ห้องพักอัจฉริยะ
SMART ROOM

โดย

นายนวพล วงศ์เกษมสรรค์

นายภาณุพงษ์ โพธิเวชกุล

นายภานุชา ไทยภักดี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ห้องพักอัจฉริยะ
SMART ROOM

โดย

นายนวพล	วงศ์เกษมสรรค์	56010661
นายภาณุพงษ์	โพธิเวชกุล	56010928
นายภานุชา	ไทยภักดี	56010933

T149407

อาจารย์ที่ปรึกษา

ผศ.ดร. ธเนศ พัฒนธาดาพงษ์

ผศ.ดร. นภัทร สระเอี่ยม

.b 12885548
.i

ตขหมู่
.ลขทะเบียน 149407
วัน.เดือน.ปี 7 มิ.ย. 2561

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

7 / 7 / 2560

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

7 / 7 / 2560

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ปริญญานิพนธ์ปีการศึกษา 2559
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ห้องพักอัจฉริยะ
SMART ROOM

ผู้จัดทำ
1. นาย นวพล วงศ์เกษมสรรค์ 56010661
2. นาย ภาณุพงษ์ โพธิเวชกุล 56010928
3. นาย ภานุชา ไทยภักดี 56010933

..... อาจารย์ที่ปรึกษา
(ผศ.ดร.ธเนศ พัฒนธาดาพงษ์)

..... อาจารย์ที่ปรึกษา
(ผศ.ดร.นภัทร สระเอี่ยม)

กิตติกรรมประกาศ

โครงงานนี้สามารถสำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับคำแนะนำและการให้คำปรึกษาตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดียิ่ง จาก ผู้ช่วยศาสตราจารย์ ดร. นภัทร สระเอี่ยม และ ผู้ช่วยศาสตราจารย์ ดร. ธเนศ พัฒนธาดาพงษ์ คณะผู้จัดทำตระหนักถึงความตั้งใจและความทุ่มเทของอาจารย์จึงรู้สึกซาบซึ้งสำหรับคำแนะนำ และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับคณะผู้จัดทำ ขอน้อมบูชาบูรพาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ด้วยความรักและความเมตตา ตลอดจน บิดา มารดา พี่ ๆ น้อง ๆ ของคณะผู้จัดทำทุกคนที่เป็นกำลังใจและให้การสนับสนุนในทุก ๆ เรื่องเสมอมา

และที่สำคัญโครงงานฉบับนี้จะไม่สำเร็จได้ถ้าไม่มีเพื่อน ๆ ทุกคนที่ช่วยสละเวลามาแก้ปัญหาให้ทั้งที่อยู่คนละกลุ่มโครงงาน และช่วยให้กำลังใจกันและกันเสมอมา ทำให้กลุ่มของคณะผู้จัดทำสามารถจัดทำปริญญานิพนธ์นี้สำเร็จลุล่วงด้วยดี

สุดท้ายนี้หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นนั้น คณะผู้จัดทำขอน้อมรับผิดเพียงผู้เดียว และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนาโครงงานต่อไป

นายนวพล วงศ์เกษมสรรค์
นายภาณุพงษ์ โพธิเวชกุล
นายภานุชา ไทยภักดี
ผู้จัดทำ

ห้องพักอัจฉริยะ
SMART ROOM

โดย นายนวพล วงศ์เกษมสรรค์ 56010661
นายภาณุพงษ์ โพธิเวชกุล 56010928
นายภานุชา ไทยภักดี 56010933

อาจารย์ที่ปรึกษา ผศ.ดร.ธเนศ พัฒนธาดาพงษ์
ผศ.ดร.นภัทร สระเอี่ยม

บทคัดย่อ

ปัจจุบันความปลอดภัยและความสะดวกสบายถือเป็นสิ่งจำเป็นในยุคพัฒนานี้เป็นอย่างมาก เนื่องจากการรักษาความปลอดภัยในรูปแบบเดิม มีความปลอดภัยน้อยลงหรือไม่เพียงพอในยุคแห่งการพัฒนาของเทคโนโลยี ซึ่งโครงงานนี้จะเป็นการพัฒนาระบบความปลอดภัยถึง 3 ระบบด้วยกันและระบบสั่งการควบคุมด้วยเสียง โดยพัฒนาระบบรักษาความปลอดภัยโดยใช้การตรวจจับใบหน้า การสแกนนิ้ว และระบบ RFID มารวมกัน แล้วนำเอาเทคโนโลยีสั่งการด้วยเสียงมาใช้ในการควบคุมอุปกรณ์ไฟฟ้าในห้องอีกด้วย ซึ่งใช้ Raspberry Pi เป็นบอร์ดประมวลผล โดยเป็นการพัฒนาระบบด้วยภาษา Python เพื่อพัฒนาระบบรักษาความปลอดภัยและระบบควบคุมภายในห้องให้มีความหลากหลายและมีประสิทธิภาพที่ดียิ่งขึ้น

ABSTRACT

Nowadays security system and convenient is essential because the old security system has a lower safety. This project will be the development of a security system to 3 system together and control system with voice which develop a security system with face detection finger scan and RFID system to unlock instead of using the key, and the voice command technology used to control electrical device in the room. The program is written by Python language. Raspberry Pi is used to core processor for improve the security system and control system inside the room, and with even better performance.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญ (ต่อ)		IV
สารบัญรูป		V
สารบัญรูป (ต่อ)		VI
สารบัญรูป (ต่อ)		VII
สารบัญตาราง		VIII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ราสเบอรี่ พาย 3 (Raspberry Pi 3)	3
	2.2 ภาษาไพธอน	9
	2.3 OpenCV	10
	2.4 การตรวจจับใบหน้า (Face Detection)	14
	2.5 การรู้จำใบหน้า (Face Recognition)	18
	2.6 Raspberry Pi Camera Module	21
	2.7 RFID Card Reader/Detector Module Kit (RC522)	24
	2.8 Fingerprint scanner Module	26
	2.9 IR Module	31
	2.10 Speech recognition	33

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**35**
	3.1 การออกแบบ	35
	3.2 เครื่องมือที่ใช้ในการทดลอง	42
	3.3 การจัดเก็บผลการทดลอง	46
บทที่ 4	**ผลการทดลอง**	**47**
	4.1 ผลจากการอ่านข้อมูลใบหน้าที่เก็บไว้ใน Raspberry Pi	48
	4.2 ผลจากการอ่านข้อมูลลายนิ้วมือที่เก็บไว้ใน Raspberry Pi	47
	4.3 ผลจากการอ่านข้อมูลการ์ด RFID ที่เก็บไว้ใน Raspberry Pi	54
	4.4 ผลจากการแปลงเสียงพูดเป็นข้อความ	55
	4.5 ผลจากการแบ่งข้อความที่ได้รับเป็นคำ	57
	4.6 ผลจากการทดลองนำข้อความที่ได้ไปควบคุมอุปกรณ์	58
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**59**
	5.1 สรุปผล	59
	5.2 ปัญหาและอุปสรรค	59
	5.3 ข้อเสนอแนะ	59
บรรณานุกรม		**60**
ภาคผนวก ก	ชุดคำสั่งของระบบตรวจจับใบหน้า	61
ภาคผนวก ข	ชุดคำสั่งของระบบสแกนลายนิ้วมือและระบบ RFID	66
ภาคผนวก ค	ชุดคำสั่งของระบบสั่งงานด้วยเสียง	72

สารบัญรูป

รูปที่		หน้า
1.1	บล็อกไดอะแกรมภาพรวมของระบบ	2
2.1	ส่วนประกอบของ Raspberry Pi 3	5
2.2	ขา GPIO ของ Raspberry Pi 3	5
2.3	ตัวอย่างของไฟล์ Raspbian	6
2.4	โปรแกรม Win32DiskImager ลง Raspbian ให้กับ Micro SD card	7
2.5	หน้าต่างหลังจาก Boot เสร็จ	8
2.6	ภาพหน้าจอ Raspberry Pi หลังจาก reboot เสร็จแล้ว	8
2.7	แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV	14
2.8	แสดงเทคนิคการรวมภาพ	15
2.9	ตัวอย่างรูปแบบของ Haar like feature	16
2.10	กระบวนการทำงานของ Classifier	16
2.11	แสดงตัวอย่างตัวกรองสองตัวแรกใน Viola-Jones Cascade	17
2.12	ตัวอย่างตัวกรองอื่น ๆ ใน Viola-Jones Cascade	17
2.13	Raspberry Pi Camera Module	21
2.14	ลักษณะการติดตั้ง Raspberry Pi Camera Module	22
2.15	ระบบหลังการติดตั้งโมดูลกล้องเสร็จเรียบร้อยแล้ว	22
2.16	หน้าต่าง Enable Camera	23
2.17	RFID Module	24
2.18	ความถี่ RFID ในย่านต่าง ๆ	24
2.19	บล็อกการทำงานของเครื่องอ่าน	25
2.20	Fingerprint scanner module	26
2.21	USB-TTL converting board	27
2.22	การเชื่อมต่อโมดูล Fingerprint กับ USB-TTL	27
2.23	กระบวนการทำงานของระบบสแกนนิ้ว	29
2.24	หลักการตรวจสอบความถูกต้อง	30
2.25	IR Receiver/Transmitter Sensor	31
2.26	รับเสียงมาจาก Microphone	33
2.27	ประมวลเสียงจากไฟล์เสียง wav	34
2.28	แสดงตัวอย่างชุดคำสั่งเบื้องต้น	34

สารบัญรูป(ต่อ)

หน้า

รูปที่

3.1 บล็อกไดอะแกรมของระบบรักษาความปลอดภัย 36
3.2 การสั่งงานด้วยเสียงภายในห้อง 36
3.3 Flow Chart กระบวนการทำงานของระบบการตรวจจับใบหน้าเพื่อยืนยันตัวบุคคล 37
3.4 Flow Chart ของระบบรักษาความปลอดภัยแบบสแกนบัตรและแบบสแกนลายนิ้วมือเพื่อยืนยันตัวบุคคล 38
3.5 Flow Chart กระบวนการทำงานของการสั่งงานด้วยเสียง 40
3.6 Raspberry Pi 3 42
3.7 Fingerprint scanner 42
3.8 RFID Module 43
3.9 Raspberry Pi Camera Module 43
3.10 Microphone 44
3.11 Infrared 44
3.12 Transistor-powered IR transmitter schematics 45
3.13 กล่องชิ้นงาน 45
4.1 คำสั่งเปิดไฟล์ที่ใช้งาน 47
4.2 การเตรียมการทดลอง 47
4.3 ผู้ใช้งานบุคคลที่ 1 มีใบหน้าไม่ตรงกับฐานข้อมูล 48
4.4 ผู้ใช้งานบุคคลที่ 2 มีใบหน้าไม่ตรงกับฐานข้อมูล 48
4.5 ผู้ใช้งานบุคคลที่ 3 มีใบหน้าไม่ตรงกับฐานข้อมูล 49
4.6 ผู้ใช้งานบุคคลที่ 4 มีใบหน้าไม่ตรงกับฐานข้อมูล 49
4.7 ผู้ใช้งานบุคคลที่ 5 มีใบหน้าไม่ตรงกับฐานข้อมูล 50
4.8 ไฟล์ภาพฐานข้อมูล 51
4.9 ผู้ใช้งานบุคคลที่ 1 มีใบหน้าตรงกับฐานข้อมูล 51
4.10 ผู้ใช้งานบุคคลที่ 2 มีใบหน้าตรงกับฐานข้อมูล 52
4.11 ผู้ใช้งานบุคคลที่ 3 มีใบหน้าตรงกับฐานข้อมูล 52
4.12 ผู้ใช้งานบุคคลที่ 4 มีใบหน้าตรงกับฐานข้อมูล 53
4.13 ผู้ใช้งานบุคคลที่ 5 มีใบหน้าตรงกับฐานข้อมูล 53
4.14 ผลจากการสแกนลายนิ้วมือ 54

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.15 ผลจากการสแกนบัตร RFID 55
4.16 การทดลองเปลี่ยนเสียงพูดเป็นข้อความ 56
4.17 การแยกคำจากข้อความ 57
4.18 การสั่งงานอุปกรณ์ 58

สารบัญตาราง

หน้า

ตารางที่

4.1 ผลการทดลองหาความแม่นยำของการสแกนนิ้ว 54
4.2 ผลการทดลองหาความแม่นยำของการสแกนบัตร RFID 55
4.3 ผลการทดลองหาความแม่นยำของการแปลงเสียงเป็นข้อความ 57

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนยีมีบทบาทสำคัญในชีวีตประจำวัน ทำให้เกิดความสะดวกสบายในการดำเนินชีวิตมากยิ่งขึ้น ความปลอดภัยของทรัพย์สินหรือตัวบุคคลภายในห้องพักอาศัยนั้นเป็นสิ่งสำคัญอันดับแรกเพราะไม่มีผู้ใดประสงค์ให้เกิดความเสียหายขึ้นกับตนเอง เพื่อให้เข้ากับยุคสมัย ได้มีการพัฒนาอุปกรณ์รักษาความปลอดภัยต่าง ๆ เพื่อช่วยเพิ่มประสิทธิภาพของการป้องกันมากยิ่งขึ้น เช่น ลูกบิดประตูอาจมีการเปลี่ยนลักษณะ แม่กุญแจอาจใช้รหัสแทนใช้ลูกกุญแจ เป็นต้น การนำเอาเทคโนโลยีที่ทันสมัยและเกี่ยวข้องกับระบบรักษาความปลอดภัยมาช่วยในการป้องกันความเสียหายที่จะเกิดขึ้นต่อทรัพย์สินและตัวบุคคล จะทำให้ห้องพักอาศัยมีความปลอดภัยมากยิ่งขึ้น

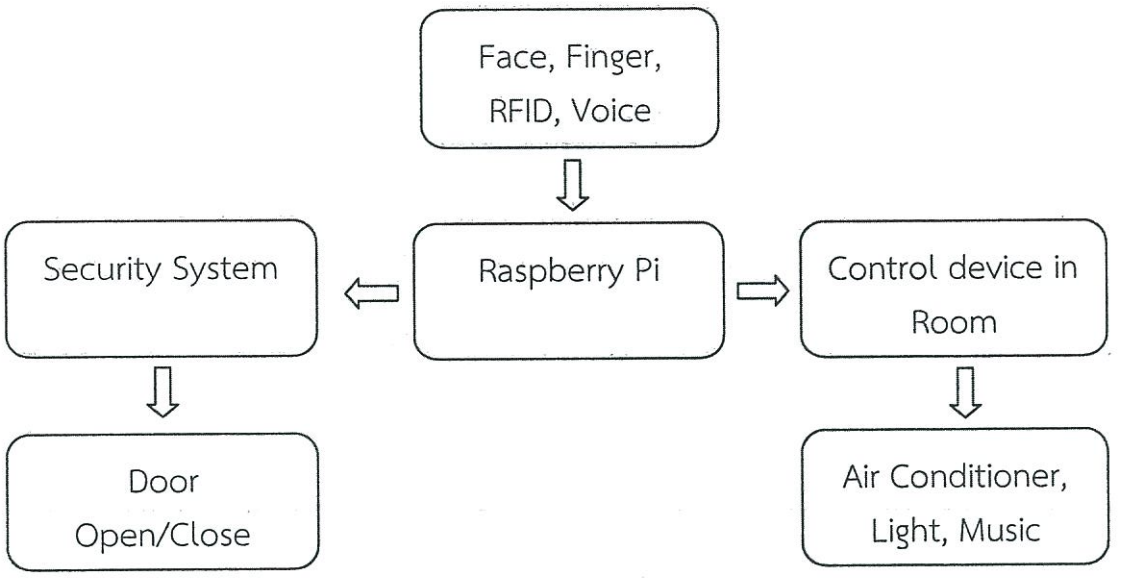
และภายในห้องได้มีการนำเอาเทคโนโลยีต่าง ๆ มาอำนวยความสะดวกให้กับผู้ที่ได้พักอาศัย สามารถควบคุมอุปกรณ์ไฟฟ้าด้วยคำสั่งเป็นเสียงพูด เช่น หลอดไฟ, เครื่องปรับอากาศ เป็นต้น

1.2 วัตถุประสงค์

1) เพื่อทำให้ระบบรักษาความปลอดภัยมีความสะดวกมากขึ้น
2) เพื่อศึกษาหลักการของการตรวจจับใบหน้าและการเปรียบเทียบใบหน้า
3) เพื่อศึกษาหลักการของโปรแกรม OpenCV ที่ใช้ในการประมวลผลภาพ
4) เพื่อศึกษาการพัฒนาระบบโดยใช้ภาษา Python
5) เพื่อศึกษาการทำงานของ Raspberry Pi และระบบปฏิบัติการ Raspbian
6) เพื่อศึกษาและพัฒนาระบบรักษาความปลอดภัยด้วยการใช้ RFID
7) เพื่อศึกษาหลักการทำงานของการสแกนลายนิ้วมือ และการเก็บค่าลายนิ้วมือ
8) ศึกษาการเขียนโปรแกรมเพื่อควบคุมการเปิด-ปิดประตู และการเก็บข้อมูลผู้ใช้งาน
9) ศึกษาหลักการทำงานรีโมท อินฟราเรด ของเครื่องปรับอากาศ

1.3 ขอบเขตของปริญญานิพนธ์

เพื่อให้เป็นห้องที่สามารถควบคุมระบบหรืออุปกรณ์เครื่องใช้ไฟฟ้าต่างๆด้วยคำสั่งเสียง และเพิ่มระบบ รักษาความปลอดภัยของห้องด้วยการใช้เทคโนโลยีรักษาความปลอดภัยเพื่อยืนยันตัวบุคคลหลายรูปแบบร่วมกัน โดยระบบทั้งหมดนี้จะประมวลผลผ่าน Raspberry Pi และมีฐานข้อมูลไว้ใช้เก็บข้อมูลหรือ เพื่อใช้เรียกดูข้อมูลได้



รูปที่ 1.1 บล็อกไดอะแกรมภาพรวมของระบบ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง "ห้องพักอัจฉริยะ" เป็นการรวบรวมและพัฒนาระบบรักษาความปลอดภัยด้วยการตรวจจับใบหน้า ระบบรักษาความปลอดภัยด้วยการสแกนลายนิ้วมือ ระบบรักษาความปลอดภัยด้วยเทคโนโลยี RFID โดยระบุตัวตนของบุคคลนั้นด้วย เพื่อใช้ในการเข้าพักอาศัยในห้อง และควบคุมการใช้งานเครื่องใช้ไฟฟ้าภายในห้องผ่านคำสั่งเสียง โดยใช้ Raspberry Pi ที่มีระบบปฏิบัติการแบบ Raspbian และพัฒนาระบบโดยใช้ภาษา Python และนำ Raspberry Pi Camera Module, Fingerprint Module และ RFID Reader Module มาเป็นอุปกรณ์ในการประมวลผลที่ Raspberry Pi ซึ่งการตรวจจับใบหน้านั้นโปรแกรมที่ใช้ในการประมวลผลภาพและระบุตัวตนนั้นคือโปรแกรม OpenCV จากนั้นเมื่อทำการระบุตัวตนได้แล้ว ก็จะทำการส่งค่าไปยังพอร์ต GPIO เพื่อทำให้พอร์ต GPIO มีการจ่ายไฟไปยังกลอนไฟฟ้าเพื่อทำการปลดล็อคประตู โดยที่หลักการของระบบสแกนนิ้วและ RFID จะมีหลักการคล้ายกัน แต่ถ้าไม่สามารถระบุตัวตนของบุคคลนั้นได้ ระบบก็จะทำการบันทึกภาพใบหน้านั้นไว้ เพื่อนำไปใช้ประโยชน์เมื่อเกิดอันตรายต่อทรัพย์สินหรือตัวบุคคลต่อไป ซึ่งทฤษฎีและหลักการทั้งหมดที่เกี่ยวข้องมีดังต่อไปนี้

2.1 ราสเบอร์รี่พาย 3 (Raspberry Pi 3)

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล์ หรือเล่นเกม อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition Video)

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian), Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน Micro SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ได้อีกด้วย โดยปริญญานิพนธ์นี้ได้ใช้บอร์ด Raspberry Pi 3 Model B

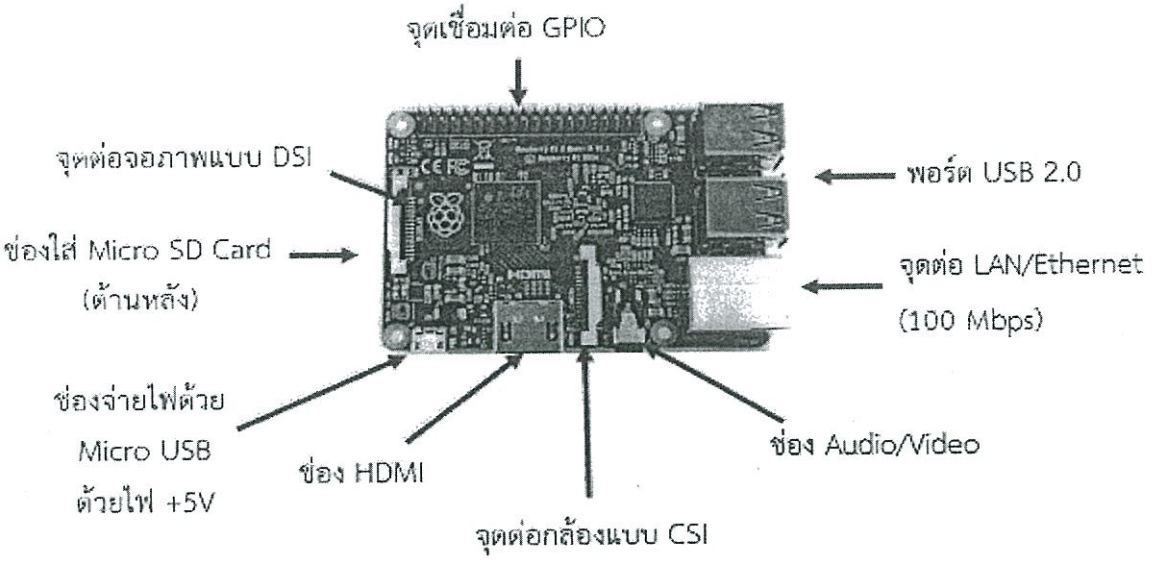
2.1.1 คุณสมบัติทางเทคนิค

- ชิปควบคุมหลัก : Broadcom BCM 2837 ซึ่งรวมหน่วยประมวลผลกลาง (CPU) หน่วยประมวลผลกราฟิก (GPU) และหน่วยความจำ SDRAM ไว้ภายในตัวเดียวกัน

- หน่วยประมวลผลกลางหรือ CPU : ARM Cortex-A53 4-core ความถี่สัญญาณนาฬิกา 1.2 GHz
- หน่วยประมวลผลกราฟิกหรือ GPU : Broadcom VideoCore IV แบบคอร์คู่ รองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
- หน่วยความจำ SDRAM : 1 GB LPDDR 2
- จุดต่อ : USB 2.0 (4 พอร์ต), HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับโทรทัศน์หรือจอแสดงผลที่มีจุดต่อแบบ HDMI, จุดต่อเอาต์พุตเสียงเป็นแจ็คหูฟัง 3.5 มิลลิเมตร, จุดต่ออีเธอร์เน็ตหรือจุดต่อระบบ LAN, จุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I^2C (Inter-Integrated Circuit), ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และซ็อกเก็ตของ Micro SD card สำหรับเสียบ Micro SD card ที่ติดตั้งระบบปฏิบัติการเรียบร้อยแล้ว
- ความต้องการไฟเลี้ยง : 5 V กระแส 2.5 A ขึ้นไป
- ขนาด : 85 มิลลิเมตร x 56 มิลลิเมตร x 17 มิลลิเมตร น้ำหนัก : 40 กรัม
- Bluetooth 4.0
- 802.11n wireless LAN

