



ระบบช่วยลงจอดมัลติคอปเตอร์โดยใช้กล้องตรวจจับตำแหน่ง

ทศพล เกตุพันธุ์

ภูวดินทร์ พงษ์สิงห์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ และ เทคโนโลยีมัลติมีเดีย

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

ปีการศึกษา 2557



MULTICOPTER LANDING SYSTEM USING CAMERA POSITION DETECTION

Tossapol Gettupun

Pubadin Pongsing

TNI

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science

Department of Information Technology and Technology Multimedia

Faculty of Information Technology

Thai - Nichi Institute of Technology

Academic Year 2014

หัวข้อปริญญานิพนธ์

ระบบช่วยลงจอดมัลติคอปเตอร์โดยใช้กล้องตรวจจับตำแหน่ง
Multicopter Landing System Using Camera Position Detection

โดย

นาย ทศพล เกตุพันธุ์
นาย ภูผินทร์ พงษ์สิงห์

สาขาวิชา

เทคโนโลยีสารสนเทศ และ เทคโนโลยีมัลติมีเดีย

อาจารย์ที่ปรึกษาปริญญานิพนธ์

ผู้ช่วยศาสตราจารย์ ร้อยตรี ตรีรัตน์ เมตต์การุณจิต

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย – ญี่ปุ่น อนุมัติให้แนบปริญญานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตร์บัณฑิต

..... คณบดีคณะเทคโนโลยีสารสนเทศ
(รศ. ดร. รัตติกร วรากุลศิริพันธุ์)
วันที่ เดือน พ.ศ.

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ
(ดร. จูติพร เลิศรัตน์เดชากุล)

..... กรรมการ
(อาจารย์ณปภัช วิชัยดิษฐ์)

..... กรรมการ และ อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ร้อยตรี ตรีรัตน์ เมตต์การุณจิต)

ชื่อผู้จัดทำโครงการ : นาย ทศพล เกตุพันธุ์
: นาย ภูบดินทร์ พงษ์สิงห์
ชื่อโครงการ : ระบบช่วยลงจอดมัลติคอปเตอร์โดยใช้กล้องตรวจจับตำแหน่ง
อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ร้อยตรี ตรีรัตน์ เมตต์การุณจิต

บทคัดย่อ

โครงการนี้พัฒนาระบบช่วยลงจอดมัลติคอปเตอร์ โดยใช้กล้องเว็บแคมที่ติดตั้งอยู่บนมัลติคอปเตอร์ตรวจจับสัญญาณ ที่จุดลงจอดโดยพิจารณาจากค่าสีพื้น ที่ และระยะที่น้อยกว่าค่าที่กำหนด จะส่งคำสั่ง Landing ให้ลงจอดได้มายัง สถานีควบคุมภาคพื้นดิน (Ground Control Station) เพื่อให้นักบินสามารถมั่นใจได้ว่า จะให้มัลติคอปเตอร์ลงจอดได้อย่างปลอดภัย อันเนื่องมาจากค่าตำแหน่ง จีพีเอส ที่ผิดพลาด หรือระบบส่งภาพที่ขัดข้อง โดยมัลติคอปเตอร์ใช้บอร์ด ควบคุมเอพีเอ็ม (APM) เชื่อมต่อการสื่อสาร มายังโปรแกรมมิชชันแพลนเนอร์ (Mission Planner) และใช้บอร์ด ราสเบอร์รี่พาย (Raspberry Pi) ประมวลผลภาพของโปรแกรมโอเพ่นซีวี (OpenCV) เพื่อตรวจจับจุดลงจอดและส่งข้อมูลไร้สายผ่านสายพาวเวอร์มายังคอมพิวเตอร์ จากผลการทดสอบในการบังคับมัลติคอปเตอร์ ใช้สายตามอง (Line of sight : LOS) เพื่อลงจอดมัลติคอปเตอร์ มีค่าความคลาดเคลื่อนจากจุดลงจอดมัลติคอปเตอร์ เฉลี่ย 76 เซนติเมตร เนื่องจากแรงลมที่พัดขณะที่มัลติคอปเตอร์ลงจอด ทำให้ตำแหน่งเปลี่ยนไป เปรียบเทียบกับการลงจอด แบบใช้จีพีเอส มีค่าความคลาดเคลื่อนมัลติคอปเตอร์ เฉลี่ย 346 เซนติเมตร โดยระบบการลงจอดมัลติคอปเตอร์ที่พัฒนา ช่วยให้การลงจอดมัลติคอปเตอร์มีความผิดพลาดน้อยกว่าการใช้จีพีเอส

(ผู้ช่วยศาสตราจารย์ ร้อยตรี ตรีรัตน์ เมตต์การุณจิต)

ที่ปรึกษาโครงการ

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง ระบบช่วยลงจอดมัลติคอปเตอร์ โดยใช้กล้องตรวจจับตำแหน่ง เล่มนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือจากอาจารย์เป็นอย่างดี ผู้ช่วยศาสตราจารย์ ร้อยตรี ตรีรัตน์ เมตต์การุณจิต อาจารย์ที่ปรึกษาโครงการ ที่ได้ให้คำแนะนำต่างๆ จนกระทั่งส่งผลให้ปริญญานิพนธ์ สำเร็จลุล่วงด้วยดี

ขอขอบคุณ คณาจารย์ตลอดจนผู้เกี่ยวข้องทุกท่าน ที่ได้ให้คำปรึกษาและให้ความช่วยเหลือเป็นอย่างดี จนปริญญานิพนธ์เล่มนี้สำเร็จลุล่วง ผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

นาย ทศพล เกตุพันธุ์

นาย ภูบินทร์ พงษ์สิงห์

TNI

สารบัญ

หน้า

บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูปภาพ	จ
สารบัญตาราง	ช
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล	1
1.2 ปัญหาที่พบ	3
1.3 วัตถุประสงค์ของการทำโครงการ	3
1.4 ขอบข่ายโครงการ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	3
1.6 ระยะเวลาการดำเนินงาน	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 มัลติคอปเตอร์	5
2.2 ราบเบอร์รี่พาย	7
2.3 บอร์ด เอพีเอ็ม	14
2.4 มิชชั่นแพลนเนอร์	16
2.5 ไอเฟ่น ซีวี	17
2.6 ไพทอน	18

สารบัญ (ต่อ)

หน้า

บทที่ 3 การพัฒนาระบบช่วยจดมัลติคอปเตอร์

3.1 มัลติคอปเตอร์และอุปกรณ์ที่ใช้ 21

3.2 ระบบตรวจจับจุดลงจอด 26

บทที่ 4 ผลการทดสอบ

4.1 การทดลองแบบการควบคุมโดยใช้สายตา LOS 32

4.2 การทดลองแบบการควบคุมโดยใช้ระบบจีพีเอส RTL 33

4.3 สรุปผลการทดสอบ 34

บทที่ 5 บทสรุปและข้อเสนอแนะ

5.1 สรุป 36

5.2 ปัญหาในการทำงาน 37

5.3 ข้อเสนอแนะ 37

บรรณานุกรม

ภาคผนวก

TNI

TAHT - NICHI INSTITUTE OF TECHNOLOGY

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 ตัวอย่างมัลติคอปเตอร์	2
รูปที่ 1.2 มัลติคอปเตอร์ใช้สแกนภาพและแปลงเป็น GIS	2
รูปที่ 1.3 ตัวอย่างบอร์ด ราสเบอร์รี่พาย	2
รูปที่ 1.4 ตัวอย่างบอร์ด เอฟเอ็ม	3
รูปที่ 2.1 ตัวอย่างมัลติคอปเตอร์	6
รูปที่ 2.2 ตัวอย่างรีโมท	7
รูปที่ 2.3 ตัวอย่างบอร์ด ราสเบอร์รี่พาย	7
รูปที่ 2.4 ตัวอย่างโครงสร้างบอร์ด ราสเบอร์รี่พาย	10
รูปที่ 2.5 ตัวอย่างบอร์ด ราสเบอร์รี่พาย	10
รูปที่ 2.6 ตัวอย่าง ฟิน บอร์ดราสเบอร์รี่พาย 1	11
รูปที่ 2.7 ตัวอย่าง ฟิน บอร์ดราสเบอร์รี่พาย 2	11
รูปที่ 2.8 ตัวอย่างสาย อาซีเอ	11
รูปที่ 2.9 ตัวอย่าง แอลอีดี สถานะบอร์ดราสเบอร์รี่พาย	12
รูปที่ 2.10 ตัวอย่าง กล่องโมดูลราสเบอร์รี่พาย	12
รูปที่ 2.11 ตัวอย่าง สาย HDMI	12
รูปที่ 2.12 ตัวอย่างตัวแปลง HDMI เป็น VGA	12
รูปที่ 2.13 หน้าจอ ราสเบอร์รี่พาย	13
รูปที่ 2.14 บอร์ด APM 2.6	14
รูปที่ 2.15 รายละเอียด APM 2.6	14
รูปที่ 2.16 หลักการเชื่อมต่อ APM 2.6	15

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.17 อุปกรณ์เชื่อมต่อ APM 2.6	15
รูปที่ 2.18 ตัวอย่าง Mission Planner	15
รูปที่ 2.19 ตัวอย่างการกำหนด Waypoint	16
รูปที่ 2.20 เครื่องหมายโอเพ่นซีวี	17
รูปที่ 2.21 ตัวอย่างโค้ดไพทอน	20
รูปที่ 3.1 แผนผังระบบช่วยลงจอด	21
รูปที่ 3.2 ตัวอย่างมัลติคอปเตอร์	22
รูปที่ 3.3 ตัวรับดาวเทียม	23
รูปที่ 3.4 บอร์ดราสเบอร์รี่พาย 2	23
รูปที่ 3.5 ตัวรับส่ง Wi-Fi	24
รูปที่ 3.6 กล้องเว็บแคม	24
รูปที่ 3.7 รีโมทควบคุม	25
รูปที่ 3.8 คอมพิวเตอร์	25
รูปที่ 3.9 แผนผังระบบตรวจจับจุดลงจอด	26
รูปที่ 3.12 ไดอะแกรมระบบค้นหาสัญลักษณ์	27
รูปที่ 3.13 ขณะลงจอดตามสัญลักษณ์	28
รูปที่ 3.14 ราสเบอร์รี่พายติดตั้งบนมัลติคอปเตอร์	29
รูปที่ 3.15 โปรแกรมมิชชันแพลนเนอร์	30
รูปที่ 3.16 หน้าจอร์ับคำสั่ง Landing	31
รูปที่ 4.1 การทดลองระบบตรวจหาจุดลงจอด	32

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.2 กราฟค่า Distant แบบ LOS	33
รูปที่ 4.3 กราฟค่า Distant แบบ RTL	34
รูปที่ 4.4 กราฟค่า Distant แบบ LOS และ RTL	34



สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน	4
ตารางที่ 2.1 คุณสมบัติราคาเบอร์รี่พาย	8
ตารางที่ 4.1 ตารางการทดลอง LOS	32
ตารางที่ 4.2 ตารางการทดลอง RTL	33

