ระบบฐานกล้องวงจรปิดอัจฉริยะตรวจจับการเคลื่อนไหว

Smart CCTV Enhancement with Motion Tracking

ปาราเมศ บัวงาม

อรุชา ตรีสุรนรา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2565

ระบบฐานกล้องวงจรปิดอัจฉริยะตรวจจับความเคลื่อนไหว

Smart CCTV Enhancement with Motion Tracking

อรุชา ตรีสุรนรา

ปาราเมศ บัวงาม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2565

.....ประธานกรรมการสอบปริญญานิพนธ์

(อาจารย์ภูสิษฐ์ วงศ์เจตจันทร์)

.....กรรมการสอบ

(อาจารย์สุเทพ ทัพธวัช)

.....กรรมการทรงผู้คุณวุฒิ

(พลอากาศโท ดร.พาห์รณ สงวนโภคัย)

.....หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์

(พันตรี ดร.นรณัฐ สงวนศักดิ์โยธิน)

.....คณบดีคณะวิศวกรรมศาสตร์

(ผู้ช่วยศาสตราจารย์ ดร.ยงยุทธ นารายฎร์)

หัวข้อปริญญานิพนธ์	ระบบฐานกล้องวงจรปิดอัจฉริยะตรวจจับการเคลื่อนไหว		
หน่วยกิจปริญญานิพนธ์	2 หน่วยกิต		
รายชื่อคณะผู้จัดทำ	ปาราเมศ	บัวงาม	6204000009
	อรุชา	ตรีสุรนรา	6204000001
อาจารย์ที่ปรึกษา	อาจารย์ภูสิษฐ์	วงศ์เจตจันทร์	
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต		
ภาควิชา	วิศวกรรมคอมพิวเตอร์		
คณะ	วิศวกรรมศาสตร์		
ปีการศึกษา	2566		

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการพัฒนาชุดฐานสำหรับการติดตั้งกล้องวงจรปิดชนิดอินฟาเรด เพื่อที่จะนำไปติดตั้งกล้องในพื้นที่ที่อาจมีการสุ่มเสี่ยงต่อการมีบุคคลเดินผ่าน โดยที่ฐานของกล้องจะออกแบบบมาให้มีชุดตรวจจับการเคลื่อนไหวพีไออาร์ ที่ติดตั้งอยู่ด้านหน้าและด้านข้างของฐานกล้องจะตรวจจับด้วยความร้อน เมื่อมีบุคคลที่ผ่านมามอเตอร์เซอร์โวจะขับเคลื่อนให้ฐานกล้องเคลื่อนที่ให้หน้ากล้องไปอยู่ตรงกับบุคคลนั้นและตัวฐานกล้องจะสั่งให้ตัววัดระยะทางทำงานหลังจากที่มีคนเดินผ่านแล้ว โดยจะมีสัญญาณแจ้งเตือนกลับมาผ่านแอปพลิเคชันไลน์ ในการพัฒนาการเขียนโปรแกรมภาษาซีกับบอร์ดอาดูอิโน่ รุ่นยูโน่อาร์สาม รับค่าจากชุดพีไออาร์สามชุดในการทดสอบความไวในการตรวจจับ และระยะทางพบว่าตัวชุดตรวจจับการเคลื่อนไหวพีไออาร์สามารถวัดระยะทางได้สูงสุด 7 เมตร แต่เมื่อนำมาวัดระยะทางสามารถวัดระยะห่างได้มากที่สุดเพียง 5 เมตร ทำให้พื้นที่การทำงานของตัวกล้องนั้นแคบลง ทางผู้จัดทำจึงได้คิดแนวทางในการแก้ไขปัญหานี้โดยการนำโมดูลลำโพงเป็นส่วนเสริม เพื่อสามารถส่งเสียงแจ้งเตือนให้ผู้ใช้งานได้รับรู้และยังแจ้งเตือนให้ผู้บุกรุกหรือบุคคลที่เดินผ่านอยู่ในพื้นที่เฝ้าระวัง

คำสำคัญ: อาดูอิโน่, ภาษาซีพลัสพลัส, กล้องวงจรปิด, เซนเซอร์ตรวจจับการเคลื่อนไหว, เซนเซอร์วัดระยะทาง, ไฟแสดงสถานะ, มอเตอร์เซอร์โว, ไลน์นอติไฟ, ไลน์

อาจารย์ที่ปรึกษา.....

(อาจารย์ภูสิษฐ์ วงเจตจันทร์)

...../...../.....

Project Title	Smart CCTV Enhancement with Motion Tracking		
Credit	2 Units		
By	Paramet	Buangam	6204000009
	Aruchar	Treesuranara	6204000001
Advisor	Phoosis	Wongjetjun	
Degree	Bachelor of Engineering		
Major	Computer Engineering		
Faculty	Engineering		
Academic year	2023		

Abstract

The objective of this project was the development of a base for the installation of infrared CCTV. To install cameras in some areas may be risky or vulnerable to people passing by. The base of camera is designed with set of PIR motion detector installed in the front and side of the camera base, which has heat detection. When there is a person passing by, the servo motor will move the camera base to the front of the camera to monitor the person. The camera base will command the distance in meters to alert after someone walks past. A notification signal is sent through the LINE application. The programming was in C++ language with an Arduino board, Uno R3 version, to received values from three PIR sets of PIRs in the detection sensitivity test. The distance detection PIR motion detector can measure distances up to 7 meters. However, when measuring distance, a maximum distance of 5 meters was measured, thereby reducing the coverage area of the camera. The manufacturers should address this issue by using speaker modules as additional components with the ability to send alerts to users and alert intruders that they are in a restricted area.

Keywords: Arduino, C++, CCTV, Motion Sensor, Distance Measuring Sensor, LED, Servo Motor, LINE Notify, LINE

Approved by

.....

กิตติกรรมประกาศ

(Acknowledgement)

การปริญญานิพนธ์ฉบับนี้นั้นเป็นส่วนหนึ่งของหลักสูตรวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม ทางผู้จัดทำโครงงานได้รับความรู้และประสบการณ์ต่างๆที่สำคัญและมีค่ามากมาย สำหรับปริญญานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปได้ด้วยดีจากความร่วมมือและการสนับสนุนดังนี้

อาจารย์ภูสิษฐ์ วงศ์เจตจันทร์ อาจารย์ที่ปรึกษา

ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา คณะกรรมการสอบ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ และตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนาม ได้มีส่วนร่วมในการให้ข้อมูลให้ความช่วยเหลือในด้านต่างๆ เป็นที่ปรึกษาให้คำแนะนำสำคัญ ชี้แนะแนวทางที่เป็นประโยชน์ จนไปถึงการสอบปริญญานิพนธ์ ทำให้งานทุกอย่างประสบความสำเร็จไปได้ด้วยดี และทำโครงงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งทางผู้จัดทำขอขอบพระคุณอย่างสูงไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ
ปาราเมศ บัวงาม
อรุชา ตรีสุรนรา

สารบัญ

หน้า

บทคัดย่อ ก

Abstract ข

กิตติกรรมประกาศ ค

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา....1

1.2 วัตถุประสงค์....1

1.3 ขอบเขตของโครงงาน....2

1.4 ประโยชน์ที่ได้รับ....2

1.5 เครื่องมือ เครื่องวัด และอุปกรณ์....3

1.6 แผนการดำเนินการ....4

บทที่ 2 การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง

2.1 กล้องวงจรปิด (CCTV) 5

2.2 ประเภทกล้องวงจรปิด (CCTV) 6

2.3 Arduino....11

2.4 ประเภทบอร์ดของ Arduino ที่ใช้....13

2.5 เซนเซอร์ตรวจจับการเคลื่อนไหว....19

2.6 หลอดไฟ LED....24

สารบัญ (ต่อ)

หน้า

2.7 Servo Motor.....25

2.8 Tinkercad.....26

2.9 Arduino IDE.....27

2.10 ภาษา C++.....28

2.11 LINE.....29

2.12 LINE Notify.....30

บทที่ 3 การวิเคราะห์และออกแบบ

3.1 การออกแบบระบบ.....31

3.2 ภาพรวมการทำงานของโครงงาน.....32

3.3 หลักการทำงานของระบบ.....33

3.4 หลักการสร้างฐานของกล้องวงจรปิด.....35

3.5 การค้นหาองศาของฐานกล้องและการขยับตัวฐานกล้อง.....36

3.6 การเชื่อมต่อสายไฟระหว่างโมดูลกับบอร์ด.....37

3.7 หลักการทำงานของเซนเซอร์ตรวจจับการเคลื่อนไหว PIR.....38

3.8 การส่งข้อความแจ้งเตือนไปยังผู้ใช้งานผ่านทาง LINE Notify.....39

3.9 การส่งค่าที่ได้รับไปที่ MyPHP Database.....41

3.10 การอธิบายการทำงานของโปรแกรม.....43

สารบัญ (ต่อ)

หน้า

บทที่ 4 การทดลอง

4.1 ทดสอบการเคลื่อนไหวตามของโมดูลเซนเซอร์ PIR.....50

4.2 การทดสอบเซนเซอร์วัดระยะทาง.....52

4.3 การทดสอบมอเตอร์ Servo.....54

4.4 การทดสอบระบบส่งข้อความแจ้งเตือนผ่านทางแอพลิเคชัน LINE.....55

4.5 การทดสอบการส่งข้อมูลระหว่างบอร์ด Uno กับบอร์ด Uno.....57

4.6 การทดสอบการทำงานของฐานกล้องวงจรปิด.....59

บทที่ 5 สรุปผลรายงานและข้อเสนอแนะ

5.1 สรุปผลโครงงาน.....62

5.2 ปัญหาการดำเนินโครงงาน.....62

5.3 ข้อเสนอแนะ.....62

บรรณานุกรม.....63

ประวัติผู้จัดทำ.....64

สารบัญตาราง

หน้า

ตารางที่ 1.1 การวางแผนดำเนินงาน....4

ตารางที่ 2.1 การจัดแจงช่องเชื่อมต่อของ NodeMCU....18

ตารางที่ 2.2 การเปรียบเทียบแต่ละรุ่นของบอร์ด Arduino Uno....18

ตารางที่ 3.1 กำหนดตำแหน่งขาของอุปกรณ์....44

ตารางที่ 4.1 ผลการทดสอบเซนเซอร์ PIR....48

ตารางที่ 4.2 ผลการทดสอบเซนเซอร์วัดระยะทาง....49

ตารางที่ 4.3 ผลการทดสอบโมดูลเซนเซอร์ Servo....51

ตารางที่ 4.4 ผลการทดสอบของฐานกล้องวงจรปิด....56

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 ตัวอย่างของกล้องวงจรปิด....6
รูปที่ 2.2 ตัวอย่างของกล้องวงจรปิดประเภท Box Camera....7
รูปที่ 2.3 ตัวอย่างของกล้องวงจรปิดประเภท Bullet Analog....7
รูปที่ 2.4 ตัวอย่างของกล้องวงจรปิดประเภท Dome Analog....8
รูปที่ 2.5 ตัวอย่างของกล้องวงจรปิดประเภท Infrared Camera....9
รูปที่ 2.6 ตัวอย่างของกล้องวงจรปิดประเภท Speed Dome....10
รูปที่ 2.7 ตัวอย่างของกล้องวงจรปิดประเภท Hidden Camera....10
รูปที่ 2.8 ตัวอย่างของบอร์ดไมโครคอนโทรลเลอร์....11
รูปที่ 2.9 สัญลักษณ์ของ Arduino....12
รูปที่ 2.10 ตัวอย่างของบอร์ด Arduino Uno R3....13
รูปที่ 2.11 อธิบายแต่ละส่วนของบอร์ด Arduino....13
รูปที่ 2.12 ตัวอย่างโครงสร้างของบอร์ด Arduino Uno R3....15
รูปที่ 2.13 ตัวอย่างของบอร์ด NodeMCU....16
รูปที่ 2.14 ตัวอย่างโครงสร้างของบอร์ด NodeMCU....17
รูปที่ 2.15 ตัวอย่างการใช้งานของเซนเซอร์ตรวจจับการเคลื่อนไหว....19
รูปที่ 2.16 หลักการทำงานของเซนเซอร์ PIR....20
รูปที่ 2.17 หลักการทำงานของ Optical Displacement Sensor....21
รูปที่ 2.18 หลักการทำงานของ Linear Proximity Sensor....22

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.19 หลักการทำงานของ Ultrasonic Displacement Sensor.....22

รูปที่ 2.20 หลักการทำงานของ Contact Displacement Sensor.....23

รูปที่ 2.21 โครงสร้างภายในของ LED.....24

รูปที่ 2.22 โครงสร้างภายในของ Servo Motor.....25

รูปที่ 2.23 ตัวอย่างการทดลองใช้งาน Tinkercad's Circuits.....26

รูปที่ 2.24 ตัวอย่างของซอฟต์แวร์ Arduino IDE.....27

รูปที่ 2.25 เครื่องหมายของภาษาคอมพิวเตอร์ C++.....28

รูปที่ 2.26 ไอคอนของแอพลิเคชัน LINE.....29

รูปที่ 2.27 ตัวอย่างภาพ LINE Notify.....30

รูปที่ 3.1 ภาพรวมของการทำงานของฐาน.....32

รูปที่ 3.2 Flow Chart หลักการทำงานของระบบ.....34

รูปที่ 3.3 การออกแบบฐานของกล้องวงจรปิด.....35

รูปที่ 3.4 หลักการทำงานของ Infrared tracking.....36

รูปที่ 3.5 โครงสร้างภายในของโมดูล Servo Motor.....36

รูปที่ 3.6 การเชื่อมต่อระหว่างบอร์ดกับ PIR Sensor และ Servo Motor.....37

รูปที่ 3.7 ตัวอย่างการทำงานของเซนเซอร์ PIR.....38

รูปที่ 3.8 ตัวอย่างโครงสร้างภายในของเซนเซอร์ PIR.....38

รูปที่ 3.9 หน้าเว็บ LINE Notify.....39

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.10 เข้าสู่ระบบ LINE Notify....39
รูปที่ 3.11 หน้า My page....40
รูปที่ 3.12 Generate token....40
รูปที่ 3.13 ตัวอย่าง Token ของ LINE Notify....41
รูปที่ 3.14 การเขียนโปรแกรม Visual Studio Code....42
รูปที่ 3.15 ตัวอย่างการเก็บข้อมูลภายใน MyPHP....42
รูปที่ 3.16 โปรแกรมการเรียกใช้งานและกำหนดขา....43
รูปที่ 3.17 การกำหนด Token ของ Line Notify....44
รูปที่ 3.18 การกำหนดตัวแปรและการควบคุม....45
รูปที่ 3.19 การเขียนโปรแกรมของโมดูลวัดระยะทาง....46
รูปที่ 3.20 การทำงานแบบวนซ้ำของฐานกล้องวงจรปิด....47
รูปที่ 3.21 โปรแกรมการส่งข้อความผ่านทาง LINE Notify....48
รูปที่ 3.22 การตั้งค่าเกี่ยวกับการส่งข้อความของ LINE Notify....49
รูปที่ 4.1 การทดสอบโมดูลเซนเซอร์ PIR....51
รูปที่ 4.2 การทดสอบโมดูลเซนเซอร์วัดระยะทาง....53
รูปที่ 4.3 การทดสอบโมดูลเซนเซอร์ Servo....55
รูปที่ 4.4 การเขียนโปรแกรมส่งข้อความ....56
รูปที่ 4.5 การส่งข้อความแจ้งเตือน....57

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.6 ตัวอย่างการเชื่อมต่อระหว่างบอร์ด Arduino.....58

รูปที่ 4.7 ภายในของฐานกล้องวงจรปิด.....60

รูปที่ 4.8 ฐานกล้องวงจรปิด.....61

บทที่ 1

บทนำ

1.1 ประวัติความเป็นมาและความสำคัญของปัญหา

ปัญหาผู้บุกรุกบุกเข้าบ้านพักอาศัยของผู้อื่น โดยมีจุดประสงค์เป็นการชิงทรัพย์หรือการกระทำที่ทำให้เกิดความเดือดร้อนและอันตรายผู้อื่นนั้นมีอยู่มากในสังคมไทย โดยทางเจ้าของที่พักอาศัยไม่สามารถจัดการปัญหานี้ได้อย่างมีประสิทธิภาพ เนื่องจากปัจจัยต่างๆอาทิเช่น เวลาทำงานการอาศัยอยู่คนเดียว และภายในที่พักอาศัยไม่มีอุปกรณ์สำหรับป้องกันผู้บุกรุก หรือไม่ได้ตรวจเช็คสภาพประตูและหน้าต่าง เป็นต้น เจ้าของที่พักอาศัยจะเลือกที่จะติดตั้งระบบกล้องวงจรปิดเพื่อแก้ปัญหา ดูแลความปลอดภัย ความเรียบร้อยภายในที่พักอาศัยและป้องกันทรัพย์สินของตน

เมื่อนำประสิทธิภาพและการใช้งานของกล้องวงจรปิดมาเปรียบเทียบกัน ปัจจุบันจะมีฟังก์ชันการทำงานที่หลากหลายและการใช้งานที่แตกต่างกัน ขึ้นอยู่กับผู้ใช้งานว่ามีความต้องการนำไปใช้งานแบบใด ดังนั้นทางผู้จัดทำจึงได้เสนอความคิดในการสร้างฐานสำหรับวางกล้องวงจรปิดที่มีฟังก์ชันการทำงานแบบกล้องวงจรปิดประเภท Infrared Camera ด้วยการอาศัยความรู้ทางด้าน IoT และบอร์ด Arduino เข้ามาช่วยในการประยุกต์และดัดแปลง โดยมีการนำเซนเซอร์ตรวจจับวัตถุ PIR เซนเซอร์วัดระยะทาง ประกอบเข้ากับฐานเพื่อทดแทนฟังก์ชันสำหรับตรวจจับการเคลื่อนไหว ฟังก์ชันการวัดระยะห่างของวัตถุจากกล้องวงจรปิด เพิ่มระบบส่งเสียงแจ้งเตือนใน Line Notify และมอเตอร์ Servo ประกอบเข้ากับจานหมุนของฐานเพื่อเพิ่มระบบการหมุนสำหรับกล้องวงจรปิดแบบตั้งจุด

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อนำความสามารถของบอร์ด Arduino มาประยุกต์ใช้เป็นฐานกล้องวงจรปิดได้

1.2.2 เพื่อศึกษาการทำงานและการใช้งานของบอร์ด Arduino

1.2.3 เพื่อศึกษาการใช้งานภาษา C++

1.2.4 เพื่อศึกษาสร้างและออกแบบแผงวงจรของ Arduino

1.3 ขอบเขตของการทำโครงงาน

1.3.1 ฐานกล้องวงจรปิดที่สามารถนำไปใช้กับกล้องวงจรปิดแบบเก่าที่ไม่มีฟังก์ชันการติดตามการเคลื่อนไหว

1.3.2 สามารถตรวจจับการเคลื่อนไหวของสิ่งมีชีวิตข้างหน้าของกล้อง

1.3.3 ฐานกล้องสามารถหมุนไปตามเซนเซอร์ที่ตรวจจับการเคลื่อนไหวได้

1.3.4 สามารถวัดระยะห่างระหว่างฐานกล้องกับวัตถุที่ตรวจพบว่าเคลื่อนที่ผ่านกล้องได้

1.3.5 สามารถส่งข้อความแจ้งเตือนให้กับผู้ใช้งานผ่าน Line Notify ได้

1.4 ประโยชน์ที่ได้รับ

1.4.1 สามารถเพิ่มความสามารถฐานกล้องให้กับกล้องวงจรปิดได้

1.4.2 ทำให้สถานที่อยู่อาศัยหรือจุดที่ผู้ใช้งานไม่สามารถดูแลได้อย่างทั่วถึงมีความปลอดภัยมากขึ้น

1.4.3 สามารถช่วยปกป้องเหตุร้ายหรือสิ่งที่ผู้ใช้งานไม่ต้องการให้เกิดขึ้น

1.4.4 สามารถตรวจจับได้ในเวลากลางวันและกลางคืนได้

1.4.5 สามารถใช้ดูแลและตรวจสอบความเรียบร้อยภายในพื้นที่อยู่อาศัยของตนเอง

1.5 เครื่องมือ เครื่องวัด และอุปกรณ์

1.5.1 ชุดบอร์ดผู้พัฒนาสำหรับ Arduino Uno R3	1	ชุด
1.5.2 โมดูลลำโพง Buzzer	1	ตัว
1.5.3 โมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR	3	ตัว
1.5.4 โมดูลวัดระยะทาง Ultrasonic	1	ตัว
1.5.5 กล้องวงจรปิดประเภท Infrared	1	ตัว
1.5.6 แผ่นกระดาน Breadboard	2	ชุด
1.5.7 สายไฟเชื่อมต่อผู้-เมีย	20	เส้น
1.5.8 ชุดบอร์ดผู้พัฒนาสำหรับ ESP8266	1	ชุด
1.5.9 เครื่องคอมพิวเตอร์	1	เครื่อง
1.5.10 กระดาษแข็งหรือวัสดุแข็งเรียบ	8	แผ่น
1.5.11 ชุด Servo Motor	1	ตัว
1.5.12 ชุดกาวร้อน	1	ชุด
1.5.13 ตัวสายแปลงไฟฟ้า 5V 3.3 m	1	สาย

