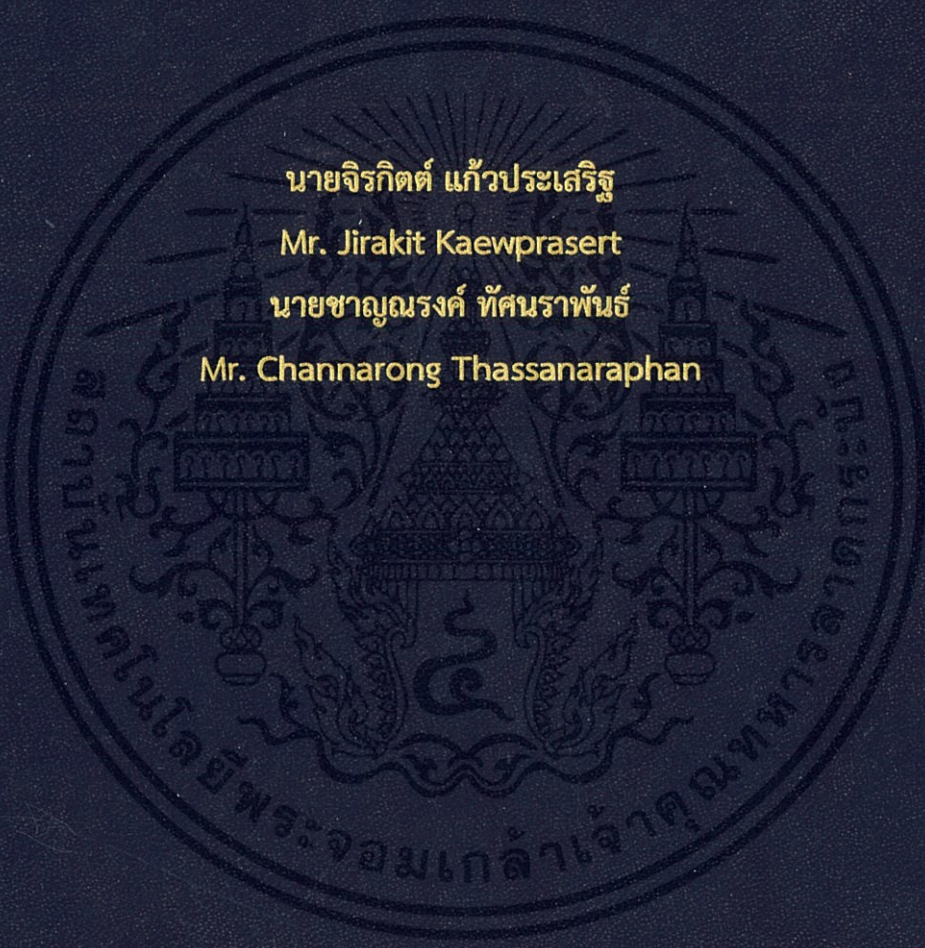


โดรนสังเกตการณ์เฝ้าระวังประจำที่
Intruder alert by Drone



นายจिरกิตต์ แก้วประเสริฐ
Mr. Jirakit Kaewprasert
นายชาญณรงค์ ทศนราพันธ์
Mr. Channarong Thassanaraphan

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

โดรนส่งเหตุการณ์เฝ้าระวังประจำที่

Intruder alert by Drone

โดย



นายจิรกิตต์ แก้วประเสริฐ

นายชาญณรงค์ ทศนราพันธ์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ พลศาสตร์ เลิศประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

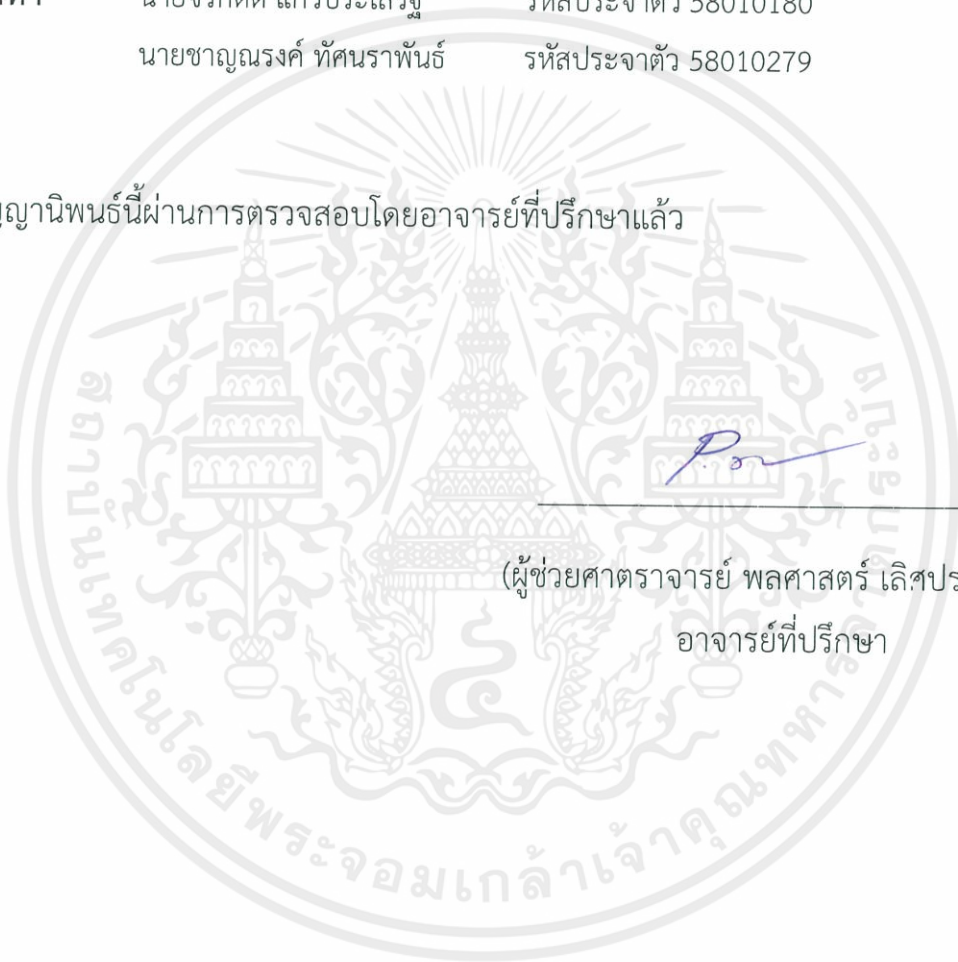
เรื่อง โดรนส่งเหตุการณ์เฝ้าระวังประจำที่

Intruder alert by Drone

ผู้จัดทำ นายจิรกิตต์ แก้วประเสริฐ รหัสประจำตัว 58010180

นายชาญณรงค์ ทศนราพันธ์ รหัสประจำตัว 58010279

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผู้ช่วยศาสตราจารย์ พลศาสตร์ เลิศประเสริฐ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์
นักศึกษา

โดรนสังเกตการณ์ใฝ่ระวังประจำที่
นายจิริกิตต์ แก้วประเสริฐ
รหัสประจำตัว 58010180
นายชาญณรงค์ ทศนราพันธ์
รหัสประจำตัว 58010279

ปริญญา

วิศวกรรมศาสตร์บัณฑิต

ภาควิชา

วิศวกรรมอิเล็กทรอนิกส์

ปีการศึกษา

2561

อาจารย์ที่ปรึกษาปริญญาานิพนธ์

ผศ.พลศาสตร์ เลิศประเสริฐ

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อศึกษาความรู้ด้านอากาศยานไร้คนขับหรือโดรนโดยใช้หน่วยประมวลผลควบคุมการบิน Pixhawk และโมดูลอื่นๆ โดยโดรนจะได้รับพลังงานจากแหล่งจ่ายไฟภาคพื้นดินด้วยการส่งแรงดันไฟฟ้ากระแสสลับและกระแสสลับที่มาจากอินเวอร์เตอร์ 2000W ทำให้สายไฟที่ใช้มีขนาดเล็กส่งขึ้นไปตามสายไฟด้วยความสูง 12 เมตร จากนั้นจะแปลงไฟฟ้ากระแสสลับเป็นกระแสตรงด้วยสวิตซ์ชิงเพาเวอร์ซัพพลาย 12V 50A เพื่อทำให้โดรนนั้นสามารถบินได้นานกว่าการใช้แบตเตอรี่แบบ Lipo ซึ่งต้องติดไปกับตัวลำและบินได้ 20-30 นาทีจึงต้องลงจอดเหมาะที่จะใช้ประโยชน์ได้เวลานานๆและสังเกตการณ์ประจำที่เท่านั้น ไม่สามารถเคลื่อนไหวได้มากกว่าความยาวสายไฟการทดลองบินด้วย 12 เมตรสามารถบินได้และกินพลังงาน 240 วัตต์และในโครงการนี้จะยกตัวอย่างการใช้ประโยชน์จากโดรนสายส่งโดยจะติดตั้ง module รับส่งข้อมูลวิทยุเพื่อที่จะทำให้โดรนเป็นตัวเสาส่งอากาศรับ-ส่งหลีกเลี่ยงการลดทอนสัญญาณจากสิ่งกีดขวาง สัญญาณจากตัวโมดูล NRF24L01 ที่ติดตั้งไว้กับเซนเซอร์ RCWL-0516 ที่เป็นเซนเซอร์ตรวจจับความเคลื่อนไหวโดยคลื่นไมโครเวฟ แล้วส่งผ่านมาแสดงผลที่จอ LCD ซึ่งเหมาะที่จะทำการตรวจจับผู้บุกรุกเป็นระบบใฝ่ระวังแจ้งเตือนเพื่อการลาดตระเวนได้ ทั้งนี้ระยะทางที่สามารถใช้งานได้ดีคือ 300 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis title	Intruder alert by drone
Students	Mr. Jirakit Kaewprasert Student ID 58010180 Mr. Channarong Thasanarapan Student ID 58010279
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2018
Thesis Advisor	Assistants Professor Polsart Lertprasert

ABSTRACT

The objective of our project was to study about unmanned aerial vehicle or drone using processor unit by Pixhawk flight controller board and other modules. Drone receive energy from energy ground station by AC voltage transmission wire. AC transmission makes use small and light wire from altitude at 12 metres Finally, AC convert to DC 12V 50A with a switching power supply. This method makes drone can fly longer than using a lipo battery wherewith operate 20-30 minutes. In order to make the drone as a transmitter-receiver antenna Signal from the NRF24L01 module that is installed with the RCWL-0516 sensor that is a microwave motion sensor. Then passed to display on the LCD screen. Which is suitable for detection intruder in area or alert surveillance area for security system and best range for operation is 300 metres.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาจากอาจารย์ที่ปรึกษาผู้ช่วยศาสตราจารย์พลศาสตร์ เลิศประเสริฐ ที่สนับสนุนอุปกรณ์และทุนในการทำโปรเจค อีกทั้งยังให้คำแนะนำและความรู้ จนบรรลุเป้าหมายที่ตั้งไว้

ขอขอบคุณร้าน Bighobbys ลำลูกกาคลอง 2 ที่ให้ความรู้เกี่ยวกับโดรน ทั้งยังแนะนำการใช้ อุปกรณ์และการประกอบโดรน จากประสบการณ์โดยตรงของเจ้าของร้าน

ขอขอบคุณ คุณ จักรพันธ์ พันโท ที่แนะนำความรู้การบิน และแนวทางการทำโปรเจคอากาศยานไร้คนขับรวมถึงสนับสนุน สถานที่ทดลองโครงการอีกด้วย

สุดท้ายนี้ขอกราบขอบพระคุณบิดาและมารดา ที่มอบกำลังใจและเงินสนับสนุนในการทำโครงการ รวมไปถึงเพื่อนๆที่คอยช่วยเหลือและแก้ปัญหาในการทำโครงการนี้

สำหรับคุณประโยชน์อันพึงเกิดจากโครงการเล่มนี้ ขอมอบให้แก่คณาจารย์ที่เคารพทุกท่าน ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังซึ่งเป็นสถานที่ให้ความรู้และประสบการณ์ในการทำโครงการฉบับนี้

จิรกิตต์ แก้วประเสริฐ
ชาณุณรงค์ ทัดศรพันธ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานการศึกษา.....	2
1.4 ขอบเขตของงานวิจัย.....	2
1.5 ประโยชน์ที่ได้รับ.....	2
บทที่ 2 หลักการและทฤษฎี.....	3
2.1 เทคโนโลยีโดรน (Drone).....	3
2.2 การทำงานของโดรน.....	3
2.2.1 การลอยตัวอยู่กับที่ (Hovering).....	3
2.2.2 การทำให้อากาศยานสามารถบินขึ้น-ลง (Throttle).....	4
2.2.3 การควบคุมอากาศยานเงยหรือก้ม (Pitch).....	5
2.2.4 การหมุนตัว (Yaw).....	5
2.3 การระบุพิกัดตำแหน่งด้วย GPS.....	6
2.4 ส่วนสร้างแรงยก.....	7
2.4.1 มอเตอร์ไฟฟ้ากระแสตรง.....	7
2.4.2 ชุดควบคุมความเร็วรอบ.....	9
2.5 แบตเตอรี่ (Battery).....	10
2.5.1 แบตเตอรี่แบบ NiMH (Nickel Metal Hydride).....	10
2.5.2 แบตเตอรี่แบบ LiPo (Lithium Polymer).....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
2.6 การควบคุมการบินแกน ROLL PITCH YAW.....	11
2.7 ชุดควบคุมการบิน.....	12
2.7.1 Inertial Measurement Unit (IMU).	11
2.7.2 เซ็นเซอร์วัดระยะทาง Ultrasonics.....	13
2.7.3 เซ็นเซอร์วัดความดันบรรยากาศ Barometer.....	13
2.8 ใบพัด (Propeller).....	13
2.8.1 ใบพัดไม้.....	14
2.8.2 ใบพัดคาร์บอนเสริมไนลอน หรือ ใบพัด APC.....	14
2.8.3 ใบพัดคาร์บอนไฟเบอร์.....	15
2.9 บอร์ดควบคุมการบิน.....	16
2.10 ชุดควบคุมสัญญาณไร้สาย.....	17
2.10.1 โมดูลดรับ-ส่งสัญญาณวิทยุ.....	17
2.10.2 รีโมทวิทยุบังคับ.....	18
2.11 สัญญาณ PWM และ PPM.....	18
2.11.1 สัญญาณ PWM.	18
2.11.2 สัญญาณ PPMสัญญาณ PPM.....	19
2.12 NRF24L01.....	20
2.12.1 GENERAL FEATURES.....	20
2.12.2 Mesh Network.....	22
2.13 RCWL-0516 เซนเซอร์ตรวจจับวัตถุ แบบคลื่นไมโครเวฟ.....	24
2.14 LCD Display.....	25
2.16 Arduino UNO.....	26
2.16.1 ส่วนที่เป็นฮาร์ดแวร์ (Hardware).....	26
2.16.2 ส่วนที่เป็นซอฟต์แวร์ (Software).....	26
บทที่ 3 วิธีดำเนินการวิจัย.....	27
3.1 การประกอบโดรนและการคำนึงถึงน้ำหนักโหลด.....	27
3.1.1 การประกอบโดรน.....	27
3.1.2 การตั้งค่าบอร์ด Flight Controller ด้วยโปรแกรม Mission Planner.....	29

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
3.1.3 การวัดแรงยกมอเตอร์และประเมนน้ำหนักโหลด.....	34
3.2 การออกแบบการตรวจจับความเคลื่อนไหวพร้อมส่งข้อมูลผ่านอุปกรณ์สื่อสารไร้สาย...37	37
3.2.1 ส่วนประกอบตัวส่งสัญญาณ.....	38
3.2.2 ส่วนประกอบตัวรับ – ส่งสัญญาณ.....	39
3.2.3 ส่วนประกอบตัวรับสัญญาณ.....	40
3.2.4 หลักการทำงานระบบรับส่งข้อมูลไร้สาย NRF24L01.....	42
3.2.5 การตั้งค่าอุปกรณ์ NRF24L01.....	43
3.2.6 การออกแบบระบบหลบหลีกวัตถุ.....	44
บทที่ 4 การทดลองและผลการทดลอง.....	47
4.1 การทดสอบกระแสและแรงดันเมื่อบินที่ความสูงต่างกัน.....	50
4.2 การทดสอบกระแสและแรงดันเมื่อบินทดสอบใช้งานจริง.....	52
4.3 การวัดระยะการรับส่งสัญญาณและคุณภาพของสัญญาณวิทยุในการรับส่งข้อมูลของ NRF24L01.....	52
4.4 การวัดระยะการตรวจจับของเซนเซอร์ RCWL-0516.....	53
4.5 การทดสอบระบบหลบหลีก.....	54
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	56
5.1 สรุปผลการทดลอง.....	56
5.2 วิเคราะห์ผลการทดลอง.....	57
บรรณานุกรม.....	58
ภาคผนวก.....	60

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงผู้พัฒนาโปรแกรมและบอร์ดควบคุมการบิน.....	17
2.2 ขาและหน้าที่ของแต่ละขา ของ NRF24L01.....	21
2.3 การต่อ NRF24L01 เข้ากับ Arduino.....	22
3.1 วัดแรงยกของมอเตอร์.....	35
4.1 การวัดระยะสัญญาณและความแรงของสัญญาณทดสอบในที่โล่ง (Open area).....	53



สารบัญรูปภาพ

รูปที่	หน้า
2.1 ตัวอย่างอากาศยานโดรน.....	3
2.2 การลอยตัวอยู่กับที่ (Hovering).....	3
2.3 การบินขึ้นของอากาศยานแบบสี่ใบพัด (Throttle).	4
2.4 การก้มของอากาศยานแบบสี่ใบพัด (Pitch).....	5
2.5 การหมุนตัวทางซ้ายของอากาศยานแบบสี่ใบพัด (Yaw).....	5
2.6 จีพีเอส โมดูล (GPS Module).	6
2.7 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้ากระแสตรง.....	8
2.8 ตัวอย่าง Brushed DC Motor.....	9
2.9 ตัวอย่าง Brushless DC Motor.....	9
2.10 ตัวอย่างชุดความเร็วรอบ (ESC).....	10
2.11 ตัวอย่างแบตเตอรี่แบบ NiMH (Nickel Metal Hydride).....	10
2.12 ตัวอย่างแบตเตอรี่แบบ LiPo (Lithium Polymer).....	11
2.13 แกน ROLL PITCH YAW.....	11
2.14 Inertial Measurement Unit (IMU).....	12
2.15 ตัวอย่างเซ็นเซอร์ HC-SR04.....	13
2.16 ตัวอย่างใบพัดไม้.....	14
2.17 ตัวอย่างใบพัดคาร์บอนเสริมไนลอน 4.....	14
2.18 ตัวอย่างใบพัดคาร์บอนเสริมไฟเบอร์.....	15
2.19 ตัวอย่างใบพัดแบบรอบจัด (Speed Propeller).....	15
2.20 ตัวอย่างใบพัดแบบรอบต่ำ (Slow Propeller).....	16
2.21 บอร์ดคอนโทรลเลอร์ Pixhawk 2.4.8.....	17
2.22 Telemetry 433 MHz.....	17
2.23 รีโมทบังคับวิทยุ Futaba.....	18
2.24 การเปรียบเทียบสัญญาณเพื่อให้ได้ PWM.....	18
2.25 สัญญาณ PWM ที่ Duty cycle ต่างๆ.....	19
2.26 คลื่น PPM-Sum.....	20
2.27 NRF24L01.....	21
2.28 ขาของ NRF24L01.....	22

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปร่าง

รูปที่	หน้า
2.29 ลักษณะการขยายกิ่งโมดูล.....	23
2.30 RCWL-0516 เซนเซอร์ตรวจจับวัตถุ แบบคลื่นไมโครเวฟ.....	24
2.31 การต่อ RCWL-0516 กับ Arduino.....	25
2.32 LCD module.....	26
2.33 Arduino UNO.....	26
3.1 บล็อกไดอะแกรมการต่ออุปกรณ์.....	28
3.2 โดรนเมื่อประกอบจนเสร็จสิ้นแล้ว.....	28
3.3 การตั้งค่าเฟรมของโดรน.....	29
3.4 ขั้นตอนการคาริเบท Accelerometer.....	29
3.5 การจัดโดรนเพื่อการคาริเบททิศทาง.....	30
3.6 หน้าต่างโปรแกรมขณะทำการคาริเบท Compass.....	30
3.7 หน้าต่างขณะ Radio Calibration.....	31
3.8 รีโมทวิทยุ Flysky รุ่น FS – i6A.....	31
3.9 การตั้งค่าโหมด.....	32
3.10 การตั้งค่า Flight mode.....	33
3.11 การตั้งค่า PID.....	34
3.12 การต่อ Power Module เก็บข้อมูลกระแสและแรงดัน.....	34
3.13 การวัดแรงยกของมอเตอร์แต่ละตัว.....	34
3.14 Level ของรีโมทวิทยุ RC.....	35
3.15 น้ำหนักของโดรนที่ใช้ 2,624 กรัม.....	38
3.16 แสดงกราฟแรงดันและกระแสจากโปรแกรม Mission Planner.....	38
3.17 แรงดันต่ำสุดสวิตชิงพาวเวอร์ซัพพลายที่ 11.05V.....	39
3.18 แรงดันต่ำสุดสวิตชิงพาวเวอร์ซัพพลายที่ 15.13V.....	39
3.19 บล็อกไดอะแกรมการทำงานของ Mesh network.....	37
3.20 การต่ออุปกรณ์ตัวส่งสัญญาณ.....	38
3.21 Schematics ของตัวส่งสัญญาณ.....	38
3.22 ชุดตัวส่งสัญญาณ.....	38

สารบัญญรูปภาพ

รูปที่	หน้า
3.23 การต่ออุปกรณ์ตัวรับ – ส่งสัญญาณ.....	39
3.24 Schematics ของตัวรับ – ส่งสัญญาณ.....	39
3.25 ชุดตัวรับ – ส่งสัญญาณ.....	40
3.26 การต่ออุปกรณ์ตัวรับสัญญาณ.....	40
3.27 Schematics ของตัวรับสัญญาณ.....	41
3.28 ชุดรับสัญญาณ.....	41
3.29 บล็อกไดอะแกรมแสดงการรับส่งข้อมูลตรวจจับการเคลื่อนไหว.....	42
3.30 จอแสดงผลเมื่อได้รับสัญญาณ.....	43
3.31 โมดูล NRF24L01 + เสาอากาศรับส่ง(PA).....	43
3.32 ไดอะแกรมการต่ออุปกรณ์ HC-SR04 และ Arduino Nano เข้ากับ Pixhawk.....	44
3.33 แสดงผลการตอบสนอง Pulse ของ Joystick	45
3.34 ภาพระยะทางของโดรนระหว่างสิ่งกีดขวาง.....	46
4.1 การทดลองบินวัดกระแสและแรงดันพร้อมแสดงหน้าตา Mission Planner ขณะใช้งาน.....	47
4.2 กราฟระดับความสูงโดรนที่ไม่เกิน 8 เมตร.....	48
4.3 กราฟแรงดันและกระแสที่ความสูงโดรนไม่เกิน 8 เมตร.....	48
4.4 กราฟระดับความสูงโดรนที่ 8 เมตร 12 เมตร และ 21 เมตร.....	49
4.5 กราฟแรงดันและกระแสที่ความสูง 8 เมตร 12 เมตร และ 21 เมตร.....	49
4.6 ภาพการทดลองและอุปกรณ์ Inverter และ lipo – Battery.....	50
4.7 กราฟความสูงของโดรนบินสูงสุด 12 เมตร	51
4.8 กราฟกราฟแรงดันและกระแสที่ความสูง 12 เมตร.....	51
4.9 แนวคิดการใช้งานและการทดลอง.....	52
4.10 คุณภาพของสัญญาณ.....	53
4.11 เซนเซอร์ RCWL-0516.....	53
4.12 ทดลองการทำงานของเซนเซอร์.....	54
4.13 การเปลี่ยนแปลงของคลื่นเมื่อมีการเคลื่อนไหว.....	54
4.14 กราฟ RC input ที่มาจาก Arduino ของเซนแนลที่ 1 (ด้านหน้า-ด้านหลัง).....	55
4.15 กราฟ RC input ที่มาจาก Arduino ของเซนแนลที่ 2 (ด้านซ้าย-ด้านขวา).....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันโดรนมีการใช้งานกันอย่างหลากหลายทั้งใช้งานในทางทหาร สำหรับบินสำรวจไปยังพื้นที่หรือปฏิบัติการแทนนักบินในพื้นที่ที่เสี่ยงอันตรายแต่ในการปฏิบัติงานในแต่ละครั้งจำใช้เวลาในการบินได้น้อยเนื่องจากแบตเตอรี่มีความจุที่น้อยและมอเตอร์กินกระแสที่มากเลยทำให้ในการบินแต่ละครั้งใช้เวลาประมาณ 20-30 นาที จึงต้องนำโดรนลงจอดเพื่อเปลี่ยนแบตเตอรี่ในการใช้งานโดรนอีกรูปแบบหนึ่งที่น่าสนใจและใช้งานในปัจจุบันคือ โดรนที่รับพลังงานผ่านสายส่ง ซึ่งจะได้รับพลังงานผ่านสายส่งทำให้สามารถใช้งานได้นานและไม่จำเป็นต้องลงจอดเพื่อเปลี่ยนแบตเตอรี่บ่อยๆ ปัจจุบันได้มีการประยุกต์ใช้งานในการถ่ายทอดสดกีฬาหรือการสำรวจโรงงาน การใช้งานดับเพลิงและอื่นๆ กลุ่มของผู้จัดทำได้จึงให้ความสนใจจึงได้เพิ่มแหล่งจ่ายไฟตรงที่สวิตซ์ชิงเพาเวอร์ซัพพลาย ที่ได้รับพลังงานไฟฟ้ากระแสสลับจากอินเวอร์เตอร์เพื่อให้น้ำหนักของสายไฟนั้นลดลงแทนที่จะจ่ายพลังงานด้วยไฟฟ้ากระแสตรงซึ่งน้ำหนักสายไฟจะหนักทั้งหมดนี้เพื่อที่จะทำให้โดรนสามารถบินได้ตลอดเวลาที่มีน้ำหนักที่เหมาะสมในการใช้งานแต่ละครั้ง ในการตั้งฐานตามแนวป่า เราจะวางเซนเซอร์ตรวจจับการเคลื่อนไหวรอบฐาน ไว้ตรวจจับสิ่งมีชีวิตที่เข้ามาบริเวณฐานที่ตั้ง เพื่อให้สัญญาณการส่ง ถูกลดทอนจากสิ่งกีดขวางน้อยที่สุดจึงส่งสัญญาณไปยังโดรนซึ่งติดตั้งเป็นตัวรับ-ส่งสัญญาณ และส่งสัญญาณกลับมาที่เสาอากาศด้านล่างผ่านจอแสดงผล

1.2 วัตถุประสงค์ของการศึกษา

- 1.2.1 เป็นการฝึกการปฏิบัติงานของคณะผู้จัดทำเพื่อเป็นพื้นฐานการปฏิบัติงานต่อไปในอนาคต รวมถึงการฝึกการทำงานเป็นหมู่คณะ
- 1.2.2 นำผลงานจากโครงการไปใช้ในการทำงานทางอิเล็กทรอนิกส์ต่อไป
- 1.2.3 บูรณาการความรู้ที่ได้จากการเรียนการสอนในภาคทฤษฎี นำมาใช้ในการปฏิบัติงานในขั้นต้น
- 1.2.4 เพื่อศึกษาวิธีการเขียนโปรแกรม
- 1.2.5 เพื่อที่จะทำเซนเซอร์ตรวจจับการเคลื่อนไหว และ โมดูลเสาอากาศทำงานเข้ากันได้
- 1.2.6 เป็นการศึกษาความรู้ทางด้านการพัฒนาอากาศยานไร้คนขับในเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 สมมุติฐานการศึกษา

- 1.3.1 โดรนสามารถบินได้เป็นเวลานานมากกว่าการใช้แบตเตอรี่
- 1.3.2 เซนเซอร์สามารถใช้งานได้แล้ว ส่งมายังจอแสดงผลการทำงานของเซนเซอร์ได้
- 1.3.3 สามารถเป็นแนวทางการพัฒนาต่อไปได้
- 1.3.4 สามารถนำไปใช้งานได้จริง

1.4 ขอบเขตของงานวิจัย

- 1.4.1 โดรนสามารถขึ้นบินได้โดยใช้แหล่งจ่ายพลังงานจากพื้นดินด้วยความสูง 12 เมตร
- 1.4.2 โดรนสามารถปฏิบัติงานได้เป็นเวลานานและมีประสิทธิภาพ
- 1.4.3 เซนเซอร์ตรวจจับการเคลื่อนไหวใช้งานได้ด้วยรัศมีการตรวจจับ 3 เมตร
- 1.4.4 การรับ – ส่งสัญญาณมีประสิทธิภาพในการทำงานไม่เกิน 500 เมตร

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถออกแบบและสร้างชุดรับ – ส่งสัญญาณตรวจจับการเคลื่อนไหวให้สามารถทำงานได้อย่างมีประสิทธิภาพ ตามวัตถุประสงค์ของผู้จัดทำที่ต้องการและสามารถนำความรู้ที่ได้มาประยุกต์ใช้สร้างผลงานที่สนใจได้

บทที่ 2

หลักการและทฤษฎี

2.1 เทคโนโลยีโดรน (Drone)

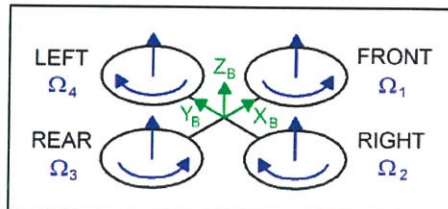
โดรน (Drone) คือ เครื่องบินที่สามารถเคลื่อนที่ได้ทั้ง 3 แนวแกน โดยที่มี 4 ใบพัดวางพาดกันเป็นกากบาท โดยใบพัดทั้ง 4 นั้นจะเป็นอิสระต่อกัน ในปัจจุบันนิยมนำมาประยุกต์ใช้ในการเก็บภาพถ่ายจากมุมสูง เพื่อใช้ในธุรกิจประชาสัมพันธ์ การจัดหาแผนที่ของสถานที่ท่องเที่ยว การสำรวจในพื้นที่ที่ประสบภัยทางธรรมชาติต่างๆ เพื่อเป็นการลดความสูญเสียต่อชีวิตและทรัพย์สินและโดรนถือเป็นหนึ่งเทคโนโลยีที่น่าจับตามองอยู่ในขณะนี้ เพราะทำหน้าที่ในการบังคับเครื่องบินแทนมนุษย์ ทั้งนี้บทบาทของ โดรน หรือเรียกอีกอย่างว่า อากาศยานไร้คนขับ กำลังเข้ามามีอิทธิพลต่อรูปแบบการทำธุรกิจในวันนี้ ต่างจากอดีตที่โดรนถูกใช้ในการทหารและภารกิจป้องกันประเทศเป็นหลัก โดยผลการศึกษาของ PwC's CEO pulse ที่ผ่านมาระบุว่า 64% ของผู้บริหารทั่วโลกเชื่อว่า ในอนาคตเทคโนโลยีหุ่นยนต์ (Robotics) จะยิ่งถูกนำมาผนวกอยู่ในรูปแบบการทำธุรกิจและถูกประยุกต์ใช้อย่างแพร่หลายครอบคลุมแทบทุกสายอุตสาหกรรม



รูปที่ 2.1 ตัวอย่างอากาศยานโดรน

2.2 การทำงานของโดรน

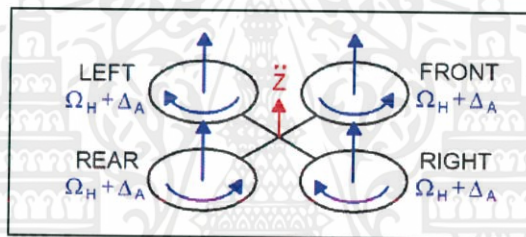
2.2.1 การลอยตัวอยู่กับที่ (Hovering)



รูปที่ 2.2 การลอยตัวอยู่กับที่ (Hovering)

ทำได้โดยการควบคุมให้ความเร็วใบพัดทั้งสี่มีความเร็วที่เท่ากันเพื่อสร้างโมเมนต์บิดที่เกิดจากด้านหนึ่งหักล้างกับโมเมนต์บิดที่เกิดอีกด้านหนึ่ง ตามที่แสดงในภาพ โดยกำหนดแกน X ชี้ไปในทิศทางด้านหน้าของอากาศยาน แกน Z อยู่ในทิศทางที่ชี้ขึ้นด้านบน และแกน Y เป็นไปตามกฎมือขวา สำหรับชื่อของใบพัดทั้งสี่ก็จะเป็น ใบพัดที่ 1 ใบพัดที่ 3 ใบพัดที่ 2 ใบพัดที่ 4 เพื่อให้อากาศยานลอยหยุดอยู่นิ่งได้ อุปกรณ์ควบคุมจะทำการควบคุมใบพัดที่อยู่ในแนวเส้นทแยงมุมกันหมุนในทิศทางเดียวกันด้วยความเร็วเท่ากัน โดยใบพัดทั้งสี่จะมีความเร็วรอบเท่ากัน โดยใบพัดทั้งสี่จะมีความเร็วรอบเท่ากัน ด้วย Hove speed Ω_H นั่นคือ $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4$ การหมุนของใบพัดเช่นนี้จะทำให้โมเมนต์รอบแกน Z ของอากาศยานหักล้างกันไปหมด ทำให้อากาศยานหยุดนิ่ง ไม่มีการหมุนรอบแกนใดๆ เมื่อแรงยกที่เกิดจากใบพัดทั้งสี่เท่ากับน้ำหนักของอากาศยาน ก็จะทำให้อากาศยานลอยนิ่งได้

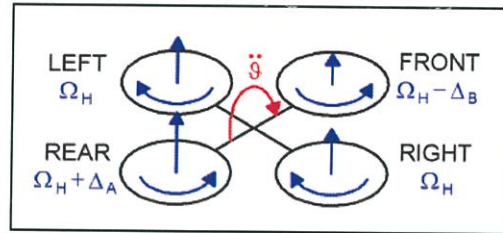
2.2.2 การทำให้อากาศยานสามารถบินขึ้น-ลง (Throttle)



รูปที่ 2.3 การบินขึ้นของอากาศยานแบบสี่ใบพัด (Throttle)

จากการที่อากาศยานหยุดนิ่งในอากาศได้ เนื่องจากแรงยกเท่ากับน้ำหนักและความเร็วรอบของใบพัดทั้งสี่เท่ากัน การที่จะให้อากาศยานยกตัวขึ้นจะสามารถเพิ่มความเร็วรอบให้มากขึ้นตามที่แสดงในรูป โดยเพิ่มความเร็วรอบของใบพัดทั้งสี่ให้เป็น $\Omega_H + \Delta_A$ เท่ากันทั้งหมดการกระทำเช่นนี้จะทำให้โมเมนต์รอบแกน Z ยังคงเท่ากับศูนย์แต่แรงใน แกน Z จะมากขึ้นทำให้แรงยกมากกว่าน้ำหนักก็จะทำให้อากาศยานยกตัวขึ้นต่อเนื่องด้วยความเร็วเท่ากับ Z ตามที่แสดงในรูปส่วนในกรณีเราต้องการจะลดระดับความสูง ก็จะทำในลักษณะเดียวกัน เพียงแต่ให้ค่าความเร็วรอบต่ำลงเป็น $\Omega_H + \Delta_A$ ก็จะทำให้แรงยกมีค่าน้อยกว่าน้ำหนักอากาศยาน จะค่อยๆ เคลื่อนตัวต่ำ ลงด้วยอัตราเร่ง Z ด้วยวิธีการนี้เราสามารถที่จะควบคุมการบินขึ้นและลงของอากาศยานได้

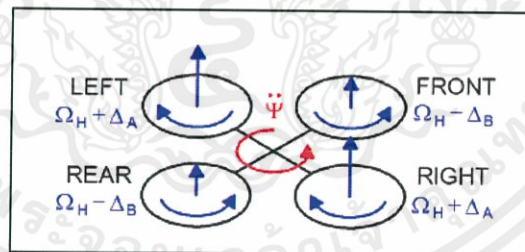
2.2.3 การควบคุมอากาศยานเงยหรือก้ม (Pitch)



รูปที่ 2.4 การก้มของอากาศยานแบบสี่ใบพัด (Pitch)

เพื่อเข้าใจการเคลื่อนที่ของอากาศยานปีกหมุนได้ดีขึ้น ในขณะที่อากาศยานอยู่ในลักษณะก้มหน้าลง แรงยกที่ได้จากใบพัดนั้นจะไม่อยู่ในแนวตั้งอีกต่อไป แต่จะอยู่ในแนวที่ทำมุมเอียงเล็กน้อยกับแนวตั้ง ทำให้แรงจากใบพัดแตกออกได้สองแนวคือ แนวตั้ง และในแนวระดับ ซึ่งแรงในแนวนี้จะทำให้หน้าที่ขับเคลื่อนอากาศยานไปข้างหน้า ในขณะที่แรงในแนวตั้งจะทำหน้าที่ สมดุลกับน้ำหนัก เพื่อให้อากาศยานลอยตัวในระดับความสูงที่คงที่ จะเป็นการแสดงการเคลื่อนที่ ของอากาศยานไปข้างหน้า จะเห็นว่าในขณะที่เคลื่อนที่ไปข้างหน้า นั้น อากาศยานจะอยู่ในลักษณะ ก้มลง คือใบพัดหลังจะสูงกว่าใบพัดหน้าเล็กน้อย ซึ่งจะทำให้แรงที่ได้จากใบพัดสร้างทั้งแรงยก เพื่อให้อากาศยานลอยตัวอยู่ได้และแรงผลักเพื่อให้อากาศยานเคลื่อนที่ไปข้างหน้าพร้อมๆกัน

2.2.4 การหมุนตัว (Yaw)



รูปที่ 2.5 การหมุนตัวทางซ้ายของอากาศยานแบบสี่ใบพัด (Yaw)

เป็นการกำหนดให้อากาศยานหมุนตัวรอบแกน Z ของอากาศยาน ซึ่งสามารถทำได้โดยการกำหนดให้ความเร็วใบพัดหน้าเท่ากับใบพัดหลังและมีความเร็วต่ำกว่าใบพัดซ้ายและใบพัดขวาเพื่อให้แรงบิดทางด้านซ้ายหรือขวามากกว่าด้านหน้าหรือด้านหลัง จึงทำให้เครื่องบินหมุนตัว รอบแกน Z ด้วยความเร็วเชิงมุมเท่ากับ $\dot{\psi}$ ตามที่แสดงในรูป การควบคุมนี้จะทำให้อากาศยานหมุนตัวกลับหน้าหลังได้

