

ระบบตรวจสอบคนเข้าอาคารโดยใช้เทคโนโลยีการสื่อสารสนามใกล้
PEOPLE ENTRANCE SYSTEM USING NFC TECHNOLOGY



โดย
นายธนโชติ อธิธิศิลป์
นายธนภัทร แสงนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบตรวจสอบคนเข้าอาคารโดยใช้เทคโนโลยีการสื่อสารสนามใกล้
PEOPLE ENTRANCE SYSTEM USING NFC TECHNOLOGY



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบคนเข้าอาคารโดยใช้เทคโนโลยีการสื่อสารสนามใกล้
People Entrance System using NFC Technology



โดย

นายธนโชติ อธิติศิลป์

56010512

นายธนภัทร แสงนวล

56010514

อาจารย์ที่ปรึกษา

ผศ.ดร. กฤษณ์ วรจรัส

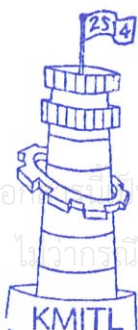
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559



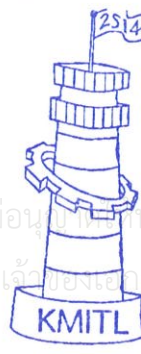
ผ่านการตรวจรูปเล่มแล้ว

อาจารย์ที่ปรึกษา

19/5/60

วิศวกรรมโทรคมนาคม

Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

กรรมการผู้ตรวจชิ้นงาน

19/5/60

วิศวกรรมโทรคมนาคม

Telecommunications Engineering

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันฯ สำหรับการให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปเผยแพร่
ในทางอื่นใด ๆ ทั้งสิ้น หากมีให้ดักแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบคนเข้าอาคารโดยใช้เทคโนโลยี NFC

PEOPLE ENTRANCE SYSTEM USING NFC TECHNOLOGY

ผู้จัดทำ

1. นายธนโชติ อธิธิศิลป์ 56010512
2. นายธนภัทร แสงนวล 56010514

..... อาจารย์ที่ปรึกษา
(ผศ.ดร. กฤษณ์ วรจิริระ)



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ผู้มีพระคุณท่านแรกที่คุณศึกษาขอขอบพระคุณ คือ ผศ.ดร.กฤษณ์ วงจรูจิระ อาจารย์ที่ปรึกษาที่ได้ให้ความรู้ สนับสนุนงบประมาณ คำแนะนำ ตรวจสอบ แก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ทุกขั้นตอน และเทคนิคการนำเสนอรายงานปากเปล่า เพื่อให้การค้นคว้าฉบับนี้ สมบูรณ์ที่สุด ผู้ศึกษาขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้ นอกจากนี้

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมโทรคมนาคมทุกท่านที่ได้สอน ได้ให้คำแนะนำ ในการการศึกษาค้นคว้าและจัดทำปริญญาานิพนธ์

ขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านที่ช่วยในการสืบค้นข้อมูลแลกเปลี่ยนความรู้ ความคิด และให้กำลังใจในการศึกษาแก่ผู้จัดทำ

ขอขอบคุณภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้สนับสนุนงบประมาณทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จ ลุล่วงไปด้วยดี

นายธนโชติ อธิธิศิลป์

นายธนภัทร แสงนวล

ผู้จัดทำ

ระบบตรวจสอบคนเข้าอาคารโดยใช้เทคโนโลยีการสื่อสารสนามใกล้
PEOPLE ENTRANCE SYSTEM USING NFC TECHNOLOGY

โดย นายธนโชติ อธิธิศิลป์ 56010512
นายธนภัทร แสงนวล 56010514

อาจารย์ที่ปรึกษา ผศ.ดร. กฤษณ์ วงจริระ

บทคัดย่อ

ปฏิญานิพนธ์นี้มีวัตถุประสงค์เพื่อประดิษฐ์ระบบตรวจสอบคนเข้าอาคาร โดยมี การตรวจสอบสองสิ่งด้วยกันคือใบหน้าและรหัสเฉพาะตัวตนของผู้ใช้งานที่มีแท็กเอ็นเอฟซี ภายในเครื่อง โทรศัพท์มือถือหรือบัตรเอ็นเอฟซีนั้น โดยเทคโนโลยีการสื่อสารสนามใกล้ (Near Field Communication) เป็นระบบระบุตัวตนด้วยรหัสแท็ก การทำงานของระบบคืออันดับแรกให้ ผู้ใช้บริการเลือกชั้นที่ต้องการโดยสัมผัสที่หน้าจอสัมผัสจากนั้นอ่านแท็กเอ็นเอฟซีของผู้ใช้งานบันทึก ข้อมูลไว้ และระบบจะทำการบันทึกใบหน้าโดยผู้ใช้งานต้องให้ใบหน้าอยู่หน้ากล้องในระยะเหมาะสม เป็นระยะเวลาหนึ่งแล้วทำการถ่ายรูปใบหน้าบันทึกไว้ ผู้ใช้งานจะได้รับอนุญาตไปชั้นที่กำหนดไว้ เท่านั้น

ABSTRACT

This project have objects to invent people go in and out checking system checking by 2 item are face and NFC tag ID from NFC smartphone or NFC tag. NFC(Near Field Communication) technology is identify system by using tag ID. Frist,user will select the floor by touching a touchscreen after that system will capture a user's face at the time user's face should be proper phase a period of time. NFC userID has been collect too. This system can only collect by 1 user per time.

สารบัญ

หน้า

กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VIII
บทที่ 1	บทนำ	1
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง	2
	2.1 บอร์ดราสเบอร์รี่พาย (RASPBERRY PI)	2
	2.2 RASPBERRY PI 7 INCH TOUCHSCREEN LCD DISPLAY	3
	2.3 คอมพิวเตอร์วิทัศน์	4
	2.4 ภาษาไพธอน	6
	2.5 การสื่อสารสนามใกล้ (NEAR FIELD COMMUNICATION)	7
	2.6 การออกแบบหน้าจอด้วย คิวที (QT CREATER, QT DESIGNER)	9
	2.7 เทรด (THREAD)	11
	2.8 SPI INTERFACES	12
	2.9 การตรวจจับใบหน้า (FACE DETECTION)	12
	2.10 WEB SERVER	16

สารบัญ (ต่อ)

	หน้า	
บทที่ 3	การออกแบบและการจัดทำปฏิญานิพนธ์	21
	3.1 การออกแบบ	22
	3.2 เครื่องมือที่ใช้ในการทดลอง	27
	3.3 การจัดเก็บผลการทดลอง	30
บทที่ 4	ผลการทดลอง	32
	4.1 การทดสอบการใช้งาน RASPBERRY PI กับ WEBCAM CAMERA	32
	4.2 การทดสอบการใช้งาน RASPBERRY PI ผ่าน NFC BOARD	33
	4.3 การทดสอบการใช้งาน RASPBERRY PI ผ่าน TOUCHSCREEN LCD DISPLAY	34
	4.4 การทดสอบการรันโปรแกรมแบบ BACKGROUND PROCESS กรณีอ่านบัตรเอ็นเอฟซี	35
	4.5 การทดสอบการรันโปรแกรมแบบ BACKGROUND PROCESS กรณีถ่ายใบหน้า	36
	4.6 การทดสอบสร้างเว็บเพื่อแสดงข้อมูลผู้เข้าออกอาคาร	36
บทที่ 5	สรุปผลและข้อเสนอแนะ	39
	5.1 สรุปผล	39
	5.2 ข้อเสนอแนะ	39
บรรณานุกรม		40
ภาคผนวก		43

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของบอร์ด RASPBERRY PI[1]	2
2.2 จุดเชื่อมต่อแบบ GPIO ของ RASPBERRY PI MODEL B[1]	3
2.3 RASPBERRY PI 7 INCH TOUCHSCREEN LCD DISPLAY[2]	4
2.4 ตัวอย่างการทำงานของคอมไพเลอร์[3]	7
2.5 การทำงานของ อินเทอร์เน็ตของ PYTHON[3]	7
2.6 โค้ดการใช้งาน SLOT	10
2.7 การเรียกใช้ไลบรารี โดยใช้คำสั่ง #INCLUDE	10
2.8 ขั้นตอนการทำงานของ QT DESIGNER[5]	11
2.9 การใช้งาน SPI INTERFACES[6]	12
2.10 ตัวอย่างลักษณะเด่นของ HAAR ที่ถูกใช้ใน OPENCV[7]	13
2.11 เทคนิคการรวมภาพ (A) หลังจากทีรวมภาพแล้วภาพที่ตำแหน่ง (X,Y) [7]	13
2.12 ตัวอย่างรูปแบบของ HAAR LIKE FEATURE[7]	14
2.13 กระบวนการทำงานของ CLASSIFIER [7]	15
2.14 ตัวอย่างตัวกรองสองตัวแรกใน VIOLA-JONES CASCADE [7]	15
2.15 ตัวอย่างตัวกรองอื่น ๆ ใน VIOLA-JONES CASCADE [7]	15
2.16 โลโก้โปรแกรม APACHE และ แบบการทำงานของ WEB SERVER[21]	16
2.17 รูปแบบการเลือกข้อมูลในตาราง[23]	17
2.18 รูปแบบการเพิ่มข้อมูลในตาราง[23]	17
2.19 รูปแบบการแก้ไขข้อมูลในตาราง[23]	17
2.20 รูปแบบการลบข้อมูลในตาราง[23]	17
2.21 หน้าต่างของโปรแกรม PHPMYADMIN	18
2.22 รูปแบบของภาษา PHP	18
2.23 เปรียบเทียบการทำงานระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้เอแจ็กซ์[26]	19
2.24 เปรียบเทียบการทำงานแบบดั้งเดิมกับแบบที่ใช้เอแจ็กซ์[26]	20

สารบัญรูป(ต่อ)

รูปที่	หน้า	
3.1	บล็อกไดอะแกรมการทำงานรวมของระบบ	21
3.2	แผนการทำงานของกล้อง	22
3.3	แผนการทำงานของเครื่องอ่านบัตร	23
3.4	แผนการทำงานของเครื่องอ่านบัตร	24
3.5	แผนการทำงานบนเว็บ	25
3.6	FLOWCHART การทำงานของระบบ	25
3.7	รูปแบบของเว็บที่ใช้งาน[27]	26
3.8	RASPBERRY PI ต่อกับ TOUCHSCREEN LCD DISPLAY	27
3.9	RASPBERRY PI ต่อกับ NFC BOARD	27
3.10	บอร์ดราสเบอร์รี่พาย 3 โมเดลปี[1]	27
3.11	หน้าจอสัมผัส[2]	27
3.12	กล้องเว็บแคม[8]	28
3.13	โมดูลอ่านค่ายูไอโอดี[9]	28
3.14	โปรแกรมคิวทีดีไซน์เนอร์[18]	28
3.15	โปรแกรมคอมพิวเตอร์วิทัศน์[20]	28
3.16	โปรแกรม APACHE[22]	29
3.17	โปรแกรม MYSQL[23]	29
3.18	โปรแกรม HTML	29
3.19	โปรแกรม PHP[25]	29
3.20	โปรแกรม AJAX[26]	30
4.1	ภาพบันทึกจากกล้องเว็บแคม	31
4.2	ภาพจาก PYTHON GUI	31
4.3	ภาพขาวดำ	31

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.4 การต่อ RASPBERRY PI กับ NFC BOARD	32
4.5 การวัดระยะระหว่าง TAGS และ NFC READER	32
4.6 TAGS และ NFC READER ตั้งฉากกัน	32
4.7 NFC READER อ่าน TAGS จากด้านข้าง	33
4.8 การต่อ RASPBERRY PI กับ TOUCHSCREEN LCD DISPLAY	33
4.9 TOUCHSCREEN LCD DISPLAY	33
4.10 หน้าจอเลือกชั้น	34
4.11 หน้าจออ่านบัตร	34
4.12 เปลี่ยนข้อความสถานะ	34
4.13 หน้าจอตรวจจับใบหน้า	35
4.14 หน้าจอตรวจจับใบหน้าที่ถ่ายได้	35
4.15 ตารางในฐานข้อมูล	35
4.16 ค่าในตารางขาเข้า	36
4.17 ค่าในตารางขาออก	36
4.18 หน้าเว็บ	36
4.19 ข้อมูลที่ประมวลผลออกมา	37

สารบัญตาราง

ตารางที่		หน้า
2.1	เปรียบเทียบ NFC กับ BLUETOOTH	9
2.2	สายสัญญาณสี่เส้นประกอบด้วย SCLK MOSI MISOI และ SS	12



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ทุกวันนี้มือถือสมาร์ทโฟนนั้นเป็นสิ่งที่ทุกคนนั้นมีติดตัวและเทคโนโลยีการสื่อสารสนามใกล้ จะเป็นสิ่งที่น่าสนใจเพื่อเพิ่มความสะดวกสบายการใช้บริการต่างๆในชีวิตประจำวันและลดการใช้อุปกรณ์หลายๆชิ้น ในปฏิญานิพนธ์นี้จึงเป็นปฏิญานิพนธ์เพื่อประดิษฐ์คิดค้นระบบที่ใช้ตรวจสอบบุคคล โดยนำไปใช้ในชีวิตประจำวันด้วยอุปกรณ์ที่ในยุคสมัยนี้มีกันทุกคน

1.2 วัตถุประสงค์

1. สร้างระบบในการเข้าอาคารที่สะดวกและสามารถตรวจสอบได้
2. เพื่อศึกษา Image Processing, HMI, NFC and Database server
3. เป็นการนำมือถือที่ทุกคนมีอยู่มาแทนที่อุปกรณ์ที่ต้องใช้เพิ่มเติม เช่น บัตรต่างๆ

1.3 ขอบเขตของปฏิญานิพนธ์

1. tagที่ใช้ต้องเป็นอุปกรณ์ที่รองรับ NFC และ RFID ได้
2. มีการบันทึกภาพใบหน้าของผู้ใช้บริการทุกครั้ง
3. สามารถใช้งานได้เพียง 1 บุคคลต่อครั้ง

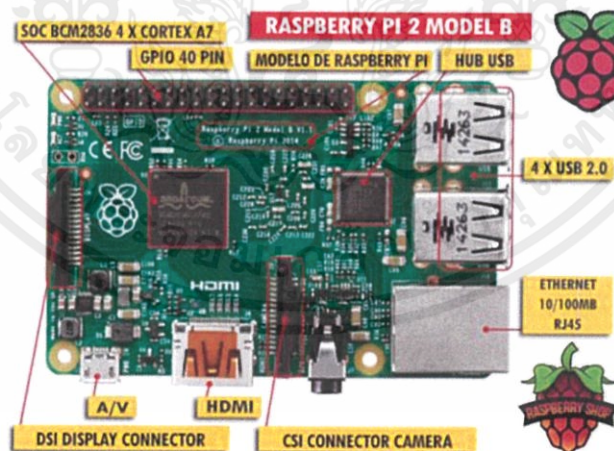
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญาานิพนธ์นี้เป็นการศึกษาการสร้างระบบที่ใช้ในการตรวจสอบคนเข้าอาคาร มีการแสดงตัวตนของผู้ใช้งานด้วยวิธีการถ่ายภาพใบหน้าและแสดงรหัสของอุปกรณ์การสื่อสารสนามใกล้ โดยใช้บอร์ดราสเบอร์รี่พายเป็นตัวส่งงานให้อุปกรณ์กล้องเว็บแคม ทำการถ่ายภาพใบหน้าในขอบเขตและขนาดที่กำหนด โดยใช้คอมพิวเตอร์วิทัศน์ ในการตรวจจับใบหน้า ซึ่งจะใช้ภาษาไพธอน และใช้คิวทีดีไซน์เนอร์ ในการสร้าง HMI กับผู้ใช้งาน โดยมีทฤษฎีที่เกี่ยวข้องดังนี้

