แขนกลควบคุมทางไกลในห้องผ่าตัด

Tele-Operation of Robotic Arm

นางสาวภัคจิรา วชิรสกุลชัย
Phakjira Vachirasakulchai

นางสาวรมิตา ธรรมจารี
Ramita Tommajaree

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

แขนกลควบคุมทางไกลในห้องผ่าตัด

Tele-Operation of Robotic Arm

นางสาวภัคจิรา วชิรสกุลชัย

Phakjira Vachirasakulchai

นางสาวรมิตา ธรรมจารี

Ramita Tommajaree

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนกลควบคุมทางไกลในห้องผ่าตัด

Tele-Operation of Robotic Arm

โดย

นางสาวภัคจิรา วชิรสกุลชัย

นางสาวรมิตา ธรรมจารี

อาจารย์ที่ปรึกษา

ดร.สรินพร วิสิฐสัทธาพงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์	สาขาวิศวกรรมชีวการแพทย์ ปีการศึกษา 2561
สาขาวิชา	วิศวกรรมชีวการแพทย์
คณะ	วิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง	แขนกลควบคุมทางไกลในห้องผ่าตัด
	Tele-Operation of Robotic Arm
นักศึกษา	นางสาวภัคจิรา วชิรสกุลชัย รหัสนักศึกษา 58010941
	นางสาวรมิตา ธรรมจารี รหัสนักศึกษา 58011042

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ดร.สรินพร วิสิฐสัทธาพงศ์)

อาจารย์ที่ปรึกษาปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์	แขนกลควบคุมทางไกลในห้องผ่าตัด
นักศึกษา	นางสาวภัคจิรา วชิรสกุลชัย รหัสนักศึกษา 58010941
	นางสาวรมิตา ธรรมจารี รหัสนักศึกษา 58011042
ปริญญา	วิศวกรรมศาสตร์บัณฑิต
สาขาวิชา	วิศวกรรมชีวการแพทย์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ดร.สรินพร วิสิฐสัทธาพงศ์

บทคัดย่อ

งานวิจัยนี้จัดทำขึ้นเพื่อช่วยให้บุคลากรทางการแพทย์ในด้านต่างๆ สามารถเข้าถึงผู้ป่วยในพื้นที่ห่างไกลได้ โดยผ่านระบบอินเตอร์เน็ต เพื่อให้ผู้ป่วยไม่ต้องเดินทางไกลเพื่อมาโรงพยาบาลในเมือง ดังนั้นจึงเป็นการช่วยลดค่าใช้จ่ายในการเดินทาง อีกทั้งยังประหยัดเวลาด้วย

โดยงานวิจัยนี้ จะเป็นการสร้างแขนกลควบคุมทางไกลเพื่อใช้ในห้องผ่าตัด และจะทำการควบคุมจากทางไกลผ่านตัวควบคุม (Joystick) ซึ่งงานวิจัยนี้จะแบ่งเป็น 2 ส่วน คือส่วนของฮาร์ดแวร์ หรือก็คือการออกแบบและปริ๊นท์สามมิติ (3d print) ของตัวแขนกล และ ตัวควบคุม และส่วนของซอฟแวร์ คือการเขียนโปรแกรมควบคุมแขนกล และส่งค่าควบคุมผ่านอินเตอร์เน็ต อีกทั้งยังมีการส่งภาพแบบทันทีจากทางฝั่งแขนกลไปยังผู้ควบคุมอีกด้วย

โดยที่จะรับส่งข้อมูลระหว่างแขนกลกับตัวควบคุมด้วย NodeMCU หรือในที่นี้จะใช้ ESP32 2 ตัว ตัวหนึ่งจะรับค่าการทำงานจากตัวควบคุมและส่งค่านั้นไปให้ ESP32 อีกตัวเพื่อที่จะนำค่าดังกล่าวไปควบคุมการทำงานของแขนกล ส่วนของการรับส่งภาพแบบทันทีจะใช้ Raspberry pi เป็นตัวส่งภาพนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Tile	Tele-Operation of Robotic Arm
Student	Miss Phakjira Vachirasakulchai Student ID 58010941
	Miss Ramita Tommajaree Student ID 58011042
Degree	Bachelor of Engineering
Program	Biomedical Engineering
Year	2018
Thesis Advisor	Dr. Sarinporn Visitsattapongse

Abstract

This research is for facilitate the healthcare worker to approach patient that located in distant areas via internet. Also, patient need not to get to the city to see a doctor. So, this research will reduce cost of travelling and time.

This research is to create tele control robot arm use in the operating room which is controlled by joystick from anywhere. It is separated into 2 part that is hardware part to design and print 3D robot arm and joystick and software part which is code to control robot arm also send controlling data via internet. Furthermore, this project will send real-time video from robot arm to controller side.

The data is sent between robot arm and joystick by nodeMCU or in this project use 2 ESP32, one will receive data from joystick and send the data to another for control robot arm.

In the section of sending real-time video, this project uses Raspberry pi as a device for sending.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงได้ด้วยดีด้วยความกรุณาและคำปรึกษาจากบุคคลหลายท่าน อีกทั้งให้ความรู้ และคำแนะนำในการแก้ปัญหาต่าง ๆ ขอขอบพระคุณ ดร.สรินพร วิสิฐสัทธาพงศ์ อาจารย์ที่ปรึกษา และ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ ที่ให้ความสนับสนุน ความช่วยเหลือ ตลอดจนชี้แนะและตรวจผลงานตลอดการทำปริญญานิพนธ์ในครั้งนี้

ขอขอบพระคุณ นายภาคภูมิ เหรียญอารีย์ นายณัฐนันท์ วันลักษณ์ และ นายอนิวัฒน์ จูห้อง นักศึกษาปริญญาโทหลักสูตรวิศวกรรมชีวการแพทย์ ที่ได้ให้คำปรึกษา คำแนะนำ และแบ่งปันความรู้ในด้านต่าง ๆ รวมถึงสนับสนุนอุปกรณ์และเครื่องมือต่าง ๆ ตลอดจนคำชี้แนะที่ดีในการแก้ปัญหาที่เกิดขึ้นจนปริญญานิพนธ์นี้เป็นไปอย่างราบรื่นและสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณเพื่อน ๆ ในภาควิชาวิศวกรรมชีวการแพทย์ที่ได้ให้คำแนะนำ แบ่งปันความรู้ สนับสนุนให้ความช่วยเหลือตลอดมา

ขอขอบพระคุณหลักสูตรวิศวกรรมชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้โอกาสศึกษาหาความรู้ รวมถึงสนับสนุน สถานที่และงบประมาณในการจัดทำปริญญานิพนธ์นี้

สุดท้ายนี้ สำหรับคุณงามความดีอันใดที่เกิดจากปริญญานิพนธ์นี้ ข้าพเจ้าขอมอบให้บิดามารดา และอาจารย์ซึ่งเป็นที่รักและเคารพยิ่ง ที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ดีให้แก่ข้าพเจ้า

ภัคจิรา วชิรสกุลชัย

รมิตา ธรรมจารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ.....I

ABSTRACT.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมา..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 1

1.3 ขอบเขตของการวิจัย..... 1

1.4 ระยะเวลา..... 2

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง..... 4

2.1 หุ่นยนต์และแขนกล..... 4

2.1.1 ส่วนประกอบของหุ่นยนต์..... 5

2.1.2 แขนกล..... 6

2.2 การควบคุมแขนกล..... 7

2.2.1 Arduino..... 7

2.2.2 Servo Motor..... 12

2.2.3 มอเตอร์ไฟฟ้า..... 15

2.2.4 Motor Drive Module L298N..... 22

2.2.5 การใช้ Arduino UNO R3 กับ L298N ควบคุมมอเตอร์..... 24

2.2.6 Joystick Module..... 25

2.2.7 JoyStick Shield expansion board for Arduino..... 26

2.3 ส่วนควบคุมแขนกลทางไกล..... 27

2.3.1 Raspberry Pi 3 Model B..... 27

2.3.2 Raspberry Pi V2 Camera Board..... 29

2.3.3 การทำ Web server เพื่อรับข้อมูลจาก Sensor และควบคุมอุปกรณ์ต่างๆ..... 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.4 NodeMCU 61
2.3.5 ESP32 64
2.4 การส่ง-รับภาพทางไกล 67
2.4.1 สตรีมมิ่งมีเดีย (Streaming Mediatechnology) 67
2.4.2 วิธีการเผยแพร่ไฟล์สตรีมมิ่งมีเดีย 68
2.4.3 ลักษณะการส่ง Streaming Media 68
2.4.4 ความแตกต่างระหว่าง streaming with web server กับ streaming with streaming media server 69
2.4.5 องค์ประกอบพื้นฐานระบบสตรีมมิ่งมีเดีย 70
2.4.6 เทคโนโลยีการส่งไฟล์ข้อมูล 71
2.4.7 Motion JPEG 71
บทที่ 3 การออกแบบและวิธีการดำเนินการโครงงาน 76
3.1 อุปกรณ์ 76
3.1.1 Micro Servo (SG90) 76
3.1.2 Servo Motor (MG995 0-180 องศา) 76
3.1.3 DC Motor 77
3.1.4 Motor Drive Module L298N 77
3.1.5 joystick module 77
3.1.6 Button Switch Module 78
3.1.7 Arduino UNO 78
3.1.8 ESP-WROOM-32 78
3.1.9 Raspberry Pi 3 Model B 79
3.1.10 Camera Module Board REV 1.3 5MP Webcam Video 1080p 720p 79
3.1.11 Micro USB Power Adapter 5V 2.5A 79
3.1.12 Switching 12V 5A 80
3.1.13 Bearing 80
3.1.14 DC Step Down Buck Converter Module (5A 5-36V เป็น 1.2V-32V) 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 วิธีการ 81
3.2.1 ส่วนการออกแบบตัวแขนกลและตัวควบคุม 81
3.2.2 ส่วนการควบคุมและการส่งค่าทางไกล 82
3.2.3 ส่วนรับ-ส่งภาพ 101
บทที่ 4 ผลการทดลอง 104
4.1 ประกอบอุปกรณ์ 104
4.1.1 การประกอบแขนส่วนต่างๆเข้ากับมอเตอร์ไฟฟ้าและเซอร์โวมอเตอร์ 104
4.1.2 แขนกล และตัวควบคุม 106
4.2 บัดกรีวงจร 107
4.3 ค่า Output ทางฝั่งควบคุม เมื่อทำการควบคุมอุปกรณ์ในทิศทางต่างๆ 108
4.4 ข้อมูลที่ publish ออกไปทางหน้าต่าง debug บน node-red 108
4.5 การทำงานของแขนกล เมื่อรับ Input เข้ามา 109
4.6 การควบคุมแขนกลเมื่อตัวควบคุมอยู่ต่างสถานที่กับแขนกล 109
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 111
5.1 สรุปผลการวิจัย 111
5.2 ปัญหาและแนวทางการแก้ปัญหา 111
5.2.1 ปัญหาทางด้านฮาร์ดแวร์ 111
5.2.2 ปัญหาทางด้านซอฟต์แวร์ 112
5.3 ข้อเสนอแนะ 112
บรรณานุกรม 113
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 1.1 ตารางแสดงแผนการดำเนินงานในภาคการศึกษาที่ 1 2

ตารางที่ 1.2 ตารางแสดงแผนการดำเนินงานในภาคการศึกษาที่ 2 3

ตารางที่ 3.1 แสดงการเชื่อม pin ของ Joystick1 กับ ESP32 93

ตารางที่ 3.2 แสดงการเชื่อม pin ของ Joystick2 กับ ESP32 93

ตารางที่ 3.3 แสดงการเชื่อม pin ของ Button กับ ESP32 93

ตารางที่ 3.4 แสดงการเชื่อม pin ของ servo1 กับ ESP32 และ DC power source 94

ตารางที่ 3.5 แสดงการเชื่อม pin ของ servo2 กับ ESP32 และ DC power source 95

ตารางที่ 3.6 แสดงการเชื่อม pin ของ servo3 กับ ESP32 และ DC power source 95

ตารางที่ 3.7 แสดงการเชื่อม pin ของ ESP32 กับ Arduino UNO 95

ตารางที่ 3.8 แสดงการเชื่อม pin ของ Arduino UNO กับ Motor drive L298N 96

ตารางที่ 3.9 แสดงการเชื่อมต่อ Motor drive L298N กับ DC power source 96

ตารางที่ 3.10 แสดงการเชื่อมต่อ Arduino UNO กับ DC power source 96

ตารางที่ 3.11 แสดงการเชื่อมต่อ ESP32 กับ DC power source 96

ตารางที่ 4.1 ค่า Output ทางฝั่งควบคุม เมื่อทาการควบคุมอุปกรณ์ในทิศทางต่างๆ 108

ตารางที่ 4.2 การทางานของแขนกล เมื่อรับ Input เข้ามา 109

ตารางที่ 4.3 ผลการควบคุม 110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 Arduino UNO R3....8
รูปที่ 2.2 รูปแบบการเขียนโปรแกรมบน Arduino....9
รูปที่ 2.3 การเลือกรุ่นบอร์ด Arduino ที่ต้องการ upload....9
รูปที่ 2.4 การเลือกหมายเลข Comport ของบอร์ด....9
รูปที่ 2.5 การกดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โค้ดโปรแกรม และการ Upload โค้ดโปรแกรม....10
รูปที่ 2.6 Layout & Pin out บอร์ด Arduino....11
รูปที่ 2.7 ส่วนประกอบภายนอก RC Servo Motor....13
รูปที่ 2.8 ส่วนประกอบภายใน RC Servo Motor....13
รูปที่ 2.9 Servo Motor Block Diagram....14
รูปที่ 2.10 ตัวอย่างการเชื่อมต่อ RC Servo Motor เข้ากับบอร์ด Arduino....14
รูปที่ 2.11 วงจรการทำงานของมอเตอร์ไฟฟ้ากระตรงแบบอนุกรม....16
รูปที่ 2.12 วงจรการทำงานมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน....17
รูปที่ 2.13 วงจรการทำงานมอเตอร์ไฟฟ้ากระตรงแบบชอร์ทชั้นท์คอมปาวด์มอเตอร์....17
รูปที่ 2.14 วงจรการทางานมอเตอร์ไฟฟ้ากระตรงแบบลองชั้นท์คอมปาวด์มอเตอร์....18
รูปที่ 2.15 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้ากระแสตรง....18
รูปที่ 2.16 แปรงถ่าน....20
รูปที่ 2.17 หลักการทำงานของ....20
รูปที่ 2.18 Motor Drive Module L298N....22
รูปที่ 2.19 Pulse Width Modulation ที่ 0%, 25%, 50%, 75%, 100% Duty Cycle มอเตอร์....23
รูปที่ 2.20 L298N ควบคุมมอเตอร์....24
รูปที่ 2.21 แสดงการต่อใช้งานอุปกรณ์เพื่อควบคุมมอเตอร์....24
รูปที่ 2.22 Joystick Module....26
รูปที่ 2.23 JoyStick Shield expansion board....27
รูปที่ 2.24 ส่วนประกอบของ Raspberry Pi 3....28
รูปที่ 2.25 Raspberry Pi Camera....30
รูปที่ 2.26 ซ็อกเก็ต CAMERA....31
รูปที่ 2.27 ต่อสายแพรกับซ็อกเก็ต CAMERA....32
รูปที่ 2.28 ดึงซ็อกเก็ต CAMERA....32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.29 สายแพรที่ต่อกับ Raspberry pi....33
รูปที่ 2.30 สายแพรที่ต่อกับซ็อกเก็ต CAMERA....33
รูปที่ 2.31 หน้า Command ของ Raspberry pi....34
รูปที่ 2.32 ใช้คำสั่ง sudo raspi-config....34
รูปที่ 2.33 หน้าเมนู Raspberry Pi Software Configuration Tool....34
รูปที่ 2.34 หน้าเมนูกดเลือก Enable เพื่อเปิดใช้งานกล้อง....35
รูปที่ 2.35 หน้าเมนู Raspberry Pi Software Configuration Tool....35
รูปที่ 2.36 หน้าเมนูกดเลือก Yes เพื่อรีบูท Raspberry Pi....36
รูปที่ 2.37 ใช้คำสั่ง sudo -o test.jpg....36
รูปที่ 2.38 หน้าต่างโฟลเดอร์ /home/pi....36
รูปที่ 2.39 โครงสร้างเครือข่ายคอมพิวเตอร์....37
รูปที่ 2.40 รูปแบบการส่งและรับข้อมูลระหว่าง ไคลเอนต์ และ เซิฟเวอร์....38
รูปที่ 2.41 หน้าต่างสำหรับเปิดใช้งาน Service ของโปรแกรม Xampp....39
รูปที่ 2.42 หน้าหลักของ Web Server....39
รูปที่ 2.43 การรับข้อมูลและการแสดงผลข้อมูลบน Web Server....42
รูปที่ 2.44 หลักการของ MQTT....43
รูปที่ 2.45 ตัวอย่าง code ที่ทดลอง....48
รูปที่ 2.46 code ที่ใช้สร้าง Web server ใน Node.js....53
รูปที่ 2.47 code ที่ใช้สร้าง Web server ใน Express.js....54
รูปที่ 2.48 NodeMCU 2 ตัวเชื่อมต่อข้อมูลหากัน....54
รูปที่ 2.49 ลากเส้นเชื่อม Node แต่ละตัว....55
รูปที่ 2.50 หน้าต่าง Edit function mode....55
รูปที่ 2.51 แสดงการลากเส้นเชื่อมกันระหว่างคนละ Protocal....56
รูปที่ 2.52 หน้าเว็บ Node.js....57
รูปที่ 2.53 รันคำสั่ง node-red ในหน้า CMD....57
รูปที่ 2.54 หน้า Node-red....57
รูปที่ 2.55 หน้า Node-red ที่แสดงการลากเส้นเชื่อม....58
รูปที่ 2.56 หน้ารายละเอียดของ Input....58
รูปที่ 2.57 หน้าการตั้งค่าของ Input....59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.58 แสดงการลากเส้นเชื่อม Debug....60
รูปที่ 2.59 หน้าเว็บ HiveMQ....60
รูปที่ 2.60 ทดสอบการส่ง Messenge....61
รูปที่ 2.61 แสดงข้อความที่ส่งทางด้านขวา....61
รูปที่ 2.62 หน้าต่าง Preferences....63
รูปที่ 2.63 หน้าต่าง Board Manager....63
รูปที่ 2.64 หน้าต่าง Preferences....66
รูปที่ 2.65 หน้าต่าง Boards Manager....67
รูปที่ 2.66 เปรียบเทียบ Download กับ Streaming....69
รูปที่ 2.67 องค์ประกอบพื้นฐานระบบสตรีมมิ่งมีเดีย....71
รูปที่ 2.68 หน้าต่างตัวอย่าง Raspberry Pi ที่ควรจะได้....74
รูปที่ 2.69 ข้อมูลที่ได้จากคำสั่ง fswebcam —verbose....75
รูปที่ 3.1 Micro Servo (SG90)....76
รูปที่ 3.2 Servo Motor (MG995 0-180 องศา)....76
รูปที่ 3.3 DC Motor....77
รูปที่ 3.4 Motor Drive Module L298N....77
รูปที่ 3.5 joystick module....77
รูปที่ 3.6 Button Switch Module....78
รูปที่ 3.7 Arduino UNO....78
รูปที่ 3.8 ESP-WROOM-32....78
รูปที่ 3.9 Raspberry Pi 3 Model B....79
รูปที่ 3.10 Camera Module Board REV 1.3 5MP....79
รูปที่ 3.11 Micro USB Power Adapter 5V 2.5A....79
รูปที่ 3.12 Switching 12V 5A....80
รูปที่ 3.13 รูปที่ 3.13 Bearing A) Thrust bearing, B) Ball bearing....80
รูปที่ 3.14 DC Step Down Buck Converter Module....81
รูปที่ 3.15 โปรแกรม Autodesk....81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 3.16 แขนส่วนต่างๆที่ออกแบบโดย Autodesk Inventor A) แขนส่วนที่ 1, B) แขนส่วนที่ 2, C) แขนส่วนที่ 3.....81
รูปที่ 3.17 E) ส่วนที่ใช้ควบคุม(Joystick), F) ส่วนที่ใช้ยึดแขนท่อนล่างสุด, G) ส่วนฐานของแขนกล, H) ส่วนที่ใช้ยึดกล้องและกรรไกร.....82
รูปที่ 3.18 การต่อ Joystick เข้ากับ Arduino.....83
รูปที่ 3.19 การต่อ Button Switch เข้ากับ Arduino.....83
รูปที่ 3.20 การต่อเซอร์โวเข้ากับ Arduino.....84
รูปที่ 3.21 การต่อมอเตอร์เข้ากับ Arduino.....84
รูปที่ 3.22 การต่อวงจรเพื่อจาลองการควบคุมแขนกล.....85
รูปที่ 3.23 การ Create Project.....87
รูปที่ 3.24 การเลือก VM instances และเลือก Create instance.....87
รูปที่ 3.25 หน้าต่างที่แสดงเพื่อให้ใส่ข้อมูล.....87
รูปที่ 3.26 หน้า Dashboard ของ VM instances.....88
รูปที่ 3.27 หน้าต่างที่แสดงขึ้นมาเมื่อ run คำสั่ง node-red.....89
รูปที่ 3.28 การเข้าใช้งาน node-red ผ่านเว็บเบราเซอร์.....90
รูปที่ 3.29 การทำงานในหน้า node-red.....90
รูปที่ 3.30 หน้าต่างตั้งค่าเมื่อเข้าไปใน mqtt (input).....90
รูปที่ 3.31 หน้าต่างตั้งค่าเมื่อเข้าไปใน mqtt (output).....91
รูปที่ 3.32 หน้าต่างตั้งค่าเมื่อเข้าไปใน debug (output).....91
รูปที่ 3.33 ปุ่ม Deploy.....92
รูปที่ 3.34 วงจรทางฝั่งตัวควบคุม.....92
รูปที่ 3.35 วงจรทางฝั่งแขนกล.....94
รูปที่ 3.36 กระบวนการการทำงานทางฝั่งตัวควบคุม.....98
รูปที่ 3.37 กระบวนการการทำงานทางฝั่งตัวควบคุม.....99
รูปที่ 3.38 กระบวนการการทำงานทางฝั่งตัวแขนกล.....100
รูปที่ 3.39 หน้าเมนู Raspberry Pi Software Configuration.....101
รูปที่ 3.40 หน้าเมนูกดเลือก Yes เพื่อรีบูท Raspberry.....102
รูปที่ 3.41 ตรวจสอบการเปิดใช้งานไดรเวอร์.....102
รูปที่ 3.42 ผลลัพธ์เมื่อรัน MJPEG Streamer.....103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 3.43 หน้าเว็บเบราเซอร์ MJPEG Streamer....103
รูปที่ 4.1 A) แขนส่วนที่ 1 ต่อเข้ากับมอเตอร์ไฟฟ้า B) แขนส่วนที่ 2 ต่อเข้ากับเซอร์โวมอเตอร์ C) แขนส่วนที่ 3 ต่อเข้ากับเซอร์โวมอเตอร์....104
รูปที่ 4.2 แขนส่วนที่ 3 ต่อเข้ากับเซอร์โวมอเตอร์ และประกอบเข้ากับส่วนตัดและกล้อง....104
รูปที่ 4.3 ส่วนที่ใช้ตัดติดกับเซอร์โวมอเตอร์....105
รูปที่ 4.4 เซอร์โวมอเตอร์ติดกับกรรไกร....105
รูปที่ 4.5 แขนกลประกอบเข้าด้วยกัน....106
รูปที่ 4.6 ตัวควบคุม....106
รูปที่ 4.7 บัดกรีสายบอร์ดที่1....107
รูปที่ 4.8 บัดกรีสายบอร์ดที่2....107
รูปที่ 4.9 ข้อมูลที่ publish ออกไปแสดงทางหน้าต่าง debug บน node-red....108
รูปที่ 4.10 การควบคุมแขนกลด้วยตัวควบคุมและมองภาพผ่านหน้าจอ....109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1ความเป็นมา

ตั้งแต่อดีตจนถึงปัจจุบัน โรงพยาบาลมักมีปัญหาการขาดแคลนบุคคลากรทางการแพทย์ รวมทั้งแพทย์ผู้เชี่ยวชาญเฉพาะทาง โดยเฉพาะในพื้นที่ต่างจังหวัดและห่างไกล เมื่อคนในพื้นที่เจ็บป่วย จึงมีความจำเป็นที่จะต้องเดินทางเข้าเมืองหรือกรุงเทพเพื่อพบหมอเฉพาะทาง เป็นเหตุให้มีความไม่สะดวกในการพบแพทย์ เสียค่าเดินทางมาก หรืออาจทำให้รักษาได้ไม่ทันการ ซึ่งการแก้ไขปัญหาขาดแคลนบุคลากรทางการแพทย์ในระบบสาธารณสุขของประเทศไทยมีมาตลอด และการแก้ไขปัญหาก็เดินหน้าอย่างต่อเนื่อง โดยเฉพาะการผลิตเพิ่ม ทั้งแพทย์ พยาบาล และสาขาวิชาชีพที่ขาดแคลน แต่จนถึงปัจจุบันปัญหาการขาดแคลนยังคงมีอยู่ เรียกได้ว่า เป็นปัญหาด้านการขาดแคลนและการกระจายตัวของบุคลากร

และเนื่องจากในปัจจุบันสังคมทั้งในประเทศไทยและต่างประเทศได้ให้ความสำคัญในเรื่องของสุขภาพ และเทคโนโลยีเพิ่มมากขึ้น รวมทั้งการควบคุมแขนกลด้วยเทคโนโลยีต่าง ๆ ทั้งนี้เทคโนโลยีทางการแพทย์ได้เจริญก้าวหน้า และได้รับการพัฒนาอย่างต่อเนื่องในปัจจุบัน กลุ่มเราจึงได้นำความรู้ ความสามารถทางเทคโลยีเหล่านี้มาพัฒนาต่อยอดเพื่อแก้ปัญหาที่พบเจอดังกล่าว

1.2วัตถุประสงค์ของโครงงาน

1.2.1 จัดทำแขนกลผ่าตัดทางไกลเพื่อให้แพทย์ผู้เชี่ยวชาญหรือเฉพาะทางในเมืองใหญ่เข้าถึงคนไข้ได้โดยไม่ต้องเดินทางไกล

1.2.2 ลดค่าใช้จ่ายในการเดินทางของแพทย์และคนไข้

1.2.3 ศึกษาการส่งค่าผ่านทางระบบอินเตอร์เน็ตเพื่อใช้ในการควบคุมแขนกลทางไกล

1.3ขอบเขตของการวิจัย

นักศึกษาได้นำความรู้พื้นฐาน และศึกษาการทำงานของมอเตอร์ควบคุมต่างๆแล้วนำมาออกแบบ และทำการประยุกต์ ให้เป็นแขนกลควบคุมทางไกล รวมถึงติดกล้องและให้ส่งภาพแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทันทีจากแขนกลกลับมาที่แพทย์ อีกทั้งยังศึกษาการส่งข้อมูลผ่านทางอินเตอร์เน็ตเพื่อควบคุมแขนกลผ่านทางไกลได้

1.3.1 จัดทำแขนกลผ่าตัดโดยใช้มอเตอร์ไฟฟ้าและเซอร์โวมอเตอร์

1.3.2 ส่งค่าผ่านทางอินเตอร์เน็ตเพื่อนำไปควบคุมแขนกลทางไกล

1.3.3 ติดกล้องและรับ-ส่งภาพแบบทันทีและต่อเนื่อง

1.4ระยะเวลา

แผนการดำเนินงาน	2561 (เทอม 1)															
	ส.ค.				ก.ย.				ต.ค.				พ.ย.				ธ.ค.			
ศึกษาความรู้พื้นฐานการควบคุมมอเตอร์ การส่งข้อมูลผ่านทางระบบอินเตอร์เน็ต																				
ออกแบบ และทำการสร้างแขนกล																				
ควบคุมแขนกลให้ขยับได้ตามคำสั่ง																				

ตารางที่ 1.1 ตารางแสดงแผนการดำเนินงานในภาคการศึกษาที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนการดำเนินงาน	2562 (เทอม 2)																			
	ธ.ค.				ม.ค.				ก.พ.				มี.ค.				เม.ย.				พ.ค.			
ติดกล้องเข้ากับแขนกล ทำให้รายงานผลภาพแบบทันทีและตลอดเวลา																								
ควบคุมแขนกลทางไกลผ่านทางระบบอินเตอร์เน็ต																								
ทดลอง และปรับปรุงแก้ไขปัญหาที่พบ																								
จัดทำรายงาน และนำเสนอ																								

ตารางที่ 1.2 ตารางแสดงแผนการดำเนินงานในภาคการศึกษาที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 หุ่นยนต์และแขนกล

หุ่นยนต์(robot) คือ เครื่องจักรกลที่สามารถทำการโปรแกรมซ้ำได้ สามารถทำงานได้หลายๆ หน้าที่ ได้รับการออกแบบมาเพื่อให้สามารถหยิบ จับ ย้าย วัสดุ อุปกรณ์ เครื่องมือหรืออุปกรณ์พิเศษต่าง ๆ โดยการตั้งโปรแกรมควบคุมการเคลื่อนที่ให้ทำงานได้ตามต้องการ

หุ่นยนต์มีหลากหลายประเภทขึ้นอยู่กับวัตถุประสงค์ในการออกแบบและการใช้งาน หากจำแนกหุ่นยนต์ตามลักษณะการเคลื่อนที่ สามารถแบ่งได้เป็น 2 ประเภทใหญ่ๆ คือ

1. หุ่นยนต์ชนิดติดตั้งอยู่กับที่ (Fixed Robot) หุ่นยนต์ประเภทนี้มีลักษณะเป็นแขนกล ซึ่งสามารถเคลื่อนไหวได้เฉพาะข้อต่อในพื้นที่จำกัด ส่วนมากใช้ในโรงงานอุตสาหกรรม
2. หุ่นยนต์ชนิดคลื่อนที่ได้ (Mobile Robot) หุ่นยนต์ประเภทนี้สามารถเคลื่อนที่ได้ด้วยการใช้ล้อ ขา ปีก หรือการเคลื่อนที่ในรูปแบบอื่น ๆ

นอกจากนี้ยังมีหุ่นยนต์อีกหลายประเภทที่หลายหน่วยงานเป็นผู้พัฒนาขึ้น ตัวอย่างเช่น

- หุ่นยนต์ฮิวแมนนอยด์ (Humanoid Robot) หุ่นยนต์ที่มีลักษณะเหมือนกับมนุษย์
- แอนดรอยด์ (Android) หุ่นยนต์ที่สามารถแสดงออกเหมือนมนุษย์
- ไซบอร์ก (Cyborg) เป็นหุ่นยนต์ที่ผสมผสานกันระหว่างสิ่งมีชีวิตกับเครื่องจักร หรือครึ่งคนครึ่งหุ่นยนต์
- นาโนบอท (Nanobot) หุ่นยนต์ที่มีขนาดเล็กมาก ขนาดประมาณ 0.5-3 ไมครอน

ความสามารถในการทำงานของหุ่นยนต์สามารถจำแนกได้ 6 ระดับ ตามเกณฑ์มาตรฐานของสมาคมหุ่นยนต์อุตสาหกรรมแห่งญี่ปุ่น (Japan Industrial Robot Association: JIRA) ดังนี้

ระดับที่ 1 กลไกที่ถูกควบคุมด้วยมนุษย์หรือหุ่นยนต์ที่ควบคุมด้วยมือ(Manual-handling Device) เช่น หุ่นยนต์ม้าแข่งและหุ่นยนต์บังคับด้วยมือ

ระดับที่ 2 หุ่นยนต์ที่ทำงานตามการโปรแกรมหรือขั้นตอนที่กำหนดไว้ แต่ไม่สามารถปรับเปลี่ยนการทำงานได้ (Fixed-sequence Robot) เช่น หุ่นยนต์เดินตามเส้นที่ไม่ใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับที่ 3 หุ่นยนต์ที่ทำงานตามการโปรแกรมและสามารถปรับเปลี่ยนแก้ไขโปรแกรมได้(Variable-sequence Robot) เช่น หุ่นยนต์ที่ควบคุมและประมวลผลข้อมูลด้วยไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์

ระดับที่ 4 หุ่นยนต์ที่มีผู้ควบคุมเป็นผู้สอนงานให้โดยหุ่นยนต์จะทำงานตามหน่วยความจำที่บันทึกไว้(Playback Robot)

ระดับที่ 5 หุ่นยนต์ที่ทำงานตามการบันทึกโปรแกรมข้อมูลเชิงตัวเลข เพื่อกำหนดตำแหน่งการเคลื่อนที่ของหุ่นยนต์ให้ทำงานอัตโนมัติตามข้อมูลตัวเลขที่โปรแกรม (Numerical Control Robot)

ระดับที่ 6 หุ่นยนต์ที่มีความฉลาดสามารถเรียนรู้จากสภาพแวดล้อมและตัดสินใจทำงานได้ด้วย

สำหรับสถาบันหุ่นยนต์แห่งสหรัฐอเมริกา กำหนดให้หุ่นยนต์ต้องมีความสามารถในระดับที่ 3-6 เท่านั้น จึงจะเรียกได้ว่าหุ่นยนต์

2.1.1 ส่วนประกอบของหุ่นยนต์

องค์ประกอบของหุ่นยนต์จะประกอบไปด้วยโครงสร้างของหุ่นยนต์ กลไก ข้อต่อต่างๆ รวมถึงวงจรไฟฟ้า วงจรอิเล็กทรอนิกส์ เซนเซอร์ ไมโครคอนโทรลเลอร์ และโปรแกรมหรือซอฟต์แวร์ ส่วนประกอบของหุ่นยนต์แบ่งออกได้เป็น 5 ส่วนดังนี้

ส่วนที่ 1 โครงสร้างและกลไก

เป็นส่วนประกอบของโครงสร้างทางกายภาพของหุ่นยนต์ ซึ่งประกอบไปด้วย แขน ขา ข้อต่อ ข้อเหวี่ยง ล้อ สายพาน มือจับ เฟือง และกลไกต่างๆ ที่ประกอบขึ้นเป็นตัวหุ่นยนต์ หากเปรียบกับมนุษย์ก็คือ แขน ขา นิ้วมือ กระดูก และข้อต่อต่างๆที่รวมกันเป็นร่างกาย

ส่วนที่ 2 ตัวขับเคลื่อน

ตัวขับเคลื่อนคือส่วนที่ทำให้หุ่นยนต์เกิดการเคลื่อนที่ ส่วนใหญ่เป็นการเคลื่อนที่โดยการหมุนเป็นวงกลม เช่น ดีซีมอเตอร์ เซอร์โวมอเตอร์ สเต็ปเปอร์มอเตอร์ สำหรับตัวขับเคลื่อนที่ทำให้เกิดการเคลื่อนที่แบบเส้นตรง เช่น วาล์วไฟฟ้า กระบอกนิวแมติกและไฮดรอลิก หากเปรียบกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร่างกายมนุษย์ก็คือ กล้ามเนื้อและเส้นเอ็นต่างๆที่ควบคุมการเคลื่อนที่ของร่างกาย

ส่วนที่ 3 เซนเซอร์

เซนเซอร์เป็นอุปกรณ์ที่ใช้ตรวจวัดสัญญาณ ข้อมูล หรือสภาวะแวดล้อมเพื่อส่งให้วงจรควบคุมนำไปประมวลผล เช่น เซนเซอร์ตรวจวัดสี เสียง แสง อุณหภูมิ ตำแหน่งการเคลื่อนที่ และภาพเคลื่อนไหว หากเปรียบกับร่างกายมนุษย์ก็คือ หู ตา จมูก ลิ้น และประสาทสัมผัสต่างๆที่ทำให้เกิดการรับรู้

ส่วนที่ 4 วงจรควบคุมและไมโครคอนโทรลเลอร์

วงจรควบคุมเป็นส่วนที่ควบคุมการทำงานของหุ่นยนต์ ซึ่งประกอบไปด้วยวงจรไฟฟ้า วงจรอิเล็กทรอนิกส์ วงจรขยายสัญญาณ และไมโครคอนโทรลเลอร์

ส่วนที่ 5 โปรแกรมหรือซอฟต์แวร์

โปรแกรมหรือซอฟต์แวร์ทำหน้าที่เสมือนเป็นสมองของหุ่นยนต์ ซึ่งทำหน้าที่ คิด คำนวณ และประมวลผลตามการโปรแกรม โดยส่วนใหญ่จะใช้โปรแกรมภาษาแอสเซมบลี ภาษาเบสิก ภาษาซี หรือภาษาภาพ (Graphics Programming)

2.1.2 แขนกล

แขนกล (Robot Arm) เป็นหุ่นยนต์ที่มีฐานยึดติดกับที่ มักถูกนำมาใช้ในอุตสาหกรรมการผลิตเพื่อใช้แทนแรงงานมนุษย์ในงานที่มีลักษณะซ้ำ ๆ งานอันตราย งานหนัก งานที่น่าเบื่อ หรืองานที่ต้องการความเที่ยงตรงแม่นยำสูง แขนกลอุตสาหกรรมที่ใช้โดยทั่วไป ได้แก่ แขนกลในโรงงานประกอบรถยนต์ แขนกลสำหรับงานเชื่อมและงานประกอบ

โครงสร้างของแขนกลประกอบด้วย ฐานของหุ่นยนต์ ชิ้นส่วนที่เป็นแขนกลหรือส่วนเชื่อมต่อ (Link) ข้อต่อหรือจุดหมุน (Joint) และปลายของแขนกล เช่น มือจับ หัวเจาะ หัวเชื่อม หรือหัวพ่นสี

2.1.2.1 ชนิดของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งชนิดของแขนกลจะแบ่งตามลักษณะรูปทรงของพื้นที่ในการทำงาน ซึ่งแขนกลอุตสาหกรรมจะเป็นแขนกล 3 แกนขึ้นไป เพื่อให้ครอบคลุมพื้นที่ทำงานซึ่งมีลักษณะแตกต่างกันตามสภาพงานจริง สามารถแบ่งชนิดของแขนกลได้ดังนี้

1. แขนกลชนิด Cartesian

แขนกลชนิดนี้จะเคลื่อนที่เป็นเส้นตรง (Prismatic) ทั้ง 3 แกน ข้อดีของแขนกลชนิดนี้ คือ ควบคุมตำแหน่งได้ง่ายเนื่องจากทุกแกนเคลื่อนที่เป็นเส้นตรง เช่น ลักษณะการทำงานของพล็อตเตอร์และเครื่องพิมพ์สามมิติ

