

อภิเนันทนาการ



สำนักหอสมุด



อุปกรณ์เสริมสำหรับพัดลมติดผนังควบคุมด้วยเสียงพูด

AN ACCESSORY FOR A FAN-WALL CONTROLLED BY VOICE



| | | |
|------------|---------|---------------|
| นายกำพล | องการ | รหัส 56362560 |
| นายมานิตย์ | ประชาติ | รหัส 56363079 |
| นายวีรพล | สุพิน | รหัส 56363178 |

มี CD

| |
|----------------------------------|
| สำนักหอสมุด มหาวิทยาลัยนครสวรรค์ |
| วันลงทะเบียน 24 ส.ค. 2561 |
| เลขทะเบียน 19220820 ✓ |
| เลขเรียกหนังสือ ฝ5 |

ก 3940
2559

CD-STL 88

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์

ปีการศึกษา 2559



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ อุปกรณ์เสริมสำหรับพัฒนาคิดค้นงานควบคุมด้วยเสียงพูด
ผู้ดำเนินโครงการ นายกัมพล องการ รหัส 56362560
 นายมานิตย์ ประชะติ รหัส 56363079
 นายวีรพล สุพิน รหัส 56363178
ที่ปรึกษาโครงการ ดร. สราวุฒิ วัฒนวงศ์พิทักษ์
สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2559

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงการ
(ดร. สราวุฒิ วัฒนวงศ์พิทักษ์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. มุขิตา สงฆ์จันทร์)

.....กรรมการ
(ดร.จิรวดี ผลประเสริฐ)

| | | | |
|-------------------|--|---------|---------------|
| ชื่อหัวข้อโครงการ | อุปกรณ์เสริมสำหรับพัดลมติดผนังควบคุมด้วยเสียงพูด | | |
| ผู้ดำเนินโครงการ | นายกัมพล | องการ | รหัส 56362560 |
| | นายมานิตย์ | ประชะติ | รหัส 56363079 |
| | นายวีรพล | สุพิน | รหัส 56363178 |
| ที่ปรึกษาโครงการ | ดร. สราวุฒิ วัฒนวงศ์พิทักษ์ | | |
| สาขาวิชา | วิศวกรรมไฟฟ้า | | |
| ภาควิชา | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| ปีการศึกษา | 2559 | | |

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการพัฒนาอุปกรณ์เสริมสำหรับพัดลมติดผนังแบบใช้สายดิ่ง ซึ่งเป็นการสั่งการด้วยเสียงพูดผ่านมือถือ โดยมีราสเบอร์รี่พายเป็นตัวประมวลผลและควบคุมการทำงานของมอเตอร์เกียร์ 12 โวลต์ ที่เป็นตัวขับเคลื่อนกลไกควบคุมการทำงานของพัดลม ในส่วนของการสั่งการในมือถือจะใช้แอปพลิเคชันที่สร้างจากแอปอินเวอร์เตอร์ แปลงคำสั่งเสียงพูดที่ได้เป็นตัวอักษรแล้วส่งค่าไปยังเว็บเซิร์ฟเวอร์บนราสเบอร์รี่พาย เพื่อประมวลผลและสั่งการควบคุมพัดลม จากผลการทดลองพบว่าอุปกรณ์นี้สามารถใช้ควบคุมพัดลมติดผนังแบบใช้สายดิ่งได้ถูกต้องและแม่นยำ นอกจากนี้ยังสามารถนำไปพัฒนาให้ใช้กับพัดลมหลายตัวได้ในอนาคต

Project title An Accessory for a Fan-Wall Controlled by Voice
Name Mr. Kampon Ongkan ID. 56362560
Mr. Manid Prachati ID. 56363079
Mr. Weerapon Supin ID. 56363178
Project advisor Mr. Sarawut Wattanawongpitak, D.Eng.
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic year 2016

Abstract

This project presents the development of the accessory for an electric wall-fan. The device provides voice command via mobile phone using the Raspberry Pi as a processor and function controller of the 12V Gear Motor. To control a wall-fan's functions, the driving force is retrieved from this gear motor. For the mobile commanding part, the application created by an inverter application is used to convert a speech into alphabets and send the results to a web server on the Raspberry Pi in order to process, command and control the wall-fan. According to the results of the experiment, this device can be used to control the electric wall-fan correctly and accurately. In addition, it can also be developed to work with more fans in the future.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นการสร้างอุปกรณ์เสริมสำหรับพัฒนาดัดผนังควบคุมด้วยเสียงพูด ซึ่งจะไม่มีทางสำเร็จไปได้ถ้าไม่ได้รับความช่วยเหลือจากบุคคลดังต่อไปนี้

ขอขอบพระคุณ ดร.สรารวุฒิ วัฒนวงศ์พิทักษ์ อาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ผู้เป็นอาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ให้คำแนะนำและให้ความช่วยเหลือแก่คณะผู้จัดทำเป็นอย่างดียิ่งตลอดมา

ขอขอบพระคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆตลอดระยะเวลาของการศึกษาเล่าเรียน ซึ่งเป็นความรู้ที่สามารถนำไปใช้ในการทำโครงการนี้และยังสามารถนำไปใช้ในการประกอบอาชีพในอนาคต

และสิ่งสำคัญที่สุดขอขอบพระคุณบิดา มารดา ที่ได้เลี้ยงดูและอบรมสั่งสอนแก่คณะผู้จัดทำจนทำให้คณะผู้จัดทำทุกคนมีวันนี้ได้ ซึ่งเป็นพระคุณอันหาที่เปรียบไม่ได้

ท้ายนี้คณะผู้จัดทำใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามมา ณ ที่นี้ ที่มีส่วนร่วมในการให้ข้อมูล เป็นที่ปรึกษาในการทำปริญญานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์คณะผู้จัดทำจึงขอขอบพระคุณไว้ ณ ที่นี้ด้วย

นายกัมพล

องการ

นายมานิตย์

ประชะติ

นายวีรพล

สุพิน

สารบัญ

หน้า

| | |
|--------------------------------------|----|
| ใบรับรองปริญญาโท | ก |
| บทคัดย่อภาษาไทย | ข |
| บทคัดย่อภาษาอังกฤษ | ค |
| กิตติกรรมประกาศ | ง |
| สารบัญ | จ |
| สารบัญตาราง | ช |
| สารบัญรูป | ฉ |
| | |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญของ โครงการงาน | 2 |
| 1.2 วัตถุประสงค์ของ โครงการงาน | 2 |
| 1.3 ขอบเขตการจัดทำโครงการงาน | 2 |
| 1.4 ตารางการดำเนินงาน | 3 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ | 4 |
| 1.6 งบประมาณ | 4 |
| | |
| บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง | 5 |
| 2.1 แอปอินเวอเตอร์ (App Inventor) | 5 |
| 2.2 ราสเบอร์รี่พาย (Raspberry Pi) | 6 |
| 2.2.1 โครงสร้างของราสเบอร์รี่พาย | 7 |
| 2.2.2 ราสเบอร์รี่พายกับอาคูโน | 9 |
| 2.3 ภาษาไพทอน (Python) | 9 |
| 2.4 ระบบการควบคุมด้วยเสียงในไพทอน | 11 |
| 2.5 โปรแกรมอาปาเช่ (Apache) | 11 |
| 2.6 ภาษาพีเอชพี (PHP) | 11 |
| 2.7 ประเภทของพัคลม | 12 |
| 2.8 สวิตช์แบบไมโคร (Microswitch) | 15 |

สารบัญ (ต่อ)

| | หน้า |
|---|-----------|
| 2.9 มอเตอร์เกียร์..... | 16 |
| บทที่ 3 ขั้นตอนวิธีการดำเนินงาน | 17 |
| 3.1 การทำงานของอุปกรณ์เสริมสำหรับพัฒมติดผนังควบคุมด้วยเสียงพูด | 17 |
| 3.1.1 สั่งการด้วยเสียงพูดผ่าน โทรศัพท์มือถือ | 18 |
| 3.1.2 กดปุ่มสวิตซ์ที่หน้าอุปกรณ์ | 19 |
| 3.2 กลไกการทำงานภายในอุปกรณ์ควบคุม | 19 |
| 3.3 โครงสร้างของอุปกรณ์ควบคุม..... | 23 |
| 3.4 อุปกรณ์ที่ใช้ | 26 |
| 3.5 การเชื่อมต่อวงจรภายในอุปกรณ์ควบคุม | 28 |
| 3.6 คำสั่งในหน้าจอของแอปพลิเคชันที่ใช้สั่งการอุปกรณ์ควบคุม | 28 |
| บทที่ 4 การทดลองและผลการทดลอง | 32 |
| 4.1 ทดลองการทำงานของกลไกภายในอุปกรณ์ควบคุม | 32 |
| 4.2 ทดลองการทำงานของแอปพลิเคชัน | 34 |
| 4.2.1 ทดลองหาความผิดพลาดในการรับเสียงของแอปพลิเคชัน | 34 |
| 4.2.2 ทดลองสั่งการด้วยเสียงผ่านแอปพลิเคชันเพื่อควบคุมพัฒม | 36 |
| 4.3 ทดสอบการทำงานของอุปกรณ์เมื่อทำงานติดต่อกัน | 37 |
| 4.3.1 ทดสอบสั่งการอุปกรณ์ติดต่อกันหลายครั้ง | 37 |
| 4.3.2 ทดสอบสั่งการแต่ละเบอร์โดยเว้นระยะห่างในการสั่งการแต่ละครั้ง | 39 |
| บทที่ 5 สรุปผลและข้อเสนอแนะ | 41 |
| 5.1 สรุปผลการดำเนินงาน..... | 41 |
| 5.2 ปัญหาและแนวทางแก้ไข | 42 |
| 5.3 ข้อเสนอแนะในการพัฒนา..... | 43 |
| เอกสารอ้างอิง | 44 |

สารบัญ (ต่อ)

| | หน้า |
|--|------|
| ภาคผนวก ก ส่วนประกอบของแอปอินเวอร์เตอร์..... | 45 |
| ภาคผนวก ข ติดตั้งระบบปฏิบัติการลินุกซ์ ให้กับราสเบอร์รี่พาย..... | 54 |
| ภาคผนวก ค เว็บไซต์เฟเวอร์อปาเซ่..... | 63 |
| ประวัติผู้ดำเนินโครงการ | 75 |



สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| 1.1 ตารางการดำเนินงาน | 3 |
| 3.1 แสดงการทำงานของมอเตอร์ควบคุมการเปลี่ยนเบอร์ | 24 |
| 4.1 แสดงการทำงานของพัดลมในสถานะต่างๆ | 33 |
| 4.2 แสดงผลการทำงานของแอปพลิเคชัน | 35 |
| 4.3 แสดงคำสั่งใกล้เคียงที่ทำให้แอปพลิเคชันทำงาน..... | 36 |
| 4.4 แสดงผลการทำงานของพัดลมเมื่อได้รับคำสั่งจากแอปพลิเคชัน | 37 |
| 4.5 แสดงผลการสั่งการพัดลมโดยเว้นระยะเวลาต่อครั้ง..... | 38 |
| 4.6 แสดงผลการสั่งการพัดลม โดยเว้นระยะเวลาในการสั่งแต่ละเบอร์ | 40 |



สารบัญรูป

| รูปที่ | หน้า |
|--|------|
| 2.1 แอปอินเวอเตอร์ (App Inventor) | 6 |
| 2.2 ส่วนประกอบของบอร์ดราสเบอร์รี่พาย(โมเดลบี) | 7 |
| 2.3 ราสเบอร์รี่พาย โมเดลเอ และ โมเดลบี | 8 |
| 2.4 โมดูลกล้องในราสเบอร์รี่พาย | 8 |
| 2.5 พัฒมติดเพดาน | 13 |
| 2.6 พัฒม โจร | 13 |
| 2.7 พัฒมติดผนัง | 14 |
| 2.8 พัฒมตั้งโต๊ะ | 14 |
| 2.9 พัฒมตั้งพื้น | 15 |
| 2.10 พัฒมอุตสาหกรรม | 15 |
| 2.11 พัฒมดูดอากาศ | 16 |
| 2.12 ไมโครสวิทช์ | 16 |
| 2.13 ตัวอย่างมอเตอร์เกียร์ | 17 |
| 3.1 การติดตั้งอุปกรณ์กับพัฒมติดผนัง | 18 |
| 3.2 การสั่งการด้วยเสียงผ่าน โทรศัพท์มือถือ | 19 |
| 3.3 การสั่งการ โดยการกดสวิทช์ | 20 |
| 3.4 ตำแหน่งเริ่มต้นของแขนกล | 21 |
| 3.5 แขนกลกด ไมโครลิมิตสวิทช์ | 21 |
| 3.6 แผนผังการทำงานของอุปกรณ์ | 22 |
| 3.7 ตัวแสดงระดับความเร็วของพัฒมแบบหมุน | 23 |
| 3.8 แสดงด้านข้างของอุปกรณ์เสริมสำหรับพัฒมติดผนัง | 24 |
| 3.9 แสดงขนาดของฐานวางมอเตอร์ | 25 |
| 3.10 แสดงขนาดของฐานวางบอร์ดและตำแหน่งของบอร์ดต่างๆ | 25 |
| 3.11 ด้านหน้าของอุปกรณ์เสริมสำหรับพัฒมติดผนัง | 26 |
| 3.12 แสดงขนาดของแขนกลตั้งสาย | 26 |
| 3.13 บอร์ดราสเบอร์รี่พาย | 27 |
| 3.14 บอร์ดขับมอเตอร์ | 27 |
| 3.15 ไมโครลิมิตสวิทช์ | 28 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|--|------|
| 3.16 มอเตอร์เกียร์ | 28 |
| 3.17 การเชื่อมต่อระหว่างอุปกรณ์ต่างๆ | 29 |
| 3.18 หน้าจอแอปพลิเคชันที่ใช้สั่งการ | 29 |
| 3.19 หน้าจอแอปพลิเคชันหลังจากกดปุ่ม initial | 30 |
| 3.20 หน้าจอแอปพลิเคชันหลังจากกดปุ่ม Speech | 31 |
| 4.1 สวิตช์หน้าอุปกรณ์ควบคุม | 32 |
| 4.2 แอปพลิเคชันที่ใช้ทดสอบหาความผิดพลาดในการรับเสียง | 34 |



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ร้านอาหารในปัจจุบันส่วนมากไม่ได้ติดตั้งเครื่องปรับอากาศแต่จะติดตั้งพัดลมแทน ซึ่งพัดลมส่วนใหญ่ที่ใช้เป็นพัดลมชนิดติดผนังเนื่องจากไม่กีดขวางทางเดินทำให้มีพื้นที่ใช้สอยในร้านมากขึ้น และประหยัดไฟกว่าแบบติดเพดาน พัดลมชนิดติดผนังนั้นเมื่อเทียบกับแบบตั้งพื้นจะใช้งานได้ไม่สะดวกกว่าแบบตั้งพื้นที่สามารถย้ายตัวพัดลมมาใกล้ๆเพื่อใช้งานได้ แต่พัดลมชนิดติดผนังไม่สามารถเคลื่อนย้ายได้ การใช้งานนั้นต้องเดินไปหาตัวพัดลมซึ่งติดตั้งอยู่ในจุดๆหนึ่งซึ่งจุดๆนั้นอาจมีสิ่งกีดขวาง เช่น โต๊ะ เก้าอี้ หรือสิ่งอื่นซึ่งจะสร้างความลำบากในการใช้งานพัดลมเป็นอย่างมาก เพราะฉะนั้นหากพัดลมชนิดนี้สามารถสั่งการด้วยเสียงได้จะสะดวกกว่าเดิม

ในปัจจุบันระบบสั่งการระยะไกลมีการใช้งานกันอย่างแพร่หลายไม่ว่าจะเป็นการใช้เพื่อควบคุมเครื่องจักรในโรงงาน หรืออุปกรณ์ของใช้ต่างๆที่ใช้ในชีวิตประจำวัน ระบบสั่งการระยะไกลมีหลายแบบ เช่น ระบบสั่งการด้วยรีโมท ระบบสั่งการด้วยเสียง ระบบสั่งการด้วยมือถือ ระบบสั่งการด้วยคอมพิวเตอร์ เป็นต้น นั่นเป็นเพราะว่าระบบสั่งการระยะไกลนั้น ช่วยเพิ่มความสะดวกสบายและทำให้การใช้งานอุปกรณ์ต่างๆง่ายขึ้น

ระบบสั่งการด้วยเสียง เป็นระบบสั่งการชนิดหนึ่งที่มีใช้ในปัจจุบัน ระบบสั่งการชนิดนี้คือระบบที่ใช้เสียงต่างๆตามที่บันทึกไว้ ไม่ว่าจะเป็นเสียงพูด เสียงปรบมือ หรือเสียงอื่นๆ มาสั่งการให้อุปกรณ์ทำงาน ซึ่งระบบสั่งการนี้เป็นระบบที่ใช้งานง่ายและสะดวกสบายมาก เพียงแค่พูดหรือปรบมือก็สามารถสั่งการเปิดปิดหรือทำงานต่างๆตามอุปกรณ์นั้นสามารถใช้งานได้นั่นเอง ด้วยเหตุนี้จึงนำระบบสั่งการด้วยเสียงมาใช้กับพัดลมซึ่งเป็นเครื่องใช้ที่ใช้กันอย่างมากในปัจจุบัน แต่การใช้งานพัดลมปกตินั้นเราต้องเดินไปที่ตัวพัดลมเพื่อใช้งาน ทั้งการเปิด-ปิด เพิ่ม-ลดความเร็วหรือการเปลี่ยนทิศทางของพัดลม หากพัดลมสามารถสั่งการด้วยเสียงได้จะทำให้การใช้งานนั้นสะดวกสบายกว่าเดิมมาก โดยเฉพาะพัดลมชนิดติดผนังซึ่งไม่สามารถย้ายที่ได้ รวมไปถึงในกรณีที่อยู่ในร้านมีคนน้อยการเปิดปิดพัดลมที่ไม่จำเป็นจะช่วยประหยัดพลังงานและเวลาของพนักงานได้อีกด้วย

ดังนั้นในโครงการนี้ได้พัฒนาอุปกรณ์ที่ใช้ในการควบคุมการทำงานของพัดลมชนิดนี้ การทำงานของอุปกรณ์จะมีความคล้ายกับการใช้งานพัดลมปกติ แต่สามารถใช้เพียงแค่เสียงไปควบคุมแทนการเดินไปที่ตัวพัดลม กลไกภายในอุปกรณ์จะทำการดึงสายที่ควบคุมพัดลมเพื่อให้พัดลมทำงานตามที่ต้องการ

1.2 วัตถุประสงค์ของโครงการ

สร้างอุปกรณ์ควบคุมพัลสมชนิดติดผนังที่สามารถสั่งการด้วยเสียงพูดได้ เพื่อเพิ่มความสะดวกสบายในการใช้งานพัลสมชนิดนี้

1.3 ขอบเขตการจัดทำโครงการ

1. อุปกรณ์นี้สามารถใช้งานได้กับพัลสม 1 ตัว ต่อ 1 เครื่อง
2. สามารถใช้ได้กับพัลสมชนิดติดผนังแบบใช้สายดึงเท่านั้น
3. ใช้รหัสเบอร์รีพายรุ่นหนึ่งโมเดลปีพัสเป็นตัวประมวลผลคำสั่ง
4. ใช้แอปอินเวอร์เตอร์ในการสร้างแอปพลิเคชันที่รับคำสั่งเสียงพูด
5. ใช้กับมือถือที่เป็นระบบแอนดรอยด์เท่านั้น
6. สั่งการได้ในเครือข่ายวงแลนเดียวกันเท่านั้น
7. ใช้มอเตอร์เกียร์ขนาด 12 โวลต์ เป็นตัวขับเคลื่อนกลไก
8. ใช้ภาษาไพทอนในการเขียนโปรแกรมในรหัสเบอร์รีพาย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

การสร้างอุปกรณ์ควบคุมพัดลมชนิดติดตั้งผนังนี้ สามารถช่วยให้เกิดความสะดวกสบาย ทั้งการใช้งานในบ้าน อาคาร และในร้านค้าร้านอาหารต่างๆ การใช้งานพัดลมที่ควบคุมด้วยเสียงจะมีส่วนช่วยอย่างมากในร้านอาหารที่มีคนเยอะเพราะไม่จำเป็นต้องเดินไปที่ตัวพัดลม จึงทำให้ไม่จำเป็นต้องรบกวนลูกค้าที่มาในร้านและยังช่วยประหยัดเวลาของพนักงานในร้าน รวมไปถึงการใช้งานพัดลมด้วยเสียงนี้จะช่วยให้ผู้พิการทางร่างกายสามารถใช้งานได้ (ยกเว้นผู้พิการทางการพูด)

1.6 งบประมาณ

| | |
|---|------------------|
| 1. บอร์ดราสเบอร์รี่พายรุ่นหนึ่ง โมเดลบีพลัส | 1,300 บาท |
| 2. แผ่นอะคริลิก | 1,200 บาท |
| 3. พัดลม | 700 บาท |
| 4. สายไฟ | 100 บาท |
| 5. อุปกรณ์ติดตั้งและอุปกรณ์ต่างๆ | 600 บาท |
| 6. ค่าถ่ายเอกสารและเช่าเล่มปริญญาบัตร | 700 บาท |
| รวมเป็นเงินทั้งสิ้น (สี่พันหกร้อยบาทถ้วน) | <u>4,600 บาท</u> |
| หมายเหตุ: ถัวเฉลี่ยทุกรายการ | |

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

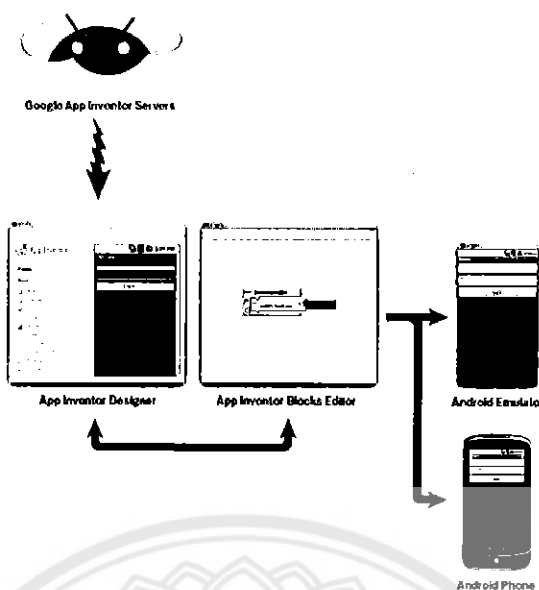
การจัดทำโครงการเรื่อง อุปกรณ์เสริมสำหรับพัฒนาคิดค้นฟังก์ชันควบคุมด้วยเสียงพูดนั้น ผู้ดำเนินโครงการได้ศึกษาค้นคว้าข้อมูลจากเว็บไซต์ที่เกี่ยวข้อง โดยแบ่งหัวข้อได้ดังนี้

1. แอปอินเวนเตอร์ (App Inventor)
2. ราสเบอร์รี่พาย (Raspberry Pi)
3. ภาษาไพทอน (Python)
4. ระบบการควบคุมด้วยเสียงในไพทอน
5. โปรแกรมอาปาเช่ (Apache)
6. ภาษาพีเอชพี (PHP)
7. ประเภทและหลักการการทำงานของพัลลัม
8. สวิตช์แบบไมโคร (Microswitch)
9. มอเตอร์เกียร์

2.1 แอปอินเวนเตอร์ (App Inventor)

แอปอินเวนเตอร์ (App Inventor) เป็นเครื่องมือที่ใช้สำหรับสร้างแอปพลิเคชันสำหรับสมาร์ทโฟนและแท็บเล็ตที่เป็นระบบปฏิบัติการแอนดรอยด์ ซึ่งบริษัทกูเกิลร่วมมือกับ MIT พัฒนาโปรแกรมแอปอินเวนเตอร์ขึ้น โดยลักษณะการเขียนโปรแกรมแบบวิชวลโปรแกรมมิ่ง คือ เขียนโปรแกรมด้วยการต่อบล็อกคำสั่ง เน้นการออกแบบเพื่อแก้ปัญหา ด้วยการสร้างโปรแกรมที่ผู้เรียนสนใจบนโทรศัพท์มือถือสมาร์ทโฟน

เซิร์ฟเวอร์แอปอินเวนเตอร์ รูปที่ 2.1 เป็นเครื่องมือให้บริการและเก็บงานต่างๆที่ผู้ใช้สร้างขึ้นมา ผู้ใช้พัฒนาโปรแกรมมือถือแอนดรอยด์ โดยสร้างงานและเขียนโปรแกรมบนเว็บเบราว์เซอร์ที่เชื่อมต่อไปยังเซิร์ฟเวอร์ของแอปอินเวนเตอร์ เมื่อได้โปรแกรมมาก็สามารถทดสอบกับโปรแกรมมือถือจำลอง หรือโทรศัพท์มือถือแอนดรอยด์ [7]



รูปที่ 2.1 เซิร์ฟเวอร์แอปอินเวอร์เตอร์

ที่มา : <http://kidsangsan.com>

ขั้นตอนการสร้างโปรแกรม เริ่มจากออกแบบหน้าตาโปรแกรมบนมือถือด้วยโปรแกรมแอปอินเวอร์เตอร์ ซึ่งใช้สำหรับสร้างส่วน โปรแกรมต่างๆที่ใช้งานใน โปรแกรมมือถือที่จะสร้างขึ้น จากนั้นเขียนโปรแกรมให้แต่ละส่วน โปรแกรม ด้วยโปรแกรมแอปอินเวอร์เตอร์บล็อกอิเตอร์ ซึ่งใช้วิธีการต่อบล็อกคำสั่ง เพื่อให้ส่วนโปรแกรมนั้นๆทำหน้าที่ตามที่ออกแบบเอาไว้

ระหว่างเขียนโปรแกรมอาจมีการแก้ไข เพิ่มเติม หรือลบบางส่วน โปรแกรมออกไป ทำให้ต้องแก้ไขโปรแกรม (debug) จนกว่าจะได้โปรแกรมตามที่ออกแบบไว้ เมื่อทุกส่วนโปรแกรมถูกสร้างเสร็จแล้ว สามารถทดสอบการใช้งาน โดยการติดตั้ง โปรแกรมลงไปในมือถือแอนดรอยด์ แล้วทดสอบการใช้งานผ่านมือถือจริงๆ แต่ถ้าไม่มีมือถือก็ยังสามารถทดสอบได้ ผ่านโปรแกรมมือถือจำลอง (Android emulator) ในคอมพิวเตอร์แทน [7]