2.1 บอร์ดราสเบอร์รี่พาย (Raspberry Pi)

บอร์ดราสเบอร์รี่พายเป็นคอมพิวเตอร์ในบอร์ดเดียว (Single Board Computer) พัฒนาขึ้นใน ประเทศสหรัฐอเมริกา โดย Raspberry Pi Foundation ภายใต้การทำงานขององค์กรไม่หวังผลกำไร สร้างคอมพิวเตอร์ในบอร์ดเดียวเพื่อใช้สำหรับการสอนและสำหรับเรียนของนักศึกษาในสาขา Computer Science ซึ่ง Raspberry Pi นั้นมีขนาดเล็กมาก มีความสามารถในการรองรับระบบปฏิบัติการ Linux ที่เรียกว่า Raspbian ที่บรรจุลงในเอสดีการ์ดสำหรับการพัฒนาไปสู่ บอร์ด Embedded Linux ซึ่งจากรูปที่ 2.1 นั้นจะเห็นส่วนประกอบต่างๆ ของบอร์ด Raspberry Pi ที่มี จุดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตทั้งผ่านพอร์ต USB, LAN, HDMI, CSI, DSI, AUDIO, ช่องสัญญาณภาพและ GPIO (General Purpose Input Output) สำหรับต่อกับวงจรรหรืออุปกรณ์ อิเล็กทรอนิกส์ต่างๆ ซึ่งอุปกรณ์ Raspberry Pi ยังมีคุณสมบัติทางเทคนิคดังต่อไปนี้



รูปที่ 2.1 ส่วนประกอบของบอร์ด Raspberry Pi[1]

2.1.1 คุณสมบัติทางเทคนิค

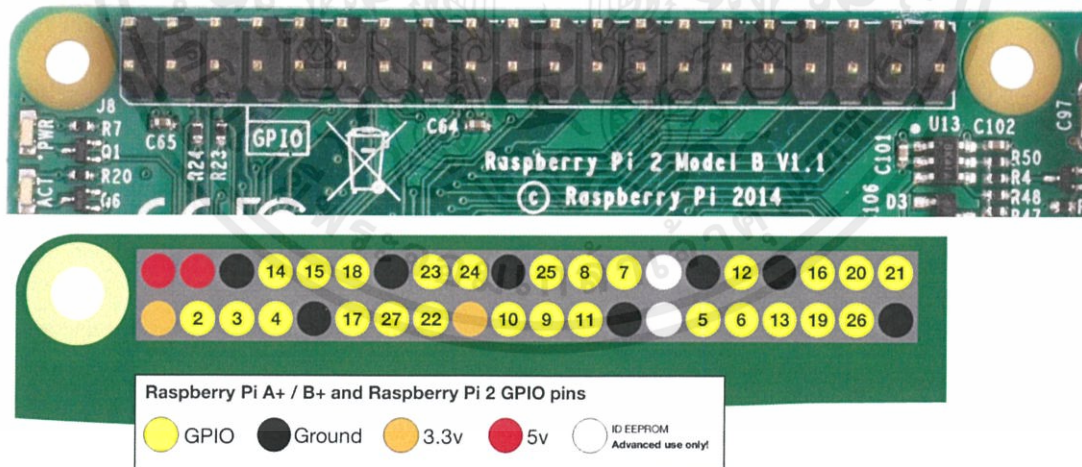
- ระบบประมวลผล Broadcom BCM2387 chipset, 1.2GHz Quad-Core ARM Cortex-A53.802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- GPU Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
- หน่วยประมวลผลกราฟิกหรือ GPU Broadcom Video core IV หรือเทียบเท่ารองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
- หน่วยความจำ 1GB LPDDR2
- ระบบปฏิบัติการ Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
- ขนาด 85 x 56 x 17mm
- การใช้พลังงาน Micro USB socket 5V1, 2.5A

2.1.2 จุดเชื่อมต่อแบบ GPIO [5] Raspberry Pi

สามารถเชื่อมต่อกับอุปกรณ์อิเล็กทรอนิกส์ได้ผ่าน GPIO (General Purpose Input Output) ซึ่งประกอบด้วย UART, SPI, PWM, I2C และอื่นๆ เพื่อใช้ในการควบคุม และสื่อสารกับอุปกรณ์อิเล็กทรอนิกส์ ซึ่งพอร์ต GPIO เป็นพอร์ตอินพุตเอาต์พุตเนกประสงค์ สามารถนำไปใช้งานได้ 21 ขา (ในเวอร์ชัน Rev.2 512MB) จึงสามารถเชื่อมต่อกับอุปกรณ์ อิเล็กทรอนิกส์ได้หลากหลายชนิดด้วยกัน รวมทั้งมีพอร์ตสำหรับจ่ายไฟให้กับอุปกรณ์อิเล็กทรอนิกส์ ด้วย ซึ่งมีทั้งขนาด 3V, 5V และกราวด์ ซึ่งได้แสดงไว้ในรูปที่ 2.2



รูปที่ 2.2 จุดเชื่อมต่อแบบ GPIO ของ Raspberry Pi Model B[1]

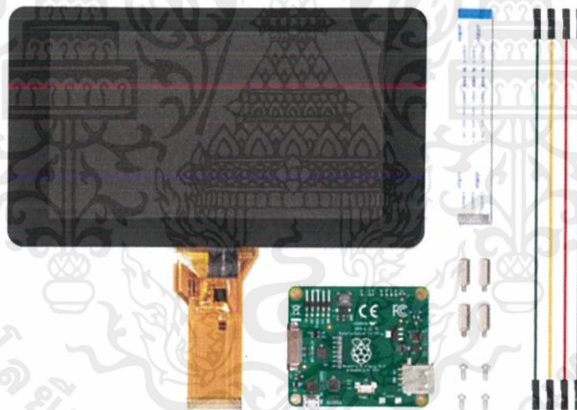
2.2 Raspberry Pi 7 inch Touchscreen LCD Display

จอแสดงผลขนาด 7 นิ้วสำหรับบอร์ด Raspberry Pi และระบบปฏิบัติการ Raspbian ความละเอียดหน้าจอ 800 x 480 จุด พร้อมสัมผัสแบบ Capacitive Touch รองรับการสัมผัสหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดพร้อมกัน (Multi-touch) สูงสุด 10 จุด เชื่อมต่อกับบอร์ด Raspberry Pi โดยตรงผ่านพอร์ต DSI (Display Serial Interface) สามารถใช้งานร่วมกันได้กับบอร์ด Raspberry Pi Model A+/B+, Pi 2 Model B และบอร์ด Raspberry Pi 3 Model B โดยด้านหลังของจอจะมีแผ่นโลหะพร้อมเจาะรูและมีเสามาให้ในชุดสำหรับยึดบอร์ด Raspberry Pi สามารถเชื่อมต่อใช้งานร่วมกับบอร์ด Raspberry Pi Model A/B ได้ ความสามารถในการแสดงสี RGB ระดับ 24-bit รองรับการสัมผัสหน้าจอแบบ Capacitive Touch ใช้นิ้วมือสัมผัสได้โดยไม่ต้องใช้ปากกา (Stylus) หรือออกแรงกดด้วยเล็บ รองรับ การสัมผัสหน้าจอพร้อมกันสูงสุด 10 จุด โดยต้องติดตั้งไลบรารีเพิ่มเติมเพื่อพัฒนาแอปพลิเคชันให้ รองรับการสัมผัสหลายจุดและในส่วน XWindow ของ Raspbian และซอฟต์แวร์อื่นๆ ทั่วไปจะยังไม่ รองรับการสัมผัสหลายจุด สัญญาณ ภาพและสัมผัสเชื่อมต่อผ่านพอร์ต DSI บนบอร์ด Raspberry Pi โดยผ่านบอร์ด PCB แปลง (มีมาให้ในชุดพร้อมกับจอ) ในการใช้งานต้องต่อใช้ไฟเลี้ยง 5 โวลต์เข้าที่ PCB สามารถใช้ได้จาก

1. แหล่งจ่ายภายนอก
2. พอร์ต USB
3. ขา 5V และ GND จาก Raspberry Pi



รูปที่ 2.3 Raspberry Pi 7 inch Touchscreen LCD Display[2]

2.3 คอมพิวเตอร์วิทัศน์

เป็นสาขาหนึ่งของวิทยาการคอมพิวเตอร์ ว่าด้วยเรื่องเกี่ยวกับการดั่งสารสนเทศจากรูปภาพหรือวิดีโอที่ค้น เครื่องมือที่ใช้ในคอมพิวเตอร์วิทัศน์ได้แก่ คณิตศาสตร์ โดยเฉพาะ เรขาคณิต พีชคณิตเชิงเส้น สถิติ และการวิจัยดำเนินงาน (การหาค่าเหมาะที่สุด) และการวิเคราะห์เชิงฟังก์ชัน โดยเครื่องมือเหล่านี้ใช้ในการสร้างขั้นตอนวิธีหรือ ขั้นตอนวิธี ในการแยกส่วนภาพ และการจัดกลุ่มภาพเพื่อให้คอมพิวเตอร์สามารถ "เข้าใจ" ทักษะภาพ หรือคุณลักษณะต่าง ๆ ในภาพ

เป้าหมายโดยทั่วไปของคอมพิวเตอร์วิทัศน์ได้แก่

- การตรวจจับ ตัดแบ่งขอบเขต ระบุตำแหน่ง และ รู้จำ วัตถุที่ต้องการในภาพ เช่น หน้าคน
- การประเมินผล สำหรับ การตัดแบ่งขอบเขตวัตถุในภาพ หรือ การวางทาบเทียบ เป็นต้น
- การวางทาบเทียบของ มุมมองต่างๆของทัศนียภาพ หรือ วัตถุหนึ่งๆ
- การติดตาม วัตถุหนึ่งๆ ในภาพต่อเนื่อง
- การเชื่อมโยงมุมมองต่างๆของทัศนียภาพหนึ่งๆ เพื่อสร้างแบบจำลองสามมิติ ของทัศนียภาพนั้น แบบจำลองดังกล่าวอาจนำมาใช้เพื่อนำทางหุ่นยนต์ ในทัศนียภาพจริง
- การกะประมาณ ท่าทางต่างๆ ของมนุษย์ และ ส่วนต่างๆของร่างกาย เช่น แขน ขา นิ้วมือ ฯลฯ ในสามมิติ
- การค้นหา รูปภาพด้วยเนื้อหาของภาพ ในฐานข้อมูลภาพขนาดใหญ่

เพื่อที่จะบรรลุซึ่งเป้าหมายเหล่านี้ ระบบคอมพิวเตอร์วิทัศน์ จะต้องใช้กระบวนการต่างๆ เช่น การรู้จำแบบ การเรียนรู้เชิงสถิติ เรขาคณิตเชิงภาพฉาย การประมวลผลภาพ ทฤษฎีกราฟ และอื่นๆ คอมพิวเตอร์วิทัศน์การรับรู้ นั้น เกี่ยวข้องอย่างใกล้ชิดกับ จิตวิทยาการรับรู้ และ การคำนวณทางชีวภาพ การประยุกต์ใช้คอมพิวเตอร์วิทัศน์อันหนึ่งที่น่าสนใจ ได้แก่ การสร้างปรากฏการลวงตาต่างๆ ในภาพยนตร์ ปัจจุบัน เราจะพบการประยุกต์ใช้ คอมพิวเตอร์วิทัศน์ในสาขาต่างๆ เช่น การแพทย์ การทหาร ระบบตรวจตราและรักษาความปลอดภัย การตรวจสอบและควบคุมคุณภาพ ระบบหุ่นยนต์ รถยนต์ และอื่นๆ ในปัจจุบัน เครื่องจักรวิทัศน์ และ การจัดการรูปภาพทางการแพทย์ที่ใช้วิธีการต่างๆทางคอมพิวเตอร์วิทัศน์ ได้รับการพัฒนา และ จัดจำหน่าย ในตลาดโลก คิดรวมเป็นมูลค่า หลายหมื่นล้านบาทต่อปี

งานของคอมพิวเตอร์วิทัศน์ จะต้องใช้ความรู้จากสาขาต่างๆ

- การรู้จำแบบ
- การเรียนรู้เชิงสถิติ
- projective geometry
- การประมวลผลภาพ
- ทฤษฎีกราฟ

มีความพยายามในการสร้างระบบที่มีความสามารถในการวิเคราะห์และประมวลผลภาพ โดยเป้าหมายหลักประการหนึ่งก็คือการสร้างระบบที่สามารถบอกได้ว่าภาพสองมิติที่ได้รับมานั้นเป็นภาพของ วัตถุใด มีลักษณะในสามมิติอย่างไร ศาสตร์ที่ว่านี้มีชื่อเรียกกันว่า Computer Vision แต่ Computer Vision ยังไม่สามารถพัฒนาให้ดีพอได้เพราะธรรมชาติของปัญหาเองที่ต้องการกู้คืนข้อมูลสามมิติจากข้อมูลเขาสองมิติ ซึ่งเป็นไปไม่ได้นอกจากมีข้อกำหนดเสริมที่เพียงพอ ดังสำนวนที่ว่า "A picture is worth a thousand words"

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษาไพธอน

Python เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูงถูกสร้างขึ้นในปี 1989 โดย Guido van Rossum ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างภาษาตัวนี้เป็นภาษาลักษณะ Open Source เหมือนอย่าง PHP

ความสามารถของ ภาษา ไพธอน

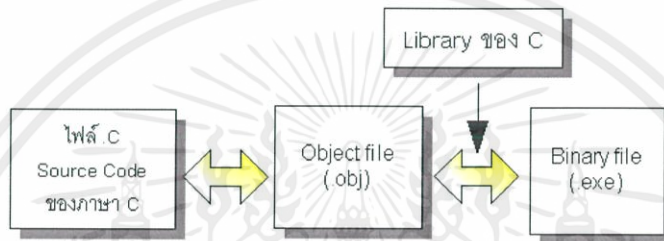
ในปัจจุบันภาษาที่ใช้ในการพัฒนา Web Application มีมากมายหลายภาษา อาทิเช่น ภาษา Perl, PHP, JAVA, ASP, Tcl, Python เป็นต้น สำหรับภาษา Python นับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษา Python ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ ซึ่งพอสรุปข้อดีของภาษา Python ได้ดังนี้

