

รถส่งเวชภัณฑ์อัตโนมัติ
AUTOMATIC TRANSPORT OF MEDICAL SUPPLIES



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถส่งเวชภัณฑ์อัตโนมัติ
AUTOMATIC TRANSPORT OF MEDICAL SUPPLIES

โดย

นางสาวพจนวรรณ	อินทร์จันทร์	61010677
นายเพชร	วิจิตรพันธุ์	61010679
นายพุทธะสอน	สุกทะเลิไล	61010769

อาจารย์ที่ปรึกษา

ศ. ดร.ชวงค์ พงศ์เจริญพานิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รถส่งเวชภัณฑ์อัตโนมัติ

AUTOMATIC TRANSPORT OF MEDICAL SUPPLIES

ผู้จัดทำ

- | | | |
|------------------|--------------|----------|
| 1. นางสาวพจนวรรณ | อินทร์จันทร์ | 61010677 |
| 2. นายเพชร | วิจิตรพันธุ์ | 61010679 |
| 3. นายพุทธะสอน | สุกทะเลิไล | 61010769 |



อาจารย์ที่ปรึกษา

(ศ. ดร.ชวรงค์ พงศ์เจริญพาณิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินโครงการเรื่อง “รถส่งเวชภัณฑ์อัตโนมัติ” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษาโครงการคือ ศ.ดร.ชวรงค์ พงศ์เจริญพาณิชย์ รวมทั้งรุ่นพี่ทุกท่านในห้องปฏิบัติการทดลองที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา คำนึงว่าวิจัยให้โครงการนี้สำเร็จ สมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดทำโครงการ ขอขอบพระคุณในความปรารถนาดี และความห่วงใยที่มีให้แก่คณะผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษามีค่ายิ่งแก่ผู้จัดทำ

นางสาวพจนวรรณ	อินทร์จันทร์
นายเพชร	วิจิตรพันธุ์
นายพุฒะสอน	สุกทะเล
	ผู้จัดทำ

รถส่งเวชภัณฑ์อัตโนมัติ

AUTOMATIC TRANSPORT OF MEDICAL SUPPLIES

โดย	นางสาวพจนวรรณ	อินทร์จันทร์	61010677
	นายเพชร	วิจิตรพันธุ์	61010679
	นายพุทธะสอน	สุกทะเลวิไล	61010769

อาจารย์ที่ปรึกษา ศ. ดร. ชวงค์ พงศ์เจริญพาณิชย์

บทคัดย่อ

จากมาตรการในการเว้นระยะห่างทางสังคม หรือ Social Distancing เนื่องจากเชื้อไวรัสโควิด-19 เพื่อช่วยลดความเสี่ยงของแพทย์และพยาบาลที่ทำการรักษาผู้ป่วยอย่างใกล้ชิด จึงได้ทำการศึกษาและจัดทำรถขนส่งเวชภัณฑ์อัตโนมัติ เพื่ออำนวยความสะดวกและเพิ่มประสิทธิภาพในการขนส่งยาและเวชภัณฑ์และลดโอกาสในการพบปะกับผู้ติดเชื้อโควิด-19 โดยไม่จำเป็น โดยระบบประกอบด้วยสองส่วนหลัก ๆ คือส่วนการพัฒนารถขนส่งเวชภัณฑ์ที่สามารถเดินตามเส้นโดยใช้อินฟราเรดเซนเซอร์และอัลตราโซนิกเซนเซอร์ในการประมวลผลและแสดงผลการควบคุมผ่านหน้าจอ LCD ขนาดเล็ก และระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชันโดยการควบคุมผ่าน Wi-Fi แล้วนำข้อมูลไปประมวลผลเพื่อสั่งการให้มอเตอร์ทำงาน ซึ่งระบบทั้งหมดจะถูกควบคุมผ่านไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Due to covid-19, social distancing measures have been implemented to prevent the possibility of doctors and nurses over-treating patients. As a result, we have researched and developed autonomous medical supply transport trucks to make it easier and more efficient to transfer medicines and medical supplies, as well as to limit the chances of coming into contact with people who are infected with COVID-19. The system consists of two main components: the development of a medical supplies transporter that can follow the lines using infrared and ultrasonic sensors to process and display control through a small LCD screen, and a remote-control system that uses the application to control it via Wi-Fi and then process the data to trigger the motor to operate, all of which are controlled by a microcontroller.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
บทที่ 2	3
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560	3
2.2 ESP32 Cam with OV2640 Camera Module	4
2.3 โมดูล FT232RL	5
2.4 เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวน	6
2.5 เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04	6
2.6 โมดูลขับเคลื่อนมอเตอร์ L298N	9
2.7 โมดูลเสียง Active Buzzer Module	9
2.8 มอเตอร์ไฟฟ้ากระแสตรง	10
2.9 จอแสดงผล	12
2.10 คีย์แพดปุ่มกด	13
2.11 มิเตอร์วัดแรงดันแบตเตอรี่	14
2.12 สวิตช์ปุ่มกด	14
2.13 หลักการทำงานของรถเดินตามเส้น	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.14 เว็บแอปพลิเคชัน	19
บทที่ 3 การออกแบบและการจัดทำปฏิญญานิพนธ์	22
3.1 การออกแบบ	22
3.2 เครื่องมือที่ใช้ในการทดลอง	36
3.3 การจัดเก็บผลการทดลอง	40
บทที่ 4 ผลการทดลอง	42
4.1 ผลการทดสอบการทำงานของอินฟราเรดเซนเซอร์	42
4.2 ผลการทดสอบการวัดระยะของอัลตราโซนิกเซนเซอร์	44
4.3 ผลการทดสอบประสิทธิภาพของแบตเตอรี่	48
4.4 ผลการทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้	49
4.5 ผลการทดสอบการแสดงวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์	49
4.6 การทดสอบระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน	51
4.7 การทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงจุดหมาย	52
4.8 การทดสอบการมอดูเลตความกว้างพัลส์ของโมดูลขับเคลื่อนมอเตอร์ L298N	53
บทที่ 5 สรุปผลและข้อเสนอแนะ	55
5.1 สรุปผล	55
5.2 ข้อเสนอแนะ	56
บรรณานุกรม	57
ภาคผนวก ก คำสั่งควบคุมการทำงานของระบบการเดินตามเส้น	61
ภาคผนวก ข คำสั่งควบคุมการทำงานระยะไกลผ่านเว็บแอปพลิเคชัน	95

สารบัญรูป

รูปที่		หน้า
2.1	บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560	4
2.2	ESP32 Cam with OV2640 Camera Module	5
2.3	โมดูล FT232RL	6
2.4	เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวน	6
2.5	เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04	7
2.6	หลักการวัดระยะทางด้วยคลื่นอัลตราโซนิก	8
2.7	แผนผังเวลาสำหรับเซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04	8
2.8	โมดูลขับเคลื่อนมอเตอร์ L298N	9
2.9	โมดูลเสียง Active Buzzer Module	10
2.10	มอเตอร์ ZGA60FM	11
2.11	หน้าจอ LCD พร้อมโมดูล I2C ขนาด 16x2	12
2.12	คีย์แพดปุ่มกด 4x4 Matrix (Black Keypad)	13
2.13	มิเตอร์วัดแรงดันแบตเตอรี่	14
2.14	สวิตช์ปุ่มกดแบบกดติดปล่อยดับ	15
2.15	สวิตช์ปุ่มกดแบบกดติดกดดับ	16
2.16	หุ่นยนต์เคลื่อนที่ไปข้างหน้า	16
2.17	หุ่นยนต์เลี้ยวไปทางด้านซ้าย	17
2.18	หุ่นยนต์เลี้ยวไปทางด้านขวา	17
2.19	หุ่นยนต์หยุดเคลื่อนที่	18
2.20	ส่วนประกอบของเว็บแอปพลิเคชัน	20
2.21	ส่วนประมวผลของเว็บแอปพลิเคชัน	21
3.1	บล็อกไดอะแกรมของรถส่งเวชภัณฑ์อัตโนมัติ	22
3.2	ตำแหน่งการติดตั้งอินฟราเรดเซนเซอร์	23
3.3	ด้านหลังของรถส่งเวชภัณฑ์	23

สารบัญรูป (ต่อ)

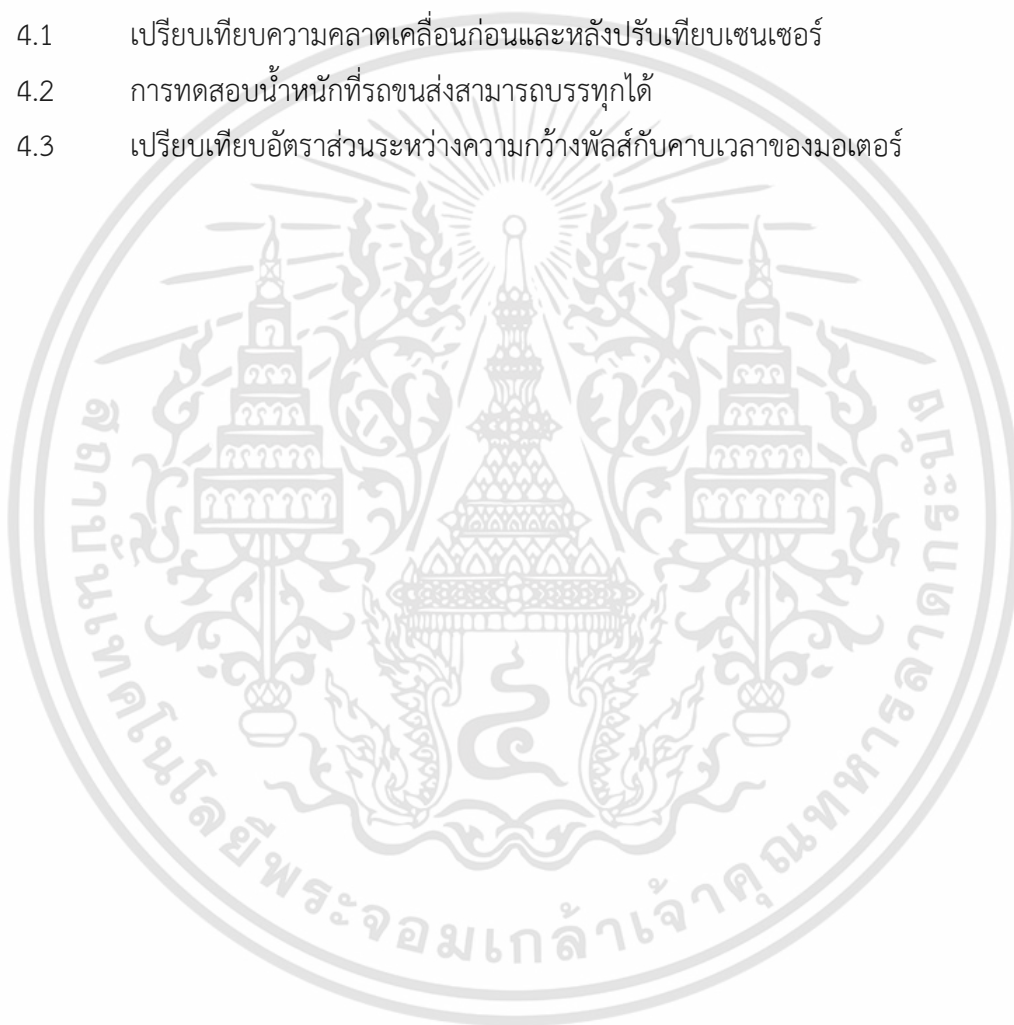
รูปที่		หน้า
3.4	ด้านหน้าของรถส่งเวชภัณฑ์	24
3.5	ด้านข้างของรถส่งเวชภัณฑ์	24
3.6	บล็อกไดอะแกรมการทำงานของระบบการเดินตามเส้น	25
3.7	ภาพการเชื่อมต่อวงจรของระบบการเดินตามเส้น	26
3.8	บล็อกไดอะแกรมการทำงานของระบบควบคุมระยะไกล	26
3.9	รถขนส่งเบนหรือเลี้ยวไปทางขวาในรูปแบบต่าง ๆ	27
3.10	รถขนส่งเบนหรือเลี้ยวไปทางซ้ายในรูปแบบต่าง ๆ	28
3.11	แผนผังการทำงานของระบบควบคุมการเดินตามเส้น	29
3.12	รูปแบบการทำงานกรณีห้องหมายเลข 1	30
3.13	แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 1	31
3.14	รูปแบบการทำงานกรณีห้องหมายเลข 2	32
3.15	แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 2	32
3.16	รูปแบบการทำงานกรณีห้องหมายเลข 3	33
3.17	แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 3	34
3.18	รูปแบบการทำงานกรณีห้องหมายเลข 4	35
3.19	แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 4	35
3.20	แผนผังการทำงานของระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน	36
3.21	แบตเตอรี่ที่มีแรงดันไฟฟ้า 12 โวลต์ 7.2 แอมแปร์ชั่วโมง	38
3.22	ล้อขับเคลื่อนรถส่งเวชภัณฑ์	39
3.23	สวิทช์สองขา	39
3.24	สวิทช์ปุ่มกดแบบกดติดปล่อยดับ	39
4.1	อินฟราเรดเซนเซอร์ตรวจพบพื้นสีขาว	42
4.2	Serial Monitor เมื่อค่าลอจิกเป็น 0	43
4.3	อินฟราเรดเซนเซอร์ตรวจพบพื้นสีดำ	43
4.4	Serial Monitor เมื่อค่าลอจิกเป็น 1	43

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.5	การจำลองการทดสอบวัตรยะด้วยอัลตราโซนิกเซนเซอร์โดยใช้วัตถุจำลองทรงสี่เหลี่ยมผืนผ้า	44
4.6	ความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้ก่อนการปรับเทียบเซนเซอร์	45
4.7	ความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้หลังการปรับเทียบเซนเซอร์	46
4.8	ค่าแรงดันไฟฟ้าขณะใช้งานของแบตเตอรี่ก้อนที่ 1 และแบตเตอรี่ก้อนที่ 2	48
4.9	หน้าต่าง Serial Monitor แสดง IP ของ ESP32 Cam	50
4.10	วิดีโอภาพเคลื่อนไหวแบบเรียลไทม์บนเว็บเบราว์เซอร์	50
4.11	หน้าเว็บแอปพลิเคชันบนคอมพิวเตอร์	51
4.12	หน้าเว็บแอปพลิเคชันบนโทรศัพท์มือถือ	52
4.13	ตำแหน่งของฝ่ามือกับการทำงานของอินฟราเรดเซนเซอร์	53

สารบัญตาราง

ตารางที่		หน้า
2.1	เงื่อนไขการเคลื่อนที่ของหุ่นยนต์เดินตามเส้น	18
4.1	เปรียบเทียบความคลาดเคลื่อนก่อนและหลังปรับเทียบเซนเซอร์	46
4.2	การทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้	49
4.3	เปรียบเทียบอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลาของมอเตอร์	53



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากเชื้อไวรัสโควิด-19 ที่แพร่ระบาดหนักในหลายประเทศทั่วโลก รวมถึงประเทศไทยที่ประสบปัญหาและยังไม่สามารถควบคุมสถานการณ์ให้เป็นปกติได้ จึงมีมาตรการในการเว้นระยะห่างทางสังคม หรือ social distancing เพื่อช่วยในการป้องกันการแพร่กระจายของเชื้อโรค แพทย์และพยาบาลที่ทำการรักษาผู้ป่วยอย่างใกล้ชิดจึงต้องสวมชุดป้องกันเชื้อไวรัส แต่ยังคงมีความเสี่ยงอยู่มากที่จะได้รับเชื้อไวรัส ปรินูญานิพนธ์นี้มีวัตถุประสงค์เพื่อช่วยลดการพบปะระหว่างแพทย์กับ ผู้ติดเชื้อไวรัสโควิด-19 และช่วยลดการใช้ชุดป้องกันเชื้อไวรัส ที่สามารถแพร่เชื้อผ่านระบบทางเดินหายใจได้ จึงได้ทำการศึกษาและจัดทำรถขนส่งเวชภัณฑ์อัตโนมัติขึ้นมา เพื่ออำนวยความสะดวกแก่บุคลากรทางการแพทย์ในการขนส่งยาและเวชภัณฑ์ให้กับผู้ป่วยที่อยู่ในระหว่างการรักษาหรือการกักตัว และเพื่อลดโอกาสและเพิ่มประสิทธิภาพการควบคุมการแพร่กระจายของเชื้อไวรัสสู่บริเวณต่าง ๆ

ดังนั้นจึงทำการศึกษาการเพิ่มประสิทธิภาพในการขนส่งยาและเวชภัณฑ์ภายในโรงพยาบาลสนาม ผู้จัดทำจึงมุ่งใช้เครื่องมือและแนวคิดแบบระบบหุ่นยนต์เดินตามเส้น มาใช้ในการวิเคราะห์ปรับปรุงและแก้ไขปัญหา เพื่อเพิ่มประสิทธิภาพในการแจกจ่ายยาและเวชภัณฑ์และลดโอกาสในการพบปะกับผู้ติดเชื้อโควิด-19 โดยไม่จำเป็น ซึ่งจะช่วยลดความเสี่ยงของแพทย์ผู้ทำการรักษาได้ โดยระบบแบ่งออกเป็น 2 ส่วนหลัก ๆ คือการพัฒนารถขนส่งเวชภัณฑ์ที่สามารถเดินตามเส้นโดยการประมวลผลด้วย Arduino Mega 2560 ที่ทำงานร่วมกับอินฟราเรดเซนเซอร์และอัลตราโซนิกเซนเซอร์และแสดงผลการควบคุมผ่านหน้าจอ LCD ขนาดเล็ก และระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชันซึ่งบอร์ดไมโครคอนโทรลเลอร์ ESP32 Cam จะรับคำสั่งจากเว็บแอปพลิเคชันผ่าน Wi-Fi แล้วจึงนำข้อมูลไปประมวลผลเพื่อสั่งการให้มอเตอร์ทำงานผ่านโมดูลขับมอเตอร์

1.2 วัตถุประสงค์

- 1) เพื่ออำนวยความสะดวกให้กับบุคลากรทางการแพทย์ในการส่งเวชภัณฑ์และสิ่งของจำเป็นให้กับผู้ป่วย
- 2) เพื่อช่วยลดความเสี่ยงของบุคลากรทางการแพทย์ในการปฏิบัติหน้าที่
- 3) เพื่อออกแบบและสร้างอุปกรณ์ที่ควบคุมรถขนส่งให้สามารถเดินตามเส้นและควบคุมจากระยะไกลผ่านเว็บแอปพลิเคชันได้

1.3 ขอบเขตของปริญญาานิพนธ์

ได้ทำการออกแบบและสร้างอุปกรณ์ที่ควบคุมรถขนส่งให้สามารถเดินตามเส้นและควบคุมจากระยะไกล ซึ่งระบบประกอบด้วยส่วนของระบบเดินตามเส้นที่ควบคุมด้วยเซนเซอร์และไมโครคอนโทรลเลอร์ที่สามารถเดินทางไปถึงเป้าหมายและกลับมาที่จุดเริ่มต้นได้ และส่วนของระบบควบคุมระยะไกลที่ควบคุมผ่านเว็บแอปพลิเคชัน ซึ่ง ESP32 Cam จะรับคำสั่งจากเว็บแอปพลิเคชันผ่าน Wi-Fi และรถขนส่งสามารถบรรทุกน้ำหนักได้ 20 กิโลกรัม และยังสามารถเคลื่อนที่ไปได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

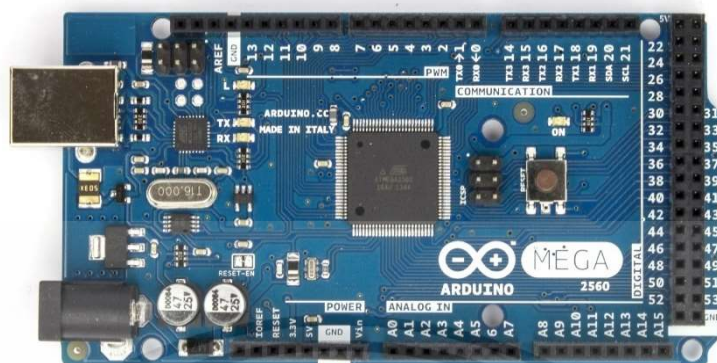
ปริญญาานิพนธ์ “รถส่งเวชภัณฑ์อัตโนมัติ” ได้จัดทำเพื่อออกแบบและสร้างอุปกรณ์ที่ควบคุมรถขนส่งระบบเดินตามเส้นและการควบคุมทางไกล ซึ่งมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560

Arduino Mega 2560 ดังแสดงในรูปที่ 2.1 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนามาจาก ATmega2560 ซึ่งเป็นบอร์ดรุ่นใหญ่ในกลุ่ม Arduino มีสัญญาณดิจิทัลแบบขาเข้าและขาออก 54 ขา โดยมี 14 ขา สามารถใช้เป็นสัญญาณขาออกแบบ Pulse-width Modulation (PWM) ได้ มีสัญญาณขาเข้าแบบแอนะล็อก 16 ขา มีการติดต่อสื่อสารดิจิทัลระหว่างอุปกรณ์ Universal asynchronous receiver-transmitter (UARTs) 4 ขา ทำงานที่ความถี่ 16 เมกะเฮิร์ตซ์ สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้อุปกรณ์แปลงไฟฟ้าจากไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรงเพื่อเริ่มต้นใช้งาน และมีปุ่ม reset สามารถต่อเข้ากับ shields ที่ออกแบบเพื่อใช้งานกับ Arduino Duemilanove หรือ Diecimila [1] คุณสมบัติของ Arduino Mega 2560 มีดังนี้ [2]

- 1) ไมโครคอนโทรลเลอร์: Atmega2560
- 2) แรงดันไฟฟ้า: 5 โวลต์
- 3) แรงดันไฟฟ้าขาเข้าที่แจ๊ค SOURCE: 7 โวลต์ ถึง 12 โวลต์
- 4) สัญญาณดิจิทัลแบบขาเข้าและขาออก: 54 ขา
- 5) สัญญาณขาเข้าแบบแอนะล็อก: 16 ขา
- 6) ไฟฟ้ากระแสตรง ต่อ I / O Pin: 20 มิลลิแอมแปร์
- 7) ไฟฟ้ากระแสตรงสำหรับ 3.3 โวลต์ Pin: 50 มิลลิแอมแปร์
- 8) หน่วยความจำแฟลช: 256 กิโลไบต์
- 9) SRAM: 8 กิโลไบต์
- 10) EEPROM: 4 กิโลไบต์
- 11) ความเร็วสัญญาณนาฬิกา: 16 เมกะเฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 [3]

2.2 ESP32 Cam with OV2640 Camera Module

โมดูล ESP32 Cam ดังที่แสดงในรูปที่ 2.2 ใช้ความสามารถของ ESP32 ในการสื่อสารแบบไร้สาย ทั้งเป็นตัวรับสัญญาณ (Station-mode) และตัวปล่อยสัญญาณ (Access point-mode) ได้ในตัวเดียว มีการติดตั้งกล้องเพื่อใช้ประโยชน์จาก CPU ที่เร็วกว่า Arduino หลายเท่า ในการรับข้อมูลจากกล้อง OV2640 ที่มีความละเอียดถึง 2 ล้านพิกเซล เพื่อส่งข้อมูลแบบไร้สายหรือจัดเก็บข้อมูลรูปภาพไว้ใน MicroSD Card ซึ่ง ESP32 Cam รุ่นนี้เป็นโมดูลที่ไม่มี Micro USB ให้ในการใช้ป้อนชุดคำสั่งจึงต้องใช้ USB to TTL เป็นตัวช่วย [4] โดยคุณสมบัติของ ESP32 Cam มีดังนี้ [5]

- 1) โมดูล Wi-Fi 802.11b/g/n ร่วมกับ BT/BLE SoC
- 2) CPU dual-core 32 บิต พลังงานต่ำ เหมาะสำหรับการใช้งานโปรเซสเซอร์
- 3) ความถี่หลักสูงถึง 240 เมกะเฮิร์ตซ์ และกำลังประมวลผลไมโครคอนโทรลเลอร์สูงสุด 600 DMIPS (Dhrystone Million Instructions per Second)
- 4) มีหน่วยความจำของ SRAM 520 กิโลไบต์ PSRAM 4 เมกะไบต์
- 5) รองรับอินเตอร์เฟซ เช่น UART/SPI/I2C/PWM/ADC/DAC
- 6) สนับสนุนกล้อง OV2640 และ OV7670 แพลชในตัว
- 7) รองรับการอัปโหลดภาพผ่าน Wi-Fi
- 8) รองรับการ์ด TF
- 9) สนับสนุนโหมดพักหน้าจอ
- 10) มี Lwip และ FreeRTOS ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

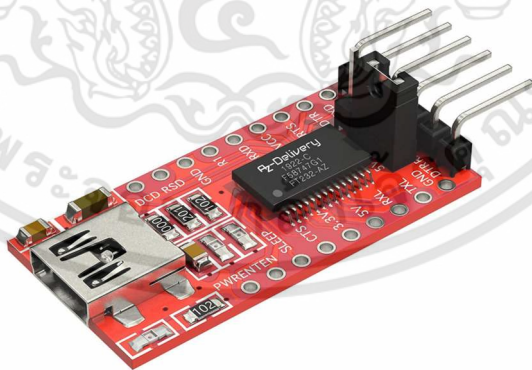
- 11) สนับสนุนโหมดการทำงาน STA/AP/STA ร่วมกับ AP
- 12) รองรับ Smart Config/AirKiss



รูปที่ 2.2 ESP32 Cam with OV2640 Camera Module [6]

2.3 โมดูล FT232RL

โมดูล FT232RL ดังแสดงในรูปที่ 2.3 เป็นโมดูลที่แปลง USB เป็น TTL UART สำหรับป้อนชุดคำสั่งลงบอร์ด Arduino รุ่นที่ไม่มี TTL หรือช่องเชื่อมต่อ USB รวมไปถึงตัวโมดูล ESP8266 Wi-Fi STB (SET TOP BOX เครื่องรับดาวเทียม) ผ่านทางขา Rx และ Tx [7]



รูปที่ 2.3 โมดูล FT232RL [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวน

เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวนด้วยแสงอินฟราเรด (Infrared : IR) ดังแสดงในรูปที่ 2.4 ใช้หลักการ Reflective Optical Sensor คือการปล่อยแสงอินฟราเรดจากหลอดออกไปให้สะท้อนกับวัตถุหรือสีอื่นต่าง ๆ ซึ่งจะมีคุณสมบัติดูดซับและสะท้อนแสงได้ต่างกัน แสงอินฟราเรดจึงสะท้อนกลับเข้ามาโดยมีโฟโตทรานซิสเตอร์ (Phototransistor) คอยตรวจจับแสงอินฟราเรดที่สะท้อนกลับมา จากนั้นจึงแปลงเป็นสัญญาณไฟฟ้าส่งออกมา [9] เมื่อวัตถุอยู่ใกล้จะมีแสงสะท้อนกลับมามากกว่าวัตถุที่อยู่ไกล จึงสามารถนำมาตรวจสอบวัตถุผ่าน หรือใช้ตรวจจับเส้นสีขาวหรือดำได้ โดยระยะการตรวจจับที่ 1 มิลลิเมตร ถึง 8 มิลลิเมตร รองรับแรงดันไฟฟ้าขนาด 3.3 โวลต์ ถึง 5 โวลต์ และให้สัญญาณขาออกมา 2 รูปแบบ คือ รูปแบบดิจิทัลสามารถปรับค่าที่ต้องการได้ เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการจะส่งสัญญาณดิจิทัลเลขฐานสองด้วยบิต 1 หากไม่ถึงระดับจะให้ส่งบิต 0 แทน และรูปแบบสัญญาณขาเข้าแบบแอนะล็อกด้วยระดับสัญญาณทางดิจิทัลขนาด 1024 บิต (0 ถึง 1023) ด้วยการเทียบระดับแรงดันไฟฟ้าในช่วง 0 โวลต์ ถึง 5 โวลต์ [10]

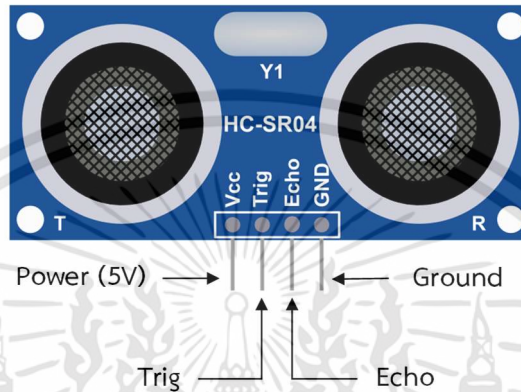


รูปที่ 2.4 เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวน [11]

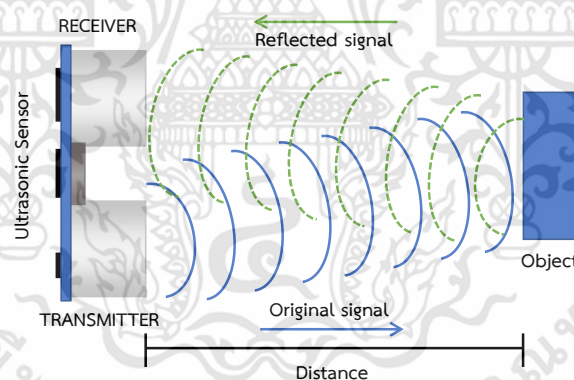
2.5 เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04

เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิกโมดูล HC-SR04 ดังแสดงในรูปที่ 2.5 เป็นโมดูลวัดระยะทางที่ใช้หลักการสะท้อนของคลื่นอัลตราโซนิก โดยหลักการวัดระยะทางด้วยคลื่นอัลตราโซนิก คือ การส่งคลื่นอัลตราโซนิกจำนวนหนึ่งออกไปสะท้อนกับวัตถุที่อยู่ด้านหน้ากลับมายังตัวรับสัญญาณ โดยระยะทางที่วัดได้จะสัมพันธ์กับระยะเวลาที่คลื่นอัลตราโซนิกเคลื่อนที่ไปกระทบ

วัตถุและสะท้อนกลับมายังตัวรับ ดังแสดงในรูปที่ 2.6 เมื่อทราบระยะเวลาที่คลื่นอัลตราโซนิกสะท้อนกลับ จึงนำระยะเวลาดังกล่าว มาเปรียบเทียบกับอัตราเร็วที่เสียงสามารถเดินทางได้ในอากาศ โดยอัตราเร็วเสียงที่เดินทางได้ในอากาศสามารถหาได้จากสมการที่ (2.1) [12]



รูปที่ 2.5 เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04 [13]



รูปที่ 2.6 หลักการวัดระยะด้วยคลื่นอัลตราโซนิก [13]

$$c = 331.4 + (0.606 \times T) \quad (2.1)$$

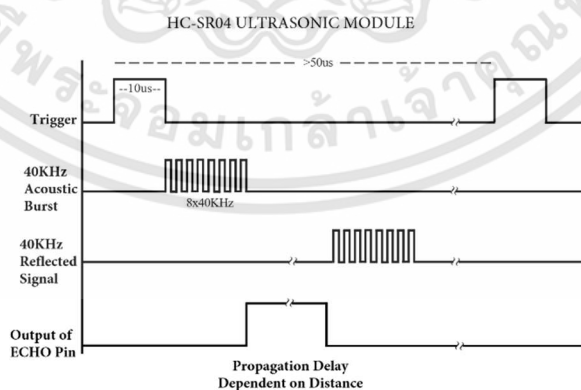
เมื่อ c คือ อัตราเร็วของเสียงในอากาศ มีหน่วยเป็น เมตรต่อวินาที

T คือ อุณหภูมิในหน่วยองศาเซลเซียส

โดยเซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04 มีคุณสมบัติ ดังนี้ [14]

- 1) แรงดันไฟฟ้าที่ใช้ในการทำงาน: 5 โวลต์
- 2) กระแสไฟฟ้าที่ใช้: 15 มิลลิแอมแปร์
- 3) ความถี่ในการทำงาน: 40 กิโลเฮิร์ตซ์
- 4) ระยะสูงสุดที่สามารถวัดได้: 4 เมตร (400 เซนติเมตร)
- 5) ระยะต่ำสุดที่สามารถวัดได้: 0.02 เมตร (2 เซนติเมตร)
- 6) ความแม่นยำ: ± 3 มิลลิเมตร
- 7) องศาในการวัด: 15 องศา
- 8) Trigger Input Signal: 10 ไมโครวินาที TTL Pulse

แผนผังเวลาสำหรับเซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04 แสดงดังรูปที่ 2.7 เมื่อมีการส่งสัญญาณเข้าขา Trig สถานะทางลอจิกเป็น LOW จากนั้นจึงเริ่มทรiggerสัญญาณ โดยให้ขา Trig มีสถานะเป็น HIGH และคงสถานะไว้เป็นระยะเวลาอย่างน้อย 10 ไมโครวินาที จากนั้นปรับสถานะเป็น LOW วงจรภายในจะเริ่มสร้างความถี่ 40 กิโลเฮิร์ตซ์ จำนวน 8 ลูกคลื่นออกไป โดยใช้ความถี่จากคริสตอลเป็นตัวอ้างอิง ซึ่งตัวส่งที่เปรียบเสมือนลำโพงจะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นวิ่งกลับมาที่ตัวรับที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้าจะผ่านตัวประมวลผล แล้วให้ค่าสัญญาณขาออกทางขา Echo เมื่อมีการส่งสัญญาณ HIGH กลับมาที่ขา Echo ให้เริ่มนับเวลาที่สัญญาณเป็น HIGH และเมื่อสัญญาณขา Echo กลับเป็น LOW ให้สิ้นสุดการนับเวลาแล้วจึงนำค่าเวลาที่นับได้ไปคำนวณหาระยะทางได้จากสมการที่ (2.2) [12]



รูปที่ 2.7 แผนผังเวลาสำหรับเซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04 [15]

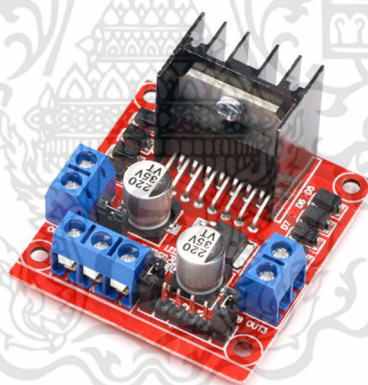
$$D = \frac{c \times t}{2} \quad (2.2)$$

เมื่อ D คือ ระยะทาง มีหน่วยเป็น เซนติเมตร

t คือ ระยะเวลาที่สัญญาณเดินทาง มีหน่วยเป็น ไมโครวินาที

2.6 โมดูลขับมอเตอร์ L298N

โมดูลขับมอเตอร์ L298N ดังแสดงในรูปที่ 2.8 เป็นโมดูลสำหรับขับมอเตอร์ 2 ตัวแบบอิสระต่อกัน ใช้กระแสไฟฟ้าสูงสุด 2 แอมแปร์ ในการขับมอเตอร์แต่ละตัว ซึ่งสามารถควบคุมโดยสัญญาณเป็น PWM เพื่อควบคุมความเร็วมอเตอร์ได้ ใช้ชิปจาก ST coparation พร้อมแผงระบายความร้อน มีแรงดันไฟฟ้า 2 ชุดในการใช้งานคือ แรงดันไฟฟ้าขนาด 5 โวลต์ สำหรับวงจร และแรงดันไฟฟ้าขนาด 5 โวลต์ ถึง 35 โวลต์ สำหรับมอเตอร์ เหมาะสำหรับนำไปใช้กับหุ่นยนต์ ระบบรถอัจฉริยะขับมอเตอร์ต่าง ๆ สามารถใช้งานร่วมกับ Arduino NodeMCU หรือ Raspberry Pi ได้ [16]



รูปที่ 2.8 โมดูลขับมอเตอร์ L298N [16]

2.7 โมดูลเสียง Active Buzzer Module

Buzzer คือลำโพงเปียโซหรือลำโพงแม่เหล็กที่ภายในอุปกรณ์มีวงจรกำเนิดความถี่ (Oscillator) อยู่ภายในตัว สามารถสร้างเสียงเตือนในรูปแบบต่าง ๆ ได้ ซึ่งโมดูลเสียง Active Buzzer Module ดังแสดงในรูปที่ 2.9 สามารถสร้างเสียงเตือนได้ง่าย โดยการต่อขา I/O ของ

Active Buzzer กับขา I/O ของ Arduino หรือไมโครคอนโทรลเลอร์ได้โดยตรง และต่อกับขา VCC ของ Active Buzzer กับแรงดันไฟฟ้ากระแสตรงขนาด 3.3 โวลต์ ถึง 5 โวลต์ และขา GND กับ GND ในการสั่งให้ Active Buzzer ส่งเสียง [17]



รูปที่ 2.9 โมดูลเสียง Active Buzzer Module [17]

2.8 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ (Motor) หมายถึงเครื่องกลไฟฟ้าชนิดหนึ่งทำหน้าที่เปลี่ยนแปลงพลังงานไฟฟ้าเป็นพลังงานกล โดยปัจจุบันมีทั้งมอเตอร์ที่รองรับไฟฟ้ากระแสสลับ (AC Motor) และมอเตอร์ที่รองรับไฟฟ้ากระแสตรง (DC Motor)

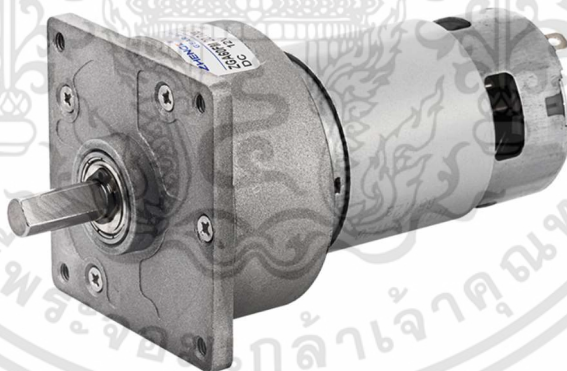
2.8.1 หลักการของมอเตอร์ไฟฟ้ากระแสตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เมื่อแรงดันกระแสไฟฟ้าตรงไหลผ่านมอเตอร์ ส่วนหนึ่งจะมีแปรงถ่านที่เป็นตัวรับแรงดันไฟฟ้าจากแหล่งจ่าย และจ่ายผ่านขั้วรับแรงดันเพื่อส่งไปให้ขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้นและกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) ทำให้เกิดแม่เหล็กขั้วเหนือ และขั้วใต้ และเกิดสนามแม่เหล็กในขณะเดียวกัน ตามคุณสมบัติเส้นแรงแม่เหล็กจะไม่ตัดกัน หากมีทิศทางตรงข้ามจะหักล้างกันและจะเสริมกันเมื่อเส้นแรงแม่เหล็กมีทิศทางเดียวกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ซึ่งวางอยู่บนแกนเพลลาเกิดการหมุนตัว ในขณะที่อาร์มาเจอร์หมุนจะเรียกว่า โรเตอร์ (Rotor) ซึ่งการหมุนที่เกิดจากอำนาจของเส้นแรงแม่เหล็กทั้งสองมีปฏิริยาต่อกัน และทำให้ขดลวดอาร์มาเจอร์หรือโรเตอร์หมุน เป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (Fleming left hand rule)

2.8.2 ชนิดของมอเตอร์ไฟฟ้ากระแสตรง

1) มอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม (Series motor) คือมอเตอร์ที่ต่อขดลวดสนามแม่เหล็กอนุกรมกับอาร์มาเจอร์ของมอเตอร์ชนิดนี้เรียกว่า ซีรีส์ฟิลด์ (Series field) มีคุณลักษณะที่ดีคือ ให้แรงบิดสูง ได้รับความนิยม เป็นต้น มอเตอร์อนุกรมมีความเร็วรอบสูงเมื่อไม่พิจารณาโหลด ในขณะที่เดียวกันหากมีโหลดเพิ่มขึ้นจะส่งผลให้อัตราเร็วลดลงแต่ไม่ส่งผลเสียกับขดลวดของมอเตอร์ จากคุณสมบัตินี้จึงนิยมนำมาใช้กับเครื่องใช้ไฟฟ้าในบ้านหลายชนิด เช่น เครื่องดูดฝุ่น เครื่องผสมอาหาร สว่านไฟฟ้า จักรเย็บผ้า และเครื่องเป่าผม มอเตอร์กระแสตรงแบบอนุกรมทำงานได้อย่างมีประสิทธิภาพเมื่อใช้งานในสถานะที่มีโหลดมากจะทำให้ความเร็วรอบนั้นลดลง ในทางกลับกันในสถานะไม่มีโหลดจะทำให้มอเตอร์มีความเร็วรอบสูงและอาจเกิดอันตรายได้ ดังนั้นก่อนเริ่มใช้งานมอเตอร์แบบอนุกรมจึงต้องมีโหลดต่ออยู่เสมอ

2) มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt motor) มอเตอร์แบบขนานนี้ ขดลวดสนามแม่เหล็ก (Field coil) จะต่อขนานกับขดลวดชุดอาร์มาเจอร์ มอเตอร์ดังกล่าวมีคุณลักษณะคือ มีความเร็วคงที่ แรงบิดเริ่มหม่นต่ำ ด้วยเหตุนี้จึงทำให้มอเตอร์ชนิดนี้เหมาะกับชิ้นงานที่ต้องการความเร็วรอบที่คงที่ เช่น พัดลมเพราะพัดลมต้องการความเร็วคงที่และเปลี่ยนความเร็วได้ง่าย มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน ดังแสดงในรูปที่ 2.10 [18]



รูปที่ 2.10 มอเตอร์ ZGA60FM [19]

2.9 จอแสดงผล

จอแสดงผล Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอแสดงผล LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD [24] จอแสดงผล LCD จะแบ่งเป็น 2 แบบตามลักษณะการแสดงผลดังนี้

1) Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16×2 ตัวอักษร หมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20×4 ตัวอักษร จะหมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด การเชื่อมต่อกับจอ Character LCD จะมีด้วยกัน 2 แบบ คือ

(ก) การเชื่อมต่อแบบขนาน เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า

(ข) การเชื่อมต่อแบบอนุกรม เป็นการเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนาน มาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 4 เส้น ก็ทำให้หน้าจอแสดงผลข้อความต่าง ๆ ออกมาได้ จอ LCD ที่มีการเชื่อมต่อแบบอนุกรม ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 หน้าจอ LCD พร้อมโมดูล I2C ขนาด 16×2 [21]

2) Graphic LCD เป็นจอแสดงผลที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอขึ้นแสง หรือปล่อยแสงออกไป ทำให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128×64 พิกเซล หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 พิกเซล และมีจุดตามแนวตั้ง 64 พิกเซล [22]

2.10 คีย์แพดปุ่มกด

คีย์แพดปุ่มกด (4x4 Matrix Keypad Module) ดังแสดงในรูปที่ 2.12 แบบ 4x4 ชนิดนี้ประกอบไปด้วยปุ่ม 16 ปุ่ม ที่ถูกจัดเรียงกันในลักษณะอาร์เรย์ แบ่งเป็นแถวแนวนอน (Rows) และแถวแนวตั้ง (Columns) ซึ่งหลักการในการตรวจสอบว่าปุ่มกดใดถูกกดบ้างในขณะนั้น จะใช้วิธีการสแกนปุ่มกด (Key scan) ที่ละหลักจนครบทุกหลัก แล้วพิจารณาว่ามีการตอบสนองออกมาในรูปแบบใดบ้าง เช่น ถ้ามีการกดเลข 1 ในขณะที่จ่ายแรงดันไฟฟ้า 5 โวลต์ ไปที่หลักที่ 1 จะมีเพียงแถวแรกเท่านั้นที่สามารถอ่านค่าแรงดันไฟฟ้าสูงได้ นอกจากนั้นจะพิจารณาเป็นค่าแรงดันไฟฟ้าต่ำ หรือถ้ามีการกดปุ่ม # จะสามารถอ่านค่าแรงดันไฟฟ้าสูงได้เพียงหลักที่ 3 เท่านั้น ซึ่งจะพบว่าการตอบสนองกลับมาจากแถวที่ 4 ดังนั้นเมื่อพบว่าเป็นการตรวจสอบหลักที่ 3 และมีแถวที่ 4 ตอบสนอง จะหมายถึงปุ่ม # [23]



รูปที่ 2.12 คีย์แพดปุ่มกด 4x4 Matrix (Black Keypad) [24]

2.11 มิเตอร์วัดแรงดันแบตเตอรี่

มิเตอร์วัดแรงดันแบตเตอรี่พร้อมแถบวัดระดับแรงดันแบตเตอรี่ ดังแสดงในรูปที่ 2.13 มีปุ่มเปิดปิดการแสดงผล โดยมีแถบบาร์ทั้งหมด 8 แถบเพื่อแสดงสถานะระดับแรงดันแบตเตอรี่ มีความคลาดเคลื่อนประมาณ 3 เปอร์เซ็นต์ ซึ่งแถบสถานะจะแสดงผลดังนี้ [25]

- 1) แรงดันแบตเตอรี่มากกว่า 9.7 โวลต์ คิดเป็น 10 เปอร์เซ็นต์ แสดง 1 แถบ
- 2) แรงดันแบตเตอรี่มากกว่า 10.2 โวลต์ คิดเป็น 25 เปอร์เซ็นต์ แสดง 2 แถบ
- 3) แรงดันแบตเตอรี่มากกว่า 10.7 โวลต์ คิดเป็น 40 เปอร์เซ็นต์ แสดง 3 แถบ
- 4) แรงดันแบตเตอรี่มากกว่า 11.1 โวลต์ คิดเป็น 50 เปอร์เซ็นต์ แสดง 4 แถบ
- 5) แรงดันแบตเตอรี่มากกว่า 11.7 โวลต์ คิดเป็น 60 เปอร์เซ็นต์ แสดง 5 แถบ
- 6) แรงดันแบตเตอรี่มากกว่า 12.2 โวลต์ คิดเป็น 75 เปอร์เซ็นต์ แสดง 6 แถบ
- 7) แรงดันแบตเตอรี่มากกว่า 12.6 โวลต์ คิดเป็น 90 เปอร์เซ็นต์ แสดง 7 แถบ
- 8) แรงดันแบตเตอรี่มากกว่า 13.0 โวลต์ คิดเป็น 100 เปอร์เซ็นต์ แสดง 8 แถบ



รูปที่ 2.13 มิเตอร์วัดแรงดันแบตเตอรี่ [25]

2.12 สวิตช์ปุ่มกด

สวิตช์ปุ่มกด (Push Button Switch) เป็นอุปกรณ์พื้นฐานทางไฟฟ้าเมื่อใช้งานต้องกดปุ่มสวิตช์ลงไป การควบคุมตัดต่อสวิตช์ ต้องกดปุ่มที่อยู่ส่วนกลางสวิตช์ ซึ่งทำหน้าที่ตัดและต่อวงจรทางไฟฟ้า เพื่อใช้ในการควบคุมการทำงานของมอเตอร์ หรือการควบคุมการทำงานของเครื่องจักรต่าง ๆ ใช้ได้กับอุตสาหกรรมทั่วไป มีทั้งแบบมีไฟ และทึบแสง โดยสวิตช์ปุ่มกดที่ใช้ในการเริ่มการ

ทำงาน (Start) เรียกว่าสวิตช์ปกติเปิด (Normally Open) หรือที่เรียกว่า NO และสวิตช์ปุ่มกดหยุดการทำงาน (Stop) เรียกว่าสวิตช์ปกติปิด (Normally Close) หรือที่เรียกว่า NC ซึ่งสามารถแบ่งสวิตช์ปุ่มกดได้ 2 ประเภท คือ

1) แบบกดติดปล่อยดับ หรือแบบสปริงรีเทิร์น ดังแสดงในรูปที่ 2.14 เมื่อมีการกดสวิตช์ปุ่มกดหน้าสัมผัสดังกล่าวจะเปลี่ยนสถานะจาก NO เป็น NC หรือจาก NC จะเป็น NO แต่เมื่อปล่อยมือออกจากสวิตช์ปุ่มกดหน้าสัมผัสจะกลับสู่สภาวะปกติในตำแหน่งเดิม โดยมีแรงผลักดันจากสปริงให้สวิตช์ปุ่มกดเข้าสู่สภาวะปกติ เป็นประเภทที่นิยมใช้ เนื่องจากมีฟังก์ชันการทำงานที่ไม่ซับซ้อน



รูปที่ 2.14 สวิตช์ปุ่มกดแบบกดติดปล่อยดับ [26]

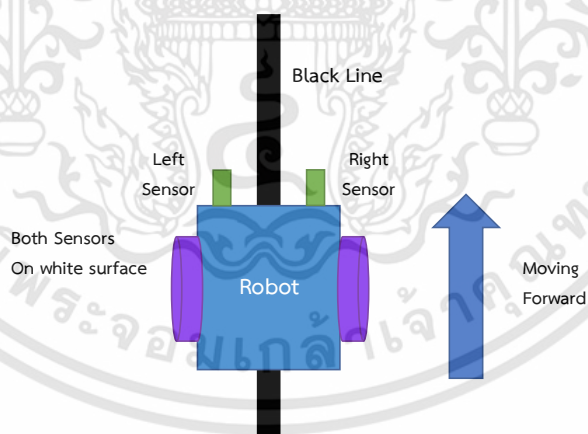
2) แบบกดติดกดดับ หรือแบบ push on / push off ดังแสดงในรูปที่ 2.15 เมื่อมีการกดสวิตช์ปุ่มกดหน้าสัมผัสดังกล่าวจะเปลี่ยนสถานะจาก NO เป็น NC หรือจาก NC จะเป็น NO แต่เมื่อปล่อยมือออกจากสวิตช์ปุ่มกดหน้าสัมผัสจะถูกล็อกไว้โดยกลไกของสวิตช์และคงสถานะไว้ ซึ่งสามารถกลับสู่สภาวะปกติในตำแหน่งเดิมได้โดยกดสวิตช์ปุ่มกดอีกครั้งทำให้คลายล็อก จะมีแรงผลักดันจากสปริงให้สวิตช์ปุ่มกดเข้าสู่สภาวะปกติ เป็นประเภทที่นิยมใช้ในฟังก์ชันการทำงานแบบไม่ซับซ้อน เช่น Start/Stop สายพานลำเลียง เป็นต้น [27]



รูปที่ 2.15 สวิตช์ปุ่มกดแบบกดติดกดดับ [28]

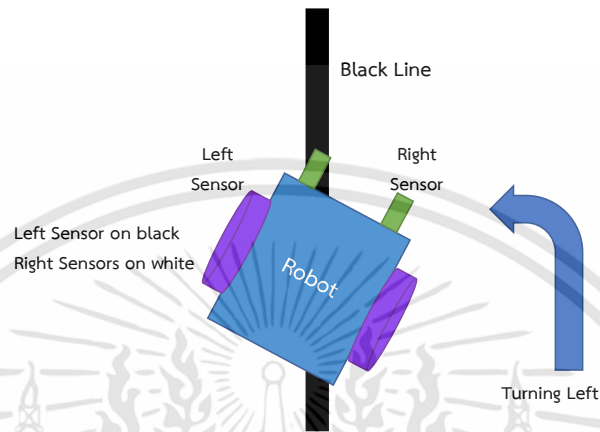
2.13 หลักการทำงานของรถเดินตามเส้น

โดยอ้างอิงหลักการทำงานของหุ่นยนต์เดินตามเส้นทำงานร่วมกับอินฟราเรดเซนเซอร์ 2 ตัว คือเซนเซอร์ด้านซ้ายและเซนเซอร์ด้านขวาของเส้นสีดำ เมื่อเซนเซอร์ทั้งด้านซ้ายและด้านขวาตรวจสอบพบพื้นสีขาวลำแสงสามารถสะท้อนกลับมาได้ทั้งคู่ ให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า ดังแสดงในรูปที่ 2.16



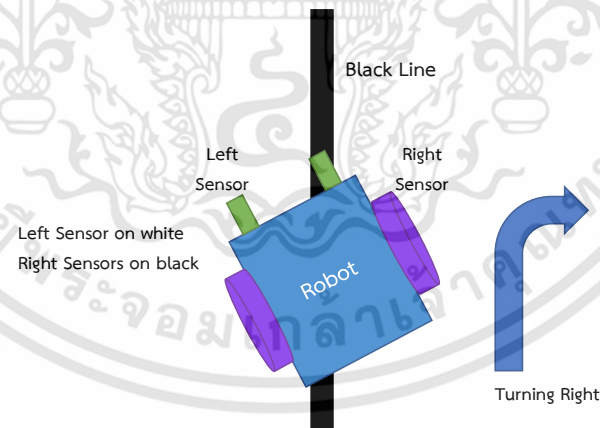
รูปที่ 2.16 หุ่นยนต์เคลื่อนที่ไปข้างหน้า [29]

หากเซนเซอร์ด้านซ้ายตรวจพบพื้นสีดำ และเซนเซอร์ด้านขวาตรวจพบพื้นสีขาว
ให้หุ่นยนต์เลี้ยวไปทางด้านซ้าย ดังแสดงในรูปที่ 2.17



