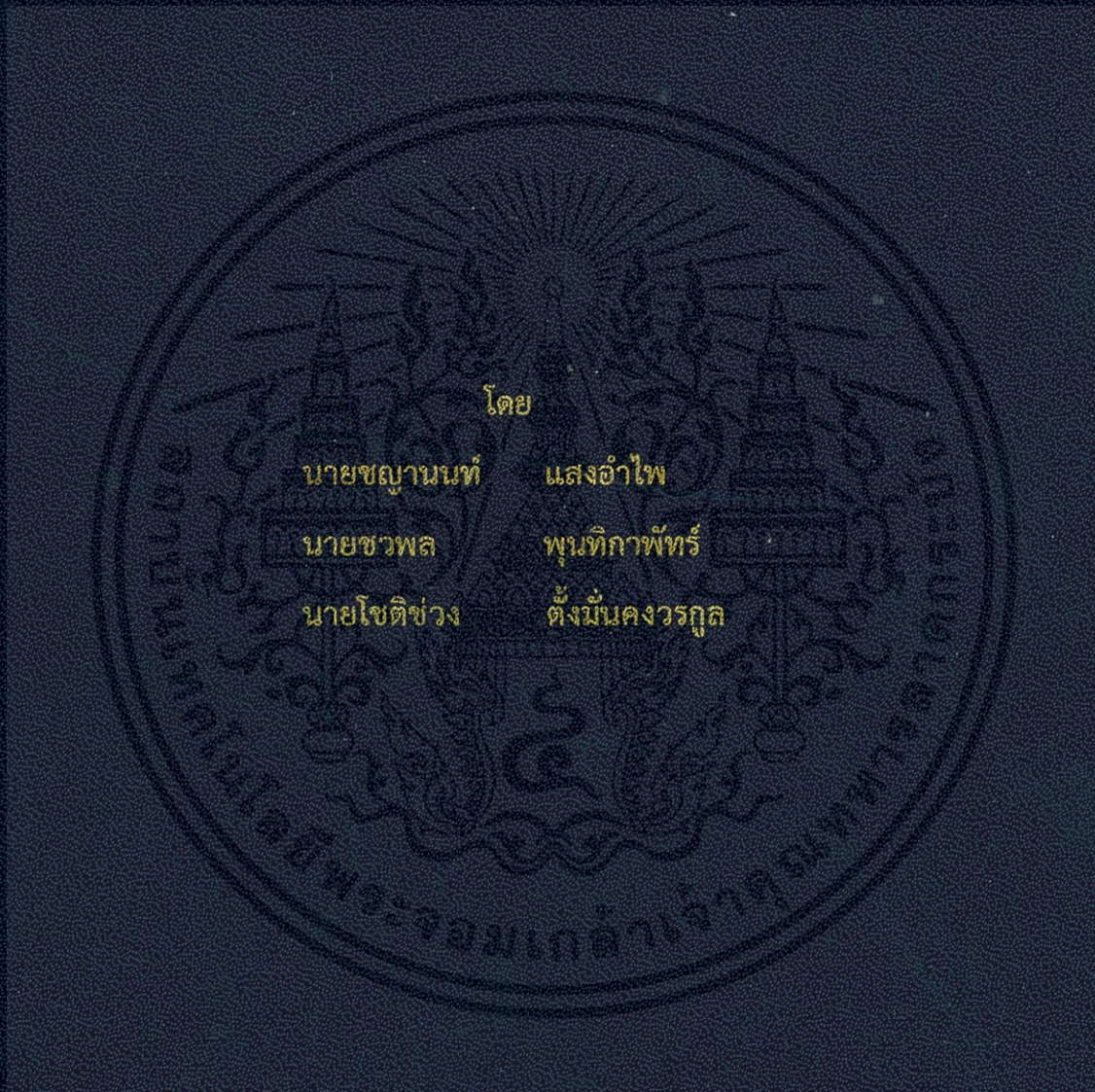


ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน SMART PHONE
APPLIANCES CONTROLLED USING EMBEDDED SYSTEM VIA
SMART PHONE