2. แขนกลชนิด Cylindrical

แขนกลชนิดนี้มีพื้นที่การทำงานเป็นรูปทรงกระบอก เคลื่อนที่แบบหมุน (Revolute) และแบบเส้นตรงผสมกัน มีข้อดีคือการควบคุมตำแหน่งไม่ซับซ้อนและสามารถทำงานในพื้นที่แคบ ๆหรือลักษณะที่เป็นโพรงได้

3. แขนกลชนิด Spherical

แขนกลชนิดนี้มีพื้นที่การทำงานเป็นรูปทรงกลม เคลื่อนที่ในแนวเส้นตรงและหมุนผสมกัน การควบคุมตำแหน่งและการเคลื่อนที่มีความซับซ้อนขึ้น มีข้อดีคือสามารถหมุนในมุมต่าง ๆได้

4. แขนกลชนิด SCARA

แขนกลชนิด SCARA (Selective Compliance Assembly Robot Arm) การเคลื่อนที่ส่วนฐานและข้อศอกจะเป็นแบบจุดหมุน การเคลื่อนที่ในแนวตั้งจะเป็นการเคลื่อนที่ขึ้นลงในแนวแกน แขนกลชนิด SCARA สามารถเคลื่อนที่ได้เร็วในแนวระนาบและมีความแม่นยำสูง

5. แขนกลชนิด Articulated

แขนกลชนิดนี้ทุกแกนเคลื่อนที่แบบหมุน มีลักษณะการเคลื่อนที่คล้ายกับแขนคน ประกอบด้วยช่วงเอวหรือฐาน แขนท่อนบน แขนท่อนล่าง และมือจับ แขนกลชนิดนี้มีระบบพิกัดที่ซับซ้อน ทำให้ควบคุมการเคลื่อนที่แบบเส้นตรงทำได้ยาก

2.2 การควบคุมแขนกล

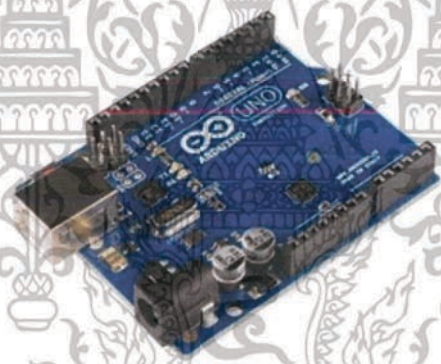
2.2.1 Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

- ส่วนที่เป็นฮาร์ดแวร์ (Hardware)

บอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ เพื่อให้ง่ายต่อการใช้งาน หรือที่เรียกกันว่า บอร์ด Arduino โดยบอร์ด Arduino เองก็มีหลายรุ่นให้เลือกใช้ โดยในแต่ละรุ่นอาจมีความแตกต่างกันในเรื่องของขนาดของบอร์ด หรือสเปก เช่น จำนวนของขารับส่งสัญญาณ, แรงดันไฟที่ใช้, ประสิทธิภาพของ MCU เป็นต้น

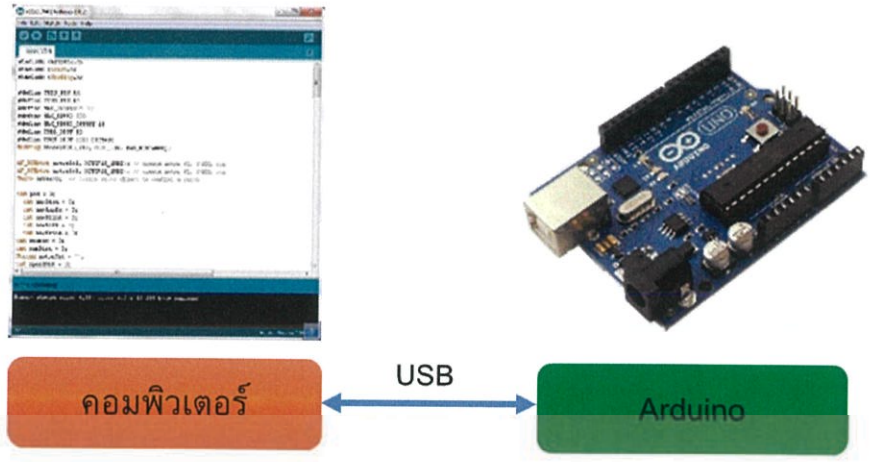


รูปที่ 2.1 Arduino UNO R3

- ส่วนที่เป็นซอฟต์แวร์ (Software)
  - ภาษา Arduino (ซึ่งจริงๆ แล้วก็คือ ภาษา C/C++) ใช้สำหรับเขียนโปรแกรมควบคุม MCU
  - Arduino IDE เป็นเครื่องมือสำหรับเขียนโปรแกรมด้วยภาษา Arduino, คอมไพล์โปรแกรม (Compile) และอัปโหลดโปรแกรมลงบอร์ด (Upload)

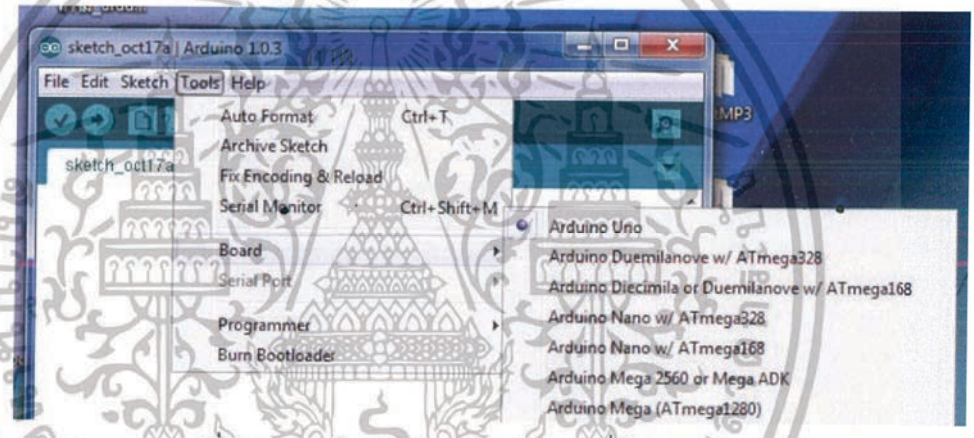
2.2.1.1 รูปแบบการเขียนโปรแกรมบน Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



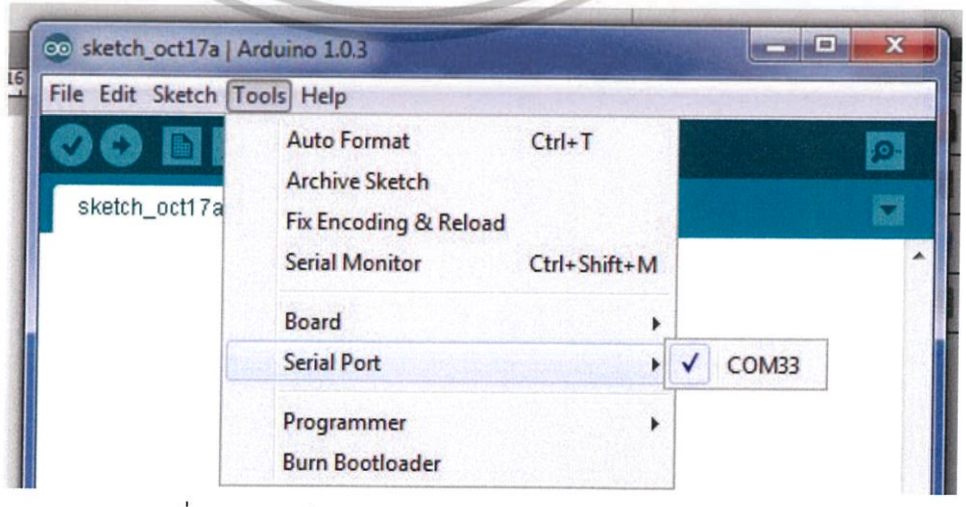
รูปที่ 2.2 รูปแบบการเขียนโปรแกรมบน Arduino

1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE



รูปที่ 2.3 การเลือกรุ่นบอร์ด Arduino ที่ต้องการ upload

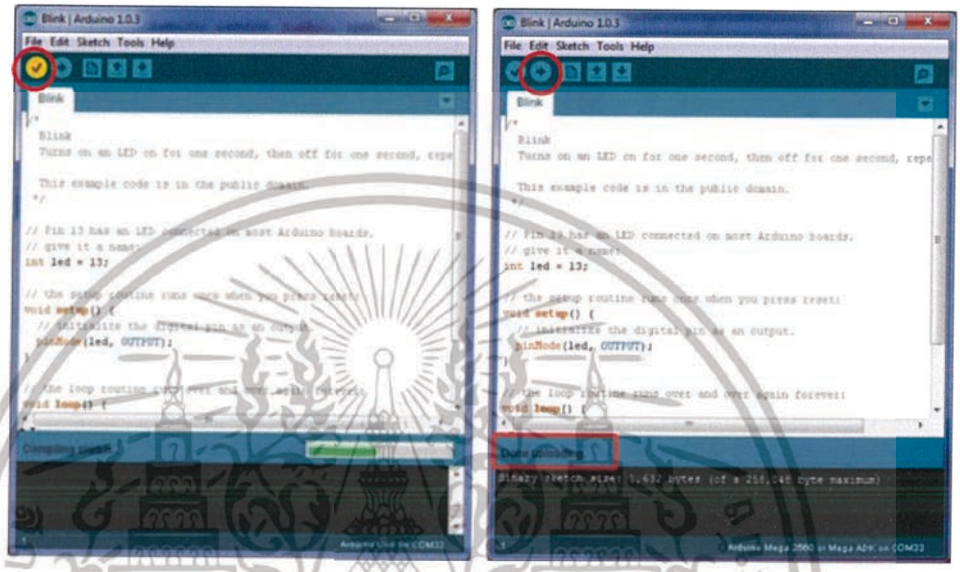
2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port



รูปที่ 2.4 การเลือกหมายเลข Comport ของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออับโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



รูปที่ 2.5 การกดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โค้ดโปรแกรม และการ Upload โค้ดโปรแกรม

2.2.1.2 การใช้งาน Arduino

Arduino ถูกใช้ประโยชน์ในลักษณะเดียวกับ MCU คือ ใช้ติดต่อสื่อสารและควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ ด้วยการเขียนโปรแกรมให้กับ MCU เพื่อควบคุมการรับส่งสัญญาณทางไฟฟ้าตามเงื่อนไขต่าง ๆ

ตัวอย่าง การประยุกต์ใช้ Arduino ในชีวิตประจำวัน เช่น ระบบเปิด/ปิดไฟในบ้านอัตโนมัติ, ระบบรดน้ำต้นไม้อัตโนมัติ, ระบบเปิด/ปิดประตูอัตโนมัติ, ระบบเครื่องซักผ้าหยอดเหรียญ หรือ ใช้ควบคุมความเร็วและทิศทางการหมุนของคุมมอเตอร์ เป็นต้น

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่าง ๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่าง ๆ เช่น Arduino X-Bee Shield, Arduino Music Shield, Arduino Relay

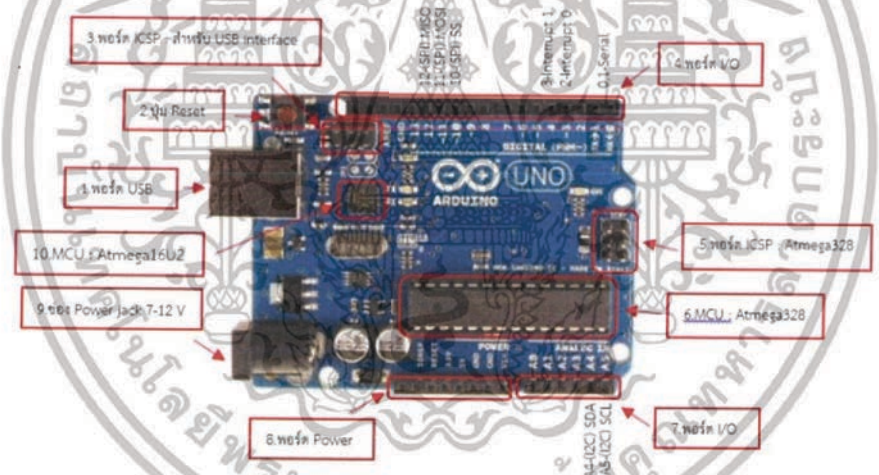
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

2.2.1.3 Layout & Pin out Arduino Board (Model: Arduino UNO R3)



รูปที่ 2.6 Layout & Pin out บอร์ด Arduino

1. USBPort: ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. I/OPort:Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/OPort: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

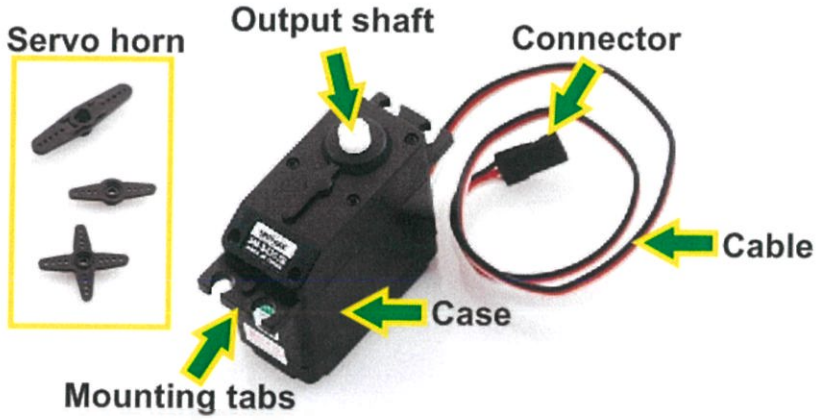
2.2.2 Servo Motor

Servo Motor คือ Motor ที่สามารถสั่งงานหรือตั้งค่าแล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่สั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ซึ่ง Feedback Control คือ ระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่าเท่ากับ หรือ ใกล้เคียงกับค่าอินพุต

ในบทความนี้จะกล่าวถึง RC Servo Motor ซึ่ งนิยมนำมาใช้ในเครื่องเล่นที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

2.2.2.1 ส่วนประกอบภายนอก RC Servo Motor

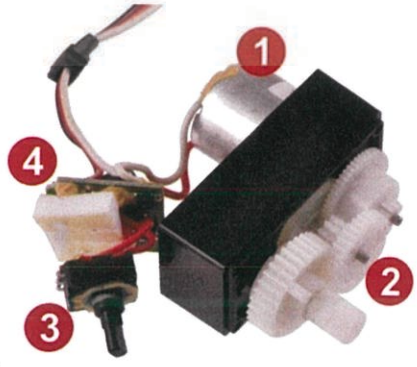
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ส่วนประกอบภายนอก RC Servo Motor

- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เพลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และ ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
  - สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
  - สายสีดำ หรือ น้ำตาล คือ กราวด์
  - สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ

2.2.2.2 ส่วนประกอบภายใน RC Servo Motor

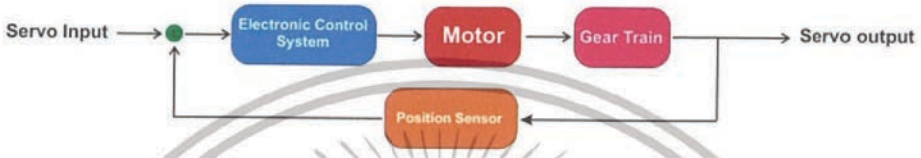


รูปที่ 2.8 ส่วนประกอบภายใน RC Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

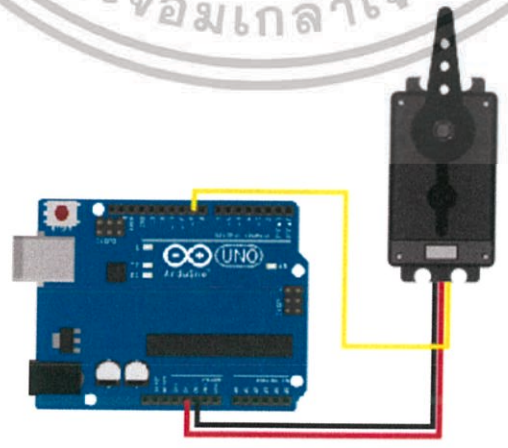
1. Motor เป็นส่วนของตัวมอเตอร์
2. Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
3. Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
4. Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

2.2.2.3 หลักการทำงานของ RC Servo Motor



รูปที่ 2.9 Servo Motor Block Diagram

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุน เป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ



รูปที่ 2.10 ตัวอย่างการเชื่อมต่อ RC Servo Motor เข้ากับบอร์ด Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 มอเตอร์ไฟฟ้า

เป็นอุปกรณ์ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังกล มอเตอร์ที่ใช้งานในปัจจุบันแต่ละชนิดก็จะมีคุณสมบัติที่แตกต่างออกไปต้องการความเร็ว รอบหรือกำลังงานที่แตกต่างกัน ซึ่งมอเตอร์จะแบ่งได้เป็น 2 ชนิด ตามลักษณะการใช้งานกระแสไฟฟ้า

2.2.3.1 มอเตอร์ไฟฟ้ากระแสสลับ (Alternating Current Motor) หรือเรียกว่าเอ.ซี มอเตอร์ (A.C. MOTOR) การแบ่งชนิดของมอเตอร์ไฟฟ้าสลับแบ่งออกเป็น 3 ชนิดได้แก่

2.2.3.1.1 มอเตอร์ไฟฟ้ากระแสสลับชนิด 1 เฟส หรือเรียกว่าซิงเกิลเฟสมอเตอร์ (A.C. Sing Phase) จะใช้กับแรงดันไฟฟ้า 220 โวลต์มีสายไฟ เข้า 2 สาย มีแรงม้าไม่สูง ส่วนใหญ่ตามบ้านเรือน

- สปลิทเฟสมอเตอร์(Split-Phase motor)
- คาปาซิเตอร์มอเตอร์ (Capacitor motor)
- รีพัลชั่นมอเตอร์ (Repulsion-type motor)
- ยูนิเวอร์แวซลมอเตอร์ (Universal motor)
- เช็ดเดดโพลมอเตอร์ (Shaded-pole motor)

2.2.3.1.2 มอเตอร์ไฟฟ้าสลับชนิด 2 เฟสหรือเรียกว่าทูเฟสมอเตอร์ (A.C.Two phas Motor)

2.2.3.1.3 มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟสหรือเรียกว่าทีเฟสมอเตอร์ (A.C. Three phase Motor) เป็นมอเตอร์ที่ใช้ในงานอุตสาหกรรมต้องใช้ระบบไฟฟ้า 3 เฟส ใช้แรงดัน 380 โวลต์ มีสายไฟเข้ามอเตอร์ 3 สาย

2.2.3.2 มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor) หรือเรียกว่าดี.ซี มอเอตร์ (D.C. MOTOR) เป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรม เพราะมีคุณสมบัติที่ดีเด่นในด้านการปรับความเร็วได้ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม เช่นโรงงานทอผ้า โรงงานเส้นใยโพลีเอสเตอร์ โรงงานถลุงโลหะหรือให้ กำลังในการขับเคลื่อนรถไฟฟ้า เป็นต้น มอเตอร์ไฟฟ้ากระแสตรงแบ่งออกเป็น 3 ชนิดได้แก่

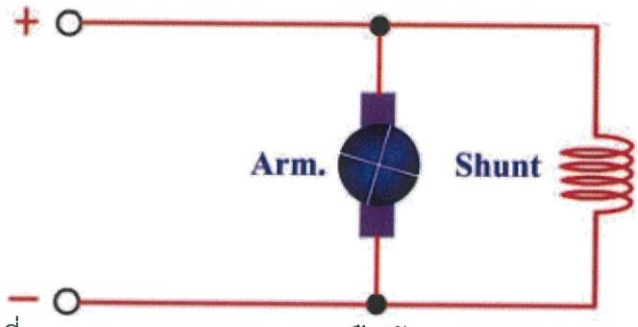
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.2.1 มอเตอร์แบบอนุกรมหรือเรียกว่าซีรีส์มอเตอร์ (Series Motor) คือมอเตอร์ที่ต่อขดลวดสนามแม่เหล็กอนุกรมกับอาร์เมเจอร์ของมอเตอร์ชนิดนี้ว่าซีรีสฟิลด์ (Series Field) มีคุณลักษณะที่ดีคือให้แรงบิดสูง นิยมใช้เป็นต้นกำลังของรถไฟฟ้ารถยกของเครนไฟฟ้าความเร็วรอบของมอเตอร์อนุกรมเมื่อไม่มีโหลดความเร็วจะสูงมาก แต่ถ้ามีโหลดมาต่อความเร็วก็จะลดลงตามโหลดมากหรือทำงานหนักความเร็วลดลงแต่ขดลวดของมอเตอร์ไม่เป็นอันตราย จากคุณสมบัตินี้จึงนิยมนำมาใช้กับเครื่องใช้ไฟฟ้าในบ้านหลายอย่าง เช่น เครื่องดูดฝุ่น เครื่องผสมอาหาร สว่านไฟฟ้า จักรเย็บผ้า เครื่องเป่าผม มอเตอร์กระแสตรงแบบอนุกรมใช้งานหนักได้ดี เมื่อใช้งานหนักกระแสจะมาก ความเร็วรอบจะลดลง เมื่อไม่มีโหลดมาต่อความเร็วจะสูงมากอาจเกิดอันตรายได้ ดังนั้นเมื่อเริ่มสตาร์ทมอเตอร์แบบอนุกรมจึงต้องมีโหลดมาต่ออยู่เสมอ

รูปที่ 2.11 วงจรการทางานของมอเตอร์ไฟฟ้ากระตรงแบบอนุกรม

2.2.3.2.2 มอเตอร์แบบขนานหรือเรียกว่าชันท์มอเตอร์ (Shunt Motor) มอเตอร์แบบขนานนี้ ขดลวดสนามแม่เหล็ก(Field Coil) จะต่อขนานกับขดลวดชุดอาเมเจอร์ มอเตอร์แบบนี้มีคุณลักษณะความเร็วคงที่แรงบิดเริ่มหมุนต่ำ แต่ความเร็วรอบคงที่ ชันท์มอเตอร์ส่วนมากเหมะกับงานพัดลม เพราะพัดลมต้องการความเร็วคงที่และต้องการเปลี่ยนความเร็วได้ง่าย

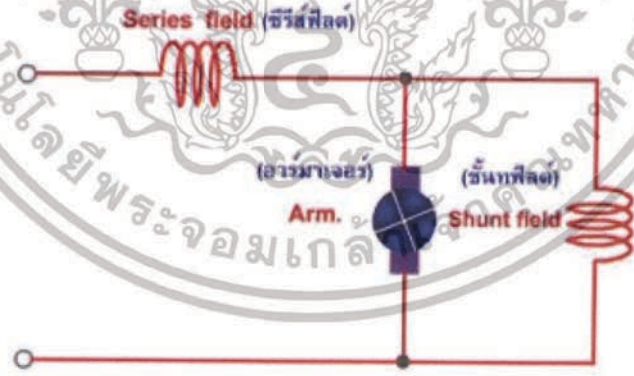
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 วงจรการทางานมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน

2.2.3.2.3 มอเตอร์ไฟฟ้าแบบผสมหรือเรียกว่าคอมเปาวด์มอเตอร์ (Compound Motor) ไฟฟ้ากระแสตรงแบบผสมนี้ จะนำคุณลักษณะที่ดีของมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน และแบบอนุกรมมารวมกัน มอเตอร์แบบผสมมีคุณลักษณะพิเศษคือมีแรงบิดสูง (High staring torque) แต่ความเร็วรอบคงที่ตั้งแต่ยังไม่มีโหลดจนกระทั่งมีโหลดเต็มที่ มอเตอร์แบบผสมมีวิธีการต่อขดลวดขนานหรือขดลวดชันท์อยู่ 2 วิธี

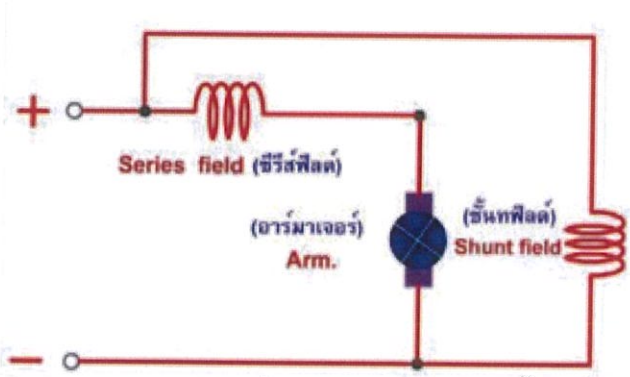
วิธีที่ 1 ใช้ต่อขดลวดแบบชันท์ขนานกับอาเมเจอร์เรียกว่า ชอทชันท์ (Short Shunt Compound Motor) ดังรูปวงจร



รูปที่ 2.13 วงจรการทางานมอเตอร์ไฟฟ้ากระตรงแบบชอร์ทชั้นท์คอมปาว์ดมอเตอร์

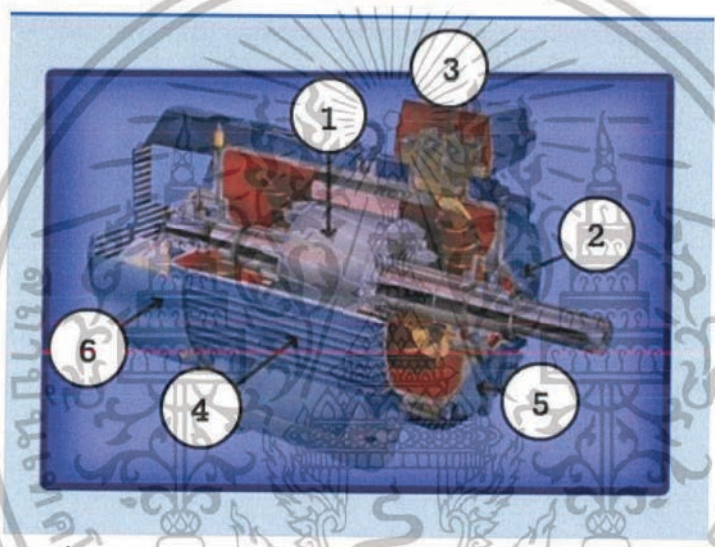
วิธีที่ 2 คือต่อขดลวดขนานกับขดลวดอนุกรมและขดลวดอาเมเจอร์ เรียกว่า ลองชั้นท์คอมปาวด์มอเตอร์(Long shunt motor) ดังรูปวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 วงจรการทางานมอเตอร์ไฟฟ้ากระตรงแบบลองชั้นท์คอมปาว์ดมอเตอร์

2.2.3.3 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้ากระแสตรง



รูปที่ 2.15 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้ากระแสตรง

1. ขดลวดสนามแม่เหล็ก (Field Coil) คือขดลวดที่ถูกพันอยู่กับขั้วแม่เหล็กที่ยึดติดกับโครงมอเตอร์ ทำหน้าที่กำเนิดขั้วแม่เหล็กขั้วเหนือ (N) และขั้วใต้ (S) แทนแม่เหล็กถาวรขดลวดที่ใช้เป็นขดลวดอาบน้ำยาฉนวน สนามแม่เหล็กจะเกิดขึ้นเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์

2. ขั้วแม่เหล็ก (Pole Pieces) คือแกนสำหรับรองรับขดลวดสนามแม่เหล็กถูกยึดติดกับโครงมอเตอร์ด้านใน ขั้วแม่เหล็กทำมาจากแผ่นเหล็กอ่อนบางๆ อัดซ้อนกัน (Lamination Sheet Steel) เพื่อลดการเกิดกระแสไหลวน (Edy Current) ที่จะทำให้ความเข้าของสนามแม่เหล็กลดลง ขั้วแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล็กทำหน้าที่ให้กำเนิดขั้วสนามแม่เหล็กมีความเข้มสูงสุด แทนขั้วสนามแม่เหล็กถาวร ผิวด้านหน้าของขั้วแม่เหล็กทำให้โค้งรับกับอาร์เมเจอร์พอดี

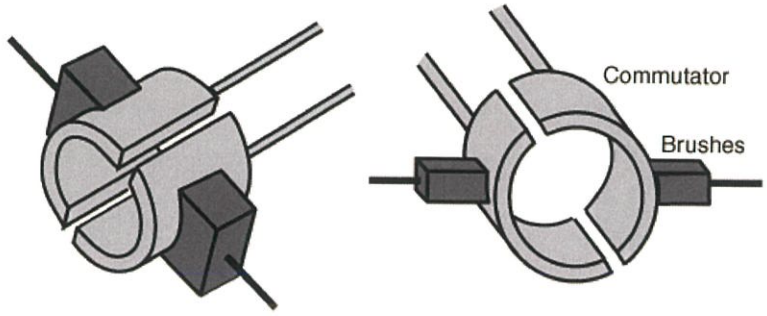
3. โครงมอเตอร์ (Motor Frame) คือส่วนเปลือกหุ้มภายนอกของมอเตอร์ และยึดส่วนอยู่กับที่ (Stator) ของมอเตอร์ไว้ภายในร่วมกับฝาปิดหัวท้ายของมอเตอร์ โครงมอเตอร์ทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กระหว่างขั้วแม่เหล็กให้เกิดสนามแม่เหล็กครบวงจร

4. อาร์เมเจอร์ (Armature) คือส่วนเคลื่อนที่ (Rotor) ถูกยึดติดกับเพลา (Shaft) และรองรับการหมุนด้วยที่รองรับการหมุน (Bearing) ตัวอาร์เมเจอร์ทำจากเหล็กแผ่นบางๆ อัดซ้อนกัน ถูกเซาะร่องออกเป็นส่วนๆ เพื่อไว้พันขดลวดอาร์เมเจอร์ (Armature Winding) ขดลวดอาร์เมเจอร์เป็นขดลวดอาบน้ำยาฉนวน ร่องขดลวดอาร์เมเจอร์จะมีขดลวดพันอยู่และมีลิ่มไฟเบอร์อัดแน่นขัดขดลวดอาร์เมเจอร์ไว้ ปลายขดลวดอาร์เมเจอร์ต่อไว้กับคอมมิวเตเตอร์ อาร์เมอเจอร์ผลักดันของสนามแม่เหล็กทั้งสอง ทำให้อาร์เมเจอร์หมุนเคลื่อนที่

5. คอมมิวเตเตอร์ (Commutator) คือส่วนเคลื่อนที่อีกส่วนหนึ่งถูกยึดติดเข้ากับอาร์เมเจอร์และเพลาร่วมกัน คอมมิวเตเตอร์ทำจากแท่งทองแดงแข็งประกอบเข้าด้วยกันเป็นรูปทรงกระบอก แต่ละแท่งทองแดงของคอมมิวเตเตอร์ถูกแยกออกจากกันด้วยฉนวนไมก้า (Mica) อาร์เมเจอร์คอมมิวเตเตอร์ทำหน้าที่เป็นขั้วรับแรงดันไฟตรงที่จ่ายมาจากแปรงถ่าน เพื่อส่งไปให้ขดลวดอาร์เมอร์

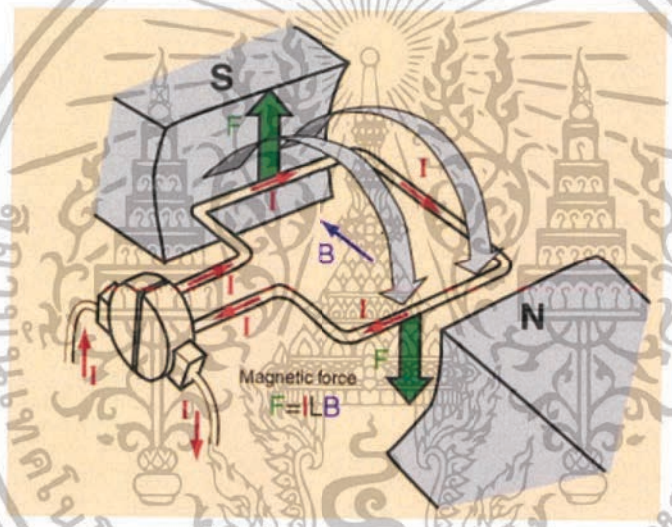
6. แปรงถ่าน (Brush) คือ ตัวสัมผัสกับคอมมิวเตเตอร์ ทำเป็นแท่งสี่เหลี่ยมผลิตมาจากคาร์บอนหรือแกรไฟต์ผสมผงทองแดง เพื่อให้แข็งและนำไฟฟ้าได้ดี มีสายตัวนำต่อร่วมกับแปรงถ่านเพื่อไปรับแรงดันไฟตรงที่จ่ายเข้ามา แปรงถ่านทำหน้าที่รับแรงดันไฟตรงจกแหล่งจ่าย จ่ายผ่านไปให้คอมมิวเตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แปรงถ่าน

2.2.3.4 การทำงานของมอเตอร์



รูปที่ 2.17 หลักการทำงานของ

การทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง มีแรงดันไฟตรงจ่ายผ่านแปรงถ่านไปคอมมิวเตเตอร์ ผ่านไปให้ขดลวดตัวนำที่อาร์เมเจอร์ทำให้ขดลวดอาร์เมเจอร์เกิดสนามแม่เหล็กไฟฟ้าขึ้นมา ทางด้านซ้ายมือเป็นขั้วเหนือ (N) และด้านขวาเป็นขั้วใต้ (S) เหมือนกับขั้วแม่เหล็กถาวรที่วางอยู่ใกล้ๆ เกิดอำนาจแม่เหล็กผลักดันกัน อาร์เมเจอร์หมุนไปในทิศทางตามเข็มนาฬิกา พร้อมกับคอมมิวเตเตอร์หมุนตามไปด้วย แปรงถ่านสัมผัสกับส่วนของคอมมิวเตเตอร์ เปลี่ยนไปในอีกปลายหนึ่งของขดลวด แต่มีผลทำให้เกิดขั้วแม่เหล็กที่อาร์เมเจอร์เหมือนกับขั้วแม่เหล็กถาวรที่อยู่ใกล้ๆอีกครั้ง ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาร์เมเจอร์ยังคงถูกผลักให้หมุนไปในทิศทางตามเข็มนาฬิกาตลอดเวลา เกิดการหมุนของอาร์เมเจอร์คือมอเตอร์ไฟฟ้าทำงาน

2.2.3.5 ข้อดีและข้อเสียระหว่างมอเตอร์ไฟฟ้ากระแสสลับ และมอเตอร์ไฟฟ้ากระแสตรง

ข้อดีของ DC motor คือ

- การควบคุมแรงบิดหรือความเร็วทำได้ง่ายและดีมาก
- มีผลตอบสนองต่อการเปลี่ยนแปลง (response) ได้รวดเร็ว
- การปรับความเร็วสามารถทำได้ในช่วงกว้าง

ข้อเสียของ DC motor คือ

- การบำรุงรักษาสูงมากเนื่องจากมีส่วนสึกหรอของแปรงถ่าน
- ราคาแพงมากเมื่อเทียบกับ AC motor ที่มีขนาดกำลังแรงม้าเท่ากัน
- มีขนาดใหญ่กว่า AC motor ที่ขนาดแรงม้าเท่ากัน
- หาแหล่งจ่ายที่เป็นไฟกระแสตรงได้ยาก
- ไม่สามารถนำไปใช้ในที่มีสารไวไฟได้

ข้อดีของ AC motor

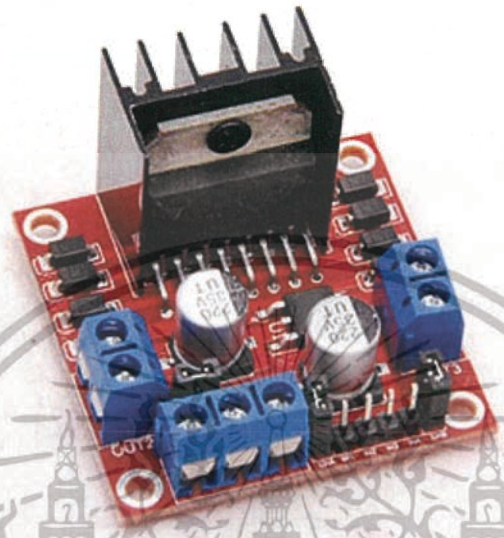
- ราคาถูกกว่า DC motor ที่ขนาดพิกัดกำลังเท่ากันเช่น ที่ 2 แรงม้า AC=4500 บาท ,DC=20000 บาท
- มีลักษณะโครงสร้างง่าย ไม่ซับซ้อน และเล็กกว่า DC motor ที่พิกัดเท่ากัน
- การบำรุงรักษาน้อยมาก แข็งแรงทนทาน
- ใช้ในสถานที่ที่มีสารไวไฟ หรือสารเคมีได้
- มีประสิทธิภาพสูงกว่า DC motor
- หาซื้อได้ง่าย เป็นที่นิยม

ข้อเสียของ AC motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การควบคุมความเร็วทำได้ยากมาก จะต้องใช้อุปกรณ์ทาง power electronics มาควบคุมคือ inverter ซึ่งค่อนข้างจะมีราคาแพง

2.2.4 Motor Drive Module L298N



รูปที่ 2.18 Motor Drive Module L298N

เมื่อพูดถึงการใช้งานมอเตอร์กระแสตรงแล้ว หลีกเลี่ยงไม่ได้ที่จะนึกถึง หุ่นยนต์ หรือแม้กระทั่งรถบังคับ ส่วนใหญ่จะใช้มอเตอร์เป็นตัวขับเคลื่อนการทำงาน เช่น การเคลื่อนที่ของรถ บังคับ หรือ แม้กระทั่งใช้เป็นกลไกในการทำงานของของหุ่นยนต์ เมื่อพูดถึงมอเตอร์ การที่จะต้องใช้งานมอเตอร์ทำงานอย่างมีประสิทธิภาพทั้งการควบคุมความเร็วรอบของมอเตอร์ และการควบคุมการหมุนทวนเข็มและตามเข็ม สิ่งเหล่านี้ทำได้ แต่ต้องมีโมดูลเสริมหรือตัวกำหนดตัวแปรที่จะทำให้สามารถควบคุมการทำงานของมอเตอร์ได้ ซึ่งโมดูลที่ใช้ได้มีด้วยกันหลายรุ่น ในที่นี้จะขอยกตัวอย่างเอารุ่น L298N motor drive มาเป็นกรณีศึกษา