2.1.2 ส่วนประกอบของบอร์ด Raspberry Pi

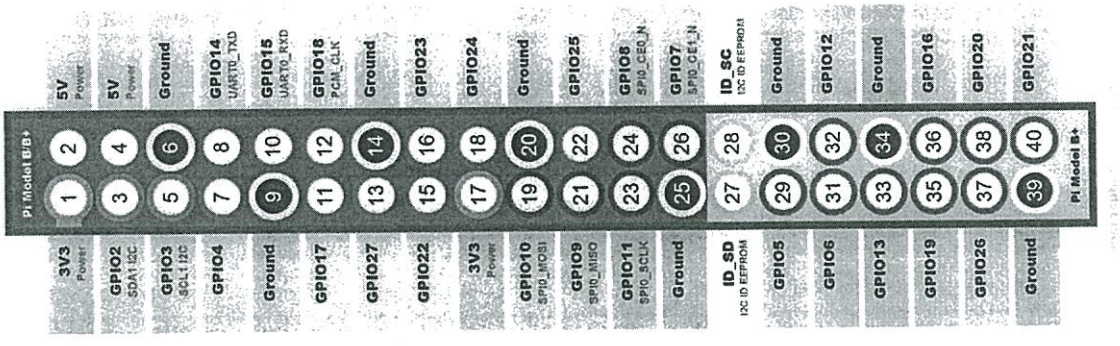
- HDMI Out (ใช้เชื่อมต่อพอร์ต HDMI เพื่อแสดงผลบนหน้าจอ)
- Ethernet Out (ใช้เชื่อมต่อหัวต่อแบบ RJ-45 เพื่อเชื่อมต่อระบบเครือข่าย LAN)
- USB 2.0 (ใช้เชื่อมต่ออุปกรณ์ USB)
- Audio Out (ใช้เชื่อมต่ออุปกรณ์เสียง)
- Camera Serial Interface (ใช้เชื่อมต่อ Raspberry Pi Camera Module)
- Micro SD Card Slot (ใช้เชื่อมต่อ Micro SD Card)
- Display Serial Interface (ใช้เชื่อมต่อจอภาพแบบ DSI หรือต่อจอภาพเพิ่มเติม)
- GPIO Headers (ใช้เชื่อมต่อ GPIO)



รูปที่ 2.1 ส่วนประกอบของ Raspberry Pi 3

2.1.3 การจัดการขา GPIO

GPIO ย่อมาจาก General Purpose Input/Output ขา Pin เหล่านี้มีไว้ใช้ในงานทั่ว ๆ ไป โดยควบคุมผ่านซอฟต์แวร์ ขา GPIO นั้นจึงสามารถใช้งานได้หลากหลายรูปแบบขึ้นอยู่กับผู้ใช้งานต้องการที่จะใช้งานอะไร เช่น ในปริญญานิพนธ์นี้จะใช้ขา GPIO ในการสั่งการกลอนไฟฟ้า



รูปที่ 2.2 ขา GPIO ของ Raspberry Pi 3

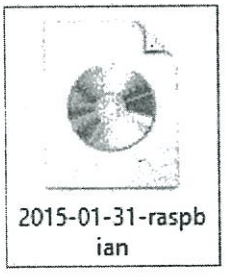
2.1.4 การเริ่มต้นการใช้งานบอร์ด Raspberry Pi

ก่อนเริ่มต้นการใช้งานบอร์ด Raspberry Pi จำเป็นที่จะต้องติดตั้งระบบปฏิบัติการให้กับบอร์ดก่อน เนื่องจากบอร์ดไม่มีหน่วยความจำแบบแฟลชเมมโมรี่มาบนบอร์ดด้วย ดังนั้นจำเป็นที่จะต้องเตรียมอุปกรณ์ต่าง ๆ ให้พร้อมเพื่อให้สามารถใช้งานบอร์ดได้ ซึ่งมีรายละเอียดอุปกรณ์ดังนี้

1. บอร์ด Raspberry Pi
2. Micro SD card ขนาด 16 GB คลาส 4 หรือเร็วกว่า
3. เมาส์และคีย์บอร์ดแบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi
4. สาย MicroUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ 5 V กระแส 2.5 A
5. สาย HDMI สำหรับต่อกับจอแสดงผล

2.1.4.1 เตรียม Software สำหรับติดตั้งระบบปฏิบัติการ Linux ลงบนบอร์ด Raspberry Pi

1. เตรียม Micro SD card ขนาดที่พอเหมาะ โดยคณะผู้จัดทำได้ใช้ Micro SD card ขนาด 16 GB
2. ดาวน์โหลดไฟล์ระบบปฏิบัติการ Raspbian ซึ่งเป็นระบบปฏิบัติการแบบ Debian ที่ถูกปรับแต่งให้ใช้สำหรับบอร์ด Raspberry Pi โดยเฉพาะ สามารถดาวน์โหลดได้จาก http://www.raspberrypi.org/downloads ปริญญานิพนธ์นี้จะใช้รุ่น Jessie.
3. เมื่อดาวน์โหลดมาแล้วจะได้ไฟล์มาดังรูป

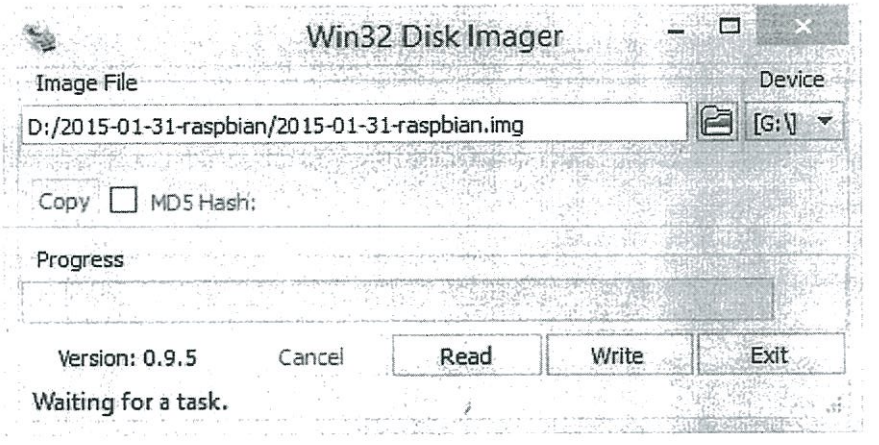


รูปที่ 2.3 ตัวอย่างของไฟล์ Raspbian

4. เข้า http://sourceforge.net/projects/win32diskimager/ เพื่อดาวน์โหลดโปรแกรม Win32DiskImager ซึ่งเป็นโปรแกรมที่ใช้ในการ

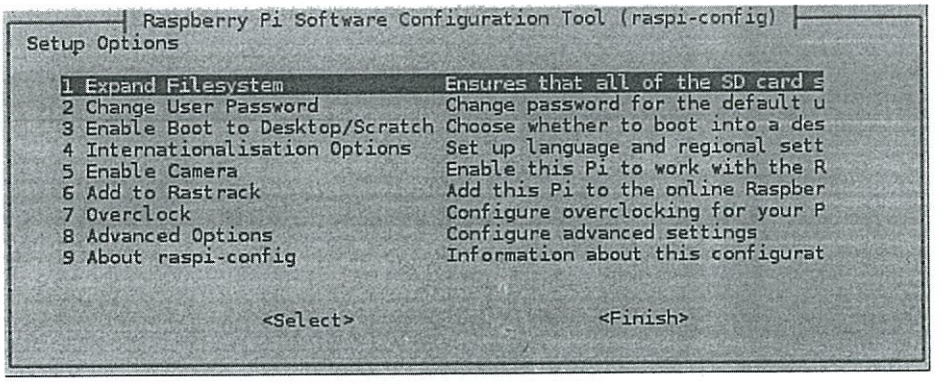
เขียนไฟล์ image ลง Micro SD card ที่เตรียมไว้ เมื่อดาวน์โหลดเสร็จให้ทำการเปิดโปรแกรม

5. ให้ Browse ไฟล์ image ระบบปฏิบัติการ Raspbian (*img) และเลือก Device ให้ถูกต้อง แล้วคลิกปุ่ม Write แสดงดังรูป เมื่อปรากฏหน้าต่างยืนยันให้คลิกปุ่ม Yes



รูปที่ 2.4 หน้าต่างโปรแกรม Win32DiskImager ลง Raspbian ให้กับ Micro SD card

6. ทำการติดตั้ง Micro SD card ที่ลง Raspbian แล้วให้กับ Raspberry Pi
7. หลังจากเสียบ Micro SD Card เรียบร้อยแล้ว ให้เสียบเมาส์ คีย์บอร์ด สายต่อจอแสดงผล HDMI จากนั้นจึงป้อนไฟเลี้ยงวงจรบอร์ดผ่านพอร์ต MicroUSB
8. หลังจากนั้นบอร์ด Raspberry Pi ก็จะเริ่มทำงาน และเริ่ม Boot ระบบ
9. หลังจากระบบปฏิบัติการ Boot เสร็จเรียบร้อย ในครั้งแรกจะปรากฏหน้าต่างแสดงดังรูป ให้เลือกเมนู 1 Expand Filesystem เพื่อขยายพื้นที่บน Micro SD Card ให้ใช้งานได้เต็มความจุ เนื่องจากการลง Raspbian ครั้งแรกจะใช้พื้นที่บน Micro SD Card เพียง 4 GB และจะไม่สามารถมองเห็นพื้นที่ส่วนที่เหลือของการ์ดได้



รูปที่ 2.5 หน้าต่างหลังจาก Boot เสร็จ

10. จากนั้นเลือก Set up ความสามารถอื่นตามที่ต้องการใช้ เช่น ในปริญญานิพนธ์นี้ต้องการใช้ควบคู่กับกล้องก็ทำการเลือกข้อ 5 ตาม รูปที่ 2.5 เพื่อเปิดใช้งานกล้อง เป็นต้น จากนั้นให้กด <Finish> แล้วทำการ Reboot
11. หลังจาก Reboot แล้วจะปรากฏหน้าจอของ Raspberry Pi ดังภาพ

รูปที่ 2.6 ภาพหน้าจอ Raspberry Pi หลังจาก reboot เสร็จแล้ว

2.2 ภาษาไพธอน

ภาษาไพธอน พัฒนาโดย Guido van Rossum ชาวดัตช์ ซึ่งมีคุณสมบัติเด่นดังต่อไปนี้

- สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือ สามารถทำงานได้ทุก ๆ CPU หลาย ๆ ระบบปฏิบัติการ เพียงแค่ผู้เขียนโปรแกรมเขียนบนแพลตฟอร์มใด ๆ แล้วนำโปรแกรมที่ได้ไปให้ทำงานต่างแพลตฟอร์มกันได้

- ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมแปลภาษา (Interpreter) โดยปกติแล้วการเขียนโปรแกรมด้วยภาษาโปรแกรมมิ่งทั่ว ๆ ไปแล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภท Open Source

- ภาษาไพธอนได้นำเอาข้อดีของภาษาโปรแกรมมิ่งอื่น ๆ เข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java และ Perl เป็นต้น

- ใช้ในการพัฒนา Web Service ได้ ปัจจุบันการพัฒนาซอฟต์แวร์ได้เน้นในการแลกเปลี่ยนข้อมูลซึ่งกันและกัน ทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน เพื่อสะดวกในการเรียกใช้ข้อมูลทำงานร่วมกันระหว่างแอพพลิเคชั่น ไม่ต้องใช้ซอฟต์แวร์อื่น ๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไป

- เป็นภาษาที่มีรูปแบบการเขียนที่เข้าใจง่าย เป็นระเบียบ ซึ่งมีคำสั่งต่าง ๆ ที่อำนวยความสะดวกอยู่ในตัว ช่วยให้การเขียนสั้นลง และสร้างตัวแปรได้ง่าย โดยไม่ต้องประกาศชนิดของตัวแปร

- มีชุดคำสั่งเสริม หรือเรียกกันว่าไลบรารี (Library) ซึ่งมีบุคคลอื่นได้เขียนเอาไว้ค่อนข้างมาก จึงสามารถนำมาใช้เพื่อเพิ่มประสิทธิภาพการทำงานได้อย่างกว้างไกลขึ้นอีก

2.3 OpenCV

OpenCV เป็นไลบรารีสำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) ซึ่งมีประสิทธิภาพสูง ใช้งานง่ายและดาวน์โหลดได้ฟรี มีความสามารถมากมาย เช่น สามารถทำภาพเบลอหา threshold หา Histogram ของภาพได้ แต่ความสามารถโดยส่วนใหญ่แล้วจะทำการค้นหาขอบของภาพ การตรวจสอบการเคลื่อนไหวและการทำ Image segmentation และสามารถจัดการกับข้อมูลแบบวิดีโอได้ด้วย

OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม การเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งจากไลบรารีเหล่านั้น ภาษาที่นิยมเขียน เช่น ภาษา C, C++ และ Python ไลบรารี OpenCV จะประกอบด้วยสองส่วนคือ data structure ที่ใช้ในการระบุโครงสร้างข้อมูลต่าง ๆ เช่น รูปภาพ เมตริกซ์ และพิกัด สำหรับอีกส่วนคือ algorithm หรือกระบวนการที่จะใช้ในการประมวลผลภาพในรูปแบบต่าง ๆ OpenCV จะประกอบด้วยไลบรารีอยู่ 4 ชุด ได้แก่ CXCORE, CV, Machine Learning และ HighGUI เช่น ตัวอย่างคำสั่งในการประกาศรูปภาพคือ IplImage, CvMat ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับภาพที่เป็นอาร์เรย์ 2 มิติเช่น การหาขอบหรือมุมของภาพ การทำฮิสโตแกรม (Histogram) เป็นต้น

2.3.1ขั้นตอนการลง OpenCV บน Raspberry Pi

1. อัพเดท repository เพื่อให้ไฟล์ของ Raspbian เป็นเวอร์ชั่นล่าสุด

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. ติดตั้งเครื่องมือ

```
$ sudo apt-get -y install build-essential pkg-config cmake
cmake-curses-gui libpng12-0 libpng12-dev libpng++-dev libpng3
libpnglite-dev zlib1g-dbg zlib1g zlib1g-dev pngtools libtiff4
libtiff4-dev libtiffxx0c2 libtiff-tools libeigen3-dev
```

```
$ sudo apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg
libjpeg-progs      ffmpeg      libavcodec-dev      libavcodec53
libavformat53      libavformat-dev      libgstreamer0.10-0-dbg
libgstreamer0.10-0      libgstreamer0.10-dev      libxine1-ffmpeg
libxine-dev      libxine1-bin      libunicap2      libunicap2-dev      swig
libv4l-0      libv4l-dev      python-numpy      libpython2.7      python-dev
python2.7-dev libgtk2.0-dev
```

3. ดาวโหลดและแตกไฟล์ OpenCV

```
$ wget -o opencv-2.4.10.zip
  http://sourceforge.net/projects/
  opencvlibrary/files/opencv-unix/2.4.10/
  opencv-2.4.10.zip/download
```

```
$ unzip opencv-2.4.10.zip
$ cd opencv-2.4.10
$ mkdir release
$ cd release
$ ccmake ../
```

จากนั้นกด “c” เพื่อปรับค่า ดังนี้

```
ANT_EXECUTABLE                  ANT_EXECUTABLE-NOTFOUND
BUILD_DOCS                      ON
BUILD_EXAMPLES                  ON
BUILD_JASPER                    ON
BUILD_JPEG                      ON
BUILD_OPENEXR                   ON
BUILD_PACKAGE                   ON
BUILD_PERF_TESTS                ON
BUILD_PNG                       ON
BUILD_SHARED_LIBS               ON
BUILD_TBB                       OFF
BUILD_TESTS                     ON
BUILD_TIFF                      ON
BUILD_WITH_DEBUG_INFO           ON
BUILD_ZLIB                      ON
BUILD_opencv_apps               ON
BUILD_opencv_calib3d            ON
BUILD_opencv_contrib            ON
BUILD_opencv_core               ON
BUILD_opencv_features2d         ON
BUILD_opencv_flann              ON
BUILD_opencv_gpu                ON
BUILD_opencv_highgui            ON
BUILD_opencv_imgproc            ON
BUILD_opencv_legacy             ON
BUILD_opencv_ml                 ON
BUILD_opencv_nonfree            ON
BUILD_opencv_objdetect          ON
BUILD_opencv_ocl                ON
BUILD_opencv_photo              ON
BUILD_opencv_python             ON
BUILD_opencv_stitching          ON
BUILD_opencv_superres           ON
BUILD_opencv_ts                 ON
BUILD_opencv_video              ON
BUILD_opencv_videostab          ON
BUILD_opencv_world              OFF
CLAMDBLAS_INCLUDE_DIR           CLAMDBLAS_INCLUDE_DIR-
NOTFOUND
CLAMDBLAS_ROOT_DIR              CLAMDBLAS_ROOT_DIR-NOTFOUND
```

```
CLAMDFFT_INCLUDE_DIR          CLAMDFFT_INCLUDE_DIR-NOTFOUND
CLAMDFFT_ROOT_DIR             CLAMDFFT_ROOT_DIR-NOTFOUND
CMAKE_BUILD_TYPE                    Release
CMAKE_CONFIGURATION_TYPES           Debug;Release
CMAKE_INSTALL_PREFIX                /usr/local
CMAKE_VERBOSE                             OFF
CUDA_BUILD_CUBIN                          OFF
CUDA_BUILD_EMULATION                      OFF
CUDA_HOST_COMPILER                  /usr/bin/gcc
CUDA_SDK_ROOT_DIR             CUDA_SDK_ROOT_DIR-NOTFOUND
CUDA_SEPARABLE_COMPILATION                OFF
CUDA_TOOLKIT_ROOT_DIR         CUDA_TOOLKIT_ROOT_DIR-
NOTFOUND
CUDA_VERBOSE_BUILD                        OFF
EIGEN_INCLUDE_PATH                  /usr/include/eigen3
ENABLE_NEON                               OFF
ENABLE_NOISY_WARNINGS                     OFF
ENABLE_OMIT_FRAME_POINTER                 ON
ENABLE_PRECOMPILED_HEADERS                ON
ENABLE_PROFILING                          OFF
ENABLE_SOLUTION_FOLDERS                   OFF
ENABLE_VFPV3                              OFF
EXECUTABLE_OUTPUT_PATH /home/pi/opencv-
2.4.10/release/bin
GIGEAPI_INCLUDE_PATH          GIGEAPI_INCLUDE_PATH-NOTFOUND
GIGEAPI_LIBRARIES             GIGEAPI_LIBRARIES-NOTFOUND
INSTALL_CREATE_DISTRIB                    OFF
INSTALL_C_EXAMPLES                        OFF
INSTALL_PYTHON_EXAMPLES                   OFF
INSTALL_TO_MANGLED_PATHS                  OFF
OPENCV_CONFIG_FILE_INCLUDE_DIR
     /home/pi/opencv/opencv-2.4.10/release
OPENCV_EXTRA_MODULES_PATH
OPENCV_WARNINGS_ARE_ERRORS                OFF
OPENEXR_INCLUDE_PATH          OPENEXR_INCLUDE_PATH-NOTFOUND
PVAPI_INCLUDE_PATH            PVAPI_INCLUDE_PATH-NOTFOUND
PYTHON_NUMPY_INCLUDE_DIR
     /usr/lib/pymodules/python2.7/numpy/core/include
PYTHON_PACKAGES_PATH          lib/python2.7/dist-packages
SPHINX_BUILD                  SPHINX_BUILD-NOTFOUND
WITH_1394                                 OFF
WITH_CUBLAS                               OFF
WITH_CUDA                                 OFF
WITH_CUFFT                                OFF
WITH_EIGEN                                ON
WITH_FFMPEG                               ON
WITH_GIGEAPI                              OFF
WITH_GSTREAMER                            ON
WITH_GTK                                  ON
```

```
WITH_JASPER                    ON
WITH_JPEG                      ON
WITH_LIBV4L                    ON
WITH_NVCUVID                   OFF
WITH_OPENCL                    ON
WITH_OPENCLAMDBLAS             ON
WITH_OPENCLAMDFFT              ON
WITH_OPENEXR                   ON
WITH_OPENGL                    ON
WITH_OPENMP                    OFF
WITH_OPENNI                    OFF
WITH_PNG                       ON
WITH_PVAPI                     ON
WITH_QT                        OFF
WITH_TBB                       OFF
WITH_TIFF                      ON
WITH_UNICAP                    OFF
WITH_V4L                       ON
WITH_XIMEA                     OFF
WITH_XINE                      OFF
```

กด “c” อีกครั้งเพื่อใช้การตั้งค่า

กด “g” เพื่อสร้าง Makefile

4. ทำการติดตั้ง OpenCV โดยใช้ Makefile

```
$ make -j4
$ sudo make install
$ sudo ldconfig
```

5. ทำการตรวจสอบ OpenCV ว่าลงสำเร็จหรือไม่

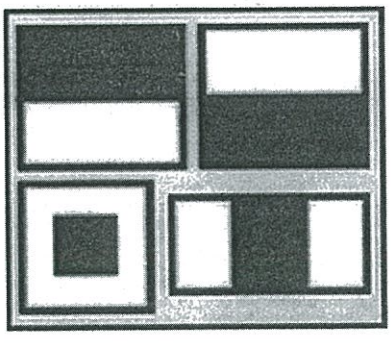
```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__'2.4.10'
```

2.4 การตรวจจับใบหน้า (Face Detection)

การตรวจจับใบหน้าคือ กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นจะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไป เพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธี ซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมาก

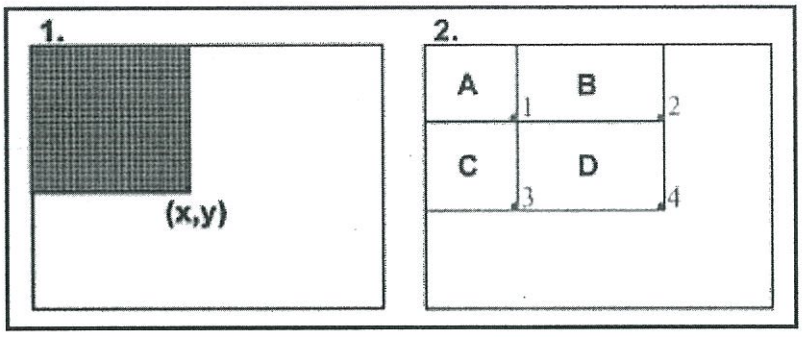
วิธีการที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตราความถูกต้องในการตรวจหาสูง ซึ่ง Paul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์ในปี ค.ศ. 2001 โดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอวิธีการแทนรูปภาพที่เรียกว่า "Integral Image" ซึ่งช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้น และได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features (features ที่ให้ classifier ที่มีประสิทธิภาพสูงสุด) นอกจากนี้ยังได้อธิบายถึงการรวม classifier แบบ cascade ซึ่งช่วยให้ส่วนพื้นหลังของภาพถูกปฏิเสธได้เร็วและเน้นการคำนวณไปที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการมีอยู่ของลักษณะเด่นของ Haar ถูกกำหนดโดยการลบค่าเฉลี่ยบริเวณจุดภาพมืดออกจากค่าเฉลี่ยบริเวณจุดภาพสว่าง ถ้าผลลัพธ์ที่ได้มีค่ามากกว่าค่าขีดแบ่ง แสดงว่ามีลักษณะเด่นของ Haar อยู่



รูปที่ 2.7 แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV

การตรวจสอบหาลักษณะเด่นของ Haar ในแต่ละส่วนของภาพอย่างมีประสิทธิภาพนั้น Viola และ Jones ใช้เทคนิคที่เรียกว่าการรวมภาพ (Integral Image) โดยการรวมจุดภาพเล็ก ๆ เข้าด้วยกัน จากรูปที่ 2.8 การจะหาค่าของรูปสี่เหลี่ยม D = G (x4,y4) - G (x2,y2) - G (x3,y3) + G (x1,y1)



(a) (b)

รูปที่ 2.8 แสดงเทคนิคการรวมภาพ (a) หลังจากที่รวมภาพแล้ว จุดภาพที่ตำแหน่ง (x,y) จะรวมค่าของทุกจุดภาพในสี่เหลี่ยมสีดำ, (b) ผลรวมค่าของทุกจุดภาพในสี่เหลี่ยม

$$D = G(x4,y4) - G(x2,y2) - G(x3,y3) + G(x1,y1)$$

ในการเลือกลักษณะเด่นของ Haar และการตั้งระดับค่าขีดแบ่ง วิธีการของ Viola-Jones ใช้ machine-learning method ที่เรียกว่า "AdaBoost" ซึ่งรวมตัวกรองจำนวนมาก (weak classifier) เข้าด้วยกันเพื่อสร้าง strong classifier ที่นำคำตอบที่ weak classifier คิดว่าภาพในส่วนนั้นเป็นใบหน้ามารวมกันเป็นคำตอบสุดท้ายว่าภาพส่วนใดบ้างเป็นภาพใบหน้าซึ่ง AdaBoost จะทำการเลือกชุดของ weak classifier ในการรวมและกำหนดน้ำหนักของแต่ละตัว การรวมกันของน้ำหนักก็คือ strong classifier นั่นเอง

