

ระบบแสดงผลภาคพื้นดินแบบปกพาสำหรับอากาศยานไร้คนขับแบบ 4 ใบพัด

นายธนัช โต๊ะทอง

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ  
ปีการศึกษา 2557

**Portable Ground Station for Quadrotor**

**Mr. Thanat Tothong**

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF BACHELOR OF COMPUTER ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK  
ACADEMIC YEAR 2014

ปริญญาบัตรเรื่อง : ระบบแสดงผลภาคพื้นดินแบบพกพาสำหรับอาคารayan ไร้คนขับแบบ 4  
 ใบพัด  
 ชื่อ : นายธนัช ໂຕະກອງ  
 สาขาวิชา : วิศวกรรมคอมพิวเตอร์  
 ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
 คณะ : วิศวกรรมศาสตร์  
 อาจารย์ที่ปรึกษา : ดร.ревัต ศิริโภคภิรมย์  
 ปีการศึกษา : 2557

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าฯ พระนครเหนือ อนุมัติให้  
 ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
 สาขาวิชาวิศวกรรมคอมพิวเตอร์

  
 (ผู้ช่วยศาสตราจารย์ ดร.นภาดล วิวัฒน์)

หัวหน้าภาควิชาวิศวกรรมไฟฟ้า  
 และคอมพิวเตอร์

  
 (ดร.ревัต ศิริโภคภิรมย์)

ประธานกรรมการ

  
 (ดร.คุณชา ประเสริฐสุวรรณ)

กรรมการ

  
 (ผู้ช่วยศาสตราจารย์คุณภาพ สวัสดิพิศาล)

กรรมการ

ลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าฯ พระนครเหนือ

Project Report Title : Portable Ground Station for Quadrotor  
Name : Mr. Thanat Tothong  
Major Field : Computer Engineering  
Department : Electrical and Computer Engineering  
Faculty : Engineering  
Project Advisor(s) : Dr. Rawat Siripokarpirom  
Academic Year : 2014

Accepted by the Faculty of Engineering, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computer Engineering

.....  
*M.* ..... Chairperson of Department of Electrical  
(Asst. Prof. Noppadol Wiwatcharagoses) and Computer Engineering

.....  
*Rawat Sir* ..... Chairperson  
(Dr. Rawat Siripokarpirom)

.....  
*Danucha Prasertsom* ..... Member  
(Dr. Danucha Prasertsom)

.....  
*S. Khat* ..... Member  
(Asst. Prof. Khatathap Swatdipisal)

Copyright of the Department of Electrical and Computer Engineering, Faculty of Engineering  
King Mongkut's University of Technology North Bangkok

## บทคัดย่อ

เนื่องจากในปัจจุบันนั้นการใช้งานมัลติโอล็อตเป็นไปอย่างแพร่หลาย อาทิเช่น การใช้งานถ่ายภาพมุมสูง งานวิจัย หรือจะเป็นกิจกรรมเพื่อฝึกคลาย แต่เนื่องจากการใช้งานมัลติโอล็อตนั้น เดิมไปด้วยความเสี่ยง เนื่องจากมีความไม่แน่นอนของตัวอุปกรณ์ที่หลากหลาย ซึ่งหากมีข้อผิดพลาดย่อมก่อให้เกิดความเสียหายและอันตรายตามมา ซึ่งพระเกิดจากอุปกรณ์ที่ใช้ส่วนใหญ่นั้นไม่มีข้อมูลของสถานะที่คอยส่งกลับมาบอกผู้ใช้งาน หรือถ้ามีจะเป็นอุปกรณ์ขนาดใหญ่ อาทิ เช่นคอมพิวเตอร์แบบพกพา

พระภณนี้ผู้จัดทำโครงการนี้จึงเล็งเห็นว่าควรจัดทำอุปกรณ์ระบบแสดงผลภาคพื้นดิน แบบพกพาสำหรับอากาศยานไร้คนขับแบบ 4 ใบพัดขึ้นมาเพื่อแก้ปัญหาที่กล่าวมาข้างต้น โดยระบบจะแสดงข้อมูลต่าง ๆ เกี่ยวกับมัลติโอล็อตเพื่อให้ผู้ใช้งานสามารถทราบข้อมูลสถานะได้ตลอดเวลา อาทิเช่น จำนวนสัญญาณดาวเทียม แรงดันในแบตเตอรี่ เวลาที่ใช้ไปในการบิน ความสูงของตัวมัลติโอล็อต และยังสามารถแสดงค่าและปรับพารามิตเตอร์ให้แก่ระบบได้อีก และนอกเหนือจากนี้ระบบยังสามารถทำตัวเป็นรีโมทเพื่อควบคุมตัวมัลติโอล็อตเพื่อเป็นรีโมททางเลือกได้อีกด้วย

## **Abstract**

Nowadays, usage of multi-rotor was so widespread. For example, aerial photography, research purpose, and recreation activity. However, an operation of multi-rotor is full of risk because it involved many uncertainty factors, which may lead to danger or damage. The lack of data feedback provided to a pilot is the major cause of damage. Sometimes it requires a non-handy equipment such as a computer.

Therefore, this project aims to provide portable equipment that able to solve this problem. This equipment will display various of data of multi-rotor to a pilot eg. a number of locked satellite, battery voltage, flight duration and ability to change parameter setting in flight controller.

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาโครงการ ดร.เรวัต ศิริโภกาภิรมย์ และ ดร.ดนุชา ประเสริฐสม ที่ค่อยสอนให้คำแนะนำและปรึกษาและแบ่งคิดในการทำงานและใช้ชีวิตจนกระทั่งสามารถบรรลุเป้าหมายของโครงการนี้ได้

รวมถึงคณาจารย์ภาควิชาศิลปกรรม ไฟฟ้าและคอมพิวเตอร์ ที่ได้มอบวิชาและความรู้มาตลอดจนสามารถนำมาใช้ในโครงการนี้เสร็จสมบูรณ์ รวมทั้งสถานบันการศึกษามหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ได้อี่ออุปกรณ์ สถานที่ ให้ได้ทำการเรียนรู้ ตลอดจนปฏิบัติงานจนสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ผู้จัดทำกราบขอบพระคุณมารดาบิดาและครอบครัว รวมไปถึงเพื่อน ๆ ผู้เป็นแรงบันดาลและแรงผลักดันให้แก่ผู้จัดทำโครงการนี้ที่มีความสำเร็จ

ธนัช โต๊ะทอง

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	๑
บทคัดย่อภาษาอังกฤษ	๒
กิตติกรรมประกาศ	๓
สารบัญตาราง	๔
สารบัญภาพ	๕
บทที่ 1. บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของงาน	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 มัลติโรเตอร์ (Multirotor)	4
2.2 อุปกรณ์ควบคุมการบิน	4
2.3 Ground Station	6
2.4 Transmission System	12
2.5 ไมโครคอนโทรลเลอร์สำหรับการออกแบบ Ground Station	14
2.6 ผู้ใช้และผลสำหรับการออกแบบ Ground Station	15
2.7 SD Card	17
2.8 Real time clock	18
2.9 Rotary Encoder	19
บทที่ 3. วิธีการดำเนินงาน	21
3.1 การวิเคราะห์เพื่อการออกแบบระบบ	21
3.2 การออกแบบระบบด้านฮาร์ดแวร์	22
3.3 การออกแบบระบบด้านซอฟต์แวร์	28
3.4 โครงสร้างซอฟต์แวร์	29
3.5 การออกแบบโปรแกรม	39

## สารบัญ (ต่อ)

	หน้า
3.6 การออกแบบการติดตั้งอุปกรณ์	42
บทที่ 4. ผลการดำเนินงาน	49
4.1 การทดสอบhardtwareที่นำมาใช้งาน	49
4.2 การทดสอบการรับส่งข้อมูลระหว่างส่วนแสดงผลและเมมฟลิงค์เกตเวย์	74
4.3 การทดสอบระบบจริง	88
4.4 ผลการทดสอบประสิทธิภาพของการแบ่งเวลาในการดำเนินงานของแต่ละ Task 100	
4.5 การทดสอบอัตราการสื้นเปลี่ยนพลังงาน	103
บทที่ 5. สรุปผล ข้อเสนอแนะ และปัญหา	105
5.1 สรุปผล	105
5.2 ปัญหาที่พบในการดำเนินงาน	105
5.3 ข้อเสนอแนะ	105
เอกสารอ้างอิง	107
ประวัติผู้แต่ง	108

## สารบัญตาราง

ตารางที่	หน้า
2-1 ตารางเปรียบเทียบสเปคของ Microcontroller	15
2-2 ตารางเปรียบเทียบคุณสมบัติของซอฟต์แวร์รุ่นต่าง ๆ	16
2-3 บ่งบอกการแบ่งประเภท SD card ตามความจุ	17
2-4 บ่งบอกการแบ่งประเภท SD card ตามประสิทธิภาพของการเขียนข้อมูล	17
2-5 แสดงถึงพฤติกรรมของสัญญาณที่ทำให้ Rotary Encoder หมุนในทิศทางทวน เข็มนาฬิกา	19
2-6 แสดงถึงพฤติกรรมของสัญญาณที่ทำให้ Rotary Encoder หมุนในทิศทางตาม เข็มนาฬิกา	19
3-1 การออกแบบ Ground Station เปื้องตื้น	22
4-1 แสดงการทดสอบการเขียนข้อมูลไปยัง SD Card ด้วยขนาดข้อมูลต่าง ๆ	56
4-2 แสดงผลการทดสอบการเขียนข้อมูลขนาด 1000 byte โดยการเขียนข้อมูลแต่ละ ครั้งมีขนาดไม่เท่ากัน	56
4-3 แสดงผลการทดสอบการเขียนข้อมูลไปยัง SD Card ด้วยขนาดข้อมูลที่ไม่เท่ากัน	56
4-4 แสดงผลการทดสอบเวลาที่ใช้ในการเปิดและปิดไฟล์จาก SD Card	57
4-5 แสดง Pin mapping ระหว่าง Ground station และ Mavlink gateway	76
4-6 แสดง Pin mapping ระหว่าง Ground Station และ Mavlink Gateway	79
4-7 แสดงถึงผังการเชื่อมต่อระหว่าง Arduino DUE และ Mavlink Gateway	84
4-8 แสดงถึงผังการเชื่อมต่อระหว่าง Arduino DUE และ Module Joy stick	85
4-9 แสดงถึงเวลาที่ใช้ทำงานในแต่ละ Task	101
4-10 แสดงค่าของการวัดกระแสของระบบโดยเพิ่มอุปกรณ์เข้าไปในระบบครั้งละ หนึ่งชิ้น	104
4-11 แสดงถึงอัตราส่วนเปลี่ยนของแต่ละอุปกรณ์ของ Portable Ground Station	104

## สารบัญภาพ

ภาพที่	หน้า
1-1 ภาพรวมของระบบ	2
2-1 Multi-rotor	4
2-2 แสดงถึงอุปกรณ์ควบคุมการบิน MultiWii	5
2-3 แสดงถึงอุปกรณ์ควบคุมการบิน Ardupilot	5
2-4 แสดงถึงอุปกรณ์ควบคุมการบิน NAZA – M ของ DJI	6
2-5 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม MultiWii	7
2-6 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Mission planner	8
2-7 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม QGroundControl	9
2-8 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Naza Assistant Software	10
2-9 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม IPAD Ground Station ผ่าน IPAD	11
2-10 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Droid Planner ผ่าน Android device	11
2-11 Ardu Station Commercial hardware Station ที่จัดทำโดย jDrones	12
2-12 Futaba FHSS (Futaba Frequency Hopping Spread Spectrum)	13
2-13 Bluetooth	13
2-14 Xbee pro	14
2-15 หน้าจอ TFT ขนาด 3.2 นิ้ว	16
2-16 SD Card	18
2-17 Real time clock	18
2-18 แสดงถึงรูปถ่ายภาพของการหมุนซ้ายขวาของ Rotary encoder	20
2-19 แสดงถึง Rotary Encoder	20
3-1 การออกแบบ Ground Station เป็นต้น	22
3-2 Arduino DUE	24
3-3 Block diagram ระหว่าง Arduino DUE และ GLCD โมดูล LB04301	24
3-4 GLCD Module LB04301 Chipset SSD1963	24
3-5 GLCD Module LB04301 Chipset SSD1963 ด้านหลังพร้อม SD card Module	25

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-6 แสดงการเชื่อมต่อของส่วนสื่อสารกับผู้ใช้งานและส่วนความคุณการบินโดยใช้การออกแบบลักษณะที่ 1	25
3-7 แสดงการเชื่อมต่อของส่วนสื่อสารกับผู้ใช้งานและส่วนความคุณการบินโดยใช้การออกแบบลักษณะที่ 2	26
3-8 แสดงการเชื่อมต่อของส่วนเก็บบันทึกข้อมูลการบินเข้ากับ Arduino DUE	27
3-9 Block diagram ของ Ground Station	28
3-10 Flow chart ของ Ground station	31
3-11 Flow chart subroutine setup	32
3-12 Flow chart subroutine Receive_serial	34
3-13 Flow chart subroutine ProcessSerial	35
3-14 Flow chart subroutine Flust_SD	36
3-15 Flow chart ของ Subroutine Draw LCD	37
3-16 Flow chart ของ Subroutine RC_override	38
3-17 การออกแบบใบหน้ารีโมทโคล	39
3-18 ข้อมูลของความสัมพันธ์ระหว่าง CMD 11 กับ PAYLOAD	40
3-19 ข้อมูลของความสัมพันธ์ระหว่าง CMD 12 กับ PAYLOAD	40
3-20 ข้อมูลของความสัมพันธ์ระหว่าง CMD 13 กับ PAYLOAD	40
3-21 ข้อมูลของความสัมพันธ์ระหว่าง CMD 14 กับ PAYLOAD	40
3-22 ข้อมูลของความสัมพันธ์ระหว่าง CMD 15 กับ PAYLOAD	41
3-23 ข้อมูลของความสัมพันธ์ระหว่าง CMD 50 กับ PAYLOAD	41
3-24 แสดงถึงข้อมูลของความสัมพันธ์ระหว่าง CMD 200 กับ PAYLOAD	41
3-25 แสดงถึงข้อมูลของความสัมพันธ์ระหว่าง CMD 201 กับ PAYLOAD	41
3-26 การออกแบบ PCB สำหรับการเชื่อมต่อ Arduino DUE และ GLCD	42
3-27 PCB Pin converter ระหว่าง Arduino DUE และ GLCD ด้านบน	43
3-28 PCB Pin converter ระหว่าง Arduino DUE และ GLCD ด้านล่าง	43
3-29 PCB Pin converter หลังจากบัดกรีสำเร็จ	43
3-30 Module joy stick	44

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-31 PCB Module joy stick	45
3-32 PCB Push button & terminal	46
3-33 PCB Push button & terminal ด้านบน	46
3-34 PCB Push button & terminal ด้านล่าง	47
3-35 PCB Push button & terminal หลังจากบัดกรีสำเร็จ	47
3-36 Main PCB	48
3-37 Main PCB หลังบัดกรีสำเร็จ	48
4-1 แสดงถึงการเชื่อมต่อระหว่าง GLCD Module และ Arduino DUE	50
4-2 แสดงถึงเวลาที่ใช้ในการ Clear Screen	51
4-3 แสดงถึงเวลาที่ใช้ในการ Fill Color ทั้งหน้าจอ	51
4-4 ตัวอย่างการขับค่าสัญญาณของคำสั่ง SetPIXEL	52
4-5 แสดง Schematic การเชื่อมต่อระหว่าง SD card module และ Arduino DUE	53
4-6 แสดงถึงการเชื่อมต่อระหว่าง Arduino DUE และ SD Card module	54
4-7 แสดงถึงเวลาที่ใช้ในการเขียนข้อมูลขนาด 1 byte ไปยัง SD card โดยที่ใช้เวลาในการเขียน 1.19 ms	55
4-8 แสดงถึงเวลาที่ใช้ในการ Flush ข้อมูลขนาด 1 byte ไปยัง SD card โดยที่ใช้เวลาในการ flush 12.7 ms	55
4-9 ผังการเชื่อมต่อวงจรระหว่าง Arduino DUE และ Rotary Encoder	58
4-10 แสดงถึงการเชื่อมต่อระหว่าง Arduino DUE และ Rotary Encoder	58
4-11 แสดงถึงการสร้าง Pulse หนึ่งลูกสัญญาณและ Arduino สามารถจับค่าสัญญาณได้	59
4-12 แสดงถึงจังหวะการนับสัญญาณด้วย Library Encoder.h	60
4-13 แสดงถึงการหมุน Rotary Encoder แบบตามเข็มนาฬิกา	61
4-14 แสดงถึงการหมุน Rotary Encoder แบบวนเข็มนาฬิกา	61
4-15 แสดงผังการเชื่อมต่อระหว่าง Arduino DUE และ Real Time Clock	63
4-16 แสดงการเชื่อมต่อระหว่าง Arduino DUE และ Real Time Clock	63
4-17 แสดงถึงการตั้งค่าให้แก่ Real time clock ด้วย Arduino DUE	64

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4-18 แสดงถึงการตั้งค่า Real time clock ด้วย Arduino DUE สำเร็จ	64
4-19 แสดงถึงการอ่านค่า Real time clock ด้วย Arduino DUE และแสดงค่าทุก ๆ 1 วินาที	64
4-20 ผังแสดงการเชื่อมต่อระหว่าง Arduino DUE และ Push Button	66
4-21 แสดงการเชื่อมต่อระหว่าง Arduino DUE และปุ่มกดแบบกดติดปล่อยดับ	66
4-22 แสดงถึงการกดปุ่มเพื่อลดหรือเพิ่มค่าให้แก่ค่าตัวแปรและแสดงออกทาง Serial	67
4-23 แสดงถึง Time responds ของปุ่มกดเมื่อใช้งานแบบ Interrupt	68
4-24 แสดงการเชื่อมต่อระหว่าง Arduino DUE และ ON/OFF Switch	69
4-25 แสดงถึงความสัมพันธ์ระหว่างตำแหน่งของ ON/OFF switch และ LED บน	70
4-26 แสดงถึงความสัมพันธ์ระหว่างตำแหน่งของ ON/OFF switch และ LED บน	70
4-27 แสดงถึงการเชื่อมต่อ Joy stick เข้ากับ Module Joy stick	71
4-28 แสดงถึงการเชื่อมต่อ Module Joy stick เข้ากับ Arduino DUE	72
4-29 แสดงถึงการเชื่อมต่อ Module Joy stick เข้ากับ Arduino DUE	72
4-30 แสดงถึงการวัดค่าเวลาของการ อ่านค่า Analog ทั้ง 4 Chanel โดยใช้เวลา 18.3 us	73
4-31 แสดงถึงการวัดค่าเวลาของการ อ่านค่า Analog ทั้ง 4 Chanel ทำการ map ค่าของ Analog input โดยที่ใช้เวลา 22 us	73
4-32 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway	74
4-33 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway	75
4-34 แสดงทดสอบการรับ/ส่งข้อมูลพื้นฐานหรือ Data stream	77
4-35 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway	78
4-36 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway	78
4-37 แสดงถึงการสั่งหยุด Streaming Message และวิธีรีบทำการขอ Parameter จาก Mavlink Gateway	79
4-38 แสดงถึงการ Streaming Parameter จนเสร็จแล้วจึงเริ่มทำการขอ Stream data ไปยัง Mavlink Gateway	80

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4-39 แสดงถึงการเริ่ม Stream data หลังจากที่ Ground Station ทำการขอ Stream data หลังจากที่รับ Parameter ได้สำเร็จ	80
4-40 Serial monitor ของ Mavlink Gateway เมื่อทำการส่ง Parameter	81
4-41 ผลการทดสอบการทดสอบการส่งค่า Parameter	82
4-42 แสดงถึงการปรับค่าและส่งค่าคืนไปยัง Ground Station ของ Mavlink Gateway	82
4-43 แสดงถึง Block diagram ของการทดลองการส่งค่า RC override ด้วยความถี่ 20 Hz	83
4-44 แสดงผังวงจรการเชื่อมในการทดสอบการส่งค่า RC override ด้วยความถี่ 20 Hz	84
4-45 แสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground Station	85
4-46 แสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground Station	86
4-47 รูปแสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground	86
4-48 แสดงถึงเวลาที่ใช้ในการ Override RC 1 ครั้งและทำคำสั่ง Override RC ทุก ๆ 5 Hz	87
4-49 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway	89
4-50 การเปิดระบบของ Ground Station สามารถทำงานได้ตามเงื่อนไขข้างต้น ที่กล่าวไว้	89
4-51 Serial monitor การรับค่า Parameter มาจาก Mavlink Gateway	90
4-52 แสดง Text File ที่เกิดจากการเก็บข้อมูลลง SD card ของ Ground Station	90
4-53 การส่งข้อมูลจาก Parameter มาจาก Mavlink Gateway	91
4-54 แสดงถึงหน้าแสดงผลสถานะของลำหน้าที่ 1	91
4-55 แสดงผลของสถานะของมัลติโตร์หน้าที่ 2	92
4-56 แสดงถึงการส่งค่า Parameter จาก Ground Station ไปยัง Mavlink Gateway	92

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4-57 แสดงรับค่า Parameter จาก Mavlink Gateway ไปยัง Ground Station	93
4-58 แสดง Portable Ground Station จากด้านบน	93
4-59 แสดง Portable Ground Station จากด้านใน อะคิลิกแผ่นด้านล่าง	94
4-60 แสดง Portable Ground Station จากด้านใน อะคิลิกแผ่นด้านบน	94
4-61 แสดง Portable Ground Station จากด้านข้าง	95
4-62 แสดงการทดลองทำการบินที่สถานที่จริง	96
4-63 แสดงถึงการบังคับระหว่าง Portable Ground Station และอุปกรณ์ควบคุมแบบ 2.4 GHz	96
4-64 ขณะทำการบินด้วย Portable Ground Station ที่ความสูงประมาณ 5- 10 เมตร	97
4-65 ขณะทำการติดตั้งอุปกรณ์ก่อนทำการทดสอบ	97
4-66 แสดงถึงความสูงของการบินในระยะเวลาต่าง ๆ	98
4-67 แสดงถึงระดับแรงดันของแบตเตอรี่ภายในของมัคติโรเตอร์เทียบกับเวลาในการบิน	98
4-68 แสดงถึงระดับการใช้งานของกระแสที่แบตเตอรี่ทำการจ่ายให้แก่เมอเตอร์ทึ่ส่วนตัวในเวลาต่างๆ	99
4-69 แสดงถึงจำนวนดาวเทียมที่ระบบสามารถตรวจจับได้ในเวลาต่าง ๆ	99
4-70 แสดงถึงเวลาที่ใช้ในการทำ Task Draw screen task 1 โดยใช้เวลาน้อยที่สุด	100
4-71 แสดงถึงเวลาที่ใช้ในการทำ Task Draw screen task 1 โดยใช้เวลามากที่สุด	101
4-72 แสดงถึงเวลาที่แต่ละ Task ใช้ในการดำเนินการ	102
4-73 แสดงถึงเวลาที่แต่ละ Task ใช้ในการดำเนินการ	102
4-74 แสดงถึงการทดลองการวัดกระแสอุปกรณ์ชิ้นต่าง ๆ	103
ของ Portable Ground Station	

## บทที่ 1

### บทนำ

#### 1.1 หลักการและเหตุผล

เนื่องจากในปัจจุบันอากาศยานไร้คนขับแบบสี่ใบพัด ได้เป็นที่นิยมเพิ่มขึ้นอย่างมากจึงส่งผลให้ อุปกรณ์ควบคุมการบินมีหลากหลายชนิด ซึ่งเป็นไปในรูปแบบของ Open Source และ commercial software ซึ่งส่วนใหญ่ในส่วนของการเชื่อมต่อกับ อุปกรณ์ควบคุมการบินต้องทำผ่านคอมพิวเตอร์ซึ่ง ส่งผลต่อความไม่สะดวกในการนำไปใช้งานในพื้นที่ต่าง ๆ รวมทั้งการควบคุมกล้องที่ติดตั้งบนอากาศยาน ไร้คนขับแบบสี่ใบพัดโดยทั่วไปจะต้องใช้อุปกรณ์ควบคุมที่มีจำนวนช่องสัญญาณเป็นจำนวนมาก หรือ ใช้อุปกรณ์ควบคุมตัวที่สองเพื่อควบคุม ซึ่งส่งผลถึงราคาอุปกรณ์ที่จะต้องเพิ่มขึ้นเป็นอย่างมาก

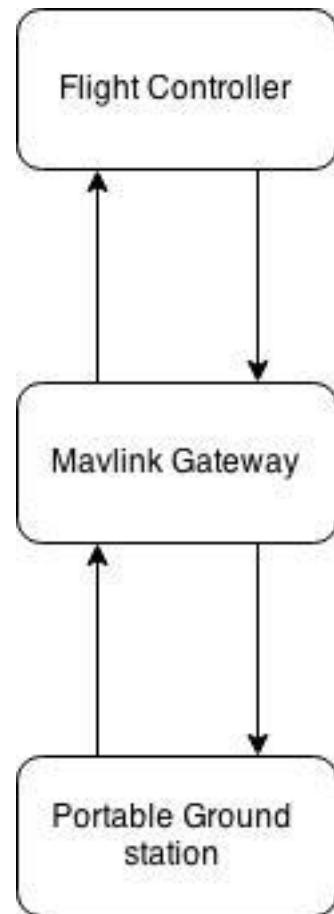
#### 1.2 วัตถุประสงค์ของโครงการ

จัดทำอุปกรณ์เพื่อใช้งานอ่านค่าข้อมูลต่าง ๆ จากอากาศยานไร้คนขับแบบสี่ใบพัดและทำการแสดงผลผ่านจอภาพฟิก แอลซีดีรวมทั้งมี อุปกรณ์ส่วนควบคุม ที่สามารถส่งค่ากลับไปควบคุมอากาศยานไร้คนขับแบบสี่ใบพัดผ่าน Mavlink Gateway ไปยังอุปกรณ์ที่เรียกว่า Portable ground station นั้นเอง

#### 1.3 ขอบเขตของงาน

- 1.3.1 หน้าจอภาพฟิกแอลซีดีจะต้องสามารถแสดงผลและอัพเดทข้อมูลที่ความถี่ไม่ต่ำกว่า 5Hz
- 1.3.2 SD Card จะต้องบันทึกข้อมูลที่ได้รับจาก Mavlink Gateway ทุก ๆ 1 วินาที
- 1.3.3 ในการเขียนข้อมูลลง SD Card นั้นจะมีการบันทึกเวลาปัจจุบันทุกรั้งที่ทำการบันทึก
- 1.3.4 หน้าจอต้องสามารถแสดงค่า Parameter ให้แก่ผู้ใช้งานได้
- 1.3.5 สามารถรับข้อมูลจาก Mavlink Gateway เพื่อนำมาแสดงผลบนหน้าจอ GLCD ได้
- 1.3.6 ในการส่งข้อมูลกลับไปที่ Mavlink Gateway นั้น Portable Ground Station นั้นต้องสามารถ ส่งคำสั่งเพื่อควบคุมการบิน ให้แก่ Mavlink Gateway ด้วยความถี่ไม่ต่ำกว่า 20 Hz
- 1.3.7 ในการส่งข้อมูลกลับไปที่ Mavlink Gateway นั้น Portable Ground Station นั้นต้องสามารถ ปรับค่า Parameter และส่งกลับไปยัง Mavlink ได้

- 1.3.8 สามารถใช้อุปกรณ์ Rotary encoder ในการปรับค่า Parameter ให้แก่ระบบได้
- 1.3.9 Portable Ground Station และ Mavlink Gateway สามารถรับ/ส่งข้อมูลด้วย Custom Protocol ได้อย่างถูกต้อง
- 1.3.10 ผู้ใช้งานสามารถเปลี่ยนหน้าการแสดงผลของจอกราฟฟิกแล็ซีดีได้ด้วยการกดปุ่ม
- 1.3.11 ผู้ใช้งานสามารถเปิดและปิดการใช้งาน Override RC ด้วย สวิตซ์สองทางได้



ภาพที่ 1-1 ภาพรวมของระบบ

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 เรียนรู้การใช้งานและอุปกรณ์เชิงไมโครคอนโทรลเลอร์ อาทิเช่น Arduino, หน้าจอกราฟฟิกแอลซีดี, อุปกรณ์เชื่อมต่อ SD Card และอื่น ๆ
- 1.4.2 ได้เรียนรู้ในการออกแบบ Protocol ในการสื่อสารในเชิง Embedded system
- 1.4.3 เรียนรู้การออกแบบการ Share Task สำหรับระบบที่ไม่ใช่ Real time operation system
- 1.4.4 ทักษะในการออกแบบ PCB
- 1.4.5 ทักษะในการออกแบบกล่องสำหรับการใส่ชิ้นงาน Portable ground station

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 มัลติโรเตอร์ (Multirotor)

คืออากาศยานไร้คนขับ (unmanned aerial vehicle) แบบปีกหมุนที่มีมอเตอร์มากกว่า 4 ตัวขึ้นไป โดยที่จุดเด่นของมัลติโรเตอร์คือความสามารถในการขึ้นลงแนวตั้งและสามารถใช้งานในที่แคบได้โดยที่หลักการทำงานของมัลติโรเตอร์นั้น อาศัยแรงจากการหมุนของมอเตอร์



ภาพที่ 2-1 Multi-rotor

([http://www.bhphotovideo.com/images/images2500x2500/dji\\_inspire\\_1\\_1097099.jpg](http://www.bhphotovideo.com/images/images2500x2500/dji_inspire_1_1097099.jpg))

#### 2.2 อุปกรณ์ควบคุมการบิน

คืออุปกรณ์ที่เปรียบเสมือนสมองของ Multirotor ซึ่งมีหน้าที่รักษาและปรับเสถียรภาพของมัลติโรเตอร์โดยอ่านค่าจาก Sensor ต่าง ๆ เช่น accelerometer, barometer, compass เป็นต้น โดยที่อุปกรณ์ควบคุมการบินจะรับคำสั่งจากผู้ควบคุมและสามารถส่งข้อมูลกลับมาแสดงผลบนภาคพื้นดิน (Ground Station) ได้ โดยอุปกรณ์ควบคุมการบินในปัจจุบันมีหลายหลายชนิด เช่น MultiWii, Arduipilot, DJI NASA, KK Board ซึ่งแตกต่างกันไปในเรื่องของคุณภาพและราคา

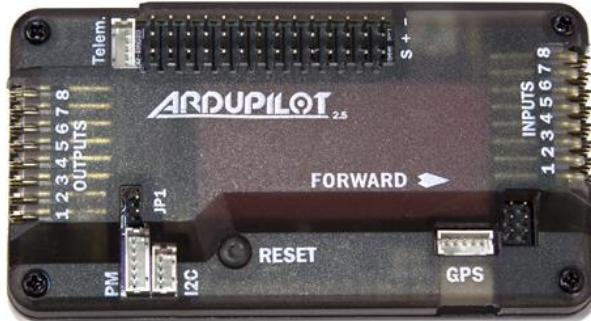
2.2.1 MultiWii เป็น อุปกรณ์ควบคุมการบินแบบ Open Source ที่ราคาถูกและหาซื้อได้ที่สุดในท้องตลาด โดยที่มีความสามารถพื้นฐานที่ครบครัน เช่น โหมดการบินแบบ Acrobatic, Stabilize และรองรับการติดตั้ง GPS เพราะฉะนั้นจึงเหมาะสมแก่การเริ่มต้นในการศึกษามัลติโรเตอร์



ภาพที่ 2-2 แสดงถึงอุปกรณ์ควบคุมการบิน MultiWii

([http://www.pyramidmodels.com/shop/product.php/1116/multiwii\\_328p\\_flight\\_controller\\_w\\_ftdi\\_dsm2\\_port](http://www.pyramidmodels.com/shop/product.php/1116/multiwii_328p_flight_controller_w_ftdi_dsm2_port))

2.2.2 Arduipilot เป็นอุปกรณ์ควบคุมการบินแบบ Open Source ที่มีความซับซ้อนและมีเสถียรภาพสูงกว่า MultiWii พoSมควรโดยที่ ชิริสของ Arduipilot มีทั้งการประมวลผลแบบ 8 บิต (APM) และ 32 บิต (Pixhawk) ทั้งสองตัวนี้มีฟังก์ชันการทำงานเหมือนกัน โดยที่สามารถทำงานได้หลายหลายรูปแบบ เช่น Acrobatic, Stabilize, Loiter และมีคำสั่ง Failsafe เพื่อใช้ในการนำเครื่องลงชุกเฉินและสามารถบินตามพิกัด ได้อีกด้วย



ภาพที่ 2-3 แสดงถึงอุปกรณ์ควบคุมการบิน Arduipilot

(<http://www.ardupilot.co.uk/>)

2.2.3 DJI NAZA เป็นอุปกรณ์ควบคุมการบินแบบ Commercial Source ที่มีหลากหลายราคา ตั้งแต่ราคาถูกจนถึงแพงมากแต่เนื่องจากเป็น Commercial Source ส่วนใหญ่จึงสามารถทำได้แค่ plug and play แต่เรื่องการบินความเสถียรถือว่าทำได้ดีตามราคา



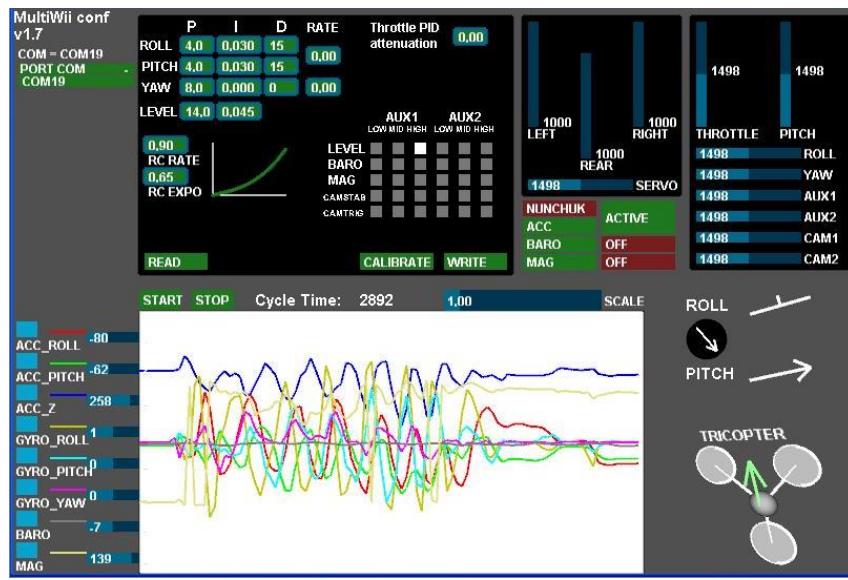
ภาพที่ 2-4 แสดงถึงอุปกรณ์ควบคุมการบิน NAZA – M ของ DJI  
(<http://www.dji.com/product/naza-m>)

### 2.3 Ground Station

คืออุปกรณ์แสดงผลภาพเพื่อดูที่รับข้อมูลกลับลงมาจากอุปกรณ์ควบคุมการบินเพื่อนำกลับมาแสดงผลเพื่อให้ผู้ใช้งานทราบสถานะของระบบและ Ground Station บางชนิดสามารถส่งคำสั่งกลับไปยังอุปกรณ์ควบคุมการบินได้อีกด้วย โดยที่ Ground Station มีหลายรูปแบบให้เลือกใช้งาน เช่น Software Ground Station, Tablet Ground Station, Hardware Ground Station

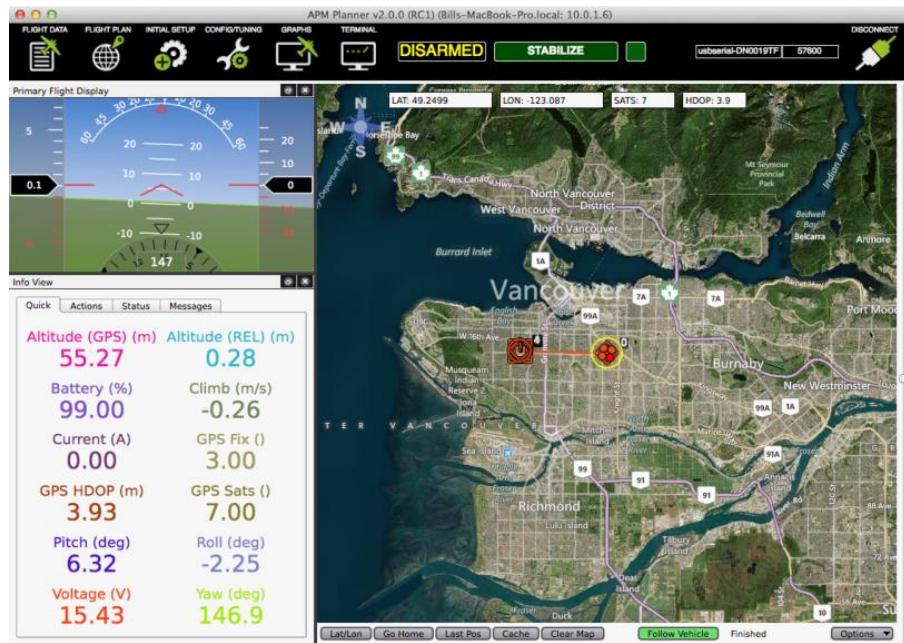
2.3.1 Software Ground Station คือ Ground Station ที่นิยมใช้มากที่สุดแบบหนึ่งเนื่องจากสะดวกและใช้อุปกรณ์ต่อพ่วงน้อย เช่นสาย Micro USB โดยที่โปรแกรมที่เป็นที่นิยมได้แก่ Mission Planner, MultiWii, Qground sControl, Naza Assistance Software โดยที่ Software ที่นำมาใช้งานขึ้นอยู่กับอุปกรณ์ควบคุมการบินที่เลือกใช้เนื่องจากล้วนใหญ่ผู้ผลิตอุปกรณ์ควบคุมการบินนั้นจะพัฒนา Ground Station ของตนเองออกมาเพื่อรับการใช้งานอีกด้วย