โดย

นายชญาณนท์

แสงอำไพ

นายชวพล

พูนทิกาพัทร์

นายโชติช่วง

ตั้งนันทวงกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน SMART PHONE
APPLIANCES CONTROLLED USING EMBEDDED SYSTEM VIA
SMARTPHONE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน SMARTPHONE
APPLIANCES CONTROLLED USING EMBEDDED SYSTEM VIA SMARTPHONE

โดย

นายชฎานนท์	แสงอำไพ	53010281
นายชวพล	พุนทิกาพัทธ์	53010327
นายโชติช่วง	ตั้งมั่นคงวรกุล	53010390

อาจารย์ที่ปรึกษา
รศ.ดร.พิพัฒน์ พรหมมี

อาจารย์ที่ปรึกษาร่วม
รศ. สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556



ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน SMART PHONE

APPLIANCES CONTROLLED USING EMBEDDED SYSTEM VIA SMARTPHONE

ผู้จัดทำ

- | | | |
|---|----------------------------|----------|
| 1 | นายชยานนท์ แสงอำไพ | 53010281 |
| 2 | นายชวพล พุนทิกาพัทธ์ | 53010327 |
| 3 | นายโชติช่วง ตังมั่นคงวรกุล | 53010390 |


..... อาจารย์ที่ปรึกษา
รศ. ดร. พิชิตน์ พรหมมี


..... อาจารย์ที่ปรึกษาร่วม
รศ. สมยศ จุณณะปิยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์เรื่องนี้ จะไม่สำเร็จได้หากขาดบุคคลเหล่านี้ จึงขอขอบคุณผู้ปกครองที่คอยให้กำลังใจ ให้คำปรึกษา ขอขอบคุณ รศ.ดร. พิพัฒน์ พรหมมี และ รศ.สมยศ จุณณะปิยะ ซึ่งเป็นอาจารย์ที่ปรึกษาที่คอยให้คำแนะนำ กำลังใจ ในการค้นคว้า ทำวิทยานิพนธ์ครั้งนี้ และขอขอบคุณบรรดาเพื่อนๆ ในคณะที่คอยช่วยเหลือในทุกเรื่องในการศึกษาที่สถาบันการศึกษาแห่งนี้



นาย ชญานนท์ แสงอำไพ
นาย ชวพล พุนทิกาพัทธ์
นาย โชติช่วง ตังมั่นคงวรกุล
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน SMART PHONE
 APPLIANCES CONTROLLED USING EMBEDDED SYSTEM VIA SMARTPHONE

โดย นาย ชญานนท์ แสงอำไพ 53010281
 นาย ชวพล พุนทิกาพัทร์ 53010327
 นาย โชติช่วง ตั้งมันคงวรกุล 53010390

อาจารย์ที่ปรึกษา รศ. ดร. พิพัฒน์ พรหมมี
 อาจารย์ที่ปรึกษาร่วม รศ. สมยศ จุณณะปิยะ

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อศึกษาและสร้างระบบควบคุมอุปกรณ์ไฟฟ้าด้วยสมองกลฝังตัวผ่านแอปพลิเคชันแอนดรอยด์ บน SMARTPHONE ซึ่งประกอบด้วย บอร์ด RASPBERRY PI SMARTPHONE และ ชุดรีเลย์สวิตช์ สามารถนำไปควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้านได้อย่างอิสระ โดยมีการติดต่อสื่อสารกันระหว่างสมองกลฝังตัวกับแอปพลิเคชันบน SMARTPHONE เพื่อนำไปควบคุมชุดรีเลย์ให้สั่งงานเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้านได้ตามต้องการ

