หุ่นอากาศยานไร้คนขับ
Auto Pilot Aerial Robot

จิรภัทร์ จันทนา
Jirapat Janthana
ชุมพล ศิริวัฒนสิทธิ์
Chumpol Sitiwattanasit
ถิรวิทย์ แซ่โง้ว
Tirawit Sae-Ngo

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2556

หุ่นอากาศยานไร้คนขับ

Auto Pilot Aerial Robot

โดย

จิรภัทร์ จันทนา

ชุมพล ศิริวัฒนสิทธิ์

ถิรวิทย์ แซ่โง้ว

อาจารย์ที่ปรึกษา

ดร. แสงระวี บัวแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระเจ้าเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์อากาศยานไร้คนขับ

Auto Pilot Aerial Robot

ผู้จัดทำ	นายจิรภัทร์	จันทนา	รหัสประจำตัว 53010216
	นายชุมพล	ศิริวัฒนสิทธิ์	รหัสประจำตัว 53010383
	นายถิรวิทย์	แซ่โง้ว	รหัสประจำตัว 53010562

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ดร.แสงระวี บัวแก้ว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์อากาศยานไร้คนขับ			
นักศึกษา	นายจิรภัทร์	จันทนา	รหัสประจำตัว	53010216
	นายชุมพล	ศิริวัฒนสิทธิ์	รหัสประจำตัว	53010383
	นายถิรวิทย์	แซ่โง้ว	รหัสประจำตัว	53010562
ปริญญา	วิศวกรรมศาตรบัณฑิต			
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์			
ปีการศึกษา	2556			
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ดร.แสงระวี	บัวแก้ว		

บทคัดย่อ

โครงงานหุ่นอากาศยานไร้คนขับนี้ มีจุดมุ่งหมายเพื่อทำการศึกษาไมโครคอนโทรลเลอร์ (Microcontroller) การออกแบบวัสดุทางเลือกเพื่อใช้สร้างโครงของหุ่น และการเขียนโปรแกรมสำหรับควบคุม “หุ่นอากาศยานไร้คนขับ” จากระยะไกลด้วยระบบไร้สายโดยใช้แม็ทแล็บ(Matlab) ทั้งนี้สามารถควบคุมหุ่นโดยใช้รีโมท(Remote) หรือให้หุ่นบินด้วยระบบอัตโนมัติได้ หุ่นจะมีสี่ใบพัดและมีบอร์ดอาร์ดุยโน(Arduino) เป็นตัวสั่งการ โดยบอร์ดจะประกอบด้วยระบบรักษาระดับเพดานของการบินคือ ไจโรสโคป(Gyroscope) เพื่อวัดมุมของหุ่น นอกจากนี้ยังมีระบบ จีพีเอส(GPS) เพื่อทำการตรวจวัดพิกัด, ความเร็วและส่งข้อมูลมาสู่โปรแกรมในเครื่องคอมพิวเตอร์ เพื่อการศึกษาพัฒนาระบบของหุ่นให้มีประสิทธิภาพและนำไปใช้งานในด้านต่างๆได้จริงในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Autopilot Aerial Robot		
Student	Jirapat	Janthana	Student ID 53010216
	Chumpol	Siriwattanasit	Student ID 53010383
	Tirawit	Sae-Ngo	Student ID 53010562
Degree	Bachelor of Engineering		
Program	Electronics Engineering		
Year	2013		
Thesis Advisor	Dr.Seangrawee Buakaew		

Abstract

The objective of "Autopilot Aerial Robot" project is to learn about microcontroller device, to design alternative materials for robot frame, and to design robot controller program from Matlab. Also the robot can be controlled by remote or autopilot. The robot consisted with four propellers and commanded by Arduino board, that consisted with gyroscope that can measure the angle and level flight of robot. In addition the robot contain with GPS for measure the coordinates, velocity, and export the data to program in receiver computer for develop and make the robot system more efficiency and more useful in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำโครงงานหุนยนต์อากาศยานไร้คนขับนี้ ได้รับการสนับสนุนและช่วยเหลือให้คำแนะนำจาก อาจารย์ภาควิชาอิเล็กทรอนิกส์ และได้รับความอนุเคราะห์ในด้านอุปกรณ์และเครื่องมือจากภาควิชาอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่ง ดร.แสงระวี บัวแก้ว ที่ให้แนวคิดและคำปรึกษา การแก้ไขปัญหาและการสนับสนุนในการทำวิจัย รวมไปถึงบิดาแลมารดาของผู้ทำโครงงาน ที่ให้การสนับสนุนในด้านทุนทรัพย์เพื่อทำการจัดหาอุปกรณ์บางส่วน ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาของทุกท่านอย่างที่สุดและขอกราบขอบพระคุณเป็นอย่างสูง

จิรภัทร์ จันทนา
ชุมพล ศิริวัฒนสิทธิ
ถิรวิทย์ แซ่โง้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญรูป VIII
บทที่ 1 บทนำ 1
1.1 ความเป็นมาของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตในโครงงาน 2
1.4 ประโยชน์ที่จะได้รับ 2
1.5 รายละเอียดโดยย่อ 2
บทที่ 2 ทฤษฎีและหลักการ 3
2.1 Unmaned Aerial Vehicle 3
2.1.1 หลักการการบินแบบควอดโรเตอร์(Quadrotor) 4
2.1.1.1 โฮเวอริง(Hovering) 4
2.1.1.2 ทรอตเทิล(Throttle) 4
2.1.1.3 โรลล์(Roll) 5
2.1.1.4 พิทช์(Pitch) 5
2.1.1.5 ยอว์(Yaw) 5
2.2 PID control 7
2.3 Accelerometer 8
2.4 Gyroscope 12
2.5 การเลือกใช้มอเตอร์ 13
2.6 การเลือกใช้ชุดควบคุมความเร็ว 14
2.6.1 การใช้งานสัญญาณ PWM ในการควบคุม มอเตอร์บรัชเลส 15
2.7 แบตเตอรี่ลิเทียมโพลิเมอร์ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 3 วิธีการและขั้นตอนการดำเนินงาน 17
3.1 การออกแบบเครื่องต้นแบบอากาศยานไร้นักบิน 18
3.1.1 ส่วนลำตัวของอากาศยาน 18
3.1.2 ส่วนแขนของอากาศยาน 18
3.2 การเลือกใช้ไมโครคอนโทรลเลอร์ 19
3.3 โฟลวชาร์ทการอ่านค่าและสั่งงานมอเตอร์ 20
3.4 การจัดเตรียมอุปกรณ์ 21
3.4.1 การทำในส่วนโครงหุ่นยนต์ 21
3.4.2 การติดตั้งมอเตอร์,ใบพัดและ ESC 21
3.4.3 การติดตั้งบอร์ดควบคุมเข้ากับลำตัว 22
3.4.4 การเชื่อมต่ออุปกรณ์ทั้งหมดเข้าสู่บอร์ดควบคุม 22
3.4.5 เชื่อมต่อบอร์ดจ่ายไฟและแบตเตอรี่ 23
3.4.6 ติดตั้ง code ลงบอร์ด 24
3.5 ออกแบบลายวงจรพิมพ์และสร้างแผ่นวงจรพิมพ์ 26
3.6 การดูข้อมูลผ่านซีเรียลพอร์ท 27
3.7 การใช้การประมวลผลภาพเข้ามาช่วยเพื่อการบินติดตามเป้าหมายอัตโนมัติ . 28
3.7.1 เข้าโปรแกรม Matlab 28
3.7.2 ที่หน้าต่าง Command Window 28
3.7.3 เลือกใช้ 'winvideo' 28
3.7.4 การนำข้อมูลที่ได้มาประกาศใช้งานกล้องในหน้าต่าง Script 29
3.7.5 ใช้การจับภาพจากเฟรมปัจจุบันเข้ามาเพื่อประมวลผล 29
3.7.6 แปลงภาพRGBเป็นภาพระดับเทา 30
3.7.7 ใช้ Median Filtering 30
3.7.8 การแปลงภาพระดับเทาเป็นภาพ Binary 31
3.7.9 การกำจัด Noise ที่มีขนาดเล็ก 31
3.7.10 ทำการซ้อนทับภาพด้วยโครงร่างวงกลม 32
3.7.11 เติมช่องโหว่ในวงกลมทั้งหมดที่พบ 32
3.7.12 ใช้ทฤษฎีการหาขอบวงกลมและ Roundness 33
3.7.13 กำหนดเส้น X1,X2,Y1,Y2 เพื่อสร้างพื้นที่แบบพิกัดตาราง 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 การทดลองและผลการทดลอง 35

4.1 การทดลองเปิดระบบการทำงาน 35

4.2 การทดลองวัดสัญญาณที่ออกจากขาOutput 37

4.3 ข้อมูลจากซีเรียลพอร์ทที่เป็น Analog จากเซ็นเซอร์ 45

4.4 ข้อมูลจากซีเรียลพอร์ทของ Gyroscope ที่ผ่านการแปลงและคำนวณค่า 45

4.5 ข้อมูลจากซีเรียลพอร์ทของ Accelerometer ผ่านการแปลงและคำนวณค่า 46

บทที่ 5 สรุปผลการทดลอง 48

5.1 สรุปผลการทดลอง 48

5.2 ปัญหาและแนวทางการแก้ไข 48

5.3 แนวทางการพัฒนา 49

บรรณานุกรม 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 สรุปการใช้งานขาของบอร์ดมัลติวี(Multiwii) 25
3.2 แสดงค่ารับค่าตัวอักษรและรูปแบบการเคลื่อนที่สำหรับบอร์ดอาดุยโน(Arduino) 34
4.1 แสดงผลคำสั่งที่ได้รับจากรีซีฟเวอร์(Receiver) 35
4.2 แสดงผลสัญญาณที่ได้รับจากเอาท์พุท(Output) 37
4.3 ค่า Duty Cycle เป็นค่า 0-255 ที่ใช้ในการจ่ายให้ ESC ของ Arduino Board 47
4.4 ค่า Duty Cycle เป็นเปอร์เซ็นต์ ที่ใช้ในการจ่ายให้ ESC ของ Arduino Board 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ควอดโรเตอร์(Quadrotor) 3
2.2 โฮเวอริง(Hovering) 4
2.3 ทรอตเทิล(Throttle) 4
2.4 โรลล์(Roll) 5
2.5 พิทช์(Pitch) 5
2.6 ยอว์(Yaw) 5
2.7 จำลองสภาวะนิ่ง 8
2.8 จำลองสภาวะเคลื่อนที่ไปทางซ้าย 8
2.9 สภาวะแรงโน้มถ่วง 9
2.10 สภาวะเคลื่อนที่ในกรณีสองแนวแรง 10
2.11 ลูกศรแสดงทิศของแนวแรง 10
2.12 มุมของแนวแรง 11
2.13 แสดงอัตรามุมที่เปลี่ยนไป 12
2.14 แสดงการทำงานของมอเตอร์ไร้แปลงถ่าน 13
2.15 มอเตอร์บลัสเลสของ DJI INNOVATION 14
2.16 ชุดควบคุมความเร็ว ของ DJI INNOVATION 14
2.17 แสดงความกว้างของสัญญาณพัลส์ สัมพันธ์กับความเร็วในแบบต่างๆ 15
2.18 แบตเตอรี่ลิเทียมพอลิเมอร์ของ Everest 16
3.1 การออกแบบส่วนลำตัว 18
3.2 การออกแบบส่วนแขน 18
3.3 ไอซีเบอร์ ATMega 328P 19
3.4 โฟลวชาร์ทการอ่านค่าและสั่งงานมอเตอร์ 20
3.5แขนอากาศยาน 21
3.6 แขนอากาศยานทั้งสี่ 21
3.7 แขนและลำตัวอากาศยาน 22
3.8 การเชื่อมต่อลำตัว 22
3.9 การติดตั้ง ESC มอเตอร์ และ ใบพัด 23
3.10 การติดตั้งบอร์ดบนลำตัวยาน 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.11 การใช้งาน Pin ต่างๆของบอร์ด Multiwii 24
3.12 การเชื่อมต่อมอเตอร์กับPin ของบอร์ด 25
3.13 บอร์ดจ่ายไฟเลี้ยง 26
3.14 ลายวงจรส่วนจ่ายไฟ 26
3.15 การเชื่อมต่อเข้ากับคอมพอร์ทเพื่อดูข้อมูล 27
3.16 ผลของการใช้คำสั่งตรวจสอบอุปกรณ์ 28
3.17 การเรียกดูคุณสมบัติอุปกรณ์ 28
3.18 ภาพของวัตถุสีแดง 29
3.19 ภาพระดับเทา 30
3.20 แสดงการกำจัดสิ่งรบกวน 30
3.21 ภาพขาวดำ 31
3.22 การกำจัดนอยส์(Noise) 31
3.23 สร้างขอบเขตวงกลม 32
3.24 เติมเต็มช่องโหว่ในวงกลม 32
3.25 ใช้แพล็ทฟอร์ม(Platform) หาจุดศูนย์กลางวัตถุ 33
3.26 ตัวอย่างการกำหนดเขตแดน 33
4.1 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินในแนวดิ่ง 35
4.2 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินเอียงข้าง 36
4.3 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินเอียงหน้าหรือหลัง 36
4.4 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินหมุนรอบแกนตั้ง .. 37
4.5 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อคันบังคับอยู่ตรงกลาง 38
4.6 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับไปทางซ้าย 39
4.7 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับไปทางขวา 40
4.8 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับไปข้างหลัง 41
4.9 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับไปข้างหน้า 42
4.10 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับเลี้ยวไปทางซ้าย .. 43
4.11 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii) เมื่อโยกคันบังคับเลี้ยวไปทางขวา ... 44
4.12 แสดงข้อมูลของเซ็นเซอร์ที่วิ่งผ่านพอร์ทซีเรียล(Serial Port) 45
4.13 แสดงค่าของไจโรสโคป(Gyroscope)ที่ถูกแปลงค่าเป็นดิจิตอลแล้ว 45
4.14 แสดงแสดงค่าของไจโรสโคป(Accelerometer)ที่ถูกแปลงค่าเป็นดิจิตอลแล้ว 46

เอกสารนี้มีลิขสิทธิ์ ห้ามทำซ้ำ ดัดแปลง หรือนำไปใช้ในเชิงพาณิชย์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น จำกัดการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงงาน

อากาศยานไร้นักบิน (Unmanned Aerial Vehicle) เป็นยานบินบังคับ4ใบพัด (Quardrotor) ที่ใช้ในการสำรวจระยะไกล ซึ่งทำงานโดยระบบ Auto pilot ควบคุมตัวอากาศยานให้รักษาระดับเพดานการบินโดยส่วนประกอบของอากาศยานไร้นักบินมีสองชุดคือ ชุดควบคุม และชุดเก็บข้อมูล ชุดควบคุมจะทำหน้าที่ควบคุมส่วนเคลื่อนที่ต่าง ๆ ของอากาศยานและระบบไฟฟ้าสำหรับควบคุมกลไกในควอดโรเตอร์ เช่น ระบบมอเตอร์ ในการควบคุมแบบอัตโนมัติจะทำงานโดยอาศัยระบบของมอเตอร์และเซนเซอร์ตรวจสอบสถานะของควอดโรเตอร์เพื่อบังคับให้เครื่องบินสามารถทรงตัวอยู่ได้ในอากาศส่วนชุดเก็บข้อมูล จะเป็นชุดที่ใช้เก็บข้อมูล ความเร่งเชิงระนาบและค่าพารามิเตอร์ความเอียงจากตัวอากาศยานผ่าน Accelerometer,Gyroscope และส่งข้อมูลทั้งหมดในย่านความถี่ 2.4 GHz ซึ่งข้อมูลที่ส่งมานี้จะแสดงค่าพารามิเตอร์ต่าง ๆ ที่เกิดขึ้นกับตัวอากาศยานในขณะบิน

การประยุกต์ใช้อากาศยานไร้นักบิน เช่นในการสำรวจระยะไกลหรือที่สูง การสำรวจแหล่งต้นน้ำลำธารหรือสถานที่ที่เข้าถึงได้ยาก เช่น เขตป่าทึบ, ภูเขา, การติดตามการจราจรบนท้องถนนและทางอากาศ ข้อดีของระบบนี้คือสามารถสำรวจพื้นที่ขนาดใหญ่ในเวลาอันสั้นและมีความปลอดภัยสูงเนื่องจากไร้นักบินและประหยัดค่าใช้จ่ายกว่าใช้เครื่องบินจริงในการปฏิบัติภารกิจโครงงานนี้จึงเน้นการศึกษาและสร้างเครื่องต้นแบบอากาศยานไร้นักบินที่สามารถรักษาระดับเพดานการบินโดยอัตโนมัติที่สามารถส่งข้อมูล, พิกัดต่าง ๆ ลงมาสู่ภาคพื้นดินและสามารถนำไปประยุกต์ใช้งานด้านอื่นๆ

1.2 วัตถุประสงค์ของโครงงาน

โครงงานนี้มีวัตถุประสงค์ดังต่อไปนี้

1. ศึกษาเกี่ยวกับอุปกรณ์ไมโครคอนโทรลเลอร์เพื่อให้สามารถนำไปใช้ประยุกต์ใช้ในงานต่างๆได้
2. ศึกษาคำสั่งที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์ เขียนคำสั่ง และสร้างวงจรที่ทำให้วงจรไมโครคอนโทรลเลอร์สามารถทำตามวัตถุประสงค์ที่ต้องการได้

3.เพื่อศึกษาการสร้างเครื่องต้นแบบอากาศยานไร้นักบิน

1.3 ขอบเขตในโครงงาน

ศึกษา ทำความเข้าใจเกี่ยวกับระบบ คำสั่งที่ใช้ในการควบคุมอากาศยาน และนำไปประยุกต์สร้างระบบที่สามารถทำงานตามที่ต้องการได้

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่จะได้รับ

เป็นแนวทางการพัฒนาและออกแบบอากาศยานไร้นักบินที่ใช้งานได้จริง ได้ความรู้และความเข้าใจเกี่ยวกับโครงสร้าง การใช้งาน และคำสั่งต่างๆของอุปกรณ์หลักการบินเบื้องต้นของหุ่นอากาศยานต์ไร้คนขับ และสามารถนำไปประยุกต์ใช้งานต่างๆได้

1.5 รายละเอียดโดยย่อ

บทที่ 1 บทนำ เป็นการกล่าวถึงรายละเอียดความเป็นมาและวัตถุประสงค์ของโครงงาน เพื่อให้เห็นภาพรวมก่อนที่จะเข้าสู้บททฤษฎีต่อไป

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึงรายละเอียด ทฤษฎีเบื้องต้นของอุปกรณ์ต่างๆที่ใช้ในโครงงานอย่างคร่าวๆ

บทที่ 3 วิธีการและขั้นตอนการดำเนินงานกล่าวถึงรายละเอียดของอุปกรณ์ในส่วนต่างๆและวงจรควบคุมควบคุม

บทที่ 4 การทดลอง และผลการทดลอง จะกล่าวถึงวิธีการทดสอบ และผลการทดสอบหลังการนำอุปกรณ์และวงจรควบคุมที่ได้สร้างจากบทที่ 3 แล้วทำการวิเคราะห์ผลที่เกิดขึ้น

บทที่ 5 สรุปผลการทดลอง จะสรุปผลลัพธ์ที่ได้จากการปฏิบัติโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 Unmanned Aerial Vehicle (UAV)

ยูเอวี(UAV) คือหุ่นยนต์รูปแบบหนึ่งที่สามารถบินได้ สามารถทำงานได้โดยไม่มีคนบังคับ แต่ในเครื่องบิน จะมีความสามารถในการทรงตัว ประคองตัวเอง ให้ลอยตัวได้อย่างมีเสถียรภาพก่อนเป็นระบบแรก ส่วนระบบที่สองคือ สามารถกำหนดเส้นทางในการเคลื่อนที่ได้ และส่วนอื่นๆที่สามารถทำให้ยานบินมีประสิทธิภาพเพิ่มมากขึ้น

รูปที่ 2.1 ควอดโรเตอร์(Quadrotor)

ข้อดี คือ รูปแบบของเครื่องบิน แบบ สี่ใบพัดจะบินได้นิ่งมากกว่า เฮลิคอปเตอร์ใบพัดเดียว เพราะว่าใช้ใบพัด 4 ใบเป็นตัวสร้างแรงยก บางตัว 6-8 ใบ ทำให้มีแรงมากขึ้น จึงสามารถออกแบบให้ใช้ใบพัดที่เล็กกว่า เพราะว่ากระจายแรงยกไปใบต่างๆ เลยเป็นข้อดีอีกหลายอย่าง มีแรงสั่นน้อยกว่าใบพัดไม่ตีกับวัตถุอื่นๆ จึงเหมาะกับงาน ยูเอวี(UAV)

กลไก สร้างง่ายกว่า เพราะว่าถ้าเปรียบเทียบกับ เฮลิคอปเตอร์ปกติ จะใบพัดหลัก ชุดเดียว ใบพัดอาจต้องมีขนาดใหญ่ ตามหลักและ ถ้าของหนักๆ หมุนเร็วๆ จะเกิดแรงหนีศูนย์ได้ง่ายกว่าครับ ทำให้ตัวเฮลิคอปเตอร์แบบใบเดียว มีโอกาส สั่นได้มากกว่า

กลไกของใบพัดของ ควอดโรเตอร์(Quadrotor) ไม่ต้องมีกลไกซับซ้อน เฮลิคอปเตอร์จริงจะใช้วิธีปรับเอียง ในการสร้างแรงยก ในทิศทาง ทำให้มีกลไกที่ซับซ้อนกว่า แต่มันก้อเป็นความสวยงามทางการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย ระบบควบคุมของ ควอดโรเตอร์จะซับซ้อนมากกว่า เฮลิคอปเตอร์ในปัจจุบัน และต้องมีระบบเซ็นเซอร์ที่ดี ซึ่ง gyro /accelerometer มีความเป็นไปได้ในอนาคตว่าจะมีราคาถูกลง ดีขึ้น ปัญหาจึงเหลือแค่ระบบควบคุม

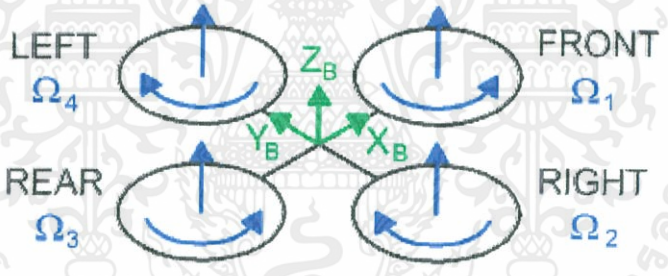
ปัญหาเรื่องแหล่งพลังงาน เครื่องบินแบบ ควอดโรเตอร์ ใส่มอเตอร์(Motor) 4 ตัวเป็นอย่างน้อย ต่อการบินหนึ่งครั้ง อย่างมากจะได้ 15 นาที