2.3 การระบุพิกัดตำแหน่งด้วย GPS

GPS คือ ระบบระบุตำแหน่งบนพื้นโลก ย่อมาจากคำว่า Global Positioning System ชื่อเต็มของระบบนี้คือ NAVSTAR Global Positioning System คำว่า NAVSTAR เป็นอักษรย่อมาจาก Navigation Satellite Timing and Ranging ภาคของคำว่าดาวเทียมสำหรับนำร่อง คือ ระบบที่ระบุตำแหน่งทุกแห่งบนโลก จากกลุ่มดาวเทียม 24 ดวง ที่โคจรรอบโลก ในระดับสูงที่พ้นจากคลื่นวิทยุรบกวนของโลกและวิธีการที่สามารถให้ความถูกต้องเพียงพอที่จะใช้ชี้บอกตำแหน่งได้ทุกแห่งบนโลก ตลอดเวลา 24 ชั่วโมง จากการนำมาใช้งานจริงจะให้ความถูกต้องสูง โดยที่ความคลาดเคลื่อนมาตรฐานของตำแหน่งทางราบต่ำกว่า 50 เมตร และถ้าเป็นแบบวิธี "อนุพันธ์" (Differential) จะให้ความถูกต้องถึงระดับเซนติเมตรจากการพัฒนาทางด้านอุปกรณ์คอมพิวเตอร์ทำให้สามารถผลิตเครื่องรับ GPS ที่มีขนาดลดลงและมีราคาถูกลงกว่าเครื่องรับระบบ TRANSIT เดิม

จีพีเอส โมดูล (GPS Module) เป็นเครื่องรับสัญญาณ GPS ตัวรับสัญญาณ GPS นั้นจะสามารถระบุตำแหน่งได้แม่นยำ ขึ้นอยู่กับจำนวนดาวเทียม ที่สามารถรับสัญญาณได้ในขณะนั้น การระบุพิกัดขั้นต่ำที่ต้องการดาวเทียมอย่างน้อย 3 ดวง เพื่อระบุตำแหน่งในแกน X , Y , Z (X คือข้อมูลว่าดาวเทียมนี้ คือ ตัวไหน, Y คือ ตำแหน่งของดาวเทียม และ Z คือ เวลาที่ส่งข้อมูลในขณะนี้) ให้ได้ตำแหน่งที่ถูกต้อง โดยตัวเครื่อง GPS จะนำสัญญาณนาฬิกา ที่ได้รับจากดาวเทียมทั้ง 3 ดวง มาคำนวณระยะห่าง ของเครื่องรับสัญญาณ GPS กับดาวเทียมเพื่อแจ้งพิกัดปัจจุบัน



รูปที่ 2.6 จีพีเอส โมดูล (GPS Module)

2.4 ส่วนสร้างแรงยก

2.4.1 มอเตอร์ไฟฟ้ากระแสตรง

การทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง มีแรงดันไฟตรงจ่ายผ่านแปรงถ่านไปคอมมิวเตเตอร์ ผ่านไปให้ขดลวดตัวนำที่อาร์เมเจอร์ ทำให้ขดลวดอาร์เมเจอร์เกิดสนามแม่เหล็กไฟฟ้าขึ้นมา ทางด้านซ้ายมือเป็นขั้วเหนือ (N) และด้านขวาเป็นขั้วใต้ (S) เหมือนกับขั้วแม่เหล็กถาวรที่วางอยู่ใกล้ๆ เกิดอำนาจแม่เหล็กผลักดันกัน อาร์เมเจอร์หมุนไปในทิศทางตามเข็มนาฬิกา พร้อมกับคอมมิวเตเตอร์หมุนตามไปด้วย แปรงถ่านสัมผัสกับส่วนของคอมมิวเตเตอร์ เปลี่ยนไปไปอีกปลายหนึ่งของขดลวด แต่มีผลทำให้เกิดขั้วแม่เหล็กที่อาร์เมเจอร์เหมือนกับขั้วแม่เหล็กถาวรที่อยู่ใกล้ๆ อีกครั้ง ทำให้อาร์เมเจอร์ยังคงถูกผลักให้หมุนไปในทิศทางตามเข็มนาฬิกาตลอดเวลา

ส่วนประกอบหลักๆของมอเตอร์ไฟฟ้ากระแสตรง ประกอบด้วยส่วนต่างๆ ดังนี้

1. ขดลวดสนามแม่เหล็ก (Field Coil) คือ ขดลวดที่ถูกพันอยู่กับขั้วแม่เหล็กที่ยึดติดกับโครงมอเตอร์ ทำหน้าที่กำเนิดขั้วแม่เหล็กขั้วเหนือ (N) และขั้วใต้ (S) แทนแม่เหล็กถาวร ขดลวดที่ใช้เป็นขดลวดอาบนํ้ายานวน สนามแม่เหล็กจะเกิดขึ้นเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์

2. ขั้วแม่เหล็ก (Pole Pieces) คือ แกนสำหรับรองรับขดลวดสนามแม่เหล็กถูกยึดติดกับโครงมอเตอร์ด้านใน ขั้วแม่เหล็กทำมาจากแผ่นเหล็กอ่อนบางๆ อัดซ้อนกัน (Lamination Sheet Steel) เพื่อลดการเกิดกระแสไหลวน (Eddy Current) ที่จะทำให้ความเข้าของสนามแม่เหล็กลดลง ขั้วแม่เหล็กทำหน้าที่ให้กำเนิดขั้วสนามแม่เหล็กมีความเข้มสูงสุด แทนขั้วสนามแม่เหล็กถาวร ผิวด้านหน้าของขั้วแม่เหล็กทำให้โค้งรับกับอาร์เมเจอร์พอดี

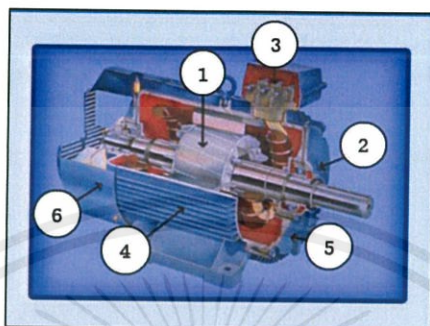
3. โครงมอเตอร์ (Motor Frame) คือ ส่วนเปลือกหุ้มภายนอกของมอเตอร์ และยึดส่วนอยู่กับที่ (Stator) ของมอเตอร์ไว้ภายในร่วมกับฝาปิดหัวท้ายของมอเตอร์ โครงมอเตอร์ทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กระหว่างขั้วแม่เหล็กให้เกิดสนามแม่เหล็กครบวงจร

4. อาร์เมเจอร์ (Armature) คือ ส่วนเคลื่อนที่ (Rotor) ถูกยึดติดกับเพลา (Shaft) และรองรับการหมุนด้วยที่รองรับการหมุน (Bearing) ตัวอาร์เมเจอร์ทำจากเหล็กแผ่นบางๆ อัดซ้อนกัน ถูกเจาะร่องออกเป็นส่วนๆ เพื่อไว้พันขดลวดอาร์เมเจอร์ (Armature Winding) ขดลวดอาร์เมเจอร์เป็นขดลวดอาบนํ้ายานวน ร่องขดลวดอาร์เมเจอร์จะมีขดลวดพันอยู่และมีลิมไฟเบอร์อัดแน่นยึดขดลวดอาร์เมเจอร์ไว้ ปลายขดลวดอาร์เมเจอร์ต่อไว้กับคอมมิวเตเตอร์ อาร์เมเจอร์ผลักดันของสนามแม่เหล็กทั้งสอง ทำให้อาร์เมเจอร์หมุนเคลื่อนที่

5. คอมมิวเตเตอร์ (Commutator) คือ ส่วนเคลื่อนที่อีกส่วนหนึ่ง ถูกยึดติดเข้ากับอาร์เมเจอร์และเพลาร่วมกัน คอมมิวเตเตอร์ทำจากแท่งทองแดงแข็งประกอบเข้าด้วยกันเป็นรูปทรงกระบอก แต่ละแท่งทองแดงของคอมมิวเตเตอร์ถูกแยกออกจากกันด้วยฉนวนไมก้า (Mica) คอมมิวเตเตอร์ทำหน้าที่เป็นขั้วรับแรงดันไฟตรงที่จ่ายมาจากแปรงถ่าน เพื่อส่งไปให้ขดลวดอาร์เมเจอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



รูปที่ 2.7 ส่วนประกอบหลักๆ ของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์กระแสตรงแบบมีแปรงถ่าน (Brush DC Motor)

มอเตอร์ไฟฟ้ากระแสตรงประเภทนี้เป็นมอเตอร์แบบง่ายที่สุด มีใช้งานมากกว่า 100 ปี กำลังที่เกิดขึ้นได้มาจากแบตเตอรี่ไปสู่มอเตอร์โดยผ่านสายไฟ โดยกระแสไฟฟ้าจะผ่านแปรงถ่านไปสู่ คอมมิวเตเตอร์ ที่ทำจากทองแดง และจะผ่านไปยังขดลวดทองแดง และกลับมายังแปรงถ่านอีกด้านหนึ่ง แล้วจากนั้นจึงย้อนกลับมายังแบตเตอรี่ ขดลวดที่ยึดติดกับเพลามอเตอร์ทำหน้าที่เป็นแม่เหล็กที่ผลักกับแม่เหล็กถาวรที่ถูกยึดกับตัวเคสมอเตอร์ โดยปกติจะมีขดลวดสองขด เมื่อมีแรงเกิดขึ้นจากขดลวดหนึ่ง แรงผลักของแม่เหล็ก จะพยายามหมุนเพล่าให้ห่างจากแม่เหล็กที่ยึดติดไว้ ถ้าปราศจากคอมมิวเตเตอร์มอเตอร์จะหมุนไปได้เล็กน้อยแล้วหยุด แต่คอมมิวเตเตอร์ทำหน้าที่เหมือนสวิทช์ความเร็วสูง ดังนั้นเมื่อขดลวดหมุนผ่านแม่เหล็ก กำลังจะถูกส่งขดต่อไป ทำให้เกิดการหมุนต่อเนื่อง ที่ความเร็วต่ำการหมุนแต่ละครั้งรู้สึกได้ แต่เมื่อหมุนที่ความเร็วสูงขึ้นหลายพันรอบต่อวินาทีการหมุนจะนิ่มนวลต่อเนื่อง มอเตอร์ของเล่นเด็กอาจมีแค่สองหรือสามขดลวดหมุนจากคอมมิวเตเตอร์ไฟฟ้าไหลผ่านจากแบตเตอรี่เข้ายังแปรงถ่านและกระแสจะไปยังคอมมิวเตเตอร์ผ่านขดลวด เมื่อมอเตอร์หมุนกระแสไฟยังผ่านไปยังขดลวดอีกขดด้วย กระแสไฟฟ้าจะไหลกลับผ่านแปรงถ่านอีกด้าน และกลับเข้าสู่แบตเตอรี่ กระแสที่ไหลผ่านขดลวดจะทำให้เกิดแม่เหล็กไฟฟ้าผลักกับแม่เหล็กถาวรที่ยึดติดกับโครงมอเตอร์ ทำให้ขดลวด คอมมิวเตเตอร์ และเพล่าหมุน



รูปที่ 2.8 ตัวอย่าง Brushed DC Motor

มอเตอร์กระแสตรงแบบไร้แปรงถ่าน (Brushless DC Motor)

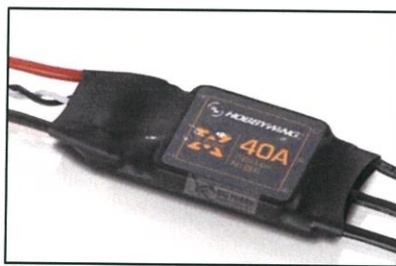
มอเตอร์แบบไร้แปรงถ่าน คล้ายๆกับมอเตอร์ดีซี แต่แม่เหล็กภายในมอเตอร์จะเป็นตัวหมุน ขดลวดภายในมอเตอร์จะได้รับกำลังจากแบตเตอรี่ถูกวางในตัวเคสมอเตอร์หรือภายในแม่เหล็กที่หมุน ด้วยการจัดเรียงแบบนี้ทำให้ข้อดีคือไม่มีแปรงถ่านที่จะถ่ายเทกำลังผ่านส่วนที่หมุนของมอเตอร์ ทำให้มีการบำรุงรักษาต่ำและมอเตอร์จะหมุนอิสระ



รูปที่ 2.9 ตัวอย่าง Brushless DC Motor

2.4.2 ชุดควบคุมความเร็วรอบ

ชุดควบคุมความเร็วรอบแบบอิเล็กทรอนิกส์หรือสปีดคอนโทรลหรือ ESC เป็น อุปกรณ์ควบคุมความเร็วรอบของมอเตอร์ไฟฟ้า ทำให้เราสามารถเร่งหรือเบรคของมอเตอร์ได้ ซึ่งสปีดคอนโทรลจะมีสายไฟออกมาทั้งสองด้าน ด้านหนึ่งจะใช้จ่ายกระแสไฟฟ้าให้แก่มอเตอร์ ส่วนอีกด้านหนึ่งก็จะมีสายไฟออกมาอีก 2 เส้นเพื่อรับกระแสไฟจากแบตเตอรี่และในด้านที่มี สายไฟ นี้ก็จะมีสายไฟเส้นเล็กพร้อมแจ๊คอีก 1 ตัวเพื่อเสียบเข้ากับเครื่องรับสัญญาณ (receiver) สำหรับรับคำสั่งจากเครื่องส่งวิทยุผ่านเครื่องรับสัญญาณอีกที เมื่อเร่งเดินเบาที่เครื่องส่งวิทยุ เครื่องรับวิทยุจะรับคำสั่งจากเครื่องส่งผ่านสปีดคอนโทรลเพื่อจ่ายกระแสมากหรือน้อยให้มอเตอร์เร่งหรือเดินเบาได้ตามความต้องการ



รูปที่ 2.10 ตัวอย่างชุดความเร็วรอบ (ESC)

2.5 แบตเตอรี่ (Battery)

เป็นส่วนเก็บพลังงานสำรองสำหรับการบิน แบตเตอรี่ที่ใช้เป็นชนิดที่มีน้ำหนักเบาและมีความสามารถจ่ายพลังงานสูงได้อย่างต่อเนื่อง แบ่งเป็นสองประเภทใหญ่ๆ คือ

2.5.1 แบตเตอรี่แบบ NiMH (Nickel Metal Hydride)

สำหรับอากาศยานขนาดเล็กแบบ FP นั้น บางรุ่นก็จะใช้แบตเตอรี่แบบ NiMH แบตเตอรี่แบบนี้ใช้งานง่ายและการดูแลรักษาก็ไม่ยุ่งยากนัก ในปัจจุบันแบตเตอรี่แบบนี้เป็นที่นิยมใช้กันมากกว่าแบตเตอรี่แบบ NiCd (Nickel Cadmium) แต่ที่ใช้กันมากเนื่องจากมีความจุมากกว่าถึง 2-3 เท่า นอกจากนี้ NiCd ยังมีสารที่เป็นพิษต่อสิ่งแวดล้อมอีกด้วย ความจุของแบตเตอรี่นิยมบอกในรูปของค่าประจุที่แบตเตอรี่จ่ายได้เช่น 1000 mAh หมายความว่า ใน 1 ชั่วโมงจะสามารถจ่ายกระแสออกมาได้ 1 A



รูปที่ 2.11 ตัวอย่างแบตเตอรี่แบบ NiMH (Nickel Metal Hydride)

2.5.2 แบตเตอรี่แบบ LiPo (Lithium Polymer)

เทคโนโลยีใหม่ที่ทำให้อากาศยานบังคับที่ใช้พลังงานไฟฟ้าได้รับความนิยม มากในปัจจุบันคือ แบตเตอรี่แบบ LiPo (Lithium Polymer) แบตเตอรี่แบบนี้มีข้อได้เปรียบกว่าแบบ NiMH หลายอย่างเช่น น้ำหนักเบาให้พลังงานสูง ความจุสูงและประสิทธิภาพสูงกว่ามาก นอกจากนี้แบตเตอรี่แบบ LiPo ยังมีอัตราการคายประจุสูงมากทำให้มอเตอร์ได้รับพลังงานเพียงพอ กับกำลังที่ต้องการ ส่วนข้อเสียของแบตเตอรี่แบบนี้ก็คือ มีราคาสูงถ้าดูแลรักษาไม่ดี แบตเตอรี่แบบนี้จะชาร์จได้เพียง 300-400

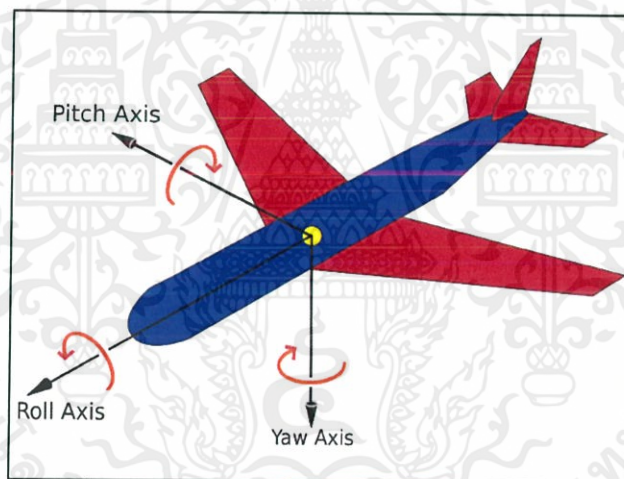
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งเท่านั้น สารประกอบที่อยู่ภายในแบตเตอรี่แบบ LiPo ติดไฟง่ายจึง อาจเกิดการระเบิดได้ถ้าไม่ ชาร์จหรือจัดเก็บอย่างถูกวิธี รวมทั้งถ้าติดตั้งไว้ในตำแหน่งที่ไม่ปลอดภัย จะทำให้เกิดการระเบิดได้เมื่อ เกิดการกระแทก ดังนั้นการใช้แบตเตอรี่แบบนี้จึงต้องมีข้อควรระวังที่ ต้องปฏิบัติตามอย่างเคร่งครัด



รูปที่ 2.12 ตัวอย่างแบตเตอรี่แบบ LiPo (Lithium Polymer)

2.6 การควบคุมการบินแกน ROLL PITCH YAW



รูปที่ 2.13 แกน ROLL PITCH YAW

- แกนตามยาว ตั้งแต่หัวไปจรดหางเรียกว่า แกน longitudinal axis, และการหมุนหรือ เคลื่อนที่ของลำตัวรอบแกนนี้เราเรียกลักษณะเช่นนี้ว่า "Roll "
- แกนขวางจากปีกข้างหนึ่งไปยังปีก อีกข้างหนึ่งเรียกว่า แกน lateral axis, และการหมุนหรือ เคลื่อนที่ลำตัวรอบจุดนี้เราเรียกลักษณะเช่นนี้ว่า "Pitch "
- แกนในแนวตั้งที่ผ่านจุดที่เรียกว่า center of gravity (เมื่อเครื่องบินอยู่ในแนวระดับ) เรา เรียกว่าแกน vertical axis, และการหมุนหรือเคลื่อนที่ของลำตัวรอบจุดนี้เราเรียกลักษณะเช่นนี้ว่า "Yaw "

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

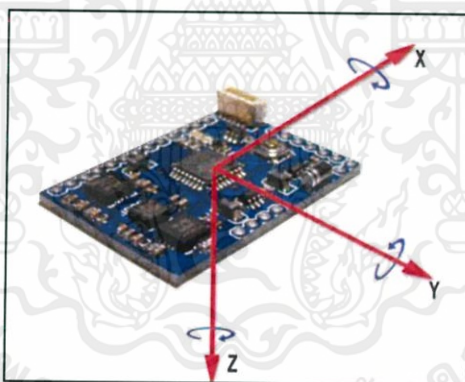
การควบคุมการเคลื่อนที่ของโดรนจะใช้การเอียง การยกหน้า ยกหลัง ตามแกนเหล่านี้มาควบคุมการเดินหน้า ถอยหลังรวมถึงการเลี้ยวด้วย ด้วยหลักการนี้ในเรื่องของมุมเอียง และการหมุนของสามแกนนี้จะได้มาจาก Accelerometer และ Gyroscopes มาคำนวณด้วยกรรมวิธีคณิตศาสตร์เพื่อใช้ควบคุมอากาศยานให้สมดุลร่วมกับ PID เซนเซอร์เหล่านี้จะอยู่ในเซนเซอร์ IMU

2.7 ชุดควบคุมการบิน

จะทำหน้าที่ควบคุมความเสถียรในการบินด้วยไมโครคอนโทรลเลอร์ต่างๆ เซนเซอร์ที่ใช้วัดความเร็ว 3 แกน ควบคุมระดับความสูง และความดันบรรยากาศ ดังนี้

2.7.1 Inertial Measurement Unit (IMU)

เซนเซอร์หรืออุปกรณ์ที่ใช้ใน INS (Inertial Navigation System) เป็นส่วนประกอบหลักที่ใช้ในเครื่องบิน, ยานอวกาศ และ เรือ เช่นเดียวกับจรวดซีปนาวุธ IMU ประกอบด้วย 2 ส่วนหลัก คือ accelerometers 3 ทิศทาง และ gyroscopes 3 ทิศทาง ซึ่งรับความเร่งยานพาหนะและความเร็วเชิงมุมตามลำดับ ทำหน้าที่วัดความเร็ว 3 แกนคือ X,Y,Z เพื่ออ้างอิงกับแรงโน้มถ่วงของโลก กับ วัดความเร็วเชิงมุมที่อาศัยแรงเฉื่อยของ Rotor ที่หมุน เพื่อช่วยในการรักษาทิศทางของแกนหมุนให้อยู่ในตำแหน่งเดิมตลอดเวลาที่มีการหมุน



รูปที่ 2.14 Inertial Measurement Unit (IMU)

- Accelerometer : เป็นอุปกรณ์ที่ใช้วัดความเร็วเฉพาะแกนที่เจาะจงนั้นคือ แกน X , Y , Z โดยเทียบกับแรงโน้มถ่วงของโลกสามารถประยุกต์ใช้สำหรับวัดความเอียงได้
- Gyroscopes : เป็นอุปกรณ์วัดความเร็วเชิงมุมเพื่อสำหรับใช้ในการรักษาระดับการทรงตัวให้เสถียร

ในเมื่อ Accelerometer สามารถวัดความเอียงได้เรียบร้อยแล้ว ไม่มีความจำเป็นจะต้องวัดความเร็วเชิงมุม เพราะว่า Output ของ Accelerometer มีผลกับแรงโน้มถ่วง นั่นหมายถึงค่า Output ของ Accelerometer ไม่มีทางหยุดนิ่งเฉย แม้ปล่อยทิ้งไว้ มันก็จะวิ่งขึ้นๆ ลงๆ สั่นไปสั่นมา ต่างกับ Gyroscope ที่ปล่อยทิ้งไว้ค่า Output ที่ได้ก็จะนิ่งไม่เกิดการสั่นและถ้าเรานำ Output ของเอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gyroscope มาใช้กรรมวิธีการ Discrete Integral ก็จะสามารถหามุมได้เนื่องจาก Output ของ Gyroscope เป็นความเร็วเชิงมุม)

2.7.2 เซ็นเซอร์วัดระยะทาง Ultrasonics

เป็นเซ็นเซอร์ที่ทำงานโดยอาศัยคลื่นเสียงที่มีความถี่สูงกว่า 20 kHz ซึ่งเป็นคลื่นในย่านที่มนุษย์ไม่สามารถได้ยินเสียง เซ็นเซอร์ชนิดอัลตราโซนิกทำงานโดยอาศัยการกระจายหรือการเคลื่อนที่ของคลื่นเสียงไปกระทบกับพื้นผิวของตัวกลาง ซึ่งอาจเป็นของแข็งหรือของเหลวบางส่วนของคลื่นเสียงจะแทรกผ่านเข้าไปในตัวกลางนั้นและส่วนใหญ่ของคลื่นความถี่สูงนี้จะสะท้อนกลับเรียกว่า "Echo" โดยช่วงเวลาของการสะท้อนกลับของคลื่นเสียงเป็นสัดส่วนโดยตรงกับระยะห่างระหว่างวัตถุกับเซ็นเซอร์ ทำให้สามารถวัดระยะความสูงจากตัวโดรนถึงพื้นก่อนลงจอด



รูปที่ 2.15 ตัวอย่างเซ็นเซอร์ HC-SR04

2.7.3 เซ็นเซอร์วัดความดันบรรยากาศ Barometer

บารอมิเตอร์ (Barometer) คือ เครื่องมือตรวจวัดความดันบรรยากาศ สำหรับวัดค่าความกดดันที่เกิดจากแรงดันของอากาศ โดยใช้ของเหลวหรือวัสดุแข็งที่สัมผัสโดยตรงกับอากาศ แนวโน้มการเปลี่ยนแปลงความกดดัน สามารถนำไปทำนายการเปลี่ยนแปลงสภาพอากาศในช่วงเวลาสั้น ๆ การวัดความกดดันอากาศหลายจุด นำมาประมวลผลภายในการวิเคราะห์อากาศพื้นผิว (Surface Weather Analysis) เพื่อช่วยค้นหาร่องความกดอากาศ (Surface Troughs), ระบบความกดอากาศสูง (High Pressure Systems) และเส้นความกดอากาศเท่า (Frontal Boundaries)

2.8 ใบพัด (Propeller)

ใบพัด (Propeller) คืออุปกรณ์หมุนชนิดขับเคลื่อนด้วยกำลัง มีดุมหนึ่งดุม และแขนหรือกลีบ รูปร่างปิดอย่างเกลียวตั้งแต่สองอันขึ้นไป อุปกรณ์นี้ติดตั้งอยู่บนอากาศยาน หรือเรือบางแบบ และ ใช้สำหรับขับเคลื่อน หรือดึงอากาศยานไปข้างหน้าผ่านอากาศโดยการกระทำของใบพัดหมุนกินอากาศหรือของไหลอื่น

ข้อมูลทั่วไปของใบพัด

Thrust เป็นแรงที่ใช้ขับเคลื่อนอากาศยานไปในอากาศ Thrust เป็นระบบที่สร้างขึ้นเพื่อใช้ผลักดันอากาศยาน ซึ่งมีอยู่หลายรูปแบบและหลายวิธีในการสร้างระบบผลักดันนี้ขึ้นมา ใบพัดเป็นระบบขับเคลื่อนอากาศยานรูปแบบหนึ่ง ใบพัดประกอบด้วยด้วยใบหรือกลีบตั้งแต่สองกลีบขึ้นไป ต่อกันด้วยศูนย์กลางซึ่งเรียกว่า Hub ทำหน้าที่ยึดใบพัดแต่ละกลีบเข้ากับ Shaft ของเครื่องยนต์ชนิดของใบพัด

2.8.1 ใบพัดไม้

เป็นใบพัดที่แข็งแรงและมีน้ำหนักเบา น้ำหนักที่เบาของใบพัดจะช่วยให้ได้รอบมอเตอร์ที่สูงขึ้น



รูปที่ 2.16 ตัวอย่างใบพัดไม้

2.8.2 ใบพัดคาร์บอนเสริมไนลอน หรือ ใบพัด APC

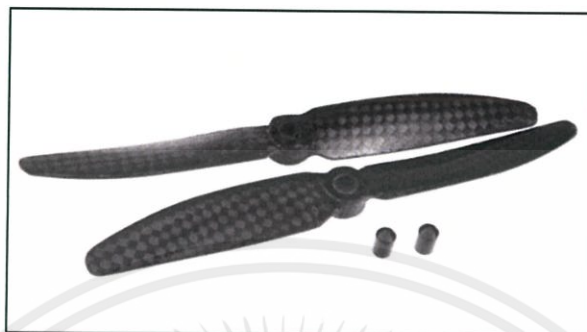
เป็นใบพัดที่นิยมใช้ทั่วไป มีน้ำหนักมากกว่าใบพัดชนิดไม้แต่มีความยืดหยุ่นสูงกว่า สำหรับการลงจอดของอากาศยานที่ใช้ใบพัดชนิดนี้อาจทำได้ไม่ดี แต่ยังดีกว่าใบพัดชนิดไม้



รูปที่ 2.17 ตัวอย่างใบพัดคาร์บอนเสริมไนลอน 4

2.8.3 ใบพัดคาร์บอนไฟเบอร์

เป็นใบพัดที่นิยมใช้ในการแข่งขัน ใบพัดมีความยืดหยุ่นสูงกว่าใบพัดชนิดอื่น และมีพื้นที่ใบขนาดใหญ่



รูปที่ 2.18 ตัวอย่างใบพัดคาร์บอนเสริมไฟเบอร์

ใบพัดที่นิยมใช้กับอากาศยานหลายใบพัด แบ่งออกเป็น 2 ชนิดคือ

- ใบพัดแบบรอบจัด (Speed Propeller) ใบพัดชนิดนี้เหมาะกับอากาศยานประเภทบินเร็ว และมีน้ำหนักเบา เน้นรอบจัดเพื่อความคล่องตัวในการบิน



รูปที่ 2.19 ตัวอย่างใบพัดแบบรอบจัด (Speed Propeller)

- ใบพัดแบบรอบต่ำ (Slow Propeller) ใบพัดชนิดนี้ใช้กับอากาศยานที่มีขนาดใหญ่ มีน้ำหนักตัวลำที่มาก โดยใบพัดชนิดนี้จะเน้นสำหรับแรงดึง ให้รอบต่ำและทอร์คสูง



รูปที่ 2.20 ตัวอย่างใบพัดแบบรอบต่ำ (Slow Propeller)

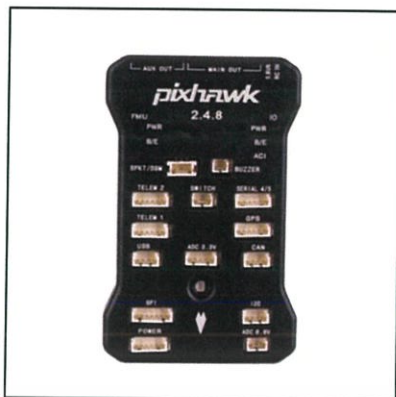
ส่วนลักษณะของใบพัดที่แตกต่างกันนั้น เกิดจากการออกแบบให้มีลักษณะพิเศษ เช่น เพื่อลดเสียงการรบกวน หรือ เพื่อเน้นรอบของการบิน เป็นต้น

2.9 บอร์ดควบคุมการบิน

ทางผู้จัดทำได้ใช้บอร์ดควบคุมการบินซึ่งเป็นชุดพัฒนาการบินในรูปแบบเปิดหรือ Opensource มีให้เลือกใช้ตามค่ายผู้ผลิตต่างๆ ชุดบอร์ดเหล่านี้จะมาพร้อมกับการรองรับเซ็นเซอร์ควบคุมการบิน ซึ่งได้แก่ Gyroscope , Barometer , Accelrometer , Magnetometer และยังมีรองรับ GPS module อีกด้วย สามารถนำไปประยุกต์ใช้ได้ทันที ปัจจุบันสามารถเลือกซื้อได้จากค่ายผู้ผลิตและตัวแทนจำหน่าย

ตารางที่ 2.1 แสดงผู้พัฒนาโปรแกรมและบอร์ดควบคุมการบิน

ผู้พัฒนา	เว็บไซต์
Arducopter	http://code.google.com/p/arducopter
Openpilot	http://www.openpilot.org
Paparazzi	http://paparazzi.enac.fr
Pixhawk	http://pixhawk.ethz.ch
Mikrokopter	http://www.mikrokopter.de
KKmulticopter	http://www.kkmulticopter.com
Multiwii	http://www.multiwii.com
Aeroquad	http://www.aeroquad.com



รูปที่ 2.21 บอร์ดคอนโทรลเลอร์ Pixhawk 2.4.8

2.10 ชุดควบคุมสัญญาณไร้สาย

การรับ-ส่งสัญญาณวิทยุนั้นประกอบไปด้วยส่วนสำคัญ 3 ส่วนด้วยกันคือ เครื่องส่งวิทยุ (Transmitter) ตัวคลื่นวิทยุ (Radio wave) และ เครื่องรับวิทยุ (Receiver)

2.10.1 โมดูลรับ-ส่งสัญญาณวิทยุ

ตัวส่งสัญญาณวิทยุ ใช้งานในคลื่นความถี่ 433 MHz ได้ระยะทาง ประมาณ 500-800 เมตร การส่งสัญญาณแปลงจาก USB เป็น RS232 ใช้โปรโตคอล MAVlink เป็นโปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ภาคพื้นดิน โดยมีการสื่อสารกับภาคพื้นดินผ่านโมดูลไร้สาย



รูปที่ 2.22 Telemetry 433 MHz

2.10.2 รีโมทวิทยุบังคับ

เป็นอุปกรณ์รีโมทใช้บังคับโดรน โดยจะมีตัวรับสัญญาณรีโมทติดที่ตัวโดรน และรีโมทจะส่งคำสั่งไปด้วยโมดูลตัวส่งเพื่อควบคุมตัวโดรน โดยใช้โมดูล Telemetry 433 MHz เป็นตัวส่งได้อีกทั้งสามารถแยกชุดคำสั่งการส่งสัญญาณของแต่ละแกนแนลโดยมีทั้งหมด 6 แกนแนล



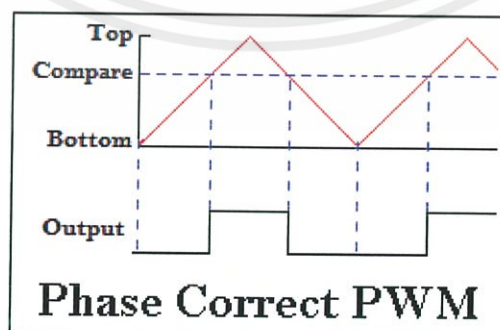
รูปที่ 2.23 รีโมทบังคับวิทยุ Futaba

2.11 สัญญาณ PWM และ PPM

2.11.1 สัญญาณ PWM

PWM ย่อมาจาก Pulse Width Modulation คือการนำสัญญาณสามเหลี่ยมและสัญญาณที่ต้องการปรับความกว้างของพัลส์มาเปรียบเทียบกับ ตัวอย่างเช่น ถ้านำสัญญาณสามเหลี่ยมมาเปรียบเทียบกับสัญญาณไฟฟ้ากระแสตรงจะได้ความกว้างของพัลส์คงที่และความกว้างของพัลส์นั้นจะขึ้นอยู่กับแอมพลิจูดของสัญญาณไฟตรง

ดังรูป



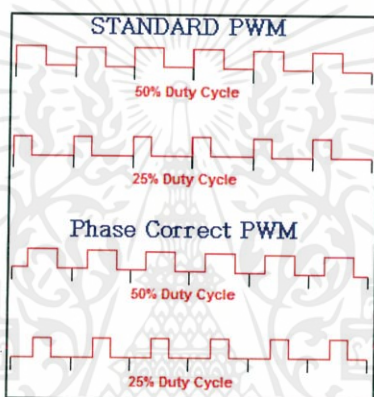
รูปที่ 2.24 การเปรียบเทียบสัญญาณเพื่อให้ได้ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ประโยชน์จาก PWM

ในระบบ digital นั้นจะมีสัญญาณแค่ High และ Low ถ้าต้องการจะนำระบบ digital ไปควบคุมอุปกรณ์ที่ต้องควบคุมด้วยสัญญาณ analog เช่น การควบคุมความเร็วมอเตอร์, การควบคุมความสว่างของหลอดไฟ ฯลฯ ในระบบ digital จะไม่สามารถควบคุมอุปกรณ์พวกนี้ได้ตรงๆ เราจะต้องใช้ PWM ในการควบคุม โดยจะปรับคาบเวลาของ High-Low (Duty cycle) ดังตัวอย่างต่อไปนี้

- 50% Duty cycle หมายความว่า คาบเวลาที่เป็น High 50% และ คาบเวลาที่เป็น Low 50%
- 25% Duty cycle หมายความว่า คาบเวลาที่เป็น High 25% และ คาบเวลาที่เป็น Low 75%
- 80% Duty cycle หมายความว่า คาบเวลาที่เป็น High 80% และ คาบเวลาที่เป็น Low 20%

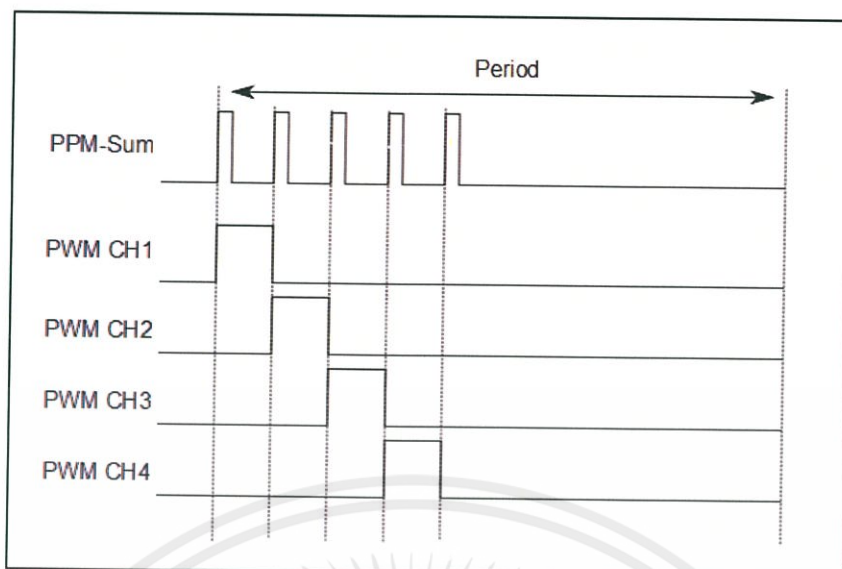


รูปที่ 2.25 สัญญาณ PWM ที่ Duty cycle ต่างๆ

ในการปรับคาบเวลาจะมีผลกับกระแสไฟฟ้าที่จ่ายออกไปยังอุปกรณ์ที่จะควบคุมดังนี้ ถ้าปรับ 50% Duty cycle จะทำให้มีช่วงที่เป็น High 50% ดังนั้นกระแสไฟฟ้าที่จ่ายออกไปจะเหลือแค่ 50% ของทั้งหมด เมื่อนำไปควบคุมมอเตอร์ จะทำให้มอเตอร์หมุนด้วยความเร็ว 50% ของความเร็วสูงสุด

2.11.2 สัญญาณ PPM สัญญาณ PPM

เป็นเทคนิคหนึ่งในการใช้สัญญาณเพียงหนึ่งช่อง แต่สามารถนำไป ใช้สร้างสัญญาณ PWM สำหรับ R/C (Radio Control) Servo ได้มากกว่าหนึ่งช่องสัญญาณ รูปภาพต่อไปนี้แสดงตัวอย่างของ รูปคลื่นสัญญาณ PPM-Sum สำหรับสัญญาณ PWM จำนวน 4 ช่อง



รูปที่ 2.26 คลื่น PPM-Sum

รูปที่ 2.27 แสดงคลื่นสัญญาณ PPM-Sum แทนการสร้างสัญญาณ PWM จำนวน 4 สัญญาณ เห็นได้ว่าสัญญาณ PPM-Sum เป็นสัญญาณแบบมีคาบ (periodic) แต่ละคาบถูกแบ่งออกได้เป็น Pulse ที่มีความกว้างช่วง HIGH คงที่ (เช่น ในช่วง 250..400 usec) แต่ความกว้างช่วง LOW ของ Pulse แต่ละอัน อาจไม่เท่ากัน ความกว้างของ Pulse แต่ละอัน (ความกว้างในช่วง HIGH และ LOW โดยรวม) จะเป็นตัวกำหนดค่า Duty Cycle สำหรับสัญญาณ PWM โดยเรียงตามลำดับของ ช่องสัญญาณ โดยทั่วไปจำนวนของสัญญาณ PWM มักอยู่ในช่วง 6-8 สัญญาณ ในบางกรณี จะใช้ สัญญาณ Inverted PPM-Sum คือ เป็นสัญญาณกลับลอจิก (ลอจิกตรงข้าม) ของสัญญาณ PPM-Sum