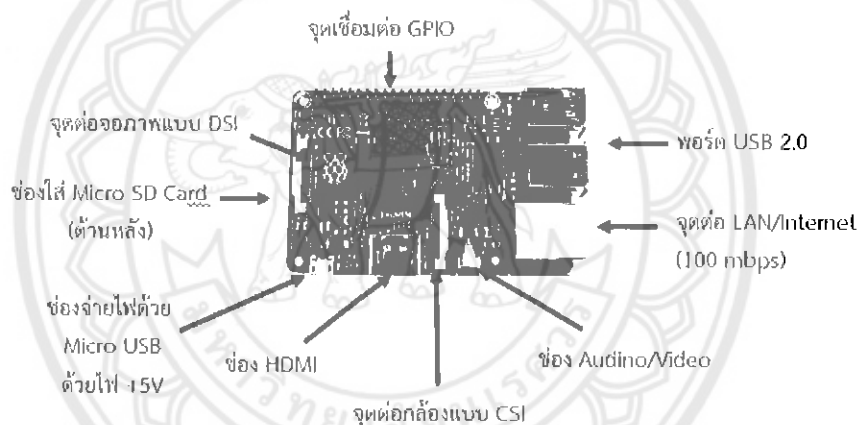
2.2 ราสเบอร์รี่พาย (Raspberry Pi)

ราสเบอร์รี่พาย (Raspberry Pi) คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับ จอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการจัดทำโครงการทางด้าน อิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการใช้งาน อินเทอร์เน็ต ส่งอีเมล หรือเล่นเกม อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วยบอร์ดราสเบอร์รี่พายรองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) , Fedora และ Arch Linux เป็นต้น

โดยติดตั้งระบบปฏิบัติการบนหน่วยความจำ บอร์ดราสเบอร์รี่พายนี้ถูกออกแบบมาให้มีตัวประมวลผลและแรมอยู่รวมภายในชิปเดียวกัน โดยมีจุดเชื่อมต่อ GPIO ให้ผู้ใช้งานสามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้ ด้วยคุณสมบัติทางเทคนิคของบอร์ดราสเบอร์รี่พายปัจจุบันมีด้วยกันสองโมเดล คือ โมเดล A และ โมเดล B ซึ่งทั้งสองโมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน [1]

2.2.1 โครงสร้างของราสเบอร์รี่พาย

ราสเบอร์รี่พายมีทั้งหมด 33 พอร์ต และมีจุดเชื่อมต่อกับอุปกรณ์ต่างๆ ดังรูปที่ 2.2 ในส่วนของราสเบอร์รี่พาย(โมเดลบี) นั้นมีหน่วยประมวลผล (CPU) แบบบอร์ดแคม BCM2836 900 MHz, Quad-core แรม (RAM) 1 GB รองรับระบบปฏิบัติการ (OS) ได้ทั้งแบบวินโดวส์ และลินุกซ์ขนาดของหน่วยความจำขึ้นอยู่กับหน่วยความจำที่ใช้ และมีขนาด 8.6×5.4×1.7 เซนติเมตร [1]



รูปที่ 2.2 ส่วนประกอบของบอร์ด ราสเบอร์รี่พาย(โมเดลบี)

ที่มา : <http://www.homeofmaker.com/?p=891>

พอร์ตต่างๆในราสเบอร์รี่พายจะมีดังต่อไปนี้

1. พอร์ต GPIO ซึ่งในโมเดลเอและบีทุกพินจะเหมือนกัน แต่โมเดลบีจะแตกต่างกันในรายละเอียดดังรูปที่ 2.3

| | | | |
|-----------|----|----|------------|
| 3.3V | 1 | | |
| 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 |

(ก)

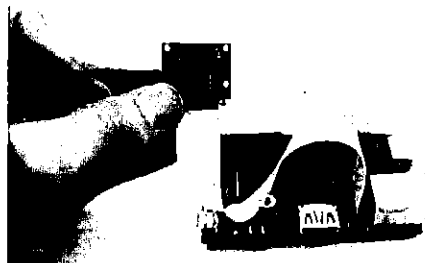
| | | | |
|-----------|----|----|------------|
| 3.3V | 1 | | |
| I2C1 SDA | 3 | | |
| I2C1 SCL | 5 | 6 | GROUND |
| GPIO4 | 7 | 8 | UART TXD |
| GROUND | | 10 | UART RXD |
| GPIO 17 | 11 | 12 | GPIO 18 |
| GPIO 27 | 13 | 14 | GROUND |
| GPIO 22 | 15 | 16 | GPIO 23 |
| 3.3V | 17 | 18 | GPIO 24 |
| SP10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

(ข)

รูปที่ 2.3 รัสเบอร์รี่พาย โมเดลเอ (ก) และ โมเดลบี (ข)

ที่มา : www.hobbytronics.co.uk/raspberry-pi-gpio-pinout

2. จุดเชื่อมต่อสัญญาณเสียง(Audio Output) ขนาด 3.5 มิลลิเมตร
3. ชิพควบคุมแลน (LAN Controller)
4. พอร์ตยูเอสบี 2.0 จำนวน 2 พอร์ต
5. พอร์ต RJ-45 Ethernet LAN 10/100 Mbps
6. พอร์ต CSI (Camera Serial Interface) สำหรับเชื่อมต่อ โมดูลกล้องรูปที่ 2.4
7. ชิพบอร์ดแคม BCM2387 chipset. 1.2 GHz Quad-Core ARM Cortex-A53
8. พอร์ต Micro USB Power สำหรับเป็นไฟเลี้ยงวงจรบอร์ดรัสเบอร์รี่พาย



รูปที่ 2.4 โมดูลกล้องในรัสเบอร์รี่พาย

ที่มา : www.thaieasyelec.com/article-wiki/embedded-electronics-application/บทความพัฒนา

โปรแกรมบน-raspberry-pi-ด้วย-qt.html

9. พอร์ต DSI (Display Serial Interface) ใช้สำหรับต่อจอแสดงผล

10. ช่องเสียบหน่วยความจำอยู่บริเวณด้านล่างของบอร์ด

2.2.2 ราสเบอร์รี่พายกับอาดูโน่

หากเปรียบเทียบกันในเรื่องของการใช้งานแล้วราสเบอร์รี่พาย จะคล้ายกับเครื่องคอมพิวเตอร์ที่มีวินโดวส์ หรือเครื่องแมค สามารถท่องเว็บไซต์ ดูหนังฟังเพลง หัดเขียนโปรแกรม และอื่นๆ ซึ่งจะต่างจากอันดูโน่ที่เป็นเพียงแค่ไมโครคอนโทรลเลอร์เท่านั้น ไม่มีระบบปฏิบัติการของตัวเอง เขียนโปรแกรมแล้วสั่งให้ทำงานเท่านั้น หากจะให้ราสเบอร์รี่พายมาทำงานแบบอาดูโน่ นั้น สามารถเขียนโปรแกรมเพื่อควบคุมพอร์ต GPIO ที่มากับราสเบอร์รี่พายได้ โดยใช้ภาษาไพทอน หรือภาษาซี ซึ่งมีความเร็วสูงกว่าอีกด้วย [1]

2.3 ภาษาไพทอน (Python)

ไพทอน (Python) คือ ชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมา โดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษาไพทอน ได้ทั้งบนระบบยูนิกซ์, ลินุกซ์, วินโดวส์ เอ็นที, วินโดวส์สองพัน, วินโดวส์เอกซ์พี หรือแม้แต่ระบบ FreeBSD ทำให้ทุกคนสามารถที่จะนำไพทอนมาพัฒนาโปรแกรมของได้ฟรี และความเป็นโอเพนซอร์ส (open source) ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ไพทอนมีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน

โค้ดของไพทอนถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ ก็จะประมวลผลไปที่ละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ ไพทอนเวอร์ชันแรกคือเวอร์ชัน 0.9.0 ออกมาเมื่อปี 2533 และเวอร์ชันปัจจุบันคือ 2.5.2 [4]

คุณลักษณะเด่นของภาษาไพทอน

2.7.1 ไพทอน รวมมาตรฐานการอินเทอร์เฟซ Tkinter ซึ่งสนับสนุนบนระบบเอ็กซ์วินโดวส์ ไมโครซอฟท์วินโดวส์ และแมคอินทอช การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ

2.7.2 เป็น Dynamic typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก

2.7.3 มี Built-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ไพทอนซึ่งประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง

2.7.4 มีเครื่องมือต่างๆมากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนค่า เป็นต้น

2.7.5 มีโมดูลสำหรับจัดการ Regular Expression

2.7.6 มีโมดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM, Image, CORBA, ORBs, XML เป็นต้น

2.6.7 จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ

2.7.8 อนุญาตให้ฝั่งชุดคำสั่งของไพทอน เอาไว้ภายในโค้ดภาษา C/C++ ได้

2.7.9 มีการอนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Library (DLL) เพื่อใช้ร่วมกับไพทอน

2.7.10 มีโมดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก Process thread regular, expression, xml, GUI และอื่นๆ

2.7.11 มีโมดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ตลอดจนใช้งานคำสั่ง FTP , Gopher, XML และอื่นๆอีกมาก

2.7.12 สามารถประมวลผลทางด้านวิทยาศาสตร์ และด้านวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ

2.7.13 มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, Sybase, Oracle , Informix, ODBC และอื่นๆ

2.7.14 มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลหรือภาพชัดหรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในรูปแบบต่างๆได้อย่างสะดวกและมีประสิทธิภาพ

2.7.15 มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์

2.7.16 มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer

2.7.17 มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

2.4 ระบบการควบคุมด้วยเสียงในโฟตอน

การรู้จำเสียงพูด (Speech Recognition) คือระบบโปรแกรมคอมพิวเตอร์ที่สามารถแปลงเสียงพูด (Audio File) เป็นข้อความตัวอักษร (Text) โดยสามารถแจกแจงคำพูดต่างๆ ที่มนุษย์สามารถพูดใส่ไมโครโฟน โทรศัพท์ หรืออุปกรณ์อื่นๆ และเข้าใจคำศัพท์ทุกคำอย่างถูกต้องเกือบ 100% โดยเป็นอิสระจากขนาดของกลุ่มคำศัพท์ ความดังของเสียงและลักษณะการออกเสียงของผู้พูด โดยระบบจะรับฟังเสียงพูดและตัดสินใจว่าเสียงที่ได้ยินนั้นเป็นคำๆใด รองรับ Google Speech Recognition API , Wit.ai , IBM Speech to Text และ AT&T Speech to Text [4]

- รองรับภาษาไทย (เฉพาะ Google Speech Recognition API)
- ใช้ BSD license

2.5 โปรแกรมอาปาเช่ (Apache)

อาปาเช่ (Apache) เป็น โปรแกรมสร้างเว็บเซิร์ฟเวอร์ สามารถติดตั้งอาปาเช่บนราสเบอร์รี่พาย และเมื่อติดตั้งเสร็จจะได้เว็บเซิร์ฟเวอร์เพื่อใช้งานหรือทดสอบสคริปต์ต่างๆ อาปาเช่สามารถแสดงผลไฟล์ HTML บน http และหากติดตั้งโมดูลเพิ่ม จะสามารถให้บริการเว็บไซต์ด้วยภาษาพีเอสพี (PHP)

การทำเว็บเซิร์ฟเวอร์บนราสเบอร์รี่พายนั้นเริ่มเป็นที่นิยม เพราะมีต้นทุนถูกและเหมาะสมกับงานที่ต้องการใช้งานส่วนตัว อีกทั้งยังมีตัวระบบปฏิบัติการที่มีพื้นฐานมาจากลินุกซ์จึงทำให้การใช้งานไม่ต่างจากเว็บเซิร์ฟเวอร์ [11]

2.6 ภาษาพีเอสพี (PHP)

พีเอสพี (PHP) นั้นเป็นภาษาสำหรับใช้ในการเขียนโปรแกรมบนเว็บไซต์ สามารถเขียนได้หลากหลาย โปรแกรมเช่นเดียวกับภาษาทั่วไป อาจมีข้อสงสัยว่าต่างจาก HTML อย่างไร คำตอบคือ HTML นั้นเป็นภาษาที่ใช้ในการจัดรูปแบบของเว็บไซต์ จัดตำแหน่งรูป จัดรูปแบบตัวอักษร หรือใส่สีสันให้กับเว็บไซต์ แต่พีเอสพีนั้นเป็นส่วนที่ใช้ในการคำนวณ ประมวลผล เก็บค่า และทำตามคำสั่งต่างๆ อย่างเช่น รับค่าจากรูปแบบที่เราทำ รับค่าจากช่องคำตอบของเว็บบอร์ดและเก็บไว้เพื่อนำมาแสดงผลต่อไป แม้แต่กระทั่งใช้ในการเขียน CMS ยอดนิยมเช่น Drupal , Joomla พุดง่ายๆคือเว็บไซต์จะโต้ตอบกับผู้ใช้ได้ ต้องมีภาษาพีเอสพีส่วน HTML หรือ Java ใช้เป็นเพียงแค่ตัวควบคุมการแสดงผลเท่านั้น

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-เซิร์ฟสคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดู และคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพีได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผลการอ่านข้อมูลจากฐานข้อมูล ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะ CGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสคริปต์พีเอชพี ทำงานผ่านพีเอชพีพาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ในยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

เมื่อใช้พีเอชพีในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ CCVS functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน [10]

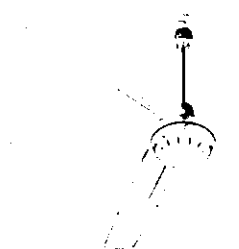
2.7 ประเภทของพัดลม

พัดลม (Fan) คือเครื่องจักรที่มีใบพัดหมุนซึ่งส่งผ่านอากาศหรือแก๊สอย่างต่อเนื่องที่ความดัน (Pressure) ค่าหนึ่ง โดยไม่เปลี่ยนแปลงค่าความหนาแน่นมากนัก หรือพัดลมหมายถึงปั๊มอัดอากาศความดันต่ำที่ใช้กำลังงานจากมอเตอร์ เพื่อเปลี่ยนเป็นปริมาตรการไหลของอากาศหรือแก๊สที่ความดันค่าหนึ่ง ดังนั้น พัดลมจึงทำหน้าที่หลักในการขับเคลื่อนอากาศปริมาณมากที่ความดันซึ่งเพียงพอที่จะเอาชนะความต้านทานของระบบที่พัดลมติดตั้งอยู่

พัดลมมีหลายชนิดตามขนาดอัตราไหลและความดันของของไหลที่ลำเลียง และตามวัตถุประสงค์การใช้งาน แบ่งเป็นประเภทใหญ่ๆ ได้เป็นแบบ centrifugal ซึ่งทำงานด้วยการให้แรงหนีศูนย์กลางให้เกิดกระแสในทิศทางตั้งฉากกับแกนแบบ axial flow ซึ่งสร้างกระแสของไหล (อากาศ) ในทิศทางเดียวกับเพลแบบ cross flow ซึ่งมีคุณสมบัติอยู่ระหว่างทั้งสองแบบข้างต้น และแบบอื่นๆ อย่างไรก็ตาม เพื่อสามารถติดตั้งและเชื่อมต่อกับท่อต่างๆ ได้สะดวก พัดลมแบบ centrifugal บางครั้งดูภายนอกแล้วจะมีลักษณะเหมือนกับแบบ axial flow โดยทั่วไปพัดลมแบบ axial flow จะเหมาะกับความดันต่ำ-อัตราไหลสูง ส่วนแบบ centrifugal จะเหมาะกับความดันสูง [2]

พัดลมในปัจจุบันนี้มีให้เลือกใช้ได้หลากหลายประเภท [3] ดังนี้

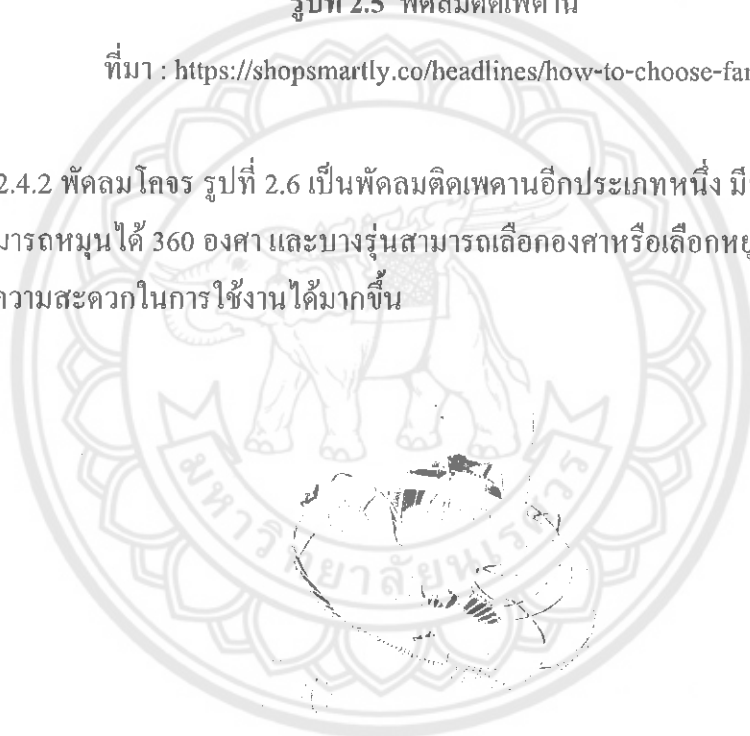
2.4.1 พัดลมติดเพดาน รูปที่ 2.5 หรือเรียกย่อๆว่า พัดลมเพดาน มีลักษณะที่เป็นใบพัดทรงคอปเตอร์ และที่เป็นลวดลายต่างๆ เช่น ลายใบไม้ พัดลมแบบนี้มีหลายขนาดให้เลือกใช้



รูปที่ 2.5 พัดลมติดเพดาน

ที่มา : <https://shopsmartly.co/headlines/how-to-choose-fan/>

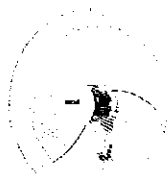
2.4.2 พัดลมโคมจร รูปที่ 2.6 เป็นพัดลมติดเพดานอีกประเภทหนึ่ง มีรูปทรงเหมือนพัดลมข้างฝา สามารถหมุนได้ 360 องศา และบางรุ่นสามารถเลือกองศาหรือเลือกหยุดหมุนให้อยู่กับที่ได้ ซึ่งอำนวยความสะดวกในการใช้งานได้มากขึ้น



รูปที่ 2.6 พัดลมโคมจร

ที่มา : <http://www.lazada.co.th/shop-ceiling-fans/>

2.4.3 พัดลมติดผนัง รูปที่ 2.7 หรือ พัดลมข้างฝา เป็นพัดลมที่เหมาะสมกับสถานที่ที่ไม่เหมาะกับการติดพัดลมเพดาน เช่น ห้องที่ตีฝ้าเพดานลงมาต่ำ



รูปที่ 2.7

รูปที่ 2.7

รูปที่ 2.7 พัดลมติดผนัง

ที่มา : <http://www.lazada.co.th/shop-wall-fans/>

2.4.4 พัดลมตั้งโต๊ะ รูปที่ 2.8 เป็นพัดลมขนาดเล็กที่สามารถวางบนพื้นหรือบนชั้นวาง



รูปที่ 2.8 พัดลมตั้งโต๊ะ

ที่มา : <http://www.wemall.com/products/mira-18-m-181-2386933795346.html>

2.4.5 พัดลมตั้งพื้น รูปที่ 2.9 พัดลมตั้งพื้นจะมีขนาดใหญ่กว่าพัดลมตั้งโต๊ะ ลำตัวเครื่องจะยาวกว่า และในบางรุ่นก็สามารถยืดหดลำตัวได้

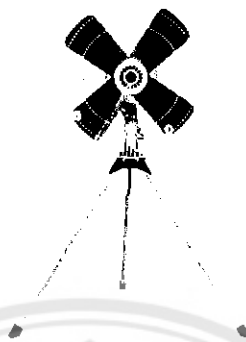


รูปที่ 2.9

รูปที่ 2.9 พัดลมตั้งพื้น

ที่มา : <http://www.boonthavorn.com/hardware/product-detail.php?id=1051185>

2.4.6 พัดลมอุตสาหกรรม รูปที่ 2.10 และพัดลมสามขา พัดลมรุ่นนี้นั้นมีใบพัดขนาดใหญ่ ลำเครื่องใหญ่ สามารถใช้ได้ทั้งในและนอกอาคาร รวมทั้งสามารถปรับเปลี่ยนการใช้งานได้ หลากหลาย โดยมีทั้งแบบตั้งพื้นและติดข้างฝา



รูปที่ 2.10 พัดลมอุตสาหกรรม

ที่มา : http://otto.co.th/products/product_detail.php?group=16&subgroup=21&id=119

2.4.7 พัดลมดูดอากาศ รูปที่ 2.11 ระบายอากาศ และฝังฝ้าเพดานห้องน้ำ มีไว้เพื่อระบาย อากาศภายในห้อง สามารถใช้ได้กับทุกห้อง ยกเว้นฝังฝ้าเพดานห้องน้ำที่เหมาะสมสำหรับการใช้ใน ห้องน้ำ



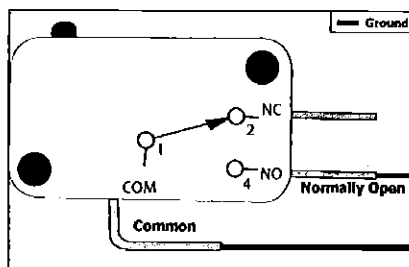
รูปที่ 2.11 พัดลมดูดอากาศ

ที่มา : <http://www.homeest.com>

2.8 สวิตช์แบบไมโคร (Microswitch)

สวิตช์แบบไมโคร (Microswitch) คือ สวิตช์แบบกดชนิดกดติดปลั๊กดัดมันเอง แต่เป็น สวิตช์ที่สามารถใช้แรงจํานวนน้อยๆกดปุ่มสวิตช์ได้ ก้านสวิตช์แบบไมโครสวิตช์มีด้วยกันหลาย แบบ อาจเป็นปุ่มกดเฉยๆหรืออาจมีก้านแบบโยกได้มากกดปุ่มสวิตช์อีกทีหนึ่ง การควบคุมตัดต่อ สวิตช์ ทำได้โดยกดปุ่มสวิตช์หรือกดก้านคันโยกเป็นการต่อ (ON) และเมื่อปล่อยมือออกจากปุ่ม

หรือก้านคันโยกเป็นการตัด จากรูปที่ 2.12 คือสถานะเมื่อทำการกดสวิทช์ขาคอมมอนจะต่อกับขา (NC) และเมื่อทำการปล่อยสวิทช์ขาคอมมอนจะกลับมาต่อกับขา (NO) เหมือนเดิม [8]

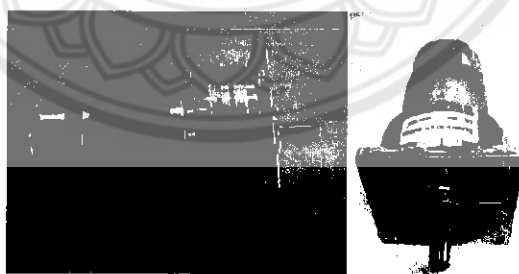


รูปที่ 2.12 ไมโครลิมิตสวิทช์

ที่มา : <http://group.wunjun.com/pimchanok/topic/196628-5151>

2.9 มอเตอร์เกียร์

มอเตอร์เกียร์ รูปที่ 2.13 เป็นเครื่องกลที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล ในรูปแบบของการหมุนเคลื่อนที่ โดยการเหนี่ยวนำแม่เหล็กไฟฟ้า ด้วยส่วนที่หมุนได้พันด้วยขดลวด มอเตอร์เกียร์มีโครงสร้างที่สำคัญ คือส่วนแม่เหล็กถาวร และส่วนของขดลวดตัวนำซึ่งมีโครงสร้างคล้ายกับเครื่องกำเนิดไฟฟ้า มอเตอร์เกียร์ทำงาน โดยการหมุนเคลื่อนที่ของขดลวดตัวนำ และทิศทางการเคลื่อนที่การทำงานของมอเตอร์ไฟฟ้า เรียกได้ว่าเป็นอุปกรณ์ที่นิยมใช้กันมากที่สุด และยังได้แพร่หลายไปยังโรงงานอุตสาหกรรมต่างๆ เป็นอุปกรณ์ที่ได้ใช้ในการควบคุมงานและเครื่องจักรต่างๆ ในโรงงานอุตสาหกรรม เครื่องจักรกลอุตสาหกรรม เครื่องลำเลียง เป็นต้น



รูปที่ 2.13 ตัวอย่างมอเตอร์เกียร์

ที่มา : <http://www.sangtawan.net/index.php?lay=show&ac=article&Id=539611293&Ntype=6>

การเลือกใช้มอเตอร์เกียร์นั้นต้องเลือกให้เหมาะสมกับการใช้งาน ซึ่งในบางครั้งจำเป็นต้องเลือกรูปแบบผสมที่มีลักษณะพิเศษที่มอเตอร์เกียร์สามารถสร้างอัตราทดได้หลากหลาย โดยเฉพาะอัตราทดอย่างสูงซึ่งมีรอบไม่ถึง 10 รอบต่อนาที หรือต่ำกว่า 1 รอบต่อนาที [9]

บทที่ 3

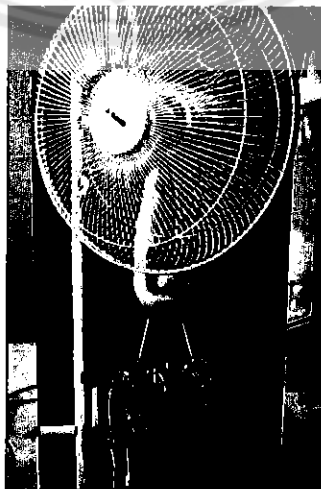
ขั้นตอนวิธีการดำเนินงาน

เนื้อหาในบทนี้จะกล่าวถึง การทำงานของอุปกรณ์เสริมสำหรับพัดลมติดผนังควบคุมด้วยเสียงพูด ขั้นตอนการออกแบบ และกลไกที่ใช้ในการดึงสายควบคุมของพัดลมติดผนัง โดยแบ่งออกเป็น 6 ส่วนดังนี้

