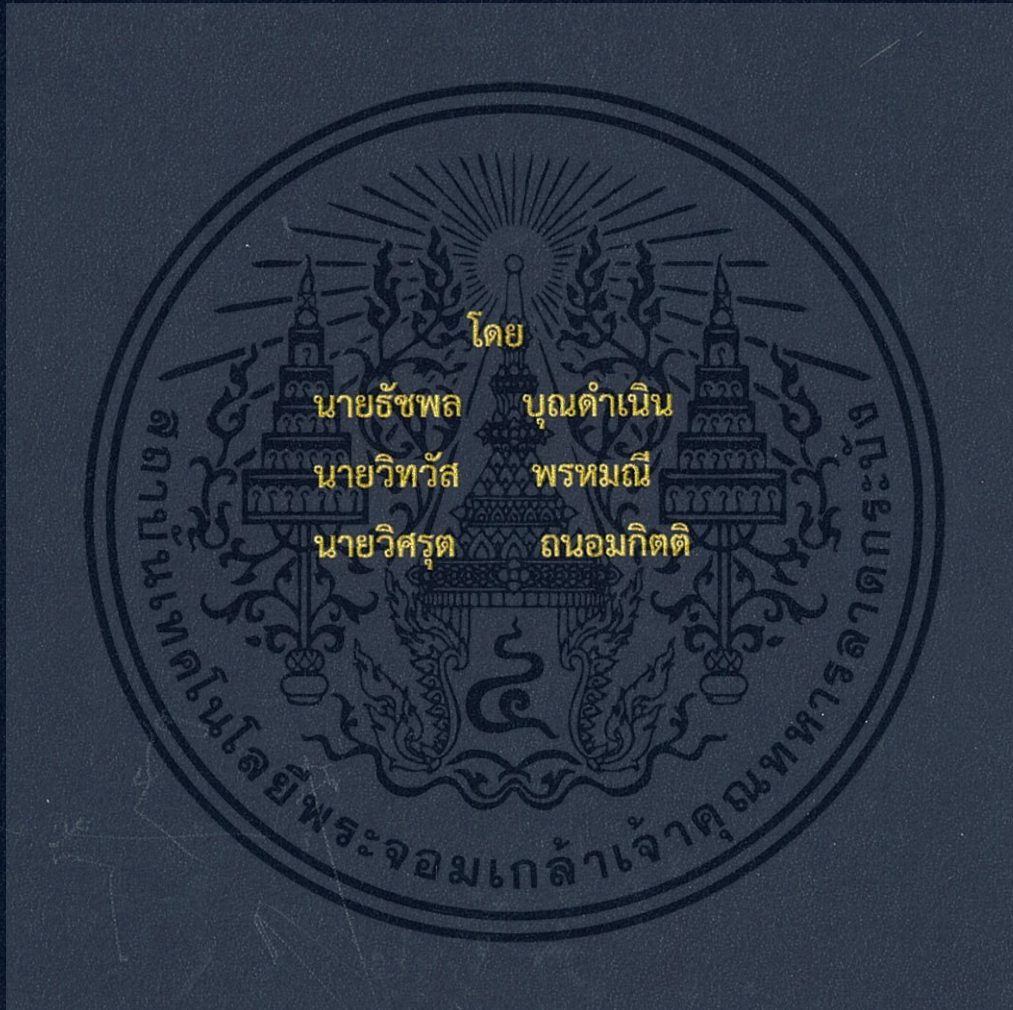


ระบบควบคุมฝูงโดรน
MULTICONTROL DRONES



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

ปริญญาโทปีการศึกษา 2558

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมฝูงโดรน

MULTICONTROL DRONES

ผู้จัดทำ

1. นาย ชัชพล บุญดำเนิน 55010554
2. นาย วิทวัส พรหมณี 55011148
3. นาย วิศรุต ถนอมกิตติ 55011158


.....
(ผศ.ดร. สิริภพ ตูประกาย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินโครงการ “ระบบควบคุมฝูงโดรน” สามารถสำเร็จลุล่วงไปได้ด้วยดีทางคณะผู้จัดทำต้องขอขอบคุณหลายฝ่ายคือ

ขอขอบคุณ ผศ.ดร.สิริภพ ตู้ประกาย อาจารย์ที่ปรึกษาที่ให้ความกรุณาคอยดูแลเอาใจใส่ให้คำแนะนำ แนะนำแนวทางแก้ไขปัญหาในการจัดทำโครงการรวมทั้งสนับสนุนเครื่องมือและอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้ในระหว่างการจัดทำโครงการนี้

ขอขอบคุณสาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การศึกษา และสนับสนุนทุนในการจัดทำโครงการครั้งนี้

ขอขอบคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านทั้งบิดามารดา เพื่อน และรุ่นพี่ทุกคนที่ได้คอยช่วยเหลือสนับสนุน และคอยให้กำลังใจ ตลอดเวลาที่จัดทำโครงการ

นาย ธัชพล

บุญดำเนิน

นาย วิทวัส

พรหมณี

นาย วิศรุต

ถนอมกิตติ

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมฝูงโดรน

MULTICONTROL DRONES

โดย	นาย ชัชพล	บุญดำเนิน	55010554
	นาย วิทวัส	พรหมณี	55011148
	นาย วิศรุต	ถนอมกิตติ	55011158

อาจารย์ที่ปรึกษา ผศ.ดร. สิริภพ ตู้ประกาย

บทคัดย่อ

เนื่องจากในปัจจุบันโดรนสามารถหาซื้อได้ง่ายรวมถึงมีวิธีการใช้งานที่หลายหลายมากขึ้น และมีวิธีการควบคุมไม่ยุ่งยากมากแต่การควบคุมนั้นสามารถทำได้เพียงครั้งละหนึ่งลำเท่านั้น จึงคิดที่จะเพิ่มประสิทธิภาพในการควบคุมโดรนได้หลายลำพร้อมกัน โดยใช้อุปกรณ์ควบคุมเพียงหนึ่งเครื่องโดยสามารถเลือกควบคุมแบบหนึ่งลำหรือหลายลำได้ตามที่ต้องการ และสามารถแสดงภาพจากโดรนหลายลำได้พร้อมกัน ทั้งยังเลือกรูปแบบการใช้งาน เช่น การควบคุมโดรนเฉพาะตัวที่ต้องการ หรือการเลือกดูกล้องของโดรนตัวที่สนใจได้ด้วยการใช้ไมโครคอนโทรลเลอร์ และอุปกรณ์ส่งสัญญาณไร้สายในการควบคุมรูปแบบการทำงานตามต้องการ

ABSTRACT

Nowaday drone can be purchased easily, can be used in various ways and easy to control. The significantly trouble is the drone can be controlled only one drone at a time, so that this project want to control drones have simultaneously using one control device for the purpose of increase efficiency and solve this problem. You can choose to control one of the drone or many can show picture and choose the function that work such as control drone that you need, or choose a camera view of drone as required by the microcontroller software, and wireless transmission of control.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 ส่วนประกอบของ UAV	2
2.2 สัญญาณควบคุม	6
2.3 ARDUINO NANO	7
2.4 โปรแกรม MICROSOFT VISUAL STUDIO	9
2.5 โหมดการบิน (FLIGHT MODES)	12
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	14
3.1 การออกแบบ	14
3.2 เครื่องมือที่ใช้ในการทดลอง	20
3.3 การจัดเก็บผลการทดลอง	21

สารบัญ (ต่อ)

	หน้า
บทที่ 4	
ผลการทดลอง	26
4.1 ทดสอบการควบคุม	26
4.2 ค่าสัญญาณ OUTPUT จาก APM เข้าสู่ MOTOR	28
4.3 ผลการทดสอบโปรแกรมควบคุมรูปแบบการทำงาน	34
4.4 ผลการทดสอบการควบคุมรูปแบบการทำงานผ่านโปรแกรม MICROSOFT VISUAL STUDIO	37

บทที่ 5		41
สรุปผลและข้อเสนอแนะ		
5.1 สรุปผล		41
5.2 ข้อเสนอแนะ		41

บรรณานุกรม		42
-------------------	--	-----------

ภาคผนวก		43
----------------	--	-----------

สารบัญรูป

รูปที่	หน้า
2.1 ลักษณะโครงเครื่องบิน (Airframe) แบบขึ้นลงแนวตั้ง (บน) แบบ 4X, (ล่าง) UAV แบบ 6Y	2
2.2 ลักษณะโครงเครื่องบิน (Airframe) แบบขึ้นลงแนวตั้ง (บน) แบบ 6Y, (ล่าง) UAV แบบ 6X	3
2.3 APM (ArduPilot ATMEGA2560)	3
2.4 Power Module	4
2.5 Motor รุ่น EMAX MT2213-935KV	5
2.6 Speed Control	5
2.7 แบตเตอรี่	6
2.8 ตัวอย่างของสัญญาณ Gaussian	6
2.9 Arduino nano	7
2.10 หน้าที่ในแต่ละขาของ Arduino nano	7
2.11 เลือก I agree... และ Install ตามลำดับ	9
2.12 กำลังติดตั้ง	10
2.13 ติดตั้งเรียบร้อยแล้ว	10
2.14 หน้าจอหลักของโปรแกรม Visual Studio Express 2013 for Windows	11
2.15 การสร้าง Project สามารถคลิกได้ที่เมนู FILE -> New Project	11
2.16 จะเห็นว่าการเขียนโปรแกรมบน Windows Store จะรองรับได้ 4 รูปแบบหลัก ๆ คือ JavaScript , Visual Basic , Visual C# , Visual C++	12
3.1 บล็อกไดอะแกรมของอากาศยานไร้คนขับ	14
3.2 ค่า Angle of Attack ของใบพัดซึ่งวัดค่าได้เท่ากับ 15 องศา	16
3.3 กราฟแสดงความสัมพันธ์ Coefficient of Lift กับ Angle of Attack ซึ่ง Angle of Attack มีค่าเท่ากับ 15 องศาทำให้ค่า Coefficient Lift มีค่าเท่ากับ 1.7	17
3.4 Flowchart แสดงการทำงาน	18
3.5 Flowchart ของการเลือกรูปแบบการทำงาน	19

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 ArduPilot ATMEGA2560	20
3.7 แสดงภาพของรีโมทคอนโทรล	20
3.8 แสดงภาพของโดรน	21
3.9 การวัดสัญญาณ Output จาก APM	21
3.10 การวัดสัญญาณ Output จาก APM โดยใช้ Oscilloscope	22
3.11 ทำการ Connect บอร์ด APM กับ โปรแกรม Mission Planner	23
3.12 ทำการเลือก Tuning	23
3.13 เลือกค่าที่ต้องการวัด	24
3.14 กราฟสัญญาณที่วัดค่าได้	24
3.15 ส่วนแสดงผลของโปรแกรม	25
4.1 ไฟสัญญาณเตือนจากอุปกรณ์	26
4.2 ควบคุมยานบินให้ลอยอยู่กับที่ และเคลื่อนที่ในทิศทางต่าง ๆ	27
4.3 ให้นานบินเคลื่อนที่ไปยังเส้นขอบที่กำหนดไว้ที่ระยะ 200 เมตร	28
4.4 ยานบินบินกลับมายังจุดเริ่มต้น	28
4.5 มอเตอร์แต่ละตัวที่ใช้อ้างอิง	29
4.6 Throttle 0%	29
4.7 ปรับ Throttle 100%	30
4.8 ปรับ Throttle 50%	30
4.9 ปรับ Throttle 50% และปรับ Pitch สูงสุด	31
4.10 ปรับ Throttle 50% และปรับ Pitch ต่ำสุด	31
4.11 ปรับ Throttle 50% และปรับ Roll สูงสุด	32
4.12 ปรับ Throttle 50% และปรับ Roll ต่ำสุด	32
4.13 ปรับ Throttle 50% และปรับ Yaw สูงสุด	33
4.14 ปรับ Throttle 50% และปรับ Yaw ต่ำสุด	33
4.15ก แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของโดรนตัวที่ 1	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.15ข แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของ ของโดรนตัวที่ 2	34
4.16ก แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของ ของโดรนตัวที่ 1	35
4.16ข แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของ ของโดรนตัวที่ 2	35
4.17ก แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของ ของโดรนตัวที่ 1	36
4.17ข แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของ ของโดรนตัวที่ 2	36
4.18 หน้าจอการควบคุมโหมด “OFF”	37
4.19ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1	37
4.19ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2	37
4.20 หน้าจอการควบคุมโหมด “1”	38
4.21ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1	38
4.21ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2	38
4.22 หน้าจอการควบคุมโหมด “2”	39
4.23ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1	39
4.23ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2	39
4.24 หน้าจอการควบคุมโหมด “3”	40
4.25ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1	40
4.25ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2	40

สารบัญตาราง

ตารางที่		หน้า
2.1	Motor for Multi Copter	4
3.1	แสดงค่าความหนาแน่นอากาศ	16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันเทคโนโลยีได้พัฒนาในทุก ๆ ด้านอย่างรวดเร็ว และได้มีบทบาทในชีวิตประจำวันมากมาย เทคโนโลยีหนึ่งที่มีความน่าสนใจคือเทคโนโลยีอากาศยานไร้คนขับ ซึ่งมีหลากหลายรูปแบบเพื่อไปใช้ประโยชน์ในด้านต่าง ๆ เช่นถ่ายภาพหรือวิดีโอ, สำรวจเส้นทาง, สิ่งแวดล้อม, ปัญหาการจราจรและติดตามเป้าหมายต่าง ๆ เป็นต้น

ปัจจุบันอากาศยานไร้คนขับเริ่มมีความนิยมเพิ่มขึ้น เนื่องจากมีราคาอุปกรณ์ที่ถูกกว่าในอดีต อีกทั้งยังสามารถปรับใช้กับลักษณะงานได้หลายประเภท ทำให้ในปัจจุบันมีความต้องการและรูปแบบการใช้งานอากาศยานไร้คนขับมากขึ้นแต่ในการควบคุมสามารถทำได้เพียงครั้งละหนึ่งลำเท่านั้น ทางผู้จัดทำจึงจัดทำโครงการนี้ขึ้น เพื่อเพิ่มประสิทธิภาพในการควบคุมโดรนได้หลายลำพร้อมกันโดยใช้อุปกรณ์ควบคุมหนึ่งเครื่อง

1.2 วัตถุประสงค์

1. ศึกษาและออกแบบระบบควบคุมโดรน
2. ศึกษาและออกแบบระบบควบคุมผ่านคอมพิวเตอร์
3. ศึกษาและออกแบบระบบควบคุมฝูงโดรน

1.3 ขอบเขตของปริญญาานิพนธ์

จากการศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับการดำเนินงานและพัฒนาโครงการนี้ ผู้จัดทำโครงการได้วางแผนขอบเขตของโครงการ ได้ว่าสามารถควบคุมยานบินไร้คนขับ หลายลำพร้อมกันได้ โดยสามารถเลือกควบคุมแบบหนึ่งลำ หรือหลายลำได้ โดยใช้ไมโครคอนโทรเลอร์และอุปกรณ์ส่งสัญญาณไร้สาย เพื่อนำไปใช้งานได้หลากหลายรูปแบบมากขึ้น เราจะศึกษาและตรวจสอบถึงคุณลักษณะและความสามารถของยานบินไร้คนขับ เพื่อนำไปประยุกต์ต่อในการควบคุมและใช้งาน