1. ง่ายต่อการเรียนรู้ โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษา C มาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
2. ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปลภาษา Python อยู่ภายใต้ลิขสิทธิ์ GNU
3. ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux, Windows 95/98/ME, Windows NT, Windows 2000, OS/2
4. ภาษา Python ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา C, C++, Java, Perl
5. ภาษา Python เป็นภาษาประเภท Server side Script คือการทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง
6. ใช้พัฒนา Web Service โดยที่ภาษา Python สามารถนำมาพัฒนาเว็บเซอร์วิส รวมทั้งใช้บริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Framework (CMF) ตัวอย่าง CMF ที่มีชื่อเสียงมากและเบื้องหลังทำงานด้วย Python คือ Plone

หลักการทํางานของภาษา Python

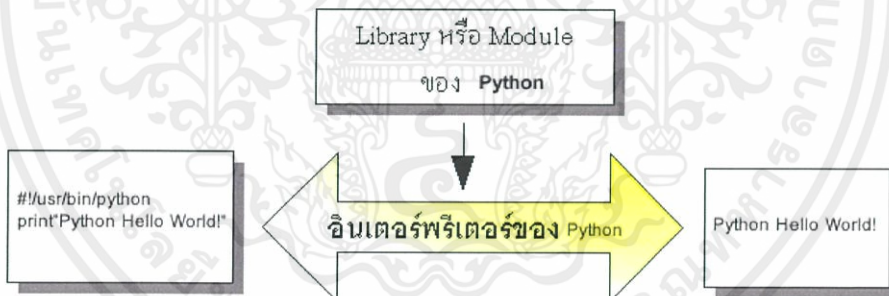
ตัวแปลภาษาคืออะไร เมื่อเราได้เขียนโค้ดขึ้นมาตามโครงสร้างของโปรแกรมภาษาใดก็ตาม และการจะให้โค้ดคำสั่งเหล่านั้นทํางานได้ก็จะต้องมีตัวแปลภาษามาจัดการแปลโค้ดคำสั่ง เพื่อให้ทํางานตามที่เราร้องขอ โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภทใหญ่ ๆ คือ

1. คอมไพเลอร์ (Compiler) เป็นตัวแปลภาษาสำหรับภาษา C, C++, Pascal การทํางานก็คือจะตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการคอมไพล์ ถ้าไม่มีข้อผิดพลาดก็จะทำการแปลโค้ดคำสั่งของเราให้เป็นไฟล์นามสกุล .obj (object file) จากนั้นก็ทำการแปลไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทํางานต่อไป ดังตัวอย่างการทํางานของคอมไพเลอร์ภาษา C ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างการทํางานของคอมไพเลอร์[3]

2. อินเตอร์พรีเตอร์ (Interpreter) จะทํางานเป็นบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทํางานให้ผลออกมาเลย ดังแสดงในรูป 2.5



รูปที่ 2.5 การทํางานของ อินเตอร์พรีเตอร์ของ Python[3]

จากรูปตัวอย่างในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารี (Library) หรือโมดูล (Module) ของภาษา Python อินเตอร์พรีเตอร์ของภาษา Python ก็จะไปทำการเรียกฟังก์ชันเหล่านั้นให้ทํางานแล้วจึงแสดงผลการทํางานออกมา

2.5 การสื่อสารสนามใกล้ (Near Field Communication)

การสื่อสารสนามใกล้ (อังกฤษ: near field communication ; NFC) เป็นเทคโนโลยีสื่อสารไร้สายระยะสั้นระยะประมาณ 4 ซม. ที่ใช้ได้กับโครงสร้างพื้นฐานแบบไร้สัมผัส ช่วยสนับสนุนรองรับการสื่อสารระหว่างเครื่องมืออิเล็กทรอนิกส์ในระยะใกล้ ๆ การสื่อสารสนามใกล้ถูกพัฒนาขึ้นโดยโซนี่และ NXP โดยใช้คลื่นความถี่ 13.56 MHz. บนพื้นฐานมาตรฐาน ISO 14443 (Philips MIFARE and Sony's FeliCa) ปัจจุบันบริษัททั้งสองได้ร่วมมือกับบริษัทผู้ผลิตและพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรศัพท์เคลื่อนที่ จัดตั้งเป็น NFC Forum เพื่อให้เกิดการใช้งานในรูปแบบต่าง ๆ มากขึ้น ในระยะเริ่มแรกมีบริษัทโทรศัพท์มือถือชั้นนำของโลกประกาศนำเทคโนโลยีนี้มาใช้กับโทรศัพท์มือถือแล้ว เช่น โนเกีย, ซัมซุง, โมโตโรลา เป็นต้น การสื่อสารสนามใกล้มักนำมาประยุกต์ใช้กับการแลกเปลี่ยนข้อมูลแบบ peer-to-peer เช่น เพลง เกม และรูปภาพ และประยุกต์ใช้กับการชำระเงินที่ต้องการความรวดเร็วและมีมูลค่าไม่สูง เช่น ใช้โทรศัพท์เคลื่อนที่เพื่อการชำระเงิน โดยวิธีการแตะบนเครื่องอ่านหรือเครื่องชำระเงินในร้านอาหารจานด่วน ร้านขายสินค้า ระบบการซื้อขายตัว และระบบการชำระเงินค่าโดยสารในระบบขนส่งมวลชน เป็นต้น การชำระเงินแบบไร้สัมผัสนี้ก่อให้เกิดการชำระเงินที่ง่ายและรวดเร็ว ลดการเข้าคิวชำระเงินในร้านค้า ห้างสรรพสินค้า และร้านสะดวกซื้อต่าง ๆ

2.5.1 ประโยชน์ของ NFC

1. โอนถ่ายข้อมูลได้แค่เพียง "แตะ" ซึ่งมีความเร็วมากกว่าการโอนถ่าย Bluetooth
2. จ่ายเงินผ่านมือถือแทนบัตรเงินสดต่างๆ โดยเชื่อมต่อกับบัญชีธนาคาร เช่น ตู้รถไฟฟ้ 7-11 ฯลฯ เปรียบเสมือนกระเป๋าดังอิเล็กทรอนิกส์
3. อ่าน Tag ต่างๆที่มี NFC อยู่ ซึ่ง Tag นั้นจะแสดงข้อมูลรายละเอียดต่างๆ ตามที่เจ้าของสังการไว้ เมื่อเราแตะจะแสดงข้อมูลนั้นบนโทรศัพท์มือถือของเรา

2.5.2 เปรียบเทียบกับบลูทูธ

การสื่อสารสนามใกล้และบลูทูธเป็นเทคโนโลยีการสื่อสารระยะใกล้คล้ายกันที่ปรากฏในโทรศัพท์มือถือ การสื่อสารสนามใกล้นั้นจะเปิดสัญญาณได้เร็วกว่าระบบบลูทูธ และการเชื่อมต่อระหว่างสองเครื่องจะสามารถเชื่อมต่อได้เร็ว แตกต่างจากบลูทูธที่จำเป็นต้องระบุตัวเครื่องของคู่ที่ต้องการเชื่อมต่อ ในขณะที่เดียวกันอัตราการส่งข้อมูลของเอ็นเอฟซี (424 kbit/s) มีอัตราที่ช้ากว่าบลูทูธ (2.1 Mbit/s) ด้วยระยะทางที่สั้นกว่าคือ 20 ซม. ซึ่งมีข้อเด่นคือจะไม่มีการแทรกแซงของข้อมูลหากในบริเวณมีการใช้งานอยู่จำนวนมาก

ตารางที่ 2.1 เปรียบเทียบ NFC กับ Bluetooth[4]

	NFC	บลูทูธ	บลูทูธพลังงานต่ำ
การเข้ากันได้กับอาร์เอฟไอดี	ISO 18000-3	active	active
มาตรฐาน	ISO/IEC	Bluetooth SIG	Bluetooth SIG
มาตรฐานเครือข่าย	ISO 13157 etc.	IEEE 802.15.1	IEEE 802.15.1
ชนิดของเครือข่าย	Point-to-point	WPAN	WPAN
การเข้ารหัส	not with RFID	available	available
ระยะ	< 0.2 m	~10 m (class 2)	~1 m (class 3)
ความถี่	13.56 MHz	2.4-2.5 GHz	2.4-2.5 GHz
อัตราบิต	424 kbit/s	2.1 Mbit/s	~1.0 Mbit/s
ระยะเวลาเปิด	< 0.1 s	< 6 s	< 1 s
การใช้พลังงาน	< 15mA (อ่าน)	ต่างไปตามชั้น	< 15 mA (xmit)

2.6 การออกแบบหน้าจอด้วย คิวที (Qt Creator, Qt Designer)

QT Creator [12] โดย QT เป็น Cross-Platform Application และ Framework สำหรับสร้าง User Interface (UI) บนพื้นฐานของภาษา C++ QML CSS และ JavaScript มีความโดดเด่นที่สามารถพัฒนาเพียงครั้งเดียวแล้วคอมไพล์ออกมาสำหรับแพลตฟอร์มที่แตกต่างกันได้ตามต้องการ นอกจากนี้ยังสามารถออกแบบหน้าจอ UI ได้สวยงาม มี Widget ที่หลากหลายและครบถ้วน และหากไม่เพียงพอต่อความต้องการ สามารถนำเอาไฟล์จากภายนอกเข้ามาประกอบเป็น UI ได้อย่างรวดเร็วและง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการพัฒนาแอปพลิเคชันด้วยภาษา C++ เพื่อควบคุม Hardware ของ Raspberry Pi และสร้าง Graphic User Interface เพื่อรับคำสั่งและแสดงผลให้กับผู้ใช้งานผ่านหน้าจอกอมพิวเตอร์

2.6.1 ประเภทวิดเจ็ตที่ใช้งาน

- QPushButton ใช้เปิดการใช้งานปุ่มกด
- QLabel ใช้แสดงป้ายตัวอักษร
- QlineEdit ใช้แสดงแถบข้อความ

2.6.2 การใช้งานสล็อต

การใช้งาน slot คือการกำหนดหน้าที่ให้ widget ต่างๆโดยจะมีการกำหนดแสดงได้ดังนี้

```
void MainWindow::on_push_res_clicked()
{
    QString cancel ;
    cancel = "0" ;
    ui->lineEdit->setText(QString(cancel));
    text = "0";
}
```

รูปที่ 2.6 โค้ดการใช้งาน slot

โดยที่ on_push_res_clicked() คือการกำหนดให้ทำงาน เมื่อมีการกดที่ปุ่ม จะให้แสดงค่าตัวแปร cancel บน QlineEdit

2.6.3 การเรียกใช้ไลบรารีต่างๆในควิที

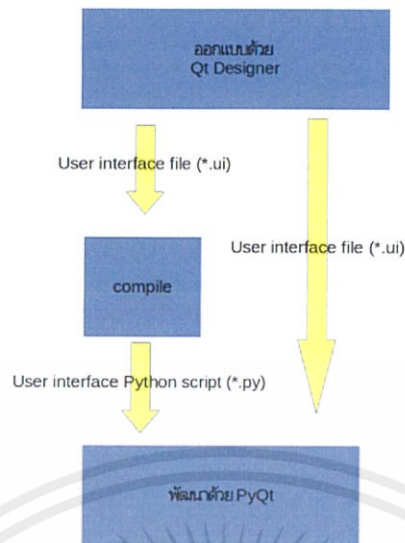
การใช้งานในรูปแบบต่างๆนั้นจะมีรูปแบบที่มีการออกแบบไว้ก่อนแล้ว โดยในส่วนของงานจะใช้ไลบรารีต่างๆ ดังนี้