1.6 แผนการดำเนินการ

ตารางที่ 1.1 การวางแผนดำเนินงาน

เลขที่	หัวข้องาน	2566								
		ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
1	ศึกษาและสืบค้นข้อมูลเกี่ยวกับโครงงาน									
2	นำเสนอหัวข้อและรูปแบบของโครงงาน									
3	ศึกษาหลักการทำงานของกล้องวงจรปิด									
4	ศึกษาหลักการทำงานของ Arduino									
5	ออกแบบฮาร์ดแวร์และการต่อวงจร									
6	ศึกษาการทำงานของโมดูลเสริมต่างๆ									
7	ทำการเขียนโปรแกรมและส่งลงบอร์ด Arduino									
8	ทดสอบการใช้งาน									
9	ปรับปรุงแก้ไขและจดบันทึก									
10	จัดทำปริญญานิพนธ์									

ตามแผนการ ตามงานจริง

บทที่ 2
การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง

ในการจัดทำโครงงานนี้ คณะผู้จัดทำได้ทำการศึกษาค้นคว้า ทฤษฎี แนวคิดและโปรแกรมทั้งหมดที่ใช้ในการทำโครงงานให้โครงงานสำเร็จตามวัตถุประสงค์และตอบโจทย์ตามที่ได้ตั้งเป้าหมายและคาดหวังไว้

2.1 กล้องวงจรปิด (CCTV)

กล้องวงจรปิด หรือ CCTV ย่อมากจาก Closed Circuit Television คือ ระบบกล้องโทรทัศน์ที่ถ่ายภาพแบบเคลื่อนไหว แล้วส่งกลับมาที่เครื่องบันทึกภาพ (DVR, NVR) เพื่อบันทึกภาพทั้งภาพนิ่งและภาพเคลื่อนไหวได้ และยังมีส่วนที่ไว้สำหรับดูภาพแบบเรียลไทม์หรือออนไลน์ผ่านทางจอภาพ (Monitor) เป็นระบบสำหรับใช้เพื่อการรักษาความปลอดภัย หรือใช้เพื่อสอดส่องดูแลเหตุการณ์และสถานการณ์ต่างๆ

การทำงานของ CCTV เริ่มต้นจากตัวกล้องที่เป็นตัวรับสัญญาณภาพ จะได้รับภาพจากแสงส่องไปยังวัตถุที่ต้องการ แสงนั้นจะตกกระทบวัตถุแล้วจึงสะท้อนกลับออกมา (ประสิทธิภาพกล้องนั้นขึ้นอยู่กับความไวของแสง ซึ่งมีผลให้คุณภาพของการทำงานแตกต่างกันออกไป) และภายในจะมีตัวที่แปลงสัญญาณภาพเป็นสัญญาณไฟฟ้า เพื่อส่งต่อสัญญาณตามสายที่เชื่อมจากกล้องไปสู่เครื่องรับสัญญาณภาพ (DVR) แล้วส่งต่อไปยังจอรับภาพ (Monitor) เพื่อแสดงภาพที่ได้จากตัวกล้อง โดยปกติตัวกล้องและจอภาพจะอยู่ต่างสถานที่กัน ซึ่งหลักการทำงานของระบบกล้องวงจรปิดไม่ซับซ้อนมาก แต่หากต้องการให้ภาพออกมาดี ต้องประกอบด้วยองค์ประกอบหลายอย่าง เช่น เครื่องบันทึกภาพ กล้องวงจรปิด ระบบควบคุม สายสัญญาณ เป็นต้น

รูปที่ 2.1 ตัวอย่างของกล้องวงจรปิด

2.2 ประเภทของกล้องวงจรปิด (CCTV)

กล้องวงจรปิดสามารถแบ่งออกได้เป็นหลายประเภทขึ้นอยู่กับการใช้งานและสถานที่ที่นำไปใช้งาน โดยความเหมาะสมและการเข้ากันได้ของพื้นที่จะเป็นตัวกำหนดประเภทของกล้องวงจรปิดนั้นๆไป โดยกล้องวงจรปิดสามารถแบ่งออกเป็นประเภทได้ดังนี้

2.2.1 Box Camera

เป็นกล้องมาตรฐานทรงกระบอก เหมาะสำหรับการติดตั้งภายในตัวบ้านหรืออาคารที่มีแสงสว่างเพียงพอต่อตัวกล้อง ตัวกล้องนั้นไม่มีระบบตรวจจับความร้อน (Infrared) ติดตั้งไว้ จึงจำเป็นต้องซื้อส่วนเสริมหรือเลนส์สำหรับกล้องวงจรปิดมาติดแยก

รูปที่ 2.2 ตัวอย่างของกล้องวงจรปิดประเภท Box Camera

2.2.2 Bullet Analog Camera

กล้องวงจรปิดทรงกระบอกมีทั้งระบบดิจิทัลและอนาล็อค ซึ่งขึ้นอยู่กับความต้องการของผู้ใช้งาน โดยตัวกล้องมีลักษณะกลมและยาว กล้องประเภทนี้จะมีเลนส์ที่ไม่สามารถเปลี่ยนแปลงขนาดได้ เหมาะต่อการติดตั้งที่บริเวณรั้วหรือตามแนวหลังคา เป็นต้น

รูปที่ 2.3 ตัวอย่างของกล้องวงจรปิดประเภท Bullet Analog

2.2.3 Dome Analog Camera

กล้องวงจรปิดที่มีรูปร่างเป็นโดมครอบและมีตัวฐานเรียบ มีระบบแบบดิจิทัลและแบบอนาล็อก ขึ้นอยู่กับความต้องการของผู้ใช้งาน โดยที่กล้องวงจรปิดประเภทนี้มีทั้งแบบติดตั้งและไม่ติดตั้งระบบตรวจจับความร้อน เหมาะต่อการติดตั้งไว้บนเพดานห้อง

รูปที่ 2.4 ตัวอย่างของกล้องวงจรปิดประเภท Dome Analog

2.2.4 Infrared Camera

กล้องที่มีหลอดไฟ LED อยู่รอบๆเลนส์ของกล้องวงจรปิด โดยมีทั้งแบบกล้องวงจรปิดแบบ Dome Analog และ Bullet Analog หลอดไฟ LED จะช่วยกล้องวงจรปิดสามารถถ่ายภาพตอนกลางคืนได้ชัดเจนมากขึ้น โดยในปัจจุบันได้มีการพัฒนาขึ้นเป็นหลอดไฟ LED Infrared ในภายหลัง หากมีจำนวน LED Infrared ยิ่งมาก ก็จะสามารถทำให้ถ่ายภาพตอนกลางคืนได้ไกลขึ้นและชัดเจนมากขึ้น

รูปที่ 2.5 ตัวอย่างของกล้องวงจรปิดประเภท Infrared Camera

2.2.5 Speed Dome PTZ Camera (Pan-Tilt-Zoom)

กล้องวงจรปิดประเภท Speed Dome มีลักษณะเป็นลูกบอลกลมใหญ่ ใช้สำหรับดูภาพบริเวณรอบๆตัวกล้อง สามารถตั้งค่าให้หมุนไปในตำแหน่งที่ต้องการได้ และสามารถขยายภาพได้มากกว่ากล้องวงจรปิดบางรุ่นได้ สามารถพบเห็นได้ตามห้างสรรพสินค้าหรือตามพื้นที่งานก่อสร้างขนาดใหญ่เพื่อดูความคืบหน้าของงาน เหมาะสมต่อการติดไว้ในที่สูงเพื่อที่จะสามารถมองเห็นบริเวณรอบๆได้

รูปที่ 2.6 ตัวอย่างของกล้องวงจรปิดประเภท Speed Dome

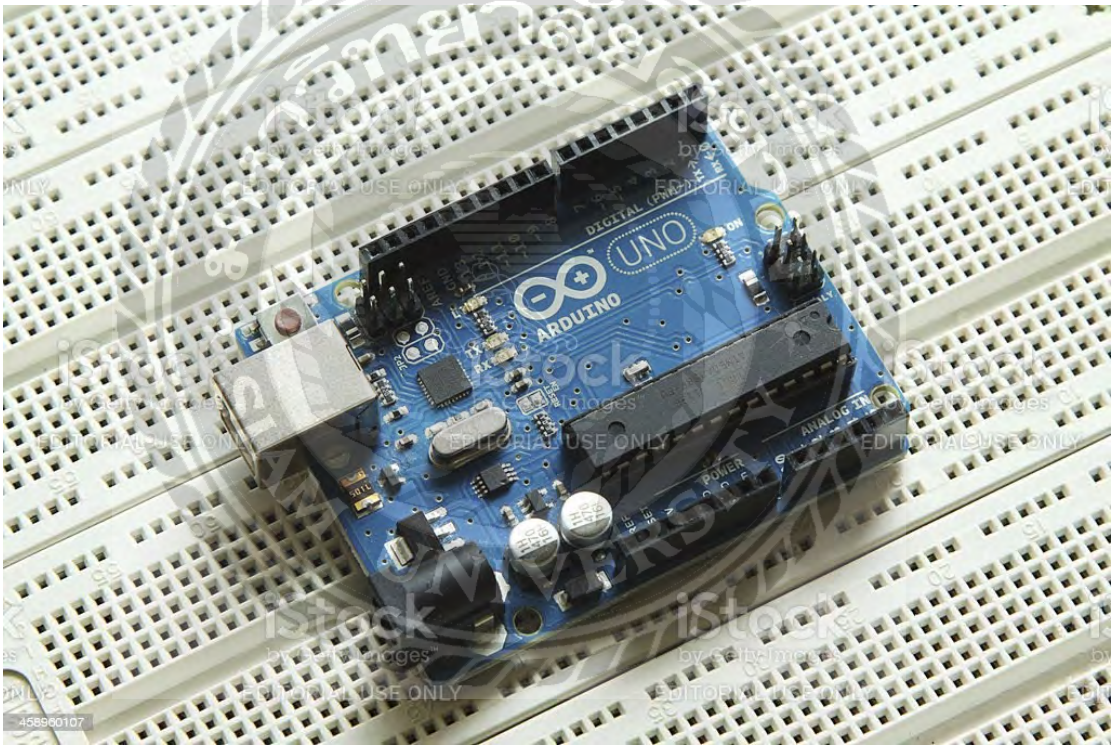
2.2.6 Hidden Camera

เป็นกล้องวงจรปิดที่มีขนาดเล็ก ถูกออกแบบมาให้สำหรับการซ่อนไว้ตามมุมอับ หรือสิ่งของต่างๆ บางรุ่นสามารถบันทึกเสียงได้เพื่อนำไปใช้งานสำหรับการดักฟังและแอบถ่าย

รูปที่ 2.7 ตัวอย่างของกล้องวงจรปิดประเภท Hidden Camera

2.3 Arduino

Arduino เป็นฮาร์ดแวร์แบบ Open-Source ที่ถูกออกแบบมสำหรับการสร้างและประยุกต์ใช้กับสิ่งต่างๆได้ โดยตัวบอร์ดของ Arduino จะเป็นแบบ บอร์ดไมโครคอนโทรลเลอร์ประเภทเดี่ยวและบอร์ดแบบไมโครคอนโทรลเลอร์ธรรมดา โดยตัวบอร์ดทั้งสองประเภทจะประกอบไปด้วยช่องสำหรับการเชื่อมต่อแบบ Digital และแบบ Analog ประเภท Input และ Output (I/O) เพื่อไว้สำหรับเชื่อมต่อกับส่วนเสริมที่ผู้ใช้ต้องการใช้งานหรือใช้สำหรับเชื่อมต่อกับตัวกลางอย่าง Breadboard เป็นต้น และจะประกอบด้วยชุดสื่อสารแบบ Serial แต่บางตัวบอร์ดจะทำการเชื่อมต่อสื่อสารและส่งตัวโปรแกรมผ่านทาง Universal Serial Bus (USB) ในบางรุ่น



รูปที่ 2.8 ตัวอย่างของบอร์ดไมโครคอนโทรลเลอร์

โปรเจค Arduino ถูกสร้างขึ้นมาในปี ค.ศ. 2005 ให้เป็นเครื่องหมายเพื่อใช้งานการศึกษาของนักเรียนในสถาบัน Interaction Design Institute Ivrea ของประเทศอิตาลี โดยมีจุดประสงค์ให้เป็นเครื่องหมายที่มีค่าใช้จ่ายต่ำที่สุดเท่าที่จะทำได้แต่สะดวกต่อการใช้งานและง่ายต่อการเริ่มเรียนรู้จนกระทั่งฝึกจนชำนาญ โดยตัวบอร์ดในยุคเริ่มแรกถูกออกแบบมาให้สามารถใช้งานกับโมดูลเซนเซอร์ได้เพียงไม่กี่อย่างเท่านั้น โดยในภายหลังได้มีการนำมาพัฒนาและต่อยอดต่อจน Arduino มีความสามารถที่หลากหลายและกว้างขวางขึ้นมากพอที่จะสามารถนำไปประยุกต์ใช้กับการสร้างหุ่นยนต์บางประเภทได้

บอร์ดไมโครคอนโทรลเลอร์สามารถรับโปรแกรมที่ถูกเขียนด้วยภาษาคอมพิวเตอร์ C และภาษา C++ ได้เพียงเท่านั้น เนื่องจากตัวบอร์ดได้ใช้งาน API แบบพื้นฐานหรือในอีกชื่อคือ “Arduino Programming Language ” ที่ถูกพัฒนาโดยมี Processing Language เป็นต้นแบบ และบอร์ดไมโครคอนโทรลเลอร์นั้นมีตัวเครื่องมือสำหรับการพัฒนาและปรับแต่งบอร์ดของตนเองในชื่อ Arduino IDE (Arduino Integrated Development Environment)

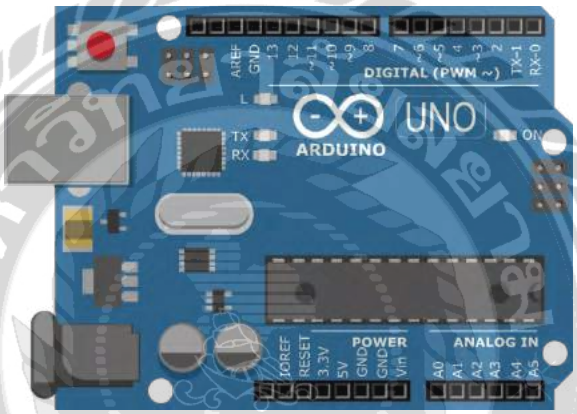
จุดเด่นของ Arduino Board คือมีความง่ายในการต่อกับอุปกรณ์เสริมต่างๆที่ผู้ใช้งานสามารถต่อวงจรไฟฟ้าได้อย่างง่าย โดยเชื่อมต่อที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริมประเภทต่างๆได้ การใช้งานบอร์ด Arduino ในปัจจุบัน เป็นระบบเปิดที่สมบูรณ์ที่ช่วยให้ผู้ใช้สามารถสร้างได้อย่างอิสระปรับให้เข้ากับความต้องการเฉพาะ ซอฟต์แวร์ก็เป็นระบบเปิดและมีการเติบโตผ่านการมีส่วนร่วมของผู้ใช้ทั่วโลก

รูปที่ 2.9 สัญลักษณ์ของ Arduino

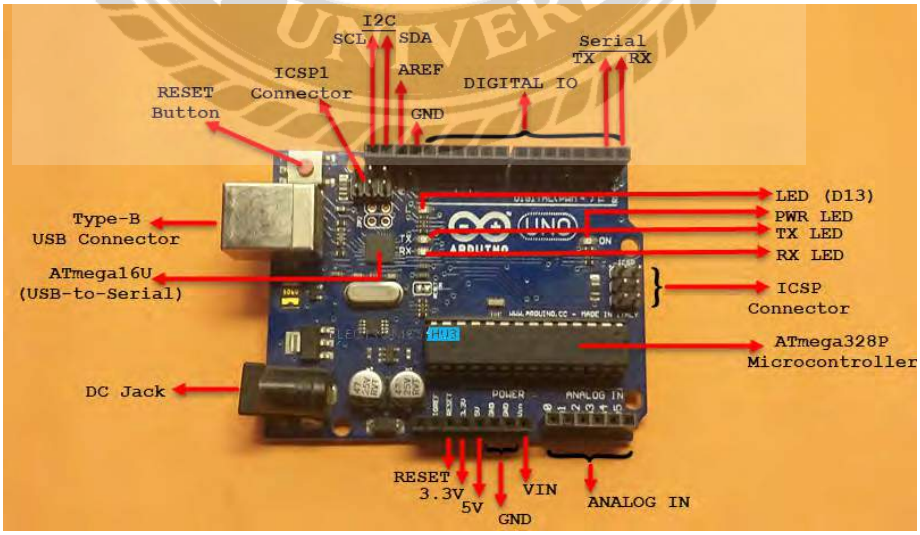
2.4 ประเภทบอร์ดของ Arduino ที่ใช้

2.4.1 Arduino Uno R3

บอร์ด Arduino Uno R3 เป็นบอร์ดที่ถูกพัฒนาโดยทีม Arduino.cc และถูกปล่อยออกมาในปี ค.ศ. 2010 โดยตัวบอร์ดจะประกอบไปด้วยขาเชื่อมต่อแบบดิจิทัลและอนาล็อคทั้งหมด 14 ขาโดยสามารถทำหน้าที่เป็นช่อง Input และ Output ได้ทั้งคู่ สามารถเขียนโปรแกรมลงไปได้โดยใช้งานโปรแกรม Arduino IDE ผ่านทางสายเชื่อมต่อ USB ประเภท B โดยตัวบอร์ดจำเป็นต้องมีแหล่งพลังงานเชื่อมต่อไม่ว่าจะผ่านทางสายเชื่อมต่อ USB หรือผ่านทางตัวแปลงกระแสไฟฟ้าก็ได้



รูปที่ 2.10 ตัวอย่างของบอร์ด Arduino Uno R3



รูปที่ 2.11 อธิบายแต่ละส่วนของบอร์ด Arduino

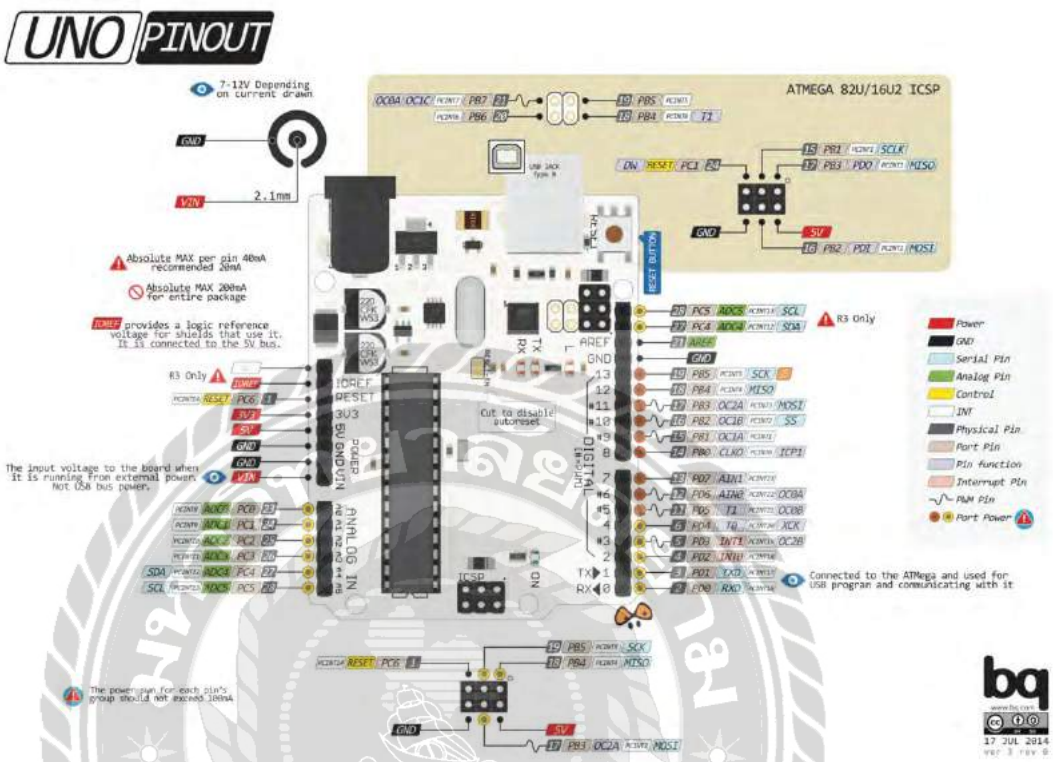
ข้อมูลทางเชิงเทคนิคของบอร์ด

- บอร์ดเป็นแบบประเภทไมโครคอนโทรลเลอร์ (MCU)
- ใช้ Microchip ประเภท “ATmega328P”
- มี Clock Speed ที่ 16 MHz
- ต้องการแหล่งพลังงานอย่างน้อย 5 โวลต์เพื่อทำงาน
- มีช่องขาเชื่อมต่อดิจิทัลทั้งหมด 14 ขา ทำงานทั้งแบบ Input และ Output
- มีช่องขาเชื่อมต่ออนาล็อคทั้งหมด 6 ขา ทำงานทั้งแบบ Input และ Output
- ไฟฟ้ากระแสตรงที่ส่งออกแก่ช่อง Input/Output แต่ละช่องเท่ากับ 20 mA
- ไฟฟ้ากระแสตรงสำหรับช่อง 3.3V เท่ากับ 50 mA
- มีน้ำหนักทั้งหมด 25 กรัม
- ราคาตามตลาดอยู่ตั้งแต่ 120 – 170 บาท
- แหล่งพลังงาน :
    - สายเชื่อมต่อ USB : สายเชื่อมต่อ USB สามารถส่งกระแสไฟฟ้าให้กับบอร์ด Arduino ได้ในช่วง 4.75 – 5.25 โวลต์ โดยภายในบอร์ดจะมีช่องสำหรับสาย USB ประเภท B
    - ช่องเสียบสายเชื่อมต่อแจ็คหัวกลม 5.5 ม : บอร์ด Arduino สามารถรองรับกระแสไฟฟ้าได้ในช่วง 6 – 20 โวลต์