1. การทำงานของอุปกรณ์เสริมสำหรับพัดลมติดผนังควบคุมด้วยเสียงพูด
2. กลไกการทำงานภายในอุปกรณ์ควบคุม
3. โครงสร้างของอุปกรณ์ควบคุม
4. อุปกรณ์ที่ใช้
5. การเชื่อมต่อวงจรภายในอุปกรณ์ควบคุม
6. คำสั่งในหน้าจอของแอปพลิเคชันที่ใช้สั่งการอุปกรณ์ควบคุม

3.1 การทำงานของอุปกรณ์เสริมสำหรับพัดลมติดผนังควบคุมด้วยเสียงพูด

การทำงานของอุปกรณ์เริ่มจากตั้งค่าพัดลมให้อยู่ในสถานะที่ยังไม่ทำงาน และทำการติดตั้งอุปกรณ์กับพัดลม โดยการนำสายควบคุมการทำงานของพัดลมมาติดกับแขนกลดึงสายของตัวอุปกรณ์ซึ่งรูปการติดตั้งโดยรวมจะเป็นดังรูปที่ 3.1



อุปกรณ์เสริม
สำหรับพัดลมติด
ผนังควบคุมด้วย
เสียงพูด

รูปที่ 3.1 การติดตั้งอุปกรณ์กับพัดลมติดผนัง

การใช้งานอุปกรณ์เสริมนี้สามารถใช้งานได้ 2 วิธี คือใช้แอปพลิเคชันสั่งการ และใช้ สวิตช์หน้าอุปกรณ์ควบคุมในการสั่งการ

3.1.1 สั่งการด้วยเสียงพูดผ่านโทรศัพท์มือถือ

สั่งการด้วยเสียงพูดผ่าน โทรศัพท์มือถือเพื่อไปควบคุมบอร์ดราสเบอร์รี่พายที่มีการใส่ โปรแกรมที่ใช้ควบคุมมอเตอร์เพื่อให้ตัวมอเตอร์ทำการขับเคลื่อนแขนกลที่ใช้คิงสายเพื่อควบคุม การทำงานของพัดลมดังรูปที่ 3.2



รูปที่ 3.2 การสั่งการด้วยเสียงผ่าน โทรศัพท์มือถือ

มีหลักการทำงานดังนี้

- 1) รับเสียงที่สั่งการจาก โทรศัพท์มือถือด้วยแอปพลิเคชันที่สร้างจากแอปอินเวอร์เตอร์
- 2) เสียงที่ได้รับจะถูกเปลี่ยนเป็นตัวอักษรด้วยระบบรับเสียงของกูเกิล และส่งไปยังเว็บ เซิร์ฟเวอร์ของราสเบอร์รี่พาย
- 3) ค่าตัวอักษรที่ส่งไปยังเว็บเซิร์ฟเวอร์จะถูกเปรียบเทียบกับค่าตัวอักษรที่เก็บไว้ในเว็บ เซิร์ฟเวอร์ที่เขียนขึ้นด้วยภาษาพีเอชพี
- 4) เมื่อค่าตัวอักษรที่รับมามีค่าเหมือนกับตัวอักษรที่ตั้งไว้ในพีเอชพี ระบบจะดึงไฟล์ไพ ทอนมาทำงาน เช่นค่าที่รับมาคือ Three ราสเบอร์รี่พายจะดึงไฟล์ที่ชื่อว่า three.py ในราสเบอร์รี่พาย มาทำงาน
- 5) ราสเบอร์รี่พายจะทำงานตามไฟล์ที่ดึงค่ามา โดยแต่ละไฟล์จะมีคำสั่งการทำงานที่ แตกต่างกันออกไป
- 6) ราสเบอร์รี่พาย จะส่งคำสั่ง ไปยังมอเตอร์ให้ทำงาน

7) มอเตอร์จะทำงานตามคำสั่งของรหัสเบอร์รี่พาย โดยจะนับการทำงานเป็นครั้งโดยมี
 ลิมิตสวิตช์เป็นตัวนับ เช่นรหัสเบอร์รี่พายดึงไฟล์ three.py มาทำงานมอเตอร์ก็จะทำงาน และจะหยุด
 ทำงานเมื่อลิมิตสวิตช์ถูกกด 3 ครั้ง

8) มอเตอร์ซึ่งต่อกับกลไกที่ใช้ในการดึงสายควบคุมพัลลวม เมื่อมอเตอร์ทำงานกลไกที่ใช้
 ดึงสายพัลลวมก็จะทำงานด้วย ทำให้สามารถใช้งานพัลลวมได้

3.1.2 กดปุ่มสวิตช์ที่หน้าอุปกรณ์

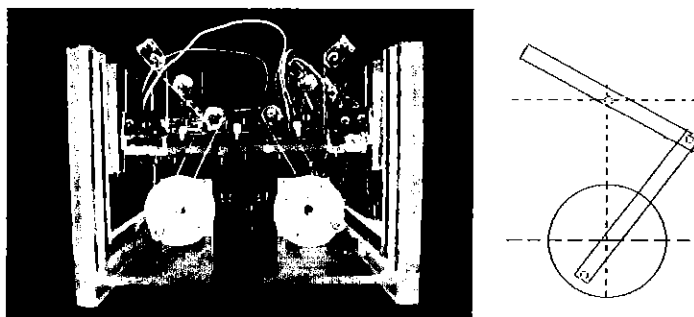
กดปุ่มสวิตช์ที่หน้าอุปกรณ์เพื่อทำการสั่งการบอร์ดราสเบอร์รี่พายโดยตรง ดังรูปที่ 3.3
 บอร์ดราสเบอร์รี่พายจะทำการควบคุมมอเตอร์เช่นเดียวกับการสั่งการด้วยเสียงพูด



รูปที่ 3.3 การสั่งการ โดยการกดสวิตช์

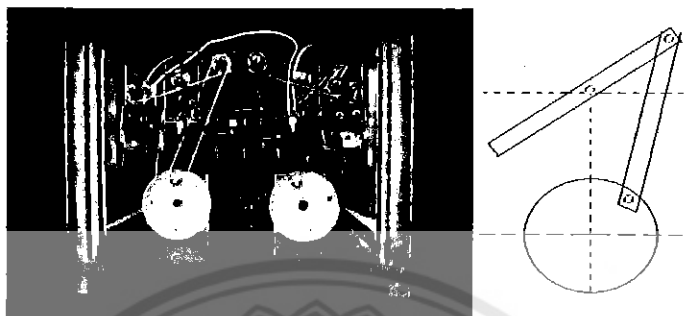
3.2 กลไกการทำงานภายในอุปกรณ์ควบคุม

การทำงานของกลไกในอุปกรณ์เสริมสำหรับพัลลวมติดผนังควบคุมด้วยเสียงพูด จะมีการ
 แบ่งการทำงานของมอเตอร์ 2 ตัว โดยมอเตอร์ตัวที่ 1 จะมีการทำงาน 3 ขั้นตอน เริ่มจากขั้นตอนที่ 1
 แขนกลอยู่ในสถานะเริ่มต้น เป็นสถานะหลังจากที่ทำการติดตั้งตัวอุปกรณ์แล้ว และทำการนำ
 สายควบคุมพัลลวมติดผนังมาใส่ในตัวแขนกลแล้ว ซึ่งตำแหน่งของแขนกล จะเป็นดังรูปที่ 3.4



รูปที่ 3.4 ตำแหน่งเริ่มต้นของแขนกล

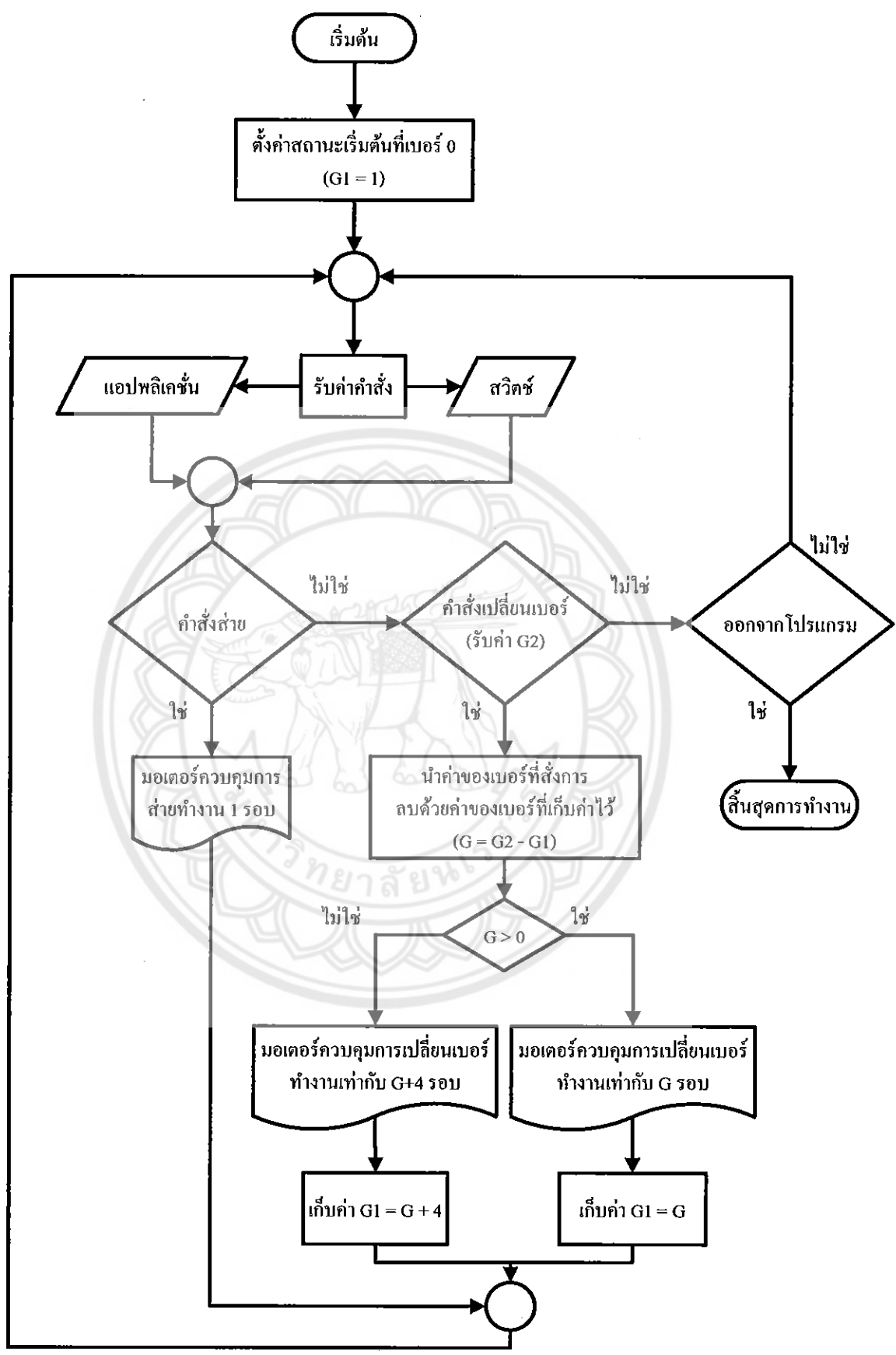
หลังจากมีการสั่งการด้วยเสียงจากโทรมือถือหรือการกดสวิตช์จากบอร์ดสวิตช์แล้ว บอร์ดราสเบอร์รี่พายจะทำการสั่งการมอเตอร์ให้หมุนตามที่ได้เขียนโปรแกรมควบคุมไว้ การหมุนของมอเตอร์นั้นจะทำให้แขนกลขยับตามไปด้วย เข้าสู่ขั้นตอนที่ 2 ซึ่งจะทำให้ตำแหน่งของแขนกลเปลี่ยนเป็นสถานะที่ 2 คือสถานะที่แขนกลไปกด ไมโครลิมิตสวิตช์ดังรูปที่ 3.5



รูปที่ 3.5 แขนกลกดไมโครลิมิตสวิตช์

ไมโครลิมิตสวิตช์ทั้งสองตัวจะทำหน้าที่ตรวจสอบการทำงานของมอเตอร์ทั้งสองตัวว่ามีการทำงานหรือไม่ เมื่อมอเตอร์ตัวที่ 1 ทำงานจะหมุนแขนกลไปกดไมโครลิมิตสวิตช์ซึ่งจะเป็นการตรวจสอบว่ามอเตอร์ทำงานกี่รอบ จำนวนรอบของมอเตอร์จะส่งผลกับระดับความเร็วของพัลลวม การหมุน 1 รอบ จะเป็นการทำงานของพัลลวมเบอร์ 1 การหมุนรอบที่ 2 จะเป็นการทำงานของพัลลวมเบอร์ 2 หมุนรอบที่ 3 จะเป็นการทำงานของพัลลวมเบอร์ 3 และการหมุนรอบที่ 4 จะเป็นการหยุดการทำงานของพัลลวมและแขนกลจะเข้าสู่สถานะเริ่มต้นนั่นคือขั้นตอนที่ 3

ส่วนมอเตอร์ตัวที่ 2 นั้น จะเป็นมอเตอร์ที่ใช้กำหนดการสายของพัลลวม โดยการหมุนแขนกลไปกดไมโครลิมิตสวิตช์เช่นเดียวกัน ซึ่งการกดไมโครลิมิตสวิตช์ครั้งที่ 1 จะเป็นการเริ่มการสายของพัลลวม การกดไมโครลิมิตสวิตช์ครั้งที่ 2 จะเป็นการหยุดการสายของพัลลวม ซึ่งการทำงานของมอเตอร์ทั้งสองตัวจะเป็นดังแผนผังการทำงานรูปที่ 3.6



รูปที่ 3.6 แผนผังการทำงานของอุปกรณ์

จากรูปแผนผังแสดงการทำงาน ที่สถานะเริ่มต้นพัลลวมจะปิด (เบอร์ 0) จึงกำหนดให้สถานะเริ่มต้น $G1$ เท่ากับ 0 ก่อนที่จะรับคำสั่งเข้ามาประมวลผล โดยคำสั่งจะมีอยู่ 3 คำสั่ง ได้แก่

คำสั่งสาย คำสั่งเปลี่ยนเบอร์ และคำสั่งออกจาก โปรแกรมโดยจะอธิบายการทำงานของแต่ละคำสั่งดังนี้

1. คำสั่งสาย เมื่ออุปกรณ์ได้รับคำสั่งสายมาจะส่งการไปยังมอเตอร์ที่ควบคุมการสายของพัลลวมให้ทำงาน 1 รอบแล้วกลับมาจับรับคำสั่งต่อไป

2. คำสั่งเปลี่ยนเบอร์ เมื่ออุปกรณ์ได้รับคำสั่งเปลี่ยนเบอร์จะมีการประมวลผลโดยมอเตอร์ควบคุมการเปลี่ยนเบอร์จะทำงานตามสมการ $G=G2-G1$ ตามแผนผังการทำงานโดยผลลัพธ์จะแบ่งเป็น 2 กรณีดังนี้

- กรณีที่ 1 $G2$ มากกว่า $G1$ เช่น เมื่อพัลลวมอยู่ที่เบอร์ 0 ($G1=0$) ถ้าพูดคำสั่งเปลี่ยนเบอร์ไปที่เบอร์ 3 (รับค่า $G2=3$) จากสมการ $G=G2-G1$ จะได้ว่า $G=3-0$ มอเตอร์ควบคุมการเปลี่ยนเบอร์ก็จะทำงานเท่ากับ G ครั้งแล้วเก็บค่า G ไว้เป็นสถานะปัจจุบัน ($G1=G$) ก่อนจะกลับมาจับรับคำสั่งใหม่

- กรณีที่ 2 $G2$ น้อยกว่า $G1$ จากกรณีที่ 1 สถานะปัจจุบันของพัลลวมจะอยู่ที่เบอร์ 3 ($G1=3$) ถ้าออกคำสั่งเปลี่ยนเบอร์ไปที่เบอร์ 1 (รับค่า $G2=1$) จากสมการ $G=G2-G1$ จะได้ว่า $G=1-3=-2$ ในกรณีนี้ค่า G จะติดลบซึ่งพัลลวมติดผนังแบบคิงสายไม่สามารถเปลี่ยนเบอร์ย้อนกลับได้เนื่องจากการทำงานของพัลลวมติดผนัง จำต้องทำงานแบบตามลำดับเบอร์คือ เบอร์ 1 เบอร์ 2 เบอร์ 3 และ ปิดตามลำดับ ดังรูปที่ 3.7 ซึ่งเป็นรูปตัวแสดงระดับความเร็วของพัลลวม โดยจะหมุนตามเข็มนาฬิกาเท่านั้น ไม่สามารถหมุนทวนเข็มนาฬิกาได้ ถึงแม้ว่าพัลลวมติดผนังแบบใช้สายคิงบางชนิดจะไม่มีตัวแสดงระดับความเร็วของพัลลวมแบบหมุน แต่การทำงานของพัลลวมชนิดนี้ มีการทำงานแบบเดียวกัน



รูปที่ 3.7 ตัวแสดงระดับความเร็วของพัลลวมแบบหมุน

จากการศึกษาการทำงานของพัลลวมติดผนังแบบใช้สายคิงแล้ว จึงได้นำค่า G ที่ติดลบมาบวก 4 เพื่อให้ได้ผลลัพธ์การทำงานที่ถูกต้องโดยมอเตอร์ควบคุมการเปลี่ยนเบอร์จะทำงานเท่ากับ $G+4$ ครั้ง เมื่อ $-2+4=2$ มอเตอร์จะทำงาน 2 รอบ แล้วจะกลับมาอยู่ที่เบอร์ 1 ตามต้องการ แล้วเก็บค่า

สถานะปัจจุบันไว้ ($G1=G+4$) แล้วกลับมารอรับคำสั่งต่อไป โดยการทำงานของมอเตอร์ควบคุมการเปลี่ยนเบอร์จะทำงานดังตารางที่ 3.1

ตารางที่ 3.1 แสดงการทำงานของมอเตอร์ควบคุมการเปลี่ยนเบอร์

| สถานะเบอร์เริ่มต้น (G1) | คำสั่งเปลี่ยนเบอร์ (รับค่า G2) | | | |
|-------------------------|--------------------------------|--------------------|--------------------|--------------------|
| | 0 | 1 | 2 | 3 |
| 0 | มอเตอร์ไม่ทำงาน | มอเตอร์ทำงาน 1 รอบ | มอเตอร์ทำงาน 2 รอบ | มอเตอร์ทำงาน 3 รอบ |
| 1 | มอเตอร์ทำงาน 3 รอบ | มอเตอร์ไม่ทำงาน | มอเตอร์ทำงาน 1 รอบ | มอเตอร์ทำงาน 2 รอบ |
| 2 | มอเตอร์ทำงาน 2 รอบ | มอเตอร์ทำงาน 3 รอบ | มอเตอร์ไม่ทำงาน | มอเตอร์ทำงาน 1 รอบ |
| 3 | มอเตอร์ทำงาน 1 รอบ | มอเตอร์ทำงาน 2 รอบ | มอเตอร์ทำงาน 3 รอบ | มอเตอร์ไม่ทำงาน |

3. ออกจากโปรแกรม คือคำสั่งสิ้นสุดการทำงาน

3.3 โครงสร้างของอุปกรณ์ควบคุม

โครงสร้างของอุปกรณ์เสริมสำหรับพัดลมติดผนัง จะเป็นโครงสร้างที่ใช้อะคริลิกหนา 0.5 เซนติเมตร ด้านข้างประกอบด้วยอะคริลิกสูง 12 เซนติเมตร กว้าง 16 เซนติเมตร ดังรูปที่ 3.8



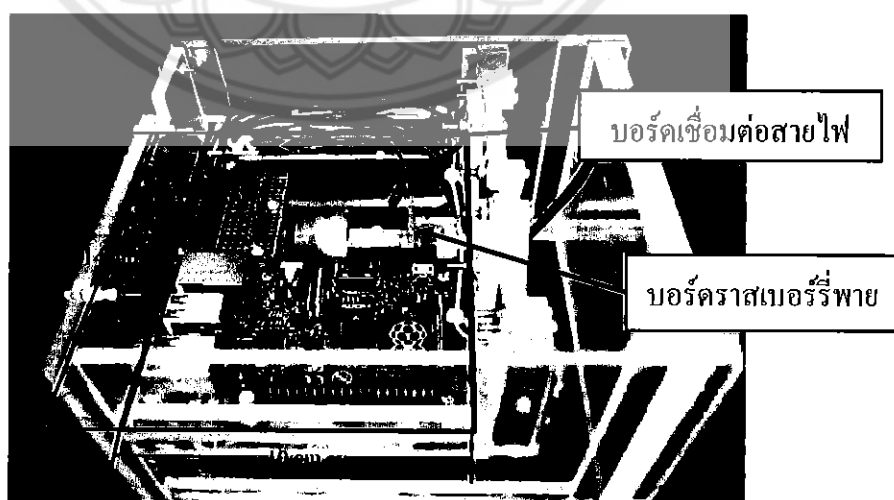
รูปที่ 3.8 แสดงด้านข้างของอุปกรณ์เสริมสำหรับพัดลมติดผนัง

ฐานวางมอเตอร์ เพื่อใช้วางมอเตอร์กระแสตรงขนาด 12 โวลต์ แกนเพลลา 0.6 เซนติเมตร ความเร็วในการหมุน 50 รอบต่อนาที จำนวน 2 ตัว ที่ใช้ในการหมุนแกนกลที่ใช้ในการดึงสายที่ควบคุมการทำงานของพัดลมติดผนัง ฐานวางมอเตอร์นี้เป็นแผ่นอะคริลิกหนา 0.5 เซนติเมตร ขนาด 15×16 เซนติเมตร วางสูงจากพื้น 1 เซนติเมตร ดังรูปที่ 3.9



รูปที่ 3.9 แสดงขนาดของฐานวางมอเตอร์

ฐานวางบอร์ด เป็นฐานที่ใช้วางบอร์ดควบคุมต่างๆที่ใช้ในอุปกรณ์เสริมสำหรับพัดลมติดผนังนี้ ประกอบไปด้วย 1) บอร์ดราสเบอร์รี่พายที่ใช้ในการประมวลผลคำสั่ง ใช้ควบคุมบอร์ดขับมอเตอร์และเป็นบอร์ดที่ใช้ในการเขียนโปรแกรมที่ใช้ควบคุมการทำงานทั้งหมด 2) บอร์ดเชื่อมต่อสายไฟที่ใช้ในการเชื่อมสายไฟจากอุปกรณ์และบอร์ดต่างๆเพื่อความเป็นระเบียบ ฐานวางบอร์ดนี้มีขนาด 14×10 เซนติเมตร ดังรูปที่ 3.10 วางสูงจากฐานวางมอเตอร์ 6 เซนติเมตร

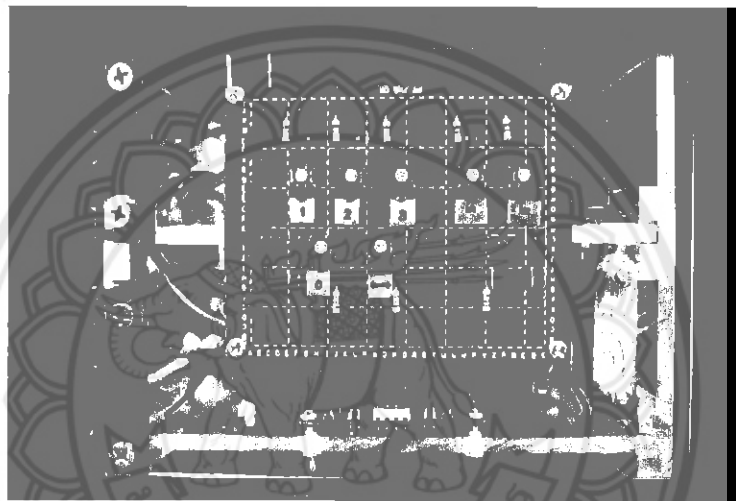


รูปที่ 3.10 แสดงขนาดของฐานวางบอร์ดและตำแหน่งของบอร์ดต่างๆ

17220820



ด้านหน้าของอุปกรณ์เสริมสำหรับพัดลมติดผนังจะเป็นแผ่นอะคริลิกหนา 0.5 เซนติเมตร สูง 12 เซนติเมตร กว้าง 16 เซนติเมตร สูง 12 เซนติเมตร ดังรูปที่ 3.11 ซึ่งจะประกอบไปด้วยบอร์ดสวิทช์ที่ใช้ในการควบคุมอุปกรณ์ด้วยมือซึ่งจะใช้ในกรณีที่การควบคุมด้วยเสียงมีปัญหา บอร์ดสวิทช์ประกอบไปด้วยปุ่ม 8 ปุ่ม ได้แก่ปุ่มเพิ่มระดับ 1,2,3 ปุ่ม 0 ที่ใช้ปิดพัดลม และมีปุ่ม M1,M2 ที่ใช้ในการควบคุมมอเตอร์ตัวที่ 1 และ 2 โดยการกดปุ่มค้างเพื่อให้ออเตอร์ทำงานและปล่อยเพื่อให้ออเตอร์หยุดทำงาน จะใช้ในการติดตั้งครั้งแรกหรือในกรณีที่อุปกรณ์มีปัญหา ปุ่มสีแดงคือปุ่มรีเซ็ตที่มีไว้ปรับให้โปรแกรมในบอร์ดราสเบอร์รี่พายกลับไปที่สถานะเริ่มต้น และมีบานพับที่ติดกับด้านข้างของอุปกรณ์ ทำให้ด้านหน้าของอุปกรณ์สามารถเปิดปิดได้



รูปที่ 3.11 ด้านหน้าของอุปกรณ์เสริมสำหรับพัดลมติดผนัง

ด้านหลังของอุปกรณ์เสริมสำหรับพัดลมติดผนังเป็นแผ่นอะคริลิกหนา 0.5 เซนติเมตร สองแผ่นติดกัน สามารถถอดเข้าออกได้ โดยรวมแล้วตัวอุปกรณ์จะมีปริมาตร 16×16×12 ลูกบาศก์เซนติเมตร น้ำหนักรวม 2 กิโลกรัม

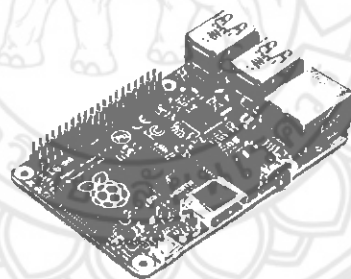
แกนกลตั้งสาย เป็นอุปกรณ์ที่ใช้เพื่อทำการตั้งสายควบคุมการทำงานของพัดลมติดผนัง ใช้อะคริลิกหนา 0.5 เซนติเมตร มี 2 ส่วน ได้แก่ส่วนล่างมีความยาว 7.5 เซนติเมตร กว้าง 1.0 เซนติเมตร ส่วนบนความยาว 6.0 เซนติเมตร กว้าง 1 เซนติเมตร ดังรูปที่ 3.12