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QProcess>
#include <QLineEdit>
#include <QFile>
#include <QTextStream>
#include <QMessageBox>
#include <QtNetwork/QTcpSocket>
```

รูปที่ 2.7 การเรียกใช้ไลบรารี โดยใช้คำสั่ง #include

2.6.4 การใช้ฟอร์มในควิที

ฟอร์มเป็นการสร้างหน้าจอที่ใช้แสดงผลเพื่อใช้งานจริง โดยในโปรแกรม QT จะมีให้ออกแบบได้โดยการใช้ drag and drop เพื่อความสะดวกในการออกแบบหน้าจอตนเอง



รูปที่ 2.8 ขั้นตอนการทำงานของ Qt Designer[5]

2.7 เทรด (Thread)

เทรด คือหน่วยการทำงานย่อยที่อยู่ในโปรเซส (Process) มีการแบ่งทรัพยากรภายในโปรเซสนั้น ปกติโปรเซสที่มี 1 เทรดจะเรียกว่าซิงเกิลเทรด (Single Thread) ซึ่งมักพบในระบบปฏิบัติการสมัยก่อน แต่ถ้า 1 โปรเซสมีหลายเทรด จะเรียกว่ามัลติเทรด (Multithread)

เหตุผลที่ต้องมีเทรด ก็เพื่อดึงความสามารถของหน่วยประมวลผลกลางให้เกิดประโยชน์สูงสุด โดยเทรดแต่ละตัวของโปรเซสเดียวกันจะทำงานแตกต่างกัน แต่มีความเกี่ยวข้องกันบางอย่าง และต้องทำงานอยู่ภายใต้สภาพแวดล้อมเดียวกัน

2.7.1 องค์ประกอบภายในเทรด

- Threads ID คือหมายเลขเทรดที่อยู่ในโปรเซส
- Counter คือตัวนับเพื่อติดตามคำสั่งที่ถูกดำเนินการเป็นลำดับถัดไป
- Register คือหน่วยความจำเก็บค่าตัวแปรที่ทำงานอยู่ปัจจุบัน
- Stack คือการเก็บประวัติการทำงาน

2.7.2 ประโยชน์ของมัลติเทรด

- 1) การตอบสนอง ในเรื่องการทำงานที่มีการตอบสนองกับผู้ใช้ มัลติเทรดช่วยลดและป้องกันการรอการทำงานจนจบโปรเซส
- 2) ประหยัดการใช้หน่วยความจำในการทำงานของโปรเซส เนื่องจากแต่ละเทรดมีการใช้หน่วยความจำของโปรเซสร่วมกัน
- 3) ประสานจังหวะการทำงานและใช้ทรัพยากรของโปรเซสร่วมกันได้

2.8 SPI Interfaces

SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารรูปแบบหนึ่ง ที่ใช้ในการติดต่อสื่อสารกับอุปกรณ์ ตัวอย่างเช่น ปริ้นเตอร์ กล้องถ่ายรูป เครื่องสแกนเนอร์ และอื่นๆ อีกมากมาย ถึงแม้ว่าการสื่อสารของ USB ที่มีฟังก์ชันการทำงานที่กว้างกว่า แต่การสื่อสารในรูปแบบ SPI ก็ยังถูกใช้งานกันอยู่ในบาง Application SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น MASTER ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น SLAVE และส่งข้อมูลในโหมด Full-duplex นั้นหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง MASTER และ SLAVE ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ ไม่ได้มีมาตรฐานกำหนดตายตัว ว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือ format แบบไหน เป็นการคิด protocol การสื่อสารกันเอาเอง

อุปกรณ์ที่ยังคงมีการใช้การสื่อสารแบบ SPI อยู่

- การแปลงข้อมูลจาก Analog to Digital หรือจาก Digital to Analog
- การติดต่อกับหน่วยความจำ EEPROM และ FLASH
- ไอซีประเภท Real Time Clock : RTC
- เซ็นเซอร์ จำพวก Temperature sensor , Pressure sensor
- อื่น ๆ เช่น signal mixer , Potentiometer , LCD controller , USART , CAN

controller , USB controller , Amplifier



รูปที่ 2.9 การใช้งาน SPI Interfaces[6]

2.8.1 พื้นฐานการทำงาน

SPI ต้องการสายสัญญาณ สี่เส้น บางครั้งเราเรียกว่าบัสอนุกรม "four wire" เส้นสัญญาณทั้งสี่เส้น ได้แก่

ตารางที่ 2.2 สายสัญญาณสี่เส้นประกอบด้วย SCLK MOSI MISO และ SS[6]

Line	Name	Description
SCLK	Serial Clock	Output from master
MOSI/SIMO	Master Output, Slave Input	Output from master
MISO/SOMI	Master Input, Slave Output	Output from slave
SS	Slave Select	Output from master (active low)

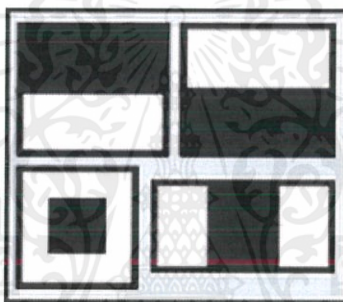
2.9 การตรวจจับใบหน้า (Face Detection)

การตรวจจับใบหน้าคือ กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นจะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไป เพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธี ซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

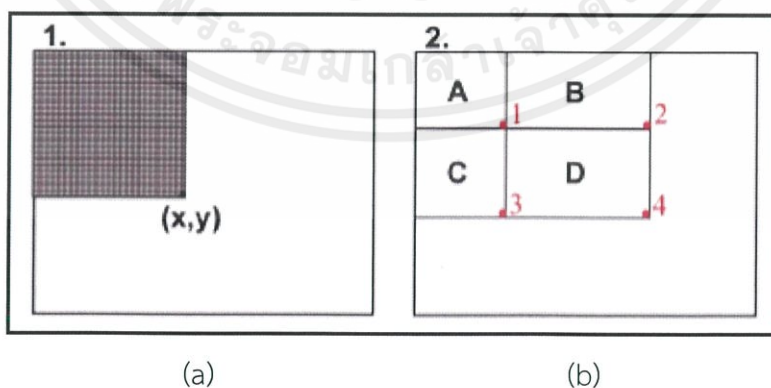
วิธีการที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตราความถูกต้องในการตรวจหาสูง ซึ่ง Paul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์ในปี ค.ศ. 2001 โดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอวิธีการแทนรูปภาพที่เรียกว่า “Integral Image”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

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการมีอยู่ของลักษณะเด่นของ Haar ถูกกำหนดโดยการลบค่าเฉลี่ยบริเวณจุดภาพมืดออกจากค่าเฉลี่ยบริเวณจุดภาพสว่าง ถ้าผลลัพธ์ที่ได้มีค่ามากกว่าค่าขีดแบ่ง แสดงว่ามีลักษณะเด่นของ Haar อยู่



รูปที่ 2.10 ตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV[7]

การตรวจสอบหาลักษณะเด่นของ Haar ในแต่ละส่วนของภาพอย่างมีประสิทธิภาพนั้น Viola และ Jones ใช้เทคนิคที่เรียกว่าการรวมภาพ (Integral Image) โดยการรวมจุดภาพเล็ก ๆ เข้าด้วยกัน จากรูปที่ 2.12 การจะหาค่าของรูปสี่เหลี่ยม $D = G(x_4, y_4) - G(x_2, y_2) - G(x_3, y_3) + G(x_1, y_1)$



รูปที่ 2.11 เทคนิคการรวมภาพ (a) หลังจากการรวมภาพแล้ว จุดภาพที่ตำแหน่ง (x, y) [7]

จะรวมค่าของทุกจุดภาพในสี่เหลี่ยมสีดำ, (b) ผลรวมค่าของทุกจุดภาพในสี่เหลี่ยม

$$D = G(x_4, y_4) - G(x_2, y_2) - G(x_3, y_3) + G(x_1, y_1)$$

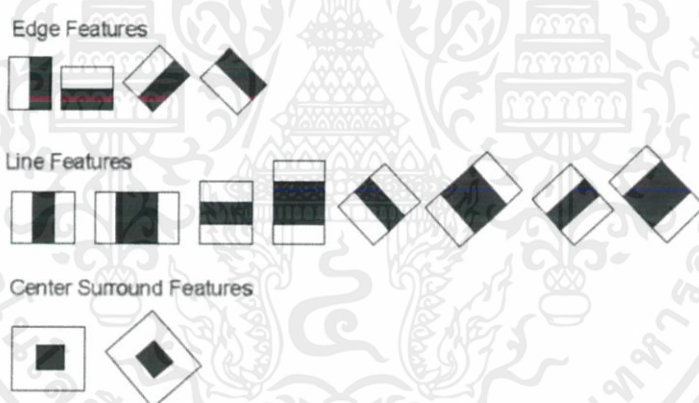
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.9.1 การเรียนรู้แบบ AdaBoost และ Cascade Classifier

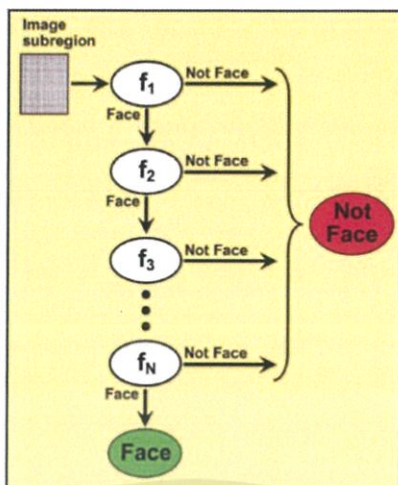
ขั้นตอนวิธีการเรียนรู้แบบ AdaBoost (AdaBoost learning algorithm) เป็นขั้นตอนวิธีการเรียนรู้เพื่อค้นหาค่าของกลุ่มพิกเซลที่มีลักษณะใกล้เคียงกับภาพใบหน้า โดยที่ภาพ positive คือ ภาพใบหน้าที่ต้องการตรวจจับ ส่วนภาพ negative คือ ภาพทั่ว ๆ ไปที่ไม่ใช่ใบหน้า ซึ่งการจำแนกกลุ่มของพิกเซลจะทำภายในส่วนย่อย (sub window)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

โดยที่ x คือภาพตัวอย่าง และ Sum คือผลรวมของภาพอินทิกรัลในบริเวณพื้นที่ขาวและดำของ x

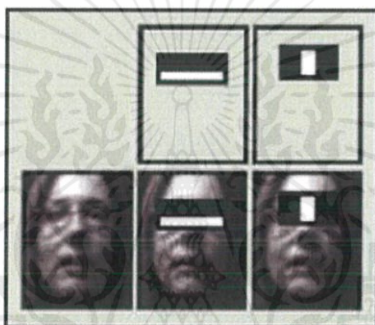


รูปที่ 2.12 ตัวอย่างรูปแบบของ Haar like feature[7]

Cascade Classifier เป็นวิธีการจำแนกรูปร่างที่ต้องการ โดยนำการจำแนกดังที่กล่าวมาแล้วมาทำซ้ำหลาย ๆ รอบ (stage) ซึ่งในแต่ละรอบก็จะตัดพื้นที่ที่เป็นภาพ negative ออกไปในทุกรอบที่พบ เมื่อจบกระบวนการแล้วจำนวนของ sub window ที่เป็น negative จะลดลงจนได้ใบหน้าที่ต้องการ และในการจำแนกเพื่อหารูปร่างที่ต้องการนี้จะทำกับภาพอินทิกรัล ซึ่งขั้นตอนของ Cascade Classifier แสดงได้ดังรูปที่ 2.13



รูปที่ 2.13 กระบวนการทำงานของ Classifier[7]



รูปที่ 2.14 ตัวอย่างตัวกรองสองตัวแรกใน Viola-Jones Cascade[7]

จากรูปที่ 2.15 จะพบว่าตัวกรองแรกใช้หลักการที่ว่าบริเวณแก้มสว่างกว่าบริเวณดวงตา และตัวกรองตัวที่สองใช้หลักการที่ว่าบริเวณสันจมูกจะสว่างกว่าบริเวณดวงตา



รูปที่ 2.15 ตัวอย่างตัวกรองอื่น ๆ ใน Viola-Jones Cascade[7]

หลักการของอัลกอริทึมค้นหาหน้าของ Viola-Jones คือการใช้ตัวตรวจหาสแกนหลาย ๆ ครั้งบนภาพเดิมด้วยขนาดที่แตกต่างกัน ถึงแม้ว่าจะมีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ได้โดยใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจว่าบริเวณใด ๆ ไม่ใช่ใบหน้านั้น ทำได้เร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded Classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.18 ซึ่งเมื่อ sub-window ถูกจัดประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็นมีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้นของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยังมีมากขึ้น

2.10 Web Server

เว็บเซิร์ฟเวอร์ (Web Server) คือเครื่องคอมพิวเตอร์ ที่ทำการติดตั้ง Software เกี่ยวกับ Server ที่ทำหน้าที่บริการและส่งข้อมูลให้กับ Client ที่ทำการ Request ข้อมูล และ Web Server จะทำการ Response เพื่อส่งผลลัพธ์ไปยัง Client โดย Web Server ทั่ว ๆ ไปที่เราคุ้น ๆ กันดีคือ ติดตั้ง Software ที่ชื่อว่า IIS (Internet Information Services) และ Apache โดย Service เหล่านี้ จะบริการข้อมูลผ่าน Port : 80 และรับส่งข้อมูลผ่าน Web Browser ส่วนมาก



รูปที่ 2.16 โลโก้ของโปรแกรม apache และ รูปแบบการทำงานของ web server [21]

2.10.1 Apache (httpd) Web Server

Apache (Httpd) Web Server เป็นโปรแกรมที่ทำหน้าที่เป็น Web Server หรือแปลงให้ Computer ที่ติดตั้ง Apache สามารถให้บริการ Web Site ที่สามารถเรียกจากเครื่องอื่น ๆ ที่เป็น Client ได้ เช่น เรียกผ่าน Web Browser ของ IE , Chrome หรือ Firefox และ Apache ยังสามารถทำงานร่วมกับภาษาที่ใช้พัฒนาเว็บไซต์ได้หลากหลาย เช่น PHP , Ruby ,JSP และอื่น ๆ

2.10.2 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา php ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิซวลเบสิกดอทเน็ต ภาษาจาวา หรือภาษาซี

ซาร์ป เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ซ (Open Source) ที่ถูกนำไปใช้งานมากที่สุด

การ SELECT เป็นคำสั่งของฐานข้อมูลที่ใช้ในการดำเนินการบ่อยที่สุด โดย SELECT คือ การเลือกคอลัมน์ที่เรากำหนดเอาไว้ FORM คือการเลือกตารางข้อมูล WHERE คือการชี้เฉพาะตำแหน่งที่ต้องการเลือกในตารางข้อมูล mysql_query คือการใช้ข้อมูลที่เลือกไป mysql_fetch_assoc คือการนำข้อมูลหลายๆข้อมูลกับในตัวแปรเดียวในรูปแบบ array

```
$get_ticket = "SELECT * FROM `in_history` WHERE `uid` = '$id_1'";
$get_ticket_query = mysql_query($get_ticket);
while ($get_ticket_object = mysql_fetch_assoc($get_ticket_query)) {
    $return_result[] = $get_ticket_object;
}
```

รูปที่ 2.17 รูปแบบการเลือกข้อมูลในตาราง[23]

การ INSERT เป็นคำสั่งที่ใช้ในการเพิ่มข้อมูลในตาราง โดย INSERT INTO คือการเพิ่มข้อมูลลงในตาราง VALUE คือ ค่าที่ต้องการเพิ่มลงในคอลัมน์นั้นๆ mysql_query คือการใช้ข้อมูลที่เลือกไป

```
$kiosk_log = "INSERT INTO `kiosk_log` (id, log_type, log_function, result, message, element_1, element_1_type, element_2, element_2_type, severity, user, time) VALUE (NULL, '$log_type', '$log_function', '$result', '$message', '$element_1', '$element_1_type', '$element_2', '$element_2_type', '$severity', '$user', now());";
$kiosk_log = mysql_query($kiosk_log);
```

รูปที่ 2.18 รูปแบบการเพิ่มข้อมูลในตาราง[23]

การ UPDATE เป็นคำสั่งที่ใช้แทนค่าบางค่าในตาราง

```
$host_to_hostgroup_insert = "UPDATE lnkHostToHostgroup SET ";
$host_to_hostgroup_insert .= "idMaster = '$idMaster'";
$host_to_hostgroup_insert .= ",idSlave = '$hostgroups_id'";
$host_to_hostgroup_insert .= ",exclude = '0'";
$host_to_hostgroup_insert .= "WHERE idMaster = '$idMaster'";
$host_to_hostgroup_query = mysql_query($host_to_hostgroup_insert );
```

รูปที่ 2.19 รูปแบบการแก้ไขข้อมูลในตาราง[23]

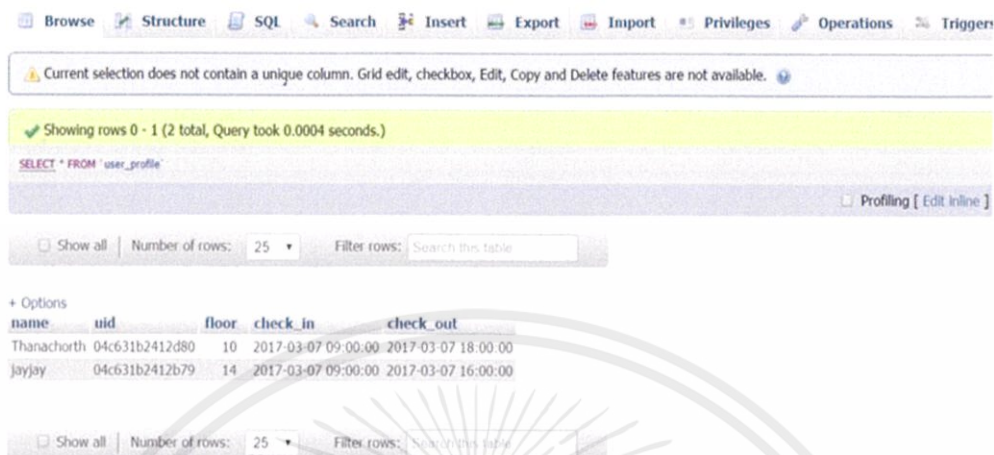
การ DELETE เป็นคำสั่งที่ใช้ในการลบค่าในตาราง