TNI

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

มัลติคอปเตอร์ คือ เครื่องบิน ขานพาหนะ ที่สามารถบินได้โดยอาศัยใบพัดเป็นหลัก ที่สามารถควบคุมจากระยะไกลมีทั้งที่บินด้วยตัวเอง และการบังคับด้วยคลื่นวิทยุ ดังรูปที่ 1.1 เราสามารถใช้ประโยชน์จากมัลติคอปเตอร์ ได้อย่างหลากหลาย ด้วยความสามารถเฉพาะตัวและขนาดที่กระทัดรัด มันจึงถูกนำมาใช้ได้แม้ในสถานที่ที่มีผู้คนพลุกพล่าน หรือมีสิ่งกีดขวางมากมาย เช่น การถ่ายภาพมุมสูง ถ่ายวิดีโอมุมสูง รวมทั้งใช้ในการถ่ายทำภาพยนตร์ และการเก็บภาพลักษณะภูมิประเทศด้วยซึ่งในอนาคต มัลติคอปเตอร์ อาจถูกนำมาใช้งานเพิ่มมากขึ้น เนื่องจากหุ่นยนต์เหล่านี้สามารถบินได้ทั้งในแนวเส้นตรงและแนวโค้ง หากมีการพัฒนาด้านกำลังและประสิทธิภาพหุ่นยนต์เหล่านี้ ยังสามารถนำมาใช้ในงานด้านต่าง ๆ เช่น การขนส่ง และการก่อสร้าง โดยเฉพาะอย่างยิ่งด้านการเคลื่อนย้ายหรือขนส่งวัสดุอุปกรณ์ต่างๆ และยังสามารถสแกนพื้นที่ให้เป็นระบบสารสนเทศภูมิศาสตร์ (Geographic Information System : GIS) ดังรูปที่ 1.2 คือการทำให้มองเห็นในรูปร่างของข้อมูล อาจจะเป็นแผนที่หรือแสดงผลเป็นสามมิติก็ตาม โดยเราจะใช้มัลติคอปเตอร์ ที่ติดกล้องทำการสแกนภาพ และแปลงเป็น GIS เพื่อนำข้อมูลมาใช้ประโยชน์ เช่น บอกพิกัดทางภูมิศาสตร์ เป็นต้น ดังรูปภาพ 1.2 ในส่วนของระบบควบคุมการลงจอด ของมัลติคอปเตอร์ เนื่องในปัจจุบันการลงจอดของ มัลติคอปเตอร์ มีอยู่หลายรูปแบบด้วยกัน ไม่ว่าจะเป็นการบังคับให้ลงจอดแบบแมนนวล หรือลงจอดตามพิกัดดาวเทียม (Global Positioning System : GPS) ก็ตาม แต่ในปัจจุบันมัลติคอปเตอร์ สามารถลอยตัวอยู่บนอากาศได้ไม่เกิน 10 – 15 นาที เนื่องจากมีปัญหาเกี่ยวกับพลังงานหรือแบตเตอรี่ เมื่อแบตเตอรี่หมดมัลติคอปเตอร์ จะค่อย ๆ ลงจอดอย่างช้า ๆ เราจึงต้องมีการคาดการณ์หรือกำหนดจุดลงจอดไว้ล่วงหน้า เพื่อให้มัลติคอปเตอร์ สามารถลงจอดได้ตามที่กำหนดไว้แบบอัตโนมัติใน โครงงานนี้จะเขียนโปรแกรมให้ มัลติคอปเตอร์ลงจอดตามจุดที่กำหนดได้อย่างถูกต้อง โดยจะใช้ บอร์ดราสเบอร์รี่พาย ดังรูป 1.3 เพื่อใช้ในการ สั่งงานบอร์ดควบคุมมัลติคอปเตอร์ (APM) เพื่อให้ บอร์ด APM สั่งงานไปที่ มัลติคอปเตอร์ เพื่อให้มัลติคอปเตอร์สามารถลงจอดตามจุดที่กำหนดไว้ โดยจะมีการติดกล้องไว้ที่ มัลติคอปเตอร์ เพื่อให้บอร์ดราสเบอร์รี่พาย สามารถมองเห็นจุดที่กำหนดและลงจอดได้ตามที่กำหนดไว้



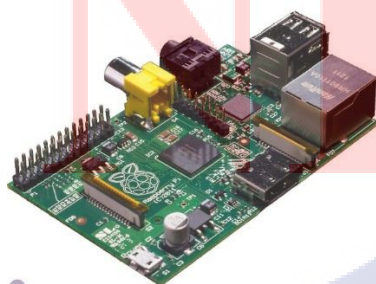
รูปที่ 1.1 ตัวอย่างมัลติคอปเตอร์

มัลติคอปเตอร์ แบบหกใบพัดที่ใช้ในการกิจ ต่าง ๆ เช่นการถ่ายภาพทางอากาศ การขนส่ง รวมไปถึงการใช้ในการสอดแนม



รูปที่ 1.2 มัลติคอปเตอร์ใช้ในสแกนภาพและแปลงเป็น GIS

รูปภาพการสแกนภาพ ทางอากาศและแปลงภาพเป็นระบบ GIS เพื่อใช้ในการทหาร หรือ บอกลักษณะภูมิประเทศ



รูปที่ 1.3 ตัวอย่างบอร์ด Raspberry Model B

บอร์ดราสเบอร์รี่พาย หรือ คอมพิวเตอร์ขนาดเล็ก ที่ใช้ระบบปฏิบัติการ ลินุกซ์ในการประมวลผล

1.2 ปัญหาที่พบ

- 1) มัลติคอปเตอร์สามารถลอยตัวบนอากาศได้ไม่นานจึงต้องกำหนดจุดลงจอด
- 2) การกำหนดจุดลงจอดอาจผิดพลาดได้เมื่อมีวัตถุใกล้เคียงกับสิ่งที่กำหนด
- 3) อาจมีปัญหาเกี่ยวกับลม ทำให้มีการคลาดเคลื่อนในการลงจอด
- 4) อาจมีปัญหาเกี่ยวกับการสั่งงาน หรือมีสัญญาณรบกวน
- 5) อาจมีปัญหาเกี่ยวกับระบบการทำงานของ จีพีเอส

1.3 วัตถุประสงค์ของการทำโครงการ

- 1) เพื่อศึกษาการประมวลผลภาพผ่าน บอร์ด ราสเบอร์รี่พาย
- 2) เพื่อศึกษาการควบคุม มัลติคอปเตอร์ผ่านบอร์ด เอฟเอ็ม
- 3) เพื่อสร้างระบบช่วยค้นหาจุดลงจอดจากเครื่องหมายที่กำหนดโดยส่งค่ามายัง

คอมพิวเตอร์

1.4 ขอบข่ายโครงการ

สามารถใช้ บอร์ด ราสเบอร์รี่พาย ทำการค้นหาจุดลงจอดของ มัลติคอปเตอร์และส่งข้อมูลมาที่คอมพิวเตอร์ และสั่งงานผ่านบอร์ด เอฟเอ็ม ให้มัลติคอปเตอร์ลงตามจุดที่กำหนด

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

- 1) ได้รับความรู้ในการเขียน โปรแกรมประมวลผลภาพ รวมไปถึงการตรวจจับ ภาพที่กำหนด และสั่งการมัลติคอปเตอร์ให้ลงจอดตามจุดที่กำหนด
- 2) ได้เรียนรู้ การใช้งาน ราสเบอร์รี่พาย และ บอร์ด เอฟเอ็ม มาใช้สั่งการให้ มัลติคอปเตอร์สามารถลงจอดได้ตามจุดที่กำหนด
- 3) สามารถนำ ระบบค้นหาจุดลงจอด มาประยุกต์ใช้ในการลงจอดของมัลติคอปเตอร์

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 มัลติคอปเตอร์

มัลติคอปเตอร์ คือ เครื่องบิน ขานพาหนะ ที่สามารถบินได้โดยอาศัยใบพัดเป็นหลัก ที่สามารถควบคุมจากระยะไกลมีทั้งที่บินด้วยตัวเอง และการบังคับด้วยคลื่นวิทยุ ดังรูปที่ 2.1 เราสามารถใช้ประโยชน์จากมัลติคอปเตอร์ ได้อย่างหลากหลาย ด้วยความสามารถเฉพาะตัวและขนาดที่กระทัดรัด ส่วนประกอบของมัลติคอปเตอร์ จะมีมอเตอร์ขับเคลื่อน โดยมอเตอร์ชนิดนี้ จะไม่มีแปลงถ่วงตัวจนลวดจะอยู่หนึ่ง แต่ส่วนที่เป็นแม่เหล็กจะหมุนแทน โดยจะมีสปริงคอนโทรล ทำหน้าที่ควบคุมความเร็วของมอเตอร์ โดยจะรับคำสั่งมาจากบอร์ด และจ่ายไฟไปยังมอเตอร์โดยจะมีใบพัดติดอยู่ที่มอเตอร์ โดยเราสามารถเลือกขนาดของใบพัดได้ตามต้องการ รวมไปถึงมีเครื่องรับสัญญาณวิทยุหรือ Receiver เป็นอุปกรณ์รับสัญญาณจากเครื่องส่ง และแปลงสัญญาณเพื่อไป ควบคุมการทำงานของมอเตอร์เพื่อให้มอเตอร์ทำงาน รวมไปถึง มีบอร์ด APM ใช้ในการสั่งการในการบังคับมัลติคอปเตอร์

มัลติคอปเตอร์ มีด้วยกันหลายแบบโดยแบ่งแยกกันด้วยจำนวน ของใบพัดมัลติคอปเตอร์ส่วนใหญ่ จะมี 4 ใบพัด 6 ใบพัด หรือ 8 ใบพัด จำนวนใบพัดนั้นมีส่วนในการควบคุมเป็นอย่างมาก มัลติคอปเตอร์ที่มีจำนวนใบพัดน้อย จะควบคุมให้หนึ่งได้ยากกว่า มัลติคอปเตอร์ที่มีใบพัดมาก โดยมัลติคอปเตอร์ที่มี 4 ใบพัดหรือ 8 ใบพัดจะสามารถรับน้ำหนักได้มากกว่า ส่วนประกอบของมัลติคอปเตอร์หลักๆ มี

- 1) Flight Control Board เป็นส่วนสำคัญในการควบคุม มัลติคอปเตอร์ เป็นตัวควบคุมส่วนกลางที่จะคอยรับคำสั่ง เพื่อไปประมวลผลและส่งคำสั่งไปยังมอเตอร์แต่ละตัว จะคอยควบคุมความแรงในการหมุนของแกนต่างๆ
- 2) Electronic Speed Controller เป็นอุปกรณ์ที่ใช้ควบคุมมอเตอร์แต่ละตัว ซึ่งจะรับคำสั่งมาจาก Flight Control Board แล้วจัดการจ่ายไฟเลี้ยงจาก Battery ไปยังมอเตอร์ตามคำสั่งโดยจะมีการจ่าย กระแสไฟน้อยหรือมากขึ้นอยู่กับคำสั่ง ของ Flight Control Board
- 3) Motors มอเตอร์ของ มัลติคอปเตอร์ ใช้มอเตอร์ชนิด Brushless คือเป็นมอเตอร์ที่ไม่มีแปลงถ่วง มอเตอร์ชนิดนี้ขดลวดจะอยู่หนึ่ง แต่ส่วนที่เป็นแม่เหล็กจะหมุนแทน โดย