รูปที่ 3.12 แสดงขนาดของแขนกลดึงสาย

3.4 อุปกรณ์ที่ใช้

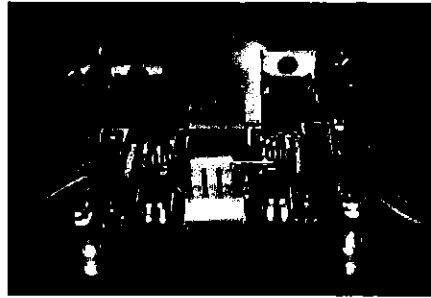
3.4.1 บอร์ดราสเบอร์รี่พายรุ่นหนึ่ง โมเดลบีพลัส รูปที่ 3.13 เป็นอุปกรณ์หลักที่ใช้ในการเขียนโปรแกรมควบคุม และเป็นอุปกรณ์ที่ใช้รับค่าการสั่งการต่างๆแล้วส่งค่าคำสั่งที่ได้รับเพื่อควบคุมการทำงานของกลไกภายในอุปกรณ์



รูปที่ 3.13 บอร์ดราสเบอร์รี่พาย

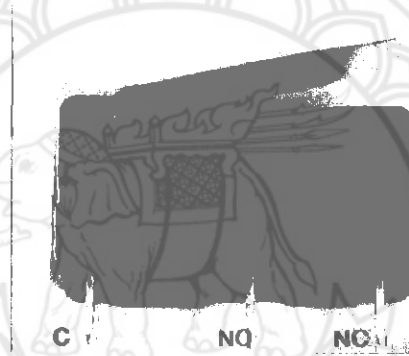
ที่มา : <https://www.adafruit.com/categories/105>

3.4.2 บอร์ดขับมอเตอร์ รูปที่ 3.14 เป็นบอร์ดที่รับเอาที่พูด จากบอร์ดราสเบอร์รี่พาย หลังจากนั้นจะทำการสั่งให้มอเตอร์ทำงาน สาเหตุที่ต้องใช้เพราะว่าบอร์ดราสเบอร์รี่พายไม่สามารถสั่งการมอเตอร์โดยตรงได้ เนื่องจากไม่มีวงจรกำลังในตัวราสเบอร์รี่พายจึงต้องเพิ่มบอร์ดขับมอเตอร์เข้าไป



รูปที่ 3.14 บอร์ดขับมอเตอร์

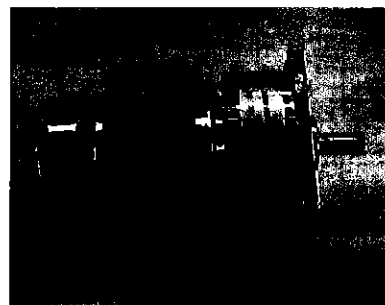
3.4.3 ไมโครลิมิตสวิตช์ รูปที่ 3.15 เป็นอุปกรณ์ที่ใช้ตรวจสอบการทำงานของมอเตอร์ เพื่อแจ้งบอร์ดทราบเบอร์รี่พายว่า มอเตอร์ควบคุมให้พัคลมอยู่สถานะใด ซึ่งตัวไมโครลิมิตสวิตช์ยาว 2 เซนติเมตร กว้าง 1 เซนติเมตร



รูปที่ 3.15 ไมโครลิมิตสวิตช์

ที่มา : <http://www.micontechlab.com/product/752/1x-micro-switch-limit-switch>

3.4.4 มอเตอร์เกียร์กระแสตรง 12 โวลต์ ความเร็ว 50 รอบต่อนาที รูปที่ 3.16 เป็นมอเตอร์ไฟฟ้ากระแสตรงที่นำมาต่อกับเกียร์บล็อก เป็นอุปกรณ์ที่ใช้เพื่อขับเคลื่อนแขนกลที่ใช้คิงสายควบคุมการทำงานของพัคลม มอเตอร์จะรับคำสั่งจากบอร์ดขับมอเตอร์

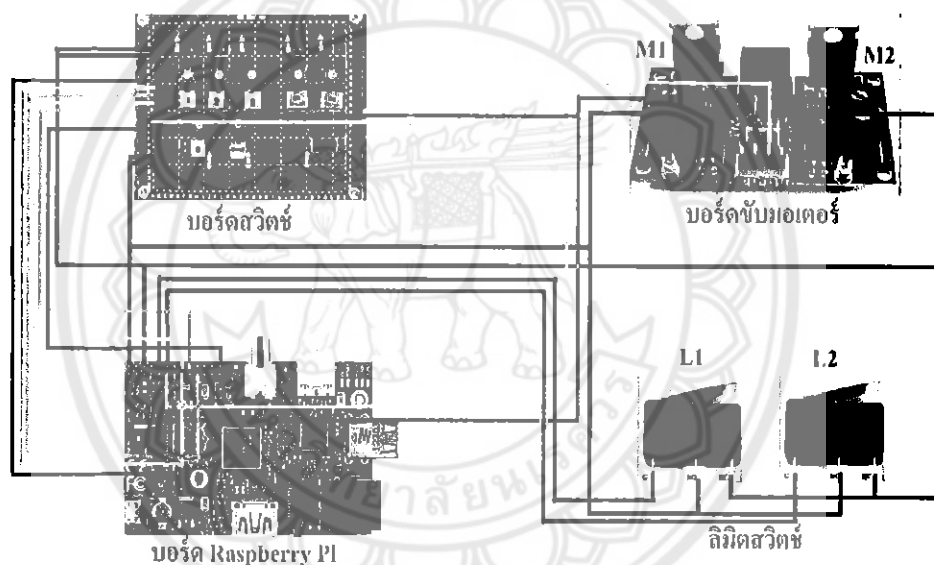


รูปที่ 3.16 มอเตอร์เกียร์

ที่มา : <http://www.sangtawan.net/index.php?lay=show&ac=article&Id=539611293&Ntype=6>

3.5 การเชื่อมต่อวงจรภายในอุปกรณ์ควบคุม

ภายในอุปกรณ์มีการเชื่อมต่อของวงจรต่างๆ ได้แก่ ภาสเบอร์รี่พาย บอร์ดสวิตช์ ลิมิตสวิตช์ และบอร์ดขับเคลื่อนมอเตอร์ซึ่งจะทำงานสัมพันธ์กัน โดยเริ่มจากบอร์ดสวิตช์ทำการส่งค่าที่ถูกกดสวิตช์ไปยังภาสเบอร์รี่พาย หลังจากนั้นภาสเบอร์รี่พายจะทำการประมวลผลคำสั่งตาม โปรแกรมที่ได้กำหนดไว้แล้วส่งค่าไปยังบอร์ดขับเคลื่อนมอเตอร์เพื่อสั่งให้มอเตอร์ทำงานหลังจากที่มอเตอร์ทำงาน แชนกกลที่ต่อกับมอเตอร์จะทำการเคลื่อนย้ายไปกดลิมิตสวิตช์ตามจำนวนครั้งที่กำหนด หลังจากนั้นลิมิตสวิตช์จะทำการส่งค่าตามจำนวนครั้งที่ถูกกดส่งกลับไปที่ภาสเบอร์รี่พายเพื่อตรวจสอบว่าเป็นไปตามที่โปรแกรมไว้หรือไม่ หลังจากนั้นภาสเบอร์รี่พายจะทำการสั่งให้มอเตอร์หยุดทำงานเมื่อครบตามที่โปรแกรมไว้แล้ว ซึ่งการเชื่อมต่อระหว่างอุปกรณ์ต่างๆเป็นดังรูปที่ 3.17



รูปที่ 3.17 การเชื่อมต่อระหว่างอุปกรณ์ต่างๆ

3.6 คำสั่งในหน้าจอของแอปพลิเคชันที่ใช้สั่งการอุปกรณ์ควบคุม

การสั่งการอุปกรณ์ควบคุมนั้นสามารถใช้โทรศัพท์มือถือระบบแอนดรอยด์เครื่องใดก็ได้ตามที่มีแอปพลิเคชัน หน้าจอของแอปพลิเคชันนั้นเป็นดังรูปที่ 3.18 โดยส่วนประกอบภายในหน้าจอแอปพลิเคชันมีดังนี้

© ลิขสิทธิ์ 2256 17 54

Wall Fan

แอปพลิเคชันควบคุมพัดลมผ่านเสียง

ข้อแนะนำ

- กด initial เมื่อพัดลมไม่ได้อยู่ที่เบอร์ 0
- กด speech เมื่อต้องการที่จะเปลี่ยนเบอร์พัดลม

Initial

Speech

สถานะ 0

Exit

รูปที่ 3.18 หน้าจอแอปพลิเคชันที่ใช้สั่งการ

3.6.1 ด้านบนสุดของหน้าจอ คือช่องแสดงชื่อแอปพลิเคชัน

3.6.2 ช่องข้อแนะนำ คือส่วนที่บอกให้ผู้ใช้ทราบว่าควรจะกดปุ่มใด ซึ่งจะเปลี่ยนไป

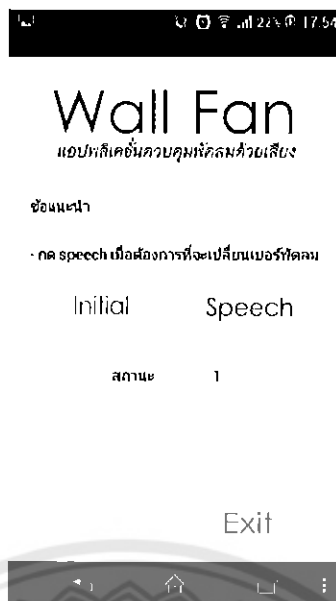
หลังจากกดปุ่ม initial และ Speech

3.6.3 ปุ่ม Initial คือปุ่มที่ใช้กำหนดค่าให้แอปพลิเคชันมีค่าการทำงานตรงกับพัดลม เนื่องจากในการเปิดแอปพลิเคชันขึ้นมาครั้งแรกแอปพลิเคชันจะไม่ทราบว่าพัดลมอยู่ในสถานะใด จึงจำเป็นที่จะต้องตั้งค่าให้แอปพลิเคชันมีค่าการทำงานตรงกับพัดลมเพื่อที่จะทำให้สามารถสั่งการพัดลมได้ เช่น เริ่มแรกพัดลมทำงานเบอร์ 1 จำเป็นต้องกดปุ่ม initial แล้วพูดว่า One เพื่อตั้งค่าให้แอปพลิเคชันทำงานตรงกับพัดลมแล้วหลังจากนั้น ช่องคำแนะนำจะเปลี่ยนเป็น “กด Speech เมื่อต้องการที่จะเปลี่ยนเบอร์พัดลม”

3.6.4 ปุ่ม Speech คือปุ่มที่ใช้สั่งการเปลี่ยนเบอร์พัดลม โดยมีคำสั่งที่ทำให้พัดลมทำงานได้แก่ One, Two, Three, Close และ Swing :ซึ่งก่อนที่จะกดปุ่ม Speech จะต้องสั่งเกต 2 อย่าง คือ

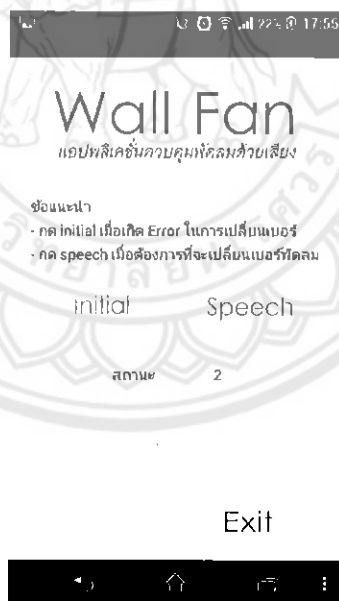
1) พัดลมอยู่ในสถานะเบอร์ 0 ในกรณีนี้ สามารถกดปุ่ม Speech ได้ทันที

2) พัดลมไม่ได้อยู่ในสถานะเบอร์ 0 ในกรณีนี้ต้องกดปุ่ม initial และทำการตั้งค่าสถานะของแอปพลิเคชันก่อน หลังจากนั้นช่องคำแนะนำจะเปลี่ยนเป็น “กด Speech เมื่อต้องการที่จะเปลี่ยนเบอร์พัดลม” ดังรูปที่ 3.19 จึงจะสามารถกดปุ่ม Speech เพื่อสั่งคำสั่งได้



รูปที่ 3.19 หน้าจอแอปพลิเคชันหลังจากกดปุ่ม initial

หลังจากกดปุ่ม Speech แล้ว แอปพลิเคชันจะแสดงหน้าจอดังรูปที่ 3.20 ซึ่งช่องคำแนะนำ จะมีคำแนะนำ 2 อย่าง คือ



รูปที่ 3.20 หน้าจอแอปพลิเคชันหลังจากกดปุ่ม Speech

- กด initial เมื่อเกิด Error ในการเปลี่ยนเบอร์ ในกรณีที่แอปพลิเคชันจะแสดงข้อผิดพลาด ขึ้นบนหน้าจอ จะต้องกด Initial แล้วตั้งค่าการทำงานของแอปพลิเคชันให้ตรงกับพัดลมใหม่ก่อน ไมเช่นนั้น หากกด Speech แล้วสั่งการพัดลมจะทำงานไม่ตรงกับคำสั่ง

- กด Speech เมื่อต้องการที่จะเปลี่ยนเบอร์พัฒนา ในกรณีนี้ สามารถกดปุ่ม Speech เพื่อใช้งานแอปพลิเคชันต่อไปได้

3.6.5 ช่องสถานะ คือช่องที่แสดงสถานะการทำงานของแอปพลิเคชัน ซึ่งช่องนี้ต้องแสดงค่าที่ตรงกับเบอร์ของพัฒนาที่ทำงานอยู่ โดยสามารถตั้งค่าได้จากการกดปุ่ม initial ไม่เช่นนั้นการใช้งานแอปพลิเคชันจะเกิดการผิดพลาด

3.6.6 ปุ่ม Exit คือปุ่มที่ใช้จบการทำงานของแอปพลิเคชัน เมื่อกดปุ่มนี้แอปพลิเคชันจะถูกตั้งค่าให้เป็นค่าเริ่มต้น (สถานะ 0) หากเปิดแอปพลิเคชันขึ้นมาใหม่ ต้องกดปุ่ม initial เพื่อตั้งค่าอีกครั้ง



บทที่ 4

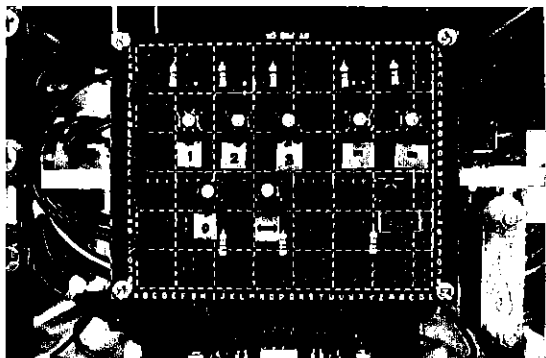
การทดลองและผลการทดลอง

การทดลองการทำงานของอุปกรณ์เสริมสำหรับพัฒนาดัดผนังควบคุมด้วยเสียงพูดนั้น จะเป็นการทดลองว่าอุปกรณ์นี้สามารถใช้ควบคุมพัฒนาดัดผนังให้สามารถทำงานได้ตามปกติหรือไม่ โดยมีการทดลองทั้งกลไกภายในอุปกรณ์ และแอปพลิเคชัน ดังนี้

1. ทดลองการทำงานของกลไกภายในอุปกรณ์ควบคุม
2. ทดลองการทำงานของแอปพลิเคชัน
 - 2.1 ทดลองหาความคิดพลาดในการรับเสียงของแอปพลิเคชัน
 - 2.2 ทดลองสั่งการด้วยเสียงผ่านแอปพลิเคชันเพื่อควบคุมพัฒนาดัดผนัง
3. ทดสอบการทำงานของอุปกรณ์เมื่อทำงานติดต่อกัน
 - 3.1 ทดสอบสั่งการอุปกรณ์ติดต่อกันหลายครั้ง
 - 3.2 ทดสอบสั่งการแต่ละเบอร์โดยเว้นระยะห่างในการสั่งการแต่ละครั้ง

4.1 ทดลองการทำงานของกลไกภายในอุปกรณ์ควบคุม

การทดลองนี้เป็นการทดลองว่ากลไกภายในอุปกรณ์ควบคุมสามารถใช้งานได้หรือไม่ และยังเป็น การทดลอง โปรแกรมที่เขียนลงในรหัสบอร์ดรีพายอีกด้วย ซึ่งจะทดลองโดยการกดสวิทช์ ที่หน้ากล่องควบคุมดังรูปที่ 4.1 แล้วสังเกตการณ์ทำงานของพัฒนาดัดผนัง หากสวิทช์สามารถสั่งการให้พัฒนาดัดผนังทำงานได้ แสดงว่ากลไกภายในตัวอุปกรณ์และโปรแกรมที่เขียนสามารถทำงานได้ตามต้องการ



รูปที่ 4.1 สวิทช์หน้าอุปกรณ์ควบคุม

ขั้นตอนการทดลอง

1) กดสวิตช์ที่หน้าอุปกรณ์ เบอร์ 1 ในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ

2) กดสวิตช์ที่หน้าอุปกรณ์ เบอร์ 2 ในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ

3) กดสวิตช์ที่หน้าอุปกรณ์ เบอร์ 3 ในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ

4) กดสวิตช์ที่หน้าอุปกรณ์ ปุ่ม Swing ในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ

หลังจากได้ทำการทดลองกดสวิตช์ได้ผลการทำงานของพัดลมดังตารางที่ 4.1

ตารางที่ 4.1 แสดงการทำงานของพัดลมในสถานะต่างๆ

| สถานะ เบอร์เริ่มต้น | เบอร์ที่กดสวิตช์ | | | | |
|------------------------|------------------|----------|----------|-----------|----------|
| | 1 | 2 | 3 | 0 | สาย |
| 0 | ทำงาน | ทำงาน | ทำงาน | ไม่ทำงาน | ไม่ทำงาน |
| 1 | ไม่ทำงาน | ทำงาน | ทำงาน | หยุดทำงาน | ทำงาน |
| 2 | ทำงาน | ไม่ทำงาน | ทำงาน | หยุดทำงาน | ทำงาน |
| 3 | ทำงาน | ทำงาน | ไม่ทำงาน | หยุดทำงาน | ทำงาน |

หมายเหตุ ทำงาน คือ พัดลมสามารถเปลี่ยนเบอร์ได้ตามคำสั่ง

ไม่ทำงาน คือ พัดลมไม่มีการเปลี่ยนเบอร์ เนื่องจากเบอร์ที่กดเป็นเบอร์ที่ทำงานอยู่แล้ว

หยุดทำงาน คือ พัดลมหยุดทำงานเนื่องจากเบอร์ที่สั่งเป็นเบอร์สั่งปิดพัดลม

จากผลการทดลองในตารางที่ 4.1 แสดงให้เห็นว่าสวิตช์สามารถสั่งการให้พัดลมทำงานได้ตามที่ต้องการ จึงสามารถสรุปได้ว่ากลไกภายในอุปกรณ์ควบคุม และ โปรแกรมที่เขียนในรหัสเบอร์รีพายสามารถทำงานได้ปกติ

4.2 ทดลองการทำงานของแอปพลิเคชัน

การทดลองนี้เป็นการทดลองว่าแอปพลิเคชันที่สร้างขึ้นสามารถใช้สั่งการให้พัดลมทำงานได้ตามปกติหรือไม่ โดยแบ่งเป็นทดลองหาความผิดพลาดในการรับเสียงของแอปพลิเคชัน และทดลองสั่งการด้วยเสียงผ่านแอปพลิเคชันเพื่อควบคุมพัดลม

4.2.1 ทดลองหาความผิดพลาดในการรับเสียงของแอปพลิเคชัน

การทดลองหาความผิดพลาดในการรับเสียงของแอปพลิเคชันนั้น เป็นการทดลองเพื่อให้ทราบว่าหลังจากที่สั่งการด้วยเสียงแล้วแอปพลิเคชันรับเสียงได้ถูกต้องหรือไม่ โดยแอปพลิเคชันที่ใช้ทดลองจะมีหน้าตาดังรูปที่ 4.2 ซึ่งจะมีการแสดงค่าเสียงที่แอปพลิเคชันรับได้ในช่องสถานะ จึงสามารถทำให้ทราบได้ว่าเสียงที่สั่งการไป แอปพลิเคชัน ได้รับค่าว่าอย่างไร



รูปที่ 4.2 แอปพลิเคชันที่ใช้ทดลองหาความผิดพลาดในการรับเสียง

ขั้นตอนการทดลอง

- 1) เปิดแอปพลิเคชัน กดปุ่ม Speech
- 2) พูดคำสั่ง One เป็นจำนวน 10 ครั้ง แล้วบันทึกเสียงที่ได้ในแต่ละครั้ง
- 3) พูดคำสั่ง Two เป็นจำนวน 10 ครั้ง แล้วบันทึกเสียงที่ได้ในแต่ละครั้ง
- 4) พูดคำสั่ง Three เป็นจำนวน 10 ครั้ง แล้วบันทึกเสียงที่ได้ในแต่ละครั้ง
- 5) พูดคำสั่ง Close เป็นจำนวน 10 ครั้ง แล้วบันทึกเสียงที่ได้ในแต่ละครั้ง

6) พูดคำสั่ง Swing เป็นจำนวน 10 ครั้ง แล้วบันทึกเสียงที่ได้ในแต่ละครั้ง

7) ปิดแอปพลิเคชันโดยกดปุ่ม Exit

ช่อง “เสียงที่ได้” ในแอปพลิเคชันนั้นเป็นช่องที่แสดงว่าแอปพลิเคชันได้รับเสียงค่าใด หลังจากนั้นได้ทำการทดลองสั่งการด้วยเสียงที่ละเบอร์เป็นจำนวนเบอร์ละ 10 ครั้ง ได้ผลการทดลองดังตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการทำงานของแอปพลิเคชัน

| จำนวนครั้งที่ทำการทดลอง | ผลที่แสดงบนช่องสถานะของแอปพลิเคชัน | | | | |
|-------------------------|------------------------------------|------|-------|-------|--------|
| | One | Two | Three | Close | Swing |
| 1 | One | Too | Tee | Close | Swing |
| 2 | One | Too | T | Close | Sewing |
| 3 | One | Two | Free | Close | Sewing |
| 4 | One | Tool | Three | Cross | Swing |
| 5 | Once | Too | Free | Close | Swing |
| 6 | One | Two | Tee | Close | Swing |
| 7 | wan | Two | Three | Cross | Sewing |
| 8 | Once | Two | Free | Close | Celing |
| 9 | One | Too | Tree | Cross | Swing |
| 10 | One | Tool | Tree | Close | Swing |

จากการทดลองผลปรากฏว่าเสียงที่แอปพลิเคชันรับได้นั้น มีบางส่วนเป็นเสียงที่ใกล้เคียงกับเสียงที่ต้องการ ซึ่งเสียงที่ใกล้เคียงเหล่านี้ไม่สามารถทำให้อุปกรณ์ทำงานได้ ความผิดพลาดนี้อาจเกิดจากความผิดพลาดในการส่งเสียงของผู้พูดและเกิดจากเสียงรบกวนภายนอก ดังนั้นจึงได้สร้างให้แอปพลิเคชันสามารถรับเสียงที่ใกล้เคียงได้ โดยเสียงใกล้เคียงที่ทำให้อุปกรณ์ทำงานได้ มีดังตารางที่ 4.3

ตารางที่ 4.3 แสดงคำสั่งใกล้เคียงที่ทำให้แอปพลิเคชันทำงาน

| คำสั่งจริงที่ ต้องการ | คำสั่งใกล้เคียงที่แอปพลิเคชันสามารถทำงานได้ |
|--------------------------|---|
| One | Wan/Once |
| Two | Too/tool/true |
| Three | Tree/Tee/T/Free |
| Close | Cross |
| Swing | Sewing,Celing |

4.2.2 ทดลองสั่งการด้วยเสียงผ่านแอปพลิเคชันเพื่อควบคุมพัดลม

ทดลองใช้แอปพลิเคชันในมือถือสั่งการอุปกรณ์ควบคุมพัดลม เป็นการทดลองว่าแอปพลิเคชันสามารถสั่งการพัดลมผ่านอุปกรณ์ควบคุมได้หรือไม่ โดยทำการสั่งในสถานะต่างๆ เช่น เมื่อพูดคำสั่ง One ในขณะที่พัดลมอยู่ในสถานะเบอร์ 0 พัดลมต้องสามารถเปลี่ยนจากสถานะเบอร์ 0 ไปเป็นเบอร์ 1 ได้

ขั้นตอนการทดลอง

- 1) เปิดแอปพลิเคชัน กดปุ่ม Speech แอปพลิเคชันอยู่ในสถานะ 0
- 2) พูดคำสั่ง One ในแอปพลิเคชันในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ
- 3) พูดคำสั่ง Two ในแอปพลิเคชันในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ
- 4) พูดคำสั่ง Three ในแอปพลิเคชันในขณะที่พัดลมอยู่ในสถานะ เบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ
- 5) พูดคำสั่ง Swing ในแอปพลิเคชันในขณะที่พัดลมอยู่ในสถานะเบอร์ 0, เบอร์ 1, เบอร์ 2, และเบอร์ 3 ตามลำดับ

6) ปิดแอปพลิเคชันโดยกดปุ่ม Exit

จากขั้นตอนการทำงานได้ผลการทดลองดังตารางที่ 4.4

ตารางที่ 4.4 แสดงผลการทำงานของพัคลมเมื่อได้รับคำสั่งจากแอปพลิเคชัน

| สถานะเบอร์ เริ่มต้น | คำสั่งจากแอปพลิเคชัน | | | | |
|------------------------|----------------------|----------|----------|-----------|----------|
| | One | Two | Three | Close | Swing |
| 0 | ทำงาน | ทำงาน | ทำงาน | ไม่ทำงาน | ไม่ทำงาน |
| 1 | ไม่ทำงาน | ทำงาน | ทำงาน | หยุดทำงาน | ทำงาน |
| 2 | ทำงาน | ไม่ทำงาน | ทำงาน | หยุดทำงาน | ทำงาน |
| 3 | ทำงาน | ทำงาน | ไม่ทำงาน | หยุดทำงาน | ทำงาน |

