

โครงการวิศวกรรมคอมพิวเตอร์
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

เรื่อง

แพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติ
ของเครือข่ายไร้สายโดยใช้โดรน

Platform for Automatic Wireless Network Topology
Management Using Drone

โดย

ชื่อนิสิต นายพชร พึ่งทองหล่อ

5610502349

พ.ศ. 2559

แพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติ
ของเครือข่ายไร้สายโดยใช้โดรน
Platform for Automatic Wireless Network Topology Management Using Drone

โดย

นายพชร พึ่งทองหล่อ 5610502349

โครงการวิศวกรรมคอมพิวเตอร์
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

ตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
วิศวกรรมคอมพิวเตอร์

ได้รับการพิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษา วันที่ เดือน พ.ศ.

(รศ.ดร.อนันต์ ผลเพิ่ม)

..... วันที่ เดือน พ.ศ.

(ผศ.ดร.ชัยพร ใจแก้ว)

..... วันที่ เดือน พ.ศ.

(ผศ.ดร.อภิรักษ์ จันทร์สร้าง)

หัวหน้าภาควิชา วันที่ เดือน พ.ศ.

(รศ.ดร.อนันต์ ผลเพิ่ม)

นายพชร พึ่งทองหล่อ

ปีการศึกษา 2559

แพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรน

ปริญญาตรีวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์) ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

การประยุกต์ใช้โดรนเป็นตัวกลางในสร้างเครือข่ายการรับส่งข้อมูลแบบอัตโนมัติเพื่อเก็บข้อมูลในพื้นที่ที่เข้าถึงได้ยากหรือสถานแบบตายตัวที่มีความถี่ในการเก็บข้อมูลต่ำสามารถทำได้ง่ายขึ้นในงานวิจัยนี้ได้ทำการพัฒนาให้โดรนสามารถจัดเรียงตัวเพื่อสร้างเครือข่ายแบบอัตโนมัติตามเงื่อนไขของผู้ใช้ตามหลักการของ bi-connected graph และสามารถทราบสถานะของโดรนผ่านทางโปรแกรมประยุกต์บนสมาร์ตโฟนทำให้การเชื่อมต่อของเครือข่ายมีประสิทธิภาพมีความน่าเชื่อถือและรองรับการปรับเปลี่ยนเส้นทางใหม่ได้รวดเร็วยิ่งขึ้นในกรณีที่การเชื่อมต่อมีปัญหาเกิดขึ้น จากการทดสอบระบบที่ใช้โดรนจำนวน 3 ตัวซึ่งระยะระหว่างตัวประมาณ 60 เมตรพบว่าเมื่อเส้นทางเดิมไม่สามารถส่งข้อมูลได้รวมถึงการปรับเปลี่ยนเส้นทางใหม่ให้สามารถกลับมาส่งข้อมูลได้ใช้เวลาไม่เกิน 6 วินาทีและสามารถแสดงผลสถานะของโดรนผ่านทางโปรแกรมประยุกต์บนสมาร์ตโฟนได้อย่างถูกต้อง

คำสำคัญ โดรน เครือข่ายโดรน เครือข่ายแลนไร้สาย

Potchara Pheunghonglor

Academic Year 2016

Platform for Automatic Wireless Network Topology Management Using Drone

Bachelor Degree in Computer Engineering. Department of Computer Engineering.

Faculty of Engineering, Kasetsart University.

Abstract

The network of drones can be easily deployed for connecting the difficult to access or low data acquisition rate areas. In this project, the platform for automatic wireless network topology management using drone with bi-connected graph concept has been proposed. The goal is for creating the efficient and reliable network. The network status can be monitored via mobile application and can be used for path recovery process. From the experiment, 3 drones with 60 meters apart, the new data connection path can be recovered within 6 seconds after the failure occurs and can display status of drones via mobile application.

Keywords UAV, OLSR, Ad-hoc, Topology Control, Mesh Network

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความรู้และความช่วยเหลือจาก รศ.ดร.อนันต์ ผลเพิ่ม ผศ.ดร.ชัยพร ใจแก้วและผศ.ดร.อภิรักษ์ จันทร์สร้าง อาจารย์ที่ปรึกษาโครงการทั้งสามท่านที่ได้ให้คำแนะนำ ข้อเสนอแนะ ความรู้และแนวคิด ตลอดจนช่วยตรวจสอบข้อบกพร่องและแก้ไข ปัญหาต่างๆที่เกิดขึ้นระหว่างการทำโครงการมาโดยตลอด ผู้พัฒนาจึงขอกราบขอบพระคุณเป็นอย่างสูงและขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ที่ได้เอื้อเฟื้อสถานที่และอำนวยความสะดวกในการจัดหาอุปกรณ์สำหรับพัฒนาโครงการรวมถึงช่วยสนับสนุนทุนในการพัฒนาโครงการ

ขอขอบคุณ ดร.ไชยวัฒน์ กล้าพล อาจารย์ประจำห้องปฏิบัติการวิจัย ISAAC ภาควิชาวิศวกรรมการบินและอวกาศ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ที่ให้ความอนุเคราะห์อุปกรณ์ที่ใช้ในการจัดทำโครงการ

ขอขอบคุณผู้ที่ให้ความช่วยเหลือท่านอื่นๆทั้งสมาชิกห้องปฏิบัติการวิจัยเครือข่ายไร้สาย (Intelligent Wireless Network Group : IWING) ทุกท่านรวมถึงเพื่อนนิสิตและอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ที่ได้ให้ความช่วยเหลือรวมถึงครอบครัวของผู้พัฒนาโครงการที่ให้กำลังใจมาโดยตลอดจนโครงการนี้สำเร็จลุล่วงไปด้วยดี

นายพชร พึ่งทองหล่อ
ผู้จัดทำ

สารบัญ

บทคัดย่อ.....	i
Abstract.....	ii
กิตติกรรมประกาศ.....	iii
สารบัญภาพ.....	iv
สารบัญตาราง.....	v
1. บทนำ.....	1
1.1. วัตถุประสงค์ของโครงการ.....	2
1.2. ขอบเขตของโครงการ.....	2
1.3. ประโยชน์ที่ได้รับ.....	3
2. ทฤษฎีและงานที่เกี่ยวข้อง.....	4
2.1. ทฤษฎีที่เกี่ยวข้อง.....	4
2.2. งานที่เกี่ยวข้อง.....	7
3. เครื่องมือที่ใช้ในการทำโครงการ.....	8
3.1. ฮาร์ดแวร์.....	8
3.2. ระบบปฏิบัติการ.....	10
3.3. ภาษาที่ใช้ในการพัฒนา.....	10
3.4. ซอฟต์แวร์และไลบรารี.....	10
4. วิธีการดำเนินโครงการ.....	12
4.1. ภาพรวมของระบบ.....	12
4.2. รายละเอียดของระบบที่พัฒนา.....	14
4.3. องค์ประกอบโดยรวมของระบบ.....	15
4.4. ขั้นตอนการพัฒนา.....	17
5. ผลการดำเนินโครงการและวิจารณ์.....	18
5.1. สภาพแวดล้อมในการทดสอบ.....	18
5.2. ผลการทดสอบและวิจารณ์ผล.....	19
6. สรุปผลการดำเนินงานและข้อเสนอแนะ.....	22

6.1. ข้อสรุป.....	22
6.2. ปัญหาและอุปสรรค.....	22
6.3. แนวทางการพัฒนาต่อ.....	22
6.4. ข้อเสนอแนะ.....	23
7. บรรณานุกรม.....	24
8. ภาคผนวก.....	25
8.1. คู่มือการติดตั้ง.....	25
8.2. คู่มือการใช้งาน.....	31
9. ประวัติனிสีต.....	33

สารบัญภาพ

รูปที่ 2.1 Adhoc Network.....	5
รูปที่ 2.2 OLSR data flow diagram.....	5
รูปที่ 2.3 ตัวอย่าง UAV หรือ โดรน.....	6
รูปที่ 2.4 ระบบ GPS.....	6
รูปที่ 3.1 Banana Pi M1+.....	8
รูปที่ 3.2 Quadrotor Drone.....	9
รูปที่ 3.3 Futaba T8J Transmitter & Receiver.....	9
รูปที่ 4.1 ภาพรวมของระบบ.....	12
รูปที่ 4.2 CLI for station control.....	13
รูปที่ 4.3 องค์ประกอบโดยรวมของระบบ.....	15
รูปที่ 4.4 Quadrotor Drone.....	16
รูปที่ 5.1 การสร้างเครือข่ายของโดรน.....	18
รูปที่ 5.2 ผลการทดสอบความสัมพันธ์ของอัตราการรับส่งข้อมูลและเวลา.....	19
รูปที่ 5.3 สถานะการเชื่อมต่อของบอร์ดเพื่อใช้เป็นเส้นทางและเวลา.....	20
รูปที่ 5.4 สถานะของโดรนแสดงผ่านโมบายแอปพลิเคชันโดยการจัดเรียงตัวแบบเส้นตรง.....	20
รูปที่ 5.5 สถานะของโดรนแสดงผ่านโมบายแอปพลิเคชันโดยการจัดเรียงตัวแบบ redundant.....	21
รูปที่ 8.1 Icon ของ System Preferences และ Network.....	25
รูปที่ 8.2 การเพิ่ม network service.....	26
รูปที่ 8.3 การตั้งชื่อ network.....	26
รูปที่ 8.4 ปุ่มตั้งค่า network.....	27
รูปที่ 8.5 การตั้งค่า network IP และ Subnet Mask.....	27
รูปที่ 8.6 drone_network created.....	28
รูปที่ 8.6 Mobile Phone static IP setting.....	32

สารบัญตาราง

ตารางที่ 3.1 คุณสมบัติของ Banana Pi M1+.....	8
--	---

1 บทนำ

ปัจจุบันการรับส่งข้อมูลแบบไร้สายนั้นถูกใช้งานอย่างแพร่หลายโดยการส่งข้อมูลแบบไร้สายจะต้องทำการติดตั้งอุปกรณ์ตัวกลางเอาไว้ล่วงหน้าโดยจะมีตำแหน่งตายตัวไม่สามารถที่จะทำการเคลื่อนย้ายได้โดยบางครั้งต้องทำการติดตั้งหรือซ่อมแซมสถานีแบบตายตัวในพื้นที่ที่ยากต่อการเข้าถึงเช่น ภูเขา ป่าไม้ เป็นต้น ในกรณีฉุกเฉิน เช่น การเกิดภัยพิบัติต่างๆเช่น น้ำท่วม ทำให้การเข้าไปเก็บข้อมูลทำได้ยากหรือในกรณีที่ต้องการเก็บข้อมูลที่เก็บแล้วเว้นช่วงเวลายาวนานก่อนทำการเก็บครั้งต่อไปการสร้างสถานีแบบตายตัวจะไม่คุ้มค่าเนื่องจากต้องใช้ค่าใช้จ่ายสูง

จากปัญหาที่เกิดขึ้นทำให้เกิดแนวคิดในการแก้ไขปัญหาโดยใช้เทคโนโลยีการบินไร้คนขับขนาดเล็กหรือโดรนเข้ามาช่วยเป็นตัวกลางในการติดต่อสื่อสารในตำแหน่งที่เหมาะสมเพื่อสร้างเป็นระบบเครือข่ายไร้สายโดยโครงงานนี้เป็นโครงงานต่อเนื่องของ “โครงงานระบบเครือข่ายไร้สายโดยใช้โดรน” [1] ซึ่งพัฒนาโดย นายณัฐภูมิ เอื้อศักดิ์สุภา นิสิตคณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ โดยส่วนที่พัฒนาไปแล้วได้แก่การควบคุมโดรนผ่านคอนโทรลเลอร์โดยสามารถส่งงานโดรนจากสถานีต้นทางเพื่อบินไปยังตำแหน่งที่กำหนด ทำการสร้างเครือข่าย รวมไปถึงการรับส่งข้อมูลผ่านเครือข่ายไร้สาย แต่โครงงานดังกล่าวจำเป็นต้องใช้การควบคุมโดรนไปยังตำแหน่งที่ต้องการโดยส่งจากสถานีจึงจะสามารถส่งข้อมูลได้โดยยังไม่สามารถทำได้อย่างอัตโนมัติต่อมาโครงงานนี้ได้ถูกนำมาพัฒนาเป็น “โครงงานระบบเครือข่ายไร้สายโดยใช้โดรนอัตโนมัติ” [2] ซึ่งถูกพัฒนาโดย นายปริญญา เปียพนม นิสิตคณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์โดยส่วนที่ถูกพัฒนาได้แก่ การรับส่งข้อมูลผ่านเครือข่ายไร้สายโดยที่ควบคุมโดรนผ่านคอนโทรลเลอร์แบบอัตโนมัติโดยสามารถที่จะส่งงานโดรนจากสถานีต้นทางเพื่อให้โดรนทำการจัดเรียงตัวแบบเส้นตรงเพื่อทำการรับส่งข้อมูลระหว่างสถานีต้นทางและปลายทางแต่โครงงานดังกล่าวยังไม่สามารถที่จะทำการจัดเรียงตัวของโดรนตามเงื่อนไขของผู้ใช้และยังไม่มีแสดงผลสถานะของโดรนที่บินอยู่