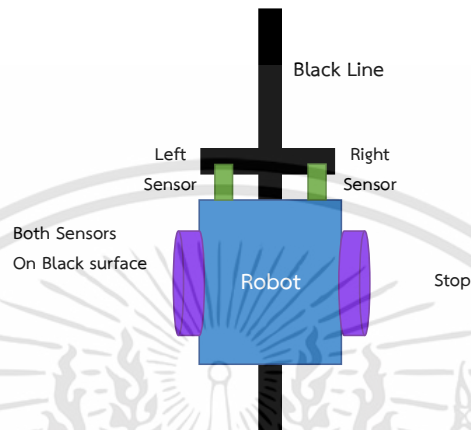
รูปที่ 2.17 หุ่นยนต์เลี้ยวไปทางด้านซ้าย [29]

หากเซนเซอร์ด้านขวาตรวจพบพื้นสีดำ และเซนเซอร์ด้านซ้ายตรวจพบพื้นสีขาว
ให้หุ่นยนต์เลี้ยวไปทางด้านขวา ดังแสดงในรูปที่ 2.18



รูปที่ 2.18 หุ่นยนต์เลี้ยวไปทางด้านขวา [29]

หากเซนเซอร์ทั้งสองตรวจพบเส้นสีดำทั้งคู่ ให้หุ่นยนต์หยุดเคลื่อนที่ ดังแสดงในรูปที่ 2.19



รูปที่ 2.19 หุ่นยนต์หยุดเคลื่อนที่ [29]

โดยสรุป มี 4 เงื่อนไขที่หุ่นยนต์เดินตามเส้น ซึ่ง Arduino รับค่าจากเซนเซอร์ด้านซ้ายและด้านขวา คือ (0,0) (0,1) (1,0) และ(1,1) แสดงตามตารางที่ 2.1 [29]

ตารางที่ 2.1 เงื่อนไขการเคลื่อนที่ของหุ่นยนต์เดินตามเส้น

สัญญาณลอจิกขาเข้า		สัญญาณลอจิกขาออก				การเคลื่อนที่ของหุ่นยนต์
เซนเซอร์ซ้าย	เซนเซอร์ขวา	มอเตอร์ซ้าย		มอเตอร์ขวา		
LS	RS	LM1	LM2	RM1	RM2	
0	0	0	0	0	0	เดินหน้า
0	1	1	0	0	0	เลี้ยวขวา
1	0	0	0	1	0	เลี้ยวซ้าย
1	1	1	0	1	0	หยุด

2.14 เว็บแอปพลิเคชัน

แอปพลิเคชันที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้ในเว็บเบราว์เซอร์ได้โดยตรง ซึ่งไม่ต้องติดตั้งแอปพลิเคชันลงเครื่องสมาร์ทโฟน ทำให้ภาพรวมมีการใช้ทรัพยากรค่อนข้างต่ำ จึงสามารถเปิดใช้งานได้รวดเร็ว ซึ่งภายในตัวเว็บแอปพลิเคชันถูกเพิ่มประสิทธิภาพให้มีการทำงานรวดเร็วกว่าการเปิดแอปพลิเคชันแบบปกติ จึงมีหน้าต่างที่เป็นมิตรและใช้งานค่อนข้างง่าย ซึ่งในปัจจุบันนี้มีเว็บแอปพลิเคชันที่หลากหลายทั้งในรูปแบบการใช้งานระดับโลก ระดับมหาวิทยาลัย ไปจนถึงระดับบริษัท หรือการมีเว็บแอปพลิเคชันเป็นของตัวเองเช่นกัน [30]

2.14.1 ส่วนประกอบของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันประกอบด้วยการทำงานของระบบเทคโนโลยีต่าง ๆ มากมาย เช่น โปรแกรมเว็บแอปพลิเคชัน (Web Application) เว็บเซิร์ฟเวอร์ (Web Server) เว็บเซิร์ฟเวอร์ซอฟต์แวร์ (Web Server Software) ฐานข้อมูล (Database) เว็บเบราว์เซอร์ (Web Browser) และอื่น ๆ ซึ่งแต่ละส่วนมีหน้าที่และการทำงานที่แตกต่างกันออกไปดังแสดงในรูปที่ 2.20 ส่วนประกอบของเว็บแอปพลิเคชันสามารถแยกออกเป็นสองส่วนหลัก ๆ คือ เทคโนโลยีฝั่งผู้ใช้งาน (Client-side Technology) และเทคโนโลยีฝั่งเซิร์ฟเวอร์ (Server-side Technology)

2.14.1.1 เทคโนโลยีฝั่งผู้ใช้งาน (Client-side Technology)

1) เว็บเบราว์เซอร์ (Web Browser) เป็นซอฟต์แวร์ที่ผู้ใช้งานใช้ในการเข้าถึงเว็บแอปพลิเคชัน โดยเมื่อเริ่มต้นผู้ใช้งานทำการใส่ URL หรือชื่อของเว็บไซต์ที่ต้องการเข้าใช้งาน เมื่อเบราว์เซอร์ได้รับชื่อของเว็บไซต์ก็จะทำการแปลงจากชื่อของเว็บไซต์เป็น IP address ผ่านทาง DNS ให้เป็นข้อมูลที่ใช้ในการแสดงผลให้กับผู้ใช้งาน

2) ส่วนต่อความสามารถเว็บและเบราว์เซอร์ (Web Plugin และ Browser Add-on/Extension) โดย Web Plugin (ส่วนต่อความสามารถเว็บ) คือโปรแกรมที่ถูกเขียนให้ทำงานร่วมกับเว็บเบราว์เซอร์ Web Plugin ที่เป็นที่รู้จัก เช่น Adobe Flash PDF reader Silverlight Java Applet และอื่น ๆ และ Browser Add-one/Extension (ส่วนเพิ่มความสามารถเบราว์เซอร์) เป็นโปรแกรมที่ใช้ในการเพิ่มความสามารถให้กับเบราว์เซอร์ ซึ่งส่วนเพิ่มความสามารถเบราว์เซอร์เหล่านี้จะเน้นเพิ่มความสามารถให้กับเบราว์เซอร์มากกว่าการประมวลผลเนื้อหาเว็บไซต์

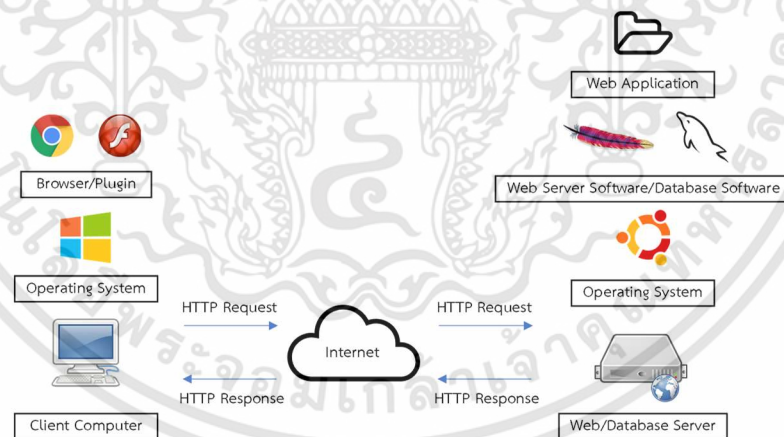
3) ระบบปฏิบัติการ (Operating System) ทำหน้าที่ในการจัดการกับทรัพยากรของเครื่องคอมพิวเตอร์ โดยสร้างการเชื่อมต่อระหว่างเครื่องผู้ใช้งานและเครื่องเซิร์ฟเวอร์ ดังนั้นการทำงานของระบบปฏิบัติการจะเป็นสิ่งที่ผู้ใช้งานมองไม่เห็นแต่ก็มีความสำคัญ

2.14.1.2 เทคโนโลยีฝั่งเซิร์ฟเวอร์ (Server-side Technology)

1) เว็บแอปพลิเคชัน (Web Application) เป็นหัวใจหลักของเว็บไซต์ เนื่องจากทำหน้าที่ติดต่อกับผู้ใช้งาน รับและแสดงข้อมูล ประมวลผลข้อมูล จัดการข้อมูลในฐานข้อมูล และอื่น ๆ ซึ่งเว็บแอปพลิเคชันเป็นซอฟต์แวร์ที่ให้บริการผู้ใช้งานทั่วโลกผ่านอินเทอร์เน็ต

2) เว็บเซิร์ฟเวอร์ซอฟต์แวร์ (Web Server Software) เป็นโปรแกรมที่ทำงานอยู่บนเว็บเซิร์ฟเวอร์ซึ่งหน้าที่หลักของเว็บเซิร์ฟเวอร์ซอฟต์แวร์ คือการประมวลผล HTTP request ที่ได้รับมาและตอบกลับด้วย HTTP response อย่างไรก็ตามเว็บเซิร์ฟเวอร์ซอฟต์แวร์ที่ได้รับความนิยมอย่างแพร่หลายมากที่สุดในปัจจุบันคือ Apache HTTP server และผู้ใช้งานมักจะใช้คู่กับ PHP และ MySQL

3) ระบบปฏิบัติการ (Operating System) ระบบปฏิบัติการฝั่งของเซิร์ฟเวอร์มีหน้าที่ในการจัดการกับทรัพยากรของเครื่องเซิร์ฟเวอร์ อย่างเช่น CPU memory และ bandwidth เป็นต้น เนื่องจากเว็บแอปพลิเคชันเป็นบริการที่เปิดให้ผู้ใช้งานเข้าถึงได้ตลอดเวลา ดังนั้น ระบบปฏิบัติการบนเซิร์ฟเวอร์จึงต้องมีความเสถียรและสามารถจัดการกับทรัพยากรของเครื่องได้อย่างมีประสิทธิภาพ [31]



รูปที่ 2.20 ส่วนประกอบของเว็บแอปพลิเคชัน [32]

2.14.2 ลักษณะการทำงานของเว็บแอปพลิเคชัน

การทำงานของเว็บแอปพลิเคชันนั้นโปรแกรมส่วนหนึ่งจะวางตัวอยู่บน Rendering Engine ซึ่งตัว Rendering Engine จะทำหน้าที่นำเอาชุดคำสั่งหรือรูปแบบโครงสร้างข้อมูลที่ใช้ในการแสดงผล นำมาแสดงผลบนพื้นที่ส่วนหนึ่งในจอภาพ โปรแกรมส่วนที่วางตัวอยู่บน Rendering Engine จะทำหน้าที่เปลี่ยนแปลงแก้ไขสิ่งที่แสดงผล จัดการตรวจสอบข้อมูลที่รับเข้ามาเบื้องต้นและการประมวลผลบางส่วนแต่ส่วนการทำงานหลักจะวางตัวอยู่บนเซิร์ฟเวอร์ ในลักษณะเว็บแอปพลิเคชันแบบเบื้องต้น โดยฝั่งเซิร์ฟเวอร์จะประกอบไปด้วยเว็บเซิร์ฟเวอร์ซึ่งทำหน้าที่เชื่อมต่อกับเครื่องลูกข่ายตามโปรโตคอล HTTP/HTTPS โดยนอกจากเว็บเซิร์ฟเวอร์จะทำหน้าที่ส่งไฟล์ที่เกี่ยวข้องกับการแสดงผลตามมาตรฐาน HTTP ตามปกติทั่วไปแล้ว เว็บเซิร์ฟเวอร์จะมีส่วนประมวลผลซึ่งอาจจะเป็นตัวแปลภาษา เช่น Script Engine ของภาษา PHP หรืออาจจะมีการติดตั้ง .NET Framework ซึ่งมีส่วนแปลภาษา CLR (Common Language Runtime) ที่ใช้แปลภาษา intermediate จากโค้ดที่เขียนด้วย VB.NET หรือ C#.NET หรืออาจจะเป็น J2EE ที่มีส่วนแปลไบต์โค้ดของคลาสที่ได้จากโปรแกรมภาษา Java เป็นต้น [33] ดังแสดงในรูปที่ 2.21



รูปที่ 2.21 ส่วนประมวลผลของเว็บแอปพลิเคชัน [34]

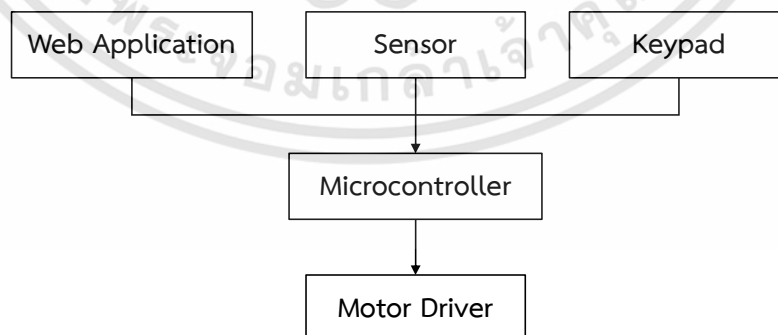
บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

รถส่งเวชภัณฑ์อัตโนมัติประกอบด้วยอินฟราเรดเซนเซอร์ที่ใช้สำหรับกำหนดทิศทางของรถขนส่ง อัลตราโซนิกเซนเซอร์ที่ใช้สำหรับตรวจจับวัตถุในระยะไกลเพื่อสร้างความปลอดภัยและคีย์แพดซึ่งจะเป็นอุปกรณ์ที่กำหนดจุดหมายปลายทางให้กับรถขนส่ง ซึ่งจะมีการประมวลผลข้อมูลผ่านบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 และ ESP32 Cam ที่ใช้สำหรับควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน หลักการทำงานของระบบจะเริ่มทำงานก็ต่อเมื่อมีการป้อนหมายเลขห้องที่ต้องการผ่านคีย์แพด ซึ่งจะแสดงผลจุดหมายปลายทางผ่านหน้าจอ LCD ผ่านการประมวลผลของบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 และเมื่อได้รับสัญญาณแอนะล็อกขาเข้าที่มาจากอินฟราเรดเซนเซอร์และอัลตราโซนิกเซนเซอร์ โดยรถขนส่งจะเดินตามเส้นสีดำและเลี้ยวเมื่อพบทางแยกตามเงื่อนไขในชุดคำสั่งและในขณะที่เคลื่อนที่เมื่อมีวัตถุมาตัดขวางผ่านอัลตราโซนิกเซนเซอร์ในระยะ 50 เซนติเมตร อัลตราโซนิกเซนเซอร์จะส่งการผ่านโมดูลขับมอเตอร์ให้มอเตอร์ทั้งสองตัวหยุดการทำงานและในส่วนของการควบคุมผ่านเว็บแอปพลิเคชันจะเป็นการควบคุมการเดินหน้า ถอยหลัง เลี้ยวซ้ายหรือเลี้ยวขวา ซึ่งบอร์ดไมโครคอนโทรลเลอร์ ESP32 Cam จะรับคำสั่งจากเว็บแอปพลิเคชันผ่าน Wi-Fi แล้วจึงนำข้อมูลไปประมวลผลเพื่อสั่งการให้มอเตอร์ทำงานผ่านโมดูลขับมอเตอร์ บล็อกไดอะแกรมของระบบแสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของรถส่งเวชภัณฑ์อัตโนมัติ

3.1.2 การออกแบบและประกอบรถส่งเวชภัณฑ์

โครงสร้างของรถส่งเวชภัณฑ์จะมีความกว้าง 35 เซนติเมตร ความยาว 55 เซนติเมตร และมีความสูงจากพื้น 77 เซนติเมตร โดยทำการติดตั้งอินฟราเรดเซนเซอร์ทั้ง 5 ตัวไว้ด้านหน้า บริเวณใต้พื้นสแตนเลสโดยใช้ไม้เป็นจุดยึดอินฟราเรดเซนเซอร์เพื่อให้เซนเซอร์ทุกตัวอยู่ในระนาบเดียวกันดังแสดงในรูปที่ 3.2 ติดตั้งอัลตราโซนิกเซนเซอร์ไว้ด้านหน้าของรถขนส่งโดยมีความสูงจากพื้น 32 เซนติเมตร ติดตั้ง ESP32 Cam ไว้ด้านหน้าของรถขนส่งที่ความสูงจากพื้น 62 เซนติเมตร ติดตั้งอินฟราเรดเซนเซอร์ไว้ด้านข้างของรถขนส่งที่ความสูงจากพื้น 62 เซนติเมตร และจัดเก็บอุปกรณ์ส่วนที่เหลือไว้ภายในกล่อง ซึ่งด้านหลังของรถส่งเวชภัณฑ์แสดงดังรูปที่ 3.3 ด้านหน้าของรถส่งเวชภัณฑ์แสดงดังรูปที่ 3.4 และด้านข้างของรถส่งเวชภัณฑ์แสดงดังรูปที่ 3.5

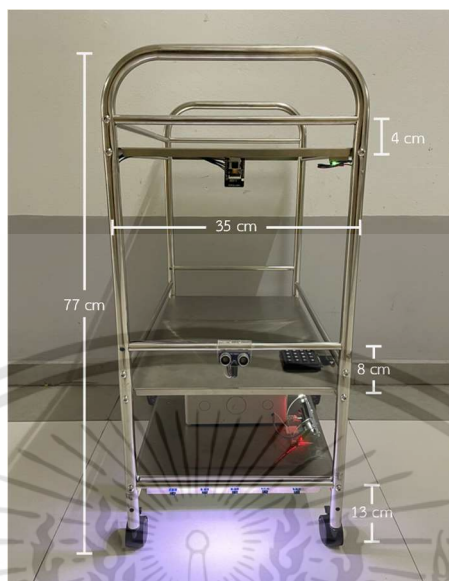


รูปที่ 3.2 ตำแหน่งการติดตั้งอินฟราเรดเซนเซอร์

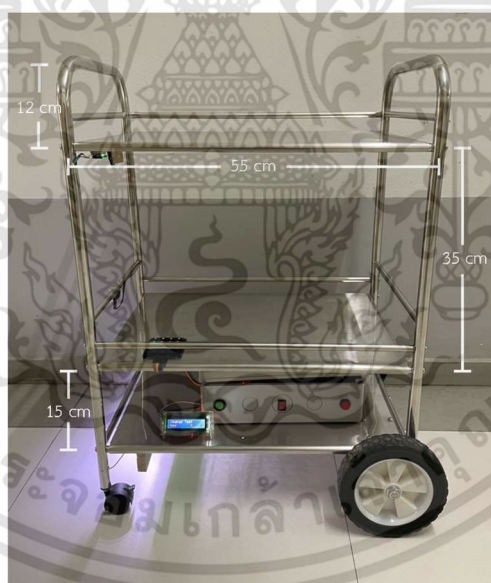


รูปที่ 3.3 ด้านหลังของรถส่งเวชภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ด้านหน้าของรถส่งเวชภัณฑ์



รูปที่ 3.5 ด้านข้างของรถส่งเวชภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

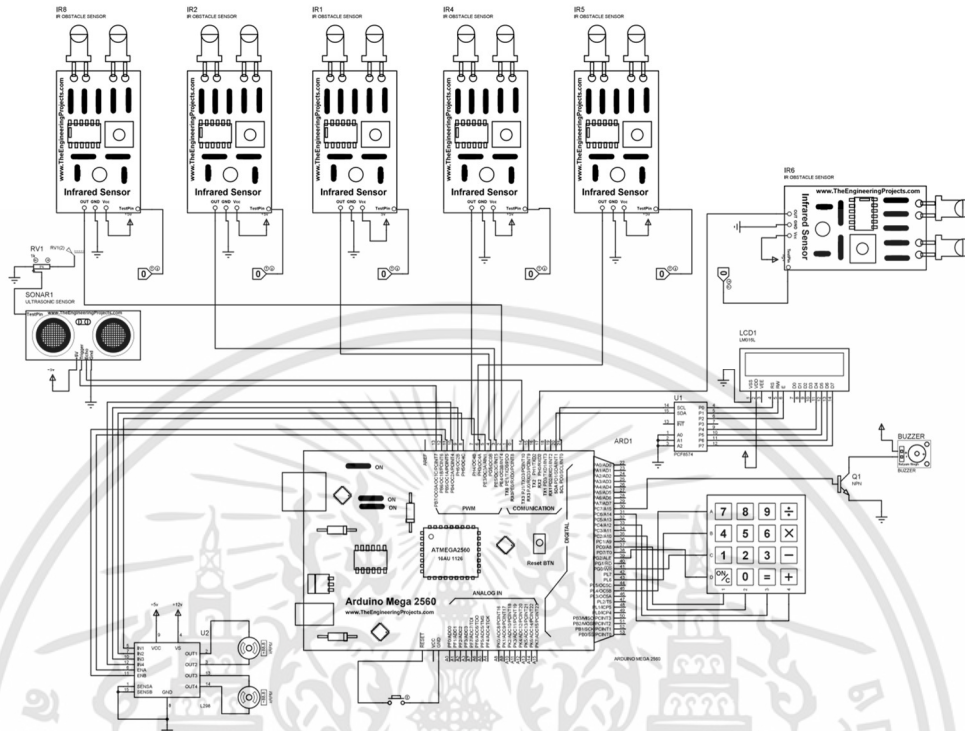
3.1.3 การออกแบบวงจร

3.1.3.1 การออกแบบระบบควบคุมการเดินตามเส้น

สำหรับการเคลื่อนที่แบบเดินตามเส้น โดยมีการรับค่าการตรวจจับเส้นจากอินฟราเรดเซนเซอร์ทั้ง 5 ตัว คือ เซนเซอร์ด้านซ้าย 2 ตัว เซนเซอร์ด้านขวา 2 ตัว และเซนเซอร์ตรงกลางตำแหน่งของเส้นสีดำ 1 ตัว และรับค่าการตรวจจับสิ่งกีดขวางจากอัลตราโซนิกเซนเซอร์ด้านหน้า 1 ตัว โดยนำค่าที่ได้มาประมวลผลผ่านบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 และส่งคำสั่งไปยังส่วนของการส่งออกข้อมูลซึ่งประกอบไปด้วยโมดูลขับเคลื่อนมอเตอร์ L298N และมอเตอร์ไฟฟ้ากระแสตรง โดยโมดูลขับเคลื่อนมอเตอร์ L298N จะรับคำสั่งส่วนประมวลผลข้อมูลและไปควบคุมการหมุนของมอเตอร์ เพื่อให้รถขนส่งเคลื่อนที่ไปตามเงื่อนไขที่กำหนด บล็อกไดอะแกรมการทำงานของระบบการเดินตามเส้นแสดงได้ดังรูปที่ 3.6 และภาพการเชื่อมต่อวงจรของระบบการเดินตามเส้นแสดงได้ดังรูปที่ 3.7



รูปที่ 3.6 บล็อกไดอะแกรมการทำงานของระบบควบคุมการเดินตามเส้น



รูปที่ 3.7 ภาพการเชื่อมต่อวงจรของระบบการเดินตามเส้น

3.1.3.2 การออกแบบระบบควบคุมระยะไกล

สำหรับการควบคุมการเคลื่อนที่ผ่านเว็บแอปพลิเคชัน โดยเว็บแอปพลิเคชัน จะส่งข้อมูลไปประมวลผลผ่านบอร์ดไมโครคอนโทรลเลอร์ ESP32 Cam และส่งคำสั่งไปยังโมดูลขับเคลื่อน L298N และมอเตอร์ไฟฟ้ากระแสตรง โดยโมดูลขับเคลื่อน L298N จะรับคำสั่งส่วนประมวลผลข้อมูลและไปควบคุมการหมุนของมอเตอร์ เพื่อให้รถขนส่งเคลื่อนที่ตามเงื่อนไขคำสั่งบล็อกไดอะแกรมการทำงานของระบบระยะไกลแสดงได้ดังรูปที่ 3.8



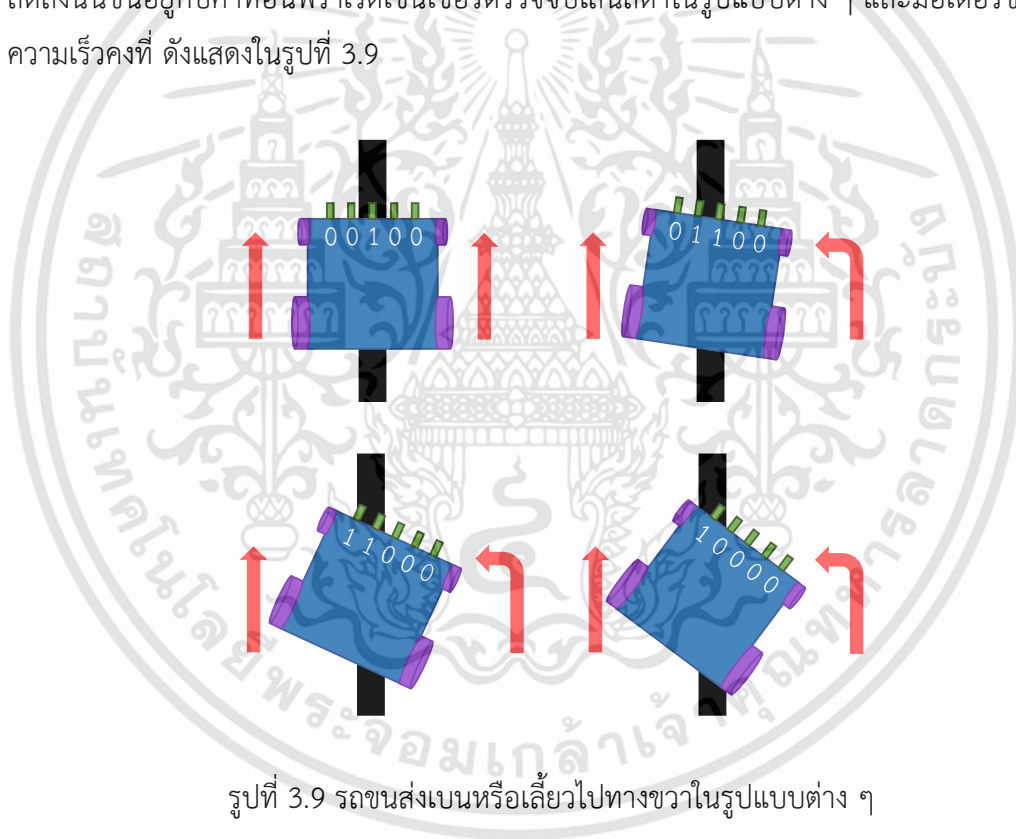
รูปที่ 3.8 บล็อกไดอะแกรมการทำงานของระบบควบคุมระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การออกแบบการเคลื่อนที่ของรถส่งเวชภัณฑ์