ABSTRACT

This project presents an appliances controlled using embedded system via Smartphone. It contains a Raspberry Pi board, Smartphone and Relay switch. The system can be used to control the electrical appliance switching's independently. The embedded communicates with android application on Smartphone.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VI
บทที่ 1	1
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	2
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 RASPBERRY PI	2
2.1.1 คุณสมบัติทางเทคนิค	2
2.1.2 การเริ่มต้นใช้งานบอร์ด RASPBERRY PI	3
2.1.3 การใช้งานกับระบบเครือข่ายและเชื่อมต่ออินเทอร์เน็ต	3
2.1.4 การจัดขาของ GPIO	4
2.2 รีเลย์	5
2.2.1 โครงสร้างของรีเลย์	5
2.2.2 หลักการทำงานของรีเลย์	6
2.3 ภาษา PHP	6
2.3.1 การนำภาษา PHP ไปใช้งาน	7
2.3.2 การทำงานของภาษา PHP กับ WEB SERVER	7
2.3.3 ข้อดีและข้อจำกัดของภาษา PHP	7
2.4 ภาษา PYTHON	7
2.4.1 หลักการทำงานของภาษา PYTHON	7
2.4.1.1 คอมไพเลอร์ (COMPILER)	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.1.2 อินเทอร์พรีเตอร์ (INTERPERTER)	8
2.4.2 จุดเด่นของภาษาไพธอน (PYTHON)	8
2.5 การจัดการฐานข้อมูล MYSQL	9
2.5.1 ความรู้พื้นฐานเกี่ยวกับUSER-NAME และรหัสผ่านใน MYSQL	9
2.5.2 การใช้งาน MYSQL	9
2.6 WEB SERVER	10
2.6.1 หลักการทำงาน WEB SERVER	10
2.7 ระบบปฏิบัติการแอนดรอยด์ (ANDROID)	10
2.7.1 สถาปัตยกรรมแอนดรอยด์ (ANDROID ARCHITECTURE)	11
2.7.1.1 APPLICATION	12
2.7.1.2 APPLICATION FRAMEWORK	12
2.7.1.3 LIBRARIES	12
2.7.1.4 LINUX KERNEL	13
2.8 ภาษา JAVA	13
2.8.1 คุณลักษณะเด่นของภาษา JAVA	13
2.8.2 รูปแบบของภาษา JAVA	14
2.8.3 ซอร์สโค้ด (SOURCE CODE)	14
บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์	15
3.1 การออกแบบ	15
3.1.1 การออกแบบทางฮาร์ดแวร์	16
3.1.2 การออกแบบทางซอฟต์แวร์	18
3.1.2.1 การออกแบบทางซอฟต์แวร์ของฝั่ง CLIENT	18
3.1.2.2 การออกแบบทางซอฟต์แวร์ของฝั่ง SERVER	19
3.2 เครื่องมือที่ใช้ในการทดลอง	20
3.2.1 RASPBERRY PI (บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก)	20
3.2.2 GPIO	20

สารบัญ (ต่อ)