ดังนั้นจึงได้คิดพัฒนาโครงงานต่อยอด โดยที่ผู้ใช้สามารถสั่งให้การจัดเรียงตัวของเครือข่ายโดรนคำนึงถึงปัจจัยต่างๆเช่น ทำให้การจัดเรียงตัวคำนึงถึงความต่อเนื่องในการเชื่อมต่อหรือลดเวลาในการฟื้นคืนสภาพของการเชื่อมต่อให้น้อยที่สุดเพื่อรับส่งข้อมูลรวมไปถึงสามารถแสดงผลสถานะต่างๆของโดรนเช่น ตำแหน่งเส้นทางที่เชื่อมต่อ และเส้นทางที่ใช้ในการรับส่งข้อมูลของโดรน สถานีต้นทางและสถานีปลายทางผ่านทางโมบายแอปพลิเคชันทำให้ผู้ใช้งานรับรู้ถึงสถานะของโดรนในขณะนั้นหรือปัญหาที่เกิดขึ้นทำให้การใช้โดรนเข้ามาเป็นตัวกลางสามารถที่จะเพิ่มความยืดหยุ่นและความน่าเชื่อถือของการรับส่งข้อมูลได้

1.1 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างเครือข่ายเน็ตเวิร์คโดยใช้โดรนเป็นตัวกลางในการรับส่งข้อมูลในสถานที่ที่เข้าถึงยาก
2. เพื่อลดค่าใช้จ่ายที่จะต้องทำการติดตั้งหรือซ่อมแซมสถานีแบบตายตัวซึ่งการเข้าถึงทำได้ยาก
3. เพื่อที่จะให้ผู้ใช้สามารถควบคุมการสร้างเครือข่ายโดยมีเงื่อนไขตามความต้องการ
4. เพื่อแก้ไขปัญหาที่เกิดจากความไม่ยืดหยุ่นของสถานีรับส่งข้อมูลแบบตายตัว
5. เพื่อที่จะลดช่วงเวลาที่ไม่สามารถรับส่งข้อมูลได้เมื่อโดรนมีเหตุขัดข้อง
6. เพื่อที่จะเป็นพื้นฐานของแนวคิดการต่อยอดการหาเส้นทางสื่อสารข้อมูลแบบไร้สายบนโดรนที่มีความยืดหยุ่นของการรับส่งข้อมูล
7. เพื่อเป็นพื้นฐานของระบบควบคุมอัตโนมัติบนโดรนโดยการใช้อัลกอริทึมต่างๆเพื่อให้โดรนสามารถทำงานได้ตามเงื่อนไขต่างๆที่ระบุโดยผู้ใช้
8. เพื่อแสดงสถานะต่างๆของโดรนผ่านทางโปรแกรมประยุกต์บนสมาร์ตโฟน

1.2 ขอบเขตของโครงการ

การพัฒนาแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนมีขอบเขตและข้อจำกัดในการพัฒนาดังต่อไปนี้

1.2.1 ขอบเขตของโครงการ

- โดรนสามารถที่จะบินไปยังเป้าหมายเพื่อที่จะสร้างเครือข่ายได้อย่างอัตโนมัติโดยที่สามารถที่จะทำตามเงื่อนไขที่ผู้ใช้ระบุ
- สามารถแสดงผลตำแหน่งของโดรนรวมถึงค่าสถานะต่างๆเช่น ตำแหน่ง เส้นทาง การเชื่อมต่อ เป็นต้น ผ่านทางโมบายแอปพลิเคชัน
- โดรนสามารถที่จะรับส่งข้อมูลจากต้นทางและปลายทางได้อย่างถูกต้อง

1.2.2 ข้อจำกัดของโครงการ

- การควบคุมโดรนใช้ระบบปฏิบัติการ Linux
- ประสิทธิภาพในการทำงานของโดรนขึ้นอยู่กับหลายปัจจัยเช่น สัญญาณรบกวน และสภาพแวดล้อมของสถานที่
- ในการนำโดรนขึ้นบินจำเป็นต้องมีนักบินควบคุมการบินด้วยรีโมตหากมีเหตุขัดข้องจะสามารถควบคุมด้วยรีโมตเพื่อนำโดรนบินกลับมามากอย่างปลอดภัย
- เนื่องจากแบตเตอรี่บนโดรนสามารถบรรจุพลังงานได้จำกัดรวมถึงโดรนใช้พลังงานอย่างมากในการบินการควบคุมโดรนทำได้ในเวลาจำกัดซึ่งการออกแบบให้มีการจัดการอัตโนมัติและการประมวลผลมากขึ้นอาจส่งผลให้เวลาที่ใช้ในการบินเหลือน้อยลง

1.3 ประโยชน์ที่ได้รับ

1. เนื่องจากในพื้นที่ที่เข้าถึงได้ยากหรือสถานที่ที่มีความถี่ในการเก็บข้อมูลต่ำไม่มีความจำเป็นในการติดตั้งสถานีรับส่งข้อมูลแบบตายตัว
2. สามารถใช้งานชั่วคราวในกรณีที่มีสถานีแบบตายตัวเสียหาย
3. การทราบถึงสถานะต่างๆของโดรนเช่น ตำแหน่ง พลังงาน สามารถทราบรูปแบบการสร้างเครือข่ายและปัญหาที่เกิดขึ้น
4. การสร้างเครือข่ายเพื่อรับส่งข้อมูลแบบอัตโนมัติสามารถที่จะรับส่งข้อมูลได้แม้ว่าจะมีโดรนที่เกิดปัญหา
5. การจัดเรียงตัวของโดรนโดยคำนึงถึงเงื่อนไขต่างๆจะสามารถช่วยให้แก้ไขปัญหาเฉพาะหน้าได้ดีขึ้น

2 ทฤษฎีและงานที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

- IEEE 802.11

เป็นมาตรฐานที่ถูกกำหนดมาให้ใช้กับเครือข่ายแลนไร้สายโดยจะทำการส่งข้อมูลบนคลื่นวิทยุที่ย่านความถี่ 2.4 Ghz และ 5 Ghz โดย Institute of Electrical and Electronics Engineers หรือ IEEE เป็นองค์กรที่กำหนดมาตรฐานการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ซึ่งได้กำหนดมาตรฐานสำหรับเครือข่ายไวร์เลสแลนขึ้นคือมาตรฐาน IEEE802.11 ซึ่งถูกนำมาใช้กับการรับส่งข้อมูลบน โดรนโดยมีมาตรฐานย่อยคือ a, b, g และ n ได้แก่

- IEEE 802.11a

ทำงานที่ย่านความถี่ 5 GHz ความเร็วในการรับส่งข้อมูล 54 Mbps ใช้กับการรับส่งข้อมูลที่มีความละเอียดสูงได้ เช่นการ stream วิดีโอ สามารถปรับความเร็วให้ลดลงเพื่อเพิ่มระยะทางในการเชื่อมต่อ

- IEEE 802.11b

ทำงานที่คลื่นความถี่ 2.4 GHz ความเร็วในการรับส่งข้อมูล 11 Mbps และมีการเข้ารหัสข้อมูลแบบ WEP 128-bit โดยมาตรฐานนี้เป็นที่นิยมในการใช้อย่างแพร่หลาย

- IEEE 802.11g

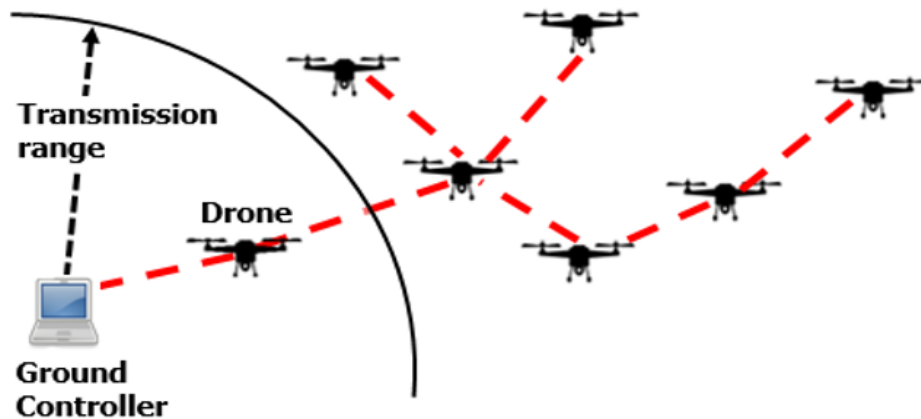
ทำงานที่คลื่นความถี่ 2.4 GHz ความเร็วในการรับส่งข้อมูล 36-54 Mbps โดยเป็นมาตรฐานที่นิยมใช้เป็นจำนวนมากและดีกว่ามาตรฐานแบบ IEEE 802.11b

- IEEE 802.11n

ทำงานที่คลื่นความถี่ 2.4 GHz และ 5 GHz ความเร็วในการรับส่งข้อมูล 300-450 Mbps โดยบนอุปกรณ์กระจายสัญญาณมีเสาสัญญาณ 2-4 เสาและสามารถใช้งานร่วมกับมาตรฐาน IEEE 802.11b และ IEEE 802.11

- Ad hoc

เป็นเน็ตเวิร์คที่ประกอบขึ้นจากอุปกรณ์คำนวณที่สามารถเคลื่อนย้ายได้เช่นบอร์ดที่ควบคุมโดรน ดังรูปที่ 2.1 เป็นต้น โดยเป็นเน็ตเวิร์คแบบชั่วคราวที่สามารถสร้างขึ้นได้อย่างรวดเร็วหรือทันทีโดยไม่มีการจัดการศูนย์กลางแต่ละ node สามารถเป็นได้ทั้ง host และ router นอกจากนี้ ยังไม่มีโครงสร้างพื้นฐานที่ยึดติดตายตัวทำให้ช่วยลดค่าใช้จ่ายลงไปได้อย่างมาก

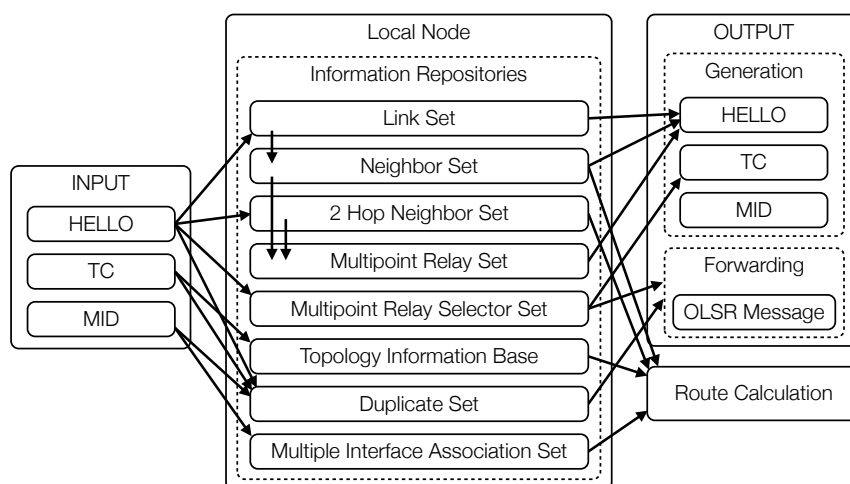


รูปที่ 2.1 Adhoc network

[ที่มา : <https://www.semanticscholar.org/paper/Multi-drone-control-and-network-self-recovery-for-Kim-Nam/94792f427ba01db12fc2f7b139983c495666e858/figure/0>]