2.12 NRF24L01

NRF24L01 คือ Module สำหรับรับและส่งสัญญาณวิทยุ แบบ Transceiver คือสามารถทำหน้าที่ทั้งเป็นตัวรับ(Receiver) และส่ง(Transmitter) ได้ในตัวเอง โดยใช้ความถี่ในการรับ-ส่งข้อมูล 2.4 GHz และที่สำคัญสามารถใช้งานได้ง่ายเหมาะกับ Arduino และมีราคาถูกด้วย จึงเป็นหนึ่งในโมดูลสื่อสารไร้สายที่มีความนิยมสูงมาก

2.12.1 GENERAL FEATURES

- โมดูลสื่อสารไร้สาย Wireless ด้วยคลื่นวิทยุ 2.4 GHz
- แบบ Transceiver (Transmitter + Receiver) ส่งและรับสัญญาณในตัวเดียว
- ไฟเลี้ยง : 1.9 - 3.6 Volts
- ระยะรับส่งสูงสุด : 1100 m

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความเร็วในการรับส่ง : 250 Kbps , 1 Mbps , 2 Mbps
- พร้อมสายอากาศขนาดเล็กในตัว
- ขนาด : 34 x 17 x 1 mm (กว้าง x ยาว x หนา)
- ใช้ความถี่สูงถึง 2.4 GHz ทำให้รับ-ส่งข้อมูลได้รวดเร็วและที่ละมากๆ
- ความเร็วในการรับ-ส่ง สูงสุดถึง 2 Mb ต่อวินาที
- กินกระแสต่ำ ไม่เกิน 14 mA

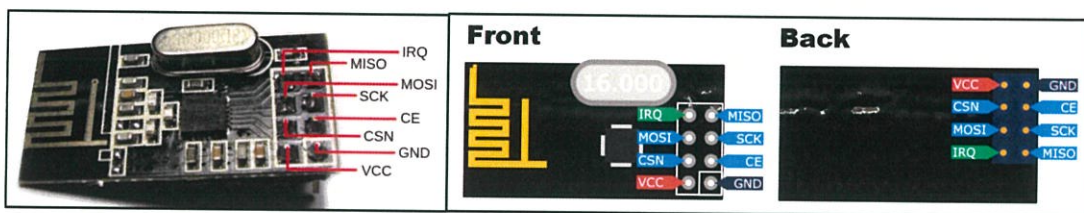


รูปที่ 2.27 NRF24L01

ตารางที่ 2.2 ขาและหน้าที่ของแต่ละขา ของ NRF24L01

ตำแหน่ง	ชื่อ	หน้าที่
PIN 1	GND	Ground
PIN 2	VCC	ไฟเลี้ยง 1.9 - 3.6 V (ใช้ 3.3 V จาก Arduino)
PIN 3	CE	Chip Enable , Rx หรือ Tx (รับหรือส่ง)
PIN 4	CSN	Chip Select Not
PIN 5	SCK	Clock สัญญาณนาฬิกา
PIN 6	MOSI	Slave Data Input (Master Output - Slave Input)
PIN 7	MISO	Slave Data Outout (Master Input - Slave Output)
PIN 8	IRQ	Interupt Requeset pin (Active Low)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 ขาของ NRF24L01

ตารางที่ 2.3 การต่อ NRF24L01 เข้ากับ Arduino

โดยจะใช้งาน 7 Pin จากทั้งหมด 8 (ปล่อย Pin IRQ เอาไว้) โดยต่อ Pin ทั้ง 7 เข้ากับ Arduino

NRF24L01	Arduino
GND	GND
VCC	3.3 V
CE	Digital 9
CSN	Digital 10
MOSI	Digital 11
MISO	Digital 12
SCK	Digital 13

2.12.2 Mesh Network

รูปแบบการจัดโครงข่ายลักษณะ Tree Topology มองอย่างผิวเผินสำหรับโมดูล nRF24L01 หลายคนอาจเห็นว่าเป็นเพียงโมดูลสื่อสารไร้สายใช้เพื่อส่งข้อมูลจากจุดหนึ่งไปอีกจุดหนึ่งเท่านั้น จริงๆ แล้วตัวโมดูลเองมีความสามารถในการเชื่อมโยงกันเป็นกลุ่มได้ โดยสูงสุดอยู่ที่ 6 โมดูลในแต่ละกลุ่ม โดยแต่ละโมดูลจะมีหมายเลขประจำตัวเป็น 00 ถึง 05 ดูเหมือนจำนวนโมดูลจะเป็นข้อจำกัดของการใช้งาน แต่จริงแล้วได้มีการเขียนไลบรารีเพื่อขยายจำนวนโมดูลให้เพิ่มขึ้นได้ เป็นลักษณะรูปแบบกิ่งของต้นไม้ (Tree Topology) การแตกกิ่งจะเริ่มจากโมดูลแรก (โมดูล 00) จะมีกิ่งที่เชื่อมโยงได้คือ 01, เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

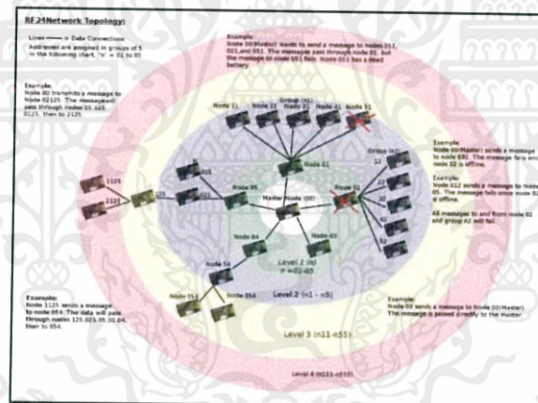
02, 03, 04 และ 05 จากนั้นในแต่ละกิ่งก็จะสามารถแตกกิ่งออกไปได้อีก 5 กิ่ง ตัวอย่างกิ่งที่ 04 ก็จะมีกิ่งที่แตกออกได้เป็น 014, 024, 034, 044, 054 และกิ่งที่ 05 ก็จะได้กิ่ง 015, 025, 035, 045, 055

สำหรับการส่งข้อมูล กิ่งย่อยจะสามารถรับส่งข้อมูลได้เฉพาะกับกิ่งหลัก (parent) และกิ่งย่อยอื่นที่แตกออกจากตัวเอง (child) จะรับส่งข้อมูลข้ามกิ่งหลักไม่ได้ การจะรับส่งข้อมูลไปยังกิ่งย่อยอื่นจะอาศัยการฝากข้อมูลส่งไปที่กิ่งหลัก กิ่งหลักจะส่งไปที่กิ่งหลักถัดไปและกิ่งย่อยอื่น กิ่งแกนจะส่งข้อมูลผ่านไปยังกิ่งหลักของกิ่งย่อยที่เราต้องการส่งข้อมูล สมมติว่ากิ่งย่อย 014 ต้องการส่งข้อมูลให้กิ่งย่อย 035 การส่งข้อมูลจะเป็นดังนี้

014 -> 04 -> 00 -> 05 -> 035

โดยที่ข้อมูลที่ส่งไปจะมีส่วนหัว (Header) ที่บอกว่ากิ่งไหนเป็นตัวส่ง และต้องการส่งให้กิ่งไหน ทำให้เมื่อกิ่งย่อยได้รับข้อมูลแล้วต้องการส่งข้อมูลกลับก็จะทราบว่าจะส่งกลับไปให้กิ่งไหน เส้นทางจะเป็นดังนี้

035 -> 05 -> 00 -> 04 -> 014



รูปที่ 2.29 ลักษณะการขยายกิ่งโมดูล

จำนวนกิ่งทั้งหมดที่สามารถสร้างได้จากโครงข่าย 6 ชั้น $1 + 5 + 25 + 125 + 625 + 3,125 = 3,906$ กิ่ง ซึ่งเป็นจำนวนมากที่เดียวในทางทฤษฎี แต่ในทางปฏิบัติแล้วข้อจำกัดอาจเกิดจากสิ่งอื่นๆ ทำให้จำนวนที่ใช้งานจริงอาจน้อยกว่า

การนำไปใช้งาน Mesh network มีประโยชน์ในการขยายโครงข่ายที่ต้องการลูกข่ายจำนวนมาก โดยที่ตัวลูกข่ายเองสามารถทำตัวเป็นแม่ข่ายได้ในขณะเดียวกัน ดูได้จากการวางโครงข่ายข้างต้น ข้อมูลจะวิ่งไปกลับผ่านแต่ละกิ่งไปยังฐาน (Node 00) และยังสามารถส่งข้อมูลไปยังลูกข่ายตัวอื่นๆได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำไปใช้งานโดยส่วนใหญ่ใช้เพื่ออ่านข้อมูลของเซนเซอร์ เพื่อนำค่ามาแสดงหรือประมวลผล ที่ส่วนกลาง โดยผ่านคำสั่งไปที่โหนดฐาน จากนั้นคำสั่งจะถูกกระจายไปจนถึงปลายทาง และลูกข่าย ปลายทางจึงทำการส่งข้อมูลกลับไปยังโหนดฐาน ไม่เพียงแต่จะตอบสนองต่อคำสั่ง แต่ละโหนดก็ยัง สามารถส่งข้อมูลไปที่โหนดฐาน

ตามคาบเวลาที่กำหนด ลักษณะการส่งข้อมูลตามคาบเวลา อาจก่อให้เกิดการชนกันของข้อมูลได้กับ โครงข่ายที่มีสมาชิกมากๆ ซึ่งอาจแก้ไขด้วยการจัดคาบเวลาในการส่งข้อมูลแบบซุ่ม

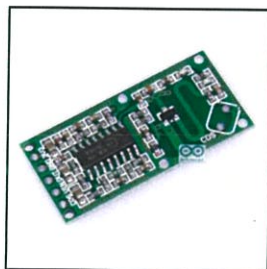
ข้อดีของการสร้างโครงข่ายลักษณะนี้คือ การที่แต่ละโหนดไม่จำเป็นต้องมีกำลังส่งสูง การ เชื่อมโยงจะอาศัยการส่งกันเป็นทอดๆไป

2.13 RCWL-0516 เซนเซอร์ตรวจจับวัตถุ แบบคลื่นไมโครเวฟ

โมดูลเซนเซอร์เรดาร์ doppler ขนาดเล็กนี้มาพร้อมกับแรงดันไฟฟ้ากว้าง DC 4-28V ซึ่งเป็น โมดูลเซนเซอร์เรดาร์ตรวจจับการเคลื่อนไหวขนาดเล็กพร้อมกับชิป RCWL-9196 ตามเทคโนโลยีการ เหนี่ยวนำไมโครเวฟของ doppler มันจะส่งสัญญาณระดับสูง TTL โดยอัตโนมัติเมื่อมีการเคลื่อนไหว 360 องศาไม่มีการตรวจจับมุมตาบอดและระยะเซนเซอร์สูงสุด 7 เมตร พร้อมเวลาหน่วงและความไว ที่ปรับได้ มันเหมาะสำหรับสวิตช์ไฟไมโครเวฟเซนเซอร์การเคลื่อนไหว

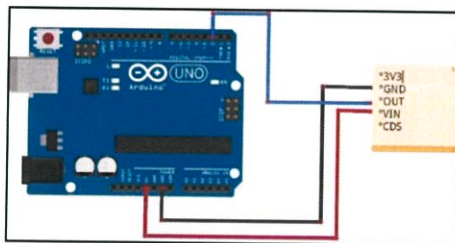
การใช้งานจะเหมือนกับเซนเซอร์ตรวจจับความเคลื่อนไหวแบบ PIR ซึ่งใช้หลักการตรวจจับ ความร้อนแบบอินฟราเรด ให้เอาต์พุตที่ตรวจจับได้เป็นแบบ digital แต่ข้อดีของ PIR ที่ใช้การ ตรวจจับความร้อนคือ ช่วงตรวจจับที่แคบ ถ้าใช้งานภายนอกอาจถูกรบกวนจากอุณหภูมิได้ง่าย ทำให้ ตรวจจับผิดพลาดบ่อย

ถ้างานตรวจจับการเคลื่อนไหว เป็นลักษณะ ตรวจจับในวงกว้าง มีรัศมีไกล ใช้งานภายนอก เช่น ทำไฟอัตโนมัติหลังบ้าน ไฟโรงรถ ตรวจจับผู้บุกรุก เซนเซอร์ Microwave Radar RCWL-0516 ตัวนี้เหมาะสมที่สุด ใช้งานง่าย เมื่อตรวจโมดูลมีจับการเคลื่อนไหวจะส่งสัญญาณเอาต์พุตเป็นแบบ digital ค่า 0 หรือ 1 ออกมา



รูปที่ 2.30 RCWL-0516 เซนเซอร์ตรวจจับวัตถุ แบบคลื่นไมโครเวฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 การต่อ RCWL-0516 กับ Arduino

2.14 LCD Display

คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือ ด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึก ก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้จะไม่สว่าง ผลึกมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่นสีเขียว หรือ สีฟ้า ฯลฯ ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือ แล้วพบกับพื้นหลังสีต่างๆกัน

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่นนาฬิกาดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น

จอ LCD แบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

1. Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่นจอ LCD ขนาด 16x2 หมายถึงใน 1 แถวมีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถวมี ตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด

2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอขึ้นแสง หรือปล่อยแสงออกไป ทำ ให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด



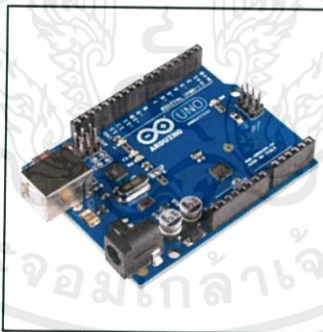
รูปที่ 2.32 LCD module

2.16 Arduino UNO

Arduino คือ Open-Source Platform สำหรับการสร้างต้นแบบทางอิเล็กทรอนิกส์ โดยมีจุดมุ่งหมายให้ Arduino Platform เป็น Platform ที่ง่ายต่อการใช้งาน, โดย Arduino Platform ประกอบไปด้วย

2.16.1 ส่วนที่เป็นฮาร์ดแวร์ (Hardware)

บอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ เพื่อให้ง่ายต่อการใช้งาน หรือที่เรียกกันว่า บอร์ด Arduino, โดยบอร์ด Arduino เองก็มีหลายรุ่นให้เลือกใช้ โดยในแต่ละรุ่นอาจมีความแตกต่างกันในเรื่องของขนาดของบอร์ด หรือสเปค เช่น จำนวนของขาที่ส่งสัญญาณ, แรงดันไฟที่ใช้, ประสิทธิภาพของ MCU เป็นต้น



รูปที่ 2.33 Arduino UNO

2.16.2 ส่วนที่เป็นซอฟต์แวร์ (Software)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินการวิจัย

3.1 การประกอบโดรนและการคำนึงถึงน้ำหนักโหลด

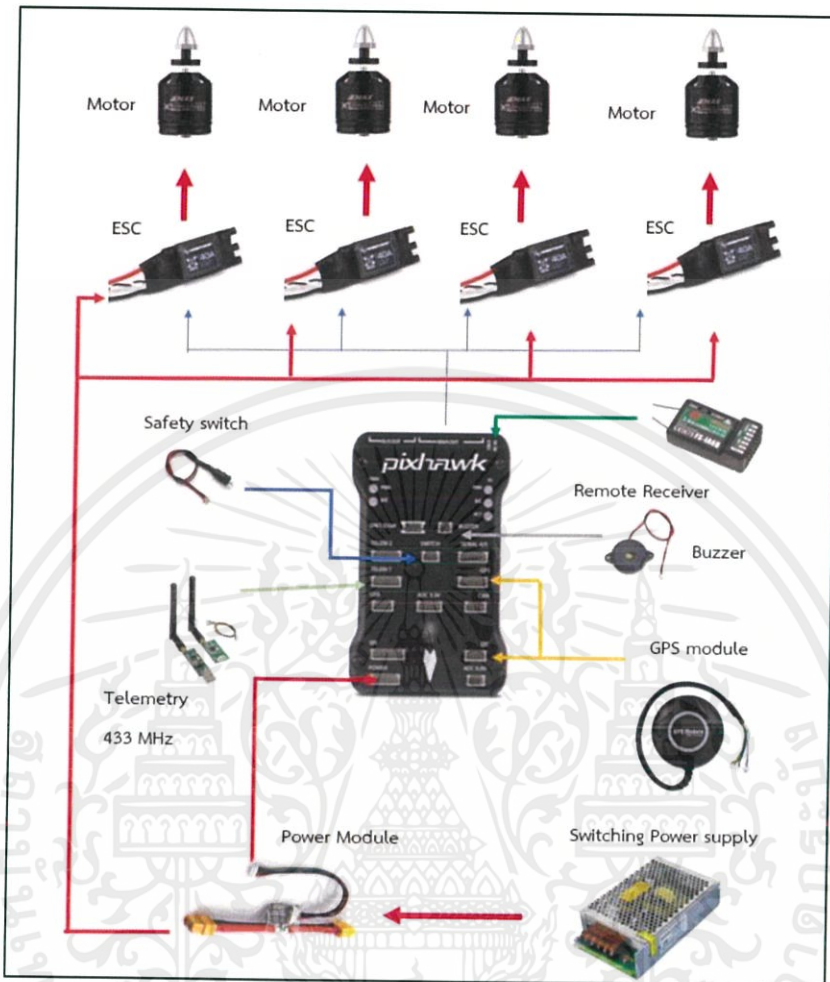
ในการประกอบจะต้องทำการเลือกอุปกรณ์และตั้งค่าซอฟต์แวร์ Flight controller ด้วยโปรแกรม Mission planner การตั้งค่าอุปกรณ์ การตั้งค่า PID การประเมินน้ำหนักโหลด และกระแสไฟฟ้าสูงสุดที่มอเตอร์ดึงกระแสไปใช้งาน

3.1.1 การประกอบโดรน

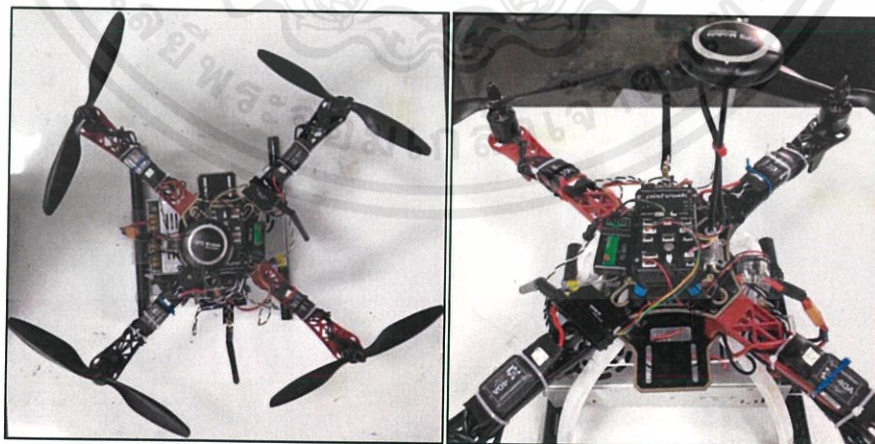
ส่วนประกอบของโดรนที่ใช้

1. บอร์ด Flight Controller Pixhawk 2.4.6
2. มอเตอร์ 4 ตัว รุ่น MT2216 – 810KV
3. ชุดควบคุมความเร็วมอเตอร์ (Electronics Speed controller) Hobbywing 4 ตัว
4. ไบพัตคาร์บอนไฟเบอร์ 4 ใบ รุ่น EMAX1045
5. เฟรมโดรน F450
6. GPS M8N module
7. Telemetry 433 MHz
8. Buzzer
9. Safety switch
10. Remote Receiver FS – i6A
11. Switching power supply 12V DC 50A
12. รีโมทวิทยุรุ่น FS-I6A พร้อมด้วยตัวรับวิทยุ

การต่ออุปกรณ์



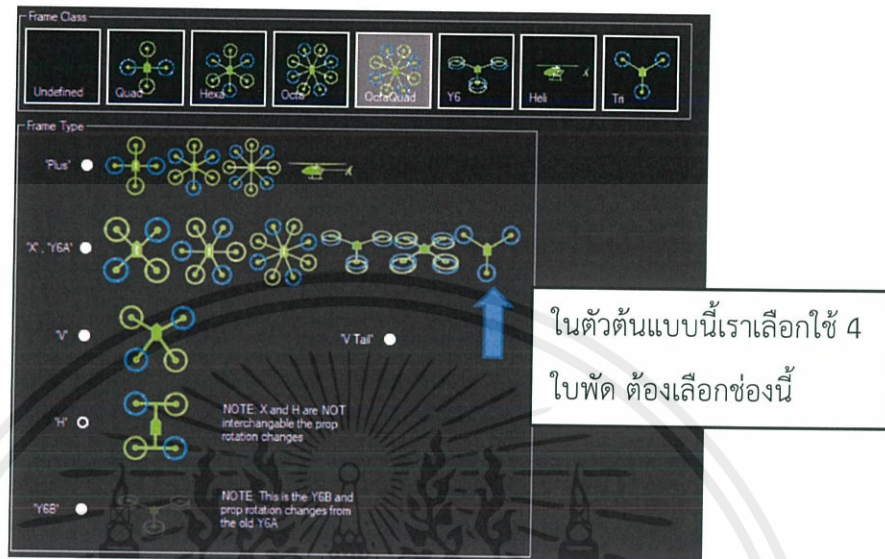
รูปที่ 3.1 บล็อกไดอะแกรมการต่ออุปกรณ์



รูปที่ 3.2 โดรนเมื่อประกอบจนเสร็จสิ้นแล้ว

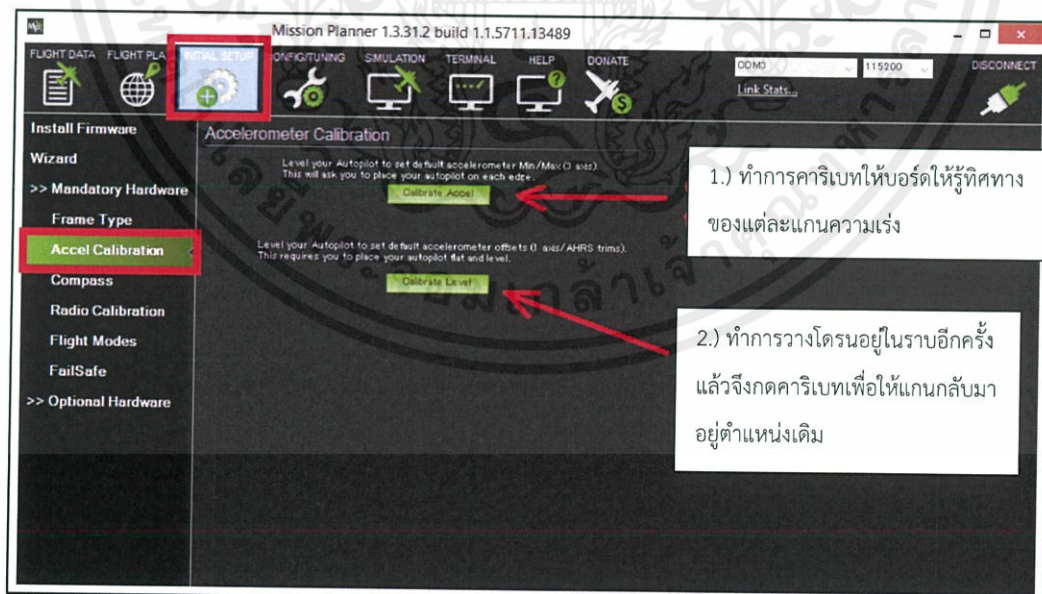
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การตั้งค่าบอร์ด Flight Controller ด้วยโปรแกรม Mission Planner ตั้งค่ารูปแบบของโดรน



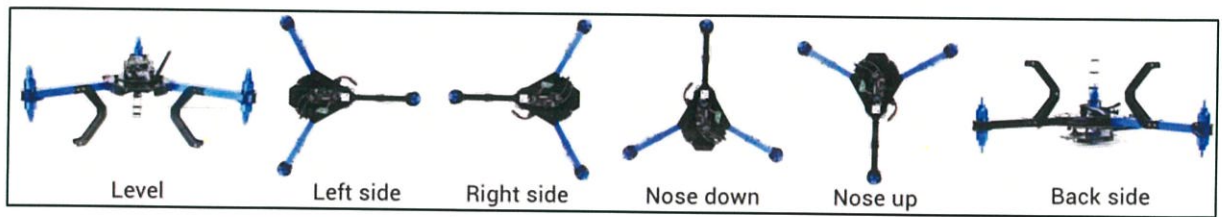
รูปที่ 3.3 การตั้งค่าเฟรมของโดรน

ทำการคาร์ิบเรท Accelerometer



รูปที่ 3.4 ขั้นตอนการคาร์ิบเรท Accelerometer

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

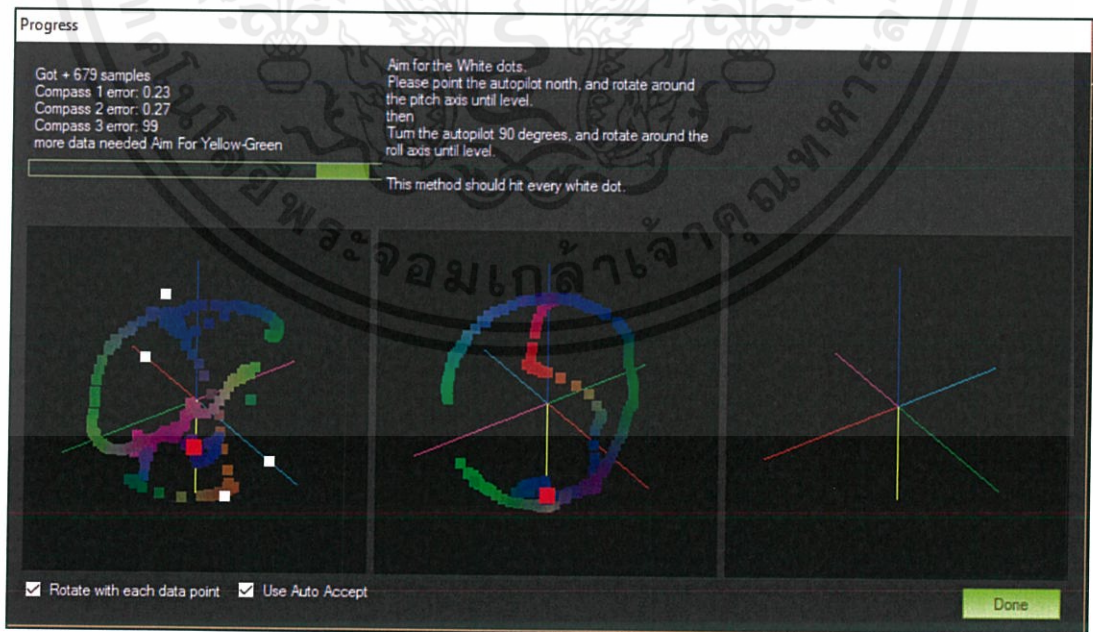


รูปที่ 3.5 การจัดโดรนเพื่อการคาริเบรททิศทาง

- Level : วางในแนวราบ
- Left side : วางโดรนตะแคงซ้าย
- Right side : วางโดรนตะแคงขวา
- Nose down : วางโดรนให้ทิศด้านหน้าของโดรนก้มหัวลง
- Nose up : วางโดรนให้ทิศด้านหน้าของโดรนชี้ขึ้นด้านบน
- Back side : วางหางโดรนเพื่อให้ทิศด้านหน้าของโดรนไปอยู่ด้านหลัง

ทำการคาริเบรท Compass

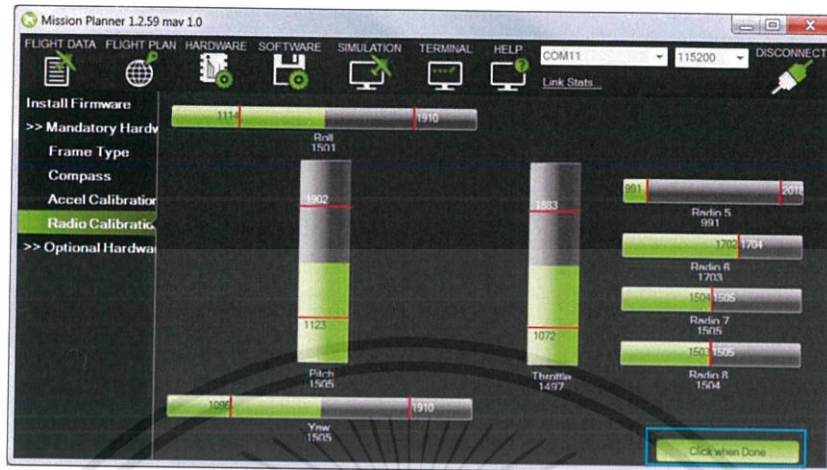
เข้าเมนู Compass แล้วทำการคาริเบรททิศทางของเข็มทิศให้รู้จักมุมการหมุน โดยการขยับตัวโดรนไปรอบทิศทางให้สัมผัสกับจุดแดงที่แสดงบนโปรแกรม โปรแกรมจะเก็บค่ามุมทิศทางไปเรื่อยๆจนการคาริเบรทเสร็จสิ้น



รูปที่ 3.6 หน้าต่างโปรแกรมขณะทำการคาริเบรท Compass

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Radio calibration



รูปที่ 3.7 หน้าต่างขณะ Radio Calibration

ทำการปรับรีโมทวิทยุบังคับให้คันเร้งนั้นมีการปรับค่าสูงสุดต่ำสุดตามที่ต้องการ โดยเมื่อหมุนคันเร้งหรือสวิตซ์ต่างๆแสดงการเปลี่ยนแปลงในโปรแกรมให้เราทำการปรับค่าต่ำสุดไปสูงสุดแล้วทำการกด Click when Done เป็นอันเสร็จสิ้น

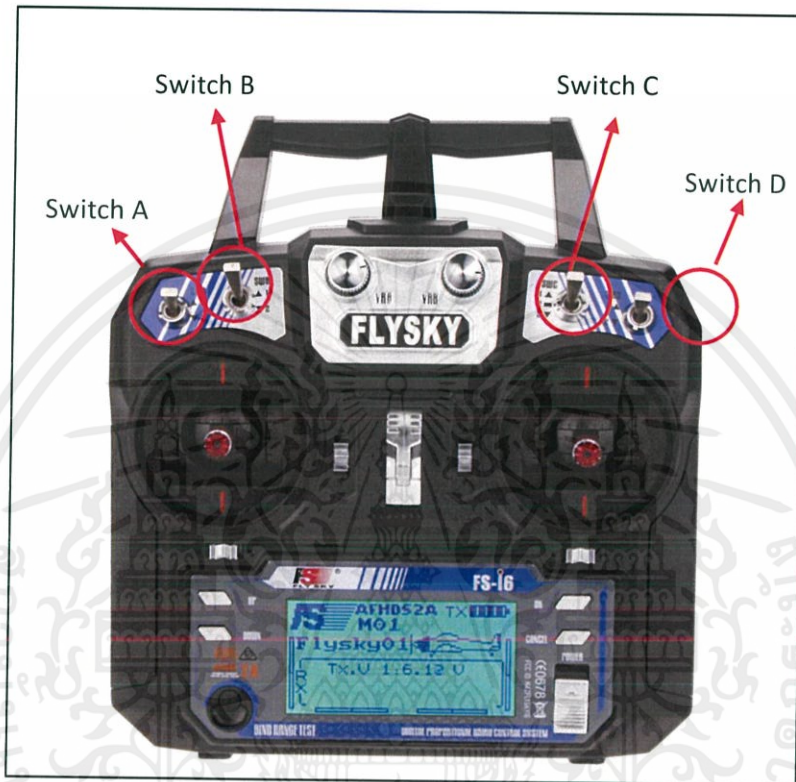


รูปที่ 3.8 รีโมทวิทยุ Flysky รุ่น FS - i6A

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flight Mode

ทำการเลือกโหมดการบินแต่ละโหมดโดยสวิตช์ที่ตัวรีโมทจะเป็นตัวเลือกโหมดในการบิน โดย สวิตช์ B และ C จะทำงานร่วมกัน โดยเมื่อ Switch B อยู่ในตำแหน่งที่ 1 จะได้ 3 flight mode ใน สวิตช์ C และเมื่ออยู่ในตำแหน่งที่ 2 จะได้อีก 3 flight mode เช่นกัน



รูปที่ 3.9 การตั้งค่าโหมด

- Switch A : ตั้งค่าโหมด Auto ทำงานโดยเราจะกำหนดจุด Waypoint บนแผนที่โดย GPS
- Switch B : จะทำงานร่วมกับ switch C โดยตำแหน่งที่ 1 จะใช้ flight mode ที่ 1,2,4 ส่วนตำแหน่งที่ 2 จะใช้ flight mode ที่ 3,5,6
- Switch C : จะตั้งค่า flight mode ไว้ตามลำดับดังนี้
 1. Loiter : โหมดลือคความสูงและตำแหน่งด้วย GPS
 2. AltHold : โหมดลือคความสูงโดยอัตโนมัติ
 3. PosHold : โหมดนี้คล้าย Loiter ควบคุม roll และ pitch ได้เมื่อจอย throttle ไม่อยู่ตรงกลาง
 4. Stabilize : โหมดควบคุมแบบ manual กำหนดการเอียงของโดรนไว้ที่ 45 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. PosHold : เหมือนข้อ 3
 6. PosHold : เหมือนข้อ 3
- Switch D : ตั้งค่าโหมด RTL หรือ Return to Launch ให้บินกลับมายังจุดที่เริ่มบินด้วย GPS



รูปที่ 3.10 การตั้งค่า Flight mode

การตั้งค่า PID

การตั้งค่า PID ค่าที่ได้มานั้นจะมาจากการทดลองโดยทำการทดลองให้ลอยอยู่กับที่แล้วปรับจนค่าไปเรื่อยๆจนได้ค่าที่เหมาะสมแล้ว โดยผลการจูนหากตั้งค่าไม่ดีจะมีผลโดยส่วนมากคือ Roll และ Pitch หากค่าสูงไปจะเกิดการแกว่งของค่า Roll และ Pitch ทำให้โดรนมีอาการสั่นได้ หากต่ำไป จะทำให้การตอบสนองช้า โดยส่วนมากการจูนจะตั้งค่า PID ให้เหมือนกัน YAW หากค่า P สูงไปโดรนอาจจะสั่น หากต่ำไปโดรนอาจจะทรงตัวได้ไม่มั่นคง เป็นต้น การจูนจะคำนึงถึงค่า P,I,D ดังนี้

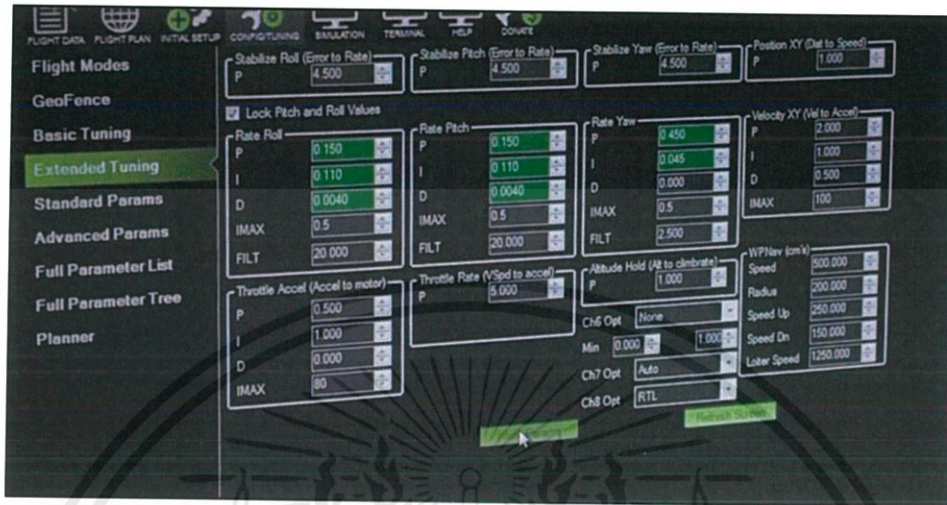
P – Gain : หากสูงเกินไปจะทำให้โดรนอาจจะมีการสั่นสูงมาก,หากต่ำไปโดรนอาจจะมีการตอบสนองต่อการควบคุมช้าลง

I – Gain : หากสูงเกินไปเมื่อโดรนมีการเอียงจะเอียงกลับคืนสู่สมดุลช้ากว่าปกติหรือแก้ไขสู่สมดุลช้าลง , หากต่ำเกินไปเมื่อโดรนเอียงจะปรับการเอียงด้วยความถี่สูงทำให้เกิดการสั่นแล้วจึงสู่สมดุลได้

D – Gain : หากสูงเกินไปจะเกิดสัญญาณรบกวนอาจส่งผลกระทบต่อมอเตอร์ให้ร้อนเกินได้ , หากต่ำเกินไปการปรับสู่สมดุลจะช้าลง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

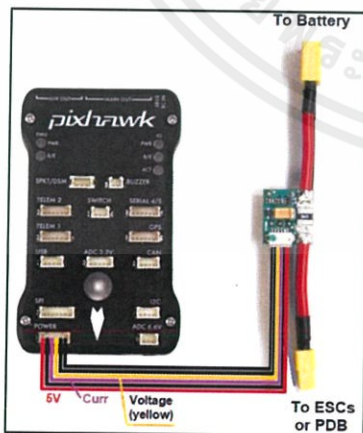
โดยปกติแล้วเราจะปรับค่า Roll และ Pitch ให้เหมือนกันเพราะควบคุมการลอยเอียงเหมือนกัน ส่วน Yaw จะควบคุมการหมุนตัวโดรน ดังนั้นจะปรับค่า PID ได้ประมาณดังนี้



รูปที่ 3.11 การตั้งค่า PID

3.1.3 การวัดแรงยกมอเตอร์และประเมินน้ำหนักโหลด

แรงยกมอเตอร์ของโดรนนั้นเป็นสิ่งสำคัญเพราะจะทำให้ทราบว่าโดรนนั้นสามารถบรรทุกน้ำหนักได้เท่าไหร่โดยเราจะทำการวัดแรงยกมอเตอร์แต่ละตัวโดยการทดลองเราจะใช้แบตเตอรี่ 14.8V 5500 mAh ในการทดลองโดยจ่ายไฟผ่าน Power module ของ flight controller ซึ่งมีวงจรที่ทำหน้าที่วัดกระแสและแรงดันที่ใช้งานขณะบินพร้อมบันทึกข้อมูลเป็น log file ในการใช้งานแต่ละครั้ง



รูปที่ 3.12 การต่อ Power Module

เก็บข้อมูลกระแสและแรงดัน



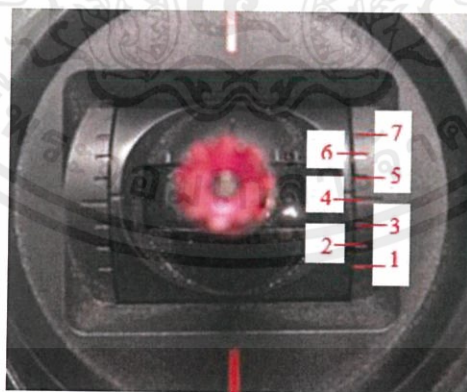
รูปที่ 3.13 การวัดแรงยกของมอเตอร์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 วัดแรงยกของมอเตอร์

