้อยลงโดยการเบลอภาพ
- ทำกระบวนการภาพให้เป็นเส้นตรงอยู่ในรูปขาว-ดำ
- เข้าสู่อัลกอริทึมการหาค่าของ แกน X แกน Y จากกราฟ เพื่อทำการตัดส่วนที่ไม่จำเป็นออกไป
- ทำการจับขอบเส้นโดยฟังก์ชันที่ขอบเส้นส่วนตัดตามลำดับที่ผ่านมา

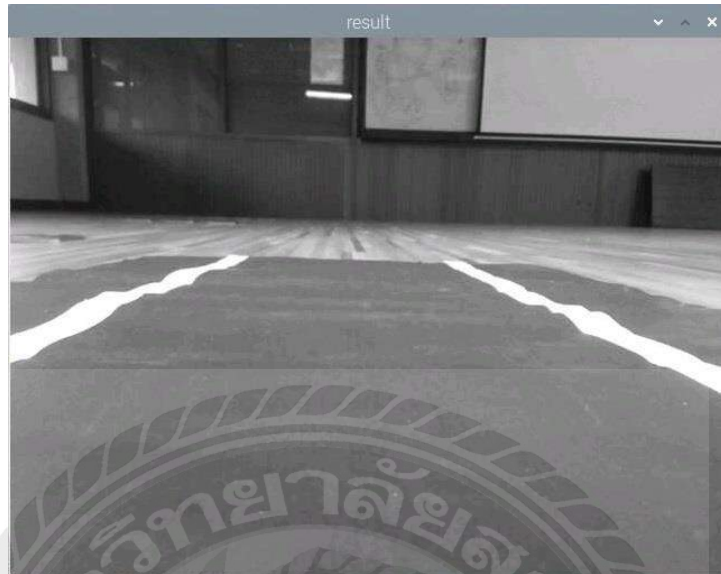
ผลการทดลองที่ได้

4.6.8 ถ่ายรูปจากกล้องคาเมราพาย (Camera Pi)



รูปที่ 4.2 ถ่ายภาพจากกล้องคาเมราพาย (Camera Pi)

4.6.9 นำรูปจากภาพถ่ายมาปรับเป็นรูปสีเทา



รูปที่ 4.3 ปรับภาพให้เป็นสีเทา

4.6.10 ทำการลดละเอียดของภาพสีเทานี้น้อยลงโดยการเบลอภาพ



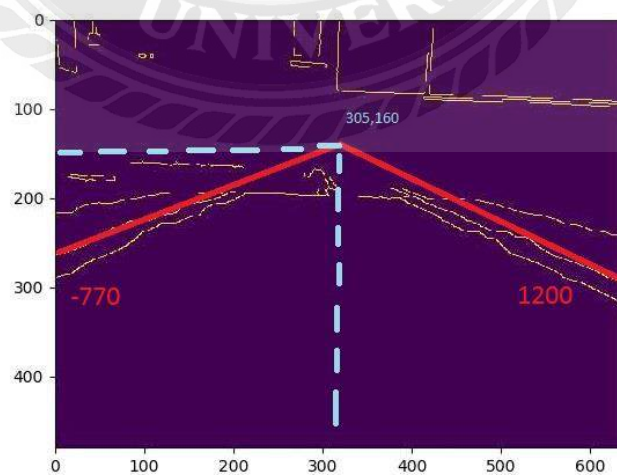
รูปที่ 4.4 ภาพที่ลดความละเอียดของภาพสีเทา