งานที่เอาควอดโรเตอร์ไปใช้ส่วนมากจะเป็นงานสำรวจ บินถ่ายรูปทางอากาศ กับ เป็น UAV(กำลังพัฒนา) ส่วนมากจะเป็นรูปแบบงานวิจัยที่เอาควอดโรเตอร์ไปประยุกต์ ไอเดียมากกว่า หรือเป็นงานอดิเรกก็ได้

2.1.1 หลักการการบินแบบควอดโรเตอร์(Quadrotor)

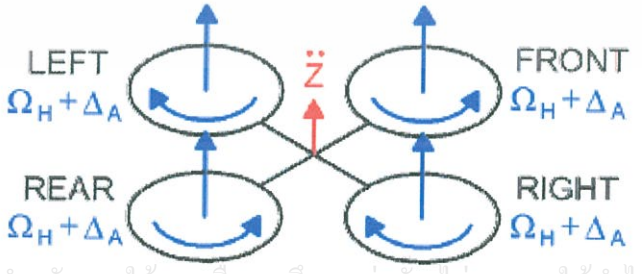
การบินของควอดโรเตอร์(Quadrotor) จะการเคลื่อนที่ 4 ทิศทาง หรือ เฮลิคอปเตอร์ จะเรียกว่า 4CH คือประกอบด้วย ขึ้น-ลง ,เดินหน้า-ถอยหลัง ,เอียงซ้าย-เอียงขวา และ หมุนซ้าย-หมุนขวา ซึ่งสามารถเคลื่อนที่ได้อย่างไรนั้น มีเทคนิคการควบคุมควอดโรเตอร์ ดังนี้

2.1.1.1 โฮเวอริง(Hovering) หรือ การลอยตัวเฉยๆ ทำได้โดยควบคุมให้ความเร็วใบพัดทั้งสี่ตัว มีความเร็วที่เท่ากัน เพื่อสร้างแรงบิด(Torque) และหักล้างแรงบิด ดูจากรูปจะเห็นว่าใบพัดจะหมุนกันคนล่ะทิศทาง ใบพัดหน้าและหลัง จะหมุน ตามเข็ม ใบพัด ซ้ายและขวาจะหมุนทวนเข็ม ทำให้เครื่องบินไม่หมุนตัว



รูปที่ 2.2 โฮเวอริง(Hovering)

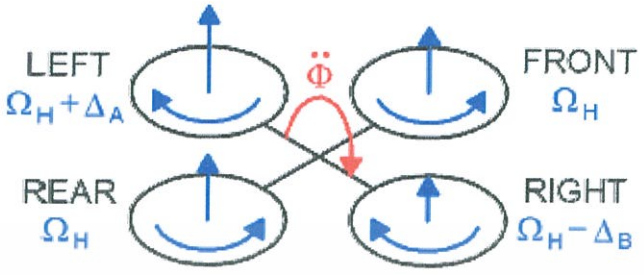
2.1.1.2 ทรอตเทิล(Throttle) คือ เร่ง ความเร็ว ให้เครื่องบิน บิน ขึ้นลง ดูจากรูปใบพัดทั้งสี่ใบจะต้องเพิ่มความเร็ว ทุกใบพัด ที่เท่ากัน ทำให้เครื่องบิน ลอยตัวขึ้นได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

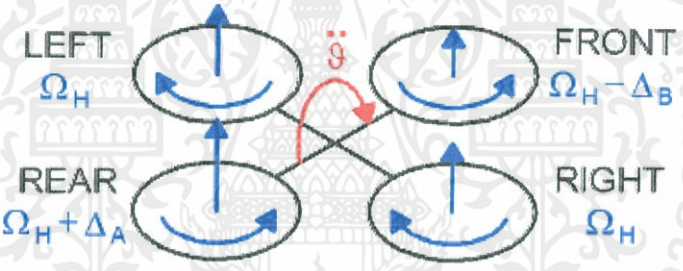
รูปที่ 2.3 ทรอตเทิล(Throttle)

2.1.1.3 **โรลล์(Roll) เอียงตัวซ้าย-ขวา** ดูจากรูป ใบพัด หน้า(FRONT) หลัง(REAR) จะความเร็วเท่าเดิม แต่ความเร็วใบพัดซ้าย (LEFT) จะหมุนเร็วขึ้น ทิศทางนี้จะยกตัว ใบพัดขวา (RIGHT) จะช้าลงทิศทางนี้จะตกลง จึงทำให้เกิดการ เอียงตัวไปทางขวาได้ ส่วนเอียงตัวซ้าย ก็จะใช้วิธีคล้ายกัน



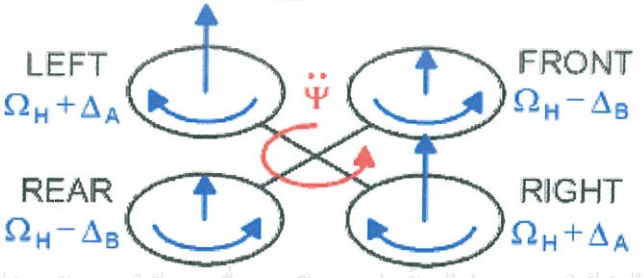
รูปที่ 2.4 โรลล์(Roll)

2.1.1.4 **พิทช์(Pitch) เอียงหน้าและหลัง** คล้ายกับการ Roll แต่เปลี่ยนเป็น ใบพัดซ้าย (LEFT) ขวา(RIGHT) จะความเร็วคงที่ แต่ความเร็วใบพัดหลัง(REAR) จะหมุนเร็วขึ้น ทางหลังจะยกใบพัดหน้า(FRONT) จะหมุนช้ากว่า ทางหน้าจะตก จึงทำให้เครื่องบินเอียงไปข้างหน้า



รูปที่ 2.5 พิทช์(Pitch)

2.1.1.5 **ยอว์(Yaw) หรือการหมุนตัว** ให้ความเร็วใบพัด หน้า(FRONT)-หลัง(REAR) มากกว่า ความเร็วใบพัด ซ้าย(LEFT) -ขวา(RIGHT) เพื่อให้แรงบิด ด้านซ้าย หรือ ขวา มากกว่า จึงทำให้เครื่องบินหมุนตัวได้



รูปที่ 2.6 ยอว์(Yaw)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าการควบคุมการเคลื่อนที่ต่างๆ จะเกิดจากการควบคุมความเร็วของใบพัดทั้งสี่ มาวิเคราะห์ อะไรบ้างที่จะทำให้มันบินไม่ได้ จะได้หาวิธีแก้ไขปัญหาไว้ก่อนเลย เป็นไปได้ว่า ถ้าความเร็วมอเตอร์ไม่ถูกต้อง จะเกิดการเคลื่อนที่แบบอื่นได้ทันที ระบบควบคุม หรือ การเซ็นเซอร์(Sensor) ความเอียงผิด ไม่ดี ก็มีโอกาสทำให้เครื่องบินควอดโรเตอร์ตกได้โดยง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 พีไอดี(PID)

สำหรับควอดโรเตอร์(Quadrotor) จริงๆมีมามาตั้งแต่ปี 50 กองทัพอากาศของอเมริกาทดลองทำไว้ แต่พึ่งจะมีทำมากขึ้น ในช่วง สิบปี เพราะของหลายๆอย่าง หาได้ง่ายแล้ว และเหลือจะมีราคาไม่แพง ซึ่งส่วนมากที่เห็นประยุกต์มาจาก อุปกรณ์ทำเครื่องบินจำลอง เครื่องบินบังคับวิทยุ ซึ่งในไทยในตอนนี้อุปกรณ์ก้อแพงแล้ว เทียบกับแต่ก่อนจะแพงมาก

อุปกรณ์ที่เหมือนกัน คือมอเตอร์(Motor) กับใบพัด 4 ใบ มอเตอร์ที่ใช้กันส่วนมากจะเป็น brushless และ สำหรับใบพัด จะมีลักษณะพิเศษ คือ ใบพัดจะมีสองชุด ชุดแรกเอียงซ้าย ชุดสองเอียงขวา ในประเทศไทย ได้มีการสำรวจ แล้วว่าสามารถหาได้ และใช้เซ็นเซอร์ไจโรมิเตอร์ *(Gyroscope) 3 แกน*

วิธีการควบคุมแบบพีไอดี(PID) ซึ่งเป็นส่วนหนึ่งของการควบคุมในระบบวงปิดหรือระบบควบคุมแบบป้อนกลับ (Closed-loop Control Systems, Feedback Control Systems) ซึ่งมักจะมีคนใช้งานอุปกรณ์ควบคุมอุณหภูมิสำหรับการตั้งค่า PID นั้นนอกจากนั้นยังใช้ในการควบคุมความสมดุลของเครื่องบินได้ดี เพื่อให้สามารถปรับตั้งค่า PID โดยใช้คำสั่งอย่างง่าย ความหมายของการควบคุมแบบ PID จะประกอบไปด้วยส่วนการควบคุมที่สำคัญด้วยกันคือ

P roportinal control action (P - Action)

I ntegral control action (I-Action)

D erivative control action (D-Action)

P Action : เป็นการกำหนดการทำงานของ output ให้เป็นสัดส่วนเปอร์เซ็นกับค่า error หรือการเปลี่ยนแปลงของค่าที่วัดได้

$$\text{Output} = (\text{error } X\ 100) / Pb\ ;\ \text{error} = (\text{ค่าset point}) - (\text{ค่าที่วัดได้}) \qquad (2.1)$$

ในทางปฏิบัติ P Action จะเข้าใกล้ค่าหนึ่ง ซึ่งไม่ใช่ค่า set point จริง ซึ่งเรียกว่าค่า offset ปรับค่า P น้อย เครื่องจะเอียงอย่างช้าๆ หากปรับค่า P สูงๆ เครื่องจะ Pitch หรือ Roll ไวมากจนกระพือ

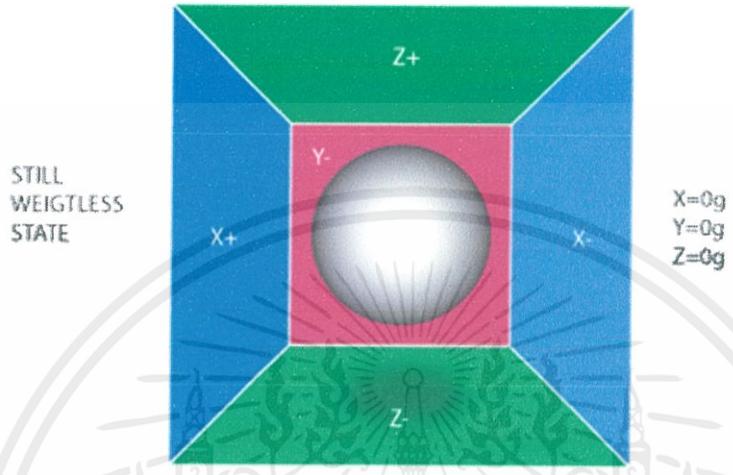
I Action : จะใช้ในการแก้ปัญหาออฟเซ็ท(Offset) ระบบควบคุม I Action จะเข้าไปช่วยกำจัดค่าออฟเซ็ทที่ยังคงมีอยู่ให้ระบบเข้าสู่เซ็ทพอยท์(Set Point)โดยค่าเอาท์พุท(Output) ที่ออกมาจะขึ้นอยู่กับอินทิกรัลไทม์(Integral Time) ที่กำหนดขึ้นมาตั้งแต่ต้น หากกำหนดให้อินทิกรัลไทม์น้อย ระบบจะเข้าสู่เซ็ทพอยท์ได้อย่างรวดเร็วแต่จะเกิดการกระเพื่อมของ โปรเซส(Process) มากด้วย และหากอินทิกรัลไทม์มากจะเกิดฮันติง(Hunting) น้อย แต่จะใช้เวลานานกว่าระบบจะเข้าสู่เซ็ทพอยท์การปรับค่า I น้อย เครื่องจะเอียงไวมากจนแกว่ง หากปรับค่า I สูง เครื่องจะเอียงค้างเมื่อเลี้ยวด้านใดด้านหนึ่ง

D Action : ในกรณีที่มีการรบกวนระบบจากภายนอก ดิสเทอร์แบนซ์(Disturbance) เป็นผลให้โปรเซส ของระบบมีการเปลี่ยนแปลงอย่างทันที่ทันใด เราควรจะใช้การควบคุมแบบ D Action Derivative ซึ่งจะมีการตอบสนองที่รวดเร็ว เป็นผลให้ระบบเข้าสู่เซ็ทพอยท์ได้รวดเร็วขึ้นการปรับค่า D น้อย จะทำให้เอียงเร็วมาก หากค่า D เยอะเกินไปจะฝืด หากใช้ความเร็วสูงเครื่องจะคว่ำได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

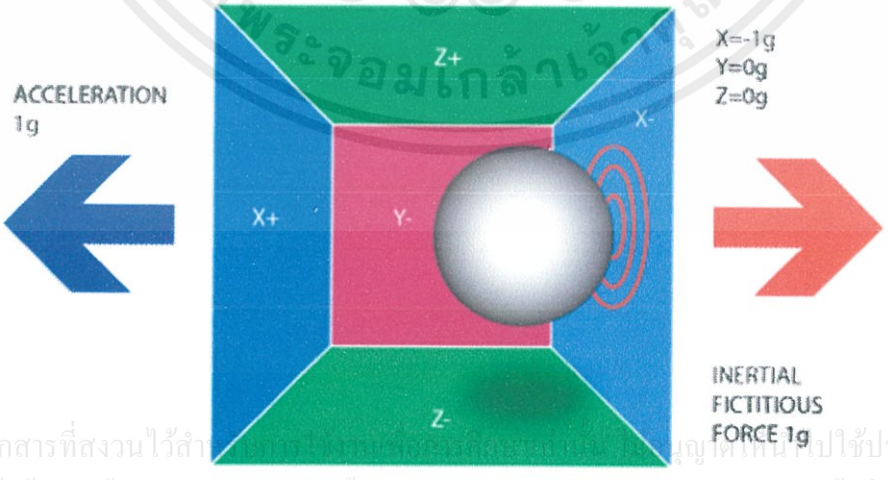
2.3 Accelerometer

หลักการทำงานของ accelerometer ให้จินตนาการถึงกล่องลูกบาศก์ที่มีวัตถุลอยอยู่ใจกลางกล่อง ดังแสดงในภาพ



รูป 2.7 จำลองสภาวะนิ่ง

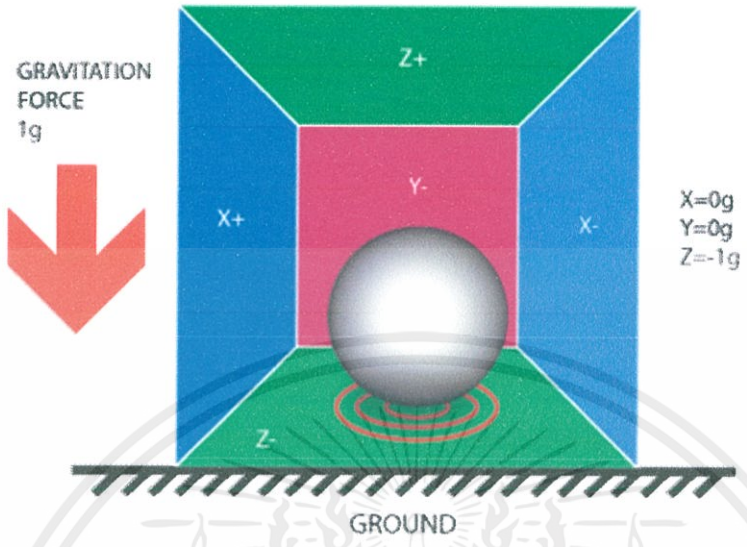
ให้จินตนาการว่ากล่องนั้นอยู่ในสภาพไร้แรงโน้มถ่วง เป็นผลให้ลูกบอลลอยอยู่ใจกลางกล่อง จากภาพข้างบนนั้น จะสังเกตเห็นได้ว่ามีการกำหนดแกน X,Y และ Z ไว้ที่ผนังของกล่องโดยให้ผนังกล่องนั้นมีความไวสูงต่อแรงตกกระทบ ถ้าหากเราเลื่อนกล่องไปทางซ้ายในทันทีทันใด (กำหนดให้เคลื่อนที่ด้วยความเร่ง 1 g = 9.8 m/s^2) ลูกบอลจะกระทบกับผนัง X- ค่าที่วัดได้จากแรงกระทบคือ –1g บนแกน X ดังภาพต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 จำลองสภาวะเคลื่อนที่ไปทางซ้าย

จะเห็นได้ว่ามีการอ่านค่าออกมาให้เห็น แต่นี่เป็นเพียงการยกตัวอย่างเท่านั้นเพราะในความเป็นจริงแล้วจะมีแรงโน้มถ่วงเข้ามาเกี่ยวข้องดังจะเห็นได้ในภาพต่อไปนี้

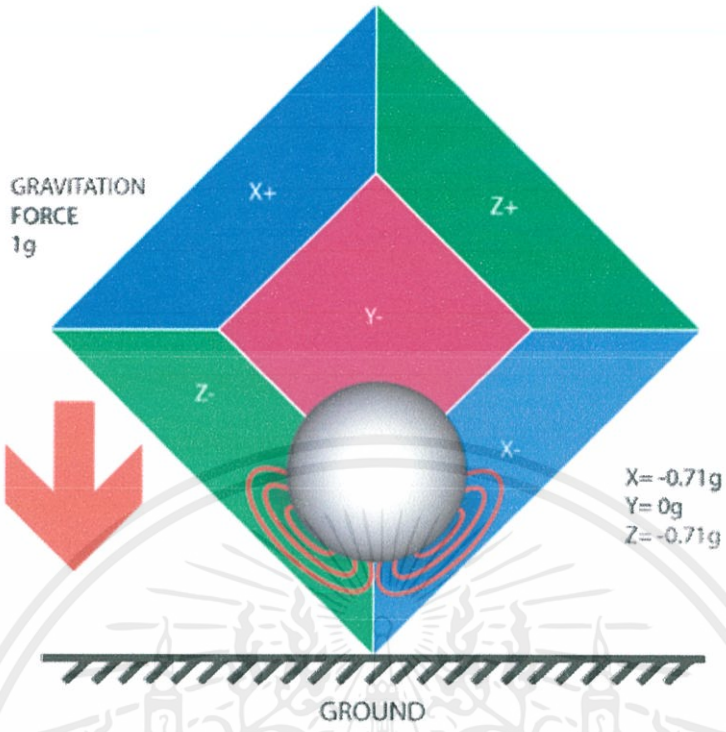


รูป 2.9 สภาวะแรงโน้มถ่วง

ในกรณีนี้ กล่องไม่ได้เคลื่อนที่แต่มีการอ่านค่าออกมาคือ -1g บนแกน Z แรงที่มากระทำต่อลูกบอลเกิดจากแรงโน้มถ่วงของโลก โดยนิยามว่าแรงที่เกิดขึ้นนั้นมิได้เกิดขึ้นจากความเร่งที่มากระทำแต่เกิดจากแรงโน้มถ่วง

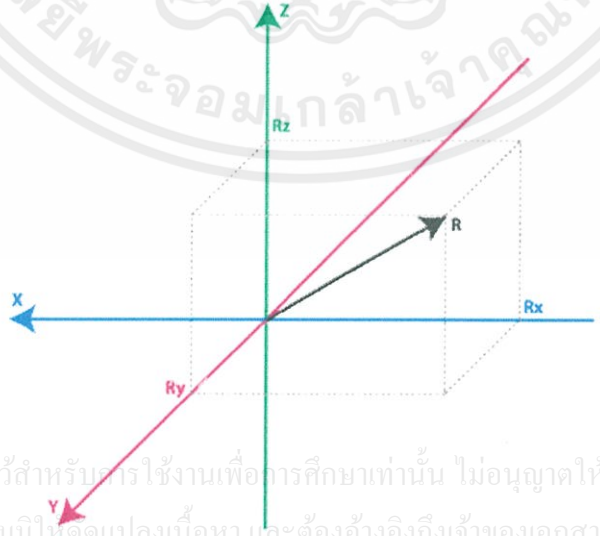
ค่าที่แท้จริงอาจวัดได้จากหลายแกนในช่วงเวลาเดียวกัน อาจมากถึง 3 แกน จากภาพต่อไปนี้จะเป็นการทำให้กล่องเอียงไปตามเข็มนาฬิกา 45 องศา ลูกบอลจะกระทบกับผนัง Z- และ X-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.10 สภาวะเคลื่อนที่ในกรณีสองแนวแรง

ปริมาณ 0.71 เกิดจากการประมาณค่าสำหรับ SQRT(1/2) หลักการนี้จะช่วยให้ภาพต่อไปนั้นเข้าใจได้ง่ายขึ้นสำหรับการอธิบายหลักการของ accelerometer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 ลูกศรแสดงทิศของแนวแรง

จากภาพลูกศร R คือ เวกเตอร์แรงที่ accelerometer วัดออกมาได้โดยจะประกอบด้วย Rx,Ry,Rz บน แกน X,Y,Z และจะให้หลักการพื้นฐานทางคณิตศาสตร์ 3มิติ ของ Pythagoras คือ

$$R^2 = Rx^2 + Ry^2 + Rz^2 \qquad (2.2)$$

หากรวมสมการ SQRT(1/2) = 0.71 เข้ากับ สมการ (1) โดยมีค่าแรงโน้มถ่วงคือ 1g จะได้เป็น $1^2 = (-SQRT(1/2))^2 + 0^2 + (-SQRT(1/2))^2$ ได้คำตอบคือ R = 1, Rx = -SQRT(1/2), Ry = 0 , Rz = -SQRT(1/2)

ส่วนมากแล้ว accelerometer นั้นจะเป็น analog จึงจำเป็นที่จะต้องใช้ ADC (analog to digital converter)

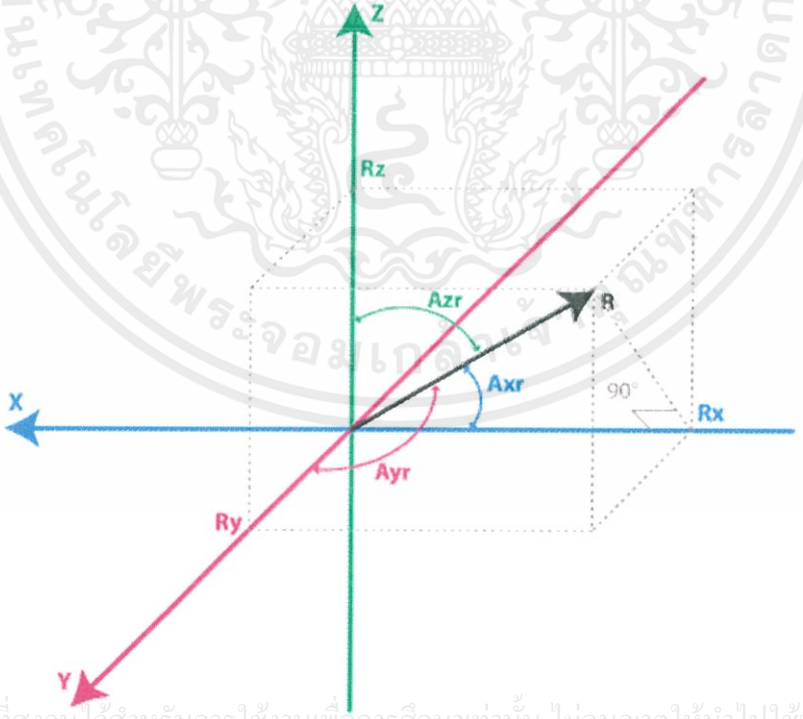
โดยในที่นี้จะยกตัวอย่างเป็น 10-bit ADC คือจะให้ค่า output ในช่วง 2^{10}-1 = 0 ถึง 1023