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ส่วนประกอบของ UAV

2.1.1 โครงเครื่องบิน (Airframe) โครงสร้างอาจมีรูปร่างต่าง ๆ กัน ตามลักษณะชนิดของใบพัดและการจัดวางตำแหน่งมอเตอร์ รวมถึงตามลักษณะการใช้ เช่น 4X, 6X, 8X, 6Y ส่วนวัสดุที่ใช้ก็มีหลายแบบ เช่น โลหะ พลาสติกผสม คาร์บอนไฟเบอร์ผสม และวัสดุตุตกลื่นคลื่นเรดาร์ เป็นต้น ดังรูปที่ 2.1 และรูปที่ 2.2

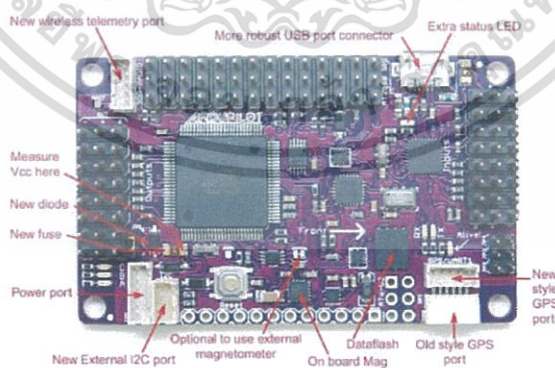


รูปที่ 2.1 ลักษณะโครงเครื่องบิน (Airframe) แบบขึ้นลงแนวตั้ง
(บน) แบบ 4X , (ล่าง) UAV แบบ 6Y [1]



รูปที่ 2.2 ลักษณะโครงเครื่องบิน (Airframe) แบบขึ้นลงแนวตั้ง
(บน) แบบ 6Y , (ล่าง) UAV แบบ 6X [1]

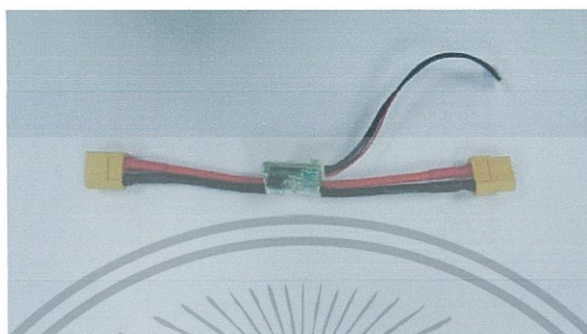
2.1.2 บอร์ดคอนโทรล (Flight control) ในที่นี้จะใช้บอร์ด ArduPilot ATMEGA2560 ทำหน้าที่ในการประมวลผลคำสั่งร่วมกับอุปกรณ์เซนเซอร์เพื่อควบคุมการบิน ตลอดจนปรับตั้งค่าต่าง ๆ ของอากาศยาน ดังรูปที่ 2.3



รูปที่ 2.3 APM (ArduPilot ATMEGA2560) [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 Power Module จะใช้โมดูลจ่ายไฟทำการแปลงไฟขนาด 11.1 Volt ลงมาเหลือขนาด 5 Volt ควบคุมการจ่ายไฟที่เข้าสู่บอร์ด APM และ ESC รวมถึง Sensor ต่าง ๆ ของตัวอากาศยาน ดังรูปที่ 2.4



รูปที่ 2.4 Power Module [1]

2.1.4 มอเตอร์ (Brushless Motor) ในการเลือกใช้มอเตอร์จะต้องคำนวณน้ำหนักของตัวล้อย่างคร่าวๆ เช่น ตัวล้อยี่จะมีน้ำหนักตกที่ประมาณ 2.5-3.0 kg จากตารางการทดสอบของบริษัทผู้ผลิตมอเตอร์ที่ให้กำลังในการยกสามารถยกตัวล้อยากาศยานขึ้นบิน และประหยัดเพื่อให้ได้ระยะเวลาการบินนานจึงใช้มอเตอร์รุ่น EMax MT2213-935KV แรงดันไฟฟ้า 11.1 Voltage ใช้ใบพัดขนาด 1045 ดังตารางที่ 2.1 และรูปที่ 2.5

ตารางที่ 2.1 Motor for Multi Copter

Motor for Multi Copter			
Model Type	Lipo	RPM/V	Propeller
2213	3S	935	1045
2213	4S	935	8045
2215	3S	1020	1045
2215	4S	1020	8045
2218	3S	950	1145
2218	4S	950	9045

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multi copter motor MT2213-935KV (1045Combo)



KV.....935
 WELGHT.....55G
 DIAMETER.....27.9MM
 LENGHT(MM).....39.7MM
 3S(LI-PO).....1045(PROP)
 MAX THRUST.....850G
 ESC recommended.....18A



รูปที่ 2.5 Motor รุ่น EMAX MT2213-935KV [1]

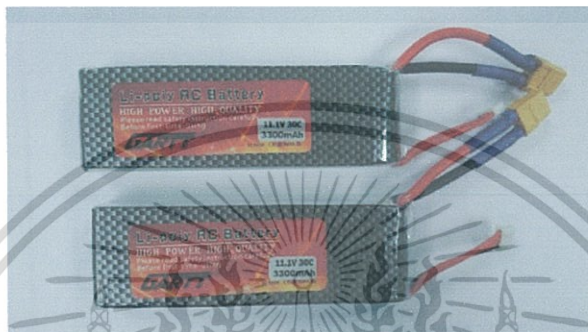
2.1.5 สปีดคอนโทรล (Speed Control) ที่ใช้ในการควบคุมความเร็วของมอเตอร์ จากตารางการทดสอบของมอเตอร์กับขนาดของใบที่จะใช้ โดยจะมีขาสัญญาณควบคุมที่ต่อเข้ากับตัวบอร์ด ArduPilot ATMEGA2560และขาที่ต่อกับไฟเลี้ยงจากแบตเตอรี่ดังรูปที่ 2.6



รูปที่ 2.6 Speed Control [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 แบตเตอรี่ (Battery) เลือกใช้แบตเตอรี่ลิเทียมโพลีเมอร์ (Li-Po Battery) ที่มีอัตราการจ่ายกระแสไฟที่สูงพอและเหมาะสมกับสเปคของมอเตอร์ที่ใช้ตามตารางการทดสอบ มีปริมาณความจุมากเพียงพอและน้ำหนักไม่เยอะเกินไปเพื่อให้อากาศยานบินได้นานที่สุด โดยควรพิจารณาตำแหน่ง CG ให้มากที่สุด อาจวางไว้ส่วนบนหรือแขวนในตำแหน่งส่วนกลางและล่างสุดของตัวลำดังรูปที่ 2.7

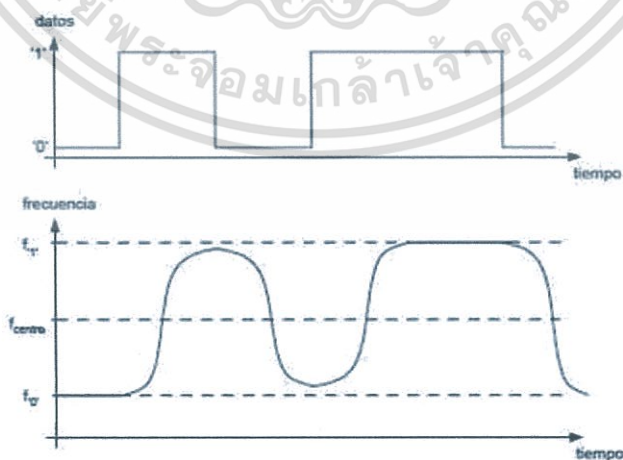


รูปที่ 2.7 แบตเตอรี่ [1]

2.2 สัญญาณควบคุม

2.2.1 สัญญาณ Gaussian

GFSK เป็นประเภทของFSK โดยใช้ตัวกรอง Gaussian ให้เพื่อทำให้สัญญาณการเบี่ยงเบนความถี่เรียบขึ้นในเชิงบวก/เชิงลบ ซึ่งเป็นตัวแทนของไบนารี 1 หรือ 0 มักถูกใช้โดย DECT, Bluetooth, CypressWirelessUSB, นอร์ดิกเซมิคอนดักเตอร์, Texas Instruments LPRF, Z-Wave และอุปกรณ์ Wavenis สำหรับอัตราการส่งข้อมูลลูทอพื้นฐานส่วนเบี่ยงเบนขั้นต่ำที่ 115 เฮิรท์

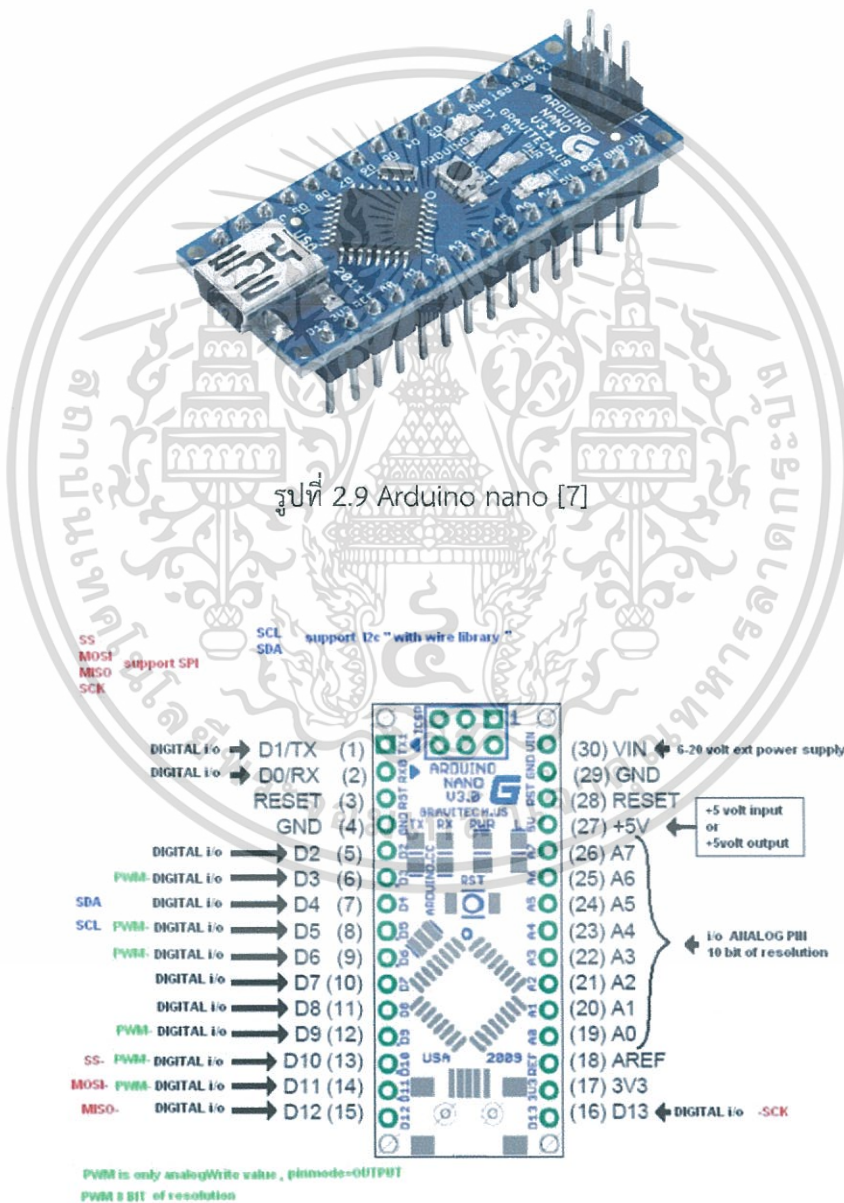


รูปที่ 2.8 ตัวอย่างของสัญญาณ Gaussian [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Arduino Nano

Arduino Nano 3.0 เป็น Arduino ที่ใช้หน่วยประมวลผลผล ATmega328 เช่นเดียวกับ Arduino Uno ความสามารถจึงเท่ากัน แตกต่างที่ Arduino Nano 3.0 ถูกออกแบบให้มีขนาดเล็ก โดยตัดส่วนของ Socket ที่ไม่จำเป็นออก และยังคงความสามารถในการติดต่อผ่าน USB port เหมือนบอร์ด Arduino ตัวใหญ่ไว้ อาจจะเรียกได้ว่า Arduino Nano 3.0 ตัวนี้ คือ Arduino Uno ขนาดย่อส่วนลงมา



รูปที่ 2.10 หน้าที่ในแต่ละขาของ Arduino nano [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียด

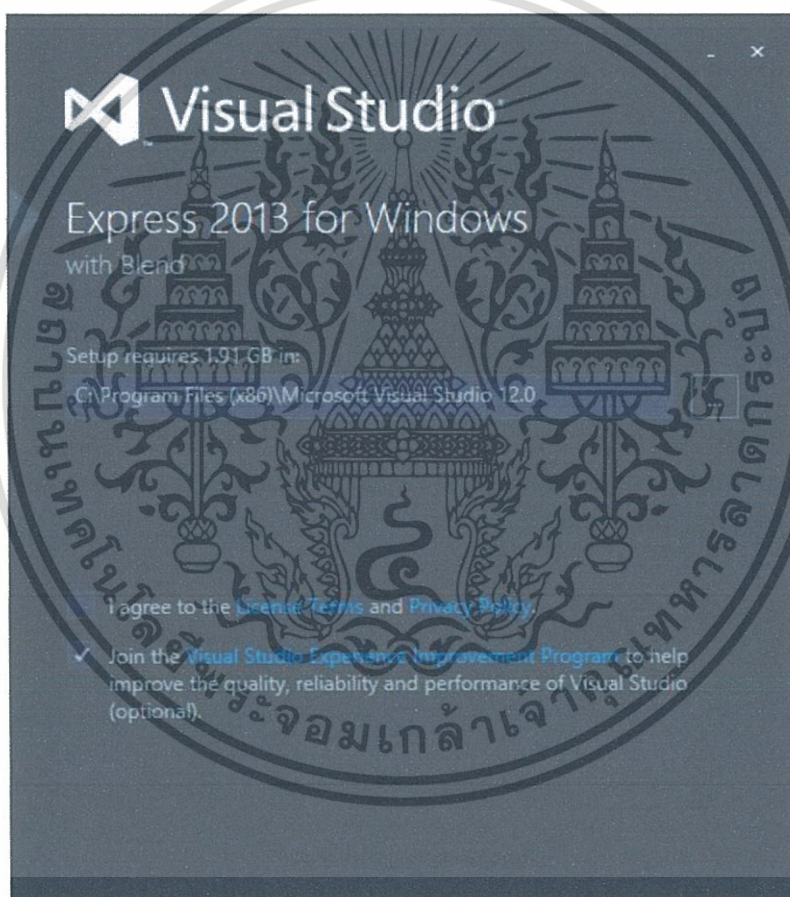
1. ไมโครคอนโทรลเลอร์	ATmega328
2. แหล่งจ่ายไฟ	5V
3. ไฟเข้า(แนะนำ)	7-12V
4. ไฟเข้า (จำกัดไว้ที่)	6-20V
5. ขาดิจิตอล I/O	14 ขา (6 รองรับเอาต์พุตแบบ PWM)
6. ขาอะนาล็อกอินพุต	8 ขา
7. กระแสไฟฟ้า DC ต่อขา I/O	40 mA
8. กระแสไฟฟ้าออก DC สำหรับขา 3.3V	50 mA
9. Flash Memory	32 KB (ATmega328)
10. SRAM	2 KB (ATmega328)
11. EEPROM	1 KB (ATmega328)
12. Clock Speed	16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โปรแกรม Microsoft Visual Studio