4.6.11 ทำกระบวนการภาพให้เป็นเส้นตรงอยู่ในรูปขาว-ดำ



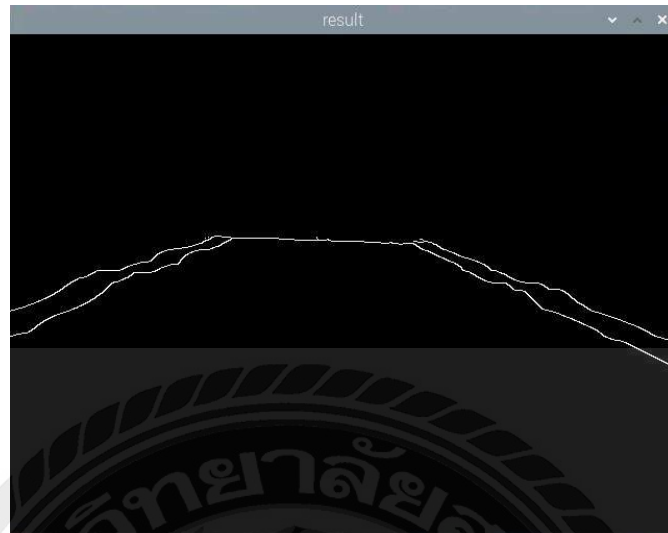
รูปที่ 4.5 ภาพที่ทำให้เป็นขอบเส้นในของรูปขาว-ดำ

4.6.12 เข้าสู่อัลกอริทึมการหาค่าของ แกน X แกน Y จากกราฟ เพื่อทำการตัดส่วนที่ไม่จำเป็นออกไป



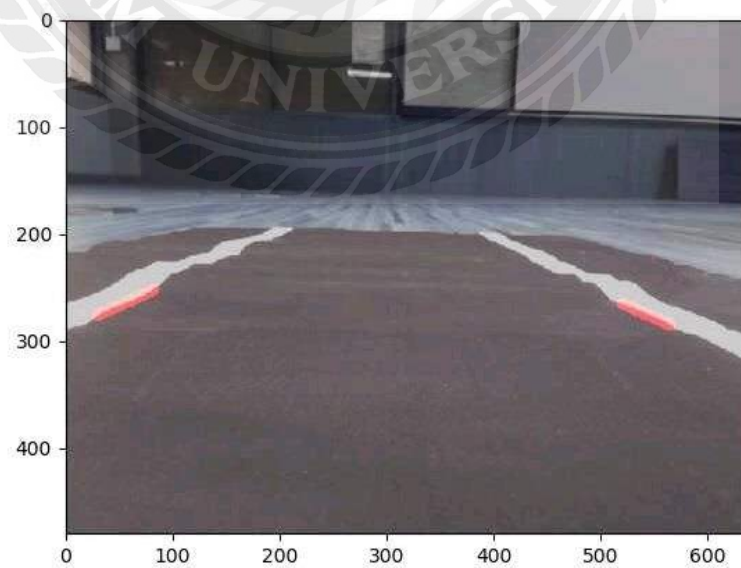
รูปที่ 4.6 กราฟรูปภาพแกน X แกน Y

4.6.13 แสดงรูปส่วนตัดที่ไม่ต้องการออกเพื่อเริ่มทำการขั้นตอนต่อไป



รูปที่ 4.7 รูปภาพที่เลือกขอบเส้นส่วนจำเป็น

4.6.14 ทำการจับขอบเส้นโดยโฟกัสที่ส่วนตัดตามลำดับที่ผ่านมา



รูปที่ 4.8 รูปภาพที่ทำการจับขอบเส้น

บทที่ 5

สรุปผลและข้อเสนอแนะ

การจัดทำโครงงานรถขับเคลื่อนอัตโนมัติได้สรุปผลการดำเนินงาน ปัญหา และข้อเสนอแนะในการทำงานดังต่อไปนี้

5.1 สรุปผลการดำเนินโครงงาน

การดำเนินงานในโครงงานรถขับเคลื่อนอัตโนมัติ โดยทำการทดลอง สรุปผลการทดสอบการใช้งานได้ดังนี้

- 5.1.1 การทดลองส่งคำสั่งบอร์ดราสเบอรี่พายเพื่อไปควบคุมมอเตอร์ล้อรถให้เคลื่อนที่กำหนดทิศทาง ซ้าย ขวา ถอยหลัง หยุด และความเร็วของมอเตอร์ล้อรถ
- 5.1.2 การทดสอบถ่ายรูปจากกล้องคาเมราพาย (Camera Pi) โดยใช้ภาษา Python
- 5.1.3 การทดสอบนำภาพปกติมาปรับเป็นรูปสีเทา
- 5.1.4 การทดสอบลดละเอียดของภาพสีเทาโดยทำภาพเบลอ
- 5.1.5 การทดสอบทำกระบวนการภาพให้เป็นเส้นตรงหรือรูปที่มีส่วนแค่เส้นขาว-ดำ
- 5.1.6 การทดสอบเข้าสู่อัลกอริทึมการหาค่าของ แกน X แกน Y จากกราฟเพื่อทำการตัดส่วนที่ไม่จำเป็นออกไป
- 5.1.7 การทดสอบจับขอบเส้นโดยไฟกสที่ขอบเส้นส่วนตัดตามลำดับที่ผ่านมา