2.4.1 การเรียนรู้แบบ AdaBoost และ Cascade Classifier

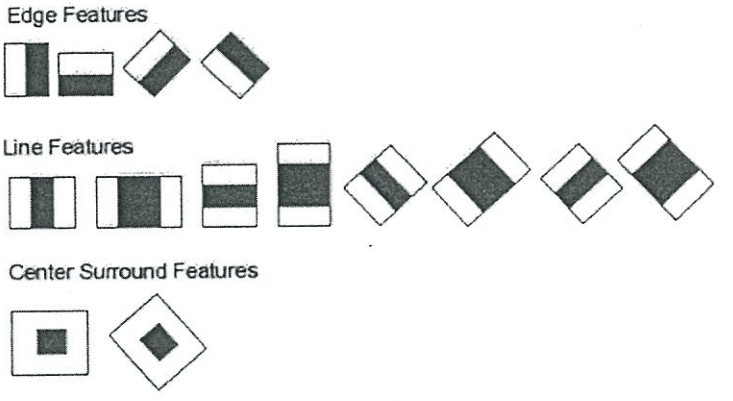
ขั้นตอนวิธีการเรียนรู้แบบ AdaBoost (AdaBoost learning algorithm) เป็นขั้นตอนวิธีการเรียนรู้เพื่อค้นหาค่าของกลุ่มพิกเซลที่มีลักษณะใกล้เคียงกับภาพใบหน้า โดยที่ภาพ positive คือ ภาพใบหน้าที่ต้องการตรวจจับ ส่วนภาพ negative คือ ภาพทั่ว ๆ ไปที่ไม่ใช่ใบหน้า ซึ่งการจำแนกกลุ่มของพิกเซลจะทำภายในส่วนย่อย (sub window) ของภาพโดยใช้ weak classifier ของ feature ที่ j (h_j)

$$h_j(x) = \begin{cases} -1 & ;if\ f_j(x) \leq 0 \\ 1 & ;\ otherwise \end{cases} \quad (2.1)$$

เมื่อ $f(x)$ คือค่าความแตกต่างของผลรวมพิกเซลในพื้นที่สีขาว และสีดำของ Haar-like feature มีค่าตามสมการที่ 2.2

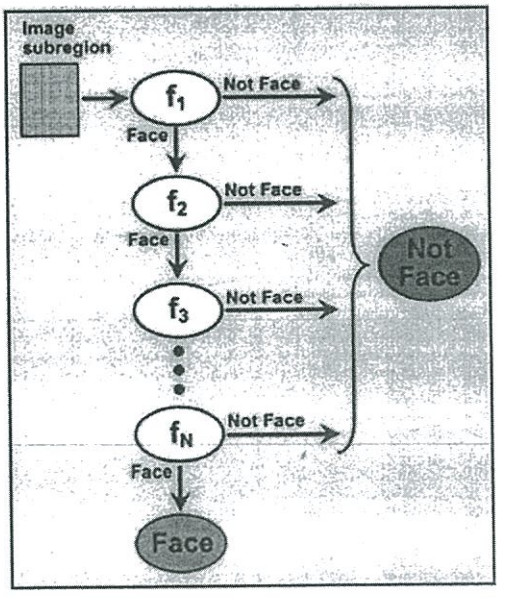
$$f(x) = Sum_{Black\ rectangle} - Sum_{White\ rectangle} \quad (2.2)$$

โดยที่ x คือภาพตัวอย่าง และ Sum คือผลรวมของภาพอินทิกรัลในบริเวณพื้นที่ขาวและดำของ x

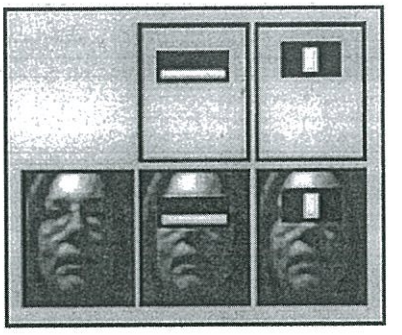


รูปที่ 2.9 ตัวอย่างรูปแบบของ Haar like feature

Cascade Classifier เป็นวิธีการจำแนกรูปร่างที่ต้องการ โดยนำการจำแนกดังที่กล่าวมาแล้วมาทำซ้ำหลาย ๆ รอบ (stage) ซึ่งในแต่ละรอบก็จะตัดพื้นที่ที่เป็นภาพ negative ออกไปในทุกรอบที่พบ เมื่อจบกระบวนการแล้วจำนวนของ sub window ที่เป็น negative จะลดลงจนได้ใบหน้าที่ต้องการ และในการจำแนกเพื่อหารูปร่างที่ต้องการนี้จะทำกับภาพอินทิกรัล ซึ่งขั้นตอนของ Cascade Classifier แสดงได้ดังรูปที่ 2.10

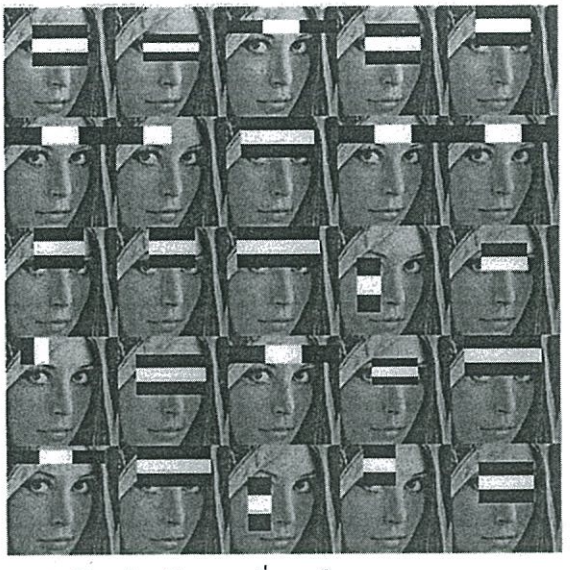


รูปที่ 2.10 กระบวนการทำงานของ Classifier



รูปที่ 2.11 แสดงตัวอย่างตัวกรองสองตัวแรกใน Viola-Jones Cascade

จากรูปที่ 2.11 จะพบว่าตัวกรองแรกใช้หลักการที่ว่าบริเวณแก้มสว่างกว่าบริเวณดวงตา และตัวกรองตัวที่สองใช้หลักการที่ว่าบริเวณสันจมูกจะสว่างกว่าบริเวณดวงตา



รูปที่ 2.12 ตัวอย่างตัวกรองอื่น ๆ ใน Viola-Jones Cascade

หลักการของอัลกอริทึมค้นหาหน้าของ Viola-Jones คือการใช้ตัวตรวจหาสแกนหลาย ๆ ครั้งบนภาพเดิมด้วยขนาดที่แตกต่างกัน ถึงแม้ว่าจะมีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ได้โดยใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจว่าบริเวณใด ๆ ไม่ใช่ใบหน้านั้น ทำได้เร็วกว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded Classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.12 ซึ่ง

149407

เมื่อ sub-window ถูกจัดประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็นมีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายิ่งมีจำนวนชั้นของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยิ่งมีมากขึ้น

2.5 การรู้จำใบหน้า (Face Recognition)

ระบบรู้จำใบหน้า ถือว่าเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคลโดยใช้คุณลักษณะจำเพาะทางสรีระ (Biometric) ซึ่งระบบจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิทัลหรือภาพจากกล้องวิดีโอของบุคคลที่สนใจกับฐานข้อมูลใบหน้าที่มีอยู่ และเมื่อเปรียบเทียบเสร็จก็จะแสดงผลใบหน้าที่อยู่ในฐานข้อมูลที่มีใบหน้าเหมือนกับภาพที่นำมาเปรียบเทียบออกมา ระบบรู้จำใบหน้านั้นได้ถูกพัฒนามาอย่างต่อเนื่องเป็นเวลานานมากแล้ว เนื่องจากเป็นระบบที่ได้รับความสนใจมากจากนักวิชาการหลายสาขาวิชา จึงทำให้ระบบรู้จำใบหน้ามีผู้คนสนใจศึกษาและพัฒนากันอย่างมากมาย ทำให้มีการพัฒนาอัลกอริทึมในการทำงานของระบบออกมามากมายหลายรูปแบบแตกต่างกันไป อันเนื่องมาจากปัจจัยด้านองค์ความรู้และเทคโนโลยีของอุปกรณ์ต่าง ๆ ที่พัฒนาขึ้นให้มีความเหมาะสมที่จะนำมาใช้ในระบบ จึงทำให้ต้องออกแบบอัลกอริทึมใหม่ให้เหมาะสมกับอุปกรณ์ใหม่ ๆ ด้วย ในปัจจุบันระบบรู้จำใบหน้าได้มีการพัฒนาไปอย่างมาก ทำให้ระบบมีความน่าเชื่อถือมากขึ้น จนมีการนำระบบรู้จำใบหน้ามาใช้ประโยชน์กันอย่างแพร่หลายในต่างประเทศ หลาย ๆ ประเทศได้มีการนำระบบรู้จำใบหน้ามาติดตั้งในสนามบินเพื่อป้องกันคนร้ายหนีเข้าออกนอกประเทศ และใช้สำหรับการยืนยันตัวคนร้ายในคดีต่าง ๆ ด้วย

2.5.1 หลักการทำงานของระบบรู้จำใบหน้า

ระบบรู้จำใบหน้าถูกออกแบบมาให้ทำการเปรียบเทียบใบหน้าบุคคลที่สนใจกับฐานข้อมูลใบหน้าที่มีอยู่ โดยระเบียบวิธีการที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจแตกต่างกันไป แล้วแต่การออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมีระเบียบวิธีการในการทำงานในขั้นตอนการสร้างอย่างไร แต่ขั้นตอนการทำโดยรวมของระบบก็ยังคงเหมือนกันอยู่ ตัวอย่างของระเบียบวิธีการรู้จำใบหน้าได้แก่ PCA หรือ Principal Component Analysis หรือเรียกว่า การวิเคราะห์องค์ประกอบหลัก คือเทคนิคในการผสมลักษณะเด่นในเวคเตอร์นำเข้าเพื่อสร้างเวคเตอร์ใหม่ที่อยู่ในปริภูมิ (Subspace) ที่มีมิติน้อยกว่าเวคเตอร์เดิมโดยการผสมที่ใช้นั้นจะเป็นการผสมเชิงเส้นหรือ Linear Combination นั่นคือการเอาลักษณะเด่นมาคูณค่าคงที่แล้วจึงบวกกัน

ปริญญานิพนธ์นี้ได้ใช้หลักการ Fisherfaces ซึ่งจะเป็นวิธีการที่ปรับปรุงประสิทธิภาพของเทคนิคใบหน้าไอเกน (Eigenfaces) โดยใช้วิธี class-specific transformation matrix ดังนั้นจึงไม่จำเป็นต้องจับภาพให้ชัดเจนเหมือนวิธี Eigenfaces การวิเคราะห์ค่าความต่างของใบหน้าจะถูกนำมาใช้แทนการวิเคราะห์คุณลักษณะของใบหน้าแทน เพื่อระบุความแตกต่างระหว่างตัวบุคคล ที่สำคัญคือการทำงานแบบ Fisherfaces มีปัจจัยหลักที่ขึ้นอยู่กับข้อมูลอินพุตอีกด้วย ในทางปฏิบัติแล้วการใช้งาน Fisherfaces กับภาพในฐานข้อมูลมีลักษณะองค์ประกอบของแสงที่พอเหมาะก็จะทำให้สามารถระบุตัวตนได้อย่างถูกต้อง เมื่อทดลองสแกนใบหน้าที่อาจมีองค์ประกอบของแสงที่ไม่เพียงพอ จะทำให้ผลลัพธ์ที่แสดงออกมานั้นอาจจะไม่สามารถระบุตัวตนได้

2.5.1.1 Fisherfaces

คือ การวิเคราะห์จำแนกประเภทเชิงเส้น (Linear Discriminate Analysis : LDA) มีจุดประสงค์เพื่อจำแนกวัตถุต่าง ๆ LDA จะนิยมใช้ในการจำแนกข้อมูลและการลดมิติ LDA รู้จักกันในอีกชื่อหนึ่งคือ Fisher Linear Discriminate (FLD) เป้าหมายของการทำ LDA คือการเพิ่มปริมาณการกระจายระหว่างกลุ่มให้มากที่สุด และลดปริมาณการกระจายภายในกลุ่มให้ได้น้อยที่สุด โดย LDA แบ่งเป็นขั้นตอนการฝึกสอน และขั้นตอนการทดสอบเช่นเดียวกับ Eigenfaces

1. การกระจายตัวภายในกลุ่ม (within-class scatter matrix) แทนโดย S_W สามารถหาได้จากสมการ 2.3

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (\Gamma_i^j - \mu_j)(\Gamma_i^j - \mu_j)^T \tag{2.3}$$

เมื่อ C คือ จำนวนกลุ่ม

N_j คือ จำนวน Sample ในกลุ่มที่ j

Γ_i^j คือ Sample ที่ i ของกลุ่มที่ j

μ_j คือ Mean ของกลุ่มที่ j

2. การกระจายตัวระหว่างกลุ่ม (between-class scatter matrix) แทนโดย S_B สามารถหาได้จากสมการ 2.4

$$S_B = \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T \tag{2.4}$$

เมื่อ μ คือ Mean รวมของกลุ่มทั้งหมด

3. จากนั้นสามารถหาค่า eigenvectors ของ S_W, S_B จากเมตริกซ์ W จากสมการ 2.5

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{2.5}$$

2.5.1.2 การคำนวณหาค่าระยะห่าง Euclidean distance

คำนวณหาค่าระยะห่างระหว่างค่าสัมประสิทธิ์ใบหน้าที่นำมาทดสอบกับค่าสัมประสิทธิ์ภาพใบหน้าในฐานข้อมูลด้วยวิธีหาค่าระยะห่าง Euclidean ดังสมการที่ 2.6

$$\varepsilon_k = \|\Omega_p - \Omega_k\| = \sqrt{\sum_{i-1}^{k}(\omega_i - \omega_{ki})^2} \tag{2.6}$$

โดยที่ 1≤ k ≤ M

เมื่อ ε_k = ค่าระยะห่าง Euclidean

Ω_p = ใบหน้าไอเกนที่ k

Ω_k = ใบหน้าทดสอบ

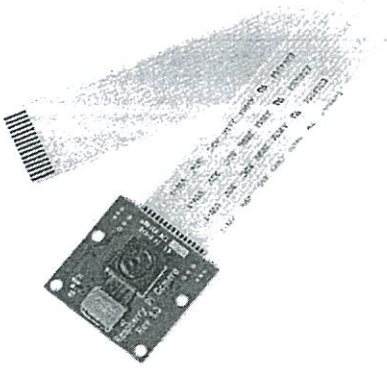
ω_{ki} = สัมประสิทธิ์แต่ละตัวใน Ω_k

ω_i = สัมประสิทธิ์แต่ละตัวใน Ω_p

M = จำนวนรูปภาพที่นำมาทดสอบ

จากสมการที่ 2.6 คือสมการหาค่าระยะห่าง Euclidean ระหว่างใบหน้าในฐานข้อมูลและใบหน้าอินพุต จะได้ค่าระยะห่าง Euclidean ทั้งหมด M ค่า เป็นตัวแทนแต่ละภาพในฐานข้อมูล โดยวิธีการ Fisherfaces จะนำค่าระยะห่าง Euclidean ที่น้อยที่สุดมาตัดสินว่ารูปที่เป็นอินพุตนั้น มีความเหมือนกับภาพใดในฐานข้อมูลมากที่สุด

2.6 Raspberry Pi Camera Module



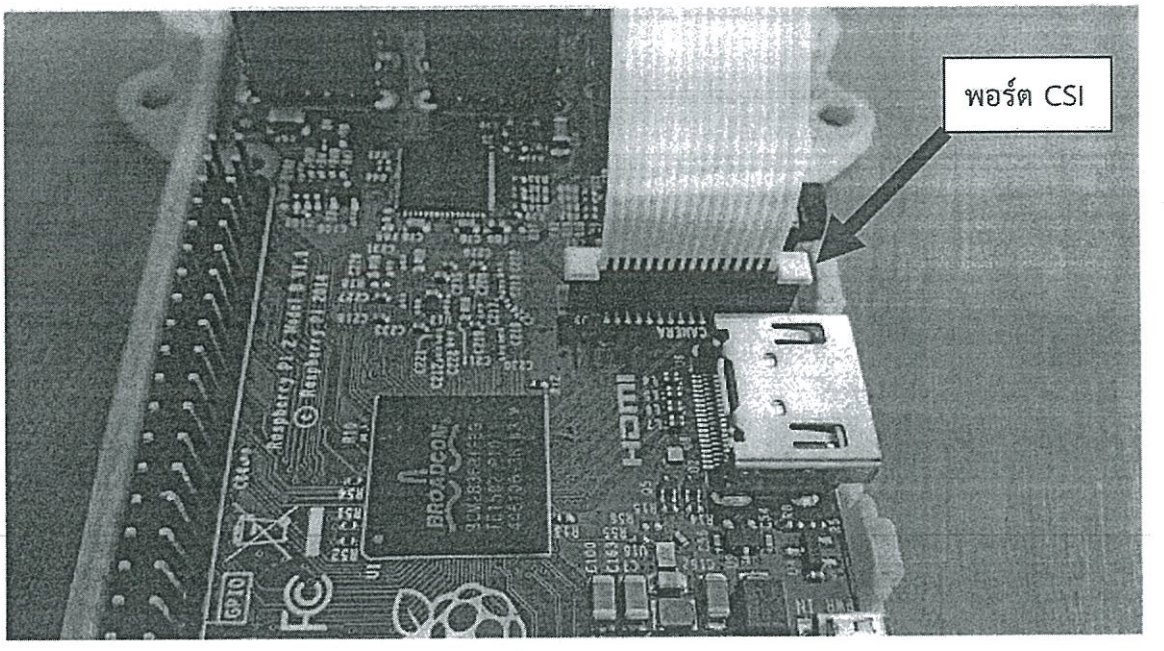
รูปที่ 2.13 Raspberry Pi Camera Module

Raspberry Pi Camera Module เป็นโมดูลกล้องที่ออกแบบมาเพื่อใช้งานร่วมกับบอร์ด Raspberry Pi โดยสามารถเชื่อมต่อกับพอร์ต CSI บนบอร์ดได้ทันที

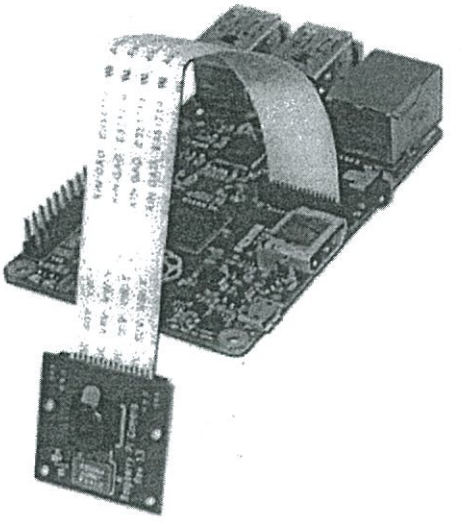
2.6.1 คุณสมบัติทางเทคนิค

- ขนาดบอร์ด 25 มิลลิเมตร x 20 มิลลิเมตร x 9 มิลลิเมตร
- ความละเอียดของกล้อง 5 MP (2592 พิกเซล x 1944 พิกเซล)
- ใช้เซ็นเซอร์ Omnivision OV5647 โดยการโฟกัสเป็นแบบคงที่
- สามารถถ่ายวิดีโอความละเอียด 1080p30, 720p60 และ 640x480p60/90

2.6.2 การติดตั้งโมดูลกล้องเข้ากับ Raspberry Pi และการใช้งานเบื้องต้น



รูปที่ 2.14 ลักษณะการติดตั้ง Raspberry Pi Camera Module

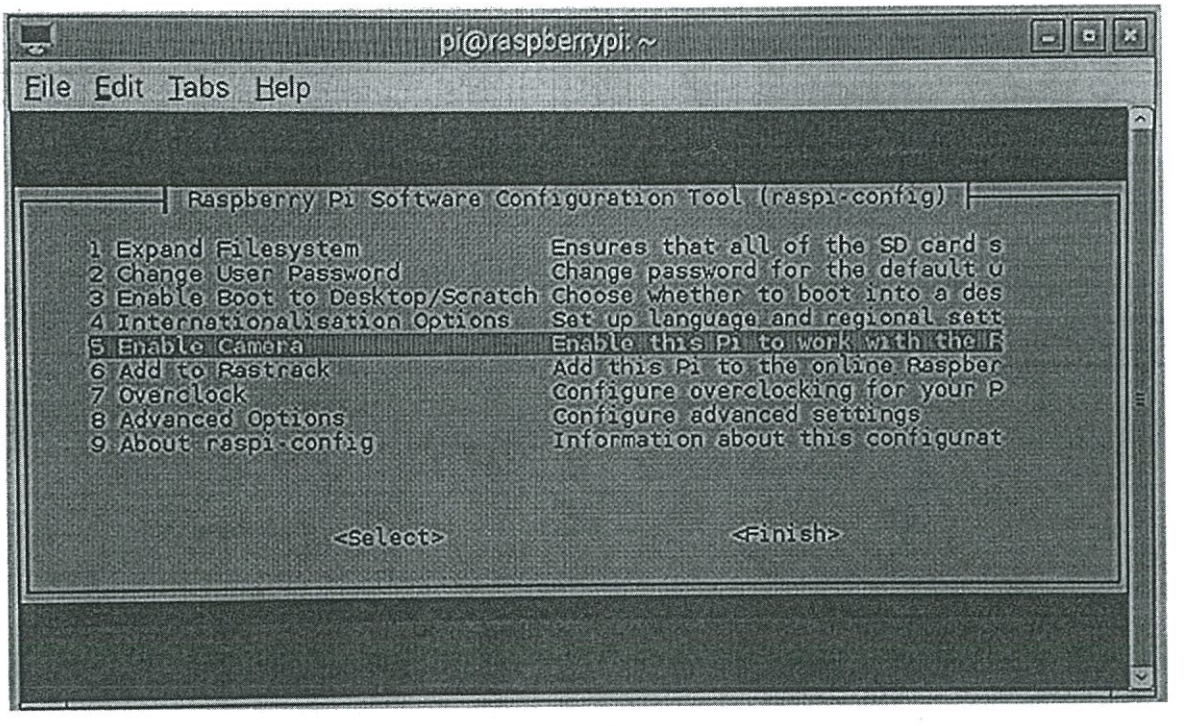


รูปที่ 2.15 ระบบหลังการติดตั้งโมดูลกล้องเสร็จเรียบร้อยแล้ว

1. เมื่อติดตั้งเสร็จแล้วให้ทำการเชื่อมต่อ Raspberry Pi กับอินเทอร์เน็ต แล้วทำการอัพเดทไฟล์ของระบบปฏิบัติการเป็นเวอร์ชั่นล่าสุดก่อนโดยใช้คำสั่ง