```
$del_host_contact_Str = "DELETE FROM `lnkHostToContact` WHERE `idMaster` = '$idMaster'";
$del_host_contact = mysql_query($del_host_contact_Str);
```

รูปที่ 2.20 รูปแบบการลบข้อมูลในตาราง[23]

PHPMyAdmin เป็นสคริปต์ติดต่อฐานข้อมูลที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง

ตารางใหม่ และยังมีฟังก์ชันที่ใช้สำหรับการทดสอบการสืบค้นข้อมูลด้วยภาษา SQL พร้อมกันนั้น ยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกันกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล



รูปที่ 2.21 หน้าต่างของโปรแกรม PHPMyAdmin

2.10.3 PHP

ภาษา PHP มีลักษณะเป็น embedded script หมายความว่าเราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่ง(Tag) ของ HTML ได้ และสร้างไฟล์ที่มีนามสกุลเป็น .php, .php3 หรือ .php4 ซึ่งไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาต่างๆ มารวมกันได้แก่ C, Perl และ Java

```
<html>
<head>
<title>Example 1 </title>
</head>
<body>
<?
echo "Hi, I'm a PHP script!";
?>
</body>
</html>
```

รูปที่ 2.22 รูปแบบของภาษา PHP

2.10.4 AJAX

Asynchronous JavaScript and XML เป็นกลุ่มของเทคนิคในการพัฒนาเว็บ แอปพลิเคชันเพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็ว และการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานโดยรวม เอแจ็กซ์นั้นไม่ใช่เทคโนโลยีใหม่ แต่เป็นเทคนิคที่ได้ใช้เทคโนโลยีหลายอย่างที่มีอยู่แล้ว รวมกันดังต่อไปนี้:

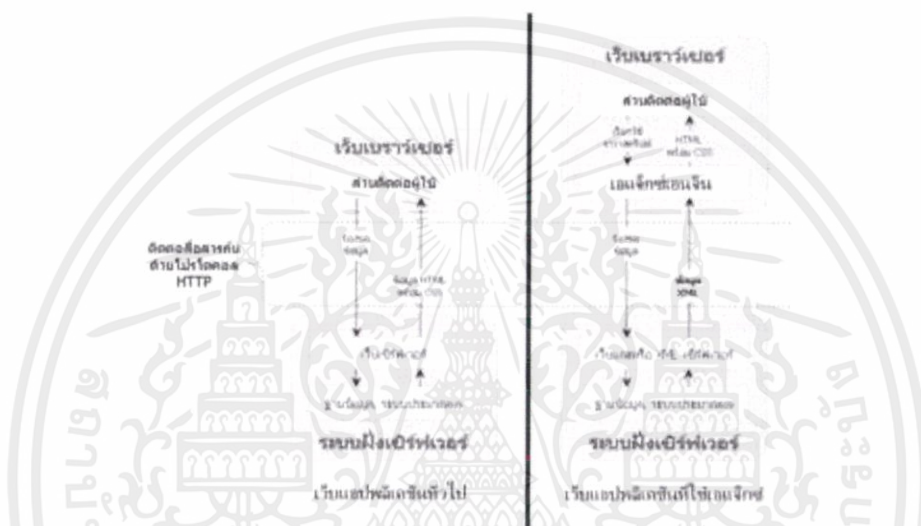
XHTML (หรือ HTML) และ CSS ใช้ในการแสดงผลและรูปแบบข้อมูล

ECMAScript เช่นจาวาสคริปต์ ในการเข้าถึง Document Object Model (DOM)

เพื่อใช้ในการแสดงข้อมูลที่มีการเปลี่ยนแปลงหรือโต้ตอบกับผู้ใช้

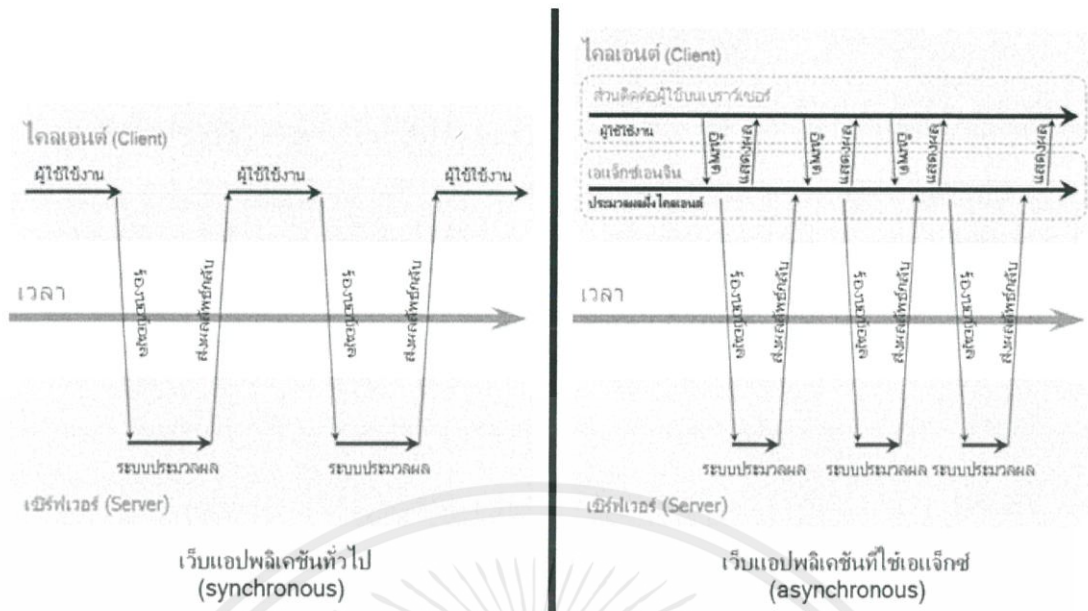
XMLHttpRequest ใช้ในการแลกเปลี่ยนข้อมูล Asynchronously กับเว็บเซิร์ฟเวอร์

XML ใช้เป็นรูปแบบข้อมูลในการแลกเปลี่ยน ซึ่งรูปแบบอื่นก็สามารถใช้ได้เช่นกันไม่ว่าจะเป็น HTML, JSON, EBML, หรือ ข้อความธรรมดา



รูปที่ 2.23 เปรียบเทียบการทำงานระหว่างเว็บแอปพลิเคชันแบบดั้งเดิมกับแบบที่ใช้เอแจ็กซ์[26]

วิธีการทำงานของเว็บแอปพลิเคชันแบบดั้งเดิมนั้น โดยปกติแล้วเมื่อผู้ใช้ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์ ตัวเว็บเบราว์เซอร์จะทำการส่งข้อมูลการร้องขอโดยใช้โพรโทคอล HTTP เพื่อติดต่อกับเว็บเซิร์ฟเวอร์ และที่เว็บเซิร์ฟเวอร์จะทำการประมวลผลจากการร้องขอที่ได้รับ และส่งผลลัพธ์เป็นหน้า HTML กลับไปให้ผู้ใช้ วิธีการข้างต้นเป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) ซึ่งผู้ใช้จะต้องรอระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ ซึ่งเป็นหลักการทำงานแบบ Synchronous แต่การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิคเอแจ็กซ์จะเป็นการทำงานแบบ Asynchronous หรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยเซิร์ฟเวอร์จะทำการส่งผลลัพธ์เป็นเว็บเพจให้ผู้ใช้ทันทีโดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเว็บเพจที่ผู้ใช้ได้รับจะทำการดึงข้อมูลในส่วนต่างๆ ที่หลัง หรือจะดึงข้อมูลก็ต่อเมื่อผู้ใช้ต้องการเท่านั้น (ทำงานอยู่เบื้องหลัง)

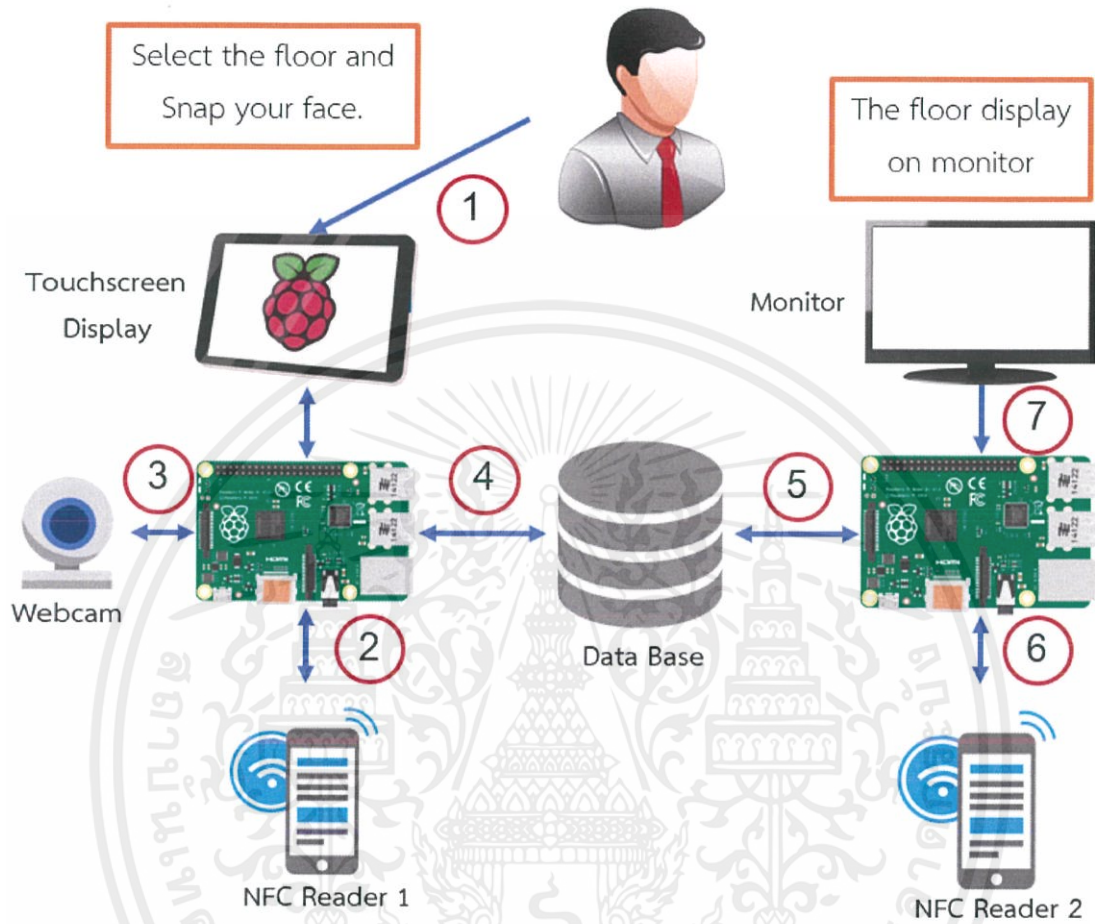


รูปที่ 2.24 เปรียบเทียบการทำงานแบบดั้งเดิมกับแบบที่ใช้เอเจ็ทซ์[26]



บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ

จากบล็อกไดอะแกรมสามารถอธิบายการทำงานได้ดังนี้ เมื่อผู้ใช้งานทำการเลือกชั้นที่ต้องการจะไปโดยการสัมผัสที่หน้าจอของ Raspberry Pi Touchscreen LCD Display ระบบจะทำการส่งค่าชั้นไปที่ Database ชั้นต่อมาจะให้ผู้ใช้งานถ่ายภาพหน้าของตนเองโดยการถ่ายจาก Webcam camera ภาพจะถูกเก็บบันทึกไว้ที่ Raspberry Pi #1 ชั้นถัดมาจะให้ผู้ใช้งานนำบัตร NFC หรือมือถือที่สามารถใช้งาน NFC ไปแตะที่ NFC Reader #1 ระบบจะทำการบันทึกค่า UID ของผู้ใช้งานไว้ที่ Database ชั้นต่อมาผู้ใช้งานต้องมา (เป็นการออกจากอาคาร) แตะบัตร NFC ที่ NFC Reader #2 ระบบทำการบันทึกค่า UID1 และเวลา

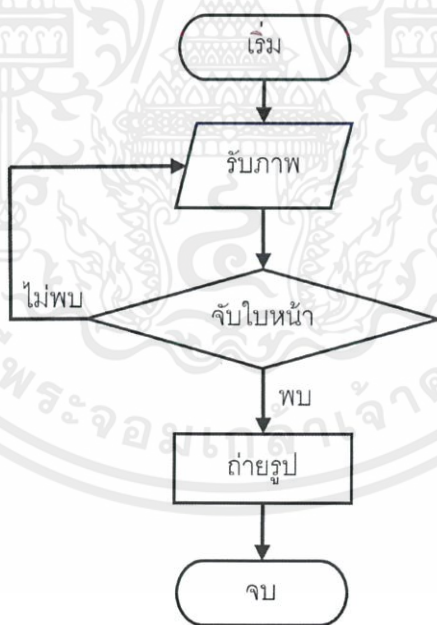
3.1 การออกแบบ

3.1.1 การออกแบบหน้าที่การทำงานของภาษา Python

ในโปรแกรมของระบบนี้โดยหลักการใช้ภาษาไพธอนเป็นส่วนใหญ่ โดยเริ่มที่ขั้นตอนการอ่าน NFC Tags เพื่อแสดงตนจะทำการรันโปรแกรมภาษาไพธอนไว้รอก่อน เมื่ออ่านได้จาก Raspberry Pi #1 จะทำการส่งค่า UID ไปยัง Database การตรวจจับใบหน้าโปรแกรมไพธอนจะทำการดึง OpenCV และ Haar Cascade มาใช้ เมื่อมาถึงขั้นตอนการออกจากอาคาร

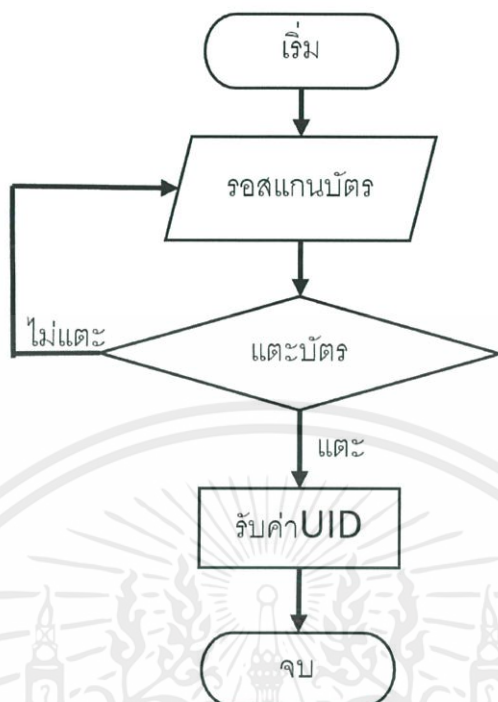
3.1.2 การออกแบบหน้าที่การทำงานของ OpenCV