2.3.1.1 Multiwii เป็น Software ที่ออกแบบมาให้ใช้งานกับอุปกรณ์ควบคุมการบินของ Multiwii จึงถูกออกแบบมาให้ใช้งานง่าย สามารถปรับค่า PID อย่างง่ายให้แก่ระบบได้, ดูค่าข้อมูลพื้นฐานเช่น Pitch Roll Yaw ความเร็วในแต่ละแกนได้ และยังสามารถดูสถานะลำ และสามารถปรับค่าใหม่ต่าง ๆ ผ่านโปรแกรมได้เลย



ภาพที่ 2-5 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม MultiWii

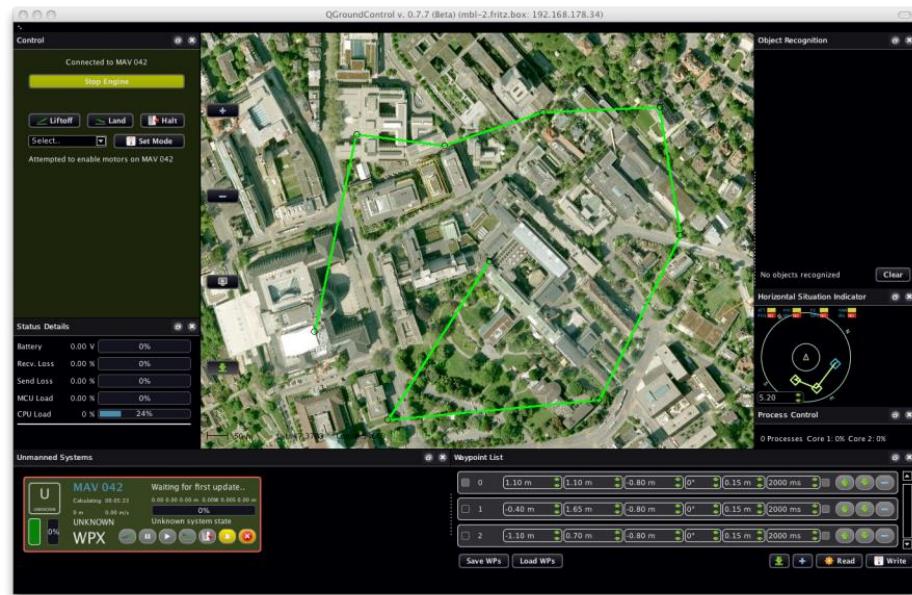
2.3.1.2 Mission Planner เป็น Software ที่ออกแบบมาเพื่อใช้งานกับตระกูล Arduipilot โดยเฉพาะดังนั้นจึงสามารถทำงานได้ดีกว่า MultiWii มาก แต่ด้วยการที่สามารถทำงานได้มากกว่า จึงทำให้การใช้งานนั้นยากกว่า MultiWii พoSมาร์จิ้งไม่เหมาะสมกับผู้เริ่มต้นในการใช้งาน โดยที่ความสามารถที่มากกว่า MultiWii นั้นคือสามารถปรับค่า Parameter ของตัว Multirotor อย่างละเอียดได้ สามารถกำหนดค่าการบินแบบเป็นพิกัดได้ สามารถอ่านค่า log จากการบินในแต่ละไฟล์ และนำเสนอข้อมูลในรูปแบบกราฟได้



ภาพที่ 2-6 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Mission planner

(<http://diydrones.com/profiles/blogs/apm-planner-2-0-rc1>)

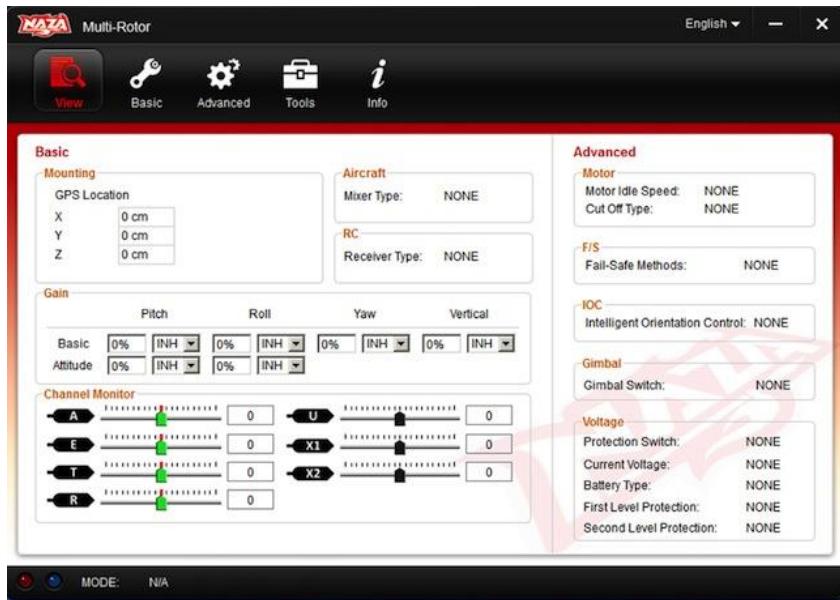
2.3.1.3 QGroundControl เป็น Open Source ที่ออกแบบมาสำหรับผู้ใช้งานขั้นสูง โดยที่ตัวโปรแกรมนี้ออกแบบมาเพื่อรองรับการพัฒนาต่อ โดยที่ลิขิตริมากกว่า Mission Planner นั้นคือ มี Developer Mode และ 3D map แต่ยังไม่เป็นที่นิยมเนื่องจากใช้งานยากและเพิ่งออกมาก็ได้ไม่นาน



ภาพที่ 2-7 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม QGroundControl

(<http://www.iheartrobotics.com/2010/08/qgroundcontrol-updates.html>)

2.3.1.4 Naza Assistant Software เป็น commercial source ที่ได้รับการพัฒนาจากทางบริษัท DJI เพื่อรองรับการใช้งานผ่านอุปกรณ์ควบคุมการบินของค่าย DJI เท่านั้นและ การใช้งานยังไม่รองรับการพัฒนาแบบต่อยอดหรือปรับค่าเพียงค่าพื้นฐานเท่านั้น

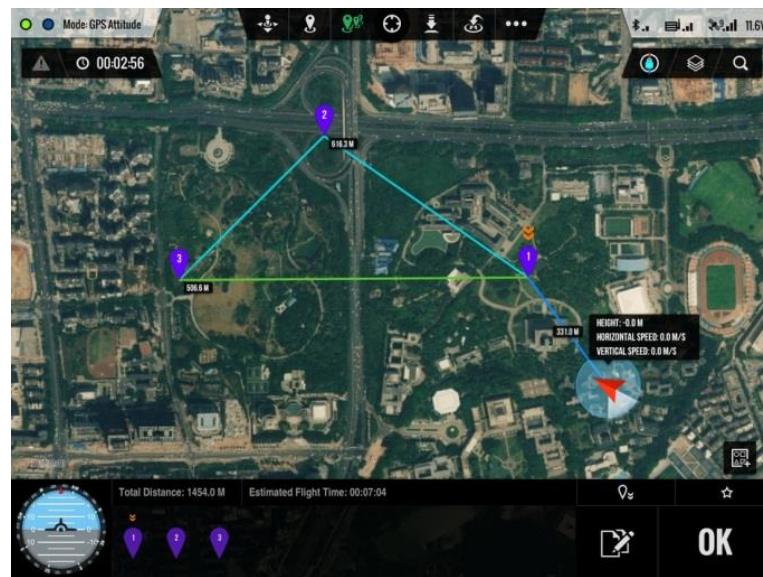


ภาพที่ 2-8 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Naza Assistant Software

([http://www.lafcpug.org/Tutorials/basic\\_arial\\_videoigraphy\\_pt2.html](http://www.lafcpug.org/Tutorials/basic_arial_videoigraphy_pt2.html))

2.3.2 Mobile/Tablet Ground Station เป็นจากปัจจุบัน โทรศัพท์เคลื่อนที่และแทปเลตเป็นอุปกรณ์ที่แพร่หลายดังนั้นจึงมีค่ายที่พัฒนา Ground Station ออกแบบให้ความสนใจในการที่จะพัฒนา Mobile/Tablet Platform ออกแบบให้ผู้ใช้ได้ใช้งาน

2.3.2.1 IPAD Ground Station คือ Ground Station ที่ทาง DJI ได้พัฒนาขึ้นมาเพื่อให้ผู้ใช้งาน DJI Flight Controller มีความสะดวกในการใช้งานมากขึ้น โดยที่ฟังก์ชันการทำงานหลัก ๆ นั้นเอาไว้สั่งบินตามพิกัดและส่งค่ากลับมาแสดงผลสู่ผู้ใช้งานเพียงเล็กน้อย เช่น ความสูง เวลาที่ใช้ในการบิน



ภาพที่ 2-9 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม IPAD Ground Station ผ่าน IPAD  
<http://www.twinprohobby.com/index.php?page=product&productid=2984> )

2.3.2.2 Droid Planner ก็อ Groud Station ที่ ออกแบบรองรับอุปกรณ์ควบคุมการบิน ตระกูล Arduipilot และเป็น Platform Android ซึ่งส่วนใหญ่สามารถทำงานได้เหมือน Mission Planner ทุกประการ เช่น ปรับค่า PID, ตั้งค่าโหมดการบิน ดูอาการของตัวลำ്ഭแบบเป็น Real time Graphic



ภาพที่ 2-10 แสดงถึงส่วนแสดงผลของผู้ใช้งานในโปรแกรม Droid Planner ผ่าน Android device  
<https://play.google.com/store/apps/details?id=com.droidplanner>)

2.3.3 Hardware Ground Station คือ Ground Station ที่ส่วนใหญ่จะออกแบบและพัฒนาเอง หรือเรียกอีกอย่างว่า DIY Ground Station ขึ้นอยู่กับผู้ออกแบบว่าต้องการใช้ Ground Station ของตัวเองนั้นมีระบบอย่างไรและสามารถทำอะไรได้บ้าง โดยที่หัวข้อโครงการนี้เป็นส่วนของการออกแบบ DIY Ground Station เช่นกัน ตัวอย่าง Hardware ground Station ในเชิง commercial นั้นคือ ArduStation

2.3.3.1 ArduStation คือ Commercial hardware Station ที่จัดจำหน่ายโดย jDrones ([http://store.jdrones.com/ArduStation\\_p/gcsad001.htm](http://store.jdrones.com/ArduStation_p/gcsad001.htm)) โดยที่ใช้ Arduino Atmega328P เป็นหน่วยประมวลผลหลักมาพร้อมกับ socket ไว้สำหรับ Xbee และมีจอ Monochrome สำหรับการแสดงผลแบบ 6 บรรทัด โดยที่มีซอฟต์แวร์ของ Arduipilot รองรับการทำงานอยู่



**ภาพที่ 2-11** Ardu Station Commercial hardware Station ที่จัดจำหน่ายโดย jDrones

(<http://store.jdrones.com/v/vspfiles/photos/gcsad001-2.jpg>)

## 2.4 Transmission System

คือตัวกลางที่มีไว้ใช้ในการสื่อสารระหว่างผู้ใช้ (User) กับอุปกรณ์ควบคุมการบินเพื่อส่งการมัลติโตร์หรือรับข้อมูลกลับมาแสดงผลที่ Ground Station โดยทั่วไปในการควบคุมการเคลื่อนที่ของมัลติโตร์นั้นจะใช้การส่งผ่าน Remote ที่คลื่นความถี่ RC Radio และ ข้อมูลส่งกลับมาสามารถทำได้หลายรูปแบบเช่น Bluetooth, Zigbee, WI-Fi

2.4.1 Rc Radio เป็นความถี่ที่นิยมใช้ในการสื่อสารระหว่างรีโมททวิตบังคับ และ Multirotor ที่มีความน่าเชื่อมถือสูง โดยที่เทคโนโลยีในการรับส่งนั้นขึ้นอยู่กับผู้ผลิตซึ่งมีหลายเจ้า อาทิเช่น Futaba FHSS (Futaba Frequency Hopping Spread Spectrum), JR Spektrum DSMX technology, Frsky ACCST (Advanced Continuous Channel Shifting Technology)



ภาพที่ 2-12 Futaba FHSS (Futaba Frequency Hopping Spread Spectrum)

(<http://www.futaba-rc.com/systems/futk9200-10j/>)

2.4.2 Bluetooth เป็นอีกหนึ่งอุปกรณ์ที่นิยมใช้ในการรับส่งข้อมูลระหว่างตัวลำและ Ground Station เมื่อจากใช้งานง่ายและราคาถูกและเป็นการเชื่อมต่อแบบ Serial แต่มีข้อเสียเรื่องระยะทางในการส่งสามารถส่งได้ในระยะไม่เกิน 20 เมตรขึ้นอยู่กับสภาพแวดล้อม



ภาพที่ 2- 13 Bluetooth

([http://www.emartee.com/product/41356/Serial-Port-Bluetooth-Module-With-Baseboard-\(Master\)](http://www.emartee.com/product/41356/Serial-Port-Bluetooth-Module-With-Baseboard-(Master)))

2.4.3 Zigbee เป็น Datalink ชนิดหนึ่งที่สามารถนำมาใช้งานแทน Bluetooth ได้อย่างมีประสิทธิภาพกว่าและออกแบบมาเพื่อให้ใช้พลังงานต่ำโดยที่ไม่เป็นที่นิยมในการใช้งานเนื่องจากมีราคางบประมาณโดยมีบริษัทผลิตเป็น product ที่นิยมใช้ได้แก่ Xbee โดยที่หลากหลายให้เลือก



ภาพที่ 2-14 XBee pro

(<http://www.emartee.com/product/41356>)

## 2.5 ไมโครคอนโทรลเลอร์สำหรับการออกแบบ Ground Station

ในการออกแบบ Ground Station สำหรับอากาศยานสี่ใบพัดแบบพกพา นั้นจำเป็นต้องใช้ ไมโครคอนโทรลเลอร์ เป็นเสมือนศูนย์กลางในการติดต่อสื่อสารระหว่าง Ground Station เช้ากับ MavLink Gateway ใช้ติดต่อกับส่วนแสดงผล ใช้ติดต่อสื่อสารกับ Hardware อุปกรณ์ เช่น joystick เพื่อ ใช้ในการส่งค่ากลับไปยัง MavLink Gateway ดังนั้นการเลือกใช้งาน ไมโครคอนโทรลเลอร์จึงมีผล อย่างมากต่อประสิทธิภาพการทำงานของระบบ จึงมีการเลือก ไมโครคอนโทรลเลอร์มาพิจารณา ดังต่อไปนี้ Arduino MEGA, Arduino DUE, STM32F4 Discovery โดยนำมาเปรียบเทียบเป็นตาราง ดังนี้

ตารางที่ 2-1 ตารางเปรียบเทียบสเปคของ Microcontroller

Type	Arduino Mega 2560	Arduino DUE	STM32 Discovery
Chipset	ATmega2560	AT91SAM3X8E	STM32F4
Family	AVR	ARM	ARM
Voltage level	5V	3.3V	3.3V
Clock speed	16 MHz	84 MHz	168 MHz
No. Analog pin	16	12	16
No. Digital pin	54	54	100
No. UART	4	4	6
No. SPI	1	1	3
NO. I2C	1	2	3
Internal real time clock	no	yes	yes
EPROM SIZE	4 KB	-	-
SRAM SIZE	8 KB	96 KB	192 KB
Flash SIZE	256 KB	512 KB	1 M
USB SIZE	Regular	Micro	Mini
PRICE (2015)	1650	1575	720

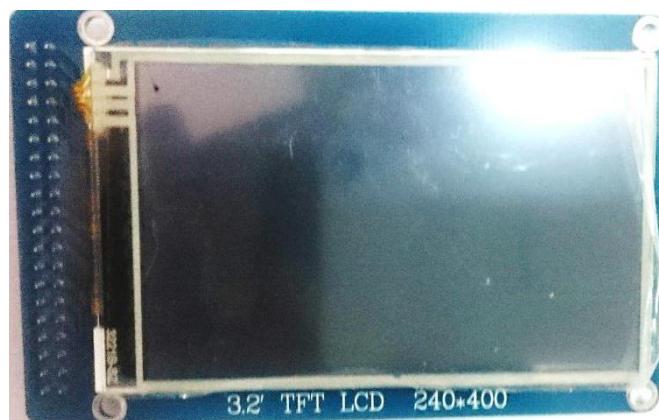
## 2.6 ส่วนแสดงผล สำหรับการออกแบบ Ground Station

เนื่องจากในการออกแบบ Ground Station นั้นส่วนแสดงผลถือเป็นส่วนสำคัญ เพราะเป็นส่วนที่ต้องติดต่อกับผู้ใช้งาน (USER) โดยตรง ซึ่งการใช้งานส่วนใหญ่นั้นเป็นสถานที่โล่งแจ้ง (Outdoor) ดังนั้นปัจจัยในการพิจารณาจึงมีหลายปัจจัย อาทิ เช่น ขนาดของจอ ความสว่างของจอ ความละเอียดของจอและราคาไม่สูงมากนัก และเนื่องจากเป็นงานที่ต้องใช้ความละเอียดของจอ ดังนั้นจึงทำการตัดขอที่เป็น Monochrome ออกไป และเน้นการพิจารณาไปยังไปกราฟฟิกแอลซีดีแทน จึงเลือกจากมาพิจารณาดังนี้ โมดูล ITDB24E\_8 (Chipset S6D1121), TFT01\_18SP (Chipset ST77355) และ LB04301 (Chipset SD1963) โดยที่แต่ละโมดูลมีข้อเปรียบเทียบดังนี้

ตารางที่ 2-2 ตารางเปรียบเทียบคุณสมบัติของจอกาฟฟิกแอลซีดีรุ่นต่างๆ

<b>Type of Module</b>	ITDB24E_8	TFT01_18SP	LB04301
<b>Chipset</b>	S6D1121	ST77355	SD1963
	2.4”	1.8”	3.2”
<b>Resolution</b>	240 x 320 pixels	128x160 pixel	240x400 pixels
<b>Mode</b>	8080	SPI Interface	8080
<b>Interface Type</b>	8-bit data 4 pin interface (CS,D/C,R/D,WR)	5 pin interface (CS, RS, MOSI, MISO, RST)	16-bit data 4 pin interface (CS,D/C,R/D,WR)
<b>Socket SD card</b>	Yes	Yes	Yes
<b>Logic level</b>	3.3V , 5V	3.3V	3.3V

ข้อสังเกตว่าที่มีขนาดเล็กและมีความละเอียดของจอต่ำนี้มักจะใช้จำนวนขาในการส่งข้อมูลน้อย เช่น TFT01\_18SP นั้นใช้เพียง 5 pin ในการเชื่อมต่อและเป็นการเชื่อมต่อแบบ SPI และในขณะที่จอที่มีขนาดใหญ่และมีความละเอียดสูงกว่าอย่างเช่น LB04301 นั้น ใช้ขาในการติดต่อถึง 20 pin โดยแบ่งเป็น ข้อมูลขนาด 16 pin และขาในการเชื่อมต่ออีก 4 pin



ภาพที่ 2-15 หน้าจอ TFT ขนาด 3.2 นิ้ว

## 2.7 SD Card

SD Card หรือ secure digital card เป็นอุปกรณ์จัดเก็บข้อมูลขนาดเล็กที่เป็นนิยมในปัจจุบันเนื่องจากพกพาง่ายสะดวกและราคาไม่แพง โดยถ้าหากเราแบ่งชนิดของข้อมูลตามความจุของ SD card สามารถแบ่งได้ดังนี้

**ตารางที่ 2-3** บ่งบอกการแบ่งประเภท SD card ตามความจุ

Type of SD card	Volume
SD standard capacity	Maximum 16GB
SD high capacity	Maximum 32GB
SD extended capacity	Maximum 2 TB

นอกจากความเร็วแล้วนั้น SD Card ยังแบ่งประเภทของอุปกรณ์ตาม Class ของความเร็วในการเขียนอ่านข้อมูลอีกด้วย โดยสามารถแบ่งได้ดังนี้

**ตารางที่ 2-4** บ่งบอกการแบ่งประเภท SD card ตามประสิทธิภาพของการเขียนข้อมูล

Class	Minimum Performance	Application
Class 2	2 MB/s	SD video recording
Class 4	4 MB/s	High-definition video (HD) recording (720 P)
Class 6	6 MB/s	Full HD (1080p) video recording
Class 10	10 MB/s	Real-time broadcasts and large HD video files (UHS bus)
UHS Speed Class 1 (U1)	10 MB/s	Real-time broadcasts and large HD video files (UHS bus)
UHS Speed Class 3 (U3)	30 MB/s	4K video files (UHS bus)

โดยที่ File System ของ SD card แบ่งออกได้เป็นหลายประเภท อาทิเช่น FAT12, FAT16, FAT32, MBR และ NTFS โดยที่นิยมใช้ในทั่วไปและกลุ่มสมองกลฝังตัวนั้นคือ FAT32 ซึ่ง FAT32 นั้นในแต่ละเซกเตอร์นั้นต้องมีพื้นที่ไม่ต่ำกว่า 7.8 GB และใช้ interface ที่นิยมใช้ในกลุ่มสมองกลฝังตัวคือการเขื่อมต่อแบบ SPI

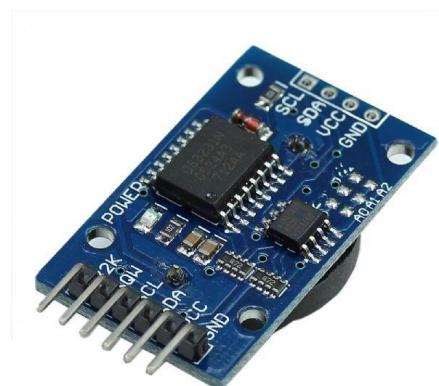


ภาพที่ 2-16 SD Card

(<http://www.amazon.com/SanDisk-Extreme-Class-Memory-SDSDX-032G-X46/dp/B004Q3C98S>)

## 2.8 Real time clock

Real time clock คืออุปกรณ์ที่ใช้ในการบอกรเวลาจริงเนื่องจากไม่ต้องการให้เวลาล่าช้า แต่ต้องแบ่งเวลาในการไปทำ Task อื่น เพราะจะนั่นการมีตัวบ่งบอกเวลาจากแหล่งอื่นนั้นจึงมีความแม่นยำสูงกว่า โดยที่ Real time clock นั้นสามารถทำงานได้โดยใช้ Crystal ในการนับเวลาและใน Real time clock บางรุ่นนั้นมี IC ในการวัดอุณหภูมิเพื่อปรับค่าชดเชยจาก Error ที่เกิดจากอุณหภูมิที่สูงขึ้นหรือลงลดได้อีกด้วย



## ภาพที่ 2-17 Real time clock

([http://www.spelecshop.com/1402480/592140/DS3231-High-Precision-Real-Time-Clock-\(RTC\).html](http://www.spelecshop.com/1402480/592140/DS3231-High-Precision-Real-Time-Clock-(RTC).html))

## 2.9 Rotary Encoder

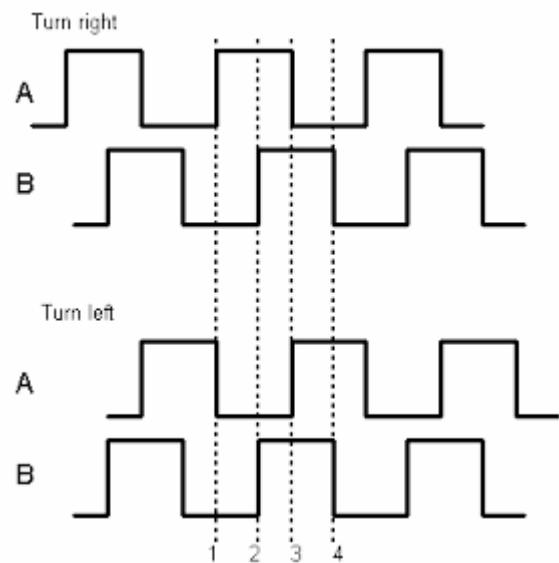
Rotary Encoder คือ อุปกรณ์ที่ใช้ในการ Encoder รูปแบบหนึ่งซึ่งสามารถแปลงตำแหน่งเชิงมุมให้มาอยู่ในรูปของสัญญาณดิจิทัลได้ โดยที่ Encoder นี้มีสองแบบได้แก่ Absolute Encoder และ Incremental Encoder ซึ่ง Absolute Encoder สามารถบอกได้เพียงตำแหน่งของแกนของ Encoder ในขณะที่ Incremental สามารถบอกข้อมูลได้ละอ่อนกว่า เช่นทิศทางของการหมุน ตำแหน่งปัจจุบันของการหมุน และระยะทางของการหมุน โดยที่การประยุกต์ของ Rotary Encoder นั้นหลากหลาย เช่น การใช้งานในเชิงของหุ่นยนต์ การใช้ในแมสที่เป็น Track ball เป็นต้น ในส่วนของการใช้งานของการโปรแกรมมิ่งนั้นจะมุ่งเน้นไปที่การศึกษาของ Encoder แบบ incremental โดยที่การทำงานนั้นจะมีสัญญาณ 2 เส้นคือ Signal A และ Signal B ที่เป็น Output ซึ่งเรียกว่า Quadrature outputs โดยที่มีผังของ Output ดังนี้

ตารางที่ 2-5 แสดงถึงพฤติกรรมของสัญญาณที่ทำให้ Rotary Encoder หมุนในทิศทางทวนเข็มนาฬิกา

Phase	Signal A	Signal B
1	0	0
2	0	1
3	1	1
4	1	0

ตารางที่ 2-6 แสดงถึงพฤติกรรมของสัญญาณที่ทำให้ Rotary Encoder หมุนในทิศทางตามเข็มนาฬิกา

Phase	Signal A	Signal B
1	1	0
2	1	1
3	0	1
4	0	0



ภาพที่ 2-18 แสดงถึงรูปสัญญาณของการหมุนซ้ายขวาของ Rotary encoder

(<http://alibaspinar-eee.blogspot.com/2013/01/rotary-encoder.html>)



ภาพที่ 2-19 แสดงถึง Rotary Encoder

(<http://www.arduinotronics.com/product/147/rotary-encoder-module-brick-sensor>)

## บทที่ 3

### วิธีการดำเนินงาน

ในการออกแบบและพัฒนาระบบความคุณและแสดงผลภาคพื้นดินแบบปกพำสำหรับอากาศยาน ไร้คนขับแบบ 4 ในพัดนั้น เป็นการออกแบบและพัฒนาระบบเพื่อใช้การในแสดงผลและเก็บข้อมูลของการบินและความคุณอากาศยาน ไร้คนขับสีใบพัด โดยได้แบ่งแผนการดำเนินงานดังต่อไปนี้

- 3.1 การวิเคราะห์เพื่อการออกแบบระบบ
- 3.2 การออกแบบระบบด้านฮาร์ดแวร์
- 3.3 การออกแบบระบบด้านซอฟต์แวร์
- 3.4 โครงสร้างซอฟต์แวร์
- 3.5 การออกแบบโปรโตคอล
- 3.6 การออกแบบการติดตั้งอุปกรณ์

#### 3.1 การวิเคราะห์เพื่อการออกแบบระบบ

ในการออกแบบระบบความคุณและแสดงผลภาคพื้นดินแบบปกพำสำหรับอากาศยาน ไร้คนขับแบบ 4 ในพัดนั้น แบ่งการออกแบบฮาร์ดแวร์เป็น 3 ส่วนได้แก่ ส่วนอุปกรณ์แสดงผลให้กับผู้ใช้งาน ส่วนอุปกรณ์ควบคุมการบินและติดต่อกับผู้ใช้ และส่วนอุปกรณ์เก็บบันทึกข้อมูลการบิน

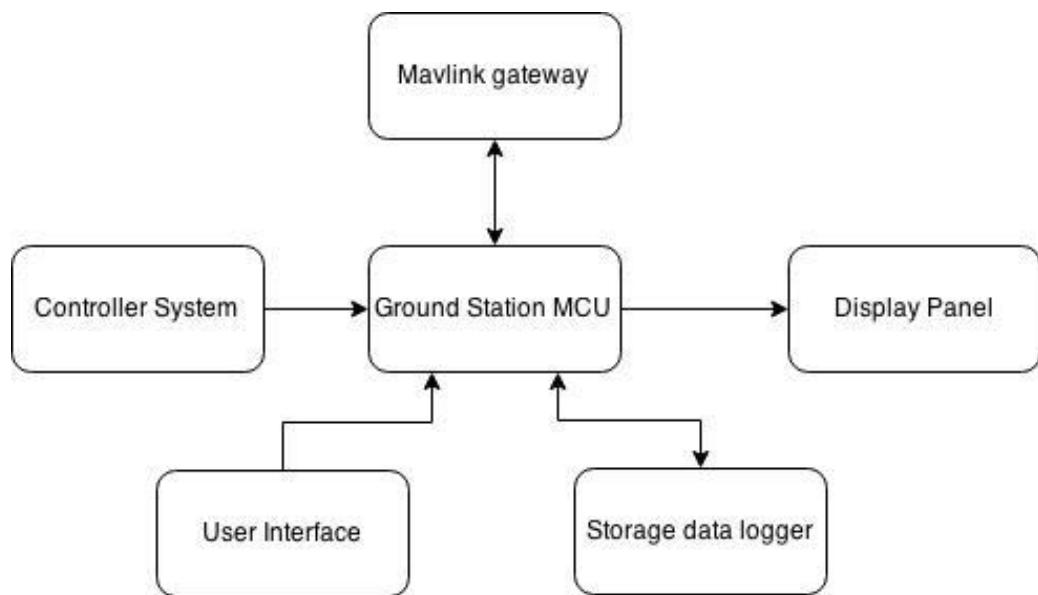
3.1.1 ส่วนอุปกรณ์แสดงผลให้กับผู้ใช้งาน ใน การออกแบบส่วนแสดงผลนั้น ได้เลือกใช้จอที่มีความละเอียดสูงและขนาดใหญ่เพื่อให้เพียงพอต่อการใช้งานของผู้ใช้งาน Backlight สว่างเพียงพอต่อการนำไปใช้งานนอกสถานที่และมีบัสในการเชื่อมต่อที่มีความเร็วสูงเพียงพอเพื่อให้การใช้งานเป็นไปอย่างมีประสิทธิภาพ

3.1.2 ส่วนอุปกรณ์ควบคุมการบินและติดต่อกับผู้ใช้ ในส่วนของการควบคุมการบินต้องใช้อุปกรณ์ที่เชื่อมต่อได้ มีมาตรฐาน และผู้ใช้สามารถเรียนรู้ได้โดยง่ายเพื่อความปลอดภัยของอากาศยาน 4 ในพัด และการเรียนรู้ในการใช้งานระบบความคุณและแสดงผลภาคพื้นดินแบบปกพำสำหรับอากาศยาน ไร้คนขับแบบ 4 ในพัด และในส่วนของการติดต่อกับผู้ใช้งานนั้น ได้คำนึงถึงความสะดวกของผู้ใช้เพื่อให้สามารถเข้าใจที่จะใช้งานได้ง่าย

3.1.3 ส่วนอุปกรณ์เก็บบันทึกข้อมูลการบิน ออกแบบเพื่อให้สามารถบันทึกข้อมูลของระบบภายในของมัลติโรเตอร์ในการบิน เพื่อให้ผู้ใช้งานสามารถมาเยือนดูภายหลังได้หากเกิดสุดวิสัย

ขึ้นกับการบินของผู้ใช้ หรือการวิเคราะห์ข้อมูลย้อนหลังก็สามารถทำได้ เช่นกัน โดยที่การบันทึกข้อมูลนั้นควรอ้างอิงกับเวลาจริงเพื่อสามารถนำไปใช้ตรวจสอบย้อนหลังกับการใช้งานจริงได้โดยใช้เวลาเป็นตัวอ้างอิง

หลักจากวิเคราะห์ปัญหาเบื้องต้น จึงสามารถนำมาออกแบบเป็น บล็อกไซด์แกรมเพื่อแก้ปัญหาได้ดังภาพ



ภาพที่ 3-1 การออกแบบ Ground Station เบื้องต้น

จากการทดสอบถึงหน่วยอย่างต่าง ๆ ที่ผู้จัดทำได้ออกแบบโครงสร้างของ Ground Station เบื้องต้นขึ้นมาในหัวข้อถัดไปจะพูดถึงการออกแบบและเลือกอุปกรณ์ที่เหมาะสมกับการสร้างชิ้นงาน Ground Station

### 3.2 การออกแบบระบบค้านฮาร์ดแวร์

3.2.1 สำหรับการเลือกอุปกรณ์ประมวลผลสำหรับ Ground Station นั้น ได้เลือกรุ่นของ Microcontroller มาพิจารณา จากบทที่ 2 มีตัวเลือกดังนี้ Arduino Mega, Arduino DUE, STM32 Discovery โดยนำตารางที่ 2-1 มาเปรียบเทียบดังนี้

ตารางที่ 3- 1 ตารางเปรียบเทียบประสิทธิภาพของ ไมโครคอนโทรลเลอร์ในรุ่นต่างๆ

Type	Arduino Mega 2560	Arduino DUE	STM32 Discovery
Chipset	ATmega2560	AT91SAM3X8E	STM32F4
Family	AVR	ARM	ARM
Voltage level	5V	3.3V	3.3V
Clock speed	16 Mhz	84 Mhz	168 Mhz
No. Analog pin	16	12	16
No. Digital pin	54	54	100
No. UART	4	4	6
No. SPI	1	1	3
NO. I2C	1	2	3
Internal real time clock	no	yes	yes
EPROM SIZE	4 KB	-	-
SRAM SIZE	8 KB	96 KB	192 KB
Flash SIZE	256 KB	512 KB	1 M
USB SIZE	Regular	Micro	Mini
PRICE (2015)	1650	1575	720

จากตารางข้างต้นนั้นนั่นจะเห็นว่า STM32 นั้นมีประสิทธิภาพสูงกว่าอีกทั้งสองตัว แต่ในขณะที่ตระกูล Arduino มีข้อได้เปรียบมากกว่าในเชิงของชุมชนนักพัฒนา เพราะว่ามีการใช้งานที่หลากหลายแต่ต่อยอดได้ง่าย เนื่องจากมีความซับซ้อนของการใช้งานน้อยกว่า STM32 discovery แต่ในขณะที่ต้องการความเร็วในการเชื่อมต่อที่สูง Arduino Mega จึงไม่ตอบปัญหาในข้อนี้ได้จึงได้ทำการเลือก Arduino DUE ที่ความเร็วในการประมวลผลสูงสุดในกลุ่ม Arduino ด้วยกัน



ภาพที่ 3-2 Arduino DUE

3.2.2 ส่วนอุปกรณ์แสดงผลให้กับผู้ใช้งาน ในการออกแบบส่วนแสดงผลนั้นผู้จัดทำต้องการ  
จะที่มีขนาดใหญ่ขึ้นจาก 3.2" เพราะเนื่องจากน่าจะมีขนาดใหญ่เพียงพอสำหรับการใช้งาน โดยที่จาก  
บทที่ 2 นี้มีการเปรียบเทียบคุณสมบัติของหน้าจอจึงเลือก โมดูล LB04301 ที่ใช้ Chipset SSD1963  
ซึ่งเป็นการ interface แบบ 16-bit parallel



ภาพที่ 3-3 Block diagram ระหว่าง Arduino DUE และ GLCD โมดูล LB04301



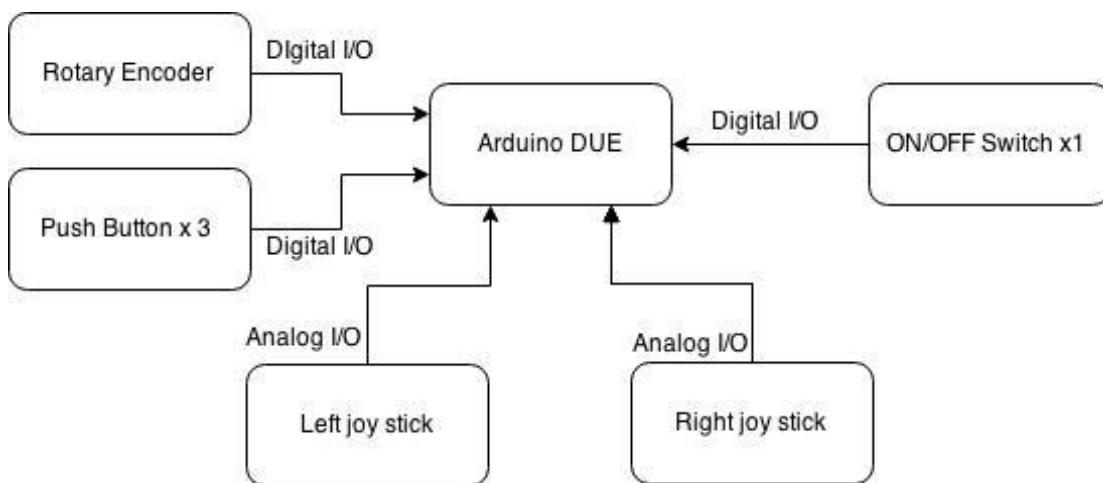
ภาพที่ 3-4 GLCD Module LB04301 Chipset SSD1963