จะมีสองแบบคือ หมุนนอกและหมุนใน แต่ถ้าเป็นมัลติคอปเตอร์ที่มีขนาดใหญ่ส่วนมากจะใช้แบบ หมุนนอกครับ สาเหตุที่ต้องใช้มอเตอร์แบบนี้เพราะมีขนาดเบาและกินพลังงานน้อยทำให้สามารถใช้งานได้นานขึ้น

- 4) Propellers ในส่วนนี้คือใบพัดจะมีด้วย กันหลายรูปแบบ โดยจะต่างกันที่ขนาด ความยาวซึ่งต้องเลือก ให้เหมาะสมกับ มัลติคอปเตอร์ให้มากที่สุด
- 5) Transmitter / Receiver เป็นส่วนการบังคับด้วย สัญญาณวิทยุโดยจะรับคำสั่งมาจาก ภายนอกพื้นดิน โดยในส่วนนี้ต้องมีช่องสัญญาณอย่างน้อย 4 ช่องสัญญาณ
- 6) Battery แบตเตอรี่ที่ใช้กับ มัลติคอปเตอร์ จะใช้ชนิด ลิเทียม-พอลิเมอร์ (Lithium Polymer) สาเหตุที่ใช้แบตเตอรี่ชนิดนี้ในการสร้าง มัลติคอปเตอร์ ก็เพราะมีน้ำหนักเบาและมีประสิทธิภาพสูง



รูปที่ 2.1 ตัวอย่างมัลติคอปเตอร์

มัลติคอปเตอร์แบบหกใบพัดดังรูปที่ 2.1 ที่ควบคุมโดยใช้ระบบวิทยุ และระบบจีพีเอส ในการควบคุม เพื่อนำไปใช้ในการถ่ายภาพทางอากาศ



รูปที่ 2.2 ตัวอย่างรีโมท

รีโมทที่ใช้ในการควบคุม มัลติคอปเตอร์ ดังรูปที่ 2.2 โดยใช้สัญญาณคลื่นวิทยุ

2.2 ราวเบอร์พาย



รูปที่ 2.3 ตัวอย่าง Board Raspberry Pi

บอร์ดราวเบอร์พาย หรือ คอมพิวเตอร์ขนาดเล็ก ที่ใช้ระบบปฏิบัติการ ลินุกซ์ในการประมวลผล ดังรูปที่ 2.3

2.2.1 บอร์ดราสเบอร์รี่พาย

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ ดังรูปที่ 2.3 สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด ราสเบอร์รี่พาย รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ โดยติดตั้งบน SD Card บอร์ด ราสเบอร์รี่พาย นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

2.2.2 คุณสมบัติทางเทคนิคของบอร์ดราสเบอร์รี่พาย

บอร์ดราสเบอร์รี่พาย ปัจจุบันมีด้วยกัน 2 โมเดล คือ โมเดล A และ โมเดล B ดังรูปที่ 2.4 ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน รายละเอียดดังตารางที่

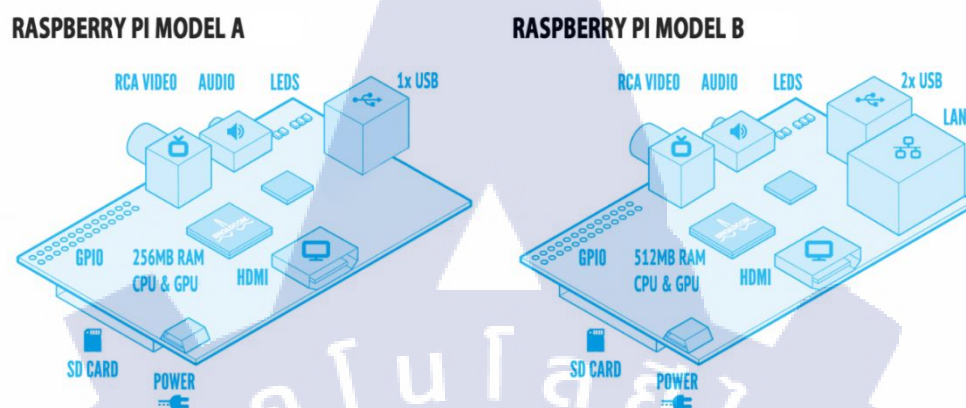
2.1

ตารางที่ 2.1 คุณสมบัติราสเบอร์รี่พาย

	โมเดล A	โมเดล B (Revision 2)
System on a chip (SoC)	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM and Single USB Port)	
CPU	700MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)	
GPU	Broadcom VideoCore IV @ 250 MHz OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1, 1080p 30 h.264/MPEG-4 AVC high-profile decoder and encoder	

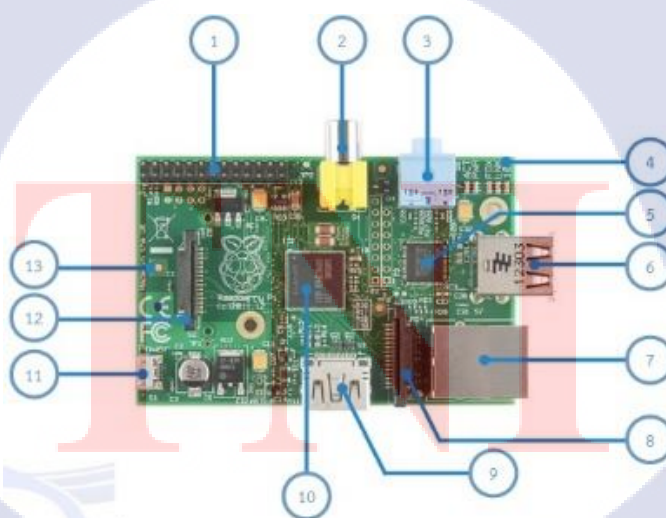
Memory (SDRAM)	256 MB (Shared with GPU)	512 MB (Shared with GPU)
USB 2.0 Ports	1(direct form BCM2835)	2 (via the build in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of RPF designed camera module (ออกแบบมาให้เชื่อมต่อกับ Raspberry Pi Camera Module โดยเฉพาะ)	
Video Outputs	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards. (มีทั้งสองแบบ คือ แบบ RCA และแบบ HDMI)	
Audio Outputs	3.5 mm jack, HDMI, and as of revision 2 boards, I ² S audio (also potentially for audio input)	
Onboard storage	SD/ MMC/ SDIO card slot (3.3V card power support only)	
Onboard network	None	10/100 Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-level peripherals Low-level peripherals	8 x GPIO, UART, I ² C Bus, SPI Bus with two chip selects, I ² S audio +3.3V, +5V, Ground	
Power ratings	300 mA (1.5 W)	700 mA (3.5 W)
Power source	5 Volt via Micro USB or GPIO header	
Size	85.60 mm x 53. Mm (3.370 inch x 2.125 inch)	
Weight	45 g. (1.6 oz.)	

2.2.3 ตัวอย่างโครงสร้างบอร์ดราสเบอร์รี่พาย ทั้ง 2 โมเดล



รูปที่ 2.4 ตัวอย่าง โครงสร้างบอร์ด Raspberry

2.2.4 ส่วนประกอบของบอร์ด Raspberry Pi (Model B)



รูปที่ 2.5 ตัวอย่างบอร์ด Raspberry

ส่วนประกอบของบอร์ด ราสเบอร์รี่พาย มีส่วนประกอบหลัก ๆ อยู่ 13 ส่วน ดังนี้ดังรูปที่ 2.5

- 1) พอร์ต GPIO ซึ่งในโมเดล A และ B (Revision 1) ทุก Pin จะเหมือนกัน แต่โมเดล B (Revision 2) จะแตกต่างกัน รายละเอียดดังรูปที่ 2.6 และ 2.7

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND		10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

รูปที่ 2.6 ตัวอย่าง PIN บอร์ด Raspberry Pi V1 รูปที่ 2.7 ตัวอย่าง PIN บอร์ด Raspberry Pi V2

- 2) พอร์ตเชื่อมต่อสัญญาณภาพออกแบบ RCA ตัวอย่างของสายที่เชื่อมต่อแสดงดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างสาย RCA

- 3) จุดเชื่อมต่อสัญญาณเสียงขนาด 3.5 มิลลิเมตร
4) LED แสดงสถานะของบอร์ด อยู่ภายในบริเวณกรอบสีแดง ดังรูปที่ 2.9



รูปที่ 2.9 ตัวอย่าง LED สถานะ Raspberry Pi

- 5) ชิพควบคุม แลน (LAN Controller)
- 6) พอร์ต USB 2.0 จำนวน 2 พอร์ต
- 7) พอร์ต RJ-45 Ethernet LAN 10/100 Mbps
- 8) พอร์ตซีเอสไอ (Camera Serial Interface)สำหรับเชื่อมต่อโมดูลกล้องดังรูปที่ 2.10



รูปที่ 2.10 ตัวอย่าง Module Camera

- 9) พอร์ต HDMI สำหรับเชื่อมต่อสัญญาณภาพและเสียง ตัวอย่างสาย HDMI และ ตัวแปลง HDMI to VGA แสดงดังรูปที่ 2.11 และ 2.12

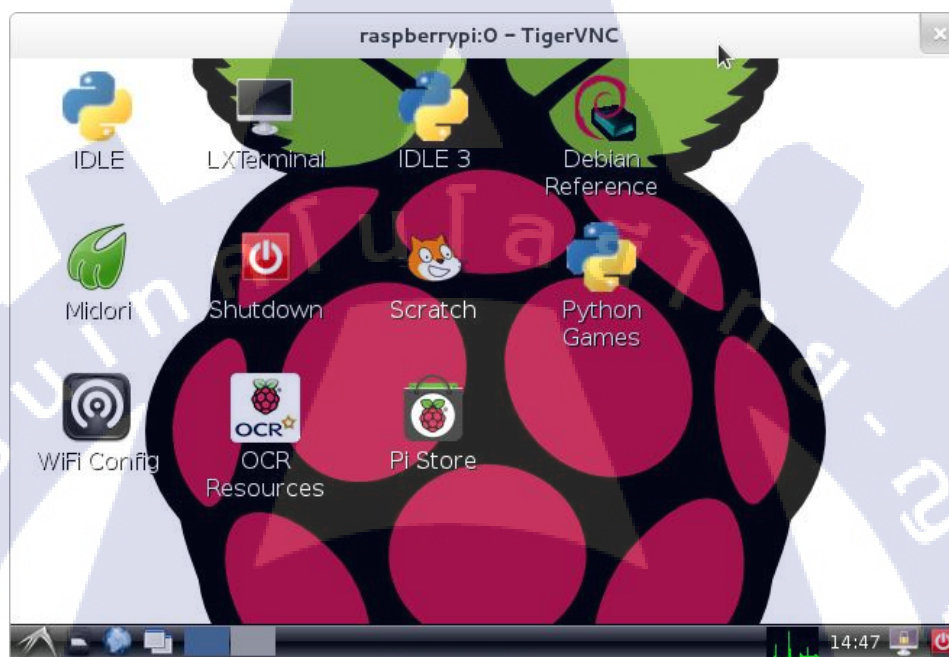


รูปที่ 2.11 ตัวอย่าง สาย HDMI



รูปที่ 2.12 ตัวอย่าง HDMI To VGA

- 10) ชิป Broadcom BCM2835 ARM11 700MHz
- 11) พอร์ต Micro USB Power สำหรับเป็นไฟเลี้ยงวงจรบอร์ด Raspberry Pi
- 12) พอร์ต ดีเอสไอ (Display Serial Interface) ใช้สำหรับต่อจอแสดงผล
- 13) ช่องเสียบ SD Card อยู่บริเวณด้านล่างของบอร์ด