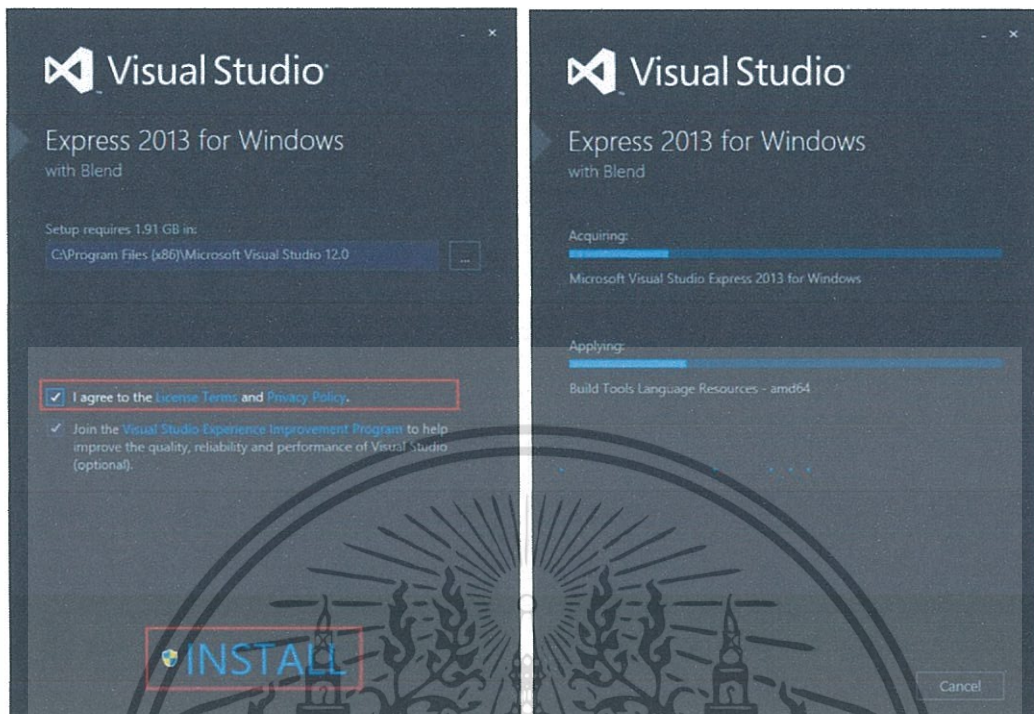
Microsoft Visual Studio คือ Integrated Development Environment พัฒนาขึ้นโดย Microsoft ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ ฟ็อคเกตพีซี Smartphone และ เว็บเบราร์เซอร์ ในปัจจุบัน วิชาลสตูดิโอั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษาคอตเน็ตในโปรแกรมเดียวกัน เช่น VB.NET C++ C# J# เป็นต้น

การติดตั้งโปรแกรม

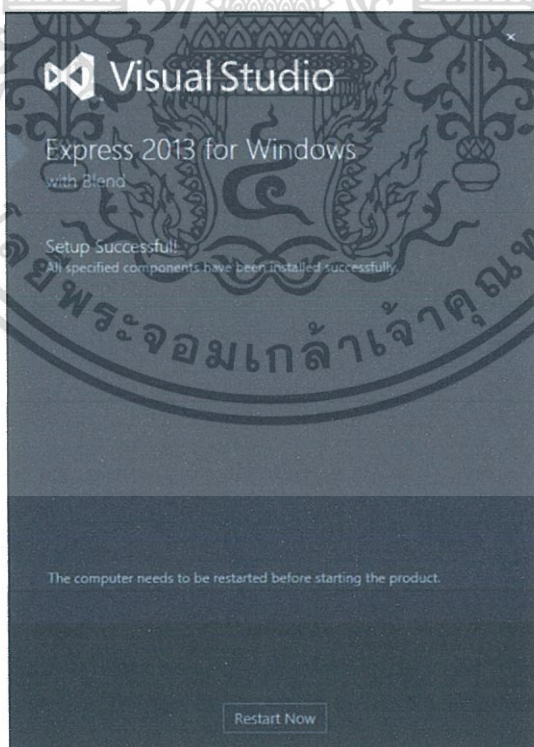


รูปที่ 2.11 เลือก I agree... และ Install ตามลำดับ [12]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



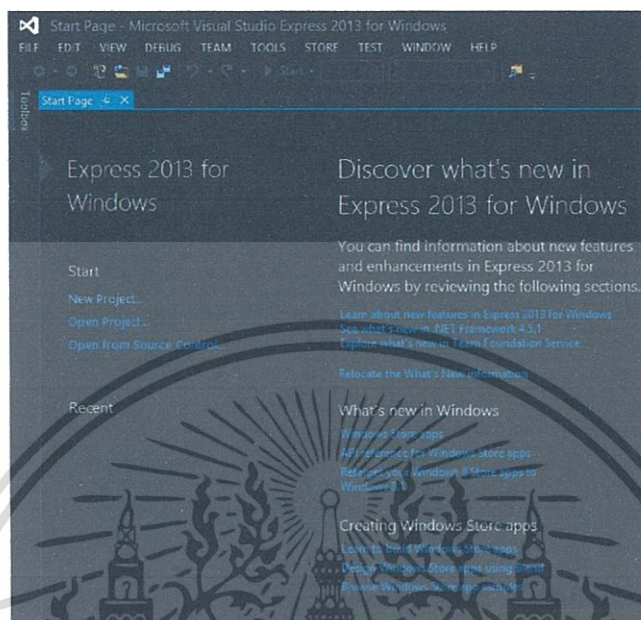
รูปที่ 2.12 กำลังติดตั้ง [12]



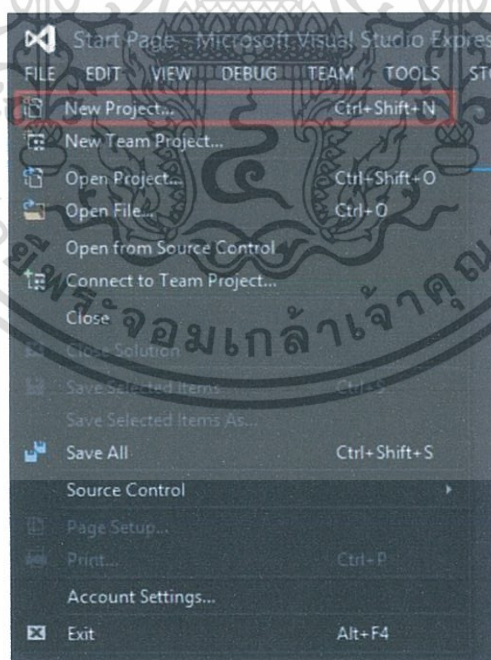
รูปที่ 2.13 ติดตั้งเรียบร้อยแล้ว [12]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Windows Store Apps บน Visual Studio Express 2013 for Windows

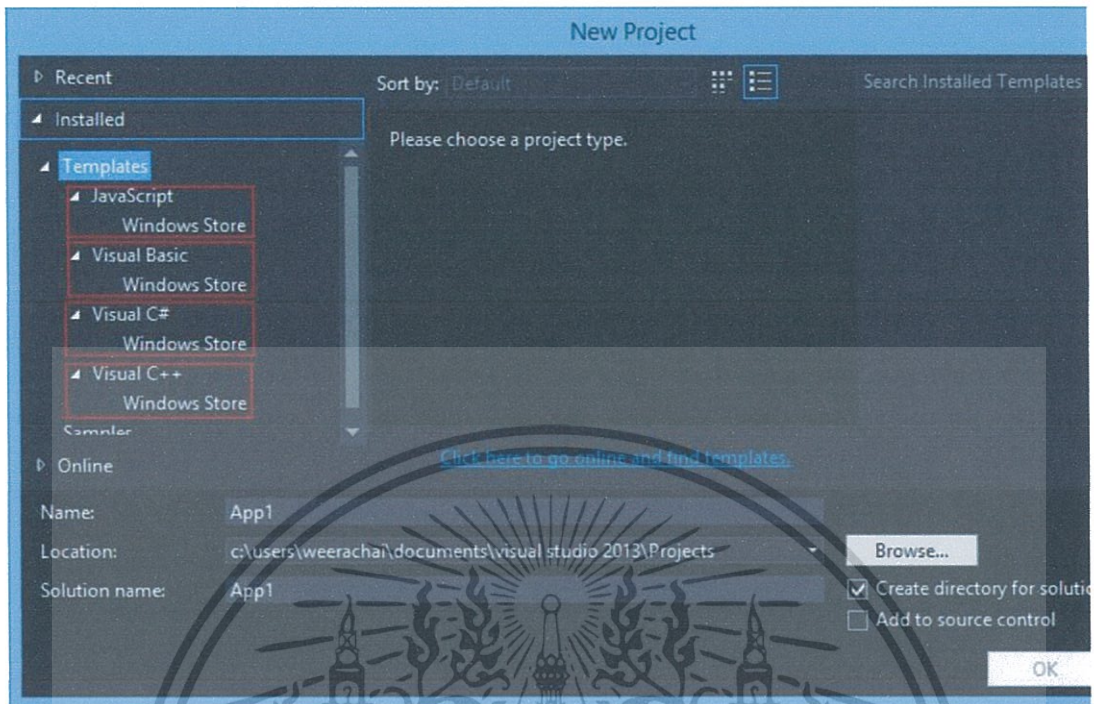


รูปที่ 2.14 หน้าจอหลักของโปรแกรม Visual Studio Express 2013 for Windows [12]



รูปที่ 2.15 การสร้าง Project สามารถคลิกได้ที่เมนู FILE -> New Project [12]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 จะเห็นว่าการเขียนโปรแกรมบน Windows Store จะรองรับได้ 4 รูปแบบหลัก ๆ
คือ JavaScript , Visual Basic , Visual C# , Visual C++ [12]

2.5 โหมดการบิน (Flight Modes)

โดยโหมดการบินของหุ่นยนต์นั้น ปัจจุบันมีทั้งหมด 8 โหมด คือ

1. Acro คือโหมดบินแบบ Aerobatic จะไม่มีการปรับความเอียงกลับมายังแนวระดับอัตโนมัติ มีเพียงการป้องกันไม่ให้เครื่องเอียงจนเกิดการพลิกคว่ำได้
2. Stabilize คือโหมดบินแบบ Stabilize Mode เป็น Mode ในการควบคุมที่ พื้นฐานที่สุด จะทำหน้าที่คอยปรับความเอียงกลับมายังแนวระดับอัตโนมัติ
3. Simple คือโหมดพื้นแบบเดียวกับ Stabilize แต่มุมเอียง Roll กับ Pitch จะเอียงรักษาการเคลื่อนที่ไปในทิศทางต่าง ๆ ได้ง่ายกว่า Stabilize
4. AltHold เป็นโหมดรักษาความสูงอัตโนมัติ ซึ่งระบบควบคุมจะควบคุม Throttle ให้เครื่องบินรักษาระดับ แต่ยังสามารถควบคุมทิศทาง (Roll, Pitch, Yaw) ได้อยู่โดยโหมดนี้ควรต้องมี Sonar (Ultrasonic Sensor) มาใช้ในการตรวจจับความสูงด้วย
5. Loiter เป็นโหมดที่พัฒนามาจาก AltHold โดยเป็นโหมดที่รักษาความสูงและตำแหน่งด้วยตัวเอง โดยใช้ GPS ในการ set ตำแหน่งล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. RTL (Return to Launch) คือโหมดเรียก UAV กลับมายังตำแหน่งแรกที่ GPS ตรวจจับตำแหน่งแรกได้ โดยสามารถที่จะกำหนดความสูงที่ตำแหน่งเดิมได้

7. Guided เป็นโหมดที่สั่งให้ตัวลำ UAV บินไปยังจุด Waypoint ที่เรากำหนดใน Ground Station และเมื่อไปถึงตำแหน่งนั้นแล้ว ระบบจะทำการไปโหมด Loiter เองอัตโนมัติ

8. Auto เป็นโหมดที่บินอัตโนมัติ โดยจะบินไปยังจุด Waypoint ที่เรากำหนดไว้เป็นจุด ๆ ไป และในแต่ละ Waypoint ยังสามารถกำหนดภารกิจให้ Robot บินในลักษณะต่าง ๆ ได้ เช่น ไปถึงแล้ว กลับ รอเวลาในโหมด Loiter หมุนตัว ลงจอด ปรับตำแหน่ง ถ่ายรูป ฯลฯ



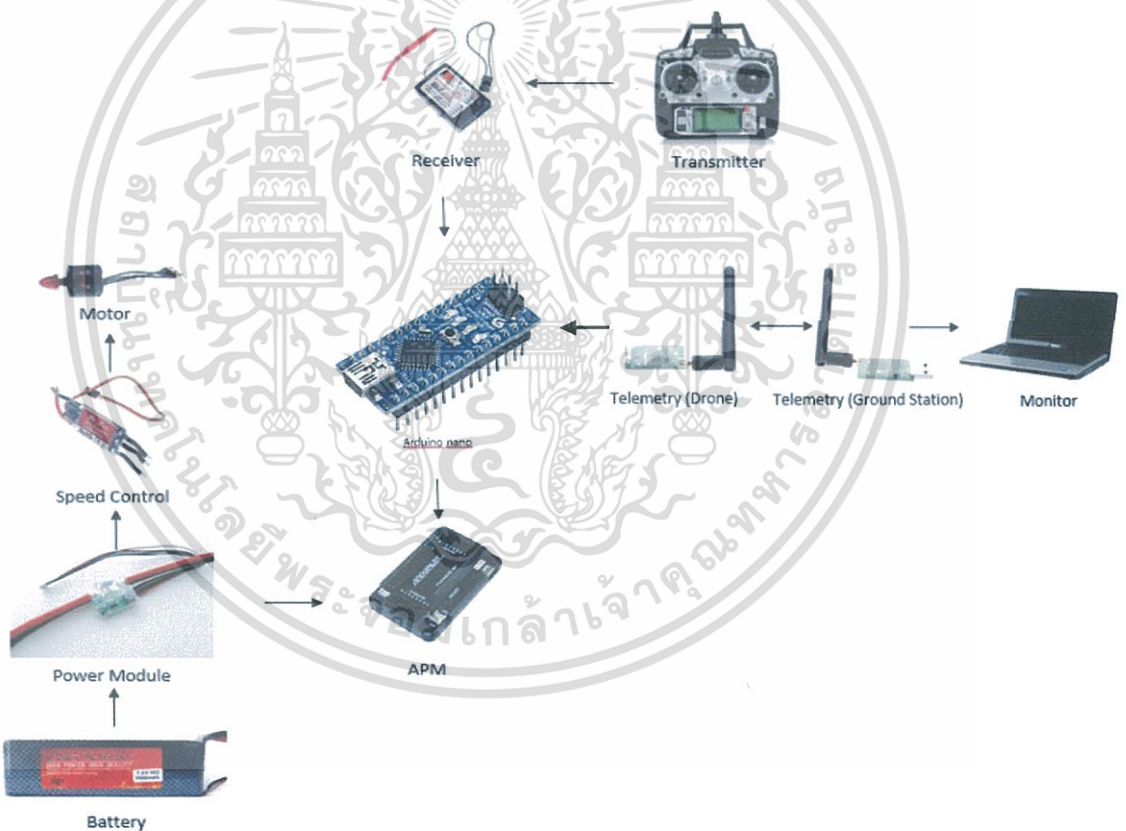
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญาานิพนธ์