ในหลักการทำงานของการตรวจจับใบหน้านั้นจะต้องมีกระบวนการในการนำภาพจริงที่ได้จาก Webcam camera ซึ่งจะมีลักษณะเป็นภาพ BGR (Blue Green Red) color space มาแปลงเป็นภาพที่เป็นลักษณะ Grey color space ซึ่งเป็นการแปลงมาเป็นลักษณะของภาพขาวดำออกมาอย่างชัดเจน ในส่วนนี้จะมีการนำ Haar Cascade มาใช้ในการประมวลผลตำแหน่งของตาและปาก จากนั้นทำการตรวจจับใบหน้าและวงกรอบสีเหลี่ยมสีน้ำเงินบอกขอบเขตใบหน้าที่ตรวจจับได้



รูปที่ 3.2 แผนการทำงานของกล้อง

3.1.3 การออกแบบหน้าที่การทำงานของ NFC PN532 Board



รูปที่ 3.3 แผนการทำงานของเครื่องอ่านบัตร

เราจะทำการรันการทำงานของโปรแกรมภาษาไพธอน เพื่อรอการตรวจจับ Tags ของบัตรต่างๆหรือโทรศัพท์มือถือ โดยการให้ NFC PN532 Board เชื่อมต่อกับ Raspberry Pi ด้วย interface คือ SPI(Serial Peripheral Interface) ซึ่งจะใช้ช่องทางการสื่อสารกันด้วย 4 ช่องทาง คือ SCK MOSI MISO และSS

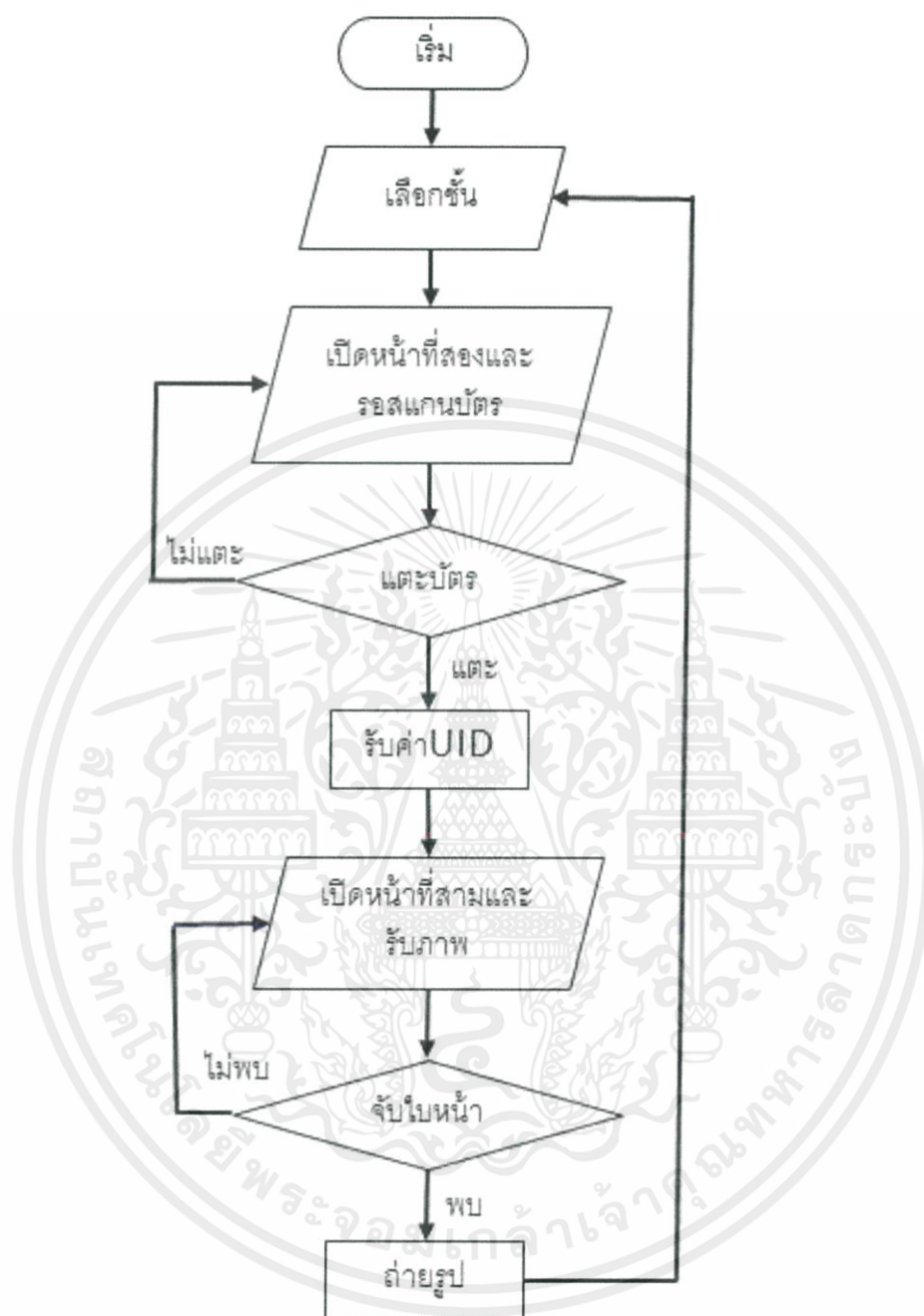
3.1.4 การใช้โปรแกรม QT เพื่อสร้างหน้าจอแรกเข้าอาคาร

3.1.4.1) เลือกชั้นของอาคาร

เมื่อแรกเข้าอาคารเราจะใช้หน้าจอสัมผัสในการกดเพื่อเลือกชั้นของอาคารที่ต้องการจะไปโดยการสร้างฟอร์มของหน้าจอสัมผัสเราจะสร้างโดยใช้โปรแกรม Qt Designer โดยฟอร์มจะมีลักษณะเป็นปุ่มกด

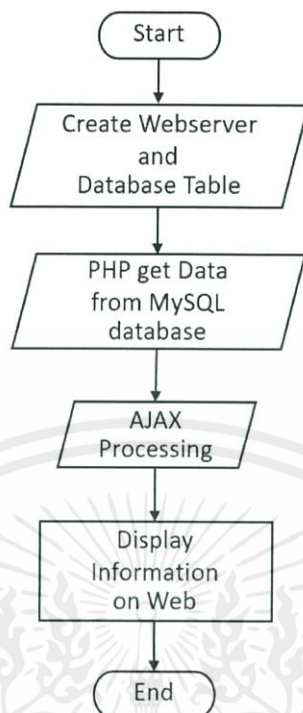
3.1.4.2) ส่งข้อมูลไปยัง Database

เมื่อผู้เข้าอาคารใช้งานหน้าจอสัมผัสจนจบการทำงานแล้วหน้าจอจะส่งข้อมูลทั้งหมดไปที่ Database ภายในรหัสเบอร์รียาย



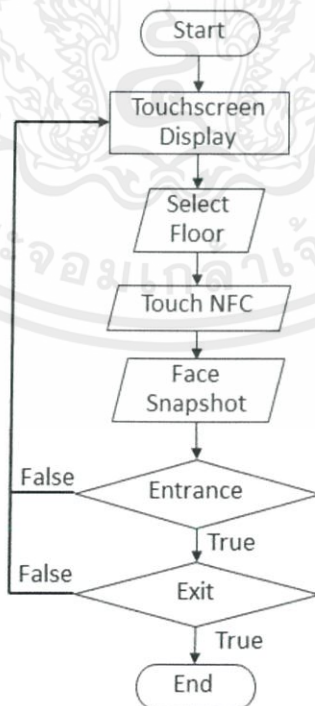
รูปที่ 3.4 แผนการทำงานเครื่องอ่านบัตร

3.1.5 การนำข้อมูลจาก GUI มาแสดงบนเว็บ



รูปที่ 3.5 แผนการทำงานบนเว็บ

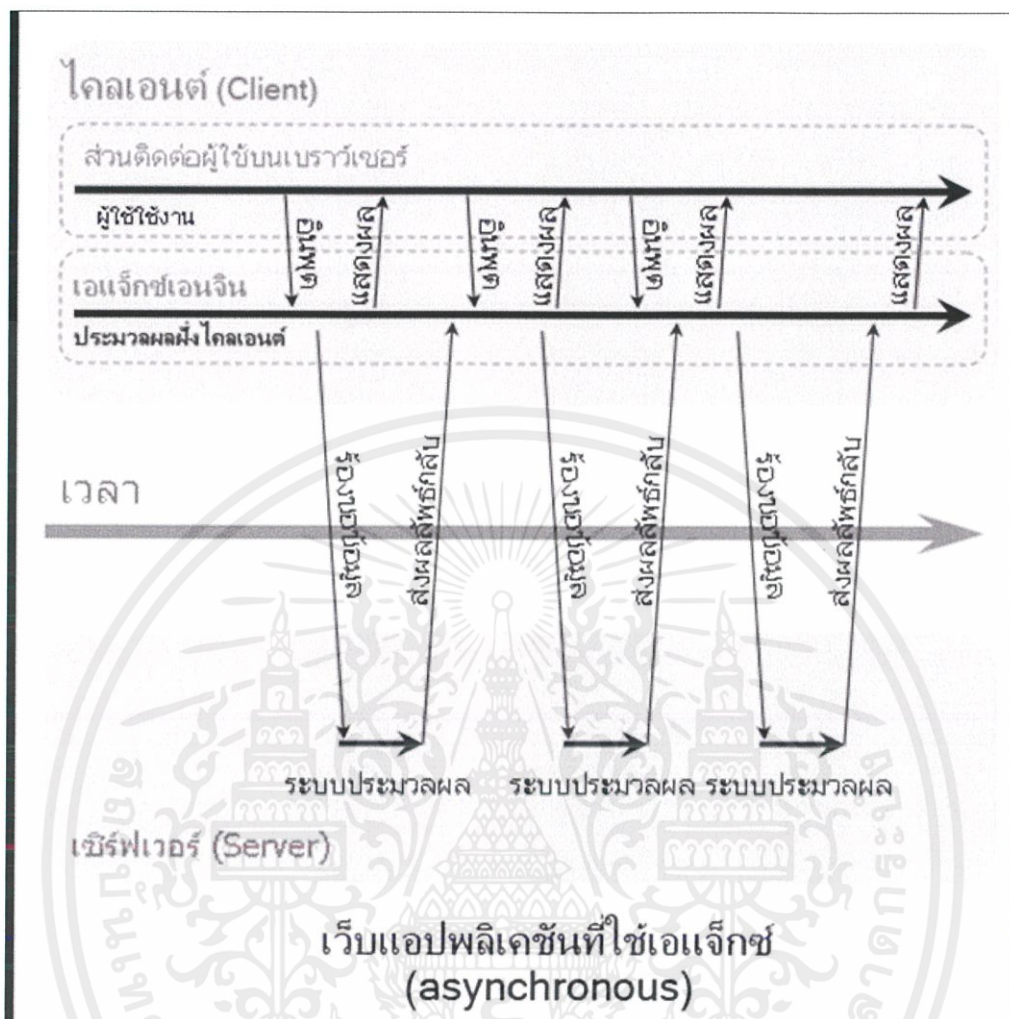
3.1.6 Flowchart การทำงานของระบบ



รูปที่ 3.6 Flowchart การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

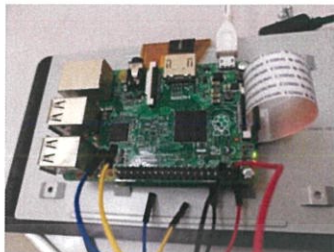
3.1.7 รูปแบบการทำงานของเว็บที่แสดงข้อมูลประวัติของผู้เข้า-ออกอาคาร



รูปที่ 3.7 รูปแบบของเว็บที่ใช้งาน[27]

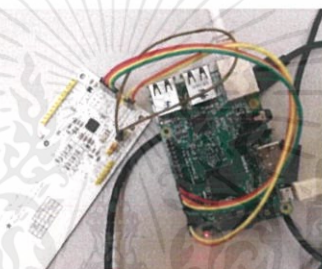
3.1.8 ภาพการต่ออุปกรณ์

3.1.5.1) Raspberry Pi ต่อกับ Touchscreen LCD Display



รูปที่ 3.8 Raspberry Pi ต่อกับ Touchscreen LCD Display

3.1.5.2) Raspberry Pi ต่อกับ NFC Board



รูปที่ 3.9 Raspberry Pi ต่อกับ NFC Board

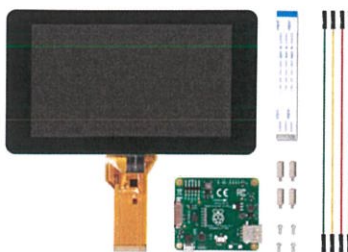
3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 บอร์ดราสเบอร์รี่พาย (Raspberry Pi 3 Model B)



รูปที่ 3.10 บอร์ดราสเบอร์รี่พาย 3 โมเดลบี[1]

3.2.2 Raspberry Pi 7 inch Touchscreen LCD Display



รูปที่ 3.11 หน้าจอสัมผัส[2]

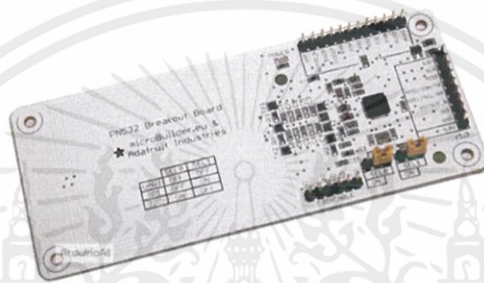
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 WEBCAM OKER OE-177 (New Version)



รูปที่ 3.12 กล้องเว็บแคม[8]

3.2.4 RFID NFC PN532 Shield



รูปที่ 3.13 โมดูลอ่านค่ายูเอตไอ[9]

3.2.5 โปรแกรมควิทีดีไซน์เนอร์ (Qt Designer)



รูปที่ 3.14 โปรแกรมควิทีดีไซน์เนอร์[18]

3.2.6 โปรแกรมคอมพิวเตอร์วิทัศน์ (OpenCV Program)



รูปที่ 3.15 โปรแกรมคอมพิวเตอร์วิทัศน์[20]

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 โปรแกรม APACHE2 Web Server



รูปที่ 3.16 โปรแกรม APACHE[22]

3.2.8 โปรแกรม MySQL



รูปที่ 3.17 โปรแกรม MySQL[23]

3.2.9 ภาษา HTML



รูปที่ 3.18 โปรแกรม HTML

3.2.10 ภาษา PHP



รูปที่ 3.19 โปรแกรม PHP[25]

3.2.11 ภาษา AJAX



รูปที่ 3.20 โปรแกรม AJAX[26]

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทำงานของ Raspberry Pi กับ Webcam camera

ในส่วนนี้จะทำการเขียนโค้ดโปรแกรมคำสั่งควบคุม โดยจะประมวลผลใน Raspberry Pi และจะติดต่อกับกล้อง Webcam camera เพื่อส่งให้จับภาพ

3.3.2 การทำงานของ Raspberry Pi กับ NFC Board

ในส่วนนี้จะทำการเขียนโค้ดโปรแกรมคำสั่งควบคุม NFC Board รอกการอ่าน NFC Tags โดยจะประมวลผลและบันทึกค่า UID ไว้ใน Raspberry Pi

3.3.3 การทำงานของ Raspberry Pi กับ Touchscreen LCD Display

ในส่วนนี้จะทำการเขียนคำสั่งควบคุมลงในโปรแกรม Qt Designer เพื่อนำฟอร์มที่สร้างไว้แสดงผลผ่านหน้าจอสัมผัสของ Raspberry Pi

3.3.4 การรันโปรแกรมแบบ Background process กรณีอ่านบัตรเอ็นเอฟซี

ในส่วนนี้จะเป็นการเขียนคลาสเพิ่มขึ้นมาในโปรแกรม Qt Designer เพื่อทำการรันโปรแกรมหน้าจอGUIพร้อมกับโปรแกรมอ่านบัตร

3.3.5 การรันโปรแกรมแบบ Background process กรณีถ่ายใบหน้า

ในส่วนนี้จะเป็นการเขียนคลาสเพิ่มขึ้นมาในโปรแกรม Qt Designer เพื่อทำการรันโปรแกรมGUIพร้อมกับโปรแกรมถ่ายใบหน้า

3.3.6 การนำข้อมูลจาก GUI มาแสดงบน Web

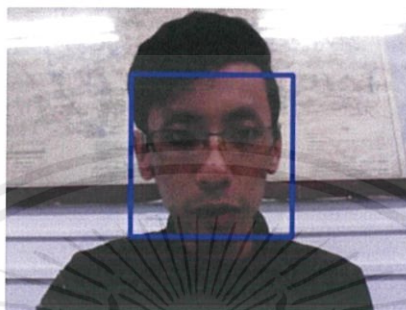
ในส่วนนี้จะเป็นการนำข้อมูลทั้งหมดที่ได้เก็บไว้แล้วในแต่ละตาราง รวบรวมนำมาแสดงบนหน้าเว็บ เพื่อให้สะดวกแก่การใช้งานและการแบ่งส่วนข้อมูลให้ชัดเจน

บทที่ 4

ผลการทดลอง

4.1 การทดสอบการใช้งาน Raspberry Pi กับ Webcam camera

4.1.1 เป็นการแสดงภาพวิดีโอจากกล้องเว็บแคม



รูปที่ 4.1 ภาพบันทึกจากกล้องเว็บแคม

4.1.2 จับเวลา 2 วินาทีหลังจากจับใบหน้าได้