5.2 ปัญหาการดำเนินโครงงาน

- 5.2.1 วัสดุที่ใช้ในการทำสนามเป็นกระดาษอ่อนเพื่อนำมาทดลองนั้น มีความมันของสีสนาม ในบางครั้งทำให้การหาขอบเส้นสนามมีความผิดพลาดได้
- 5.2.2 ล้อหลังไม่มีความสมดุล มีความหนักมากทำให้บางครั้งการเคลื่อนที่ของรถจำลองมีความผิดพลาด

5.3 ข้อเสนอแนะ

- 5.3.1 ควรมีเซ็นเซอร์ตัวอื่นมาช่วยด้วยทำให้มีการคำนวณที่แม่นยำ
- 5.3.2 ควรทำเป็น 4 ล้อ ช่วยในการขับเคลื่อนที่มีความสมดุล

บรรณานุกรม

วรรณพร เจริญแสนสวย. (2557). *การเขียนผังงาน (Flowchart)*. เข้าถึงได้จาก

<https://stwannaporn.wordpress.com/2014/06/29/การเขียนผังงาน-flowchart/>

Supphachoke Suntiwichaya. (2562). *Compile OpenCV 4 บน Raspberry pi*. เข้าถึงได้จาก

<https://medium.com/nectec/compile-opencv-4-บน-raspberry-pi-4-a89018160685>

CRMA IT Knowledge. (ม.ป.ป.). *ทำความรู้จัก Raspberry Pi*. เข้าถึงได้จาก

<http://www2.crma.ac.th/itd/Know/RBPI/index.asp>

Narin Group. (2559). *การใช้ชุดขับเคลื่อนมอเตอร์ L298N Dual H-Bridge Motor Controller*. เข้าถึงได้จาก

<http://naringroup.blogspot.com/2016/03/robot-l298n-dual-h-bridge-motor.html>

raspberrhome. (ม.ป.ป.). *การติดตั้ง Raspbian บน Raspberry Pi แบบไม่ต่อจอภาพ*. เข้าถึงได้จาก

<http://www.raspberrhome.net/article/1/การติดตั้ง-raspbian-บน-raspberry-pi-แบบไม่ต่อจอภาพ>

Sarayut Nonsiri. (ม.ป.ป.). *ภาษาโปรแกรม Python คืออะไร*. เข้าถึงได้จาก

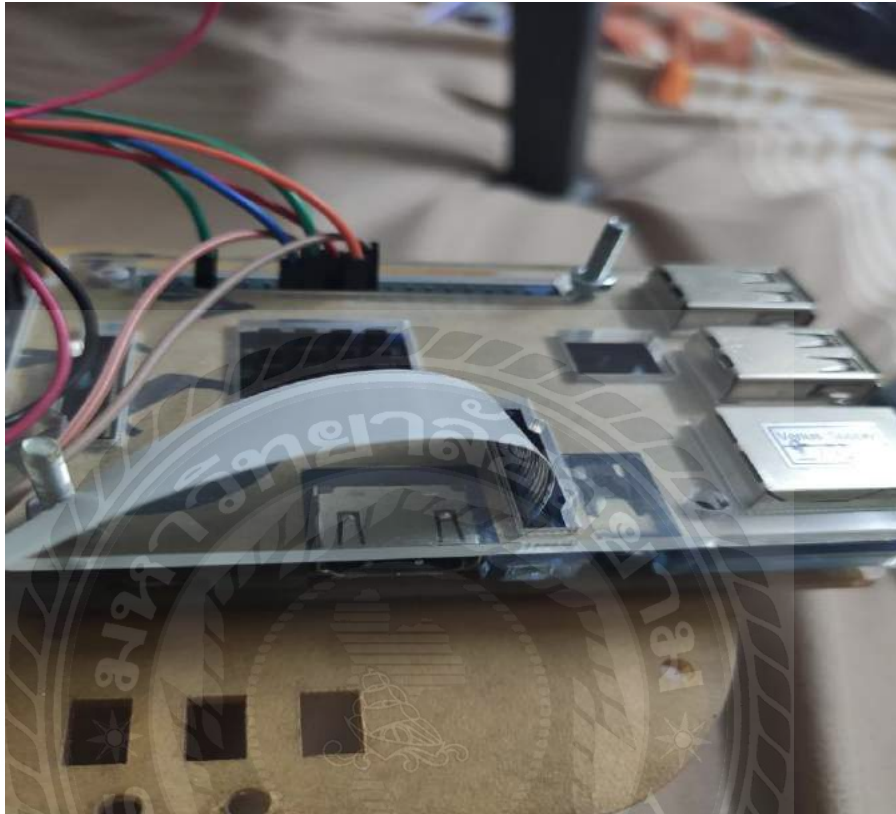
<https://www.9experttraining.com/articles/python-คืออะไร>