- OLSR Protocol

OLSR (Optimized Link State Routing Protocol) เป็นโพรโทคอลที่ใช้ในการแลกเปลี่ยนตารางข้อมูลของเส้นทางการส่งผ่านข้อมูลระหว่างอุปกรณ์เครือข่ายต่างๆ ที่ทำงานในระดับของเน็ตเวิร์คเลเยอร์โดยเหมาะกับ Ad hoc Network และเครือข่ายไร้สายอื่นๆ โดยจะมีความเสถียรและควบคุมการเชื่อมต่อกับโหนดที่อยู่ใกล้เคียงและมี diagram การทำงานดังแสดงในรูปที่ 2.2



รูปที่ 2.2 OLSR data flow diagram

[ที่มา : <https://upload.wikimedia.org/wikipedia/commons/3/36/Olsr-overview.pdf>]

- UAV (drone)

เทคโนโลยีการบินไร้คนขับตามหลักแล้วหมายถึงโดรนซึ่งเป็นอากาศยานที่ควบคุมจากระยะไกลแบ่งเป็น 2 แบบคือแบบควบคุมจากระยะไกลและแบบที่สามารถบินได้เองอัตโนมัติที่ใช้โปรแกรมคอมพิวเตอร์ที่ซับซ้อนติดตั้งเอาไว้ดังแสดงตัวอย่างของโดรนในรูปที่ 2.3 โดยการใช้โดรนนั้นทำได้สะดวก ประหยัดค่าใช้จ่ายในเรื่องค่าเช่านักบินและเครื่องบินอีกทั้งสามารถใช้ประโยชน์ได้ในหลายทางเช่น การติดกล้องเพื่อสังเกตการณ์ ถ่ายรูปในบริเวณต่างๆหรือแม้กระทั่งใช้ในการกิจโจมตีของทหาร เป็นต้น



รูปที่ 2.3 ตัวอย่าง UAV หรือ โดรน

[ที่มา : <http://www.dronesglobe.com/wp-content/uploads/2015/09/xiro-xplorer-v-review.jpg>]

- GPS

ใช้เพื่อระบุข้อมูลของตำแหน่งและเวลาโดยอาศัยการคำนวณจากความถี่ของสัญญาณนาฬิกาที่ส่งมาจากดาวเทียมต่างๆที่โคจรรอบโลกดังแสดงในรูปที่ 2.4 ทำให้สามารถระบุตำแหน่งบนจุดต่างๆที่รับสัญญาณได้ทั่วโลกและทุกสภาพอากาศ



รูปที่ 2.4 ระบบ GPS

[ที่มา : <http://www.grandprix.co.th/wp-content/uploads/2016/04/gps5.jpg>]

2.2 งานที่เกี่ยวข้อง

- กพร.รูกใช้ UAV บินดูแลเหมืองแร่ นำร่องพื้นที่เหมืองหินปูน-แร่ยิปซัม [10]

กรมอุตสาหกรรมพื้นฐานและการเหมืองแร่ใช้เทคโนโลยีอากาศยานไร้คนขับในงานตรวจสอบและกำกับดูแลเหมืองแร่เพื่อเพิ่มประสิทธิภาพการทำงานลดความเสี่ยงในจุดอันตราย ลดต้นทุนการตรวจสอบและกำกับดูแลเหมืองแร่ นำร่องพื้นที่เหมืองหินปูน-ยิปซัม

- ระบบเครือข่ายไร้สายโดยใช้โดรน [1]

โครงการประยุกต์ใช้โดรนเพื่อสร้างเครือข่ายซึ่งถูกพัฒนาโดยนายณัฐวุฒิ เอื้อศักดิ์สุภา นิสิตคณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ โดยเนื้อหาของได้แก่การควบคุมโดรนผ่านคอนโทรลเลอร์โดยสามารถสั่งงานโดรนจากสถานีต้นทางเพื่อบินไปยังตำแหน่งที่กำหนด ทำการสร้างเครือข่ายรวมไปถึงการรับส่งข้อมูลผ่านเครือข่ายไร้สาย

- ระบบเครือข่ายไร้สายโดยใช้โดรนอัตโนมัติ [2]

โครงการประยุกต์ใช้โดรนเพื่อสร้างเครือข่ายแบบอัตโนมัติซึ่งถูกพัฒนาโดยนายปริญญา เปียพนม นิสิตคณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ โดยส่วนที่ถูกพัฒนาได้แก่การรับส่งข้อมูลผ่านเครือข่ายไร้สายโดยที่ควบคุมโดรนผ่านคอนโทรลเลอร์แบบอัตโนมัติโดยสามารถที่จะสั่งงานโดรนจากสถานีต้นทางเพื่อให้โดรนทำการจัดเรียงตัวแบบเส้นตรงเพื่อทำการรับส่งข้อมูลระหว่างสถานีต้นทางและปลายทาง

3 เครื่องมือที่ใช้ในการทำโครงงาน

3.1 ฮาร์ดแวร์

- Banana Pi M1+

บอร์ดคอมพิวเตอร์ที่ใช้บนตัวโดรนดังรูปที่ 3.1 ทำหน้าที่คำนวณตำแหน่งในการจัดเรียงตัวของเครื่องข่ายโดรนรวมไปถึงการรับส่งข้อมูลระหว่างโดรนกับสถานีต้นทางหรือสถานีปลายทางและโดรนตัวอื่นๆ อีกทั้งยังเป็นตัวเซิร์ฟเวอร์ที่ส่งข้อมูลสถานะของโดรนมายังโมบายแอปพลิเคชันบนสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ iOS และมีคุณสมบัติดังตารางที่ 3.1



รูปที่ 3.1 Banana Pi M1+

[ที่มา : <http://www.banana-pi.org/images/bpi-images/M1+/plus3.jpg>]

ตารางที่ 3.1 คุณสมบัติของ Banana Pi M1+

CPU	A20 ARM Cortex-A7™ Dual-core
GPU	ARM Mali-400 MP2 with Open GL ES 2.0/1.1
Memory	1 GB DDR3 SDRAM
Storage Support	SD Card(up to 64GB)/SATA(up to 2TB)
Onboard Network	10/100/1000 Ethernet RJ45
Wifi	WiFi b/g/n
USB Ports	2x USB 2.0
Dimensions	92mm x 60mm
Weight	48g

- **Quadrotor Drone**

อากาศยานไร้คนขับ (UAV) แบบ 4 ใบพัดที่ประกอบไปด้วย Auto pilot ที่ใช้ในการสั่งการโดรนรวมไปถึงการขอข้อมูลสถานะต่างๆของโดรนตัวอ่านค่า GPS รวมไปถึง บารอมิเตอร์ที่ใช้ในการวัดระดับความสูงและการลงจอด ดังรูปที่ 3.2



รูปที่ 3.2 Quadrotor Drone

- **Futaba T8J (Remote controller for drone)**

อุปกรณ์รีโมทที่ใช้ในการควบคุมโดรน ในทุกครั้งที่บินจะต้องมีการควบคุมด้วยรีโมทเพื่อป้องกันกรณีที่เกิดปัญหาจะสามารถนำโดรนกลับมาได้อย่างปลอดภัย



รูปที่ 3.3 Futaba T8J Transmitter & Receiver

[ที่มา : http://www.surabayahobby.com/images/big/2103003110_424_10_1881536788.jpg]

3.2 ระบบปฏิบัติการ

- Mac OS X version 10.12.4

ระบบปฏิบัติการยูนิกซ์ที่ใช้เพื่อพัฒนาโปรแกรมที่ใช้ในการควบคุมโดรนและใช้ในการทดสอบระบบที่พัฒนาโดรน

- Bananian Linux

ระบบปฏิบัติการลินุกซ์ที่ติดตั้งบน Banana Pi M1+ ใช้ในการสั่งการควบคุมโดรนและฟังก์ชันต่างๆที่เกี่ยวข้องกับโดรน

3.3 ภาษาที่ใช้ในการพัฒนา

- Python

ใช้สำหรับเขียนเพื่อพัฒนาโปรแกรมที่ตัวโดรนแต่ละตัวเพื่อใช้ในการสื่อสารและจัดการโครงข่ายของโดรน (พัฒนาบน Banana Pi M1+) รวมไปถึงการคำนวณต่างๆที่ใช้บนโดรน

- JavaScript

ใช้สำหรับเขียนเพื่อพัฒนาส่วนของโมบายแอปพลิเคชัน

3.4 ซอฟต์แวร์และไลบรารี

- OLSR Library

เป็นไลบรารีที่ใช้เพื่อค้นหาเส้นทางการส่งข้อมูลในเครือข่าย adhoc ของโดรน

- Mavproxy

เป็นโปรแกรมที่ใช้ในการติดต่อกับ Autopilot ของโดรน ใช้ในการสั่งการและควบคุมโดรน เช่น สั่งให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการ กำหนดความสูงในการบิน รวมไปถึงค่าสถานะต่างๆของโดรนเช่น พลังงาน ตำแหน่ง เป็นต้น

- Iperf3

ใช้ในการทดสอบแบนด์วิดท์ของการรับส่งข้อมูลจากสถานีดันทางไปยังสถานีปลายทางผ่านโดรน

- **React Native**

เป็นเฟรมเวิร์คของ ReactJS ที่ใช้ในการพัฒนาโมบายแอปพลิเคชัน

- **ExpressJS**

ใช้เพื่อพัฒนาส่วนของ API ในการจัดการกับข้อมูลต่างๆที่อยู่บนโดรน

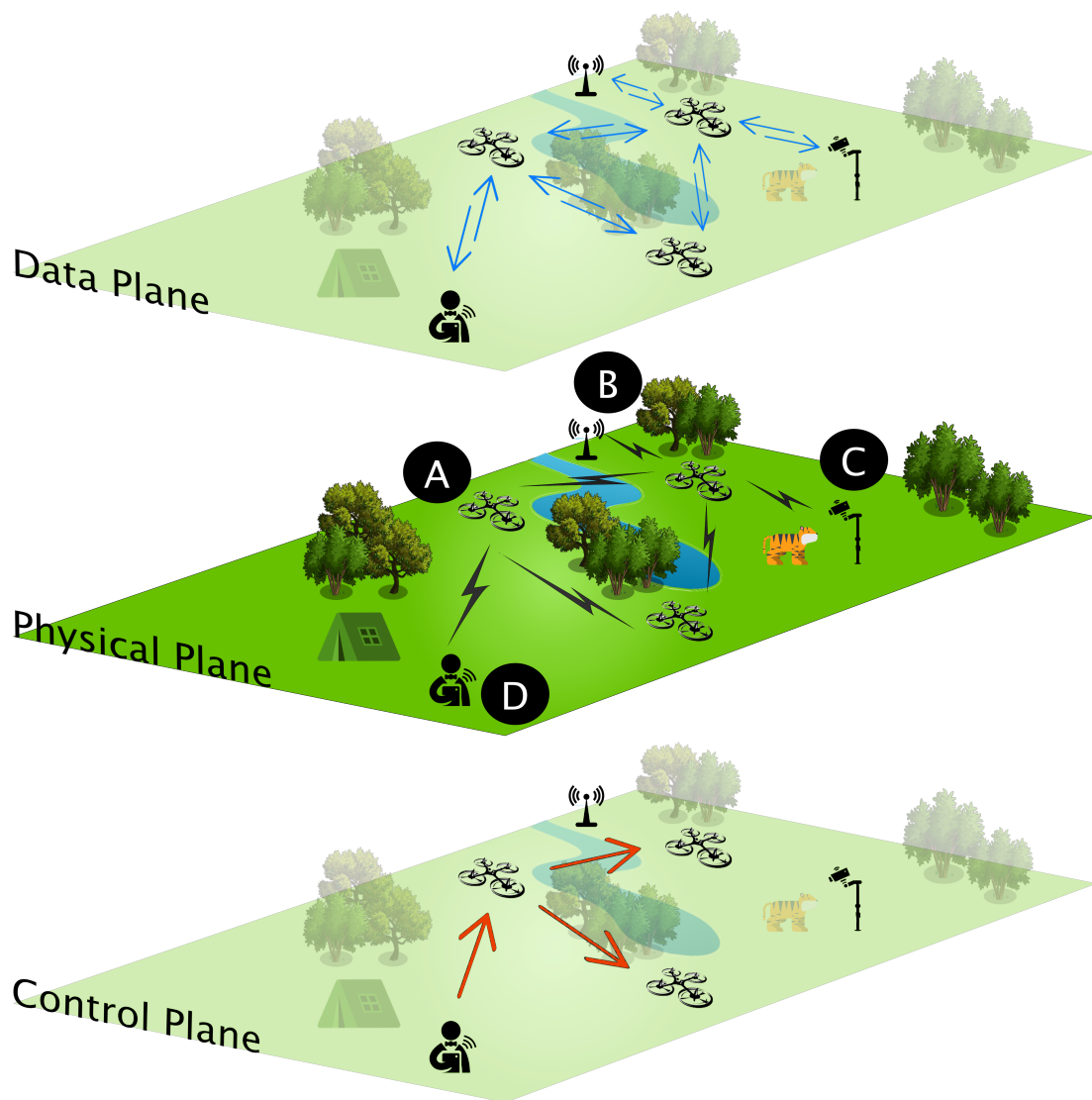
- **Xcode**

ใช้ในการทดสอบโมบายแอปพลิเคชันบนสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ iOS