ภาพที่ 3-5 GLCD Module LB04301 Chipset SSD1963 ด้านหลังพร้อม SD card Module

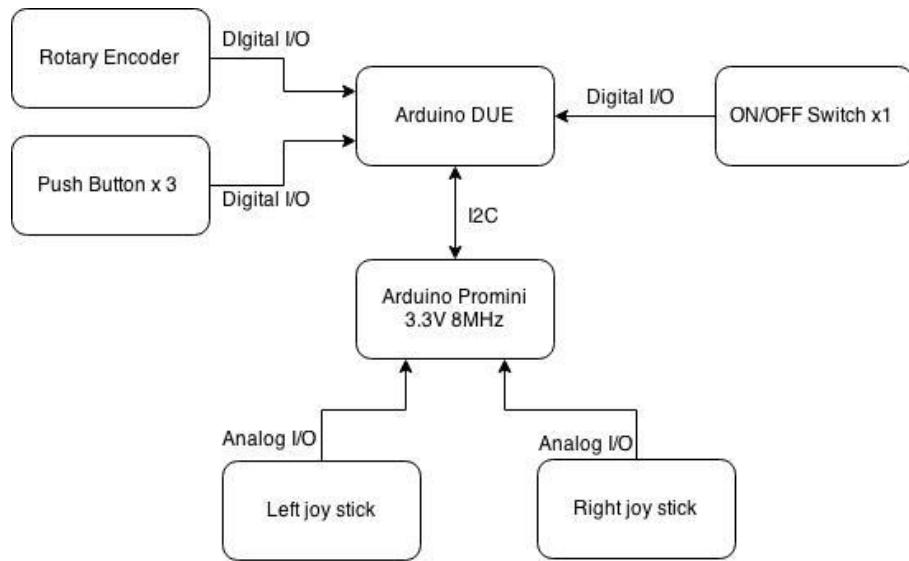
3.2.3 ส่วนอุปกรณ์ควบคุมการบิน ในการออกแบบส่วนติดต่อกับผู้ใช้งานนี้ ได้ทำการเลือกใช้งานปุ่มกดแบบกดติดปล่อยดับเพื่อให้ผู้ใช้งานสามารถเลือกเมนูหรือหน้าแสดงผล, ON/OFF switch เพื่อสำหรับการให้ผู้ใช้งานเลือกโหมด ได้ว่าต้องการควบคุมการบินหรือไม่ และ Rotary encoder ในการปรับแต่งค่าต่าง ๆ ให้แก่ระบบควบคุมและแสดงผลภาคพื้นดินแบบพกพา สำหรับอากาศยาน ไร้คนขับแบบ 4 ใบพัด ส่วนควบคุมนี้ ได้เลือกใช้ Gimbal module แบบสองแกน โดยที่มีลักษณะการเชื่อมต่อแบบ Analog output

3.2.3.1 การเชื่อมต่อลักษณะที่ 1 นี้ ได้ทำการเชื่อมต่อ Gimbal module แบบสองแกน เข้าโดยตรงกับ Analog input ของ Arduino DUE



ภาพที่ 3-6 แสดงการเชื่อมต่อของส่วนสื่อสารกับผู้ใช้งานและส่วนควบคุมการบิน โดยใช้การออกแบบลักษณะที่ 1

3.2.3.2 การเชื่อมต่อลักษณะที่ 2 (การเชื่อมต่อทางเลือก) นั้นได้ทำการเชื่อมต่อ Gimbal module แบบสองแกนผ่าน Arduino pro 3.3 V 8 MHz จากนั้นจึงให้ผ่าน Arduino pro 3.3 V 8 MHz ส่งค่าผ่าน I2C ไปยัง Arduino DUE ดังภาพ



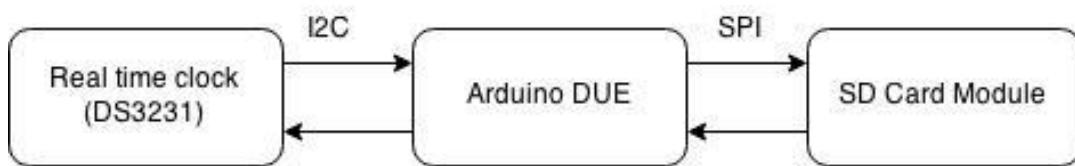
ภาพที่ 3-7 แสดงการเชื่อมต่อของส่วนสื่อสารกับผู้ใช้งานและส่วนความคุณการบิน โดยใช้การออกแบบลักษณะที่ 2

โดยการต่อในแบบ 3.2.3.2 นั้นมีข้อดีตรงที่ไม่เปลืองทรัพยากรถอยในของ Arduino DUE ที่ต้องค่าอ่านค่า Analog input จาก Gimbal module ตลอดเวลา และใช้ Arduino pro mini 3.3V แทน แต่หากว่า Arduino DUE นั้นมี Clock speed สูงถึง 84 MHz ดังนั้นมีประสิทธิภาพเพียงพอในการที่จะอ่านค่า Analog input จาก Gimbal module ดังนั้นจึงเป็นการสืบเปลืองอุปกรณ์ Arduino pro mini 3.3V 8MHz ดังนั้นผู้จัดทำจึงเลือกการเชื่อมต่อการใช้งานดังหัวข้อ 3.2.3.1 แทน โดยที่ผังการเชื่อมต่ออย่างละเอียดจะอยู่ในบทที่ 4

### 3.2.4 ส่วนอุปกรณ์เก็บบันทึกข้อมูลการบิน

ในส่วนของการบันทึกข้อมูลของการบินนี้เลือกใช้การเก็บข้อมูลด้วย SD card เนื่องจากมีความสะดวกและใช้งานง่าย โดยที่ใช้ Module SD card ที่มาพร้อมกับ GLCD Module LB04301 และเพื่อความแม่นยำของการบันทึกและสามารถตรวจสอบย้อนหลังเมื่อเทียบกับเวลาจริง ได้จึงได้ทำการเพิ่มโมดูล Real time clock เบอร์ DS3231 เนื่องจากมีความแม่นจำสูงและราคาไม่แพงเพื่อทำการบันทึกเวลาขณะที่ทำการบันทึกค่าของข้อมูล ไปยัง SD card ซึ่ง SD card นั้นทำการเชื่อมต่อด้วย SPI บัส และโมดูล Real time clock เชื่อมต่อด้วย I2C ซึ่งเชื่อมต่อได้ดังภาพ ข้อสังเกต

เนื่องจาก Arduino DUE นั้นได้มี Real time clock ภายในอยู่แล้วแต่เหตุที่นำ Real time clock มาต่อเพิ่มเนื่องจากว่า Real time clock ภายในนั้นไม่อนุญาตให้ต่อ Battery สำรองจากภายนอกได้นั้นทำให้ Real time clock ภายในนั้นไม่สามารถบอกรเวลาปัจจุบันได้นั้นเองซึ่งทำให้การใช้งานต้องตั้งค่าเวลาปัจจุบันใหม่ทุกครั้งซึ่งทำให้ไม่เหมาะสมแก่การใช้งานจริง

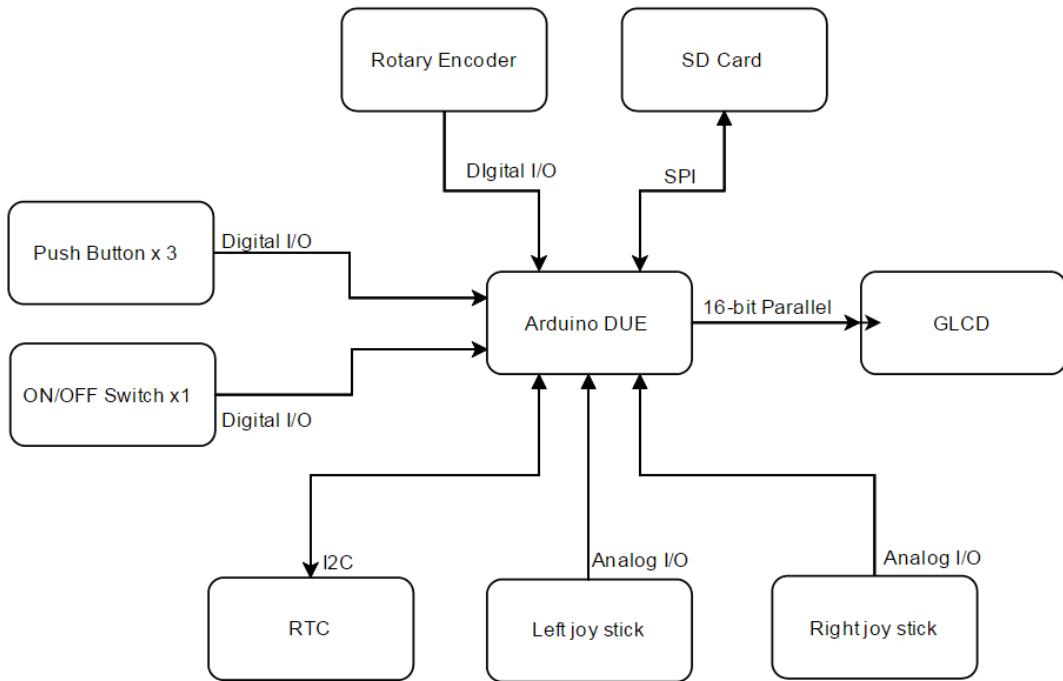


ภาพที่ 3-8 แสดงการเชื่อมต่อของส่วนเก็บบันทึกข้อมูลการบินเข้ากับ Arduino DUE

หลังจากที่ได้ทำการเลือกอุปกรณ์มาให้เหมาะสมสมกับการสร้างชิ้นงานแล้วจึงสรุปการเลือกใช้อุปกรณ์ที่ได้เลือกมาดังนี้

- |  |              |
|--|--------------|
| - Arduino DUE                                | จำนวน 1 ชิ้น |
| - GLCD Module LB04301 มาพร้อม SD card socket | จำนวน 1 ชิ้น |
| - Real time clock (DS3231)                   | จำนวน 1 ชิ้น |
| - Gimbal module (Throttle + Yaw)             | จำนวน 1 ชิ้น |
| - Gimbal module (Pitch + Roll)               | จำนวน 1 ชิ้น |
| - ON/OFF switch                              | จำนวน 1 ชิ้น |
| - Push button                                | จำนวน 3 ชิ้น |

และนำมาเขียนเป็น Block diagram ได้ดังนี้



ภาพที่ 3-9 Block diagram ของ Ground Station

### 3.3 การออกแบบระบบด้านซอฟต์แวร์

ระบบด้านซอฟต์แวร์ ที่ใช้กับ Ground Station นั้นต้องคำนึงการสร้างซอฟต์แวร์ ที่สามารถทำงานได้ตรงกับเงื่อนไขของผู้อุปกรณ์ที่กล่าวไว้ในบทที่ 1 ดังนั้นการออกแบบซอฟต์แวร์จึงมีความสำคัญต่อ Ground Station เป็นอย่างมาก โดยที่ส่วนการทำงานของซอฟต์แวร์มีส่วนที่สำคัญดังนี้

3.3.1 ในการติดต่อสื่อสารระหว่าง Ground Station และ Mavlink gateway เนื่องจากข้อมูลที่ส่งมาเป็น Custom protocol ซึ่งผู้จัดทำสร้างขึ้นมา (รายละเอียดโปรโตคอลอยู่ที่หัวข้อ 3.4) ซึ่งข้อมูลที่รับ/ส่งนั้นจำเป็นที่จะต้องทำการ Decode ข้อมูลที่ได้รับจาก Mavlink gateway ก่อนที่จะนำค่านั้น ๆ มาใช้ได้งาน จำเป็นต้องออกแบบซอฟต์แวร์เพื่อเป็นเครื่องมือที่ใช้ในการ Decode custom protocol ของการสื่อสารระหว่าง Ground Station และ Mavlink gateway เพื่อให้สามารถที่จะเข้าถึงข้อมูลนั้น ๆ ได้โดยง่าย

3.3.2 การออกแบบซอฟต์แวร์เพื่อทำการอ่านค่าจาก Mavlink gateway และนำมาแสดงผลบนหน้าจอ GLCD เพื่อให้ผู้ใช้งานสามารถทราบสถานะที่สำคัญของ Flight controller โดยมีความถี่ของการแสดงผลของ GLCD ต้องไม่ต่ำกว่า 5 Hz เพื่อที่ผู้ใช้งานสามารถอ่านค่าล่าสุดที่มาจากการของ Flight controller ได้เสมอ และเนื่องจากข้อมูลที่สำคัญนั้นไม่สามารถแสดงทั้งหมดภายในหน้าจอได้จึง

ต้องมีหน้าของการแสดงผลของหน้าจอ GLCD 2 หน้าเพื่อสามารถที่จะแสดงค่าได้อ่าย่างชัดเจน ซึ่งในการเปลี่ยนหน้าแต่ละหน้านั้นได้ออกแบบซอฟต์แวร์เพื่อสั่งการ Push button แบบ Interrupt และมี denouncing เพื่อทำการเปลี่ยนหน้าสามารถเป็นไปได้อย่างมีประสิทธิภาพ

3.3.3 สำหรับการอ่านค่าและปรับค่าของ Parameter ระหว่าง Ground Station และ Mavlink gateway นั้น จะเริ่มต้นทำการขอ Parameter จาก Mavlink gateway ทุกครั้งที่ทำการเปิดระบบและนำ Parameter ที่ได้รับไปเขียนลง SD card และทำการ Flush เมื่อรับค่า Parameter สำเร็จ เพราะเนื่องจากเก็บค่า Parameter ทั้งหมด 400 ค่า แต่ละค่ามีความยาว 21 Byte (การเขียนอธิบายเรื่องขนาด Parameter อยู่ในหัวข้อ 3.4) นั้นทำให้เปลืองทรัพยากรของ Microcontroller และไม่สามารถนำไปใช้งานกับ Microcontroller รุ่นที่มีขนาดเล็ก อาจจะทำหน่วยความจำไม่เพียงพอต่อการใช้งานได้ และในการจัดแสดงข้อมูลของ Parameter นั้น GLCD ต้องสามารถแสดงค่าและชื่อของ Parameter ทุกด้วยที่จะทำการจัดแสดงข้อมูลเป็นหน้า หน้าละ 11 ตัว เพื่อให้ตัวอักษรใหญ่พอเพื่อที่จะให้ผู้ใช้งานสามารถใช้งานได้โดยง่าย และ โดยที่ค่าที่นำมาแสดงเป็นค่าที่ดึงออกมาจาก SD card นั้นเอง ในการปรับค่านั้นต้องเขียนซอฟต์แวร์เพื่อสั่งการ Rotary encoder สำหรับการปรับค่า และทำการส่งค่ากลับไปยัง Mavlink gateway

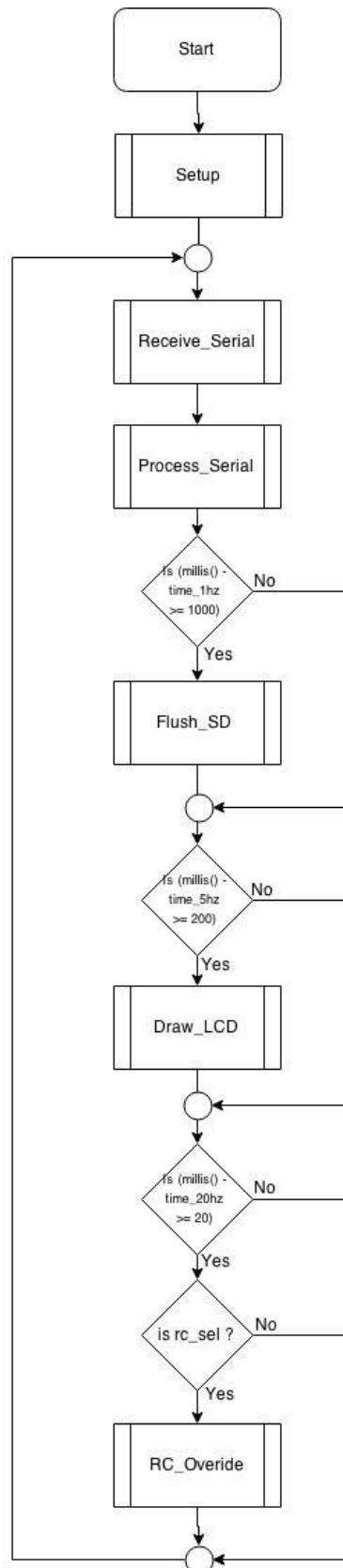
3.3.4 ในการที่ผู้ใช้งานนั้นสามารถใช้ Gimbal module เพื่อใช้ในการ RC override Flight controller นั้นจำเป็นต้องมีความถี่ในการส่งไม่ต่ำกว่า 20 Hz เพื่อให้สามารถส่งคำสั่งเป็นไปอย่างรวดเร็วและต่อเนื่อง และต้องมีส่วนที่แข็งให้ผู้ใช้งานทราบว่าเปิดใช้งาน RC override อยู่บนหน้าจอ GLCD ด้วย ซึ่งการเปิดใช้งาน RC override นั้นสามารถทำได้โดยทำการปรับตำแหน่งของ ON/OFF Switch ซึ่งต้องมีซอฟต์แวร์มารองรับการทำงานของ ON/OFF Switch อีกด้วย

3.3.5 ในส่วนของการเก็บข้อมูลลง SD card นั้นจะทำการเขียนซอฟต์แวร์เพื่อทำการ เขียนข้อมูลลง SD card ทุก ๆ ครั้งที่ได้รับข้อมูลใหม่จาก Mavlink gateway สำเร็จพร้อมทั้งบันทึกเวลาจริงด้วยเสมอจากนั้นทำการ Flush ข้อมูลลงไปยัง SD card ทุก ๆ 1 วินาที

### 3.4 โครงสร้างซอฟต์แวร์

หลังจากที่รับรู้ถึงโครงสร้างของซอฟต์แวร์ได้ทำการออกแบบซอฟต์แวร์ที่มีลักษณะดังนี้ ซอฟต์แวร์จะเริ่มจากการ setup และทำการร้องขอข้อมูลจาก Parameter Mavlink gateway และเริ่มต้นทำงาน โดยที่หลังจากรับ Parameter สำเร็จนั้นจะเริ่มทำการร้องขอข้อมูลเพื่อให้ Mavlink gateway ส่งข้อมูลมาให้และเริ่มเปิดใช้งานจอ GLCD พร้อมทั้งเขียนข้อมูล Parameter ที่ได้รับมาลงไปยัง SD card หลังจากนั้นเข้าสู่การทำงานหลักของโปรแกรม โดยที่มีฟังก์ชัน Receive serial ที่ทำหน้าที่รับข้อมูล Serial จาก Mavlink gateway เมื่อรับค่าจักรอบได้ทำการส่งค่าต่อไปยัง ฟังก์ชัน

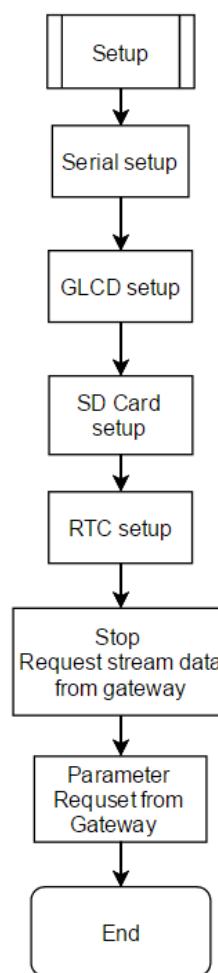
Process serial เพื่อทำการเขียนลง SD card และแสดงค่าอุณหภูมิเพื่อทำการ debug หลังจากนั้นจะมีฟังก์ชัน Flush SD card ที่จะทำการ Flush ข้อมูลไปยัง SD card โดยที่จะทำงานทุก ๆ 1 วินาที และมีฟังก์ชันสำหรับการอัพเดทหน้า GLCD ที่ทำงานด้วยความถี่ 5 ครั้งต่อวินาที และฟังก์ชันสำหรับการส่งข้อมูล RC override ที่copyอ่านค่าจาก Analog จำนวน 4 channel และทำการส่งค่า RC override ออกไปด้วยความถี่ 20 ครั้งต่อวินาที โดยที่สามารถเขียนเป็น flow chart ได้ดังนี้



ภาพที่ 3-10 Flow chart ของ Ground station

### 3.4.1 Setup

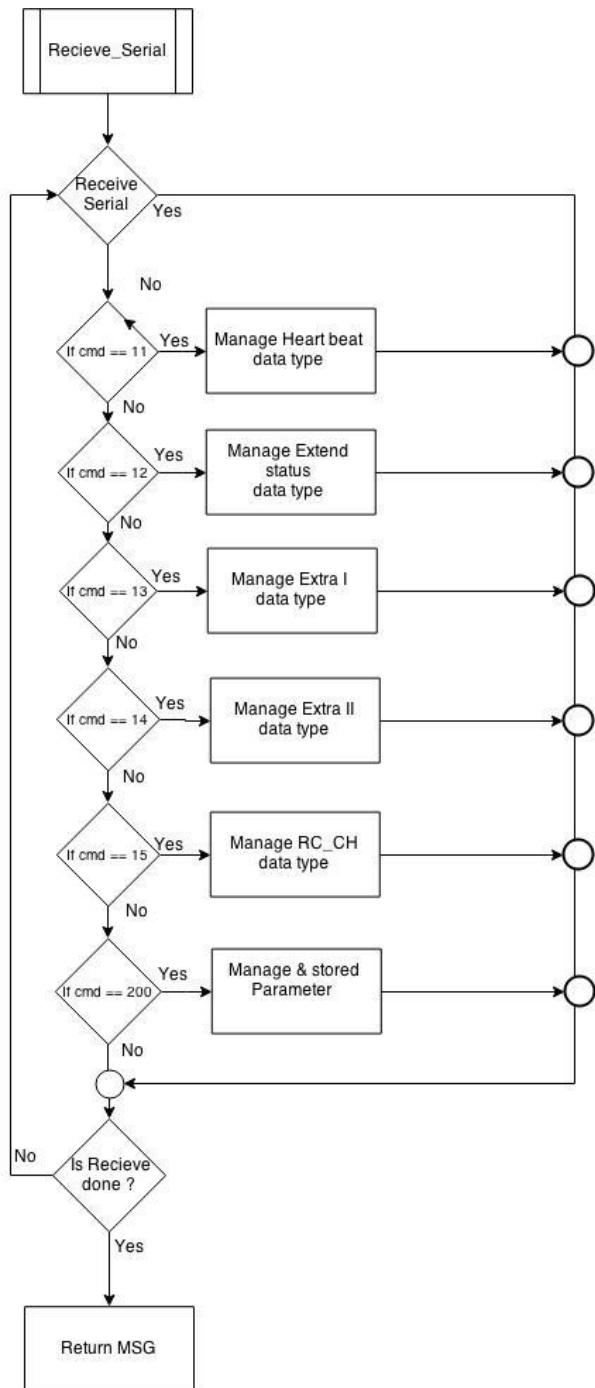
เริ่มต้นจากการกำหนดค่าเริ่มต้นให้แก่ระบบโดยที่กำหนดค่าเริ่มต้นให้แก่ตัวแปรต่างๆที่ใช้งานและกำหนดค่าเริ่มต้นให้แก่โมดูล Real time clock, SD card, GLCD และเปิดการใช้งาน Serial 1,Serial 2 สำหรับการ debug และสื่อสารกับ Mavlink gateway ตามลำดับ หลังจากนั้นทำการส่งข้อมูลไปบอค Mavlink gateway เพื่อหยุดการรับข้อมูลแล้วเริ่มทำการร้องขอ Parameter จาก Mavlink gate ตามลำดับ



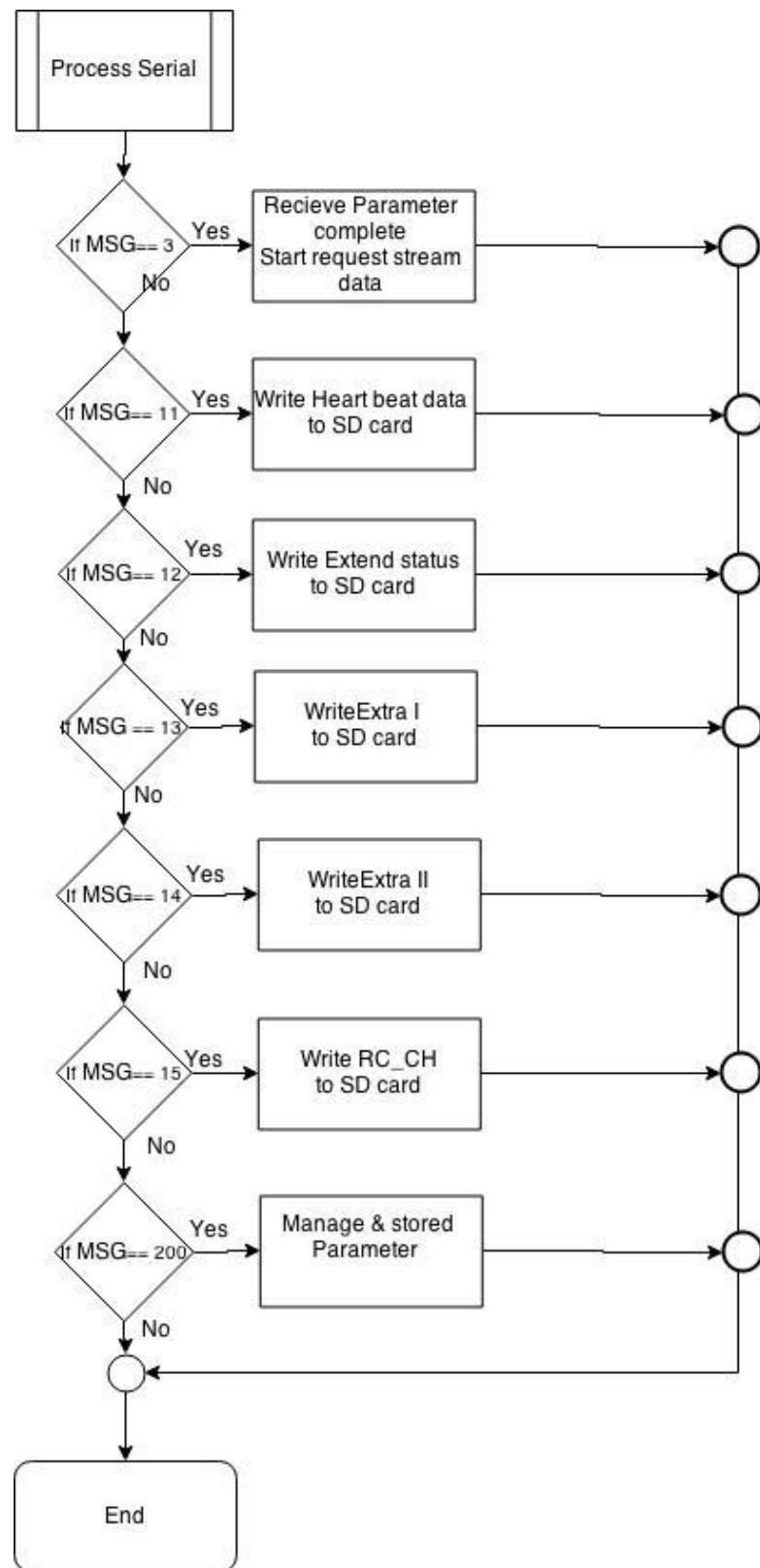
ภาพที่ 3-11 Flow chart subroutine setup

### 3.4.2 Receive & Process Serial

เมื่อทำการร้องขอ Parameter แล้วจะเริ่มต้นเข้าสู่การทำงานของโปรแกรมหลัก โดยที่ Mavlink gateway จะเริ่มทำการส่ง Parameter มาให้ Ground Station โดยทราบได้จากตัวแปรชื่อ CMD เป็นตัวแปรที่บอกว่าข้อมูลที่ได้รับจาก Mavlink gateway นั้นเป็นข้อมูลชนิดใดแล้วทำการ Process โดยตรวจสอบจาก CMD เพื่อคุณจำนวนข้อมูลที่เหลือ อาทิเช่น ข้อมูลที่มี CMD เป็น 200 ซอฟต์แวร์ จะทำการตรวจสอบจำนวน Byte ที่เหลือโดยกรณีนี้จะเหลือ 21 Byte (อ้างอิงจาก ไปรโ托คอลเน็ต웤ที่ 3.4) mana เป็นตัวอ้างอิงในการรับข้อมูลครั้งต่อไปหลังจากที่รับข้อมูลจนครบนั้น ก็จะนำข้อมูลที่รับมาไปเข้ากระบวนการแยกข้อมูลตามประเภทของข้อมูลที่ได้รับ หลังจากที่ได้รับข้อมูล Parameter จนครบนั้น Mavlink จะส่งความมาเพื่อบอกว่าสิ้นสุดการส่ง Parameter (CMD = 3) ซึ่งหลังจากนั้น Ground Station จะทำการส่งการร้องขอ Streaming ข้อมูล หมายเหตุ Groud Station โดยที่หลังจากนี้ทั้งสองฝั่งที่ทำงานต่อเนื่องกันโดยที่ Receive serial จะมีหน้าที่รับข้อมูลที่มาจาก Mavlink gateway และเมื่อรับมาสำเร็จจะส่ง MSG มาบอกยัง Process Serial ให้ทำการประมวลข้อมูลที่ได้รับและทำการบีบอัดข้อมูลลงยัง SD card พร้อมเวลาจริงทุกครั้ง ที่ได้รับ MSG จาก Receive serial โดยที่การทำงานเป็นไปดัง Flow chart ด้านล่างนี้



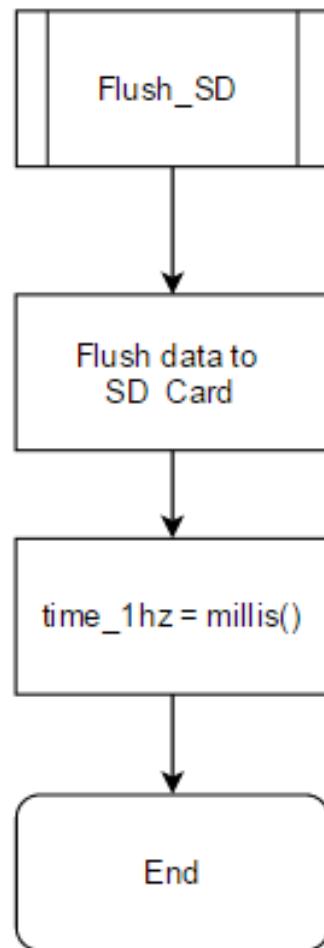
ภาพที่ 3-12 Flow chart subroutine Receive\_serial



ภาพที่ 3-13 Flow chart subroutine ProcessSerial

### 3.4.3 Flush\_SD

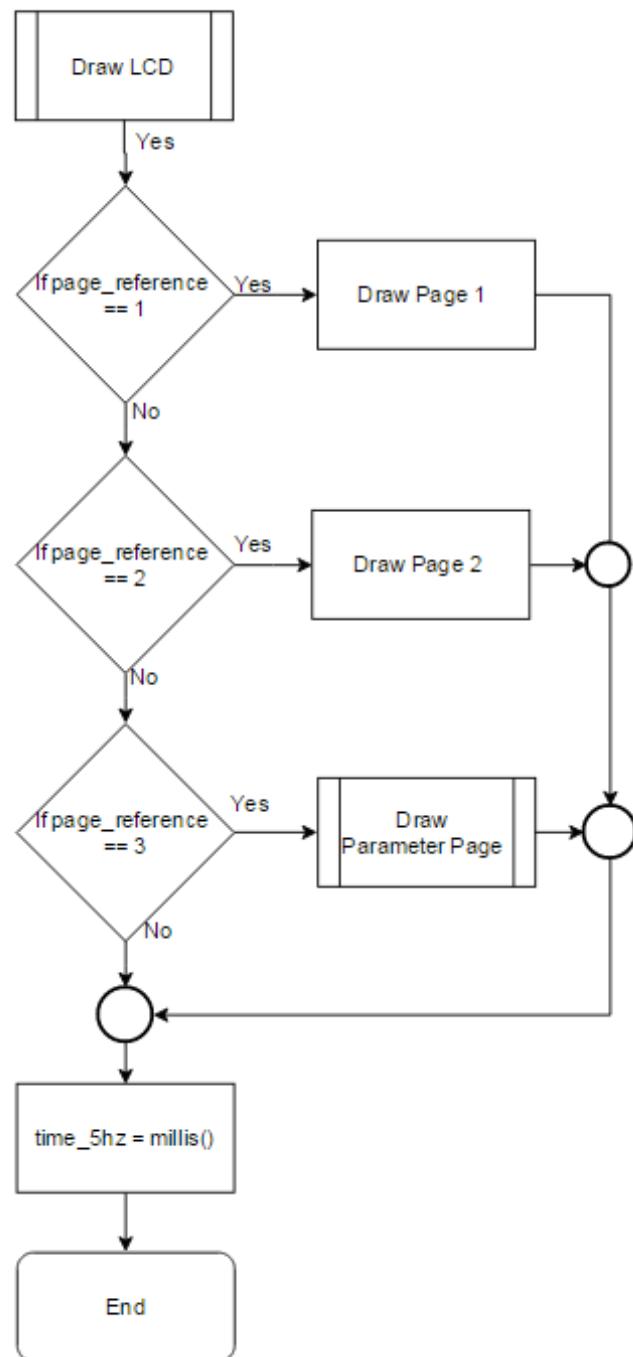
ในส่วนของการ Flush SD card นี้จะทำการ Flush ทุก ๆ 1 วินาที เมื่อทำการ Flush สำเร็จจะทำการรีเซ็ตเวลาที่ใช้ในการนับเวลา ดัง Flow chart ด้านล่าง



ภาพที่ 3-14 Flow chart subroutine Flush\_SD

### 3.4.4 Draw\_LCD

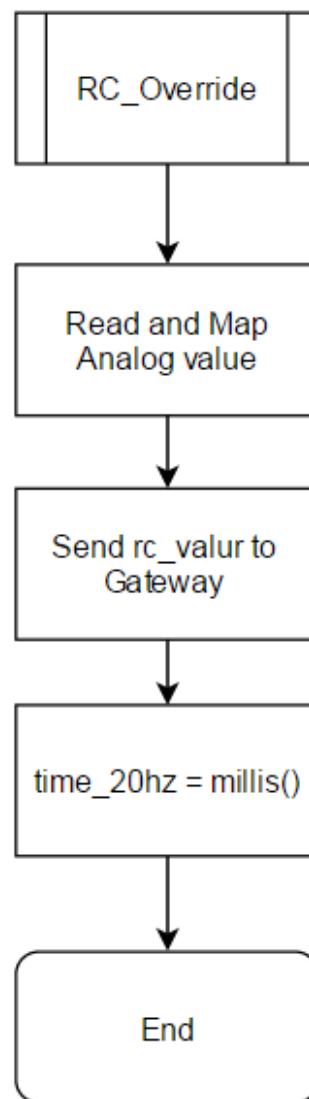
ในส่วนของฟังก์ชัน Draw\_LCD โดยจะทำการอัพเดททุก ๆ ห้าวินาทีโดยที่การวาดนั้นขึ้นอยู่กับ page\_preference เป็นตัวกำหนดในการวาดหน้าจอดัง Flow chart ด้านล่าง



ภาพที่ 3-15 Flow chart ของ Subroutine Draw LCD

### 3.4.5 RC\_override

ในส่วนของฟังก์ชัน RC\_override นี้ทำหน้าที่ในการอ่านค่า Analog และทำการปรับค่าให้อยู่ในช่วงที่ Mavlink gate ใช้งานและทำการส่งค่าออกไปยัง Mavlink gateway



ภาพที่ 3-16 Flow chart ของ Subroutine RC\_override

โดยที่ในการพัฒนาซอฟต์แวร์ดังกล่าวมี ได้ใช้โปรแกรมพัฒนาและ Library ดังนี้

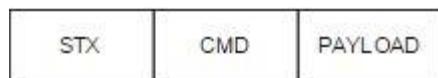
- Arduino IDE เวอร์ชัน 1.5.7
- # Library <SPI.h>  
สำหรับการสื่อสารด้วยโปรโตคอลเอส皮ไอ
- # Library <UTFT.h>  
สำหรับการเขียนต่อและสั่งงานกราฟฟิคແອລືດີກັບ Arduino
- # Library <Wire.h>  
สำหรับการสื่อสารระหว่าง Real time clock กับ Arduino DUE ด้วยโปรโตคอล-