3.1 การออกแบบ

แนวคิดในการออกแบบอากาศยานไร้คนขับที่สามารถควบคุมได้หลายลำ โดยจะมีบอร์ดไมโครคอนโทรลเลอร์คอยควบคุม โดยควบคุมผ่าน APM เพื่อควบคุม Drone และ Calibrate รวมถึงแสดงสถานะต่าง ๆ ส่วนอากาศยานไร้คนขับจะรับสัญญาณจากตัว Transmitter และนำไปสั่งการ Speed Control ที่เชื่อมต่อกับใบพัด โดรนจะทำการออกแบบ เพื่อให้สามารถควบคุมตัว อากาศยานได้โดยไม่ต้องใช้ตัวควบคุมเพียงตัวเดียว ดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของอากาศยานไร้คนขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การคำนวณค่าแรงยกตัวของอากาศยานไร้คนขับ

$$\text{จากสูตร} \quad F_L = C_L \times A \times d \times \frac{v^2}{2}$$

ค่า d คือ density of air ที่อุณหภูมิ +30 องศาเซลเซียส = 1.164

ค่า C_L คือ ค่าสัมประสิทธิ์แรงยกตัว (Lift Coefficient)

ซึ่งจะหาค่า ω_m จากสูตร $RPM \times v$ จะได้ ; ค่า $RPM = 935 \text{ RPM/v}$ คือค่ามอเตอร์หมุนต่อรอบ

$$\omega_m = 935 \times 11.1 = 10378.5 \text{ RPM}$$

เปลี่ยนจาก rpm เป็น rad/s

$$\omega_m = \frac{10378.5 \times 2\pi}{60} = 1086.834 \text{ rad/s}$$

จากนั้นหาค่าความเร็ว (v) จาก

$$\begin{aligned} v &= \omega_m r \\ &= 1086.834 \times 0.065 \\ &= 70.644 \text{ m/s} \end{aligned}$$

ค่า r คือความยาวของส่วนของใบพัด 1 ใบ

จากนั้นหาค่าพื้นที่ของใบพัดจาก

$$\begin{aligned} A &= \text{ChordLength} \times \text{WingLength} \\ &= 0.065 \times 0.025 \\ &= 1.625 \times 10^{-3} \text{ m}^2 \end{aligned}$$

นำเข้าสู่สูตรการหาแรงยกตัว

$$\begin{aligned} F_L &= C_L \times A \times d \times \frac{v^2}{2} \\ &= 1.7 \times 1.625 \times 10^{-3} \times 1.164 \times \frac{70.644^2}{2} \\ &= 8.024 \text{ N} \end{aligned}$$

อากาศยานไร้คนขับใช้ใบพัดทั้งหมด 4 ใบพัด จะได้

$$8.024 \times 4 = 32.096 \text{ N}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

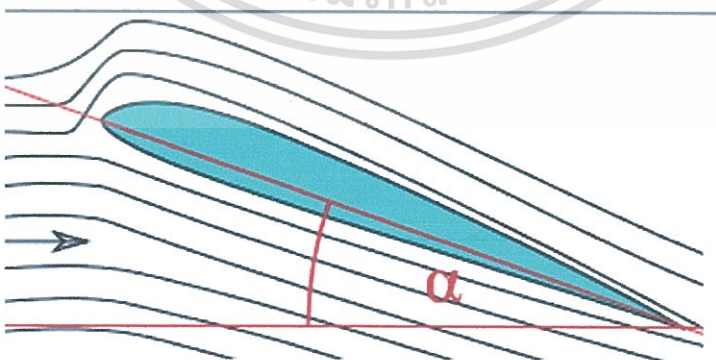
จากนั้นแปลงจาก N เป็น kg จะได้ น้ำหนักที่มากที่สุดที่สามารถยกตัวขึ้นได้

$$\frac{32.096}{9.8} = 3.275kg$$

ตารางแสดงความสัมพันธ์ของอุณหภูมิที่มีผลต่อคุณลักษณะของอากาศซึ่งจะนำค่าที่อุณหภูมิห้อง (25 องศาเซลเซียส) มาใช้หาค่า ความหนาแน่นของอากาศหรือค่า ρ ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงค่าความหนาแน่นอากาศ

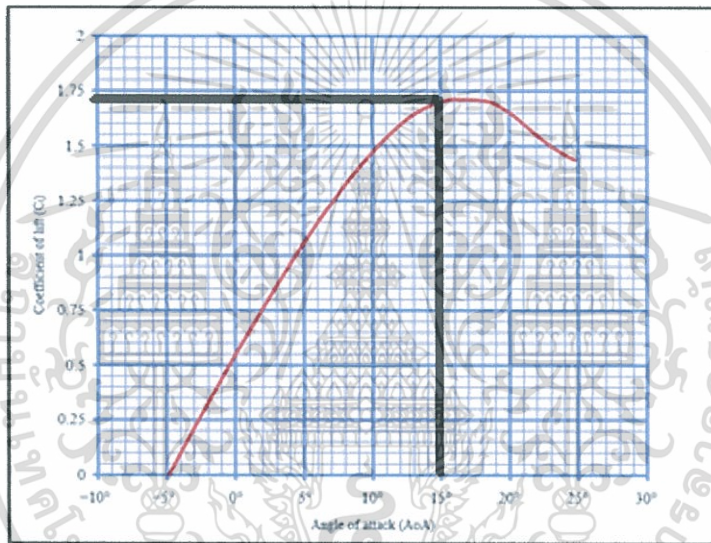
Effect of temperature on properties of air			
Temperature T ($^{\circ}C$)	Speed of sound c ($m \cdot s^{-1}$)	Density of air ρ ($kg \cdot m^{-3}$)	Characteristic specific acoustic impedance z_0 ($Pa \cdot m^{-1} \cdot s$)
+35	351.88	1.1455	403.2
+30	349.02	1.1644	406.5
+25	346.13	1.1839	409.4
+20	343.21	1.2041	413.3
+15	340.27	1.2250	416.9
+10	337.31	1.2466	420.5
+5	334.32	1.2690	424.3
0	331.30	1.2922	428.0



รูปที่ 3.2 ค่า Angle of Attack ของใบพัดซึ่งวัดค่าได้เท่ากับ 15 องศา

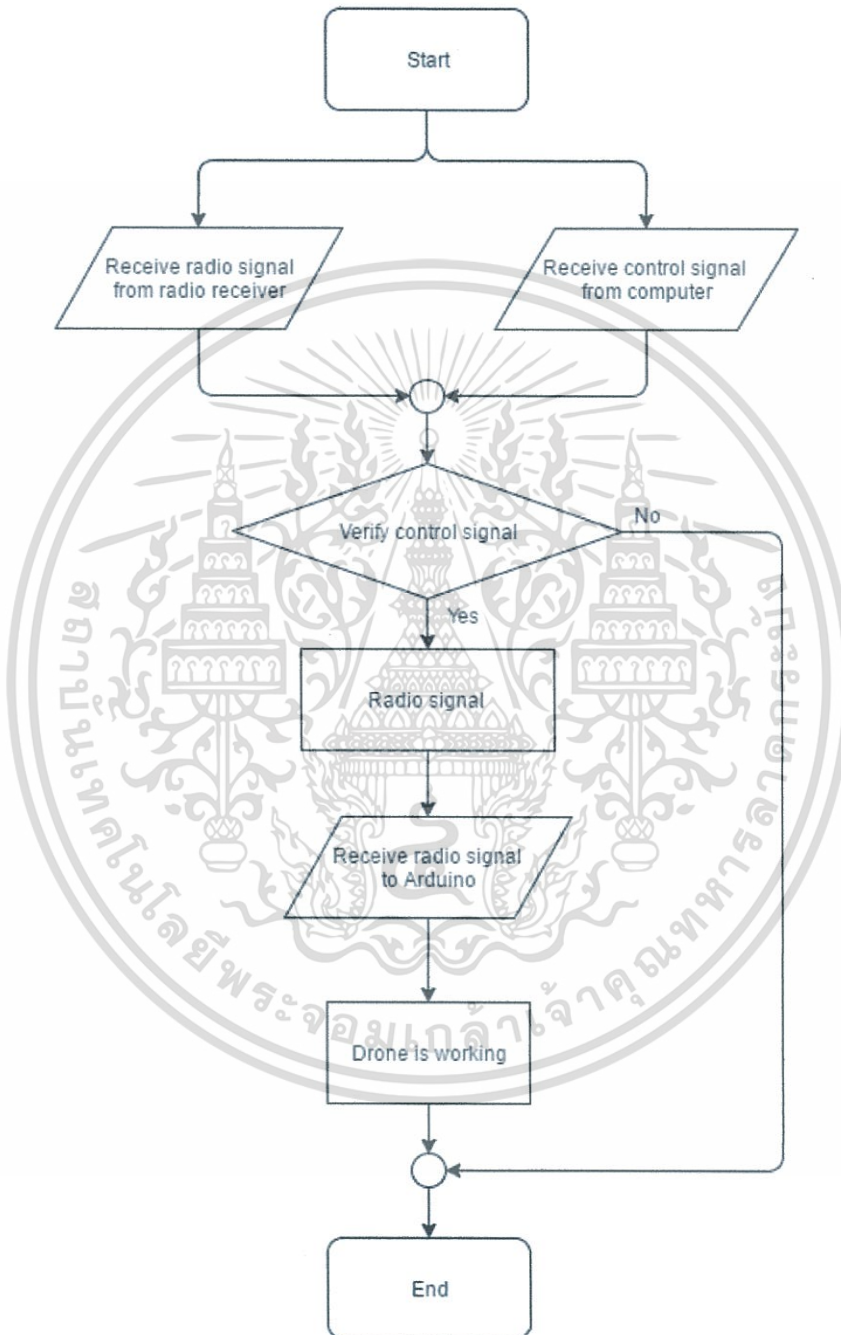
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 คือมุมที่เกิดจากส่วนของด้านหลังของใบพัดกับระนาบการหมุนของใบพัดเป็นมุมที่เกิดขึ้นตลอดระยะเวลาความยาวของใบพัดตั้งแต่โคนถึงปลายใบพัดจะมีค่าไม่เท่ากัน เหตุผลในการที่มุมตลอดใบพัดมีค่าต่างกัน เพราะวาระยะความยาวของใบพัดจากแกนศูนย์กลางการหมุนไม่เท่ากัน ทำให้ความเร็วของใบพัดแต่ละส่วนมีความเร็วไม่เท่ากันซึ่งมีผลกับแรงที่เกิดขึ้น ดังนั้นการออกแบบจึงต้องการให้ทุก ๆ ส่วนของใบพัดมีมุม Angle of Attack ของส่วนของตัวเองที่มีประสิทธิภาพมากที่สุดที่สามารถสร้างแรงดันตามความเร็วรอบที่ออกแบบมาแล้วจึงนำมาเขียนเป็นกราฟแสดงความสัมพันธ์ Coefficient of Lift กับ Angle of Attack ในรูปที่ 3.3 ซึ่งค่า Coefficient of Lift คิดจากคิตส์ส่วนพื้นที่ต่อความเร็ว



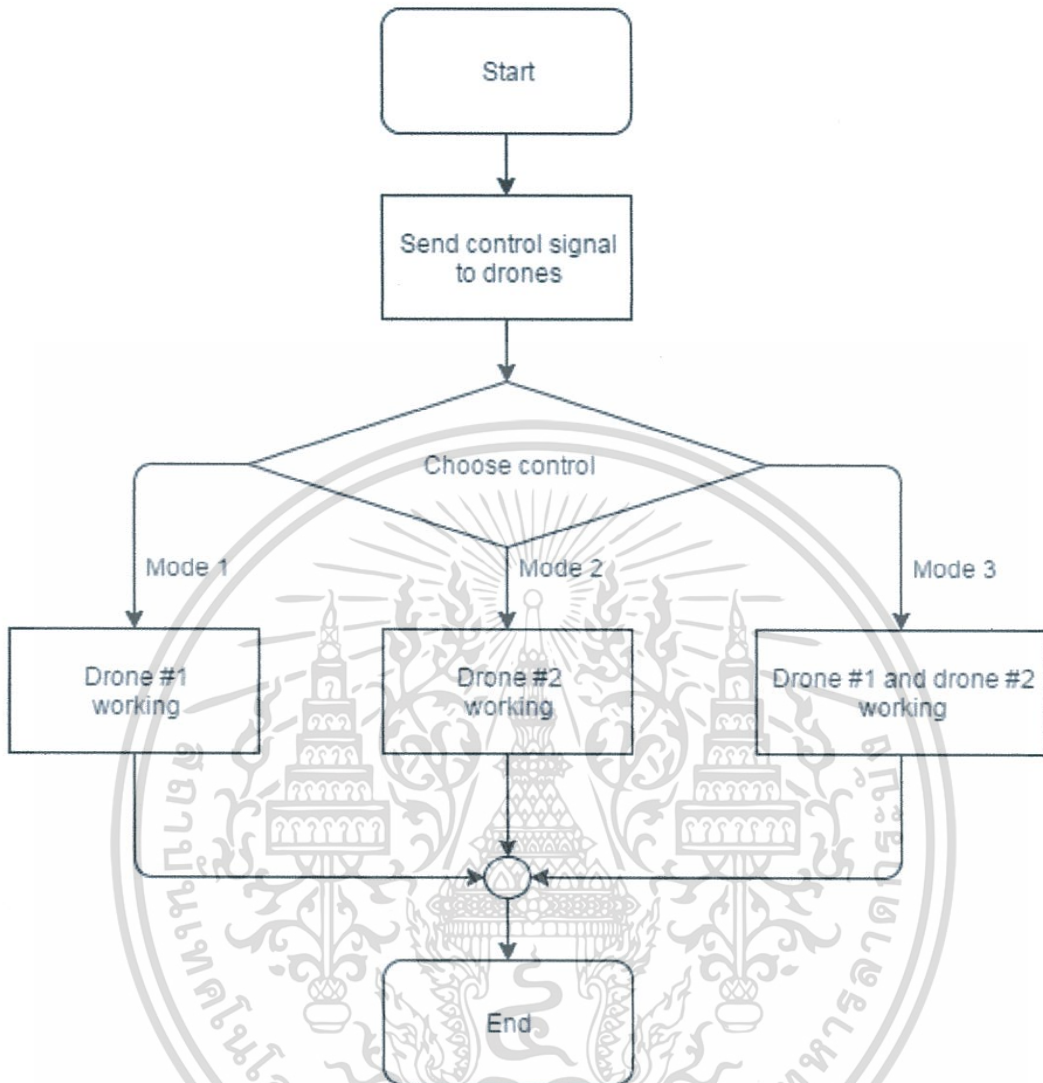
รูปที่ 3.3 กราฟแสดงความสัมพันธ์ Coefficient of Lift กับ Angle of Attack ซึ่ง Angle of Attack มีค่าเท่ากับ 15 องศาทำให้ค่า Coefficient Lift มีค่าเท่ากับ 1.7