4 วิธีการดำเนินการ

4.1 ภาพรวมของระบบ

ภาพรวมของระบบแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนแสดงได้ดังรูปที่ 4.1 โดยระบบเครือข่ายไร้สายโดยใช้โดรน (A) เพื่อที่จะสามารถรับส่งข้อมูลผ่านโดรนจากสถานีปลายทาง (B) หรืออุปกรณ์ต่างๆเช่นกล้อง (C) มายังผู้ใช้ผ่านโดรนที่จัดเรียงตัวเพื่อสร้างเป็นเครือข่ายเพื่อรับส่งข้อมูลแบบอย่างอัตโนมัติและกำหนดเงื่อนไขในการจัดเรียงตัวได้รวมไปถึงการแสดงผลสถานะต่างๆของโดรน (D) จะช่วยให้ทราบค่าต่างๆเช่นตำแหน่งและพลังงาน เป็นต้น



รูปที่ 4.1 ภาพรวมของระบบ

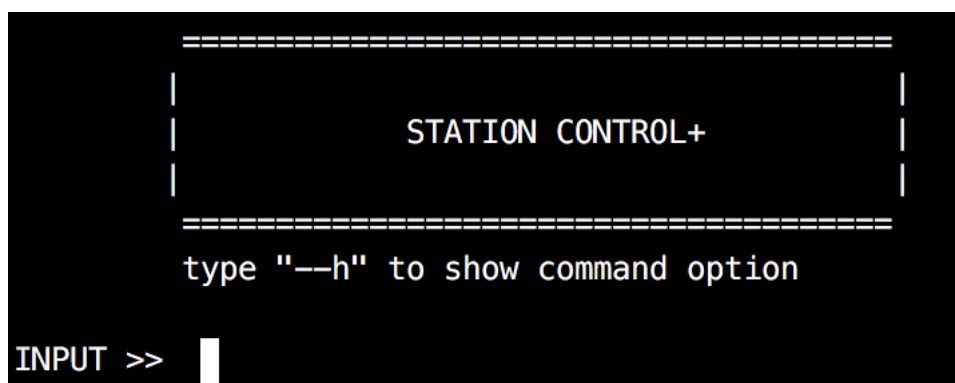
โดยระบบแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนสามารถแบ่งออกเป็น 3 ส่วนหลักได้แก่

1. ส่วนของผู้ใช้งาน

เป็นส่วนที่ประกอบไปด้วย 2 ส่วนย่อยได้แก่

1.1 User Control

เป็นส่วนที่ผู้ใช้สามารถควบคุมโดรนผ่าน User Interface ซึ่งเป็น Command Line Interface โดยใช้ระบบปฏิบัติการ unix ดังรูปที่ 4.2 โดยจะสามารถกำหนดค่าพื้นฐานต่างๆให้แก่โดรนเช่น ตำแหน่งของสถานีต้นทาง ตำแหน่งของสถานีปลายทาง วิธีในการจัดเรียงตัวของโดรน เป็นต้น



รูปที่ 4.2 CLI for station control

1.2 Monitoring via Mobile Application

เป็นส่วนที่ใช้แสดงผลสถานะต่างๆของโดรนได้แก่ ตำแหน่งของโดรน เส้นทางที่ใช้ในการรับส่งข้อมูล เส้นทางที่มีการเชื่อมต่อ ระหว่างโดรน สถานีต้นทางและสถานีปลายทาง

2. Wireless Mesh Network via Drones

เป็นส่วนที่เกิดการจัดรูปแบบของโดรนตามเงื่อนไขที่ผู้ใช้กำหนดและเป็นส่วนที่มีการสร้างเครือข่ายเพื่อรับส่งข้อมูลระหว่างโดรน สถานีต้นทางและสถานีปลายทาง โดยจะใช้ โพรโทคอล OLSR ในการเลือกเส้นทางที่ใช้ส่งข้อมูล

3. Destination node

เป็นส่วนของสถานีปลายทางที่มีได้หลากหลายอย่างเช่น เซ็นเซอร์ ที่มีการเก็บข้อมูลต่างๆเช่นค่าอุณหภูมิหรือสถานีเก็บข้อมูลที่อยู่ห่างไกลและเข้าถึงได้ยากและกล้องที่ทำการเก็บ Video Streaming เป็นต้นโดยจะทำการรับส่งข้อมูลต่างๆผ่าน Wireless Mesh Network ของโดรนกับสถานีต้นทาง

จากภาพรวมของระบบสามารถแบ่งการทำงานออกเป็น 3 ลำดับชั้น ได้แก่

1. Data Plane

เป็นเลเยอร์ที่เกี่ยวข้องกับการรับส่งข้อมูลจากสถานีต้นทาง โดรนและสถานีปลายทางโดยจะใช้โปรโตคอล OLSR สำหรับการค้นหาเส้นทางที่จะใช้รับส่งข้อมูลโดยเส้นทางที่ใช้จะถูกแสดงผ่านโมบายแอปพลิเคชันด้วยเส้นทึบและเส้นทางที่มีการเชื่อมต่อแต่ยังไม่ได้ใช้เป็นเส้นทางในการรับส่งข้อมูลจะแสดงผ่านโมบายแอปพลิเคชันด้วยเส้นประ

2. Physical Plane

เป็นเลเยอร์ที่แสดงให้เห็นถึงภาพรวมของระบบในพื้นที่ที่เข้าถึงได้ยาก

3. Control Plane

เป็นเลเยอร์ที่เกี่ยวข้องกับการควบคุมการจัดเรียงตัวของโดรนโดยที่โดรนหัวหน้าจะทำการคำนวณตำแหน่งของโดรนทั้งหมดแล้วทำการส่งข้อมูลไปยังโดรนตัวอื่นๆรวมถึงโดรนตัวอื่นทำการตอบรับ

4.2 รายละเอียดของระบบที่พัฒนา

4.2.1 Input/Output Specification

- Input Specification
 - รับค่าตำแหน่งของสถานีต้นทางและปลายทางที่ต้องการให้รับส่งข้อมูลจากผู้ใช้
 - รับข้อมูลเงื่อนไขของการจัดเรียงตัวเพื่อการสร้างเครือข่าย
 - รับข้อมูลจากโดรนข้างเคียงเมื่อมีโดรนที่ใช้งานไม่ได้
 - รับการร้องขอเพื่อแสดงผลค่าสถานะของโดรนบนโมบายแอปพลิเคชัน
- Output Specification
 - โดรนบินไปยังตำแหน่งต่างๆเพื่อทำการรับส่งข้อมูลจากต้นทางและปลายทาง
 - ทำการจัดเรียงตัวเพื่อสร้างเครือข่ายตามเงื่อนไขที่ผู้ใช้กำหนด
 - ทำการรายงานกลับมาที่ผู้ใช้ และส่งโดรนตัวใหม่ออกไปกรณีที่ไม่สามารถส่งข้อมูลต่อได้
 - แสดงผลค่าสถานะต่างๆบนโมบายแอปพลิเคชันที่ใช้ระบบปฏิบัติการ iOS

4.2.2 Functional Specification

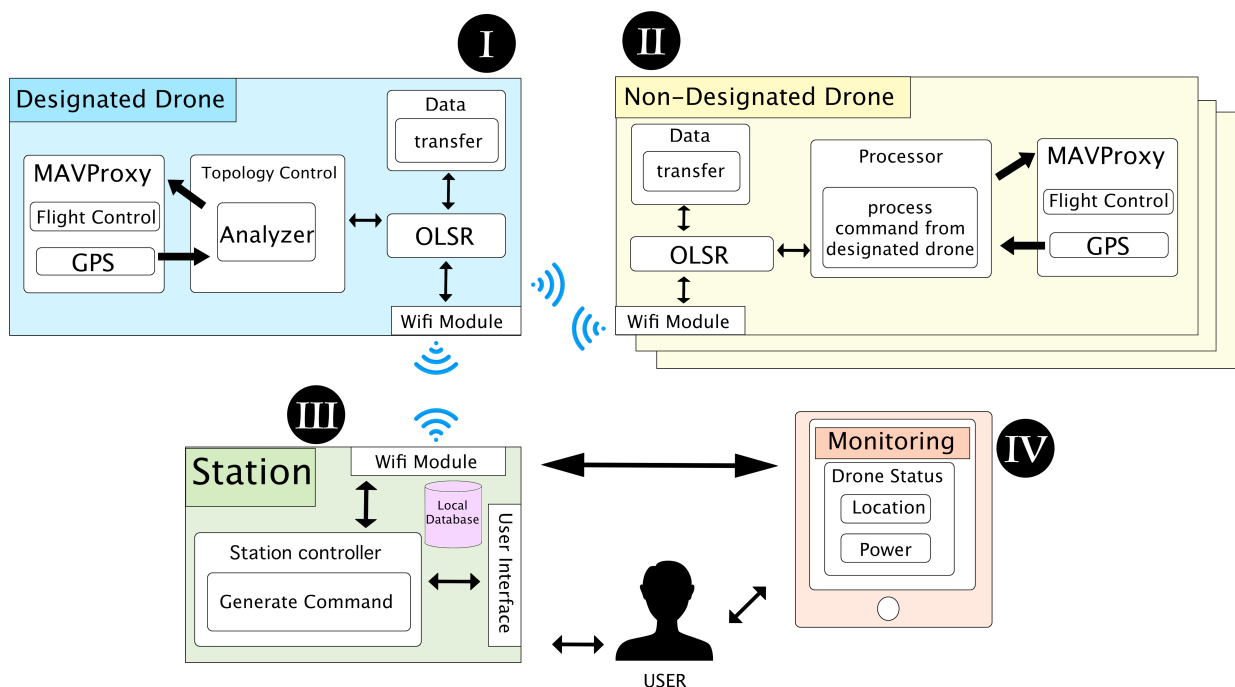
- ผู้ใช้สามารถที่จะสั่งโดรนด้วยคำสั่งพื้นฐาน
- ผู้ใช้สามารถสั่งงานให้โดรนทำการสร้างเครือข่ายตามเงื่อนไขของผู้ใช้
- สามารถตรวจสอบตำแหน่งของโดรนผ่านทางโมบายแอปพลิเคชัน
- สามารถตรวจสอบพลังงานคงเหลือของโดรนผ่านทางโมบายแอปพลิเคชัน
- สามารถตรวจสอบรูปแบบการสร้างเครือข่ายของโดรนผ่านทางโมบายแอปพลิเคชัน

4.2.3 กลุ่มผู้ใช้งาน

กลุ่มผู้ใช้งานเป็นบุคคลที่ต้องการจะรับส่งข้อมูลจากสถานีปลายทางที่เข้าถึงได้ยากเช่น ป่า เป็นต้น โดยผู้ใช้งานอาจเป็นเจ้าหน้าที่ของอุทยานที่มีความต้องการที่จะเก็บข้อมูลจากที่ต่างๆในความดูแลมาเพื่อตรวจสอบสิ่งผิดปกติที่เกิดขึ้น เป็นต้น