หมายเหตุ ทำงาน คือ พัคลมสามารถเปลี่ยนเบอร์ได้ตามคำสั่ง

ไม่ทำงาน คือ พัคลมไม่มีการเปลี่ยนเบอร์ เนื่องจากเบอร์ที่กด เป็นเบอร์ที่ทำงานอยู่แล้ว

หยุดทำงาน คือ พัคลมหยุดทำงานเนื่องจากเบอร์ที่สั่งเป็นเบอร์สั่งปิดพัคลม

จากการทดลองพบว่า พัคลมสามารถทำงานตามสถานะต่างๆตามที่ได้สั่งการด้วยเสียงบนแอปพลิเคชัน รวมไปถึงสามารถส่ายเมื่อได้รับคำสั่ง Swing และสามารถหยุดการทำงานเมื่อได้รับคำสั่ง Close อีกด้วย

4.3 ทดสอบการทำงานของอุปกรณ์เมื่อทำงานติดต่อกัน

การทดสอบนี้เป็นการทดสอบว่า เมื่อแอปพลิเคชันสั่งการพัคลมติดต่อกันจะเกิดความผิดพลาดใดหรือไม่ โดยการทดสอบมีดังนี้

4.3.1 ทดสอบสั่งการอุปกรณ์ติดต่อกันหลายครั้ง

ทำการสั่งการพัคลมผ่านแอปพลิเคชันเป็นจำนวน 10 ครั้ง โดยเริ่มจาก One, Two, Three, และ Close ตามลำดับ แล้วสั่งการหลังจากที่พัคลมหยุดทำงานในแต่ละครั้ง ซึ่งแต่ละครั้งเว้นเวลาห่างกัน 30 วินาที โดยไม่ปิดแอปพลิเคชันเพื่อทดสอบว่ามีปัญหาใดๆเมื่อสั่งติดต่อกันหรือไม่

ขั้นตอนการทดสอบ

- 1) เปิดแอปพลิเคชัน กดปุ่ม Speech สถานะแอปพลิเคชันอยู่ที่ 0
- 2) ทำการสั่งการคำสั่ง One, Two, Three, และ Close ผ่านแอปพลิเคชันติดต่อกันเป็นจำนวน 10 ครั้ง แล้วบันทึกผลการทดลองในแต่ละครั้ง
- 3) ปิดแอปพลิเคชัน โดยกดปุ่ม Exit

หลังจากทำการทดลองสั่งการ 10 ครั้ง ได้ผลการทดลองดังตารางที่ 4.5

ตารางที่ 4.5 แสดงผลการสั่งการพัฒนาระยะเวลาในการสั่ง

| จำนวนครั้งที่สั่ง | ผลการสั่งการ |
|-------------------|--------------------------------------|
| 1 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 2 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 3 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 4 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 5 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 6 | พัฒนาระยะเวลาการทำงานไม่ตรงกับคำสั่ง |
| 7 | พัฒนาระยะเวลาการทำงานไม่ตรงกับคำสั่ง |
| 8 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |
| 9 | พัฒนาระยะเวลาการทำงานไม่ตรงกับคำสั่ง |
| 10 | พัฒนาระยะเวลาในการทำงานได้ตามปกติ |

จากตารางจะเห็นได้ว่า บางครั้งพัฒนาระยะเวลาการทำงานไม่ตรงกับคำสั่ง ซึ่งสาเหตุเกิดจากการที่แอปพลิเคชันทำงานผิดพลาด โดยที่สามารถแก้ปัญหาได้สองวิธี วิธีแรก คือปิดแอปพลิเคชันแล้วเว้นระยะเวลาสั่งการ วิธีที่สอง คือเปลี่ยนรุ่นของรหัสเบอรัรี่พาย เนื่องจากรหัสเบอรัรี่พายที่ใช้มีความเร็วในการประมวลผลช้าหรือรหัสเบอรัรี่พายรุ่นนี้ไม่เหมาะกับการใช้งานในลักษณะนี้ ซึ่งจากการทดลองนำรหัสเบอรัรี่พายรุ่นที่ใหม่กว่ามาใช้ ผลปรากฏว่าไม่เกิดปัญหานี้ ดังนั้นสามารถแก้ปัญหาได้โดยการเปลี่ยนรหัสเบอรัรี่พายเป็นที่รุ่นใหม่กว่า

4.3.2 ทดสอบสั่งการแต่ละเบอร์โดยเว้นระยะห่างในการสั่งการแต่ละครั้ง

ทำการสั่งการพัฒนาคำนวณแอปพลิเคชัน โดยสั่งให้พัฒนาคำนวณที่แต่ละเบอร์ซึ่งแต่ละเบอร์มีระยะเวลาในการสั่งการห่างกัน 5 นาที ในการสั่งการ 1 ครั้งจะปิดแอปพลิเคชันเมื่อจบคำสั่งและทำการเปิดใหม่เมื่อจะสั่งคำสั่งต่อไป เพื่อทดสอบว่าพัฒนาคำนวณสามารถทำงานได้ตามคำสั่งหรือไม่ เนื่องจากในการใช้งานจริง การใช้งานพัฒนาคำนวณจะสั่งการพัฒนาคำนวณเป็นระยะไม่ได้สั่งต่อเนื่อง ดังนั้นการทดสอบนี้จึงเป็นการทดสอบว่าอุปกรณ์เสริมนี้สามารถนำไปใช้งานจริงได้หรือไม่

ขั้นตอนการทดสอบ

- 1) เปิดแอปพลิเคชันกดปุ่ม Speech พูดคำสั่ง One และปิดแอปพลิเคชัน แล้วเว้นระยะเวลาสั่งครั้งต่อไป 5 นาที
- 2) เปิดแอปพลิเคชันกดปุ่ม Speech พูดคำสั่ง Two และปิดแอปพลิเคชัน แล้วเว้นระยะเวลาสั่งครั้งต่อไป 5 นาที
- 3) เปิดแอปพลิเคชันกดปุ่ม Speech พูดคำสั่ง Three และปิดแอปพลิเคชัน แล้วเว้นระยะเวลาสั่งครั้งต่อไป 5 นาที
- 4) เปิดแอปพลิเคชันกดปุ่ม Speech พูดคำสั่ง Close และปิดแอปพลิเคชัน แล้วเว้นระยะเวลาสั่งครั้งต่อไป 5 นาที
- 5) ทำข้อ 1 ถึง 4 ซ้ำอีก 11 รอบ
- 6) ปิดแอปพลิเคชันโดยกดปุ่ม Exit

หลังจากทำการทดสอบสั่งการแต่ละเบอร์ได้ผลการทดสอบดังตารางที่ 4.6

ตารางที่ 4.6 แสดงผลการสั่งการพัฒนาคำนวณโดยเว้นระยะเวลาในการสั่งแต่ละเบอร์

| จำนวนครั้งที่สั่ง | คำสั่งที่ใช้ | ผลการสั่งการ |
|-------------------|--------------|---------------------------------|
| 1 | One | พัฒนาคำนวณสามารถทำงานได้ตามปกติ |
| 2 | Two | พัฒนาคำนวณสามารถทำงานได้ตามปกติ |
| 3 | Three | พัฒนาคำนวณสามารถทำงานได้ตามปกติ |
| 4 | Close | พัฒนาคำนวณสามารถทำงานได้ตามปกติ |

ตารางที่ 4.6 แสดงผลการสั่งการพัฒนา โดยเว้นระยะเวลาในการสั่งแต่ละเบอร์ (ต่อ)

| จำนวนครั้งที่สั่ง | คำสั่งที่ใช้ | ผลการสั่งการ |
|-------------------|--------------|----------------------------|
| 5 | One | พัฒนาสามารถทำงานได้ตามปกติ |
| 6 | Two | พัฒนาสามารถทำงานได้ตามปกติ |
| 7 | Three | พัฒนาสามารถทำงานได้ตามปกติ |
| 8 | Close | พัฒนาสามารถทำงานได้ตามปกติ |
| 9 | One | พัฒนาสามารถทำงานได้ตามปกติ |
| 10 | Two | พัฒนาสามารถทำงานได้ตามปกติ |
| 11 | Three | พัฒนาสามารถทำงานได้ตามปกติ |
| 12 | Close | พัฒนาสามารถทำงานได้ตามปกติ |

จากตารางจะเห็นได้ว่าในการสั่งการ 12 ครั้ง พัฒนาสามารถทำงานได้ตามปกติทุกครั้ง สาเหตุที่ไม่เกิดความผิดพลาดเหมือนกับการสั่งการติดต่อกัน เนื่องจากมีการเว้นช่วงเวลาในการสั่ง การทำให้รหัสเบอร์รีพายมีการพักการทำงาน รหัสเบอร์รีพายจึงพร้อมในการรับคำสั่งต่อไป ดังนั้น สามารถสรุปได้ว่าอุปกรณ์เสริมนี้สามารถนำไปใช้จริงได้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการนี้ได้ออกแบบและสร้างอุปกรณ์เสริมสำหรับพัฒนาชนิดผนังควบคุมด้วยเสียงพูด โดยการสร้างอุปกรณ์ควบคุมรูปทรงสี่เหลี่ยมที่ทำจากอะคริลิก ซึ่งประกอบไปด้วยมอเตอร์, ราวเบอร์รี่พายรุ่นหนึ่ง โมเดลบีพลัส, บอร์ดขับมอเตอร์, บอร์ดสวิทช์, แขนกลที่ทำจากอะคริลิกที่ต่อกับมอเตอร์เพื่อใช้เป็นตัวดึงสายควบคุมการทำงานของพัดลม รวมไปถึงได้สร้างแอปพลิเคชันจากแอปอินเวอร์เตอร์เพื่อใช้เป็นตัวรับเสียงสั่งการอีกด้วย

จากการทดลองการทำงานของกลไกภายในอุปกรณ์ควบคุม ได้ผลการทดลองว่าหลังจากที่ได้ทำการกดสวิทช์ในแต่ละเบอร์แล้ว โปรแกรมในราวเบอร์รี่พายสามารถสั่งการให้มอเตอร์ทำงานตามที่ต้องการได้และมอเตอร์สามารถขับเคลื่อนกลไกแขนควบคุมให้ทำการดึงสายควบคุมพัดลมได้ จึงส่งผลให้พัดลมสามารถทำงานได้ตามปกติ

จากการทดลองการทำงานของแอปพลิเคชัน ได้ผลการทดลองว่า การรับเสียงของแอปพลิเคชันนั้นมีการรับเสียงอื่นที่ใกล้เคียงกับเสียงที่ต้องการ จึงได้ทำการเพิ่มคำสั่งให้แอปพลิเคชันสามารถเทียบเสียงใกล้เคียงเหล่านั้นแล้วเปลี่ยนเป็นเสียงที่ต้องการเพื่อส่งคำสั่งที่ได้ไปให้ราวเบอร์รี่พาย และจากการทดลองสั่งการด้วยเสียงผ่านแอปพลิเคชันเพื่อควบคุมพัดลมปรากฏว่าพัดลมสามารถทำงานได้ตามปกติ

จากการทดสอบการทำงานของอุปกรณ์ติดต่อกันหลายครั้งผลปรากฏว่าหากมีการสั่งหลายรอบติดกันโดยไม่ปิดแอปพลิเคชันจะเกิดการผิดพลาดในบางครั้งเนื่องจากราวเบอร์รี่พายที่ใช้มีการประมวลผลช้าหรือราวเบอร์รี่พายรุ่นนี้ไม่เหมาะกับการทำงานในลักษณะนี้ แต่หากมีการเปิดปิดแอปพลิเคชันในการสั่งแต่ละครั้งและเว้นช่วงเวลาในการสั่งการ ความผิดพลาดนี้จะไม่เกิดขึ้น ดังนั้นอุปกรณ์เสริมสำหรับพัฒนาชนิดผนังควบคุมด้วยเสียงพูดนี้สามารถนำไปใช้งานจริงได้

5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่พบในระหว่างการทำโครงการนั้น มีทั้งในส่วนของตัวอุปกรณ์ และซอฟต์แวร์ ดังนี้

ตัวอุปกรณ์

การเกิดความผิดพลาดของการส่งค่าตัวอักษรไปยังเซิร์ฟเวอร์บนราสเบอร์รี่พายทำให้เกิดข้อผิดพลาดของการส่งการพัฒม ทำให้สถานะของพัฒมไม่ตรงกับการส่งการในบางครั้ง

แนวทางการแก้ไข

ใช้ราสเบอร์รี่พายรุ่นอื่นที่ใหม่กว่า เนื่องจากราสเบอร์รี่พายรุ่นหนึ่ง โมเดลบีพลัสที่ใช้ มีการประมวลผลที่ช้าซึ่งอาจเกิดจากรุ่นนี้ไม่เหมาะกับการใช้งานในเรื่องเสียง หากทำการเปลี่ยนเป็นรุ่นที่ใหม่กว่า เช่นราสเบอร์รี่พายสาม โมเดลบีจะสามารถแก้ปัญหานี้ได้

ซอฟต์แวร์

หากย้ายการติดตั้งอุปกรณ์ที่มีการเปลี่ยนวงแลนด้วย จะต้องมีการเปลี่ยนไอพีอินเตอร์เน็ตในโค้ดโปรแกรมของเว็บเซิร์ฟเวอร์ใหม่ทุกครั้ง ซึ่งปัญหานี้เกิดจากเว็บเซิร์ฟเวอร์ที่ใช้ไม่สามารถเปลี่ยนไอพีวงแลนได้เองโดยอัตโนมัติ

แนวทางการแก้ไข

สร้างเว็บเซิร์ฟเวอร์ด้วยโปรแกรมอื่นที่มีความหลากหลายและมีความสามารถมากกว่านี้ ซึ่งอาจเป็นโปรแกรมที่มีค่าใช้จ่ายในการใช้งาน

5.3 ข้อเสนอแนะในการพัฒนา

5.3.1 ออกแบบโครงสร้างของตัวอุปกรณ์ให้มีขนาดและน้ำหนักที่น้อยกว่านี้ เพื่อความกะทัดรัด ไม่ใช้พื้นที่ในการติดตั้งมาก รวมไปถึงสามารถประหยัดค่าใช้จ่ายในการทำโครงสร้างได้อีกด้วย

5.3.2 พัฒนาให้ตัวอุปกรณ์สามารถใช้กับพัดลมชนิดอื่นๆได้เพื่อให้สามารถใช้งานได้ครอบคลุม

5.3.3 สร้างตัวอุปกรณ์และแอปพลิเคชันให้สามารถเชื่อมต่อกันด้วยการเชื่อมต่อแบบอื่นๆ เพื่อความสะดวกในการใช้งานในกรณีที่ไม่มีการเชื่อมต่อชนิดนั้นๆ

5.3.4 เพิ่มให้อุปกรณ์เสริมสำหรับพัดลมติดผนังนี้สามารถทำงานกับพัดลมหลายตัวต่ออุปกรณ์หนึ่งเครื่อง เนื่องจากราสเบอร์รี่พายยังมีเอาต์พุตและอินพุตมากกว่า 4 พอร์ต ซึ่งจะสามารถทำให้ใช้งานกับพัดลมได้อีกหลายตัว

5.3.5 ศึกษาการทำงานของเซิร์ฟเวอร์ราสเบอร์รี่พายให้ละเอียด เนื่องจากการทำงานของเซิร์ฟเวอร์ที่ได้ยังมีข้อจำกัดเรื่องระยะเวลาการควบคุมพัดลมติดผนัง ซึ่งต้องเป็น ไอพีวงแลนเดียวกัน ดังนั้นถ้าหากศึกษาและพัฒนาเรื่องของเว็บเซิร์ฟเวอร์ราสเบอร์รี่พาย จะสามารถทำให้การควบคุมขยายเป็นวงกว้างมากขึ้น อาจสามารถทำให้ใช้งานได้ทุกที่มีสัญญาณอินเทอร์เน็ต

5.3.6 การทำงานของราสเบอร์รี่พายรุ่นหนึ่ง โมเดลบีพลัส ยังมีการทำงานได้ที่ล่าช้าเมื่อเทียบกับการทำงานของบอร์ดรุ่นล่าสุด ซึ่งหากสามารถเปลี่ยนเป็นราสเบอร์รี่พายเป็นรุ่นใหม่ จะทำให้การทำงานของระบบสามารถพัฒนาได้อีกมากมาย

เอกสารอ้างอิง

- [1] “การทำงานของ Raspberry Pi” สืบค้นเมื่อเดือนตุลาคม วันที่ 2 พ.ศ. 2559 จาก <http://www.sathittham.com> หัวข้อ [RPI]
- [2] “หลักการการทำงานของพัดลม” สืบค้นเมื่อเดือนตุลาคม วันที่ 2 พ.ศ. 2559 จาก <http://ienergyguru.com/2015/09/fan>
- [3] “ประเภทของพัดลม” สืบค้นเมื่อเดือนตุลาคม วันที่ 3 พ.ศ. 2559 จาก <https://luckymisu.wordpress.com>
- [4] “ระบบการควบคุมด้วยเสียงใน Python” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 6 พ.ศ. 2559 จาก <https://python3.wannaphong.com/2015/03/>
- [5] “ตัวอย่างการใช้งาน Python” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 6 พ.ศ. 2559 จาก <https://sites.google.com/site/dotpython/bth-thi-5/5-5>
- [6] “หลักการ Python” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 6 พ.ศ. 2559 จาก www.mindphp.com
- [7] “การสร้างแอป AppInventor” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 13 พ.ศ. 2559 จาก <http://www.slideshare.net/jokesparrow/3-mit-app-inventor-2>
- [8] “ลิมิตสวิตช์ สวิตช์แบบไมโคร” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 13 พ.ศ. 2559 จาก <http://group.wunjun.com/pimchanok/topic/196628-5151>
- [9] “หลักมอเตอร์เกียร์และมอเตอร์ไฟฟ้า” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 13 พ.ศ. 2559 จาก <http://www.euromachthailand.com>
- [10] “โครงสร้างภาษาPHP” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 13 พ.ศ. 2559 จาก <http://www.mwit.ac.th/~jeab/40201/ch3.php>
- [11] “Apache คืออะไร” สืบค้นเมื่อเดือนพฤศจิกายน วันที่ 13 พ.ศ. 2559 จาก <http://www.mindphp.com>

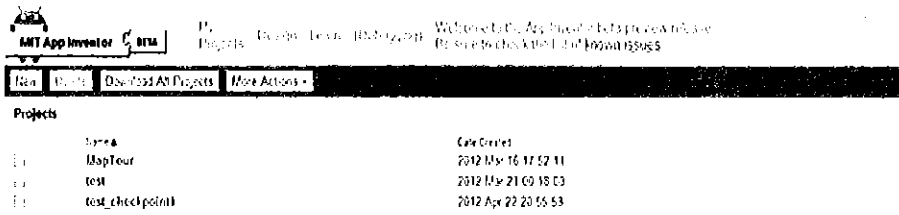


โปรแกรมแอปอินเวอร์เตอร์ ช่วยให้สามารถพัฒนาแอปพลิเคชันสำหรับ โทรศัพท์ระบบปฏิบัติการแอนดรอยด์ ซึ่งทำผ่านการ ใช้เว็บเบราว์เซอร์และทดสอบบน โทรศัพท์ที่เชื่อมต่ออยู่กับคอมพิวเตอร์หรือทดสอบบน โทรศัพท์จำลองบนเครื่องคอมพิวเตอร์ งานที่สร้างทั้งหมดจะถูกจัดเก็บไว้บนเซิร์ฟเวอร์แอปอินเวอร์เตอร์ ซึ่งจะช่วยให้สามารถพัฒนางานต่อที่เครื่องคอมพิวเตอร์เครื่องใดก็ได้ เพียงแค่ได้มีการเชื่อมต่อกับระบบอินเทอร์เน็ตไว้เท่านั้น

การสร้างแอปพลิเคชันจะแบ่งการทำงานออกเป็นสองส่วน คือ ส่วนออกแบบ (App Inventor Designer) ที่จะทำให้เราเลือกคอมโพเนนท์ที่ต้องการสำหรับที่จะให้สร้างแอปพลิเคชัน ส่วนที่สองเป็นส่วนการเขียนโค้ด (App Inventor Blocks Editor) ที่ให้เราเขียนโค้ดด้วยการต่อบล็อกต่างๆเข้าด้วยกันเป็นคำสั่ง ซึ่งจะเป็นการกำหนดพฤติกรรมหรือเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนท์ การเขียนโปรแกรมจะเสมือนการต่อชิ้นส่วนตัวต่อจิ๊กซอว์เข้าด้วยกัน ในแต่ละขั้นตอนการสร้างจะสามารถทำการทดสอบได้ทุกขณะ และเมื่อสร้างเสร็จสมบูรณ์แล้วจะสามารถแพ็คเกจแอปพลิเคชันเพื่อนำไปใช้งานบน โทรศัพท์ระบบปฏิบัติการแอนดรอยด์ เครื่องใดก็ได้ หรือหากไม่มีโทรศัพท์ระบบปฏิบัติการ แอนดรอยด์ก็สามารถที่จะทดสอบได้บน โทรศัพท์จำลองที่ทำงานอยู่บนคอมพิวเตอร์ซึ่งจะมีลักษณะการทำงานเหมือน โทรศัพท์จริงทุกประการ สภาพแวดล้อมในการพัฒนาด้วยโปรแกรมแอปอินเวอร์เตอร์นั้น สนับสนุนระบบปฏิบัติการที่หลากหลาย ไม่ว่าจะเป็นระบบปฏิบัติการ แมค OS X, GNU / ลินุกซ์และระบบปฏิบัติการวินโดวส์ และแอปพลิเคชันที่สร้างขึ้นนั้นสามารถติดตั้งและทำงานได้บน โทรศัพท์ระบบปฏิบัติการแอนดรอยด์หลากหลายรุ่นที่เป็นที่นิยมในปัจจุบัน

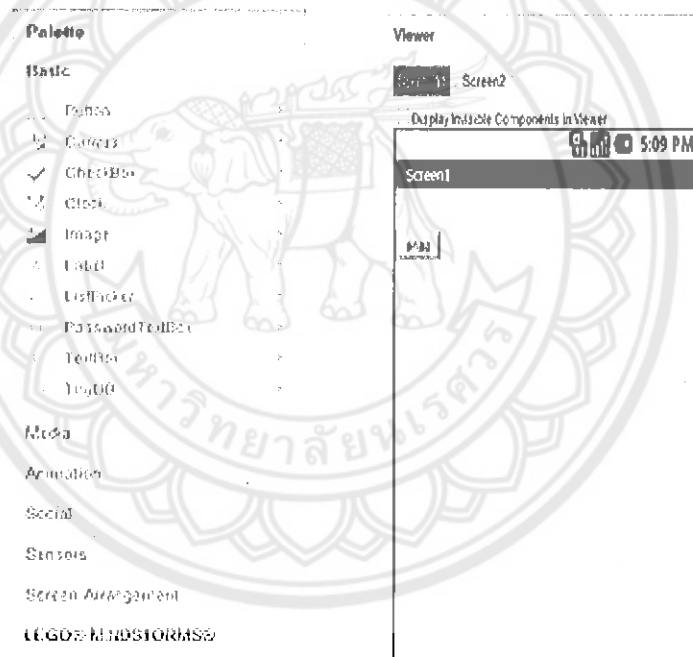
1.1.1 ส่วนออกแบบ (App Inventor Designer)

ในขั้นตอนแรกของการสร้างแอปพลิเคชันด้วยแอปอินเวอร์เตอร์เริ่มจากการเลือกคอมโพเนนท์ที่ต้องการและจัดวางลงในส่วนของการออกแบบ โดยจะผ่านส่วนของการออกแบบ (App Inventor Designer) แสดงให้เห็นถึงแอปพลิเคชันที่สร้างขึ้นในหน้าต่างเว็บเบราว์เซอร์ โดยด้านซ้ายจะเป็นส่วนของคอมโพเนนท์ที่แอปอินเวอร์เตอร์เตรียมไว้ให้จัดเรียงเป็นหมวดหมู่ เช่น ปุ่ม (button) ข้อความ (label) กล่องข้อความ (text box) เป็นต้น ผู้ใช้ทำการเพิ่มคอมโพเนนท์ที่เลือกด้วยการคลิกลากลงไปวางไว้ในโปรเจกต์



ภาพแสดงให้เห็นถึงแอปพลิเคชันที่สร้างขึ้น

อินเตอร์เฟซบนหน้าเว็บแอปอินเวอร์เตอร์ นั้นจะประกอบด้วยแท็บที่จะปรากฏในส่วนบนของหน้าเว็บซึ่งจะใช้ในการเข้าไปจัดการโปรเจก (My Projects) ส่วนการออกแบบ (Design) ส่วนการเรียนรู้คำสั่ง (Learn) ในหน้าจอการจัดการโปรเจก จะสามารถเข้าไปจัดการสร้าง ลบ คำนวณโหลด หรือเลือกโปรเจกที่สร้างและได้ทำการบันทึกไว้เพื่อกลับมาแก้ไขในหน้าจอส่วนการออกแบบได้



ภาพหน้าจอการจัดการโปรเจก

ในส่วนหน้าจอการออกแบบ ปุ่มที่อยู่ทางด้านบนจะใช้เพื่อการบันทึกโปรเจกในลักษณะต่างๆ การเพิ่มและลบหน้าจอ Screen ปุ่มสำหรับการเปิดส่วนการเขียนโค้ด (Open the Blocks Editor) และการจัดแพ็คเกจแอปพลิเคชันเพื่อนำไปใช้งานบนโทรศัพท์ระบบปฏิบัติการ Android ต่อไป ในการสร้างแอปพลิเคชันที่หน้าจอส่วนการออกแบบนี้ ผู้ใช้จะเลือกคอมโพเนนต์ที่อยู่ทางด้านซ้ายของหน้าจอ คลิกลากเพื่อนำมาวางลงในส่วน Viewer ที่อยู่ตรงกลางหน้าจอ ดังภาพที่ 4 หลังจากนั้นคอมโพเนนต์ที่เลือกนำมาวางจะปรากฏในส่วน Viewer ตามมุมมองของผู้ใช้ซึ่ง