3.1.2 Flowchart แสดงการทดสอบการสั่งการข้อมูลและการเลือกรูปแบบ แสดงตั้ง
รูปที่ 3.4 และรูปที่ 3.5



รูปที่ 3.4 Flowchart แสดงการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Flowchart ของการเลือกรูปแบบการทำงาน

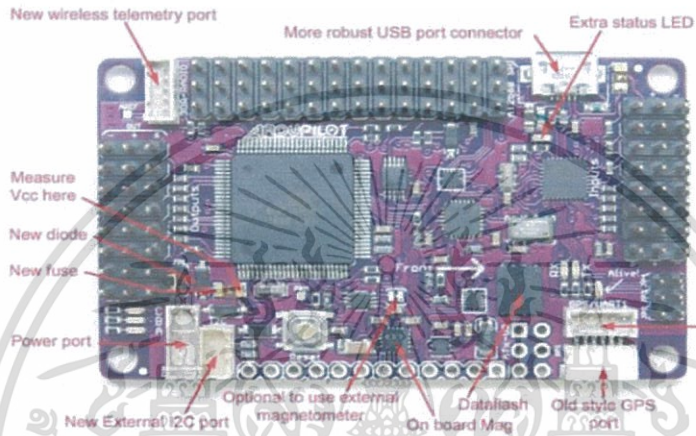
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ส่วนของ Motor

ในส่วนของ Motor เริ่มจากแบตเตอรี่ขนาด 11.1 V 3300 mA ป้อนไฟไปยัง Power Module ผ่านไปยัง Speed Control เพื่อทำการขับให้มอเตอร์ทำงานต่อไป

3.2.2 ส่วนของ APM แสดงดังรูปที่ 3.6



รูปที่ 3.6 ArduPilot ATMEGA2560

3.2.3 ส่วนของ hardware โดยอื่น ๆ เช่น รีโมทคอนโทรลที่ใช้สำหรับควบคุมการดำเนินการของตัวโดรนที่เราใช้โดยไม่ใช่สายไฟเป็นตัวส่งสัญญาณแต่ใช้สัญญาณวิทยุแทน และใช้พลังงานจากแบตเตอรี่ทั้งยังมีขนาดเหมาะสมมือ ทั้งยังมีตัวส่งและรับสัญญาณ(Carrier) เพื่อถอดหรือรับรหัสที่ถูกส่งมาใช้ควบคุมอุปกรณ์ต่าง ๆ และสุดท้ายคือโดรนที่เรานำมาใช้งาน ดังรูปที่ 3.7 และ 3.8 ตามลำดับ



รูปที่ 3.7 แสดงภาพของรีโมทคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

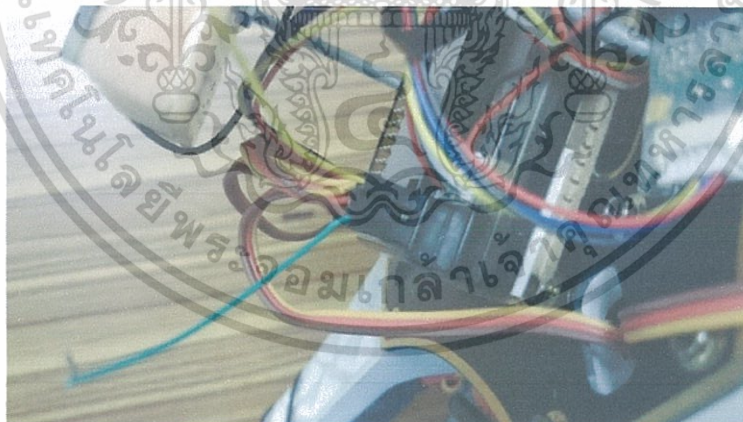


รูปที่ 3.8 แสดงภาพของโดรน

3.3 การจัดเก็บผลการทดลอง

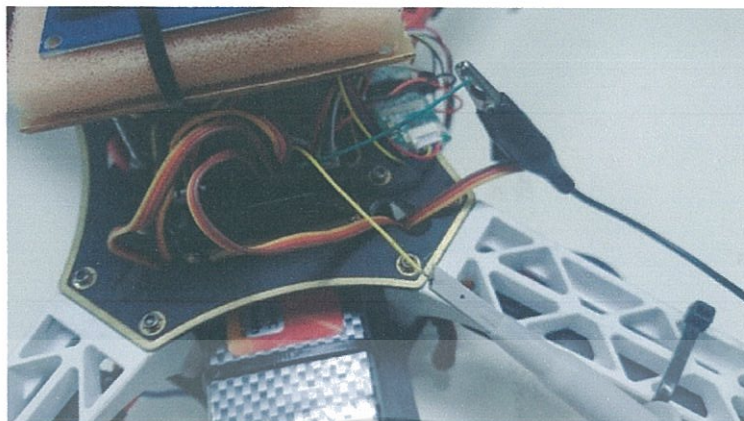
3.3.1 การวัดสัญญาณ Output จาก APM ไปยัง ESC

นำเอาสาย Jump ต่อจากช่องสัญญาณที่ต้องการวัดและช่อง Ground โดยไม่มีการ Jump สายจากช่องไฟเลี้ยง (ช่องกลาง) จากนั้นนำเอา Oscilloscope มาวัดโดยนำเอาขา + วัดที่สายสัญญาณ และขา - วัดที่ขา Ground ดังรูปที่ 3.9 และรูปที่ 3.10



รูปที่ 3.9 การวัดสัญญาณ Output จาก APM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การวัดสัญญาณ Output จาก APM โดยใช้ Oscilloscope

3.3.2 จัดเก็บข้อมูลการเคลื่อนที่ในแนวแกนนอน

1. ทำการต่อแบตเตอรี่เข้าสู่ Drone
2. ทำการเชื่อมต่อกับรีโมทคอนโทรล
3. เลือกความสูงที่เหมาะสมและระยะห่างตามที่ต้องการเก็บผล
4. เดินให้ระยะห่างจาก Drone จนกระทั่ง Drone เริ่มมีการเคลื่อนที่
5. ปรับระยะห่างอื่น ๆ ตามลำดับ
6. สังเกตและบันทึกผลการทดลอง

3.3.3 จัดเก็บข้อมูลการเคลื่อนที่ในแนวแกนตั้ง

1. ทำการต่อแบตเตอรี่เข้าสู่ Drone
2. ทำการเชื่อมต่อกับรีโมทคอนโทรล
3. เลือกความสูงที่เหมาะสมและระยะห่างตามที่ต้องการเก็บผล
4. เดินให้ระยะห่างจาก Drone จนกระทั่ง Drone เริ่มมีการเคลื่อนที่
5. ปรับระยะห่างอื่น ๆ ตามลำดับ
6. สังเกตและบันทึกผลการทดลอง

หมายเหตุ : จากการทดลองควบคุมโดรน จะแบ่งเป็น 3 กรณี คือ

1. ใช้รีโมทคอนโทรลควบคุมโดรนลำที่ 1 เพียงลำเดียว
2. ใช้รีโมทคอนโทรลควบคุมโดรนลำที่ 2 เพียงลำเดียว
3. ใช้รีโมทคอนโทรลควบคุมโดรนทั้งสองลำพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การวัดความกว้าง Pulse โดยใช้ โปรแกรม Mission Planner

ทำการ Connect บอร์ด APM กับตัวโปรแกรมโดยใช้ Baud Rate ให้ถูกต้องโดยใช้ 57600 bps (สำหรับ Telemetry) และ 115200 bps (สำหรับ USB) ดังรูป 3.11 หลังจากนั้นเลือกช่อง Tuning ด้านล่างโปรแกรมดังรูป 3.12 จะขึ้นค่าพารามิเตอร์สำหรับการวัดค่าโปรแกรมให้เลือก หลังจากทำการเลือกเสร็จทำการกดตกลงดังรูป 3.13 จะขึ้นกราฟสำหรับวัดค่าดังรูป 3.14

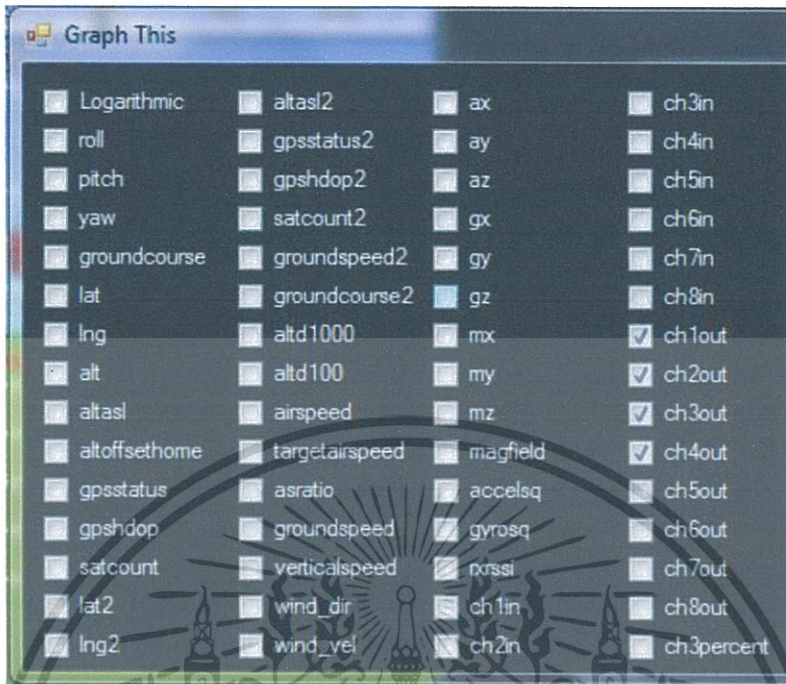


รูปที่ 3.11 ทำการ Connect บอร์ด APM กับ โปรแกรม Mission Planner



รูปที่ 3.12 ทำการเลือก Tuning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 เลือกค่าที่ต้องการวัด



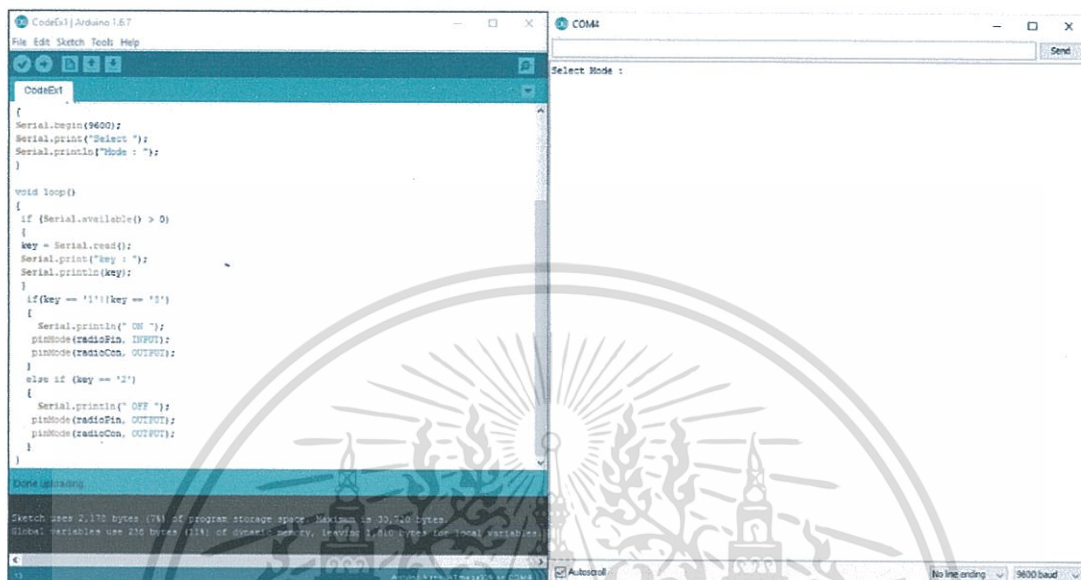
รูปที่ 3.14 กราฟสัญญาณที่วัดค่าได้

หมายเหตุ : ใช้ในการวัดสัญญาณที่ไม่สามารถวัดค่าได้โดยใช้ Oscilloscope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 ทดสอบรูปแบบการทำงาน โดยใช้โปรแกรมควบคุม

ส่วนของการประมวลผลจะแสดงผลผ่านหน้าจอซีเรียลมอนิเตอร์ ดังรูปที่ 3.15



```

Code[1]: Arduino 1.6.7
File Edit Sketch Tools Help

CodeEdit
[
Serial.begin(9600);
Serial.print("Select ");
Serial.println("Mode : ");
]

void loop()
{
  if (Serial.available() > 0)
  {
    key = Serial.read();
    Serial.print("key : ");
    Serial.println(key);

    if(key == '1' || key == '0')
    {
      Serial.println(" ON ");
      pinMode(radioPin, INPUT);
      pinMode(radioCon, OUTPUT);
    }
    else if (key == '2')
    {
      Serial.println(" OFF ");
      pinMode(radioPin, OUTPUT);
      pinMode(radioCon, OUTPUT);
    }
  }
}

Done uploading.

Sketch uses 2,170 bytes (7%) of program storage space. Maximum is 30,720 bytes.
Global variables use 230 bytes (11%) of dynamic memory, leaving 1,610 bytes for local variables.
COM1
Select Mode :
Send
Autosave
No line ending 9600 baud

```

รูปที่ 3.15 ส่วนแสดงผลของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ทดสอบการควบคุม

4.1.1 ทดสอบการควบคุมแบบเดี่ยว เป็นการทดสอบโดยการใช้ความถี่ที่ต่างกันในการควบคุม โดรนแต่ละตัว