### ไอส-แคร์ชี

- # Library <SD.h>  
สำหรับการเขียนอ่านข้อมูลลง SD card ใช้คู่กับ Library <SPI.h>
- # Library <DS3232RTC.h>  
สำหรับการติดต่อตัวอัตโนมัติและตั้งค่าให้กับ Real time clock
- # Library <Encoder.h>  
สำหรับการใช้งาน Rotary encoder
- # Library <GatewayDEC.h>  
สำหรับการรับส่งข้อมูลแบบ Serial กับ Mavlink gateway โดยผู้จัดทำเป็นคนเขียน  
ขึ้นมาซึ่งรายละเอียดของ Library อยู่ที่ภาคผนวก

### 3.5 การออกแบบโปรโตคอล

ในการติดต่อตัวอัตโนมัติระหว่างระบบควบคุมและแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยาน ไร้คนขับแบบ 4 ใบพัด และ Mavlink gateway นั้นอีกหนึ่งส่วนที่สำคัญนั้นคือโปรโตคอลระหว่างการสื่อสารเนื่องจากโปรโตคอลนี้มีผลต่อความเร็วรวมของระบบและข้อผิดพลาดของข้อมูล เนื่องจากข้อมูลที่ต้องการให้สื่อสารกันมีขนาดเล็กและกระชับดังนั้นจึงให้ความสนใจไปที่ไบ奴ารี โปรโตคอล โดยที่กำหนดให้โปรโตคอลประกอบไปด้วยสามส่วนดังนี้ STX คือส่วนเริ่มต้นของโปรโตคอล CMD ค่าสั่งของโปรโตคอล PAYLOAD คือข้อมูลของโปรโตคอลดังภาพ



ภาพที่ 3-17 การออกแบบไบ奴ารี โปรโตคอล

โดยที่ STX มีขนาด 1 ไบต์ มีค่าเป็น 0XFE

CMD มีขนาด 1 ไบต์ มีค่าเป็นเลขฐาน 10 ตั้งแต่ 0 – 255

โดยที่ CMD มีค่า 0 หมายถึง การร้องขอค่าอาร์ทบีทจาก Mavlink gateway

โดยที่ CMD มีค่า 1 หมายถึง การร้องขอให้หยุดส่งค่าจาก Mavlink gateway

โดยที่ CMD มีค่า 2 หมายถึง การร้องขอค่าพารามิเตอร์จาก Mavlink gateway

โดยที่ CMD มีค่า 3 หมายถึง รับพารามิเตอร์จาก Mavlink gateway สำเร็จ

โดยที่ CMD มีค่า 11 - 15 หมายถึง การได้รับค่าอาร์ทบีทและข้อมูลของอากาศยาน ไร้คนขับจาก-

### Mavlink gateway

โดยที่ CMD มีค่า 50 หมายถึง การส่งค่าควบคุมการบินของอากาศยาน ไปยัง Mavlink gateway

โดยที่ CMD มีค่า 200 หมายถึง การได้รับค่าพารามิเตอร์จาก Mavlink gateway

โดยที่ CMD มีค่า 201 หมายถึง การได้ส่งค่าพารามิเตอร์ไปยัง Mavlink gateway

#### 3.5.1 ความสัมพันธ์ของ CMD และ PAYLOAD มีดังต่อไปนี้

CMD 0, 1, 2, 3 ไม่มี PAYLOAD รับและส่งคำสั่งเพียงอย่างเดียว

str[10]	int_8	int_8	int_16	int_16	int_8	CMD = 11 PAYLOAD : Heart Beat 17 BYTE
FLIGHT_MODE	MAV_TYPE	ARM STATUS	Voltage Battery	Current	Percent Battery	

ภาพที่ 3-18 ข้อมูลของความสัมพันธ์ระหว่าง CMD 11 กับ PAYLOAD

int_8	int_16	int_8	int_32	int_32	CMD = 12 PAYLOAD : EXTEND STATUS 12 BYTE
FIX_TYPE	EPH	SATELLITE_VISIBLE	Latitude	Latitude	

ภาพที่ 3-19 ข้อมูลของความสัมพันธ์ระหว่าง CMD 12 กับ PAYLOAD

float	float	float	CMD = 13 PAYLOAD : EXTRA 1 12 BYTE
PITCH	ROLL	YAW	
#30	#30	#30	

ภาพที่ 3-20 ข้อมูลของความสัมพันธ์ระหว่าง CMD 13 กับ PAYLOAD

float	int_16	float	CMD = 14 PAYLOAD : EXTRA 2 10 BYTE
ALTTITUDE	COMPASS	GROUND_SPEED	
#74	#74	#74	

ภาพที่ 3-21 ข้อมูลของความสัมพันธ์ระหว่าง CMD 14 กับ PAYLOAD

int_16 CH_1	int_16 CH_2	int_16 CH_3	int_16 CH_4	int_16 CH_5	int_16 CH_6	int_16 CH_7	int_16 CH_8
CMD = 15 PAYLOAD : RC_CH 16 BYTE							

ภาพที่ 3-22 ข้อมูลของความสัมพันธ์ระหว่าง CMD 15 กับ PAYLOAD

int_32 Throttle	int_32 YAW	int_32 PITCH	int_32 ROLL
CMD = 50 PAYLOAD : RC control 16 BYTE			

ภาพที่ 3-23 ข้อมูลของความสัมพันธ์ระหว่าง CMD 50 กับ PAYLOAD

int_16 Parameter Index	str[15] Parameter name	Float Parameter Value
CMD = 200 PAYLOAD : Receive Parameter 21 BYTE		

ภาพที่ 3-24 แสดงถึงข้อมูลของความสัมพันธ์ระหว่าง CMD 200 กับ PAYLOAD

int_16 Parameter Index	Float Parameter Value
CMD = 201 PAYLOAD : Send Parameter 6 BYTE	

ภาพที่ 3-25 แสดงถึงข้อมูลของความสัมพันธ์ระหว่าง CMD 201 กับ PAYLOAD

3.5.2 ความถี่ของ CMD นั้นขึ้นอยู่กับความถี่ในการเปลี่ยนแปลงของค่าต่าง ๆ ที่อยู่ใน PAYLOAD โดยสามารถจัดกลุ่มได้ดังนี้ CMD 11, CMD 12 จะมีความถี่ของการส่งอยู่ที่ 1 Hz และ CMD 13, CMD 14, CMD 15 จะมีความถี่ในการส่งอยู่ที่ 5 Hz และ CMD 50 มีความถี่สูงที่สุดเนื่องจากเป็นส่วนที่ต้องติดต่อกับอากาศยาน ไร้คนขับอยู่ตลอดเวลา และมีความสำคัญสูงที่สุดเนื่องจากความปลอดภัยของอากาศยาน ไร้คนขับจึงส่งที่ความถี่ 20 Hz ส่วนพารามิเตอร์นี้เป็นส่วนที่มีการรับส่งให้เสร็จก่อนการใช้งานระบบและความถี่ตรงตามกับ Mavlink gateway

### 3.6 การออกแบบการติดตั้งอุปกรณ์

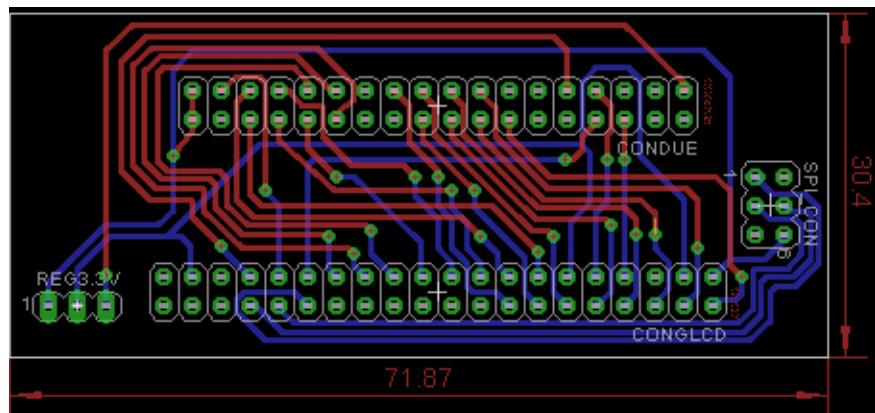
เนื่องจากในสร้างชิ้นงาน Ground Station นั้นประกอบไปด้วย Module และอุปกรณ์อื่น ๆ เป็นจำนวนมาก จึงทำให้ไม่สามารถสร้างชิ้นงานที่มีความเป็นระเบียบเรียบร้อยและแข็งแรงได้ ดังนั้น การสร้าง Module หรือสร้างวัสดุมาครอบ Ground Station นั้นจึงเป็นสิ่งที่จำเป็นต่อการสร้างชิ้นงาน

#### 3.6.1 การสร้าง Module สำหรับส่วนเชื่อมต่อ Ground Station

เหตุสำคัญของการสร้าง Module สำหรับ Ground Station นั้นเป็นไปเพื่อลดความยุ่งยากและเพิ่มความเป็นระเบียบของการเชื่อมต่อ Module ต่าง ๆ เข้ากับ Ground Station โดยที่ Module แบ่งออกเป็น 3 ชิ้นดังนี้

##### 3.6.1.1 Module สำหรับการเชื่อมต่อ GLCD

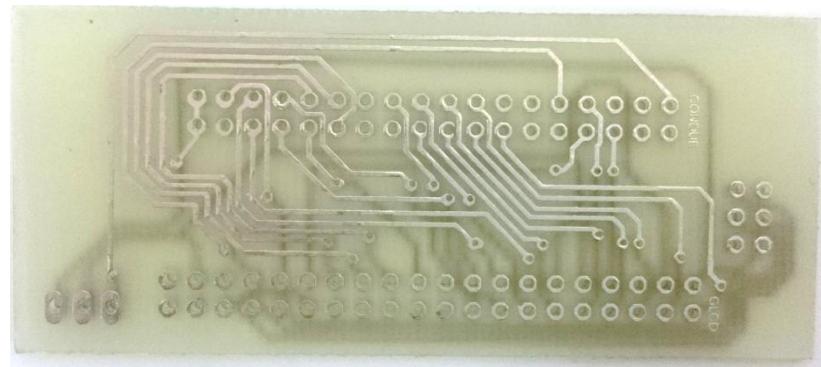
เนื่องจาก Pin layout ระหว่าง Arduino DUE และ Module LB04301 Chipset SSD1963 นั้นแตกต่างกันอย่างสิ้นเชิง จึงได้สร้าง Module นี้ขึ้นมาเพื่อทำการเป็น Pin converter ระหว่าง Arduino DUE และ Module LB04301 เพื่อที่จะสามารถต่อสายแพร์ 2x20 เข้ากับ Module LB04301 โดยโคลนที่ Module ชิ้นนี้เป็น Printed circuit board (PCB) หนึ่งแผ่น โดยมีลักษณะของแผ่น PCB ตามภาพ



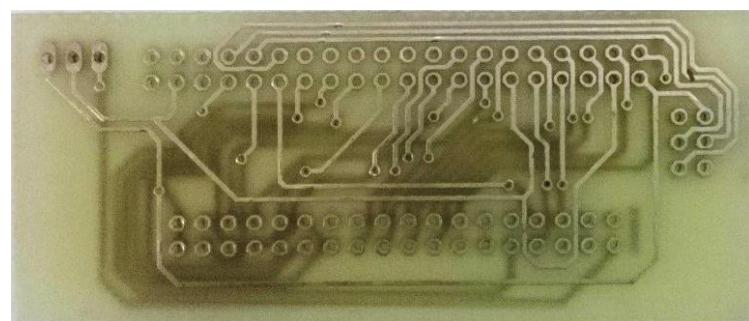
ภาพที่ 3-26 การออกแบบ PCB สำหรับการเชื่อมต่อ Arduino DUE และ GLCD

โดยจากภาพ จะเห็นว่า CONDUE คือ connector สำหรับ Port 18x2 ของ Arduino DUE และ CONGLCD คือส่วนที่เชื่อมต่อ กับ GLCD ด้วยสายแพร์ขนาด 2x20 เพราะฉะนั้นจึงทำการเดินสายทองแดงเป็นสองด้านเพื่อสำหรับการบัดกรี Pin header ในแต่ละด้าน มี Regulator ในการแปลงค่าแรงดันจาก Arduino DUE 5V เพื่อมาใช้เป็นแรงดันสำหรับการเลี้ยงจอที่ใช้ logic level 3.3V และมีพอร์ต SPI สำหรับ SD card บน GLCD ที่รองรับการเชื่อมต่อเข้ากับ

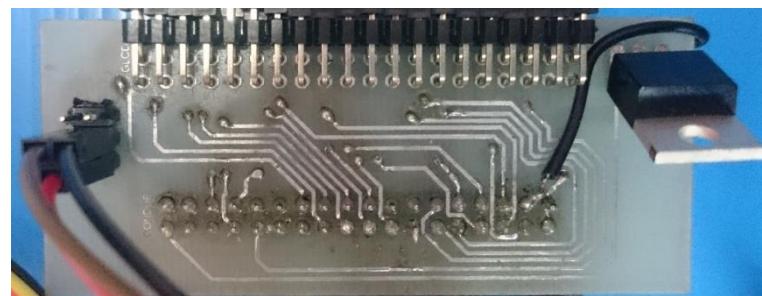
Arduino DUE อีกด้วย โดยที่ความกว้างและยาวของ แผ่น PCB นั้นมีขนาด 7.187x3.04 cm โดยที่ใช้ ความกว้างของลายทองแดงขนาด 15 mill และ ขนาดของ hole 8 mill



ภาพที่ 3-27 PCB Pin converter ระหว่าง Arduino DUE และ GLCD ด้านบน



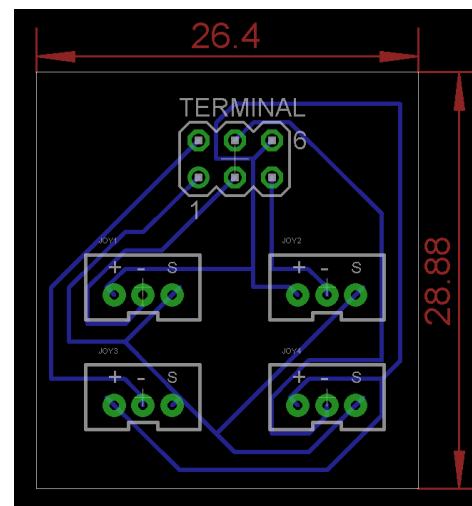
ภาพที่ 3-28 PCB Pin converter ระหว่าง Arduino DUE และ GLCD ด้านล่าง



ภาพที่ 3-29 PCB Pin converter หลังจากบัดกรีสำเร็จ

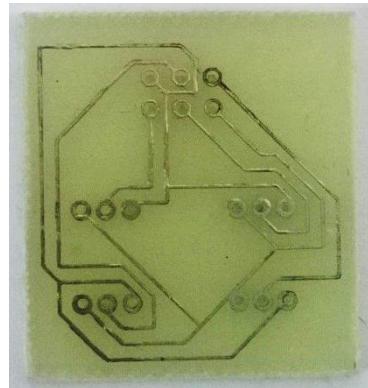
### 3.6.1.2 Module สำหรับการเชื่อมต่อ Joy stick

เนื่องจากในการออกแบบนั้นต้องใช้ Joy stick ทั้งหมด 2 ชิ้น และหนึ่งชิ้นนั้นต้องใช้ไฟเลี้ยงถึง 2 ตัวทำให้ใช้ทั้งหมด 4 ตัว ซึ่งเป็นการสืบเปลี่ยนหากทำการเชื่อมต่อโดยตรงและ Arduino DUE ไม่มีไฟเลี้ยง 3.3 v ถึง 4 จุด ดังนั้นจึงออกแบบ Module joy stick นี้เพื่อทำการรวมสายสัญญาณและสายไฟทุกเส้นของ Joy stick เพื่อให้ง่ายต่อการต่อพ่วง และทำให้มีความเป็นระเบียบมากขึ้น โดยจัดทำ PCB ที่ออกแบบดังนี้



ภาพที่ 3-30 Module joy stick

จากภาพเป็น PCB 1 ด้านที่มี slot สำหรับการใช้งาน Joy stick 4 slot และมี Terminal ขนาด 3x2 สำหรับการเชื่อมต่อเข้ากับ Arduino DUE โดยที่มีสัญญาณ 4 เส้น ได้แก่ สัญญาณจาก Joy stick ไฟเลี้ยง และ กราวด์ โดยที่ความกว้างและยาวของแผ่น PCB นั้นมีขนาด  $2.64 \times 2.88$  cm โดยที่ใช้ความกว้างของลายทองแดงขนาด 15 mill และขนาดของ hole 8 mill

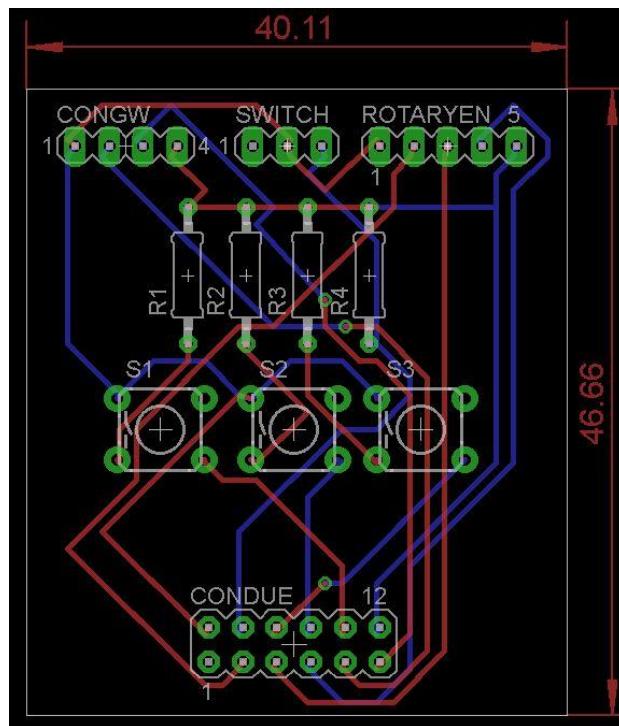


ภาพที่ 3-31 PCB Module joy stick

โดยที่ผลการทดลองของการใช้ Module joystick อยู่ในบทที่ 4 Pin mapping

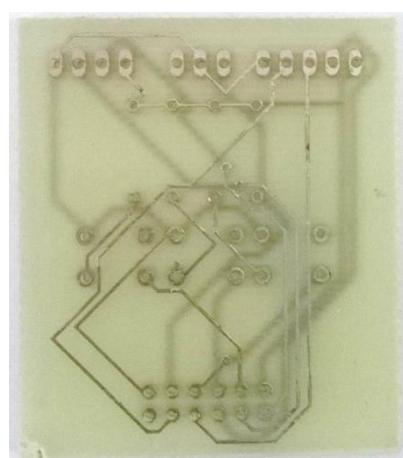
### 3.6.1.3 Module สำหรับการเชื่อมต่อ Push button และ Terminal ต่าง ๆ

เนื่องจาก Ground Station นั้นมี Push button ที่จำเป็นสำหรับการใช้งาน จำนวนสามชิ้น และยังมี Rotary encoder และ ON/OFF switch เป็นส่วนประกอบ และยังมีการสื่อสารแบบ Serial จาก Mavlink gateway อีกด้วย จึงได้จัดทำ Module ชิ้นนี้ขึ้นเพื่อรวมปุ่มกด และ terminal ต่าง ๆ เข้าด้วยกันเพื่อแก้ปัญหาจำนวนไฟเลี้ยงของ Arduino DUE ไม่เพียงพอต่อการใช้งานจึงออกแบบ PCB ได้ดังนี้

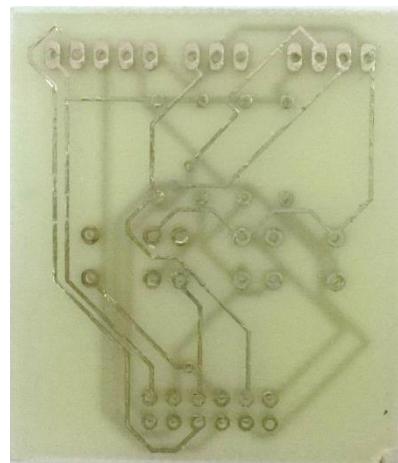


ภาพที่ 3-32 PCB Push button & terminal

โดยที่การออกแบบ PCB นี้ประกอบด้วยปุ่มแบบกดติดปล่อยขึ้นจำนวน 3 ชิ้น และมี Terminal สำหรับ Switch สองทางจำนวนหนึ่งชิ้น Terminal สำหรับการเชื่อมต่อ กับ Mavlink gateway และ Terminal สำหรับการเชื่อมต่อ Rotary encoder และมี Output ขนาด 6x2 ไปยัง Arduino DUE โดยที่ใช้ความกว้างของลายทองแดงขนาด 15 mill และขนาดของ hole 8 mill



ภาพที่ 3-33 PCB Push button & terminal ด้านบน

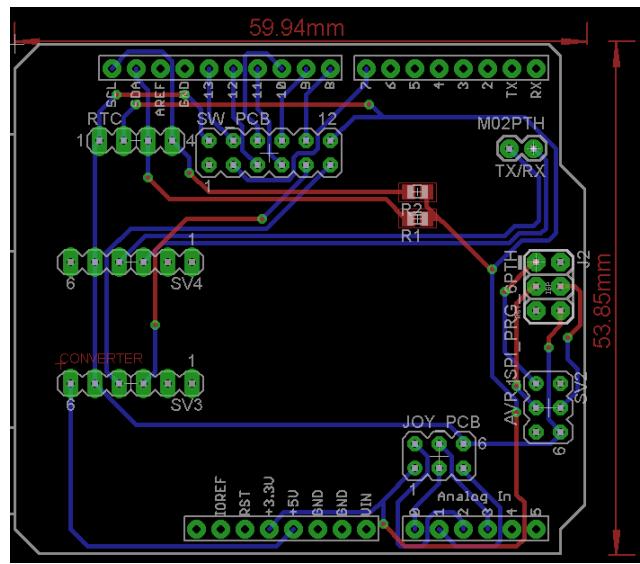


ภาพที่ 3-34 PCB Push button & terminal ด้านล่าง



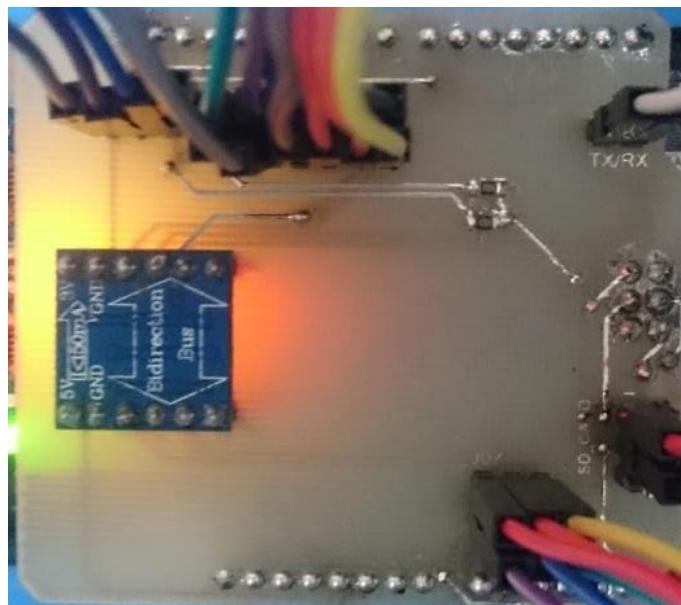
ภาพที่ 3-35 PCB Push button & terminal หลังจากบัดกรีสำเร็จ

3.6.1.4 Module สำหรับสำหรับการเชื่อม Module อื่น ๆ รวมกับ Arduino DUE เนื่องจากในการออกแบบ PCB สำหรับการเชื่อมต่ออุปกรณ์ต่าง ๆ เช่นกับแต่ละ PCB Module นั้นยังคงไม่สามารถจัดการให้สายไฟให้ทำการเชื่อมต่อเข้ากับ Arduino DUE ได้อย่างเป็นระเบียบจึงได้ทำการออกแบบ Terminal สำหรับทุก Module มาเพื่อเชื่อมต่อ กับ Main Module นี้ และในการเชื่อมต่อระหว่าง Portable Ground Station และ Mavlink Gateway นั้นคือการเชื่อมต่อระหว่าง Arduino DUE และ Arduino Pro Mini 16MHz 5V นั้นไม่สามารถทำได้โดยตรงจึงได้ทำการเพิ่งอุปกรณ์ Voltage Converter เช่นไปยัง Module นี้ด้วย โดยที่ Main Module นี้ได้ออกแบบจาก Arduino Uno Shield ดังภาพด้านล่าง



ภาพที่ 3-36 Main PCB

โดยที่ PCB ชิ้นนี้ยังมีพอร์ตสำหรับการเชื่อมต่อ SPI ที่มาระหว่าง PCB Pin converter อีกด้วย โดยที่ใช้ความกว้างของลายทองแดงขนาด 15 mm และขนาดของ hole 8 mm



ภาพที่ 3-37 Main PCB หลังบัดกรีสำเร็จ

## บทที่ 4

### ผลการดำเนินงาน

หลังจากผ่านขั้นตอนการออกแบบของ การสร้างระบบควบคุมและแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยาน ໄร์คันขั้นแบบ 4 ใบพัดเสริจสีน កีได้มานำเสนอขั้นตอนการดำเนินงาน ดังนี้ บทนี้จะพูดถึงผลการดำเนินงานทั้งผลของการสร้างชิ้นงาน โดยแบ่งเป็นส่วนย่อย ๆ ดังนี้

- 4.1 การทดสอบฮาร์ดแวร์ที่นำมาใช้
- 4.2 การทดสอบการรับส่งข้อมูลระหว่างส่วนแสดงผลและแม่พิมพ์เกตเวย์
- 4.3 การทดสอบระบบจริง
- 4.4 ผลการทดสอบประสิทธิภาพของการแบ่งเวลาในการดำเนินงานของแต่ละ Task
- 4.5 การทดสอบอัตราการสีนเปลี่ยนแปลงพลังงาน

#### 4.1 การทดสอบฮาร์ดแวร์ที่นำมาใช้งาน

หลังจากที่ทำการเลือกอุปกรณ์สำหรับการสร้างชิ้นงานไปแล้วนั้น ขั้นตอนต่อไปจึงเป็นขั้นตอนของการทดสอบฮาร์ดแวร์ที่นำมาใช้งานกับระบบเพื่อความสามารถและข้อจำกัดของตัวฮาร์ดแวร์ในแต่ละชิ้นเพื่อคุ้ว่าเพียงพอต่อการใช้งานของระบบหรือไม่

##### 4.1.1 การทดสอบการใช้งาน จอ TFT 3.2" 400x320 pixel, Chipset SSD1963

ในการทดสอบการใช้งานจนนี้เราจะทำการวัดประสิทธิภาพของจอเพื่อสังเกตว่า ความเร็วในการอัปเดตข้อมูลของจอนั้นสูงที่สุดใช้เวลาในการดำเนินการเท่าไหร่ operation time ของ Chipset และ operation sequential ของ Chipset SD 1963 ทำงานอย่างไร โดยที่มีอุปกรณ์และขั้นตอนการดำเนินงานดังนี้

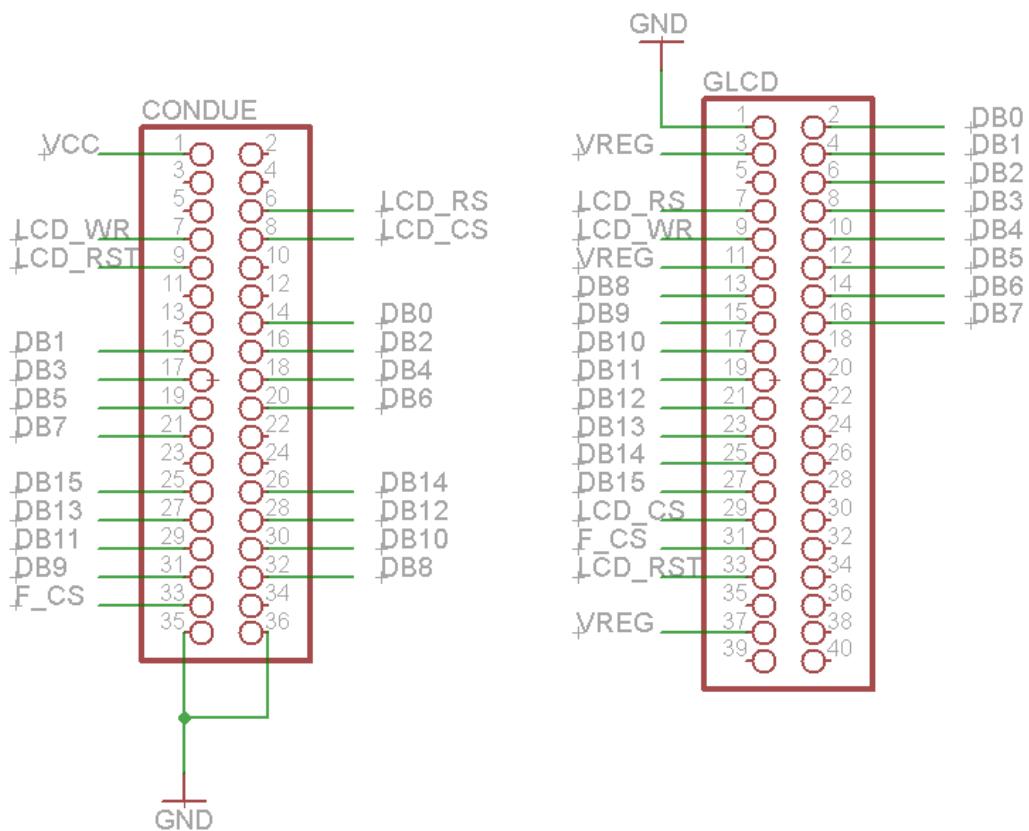
##### อุปกรณ์ที่ใช้ในการทดสอบ จอ TFT 3.2" 400x320 pixel, Chipset SSD1963

- |  |              |
|--|--------------|
| - TFT 3.2" 400x320 pixel , Chipset SSD1963 | จำนวน 1 ชิ้น |
| - บอร์ด Arduino DUE                        | จำนวน 1 ชิ้น |
| - LCD TFT Adapter                          | จำนวน 1 ชิ้น |
| - Oscilloscope                             | จำนวน 1 ชิ้น |

##### ซอฟต์แวร์ที่ใช้ในการทดสอบ จอ TFT 3.2" 400x320 pixel, Chipset SSD1963

- OS Windows 7 64-bit Service pack I

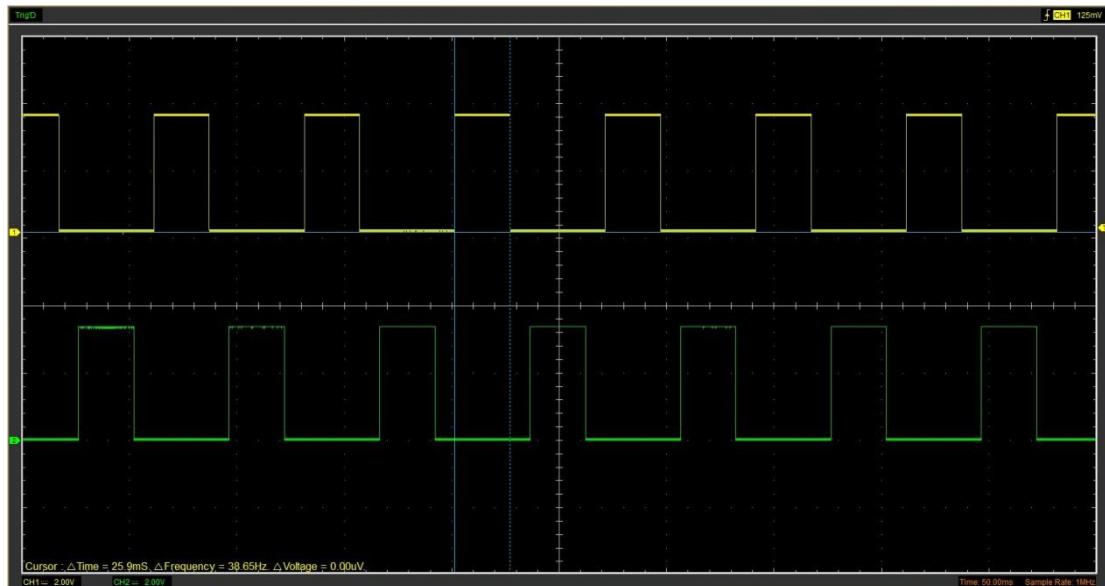
- Arduino IDE version 1.5.7
- Library UTFT version 2.7 (<http://www.rinkydinkelectronics.com/library.php?id=51>)
- ในส่วนของขั้นตอนการดำเนินงานนี้คือเริ่มจากทำการเชื่อมต่อวงจรของ TFT 3.2"
- 400x320 pixel , Chipset SSD1963 เข้ากับ Arduino DUE โดยมีผังการเชื่อมต่อวงจรดังนี้



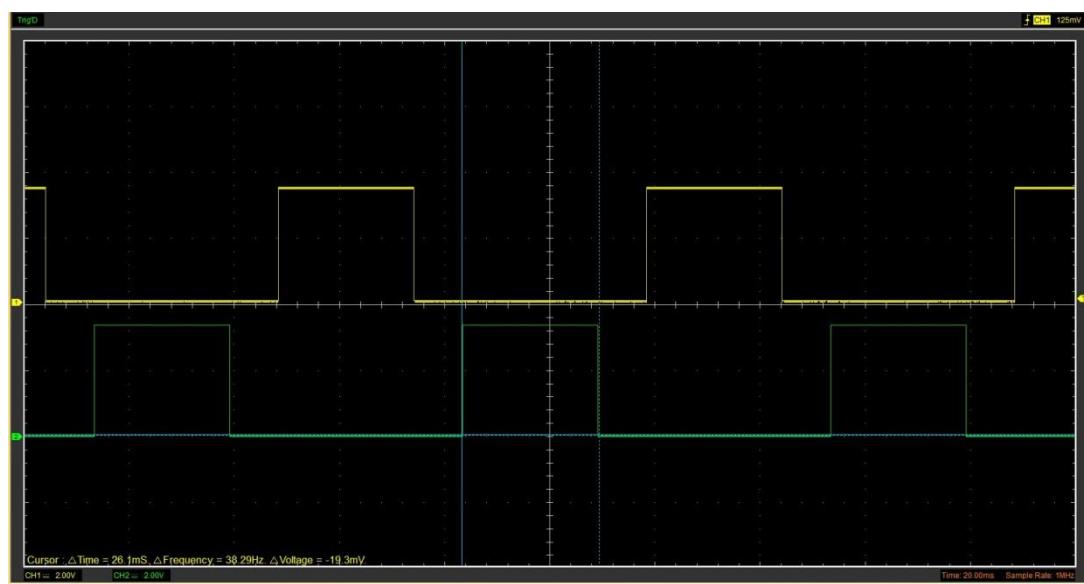
ภาพที่ 4-1 แสดงถึงการเชื่อมต่อระหว่าง GLCD Module และ Arduino DUE

หลังจากทำการเชื่อมต่อวงจรสำเร็จ ทำการเขียนโค้ดเพื่อทดสอบประสิทธิภาพของจอยโดยทำการเขียนโค้ด เพื่อ Clear screen และการ Fill color ไปยังหน้าจอโดยที่ใช้เวลาห่างกันครั้งละ 10 ms จากนั้นทำการเปิดพอร์ต I/O เพื่อการวัดความเร็วการทำงานจากนั้นใช้ Oscilloscope ในการวัดค่าสัญญาณของ TFT 3.2" 400x320 pixel , Chipset SSD1963 บน Arduino DUE และทำการสังเกต Sequential operation ของการทำงานคำสั่งต่าง ๆ ของ Module GLCD

โดยที่ผลการทดลองเป็นไปดังนี้



ภาพที่ 4-2 แสดงถึงเวลาที่ใช้ในการ Clear Screen



ภาพที่ 4-3 แสดงถึงเวลาที่ใช้ในการ Fill Color ทั้งหน้าจอ

จากการทดลองพบว่าเวลาที่ใช้ในการวาดหน้าจอใช้เวลาประมาณ 25 ms

หลังจากนั้น ได้ทำการทดลองจับค่าสัญญาณทั้ง 4 เส้นของ GLCD โนมูล ได้แก่ CS, WR, RD และ RS ตามลำดับซึ่ง ได้ทดลองและสังเกตพฤติกรรมจากคำสั่ง SetPIXEL หรือทำการเขียนข้อมูลไปยังหน้าจอ GLCD นั้นเอง โดยที่ สัญญาณ WR จะทำหน้าที่ในการเขียนข้อมูล โดยที่ WR จะเป็น LOW เมื่อ CS เป็น LOW , RD สำหรับการอ่านข้อมูล, CS คือ Chip Select นั้นเอง



ภาพที่ 4-4 ตัวอย่างการจับค่าสัญญาณของคำสั่ง SetPIXEL

โดยที่จากภาพที่ 4-3 D0 คือ RS, D1 คือ RD, D2 คือ WR, D3 คือ CS และ D4 คือ toggle I/O ตามลำดับ