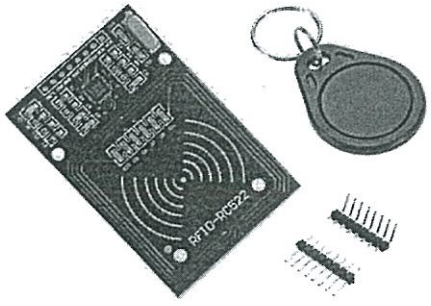
```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. ทำการ Enable Camera ในหน้า config
3. ทดสอบการใช้งานโดยการพิมพ์คำสั่ง `raspistill -o test.jpg` จะเป็นการถ่ายรูป แล้วเซฟรูปในชื่อของ test.jpg
4. ทดสอบการใช้งานโดยการพิมพ์คำสั่ง `raspivid -o test.h264` จะเป็นการถ่ายวิดีโอแล้วบันทึกเก็บไว้ 5 วินาทีตามค่า default



รูปที่ 2.16 หน้าต่าง Enable Camera

2.7 RFID Card Reader/Detector Module Kit (RC522)

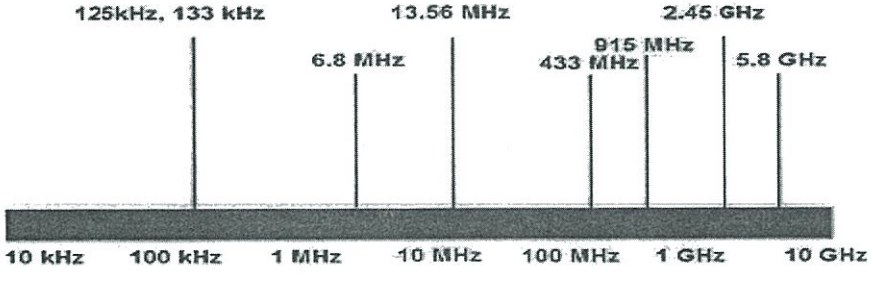


รูปที่ 2.17 RFID Module

การใช้งาน RFID Module สามารถใช้ร่วมกับ Raspberry Pi ได้เป็นอย่างดี เพราะมี library ที่ทำมาสำหรับ Raspberry Pi ไว้เรียบร้อย สามารถอ่านข้อมูลจาก Tag ได้

RFID ที่นิยมใช้กันมี 3 ย่านความถี่ แต่ละความถี่ก็มีข้อดีข้อเสียแตกต่างกัน จึงต้องเลือกใช้ให้เหมาะกับงาน

- ความถี่ต่ำ หรือ LF ความถี่ 125 KHz มีระยะการอ่านที่ใกล้ ส่งข้อมูลได้ช้า แต่ทะลุทะลวงสิ่งกีดขวางได้ดี
- ความถี่สูง หรือ HF ความถี่ 13.5 MHz มีระยะการอ่านที่ไกลกว่า LF ส่งข้อมูลได้ไกลกว่า สามารถทะลุทะลวงสิ่งกีดขวางได้ดี ความถี่นี้จึงเป็นที่นิยมใช้งานกันมากที่สุด
- ความถี่สูงยิ่ง หรือ UHF ความถี่ในช่วง 860 ถึง 960 MHz ส่งข้อมูลได้รวดเร็วและได้ไกลหลายเมตร ข้อด้อยคือทะลุทะลวงสิ่งกีดขวางได้ไม่ค่อยดี



รูปที่ 2.18 ความถี่ RFID ในย่านต่าง ๆ

การเลือกใช้ Tag และเครื่องอ่าน จะต้องเป็นความถี่เดียวกันถึงจะสามารถอ่านข้อมูลร่วมกันได้ใน RFID Module (RC522) นี้เลือกใช้ RFID ย่านความถี่ HF 13.5 MHz ซึ่งเป็นที่นิยมใช้ในงานทั่วไปมากที่สุด มีราคาไม่แพงมีประสิทธิภาพการทำงานที่ดี โดยระบบ RFID เป็นระบบที่ใช้ระบุตัวตน โดยใช้ Tag เป็นตัวบอก ID ซึ่งระบบ RFID ประกอบไปด้วย

1. Tag
2. Reader Module
3. Software ที่ใช้ควบคุมการทำงาน

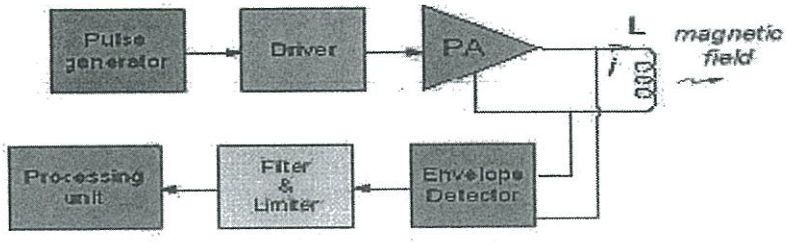
2.7.1 คุณสมบัติของ RFID

- ทำงานที่กระแส 13-26 mA / DC 3.3 V
- ความถี่ 13.5 MHz
- ระยะการสแกน 0-30 mm
- โปรโตคอล SPI
- ความเร็วในการสื่อสารข้อมูลสูงสุดที่ 10 Mbit/sec
- ขนาด: 40 mm × 60 mm

2.7.2 เครื่องอ่าน (Reader)

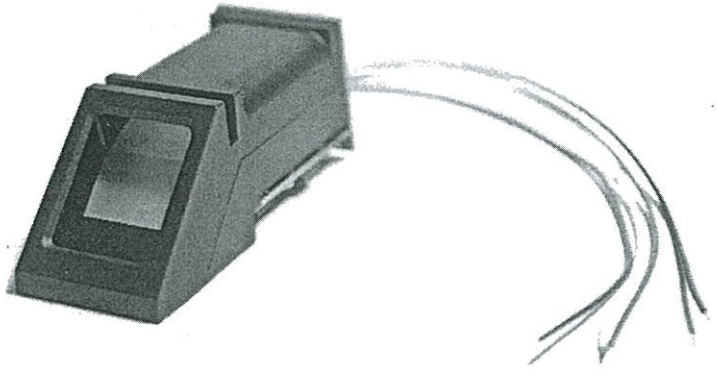
หน้าที่ของเครื่องอ่านคือ การเชื่อมต่อเพื่อเขียน หรืออ่านข้อมูลลงในแท็กด้วยสัญญาณความถี่วิทยุภายในเครื่องอ่าน จะประกอบด้วย เสาอากาศที่ทำจากขดลวดทองแดงเพื่อใช้รับส่งสัญญาณ ภาครับและภาคส่งสัญญาณวิทยุและวงจรควบคุมการอ่าน-เขียนข้อมูล โดยทั่วไปเครื่องอ่านจะประกอบด้วยส่วนประกอบหลักดังนี้

- ภาครับและส่งสัญญาณวิทยุ
- ภาคสร้างสัญญาณพาหะ
- ขดลวดที่ทำหน้าที่เป็นสายอากาศ
- วงจรจูนสัญญาณ
- หน่วยประมวลผลข้อมูล และภาคติดต่อกับไมโครคอนโทรลเลอร์



รูปที่ 2.19 บล็อคการทำงานของเครื่องอ่าน

2.8 Fingerprint scanner Module



รูปที่ 2.20 Fingerprint scanner module

โมดูลสแกนลายนิ้วมือนี้ สามารถเพิ่มค้นหาลายนิ้วมือได้อย่างง่ายแล้วยังสามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ด้วย TTL serial และส่งแพ็คเกตของข้อมูลเพื่อไปบันทึกภาพลายนิ้วมือ, ค้นหาลายนิ้วมือ, และสืบค้นข้อมูลลายนิ้วมือ โดยสามารถเพิ่มลายนิ้วมือได้มากถึง 162 ลายนิ้วมือ ซึ่งเก็บข้อมูลไว้บน FLASH memory ไฟ LED สีแดงจะสว่างขึ้นระหว่างรอรับภาพลายนิ้วมือ

2.8.1 คุณสมบัติของโมดูล Fingerprint scanner

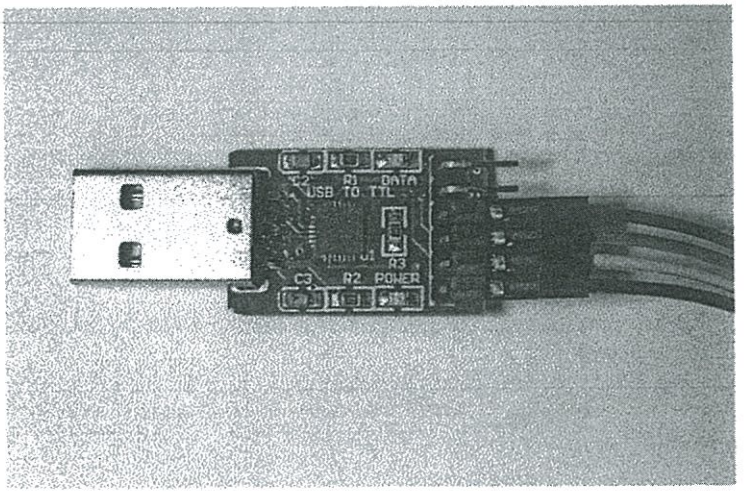
จากการศึกษาโมดูล Fingerprint scanner ในเบื้องต้นพบว่ามีคุณสมบัติทางเทคนิคดังนี้

- ไฟเลี้ยง : 3.6 - 6.0 VDC
- ทำงานที่กระแส : สูงสุด 120 mA
- กระแสสูงสุด : 150 mA
- เวลาในการเก็บภาพลายนิ้วมือ : <1.0 วินาที
- ขนาดหน้าจอ : 14 มม. x 18 มม.
- ไฟล์ลายนิ้วมือ : 256 ไบต์
- ไฟล์ต้นแบบ : 512 ไบต์
- บรรจุลายนิ้วมือได้ : 162 แบบ
- อัตราความผิดพลาดที่ยอมรับได้ : <0.001%

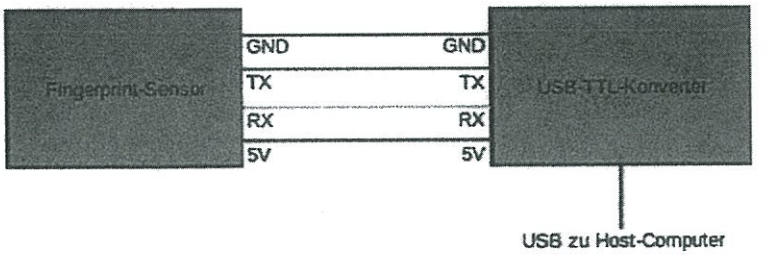
- อัตราความผิดพลาดที่ปฏิเสธ : < 1.0%
- อินเตอร์เฟส : TTL Serial
- ทำงานในอุณหภูมิ : -20 °C to +50 °C
- ทำงานที่ความชื้น : 40% - 85% RH
- ขนาดอุปกรณ์ : 56 มิลลิเมตร x 20 มิลลิเมตร x 21.5 มิลลิเมตร
- น้ำหนัก : 20 กรัม

2.8.2 การเชื่อมต่อโมดูล Fingerprint scanner

ราสเบอรรี่ พาย มีพอร์ต GPIO สำหรับต่อแบบ UART และสามารถต่อด้วยตัวแปลง USB-TTL โดยเป็นการต่อโดยตรง



รูปที่ 2.21 USB-TTL converting board



รูปที่ 2.22 การเชื่อมต่อโมดูล Fingerprint กับ USB-TTL

2.8.3 การติดตั้ง Software Fingerprint scanner

1. สร้างแพ็คเกจใหม่

```
~$ sudo apt-get install devscripts
~$ git clone
https://github.com/bastianraschke/pyfingerprint.git
~$ cd ./pyfingerprint/src/
~$ debuild
```

2. การติดตั้งโปรแกรม

```
~$ sudo dpkg -I ../python-fingerprint*.deb
~$ sudo apt-get -f install
~$ sudo usermod -a -G dialout pi
~$ sudo reboot
```

3. ตรวจสอบการทำงานของโปรแกรม

```
~$ python2 /usr/share/doc/python-
fingerprint/examples/example_enroll.py
~$ python2 /usr/share/doc/python-
fingerprint/examples/example_search.py
~$ python2 /usr/share/doc/python-
fingerprint/examples/example_delete.py
~$ python2 /usr/share/doc/python-
fingerprint/examples/example_downloadimage.py
```

2.8.4 หลักการทำงานของ Fingerprint scanner module

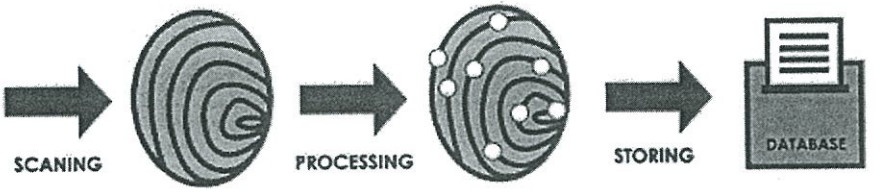
บริเวณปลายนิ้วมือของมนุษย์โดยทั่วไปจะเห็นลายนิ้วมือที่มีลักษณะประกอบไปด้วยเส้น 2 ลักษณะคือ เส้นนูน (Ridges) และ เส้นร่อง (Furrows) ซึ่งเส้นทั้ง 2 ลักษณะจะอยู่สลับกันไปตลอด โดยจุดลักษณะสำคัญบนลายนิ้วมือ (Characteristics) คือ ตำหนิต่าง ๆ บนลายนิ้วมือ สามารถแบ่งได้เป็น 2 ลักษณะดังนี้

1. ตำหนิและลักษณะต่าง ๆ ของลายเส้นทั่ว ๆ ไปเช่น เส้นตรง, เส้นโค้ง, จุด, เส้นแตก, เส้นวกกลับ, เส้นเวียน, เส้นขาด, เส้นทะเลสาบ, เส้น 2 เส้นมาพบกัน
2. ลักษณะพิเศษบางอย่าง เช่น

- ไบฟูร์เคชัน (Bifurcation) คือ เส้นขอบหนึ่งที่ได้ถูกแยกออกเป็น 2 เส้นหรือมากกว่าสองเส้น
- ไดเวอร์เจนซ์ (Divergence) คือ เส้นขอบที่ได้วิ่งขนานกันมาหรือเกือบจะขนานและได้แยก ถ่างออกไป
- จุดมินูเทีย (Minutiae) คือ จุดบนปลายเส้นหยุดหรือเส้นแยก

2.8.5 หลักการวิเคราะห์ลายนิ้วมือ

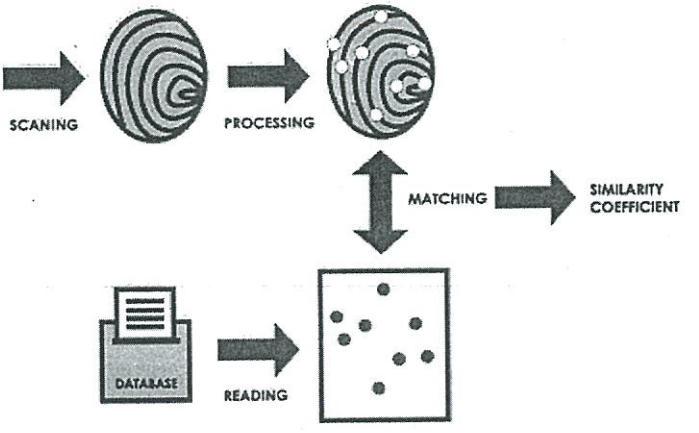
การวิเคราะห์ลายนิ้วมือของบุคคลโดยทั่วไปนั้น จะเริ่มต้นด้วยการนำลายนิ้วมือแต่ละนิ้วของแต่ละบุคคล มาหาจุดลักษณะเฉพาะที่สำคัญ กระบวนการแรกเริ่มของการตรวจพิสูจน์ลายนิ้วมือคือ การอ่านภาพลายนิ้วมือเข้ามาเก็บไว้ในหน่วยความจำที่เก็บข้อมูลไว้ โดยข้อมูลที่อ่านหรือสแกนเข้ามานั้นจะนำมาผ่านการประมวลผลก่อนแล้วจึงเก็บข้อมูลนั้นไว้ ซึ่งข้อมูลนี้จะถูกเก็บไว้เป็นต้นแบบหรือรหัสข้อมูลของผู้ใช้แต่ละคน ในขั้นตอนก่อนที่จะนำลายนิ้วมือเข้าไปเก็บนั้นจะต้องผ่านขั้นตอนของการประมวลผลอีกครั้ง โดยในกระบวนการนี้จะทำให้ภาพที่ได้รับการสแกนเข้ามาเกิดความสมบูรณ์มากขึ้นเพราะเมื่อเครื่องได้รับการสแกนภาพเข้ามาแล้ว ภาพที่อ่านได้อาจไม่ชัดเจน หรือพร่าเลือน จะทำให้การประมวลผลในขั้นตอนถัดไปทำได้ยากลำบากมากขึ้น ซึ่งอาจจะทำให้ผลที่ได้ไม่ถูกต้องตามที่ควรจะเป็น เมื่อเกิดปัญหาเช่นนี้ในกระบวนการนี้จึงได้มีการกระทำหลายกระบวนการด้วยกันคือ การกำจัดสัญญาณรบกวน, การปรับความมืดสว่าง, ความแตกต่างของตัวภาพและฉากของภาพ, การแปลงภาพเป็นภาพสองระดับ, การทำให้เส้นลายนิ้วมือบาง, การปรับภาพหลังจากแปลงภาพเป็นสองระดับ, การหาค่า Threshold ของการปรับภาพเป็นภาพสองระดับ และอื่น ๆ อีกมาก ซึ่งกระบวนการจะมีมากหรือน้อยนั้น ขึ้นอยู่กับว่าตัวอุปกรณ์นั้นมีการอ่านค่าลายนิ้วมือที่ได้ออกมามีภาพละเอียดและสมบูรณ์เพียงใด เมื่อได้ลายนิ้วมือที่ผ่านการประมวลผลแล้วก็จะนำข้อมูลหรือภาพนี้ไปจัดเก็บในหน่วยความจำ โดยภาพที่ถูกจัดเก็บไว้นี้ จะเก็บไว้เพื่อใช้ในการเปรียบเทียบกับลายนิ้วมือที่ได้รับการสแกนเข้ามาเมื่อนำตัวอุปกรณ์นี้ไปใช้งาน



รูปที่ 2.23 กระบวนการทำงานของระบบสแกนนิ้ว

จากรูปที่ 2.23 เริ่มต้นด้วยการสแกนลายนิ้วมือเข้ามาแล้วนำภาพที่ได้ผ่านการประมวลผลซึ่งจะได้ภาพที่มีประสิทธิภาพมากขึ้นแล้วจึงเก็บภาพนั้นไว้ หลังจากเก็บภาพไว้แล้วก็จะมาถึงขั้นตอนการนำไปใช้งาน เมื่อตัวอุปกรณ์ได้ถูกบันทึกหรือเก็บลายนิ้วมือของผู้ที่จะนำไปใช้แล้ว ขั้นตอนในการใช้ก็จะคล้ายกับตอนอ่านลายนิ้วมือเข้ามาเก็บไว้ เพียงแต่การอ่านเข้ามาครั้งนี้ข้อมูลที่

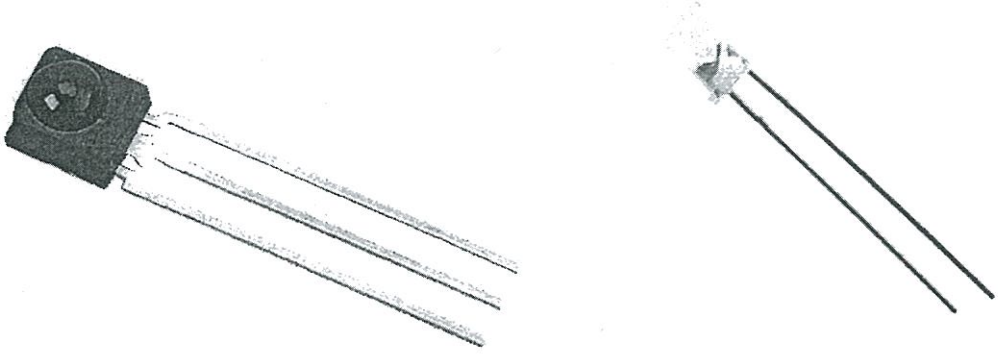
ได้จะถูกนำไปเก็บไว้ที่หน่วยความจำ ซึ่งหลังจากสแกนเข้ามาแล้วประมวลผลแล้วก็จะทำการเก็บข้อมูลไว้ที่ส่วนของหน่วยความจำชั่วคราว ส่วนในขั้นตอนถัดไปก็จะนำข้อมูลที่เก็บอยู่ในส่วนของหน่วยความจำถาวร กับส่วนที่เก็บอยู่ในหน่วยความจำชั่วคราวนั้นมาทำการเปรียบเทียบกัน (Matching) เมื่อได้ผลแล้วก็จะแจ้งผลให้ผู้ใช้ทราบว่ามีความเหมือนกันมากน้อยแค่ไหน



รูปที่ 2.24 หลักการตรวจสอบความถูกต้อง

จากรูปที่ 2.24 จะแสดงให้เห็นถึงกระบวนการเปรียบเทียบลายนิ้วมือที่ได้รับการสแกนเข้ามา โดยเริ่มที่ การสแกนภาพเข้ามา แล้วทำการประมวลผลขั้นตอนเดียวกันกับการจัดเก็บตอนแรกแล้วนำภาพที่เก็บไว้ในตอนแรกมาเปรียบเทียบกับภาพที่สแกนเข้ามา ณ ตอนนั้นเพื่อเปรียบเทียบว่ามีความเหมือนหรือแตกต่างกันมากเพียงใด ทั้งสองขั้นตอนนั้นต้องผ่านการประมวลผลซึ่งจะทำให้ได้ภาพที่มีประสิทธิภาพในการเปรียบเทียบมากขึ้น ซึ่งขั้นตอนนี้มีหลักการและทฤษฎีมากมายที่ต้องทำความเข้าใจและศึกษาเป็นตัวอย่างในการทำการประมวลผลภาพก่อนทำขั้นตอนอื่น ซึ่งผลที่ได้จากการทำจะทำให้ได้จุดลักษณะเฉพาะ ซึ่งจุดเหล่านี้เองจะเป็นสิ่งในการเปรียบเทียบลายนิ้วมือของแต่ละคนหรือกล่าวได้ว่าเป็นตัวบ่งชี้ความแตกต่างของนิ้วแต่ละคน

2.9 IR Module



รูปที่ 2.25 IR Receiver/Transmitter Sensor

2.9.1 คุณสมบัติ

ระบบจะเชื่อมต่อกับส่วนเชื่อมต่อผู้ใช้ (IR module interface) สำหรับการบันทึกค่าสัญญาณอินฟราเรดจากรีโมท มีคุณสมบัติดังนี้

- ตัวรับอินฟราเรดจะรับที่ความถี่พาหะ 38 KHz
- ความน่าเชื่อถือสูง และ ความเข้มการแผ่รังสีสูง
- ความยาวคลื่น 940 นาโนเมตร
- ใช้แรงดันไฟฟ้าต่ำ

การแปลงสัญญาณ IR จากรีโมทคอนโทรลของเครื่องปรับอากาศให้เป็นข้อมูลดิจิทัลโดยทั่วไป ในการติดตั้งและใช้งานเครื่องปรับอากาศ มักจะมาพร้อมกับอุปกรณ์รีโมทคอนโทรล (Remote Control) สำหรับผู้ใช้อุปกรณ์ประเภทนี้ส่งสัญญาณด้วยแสงอินฟราเรด (Infrared) เพื่อเปิด-ปิด หรือปรับค่าอุณหภูมิได้

2.9.2 การตั้งโปรแกรม Library ที่ช่วยจัดการ IR

1. ลง library lirc

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install lirc
$ sudo reboot
```

2. ทดสอบ IR Sensor บน Raspberry Pi

```
$ mode2 -d /dev/lirc0
```

3. Create remote configuration

```
$ irrecord – list-namespace
$ irrecord -d /dev/lirc0 aircon.conf
```

2.9.3 ทดสอบสัญญาณอินฟราเรด

ขั้นตอนสุดท้ายคือเรียกใช้งานคลื่นอินฟราเรดที่เราเก็บค่าไว้ในไลบรารี /etc/lirc/lircd.conf มาใช้งานโดยใช้คำสั่งดังนี้

```
$ irsend SEND_ONCE Air ON      #คำสั่งเปิดแอร์
$ irsend SEND_ONCE Air OFF     #คำสั่งปิดแอร์
$ irsend SEND_ONCE Air KEY_30 #คำสั่งปรับอุณหภูมิไปที่ 30 °C
$ irsend SEND_ONCE Air KEY_29 #คำสั่งปรับอุณหภูมิไปที่ 29 °C
$ irsend SEND_ONCE Air KEY_28 #คำสั่งปรับอุณหภูมิไปที่ 28 °C
$ irsend SEND_ONCE Air KEY_27 #คำสั่งปรับอุณหภูมิไปที่ 27 °C
$ irsend SEND_ONCE Air KEY_26 #คำสั่งปรับอุณหภูมิไปที่ 26 °C
$ irsend SEND_ONCE Air KEY_25 #คำสั่งปรับอุณหภูมิไปที่ 25 °C
$ irsend SEND_ONCE Air KEY_24 #คำสั่งปรับอุณหภูมิไปที่ 24 °C
$ irsend SEND_ONCE Air KEY_23 #คำสั่งปรับอุณหภูมิไปที่ 23 °C
$ irsend SEND_ONCE Air KEY_22 #คำสั่งปรับอุณหภูมิไปที่ 22 °C
$ irsend SEND_ONCE Air KEY_21 #คำสั่งปรับอุณหภูมิไปที่ 21 °C
```

2.10 Speech recognition

Speech recognition เป็นเทคโนโลยีที่ทำให้คอมพิวเตอร์รู้จำเสียงมนุษย์ เพื่อแปลงเสียงมนุษย์เป็นคำสั่งให้คอมพิวเตอร์เข้าใจ ในการพัฒนาเทคโนโลยี Speech recognition นี้ เป็นเทคโนโลยีขั้นสูง ต้องลงทุนสูงและต้องอาศัยประสบการณ์ ความรู้ของโปรแกรมเมอร์เป็นอย่างยิ่ง ซึ่งยากเกินไป ในปริญญานิพนธ์นี้จะใช้ Speech recognition ด้วย Google Speech Recognition API (รองรับภาษาไทย) ในการใช้งาน Speech recognition ด้วย Google Speech Recognition API ใน Python จะต้องใช้โมดูล Speech Recognition (BSD license) รองรับทั้ง Python 2, Python 3 และรองรับทั้ง Linux, Mac OS, Windows

- รองรับ Google Speech Recognition API, Wit.ai, IBM Speech to Text และ AT&T Speech to Text
- รองรับภาษาไทย (เฉพาะ Google Speech Recognition API)
- ใช้ BSD license

2.10.1 การติดตั้ง Software

```
$ pip install SpeechRecognition
$ pip install PyAudio
```

2.10.2 ตัวอย่างการใช้งาน

```
import speech_recognition as sr
r = sr.Recognizer()
with sr.Microphone() as source:                    # เรียกใช้
Microphone พื้นฐานของระบบ
    audio = r.listen(source)                       # รับเสียงเข้ามาแล้ว
ประมวลผลส่งไปยัง Google Speech Recognition API

try:
    print("You said " + r.recognize_google(audio))    # แสดง
ข้อความจากเสียงด้วย Google Speech Recognition
except LookupError:                                # ประมวลผลแล้วไม่รู้
จักหรือเข้าใจเสียง
    print("Could not understand audio")
```