ส่วนเสริมที่สร้างมาพร้อมกับบอร์ด

- **หลอดไฟ LED** : โดยจะมีหลอดไฟ LED ถูกสร้างมาแบบ built-in ในบริเวณของช่องเชื่อมต่อดิจิทัลที่ 13
- **VIN:** เป็นช่อง Input ของกระแสไฟฟ้าหน่วยโวลต์เพื่อส่งต่อเข้าไปยังบอร์ด Arduino โดยตรง
- **5V:** เป็นช่องสำหรับส่งกระแสไฟฟ้าออกเป็นจำนวน 5 โวลต์จากตัวบอร์ด Arduino เอง มีไว้สำหรับส่งกระแสไฟฟ้าให้กับโมดูลส่วนเสริมตัวอื่นๆ
- **3V3:** เป็นช่องสำหรับส่งกระแสไฟฟ้าออกเป็นจำนวน 3.3 โวลต์จากตัวบอร์ด Arduino เอง มีไว้สำหรับส่งกระแสไฟฟ้าให้กับโมดูลส่วนเสริมตัวอื่นๆ
- **GND:** เป็นช่องสำหรับต่อสาย Ground

- **Reset หรือ Rst:** เป็นปุ่มกดที่ถูกสร้างแบบ built-in ติดมากับบอร์ดตั้งแต่แรก มีหน้าที่เริ่มต้นการทำงานของตัวโปรแกรมที่ถูกเขียนไว้ในบอร์ดใหม่ทั้งหมดตั้งแต่แรก



รูปที่ 2.12 ตัวอย่างโครงสร้างของบอร์ด Arduino Uno R3

2.4.2 Arduino NodeMCU 32S

บอร์ด NodeMCU เป็นบอร์ดสำหรับการสร้างแพลตฟอร์ม IoT แบบ Open source โดยตัวบอร์ดนั้นมีจุดประสงค์หลักเพื่อเป็นเหมือนตัวทดลองสำหรับการพัฒนาโปรแกรมหรือ IoT ของผู้พัฒนา โดย Firmware ของบอร์ด NodeMCU จะใช้งานภาษา Lua สำหรับประมวลผล และมีตัว ESP32 ทำหน้าที่เป็นระบบประมวลผลหลัก โดยที่ ESP32 นี้จะประกอบไปด้วยชิฟ Wi-Fi และ Bluetooth ที่ถูกสร้างไว้แบบ built-in แล้ว

รูปที่ 2.13 ตัวอย่างของบอร์ด NodeMCU

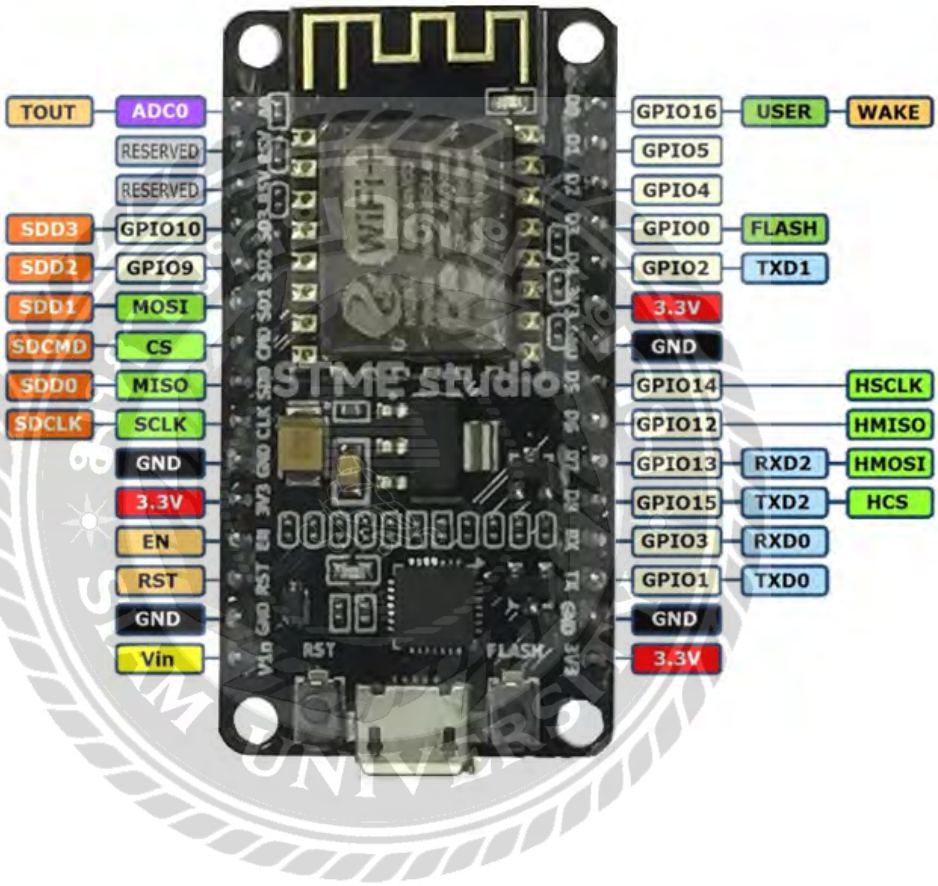
ข้อมูลทางเชิงเทคนิคของบอร์ด

- ตัวบอร์ดเป็นแบบประเภทบอร์ดไมโครคอนโทรลเลอร์เดี่ยว
- มีช่องสำหรับการเชื่อมต่อผ่านทาง GPIO ทั้งหมด 12 ช่อง โดยทำหน้าที่เป็นทั้ง I/O
- ตัวบอร์ดใช้งาน ESP32 หรือ ESP8266 สำหรับระบบประมวลผล
- ตัวบอร์ดมีหน่วยความจำอยู่ที่ 128 Kbytes
- ตัวบอร์ดมีพื้นที่ความจุอยู่ที่ 4 Mbytes
- ตัวบอร์ดสามารถรับแหล่งพลังงานผ่านทางสายเชื่อมต่อ USB ได้
- ตัวบอร์ดมีราคาอยู่ที่ประมาณ 35 – 90 บาท

ส่วนเสริมที่สร้างมาพร้อมกับบอร์ด

- **VIN:** เป็นช่อง Input ของกระแสไฟฟ้าหน่วยโวลต์เพื่อส่งต่อเข้าไปยังบอร์ด Arduino โดยตรง
- **5V:** เป็นช่องสำหรับส่งกระแสไฟฟ้าออกเป็นจำนวน 5 โวลต์จากตัวบอร์ด Arduino เอง มีไว้สำหรับส่งกระแสไฟฟ้าให้กับโมดูลส่วนเสริมตัวอื่นๆ

- **3V3:** เป็นช่องสำหรับส่งกระแสไฟฟ้าออกเป็นจำนวน 3.3 โวลต์จากตัวบอร์ด Arduino เอง มีไว้สำหรับส่งกระแสไฟฟ้าให้กับโมดูลส่วนเสริมตัวอื่นๆ
- **GND:** เป็นช่องสำหรับต่อสาย Ground
- **Reset หรือ Rst:** เป็นปุ่มกดที่ถูกสร้างแบบ built-in ติดมากับบอร์ดตั้งแต่แรก มีหน้าที่เริ่มต้นการทำงานของตัวโปรแกรมที่ถูกเขียนไว้ในบอร์ดใหม่ทั้งหมดตั้งแต่แรก



รูปที่ 2.14 ตัวอย่างโครงสร้างของบอร์ด NodeMCU

ตัวบอร์ด NodeMCU นั้นจะมีการใช้ชื่อเรียกของช่องเชื่อมต่อแตกต่างจากบอร์ดตัวอื่น โดยบอร์ด NodeMCU จะใช้ชื่อว่า General Purpose Input/Output หรือ GPIO สำหรับชื่อในแต่ละช่องเชื่อมต่อ สามารถแยกออกมาเป็นแต่ละช่องได้ดังนี้

ตารางที่ 2.1 การจัดแจงช่องเชื่อมต่อของ NodeMCU

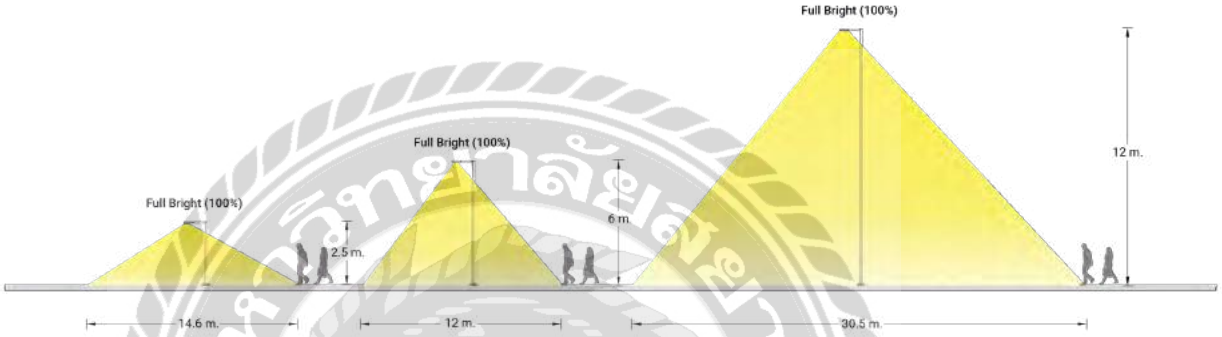
ช่อง I/O	ขาของ NodeMCU
0	GPIO16
1	GPIO5
2	GPIO4
3	GPIO0
4	GPIO2
5	GPIO14
6	GPIO12
7	GPIO13
8	GPIO15
9	GPIO3
10	GPIO1
11	GPIO9
12	GPIO10

ตารางที่ 2.2 การเปรียบเทียบแต่ละรุ่นของบอร์ด Arduino Uno

Board Name & Part#	Board Size Group	Board Commun-ication	MCU Part# & Pins	MCU I/O Voltage	MCU Core	MCU Clock	MCU Flash	MCU SRAM	MCU EEPROM	MCU USART & UART	MCU SPI	MCU I²C	MCU Other Bus Peripherals	MCU Timers 32/24/16/8 /WD/RT/RC	MCU ADC / DAC	MCU Engines
Uno R3,[12] A000066,[13] Uno R3 SMD,[14] A000073[15]	Uno	USB-B	ATmega328P[16] 28 pin DIP 32 pin SMD	5V (1.8-5.5V)	8bit AVR	16 MHz*	32 KB	2 KB	1 KB	1, 0	1	1	None	0, 0, 1, 2, WD	10bit, None	None
Uno WiFi R2,[17] ABX00021[18]	Uno	USB-B, WiFi, Bluetooth	ATmega4809,[19] 48 pin	5V (1.8-5.5V)	8bit AVR	16 MHz*	48 KB	6 KB	0.25 KB	4, 0	1	1	None	0, 0, 5, 0, WD, RT	10bit, None	None
Leonardo,[20] A000057[21]	Uno	USB-Micro-B	ATmega32U4,[22] 44 pin	5V (2.7-5.5V)	8bit AVR	16 MHz	32 KB	2.5 KB	1 KB	1, 0	1	1	USB-FS	0, 0, 2, 2*, WD	10bit, None	None

2.5 เซนเซอร์ตรวจจับการเคลื่อนไหว

เซนเซอร์ตรวจจับการเคลื่อนไหว หรืออีกชื่อคือ “เครื่องตรวจจับการเคลื่อนไหว” เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ออกแบบมาเพื่อใช้ในการตรวจจับความเคลื่อนไหวทางกายภาพของสิ่งมีชีวิต โดยจะนิยมนำมาใช้ระบบดูแลรักษาความปลอดภัยภายในบ้าน หรือในธุรกิจ เพื่อตรวจจับผู้บุคคลที่เข้ามาในสถานที่ ตัวอย่างของการใช้งานเช่น เสาไฟที่สามารถสว่างเองได้เมื่อมีคนเดินอยู่ภายในระยะที่กำหนด เป็นต้น



รูปที่ 2.15 ตัวอย่างการใช้งานของเซนเซอร์ตรวจจับการเคลื่อนไหว

ประเภทของเซนเซอร์ตรวจจับความเคลื่อนไหว

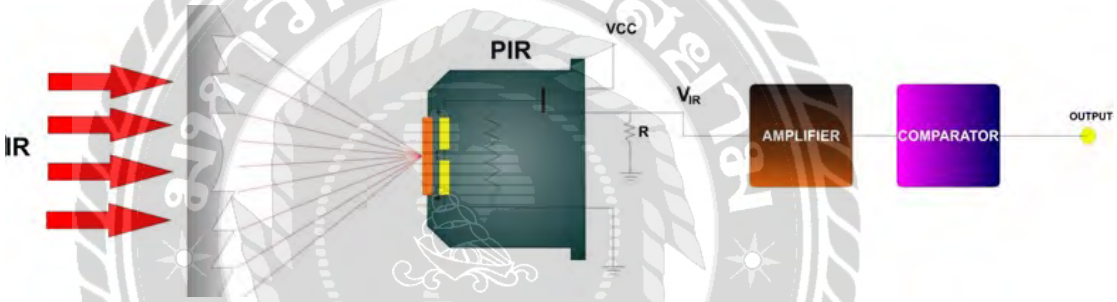
เซนเซอร์ตรวจจับความเคลื่อนไหวที่นิยมใช้ในปัจจุบัน จะแบ่งเป็น 3 ประเภทหลัก ๆ ดังนี้

1. **Passive Infrared Sensors (PIR)**: เป็นเซนเซอร์ที่ใช้แสงอินฟราเรดในการตรวจจับความร้อนภายในร่างกายของสิ่งมีชีวิตขณะที่เคลื่อนที่
2. **Ultrasonic**: เป็นเซนเซอร์ที่ตรวจจับความเคลื่อนไหวโดยการตรวจวัดการสะท้อนของคลื่น Ultrasonic ที่ปล่อยออกไป ในขณะที่มีวัตถุเคลื่อนที่
3. **Microwave**: มีหลักการทำงานคล้ายคลึงกับ Ultrasonic แต่จะปล่อยคลื่นไมโครเวฟออกมาแทน

2.5.1 PIR Motion Sensor Module

PIR Sensor นั้นย่อมาจาก Passive infra-red sensor หรือความหมายโดยตรงก็คือ เซนเซอร์ตรวจจับคลื่น Infrared อุปกรณ์ Sensor ชนิดหนึ่งที่ใช้ตรวจจับคลื่นรังสี Infrared ที่แพร่มาจาก มนุษย์หรือสัตว์ที่มีการเคลื่อนไหว ทำให้มีการนำเอา PIR มาประยุกต์ใช้งานกันเป็นอย่างมาก เพื่อตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต หรือตรวจจับการบุกรุกในงานรักษาความปลอดภัย

การทำงานของ PIR Sensor จะมีการตรวจจับรังสี Infrared อยู่ 2 ชุดด้วยกัน เมื่อมีสิ่งมีชีวิตที่มีความอบอุ่นในร่างกายเคลื่อนที่เข้ามาในโซน PIR จะสามารถตรวจจับคลื่นรังสี Infrared ที่แพร่ออกมา และจับเปลี่ยนคลื่นรังสี Infrared ให้กลายเป็นคลื่นกระแสไฟฟ้าแทน

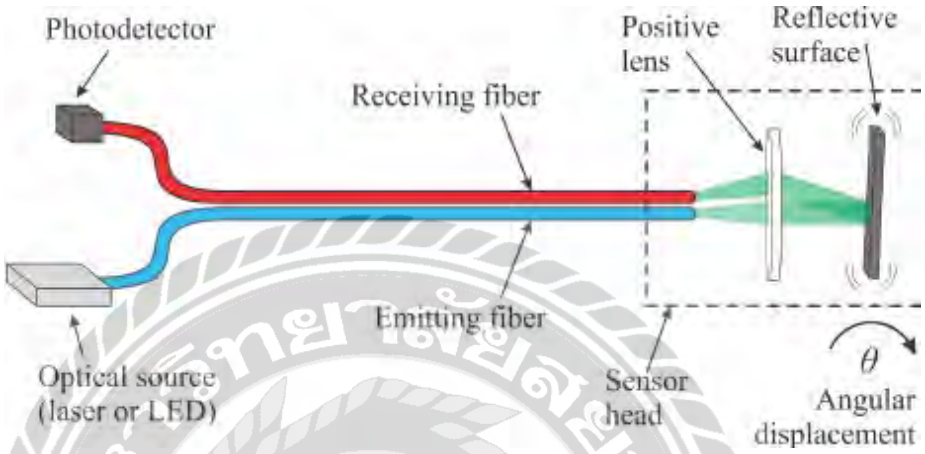


รูปที่ 2.16 หลักการทำงานของเซนเซอร์ PIR

2.5.2 เซนเซอร์วัดระยะทาง (Ultrasonic)

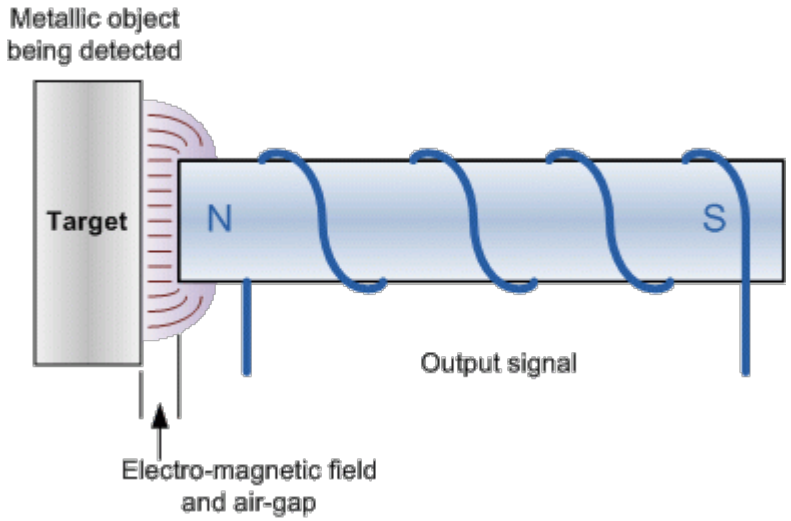
เซนเซอร์วัดระยะสามารถตรวจจับวัตถุได้โดยที่ไม่จำเป็นต้องสัมผัสกับวัตถุโดยตรง ทำให้ชิ้นงานที่ต้องการตรวจสอบไม่เกิดรอยหรือเกิดการชำรุดเสียหาย หลักการทำงานของเซนเซอร์วัดระยะทางจะใช้ลำแสงเป็นตัวส่ง ด้วยการยิงเลเซอร์ไปที่วัตถุและสะท้อนกลับมายังตัวรับของเซนเซอร์ และจะทำการประมวลผลค่าแสงที่สะท้อนกลับมาแล้วแสดงผลในรูปแบบตัวเลขผ่านหน้าจอแสดงผลแบบดิจิทัลหรือแบบอนาล็อคยัง โดยการเชื่อมต่อกับคอมพิวเตอร์ การจำแนกประเภทเซนเซอร์วัดระยะทางสามารถจำแนกได้ดังนี้

1. **Optical Displacement Sensor** ใช้ระบบวัดแบบสามเหลี่ยม (triangulation measurement system) โดยเซนเซอร์บางตัวใช้หลักการ PSD (Position Sensitive Detector) และ บางตัวก็ใช้เซนเซอร์รับภาพแบบ CCD และ CMOS เป็นตัวรับแสง



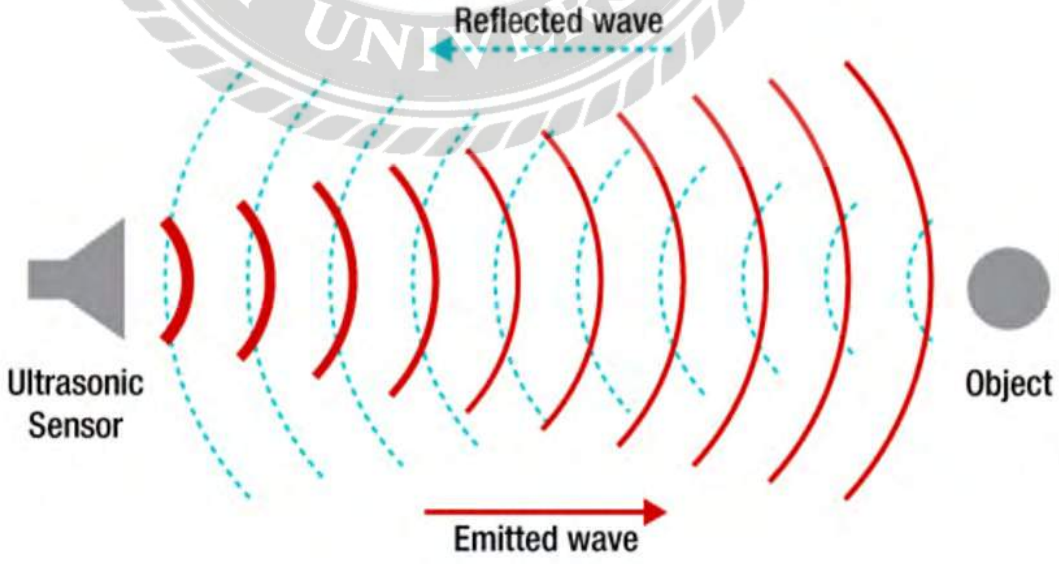
รูปที่ 2.17 หลักการทำงานของ Optical Displacement Sensor