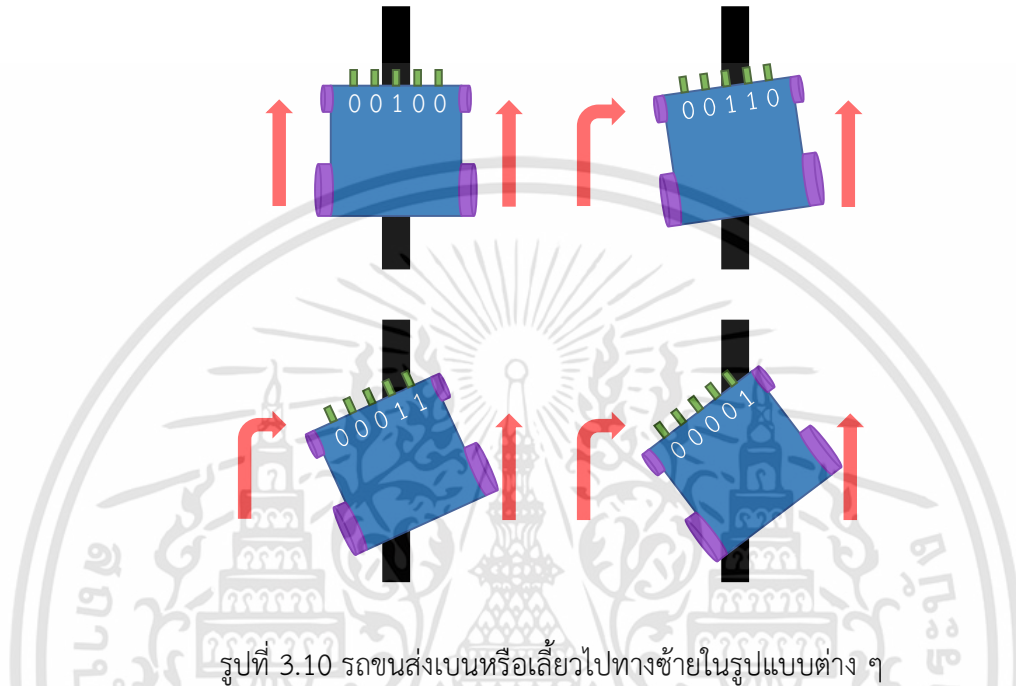
โดยหลักการทำงานของรถส่งเวชภัณฑ์อัตโนมัติแบบเดินตามเส้นจะทำงานร่วมกับอินฟราเรดเซนเซอร์ 5 ตัว คือ เซนเซอร์ซ้าย 2 ตัว เซนเซอร์ขวา 2 ตัว และเซนเซอร์ตรงกลาง ตำแหน่งของเส้นสีดำ 1 ตัว เมื่ออินฟราเรดเซนเซอร์ทั้ง 5 ตัว ตรวจสอบแสงสีขาวถ้าแสงจะสามารถสะท้อนกลับมาทำให้รถขนส่งเคลื่อนที่ไปข้างหน้า

หากอินฟราเรดเซนเซอร์ด้านซ้ายตรวจพบพื้นสีดำ และอินฟราเรดเซนเซอร์ด้านขวาตรวจพบพื้นสีขาว โดยมีสัญญาณลอจิกขาเข้าเป็น [01100 11000 และ 10000] ให้รถขนส่งเลี้ยวไปทางซ้าย ซึ่งการเลี้ยวซ้ายจะทำการเลี้ยวโดยการลดความเร็วของมอเตอร์ด้านซ้ายลง แต่ความเร็วที่ลดลงนั้นขึ้นอยู่กับค่าที่อินฟราเรดเซนเซอร์ตรวจจับเส้นสีดำในรูปแบบต่าง ๆ และมอเตอร์ขวาจะมีความเร็วคงที่ ดังแสดงในรูปที่ 3.9



หากอินฟราเรดเซนเซอร์ด้านขวาตรวจพบพื้นสีดำ และอินฟราเรดเซนเซอร์ด้านซ้ายตรวจพบพื้นสีขาว โดยมีสัญญาณลอจิกขาเข้าเป็น [00110 00011 และ 00001] ให้รถขนส่งเลี้ยวไปทางขวา ซึ่งการเลี้ยวขวาจะทำการเลี้ยวโดยการลดความเร็วของมอเตอร์ด้านขวาลง แต่ความเร็วที่

ลดลงนั้นขึ้นอยู่กับค่าที่อินฟราเรดเซนเซอร์ตรวจจับเส้นสีดำในรูปแบบต่าง ๆ และมอเตอร์ซ้ายจะมีความเร็วคงที่ ดังแสดงในรูปที่ 3.10



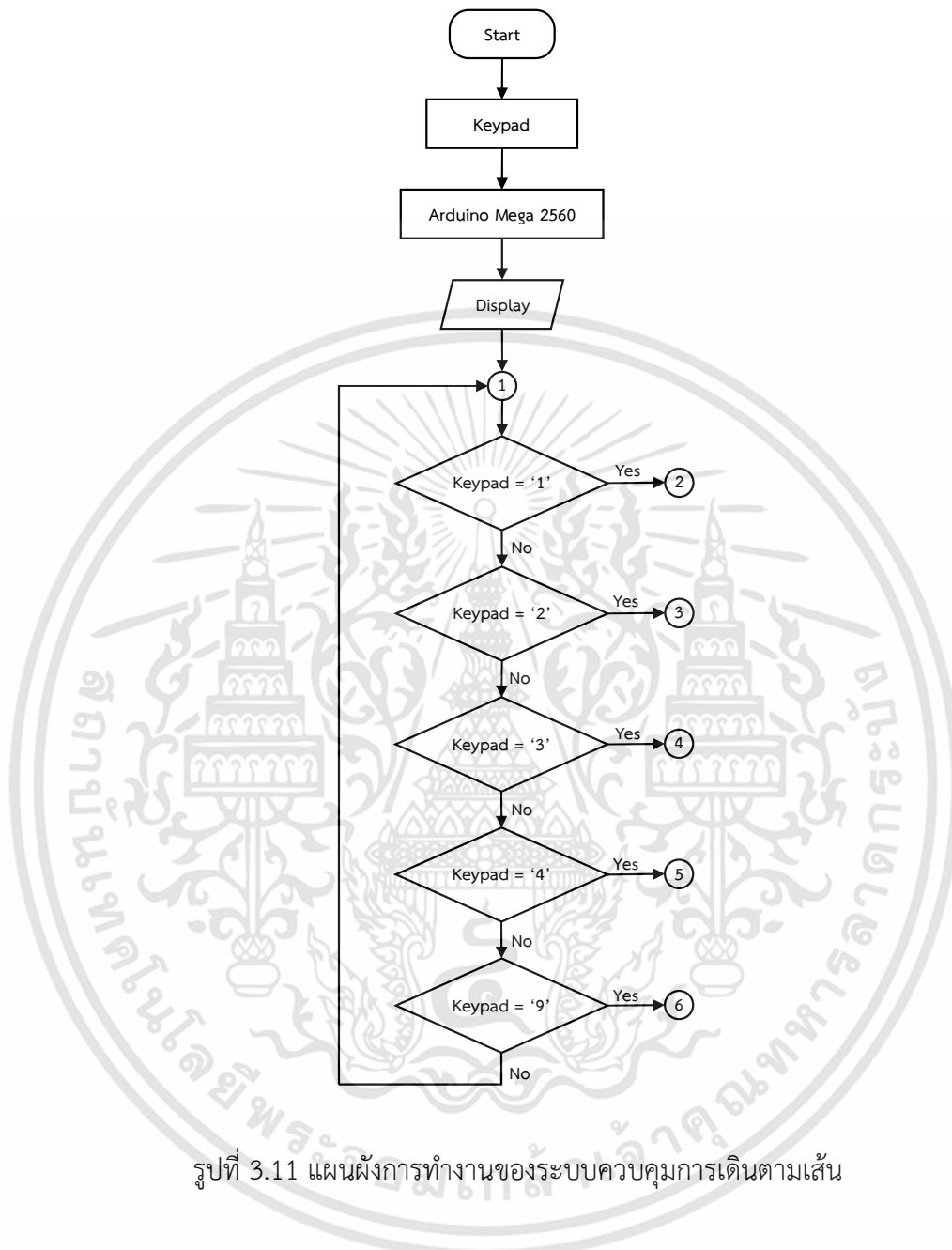
รูปที่ 3.10 รถขนส่งเบนหรือเลี้ยวไปทางซ้ายในรูปแบบต่าง ๆ

3.1.5 การออกแบบชุดคำสั่งการควบคุมการทำงานของรถส่งเวชภัณฑ์

ระบบควบคุมการทำงานจะแบ่งออกเป็น 2 ระบบ คือ ระบบควบคุมการเดินตามเส้น และระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน และแผนผังการทำงานของระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

3.1.5.1 ระบบควบคุมการเดินตามเส้น

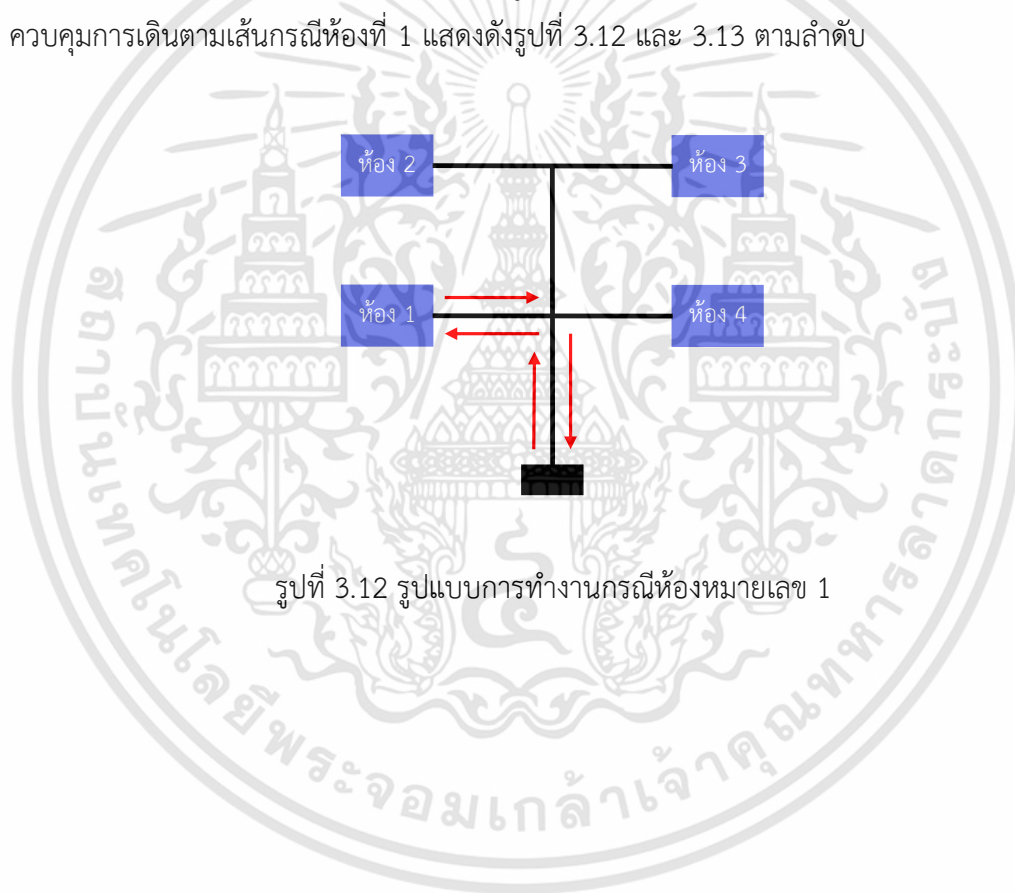
เริ่มต้นการทำงานโดยกดหมายเลขห้อง 1 2 3 หรือ 4 ที่คีย์แพด และกดเครื่องหมาย # เพื่อยืนยัน หลังจากนั้นบอร์ด Arduino Mega 2560 จะประมวลผลและส่งค่าไปยังจอแสดงผลเพื่อแสดงค่าที่ได้กดยืนยัน เมื่อเงื่อนไขถูกต้องรถขนส่งจะเคลื่อนที่ตามเส้นไปด้านหน้าตามเงื่อนไขที่กำหนดไว้ แสดงดังรูปที่ 3.11



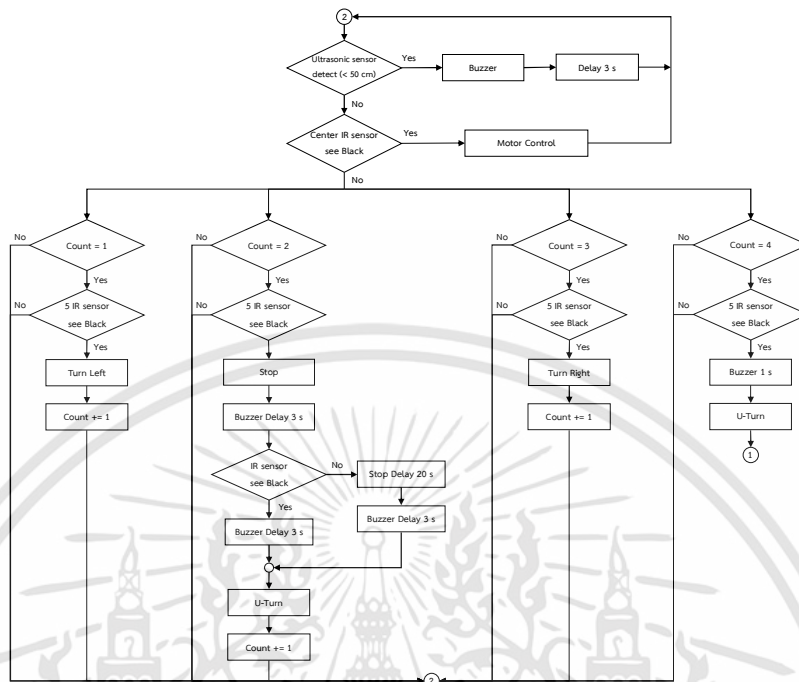
รูปที่ 3.11 แผนผังการทำงานของระบบควบคุมการเดินตามเส้น

กรณีกีดหมายเลข 1 ที่คีย์แพด เมื่อต้องการเดินทางไปยังห้องหมายเลข 1 เจ็อนไขการเคลื่อนที่ คือ เมื่ออัลตราโซนิกเซนเซอร์ตรวจพบสิ่งกีดขวางภายในระยะ 50 เซนติเมตร รถขนส่งจะหยุดทำงานพร้อมส่งเสียงแจ้งเตือนว่ามีสิ่งกีดขวาง หากสิ่งกีดขวางอยู่ในระยะที่มากกว่า 50 เซนติเมตร รถขนส่งจะเคลื่อนที่ตามเส้นต่อไปตามระบบควบคุมของมอเตอร์ จนกว่าจะพบสี่แยกและอินฟราเรดเซนเซอร์ทั้ง 5 ตัว อ่านค่าได้ 11111 ซึ่งหมายความว่ารถขนส่งได้พบสี่แยกและ

ทำการนับค่า (count) เพิ่มขึ้นทีละ 1 ค่า หากไม่พบรถขนส่งจะเคลื่อนที่เดินตามเส้นต่อไป เมื่อพบทางแยกที่ 1 จะทำการเลี้ยวซ้าย และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 2 จะทำการหยุดรถขนส่งพร้อมส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ถ้าได้รับสัญญาณตอบรับจากอินฟราเรดเซนเซอร์ ด้านบนจะส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที จากนั้นทำการกลับรถขนส่ง แต่หากไม่ได้รับสัญญาณตอบรับรถขนส่งจะหยุดเป็นเวลา 20 วินาที และส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ก่อนทำการกลับรถขนส่ง และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 3 จะทำการเลี้ยวขวา และนับค่า (count) เพิ่ม 1 ค่า และเมื่อพบทางแยกที่ 4 จะทำการส่งเสียงแจ้งเตือน 1 ครั้ง และทำการกลับรถเพื่อรับค่าใหม่ ซึ่งเป็นอันสิ้นสุดการทำงาน รูปแบบการทำงานและแผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 1 แสดงดังรูปที่ 3.12 และ 3.13 ตามลำดับ



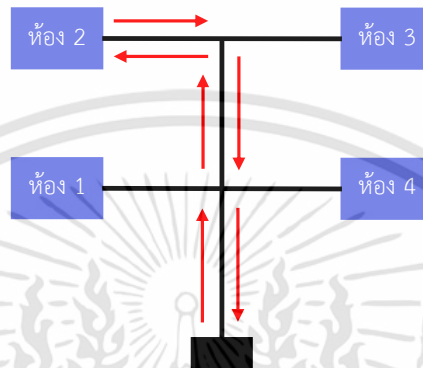
รูปที่ 3.12 รูปแบบการทำงานกรณีห้องหมายเลข 1



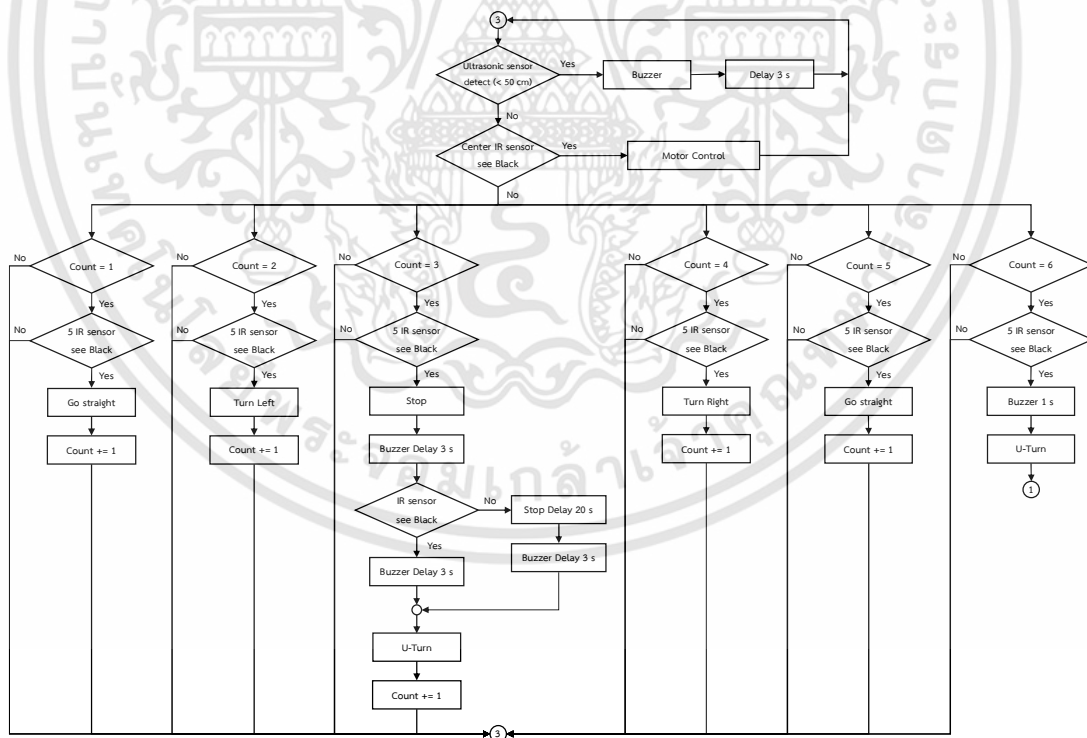
รูปที่ 3.13 แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 1

กรณีกำหนดหมายเลข 2 ที่คีย์แพด เมื่อต้องการเดินทางไปยังห้องหมายเลข 2 เงื่อนไขการเคลื่อนที่ คือ เมื่ออัลตราโซนิกเซนเซอร์ตรวจพบสิ่งกีดขวางภายในระยะ 50 เซนติเมตร รถขนส่งจะหยุดทำงานพร้อมส่งเสียงแจ้งเตือนว่ามีสิ่งกีดขวาง หากสิ่งกีดขวางอยู่ในระยะที่มากกว่า 50 เซนติเมตร รถขนส่งจะเคลื่อนที่ตามเส้นต่อไปตามระบบควบคุมของมอเตอร์ จนกว่าจะพบสี่แยกและอินฟราเรดเซนเซอร์ทั้ง 5 ตัวอ่านค่าได้ 11111 ซึ่งหมายความว่ารถขนส่งได้พบสี่แยกและทำการนับค่า (count) เพิ่มขึ้นทีละ 1 ค่า หากไม่พบรถขนส่งจะเคลื่อนที่เดินตามเส้นต่อไป เมื่อพบทางแยกที่ 1 รถขนส่งจะเคลื่อนที่ตามเส้นต่อไป และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 2 จะทำการเลี้ยวซ้าย และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 3 จะทำการหยุดรถขนส่งพร้อมส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ถ้าได้รับสัญญาณตอบรับจากอินฟราเรดเซนเซอร์ด้านบนจะส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที จากนั้นทำการกลับรถขนส่ง แต่หากไม่ได้รับสัญญาณตอบรับรถขนส่งจะหยุดเป็นเวลา 20 วินาที และส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ก่อนทำการกลับรถขนส่ง และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 4 จะทำการเลี้ยวขวา และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 5 รถขนส่งจะเคลื่อนที่ตามเส้นต่อไป และนับค่า (count) เพิ่ม 1 ค่า

และเมื่อพบทางแยกที่ 6 จะทำการส่งเสียงแจ้งเตือน 1 ครั้ง และทำการกลับรถเพื่อรับค่าใหม่ ซึ่งเป็นอันสิ้นสุดการทำงาน รูปแบบการทำงานและแผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 2 แสดงดังรูปที่ 3.14 และ 3.15 ตามลำดับ



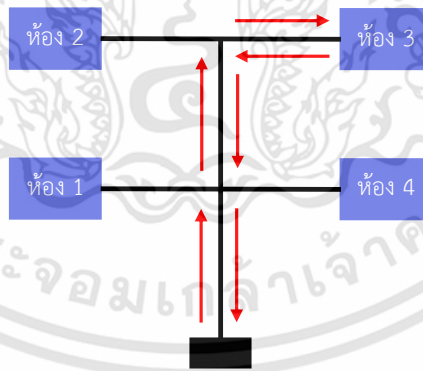
รูปที่ 3.14 รูปแบบการทำงานกรณีห้องหมายเลข 2



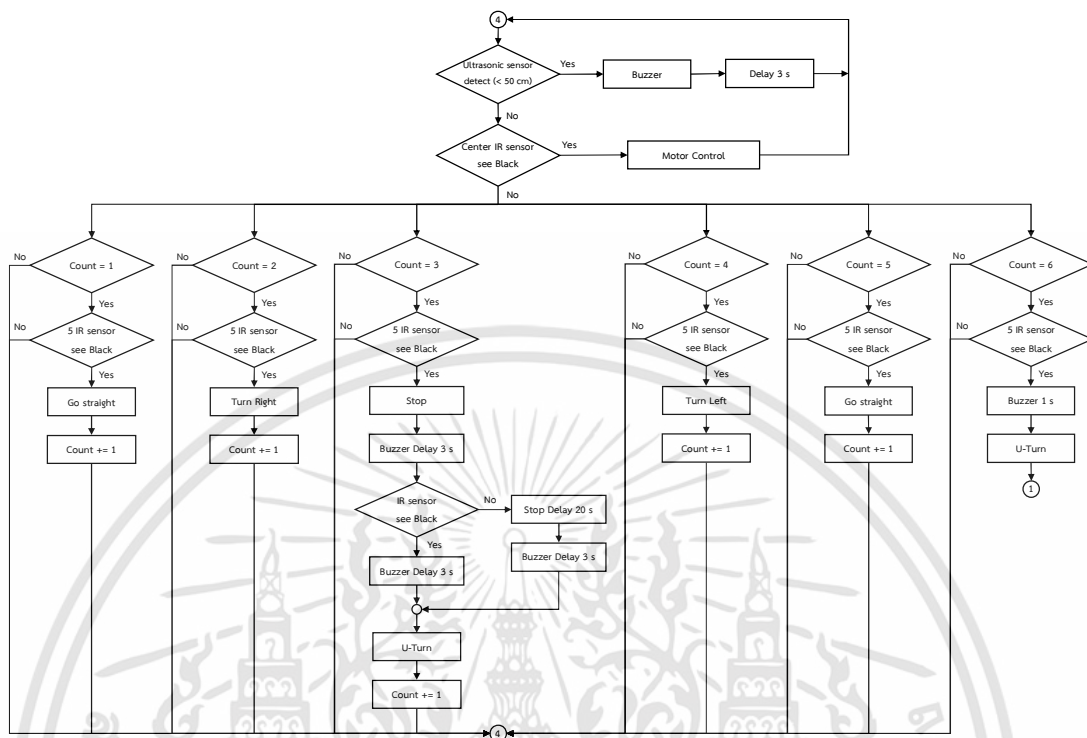
รูปที่ 3.15 แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีศึกษานี้หมายถึง 3 ที่คีย์แพด เมื่อต้องการเดินทางไปยังห้องหมายเลข 3 เงื่อนไขการเคลื่อนที่ คือ เมื่ออัลตราโซนิกเซนเซอร์ตรวจพบสิ่งกีดขวางภายในระยะ 50 เซนติเมตร รถขนส่งจะหยุดทำงานพร้อมส่งเสียงแจ้งเตือนว่ามีสิ่งกีดขวาง หากสิ่งกีดขวางอยู่ในระยะที่มากกว่า 50 เซนติเมตร รถขนส่งจะเคลื่อนที่ตามเส้นต่อไปตามระบบควบคุมของมอเตอร์ จนกว่าจะพบสี่แยกและอินฟราเรดเซนเซอร์ทั้ง 5 ตัวอ่านค่าได้ 11111 ซึ่งหมายความว่ารถขนส่งได้พบสี่แยกและทำการนับค่า (count) เพิ่มขึ้นทีละ 1 ค่า หากไม่พบรถขนส่งจะเคลื่อนที่เดินตามเส้นต่อไป เมื่อพบทางแยกที่ 1 รถขนส่งจะเคลื่อนที่ตามเส้นต่อไป และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 2 จะทำการเลี้ยวขวา และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 3 จะทำการหยุดรถขนส่งพร้อมส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ถ้าได้รับสัญญาณตอบรับจากอินฟราเรดเซนเซอร์ด้านบนจะส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที จากนั้นทำการกลับรถขนส่ง แต่หากไม่ได้รับสัญญาณตอบรับรถขนส่งจะหยุดเป็นเวลา 20 วินาที และส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ก่อนทำการกลับรถขนส่ง และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 4 จะทำการเลี้ยวซ้าย และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 5 รถขนส่งจะเคลื่อนที่ตามเส้นต่อไป และนับค่า (count) เพิ่ม 1 ค่า และเมื่อพบทางแยกที่ 6 จะทำการส่งเสียงแจ้งเตือน 1 ครั้ง และทำการกลับรถเพื่อรับค่าใหม่ ซึ่งเป็นอันสิ้นสุดการทำงาน รูปแบบการทำงานและแผนผังการทำงานของระบบควบคุมการเดินทางตามเส้นกรณีห้องที่ 3 แสดงดังรูปที่ 3.16 และ 3.17 ตามลำดับ