รูปที่ 2.26 รับเสียงมาจาก Microphone

```
import speech_recognition as sr
r = sr.Recognizer()
with sr.WavFile("test.wav") as source:                # ใช้
"test.wav"  เป็นแหล่งให้ข้อมูลเสียง
    audio = r.record(source)                          # ส่งข้อมูล
เสียงจากไฟล์
try:
    print("Transcription: " + r.recognize_google(audio))   #
แสดงข้อความจากเสียงด้วย Google Speech Recognition
except sr.RequestError as e:                                  #
ประมวลผลแล้วไม่รู้จักหรือเข้าใจเสียง
    print("Could not understand audio")
```

รูปที่ 2.27 ประมวลเสียงจากไฟล์เสียง wav

```
import speech_recognition as sr
r = sr.Recognizer()
with sr.Microphone() as source:
    audio = r.listen(source)
try:
    print("You said " + r.recognize_google(audio,language =
"th-TH"))    # แสดงข้อความจากเสียงด้วย Google Speech Recognition
และกำหนดค่าภาษาเป็นภาษาไทย
except sr.RequestError as e:                                  #
ประมวลผลแล้วไม่รู้จักหรือเข้าใจเสียง
    print("Could not understand audio")
```

รูปที่ 2.28 แสดงตัวอย่างชุดคำสั่งเบื้องต้น

บทที่ 3

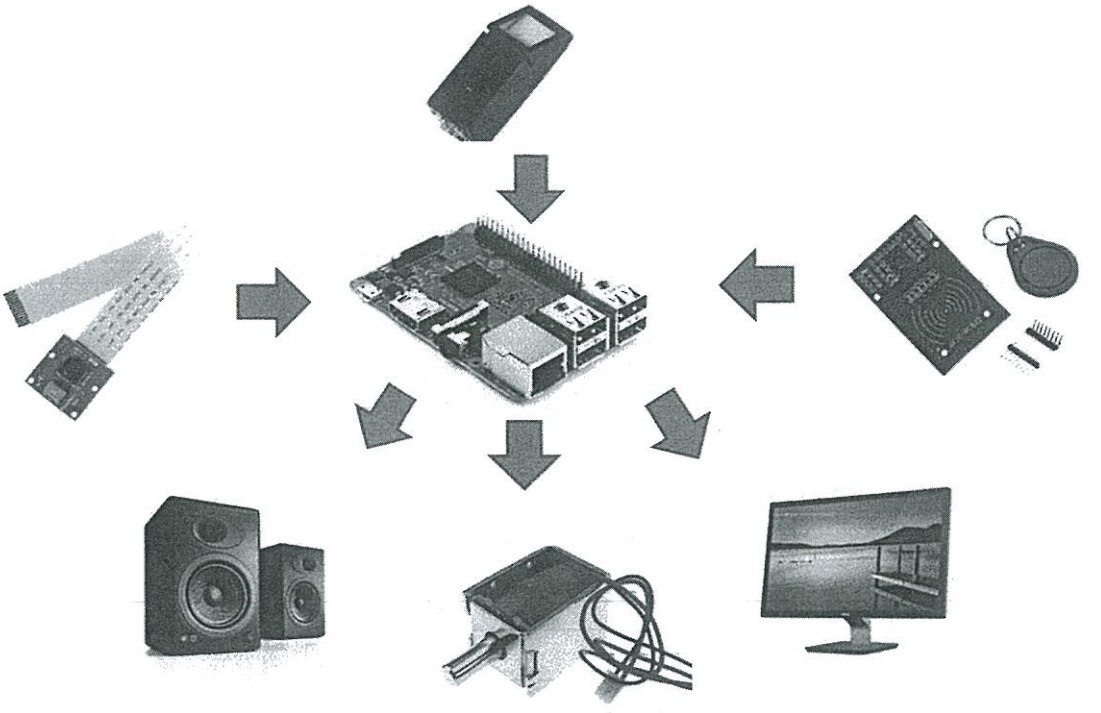
การออกแบบและการจัดทำโครงงาน

3.1 การออกแบบ

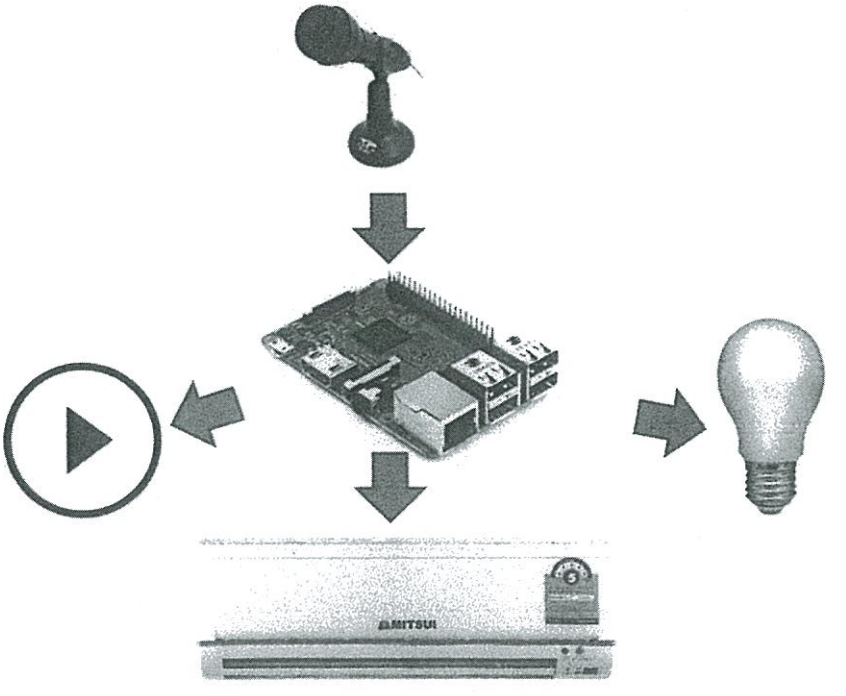
การทำงานของระบบประกอบไปด้วยส่วนของการรักษาความปลอดภัย และส่วนของการสั่งงานอุปกรณ์ต่าง ๆ ด้วยคำสั่งเสียง ในส่วนของระบบรักษาความปลอดภัย ซึ่งจะประกอบไปด้วย 1.) การตรวจจับใบหน้าเพื่อยืนยันตัวบุคคล 2.) การสแกนบัตรเพื่อยืนยันตัวบุคคล และ 3.) การสแกนลายนิ้วมือเพื่อยืนยันตัวบุคคล โดยมีเงื่อนไขการใช้งานคือ เลือกใช้ระบบรักษาความปลอดภัยเพียงหนึ่งระบบ เพื่อยืนยันตัวบุคคลในการผ่านเข้าประตู จากนั้นจะประมวลผลและตรวจสอบบุคคลที่เข้ามายืนยัน เทียบกับฐานข้อมูลที่เก็บข้อมูลของผู้ใช้งาน โดยจะเก็บข้อมูลเวลาการใช้งานห้องของสมาชิกนั้น ๆ และในส่วนของระบบคำสั่งเสียงในห้องจะรับค่าอินพุตเป็นเสียง โดยใช้คีย์เวิร์ดที่ตั้งไว้เป็นการเปิดการสั่งงานอุปกรณ์ในห้อง ได้แก่ หลอดไฟ, เครื่องปรับอากาศ และเครื่องเล่นเพลง ทั้งหมดนี้ผ่านการประมวลผลการใช้งานด้วย Raspberry Pi 3

3.1.1 Block Diagram

จะประกอบไปด้วยสองส่วนหลักคือในส่วนของระบบรักษาความปลอดภัย และส่วนของการสั่งงานด้วยเสียงภายในห้อง การทำงานของระบบแสดงได้ดังรูป โดยในส่วนของระบบรักษาความปลอดภัย จะเริ่มรับอินพุตที่ป้อนเข้ามายังระบบรักษาความปลอดภัย จากนั้นจะส่งข้อมูลที่ได้มาตรวจสอบเปรียบเทียบในฐานข้อมูล ซึ่งถ้าหากข้อมูลที่รับมาตรงกับฐานข้อมูลประตูแม่เหล็กไฟฟ้าจะเปิด ในทางกลับกัน ถ้าไม่ตรงกับฐานข้อมูลประตูจะไม่เปิด และในส่วนของการสั่งงานด้วยเสียงภายในห้องจะรับอินพุตเป็นเสียง โดยจะมีชื่อเรียกที่เป็นคีย์เวิร์ดเพื่อใช้ในการเรียกใช้งานสั่งการอุปกรณ์ไฟฟ้าภายในห้อง โดยทั้งหมดจะใช้ Raspberry Pi 3 ในการประมวลผล

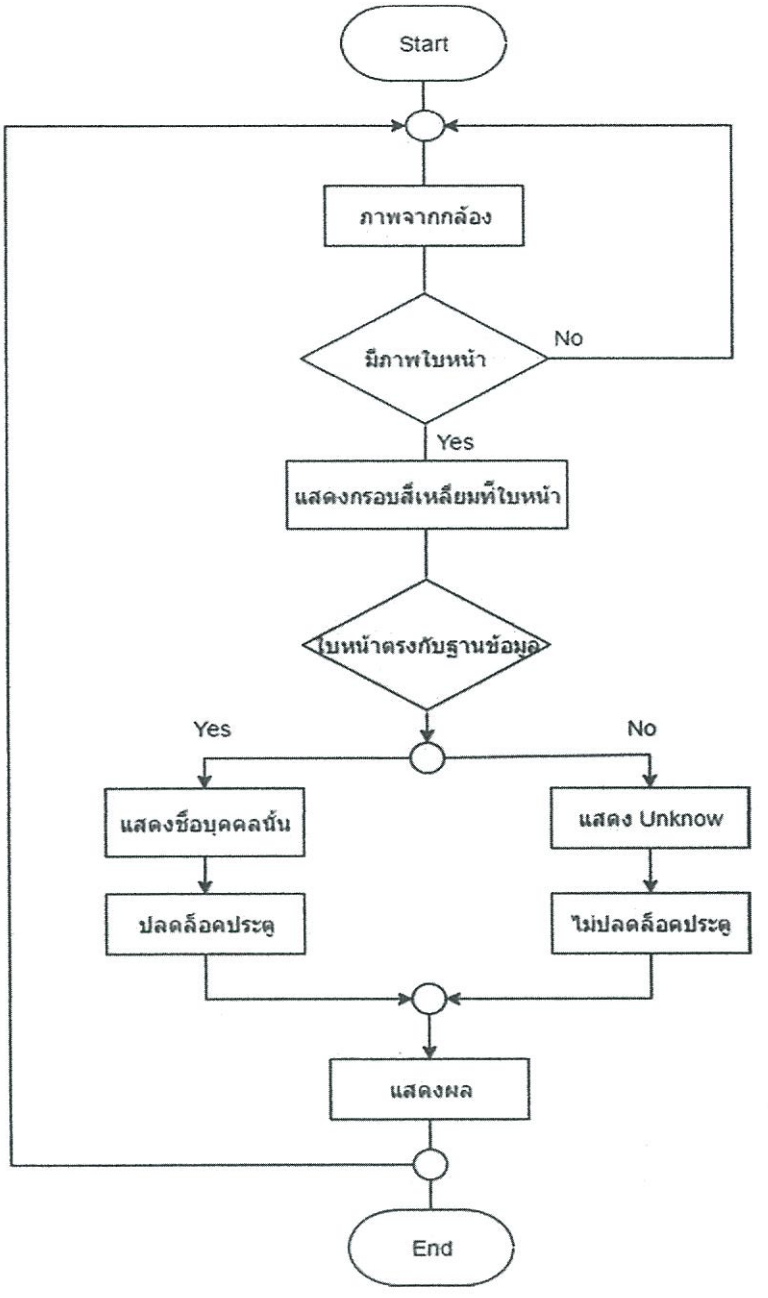


รูปที่ 3.1 บล็อกไดอะแกรมของระบบรักษาความปลอดภัย



รูปที่ 3.2 การสั่งงานด้วยเสียงภายในห้อง

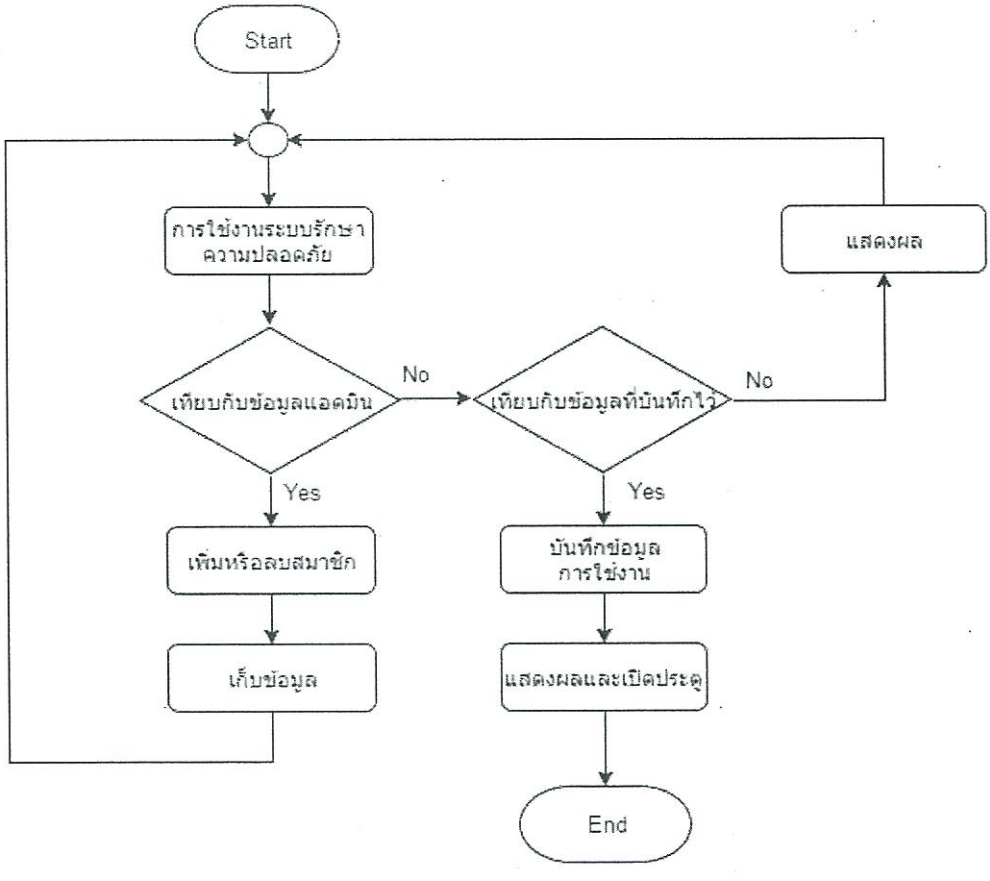
3.1.2 Flow Chart กระบวนการทำงานของระบบการตรวจจับใบหน้าเพื่อยืนยันตัวบุคคล



รูปที่ 3.3 Flow Chart กระบวนการทำงานของระบบการตรวจจับใบหน้าเพื่อยืนยันตัวบุคคล

Flow Chart ของระบบตรวจจับใบหน้าเพื่อยืนยันตัวบุคคล จะทำงานโดยเริ่มจากการใช้กล้องถ่ายภาพแบบ real-time หากมีภาพของผู้ใช้งานระบบจะทำการตรวจจับใบหน้าและสร้างกรอบรอบใบหน้าไว้ซึ่งจะแสดงผลในจอแสดงผล จากนั้นจะไปเทียบภาพหน้าของผู้ใช้งานในขณะนั้นกับภาพที่มีในฐานข้อมูล หากพบว่าไม่ตรงกับฐานข้อมูลจะแสดง Unknow บนกรอบที่เป็นสีแดงในจอแสดงผลและประตูจะไม่เปิด แต่หากตรวจพบว่าภาพของผู้ใช้งานนั้นตรงกับฐานข้อมูลประตูจะเปิดและจะแสดงชื่อของบุคคลนั้น ๆบนเฟรมสีฟ้าในจอแสดงผล โดยเป็นชื่อที่มาจากชื่อภาพในฐานข้อมูล

3.1.3 Flow Chart ของระบบรักษาความปลอดภัยแบบสแกนบัตรและแบบสแกนลายนิ้วมือ เพื่อยืนยันตัวบุคคล

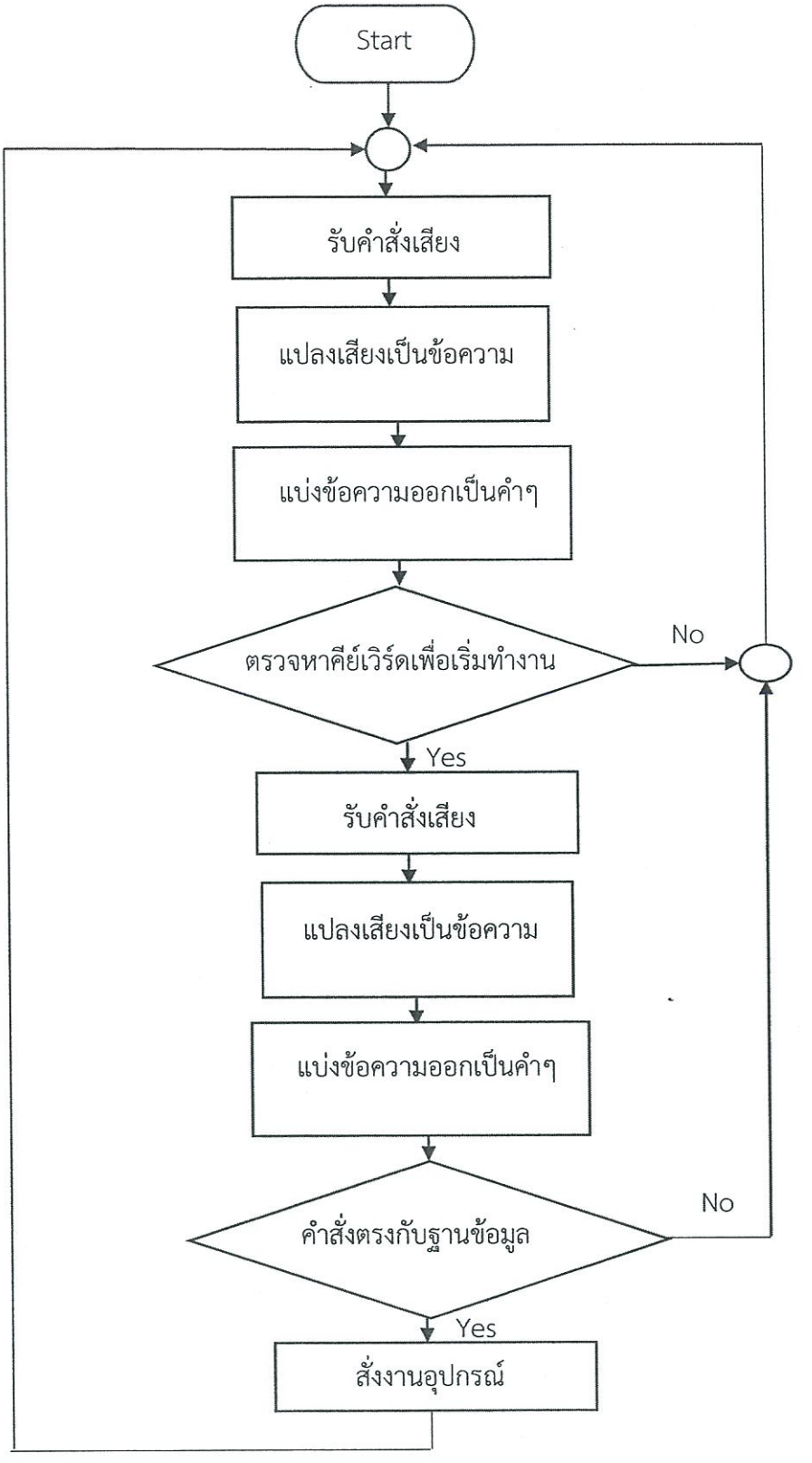


รูปที่ 3.4 Flow Chart ของระบบรักษาความปลอดภัยแบบสแกนบัตรและแบบสแกนลายนิ้วมือเพื่อยืนยันตัวบุคคล

Flow Chart ของทั้งสองระบบจะเหมือนกัน แต่จะแตกต่างกันที่อินพุตที่เป็นเป็นการ์ด RFID และ ลายนิ้วมือ โดยจากรูปจะเห็นว่ามีสองระบบคือ ระบบการใช้งานปกติจะใช้การ์ด RFID มาสแกน จากนั้นระบบจะตรวจสอบเทียบกับฐานข้อมูลซึ่งจะแสดงผล และเปิดประตู ในกรณีที่ไม่มีข้อมูลของการ์ดที่ใช้งานในระบบจะแสดงผลว่าใช้งานไม่ได้ และประตูจะไม่เปิด ในส่วนของระบบเพิ่มหรือลบสมาชิกจะใช้มาสเตอร์การ์ดที่กำหนดในการใช้งานการเข้าสู่โหมด จากนั้นจะใช้การ์ดที่ต้องการเพิ่มหรือลบสมาชิกก่อนจะเก็บข้อมูลลงฐานข้อมูล และใช้มาสเตอร์การ์ดแตะเพื่อทำการออกจากมาสเตอร์โหมดอีกที

3.1.4 Flow Chart กระบวนการทำงานของการสั่งงานด้วยเสียง

Flow Chart ของระบบการสั่งงานด้วยเสียงนั้น แรกเริ่มนั้นจะรับเสียงพูดไปแปลงเป็นข้อความโดยใช้ API ของ Google ในการแปลง จากนั้นแบ่งข้อความออกเป็นคำ ๆ เพื่อให้ง่ายต่อการหาคำที่เป็นคีย์เวิร์ดในการเรียกใช้งาน โดยในที่นี้ใช้ชื่อ “คอมพิวเตอร์” เพื่อเปิดการสั่งงาน จากนั้นจะสามารถสั่งงานอุปกรณ์ด้วยเสียงได้โดยวิธีต่าง ๆ ในตอนต้นจะเหมือนกันคือ รับเสียง, แปลงเสียงเป็นข้อความ, แบ่งข้อความเป็นคำ ๆ จากนั้นจะเทียบคำจากผู้ใช้งานกับคำสั่งที่มีเพื่อนำไปสั่งงานอุปกรณ์ต่อไป



รูปที่ 3.5 Flow Chart กระบวนการทำงานของการสั่งงานด้วยเสียง

3.1.5 คำที่ใช้ในการเปรียบเทียบกับข้อความบางส่วน

คำสั่งต่าง ๆ ที่ใช้เป็นคีย์เวิร์ดเพื่อเปรียบเทียบ และนำไปสั่งงาน ซึ่งมีคำสั่งโดยประมาณ ดังนี้

3.1.5.1 ควบคุมหลอดไฟ

เปิดไฟ ใช้คำสั่ง : “เปิดไฟ”, “ติดไฟ”, “สว่าง”

ปิดไฟ ใช้คำสั่ง : “ปิดไฟ”, “ดับไฟ”, “มืด”

3.1.5.2 ควบคุมเครื่องปรับอากาศ

เปิดเครื่องปรับอากาศ ใช้คำสั่ง : “ติดเครื่องปรับอากาศ”, “เปิดแอร์”, “ติดแอร์”, “เปิดเครื่องปรับอากาศ”

ปิดเครื่องปรับอากาศ ใช้คำสั่ง : “ปิดแอร์”, “ปิดเครื่องปรับอากาศ”, “ดับแอร์”, “ดับเครื่องปรับอากาศ”

ปรับอุณหภูมิ ใช้คำสั่ง : “T องศา”, “ปรับอุณหภูมิ T องศา” (T คืออุณหภูมิที่ต้องการ)

3.1.5.3 ควบคุมเครื่องเล่นเพลง

เปิดเพลง ใช้คำสั่ง : “เล่นเพลง”, “เริ่มเพลง”

ปิดเพลง ใช้คำสั่ง : “หยุดเล่น”

เพลงถัดไป ใช้คำสั่ง : “เพลงต่อไป”, “เพลงถัดไป”, “เพลงหน้า”

เพลงที่แล้ว ใช้คำสั่ง : “เพลงที่แล้ว”, “เพลงก่อนหน้า”, “เพลงเมื่อกี้”

เพิ่มเสียง ใช้คำสั่ง : “เพิ่มเสียงขึ้น” , “ดังขึ้น” (ระดับเสียงที่จะเพิ่มขึ้นมี 5 ระดับ)

ลดเสียง ใช้คำสั่ง : “ลดเสียงลง” , “เบาลง” (ระดับเสียงที่จะลดลงมี 5 ระดับ)