	หน้า
3.2.3 จอแสดงผล	21
3.2.4 KEY-BOARD & MOUSE	21
3.2.5 สาย LAN	21
3.2.6 ชุด RELAY SWITCH	22
3.2.7 ชุดวงจร RELAY SWITCH	22
3.2.8 SMART PHONE	22
3.2.9 ชุดวงจร OPTO-ISOLATED RELAY	23
3.3 การจัดเก็บผลการทดลอง	23
บทที่ 4 ผลการทดลอง	25
4.1 ตรวจสอบ IP ADDRESS ทั้งทางฝั่ง CLIENT และ SERVER	25
4.2 ตรวจสอบการเชื่อมต่อกันระหว่าง SERVER และ CLIENT	26
4.3 ตรวจสอบ Protocol สื่อสารระหว่าง CLIENT และ SERVER	26
4.4 CONFIG ค่าต่างๆ ของกล้อง WEBCAM	27
4.5 เริ่มการใช้งานกล้อง WEBCAM	28
4.6 เริ่มต้นการทำงาน ANDORID APPLICATION	29
4.7 เก็บผลการทดลองจากระบบฐานข้อมูล	31
4.8 จัดเก็บผลการทดลองที่ได้จากหน้าต่าง LXTERMINAL	31
4.9 จัดเก็บผลการทดลองที่ได้จาก GPIO	32
4.10 ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า	32
4.11 สัญญาณ TTL ที่วัดได้หลังจากผ่านวงจรตรวจจับแรงดัน 220V	34
4.12 ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า	35
4.13 เปรียบเทียบผลที่ได้จากการทดลองเมื่อป้อนข้อมูลต่างๆ	36
บทที่ 5 สรุปผลและข้อเสนอแนะ	41
5.1 สรุปผล	41
5.2 ข้อเสนอแนะ	41
บรรณานุกรม	42
ภาคผนวก	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	RASPBERRY PI	2
2.2	การเชื่อมต่อ RASPBERRY PI เข้ากับ INTERNET	3
2.3	โปรแกรม LXTERMINAL	4
2.4	การใช้คำสั่ง SUDO APT-GET UPGRADE บน LXTERMINAL	4
2.5	การจัดเรียงขา GPIOแบบมาตรฐาน	5
2.6	การจัดเรียงขา GPIOเมื่อใช้งานกับโปรแกรม WIRINGPI	5
2.7	รีเลย์ที่ใช้ในงาน	5
2.8	สัญลักษณ์ของรีเลย์	5
2.9	สัญลักษณ์ของรีเลย์แทนโครงสร้างรีเลย์	6
2.10	สถานะการทำงานของรีเลย์	6
2.11	การทำงานคอมไพเลอร์ (COMPILER)	8
2.12	การทำงาน อินเทอร์พรีเตอร์ (INTERPERTER)	8
2.13	การทำงานของ WEB SERVER	10
2.14	สถาปัตยกรรมแอนดรอยด์	11
3.1	บล็อกไดอะแกรมโดยรวมของระบบการทำงาน	15
3.2	การออกแบบทางฮาร์ดแวร์ส่วนของชุดวงจรรีเลย์สวิตช์	16
3.3	การออกแบบทางฮาร์ดแวร์ของวงจรตรวจจับแรงดัน 220V ISOLATED	17
3.4	การออกแบบทางซอฟต์แวร์ของฝั่ง CLIENT	18
3.5	การออกแบบทางซอฟต์แวร์ของฝั่ง SERVER	19
3.6	แสดงส่วนประกอบของบอร์ด RASPBERRY PI	20
3.7	การจัดเรียงขา GPIOแบบมาตรฐาน	20
3.8	การจัดเรียงขา GPIOเมื่อใช้งานกับโปรแกรม WIRINGPI	20
3.9	จอแสดงผล	21
3.10	KEYBOARD & MOUSE	21
3.11	สาย LAN	21
3.12	กล้อง WEB CAMERA	22
3.13	ชุด RELAY SWITCH	22
3.14	SMART PHONE	22
3.15	ชุดวงจรตรวจจับแรงดัน 220V ISOLATED	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.1	IP ADDRESS ของฝั่ง CLIENT (ANDROID APPLICATION)	25
4.2	IP ADDRESS ของฝั่ง SERVER (สมองกลฝั่งตัว)	25
4.3	ผลลัพธ์ที่ได้จากคำสั่ง PING	26