4.1.1.1 ทดสอบความนิ่งของเครื่อง

จุดประสงค์

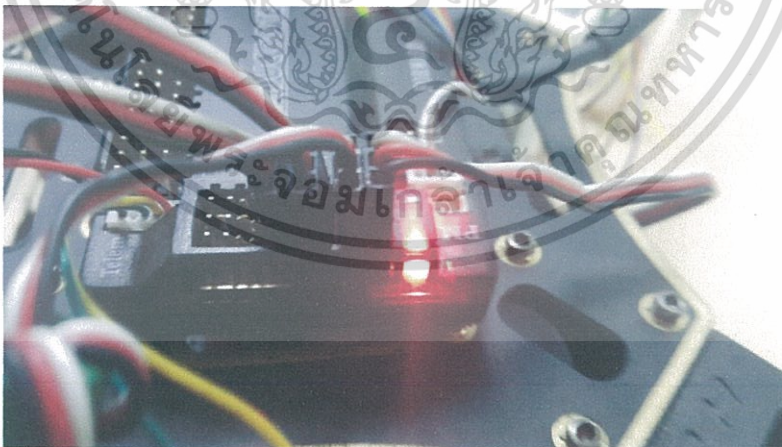
1. ทำให้ยานบินนิ่ง และตรวจหาปัญหาในการควบคุม
2. เพื่อสังเกตไฟสัญญาณเตือนจากอุปกรณ์ควบคุม

วิธีการทดสอบ

1. นำยานบินขึ้นอยู่เหนือพื้นดินประมาณ 2 เมตร แล้วปล่อยค้างให้อยู่กับที่
2. ทดสอบเครื่องควบคุมที่ละแนวแกน
3. สังเกตการเปลี่ยนแปลงและบันทึกผล

ผลการทดลอง

1. ไฟสัญญาณจะเริ่มเปลี่ยนจากสีแดงกระพริบเป็นสีแดงค้างเมื่อมีการนำเครื่องขึ้นเพื่อแสดงว่าเครื่องเชื่อมต่อกับรีโมทเรียบร้อยแล้ว



รูปที่ 4.1 ไฟสัญญาณเตือนจากอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สามารถลอยนิ่งอยู่กับที่ได้



รูปที่ 4.2 ควบคุมยานบินให้ลอยอยู่กับที่ และเคลื่อนที่ในทิศทางต่าง ๆ

3. ควบคุมการเคลื่อนที่แกน x และแกน y ได้ปกติ

4. ควบคุมการขึ้นลงในแกน z ได้ปกติ

4.1.1.2 ทดสอบระยะเวลาการควบคุม

จุดประสงค์

1. ทดสอบหาระยะทางที่สามารถควบคุมโดยใช้รีโมท

วิธีการทดสอบ

1. ทดสอบควบคุมยานบินจากฝั่งหนึ่งของสนามกีฬาไปอีกฝั่งที่มีระยะทางทั้งหมด 200 เมตร และบินกลับมายังจุดเริ่มต้น

ผลการทดลอง

1. ยานบินสามารถเคลื่อนไปยังอีกฝั่งของสนามและกลับมายังจุดเริ่มต้นได้



รูปที่ 4.3 ให้อยานบินเคลื่อนที่ไปยังเส้นขอบที่กำหนดไว้ที่ระยะ 200 เมตร



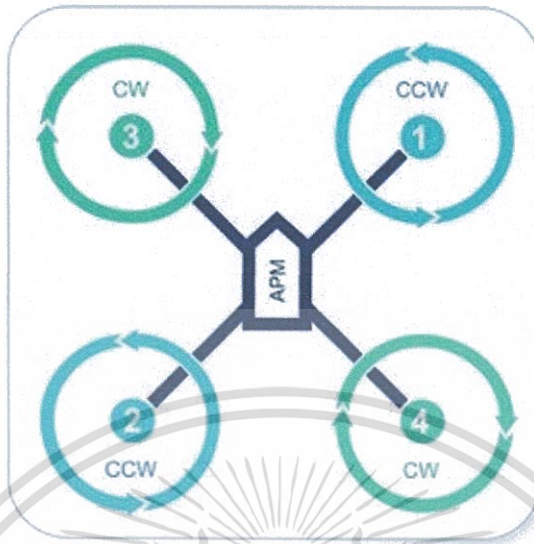
รูปที่ 4.4 อยานบินบินกลับมายังจุดเริ่มต้น

2. การควบคุมเป็นไปได้อย่างยากลำบากเนื่องจากแสงแดดจ้าทำให้มองเห็นหน้าจอได้ไม่ชัด

4.2 คำสัญญาณ Output จาก APM เข้าสู่ Motor

เมื่อทำการบังคับรีโมทคอนโทรลในส่วนของารรับสัญญาณ Input เข้าสู่บอร์ด APM และสัญญาณ Output จากบอร์ด APM ไปยัง ESC สามารถวัดสัญญาณที่ได้ออกมาในรูปแบบของสัญญาณ PWM ซึ่งสามารถวัดค่าความกว้าง Pulse โดยที่ APM Output ที่ออกจากบอร์ด APM จะถูกไปยัง ESC โดยอ้างอิงมอเตอร์แต่ละตัว ดังรูปที่ 4.5

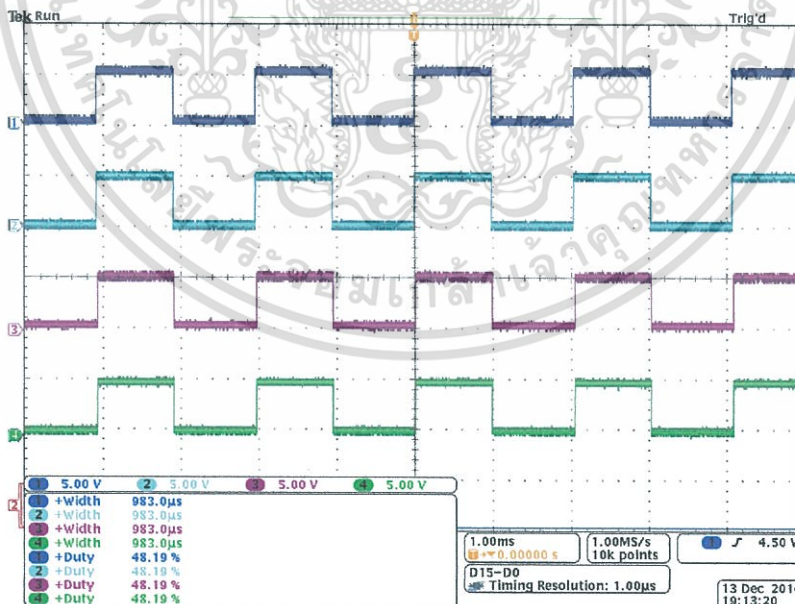
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 มอเตอร์แต่ละตัวที่ใช้อ้างอิง

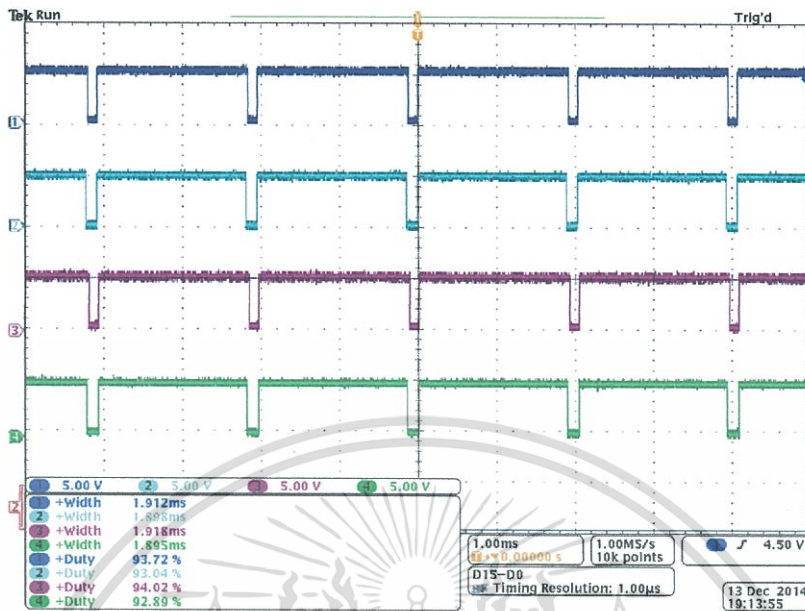
ซึ่งจะได้สัญญาณดังต่อไปนี้

หมายเหตุ : หลักการทำงานของ Roll Pitch Yaw จะทำงานได้ก็ต่อเมื่อ Throttle ทำงานเท่านั้น

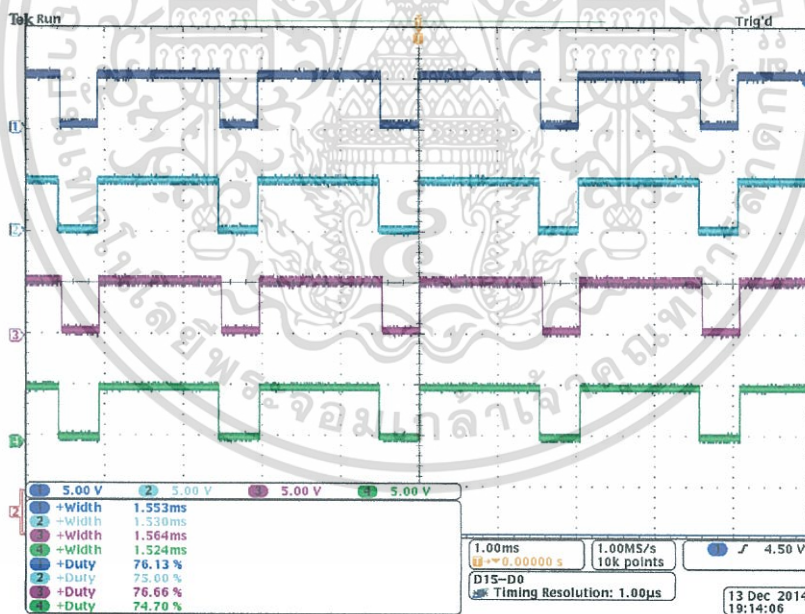


รูปที่ 4.6 ปรับ Throttle 0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



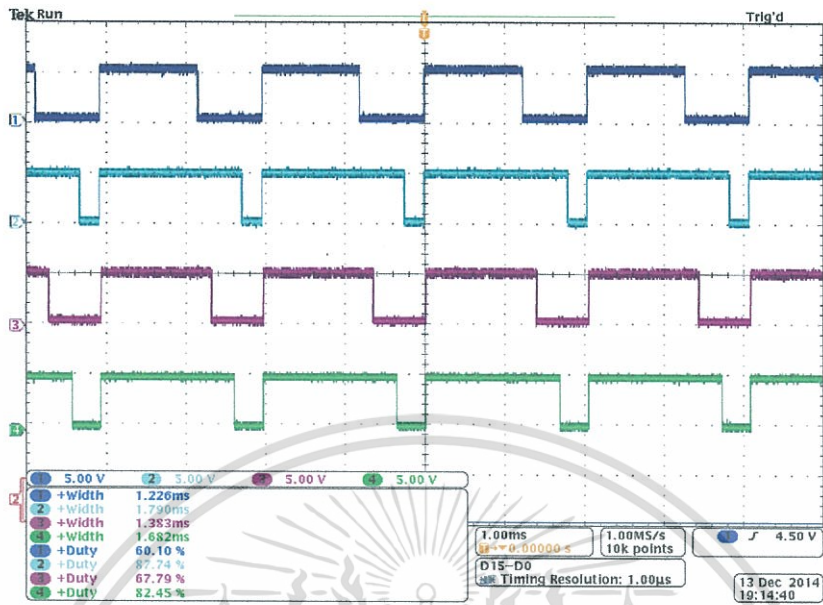
รูปที่ 4.7 ปรับ Throttle 100%



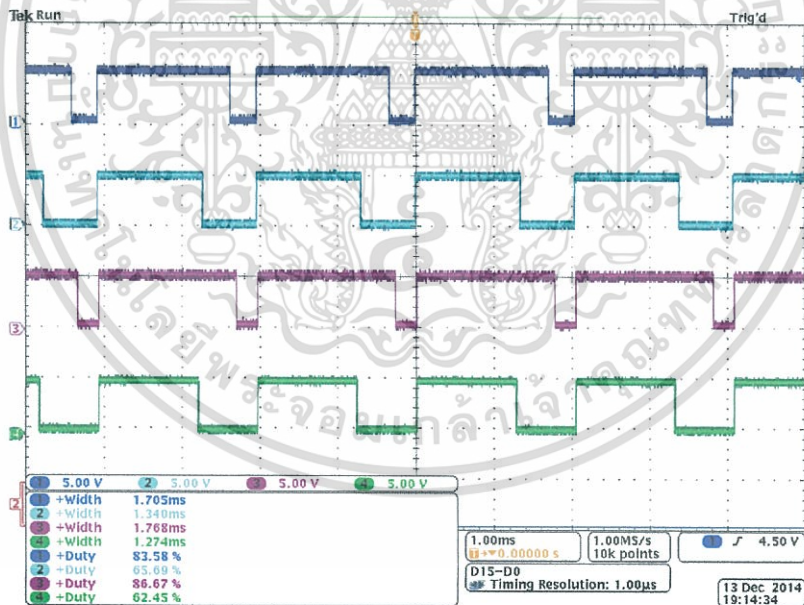
รูปที่ 4.8 ปรับ Throttle 50%

จากรูปที่ 4.6, 4.7 และ 4.8 จะเห็นได้ว่าค่า Pulse ที่ส่งไปยัง Motor แต่ละตัวจะมีค่าใกล้เคียงกัน ตามหลักการของ Throttle เพื่อใช้ยกตัว หรือลดระดับความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



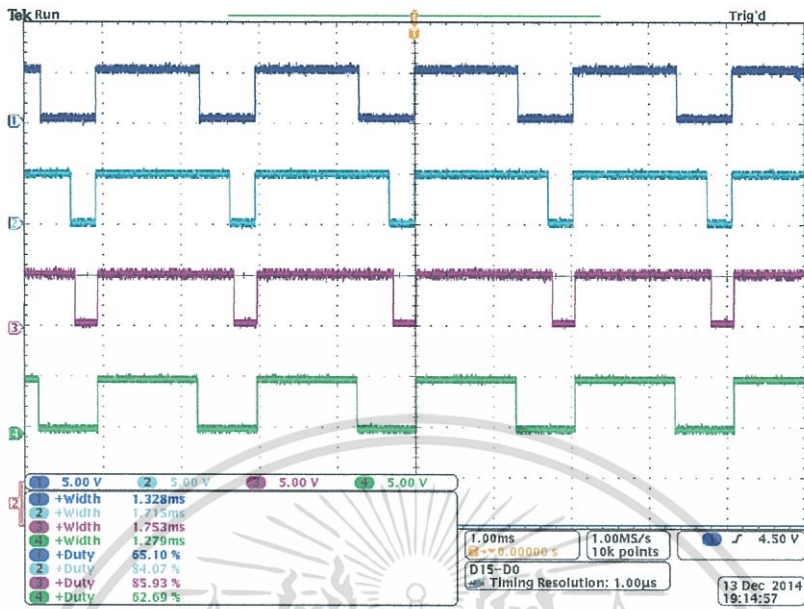
รูปที่ 4.9 ปรับ Throttle 50% และปรับ Pitch สูงสุด