#### 4.1.2 การทดสอบการใช้งาน Module SD card

ในการทดสอบการใช้งาน Module SD card เป็นไปเพื่อแสดงถึงประสิทธิภาพของ Module SD card ในการเขียนข้อมูลลงไปยัง SD Card และแสดงถึงความแม่นยำของการเขียนและการอ่านข้อมูลไปยัง SD Card โดยที่มีอุปกรณ์และขั้นตอนการดำเนินงานดังนี้

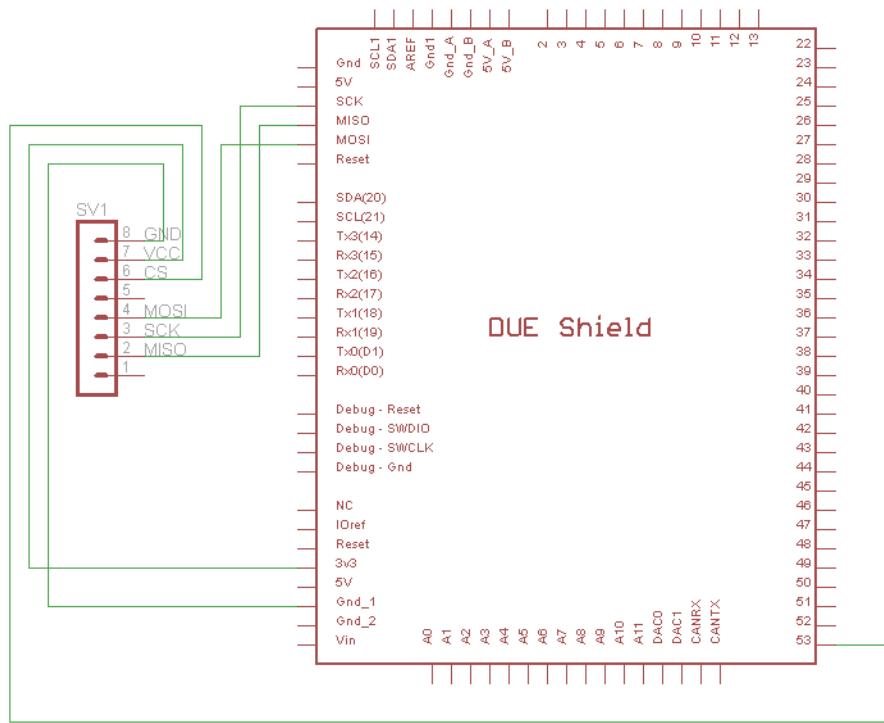
อุปกรณ์ที่ใช้ในการทดสอบ Module SD card

- SD Card Module จำนวน 1 ชิ้น
- Arduino DUE จำนวน 1 ชิ้น
- SD Card KINGMAX ขนาด 4 GB เป็น File System แบบ FAT32 Class 6 จำนวน 1 ชิ้น

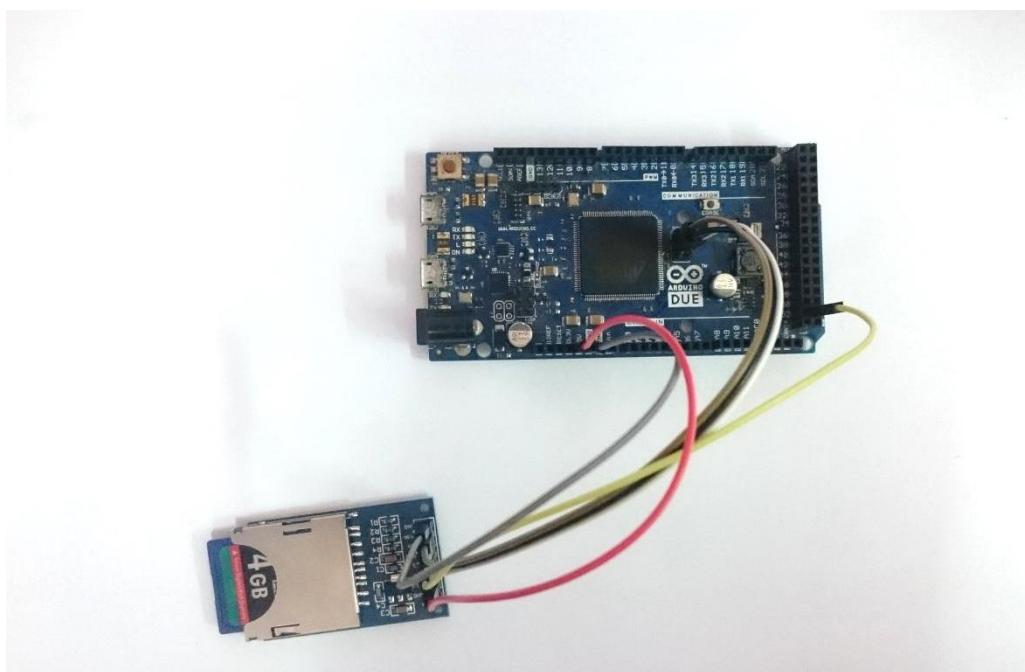
### ซอฟต์แวร์ที่ใช้ในการทดสอบ Module SD card

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7
- Library SD version 1.0 (<http://www.arduino.cc/en/Reference/SD>)

ในส่วนของการดำเนินงานนี้เริ่มจากทำการเชื่อมต่อ SD Card Module กับ Arduino DUE ที่พอร์ต SPI โดยมีผังการเชื่อมต่อของดังนี้



ภาพที่ 4-5 แสดง Schematic การเชื่อมต่อระหว่าง SD card module และ Arduino DUE

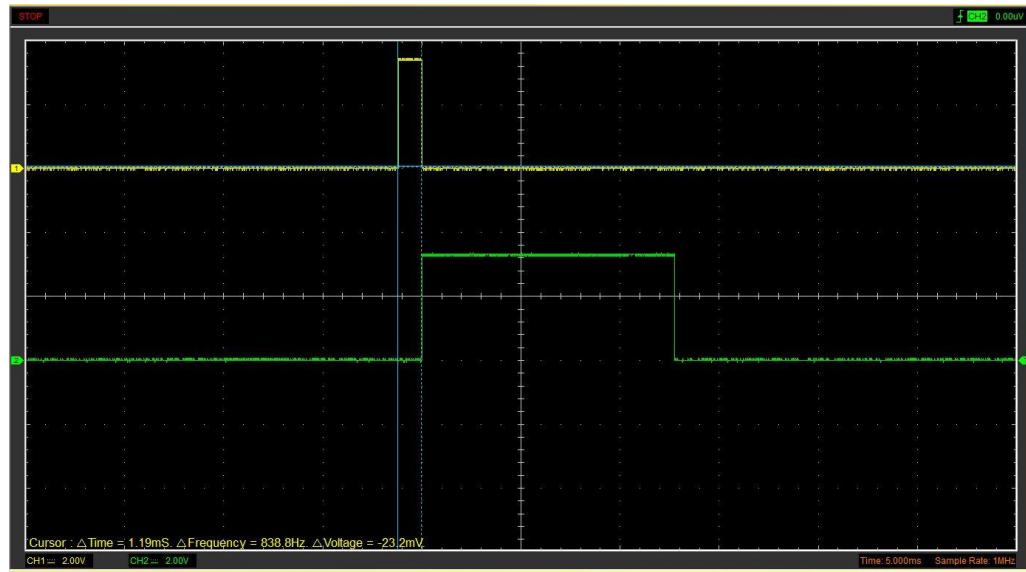


**ภาพที่ 4-6** แสดงถึงการเชื่อมต่อระหว่าง Arduino DUE และ SD Card module

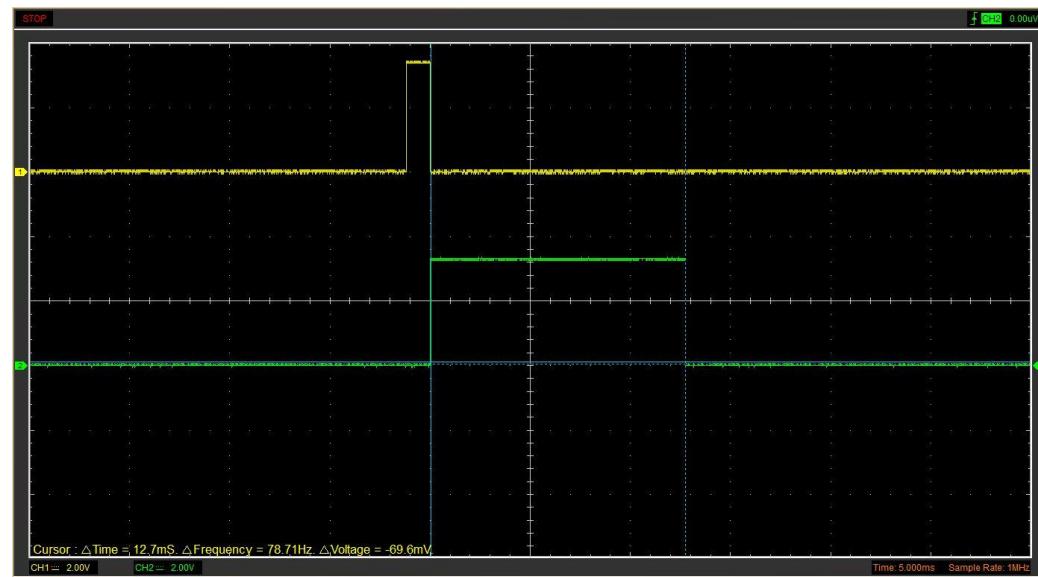
ขั้นตอนของการเขียนโปรแกรมเพื่อทำการทดสอบประสิทธิภาพของ SD Card Module ร่วมกับ Arduino DUE โดยการทดสอบแบ่งเป็นสามกรณี ได้แก่ การเขียนข้อมูลลง SD Card การอ่านข้อมูลจาก SD Card และการเปิดปิดไฟล์ โดยที่การทดสอบแรกนั้นทำการเขียน Code เพื่อเขียนข้อมูลลง SD Card จำนวน 1 byte จากนั้นทำการ Flush ข้อมูลลง SD Card และทำการจับเวลาที่ใช้เขียนและ Flush ข้อมูลจากนั้นทำการเพิ่มข้อมูลที่เขียนเป็น 100 Byte และ 1,000 Byte ตามลำดับ การทดลองที่สองคือ ทดลองการเขียนข้อมูล 1,000 byte โดยที่เขียนข้อมูลครึ่งละ 1 byte 1,000 ครั้ง และเขียนข้อมูล 10 byte จำนวน 100 ครั้ง เพื่อหาความแตกต่างเวลาที่ใช้ในการเขียนและ Flush ข้อมูลลง SD card การทดลองเพื่อวัดประสิทธิภาพของการอ่านข้อมูลนั้นทำได้โดยอ่านข้อมูลจาก SD card ขนาด 1 byte 1 ครั้ง และขนาด 1 byte จำนวน 1,000 ครั้งเพื่อทดสอบความเสถียรของการเขียนข้อมูล จากนั้นในการทดสอบการเปิดไฟล์ไฟล์ได้ทำการเขียนเปิดและปิดไฟล์และทำการวัดเวลาในการเปิด-ปิด

โดยที่ผลการทดลองเป็นไปตามนี้

การทดลองการเขียนข้อมูลเพื่อหาเวลาในการเขียนและ Flush ข้อมูลลง SD Card



ภาพที่ 4-7 แสดงถึงเวลาที่ใช้ในการเขียนข้อมูลขนาด 1 byte ไปยัง SD card โดยที่ใช้วลามในการเขียน 1.19 ms



ภาพที่ 4-8 แสดงถึงเวลาที่ใช้ในการ Flush ข้อมูลขนาด 1 byte ไปยัง SD card โดยที่ใช้วลามในการ Flush 12.7 ms

ผลการทดลองของการใช้งาน SD Card สามารถจัดทำเป็นตาราง ได้ดังนี้

ตารางที่ 4-1 แสดงการทดสอบการเขียนข้อมูลไปยัง SD Card ด้วยขนาดข้อมูลต่าง ๆ

Size of data	Write time (ms)	Flush Time (ms)
1 – byte	1.19	12.7
100 – bytes	1.22	12.7
1000 - bytes	14.7	12.7

ตารางที่ 4-2 แสดงผลการทดสอบการเขียนข้อมูลขนาด 1000 byte โดยการเขียนข้อมูลแต่ละครั้งมีขนาดไม่เท่ากัน

Data per each write	Write time (ms)	Average time per byte (ms)	Flush Time (ms)
1 – byte	50.3	5.03	12.7
10 – bytes	14.2	1.42	12.7
1000 - bytes	14.7	1.47	12.7

ตารางที่ 4-3 แสดงผลการทดสอบการเขียนข้อมูลไปยัง SD Card ด้วยขนาดข้อมูลที่ไม่เท่ากัน

Read data	Read time (ms)
1 – byte	7 us
1000 – bytes	7 ms

ตารางที่ 4-4 แสดงผลการทดสอบเวลาที่ใช้ในการเปิดและปิดไฟล์จาก SD Card

Operation	Operation time (us)
Open file	57.7
Close file	1.5

จากการทดลองการใช้งาน SD Card สรุปได้ว่า

- ในการเขียนข้อมูลนั้นมีเวลาแพร่ผันต่างกับจำนวนข้อมูลต่อการเขียน 1 ครั้ง
- การ Flush ข้อมูลนั้นใช้เวลาคงที่เสมอ
- ในการเขียนข้อมูล 1 ครั้งนั้นมี Overhead ต่อการเขียนเสมอ เพราะฉะนั้นไม่ควรเขียนไฟล์ที่มีความยาวน้อยเกินไป เพราะจะสิ้นเปลืองเวลา Overhead
- การอ่านข้อมูลจาก SD Card นั้น ใช้เวลา 7 us ต่อ 1 byte
- การเปิดไฟล์ใช้เวลา ประมาณ 60 us และการปิดไฟล์ใช้เวลา 1.5 us

#### 4.1.3 การทดสอบการใช้งาน Rotary encoder

ในการทดลองการใช้งานนี้เป็นไปเพื่อทดสอบและทดลองการใช้งาน Arduino DUE ร่วมกับ Rotary encoder เพื่อทำการทดลองการใช้งานและสังเกตการทำงานของอุปกรณ์ โดยที่มีอุปกรณ์และขั้นตอนการดำเนินงานดังนี้

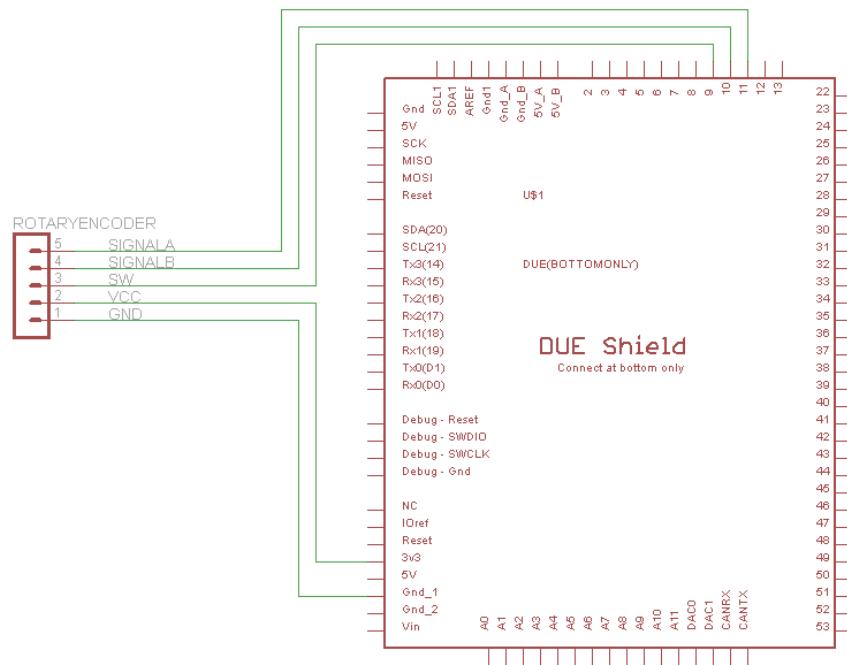
อุปกรณ์ที่ใช้ในการทดสอบ Rotary encoder

- Rotary encoder จำนวน 1 ชิ้น
- Arduino DUE จำนวน 1 ชิ้น

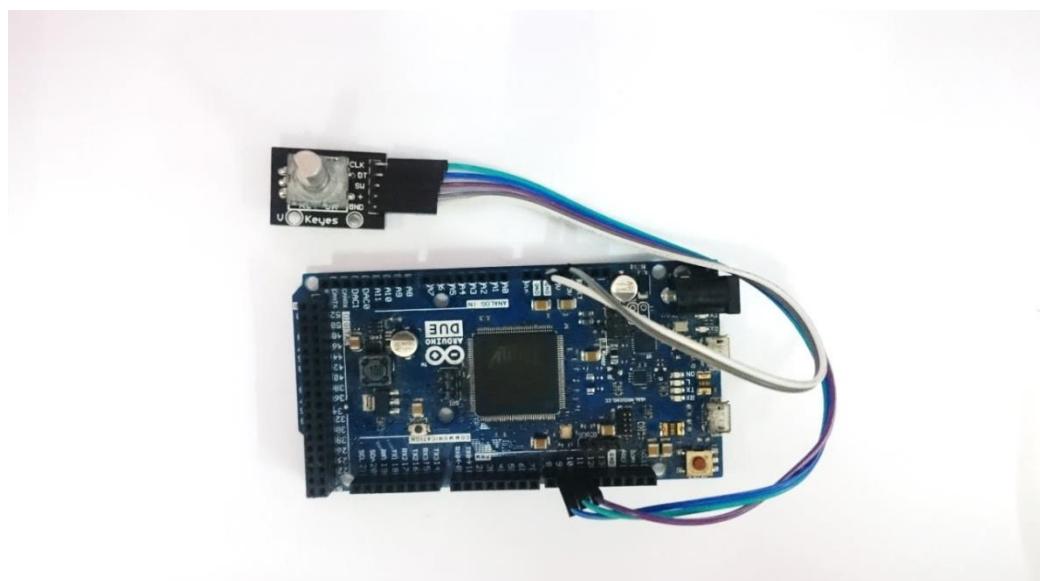
ซอฟต์แวร์ที่ใช้ในการทดสอบ Rotary encoder

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7
- Library Encode version 1.2 ([http://www.pjrc.com/teensy/td\\_libs\\_Encoder.html](http://www.pjrc.com/teensy/td_libs_Encoder.html))

ในส่วนของการดำเนินงานนี้เริ่มจากการเชื่อมต่อ Rotary encoder เข้ากับ Arduino DUE ที่พอร์ต Digital I/O ดังผังวงจรต่อไปนี้



ภาพที่ 4-9 ผังการเชื่อมต่อวงจรระหว่าง Arduino DUE และ Rotary Encoder

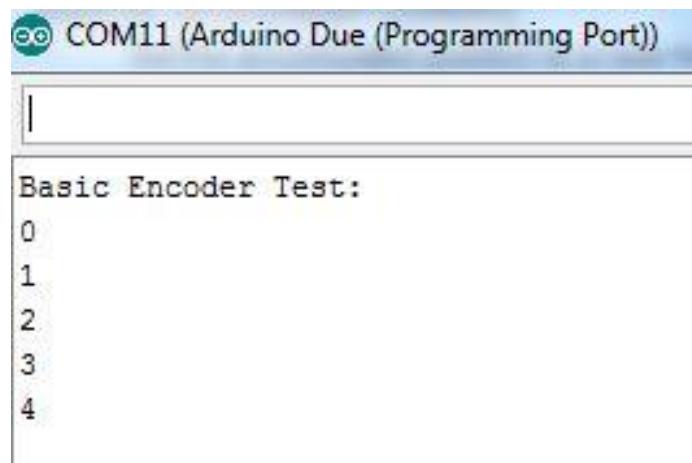


ภาพที่ 4-10 แสดงถึงการเชื่อมต่อระหว่าง Arduino DUE และ Rotary Encoder

ขั้นตอนของการเขียนโปรแกรมเพื่อทำการทดสอบ Rotary encoder นั้นคือการใช้ขา interrupt บน Arduino DUE เพื่อทำการนับค่าแบบ Incremental สำหรับการหมุนทั้งสองทิศทางของ

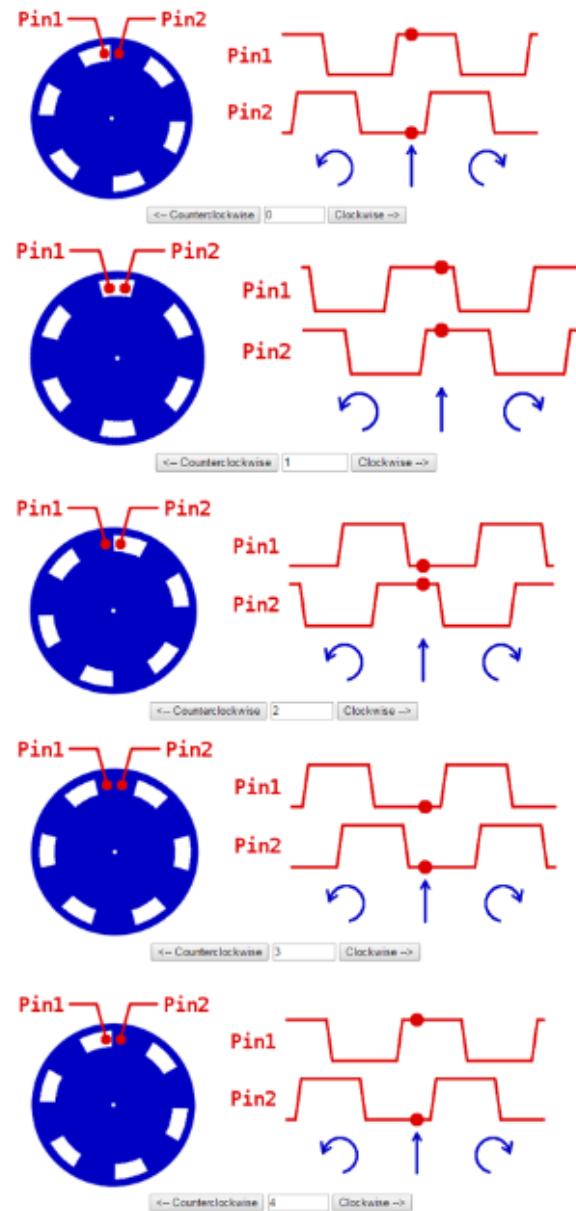
Rotary encoder แล้วทำการแสดงค่าผ่าน Serial ที่ใช้ Buadrate 9600 และทำการใช้ Oscilloscope สังเกตการทำงานของของ Rotary encoder

ผลการทดลองการใช้งาน Rotary encoder ร่วมกับ Arduino DUE



```
Basic Encoder Test:  
0  
1  
2  
3  
4
```

ภาพที่ 4-11 แสดงถึงการสร้าง Pulse หนึ่งลูกสัญญาณและ Arduino สามารถจับค่าสัญญาณได้ทั้งหมดสี่ครั้ง

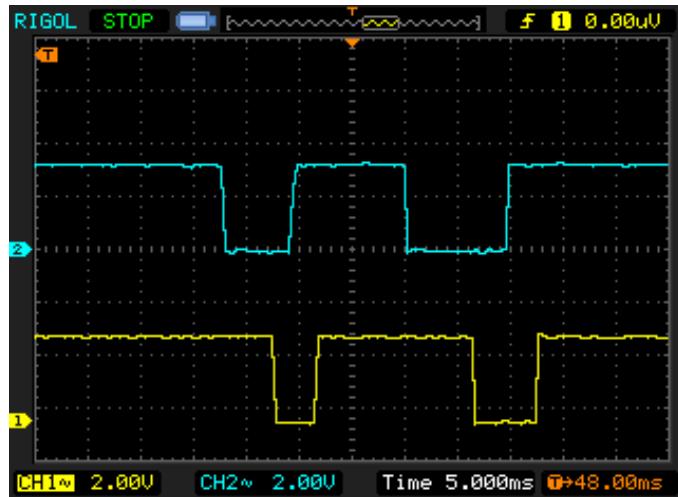


ภาพที่ 4-12 แสดงถึงจังหวะการนับสัญญาณด้วย Library Encoder.h

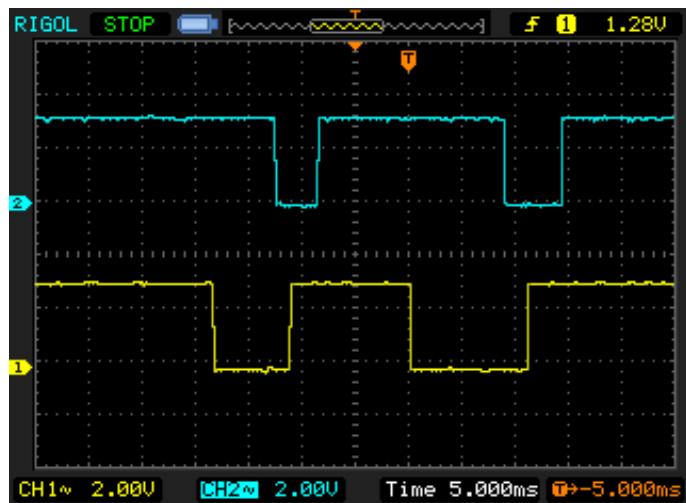
([http://www.pjrc.com/teensy/td\\_libs\\_Encoder.html](http://www.pjrc.com/teensy/td_libs_Encoder.html))

จาก Serial monitor และภาพการทดลองสามารถอ่านข้อมูลได้ว่าการจับสัญญาณของ Arduino DUE ในการทำงานร่วมกับ Rotary encoder เป็นไปอย่างถูกต้อง โดยสัญญาณ 1 pulse จะสามารถตรวจสอบความเปลี่ยนแปลงได้ทั้งหมด 4 จังหวะ

ผลการทดลองการจับสัญญาณ Rotary encoder ด้วย Oscilloscope



ภาพที่ 4-13 แสดงถึงการหมุน Rotary Encoder แบบตามเข็มนาฬิกา



ภาพที่ 4-14 แสดงถึงการหมุน Rotary Encoder แบบวนเข็มนาฬิกา

จากภาพที่ 4-13 และภาพที่ 4-14 แสดงถึงค่าของทิศทางของ Signal A (แทนด้วย Oscilloscope CH1) และ Signal B (แทนด้วย Oscilloscope CH2) โดยที่ภาพที่ 4-13 เป็นการหมุนในทิศทางตามเข็มนาฬิกา เพราะเนื่องจากสัญญาณ Signal A ได้มาก่อน Signal B และ ภาพที่ 4-14 เป็น

การหมุนในทิศทางทวนเข็มนาฬิกา เพราะเมื่อong จากสัญญาณ Signal B ได้มาก่อน Signal A ซึ่งเป็นไปตามทฤษฎีที่เกี่ยวข้องจากบทที่ 2.9

#### สรุปผลการทดลองการใช้งาน Rotary Encoder

- Library Encoder.h สามารถใช้งานร่วมกับ Arduino DUE ได้และสามารถทำงานได้อย่างถูกต้อง
- การทำงานของ Rotary encoder เป็นไปตามทฤษฎีที่อ้างถึงจากบทที่ 2.9

#### 4.1.4 การทดสอบและใช้งาน Real time clock เบอร์ DS3231

ส่วนของการทดสอบการใช้งาน Real time clock นั้นเพื่อการใช้งานและตั้งเวลาเริ่มต้นให้แก่ Real time clock โดยที่มีอุปกรณ์และขั้นตอนการดำเนินงานดังนี้

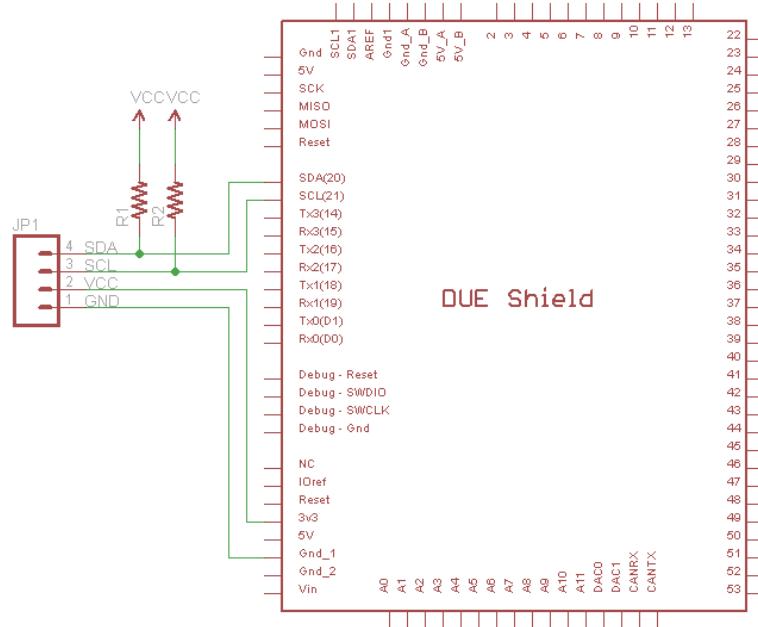
##### อุปกรณ์ที่ใช้ในการทดสอบ Real time clock เบอร์ DS3231

- |                                |              |
|--------------------------------|--------------|
| - Real time clock เบอร์ DS3231 | จำนวน 1 ชิ้น |
| - Arduino DUE                  | จำนวน 1 ชิ้น |
| - ตัวต้านทานขนาด 10 K          | จำนวน 2 ตัว  |

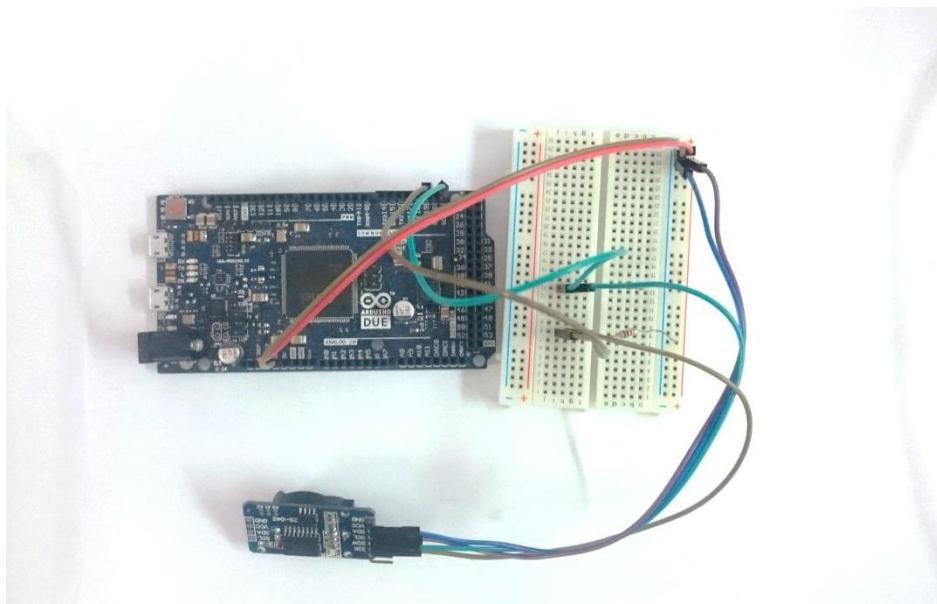
##### ซอฟต์แวร์ที่ใช้ในการทดสอบ Real time clock เบอร์ DS3231

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7
- Library DS3232RTC ([http://www.pjrc.com/teensy/td\\_libs\\_DS1307RTC.html](http://www.pjrc.com/teensy/td_libs_DS1307RTC.html))

ในส่วนของการทดลองนั้นเริ่มจากการตั้งค่าวันเดือนปีและเวลาให้แก่ Real time clock จากนั้นทำการแสดงค่าวันเวลาปัจจุบันผ่านทาง Serial monitor โดยที่ต่อวงจรเข้ากับ Arduino DUE และใช้ Resistor pull-up ขนาด 10 k ดังภาพที่ 4-15

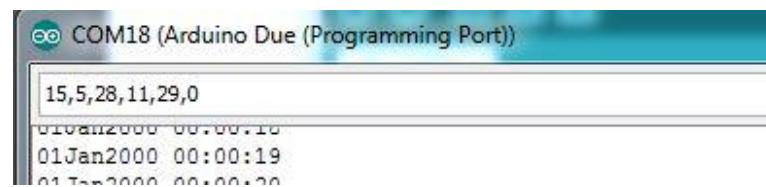


ภาพที่ 4-15 แสดงผังการเชื่อมต่อระหว่าง Arduino DUE และ Real Time Clock



ภาพที่ 4-16 แสดงการเชื่อมต่อระหว่าง Arduino DUE และ Real Time Clock

### ผลการทดสอบการใช้งาน Real time clock



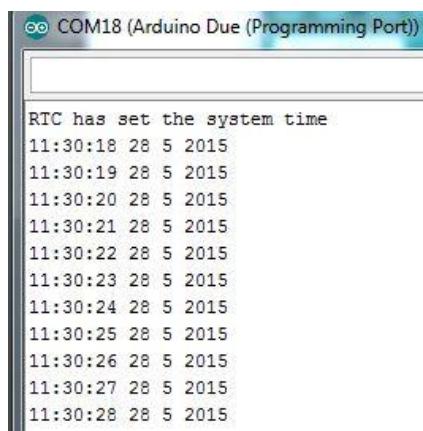
**ภาพที่ 4-17** แสดงถึงการตั้งค่าให้แก่ Real time clock ด้วย Arduino DUE

จากภาพที่ 417 เป็นการตั้งค่า ปี,เดือน,วัน,ชั่วโมง,นาที และวินาที ตามลำดับ ให้แก่ Real time clock module เพื่อเป็นการนำไปใช้งานในครั้งต่อไป

```
01Jan2000 00:00:56
RTCA set to: 28May2015 11:29:00
28May2015 11:29:00 25.50 C 77.90 F
28May2015 11:29:01
28May2015 11:29:02
28May2015 11:29:03
```

**ภาพที่ 4-18** แสดงถึงการตั้งค่า Real time clock ด้วย Arduino DUE สำเร็จ

หลังจากทำการตั้งค่าให้แก่ Real time clock ได้สำเร็จ Serial monitor จะแสดงข้อมูลกดลับมาดังภาพที่ 4-18 และหลังจากนั้นทำการเขียนโปรแกรมเพื่อทดสอบการอ่านค่าเวลาของ Real time clock โดยทำการนำ Arduino DUE ไปอ่านค่าเวลาจาก Real time clock และทำการแสดงผลด้วย Serial monitor ทุก ๆ 1 วินาที ได้ผลการทดลองดังภาพ**4-19**



**ภาพที่ 4-19** แสดงถึงการอ่านค่า Real time clock ด้วย Arduino DUE และแสดงค่าทุกๆ 1 วินาที

## สรุปผลการทดลอง Real time clock เบอร์ DS3231

- Arduino DUE สามารถทำงานร่วมกับ Library DS3232RTC ได้และสามารถทำงานได้อย่างถูกต้อง
- สามารถมาทำตั้งค่าเวลาให้แก่ Real time clock ได้โดยทำงานรันโปรแกรม configuration file และทำการป้อนค่า ปี,เดือน,วัน,ชั่วโมง,นาที และวินาที ตามลำดับ ให้แก่ Real time clock module เพื่อเป็นการตั้งค่าเวลาจริงให้แก่ Real time clock

### 4.1.5 การทดสอบการใช้งานปุ่มกดแบบ Interrupt บน Arduino DUE

ในส่วนของการทดสอบปุ่มกดนี้ ได้ทำการทดลองโดยใช้งาน Internal interrupt บน Arduino DUE เพื่อทดสอบว่าระบบ Internal interrupt และปุ่มกดสามารถใช้ร่วมกันงาน ได้จริงและวัด Responding Time ของปุ่มกด โดยที่อุปกรณ์และการดำเนินงานดังนี้

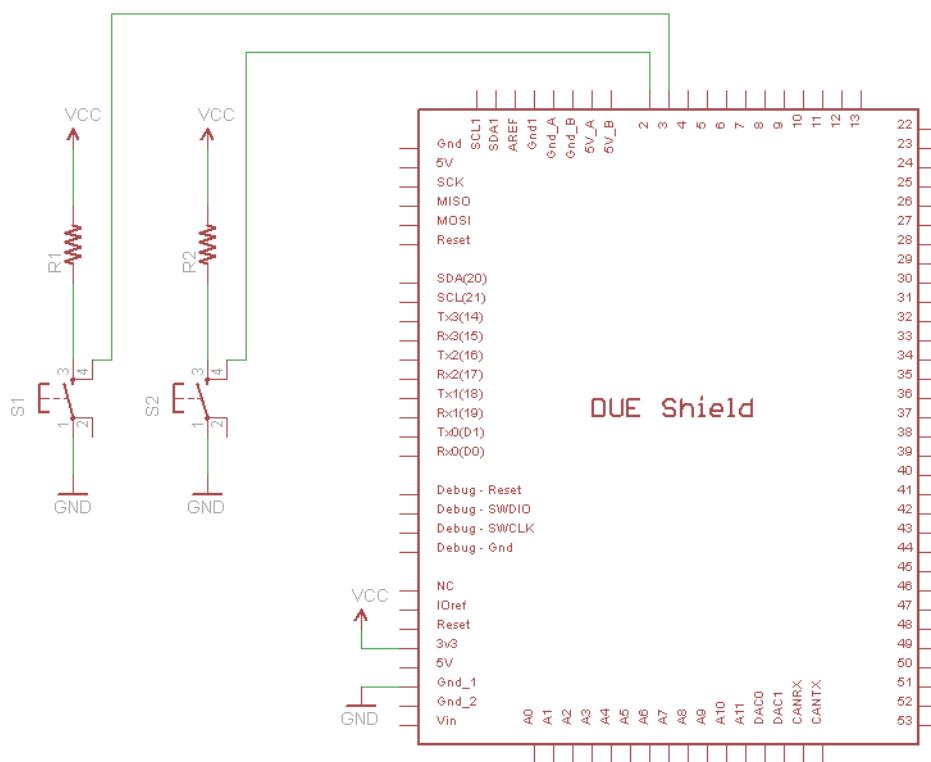
อุปกรณ์ที่ใช้ในการทดสอบ ปุ่มกดและ Internal interrupt บน Arduino DUE