4.3 องค์ประกอบโดยรวมของระบบ

ในส่วนของโครงสร้างสามารถแบ่งออกเป็น 4 ส่วนหลักๆตามรูปที่ 4.3 ประกอบด้วย



รูปที่ 4.3 องค์ประกอบโดยรวมของระบบ

- Quadrotor drone

โดรนแบบ 4 ใบพัดที่ออกแบบและพัฒนาโดยภาควิชาวิศวกรรมการบินและอวกาศ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ได้ทำการติดตั้ง Banana Pi M1+ และมีตัวอ่านค่า GPS โดยสามารถขอได้จาก autopilot ดังแสดงในรูปที่ 4.4

- Designated Drone (โดรนหัวหน้า) (I)

ทำหน้าที่รับคำสั่งจากผู้ใช้งานเช่นตำแหน่งของสถานีต้นทางและปลายทางที่จะทำการเชื่อมต่อเพื่อรับส่งข้อมูลและทำการประมวลผลในการสร้างเครือข่ายระหว่างสถานีต้นทางและปลายทางให้แก่โดรนทั้งหมดโดยอัลกอริทึมที่สำคัญที่เอามาใช้ในการคำนึงถึงการจัดเรียงตัว

ของโดรนคือการประยุกต์ biconnected graph ซึ่งเป็น graph ที่หากมีการนำเส้นในกราฟออกไปจะยังคงสามารถไปถึงทุกโหนดได้ นำมาใช้ในการจัดเรียงตัวของโดรนเมื่อมีโดรน 3 ตัวจะให้โดรนฟอร์มตัวเป็นรูปสามเหลี่ยมเพื่อให้เกิดความ redundant ในกรณีที่มีโดรนที่ไม่สามารถรับส่งข้อมูลได้จะสามารถเลือกใช้อีกเส้นทางได้ทันทีตามโปรโตคอล OLSR

- **Non-designated Drone (II)**

ทำหน้าที่รับคำสั่งจากโดรนหัวหน้าและสื่อสารกับโดรนตัวอื่นๆเพื่อจัดรูปแบบของการสร้างเครือข่ายที่ใช้ในการรับส่งข้อมูล

- **Station Control (III)**

ในส่วนนี้ซึ่งเป็นส่วนติดต่อกับผู้ใช้ ประกอบไปด้วย 2 ส่วน คือ

1. Interface ที่ใช้ในการกำหนดตั้งค่าเริ่มต้นต่างๆให้กับโดรนเช่น ตำแหน่งของจุดเริ่มต้นกับปลายทางการจัดเรียงตัวตามเงื่อนไขของผู้ใช้
2. ส่วนรับส่งข้อมูลไปยังโดรนเพื่อส่งคำสั่งจากผู้ใช้งานไปยังโดรนรวมถึงรับข้อมูลสถานะต่างๆของโดรน

- **Monitoring via Mobile Application (IV)**

ทำหน้าที่ในการแสดงผลสถานะของโดรนได้แก่ ตำแหน่งเส้นทางที่ใช้ในการรับส่งข้อมูลเส้นทางในการเชื่อมต่อรวมไปถึงการจัดเรียงตัวของโดรนผ่านทางโปรแกรมประยุกต์บนสมาร์ทโฟน

โดยทุกองค์ประกอบทำหน้าที่เพื่อตอบสนองผู้ใช้งานโดยทำการรับส่งข้อมูลที่ส่งผ่านเครือข่ายของโดรนมาให้กับผู้ใช้โดยใช้ OLSR โปรโตคอลเพื่อหาเส้นทางในการส่งข้อมูล



รูปที่ 4.4 Quadrotor Drone

4.4 ขั้นตอนการพัฒนา

1. ศึกษาเทคโนโลยีที่เกี่ยวข้องในการพัฒนาระบบ

- ศึกษาหลักการทำงานและการควบคุมโดรนพื้นฐาน
- ศึกษาการทำงานของคอนโทรลเลอร์ที่ใช้กับโดรน
- ศึกษาการรับส่งข้อมูลแบบไร้สาย
- ศึกษาอัลกอริทึมต่างๆที่เกี่ยวข้อง
- ศึกษา tools ที่ใช้ในการทำโมบายแอปพลิเคชัน

2. ออกแบบระบบที่ใช้ในการพัฒนา

- ออกแบบอัลกอริทึมสำหรับการจัดเรียงตัวของโดรน
- ออกแบบระบบที่ใช้ในการติดต่อกับโดรนโดยรับเงื่อนไขของการสร้างเครือข่ายจากผู้ใช้
- ออกแบบอินเตอร์เฟซที่จะใช้เพื่อให้ผู้ใช้สามารถดูสถานะของโดรนผ่านโมบายแอปพลิเคชัน

3. พัฒนาระบบ

- ติดตั้งระบบปฏิบัติการและไลบรารีที่จำเป็นในการพัฒนา
- พัฒนา controller ที่ใช้ควบคุมโดรน
- พัฒนาอัลกอริทึมที่ใช้ในการสร้างเครือข่ายของโดรน
- พัฒนาโมบายแอปพลิเคชันที่ใช้แสดงสถานะต่างๆของโดรน

4. ทดสอบการใช้งาน

- ทดสอบการเปลี่ยนแปลงรูปแบบของโดรน
- ทดสอบการสร้างเครือข่ายของโดรนผ่านเงื่อนไขของผู้ใช้
- ทดสอบเวลาที่ใช้ในการค้นหาเส้นทางในการรับส่งข้อมูลใหม่เมื่อมีโดรนที่ใช้งานไม่ได้
- ทดสอบอัตราการรับส่งข้อมูลผ่านเครือข่ายที่สร้างจากโดรน
- ทดสอบการแสดงผลสถานะต่างๆของโดรนผ่านโมบายแอปพลิเคชัน

5. จัดทำเอกสารโครงการ

- จัดทำรูปเล่มรายงาน
- จัดทำเอกสารคู่มือการใช้งานระบบ

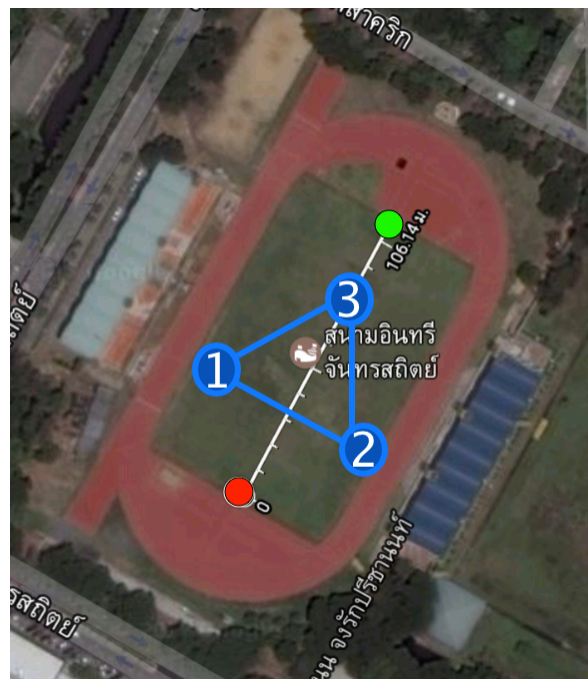
5 ผลการดำเนินโครงการและวิจารณ์

ใช้โดรนเพื่อสร้างเครือข่ายระหว่างต้นทางกับปลายทางโดยให้มีการจัดเรียงตัวตามเงื่อนไขของผู้ใช้งานคือต้องการให้ความต่อเนื่องในการส่งข้อมูลสูงที่สุด (redundant) ทดสอบเวลาที่ใช้ในสร้างเส้นทางที่ใช้ในการรับส่งข้อมูลใหม่เมื่อมีโดรนที่ไม่สามารถส่งข้อมูลได้รวมถึงทดสอบการแสดงผลสถานะของโดรนได้แก่ ตำแหน่ง เส้นทางที่ใช้รับส่งข้อมูล เส้นทางที่มีการเชื่อมต่อระหว่างโดรน สถานีต้นทางและสถานีปลายทาง

5. สภาพแวดล้อมในการทดสอบ

5.1.1 ทำการทดสอบเวลาที่ใช้ในการค้นหาเส้นทางในการรับส่งข้อมูลใหม่เมื่อมีโดรนที่ไม่สามารถรับส่งข้อมูลได้

ทำการทดสอบโดยใช้โดรนขนาด 4 ใบพัดทั้งหมด 3 ตัวและทำการทดสอบที่สนามอินทรีจันทร์สถิตย์ มหาวิทยาลัยเกษตรศาสตร์ซึ่งเหมาะสมกับการทดลองเนื่องจากเป็นพื้นที่โล่งระยะทางจากต้นทาง (จุดสีแดง) ไปถึงสถานีปลายทาง (จุดสีเขียว) คือ 106.14 เมตรโดยมีระยะห่างระหว่างโดรนแต่ละตัว 60 เมตรโดยใช้เทคโนโลยี IEEE802.11 โพรโทคอล OLSR และ ใช้ iperf3 [10] ในการทดสอบการส่งข้อมูลแบบ UDP ให้โดรนตัวที่ 1 ขาดจากการเชื่อมต่อแล้ววัดเวลาที่ใช้ในการค้นหาเส้นทางรับส่งข้อมูลใหม่รวมไปถึงอัตราการส่งข้อมูลในช่วงเวลาดังกล่าวโดยโดรนทั้ง 3 ตัวสร้างเครือข่ายดังแสดงรูปที่ 5.1



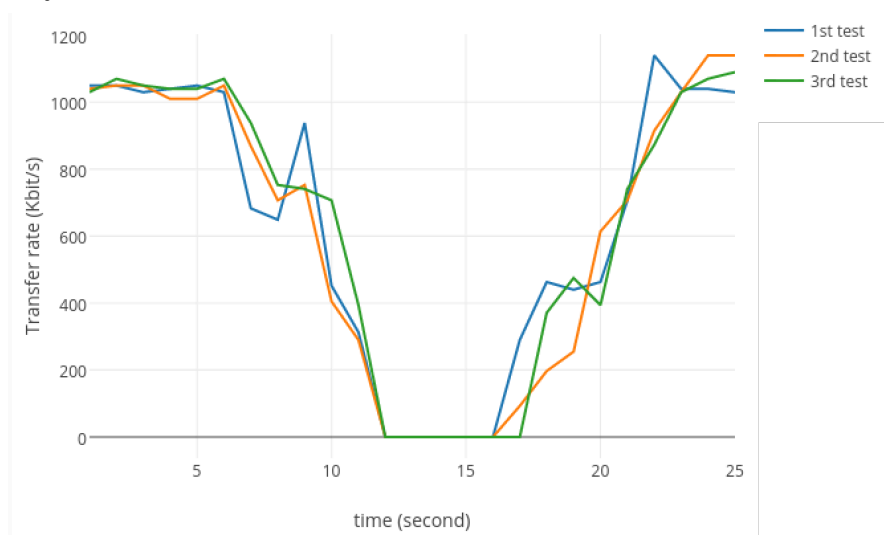
รูปที่ 5.1 การสร้างเครือข่ายของโดรน

5.1.2 การทดสอบการแสดงผลสถานะของโดรนบนโมบายแอปพลิเคชัน

ทำการทดสอบโดยใช้โดรนขนาด 4 ใบพัดทั้งหมด 3 ตัวและทำการทดสอบที่สนามอินทรีจันทร์สถิตย์ มหาวิทยาลัยเกษตรศาสตร์ซึ่งเหมาะกับการทดลองเนื่องจากเป็นพื้นที่โล่งระยะทางจากต้นทาง (จุดสีแดง) ไปถึงสถานีปลายทาง (จุดสีเขียว) คือ 106.14 เมตรโดยมีระยะห่างระหว่างโดรนแต่ละตัว 60 เมตรโดยใช้เทคโนโลยี IEEE802.11 โปรโตคอล OLSR ในการทดสอบการแสดงผลสถานะของโดรนผ่านโปรแกรมประยุกต์บนสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ iOS โดยให้จุดเริ่มต้นและปลายทางคือจุดเดียวกับที่ทดลองสร้างเครือข่ายดังรูปที่ 5.1 แล้วทดลองให้โดรนสร้างเครือข่ายโดยจัดเรียงตัวแบบเส้นตรงและแบบ redundant