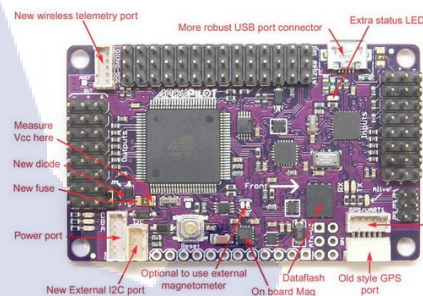
รูปที่ 2.13 หน้าจอ Raspberry Pi

เดสทอปของบอร์ดราสเบอร์รี่พาย ที่ใช้ระบบปฏิบัติการลินุกซ์ ในการประมวลผลและสั่งการทำงานของระบบต่าง ๆ ดังรูปที่ 2.13

2.3 Ardu Pilot Mega



รูปที่ 2.14 บอร์ด APM 2.6



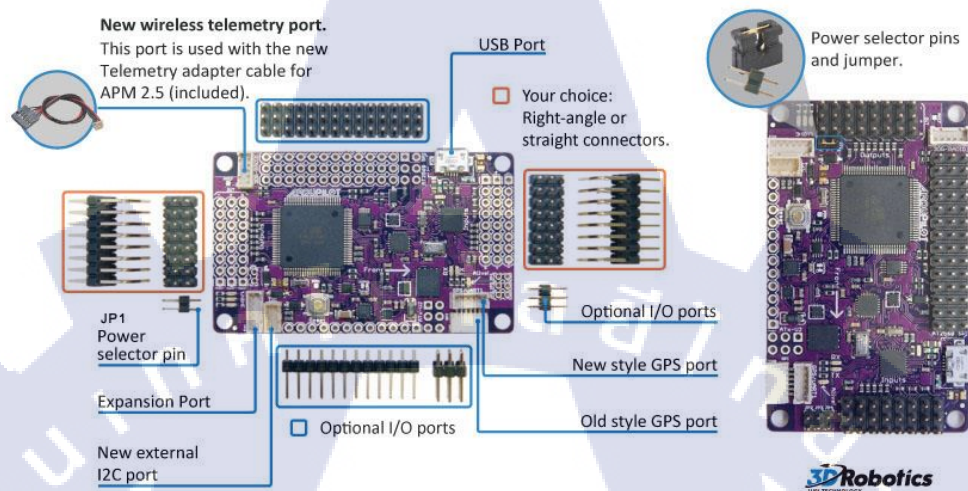
รูปที่ 2.15 รายละเอียด APM 2.6

APM 2.6 บอร์ด ควบคุมยานในรูปแบบต่าง ๆ ดังรูปที่ 2.14 และ 2.15 โดยสามารถลง Firmware เพื่อปรับการทำงานได้ทั้ง รถ เรือ และอากาศยานทุกรูปแบบตัว APM 2.6 แยกตัว เจ็มทิส อิเล็กทรอนิกส์ ออกไปใช้บน GPS Module จึงสามารถเลือกติดตั้งเพื่อหลบสัญญาณรบกวนจากมอเตอร์และอุปกรณ์ส่งสัญญาณอื่นๆ ได้ง่ายและมีประสิทธิภาพในการจับสัญญาณที่รวดเร็ว

บอร์ด APM นี้สามารถควบคุมเครื่องบิน UAV Multicopter และ เฮลิคอปเตอร์ รองรับการ ติด GPS ทำการบินไปยังจุดต่างๆ อัตโนมัติ หรือกลับมายังจุด Take off ได้เอง โดยบอร์ดสามารถรับการเชื่อมต่อกับ อุปกรณ์เสริมหลายอย่าง เช่น

- 1) Minimosd เป็นอุปกรณ์แสดงสถานการณ์บินผ่านหน้าจอคอมพิวเตอร์
- 2) APM Power Module เป็นอุปกรณ์แสดงสถานะแบตเตอรี่
- 3) Radio Telemetry เป็นอุปกรณ์เชื่อมต่อคอมพิวเตอร์ไร้สาย
- 4) GPS เป็นอุปกรณ์เชื่อมต่อสัญญาณดาวเทียม

บอร์ด APM 2.6 นี้ได้ถูกพัฒนามาจาก APM 2.5 โดยในบอร์ดตัวนี้จะไม่มีเข็มทิศ ฝังอยู่ในบอร์ด โดยเราจะต้องติดตั้ง เข็มทิศและGPS เพิ่มโดยใช้อุปกรณ์ภายนอกเพื่แก้ปัญหการรบกวน สัญญาณจากมอเตอร์ ดังรูปที่ 2.16 และ 2.17



รูปที่ 2.16 หลักการเชื่อมต่อ APM 2.6



รูปที่ 2.17 อุปกรณ์เชื่อมต่อ APM 2.6

2.4 โปรแกรมมิชชันแพลนเนอร์



รูปที่ 2.18 ตัวอย่าง Mission Planner

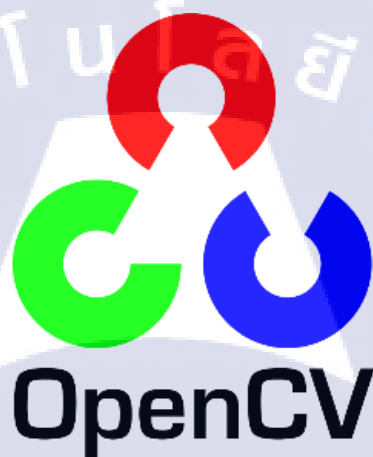
โปรแกรม มิชชันแพลนเนอร์ เป็นระบบควบคุมภาคพื้นดินใช้สำหรับควบคุมบอร์ด APM ดังรูปที่ 2.18 สามารถใช้งานได้กับวินโดว แท่นั้น โดยสามารถวางแผนการทำภารกิจของ มัลติคอปเตอร์ล่วงหน้าได้ ในหน้าโปรแกรมจะมีการแสดงผลของข้อมูลต่างๆ ไม่ว่าจะเป็นแผนที่ รวมไปถึงระดับความสูงและความเอียงของ มัลติคอปเตอร์โดยข้อมูลทั้งหมดนี้สามารถนำมาใช้ในการประมวลผลการบินเพื่อเพิ่มความปลอดภัยของมัลติคอปเตอร์



รูปที่ 2.19 ตัวอย่างการกำหนด Waypoint

Mission Planner สามารถกำหนด Waypoint เพื่อวางแผนในการบินได้ ดังรูปที่ 2.19 โดยการกำหนดจุดที่แผนที่ ในการการทำ Waypoint สามารถกำหนดได้ว่าต้องการใช้ความสูงเท่าไรและยังสามารถ กำหนดจุดให้มอเตอร์คอปเตอร์ลงจอด ตามจุดที่กำหนดได้

2.5 โอเพ่น ซีวี



รูปที่ 2.20 เครื่องหมายโอเพ่น ซีวี

ปัจจุบัน โอเพ่น ซีวี เป็น Open Source Library ที่กำลังเป็นที่นิยมสำหรับการพัฒนาโปรแกรม ดังรูปที่ 2.20 ทาง Computer Vision ทั้งนี้ Open CV นั้นไม่ได้เพิ่งเริ่มเกิดขึ้นในระยะหลังๆ หากแต่เริ่มต้นมาตั้งแต่ ปี 1999 ที่บริษัท Intel โดย Gary Bradski มีความประสงค์ที่จะสนับสนุนการวิจัยทาง Computer Vision ให้สามารถสร้างโปรแกรมประยุกต์ที่สามารถนำไปใช้ทางการค้าได้ ทาง Intel ทำเพื่อสร้างความต้องการ computer ที่ทรงพลังมากขึ้นกว่าเดิม ต่อมา Vadim Pisarevsky ได้เข้าร่วมกับ Gary บริหารทีม Open CV รัสเซีย ของ Intel หลังจากนั้น ทีมงาน Open CV ได้ย้ายไปบริษัท และทีมวิจัยต่างๆ เช่น Willow Garage ที่ทำการวิจัยทางหุ่นยนต์

ในปี 2008 ทาง Willow Garage ได้เล็งเห็นว่าหุ่นยนต์ต้องการความสามารถในการมองเห็น และความต้องการนี้เพิ่มขึ้นอย่างรวดเร็ว ทำให้กลับมาสนับสนุน Open CV อีกครั้งหนึ่ง ทำให้ ระยะ 2-3 ปีหลังมานี้ เราได้พบชื่อ Open CV ในการพัฒนาโปรแกรมต่างๆมากขึ้น

โอเพ่น ซีวี พัฒนาขึ้นโดยได้รับการสนับสนุนจาก Intel Corporation จำกัด เป็นซอฟต์แวร์แบบเปิดเผยรหัส (Library Open Source) สำหรับใช้ในการประมวลผลภาพ (Image Processing) เพื่อให้สามารถนำไปต่อยอดพัฒนาโปรแกรมต่าง ๆ ได้ง่าย ใช้ได้บนระบบปฏิบัติการที่เป็น Linux และ Microsoft Windows และสามารถพัฒนาโปรแกรมได้หลากหลายภาษา

2.5.1 การนำ Open CV มาใช้ประโยชน์

เพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) ก็สามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องVDO หรือ VDO File เป็นไปได้อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ

ฟังก์ชันต่างๆของ Open CV จะสามารถเรียกใช้งานได้จะต้องมีการเรียก ไฟล์ส่วนหัว (Header file) และลิงค์ (Link) ไบเบรารีต่างๆ รวมถึง DLL (Dynamic Link Library)

2.6 ไพทอน

ไพทอน คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่มีขีดติดกับแพลตฟอร์ม คือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น OpenSource ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน ดังรูปที่ 2.22

ไค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ คือจะประมวลผลไปที่ละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันแรกคือ เวอร์ชัน 0.9.0 ออกมาเมื่อปี 2533 และเวอร์ชันปัจจุบันคือ 2.5.2

2.6.1 คุณลักษณะเด่นของภาษา Python

- 1) สนับสนุนแนวแบบคิออปเจกต์โอเรียนเทด หรือ OOP (Object Oriented Programming)
- 2) เป็น Open Source
- 3) โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการได้หลากหลาย
- 4) สนับสนุนเทคโนโลยี COM ของ Ms-windows
- 5) Python รวมมาตรฐานการอินเตอร์เฟส Tkinter ซึ่งสนับสนุนบนระบบ X windows และ Ms-windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ
- 6) เป็น Dynamic typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
- 7) มี Built-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ ใน Python ประกอบด้วยลิสต์ ดิกชันนารี และสตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
- 8) มีเครื่องมือต่างๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนค่า เป็นต้น
- 9) มีมอดูลสำหรับจัดการ Regular Expression
- 10) มีมอดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM และ Image และ CORBA และ ORBs และ XML เป็นต้น
- 11) จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
- 12) อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโค้ดภาษา C/C++ ได้
- 13) อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Libray เพื่อใช้ร่วมกับ Python
- 14) มีมอดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก โปรเซส เรด regular และ expression และ xml และ GUI และอื่นๆ
- 15) ประกอบด้วยมอดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets และทำหน้าที่เป็น CGI Script ครอบงำใช้งานคำสั่ง FTP และ Gopher และ XML และอื่นๆอีกมาก
- 16) สามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
- 17) มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL และ Sybase และ Oracle และ Informix และ ODBC และอื่นๆ

18) มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในรูปแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ

19) มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์

20) มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer

21) มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

```
import thread
import threading
from time import sleep

def myfunction(arg1,arg2):
    for i in range(10):
        sleep(arg2)
        print "running: %s - numner %d" %(arg1,i)
    print "end threading" + ' ' + arg1

def myfunction2(arg1,arg2):
    for i in range(10):
        sleep(arg2)
        print "running: %s - numner %d" %(arg1,i)
    print "end threading" + ' ' + arg1

def main():
    thread.start_new_thread(myfunction,("Prog01",1))
    thread.start_new_thread(myfunction2,("Prog02",0.5))

main()
```

รูปที่ 2.21 ตัวอย่างโค้ดไพทอน

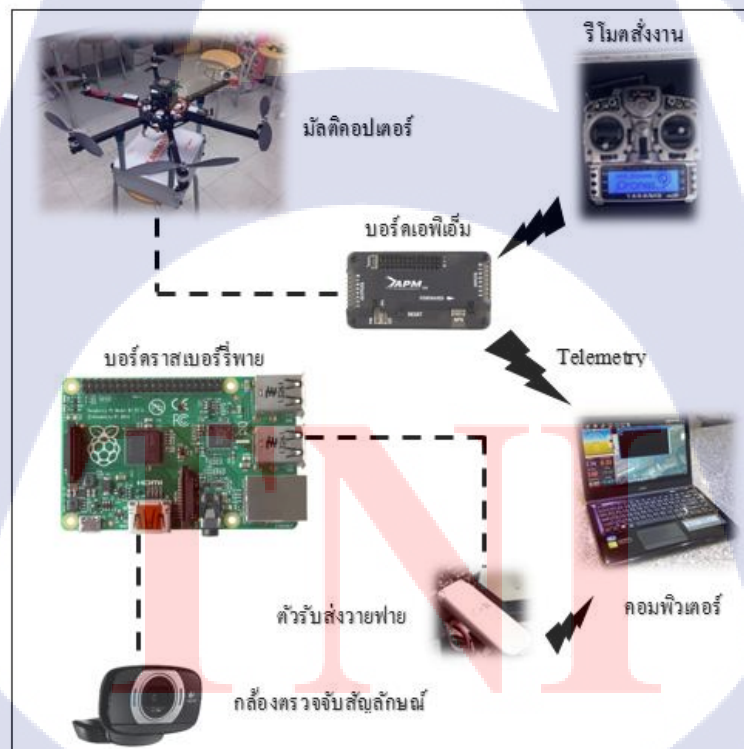
บทที่ 3

การพัฒนาระบบช่วยจอดมัลติคอปเตอร์

ในส่วนดำเนินงานนี้ ผู้จัดทำนำแนวคิดและทฤษฎีต่าง ๆ จากที่ได้ศึกษามาแล้ว ในบทที่ 2 มาประยุกต์ใช้ในการพัฒนางาน

3.1 มัลติคอปเตอร์และอุปกรณ์ที่ใช้

3.1.1 แผนผังระบบช่วยลงจอดของมัลติคอปเตอร์



รูปที่ 3.1 ไคอะแกรมระบบช่วงลงจอด

ไต่จะแกระบบการทำงานของมัลติคอปเตอร์ดังรูปที่ 3.1 โดยใช้มัลติคอปเตอร์แบบหกใบพัดในการทดสอบ เริ่มด้วยการติดตั้งบอร์ด เอพีเอ็มบนมัลติคอปเตอร์เพื่อใช้ในการควบคุม และติดตั้งบอร์ดราสเบอร์รี่พาย เพื่อใช้ในการตรวจจับค่าสี โดยทำการลงระบบปฏิบัติการลินุกซ์ และทำการลงโปรแกรมไพทอน และโอเพ่นซีวี เพื่อใช้ในการตรวจจับค่าสี โดยทำการติดตั้งกล้อง ใว้ได้มัลติคอปเตอร์ เพื่อใช้ในการจับค่าสี เมื่อมัลติคอปเตอร์ บินมายังจุดที่กำหนดกล้องจะทำการค้นหาจุดลงจอด เมื่อเจอจุดลงจอดแล้วจะทำการส่งค่าไป ที่คอมพิวเตอร์ผ่านระบบ Wi - Fi โดยจะส่งคำว่า Landing และค่าพารามิเตอร์จุดศูนย์กลาง ของตำแหน่งที่กำหนด เมื่อค้นหาจุดที่กำหนดครบแล้วนักบินจึงทำการส่งงานไปที่ บอร์ดเอพีเอ็มเพื่อให้มัลติคอปเตอร์ ลงจอดตามจุดที่กำหนด

ขั้นตอนการทดสอบระบบช่วยลงจอดของมัลติคอปเตอร์

- 1) มัลติคอปเตอร์บินอยู่บนอากาศ ทำการตรวจจับสัญญาณที่กำหนด
- 2) เจอสัญญาณที่กำหนด ทำการตรวจจับค่าสีและวัดพื้นที่
- 3) ส่งค่า Landing มาที่คอมพิวเตอร์
- 4) นักบินสั่ง มัลติคอปเตอร์ลงจอดตามจุดที่กำหนด

3.1.2 อุปกรณ์ที่ใช้

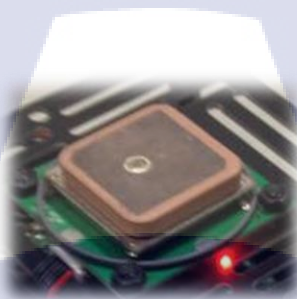
- 1) มัลติคอปเตอร์หกใบพัด



รูปที่ 3.2 ตัวอย่างมัลติคอปเตอร์

มัลติคอปเตอร์แบบหกใบพัดดังรูปที่ 3.2 ที่ควบคุมโดยใช้บอร์ดคอนโทรลเลอร์ (APM) ผ่านระบบวิทยุ โดยมีจีพีเอสในการบอกตำแหน่ง ของมัลติคอปเตอร์ เพื่อนำไปใช้ในการถ่ายภาพทางอากาศ โดยมีบอร์ดราสเบอร์รี่พาย ใช้ในการค้นหาจุดลงจอดโดยการจับค่าสีตามที่กำหนด และมี UAV Telemetry ใช้ในการเชื่อมต่อกับคอมพิวเตอร์ผ่านโปรแกรม Mission Planner เพื่อใช้ในการตรวจสอบระบบ และทำการกำหนดจุด Waypoint

2) อุปกรณ์รับสัญญาณจีพีเอส



รูปที่ 3.3 ตัวรับดาวเทียม

ตัวรับสัญญาณจีพีเอส ดังรูป 3.3 ใช้ในการเชื่อมต่อกับดาวเทียม เพื่อทำการกำหนดจุดตำแหน่งที่อยู่ และส่งค่าไปยังบอร์ดเอพียเอ็ม เพื่อให้บอร์ดเอพียเอ็ม ส่งค่าพิกัดไปที่ คอมพิวเตอร์ หรือใช้ในการกำหนดจุดเวฟอย

3) บอร์ดราสเบอร์รี่พาย



รูปที่ 3.4 บอร์ด Raspberry Pi 2

บอร์ดราสเบอร์รี่พาย ใช้ระบบปฏิบัติการลินุกซ์ ดังรูปที่ 3.4 ในการทำงาน โดยจะลงโอเพ่นซีวี และไฟทอน โดยทำการเชื่อมต่อกับกล้อง เพื่อทำการค้นหาจุดลงจอด เมื่อกล้องเจอค่าสีที่กำหนดแล้วจะทำการส่งค่าไปที่คอมพิวเตอร์ เพื่อทำการส่งลงจอดตามจุดที่กำหนด

4) อุปกรณ์รับส่ง (Wi - Fi)



รูปที่ 3.5 ตัวรับส่ง ไร้สาย

ตัวรับส่งไร้สายใช้ในการเชื่อมต่อ บอร์ดราสเบอร์รี่พาย ดังรูปที่ 3.5 กับคอมพิวเตอร์ เพื่อใช้ในการรับส่งค่าจาก บอร์ดราสเบอร์รี่พาย ไปยังคอมพิวเตอร์ เพื่อใช้ในการแสดงผล

5) กล้องเว็บแคม



รูปที่ 3.6 กล้องเว็บแคม

กล้องใช้ในการตรวจจับค่าสี โดยจะทำการติดตั้งอยู่ข้างใต้ มัลติคอปเตอร์ ดังรูปที่ 3.6 และทำการเชื่อมต่อไปยังบอร์ด ราสเบอร์รี่พาย เพื่อใช้งานกับโปรแกรม ไพทอนและโอเพ่นซีวี

5) รีโมทควบคุม



รูปที่ 3.7 รีโมทควบคุม

รีโมทที่ใช้ควบคุมมัลติคอปเตอร์ ดังรูปที่ 3.7 ที่สั่งงานโดยใช้ คลื่นวิทยุ ใช้ในการควบคุมมัลติคอปเตอร์ และสั่งงานบอร์ดเอพียเอ็ม ให้ทำงานตามที่ต้องการ และส่งลงจอดในจุดที่กำหนดไว้

6) คอมพิวเตอร์

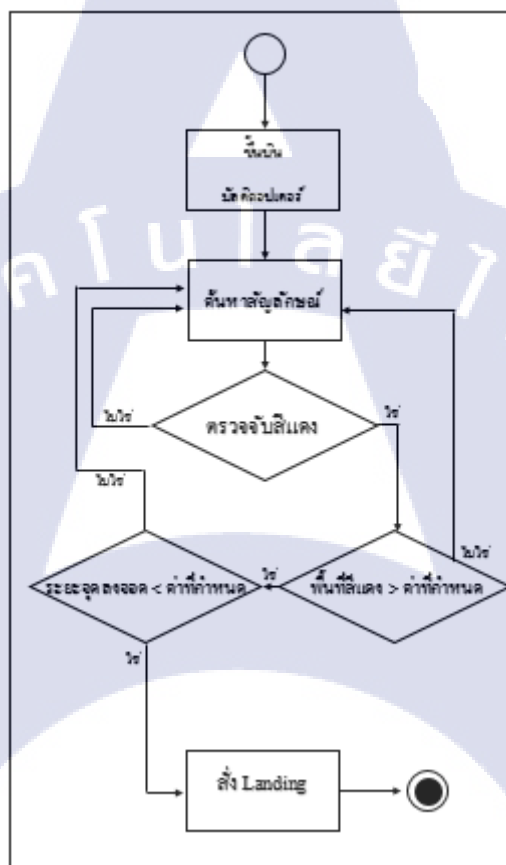


รูปที่ 3.8 คอมพิวเตอร์

คอมพิวเตอร์ใช้ในการแสดงผล ดังรูปที่ 3.8 ที่รับค่ามาจาก บอร์ดราสเบอร์รี่พาย โดยผ่านโปรแกรม Mission Planner โดยจะมีการรายงานผลต่าง ๆ เช่น ค่าความสูง ค่าความเร่ง หรือ พิกัดแผนที่ เป็นต้น