L298N เป็นชุดขับมอเตอร์ชนิด H-Bridge ซึ่งส่วนใหญ่จะถูกนำไปใช้ในการควบคุมทิศทางและความเร็วของมอเตอร์ ซึ่งสามารถควบคุมมอเตอร์ได้ทั้งหมด 2 Channel

หลักการทำงาน

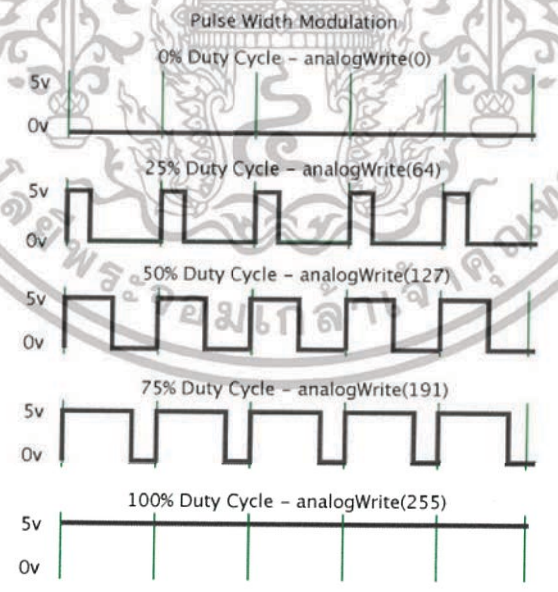
วงจร H-Bridge ของ L298N จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ PWM (Pulse Width Modulation) ซึ่งPWM หมายถึง การควบคุมช่วงจังหวะการทำงานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิเล็กตรอน หากจินตนาการถึงแปรงขดลวดในมอเตอร์เป็นระหัดวิดน้ำและอิเล็กตรอนเป็นน้ำที่ตกลงมาจากระหัดวิดน้ำ ค่าแรงดันไฟฟ้าก็คล้ายกับกระแสน้ำที่ไหลผ่านระหัดวิดน้ำด้วยความเร็วคงที่ ยิ่งกระแสน้ำไหลเร็วเท่าไรก็จะหมายความว่าแรงดันไฟฟ้ายิ่งสูงขึ้น แต่มอเตอร์มีอัตราความเร็วคงที่และสามารถเสียหายได้หากมีแรงดันไฟฟ้าสูงไหลผ่านหรือหยุดทันทีเพื่อที่จะหยุดมอเตอร์ ดังนั้น PWM คล้ายกับการควบคุมระหัดวิดน้ำให้ตักน้ำในจังหวะคงที่ที่กระแสน้ำคงที่ ยิ่งระหัดวิดน้ำหมุนเร็วเท่าไรช่วงของ pulse ก็จะยาวขึ้น ในทางกลับกันถ้าระหัดวิดน้ำหมุนช้าช่วงของ pulse จะสั้นลง ดังนั้นเพื่อยืดอายุการใช้งานของมอเตอร์จึงควรที่จะควบคุมมอเตอร์ด้วย PWM

2.2.4.1 การควบคุมมอเตอร์ด้วย Pulse Width Modulation

Pulse Width Modulation (PWM) เป็นเทคนิคการส่งสัญญาณแบบสวิต หรือ ส่งค่าดิจิตอล 0-1 โดยให้สัญญาณความถี่คงที่ การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำ ที่ต่างกัน ก็จะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิต ต่างกันด้วย

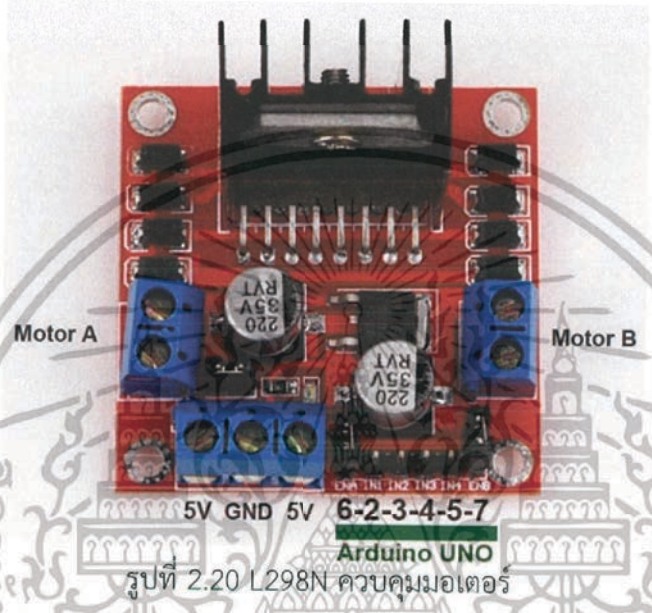


รูปที่ 2.19 Pulse Width Modulation ที่ 0%, 25%, 50%, 75%, 100% Duty Cycle มอเตอร์

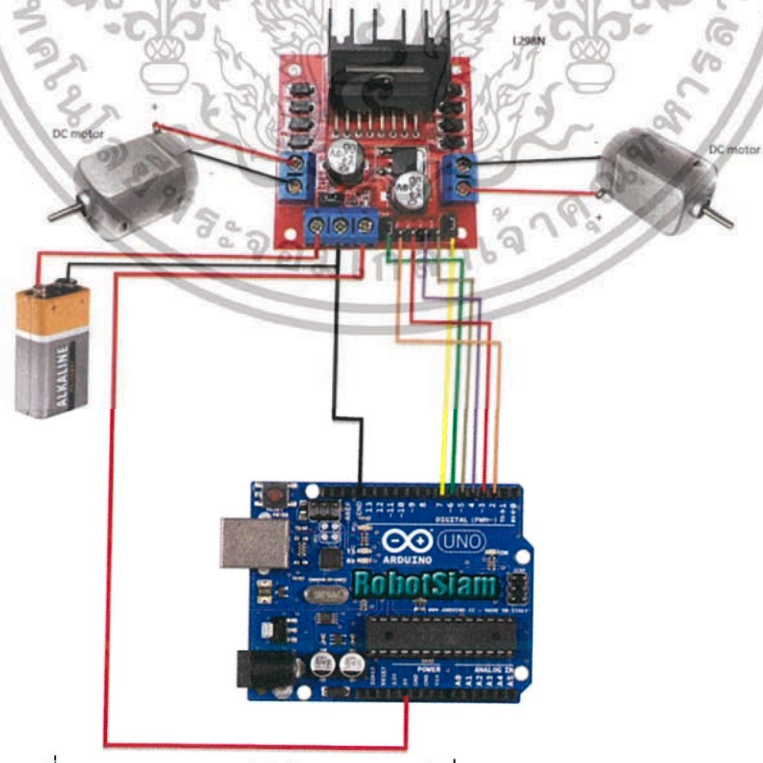
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโมดูล PWM ของ Arduino มีความละเอียด 8 bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณแบบดิจิตอล จะได้ 0 ถึง 255 ซึ่งเราสามารถเทียบสัดส่วนคำนวนจากเลขจริงเป็นเลขทางดิจิตอลได้

2.2.5 การใช้ Arduino UNO R3 กับ L298N ควบคุมมอเตอร์



รูปที่ 2.20 L298N ควบคุมมอเตอร์



รูปที่ 2.21 แสดงการต่อใช้งานอุปกรณ์เพื่อควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการต่อกับ Arduino นั้น ขา IN1,IN2,IN3 และ IN4 นั้น สามารถต่อกับพอร์ตDigital ใดๆก็ได้ เนื่องจาก 4 ขานี้ จะใช้ในการควบคุมสัญญาณลอจิกบอกทิศทางให้กับมอเตอร์ ส่วน ENA และ ENB นั้น จำเป็นที่จะต้องต่อกับพอร์ต Digital ที่รองรับ PWMเนื่องจากจะต้องใช้สัญญาณ PWM ในการควบคุมความเร็วของมอเตอร์

2.2.5.1 โครงสร้างcodeคร่าวๆของ Arduino

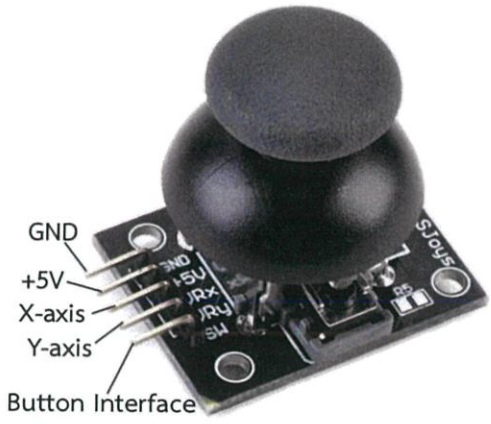
codeการทำงานของ Arduino มีการพัฒนาขึ้นเรื่อยๆ แต่ยังไม่มีสร้าง library ที่เกี่ยวกับตัว L298N Dual H-Bridge เพื่อควบคุมมอเตอร์ ดังนั้นผู้ใช้งานจึงต้องประกาศpinเพื่อใช้งานขึ้นมาเอง สามารถใช้โค้ด int dir(number)Pin(letter) ต่อเข้ากับpinดิจิตอลที่เลือกใช้ แค่นี้ก็สามารถทำงานได้อย่างถูกต้องและช่วยให้ตัว L298N Dual H-Bridge ควบคุมมอเตอร์ได้อย่างอเนกประสงค์ ถ้าบอร์ด Arduino ที่เลือกใช้งานมีpinหลายตัว และถ้าต้องการปรับความเร็วของมอเตอร์ด้วย PWM สามารถใช้คำสั่ง int speedPin(letter) แล้วต่อเข้ากับpinที่เลือกใช้ หากต้องการวิธีลัดเพื่อใช้งาน PWM อย่างรวดเร็วสามารถเลือกpinใช้งานได้ตามรายการด้านล่าง

- AT MEGA –PWM 2-13 และ 44-46 ตั้งค่าเอาท์พุทของ PWM ให้เป็น 8 bit ด้วยฟังก์ชั่น analogWrite()
- UNO-PWM 3, 5, 6, 9, 10 และ 11 ตั้งค่าเอาท์พุทของ PWM ให้เป็น 8 bit ด้วยฟังก์ชั่น analogWrite()

2.2.6 Joystick Module

อุปกรณ์ X/Y Joystick (หรือ Thumb Joystick) มีคันโยกแบบสองแกน โดยมีตัวต้านทานปรับค่าได้สองตัวสำหรับแต่ละแกน ถ้านำไปต่อกับแรงดันไฟเลี้ยง +5V และ GND ก็จะใช้งานเป็นวงจรแบ่งแรงดัน (Voltage Divider) และค่าของสัญญาณหรือระดับแรงดัน จะเปลี่ยนตามตำแหน่งของคันโยก ถ้านำไปต่อกับขาแอนะล็อก-อินพุต (Analog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 Joystick Module

Input) ของไมโครคอนโทรลเลอร์ เช่น อาดูโน่ ก็สามารถระบุตำแหน่งหรือมุมของคันโยกในแต่ละแกนได้

2.2.6.1 ขาต่างๆของ XY Joystick Module

GND -- ขา Ground (นำไปต่อกับขา GND ของอาดูโน่)

+5V -- ขาแรงดันไฟเลี้ยง (นำไปต่อกับขา 5V ของอาดูโน่)

VRx -- ขาสัญญาณแอนะล็อกสำหรับแกน X (นำไปต่อกับขาแอนะล็อก-อินพุตของอาดูโน่)

VRy -- ขาสัญญาณแอนะล็อกสำหรับแกน Y (นำไปต่อกับขาแอนะล็อก-อินพุตของอาดูโน่

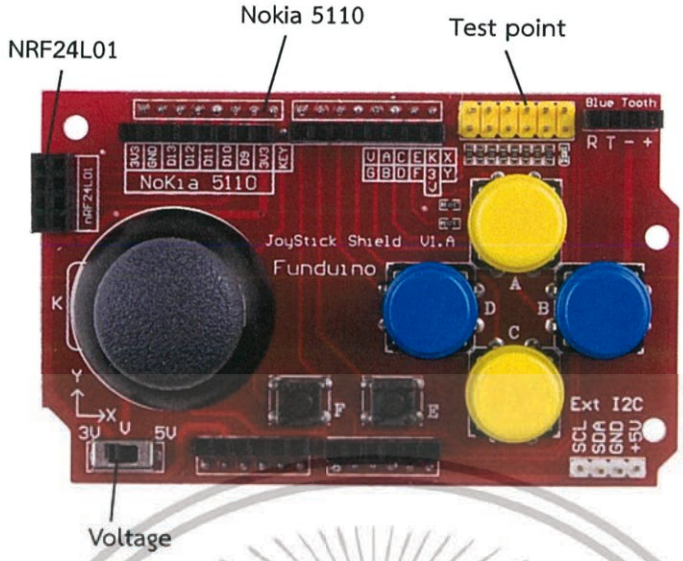
SW -- ขาสัญญาณดิจิทัลจากปุ่มกดแบบ Active-low ต้องต่อตัวต้านทานแบบ Pull-up เพิ่ม

เช่น 4.7k หรือ 10k โอห์ม (หรือบัดกรีลงบนแผงวงจรได้ ตรงจุดที่ทำเป็น pads ไว้ แต่ต้องเป็นอุปกรณ์แบบ SMD)

2.2.7 *JoyStick Shield expansion board for Arduino*

จอยสติกสำหรับเสียบลงไปบนอาดูโน่ เพื่อใช้อาดูโน่ควบคุม พร้อมช่องรองรับการเชื่อมต่อแบบไร้สาย NRF24L01 และจอแสดงผล Nokia 5110 LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 JoyStick Shield expansion board

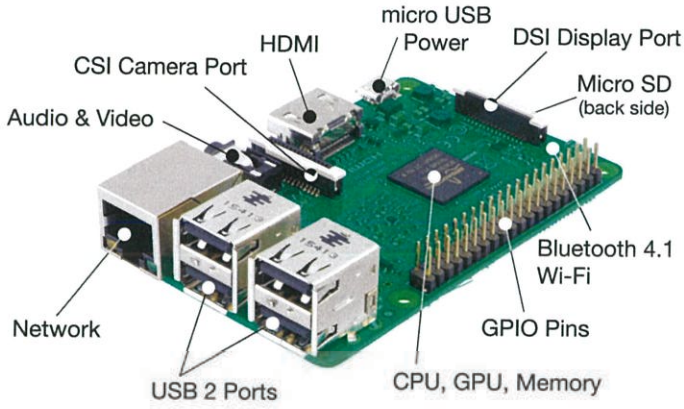
2.3 ส่วนควบคุมแขนกลทางไกล

2.3.1 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B ใหม่จาก Raspberry Pi Foundation เป็นผลิตภัณฑ์รุ่นล่าสุดที่รวมเอาหน่วยประมวลผล ARM Cortex-A53 สี่คอร์แบบ 64 บิตเพื่อมอบประสิทธิภาพที่ได้รับการปรับปรุงพัฒนาอย่างมีนัยสำคัญ ตลอดจนนำเสนอการเชื่อมต่อ Bluetooth และ LAN ไร้สายที่เป็นมาตรฐานของโครงการพัฒนา Internet of Things (IoT) โดยที่มีเป้าหมายเพื่อใช้ในอุตสาหกรรมในบ้าน และในโรงเรียนและผู้ร่วมงานทั่วโลก

Raspberry Pi 3 บูทจากไมโคร SD การ์ด และใช้ตัวจัดการการติดตั้ง NOOBS (New Out Of the Box Software) ระบบปฏิบัติการ Raspbian มาตรฐานที่ติดตั้งมาพร้อมกับแอปพลิเคชันการผลิต และเครื่องมือด้านโปรแกรมต่าง ๆ รวมถึง Node-RED เครื่องมือนี้รองรับสำหรับการเชื่อมอุปกรณ์ฮาร์ดแวร์ API และบริการออนไลน์เข้าด้วยกัน ทำให้บอร์ดมีความเหมาะสมสำหรับการพัฒนาและสร้างต้นแบบโครงการ IoT อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ส่วนประกอบของ Raspberry Pi 3

2.3.1.1 ส่วนประกอบและคุณสมบัติของ Raspberry Pi

จุดเชื่อมต่อ GPIO

- เชื่อมต่อกับบอร์ด Raspberry Pi ผ่านทางคอนเน็กเตอร์ GPIO 40 ขา
- มีจุดต่อพอร์ต GPIO 40 ขาครบทุกขาเป็นแบบ IDC ตัวผู้ระยะห่างระหว่างขา 2.54 มม.
- มีจุดต่อแบบ JST 2 มม. 3 ขา จำนวน 10 ขา ประกอบด้วย GPIO2/SDA, 3/SCL, 4/1-Wire, 5, 14/TxD, 15/RxD, 24, 25, 26 และ 27
- มีสวิตช์กดติดปล่อยดับพร้อมใช้งานต่อกับขา GPIO13
- มีสวิตช์แบบจอยสติ๊ก 4 ทิศทางและปุ่มกดกึ่งกลาง ต่อกับขา GPIO6, 17, 18, 22 และ 23
- มีจอแสดงผล OLED 0.96 นิ้ว ความละเอียด 128x64 จุด ใช้การเชื่อมต่อแบบบัส I2C
- มี LED 3 สี RGB แบบอนุกรมและโปรแกรมได้ เบอร์ WS2812B ต่อกับขาพอร์ต GPIO12
- ขนาด 65 x 56 มม. โดยออกแบบรูปร่างของแผ่นวงจรพิมพ์ ให้มีขนาดระยะห่างของรูยึดช่องว่างต่างๆ เป็นไปตามข้อกำหนดของการออกแบบบอร์ด HAT ของ Raspberry Pi Foundation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดเชื่อมต่อ Port USB

- ใช้สำหรับหรือช่องทางในการเชื่อมต่อและสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกอื่นๆ

ช่อง Audio/Vidio

- Audio Output อยู่ 2 ช่องด้วยกันคือผ่านทางช่อง HDMI และแจ๊ค 3.5mm ที่เป็นทั้ง Stereo Audio และ Composite Video โดยบอร์ด Raspberry Pi จะเลือกให้โดย อัตโนมัติว่าจะใช้งานช่องไหน

จุดเชื่อมต่อ LAN/ Internet

- ในการเชื่อมต่อ Internet

จุดต่อกล้องแบบ CSI (Camera Serail interface)

- พอร์ต CSI (Camera Serial Interface) ใช้สำหรับเชื่อมต่อโมดูลกล้อง

จุดเชื่อมต่อ HDMI

- ใช้สำหรับเชื่อมต่อกับจอแสดงผล หากเลือกใช้จอ Monitor ที่ไม่มีพอร์ต HDMI รองรับต้องใช้ตัวแปลง HDMI to VGA ด้วย หรือเชื่อมต่อสายวีดีโอ RCA ก็ได้เช่นเดียวกัน (เลือกอย่างใดอย่างหนึ่ง)

ช่องจ่ายไฟด้วย Micro USB Power

- ทำหน้าที่เพื่อจ่ายไฟเลี้ยงให้กับวงจร สามารถเลือกใช้แหล่งจ่ายไฟจากพอร์ต USB ของเครื่องคอมพิวเตอร์ได้

ช่องเสียบ SD card

- ตัวนี้จะอยู่ด้านหลังของบอร์ดใช้สำหรับติดตั้งระบบปฏิบัติการ Linux ต้องมีความจุมากกว่า 2GB ขึ้นไป แต่แนะนำให้ใช้ ขนาด 4GB หรือมากกว่า

พอร์ต DSI (Display Serial Interface)

- ใช้สำหรับต่อจอแสดงผล เช่น จอแสดงผลแบบ TFT Touch Screen เป็นต้น

ชิป Bluetooth

2.3.2 Raspberry Pi V2 Camera Board

เป็นโมดูลกล้องที่ออกแบบมาใช้งานร่วมกับบอร์ด Raspberry Pi โดย สามารถเชื่อมต่อกับซ๊อกเก็ต CAMERA บนบอร์ด ที่มีการเชื่อมต่อแบบ CSI bus ได้ทันที และสามารถเชื่อมต่อกับบอร์ด Raspberry Pi ได้ทุกรุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Raspberry Pi Camera Board ใช้เซ็นเซอร์รับภาพ IMX219PQ ของโซนี่ซึ่งช่วยประมวลผลภาพวิดีโอความเร็วสูงและมีความไวสูง และยังเหมาะสำหรับงานวิดีโอความละเอียดสูงและการถ่ายภาพนิ่ง สามารถถ่ายภาพแบบ time-lapse และ slow-motion นอกจากนี้ยังมีฟังก์ชั่นการควบคุมอัตโนมัติเช่นการควบคุมการเปิดรับความสมดุลสีแสงขาว (white balance) และการตรวจจับความสว่างอีกด้วย

2.3.2.1 คุณสมบัติของ Raspberry Pi V2 Camera Board

รูปที่ 2.25 Raspberry Pi Camera

- ความละเอียดสูง 8 ล้านพิกเซล
- ถ่ายวิดีโอคุณภาพระดับ HD ความคมชัด 1080p, 720p และ 640x480 ด้วยอัตราแสดงผล 30 (1080p), 60 (720p และ 640x480) และ 90 (640x480) เฟรมต่อวินาที
- เซ็นเซอร์รับแสงจาก Sony IMX219PQ CMOS image sensor
- ความละเอียดสูงสุดที่สนับสนุน 3280 x 2464
- อัตราเฟรมสูงสุดในการจับภาพ 30 เฟรมต่อวินาที
- อุณหภูมิในการทำงานสูงสุด +60oC
- อุณหภูมิในการทำงานต่ำสุด -20oC
- ขนาด 23.86 x 25 x 9 มม.
- น้ำหนัก 3 กรัม
- ต่อกับบอร์ด Raspberry Pi ด้วยบัส CSI (Common System Interface) ซึ่งเป็นการเชื่อมต่อแบบ point-to-point บัสนี้พัฒนาโดย Intel ออกแบบมาเพื่อ

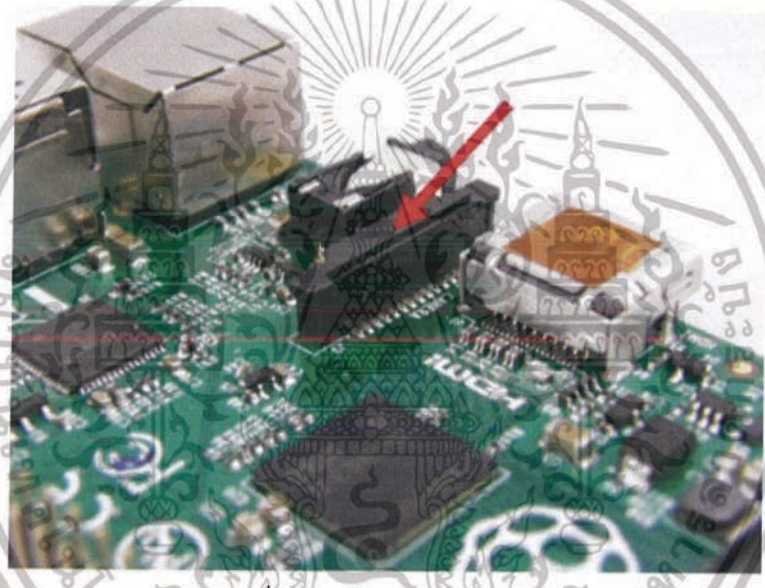
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลความเร็วสูง 12 ถึง 16 GB/s ด้วยการใช้เทคนิค low-voltage differential signaling

- ใช้กระแสไฟฟ้าต่ำเหมาะกับอุปกรณ์กล้องที่ต้องถ่ายข้อมูลจำนวนมากอย่างรวดเร็ว
- ใช้กับบอร์ด Raspberry Pi ที่ติดตั้งอิมเมจเวอร์ชั่นล่าสุด
- สายแพ 15 จุดต่อ ยาว 15CM.

2.3.2.2 วิธีการต่อใช้งาน

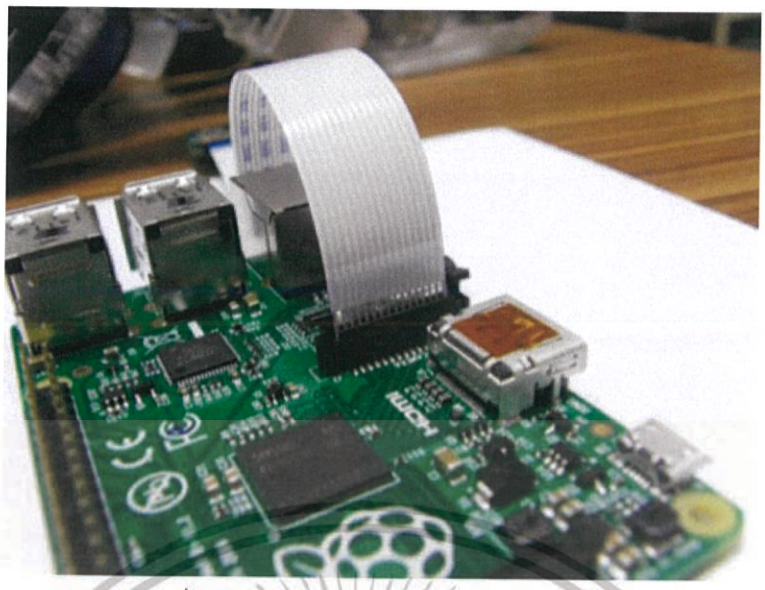
2.3.2.2.1 การเชื่อมต่อโมดูลกล้องจะต้องเชื่อมต่อที่ซ็อกเก็ต CAMERA ดังรูป



รูปที่ 2.26 ซ็อกเก็ต CAMERA

2.3.2.2.2 ให้ทำการดึงซ็อกเก็ต CAMERA ขึ้นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



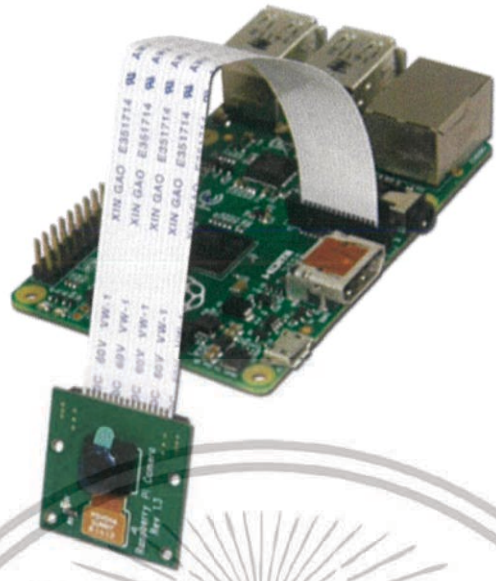
รูปที่ 2.27 ต่อสายแพกับซ็อกเก็ต CAMERA

2.3.2.2.3 ทำการเสียบสายแพโดยหันด้านที่เป็นสายสัญญาณออก ดังรูป

รูปที่ 2.28 ดึงซ็อกเก็ต CAMERA

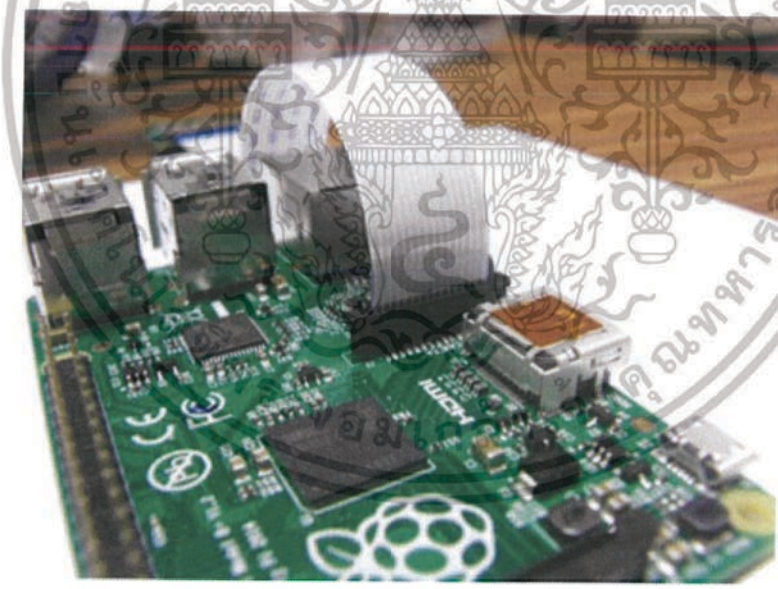
2.3.2.2.4 จากนั้นให้กดซ็อกเก็ต CAMERA เพื่อล็อคสายสัญญาณ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 สายแพที่ต่อกับ Raspberry pi

2.3.2.2.5 เมื่อติดตั้งเสร็จเรียบร้อยแล้วจะได้ ดังรูป



รูปที่ 2.30 สายแพที่ต่อกับซ็อกเก็ต CAMERA

2.3.2.3 การทดสอบ Raspberry Pi Camera Board เบื้องต้น

2.3.2.3.1 ให้ทำการเชื่อมจ่ายไฟเข้าบอร์ด Raspberry Pi และเชื่อมต่ออินเตอร์เน็ตให้เรียบร้อยจากนั้นให้ทำการอัพเดท เวอร์ชั่นของระบบปฏิบัติการเป็นเวอร์ชั่นล่าสุดก่อนโดยใช้คำสั่ง

sudo apt-get update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

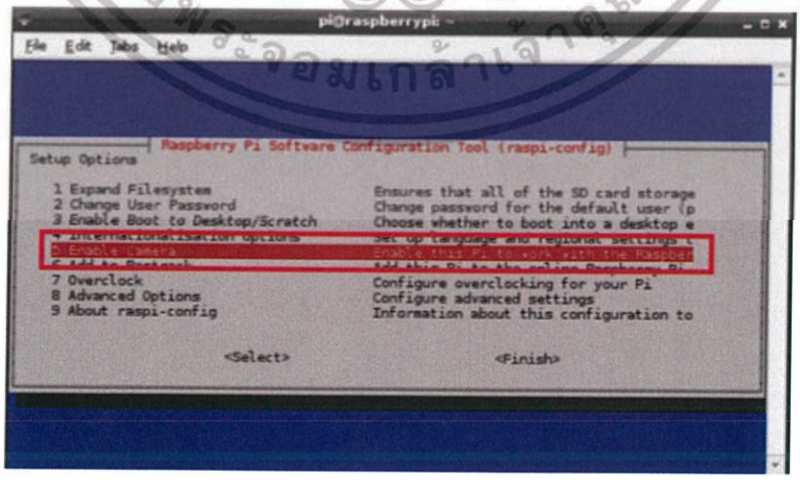
sudo apt-get upgrade

รูปที่ 2.31 หน้า Command ของ Raspberry pi

2.3.2.3.2 จากนั้นให้ใช้คำสั่ง sudo raspi-config เพื่อเข้าเมนูการตั้งค่า ดังรูป

รูปที่ 2.32 ใช้คำสั่ง sudo raspi-config

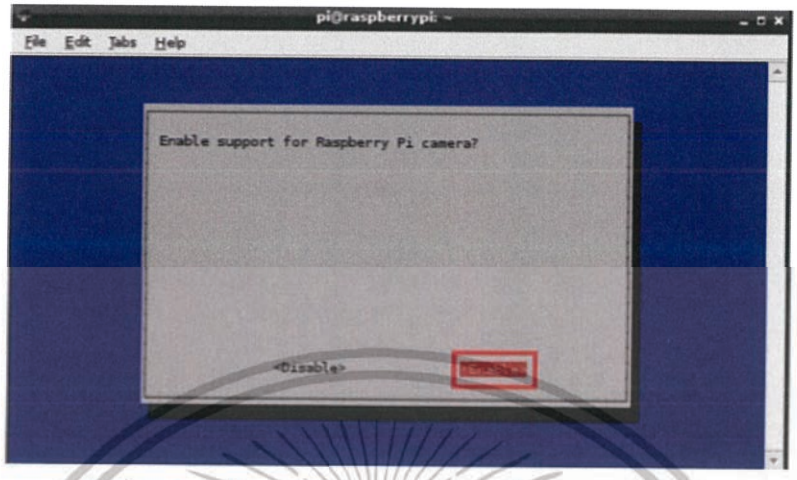
2.3.2.3.3 เลือกที่เมนู Enable Camera ดังรูป



รูปที่ 2.33 หน้าเมนู Raspberry Pi Software Configuration Tool

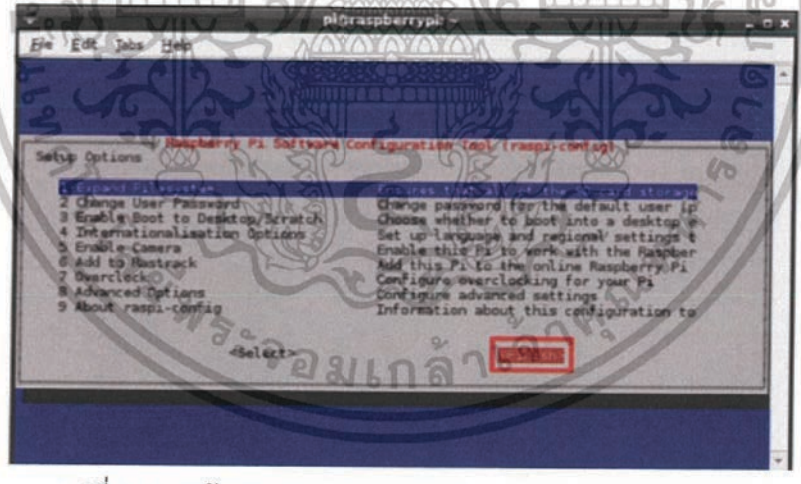
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.3.4 เลือก Enable เพื่อเปิดการใช้งานโมดูลกล้อง และกด Enter ดังรูป



รูปที่ 2.34 หน้าเมนูกดเลือก Enable เพื่อเปิดใช้งานกล้อง

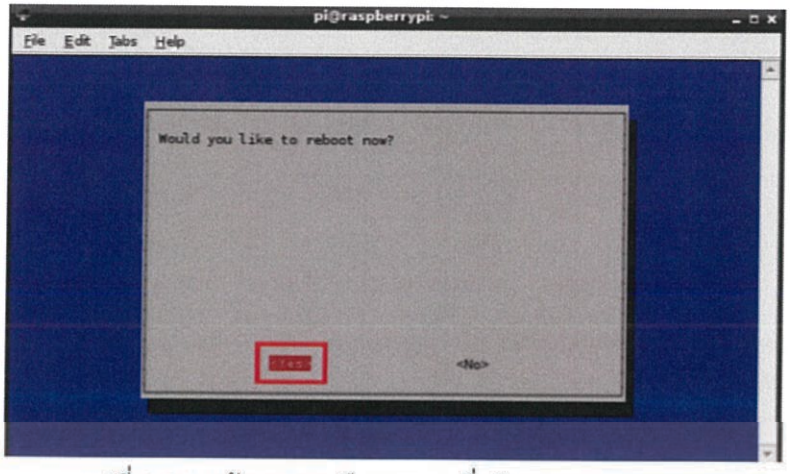
2.3.2.3.5 กดปุ่ม Tab บนคีย์บอร์ดเพื่อเลื่อนแถบสีแดงมาที่ Finish จากนั้นกดปุ่ม Enter ดังรูป



รูปที่ 2.35 หน้าเมนู Raspberry Pi Software Configuration Tool

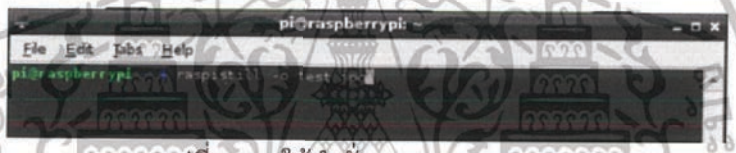
2.3.2.3.6 เลือก Yes เพื่อรีบูทบอร์ด Raspberry Pi ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 หน้าเมนูกดเลือก Yes เพื่อรีบูท Raspberry Pi

2.3.2.3.7 หลังจากที่ทำการรีบูทเสร็จเรียบร้อยให้ทดลองใช้คาสั่ง raspistill เพื่อทำการถ่ายภาพนิ่ง โดยตัวอย่างนี้จะทดลองถ่ายภาพนิ่งชื่อ test.jpg โดยใช้คำสั่ง raspistill -o test.jpg



รูปที่ 2.37 ใช้คำสั่ง sudo -o test.jpg

2.3.2.3.8 จากนั้นให้ไปที่ /home/pi จะเห็นไฟล์test.jpgดังรูป

รูปที่ 2.38 หน้าต่างโฟลเดอร์ /home/pi

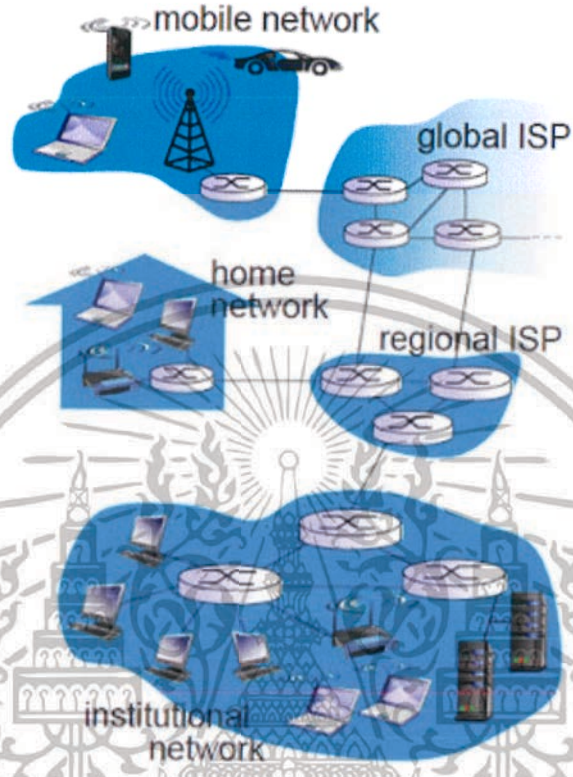
2.3.3 การทำ Web server เพื่อรับข้อมูลจาก Sensor และควบคุมอุปกรณ์ต่างๆ