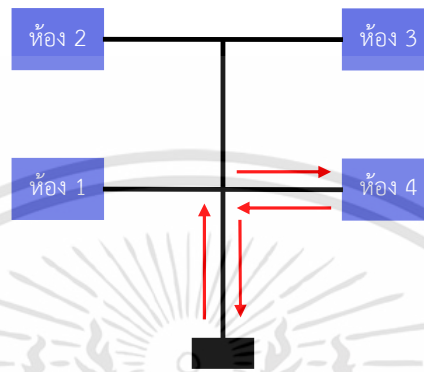
รูปที่ 3.16 รูปแบบการทำงานกรณีห้องหมายเลข 3



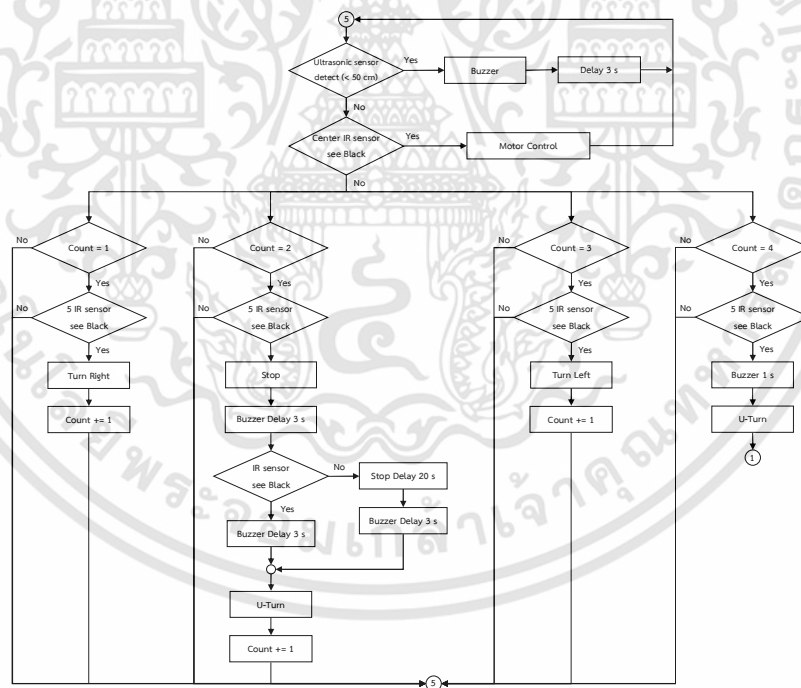
รูปที่ 3.17 แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 3

กรณีกำหนดหมายเลข 4 ที่คีย์แพด เมื่อต้องการเดินทางไปยังห้องหมายเลข 4 เงื่อนไขการเคลื่อนที่ คือ เมื่ออัลตราโซนิกเซนเซอร์ตรวจพบสิ่งกีดขวางภายในระยะ 50 เซนติเมตร รถขนส่งจะหยุดทำงานพร้อมส่งเสียงแจ้งเตือนว่ามีสิ่งกีดขวาง หากสิ่งกีดขวางอยู่ในระยะที่มากกว่า 50 เซนติเมตร รถขนส่งจะเคลื่อนที่ตามเส้นต่อไปตามระบบควบคุมของมอเตอร์ จนกว่าจะพบสี่แยกและอินฟราเรดเซนเซอร์ทั้ง 5 ตัวอ่านค่าได้ 11111 ซึ่งหมายความว่ารถขนส่งได้พบสี่แยกและทำการนับค่า (count) เพิ่มขึ้นทีละ 1 ค่า หากไม่พบรถขนส่งจะเคลื่อนที่เดินตามเส้นต่อไป เมื่อพบทางแยกที่ 1 จะทำการเลี้ยวขวา และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 2 จะทำการหยุดรถขนส่งพร้อมส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ถ้าได้รับสัญญาณตอบรับจากอินฟราเรดเซนเซอร์ด้านบนจะส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที จากนั้นทำการกลับรถขนส่ง แต่หากไม่ได้รับสัญญาณตอบรับรถขนส่งจะหยุดเป็นเวลา 20 วินาที และส่งเสียงแจ้งเตือนเป็นเวลา 3 วินาที ก่อนทำการกลับรถขนส่ง และนับค่า (count) เพิ่ม 1 ค่า เมื่อพบทางแยกที่ 3 จะทำการเลี้ยวซ้าย และนับค่า (count) เพิ่ม 1 ค่า และเมื่อพบทางแยกที่ 4 จะทำการส่งเสียงแจ้งเตือน 1 ครั้ง และทำการกลับรถ

เพื่อรับค่าใหม่ ซึ่งเป็นอันสิ้นสุดการทำงาน รูปแบบการทำงานและแผนผังการทำงานของระบบ ควบคุมการเดินตามเส้นกรณีห้องที่ 4 แสดงดังรูปที่ 3.18 และ 3.19 ตามลำดับ



รูปที่ 3.18 รูปแบบการทำงานกรณีห้องหมายเลข 4

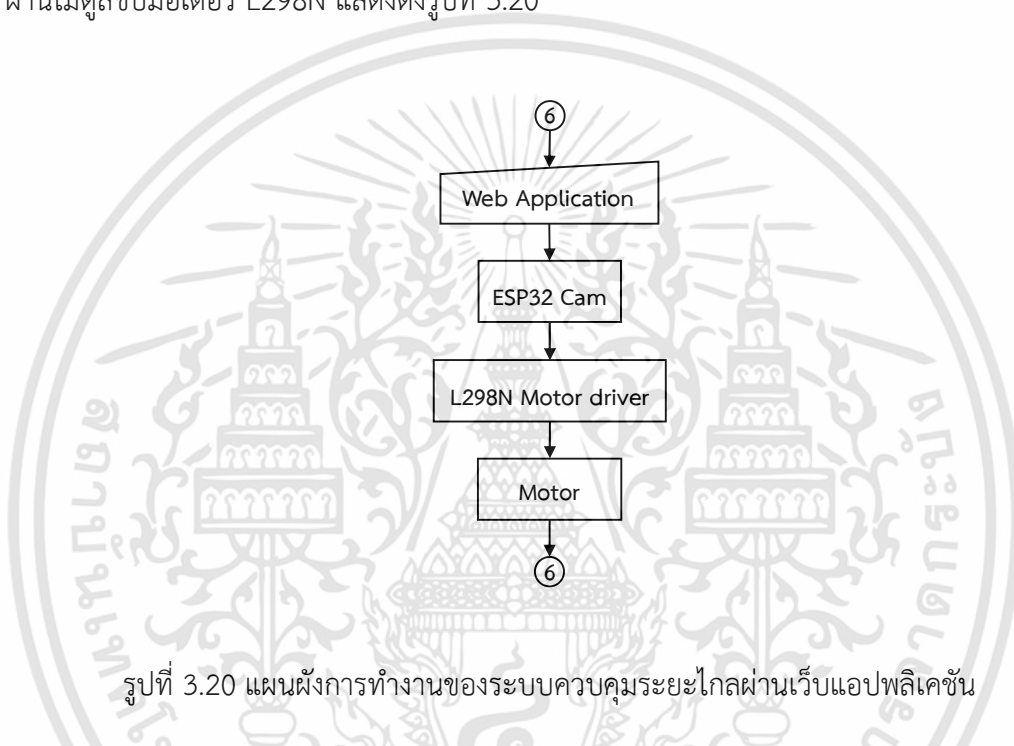


รูปที่ 3.19 แผนผังการทำงานของระบบควบคุมการเดินตามเส้นกรณีห้องที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 ระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

เริ่มต้นจากการเชื่อมต่อ ESP32 Cam และเว็บแอปพลิเคชันผ่านเครือข่าย Wi-Fi จากนั้นเมื่อกดปุ่มสั่งการให้รถขนส่งเดินหน้า ถอยหลัง และเลี้ยวซ้ายหรือเลี้ยวขวา เว็บแอปพลิเคชันจะส่งข้อมูลไปประมวลผลผ่านไมโครคอนโทรลเลอร์ ESP32 Cam ซึ่งจะทำหน้าที่รับคำสั่งจากโทรศัพท์มือถือหรือคอมพิวเตอร์ โดยจะควบคุมการทำงานของมอเตอร์ในการเคลื่อนที่ผ่านโมดูลขับมอเตอร์ L298N แสดงดังรูปที่ 3.20



รูปที่ 3.20 แผนผังการทำงานของระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ไมโครคอนโทรลเลอร์

ในปฏิญานิพนธ์นี้ผู้จัดทำได้เลือกใช้ไมโครคอนโทรลเลอร์ 2 ตัว ดังนี้

3.2.1.1 Arduino Mega 2560

เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีสัญญาณดิจิทัลแบบขาเข้าและขาออก 54 ขา ซึ่งมีพอร์ตจำนวนมาก และเหมาะสำหรับใช้ควบคุมการทำงานของอินฟราเรดเซนเซอร์ และ อัลตราโซนิกเซนเซอร์

3.2.1.2 ESP32 Cam

เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีชิป ESP32 ที่สามารถใช้ในการสื่อสารไร้สาย เป็นทั้งตัวรับสัญญาณและตัวปล่อยสัญญาณได้ในตัว และมีการติดตั้งกล้อง OV2640 เพื่อใช้ประโยชน์จาก CPU ในการรับข้อมูลจึงใช้ในการควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

3.2.2 เซนเซอร์ตรวจจับเส้นขาวดำ

สำหรับปริญญาานิพนธ์นี้ผู้จัดทำได้เลือกใช้เซนเซอร์ตรวจจับสิ่งกีดขวาง เส้นขาวดำ นับจำนวน (TCRT5000 infrared reflectance Obstacle avoidance line tracking sensor) ซึ่งเป็นเซนเซอร์ที่ตรวจจับด้วยแสงอินฟราเรด โดยมีสัญญาณขาออก 2 รูปแบบ คือ รูปแบบสัญญาณดิจิทัล และรูปแบบของสัญญาณแอนะล็อก

3.2.3 เซนเซอร์วัดระยะทาง

สำหรับปริญญาานิพนธ์นี้ผู้จัดทำได้เลือกใช้เซนเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิก HC-SR04 เป็นโมดูลวัดระยะทางที่ใช้หลักการสะท้อนของคลื่นอัลตราโซนิก ในการตรวจจับสิ่งกีดขวางในระยะ 50 เซนติเมตร

3.2.4 โมดูลเสียง

ในปริญญาานิพนธ์นี้ผู้จัดทำได้เลือกใช้โมดูลเสียง Active Buzzer Module ในการส่งเสียงแจ้งเตือนเมื่อถึงเป้าหมายเป็นเวลา 3 วินาที

3.2.5 โมดูลขับมอเตอร์

ในปริญญาานิพนธ์นี้ผู้จัดทำได้เลือกใช้โมดูลขับมอเตอร์ L298N สำหรับควบคุมความเร็วของมอเตอร์ และทิศทางการเคลื่อนที่ของรถขนส่งเวชภัณฑ์

3.2.6 มอเตอร์ขับเคลื่อน

ในปริญญาานิพนธ์นี้ผู้จัดทำได้เลือกใช้มอเตอร์ไฟฟ้ากระแสตรงแรงดัน 12 โวลต์ 20 รอบต่อนาที เนื่องจากสามารถขนส่งเวชภัณฑ์ได้จำนวนมากจึงต้องการแรงบิดสูงเพื่อให้สามารถขับเคลื่อนวัตถุที่มีน้ำหนักมากได้

3.2.7 คีย์แพด

ในปฏิญานิพนธ์นี้ผู้จัดทำได้เลือกใช้คีย์แพดปุ่มกด (4×4 Matrix Keypad Module) ใช้ในการป้อนเงื่อนไขการทำงานของรถขนส่งเวชภัณฑ์ ให้สามารถเคลื่อนที่ตามเป้าหมายได้

3.2.8 จอแสดงผล

ในปฏิญานิพนธ์นี้ผู้จัดทำได้เลือกใช้จอแสดงผล LCD พร้อมโมดูล I2C ขนาด 16×2 ในการแสดงหมายเลขห้องหรือฟังก์ชันการควบคุมผ่านเว็บแอปพลิเคชัน

3.2.9 แบตเตอรี่

ในปฏิญานิพนธ์นี้ผู้จัดทำได้เลือกใช้แบตเตอรี่ที่มีแรงดันไฟฟ้าขนาด 12 โวลต์ 7.2 แอมแปร์ชั่วโมงดังแสดงในรูปที่ 3.20 ใช้สำหรับขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง และจ่ายไฟให้บอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.21 แบตเตอรี่ที่มีแรงดันไฟฟ้าขนาด 12 โวลต์ 7.2 แอมแปร์ชั่วโมง

3.2.10 ล้อขับเคลื่อน

ในปฏิญานิพนธ์นี้ผู้จัดทำได้เลือกใช้ล้อขนาดเส้นผ่านศูนย์กลาง 7 นิ้ว เป็นล้อสำหรับการขับเคลื่อนตัวรถส่งเวชภัณฑ์ แสดงดังรูปที่ 3.22



รูปที่ 3.22 ล้อขับเคลื่อนรถส่งเวชภัณฑ์

3.2.11 สวิตช์

ในปฏิญานพนธ์นี้ผู้จัดทำได้เลือกใช้สวิตช์สองขาสำหรับแรงดันไฟฟ้ากระแสตรง 12 โวลต์ แสดงดังรูปที่ 3.23 เป็นการเปิดปิดการจ่ายไฟฟ้ากระแสตรงให้บอร์ดไมโครคอนโทรลเลอร์ และสวิตช์ปุ่มกดแบบกดติดปล่อยดับดังแสดงในรูปที่ 3.24 เป็นการรีเซตคำสั่ง เพื่อรับค่าใหม่ผ่าน คีย์แพด



รูปที่ 3.23 สวิตช์สองขา



รูปที่ 3.24 สวิตช์ปุ่มกดแบบกดติดปล่อยดับ [35]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบการทำงานของอินฟราเรดเซนเซอร์

ทำการตรวจจับเส้นสีดำด้วยโมดูลอินฟราเรดเซนเซอร์รุ่น TCRT5000 โดยการเขียนชุดคำสั่งและระบุเงื่อนไขการทำงานในโปรแกรม Arduino IDE และปรับเทียบระยะเวลาการตรวจจับสีดำผ่านตัวต้านทานปรับค่าได้

3.3.2 การทดสอบการวัดระยะของอัลตราโซนิกเซนเซอร์

ทำการวัดระยะการตรวจจับวัตถุด้วยอัลตราโซนิกเซนเซอร์รุ่น HC-SR04 โดยการเขียนชุดคำสั่งในโปรแกรม Arduino IDE ส่งการผ่านอุปกรณ์ไมโครคอนโทรลเลอร์ Arduino Uno R3 ในการตรวจวัดระยะของวัตถุ จากนั้นบันทึกค่าที่วัดได้และนำไปปรับเทียบเพื่อลดความคลาดเคลื่อน

3.3.3 การทดสอบประสิทธิภาพของแบตเตอรี่

ทำการทดสอบโดยการกำหนดเงื่อนไขในชุดคำสั่งให้รถขนส่งเคลื่อนที่ตามเส้นแล้วเมื่อเจอทางแยกจะเลี้ยวซ้ายหมุนกลับตลอดจากนั้นบันทึกค่าความต่างศักย์ของแบตเตอรี่ทุก 30 นาทีจนกระทั่งมีระดับแรงดันไฟฟ้าต่ำกว่า 11.00 โวลต์

3.3.4 การทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้

ทำการทดสอบโดยเพิ่มน้ำหนักที่บรรทุกตั้งแต่ 5 กิโลกรัม ถึง 20 กิโลกรัม แล้วสังเกตสถานะการทำงานของตัวรถและบันทึกความเร็วในการเคลื่อนที่

3.3.5 ทดสอบการแสดงวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์

ทำการทดสอบการแสดงวิดีโอภาพเคลื่อนไหวผ่านโมดูล ESP32 Cam โดยการป้อนชุดคำสั่งผ่านโปรแกรม Arduino IDE แล้วทำการแก้ไขชื่อ Wi-Fi และรหัสผ่านที่ใช้เชื่อมต่อกับโมดูลแล้วนำหมายเลข IP ของ ESP32 Cam ไปใช้งาน

3.3.6 การทดสอบระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

ทดสอบการควบคุมการเคลื่อนที่ของรถขนส่งผ่านเว็บแอปพลิเคชัน ซึ่งจะมีหลักการทำงานคล้ายกับรีโมทควบคุมไร้สายผ่าน Wi-Fi โดยควบคุมการทำงานของมอเตอร์ผ่านบอร์ดประมวลผล ESP32 Cam ซึ่งจะทำหน้าที่รับคำสั่งจากโทรศัพท์มือถือหรือคอมพิวเตอร์

3.3.7 การทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงจุดหมาย

ทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงจุดหมาย โดยเชื่อมต่ออินฟราเรดเซนเซอร์กับสัญญาณดิจิทัลขาเข้าของบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 เพื่อนำข้อมูลไปประมวลผลตามเงื่อนไขการใช้งานในกรณีที่รถขนส่งเดินทางไปถึงจุดหมายของแต่ละห้องให้มีการรับค่าสัญญาณลอจิกเพื่อยืนยันว่าถึงจุดหมายแล้ว โดยการใช้ฝ่ามือกดบังอินฟราเรดเซนเซอร์ด้านบนเพื่อยืนยันว่ารถขนส่งเดินทางไปถึงจุดหมาย

3.3.8 การทดสอบการมอดูเลตความกว้างพัลส์ของโมดูลขับเคลื่อนมอเตอร์ L298N

ทดสอบการมอดูเลตความกว้างพัลส์โดยเชื่อมต่อมอเตอร์ด้านซ้ายและด้านขวา กับโมดูลขับเคลื่อนมอเตอร์ L298N เพื่อควบคุมทิศทางตามเงื่อนไขที่กำหนดในชุดคำสั่งให้รถขนส่งลดอัตราเร็วของมอเตอร์ด้านซ้ายหรือด้านขวาในกรณีที่รถขนส่งเคลื่อนที่ออกจากเส้น โดยลดอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลา

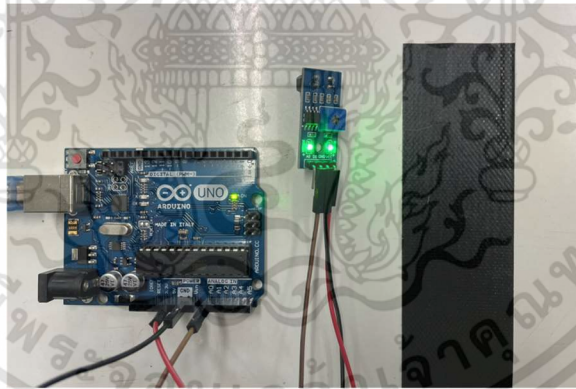
บทที่ 4

ผลการทดลอง

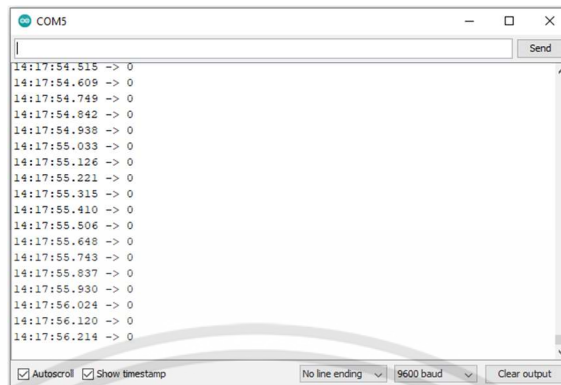
สำหรับการทำงานของรถส่งเวชภัณฑ์ ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบ โดยแบ่งการทดสอบและจัดเก็บผลการทดสอบ ดังต่อไปนี้

4.1 ผลการทดสอบการทำงานของอินฟราเรดเซนเซอร์

ทำการทดสอบการทำงานของอินฟราเรดเซนเซอร์ โดยการเขียนชุดคำสั่งและระบุเงื่อนไขการทำงานในโปรแกรม Arduino IDE โดยอินฟราเรดเซนเซอร์จะมีตัวรับสัญญาณและตัวส่งสัญญาณภายในตัว เมื่ออินฟราเรดเซนเซอร์ตรวจพบพื้นสีขาวตัวรับสัญญาณสามารถรับลำแสงที่สะท้อนกลับมาได้แสดงดังรูปที่ 4.1 จึงแสดงค่าสัญญาณลอจิกเป็น 0 แสดงดังรูปที่ 4.2 และเมื่ออินฟราเรดเซนเซอร์ตรวจพบพื้นสีดำตัวรับสัญญาณไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้แสดงดังรูปที่ 4.3 จึงแสดงค่าสัญญาณลอจิกเป็น 1 แสดงดังรูปที่ 4.4



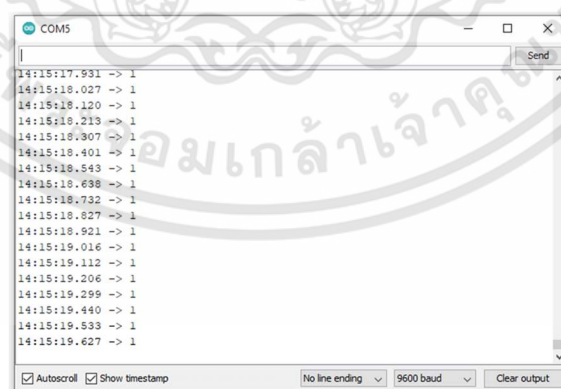
รูปที่ 4.1 อินฟราเรดเซนเซอร์ตรวจพบพื้นสีขาว



รูปที่ 4.2 Serial Monitor เมื่อค่าลอจิกเป็น 0



รูปที่ 4.3 อินฟราเรดเซนเซอร์ตรวจพบพื้นสีดำ

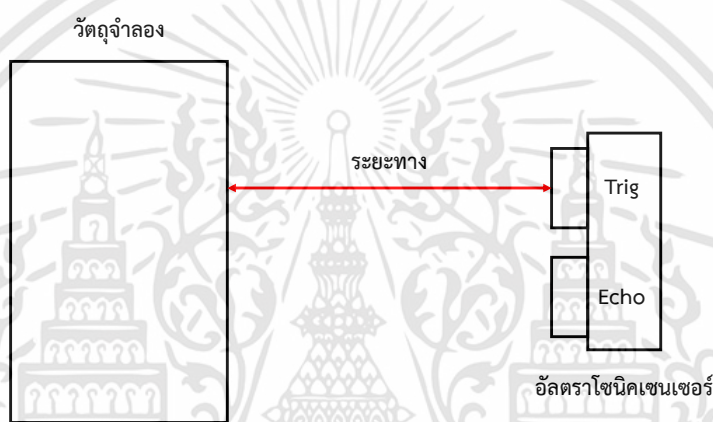


รูปที่ 4.4 Serial Monitor เมื่อค่าลอจิกเป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

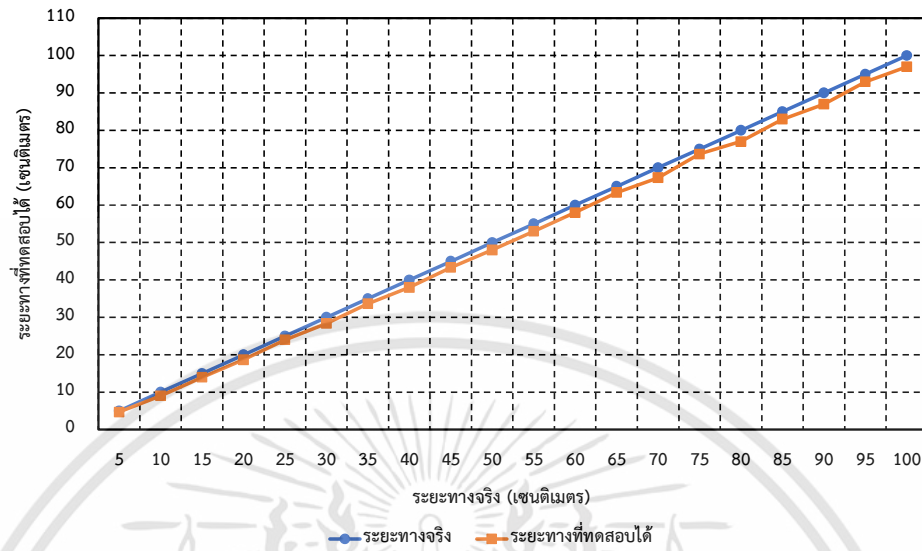
4.2 การทดสอบการวัดระยะของอัลตราโซนิกเซนเซอร์

การปรับเทียบเซนเซอร์โดยใช้วัตถุจำลองทรงสี่เหลี่ยมผืนผ้าที่มีขนาดเส้นผ่านศูนย์กลาง 9.5 เซนติเมตร ทำการบันทึกค่าระยะทางที่วัดได้ในหน่วยเซนติเมตร โดยทำการทดสอบ 3 ครั้ง และคำนวณค่าเฉลี่ยของระยะทางที่วัดได้พร้อมวิเคราะห์และคำนวณค่าเปอร์เซ็นต์ความคลาดเคลื่อน จากนั้นทำการปรับเทียบและคำนวณเปอร์เซ็นต์ของความแตกต่างค่าความคลาดเคลื่อนก่อนและหลังปรับเทียบ ภาพจำลองการทดสอบแสดงดังรูปที่ 4.5



รูปที่ 4.5 การจำลองการทดสอบวัดระยะด้วยอัลตราโซนิกเซนเซอร์โดยใช้วัตถุจำลองทรงสี่เหลี่ยมผืนผ้า

ทำการบันทึกข้อมูลระยะทางที่ทดสอบได้จากอัลตราโซนิกเซนเซอร์และพิจารณาความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้ในช่วง 5 เซนติเมตร ถึง 100 เซนติเมตร ก่อนการปรับเทียบเซนเซอร์ ดังแสดงในรูปที่ 4.6