- Push button แบบกดติดปล่อยค้าง จำนวน 2 ชิ้น
- ตัวต้านทาน ขนาด 10K จำนวน 2 ชิ้น
- Arduino DUE จำนวน 1 ชิ้น

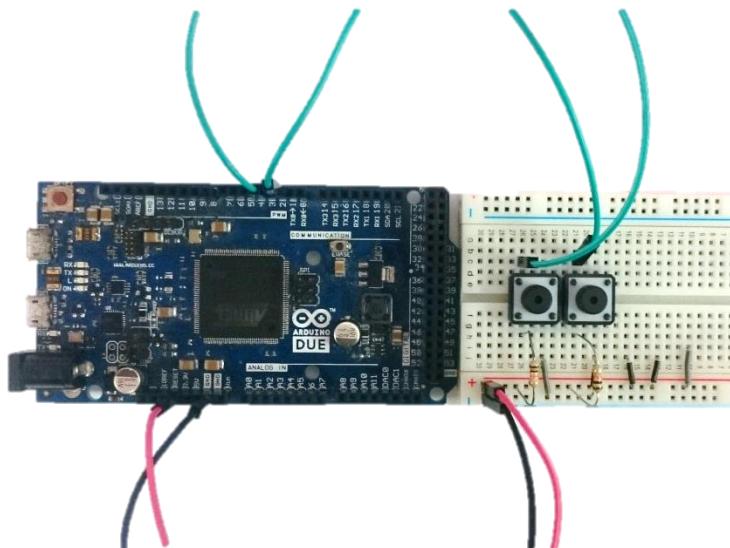
ซอฟต์แวร์ที่ใช้ในการทดสอบปุ่มกดและ Internal interrupt บน Arduino DUE

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7

ในส่วนของการดำเนินงานนี้เริ่มจากการเชื่อมต่อ Push button เข้ากับ Arduino DUE โดยต่อปุ่มกดแบบ Pull-up โดยมีผังการเชื่อมต่อ ดังนี้



ภาพที่ 4-20 ผังแสดงการเชื่อมต่อระหว่าง Arduino DUE และ Push Button



ภาพที่ 4-21 แสดงการเชื่อมต่อระหว่าง Arduino DUE และปุ่มกดแบบกดติดปล่อยดับ

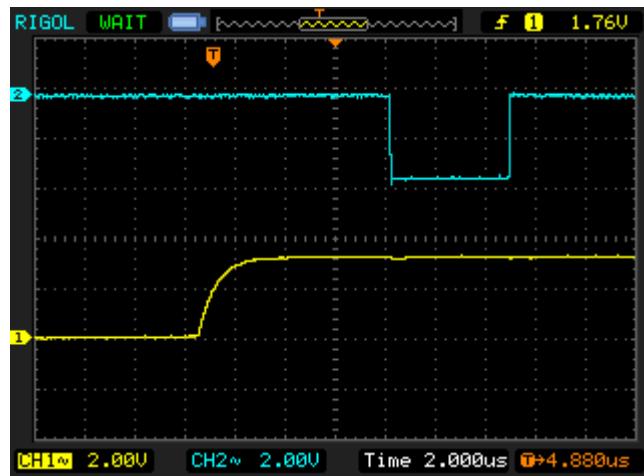
ในส่วนของการทดลองนี้ได้ทำการเปลี่ยนโภคเดบันช์แบบ Interrupt บน Arduino DUE และใช้ Software debounce ในการจัดการกับปุ่มที่มีอคติปุ่มแล้วเกิดปัญหา bouncing ในแต่ละครั้ง โดยที่เปลี่ยนโภคเดบันช์เพื่อนับค่าโดยที่ทำการกดปุ่มที่ 1 นั้นจะทำการเพิ่มค่าและปุ่มที่ 2 ก็จะลดค่าและแสดงค่าออกทาง Serial ที่มี Buadrate 9600

ผลการทดลองการกดปุ่มสองปุ่มเพื่อลดหรือเพิ่มค่าตัวแปรและแสดงค่าออกทาง Serial monitor

```
Interrupt with Debouncing test
Decrease count = -1
Decrease count = -2
Decrease count = -3
Decrease count = -4
Decrease count = -5
Increase count = -4
Increase count = -3
Increase count = -2
Increase count = -1
Increase count = 0
Increase count = 1
Increase count = 2
Increase count = 3
Increase count = 4
Increase count = 5
```

ภาพที่ 4-22 แสดงถึงการกดปุ่มเพื่อลดหรือเพิ่มค่าให้แก่ค่าตัวแปรและแสดงออกทาง Serial monitor

การทดสอบ Time respond ของปุ่มกดแบบ Interrupt สามารถทำได้โดยทำการกดปุ่มจากนั้นไป Toggle ขาของ Digital I/O แล้วใช้ Oscilloscope ในการจับเวลาความต่างของช่วงขาขึ้นของปุ่มกดแล้วขา Digital I/O ดังภาพที่ 4-23 โดยที่สัญญาณช่องที่ 1 คือ สัญญาณปุ่มกดที่ทำงานแบบ Active low และตั้งค่า Interrupt แบบ Rising edge สัญญาณช่องที่ 2 คือ สัญญาณขา Digital I/O ที่สั่งให้ toggle เมื่อสัญญาณจากปุ่มกดเป็น High จากภาพจะเห็นว่าสัญญาณใช้ Time responds เป็นเวลาประมาณ 6 us



ภาพที่ 4-23 แสดงถึง Time responds ของปุ่มกดเมื่อใช้งานแบบ Interrupt

#### 4.1.6 การทดสอบการใช้ ON/OFF Switch บน Arduino DUE

ในส่วนของการทดสอบปุ่มกดนี้ ได้ทำการทดลองโดยใช้งาน ON/OFF Switch บน Arduino DUE เพื่อทดสอบว่าระบบสามารถใช้งานร่วมกันได้จริง โดยที่อุปกรณ์และการดำเนินงานดังนี้

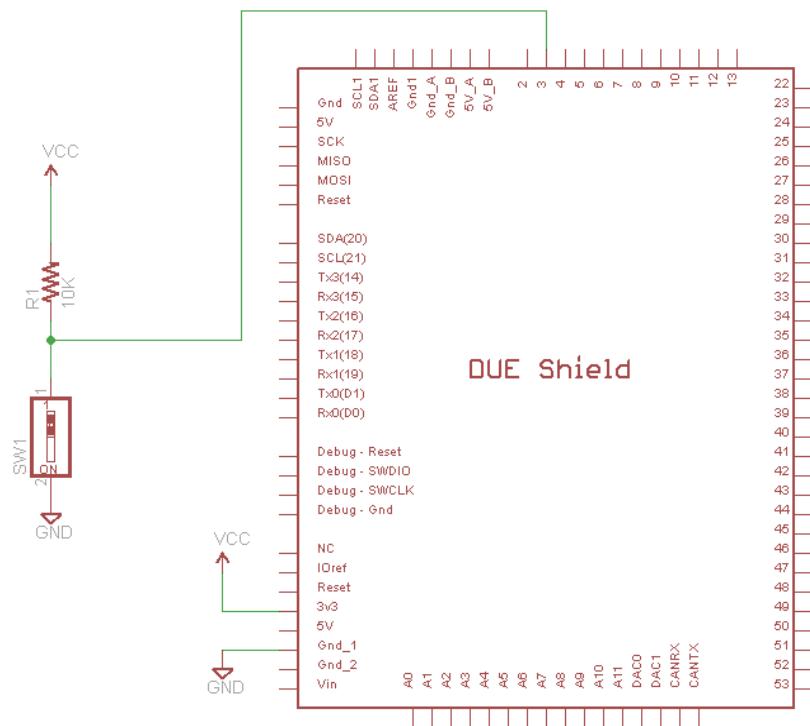
อุปกรณ์ที่ใช้ในการทดสอบ การใช้งาน ON/OFF Switch บน Arduino DUE

- ON/OFF Switch    จำนวน 1 ชิ้น
- ตัวต้านทาน ขนาด 10K                                  จำนวน 1 ชิ้น
- Arduino DUE    จำนวน 1 ชิ้น

ซอฟต์แวร์ใช้ในการทดสอบ การใช้งาน ON/OFF Switch บน Arduino DUE

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7

ในส่วนของการดำเนินงานนี้เริ่มจากทำการเชื่อมต่อ ON/OFF Switch เป็นกับ Arduino DUE ที่พอร์ต Digital I/O ดัง ภาพที่ 4-24

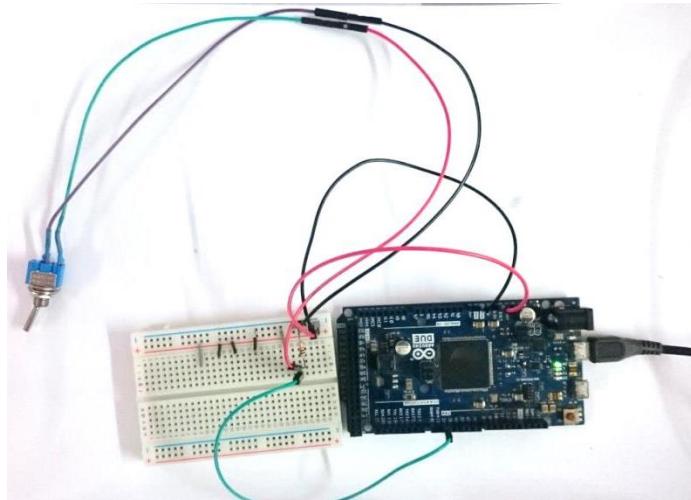


ภาพที่ 4-24 แสดงการเชื่อมต่อระหว่าง Arduino DUE และ ON/OFF Switch

ในส่วนของการทดลองนี้ได้ทำการเขียนโค้ดแบบ Interrupt บน Arduino DUE โดยที่ถ้าเกิดค่าที่ Digital input ของ xm มีค่าเป็น High จะสั่งให้ LED บนบอร์ด Arduino DUE ติด และหากขา Digital input มีค่าเป็น Low จะสั่งให้ LED บนบอร์ด ดับ

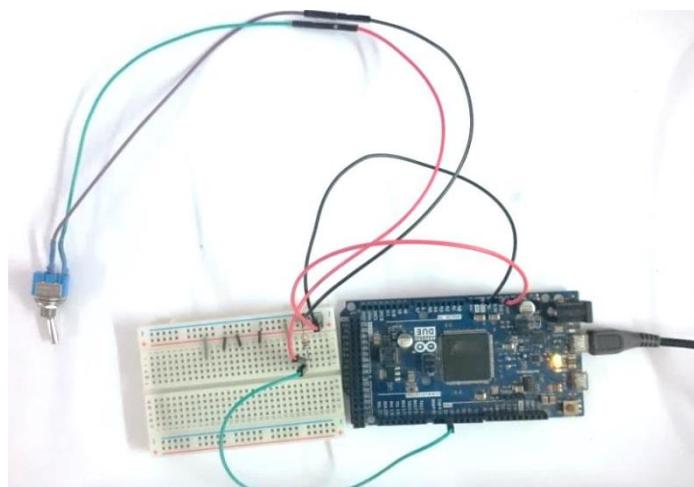
ผลการทดลองการใช้งาน ON/OFF Switch บน Arduino DUE

อุปกรณ์ ON/OFF Switch สามารถทำงานได้ตามที่ต้องการโดยมีผลการทดลองตามภาพดังนี้



ภาพที่ 4-25 แสดงถึงความสัมพันธ์ระหว่างตำแหน่งของ ON/OFF switch และ LED บน

Arduino DUE



ภาพที่ 4-26 แสดงถึงความสัมพันธ์ระหว่างตำแหน่งของ ON/OFF switch และ LED บน

Arduino DUE

จากภาพที่ 4-25 และ 4-26 LED บนบอร์ด Arduino DUE จะดับเมื่อ Switch อยู่ในตำแหน่ง OFF และ LED บนบอร์ด Arduino DUE จะติดเมื่อ Switch อยู่ในตำแหน่ง ON

#### 4.1.7 การทดสอบการประสิทธิภาพของการอ่านค่า Analog บน Arduino DUE

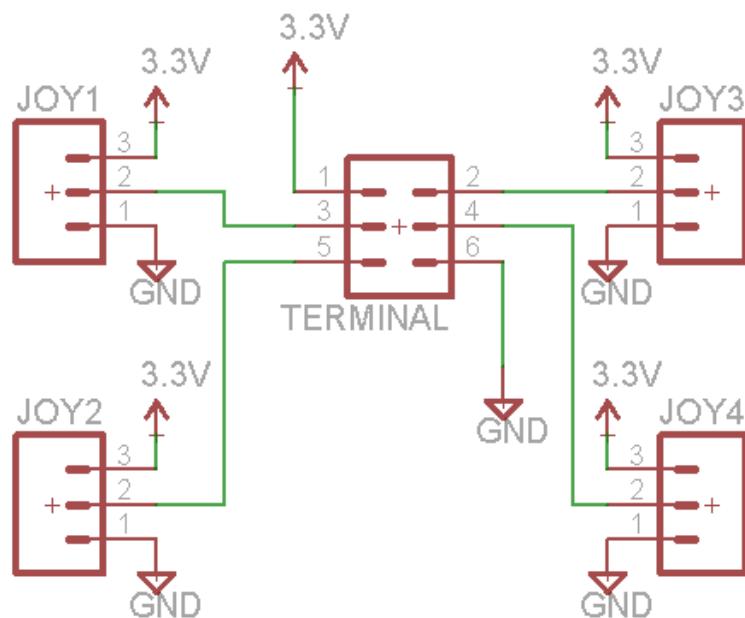
ในส่วนของการทดสอบการประสิทธิภาพของการอ่านค่า Analog บน Arduino DUE เพื่อทดสอบว่า Arduino DUE นั้นมีความสามารถในการอ่านค่า Analog ทั้งหมด 4 input อย่างไร อุปกรณ์ที่ใช้ในการทดสอบการประสิทธิภาพของการอ่านค่า Analog บน Arduino DUE

- Joy stick (ตัวด้านท่านแบบปรับค่าได้) จำนวน 4 ชิ้น
- Module Joy stick จำนวน 1 ชิ้น
- Arduino DUE จำนวน 1 ชิ้น

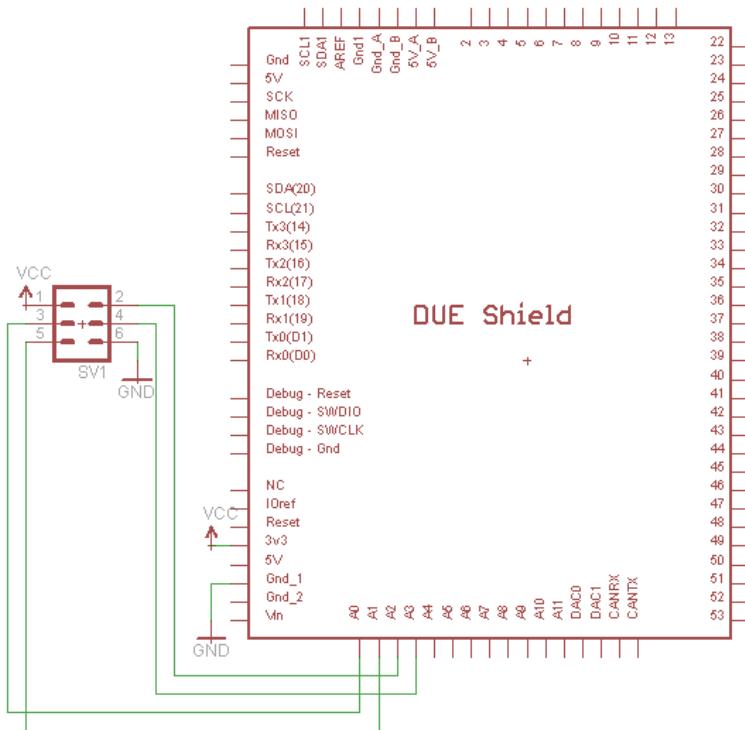
ซอฟต์แวร์ใช้ในการทดสอบประสิทธิภาพของการอ่านค่า Analog บน Arduino DUE

- OS Windows 7 64-bit Service pack I
- Arduino IDE version 1.5.7

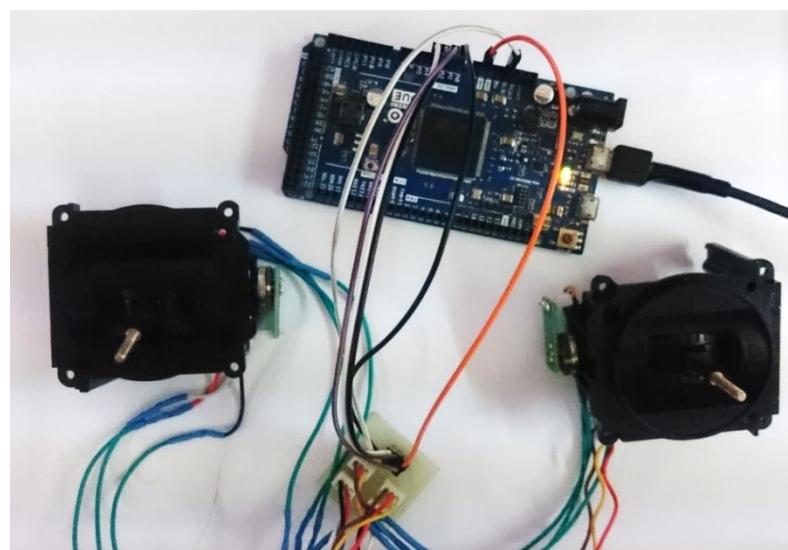
ในส่วนของการคำนึงงานนี้เริ่มจากการเชื่อมต่อ Joy stick เข้ากับ Joy stick module จากนั้นนำ joy stick module ไปเชื่อมต่อ กับ ที่พอร์ต Digital I/O ของ Arduino DUE ดังภาพ



ภาพที่ 4-27 แสดงถึงการเชื่อมต่อ Joy stick เข้ากับ Module Joy stick



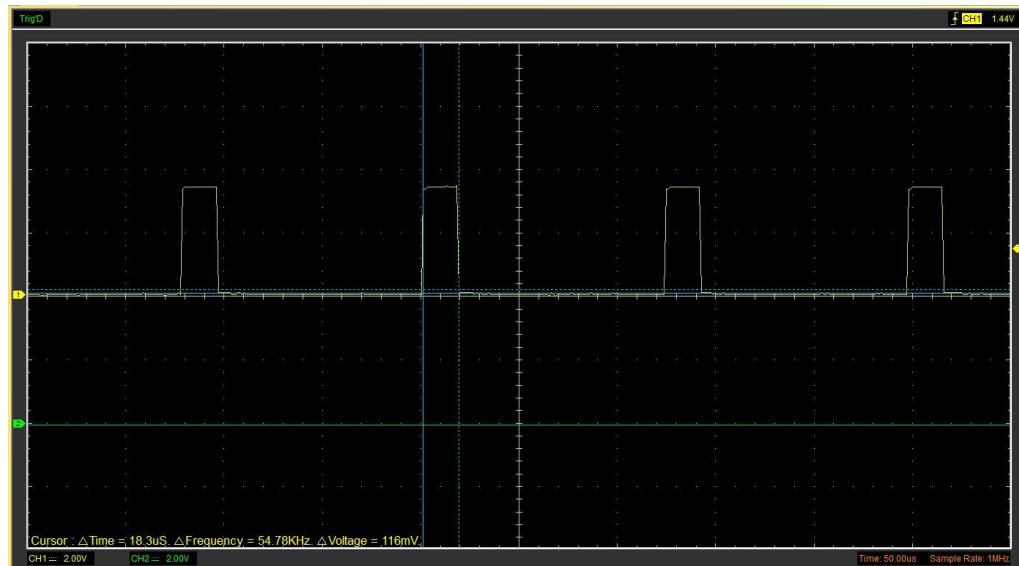
ภาพที่ 4-28 แสดงถึงการเชื่อมต่อ Module Joy stick กับ Arduino DUE



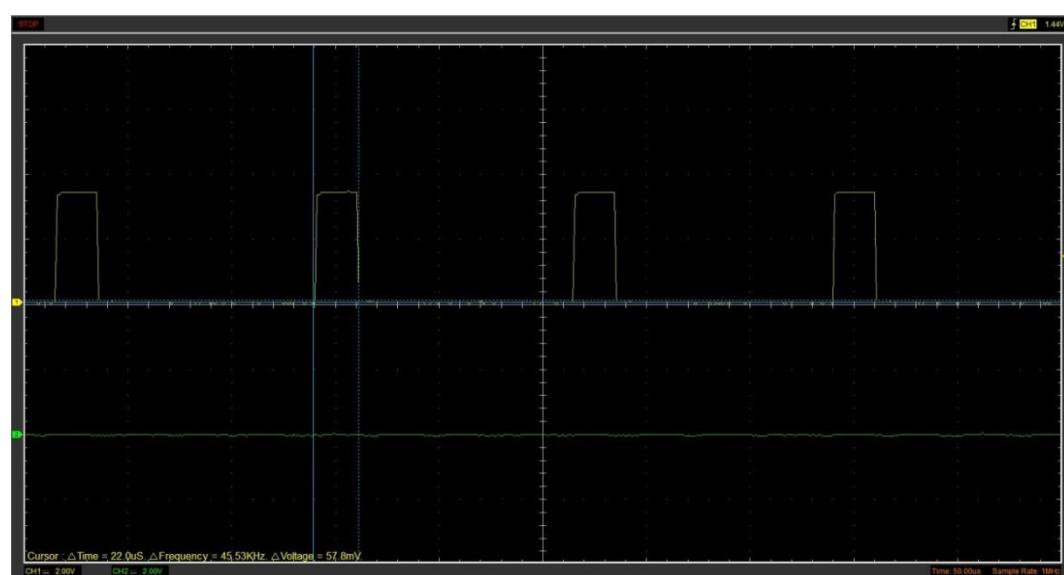
ภาพที่ 4-29 แสดงถึงการเชื่อมต่อ Module Joy stick กับ Arduino DUE

การทดลองเริ่มจากการเขียนโค้ดเพื่อทำการทดสอบประสิทธิภาพของการอ่านค่าจาก Analog ทั้งหมด 4 channel จากนั้นทำการ toggle ขา digital I/O เพื่อจับเวลาการอ่านค่า Analog

ทุก ๆ 1 us หลังจากนั้นทำการ map ค่าของสัญญาณของช่วง Analog read จากช่วง 200 - 900 3.3 volt/1024 unit ให้อยู่ในช่วง 1000 - 1950 us เพื่อเตรียมพร้อมสำหรับการส่งค่าไปยัง Mavlink gateway จากนั้นทำการ toggle ขา digital I/O เพื่อทำการวัดเวลาในการดำเนินการโดยที่ผลการทดลองเป็นไปดังนี้



ภาพที่ 4-30 แสดงถึงการวัดค่าเวลาของการ อ่านค่า Analog ทั้ง 4 Chanel โดยใช้เวลา 18.3 us



ภาพที่ 4-31 แสดงถึงการวัดค่าเวลาของการอ่านค่า Analog ทั้ง 4 Chanel ทำการ map ค่าของ Analog input โดยที่ใช้เวลา 22 us

ผลการทดลองนี้ก่อตัวได้ว่าสามารถใช้เวลาในการทำการอ่านค่า Analog input และทำการ map ค่าให้อยู่ในช่วง 1000 - 1950 us เพื่อเตรียมการส่งไปยัง Mavlink Gateway ใช้เวลาอยู่ที่ 22 us

#### 4.2 การทดสอบการรับส่งข้อมูลระหว่างส่วนแสดงผลและเมฟลิงค์เกตเวย์

ในการนำข้อมูลมาแสดงผลของระบบควบคุมและแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยานไร้คนขับแบบ 4 ใบพัดนั้น ได้นำข้อมูลจากทาง Mavlink Gateway มาแสดง เพราะฉะนั้นจึงต้องมีการเขียนต่อและรับส่งข้อมูลถึงกันด้วย Custom protocol จึงต้องแสดงให้เห็นว่าสามารถรับส่งข้อมูลกันได้อย่างถูกต้อง

โดยที่แบ่งการทดสอบออกเป็น 3 ส่วน ได้แก่

- 1) การทดสอบการรับ/ส่ง ข้อมูลพื้นฐานหรือ Data Stream
- 2) การทดสอบการรับค่า Parameter
- 3) การทดสอบการส่งค่า RC override ด้วยความถี่ 20 Hz

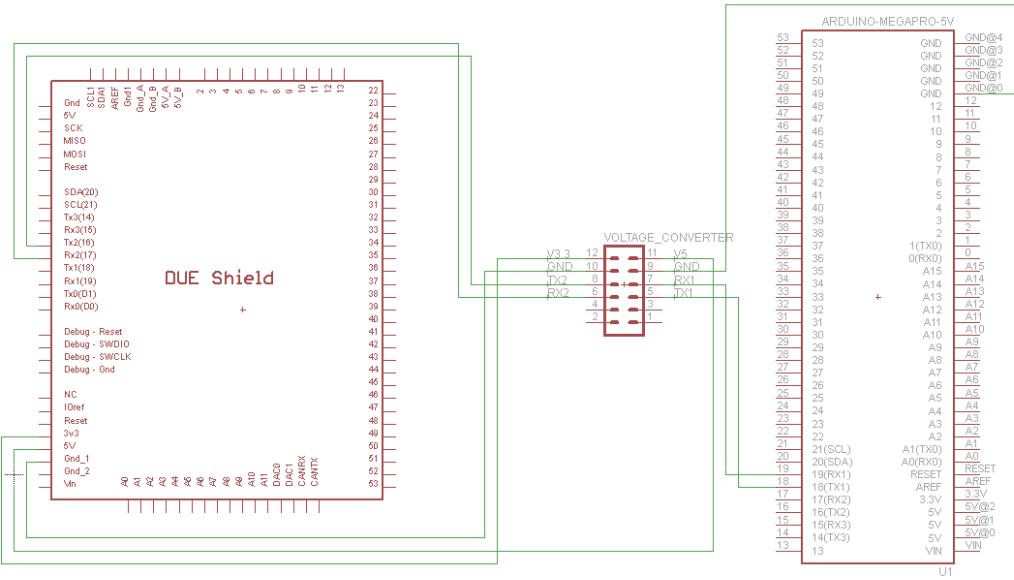
##### 4.2.1 การทดสอบการรับ/ส่ง ข้อมูลพื้นฐานหรือ Data stream

ในการสื่อสารกันระหว่าง Ground Station และ Mavlink gateway นั้นจะมีข้อมูลพื้นฐานในการสื่อสารกันทั้งหมด 5 แบบ ได้แก่ โดยที่ CMD อยู่ระหว่าง 11 – 15 โดยจากบทที่ 3 CMD คือ Message ID โดยที่สามารถทำการทดลองได้โดยทำการกำหนดค่าของแต่ละ Message ID ให้เริ่มจากค่าคงที่ค่าหนึ่ง จากนั้น Mavlink Gateway จะทำการส่งข้อมูลยัง Ground Station และจึงทำการเพิ่มค่าของแต่ละข้อมูลทีละ 1 เมื่อสำเร็จในการส่งของแต่ละรอบจากนั้น Ground Station จะนำค่าที่ได้รับแสดงค่าออกมาทาง Serial monitor



ภาพที่ 4-32 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway

ภาพผังการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway



ภาพที่ 4-33 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway

เนื่องจาก Arduino DUE สามารถรับแรงดันได้เพียง 3.3 V แต่ในขณะ Arduino Mega ต้องใช้แรงดันในการทำงาน 5 V จึงจำเป็นต้องใช้ Voltage Converter ในการแปลงค่าแรงดันระหว่างทั้ง 2 บอร์ด

ตารางที่ 4-5 แสดง Pin mapping ระหว่าง Ground station และ Maylink gateway

Arduino DUE		Module : Voltage converter		Arduino Mega	
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
nc	nc	VCC LV	12	nc	nc
VCC (5V)	5V	VCC HV	11	nc	nc
GND	GND	GND1	10	nc	nc
nc	nc	GND2	9	GND	GND
TX2	16	Bi-directional 1 LV	8	nc	nc
nc	nc	Bi-directional 1 HV	7	RX1	19
RX2	17	Bi-directional 2 LV	6	nc	nc
nc	nc	Bi-directional 2 HV	5	TX2	18
nc	nc	Bi-directional 3 LV	4	nc	nc
nc	nc	Bi-directional 3 HV	3	nc	nc
nc	nc	Bi-directional 4 LV	2	nc	nc
nc	nc	Bi-directional 4 HV	1	nc	nc

ผลการทดลองของ การทดสอบการรับ/ส่งข้อมูลพื้นฐานหรือ Data stream

```

Case CMD 13 done
Case : 13 (EXTRA I) Pitch : 49.00 Roll : 39.00 Yaw : 59.00
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 129.00 Heading : 59 Ground speed : 229.00
Case 15 done
Case 15 (RC value) : 1029 1129 1229 1329 1429 1529 1629 1729
Case CMD 11 done
Case : 11 (Heart beat) Flight mode : Stabilize Mav type : 2 Arm Status : 0 voltage : 10030 current : 1030 percent batt : 30
Case CMD 12 done
Case : 12 (EXTEND status) Fix type : 2 No.Satellite : 0 Lat : 109971102 Long : 11993118 HDOP : 2030
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : 50.00 Roll : 40.00 Yaw : 60.00
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 130.00 Heading : 60 Ground speed : 230.00
Case 15 done
Case 15 (RC value) : 1030 1130 1230 1330 1430 1530 1630 1730
Case CMD 11 done
Case : 11 (Heart beat) Flight mode : Stabilize Mav type : 2 Arm Status : 1 voltage : 10031 current : 1031 percent batt : 31
Case CMD 12 done
Case : 12 (EXTEND status) Fix type : 3 No.Satellite : 1 Lat : 109971103 Long : 11993119 HDOP : 2031
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : 51.00 Roll : 41.00 Yaw : 61.00
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 131.00 Heading : 61 Ground speed : 231.00
Case 15 done
Case 15 (RC value) : 1031 1131 1231 1331 1431 1531 1631 1731
Case CMD 11 done
Case : 11 (Heart beat) Flight mode : Stabilize Mav type : 2 Arm Status : 0 voltage : 10032 current : 1032 percent batt : 32
Case CMD 12 done
Case : 12 (EXTEND status) Fix type : 0 No.Satellite : 2 Lat : 109971104 Long : 11993120 HDOP : 2032
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : 52.00 Roll : 42.00 Yaw : 62.00
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 132.00 Heading : 62 Ground speed : 232.00
Case 15 done

```

Autoscroll

ภาพที่ 4-34 แสดงทดสอบการรับ/ส่งข้อมูลพื้นฐานหรือ Data stream

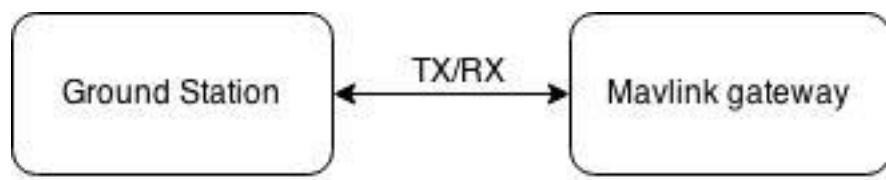
จากภาพแสดงถึงการส่งข้อมูลจาก Mavlink Gateway ไปยัง Ground Station แล้วจึงทำ  
การ Decode Massage ออกรมาเพื่อแสดงผลทาง Serial monitor  
ผลการทดลองการรับ/ส่งข้อมูลพื้นฐานหรือ Data stream

- Mavlink Gateway สามารถส่งข้อมูลมาให้ Ground Station ได้อย่างถูกต้อง
- Ground Station สามารถรับข้อมูลและนำค่ามาแสดงผลยัง Serial monitor ได้อย่าง  
ถูกต้อง

#### 4.2.2 การทดสอบการรับค่า Parameter

ในส่วนของการสื่อสารระหว่าง Ground Station และ Mavlink Gateway นั้น นอกจาก  
ส่วนของการสื่อสารระหว่าง Ground Station และ Mavlink Gateway นั้น ได้มีส่วนของการรับและปรับค่า Parameter  
ระหว่าง Ground Station และ Mavlink Gateway โดยที่ Message ID ของการสื่อสารอยู่ที่ 200, 201  
โดยการทดลองนี้จะทำการทดสอบสองส่วนคือ ส่วนที่หนึ่งรับค่า Parameter จาก Mavlink gateway  
ในส่วนของการทดสอบนี้เริ่มจากการ Ground Station ส่ง Message (CMD = 2) ไปขอ parameter  
จาก Mavlink Gateway จากนั้น Mavlink Gateway จะเริ่มทำการ Stream Parameter กลับมาให้แก่  
Ground Station (CMD = 200) โดยที่ระหว่างที่ Mavlink Gateway ทำการการส่ง Ground Station จะ

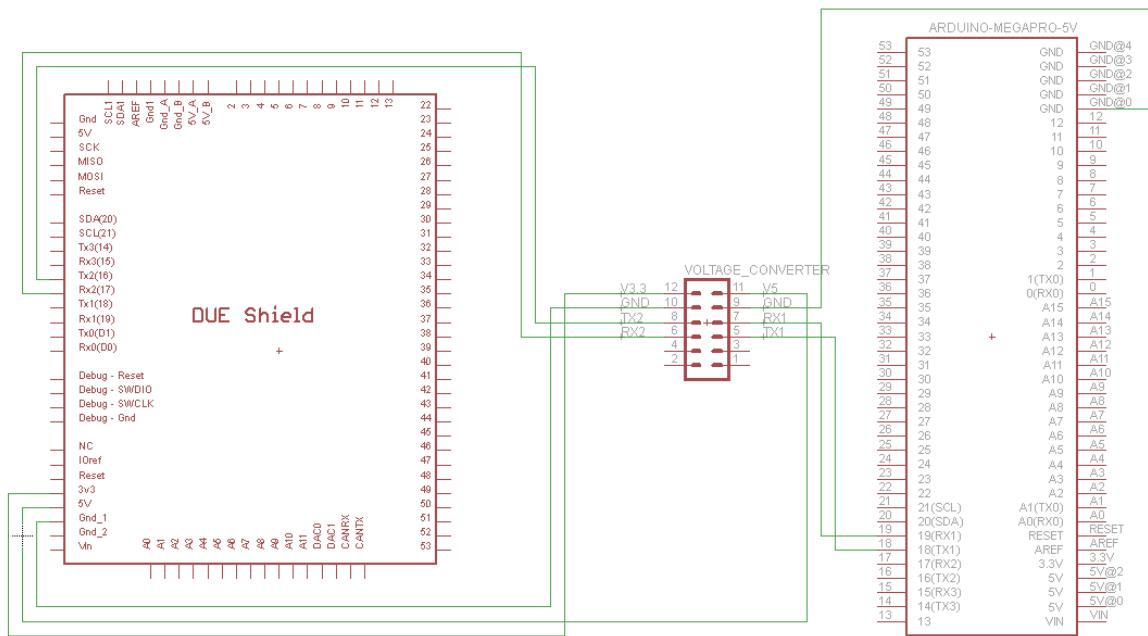
แสดงค่า Parameter แต่ละตัวทุกครั้งที่ได้รับผ่านทาง Serial monitor และเมื่อ Mavlink Gateway ส่ง parameter มาครบจะมี Massage (CMD=3) ส่งมายัง Ground Station เพื่อเป็นสัญญาณสิ้นสุดการ Stream parameter



ภาพที่ 4-35 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway

ในส่วนของการเชื่อมต่อจะนับได้ทำการใช้วงจรเดียวกับการทดลองที่ 4.2.1 จึงสามารถต่อได้ดังนี้