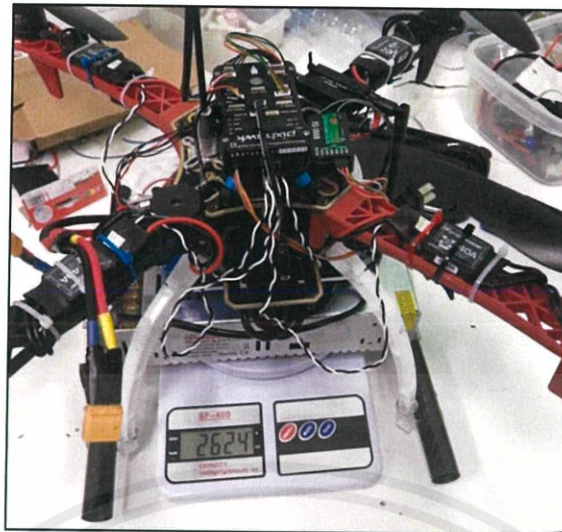
Level	แรงยกมอเตอร์ ตัวที่ 1 (g)	แรงยกมอเตอร์ ตัวที่ 2 (g)	แรงยกมอเตอร์ ตัวที่ 3 (g)	แรงยกมอเตอร์ ตัวที่ 4 (g)
1	16	24	22	20
2	95	107	100	118
3	155	161	138	158
4	240	260	220	230
5	334	330	295	280
6	452	478	448	467
7	636	623	618	621

จากผลรวมของแรงยกมอเตอร์ที่ระดับ 7 นั้น ดังนั้น โดรนสามารถยกน้ำหนักได้มากที่สุดที่ 2,498 กรัม



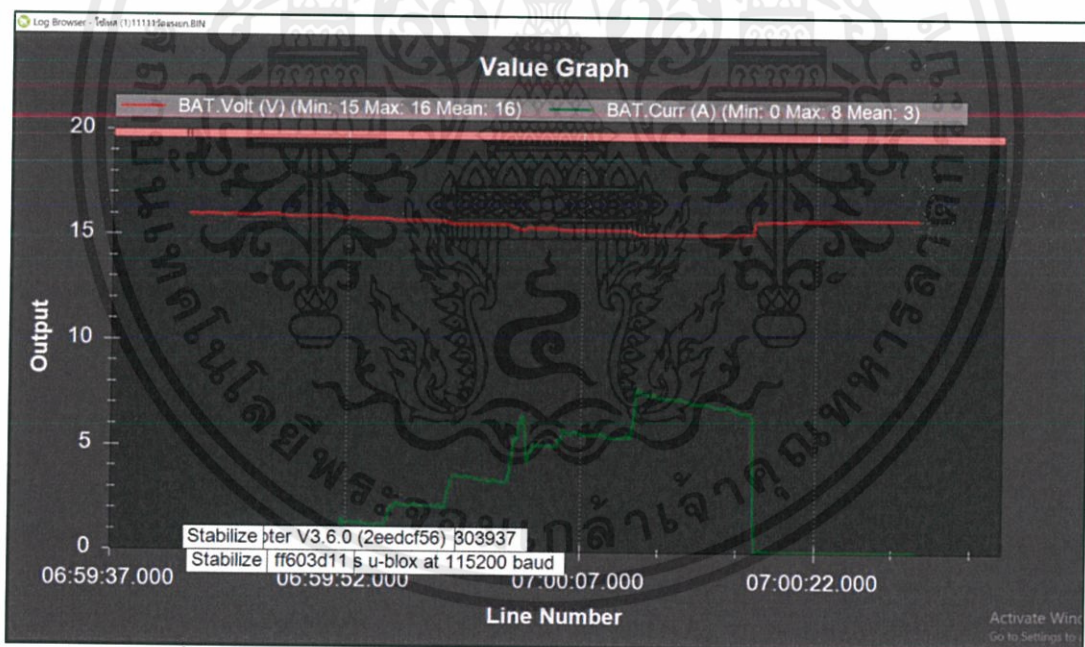
รูปที่ 3.14 Level ของรีโมทวิทยุ RC

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 น้ำหนักของโดรนที่ใช้ 2,624 กรัม

จากการทดลองนำข้อมูล log file ที่ได้จากการวัดมา Plot graph ด้วยโปรแกรม Mission Planner

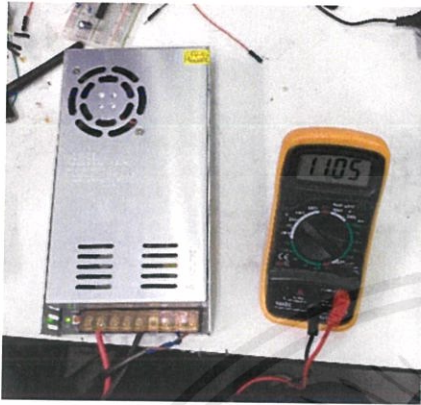


รูปที่ 3.16 แสดงกราฟแรงดันและกระแสจากโปรแกรม Mission Planner

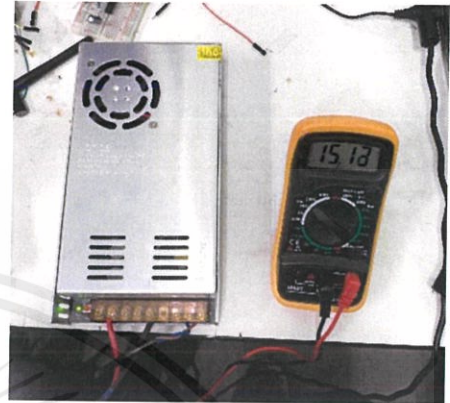
จากกราฟพล็อตจากการทดลองด้วยมอเตอร์ 1 ตัวดังนั้นกระแสรวมเมื่อใช้ 4 ตัวคือ $8 \times 3 = 24 \text{ A}$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้จัดทำได้คาดคะเนไว้ว่าเมื่อมอเตอร์รับน้ำหนักโหลดได้มากขึ้นจึงควรเผื่อกระแสให้มากขึ้นไว้ โดยเลือก Switching Power Supply 12V 50A สามารถปรับระดับแรงดันได้ 11.0V – 15.11V



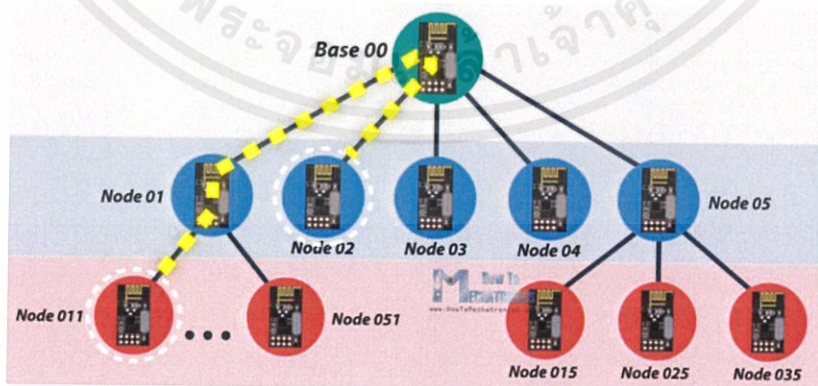
รูปที่ 3.17 แรงดันต่ำสุดสวิตซ์ซิงพาวเวอร์
ซ์พพลายที่ 11.05V



รูปที่ 3.18 แรงดันต่ำสุดสวิตซ์ซิง
พาวเวอร์ซ์พพลายที่ 15.13V

3.2 การออกแบบการตรวจจับความเคลื่อนไหวพร้อมส่งข้อมูลผ่านอุปกรณ์สื่อสารไร้สาย

จากที่กล่าวไปในการใช้งานโดรนที่ใช้พลังงานจากสายส่งในการประยุกต์ใช้ประโยชน์ต่างๆ ผู้จัดทำเล็งเห็นความสามารถของ NRF24L01 ที่สามารถใช้งานในรูปแบบของ Mesh network ซึ่งก็คือการรับส่งข้อมูลออกไปในรูปแบบของโครงข่ายส่งข้อมูลออกไปเป็นทอดๆ ผู้จัดทำจึงนำมาประยุกต์ใช้งานในการรับส่งไร้สายซึ่งช่วยขจัดเคยปัญหาเรื่องการรับส่งที่อาจจะสั้นเกินไปได้

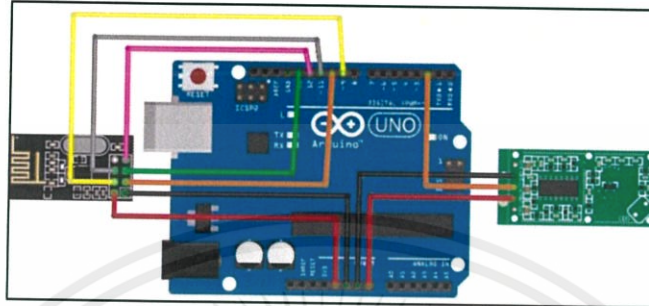


รูปที่ 3.19 บล็อกไดอะแกรมการทำงานของ Mesh network

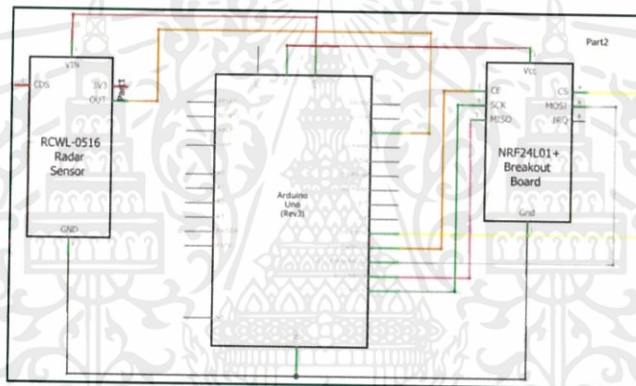
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ส่วนประกอบตัวส่งสัญญาณ

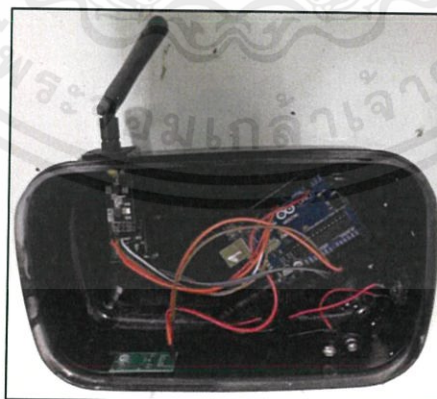
ในส่วนประกอบของตัวส่งสัญญาณจะประกอบไปด้วย Arduino UNO , เซนเซอร์ RCWL-0516 และ โมดูลเสาอากาศไร้สาย NRF24L01



รูปที่ 3.20 การต่ออุปกรณ์ตัวส่งสัญญาณ



รูปที่ 3.21 Schematics ของตัวส่งสัญญาณ

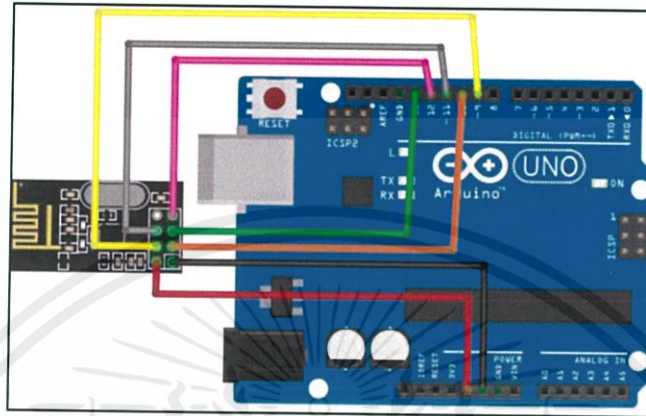


รูปที่ 3.22 ชุดตัวส่งสัญญาณ

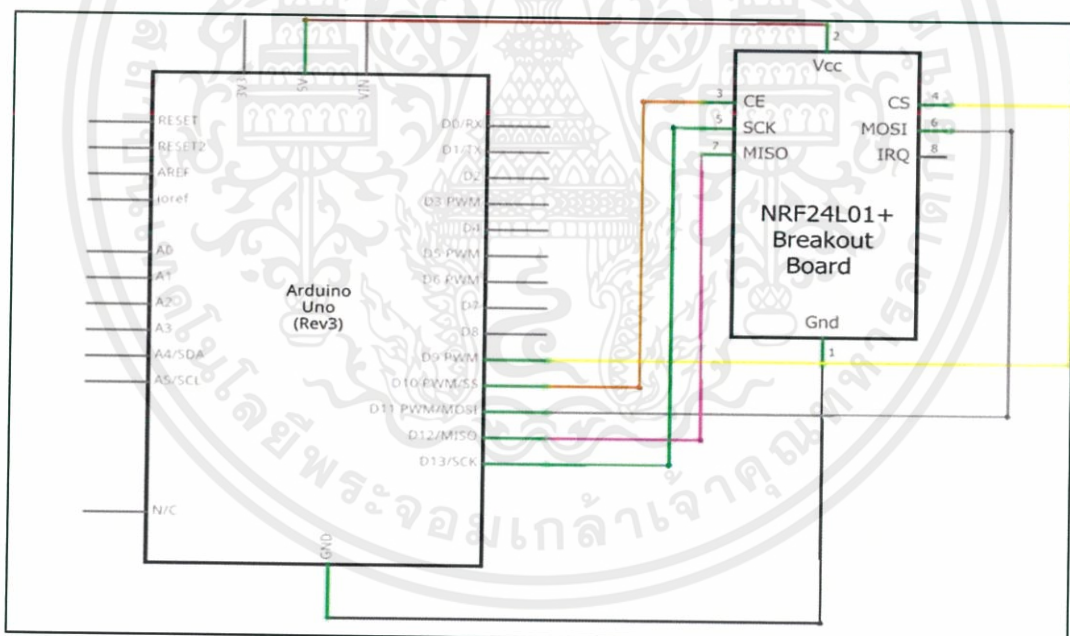
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนประกอบตัวรับ - ส่งสัญญาณ

ในส่วนประกอบของตัวรับ - ส่งสัญญาณจะประกอบไปด้วย Arduino UNO และ โมดูลเสาอากาศไร้สาย NRF24L01

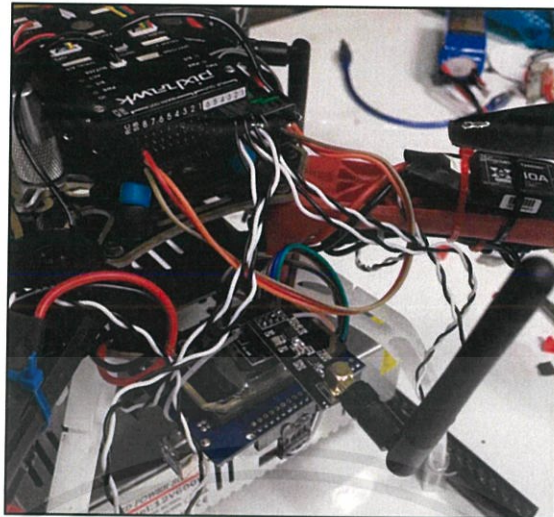


รูปที่ 3.23 การต่ออุปกรณ์ตัวรับ - ส่งสัญญาณ



รูปที่ 3.24 Schematics ของตัวรับ - ส่งสัญญาณ

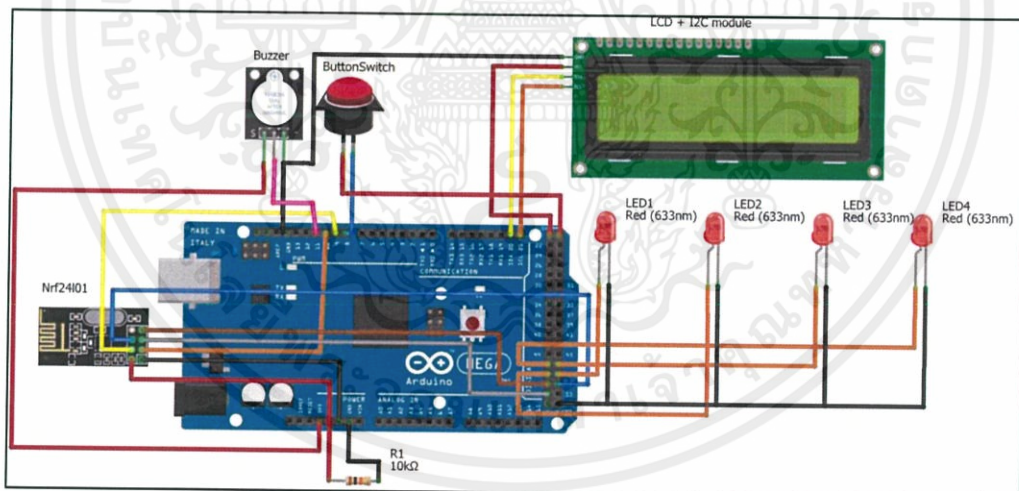
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 ชุดตัวรับ - ส่งสัญญาณ

3.2.3 ส่วนประกอบตัวรับสัญญาณ

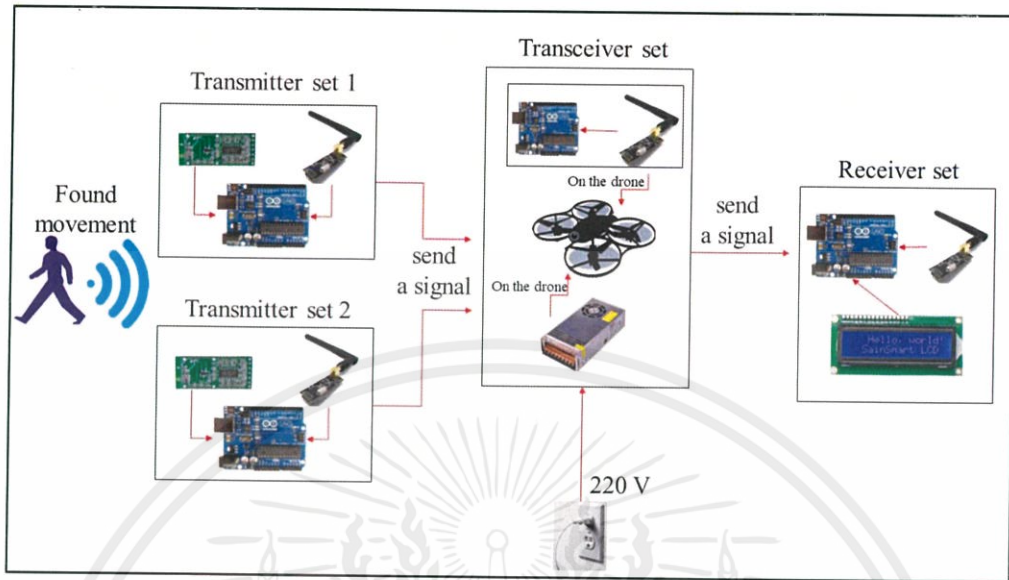
ในส่วนประกอบของตัวรับสัญญาณจะประกอบไปด้วย Arduino Mega , โมดูลเสาอากาศไร้สาย NRF24L01 , ไฟ LED , จอแสดงผล , สวิตช์ และ Buzzer



รูปที่ 3.26 การต่ออุปกรณ์ตัวรับสัญญาณ

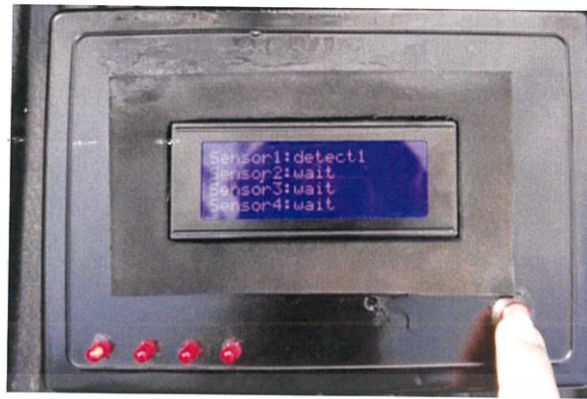
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 หลักการทำงานระบบรับส่งข้อมูลไร้สาย NRF24L01



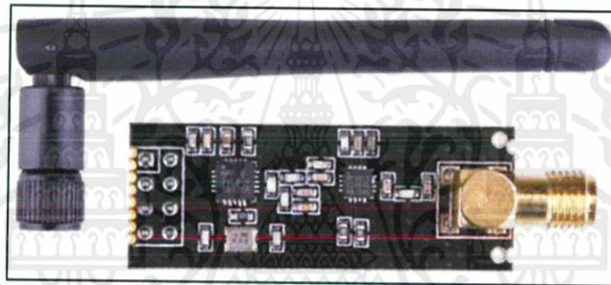
รูปที่ 3.29 บล็อกไดอะแกรมแสดงการรับส่งข้อมูลตรวจจับการเคลื่อนไหว

บล็อกไดอะแกรมนี้แสดงการทำงานโดย เมื่อมีการพบความเคลื่อนไหวบริเวณใกล้เซนเซอร์ RCWL-0516 ตัวเซนเซอร์จะทำงานโดยส่งค่าลอจิก HIGH เมื่อมีการตรวจพบแล้วส่งค่าประมวลผลในไมโครคอนโทรลเลอร์ Arduino โดยแสดงค่าง่ายๆให้เข้าใจง่ายคือ Detect แล้วจึงค่านี้นี้ผ่านโมดูลสื่อสารไร้สาย NRF24L01 ซึ่งตามคุณสมบัติที่ผู้ผลิตกล่าวคือสูงสุดที่ 1100 เมตร แต่จะมีการลดทอนของสัญญาณการรับส่งลงเมื่อผ่านสิ่งกีดขวางจะทำให้ระยะการส่งน้อยลงไป ดังนั้นจึงได้ใช้โดรนบินขึ้นฟ้าไปโดยขอบเขตโปรเจกต์นี้คือ 12 เมตรเป็นตัวรับ-ส่งสัญญาณเพื่อเลี่ยงปัญหาการที่สัญญาณจะถูกลดทอนในแนวราบจากนั้นจะส่งข้อมูลต่อไปยังโมดูล NRF24L01 ที่อยู่กับชุดแสดงผลผ่านจอ LCD ด้วย Arduino ที่อยู่ภาคพื้นดิน หากมีการตรวจพบจะแสดงที่หน้าจอทันทีและส่งเสียงแจ้งเตือนด้วย Buzzer



รูปที่ 3.30 จอแสดงผลเมื่อได้รับสัญญาณ

3.2.5 การตั้งค่าอุปกรณ์ NRF24L01



รูปที่ 3.31 โมดูล NRF24L01 + เสาอากาศรับส่ง(PA)

ในส่วนการตั้งค่าต่อไปนี้เราจะกำหนดไว้ในส่วนโค้ดในโปรแกรม Arduino IDE

- การตั้งค่าย่านความถี่และแชนแนล (Frequency Channel)

อุปกรณ์นั้นมีแชนแนลในย่านความถี่ 2.4 – 2.525 GHz โดยกำหนดแชนแนลได้ตั้งแต่แชนแนลที่ 0 – 125 คำนวณได้จากสูตร $Freq_{(Selected)} = 2400 + CH_{(Selected)}$ ซึ่งระบุไว้ในดาต้าชีทของผู้ผลิต

โดย $Freq_{(Selected)}$ คือย่านความถี่ที่ต้องการ (GHz)

$CH_{(Selected)}$ คือ แชนแนลที่ต้องการตั้งค่า (MHz)

ดังนั้นเราต้องค่า Channel ไว้ที่ 90 นั่นก็คือย่านความถี่ช่องสัญญาณคือ 2.490 GHz

```
network.begin(90, this_node); //(channel, node address)
```

- การตั้งค่าอัตราการส่งข้อมูล (Air data rate)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล NRF24L01 นั้นสามารถตั้งค่าอัตราการส่งได้ 3 ระดับคือ 250 Kbps , 1 Mbps , 2 Mbps โดยมีค่า RF output คือ -6 dBm , -12 dBm , -18 dBm ซึ่งเมื่อกำหนดที่ตัวรับการตั้งค่า 250 kbps จะมีค่าการตอบสนอง (Sensivity หรือ Receive power) ดีที่สุด โดยค่าการตอบสนองของ 250 kbps นั้นคือ -94 dBm ในขณะที่ค่าที่เหลือนั้นมีค่าการตอบสนองที่น้อยกว่าคือ -88 dBm และ -82 dBm ซึ่งจะเป็นผลให้ได้ระยะทางการรับส่งที่น้อยกว่า ดังนั้นเราเลือก “ 250 kbps “

```
radio.setDataRate(RF24_250KBPS);
```

- การตั้งค่าฟังก์ชัน Power Amplifier (PA)

ในการตั้งค่า level ของ PA นั้นเราจะเลือกการตั้งค่าด้วยคำสั่ง RF24_PA_MIN , RF24_PA_LOW , RF24_PA_HIGH และ RF24_PA_MAX ผู้จัดทำได้ทำการทดสอบการส่งโดยตั้งค่าระดับ PA level ไว้ที่ระดับสูงสุดคือ MAX พบว่าอุปกรณ์นั้นไม่ส่งข้อมูลมา แต่เมื่อเปลี่ยนระดับทั้งสามจุดต่างกันแล้วทดสอบพบว่าสามารถรับส่งได้ดี นั้นหมายความว่าควรตั้งกำลังขยายที่เหมาะสม อุปกรณ์จึงจะทำงานดังนั้นเราจะตั้งกำลังขยายอุปกรณ์ที่ตัวโดรนกับตัวอุปกรณ์แสดงผลด้วยคำสั่ง `radio.setPALevel(RF24_PA_LOW);` เพราะจากความสูงห่างกันเพียง 12 เมตร ส่วนตัวเซนเซอร์จะตั้งการส่งไว้ที่ `radio.setPALevel(RF24_PA_MAX);` เพื่อให้ได้ระยะทางที่ไกลที่สุดเท่าที่จะเป็นไปได้ โดยเราจะแสดงผลการวัดในบถัดไป

3.2.6 การออกแบบระบบหลบหลีกวัตถุ



รูปที่ 3.32 ไดอะแกรมการต่ออุปกรณ์ HC-SR04 และ Arduino Nano เข้ากับ Pixhawk

กรณีมีสิ่งกีดขวางสองด้าน

จะวัดระยะทางเปรียบเทียบระหว่างสองด้านหากระยะทางด้าน A น้อยกว่า B จะเคลื่อนที่ออกจากผนัง A แล้วค่อยกลับมาจุดสมดุล ซึ่งเช่นเดียวกับด้าน B หากวัดระยะทางได้น้อยกว่าก็จะทำงานตามเงื่อนไขไปเรื่อยๆจนระยะทางทั้งสองด้านอ่านได้มากกว่า 1.5 เมตร



รูปที่ 3.34 ภาพระยะทางของโดรนระหว่างสิ่งกีดขวาง



บทที่ 4

การทดลองและผลการทดลอง

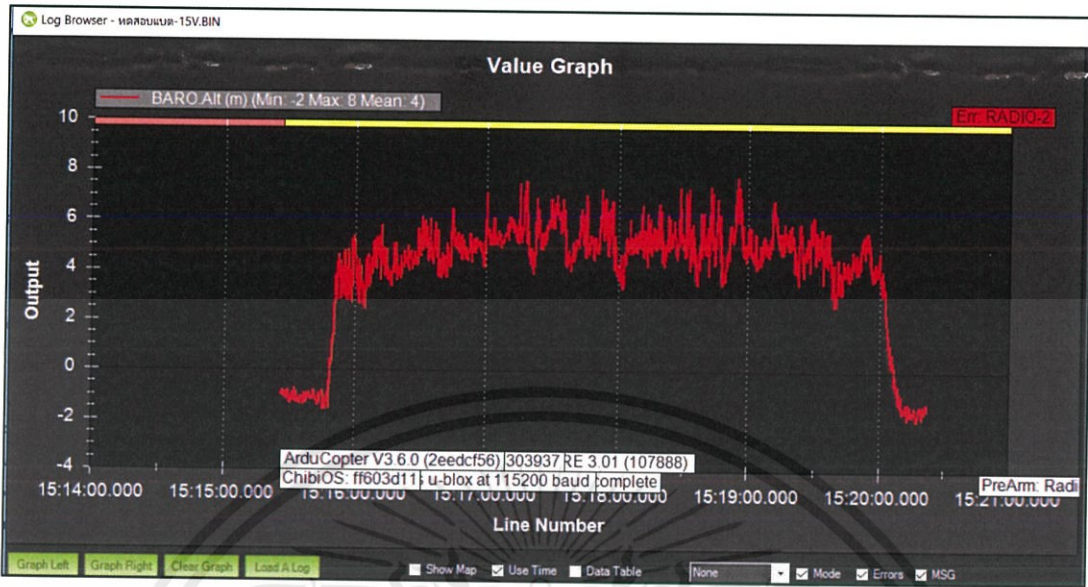
4.1 การทดสอบกระแสและแรงดันเมื่อบินที่ความสูงต่างกัน

ผู้จัดทำได้ทำการทดลองบินในระยะความสูงที่แตกต่างกันเพื่อทดลองว่าที่ความสูงต่างกันแรงดันและกระแสนั้นจะมีค่าแตกต่างกันหรือไม่ การทดลองนี้เราใช้แบตเตอรี่ 14.8V 5,200 mAh

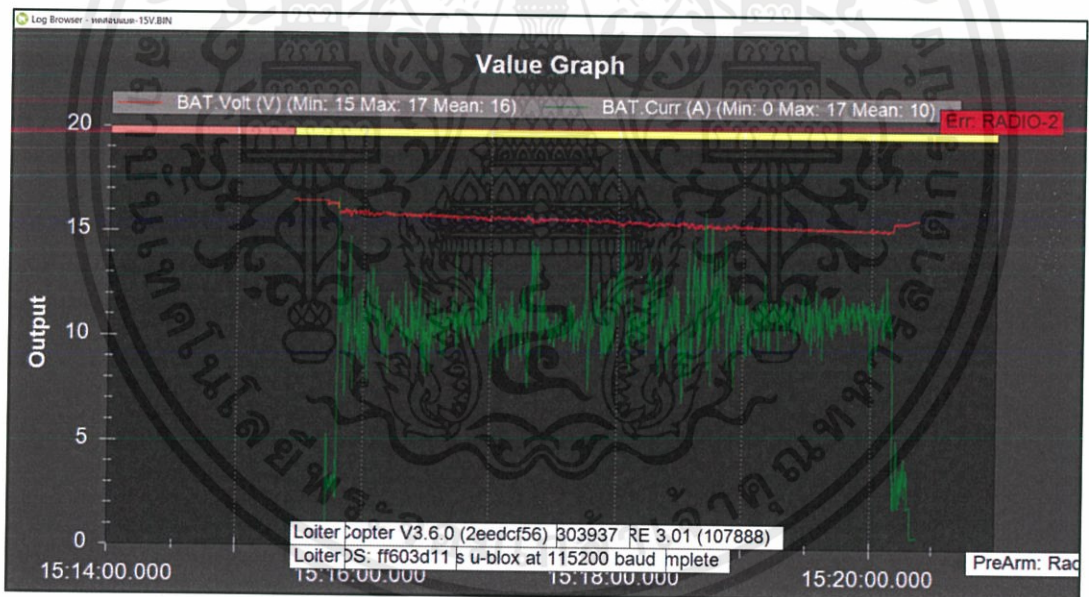


รูปที่ 4.1 การทดลองบินวัดกระแสและแรงดันพร้อมแสดงหน้าต่าง Mission Planner ขณะใช้งาน

ที่ความสูง 8 เมตร พบว่าใช้กระแสสูงสุด 17 A มีค่ากระแสเฉลี่ย 10 A



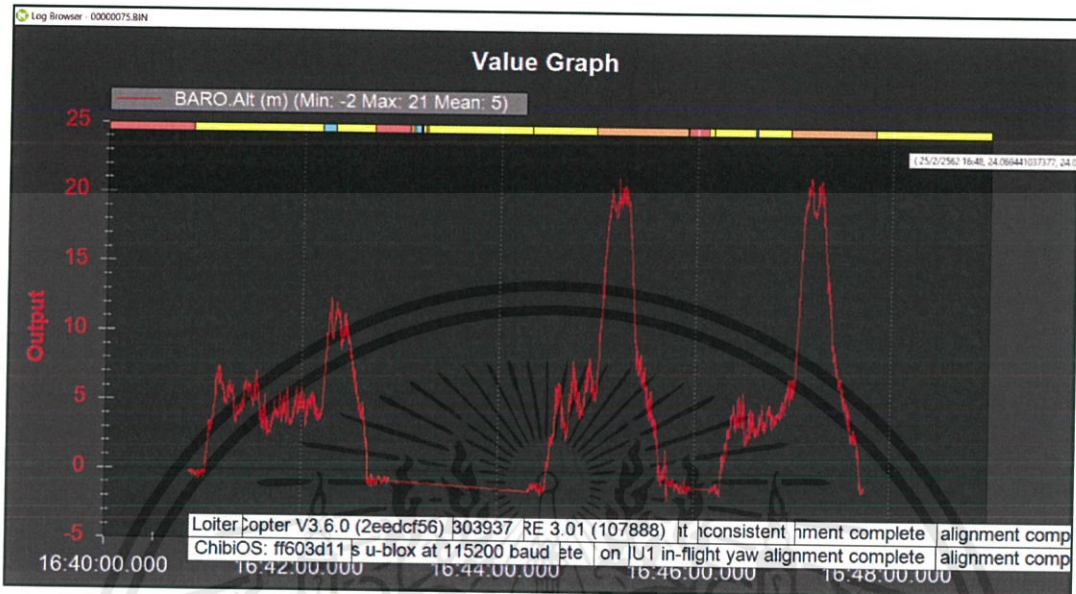
รูปที่ 4.2 กราฟระดับความสูงโดรนที่ไม่เกิน 8 เมตร



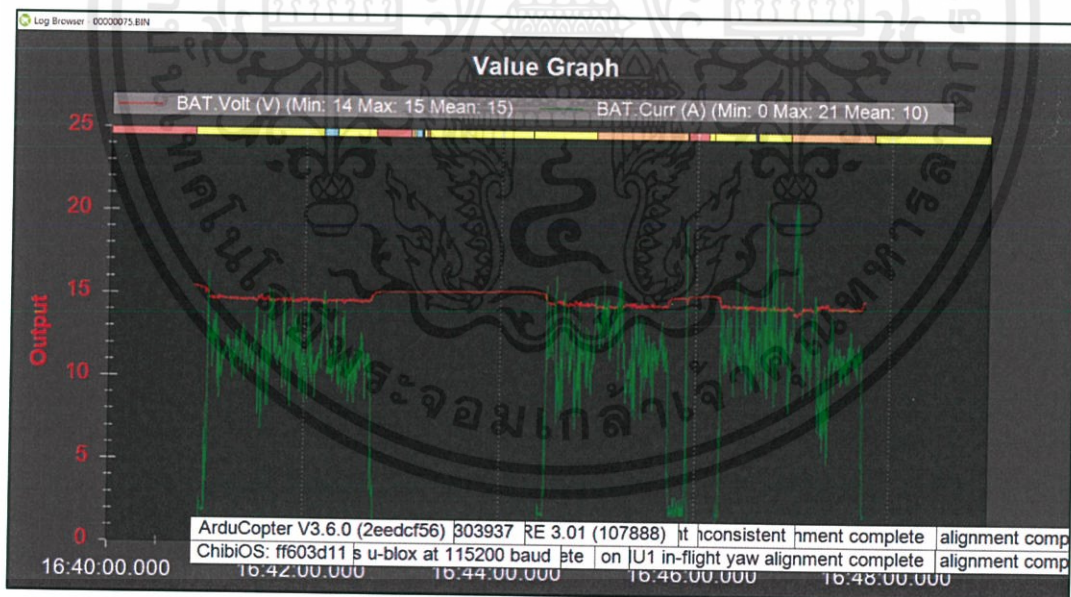
รูปที่ 4.3 กราฟแรงดันและกระแสที่ความสูงโดรนไม่เกิน 8 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระดับความสูง 7 เมตร 12 เมตร และ 21 เมตร พบว่าใช้กระแสคงที่ในแต่ละระดับ 16 A กระแสเฉลี่ย 10 A



รูปที่ 4.4 กราฟระดับความสูงโดรนที่ 8 เมตร 12 เมตร และ 21 เมตร



รูปที่ 4.5 กราฟแรงดันและกระแสที่ความสูง 8 เมตร 12 เมตร และ 21 เมตร

จากการทดลองนี้สรุปได้ว่าที่ความสูงต่าง ๆ นั้นจะใช้กระแสไฟฟ้าใกล้เคียงกันสูงสุดที่ 16 – 17 A

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบกระแสและแรงดันเมื่อบินทดสอบใช้งานจริง

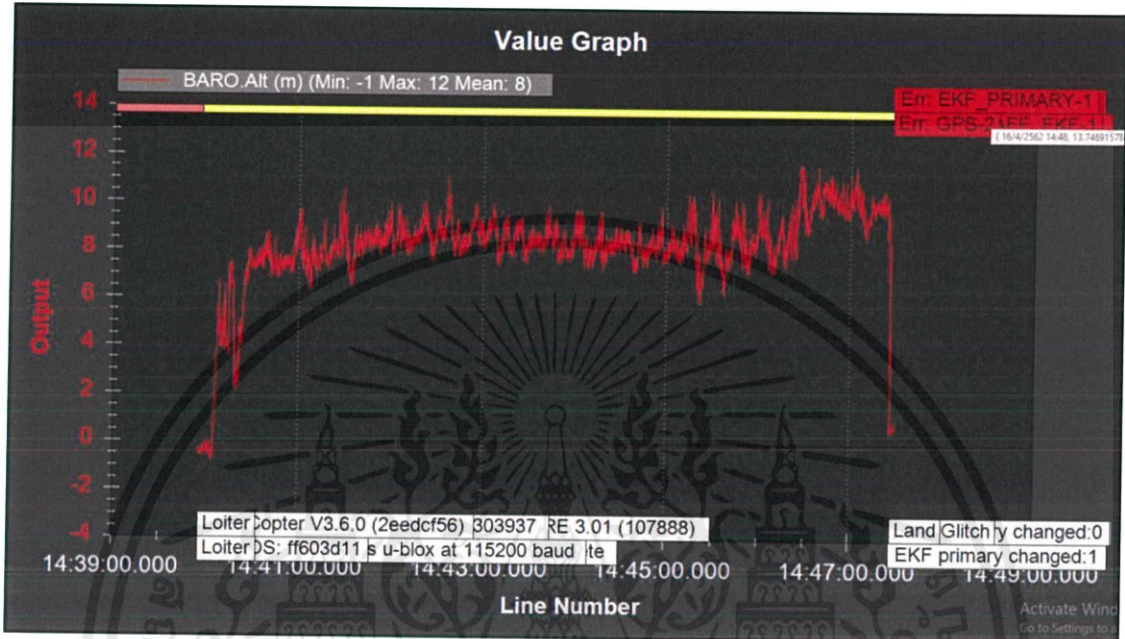
ผู้จัดทำตั้งขอบเขตของโดรนใช้พลังงานสายส่งให้ทำงานที่ไม่เกิน 13 เมตร และทดสอบการใช้พลังงานด้วย log file แสดงผลผ่านโปรแกรม Mission Planner การทดลองนี้จะใช้อุปกรณ์ด้านแหล่งจ่ายพลังงานได้แก่ อินเวอร์เตอร์ 2000W สายไฟ AC ความยาว 15 เมตร และแบตเตอรี่ lipo 12V 11,000 mAh เป็นแหล่งจ่ายไฟฟ้ากระแสตรง เมื่อจ่ายเข้ากับอินเวอร์เตอร์ แรงดันขาออกจะเป็นไฟฟ้ากระแสสลับ 220 V แล้วส่งผ่านสายส่งขึ้นไปยังสวิตซ์ซึ่งพาวเวอร์ซัพพลายเพื่อให้จ่ายไฟฟ้ากระแสตรง 15V กระแสสูงสุด 50A หลักการนี้ทำเพื่อชดเชยการส่งพลังงานแบบ DC เนื่องจากจะต้องใช้สายไฟที่ใหญ่และมีน้ำหนักมาก ต่างจากการส่งพลังงานแบบ AC ที่ส่งได้แรงดันสูงแต่กระแสน้อย ทำให้ขนาดของสายไฟเล็กลงและน้ำหนักเบา