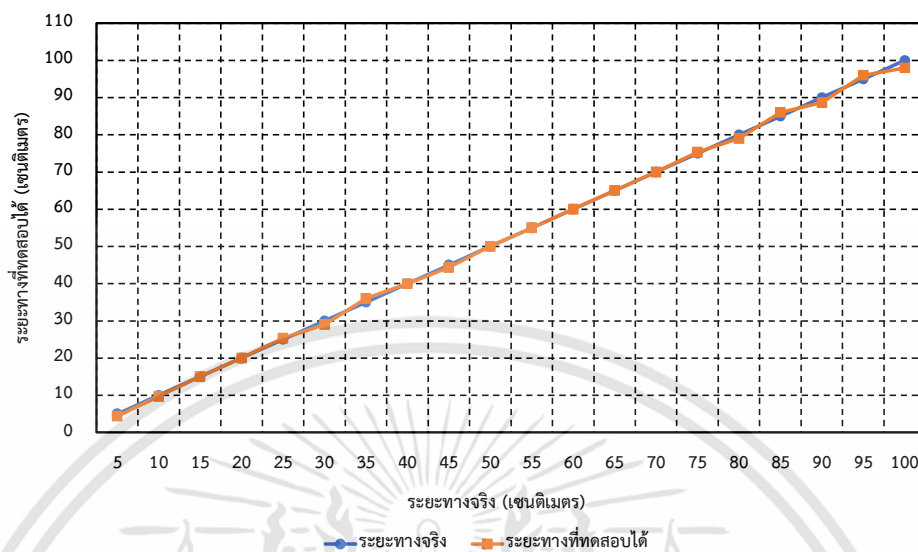
รูปที่ 4.6 ความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้ก่อนการปรับเทียบเซนเซอร์

ทำการเก็บข้อมูลการปรับเทียบระยะทางที่ทดสอบได้จากอัลตราโซนิกเซนเซอร์โดยใช้สมการที่ (4.1) และพิจารณาความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้ในช่วง 5 เซนติเมตร ถึง 100 เซนติเมตร หลังการปรับเทียบเซนเซอร์ ดังแสดงในรูปที่ 4.7

$$D = t \times 0.0173681 \quad (4.1)$$

เมื่อ D คือ ระยะทาง มีหน่วยเป็น เซนติเมตร

t คือ ระยะเวลาที่สัญญาณเดินทาง มีหน่วยเป็น ไมโครวินาที



รูปที่ 4.7 ความสัมพันธ์ระหว่างระยะทางจริงและระยะทางที่ทดสอบได้หลังการปรับเทียบเซนเซอร์

จากนั้นทำการเปรียบเทียบความคลาดเคลื่อนก่อนการปรับเทียบและหลังปรับเทียบเซนเซอร์ที่วัดได้พบว่าหลังจากการปรับเทียบมีความคลาดเคลื่อนโดยเฉลี่ยลดลง โดยแสดงผลลัพธ์ดังตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบความคลาดเคลื่อนก่อนและหลังปรับเทียบเซนเซอร์

ระยะทางจริง (เซนติเมตร)	เปอร์เซ็นต์ ความคลาดเคลื่อน ก่อนการปรับเทียบ (เปอร์เซ็นต์)	เปอร์เซ็นต์ ความคลาดเคลื่อน หลังการปรับเทียบ (เปอร์เซ็นต์)	ความแตกต่างของ เปอร์เซ็นต์ความ คลาดเคลื่อนก่อนและหลัง ปรับเทียบ (เปอร์เซ็นต์)
5	6.67	13.33	6.67
10	10.00	3.33	6.67
15	6.67	0.00	6.67
20	6.67	0.00	6.67
25	4.00	1.33	2.67
30	5.56	3.33	2.23

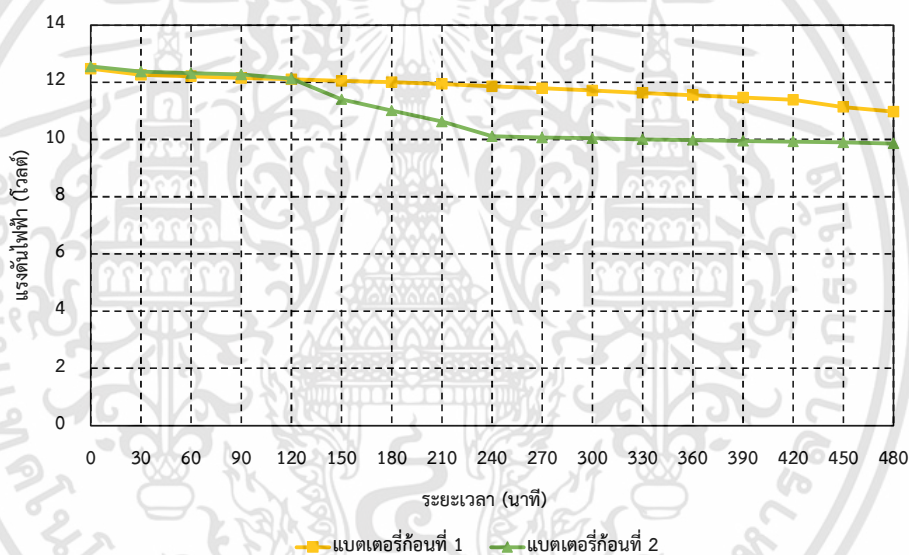
ตารางที่ 4.1 เปรียบเทียบความคลาดเคลื่อนก่อนและหลังปรับเทียบเซนเซอร์ (ต่อ)

ระยะทางจริง (เซนติเมตร)	เปอร์เซ็นต์ ความคลาดเคลื่อน ก่อนการปรับเทียบ (เปอร์เซ็นต์)	เปอร์เซ็นต์ ความคลาดเคลื่อน หลังการปรับเทียบ (เปอร์เซ็นต์)	ความแตกต่างของ เปอร์เซ็นต์ความ คลาดเคลื่อนก่อนและหลัง ปรับเทียบ (เปอร์เซ็นต์)
35	3.81	2.86	0.95
40	5.00	0.00	5.00
45	3.70	1.48	2.22
50	4.00	0.00	4.00
55	3.64	0.00	3.64
60	3.33	0.00	3.33
65	2.56	0.00	2.56
70	3.81	0.00	3.81
75	1.78	0.44	1.33
80	3.75	1.25	2.50
85	2.35	1.18	1.18
90	3.33	1.48	1.85
95	2.11	1.05	1.05
100	3.00	2.00	1.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบประสิทธิภาพของแบตเตอรี่

ทำการทดสอบประสิทธิภาพของแบตเตอรี่จำนวน 2 ก้อน ซึ่งแบตเตอรี่ก้อนที่ 1 มีค่าแรงดันไฟฟ้าเริ่มต้นเท่ากับ 12.47 โวลต์ และแบตเตอรี่ก้อนที่ 2 มีค่าแรงดันไฟฟ้าเริ่มต้นเท่ากับ 12.55 โวลต์ โดยทำการเชื่อมต่อก้อนที่ 1 เข้ากับโมดูลขับเคลื่อนมอเตอร์ที่ใช้สำหรับขับเคลื่อนรถส่งเวชภัณฑ์ และเชื่อมต่อก้อนที่ 2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ที่ใช้สำหรับประมวลผลข้อมูล และทำการบันทึกค่าแรงดันไฟฟ้าของแบตเตอรี่ทุก 30 นาที จนกระทั่งแรงดันไฟฟ้ามีค่าน้อยกว่า 11.00 โวลต์ แนวโน้มค่าแรงดันไฟฟ้าขณะใช้งานของแบตเตอรี่ก้อนที่ 1 เปรียบเทียบกับแบตเตอรี่ก้อนที่ 2 ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 ค่าแรงดันไฟฟ้าขณะใช้งานของแบตเตอรี่ก้อนที่ 1 และแบตเตอรี่ก้อนที่ 2

จากการทดสอบพบว่าแบตเตอรี่ก้อนที่ 1 มีระยะเวลาในการใช้งานประมาณ 480 นาที หรือ 8 ชั่วโมง โดยมีค่าแรงดันไฟฟ้าอยู่ที่ 10.97 โวลต์ และแบตเตอรี่ก้อนที่ 2 มีระยะเวลาในการใช้งานประมาณ 180 นาที หรือ 3 ชั่วโมง มีค่าแรงดันอยู่ที่ 11.01 โวลต์

4.4 การทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้

ทำการทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้ โดยกำหนดให้รถขนส่งเคลื่อนที่ไปกลับ 1 รอบ เป็นระยะทาง 4 เมตร โดยกำหนดเงื่อนไขเมื่อเจอทางแยกจะเลี้ยวซ้ายเสมอ และทำการเพิ่มน้ำหนักครั้งละ 2.5 กิโลกรัม จนถึง 20 กิโลกรัม แสดงผลการทดสอบดังตารางที่ 4.2

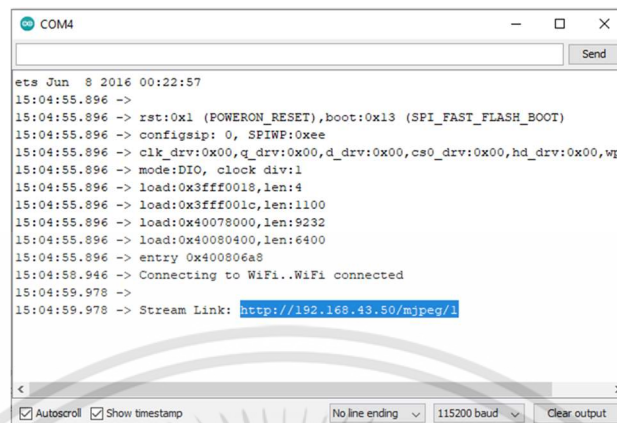
ตารางที่ 4.2 การทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้

น้ำหนัก (กิโลกรัม)	อัตราเร็ว (เซนติเมตรต่อวินาที)
0	9.85
2.5	9.16
5.0	7.87
7.5	7.44
10.0	6.45
12.5	5.38
15.0	5.04
17.5	4.80
20.0	4.58

จากการทดสอบพบว่าเมื่อเพิ่มน้ำหนักในการบรรทุกจะเกิดแรงเฉื่อยในการเคลื่อนที่ทำให้เวลาที่ใช้ในการเคลื่อนที่เพิ่มมากขึ้น โดยมีอัตราเร็วเฉลี่ยเท่ากับ 6.26 เซนติเมตรต่อวินาที และเมื่อรถขนส่งรับน้ำหนักมากกว่า 15 กิโลกรัม จะส่งผลต่อการเลี้ยวเนื่องจากการขับเคลื่อนด้วยล้อหลัง

4.5 ทดสอบการแสดงวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์

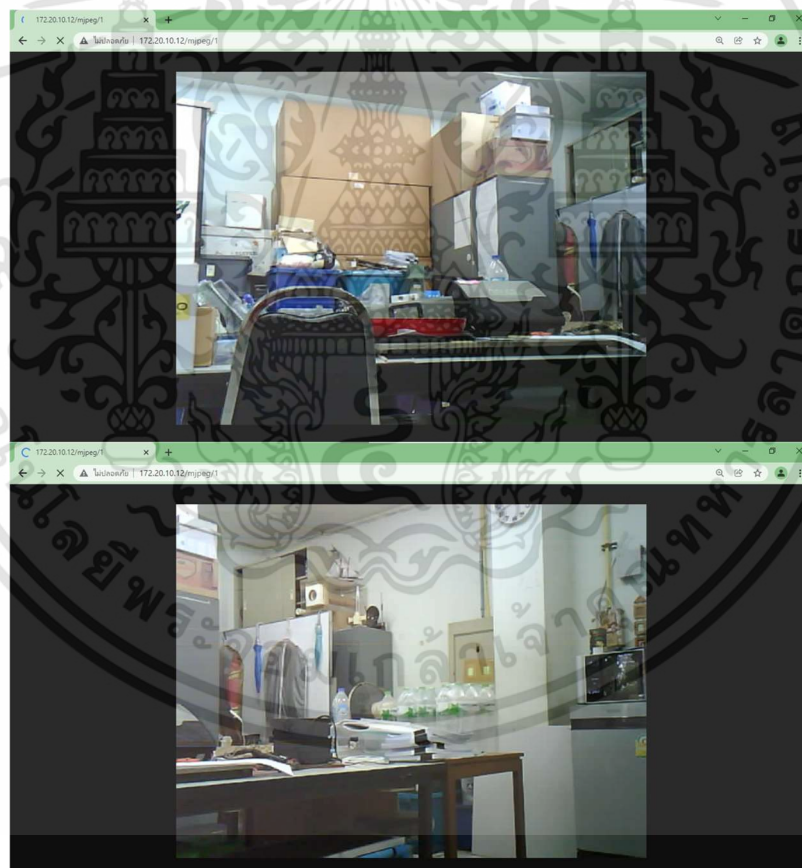
ทำการทดสอบการแสดงวิดีโอภาพเคลื่อนไหวผ่านโมดูล ESP32 Cam โดยการป้อนชุดคำสั่งผ่านโปรแกรม Arduino IDE และทำการแก้ไขชื่อ Wi-Fi และรหัสผ่านที่ใช้สำหรับเชื่อมต่อกับโมดูล จากนั้นนำหมายเลข IP ของ ESP32 Cam ดังแสดงในรูปที่ 4.10 ป้อนลง URL บนเว็บเบราว์เซอร์ ซึ่งเว็บเบราว์เซอร์จะแสดงวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์ดังแสดงในรูปที่ 4.11



```

COM4
ets Jun  8 2016 00:22:57
15:04:55.896 ->
15:04:55.896 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
15:04:55.896 -> configisp: 0, SPIWP:0xee
15:04:55.896 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp
15:04:55.896 -> mode:DIO, clock div:1
15:04:55.896 -> load:0x3fff0018,len:4
15:04:55.896 -> load:0x3fff001c,len:1100
15:04:55.896 -> load:0x40078000,len:9232
15:04:55.896 -> load:0x40080400,len:6400
15:04:55.896 -> entry 0x400806a8
15:04:58.946 -> Connecting to WiFi..WiFi connected
15:04:59.978 ->
15:04:59.978 -> Stream Link: http://192.168.43.50/mjpeg/1
  
```

รูปที่ 4.10 หน้าต่าง Serial Monitor แสดง IP ของ ESP32 Cam

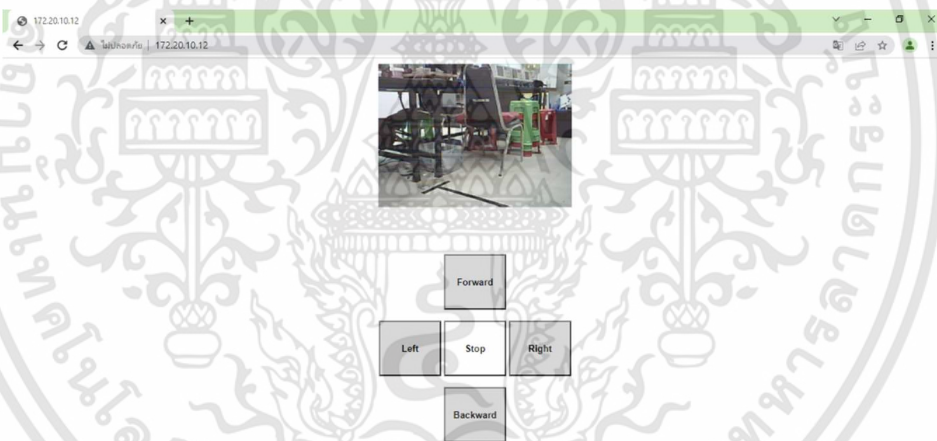


รูปที่ 4.11 วิดีโอภาพเคลื่อนไหวแบบเรียลไทม์บนเว็บเบราว์เซอร์

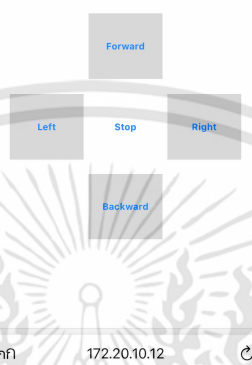
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดสอบระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชัน

ทำการทดสอบการควบคุมการเคลื่อนที่ของรถขนส่งผ่านเว็บแอปพลิเคชัน จะมีหลักการทำงานคล้ายกับรีโมทควบคุมไร้สายผ่าน Wi-Fi โดยเมื่อกดปุ่มคำสั่งผ่านเว็บแอปพลิเคชัน ข้อมูลจะถูกส่งไปประมวลผลผ่านโมดูลไมโครคอนโทรลเลอร์ ESP32 Cam จากนั้นส่งคำสั่งผ่านบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 เพื่อควบคุมการทำงานของมอเตอร์ผ่านโมดูลขับมอเตอร์ L298N เพื่อให้รถขนส่งเคลื่อนที่ตามเงื่อนไขคำสั่งทั้งการเดินหน้า การถอยหลัง การเลี้ยวซ้ายและการเลี้ยวขวา ผ่านวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์ โดยจากการทดสอบในพื้นที่โล่งพบว่า ที่ระยะห่างระหว่างรถขนส่งกับผู้ควบคุมตั้งแต่ 15 เมตร ถึง 30 เมตร ภาพจะหน่วงเป็นเวลา 1 วินาที และที่ระยะห่างมากกว่า 30 เมตร ภาพจะหน่วงเป็นเวลา 3 วินาที ซึ่งระยะห่างที่ไกลที่สุดที่สามารถควบคุมการเคลื่อนที่ของรถขนส่งได้ไม่เกินระยะ 50 เมตร หน้าเว็บแอปพลิเคชันผ่านคอมพิวเตอร์และโทรศัพท์มือถือแสดงดังรูปที่ 4.12 และ 4.13



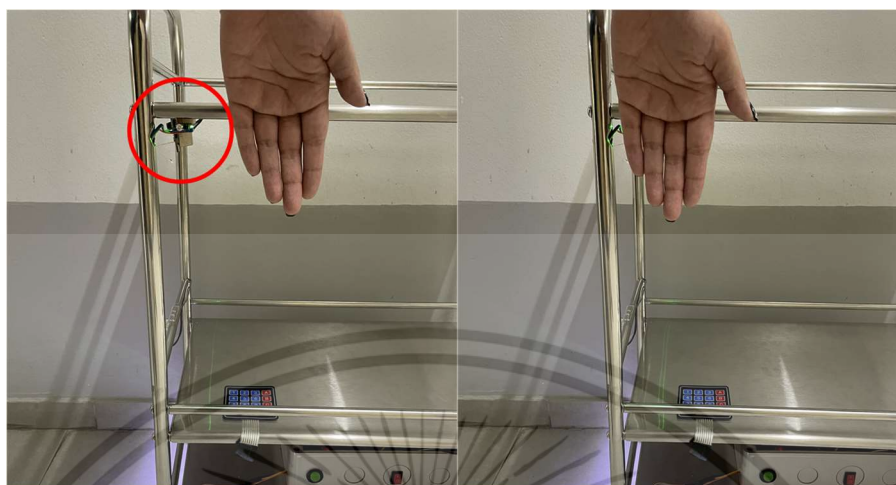
รูปที่ 4.12 หน้าเว็บแอปพลิเคชันบนคอมพิวเตอร์



รูปที่ 4.13 หน้าเว็บแอปพลิเคชันบนโทรศัพท์มือถือ

4.7 การทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงจุดหมาย

ทำการทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงเป้าหมาย เพื่อนำข้อมูลไปประมวลผลตามเงื่อนไขการใช้งานในกรณีที่รถขนส่งเดินทางไปถึงจุดหมายของแต่ละห้องให้มีการรับค่าสัญญาณลอจิกโดยใช้หลักการทำงานของอินฟราเรดเซนเซอร์เพื่อยืนยันว่าถึงจุดหมายแล้ว โดยการวางฝ่ามือหรือหลังมือบดบังการทำงานของอินฟราเรดเซนเซอร์ที่แสดงในวงกลมสีแดงดังรูปที่ 4.14 เพื่อยืนยันว่าถึงจุดหมายก่อนรถขนส่งจะเดินทางไปยังห้องถัดไป



(ก) ฝ่ามือยังไม่กดบัง

(ข) ฝ่ามือกดบัง

รูปที่ 4.14 ตำแหน่งของฝ่ามือกับการทำงานของอินฟราเรดเซนเซอร์

4.8 การทดสอบการมอดูเลตความกว้างพัลส์ของโมดูลขับเคลื่อนมอเตอร์ L298N

ทำการทดสอบโดยการกำหนดเงื่อนไขในชุดคำสั่งให้รถขนส่งลดอัตราเร็วของมอเตอร์ด้านซ้ายหรือด้านขวาในกรณีที่รถขนส่งเคลื่อนที่ออกจากเส้น โดยลดอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลาแสดงดังตารางที่ 4.3

ตารางที่ 4.3 เปรียบเทียบอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลาของมอเตอร์

สถานะของอินฟราเรดเซนเซอร์	อัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลา (Duty Cycle)		อัตราเร็วมอเตอร์ (เซนติเมตรต่อวินาที)	
	มอเตอร์ซ้าย	มอเตอร์ขวา	มอเตอร์ซ้าย	มอเตอร์ขวา
00100	255	255	18.62	18.62
01100	195	255	14.24	18.62
11000	75	255	5.48	18.62
10000	15	255	1.10	18.62
00110	255	195	18.62	14.24
00011	255	75	18.62	5.48
00001	255	15	18.62	1.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบโดยการลดอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลา (Duty Cycle) เพื่อให้รถขนส่งเคลื่อนที่ตามเส้นได้ตรงตามเงื่อนไขที่กำหนดและไม่หลุดออกจากเส้น โดยขณะที่อินฟราเรดเซนเซอร์ด้านหน้ามีสถานะลอจิกเป็น 00100 จะมีอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลา (Duty Cycle) ทั้งมอเตอร์ด้านซ้ายและขวาที่ 255 บิตและความเร็วของมอเตอร์ทั้งสองมีค่าเท่ากับ 18.62 เซนติเมตรต่อวินาที และเมื่อรถเบนหลุดออกจากเส้นจะทำการลดความเร็วโดยการลดอัตราส่วนระหว่างความกว้างพัลส์กับคาบเวลา (Duty Cycle) ของมอเตอร์ด้านซ้ายและขวา



บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีเป้าหมายเพื่อช่วยลดการพบปะระหว่างแพทย์กับผู้ติดเชื้อไวรัสโคโรนา-19 และช่วยลดการใช้ชุดป้องกันเชื้อไวรัส ที่สามารถแพร่เชื้อผ่านระบบทางเดินหายใจได้ จึงได้ทำการศึกษาและจัดทำรถขนส่งเวชภัณฑ์อัตโนมัติขึ้นมา โดยระบบรถส่งเวชภัณฑ์ประกอบด้วย 2 ส่วน คือ ส่วนการพัฒนารถขนส่งเวชภัณฑ์ที่สามารถเดินตามเส้นโดยใช้อินฟราเรดเซนเซอร์และอัลตราโซนิกเซนเซอร์ในการประมวลผลและแสดงผลการควบคุมผ่านหน้าจอ LCD ขนาดเล็ก และระบบควบคุมระยะไกลผ่านเว็บแอปพลิเคชันโดยการควบคุมผ่าน Wi-Fi แล้วนำข้อมูลไปประมวลผลเพื่อสั่งการให้มอเตอร์ทำงาน ซึ่งระบบทั้งหมดจะถูกควบคุมผ่านไมโครคอนโทรลเลอร์

ผู้จัดทำได้ออกแบบและจัดทำรถขนส่งเวชภัณฑ์อัตโนมัติ ทดสอบการทำงานของอินฟราเรดเซนเซอร์ เมื่ออินฟราเรดเซนเซอร์ตรวจพบพื้นสีขาวตัวรับสัญญาณสามารถรับลำแสงที่สะท้อนกลับมาได้แสดงดังรูปที่ 4.1 จึงแสดงค่าสัญญาณลอจิกเป็น 0 แสดงดังรูปที่ 4.2 และเมื่ออินฟราเรดเซนเซอร์ตรวจพบพื้นสีดำตัวรับสัญญาณไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้จึงแสดงค่าสัญญาณลอจิกเป็น 1 และทำการปรับเทียบเซนเซอร์โดยใช้วัตถุจำลองทรงสี่เหลี่ยมผืนผ้าที่มีขนาดเส้นผ่านศูนย์กลาง 9.5 เซนติเมตร และทำการเก็บค่าระยะทางที่วัดได้ในหน่วยเซนติเมตร โดยทำการทดสอบ 3 ครั้ง จะได้ค่าความแตกต่างของความคลาดเคลื่อนก่อนและหลังปรับเทียบเฉลี่ยเท่ากับ 3.33 เปอร์เซ็นต์ ซึ่งเป็นค่าที่ยอมรับได้ ทำการทดสอบประสิทธิภาพของแบตเตอรี่จำนวน 2 ก้อน ซึ่งแบตเตอรี่ก้อนที่ 1 มีค่าแรงดันไฟฟ้าเริ่มต้นเท่ากับ 12.47 โวลต์ และแบตเตอรี่ก้อนที่ 2 มีค่าแรงดันไฟฟ้าเริ่มต้นเท่ากับ 12.55 โวลต์ ทำการบันทึกค่าแรงดันไฟฟ้าของแบตเตอรี่ทุก 30 นาที จนกระทั่งแรงดันไฟฟ้าของแบตเตอรี่มีค่าน้อยกว่า 11.00 โวลต์ การทดสอบพบว่าแบตเตอรี่ก้อนที่ 1 มีค่าแรงดันไฟฟ้าเท่ากับ 10.97 โวลต์ มีระยะเวลาการใช้งานประมาณ 480 นาที หรือ 8 ชั่วโมง และแบตเตอรี่ก้อนที่ 2 มีค่าแรงดันไฟฟ้าเท่ากับ 11.01 โวลต์ มีระยะเวลาการใช้งานประมาณ 180 นาที หรือ 3 ชั่วโมง โดยสามารถทราบค่าแรงดันแบตเตอรี่ผ่านมิเตอร์วัดแรงดันแบตเตอรี่ได้ ทำการทดสอบน้ำหนักที่รถขนส่งสามารถบรรทุกได้โดยทำการเพิ่มน้ำหนักครั้งละ 2.5 กิโลกรัม จนถึง 20 กิโลกรัม จากการทดสอบพบว่าเมื่อเพิ่มน้ำหนักในการบรรทุกจะเกิดแรง

เมื่อทำให้เวลาที่ใช้ในการเคลื่อนที่เพิ่มมากขึ้นและเมื่อรถขนส่งรับน้ำหนักมากกว่า 15 กิโลกรัม จะส่งผลต่อการเลี้ยวเนื่องจากเป็นการขับเคลื่อนด้วยล้อหลังโดยความเร็วในการเคลื่อนที่ปกติขณะไม่ได้บรรทุกสิ่งของจะมีความเร็วเฉลี่ย 0.5537 กิโลเมตรต่อชั่วโมง ทำการทดสอบการแสดงผลวิดีโอภาพเคลื่อนไหวผ่านโมดูล ESP32 Cam โดยนำหมายเลข IP ของ ESP32 Cam ป้อนลง URL บนเว็บเบราว์เซอร์ ซึ่งเว็บเบราว์เซอร์จะแสดงผลวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์ ทำทดสอบการควบคุมการเคลื่อนที่ของรถขนส่งผ่านเว็บแอปพลิเคชัน จะพบว่าสัญญาณ Wi-Fi มีผลต่อความเสถียรในการควบคุมรวมถึงระยะห่างระหว่างผู้ควบคุมกับตัวรถในการควบคุม และจากการทดสอบฟังก์ชันเพื่อยืนยันเมื่อถึงจุดหมายพบว่าสามารถทำงานได้อย่างถูกต้องและแม่นยำ