2.3.3.1 การส่งข้อมูลผ่านหน้าเว็บ

ทุกวันนี้เวลาที่เราจะส่งและรับข้อมูลผ่านอินเตอร์เน็ตหรือเปิดเว็บไซต์อะไรก็ตาม อุปกรณ์ที่เราใช้งานนั้นต้องมีการเชื่อมต่ออินเตอร์เน็ต ไม่ว่าจะเป็นการเชื่อมต่อผ่านสาย LAN , Wi-Fi หรือผู้ให้บริการอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เราเรียกกันย่อๆ ว่า ISP (Internet Service Provider) อย่าง CAT, true, 3BB หรือ ToT ซึ่งมีโครงสร้างและรูปแบบการเชื่อมต่อเครือข่ายดังแสดงใน



รูปที่ 2.39 โครงสร้างเครือข่ายคอมพิวเตอร์

ในการส่งและรับข้อมูลระหว่างเครือข่ายคอมพิวเตอร์ และอุปกรณ์ของเรา (ไคลเอนต์) จะมีบราวเซอร์ทำหน้าที่ร้องขอและรับข้อมูล (โดยใช้ HTTP Protocol) จากเซิฟเวอร์มาแสดงผล ซึ่งในส่วนของเซิฟเวอร์เองก็จะรอรับการร้องขอแล้วส่งข้อมูล (โดยใช้ HTTP Protocol) ที่ถูกร้องขอกลับไป ดังแสดงในรูปที่ 2.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.40 รูปแบบการส่งและรับข้อมูลระหว่าง ไคลเอนต์ และ เซิฟเวอร์

ข้อความ HTTP มีสองชนิด คือ request (ข้อความขอบริการ), response (ข้อความตอบกลับ) และในการส่งข้อมูลไปยังเซิฟเวอร์ ใช้วิธีใส่ข้อมูลลงไปในข้อความขอบริการ (request) ซึ่งเป็นเทคนิคหนึ่งที่ใช้ในการส่งข้อมูล

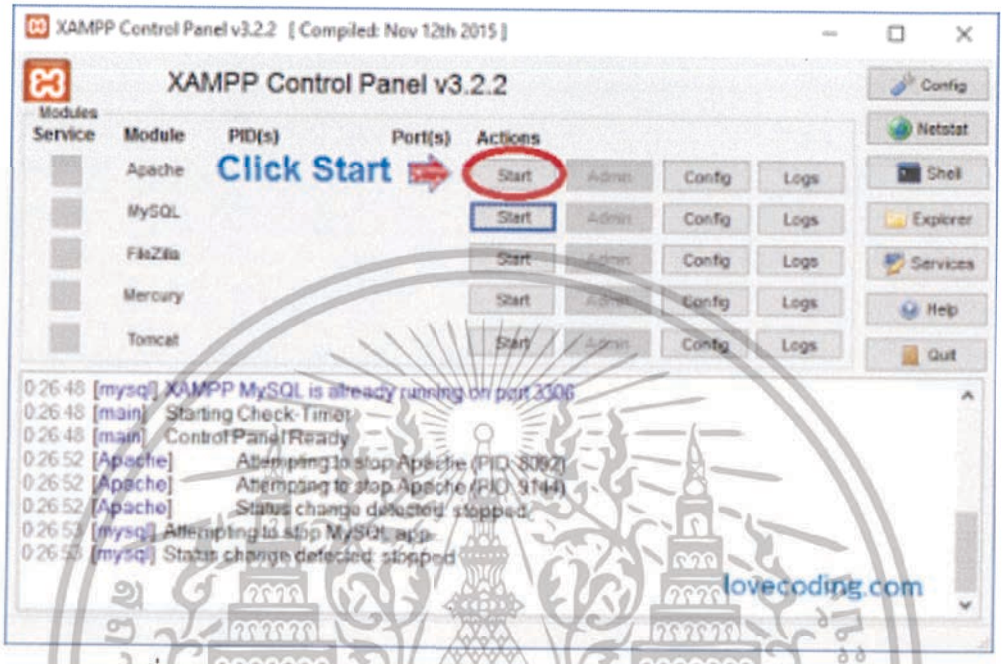
หมายเหตุ การส่งข้อมูลผ่านหน้าเว็บมีหลายรูปแบบ สำหรับวิธีการที่จะทดสอบเป็นวิธีการส่งข้อมูลแบบ GET ผ่าน HTML Forms

2.3.3.2 การติดตั้งโปรแกรมที่ใช้ในการจำลอง Web Server

ในที่นี้จะใช้Xampp ซึ่งXamppคือ โปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ของเราให้ทำงานในลักษณะของ Web Server เพื่อไว้ทดสอบสคริปหรือเว็บไซต์ในเครื่องของเรา หลังจากดาวน์โหลดโปรแกรมแล้ว ให้ทำการติดตั้งโปรแกรมโดยดับเบิ้ลคลิกที่ไฟล์.exe ที่ดาวน์โหลดมา จากนั้นก็ให้คลิกที่ Next ไปเรื่อยๆ จนถึงFinish เป็นอันเสร็จสิ้นวิธีการติดตั้งโปรแกรม ซึ่งก็สามารถที่จะจำลองเครื่องคอมพิวเตอร์ให้เป็น Web Server ได้แล้ว การเปิดใช้งาน Service ของโปรแกรม Xampp ให้คลิกที่ไอคอนสีส้มที่เขียนว่า Xampp Control Panel ซึ่งอยู่ที่หน้าเดสก์ท็อปของเรา จาก

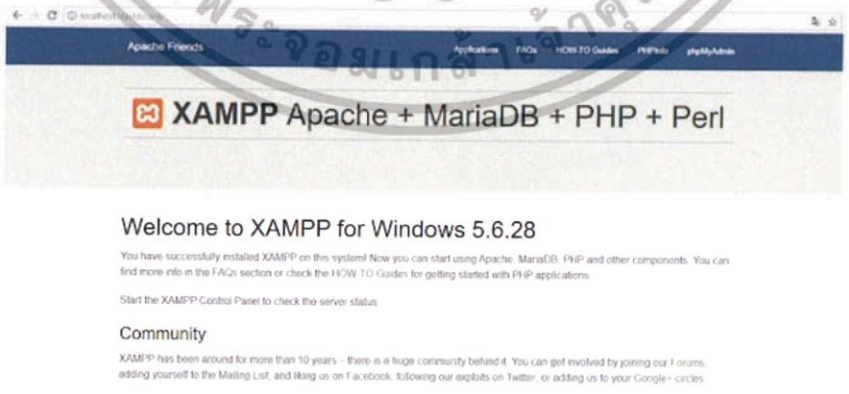
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นจะปรากฏหน้าต่างคอนโซลขึ้นมาให้เราคลิก Start ที่ลูกศรสีแดงชี้อยู่เพื่อเปิด Service ของ Web Server ดังแสดงในรูปที่ 2.41



รูปที่ 2.41 หน้าต่างสำหรับเปิดใช้งาน Service ของโปรแกรม Xampp

เมื่อเปิด Service เสร็จเรียบร้อยแล้วให้มาทดสอบโดยการเปิดบราวเซอร์พิมพ์ URL localhost หรือ127.0.0.1 ซึ่งเป็น Host name ของเครื่องคอมพิวเตอร์เรา จะปรากฏหน้าต่างดังแสดงในรูปที่ 2.42



รูปที่ 2.42 หน้าหลักของ Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.2.1การเขียนโค้ดส่วนรับข้อมูลฝั่งเซิฟเวอร์โดยใช้ภาษา HTML และ PHP

ก่อนเขียนโค้ดให้สร้างไฟล์ get_data.php ในไดเรกทรอรี่ C:\xampp\htdocs\ ก่อนจึงจะมาเขียนโค้ดส่วนที่ใช้ในการรับข้อมูลที่ส่งแนบมากับ request (ข้อความขอบริการ) ซึ่งสามารถ copy โค้ดจากด้านล่างไปวางในไฟล์ get_data.php ได้เลย

```
<!-- HTML <form> method Attribute -->
<!DOCTYPE html>
<html>
<head> <!-- ส่วนหัวของเว็บไซต์ -->
<title>Lovecoding.com</title>  <!-- ชื่อหัวเรื่องของเว็บไซต์ -->
</head>
<body>  <!-- ส่วนเนื้อหาของเว็บไซต์ -->

<h1>Lovecoding.com</h1>

<form id="form1" name="form1" method="get" action=""> <!-- ฟอร์มที่ใช้ในการรับส่งข้อมูลผ่านหน้าเว็บ -->
```

<!-- อธิบายการทำงานของโปรแกรม URL ที่เราทดสอบ http://localhost/get_data.php?data1=lovecoding&data2=blogger

เราได้ทำการส่งข้อมูล data1 และ data2 แนบมากับคำร้องขอซึ่งข้อมูลเมื่อเราแยกออกมาแล้วจะได้ดังนี้

รูปแบบของข้อมูล name = value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ 1 data1 = lovecoding

ข้อมูลที่ 2 data2 = blogger

if(isset($_GET['data1'])) เมื่อมีข้อมูล data1 ส่งมา

if(isset($_GET['data2'])) เมื่อมีข้อมูล data2 ส่งมา

$_GET['data1'] คือการรับข้อมูลที่ถูกส่งมาจาก Client โดยข้อมูลที่รับมานั้นต้องเป็นชื่อ data1 เท่านั้น

$_GET['data2'] คือการรับข้อมูลที่ถูกส่งมาจาก Client โดยข้อมูลที่รับมานั้นต้องเป็นชื่อ data2 เท่านั้น

echo คือการสั่งแสดงผลข้อมูลที่รับมาบนหน้าเว็บ -->

```
<?php
    if(isset($_GET['data1']))echo $_GET['data1'] ."<br>";
    if(isset($_GET['data2']))echo $_GET['data2'] ."<br>";
?>
```

</form>

</body>

</html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโค้ดโปรแกรมด้านบนใช้สำหรับตรวจสอบข้อมูลที่ส่งแนบมากับ request (ข้อความขอบริการ) จากไคลเอนต์ ว่ามีข้อมูลส่งมาจริงหรือไม่ ถ้ามีข้อมูลส่งมาจริงจะทำการแสดงผลข้อมูลที่ส่งมาบนหน้าจอ วิธีการทดสอบให้เปิดเว็บบราวเซอร์ แล้วเข้าไปที่ URL http://localhost/get_data.php?data1=lovecoding&data2=blogger ดังแสดงในรูปที่ 2.43

รูปที่ 2.43 การรับข้อมูลและการแสดงผลข้อมูลบน Web Server

ซึ่งจะเห็นข้อความที่ส่งแนบมากับคำร้องขอแสดงบนหน้าเว็บ “lovecoding blogger” ตรงข้อความที่ส่งมาสามารถเปลี่ยนดูได้โดยให้เราเปลี่ยนที่ http://localhost/get_data.php?data1=lovecoding&data2=blogger แล้วลองทดสอบดูอีกครั้ง ทีนี้หน้าเว็บก็จะแสดงข้อความที่เราส่งแนบไปกับลิงค์คำร้องขอ

2.3.3.3 Google Cloud Platform

Cloud Computing คืออะไร

“Cloud Computing” หรือแปลเป็นภาษาไทยว่า “การประมวลผลแบบกลุ่มเมฆ” เป็นลักษณะของการทำงานของผู้ใช้งานคอมพิวเตอร์ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ต ที่ให้บริการใดบริการหนึ่งกับผู้ใช้ โดยผู้ให้บริการจะแบ่งปันทรัพยากรให้กับผู้ต้องการใช้งานนั้น

โดยสถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (NIST) ได้ให้จำกัดความไว้ว่า “เป็นรูปแบบที่ให้บริการเพื่อใช้ทรัพยากรคอมพิวเตอร์ร่วมกับผู้อื่นผ่านทางเครือข่ายเน็ตเวิร์ค เช่น โครงข่ายเน็ตเวิร์ค เซิร์ฟเวอร์ หน่วยเก็บสำรองข้อมูล แอพพลิเคชั่นและเซอร์วิสต่างๆ เป็นต้น โดยขึ้นอยู่กับความต้องการของผู้ใช้ ”

ประเภทของ Cloud Computing

- **Public Cloud (คลาวด์แบบสาธารณะ)** คือ คลาวด์ที่สร้างขึ้นเพื่อให้ทุกคนสามารถใช้งานได้ ตัวอย่างเช่น บริการคลาวด์ของอเมซอน Amazon Elastic Compute Cloud (EC2) หรือของ Google อย่าง Gooel App Engine
- **Private Cloud (คลาวด์ส่วนตัว)** คือ คลาวด์ที่สร้างขึ้นเพื่อใช้ภายในองค์กรเองเท่านั้น ผู้ให้บริการและผู้ใช้สามารถควบคุมและปรับปรุงระบบความปลอดภัยได้ด้วยตนเอง
- **Hybride Cloud (คลาวด์แบบผสม)** คือ ระบบที่ ผสมผสานระหว่าง Private Cloud และ Public Cloud ซึ่งเป็นการรวมเอาข้อดีของการทำงานทั้งสองแบบมาผสมผสานเพื่อตอบโจทย์การใช้งานเชิงธุรกิจ

Google Cloud Platform

Google Cloud Platform หรือเรียกย่อๆว่า GCP เป็นกลุ่มทรัพยากรในการประมวลผล (computing resource) ของ Google ที่เปิดให้ทุกคนสามารถเข้าใช้งานได้ในรูปแบบของ Public Cloud

2.3.3.4 Protocol

โปรโตคอล คือ ข้อกำหนดหรือข้อตกลงในการสื่อสารระหว่างคอมพิวเตอร์ หรือภาษาสื่อสารที่ใช้เป็น ภาษากลางในการสื่อสารระหว่างคอมพิวเตอร์ด้วยกัน การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบจะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีการสื่อสารที่เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล (Protocol) เช่นเดียวกับคนเราที่ต้องมีภาษาพูดเพื่อให้สื่อสารเข้าใจกันได้

โปรโตคอลช่วยให้ระบบคอมพิวเตอร์สองระบบ ที่แตกต่างกันสามารถสื่อสารกันอย่างเข้าใจได้ คือข้อตกลงที่กำหนดเกี่ยว กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ต่างๆ ทั้งวิธีการส่งและรับข้อมูล วิธีการตรวจสอบข้อผิดพลาดของการส่งและรับข้อมูล การแสดงผลข้อมูลเมื่อส่งและรับกันระหว่างเครื่องสองเครื่อง ดังนั้นจะเห็นได้ว่าโปรโตคอลมีความสำคัญมากในการสื่อสารบนเครือข่าย หากไม่มีโปรโตคอลแล้ว การสื่อสารบนเครือข่ายจะไม่สามารถเกิดขึ้นได้

ตัวอย่างของโปรโตคอล

- โปรโตคอล HTTP หรือ Hypertext Transfer Protocol จะใช้เมื่อเรียกโปรแกรมบราวเซอร์ (Browser)
- โปรโตคอล TCP/IP หรือ Transfer Control Protocol/Internet Protocolคือเครือข่ายโปรโตคอลที่สำคัญมากที่สุด เนื่องจากเป็นโปรโตคอลที่ใช้ในระบบเครือข่าย Internet รวมทั้ง Intranet ซึ่งประกอบด้วย 2 โปรโตคอลคือ TCP และ IP
- โปรโตคอล SMTP หรือ Simple Mail Transfer Protocol คือ โปรโตคอล ที่ใช้ในการรับส่งจดหมายอิเล็กทรอนิกส์บนเครือข่ายอินเทอร์เน็ต
- นอกจากโปรโตคอลที่กล่าวมาข้างต้นแล้ว ยังมีโปรโตคอลต่างๆอีกมากมาย เช่น การโอนย้ายแฟ้มระหว่างกัน ใช้โปรโตคอลชื่อ FTP หรือ File Transfer Protocol การโอนย้ายข่าวสารระหว่างกันก็ใช้โปรโตคอลชื่อ NNP หรือ Network News Transfer Protocol และยังมีโปรโตคอลที่สำคัญสำหรับการสอบถามข้อมูลข่าวสารระหว่างกัน ซึ่งเป็นโปรโตคอลที่มีประโยชน์มาก โปรโตคอลนี้มีชอว่า ICMP หรือ Internet Control Message Protocolเป็นต้น

2.3.3.4.1 HTTP และ HTTPS

HTTP ย่อมาจาก Hypertext Transfer Protocol คือ โพรโทคอลสื่อสารสำหรับการแลกเปลี่ยนสารสนเทศผ่านอินเทอร์เน็ต โดยหลักแล้วใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับเอกสารข้อความหลายมิติที่นำไปสู่การเชื่อมต่อกับ World Wide Web(WWW)จะใช้เมื่อเรียกโปรแกรมweb browser เช่น Firefox, Google Chrome, Safari,Opera และ IE Microsoft Internet Explorer เรียกดูข้อมูลหรือเว็บเพจ โปรแกรมบราวเซอร์ดังกล่าวจะใช้โปรโตคอล HTTP ซึ่งโปรโตคอลนี้ทำให้เซิร์ฟเวอร์ส่งข้อมูลมาให้บราวเซอร์ตามต้องการ และบราวเซอร์จะนำข้อมูลมาแสดงผลบนจอภาพได้อย่างถูกต้อง

ในการแลกเปลี่ยนข้อมูลกันระหว่าง Server และ Client ของ World Wide Web (Server) โดยส่งข้อมูลแบบ Clear text คือ ข้อมูลที่ทำการส่งไปนั้น ไม่ได้ทำการเข้ารหัส ทำให้สามารถถูกดักจับและอ่านข้อมูลได้ง่าย

https ย่อมาจาก Hypertext Transfer Protocol over Secure Socket Layer หรือ http over ssl คือ โปรโตคอลที่ระบุถึงการเชื่อมต่อแบบ Secure http โปรโตคอล https สร้างเพื่อความปลอดภัย ในการสื่อสารผ่านอินเตอร์เน็ตข้อมูลที่ทำการส่งได้ถูกเข้ารหัสเอาไว้ โดยใช้ Asymmetric Algorithm ซึ่งถ้าถูกดักจับได้ก็ไม่สามารถที่จะอ่านข้อมูลนั้นได้รู้เรื่อง โดยข้อมูลนั้นจะสามารถอ่านได้เข้าใจเฉพาะClient กับเครื่อง Server เท่านั้น นิยมใช้กับเว็บไซต์ที่ต้องการความปลอดภัยสูง เช่น เว็บไซต์ของธนาคาร ร้านค้าออนไลน์ เป็นต้น

ความแตกต่างของ http และ https

เว็บไซต์ที่มี https จะเพิ่มความปลอดภัยในการรับส่งข้อมูลให้กับเรามากขึ้น ในขณะที่เราทำธุรกรรมออนไลน์ เพราะ https จะช่วยปกป้องข้อมูลสำคัญที่ส่งออกจากคอมพิวเตอร์ของเราผ่านเครือข่ายอินเทอร์เน็ตเข้าไปยังเซิร์ฟเวอร์ของหน้า ร้านออนไลน์ ทำให้อาชญากรคอมพิวเตอร์ไม่สามารถนำข้อมูลไปใช้งานได้ แต่สำหรับเว็บไซต์ส่วนใหญ่ที่ใช้ http อาชญากรคอมพิวเตอร์จะสามารถขโมยข้อมูลที่ถูกส่ง

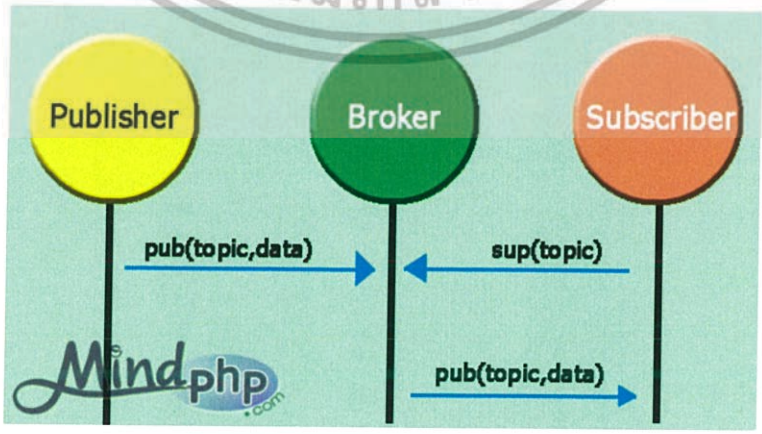
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคอมพิวเตอร์ของเราไปยังเว็บไซต์อีคอมเมิร์ซ (เซิร์ฟเวอร์ของหน้าร้านออนไลน์) ได้อย่างง่ายดาย

2.3.3.4.2 MQTT

Message Queuing Telemetry Transport (MQTT) เป็น Protocol ที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (machine-to-machine) คืออุปกรณ์กับอุปกรณ์ สนับสนุนเทคโนโลยี iot (Internet of Things) คือเทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้ โดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถ ควบคุมอุปกรณ์ต่าง ๆ จากที่อื่นได้ เช่นการสั่งปิดเปิดไฟในบ้านจากที่อื่น ๆ

เนื่องจากโปรโตคอลตัวนี้มีน้ำหนักเบา ออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก การรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็กแบนร์วิธต่ำ ใช้หลักการแบบ publisher / subscriber คล้ายกับหลักการที่ใช้ใน Web Service ที่ต้องใช้ Web Server เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้ตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่ จัดการคิว รับ — ส่ง ข้อมู ลระหว่างอุปกรณ์ และทั้งในส่วนที่เป็ น Publisher และ Subscriber ดังภาพ



รูปที่ 2.44 หลักการของ MQTT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพจะเห็นได้ว่า Topic จะเป็นตัวอ้างอิงหลัก ข้อมูลทีจะ Publisher ออกไปยัง Broker จะต้องมี topic กำกับไว้เสมอ ทางฝ่าย subscriber ก็จะอ้างถึง topic เพื่อเรียกข้อมูลที่ต้องการ เหมือนกับการสมัครเป็นสมาชิกของหนังสือพิมพ์ฉบับหนึ่ง ชื่อของหนังสือก็เปรียบเหมือน topic และผู้ผลิตก็คือ publisher เมื่อถึงเวลาที่หนังสือเสร็จ ผู้ส่ง Broker ก็จะนำหนังสือพิมพ์มาส่งให้เรา

ตัวอย่าง แอพพลิเคชั่นที่ใช้งานคือ Facebook Mesenger

สรุปองค์ประกอบของ MQTT Protocol จะประกอบไปด้วย Broker , Publisher และ Subscriber แต่ละอย่างก็จะทำหน้าที่แตกต่างกันออกไปโดย

- Broker ทำหน้าที่เป็นตัวกลางคอยจัดการกับ ข้อความโดย อ้างอิงจาก Topic
- Publisher จะทำหน้าที่คอยส่งข้อมูลไปยังหัวข้อนั้น ๆ
- Subscriber จำทำหน้าที่คอยดูการเปลี่ยนแปลงของ message ที่อ้างอิงด้วย Topic เช่นถ้ามีหัวข้อหน้าสนใจและมีการเปลี่ยนแปลงก็จะทากรดึงข้อมูลนั้น ๆ มาใช้งาน

2.3.3.5 Node.js

Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript

อธิบายสั้นๆว่า “Node.js” ก็คือ JavaScript ที่ทำหน้าที่อยู่ฝั่ง Backend ทำตัวเป็น Web Server จากเดิมที่เคยอยู่ฝั่ง Frontend ทำหน้าที่ร่วมกันกับ html ในปัจจุบันมีผู้พัฒนาจนมาเป็น Node.js

2.3.3.5.1 ดาวน์โหลดและติดตั้งเพื่อเริ่มต้นการใช้งาน

ขั้นตอนมีดังนี้

1. เข้าไปที่ https://nodejs.org/en/download/ เลือก OS ตามที่เราใช้งาน จากนั้นก็ติดตั้งได้เลย เมื่อติดตั้งเสร็จแล้วให้เราพิมพ์ตามนี้

```
node -v
8.10.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้เลขเวอร์ชั่นที่เราติดตั้งไป ถ้าเลขขึ้นหมายความว่า Node.js พร้อมจะให้เราใช้งานแล้ว

2. เช็ค npm

npm คือ Node package manager เป็นตัวจัดการ package เสริมต่างๆที่เราจะเอามาใช้กับ Node.js เมื่อตอนเราติดตั้ง Node.js มันจะ Install npm มาให้เราด้วย การเช็ค npm พิมพ์ตามด้านล่าง

```
npm -v
5.7.1
```

3. เริ่มต้นการใช้ Node.js

เปิด Editor

แนะนำให้ใช้ vs code เนื่องจากมี terminal ในตัว และยังมี extension สำหรับหลายๆภาษาที่ครบครัน

```
console.log("hello world!");
```

hello.js hosted with ❤ by GitHub

รูปที่ 2.45 ตัวอย่าง code ที่ทดลอง

หลังจากนั้น save แล้วสั่ง node (ชื่อไฟล์) ตามด้านล่าง

```
node hello.js
> hello world!
```

แต่ถ้าหากใช้ terminal ให้เริ่มพิมพ์คำสั่ง node บน Terminal เราสามารถที่จะรันคำสั่ง JavaScript บน Terminal ได้เลย

ตัวอย่าง Code ตามด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
$ node
> console.log('Hello
Node.js');
Hello Node.js
undefined
> function sayHi()
{ return "Hi!"; }
undefined
> sayHi();
'Hi!'
> 10 + 10
20
>
```

```
console.log('Hello
Node.js');
function sayHi() {
  return 'Hi!';
}

console.log(sayHi());
```

หรือทดสอบโดยการสร้างไฟล์ JavaScript ขึ้นมาไฟล์นึง เช่น test.js จากนั้น ใส่ Code ตามด้านล่าง

จากนั้น Run ไฟล์ JavaScript ให้ใช้คำสั่ง node FILENAME

```
node test.js
```

จะได้ผลดังนี้

```
Hello Node.js
Hi!
```

ต่อมาเข้าหน้าเว็บไซต์หลักของ Node.js จะเห็น Code ตัวอย่าง สำหรับแสดงคำว่า Hello World ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

- require('http') : เป็นการ import module http ซึ่งเป็น module หลักของ Node.js
- createServer() : เป็นการสร้าง Server ของ Node.js โดยรับ function ที่มี request และ response
- res.writeHead() : เป็นการกำหนด Content Type
- res.end() : เป็นการสิ้นสุด response และส่งคำว่า Hello World
- .listen(port, address) : เป็นการกำหนด port และ address ของเว็บ Server (address เป็น optional)

ลอง Copy Code ด้านบน แล้วเซฟไว้ที่ไฟล์ชื่อ server.js จากนั้นทำการสั่ง

```
node test.js
```

รัน Node บน command line ด้วยคำสั่ง

จากนั้นเปิดเว็บบราวเซอร์ที่ http://localhost:1337/ จะเห็นข้อความ Hello World แสดงบนจอภาพ

4. ทดลองเริ่มต้นโปรเจ็คแรกด้วยโมดูล fs

เริ่มต้นเขียน Node.js จริงๆ โดยใช้ Built-in Module ที่ชื่อว่า fs (File System)

โดยเราจะทำการสร้างโปรเจ็คขึ้นมา 1 โฟลเดอร์ ภายในประกอบไปด้วยไฟล์ 2ไฟล์คือ

- package.json : เป็นไฟล์ที่เอาไว้กำหนด ชื่อโปรเจ็ค เวอร์ชั่นของโปรเจ็คของเรา รวมถึง dependencies ต่างๆ
- app.js : ไฟล์หลักของเราจะเขียน Code ที่นี่ (ตั้งชื่ออะไรก็ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ package.json จะใช้รูปแบบ JSON และบังคับว่าเราต้องกำหนด name และ version ทุกครั้ง ส่วนค่าอื่นๆ จะกำหนดหรือไม่กำหนดก็ได้ ตัวอย่าง เช่น

```
{
  "name": "hello-node",
  "version": "0.0.1"
}
```

หรือจะใช้คำสั่ง npm init เพื่อสร้างไฟล์ package.json ก็ได้เช่นกัน

การเขียนไฟล์

```
var fs = require('fs');

fs.writeFile('message.txt', 'Hello Node.js',
function(err) {
  if (err) throw err;

  console.log('Saved!');
});
```

สั่งรัน node app.js จะเห็นว่ามีไฟล์ message.txt เพิ่มขึ้นมา 1 ไฟล์ เนื่องจากเราสั่ง เขียนไฟล์ด้วย Node.js

fs.writeFile(filename, data, callback) : จะรับ parameter ตัวแรกเป็นชื่อไฟล์ ตัวที่สองเป็นข้อมูลที่เราจะเขียนลงไปในไฟล์ และตัวสุดท้ายจะเป็น callback function

การอ่านไฟล์

ทดลองแก้ไขไฟล์ message.txt โดยการเพิ่มข้อความตามที่ต้องการ จากนั้นเขียน Node.js ให้มันอ่านไฟล์นี้ดูด้วยคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var fs = require('fs');

fs.readFile('message.txt', function(err, data) {
  if (err) throw err;

  console.log(data);
});
```

callback ของ readFile จะมี 2 arguments คือ error object และ data ซึ่งเป็นข้อมูลที่ได้จากการอ่านไฟล์

Callback function

Callback function คือ function ตัวหนึ่ง (anonymous function) ที่ไม่มีชื่อฟังค์ชัน เป็นฟังค์ชันที่คอยรับข้อมูลหลักจากที่เราสั่งให้ฟังค์ชันนั้นๆ ทำอะไรเสร็จแล้ว มันก็จะไปเรียก callback นั่นเอง

ใน Node.js ฟังค์ชันส่วนมากจะทำงานแบบ asynchronous คือไม่ต้องรอการทำงานนั้นๆจบ ก็จะไปทำฟังค์ชั่นอื่นต่อเลย โดยใช้ callback คอยตรวจสอบว่าฟังค์ชันจบเมื่อไหร่ ฉะนั้น หากเรามีการใช้ fs.readFile และ fs.writeFile เรียงต่อๆกันหลายๆฟังค์ชั่น ก็ไม่จำเป็น ที่มันจะทำงานเรียงจากบนลงล่างเสมอไป ฟังค์ชันที่ประกาศล่างสุด อาจจะทำงานเสร็จก่อนก็เป็นได้

สำหรับ callback function หากบางคนดู anonymous function ไม่ออก อย่างเช่น

```
fs.readFile('message.txt', function(err,
data) {
  if (err) throw err;

  console.log(data);
});
```

อาจจะสร้างเป็นการประกาศฟังค์ชันแบบนี้ก็ได้ เพื่อให้อ่านง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var readFileCallback = function(err, data) {
  if (err) throw err;

  console.log(data);
};
```

5. การสร้าง Web Server ด้วย Node.js

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Hello World\n');
});

server.listen(port, hostname, () => {
    console.log(`Server running at http://${hostname}:${port}/`);
});
```

server.js hosted with ❤ by GitHub view raw

รูปที่ 2.46 code ที่ใช้สร้าง Web server ใน Node.js

หลังจากเขียน Code ตามรูปที่ 2.67 ให้ save แล้วสั่ง Run ตามด้านล่าง

```
node server.js
> Server running at http://127.0.0.1:3000/
```

หลังจากนั้นให้เปิด Browser แล้วเข้าไปที่ localhost:3000 จะพบ Hello World

2.3.3.6 Express.js

Express.js เป็น Module(package) หนึ่งในการจัดการ Web Server โดยที่ Express.js จะจัดการเรื่องต่างๆให้ ไม่ว่าจะเป็น Request, Response , Rounter , Session , etc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.6.1 ติดตั้ง Express.js กันก่อน