```
>>> ----- RESTART -----
>>>
1
2
Finish For Snapshot
>>> |
```

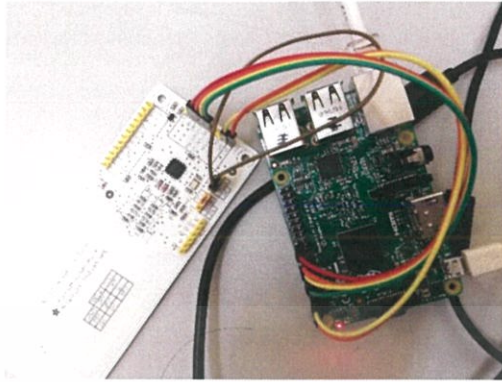
รูปที่ 4.2 ภาพจาก Python GUI

4.1.3 แปลงภาพจากสีปกติเป็นขาวดำ



รูปที่ 4.3 ภาพขาวดำ

4.2 การทดสอบการใช้งาน Raspberry Pi ผ่าน NFC Board



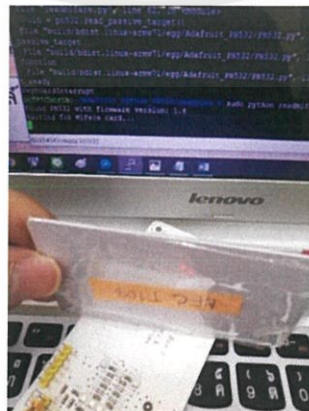
รูปที่ 4.4 การต่อ Raspberry Pi กับ NFC Board

4.2.1 อ่านแบบปกติ เป็นการแสดงว่า NFC สามารถอ่านค่า Tags ได้ที่ระยะห่าง 9 เซนติเมตร ซึ่งตามคุณสมบัติของ PN 532 คืออ่านได้ที่ระยะห่าง 10 เซนติเมตร คลาดเคลื่อนจากการทดลองจริง 1 เซนติเมตร หรือคิดเป็นร้อยละ 10



รูปที่ 4.5 การวัดระยะระหว่าง Tags และ NFC Reader

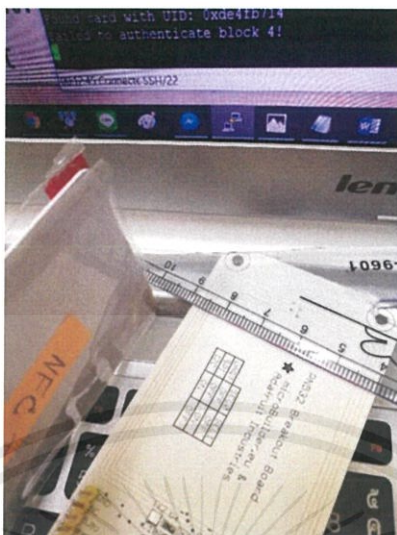
4.2.2 อ่านค่า Tags โดยเอียงบัตร 90 องศา กับ NFC Reader เป็นการแสดงให้เห็นว่าบอร์ดสามารถไม่อ่านค่า Tags ที่มุมเอียง 90 องศา กับบัตรได้



รูปที่ 4.6 Tags และ NFC Reader ตั้งฉากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 NFC Reader สามารถอ่าน Tags จากด้านข้างได้ที่ระยะ 7 มิลลิเมตร

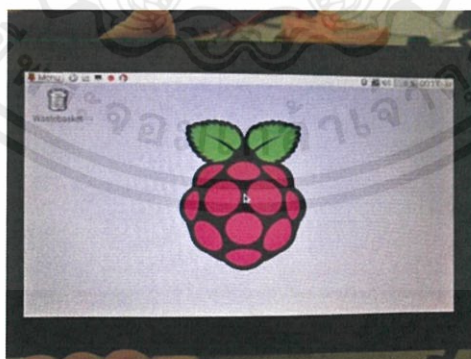


รูปที่ 4.7 NFC Reader อ่าน Tags จากด้านข้าง

4.3 การทดสอบการใช้งาน Raspberry Pi ผ่าน Touchscreen LCD Display



รูปที่ 4.8 การต่อ Raspberry Pi กับ Touchscreen LCD Display



รูปที่ 4.9 Touchscreen LCD Display

4.4 การทดสอบการรันโปรแกรมแบบ Background process กรณีอ่านบัตรเอ็นเอฟซี

4.4.1 เริ่มต้นการใช้งาน เปิดหน้าจอทำการเลือกชั้น



รูปที่ 4.10 หน้าจอเลือกชั้น

4.4.2 เมื่อทำการกดปุ่มจะทำการเปิดหน้าจอที่ 2 และรันโปรแกรมอ่านค่า NFC tag ในเวลาเดียวกัน



รูปที่ 4.11 หน้าจออ่านบัตร

4.4.3 เมื่อทำการอ่านบัตรเสร็จสิ้นจะเปลี่ยนข้อความสถานะ



รูปที่ 4.12 เปลี่ยนข้อความสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดสอบการรันโปรแกรมแบบ Background process กรณีถ่ายใบหน้า

4.5.1 เมื่อทำการอ่านบัตรเรียบร้อยแล้วเปิดหน้าจอที่ 3 ทันทีและรันโปรแกรมถ่ายรูปใบหน้าในเวลาเดียวกัน



รูปที่ 4.13 หน้าจอตรวจจับใบหน้า

4.5.2 เมื่อทำการถ่ายรูปใบหน้าเรียบร้อยแล้ว นำมาแสดงที่หน้าจอ



รูปที่ 4.14 หน้าจอตรวจจับใบหน้าที่ได้ถ่ายได้

4.6 การทดสอบสร้างเว็บเพื่อแสดงข้อมูลผู้เข้าออกอาคาร

4.6.1 สร้างตารางเพื่อที่จัดเก็บข้อมูลลงในฐานข้อมูล

Table	Action	Rows	Type	Collation	Size	Overhead
in_history	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	16 KiB	-
out_history	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	16 KiB	-
user_profile	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16 KiB	-
3 tables	Sum	17	InnoDB	utf8_general_ci	48 KiB	0 B

รูปที่ 4.15 ตารางในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 ขาเข้าข้อมูลที่ได้จากการแตะบัตรเข้าไปเก็บไว้ในตาราง

Show all | Number of rows: 25 ▼ | Filter rows: Search this table

+ Options

uid	floor	date_in
04c631b2412d80	12	2017-04-03 15:31:15
04c631b2412d80	13	2017-04-03 07:23:36
04c631b2412d80	14	2017-03-01 11:44:40
04c631b2412d80	10	2017-03-30 13:51:48
04c631b2412d81	9	2017-03-30 21:25:35
04c631b2412d81	6	2017-03-30 19:40:27
04c631b2412d81	5	2017-03-29 20:21:14

Show all | Number of rows: 25 ▼ | Filter rows: Search this table

รูปที่ 4.16 ค่าในตารางขาเข้า

4.6.3 ขาออกข้อมูลที่ได้จากการแตะบัตรออกไปเก็บไว้ในตาราง

1 ▼ | Show all > >> | Number of rows: 25 ▼

+ Options

uid	date_out
04c631b2412d80	2017-04-04 20:41:36
04c631b2412d81	2017-04-04 16:41:45
04c631b2412d80	2017-03-03 09:45:39
04c631b2412d80	2017-03-22 14:36:41

รูปที่ 4.17 ค่าในตารางขาออก

4.6.4 การนำข้อมูลที่ได้มาแสดงในเว็บ

← → ↻ 127.0.0.1:8080/chorth1.php

User history

UID	username	lastname	permission	date_created	date_edited	
04c631b2412d80	chorth	chorth	1	2017-04-04 00:00:00	2017-04-20 00:00:00	show history
04c631b2412d81	jay	jay	2	2017-04-10 00:00:00	2017-04-13 00:00:00	show history

รูปที่ 4.18 หน้าเว็บ

4.6.5 แสดงผลข้อมูลที่ประมวลผลออกมาในหน้าเว็บ

User history

UID	Modal Header	
	in_history	out_history
04c631b2412d80122017-04-03 15:31:15	2017-04-04 20:41:36	
04c631b2412d80132017-04-03 07:23:36	2017-03-03 09:45:39	
04c631b2412d80142017-03-01 11:44:40	2017-03-22 14:36:41	
04c631b2412d80102017-03-30 13:51:48	2017-03-28 18:52:48	

รูปที่ 4.19 ข้อมูลที่ประมวลผลออกมา



บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปฏิญานิพนธ์นี้เป็นการศึกษาและประกอบอุปกรณ์ขึ้นเป็นระบบการตรวจสอบคนเข้า-ออกอาคาร โดยในด้านของฝั่งส่งใช้บอร์ดราสเบอร์รี่พายเชื่อมต่อกับ NFC Card Reader, Webcam camera และ Touchscreen LCD Display ส่วนในด้านของฝั่งรับใช้บอร์ดราสเบอร์รี่พาย เชื่อมต่อกับ NFC Card Reader และ Monitor ใช้ในการแสดงผล ในขั้นการเริ่มทำงานคือผู้ใช้งานเลือกชั้นต่อนั้นทำการแตะบัตรและทำการถ่ายรูปจะมีการส่งข้อมูลระบุตัวตนไปยังฐานข้อมูลทันที เมื่อฐานข้อมูลได้รับข้อมูลจะทำการบันทึกไว้ตรวจสอบในขั้นตอนของการเข้าอาคารและการออกจากอาคารต่อไป โดยปฏิญานิพนธ์นี้มีจุดประสงค์หลัก คือ เพื่อประดิษฐ์อุปกรณ์ที่สามารถตรวจสอบคนเข้า-ออกอาคารได้อย่างครบถ้วนและปลอดภัย ด้วยขั้นตอนการยืนยันตัวตนด้วยการแตะบัตรการบันทึกใบหน้าและเพิ่มความสะดวกรวดมากยิ่งขึ้นด้วยการใช้โทรศัพท์มือถือแทนบัตรต่างๆได้

5.2 ข้อเสนอแนะ

ปฏิญานิพนธ์การตรวจสอบคนเข้า-ออกอาคารมีวิธีการยืนยันตัวตนด้วยการถ่ายรูปและอ่านเอ็นเอฟซีมีข้อจำกัดเล็กน้อย เช่น ในการถ่ายภาพนิ่งของผู้ใช้งานการตรวจจับของภาพยังไม่เป็นแบบเรียลไทม์ เนื่องจากประสิทธิภาพของอุปกรณ์

1. ออกแบบให้ทั้งฝั่งรับและฝั่งส่งเชื่อมต่อกันแบบไร้สาย

บรรณานุกรม

- [1] “RASPBERRY PI 3 MODEL B”
<https://www.raspberrypi.org>
- [2] “Raspberry Pi 7 inch Touchscreen Display”
<http://www.thaieasyelec.com/products/display/lcd-oled/raspberry-pi-7-inch-touchscreen-lcd-display-official-from-raspberrypi.org-detail.html>
- [3] “ตัวแปลภาษาและข้อผิดพลาด”
<https://darawan00.wordpress.com/>
- [4] “เปรียบเทียบ NFC กับ Bluetooth”
<http://www.lgmobilelover.com/home/what-difference-nfc-and-bluetooth/>
- [5] “การพัฒนาซอฟต์แวร์ด้วย PyQt4 บน Raspberry Pi (Jessie) ตอนที่ 1”
<http://raspberrypi-thailand.blogspot.com/2016/01/pyqt4-raspberry-pi-jessie-1.html>
- [6] “การติดต่อสื่อสารด้วย SPI : Serial Peripheral Interface”
<http://www.123microcontroller.com/Hardware-Interfacing/SPI-Serial-Peripheral-Interface-communication>
- [7] “Face Detection using Haar Cascades”
http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
- [8] “Webcam & Camera”
http://www.chomeweb.com/webcam_Other.html
- [9] “PN532 RFID Module”
http://linksprite.com/wiki/index.php5?title=PN532_RFID_Module.
- [10] “การสื่อสารสนามใกล้.”
<https://th.wikipedia.org/wiki/การสื่อสารสนามใกล้>.
- [11] Kevin Townsend. “Adafruit NFC/RFID on Raspberry Pi.”
<https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi/overview>.
- [12] Tony Dicola. “Raspberry Pi NFC Minecraft Blocks.”
<https://learn.adafruit.com/raspberry-pi-nfc-minecraft-blocks/hardware-wiring>.

- [13] Joeman. “Raspberry Pi 7” Touchscreen Display.”
<https://www.element14.com/community/docs/DOC-78156/v/raspberry-pi-7-touchscreen-display>.
- [14] Na-Rae Han. “File Reading and Writing Methods.”
http://www.pitt.edu/~naraehan/python2/reading_writing_methods.html.
- [15] Harrison Kinsley. “Haar Cascade Object Detection Face & Eye OpenCV Python Tutorial.”
<https://pythonprogramming.net/haar-cascade-face-eye-detection-python-opencv-tutorial>.
- [16] “Qt Tutorials For Beginners 22 – Qfile.”
<https://www.youtube.com/watch?v=tY6nW3Wm3NE&list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA&index=22>
- [17] “libnfc-1.7.1”
http://www.libnfc.org/api/examples_page.html.
- [18] “บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt”
<http://thaieasyelec.com/article-wiki/embedded-electronics-A2-qt.html>.
- [19] “Thread คืออะไร”
<http://www.mindphp.com/คู่มือ/73-คืออะไร/2264-thread-คืออะไร.html>
- [20] “OpenCV”
<http://www.willowgarage.com/pages/software/opencv>
- [21] “Web Server”
<http://www.thaicreate.com/web-host/web-host-web-server-web-hosting.html>, <http://www.thaicreate.com/web-host/web-host-apache-httpd.html>
- [22] “Apache (httpd) Web Server”
<http://www.thaicreate.com/web-host/web-host-apache-httpd.html>
- [23] “MySQL มีความสำคัญอย่างไรกับเซิร์ฟเวอร์”
<http://www.th.easyhostdomain.com/dedicated-servers/mysql.html>
- [24] “PHPmyadmin”
<http://www.krukanon.com/web/index.php -phpmyadmin>

- [25] “โครงสร้างของภาษา PHP”
<http://www.mwit.ac.th/~jeab/40201/ch3.php>
- [26] “เอแจ็กซ์”
<https://th.wikipedia.org/wiki/ajax>
- [27] “เว็บแอปพลิเคชันที่ใช้เอแจ็กซ์”
https://th.wikipedia.org/media/File:Ajax_Application_Model-th.png
- [28] “ภาษา html”
https://image.freepik.com/free-icon/html-file-with-code-symbol_318-45756.jpg





เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from PyQt4 import QtCore, QtGui
import sys
import time
import os

try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:

    def _fromUtf8(s):
        return s

try:
    _encoding = QtGui.QApplication.UnicodeUTF8

    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
except AttributeError:

    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

class Ui_MainWindow(object):

    def setupUi(self, MainWindow):
        MainWindow.setObjectName(_fromUtf8("MainWindow"))
        MainWindow.resize(800, 480)
        self.centralwidget = QtGui.QWidget(MainWindow)
        self.centralwidget.setObjectName(_fromUtf8("centralwidget"))
        self.labelp1 = QtGui.QLabel(self.centralwidget)
        self.labelp1.setGeometry(QtCore.QRect(330, 10, 151, 21))
        self.labelp1.setObjectName(_fromUtf8("labelp1"))
        self.btnfr1 = QtGui.QPushButton(self.centralwidget)
        self.btnfr1.setGeometry(QtCore.QRect(310, 60, 191, 81))
        self.btnfr1.setObjectName(_fromUtf8("btnfr1"))
        self.btnfr1.clicked.connect(MainWindow.close)# MainWindow closed when push button
        self.btnfr1.clicked.connect(self.openp2)#Open Page2 when push button
        self.btnfr2 = QtGui.QPushButton(self.centralwidget)
        self.btnfr2.setGeometry(QtCore.QRect(310, 160, 191, 81))
        self.btnfr2.setObjectName(_fromUtf8("btnfr2"))
        self.btnfr2.clicked.connect(MainWindow.close)# MainWindow closed when push button
        self.btnfr2.clicked.connect(self.openp2)#Open Page2 when push button
        self.btnfr3 = QtGui.QPushButton(self.centralwidget)
        self.btnfr3.setGeometry(QtCore.QRect(310, 250, 191, 81))
        self.btnfr3.setObjectName(_fromUtf8("btnfr3"))
        self.btnfr3.clicked.connect(MainWindow.close)# MainWindow closed when push button
        self.btnfr3.clicked.connect(self.openp2)# Open Page2 when push button
        self.btnclose = QtGui.QPushButton(self.centralwidget)
        self.btnclose.setGeometry(QtCore.QRect(660, 0, 111, 51))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.btnClose.clicked.connect(sys.exit)# System closed when push button

font = QtGui.QFont()
font.setPointSize(16)
self.btnClose.setFont(font)

self.btnClose.setObjectName(_fromUtf8("btnClose"))
MainWindow.setCentralWidget(self.centralwidget)
self.menuBar = QtGui.QMenuBar(MainWindow)
self.menuBar.setGeometry(QtCore.QRect(0, 0, 800, 31))
self.menuBar.setObjectName(_fromUtf8("menuBar"))
MainWindow.setMenuBar(self.menuBar)

self.statusBar = QtGui.QStatusBar(MainWindow)
self.statusBar.setObjectName(_fromUtf8("statusBar"))
MainWindow.setStatusBar(self.statusBar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
def openp2(self):
    f = open('data.txt','w')
    f.write("1\n")
    f.close()
    os.system("python ./p22.py")
def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow", None))
    self.btnfr3.setText(_translate("MainWindow", "3", None))
    self.label1.setText(_translate("MainWindow", "Select The Floor", None))
    self.btnfr2.setText(_translate("MainWindow", "2", None))
    self.btnfr1.setText(_translate("MainWindow", "1", None))
    self.btnClose.setText(_translate("MainWindow", "Close", None))
if __name__ == "__main__":
    app = QtGui.QApplication(sys.argv)
    MainWindow = QtGui.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```



ภาคผนวก ข

โค้ดโปรแกรมหน้าจอที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from PyQt4 import QtCore, QtGui
import os
import time
try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:
    def _fromUtf8(s):
        return s
try:
    _encoding = QtGui.QApplication.UnicodeUTF8
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)
class WorkThread(QtCore.QThread):
    def __init__(self):
        QtCore.QThread.__init__(self)
    def run(self):
        print('Please Scan your Card ...')
        from nfc import nfcroot
        temp = nfcroot()
        f = open("data.txt", "a")
        f.write(temp)
        f.close()
        self.emit(QtCore.SIGNAL('status')) # Send 'status' signal
        time.sleep(2)
        self.emit(QtCore.SIGNAL('open')) # Send 'open' signal
class Ui_Form1(QtGui.QMainWindow):
    def setupUi(self, Form1):
        self.mythread = WorkThread()
        self.mythread.start() # Start background processing
        Form1.setObjectName(_fromUtf8("Form1"))
        Form1.resize(800, 480)
        self.btnback = QtGui.QPushButton(Form1)
        self.btnback.setGeometry(QtCore.QRect(20, 390, 181, 71))
        font = QtGui.QFont()
        font.setPointSize(20)
        self.btnback.setFont(font)
        self.btnback.setObjectName(_fromUtf8("btnback"))
        self.btnback.clicked.connect(Form1.close) #Form1 closed when pushbutton
        self.btnback.clicked.connect(self.backpage) #Open Page1 when pushbutton

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.labelp2 = QtGui.QLabel(Form1)
self.labelp2.setGeometry(QtCore.QRect(50, 80, 711, 241))
font = QtGui.QFont()
font.setPointSize(60)
self.labelp2.setFont(font)
self.labelp2.setObjectName(_fromUtf8("labelp2"))
self.retranslateUi(Form1)
QtCore.QMetaObject.connectSlotsByName(Form1)
self.connect(self.mythread, QtCore.SIGNAL('status').self.statusshow)#Change text when detected 'status' signal
self.connect(self.mythread, QtCore.SIGNAL('open').self.openp3) #Open Page3 when detected 'open' signal
def backpage(self):
    os.system("python ./p11.py")
def retranslateUi(self, Form1):
    Form1.setWindowTitle(_translate("Form1", "Form1", None))
    self.btnback.setText(_translate("Form1", "Back", None))
    self.labelp2.setText(_translate("Form1", "Scan your NFC card ", None))
    @QtCore.pyqtSlot()
def statusshow(self):
    self.labelp2.clear()
    self.labelp2.setText("Finish!!")
    @QtCore.pyqtSlot()
def openp3(self):
    os.system("python ./p33.py") #Run p33.py
    Form1.close()
if __name__ == "__main__":
    import sys
    app = QtGui.QApplication(sys.argv)
    Form1 = QtGui.QWidget()
    ui = Ui_Form1()
    ui.setupUi(Form1)
    Form1.show()
    sys.exit(app.exec_())