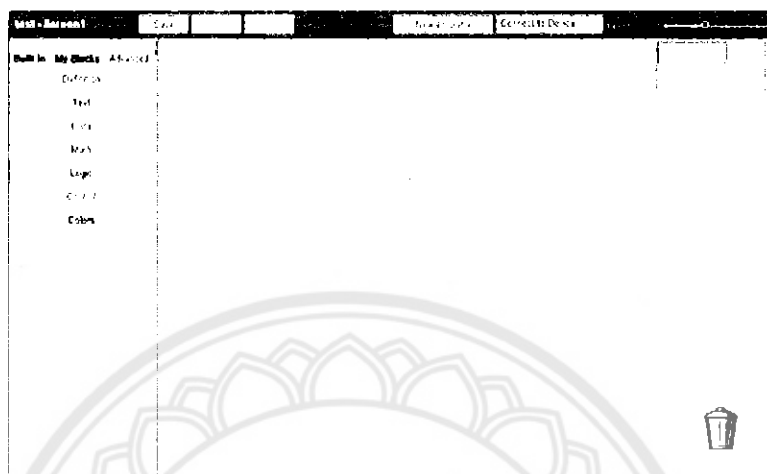
สามารถเลือกจัดวางลงในตำแหน่งที่เหมาะสมได้ตามต้องการ และคอมโพเนนท์นั้นยังปรากฏในส่วนรายการคอนโพเนนท์ (Components) เรียงกันเป็นรายการเพื่อให้ดูง่ายและสามารถเลือกคอมโพเนนท์ที่ต้องการกำหนดคุณสมบัติจากรายการนี้แล้วกำหนดคุณสมบัติต่างๆ ที่หน้าจอส่วนคุณสมบัติ (Properties) ซึ่งจะเป็นคุณสมบัติเฉพาะของคอมโพเนนท์นั้นๆ



ภาพส่วนรายการคอนโพเนนท์

นอกจากในกลุ่มของคอมโพเนนท์ทั่วไปแล้วยังมีคอมโพเนนท์ที่มองไม่เห็น (Non-Visible Components) ซึ่งเมื่อนำมาวางในหน้าจอวิวเวอร์แล้วจะไม่ปรากฏคอมโพเนนท์ดังกล่าวที่หน้าจอ วิวเวอร์ แต่จะปรากฏที่หน้าจอรายการคอนโพเนนท์แทน คอมโพเนนท์ที่มองไม่เห็นนี้จะประกอบไปด้วยคอมโพเนนท์ในกลุ่มเซนเซอร์ ซึ่งประกอบไปด้วยคอมโพเนนท์ที่เกี่ยวข้องกับการเรียกใช้ตัวตรวจจับต่างๆ ที่มีอยู่ในโทรศัพท์ เช่น ระบบ GPS หรือ Accelerometers เป็นต้น กลุ่ม Notifiers ซึ่งเกี่ยวข้องกับความสามารถในการแจ้งเตือนต่างๆ หรือการเขียนบันทึกกิจกรรมของโทรศัพท์ ซึ่งคอมโพเนนท์ในกลุ่ม Notifiers นั้นจะมองไม่เห็นหรือถูกซ่อนไว้ แต่จะสามารถมองเห็นได้เมื่อเกิดการแจ้งเตือนหรือสอบถามโดยมีการโต้ตอบกับผู้ใช้ในรูปแบบของข้อความ เสียง ปุ่มหรือช่องสำหรับกรอกข้อมูล ที่จะแสดงให้ผู้ใช้เห็นเป็นครั้งคราวเท่านั้น กลุ่ม Clocks ซึ่งเกี่ยวข้องกับฟังก์ชันของเวลา ตัวจับเวลา และการตั้งค่าเวลา กลุ่ม ActivityStarters ซึ่งเกี่ยวข้องกับการสั่งให้แอปพลิเคชันอื่นที่ติดตั้งอยู่ในโทรศัพท์ทำงาน เช่น โปรแกรมอ่านบาร์โค้ด (barcode scanner) หรือ

โปรแกรมอ่านออกเสียงจากข้อความ (text to speech) เป็นต้น กลุ่ม Web Services เช่น คอมโพเนนต์เกี่ยวกับเกมออนไลน์ (Game Client) คอมโพเนนต์เกี่ยวกับฐานข้อมูลบนเว็บ และคอมโพเนนต์เกี่ยวกับการใช้บริการ Twitter เป็นต้น

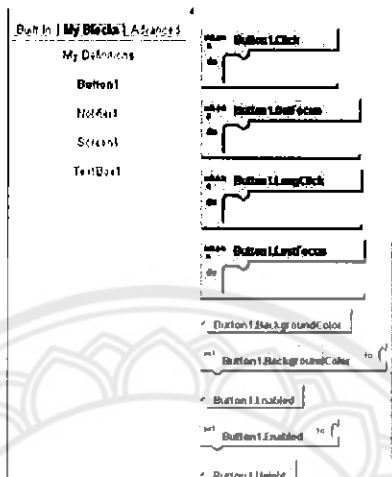


ภาพส่วนหน้าจอการเขียนโค้ดแสดง

1.1.2 ส่วนการเขียนโค้ด (App Inventor Blocks Editor)

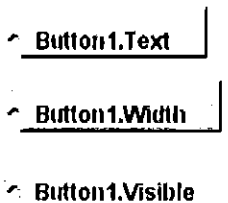
หลังจากที่ทำการเลือกจัดวางคอมโพเนนต์ที่จะใช้สำหรับโปรเจกต์ครบแล้ว ผู้ใช้จะสามารถเขียนโค้ดคำสั่งสำหรับแอปพลิเคชันได้ในส่วนการเขียนโค้ด (App Inventor Blocks Editor) สำหรับพื้นที่การทำงานในส่วนหน้าจอการเขียนโค้ดแสดง ซึ่งจะประกอบไปด้วยคำสั่งที่อยู่ในรูปของบล็อกกรวยรวมไว้บริเวณด้านซ้ายของหน้าจอ ผู้ใช้สามารถเลือกคำสั่งที่ต้องการโดยการคลิกลากบล็อกคำสั่งมาวางไว้ในโปรเจกต์คือบริเวณที่เป็นพื้นที่วางตรงกลางหน้าจอ ตัวอย่างของบล็อกคำสั่งดังแสดงในภาพที่ 4 ซึ่งจะเป็นคำสั่งพื้นฐานที่ผู้ใช้นำมาใช้ในการสร้างแอปพลิเคชันขึ้นมา บล็อกเหล่านี้จะถูกแยกและจัดแบ่งออกเป็นกลุ่มๆ ตามลักษณะของคำสั่ง ตัวอย่างเช่น บล็อกข้อความที่ใช้ในการทำงานที่เกี่ยวข้องกับข้อความที่เป็นสายอักขระ บล็อกทางคณิตศาสตร์ที่ใช้ในการทำงานที่เกี่ยวข้องกับฟังก์ชันทางคณิตศาสตร์ ตัวเลข หรือเครื่องหมายทางคณิตศาสตร์ เป็นต้น App Inventor ยังสามารถสร้างกระบวนการทำงาน (procedure) และตัวแปร (variable) ได้โดยการเลือกใช้บล็อกในส่วนที่เกี่ยวกับการสร้างกระบวนการทำงานและเหตุการณ์ (event handler) ที่เกิดกับคอมโพเนนต์ โดยบล็อกที่เกี่ยวข้องกับคอมโพเนนต์จะถูกจัดเตรียมไว้ให้ตามคอมโพเนนต์ที่ผู้ใช้เลือกนำมาวางไว้ในโปรเจกต์และจัดเก็บรวมกันไว้ในแท็บ My Blocks แยกไว้ต่างหาก บล็อกที่เกี่ยวข้องกับคอมโพเนนต์เหล่านี้จะแบ่งออกได้เป็น 4 แบบตามประเภทของคำสั่ง คือ ประเภทการ

เรียกค่าคุณสมบัติจากคอมโพเนนต์ (property getter) ประเภทการกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์ (property setter) ประเภทเหตุการณ์ (event handler) และประเภทการเรียกใช้กระบวนการทำงาน (method call)



ภาพตัวอย่างของบล็อกคำสั่ง

1) การเรียกค่าคุณสมบัติจากคอมโพเนนต์ (property getter) บล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านซ้าย โดยการทำงานจะทำการอ่านค่าคุณสมบัติจากคอมโพเนนต์แล้วส่งค่านั้นกลับมาในรูปของข้อความ ตัวเลข หรือค่าทางตรรกศาสตร์ แต่ในบางคอมโพเนนต์อาจมีค่าที่มีรูปแบบที่ซับซ้อนมากกว่าเช่น ค่า GPS จากคอมโพเนนต์ตรวจจับตำแหน่ง (Location Sensor) เป็นต้น ซึ่งจะมีรูปแบบเฉพาะแตกต่างกันไป แต่ทั้งนี้การอ่านค่านั้นทำได้ง่ายมาก ถึงแม้จะเป็นการอ่านค่า GPS ซึ่งโดยปกติมีกระบวนการทำงานที่ซับซ้อน แต่ผู้ใช้สามารถอ่านค่า GPS ได้ผ่านคอมโพเนนต์ตรวจจับตำแหน่งเหมือนอ่านค่าข้อความจากกล่องข้อความ ด้วยกระบวนการนี้ทำให้มั่นใจได้ว่าผู้ใช้จะไม่ต้องกังวลในเรื่องความซับซ้อนของการเข้าถึงค่าและข้อมูลต่างๆ ที่ต้องการ



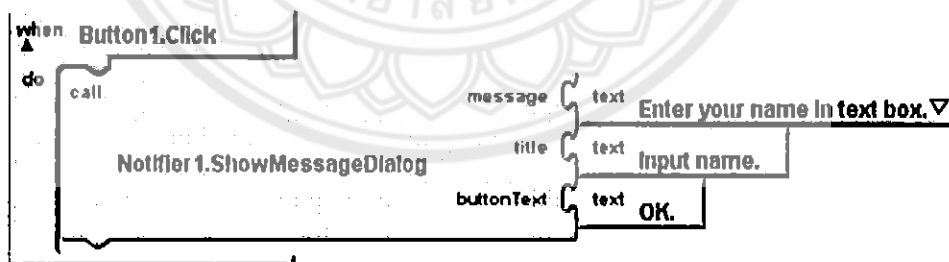
ภาพคอมโพเนนต์

2) การกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์ (property setter) บล็อกประเภทที่ใช้กำหนดค่าคุณสมบัติให้กับคอมโพเนนต์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านขวา โดยจะสามารถทำการกำหนดค่าหรือเปลี่ยนแปลงค่าคุณสมบัติให้กับคอมโพเนนต์ที่ต้องการด้วยค่าของบล็อกที่นำมาต่อเข้ากับช่องต่อที่อยู่ทางด้านขวา ช่องต่อนี้จะมีรูปร่างเป็นช่องรับซึ่งจะต่อเข้าได้พอดีกับบล็อกที่มีรูปร่างเหมือนบล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์ ซึ่งจะทำให้ผู้ใช้เลือกบล็อกที่จะนำมาต่อเข้าด้วยกันได้อย่างง่ายดายและลดข้อผิดพลาดในการเลือกต่อบล็อกที่ไม่ถูกต้อง



ภาพการกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์

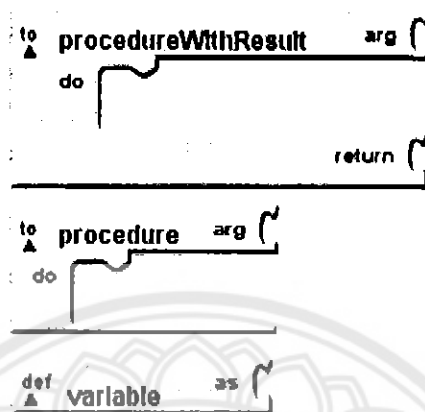
3) เหตุการณ์ (event handler) บล็อกประเภทเหตุการณ์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านล่าง ซึ่งบล็อกประเภทนี้จะทำงานเมื่อเกิดเหตุการณ์ต่างๆ ขึ้นกับคอมโพเนนต์ เช่น การคลิกที่ปุ่ม ซึ่งจะทำงานตามบล็อกคำสั่งที่ต่อลงไปทางด้านล่างภายในบล็อกเหตุการณ์ แสดงให้เห็นถึงเหตุการณ์เมื่อมีการคลิกปุ่มแล้วให้มีการแสดงหน้าต่างข้อความโต้ตอบขึ้นมาเพื่อเตือนให้ผู้ใช้ทำการป้อนข้อมูลลงในกล่องข้อความ เป็นต้น



ภาพบล็อกคำสั่งที่ต่อลงไปทางด้านล่างภายในบล็อกเหตุการณ์

4) การเรียกใช้กระบวนการทำงาน (method call) บล็อกประเภทเรียกใช้กระบวนการทำงานจะมีลักษณะเหมือนกับบล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์ที่มีลักษณะเป็นช่องต่ออยู่ทางด้านซ้าย บล็อกประเภทนี้จะถูกสร้างขึ้นเมื่อผู้ใช้ได้มีการสร้างกระบวนการทำงานโดยอาศัยบล็อกประเภทกระบวนการทำงานในการสร้าง เช่น การสร้างฟังก์ชันการทำงาน การสร้างตัวแปร เป็นต้น และจะมีชื่อเรียกเฉพาะตัวตามที่ผู้ใช้เป็นผู้ตั้งให้ เมื่อมีการสร้าง

กระบวนการทำงานขึ้น บล็อกประเภทเรียกใช้กระบวนการทำงานก็จะถูกสร้างขึ้นและปรากฏในตัวเลือกเพื่อให้ผู้ใช้เลือกที่มาวางลงในโปรเจกต์เมื่อต้องการให้เกิดการเรียนรู้ใช้กระบวนการทำงานดังกล่าว



ภาพกระบวนการทำงานในการสร้าง

1.1.3 ส่วนของการแพ็คเกจและการเรียกใช้งานแอปพลิเคชัน

เมื่อแอปพลิเคชันได้ถูกออกแบบและทำการเขียนโค้ดคำสั่งเป็นที่เรียบร้อยแล้ว ผู้ใช้สามารถสั่งให้แอปพลิเคชันทำการแพ็คเกจแอปพลิเคชันดังกล่าวให้อยู่รูปของไฟล์ที่พร้อมจะนำไปติดตั้งเพื่อนำไปติดตั้งบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ต่อไป ผู้ใช้เพียงเลือกคลิกที่ปุ่ม Package for Phone ที่อยู่ในด้านบนของหน้าจอส่วนออกแบบ โปรแกรมแอปพลิเคชันจะทำการแพ็คเกจบนเซิร์ฟเวอร์แอปพลิเคชันและส่งไฟล์ที่พร้อมจะนำไปติดตั้งออกมาให้ผู้ใช้

ทำการดาวน์โหลดเก็บไว้ในเครื่องคอมพิวเตอร์เพื่อนำไปติดตั้งและเรียกใช้งานบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์เครื่องใดก็ได้ หรือหากไม่มีโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ก็สามารถที่จะทดสอบการทำงานของแอปพลิเคชันได้บนโทรศัพท์จำลองที่ทำงานอยู่บนคอมพิวเตอร์ซึ่งจะมีลักษณะการทำงานเหมือนโทรศัพท์จริงทุกประการ



ภาพแอปพลิเคชันที่ได้บน โทรศัพท์



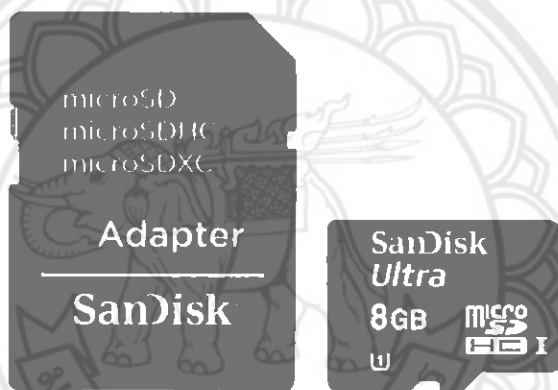


เริ่มต้นการติดตั้งระบบปฏิบัติการ

ก่อนเริ่มต้นการใช้งานบอร์ดราสเบอร์รี่พายจำเป็นที่จะต้องติดตั้งระบบปฏิบัติการให้กับบอร์ดก่อนเนื่องจากบอร์ดไม่มีหน่วยความจำแบบแฟลชเมมโมรี่มาบนบอร์ดด้วย ดังนั้น จำเป็นที่จะต้องเตรียมอุปกรณ์ต่างๆ ให้พร้อมเพื่อให้สามารถใช้งานบอร์ดได้ ซึ่งมีรายละเอียดอุปกรณ์ดังนี้

1. บอร์ดราสเบอร์รี่พาย

2. SD Card สำหรับติดตั้งระบบปฏิบัติการลินุกซ์ ต้องมีความจุมากกว่า 2 GB ขึ้นไป แต่แนะนำให้ใช้ ขนาด 4GB หรือมากกว่า สำหรับคู่มือฉบับนี้จะใช้ขนาด 8GB ควรเลือกใช้การ์ดที่มีความเร็วสูงอย่าง Class 10 เพื่อประสิทธิภาพการทำงานของระบบโดยรวม



SanDisk Micro SD Ultra 8GB 30MB/s Class10 with Adapter

3. เมาส์และคีย์บอร์ดแบบ USB

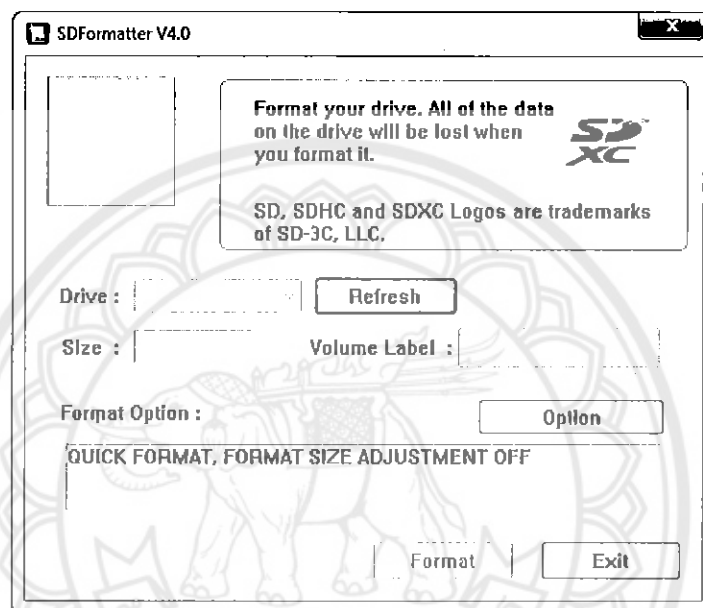
4. สาย Micro USB เพื่อจ่ายไฟเลี้ยงวงจร สามารถเลือกใช้แหล่งจ่ายไฟจากพอร์ต USB ของเครื่องคอมพิวเตอร์ได้

5. สาย HDMI เพื่อเชื่อมต่อกับจอแสดงผล หากเลือกใช้จอมอนิเตอร์ที่ไม่มีพอร์ต HDMI รองรับต้องใช้ตัวแปลง HDMI to VGA ด้วย หรือเชื่อมต่อสายวีดีโอ RCA ก็ได้เช่นเดียวกัน (เลือกอย่างใดอย่างหนึ่ง)

เตรียมซอฟต์แวร์ สำหรับติดตั้งระบบปฏิบัติการลินุกซ์ลงบนบอร์ดราสเบอร์รี่พาย

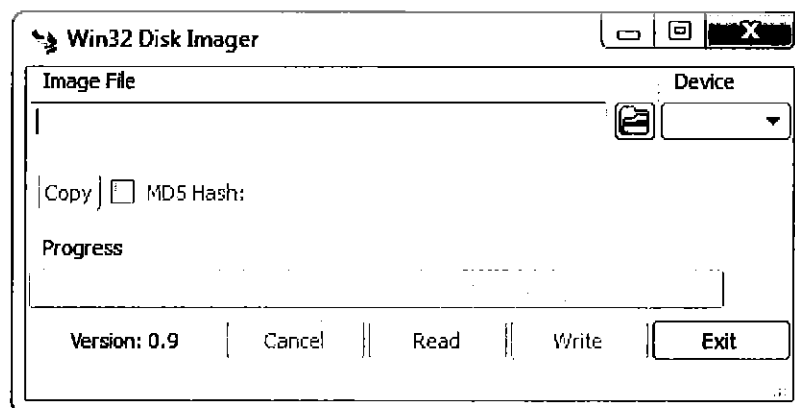
คู่มือฉบับนี้จะจัดเตรียมซอฟต์แวร์ที่รองรับระบบปฏิบัติการวินโดวส์เป็นหลัก และต้องติดตั้งลงบนเครื่องคอมพิวเตอร์ดังนี้

1. โปรแกรม SD Formatter 4.0 ใช้สำหรับ Format Disk สามารถดาวน์โหลดได้จากลิงค์ https://www.sdcard.org/downloads/formatter_4/eula_windows/



ตัวอย่างหน้าต่าง โปรแกรม SD Formatter Version 4.0

2. โปรแกรม Win32 Disk Imager ใช้สำหรับเขียนไฟล์ระบบปฏิบัติการที่เป็นไฟล์ Image (*.img) ลงบน SD Card สามารถดาวน์โหลดได้จากลิงค์ <http://sourceforge.net/projects/win32diskimager/>



ตัวอย่าง โปรแกรม Win32 Disk Imager

3. ไฟล์ระบบปฏิบัติการ คู่มือนี้ติดตั้งระบบปฏิบัติการราสเบียนเป็นระบบปฏิบัติการ Debian Wheezy ที่ถูกปรับแต่งให้ใช้สำหรับบอร์ดราสเบอร์รี่พาย โดยเฉพาะ เป็นลินุกซ์ ที่ให้ใช้งานได้ฟรี สามารถดาวน์โหลดได้จากลิงค์ <http://www.raspberrypi.org/downloads>

ขั้นตอนการติดตั้งระบบปฏิบัติการ Raspbian ให้กับบอร์ดราสเบอร์รี่พาย

1. หากมีข้อมูลอยู่ในเอสดีการ์ดให้ทำการ Format ด้วยโปรแกรม SD Formatter 4.0 หรือ โปรแกรมอื่นๆ ก็ได้ ถ้าหาก Format แล้วให้ข้ามขั้นตอนนี้ได้เลย

2. เมื่อดาวน์โหลดไฟล์ระบบปฏิบัติการ Raspbian มาแล้วจะได้เป็นไฟล์ Zip ให้แตกไฟล์จะได้เป็นไฟล์ Image (*.img) มาแสดงดังรูป

Name

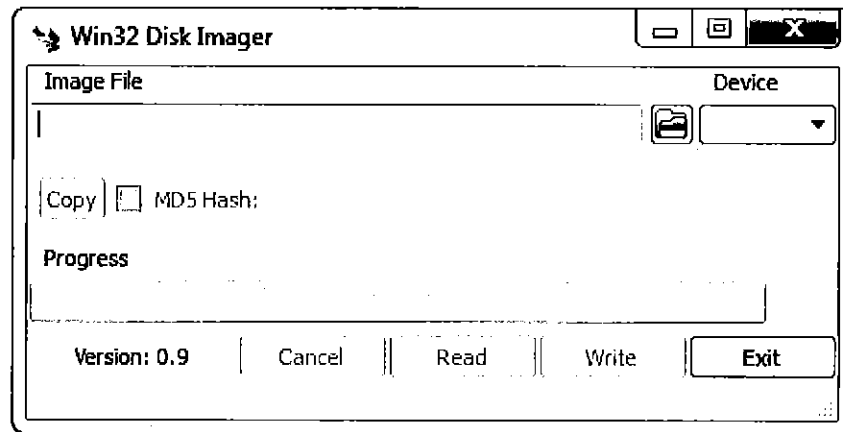
2013-09-25-wheezy-raspbian.img

ไฟล์ Image

3. เมื่อดาวน์โหลดโปรแกรม Win32 Disk Imager มาแล้วจะได้เป็นไฟล์ Zip ให้แตกไฟล์และรันโปรแกรมแสดงดังรูป

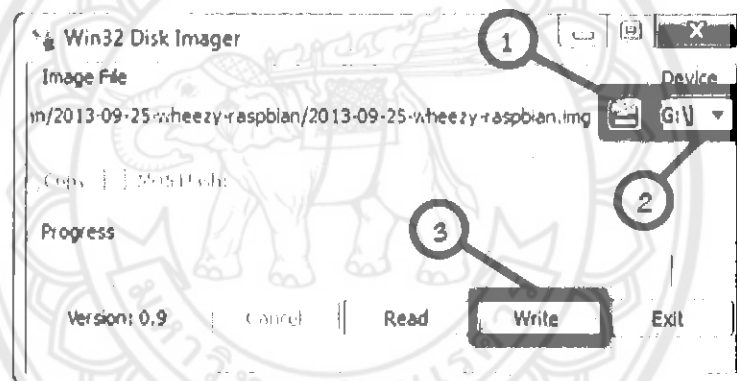
- Changelog
- GPL-2
- LGPL-2.1
- libgcc_s_dw2-1.dll
- libstdc++-6.dll
- mingwm10.dll
- QtCore4.dll
- QtGui4.dll
- README
- Win32DiskImager

รันไฟล์ Win32DiskImager

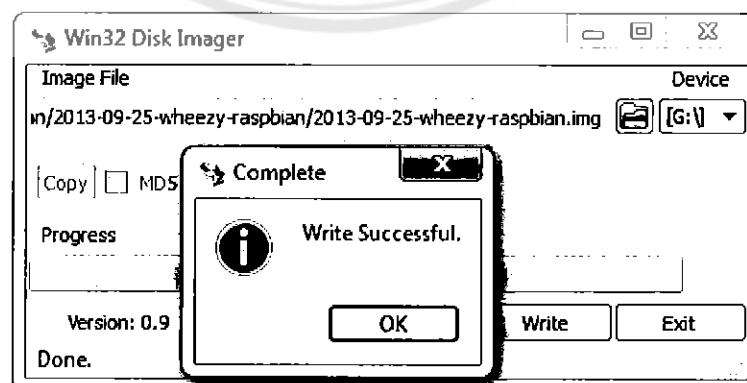


หลังจากรัน โปรแกรมจะปรากฏหน้าต่างโปรแกรม

4. ให้ Browse ไฟล์ Image ระบบปฏิบัติการ Raspbian (*.img) และเลือก Device ให้ถูกต้อง แล้วคลิกปุ่ม Write แสดงดังรูป และจะปรากฏหน้าต่างยืนยัน ให้คลิกปุ่ม Yes