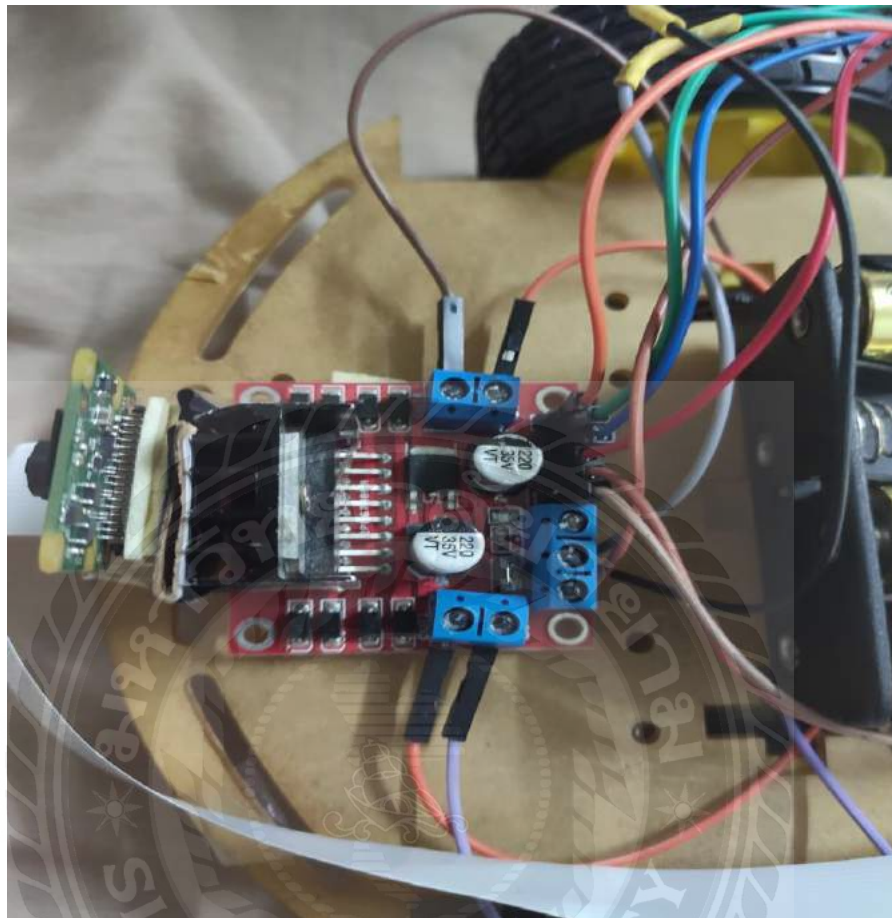


ภาคผนวก ก
การติดตั้งอุปกรณ์ฮาร์ดแวร์

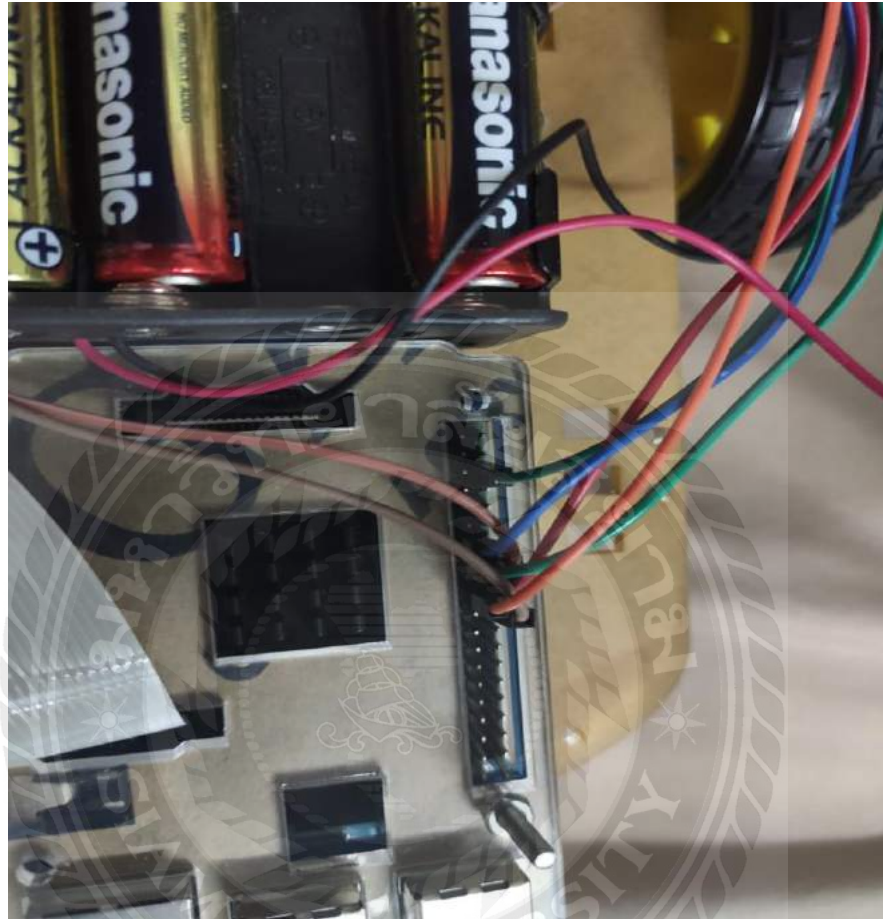
การประกอบบอร์ด Raspberry Pi กับ อุปกรณ์ รถจำลอง



รูปที่ ก.1 การเชื่อมต่อสายระหว่างบอร์ด raspberry Pi กับ Camera Pi



รูปที่ ก.2 การเชื่อมต่อสายไฟระหว่างบอร์ด Raspberry Pi กับ โมดูลขับเคลื่อนมอเตอร์ (L298N)



รูปที่ ก.3 พอร์ตที่ใช้ในการเชื่อมต่อของรถจำลอง

ประวัติผู้จัดทำ

นายณัฐภูมิภัทร์ เขมะทัตส์

วันเดือนปีเกิด 1 เดือน พฤษภาคม พ.ศ. 2531

รหัสนักศึกษา 5804000018

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์



ที่อยู่ 75 หมู่ 9 ตำบลกระดาง อำเภอบางคนที จังหวัดสมุทรสงคราม 75120

ประวัติการศึกษา

- ปีพุทธศักราช 2551 ประกาศนียบัตรวิชาชีพ แผนกยานยนต์ วิทยาลัยสารพัดช่างสมุทรสงคราม

นายยุทธนา พิมพ์เสน

วันเดือนปีเกิด 4 เดือน พฤศจิกายน พ.ศ. 2539

รหัสนักศึกษา 5804000025

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์



ที่อยู่ 44/1 หมู่ 4 ตำบลโคกปี่ อำเภอศรีมโหสถ จังหวัดปราจีนบุรี 25190

ประวัติการศึกษา

- ปีพุทธศักราช 2557 มัธยมศึกษาตอนปลาย โรงเรียนมัธยมวัดใหม่กรรทอง ในพระราชูปถัมภ์ฯ