สมการที่ใช้สำหรับการอ่านค่า ADC

$$Rx = (AdcRx * Vref / 1023 - VzeroG) / Sensitivity \qquad (2.3)$$
$$Ry = (AdcRy * Vref / 1023 - VzeroG) / Sensitivity$$
$$Rz = (AdcRz * Vref / 1023 - VzeroG) / Sensitivity$$

โดยที่ AdcR...คือ ค่าที่วัดได้จากโมดูล accelerator ตามแกนนั้น Vre แรงดันที่ใช้ในการทำค่า AdcR ให้เป็น VzeroG แรงดันที่มีค่าเท่ากับ 0 g พอดีขึ้นอยู่กับชนิดของอุปกรณ์ Sensitivity ขึ้นอยู่กับชนิดของอุปกรณ์

เมื่อได้รับค่าจากอุปกรณ์แล้วต่อไปจะเป็นการคำนวณมุมของเวกเตอร์ R ดังภาพต่อไปนี้



รูป 2.12 มุมของแนวแรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพกำหนดมุมระหว่างแกน X,Y,Z กับ เวกเตอร์ R เป็น Axr,Ayr,Azr และใช้ทฤษฎีมุมภายในรูปสามเหลี่ยมในการอธิบายคือ

$$\cos(Axr) = \cos X = Rx / R \quad (2.4)$$
$$\cos(Ayr) = \cos Y = Ry / R$$
$$\cos(Azr) = \cos Z = Rz / R$$

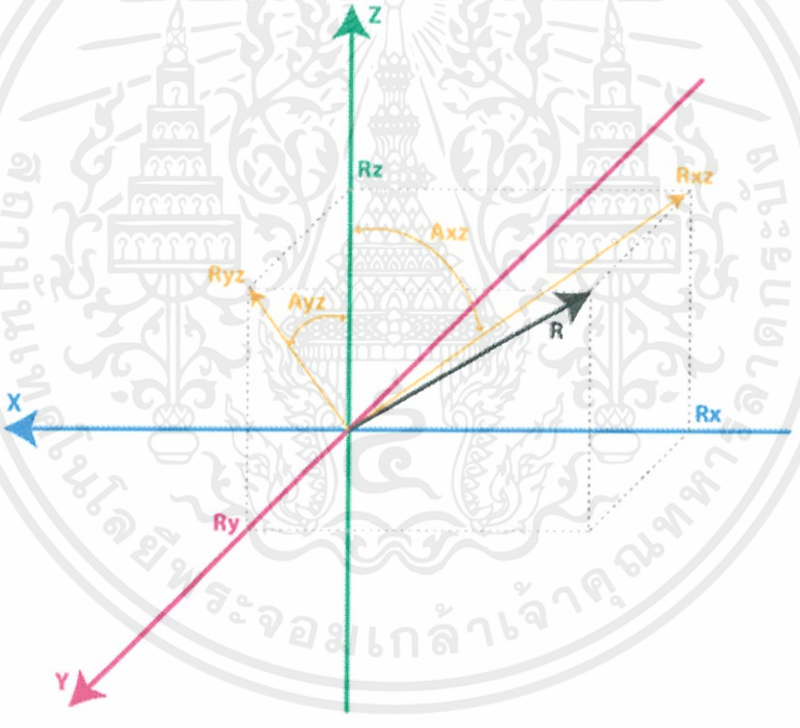
โดยจากสมการที่ (1) $R = SQRT(Rx^2 + Ry^2 + Rz^2)$

และจะสามารถหามุมได้โดยใช้หลักการของฟังก์ชัน arcos( ) (ส่วนกลับของฟังก์ชัน cos( )) คือ

$$Axr = \arccos(Rx/R) \quad (2.5)$$
$$Ayr = \arccos(Ry/R)$$
$$Azr = \arccos(Rz/R)$$

2.4 Gyroscope

เราจะข้ามเนื้อหาของส่วนแบบจำลองเพื่อเข้าสู่เวกเตอร์ตามภาพต่อไปนี้



รูป 2.13 แสดงอัตรามุมที่เปลี่ยนไป

Gyroscope แต่ละตัวนั้นจะทำหน้าที่วัดการหมุนรอบแกน X,Y,Z โดยจะเห็นว่า Axz เป็นมุมระหว่าง Rxz และแกน Z ต่อไปจะเป็นการคำนวณอัตรการเปลี่ยนแปลงโดยใช้สมการดังต่อไปนี้

$$RateAxz = (Axz1 - Axz0) / (t1 - t0) \quad deg/s \quad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Axz0 คือ มุม ณ เวลาก่อนเกิดการหมุน

Axz1 คือ มุม ณ เวลานั้นๆหลังเกิดการหมุนแล้ว

t0 คือ เวลาเริ่มต้น

t1 คือ เวลานั้นๆที่เกิดการหมุนแล้ว

จากสมการที่ได้เมื่อนำมาประยุกต์กับสมการที่ (2.3) จะได้เป็นสมการค่า gyroscope สำหรับ 10-bit ADC คือ

$$RateAxz = (AdcGyroXZ * Vref / 1023 - VzeroRate) / Sensitivity \quad (2.6)$$

โดย AdcGyroXZ คือ ค่าที่โมดูล gyroscope ของแกน Y ส่งมาให้

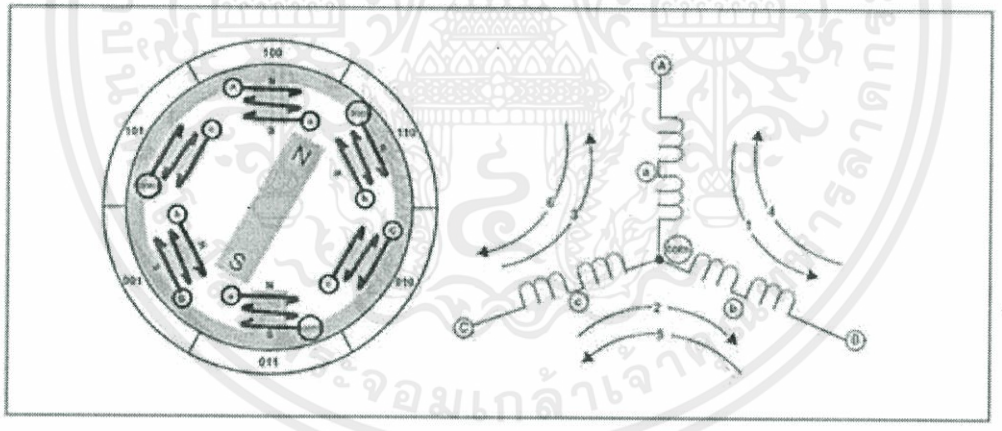
Vref คือ แรงดันที่ใช้ในการทำค่า AdcGyro ให้เป็น Volt

VzeroRate คือ แรงดันณจุดที่ไม่เกิดการหมุนใดๆ

Sensitivity คือ ความตอบสนอง ขึ้นอยู่กับชนิดของอุปกรณ์

2.5 การเลือกใช้มอเตอร์

มอเตอร์ไร้แปลงถ่าน (Brushless DC Motor)การทำงานของมอเตอร์ไร้แปลงถ่านจะมีขดลวดอยู่ 3 ชุด และจะทำงานครั้งละ 2 ชุดสับเปลี่ยนกันเพื่อสร้างสนามแม่เหล็กที่เหมาะสม จะเห็นได้ว่ามอเตอร์ไร้แปลงถ่านจะมีสายจำนวน3 เส้นต่อออกมา ดังภาพที่ 2.14 คือ ขั้ว A, B และ C



รูปที่ 2.14 แสดงการทำงานของมอเตอร์ไร้แปลงถ่าน

จากภาพที่ 2.14 ด้านขวาแสดงภาคตัดมอเตอร์ จะเห็นตัวหมุน (rotor) ที่เป็นแม่เหล็กถาวรอยู่ตรงกลาง และขวดลวด A, B และ C อยู่รอบ ๆ แต่หากเป็นมอเตอร์ที่เปลือกหมุน (out-runner) จะ

กลับกันคือขดลวดจะอยู่ตรงกลางแล้วมีเปลือกแม่เหล็กเป็นวงกลมอยู่รอบนอกแทน จะมีกระแสไหลผ่านขดลวดจาก A ไป B ตามเส้นทางที่ 1 ทำให้เกิดสนามแม่เหล็กสำหรับดึงดูดแกนหมุนให้อยู่ระหว่างขั้วของ A และ B หรือมอเตอร์จะวางตัวอยู่ และหากให้การกระแสไหลตามเส้นทางที่ 6 คือ จากขั้ว A ไปยังขั้ว C จะทำให้เกินสนามแม่เหล็กระหว่างขั้ว A และ C จะทำให้ตัวหมุนเคลื่อนไปในทิศทางทวนเข็มนาฬิกาไปอยู่ระหว่างขั้ว C และ A แทน หรือขั้ว N และหากเปลี่ยนกระแสให้ไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ในเส้นทางที่ 5 คือ จาก B ไปยัง C ต่อไปอีก ตัวหมุนจะหมุนทวนเข็มนาฬิกาต่อไปอีก ขั้ว N จะอยู่ที่ระหว่างขั้ว B และ C หรือที่ประมาณ 9 นาฬิกา ดังนั้นหากให้เกิดการหมุนต่อเนื่องจะต้องทำการสับขั้วให้กระแสไหลในจังหวะที่เหมาะสมตามลำดับ

ในโครงงานนี้ได้ใช้มอเตอร์ Brushless ของ DJI INNOVATION รุ่น DJI 2212-920, ใช้ไฟเลี้ยง 11.1V-14.9V (3-4 เซลล์ Lipo), 920 KV, ขนาด 12*22 มม. สามารถคำนวณความเร็วได้ดังนี้

11.1V*920KV = 10,212 RPM

รูปที่ 2.15 มอเตอร์ Brushless ของ DJI INNOVATION

2.6 การเลือกใช้ชุดควบคุมความเร็ว

ควอดโรเตอร์ จะใช้ชุดควบคุมความเร็ว(ESC)ทั้งหมด 4 ตัว มีคุณสมบัติที่เหมือนกันทั้ง 4 ตัว คุณสมบัติของชุดควบคุมความเร็วที่เลือกใช้โดย พิจารณาดังต่อไปนี้

ชุดควบคุมความเร็วทนกระแส 30 A เลือกใช้ชุดควบคุมความเร็วขนาดเท่านี้เพราะว่า มอเตอร์กินกระแสไม่เกิน 30 A เพราะถ้าใช้ต่ำ กว่านี้จะทำให้ชุดควบคุมความเร็วอาจไหม้ได้ โดยชุด ควบคุมความเร็ว จะต่อเข้ากับชุดควบคุม ที่ขาไอซี ชุดควบคุมความเร็วตัวที่ 1 จะต่อที่ขา PIN3 ชุดควบคุมความเร็วตัวที่ 2 จะต่อที่ขา PIN9 ชุด ควบคุมความเร็วตัวที่ 3 จะต่อที่ขา PIN10 ชุด ควบคุมความเร็วตัวที่ 4 จะต่อที่ขา PIN11 ส่วนอีก 2 เส้นจะเป็นสายไฟ 5 โวลต์ และกราวด์

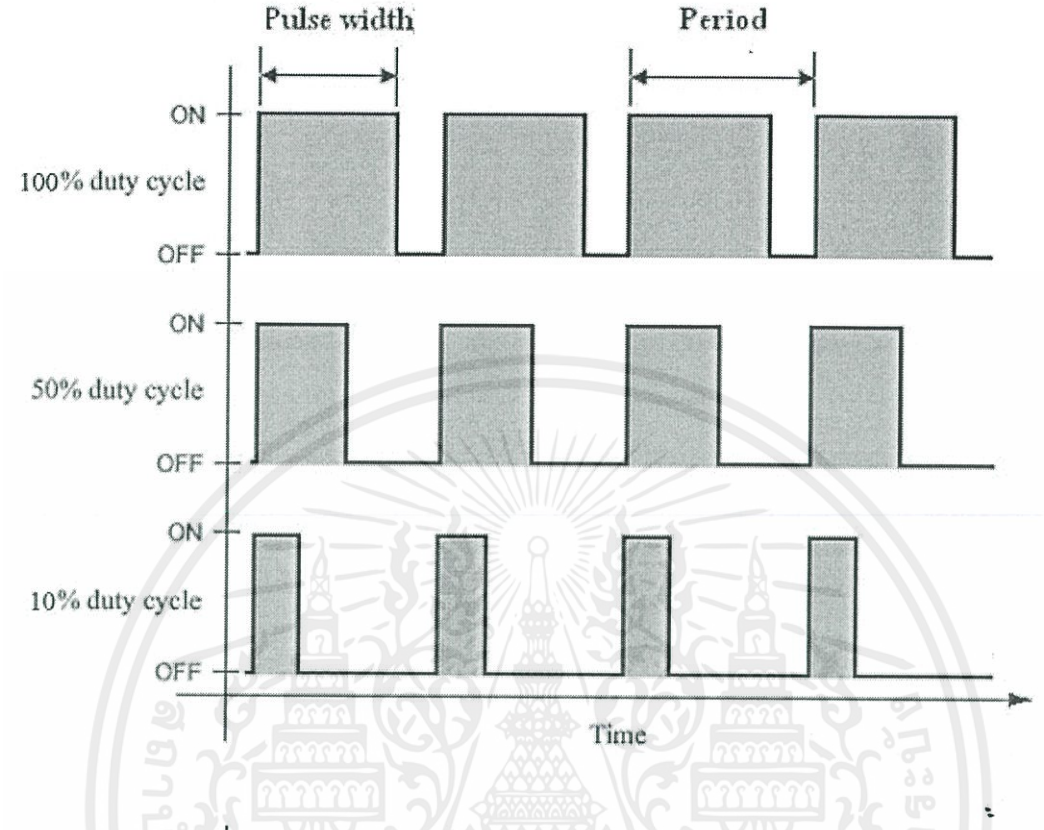
ในโครงงานนี้ได้ใช้ ชุดควบคุมความเร็วของ DJI รุ่น OPTO ,ใช้ไฟเลี้ยง 3-4 เซลล์ Lipo ,ขนาด 30 Amp, ความถี่ 30-450 Hz

รูปที่ 2.16 ชุดควบคุมความเร็ว ของ DJI INNOVATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การใช้งานสัญญาณ PWM ในการควบคุม มอเตอร์บรัชเลส

PWM (Pulse Width Modulation) เป็นวิธีหนึ่งที่นิยม ใช้กันมากในงานควบคุม เช่นการควบคุมความเร็ว มอเตอร์ หรือการควบคุมทิศทางการหมุนของเซอร์ โวมอเตอร์



รูปที่ 2.17 แสดงความกว้างของสัญญาณพัลส์ สัมพันธ์กับความเร็วในแบบต่างๆ

จากรูปที่ 2.6.2 จะเห็นได้ว่าหลักการของวิธี นี้คือกระแสที่ป้อนเข้ามอเตอร์จะเป็นสัญญาณพัลส์ที่มีความถี่คงที่ แต่ความกว้างของสัญญาณพัลส์เปลี่ยนแปลงได้ถ้าสัญญาณพัลส์แคบความเร็วจะต่ำ จากรูป จะเห็นได้ว่าหลักการของวิธี นี้คือกระแสที่ป้อนเข้ามอเตอร์จะเป็นสัญญาณพัลส์ ที่มีความถี่คงที่ แต่ความกว้างของสัญญาณพัลส์เปลี่ยนแปลงได้ถ้าสัญญาณพัลส์แคบความเร็วจะต่ำ แต่ถ้าพัลส์กว้างความเร็วจะสูง ลักษณะของสัญญาณพัลส์ที่มีความกว้างของสัญญาณพัลส์เป็น 100% 50% และ 10% ของแต่ละค่าคาบเวลาจะทำ ให้ความเร็วของมอเตอร์นั้นหมุนต่างกัน ความกว้าง ของสัญญาณพัลส์ที่ 100 % จะทำ ให้มอเตอร์หมุนเร็วมากที่สุด ส่วนความกว้างของสัญญาณพัลส์ที่ 10% จะทำให้มอเตอร์หมุนช้าที่สุด

2.7 แบตเตอรี่ลิเทียมโพลิเมอร์ (Lithium-ion polymer battery)

เป็นแบตเตอรี่ชนิดใหม่ พัฒนามาจากเทคโนโลยีของแบตเตอรี่ลิเทียมไอออน ทำจากธาตุลิเทียม (li น้ำหนักต่อ 1 ลบ.ซม.) จึงมีน้ำหนักเบา มีพลังไฟฟ้าสูงเมื่อเทียบกับน้ำหนัก ตัวแบตเตอรี่เป็นรูปทรงสี่เหลี่ยมแบน มีเปลือกเป็นถุงอลูมิเนียมและยังเป็นมิตรกับสิ่งแวดล้อม ข้อดีคือสามารถเก็บพลังงานไฟฟ้าได้มากกว่าแบบ Nickel Cadmium (NiCd) ถึง 350 %และน้ำหนักที่น้อยกว่า 10 - 20 % โดยประมาณ และอัตราของแรงดันไฟฟ้าที่ลดลงเองโดยที่เราไม่ได้ใช้งาน (Self Discharge) อยู่ที่ประมาณ 5% แต่ข้อเสียของมันคือห้ามใช้จนแรงดันไฟฟ้าต่ำกว่า 3.0 V เพราะอาจเสียหายได้ แบตเตอรี่ lipo แต่ละก้อนโดยปกติจะระบุ spec ไว้อย่างชัดเจน เช่น จำนวน V , mAh และ C เอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติแล้วจะมีแรงดันไฟฟ้าที่ 3.7 V เมื่อชาร์จจนเต็มจะมีแรงดันเท่ากับ 4.23 V สามารถใช้งานได้ 500 - 600 ครั้ง

ในโครงงานนี้ได้ใช้แบตเตอรี่ลิเทียมโพลิเมอร์ ของ EVEREST รุ่น 3S1P ,ความจุ 2200 mAh, แรงดันไฟฟ้า 11.1V ,3 เซลล์, ค่าคายประจุคงที่ 30C ดังนั้นจึงสามารถจ่ายประจุได้มากถึง

$$2200\ mAh * 30C / 1000 = 66\ Amps$$

รูปที่ 2.18 แบตเตอรี่ Li-po ของ Everest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการและขั้นตอนการดำเนินงาน

ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบเครื่องต้นแบบอากาศยานไร้คนขับ ซึ่งการออกแบบจะต้องอาศัยทฤษฎีในบทที่ 2 มาใช้ในการออกแบบเพื่อให้อากาศยานมีคุณสมบัติตามที่ต้องการ หลังจากนั้นจัดเตรียมอุปกรณ์ที่ได้จากการออกแบบ การประกอบอากาศยาน และการทดสอบเครื่องต้นแบบอากาศยานไร้นักบินโดยขั้นตอนการดำเนินงานสามารถแบ่งขั้นตอนได้ดังนี้

3.1 การออกแบบเครื่องต้นแบบอากาศยานไร้นักบิน

3.2 การเลือกไมโครคอนโทรลเลอร์

3.3 บล็อกไดอะแกรมการทำงานของ Software

3.4 การจัดเตรียมอุปกรณ์

3.5 ออกแบบลายวงจรพิมพ์และสร้างแผ่นวงจรพิมพ์

3.6 การดูข้อมูลผ่านซีเรียลพอร์ท

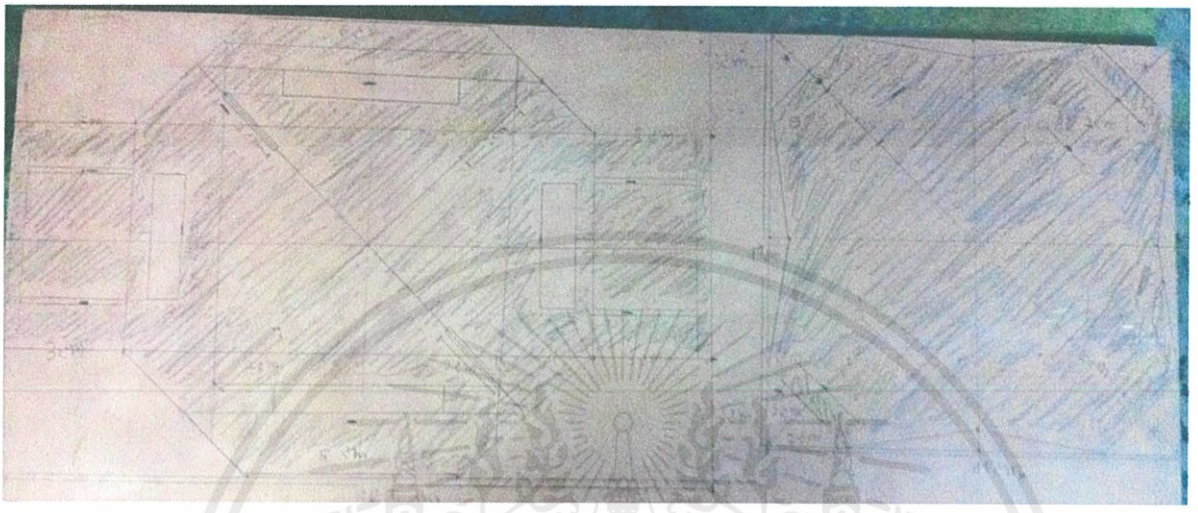
3.7 การใช้การประมวลผลภาพเข้ามาช่วยเพื่อการบินติดตามเป้าหมายอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การออกแบบเครื่องต้นแบบอากาศยานไร้นักบิน

3.1.1 ส่วนลำตัวของอากาศยาน (Body Frame)