```
npm install express --save
```

```
1  const express = require('express');
2
3  const hostname = '127.0.0.1';
4  const port = 3000;
5
6  let server = express();
7
8  server.get('/', function(req, res, next) {
9      res.send('Hello world from Express.js');
10 });
11
12 server.listen(port, hostname, () => {
13     console.log(`Server running at http://${hostname}:${port}/`);
14 });
```

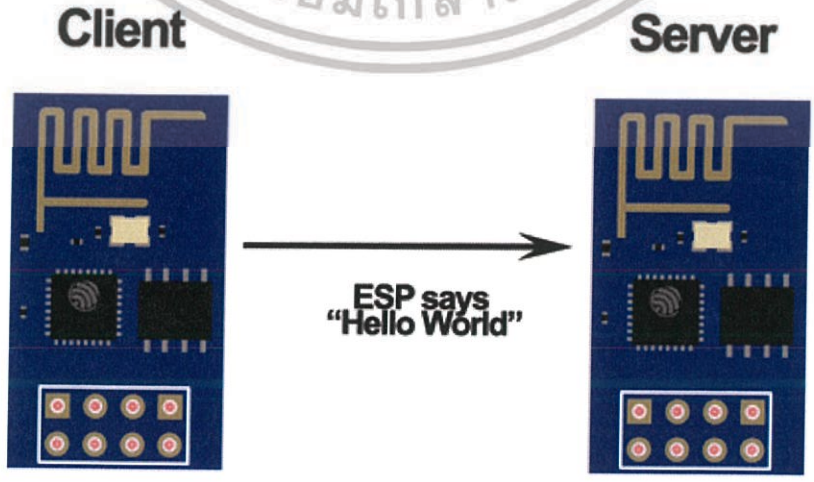
server.js hosted with ❤ by GitHub view raw

รูปที่ 2.47 code ที่ใช้สร้าง Web server ใน Express.js

เมื่อเราลองสั่ง Run Code ตามรูปที่ 2.68 แล้วเข้าไปที่ localhost:3000 จะพบ Hello World แต่คราวนี้มาจาก Express.js

2.3.3.7 Node-Red

Node-RED เป็น tool ใช้สำหรับเชื่อมต่ออุปกรณ์เข้าด้วยกันผ่าน APIs และ ระบบออนไลน์ใหม่ๆ เช่น ถ้ามี NodeMCU 2 ตัว คุณจะให้มันเชื่อมต่อส่งข้อมูลหากัน



รูปที่ 2.48 NodeMCU 2 ตัวเชื่อมต่อข้อมูลหากัน

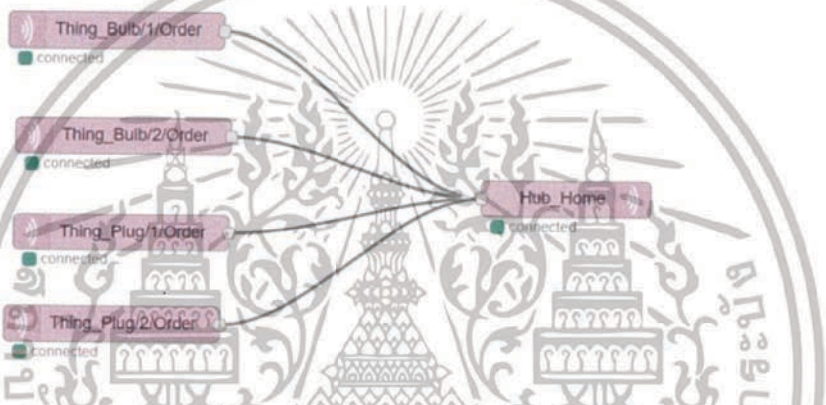
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีแรกคือ มีตัวหนึ่งเป็น Client อีกตัวเป็น Server (TCP/IP) ถ้าจะส่งข้อมูลไปข้างนอก ตัว Server จะต้องจัดการอยู่ Node เดียว

แล้วถ้า Server ต้องส่งข้อมูลไปปลายทางอีก 100 ต้องมาเขียนโค้ดจัดการเอง100 ที

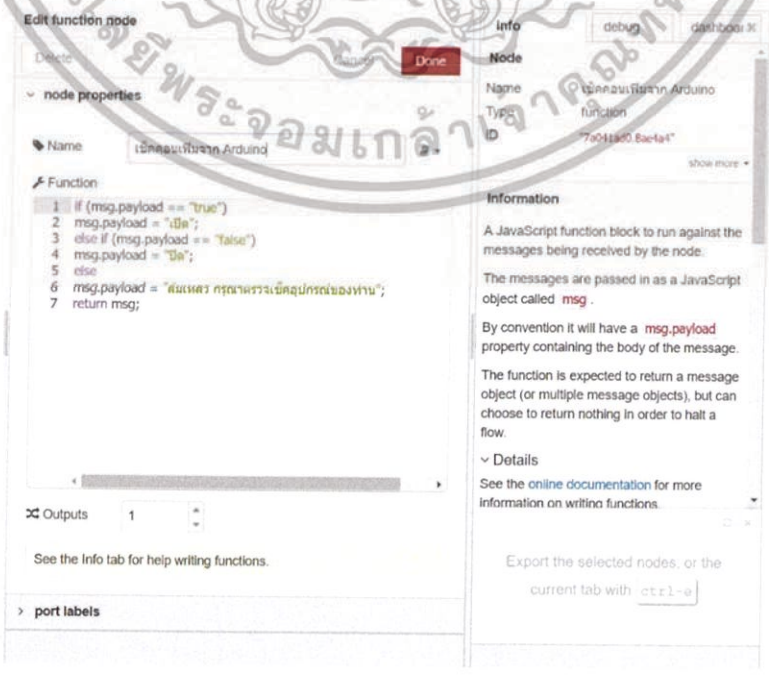
เพนาะฉะนั้น Node-RED จะเข้ามาแก้ปัญหาตรงนี้

- Node-RED จะรับข้อมูลจาก Node ทุกตัว (สมมุติคุยกันผ่าน MQTT) แล้วเราก็แค่ลากเส้นไปยังปลายทาง อยากได้ 10 ปลายทาง ก็ลาก 10 เส้น



รูปที่ 2.49 ลากเส้นเชื่อม Node แต่ละตัว

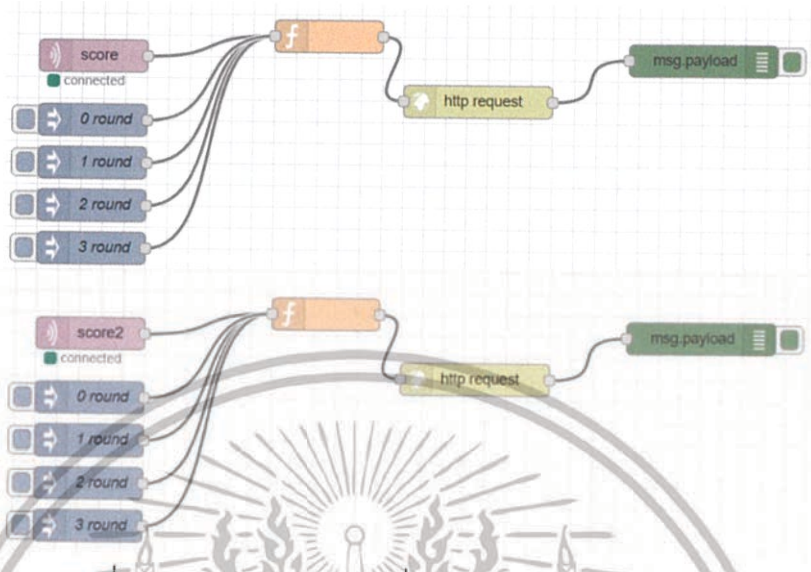
- ระหว่างลากเส้น จะดัดแปลง แก้ไข เพิ่มเติมยังไงก็ได้



รูปที่ 2.50 หน้าต่าง Edit function mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จะเปลี่ยน protocol ปลายทางก็ได้ ไม่มีอะไรเสียหาย เช่น ต้นทางเป็น mqtt ปลายทางเป็น http ก็ได้ไม่มีปัญหา



รูปที่ 2.51 แสดงการลากเส้นเชื่อมกันระหว่างคนละ Protocal

ขั้นตอนการติดตั้ง

1. ติดตั้ง NodeJS https://nodejs.org/en/download/

รูปที่ 2.52 หน้าเว็บ Node.js

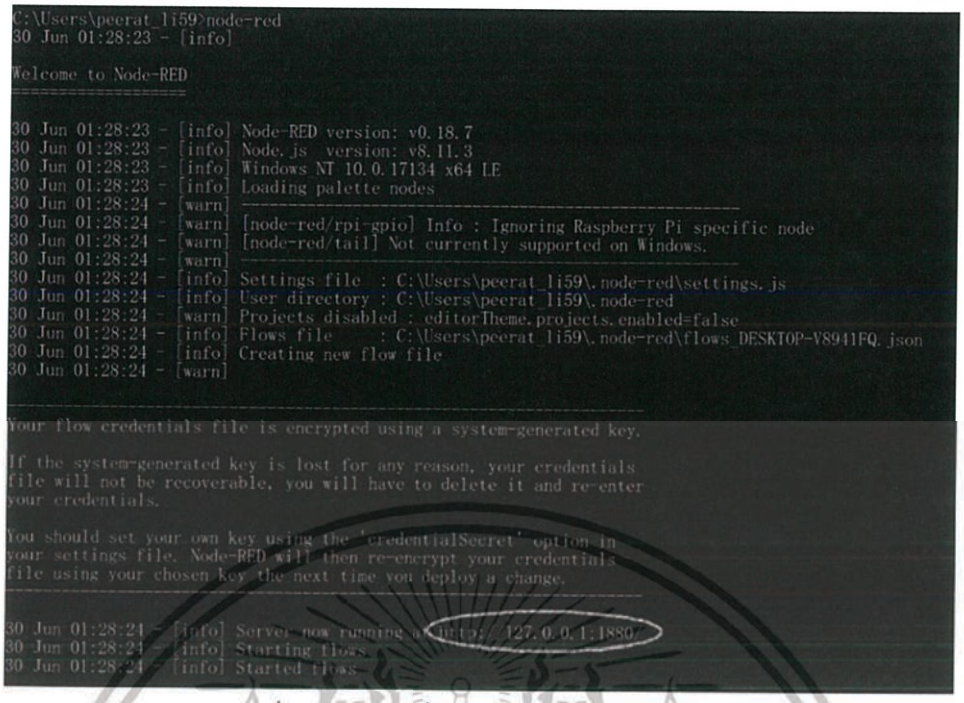
2. เปิดหน้า CMD แล้วพิมพ์ชุดคำสั่ง

สำหรับ MAC/Linuxsudo npm install -g —unsafe-perm node-red

สำหรับ Windows npm install -g —unsafe-perm node-red

3. รันคำสั่ง node-red ในหน้า CMD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.53 รันคำสั่ง node-red ในหน้า CMD

4. เข้าเว็บ http://127.0.0.1:1880 จะได้ Node-RED พร้อมใช้งาน

รูปที่ 2.54 หน้า Node-red

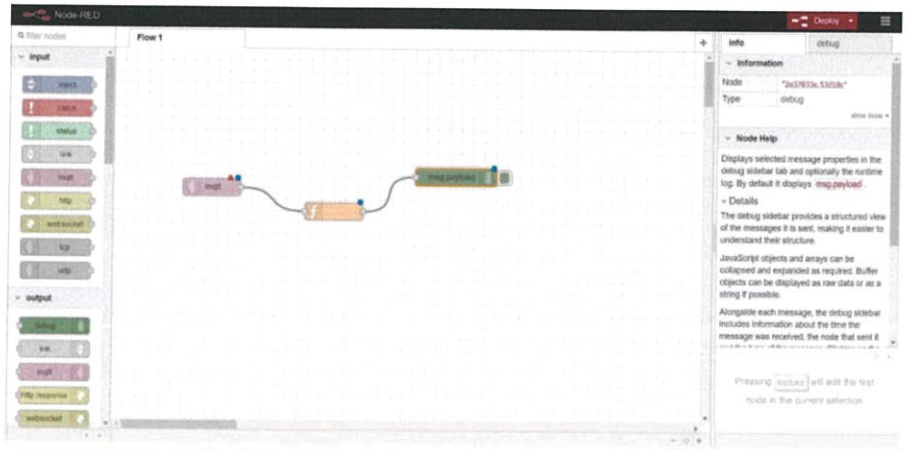
เริ่มต้นใช้งาน Node-RED

Node ที่เราจะใช้งานมีอยู่ 3 ประเภท

- Input จะมีจุดให้ลากด้านขวาด้านเดียว node ที่นำข้อมูลเข้า node-red
- Function มีจุดทั้ง2ข้าง สามารถจัดการข้อมูลทั้งขาเข้า และ ขาออก
- Output จะมีจุดให้ลากแค่ด้านซ้าย นำข้อมูลออกไปแสดงผล

โดยเราสามารถลาก Node มาวางตรงไหนก็ได้บน Flow ของเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



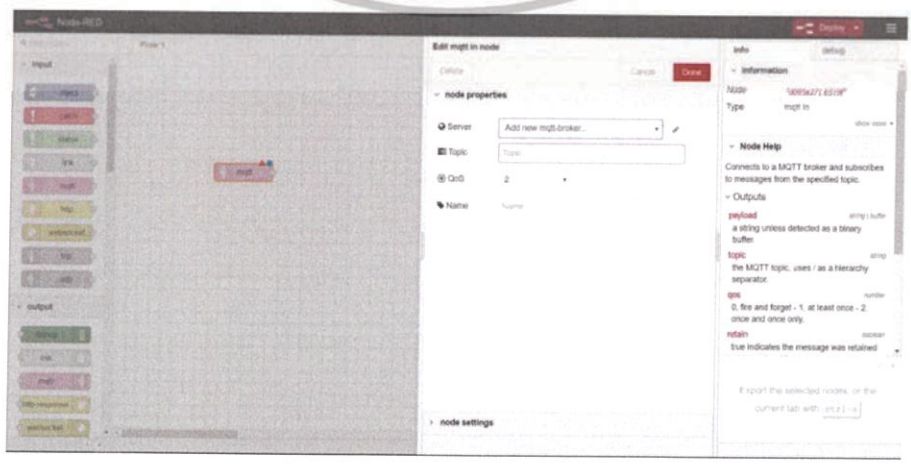
รูปที่ 2.55 หน้า Node-red ที่แสดงการลากเส้นเชื่อม

โดย Node ประเภท 1 และ 3 จะต้องตั้งค่าเซิฟเวอร์ก่อนถึงจะใช้งานได้

ส่วนประเภท 2 แค่ตั้งค่า output กับ input (ลากเส้น)ให้มันก็สามารถใช้งานได้แล้ว

เริ่มต้น ลองลาก Node ที่เป็น input ที่ชื่อว่า mqtt มาวางบนจอ แล้วคลิกดูรายละเอียด

ก็จะเจอตั้งค่าแบบนี้ขึ้นมา



รูปที่ 2.56 หน้ารายละเอียดของ Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Server : ตั้งค่า/กำหนด Server mqtt ที่เราจะ connect

Topic : หัวข้อในการพูดคุย

QoS : ลำดับความสำคัญ (กรณีมีหลายๆ Node มาต่อกันแล้วมีข้อความเข้าพร้อมๆกัน)

Name : ชื่อ node ที่เราจะตั้งให้กับอุปกรณ์

กดรูปปากกาตรง Server เราก็จะเจอกับหน้าตั้งค่า Server แบบนี้

รูปที่ 2.57 หน้าการตั้งค่าของ Input

Name : ชื่อเซิฟเวอร์ที่เราต้องการแสดงบนจอของเรา

Server : broker.hivemq.com Port : 1883 (ใช้งานได้ฟรี)

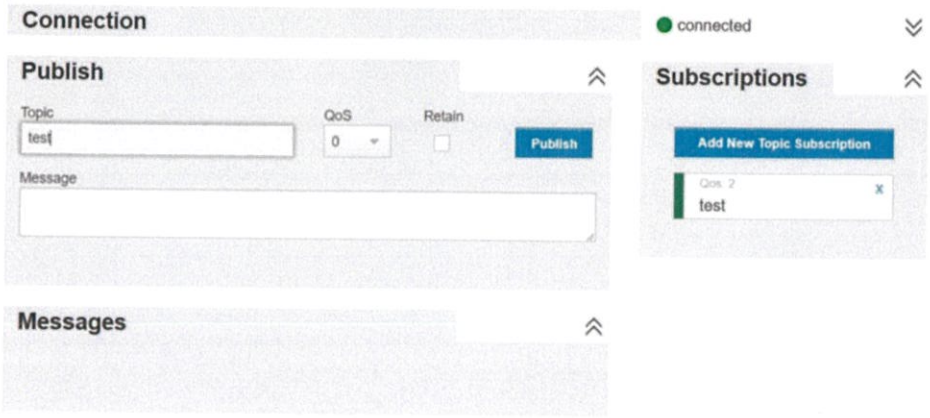
** Port ใช้ใน Node-RED จะเป็น Port ใช้กับ Hardware ส่วนมากจะเป็น 1883

จากนั้นกด Add เพื่อเซฟการตั้งค่าของเรา

จากนั้นเพิ่ม Topic เข้าไป เป็นคำว่า Test

Topic คือ หัวข้อที่จะพูดคุยกันในเครือข่ายเดียวกัน โดยทั้งผู้รับและผู้ส่งจะต้องตั้งหัวข้อในการส่งข้อความเป็นหัวข้อเดียวกัน จึงจะสื่อสารกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.60 ทดสอบการส่ง Messenge

รูปที่ 2.58 แสดงการลากเส้นเชื่อม Debug

ลาก output ตัวแรกสุดที่ชื่อว่า Debug มา และลากเส้นจาก mqtt เข้า Debug ได้เลย แล้วกด Deploy มุมขวาบน

วิธีทดสอบ

รูปที่ 2.59 หน้าเว็บ HiveMQ

1. เข้าเว็บไซต์ http://www.hivemq.com/demos/websocket-client/

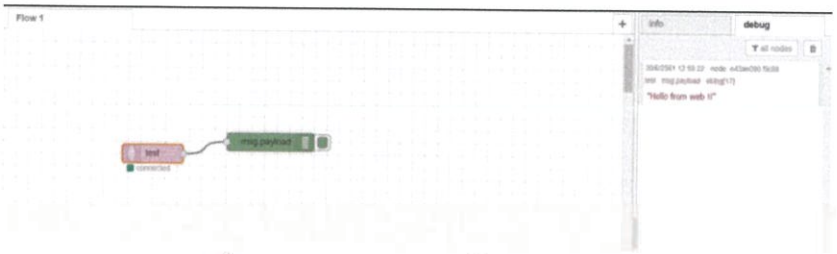
กด Connect เข้าไปได้เลยครับ

กดปุ่ม Subscriptions เปลี่ยนข้อความเป็น test

ตั้งชื่อ Topic ว่า test (Topic ตามที่ตั้งไว้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วลองกลับไปดูหน้า Node-red ปุ่มข้างล่าง Deploy ให้ปรับเป็นหน้า Debug ก็จะแสดงข้อความที่ส่ง (ถ้า Topic ผู้รับ-ผู้ส่งตรงกัน)



รูปที่ 2.61 แสดงข้อความที่ส่งทางด้านขวา

เพียงแค่นี้ก็สามารถส่งข้อมูลจากที่หนึ่งไปยังอีกอีกที่หนึ่งได้ง่ายๆแล้ว โดยผู้ส่งอาจจะเป็น NodeMCU, Web , Line ก็ได้

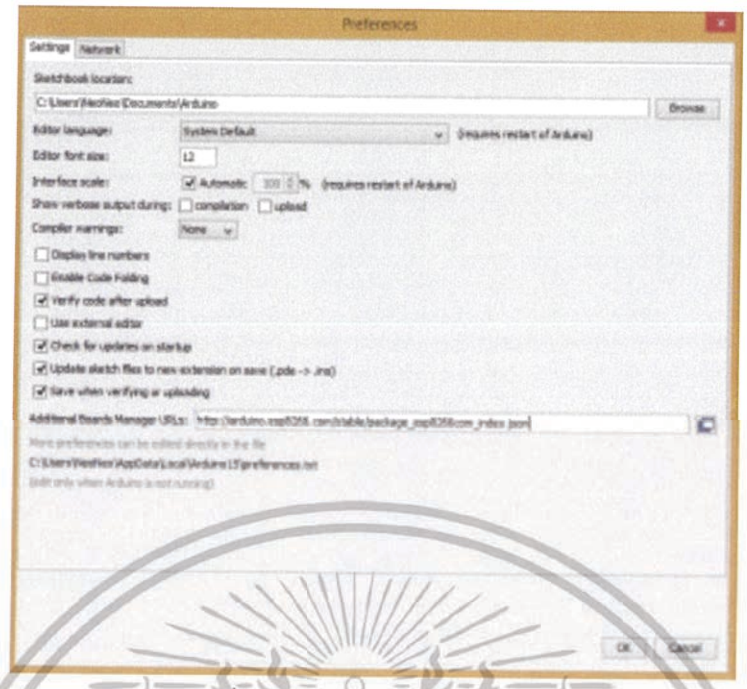
ซึ่ง MQTT รองรับบนทุกอุปกรณ์ (Web,Hardware,Mobile) และเกือบทุกภาษา c,c++ ,java,c#,js,php,python อื่นๆอีกมากมาย

อีกทั้ง Node-RED ยังรองรับ Node อีกมากมาย เช่น Google IoT , AWS , Netpi,Weather Station

2.3.4 NodeMCU

NodeMCU คือ แพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น open source สามารถเขียนโปรแกรมด้วยภาษา Lau ได้ ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล WiFi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเตอร์เน็ตนั่นเอง ตัวโมดูลESP8266นั้นมีอยู่ด้วยกันหลายรุ่นมาก ตั้งแต่เวอร์ชันแรกที่เป็น ESP-01 ไล่ไปเรื่อยๆจนปัจจุบันมีถึง ESP-12 แล้ว และที่ฝังอยู่ในNodeMCU version แรกนั้นก็เป็น ESP-12 แต่ใน version2 นั้นจะใช้เป็น ESP-12E แทน ซึ่งการใช้งานโดยรวมก็ไม่แตกต่างกันมากนัก NodeMCUนั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.62 หน้าต่าง Preferences

buil in มาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ และเมื่อไม่นานมานี้ก็มีนักพัฒนาที่สามารถทำให้ Arduino IDE ใช้งานร่วมกับ Node MCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้เราสามารถใช้งานมันได้หลากหลายมากยิ่งขึ้น NodeMCUตัวนี้สามารถทำอะไรได้หลายอย่างมากโดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่านWiFi และอื่นๆอีกมากมาย

ติดตั้ง Arduino IDE ลงบน ESP8266 NodeMCU

1. ติดตั้ง Arduino IDE เวอร์ชั่น 1.6.4 หรือ ใหม่กว่า โดย Download ตัวติดตั้งได้จาก http://www.arduino.cc/en/main/software
2. เมื่อทำการติดตั้ง Arduino IDE เรียบร้อยแล้ว ให้เปิด Arduino IDE ขึ้นมา ไปที่ Menu File >> Preferences
3. ใส่ URL >> ลงใน Addition Board Manager URLs: ดังนี้ http://arduino.esp8266.com/stable/package_esp8266com_index.json

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. (สามารถดูข้อมูลเพิ่มเติมหากมีการเปลี่ยนแปลงได้จาก https://github.com/esp8266/Arduino)
5. แล้วกด OK

6. จากนั้นไปที่ Menu Tools >> Boar:"xxxxxx" >> Board Manager...
7. เลือก Type เป็น Contributed ไปที่ ESP8266 และกด Install
8. รอจนติดตั้งเสร็จ

รูปที่ 2.63 หน้าต่าง Board Manager

ESP8266 จะมีบอร์ดให้เลือกใช้งานอยู่ด้วยกัน 3 บอร์ด ได้แก่

- Generic ESP8266 Module >> บอร์ด ESP8266 ทั่วไปไม่เจาะจง หรือบอร์ดที่สร้างขึ้นเอง
- Olimex MOD-WIFI-ESP8266 >> บอร์ด ESP8366 ที่บริษัท Olimex เป็นผู้สร้าง
- NodeMCU (ESP8266 ESP12) >> บอร์ด ESP8266 ที่เป็นบอร์ด NodeMCU ที่เราจะใช้งานนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดเด่นของ NodeMCU

- สามารถเชื่อมต่อกับ WiFi ได้โดยไม่ต้องติดตั้งโมดูล WiFi เพิ่มเติม
- ราคาถูกมาก เมื่อเทียบกับบอร์ดที่มี WiFi ในตัวรุ่นอื่นๆ (ราคาในไทยประมาณ 160บาท)
- สามารถเขียน และอัปโหลดโปรแกรมลงบอร์ดด้วยโปรแกรม Arduino IDE ผ่านสาย USB แบบเดียวกับที่ใช้ชาร์จโทรศัพท์ได้
- สามารถอัปโหลดโปรแกรมผ่าน WiFi ได้ เรียกว่า *Over the Air (OTA)*
- ตัวบอร์ดมีขนาดเล็ก (ประมาณ 5.5 x 3 cm.)

2.3.5 ESP32

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth 4.2 BLE ในตัว ผลิตโดยบริษัท Espressif จากประเทศจีน โดยราคา ณ ที่เขียนบทความอยู่นี้ มีราคาไม่เกิน 500 บาท (บอร์ดพัฒนาสำเร็จรูป) โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

- ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
- มีแรมในตัว 512KB
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct
- มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
- ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C

นอกจากนี้ ESP32 ยังมีเซ็นเซอร์ต่าง ๆ มาในตัวด้วย ดังนี้

- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซ็นเซอร์แม่เหล็ก
- เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ดังนี้

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

นอกจากนี้ ESP32 ยังรองรับฟังก์ชั่นเกี่ยวกับความปลอดภัยต่าง ๆ ดังนี้

- รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise
- มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว

ในด้านประสิทธิ์ภาพการใช้งาน ตัว ESP32 สามารถทำงานได้ดี โดย

- รับ – ส่ง ข้อมูลได้ความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b
- เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ – ส่งข้อมูลได้ที่ความเร็ว 135Mbps
- ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

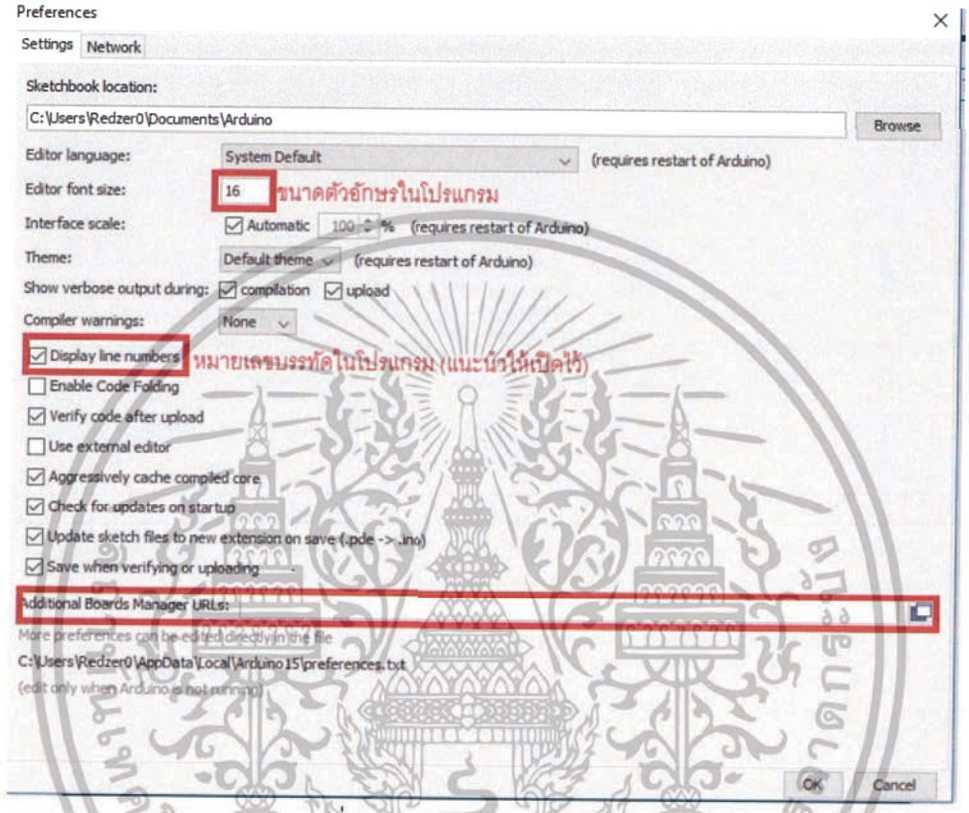
จะเห็นได้ว่า ในราคาไม่ถึง 500 บาท (บอร์ดพัฒนาสำเร็จรูป) และโมดูลเปล่าราคาไม่ถึง 400 บาท สามารถให้ประสิทธิ์ภาพได้เกินราคา ด้วยเหตุนี้ ESP32 จึงเหมาะสำหรับนำมาใช้งานมาก ด้วยเหตุผลทางด้านราคา และประสิทธิ์รูปที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5.1 การติดตั้ง ESP32 CORE

เริ่มจากเปิดโปรแกรม Arduino IDE ขึ้นมา

จากนั้นคลิกเลือกที่ File > Preference

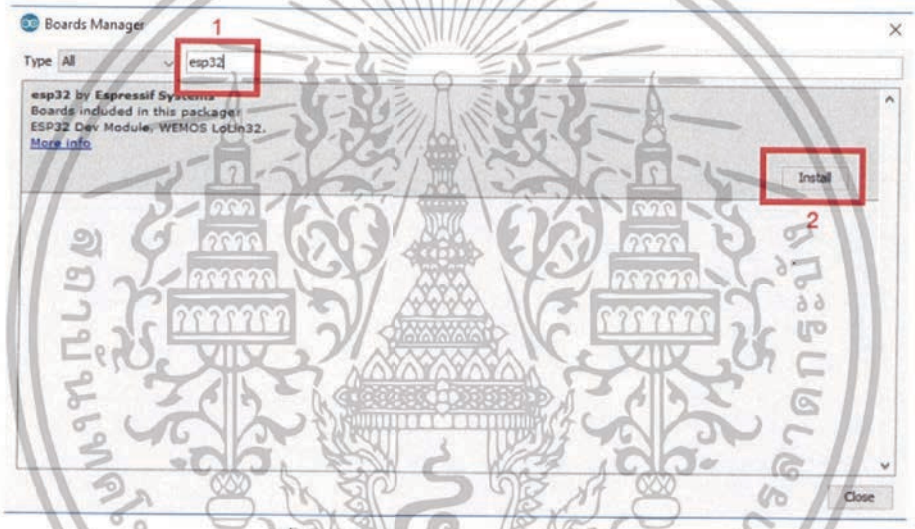


รูปที่ 2.64 หน้าต่าง Preferences

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะได้หน้าต่างแบบนี้ขึ้นมา โดยในช่อง Additional Borads Manager URLs: ให้ใส่ https://dl.espressif.com/dl/package_esp32_index.json แล้วกด OK ได้เลย

จากนั้นกดปุ่ม Tools > Board: "..." > Boards Manager
พอหน้าต่างเด้งขึ้นมาแล้วรอจนแทปด้านล่างมันวิ่ง Update หรือ Download ให้เสร็จก่อน จากนั้นพิมพ์ตรงช่องค้นหาว่า "esp32" มันก็จะขึ้นมาแบบรูปด้านล่างนี้จากนั้นให้กด Install



รูปที่ 2.65 หน้าต่าง Boards Manager

วิธีการตรวจสอบว่า ESP32 Core ติดตั้งเรียบร้อยไหมก็ให้ไปที่ Tools > Borads : "..." แล้วลองเลื่อนลงมาดูถ้าเจอ บอร์ด ESP32 รุ่นต่างๆ แปลว่าติดตั้งเรียบร้อยหร้อมใช้งาน

2.4 การส่ง-รับภาพทางไกล

2.4.1 สตรีมมิ่งมีเดีย (StreamingMediatechnology)

เป็นการส่งข้อมลูผ่านระบบเครือข่ายอย่างต่อเนื่องเหมือนการไหลของกระแสน้ำ พัฒนาขึ้นมาเพื่อใช้ในการส่งข้อมูลข่าวสารเพื่อเผยแพร่ แพร่ภาพ หรือแสดงผลผ่านทางระบบเครือข่ายต่าง ๆ และอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีนี้สามารถ เล่น เพลง วีดีโอ และ ข้อมูลแบบแอนนิเมชั่น โดยส่งผ่านอินเตอร์เน็ตในเวลาแบบ realtime คือไมต้องรอการดาวน์โหลดข้อมูลทั้งหมดมาก่อน Client หรือผู้เล่นจะสามารถเล่นไฟล์ได้ในทันทีที่ยังมีกระบวนการของการส่งอยู่ การดาวน์โหลดไฟล์ที่เป็นภาพและมีเสียงประกอบเป็น Video หรือ audio จากอินเตอร์เน็ตกำลังป็นที่นิยมอย่างมาก ซึ่งถ้าเป็นเมื่อก่อนจะต้องใช้เวลานานมากในการดาวน์โหลดข้อมลูทั้งหมดมาเก็บไว้ในฮาร์ดดิสก์แล้วจึงทำการเล่น วีดิโอได้ ซึ่งทำให้ผู้ใช้รู้สึกเบื่อที่จะดาวน์โหลดไฟล์นั้น ๆ จึงได้มีเทคโนโลยีของ Streaming เกิดขึ้น เพื่อความรวดเร็วในการดาวน์โหลดไฟล์จากอินเตอร์เน็ตมารับฟังหรือรับชม

โดยที่วิธีการส่งข้อมลู Audio และ Video ผ่าน web browser มี 2 ประเภทใหญ่ ๆ คือ การใช้ Web Server ในการนำข้อมูล ส่งไปยังโปรแกรมที่ใช้นำเสนอสื่อนั้นๆ และอีกวิธีหนึ่งคือการใช้StreamingMediaServerซึ่งจะใช้Serverโดยเฉพาะในการให้บริการข้อมูล Audio/Video โดยที่ Streaming Media file จะเริ่มเกือบจะในทันทีที่เล่น ระหว่างที่ ข้อมลูกาลังถูกส่งผ้ชูมสามารถรับฟัง/ชมสื่อนั้น ๆ ได้ทันทีโดยไมจำเป็นต้องรอให้downloadข้อมลูทั้งหมดก่อน

2.4.2 วิธีการเผยแพรไฟล์สตรีมมิ่งมีเดีย

-Unicast : เป็นวิธีการส่งผ่านไฟล์สตรีมมิ่งมีเดีย แบบOn-Demand ไปยังเครื่องของผู้ชมในลักษณะจุดต่อจุด(Point-to- Point)

-Multicast : เป็นวิธีการส่งไฟล์สตรีมมิ่งมีเดียไปยังเครื่องผู้ชมที่ได้ทำการตดิต่อหรือเชื่อมโยงกับสตรีมมิ่งเซิร์ฟเวอร์ผ่านทางเครือข่ายอินเทอร์เน็ต

-Broadcast : เป็นวิธีการส่งไฟล์สตรีมมิ่งมีเดียแบบถ่ายทอดสด(LiveBroadcasting) ไปยังเครื่องของผู้ชมหลาย ๆ จุด พร้อม ๆ กัน

2.4.3 ลักษณะการส่ง Streaming Media

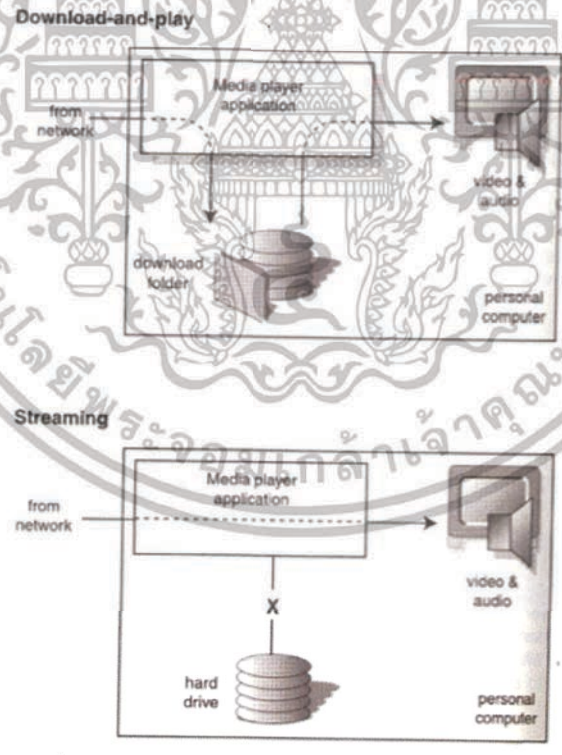
- ไฟล์ออนดีมานด์ (On-Demand) :เป็นไฟล์ที่สามารถเรียกใช้ได้ทันทีเมื่อต้องการ ไฟล์เหล่านี้จะถูกเข้ารหัสในรูปแบบที่เหมาะสมต่อการแสดงผลแบบสตรีมมิ่งแล้วนำไฟล์จัดเก็บไว้เซอร์ฟเวอร์ ซึ่งทุกคนสามารถเรียกใช้งานพร้อมกันได้หลายคนในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละคนสามารถควบคุมฟังก์ชั่นการทางานได้อย่างอิสระ (หยุดการแสดงผล, แสดงผลย้อนกลับ หรือ แสดงผลซ้ำ)

- การถ่ายทอดสด (Live Broadcasting) : เป็นการถ่ายทอดเหตุการณ์ที่เกิดขึ้น ณ เวลานั้นโดยผู้ชมได้รับชมเหตุการณ์ต่างๆได้ เป็นปัจจุบันและทันท่วงทีด้วยวิธีการแปลงสัญญาณจากกล้องวิดีโอเป็นข้อมลูดจิตอล เครื่องเซิร์ฟเวอร์จะทำการถ่ายทอดไปยังเครื่องผู้ชมปลายทาง สามารถเรียกใช้งานได้พร้อม ๆ กันเป็นจำนวนมาก ๆ ได้

- โปรเกรสซีฟดาวน์โหลด (Progressive Download) เป็นการผสมผสานการส่งข้อมูลแบบสตรีมมิ่งและการดาวน์โหลดเข้าด้วยกัน โดยอาศัยพื้นที่บางส่วนภายในหน่วยความจำชั่วคราวของเครื่องคอมพิวเตอร์ที่เรียกว่าบัฟเฟอร์(Buffer)มาใช้เก็บพักข้อมูล ในระหว่างการดาวน์โหลดอยู่นั้นผู้ชมสามารถที่จะเล่นหรือแสดงผลไฟล์ได้ก่อนที่การดาวน์โหลดจะเสร็จสิ้นสมบูรณ์ มักใช้กับไฟล์มัลติมีเดียที่ไม่ใหญ่มากนัก



รูปที่ 2.66 เปรียบเทียบ Download กับ Streaming

2.4.4 ความแตกต่างระหว่าง streaming with web server กับ streaming with streaming media server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Streaming With Web Server

การใช้งาน เริ่มจากทำการแปลง Audio/Video ให้อยู่ในรูปแบบสื่อที่เหมาะสมสำหรับการส่งข้อมลูบน internet โดยพิจารณาจาก bandwidth และทำการ upload แฟ้มมัลติมีเดียไปยัง webserver และสร้างเว็บเพจที่ระบุ URLของ แฟ้มมัลติมีเดียนั้นๆ เมื่อมีการเรียกใช้งานแฟ้มมัลติมีเดีย client-sideplayer จะทำงานและเริ่ม download แฟ้มมัลติมีเดีย เมื่อแฟ้มทั้งหมด download เสร็จสิ้นแล้ว จึงทำการ play ไฟล์นั้นๆ

การส่งข้อมูล Web Server ใช้การติดต่อผ่าน HyperText Transport Protocol (HTTP) ในการติดต่อระหว่าง server และ client ซึ่ง HTTP จะควบคุม Transmission Control Protocol (TCP) ซึ่งจะจัดการเกี่ยวกับการส่งข้อมูลทั้งหมด

- Streaming With Streaming Media Server

การใช้งาน แฟ้มที่ได้จะ upload ไปยัง Streaming Media Server เมื่อแฟ้มมัลติมีเดียถูกเรียกใช้งาน Web Browser จะส่งไฟล์ขนาดเล็กที่ เรียกว่า Meta File ไปยัง Client Player

การส่งข้อมูล : จะใช้ protocol User Datagram Protocol (UDP) ซึ่งมีจุดเด่นที่ความเร็ว ขนาดเล็ก และไม่มีการทำงานเกี่ยวกับการส่งข้อมูลซ้ำหรือคำนวณอัตราการส่งข้อมูล ซึ่งจะเหมาะกับการส่งข้อมูลแบบ realtime ซึ่งข้อมูลที่สูญหายบางส่วนหรือข้อมูลที่เกิด delay จะถูกละความสนใจไป นอกจากนี้อาจมีการใช้งาน Protocol เฉพาะสำหรับ การ streaming media เลยก็ได้เช่น Realtime Streaming Protocol (RTSP)

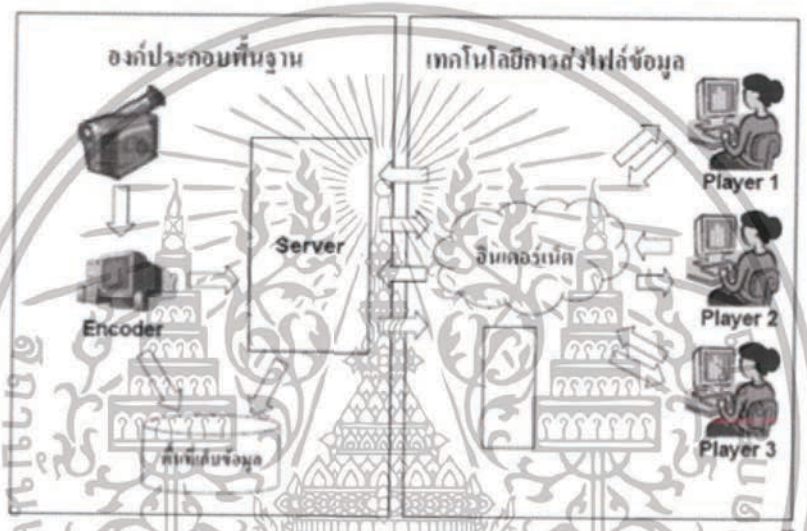
2.4.5 องค์ประกอบพื้นฐานระบบสตรีมมิ่งมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เครื่องเข้ารหัส (Encoder) เป็นเครื่องมัลติมีเดียพีซี(MultimediaPC) ที่ได้ทำการติดตั้งซอฟต์แวร์หรือโปรแกรมไว้สำหรับใช้แปลงไฟล์เสียงและวีดีโอให้อยู่ในรูปของสตรีมมิ่งเช่น MPEG, WMF

- เครื่องเซร์ฟเวอร์ (Servers) เป็นเครื่องที่ได้ติดตั้งซอฟต์แวร์หรือโปรแกรมสำหรับใช้บริการจัดการกับสตรีมมิ่งมีเดีย

- เครื่องผู้ชม(Player) เป็นเครื่องที่ได้รับการติดตั้งซอฟต์แวร์หรือโปรแกรมที่ใช้ในการแสดงผล(Decode)



รูปที่ 2.67 องค์ประกอบพื้นฐานระบบสตรีมมิ่งมีเดีย

2.4.6 เทคโนโลยีการส่งไฟล์ข้อมูล

-โพรโตคอล (Protocol) : เป็นข้อกำหนดสำหรับควบคุมการแลกเปลี่ยนข้อมูลข่าวสารผ่านระบบเครือข่าย ใช้สำหรับแลกเปลี่ยนข้อมูลข่าวสารระหว่างกันของแต่ละส่วนประกอบด้วยมาตรฐานเดียวกัน

-ไฟล์ฟอร์แมต (File Format) : ใช้สำหรับการกำหนดรูปแบบมาตรฐานของการแลกเปลี่ยนข้อมูลระหว่างกัน

-โคเด็ค (Codec) : ใช้สำหรับการเข้ารหัสและถอดรหัสข้อมูลข่าวสารที่บรรจุอยู่ภายในไฟล์ฟอร์แมท

2.4.7 Motion JPEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในมัลติมีเดีย Motion JPEG (M-JPEG หรือ MJPEG) เป็นรูปแบบการบีบอัดวิดีโอที่แต่ละเฟรมวิดีโอหรือเขตข้อมูลอินเทอร์เลซของลำดับวิดีโอดิจิทัลจะถูกบีบอัดแยกต่างหากเป็นภาพ JPEG เดิมทีได้รับการพัฒนาสำหรับแอปพลิเคชั่นมัลติมีเดียพีซี แต่ปัจจุบัน M-JPEG ถูกใช้โดยอุปกรณ์จับภาพวิดีโอเช่นกล้องดิจิตอลกล้อง IP และเว็บแคมรวมถึงระบบแก้ไขวิดีโอที่ไม่ใช่เชิงเส้น

M-JPEG ได้รับการสนับสนุนโดย QuickTime Player, คอนโซล PlayStation และเว็บเบราว์เซอร์เช่น Safari, Google Chrome, Mozilla Firefox และ Microsoft Edge

ไม่เหมือนเช่น Motion JPEG 2000 (และรูปแบบวิดีโอทั่วไป) ที่อนุญาตให้ใช้การรับส่งสัญญาณเสียงรุ่นเก่า Motion JPEG ไม่ได้ให้รหัสเสียงใด ๆ เนื่องจากเป็นเพียงการต่อเฟรม JPEG ที่ยังคงอยู่ ในรูปแบบภาชนะที่เหมาะสมเช่น AVI สามารถให้เสียงได้

ซอฟต์แวร์และอุปกรณ์ที่ใช้มาตรฐาน M-JPEG ได้แก่ เว็บเบราว์เซอร์เครื่องเล่นสื่อคอนโซลเกมกล้องดิจิทัลกล้อง IP เว็บแคมเซิร์ฟเวอร์สตรีมมิ่งกล้องวิดีโอและโปรแกรมตัดต่อวิดีโอที่ไม่ใช่เชิงเส้น

M-JPEG เป็นรูปแบบการบีบอัดเฉพาะ intraframe ในขณะที่รูปแบบวิดีโอ interframe ที่ทันสมัยเช่น MPEG1, MPEG2 และ H.264 / MPEG-4 AVC ให้อัตราส่วนการบีบอัดที่ 1:50 หรือดีกว่า

การที่ M-JPEG ขาดการทำนายเฟรมจำกัด ประสิทธิภาพของมันที่ 1:20 หรือต่ำกว่าขึ้นอยู่กับความทนต่อการสร้างพื้นที่ในเอาต์พุตที่บีบอัด เนื่องจากเฟรมถูกบีบอัดแยกจากกัน M-JPEG จึงลดความต้องการการประมวลผลและหน่วยความจำของอุปกรณ์ฮาร์ดแวร์ลง

2.4.7.1 MJPG-Streamer

MJPG-Streamer คือ โปรแกรมประเภทคำสั่งที่สามารถพิมพ์คำสั่งลงไป แล้วโปรแกรมจะทำการดึงภาพออกจากเว็บแคมของเรา แล้วทำการส่งไปเก็บไว้ เพื่อที่เราจะสามารถเข้ามาดูได้ โดยผ่านเว็บบราวเซอร์หรือผ่านทางมือถือก็ได้

<u>การทำ MJPG-Streamer</u>

มีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อทำการพิมพ์คำสั่ง sudo apt-get update แล้ว ให้ไปโหลด package เหล่านี้มาก่อน ได้แก่