2. **Linear Proximity Sensor** อาศัยหลักการทำงานจากสนามไฟฟ้าและสนามแม่เหล็ก บริเวณส่วนหัวของเซนเซอร์จะมีสนามแม่เหล็กซึ่งใช้ความถี่สูง โดยรับสัญญาณมาจากวงจรกำเนิดความถี่ เมื่อมีวัตถุหรือชิ้นงานที่เป็นโลหะ เข้ามาอยู่ในบริเวณที่สนามมาเหล็กสามารถส่งไปถึง จะทำให้เกิดการเปลี่ยนแปลงค่าความเหนี่ยวนำแล้วทำการส่งสัญญาณควบคุมออกมา สามารถตรวจจับวัตถุได้ทั้งในระยะใกล้และไกลได้อย่างรวดเร็วและแม่นยำ



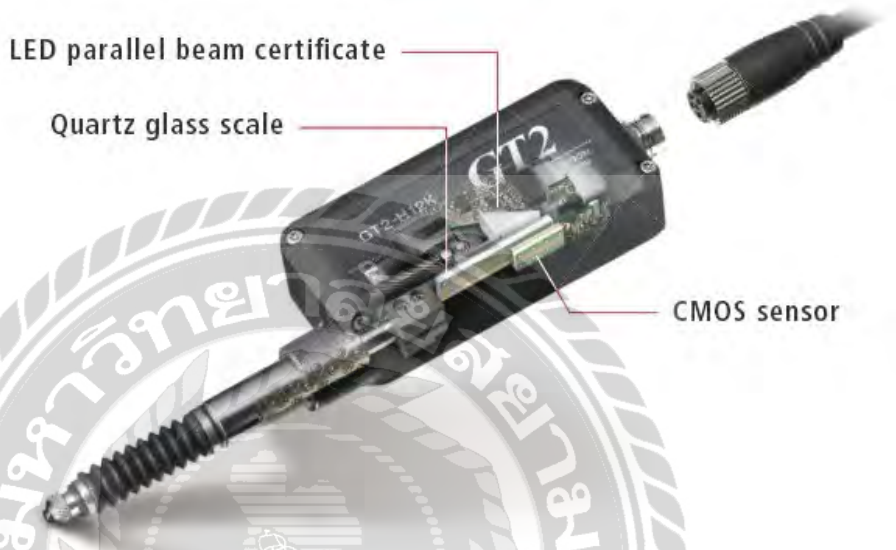
รูปที่ 2.18 หลักการทำงานของ Linear Proximity Sensor

3. **Ultrasonic Displacement Sensor** ใช้หลักการสะท้อนของคลื่นความถี่เสียงในการตรวจจับวัตถุต่างๆ โดยการส่งคลื่นความถี่เสียงในช่วง Ultrasound ไปยังวัตถุหรือชิ้นงาน แล้วรับคลื่นที่สะท้อนกลับจากวัตถุนั้นมาคำนวณหาค่าระยะทาง โดยคำนวณความสัมพันธ์ระหว่างเวลาในการเดินทางของคลื่น กับความเร็วของเสียง



รูปที่ 2.19 หลักการทำงานของ Ultrasonic Displacement Sensor

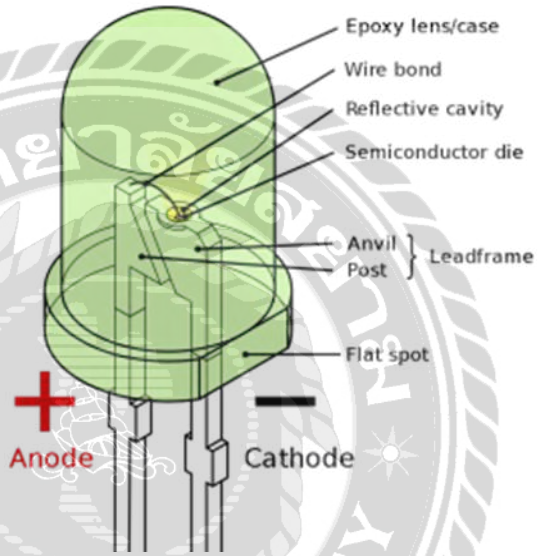
4. **Contact Displacement Sensor** วัดระยะผ่านการสัมผัสวัตถุโดยตรง จึงให้ความแม่นยำในการวัดที่เหนือกว่าเมื่อเปรียบกับเซนเซอร์แบบไร้สัมผัส แต่ก็มีระยะการใช้งานที่สั้นกว่า เซนเซอร์ชนิดนี้มีหลายรูปแบบให้เลือกใช้ขึ้นอยู่กับเงื่อนไขในการใช้งาน และวัตถุที่ต้องการวัด



รูปที่ 2.20 หลักการทำงานของ Contact Displacement Sensor

2.6 หลอดไฟ LED

หลอดไฟ LED (Light Emitting Diode) หรือไดโอดชนิดเปล่งแสง สามารถนำไปติดตั้งบนอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เพื่อใช้เป็นแหล่งกำเนิดแสง แสงที่เปล่งออกมาประกอบด้วยคลื่นความถี่เดียวและต่อเนื่องกัน ต่างจากแสงธรรมดาที่คนมองเห็น สามารถเปล่งแสงได้เมื่อได้รับกระแสไฟฟ้าเพียงเล็กน้อย หลอดไฟ LED ทำมาจากสารกึ่งตัวนำ (Semiconductor) 2 ชนิดมาวงติดกัน ประกอบด้วย ส่วนที่เป็นขั้วบวก (Positive Type) และส่วนที่เป็นขั้วลบ (Negative Type)



รูปที่ 2.21 โครงสร้างภายในของ LED

โดยส่วนที่เชื่อมระหว่างสารกึ่งตัวนำขั้วบวกและขั้วลบ เรียกว่า “P-N Junction” เมื่อกระแสไฟฟ้าไหลผ่านสารกึ่งตัวนำทั้ง 2 ชนิด อิเล็กตรอนที่อยู่ฝั่งขั้วลบจะวิ่งไปที่ฝั่งขั้วบวก และจะเปล่งแสงสว่างออกมา รูปแบบของ LED ปัจจุบันจะแบ่งตามลักษณะของ Packet แบ่งได้เป็น 2 ประเภทคือ

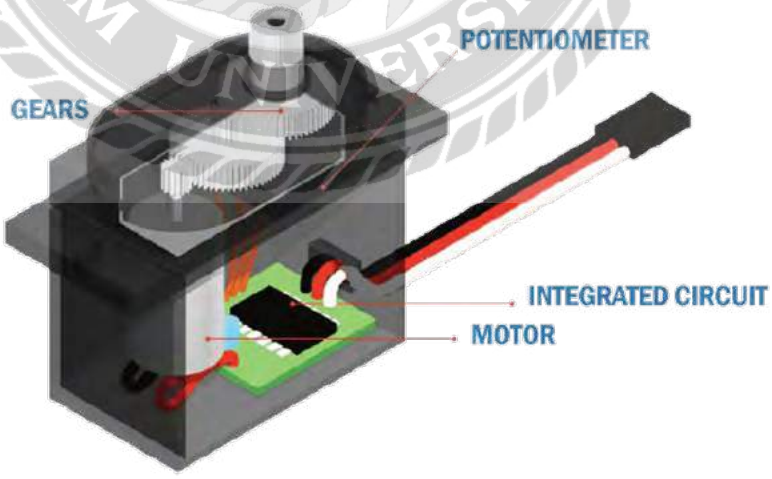
1. **Lamp Type** เป็น LED ชนิดที่พบกันอยู่ทั่วไปมีขายื่นออกมาจากตัว Epoxy 2 ขาหรือมากกว่า โดยมีขนาดเส้นผ่านศูนย์กลางตั้งแต่ 3 mm. ขึ้นไป ออกแบบให้ขับกระแสไฟฟ้าได้ไม่เกิน 150 mA

2. **Surface Mount Type (SMD)** มีลักษณะ packet เป็นตัวบางๆเวลาประกอบต้องใช้เครื่องมือชนิดพิเศษ ขนาดการขับกระแสตั้งแต่ 20mA - 1A สำหรับ LED แบบ SMD ถ้าขับกระแสตั้งแต่ 300 mA ขึ้นไป จะเรียกว่า Power LED ถ้าขับกระแสตั้งแต่ 300 mA ขึ้นไป จะเรียกว่า Power LED การใช้งานส่วนใหญ่จะใช้ภายในเนื่องจากสารเคลือบหน้าหลอด LED ส่วนใหญ่จะเป็นซิลิโคน ซึ่งละอองน้ำหรือความชื้นสามารถซึมผ่านได้

2.7 Servo Motor

เซอร์โวมอเตอร์ (Servo Motor) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีการควบคุมระบบขับเคลื่อนต่างๆ ไม่ว่าจะเป็นระยะ ความเร็ว แรงบิด มุมการหมุน แรงตำแหน่ง โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) ระบบควบคุมของมอเตอร์จะประกอบด้วยไฟฟ้าคอนโทรลและเครื่องกล ระบบควบคุมจะมีการวัดค่าเอาต์พุตของระบบ แล้วนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบมีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต

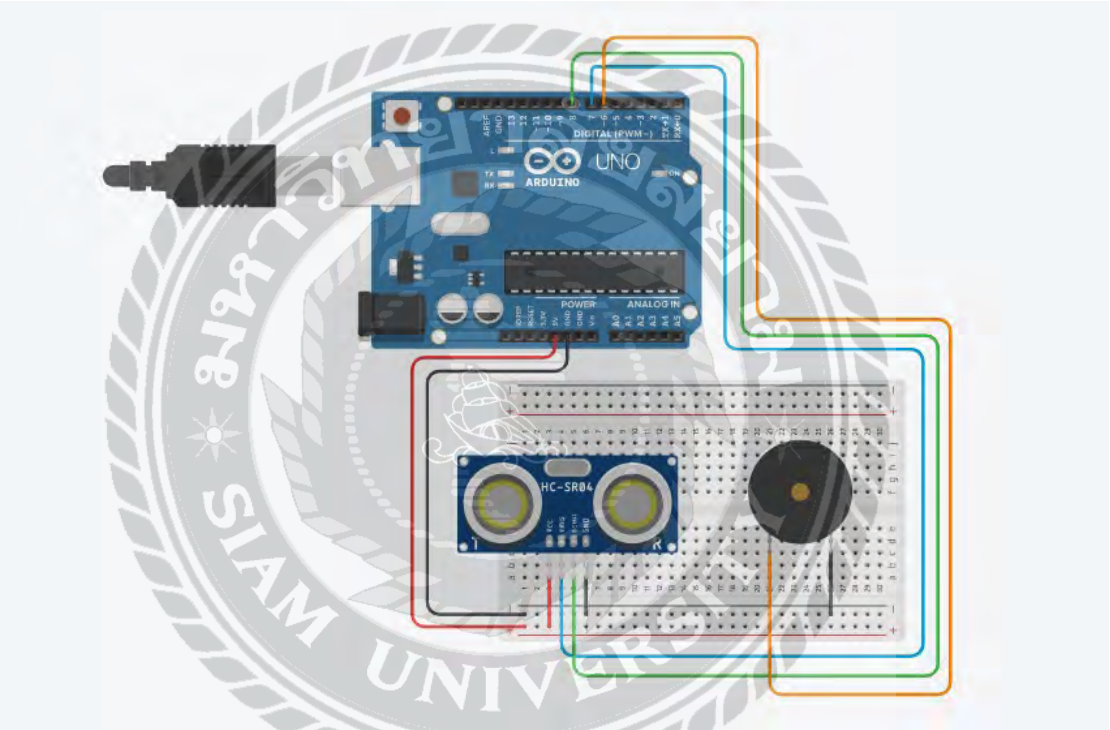
เซอร์โวมอเตอร์ในปัจจุบันมีหลากหลายรูปแบบให้เลือก ทั้งแบบกระแสตรง (DC Motor) และแบบการะแสสลับ (AC Motor) ด้วยคุณสมบัติเด่นของ Servo Motor คือสามารถให้แรงบิดที่สูง (มีหน่วยเป็นนิวตันเมตร) ทำงานได้รวดเร็ว สามารถเคลื่อนที่ได้อย่างแม่นยำ และทำงานได้เงียบ



รูปที่ 2.22 โครงสร้างภายในของ Servo Motor

2.8 Tinkercad

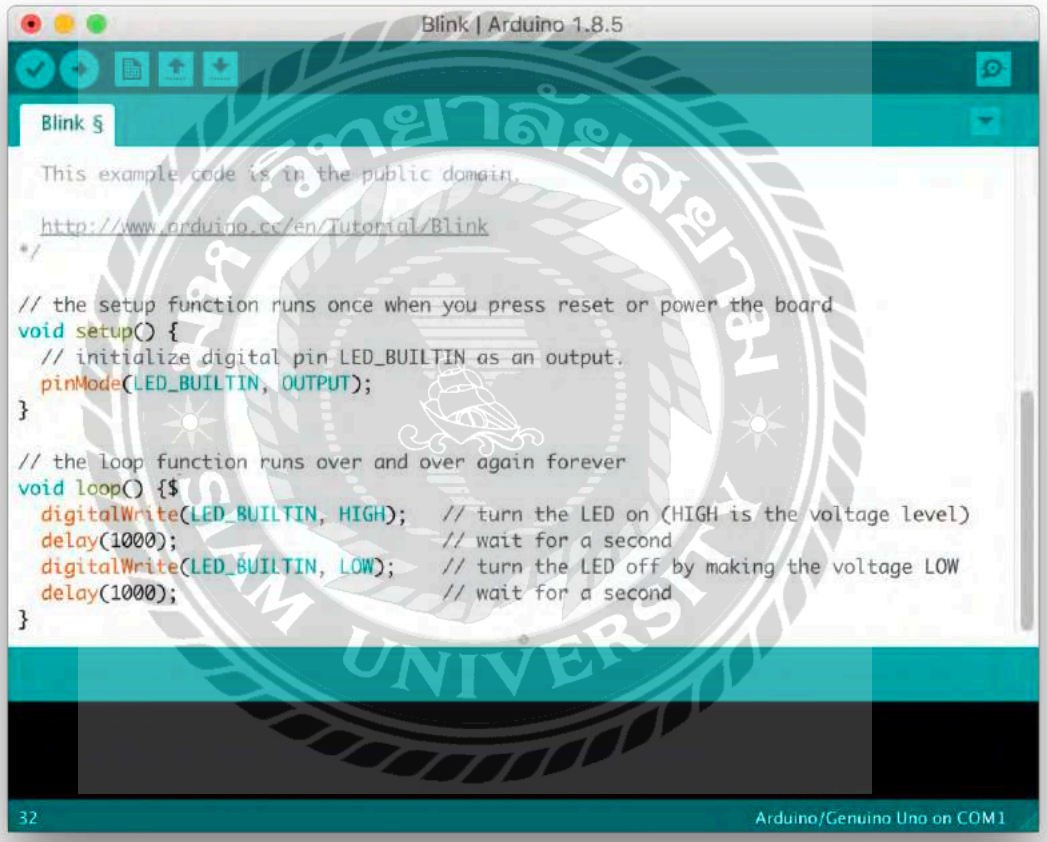
เป็นเว็บไซตำหรับกาสร้างโมเดล 3 มิติและการจำลองที่ผู้ใช้งานสามารถเข้าไปใช้งานได้โดยไม่มีค่าใช้จ่าย โดยภายในเว็บไซต์นั้นจะมีโหมดที่ชื่อว่า “Circuits” เป็นโมหดที่สร้างโมเดลจำลองของอุปกรณ์อิเล็กทรอนิกส์ขึ้นมาโดยทำการอิงมาจาก Arduino Uno ที่เป็นบอร์ดไมโครคอนโทรลเลอร์ โดยผู้ใช้งานสามารถจำลองการสร้างและเชื่อมต่อของบอร์ด Arduino Uno ผ่านทางเว็บไซต์ได้ก่อนที่จะทำการเขียนโปรแกรมลงไปภายในช่องที่กำหนดบนเว็บไซต์เพื่อดูผลทดสอบที่ตามมา



รูปที่ 2.23 ตัวอย่างการทดลองใช้งาน Tinkercad’s Circuits

2.9 Arduino IDE

เป็นซอฟต์แวร์สำหรับการปรับแต่งและแก้ไขบอร์ด Arduino โดยการเขียนโปรแกรมลงไปในบอร์ดขณะที่เชื่อมต่อโปรแกรมและบอร์ดผ่านทางสายเชื่อมต่อ USB โดยซอฟต์แวร์นั้นรองรับภาษา C และ C++ ในการเขียนและแก้ไขบอร์ด Arduino ผู้ใช้งานสามารถเลือกที่จะทำการเขียนโปรแกรมลงไปในบอร์ด Arduino หรือเลือกที่จะใช้งานตัวโปรแกรมสำเร็จรูปที่มีภายในซอฟต์แวร์นั้น ตัวซอฟต์แวร์ยังมีตัวช่วยเสริมเป็นคลัง Library สำหรับโมดูลและส่วนเสริมอื่นๆที่สามารถใช้งานกับบอร์ด Arduino ให้ผู้ใช้งานสามารถเลือกนำมาใช้งานได้อย่างกว้างขวาง



รูปที่ 2.24 ตัวอย่างของซอฟต์แวร์ Arduino IDE

2.10 ภาษา C++

ภาษา C++ เป็นภาษาสำหรับการเขียนโปรแกรมคอมพิวเตอร์ที่มีอเนกประสงค์เป็นโครงสร้างภาษาที่มีการจัดข้อมูลแบบสติก (Statically typed) และสนับสนุนรูปแบบการเขียนโปรแกรมได้อย่างหลากหลายรูปแบบ (Multi-Paradigm Language) ภาษา C++ ถูกพัฒนาโดยนายเบียเนอ สเดราสดร็อบ (Bjarne Stroustrup) จากบริษัทเบลล์แล็บส์ (Bell Labs) โดยมีความต้องการที่จะพัฒนาภาษาซีดั้งเดิมที่มีอยู่แล้วให้มีความสามารถมากกว่าแค่การสร้างคลาส จากนั้นจึงได้เพิ่มคุณสมบัติต่างๆตามมาภายหลัง

ภาษา C++ ถูกออกแบบมาสำหรับการทำงานภายใต้สิ่งแวดล้อมในระบบปฏิบัติการ UNIX เพื่อให้ผู้เขียนโปรแกรมได้ใช้งานอย่างมีประสิทธิภาพมากขึ้น ซึ่งมีประสิทธิภาพและความยืดหยุ่นสามารถพัฒนาได้ในหลายๆแพลตฟอร์ม ภาษาซีพลัสพลัสนี้จะมีการทำงานที่เร็วเทียบเท่าภาษาซีและยังสามารถดำเนินการกับฮาร์ดแวร์ได้ ภาษาบางภาษาอาจจะไม่สนับสนุนคุณลักษณะนี้

รูปที่ 2.25 เครื่องหมายของภาษาคอมพิวเตอร์ C ++

2.11 LINE

เป็นโปรแกรมแชทที่สามารถใช้งานได้ทั้งโทรศัพท์มือถือที่มีระบบปฏิบัติการ IOS, Android, Windows Phone ล่าสุดสามารถใช้งานได้บนคอมพิวเตอร์ PC และ Mac ได้แล้ว ด้วยความที่มีฟังก์ชันมากมาย สามารถแชท ส่งรูป ส่งไอคอน ส่ง Sticker ตั้งค่าคุยกันเป็นกลุ่ม ฯลฯ ทำให้มีผู้ใช้งานแอพนี้เป็นจำนวนมาก นอกจากนี้ LINE ยังมีโปรแกรมเสริม ทั้ง LINE Camera ที่ถ่ายภาพฟรี พร้อมกรอบกว่า 100 แบบ และแสตมป์แต่งภาพมากกว่า 600 แบบเก๋ๆ โดยจุดเด่นของแอพพลิเคชั่นนี้คือ การตกแต่งภาพหลากหลายรวมไปถึงการถ่ายภาพผ่านฟิลเตอร์ถึง 14 แบบ ที่ช่วยปรับแต่งภาพและรายละเอียดให้ภาพดูดียิ่งขึ้น พร้อมด้วยพู่กันกว่า 156 ชนิด เพื่อให้ผู้ใช้ได้แต่งแต้มด้วยแสตมป์และเลือกแบบตัวอักษรต่างๆ พิมพ์ข้อความลงบนภาพตามสไตล์ของตัวเองและสามารถแชร์ภาพได้โดยตรง