3.2 ระบบตรวจจับจุดลงจอด

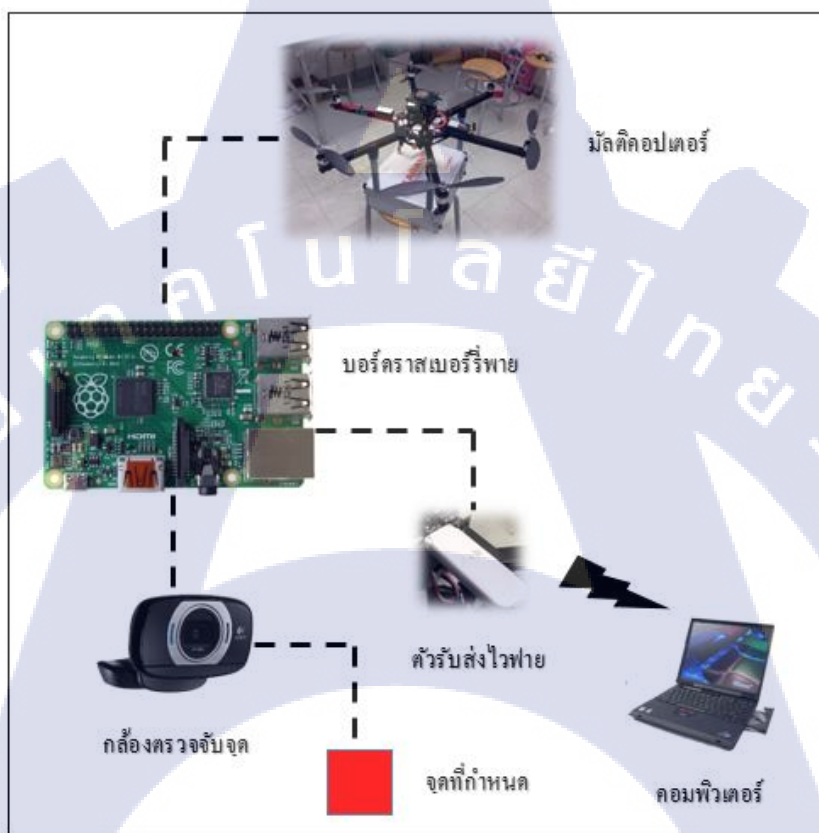
3.2.1 โพลวัชารต์ของระบบตรวจจับจุดลงจอด



รูปที่ 3.9 แผนผังระบบตรวจจับจุดลงจอด

จากแผนผัง จะเห็นได้ว่าเมื่อมัลติคอปเตอร์ขึ้นบินจะทำการค้นหาสัญญาณ โดยการตรวจจับค่าสี เมื่อพบสัญญาณจะทำการคำนวณค่าสีถ้าตรงตามเงื่อนไขจะส่งคำสั่งไปคำนวณจุดกึ่งกลาง ถ้าไม่ผ่านเงื่อนไขจะวนกลับไปค้นหาสัญญาณ ถ้าผ่านเงื่อนไขจะทำการส่ง Landing ถ้าไม่ผ่านเงื่อนไข จะวนกลับไปปรับใหม่เมื่อค้นหาตำแหน่งลงจอดพบจะทำการตรวจค่าความคลาดเคลื่อนของจุดลงจอด ถ้าตรงกับที่กำหนดจะทำการส่งค่ามายังคอมพิวเตอร์ แต่ถ้าไม่สามารถผ่านเงื่อนไขที่กำหนดได้ จะทำการกลับไปค้นหาตำแหน่งจุดลงจอดใหม่

1) ไตอะแกรมการเชื่อมต่อระบบตรวจจับสัญญาณ



รูปที่ 3.12 ไตอะแกรมระบบค้นหาสัญญาณ

ไตอะแกรมระบบการทำงานของการทำงานตรวจจับค่า ดังรูปที่ 3.12 โดยใช้มัลติคอปเตอร์แบบหกใบพัดในการทดสอบ เริ่มด้วยการติดตั้งบอร์ด ราสเบอร์รี่พาย เพื่อใช้ในการตรวจจับค่าสี โดยทำการลงระบบปฏิบัติการลินุกซ์ และทำการลงโปรแกรมไพทอน และโอเพ่นซีวี เพื่อใช้ในการตรวจจับค่าสี โดยทำการติดตั้งกล้อง ไปได้มัลติคอปเตอร์ เพื่อใช้ในการจับค่าสี เมื่อมัลติคอปเตอร์ บินมายังจุดที่กำหนดกล้องจะทำการค้นหาจุดลงจอด เมื่อเจอจุดลงจอดแล้วจะทำการส่งค่าไป ที่คอมพิวเตอร์ผ่านระบบ Wi - Fi โดยจะส่งคำว่า Landing และค่าพารามิเตอร์จุดศูนย์กลาง ของตำแหน่งที่กำหนด

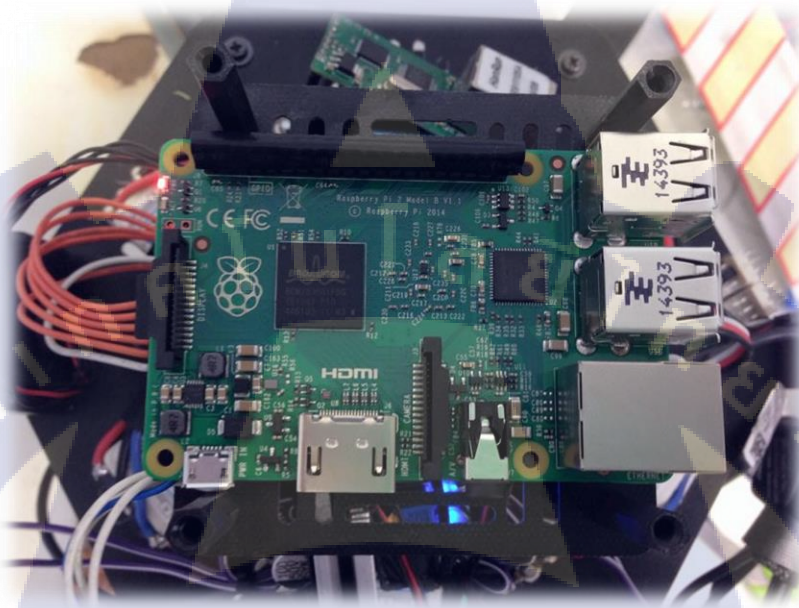
2) การทำงานของอุปกรณ์ตรวจจับจุดลงจอด



รูปที่ 3.13 มัลติคอปเตอร์ขณะลงจอดตามสัญลักษณ์

รูปภาพขณะที่มัลติคอปเตอร์ทำการลงจอด โดยทำการตรวจจับสัญลักษณ์ ดังรูปที่ 3.13 โดยเมื่อตรวจเจอสัญลักษณ์แล้ว จะทำการส่งคำว่า Landing ไปที่คอมพิวเตอร์ เมื่อนักบินได้รับแล้วจะทำการสั่งให้มัลติคอปเตอร์ลงจอดตามจุดที่กำหนด

- การติดตั้งรหัสเบอร์รีพายบนมัลติคอปเตอร์



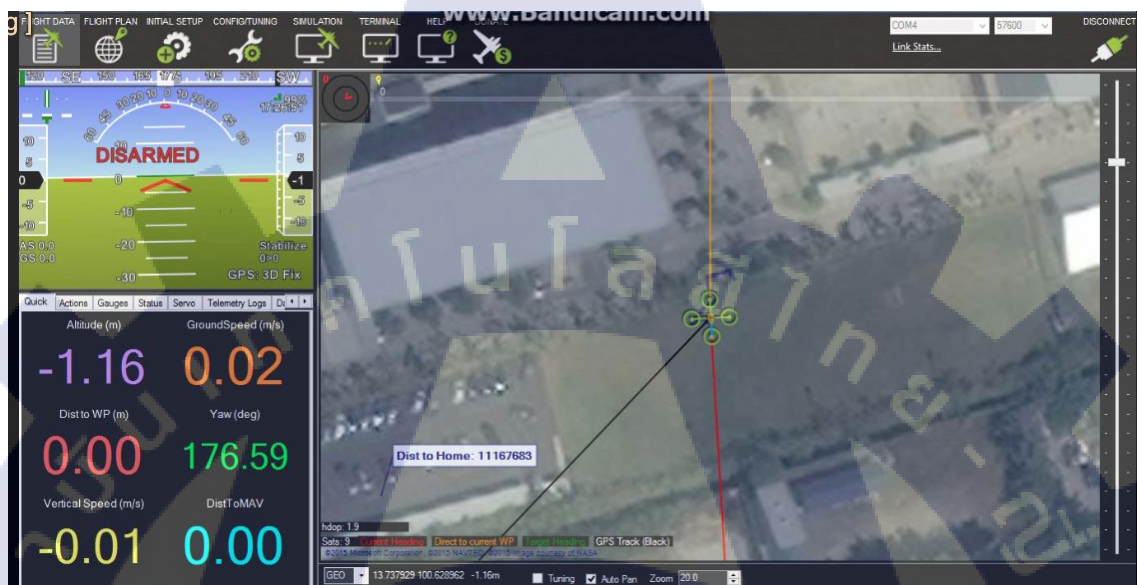
รูปที่ 3.14 รหัสเบอร์รีพายติดตั้งบนมัลติคอปเตอร์

รูปภาพการติดตั้ง บอร์ดรหัสเบอร์รีพาย บนมัลติคอปเตอร์ ดังรูปที่ 3.14 โดยใช้รหัสเบอร์รีพาย เวอร์ชัน B 2

TNI

TAHITI - NICHI INSTITUTE OF TECHNOLOGY

- โปรแกรมมิชชั่นแพลนเนอร์ ที่ใช้สั่งมัลติคอปเตอร์ให้ลงจอด

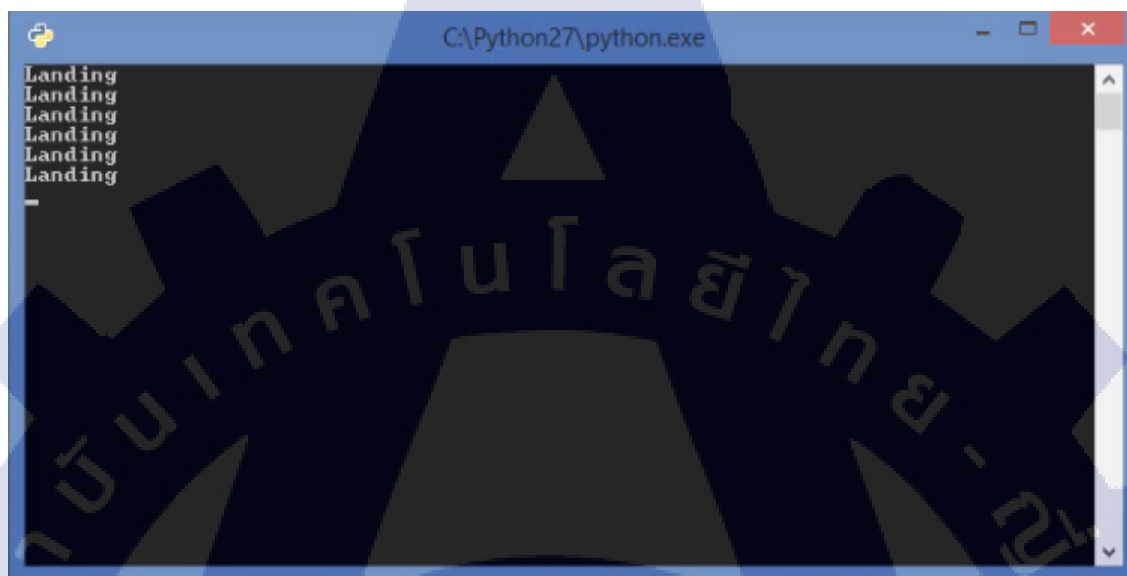


รูปที่ 3.15 โปรแกรมมิชชั่นแพลนเนอร์

หน้าต่างโปรแกรมมิชชั่นแพลนเนอร์ บนคอมพิวเตอร์ ดังรูปที่ 3.15 ใช้รายงานค่าต่าง ๆ ให้กับนักบินทราบ และกำหนดการทำงานต่าง ๆ ของมัลติคอปเตอร์