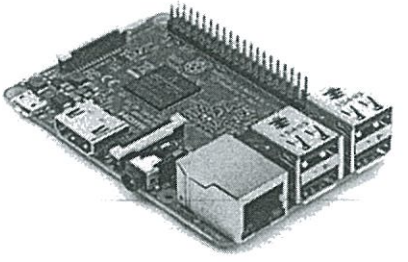
3.1.5.4 แนะนำการทำงาน

แนะนำตัว ใช้คำสั่ง : “ชื่ออะไร”, “แนะนำตัวให้ฟังหน่อย”

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Raspberry Pi 3

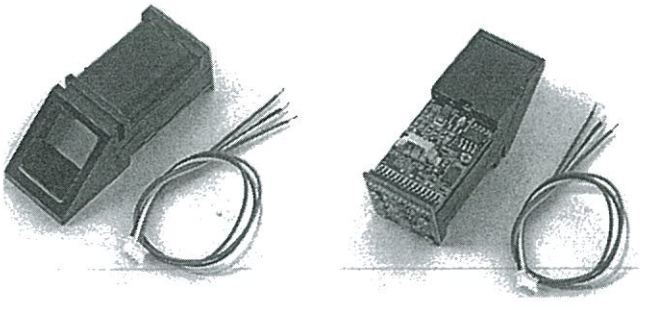
เป็นอุปกรณ์ที่ใช้เขียนโปรแกรมเพื่อประมวลผลข้อมูลที่ได้รับ เพื่อนำไปใช้ต่อไปในการสั่งงานอุปกรณ์อื่น ๆ ผ่านพอร์ต GPIO



รูปที่ 3.6 Raspberry Pi 3

3.2.2 Fingerprint Scanner

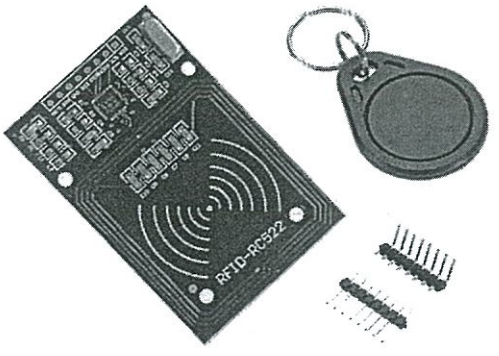
เป็นโมดูลที่ใช้ในการสแกนลายนิ้วมือและเก็บลายนิ้วมือได้ 162 ลายนิ้วมือ โดยจะเก็บค่าของแต่ละลายนิ้วมือไว้ในโมดูล



รูปที่ 3.7 Fingerprint scanner

3.2.3 RFID Module

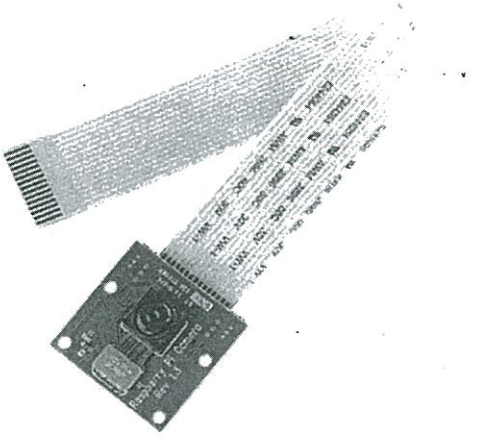
โมดูล RFID เป็นโมดูลที่รับค่าจากการ์ดที่มีความถี่เดียวกัน ซึ่งจะนำค่าจากการ์ดนั้น ๆ มาประมวลผลอีกที



รูปที่ 3.8 RFID Module

3.2.4 Raspberry Pi Camera Module

โมดูลกล้องที่ใช้ต่อกับ Raspberry Pi ใช้ในการทดลองการตรวจจับใบหน้า



รูปที่ 3.9 Raspberry Pi Camera Module

3.2.5 Microphone

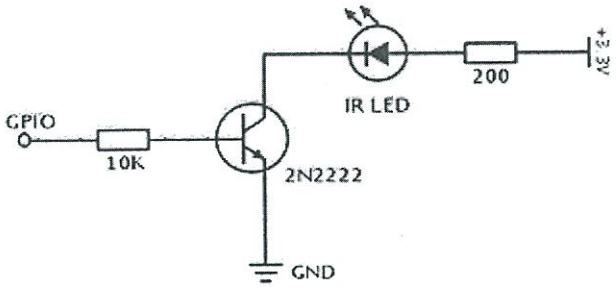
เป็นอุปกรณ์ที่ใช้รับเสียงเพื่อนำไปประมวลผลที่บอร์ด Raspberry Pi

รูปที่ 3.10 Microphone

3.2.6 Infrared

ใช้ส่งและรับค่าสัญญาณอินฟราเรดเพื่อไปควบคุมอุปกรณ์ต่าง ๆ ซึ่งค่าที่ได้จะได้จากการเก็บสัญญาณอินฟราเรดมีลักษณะเป็นพัลส์ จากรีโมทที่ใช้งานของอุปกรณ์นั้น ๆ จริง โดยจะออกแบบตามวงจรในรูปที่ 3.12

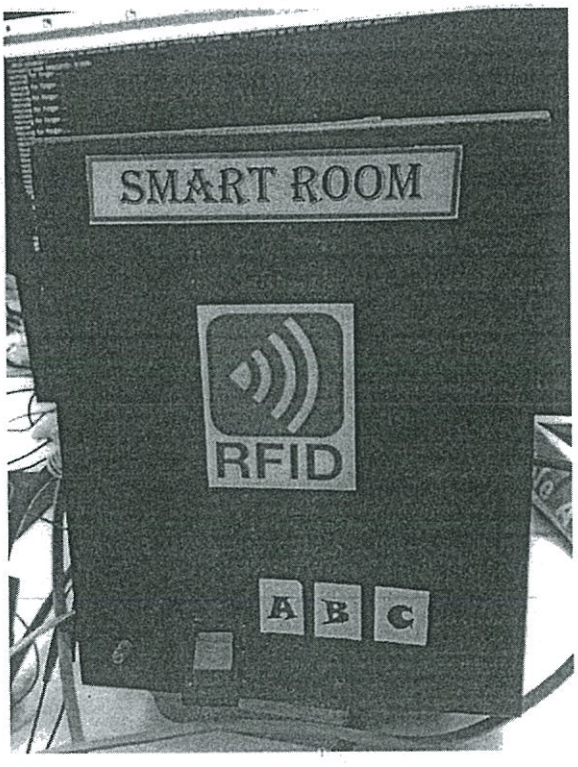
รูปที่ 3.11 Infrared



รูปที่ 3.12 Transistor-powered IR transmitter schematics

3.2.7 การรวมอุปกรณ์เข้าด้วยกัน

รวมในส่วนของอุปกรณ์เข้าด้วยกันและจัดเก็บสายภายในกล่องเพื่อให้ง่ายต่อการใช้งาน



รูปที่ 3.13 กล่องชิ้นงาน

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทำให้ Raspberry Pi สามารถติดต่อกับโมดูล Fingerprint Scanner เพื่ออ่านข้อมูลลายนิ้วมือที่เก็บภายในโมดูล

3.3.2 การทำให้ Raspberry Pi สามารถติดต่อกับโมดูลกล้องเพื่อใช้ในการถ่ายภาพ และประมวลผลเพื่อวิเคราะห์ใบหน้า

3.3.3 การทำให้ Raspberry Pi สามารถติดต่อกับโมดูล RFID เพื่อใช้ในการอ่านค่า, เพิ่มหรือลดสมาชิกและประมวลผลบัตร

3.3.4 การทดลองส่วนรู้จำภาพใบหน้าบุคคลว่าสามารถระบุได้ว่าบุคคลในภาพคือใครได้อย่างแม่นยำหรือไม่เมื่อเทียบกับภาพในฐานข้อมูล

3.3.5 เก็บข้อมูลของแต่ละบุคคลที่สามารถผ่านเข้าประตูได้ทั้งสามรูปแบบ

3.3.6 การเก็บข้อมูลความถูกต้องการทดลองของการแปลงเสียงพูดเป็นข้อความ

3.3.7 การทดลองผลการแบ่งข้อความที่ได้รับเป็นคำ

3.3.8 การทดลองผลการสั่งงานเครื่องปรับอากาศโดยใช้อินฟราเรด

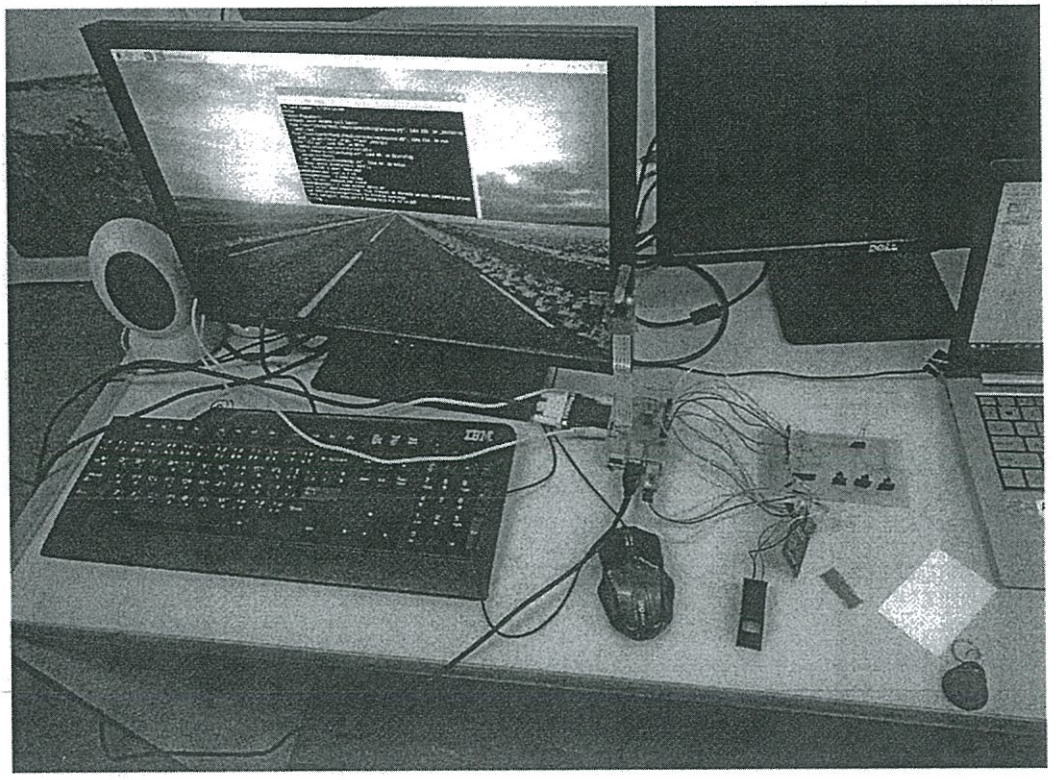
บทที่ 4

ผลการทดลอง

จากการออกแบบการทำงานของระบบในบทที่ 3 แล้ว ในบทนี้จะเป็นการแสดงผลของการทดลองการทำงานร่วมกันของระบบรักษาความปลอดภัยทั้งสามรูปแบบ และการทดลองระบบการสั่งงานด้วยเสียง โดยได้แบ่งการทดลองออกเป็นหัวข้อต่าง ๆ ดังต่อไปนี้

```
pi@raspberrypi:~/allin $ sudo python all.py
```

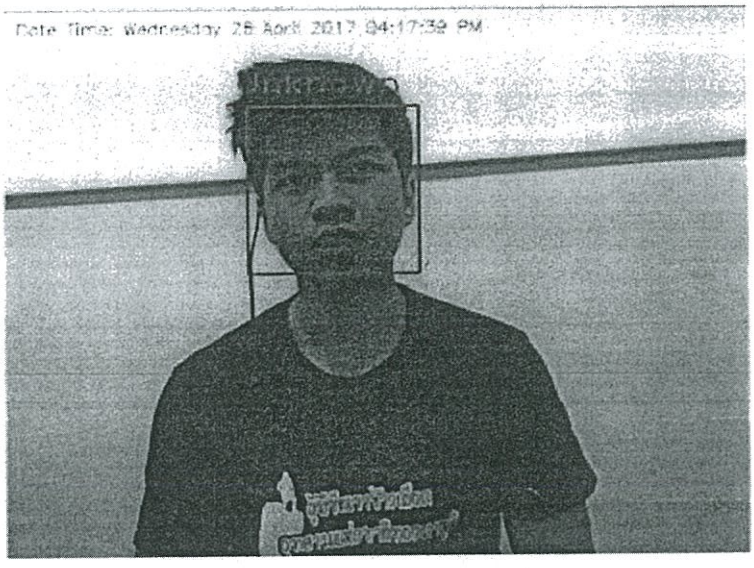
รูปที่ 4.1 คำสั่งเปิดไฟล์ที่ใช้งาน



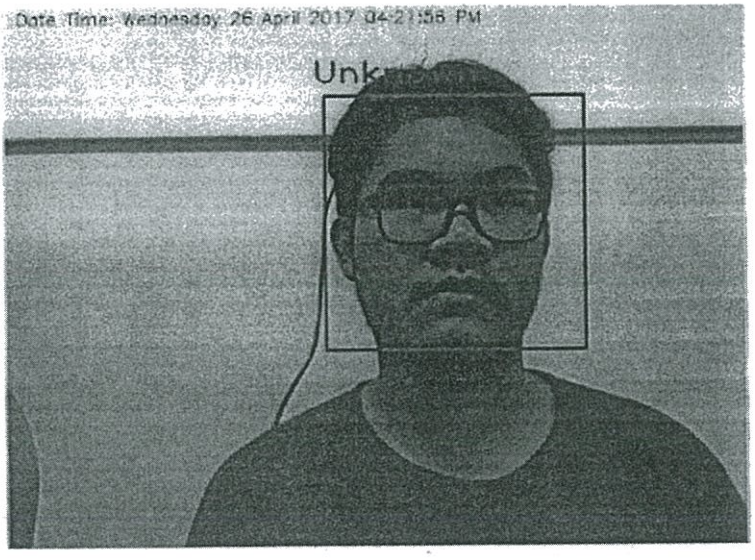
รูปที่ 4.2 การเตรียมการทดลอง

4.1 ผลจากการอ่านข้อมูลใบหน้าที่เก็บไว้ใน Raspberry Pi

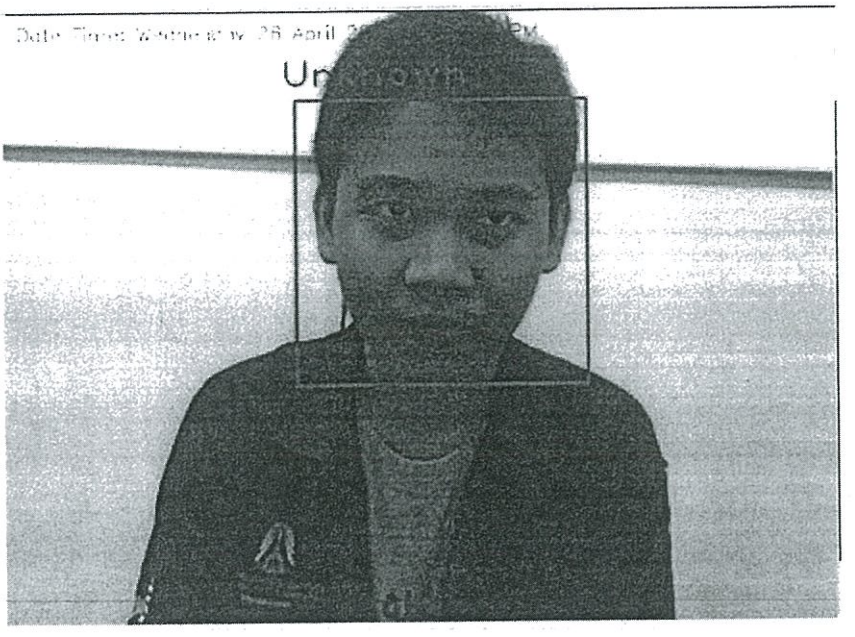
ในส่วนของการทดลองนี้เป็นการตรวจสอบใบหน้าของผู้ใช้งานว่าถูกต้องตรงกับฐานข้อมูลหรือไม่ซึ่งจะแสดงผลดังภาพ



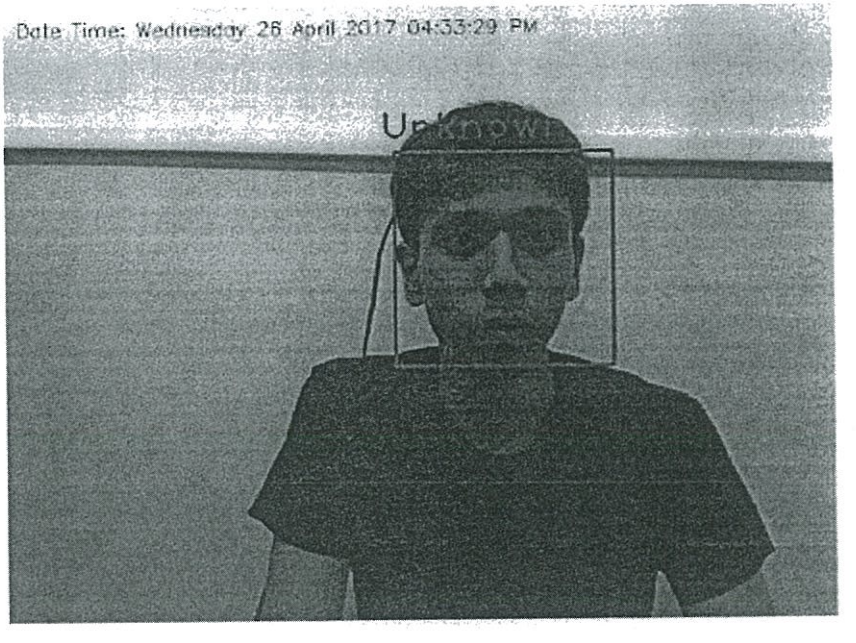
รูปที่ 4.3 ผู้ใช้งานบุคคลที่ 1 มีใบหน้าไม่ตรงกับฐานข้อมูล



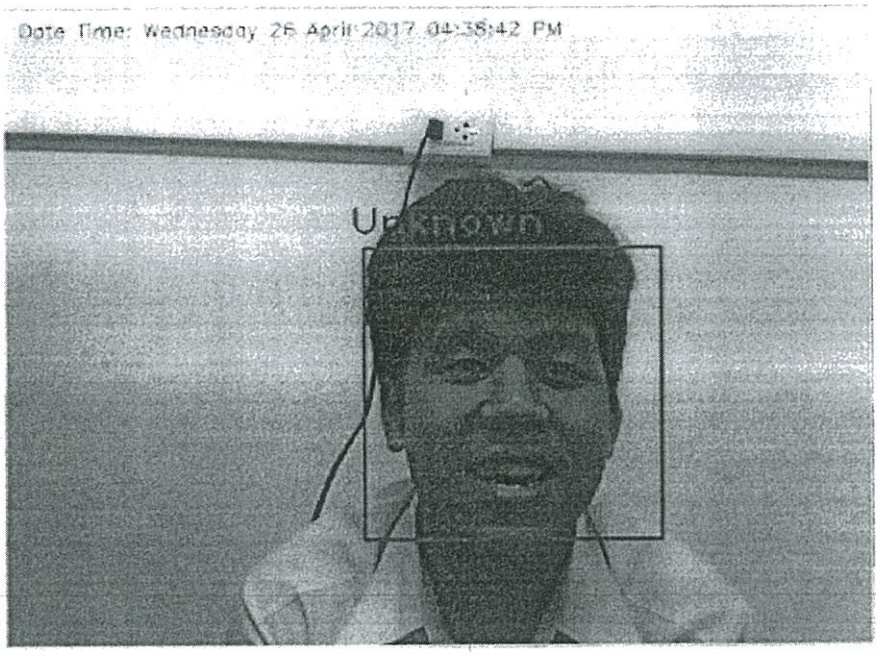
รูปที่ 4.4 ผู้ใช้งานบุคคลที่ 2 มีใบหน้าไม่ตรงกับฐานข้อมูล



รูปที่ 4.5 ผู้ใช้งานบุคคลที่ 3 มีใบหน้าไม่ตรงกับฐานข้อมูล

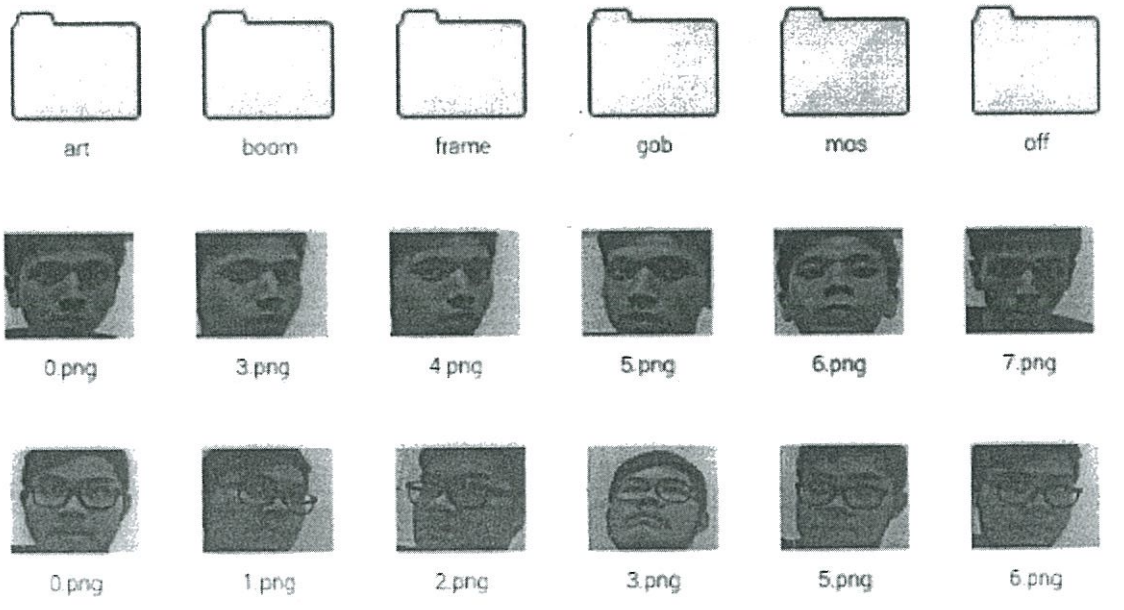


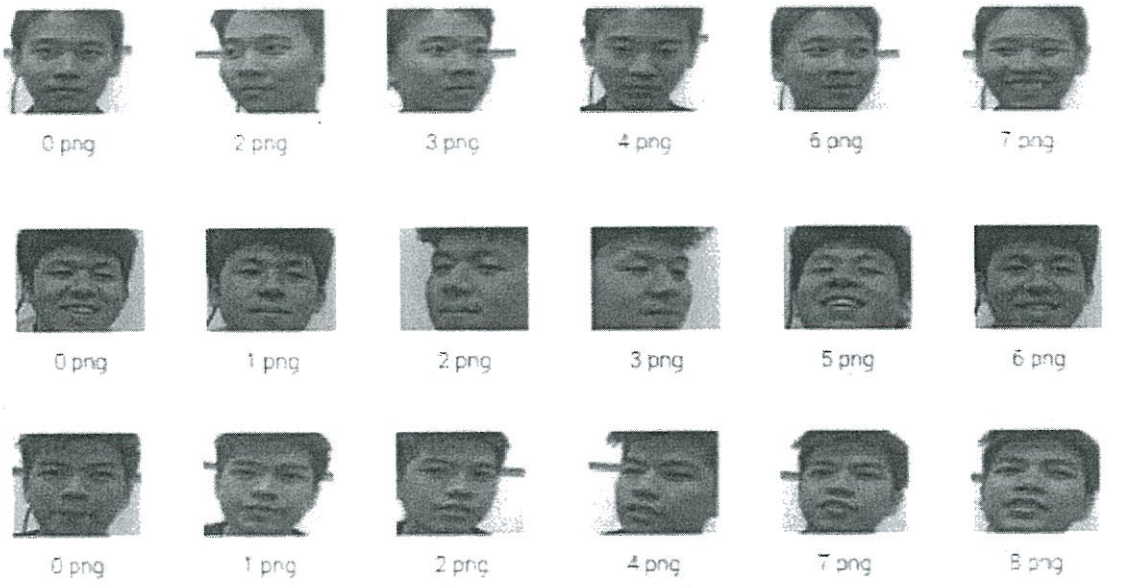
รูปที่ 4.6 ผู้ใช้งานบุคคลที่ 4 มีใบหน้าไม่ตรงกับฐานข้อมูล