4.4	ผลการตรวจจับการรับ-ส่งข้อมูลด้วยโปรแกรม WIRESHARK	27
4.5	หน้าต่างโปรแกรม WIRESHARKตรวจจับข้อมูล PACKAGE ต่างๆ	27
4.6	คำสั่ง NANO GST-SERVER.SH บนหน้าจอ TERMINAL	28
4.7	การ CONFIG ค่าของกล้อง WEBCAM	28
4.8	สถานะของกล้อง WEBCAM พร้อมใช้งาน	29
4.9	หน้าจอ INTERFACE บน ANDROID APPLICATION	29
4.10	หน้าจอ INTERFACE สำหรับควบคุมอุปกรณ์ไฟฟ้า	30
4.11	การป้อนค่า INPUT ต่างๆเข้าไปบนหน้าจอ INTERFACE	30
4.12	ผลลัพธ์ที่ได้จากระบบฐานข้อมูล	31
4.13	ผลลัพธ์ที่ได้บนหน้าจอ LXTERMINAL บนสมองกลฝั่งตัว	31
4.14	ผลลัพธ์ที่ได้จาก GPIO ของสมองกลฝั่งตัว	32
4.15	วงจรตรวจจับแรงดัน 220V ISOLATED	32
4.16	MONITOR FIELD ในระบบฐานข้อมูล	33
4.17	สถานะของอุปกรณ์ไฟฟ้าชนิดต่างๆ	33
4.18	สถานะล่าสุดที่กดปุ่มควบคุมช่องทางต่างๆ	34
4.19	ระดับสัญญาณดิจิตอลที่วัดได้เมื่ออุปกรณ์ไฟฟ้าทำงาน	34
4.20	ระดับสัญญาณดิจิตอลที่วัดได้เมื่ออุปกรณ์ไฟฟ้าไม่ทำงาน	35
4.21	สถานะพัดลมทำงานในระดับต่ำ	36
4.22	กดปุ่ม FAN HIGH บนหน้าแอปพลิเคชันแอนดรอยด์	36
4.23	LED ช่องทางที่ 2 ติด	37
4.24	พัดลมติดในระดับสูง (FAN HIGH)	37
4.25	ปรากฏสถานะของอุปกรณ์ไฟฟ้าเมื่อกดปุ่ม MONITOR	38
4.26	เมื่อกดปุ่ม LIGHT1: ON บนหน้าแอปพลิเคชันแอนดรอยด์	38
4.27	สถานะ LED ช่องทางที่ 3 ติด	39
4.28	หลอดไฟดวงที่ 1 ติด	39
4.29	ปรากฏสถานะของอุปกรณ์ไฟฟ้าชนิดต่างๆเมื่อกดปุ่ม MONITOR	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การอธิบายส่วนประกอบของ APPLICATION FRAMEWORK	12
2.2 การอธิบายส่วนประกอบของ LIBRARIES	13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในปัจจุบันอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆ ภายในบ้านจะควบคุมการทำงาน เปิด - ปิด จาก ระยะใกล้เท่านั้นทำให้เกิดปัญหาในการควบคุมการ เปิด - ปิด ในกรณีที่อยู่ห่างไกลจากเครื่องใช้ไฟฟ้านั้นๆ ในบางครั้งที่มีความจำเป็น ยกตัวอย่างเช่น บุคคลส่วนใหญ่รีบเร่งออกไปทำงาน บางครั้งจึงลืมปิดเครื่องใช้ไฟฟ้าในบ้าน ส่งผลให้เครื่องใช้ไฟฟ้าเกิดความเสียหาย หรือเกิดความอันตรายต่อทรัพย์สิน และเปลืองค่าใช้จ่าย โครงการเรื่อง “ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone” เป็นโครงการที่พัฒนาขึ้นเพื่อแก้ปัญหาข้างต้น โดยจัดทำอุปกรณ์ควบคุมการ เปิด-ปิด ตามจุดต่างๆ ได้อย่างอิสระ หรือใช้ทำเป็นชุดควบคุม เพื่ออำนวยความสะดวกสบายในการ เปิด-ปิด เครื่องใช้ไฟฟ้าต่างๆ ได้ไม่จำกัดระยะทาง โดยพัฒนาโปรแกรมควบคุมจากแอปพลิเคชันแอนดรอยด์ (Client) แล้วส่งข้อมูลไปยังเครื่องสมองกลฝังตัว (Server) เพื่อควบคุม GPIO ของเครื่องสมองกลฝังตัว นอกจากนี้ยังสามารถแสดงสถานะการทำงานของเครื่องใช้ไฟฟ้าแบบภาพต่อเนื่องผ่าน Smart Phone ได้อีกด้วย