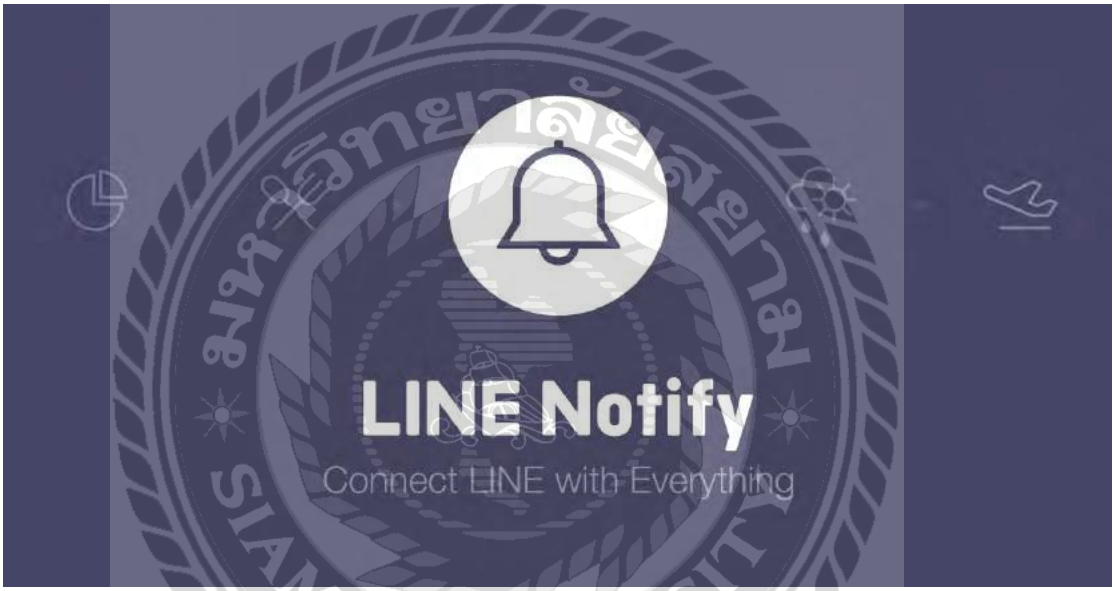
สิ่งที่โดดเด่นของไลน์

1. สามารถเพิ่มกลุ่มสนทนาหรือเชิญเพื่อนได้ถึง100คน
2. ออกแบบให้สามารถโทร.หากันฟรีแบบ1ต่อ1
3. พัฒนาคุณภาพของการโทรให้ดียิ่งขึ้น โดยตัดเสียงรบกวนและเสียงแทรกจากบริเวณรอบๆ ทำให้ผู้ใช้สามารถพูดคุย
4. ส่งวิดีโอและข้อความเสียงฟรี

รูปที่ 2.26 ไอคอนของแอพลิเคชัน LINE

2.12 LINE Notify

เป็นระบบบริการการแจ้งเตือนที่ขึ้นตรงกับแอพพลิเคชั่น LINE เพื่อไว้สำหรับให้ผู้พัฒนาที่ต้องการพัฒนาเว็บไซต์หรือพัฒนาในทางด้านอื่นที่จำเป็นต้องใช้งานการแจ้งเตือนของ LINE สามารถพัฒนาได้อย่างง่ายดายมากขึ้น โดยตัว LINE Notify จะทำการให้เลือกกลุ่มเป้าหมายหรือในอีกความหมายหนึ่งคือระบบช่องแชทภายในแอพพลิเคชั่นเองได้สูงสุดเพียง 5 รายการเท่านั้น หลังจากที่ทำการเลือกกลุ่มเป้าหมายแล้ว ทางเว็บไซต์จะให้ Token หรือรหัสลับกลับมาให้กับผู้ใช้งานสามารถนำไปต่อยอดและใช้งานได้ต่อไป



รูปที่ 2.27 ตัวอย่างภาพ LINE Notify

ตัวอย่างการใช้งาน LINE Notify

- สามารถนำไปใช้งานกับ Web Service ต่างๆในการสร้างหรือพัฒนา Web Application ได้
- สามารถนำไปประยุกต์ใช้งานกับ ESP8266 หรือ ESP32 ในการส่งข้อความหาผู้ใช้งานได้
- สามารถนำไปประยุกต์ใช้งานกับ Dialog flow ในการสร้าง LINE Application ได้
- สามารถนำมาใช้แจ้งเตือนเมื่อนำไปเชื่อมต่อกับอุปกรณ์ IoT ที่รองรับภายในบ้านได้

บทที่ 3

การวิเคราะห์และการออกแบบ

ในบทความนี้จะเป็นการวิเคราะห์และการออกแบบเกี่ยวกับตัวฐานของกล้องวงจรปิด โดยจะอธิบายโครงสร้างของฐานกล้องวงจรปิด, ระบบความคุมของบอร์ด Arduino, การทำงานของเซนเซอร์แต่ละตัว ดังต่อไปนี้

3.1 การออกแบบระบบ

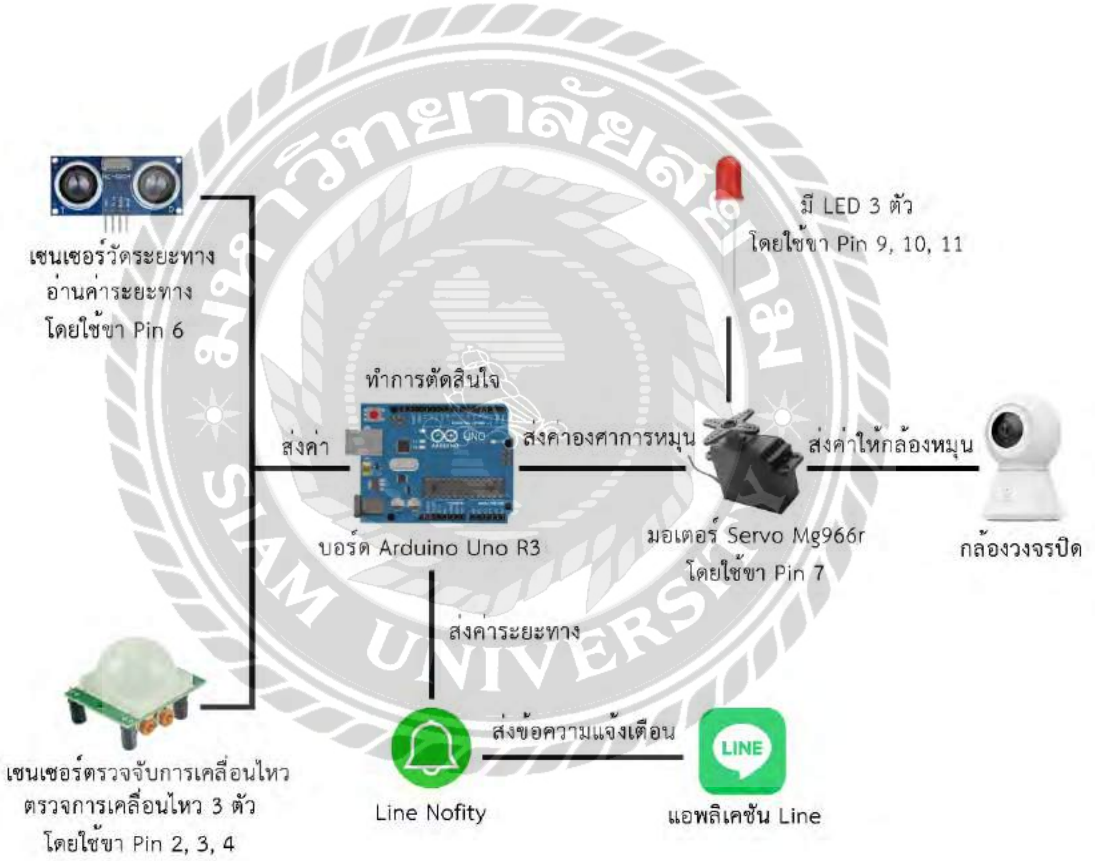
ในโครงงานนี้ทำการสร้างระบบตรวจจับการเคลื่อนไหวโดยใช้ PIR Motion Sensor Module ที่เป็นเซนเซอร์ตรวจจับการเคลื่อนไหวของบอร์ด Arduino โดยหลังจากที่ PIR Motion Sensor สามารถตรวจจับสิ่งเคลื่อนไหวโดยอาศัยหลักการตรวจจับความร้อนกับวัตถุที่อยู่ข้างหน้าเซนเซอร์ได้แล้ว จะสั่งให้ SG90 Servo Motor หมุนไปในทิศทางที่สิ่งนั้นอยู่ โดยในโครงงานนี้ผู้จัดทำได้เลือกใช้ SG90 Servo Motor ที่สามารถหมุนได้เพียง 180 องศาเนื่องจากภายในตัวฐานนั้นมีสายไฟเชื่อมต่อระหว่าง Breadboard กับตัวโมดูลแต่ละตัวอยู่มากเกินไป จึงไม่สามารถทำให้หมุนถึง 360 องศา

3.1.1 การออกแบบระบบฮาร์ดแวร์

ผู้จัดทำได้เลือกใช้ บอร์ด Arduino Uno R3 เนื่องจากเป็นบอร์ดที่สามารถรองรับ Library ของ SG90 Servo Motor และ Line Notify ได้ โดยผู้จัดทำได้นำบอร์ด Arduino Uno R3 ไปติดไว้กับตัวฐานของกล้องแล้วจึงเชื่อมตัวบอร์ดกับโมดูลแต่ละตัวผ่านตัวกลางอย่าง Breadboard แล้วจึงนำตัว Breadboard แต่ละแผ่นไปจัดวางไว้ภายในตัวฐานอย่างเป็นระเบียบโดยไม่ให้สายเชื่อมต่อแต่ละตัวพันกันจนดึงสายตัวอื่นออกมา หลังจากที่จัดวางภายในตัวฐานแล้ว ก็ทำการเจาะรูภายนอกของตัวฐานแต่ละจุดเพื่อนำโมดูลแต่ละตัวไปติดไว้ในจุดต่างๆ

3.2 ภาพรวมการทำงานของโครงงาน

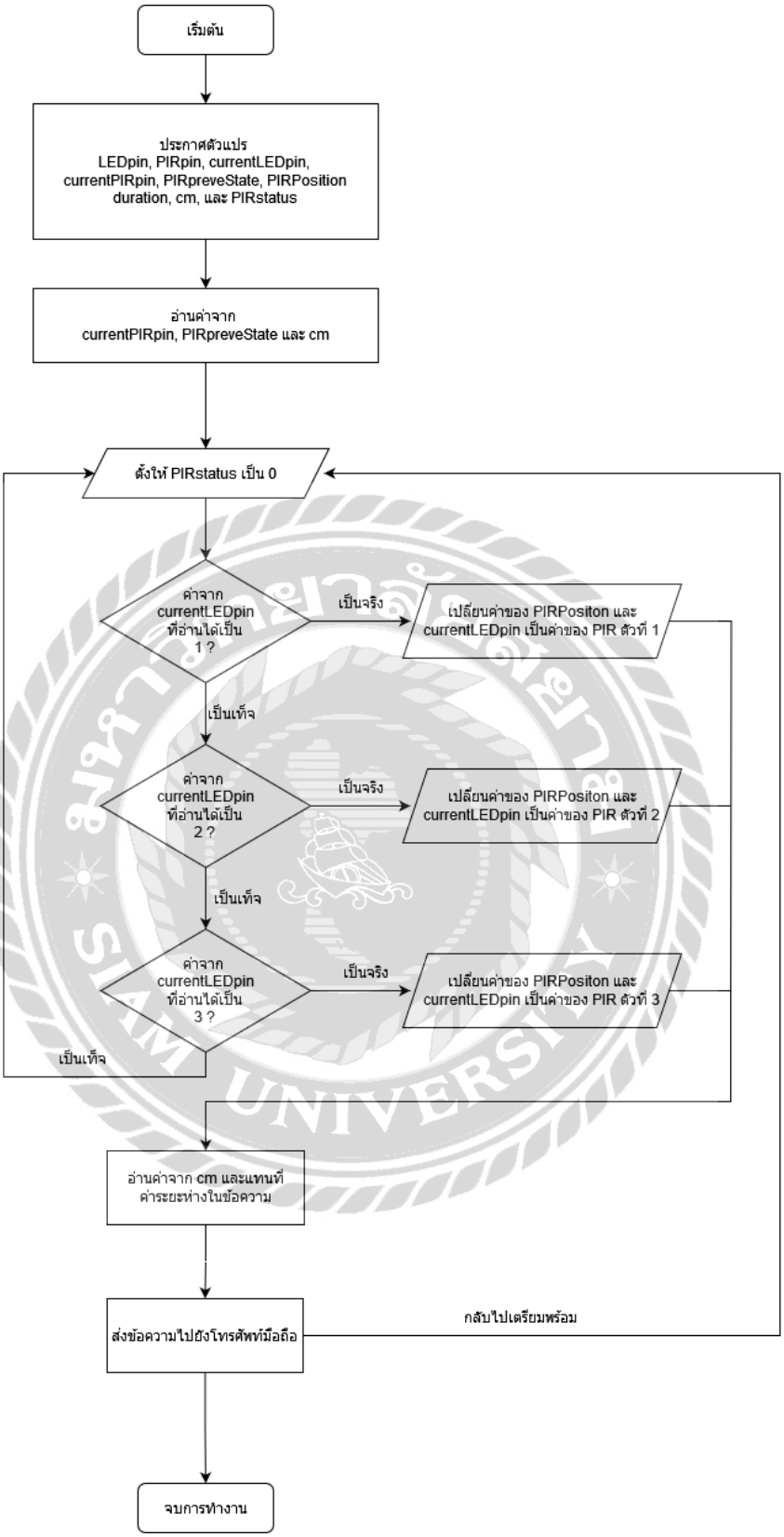
ภาพรวมของโครงงานคือการทำตัวฐานของกล้องวงจรปิดสามารถวัดระยะทางระหว่างตัวฐานกับวัตถุข้างหน้าและขยับตัวฐานตามวัตถุได้โดยที่ไม่จำเป็นต้องมีผู้ใช้เป็นคนควบคุม โดยเซนเซอร์วัดระยะทางจะทำการส่งค่าระยะทางกลับมาพร้อมกับค่าของเซนเซอร์ตรวจจับการเคลื่อนไหว เมื่อบอร์ด Arduino Uno ได้รับค่าที่ส่งกลับมาแล้วจะทำการระบุตำแหน่งของเซนเซอร์ตรวจจับการเคลื่อนไหว โดยจะมีหลอดไฟ LED ติดตามตำแหน่งของเซนเซอร์ตรวจจับการเคลื่อนไหวที่ส่งค่ากลับมา หลังจากนั้นจึงจะสั่งให้โมดูล Servo Motor ทำการขยับตัวที่หมุนตามตำแหน่งที่ได้กำหนดไว้ โดยจะมีหลักการทำงานตามภาพดังนี้



รูปที่ 3.1 ภาพรวมของการทำงานของฐาน

3.3 หลักการทำงานของระบบ

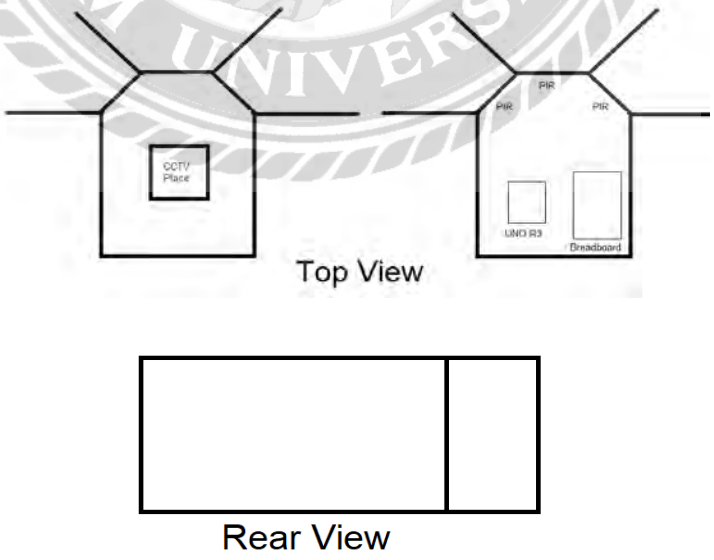
เมื่อมีสิ่งของหรือวัตถุที่มีความร้อนมากพอทำการผ่านบริเวณของหน้าของเซนเซอร์ตรวจจับการเคลื่อนไหว PSI ตัวเซนเซอร์จะถูกตั้งค่าให้เป็น “HIGH” แล้วจึงส่งค่าไปเก็บไว้ที่ตัวแปรสำหรับเก็บค่าสถานะของเซนเซอร์ตรวจจับการเคลื่อนไหวนั้น หลังจากนั้นมอเตอร์ SG960 Servo จะทำอ่านค่าที่ถูกเก็บไว้ในตัวแปรสำหรับแสดงถานะของเซนเซอร์ตรวจจับการเคลื่อนไหวตัวนั้นๆ แล้วทำการปรับค่าองศาการหมุนไปยังตำแหน่งที่ได้กำหนดไว้แบบเฉพาะสำหรับมุมแต่ละมุมที่ตัวเซนซอร์ตรวจจับการเคลื่อนไหวนั้นอยู่ หลังจากนั้นจะทำการหน่วงเวลาของเซนเซอร์ตรวจจับการเคลื่อนไหวไว้ หากเซนเซอร์ตรวจจับการเคลื่อนไหวตัวนั้นๆยังคงสามารถอ่านค่าเป็น HIGH ได้อยู่ ตัวโมดูลลำโพง Buzzer จะทำการร้องส่งเสียงเตือนออกมา แต่หากเซนเซอร์ตรวจจับการเคลื่อนไหวอ่านค่าเป็น LOW หลังจากถูกหน่วงเวลาไว้แล้ว จะทำการส่งค่ากลับไปที่มอเตอร์ SG960 Servo เพื่อให้ตัวมอเตอร์นั้นปรับค่ามุมองศาของตัวเองกลับไปอยู่ในตำแหน่งเริ่มต้นหรือตำแหน่งตรงกลาง



รูปที่ 3.2 Flow Chart หลักการทำงานของระบบ

3.4 หลักการสร้างฐานของกล้องวงจรปิด

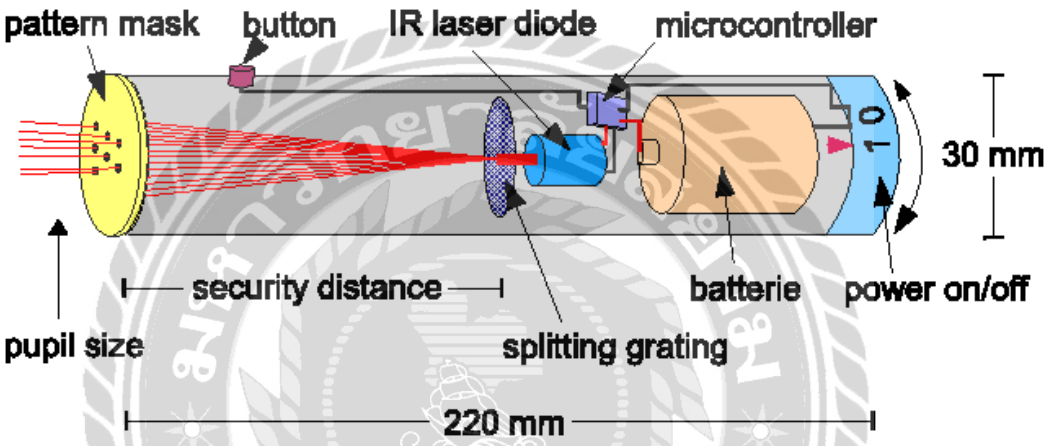
ตัวฐานของกล้องวงจรปิดนั้นจำเป็นที่จะต้องมีขนาดใหญ่อยู่พอสมควรเนื่องจากต้องใส่บอร์ด Arduino Uno R3 ที่มีขนาดใหญ่กว่าบอร์ดธรรมดา และจำเป็นต้องใส่ตัวแผง Breadboard ที่มีหน้าที่เป็นตัวกลางการเชื่อมต่อระหว่างบอร์ด Arduino กับโมดูลและเซนเซอร์แต่ละตัว โดยตัวฐานนั้นจะมีรูปทรงเป็นห้าเหลี่ยม 2 ชั้นมาต่อกันเพื่อให้สามารถแบ่งตัวโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหวไว้แต่ละมุมที่ข้างหน้าฐานได้ง่ายมากขึ้น โดยเมื่อนำโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหวไปไว้ในตำแหน่งที่เหมาะสมแล้ว จึงนำเอาวัตถุแข็งที่เป็นสี่เหลี่ยมจัตุรัสมาวางปิดไว้ทั้ง 2 ฝั่งของตัวโมดูลเอง เพื่อไม่ให้ระยะการตรวจจับของโมดูลเซนเซอร์แต่ละตัวทับกับตัวอื่น หลังจากนั้นจึงนำบอร์ด Arduino Uno R3 วางไว้ข้างในของตัวฐานโดยให้อยู่ในลักษณะแนวนอนราบกับตัวพื้นของฐานกล้องมากที่สุดเท่าที่จะทำได้ เนื่องจากหากตัวบอร์ดถูกวางไว้ในลักษณะเอียงหรือแนวตั้ง อาจส่งผลให้บอร์ด Arduino ล่มได้ ส่งผลให้สายที่ทำการเชื่อมต่อระหว่างบอร์ดกับแผง Breadboard หลุดออกจากกันได้ หลังจากนั้นจึงนำแผง Breadboard วางไว้ในแนวราบหรือแนวตั้งกับฐานโดยทำการติดกาวหรือเทปใสไว้ที่ข้างหลังของแผง Breadboard แต่ละแผงไว้ นำโมดูลลำโพง Buzzer ติดไว้ที่ด้านข้างหรือด้านหลังของตัวฐานไว้แล้วจึงนำโมดูลเซนเซอร์วัดระยะทางติดไว้ที่ตรงกลางของฐานหมุนของตัวกล้องวงจรปิดเพื่อให้สามารถวัดระยะทางสิ่งที่อยู่ตรงกลางของกล้องวงจรปิดได้