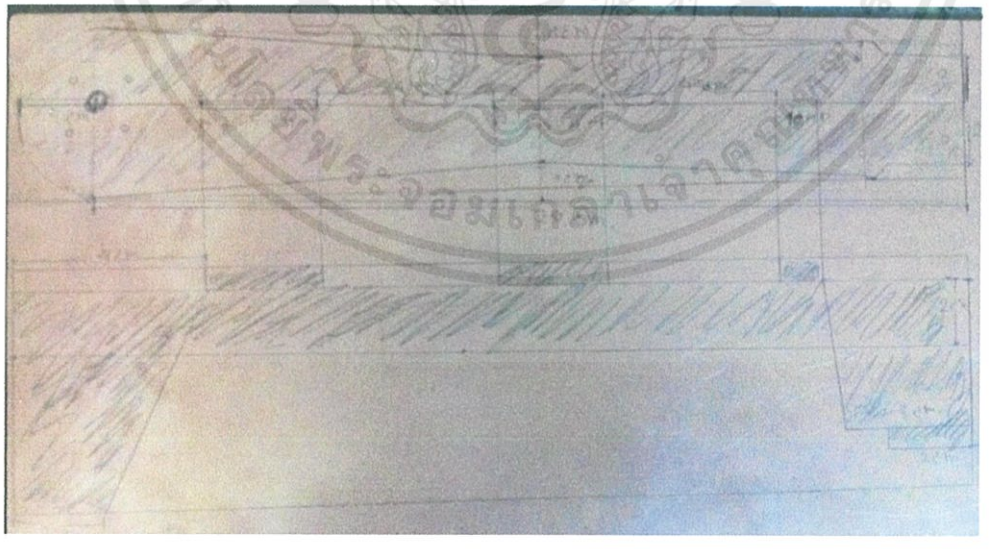
ทำจากแผ่นPCB ชนิด Epoxy เพื่อความทนทานและมีความยืดหยุ่นสูงกว่าอลูมิเนียมเพื่อป้องกันปัญหาจากการตกแล้วเสียรูปทรง



รูปที่ 3.1 การออกแบบส่วนลำตัว

3.1.2 ส่วนแขนของอากาศยาน (Arms)

ทำจากแผ่น PCB ชนิด Epoxy เพื่อความทนทานและมีความยืดหยุ่นสูงกว่าอลูมิเนียมเพื่อป้องกันปัญหาจากการตกแล้วเสียรูปทรง โดยทำทั้งหมด 4 ชุด

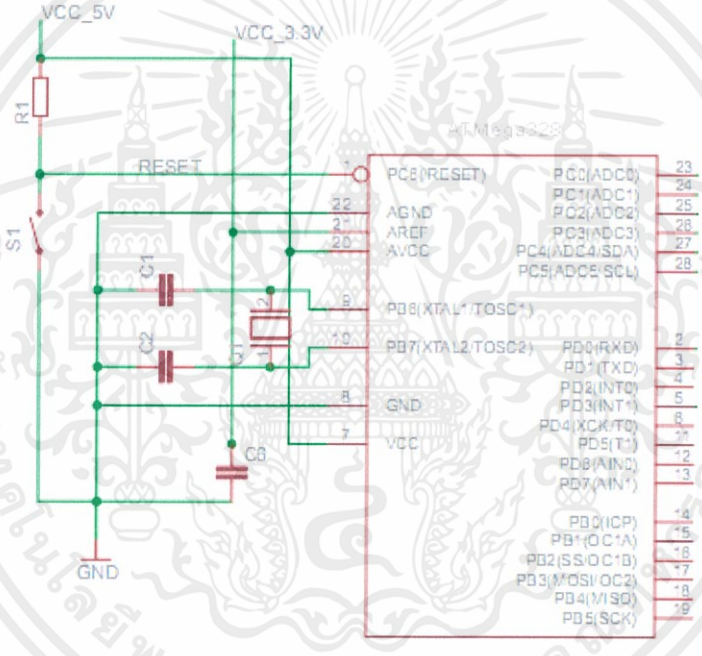


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 การออกแบบส่วนแขน

3.2 การเลือกไมโครคอนโทรลเลอร์

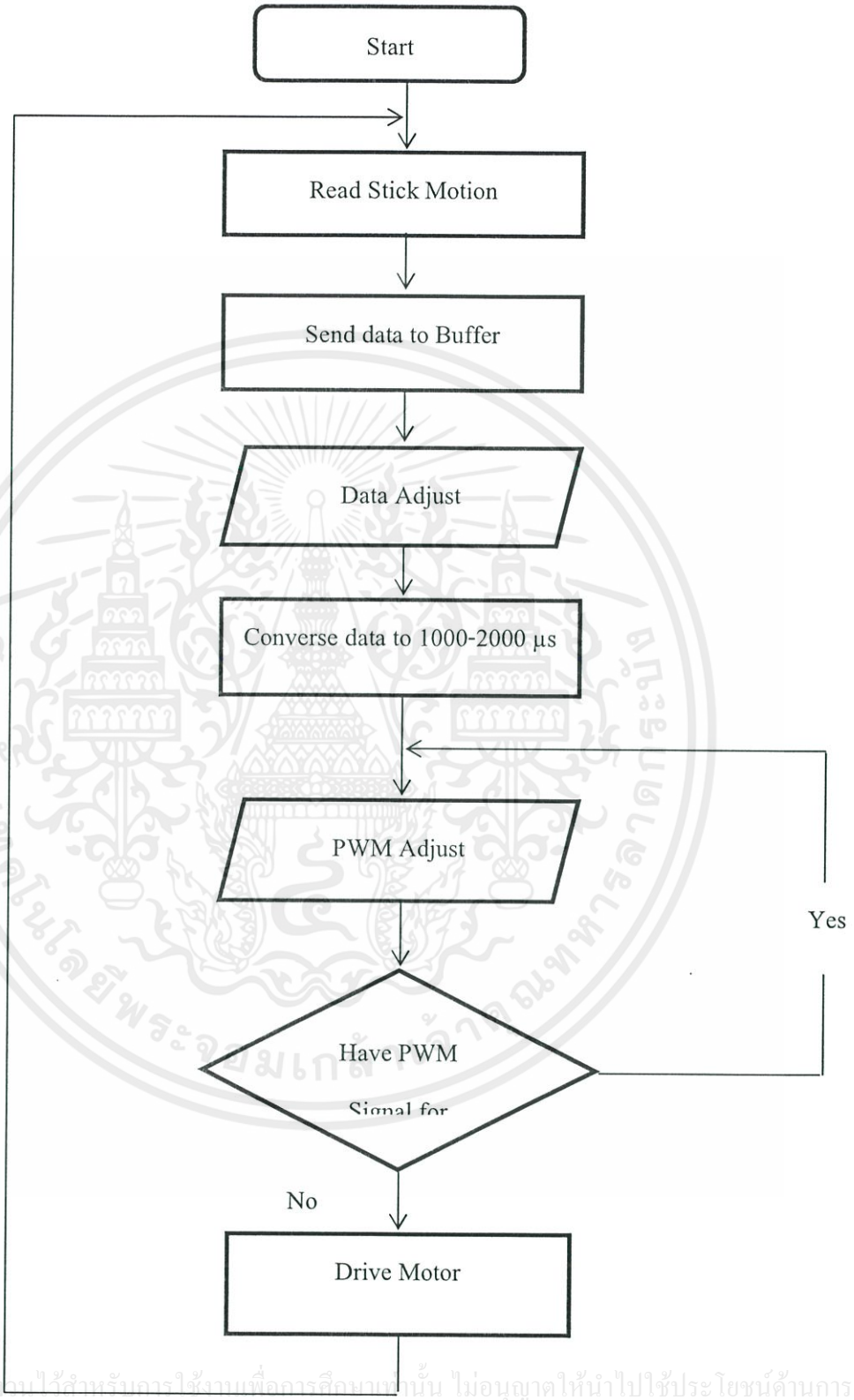
ในโปรเจคนี้จะใช้ไมโครคอนโทรลเลอร์Multiwii SE V2.0 เบอร์ ATMega 328P เนื่องจากมีฟังชั่นที่สามารถรองรับการทำงานของหุ่นอากาศยานไร้คนขับได้ครบทุกฟังก์ชั่น โครงสร้างของไมโครคอนโทรลเลอร์รุ่นนี้มีโมดูล Pulse Width Mode : PWM จำนวน 6 ชุด คือ 0C0A, 0C1A, 0C2A, 0C0B, 0C1B และ 0C2B โดยใช้การเชื่อมต่อกับพอร์ตอนุกรม เพื่อทำการอ่านข้อมูลและประมวลผลคำสั่งไปยังอุปกรณ์ต่าง ๆ โดยที่ไมโครคอนโทรลเลอร์นั้นจะต้องมีวงจรพื้นฐานที่ต้องต่อทุกครั้งเพื่อให้ทำงานตามคำสั่งที่เราต้องการและทำงานอย่างมีประสิทธิภาพ ส่วนพอร์ทที่เหลือจะใช้เชื่อมต่อกับอุปกรณ์ภายนอกเพื่อใช้งานตามต้องการ ซึ่งข้อมูลต่าง ๆ สามารถหาได้จาก Data Sheet ของไมโครคอนโทรลเลอร์



รูปที่ 3.3 ไอซีเบอร์ ATMega 328P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โฟลวชาร์ทการอ่านค่าและสั่งงานมอเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 โฟลวชาร์ทการอ่านค่าและสั่งงานมอเตอร์

3.4 การจัดเตรียมอุปกรณ์

3.4.1 การทำในส่วนโครงหุ่นยนต์

3.4.1.1 ตัดโครงส่วนแขน (Arms) ตามแบบที่เขียนไว้แล้วประกอบเข้าด้วยกัน

รูปที่ 3.5 แขนอากาศยาน

3.4.1.2 ทำเช่นเดียวกันกับอีก 3 แขนที่เหลือ

รูปที่ 3.6 แขนอากาศยานทั้งสี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.3 ประกอบแขนทั้ง 4 เข้ากับลำตัวส่วนล่าง

รูปที่ 3.7 แขนและลำตัวอากาศยาน

3.4.1.4 เชื่อมตะกั่วตามรอยต่อเพื่อความทนทาน

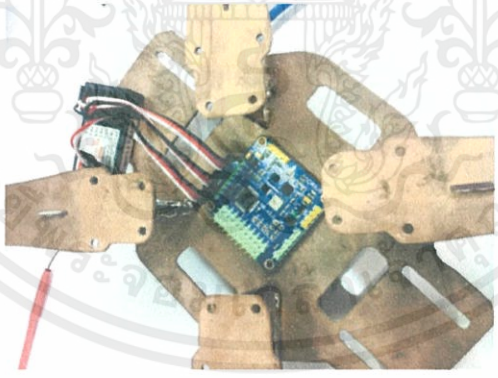
รูปที่ 3.8 การเชื่อมต่อลำตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การติดตั้งมอเตอร์,ใบพัดและ ESC

รูปที่ 3.9 การติดตั้ง ESC มอเตอร์ และ ใบพัด

3.4.3 การติดตั้งบอร์ดควบคุมเข้ากับลำตัว

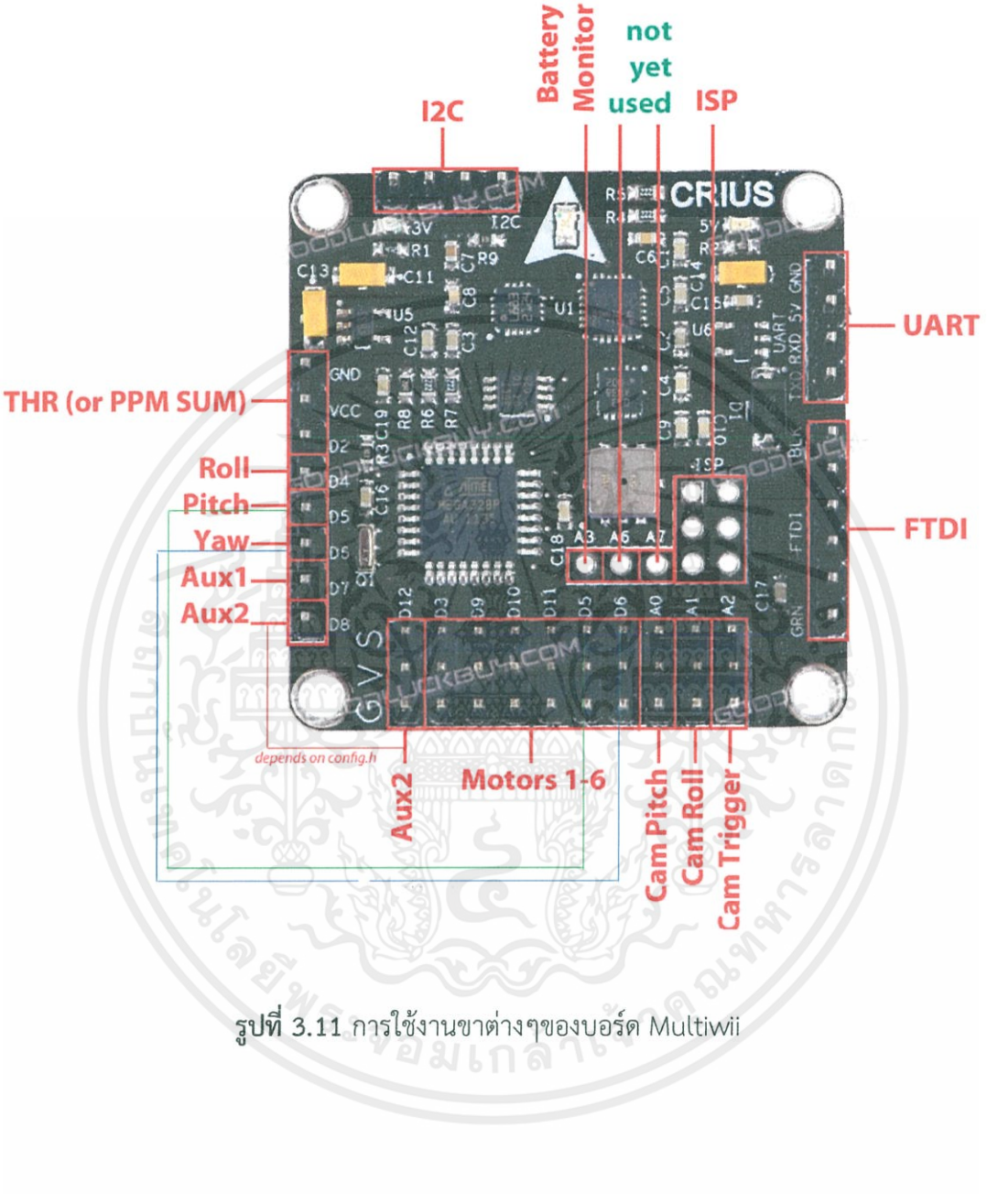


รูปที่ 3.10 การติดตั้งบอร์ดบนลำตัวยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การเชื่อมต่ออุปกรณ์ทั้งหมดเข้าสู่บอร์ดควบคุม

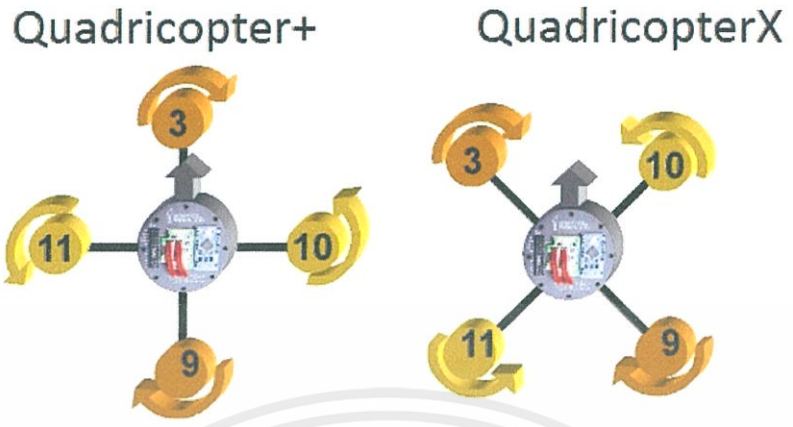
ใช้รูปภาพดังต่อไปนี้อธิบายเพิ่มเติมเพื่อให้ง่ายต่อการเชื่อมต่อ



รูปที่ 3.11 การใช้งานขาต่างๆของบอร์ด Multiwii

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยอากาศยานสี่ใบพัดจะให้การเชื่อมต่อมอเตอร์ทั้งสี่ดังภาพต่อไปนี้



รูปที่ 3.12 การเชื่อมต่อมอเตอร์กับขาของบอร์ด

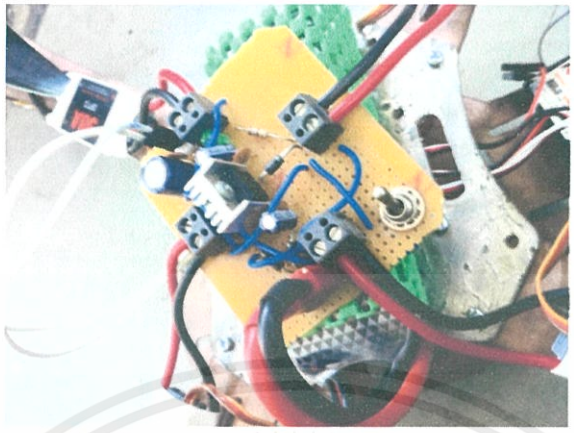
ตารางที่ 3.1 สรุปการใช้งาน Pin บอร์ด Multiwii

ลักษณะ	ตำแหน่ง Pin ของ Board	อุปกรณ์
Throttle	D2	Receiver CH5
Roll	D4	Receiver CH1
Pitch	D5	Receiver CH2
Yaw	D6	Receiver CH4
Aux 1	D7	Receiver CH6
Motor Front	D3	ESC-Front
Motor Rear	D9	ESC-Rear
Motor Right	D10	ESC-Right
Motor Left	D11	ESC-Left
VCC	VCC	Vcpu- Regulator Board
GND	GND	Gnd- Regulator Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 เชื่อมต่อบอร์ดจ่ายไฟและแบตเตอรี่

มี port 5V จ่ายไฟเลี้ยงบอร์ดไมโคร และ 4 port 11.1V จ่ายไฟเลี้ยงมอเตอร์

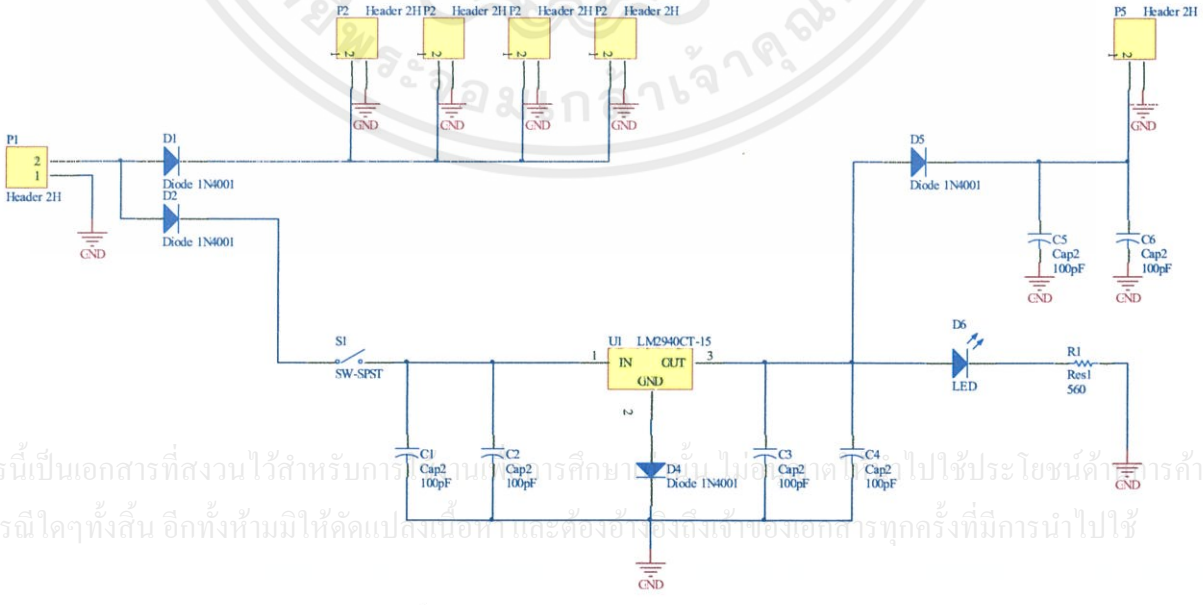


รูปที่ 3.13 บอร์ดจ่ายไฟเลี้ยง

3.4.6 ติดตั้ง code ลงบอร์ด ติดตั้งผ่าน โปรแกรม Arduino 1.0.2 โดยใช้ code ต้นแบบของ Mega 2.0

3.5 ออกแบบลายวงจรพิมพ์และสร้างแผ่นวงจรพิมพ์

ส่วนจ่ายไฟให้กับหุ่นยนต์ ส่วนนี้เป็นส่วนที่รับ Input จากแหล่งกำเนิด และนำออกไปจ่ายให้กับส่วนต่างๆของหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถทำงานได้ตามที่ออกแบบไว้ โดยวงจรส่วนนี้จะนำไฟออกไปจ่ายให้กับวงจรส่วนควบคุม ซึ่งมีไมโครคอนโทรลเลอร์บอร์ด Mutiwii V2.0 เป็นส่วนประกอบหลักของวงจร และตัวควบคุมความเร็วมอเตอร์



รูปที่ 3.14 ลายวงจรส่วนจ่ายไฟ

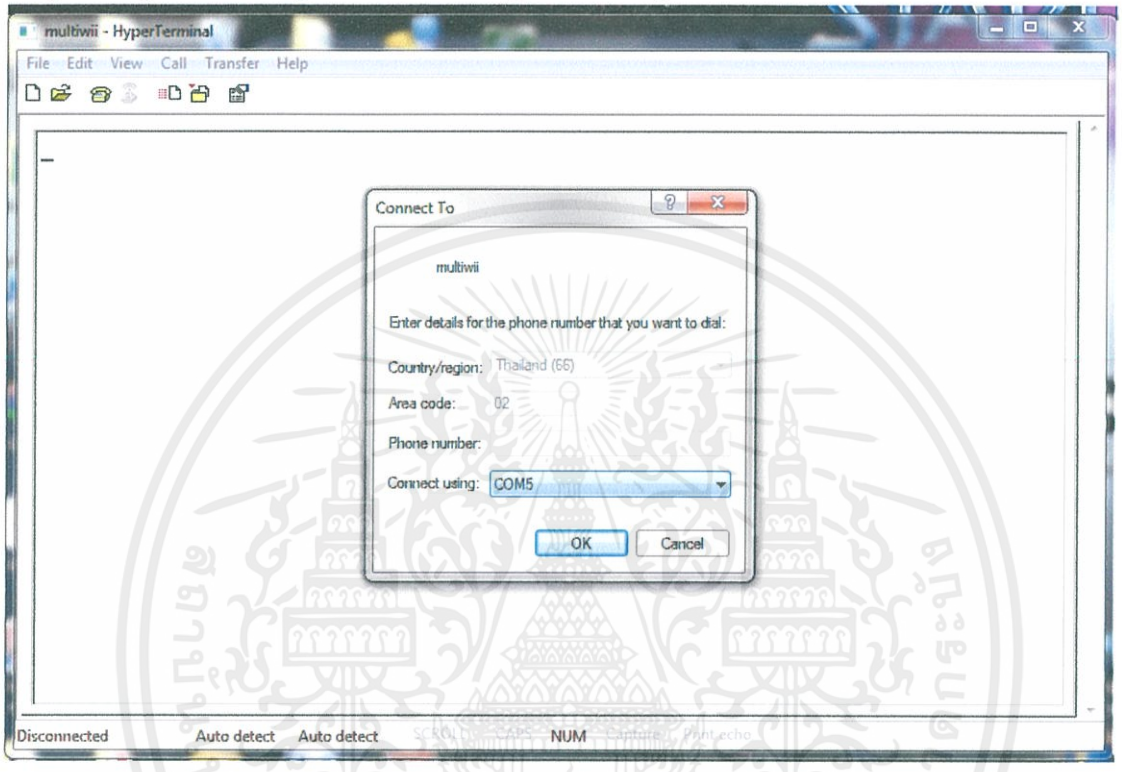
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การดูข้อมูลผ่านซีเรียลพอร์ท