1.2 วัตถุประสงค์

1. สามารถเขียนโปรแกรมสำหรับสร้างและพัฒนาแอปพลิเคชันแอนดรอยด์
2. สามารถเขียนโปรแกรมเพื่อควบคุม GPIO ของเครื่องสมองกลฝังตัว
3. สามารถเขียนโปรแกรมเพื่อสร้างระบบฐานข้อมูลบนเว็บเซิร์ฟเวอร์ได้
4. สามารถเชื่อมต่อและส่งข้อมูลจากแอปพลิเคชันแอนดรอยด์เข้าไปไว้ในระบบฐานข้อมูลได้
5. สามารถเชื่อมต่อระหว่างระบบฐานข้อมูลและ GPIO ของสมองกลฝังตัวได้
6. สามารถควบคุม GPIO บนเครื่องสมองกลฝังตัวจากแอปพลิเคชันแอนดรอยด์ผ่านระบบฐานข้อมูลได้
7. สามารถควบคุมอุปกรณ์ไฟฟ้าบน Smart Phone ผ่านระบบอินเทอร์เน็ต ได้จากระยะไกล

1.3 ขอบเขตของโครงการ

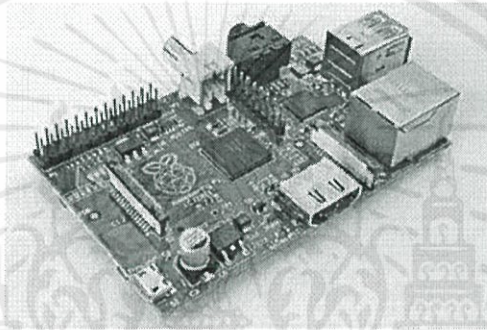
โครงการนี้ จัดทำขึ้นเพื่อสามารถควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าจากแอปพลิเคชันแอนดรอยด์โดยจะมีการส่งค่าจากแอปพลิเคชันแอนดรอยด์ไปยังระบบฐานข้อมูลในส่วนของ Client ผ่าน URL บน Web browser จากนั้น ในส่วนของ Sever (สมองกลฝังตัว) จะทำการตรวจสอบค่าที่ได้รับมาในระบบฐานข้อมูลเพื่อใช้ในการควบคุม GPIO ของสมองกลฝังตัวต่อไป

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry pi

Raspberry คือบอร์ดไมโครคอมพิวเตอร์แบบแผ่นเดียวที่บรรจุความสามารถเพื่อรองรับระบบปฏิบัติการ Linux ที่ถูกบรรจุลงใน SD การ์ดสำหรับการพัฒนาไปสู่บอร์ด Embedded Linux ซึ่งประกอบด้วยจุดเชื่อมต่ออุปกรณ์อินพุต, เอาต์พุตทั้งผ่านพอร์ต USB, LAN, HDMI, ช่องสัญญาณภาพ และ GPIO สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์ ซึ่งแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 Raspberry pi

(แหล่งที่มา http://en.wikipedia.org/wiki/Raspberry_Pi)

2.1.1 คุณสมบัติทางเทคนิค

- ชิปควบคุมหลัก (Broadcom BCM2835) หรือเทียบเท่าซึ่งรวมซีพียู, หน่วยประมวลผลกราฟิกหรือ GPU และหน่วยความจำ SDRAM ไว้ภายในตัวถังเดียวกัน
- หน่วยประมวลผลกลางหรือ CPU: ARM11 คอร์ ARM1176JZF-S ความเร็ว 700MHz
- หน่วยประมวลผลกราฟิกหรือ GPU: Broadcom Video Core IV หรือเทียบเท่าใช้สำหรับรองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
- หน่วยความจำ SDRAM: 512 MB
- จุดต่อ: USB 2.0 (2 พอร์ต), แจ็ก RCA และ HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับ โทรทัศน์หรือจอแสดงผล ที่มีจุดต่อแบบ RCA ตัวเมียหรือ HDMI, จุดต่อเอาต์พุตเสียงเป็นแจ็กหูฟัง 3.5 มิลลิเมตร, จุดต่อ internet หรือจุดต่อระบบ LAN, connector หรือจุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input / Output: GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I2C, I2S, ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และ socket ของ SD การ์ดสำหรับเสียบ SD การ์ดที่ติดตั้งระบบปฏิบัติการเรียบร้อยแล้ว
- ความต้องการไฟเลี้ยง: +5V 700mA เป็นอย่างน้อย
- ขนาด: 85.60 x 53.98 มิลลิเมตร หรือ 3.370 x 2.125 นิ้ว