TNI

- หน้าต่างโปรแกรมรับค่า Landing บนคอมพิวเตอร์



รูปที่ 3.16 หน้าจอร์ับคำสั่ง Landing

หน้าต่างการรับคำสั่ง Landing บนคอมพิวเตอร์ ดังรูปที่ 3.16 โดยเมื่อนักบินเห็นคำสั่ง Landing จะทำการสั่งมัลติคอปเตอร์ให้ลงจอดตามจุดที่กำหนดทันที

บทที่ 4

ผลการทดสอบ

สำหรับบทนี้ ผู้พัฒนาได้ทำการทดสอบประสิทธิภาพการทำงานของ การตรวจสอบจุดลงจอดของมัลติคอปเตอร์ แล้วเก็บรวบรวมข้อมูลการทำงาน โดยแบ่งออกเป็น 2 แบบการควบคุม LOS และ RTL โดยมีการเก็บรวบรวม 3 ช่วงเวลาคือ เช้ากลางวันและเย็น โดยเก็บรวบรวม 3 หัวข้อคือ ระยะที่คลาดเคลื่อน และ ความแรงลง ในช่วงเวลาที่ทดลอง



รูปที่ 4.1 การทดลองระบบตรวจหาจุดลงจอด

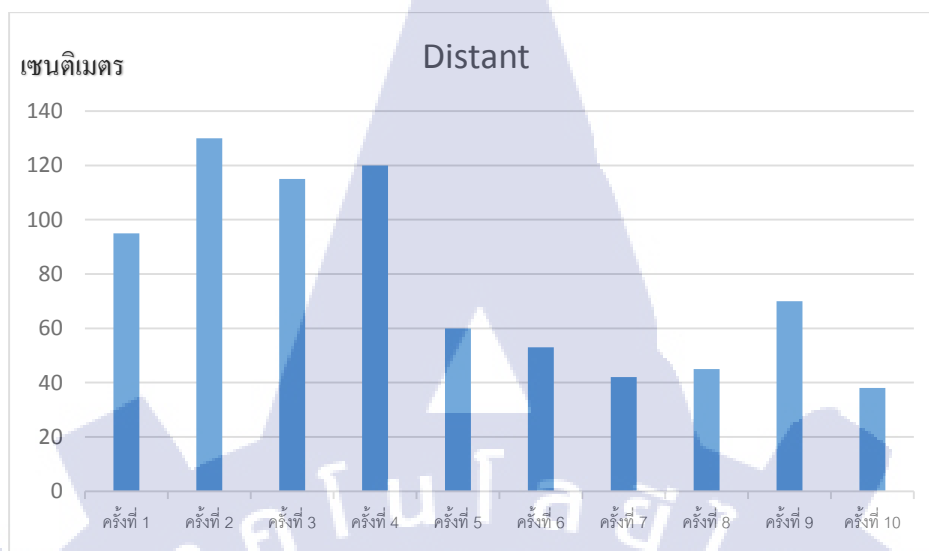
4.1 การทดลองแบบการควบคุมโดยใช้สายตา (Line of sight)

4.1.1 ตารางการทดลอง LOS

ตารางที่ 4.1 ตารางการทดลอง LOS

No	Distant	Wind
1	95 เซนติเมตร	แรง
2	130 เซนติเมตร	แรง
3	115 เซนติเมตร	ปานกลาง
4	120 เซนติเมตร	ปานกลาง
5	60 เซนติเมตร	น้อย
6	53 เซนติเมตร	น้อย
7	42 เซนติเมตร	น้อย
8	45 เซนติเมตร	น้อย
9	70 เซนติเมตร	ปานกลาง
10	38 เซนติเมตร	น้อย

4.1.2 กราฟแสดงค่า Distant ในการทดลองแบบ LOS



รูปที่ 4.2 กราฟค่า Distant แบบ LOS

กราฟการแสดงผลการทดลอง แบบ LOS และจัดบันทึกค่า Distant เพื่อเก็บเป็นประวัติและหาข้อสรุปของค่าความคลาดเคลื่อนในการทดลอง ค่าเฉลี่ย 76 เซนติเมตร

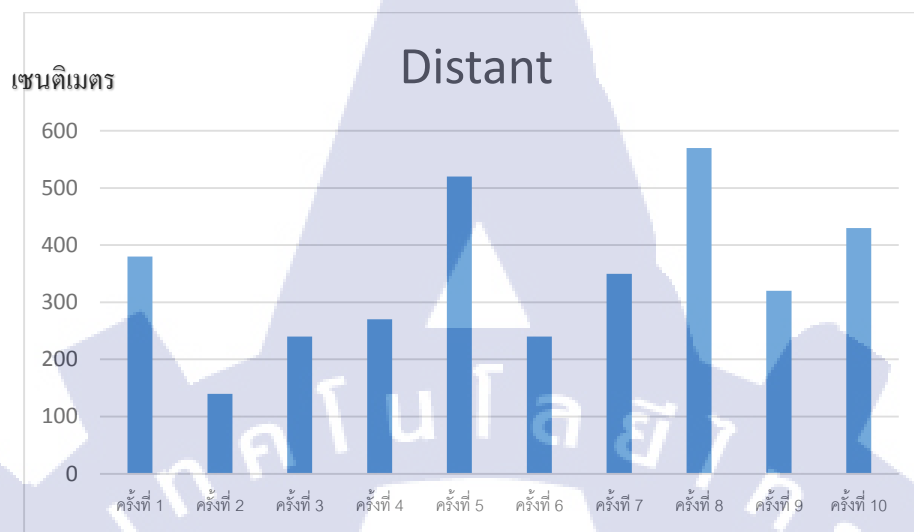
4.2 การทดลองแบบการลงจอดโดยใช้จีพีเอสในโหมดกลับจุดเริ่มต้น (Return to land : RTL)

4.2.1 ตารางการทดลอง RTL

ตารางที่ 4.2 ตารางการทดลอง RTL

No	Distant	Wind
1	380 เซนติเมตร	น้อย
2	140 เซนติเมตร	น้อย
3	240 เซนติเมตร	น้อย
4	270 เซนติเมตร	น้อย
5	520 เซนติเมตร	น้อย
6	240 เซนติเมตร	น้อย
7	350 เซนติเมตร	ปานกลาง
8	570 เซนติเมตร	แรง
9	320 เซนติเมตร	ปานกลาง
10	430 เซนติเมตร	แรง

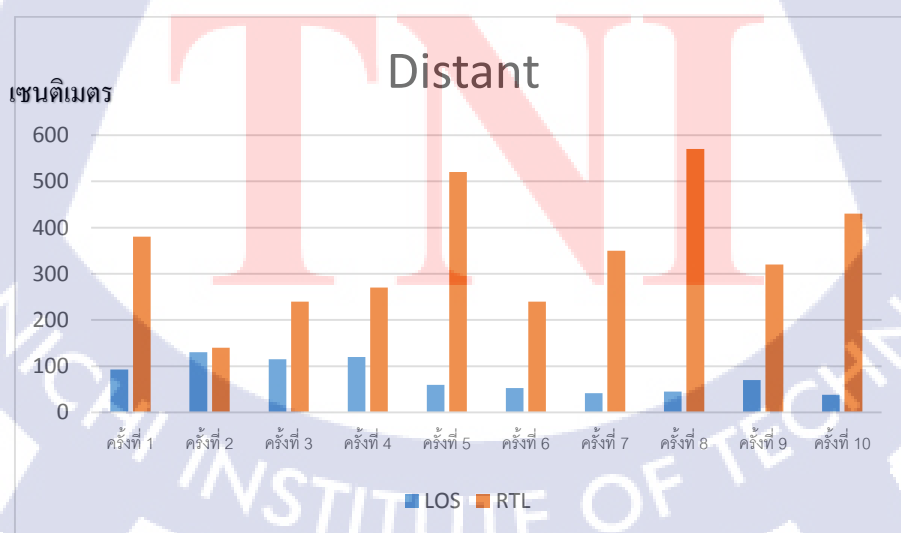
4.2.2 กราฟแสดงค่า Distant ในการทดลองแบบ RTL



รูปที่ 4.3 กราฟค่า Distant แบบ RTL

กราฟการแสดงผลการทดลอง แบบ RTL และจัดบันทึกค่า Distant เพื่อเก็บเป็นประวัติและหาข้อสรุปของค่าความคลาดเคลื่อนในการทดลอง ค่าเฉลี่ย 346 เซนติเมตร

4.3 เปรียบเทียบการทดสอบการลงจอดระหว่าง LOS กับ RTL



รูปที่ 4.4 กราฟค่า Distant แบบ LOS และ RTL

กราฟการแสดงผลการทดลอง แบบ LOS และ RTL และจัดบันทึกค่า Distant หรือจุดกึ่งกลาง สัญลักษณ์เพื่อเก็บเป็นประวัติและหาข้อสรุปของค่าความคลาดเคลื่อนในการทดลอง ค่าเฉลี่ย LOS เท่ากับ 76 เซ็นติเมตร และ ค่าเฉลี่ย RTL เท่ากับ 346 เซ็นติเมตร ในการทดสอบนั้นมีตัวแปรมากมาย ไม่ว่าจะเป็น แรงแลม สภาพอากาศ ในเวลาที่ทดสอบหรือค่าของแสงก็มีผลต่อการทดสอบ จึงต้องมีการตรวจเช็คค่าแสงอยู่ตลอดเวลา และมีการจัดบันทึกการทดสอบโดยละเอียด เพื่อหาค่าเฉลี่ยในการทดสอบ

จากผลทดสอบและค่าเฉลี่ยที่ทำการเก็บรวบรวม สรุปได้ว่าการทดลองแบบ LOS ได้ผลการทดลองสรุปได้ว่า การลงจอดของมัลติคอปเตอร์ มีความแม่นยำสูง สรุปค่า Distant เฉลี่ยได้ 76 เซ็นติเมตร และ ผลการทดลองแบบ RTL โดยใช้ GPS ในการกำหนดจุดลงจอด ได้ผลสรุปว่า มีความถูกต้องสูง แต่มีความแม่นยำน้อย และมี Distant เฉลี่ยได้ 346 เซ็นติเมตร มากกว่าระบบการลงจอดที่พัฒนา 270 เซ็นติเมตร แต่การทดลองทั้งสองแบบนี้ ก็มีตัวแปรต่าง ๆ มากมาย ไม่ว่าจะเป็น แรงแลม สภาพบรรยากาศ หรือค่าของแสง ก็มีผลทำให้เกิดความคลาดเคลื่อนของการลงจอดของมัลติคอปเตอร์