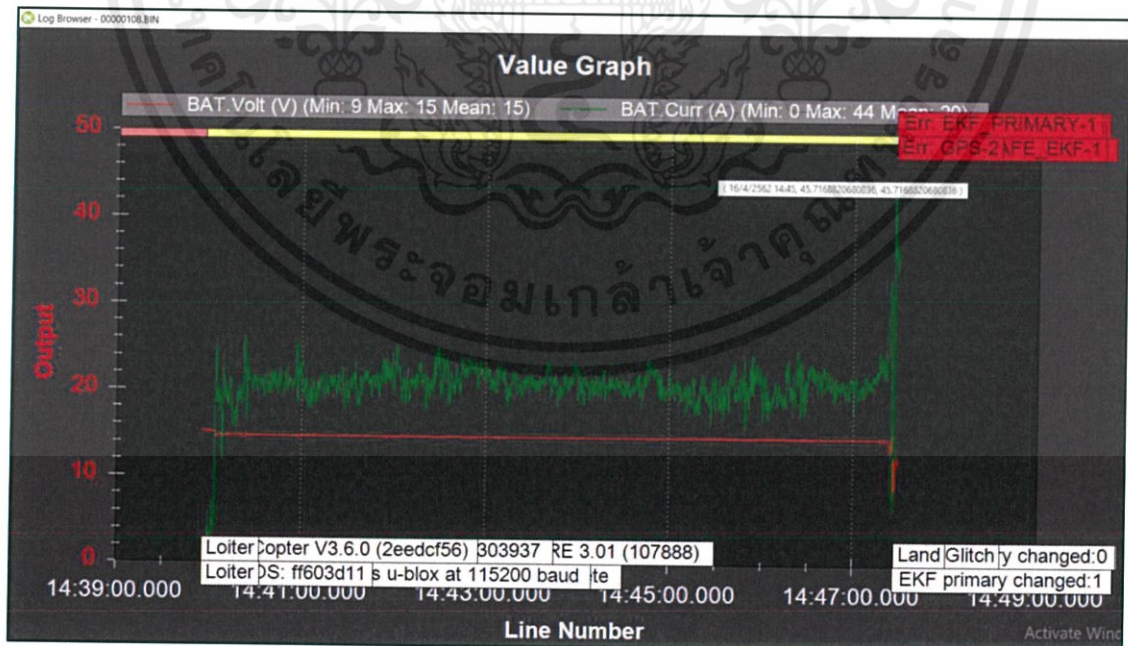
รูปที่ 4.6 ภาพการทดลองและอุปกรณ์ Inverter และ lipo - Battery

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระดับความสูงที่สุด 12 เมตร พบว่าใช้กระแสเฉลี่ย 20A ในช่วงท้ายกระแสพุ่งสูงถึง 40A เนื่องจากโดรนเกิดอุบัติเหตุ มอเตอร์ 1 ตัวใบพัดหลุดทำให้กระแสต้องถูกขับออกมาสูงมากเพื่อให้โดรนทรงตัวแต่ไม่มากพอ



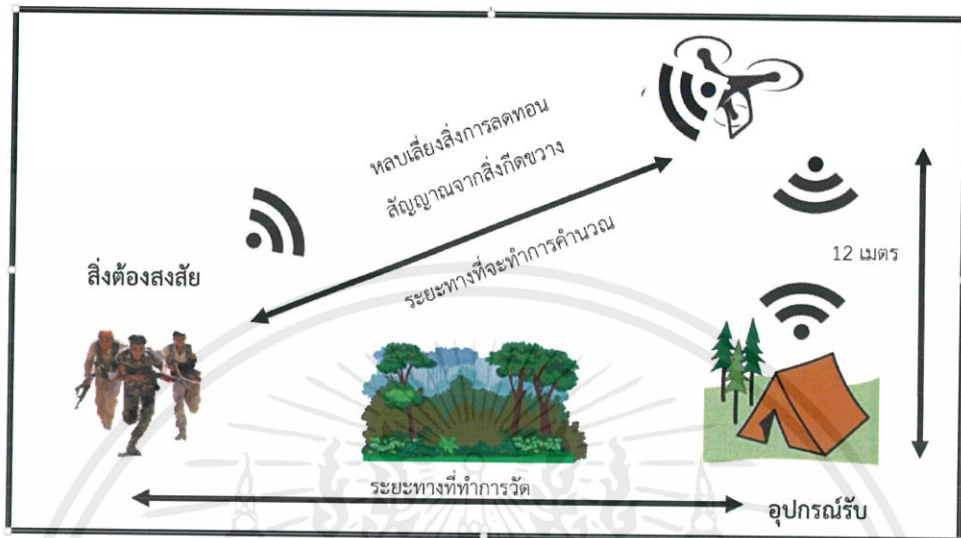
รูปที่ 4.7 กราฟความสูงของโดรนบินสูงสุด 12 เมตร



รูปที่ 4.8 กราฟกราฟแรงดันและกระแสที่ความสูง 12 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การวัดระยะการรับส่งสัญญาณและคุณภาพของสัญญาณวิทยุในการรับส่งข้อมูลของ NRF24L01



รูปที่ 4.9 แนวคิดการใช้งานและการทดลอง

จากหลักการในภาพอธิบายได้ว่า เราจะส่งสัญญาณตรวจจับจากภาคพื้นขึ้นสู่อากาศเข้าหาโดรนที่ภาครับโมดูล NRF24L01 โดยให้โดรนบินสูงเท่าที่จะทำได้ตามความยาวสายไฟและน้ำหนัก เพื่อให้สัญญาณนั้นหลบเลี่ยงสิ่งกีดขวางให้มากที่สุดให้สัญญาณถูกลดทอนน้อยที่สุด จากนั้นสัญญาณวิทยุจาก NRF24L01 จะถูกส่งต่อลงมาที่ภาครับที่อยู่ในส่วนภาคพื้นดิน

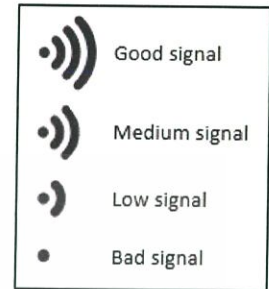
ในการทดลองเราจะวัดระยะทางในแนวราบของการส่งสัญญาณเป็นระยะๆ แล้วทำการคำนวณระยะทางสัญญาณจากเซ็นเซอร์ถึงโดรน โดยจากรูปและการวางตำแหน่งเปรียบเหมือนสามเหลี่ยมมุมฉาก ดังนั้นจะใช้ทฤษฎีของพีทาโกรัสในการคำนวณระยะทางจากสูตรคือ

$$\text{ระยะคำนวณ} = \sqrt{\text{ระยะที่วัด}^2 + 12^2} \quad (4.1)$$

จากการปรับตั้งค่า level PA จากบทที่ 3 ที่ได้ตั้งค่าไว้แล้ว เราได้ทำการทดลองใช้งานพร้อมกับวัดระยะทางไว้พร้อมกับดูผลการส่งว่ามีคุณภาพเช่นไรจึงสรุปได้ว่ามีระยะทางได้ไกลสุดเพียง 500 เมตร แสดงผลการทดลองดังตารางต่อไปนี้

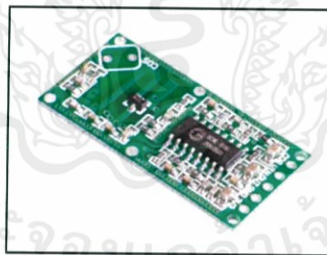
ตารางที่ 4.1 การวัดระยะสัญญาณและความแรงของสัญญาณทดสอบในที่โล่ง (Open area)

ระยะของการส่งสัญญาณ (m)	ระยะทางคำนวณ (m)	คุณภาพของสัญญาณ
50	51.4	Good signal
100	100.72	Medium signal
150	150.5	Low signal
200	200.36	Good signal
300	300.24	Good signal
400	400.2	Low signal
500	500.14	Bad signal



รูปที่ 4.10 คุณภาพของสัญญาณ

4.4 การวัดระยะการตรวจจับของเซนเซอร์ RCWL-0516

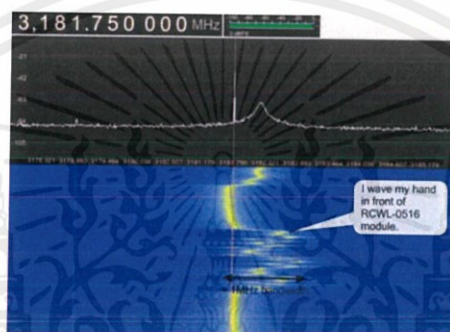


รูปที่ 4.11 เซนเซอร์ RCWL-0516

เมื่อใช้แรงดัน 9 V จ่ายให้เซนเซอร์มีระยะทางตรวจจับอยู่ที่ 3 เมตรเป็นรัศมีวงกลม แต่บางครั้งอาจจะไม่มีการส่งค่าตรวจจับเนื่องจากความไม่เสถียรของเซนเซอร์ในบางครั้ง การทำงานเมื่อมีการเคลื่อนไหว คลื่นสัญญาณจะเปลี่ยนไปแล้วส่งค่าลอจิก HIGH ออกมาเพื่อแสดงการตรวจจับ



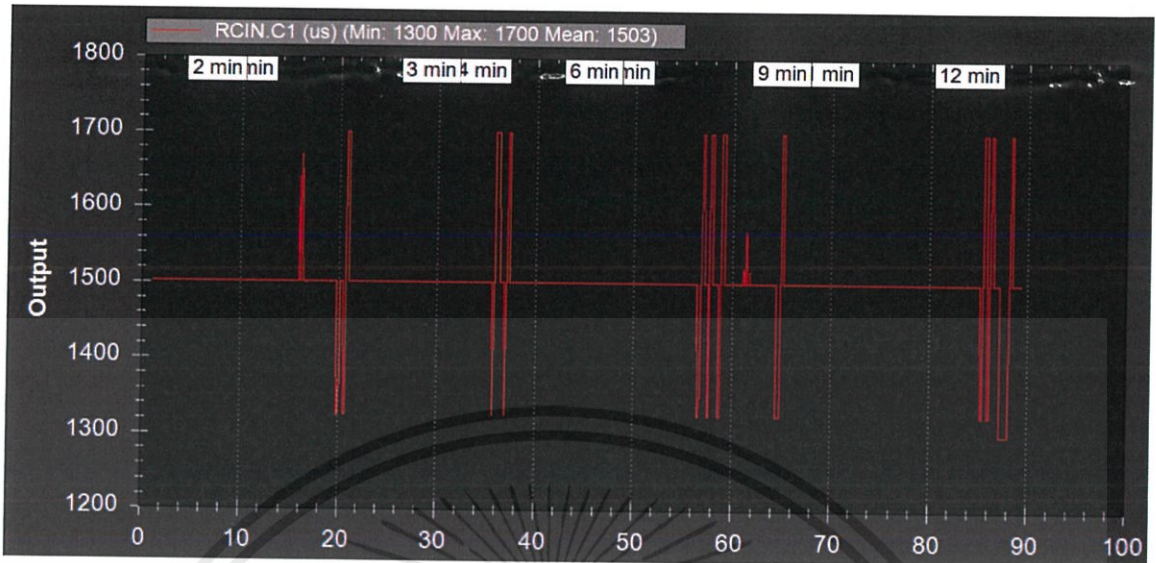
รูปที่ 4.12 ทดลองการทำงานของเซนเซอร์



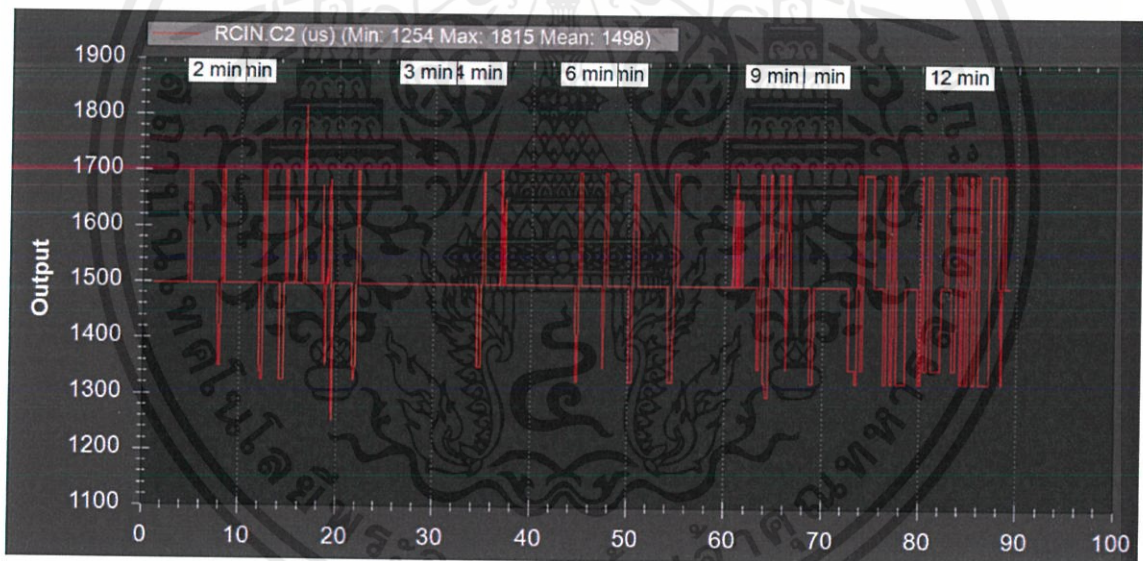
รูปที่ 4.13 การเปลี่ยนแปลงของคลื่นเมื่อมีการเคลื่อนไหว

4.5 การทดสอบระบบหลัก

ผลการทดลองเราจะแสดงค่าการตอบสนองตามเงื่อนไขที่กำหนดไว้ด้วยโปรแกรม Mission planner ซึ่งค่าที่ออกมาจะตรงกับเงื่อนไขที่เรากำหนดไว้ นั่นคือ เคลื่อนที่ด้วยค่า Analog 1700 μS และ เคลื่อนที่กลับมาจุดสมดุลด้วยค่า 1300 μS หรืออาจจะกลับค่ากันในกรณีทิศทางตรงกันข้าม



รูปที่ 4.14 กราฟ RC input ที่มาจาก Arduino ของแชนแนลที่ 1 (ด้านหน้า-ด้านหลัง)



รูปที่ 4.15 กราฟ RC input ที่มาจาก Arduino ของแชนแนลที่ 2 (ด้านซ้าย-ด้านขวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองในเรื่องการทดสอบกระแสและแรงดันที่ความสูงต่างๆ พบว่าจะใช้กระแสที่มีค่าใกล้เคียงกันแม้ว่าความสูงจะต่างกัน จากการหาข้อมูลและสอบถามผู้เชี่ยวชาญเรื่องโดรน มีข้อมูลว่า เมื่อโดรนบินสูงขึ้นโดยจะทำการเร่งความเร็วรอบมอเตอร์ขึ้นไป จะทำให้เกิดแรงยก (T) มากขึ้น เอาชนะน้ำหนักของ โดรน (Mg) และเมื่อหยุดนิ่งในความสูงที่ต้องการจะลอยตัวกลางอากาศด้วยแรงยกมีค่าเท่ากับน้ำหนัก นั้นหมายความว่า จะทำให้ความเร็วรอบมีค่าเท่าเดิมส่งผลให้กระแสไฟฟ้าที่ใช้ขับเคลื่อนมอเตอร์มีค่าเท่าเดิมเช่นกัน

จากการทดลองเรื่องการวัดระยะการรับส่งสัญญาณและคุณภาพของสัญญาณวิทยุในการรับส่งข้อมูลของ NRF24L01 นั้นผลออกมามีค่าไม่เท่ากับระยะทางที่ผู้ผลิตระบุไว้ว่าใช้งานได้ 1,100 เมตร จากที่สรุปไว้ในตารางสรุปผลนั้นระยะใช้งานควรอยู่ที่ 300 เมตร เพราะหลังจากนั้นการส่งจะเริ่มไม่เสถียร รวมไปถึงการส่งสัญญาณที่อาจจะช้าเนื่องจากการตีเลยของตัวอุปกรณ์ที่เป็นตัวรับและส่ง รวมไปถึงอัตราการส่ง (Data rate)

ในเรื่องของเซ็นเซอร์ตรวจจับความเคลื่อนไหวนั้นบางครั้งอาจจะทำงานผิดพลาดซึ่งก็เป็นปัญหาในด้านหนึ่งของเซ็นเซอร์นี้

จากการทดลองระบบหลบหลีกพบว่าใช้ได้น่าพอใจ แต่มีบางครั้งการหลบหลีกไม่สามารถหลบพ้นได้เนื่องจากการส่งสัญญาณวิทยุเรากำหนดที่ค่าอนาล็อกค่าๆหนึ่งซึ่งในความเป็นจริงอาจหลบไม่พ้น

5.2 วิจารณ์ผลการทดลอง

ในเรื่องของน้ำหนักโหลดอาจจะมากเกินไปเนื่องจากต้องใช้สวิตชิงพาวเวอร์ซัพพลาย ดังนั้นเป็นไปได้ในอนาคตอาจจะพัฒนาออกแบบวงจรสวิตชิงพาวเวอร์ซัพพลายใหม่ให้เล็กลง

การทดลองบางครั้งมีการตกเสียหาย ควรปรับปรุงโครงใหม่เก็บสายไฟให้มิดชิดมากที่สุด การส่งสัญญาณของ NRF24L01 ส่งได้ระยะไม่ตรงกับสเปคมาก อาจจะผลมาจากสัญญาณรบกวนที่มาจากสาย PCB เอง ดังนั้นอาจจะไปใช้รุ่นใหม่ที่มีการออกแบบใหม่หรืออาจจะใช้ Lorawan เพื่อให้ได้การส่งที่ดีขึ้น

ในส่วนของภาคเซ็นเซอร์ผู้จัดทำได้ใช้แหล่งจ่ายเป็นถ่าน Panasonic 9V อาจจะจ่ายกระแสได้ไม่คงที่ ทำให้การส่งสัญญาณสั้นลงหรือเซ็นเซอร์ทำงานผิดพลาดได้ ควรเปลี่ยนไปใช้ถ่าน 18650 ซึ่งจ่ายกระแสได้สูงและคงที่แต่ก็มีข้อจำกัดด้วยราคาแพง



บรรณานุกรม

- [1.] ชนิดของมอเตอร์ (ออนไลน์).
สืบค้นจาก : blog.ebikr.com/posts/2762 [20 ตุลาคม 2561]
- [2.] ปัทมา ชุกลีน, "การออกแบบและสร้างเครื่องต้นแบบอากาศยานสี่ใบพัดที่บังคับการเคลื่อนที่ด้วยตนเอง", (วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขา วิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีสุรนารี, 2557).
[20 ตุลาคม 2561]
- [3.] หลักการทำงานของเครื่องบิน (ออนไลน์).
สืบค้นจาก : www.bingun.com [1 พฤศจิกายน 2561]
- [4.] หลักการ accelerometer และ gyroscope (ออนไลน์).
สืบค้นจาก : sites.google.com/site/thaimulticopter [1 พฤศจิกายน 2561]
- [5.] PWM คืออะไร (ออนไลน์).
สืบค้นจาก : www.thitiblog.com/blog/6142 [7 พฤศจิกายน 2561]
- [6.] การสร้างสัญญาณ PPM-Sum (ออนไลน์).
สืบค้นจาก : cpre.kmutnb.ac.th/es/learning/ [10 พฤศจิกายน 2561]
- [7.] Multicopter PID Tuning Guide (ออนไลน์).
สืบค้นจาก : docs.px4.io/en [10 พฤศจิกายน 2561]
- [8.] ข้อมูลบอร์ด Pixhawk (ออนไลน์).
สืบค้นจาก : ardupilot.org [10 พฤศจิกายน 2561]
- [9.] ข้อมูลการทำ Mesh network (ออนไลน์).
สืบค้นจาก : <https://supotfrontwheel.blogspot.com/> [2 กุมภาพันธ์ 2562]
- [10.] ข้อมูลการใช้งาน NRF24L01 (ออนไลน์).
สืบค้นจาก : <https://howtomechatronics.com/> [8 กุมภาพันธ์ 2562]

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [11.] ข้อมูลเกี่ยวกับ NRF24L01 (ออนไลน์).
สืบค้นจาก: <https://lastminuteengineers.com/> [8 กุมภาพันธ์ 2562]
- [12.] ข้อมูลเซนเซอร์ RCWL-0516 (ออนไลน์).
สืบค้นจาก: <https://www.arduinoall.com/> [8 กุมภาพันธ์ 2562]
- [13.] ข้อมูลจอ LCD (ออนไลน์).
สืบค้นจาก : http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_6.pdf
[16 กุมภาพันธ์ 2562]
- [14.] ข้อมูล switching power supply (ออนไลน์).
สืบค้นจาก : <https://mall.factomart.com/principle-of-switching-power-supply/>
[16 กุมภาพันธ์ 2562]
- [15.] ข้อมูล switching power supply (ออนไลน์).
สืบค้นจาก: https://www.cpe.ku.ac.th/~yuen/204471/power/switching_regulator/
[16 กุมภาพันธ์ 2562]



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการเขียนโปรแกรมของระบบรับส่งแจ้งเตือนผู้บุกรุก

- โค้ดในส่วนของภาคส่งเซ็นเซอร์

```
#include <RF24Network.h>

#include <RF24.h>

#include <SPI.h>

RF24 radio(10, 9);      // nRF24L01 (CE,CSN)

RF24Network network(radio);  // Include the radio in the network

const uint16_t this_node = 011; // Address of our node in Octal format ( 04,031, etc)

const uint16_t node01 = 01; // Address of the other node in Octal format

char msg[20] = "" ;

void setup()
{
  SPI.begin();
  radio.begin();
  network.begin(90, this_node); //(channel, node address)

  radio.setDataRate(RF24_250KBPS);
  radio.setPALevel(RF24_PA_MAX);

  pinMode(2, INPUT);

  Serial.begin(9600);
}

void loop()
{
  network.update();

  //==== Sending =====//

  if (digitalRead(2)==1)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char msg[] = "detect1";

RF24NetworkHeader header(node01);

bool ok = network.write(header, &msg, sizeof(msg));

delay(300);

Serial.println("Detect1");
}

if (digitalRead(2)==0)
{
char msg[] = "no1 ";

RF24NetworkHeader header(node01);

bool ok = network.write(header, &msg, sizeof(msg));

Serial.println("unDetect1");

delay(300);
}
}

```

- โค้ดในส่วนของภาคส่งและรับ

```

#include <RF24Network.h>

#include <RF24.h>

#include <SPI.h>

RF24 radio(10, 9); // nRF24L01 (CE,CSN)

RF24Network network(radio); // Include the radio in the network

const uint16_t master00 = 00;

const uint16_t this_node = 01; // Address of this node in Octal format ( 04,031, etc)

const uint16_t node01 = 011; // Address of the other node in Octal format

const uint16_t node02 = 021;

const uint16_t node03 = 031;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const uint16_t node04 = 041;

char msg[20] = "" ;

void setup()

{

  Serial.begin(9600);

  SPI.begin();

  radio.begin();

  network.begin(90, this_node); //(channel, node address)

  radio.setDataRate(RF24_250KBPS);

  radio.setPALevel(RF24_PA_LOW);

}

void loop()

{

  network.update();

  //===== Receiving =====//

  while ( network.available() )

  { // Is there any incoming data?

    RF24NetworkHeader header;

    char msg[20] = "" ;

    network.read(header, &msg, sizeof(msg));

    // check header from sensor1

    if (header.from_node == 011)

    {

      RF24NetworkHeader header(master00);

      // sent from sensor1 to ground

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bool ok = network.write(header, &msg, sizeof(msg));

Serial.print("node01 = ");

Serial.println(msg);

delay(10);

}

// check header from sensor2

if (header.from_node == 021)

{

RF24NetworkHeader header(master00);

// sent from sensor2 to ground

bool ok = network.write(header, &msg, sizeof(msg));

Serial.print("node02 = ");

Serial.println(msg);

delay(20);

}

// check header from sensor3

if (header.from_node == 031)

{

RF24NetworkHeader header(master00);

// sent from sensor3 to ground

bool ok = network.write(header, &msg, sizeof(msg));

Serial.print("node03 = ");

Serial.println(msg);

delay(20);

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// check header from sensor4

if (header.from_node == 041)

{

RF24NetworkHeader header(master00);

// sent from sensor4 to ground

bool ok = network.write(header, &msg, sizeof(msg));

Serial.print("node04 = ");

Serial.println(msg);

delay(20);

}

}

}

- โค้ดในส่วนของภาครับ

#include <RF24Network.h>

#include <RF24.h>

#include <SPI.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4); //Module IIC/I2C Interface

RF24 radio(10, 9); // nRF24L01 (CE,CSN)

RF24Network network(radio); // Include the radio in the network

const uint16_t this_node = 00; // Address of this node in Octal format ( 04,031, etc)

const uint16_t node01 = 01;

char msg[20] = "";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int pinButtonSensor = 8;

int checkSensor1 = 0;

int checkSensor2 = 0;

int checkSensor3 = 0;

int checkSensor4 = 0;

int LED1 = 49;

int LED2 = 48;

int LED3 = 47;

int LED4 = 46;

int buzzer = 11;

void setup()
{
  Serial.begin(9600);
  SPI.begin();
  radio.begin();
  network.begin(90, this_node); //(channel, node address)
  radio.setDataRate(RF24_250KBPS);
  radio.setPALevel(RF24_PA_LOW);
  pinMode(buzzer,OUTPUT);
  pinMode(LED1,OUTPUT);

  pinMode(LED2,OUTPUT);

  pinMode(LED3,OUTPUT);

  pinMode(LED4,OUTPUT);

  pinMode(pinButtonSensor,INPUT);

  lcd.begin();

  lcd.backlight();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    lcd.home();

    lcd.setCursor(0, 0);

    lcd.print("Sensor1:");

    lcd.setCursor(0, 1);

    lcd.print("Sensor2:");

    lcd.setCursor(0, 2);

    lcd.print("Sensor3:");

    lcd.setCursor(0, 3);

    lcd.print("Sensor4:");
}
void loop()
{
    lcd.setCursor(8, 0);
    lcd.print("wait ");
    lcd.setCursor(8, 1);
    lcd.print("wait ");
    lcd.setCursor(8, 2);
    lcd.print("wait ");
    lcd.setCursor(8, 3);
    lcd.print("wait ");

    if(digitalRead(pinButtonSensor) == 1)
    {
        // ปิดเสียงทั้งหมดในนี้

        digitalWrite(LED1,LOW);

        digitalWrite(LED2,LOW);

        digitalWrite(LED3,LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(LED4,LOW);

digitalWrite(buzzer,HIGH);
}

network.update();

//===== Receiving =====///////// ฝั่ง LCD

while ( network.available() )
{ // Is there any incoming data?

  RF24NetworkHeader header;

  // กำหนดตัวแปร msg เพื่อรับค่าข้อมูล

  char msg[20] = "";

  network.read(header, &msg, sizeof(msg)); // Read the incoming data
  Serial.print(header.from_node);
  Serial.print(" : ");

  // แปลงตัวแปร msg เป็น string เพื่อต่อการ check เงื่อนไข
  String stringMsg = String(msg);

  // แสดงผลเพื่อเช็คค่าเฉยๆ
  Serial.print(stringMsg);
  Serial.println(stringMsg.length());

  delay(1000);

  int checkStateButton = digitalRead(pinButtonSensor);

  if(checkStateButton == 1)

  {

    // ปิดเสียงทั้งหมดในนี้

    digitalWrite(LED1,LOW);

    digitalWrite(LED2,LOW);

    digitalWrite(LED3,LOW);

    digitalWrite(LED4,LOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(buzzer,HIGH);
}

if(stringMsg == "no1 " || stringMsg == "detect1")
{
lcd.setCursor(8, 0);

lcd.print(msg);

checkSensor1=0;

if(stringMsg == "detect1" && checkStateButton == 0)
{
// busser ร้อง
digitalWrite(LED1,HIGH);
digitalWrite(buzzer,LOW);
delay(3000);
}
else
{
// ให้ busser หยุดร้อง เมื่อได้ค่า no2
digitalWrite(LED1,LOW);
digitalWrite(buzzer,HIGH);

}
}
}

```

```

if(stringMsg == "no2 " || stringMsg == "detect2")

```

```

{

lcd.setCursor(8, 1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd.print(msg);

checkSensor2=0;

if(stringMsg == "detect2" && checkStateButton == 0)
{
    // busser ร้อง

    digitalWrite(LED2,HIGH);

    digitalWrite(buzzer,LOW);

    delay(3000);

}
else
{
    // ให้ busser หยุดร้อง เมื่อได้ค่า no2

    digitalWrite(LED2,LOW);

    digitalWrite(buzzer,HIGH);

}
}

if(stringMsg == "no3 " || stringMsg == "detect3")
{

    lcd.setCursor(8, 2);

    lcd.print(msg);

    checkSensor3=0;

    if(stringMsg == "detect3" && checkStateButton == 0)

    {

        // busser ร้อง

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(LED3,HIGH);

digitalWrite(buzzer,LOW);

delay(3000);

}

else
{
// ให้ busser หยุดร้อง เมื่อได้ค่า no2

digitalWrite(LED3,LOW);

digitalWrite(buzzer,HIGH);

}

}

if(stringMsg == "no4 " || stringMsg == "detect4")
{
lcd.setCursor(8, 3);

lcd.print(msg);

checkSensor4=0;

if(stringMsg == "detect4" && checkStateButton == 0)
{

// busser ร้อง

digitalWrite(LED4,HIGH);

digitalWrite(buzzer,LOW);

delay(3000);

}

else
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// ให้ busser หยุดร้อง เมื่อได้ค่า no2

digitalWrite(LED4,LOW);

digitalWrite(buzzer,HIGH);
}
}

checkSensor1++;

checkSensor2++;

checkSensor3++;

checkSensor4++;

if(checkSensor1 >= 10)
{
  lcd.setCursor(8, 0);
  lcd.print("      ");
  lcd.setCursor(8, 0);
  lcd.print("wait");
  // busser หยุดร้องเมื่อไม่ได้รับค่าอะไร
  digitalWrite(LED1,LOW);
}

if(checkSensor2 >= 10)
{
  lcd.setCursor(8, 1);

  lcd.print("      ");

  lcd.setCursor(8, 1);

  lcd.print("wait");

  digitalWrite(LED2,LOW);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(checkSensor3 >= 10)
{
  lcd.setCursor(8, 2);
  lcd.print("    ");
  lcd.setCursor(8, 2);
  lcd.print("wait");
  digitalWrite(LED3,LOW);
}
if(checkSensor4 >= 10)
{
  lcd.setCursor(8, 3);
  lcd.print("    ");
  lcd.setCursor(8, 3);
  lcd.print("wait");
  digitalWrite(LED4,LOW);
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการเขียนโปรแกรมหลัก

```
#include <checksum.h>
#include <mavlink_conversions.h>
#include <mavlink_helpers.h>
#include <mavlink_types.h>
#include <protocol.h>
#include <NewPing.h>

NewPing sonar0(3, 4, 300);

NewPing sonar1(5, 6, 300);

NewPing sonar2(7, 8, 300);

NewPing sonar3(9, 10, 300);

NewPing sonar4(11, 12, 100);

//ตัวแปรที่ใช้ควบคุมว่า HeartBeat จะถูกส่งทุกวินาที
unsigned long HeartbeatTime = 0;

//
ตัวแปรที่ใช้เพื่อส่ง RCOVERRIDE เดียวเท่านั้นในแต่ละครั้ง
//ที่ได้รับการแก้ไขและไม่อ้อมตัวถึงตัวควบคุมของคำสั่งชื่อที่ซ้ำซ้อน

uint16_t Pitch = 0;
uint16_t Roll = 0;
uint16_t PitchOut = 0;
uint16_t RollOut = 0;
uint16_t PitchOutTemp = 0;
uint16_t RollOutTemp = 0;
uint8_t n = 0;

///Define เป็นการกำหนดค่านิพจน์ต่างๆ ให้กับชื่อของตัวคงที่ ///
เช่น #define ledPin 3

#define NDistance 5

#define DistanceFence 100 //ระยะทางที่ตัวควบคุมเริ่มทำงาน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define AltMin      70 // ความสูงที่การควบคุมเริ่มทำ
#define DistMin    50 //ความแตกต่างขั้นต่ำระหว่างสองระยะทางของแกนเดียวกันกับ
การเคลื่อนที่
// บันทึกเพื่อบันทึกข้อมูลของเซ็นเซอร์แต่ละตัว unsigned long var = val;
var - ชื่อตัวแปร
val - ค่าของตัวแปรที่ต้องการกำหนด//
struct Sensores
{
  uint16_t Distance[NDistance] = {0};
  uint16_t AverageDistance      = 0;
  bool Fence                    = false;
  bool Active                   = false;
};
#define NSensores 5
Sensores Sensor[NSensores];

//=====PROGRAM=====
=====//
void setup()
{
  Serial.begin(57600);
}
void loop()
{
  if ( ( millis() - HeartbeatTime) > 1000 )
  {
    HeartbeatTime = millis();
    FHeartBeat();
  }
  FSensores();
  FRCOVERRIDE();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void FSensores()
{
    ShiftArrays();
    MeasureSensores();
    MediaDistance();
    CheckDistance();
}
void FRCOVERRIDE()
{
    mavlink_message_t msg;
    uint8_t buf[MAVLINK_MAX_PACKET_LEN];
    uint16_t len;
    Pitch = CheckPitch(Pitch);
    Roll = CheckRoll(Roll);
    if( Pitch != PitchOutTemp || Roll != RollOutTemp )
    {
        n = 0;
        PitchOutTemp = Pitch;
        RollOutTemp = Roll;
    }
    else
    {
        n += 1;
        if(n == 4)
        {
            RollOut = RollOutTemp;
            PitchOut = PitchOutTemp;
            RCOVERRIDE(&msg, len, buf, PitchOut, RollOut);
        }
    }
}
void ShiftArrays()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  for (uint8_t i = 0; i < NSensores; i++)
  {
    for (uint8_t j = NDistancias - 1; j > 0; j--)
    {
      Sensor[i].Distance[j] = Sensor[i].Distance[j - 1];
    }
  }
}

void MeasureSensores()
{
  Sensor[0].Distance[0] = sonar0.ping_cm();
  Sensor[1].Distance[0] = sonar1.ping_cm();
  Sensor[2].Distance[0] = sonar2.ping_cm();
  Sensor[3].Distance[0] = sonar3.ping_cm();
  Sensor[4].Distance[0] = sonar4.ping_cm();
}

void AverageDistance()
{
  for (uint8_t i = 0; i < NSensores; i++)
  {
    int Total = 0;
    uint8_t Num = 0;
    for (uint8_t j = 0; j < NDistance; j++)
    {
      if (Sensor[i].Distance[j] != 0 && Sensor[i].Distance[j] < 300)
      {
        Total += Sensor[i].Distance[j]; ////x += 10; จะเทียบเท่า x = x + 10///
        Num += 1;
      }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Num > 3)
{
    Sensor[i].AverageDistance = Total / Num;
}
else
{
    Sensor[i].AverageDistance = 0;
}
}
}
void CheckDistance()
{
    for (uint8_t i = 0; i < NSensores; i++)
    {
        if (Sensor[i].AverageDistance != 0 && Sensor[i].AverageDistance < DistanceCerca)
        {
            Sensor[i].Fence = true;
        }
        else
        {
            Sensor[i].Fence = false;
        }
    }
}
uint16_t CheckPitch(uint16_t Pitch)
{
    int16_t Difference = Sensor[0].AverageDistance - Sensor[2].AverageDistance
    if( Sensor[4].AverageDistance > AltMin || Sensor[4].AverageDistance == 0 )
    {
        if( abs(Difference) > DistMin )
        {
            if( Sensor[0].Fence == true )
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( Sensor[2].Fence == true )
{
    if( Sensor[0].AverageDistance < Sensor[2].AverageDistance )
    {
        return( Pitch = ValueRC( Sensor[0].AverageDistance, 1 ) );
    }else
    {
        return( Pitch = ValueRC( Sensor[2].AverageDistance, 0 ) );
    }
}
else
{
    return( Pitch = ValueRC( Sensor[0].AverageDistance, 1 ) );
}
else
{
    if( Sensor[2].Fence == true )
    {
        return( Pitch = ValueRC( Sensor[2].AverageDistance, 0 ) );
    }
    else
    {
        return( Pitch = 0 );
    }
}
}
else if( Sensor[0].Fence == true && Sensor[2].AverageDistance == 0 )
{
    return( Pitch = ValueRC( Sensor[0].AverageDistance, 1 ) );
}
else if ( Sensor[0].AverageDistance == 0 && Sensor[2].Fence == true )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    return( Pitch = ValueRC( Sensor[2].AverageDistance, 0 ) );
}
else
{
    return( Pitch = 0 );
}
}
else
{
    return( Pitch = 0 );
}
}
uint16_t CheckRoll(uint16_t Roll)
{
    int16_t Difference = Sensor[1].AverageDistance - Sensor[3].AverageDistance;
    if( Sensor[4].AverageDistance > AltMin || Sensor[4].AverageDistance == 0 )
    {
        if( abs(Difference) > DistMin )
        {
            if( Sensor[1].Fence == true )
            {
                if( Sensor[3].Fence == true )
                {
                    if( Sensor[1].AverageDistance < Sensor[3].AverageDistance )
                    {
                        return( Roll = ValueRC( Sensor[1].AverageDistance, 0 ) );
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{

return( Roll = ValueRC( Sensor[3].AverageDistance, 1 ) );

}

}else{

return( Roll = ValueRC( Sensor[1].AverageDistance, 0 ) );

}

}

else
{

if( Sensor[3].Fence == true )
{

return( Roll = ValueRC( Sensor[3].AverageDistance, 1 ) );

}

else
{

return( Roll = 0 );

}

}

}

else if( Sensor[1].Fence == true && Sensor[3].AverageDistance == 0 )
{

return( Roll = ValueRC( Sensor[1].AverageDistance, 0 ) );