5.2 ผลการทดสอบและการวิจารณ์ผล

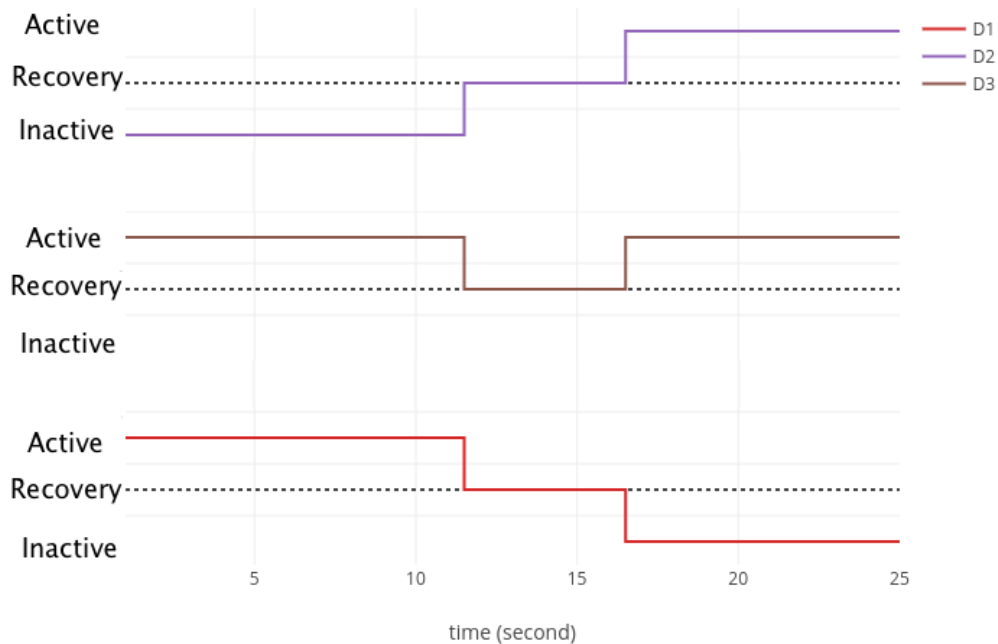
จากการทดสอบพบว่าเมื่อโดรนตัวที่ 1 ขาดจากการเชื่อมต่อ ระยะเวลาที่ใช้ในการค้นหาเส้นทางการรับส่งข้อมูล เฉลี่ยอยู่ที่ 5.33 วินาที อัตราการส่งข้อมูลเฉลี่ยอยู่ที่ 125.57-1086.67 Kbit/s ซึ่งถือว่าสามารถกู้คืนการรับส่งข้อมูลกลับมาในสภาวะปกติได้รวดเร็วเร็วกว่าการรอให้โดรนบินมาแทนตำแหน่งที่ไม่สามารถรับส่งข้อมูลได้ดังรูปที่ 5.2



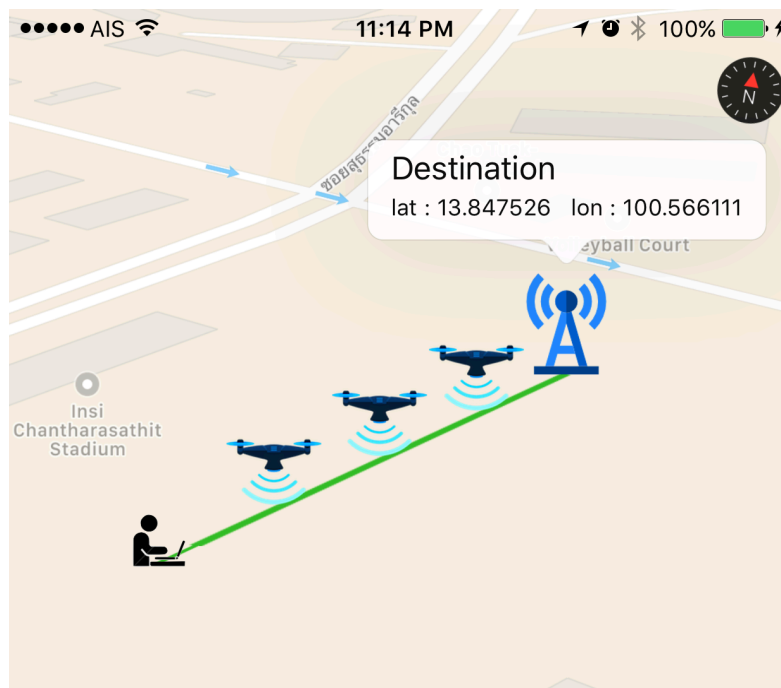
รูปที่ 5.2 ผลการทดสอบความสัมพันธ์ของอัตราการรับส่งข้อมูลและเวลา

และการเลือกเส้นทางที่ถูก active โดยดูจาก routing table ของโปรโตคอล OLSR เส้นทางแรกคือ source->1->3->destination และเส้นทางหลังจากโดรนตัวที่ 1 ขาดการเชื่อมต่อจากเครือข่ายคือ source->2->3->destination ซึ่งจากการทดลองพบว่าการหาเส้นทางใหม่หลังจากขาดการเชื่อมต่อทำได้รวดเร็วซึ่งหากโดรนมีการจัดเรียงตัวสร้างเครือข่ายที่ใหญ่ขึ้นโดยใช้หลักของ bi-connected graph จะสามารถทำได้รวดเร็วเช่นกันดังรูปที่ 5.3 D1, D2, D3 แทนโดรนตัวที่ 1 โดรนตัวที่ 2 และโดรนตัวที่ 3 ตามลำดับ เส้นประแสดงถึงสถานะที่โดรนกำลังคำนวณหาเส้นทางใหม่เหนือเส้นประหมายถึงในขณะนั้นมีการเชื่อมต่อผ่านโดรนนั้นในขณะที่ได้เส้นประหมายถึงในขณะนั้นไม่มีการเชื่อมต่อที่โดรนนั้นและการแสดงผลสถานะของโดรนได้แก่ตำแหน่งและเส้นทางที่ใช้ในเครือข่ายของโดรนได้แก่ เส้นทางที่ใช้ในการรับส่งข้อมูล

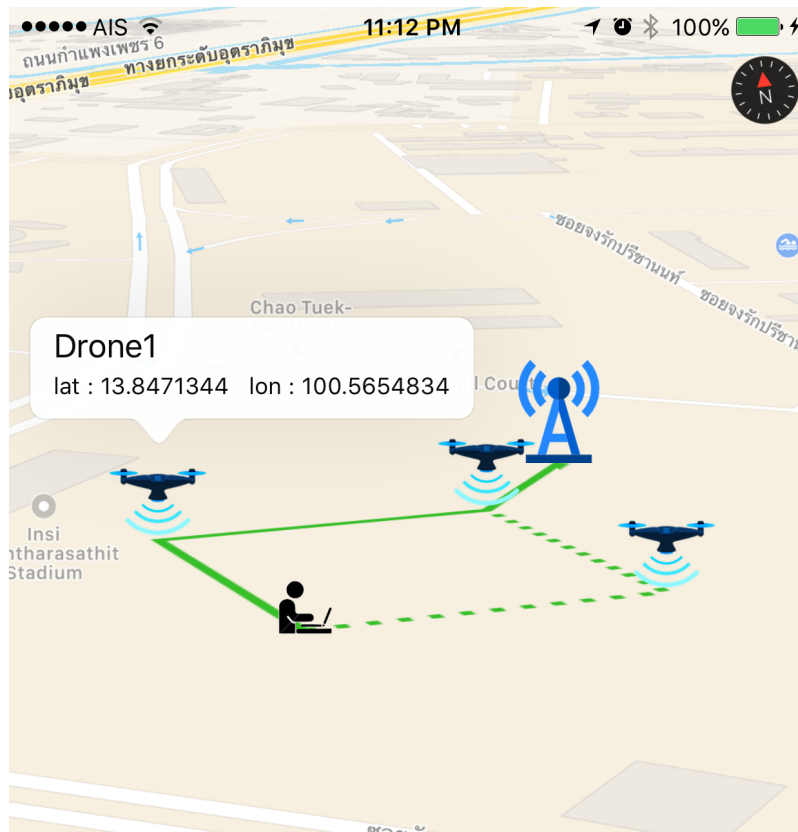
และเส้นทางที่มีการเชื่อมต่อระหว่างโดรน สถานีต้นทางและสถานีปลายทางแสดงด้วยเส้นทึบและเส้นประตามลำดับสามารถทำได้อย่างถูกต้องดังแสดงในรูปที่ 5.3 สำหรับการจัดเรียงตัวแบบเส้นตรง และรูปที่ 5.4 สำหรับการจัดเรียงตัวแบบ redundant ดังรูปที่ 5.5



รูปที่ 5.3 สถานะการเชื่อมต่อของบอร์ดเพื่อใช้เป็นเส้นทางและเวลา



รูปที่ 5.4 สถานะของโดรนแสดงผ่านโมบายแอปพลิเคชันโดยการจัดเรียงตัวแบบเส้นตรง



รูปที่ 5.5 สถานะของโดรนแสดงผ่านโมบายแอปพลิเคชันโดยการจัดเรียงตัวแบบ redundant

6 สรุปผลการดำเนินงานและข้อเสนอแนะ

6.1 ข้อสรุป

แพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนสามารถใช้งานได้ในพื้นที่ที่เข้าถึงได้ยากสามารถลดค่าใช้จ่ายจากการสร้างสถานีแบบตายตัวและสามารถเพิ่มความยืดหยุ่นให้กับการใช้งานโดยผู้ใช้สามารถกำหนดเงื่อนไขการสร้างโครงข่ายได้และนับว่าเป็นอีกหนึ่งทางเลือกที่ดีในการรับส่งข้อมูลในสถานที่ที่เข้าถึงได้ยาก

ผลจากการดำเนินงานพบว่าในกรณีที่ต้องการให้การรับส่งนั้นมีระยะเวลาในการกู้คืนการเชื่อมต่อกลับมาให้เร็วที่สุดสามารถทำได้ดีกว่าการนำโดรนไปแทนที่เมื่อมีโดรนที่ไม่สามารถรับส่งข้อมูลได้ทำให้การรับส่งเป็นไปอย่างต่อเนื่องมากขึ้นแต่จะต้องใช้จำนวนโดรนที่มากกว่าการสร้างเครือข่ายแบบเป็นเส้นตรงรวมไปถึงประสิทธิภาพในการรับส่งข้อมูลขึ้นกับสถานะของสภาพแวดล้อมในขณะนั้นเป็นอย่างยิ่งและการใช้งานของโดรนยังไม่สามารถใช้งานได้ยาวนานนัก

6.2 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคที่พบในการพัฒนาระบบแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรน

1. สภาพแวดล้อมมีผลต่อการใช้งานโดรนเป็นอย่างมากยกตัวอย่างเช่น สภาพอากาศที่ร้อนมากเกินไปส่งผลทำให้การประมวลผลมีประสิทธิภาพลดลง
2. วิธีการนำโดรนขึ้นบินต้องใช้หลายขั้นตอนรวมถึงต้องมีนักบินที่ควบคุมโดรนผ่านรีโมตเพื่อในกรณีฉุกเฉินจะสามารถเปลี่ยนการควบคุมไปใช้รีโมตเพื่อนำโดรนกลับมาอย่างปลอดภัย
3. ปัญหาของสัญญาณรบกวนส่งผลต่อประสิทธิภาพในการรับส่งข้อมูลผ่านเครือข่ายของโดรน
4. อุปกรณ์ฮาร์ดแวร์ที่ใช้ อาจทำให้มีการแสดงผลผิดพลาดในบางครั้ง
5. อุปกรณ์ที่ใช้กับโดรนมีราคาสูง

6.3 แนวทางการพัฒนาต่อ

แนวทางในการพัฒนาระบบแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนได้แก่