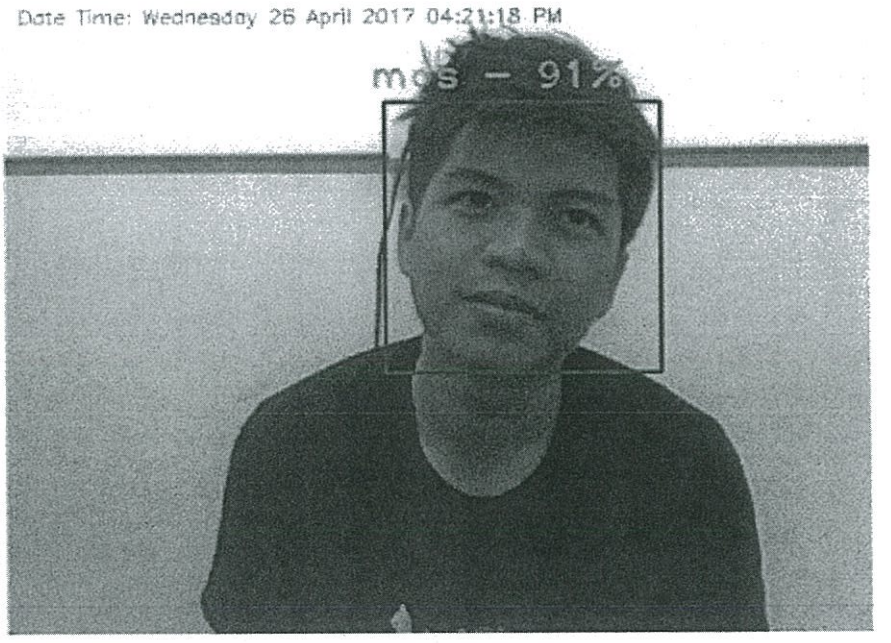
รูปที่ 4.7 ผู้ใช้งานบุคคลที่ 5 มีใบหน้าไม่ตรงกับฐานข้อมูล

จากนั้นจะทำการบันทึกข้อมูลไฟล์ภาพผู้ใช้งานเหล่านี้ด้วยการถ่ายภาพเก็บไว้ในไฟล์ฐานข้อมูลเป็นจำนวนหนึ่งโดยใช้คำสั่งโปรแกรมถ่ายภาพและบันทึกรูปหน้า โดยสังเกตเห็นว่าหลังจากบันทึกฐานข้อมูลแล้ว สามารถตรวจจับใบหน้านั้น ๆ เทียบกับฐานข้อมูลและแสดงชื่อของบุคคลนั้น ๆ ตามชื่อไฟล์ที่บันทึกได้อีกด้วย

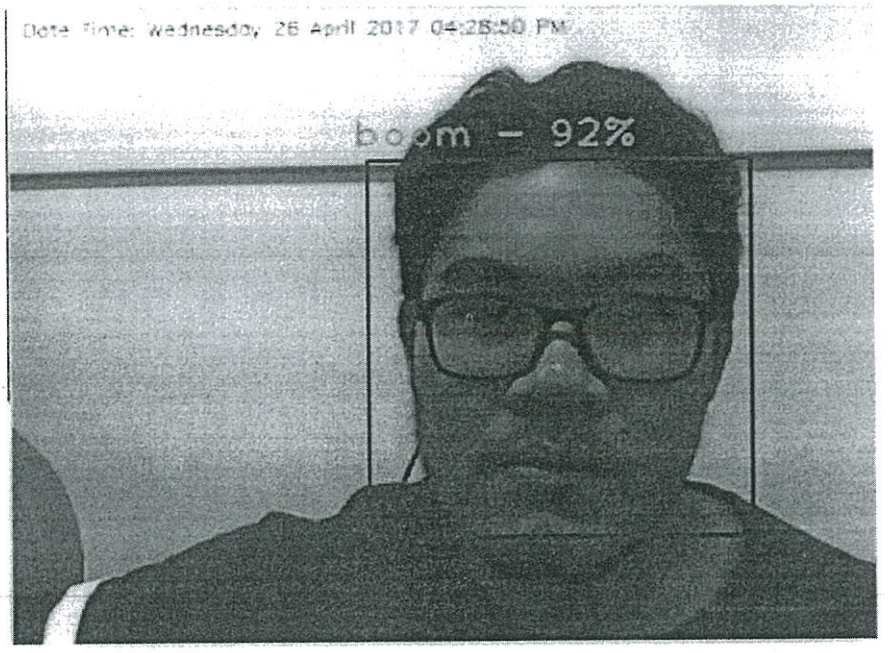




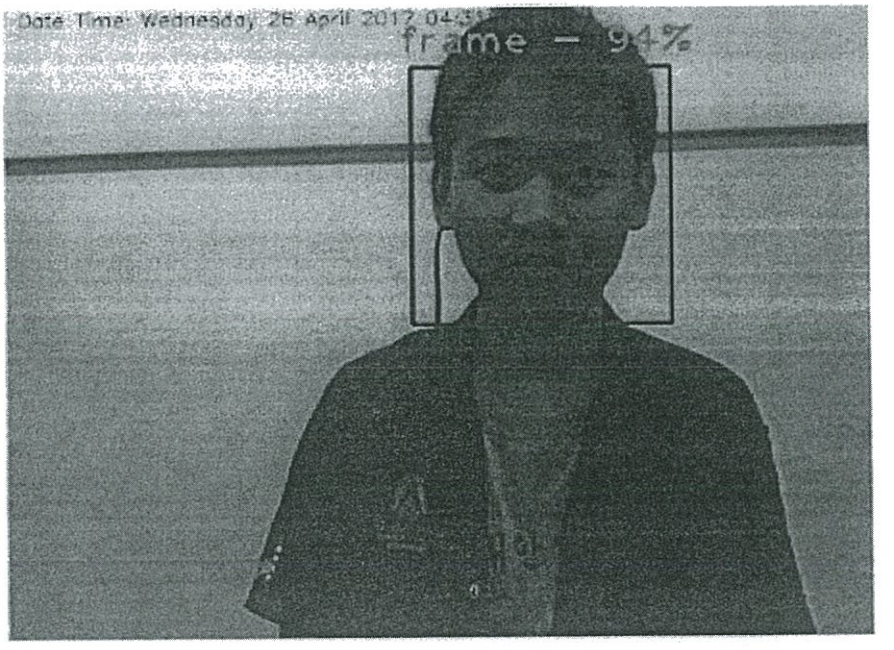
รูปที่ 4.8 ไฟล์ภาพฐานข้อมูล



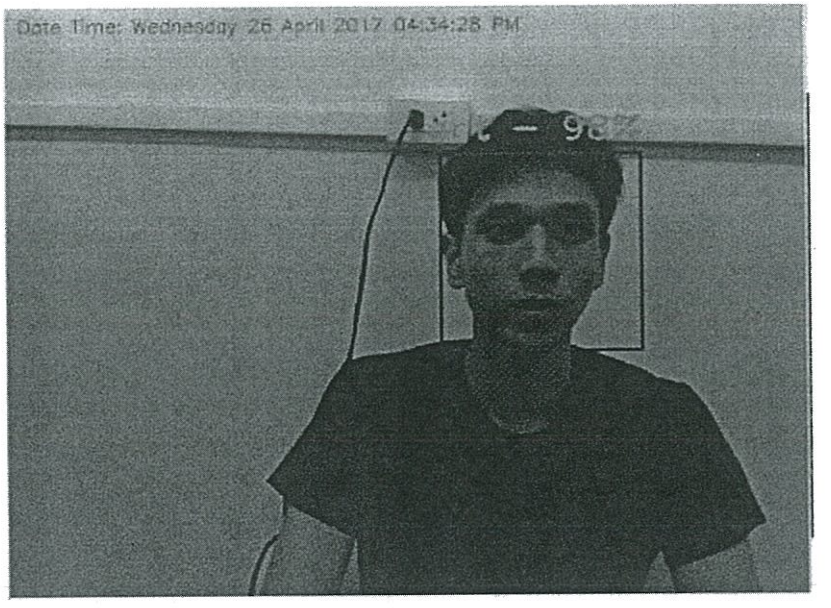
รูปที่ 4.9 ผู้ใช้งานบุคคลที่ 1 มีใบหน้าตรงกับฐานข้อมูล



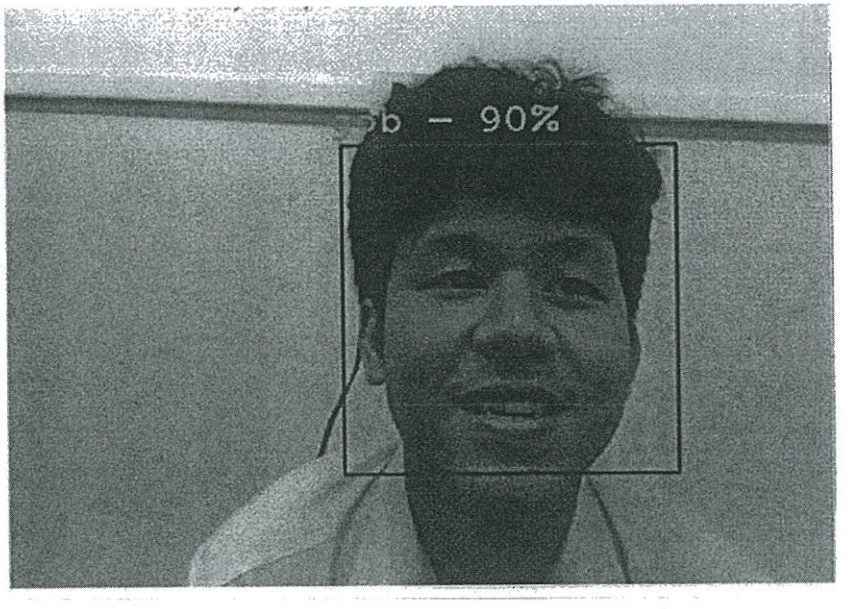
รูปที่ 4.10 ผู้ใช้งานบุคคลที่ 2 มีใบหน้าตรงกับฐานข้อมูล



รูปที่ 4.11 ผู้ใช้งานบุคคลที่ 3 มีใบหน้าตรงกับฐานข้อมูล



รูปที่ 4.12 ผู้ใช้งานบุคคลที่ 4 มีใบหน้าตรงกับฐานข้อมูล

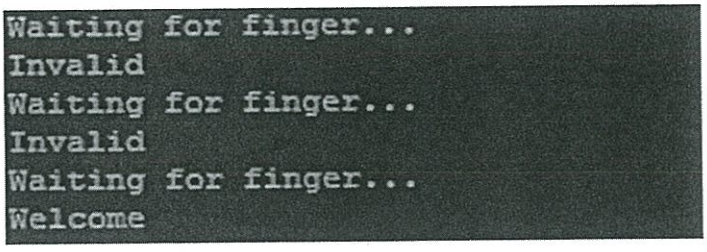


รูปที่ 4.13 ผู้ใช้งานบุคคลที่ 5 มีใบหน้าตรงกับฐานข้อมูล

ปัจจัยสำคัญในการตรวจจับใบหน้า คือ แสง ถ้าหากผู้ใช้งานนำอุปกรณ์ไปใช้งานในที่ที่มีแสงแตกต่างกับแสงในภาพที่บันทึกไว้ จะทำให้ยากต่อการตรวจจับ

4.2 ผลจากการอ่านข้อมูลลายนิ้วมือที่เก็บไว้ใน Raspberry Pi

จากการทดลองการสแกนลายนิ้วมือจะเห็นผลเป็นดังรูปคือ เมื่อมีผู้ใช้ที่ไม่มีข้อมูลลายนิ้วมือในฐานข้อมูล จะแสดงผลออกมาเป็น “Invalid” ในทางกลับกัน เมื่อมีผู้ใช้ที่มีข้อมูลลายนิ้วมือในฐานข้อมูล จะแสดงผลออกมาเป็น “Welcome”



รูปที่ 4.14 ผลจากการสแกนลายนิ้วมือ

ขั้นตอนการทดลองการสแกนนิ้วคือใช้นิ้วที่มีในฐานข้อมูลหรือไม่มีในฐานข้อมูล สแกนนิ้วละห้าครั้ง ปรากฏว่ามีจำนวนของความถูกต้องเท่ากับ 5 ครั้ง ของทุกนิ้วในแต่ละผู้ใช้งาน โดยผลการสแกนมีความแม่นยำสูงถูกต้องทุกครั้ง

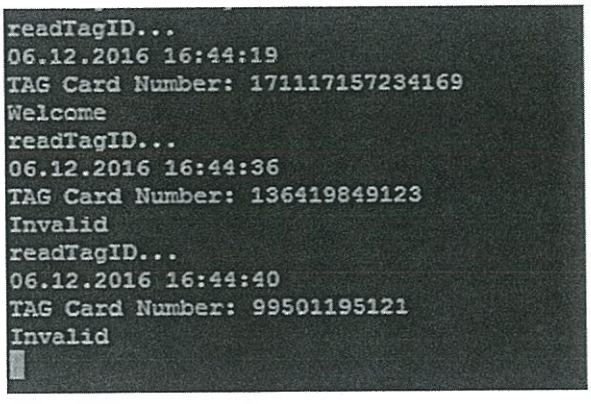
ตารางที่ 4.1 ผลการทดลองหาความแม่นยำของการสแกนนิ้ว

นิ้ว / บุคคลที่	โป้ง (ครั้ง)	ชี้ (ครั้ง)	กลาง (ครั้ง)	นาง (ครั้ง)	ก้อย (ครั้ง)	เปอร์เซ็นต์ความถูกต้อง
1	5	5	5	5	5	100%
2	5	5	5	5	5	100%
3	5	5	5	5	5	100%

4.3 ผลจากการอ่านข้อมูลการ์ด RFID ที่เก็บไว้ใน Raspberry Pi

จากการทดลองการสแกนบัตรจะเห็นผลเป็นดังรูปคือ เมื่อมีผู้ใช้ที่ไม่มีข้อมูลของบัตรนักศึกษาหรือบัตร RFID อื่น ๆ ในฐานข้อมูล จะแสดงผลออกมาเป็น “Invalid” พร้อมบอกเลข

ประจำตัวของบัตรนั้น ๆ ในทางกลับกัน เมื่อมีผู้ใช้ที่มีข้อมูลของบัตร RFID ในฐานข้อมูล จะแสดงผลออกมาเป็น “Welcome” พร้อมบอกเลขประจำตัวของบัตรนั้น ๆ ด้วยเช่นกัน



รูปที่ 4.15 ผลจากการสแกนบัตร RFID

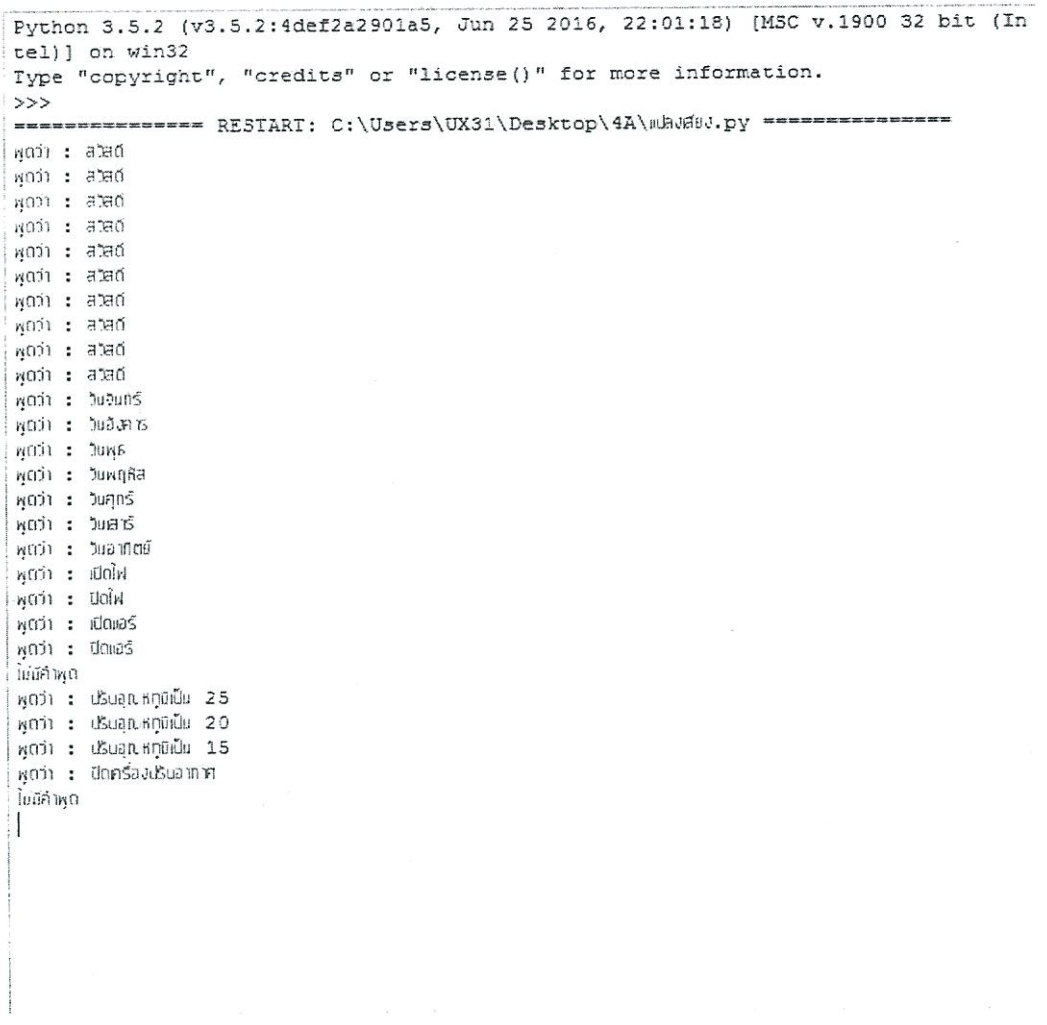
ขั้นตอนการทดลองการสแกนบัตร RFID โดยใช้บัตรนักศึกษา ทดลองบัตรละห้าครั้ง ผลการทดลองเป็นดังตาราง

ตารางที่ 4.2 ผลการทดลองหาความแม่นยำของการสแกนบัตร RFID

บัตรที่	จำนวน (ครั้ง)	เปอร์เซ็นต์ความถูกต้อง
1	5	100%
2	5	100%
3	5	100%

4.4 ผลจากการแปลงเสียงพูดเป็นข้อความ

จากการทดลองจะรับคำสั่งเสียงจากไมโครโฟน แล้วเปลี่ยนเป็นข้อความ โดยได้ทำการทดลองตรวจความถูกต้องของข้อความ ผลที่ได้เป็นดังรูปที่ 4.16



รูปที่ 4.16 การทดลองเปลี่ยนเสียงพูดเป็นข้อความ

ตารางถัดไปเป็นการตรวจสอบความถูกต้องเพื่อหาเปอร์เซ็นต์ความแม่นยำความถูกต้องโดยการแปลงเสียงพูดเป็นข้อความ ในคำสั่งการใช้งานต่าง ๆ โดยพูดคำนั้น ๆ 10 ครั้ง ในห้องที่ไม่มีเสียงรบกวน

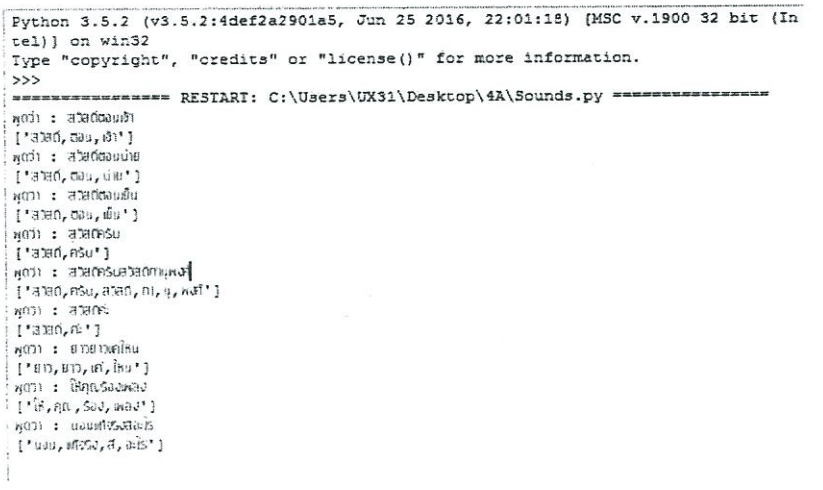
ตารางที่ 4.3 ผลการทดลองหาความแม่นยำของการแปลงเสียงเป็นข้อความ

คำสั่ง	จำนวนคำถูกต้อง	จำนวนคำผิด	เปอร์เซ็นต์ความถูกต้อง
เปิดไฟ	10	0	100%
ปิดไฟ	9	1	90%
เปิดแอร์	7	3	70%
ปิดแอร์	8	2	80%
เปิดเพลง	10	0	100%
ปิดเพลง	9	1	90%

จากการทดลองจะได้เปอร์เซ็นต์ความถูกต้องรวมประมาน 88 % โดยเปอร์เซ็นต์ความแม่นยำนั้นขึ้นอยู่กับหลายปัจจัย เช่น เสียงรบกวนอื่น ๆ ภายในห้อง, การเชื่อมต่ออินเทอร์เน็ต, ข้อความเสียงที่ยาวเกินไป เป็นต้น ส่งผลให้ค่าที่ได้ผิดเพี้ยนไป

4.5 ผลจากการแบ่งข้อความที่ได้รับเป็นคำ

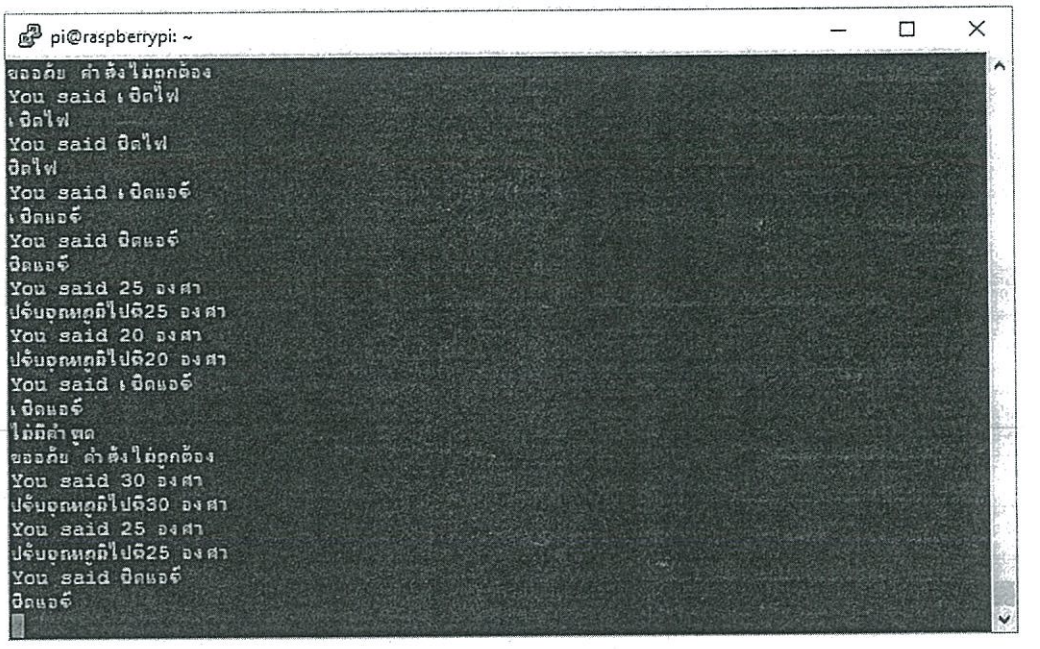
จากการทดลองเป็นการแบ่งคำออกจากข้อความเพื่อใช้ในการตรวจสอบหาคีย์เวิร์ดในการสั่งงาน



รูปที่ 4.17 การแยกคำจากข้อความ

4.6 ผลจากการทดลองนำข้อความที่ได้ไปควบคุมอุปกรณ์

คำสั่งต่าง ๆ ที่ใช้เป็นคีย์เวิร์ดเพื่อเปรียบเทียบ และนำไปสั่งงานมีคำสั่งโดยประมาณดังนี้