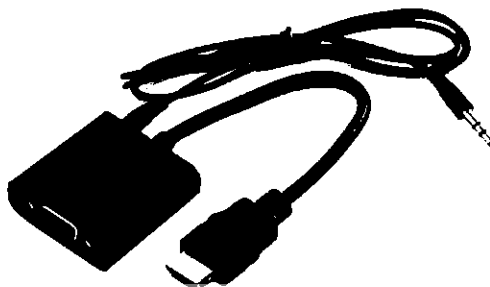


5. รอจนกว่า Progress Bar ครบ 100% และปรากฏหน้าต่างแสดงดังรูป แล้วให้กดปุ่ม OK และ Exit



6. หากเชื่อมต่อบอร์ดราสเบอร์รี่พายกับจอคอมพิวเตอร์ผ่านอุปกรณ์แปลง HDMI-to-VGA ให้แก้ไขไฟล์ config.txt ตามเอกสารในลิงค์

http://www.thaieasyelec.com/downloads/EACC027/HDMI_to_VGA_Cofiguration.zip



7. จากนั้นถอด SD Card ออกจากเครื่องคอมพิวเตอร์ แล้วนำไปเสียบที่บอร์ดราสเบอร์รี่พาย



ที่มา: <http://robotics.keckist.edu.np/category/raspberry-pi>

8. หลังจากเสียบการ์ดความจำเรียบร้อยแล้ว ให้เสียบเมาส์ คีย์บอร์ด สายต่อจอแสดงผล HDMI หรือ RCA สายไฟเลี้ยงวงจรบอร์ด Micro USB และอื่นๆ

9. หลังจากนั้นบอร์ดราสเบอร์รี่พายก็จะเริ่มทำงาน และเริ่ม Boot ระบบดังรูป

```

Using makefile-style concurrent boot in runlevel 2.
Network Interface Plugging Daemon...skip eth0...done.
Starting NFS common utilities: statd.
Starting enhanced syslogd: rsyslogd.
Starting system message bus: dbus.
Starting periodic command scheduler: cron.
Starting NTP server: ntpd.
Starting portmap daemon...already running..
Starting Hardware abstraction layer: hald.
Starting internet superserver: xinetd.
My network IP address is 10.0.2.15

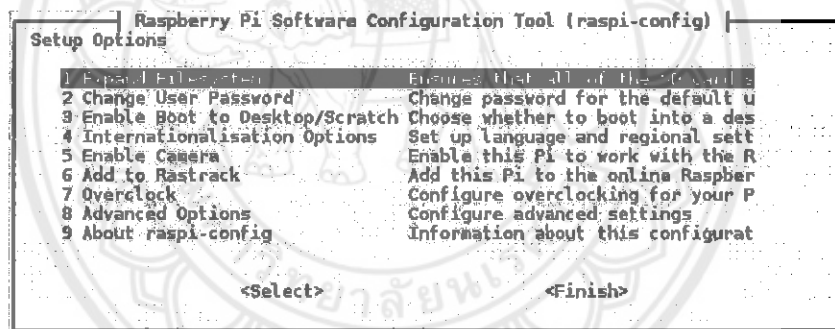
Debian GNU/Linux 6.0 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 3.1.9+ #2 Mon Apr 16 04:53:15 EST 2012; root@pi

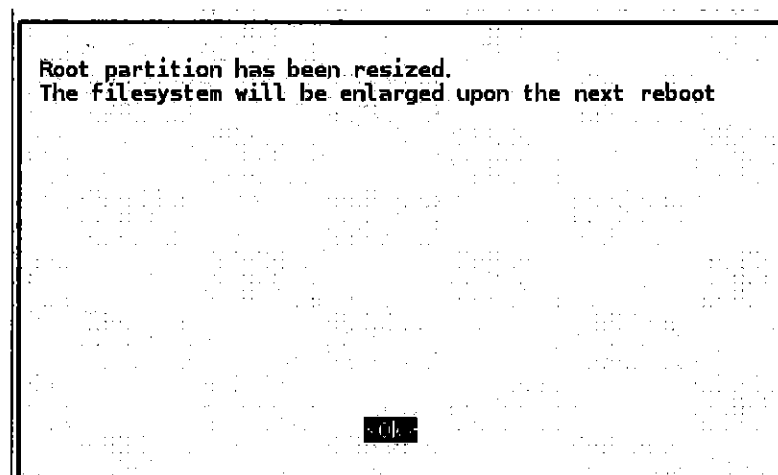
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

```

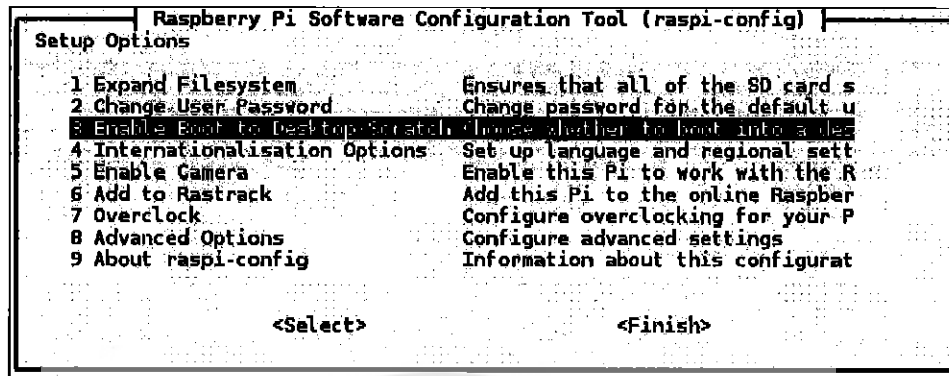
10. หลังจากระบบปฏิบัติการ Boot เสร็จเรียบร้อยแล้วจะปรากฏหน้าต่างแสดงดังรูป ให้เลือกเมนู 1 Expand Filesystem เพื่อขยายพื้นที่บนการ์ดความจำให้ใช้งานได้เต็มความจุ เลือกด้วยลูกศรขึ้นลง แล้วกด Enter



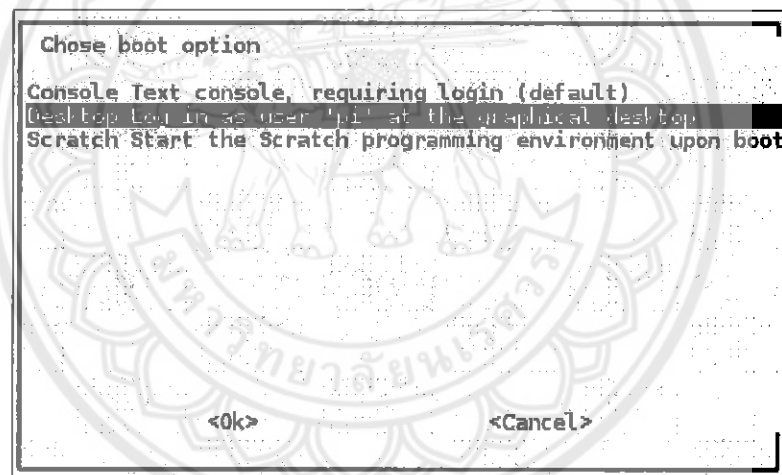
11. จะปรากฏหน้าต่างแสดงดังรูป ให้กด Enter อีกครั้ง



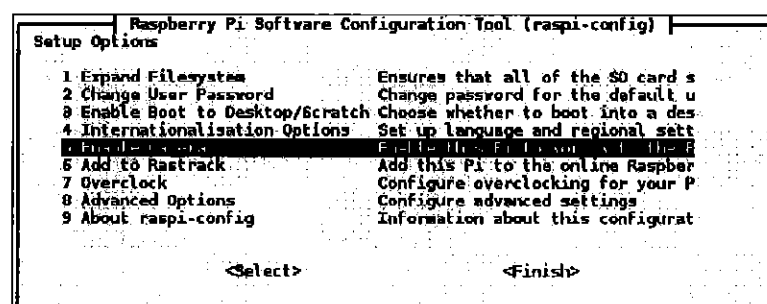
12. กำหนดรูปแบบการใช้งานระบบปฏิบัติการให้ใช้งานในโหมด Graphic ให้เลือกเมนู 3 Enable Boot to Desktop/Scratch แล้วกดแป้นพิมพ์ Enter



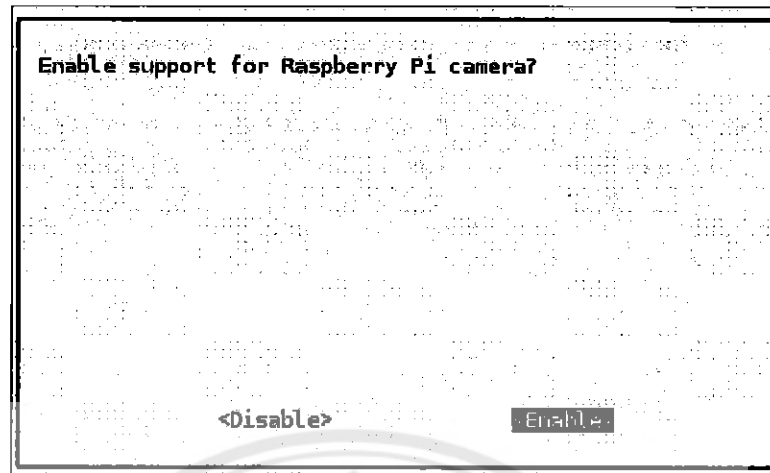
13. หลังจากนั้นจะปรากฏตัวเลือกมาทั้งหมด 3 ตัวเลือก ให้เลือก Desktop Log in as user 'pi' at the graphical desktop แล้วกดแป้นพิมพ์ Enter



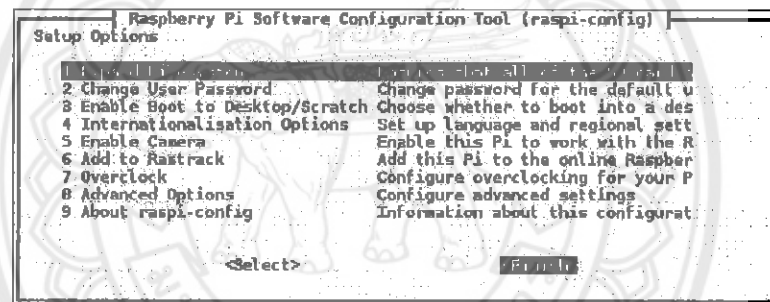
14. หากต้องการใช้งาน โมดูลกล้อง (Raspberry Pi Camera Module) ต้องเปิดฟังก์ชันการใช้งานนี้ ด้วย โดยเลือกเมนู 5 Enable Camera แล้วกด Enter แต่ถ้าหากไม่ต้องการใช้ให้ข้ามขั้นตอนนี้ไป



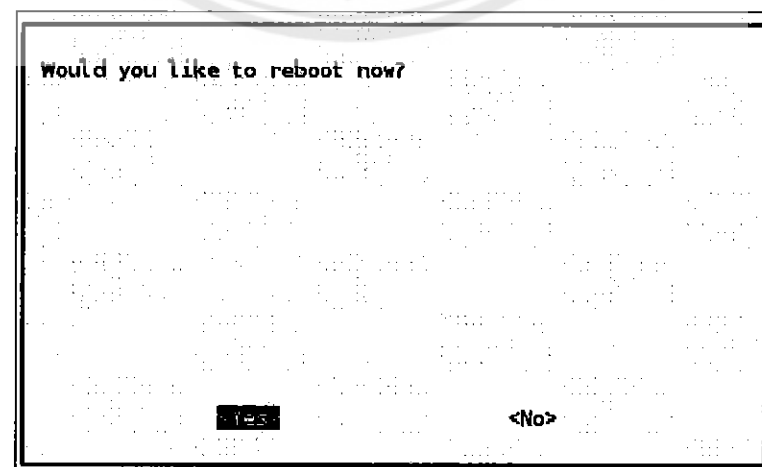
15. หลังจากนั้นให้เลือก Enable แล้วกดแป้นพิมพ์ Enter



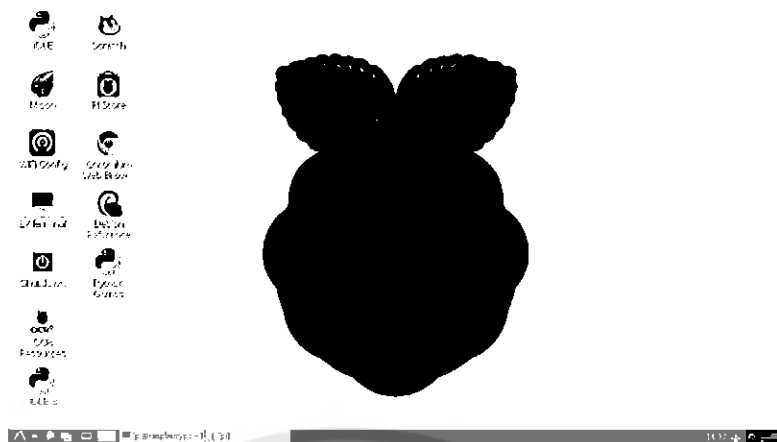
16. สุดท้ายให้เลื่อนไปที่ Finish แล้วกด Enter เพื่อบันทึกค่าระบบและ Reboot ระบบใหม่



17. ระบบจะถามอีกครั้งว่าต้องการจะ Reboot ระบบใหม่นี้หรือไม่ เลือก Yes แล้วกด Enter ระบบก็จะ Reboot ใหม่ทันที



18. หลังจากทีระบบ Reboot ใหม่เสร็จเรียบร้อยแล้ว จะแสดงหน้า Desktop ของ Raspbian ดังรูป





APACHE



อาปาเช่ เป็นเว็บเซิร์ฟเวอร์เพียงหนึ่งเดียวที่อยู่คู่กับระบบปฏิบัติการลินุกซ์ทุกคิสรวิวัฒน์ มาเป็นเวลาหลายปีแล้ว เช่นเดียวกับลินุกซ์เรดแฮทที่ได้รวมเอาโปรแกรมอาปาเช่ไว้ในชุดติดตั้ง พร้อมให้เราใช้งานได้ที่ไหนก็ได้ ไม่ต่างอะไรกับบะหมี่สำเร็จรูป แค่เติมน้ำร้อนลงไปก็รับประทานได้ทันที จึงไม่ใช่เรื่องยากเลยที่เราจะตั้งเครื่องพีซีซักตัวหนึ่งขึ้นเป็นเว็บเซิร์ฟเวอร์ ให้บริการเว็บได้ทั้งระบบอินเทอร์เน็ตภายในองค์กร ไปจนถึงจัดตั้งเว็บไซด์เผยแพร่ข้อมูลไปทั่วโลก และนี่คืออีกหนึ่งการนำเอาลินุกซ์มาใช้งานที่คุ้มค่าที่สุด

เว็บเซิร์ฟเวอร์ที่นิยมใช้มากที่สุดในโลก

ข้อมูลการสำรวจจากเว็บไซด์ทั่วโลกโดย Netcraft เป็นสิ่งที่ยืนยันถึงจำนวนของอาปาเช่เว็บเซิร์ฟเวอร์ที่มีสัดส่วนการใช้งานสูงกว่าเว็บเซิร์ฟเวอร์อื่น ๆ อย่างเห็นได้ชัด และยังมีแนวโน้มที่จะเพิ่มสูงมากยิ่งขึ้นอีกด้วย จากจุดเริ่มต้นที่อาศัยโค้ดจากเว็บเซิร์ฟเวอร์มาตรฐาน NCSA (องค์กรกลางผู้กำหนดมาตรฐานโปรโตคอล HTTP ,มาตรฐานภาษา HTML และมาตรฐานอื่น ๆ ที่เกี่ยวข้องกับงานบริการบนเว็บทั้งหมด) พัฒนาอย่างต่อเนื่องด้วยกำลังของชุมชนนักพัฒนาจากทุกมุมโลก ผ่านโมเดลการพัฒนาแบบฟรีซอฟต์แวร์ ภายใต้การกำกับดูแลของ Apache Foundation (<http://www.apache.org>) ทำให้เกิดซอฟต์แวร์เว็บเซิร์ฟเวอร์ที่มีเสถียรภาพการทำงานที่เชื่อถือได้ มีประสิทธิภาพสูง และแข็งแกร่ง

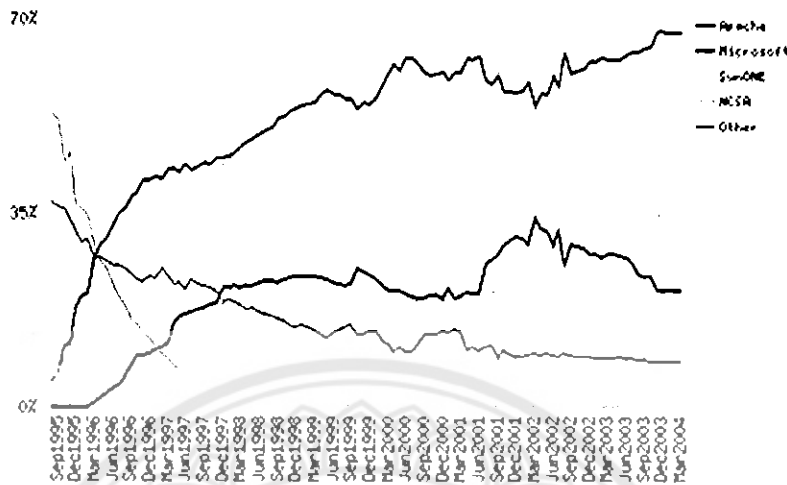
จากซอฟต์แวร์ที่เริ่มต้นจากส่วนประกอบเล็ก ๆ หรือ "patches" จำนวนมากมาย จนทำให้ถูกเรียกขานว่า " a patchy " ผ่านช่วงระยะเวลาของการพัฒนามาถึงสิบปี จนกลายมาเป็นเว็บเซิร์ฟเวอร์ที่ได้มาตรฐาน และได้รับความนิยมสูงสุดในวันนี้ อาปาเช่ยังคงความเป็นฟรีซอฟต์แวร์ไว้อย่างมั่นคง กล่าวได้ว่าถึงวันนี้อาปาเช่เป็นแม่แบบของฟรีซอฟต์แวร์ที่ประสบความสำเร็จแล้วในโลกของความเป็นจริง และเป็นอีกหนึ่งโปรเจกต์ที่ก้าวข้ามพันอุปสรรคของโมเดลการพัฒนาแบบฟรีซอฟต์แวร์ได้สำเร็จแล้ว

March 2004 Web Server Survey

Web Server Survey

In the March 2004 survey we received responses from 46,038,131 sites.

Market Share for Top Servers Across All Domains August 1995 - March 2004



Top Developers

| Developer | February 2004 | Percent | March 2004 | Percent | Change |
|-----------|---------------|---------|------------|---------|--------|
| Apache | 31703884 | 67.21 | 32280582 | 67.20 | -0.01 |
| Microsoft | 9849971 | 20.88 | 10099760 | 21.02 | 0.14 |
| SunONE | 1657295 | 3.51 | 1651575 | 3.44 | -0.07 |
| Zeus | 755227 | 1.60 | 762716 | 1.59 | -0.01 |

รายงานผลสำรวจเว็บเซิร์ฟเวอร์จาก NetCraft

ประโยชน์จากเว็บเซิร์ฟเวอร์

ไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ชนิดใดก็ตาม คุณประโยชน์ที่จะได้รับย่อมเป็นสิ่งแรกที่เราต้องคำนึงถึง ความหมายสั้น ๆ ของบริการบนเว็บก็คือ มันเป็นวิธีที่ง่ายที่สุดที่จะเผยแพร่เอกสารข้อมูลไปในเครือข่ายคอมพิวเตอร์ เพราะเพียงแค่ผู้ใช้บริการเปิดโปรแกรมเว็บเบราว์เซอร์ (ซึ่งมีติดตั้งอยู่ในคอมพิวเตอร์หรือเครื่องมือสื่อสารสมัยใหม่อยู่แล้ว) ก็จะสามารถเข้าถึงเอกสารเว็บได้โดยอ้างชื่อของเว็บไซต์ ต่อจากนั้นก็พบกับเอกสารข้อความ สื่อมัลติมีเดีย บริการดาวน์โหลด และกิจกรรมที่เป็นอินเทอร์เน็ตที่ฟาสต์ได้อย่างง่ายดาย

เว็บเซิร์ฟเวอร์จะเป็นศูนย์กลาง หรือจุดเริ่มต้นของการประยุกต์ใช้งานเครือข่ายคอมพิวเตอร์ของคุณได้อย่างหลากหลาย เริ่มต้นจากการเป็นแหล่งเผยแพร่ข่าวสารในองค์กร เปิดให้บริการอีเมลผ่านเว็บ (Web based Mail Services) รวมไปถึงการใช้งานแอปพลิเคชันผ่านเว็บหรือ Web based Application ที่พัฒนาขึ้น โดยเฉพาะ ซึ่งมีองค์กรเป็นจำนวนมากที่เริ่มหันมาสนใจพัฒนาแอปพลิเคชันเฉพาะขององค์กรในลักษณะเช่นนี้ ทั้งนี้เหตุผลหลักก็คือความต้องการที่จะพัฒนาแอปพลิเคชันขึ้น โดยตั้งอยู่บนระบบโครงสร้างที่เป็นมาตรฐาน ซึ่งจะช่วยลดปัญหาที่จะต้องเผชิญกับการเปลี่ยนแปลงของเทคโนโลยีที่ถูกกำหนดขึ้น โดยผู้ผลิตซอฟต์แวร์หรือฮาร์ดแวร์รายใหญ่นั้นเอง

สำหรับองค์กรที่มีขนาดเล็ก และยังขาดความพร้อมในเรื่องที่จะพัฒนาซอฟต์แวร์ขึ้นใช้งานเอง ก็ยังมีทางเลือกออกอีกมากที่จะนำแอปพลิเคชันสำเร็จรูปมาใช้งานร่วมกับเว็บเซิร์ฟเวอร์ได้เช่นกัน ซึ่งมีโปรเจกต์ในแบบฟรีซอฟต์แวร์ให้เลือกใช้ได้เช่นกัน และส่วนใหญ่จะสามารถใช้ได้เป็นอย่างดีกับอาปาเซ่ เนื่องจากความแพร่หลายของอาปาเซ่ในหมู่นักพัฒนาซอฟต์แวร์นั่นเอง

ติดตั้งคอนฟิกแบบเร่งด่วน

ดังที่ได้กล่าวไปแล้วว่าในเรดแฮตลินุกซ์จะมีโปรแกรมอาปาเซ่เว็บเซิร์ฟเวอร์ เวอร์ชัน 2.0 ให้มาด้วยแล้ว คุณทราบหรือไม่ว่าขั้นตอนการปลุกให้อาปาเซ่ตื่นขึ้นมาทำงานมันเป็นเรื่องง่ายยิ่งกว่าการโทรไปสั่งพิชซ่าเสียอีก ถ้าคุณติดตั้งเรดแฮตลินุกซ์ในแบบเลือกครบทุกแพคเกจ หรือ EveryThing ก็จะมีโปรแกรมนี้พร้อมอยู่แล้วในเครื่องอย่างแน่นอน ซึ่งแพคเกจของโปรแกรมนี้นี้จะชื่อว่า httpd ดังนั้นเราจะลองค้นหาแพคเกจที่ติดตั้งอยู่ในเครื่องได้ด้วยคำสั่ง rpm ดังนี้

```
# rpm -qa | grep http
```

ถ้ามีแพคเกจนี้เรียบร้อยแล้ว และได้เซตค่าคอนฟิกเกี่ยวกับระบบเครือข่าย TCP/IP แล้ว การสั่งให้อาปาเซ่ทำงานจะใช้คำสั่งดังนี้

```
# service httpd restart
```

```
# chkconfig --level 35 httpd on
```

จะเป็นการสั่งให้บริการของ อาปาเซ่ เริ่มทำงานใหม่ (กรณีที่ไม่เคยเปิดให้บริการมาก่อนจะแจ้ง Fail ขณะ Shutdown จึงถือว่าเป็นเรื่องปกติ) ส่วนอีกคำสั่งเป็นการกำหนดให้ อาปาเซ่ เริ่มต้นทำงานเองเมื่อเริ่มเปิดเครื่องใหม่โดยอัตโนมัติ ซึ่งตัวบริการของอาปาเซ่จะคอยให้บริการที่ TCP port หมายเลข 80 และ 443 (เป็น โปรโตคอล HTTP และ HTTPS ตามลำดับ) คำสั่ง netstat จะแสดงให้เห็นการทำงานดังกล่าว

```

root@Jack:~# netstat -lnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:1024             0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:1025           0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:783           0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:111              0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:6000             0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:80               0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:22               0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:631            0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:25             0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:443              0.0.0.0:*                LISTEN
root@Jack:~#

```

ตรวจดูพอร์ตของอาปาเช่

เมื่อเห็นพอร์ต 80 ปรากฏขึ้น แสดงว่าอาปาเช่เริ่มให้บริการแล้ว เมื่อลองเปิดโปรแกรมเว็บเบราว์เซอร์เข้ามาที่โฮสต์ที่เป็นเว็บเซิร์ฟเวอร์นี้ เช่น ถ้าคุณติดตั้งโปรแกรม Mozilla ไว้ในเครื่องแล้ว ก็เรียกไปที่ `http://localhost` ก็จะเห็นหน้า Test Page ที่เรดแฮตสร้างไว้ให้ แค่พิมพ์คำสั่งไม่กี่ครั้งก็ทำให้เว็บเซิร์ฟเวอร์เริ่มทำงานแล้ว



หน้า Test Page ของอาปาเช่

ลักษณะทางกายภาพของอปาเซ่

อปาเซ่ถูกสร้างขึ้นจากการนำเอาโปรแกรมขนาดเล็กที่ทำหน้าที่แตกต่างกันหลาย โมดูลมาทำงานร่วมกันเป็นเว็บเซิร์ฟเวอร์ จึงทำให้มีส่วนประกอบเป็น โมดูล (ที่พัฒนาด้วยภาษาซี) ส่วนหนึ่งจะเป็นส่วนแกนกลางที่ทำหน้าที่บริหารจัดการทั้งหมด เรียกว่า Core.c โมดูลต่อมาคือ โมดูลที่ทำหน้าที่บริหารหน่วยความจำ (Memory Management) และบริหาร โพรเซสงานย่อย (Child Process) ที่รองรับการให้บริการที่เรียกเข้ามาพร้อม ๆ กันจำนวนมากจากภายนอก (Multi-Processing Models หรือ MPM) ซึ่งอปาเซ่มีโมเดลการทำงานด้านนี้รองรับไว้ 3 โมเดลด้วยกัน คือ Workers สำหรับรองรับงานจำนวนมากๆ ในขณะที่ต้องการหน่วยความจำไม่มากนัก Prefork สำหรับงานที่ต้องการประสิทธิภาพและความเร็วแต่จะต้องใช้ทรัพยากรระบบมากกว่า และ Per Child ออกแบบมาเพื่อรองรับงานได้แตกต่างกันโดยแยกตามยูสเซอร์ที่ร้องขอบริการเข้ามา (ขณะนี้อยู่ระหว่างการพัฒนา)