เข้าโปรแกรม HyperTerminal หรือ Arduino IDE

เลือกพอร์ทที่ได้ทำการต่อบอร์ดไว้ ในที่นี้คือ COM5 กด OK

ข้อมูลปรากฏขึ้น โดยขึ้นอยู่กับคำสั่งที่ใช้เรียกดูชนิดของข้อมูล



รูปที่ 3.15 การเชื่อมต่อเข้ากับคอมพอร์ทเพื่อดูข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การใช้การประมวลผลภาพเข้ามาช่วยเพื่อการบินติดตามเป้าหมายอัตโนมัติ

3.7.1 เข้าโปรแกรม Matlab และเปิด New > Script

3.7.2 ที่หน้าต่าง Command Window พิมพ์ imaqhwinfo จะปรากฏข้อมูลอุปกรณ์เกี่ยวกับภาพ

```
Command Window
  >> imaqhwinfo

  ans =

      InstalledAdaptors: {'gentl'  'gige'  'matrox'  'winvideo'}
          MATLABVersion: '8.0 (R2012b)'
            ToolboxName: 'Image Acquisition Toolbox'
         ToolboxVersion: '4.4 (R2012b)'

fx >>
```

รูปที่ 3.16 ผลของการใช้คำสั่งตรวจสอบอุปกรณ์

3.7.3 ในที่นี้จะเลือกใช้ 'winvideo' ซึ่งเป็นอุปกรณ์จำพวก webcam และ build-in camera โดยสามารถตรวจสอบคุณสมบัติโดยการใช้คำสั่ง imaqhwinfo('winvideo',1)

```
Command Window
  >> imaqhwinfo('winvideo',1)

  ans =

              DefaultFormat: 'YUY2_160x120'
        DeviceFileSupported: 0
                 DeviceName: 'USB 2.0 Camera'
                   DeviceID: 1
      VideoInputConstructor: 'videoinput('winvideo', 1)'
    VideoDeviceConstructor: 'imaq.VideoDevice('winvideo', 1)'
          SupportedFormats: {'YUY2_160x120'  'YUY2_176x144'  'YUY2_320x240'  'YUY2_640x480'}

fx >>
```

รูปที่ 3.17 การเรียกดูคุณสมบัติอุปกรณ์

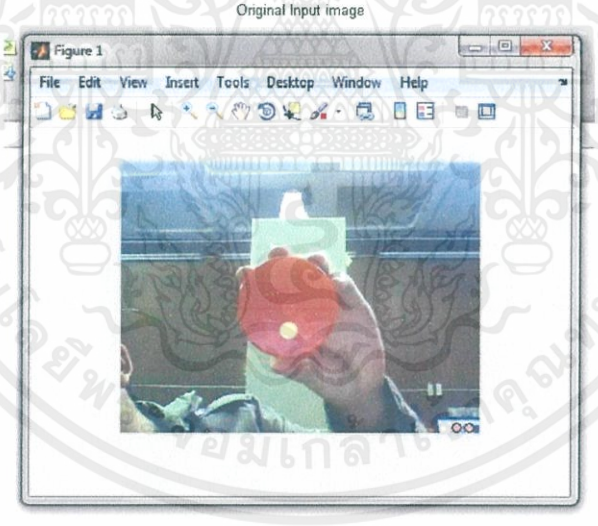
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 จากนั้นนำข้อมูลที่ได้มาประกาศใช้งานกล้องในหน้าต่าง Script และกำหนดชุดคำสั่งพื้นฐาน

```
vid = videoinput('winvideo',1,'YUY2_320x240');
% set video/camera spec.
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5;
start(vid)
```

3.7.5 ใช้การจับภาพจากเฟรมปัจจุบันเข้ามาเพื่อประมวลผลโดยการแยกสีของวัตถุในที่นี้จะเลือกใช้วัตถุสีแดง

```
        rgb = getsnapshot(vid);
    [a b c]=size(rgb);
    y=a;
    x=b;
    rgb1=rgb(:,:,1);rgb2=rgb(:,:,2); rgb3=rgb(:,:,3);
```

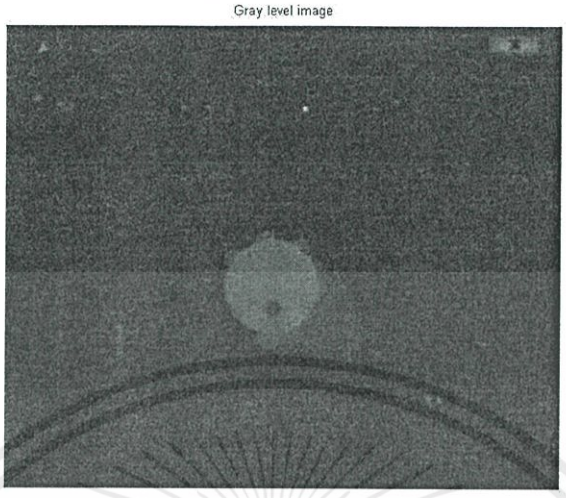


รูปที่ 3.18 ภาพของวัตถุสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.6 แปลงภาพRGBเป็นภาพระดับเทา

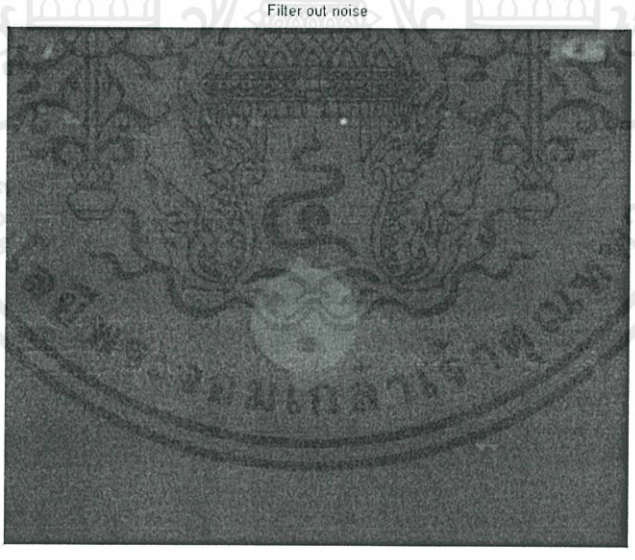
```
diff_im = imsubtract(rgb1, rgb2gray(rgb));
```



รูปที่ 3.19 ภาพระดับเทา

3.7.7 ใช้ Median Filtering เข้ามาช่วยลด Noise

```
diff_im = medfilt2(diff_im, [3 3]);
```

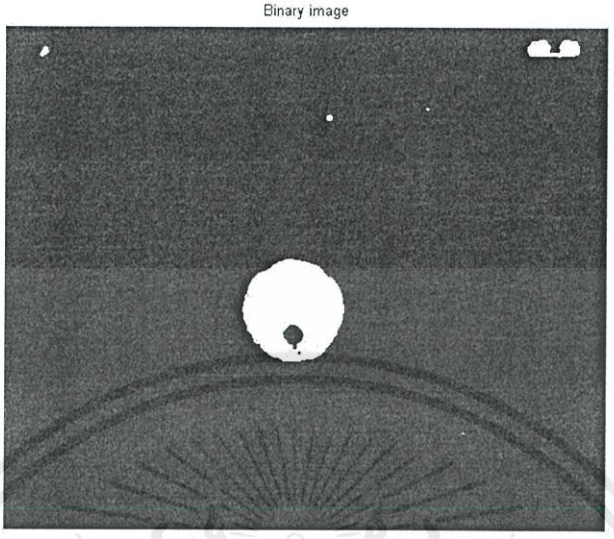


รูปที่ 3.20 แสดงการกำจัดสิ่งรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.8 แปลงภาพระดับเทาเป็นภาพ Binary

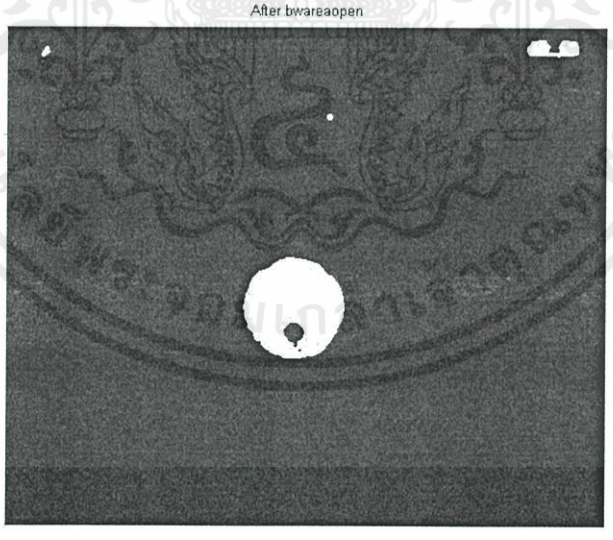
```
diff_im = im2bw(diff_im,0.18);
```



รูปที่ 3.21 ภาพขาวดำ

3.7.9 กำจัด Noise ที่มีขนาดเล็ก

```
bw = bwareaopen(diff_im,16);
```

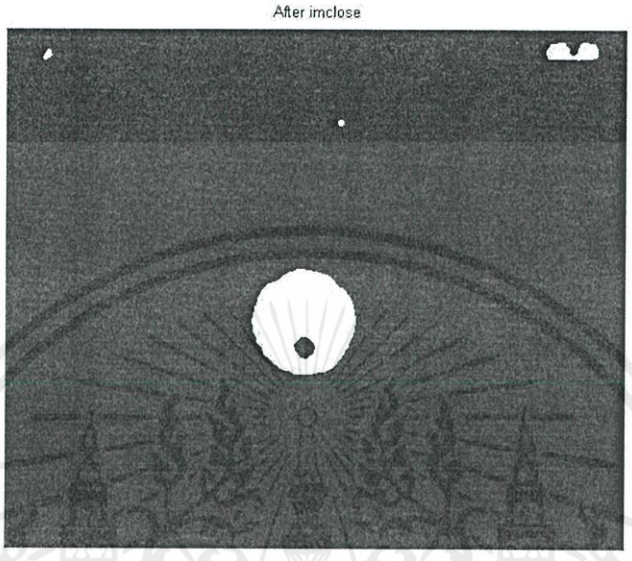


รูปที่ 3.22 การกำจัดนอยส์(Noise)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.10 ทำการซ้อนทับภาพด้วยโครงร่างวงกลม

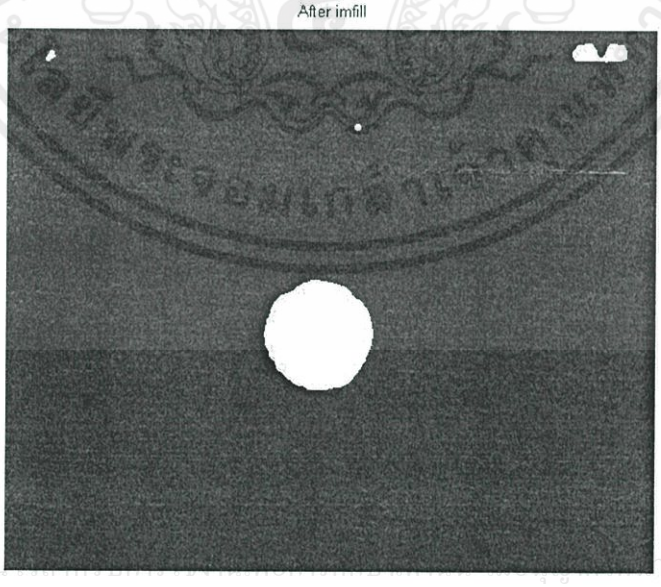
```
se = strel('disk',4);
bw = imclose(bw,se);
```