รูปที่ 4.18 การสั่งงานอุปกรณ์

ในส่วนของการเก็บข้อมูลสัญญาณอินฟราเรดนั้น จะเก็บโดยการบันทึกค่าคำสั่งต่าง ๆ จากรีโมทของเครื่องปรับอากาศนั้น ๆ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงงานนี้เป็นการนำเทคโนโลยีการรักษาความปลอดภัยทั้งสามรูปแบบมาใช้งานร่วมกันบนบอร์ด Raspberry Pi โดยมีระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า โดยสามารถที่จะตรวจจับภาพได้ว่าในภาพนั้นมีใบหน้าของบุคคลอยู่หรือไม่และเมื่อสามารถตรวจจับได้ว่ามีใบหน้าของบุคคลอยู่ก็จะทำการแสดงกรอบขึ้นมาครอบใบหน้านั้น ถ้าใบหน้าของบุคคลนั้นตรงกับใบหน้าที่มีอยู่ในฐานข้อมูล ก็จะแสดงชื่อของบุคคลนั้นขึ้นมา, ระบบรักษาความปลอดภัยโดยการสแกนลายนิ้วมือ จะใช้นิ้วมือที่เก็บข้อมูลในระบบไว้ในการใช้งานการสแกน และระบบรักษาความปลอดภัยโดยใช้เทคโนโลยี RFID ที่จะใช้บัตรนักศึกษาหรือการ์ดที่มีความถี่ตรงกับโมดูล RFID และมีการเก็บข้อมูลการ์ดในระบบเพื่อใช้ในการสแกน ทั้งสามระบบนี้จะทำงานในเวลาเดียวกัน โดยถ้าผู้ใช้งานมีข้อมูลอยู่ในระบบ สามารถเลือกใช้ระบบใดระบบหนึ่งเพื่อยืนยันตัวบุคคลในการผ่านเข้าประตูที่มีกลอนไฟฟ้าได้ ในส่วนของระบบสั่งงานด้วยเสียงภายในห้องนั้น รูปแบบการทำงานคือมีการรับอินพุตที่เป็นเสียงจากผู้ใช้งาน โดยเริ่มต้นก่อนที่จะสั่งงานต้องเรียกหาคีย์เวิร์ดก่อนที่จะใช้คำสั่งเสียงในการสั่งงาน จากนั้นเมื่อพูดคำสั่งงานด้วยเสียง ระบบจะคัดแยกคำออกจากข้อความเพื่อนำคำเหล่านั้นมาวิเคราะห์เทียบกับคำที่มีอยู่ในระบบและสั่งงานต่อไป ในส่วนของการควบคุมหลอดไฟจะใช้รีเลย์และจะใช้อินฟราเรดในการควบคุมเครื่องปรับอากาศ

5.2 ปัญหาและอุปสรรค

1. ปัญหาของระบบตรวจจับใบหน้าคือ การจดจำใบหน้านั้นอาจจะมีการคลาดเคลื่อน ซึ่งขึ้นอยู่กับปัจจัยหลายอย่าง เช่น มีแสงน้อยเกินไปจึงทำให้ภาพไม่ชัดและอาจจะระบุผู้ใช้งานผิดพลาดได้

2. ปัญหาเกิดจากระบบแปลงเสียงเป็นข้อความ จากการทดลองมีโอกาสผิดพลาดประมาณ 12% และต้องอยู่ในห้องที่ไม่มีเสียงรบกวน

5.3 ข้อเสนอแนะ

1. ในขณะที่ทำการตรวจจับใบหน้า บุคคลนั้นควรที่จะยืนหันหน้าตรงให้ใกล้เคียงกับภาพที่เก็บไว้ในฐานข้อมูลให้มากที่สุด และควรให้มีแสงสว่างที่มากพอ เพื่อเพิ่มความแม่นยำของการตรวจจับและรู้จำใบหน้ามากยิ่งขึ้น

2. ระบบสั่งการด้วยเสียงเหมาะกับการใช้งานในห้องส่วนตัว เพราะควรจะใช้งานในห้องที่ไม่มีเสียงรบกวนมากนัก

บรรณานุกรม

[1] "Accessing the Raspberry Pi Camera with OpenCV and Python."Adrian Rosebrock.
http://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/

[2] "Install guide: Raspberry Pi 3 + Raspbian Jessie + OpenCV 3." Adrian Rosebrock.
http://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/

[3] "How to install OpenCV 3 on Raspbian Jessie." Adrian Rosebrock.
http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/

[4] "How to install OpenCV on Raspberry Pi and do Face Tracking." Nitin Patil.
http://www.mindsensors.com/blog/how-to/how-to-install-opencv-on-raspberry-pi-and-do-face-tracking

[5] "Python library for ZFM-20 fingerprint sensor."
https://github.com/bastianraschke/pyfingerprint

[6] "Unable to integrate Fingerprint sensor in Raspberry Pi2." Sahil Arora.
http://raspberrypi.stackexchange.com/questions/46943/unable-to-integratefingerprint-sensor-in-raspberry-pi2

[7] "Face Recognition in Videos with OpenCV."
http://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html

[9] "Speech recognition."
https://python3.wannaphong.com/2014/12/speech-recognition

[10] "Raspberry pi infrared remote."
http://osoyoo.com/2016/12/15/raspberrypi-ir-remote/

ภาคผนวก ก

ชุดคำสั่งของระบบตรวจจับใบหน้า

```
import cv2, sys, numpy, os
import time
import RPi.GPIO as GPIO # Import GPIO library
from picamera.array import PiRGBArray
from picamera import PiCamera
import datetime
from pygame import mixer

def voice(a):
    mixer.init()
    os.chdir('/home/pi/')
    mixer.music.load(a+str(".mp3"))
    mixer.music.play()

def doorUnlock():
    GPIO.output(20,GPIO.LOW)
    GPIO.output(21,GPIO.HIGH)
    GPIO.output(20,GPIO.HIGH)
    voice("ยินดีต้อนรับ")
    time.sleep(3.0)
    GPIO.output(21,GPIO.LOW)
    GPIO.output(20,GPIO.LOW)

camera = PiCamera()

camera.resolution = (640, 480)

camera.framerate = 32

rawCapture = PiRGBArray(camera, size=(640, 480))
time.sleep(0.1)
size = 4
fn_haar = 'haarcascade_frontalface_default.xml'
fn_dir = 'database2'
a=0
b=0
q = 0
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.OUT)
GPIO.setup(20, GPIO.OUT)
GPIO.setup(26, GPIO.IN)
GPIO.setup(19, GPIO.IN)

# Create fisherRecognizer
print('Waiting...')
```

```
# Create a list of images and a list of corresponding names
(images, lables, names, id) = ([], [], {}, 0)

for (subdirs, dirs, files) in os.walk(fn_dir):

        for subdir in dirs:

                names[id] = subdir

                subjectpath = os.path.join(fn_dir, subdir)

                for filename in os.listdir(subjectpath):

                        path = subjectpath + '/' + filename

                        lable = id

                        images.append(cv2.imread(path, 0))

                        lables.append(int(lable))

                id += 1
(im_width, im_height) = (112, 92)


# Create a Numpy array from the two lists above
(images, lables) = [numpy.array(lis) for lis in [images,
lables]]


# OpenCV trains a model from the images
model = cv2.createFisherFaceRecognizer()
model.train(images, lables)


# Use fisherRecognizer on camera stream
haar_cascade = cv2.CascadeClassifier(fn_haar)

aa = 0
while True:

# capture frames from the camera

        for frame in camera.capture_continuous(rawCapture,
 format="bgr", use_video_port=True):

                image = frame.array
                timestamp = datetime.datetime.now()
```

```
                ts = timestamp.strftime("%A %d %B %Y %I:%M:%S
%p")
                ts2 = timestamp.strftime("%d %B %Y %I:%M:%S
%p")
                cv2.putText (image, "Date Time:
{}".format(ts), (10, 20),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,
0, 255), 1)

# clear the stream in preparation for the next frame

                rawCapture.truncate(0)



                gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

                mini = cv2.resize(gray, (gray.shape[1] /
size, gray.shape[0] / size))

                faces = haar_cascade.detectMultiScale(mini)



                for i in range(len(faces)):

                        face_i = faces[i]

                        (x, y, w, h) = [v * size for v in
face_i]

                        face = gray[y:y + h, x:x + w]

                        face_resize = cv2.resize(face,
(im_width, im_height))
# Try to recognize the face

                        prediction =
model.predict(face_resize)
                        accuracy = 100*((1500-
prediction[1])/1500)

                cv2.rectangle(image, (x, y), (x + w, y + h),
(0, 255, 0), 1)

                        if q == 1:
                            q = 0
                            time.sleep(1.0)
                            doorUnlock()
```

```
                        time.sleep(2.0)

            if prediction[1]<450:
                            cv2.rectangle(image, (x, y),
(x + w, y + h), (255, 0, 0), 2)
                            cv2.putText(image,'%s -
%.0f%s' % (names[prediction[0]],accuracy,"%"),(x-10,y-10),
cv2.FONT_HERSHEY_PLAIN,2, (0, 255, 0),2)

                            q = 1

                    else :
                            cv2.rectangle(image, (x, y),
(x + w, y + h), (0,0,255), 2)

cv2.putText(image,'Unknown',(x-10, y-10),
cv2.FONT_HERSHEY_PLAIN,2,(0, 0, 255),2)


          if (GPIO.input(26) == True) :
                    a0 = '0'
                    a1 = raw_input("Name : ")
                    save = a0 + str(".png")

os.makedirs('/home/pi/database2/%s'%a1)
                    os.chdir('/home/pi/database2/%s'%a1)
                    cv2.imwrite(save,face_resize)
                    a0 = int(a0)+1
                    print("capture")


                if (GPIO.input(19) == True):

                        save = str(a0) + str(".png")
                        os.chdir('/home/pi/database2/%s'%a1)
                        cv2.imwrite(save,face_resize)
                        a0 = int(a0)+1
                        print("capture")



                cv2.imshow('FaceDetector', image)

                key = cv2.waitKey(10)

                if key == 27:

                        break
```

ภาคผนวก ข

ชุดคำสั่งของระบบสแกนลายนิ้วมือและระบบ RFID

```
from multiprocessing import Process
from pyfingerprint.pyfingerprint import PyFingerprint
from decimal import Decimal
from pygame import mixer
import mysql
import nfc
import time
import RPi.GPIO as GPIO
import enroll
import delete

MASTER_TAG = "571904420396"
a = 0
a1 = 26
b1 = 19
c1 = 13
d10 = 0
d9 = 0
def readTagID():
    cardId=nfc.readNfc()
    print("readTagID...")
    return cardId

def doorlock():
    GPIO.output(21,GPIO.LOW)
    GPIO.output(20,GPIO.LOW)

def doorUnlock():
    GPIO.output(21,GPIO.HIGH)
    GPIO.output(20,GPIO.HIGH)
    voice("ยินดีต้อนรับ")
    time.sleep(2.0)
    GPIO.output(21,GPIO.LOW)
    GPIO.output(20,GPIO.LOW)

def initGpio():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(20, GPIO.OUT)
    GPIO.setup(a1, GPIO.IN)
    GPIO.setup(b1, GPIO.IN)
    GPIO.setup(c1, GPIO.IN)
    GPIO.setup(21, GPIO.OUT)

def voice(a):
    mixer.init()
```

```
    mixer.music.load(a+str(".mp3"))
    mixer.music.play()

def RFID():
    GPIO.cleanup()
    try:
        initGpio()
        doorlock()
        while True:
            cardId2 = readTagID()
            print (time.strftime("%d.%m.%Y %H:%M:%S",
time.localtime()))
            print ("TAG Card Number: "+ str(cardId2))
            if cardId2 == MASTER_TAG:
                print ("Master")
                voice("กดปุ่มเอ เพื่อบันทึกการ์ด")
                time.sleep(3.2)
                voice("บีเพื่อยกเลิกการ์ด")
                time.sleep(3.0)
                voice("ซียกเลิก")
                time.sleep(2.0)
                a = 0
                while a == 0 :

                    if(GPIO.input(a1) == True):
                        time.sleep(0.5)
                        voice("กรุณาแตะการ์ดที่ต้องการบันทึก")
                        cardId2 = readTagID()
                        mysql.AddTag(cardId2)
                        a = 1
                        time.sleep(2.0)

                    elif(GPIO.input(c1) == True):
                        voice("ยกเลิก")
                        a = 1
                        time.sleep(2.0)

                    elif(GPIO.input(b1) == True):
                        voice("กดปุ่มเอเพื่อเลือกเลขหลักสิบ")
                        time.sleep(3.0)
                        voice("บีเพื่อเลือกเลขหลักหน่วย")
                        time.sleep(2.0)
                        voice("จากนั้นกดปุ่มซี")
                        d10 = 10
                        d9 = 10
                        while a==0:
                            if(GPIO.input(a1) == True):
                                time.sleep(0.5)
```

```
                                    if d10 > 8 :
                                        d10 = 0
                                    else:
                                        d10 +=1
                                    voice(str(d10))
                                elif(GPIO.input(b1) == True):
                                    time.sleep(0.5)
                                    if d9 > 8 :
                                        d9 = 0
                                    else:
                                        d9 +=1
                                    voice(str(d9))
                                elif(GPIO.input(c1) == True):
                                    time.sleep(0.5)
                                    a = 1

                            mysql.DeleteTag(str(d10)+str(d9))
                            d10 = 0
                            d9 = 0
                            pass

                else:
                    StatusTag = mysql.ReadTagID(cardId2)
                    if StatusTag != 0:
                        print("Welcome")
                        mysql.log(cardId2)
                        doorUnlock();
                    else:
                        print("Invalid")
                        voice("ไม่พบในฐานข้อมูล")
                        time.sleep(3.0)
        except KeyboardInterrupt:
            GPIO.cleanup()
            pass


def finger():
        try:
            f = PyFingerprint('/dev/ttyUSB0', 57600,
0xFFFFFFFF, 0x00000000)

            if ( f.verifyPassword() == False ):
                raise ValueError('The given fingerprint
sensor password is wrong!')

        except Exception as e:
            print('The fingerprint sensor could not be
initialized!')
            print('Exception message: ' + str(e))
```

```
            exit(1)

        print('Currently used templates: ' +
str(f.getTemplateCount()) +'/'+ str(f.getStorageCapacity()))
        try:
            initGpio()
            doorlock()
            while True:
                print('Waiting for finger...')

                while ( f.readImage() == False ):
                    pass

                f.convertImage(0x01)

                result = f.searchTemplate()
                positionNumber = result[0]

                if( positionNumber == 0):
                    print ("Master")
                    voice("กดปุ่มเอ เพื่อบันทึกลายนิ้วมือ")
                    time.sleep(3.2)
                    voice("บีเพื่อยกเลิกลายนิ้วมือ")
                    time.sleep(3.0)
                    voice("ซียกเลิก")
                    time.sleep(2.0)
                    a = 0
                    while a == 0 :
                        if(GPIO.input(a1) == True):
                            time.sleep(0.5)
                            enroll.en()
                            a = 1
                            time.sleep(2.0)

                        elif(GPIO.input(c1) == True):
                            time.sleep(0.5)
                            voice("ยกเลิก")
                            a = 1
                            time.sleep(2.0)

                        elif(GPIO.input(b1) == True):
                            voice("กดปุ่มเอเพื่อเลือกเลขหลักสิบ")
                            time.sleep(3.5)
                            voice("บีเพื่อเลือกเลขหลักหน่วย")
                            time.sleep(2.0)
                            voice("จากนั้นกดปุ่มซี")
                            d10 = 10
                            d9 = 10
```

```
                                while a==0:
                                    if(GPIO.input(a1) == True):
                                        time.sleep(0.5)
                                        if d10 > 8 :
                                            d10 = 0
                                        else:
                                            d10 +=1
                                        voice(str(d10))
                                    elif(GPIO.input(b1) == True):
                                        time.sleep(0.5)
                                        if d9 > 8 :
                                            d9 = 0
                                        else:
                                            d9 +=1
                                        voice(str(d9))
                                    elif(GPIO.input(c1) == True):
                                        time.sleep(0.5)
                                        a = 1
                                d10 = d10*10
                                post = d10 + d9
                                delete.dlf(post)
                                mysql.Deletefin(str(d10)+str(d9))
                                d10 = 0
                                d9 = 0
                                pass

                        elif( positionNumber >=1):
                            print("Welcome")
                            doorUnlock()
                            mysql.logf(positionNumber)

                        else:
                            print("Invalid")
                            voice("ไม่พบในฐานข้อมูล")

                except Exception as e:
                    print('Operation failed!')
                    print('Exception message: ' + str(e))
                    exit(1)
                except KeyboardInterrupt:
                    GPIO.cleanup()
                    Pass
if __name__ == '__main__':
    p = Process(target=RFID)
    q = Process(target=finger)
    p.start()
    q.start()
    p.join()
    q.join()
```

ภาคผนวก ค

ชุดคำสั่งของระบบสั่งงานด้วยเสียง

```
import PyICU
import re
import time
import speech_recognition as sr
import RPi.GPIO as GPIO
import os
import tkinter as tk
from functools import partial
from tkinter import*
import tkinter
import pygame
from tkinter.filedialog import askdirectory

root = Tk()
root.title("Music Player")
root.resizable(0,0)


listmusic = []
#realnames = []

v = StringVar()
songlabel = Label(root,textvariable=v,width=35)
index = 0

pr = 0
vol = 1
def nextsong():
    global index
    index += 1
    pygame.mixer.music.load(listmusic[index])
    pygame.mixer.music.play()
    updatelabel()

def Volume(a):
    pygame.mixer.music.set_volume(a)

def prevsong():
    global index
    index -= 1
    pygame.mixer.music.load(listmusic[index])
    pygame.mixer.music.play()
    updatelabel()

def stopsong():
    pygame.mixer.music.stop()
```

```
    v.set("")
    updatelabel()

def playsong():
    pygame.mixer.music.play()
    updatelabel()

def updatelabel():
    global index
    #global songname
    v.set(listmusic[index])
    return index

def directorychooser():
    directory = "/home/pi/เพลง"
    os.chdir(directory)

    for files in os.listdir(directory):
        if files.endswith(".mp3"):
            ##realdir = os.path.realpath(files)
            ##audio = ID3(realdir)
            ##realnames.append(audio['TIT2'].text[0])
            listmusic.append(files)

    pygame.mixer.init()
    pygame.mixer.music.load(listmusic[0])
    pygame.mixer.music.play()

def initGpio():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(16, GPIO.OUT)

def turnoff():
    GPIO.output(16,GPIO.LOW)

def turnon():
    GPIO.output(16,GPIO.HIGH)


def isThai(chr):
    cVal = ord(chr)
    if(cVal >= 3584 and cVal <= 3711):
        return True
    return False
def warp(txt):
    bd =
PyICU.BreakIterator.createWordInstance(PyICU.Locale("th"))
    bd.setText(txt)
```

```
    lastPos = bd.first()
    retTxt = ""
    try:
        while(1):
            currentPos = next(bd)
            retTxt += txt[lastPos:currentPos]

            if(isThai(txt[currentPos-1])):
                if(currentPos < len(txt)):
                    if(isThai(txt[currentPos])):

                        retTxt += ','
            lastPos = currentPos
    except StopIteration:
        pass
        #retTxt = retTxt[:-1]
    return retTxt

print("สวัสดี")
b = 0
initGpio()
while(True):
    a = ""
    while (b == 1):
        r = sr.Recognizer()
        with sr.Microphone() as source:
           audio = r.listen(source)
        try:
            a = r.recognize_google(audio,language = "th-TH")
            print("You said " + a)
            if a == "คอมพิวเตอร์":
                print("โปรดออกคำสั่ง")
                b = 1
        except:
           print("ไม่มีใครพูด")

    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.listen(source)
    try:
        a = r.recognize_google(audio,language = "th-TH")
        print("You said " + a)
    except:
        print("ไม่มีคำพูด")

    txt = [warp(a)]
    #print(txt)
    txt = (','.join(str(x) for x in txt))
```

```
    n = 0
    nn = 0
    for n in range(1000) :
        if txt.find(str(n)) != -1 :
            nn = n
        n = int(n)+1
    if txt.find("สวัสดี") != -1 or txt.find("ไง") != -1 or
txt.find("ฮัลโหล") != -1 :
        print("สวัสดี")
        b = 0
    elif txt.find("ไฟ")!= -1 or txt.find("สว่าง")!= -1 or
txt.find("มืด")!= -1:
        if txt.find("เปิด")!= -1 or txt.find("สว่าง")!= -1 or
txt.find("ติด")!= -1:
            print("เปิดไฟ")
            turnon()
            b = 0
        elif txt.find("ปิด")!= -1 or txt.find("ดับ")!= -1 or
txt.find("มืด")!= -1:
            print("ปิดไฟ")
            turnoff()
            b = 0
        else:
            print("ขออภัย คำสั่งไม่ถูกต้อง")
            b = 0
    elif txt.find("แอร์")!= -1 or txt.find("เครื่อง")!= -1 and
txt.find("ปรับ")!= -1 and txt.find("อากาศ")!= -1 or txt.find("
อุณหภูมิ")!= -1 or txt.find("ร้อน")!= -1 or txt.find("เย็น")!= -1 or
txt.find("องศา")!= -1 :
        if txt.find("เปิด")!= -1 or txt.find("ติด")!= -1:
            print("เปิดแอร์")
            os.system("irsend SEND_ONCE Air On")
            b = 0
        elif txt.find("ปิด")!= -1 or txt.find("ดับ")!= -1:
            os.system("irsend SEND_ONCE Air Off")
            print("ปิดแอร์")
            b = 0
        elif txt.find("เพิ่ม")!= -1 or txt.find("ร้อน")!= -1 or
txt.find("ขึ้น")!= -1 :
            if int(nn) > 0 :
                print("เพิ่มอุณหภูมิ " + str(nn)  +" องศา")
                b = 0
            else :
                print("เพิ่มอุณหภูมิ 1 องศา")
                b = 0
```

```
            if int(nn) > 0 :
                print("ลดอุณหภูมิ " + str(nn)  +" องศา")
                b = 0
            else :
                 print("ลดอุณหภูมิ 1 องศา")
                 b = 0
        elif txt.find("อุณหภูมิ")!= -1 or txt.find("ปรับ")!= -1 or
txt.find("องศา")!= -1 :
            print("ปรับอุณหภูมิไปที่" + str(nn)  +" องศา")
            os.system("irsend SEND_ONCE Air KEY_" + str(nn))
            b = 0
        else:
            print("ขออภัย คำสั่งไม่ถูกต้อง")
            b = 0


    elif txt.find("เพลง")!= -1 or txt.find("เล่น")!= -1 or
txt.find("เสียง")!= -1 or txt.find("ดัง")!= -1 or txt.find("เบา")!=
-1 or txt.find("พอส")!= -1 or txt.find("หยุด")!= -1   :
        if txt.find("ที่")!= -1 and txt.find("แล้ว")!= -1 or
txt.find("ก่อน")!= -1 and txt.find("หน้า")!= -1 or txt.find("
เมื่อ")!= -1 and txt.find("กี้")!= -1:
            if pr == 1:
                print("เล่นเพลงที่แล้ว")
                prevsong()
                b = 0
        elif txt.find("ต่อ")!= -1 and txt.find("ไป")!= -1 or
txt.find("ถัด")!= -1 and txt.find("ไป")!= -1 or txt.find("หน้า")!=
-1 :
            if pr == 1:
                print("เล่นพลงถัดไป")
                nextsong()
                b = 0

        elif txt.find("เปิด")!= -1 and txt.find("เพลง")!= -1 or
txt.find("เล่น")!= -1 or txt.find("เริ่ม")!= -1 :
            if pr == 0:
                directorychooser()
                pr = 1
                print("เล่นเพลง")
                label = Label(text='Music Player naja')
                label.pack()

                listbox = Listbox(root)
                listbox.pack()

                listmusic.reverse()
```

```
                #realnames.reverse()

                for items in listmusic:
                    listbox.insert(0,items)

                #realnames.reverse()
                listmusic.reverse()

                butplay = Button(root,text= 'Play Song')
                butplay.pack()
                butnex = Button(root,text = 'Next Song')
                butnex.pack()
                butprev = Button(root,text = 'Previous Song')
                butprev.pack()
                butstop = Button(root,text = 'Stop Song')
                butstop.pack()
                s1=Scale(orient=HORIZONTAL)
                s1.pack()

                songlabel.pack()
                b = 0

            else :
                print("เล่นเพลง")
                playsong()

        elif txt.find("ปิด")!= -1 and txt.find("เพลง")!= -1 or
txt.find("หยุด")!= -1 or txt.find("หยุด")!= -1 and txt.find("
เล่น")!= -1 or txt.find("พอส")!= -1 :
            if pr == 1:
                print("หยุดเล่นเพลง")
                stopsong()
                b = 0
        elif txt.find("เพิ่ม")!= -1 or txt.find("ดัง")!= -1 or
txt.find("ขึ้น")!= -1:
            print("เพิ่มเสียงขึ้นนิดนึง")
            if vol < 1 :
                vol = vol+0.2
                if vol < 1 :
                    Volume(vol)
                else :
                    Volume(1)

                b = 0
        elif txt.find("ลด")!= -1 or txt.find("เบา")!= -1 :
            print("ลดเสียงลงนิดนึง")
            if vol <= 1 :
                vol = vol-0.2
```

```
                if vol > 0 :
                    Volume(vol)
                else :
                    Volume(0)
                b = 0

        else:
            print("ขออภัย คำสั่งไม่ถูกต้อง")
            b = 0

    elif txt.find("เวลา") != -1 or txt.find("โมง") != -1 or
txt.find("ตอน") != -1 and txt.find("นี้") != -1 :
        a = time.strftime("%H - %M :")
        a = a.replace('-', 'นาฬิกา')
        a = a.replace(':', 'นาที')
        print("ขณะนี้เป็นเวลา %s ค่ะ" % a)
        b = 0
    else:
        print("ขออภัย คำสั่งไม่ถูกต้อง")
        b = 0
```