ภาพผังการเชื่อมต่อระหว่าง Ground Station และ Mavlink gateway



ภาพที่ 4-36 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway

ตารางที่ 4-6 แสดง Pin mapping ระหว่าง Ground Station และ Mavlink Gateway

Arduino DUE		Module : Voltage converter		Arduino Mega	
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
nc	nc	VCC LV	12	nc	nc
VCC (5V)	5V	VCC HV	11	nc	nc
GND	GND	GND1	10	nc	nc
nc	nc	GND2	9	GND	GND
TX2	16	Bi-directional 1 LV	8	nc	nc
nc	nc	Bi-directional 1 HV	7	RX1	19
RX2	17	Bi-directional 2 LV	6	nc	nc
nc	nc	Bi-directional 2 HV	5	TX2	18
nc	nc	Bi-directional 3 LV	4	nc	nc
nc	nc	Bi-directional 3 HV	3	nc	nc
nc	nc	Bi-directional 4 LV	2	nc	nc
nc	nc	Bi-directional 4 HV	1	nc	nc

#### ผลการทดสอบการทดสอบการรับค่า Parameter

```

Stop request streaming MSG
Request Parameter
0:SYSID_SW_MREV:120.000000:4
1:SYSID_SW_TYPE:10.000000:2
2:SYSID_THISMAV:1.000000:4
3:SYSID_MYGCS:255.000000:4
4:SERIAL0_BAUD:115.000000:4
5:SERIAL1_BAUD:57.000000:4
6:SERIAL2_BAUD:57.000000:4
7:SERIAL2_PROTOCOL:1.000000:2
8:TELEM_DELAY:0.000000:2
9:RTL_ALT:1500.000000:4
10:RNGFND_GAIN:0.800000:9
11:FS_BATT_ENABLE:0.000000:2
12:FS_BATT_VOLTAGE:10.500000:9
13:FS_BATT_MAHR:0.000000:9
14:FS_GPS_ENABLE:1.000000:2
15:FS_GCS_ENABLE:1.000000:2

```

ภาพที่ 4-37 แสดงถึงการสั่งหยุด Streaming Massage และวิธีเริ่มทำการขอ Parameter จาก Mavlink Gateway

```

398:EKF_MAG_GATE:3.000000:2
399:EKF_EAS_GATE:10.000000:2
400:EKF_MAG_CAL:1.000000:2
401:EKF_GLITCH_ACCE:150.000000:4
402:EKF_GLITCH_RAD:15.000000:2
403:MIS_TOTAL:0.000000:4
404:MIS_RESTART:0.000000:2
405:RNGFND_TYPE:0.000000:2
406:RNGFND_PIN:-1.000000:2
407:RNGFND_SCALING:3.000000:9
408:RNGFND_OFFSET:0.000000:9
409:RNGFND_FUNCTION:0.000000:2
410:RNGFND_MIN_CM:20.000000:4
411:RNGFND_MAX_CM:700.000000:4
412:RNGFND_STOP_PIN:-1.000000:2
413:RNGFND_SETTLE_M:0.000000:4
414:RNGFND_RMETRIC:1.000000:2
415:TERRAIN_ENABLE:1.000000:2
416:TERRAIN_SPACING:100.000000:4
Receive parameter done
Start request streaming MSG
 Autoscroll

```

ภาพที่ 4-38 แสดงถึงการ Streaming Parameter จนเสร็จแล้วจึงเริ่มทำการขอ Stream data ไปยัง Mavlink Gateway

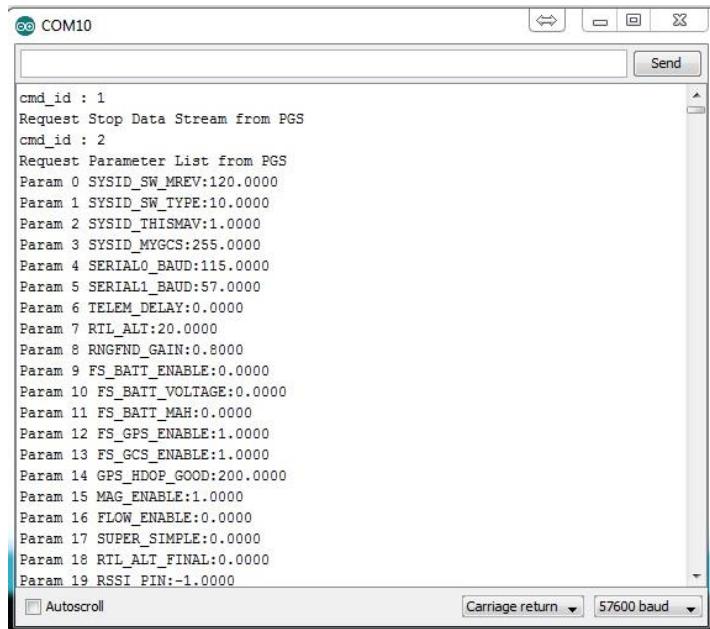
```

COM18 (Arduino Due (Programming Port))

408:RNGFND_OFFSET:0.000000:9
409:RNGFND_FUNCTION:0.000000:2
410:RNGFND_MIN_CM:20.000000:4
411:RNGFND_MAX_CM:700.000000:4
412:RNGFND_STOP_PIN:-1.000000:2
413:RNGFND_SETTLE_M:0.000000:4
414:RNGFND_RMETRIC:1.000000:2
415:TERRAIN_ENABLE:1.000000:2
416:TERRAIN_SPACING:100.000000:4
Receive parameter done
Start request streaming MSG
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.85 Roll : -1.45 Yaw : -122.55
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 5.55 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 254 3519 47419 55231 22783 40130 62747
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 5.55 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.85 Roll : -1.45 Yaw : -122.55
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 5.55 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.82 Roll : -1.46 Yaw : -122.27
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 5.79 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
 Autoscroll

```

ภาพที่ 4-39 แสดงถึงการเริ่ม Stream data หลังจากที่ Ground Station ทำการขอ Stream data หลังจากที่รับ Parameter ได้สำเร็จ



**ภาพที่ 4-40** Serial monitor ของ Mavlink Gateway เมื่อทำการส่ง Parameter

จากการทดลองการรับค่า Parameter สามารถสรุปได้ดังนี้

- 1) Ground Station สามารถทำการ request Parameter ไปยัง Mavlink Gateway เพื่อทำการขอค่า Parameter ได้
- 2) Ground Station สามารถรับค่า Parameter จาก Mavlink Gateway ได้อย่างถูกต้อง จากการทดลองเปรียบเทียบค่าระหว่าง Serial monitor
- 3) เมื่อรับค่า Parameter จนครบสามารถ ส่ง Request ไปยัง Mavlink Gateway เพื่อให้ Streaming data ต่อได้ทันที

การทดลองการส่งค่า Parameter จาก Ground Station ไปยัง Mavlink Gateway

หลังจากที่ทดสอบการรับค่า Parameter ของ Ground Station ของ Mavlink Gateway สำเร็จแล้วในส่วนนี้จะทำการทดสอบการส่งค่า Parameter กลับไปยัง Mavlink Gateway เพื่อรองรับการแก้ไขค่าของ Parameter ของ User ที่ใช้งาน จะใช้การทดลองโดยที่ใช้ Serial monitor ในการส่ง Massage (CMD 201 ) เพื่อปรับเปลี่ยนค่า Parameter และเมื่อทำการส่งสำเร็จจะมีผลกลับมารายงานทาง Serial monitor โดยที่ในส่วนนี้จะใช้การเชื่อมต่ออุปกรณ์และซอฟต์แวร์ เหมือนกับการทดลองก่อนหน้า

ผลการทดสอบการทดสอบการส่งค่า Parameter

```

Adjusted parameter ID : 97 , Value :1500.000000 Complete
65535:RC8_TRIM:1500.000000:4
Start request streaming MSG
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.82 Roll : -1.39 Yaw : -122.69
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 11.53 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.82 Roll : -1.39 Yaw : -122.69
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 11.53 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.80 Roll : -1.44 Yaw : -122.88
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 12.08 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0

```

ภาพที่ 4-41 ผลการทดสอบการทดสอบการส่งค่า Parameter

จากภาพที่ 4-41 แสดงถึงการปรับแก้ค่า Parameter ชื่อ RC8\_TRIM โดยที่ทำการส่งค่า ID และ ค่าของ Parameter โดยจากรูปได้ทำการส่ง ID คือ 97 (RC8\_TRIM มี ID คือ 97 ) และค่า 1500.00 ไปตามโพรตocols ที่ 201 เมื่อทำการส่งยัง Mavlink gateway ไปจะมีข้อความตอบกลับมาคือ 65525 : RC8\_TRIM:15000.000000:4 หมายความว่า Parameter ดังกล่าวได้ถูกปรับเปลี่ยนค่าเรียบร้อยแล้ว

```

MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
cmd_id : 201
Req to write param 97:1500.00
Req to Param
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
97,RC8_TRIM,1500.0000,4
Upload Param
65535,RC8_TRIM,1500.0000,4
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1

```

ภาพที่ 4-42 แสดงถึงการปรับค่าและส่งค่าคืนไปมายัง Ground Station ของ Mavlink Gateway

จากการทดลองการส่งค่า Parameter สามารถสรุปได้ดังนี้

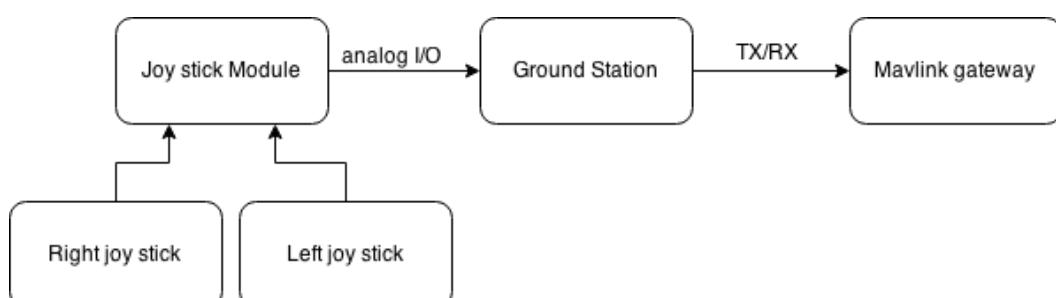
- 1) Ground Station สามารถทำการส่ง Parameter เพื่อทำการปรับแก้ค่าให้แก่ Mavlink Gateway ได้โดยใช้ Massage CMD 201 โดยใช้ ID และค่าใหม่ที่ต้องการของ Parameter
- 2) เมื่อการส่งสำเร็จจะมีค่าจาก Serial monitor ของทางฝั่ง Ground Station เพื่อทำการยืนยันว่าส่ง Parameter สำเร็จ และถ้าหาก Mavlink Gateway ได้รับค่าและสามารถนำค่าไปปรับให้แก่อุปกรณ์ควบคุมการบินได้สำเร็จจะมีข้อความแจ้งกลับมาทาง Ground Station เช่นกันโดยแจ้งเป็น Massage ID 65535 ตามด้วยชื่อและค่าใหม่ของ Parameter

#### 4.2.3 การทดลองการส่งค่า RC override ด้วยความถี่ 20 Hz

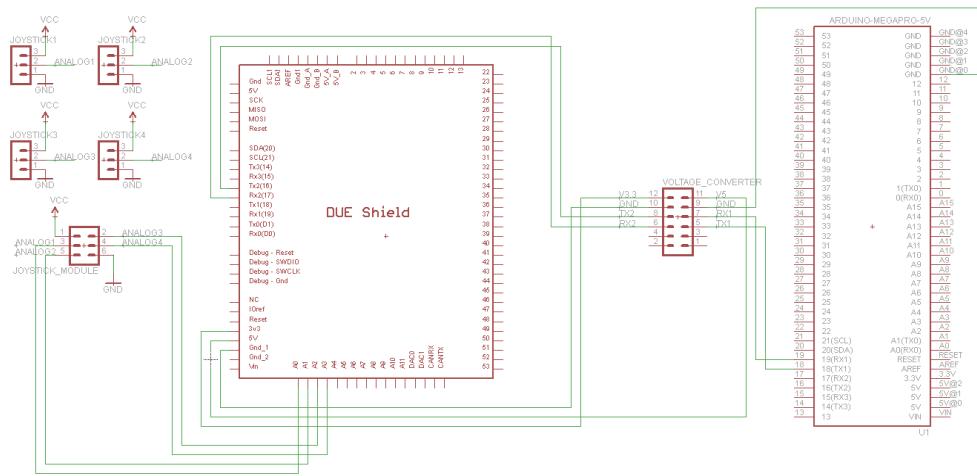
ในส่วนของการทดสอบการส่งค่า RC override จาก Ground Station ไปยัง Mavlink Gateway นั้นสามารถทำได้โดยทำการอ่านค่า Analog input จาก Ground Station ทั้งหมด 4 Chanel เพื่อเป็นตัวแทนของ Pitch, Roll, Yaw และ Throttle ตามลำดับจากนั้นทำการ map ค่า analog input ที่อ่านได้ให้อยู่ในช่วงของ 1000-1950 us เพื่อแปลงสัญญาณให้อยู่ในช่วงที่ Mavlink ต้องการใช้งาน จากนั้นจึงทำการส่งค่าออกไปด้วย Massage CMD 5

อุปกรณ์ที่ใช้ในการทดลองมีดังนี้

- |                                     |              |
|-------------------------------------|--------------|
| 1) Arduino DUE                      | จำนวน 1 ชิ้น |
| 2) Arduino Mega2560                 | จำนวน 1 ชิ้น |
| 3) Analog joystick                  | จำนวน 4 ชิ้น |
| 4) Joy stick module                 | จำนวน 1 ชิ้น |
| 5) Bi-directional voltage converter | จำนวน 1 ชิ้น |



ภาพที่ 4-43 แสดงถึง Block diagram ของการทดลองการส่งค่า RC override ด้วยความถี่ 20 Hz



ภาพที่ 4-44 แสดงผังวงจรการเชื่อมในการทดสอบการส่งค่า RC override ด้วยความถี่ 20 Hz

ตารางที่ 4-7 แสดงถึงผังการเชื่อมต่อระหว่าง Arduino DUE และ Mavlink Gateway

Arduino DUE		Module : Voltage converter		Arduino Mega	
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
nc	nc	VCC LV	12	nc	nc
VCC (5V)	5V	VCC HV	11	nc	nc
GND	GND	GND1	10	nc	nc
nc	nc	GND2	9	GND	GND
TX2	16	Bi-directional 1 LV	8	nc	nc
nc	nc	Bi-directional 1 HV	7	RX1	19
RX2	17	Bi-directional 2 LV	6	nc	nc
nc	nc	Bi-directional 2 HV	5	TX2	18
nc	nc	Bi-directional 3 LV	4	nc	nc
nc	nc	Bi-directional 3 HV	3	nc	nc
nc	nc	Bi-directional 4 LV	2	nc	nc
nc	nc	Bi-directional 4 HV	1	nc	nc

ตารางที่ 4-8 แสดงถึงผังการเชื่อมต่อระหว่าง Arduino DUE และ Module Joy stick

Arduino DUE		Module : Joy Stick	
Pin Name	Pin No.	Pin Name	Pin No.
VCC	3.3V	VCC	1
GND	GND	GND	6
Analog I/O	A0	Analog IN 1	3
Analog I/O	A1	Analog IN 2	5
Analog I/O	A2	Analog IN 3	4
Analog I/O	A3	Analog IN 4	6

ผลการทดลองการส่งค่า RC override ด้วยความถี่ 20 Hz

COM18 (Arduino Due (Programming Port))

RC OVERRIDE Send : 1512 : 1502 : 1097 : 1504 :  
RC OVERRIDE Send : 1511 : 1500 : 1095 : 1501 :  
RC OVERRIDE Send : 1511 : 1501 : 1093 : 1502 :  
RC OVERRIDE Send : 1512 : 1501 : 1097 : 1501 :  
RC OVERRIDE Send : 1513 : 1501 : 1086 : 1504 :  
RC OVERRIDE Send : 1512 : 1502 : 1099 : 1504 :  
RC OVERRIDE Send : 1512 : 1501 : 1095 : 1504 :  
RC OVERRIDE Send : 1511 : 1501 : 1097 : 1505 :  
RC OVERRIDE Send : 1510 : 1501 : 1097 : 1504 :  
RC OVERRIDE Send : 1512 : 1500 : 1097 : 1504 :  
RC OVERRIDE Send : 1512 : 1500 : 1095 : 1502 :  
RC OVERRIDE Send : 1512 : 1500 : 1095 : 1504 :  
RC OVERRIDE Send : 1511 : 1500 : 1095 : 1502 :  
RC OVERRIDE Send : 1512 : 1501 : 1097 : 1505 :  
RC OVERRIDE Send : 1512 : 1500 : 1097 : 1504 :  
RC OVERRIDE Send : 1510 : 1501 : 1097 : 1502 :  
RC OVERRIDE Send : 1504 : 1500 : 1097 : 1501 :  
RC OVERRIDE Send : 1512 : 1501 : 1095 : 1504 :  
RC OVERRIDE Send : 1512 : 1501 : 1097 : 1504 :  
RC OVERRIDE Send : 1512 : 1501 : 1095 : 1502 :  
RC OVERRIDE Send : 1512 : 1501 : 1097 : 1504 :  
RC OVERRIDE Send : 1512 : 1501 : 1095 : 1505 :  
RC OVERRIDE Send : 1511 : 1501 : 1099 : 1504 :  
RC OVERRIDE Send : 1512 : 1502 : 1095 : 1504 :  
RC OVERRIDE Send : 1511 : 1501 : 1095 : 1505 :  
RC OVERRIDE Send : 1511 : 1500 : 1095 : 1504 :  
RC OVERRIDE Send : 1510 : 1498 : 1095 : 1504 :  
RC OVERRIDE Send : 1510 : 1501 : 1097 : 1504 :  
RC OVERRIDE Send : 1511 : 1501 : 1097 : 1504 :  
RC OVERRIDE Send : 1510 : 1500 : 1097 : 1504 :  
RC OVERRIDE Send : 1511 : 1500 : 1095 : 1504 :  
RC OVERRIDE Send : 1511 : 1502 : 1097 : 1504 :  
RC OVERRIDE Send : 1512 : 1498 : 1097 : 1505 :  
  
Autoscroll

ภาพที่ 4-45 แสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground Station และ Mavlink Gateway โดยให้ Throttle อยู่ในตำแหน่งต่ำสุด

```

COM18 (Arduino Due (Programming Port))
RC OVERRIDE Send : 1512 : 1837 : 1048 : 1510 :
RC OVERRIDE Send : 1513 : 1914 : 1052 : 1511 :
RC OVERRIDE Send : 1511 : 1916 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1920 : 1047 : 1510 :
RC OVERRIDE Send : 1511 : 1921 : 1048 : 1511 :
RC OVERRIDE Send : 1512 : 1920 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1920 : 1050 : 1510 :
RC OVERRIDE Send : 1512 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1923 : 1048 : 1511 :
RC OVERRIDE Send : 1511 : 1923 : 1048 : 1511 :
RC OVERRIDE Send : 1511 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1513 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1511 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1921 : 1047 : 1510 :
RC OVERRIDE Send : 1511 : 1923 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1921 : 1048 : 1513 :
RC OVERRIDE Send : 1510 : 1924 : 1050 : 1510 :
RC OVERRIDE Send : 1511 : 1924 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1926 : 1048 : 1510 :
RC OVERRIDE Send : 1511 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1512 : 1921 : 1047 : 1510 :
RC OVERRIDE Send : 1511 : 1923 : 1048 : 1511 :
RC OVERRIDE Send : 1512 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1510 : 1924 : 1050 : 1510 :
RC OVERRIDE Send : 1511 : 1924 : 1048 : 1510 :
RC OVERRIDE Send : 1510 : 1926 : 1048 : 1510 :
RC OVERRIDE Send : 1513 : 1926 : 1050 : 1510 :
RC OVERRIDE Send : 1510 : 1924 : 1047 : 1511 :
RC OVERRIDE Send : 1508 : 1924 : 1048 : 1510 :
RC OVERRIDE Send : 1511 : 1923 : 1047 : 1510 :
RC OVERRIDE Send : 1510 : 1924 : 1036 : 1510 :
RC OVERRIDE Send : 1510 : 1924 : 1048 : 1510 :
RC OVERRIDE Send : 1508 : 1926 : 1048 : 1510 :
RC OVERRIDE Send : 1506 : 1923 : 1052 : 1508 :
RC OVERRIDE Send : 1506 : 1923 : 1048 : 1511 :
RC OVERRIDE Send : 1506 : 1923 : 1047 : 1508 :
RC OVERRIDE Send : 1502 : 1921 : 1048 : 1510 :
RC OVERRIDE Send : 1498 : 1921 : 1048 : 1501 :
RC OVERRIDE Send : 1499 : 1921 : 1048 : 1510 :
RC OVERRIDE Recv : 1513 1921 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1511 1921 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1512 1921 1048 1510 0 0 0 0
RC Override Recv : 1512 1921 1047 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1511 1923 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1512 1921 1048 1513 0 0 0 0
cmd_id : 50
RC Override Recv : 1510 1924 1050 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1511 1924 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1510 1926 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1511 1923 1047 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1510 1924 1036 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1510 1924 1048 1510 0 0 0 0
MAVLink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
cmd_id : 50
RC Override Recv : 1508 1926 1048 1510 0 0 0 0
cmd_id : 50
RC Override Recv : 1506 1923 1052 1508 0 0 0 0
cmd_id : 50
RC Override Recv : 1506 1923 1048 1511 0 0 0 0
cmd_id : 50
RC Override Recv : 1506 1923 1047 1508 0 0 0 0
cmd_id : 50
RC Override Recv : 1502 1921 1048 1510 0 0 0 0
RC Override Recv : 1498 1921 1048 1501 0 0 0 0
cmd_id : 50
RC Override Recv : 1499 1921 1048 1510 0 0 0 0
RC Override Recv : 1502 1921 1048 1510 0 0 0 0

```

**ภาพที่ 4-46** แสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground Station และ Mavlink Gateway โดยให้ Pitch อยู่ในตำแหน่งต่ำสุด

```

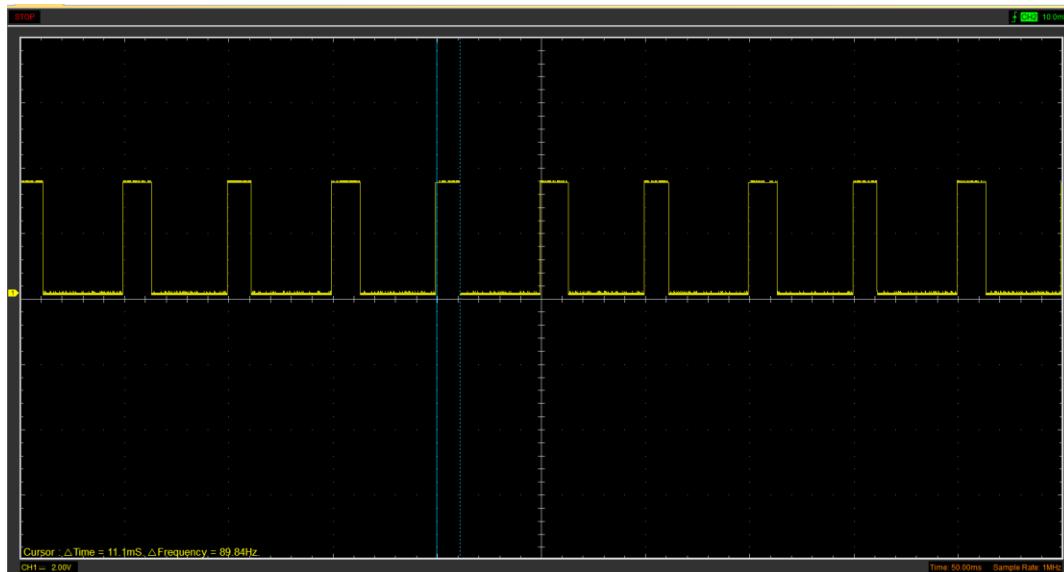
COM13
RC Override Recv : 1885 1626 1521 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1625 1519 1387 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1626 1519 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1885 1625 1519 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1626 1521 1387 0 0 0 0
MAVLink HeartBeat MAV: 2 Modes: Stabilize Armed: 1 SysId: 1 ComId: 1 Autopilot: 3 Status: 5
cmd_id : 50
RC Override Recv : 1887 1624 1519 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1626 1523 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1885 1625 1521 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1626 1517 1387 0 0 0 0
cmd_id : 50
RC Override Recv : 1885 1625 1521 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1885 1626 1519 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1624 1519 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1885 1624 1521 1389 0 0 0 0
cmd_id : 50
RC Override Recv : 1887 1626 1521 1390 0 0 0 0
cmd_id : 50

```

**ภาพที่ 4-47** รูปแสดงถึงผลการทดสอบการรับส่งค่า RC Override ระหว่าง Portable Ground Station และ Mavlink Gateway โดยทำการตั้ง ARM

ภาพที่ 4-45 แสดงถึงการทดลองปรับตำแหน่งค่า throttle (ชี้งตำแหน่งของข้อมูลเรียงกันดังนี้ Roll, Pitch, Throttle, Yaw ตามลำดับ) ให้อยู่ในช่วงต่ำสุดมีค่าโดยมีค่าประมาณ 1090 us โดยที่ Serial monitor ทางขวาคือ Serial monitor ของ Ground Station และทางซ้ายคือของ Mavlink Gateway และจากภาพที่ 4-46 เป็นการปรับค่าตำแหน่ง Pitch ให้มีค่าสูงที่สุดเพื่อทำการเปรียบเทียบว่าข้อมูลส่งถูกต้องหรือไม่โดยมีค่าประมาณ 1920 โดยสังเกตว่าทั้งสองฟังก์ชันมีค่าคล้ายกันโดยที่ฟังก์ชันอาจมีความเร็วสูงกว่าชิ้งสามารถวัดได้จากเวลาในการทำงานของระบบดังกล่าว ซึ่งจะทำการทดลองในการทดลองถัดไปและจากภาพที่ 4-47 เป็นการสั่ง Arm อุปกรณ์ควบคุมการบินด้วย Joy stick โดยที่นำมาแสดงคือ Serial monitor ของ Mavlink Gateway สถานะ Armed = 1 ซึ่งหมายความว่าระบบสามารถสั่งการอุปกรณ์ควบคุมการบินผ่านทาง Joy stick ได้

ในการทดลองต่อมาได้ทำการสร้างขา Digital I/O เพื่อทำการ toggle ขา I/O เมื่อ Ground Station ทำการ Override RC สำหรือหนึ่งครั้งแล้วทำการวัดค่าสัญญาณด้วย Oscilloscope



ภาพที่ 4-48 แสดงถึงเวลาที่ใช้ในการ Override RC 1 ครั้งและทำคำสั่ง Override RC ทุก ๆ 5 Hz

จากการทดสอบถึงเวลาที่ใช้ในการ Override RC 1 ครั้งและทำคำสั่ง Override RC ทุก ๆ 5 Hz โดยที่สามารถวัดเวลาของการทำคำสั่ง Override RC ได้เวลาประมาณ 10 ms ต่อการทำงานหนึ่งครั้ง

จากการทดลองการส่งค่า Parameter สามารถสรุปได้ดังนี้

- 1) Ground Station สามารถส่งค่า RC Override ไปยัง Mavlink Gateway ได้อีกถูกต้อง

- 2) Ground Station สามารถสั่งการอุปกรณ์ควบคุมการบินด้วย Joy stick ได้โดยใช้ Mavlink เป็นสื่อกลาง
- 3) Ground Station สามารถทำงานได้ด้วยความถี่ 20 Hz
- 4) Ground Station ใช้เวลาในการทำคำสั่ง RC Override โดยใช้เวลาประมาณ 10 ms

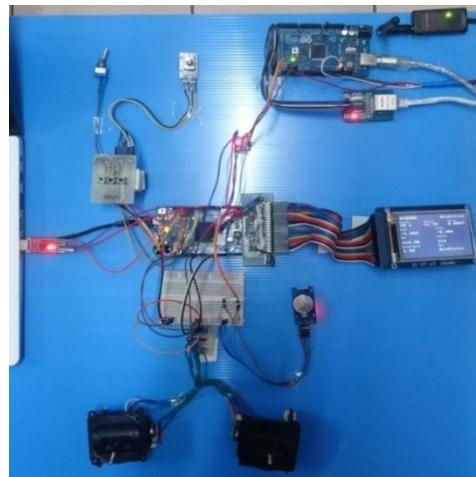
### 4.3 การทดสอบระบบจริง

หลังจากที่ทำการทดสอบอุปกรณ์ต่าง ๆ ร่วมกันแล้วนั้นเราก็ได้จัดการทำระบบจริงโดยที่เนื่องในของระบบจริงมีดังนี้

- GLCD จะต้องสามารถอัพเดทข้อมูลที่ความถี่ไม่ต่ำกว่า 5 Hz
- SD card จะต้องบันทึกข้อมูลที่ได้รับจาก Mavlink Gateway ทุก ๆ 1 วินาที
- ในการเขียนข้อมูลลง SD Card นั้นจะมีการบันทึกเวลาปัจจุบันทุกรอบที่ทำการบันทึก
- Ground Station ต้องสามารถแสดงค่า Parameter ให้แก่ผู้ใช้งานได้
- สามารถรับข้อมูลจาก Mavlink Gateway เพื่อนำมาแสดงผลบนหน้าจอ GLCD ได้
- ในการส่งข้อมูลกลับไปที่ Mavlink Gateway นั้น Ground Station นั้นต้องสามารถ Override RC ให้แก่ Mavlink Gateway ด้วยความถี่ไม่ต่ำกว่า 20 Hz
- ในการส่งข้อมูลกลับไปที่ Mavlink Gateway นั้น Ground Station นั้นต้องสามารถ ปรับค่า Parameter และส่งกลับไปยัง Mavlink ได้
- สามารถใช้ Rotary encoder ในการปรับค่าพารามิเตอร์ให้แก่ระบบได้ สามารถรับ/ส่งข้อมูล ด้วย Custom Protocol ได้อย่างถูกต้อง
- ผู้ใช้งานสามารถเปิดและปิดการใช้งาน Override RC ด้วย ON/OFF Switch ได้ โดยมีอุปกรณ์ที่ใช้ทดลองดังนี้

- Arduino DUE	จำนวน 1 ชิ้น
- GLCD Module LB04301 มาพร้อม SD card socket	จำนวน 1 ชิ้น
- Real time clock (DS3231)	จำนวน 1 ชิ้น
- Gimbal module (Throttle + Yaw)	จำนวน 1 ชิ้น
- Gimbal module (Pitch + Roll)	จำนวน 1 ชิ้น
- ON/OFF switch	จำนวน 1 ชิ้น
- Push button	จำนวน 3 ชิ้น
- PCB push button	จำนวน 1 ชิ้น
- PCB joy stick module	จำนวน 1 ชิ้น

- PCB GLCD pin converter จำนวน 1 ชิ้น
  - Mavlink Gateway จำนวน 1 ชุด
  - Multirotor จำนวน 1 ลำ
- เมื่อนำมาต่อแล้วจึงได้รูปวงจรดังนี้



ภาพที่ 4-49 แสดงถึงการเชื่อมต่อระหว่าง Ground Station และ Mavlink Gateway

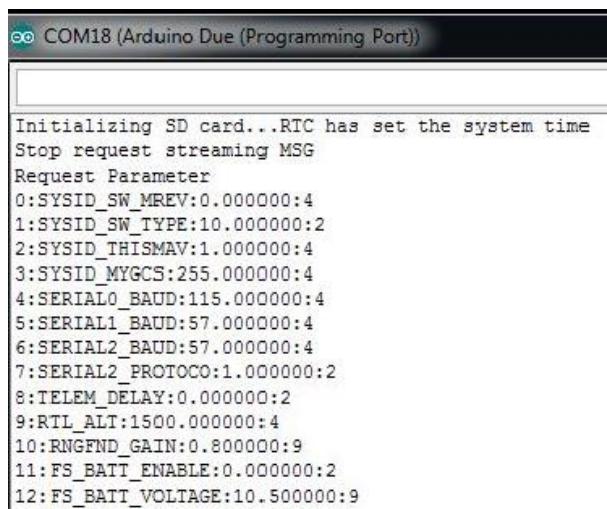
#### 4.3.1 การทดสอบการเปิดใช้รีมตันในระบบจริง

```
COM18 (Arduino Due (Programming Port))

Initializing SD card...RTC has set the system time
Stop request streaming MSG
Request Parameter
0:SYSID_SW_MREV:0.000000:4
1:SYSID_SW_TYPE:10.000000:2
2:SYSID_THISMAV:1.000000:4
3:SYSID_MYGCS:255.000000:4
4:SERIAL0_BAUD:115.000000:4
5:SERIAL1_BAUD:57.000000:4
6:SERIAL2_BAUD:57.000000:4
7:SERIAL2_PROTOCOL:1.000000:2
8:TELEM_DELAY:0.000000:2
9:RTL_ALT:1500.000000:4
10:RNGFNS_GAIN:0.800000:9
11:FS_BATT_ENABLE:0.000000:2
12:FS_BATT_VOLTAGE:10.500000:9
13:FS_BATT_MAH:0.000000:9
14:FS_GPS_ENABLE:1.000000:2
15:FS_GCS_ENABLE:1.000000:2
16:GPS_HDOP_GOOD:200.000000:4
17:MAG_ENABLE:1.000000:2
18:FLOW_ENABLE:0.000000:2
```

ภาพที่ 4-50 การเปิดระบบของ Ground Station สามารถทำงานได้ตามเงื่อนไขข้างต้นที่กล่าวไว้

#### 4.3.2 ทดสอบการรับค่า Parameter จาก Mavlink Gateway



```

Initialization SD card...RTC has set the system time
Stop request streaming MSG
Request Parameter
0:SYSID_SW_MREV:0.000000:4
1:SYSID_SW_TYPE:10.000000:2
2:SYSID_THISMAV:1.000000:4
3:SYSID_MYGCS:255.000000:4
4:SERIAL0_BAUD:115.000000:4
5:SERIAL1_BAUD:57.000000:4
6:SERIAL2_BAUD:57.000000:4
7:SERIAL2_PROTOCO:1.000000:2
8:TELEM_DELAY:0.000000:2
9:RTL_ALT:1500.000000:4
10:RNGFND_GAIN:0.800000:9
11:FS_BATT_ENABLE:0.000000:2
12:FS_BATT_VOLTAGE:10.500000:9

```

ภาพที่ 4-51 Serial monitor การรับค่า Parameter มาจาก Mavlink Gateway

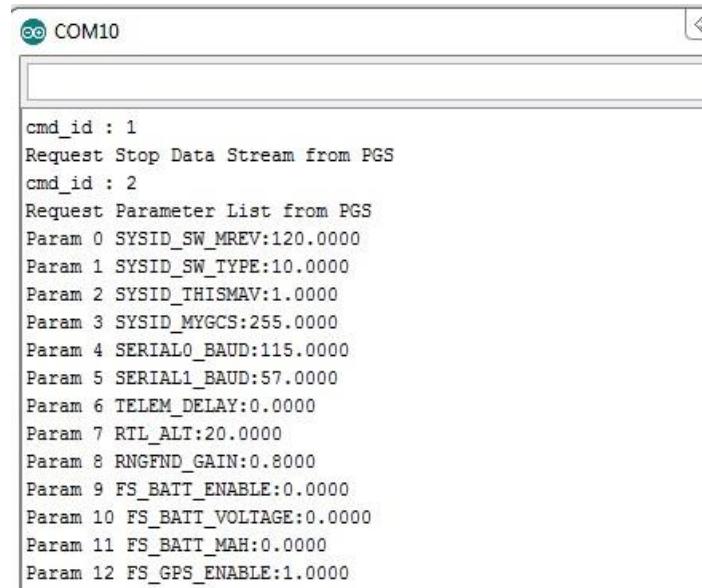


```

0,SYSID_SW_MREV:120.000000:4
1:SYSID_SW_TYPE:10.000000:2
2:SYSID_THISMAV:1.000000:4
3:SYSID_MYGCS:255.000000:4
4:SERIAL0_BAUD:115.000000:4
5:SERIAL1_BAUD:57.000000:4
6:SERIAL2_BAUD:57.000000:4
7:SERIAL2_PROTOCO:1.000000:2
8:TELEM_DELAY:0.000000:2
9:RTL_ALT:1500.000000:4
10:RNGFND_GAIN:0.800000:9
11:FS_BATT_ENABLE:0.000000:2
12:FS_BATT_VOLTAGE:0.000000:9

```

ภาพที่ 4-52 แสดง Text File ที่เกิดจากการเก็บ ข้อมูลลง SD card ของ Ground Station



```

@ COM10
=====
cmd_id : 1
Request Stop Data Stream from PGS
cmd_id : 2
Request Parameter List from PGS
Param 0 SYSID_SW_MREV:120.0000
Param 1 SYSID_SW_TYPE:10.0000
Param 2 SYSID_THISMAV:1.0000
Param 3 SYSID_MYGCS:255.0000
Param 4 SERIAL0_BAUD:115.0000
Param 5 SERIAL1_BAUD:57.0000
Param 6 TELEMETRY_DELAY:0.0000
Param 7 RTL_ALT:20.0000
Param 8 RNGFND_GAIN:0.8000
Param 9 FS_BATT_ENABLE:0.0000
Param 10 FS_BATT_VOLTAGE:0.0000
Param 11 FS_BATT_MAHR:0.0000
Param 12 FS_GPS_ENABLE:1.0000

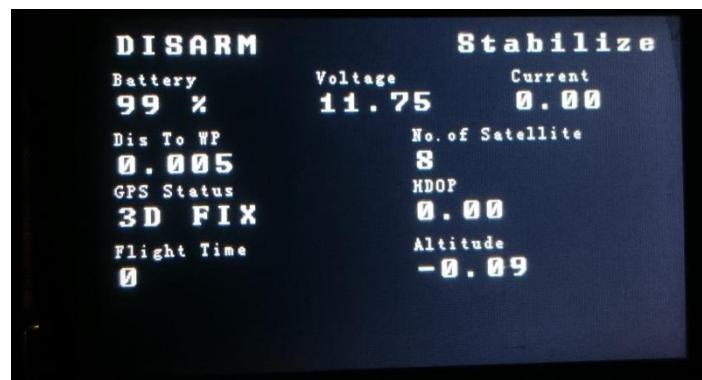
```