รูปที่ 3.23 สร้างขอบเขตวงกลม

3.7.11 เติมช่องโหว่ในวงกลมทั้งหมดที่พบ

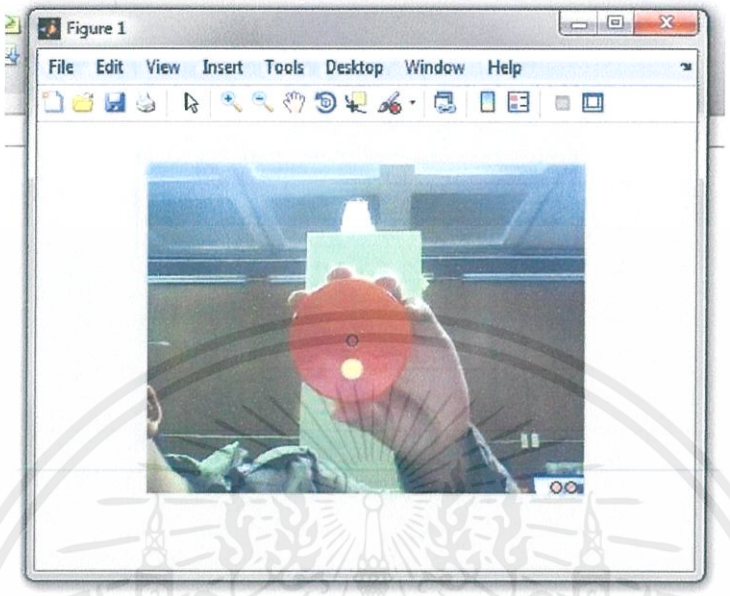
```
bw = imfill(bw,'holes');
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

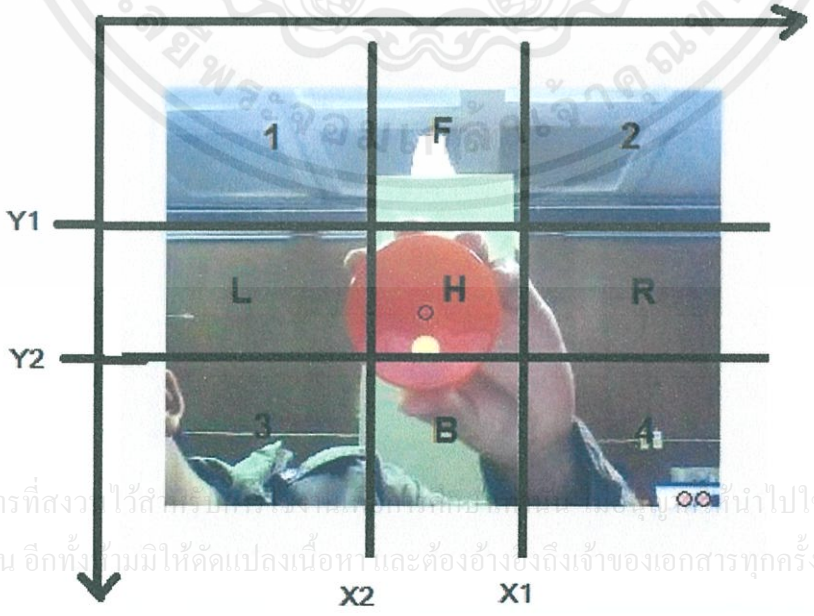
รูปที่ 3.24 เติมเต็มช่องโหว่ในวงกลม

3.7.12 **ใช้ทฤษฎีการหาขอบวงกลมและ Roundness** เพื่อหาจุดศูนย์กลางเป็นเครื่องชี้ตำแหน่งของCentriod โดยสามารถดูชุดคำสั่งได้ในภาคผนวก



รูปที่ 3.25 ใช้แพล็ทฟอร์ม(Platform)หาจุดศูนย์กลางวัตถุ

3.7.13 **กำหนดเส้น X1,X2,Y1,Y2 เพื่อสร้างพื้นที่แบบพิกัดตาราง**ถ้า Centroid อยู่ในพื้นใดก็จะส่งตัวอักษรนั้นผ่าน Serial Port เข้าไปยังบอร์ด Arduino ที่คอยรับค่าอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.26 ตัวอย่างการกำหนดเขตแดน

ตารางที่ 3.2 ตารางรับค่าตัวอักษรและรูปแบบการเคลื่อนที่สำหรับบอร์ด Arduino

อักษร	รูปแบบการเคลื่อนที่
H	บินอยู่กับที่
F	บินไปข้างหน้า
B	บินถอยหลัง
L	บินไปทางซ้าย
R	บินไปทางขวา
1	บินไปทิศซ้ายบน
2	บินไปทิศขวาบน
3	บินไปทิศซ้ายล่าง
4	บินไปทิศขวาล่าง

หมายเหตุ ใช้จุดวงกลมเล็ก(Centroid)เป็นจุดอ้างอิงสำหรับตำแหน่งใดๆในพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองเปิดระบบการทำงาน

4.1.1 สตาร์ทเครื่อง

4.1.2 เร่งความเร็วมอเตอร์

4.1.3 ทดลองบิน

การทดลองวัดสัญญาณจากรีซีฟเวอร์ โดยการโยกคันบังคับที่ตัวรีโมท

ตารางที่ 4.1 แสดงผลคำสั่งที่ได้รับจากรีซีฟเวอร์

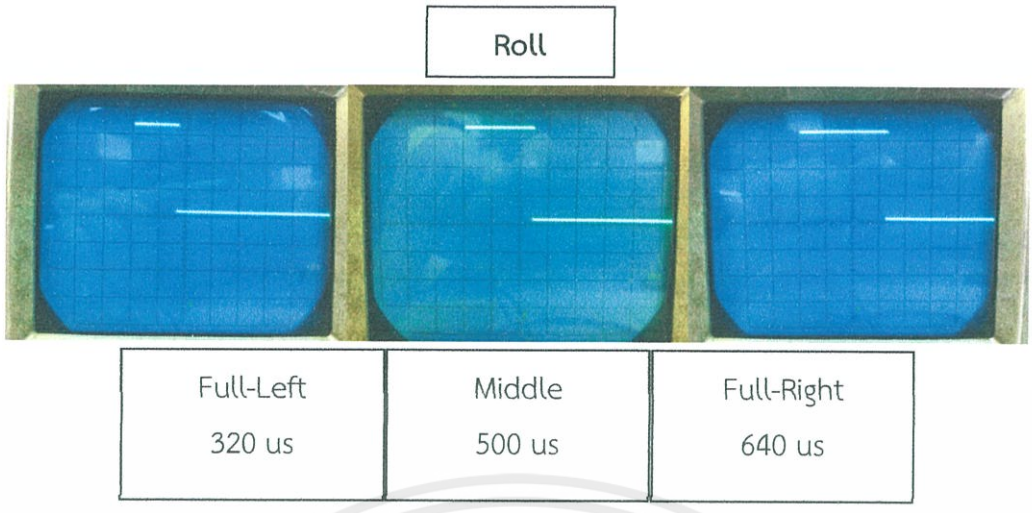
Movement	Stick Direction			Volts
	Full-Down/Left	Middle	Full-Up/Right	
Throttle	340 us	500 us	640 us	3.3
Roll	320 us	500 us	640 us	3.3
Pitch	320 us	520 us	640 us	3.3
Yaw	320 us	520 us	640 us	3.3

Throttle

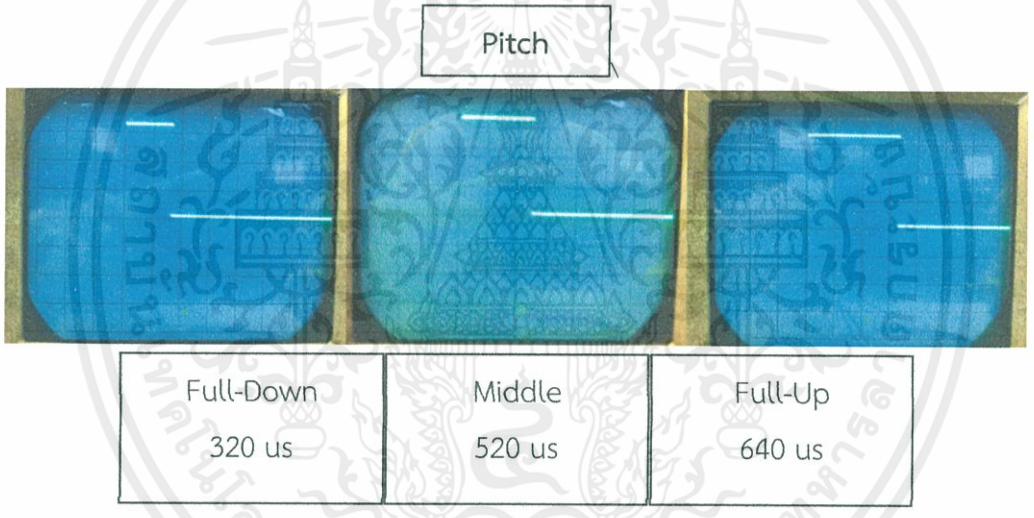
Full-Up	Middle	Full-Down
640 us	500 us	340 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินในแนวดิ่ง

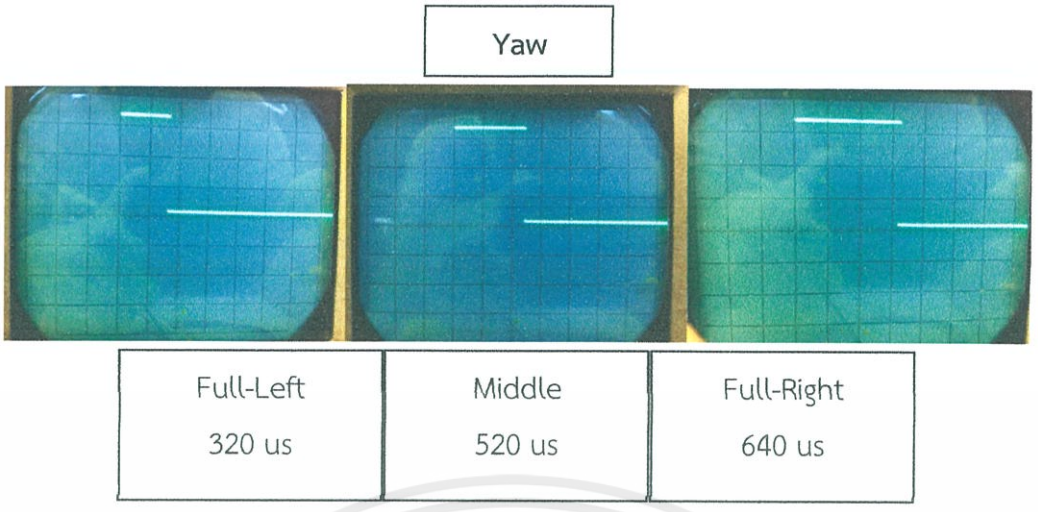


รูปที่ 4.2 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินเอียงข้าง



รูปที่ 4.3 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินเอียงหน้าหรือหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงสัญญาณพัลส์จากตัวรับเมื่อทำการโยกคันบังคับให้เครื่องบินหมุนรอบแกนตั้ง

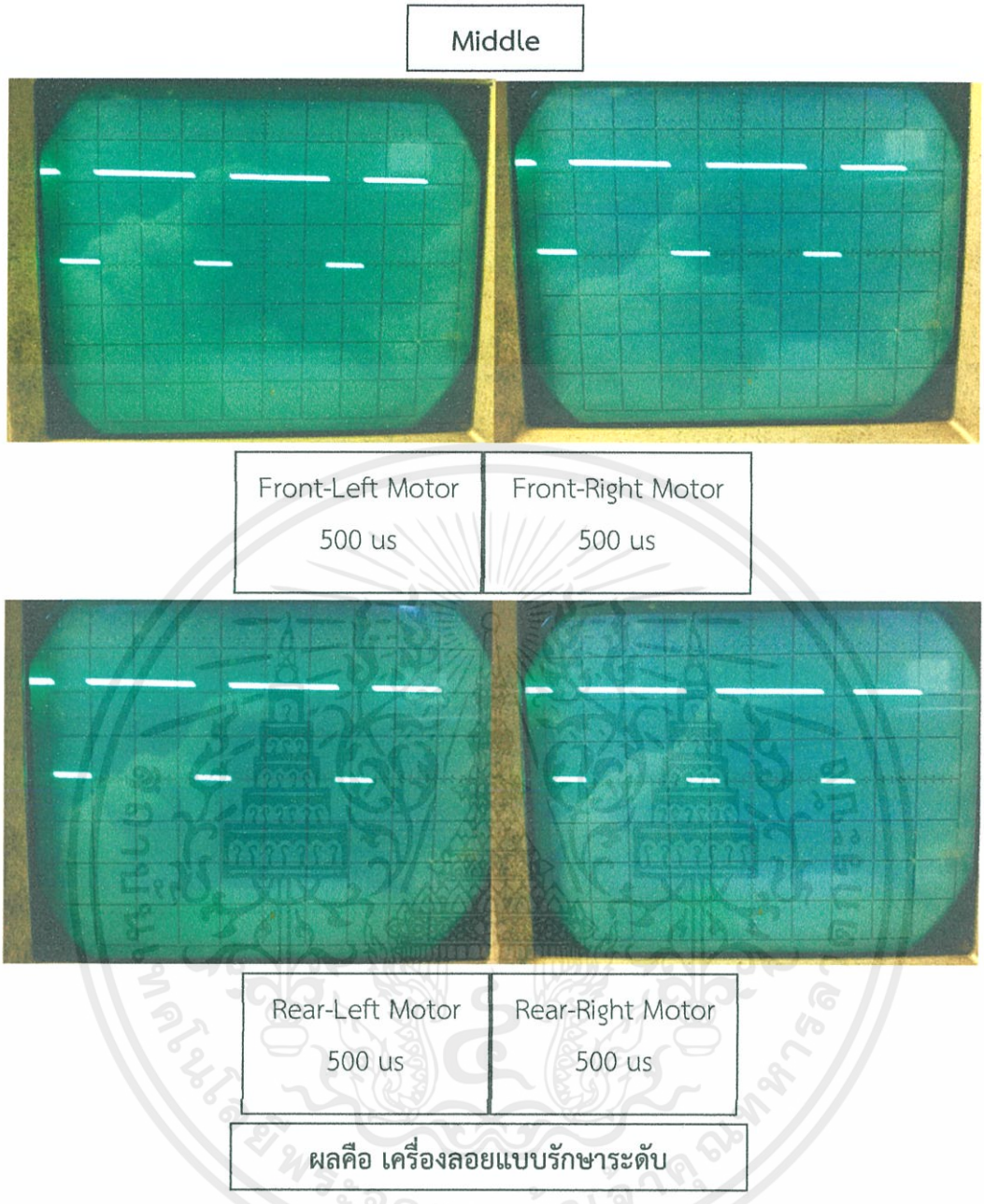
หมายเหตุ : *สโคปที่ใช้วัดตั้งค่าไว้ที่ Volts/Div = 1 volt และ Time/Div = 0.2 ms*

4.2 การทดลองวัดสัญญาณที่ออกจากขาเอาท์พุท(Output)ของบอร์ด Multiwii

ตาราง 4.2 แสดงผลสัญญาณที่ได้รับจากOutput

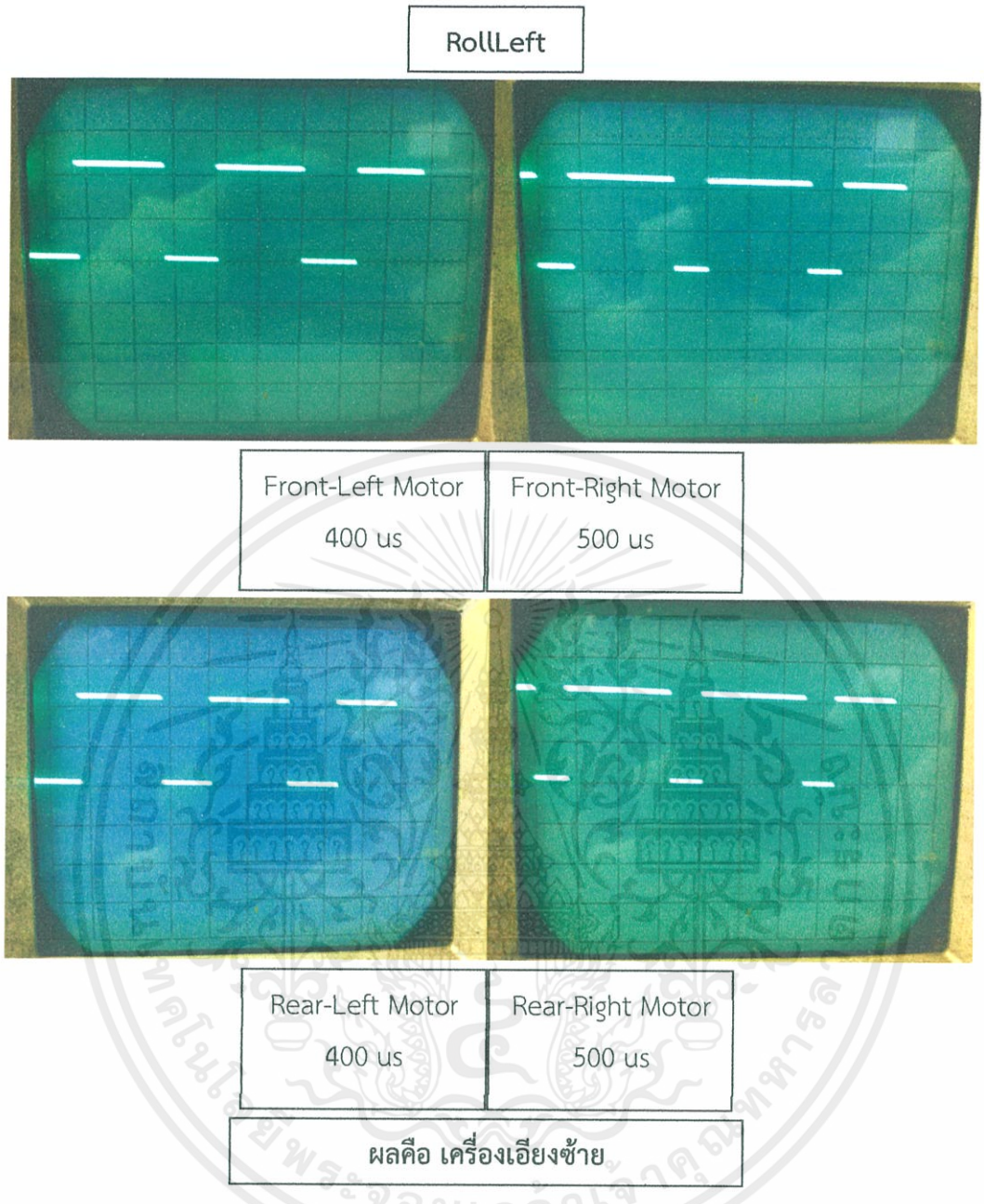
Stick / PIN	Middle	RollLeft	RollRight	PitchDown	PitchUp	YawLeft	YawRight
D3/FrontLeft Motor	500 us	400 us	500 us	500 us	400 us	500 us	400 us
D10/FrontRight Motor	500 us	500 us	400 us	500 us	400 us	400 us	500 us
D11/RearLeft Motor	500 us	400 us	500 us	400 us	500 us	400 us	500 us
D9/RearRight Motor	500 us	500 us	400 us	400 us	500 us	500 us	400 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



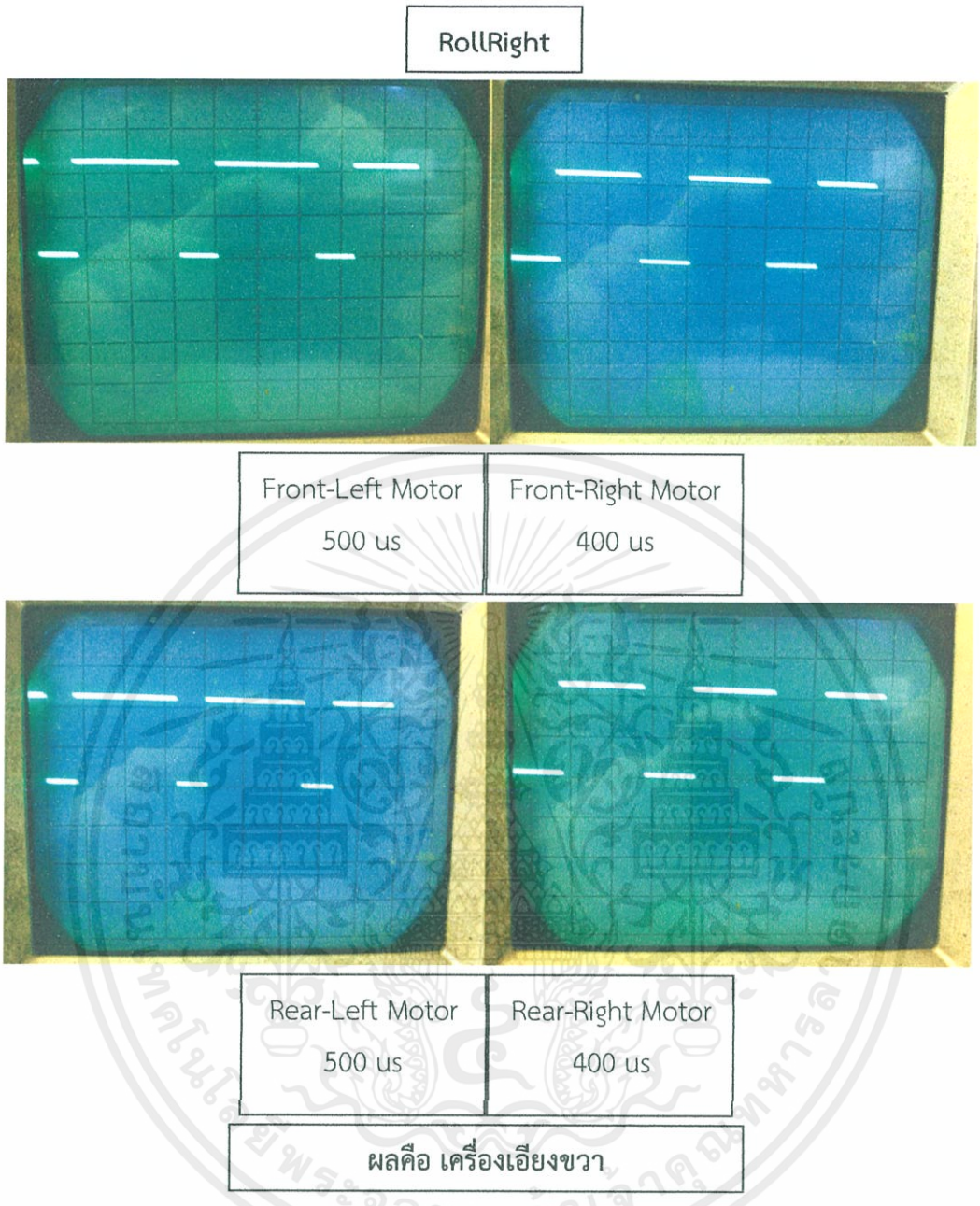
รูปที่ 4.5 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อคันบังคับอยู่ตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



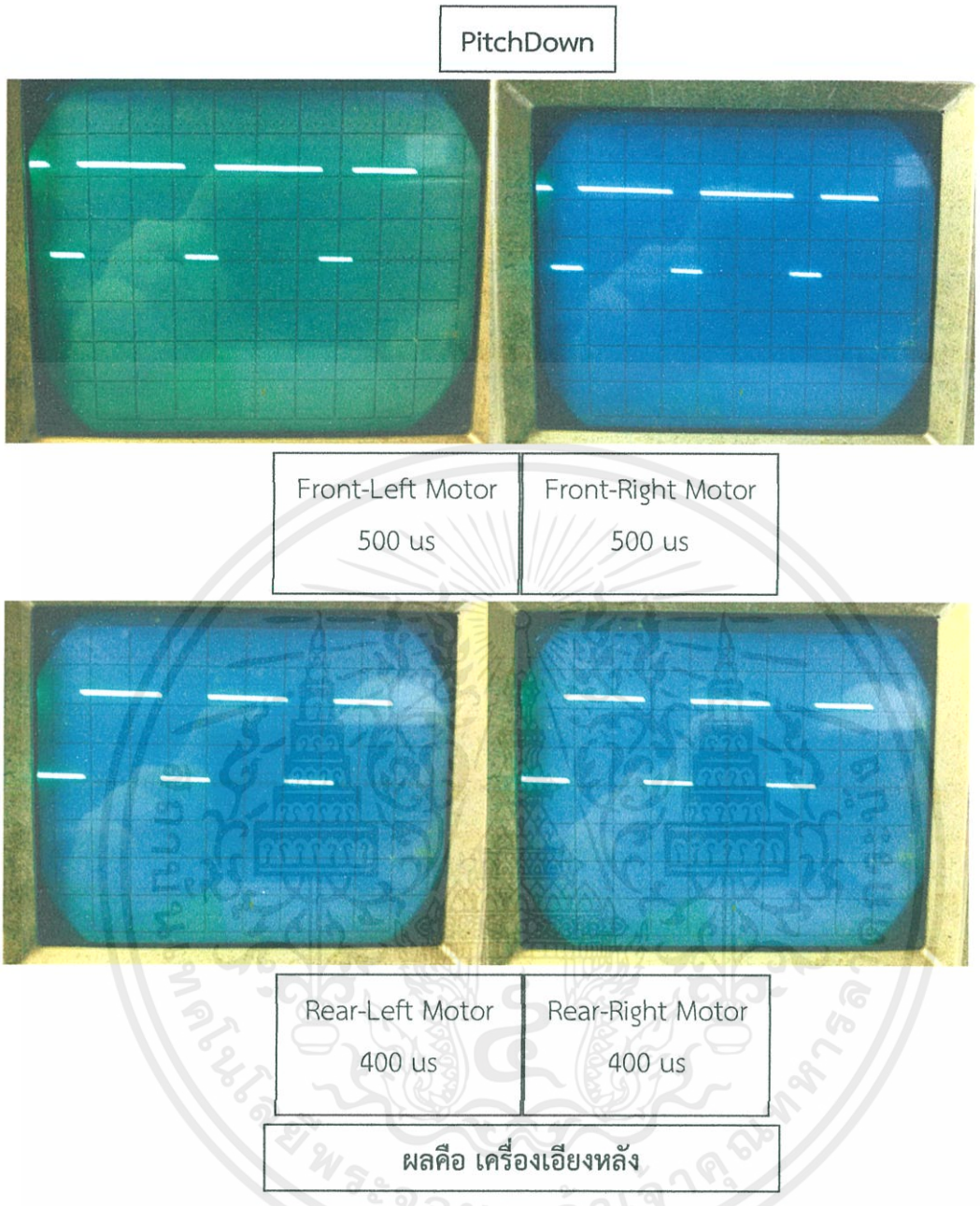
รูปที่ 4.6 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



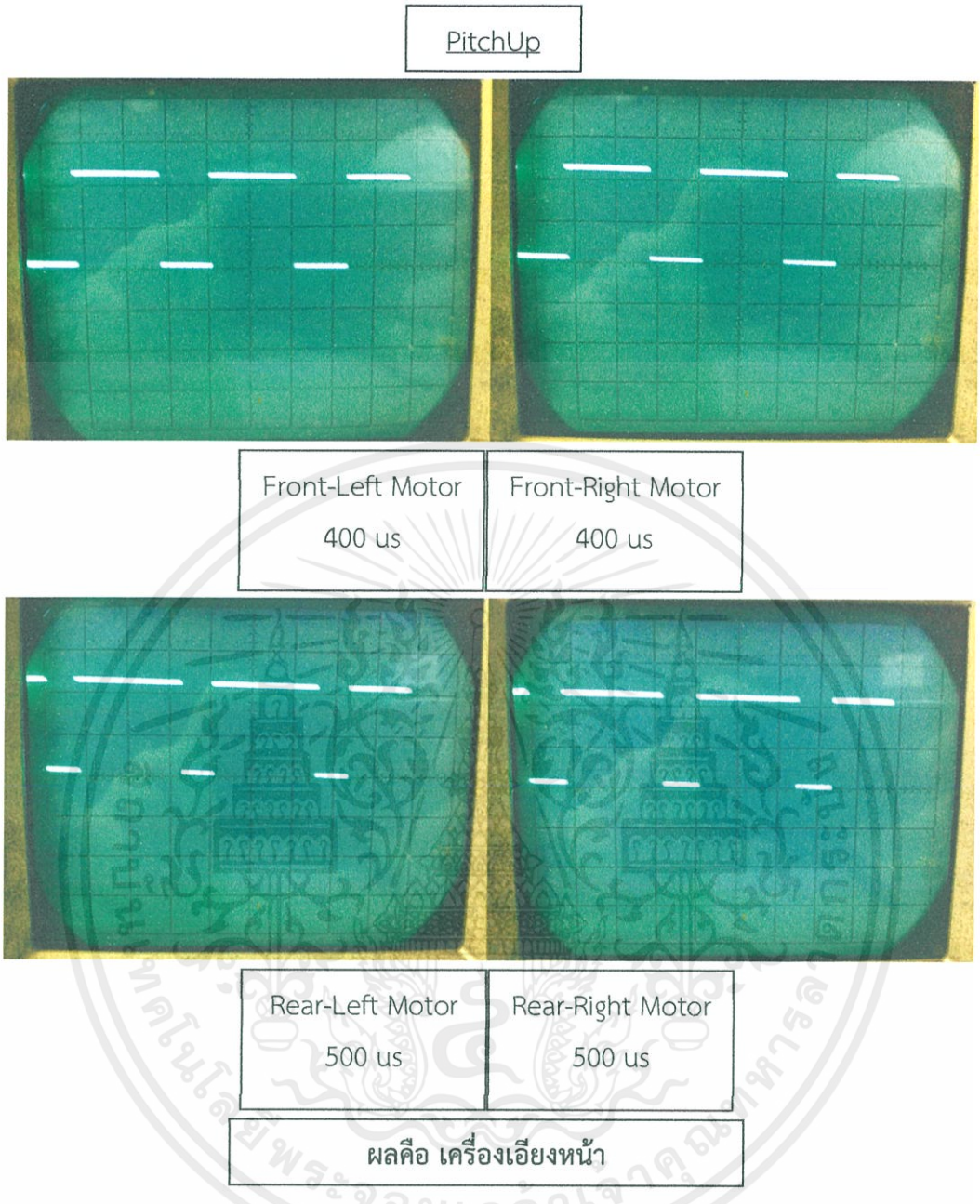
รูปที่ 4.7 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับไปทางขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



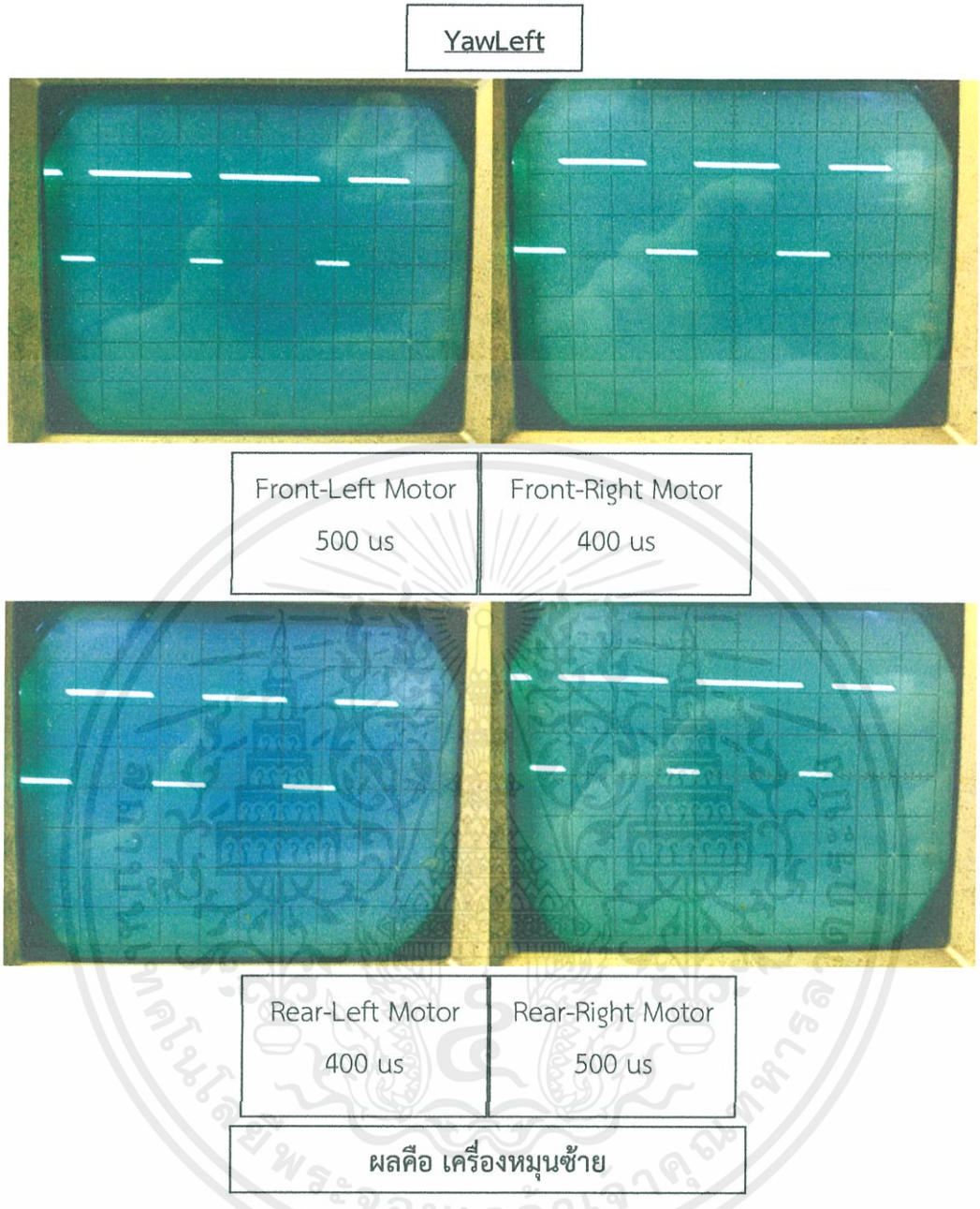
รูปที่ 4.8 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับไปข้างหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



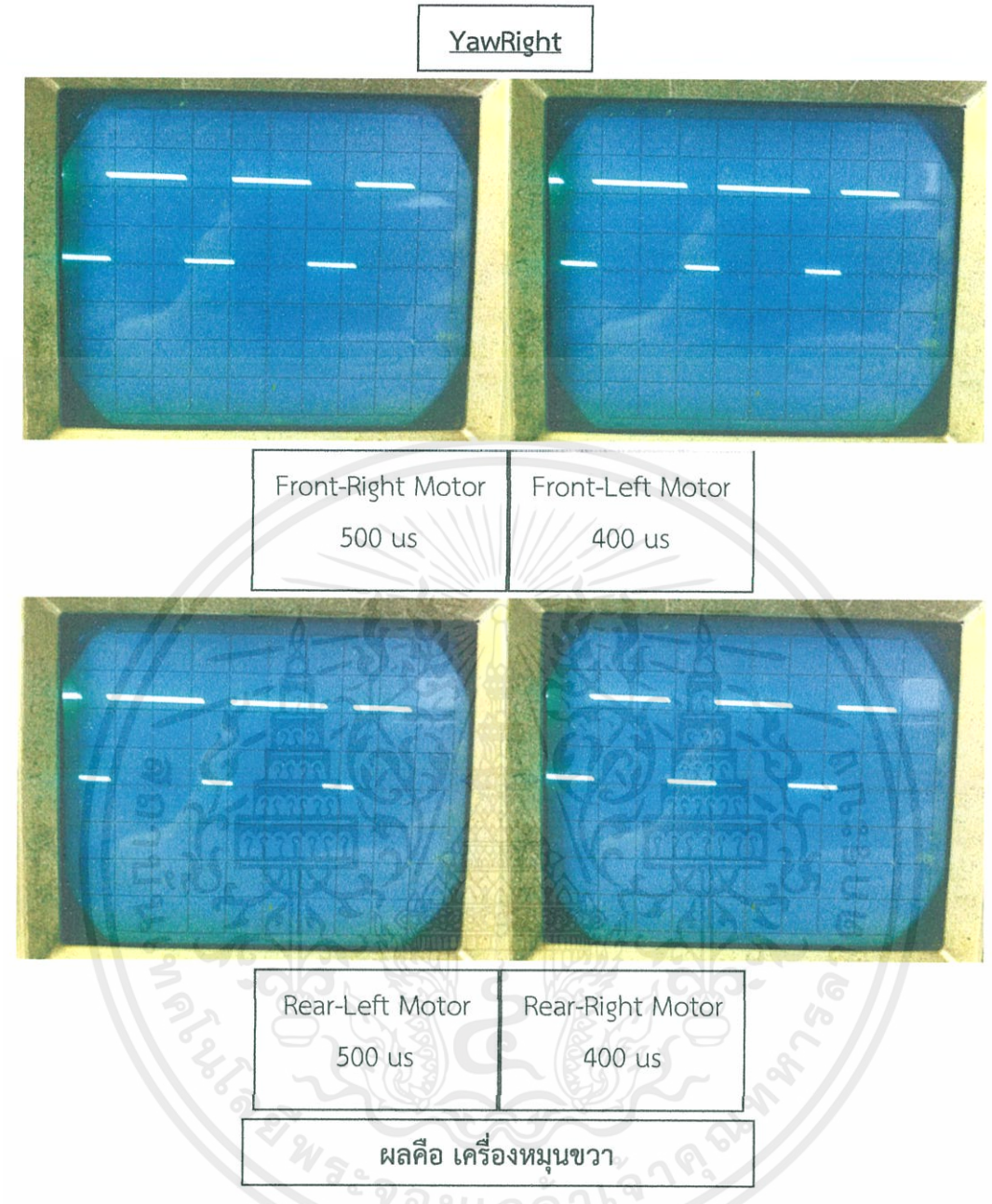
รูปที่ 4.9 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับเลี้ยวไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

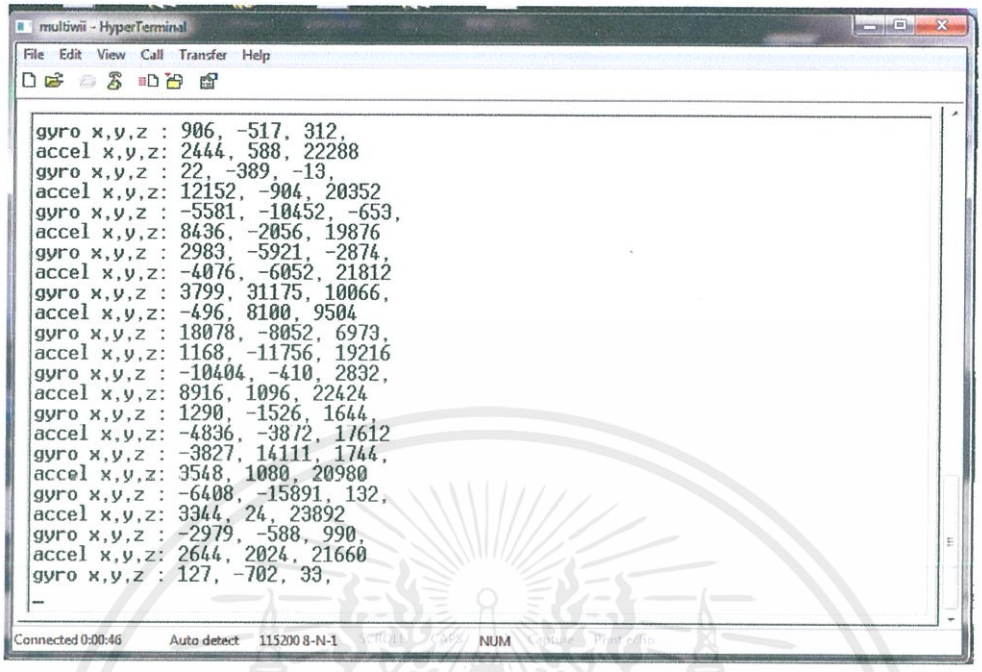


รูปที่ 4.11 สัญญาณเอาท์พุทจากขาบอร์ดมัลติวี(Multiwii)เมื่อโยกคันบังคับเลี้ยวไปทางขวา

หมายเหตุ : สโคปที่ใช้วัดตั้งค่าไว้ที่ Volts/Div = 2 volt และ Time/Div = 0.2 ms

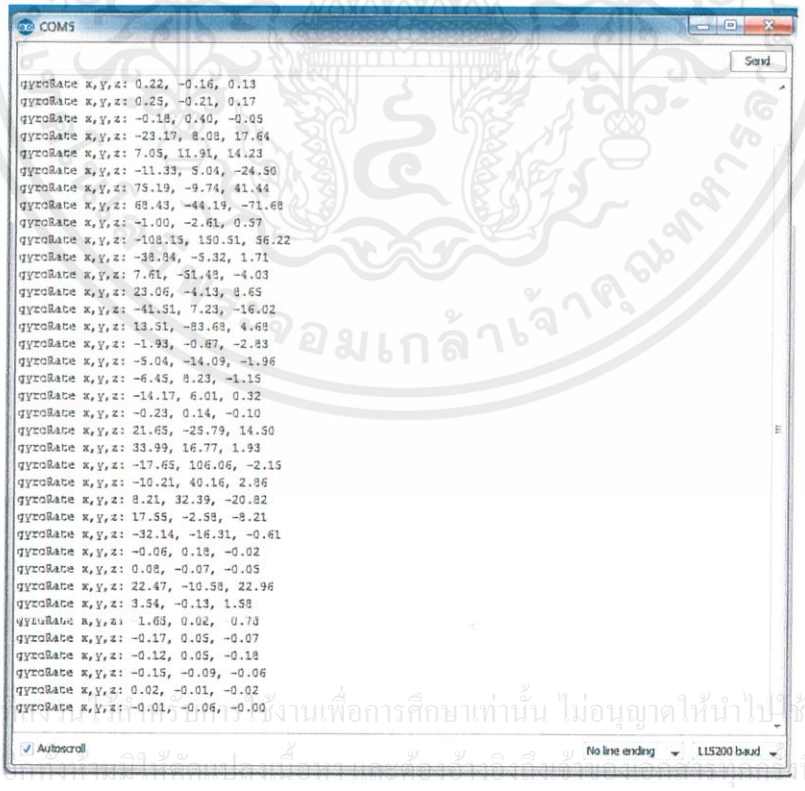
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ข้อมูลจากซีเรียลพอร์ทที่เป็น Analog จากเซ็นเซอร์



รูปที่ 4.12 แสดงข้อมูลของเซ็นเซอร์ที่วิ่งผ่านพอร์ทซีเรียล(Serial Port)

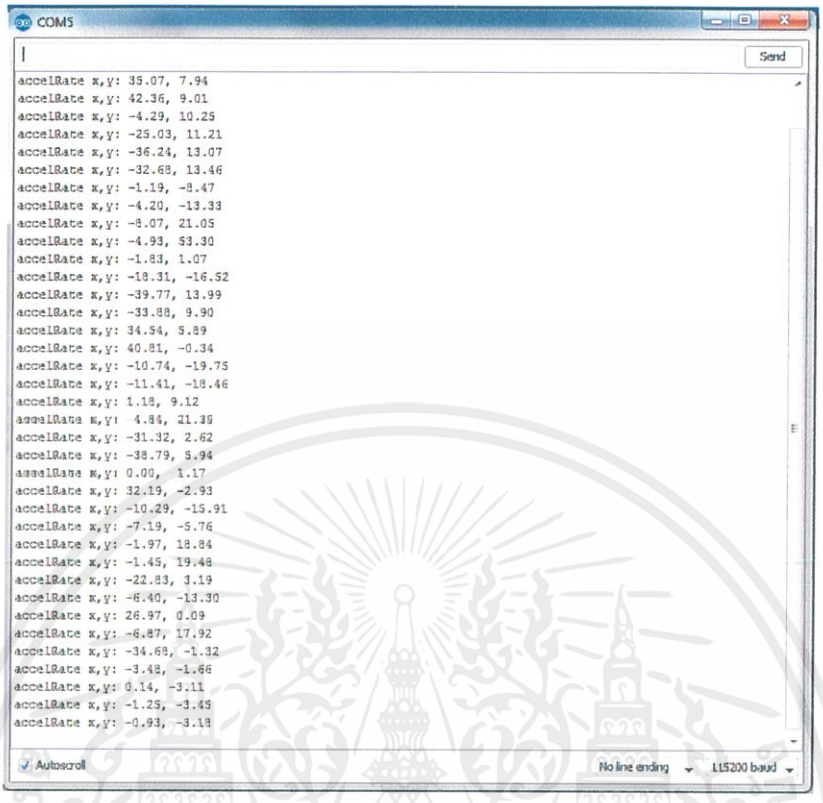
4.4 ข้อมูลจากซีเรียลพอร์ทของ Gyroscope ที่ผ่านการแปลงและคำนวณค่า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.13 แสดงค่าของไจโรสโคป(Gyroscope)ที่ถูกแปลงค่าเป็นดิจิตอลแล้ว

4.5 ข้อมูลจากซีเรียลพอร์ทของ Accelerometer ที่ผ่านการแปลงและคำนวณค่า



รูปที่ 4.14 แสดงแสดงค่าของไจโรสโคป(Accelerometer)ที่ถูกแปลงค่าเป็นดิจิตอลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.3 แสดงค่า Duty Cycle เป็นค่า 0-255 ที่ใช้ในการจ่ายให้ ESC ของ Arduino Board

Region / PIN	H	F	B	L	R	1	2	3	4
D3/FL ESC	128	102	128	102	128	102	128	128	128
D9/FR ESC	128	102	128	128	102	128	102	128	128
D10/RLESC	128	128	102	102	128	128	128	102	128
D11/RR ESC	128	128	102	128	102	128	128	128	102

หมายเหตุ จากคำสั่งจ่ายค่า PWM analogWrite(ESC,Val) นั้นค่าVal จะใช้ในการปรับ Duty Cycle โดยมีค่า 0-255 เช่นต้องการ Duty Cycle = 50% จะต้องใช้ Val = (50 * 255)/100 = 128
หากต้องการ Duty Cycle = 40% ต้องใช้ Val = (40 * 255)/100 = 102

ตาราง 4.4 แสดงค่า Duty Cycle เป็นเปอร์เซ็นต์ ที่ใช้ในการจ่ายให้ ESC ของ Arduino Board

Region / PIN	H	F	B	L	R	1	2	3	4
D3/FL ESC	50%	40%	50%	40%	50%	40%	50%	50%	50%
D9/FR ESC	50%	40%	50%	50%	40%	50%	40%	50%	50%
D10/RLESC	50%	50%	40%	40%	50%	50%	50%	40%	50%
D11/RR ESC	50%	50%	40%	50%	40%	50%	50%	50%	40%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

คณะผู้จัดทำโครงงานได้สร้างเครื่องต้นแบบอากาศยานไร้คนขับ (Autopilot Aerial Robot) ขึ้นเพื่อศึกษาการสร้างเครื่องต้นแบบอากาศยานไร้คนขับและยังเป็นแนวทางการพัฒนาออกแบบอากาศยานไร้คนขับตัวสมบูรณ์ หลังจากที่ผู้จัดทำได้สร้างเครื่องต้นแบบอากาศยานไร้คนขับเสร็จเรียบร้อยแล้ว ได้นำเครื่องต้นแบบอากาศยานไร้นักบินมาทำการบินทดสอบผลการทดสอบ สามารถใช้งานได้จริงแต่ว่ายังต้องมีการแก้ไขในหลายๆเรื่องดังจะกล่าวในหัวข้อต่อไป

5.2 ปัญหาและแนวทางการแก้ไข

5.2.1 ในการทดสอบเครื่องต้นแบบอากาศยานไร้คนขับ จำเป็นต้องอาศัยปัจจัยหลายอย่าง เช่น ดิน,ฟ้า, อากาศและกระแสลม ซึ่งเป็นอุปสรรคอย่างมากขณะทำการทดสอบการบิน

5.2.2 ในการทดสอบเครื่องต้นแบบอากาศยานไร้คนขับ จำเป็นต้องทำการทดสอบในสถานที่ที่เป็นบริเวณลานกว้าง ซึ่งหาได้ยากมาก

5.2.3 อุปกรณ์บางส่วนมีราคาแพงจึงจำเป็นต้องใช้อุปกรณ์ที่มีราคาถูกกว่าซึ่งประสิทธิภาพของการใช้งานใกล้เคียงกัน

5.2.4 โปรแกรมและค่าพารามิเตอร์ต่าง คณะผู้จัดทำได้ทำการศึกษาและคนคว้าด้วยตนเองโดยไม่มีพื้นฐานมาก่อน ทำให้ยังต้องใช้เวลาในการศึกษาทำความเข้าใจให้มากยิ่งขึ้นเพื่อพัฒนาระบบของตนเองขึ้นมา

5.2.5 เนื่องจากขาดประสบการณ์ในการบังคับ และการตั้งปรับจูนค่าความเสถียร ทำให้เกิดอุบัติเหตุขึ้นในระหว่างการทดสอบบิน เป็นผลให้ ตัวเครื่องเสียหาย

5.2.6 ในการใช้การประมวลผลภาพด้วยโปรแกรม Matlab จากการใช้โทนสีในการประมวลผลนั้นมีสิ่งที่ก่อให้เกิดปัญหาและความคลาดเคลื่อนหลายอย่าง เช่น แสงสะท้อนที่กระทบกับตัววัตถุทำให้สีนั้นผิดเพี้ยนไปจากข้อกำหนดที่ตั้งไว้ทำให้ระบบนั้นเกิดความผิดพลาดได้ง่าย

5.2.7 เนื่องจากเป็นอากาศยานที่ออกแบบมาให้บินได้ในระยะไกลแต่ด้วยข้อจำกัดของการประมวลผลภาพเพื่อจะให้เครื่องบินนั้นบินแบบอัตโนมัติ ทำให้สามารถบินได้แค่ในที่ร่มปราศจากแสงสะท้อนเท่านั้นซึ่งอาจจะขัดแย้งกับประเภทของอากาศยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.8 การจะบินอัตโนมัตินั้นจำเป็นต้องมีการรับภาพเข้ามาประมวลผลแบบ Real-Time ทำให้ต้องลงทุนสูงมากในการซื้อโมดูลเพื่อที่จะรับข้อมูลจากตัวกล้องที่ติดอยู่กับยานมาเข้าคอมพิวเตอร์ ซึ่งตรงส่วนนี้ผู้จัดทำขาดงบประมาณในการพัฒนาทำให้ทดลองได้เพียงการบินใน Ideal เท่านั้น

5.3 แนวทางการพัฒนา

5.3.1 เพิ่มประสิทธิภาพการบินให้เสถียรที่สุดโดยให้ บินได้นานและ ประหยัดพลังงาน

5.3.2 สามารถใช้ ระบบบินอัตโนมัติ ด้วยการใช้การประมวลผลทาง Matlab เข้ามาช่วยติดตามเป้าหมายได้อย่างมีประสิทธิภาพ

5.3.3 เนื่องจากการใช้กล้องทั่วไปติดไปกับยานนั้นทำให้ต้องมีการติดโมดูลรับส่งเพิ่มอีกหลายชุดเพื่อรับค่าเข้ามาประมวลผลในคอมพิวเตอร์ จากนั้นส่งกลับไปเข้าบอร์ด Arduino เพื่อเปิดฟังก์ชั่นการบินอัตโนมัติ ทำให้สิ้นเปลืองพลังงานและขั้นตอนค่อนข้างยุ่งยาก ดังนั้นในอนาคตอาจจะมีการนำโทรศัพท์เคลื่อนที่ระบบ Android เข้ามาใช้แทนอุปกรณ์ที่จำเป็นทั้งหมดเนื่องจาก สามารถใช้ระบบ OpenCV ได้เพื่อใช้ประมวณผลข้อมูลในตัวโทรศัพท์ได้ทันทีโดยไม่ต้องติดต่อกับอุปกรณ์ภาคพื้น อีกทั้งยังมีกล้องถ่ายภาพความละเอียดสูงและแบตเตอรี่ในตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

รศ.ดร.ชูชาติ ปิณฑวิรุจน์. 2555. **การประมวลผลภาพดิจิตอลด้วย MATLAB.** พิมพ์ครั้งที่ 2. กรุงเทพฯ : มีน เซอร์วิส ซัพพลาย

เอกชัย มะการ. 2552. **เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino.** พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท อีทีที จำกัด

digitalnature WordPress. **MultiwiiBoard.** [ออนไลน์]. เข้าถึงได้จาก : http :// www.multiwii.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ชุดคำสั่งสำหรับบอร์ดมัลติวี