โมดูล http_core.c จะทำหน้าที่รองรับการประมวลผลด้วย โพรโตคอล HTTP (Hyper Text Transfer Protocol) ซึ่งจะจัดการกับส่วนเซคเตอร์ตามมาตรฐาน NCSA และ โมดูล mod_so.c จะทำหน้าที่ติดต่อประสานการทำงาน โมดูลภายในเข้ากับกับ Shared Modules อื่น ๆ ที่อยู่ภายนอก ซึ่ง โมดูลภายนอกเหล่านี้เราเรียกว่า Dynamic Shared Object หรือ DSO จะมีจำนวนมากน้อยก็ขึ้นอยู่กับความต้องการใช้งานของผู้ใช้ โดยจะกำหนดได้ในขณะที่คอมไพล์โปรแกรมอปาเซ่ สำหรับกรณีของ Red Hat 9.0 จะมีการคอมไพล์มาให้เรียบร้อยแล้ว และมีโมดูล DSO ที่มีคุณสมบัติด้านต่าง ๆ เพียงพอต่อการใช้งานทั่วไป เช่น การสนับสนุนภาษาสคริปต์ การทำ Authentication แบบต่าง ๆ การสนับสนุน Server Side Include เป็นต้น โดยในส่วนของแกนกลางหรือ MPM จะเป็นโมเดลแบบ prefork เราสามารถตรวจสอบดูส่วนประกอบของอปาเซ่ได้ด้วยคำสั่ง

```

root@jack:~#
File Edit View Terminal Go Help
[root@jack root]#
[root@jack root]# httpd -l
Compiled in modules:
  core.c
  prefork.c
  http_core.c
  mod_so.c
[root@jack root]# ls /etc/httpd/modules/
libphp4.so          mod_cern_meta.so  mod_info.so       mod_setenvif.so
mod_access.so      mod_cgid.so       mod_log_config.so  mod_spelling.so
mod_actions.so     mod_tgi.so        mod_mime_magic.so  mod_ssl.so
mod_alias.so       mod_dav_fs.so     mod_mime.so        mod_status.so
mod_asis.so        mod_dav.so        mod_negotiation.so  mod_suexec.so
mod_auth_anon.so   mod_deflate.so    mod_perl.so        mod_unique_id.so
mod_auth_dfn.so    mod_dir.so        mod_proxy_connect.so  mod_userdir.so
mod_auth_digest.so  mod_env.so        mod_proxy_ftp.so   mod_usertrack.so
mod_auth_mysql.so  mod_expires.so    mod_proxy_http.so  mod_vhost_alias.so
mod_auth_pgsq1.so  mod_headers.so    mod_proxy.so
mod_auth.so        mod_include.so    mod_python.so
mod_autoindex.so   mod_include.so    mod_rewrite.so
[root@jack root]#

```

รายชื่อ โมดูลที่เป็นส่วนประกอบของอปาเซ่ใน Red Hat 9.0

โครงสร้างไดเรกทอรีที่สำคัญ

ในฐานะผู้ดูแลระบบที่จะต้องคอนฟิก Red Hat Linux ให้ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ สิ่งที่ต้องทราบในเบื้องต้นก็คือ เรื่องไดเรกทอรีของอพาเช่ ซึ่งแบ่งออกได้ 3 ส่วนใหญ่ ๆ คือ ServerRoot ,DocumentRoot และ ScriptAlias ServerRoot หมายถึง ไดเรกทอรีที่เป็นจุดเริ่มต้นของส่วนประกอบด้านคอนฟิกทั้งหมดของอพาเช่ สำหรับ Red Hat Linux 9.0 คือที่ /etc/httpd ซึ่งจะแยกออกเป็น build สำหรับการเพิ่ม โมดูลเข้าสู่เว็บเซิร์ฟเวอร์ conf เป็นที่เก็บคอนฟิกไฟล์หลักคือ httpd.conf นั่นเอง conf.d เป็นไดเรกทอรีที่ใช้เพิ่มเติมไฟล์คอนฟิกย่อยให้แก่เว็บเซิร์ฟเวอร์เพื่อความสะดวกในการปรับแก้คอนฟิกได้สะดวกกว่าการแก้ไขที่ httpd.conf เพียงจุดเดียว logs เป็นไดเรกทอรีที่ใช้เก็บล็อกไฟล์ที่บันทึกเหตุการณ์ต่าง ๆ ที่เกิดขึ้นกับเว็บเซิร์ฟเวอร์ modules เป็นไดเรกทอรีที่เก็บไฟล์โมดูล DSO ทั้งหมดไว้ และ run เป็นไดเรกทอรีที่เก็บบันทึกหมายเลขโปรเซสของอพาเช่ไว้เพื่อการควบคุมโปรเซสทั้งจากภายในและภายนอกเซิร์ฟเวอร์เอง โครงสร้างของ ServerRoot

```

root@jack:~#
File Edit View Terminal Go Help
[root@jack root]#
[root@jack root]# tree -d /etc/httpd
/etc/httpd
|-- build -> ../../usr/lib/httpd/build
|-- conf
|   |-- ssl.crl
|   |-- ssl.crt
|   |-- ssl.csr
|   |-- ssl.key
|   |-- ssl.prm
|-- conf.d
|-- logs -> ../../var/log/httpd
|-- modules -> ../../usr/lib/httpd/modules
-- run -> ../../var/run

11 directories
[root@jack root]#

```

โครงสร้างของ ServerRoot

DocumentRoot เป็นไดเรกทอรีที่ผู้ใช้งานมักจะให้ความสำคัญที่สุด เพราะ document หรือ เอกสารภาษา HTML ที่เราต้องการเผยแพร่ผ่านทางเว็บเซิร์ฟเวอร์จะเริ่มต้นแสดงผลเป็นหน้าแรกจากไดเรกทอรีนี้นั่นเอง หรือจะมองว่าเป็น Home Page เลยก็ได้ สำหรับ Red Hat Linux 9.0 จะกำหนดให้ไดเรกทอรี /var/www/html เป็น DocumentRoot ของอพาเช่ ซึ่งผู้ดูแลระบบอาจจะไปเปลี่ยนแปลงให้ใช้ไดเรกทอรีอื่นทำหน้าที่นี้แทนได้ตามต้องการ โดยที่สามารถทำได้หลายวิธีซึ่งผู้เขียนจะสาธิตให้เป็นตัวอย่างดังรูปที่ 6 เป็นการกำหนด DocumentRoot ใหม่ไปที่ไดเรกทอรี /itdestination.com โดยสร้างรอไว้ก่อน จากนั้นเปลี่ยนชื่อ /var/www/html ของเดิมที่มากับ Red Hat

Linux ไปเป็นชื่อ /var/www/html.original และสุดท้ายจึงใช้ Symbolic Link สร้างจุดเชื่อมโยงชื่อ html ขึ้นแทนที่เพื่อนำเข้าสู่ /itdestination.com เป็นอันเสร็จสิ้นการย้ายตำแหน่งไดเรกทอรี DocumentRoot โดยไม่ต้องแก้ไขคอนฟิกของอาปาเช่เลยแม้แต่บรรทัดเดียว

```

root@jack:~
File Edit View Terminal Go Help
[root@jack root]#
[root@jack root]# mkdir /itdestination.com
[root@jack root]# mv /var/www/html /var/www/html.original
[root@jack root]# ln -s /itdestination.com /var/www/html
[root@jack root]# ls -l /var/www
total 28
drwxr-xr-x  2 root    root      4096 Mar  8 23:19 cgi-bin
drwxr-xr-x  3 root    root      4096 Mar  8 22:37 error
lrwxrwxrwx  1 root    root          18 Mar 12 12:03 html -> /itdestination.com
drwxr-xr-x  4 root    root      4096 Mar  8 23:18 html.original
drwxr-xr-x  3 root    mailnan  4096 Mar  8 23:22 icons
drwxr-xr-x 13 root    root      4096 Mar  8 22:37 manual
drwxr-xr-x  2 root    root      4096 Mar  8 23:17 nut-cgi-bin
drwxr-xr-x  6 root    root      4096 Mar  8 23:59 wordtrans
[root@jack root]#

```

สถิติวิธีการย้ายตำแหน่ง DocumentRoot อย่างรวดเร็ว

ScriptAlias ไดเรกทอรีนี้จะถูกกำหนดให้เป็นที่รันโปรแกรม CGI (Common Gateway Interface) โดยเฉพาะซึ่งมีลักษณะเป็นโปรแกรมสคริปต์หรือไบนารีก็ได้ที่รันในฝั่งเว็บเซิร์ฟเวอร์ แล้วจึงส่งผลลัพธ์ของโปรแกรมกลับไปยังหน้าเว็บเพจที่ผู้ชมเว็บเพจอีกครั้ง (เช่น โปรแกรมนับจำนวนผู้เข้าชมเว็บ) สำหรับ Red Hat Linux 9.0 จะถูกกำหนดค่าไว้ที่ /var/www/cgi-bin ซึ่งผู้ดูแลระบบจะโยกย้ายไปใช้พื้นที่อื่นได้เช่นเดียวกับ DocumentRoot อย่างไรก็ตามในปัจจุบันโปรแกรมประเภท CGI มีการใช้งานที่ลดน้อยลงไปอย่างเห็นได้ชัด ซึ่งกำลังถูกแทนที่โดยภาษาสคริปต์ประเภท HTML Embedded นั่นเอง

FTP Service เพื่อคู่หูของเว็บเซิร์ฟเวอร์

เนื่องด้วยการใช้งานเว็บเซิร์ฟเวอร์นั้นต้องการการแสดงผลเอกสารเว็บที่เป็นไฟล์ข้อความภาษา HTML ไฟล์รูปภาพ และมัลติมีเดียต่าง ๆ ผู้ที่จะปรับปรุงเนื้อหาบนเว็บเพจทั้งหลายบนเว็บเซิร์ฟเวอร์จะต้องอาศัยบริการอีกตัวหนึ่งเพื่อจัดส่งไฟล์ต่าง ๆ อัปโหลดเข้าไปเก็บภายใน DocumentRoot ของเว็บเซิร์ฟเวอร์ ซึ่งบริการดังกล่าวก็คือ FTP (File Transfer Protocol)

บริการ FTP นี้ไม่ได้รวมอยู่ในตัวเว็บเซิร์ฟเวอร์ ดังนั้นผู้ดูแลระบบจะต้องคอนฟิกเซิร์ฟเวอร์ที่ให้บริการ FTP นี้ขึ้นมาใช้งานคู่กับเว็บเซิร์ฟเวอร์เสมอ ซึ่งโปรแกรมที่ทำหน้าที่เป็น FTP Server นี้ Red Hat Linux 9.0 จะให้มาพร้อมกันแล้ว คือ โปรแกรม vsftpd (Very Secure FTP Daemon) เราจึงสามารถเปิดบริการนี้ขึ้นมาได้ด้วยคำสั่งคล้าย ๆ การเปิดบริการอาปาเช่

```
# chkconfig vsftpd on
```

```
# service vsftpd restart
```

หลังจากที่ FTP Server เริ่มทำงานแล้ว รายชื่อผู้ใช้งานในระบบทุกชื่อจะสามารถใช้บริการ FTP Server นี้ได้ทันที แต่สำหรับการเพิ่มชื่อล็อกอินของยูสเซอร์ที่จะแก้ไขปรับปรุงเว็บไซต์จำเป็นต้องกำหนดให้ยูสเซอร์นั้นเริ่มต้นเข้าไปรับส่งไฟล์ที่ตำแหน่งโคเรคทอรี DocumentRoot ของเว็บเซิร์ฟเวอร์ (ตามตัวอย่างข้างต้นคือ /itdestination.com) วิธีการสร้างแอคเคาต์ของยูสเซอร์ (สมมติชื่อ webmaster) จะมีขั้นตอนดังรูปที่ 7 หลังจากนั้นยูสเซอร์ webmaster จะสามารถตกแต่งแก้ไขเว็บไซต์ได้โดยผ่านโปรแกรม FTP Client ธรรมดาทั่วไป (เช่น WS-FTP Pro หรือ CuteFTP) จากเครื่องไคลเอ็นต์ได้ตามต้องการ

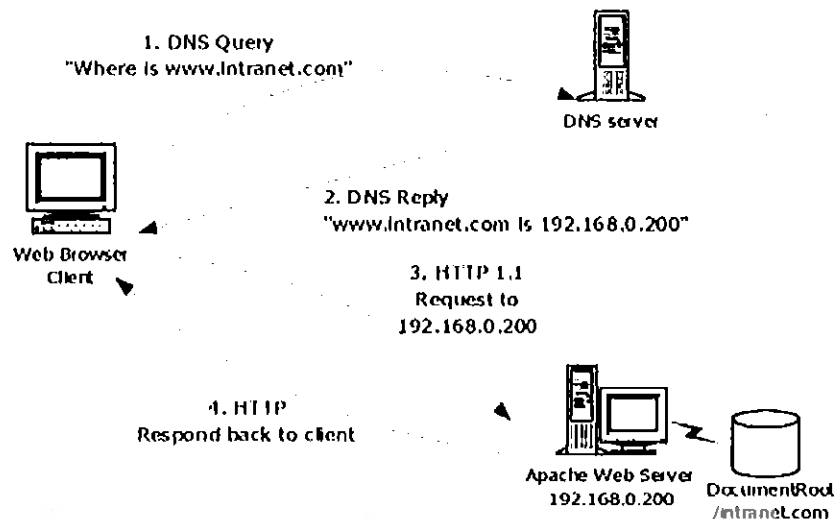


```
root@jack:~
File Edit View Terminal Go Help
[root@jack root]#
[root@jack root]# useradd -d /itdestination.com -M webmaster
[root@jack root]# passwd webmaster
Changing password for user webmaster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@jack root]# chown webmaster /itdestination.com
[root@jack root]#
```

ขั้นตอนการสร้างยูสเซอร์ webmaster เพื่อการ FTP

DNS Server อีกแรงสนับสนุนเพื่อการใช้งานที่สมบูรณ์

ในด้านการเรียกเข้าชมเว็บไซต์จากผู้ใช้ทั่วไป ระบบของเราจำเป็นต้องอาศัยระบบ Domain Name Service หรือ DNS เพื่อช่วยให้ผู้ชมสามารถเข้าถึงเว็บเซิร์ฟเวอร์ของเราได้ด้วยการเรียกด้วยชื่อของเว็บไซต์ แทนที่จะเรียกเข้ามาด้วยหมายเลขไอพี หากเป็นการสร้างเว็บเซิร์ฟเวอร์ขึ้นเพื่อใช้งานเป็นการภายในขององค์กรก็จำเป็นต้องจัดตั้ง DNS Server ขึ้นเพื่อช่วยแปลงชื่อโฮสต์ (เช่น www.intranet.com) ให้เป็นหมายเลขไอพีของโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์ของเรา ซึ่งอาจจะใช้ DNS Server ที่มีอยู่แล้วในองค์กรก็ได้ แต่ถ้ายังไม่เคยมี DNS Server มาก่อนก็สามารถคอนฟิกโปรแกรม BIND ที่มาพร้อมกับ Red Hat Linux 9.0 ให้ทำหน้าที่เป็น DNS Server ก็ได้



การทำงานร่วมกับระหว่าง DNS กับ Apache

แต่ถ้าเป็นการจัดตั้งเว็บเซิร์ฟเวอร์นี้ขึ้นเพื่อให้บริการแก่บุคคลทั่วไปในระบบเครือข่ายอินเทอร์เน็ต ก็จะต้องเชื่อมต่อเว็บเซิร์ฟเวอร์นี้เข้าสู่ระบบอินเทอร์เน็ต โดยจะต้องมีหมายเลขไอพีจริงในระบบอินเทอร์เน็ต (Real IP Address) ก่อนฟีกของระบบเครือข่ายนี้มีทางเลือกหลายทาง โดยอาจจะเชื่อมต่อเข้าสู่เครือข่ายอินเทอร์เน็ต โดยผ่านสายสื่อสารที่จัดเตรียมไว้ เช่น สายลีสต์ไลน์และเราต์เตอร์ หรืออาจจะใช้บริการจากผู้ให้บริการอินเทอร์เน็ต (Internet Data Center) โดยการนำเว็บเซิร์ฟเวอร์ของเราไปฝากไว้ที่เรียกว่า Co-Location ซึ่งเป็นอีกวิธีหนึ่งที่ประหยัดกว่าก็ได้

ต่อจากนั้นจะต้องจดทะเบียนชื่อโดเมน (Domain Name Registration) เพื่อให้ได้ชื่อโดเมนและเว็บไซต์ที่คนทั่วโลกจะเข้าถึงได้ โดยเป็นหน้าที่ของหน่วยงานที่เป็นตัวแทนจดทะเบียนชื่อโดเมนให้เราจะต้องลงทะเบียนในระบบ DNS ให้ชื่อเว็บไซต์นี้ขึ้นมาที่มีหมายเลขไอพีที่ได้เชื่อมต่อกับเว็บเซิร์ฟเวอร์ไว้ ภายหลังจากลงทะเบียนชื่อโดเมนประมาณ 2-3 วัน คนทั่วโลกก็จะเข้าถึงเว็บไซต์ของเราได้แล้ว

อาปาเซไม่ใช่อะไรแค่เว็บเซิร์ฟเวอร์

ในความเป็นจริงแล้ว สถานะของอาปาเซในปัจจุบันถูกแบ่งออกในเชิงการประยุกต์ใช้งานได้ 2 ทาง คือ การใช้งานทางตรง หรือการใช้งานโดยเน้นหนักไปในฐานะของ HTTP Server ซึ่งถูกนำไปใช้งานเป็นเว็บเซิร์ฟเวอร์โดยตรง ในส่วนนี้ยังสามารถแยกลักษณะการใช้งานออกไปได้อีกหลายทิศทางขึ้นอยู่กับลักษณะของงานและคุณลักษณะพิเศษต่าง ๆ ที่เสริมเข้าไปอีกด้วย ได้แก่

- การใช้งานเป็น Mirror Site ด้วยความสามารถจากโมดูลในกลุ่ม mod_proxy.c ทำให้เราสามารถประยุกต์ใช้อาปาเซเป็นเว็บไซต์ Mirror ได้ โดยสามารถสำเนาเนื้อหาจากเว็บไซต์ที่ได้รับการอนุญาตแล้วมาให้บริการในเซิร์ฟเวอร์ของเราได้

- ทำหน้าที่เป็น Web Redirector หรือทำหน้าที่เป็นตัวช่วยเปลี่ยนทิศทางของผู้ชมที่มาจากแหล่งต้นทางที่แตกต่างกันให้ไปสู่ URL หรือเซิร์ฟเวอร์ที่กำหนดขึ้นใหม่ได้ ซึ่งมาจากความสามารถของ โมดูล mod_rewrite.c

- การสร้างเว็บไซต์ส่วนบุคคล หรือ Personal Home Page การใช้งานแบบนี้เป็นที่นิยมมากในสถานศึกษา มหาวิทยาลัย โดยอาศัยการทำงานของ โมดูล mod_userdir.c จะช่วยให้ยูสเซอร์ทุกคนในเว็บเซิร์ฟเวอร์มีเว็บไซต์ส่วนตัวได้โดยอัตโนมัติ โดยมี URL เป็นชื่อเว็บไซต์นั้นตามด้วยเครื่องหมาย ~ และชื่อของยูสเซอร์นั้น ๆ เช่น ยูสเซอร์ gump ในเซิร์ฟเวอร์ www.tepleela.ac.th ก็จะมี URL เป็น http://www.tepleela.ac.th/~gump/ เป็นต้น ซึ่งทำให้สมาชิก นักเรียน นักศึกษา มีเว็บไซต์เป็นของตนเองที่จะใช้ฝึกหัดสร้างเว็บไซต์ และเผยแพร่ข้อมูลสู่สาธารณะได้ตามต้องการ

- การเป็น Virtual Host ลักษณะนี้เป็นที่นิยมกันมากที่สุดก็คือ การสร้างเว็บไซต์มากกว่า 1 เว็บไซต์โดยใช้เครื่องเซิร์ฟเวอร์เพียงเครื่องเดียว และใช้หมายเลขไอพีแอดเดรสเพียงหมายเลขเดียวในการอ้างถึงเว็บไซต์หลายชื่อ หรือที่เรียกว่า Name Based Virtual Host ซึ่งช่วยให้ลดค่าใช้จ่ายไปได้มาก สำหรับ Red Hat Linux 9.0 แล้วในทางเทคนิคสามารถคอนฟิกได้ทันทีในส่วนของอาปาเช่ แต่ยังคงขาดในส่วนของ FTP Server ซึ่งไม่สนับสนุนการทำ Virtual Host ในแบบ Name Based ดังนั้นจึงจำเป็นต้องนำโปรแกรม FTP Server ที่ดีชื่อว่า vsFTPd และมีคุณสมบัติด้าน Virtual Host มาใช้แทน เช่น ProFTPd หรือ PureFTPd เป็นต้น (vsFTPd สนับสนุน Virtual Host เฉพาะแบบ IP Based และ Port Based เท่านั้น)

- การเป็นเว็บเซิร์ฟเวอร์ที่สนับสนุนเทคโนโลยีเว็บอื่น ๆ Apache 1.3 และ 2.0 เป็นเพียงหนึ่งในโปรเจกของ The Apache Software Foundation เท่านั้น ยังมีโปรเจกอื่น ๆ ที่เป็นโปรเจกต่อเนื่องจากอาปาเช่อีกมากมาย เช่น Jakarta เป็นโปรเจกเสริมเพื่อทำให้อาปาเช่สนับสนุน Java Platform โดยหนึ่งในจำนวนโปรแกรมที่เป็นที่รู้จักกันเป็นอย่างดีก็คือ Tomcat 5 ซึ่งเสริมการสนับสนุน Java Servlet 2.4 และ Java Server Pages 2.0

ในทางอ้อม การประยุกต์ใช้อาปาเช่เว็บเซิร์ฟเวอร์ยังถูกนำมาใช้เพื่อเป็นส่วนประกอบในงานด้านอื่น ๆ อีก โดยอยู่ในฐานะช่องทางติดต่อระหว่างผู้ใช้กับแอปพลิเคชันต่าง ๆ ในลักษณะของ Web based User Interface ซึ่งผู้ใช้อินเทอร์เน็ตทั่วไปไม่มีความคุ้นเคยดีอยู่แล้ว อีกทั้งยังลดการบำรุงรักษาและคอนฟิกในฝั่งเครื่องไคลเอ็นท์ได้มากอีกด้วย การใช้งานในทางอ้อมที่ว่ามีได้แก่

- เป็นยูสเซอร์อินเทอร์เน็ตเฟสเข้าสู่ยูนิตตี้ อาปาเช่ถูกนำไปพัฒนาร่วมกับซอฟต์แวร์ต่าง ๆ มากมายทั้งซอฟต์แวร์เชิงพาณิชย์ และฟรีซอฟต์แวร์ เพื่อให้เป็นอินเทอร์เน็ตเฟสที่สะดวกต่อการใช้งานยิ่งขึ้น เช่น ซอฟต์แวร์บริหารจัดการโปรแกรมตรวจสอบและกำจัดไวรัส (ได้แก่ Trend Micro) ซอฟต์แวร์ช่วยการคอนฟิกและใช้งานลินุกซ์เซิร์ฟเวอร์ (ได้แก่ Webmin , Usermin)

- เป็นช่องทางแสดงผลข้อมูลระบบและเครือข่าย เนื่องจากอาปาเช่ถูกผนวกเอาไว้กับลินุกซ์เซิร์ฟเวอร์ทุกชนิดสทริบิวชั่น หรือ ถ้าเป็น โอเอสอื่น (Windows ,Mac OS X) ก็สามารติดตั้งใช้งานได้ฟรี และสามารถแสดงผลได้ทั้งตัวอักษร รูปภาพ รูปกราฟ ได้โดยตรง จึงมีการนำอาปาเช่มาใช้งานด้านการแสดงผลข้อมูลระบบ และกราฟสถิติต่าง ๆ มากมาย เช่น MRTG ใช้แสดงข้อมูลกราฟที่ได้ข้อมูลจาก Router หรือ SNMP Server โปรแกรม SARG ใช้แสดงตารางสถิติการเข้าชมเว็บไซต์ของผู้ใช้งาน Squid Proxy Server โปรแกรมประเภท Log Analyzer เป็นต้น

- ใช้เป็น Web Mail ข้อดีของการใช้งานอีเมลผ่านทางเว็บเบราว์เซอร์เป็นสิ่งที่เราต่างทราบกันเป็นอย่างดี อาปาเช่ในฐานะที่เป็น Front-End ของระบบอีเมลจึงเป็นงานอีกลักษณะหนึ่งที่เรานิยมนำมาใช้งานร่วมกับระบบ Mail Server

- เป็นอินเทอร์เน็ตของแอปพลิเคชันเฉพาะทาง มีซอฟต์แวร์เป็นจำนวนมากที่พัฒนาโดยทำงานภายใต้สภาพแวดล้อมที่เรียกว่า Web based Applications ทั้งที่เป็นการพัฒนาขึ้นเพื่อใช้งานในองค์กร โดยเฉพาะ และทั้งที่เป็นซอฟต์แวร์สำเร็จรูป เช่น โปรแกรมประเภท Groupware หรือ Web based collaboration ต่าง ๆ ระบบสนับสนุนสารสนเทศภายในองค์กร เป็นต้น



ประวัติผู้ดำเนินโครงการ



ชื่อ นายกัมพล องการ
 ภูมิลำเนา 63/2 หมู่ 5 ต. แม่จะเรา อ. แม่ระมาด จ. ตาก
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจากโรงเรียนสรรพวิทยาคม
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail: kampono56@email.nu.ac.th



ชื่อ นายมานิตย์ ประชะติ
 ภูมิลำเนา 248/4 หมู่ 15 ต. หนองกรด อ. บรรพตพิสัย
 จ. นครสวรรค์
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจาก โรงเรียนหนองกรดพิทยาคม
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail: mauidp56@email.nu.ac.th



ชื่อ นายวีรพล สุพิน
 ภูมิลำเนา 18 หมู่ 4 ต. หนองกรด อ. บรรพตพิสัย จ. นครสวรรค์
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจาก โรงเรียนหนองกรดพิทยาคม
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail: weerapong56@email.nu.ac.th