```
sudo apt-get install subversion
sudo apt-get install libv4l-dev
sudo apt-get install libjpeg8-dev
sudo apt-get install imagemagick
```

ซึ่ง Subversion เป็นตัวเอาไว้โหลดซอร์สโค้ดจาก svn server ที่นักพัฒนาโปรแกรมนิยมเอาไปเก็บไว้ ส่วน package อื่นๆ ที่เหลือ ก็เกี่ยวเนื่อง และเป็นไลบรารี่เสริมในการ compile MJPG-Streamer อีกที

จากนั้นจะทำการโหลด MJPG-Streamer ให้ใช้คำสั่งนี้ (ซึ่งจะได้ตัวอัพเดทล่าสุด)

```
svn co https://mjpg-streamer.svn.sourceforge.net/svnroot/mjpg-streamer mjpg-streamer
```

เสร็จแล้วให้เข้าไปที่โฟลเดอร์ mjpg-streamer/mjpg-streamer ด้วยคำสั่ง

```
cd mjpg-streamer/mjpg-streamer
```

จากนั้นเราจะทำการ compile โปรแกรม ด้วยคำสั่ง

```
make USE_LIBV4L2=true clean all
sudo make DESTDIR=/usr install
```

หากไม่มี error แล้ว ให้เราทำการเสียบกล้อง webcam ของเราเข้าทาง port USB ของบอร์ด Raspberry pi ก่อนที่จะเสียบสาย USB อาจจะสั่งคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tail -f /var/log/messages
```

เพื่อดูว่า uvcvideo kernel driver มองเห็นอุปกรณ์ที่เป็น camera หรือยัง ถ้ามองเห็นแล้ว ควรจะได้ข้อความตามรูปข้างล่าง

รูปที่ 2.68 หน้าต่างตัวอย่าง Raspberry Pi ที่ควรจะได้

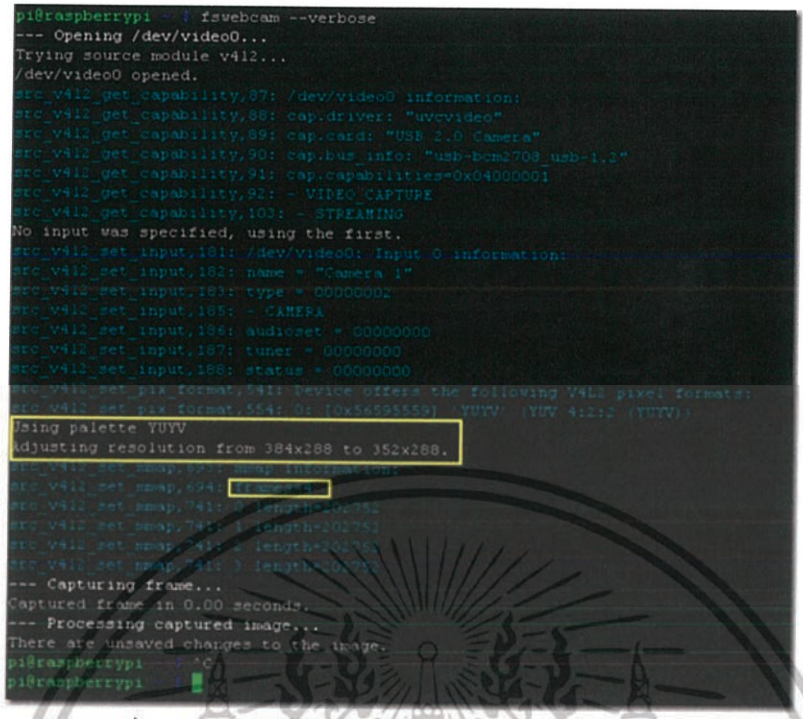
จากนั้นมาติดตั้ง fswebcam ซึ่งตัวนี้จะใช้เป็นตัวเช็คว่า จะต้องปรับค่า output ที่กล้องให้ค่าได้เป็นอะไร ซึ่งเราจะนำไปใช้ในการตั้งค่า streamer ต่อไป ให้ติดตั้ง fswebcam ด้วยคำสั่ง

```
sudo apt-get install fswebcam
```

เสร็จแล้วให้ใช้คำสั่ง

```
fswebcam –verbose
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.69 ข้อมูลที่ได้จากคำสั่ง fswebcam —verbose

ซึ่งจากข้อมูลที่ได้มา ควรที่จะปรับค่า output ที่จะได้จาก streamer ให้เป็น YUYV และ frame rate ควรจะเท่ากับ 4

นำเอาข้อมูลที่ได้นี้ มาใช้ร่วมกับคำสั่ง

```
mjpg_streamer -i "/usr/lib/input_uvc.so -d /dev/
video0 -r 352x288 -f 4 -y YUYV" -o "/usr/lib/
output_http.so -p 8090 -w /var/www/mjpg_streamer"
```

หลังจากที่สั่งคำสั่ง เรียบร้อยแล้ว จะเกิดการ stream รูปที่ได้จากกล้องไปเก็บไว้ที่ web root folder เราสามารถเปิดดูรูปที่ได้ผ่านหน้าเว็บบราวเซอร์ได้เลย โดยใส่ URL ไปที่ ip ของ Raspberry pi แล้วตามด้วยพอร์ต 8090 จากนั้นตามด้วยพารามิเตอร์ ?action=stream

สมมติว่า ip address ของบอร์ด raspberry pi เท่ากับ 192.168.2.105 จะได้ว่า

http://192.168.2.105:8090/?action=stream

ที่หน้าเว็บบราวเซอร์จะพบภาพจากกล้องเว็บแคมของเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและวิธีการดำเนินการโครงงาน

การออกแบบและวิธีการดำเนินการโครงงาน ในเบื้องต้นได้แบ่งงานวิจัยออกเป็น 2 ส่วนหลัก ได้แก่ ส่วนของอุปกรณ์ที่ใช้ในงานวิจัย และส่วนของวิธีการดำเนินงาน โดยส่วนที่2 จะอธิบายวิธีการออกแบบตัวแขนกลและตัวควบคุม การควบคุมและการส่งค่าทางไกล และการรับ-ส่งภาพ

3.1 อุปกรณ์

3.1.1 Micro Servo (SG90) ดังรูปที่ 3.1

รูปที่ 3.1 Micro Servo (SG90)

3.1.2 Servo Motor (MG995 0-180 องศา) ดังรูปที่ 3.2

รูปที่ 3.2 Servo Motor (MG995 0-180 องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 DC Motor ดังรูปที่ 3.3

รูปที่ 3.3 DC Motor

3.1.4 Motor Drive Module L298N ดังรูปที่ 3.4

รูปที่ 3.4 Motor Drive Module L298N

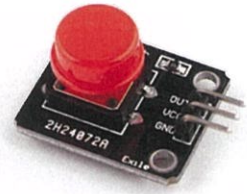
3.1.5 joystick module ดังรูปที่ 3.5

รูปที่ 3.5 joystick module

ใช้สำหรับการควบคุมทิศทาง โดยให้สัญญาณเอาต์พุตออกเป็น Analog โดยใช้ Potentiometer 2 ตัวเพื่ออ่านค่าตำแหน่งของคันโยกให้ค่าออกมาเป็น X และY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

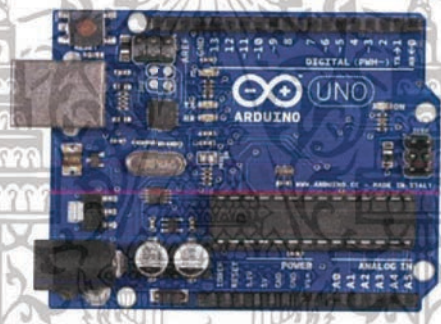
3.1.6 Button Switch Module ดังรูปที่ 3.6



รูปที่ 3.6 Button Switch Module

โมดูลปุ่มกด Button Switch เป็นปุ่มกดลักษณะกดติดปล่อยดับ เหมาะสำหรับการใช้เป็นอุปกรณ์อินพุท ใช้งานร่วมกับวงจรอื่นๆได้ โดยต่อขา VCC, GND, และขา OUT สามารถควบคุมการกดปุ่มได้ตามต้องการ ไม่จำเป็นต้องต่อตัวต้านทานเพิ่มเพื่อลดการ Bounce ของสัญญาณในการกดปุ่มแต่ละครั้ง และง่ายต่อการใช้งาน

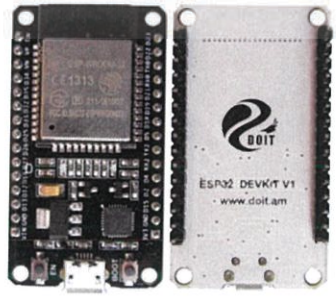
3.1.7 Arduino UNO ดังรูปที่ 3.7



รูปที่ 3.7 Arduino UNO

บอร์ด Arduino UNO นี้ถูกออกแบบมาให้ใช้งานได้ง่าย หลากหลาย มีช่องเสียบสายไฟต่างๆที่สะดวก และมี Shield ต่างๆรองรับเยอะ

3.1.8 ESP-WROOM-32 ดังรูปที่ 3.8

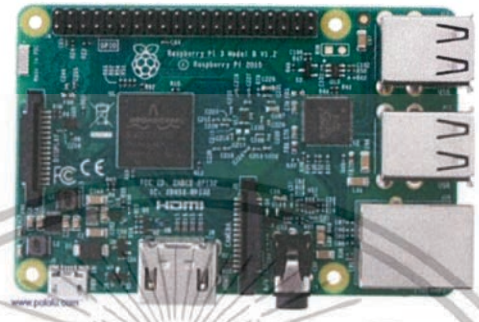


รูปที่ 3.8 ESP-WROOM-32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรุ่นที่พัฒนาต่อจาก ESP8266 ซึ่งได้แก้ไขจุดด้อยต่างๆของ esp8266 ไปจนหมด มีการปรับสเปคของ hardware ให้สูงขึ้นโดยมีการเพิ่ม CPU หลัก เร็วขึ้น มีความเสถียรภาพสูง มีGPIO เพิ่มขึ้น รองรับพอร์ตanalogได้มากกว่า และอ่านค่าได้ละเอียดถึง 12 บิต (0–4095)

3.1.9 Raspberry Pi 3 Model B ดังรูปที่ 3.9



รูปที่ 3.9 Raspberry Pi 3 Model B

3.1.10 Camera Module Board REV 1.3 5MP Webcam Video 1080p 720p ดังรูปที่ 3.10

รูปที่ 3.10 Camera Module Board REV 1.3 5MP

3.1.11 Micro USB Power Adapter 5V 2.5A ดังรูปที่ 3.11

รูปที่ 3.11 Micro USB Power Adapter 5V 2.5A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.12 Switching 12V 5A ดังรูปที่ 3.12

รูปที่ 3.12 Switching 12V 5A

3.1.13 Bearing ดังรูปที่ 3.13

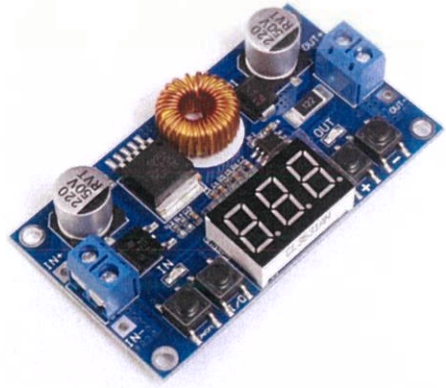
A B

รูปที่ 3.13 Bearing

A) Thrust bearing, B) Ball bearing

ใช้ Thrust bearing ร่วมกับมอเตอร์สำหรับเคลื่อนที่ซ้ายขวา และใช้ Ball bearing ร่วมกับมอเตอร์สำหรับเคลื่อนที่ขึ้นลง

3.1.14 DC Step Down Buck Converter Module (5A 5-36V เป็น 1.2V-32V) ดังรูปที่ 3.14



รูปที่ 3.14 DC Step Down Buck Converter Module

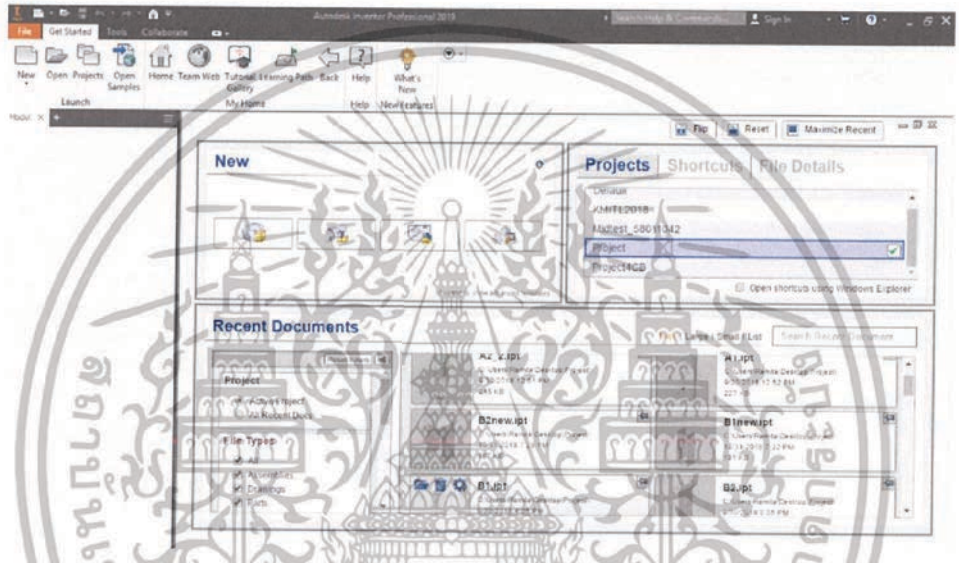
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วิธีการ

โครงงานแขนกลผ่าตัดควบคุมทางไกลนี้แบ่งออกเป็น 3 ส่วนหลักๆ

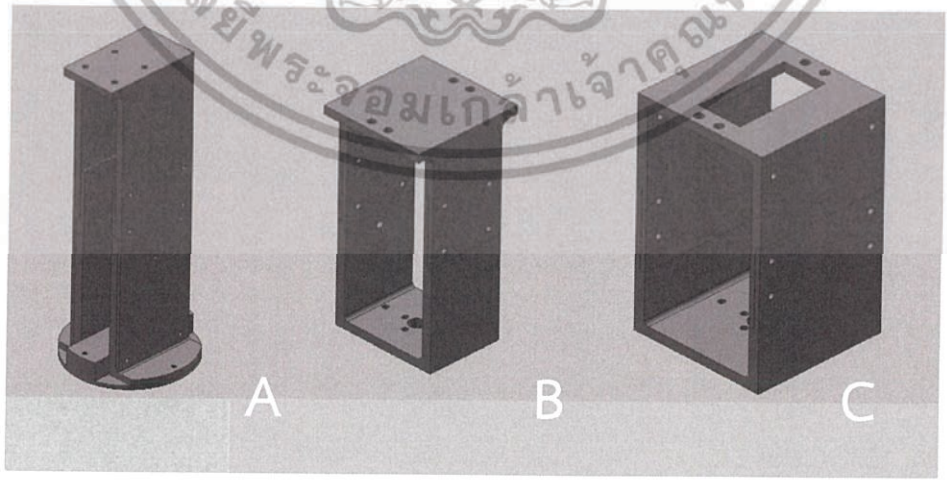
3.2.1 ส่วนการออกแบบตัวแขนกลและตัวควบคุม

โปรแกรมที่ใช้ในการออกแบบแขนกลส่วนต่างๆรวมถึง ส่วนที่ใช้ควบคุม (Joy stick) คือ Autodesk Inventor ดังรูปที่ 3.15โดยจะต้องบันทึกงานเป็นนามสกุล .stl แล้วจึงนำงานที่ออกแบบไปปริ๊นท์ กับเครื่อง 3D



รูปที่ 3.15 โปรแกรม Autodesk

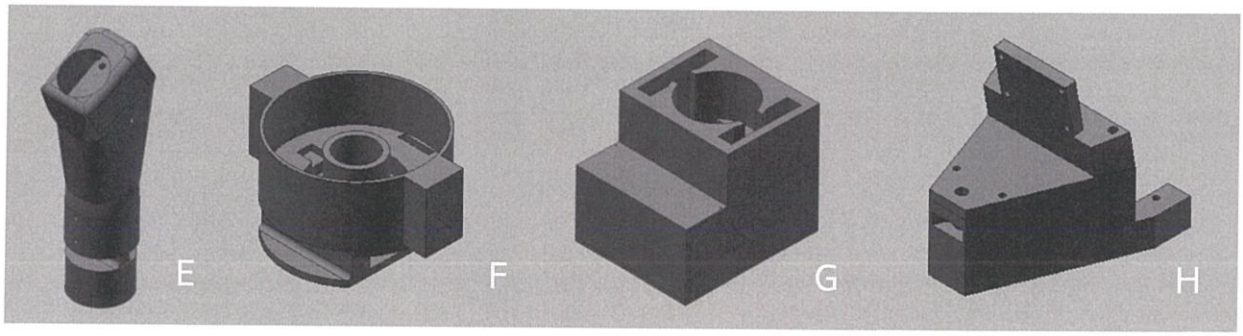
3.2.1.1 ทำการออกแบบ ได้ดังรูปที่ 3.16 และรูปที่ 3.17



รูปที่ 3.16 แขนส่วนต่างๆที่ออกแบบโดย Autodesk Inventor

A) แขนส่วนที่ 1, B) แขนส่วนที่ 2, C) แขนส่วนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 E) ส่วนที่ใช้ควบคุม (Joystick), F) ส่วนที่ใช้ยึดแขนท่อนล่างสุด, G) ส่วนฐานของแขนกล, H) ส่วนที่ใช้ยึดกล้องและกรรไกร

3.2.2 ส่วนการควบคุมและการส่งค่าทางไกล

ทำการเลือกอุปกรณ์ที่จะใช้กับการควบคุมในแต่ละส่วน โดยผู้จัดทำได้ออกแบบให้มีการควบคุม 5 ส่วน ได้แก่

1. ส่วนฐานสามารถหมุนได้รอบ เลือกใช้มอเตอร์ไฟฟ้ากระแสตรง

2. ข้อต่อที่ 1 สามารถเลื่อนขึ้นลงได้ และต้องสามารถรับน้ำหนักในส่วนถัดไปได้หมด จึงเลือกใช้มอเตอร์ไฟฟ้ากระแสตรงเช่นเดียวกัน

3. ข้อต่อที่ 2 สามารถเลื่อนขึ้นลงได้ อุปกรณ์ต้องมีน้ำหนักเบาเพื่อลดแรงที่ข้อต่อแรกต้องรับไว้ และต้องรับน้ำหนักในส่วนถัดไปได้ จึงเลือกใช้เซอร์โวมอเตอร์

4. ข้อส่วนปลายสามารถหมุนปรับทิศทางได้ น้ำหนักเบา จึงเลือกใช้ซอร์โวมอเตอร์

5. ส่วนปลายซึ่งจะต่อกับกรรไกรต้องสามารถควบคุมให้ตัดได้ จึงเลือกใช้เซอร์โวมอเตอร์ SG90 เนื่องจากมีขนาดเล็ก นำหนักเบา และสามารถควบคุมองศาในการตัดได้

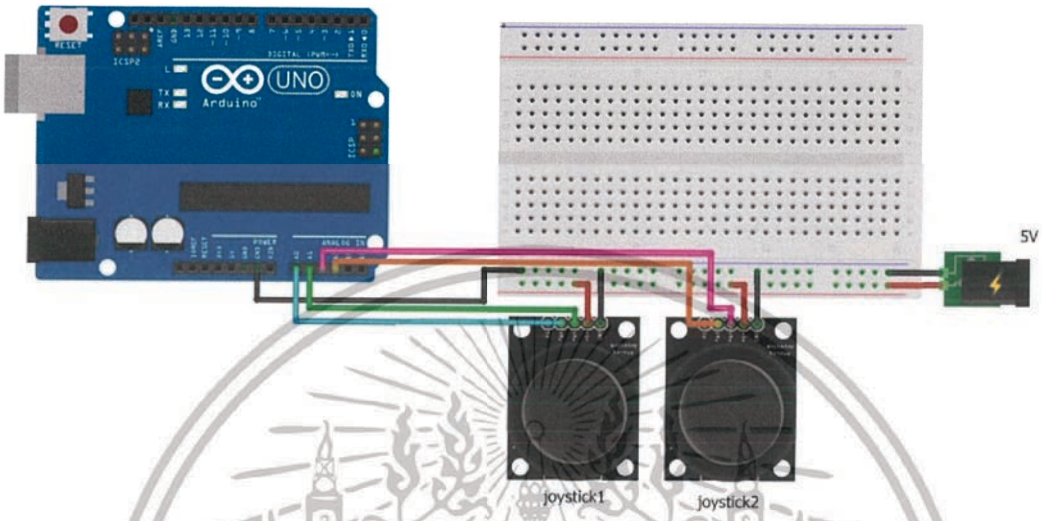
3.2.2.1 ทดสอบการควบคุมแขนกลโดยทำการต่อวงจรและเขียนโปรแกรมอัปโหลดลงบอร์ด Arduino

ในขั้นต้นของการทำโครงงาน จะใช้บอร์ด Arduino Uno ก่อน เนื่องจากเป็นขั้นตอนของการทดลอง และเหมาะกับการใช้งานในผู้เริ่มต้น

3.2.2.1.1 นำ Joystick ทั้งสองตัวต่อกับ VCC (+5V) และ GND จากนั้นขา vRy, vRx ของ joystick1 และขา vRx, vRy ของ joystick2 ต่อเข้ากับขา Analog input ของบอร์ด Arduino Uno ดังรูปที่ 3.18 ซึ่งต่อเข้ากับขา A0,A1,A2,A3 โดยใช้งานเป็นวงจรแบ่งแรงดัน

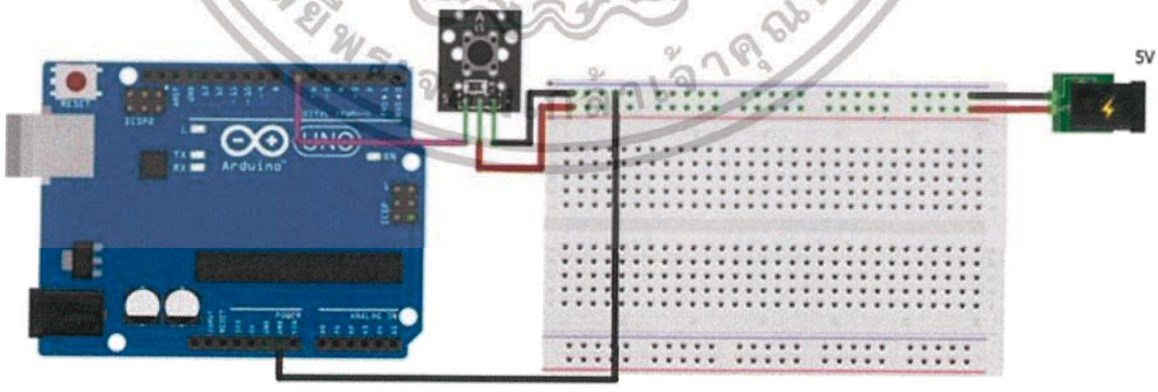
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปลี่ยนตำแหน่งของคันโยก ค่าของสัญญาณหรือระดับแรงดันจะเปลี่ยนตาม ใน Arduino UNO จะสามารถวัดโวลต์ (0-5V) ซึ่งเป็นสัญญาณแบบแอนะล็อก แล้วแปลงค่าที่อ่านได้เป็นค่าดิจิทัลแบบ 10 บิต หรือหรือมีค่าเป็นจำนวนเต็มคือ 0 ถึง 1023 ซึ่งจะสามารถระบุตำแหน่งหรือมุมของคันโยกในแต่ละแกนได้



รูปที่ 3.18 การต่อ Joystick เข้ากับ Arduino

3.2.2.1.2 ทำการต่อ Button Switch ซึ่งเป็นปุ่มกดลักษณะกดติดปล่อยดับ โดยต่อขา VCC (+5V) ,GND และขา OUT ต่อกับ Digital input ของบอร์ด Arduino Uno ดังรูปที่ 3.19 โดยต่อแบบ pull-up ให้ค่าการปล่อยเป็นแรงดันสูง และ กดเป็นแรงดันต่ำ คือ เมื่อไม่ได้กดปุ่ม จะอ่านแรงดันไฟฟ้าได้ +5 V หรือ ลอจิก HIGH (‘1’) แต่เมื่อมีการกดปุ่ม แรงดันไฟฟ้าจะลงมาที่ 0 V ซึ่งเรียกว่าลอจิก LOW (‘0’) ซึ่งเป็นการรับค่าแบบดิจิทัล

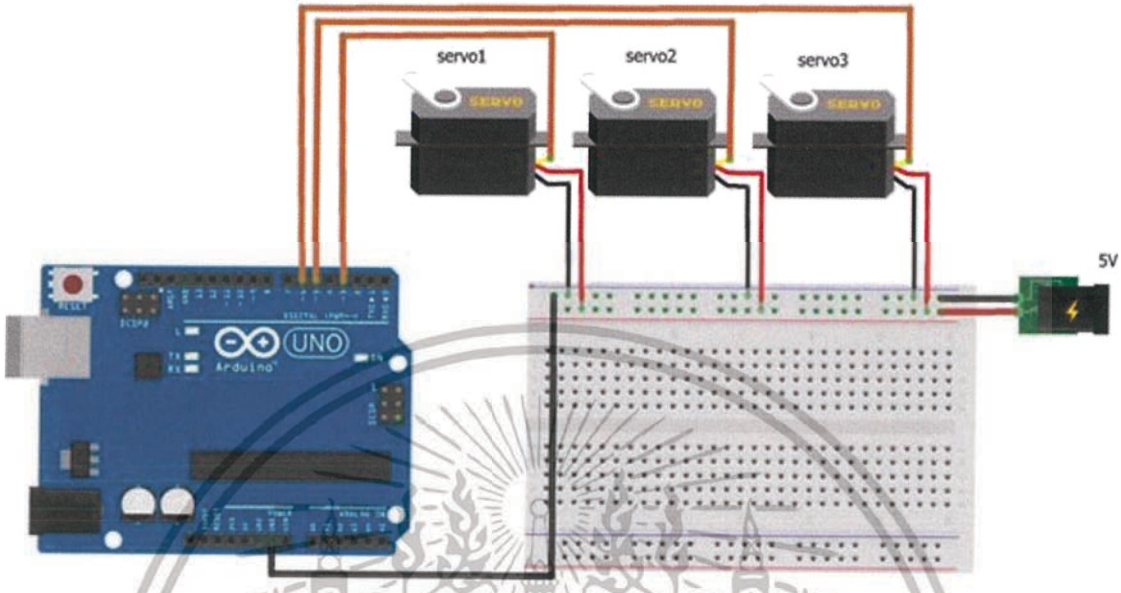


รูปที่ 3.19 การต่อ Button Switch เข้ากับ Arduino

3.2.2.1.3 ต่อเซอร์โวมอเตอร์ เส้นสีน้ำตาลต่อ GND สีแดงต่อ VCC (+5V) และ สีส้มเป็นสายสัญญาณ สำหรับสายสัญญาณจะควบคุมด้วย PWM (Pulse width modulation) โดยใช้

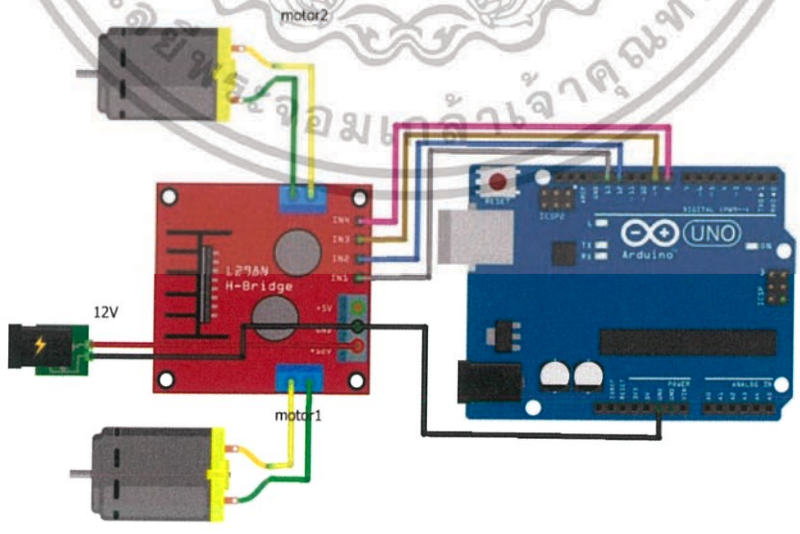
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างของสัญญาณนาฬิกา ในการสั่งงาน ให้เซอร์โวหมุนไปตามองศาที่ต้องการ จึงต้องต่อกับ pin ที่สามารถ ส่งสัญญาณแบบ PWM ได้ ในที่นี้ต่อกับ Digital pin 3 ,5 ,6 ดังรูปที่ 3.20



รูปที่ 3.20 การต่อเซอร์โวเข้ากับ Arduino

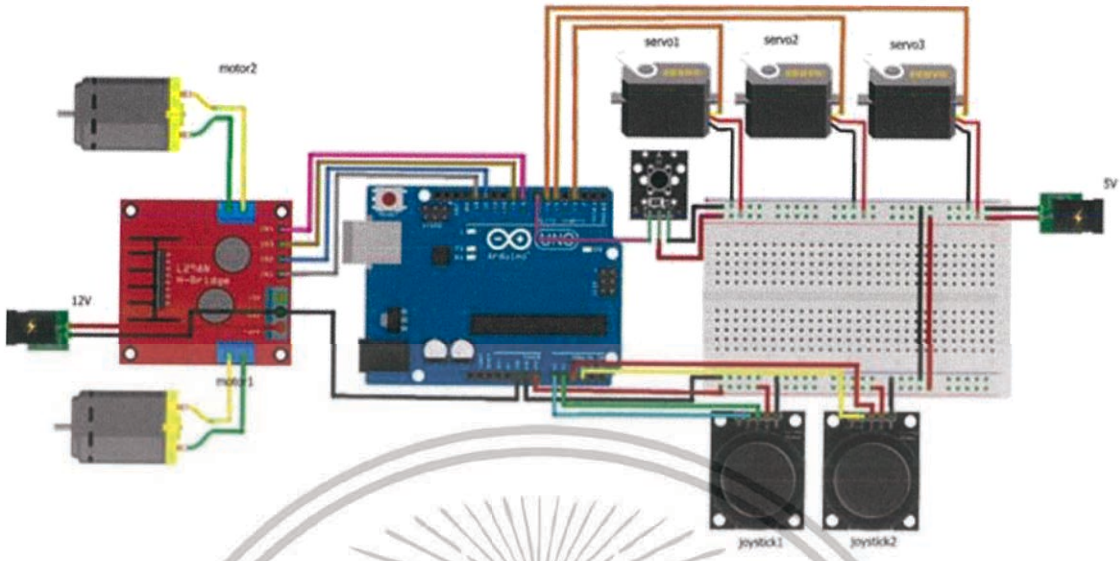
3.2.2.1.4 ต่อมอเตอร์ทั้งสองตัวเข้ากับ motor drive module L298N ซึ่งขา IN1,IN2,IN3 และ IN4 นั้น สามารถต่อกับพอร์ตดิจิทัลใดๆก็ได้ เนื่องจาก 4 ขานี้ จะใช้ในการควบคุมสัญญาณลอจิกบอกทิศทางให้กับมอเตอร์ ในที่นี้ต่อ digital pin 8,9,12,13ตามลำดับ และต่อ Vcc (+12V), GND ดังรูปที่ 3.21



รูปที่ 3.21 การต่อมอเตอร์เข้ากับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1.5 ทำการต่อรวมวงจรรวมเข้าด้วยกันจะได้วงจรรวมดังรูปที่ 3.22



รูปที่ 3.22 การต่อวงจรเพื่อจำลองการควบคุมแขนกล

3.2.2.1.6 เขียนโปรแกรมด้วย Arduino IDE เพื่อจำลองการควบคุมแขนกลในส่วนต่างๆ โดยให้ joystick1 ควบคุมส่วนที่ 1,2 และjoystick2 ควบคุมส่วนที่ 3,4 และ Button Switch ควบคุมส่วนที่ 5

Joystick 1 : ให้ Arduino ทำการอ่านค่าสัญญาณ Analog จากการเลื่อน joystick โดย Arduino จะแปลงค่าที่อ่านได้เป็นจำนวนเต็มคือ 0 ถึง 1023 จากนั้นทำการสร้างเงื่อนไข

ส่วนที่ 1 สร้างเงื่อนไขว่าหากค่าที่อ่านได้จาก vRy ของ joystick ตัวแรกเท่ากับ 1023 หรือดันไปทางซ้ายให้มอเตอร์ตัวแรกหมุนทวนเข็มนาฬิกา แต่หากเท่ากับ 0 หรือดันไปทางขวา ให้มอเตอร์ตัวแรกหมุนตามเข็มนาฬิกา

ส่วนที่ 2 สร้างเงื่อนไขว่าหากค่าที่อ่านได้จาก vRx ของjoystickตัวแรก เท่ากับ 1023 หรือดันถอยหลังให้มอเตอร์ตัวแรกหมุนตามเข็มนาฬิกา แต่หากเท่ากับ0หรือดันไปด้านหน้า ให้มอเตอร์ตัวแรกหมุนทวนเข็มนาฬิกา

Joystick 2 : ให้ Arduino ทำการอ่านค่าสัญญาณ Analog แล้วทำการ map ค่าจาก 0-1023 เป็น 0-180 ตามองศาของเซอร์โวแล้วสร้างเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 3 สร้างเงื่อนไขว่าหากค่าที่อ่านได้จาก vRx ของ joystick ตัวที่สอง มากกว่าค่าเมื่อ joystick ไม่มีการขยับ หรือมีการดันถอยหลัง ให้servo1 ขยับไปด้านหลัง แต่หากน้อยกว่าค่าเมื่อ joystick ไม่มีการขยับ หรือมีการดันไปด้านหน้า ให้servo1 ขยับไปด้านหน้า

ส่วนที่ 4 สร้างเงื่อนไขว่าหากค่าที่อ่านได้จาก vRy ของ joystick ตัวที่สองมากกว่าค่าเมื่อ joystick ไม่มีการขยับ หรือมีการดันไปทางซ้าย ให้ servo2 ขยับทวนเข็มนาฬิกา แต่หากน้อยกว่าค่าเมื่อ joystick ไม่มีการขยับ หรือมีการดันไปทางขวา ให้ servo2 ขยับตามเข็มนาฬิกา

Button Switch : จากการต่อแบบ pull-up เมื่อไม่มีการกดปุ่ม ลอจิกเป็น HIGH และ servo3 อยู่ที่ 100 องศา

ส่วนที่ 5 สร้างเงื่อนไขว่าหากมีการกด หรือ ลอจิกเป็น LOW ให้servo3 อยู่ที่ 0 องศา เมื่อปล่อยให้กลับไปอยู่ที่ 100 องศาดังเดิม

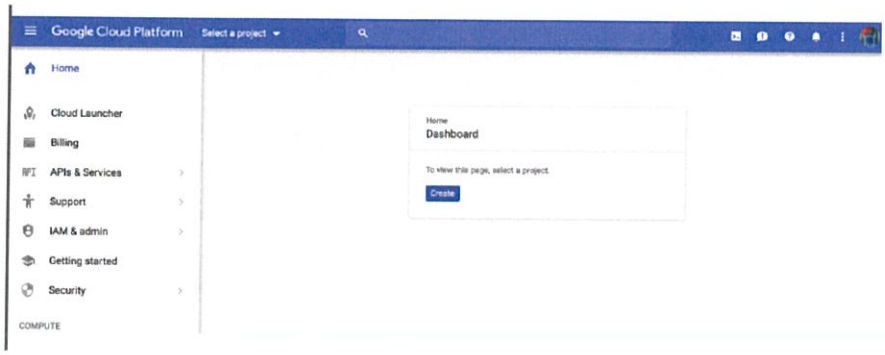
3.2.2.2 การส่งค่าทางไกล

เนื่องจากคอนเซ็ปต์ของโครงงานนี้เป็นการส่งค่าทางไกล ดังนั้นตัวควบคุมจะอยู่ที่หนึ่ง และตัวแขนกลจะอยู่อีกที่หนึ่ง โดยผู้จัดทำได้ทำการศึกษาค้นคว้า และเลือกใช้ ESP32 เป็นตัวควบคุม เนื่องจากเป็นไมโครคอนโทรลเลอร์ที่มีวายฟายติดมาด้วย ราคาถูก และสามารถใช้โปรแกรม Arduino IDE เขียนสั่งงานได้ และเลือกใช้ MQTT โปรโตคอลในการในการรับส่งข้อมูลระหว่างอุปกรณ์ 2 ตัว ซึ่งเป็นโปรโตคอลที่ออกแบบมาสำหรับงานที่ต้องการการสื่อสารแบบเรียลไทม์เช่นนี้ และเนื่องจาก MQTT โปรโตคอลใช้หลักการแบบ publisher / subscriber ในการรับส่งข้อมูลซึ่งต้องมีตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่ รับข้อความทั้งหมดที่ได้จาก Publisher แล้วจึงส่งต่อไปให้ Subscriber ตามแต่ Topic ที่ client ได้ทำการ subscribe ไว้ โดยผู้จัดทำเลือกใช้ Google Cloud Platform โดยการสร้าง GCE (Google Cloud Engine) หรือ เรียกได้ว่าเป็นการสร้าง Virtual Machine(server) บน Google Cloud Platform จากนั้นจึงติดตั้ง MQTT Broker และใช้Node-REDซึ่ง เป็น flow-based programming tool สำหรับเชื่อมต่ออุปกรณ์เข้าด้วยกันผ่าน APIs

3.2.2.2.1 สร้าง GCE (Google Cloud Engine)

1. เริ่มจาก Create Project ดังรูปที่ 3.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 การ Create Project

2. เลือก Compute Engine -> VM instances -> Create instance ดังรูปที่ 3.24

รูปที่ 3.24 การเลือก VM instances และเลือก Create instance

3. ทำการใส่ข้อมูล และคุณสมบัติเครื่องตามความเหมาะสมกับการใช้งาน ดังรูปที่ 3.25

รูปที่ 3.25 หน้าต่างที่แสดงเพื่อให้ใส่ข้อมูล

ทำการใส่ข้อมูล ดังนี้

Name : iotserver

Region : asia-southeast1 (Singapore)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Zone : asia-southeast1-b

Machine type : small (1 shared vCPU)

จากนั้นดำเนินการเปลี่ยน Boot disk เป็น Ubuntu 16.04 LTS

Identity and API access : เลือก Allow default access

Firewall : เลือกทั้ง Allow HTTP traffic และ Allow HTTPS traffic (ในเบื้องต้นเลือกอนุญาตให้ HTTPและHTTPS ผ่านเข้ามาในระบบก่อน)

4.หลังจากที่ Create VM instances success ระบบจะนำมายังหน้า Dashboard ของ VM instances ดังรูปที่ 3.26 ซึ่ง Dashboard นี้จะแสดงให้เห็น พวกค่าต่าง ๆ ที่จำเป็น โดยสามารถสั่ง Stop, Start หรือ Restart ได้จากหน้านี้

รูปที่ 3.26 หน้า Dashboard ของ VM instances

5. จากนั้นจะสามารถทำงานผ่าน SSHได้เลย โดยคลิกที่ SSH ด้านขวามือของ Server ที่ต้องการ หลังจากนั้นจะปรากฏ SSH terminal ขึ้นมา

3.2.2.2.2 ทำการติดตั้ง Node.jsและ node-red ผ่าน SSH โดยพิมพ์คำสั่ง

-> sudo apt-get update

-> curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -

-> sudo apt-get install -y nodejs

สามารถตรวจสอบเวอร์ชั่นของ node ด้วย

-> node -v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดตั้ง node-red ด้วย

-> sudo npm install -g --unsafe-perm node-red

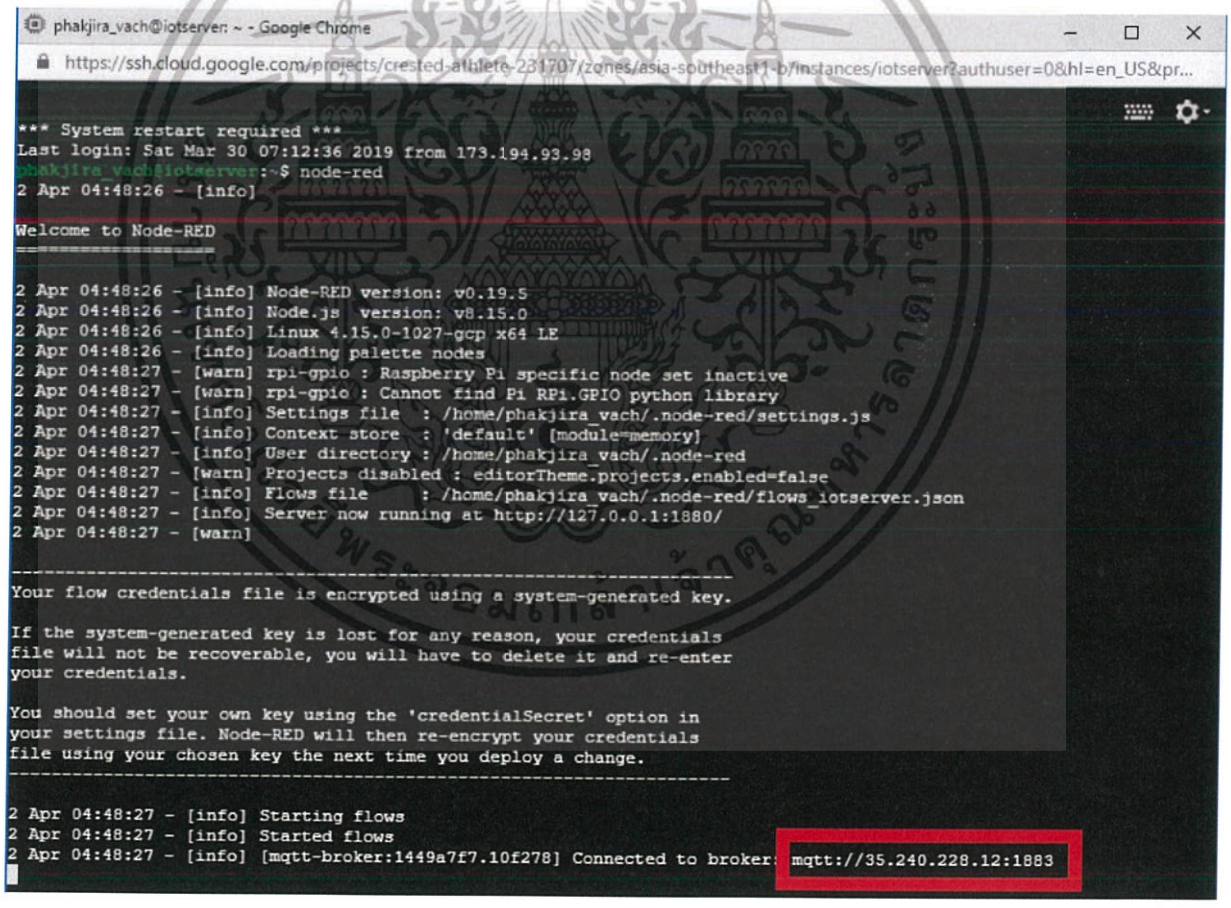
3.2.2.2.3 ทำการติดตั้ง MQTT broker โดยติดตั้ง mosquitto และ mosquitto-clients ด้วยคำสั่ง

-> sudo apt-get install mosquitto mosquitto-clients

3.2.2.2.4 การใช้ node-red และสร้าง Topic

1. เรียกใช้ node-red ในหน้าต่าง Virtual Machine ที่เข้าผ่าน SSH ดังรูปที่ 3.27

-> node-red



รูปที่ 3.27 หน้าต่างที่แสดงขึ้นมาเมื่อ run คำสั่ง node-red

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

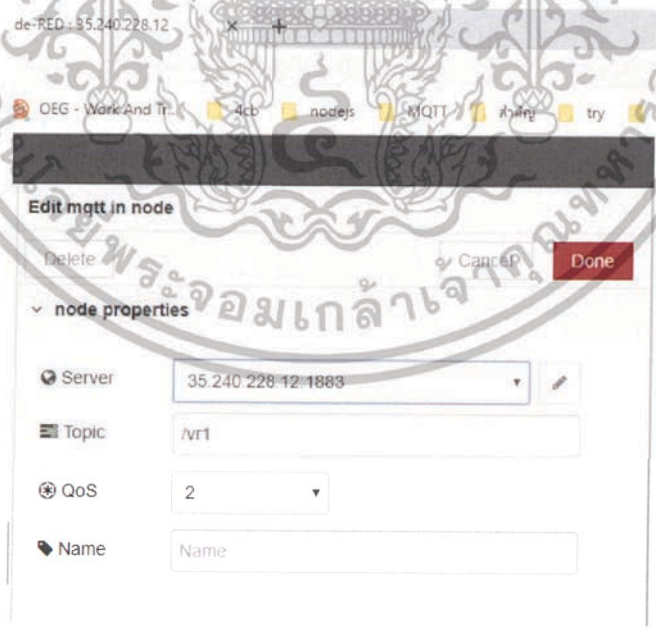
2. เปิดเว็บเบราเซอร์ และเข้าไปที่ IP Address ของ GCP VM ที่พอร์ต 1880 เพื่อใช้งาน node-red ผ่านเว็บเบราเซอร์ แสดงดังรูปที่ 3.28

รูปที่ 3.28 การเข้าใช้งาน node-red ผ่านเว็บเบราเซอร์

3. ลาก Node ที่เป็น input ชื่อ mqtt มาวาง และลาก Node ที่เป็น output ชื่อ debug มาวาง เพื่อจะ monitor ข้อมูลที่ส่งมาจาก ESP32 จากนั้นลากเส้นเชื่อม mqtt และ debug และลาก Node ที่เป็น output ชื่อ mqtt มาวาง ดังรูปที่ 3.29

รูปที่ 3.29 การทำงานในหน้า node-red

3.1 mqtt (input) ตั้งค่าดังรูปที่ 3.30



รูปที่ 3.30 หน้าต่างตั้งค่าเมื่อเข้าไปใน mqtt (input)

Server : 35.240.228.12 ,port :1883

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Server mqtt ที่จะ connect)

Topic : /vr1

(ชื่อหัวข้อในการสื่อสาร)

QoS : 2

(ลำดับความสำคัญ)

3.2 mqtt (output) ตั้งค่าดังรูปที่ 3.31

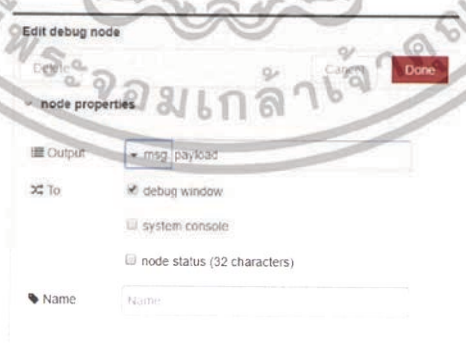
รูปที่ 3.31 หน้าต่างตั้งค่าเมื่อเข้าไปใน mqtt (output)

ตั้งค่า Serverและ Topic ให้ตรงกัน

Server : 35.240.228.12 ,port :1883

Topic : /vr1

3.3 debug (output) ตั้งค่าดังรูปที่ 3.32



รูปที่ 3.32 หน้าต่างตั้งค่าเมื่อเข้าไปใน debug (output)

Output : msg payload

To : debug window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กด Deploy ดังแสดงในรูปที่ 3.33 ซึ่งจะเป็นการ compile และ update คำสั่งให้พร้อมใช้งาน

รูปที่ 3.33 ปุ่ม Deploy

3.2.2.3 ส่วนวงจรที่ใช้กับฝั่งควบคุม และส่วนวงจรที่ใช้กับฝั่งแขนกล

เนื่องจากใช้ ESP32 เป็นตัวควบคุม จึงต้องติดตั้ง ESP32 CORE ลง Arduino IDE ด้วย โดย เปิดโปรแกรม Arduino IDE ไปที่ File - > Preference

1. ในช่อง Additional Boards Manager URLs: ให้ใส่ https://dl.espressif.com/dl/package_esp32_index.json - > OK

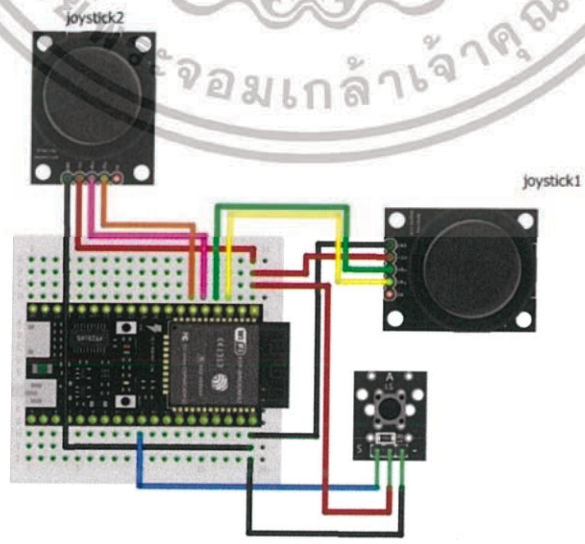
2. ไปที่ Tools -> Board: "..." > Boards Manager

3. เมื่อหน้าต่างขึ้นมาแล้ว พิมพ์ตรงช่องค้นหาว่า esp32 จากนั้นทำการ install

4. ไปที่ Tools -> เลือก Board เป็น NodeMCU-32S

ส่วนการควบคุมทางไกลนี้จะแบ่งเป็น 2 ฝั่ง ได้แก่

1. ฝั่งตัวควบคุม ประกอบด้วย joystick 2 ตัว และ button 1 ปุ่ม เมื่อต่อวงจรทั้งหมดจะได้ดังรูปที่ 3.34 ซึ่งมีขั้นตอนดังนี้



รูปที่ 3.34 วงจรทางฝั่งตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 ทำการต่อ joystick1, joystick2, button กับ ESP32 ดังตารางที่ 3.1, ตารางที่ 3.2 และตารางที่ 3.3 ตามลำดับ

Joystick1 pinout	ESP32 Pin
VRy	VP(A0)
VRx	VN(A1)
VCC	3.3V
GND	GND

ตารางที่ 3.1 แสดงการเชื่อม pin ของ Joystick1 กับ ESP32

Joystick2 pinout	ESP32 Pin
VRx	34(A6)
VRy	35(A7)
VCC	3.3V
GND	GND

ตารางที่ 3.2 แสดงการเชื่อม pin ของ Joystick2 กับ ESP32

Button pinout	ESP32 Pin
OUT	5
VCC	3.3V
GND	GND

ตารางที่ 3.3 แสดงการเชื่อม pin ของ Button กับ ESP32

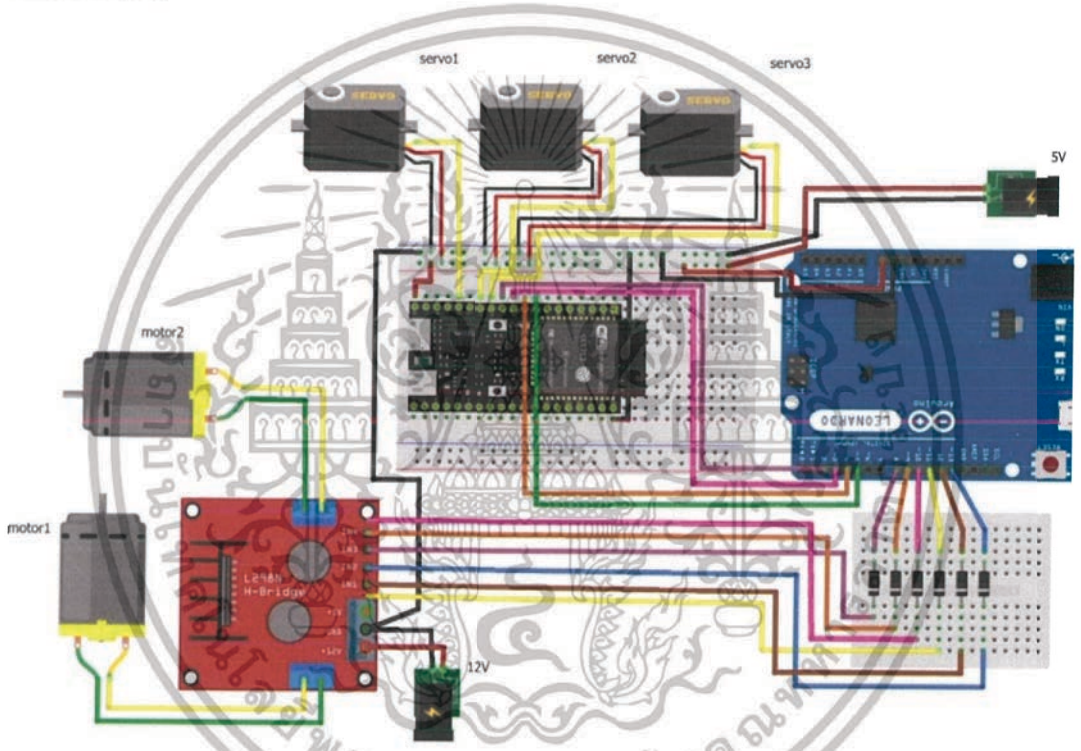
1.2 เขียนโปรแกรมด้วย Arduino IDE ทางฝั่งควบคุม และอัพโหลดลง ESP32 เพื่อควบคุมแขนกลทางไกลโดยตั้งค่าการเชื่อมต่อวายฟาย และเซิร์ฟเวอร์ MQTT Broker ก่อน ดังนี้