}

else if ( Sensor[1].AverageDistance == 0 && Sensor[3].Fence == true )
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return( Roll = ValueRC( Sensor[3].AverageDistance, 1 ) );
}
else
{

return( Roll = 0 );
}
}
else
{
return( Roll = 0 );
}
}
uint16_t ValueRC( uint16_t Distance, bool inc )/////ค่าใน()จะเท่ากับค่า return()///
{
if( Distance < 30 )
{
if( inc == true )
{
return( 1700 );
}
else
{
return( 1300 );
}
}
}
else if( Distance < 90 )
{
if( inc == true )
{
return( 1675 );
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
return( 1325 );
}
}
else if( Distance < 150 )
{
if( inc == true )
{
return( 1650 );
}
else
{
return( 1350 );
}
}
}
void FHeartBeat() {
mavlink_message_t msg;
uint8_t buf[MAVLINK_MAX_PACKET_LEN];
uint16_t len;
// System ID = 255 = GCS
mavlink_msg_heartbeat_pack(255, 0, &msg, MAV_TYPE_QUADROTOR,
MAV_AUTOPILOT_GENERIC, 0, 1, 0);
// Copy the message to send buffer
len = mavlink_msg_to_send_buffer(buf, &msg);
// Send the message (.write sends as bytes)
Serial.write(buf, len);
//Serial.write("\n\rHeartBeat\n\r");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void RCOVERRIDE(mavlink_message_t *msg, uint16_t len, uint8_t *buf, uint16_t PitchOut,
uint16_t RollOut) {
    mavlink_msg_rc_channels_override_pack(255, 0, msg, 1, 0, RollOut, PitchOut, 0, 0, 0, 0, 0,
0);
    len = mavlink_msg_to_send_buffer(buf, msg);
    Serial.write(buf, len);
    /*Serial.print("\n\rPitch: ");
    Serial.print(PitchOut);
    Serial.print(",");
    Serial.print(" Roll: ");
    Serial.print(RollOut);*/
}
/*mavlink_msg_rc_channels_override_pack(uint8_t system_id, uint8_t component_id,
mavlink_message_t* msg,
uint8_t target_system, uint8_t target_component, uint16_t chan1_raw, uint16_t chan2_raw,
uint16_t chan3_raw,
uint16_t chan4_raw, uint16_t chan5_raw, uint16_t chan6_raw, uint16_t chan7_raw,
uint16_t chan8_raw)*/
/*Channel 1 = Roll
Channel 2 = Pitch
Channel 3 = Throttle
Channel 4 = Yaw*/
/*Sensor0 = หน้า
Sensor1 = ขวา
Sensor2 = หลัง
Sensor3 = ซ้าย*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

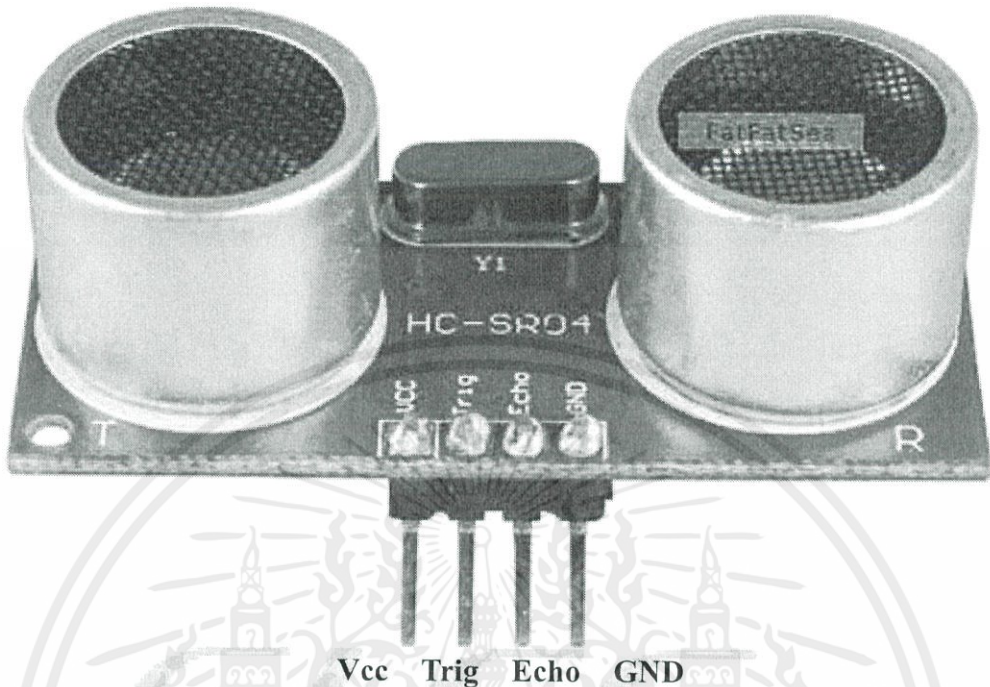
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

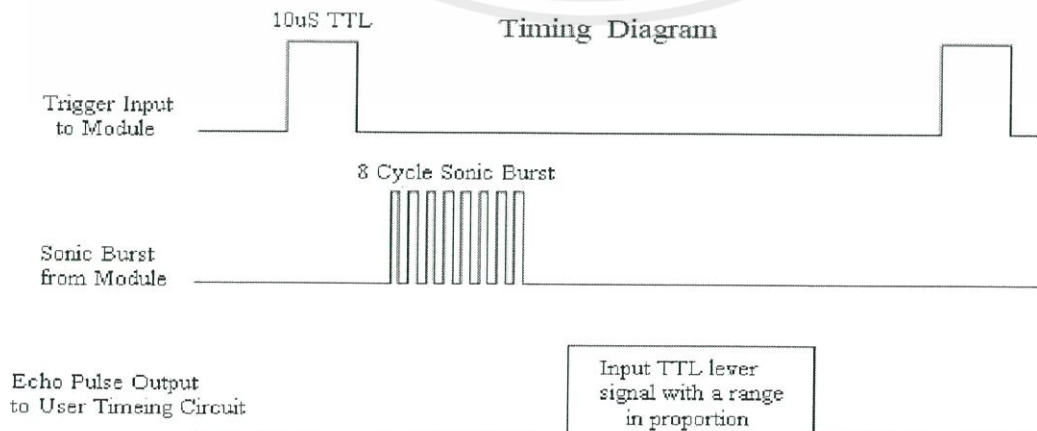
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.Elecfreaks.com



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

nRF24L01+

Single Chip 2.4GHz Transceiver

Preliminary Product Specification v1.0

Key Features

- Worldwide 2.4GHz ISM band operation
- 250kbps, 1Mbps and 2Mbps on air data rates
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 13.5mA RX at 2Mbps air data rate
- 900nA in power down
- 26µA in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with nRF24L01
- On-air compatible in 250kbps and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
- Low cost BOM
- ±60ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

Applications

- Wireless PC Peripherals
- Mouse, keyboards and remotes
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Game controllers
- Sports watches and sensors
- RF remote controls for consumer electronics
- Home and commercial automation
- Ultra low power sensor networks
- Active RFID
- Asset tracking systems
- Toys

All rights reserved.

Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

March 2008

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

All application information is advisory and does not form part of the specification.

Limiting values

Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the specifications are not implied. Exposure to limiting values for extended periods may affect device reliability.

Life support applications

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Data sheet status	
Objective product specification	This product specification contains target specifications for product development.
Preliminary product specification	This product specification contains preliminary data; supplementary data may be published from Nordic Semiconductor ASA later.
Product specification	This product specification contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Contact details

Visit www.nordicsemi.no for Nordic Semiconductor sales offices and distributors worldwide

Main office:

Otto Niensens vei 12
7004 Trondheim
Phone: +47 72 89 89 00
Fax: +47 72 89 89 89
www.nordicsemi.no



Writing Conventions

This product specification follows a set of typographic rules that makes the document consistent and easy to read. The following writing conventions are used:

- Commands, bit state conditions, and register names are written in `Courier`.
- Pin names and pin signal conditions are written in `Courier bold`.
- Cross references are underlined and highlighted in blue.

Revision History

Date	Version	Description
March 2008	1.0	

Attention!

Observe precaution for handling
Electrostatic Sensitive Device.



Contents

1	Introduction	7
1.1	Features	8
1.2	Block diagram	9
2	Pin Information	10
2.1	Pin assignment.....	10
2.2	Pin functions.....	11
3	Absolute maximum ratings	12
4	Operating conditions	13
5	Electrical specifications	14
5.1	Power consumption.....	14
5.2	General RF conditions	15
5.3	Transmitter operation	15
5.4	Receiver operation	16
5.5	Crystal specifications	19
5.6	DC characteristics	19
5.7	Power on reset.....	19
6	Radio Control	20
6.1	Operational Modes.....	20
6.1.1	State diagram	20
6.1.2	Power Down Mode	21
6.1.3	Standby Modes.....	21
6.1.4	RX mode.....	22
6.1.5	TX mode	22
6.1.6	Operational modes configuration.....	23
6.1.7	Timing Information.....	23
6.2	Air data rate.....	24
6.3	RF channel frequency	24
6.4	Received Power Detector measurements.....	24
6.5	PA control.....	25
6.6	RX/TX control.....	25
7	Enhanced ShockBurst™	26
7.1	Features	26
7.2	Enhanced ShockBurst™ overview	26
7.3	Enhanced Shockburst™ packet format.....	27
7.3.1	Preamble	27
7.3.2	Address	27
7.3.3	Packet Control Field	27
7.3.4	Payload.....	28
7.3.5	CRC (Cyclic Redundancy Check)	28
7.4	Automatic packet handling	28
7.4.1	Static and Dynamic Payload Length.....	29
7.4.2	Automatic packet assembly	29
7.4.3	Automatic packet validation	30
7.4.4	Automatic packet disassembly	30

7.5	Automatic packet transaction handling	31
7.5.1	Auto Acknowledgement.....	31
7.5.2	Auto Retransmission (ART).....	31
7.6	Enhanced ShockBurst flowcharts	33
7.6.1	PTX operation.....	33
7.6.2	PRX operation	35
7.7	MultiCeiver™	37
7.8	Enhanced ShockBurst™ timing	40
7.9	Enhanced ShockBurst™ transaction diagram	42
7.9.1	Single transaction with ACK packet and interrupts.....	42
7.9.2	Single transaction with a lost packet	43
7.9.3	Single transaction with a lost ACK packet	43
7.9.4	Single transaction with ACK payload packet.....	44
7.9.5	Single transaction with ACK payload packet and lost packet.....	44
7.9.6	Two transactions with ACK payload packet and the first ACK packet lost.....	45
7.9.7	Two transactions where max retransmissions is reached	45
7.10	Compatibility with ShockBurst™	46
7.10.1	ShockBurst™ packet format.....	46
8	Data and Control Interface	47
8.1	Features	47
8.2	Functional description	47
8.3	SPI operation	47
8.3.1	SPI Commands	47
8.3.2	SPI timing	49
8.4	Data FIFO	52
8.5	Interrupt.....	53
9	Register Map.....	54
9.1	Register map table	54
10	Peripheral RF Information	61
10.1	Antenna output.....	61
10.2	Crystal oscillator.....	61
10.3	nRF24L01+ crystal sharing with an MCU.....	61
10.3.1	Crystal parameters	61
10.3.2	Input crystal amplitude and current consumption	61
10.4	PCB layout and decoupling guidelines.....	62
11	Application example	63
11.1	PCB layout examples.....	64
12	Mechanical specifications	68
13	Ordering information	70
13.1	Package marking	70
13.2	Abbreviations	70
13.3	Product options	70
13.3.1	RF silicon.....	70
13.3.2	Development tools.....	70
14	Glossary of Terms.....	71

Appendix A - Enhanced ShockBurst™ - Configuration and Communication Example	72
Enhanced ShockBurst™ Transmitting Payload	72
Enhanced ShockBurst™ Receive Payload	73
Appendix B - Configuration for compatibility with nRF24XX.....	74
Appendix C - Constant carrier wave output for testing.....	75
Configuration	75



1 Introduction

The nRF24L01+ is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), suitable for ultra low power wireless applications. The nRF24L01+ is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz.

To design a radio system with the nRF24L01+, you simply need an MCU (microcontroller) and a few external passive components.

You can operate and configure the nRF24L01+ through a Serial Peripheral Interface (SPI). The register map, which is accessible through the SPI, contains all configuration registers in the nRF24L01+ and is accessible in all operation modes of the chip.

The embedded baseband protocol engine (Enhanced ShockBurst™) is based on packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Internal FIFOs ensure a smooth data flow between the radio front end and the system's MCU. Enhanced ShockBurst™ reduces system cost by handling all the high speed link layer operations.

The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power and air data rate. nRF24L01+ supports an air data rate of 250 kbps, 1 Mbps and 2Mbps. The high air data rate combined with two power saving modes make the nRF24L01+ very suitable for ultra low power designs.

nRF24L01+ is drop-in compatible with nRF24L01 and on-air compatible with nRF2401A, nRF2402, nRF24E1 and nRF24E2. Intermodulation and wideband blocking values in nRF24L01+ are much improved in comparison to the nRF24L01 and the addition of internal filtering to nRF24L01+ has improved the margins for meeting RF regulatory standards.

Internal voltage regulators ensure a high Power Supply Rejection Ratio (PSRR) and a wide power supply range.

1.1 Features

Features of the nRF24L01+ include:

- Radio
 - ▶ Worldwide 2.4GHz ISM band operation
 - ▶ 126 RF channels
 - ▶ Common RX and TX interface
 - ▶ GFSK modulation
 - ▶ 250kbps, 1 and 2Mbps air data rate
 - ▶ 1MHz non-overlapping channel spacing at 1Mbps
 - ▶ 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
 - ▶ Programmable output power: 0, -6, -12 or -18dBm
 - ▶ 11.3mA at 0dBm output power
- Receiver
 - ▶ Fast AGC for improved dynamic range
 - ▶ Integrated channel filters
 - ▶ 13.5mA at 2Mbps
 - ▶ -82dBm sensitivity at 2Mbps
 - ▶ -85dBm sensitivity at 1Mbps
 - ▶ -94dBm sensitivity at 250kbps
- RF Synthesizer
 - ▶ Fully integrated synthesizer
 - ▶ No external loop filter, VCO varactor diode or resonator
 - ▶ Accepts low cost ± 60 ppm 16MHz crystal
- Enhanced ShockBurst™
 - ▶ 1 to 32 bytes dynamic payload length
 - ▶ Automatic packet handling
 - ▶ Auto packet transaction handling
 - ▶ 6 data pipe MultiCeiver™ for 1:6 star networks
- Power Management
 - ▶ Integrated voltage regulator
 - ▶ 1.9 to 3.6V supply range
 - ▶ Idle modes with fast start-up times for advanced power management
 - ▶ 26 μ A Standby-I mode, 900nA power down mode
 - ▶ Max 1.5ms start-up from power down mode
 - ▶ Max 130us start-up from standby-I mode
- Host Interface
 - ▶ 4-pin hardware SPI
 - ▶ Max 10Mbps
 - ▶ 3 separate 32 bytes TX and RX FIFOs
 - ▶ 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

1.2 Block diagram

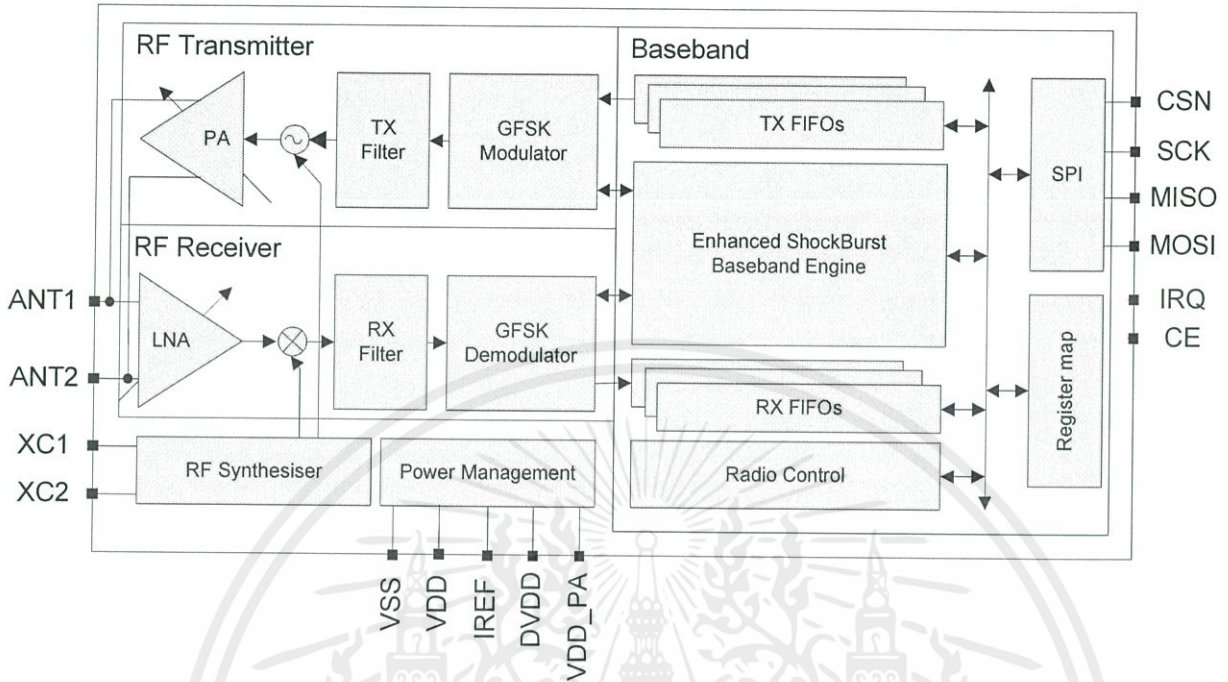


Figure 1. nRF24L01+ block diagram

2 Pin Information

2.1 Pin assignment

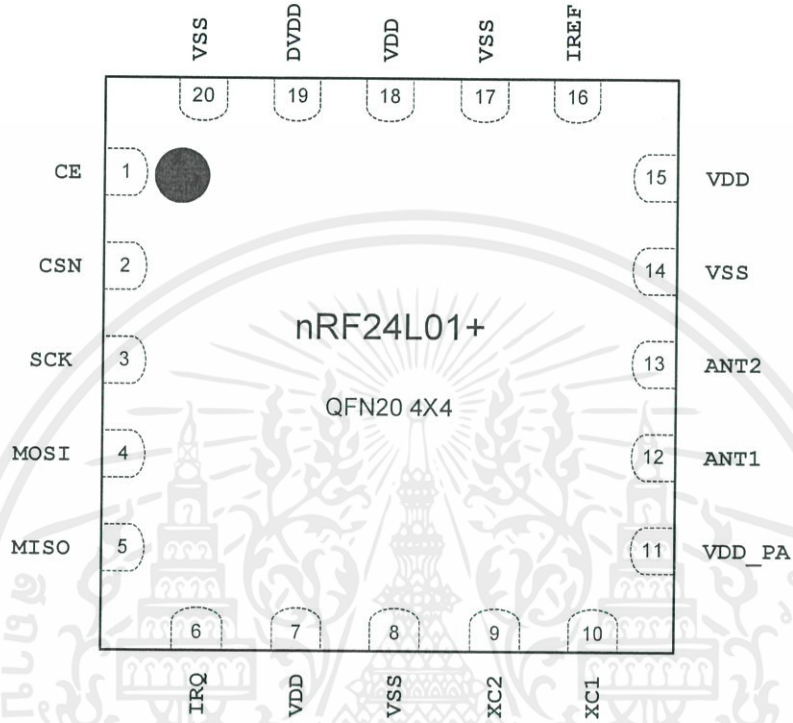


Figure 2. nRF24L01+ pin assignment (top view) for the QFN20 4x4 package

2.2 Pin functions

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin. Active low
7	VDD	Power	Power Supply (+1.9V - +3.6V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply Output (+1.8V) for the internal nRF24L01+ Power Amplifier. Must be connected to ANT1 and ANT2 as shown in Figure 29 .
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+1.9V - +3.6V DC)
16	IREF	Analog Input	Reference current. Connect a 22kΩ resistor to ground. See Figure 29 .
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+1.9V - +3.6V DC)
19	DVDD	Power Output	Internal digital supply output for de-coupling purposes. See Figure 29 .
20	VSS	Power	Ground (0V)

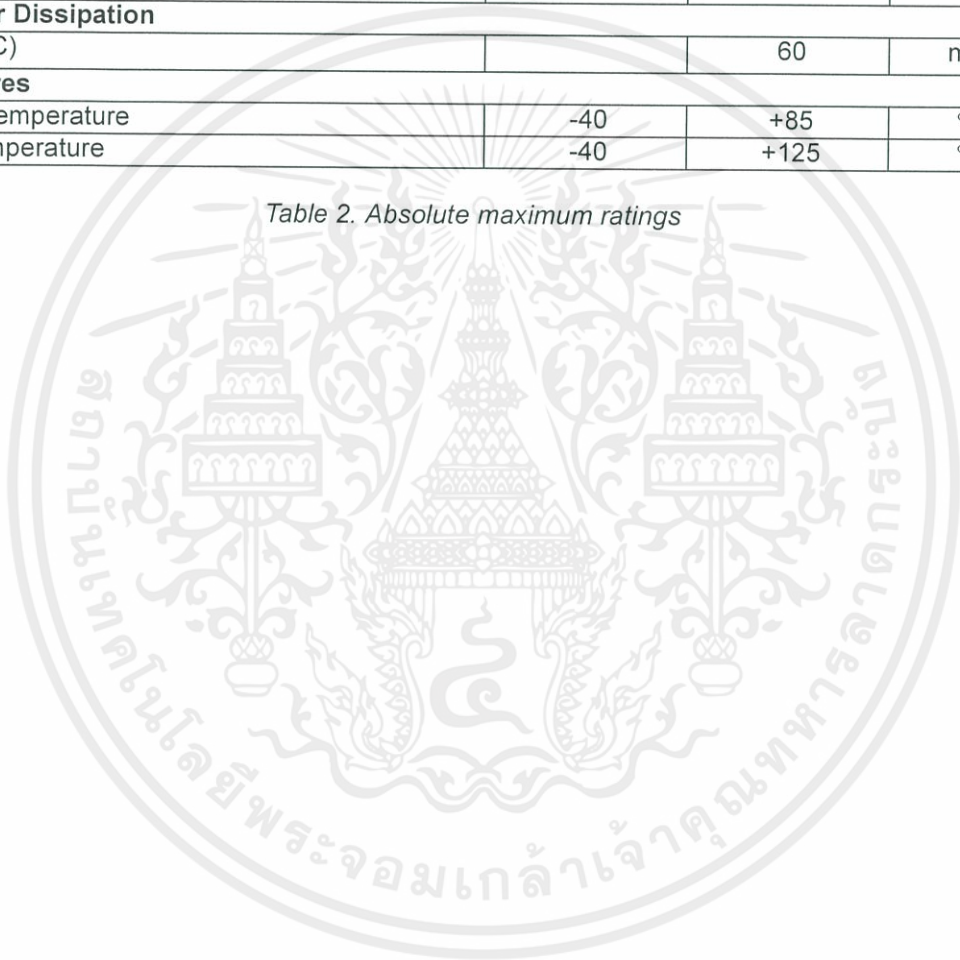
Table 1. nRF24L01+ pin function

3 Absolute maximum ratings

Note: Exceeding one or more of the limiting values may cause permanent damage to nRF24L01+.

Operating conditions	Minimum	Maximum	Units
Supply voltages			
VDD	-0.3	3.6	V
VSS		0	V
Input voltage			
V _I	-0.3	5.25	V
Output voltage			
V _O	VSS to VDD	VSS to VDD	
Total Power Dissipation			
P _D (T _A =85°C)		60	mW
Temperatures			
Operating Temperature	-40	+85	°C
Storage Temperature	-40	+125	°C

Table 2. Absolute maximum ratings



4 Operating conditions

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
VDD	Supply voltage		1.9	3.0	3.6	V
VDD	Supply voltage if input signals >3.6V		2.7	3.0	3.3	V
TEMP	Operating Temperature		-40	+27	+85	°C

Table 3. Operating conditions



5 Electrical specifications

Conditions: $V_{DD} = +3V$, $V_{SS} = 0V$, $T_A = -40^{\circ}C$ to $+85^{\circ}C$

5.1 Power consumption

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
Idle modes						
I_{VDD_PD}	Supply current in power down			900		nA
I_{VDD_ST1}	Supply current in standby-I mode	a		26		μA
I_{VDD_ST2}	Supply current in standby-II mode			320		μA
I_{VDD_SU}	Average current during 1.5ms crystal oscillator startup			400		μA
Transmit						
I_{VDD_TX0}	Supply current @ 0dBm output power	b		11.3		mA
I_{VDD_TX6}	Supply current @ -6dBm output power	b		9.0		mA
I_{VDD_TX12}	Supply current @ -12dBm output power	b		7.5		mA
I_{VDD_TX18}	Supply current @ -18dBm output power	b		7.0		mA
I_{VDD_AVG}	Average Supply current @ -6dBm output power, ShockBurst™	c		0.12		mA
I_{VDD_TXS}	Average current during TX settling	d		8.0		mA
Receive						
I_{VDD_2M}	Supply current 2Mbps			13.5		mA
I_{VDD_1M}	Supply current 1Mbps			13.1		mA
I_{VDD_250}	Supply current 250kbps			12.6		mA
I_{VDD_RXS}	Average current during RX settling	e		8.9		mA

- a. This current is for a 12pF crystal. Current when using external clock is dependent on signal swing.
- b. Antenna load impedance = $15\Omega + j88\Omega$.
- c. Antenna load impedance = $15\Omega + j88\Omega$. Average data rate 10kbps and max. payload length packets.
- d. Average current consumption during TX startup (130 μs) and when changing mode from RX to TX (130 μs).
- e. Average current consumption during RX startup (130 μs) and when changing mode from TX to RX (130 μs).

Table 4. Power consumption

5.2 General RF conditions

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
f_{OP}	Operating frequency	a	2400		2525	MHz
PLL_{res}	PLL Programming resolution			1		MHz
f_{XTAL}	Crystal frequency			16		MHz
Δf_{250}	Frequency deviation @ 250kbps			± 160		kHz
Δf_{1M}	Frequency deviation @ 1Mbps			± 160		kHz
Δf_{2M}	Frequency deviation @ 2Mbps			± 320		kHz
R_{GFSK}	Air Data rate	b	250		2000	kbps
$F_{CHANNEL\ 1M}$	Non-overlapping channel spacing @ 250kbps/1Mbps	c		1		MHz
$F_{CHANNEL\ 2M}$	Non-overlapping channel spacing @ 2Mbps	c		2		MHz

- a. Regulatory standards determine the band range you can use.
 b. Data rate in each burst on-air
 c. The minimum channel spacing is 1MHz

Table 5. General RF conditions

5.3 Transmitter operation

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
P_{RF}	Maximum Output Power	a		0	+4	dBm
P_{RFC}	RF Power Control Range		16	18	20	dB
P_{RFCR}	RF Power Accuracy				± 4	dB
P_{BW2}	20dB Bandwidth for Modulated Carrier (2Mbps)			1800	2000	kHz
P_{BW1}	20dB Bandwidth for Modulated Carrier (1Mbps)			900	1000	kHz
P_{BW250}	20dB Bandwidth for Modulated Carrier (250kbps)			700	800	kHz
$P_{RF1.2}$	1 st Adjacent Channel Transmit Power 2MHz (2Mbps)				-20	dBc
$P_{RF2.2}$	2 nd Adjacent Channel Transmit Power 4MHz (2Mbps)				-50	dBc
$P_{RF1.1}$	1 st Adjacent Channel Transmit Power 1MHz (1Mbps)				-20	dBc
$P_{RF2.1}$	2 nd Adjacent Channel Transmit Power 2MHz (1Mbps)				-45	dBc
$P_{RF1.250}$	1 st Adjacent Channel Transmit Power 1MHz (250kbps)				-30	dBc
$P_{RF2.250}$	2 nd Adjacent Channel Transmit Power 2MHz (250kbps)				-45	dBc

- a. Antenna load impedance = $15\Omega + j88\Omega$

Table 6. Transmitter operation

5.4 Receiver operation

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
	RX_{max}	Maximum received signal at <0.1% BER			0		dBm
2Mbps	RX_{SENS}	Sensitivity (0.1%BER) @2Mbps			-82		dBm
1Mbps	RX_{SENS}	Sensitivity (0.1%BER) @1Mbps			-85		dBm
250kbps	RX_{SENS}	Sensitivity (0.1%BER) @250kbps			-94		dBm

Table 7. RX Sensitivity

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	C/I_{CO}	C/I Co-channel			7		dBc
	C/I_{1ST}	1 st ACS (Adjacent Channel Selectivity) C/I 2MHz			3		dBc
	C/I_{2ND}	2 nd ACS C/I 4MHz			-17		dBc
	C/I_{3RD}	3 rd ACS C/I 6MHz			-21		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 12\text{MHz}$			-40		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 36\text{MHz}$	a		-48		dBc
1Mbps	C/I_{CO}	C/I Co-channel			9		dBc
	C/I_{1ST}	1 st ACS C/I 1MHz			8		dBc
	C/I_{2ND}	2 nd ACS C/I 2MHz			-20		dBc
	C/I_{3RD}	3 rd ACS C/I 3MHz			-30		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 6\text{MHz}$			-40		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 25\text{MHz}$	a		-47		dBc
250kbps	C/I_{CO}	C/I Co-channel			12		dBc
	C/I_{1ST}	1 st ACS C/I 1MHz			-12		dBc
	C/I_{2ND}	2 nd ACS C/I 2MHz			-33		dBc
	C/I_{3RD}	3 rd ACS C/I 3MHz			-38		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 6\text{MHz}$			-50		dBc
	C/I_{Nth}	N th ACS C/I, $f_i > 25\text{MHz}$	a		-60		dBc

a. **Narrow Band (In Band) Blocking measurements:**

0 to $\pm 40\text{MHz}$; 1MHz step size

For Interferer frequency offsets $n \cdot 2 \cdot f_{xtal}$, blocking performance is degraded by approximately 5dB compared to adjacent figures.

Table 8. RX selectivity according to ETSI EN 300 440-1 V1.3.1 (2001-09) page 27

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	C/I _{CO}	C/I Co-channel (Modulated carrier)			11		dBc
	C/I _{1ST}	1 st ACS C/I 2MHz			4		dBc
	C/I _{2ND}	2 nd ACS C/I 4MHz			-18		dBc
	C/I _{3RD}	3 rd ACS C/I 6MHz			-24		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 12\text{MHz}$			-40		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 36\text{MHz}$	a		-48		dBc
1Mbps	C/I _{CO}	C/I Co-channel			12		dBc
	C/I _{1ST}	1 st ACS C/I 1MHz			8		dBc
	C/I _{2ND}	2 nd ACS C/I 2MHz			-21		dBc
	C/I _{3RD}	3 rd ACS C/I 3MHz			-30		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 6\text{MHz}$			-40		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 25\text{MHz}$	a		-50		dBc
250kbps	C/I _{CO}	C/I Co-channel			7		dBc
	C/I _{1ST}	1 st ACS C/I 1MHz			-12		dBc
	C/I _{2ND}	2 nd ACS C/I 2MHz			-34		dBc
	C/I _{3RD}	3 rd ACS C/I 3MHz			-39		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 6\text{MHz}$			-50		dBc
	C/I _{Nth}	N th ACS C/I, $f_i > 25\text{MHz}$	a		-60		dBc

a. **Narrow Band (In Band) Blocking measurements:**

0 to $\pm 40\text{MHz}$; 1MHz step size

Wide Band Blocking measurements:

30MHz to 2000MHz; 10MHz step size

2000MHz to 2399MHz; 3MHz step size

2484MHz to 3000MHz; 3MHz step size

3GHz to 12.75GHz; 25MHz step size

Wanted signal for wideband blocking measurements:

-67dBm in 1Mbps and 2Mbps mode

-77dBm in 250kbps mode

For Interferer frequency offsets $n \cdot 2 \cdot f_{\text{xtal}}$, blocking performance are degraded by approximately 5dB compared to adjacent figures.

If the wanted signal is 3dB or more above the sensitivity level then, the carrier/interferer ratio is independent of the wanted signal level for a given frequency offset.

Table 9. RX selectivity with nRF24L01+ equal modulation on interfering signal. Measured using $P_{\text{in}} = -67\text{dBm}$ for wanted signal.

Datarate	Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
2Mbps	P_IM(6)	Input power of IM interferers at 6 and 12MHz offset from wanted signal			-42		dBm
	P_IM(8)	Input power of IM interferers at 8 and 16MHz offset from wanted signal			-38		dBm
	P_IM(10)	Input power of IM interferers at 10 and 20MHz offset from wanted signal			-37		dBm
1Mbps	P_IM(3)	Input power of IM interferers at 3 and 6MHz offset from wanted signal			-36		dBm
	P_IM(4)	Input power of IM interferers at 4 and 8MHz offset from wanted signal			-36		dBm
	P_IM(5)	Input power of IM interferers at 5 and 10MHz offset from wanted signal			-36		dBm
250kbps	P_IM(3)	Input power of IM interferers at 3 and 6MHz offset from wanted signal			-36		dBm
	P_IM(4)	Input power of IM interferers at 4 and 8MHz offset from wanted signal			-36		dBm
	P_IM(5)	Input power of IM interferers at 5 and 10MHz offset from wanted signal			-36		dBm

Note: Wanted signal level at Pin = -64 dBm. Two interferers with equal input power are used. The interferer closest in frequency is unmodulated, the other interferer is modulated equal with the wanted signal. The input power of interferers where the sensitivity equals BER = 0.1% is presented.

Table 10. RX intermodulation test performed according to Bluetooth Specification version 2.0

5.5 Crystal specifications

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
F _{xo}	Crystal Frequency			16		MHz
ΔF	Tolerance	a b			±60	ppm
C ₀	Equivalent parallel capacitance			1.5	7.0	pF
C _L	Load capacitance		8	12	16	pF
ESR	Equivalent Series Resistance				100	Ω

- a. Frequency accuracy including; tolerance at 25°C, temperature drift, aging and crystal loading.
 b. Frequency regulations in certain regions set tighter requirements for frequency tolerance (For example, Japan and South Korea specify max. +/- 50ppm)

Table 11. Crystal specifications

5.6 DC characteristics

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
V _{IH}	HIGH level input voltage		0.7V _{DD}		5.25 ^a	V
V _{IL}	LOW level input voltage		V _{SS}		0.3V _{DD}	V

- a. If the input signal >3.6V, the V_{DD} of the nRF24L01+ must be between 2.7V and 3.3V (3.0V±10%)

Table 12. Digital input pin

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
V _{OH}	HIGH level output voltage (I _{OH} =-0.25mA)		V _{DD} -0.3		V _{DD}	V
V _{OL}	LOW level output voltage (I _{OL} =0.25mA)				0.3	V

Table 13. Digital output pin

5.7 Power on reset

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
T _{PUP}	Power ramp up time	a			100	ms
T _{POR}	Power on reset	b	1		100	ms

- a. From 0V to 1.9V.
 b. Measured from when the V_{DD} reaches 1.9V to when the reset finishes.

Table 14. Power on reset

6 Radio Control

This chapter describes the nRF24L01+ radio transceiver's operating modes and the parameters used to control the radio.

The nRF24L01+ has a built-in state machine that controls the transitions between the chip's operating modes. The state machine takes input from user defined register values and internal signals.

6.1 Operational Modes

You can configure the nRF24L01+ in power down, standby, RX or TX mode. This section describes these modes in detail.

6.1.1 State diagram

The state diagram in [Figure 3](#), shows the operating modes and how they function. There are three types of distinct states highlighted in the state diagram:

- **Recommended operating mode:** is a recommended state used during normal operation.
- **Possible operating mode:** is a possible operating state, but is not used during normal operation.
- **Transition state:** is a time limited state used during start up of the oscillator and settling of the PLL.

When the V_{DD} reaches 1.9V or higher nRF24L01+ enters the Power on reset state where it remains in reset until entering the Power Down mode.

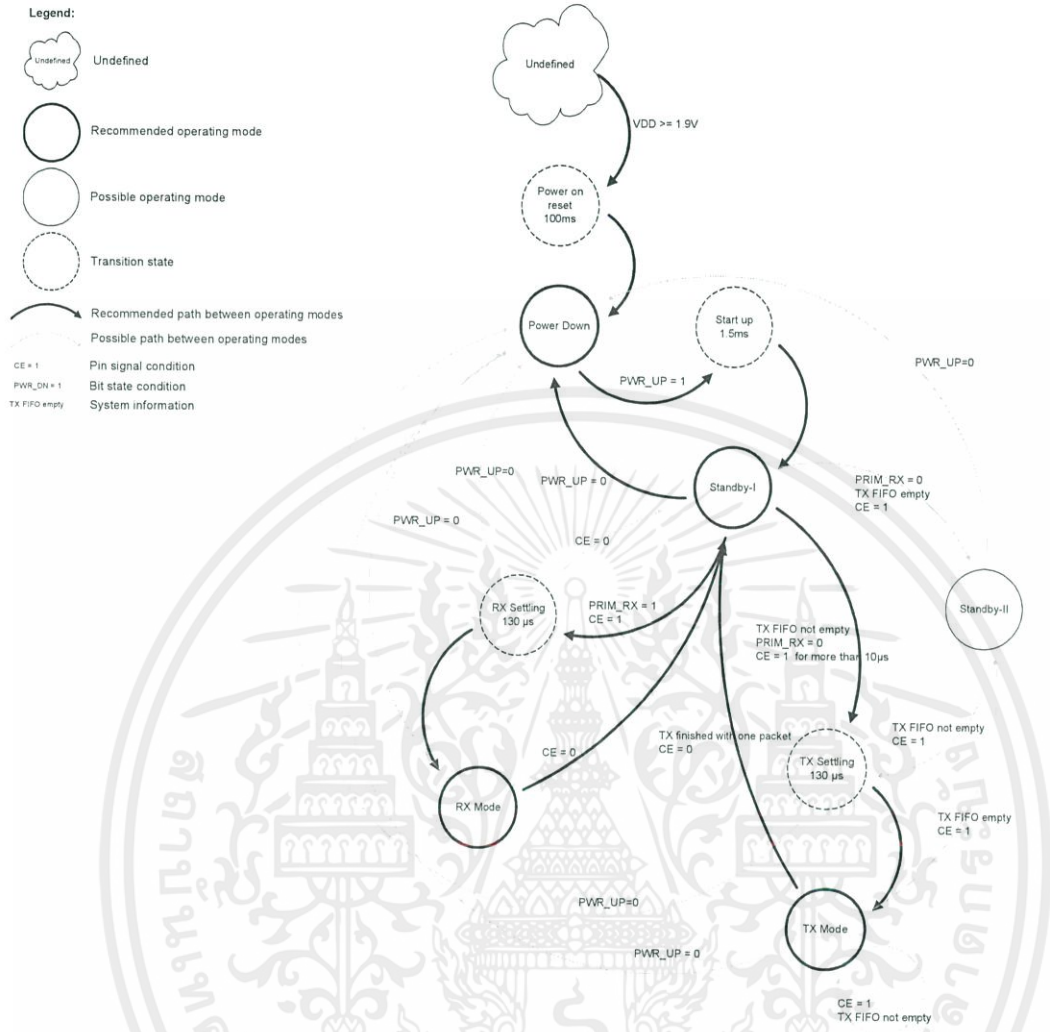


Figure 3. Radio control state diagram

6.1.2 Power Down Mode

In power down mode nRF24L01+ is disabled using minimal current consumption. All register values available are maintained and the SPI is kept active, enabling change of configuration and the uploading/downloading of data registers. For start up times see Table 16. on page 23. Power down mode is entered by setting the `PWR_UP` bit in the `CONFIG` register low.