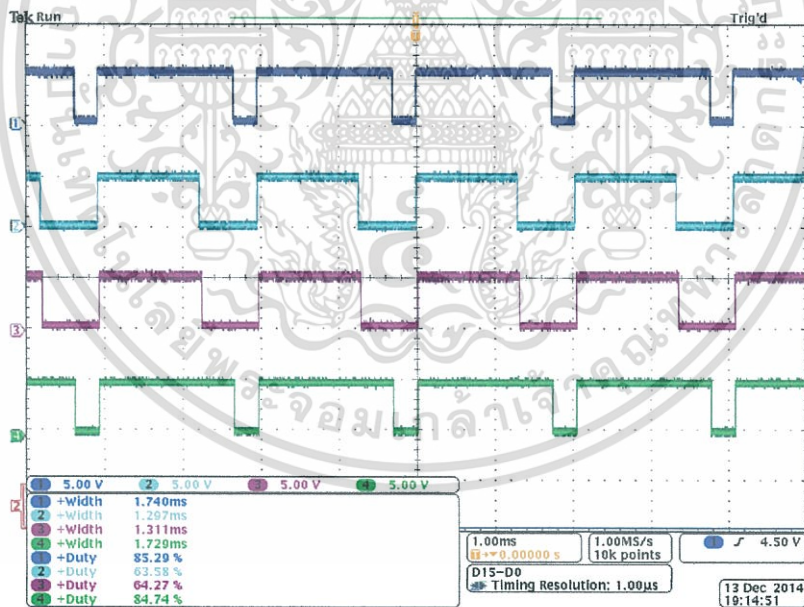
รูปที่ 4.10 ปรับ Throttle 50% และปรับ Pitch ต่ำสุด

จากรูปที่ 4.9 และ 4.10 เมื่อปรับ Pitch สูงสุด (เคลื่อนที่ไปด้านหน้า) มอเตอร์ 1, 3 จะทำงานน้อยกว่า 2, 4 จึงทำให้ตัวลำเคลื่อนที่ไปด้านหน้า และเมื่อปรับ Pitch ต่ำสุด (เคลื่อนที่ไปด้านหลัง) มอเตอร์ 1, 3 จะทำงานมากกว่า 2, 4 จึงทำให้ตัวลำเคลื่อนที่ไปด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



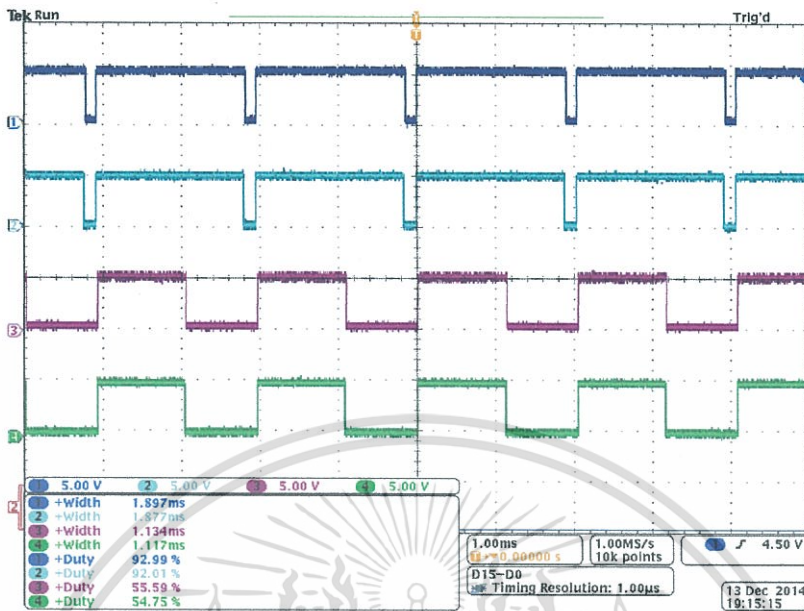
รูปที่ 4.11 ปรับ Throttle 50% และปรับ Roll สูงสุด



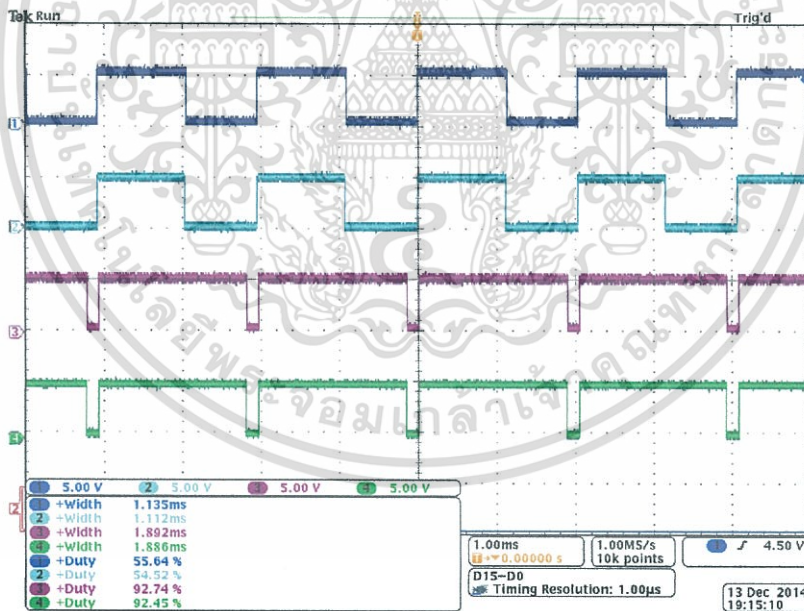
รูปที่ 4.12 ปรับ Throttle 50% และปรับ Roll ต่ำสุด

จากรูปที่ 4.11 และ 4.12 เมื่อปรับ Roll สูงสุด (เคลื่อนที่ไปด้านขวา) มอเตอร์ 1, 4 จะทำงานน้อยกว่า 2, 3 จึงทำให้ตัวลำเคลื่อนที่ไปด้านขวา และเมื่อปรับ Roll ต่ำสุด (เคลื่อนที่ไปด้านซ้าย) มอเตอร์ 1, 4 จะทำงานมากกว่า 2, 3 จึงทำให้ตัวลำเคลื่อนที่ไปด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ปรับ Throttle 50% และปรับ Yaw สูงสุด



รูปที่ 4.14 ปรับ Throttle 50% และปรับ Yaw ต่ำสุด

จากรูปที่ 4.13 และ 4.14 เมื่อปรับ Yaw สูงสุด (หมุนด้านขวา) มอเตอร์ 3, 4 จะทำงานน้อยกว่า 1, 2 จึงทำให้ตัวลำหมุนไปด้านขวา และเมื่อปรับ Yaw ต่ำสุด (หมุนด้านซ้าย) มอเตอร์ 3, 4 จะทำงานมากกว่า 1, 2 จึงทำให้ตัวลำหมุนไปด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบโปรแกรมควบคุมรูปแบบการทำงาน



รูปที่ 4.15ก แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของโดรนตัวที่ 1



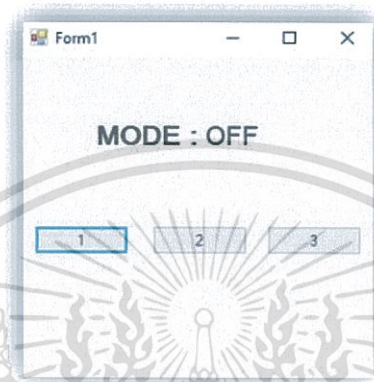
รูปที่ 4.15ข แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของโดรนตัวที่ 2

จากรูปที่ 4.15ก แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของโดรนตัวที่ 1 และรูปที่ 4.15ข แสดงผลการทำงานของโปรแกรมควบคุมรูปแบบการทำงานของโดรนตัวที่ 2 โดยที่เมื่อป้อนค่ารูปแบบที่ต้องการเป็นรูปแบบที่ 1 จะเห็นว่าโดรนตัวที่ 1 จะสามารถทำงานและส่งข้อมูลไปเพื่อควบคุมการเคลื่อนไหวได้โดยขึ้นสถานะเป็น “ON” ในขณะที่โดรนตัวที่ 2 จะไม่ทำงานโดยขึ้นสถานะเป็น “OFF” แต่หากมีการป้อนค่ารูปแบบอื่นๆที่ต้องการใช้งานลงไป จะแสดงผลเป็นแบบต่างๆกัน ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

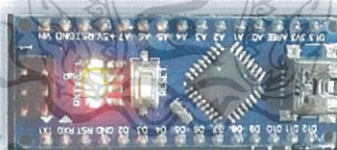
4.4 ผลการทดสอบการควบคุมรูปแบบการทำงานผ่านโปรแกรม Microsoft Visual Studio

จากการทดสอบตัวโปรแกรมที่ใช้ในการควบคุมรูปแบบการทำงานแบบต่าง ๆ โดยใช้ Arduino nano เป็นตัวควบคุมนั้น ในส่วนต่อมาเพื่อให้ง่ายต่อการใช้งาน เราจะทำการสร้างหน้าจอที่ใช้ในการควบคุมโดยใช้ Microsoft Visual Studio ออกแบบและแสดงผล ได้เป็นดังรูปที่ 4.18

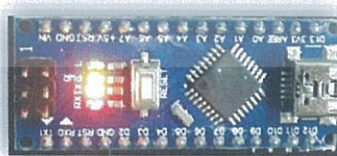


รูปที่ 4.18 หน้าจอการควบคุมโหมด “OFF”

จะเห็นว่าในตอนแรกที่ยังไม่ได้ทำการเลือกโหมดที่จะใช้ในการควบคุมการทำงานหน้าจอที่แสดงผลจะแสดงเป็น “OFF” ซึ่งทำให้ทราบว่ายังไม่ได้มีการเลือกรูปแบบการควบคุมว่าจะควบคุมแบบใดโดยตัว Arduino nano นั้นจะเป็นตัวแสดงผลผ่านทางไฟ LED ว่าควบคุมโดรนตัวใดอยู่



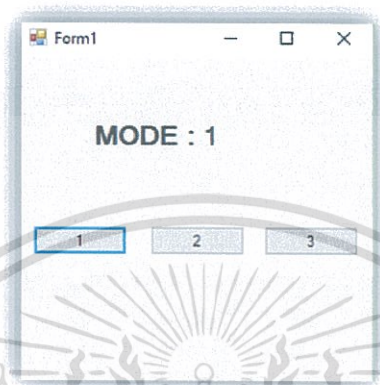
รูปที่ 4.19ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1



รูปที่ 4.19ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2

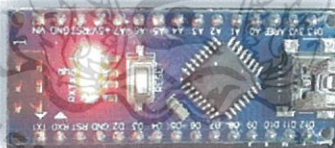
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.19ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1 และรูปที่ 4.19ข แสดงผลการทำงานของ Arduino nano ของโดรนตัวที่ 2 ซึ่งในกรณีนี้จะเห็นว่าไฟ LED ของทั้งสองตัว นั้นดับอยู่แสดงว่ายังไม่ได้มีการเลือกที่จะควบคุมโดรนตัวใด

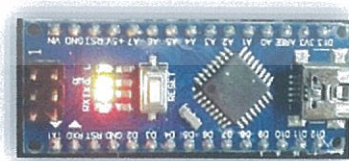


รูปที่ 4.20 หน้าจอการควบคุมโหมด “1”

ต่อมาจะทำการเลือกโหมดที่จะใช้ในการควบคุมการทำงานผ่านทางหน้าจอที่โดยจะเลือกเป็นรูปแบบที่หนึ่งแล้วผลที่ได้จะแสดงเป็น “1” ดังรูปที่ 4.20 ซึ่งทำให้ทราบว่าได้มีการเลือกรูปแบบการควบคุมแล้วว่าจะควบคุมโดรนตัวที่ 1 เพียงตัวเดียวโดยตัว Arduino nano นั้นจะเป็นตัวแสดงผลผ่านทางไฟ LED



รูปที่ 4.21ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1

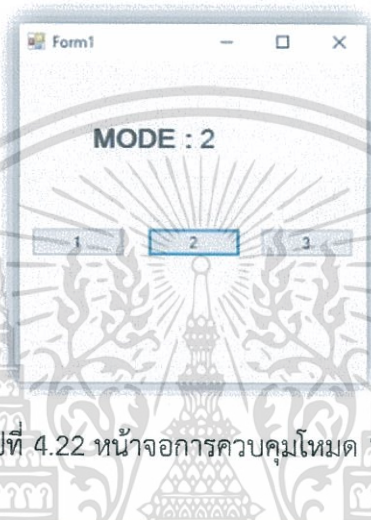


รูปที่ 4.21ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2

โดยจากรูปที่ 4.21ก ที่ใช้แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1 ไฟ LED จะสว่างขึ้นซึ่งจะทำให้ทราบว่าขณะนี้กำลังควบคุมโดรนตัวที่ 1 อยู่แต่ในรูปที่ 4.21ข ที่แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

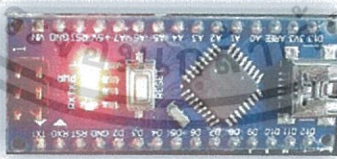
การทำงานของ Arduino nano ของโดรนตัวที่ 2 จะยังคงดับอยู่ ต่อจากนั้นจะทำการเลือกโหมดที่จะใช้ในการควบคุมการทำงานเป็นรูปแบบที่สองซึ่งหน้าจอที่แสดงผลจะแสดงเป็น “2” ดังรูปที่ 4.22 ซึ่งทำให้ทราบว่าได้มีการเลือกรูปแบบการควบคุมว่าจะควบคุมโดรนตัวที่สองเพียงตัวเดียว และตัวที่หนึ่งจะยังคงสามารถลอยอยู่กับที่ได้อยู่แต่ไม่สามารถควบคุมได้แล้วโดยตัว Arduino nano นั้นจะเป็นตัวแสดงผลผ่านทางไฟ LED ว่าควบคุมโดรนตัวใดอยู่ ในกรณีนี้จะแสดงได้ดังรูป



รูปที่ 4.22 หน้าจอการควบคุมโหมด “2”



รูปที่ 4.23ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1

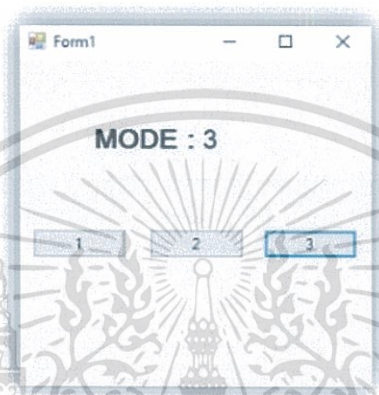


รูปที่ 4.23ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2

โดยรูปที่ 4.23ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1 นั้นไฟ LED จะดับซึ่งจะทำให้ทราบว่าขณะนี้ไม่ได้ควบคุมโดรนตัวที่ 1 แล้วแต่รูปที่ 4.23ข ที่แสดงผลการทำงานของ Arduino nano ของโดรนตัวที่ 2 ไฟ LED จะสว่างขึ้นจะทำให้ทราบว่าขณะนี้กำลังควบคุมโดรนตัวที่ 2 แล้วโดยสามารถควบคุมทิศทางและการเคลื่อนไหวได้ตามที่ต้องการในส่วนต่อไปจะเป็นการเลือกโหมดที่จะใช้ในการควบคุมการทำงานเป็นรูปแบบที่สามซึ่งหน้าจอที่แสดงผลจะแสดงเป็น “3”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