ภาพที่ 4-53 การส่งข้อมูลจาก Parameter มาจาก Mavlink Gateway

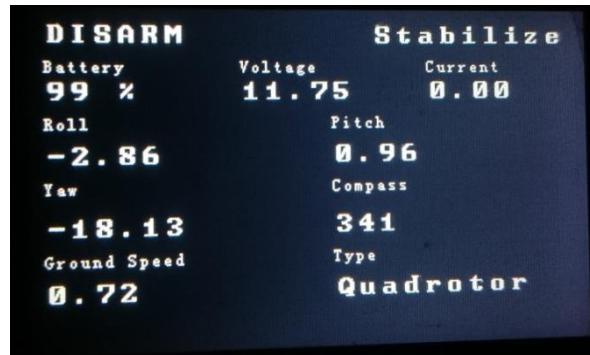
จากการทดลองการส่งค่า Parameter จาก Mavlink มาบัง Ground Station จากนั้นนำข้อมูลจากแต่ละแหล่งมาเปรียบเทียบกัน จึงสรุปได้ว่าการส่งค่า Parameter สามารถทำได้อย่างถูกต้อง

#### 4.3.3 การรับค่าเพื่อนำมาแสดงผลผ่านทางหน้าจอ GLCD

ในการแสดงผลด้วยหน้าจอ GLCD นี้จะเกิดขึ้นหลังจากที่ Ground Station นี้จะเกิดขึ้นหลังจากได้รับข้อมูล Parameter จาก Mavlink Gateway สำเร็จ โดยระหว่างทำการรับข้อมูล Parameter นี้จะมีหน้าต่าง Loading screen เพื่อเป็นการบอกถึงสถานะโดยการทดลองเป็นไปดังนี้



ภาพที่ 4-54 แสดงลิ๊งหน้าแสดงผลสถานะของลำหน้าที่ 1



ภาพที่ 4-55 แสดงผลของสถานะของมัลติโรเตอร์หน้าที่ 2

โดยที่หน้าแรกนี้จะมีข้อมูลดังนี้ ระบบทางระหว่างตัวมัลติโรเตอร์และตำแหน่งของ Ground Station, สถานะของ GPS, จำนวนดาวเทียมที่สามารถจับได้ ณ เวลานั้นๆ , HDOP, เวลาที่ใช้ในการบิน, ความสูง และในหน้าที่ 2 นั้นมีข้อมูลดังนี้ Roll , Pitch ,Yaw และเบ็มทิก , ความเร็วในแนวราบ , ชนิดของลำ และในส่วนด้านบนของทั้งสองหน้านั้นเป็นข้อมูลสำคัญที่ผู้ใช้งานควรจะทราบตลอดเวลา ดังนั้นจึงมีอยู่ในทุกหน้าซึ่งมีข้อมูลໄດ้แก่ สถานะของมัลติโรเตอร์ โหมดของการบิน แบบเตอร์ และกระແສไฟฟ้าที่ใช้งานอยู่ในขณะนั้น โดยที่ผู้ใช้งานสามารถปรับเปลี่ยนหน้าໄด้จากปุ่มกดบน push button terminal นั้นเอง

#### 4.3.4 ปรับเปลี่ยนค่า Parameter และส่งค่าการใช้งานกลับไปยัง Mavlink Gateway

```
COM18 (Arduino Due (Programming Port))

Adjusted parameter ID : 97 , Value :1500.000000 Complete
65535:RC8_TRIM:1500.000000:4
Start request streaming MSG
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.82 Roll : -1.39 Yaw : -122.69
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 11.53 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.82 Roll : -1.39 Yaw : -122.69
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 11.53 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
Case CMD 13 done
Case : 13 (EXTRA I) Pitch : -0.80 Roll : -1.44 Yaw : -122.88
Case CMD 14 done
Case : 14 (EXTRA II) Altitude : 12.08 Heading : 237 Ground speed : 0.00
Case 15 done
Case 15 (RC value) : 0 0 0 0 0 0 0 0
```

ภาพที่ 4-56 แสดงถึงการส่งค่า Parameter จาก Ground Station ไปยัง Mavlink Gateway

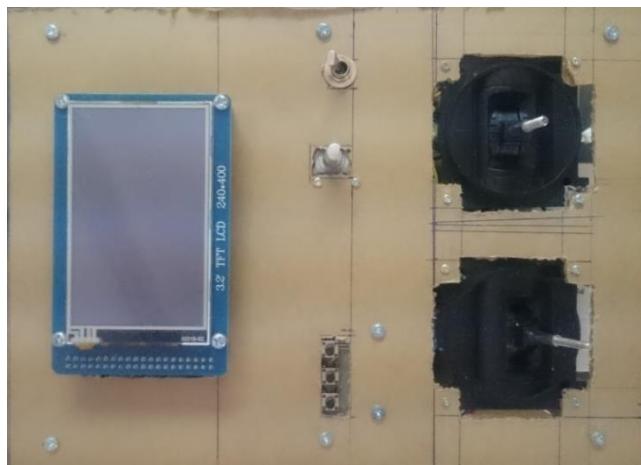
```

COM12 - PuTTY
[...]
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
cmd_id : 201
Req to write param 97:1500.00
Req to Param
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
97,RC8_TRIM,1500.0000,4
Upload Param
65535,RC8_TRIM,1500.0000,4
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1
MAVlink HeartBeat MAV: 2 Modes: Stabilize Armed: 0 SysId: 1 ComId: 1

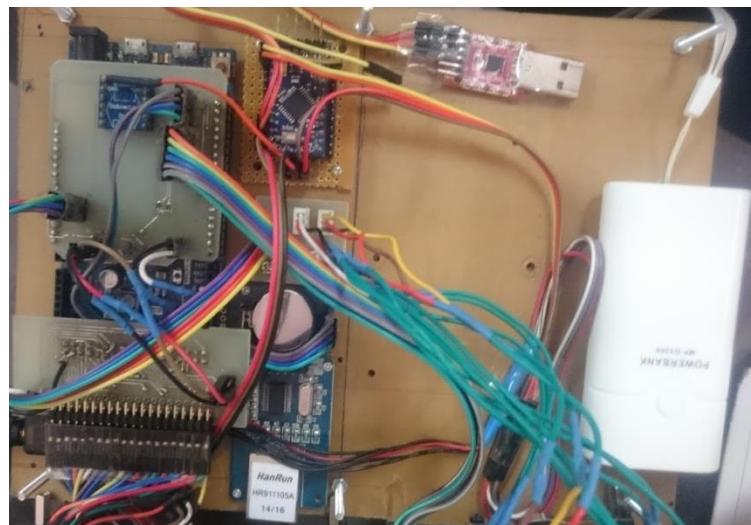
```

ภาพที่ 4-57 แสดงรับค่า Parameter จาก Mavlink Gateway ไปยัง Ground Station

จากการทดสอบระบบรวมข้างต้นนี้เป็นการยืนยันว่าระบบสามารถทำงานได้จริง จากนั้นจึงได้ทำการออกแบบกล่องใส่ชิ้นงาน Portable Ground Station โดยที่อุปกรณ์ที่ใช้ในการออกแบบชิ้นงานดังกล่าวคือ การนำแผ่นอะคิลิกมาทำการวัดขนาดและเจาะรูเพื่อทำการบรรจุ อุปกรณ์ต่าง ๆ ลงไปโดยที่มีแผ่นอะคิลิกจำนวนสองแผ่นเพื่อยึดด้านบนและล่าง โดยที่ใช้โดยที่ แผ่นอะคิลิกที่ใช้ Spacer และน็อตในการยึดระหว่างสองชั้นเข้าหากัน โดยที่แผ่นอะคิลิกมีขนาด 16.5 x 22.5 เซนติเมตร โดยที่ความกว้างระหว่างแผ่นอะคิลิกเท่ากับ 5.5 เซนติเมตร และใช้ Power Bank ขนาด 5200 mA เป็นแหล่งพลังงาน โดยที่สามารถแสดงเป็นชิ้นงานได้ดังนี้



ภาพที่ 4-58 แสดง Portable Ground Station จากด้านบน



ภาพที่ 4-59 แสดง Portable Ground Station จากค้านใน อะคิลิกแพ่นค้านล่าง



ภาพที่ 4-60 แสดง Portable Ground Station จากค้านใน อะคิลิกแพ่นค้านบน



ภาพที่ 4-61 แสดง Portable Ground Station จากด้านข้าง

หลังจากที่ขึ้นรูปชิ้นงานสำเร็จได้ทำการนำอุปกรณ์ไปทดสอบจริง โดยที่ทดสอบทั้งหมด 3 รูปแบบและทำการทดลองที่สนามฟุตบอลของมหาวิทยาลัย โดยที่เปลี่ยนการทดสอบได้ดังนี้

1. เริ่มต้นด้วยการนำ Portable Ground Station มาใช้ในการบังคับ มัลติโรเตอร์โดยที่สื่อสารไปในการทดสอบคือ เริ่มต้นจากการ Take off มัลติโรเตอร์ จากนั้นทำการบินขึ้นไปที่ความสูงที่คาดหวังคือ 15 เมตรหลังจากนั้นทำการรักษาระดับความสูงคงไว้ประมาณ 2 – 3 นาทีแล้วจึงนำมัลติโรเตอร์ร่วมทำการลงจอด
2. เริ่มต้นด้วยการนำ Portable Ground Station มาใช้ในการบังคับ มัลติโรเตอร์ในงาน Take off แล้วนำ มัลติโรเตอร์ไปที่ความสูงที่คาดหวังประมาณ 15 เมตร จากนั้นทำการสลับการควบคุมกลับมาที่ อุปกรณ์ควบคุมการบินแบบ 2.4 GHz เพื่อทำการรักษาระดับความสูงไว้ประมาณ 2 - 3 นาทีและทำการลงจอด
3. เริ่มจากการใช้อุปกรณ์ควบคุมการบินแบบ 2.4 GHz ใน การ Take off และนำขึ้นไปยังความสูง 15 เมตรแล้วจากนั้นทำการสลับควบคุมไปยัง Portable Ground Station เพื่อทำการรักษาระดับความสูงไว้ประมาณ 2 - 3 นาทีจากนั้นจึงทำการลงจอด

โดยที่การทดลองทั้ง 3 รูปแบบนี้ได้มีการป้องกันความเสียหายไว้แล้วนั้นคือการเตรียม Chanel ที่ 6 ของอุปกรณ์ควบคุมการบินแบบ 2.4 GHz ไว้ในกรณีที่สัญญาณหลุดหรือเกิดการสูญเสียการควบคุมสามารถถังการ Return to home Chanel ที่ 6 ของอุปกรณ์ควบคุมการบินแบบ 2.4 GHz ไว้ในนั้นเองและเพื่อให้การควบคุมเป็นไปได้โดยงานได้ทำการบินในโหมด รักษาระดับความสูง (Altitude hold)



ภาพที่ 4-62 แสดงการทดสอบทำการบินที่สถานที่จริง



ภาพที่ 4-63 แสดงถึงการบังคับระหว่าง Portable Ground Station และอุปกรณ์ควบคุมแบบ 2.4 GHz

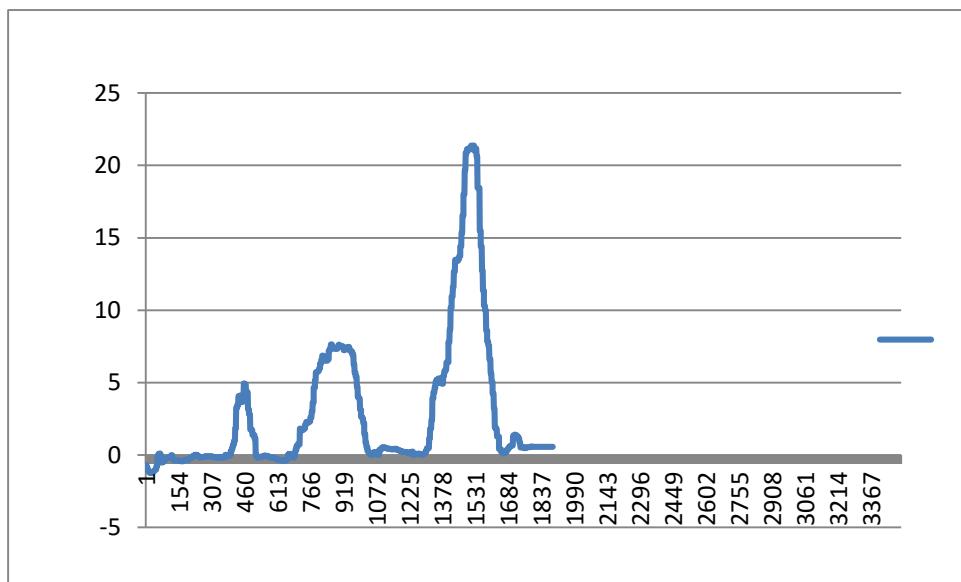


**ภาพที่ 4-64** ขณะทำการบินด้วย Portable Ground Station ที่ความสูงประมาณ 5-10 เมตร

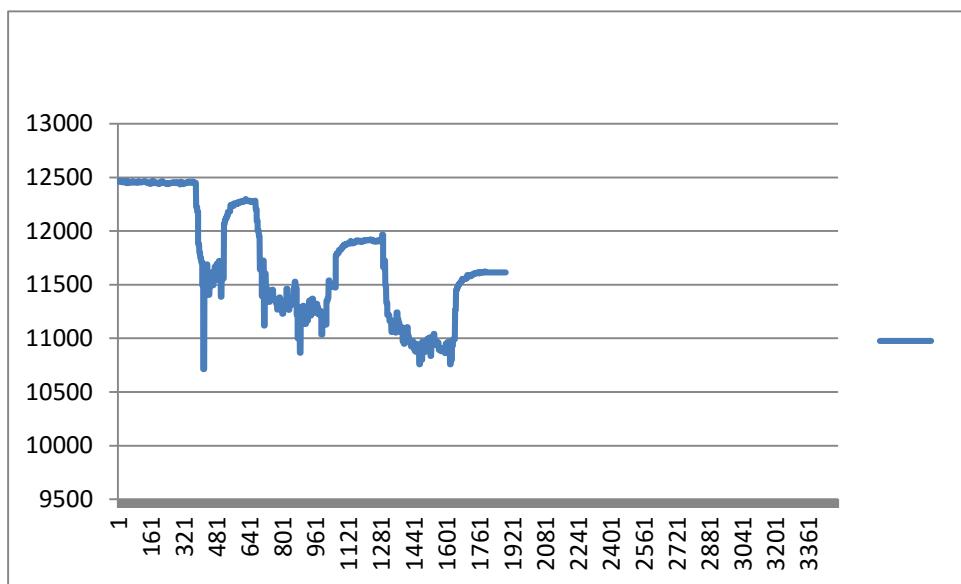


**ภาพที่ 4-65** ขณะทำการติดตั้งอุปกรณ์ก่อนทำการทดสอบ

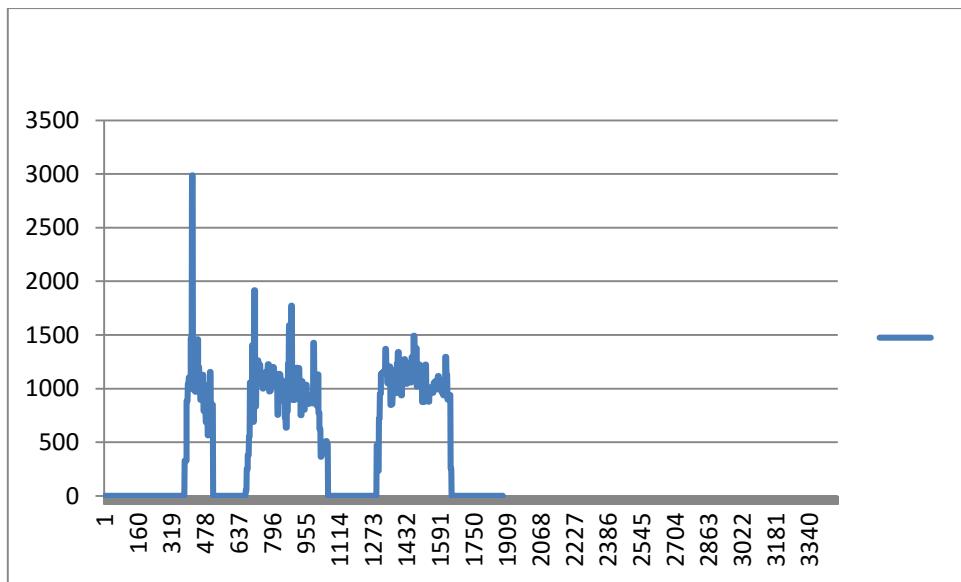
หลังจากทำการทดสอบสำเร็จได้นำข้อมูลที่ได้บันทึกจาก SD Card ของ Portable Ground Station มาแสดงผลด้วย Microsoft Excel โดยนำค่าที่ได้จากแต่ละคอลั่มน์มาแสดงผลด้วยการplot กราฟแบบ 2 มิติเพื่อให้สามารถเข้าใจข้อมูลได้ง่าย โดยทำการplotค่าที่ต้องการเทียบกับการจัดเก็บข้อมูลที่ความถี่ 5 Hz โดยกราฟที่นำมาแสดงได้แก่ ความสูง, ระดับแรงดันของแบตเตอรี่, การใช้งานของกระแส, จำนวนดาวเทียมที่สามารถรับค่าได้



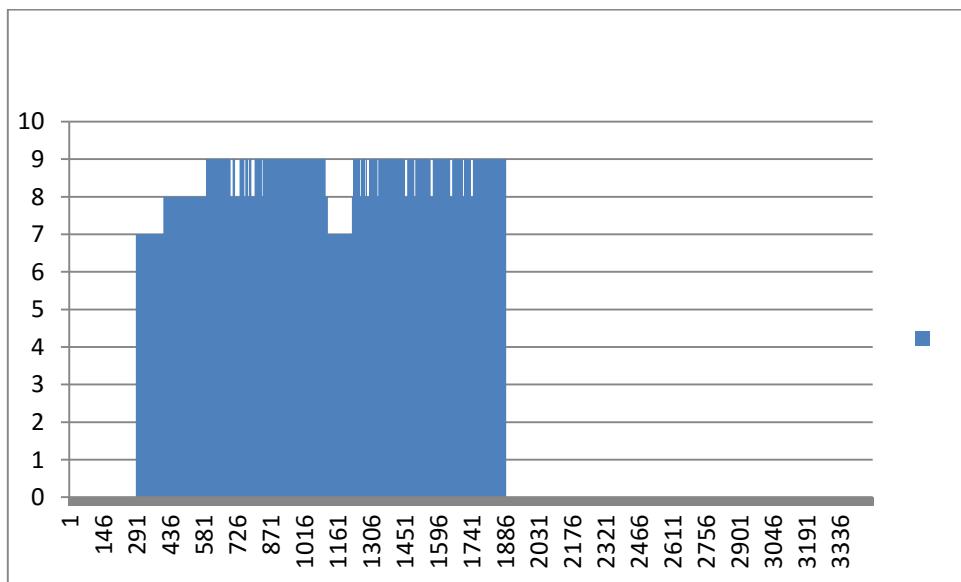
ภาพที่ 4-66 แสดงถึงความสูงของการบินในระยะเวลาต่างๆ



ภาพที่ 4-67 แสดงถึงระดับแรงดันของเบตเตอร์รี่ภายของมัลติโตร์เทียบกับเวลาในการบิน



ภาพที่ 4-68 แสดงถึงระดับการใช้งานของกระแสที่แบนตเดอร์ทำการจ่ายให้แก่เมอเตอร์ทั้งลี่ตัวในเวลาต่างๆ



ภาพที่ 4-69 แสดงถึงจำนวนดาวเทียมที่ระบบสามารถตรวจสอบได้ในเวลาต่างๆ

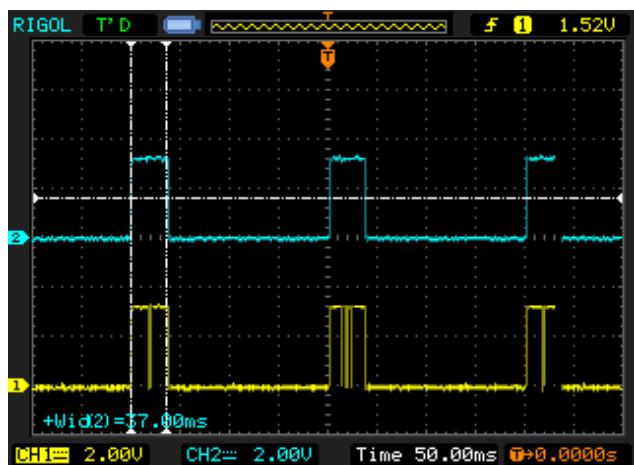
โดยที่ผลการทดสอบสามารถสรุปได้ดังนี้

1. การทดสอบสามารถทำตามเงื่อนไขที่ตั้งไว้ได้ตาม 3 ข้อด้านบน

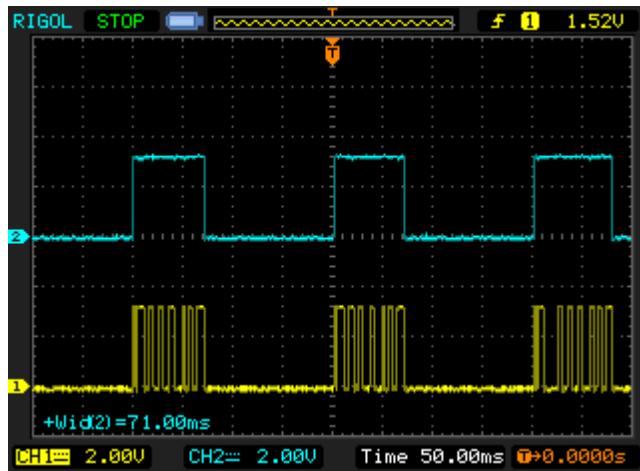
2. Portable Ground Station สามารถควบคุมมัลติโโรเตอร์ได้โดยให้ความรู้สึกเหมือนอุปกรณ์ควบคุมการบินแบบ 2.4 GHz โดยไม่สามารถสัมผัสได้ถึงความหน่วงจากการส่งข้อมูลในระยะประมาณ 15 - 20 เมตร
3. ในการควบคุมสามารถสลับการควบคุมระหว่าง Portable Ground Station และอุปกรณ์ควบคุมแบบ 2.4 GHz ได้อย่างสมบูรณ์ ก่อราก็ไม่มีการชะงักหรืออาการหน่วงของมาให้เห็น
4. ความน่าเชื่อถือของ อุปกรณ์ Portable Ground Station ที่ได้ทดลองอยู่ในระยะไม่เกิน 20 เมตร
5. ระบบบันทึกข้อมูลของ Portable Ground Station สามารถนำค่าที่ได้จากการบินมาตรวจสอบได้จริง

#### 4.4 ผลการทดสอบประสิทธิภาพของการแบ่งเวลาในการดำเนินงานของแต่ละ Task

เนื่องจากกระบวนการทำงานของ Arduino DUE นั้น ไม่ใช่ระบบปฏิบัติการเวลาจริง ดังนั้นการแบ่ง Task ของการทำงานในแต่ละ Task จึงมีความลำบากเนื่องจากจำเป็นต้องให้ทุก Task สามารถทำงานได้สำเร็จโดยที่ไม่ไปเป็นภาระของ Task อื่น ดังนั้นจึงได้ทำการวัดประสิทธิหรือเวลาในการทำงานของแต่ละ Task มาแสดงให้ดูดังภาพต่อไปนี้ โดยทำงานทดลองด้วยการรันซอฟต์แวร์สำหรับ Ground Station และเปิดทุกฟังก์ชันการทำงานและวัดค่าเวลาด้วย Logic analyzer และทำการวัดค่าเวลาในการทำงานของแต่ละ Task เพื่อทำการหาเวลาในการทำงานแต่ละ Task ที่ใช้น้อยที่สุดและมากที่สุดตามลำดับ



ภาพที่ 4-70 แสดงถึงเวลาที่ใช้ในการทำ Task Draw screen task 1 โดยใช้เวลาที่น้อยที่สุด



ภาพที่ 4-71 แสดงถึงเวลาที่ใช้ในการทำ Task Draw screen task 1 โดยใช้เวลามากที่สุด

หลังจากนั้นทำได้การคำนวณการทดลองจัดเวลาของทุกๆ Task ได้ผลออกมาดังตารางด้านล่าง

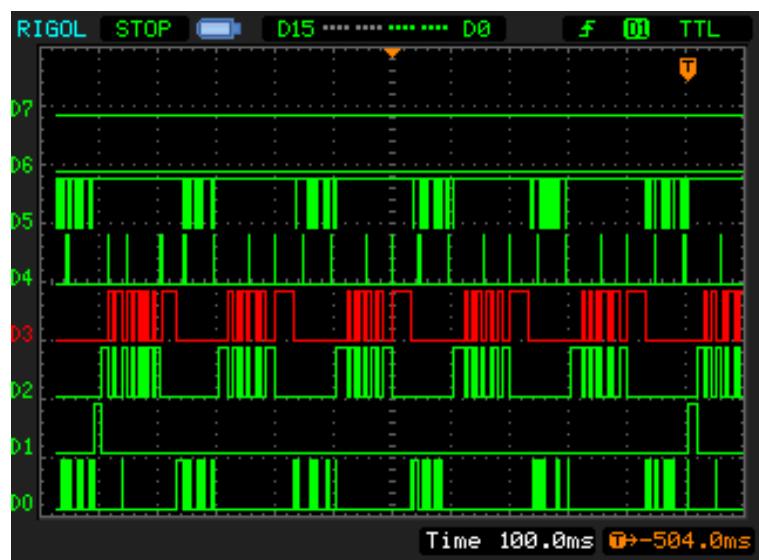
ตารางที่ 4-9 แสดงถึงเวลาที่ใช้ทำงานในแต่ละ Task

Tasks	Minimum time (ms)	Maximum time (ms)
Draw Screen 1	37	71
Draw Screen 2	65	93
RC override	1.4	1.4
Flush SD card	11	12
Write SD card	1	6.9
Receive Serial	1	20

จากตารางที่ 4-9 สามารถสรุปได้ว่าเวลาที่ใช้ในการคำนวณการหักหมดของระบบต่อหนึ่งรอบคือ ผลรวมของเวลาที่ใช้มากที่สุดของแต่ละ Task นั้นคือ 200 ms นั้นเอง



ภาพที่ 4-72 แสดงถึงเวลาที่แต่ละ Task ใช้ในการดำเนินการ



ภาพที่ 4-73 แสดงถึงเวลาที่แต่ละ Task ใช้ในการดำเนินการ

โดยที่สัญญาณแต่ละช่องคือ

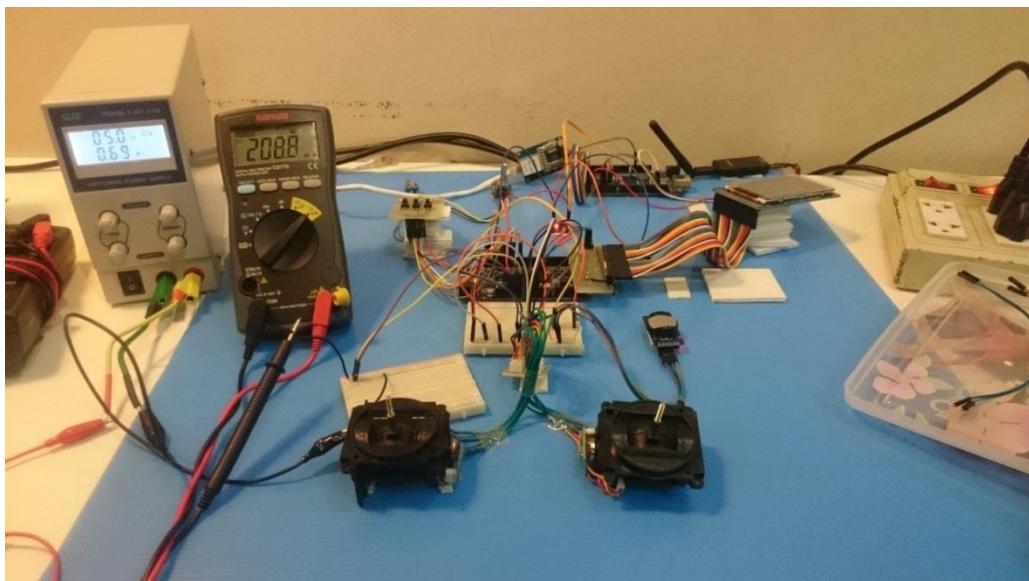
- D0 เวลาที่ใช้ในการรับและประมวลผล Serial
- D1 เวลาที่ใช้ในการ Flush SD card ด้วยความถี่ 1 Hz
- D2 เวลาที่ใช้ในการวัดของ Task ที่ 1 ความถี่ 5 Hz

- D3 เวลาที่ใช้ในการวัดขอ Task ที่ 2 ความถี่ 5 Hz
- D4 เวลาที่ใช้ในการส่ง Override RC ด้วยความถี่ 20 Hz
- D5 คือเวลาที่ใช้ในการทำงานของ Task ใหญ่ 1 รอบ

โดยสังเกตว่าแต่ละ Task สามารถแบ่งเวลาในการทำงานให้แก่กันได้หากมีการทำงานเกินเวลา ในส่วนของตนเอง และทุก Task สามารถทำงานตามความต้องการขั้นต่ำสุดของระบบได้

#### 4.5 การทดสอบอัตราการสิ้นเปลืองพลังงาน

หลังจากที่ระบบสามารถทำงานได้จริงแล้วต่อมาได้ทำการวัดกระแสหรืออัตราสิ้นเปลือง พลังงานของระบบ โดยที่ทำการวัด โดยเพิ่มอุปกรณ์ต่อพ่วงเข้าไปในระบบทีละชิ้นเพื่อดูอัตรา สิ้นเปลืองที่เพิ่มขึ้นและนำมาหาอัตราสิ้นเปลืองของอุปกรณ์แต่ละชิ้นได้ โดยที่ทำการวัดค่าของ กระแสที่ผ่าน Vin ของ Arduino DUE และผลการทดลองเป็นไปดังตารางได้ดัง



ภาพที่ 4-74 แสดงถึงการทดลองการวัดกระแสอุปกรณ์ชิ้นต่างๆของ Portable Ground Station

ตารางที่ 4-10 แสดงค่าของการวัดกระแสของระบบ โดยเพิ่มอุปกรณ์เข้าไปในระบบครั้งละหนึ่งชิ้น

Case	Arduino DUE	GCDmodule	Pushbutton terminal	Joy stick module	RTC	Current (ma)
Case 1	yes	-	-	-	-	130
Case 2	yes	yes	-	-	-	194.2
Case 3	yes	yes	yes	-	-	198
Case 4	yes	yes	yes	yes	-	202.2
Case 5	yes	yes	yes	yes	yes	208.8

หลังจากที่สามารถระบุอัตราสิ้นเปลืองรวมของพลังงานได้แล้วนั้นจึงนำค่าที่ได้กลับไปคิดเพื่อให้ได้อัตราสิ้นเปลืองของแต่ละอุปกรณ์ได้ดังตารางด้านล่าง

ตารางที่ 4-11 แสดงถึงอัตราสิ้นเปลืองของของแต่ละอุปกรณ์ของ Portable Ground Station

อุปกรณ์	อัตราการสิ้นเปลือง (ma)
Arduino DUE	130
GLCD module	64.2
Push button terminal	3.8
Joy stick module	4
RTC	5.8

## บทที่ 5

### สรุปผล ข้อเสนอแนะ และปัญหา

#### 5.1 สรุปผล

จากการทดลองและสร้างชิ้นงาน ระบบแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยาน ไร้คนขับแบบ 4 ใบพัด ร่วมกับ Mavlink Gateway นี้แสดงให้เห็นอย่างชัดเจนว่าอุปกรณ์สามารถช่วยเหลือผู้ใช้งานมัคติโรเตอร์ได้อย่างมีประสิทธิภาพ โดยที่อุปกรณ์สามารถช่วยตรวจสอบสถานะของมัคติโรเตอร์ และสามารถถูกข้อมูลของมัคติโรเตอร์ข้อนหลังเทียบกับเวลาจริงได้อีกด้วย อีกทั้งสามารถปรับค่าสถานะ ๆ ในเบื้องต้นได้หากเกิดปัญหา และยังมีรีโมทบังคับไว้ใช้ในยามฉุกเฉินได้อีกด้วย แต่ทั้งนี้ระบบยังมีข้อจำกัดอีกคือ

5.1.1 อุปกรณ์ไม่สามารถใช้งานได้หากไม่ได้เชื่อมต่อกับ Mavlink Gateway ระบบแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยาน ไร้คนขับแบบ 4 ใบพัดนี้จำเป็นต้องเชื่อมต่อกับ Mavlink Gateway อยู่ตลอดเวลา

5.1.2 นอกจากข้อมูลที่สำคัญแล้วระบบไม่สามารถแสดงข้อมูลปลีกย่อยได้ เพราะเนื่องจากขนาดของข้อมูลจำกัด และเลือกแสดงเฉพาะข้อมูลที่สำคัญเท่านั้น

5.1.3 ในการเปิดทุกครั้งจะใช้เวลานานเนื่องจากระบบต้องดึงค่า Parameter มาจากอุปกรณ์ควบคุมการบินมาก่อนทุกครั้งเพื่อกำหนดทิกแಡต์ข้อมูลที่ล่าสุดเสมอ

#### 5.2 ปัญหาที่พบในการดำเนินงาน

5.2.1 การใช้ขาติดต่อระหว่าง Arduino DUE และจอ GLCD บัษนาณ์แบบ 16-bit นี้เป็นไปอย่างยากลำบาก เพราะเนื่องจากผังการเชื่อมต่อของแต่ละอุปกรณ์นั้นไม่เหมือนกัน จึงต้องออกแบบอุปกรณ์ในการใช้งานเพิ่มเติม ซึ่งเสียเวลาในการดำเนินงานเป็นอย่างมาก

5.2.2 ในการสร้างซอฟต์แวร์ที่เกี่ยวข้องกับ task จำนวนมาก และไม่ได้ใช้ระบบปฏิบัติการเวลาจริงนี้มักพบปัญหา task ใด task ใช้เวลาในการทำงานมากเกินไป ซึ่งส่งผลให้เกิดรบกวนการทำงานของ task อื่น ๆ

#### 5.3 ข้อเสนอแนะ

5.3.1 ระบบแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยาน ไร้คนขับแบบ 4 ใบพัดนี้ควรที่จะสามารถแสดงค่าสถานะของมัคติโรเตอร์ได้จำนวนหลายลำพร้อมกัน

5.3.2 ข้อมูลที่เก็บควรจะสามารถนำไปเก็บในอินเตอร์เน็ตได้เพื่อความสะดวกใช้การใช้งานกับ  
คอมเพิล อินเตอร์เน็ตทุกหนแห่ง

5.3.3 เพิ่มฟังก์ชันของการใช้ประโยชน์จากอิหมากขึ้น อาทิเช่น การแสดงภาพสถานะของลำ  
แบบตามเวลาจริง

### ເອກສາຣອ້າງອີງ

1. Arduino Community. Arduino Mega Board. [Online].  
Available from : <http://www.arduino.cc/en/Main/arduinoBoardMega>
2. Arduino Community. Arduino Uno Board. [Online].  
Available from : <http://www.arduino.cc/en/Main/arduinoBoardUno>
3. Arduino Community. Arduino Due Board. [Online].  
Available from : <http://www.arduino.cc/en/Main/arduinoBoardDue>
4. ST Microelectronics. STM32F4 Discovery Board. [Online].  
Available from : <http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419>

## ประวัติผู้แต่ง

**ปริญญาบัตรเรื่อง** : ระบบแสดงผลภาคพื้นดินแบบพกพาสำหรับอากาศยานไร้คนขับแบบ 4 ใบพัด

**สาขาวิชา** : วิศวกรรมคอมพิวเตอร์

**ภาควิชา** : วิศวกรรมไฟฟ้าและคอมพิวเตอร์

**คณบ** : วิศวกรรมศาสตร์

**ชื่อ** : นายธนัช โต๊ะทอง

**ประวัติ**

เกิดวันที่ 20 สิงหาคม พ.ศ. 2535 อายุปีปัจจุบันเลขที่ 100/11 หมู่ 3 ตำบลปากเกร็ด อำเภอปากเกร็ด จังหวัดนนทบุรี สำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชา วิศวกรรมคอมพิวเตอร์ ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณบ วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2557