5.2 ข้อเสนอแนะ

การรับข้อมูลจากอินฟราเรดเซนเซอร์มาประมวลผลจะต้องอาศัยความแม่นยำสูง เพราะจะส่งผลโดยตรงในการเคลื่อนที่ของตัวรถขนส่ง และการตรวจจับการชนของอัลตราโซนิกเซนเซอร์สามารถตรวจจับวัตถุได้ในระยะจำกัด ซึ่งอาจแก้ไขได้โดยเลือกใช้เซนเซอร์วัดระยะที่มีความแม่นยำมากขึ้น เนื่องจากในการเคลื่อนที่อาจมีสิ่งกีดขวางด้านหน้าที่อัลตราโซนิกเซนเซอร์ไม่สามารถตรวจพบและหยุดรถได้ทันซึ่งจะทำให้รถขนส่งเสียหายได้ ส่วนของการแสดงผลวิดีโอภาพเคลื่อนไหวแบบเรียลไทม์ที่ระยะห่างระหว่างผู้ควบคุมกับรถขนส่งมากกว่า 30 เมตร ซึ่งภาพมีความหน่วงเพื่อลดข้อผิดพลาดดังกล่าวควรพัฒนาเว็บแอปพลิเคชันให้มีประสิทธิภาพมากขึ้น และการใช้งานแบตเตอรี่ เนื่องจากไม่สามารถชาร์จภายในตัวรถขนส่งได้ จึงต้องนำแบตเตอรี่ออกมาภายนอกเพื่อทำการชาร์จก่อนนำกลับไปใช้ จึงแนะนำให้ติดตั้งตัวชาร์จไว้ที่แบตเตอรี่ที่อยู่ภายในรถขนส่ง เพื่อให้ง่ายต่อการชาร์จและการนำไปใช้งานในครั้งต่อไป

บรรณานุกรม

- [1] gravitechthai. ARDUINO MEGA 2560. (2561). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <https://www.gravitechthai.com/product-detail.php?WP=qmlZAJ1CM5O0hJatrTZo7o3Q>.
- [2] LungMaker. การใช้งานบอร์ด 2560 เบื้องต้น. (2563). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <http://www.lungmaker.com/arduino-mega-2560-การใช้งาน/>.
- [3] arduino-indy. Arduino MEGA 2560. (2561). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <http://www.arduino-indy.com/product/2/arduino-mega-2560>.
- [4] ธนบดี บุหลันศรีชาติ. [โปรเจค] ใช้ esp32-cam เป็นกล้องวงจรปิดแบบพกพา. (2563). สืบค้นเมื่อ 8 ตุลาคม 2564. สืบค้นจาก: <https://www.modulemore.com/article/18/โปรเจค-ใช้-esp32-cam-เป็นกล้องวงจรปิดแบบพกพา/>.
- [5] prajak_kajchapon. ESP32 CAM with OV2640 module โมดูล ESP32-Cam พร้อมกล้อง OV2640 สำหรับ Arduino. (2563). สืบค้นเมื่อ 8 ตุลาคม 2564. สืบค้นจาก: <https://shopee.co.th/ESP32-CAM-with-OV2640-module-โมดูล-ESP32-Cam-พร้อมกล้อง-OV2640-สำหรับ-Arduino-i.56846818.3638705728>.
- [6] SP Micro Tech. โมดูล ESP32-CAM ESP32 พร้อมกล้อง OV2640 ESP32 WiFi + Bluetooth with OV2640 module. (2562). สืบค้นเมื่อ 8 ตุลาคม 2564. สืบค้นจาก: <https://www.spmicrotech.com/product/โมดูล-esp32-cam-พร้อม-กล้อง-ov2640-esp32-cam-wifi-bluetooth-with-ov2640-module/>.
- [7] mcucity. FT232RL USB TO TTL 5V 3.3V Download Cable To Serial Adapter Module For Arduino. (2559). สืบค้นเมื่อ 8 ตุลาคม 2564. สืบค้นจาก: <https://www.mcucity.com/product/45/ft232rl-usb-to-ttl-5v-3-3v-download-cable-to-serial-adapter-module-for-arduino>.
- [8] arduino4. FT232RL USB To Serial Adapter Module USB TO 232 Download Cable For Arduino. (2561). สืบค้นเมื่อ 8 ตุลาคม 2564. สืบค้นจาก: <https://www.arduino4.com/product/453/ft232rl-usb-to-serial-adapter-module-usb-to-232-download-cable-for-arduino-2>.

- [9] lnwshop. Line Track Sensor TCRT5000 IR Reflective Module. (2562). สืบค้นเมื่อ 16 พฤศจิกายน 2564. สืบค้นจาก: <http://fitrox.lnwshop.com/product/32/line-track-sensor-tcrt5000-ir-reflective-module>
- [10] allnewstep. TCRT5000 infrared reflectance Obstacle avoidance line tracking sensor เซนเซอร์เช็คสิ่งกีดขวาง เส้นขาวดำ นับจำนวน. (2560). สืบค้นเมื่อ 16 พฤศจิกายน 2564. สืบค้นจาก: <https://www.allnewstep.com/product/698/tcrt5000-infrared-reflectance-obstacle-avoidance-line-tracking-sensor-เซนเซอร์เช็คสิ่งกีดขวาง-เส้นขาวดำ>
- [11] arduino4. TCRT5000 Reflective Infrared Photoelectric Switches Module. (2559). สืบค้นเมื่อ 16 พฤศจิกายน 2564. สืบค้นจาก: <https://www.arduino4.com/product/95/tcrt5000-reflective-infrared-photoelectric-switches-module>.
- [12] cybertice. สอนใช้งาน เซนเซอร์ วัดระยะทาง HC-SR04 HC-SR05 US-100 US-016. (2562). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <https://www.cybertice.com/article/66/สอนใช้งาน-เซนเซอร์-วัดระยะทาง-hc-sr04-hc-sr05-us-100-us-016>.
- [13] Osoyoo. Micro bit lesson — Using the Ultrasonic Module. (2561). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/>.
- [14] fitrox. การใช้งาน HC-SR04 Ultrasonic Module. (2562). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <http://fitrox.lnwshop.com/article/27/tutorial-การใช้งาน-hc-sr04-ultrasonic-module/>.
- [15] ros.outletsdiscount. Ultrasonic Sensor HC-SR04 Module. สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <https://ros.outletsdiscount.ru/content?c=echo%20trig%20ultrasonic&id=34>.
- [16] mosfex. L298N โมดูลขับมอเตอร์ DC 2A Dual Channel H-Bridge Motor Driver Controller L298N Module. (2561). สืบค้นเมื่อ 8 กันยายน 2564. สืบค้นจาก: <https://www.mosfex.com/product/199/l298n-โมดูลขับมอเตอร์-dc-2a-dual-channel-h-bridge-motor-driver-controller-l298n-module/>

- [17] SP Micro Tech. โมดูลเสียง Active Buzzer Module 3.3-5V ทำงานแบบ Active Low. (2562). สืบค้นเมื่อ 8 พฤศจิกายน 2564. สืบค้นจาก: <https://www.spmicrotech.com/product/active-buzzer-module-3-3-5v-ทำงานแบบ-active-low/>.
- [18] cr-engineer. มอเตอร์ไฟฟ้ากระแสตรง. (2560). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: http://www.cr-engineer.com/images/pulldown_1304840984/DC%20electric%20motors.pdf/.
- [19] Circuitshops. มอเตอร์เกียร์ zga60Fm. (2562). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <http://www.circuitshops.com/articles/42214934/มอเตอร์เกียร์-zga60Fm.html/>.
- [20] Support Thaieasyelec. การใช้งาน Character LCD Display กับ Arduino. (2561). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <https://blog.thaieasyelec.com/how-to-use-character-lcd-display-arduino-ch1-parallel/>.
- [21] kpciot. LCD 16x2 with I2C interface Blue backlight module. (2560). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <http://www.kpciot.com/product/6/lcd-16x2-with-i2c-interface-blue-backlight-module>.
- [22] Oldest. การใช้งานจอ Character LCD กับ Arduino แบบละเอียด. (2558). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <https://www.ioxhop.com/article/30/-character-lcd-/arduino-/>.
- [23] arduitrronics. Arduino with Keypad and 4 Channel Relay. (2557). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <https://www.arduitronics.com/article/31/arduino-with-keypad-and-4-channel-relay/>.
- [24] allnewstep. 4x4 Matrix Keypad Module. (2558). สืบค้นเมื่อ 3 มีนาคม 2565. สืบค้นจาก: <https://www.allnewstep.com/product/1025/4x4-matrix-keypad-module>
- [25] padandpad. มิเตอร์สำหรับวัดแรงดันพร้อมแถบวัดระดับไฟ สำหรับการชาร์จแบตเตอรี่ 12V LED Battery Digital Volt meter with Level Indicator. (2561). สืบค้นเมื่อ 16 กุมภาพันธ์ 2565. สืบค้นจาก: https://shopee.co.th/มิเตอร์สำหรับวัดแรงดันพร้อมแถบวัดระดับไฟ-สำหรับการชาร์จแบตเตอรี่-12V-LED-Battery-Digital-Volt-meter-with-Level-Indicator-i.57748591.1767339579?gclid=EAlaQobChMI5bjvusml9gIV6Z1LBR3yFwLdEAQYBSABEgljRPPD_BwE

- [26] allnewstep. สวิตช์กดติดปล่อยดับ/กดดับปล่อยติด/รีเซตสวิตช์ AC DC 10A ปุ่มกดสีน้ำเงิน. (2558). สืบค้นเมื่อ 18 กุมภาพันธ์ 2565. สืบค้นจาก: <https://www.allnewstep.com/product/2878/สวิตช์กดติดปล่อยดับ-กดดับปล่อยติด-รีเซตสวิตช์-ac-dc-10a-ปุ่มกดสีน้ำเงิน>.
- [27] jwtech. Push Button Switch คือ?. (2564). สืบค้นเมื่อ 18 กุมภาพันธ์ 2565. สืบค้นจาก: <http://jwtech.co.th/activity/?p=703>
- [28] allnewstep. สวิตช์ กดติดกดดับ กดดับกดติด ล็อกสวิตช์ Lock switch 220V 10A ปุ่มกดสีแดง. (2558). สืบค้นเมื่อ 18 กุมภาพันธ์ 2565. สืบค้นจาก: <https://www.allnewstep.com/product/2887/สวิตช์-กดติดกดดับ-กดดับกดติด-ล็อกสวิตช์-lock-switch-220v-10a-ปุ่มกดสีแดง-2>.
- [29] PATPHUM. โปรเจกต์ หุ่นยนต์เดินตามเส้น 2 เซ็นเซอร์ ARDUINO. (2563). สืบค้นเมื่อ 11 กันยายน 2564. สืบค้นจาก: <http://std.in.th/stem/2020/02/22/โปรเจกต์-หุ่นยนต์เดินตาม/>.
- [30] Firstcraft. Web application คืออะไร? ต่างจากเว็บไซต์ทั่วไปอย่างไร?. (2563). สืบค้นเมื่อ 20 มกราคม 2565. สืบค้นจาก: <https://1stcraft.com/website-application-vs-general-website/>
- [31] Quotesaboutsmile. ส่วนประกอบสำคัญที่ควรรู้ของ WEB APPLICATION ระบบดูแลเว็บไซต์. (2564). สืบค้นเมื่อ 20 มกราคม 2565. สืบค้นจาก: <https://quotesaboutsmile.com/2021/07/webapplication-2/>.
- [32] Ohm. ส่วนประกอบของ Web Application (Website). (2559). สืบค้นเมื่อ 20 มกราคม 2565. สืบค้นจาก: <https://blog.tamacorp.co/ส่วนประกอบของ-web-application-website/>.
- [33] dailytech. เว็บไซต์แอปพลิเคชัน(Web Application) คืออะไร. (2562). สืบค้นเมื่อ 20 มกราคม 2565. สืบค้นจาก: <https://www.dailytech.in.th/web-application-เว็บไซต์-แอปพลิเคชัน>.
- [34] ipsecsystems. Web Application Development Services. สืบค้นเมื่อ 20 มกราคม 2565. สืบค้นจาก: <https://ipsecsystems.com/web-application-development-services/>.
- [35] commandrone. ปุ่มกดติด-ปล่อยดับ 3A 250V สีแดง. (2562) สืบค้นเมื่อ 18 กุมภาพันธ์ 2565. สืบค้นจาก: <https://commandronestore.com/products/bg051.php>.



ภาคผนวก ก

คำสั่งควบคุมการทำงานของระบบการเดินตามเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
int sensor6 = 18;

// IR Sensors
int sensor1 = 2; // Left most sensor
int sensor2 = 3;
int sensor3 = 4;
int sensor4 = 5; // Right most sensor
int sensor5 = 6 ;
// Initial Values of Sensors
int sensor[6] = {0, 0, 0, 0, 0, 0};

// Motor Variables
int ENA = 12;
int motorInput1 = 7;
int motorInput2 = 8;
int motorInput3 = 9;
int motorInput4 = 10;
int ENB = 11;
//Example By ArduinoAll
const int echo = 14;
int trig = 15;
//Initial Speed of Motor
int initial_motor_speed = 255;

int N;
// PID Constants
float Kp = 60 ;
float Ki = 20;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float Kd = 10;

float error = 0, P = 0, I = 0, D = 0, PID_value = 0;
float previous_error = 0, previous_I = 0;
int Buzzer = 30; //Buzzer
int i = 0;
int flag = 0;
//keypad
int e;
int left_motor_speed;
int right_motor_speed;

int c;
int num;
int knum;
char BufNum[8];
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {39, 41, 43, 45}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {31, 33, 35, 37}; //connect to the column pinouts of the
keypad{39, 41, 43, 45}

Keypad keypad_key = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
LiquidCrystal_I2C lcd(0x27, 16, 2); // PCF8574A = 0x3F

```

```

//ส่วนของ esp32 cam
int IN5 = 47;
int IN6 = 49;
int IN7 = 51;
int IN8 = 53;

void setup()
{
  //Serial.begin(9600);
  lcd.begin();
  lcd.setCursor(0, 0);
  lcd.print("Keypad Test ");
  lcd.setCursor(0, 1);
  lcd.print("Key      = ");
  pinMode(Buzzer, OUTPUT);
  digitalWrite(Buzzer, HIGH);
  lcd.blink();

  num = GetNum(4, 12, 1);
  Beep();
  Serial.print(" Keypad Test ");
  Serial.println("setCursor");
  Serial.println("Key      = ");
  Serial.println("setCursor");

  pinMode(sensor1, INPUT);
  pinMode(sensor2, INPUT);
  pinMode(sensor3, INPUT);
  pinMode(sensor4, INPUT);
  pinMode(sensor5, INPUT);
  // กำหนด input ของ esp32cam

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pinMode(IN5, INPUT);
pinMode(IN6, INPUT);
pinMode(IN7, INPUT);
pinMode(IN8, INPUT);

pinMode(motorInput1, OUTPUT);
pinMode(motorInput2, OUTPUT);
pinMode(motorInput3, OUTPUT);
pinMode(motorInput4, OUTPUT);
pinMode(ENA, OUTPUT);
pinMode(ENB, OUTPUT);
pinMode(sensor6, INPUT);

Serial.begin(9600);//setting serial monitor at a default baud rate of 9600
Serial.println("AUTOMATIC TRANSPORT OF MEDICAL SUPPLIES");
}
void loop()
{
  lcd.setCursor(12, 0);
  lcd.print(num);
  read_sensor_values();
  Serial.println(error);
  // Serial.begin(9600);
  //Ultrasonic sensor
  long duration, cm;
  pinMode(echo, OUTPUT);
  digitalWrite(echo, 0);
  delayMicroseconds(2);
  digitalWrite(echo, 1);
  delayMicroseconds(5);
  digitalWrite(echo, 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pinMode(trig, INPUT);
duration = pulseIn(trig, 1);
cm = (duration / 2) / 29.1;
Serial.print(cm);
Serial.print("cm");
Serial.print(sensor[5]);
Serial.print("sensorIR_State");
Serial.println();
delay(100);
Serial.print("num = ");
Serial.println(num);
Serial.print(sensor[0]);
Serial.print(" ");
Serial.print(sensor[1]);
Serial.print(" ");
Serial.print(sensor[2]);
Serial.print(" ");
Serial.print(sensor[3]);
Serial.print(" ");
Serial.print(sensor[4]);
Serial.print(" ");
//ควบคุมผ่าน esp32cam
if (num == 9) {

    app_contlor();

}
//ส่วนเดินตามเส้น
if (num == 0)
{
    TUT();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if (num == 1)
{
    room1();
}
if (num == 2)
{
    room2();
}
if (num == 3)
{
    room3();
}
if (num == 4)
{
    room4();
}
if ((num == 12) || (num == 21))
{
    room12();
}
if ((num == 13) || (num == 31))
{
    room13();
}
if ((num == 14) || (num == 41))
{
    room14();
}
if ((num == 23) || (num == 32))
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    room23();
}
if ((num == 24) || (num == 42))
{
    room24();
}
if ((num == 34) || (num == 43))
{
    room34();
}
if ((num == 123) || (num == 132) || (num == 213) || (num == 231) || (num == 321) ||
(num == 312))
{
    room123();
}
if ((num == 124) || (num == 142) || (num == 214) || (num == 241) || (num == 412) ||
(num == 421))
{
    room124();
}
if ((num == 134) || (num == 143) || (num == 314) || (num == 341) || (num == 413) ||
(num == 431))
{
    room134();
}
if ((num == 234) || (num == 243) || (num == 324) || (num == 342) || (num == 432) ||
(num == 423))
{
    room234();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if ((num == 1234) || (num == 1342) || (num == 1324) || (num == 1243) || (num == 1423)
        || (num == 1432) || (num == 2341) || (num == 2314) || (num == 2431) || (num == 2413) ||
        (num == 2134) || (num == 2143) || (num == 3412) || (num == 3421) || (num == 3142) ||
        (num == 3124) || (num == 3241) || (num == 3214) || (num == 4123) || (num == 4132) ||
        (num == 4213) || (num == 4231) || (num == 4312) || (num == 4321))
    {
        room1234();
    }
}

void SlideNum(void)
{
    BufNum[6] = BufNum[5];
    BufNum[5] = BufNum[4];
    BufNum[4] = BufNum[3];
    BufNum[3] = BufNum[2];
    BufNum[2] = BufNum[1];
    BufNum[1] = BufNum[0];
}

unsigned long GetNum(int Count, int X, int Y)
{ char Key, i, N;
  int Sum;
  lcd.setCursor(X + (Count - 1), Y);
  N = 0;
  for (i = 0; i < Count; i++) {
    BufNum[i] = ' ';
  }
  i = 0;
  while (Key != '#')
  { Key = NO_KEY;
    while (Key == NO_KEY) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Key = keypad_key.getKey();
}
Beep(); delay(300);
if (Key == '*') {
    for (i = 0; i < Count; i++) {
        BufNum[i] = ' ';
    } N = 0;
}
if ((N < Count) && (Key >= '0') && (Key <= '9'))
{ SlideNum();
  BufNum[0] = Key;
  N++;
}
lcd.setCursor(X, Y);
for (i = 0; i < Count; i++) {
    lcd.print(BufNum[Count - (i + 1)]);
}
lcd.setCursor(X + (Count - 1), Y);

}
Sum = 0;
for (i = 0; i < Count; i++)
{
    if (BufNum[Count - (i + 1)] == 0x20) {
        BufNum[Count - (i + 1)] = '0';
    }
    Sum = (Sum * 10) + (BufNum[Count - (i + 1)] - '0');
}
lcd.noCursor();
return (Sum);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void read_sensor_values()
{
  sensor[0] = digitalRead(sensor1);
  sensor[1] = digitalRead(sensor2);
  sensor[2] = digitalRead(sensor3);
  sensor[3] = digitalRead(sensor4);
  sensor[4] = digitalRead(sensor5);
  sensor[5] = digitalRead(sensor6);
  Serial.print(" ");
  Serial.print(sensor[0]);
  Serial.print(" ");
  Serial.print(sensor[1]);
  Serial.print(" ");
  Serial.print(sensor[2]);
  Serial.print(" ");
  Serial.print(sensor[3]);
  Serial.print(" ");
  Serial.print(sensor[4]);
  Serial.println(" ");
  Serial.print(left_motor_speed);
  Serial.print(" ");
  Serial.print(right_motor_speed);
  Serial.println(" ");
  if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0) &&
(sensor[4] == 0))
    error = 4;
  else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 0) &&
(sensor[4] == 0))
    error = 3;
  else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 0) &&
(sensor[4] == 0))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    error = 2.5;
    else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 0) &&
(sensor[4] == 0))
        error = 1;
    else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 0) &&
(sensor[4] == 0))
        error = 0;
    else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 1) &&
(sensor[4] == 0))
        error = -1;
    else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 1) &&
(sensor[4] == 0))
        error = -2.5;
    else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 1) &&
(sensor[4] == 1))
        error = -3;
    else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0) &&
(sensor[4] == 1))
        error = -4;
}

void calculate_pid()
{
    P = error;
    I = I + previous_I;
    D = error - previous_error;

    PID_value = (Kp * P) + (Ki * I) + (Kd * D);

    previous_I = I;
    previous_error = error;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void motor_control()
{
  // Calculating the effective motor speed:
  int left_motor_speed = initial_motor_speed - PID_value;
  int right_motor_speed = initial_motor_speed + PID_value;

  // The motor speed should not exceed the max PWM value
  left_motor_speed = constrain(left_motor_speed, 0, 255);
  right_motor_speed = constrain(right_motor_speed, 0, 255);

  analogWrite(ENA, left_motor_speed); //Left Motor Speed
  analogWrite(ENB, right_motor_speed); //Right Motor Speed
  Serial.print("\t");
  Serial.print(left_motor_speed);
  Serial.print("\t");
  Serial.println(right_motor_speed);
  //following lines of code are to make the bot move forward
  forward();
}

void forward()
{
  /*The pin numbers and high, low values might be different depending on your
  connections */
  digitalWrite(motorInput1, LOW);
  digitalWrite(motorInput2, HIGH);
  digitalWrite(motorInput3, LOW);
  digitalWrite(motorInput4, HIGH);
}

void reverse()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  /*The pin numbers and high, low values might be different depending on your
connections */
  digitalWrite(motorInput1, HIGH);
  digitalWrite(motorInput2, LOW);
  digitalWrite(motorInput3, HIGH);
  digitalWrite(motorInput4, LOW);
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
}
void right()
{
  /*The pin numbers and high, low values might be different depending on your
connections */
  digitalWrite(motorInput1, LOW);
  digitalWrite(motorInput2, HIGH);
  digitalWrite(motorInput3, LOW);
  digitalWrite(motorInput4, LOW);
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
  Serial.print(sensor[0]);
  Serial.print(" ");
  Serial.print(sensor[1]);
  Serial.print(" ");
  Serial.print(sensor[2]);
  Serial.print(" ");
  Serial.print(sensor[3]);
  Serial.print(" ");
  Serial.print(sensor[4]);

  Serial.println(" ");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void left()
{
  /*The pin numbers and high, low values might be different depending on your
connections */
  digitalWrite(motorInput1, LOW);
  digitalWrite(motorInput2, LOW);
  digitalWrite(motorInput3, LOW);
  digitalWrite(motorInput4, HIGH);
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
  Serial.print(sensor[0]);
  Serial.print(" ");
  Serial.print(sensor[1]);
  Serial.print(" ");
  Serial.print(sensor[2]);
  Serial.print(" ");
  Serial.print(sensor[3]);
  Serial.print(" ");
  Serial.print(sensor[4]);
  Serial.println(" ");
}
void sharpRightTurn() {
  digitalWrite(motorInput1, LOW);
  digitalWrite(motorInput2, HIGH);
  digitalWrite(motorInput3, HIGH);
  digitalWrite(motorInput4, LOW);
  Serial.print(sensor[0]);
  Serial.print(" ");
  Serial.print(sensor[1]);
  Serial.print(" ");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(sensor[2]);
Serial.print(" ");
Serial.print(sensor[3]);
Serial.print(" ");
Serial.print(sensor[4]);
Serial.println(" ");
}
void sharpLeftTurn()
{
digitalWrite(motorInput1, HIGH);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, HIGH);
Serial.print(sensor[0]);
Serial.print(" ");
Serial.print(sensor[1]);
Serial.print(" ");
Serial.print(sensor[2]);
Serial.print(" ");
Serial.print(sensor[3]);
Serial.print(" ");
Serial.print(sensor[4]);
Serial.println(" ");
}
void stop_bot()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, LOW);
digitalWrite(ENA, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(ENB, HIGH);
read_sensor_values();
}
void stop_botend()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, LOW);
Beep2();
}
void Beep1()
{
Beep();
delay(1000);
Beep();
delay(1000);
Beep();
delay(1000);
}
void Beep2()
{
Beep();
}
void Beep()
{
digitalWrite(Buzzer, LOW);
delay(50);
digitalWrite(Buzzer, HIGH);
}
void Left()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  do {
    read_sensor_values();
    analogWrite(ENA, 160 ); //Left Motor Speed
    analogWrite(ENB, 160); //Right Motor Speed
    sharpLeftTurn();

    Serial.print(PID_value);
    Serial.print("\t");
    Serial.print(left_motor_speed);
    Serial.print("\t");
    Serial.println(right_motor_speed);
  } while (error != 4) ;
}
void Right()
{
  do {
    read_sensor_values();
    analogWrite(ENA, 160); //Left Motor Speed
    analogWrite(ENB, 160); //Right Motor Speed
    sharpRightTurn();

    Serial.print(e);
    Serial.print(PID_value);
    Serial.print("\t");
    Serial.print(left_motor_speed);
    Serial.print("\t");
    Serial.println(right_motor_speed);
  } while (error != -4) ;
}

```