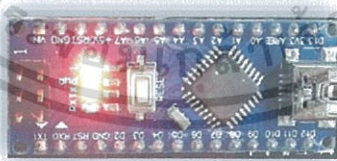
ดังรูปที่ 4.24 ซึ่งทำให้ทราบว่าได้มีการเลือกรูปแบบการควบคุมว่าจะควบคุมโดรนทั้งสองพร้อมกัน โดยตัว Arduino nano นั้นจะเป็นตัวแสดงผลผ่านทางไฟ LED ว่าควบคุมโดรนตัวใดอยู่ ในกรณีนี้จะแสดงได้ดังรูปที่ 4.25ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1 และรูปที่ 4.25ข แสดงผลการทำงานของ Arduino nano ของโดรนตัวที่ 2 ไฟ LED จะสว่างขึ้นทั้งคู่จะทำให้ทราบว่าขณะนี้กำลังควบคุมโดรนอยู่ทั้งสองตัวโดยจะสามารถควบคุมทิศทางการเคลื่อนไหวได้ตามที่ต้องการ



รูปที่ 4.24 หน้าจอการควบคุมโหมด "3"



รูปที่ 4.25ก แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 1



รูปที่ 4.25ข แสดงผลการทำงาน Arduino nano ของโดรนตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากโครงการจะทำการควบคุมยานบินไร้คนขับผ่านทางระบบควบคุมซึ่งเราใช้โปรแกรม Microsoft Visual Studio ในการออกแบบ ทำให้สามารถควบคุมยานบินไร้คนขับได้พร้อม ๆ กันโดยการทำกรเข้ารหัสไว้ แล้วส่งไปยังยานบินไร้คนขับทุกลำ ซึ่งแต่ละลำนั้นเราได้ใส่รหัสไว้แตกต่างกันไว้สำหรับแบ่งว่าจะใช้งานยานบินไร้คนขับลำไหน โดยเราใช้ Arduino nano ในการเข้ารหัสด้วยภาษา C++ ไว้สำหรับแยกยานบินไร้คนขับแต่ละลำออกจากกัน เพราะภาษา C++ มีความไม่ซับซ้อน ทำให้โปรแกรม ขณะที่กำลังทำงานใช้เวลาน้อยและลดการคลาดเคลื่อนของการรับ-ส่ง สัญญาณระหว่างยานบินไร้คนขับ กับ ระบบควบคุม ทำให้มีความเสถียรมากขึ้น

การควบคุมยานบินไร้คนขับหลายลำนั้นจะมีความคลาดเคลื่อนเนื่องจากด้านตัวยานบินไร้คนขับนั้น แต่ละลำมีแรงไม่เท่ากันและมีน้ำหนักไม่เท่ากัน จึงทำให้เวลาส่งสัญญาณไปยังยานบินไร้คนขับพร้อมกัน ในระบบนั้นสามารถทำงานพร้อมกันได้ แต่สำหรับในด้านของยานบินไร้คนขับนั้น ทำงานคลาดเคลื่อนกันเนื่องจากแรงขับของยานบินไร้คนขับแต่ละลำนั้นไม่เท่ากัน ทำให้ต้องใส่การหน่วงของโปรแกรมเพิ่ม ซึ่งทำให้ระบบทำงานช้าลง เพื่อให้สามารถทำงานไปพร้อม ๆ กันได้

5.2 ข้อเสนอแนะ

การควบคุมยานบินไร้คนขับพร้อม ๆ กันนั้น จะมีการหน่วงเวลาจากการที่มีแรงขับเคลื่อนไม่เท่ากัน ควรออกแบบให้สามารถปรับ การหน่วงของโปรแกรมได้ตาม สมรรถนะของยานบินไร้คนขับให้ใกล้เคียงกันได้ และออกแบบให้โปรแกรมควบคุมใช้งานง่ายขึ้น

บรรณานุกรม

- [1] “อากาศยานไร้คนขับ.”
[http://fieldtrip.ipst.ac.th/intro_sub_content.php?content_id=34&content_folder_id=469.](http://fieldtrip.ipst.ac.th/intro_sub_content.php?content_id=34&content_folder_id=469)
- [2] “FS-TH9X.”
[http://www.flysky-cn.com/products_detail/&productId=42.html.](http://www.flysky-cn.com/products_detail/&productId=42.html)
- [3] “FS-TH9X Manual.”
[http://www.hobbypartz.com/79p-th9x-r9b-9channel-radio.html.](http://www.hobbypartz.com/79p-th9x-r9b-9channel-radio.html)
- [4] “Gaussian frequency-shift keying.”
[https://en.wikipedia.org/wiki/Gaussian_frequency-shift_keying.](https://en.wikipedia.org/wiki/Gaussian_frequency-shift_keying)
- [5] “Gaussian Frequency Shift Keying.”
[http://flylib.com/books/en/2.519.1.71/1/.](http://flylib.com/books/en/2.519.1.71/1/)
- [6] Tomislav ŠVEDEK, Marijan HERCEG, Tomislav MATIĆ. “A Simple Signal Shaper for GMSK/GFSK and MSK Modulator Based on Sigma-Delta Look-up Table.”
[http://www.radioeng.cz/fulltexts/2009/09_02_230_237.pdf.](http://www.radioeng.cz/fulltexts/2009/09_02_230_237.pdf)
- [7] “Arduino Nano 3.0.”
[http://www.arduinoall.com/product/18/arduino-nano-3-0-%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%AA%E0%B8%A2-mini-usb.](http://www.arduinoall.com/product/18/arduino-nano-3-0-%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%AA%E0%B8%A2-mini-usb)
- [8] “Arduino Nano.”
[https://www.arduino.cc/en/Main/ArduinoBoardNano.](https://www.arduino.cc/en/Main/ArduinoBoardNano)
- [9] “สภาพอากาศที่เป็นอันตรายต่อการบิน.”
[http://www.rtafclimate4.esy.es/wx-hazard.php.](http://www.rtafclimate4.esy.es/wx-hazard.php)
- [10] “แรงยกของปีกเครื่องบิน.”
[http://www.rmutphysics.com/CHARUD/virtualexperiment/virtual2/lift-force/index.html.](http://www.rmutphysics.com/CHARUD/virtualexperiment/virtual2/lift-force/index.html)
- [11] “การเขียนโปรแกรม Quadrotor พื้นฐาน.”
[http://quad3d-tin.lnwshop.com/article/.](http://quad3d-tin.lnwshop.com/article/)
- [12] “การเขียนโปรแกรมเบื้องต้น Arduino กับ visual basic.”
[http://www.myarduino.net/article/.](http://www.myarduino.net/article/)

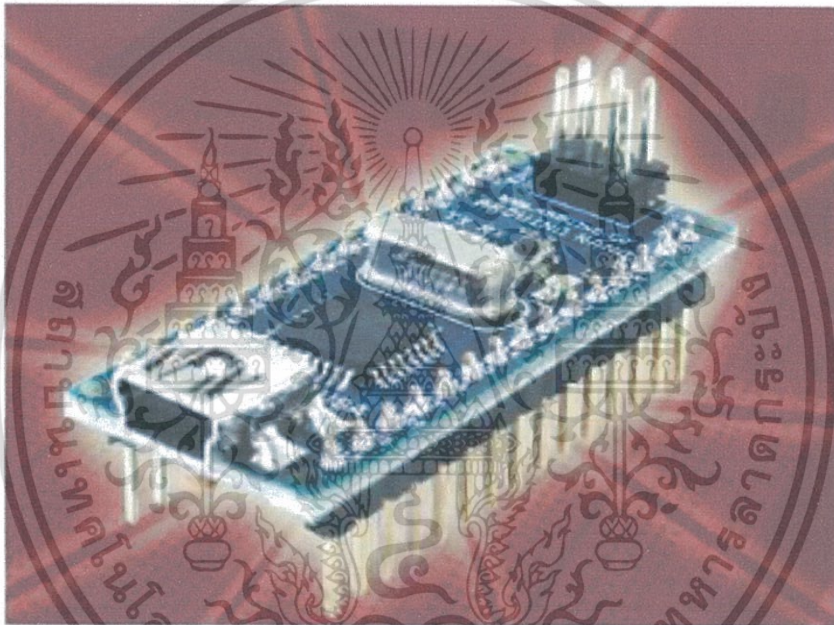
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Nano (V2.3)

User Manual



Released under the Creative Commons Attribution Share-Alike 2.5 License

<http://creativecommons.org/licenses/by-sa/2.5/>

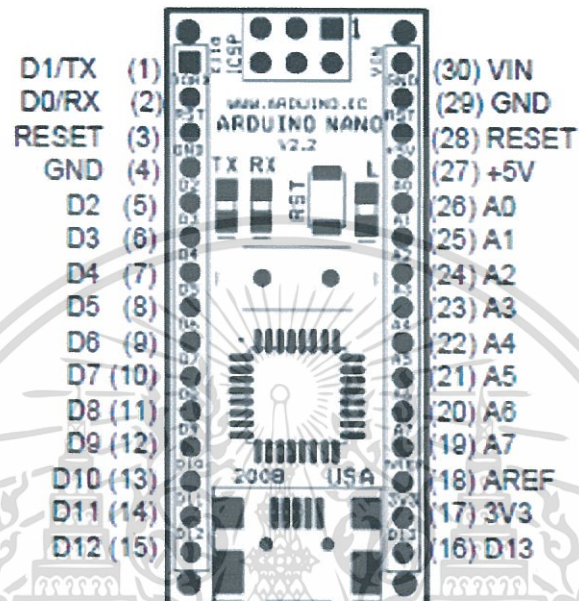
More information:

www.arduino.cc

Rev. 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

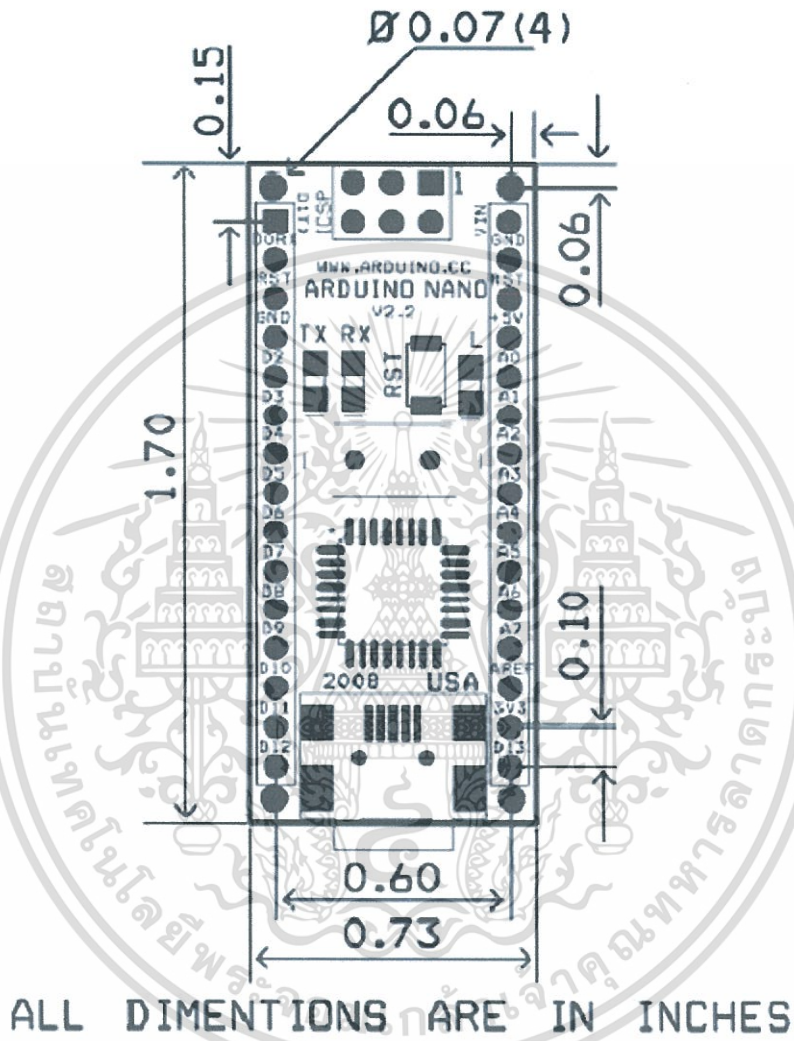
Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital Input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog Input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (Input from external power supply)
30	VIN	PWR	Supply voltage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Nano Mechanical Drawing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Nano Bill of Material

Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805	C0805C180J5GACTU	Kemet	80-C0805C180J5G	Mouser
4	1	D1	Diode, Schottky 0.5A 20V	MBR0520LT1G	ONsemi	863-MBR0520LT1G	Mouser
5	1	J1,J2	Headers, 36PS 1 Row	68000-136HLF	FCI	649-68000-136HLF	Mouser
6	1	J4	Connector, Mini-B Recept Rt. Angle	67503-1020	Molex	538-67503-1020	Mouser
7	1	J5	Headers, 72PS 2 Rows	67996-272HLF	FCI	649-67996-272HLF	Mouser
8	1	LD1	LED, Super Bright RED 100mcd 640nm 120degree 0805	APT20125RCPRV	Kingbright	604-APT20125RCPRV	Mouser
9	1	LD2	LED, Super Bright GREEN 50mcd 570nm 110degree 0805	APHCM2012CGCK-F01	Kingbright	604-APHCM2012CGCK	Mouser
10	1	LD3	LED, Super Bright ORANGE 160mcd 601nm 110degree 0805	APHCM2012SECK-F01	Kingbright	04-APHCM2012SECK	Mouser
11	1	LD4	LED, Super Bright BLUE 80mcd 470nm 110degree 0805	LTST-C170TBKT	Lite-On Inc	160-1579-1-ND	Digikey
12	1	R1	Resistor Pack, 1K +/-5% 62.5mW 4RES SMD	YC164-JR-071KL	Yageo	YC164J-1.0KCT-ND	Digikey
13	1	R2	Resistor Pack, 680 +/-5% 62.5mW 4RES SMD	YC164-JR-07680RL	Yageo	YC164J-680CT-ND	Digikey
14	1	SW1	Switch, Momentary Tact SPST 150gf 3.0x2.5mm	B3U-1000P	Omron	SW1020CT-ND	Digikey
15	1	U1	IC, Microcontroller RISC 16kB Flash, 0.5kB EEPROM, 23 I/O Pins	ATmega168-20AU	Atmel	556-ATMEGA168-20AU	Mouser
16	1	U2	IC, USB to SERIAL UART 28 Pins SSOP	FT232RL	FTDI	895-FT232RL	Mouser
17	1	U3	IC, Voltage regulator 5V, 500mA SOT-223	UA78M05CDCYRG3	TI	595-UA78M05CDCYRG3	Mouser
18	1	Y1	Crystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abracon	815-ABL-16-B2	Mouser