1. พัฒนาอัลกอริทึมในการจัดรูปแบบของโดรนเมื่อมีโดรนมากกว่า 3 ตัว
2. พัฒนาส่วนของ User Interface ที่ใช้ในการระบุค่าต่างส่งไปยังโดรนโดยอาจทำให้เป็น web application หรือ mobile application
3. เพิ่ม function ในส่วนการแสดงผลสถานะของโดรนผ่านทางโปรแกรมประยุกต์บนสมาร์ตโฟน โดยดูจากความต้องการของผู้ใช้งานและประโยชน์ที่ได้เพิ่มเติมจากการทำ function นั้นๆ

6.4 ข้อเสนอแนะ

1. ควรฝึกทักษะการโปรแกรมมิ่งภาษา Python และ JavaScript
2. ทำความเข้าใจระบบแพลตฟอร์มการจัดรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สาย โดยใช้โดรนเพื่อให้สามารถที่จะพัฒนาต่อยอดได้อย่างรวดเร็ว
3. ศึกษาหาข้อมูลเกี่ยวกับการสร้างเครือข่ายโดยใช้ UAV หรือที่เรียกว่า FANETs เพื่อกระตุ้นความคิดสร้างสรรค์หรือไอเดียในการพัฒนาต่อยอด
4. ศึกษาไลบรารีที่ใช้เพื่อที่จะทราบวิธีการเรียกค่าข้อมูลต่างๆ
5. ทดสอบโดรนในที่ที่มีสภาพแวดล้อมที่ดีไม่ร้อนเกินไป

7 บรรณานุกรม

- [1] ณัฐวุฒิ เอื้อศักดิ์สุภา, ชัยพร ใจแก้ว, อภิรักษ์ จันทร์สร้าง, อนันต์ ผลเพิ่ม*, ไชยวัฒน์ กล้าพล, “ระบบเครือข่ายไร้สายโดยใช้โดรน (Drone Wireless Mesh Network)”, In 7th ECTI Conference on Application Research and Development (ECTI-CARD 2015), Trang, Thailand, July 8-10, 2015, หน้า 59-62
- [2] ปริญญา เปียพนม, อภิรักษ์ จันทร์สร้าง, ชัยพร ใจแก้ว, อนันต์ ผลเพิ่ม*, ไชยวัฒน์ กล้าพล, “ระบบเครือข่ายไร้สายโดยใช้โดรน อัตโนมัติ (Automatic Drone Wireless Mesh Network)”, In 8th ECTI Conference on Application Research and Development (ECTI-CARD 2016), Hua Hin, Thailand July 27-29, 2016. หน้า 459-462
- [3] 802.11 IEEE wireless LAN standards. [ออนไลน์]. สืบค้นจาก: http://www.webopedia.com/TERM/8/802_11.html (วันที่สืบค้น 27 กันยายน 2559)
- [4] Ad Hoc NetWork : โครงข่ายไร้สายเฉพาะกิจ. [ออนไลน์]. สืบค้นจาก: <https://suntos.wordpress.com/2009/12/27/ad-hoc-network-โครงข่ายไร้สายเฉพาะกิจ> วันที่สืบค้น 27 กันยายน 2559
- [5] อากาศยานไร้คนขับ.[ออนไลน์].สืบค้นจาก:<http://th.wikipedia.org/wiki/อากาศยานไร้คนขับ> (วันที่สืบค้น 28 กันยายน 2559)
- [6] Optimized Link State Routing Protocol for Ad Hoc Networks. [ออนไลน์].สืบค้นจาก:<https://www.cs.jhu.edu/~dholmer/600.647/papers/OLSR.pdf> (วันที่สืบค้น 17 กันยายน 2559)
- [7] İlker Bekmezci, Ozgur Koray Sahingoz, Şamil Temel, Flying Ad-Hoc Networks (FANETs): A survey, Ad Hoc Networks, Volume 11, Issue 3, May 2013, Pages 1254-1270
- [8] ระบบกำหนดตำแหน่งบนโลก. [ออนไลน์]. สืบค้นจาก: <http://th.wikipedia.org/wiki/ระบบกำหนดตำแหน่งบนโลก> (วันที่สืบค้น 28 กันยายน 2559)
- [9] มาตรฐาน IEEE 802.11. [ออนไลน์].สืบค้นจาก: <http://wise.swu.ac.th/Default.aspx?tabid=3440> (วันที่สืบค้น 2 กุมภาพันธ์ 2560)
- [10] กพร.รุกใช้ UAV บินดูแลเหมืองแร่ นำร่องพื้นที่เหมืองหินปูน-แร่ยิปซัม. [ออนไลน์].สืบค้นจาก: <http://www.thairath.co.th/content/780891> (วันที่สืบค้น 5 กุมภาพันธ์ 2560)

8 ภาคผนวก

8.1 คู่มือการติดตั้ง

ระบบแพลตฟอร์มการจัดการรูปแบบการเชื่อมต่อแบบอัตโนมัติของเครือข่ายไร้สายโดยใช้โดรนสามารถแบ่งออกเป็น 3 ส่วนหลักได้แก่ ส่วนสถานีควบคุม ส่วนระบบฝังตัวบนโดรนและส่วนแสดงผลสถานะของโดรนผ่านโมบายแอปพลิเคชัน โดยคู่มือนี้ใช้กับคอมพิวเตอร์ระบบปฏิบัติการ Mac OSX

- ส่วนสถานีควบคุม (Station Control)

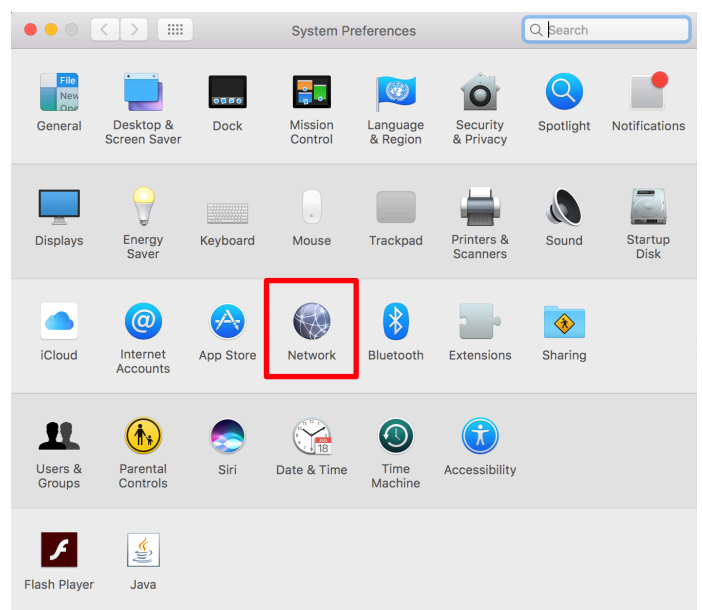
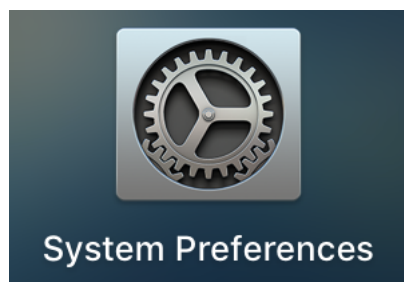
- ทำการติดตั้ง station.py และโปรแกรมที่อยู่ใน Folder ชื่อ Drone_visualization
- ทำการติดตั้ง brew โดยใช้คำสั่ง

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- ติดตั้งโปรโตคอลค้นหาเส้นทาง OLSR โดยใช้คำสั่ง
- ทำการเพิ่ม network เพื่อที่จะใช้เชื่อมต่อกับโดรนแบบ Adhoc

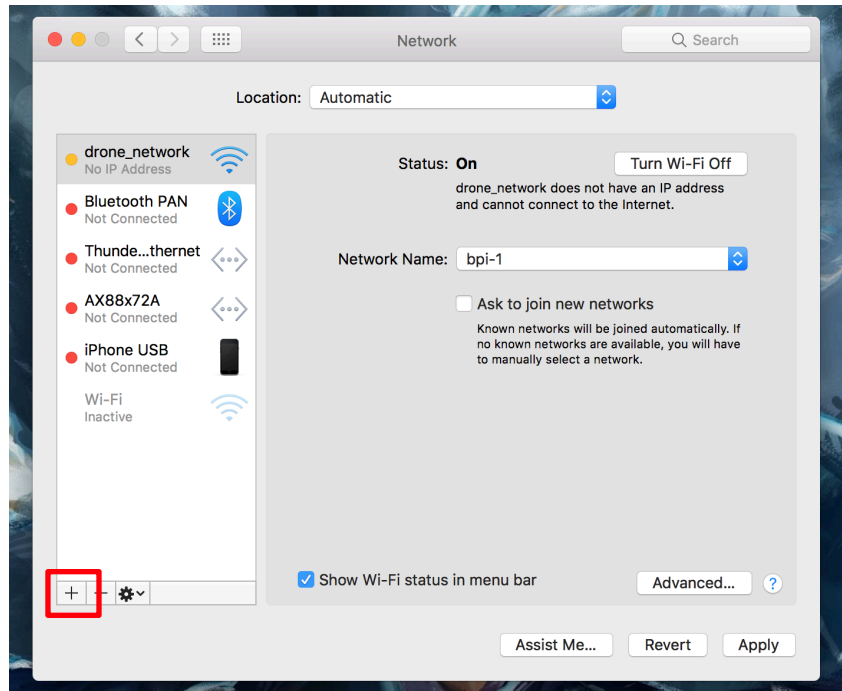
```
brew install olsrd
```

1. เริ่มจากการเข้าไปที่ System Preferences->Network ดังรูปที่ 8.1

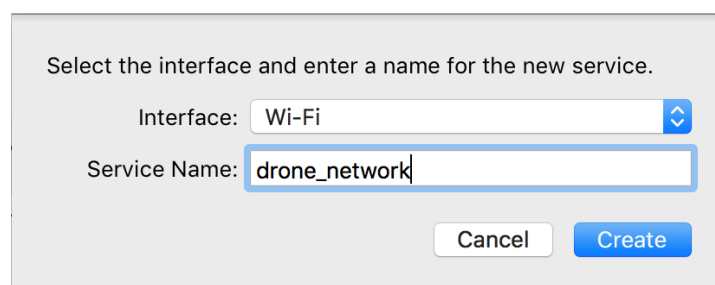


รูปที่ 8.1 Icon ของ System Preferences และ Network

2. สร้าง network ชื่อ drone_network โดยเริ่มจากกดปุ่ม + แล้วตั้งชื่อดังรูป ที่ 8.2 และ 8.3 ตามลำดับ