```
const char* ssid = “ชื่อ WIFI ที่ต้องการทำการเชื่อมต่อ”;

const char* password = "password";

const char* server = "35.240.228.12"; // server MQTT Broker ที่ใช้

const char* topic = "/vr1"; //ชื่อ topicที่ทำการตั้งไว้ก่อนหน้านี้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้ ESP32 อ่านค่าแรงดันจากการกดปุ่ม และจากการขยับ joystick แล้ว ทำการแบ่งช่วงเป็น3ช่วง ได้แก่ ค่าระหว่าง 0-1000 หรือการขยับไปทางขวาและด้านบน ค่าระหว่าง 1500-3000 หรือการขยับแค่เพียงเล็กน้อย และ ค่าระหว่าง 3000-4095 หรือการขยับไปทางซ้ายและด้านล่าง ทำการสร้างเงื่อนไขว่า หากค่าที่อ่านได้จาก joystick อยู่ในช่วงที่กำหนด หรือหากมีการกดปุ่ม ให้แสดงค่า output ออกไปเป็นตัวเลขต่างกัน และเขียนโปรแกรมให้ทำการ publish ค่าoutputเหล่านี้ไปที่ MQTT Broker ดัง topic ที่กำหนดไว้

2. ฝั่งแขนกล ประกอบด้วยมอเตอร์ 2 ตัว และเซอร์โว 3 ตัว เมื่อต่อวงจรทั้งหมดจะได้ดังรูปที่ 3.35 ซึ่งมีขั้นตอนดังนี้



รูปที่ 3.35 วงจรทางฝั่งแขนกล

2.1 ทำการต่อ servo1, servo2, servo3 กับ ESP32 ตามตารางที่ 3.4 ,ตารางที่ 3.5 และตารางที่ 3.6 ตามลำดับ

servo1 pinout	ESP32 Pin & DC power source
signal	13
VCC	5V (DC power source)
GND	GND

ตารางที่ 3.4 แสดงการเชื่อม pin ของ servo1 กับ ESP32 และ DC power source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Servo2 pinout	ESP32 Pin & DC power source
signal	12
VCC	5V (DC power source)
GND	GND

ตารางที่ 3.5 แสดงการเชื่อม pin ของ servo2 กับ ESP32 และ DC power source

Servo3 pinout	ESP32 Pin & DC power source
signal	14
VCC	5V (DC power source)
GND	GND

ตารางที่ 3.6 แสดงการเชื่อม pin ของ servo3 กับ ESP32 และ DC power source

2.2 ต่อ pin ESP32 เข้ากับขา digital pin ของ Arduino เพื่อให้ Arduino ช่วยประมวลผลการทำงานของมอเตอร์ ตามตารางที่ 3.7

ESP32 Pin	Arduino UNO
27	2
26	3
25	4
33	5

ตารางที่ 3.7 แสดงการเชื่อม pin ของ ESP32 กับ Arduino UNO

2.3 นำมอเตอร์ทั้ง 2 ตัว ต่อเข้ากับ Motor drive L298N โดย Motor1 ต่อกับ OUT1 และ OUT2ของ Motor drive L298N และ Motor2 ต่อกับ OUT3 และ OUT4 ของ Motor drive L298N ดังรูปที่ 3.35

2.4 ขา IN1,IN2,IN3,IN4,ENA,ENB ของ Motor drive ต่อเข้ากับ digital pin ของ Arduino โดยนำ ไดโอดมากั้นกระแสไฟฟ้าไหลย้อนกลับ และ ในการต่อกับ Arduino นั้น ขา IN1,IN2,IN3 และ IN4 นั้น สามารถต่อกับพอร์ต Digital ใดๆก็ได้ เนื่องจาก 4 ขานี้ จะใช้ในการควบคุมสัญญาณลอจิคบอกทิศทางให้กับมอเตอร์ ส่วน ENA และ ENB นั้นจำเป็นที่จะต้องต่อกับพอร์ต Digital ที่รองรับ PWM เนื่องจากจะต้องใช้สัญญาณ PWM ในการควบคุมความเร็วของมอเตอร์ ทำการต่อ pin ตามตารางที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino UNO	Motor drive L298N
13	IN2
12	IN1
11 (PWM)	ENA
10 (PWM)	ENB
9	IN4
8	IN3

ตารางที่ 3.8 แสดงการเชื่อม pin ของ Arduino UNO กับ Motor drive L298N

2.5 ต่อ pin out ของ Motor drive L298N, Arduino UNO, ESP32 เข้ากับ DC power source ตามตารางที่ 3.9, ตารางที่ 3.10 และ ตารางที่ 3.11

Motor drive L298N	DC power source
+12V	12V
GND	GND

ตารางที่ 3.9 แสดงการเชื่อมต่อ Motor drive L298N กับ DC power source

Arduino UNO	DC power source
Vin	5V
GND	GND

ตารางที่ 3.10 แสดงการเชื่อมต่อ Arduino UNO กับ DC power source

ESP32	DC power source
Vin	5V
GND	GND

ตารางที่ 3.11 แสดงการเชื่อมต่อ ESP32 กับ DC power source

2.6 เขียนโปรแกรมด้วย Arduino IDE ทางฝั่งแขนกล และอัพโหลดลง ESP32 โดยตั้งค่าการเชื่อมต่อ WIFI และเซิร์ฟเวอร์ MQTT Broker ก่อน ซึ่ง MQTT Broker ต้องเหมือนฝั่งควบคุม แต่ใช้วายฟายที่แตกต่างกัน จากนั้นเขียนโปรแกรมให้ทำการ subscribe ค่า output ที่มาจาก MQTT Broker ดัง topic เดียวกันกับทางฝั่งควบคุม และสร้างเงื่อนไขให้เซอร์โวแต่ละตัวทำงานตาม output

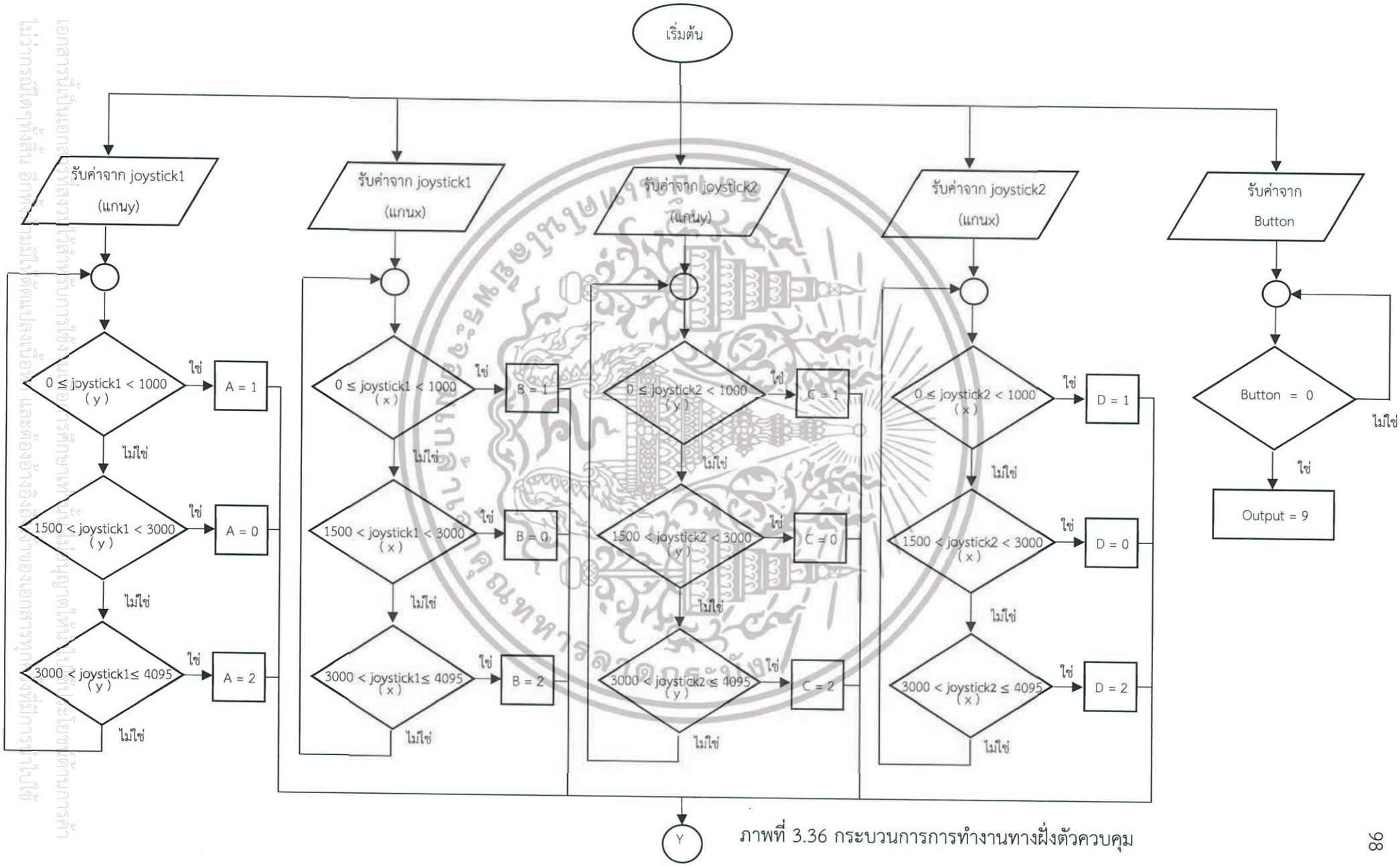
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เข้ามา และ output ที่จะใช้ควบคุมมอเตอร์ ให้ส่ง HIGH(1) และ LOW(0) ออกไปทาง pin ที่เชื่อมต่อกับ Arduino UNO

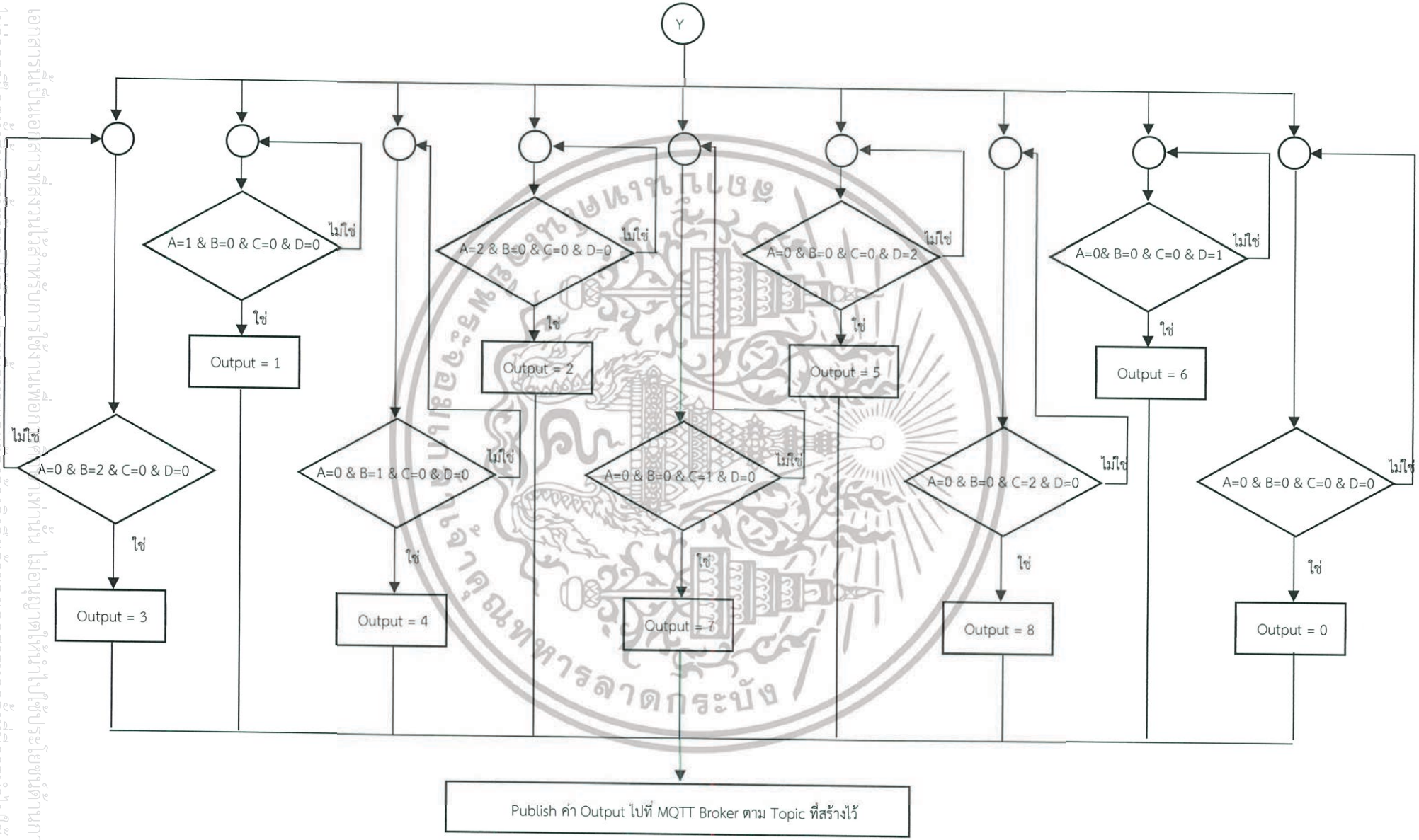
2.7 เขียนโปรแกรมด้วย Arduino IDE และอัพโหลดลง Arduino UNO โดยเขียนให้อ่านค่าจาก pin ที่เชื่อมต่อกับ ESP32 เข้ามา สร้างเงื่อนไขว่า หากค่าที่อ่านได้จาก pin ไหนเป็นมีลอจิกเป็น 1 (HIGH) และ pin ที่เหลือเป็น 0 (LOW) ให้มอเตอร์ตัวนั้นหมุนไปตามทิศและตามความเร็วที่กำหนด

กระบวนการการทำงานของโครงงานสามารถสรุปได้ดังแผนผังการทำงาน ดังนี้

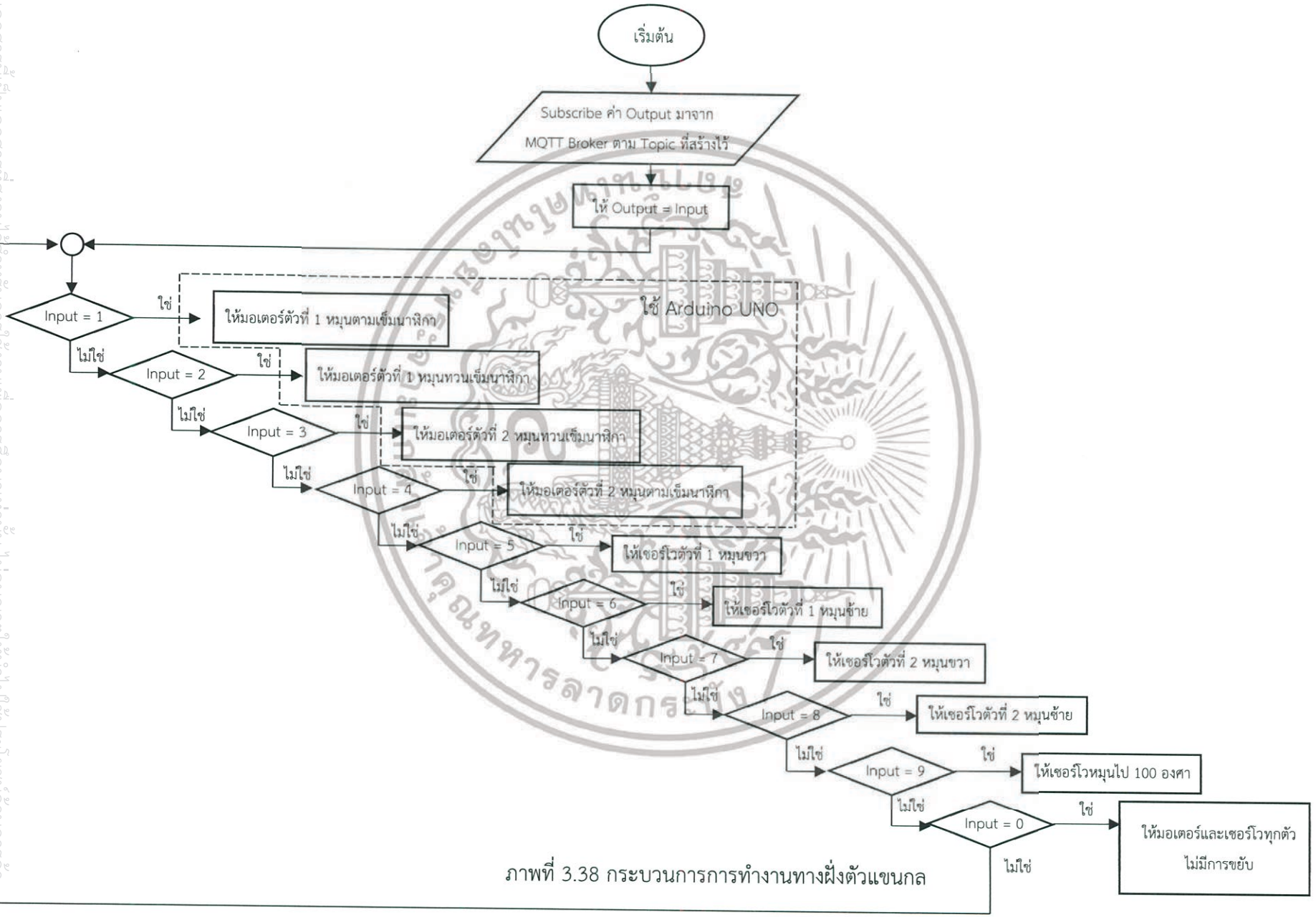
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.36 กระบวนการการทำงานทางฝั่งตัวควบคุม



ภาพที่ 3.37 กระบวนการการทำงานทางฝั่งตัวควบคุม



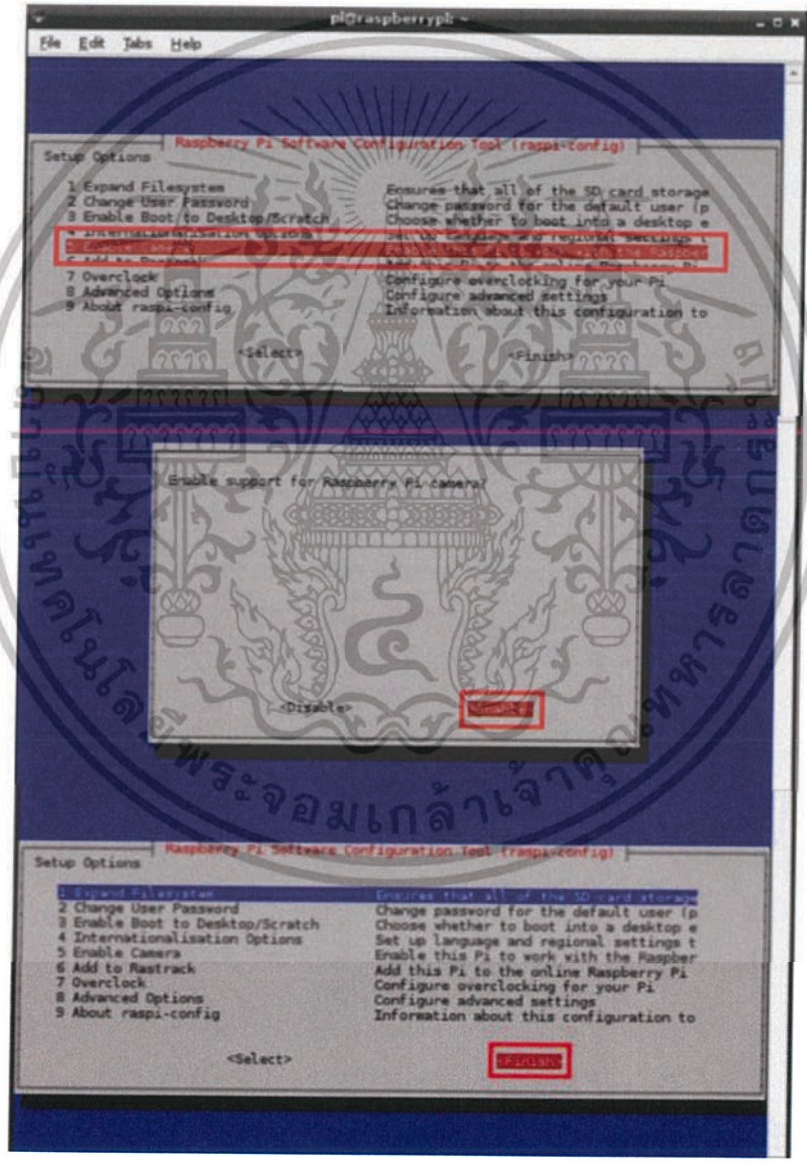
ภาพที่ 3.38 กระบวนการการทำงานทางฝั่งตัวแขนกล