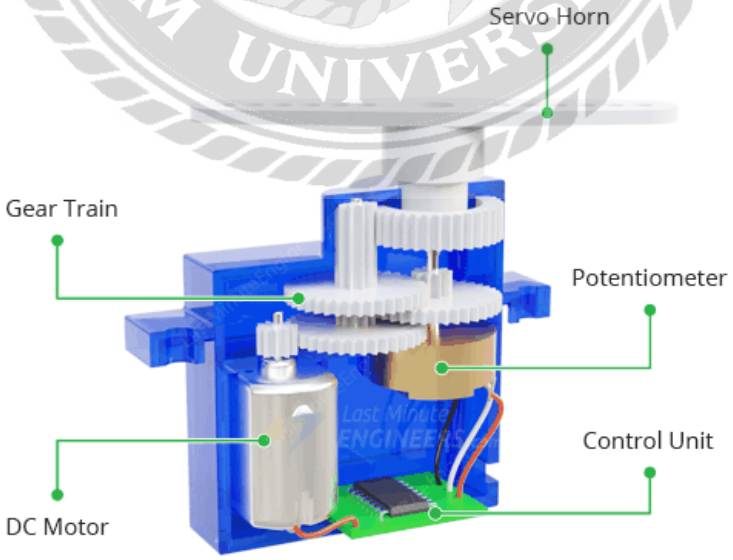
รูปที่ 3.3 การออกแบบฐานของกล้องวงจรปิด

3.5 การค้นหาองศาของฐานกล้องและการขยับตัวฐานกล้อง

ตัวโมดูลเซนเซอร์ PIR จะถูกติดตั้งไว้บริเวณส่วนหน้าของฐานกล้องโดยมีแผ่นสี่เหลี่ยมปิดที่บริเวณทั้งสองข้างของโมดูล หลังจากที่มีวัตถุความร้อนสูงมากพอเคลื่อนไหวผ่านข้างหน้าโมดูล PIR จะทำการส่งค่ากลับไปที่บอร์ด Arduino Uno เป็นค่า HIGH หลังจากนั้นตัวบอร์ดก็จะทำการปรับสถานะของ PIR Status ณ มุมนั้นให้ตรงกับตำแหน่งของตัว PIR ตัวนั้นที่ทำการส่งค่ากลับมา หลังจากที่ตัวโมดูล SG960 Servo Motor จะทำการเปลี่ยนมุมองศาไปค้างไว้ที่โมดูล PIR ตัวนั้นตามค่าที่ได้ตั้งไว้ในโปรแกรมของตัวบอร์ดเอง



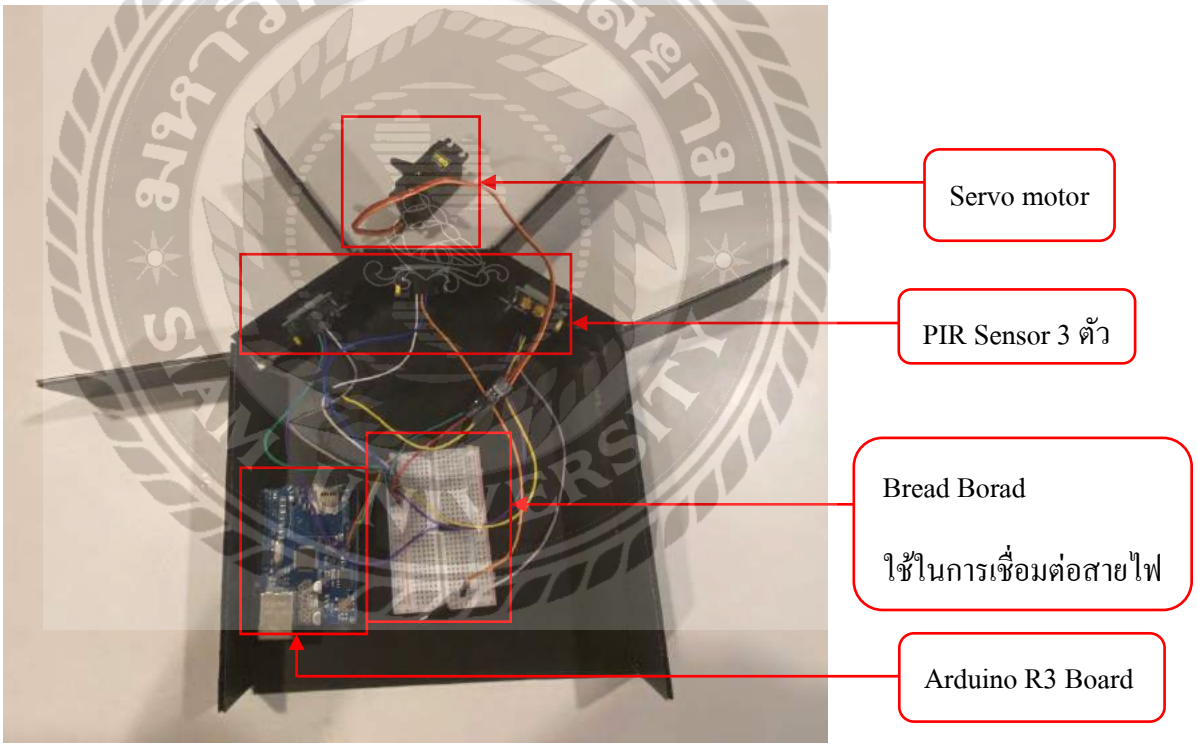
รูปที่ 3.4 หลักการทำงานของ Infrared tracking



รูปที่ 3.5 โครงสร้างภายในของโมดูล Servo Motor

3.6 การเชื่อมต่อสายไฟระหว่างโมดูลกับบอร์ด

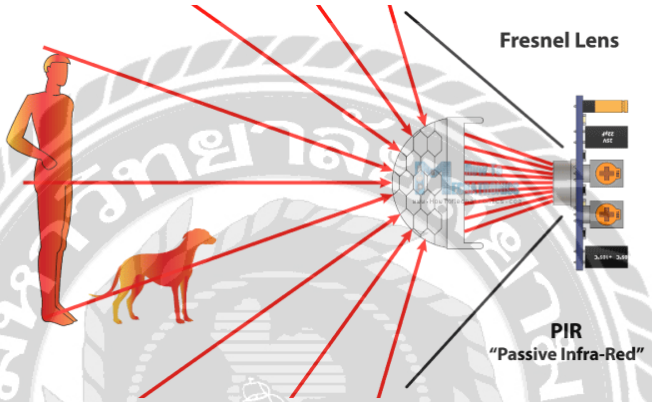
ทางผู้จัดทำได้ทำการใช้งานตัว Breadboard เป็นตัวกลางในการเชื่อมต่อระหว่างบอร์ดกับโมดูลเซนเซอร์เพื่อให้เกิดความเป็นระเบียบมากที่สุด โดยนำสายเชื่อมต่อตัวผู้-ผู้เชื่อมต่อระหว่างช่องให้พลังงาน 5 โวลต์กับช่องกระแสไฟสัญลักษณ์ “+” ของตัว Breadboard และเชื่อมสายผู้-ผู้เชื่อมต่อระหว่างช่อง Ground ในตัวย่อ “GND” ของบอร์ด Arduino Uno กับช่องสัญลักษณ์ “-“ ของตัว Breadboard เองเพื่อให้ Breadboard สามารถทำหน้าที่เป็นแหล่งพลังงานกับช่องสายดินให้กับโมดูลเซนเซอร์แต่ละตัวได้ หลังจากนั้นจึงนำสายเชื่อมต่อตัวผู้-เมียเชื่อมต่อตัวเชื่อมสำหรับ Power, Ground, Positive และ Negative กับช่องแหล่งให้พลังงานและช่องสายดินของตัว Breadboard แล้วจึงนำสายเชื่อมผู้-เมียเชื่อมต่อกับที่เชื่อม Signal หรือตัวเชื่อมสำหรับขา Pin ของแต่ละโมดูลกับช่องหมายเลขที่อยู่บนบอร์ด Arduino Uno เพื่อกำหนดขาหรือตำแหน่งสำหรับใช้ตั้งค่าภายในโปรแกรม



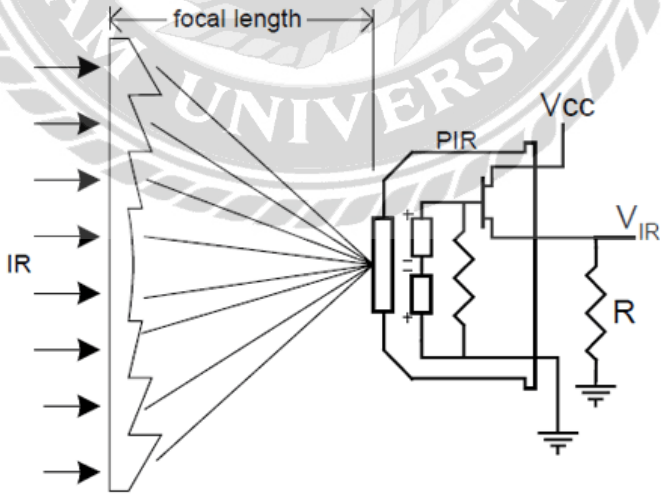
รูปที่ 3.6 การเชื่อมต่อระหว่างบอร์ดกับ PIR Sensor และ Servo Motor

3.7 หลักการทำงานของเซนเซอร์ตรวจจับการเคลื่อนไหว PIR

หลักการทำงานของโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR คือการตรวจจับความร้อนของสิ่งมีชีวิตหรือวัตถุที่มีความร้อนมากพอที่จะกระตุ้นให้เซนเซอร์ทำงานได้ หากสิ่งนั้นมีความร้อนมากพอ ตัวเซนเซอร์จะทำการตรวจจับความร้อนในจุดนั้นๆก่อนจะแปลงให้เป็นค่าพลังงานไฟฟ้าเพื่อนำไปประมวลผลให้รู้ว่ามีวัตถุอยู่ข้างหน้าต่อไป สามารถนำมาประยุกต์ใช้กับการตั้งค่าองศาการหมุนของโมดูล Servo Motor ได้โดยการตั้งค่ามุมองศาที่ต้องการให้โมดูลหมุนไปในองศานั้นๆภายในโปรแกรม Arduino IDE



รูปที่ 3.7 ตัวอย่างการทำงานของเซนเซอร์ PIR



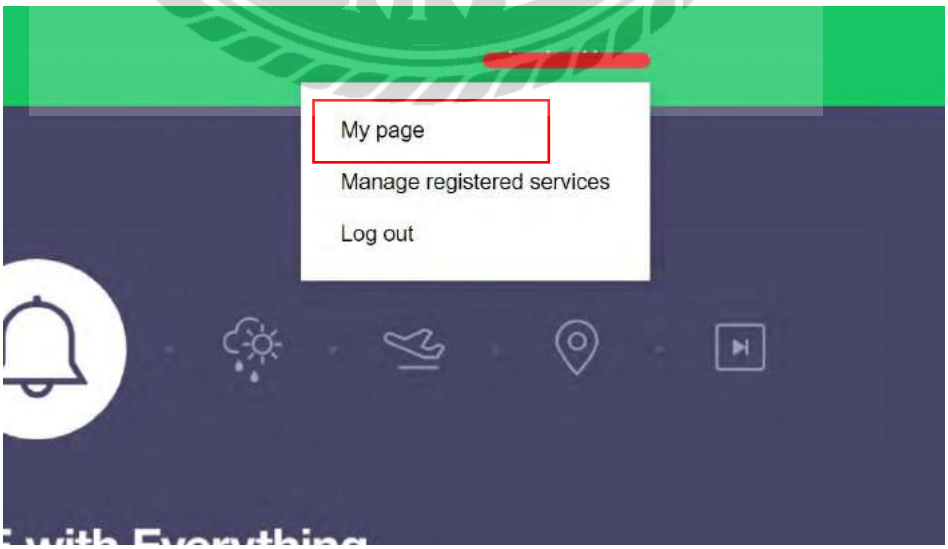
รูปที่ 3.8 ตัวอย่างโครงสร้างภายในของเซนเซอร์ PIR

3.8 การส่งข้อความแจ้งเตือนไปยังผู้ใช้งานผ่านทาง LINE Notify

ทางผู้จัดทำได้ใช้งาน LINE Notify สำหรับการส่งข้อความแจ้งเตือนไปในแอพพลิเคชั่น LINE ภายในเครื่องโทรศัพท์ของผู้ใช้งาน โดยขอ Token ในการเชื่อมต่อไปยังผู้ใช้งานเริ่มต้นด้วยการไปที่หน้าเว็บของ LINE Notify เพื่อเข้าสู่ระบบ

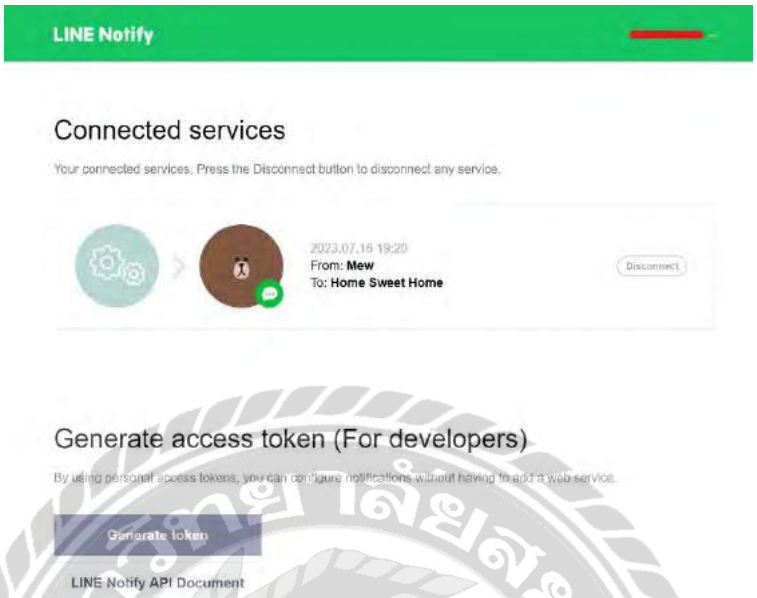
รูปที่ 3.9 หน้าเว็บ LINE Notify

เมื่อเข้าสู่ระบบแล้ว จะขึ้นชื่อบัญชีของผู้ใช้งานสามารถกดเลือก My page เพื่อเริ่มการเชื่อมต่อกับผู้ใช้งาน



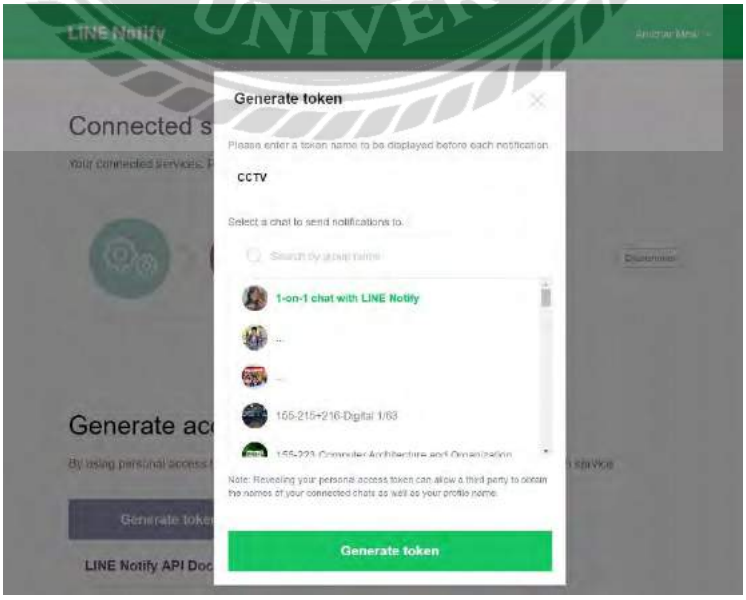
รูปที่ 3.10 เข้าสู่ระบบ LINE Notify

หน้า My page จะมีส่วนที่แสดงการเชื่อมต่อกับผู้ใช้งาน และการขอ Token ไปที่ Generate token เพื่อขอ Token



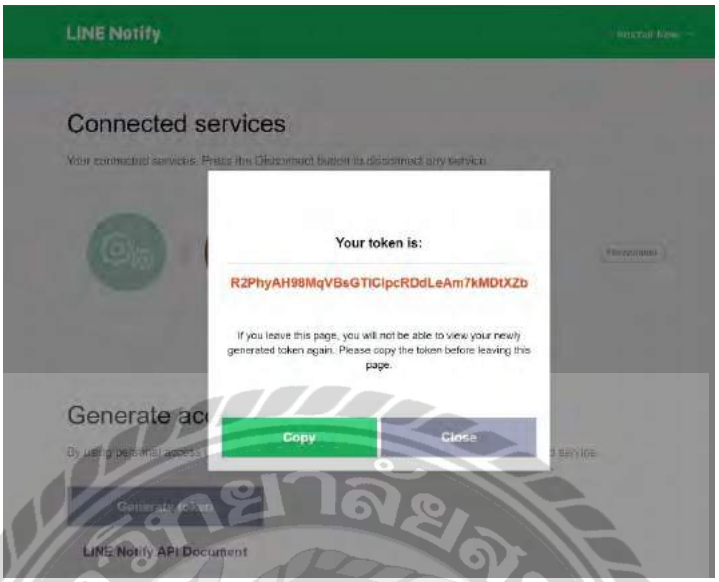
รูปที่ 3.11 หน้า My page

เมื่อ Generate token แล้ว จะต้องตั้งชื่อห้องที่ใช้สำหรับการส่งข้อความแจ้งเตือน และเลือก LINE Group ของผู้ใช้งาน ทางผู้จัดทำจะทดสอบการส่งข้อความด้วยการเลือก 1-on-1 กับ LINE Notify และตั้งชื่อ CCTV จากนั้นให้กด Generate token



รูปที่ 3.12 Generate token

เมื่อ Generate token สำเร็จ ก็จะแสดง Token เพื่อนำไปใช้งานในการส่งข้อความ โดยจะใช้ Arduino IDE ในการตั้งค่าการส่งข้อความแจ้งเตือนไปยังผู้ใช้งานเมื่อมีวัตถุความร้อนเคลื่อนไหว



รูปที่ 3.13 ตัวอย่าง Token ของ LINE Notify

3.9 การส่งค่าที่ได้รับไปที่ MyPHP Database

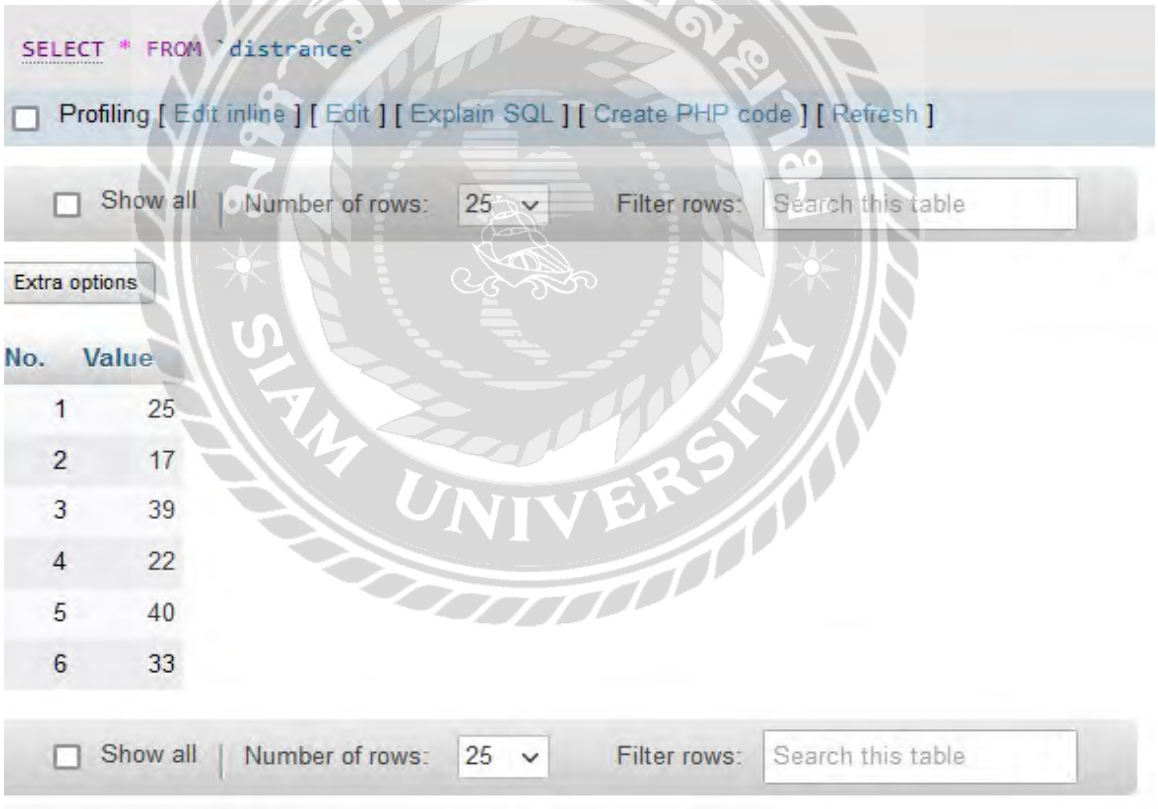
ทางผู้จัดทำได้ทำการติดตั้งตัว Ethernet Sheild W5100 ด้านบนของตัวบอร์ด Arduino Uno R3 เพื่อให้ตั้งบอร์ดสามารถเชื่อมต่ออินเทอร์เน็ตผ่านทางสายเคเบิล LAN ได้ หลังจากที่ทำการเชื่อมต่ออินเทอร์เน็ตเสร็จสิ้นแล้ว ทำการเขียนให้ตัวบอร์ด Arduino Uno R3 ทำการส่งค่าเข้าไปที่ Database ที่ทำขึ้นภายใน MyPHP ที่ถูกเขียนเป็นภาษา PHP บนโปรแกรม Visual Studio Code จากนั้นจึงเขียนโปรแกรมภายในบอร์ด Arduino Uno R3 อีกครั้งเพื่อให้สามารถรับค่าจากตัว Database เข้ามาภายในบอร์ดได้ผ่านทางอินเทอร์เน็ตที่เชื่อมสาย LAN แล้วนำค่าไปแทนในข้อความที่ถูกตั้งไว้ของ Line Notify