6.1.3 Standby Modes

6.1.3.1 Standby-I mode

By setting the `PWR_UP` bit in the `CONFIG` register to 1, the device enters standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start up times. In this mode only part of the crystal oscillator is active. Change to active modes only happens if `CE` is set high and when `CE` is set low, the nRF24L01 returns to standby-I mode from both the TX and RX modes.

6.1.3.2 Standby-II mode

In standby-II mode extra clock buffers are active and more current is used compared to standby-I mode. nRF24L01+ enters standby-II mode if \overline{CE} is held high on a PTX device with an empty TX FIFO. If a new packet is uploaded to the TX FIFO, the PLL immediately starts and the packet is transmitted after the normal PLL settling delay (130 μ s).

Register values are maintained and the SPI can be activated during both standby modes. For start up times see [Table 16. on page 23](#).

6.1.4 RX mode

The RX mode is an active mode where the nRF24L01+ radio is used as a receiver. To enter this mode, the nRF24L01+ must have the PWR_UP bit, $PRIM_RX$ bit and the \overline{CE} pin set high.

In RX mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFOs. If the RX FIFOs are full, the received packet is discarded.

The nRF24L01+ remains in RX mode until the MCU configures it to standby-I mode or power down mode. However, if the automatic protocol features (Enhanced ShockBurst™) in the baseband protocol engine are enabled, the nRF24L01+ can enter other modes in order to execute the protocol.

In RX mode a Received Power Detector (RPD) signal is available. The RPD is a signal that is set high when a RF signal higher than -64 dBm is detected inside the receiving frequency channel. The internal RPD signal is filtered before presented to the RPD register. The RF signal must be present for at least 40 μ s before the RPD is set high. How to use the RPD is described in [Section 6.4 on page 24](#).

6.1.5 TX mode

The TX mode is an active mode for transmitting packets. To enter this mode, the nRF24L01+ must have the PWR_UP bit set high, $PRIM_RX$ bit set low, a payload in the TX FIFO and a high pulse on the \overline{CE} for more than 10 μ s.

The nRF24L01+ stays in TX mode until it finishes transmitting a packet. If $\overline{CE} = 0$, nRF24L01+ returns to standby-I mode. If $\overline{CE} = 1$, the status of the TX FIFO determines the next action. If the TX FIFO is not empty the nRF24L01+ remains in TX mode and transmits the next packet. If the TX FIFO is empty the nRF24L01+ goes into standby-II mode. The nRF24L01+ transmitter PLL operates in open loop when in TX mode. It is important never to keep the nRF24L01+ in TX mode for more than 4ms at a time. If the Enhanced ShockBurst™ features are enabled, nRF24L01+ is never in TX mode longer than 4ms.

6.1.6 Operational modes configuration

The following table (Table 15.) describes how to configure the operational modes.

Mode	PWR_UP register	PRIM_RX register	CE input pin	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFOs. Will empty all levels in TX FIFOs ^a .
TX mode	1	0	Minimum 10 μ s high pulse	Data in TX FIFOs. Will empty one level in TX FIFOs ^b .
Standby-II	1	0	1	TX FIFO empty.
Standby-I	1	-	0	No ongoing packet transmission.
Power Down	0	-	-	-

- If CE is held high all TX FIFOs are emptied and all necessary ACK and possible retransmits are carried out. The transmission continues as long as the TX FIFO is refilled. If the TX FIFO is empty when the CE is still high, nRF24L01+ enters standby-II mode. In this mode the transmission of a packet is started as soon as the CSN is set high after an upload (UL) of a packet to TX FIFO.
- This operating mode pulses the CE high for at least 10 μ s. This allows one packet to be transmitted. This is the normal operating mode. After the packet is transmitted, the nRF24L01+ enters standby-I mode.

Table 15. nRF24L01+ main modes

6.1.7 Timing Information

The timing information in this section relates to the transitions between modes and the timing for the CE pin. The transition from TX mode to RX mode or vice versa is the same as the transition from the standby modes to TX mode or RX mode (max. 130 μ s), as described in Table 16.

Name	nRF24L01+	Max.	Min.	Comments
Tpd2stby	Power Down → Standby mode	1.5ms		Internal crystal oscillator
Tpd2stby	Power Down → Standby mode	150 μ s		With external clock
Tstby2a	Standby modes → TX/RX mode	130 μ s		
Thce	Minimum CE high		10 μ s	
Tpece2csn	Delay from CE positive edge to CSN low		4 μ s	

Table 16. Operational timing of nRF24L01+

When nRF24L01+ is in power down mode it must settle for 1.5ms before it can enter the TX or RX modes. If an external clock is used this delay is reduced to 150 μ s, see Table 16.. The settling time must be controlled by the MCU.

Note: If VDD is turned off the register value is lost and you must configure nRF24L01+ before entering the TX or RX modes.

6.2 Air data rate

The air data rate is the modulated signaling rate the nRF24L01+ uses when transmitting and receiving data. It can be 250kbps, 1Mbps or 2Mbps. Using lower air data rate gives better receiver sensitivity than higher air data rate. But, high air data rate gives lower average current consumption and reduced probability of on-air collisions.

The air data rate is set by the `RF_DR` bit in the `RF_SETUP` register. A transmitter and a receiver must be programmed with the same air data rate to communicate with each other.

nRF24L01+ is fully compatible with nRF24L01. For compatibility with nRF2401A, nRF2402, nRF24E1, and nRF24E2 the air data rate must be set to 250kbps or 1Mbps.

6.3 RF channel frequency

The RF channel frequency determines the center of the channel used by the nRF24L01+. The channel occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps and a bandwidth of less than 2MHz at 2Mbps. nRF24L01+ can operate on frequencies from 2.400GHz to 2.525GHz. The programming resolution of the RF channel frequency setting is 1MHz.

At 2Mbps the channel occupies a bandwidth wider than the resolution of the RF channel frequency setting. To ensure non-overlapping channels in 2Mbps mode, the channel spacing must be 2MHz or more. At 1Mbps and 250kbps the channel bandwidth is the same or lower than the resolution of the RF frequency.

The RF channel frequency is set by the `RF_CH` register according to the following formula:

$$F_0 = 2400 + RF_CH [MHz]$$

You must program a transmitter and a receiver with the same RF channel frequency to communicate with each other.

6.4 Received Power Detector measurements

Received Power Detector (RPD), located in register 09, bit 0, triggers at received power levels above -64 dBm that are present in the RF channel you receive on. If the received power is less than -64 dBm, $RDP = 0$.

The RPD can be read out at any time while nRF24L01+ is in receive mode. This offers a snapshot of the current received power level in the channel. The RPD status is latched when a valid packet is received which then indicates signal strength from your own transmitter. If no packets are received the RPD is latched at the end of a receive period as a result of host MCU setting CE low or RX time out controlled by Enhanced ShockBurst™.

The status of RPD is correct when RX mode is enabled and after a wait time of $T_{stby2a} + T_{delay_AGC} = 130\mu s + 40\mu s$. The RX gain varies over temperature which means that the RPD threshold also varies over temperature. The RPD threshold value is reduced by -5dB at $T = -40^\circ C$ and increased by +5dB at $85^\circ C$.

6.5 PA control

The PA (Power Amplifier) control is used to set the output power from the nRF24L01+ power amplifier. In TX mode PA control has four programmable steps, see [Table 17](#).

The PA control is set by the `RF_PWR` bits in the `RF_SETUP` register.

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0dBm	11.3mA
10	-6dBm	9.0mA
01	-12dBm	7.5mA
00	-18dBm	7.0mA

Conditions: $V_{DD} = 3.0V$, $V_{SS} = 0V$, $T_A = 27^\circ C$, Load impedance = $15\Omega + j88\Omega$.

Table 17. RF output power setting for the nRF24L01+

6.6 RX/TX control

The RX/TX control is set by `PRIM_RX` bit in the `CONFIG` register and sets the nRF24L01+ in transmit/receive mode.

pixhawk

AUTOPILOT



QUICK START GUIDE

3D Robotics

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARTS



1 Pixhawk

2 Buzzer

3 Safety switch

4 Micro-SD card and adapter

5 Micro-USB cable

6 Six-wire cable x2

7 Power module

8 I²C splitter module

9 Four-position I²C splitter cable

10 Three-wire servo cable

11 Mounting foam

GETTING STARTED

With the help of APM firmware, Pixhawk turns any RC plane, copter, or rover into a full-featured personal drone. Once you have a fully-assembled frame, follow this guide to install Pixhawk.

1 Mount

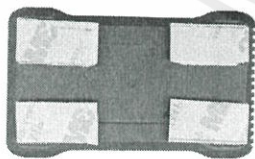
2 Connect

3 Load firmware

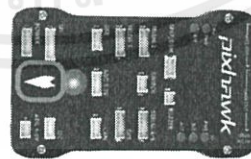
4 Calibrate

1 MOUNT

Use the provided foam to mount Pixhawk as close as possible to your vehicle's center of gravity. Make sure to orient the board with the arrow pointing forward.



Attach the foam squares to the corners of the board.

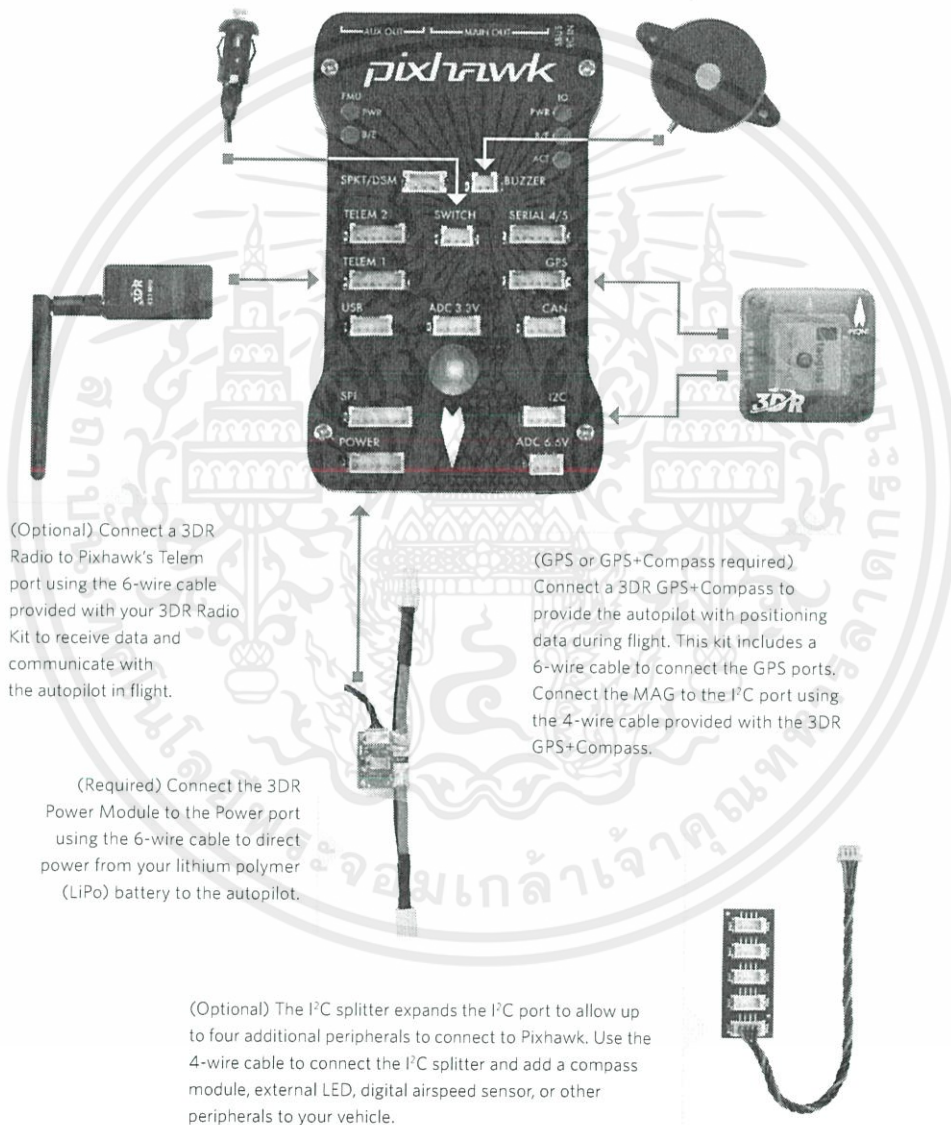


←
Vehicle front

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 CONNECT

(Required) Connect the buzzer and safety switch.



(Optional) Connect a 3DR Radio to Pixhawk's Telem port using the 6-wire cable provided with your 3DR Radio Kit to receive data and communicate with the autopilot in flight.

(GPS or GPS+Compass required)
Connect a 3DR GPS+Compass to provide the autopilot with positioning data during flight. This kit includes a 6-wire cable to connect the GPS ports. Connect the MAG to the I²C port using the 4-wire cable provided with the 3DR GPS+Compass.

(Required) Connect the 3DR Power Module to the Power port using the 6-wire cable to direct power from your lithium polymer (LiPo) battery to the autopilot.

(Optional) The I²C splitter expands the I²C port to allow up to four additional peripherals to connect to Pixhawk. Use the 4-wire cable to connect the I²C splitter and add a compass module, external LED, digital airspeed sensor, or other peripherals to your vehicle.

LOAD SD CARD INTO PIXHAWK

If the SD card is not preloaded into Pixhawk, insert the micro-SD card into the slot at the bottom end of the board.



CONNECT RADIO CONTROL

FOR PPM RC RECEIVERS AND FUTABA S.BUS RECEIVERS



Connect the ground (-), power (+), and signal (S) wires to the RC pins using the provided 3-wire servo cable.

FOR SPEKTRUM SATELLITE RECEIVERS



For a Spektrum DSM, DSM2, or DSM-X Satellite RC receiver, connect to the SPKT/DSM port.

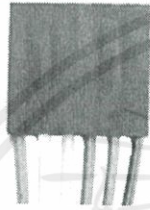
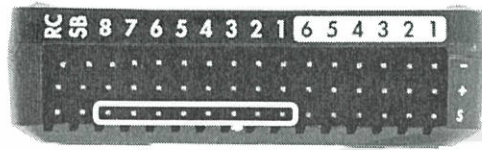
For a complete list of RC systems compatible with Pixhawk, [visit the APM wiki page here](#).

FOR PWM RECEIVERS

Purchase a PPM Encoder module to connect a PWM RC receiver to Pixhawk at [store.3dr.com](#).

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONNECT OUTPUTS



FOR COPTERS

Connect each signal wire from the PDB to the main output signal (S) pins by motor number. Connect one wire for each motor to the corresponding pin.

- | | |
|-----------------|-----------------|
| Pin 1 = Motor 1 | Pin 5 = Motor 5 |
| Pin 2 = Motor 2 | Pin 6 = Motor 6 |
| Pin 3 = Motor 3 | Pin 7 = Motor 7 |
| Pin 4 = Motor 4 | Pin 8 = Motor 8 |

FOR PLANES

For planes, connect the control channel wires to the main output signal pins.

- Pin 1 = Aileron
- Pin 2 = Elevator
- Pin 3 = Throttle
- Pin 4 = Rudder

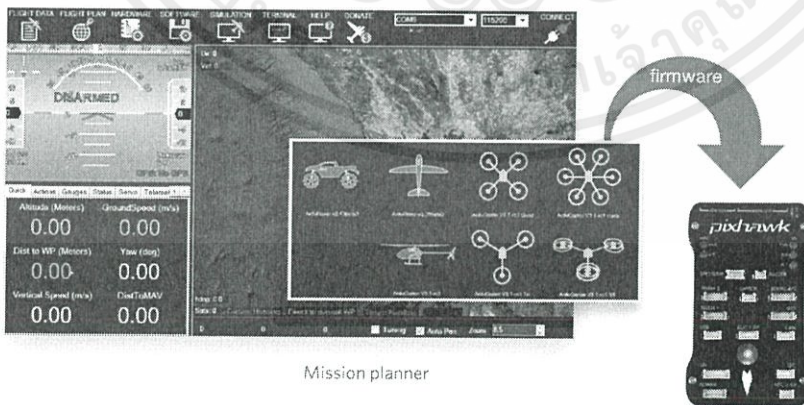
FOR ROVERS

For rovers, connect the throttle and steering wires to the main output signal pins.

- Pin 3 = Throttle
- Pin 4 = Steering

③ LOAD FIRMWARE

APM firmware is the brains of your autopilot operation and must be installed before using Pixhawk. To load firmware onto Pixhawk, install a mission planner application on your ground station computer. Choose either Mission Planner (Windows) or APM Planner for (Windows, OS X, and Linux). Both applications are available for free download from ardupilot.com.



Mission planner

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Download Mission Planner (Windows)

Ardupilot.com → Downloads → Mission Planner

Mission Planner « Downloads

Sort by: Title | Hits | **Date**

- MissionPlanner - Installer

Select the installer package to download.



Download APM Planner (Windows, OS X, and Linux)

Ardupilot.com → Downloads → APM Planner 2.0

APM Planner 2.0 « Downloads

Sort by: Title | Hits | **Date**

- APM Planner 2.0 Mac
- APM Planner 2.0 Windows
- APM Planner 2.0 Linux

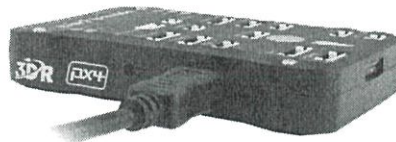
Select your platform to download.

INSTALL PLANNER

After selecting the correct file, read the safety information and select Download.

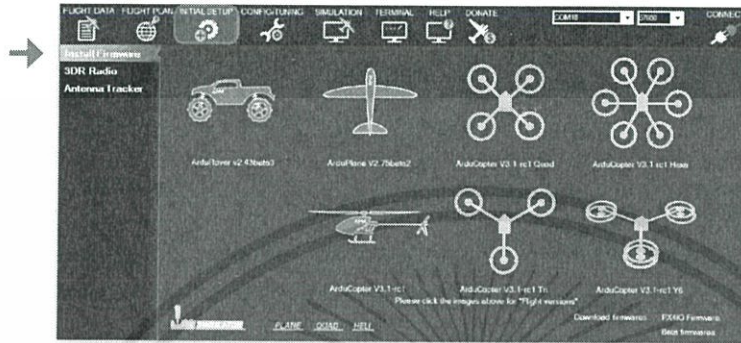
Open the file to run the setup wizard. Proceed through any security warnings, and install all suggested drivers. When the installation is complete, open the application, and connect Pixhawk to your computer using the micro-USB cable.

Your computer will automatically install the correct drivers. Do not select Connect at this time; Pixhawk can only load firmware while unconnected to Mavlink.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Select Initial Setup, Install Firmware, and select your vehicle.



When prompted, follow the directions to load the firmware. Once the status bar shows that the download is complete, power cycle the board by disconnecting and reconnecting the USB.

If you hear a musical tone, your firmware installation is complete. If you hear a series of tones followed by three beeps, disconnect the USB and reconnect while holding down the safety button. Upon restart, listen for a series of tones followed by two beeps indicating that your firmware has loaded successfully.

Everything is OK:
musical tone



Pixhawk needs your attention:
beep BEEP beep... BEEP BEEP BEEP

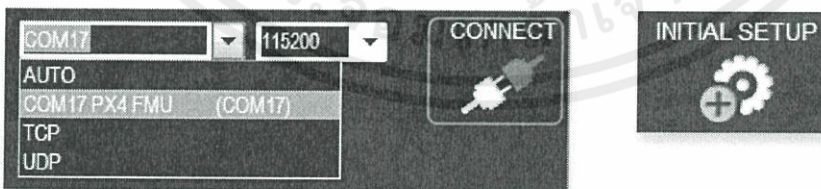


beep BEEP beep...
BEEP BEEP



4 CALIBRATE

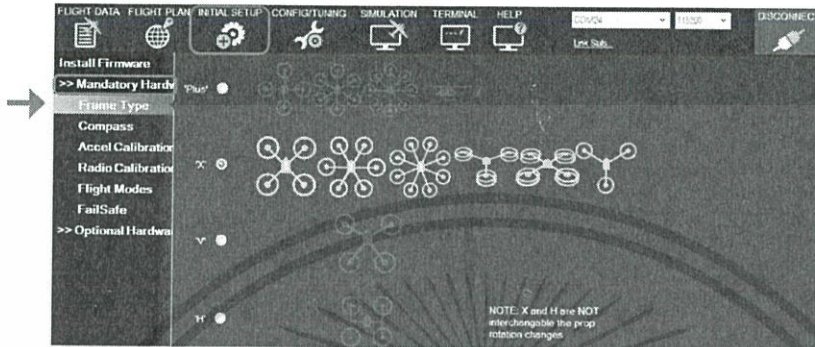
With Pixhawk connected to your computer, select the communication option from the drop-down menu for PX4 FMU, set the rate to 115200, and select the Connect icon. Select Initial Setup and Mandatory Hardware to access the calibration wizards.



Remove propellers before performing calibration.

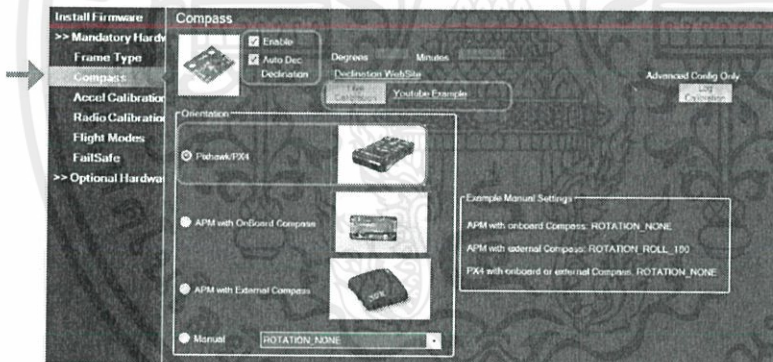
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT FRAME TYPE (COPTERS ONLY)



For copter, select your frame orientation.

CALIBRATE COMPASS



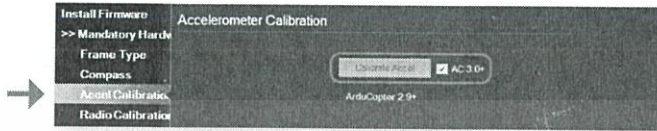
Select the options to enable the compass, to allow automatic declination calculation; and to specify Pixhawk. Select Live Calibration to launch the wizard, and follow the prompts.

Show Me

videos demonstrating live calibration techniques at 3dr.com/learn.

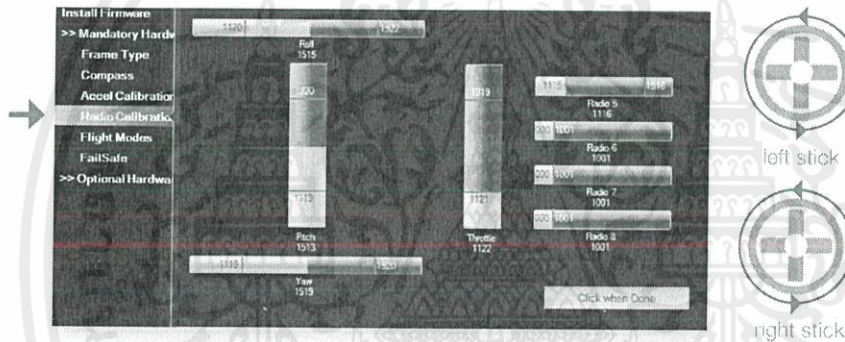
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALIBRATE ACCELEROMETER



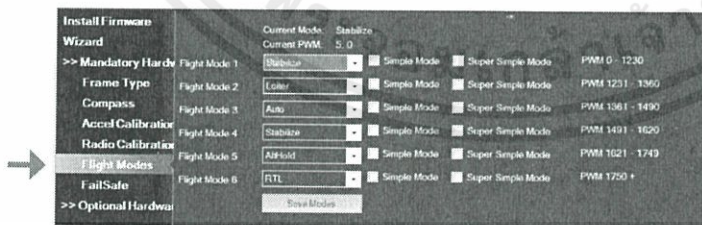
Select Accel Calibration, check the box for AC 3.0+, select Calibrate, and follow the prompts to calibrate Pixhawk's accelerometer. Make sure to wait a couple of seconds before and after changing the positions of the vehicle.

RC CALIBRATION



Select Radio Calibration to teach Pixhawk to work with your RC transmitter. Turn on your transmitter, select Calibrate Radio, and move all sticks and switches to their extreme positions. Select Click when Done once the red bars are set for all available channels.

SELECT FLIGHT MODES



Move each switch on your transmitter to its available positions. The mission planner will indicate the currently selected position with green highlighting. Select a mode for each switch position, and select Save Modes to assign.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED MEANINGS



Flashing red and blue: initializing. Please wait.



Double flashing yellow: error. System refuses to arm.



Flashing blue: disarmed, searching for GPS. Autonomous, loiter, and return-to-launch modes require GPS lock.



Flashing green: disarmed, GPS lock acquired. Ready to arm. Quick double tone when disarming from the armed state.



Solid green plus single long tone: armed and ready to fly!



Flashing yellow: RC failsafe activated.



Flashing yellow plus quick repeating tone: battery failsafe activated.



Flashing yellow and blue plus high-high-high-low tone: GPS glitch or GPS failsafe activated.



SAFETY SWITCH MEANINGS



Quick, constant blinking: performing system check. Please wait.



Intermittent blinking: system ready. Press the safety button to activate.



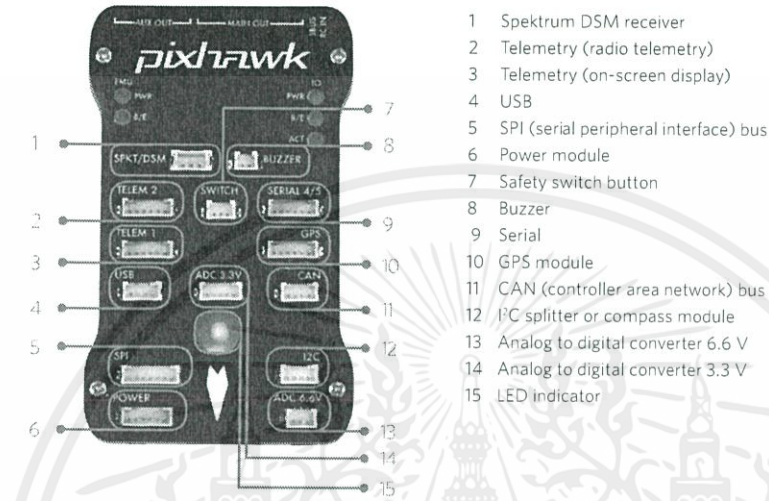
Solid: ready to arm. Proceed to the arming procedure.

Learn more

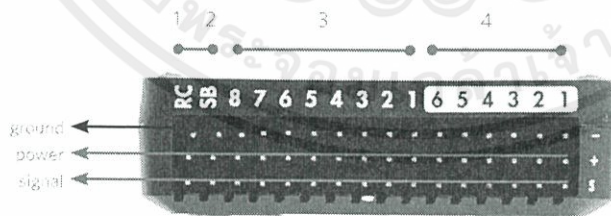
about LED meanings and buzzer tones at 3dr.com/learn.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORTS



- 1 Input/output reset button
- 2 SD card
- 3 Flight management reset button
- 4 Micro-USB port



- 1 Radio control receiver input
- 2 S.Bus output
- 3 Main outputs
- 4 Auxiliary outputs

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTE

Please note that these instructions describe basic setup for Pixhawk and do not represent the complete set of configuration procedures required to build a copter, plane, or rover.

For more information on ESC calibration, battery monitoring, failsafes, mode descriptions, and more, visit ardupilot.com. Do not operate your vehicle without a complete understanding of the online instructions.

SPECIFICATIONS

Processor

32-bit ARM Cortex M4 core with FPU
168 Mhz/256 KB RAM/2 MB Flash
32-bit failsafe co-processor

Sensors

ST Micro 16-bit gyroscope
ST Micro 14-bit accelerometer/magnetometer
MEAS barometer
MPU6000 accelerometer/magnetometer

Power

Ideal diode controller with automatic failover
Servo rail high-power (7 V) and high-current ready
All peripheral outputs over-current protected, all inputs ESC protected

Interfaces

5x UART serial ports, 1 high-power capable, 2x with HW flow control
Spektrum DSM/DSM2/DSM-X Satellite input
Futaba S.BUS input and output
PPM sum signal
RSSI (PWM or voltage) input
I²C, SPI, 2x CAN, USB
3.3 and 6.6 ADC inputs

Dimensions

Weight 38 g (1.3 oz)
Width 50 mm (2.0")
Height 15.5 mm (.6")
Length 81.5 mm (3.2")

SUPPORT

For more information about Pixhawk and other documentation, visit 3dr.com/learn. For more instruction on using APM firmware and planner software, visit ardupilot.com

For customer support, contact us at help@3dr.com or call our support line at +1 (858) 225-1414 Monday through Friday, 8 am to 5 pm, PST.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SAFETY

Operating a powered vehicle of any kind can be a lot of fun, but it carries certain inherent risks. Regulations governing the use of powered vehicles, including aircraft, vary from locale to locale, even within the same country or district. It is your responsibility to ensure that you understand and comply with all local laws and regulations.

Safety basics:

- Never operate the vehicle or software in a way that could be dangerous to you, other people, or property.
- Always keep propeller arcs free of objects and body parts while the vehicle is live.
- Keep in mind that software and hardware failures happen. Although we design our products to minimize such issues, you should always operate with the understanding that a failure could occur at any time and without warning. Accordingly, you should take the appropriate precautions to minimize danger in case of product failure.
- Never use the software or hardware for manned vehicles.
- Always operate within local laws and regulations.
- Do not operate the aircraft if you are under the age of 18.

Additional safety information:

- Be sure to maintain safe distances between people and your aircraft.
- Never operate your aircraft if your ability to do so with the utmost attention to safety is impaired in any way. Do not operate your aircraft while tired, under the influence of drugs or alcohol, or otherwise unable to operate it with the highest attention to safety.
- Environment conditions can change rapidly and can make operation difficult. If this occurs, land your aircraft and discontinue use immediately. Do not operate your aircraft if operating conditions are not ideal. This includes, but is not limited to, rain, snow or excessive wind.
- Always ensure the battery cable is disconnected from the aircraft until you are ready to fly, and ensure that your batteries are fully charged prior to use.
- Always turn on the transmitter and ensure the throttle stick is all the way down before connecting the battery.
- After landing, disarm your vehicle immediately and disconnect the battery cable.
- Do not turn off the transmitter until after you have disconnected the battery.
- Always remove the propellers while testing the motors.
- When the battery is connected, always assume the vehicle is live and the motors are armed.
- Do not attempt to fly longer than the battery's safe capacity.
- Do not operate the vehicle with excess weight attached.
- Ensure that all vehicle components are well maintained before each flight. Ensure that components are firmly attached and operating properly.
- Replace any worn or damaged components before each flight. Never operate with any damaged or worn components.
- SAFETY IS THE FIRST PRIORITY. Take all precautions necessary to ensure your own safety as well as the safety of other people and property.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISCLAIMER

THE LIMITED WARRANTIES APPLICABLE TO 3D ROBOTICS-BRANDED HARDWARE CAN BE FOUND AT WWW.3DROBOTICS.COM/TERMS/ AND CAN BE REFERENCED IN FULL AT THAT URL. 3D ROBOTICS RESERVES THE RIGHT TO UPDATE THE WARRANTIES AT ANY TIME WITHOUT EXPRESS NOTICE. 3D ROBOTICS MAKES NO OTHER WARRANTIES FOR 3D ROBOTICS-BRANDED PRODUCTS, AND MAKES NO WARRANTIES WHATSOEVER FOR SERVICE, SOFTWARE, MAINTENANCE OR SUPPORT FOR NON-3D ROBOTICS-BRANDED PRODUCTS. SUCH PRODUCTS, SOFTWARE, SERVICES, MAINTENANCE OR SUPPORT IS PROVIDED BY 3D ROBOTICS "AS IS" AND ANY THIRD-PARTY WARRANTIES, PRODUCTS, SOFTWARE, SERVICES, MAINTENANCE OR SUPPORT ARE PROVIDED BY THE ORIGINAL MANUFACTURER OR SUPPLIER, NOT BY 3D ROBOTICS. 3D ROBOTICS MAKES NO EXPRESS WARRANTIES EXCEPT THOSE STATED IN 3D ROBOTICS' APPLICABLE WARRANTY IN EFFECT ON THE DATE OF THE INVOICE, PACKING SLIP OR ACKNOWLEDGEMENT.

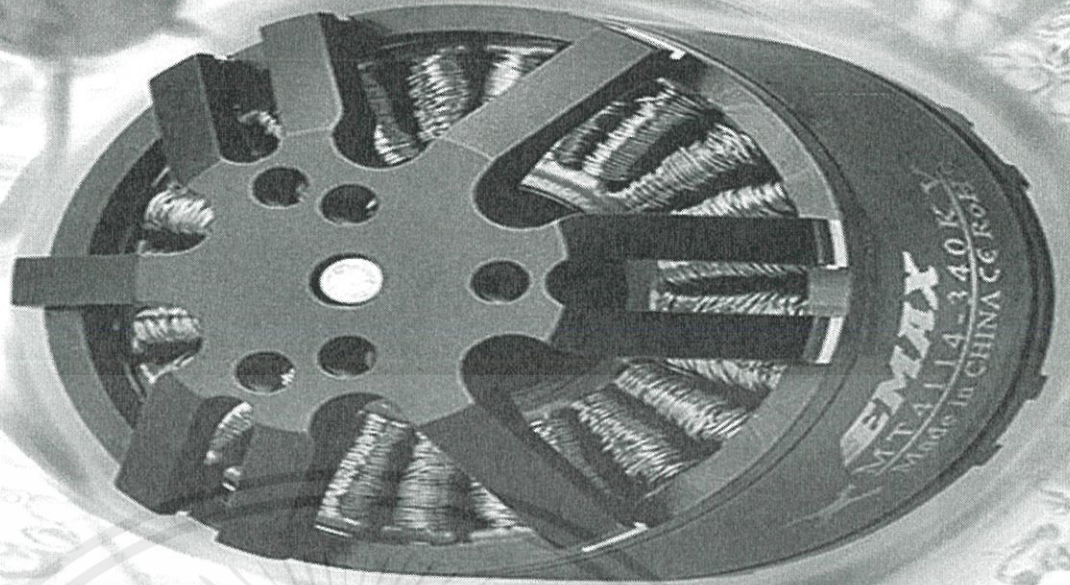
3D ROBOTICS OFFERS THE HARDWARE AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE HARDWARE, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL 3D ROBOTICS BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THE USE OF THE HARDWARE.

3D ROBOTICS ACCEPTS NO LIABILITY FOR DAMAGE(S) OR INJURIES INCURRED DIRECTLY OR INDIRECTLY FROM THE USE OF THIS PRODUCT.

SOFTWARE IS SUBJECT TO THE SEPARATE SOFTWARE LICENSE AGREEMENT ACCOMPANYING OR MADE AVAILABLE TO YOU IN CONNECTION WITH THE SOFTWARE. A PORTION OF THE SOFTWARE MAY CONTAIN OR CONSIST OF OPEN-SOURCE SOFTWARE, WHICH YOU MAY USE UNDER THE TERMS AND CONDITIONS OF THE SPECIFIC LICENSE UNDER WHICH THE OPEN-SOURCE SOFTWARE IS DISTRIBUTED. YOU AGREE THAT YOU WILL BE BOUND BY ANY AND ALL SUCH LICENSE AGREEMENTS, AND THAT YOUR USAGE OF THIS PRODUCT INDICATES YOUR ACCEPTANCE OF THOSE AGREEMENTS. TITLE TO SOFTWARE REMAINS WITH THE APPLICABLE LICENSOR(S). IN NO EVENT WILL 3D ROBOTICS BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Newly designed high-end brushless motors for multirotors

MT Series

MT2216



Lightweight design

Longer duration of flight

Low CG design

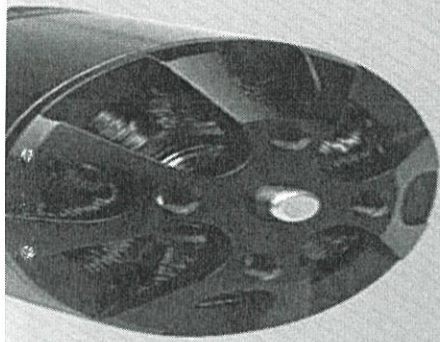
MT2216

Using imported Silicon steel coils from Japan; High quality NMB bearing of Japan; Ndfeb with strong magnetic and high temperature resistance to decrease motor's temperature increasing efficient.



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ห้ามนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่แบบลงเนื้อหา และตัดต่อเอกสารทุกครั้งที่มีการนำไปใช้

Very convenient to use; two installation of propellers: one for plastic propellers and the other one is for carbon fiber propellers



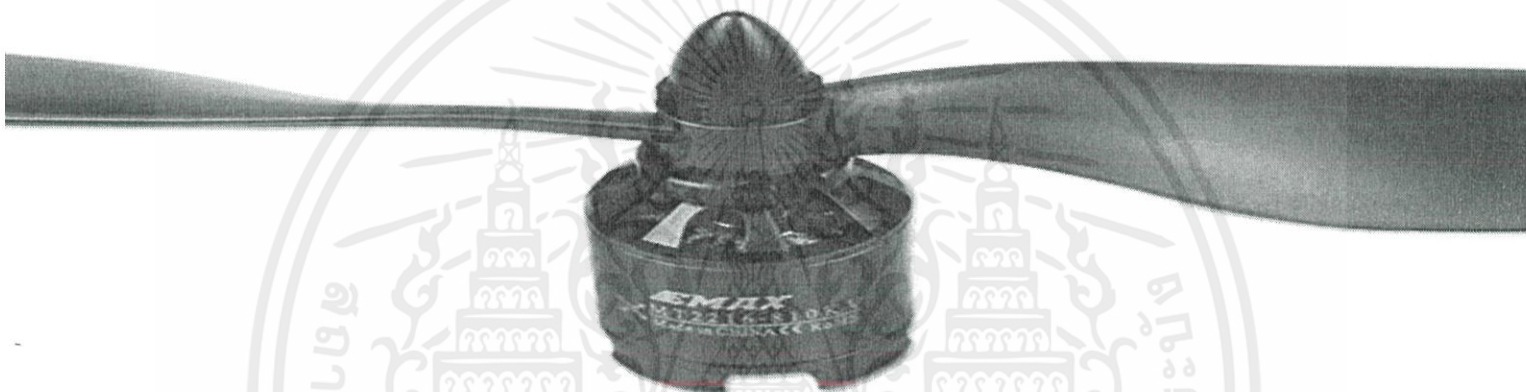
MT series motor got CW thread and CCW thread to decrease potential safety hazard of installing CCW propellers.

Low CG and light weight design; increasing efficient; longer duration of fly; more power.