สรุปผลการทดสอบ การค้นหาจุดลงจอดของมัลติคอปเตอร์ โดยใช้การตรวจสอบสัญญลักษณ์ มีความสำคัญต่อ การลงจอดของมัลติคอปเตอร์เป็นอย่างมาก เนื่องจากมีค่าความคลาดเคลื่อนน้อย จึงมีประโยชน์สำหรับนักบินเป็นอย่างมาก และสามารถนำไปพัฒนาต่อ หรือนำไปใช้จริง ได้อย่างมีประสิทธิภาพ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุป

โครงการนี้เป็นการพัฒนาการลงจอดของมัลติคอปเตอร์ ด้วยการกำหนดจุดลงจอดด้วยโค้ดสีต่าง ๆ เช่นสีแดง เป็นต้น โดยใช้บอร์ดราสเบอร์รี่พาย ในการตรวจจับค่าสี ในการพัฒนานี้จะทำการเขียนโค้ด ไพทอน เพื่อตรวจจับค่าสีและทำการตรวจจับจุดกึ่งกลางของ วัตถุที่กำหนดไว้โดยจะทำการติดตั้งบอร์ด ราสเบอร์รี่พายไว้ที่ใต้ มัลติคอปเตอร์เพื่อทำการส่งลงมายังพื้นดินเพื่อตรวจ จับค่าสีที่ได้กำหนดไว้ เมื่อมัลติคอปเตอร์บินเข้าใกล้วัตถุ หรือมัลติคอปเตอร์ บินตามจุดที่กำหนดโดยใช้จีพีเอส และเมื่อกล้องได้ตรวจจับเจอวัตถุที่กำหนด จะทำการประมวลผล และส่งมายังเครื่องคอมพิวเตอร์ โดยมีการเขียนไพทอนเพื่อส่งค่ามาที่คอมพิวเตอร์ เมื่อส่งค่ามาแล้ว นักบินจะทำการสั่งมัลติคอปเตอร์ให้ลงจอดตามจุดที่กำหนดไว้ โดยก่อนขึ้นบินผู้เล่นจะต้องทำการเชื่อมต่อระบบ Wi – Fi ให้ลิงก์กับคอมพิวเตอร์ให้สมบูรณ์ก่อนที่จะขึ้นบิน โดยโค้ดไพทอนที่อยู่ในราสเบอร์รี่พาย จะเริ่มทำงานด้วยตัวเองและคอยค้นหาจุดที่กำหนดไว้อยู่ตลอดเวลาที่ทำการบิน หรือค้นหาจุดลงจอด จากการทดลองผลที่ได้นั้นมีค่าความคลาดเคลื่อน ในการลงจอดโดยเฉลี่ย 76 ซม. ส่งผลให้มีความคลาดเคลื่อนน้อยกว่าการลงจอดแบบ RTL โดยใช้ระบบจีพีเอส มีค่าเท่ากับ 270 ซม. ในการลงจอดความผิดพลาดที่เกิดขึ้นอันเนื่องมาจากกระแสลม ที่พัดมัลติคอปเตอร์ในช่วงเวลาที่สั่ง Landing ทำให้เกิดความผิดพลาด เนื่องจากมัลติคอปเตอร์นั้น ลอยอยู่บนอากาศทำให้มีตัวแปรต่าง ๆ มากมาย ไม่ว่าจะเป็นแรงลงการทำงานของจีพีเอสผิดพลาด และยังรวมไปถึงแสงซึ่งเป็นตัวแปรสำคัญ ในการตรวจจับค่าสี เพราะอาจมีการสะท้อนของแสงแดด ก็จะทำให้ค่าสีที่เห็นนั้นเกิดการเปลี่ยนแปลง ทำให้ไม่สามารถทำการตรวจจับวัตถุตามที่กำหนดได้จึงจำเป็นต้องมีการตรวจสอบค่าของสีอยู่ตลอดเวลาเพื่อให้สามารถตรวจจับได้ตลอดเวลา รวมไปถึงต้องมีการตรวจสอบแรงลงและสภาพบรรยากาศ ก่อนทำการบินอยู่เสมอเพื่อความปลอดภัยของมัลติคอปเตอร์และทำการผู้บังคับ

อย่างไรก็ตาม การทดลองนี้เป็นแค่จุดเริ่มต้นเท่านั้น เนื่องมัลติคอปเตอร์บินอยู่บนอากาศมีตัวแปรมากมาย และยังต้องมีการปรับปรุงการตรวจจับค่าสี ให้มีความแม่นยำมากขึ้น และในการทดสอบแสดงให้เห็นว่าผู้เล่นให้ความสนใจกับมัลติคอปเตอร์และมีโอกาสที่จะให้บุคคลทั่วไปมี

โอกาสได้ทดลองใช้การตรวจจับค่าสีหรือการค้นหาจุดลงจอด เพื่อความปลอดภัยในการลงจอดและสามารถบินได้ทุกที่ทุกเวลา

5.2 ปัญหาในการทำงาน

- 1) ต้องทำการศึกษาระบบการทำงานของมัลติคอปเตอร์ และทำการฝึกบินให้ชำนาญเพื่อให้สามารถบังคับให้มัลติคอปเตอร์สามารถไปใกล้จุดที่กำหนดได้
- 2) เนื่องจากมีปัญหาจากแรงลม จึงต้องทำการตรวจสอบสภาพอากาศอยู่เสมอ เพื่อความปลอดภัย
- 3) ต้องทำการศึกษาการเขียนโค้ด ไพทอนเนื่องจากภาษาไพทอนนี้ ผู้จัดทำไม่เคยใช้งานหรือมีความรู้เลยจึงต้องทำการศึกษาและเริ่มต้นใหม่
- 4) เนื่องจากมีปัญหาเกี่ยวกับแสงจึงทำให้การตรวจจับสีของวัตถุที่มีความผิดพลาด จึงต้องทำการตรวจค่าสีอยู่เสมอ
- 5) เนื่องจากมัลติคอปเตอร์สามารถบินได้ไม่เกิน 20 นาที จึงทำให้ไม่สามารถทำการทดสอบต่อเนื่องได้ จึงต้องทำการจัดเวลาให้เหมาะสม
- 6) เมื่อทำการทดสอบแล้วเกิดข้อผิดพลาดหลายจุด จึงจำเป็นต้องแก้ไขให้สามารถใช้งานได้อย่างสมบูรณ์

5.3 ข้อเสนอแนะ

- 1) อยากให้มีการเพิ่มเติม วิชาการเขียนไพทอน และมีการอบรมการใช้งานลินุกซ์ เป็นวิชาเลือกเพื่อคนที่สนใจ เพราะเป็นสิ่งที่สามารถนำมาใช้ประโยชน์ได้จริง สำหรับนักศึกษาที่สนใจจะเป็นนักพัฒนาต่อไป
- 2) น่าจะมีการจัดอบรมเกี่ยวกับการใช้งาน บอร์ดราสเบอร์รี่พาย สำหรับผู้ที่สนใจ
- 3) การค้นคว้าจำค่านี้ควรมีการพัฒนาต่อ และเพิ่มความสมบูรณ์ให้มากยิ่งขึ้นเพื่อสามารถนำไปใช้งานได้จริง
- 4) ควรมีการปรับแต่งมัลติคอปเตอร์ให้สมบูรณ์อยู่เสมอ เพื่อทดลองอย่างมีประสิทธิภาพ

บรรณานุกรม

- [1] ardupilot , mission planner , Available (15/6/2014) :
<http://planner.ardupilot.com/wiki/mission-planner-overview/>
- [2] ardupilot , **APM Autopilot Suit** , Available (25/8/2014): <http://ardupilot.com/>
- [3] store, **Jorones** , Available (18/11/2014) : <http://store.jdrones.com/>
- [4] raspberry pi , **Downloads** , Available (15/11/2014) : <http://www.raspberrypi.org/>
- [5] Facebook , **Raspberry Pi 66 Club** , Available(11/2/2015) :
<http://www.facebook.com/RaspberryPi66?fref=ts>

ภาคผนวก

TNI

โค้ดไพทอน การตรวจจับค่าสี

```
import cv

import math

color_tracker_window = "Color Tracker"

class ColorTracker:

    def __init__(self):

        cv.NamedWindow( color_tracker_window, 1 )

        self.capture = cv.CaptureFromCAM(0)

    def run(self):

        while True:

            img = cv.QueryFrame( self.capture )

            #blur the source image to reduce color noise

            cv.Smooth(img, img, cv.CV_BLUR, 3);

            #convert the image to hsv(Hue, Saturation, Value) so its

            #easier to determine the color to track(hue)
```

```
hsv_img = cv.CreateImage(cv.GetSize(img), 8, 3)

cv.CvtColor(img, hsv_img, cv.CV_BGR2HSV)

#limit all pixels that don't match our criteria, in this case we are
#looking for purple but if you want you can adjust the first value in
#both turples which is the hue range(120,140). OpenCV uses 0-180 as
#a hue range for the HSV color model
thresholded_img = cv.CreateImage(cv.GetSize(hsv_img), 8, 1)

cv.InRangeS(hsv_img, (160, 80, 220), (180, 90, 230), thresholded_img)

#determine the objects moments and check that the area is large
#enough to be our object

moments = cv.Moments(cv.GetMat(thresholded_img,1), 0)

area = cv.GetCentralMoment(moments, 0, 0)

#there can be noise in the video so ignore objects with small areas
if(area > 100000):

    #determine the x and y coordinates of the center of the object

    #we are tracking by dividing the 1, 0 and 0, 1 moments by the area

    x1 = cv.GetSpatialMoment(moments, 1, 0)/area

    y1 = cv.GetSpatialMoment(moments, 0, 1)/area

    x2 = 320

    y2 = 240
```

```
distance = math.sqrt((x2-x1)**2+(y2-y1)**2)

#print 'x: ' + str(x) + ' y: ' + str(y) + ' area: ' + str(area)

print 'distance : ' + str(distance)

print 'area: ' + str(area)

#create an overlay to mark the center of the tracked object
overlay = cv.CreateImage(cv.GetSize(img), 8, 3)
cv.Circle(img, (int(x1), int(y1)), 2, (255, 255, 255), 20)
cv.Add(img, overlay, img)

#add the thresholded image back to the img so we can see what was
#left after it was applied
cv.Merge(thresholded_img, None, None, None, img)

#display the image
cv.ShowImage( color_tracker_window, img)

if cv.WaitKey(10) == 27:
    break

if __name__=="__main__":
    color_tracker = ColorTracker()
    color_tracker.run()
```

โค้ดไพออน ด้านรับคำสั่งลงจอด

```
import socket

client_socket = socket.socket()

server_address = '192.168.0.100'

server_port = 8080

client_socket.connect((server_address, server_port))

data= client_socket.recv(1024)

print data

execfile("client.py")
```

TNI