```
/*Multiwii se v2.0
sensor MPU6050
throttle => D2
roll => D4
pitch => D5
yaw => D6
aux => D7 */
#include <Wire.h>
#define MPU6050
int motorFRONT;
int motorREAR;
int inputKp=0, inputKi=1, inputKd=2, inputPin=3, outputPin=3;
float Setpoint, Input, Output, Kp, Ki, Kd;
byte armed;
int roll=1500;
int pitch=1500;
int yaw=1500;
int thro=1000;
int aux1=1500;
int roll1=1500;
int pitch1=1500;
int yaw1=1500;
int MOTOR_FRONTL_PIN = 3;
int MOTOR_FRONTR_PIN = 10;
int MOTOR_REARL_PIN = 11;
int MOTOR_REARR_PIN = 9;
int motorCommand_FRONTL;
int motorCommand_FRONTR;
int motorCommand_REARL;
int motorCommand_REARR;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MINCOMMAND 1000
#define MIDCOMMAND 1500
#define MAXCOMMAND 2000
int motor_FRONTL;
int motor_FRONTR;
int motor_REARL;
int motor_REARR;
//sensor MPU6050 ---------------------------------
#define MPU6050_ACCEL_XOUT_H  0x3B
#define MPU6050_ACCEL_XOUT_L  0x3C
#define MPU6050_ACCEL_YOUT_H  0x3D
#define MPU6050_ACCEL_YOUT_L  0x3E
#define MPU6050_ACCEL_ZOUT_H  0x3F
#define MPU6050_ACCEL_ZOUT_L  0x40
#define MPU6050_GYRO_XOUT_H   0x43
#define MPU6050_GYRO_XOUT_L   0x44
#define MPU6050_GYRO_YOUT_H   0x45
#define MPU6050_GYRO_YOUT_L   0x46
#define MPU6050_GYRO_ZOUT_H   0x47
#define MPU6050_GYRO_ZOUT_L   0x48
#define MPU6050_PWR_MGMT_1    0x6B
#define MPU6050_I2C_ADDRESS   0x68
unsigned char GYRO_XOUT_L;
unsigned char GYRO_XOUT_H;
unsigned char GYRO_YOUT_L;
unsigned char GYRO_YOUT_H;
unsigned char GYRO_ZOUT_L;
unsigned char GYRO_ZOUT_H;
signed int GYRO_XOUT;
signed int GYRO_YOUT;
signed int GYRO_ZOUT;
unsigned char ACCEL_XOUT_L;
unsigned char ACCEL_XOUT_H;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char ACCEL_YOUT_L;
unsigned char ACCEL_YOUT_H;
unsigned char ACCEL_ZOUT_L;
unsigned char ACCEL_ZOUT_H;
signed int ACCEL_XOUT;
signed int ACCEL_YOUT;
signed int ACCEL_ZOUT;
//calculate sensor -------------------------
float xtrim=0.0;
float ytrim=0.0;
float ztrim=0.0;
float gyroxtrim=0.0;
float gyroytrim=0.0;
float gyroztrim=0.0;
float xAngle=0.0;
float yAngle=0.0;
float zAngle=0.0;
float accXangle;
float accYangle;
float gyroXrate;
float gyroYrate;
float gyroZrate;
#define RAD_TO_DEG 57.295779513082320876798154814105
#define SERIAL_COM_SPEED 115200
// QuadRotor PID GAINS
*****************************************************************************
float KP_QUAD_ROLL= 1.78;
float KD_QUAD_ROLL= 0.45;
float KI_QUAD_ROLL= 0.5;
float KP_QUAD_PITCH= 1.78;
float KD_QUAD_PITCH= 0.45;
float KI_QUAD_PITCH= 0.5;
float KP_QUAD_YAW= 3.28;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float KD_QUAD_YAW= 0.0;
float KI_QUAD_YAW= 0.0;
#define KD_QUAD_COMMAND_PART 13.0
float Kp_RateRoll = 1.95;
float Ki_RateRoll = 0.0;
float Kd_RateRoll = 0.0;
float Kp_RatePitch = 1.95;
float Ki_RatePitch = 0.0;
float Kd_RatePitch = 0.0;
float Kp_RateYaw = 3.28;
float Ki_RateYaw = 0.0;
float Kd_RateYaw = 0.0;
float xmitFactor = 0.32;
static float previousRollRate, previousPitchRate, previousYawRate;
float currentRollRate, currentPitchRate, currentYawRate;
// Attitude control variables
float command_rx_roll=0;
float command_rx_roll_old;
float command_rx_roll_diff;
float command_rx_pitch=0;
float command_rx_pitch_old;
float command_rx_pitch_diff;
float command_rx_yaw=0;
float command_rx_yaw_old;
float command_rx_yaw_diff;
float K_aux;
float roll_I=0;
float roll_D;
float err_roll;
float err_roll_rate;
float err_roll_ant;
float pitch_I=0;
float pitch_D;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float err_pitch;
float err_pitch_rate;
float err_pitch_ant;
float yaw_I=0;
float yaw_D;
float err_yaw;
float err_yaw_rate;
float err_yaw_ant;
float control_roll=0.0;
float control_pitch=0.0;
float control_yaw=0.0;
long timer=0;
long timer_old;
float G_Dt=0.02;
//tx ----------------
int ch1;
int ch2;
int ch3;
int ch4;
int ch_aux;
int ch_aux2;
#define MAXCHANIN 5
#define MAXCHANOUT 4
int channelIn[MAXCHANIN];
int channelOut[MAXCHANOUT];
void mixer()
{ channelOut[0]=channelIn[0];
  channelOut[1]=channelIn[1];
  channelOut[2]=channelIn[2];
  channelOut[3]=channelIn[3];}
/** RX Section **/
#define CENTER 1520
int rawIn[5];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void setupRx()
{ pinMode(2, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  // interrupt on pin change PCINT
  PCICR |= (1 << PCIE2);
  PCMSK2 = (1 << PCINT18) | // pin2
  (1 << PCINT20) | // pin4
  (1 << PCINT21) | // pin5
  (1 << PCINT22) | // pin6
  (1 << PCINT23) ; // pin7 }
#define MASKPCINT0 (1<<2)
#define MASKPCINT1 (1<<4)
#define MASKPCINT2 (1<<5)
#define MASKPCINT3 (1<<6)
#define MASKPCINT4 (1<<7)
ISR(PCINT2_vect)
{ static byte newbit,oldbit,changed;
  static unsigned long startIn[5];
  static unsigned long time;
  time=micros();
  newbit=PIND;
  changed=newbit^oldbit;
  if (changed&MASKPCINT0)
    if (newbit&MASKPCINT0) startIn[0]=time;
    else rawIn[0]=time-startIn[0];
  if (changed&MASKPCINT1)
    if (newbit&MASKPCINT1) startIn[1]=time;
    else rawIn[1]=time-startIn[1];
  if (changed&MASKPCINT2)
    if (newbit&MASKPCINT2) startIn[2]=time;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else rawIn[2]=time-startIn[2];
  if (changed&MASKPCINT3)
    if (newbit&MASKPCINT3) startIn[3]=time;
    else rawIn[3]=time-startIn[3];
  if (changed&MASKPCINT4)
    if (newbit&MASKPCINT4) startIn[4]=time;
    else rawIn[4]=time-startIn[4];
  oldbit=newbit;}
void updateRx()
 { for(int i=0;i<MAXCHANIN;i++)
    if (rawIn[i]>800 && rawIn[i]<2200)
      channelIn[i]=rawIn[i]-CENTER; }
/****************** ESC ************************/
void setupEsc()
{  pinMode(9, OUTPUT);  // ขวา-หลัง
  pinMode(10, OUTPUT); // ขวา-หน้า
  pinMode(3,OUTPUT); // ซ้าย-หน้า
  pinMode(11,OUTPUT); // ซ้าย-หลัง }
void updateEsc()
{ unsigned int period;
  period=CENTER+channelOut[0];
  period=limit(1000,period,2000);
  analogWrite(9,period>>3);
  period=CENTER+channelOut[1];
  period=limit(1000,period,2000);
  analogWrite(10,period>>3);
  period=CENTER+channelOut[2];
  period=limit(1000,period,2000);
  analogWrite(3,period>>3);
  period=CENTER+channelOut[3];
  period=limit(1000,period,2000);
  analogWrite(11,period>>3); }
int limit(int mn,int in,int mx)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ int out;
  if (in<mn) out=mn;
  else if (in>mx) out=mx;
  else out=in;
  return out; }
//Motor Command
*****************************************************************************************
void configureMotors() {
  pinMode(MOTOR_FRONTL_PIN,OUTPUT);
  pinMode(MOTOR_FRONTR_PIN,OUTPUT),
  pinMode(MOTOR_REARL_PIN,OUTPUT);
  pinMode(MOTOR_REARR_PIN,OUTPUT);
  commandAllMotors(1000);}
void commandAllMotors(int motorCommand) {
  analogWrite(MOTOR_FRONTL_PIN, motorCommand/8);
  analogWrite(MOTOR_FRONTR_PIN, motorCommand/8);
  analogWrite(MOTOR_REARL_PIN, motorCommand/8);
  analogWrite(MOTOR_REARR_PIN, motorCommand/8);}
void commandMotors() {
  analogWrite(MOTOR_FRONTL_PIN, motor_FRONTL/8);
  analogWrite(MOTOR_FRONTR_PIN, motor_FRONTR/8);
  analogWrite(MOTOR_REARL_PIN, motor_REARL/8);
  analogWrite(MOTOR_REARR_PIN, motor_REARR/8);}
//sensor ----------------------
void Get_Gyro_Rates()
{ GYRO_XOUT = calculate_value(MPU6050_GYRO_XOUT_H, MPU6050_GYRO_XOUT_L,
1);
  GYRO_YOUT = calculate_value(MPU6050_GYRO_YOUT_H, MPU6050_GYRO_YOUT_L,
1);
  GYRO_ZOUT = calculate_value(MPU6050_GYRO_ZOUT_H, MPU6050_GYRO_ZOUT_L,
1);
void Get_Accel_Values()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{  ACCEL_XOUT = calculate_value(MPU6050_ACCEL_XOUT_H,
MPU6050_ACCEL_XOUT_L, 1);
  ACCEL_YOUT = calculate_value(MPU6050_ACCEL_YOUT_H,
MPU6050_ACCEL_YOUT_L, 1);
  ACCEL_ZOUT = calculate_value(MPU6050_ACCEL_ZOUT_H,
MPU6050_ACCEL_ZOUT_L, 1);}
uint16_t calculate_value(int address_h, int address_l, int size) {
  uint8_t hi, lo;
  uint16_t value;
  hi = MPU6050_read(address_h, size);
  lo = MPU6050_read(address_l, size);
  value = ((hi)<<8);
  value |= lo;
  return(value); }
 uint8_t MPU6050_read(int address, int size) {
  uint8_t value;
  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  Wire.write(address);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);
  while(Wire.available()) {
  value = Wire.read(); }
  return (value);}
  int MPU6050_write(int address, uint8_t *data, int size) {
  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  Wire.write(address);        // write the start address
  Wire.write(data, size);  // write data bytes
  Wire.endTransmission(true); // release the I2C-bus
  return (0);         // return : no error }
int channel_filter(int ch, int ch_old)
{  if (ch_old==0)
    return(ch);
  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return((ch+ch_old)>>1);     // small filtering (average filter)}
void setup()
{int i;
  int j;
  int aux;
  uint8_t c = 0;
  Wire.begin();
  //initialize the serial link with processing
  Serial.begin(115200);
  pinMode(13, OUTPUT);
  MPU6050_write(MPU6050_PWR_MGMT_1, &c, 1); //Stop device from
reseting/sleeping
  configureMotors();
  setupRx();
   for (int j = 0 ; j <= 100 ; j++)
  { Get_Gyro_Rates();
    Get_Accel_Values();
    gyroXrate = (((float)GYRO_YOUT)*0.0076335);//
    gyroYrate = (((float)GYRO_XOUT)*0.0076335);//
    gyroZrate = (((float)GYRO_ZOUT)*0.0076335);
    float R = sqrt(pow(ACCEL_XOUT,2)+pow(ACCEL_YOUT,2)+pow(ACCEL_ZOUT,2));
    float accXangle = (acos(ACCEL_YOUT/R)*RAD_TO_DEG - 90)*-1;
    float accYangle = (acos(ACCEL_XOUT/R)*RAD_TO_DEG - 90);
     xtrim = accXangle;
     ytrim = accYangle;
     gyroxtrim += gyroXrate;
     gyroytrim += gyroYrate;
     gyroztrim += gyroZrate;
     digitalWrite(13, LOW);
     delay(20);
     digitalWrite(13, HIGH);
     delay(20);   }
     gyroxtrim = gyroxtrim/100.0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  gyroytrim = gyroytrim/100.0;
  gyroztrim = gyroztrim/100.0;
  Setpoint = 0;
  roll1 = 1477;
  pitch1 = 1482;
  yaw1 = 1480;
  digitalWrite(13, LOW);
 timer = millis();//start timing }
 void loop()
{  int aux;
 float aux_float;
 if((millis()-timer)>=10)
 { timer_old = timer;
   timer=millis();
   G_Dt = (timer-timer_old)/1000.0;
  updateRx();
  mixer();
  ch1 = channelIn[1]=rawIn[1];   // Roll
  ch2 = channelIn[2]=rawIn[2];   // Pitch
  ch3 = channelIn[0]=rawIn[0];   // Throttle
  ch4 = channelIn[3]=rawIn[3];   // Yaw
  ch_aux = channelIn[4]=rawIn[4];    // Aux
  roll = channel_filter(ch1,roll);
  pitch = channel_filter(ch2,pitch);
  thro = channel_filter(ch3,thro);
  yaw = channel_filter(ch4,yaw);
  aux1 = channel_filter(ch_aux,aux1);
  Get_Accel_Values();
  Get_Gyro_Rates();
  gyroXrate = (GYRO_YOUT*0.0076335) - gyroxtrim;
  gyroYrate = (GYRO_XOUT*0.0076335) - gyroytrim;
  gyroZrate = (GYRO_ZOUT*0.0076335) - gyroztrim;
  float R = sqrt(pow(ACCEL_XOUT,2)+pow(ACCEL_YOUT,2)+pow(ACCEL_ZOUT,2));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
accXangle = (acos(ACCEL_YOUT/R)*RAD_TO_DEG - 90)*-1  - xtrim;
accYangle = (acos(ACCEL_XOUT/R)*RAD_TO_DEG - 90) - ytrim;
xAngle = (0.99*(xAngle + gyroXrate*G_Dt)) + (0.01*accYangle);
yAngle = (0.99*(yAngle + gyroYrate*G_Dt)) + (0.01*accXangle);
xAngle = constrain(xAngle, -90, 90);
yAngle = constrain(yAngle, -90, 90);
//tx===
//ค่าความเร็วจากรีโมท
  command_rx_roll_old = command_rx_roll;
  command_rx_roll = (ch1-roll1)/13.0;
  command_rx_roll_diff = command_rx_roll-command_rx_roll_old;
  command_rx_pitch_old = command_rx_pitch;
  command_rx_pitch = (ch2-pitch1)/13.0;
  command_rx_pitch_diff = command_rx_pitch-command_rx_pitch_old;
  command_rx_yaw_old = command_rx_yaw;
  command_rx_yaw = (ch4-yaw1)/13.0;
  command_rx_yaw_diff = command_rx_yaw-command_rx_yaw_old;
//yawR/pitchDown to On  and yawL/pitchUp to Off
if (thro < 1110) {
  if (yaw < 1200 && pitch > 1700 && armed == 1) {
    armed = 0;
    commandAllMotors(MINCOMMAND);  }
  if (yaw > 1700 && pitch < 1200 && armed == 0) {
    armed = 1;
    commandAllMotors(MINCOMMAND + 150);
    delay(500);
    commandAllMotors(MINCOMMAND);  }  }
if (armed == 0) {
  motor_FRONTL = MINCOMMAND;
  motor_FRONTR = MINCOMMAND;
  motor_REARL = MINCOMMAND;
  motor_REARR = MINCOMMAND;     }
if(ch_aux<1300)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{   digitalWrite(13, LOW);
    //   Rate Mode
    // ROLL CONTROL
    err_roll = ((roll- roll1) * xmitFactor) - gyroXrate;
    err_roll = constrain(err_roll,-30,30);
    roll_I += err_roll*G_Dt;
    roll_I = constrain(roll_I, -50, 50);
    roll_D = -gyroXrate;
    control_roll = Kp_RateRoll*err_roll + Kd_RateRoll*roll_D + Ki_RateRoll*roll_I;
    // PITCH CONTROL
    err_pitch = ((pitch- pitch1) * xmitFactor) - gyroYrate;
    err_pitch = constrain(err_pitch,-30,30);
    pitch_I += err_pitch*G_Dt;
    pitch_I = constrain(pitch_I, -50, 50);
    pitch_D = -gyroYrate;
    control_pitch = Kp_RatePitch*err_pitch + Kd_RatePitch*pitch_D +
Ki_RatePitch*pitch_I;
    // YAW CONTROL
    err_yaw = ((yaw- yaw1) * xmitFactor) - gyroZrate;
    err_yaw = constrain(err_yaw,-30,30);
    yaw_I += err_yaw*G_Dt;
    yaw_I = constrain(yaw_I, -50, 50);
    yaw_D = -gyroZrate;
    control_yaw = Kp_RateYaw*err_yaw + Kd_RateYaw*yaw_D + Ki_RateYaw*yaw_I;
  }else{
    digitalWrite(13, HIGH);
    //  Stable Mode
    // ROLL CONTROL
    err_roll = command_rx_roll - xAngle;
    err_roll = constrain(err_roll,-30,30);
    roll_I += err_roll*G_Dt;
    roll_I = constrain(roll_I, -50, 50);
    roll_D = -gyroXrate;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      control_roll = KP_QUAD_ROLL*err_roll + KD_QUAD_ROLL*roll_D +
KI_QUAD_ROLL*roll_I;
      // PITCH CONTROL
      err_pitch = command_rx_pitch - yAngle;
      err_pitch = constrain(err_pitch,-30,30);
      pitch_I += err_pitch*G_Dt;
      pitch_I = constrain(pitch_I, -50, 50);
      pitch_D = -gyroYrate;
      control_pitch = KP_QUAD_PITCH*err_pitch + KD_QUAD_PITCH*pitch_D +
KI_QUAD_PITCH*pitch_I;
      // YAW CONTROL
      err_yaw = ((yaw- yaw1) * xmitFactor) - gyroZrate;
      err_yaw = constrain(err_yaw,-30,30);
      yaw_I += err_yaw*G_Dt;
      yaw_I = constrain(yaw_I, -50, 50);
      yaw_D = -gyroZrate;
      control_yaw = Kp_RateYaw*err_yaw + Kd_RateYaw*yaw_D + Ki_RateYaw*yaw_I;
}
    if (armed) {
      // Calculate motor commands
      motor_FRONTL = thro + control_pitch + control_roll + control_yaw;
      motor_FRONTR = thro + control_pitch - control_roll - control_yaw;
      motor_REARL = thro - control_pitch + control_roll - control_yaw;
      motor_REARR = thro - control_pitch - control_roll + control_yaw;     }
    if (thro < 1110) {
      motor_FRONTL = 1000;
      motor_FRONTR = 1000;
      motor_REARL = 1000;
      motor_REARR = 1000;   }
     commandMotors();
    // Print the raw acceleration values
//  Serial.print(F("accel x,y,z: "));
// Serial.print(ACCEL_XOUT, DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Serial.print(F(", "));
//  Serial.print(ACCEL_YOUT, DEC);
//  Serial.print(F(", "));
//  Serial.print(ACCEL_ZOUT, DEC);
//  Serial.println(F(""));
  // Print the raw gyro values.
//   Serial.print(F("gyro x,y,z : "));
//   Serial.print(GYRO_XOUT, DEC);
//   Serial.print(F(", "));
//   Serial.print(GYRO_YOUT, DEC);
//   Serial.print(F(", "));
//   Serial.print(GYRO_ZOUT, DEC);
//   Serial.print(F(", "));
//   Serial.println(F(""));
//   Serial.print(motor_FRONTL);Serial.print("\t");
//   Serial.print(motor_FRONTR);Serial.print("\n");
//   Serial.print(motor_REARL);Serial.print("\t");
//   Serial.print(motor_REARR);Serial.print("\n");
//   Serial.print("\n");
//   Serial.print(F("gyroRate x,y,z: "));
//   Serial.print(gyroXrate);
//   Serial.print(F(", "));
//   Serial.print(gyroYrate);
//   Serial.print(F(", "));
//   Serial.print(gyroZrate);Serial.print("\n");
//    Serial.print(F("accelRate x,y: "));
//    Serial.print(accXangle);
//    Serial.print(F(", "));
//    Serial.print(accYangle);
//    Serial.print("\n");
//    delay(1000); } }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งแม็ทแล็บ(Matlab)

```
ser=serial('COM6','Baudrate',1200);
fopen(ser)
vid = videoinput('winvideo',1,'YUY2_320x240');

% set video/camera spec.
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5;
%start the video aquisition here
start(vid)

while(1)

    %snapshot a current frame from video.
    rgb = getsnapshot(vid);
    [a b c]=size(rgb);
    y=a;
    x=b;
    rgb1=rgb(:,:,1);rgb2=rgb(:,:,2); rgb3=rgb(:,:,3);
    %Change the image to gray-scale image.
    diff_im = imsubtract(rgb1, rgb2gray(rgb));
    %use median filter to fill out noise.
    diff_im = medfilt2(diff_im, [3 3]);
    % Change image to binary image
    diff_im = im2bw(diff_im,0.18);

    % fill out all pixel less than 16px
    bw = bwareaopen(diff_im,16);
    % create the structure with the shape we want config with 4 px
    se = strel('disk',4);
    %Crop the gray image by only interesting in the shape we need.
    bw = imclose(bw,se);
    %fill all holes in the shape to make it full.
    bw = imfill(bw,'holes');
    %Focus on the outer boundary. The imfill function can help this too much
    %because it fill the other contour.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[B,L] = bwboundaries(bw,'noholes');
% Traces the boundaries.
% Measures a set of properties for each labeled region in the label
% matrix L;example Positive integer L element set = 1 mean it correspond
% to region 1
stats = regionprops(L,'Area','Centroid');
% define circle threshold
circle_threshold = 0.80;
% Display the image.
imshow(rgb)

hold on

%This is a loop to bound the red objects in a circle.
%define the value of g relate to B
for g = 1:length(B)

    % obtain (X,Y) boundary coordinates corresponding to label 'g'
    boundary = B{g};

    % compute a simple estimate of the object's perimeter
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq,2)));

    % obtain the area calculation corresponding to label 'g'
    area = stats(g).Area;

    % compute the roundness metric
    metric = 4*pi*area/perimeter^2;


    % Find the circle
     if metric > circle_threshold
        centroid = stats(g).Centroid;
        plot(centroid(1),centroid(2),'ko');

     cenx=centroid(1);
     ceny=centroid(2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
X1=x/2+30; % center region of the frame
Y1=y/2+30;
X2=x/2-30;
Y2=y/2-30;

if (cenx>X2 && cenx<X1 && ceny>Y2 && ceny<Y1)
    fprintf(ser,'H');
        fprintf('\nH\n');
        %hover

 elseif (cenx>X2 && ceny<Y1 && cenx<X1)
        fprintf(ser,'F');
        fprintf('\nF\n');
       %forward

 elseif (cenx>X2 && ceny>Y2 && cenx<X1)
        fprintf(ser,'B');
        fprintf('\nB\n');
       %backward

 elseif (ceny<Y1 && cenx>X1 && ceny>Y2)
        fprintf(ser,'R');
        fprintf('\nR\n');
       %right

 elseif (ceny<Y1 && cenx<X2 && ceny>Y2)
        fprintf(ser,'L');
        fprintf('\nL\n');
        %left
elseif (ceny<Y2 && cenx<X2)
     fprintf(ser,'1');
        fprintf('\n1\n');
        %upper left
elseif (ceny<Y2 && cenx>X1)
     fprintf(ser,'2');
        fprintf('\n2\n');
        %upper right
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        elseif (ceny>Y1 && cenx<X2)
            fprintf(ser,'3');
              fprintf('\n3\n');
              %bottom left
        elseif (ceny>Y1 && cenx>X1)
            fprintf(ser,'4');
              fprintf('\n4\n');
              %bottom Right

        end

    else
              [r,c] = find(L(:,:)==g);
              bw(r,c)=0;
        end
    end
        hold off
end

% Stop the video aquisition.
stop(vid);

% Flush all the image data stored in the memory buffer.
flushdata(vid);

% Clear all variables
clear all
 fclose(ser)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งสำหรับบอร์ดอาร์ดุยโน(Arduino)