```



ภาคผนวก ค

โค้ดโปรแกรมหน้าจอตีสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from PyQt4 import QtCore, QtGui
import sys
import os
import time

try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:
    def _fromUtf8(s):
        return s

try:
    _encoding = QtGui.QApplication.UnicodeUTF8
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

class WorkThread(QtCore.QThread):
    def __init__(self):
        QtCore.QThread.__init__(self)
    def run(self):
        print('Initializing')
        from camera import cameraroot # Call camera.py class cameraroot
        temp = cameraroot() #Receive parameters from return value
        self.emit(QtCore.SIGNAL('detected'))
        time.sleep(2)
        self.emit(QtCore.SIGNAL('open'))

class Ui_Form2(QtGui.QMainWindow):
    def setupUi(self, Form2):
        self.mythread = WorkThread()
        self.mythread.start()
        Form2.setObjectName(_fromUtf8("Form2"))
        Form2.resize(800, 480)
        self.labelp3 = QtGui.QLabel(Form2)
        self.labelp3.setGeometry(QtCore.QRect(150, 80, 551, 271))
        font = QtGui.QFont()
        font.setPointSize(60)
        self.labelp3.setFont(font)
        self.labelp3.setObjectName(_fromUtf8("labelp3"))
        self.btnhome = QtGui.QPushButton(Form2)
        self.btnhome.setGeometry(QtCore.QRect(30, 390, 181, 71))
        font = QtGui.QFont()
        font.setPointSize(20)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self btnhome.setFont(font)
self btnhome.setObjectName(_fromUtf8("btnhome"))
self btnhome.clicked.connect(sys.exit)#
self btnhome.clicked.connect(self.homepage)#Open Page1 when pushed this button
self.retranslateUi(Form2)
QtCore.QMetaObject.connectSlotsByName(Form2)
self.connect(self.mythread, QtCore.SIGNAL('detected'),self.imageshow)#Show the image from face snap
self.connect(self.mythread, QtCore.SIGNAL('open'),self.openp1)#Open Page1 when detect 'open' signal
def homepage(self):
    os.system("python ./p1 1.py")
def retranslateUi(self, Form2):
    Form2.setWindowTitle(_translate("Form2", "Form2", None))
    self.labelp3.setText(_translate("Form2", "Face Snapshot", None))
    self.btnhome.setText(_translate("Form2", "HOME", None))
@QtCore.pyqtSlot()
def imageshow(self):
    print("Image show")
    timeis = time.localtime() #import the raspberry pi localtime
    a = time.strftime('%A %B %d %Y %H_%M',timeis) #import Day Date Time
    self.labelp3.clear()
    imagefeed3 = QtGui.QImage("/home/pi/4-1/project/project2/"+str(a)+".jpg") #Show image by name
    self.labelp3.setPixmap(QtGui.QPixmap(imagefeed3)) #Change text to image
@QtCore.pyqtSlot()
def openp1(self):
    print("Page1 show")
    os.system("python ./p1 1.py")
    sys.exit()
if __name__ == "__main__":
    import sys
    app = QtGui.QApplication(sys.argv)
    Form2 = QtGui.QWidget()
    ui = Ui_Form2()
    ui.setupUi(Form2)
    Form2.show()
    sys.exit(app.exec_())

```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def nferoot():
    import binascii
    import sys
    import Adafruit_PN532 as PN532
    import cv2
    import numpy as np
    import time

    CS = 18
    MOSI = 23
    MISO = 24
    SCLK = 25

    pn532 = PN532.PN532(cs=CS, sclk=SCLK, mosi=MOSI, miso=MISO)
    pn532.begin()
    pn532.SAM_configuration()
    print('Waiting for Authentication card...!')
    while True:
        uid = pn532.read_passive_target() #Save uid from NFC reading
        if uid is None:
            continue
        print 'Reading UID is : %s' % (binascii.hexlify(uid)) #Transfer binary to hex (uid type)
        uid2 = '%s' %(binascii.hexlify(uid))
        return uid2
        break

```



ภาคผนวก จ

โค้ดโปรแกรมถ่ายรูปใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def cameraroot():
    import cv2
    import numpy as np
    import time
    import imutils

    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') #Use OpenCV to detect Face
    cap = cv2.VideoCapture(0) #Read image from Webcam
    i = int()
    i == 0

    while True:
        ret, img = cap.read()

        img = imutils.resize(img,width=300)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #Change color background to grey background
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        timeis = time.localtime()
        a = time.strftime('%A %B %d %Y %H_%M',timeis)
        for (x,y,w,h) in faces:
            cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0), 5)
        cv2.imshow('FaceDetection',img)
        if len(faces) != 0: #If detected Face
            if (faces[0][2]>100) and (faces[0][3]>100):
                i += 1
                if i == 2:
                    cv2.imwrite("face-"+str(a)+".jpg", img) #Save the snap
                    return i
                    break
    cap.release() #End video streaming
    cv2.destroyAllWindows()

```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?php
//โปรแกรมสำหรับการประมวลผลข้อมูลที่เข้ามาใหม่
$link = mysql_connect('127.0.0.1', 'root', 'anundadb');
if (!$link) {
    die('Could not connect: ' . mysql_error());
} else {
    $objDB = mysql_select_db('nfc_database');
    mysql_query("SET NAMES UTF8");
}
//
Sid_1 = $_POST['id'];
function getticket($Sid_1){
    $get_ticket = "SELECT * FROM `in_history` WHERE `uid` = '$Sid_1'";
    $get_ticket_query = mysql_query($get_ticket);
    while ($get_ticket_object = mysql_fetch_assoc($get_ticket_query)) {
        $return_result[] = $get_ticket_object;
    }
    return $return_result;
}
function getticket1($Sid_1){
    $get_ticket = "SELECT * FROM `out_history` WHERE `uid` = '$Sid_1'";
    $get_ticket_query = mysql_query($get_ticket);
    while ($get_ticket_object = mysql_fetch_assoc($get_ticket_query)) {
        $return_result[] = $get_ticket_object;
    }
    return $return_result;
}
$get_ticket['in'] = getticket($Sid_1);
$get_ticket['out'] = getticket1($Sid_1);
echo json_encode($get_ticket);
?>

```