Config.php × temp.php

D: > Gozaru > htdocs > Database > Config.php

```php
$host = "localhost";
$user = "root";
$pass = "123456";
$db = "send";
mysql_connect($host, $user, $pass) or die("Could not connect to database");
mysql_select_db($db) or die("Could not connect to database");
mysql_query("SET NAMES utf8")
?>
```

รูปที่ 3.14 การเขียนโปรแกรม Visual Studio Code



รูปที่ 3.15 ตัวอย่างการเก็บข้อมูลภายใน MyPHP

3.10 การอธิบายการทำงานของโปรแกรม

ทางผู้จัดทำได้เขียนโค้ดภายในบอร์ด Arduino ด้วยโปรแกรม Arduino IDE โดยเรียกใช้งานตัว Library ของโมดูลมอเตอร์ Servo ก่อน หลังจากนั้นจึงได้ใช้คำสั่งภายใน Library ในการเรียกใช้งานตัวมอเตอร์ Servo แล้วตั้งองศาการหมุนของมอเตอร์ให้อยู่ในตำแหน่งเริ่มต้นหรือตำแหน่งที่ 0 องศา จากนั้นจึงตั้งขาพินของแต่ละโมดูลเพื่อให้สามารถทำงานร่วมกันกับบอร์ดได้ โดยตั้งขาพินของมอเตอร์ Servo โมดูลวัดระยะทาง โมดูลหลอดไฟ LED และโมดูลตรวจจับการเคลื่อนไหวตามลำดับ หลังจากนั้นเขียนเงื่อนไขในการเก็บข้อมูลและการตัดสินใจของตัวฐานกล้องวงจรปิดแล้ว เขียนโค้ดเพิ่มเติมในส่วนของการส่งข้อมูลให้เชื่อมต่อกับตัว LINE Notify ที่เป็นส่วนเสริม

รูปที่ 3.16 โปรแกรมการเรียกใช้งานและกำหนดขา

การเรียกใช้งาน Library ของโมดูลแต่ละตัวเพื่อให้สามารถใช้ชุดคำสั่งภายใน Library นั้นๆ ได้ หลังจากนั้นกำหนดขาเชื่อมต่อของโมดูลแต่ละตัว โปรแกรมจะรับรู้ว่าโมดูลใดเชื่อมต่ออยู่กับขาตำแหน่งใด โดยการทำการกำหนดขา ดังนี้

ตารางที่ 3.1 กำหนดตำแหน่งขาของอุปกรณ์

โมดูล	ตำแหน่งขาที่
เซนเซอร์ PIR ตัวที่ 1	2
เซนเซอร์ PIR ตัวที่ 2	3
เซนเซอร์ PIR ตัวที่ 3	4
โมดูล LED ตัวที่ 1	9
โมดูล LED ตัวที่ 2	10
โมดูล LED ตัวที่ 3	11
โมดูลมอเตอร์ Servo	7
เซนเซอร์วัดระยะทาง	13

```
const char* token   = "AxEa545DLQoo57pkytrzcA1";
const char* mes = "";
const char* tel = "0995232344";

byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};

IPAddress ip(192, 168, 1, 177);
IPAddress myDns(8, 8, 8, 8);

EthernetClient client;
IPAddress server(103,233,194,42);

unsigned long lastConnectionTime = 0;
const unsigned long postingInterval = 10L * 1000L;
```

รูปที่ 3.17 การกำหนด Token ของ Line Notify

ในส่วนของ Token จะเริ่มต้นที่สมัครใช้งานของ Line Notify แล้วจะได้รับ Token ให้นำมาใส่ไว้ภายในโค้ด เพื่อให้ตัวโปรแกรมสามารถส่งข้อความไปยังห้องสนทนา หลังจากนั้นจึงทำการกำหนดตัว IP Address ที่ได้จากตัวบอร์ด Arduino Uno R3 และกำหนดตัวเครือข่ายโดยในที่นี้ได้ใช้ของ Google ที่เปิดให้สามารถใช้งานได้โดยไม่มีค่าใช้จ่าย

```
void setup() {

Serial.begin(9600);
camServo.attach(7); // ตำแหน่งขาของมอเตอร์

for (int l = 0; l < 3; l++) { // ตั้งเซนเซอร์ทั้งหมดให้รับค่า
  pinMode(LEDpin[l], INPUT); }

for (int p = 0; p < 3; p++) { // ตั้งเซนเซอร์ทั้งหมดให้รับค่า
  pinMode(PIRpin[p], INPUT); }

Serial.print("Calibrating PIR Sensors");
for(int c = 0; c < 15; c++){
  Serial.print(".");
  delay(1000); // รอ 1 วินาที
  Serial.println("PIR Sensors Ready");
  camServo.write(78.5); // ตั้งให้มอเตอร์กลับไปอยู่ตำแหน่งตรงกลาง
}
}
```

รูปที่ 3.18 การกำหนดตัวแปรและการควบคุม

การกำหนดขาเชื่อมต่อของมอเตอร์ Servo เป็นขาที่ 7 ตั้งให้องศาอยู่ที่มุม 78 องศาซึ่งเป็นตำแหน่งตรงกลางเพื่อให้มีความสะดวกสบายในการเชื่อมต่อสายแต่ละตัว หลังจากนั้นจึงได้เขียนให้โค้ดของตัวบอร์ดทำงานตามลำดับตั้งแต่ลำดับที่ 1 จนถึง 3 ของตัวแปรที่ได้ทำการเก็บเลขตำแหน่งขาเชื่อมต่อของแต่ละตัวไว้ โดยในรูปจะมีเรียงลำดับของโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR และโมดูลหลอดไฟ LED

```
void loop() {

if (client.available()) {
    char c = client.read();
    Serial.write(c);
  }

long duration, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);
pinMode(inPin, INPUT);
duration = pulseIn(inPin, HIGH);
cm = microsecondsToCentimeters(duration);
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
```

รูปที่ 3.19 การเขียนโปรแกรมของโมดูลวัดระยะทาง

ทำการกำหนดให้โปรแกรมอ่านค่าที่วัดได้จากโมดูลเซนเซอร์วัดระยะทางให้อ่านค่าเป็นหน่วยไมโครวินาทีจากตัวแปร duration ที่ได้ตั้งไว้ จากนั้นจึงกำหนดให้ตัวโมดูลวัดระยะทางหน่วงตัวเองเป็นเวลาเสี้ยววินาทีเพื่อลดความผิดพลาดของการส่งข้อมูล และได้ประกาศตัวแปรในชื่อ cm ขึ้นมาให้เปลี่ยนค่าจากไมโครวินาทีเป็นเซนติเมตร

```
for (int PIR = 0; PIR < 3; PIR++) { //เริ่มการทำงานของแต่ละเซนเซอร์

  currentPIRpin = PIRpin[PIR]; // ตั้งค่าให้เป็นขาพินตัวปัจจุบันที่ตรวจจับอยู่
  PIRstatus = digitalRead(currentPIRpin);

  if (PIRstatus == HIGH) { // เปลี่ยนสถานะถ้าตรวจจับได้
    digitalWrite(currentLEDpin, HIGH); // เปิดไฟ LED

    if(PIRprevState[PIR] == 0) { // เปลี่ยนสถานะถ้าตรวจจับไม่ได้

      if (currentPIRposition != currentPIRpin && PIRprevState[PIR] == 0) {
        camServo.write(PIRposition[PIR]);
        Serial.print("Current angle : ");
        Serial.println(PIRposition[PIR]);
        delay(3000);
        currentPIRposition = currentPIRpin; // รีเซ็ตค่าสถานะกลับไปเป็นค่าแรก
        PIRprevState[PIR] = 1; // ตั้งให้ตัว PIR ตัวก่อนที่ตรวจจับได้เป็น 1 }
```

รูปที่ 3.20 การทำงานแบบวนซ้ำของฐานกล้องวงจรปิด

การกำหนดให้ตัวโปรแกรมเรียงลำดับของโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR ตั้งแต่ลำดับแรกจนถึงลำดับสุดท้ายเพื่อให้ตัวโปรแกรมได้รับรู้ว่ามีเซนเซอร์ที่พร้อมทำงานอยู่จำนวนเท่าใด และตั้งเงื่อนไขว่า หากตัวโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR ตัวใดที่มีการทำงานเกิดขึ้น ให้ตัวโปรแกรมจดจำลำดับของเซนเซอร์ตัวนั้นไว้ และสั่งให้หลอดไฟ LED ในตำแหน่งเดียวกันนั้นติดขึ้นหลังมีการทำงานของเซนเซอร์ตัวนั้นๆ หลังจากนั้นจึงตั้งเงื่อนไขซ้อนขึ้นมาว่า ถ้าหากตำแหน่งของมอเตอร์ Servo นั้นไม่ตรงกับสถานะและตำแหน่งของโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR ให้ตัวโปรแกรมทำการกำหนดค่าตำแหน่งมุมใหม่ลงไปแทนที่ค่าเก่า โดยค่าใหม่ได้มีการกำหนดอยู่ภายในตารางเก็บข้อมูลสำหรับตัวโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR นั้นๆแล้ว

```
      if (millis() - lastConnectionTime > postingInterval) {
        mes = "ตรวจพบสิ่งมีชีวิตเคลื่อนไหวอยู่ด้านหน้าฐานกล้อง ระยะห่าง : ",cm," ซ.ม";
        httpRequest();
      } else {
        delay(500);
        Serial.print("."); }

    PIRprevState[PIR] = 1; // ตั้งค่าให้ PIR ตัวก่อนยังคงเป็น 1 ถ้าค่าสถานะยังคงไม่เปลี่ยน
    }
  }

else { //
  digitalWrite(currentLEDpin, LOW); //ปิดไฟ LED ถ้าค่าสถานะเปลี่ยนไป
  PIRprevState[PIR] = 0; // ปรับให้ PIR ตัวก่อนดับ
  }

}

}
}
```

รูปที่ 3.21 โปรแกรมการส่งข้อความผ่านทาง LINE Notify

ทางผู้จัดทำได้ทำการตั้งเงื่อนไขซ้อนไว้ภายในอีกว่าถ้าหากมีการทำงานของโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR ณ ตำแหน่งใดเกิดขึ้น ให้นำค่าจากฐานข้อมูลที่ได้จากการเชื่อมต่อกับอินเทอร์เน็ตมาแทนที่ค่าของตัวแปร cm ภายในชุดข้อความที่ได้มีการเขียนไว้แล้ว หลังจากนั้นตัวโปรแกรมจึงจะทำการส่งข้อความนั้นไปยังโทรศัพท์ของผู้ใช้งานในภายหลัง

```
void httpRequest() {
  client.stop();
  if (client.connect(server, 80)) {
    Serial.println("connecting...");
    client.print("GET /~line/line.php?token=");
    client.print(token);
    client.print("&mes=");
    client.print(mes);
    client.print("&tel=");
    client.println(tel);
    client.println(" HTTP/1.1");
    client.println("Host: www.arduino.cc");
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();
    lastConnectionTime = millis();
  } else {
    Serial.println("connection failed");
  }
}
```

รูปที่ 3.22 การตั้งค่าเกี่ยวกับการส่งข้อความของ LINE Notify

จากรูปได้ทำการกำหนดค่าของฟังก์ชันที่ใช้สำหรับการส่งข้อความผ่านทาง LINE Notify ที่ได้ทำการเขียนภายในตัวโปรแกรม ในภาพรวมจะเป็นการให้ตัวโปรแกรมทำการแสดงข้อความเพื่อยืนยันสถานะการเชื่อมต่อและการส่งข้อความ

บทที่ 4
การทดลอง

เพื่อให้ทราบถึงการทำงานของฐานกล้องวงจรปิดที่ใช้งานโมดูลเซนเซอร์แต่ละตัวในการวัดระยะทางและจับการเคลื่อนไหว จึงได้ทำการทดลองการทำงานของอุปกรณ์ และโปรแกรมที่ได้ติดตั้งไว้ภายในบอร์ด Arduino Uno ว่าตรงตามที่ได้ออกแบบหรือไม่ เพื่อที่จะนำข้อผิดพลาดหรือจุดที่บกพร่องมาทำการแก้ไขและปรับปรุงให้โครงงานมีประสิทธิภาพที่มากขึ้น

4.1 ทดสอบการเคลื่อนไหวตามของโมดูลเซนเซอร์ PIR

วัตถุประสงค์

1. เพื่อกำหนดการทำงานของเซนเซอร์ PIR ได้อย่างแม่นยำมากขึ้น
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่

อุปกรณ์ที่ใช้ในการทดลอง

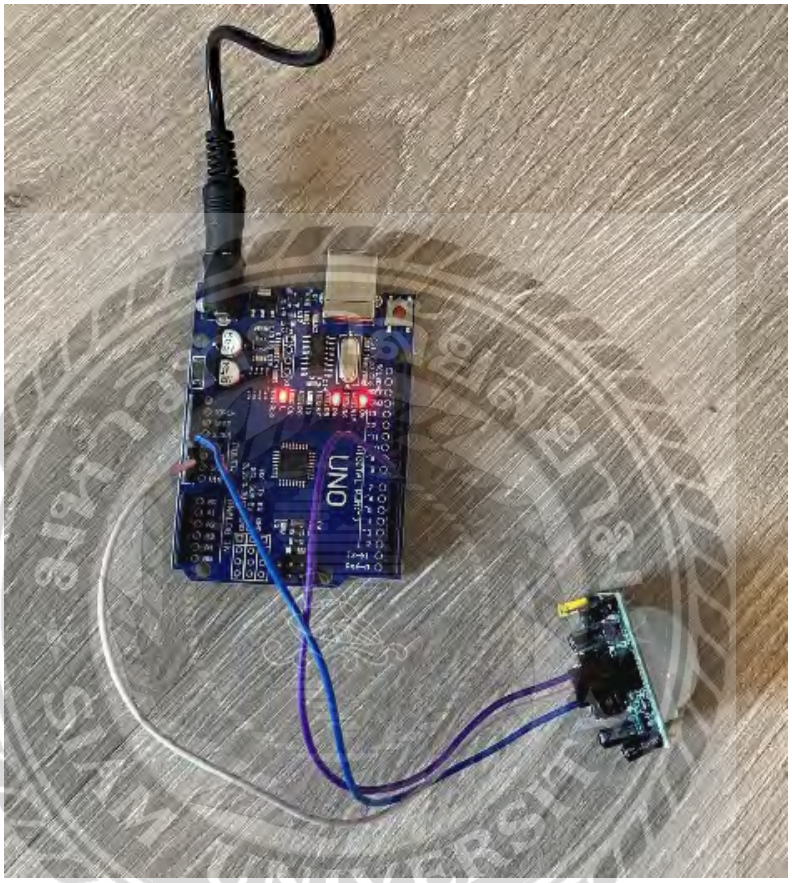
1.	บอร์ด Arduino Uno R3	1	ตัว
2.	สายเชื่อมต่อผู้-ผู้	5	เส้น
3.	เครื่องคอมพิวเตอร์	1	เครื่อง
4.	สาย USB ประเภท B	1	สาย
5.	เซนเซอร์ PIR	3	ตัว

ขั้นตอนการทดสอบ

1. ทำการกำหนดขาหรือช่องเชื่อมต่อของเซนเซอร์ PIR ให้ตรงกับโปรแกรมที่ได้เขียนไว้
2. ทำการทดสอบการทำงานของเซนเซอร์ PIR โดยการขยับมือบริเวณข้างหน้าของเซนเซอร์เป็นเวลา 1 นาที
3. ทำการทดสอบข้อ 1 กับข้อ 2 ซ้ำกับเซนเซอร์ PIR ตัวอื่นๆ

ปัญหาที่พบเจอ

1. เซนเซอร์ PIR มีความไวต่อความร้อนและแสงจากโคมไฟค่อนข้างสูง จึงทำให้ในการทดสอบบางครั้งเกิดความคาดเคลื่อน จึงจำเป็นต้องนำไปทดสอบในที่มืด ซึ่งยากต่อการจดบันทึกข้อมูล



รูปที่ 4.1 การทดสอบโมดูลเซนเซอร์ PIR

ตารางที่ 4.1 ผลการทดสอบเซนเซอร์ PIR

การทดสอบครั้งที่	ระยะที่ทำการทดสอบ	ผลลัพธ์ที่ได้
1	1 เมตร	สามารถอ่านค่าได้ปกติ
2	2 เมตร	สามารถอ่านค่าได้ปกติ
3	3 เมตร	สามารถอ่านค่าได้ปกติ
4	5 เมตร	สามารถอ่านค่าได้ปกติ
5	7 เมตรขึ้นไป	การตรวจจับมีความแม่นยำน้อยลง

4.2 การทดสอบเซนเซอร์วัดระยะทาง

วัตถุประสงค์

1. เพื่อทดสอบและดูผลการทำงานของเซนเซอร์วัดระยะทาง
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่
3. เพื่อทดสอบและแก้ไขให้เซนเซอร์วัดระยะทางมีความแม่นยำมากขึ้น

อุปกรณ์ที่ใช้ในการทดลอง

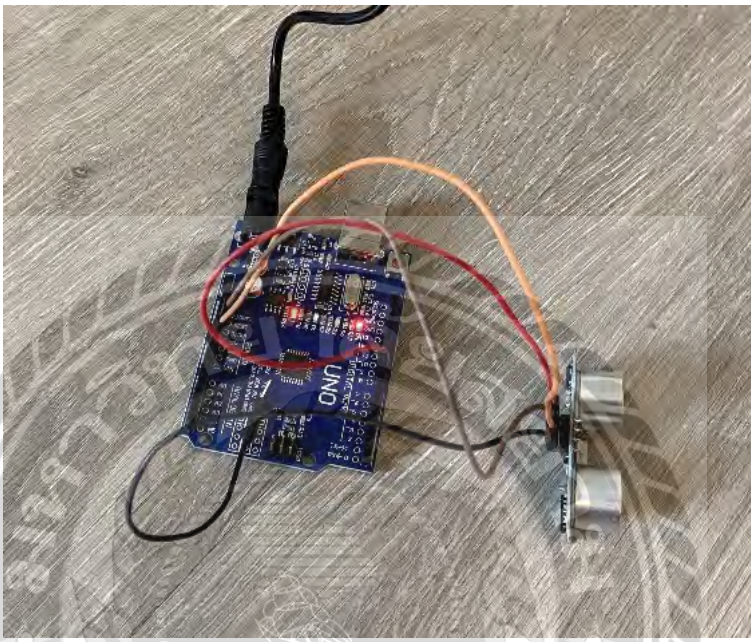
1.	บอร์ด Arduino Uno R3	1	ตัว
2.	สายเชื่อมต่อผู้-ผู้	5	เส้น
3.	เครื่องคอมพิวเตอร์	1	เครื่อง
4.	สาย USB ประเภท B	1	สาย
5.	เซนเซอร์วัดระยะทาง	1	ตัว

ขั้นตอนการทดสอบ

1. ทำการกำหนดขาหรือช่องเชื่อมต่อของเซนเซอร์วัดระยะทางให้ตรงกับโปรแกรมที่ได้เขียนไว้
2. ทำการทดสอบการทำงานของเซนเซอร์วัดระยะทางโดยการขยับมือเข้าและออกบริเวณข้างหน้าของเซนเซอร์วัดระยะทาง

ปัญหาที่พบเจอ

1. เซนเซอร์วัดระยะทางสามารถวัดระยะทางได้สูงสุดเพียง 50 เมตร จึงส่งผลให้ถ้าวัตถุอยู่ไกลมากกว่า 50 เมตร จะไม่มีการแจ้งเตือนถึงระยะห่างระหว่างกล้องวงจรปิดกับวัตถุนั้น



รูปที่ 4.2 การทดสอบโมดูลเซนเซอร์วัดระยะทาง

ตารางที่ 4.2 ผลการทดสอบเซนเซอร์วัดระยะทาง

การทดสอบครั้งที่	ระยะที่ทำการทดสอบ	ผลลัพธ์ที่ได้
1	10 ซ.ม	สามารถอ่านค่าได้ปกติ
2	20 ซ.ม	สามารถอ่านค่าได้ปกติ
3	30 ซ.ม	สามารถอ่านค่าได้ปกติ
4	40 ซ.ม	สามารถอ่านค่าได้ปกติ
5	50 ซ.ม	สามารถอ่านค่าได้ปกติ แต่หากเกินกว่านี้จะไม่สามารถอ่านได้

4.3 การทดสอบมอเตอร์ Servo

วัตถุประสงค์

1. เพื่อทดสอบและดูผลการทำงานของมอเตอร์ Servo
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่
3. เพื่อทดสอบและแก้ไขให้มอเตอร์ Servo มีความแม่นยำมากขึ้น