```
int matlab = 0;
int ESC1 = 3;
int ESC2 = 9;
int ESC3 = 10;
int ESC4 = 11;
void setup() {
Serial.begin(1200);
pinMode(ESC1,OUTPUT);
pinMode(ESC2,OUTPUT);
pinMode(ESC3,OUTPUT);
pinMode(ESC4,OUTPUT);
}

void loop() {
    if (Serial.available() > 0)
    {
     matlab=Serial.read();

    if (matlab == 'H') {
            Hover();
        }
     if (matlab == 'F') {
            Forward();
        }
    if (matlab == 'B') {
            Backward();
        }
    if (matlab == 'L') {
            Left();
    if (matlab == 'R') {
            Right();
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (matlab == '1') {
            UpperLeft();
        }
    if (matlab == '2') {
            UpperRight();
        }
    if (matlab == '3') {
            BottomLeft();
        }
    if (matlab == '4') {
            BottomRight();
        }

}
else
{
 Stop();
}
}

void Hover(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,255);
    analogWrite(ESC3,255); analogWrite(ESC4,255);
    delay(1000);
}
void Forward(void) {
    analogWrite(ESC1,200); analogWrite(ESC2,200);
    analogWrite(ESC3,255); analogWrite(ESC4,255);
    delay(1000);
}
void Backward(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,255);
    analogWrite(ESC3,200); analogWrite(ESC4,200);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(1000);
}
void Left(void) {
    analogWrite(ESC1,200); analogWrite(ESC2,255);
    analogWrite(ESC3,200); analogWrite(ESC4,255);
delay(1000);
}
void Right(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,200);
    analogWrite(ESC3,255); analogWrite(ESC4,200);
delay(1000);
}
void UpperLeft(void) {
    analogWrite(ESC1,200); analogWrite(ESC2,255);
    analogWrite(ESC3,255); analogWrite(ESC4,255);
delay(1000);
}
void UpperRight(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,200);
    analogWrite(ESC3,255); analogWrite(ESC4,255);
delay(1000);
}
void BottomLeft(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,255);
    analogWrite(ESC3,200); analogWrite(ESC4,255);
delay(1000);
}
void BottomRight(void) {
    analogWrite(ESC1,255); analogWrite(ESC2,255);
    analogWrite(ESC3,255); analogWrite(ESC4,200);
delay(1000);
}
void Stop(void) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    analogWrite(ESC1,0); analogWrite(ESC2,0);
    analogWrite(ESC3,0); analogWrite(ESC4,0);
delay(1000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้