100% balanced of each motor to decrease shaking to the minimum

MT2216

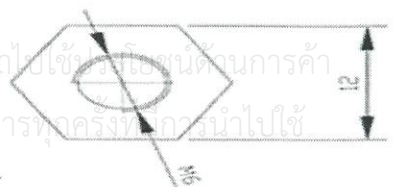
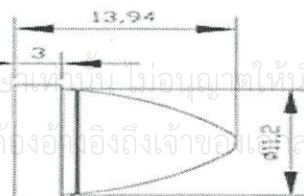
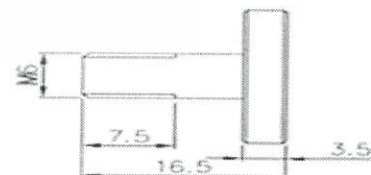
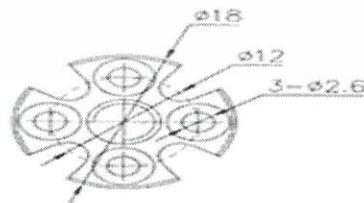
Installation for general type propeller



Installation for three-hole CF propeller

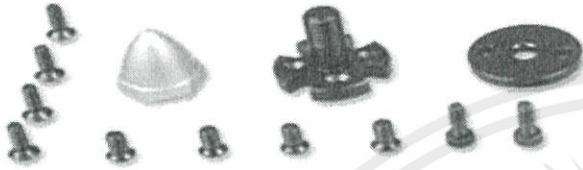
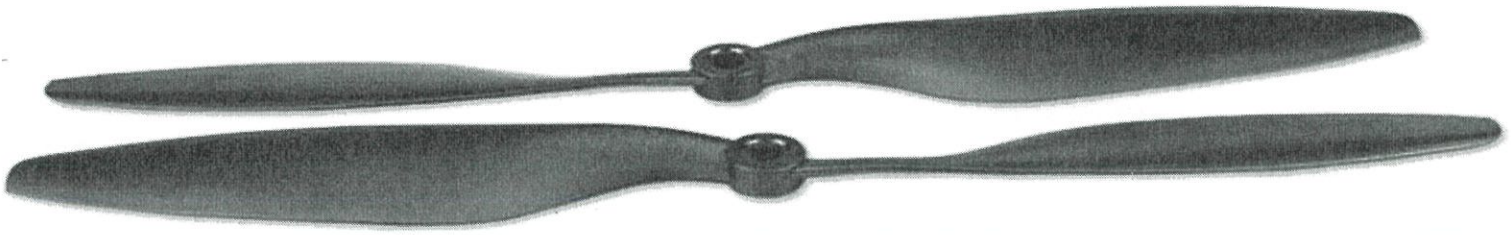


EMAX

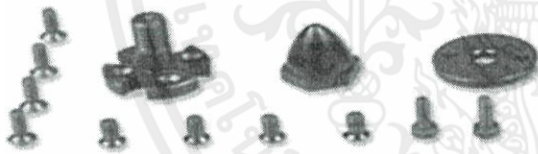


เอกสารนี้เป็นลิขสิทธิ์ของ EMAX สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อการค้า
 ไม่ว่ากรณีใดๆ ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งในการนำไปใช้

Accessory pack for CW thread motor



Accessory pack for CCW thread motor



EMAX

MT2216 SPECIFICATIONS

KV:810

MAX Thrust:950G

NO.OF CELL:3-4S

Frammework:12N14P

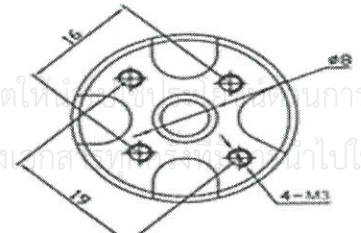
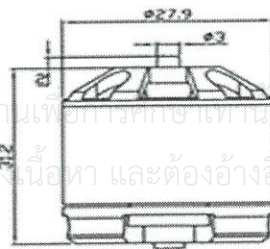
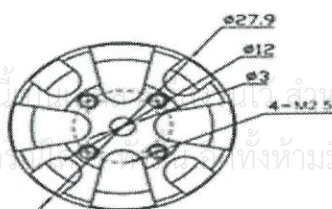
Propeller:EMAX1045

Length:33.2MM

Weight:80G

Shaft:3MM

Diameter:27.9MM



เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ EMAX สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่าการเผยแพร่ในรูปแบบใดก็ตาม ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารที่เกี่ยวข้องในเนื้อหาไปใช้

The voltage	propeller size	current (A)	thrust (G)	power (W)	efficiency (G/W)	RPM
11.1	APC10*3.8	1	130	11.1	11.7	3230
		2	220	22.2	9.9	4030
		3	300	33.3	9.0	4640
		4	380	44.4	8.6	5090
		5	440	55.5	7.9	5420
		6	490	66.6	7.4	5690
		7	550	77.7	7.1	5950
		8	610	88.8	6.9	6180
		9	650	99.9	6.5	6340
		10.2	720	113.2	6.4	6590
11.1	APC11*4.7	1	130	11.1	11.7	2570
		2	230	22.2	10.4	3230
		3	310	33.3	9.3	3690
		4	390	44.4	8.8	4080
		5	460	55.5	8.3	4380
		6	520	66.6	7.8	4620
		7	570	77.7	7.3	4840
		8	625	88.8	7.0	5080
		9	670	99.9	6.7	5190
		10	710	111	6.4	5320
		11	750	122.1	6.1	5490
		12.8	830	142.1	5.8	5700
11.1	1055 Carbon fiber Propeller	1	130	11.1	11.7	3230
		2	230	22.2	10.4	4150
		3	310	33.3	9.3	4810
		4	390	44.4	8.8	5420
		5	460	55.5	8.3	5870
		6	530	66.6	8.0	6220
		7	570	77.7	7.3	6580
		8	650	88.8	7.3	6880
		9.2	710	102.1	7.0	7290
		11.1	EMAX1045	1	130	11.1
2	220			22.2	9.9	3850
3	300			33.3	9.0	4480
4	380			44.4	8.6	5040
5	450			55.5	8.1	5440
6	500			66.6	7.5	5780
7	550			77.7	7.1	6070
8	600			88.8	6.8	6280
9.8	670			108.8	6.2	6620
14.8	APC10*3.8	1	160	14.8	10.8	3420
		2	270	29.6	9.1	4340
		3	360	44.4	8.1	4950
		4	450	59.2	7.6	5440
		5	540	74	7.3	5770
		6	600	88.8	6.8	6100
		7	670	103.6	6.5	6340
		8	730	118.4	6.2	6540
		9	780	133.2	5.9	6810
		10	830	148	5.6	6930
		11	890	162.8	5.5	7130
		12	930	177.6	5.2	7310
		13	960	192.4	5.0	7480
		14	1010	207.2	4.9	7580
		15.4	1070	227.9	4.7	7730
14.8	EMAX1045	1	160	14.8	10.8	3270
		2	260	29.6	8.8	4160
		3	340	44.4	7.7	4830
		4	420	59.2	7.1	5290
		5	500	74	6.8	5680
		6	560	88.8	6.3	6100
		7	630	103.6	6.1	6450
		8	680	118.4	5.7	6700
		9	720	133.2	5.4	6900
		10	770	148	5.2	7100
		11	820	162.8	5.0	7320
		12	860	177.6	4.8	7530
		13	900	192.4	4.7	7710
		14.6	950	216.1	4.4	7890

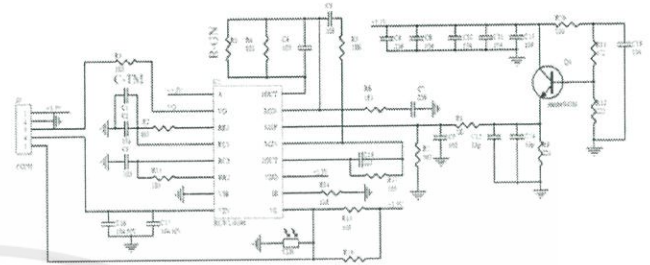
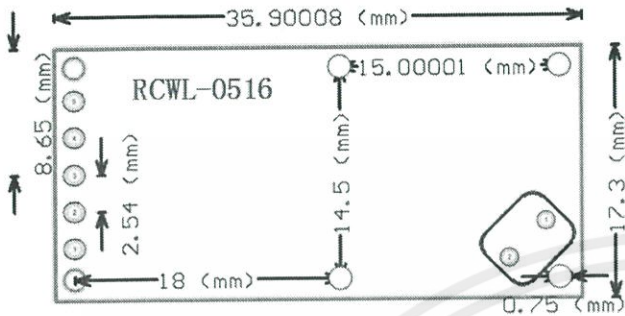
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต การนำออกไปใช้ประโยชน์ด้านการค้า หรือการนำออกไปใช้

RCWL-0516

RCWL-0516 microwave radar sensor module Human body induction switch module Intelligent sensor

Features:

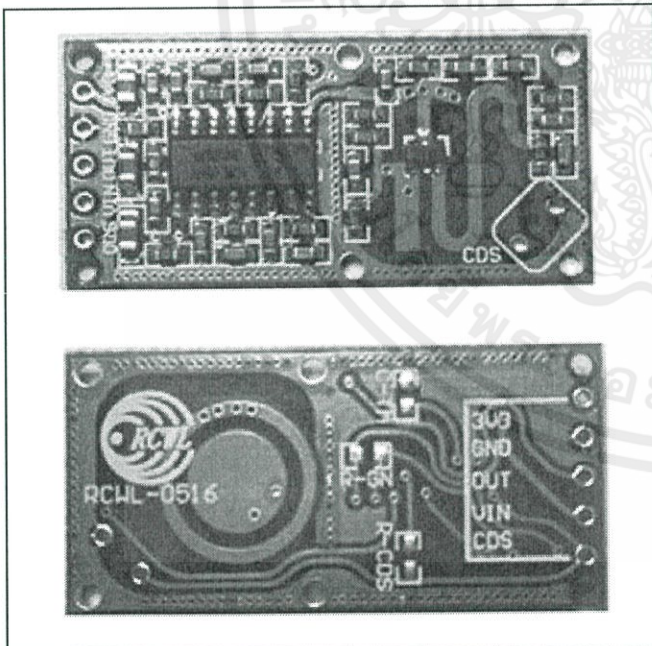
- 1, transmission signal processing control chip **RCWL-9196**
- 2, wide operating voltage range: 4.0-28.0V
- 3, compared with the traditional infrared feeling PIR, with the penetrating detection capability
- 4, block time, distance adjustable
- 5, output 3.3V power supply



Application Design Note:

- 1, the sensing face in front of the gold without any shelter.
- 2, front and rear sensing surface space to set aside more than 1CM
- 3, the module carrier plane and install as flat line
- 4, a certain application of effective detection area
- 5, the component side of the module is positive sensing face, the opposite is negative sensing surface. Negative sensing surface sensing less effective
- 6, microwave modules can not be large-scale applications in the same area, otherwise there will be mutual interference. Between single individuals over distance greater than 1M

RCWL-0516 is a doppler radar microwave motion sensor module which can act as an alternative to a PIR motion sensor. This git repository is an attempt to collect the rather scant information on this board in one place.



The unit I have was supplied by IC station (SKU 10630): <http://www.icstation.com/rcwl-0516-microwave-motion-sensor-module-radar-sensor-body-induction-module-100ma-p-10630.html> (Use coupon code 'joeics' for a 15% discount).

Operating frequency: The product specification omits the operating frequency. I found a carrier at 3.181GHz on my unit using a HackRF One SDR radio (see spectrum plot below). I suspect this frequency will vary from device to device: it would be difficult to have a tight specification with such a simple RF circuit on FR4 PCB.

Working voltage: 4 - 28V. It provides a convenient 3.3V output to drive a MCU (good for 100mA ?).

The forward side of the board is the side with components. This side should face the objects being detected. Do not obstruct forward side with anything metallic. The back side should have clearance of more than 1cm from any metal.

Board header

Pin	Function
3V3	3.3V regulated output. Max 100mA (?)
GND	Ground
OUT	Trigger: high (3.3V) if motion detected. 0V normally.
VIN	4 - 28V supply voltage
CDS	(light sensor related.. TODO)

Schematic

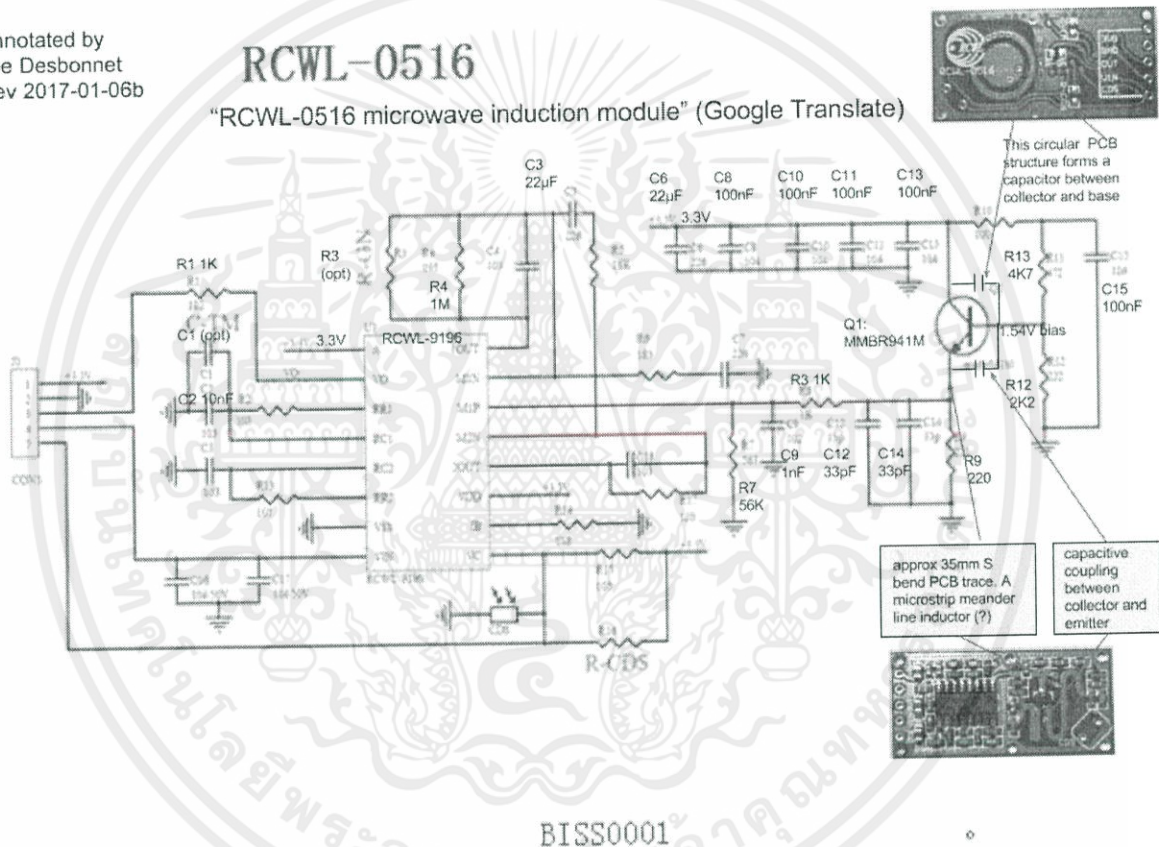
The only schematic I could find is very low resolution and it's hard to make out some of the text. However I've been reverse engineering it and adding my annotations.

There are two parts to this schematic. A microwave frequency transmitter/receiver/mixer and a much lower frequency part based on an IC (marked RCWL-9196) which is very similar to the BISS0001 IC used in PIR motion detectors.

First the microwave part:

The best explanation of how the microwave part of this works is in patent [EP3091605A1](#). It describes as similar type of module operating at 5.8GHz.

Annotated by
Joe Desbonnet
Rev 2017-01-06b



"Note that U1 except pin 8 remaining pins are compatible with the BISS0001; their function is also compatible." (Google Translate)

At the heart of the RF is a Q1 a MMBR941M high frequency NPN transistor [5] in what is probably a Colpitt oscillator [6] configuration. The schematic above is misleading because it omits a key inductor and capacitors constructed from PCB traces (a microline inductor and capacitor). The inductor is the S curve trace on the top surface and capacitors are the ring structure on the bottom surface and also the rectangular block to the left of the S curve. Using the formula at reference [12] below I calculate the inductance of the S curve to be (very approximately) 10nH.

A critical function of a doppler radar is to be able to 'mix' the reflected signal with the transmitted signal to arrive at a frequency which is the difference between the transmitted and reflected signal.

In this board Q1 also cleverly assumes the function of the mixer: [TODO: this really needs to be explained].

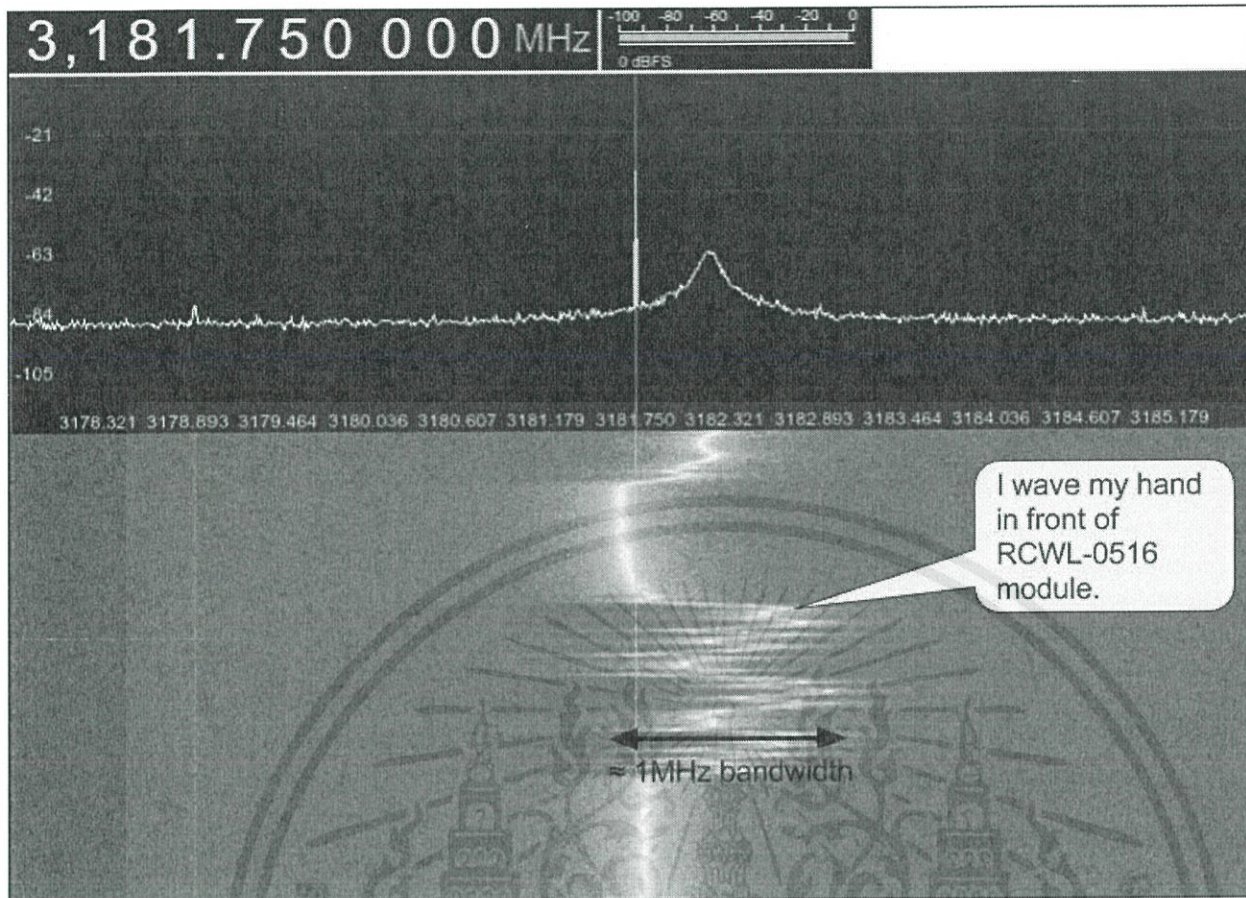
The low doppler frequency difference is extracted by a low pass RC filter (C9 = 1nF, R3 = 1k, $f_c = 1/2\pi RC \approx 160\text{kHz}$) and amplified by the RCWL-9196 IC and treated exactly the same as a signal from a PIR sensor.

Update 4 Jan 2017: finally found the signal at 3.181GHz with the HackRF One SDR!

One interesting observation: waving my hand in front of the sensor causes significant changes in the transmitting frequency, shifting by up to 1MHz.

My theory: the low frequency doppler shift causes small changes in the transistor base bias.

I used spice simulations to verify that small changes to transistor base bias causes changes in oscillation frequency. By running a few simulations I estimate that 1µV change in bias will change oscillation frequency by 1.4MHz.



The low frequency part

The core of the low frequency signal processing is an IC marked RCWL-9196. The schematic says (in chinese) that it's similar to a BISS0001 PIR IC. But there are differences. Unfortunately I can't find any hard information (eg datasheet) on this. Now can I find any information on the brand/company name "RCWL".

Pin number	BISS0001	RCWL-9196
1	A Retriggerable & non-retriggerable mode select (A=1 : re-triggerable)	3.3V regulated output (100mA max?)
2	VO Detector output pin (active high)	same
3	RR1 Output pulse width control (Tx)	same?
4	RC1 Output pulse width control (Tx)	same?
5	RC2 Trigger inhibit control (Ti)	same?
6	RR2 Trigger inhibit control (Ti)	same?
7	Vss Ground	same
8	VRF RESET & voltage reference input (Normally high. Low=reset)	Vin (4 - 28V)
9	VC Trigger disable input (VC > 0.2Vdd=enable; Vc < 0.2Vdd =disabled)	same
10	IB Op-amp input bias current setting	?
11	Vdd Supply voltage	3.3V regulated output (again?)
12	2OUT 2nd stage Op-amp output	same
13	2IN- 2nd stage Op-amp inverting input	same
14	1IN+ 1st stage Op-amp non-inverting input	same
15	1IN- 1st stage Op-amp inverting input	same
16	1OUT 1st stage Op-amp output	same

Adjustment components

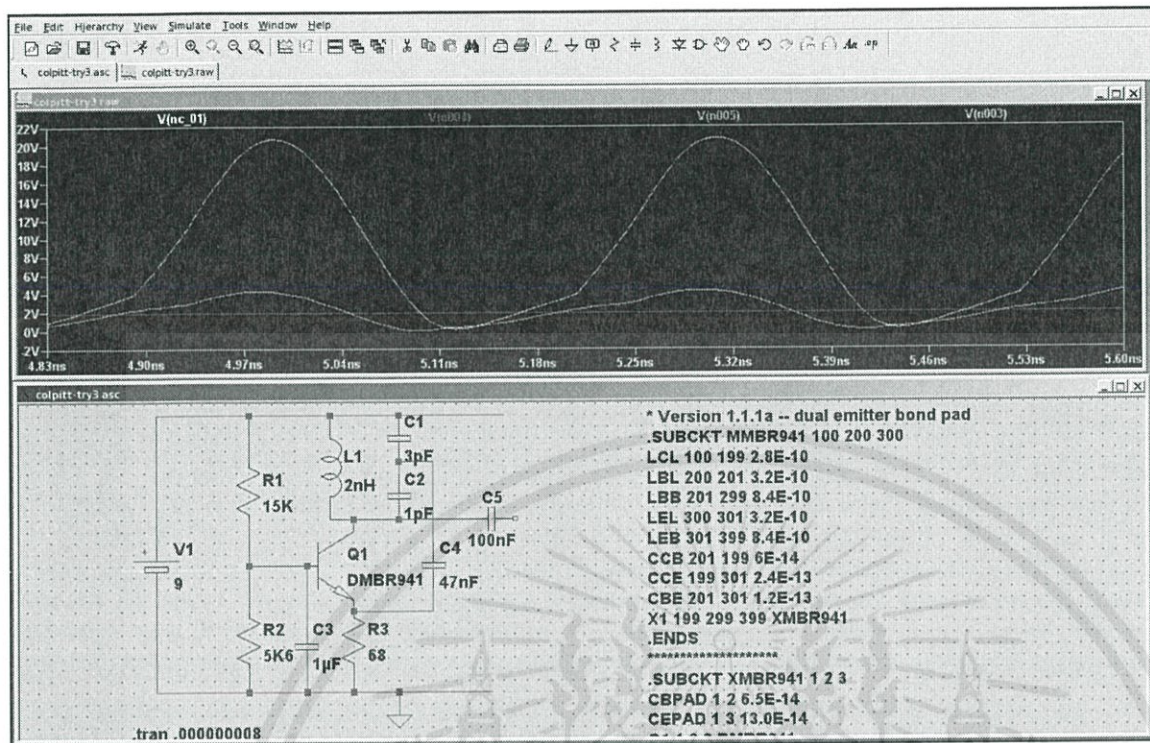
On the back of the board (the side without components) are pads for 3 optional components (0805 dimensions).

Pad	Function
C-TM	Regulate the repeat trigger time. The default (unpopulated) time is 2s. A SMD capacitor to extend the repeat trigger time. Pin 3 of the IC emits a frequency (f), and the trigger time in seconds is given by (1/f) * 32678
R-GN	The default detection range is 7m, adding a 1M resistor reduces it to 5m
R-CDS	the VCC is in parallel connection with CDS(RCWL-9196 pin 9) through R-CDS. Connect the LDR at the R-CDS to turn off the detecting function at night. (?? TODO: make sense of this)

ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Spice simulation

I started with an example Colpitt circuit [9] and substituted the 2N3904 NPN with a MMBR941 (Spice model from [10]). I am using the Windows LTSpice from Linear Technologies (available as free download [11], also works with Linux under Wine emulator). See [colpitt.asc](#) for a working Colpitt oscillator and [rcwl-0516.asc](#) for a model of the RCWL-0516 (however it does not oscillate!).

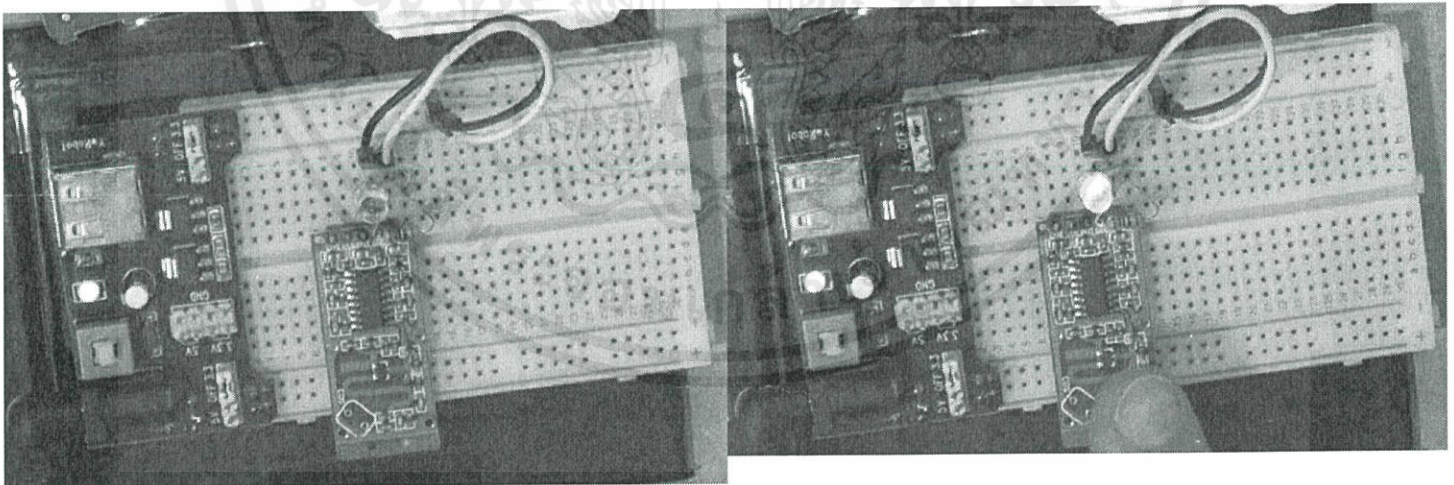


Doppler effect calculations

- If f_t is the transmitted frequency, f_r is the reflected frequency (as measured by the common transmit/receive antenna on the sensor), v is the speed of the target relative to the sensor (negative if receding, positive if advancing toward sensor), c is the speed of light and $f_d = (f_r - f_t)$ is the doppler shift, then:

$$f_r = f_t (c + v) / (c - v)$$

$$f_d = f_r - f_t = 2v f_t / (c - v)$$
 If $(c \ll v)$ then $f_d \approx 2v f_t / c$
 Assume typical human motion speed of $v = 1$ m/s, $f_t = 3.181$ GHz, $c = 2.998E8$ m/s, then $f_d = 10$ Hz.



/** Example NOT tested : Log the output pin of a RCWL-0516 radar module to a 433 MHz XY-FST transmitter. */

#include <VirtualWire.h>

#define PIN_RADAR 2
#define PIN_TX 9
#define PIN_LED 13

```
void setup() {
  Serial.begin(9600);
  pinMode(PIN_LED, OUTPUT);
  vw_set_tx_pin(PIN_TX); // Arduino pin to connect the receiver data pin
  vw_setup(6000); // bps connection speed
}
```

ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int rv = -1;

void loop() {
  digitalWrite(PIN_LED, HIGH);
  int v = digitalRead(PIN_RADAR);
  if (v != rv) {
    rv = v;
    char msg[20];
    sprintf(msg, "R %lu %d", millis() / 1000, v);
    vw_send((uint8_t *)msg, strlen(msg));
    Serial.println(msg);
    vw_wait_tx(); // Wait to finish sending the message
  }
  digitalWrite(PIN_LED, LOW);
  delay(100);
}
```

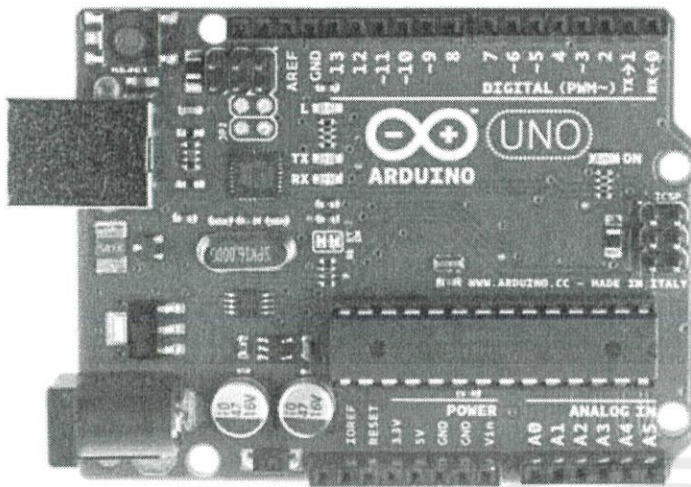
p.s. more info

<http://www.icstation.com/rcwl-0516-microwave-motion-sensor-module-radar-sensor-body-induction-module-100ma-p-10630.html>

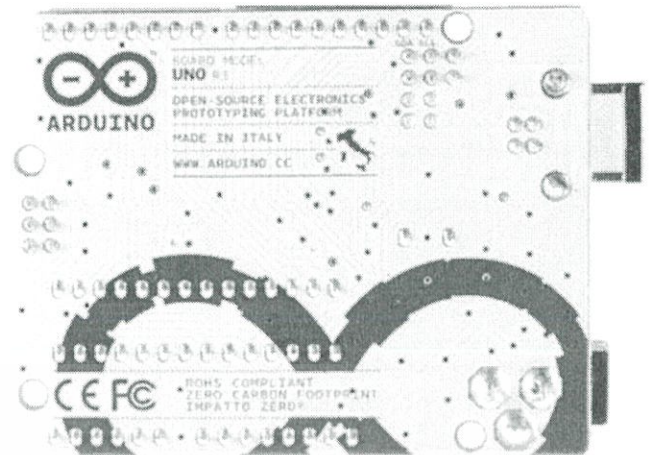


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

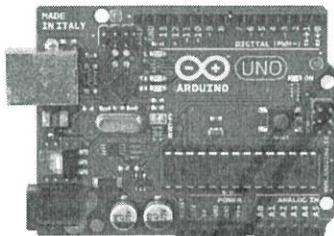
Arduino Uno



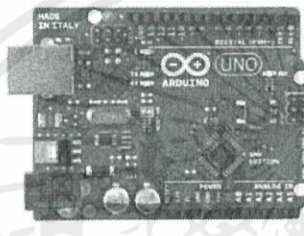
Arduino Uno R3 Front



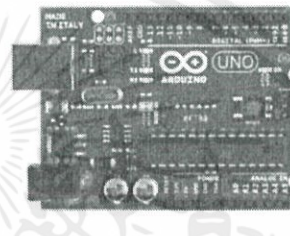
Arduino Uno R3 Back



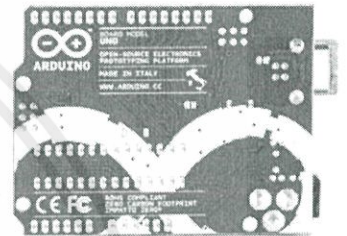
Arduino Uno R2 Front



Arduino Uno SMD



Arduino Uno Front



Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller ATmega328

Operating Voltage ที่สงวนไว้ สำ 5V บัการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาติให้นำไปใช้ประโยชน์ด้านการค้า

Input Voltage (recommended) 7-12V

ผู้ว่าการผลิต | พังสน อภทงหี หมมีเตดกแปลงเนื้อหา และต้ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตักแบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาไปใช้

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Autocorrection (Software) Reset

ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

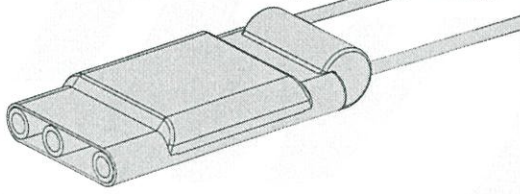
USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

X-ROTOR USER MANUAL
Multi-Rotor
 Brushless Electronic Speed Controller
 8A • 10A • 20A • 40A



Thank you for purchasing this HOBBYWING product! Brushless power systems can be very dangerous. Any improper use may cause personal injury and damage to the product and related devices. We strongly recommend reading through this user manual before use. Because we have no control over the use, installation, or maintenance of this product, no liability may be assumed for any damage or losses resulting from the use of the product. We do not assume responsibility for any losses caused by unauthorized modifications to our product. We, HOBBYWING, are only responsible for our product cost and nothing else as result of using our product.

01 Features

- Special core program for multi-rotor controllers greatly improves throttle response.
- All settings except the timing are preset, making the usage really simple, highly intelligent and adaptive.
- The twisted-pair design of the throttle signal cable effectively reduces the crosstalk produced in signal transmission and makes flight more stable.
- Compatible with various flight-controllers and supports a signal frequency of up to 621Hz. (Notes: all throttle signals over 500Hz are non-standard signals.)

02 Specifications

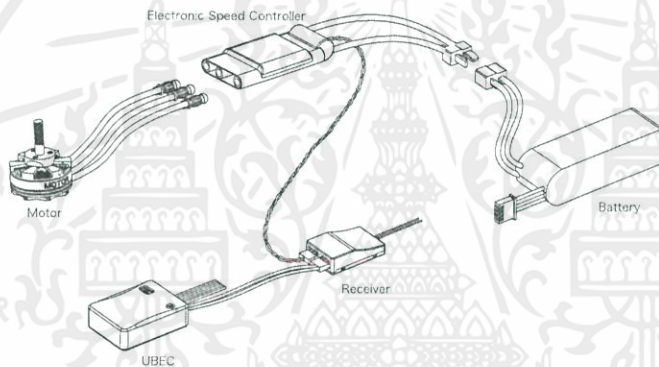
Model	Con. Current	Peak Current (10s)	BEC	LiPo	Programmable Item	Weight	Size	Applications (For reference)
XRotor 8A	8A	12A	No	2S	Timing (High/Intermediate)	7g	44.2x12.2x9.2 mm	200/250 Class (Multi-rotor)
XRotor 10A	10A	15A	No	2-3S	Timing (High/Intermediate)	6.5g	36.5x16.4x4.1 mm	250 Class (Multi-rotor)
XRotor 20A	20A	30A	No	3-4S	Timing (High/Intermediate)	14g	52.4x21.5x7 mm	330/450 Class (Multi-rotor)
XRotor 40A	40A	60A	No	2-6S	Timing (High/Intermediate)	26g	68x25x8.7 mm	550/650 Class (Multi-rotor)

03 User Guide

Throttle Calibration & ESC Programming

- Users need to calibrate the throttle range when they start to use a new XRotor brushless ESC or another transmitter.
- Users can change the Timing setting when some abnormality occurs in the ESC driving the disc-type motor or need the motor to reach a higher RPM. (Note: Intermediate Timing is the default setting.)

1 Motor Wiring

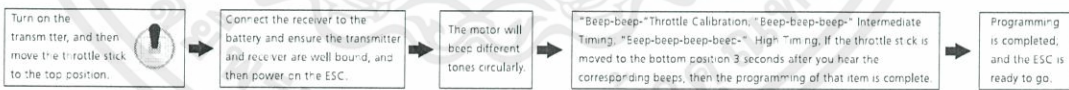


WARNING! This is an extremely powerful brushless motor system. We strongly recommend removing your propellers for your own safety and the safety of those around you before performing calibration and programming functions with this system.

2 Throttle Range Calibration



3 ESC Programming



WARNING! Generally speaking, intermediate timing is applicable to most motors and it brings higher efficiency but less heat to the motors and speed controllers. High timing can increase the RPM of the motor, but it also increases the motor temperature. Please perform a test flight or on the ground and ensure everything is functioning properly first, then fly your aircraft into the sky.

04 Normal Start-up Process and Protections



Start-up Protection: The ESC will shut down the motor if it fails to start the motor normally within 2 seconds by increasing the throttle value. In this case, you need to move the transmitter throttle stick back to the bottom position and restart the motor. (Possible causes of this problem: poor connection/disconnection between the ESC and motor wires, propellers are blocked, etc.)
Over-load Protection: The ESC will cut off the power/output when the load suddenly increases to a very high value. Normal operation will not resume until the throttle stick is moved back to the neutral position. The ESC will automatically attempt to restart when the motor and the ESC are out of sync.
Throttle Signal Loss Protection: When the ESC detects loss of signal for over 0.25 second, it will cut off the output immediately to avoid an even greater loss which may be caused by the continuous high-speed rotation of propellers or rotor blades. The ESC will resume the corresponding output after normal signals are received.

05 Trouble shooting

Trouble	Warning Tone	Possible Cause	Solution
The ESC was unable to start the motor.	"Beep-beep-beep" (The motor beeps rapidly)	The throttle stick is not at the bottom position.	Move the throttle stick to the bottom position or recalibrate the throttle range.
The ESC was unable to start the motor.	"Beep, beep, beep....." (Time interval is 1 second)	No output signal from the throttle channel on the receiver.	Check if the transmitter and receiver are well bound, Check if the throttle wire has been properly plugged into the throttle channel on the receiver.
The ESC was unable to start the motor.	"BB, BBB, BBBB" (These tones are played circularly)	The "Normal / Reverse" direction of the throttle channel on transmitter is incorrect.	Refer to the transmitter instructions and correctly set the "Normal / Reverse" direction of the throttle channel.

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ HOBBYWING กรุณาอย่าเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดก็ตาม หากมีข้อผิดพลาดหรือข้อสงสัย กรุณาติดต่อฝ่ายบริการลูกค้า
 HOBBYWING ประเทศไทย โทร. 02-010-1111 หรือ 02-010-1112