อุปกรณ์ที่ใช้ในการทดลอง

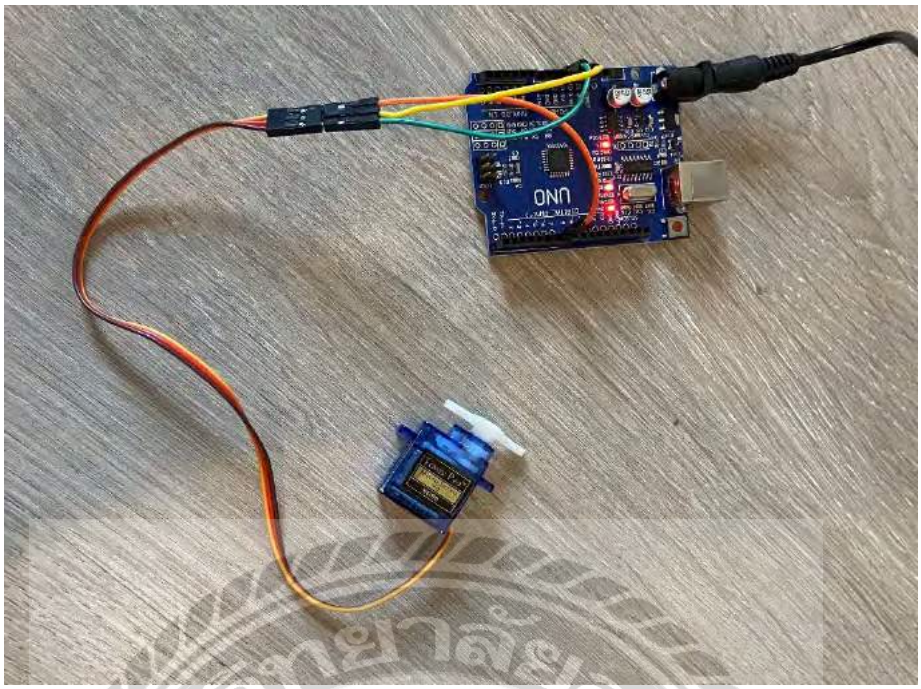
1.	บอร์ด Arduino Uno R3	1	ตัว
2.	สายเชื่อมต่อผู้-ผู้	5	เส้น
3.	เครื่องคอมพิวเตอร์	1	เครื่อง
4.	สาย USB ประเภท B	1	สาย
5.	มอเตอร์ Servo	1	ตัว

ขั้นตอนการทดสอบ

1. ทำการกำหนดขาหรือช่องเชื่อมต่อของมอเตอร์ Servo ให้ตรงกับโปรแกรมที่ได้เขียนไว้
2. ทำการทดสอบการทำงานของมอเตอร์ Servo โดยการปล่อยให้การทำงานวนซ้ำเป็นเวลา 2 นาที

ปัญหาที่พบเจอ

ไม่ประสบปัญหาที่เป็นอุปสรรค



รูปที่ 4.3 การทดสอบโมดูลเซนเซอร์ Servo

ตารางที่ 4.3 ผลการทดสอบโมดูลเซนเซอร์ Servo

การทดสอบครั้งที่	องศาที่ทำการทดสอบ	ผลลัพธ์ที่ได้
1	45 องศา	สามารถหมุนได้อย่างปกติ
2	90 องศา	สามารถหมุนได้อย่างปกติ
3	135 องศา	สามารถหมุนได้อย่างปกติ
4	180 องศา	สามารถหมุนได้อย่างปกติ

4.4 การทดสอบระบบส่งข้อความแจ้งเตือนผ่านทางแอพลิเคชั่น LINE

วัตถุประสงค์

1. เพื่อทดสอบและดูผลการทำงานของ LINE Notify
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่
3. เพื่อทดสอบและแก้ไขให้ LINE Notify สามารถส่งข้อความได้อย่างถูกต้อง

อุปกรณ์ที่ใช้ในการทดลอง

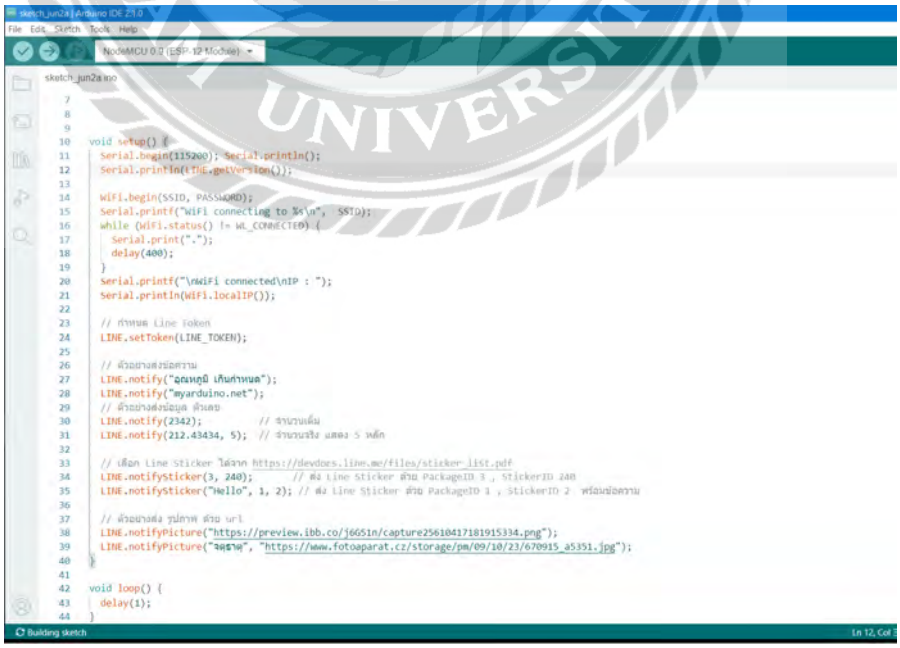
1.	บอร์ด NodeMCU ESP8266	1	ตัว
2.	สายเชื่อมต่อผู้-ผู้	5	เส้น
3.	เครื่องคอมพิวเตอร์	1	เครื่อง
4.	สาย USB ประเภท A	1	สาย
5.	ปุ่มกดสำหรับบอร์ด Arduino	1	ตัว

ขั้นตอนการทดสอบ

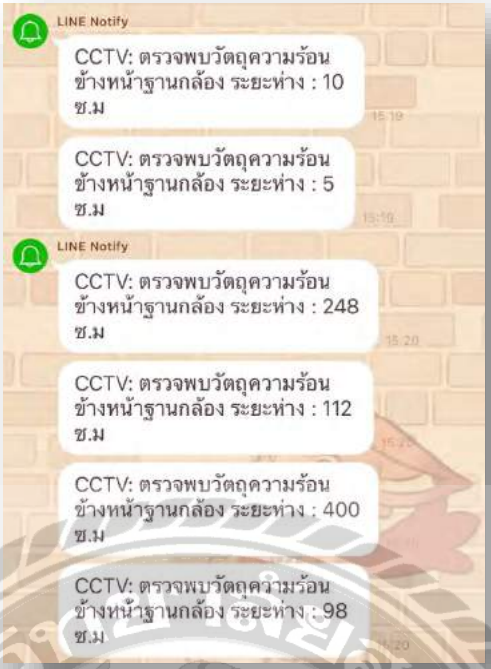
1. ทำการกำหนดภายในโปรแกรมให้ Token ของ LINE Notify ถูกต้อง
2. ทำการเขียนโปรแกรมลงไปในบอร์ด NodeMCU ESP8266 แล้วให้เริ่มทำงาน
3. รอดูผลผ่านทางหน้าจอแอพพลิเคชั่นเป็นเวลา 10 วินาที

ปัญหาที่พบเจอ

1. ในบางครั้งที่ทาง LINE Notify ไม่สามารถรับข้อมูลได้อย่างทันที อาจส่งผลให้ในบางครั้งไม่มีการแจ้งเตือนหากข้อมูลนั้นไม่ถูกส่งภายในเวลา



รูปที่ 4.4 การขียนโปรแกรมส่งข้อความ



รูปที่ 4.5 การส่งข้อความแจ้งเตือน

4.5 การทดสอบการส่งข้อมูลระหว่างบอร์ด Uno กับบอร์ด Uno

วัตถุประสงค์

1. เพื่อทดสอบและดูผลการทำงานของการรับ-ส่งข้อมูลระหว่าง 2 บอร์ด
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่
3. เพื่อทดสอบและแก้ไขให้สามารถรับ-ส่งข้อความได้อย่างถูกต้อง

อุปกรณ์ที่ใช้ในการทดลอง

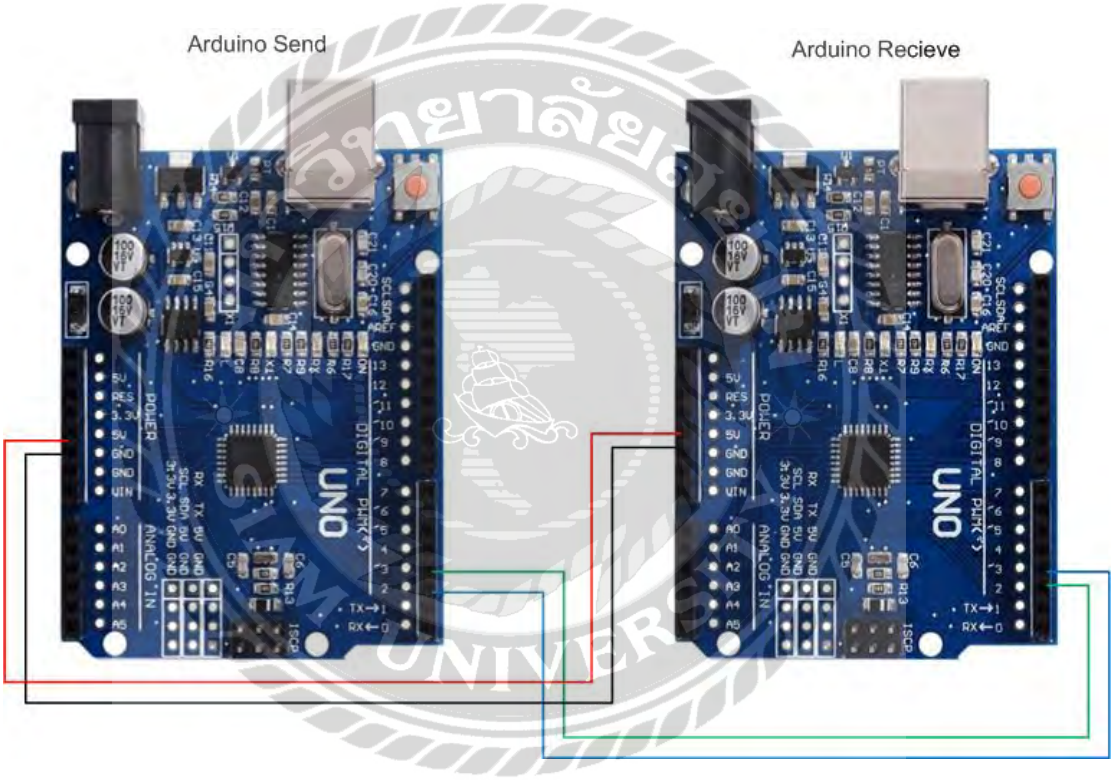
1.	บอร์ด NodeMCU ESP8266	1	ตัว
2.	บอร์ด Arduino Uno R3	1	ตัว
3.	สายเชื่อมต่อผู้-ผู้	5	เส้น
4.	เครื่องคอมพิวเตอร์	1	เครื่อง
5.	สาย USB ประเภท B	1	สาย
6.	สาย USB ประเภท A	1	สาย

ขั้นตอนการทดสอบ

1. ทำการกำหนดภายในโปรแกรมให้ Token ของ LINE Notify ถูกต้อง
2. ทำการเขียนโปรแกรมลงไปในบอร์ด NodeMCU ESP8266 แล้วให้เริ่มทำงาน
3. รอดูผลผ่านทางหน้าจอแอพพลิเคชั่นเป็นเวลา 10 วินาที

ปัญหาที่พบเจอ

ไม่ประสบปัญหาที่เป็นอุปสรรค



รูปที่ 4.6 ตัวอย่างการเชื่อมต่อระหว่างบอร์ด Arduino

4.6 การทดสอบการทำงานของฐานกล้องวงจรปิด

วัตถุประสงค์

1. เพื่อทดสอบและดูผลการทำงานของฐานกล้องวงจรปิด
2. เพื่อทดสอบโปรแกรมที่เขียนไว้ทำงานได้อย่างถูกต้องหรือไม่
3. เพื่อทดสอบและแก้ไขให้ฐานกล้องวงจรปิดสามารถทำงานได้มากขึ้น

อุปกรณ์ที่ใช้ในการทดลอง

1.	บอร์ด NodeMCU ESP8266	1	ตัว
2.	บอร์ด Arduino Uno R3	1	ตัว
3.	สายเชื่อมต่อผู้-ผู้	30	เส้น
4.	เครื่องคอมพิวเตอร์	1	เครื่อง
5.	สาย USB ประเภท B	1	สาย
6.	สาย USB ประเภท A	1	สาย
7.	เซนเซอร์ PIR	3	ตัว
8.	เซนเซอร์วัดระยะทาง	1	ตัว
9.	มอเตอร์ Servo	1	ตัว
10.	หลอดไฟ LED	3	ดวง
11.	ชุดกาวร้อน	1	ชุด

ขั้นตอนการทดสอบ

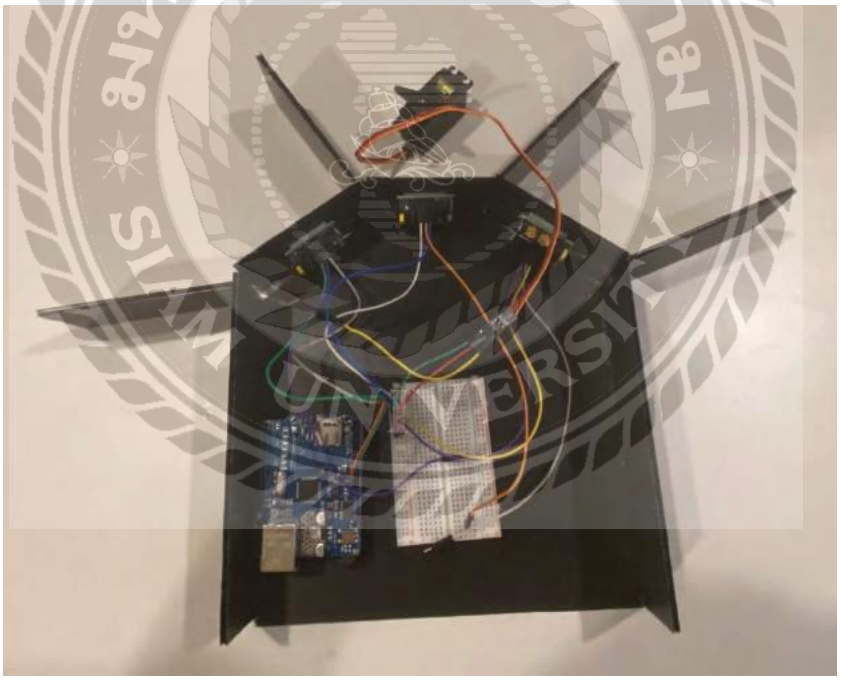
1. ทำการประกอบฐานกล้องวงจปิดขึ้นมาตามแบบที่ได้ร่างไว้
2. ทำการเชื่อมสายระหว่างแต่ละบอร์ดและโมดูลเข้าด้วยกัน
3. นำกล้องวงจรปิดวางไว้ข้างบนฐานหมุนพร้อมกับติดกาวร้อนเชื่อมระหว่างกล้องวงจรปิดกับฐานหมุน
4. ต่อแหล่งให้พลังานกับกล้องวงจรปิดและบอร์ด Arduino
5. ทดสอบใช้งานระบบฐานกล้องวงจรปิด

ปัญหาที่พบเจอ

1. แหล่งพลังงานของระบบฐานกล้องวงจรปิดใช้งานพื้นที่ค่านข้างมาก เนื่องจากจำเป็นต้องมีปลั๊กไฟตัวเมีย 2 ตัวเพื่อใช้งาน
2. ตัวฐานของกล้องวงจรปิดมีขนาดค่อนข้างใหญ่ ต้องการพื้นที่ในการวางมาก

ตารางที่ 4.4 ผลการทดสอบของฐานกล้องวงจรปิด

การทดสอบครั้งที่	วิธีการทดสอบ	ผลลัพธ์ที่ได้
1	ทดสอบการหมุนโดยวางกล้องวงจรปิดไว้ข้างบน	กล้องวงรปิดไม่สามารถยึดไว้ได้
2	ยึดกล้องวงจรปิดกับฐานด้วยกาวร้อน	กล้องวงจรปิดสามารถหมุนได้ปกติ
3	ทดสอบเซนเซอร์ PIR โดยการเดินผ่าน	เซนเซอร์ตรวจจับพร้อมกัน
4	ลดค่าการตรวจจับของเซนเซอร์ลงเล็กน้อย	สามารถตรวจจับได้โดยไม่ทับกัน



รูปที่ 4.7 ภายในของฐานกล้องวงจรปิด

รูปที่ 4.8 ฐานกล้องวงจรปิด

บทที่ 5
สรุปผลรายงานและข้อเสนอแนะ

การจัดทำโครงงานฐานกล้องวงจรปิดได้สรุปผลการดำเนินงาน ปัญหา และข้อเสนอแนะในการทำงานดังต่อไปนี้

5.1 สรุปผลโครงงาน

5.1.1 ฐานกล้องวงจรปิดสามารถตรวจจับและหมุนได้ตามจุดประสงค์ที่ตั้ง

5.1.2 การทดลองหมุนของ Servo Motor เป็นไปตามที่คาดหวัง แต่ไม่ได้แม่นยำอย่างที่วางแผน

5.1.3 โมดูลวัดระยะทางสามารถวัดระยะทางได้ตามที่คาดหวัง

5.1.4 บอร์ด Arduino สามารถส่งข้อความแจ้งเตือนผ่านทาง LINE Notify

5.2 ปัญหาการดำเนินโครงงาน

5.2.1 เนื่องจากบอร์ด NodeMCU ESP8266 ไม่รองรับ Library สำหรับมอเตอร์ Servo จึงทำให้จำเป็นต้องเปลี่ยนเป็นบอร์ด Arduino Uno R3 และใช้บอร์ด NodeMCU ESP8266 ในการรับ-ส่งข้อความแทน

5.2.2 เซนเซอร์ตรวจจับการเคลื่อนไหว PIR ตรวจจับความร้อนมากกว่าที่คาดการณ์ไว้จึงทำให้อาจเกิดความผิดพลาดขึ้นเนื่องจากระบบคิดว่ามีวัตถุอยู่ข้างหน้าได้

5.3 ข้อเสนอแนะ

5.3.1 ควรมีโมดูลเซนเซอร์ตัวอื่นมาช่วย เช่น Active Buzzer Module และ Light Alarm Moduleให้เสริมประสิทธิภาพของฐานกล้องวงจรปิดมากกว่านี้

5.3.2 ควรเพิ่มโมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว PIR เพื่อให้มีความแม่นยำของตำแหน่งองศาการหมุนกล้องวงจรปิดมากกว่านี้

5.3.3 การแจ้งเตือนด้วยโมดูลลำโพง Buzzer จะมีประสิทธิภาพมากกว่าการสังเกตจากหลอดไฟ LED เนื่องจากการแจ้งเตือนไปยังผู้ใช้งานให้รับรู้ได้จากเสียงโมดูลลำโพง Buzzer จะรับรู้ได้ดี และรวดเร็ว

บรรณานุกรม

AB. (2562). *การเก็บข้อมูลจาก Arduino ลงใน ฐานข้อมูล Database Mysql*. เข้าถึงได้จาก https://www.ab.in.th/article/33/ การเก็บข้อมูลจาก-arduino-ลงใน-ฐานข้อมูล-database-mysql-arduino-to-database

Apais. (2565). *Send Arduino Data to the Web*. เข้าถึงได้จาก https://www.instructables.com/PART-1-Send-Arduino-data-to-the-Web-PHP-MySQL-D3js/

CyberTice. (2562). *สอนใช้งาน Arduino ควบคุม Servo Motor 0-180องศา*. เข้าถึงได้จาก https://www.cybertice.com/article/109/สอนใช้งาน-arduino-ควบคุม-servo-motor-0-180องศา

CyberTice. (2562*). สอนใช้งาน Arduino วัดระยะทางด้วย เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04*. เข้าถึงได้จาก https://www.cybertice.com/article/110/สอนใช้งาน-arduino-วัดระยะทางด้วย-เซ็นเซอร์วัดระยะทาง-ultrasonic-module-hc-sr04

CyberTice. (2559). *สอนใช้งาน PIR เซนเซอร์ตรวจจับความเคลื่อนไหว Motion Sensor Module กับ arduino*. เข้าถึงได้จาก https://www.cybertice.com/article/43/สอนใช้งาน-pir-เซนเซอร์ตรวจจับความเคลื่อนไหว-motion-sensor-module-กับ-arduino

Wikipedia. (2557). Arduino. เข้าถึงได้จาก https://en.wikipedia.org/wiki/Arduino

ประวัติผู้จัดทำ

ชื่อ-สกุล : นาย ปาราเมศ บัวงาม

รหัสนักศึกษา : 6204000009

ภาควิชา : วิศวกรรมคอมพิวเตอร์

คณะ : วิศวกรรมศาสตร์

ที่อยู่ปัจจุบัน : 449 ถนนพหลโยธิน แขวงอนุสาวรีย์ เขตบางเขน กรุงเทพมหานคร 10220

ชื่อ – สกุล : นางสาวอรุชา ตรีสุรนรา

รหัสนักศึกษา : 6204000001

ภาควิชา : วิศวกรรมคอมพิวเตอร์

คณะ : วิศวกรรมศาสตร์

ที่อยู่ปัจจุบัน : 88/749 ซ.พระรามที่ 2 ซอย 69 แยก 3-15-7 แขวงแสมดำ เขตบาง ขุนเทียน กรุงเทพฯ 10150