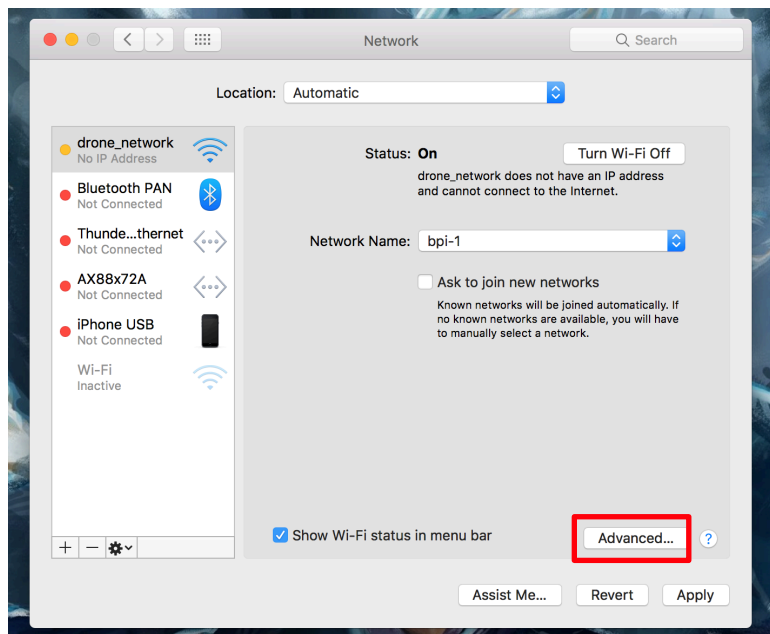


รูปที่ 8.2 การเพิ่ม network service



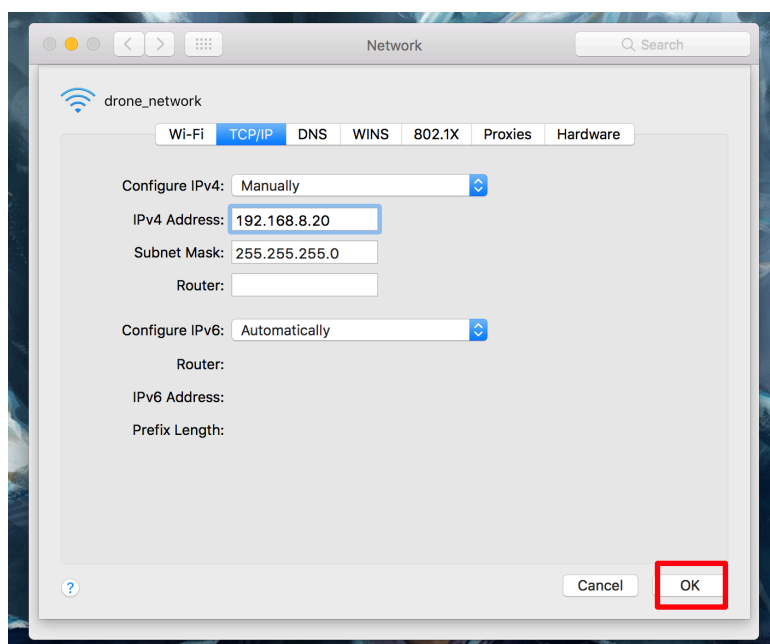
รูปที่ 8.3 การตั้งชื่อ network

3. ทำการ set ip แบบ manual โดยกดที่ปุ่ม Advanced... ดังรูปที่ 8.4



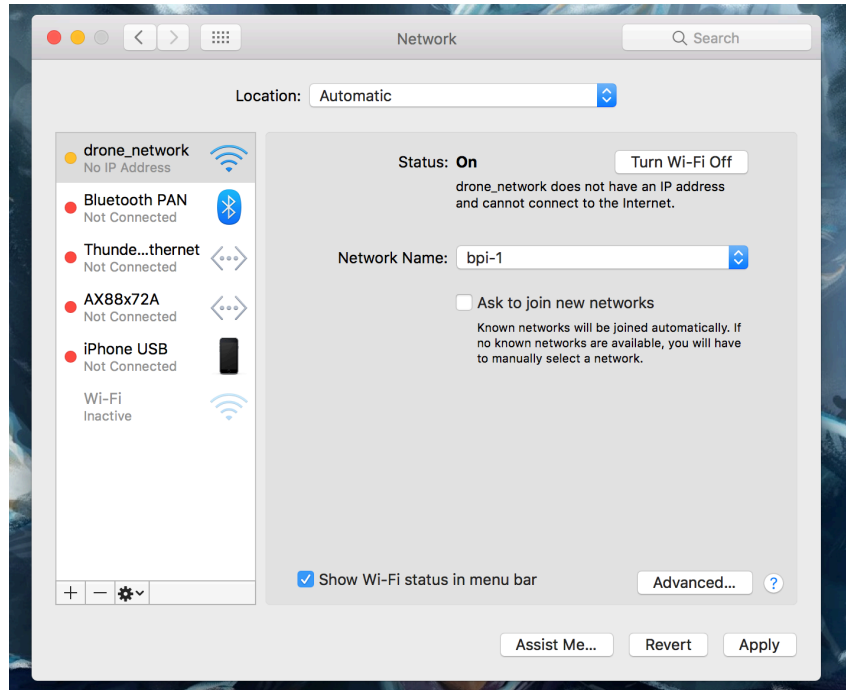
รูปที่ 8.4 ปุ่มตั้งค่า network

4. ทำการตั้งค่าเลข IP และ Subnet Mask โดย x สามารถแทนด้วยเลขระหว่าง 4-254 แล้วกด ok ดังรูปที่ 8.5



รูปที่ 8.5 การตั้งค่า network IP และ Subnet Mask

5. หากทำได้อย่างถูกต้องจะปรากฏ network ชื่อ drone_network ดังรูปที่ 8.6



รูปที่ 8.6 drone_network created

- ติดตั้ง React Native โดยใช้คำสั่งต่อไปนี้

```
brew install node
brew install watch
npm install -g react-native-cli
```

- ลง package ที่จำเป็นโดย cd ไปในโฟลเดอร์ Drone_visualization แล้วสั่ง npm install
- ส่วนระบบฝังตัวบนโดรน
 - ทำการติดตั้ง Bananian Linux
 1. Download จาก <https://www.bananian.org/download> และทำการแตกไฟล์
 2. เสียบ SD card เข้ากับคอมพิวเตอร์โดยต้องมีขนาดความจุไม่ต่ำกว่า 4 GB
 3. ทำการ identify SD card และ unmountdisk โดยจากคำสั่งด้านล่างสมมติว่าได้ output จากการ identify SD card เป็น “disk9”

```
diskutil list
diskutil unmountDisk disk9
```

4. ทำการ write image ลงบน SD card

```
sudo dd if=bananian-1604.img of=/dev/rdisk9 bs=1m && diskutil
```

5. หลังจากเสร็จ สามารถนำ SD card ไปใช้กับ Banana Pi M1+ ได้เลย

- ทำการติดตั้ง software และ package ที่ต้องใช้บน Banana Pi M1+

ติดตั้ง MAVProxy

```
sudo apt-get install python-dev python-opencv python-wxgtk3.0  
python-pip python-matplotlib python-pygame python-lxml  
  
sudo pip install MAVProxy
```

ติดตั้ง OLSR

```
sudo apt-get install olsrd
```

ติดตั้ง SSH-Server

```
sudo apt-get install openssh-server
```

ติดตั้งโปรแกรมที่ใช้ในการติดต่อกันระหว่างโดรนและสถานี ได้แก่ drone_server.py drone_client.py mavproxy-readfile.py และโปรแกรมที่ใช้ในการส่งข้อมูลสถานะของโดรนซึ่งอยู่ใน folder ชื่อ express-mysql-boilerplate-master นำมาไว้ใน Banana Pi M1+

ติดตั้ง Node.js และ npm

```
sudo apt-get update  
sudo apt-get install nodejs  
sudo apt-get install npm
```

ติดตั้ง package ที่ต้องใช้

ให้ cd เข้าไปที่ express-mysql-boilerplate-master แล้วใช้ คำสั่ง

```
npm install
```

- ทำการ set ค่า network interfaces

1. เปิดไฟล์โดยใช้คำสั่ง

```
nano /etc/network/interfaces
```

2. ตั้งค่าให้เป็นเหมือนด้านล่าง

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.8.x
    network 255.255.255.0
    wireless-channel 1
    wireless-essid bpi-1
    wireless-mode ad-hoc
```

โดยที่ x ให้แทนค่าด้วยเลข 11-13 (แต่ละ Banana Pi M1+ ต้องไม่ซ้ำกัน)

- ทำการตั้งค่าให้เริ่มต้นรันโปรแกรมต่างๆ โดยเข้าไปแก้ไขที่ไฟล์ใน /etc/rc.local ให้เพิ่มคำสั่งดังนี้

```
screen python mavproxy-readfile.py --master=/dev/ttyACM0
--baudrate=115200 --aircraft=MyCopter

olsrd -i wlan0

screen python drone_client.py
screen python drone_server.py

screen npm run prod
```

- ส่วนแสดงผลสถานะของโดรนผ่านโมบายแอปพลิเคชัน

ที่คอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Mac OSX

- ติดตั้ง Xcode ผ่านทาง AppStore
- เสียบสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ iOS แล้วเข้าไปที่ /Drone_visualization/ios/ แล้วเปิด Drone_visualization.xcodeproj
- ทำการ build application ผ่าน Xcode ลงบนสมาร์ตโฟน

8.2 คู่มือการใช้งาน

- ที่ station

- ใช้คำสั่งเพื่อเปิดหน้าต่าง command line ที่ใช้เพื่อกำหนดค่าต่างๆให้กับโดรนดังนี้

```
python station.py
```

โดยเมื่อรันแล้วสามารถที่จะทราบ option ต่างๆได้โดยใช้คำสั่ง “--h”
จะได้ manual ในการใช้คำสั่งต่างๆดังนี้

```
----- Manual -----

set d [ip]
    -set initial designated drone

set home [ip] [latitude] [longitude]
    -set home location of drone

set station [latitude] [longitude]
    -set station location

set destination [latitude] [longitude]
    -set destination location

set mode [number]
    -choose formulation mode
        - 1 : straight line mode
        - 2 : redundant mode

formulate init
    -start formulation process

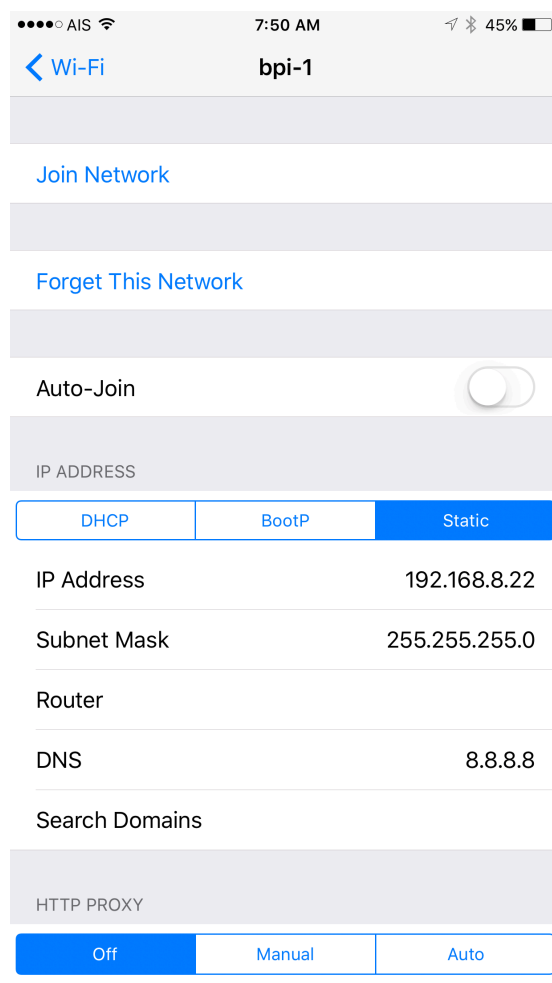
--h
    -show command option
```

ทำการ run โปรแกรม call_trace.py โดยใช้คำสั่ง

```
python call_trace.py
```

- ส่วน Mobile Application

- เชื่อมต่อกับ wifi ชื่อ bpi-1 โดยตั้งค่าให้ IP เป็นแบบ static โดยมี IP address เป็น 192.168.8.x โดยที่ x เป็นได้ตั้งแต่ 21-254 Subnet Mask คือ 255.255.255.0 ดังรูปที่ 8.6



รูปที่ 8.7 Mobile Phone static IP setting

- หลังจาก build Application จาก Xcode แล้ว ให้เชื่อมต่อกับอินเทอร์เน็ตเพื่อ verify application
- กลับมาเชื่อมต่อกับ bpi-1 อีกครั้ง แล้วทำการเปิด Application

9 ประวัตินิสิต

ชื่อ นายเพชร พึ่งทองหล่อ

ระดับการศึกษา นิสิตชั้นปีที่ 4 ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันการศึกษา มหาวิทยาลัยเกษตรศาสตร์

ที่อยู่ปัจจุบัน 246 หมู่ 4 หมู่บ้านรือคการ์เด็นเพลส ถ.รามอินทรา ซอยวัชรพล แขวงท่าแร้ง เขต

บางเขน กทม. 10220

โทรศัพท์ 080-622-0411

Email potcharajoe@gmail.com

ระดับการศึกษา ปริญญาตรี

คุณวุฒิการศึกษา	สถาบัน	ปีการศึกษาที่จบ
มัธยมศึกษาตอนปลาย	โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ศูนย์วิจัยและพัฒนาการศึกษา	2555
มัธยมศึกษาตอนต้น	โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ศูนย์วิจัยและพัฒนาการศึกษา	2552