```

void UTurn()          //เป้าหมายหยุด และยูเทรน
{
  Beep();
  do
  {
    analogWrite(ENA, 140); //Left Motor Speed
    analogWrite(ENB, 140); //Right Motor Speed
    sharpLeftTurn();
    read_sensor_values();
  }
  while (error != 2.5);
}
void UTurn_eng()     //เป้าหมายหยุด และยูเทรน
{
  Beep();
  do
  {
    analogWrite(ENA, 140); //Left Motor Speed
    analogWrite(ENB, 140); //Right Motor Speed
    sharpLeftTurn();
    read_sensor_values();
  }
  while (error != 0);
}
void untrasonic()
{
  //Ultrasonic sensor
  long duration, cm;
  pinMode(echo, OUTPUT);
  digitalWrite(echo, 0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delayMicroseconds(2);
digitalWrite(echo, 1);
delayMicroseconds(5);
digitalWrite(echo, 0);
pinMode(trig, INPUT);
duration = pulseIn(trig, 1);
cm = microsecondsToCentimeters(duration);
Serial.print(cm);
Serial.print("cm");
Serial.println();
Serial.print(sensor[5]);
Serial.print("sensorIR_State");
if (cm < 10)
{
  stop_bot();
  Beep();
  delay(2000);
}
}
void go_prograrm()
{
  read_sensor_values();
  calculate_pid();
  motor_control();
  Serial.print(left_motor_speed);
  Serial.print(" ");
  Serial.print(right_motor_speed);
  Serial.println(" ");
  //Ultrasonic sensor
  long duration, cm;
  pinMode(echo, OUTPUT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(echo, 0);
delayMicroseconds(2);
digitalWrite(echo, 1);
delayMicroseconds(5);
digitalWrite(echo, 0);
pinMode(trig, INPUT);
duration = pulseIn(trig, 1);
cm = microsecondsToCentimeters(duration);
Serial.print(cm);
Serial.print("cm");
Serial.println();
Serial.print(sensor[5]);
Serial.print("sensorIR_State");
if (cm < 10)
{
  stop_bot();
  Beep();
  delay(2000);
}
}
void Left_progarm()
{
  read_sensor_values();
  forward();
  delay(3000);
  Left();
  stop_bot();
  delay(500);
  forward();
  delay(500);
  read_sensor_values();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("Left");

}

void Right_progarm()
{
  read_sensor_values();
  forward();
  delay(3000);
  Right();
  stop_bot();
  delay(500);

  read_sensor_values();
  Serial.println("Right");
}

void UTurn_progarm()
{ int conter = 0;
  do {
    read_sensor_values();
    stop_bot();
    if (conter < 1)
    {
      Beep1();
    }
    Serial.println(conter);
    Serial.println(sensor[5]);
    if (sensor[5] == 0)
    {
      break;
    }
    conter += 1;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(100);
} while ((sensor[5] == 0) || (conter <= 200));
Beep1();
delay(500);
UTurn();
stop_bot();
delay(500);
}
void UTurn_progarm_eng()
{
    read_sensor_values();
    stop_bot();
    Beep2();
    UTurn_eng();
    stop_bot();
    delay(500);
    read_sensor_values();
    Serial.println("UTurn");
}
void end_progarm()
{
    delay(2000);
    UTurn_progarm_eng();
    setup();
}
//esp32 cam Control
void Forward32()
{
    digitalWrite(motorInput1, HIGH);
    digitalWrite(motorInput2, LOW);
    digitalWrite(motorInput3, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(motorInput4, LOW);
digitalWrite(ENA, HIGH);
digitalWrite(ENB, HIGH);
}
void Backward32()
{
digitalWrite(motorInput1, LOW);
digitalWrite(motorInput2, HIGH);
digitalWrite(motorInput3, LOW);
digitalWrite(motorInput4, HIGH);
digitalWrite(ENA, HIGH);
digitalWrite(ENB, HIGH);
}
void Left32()
{
digitalWrite(motorInput1, LOW);
analogWrite(motorInput2, 150);
analogWrite(motorInput3, 150);
digitalWrite(motorInput4, LOW);
digitalWrite(ENA, HIGH);
digitalWrite(ENB, HIGH);
}
void Right32()
{
analogWrite(motorInput1, 150);
digitalWrite(motorInput2, LOW);
digitalWrite(motorInput3, LOW);
analogWrite(motorInput4, 150]
digitalWrite(ENA, HIGH);
digitalWrite(ENB, HIGH);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Stop32()
{
    digitalWrite(motorInput1, LOW);
    digitalWrite(motorInput2, LOW);
    digitalWrite(motorInput3, LOW);
    digitalWrite(motorInput4, LOW);
    digitalWrite(ENA, LOW);
    digitalWrite(ENB, LOW);
}

long microsecondsToCentimeters(long microseconds)
{
    // ความเร็วเสียงเดินทางในอากาศคือ 340 เมตร/วินาที หรือ 29 ไมโครวินาที ต่อเซนติเมตร
    // เวลาที่ Ping ออกไปกระทบวัตถุจนถึงกลับมาที่ตัวรับ ใช้หารระยะทางได้
    // วัตถุอยู่มีระยะทางเท่ากับครึ่งหนึ่งของเวลาที่จับได้
    // เขียนเป็นสมการคำนวณระยะทาง ได้ดังนี้
    return microseconds / 29 / 2;
}

void app_contlor()
{
    //Serial.begin(9600);
    // เดินหน้า
    if ((digitalRead(IN5) == HIGH) && (digitalRead(IN6) == LOW) && (digitalRead(IN7) ==
HIGH) && (digitalRead(IN8) == LOW)) {
        Forward32();
    }
    // ถอยหลัง
    if ((digitalRead(IN5) == LOW) && (digitalRead(IN6) == HIGH) && (digitalRead(IN7) ==
LOW) && (digitalRead(IN8) == HIGH)) {
        Backward32();
    }
    // เลี้ยวซ้าย

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((digitalRead(IN5) == LOW) && (digitalRead(IN6) == HIGH) && (digitalRead(IN7) ==
HIGH) && (digitalRead(IN8) == LOW)) {
    Left32();
}
// เลี้ยวขวา
if ((digitalRead(IN5) == HIGH) && (digitalRead(IN6) == LOW) && (digitalRead(IN7) ==
LOW) && (digitalRead(IN8) == HIGH)) {
    Right32();
}
// หยุด
if ((digitalRead(IN5) == LOW) && (digitalRead(IN6) == LOW) && (digitalRead(IN7) ==
LOW) && (digitalRead(IN8) == LOW)) {
    Stop32();
}
}
void FF()
{
    while (1)
    {
        go_program();
        if ((digitalRead(sensor1) == 1) && (digitalRead(sensor3) == 1))
        {
            do
            {
                forward();
            } while ((digitalRead(sensor1) == 1));
            delay(500);
            break;
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void FL()
{
  while (1)
  {
    go_progarm();
    if ((digitalRead(sensor1) == 1) && (digitalRead(sensor3) == 1))
    {
      Left_progarm();
      break;
    }
  }
}

void FR()
{
  while (1)
  {
    go_progarm();
    if ((digitalRead(sensor1) == 1) && (digitalRead(sensor3) == 1))
    {
      do
      {
        Right_progarm();
      } while ((digitalRead(sensor5) == 1));
      break;
    }
  }
}

void FFT()
{
  while (1)
  {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

go_progam();
if ((digitalRead(sensor1) == 1) && (digitalRead(sensor4) == 1))
{
    end_progam();
    break;
}
}
}
void FUT()
{
while (1)
{
    read_sensor_values();
    calculate_pid();
    motor_control();
    untrasonic();
    if ((digitalRead(sensor2) == 1) && (digitalRead(sensor4) == 1))
    {
        UTurn_progam();
        break;
    }
}
}
void TUT()
{
while (1)
{
    go_progam();
    if ((digitalRead(sensor2) == 1) && (digitalRead(sensor4) == 1))
    {
        Left();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    stop_bot();
    delay(500);
    Beep();
    delay(1000);
  }
}
}
void room1()
{
  FL();
  FUT();
  FR();
  FFT();
}
void room12()
{
  FL();
  FUT();
  FL();
  FL();
  FUT();
  FR();
  FF();
  FFT();
}
void room123()
{
  FL();
  FUT();
  FL();
  FL();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FUT());
FF());
FUT());
FL());
FF());
FFT());
}
void room124()
{
FL());
FUT());
FL());
FL());
FUT());
FR());
FL());
FUT());
FL());
FFT());
}
void room13()
{
FL());
FUT());
FL());
FR());
FUT());
FL());
FF());
FFT());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void room134()
{
    FL();
    FUT();
    FL();
    FR();
    FUT();
    FL();
    FL();
    FUT();
    FL();
    FFT();
}
void room140()
{
    FL();
    FUT();
    FF();
    FUT();
    FL();
    FFT();
}
void room2()
{
    FF();
    FL();
    FUT();
    FR();
    FF();
    FFT();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void room23()
{
    FF();
    FL();
    FUT();
    FF();
    FUT();
    FL();
    FF();
    FFT();
}
void room234()
{
    FF();
    FL();
    FUT();
    FF();
    FUT();
    FL();
    FL();
    FUT();
    FL();
    FFT();
}
void room24()
{
    FF();
    FL();
    FUT();
    FR();
    FL();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FUT());
FL());
FFT());
}
void room3()
{
FF());
FR());
FUT());
FL());
FF());
FFT());
}
void room34()
{
FF());
FR());
FUT());
FL());
FL());
FUT());
FL());
FFT());
}
void room4()
{
FR());
FUT());
FL());
FFT());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void room1234()
{
    FL();
    FUT();
    FL();
    FL();
    FUT();
    FF();
    FUT();
    FL();
    FL();
    FUT();
    FL();
    FFT();
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "esp_camera.h"
#include <WiFi.h>

#define CAMERA_MODEL_AI_THINKER

const char* ssid = "Txpcr"; //Enter SSID WIFI Name
const char* password = "101001000"; //Enter WIFI Password

#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 21
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 19
#define Y4_GPIO_NUM 18
#define Y3_GPIO_NUM 5
#define Y2_GPIO_NUM 4
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26

```

```
#define SIOC_GPIO_NUM 27
```

```
#define Y9_GPIO_NUM 35
```

```
#define Y8_GPIO_NUM 34
```

```
#define Y7_GPIO_NUM 39
```

```
#define Y6_GPIO_NUM 36
```

```
#define Y5_GPIO_NUM 21
```

```
#define Y4_GPIO_NUM 19
```

```
#define Y3_GPIO_NUM 18
```

```
#define Y2_GPIO_NUM 5
```

```
#define VSYNC_GPIO_NUM 25
```

```
#define HREF_GPIO_NUM 23
```

```
#define PCLK_GPIO_NUM 22
```

```
#else
```

```
#error "Camera model not selected"
```

```
#endif
```

```
// GPIO Setting
```

```
extern int gpLb = 2; // Left 1
```

```
extern int gpLf = 14; // Left 2
```

```
extern int gpRb = 15; // Right 1
```

```
extern int gpRf = 13; // Right 2
```

```
extern int gpENA = 0;
```

```
extern int gpENB = 16;
```

```
extern int gpLed = 4; // Light
```

```
extern String WiFiAddr = "";
```

```
void startCameraServer();
```

```
void setup() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.begin(115200);
Serial.setDebugOutput(true);
Serial.println();

pinMode(gpLb, OUTPUT); //Left Backward
pinMode(gpLf, OUTPUT); //Left Forward
pinMode(gpRb, OUTPUT); //Right Forward
pinMode(gpRf, OUTPUT); //Right Backward
pinMode(gpLed, OUTPUT); //Light

//initialize
digitalWrite(gpLb, LOW);
digitalWrite(gpLf, LOW);
digitalWrite(gpRb, LOW);
digitalWrite(gpRf, LOW);
digitalWrite(gpLed, LOW);

camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//init with high specs to pre-allocate larger buffers
if(psramFound()){
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

//drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);

WiFi.begin(ssid, password);

```

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://'");
Serial.print(WiFi.localIP());
WiFiAddr = WiFi.localIP().toString();
Serial.println(" to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
}

#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"

extern int gpLb;
extern int gpLf;
extern int gpRb;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

extern int gpRf;
extern int gpENA;
extern int gpENB;
extern int gpLed;
extern String WiFiAddr;

void WheelAct(int nENA,int nLf, int nLb, int nRf, int nRb,int nENB);

typedef struct {
    size_t size; //number of values used for filtering
    size_t index; //current value index
    size_t count; //value count
    int sum;
    int * values; //array to be filled with values
} ra_filter_t;

typedef struct {
    httpd_req_t *req;
    size_t len;
} jpg_chunking_t;

#define PART_BOUNDARY "1234567890000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length:
%u\r\n\r\n";

static ra_filter_t ra_filter;
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t sample_size){
    memset(filter, 0, sizeof(ra_filter_t));

    filter->values = (int *)malloc(sample_size * sizeof(int));
    if(!filter->values){
        return NULL;
    }
    memset(filter->values, 0, sample_size * sizeof(int));

    filter->size = sample_size;
    return filter;
}

static int ra_filter_run(ra_filter_t * filter, int value){
    if(!filter->values){
        return value;
    }
    filter->sum -= filter->values[filter->index];
    filter->values[filter->index] = value;
    filter->sum += filter->values[filter->index];
    filter->index++;
    filter->index = filter->index % filter->size;
    if (filter->count < filter->size) {
        filter->count++;
    }
    return filter->sum / filter->count;
}

```

```

static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len){
    jpg_chunking_t *j = (jpg_chunking_t *)arg;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(!index){
    j->len = 0;
}
if(httpd_resp_send_chunk(j->req, (const char *)data, len) != ESP_OK){
    return 0;
}
j->len += len;
return len;
}

static esp_err_t capture_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    int64_t fr_start = esp_timer_get_time();

    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.printf("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }

    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline; filename=capture.jpg");

    size_t fb_len = 0;
    if(fb->format == PIXFORMAT_JPEG){
        fb_len = fb->len;
        res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
    } else {
        jpg_chunking_t jchunk = {req, 0};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    res = frame2jpg_cb(fb, 80, jpg_encode_stream, &jchunk)?ESP_OK:ESP_FAIL;
    httpd_resp_send_chunk(req, NULL, 0);
    fb_len = jchunk.len;
}
esp_camera_fb_return(fb);
int64_t fr_end = esp_timer_get_time();
Serial.printf("JPG: %uB %ums", (uint32_t)(fb_len), (uint32_t)((fr_end - fr_start)/1000));
return res;
}

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    static int64_t last_frame = 0;
    if(!last_frame) {
        last_frame = esp_timer_get_time();
    }

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }

    while(true){
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.printf("Camera capture failed");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    res = ESP_FAIL;
} else {
    if(fb->format != PIXFORMAT_JPEG){
        bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
        esp_camera_fb_return(fb);
        fb = NULL;
        if(!jpeg_converted){
            Serial.printf("JPEG compression failed");
            res = ESP_FAIL;
        }
    } else {
        _jpg_buf_len = fb->len;
        _jpg_buf = fb->buf;
    }
}
if(res == ESP_OK){
    size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
    res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
}
if(res == ESP_OK){
    res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
}
if(res == ESP_OK){
    res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
}
if(fb){
    esp_camera_fb_return(fb);
    fb = NULL;
    _jpg_buf = NULL;
} else if(!_jpg_buf){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        free(_jpg_buf);
        _jpg_buf = NULL;
    }
    if(res != ESP_OK){
        break;
    }
    int64_t fr_end = esp_timer_get_time();

    int64_t frame_time = fr_end - last_frame;
    last_frame = fr_end;
    frame_time /= 1000;
    uint32_t avg_frame_time = ra_filter_run(&ra_filter, frame_time);
    Serial.printf("MJPG: %uB %ums (%.1ffps), AVG: %ums (%.1ffps)"
        ,(uint32_t)(_jpg_buf_len),
        (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time,
        avg_frame_time, 1000.0 / avg_frame_time
    );
}

last_frame = 0;
return res;
}

static esp_err_t cmd_handler(httpd_req_t *req){
    char* buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};

    buf_len = httpd_req_get_url_query_len(req) + 1;
    if (buf_len > 1) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

buf = (char*)malloc(buf_len);
if(!buf){
    httpd_resp_send_500(req);
    return ESP_FAIL;
}
if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
    if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) == ESP_OK &&
        httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK) {
    } else {
        free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL;
    }
} else {
    free(buf);
    httpd_resp_send_404(req);
    return ESP_FAIL;
}
free(buf);
} else {
    httpd_resp_send_404(req);
    return ESP_FAIL;
}
}

int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;

if(!strcmp(variable, "framesize")) {
    if(s->pixformat == PIXFORMAT_JPEG) res = s->set_framesize(s, (framesize_t)val);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(!strcmp(variable, "quality")) res = s->set_quality(s, val);
else if(!strcmp(variable, "contrast")) res = s->set_contrast(s, val);
else if(!strcmp(variable, "brightness")) res = s->set_brightness(s, val);
else if(!strcmp(variable, "saturation")) res = s->set_saturation(s, val);
else if(!strcmp(variable, "gainceiling")) res = s->set_gainceiling(s, (gainceiling_t)val);
else if(!strcmp(variable, "colorbar")) res = s->set_colorbar(s, val);
else if(!strcmp(variable, "awb")) res = s->set_whitebal(s, val);
else if(!strcmp(variable, "agc")) res = s->set_gain_ctrl(s, val);
else if(!strcmp(variable, "aec")) res = s->set_exposure_ctrl(s, val);
else if(!strcmp(variable, "hmirror")) res = s->set_hmirror(s, val);
else if(!strcmp(variable, "vflip")) res = s->set_vflip(s, val);
else if(!strcmp(variable, "awb_gain")) res = s->set_awb_gain(s, val);
else if(!strcmp(variable, "agc_gain")) res = s->set_agc_gain(s, val);
else if(!strcmp(variable, "aec_value")) res = s->set_aec_value(s, val);
else if(!strcmp(variable, "aec2")) res = s->set_aec2(s, val);
else if(!strcmp(variable, "dcw")) res = s->set_dcw(s, val);
else if(!strcmp(variable, "bpc")) res = s->set_bpc(s, val);
else if(!strcmp(variable, "wpc")) res = s->set_wpc(s, val);
else if(!strcmp(variable, "raw_gma")) res = s->set_raw_gma(s, val);
else if(!strcmp(variable, "lenc")) res = s->set_lenc(s, val);
else if(!strcmp(variable, "special_effect")) res = s->set_special_effect(s, val);
else if(!strcmp(variable, "wb_mode")) res = s->set_wb_mode(s, val);
else if(!strcmp(variable, "ae_level")) res = s->set_ae_level(s, val);
else {
    res = -1;
}

if(res){
    return httpd_resp_send_500(req);
}

```

```

    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    return httpd_resp_send(req, NULL, 0);
}

```

```

static esp_err_t status_handler(httpd_req_t *req){
    static char json_response[1024];

    sensor_t * s = esp_camera_sensor_get();
    char * p = json_response;
    *p++ = '{';

    p+=sprintf(p, "\"framesize\":%u,", s->status.framesize);
    p+=sprintf(p, "\"quality\":%u,", s->status.quality);
    p+=sprintf(p, "\"brightness\":%d,", s->status.brightness);
    p+=sprintf(p, "\"contrast\":%d,", s->status.contrast);
    p+=sprintf(p, "\"saturation\":%d,", s->status.saturation);
    p+=sprintf(p, "\"special_effect\":%u,", s->status.special_effect);
    p+=sprintf(p, "\"wb_mode\":%u,", s->status.wb_mode);
    p+=sprintf(p, "\"awb\":%u,", s->status.awb);
    p+=sprintf(p, "\"awb_gain\":%u,", s->status.awb_gain);
    p+=sprintf(p, "\"aec\":%u,", s->status.aec);
    p+=sprintf(p, "\"aec2\":%u,", s->status.aec2);
    p+=sprintf(p, "\"ae_level\":%d,", s->status.ae_level);
    p+=sprintf(p, "\"aec_value\":%u,", s->status.aec_value);
    p+=sprintf(p, "\"agc\":%u,", s->status.agc);
    p+=sprintf(p, "\"agc_gain\":%u,", s->status.agc_gain);
    p+=sprintf(p, "\"gainceiling\":%u,", s->status.gainceiling);
    p+=sprintf(p, "\"bpc\":%u,", s->status.bpc);
    p+=sprintf(p, "\"wpc\":%u,", s->status.wpc);
    p+=sprintf(p, "\"raw_gma\":%u,", s->status.raw_gma);
    p+=sprintf(p, "\"lenc\":%u,", s->status.lenc);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

p+=sprintf(p, "\hmirror\:%u,", s->status.hmirror);
p+=sprintf(p, "\dcw\:%u,", s->status.dcw);
p+=sprintf(p, "\colorbar\:%u", s->status.colorbar);
*p++ = '}';
*p++ = 0;
httpd_resp_set_type(req, "application/json");
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, json_response, strlen(json_response));
}

static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    String page = "";
    page += "<meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=0\">\n";
    page += "<script>var xhttp = new XMLHttpRequest();</script>";
    page += "<script>function getsend(arg) { xhttp.open('GET', arg +'?' + new
Date().getTime(), true); xhttp.send() } </script>";
    page += "<p align=center><IMG SRC='http://" + WiFiAddr + ":81/stream'
style='width:280px;'></p><br/><br/>";
    //page += "<p align=center><IMG SRC='http://" + WiFiAddr + ":81/stream'
style='width:300px; transform:rotate(180deg);'></p><br/><br/>";

    page += "<p align=center> <button style=background-
color:lightgrey;width:90px;height:80px onmousedown=getsend('go')
onmouseup=getsend('stop') ontouchstart=getsend('go') ontouchend=getsend('stop')
><b>Forward</b></button> </p>";
    page += "<p align=center>";
    page += "<button style=background-color:lightgrey;width:90px;height:80px;
onmousedown=getsend('left') onmouseup=getsend('stop') ontouchstart=getsend('left')
ontouchend=getsend('stop')><b>Left</b></button>&nbsp;";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

page += "<button style=background-color:white;width:90px;height:80px
onmousedown=getsend('stop')
onmouseup=getsend('stop')><b>Stop</b></button>&nbsp;";
page += "<button style=background-color:lightgrey;width:90px;height:80px
onmousedown=getsend('right') onmouseup=getsend('stop')
ontouchstart=getsend('right') ontouchend=getsend('stop')><b>Right</b></button>";
page += "</p>";

page += "<p align=center><button style=background-
color:lightgrey;width:90px;height:80px onmousedown=getsend('back')
onmouseup=getsend('stop') ontouchstart=getsend('back') ontouchend=getsend('stop')
><b>Backward</b></button></p>";

page += "<p align=center>";
// page += "<button style=background-color:yellow;width:1 4 0 px;height:4 0 px
onmousedown=getsend('ledon')><b>Light ON</b></button>";
// page += "<button style=background-color:yellow;width:1 4 0 px;height:4 0 px
onmousedown=getsend('ledoff')><b>Light OFF</b></button>";
page += "</p>";

return httpd_resp_send(req, &page[0], strlen(&page[0]));
}

static esp_err_t go_handler(httpd_req_t *req){
    WheelAct(HIGH, HIGH, LOW, HIGH, LOW, HIGH);
    Serial.println("Go");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t back_handler(httpd_req_t *req){
    WheelAct(HIGH, LOW, HIGH, LOW, HIGH, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("Back");
httpd_resp_set_type(req, "text/html");
return httpd_resp_send(req, "OK", 2);
}

static esp_err_t left_handler(httpd_req_t *req){
    WheelAct(HIGH, LOW, HIGH, HIGH, LOW, HIGH);
    Serial.println("Left");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t right_handler(httpd_req_t *req){
    WheelAct(HIGH, HIGH, LOW, LOW, HIGH, HIGH);
    Serial.println("Right");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t stop_handler(httpd_req_t *req){
    WheelAct(HIGH, LOW, LOW, LOW, LOW, HIGH);
    Serial.println("Stop");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t ledon_handler(httpd_req_t *req){
    digitalWrite(gpLed, HIGH);
    Serial.println("LED ON");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static esp_err_t ledoff_handler(httpd_req_t *req){
    digitalWrite(gpLed, LOW);
    Serial.println("LED OFF");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    httpd_uri_t go_uri = {
        .uri      = "/go",
        .method   = HTTP_GET,
        .handler  = go_handler,
        .user_ctx = NULL
    };

    httpd_uri_t back_uri = {
        .uri      = "/back",
        .method   = HTTP_GET,
        .handler  = back_handler,
        .user_ctx = NULL
    };

    httpd_uri_t stop_uri = {
        .uri      = "/stop",
        .method   = HTTP_GET,
        .handler  = stop_handler,
        .user_ctx = NULL
    };
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

httpd_uri_t left_uri = {
    .uri      = "/left",
    .method   = HTTP_GET,
    .handler  = left_handler,
    .user_ctx = NULL
};

```

```

httpd_uri_t right_uri = {
    .uri      = "/right",
    .method   = HTTP_GET,
    .handler  = right_handler,
    .user_ctx = NULL
};

```

```

httpd_uri_t ledon_uri = {
    .uri      = "/ledon",
    .method   = HTTP_GET,
    .handler  = ledon_handler,
    .user_ctx = NULL
};

```

```

httpd_uri_t ledoff_uri = {
    .uri      = "/ledoff",
    .method   = HTTP_GET,
    .handler  = ledoff_handler,
    .user_ctx = NULL
};

```

```

httpd_uri_t index_uri = {
    .uri      = "/",
    .method   = HTTP_GET,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
.handler = index_handler,  
.user_ctx = NULL  
};  
  
httpd_uri_t status_uri = {  
.uri = "/status",  
.method = HTTP_GET,  
.handler = status_handler,  
.user_ctx = NULL  
};  
  
httpd_uri_t cmd_uri = {  
.uri = "/control",  
.method = HTTP_GET,  
.handler = cmd_handler,  
.user_ctx = NULL  
};  
  
httpd_uri_t capture_uri = {  
.uri = "/capture",  
.method = HTTP_GET,  
.handler = capture_handler,  
.user_ctx = NULL  
};  
  
httpd_uri_t stream_uri = {  
.uri = "/stream",  
.method = HTTP_GET,  
.handler = stream_handler,  
.user_ctx = NULL  
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ra_filter_init(&ra_filter, 20);
Serial.printf("Starting web server on port: '%d'", config.server_port);
if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &go_uri);
    httpd_register_uri_handler(camera_httpd, &back_uri);
    httpd_register_uri_handler(camera_httpd, &stop_uri);
    httpd_register_uri_handler(camera_httpd, &left_uri);
    httpd_register_uri_handler(camera_httpd, &right_uri);
    //httpd_register_uri_handler(camera_httpd, &ledon_uri);
    //httpd_register_uri_handler(camera_httpd, &ledoff_uri);
}

config.server_port += 1;
config.ctrl_port += 1;
Serial.printf("Starting stream server on port: '%d'", config.server_port);
if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
}
}

void WheelAct(int nENA, int nLf, int nLb, int nRf, int nRb, int nENB)
{
    //digitalWrite(gpENA, nENA);
    digitalWrite(gpLf, nLf);
    digitalWrite(gpLb, nLb);
    digitalWrite(gpRf, nRf);
    digitalWrite(gpRb, nRb);
    //digitalWrite(gpENB, nENB);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้