3.2.3 ส่วนรับ-ส่งภาพ

ในส่วนของการรับ-ส่งภาพทางไกลผ่านระบบอินเตอร์เน็ตนั้นในโครงงานนี้เราเลือกใช้แอปพลิเคชัน MJPG Streamer ในการส่งภาพแบบ real-time โดยผ่านการใช้คำสั่งจาก บอร์ด Raspberry Pi และเลือก Raspberry Pi Camera เป็น IP camera

เมื่อเราต่อ Raspberry Pi Camera ลงบอร์ด Raspberry Pi ให้ทำการตั้งค่าเปิดใช้งานกล้องก่อน โดยวิธีการมีดังนี้

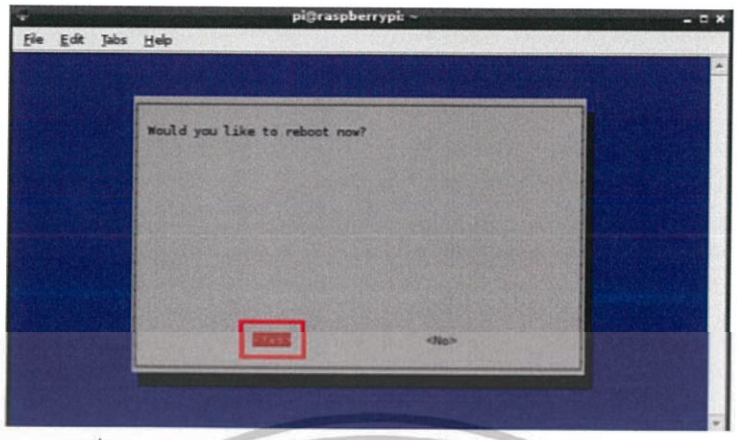
3.2.3.1 ใช้คำสั่ง sudo raspi-config เพื่อเข้าเมนูการตั้งค่า และเลือก Enable Camera จากนั้นกด Finish ดังรูปที่ 3.39



รูปที่ 3.39 หน้าเมนู Raspberry Pi Software Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2 จากนั้นทำการกดรีบูทบอร์ด Raspberry Pi ดังรูปที่ 3.40



รูปที่ 3.40 หน้าเมนูกดเลือก Yes เพื่อรีบูท Raspberry Pi

เมื่อทำการต่อกล้องลงบอร์ด Raspberry Pi แล้ว จะทำการลงแอปพลิเคชัน MJPG Streamer ลงในบอร์ด Raspberry Pi ซึ่งจะมีขั้นตอนดังนี้

3.2.3.3 ใช้คำสั่ง sudo modprobe bcm2835-v4l2 เพื่อเปิดใช้งานไดรเวอร์สำหรับกล้อง ถ้าไม่ติดข้อผิดพลาดใดๆจะเห็นอุปกรณ์ Video0 ในไดเรคเทอร์รี่ /dev ซึ่งสามารถตรวจสอบได้จากการใช้คำสั่ง ls /dev | grep vid จะขึ้นดังรูปที่ 3.41

รูปที่ 3.41 ตรวจสอบการเปิดใช้งานไดรเวอร์

3.2.3.4 ทำการติดตั้ง compiler ที่ต้องใช้ ก็คือ libjpeg-dev, imagemagick, libv4l-dev ด้วยคำสั่ง sudo apt-get install libjpeg8-dev imagemagick libv4l-dev

3.2.3.5 ทำการดาวน์โหลด MJPEG Streamer ลงบน Raspberry Pi ตามคำสั่งด้านล่าง ซึ่งไฟล์ที่ถูกดาวน์โหลดมาจะอยู่รูปของไฟล์ที่ถูกบีบอัด

wget http://terzo.acmesystems.it/download/webcam/mjpg-streamer.tar.gz

จากนั้นทำการแตกไฟล์ด้วยคำสั่ง

tar -xvzf mjpg-streamer.tar.gz

3.2.3.6 ทำการเชื่อม (link) ไลบรารี่ที่จะใช้ในการคอมไพล์ MJPEG Streamer ด้วยคำสั่ง sudo ln -s /usr/include/libv4l1-videodev.h /usr/include/linux/videodev.h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.7 เข้าไปในไดเรคเทอรี่ของ MJPEG Streamer แล้วทำการแก้ไขในไฟล์ Makefile โดยพิมพ์คำสั่งดังต่อไปนี้

cd mjpg-streamer

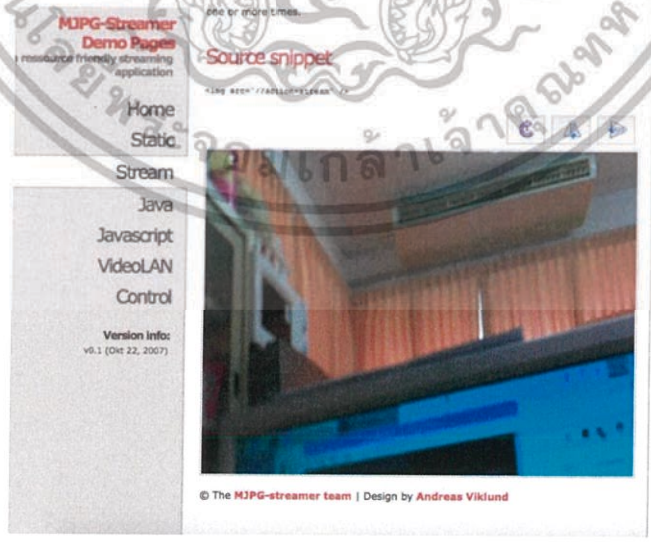
nano Makefile

แก้ไขโดยการใส่ # ในบรรทัด PLUGINS += input_gspcav1.so เพื่อคอมเมนต์เอาไว้ แล้วใช้คำสั่ง Make เพื่อคอมไพล์โปรแกรม

3.2.3.8 เมื่อติดตั้งเสร็จ ทำการรันแอปพลิเคชั่น MJPEG Streamer ด้วยคำสั่ง sudo ./mjpg_streamer -i "./input_uvc.so -f 10 -r 640x320 -n -y" -o "./output_http.so -w ./www -p 80" ถ้าไม่ติดข้อผิดพลาดใดๆจะได้ผลลัพธ์ตามรูปที่ 3.42

รูปที่ 3.42 ผลลัพธ์เมื่อรัน MJPEG Streamer

3.2.3.9 เข้าชมรูปที่ได้โดยเข้าไปที่เว็บเบราเซอร์ http://your-ip-address:8080 แทน your-ip-address ด้วย ip address ของตัวเอง ในที่นี้คือ 192.168.2.133 จะได้หน้าเว็บเบราเซอร์ตามรูปที่ 3.43



รูปที่ 3.43 หน้าเว็บเบราเซอร์ MJPEG Streamer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ประกอบอุปกรณ์

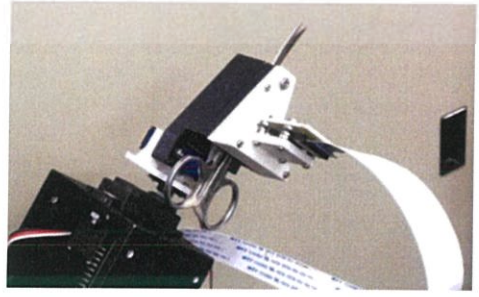
4.1.1 การประกอบแขนเข้ากับมอเตอร์ไฟฟ้าและเซอร์โวมอเตอร์ ดังรูปที่ 4.1 และ4.2

แขนส่วนที่ 1 (A) - ต่อกับมอเตอร์ไฟฟ้า 2 ตัว ตัวแรกจะใช้เพื่อหมุนแขนกลในทางซ้าย-ขวา มอเตอร์ไฟฟ้าส่วนตัวที่ 2 จะใช้เพื่อขยับแขนกลส่วนถัดไปให้ขึ้น-ลง

แขนส่วนที่ 2 (B) - ต่อกับเซอร์โวมอเตอร์เพื่อขยับแขนส่วนถัดไปให้ขึ้น-ลง

แขนส่วนที่ 3 (C) - ต่อกับเซอร์โวมอเตอร์เพื่อหมุนส่วนที่ใช้ตัด(กรรไกร)ให้หมุนซ้าย-ขวา

รูปที่ 4.1 A) แขนส่วนที่ 1 ต่อเข้ากับมอเตอร์ไฟฟ้า B) แขนส่วนที่ 2 ต่อเข้ากับเซอร์โวมอเตอร์ C) แขนส่วนที่ 3 ต่อเข้ากับเซอร์โวมอเตอร์



รูปที่ 4.2 แขนส่วนที่ 3 ต่อเข้ากับเซอร์โวมอเตอร์
และประกอบเข้ากับส่วนตัดและกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกทั้งส่วนที่ใช้ตัด จะใช้เซอร์โวมอเตอร์ในการควบคุมการตัดของกรรไกร ตามรูปที่ 4.3 และ 4.4 โดยเซอร์โวมอเตอร์จะติดอยู่กับกรรไกร โดยจะมีก้านจากเซอร์โวมอเตอร์มาขัดกับรางที่ติดไว้กับกรรไกร เมื่อเซอร์โวมอเตอร์เกิดการหมุนตั้งแต่ 1-180 องศา จะทำให้ก้านดังกล่าวเคลื่อนที่ไปตามรางดันให้กรรไกรเกิดการขยับเปิดออก และถ้าเซอร์โวมอเตอร์เกิดการหมุนกลับในทิศเดิมจะทำให้กรรไกรปิดกลับไป

รูปที่ 4.3 ส่วนที่ใช้ตัดติดกับเซอร์โวมอเตอร์

รูปที่ 4.4 เซอร์โวมอเตอร์ติดกับกรรไกร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 แขนกล และตัวควบคุม

ทำการประกอบแขนกล และตัวควบคุมส่วนต่างๆเข้าด้วยกัน จะเป็นไปตามรูปที่ 4.5 และ 4.6 ตามลำดับ

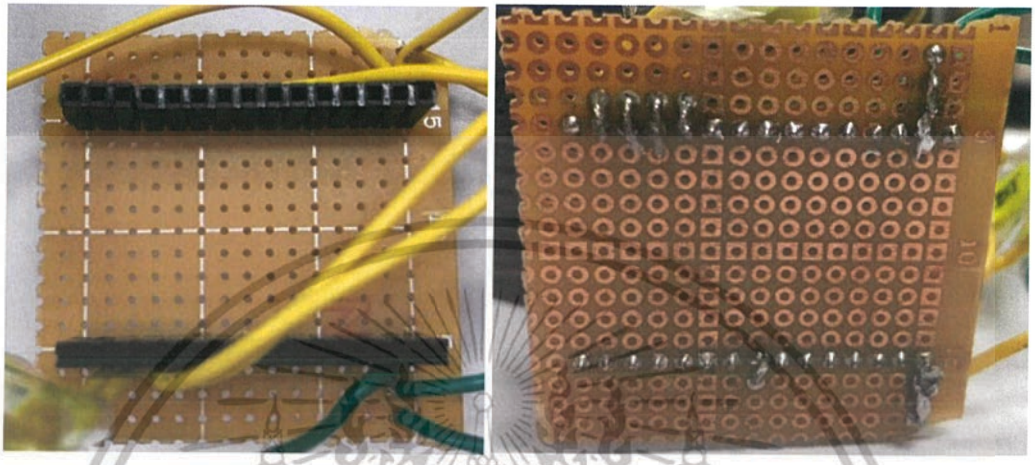
รูปที่ 4.5 แขนกลประกอบเข้าด้วยกัน

รูปที่ 4.6 ตัวควบคุม

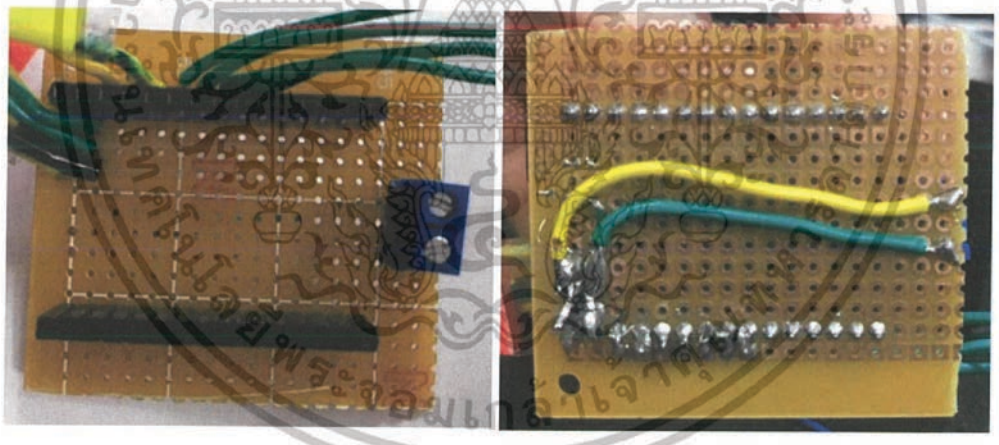
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 บัดกรีวงจร

นำอุปกรณ์มาบัดกรีลงบนบอร์ดไข่ปลาทั้ง 2 บอร์ด บอร์ดแรกสายเชื่อมต่อกับ joystick1, joystick2, button ก่อนนำไปต่อกับ ESP32 และบอร์ดที่2 สายเชื่อมต่อกับ servo1, servo2, servo3, motor1, motor2 ก่อนนำไปต่อกับ ESP32 ตามรูปที่ 4.7 และ 4.8



รูปที่ 4.7 บัดกรีสายบอร์ดที่1



รูปที่ 4.8 บัดกรีสายบอร์ดที่2

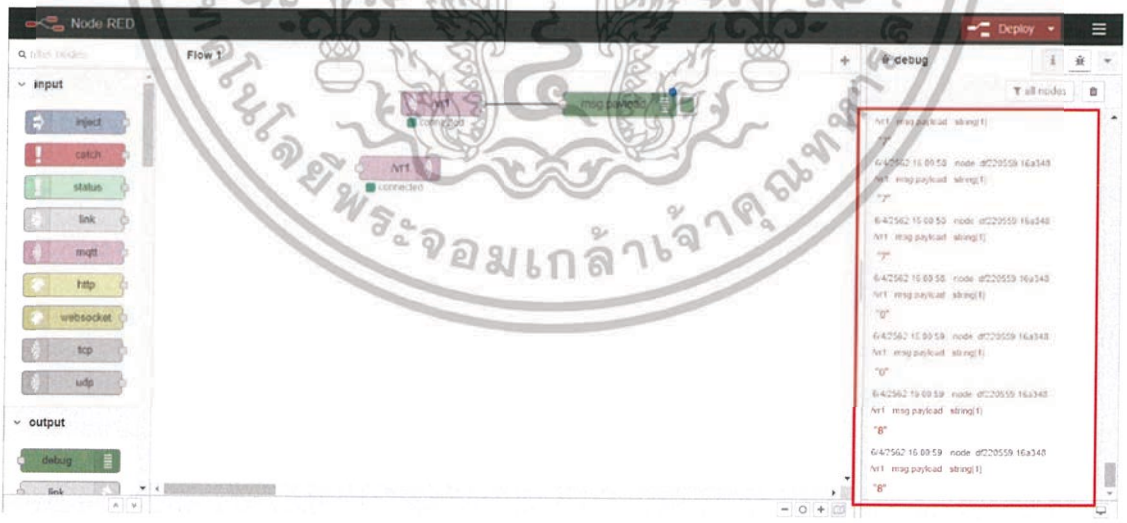
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ค่า Output ทางฝั่งควบคุม เมื่อทำการควบคุมอุปกรณ์ในทิศทางต่างๆ ดังตารางที่ 4.1

อุปกรณ์	การทำงาน	Output (publishออกไป)
Joystick1 และ Joystick2	หยุดนิ่ง	0
Joystick1	เลื่อนไปด้านขวา	1
Joystick1	เลื่อนไปด้านซ้าย	2
Joystick1	เลื่อนไปด้านหลัง	3
Joystick1	เลื่อนไปด้านหน้า	4
Joystick2	เลื่อนไปด้านหลัง	5
Joystick2	เลื่อนไปด้านหน้า	6
Joystick2	เลื่อนไปด้านขวา	7
Joystick2	เลื่อนไปด้านซ้าย	8
button	กดปุ่ม	9

ตารางที่ 4.1 ค่า Output ทางฝั่งควบคุม เมื่อทำการควบคุมอุปกรณ์ในทิศทางต่างๆ

4.4 ข้อมูลที่ publish ออกไปทางหน้าต่าง debug บน node-red ดังรูปที่ 4.9



รูปที่ 4.9 ข้อมูลที่ publish ออกไปแสดงทางหน้าต่าง debug บน node-red

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

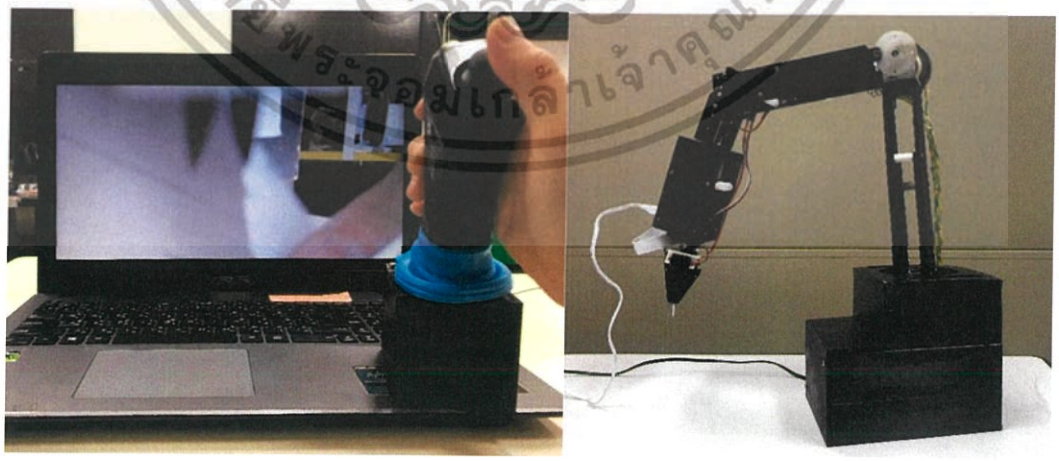
4.5 การทำงานของแขนกล เมื่อรับ Input เข้ามา ดังตารางที่ 4.2

Input (subscribeเข้ามา)	อุปกรณ์	การทำงาน
0	เซอร์โว1, เซอร์โว2, เซอร์โว 3,มอเตอร์1,มอเตอร์2	ไม่มีการเปลี่ยนแปลง
1	มอเตอร์1	หมุนตามเข็มนาฬิกา
2	มอเตอร์1	หมุนทวนเข็มนาฬิกา
3	มอเตอร์2	หมุนตามเข็มนาฬิกา
4	มอเตอร์2	หมุนทวนเข็มนาฬิกา
5	เซอร์โว1	หมุนไปทางขวา
6	เซอร์โว1	หมุนไปทางซ้าย
7	เซอร์โว2	หมุนไปทางขวา
8	เซอร์โว3	จาก100องศาหมุนกลับไป0

ตารางที่ 4.2 การทำงานของแขนกล เมื่อรับ Input เข้ามา

4.6 การควบคุมแขนกลเมื่อตัวควบคุมอยู่ต่างสถานที่กับแขนกล ดังรูปที่ 4.10

ทำการควบคุมแขนกลด้วยตัวควบคุมและมองภาพผ่านหน้าจอ แขนกลจะขยับตามการควบคุม



รูปที่ 4.10 การควบคุมแขนกลด้วยตัวควบคุมและมองภาพผ่านหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากการควบคุม จะเป็นไปตามตารางที่ 4.3

ฝั่งควบคุม	ฝั่งแขนกล
เลื่อนjoystickตัวล่างไปทางขวา	ส่วนที่ 1 หมุนขวา (0-360 องศา)
เลื่อนjoystickตัวล่างไปทางซ้าย	ส่วนที่ 1 หมุนซ้าย (0-360 องศา)
เลื่อนjoystickตัวล่างไปด้านหน้า	ส่วนที่ 2 เคลื่อนที่ลง (0-90 องศา)
เลื่อนjoystickตัวล่างไปด้านหลัง	ส่วนที่ 2 เคลื่อนที่ขึ้น (0-90 องศา)
เลื่อนjoystickตัวบนไปด้านหน้า	ส่วนที่ 3 เคลื่อนที่ลง (0-180 องศา)
เลื่อนjoystickตัวบนไปด้านหลัง	ส่วนที่ 3 เคลื่อนที่ขึ้น (0-180 องศา)
เลื่อนjoystickตัวบนไปด้านขวา	ส่วนที่ 4 หมุนขวา (0-180 องศา)
เลื่อนjoystickตัวบนไปด้านซ้าย	ส่วนที่ 4 หมุนซ้าย (0-180 องศา)
กดปุ่มด้านหน้า	กรรไกรทำการตัด (ตามองศาที่กำหนด)

ตารางที่ 4.3 ผลการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากงานวิจัยการสร้างแขนกลควบคุมทางไกลเพื่อใช้ในห้องผ่าตัดนี้ พบว่าสามารถทำการควบคุมแขนกลจากทางไกลได้จริง โดยผู้จัดทำได้ทำการออกแบบตัวควบคุมแยกกับตัวแขนกลซึ่งติดกรรไกรและกล้อง raspberry pi camera ที่เชื่อมอยู่กับ raspberry pi ซึ่งทำการคอมไพลเลอร์ไว้แล้ว ใช้งานร่วมกับไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WIFI เขียนโปรแกรมทางฝั่งควบคุมให้รับค่าจากการขยับตัวควบคุม ส่งค่าขึ้นเซิฟเวอร์ที่ได้ทำการสร้างขึ้น และนำค่าเหล่านั้นมาใช้ โดยเขียนโปรแกรมลงไมโครคอนโทรลเลอร์อีกตัวเพื่อใช้ควบคุมแขนกลซึ่งอยู่อีกที่หนึ่ง โดยฝั่งผู้ทำการควบคุมจะสามารถดูภาพทางจอแสดงผลผ่านระบบอินเตอร์เน็ต หากมีการนำไปพัฒนาเพื่อเพิ่มประสิทธิภาพจะสามารถเพิ่มความสะดวกให้กับบุคลากรทางการแพทย์ได้ในด้านต่างๆ และเพื่อให้บุคลากรทางการแพทย์สามารถเข้าถึงผู้ป่วยในพื้นที่ห่างไกลได้โดยทำการควบคุมแขนกล และทำหัตถการทางการแพทย์ผ่านตัวควบคุม และดูภาพผ่านจอแสดงผลภาพ อีกทั้งยังเป็นประโยชน์กับผู้ป่วยในพื้นที่ห่างไกลที่ศูนย์บริการทางการแพทย์ ยังไม่มีผู้เชี่ยวชาญเฉพาะทางสำหรับการรักษาโรคเฉพาะทาง ซึ่งไม่ต้องเดินทางไกลเพื่อมารับการรักษาที่โรงพยาบาลในเมือง จึงเป็นการช่วยลดค่าใช้จ่ายในการเดินทาง และประหยัดเวลา หรือกับบางโรคที่ต้องได้รับการรักษาในทันที การทำหัตถการทางการแพทย์ทางไกลจึงเป็นประโยชน์ได้ไม่น้อย

5.2 ปัญหาและแนวทางการแก้ปัญหา

5.2.1 ปัญหาทางด้านฮาร์ดแวร์

5.2.1.1 ในการทำแขนกลครั้งแรกได้ใช้เซอร์โวควบคุมการขึ้นลงในส่วนแรกที่ต่อกับฐาน พบว่าเซอร์โวที่ใช้ไม่สามารถยกแขนในส่วนบนขึ้นได้

แนวทางการแก้ปัญหา : เปลี่ยนเซอร์โวที่ใช้เป็นมอเตอร์ไฟฟ้ากระแสตรงที่มีความเร็วรอบต่ำ และแก้ไขแบบแขนกล เพื่อลดน้ำหนักลง

5.2.1.2 มอเตอร์ไฟฟ้ากระแสตรงที่ใช้ควบคุมการขึ้นลงในส่วนแรก(ส่วนที่ต่อกับฐาน)ยังไม่สามารถควบคุมการลงได้อย่างสมบูรณ์ เนื่องจากในส่วนถัดไปมีน้ำหนักมาก ทำให้มอเตอร์ต้านแรงลงไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการแก้ปัญหา : ทำการแก้ไขคำสั่งควบคุม และฮาร์ดแวร์ แต่ยังคงไม่สามารถควบคุมการลงได้อย่างสมบูรณ์ ดังนั้นควรเปลี่ยนตัวอุปกรณ์ที่มีความเหมาะสมกับการใช้งานมากกว่า

5.2.2 ปัญหาทางด้านซอฟต์แวร์

5.2.2.1 บอร์ด ESP32 เสียบ่อยครั้ง เนื่องจากทำการเชื่อมต่ออุปกรณ์หลายตัว และใช้กระแสไฟฟ้ามาก

แนวทางการแก้ปัญหา : เพิ่มกระแสไฟฟ้า

5.2.2.2 คำสั่งที่ใช้ในการเขียนโปรแกรมลงบอร์ด Arduino เพื่อควบคุมแขนกล บางคำสั่งไม่สามารถใช้งานได้กับบอร์ด ESP32

แนวทางการแก้ปัญหา : เปลี่ยนคำสั่งควบคุมการทำงาน และนำบอร์ด ESP32 เชื่อมต่อกับบอร์ด Arduino เพื่อใช้ช่วยประมวลผลในการทำงาน

5.3 ข้อเสนอแนะ

หากมีการใช้อุปกรณ์ที่มีคุณสมบัติตรงกับการใช้งาน อาจทำให้ได้ผลการวิจัยที่มีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] **Arduino คืออะไร ?** [ออนไลน์]. 2559 [ค้นเมื่อ 25 สิงหาคม 2561]. จาก https://poundxi.com/Arduino/.

[2] **มอเตอร์ (Motor) คืออะไร?** [ออนไลน์]. 2558 [ค้นเมื่อ 25 สิงหาคม 2561]. จาก http://www.psptech.co.th/page.

[3] **บทความการใช้งาน การใช้งาน L298N motor drive** [ออนไลน์]. 2560 [ค้นเมื่อ 25 สิงหาคม 2561]. จาก http://gamerbloggerport.blogspot.com/2017/09/l298n-motor-drive.html.

[4] **Servo Motor [EP1] : Servo Motor คืออะไร ? ใช้งานยังไง ?** [ออนไลน์]. 2561 [ค้นเมื่อ 25 สิงหาคม 2561]. จาก https://www.arduinospro.com/article/22/servo-motor-ep1-servo-motor/.

[5] **การใช้ Arduino UNO R3 กับ L298N ควบคุมมอเตอร์** [ออนไลน์]. 2561 [ค้นเมื่อ 25 สิงหาคม 2561]. จาก http://www.robotsiam.com/article/.

[6] **ESP8266 ตอนที่ 1 รู้จักกับ ESP และรุ่นที่นิยมใช้งาน** [ออนไลน์]. 2559 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://www.ioxhop.com/article/13/esp8266/.

[7] **การใช้งาน ESP8266 ในโหมด AP และการรับส่งข้อมูลผ่าน TCP** [ออนไลน์]. 2560 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://www.ioxhop.com/article/43/.

[8] **การส่งข้อมูลหากันระหว่าง Arduino to Nodemuc Esp8266 โดยผ่าน Serial** [ออนไลน์]. 2560 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://www.9arduino.com/article/17/.

[9] **ทำ Web server เพื่อรับข้อมูลจาก Sensor และควบคุมอุปกรณ์ต่างๆ ตอนที่ 1 การส่งข้อมูลผ่านหน้าเว็บ** [ออนไลน์]. 2561 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://lovecodingblogger.blogspot.com/2018/01/web-server-sensor-1.html

[10] **ทำ Web server เพื่อรับข้อมูลจาก Sensor และควบคุมอุปกรณ์ต่างๆ ตอนที่ 2 ส่งข้อมูลจากบอร์ด NodeMCU ขึ้น Web Server** [ออนไลน์]. 2561 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://lovecodingblogger.blogspot.com/2018/01/web-server-sensor-2-nodemcu-web-server.html

[11] **NodeMCU กับ IoT ตอนที่ 1 : NodeMCU คืออะไร** [ออนไลน์]. 2560 [ค้นเมื่อ 10 กันยายน 2561]. จาก http://sat2you.com/web/2017/01/31/nodemcu/.

[12] **พื้นฐานการใช้งาน Ethernet Shield กับ Arduino (ตอนที่ 1)** [ออนไลน์]. 2557 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://www.arduitronics.com/article/15/.

[13] **พื้นฐานการใช้งาน Ethernet Shield กับ Arduino (ตอนที่ 2)** [ออนไลน์]. 2558 [ค้นเมื่อ 10 กันยายน 2561]. จาก https://www.arduitronics.com/article/16/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[14] **ทดลองสร้าง Server บน Google Cloud Platform ฟรี** [ออนไลน์]. 2561 [ค้นเมื่อ 30 ตุลาคม 2561]. จาก https://medium.com/@gominkaewmanee/.

[15] **Protocol คืออะไร [ออนไลน์]. 2560** [ค้นเมื่อ 30 ตุลาคม 2561]. จาก https://www.mindphp.com/.

[15] **การติดตั้ง Mosquitto MQTT และการใช้งานกับ Nodemcu** [ออนไลน์]. 2561 [ค้นเมื่อ 30 ตุลาคม 2561]. จาก http://meetjoeblog.com/2018/04/08/mosquitto-mqtt-server-nodemcu-ep3-5-1/.

[16] **[Beginner] Node.js คืออะไร [ออนไลน์]. 2561** [ค้นเมื่อ 30 ตุลาคม 2561]. จาก https://medium.com/@settawatjanpuk/https-medium-com-settawatjanpuk-beginner-node-js-970383cc6e3a.

[17] **Node.js คืออะไร ? + เริ่มต้นใช้งาน Node.js** [ออนไลน์]. 2558 [ค้นเมื่อ 5 มกราคม 2561]. จาก https://devahoy.com/posts/getting-started-with-nodejs/.

[18] **เริ่มต้นใช้งาน Node-red ! ฉบับปี 2018–19** [ออนไลน์]. 2561 [ค้นเมื่อ 5 มกราคม 2561]. จาก https://medium.com/mmp-li/.

[19] **NodeMCU คืออะไร** [ออนไลน์]. 2561 [ค้นเมื่อ 20 มกราคม 2561]. จาก https://poundxi.com/nodemcu/.

[20] **ESP32 เบื้องต้น :: บทที่ 1 แนะนำ ESP32** [ออนไลน์]. 2561 [ค้นเมื่อ 2 มกราคม 2561]. จาก https://www.ioxhop.com/article/62/esp32/.

[21] **สตรีมมิ่งมีเดีย(Streaming Media technology)** [ออนไลน์]. 2558 [ค้นเมื่อ 25 มกราคม 2561]. จาก http://irrigation.rid.go.th/rid14/news58/exchange_news/105/105-7-1.pdf

[22] **How to install MJPG-Streamer** [ออนไลน์]. 2556 [ค้นเมื่อ 25 มกราคม 2561]. จาก http://irrigation.rid.go.th/rid14/news58/exchange_news/105/105-7-1.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติอุปกรณ์ต่างๆ

1. Micro Servo (SG90)

คุณสมบัติ

- No load speed: 0.12 seconds / 60 degrees (4.8V)
- Stall Torque: 1.6 kg / cm (4.8V)
- Operating temperature: -30 ~ +60 degrees Celsius
- Dead Set: 7 microseconds
- Operating voltage: 4.8V-6V
- Working current: less than 500mA
- Cable length: 180mm
- Size: 22mmx12.5mmx29.5mm
- Weight: 9 grams

2. Servo Motor (MG995 0-180 องศา)

คุณสมบัติ

- Type: MG995
- Size: About 40mm x 38mm x 20mm
- Weight: About 71g
- Speed (4.8v): 0.17sec/60degrees
- Speed (6V): 0.13sec/60degrees
- Stall Torque (4.8V): 13kg/cm
- Stall Torque(6.0V): 15kg/cm
- Towerpro-MG995 Standard Big Torque Metal Gears Servo: 1 Unit

3. DC Motor

คุณสมบัติ

- Voltage: 12V DC
- RPM: 50
- Reversibility: Reversible
- Speed: 5000r/min

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Rated torque: 12kg.cm
- Length of Motor(including spindle): 80.5mm
- Diameter: 37mm
- Shaft length: 15mm
- Shaft diameter: 6mm
- It is great for speed control .
- Replacement your electrical and testing equipment a new lease of life .
- Benefiting from high torque and low noise .
- This motor is solidly constructed .
- This motor suitable for most of high torque applications .
- Horse Power Cont.: 5W

4. Motor Drive Module L298N

คุณสมบัติ

- Dual H bridge Drive Chip : L298N
- แรงดันสัญญาณลอจิค : 5V Drive voltage: 5V-35V
- กระแสของสัญญาณลอจิค : 0-36mA
- กระแสขับมอเตอร์ : สูงสุดที่ 2A (เมื่อใช้มอเตอร์เดียว)
- กำลังไฟฟ้าสูงสุด : 25W
- ขนาด : 43 x 43 x 26 มิลลิเมตร
- น้ำหนัก : 26 กรัม
- มี Power Supply 5V ในตัว สามารถจ่ายไฟออกจากช่อง 5V (เพื่อจ่ายให้บอร์ด Arduino) ได้ เมื่อต่อไฟเลี้ยงเข้าที่ช่อง 12V

5. Arduino UNO

คุณสมบัติ

- ชิปไอซีไมโครคอนโทรเลอร์: ATmega328
- ใช้แรงดันไฟฟ้า: 5V
- รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ): 7 – 12V
- รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) : 6 – 20V
- พอร์ต Digital I/O: 14 พอร์ต (มี 6 พอร์ต PWM output)
- พอร์ต Analog Input: 6 พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กระแสไฟที่จ่ายได้ในแต่ละพอร์ต: 40mA
- กระแสไฟที่จ่ายได้ในพอร์ต 3.3V: 50mA
- พื้นที่โปรแกรมภายใน: 32KB พื้นที่โปรแกรม, 500B ใช้โดย Booloader
- พื้นที่แรม: 2KB
- พื้นที่หน่วยความจำถาวร (EEPROM): 1KB
- ความถี่คริสตัล: 16MHz
- ขนาด: 68.6x53.4 mm
- น้ำหนัก: 25 กรัม

6. ESP-WROOM-32

คุณสมบัติ

- ESP32 ทำงานแบบ Dual Core มี โปรเซสเซอร์ 2 ตัวทำงานได้พร้อมกัน
- มี Wi-Fi และ Bluetooth 4.0
- ทำงานแบบ 32 บิต
- ความถี่ Clock ความเร็วสูงสุดถึง 240 Mhz
- หน่วยความจำ RAM 512 kB
- มีขาทั้งหมด 30 ขา ข้างละ 15 ขา
- มีความสามารถหลายหลาย เช่น Capacitive Touch , Hall Sensor, ADCs , DAC , UART , SPI ,I2C และอื่น ๆ

7. Raspberry Pi 3 Model B

คุณสมบัติ

- Broadcom BCM2837 chipset running at 1.2 GHz
- 64-bit quad-core ARM Cortex-A53
- 802.11 b/g/n Wireless LAN
- Bluetooth 4.1 (Classic & Low Energy)
- Dual core Videocore IV® Multimedia co-processor
- 1 GB LPDDR2 memory
- Supports all the latest ARM GNU/Linux distributions and Windows 10 IoT
- microUSB connector for 2.5 A power supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1 x 10/100 Ethernet port
- 1 x HDMI video/audio connector
- 1 x RCA video/audio connector
- 4 x USB 2.0 ports
- 40 GPIO pins
- Chip antenna
- DSI display connector
- microSD card slot>
- Dimensions: 85 x 56 x 17 mm

8. Camera Module Board REV 1.3 5MP Webcam Video 1080p 720p

คุณสมบัติ

- Shot :1/4 5M
- Aperture : 2.9
- Focal length : 3.29
- FOV : 65 degree
- Sensor type : OmniVision OV5647 Color CMOS QSXGA (5-megapixel)
- Sensor size : 3.67 x 2.74 mm (1/4" format)
- Pixel Count : 2592 x 1944
- Pixel Size : 1.4 x 1.4 um
- Lens : f=3.6 mm, f/2.9
- Angle of View : 54 x 41 degrees
- Field of View : 2.0 x 1.33 m at 2 m
- Full-frame SLR lens equivalent : 35 mm
- Fixed Focus: 1 m to infinity
- Video : 1080p at 30 fps with codec H.264 (AVC)
- Up to 90 fps Video at VGA
- Board size : 25 x 24 mm (not including flex cable)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Micro USB Power Adapter 5V 2.5A

คุณสมบัติ

- แรงดันไฟฟ้าขาเข้า: AC 100-240V 50 / 60Hz
- เอาท์พุท: DC 5.0V 2500mA (2.5A)
- ความยาวสายเคเบิล: ประมาณ 10 ซม
- ป้องกันการลัดวงจร
- ป้องกันการโอเวอร์โหลด
- แรงดันไฟฟ้าคงที่ที่เปลี่ยน
- ฟิวส์ภายในเพื่อความปลอดภัยเป็นพิเศษ

10. Switching 12V 5A

คุณสมบัติ

- AC Input : 100-120V 60Hz / 200-220V 50Hz
- DC Output : 12V 5A
- วัสดุ: โลหะ/ aluminium bass
- การป้องกัน: Shortage , Overload & Over voltage protection
- การรับรอง: CCC/ FC / CE
- อุณหภูมิการใช้งาน: 0-40°C
- อุณหภูมิในการเก็บ: -20-60°C
- Ambient Humidity : 0-95% Non-Condensation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. DC Step Down Buck Converter Module (5A 5-36V เป็น 1.2V-32V)

คุณสมบัติ

- Input Voltage:5-36V
- Adjustable Output Voltage Range:1.2-32V
- Output Current:5A
- Output Power:75W
- Conversion Efficiency:96%
- With Thermal Protection And Short Circuit Protection
- Size:69*39*18mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้