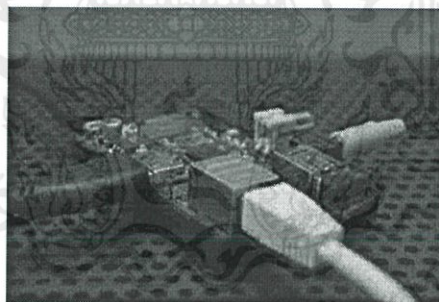
2.1.2 การเริ่มต้นใช้งานบอร์ด Raspberry Pi

ต้องเตรียมอุปกรณ์ดังนี้

- 1) บอร์ด Raspberry Pi
- 2) SDการ์ดสำหรับเก็บข้อมูลรวมไปถึงระบบปฏิบัติการควรมีความจุตั้งแต่ 4GB คลาส 4 ขึ้นไปซึ่งก็คือ SDHC การ์ดจะเป็นแบบ FAT32 หรือ NTFS ก็ได้
- 3) สาย microUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V 700mA ที่มีจุดต่อแบบ USB ไม่ควรใช้แหล่งจ่ายไฟจากพอร์ต USB ของคอมพิวเตอร์ เนื่องจากมี ความสามารถในการจ่ายกระแสไฟฟ้าไม่เพียงพอ
- 4) สาย HDMI หรือ RCA (อย่างใดอย่างหนึ่ง) สำหรับต่อกับจอแสดงผล
- 5) คีย์บอร์ดและเมาส์แบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi
- 6) สาย LAN (มีหรือไม่มีก็ได้) สำหรับเชื่อมต่อกับอินเทอร์เน็ต
- 7) ลำโพงที่มีสายต่อเป็นปลั๊กหูฟัง 3.5 มม. เพื่อต่อกับข้อเอาต์พุตสัญญาณเสียงของบอร์ดRaspberry Pi (มีหรือไม่มีก็ได้)

2.1.3 การใช้งานกับระบบเครือข่ายและเชื่อมต่ออินเทอร์เน็ตแบบมีสาย

- เมื่อต้องการเชื่อมต่ออินเทอร์เน็ต ให้ต่อสายLANเข้าที่จุดต่อ LAN ซึ่งเป็นคอนเน็กเตอร์ RJ-45 จะใช้งานได้ทันทีโดยไม่ต้องทำการตั้งค่าใดๆ (ทั้งนี้ผู้ใช้งานต้องมีการเปิดใช้บริการอินเทอร์เน็ตจากผู้ให้บริการด้วย) ซึ่งแสดงได้ดังรูปที่ 2.2

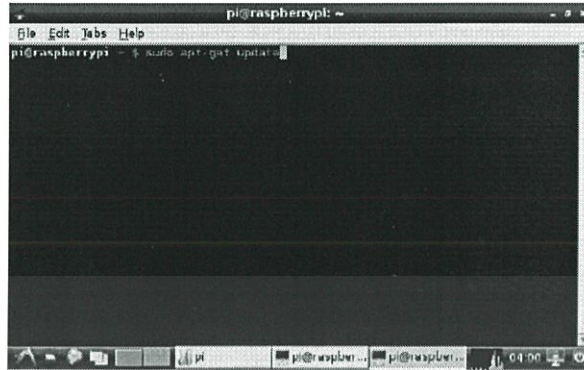


รูปที่ 2.2 การเชื่อมต่อ Raspberry pi เข้ากับ Internet

(แหล่งที่มา <http://www.tuicool.com/articles/ZBbque>)

- เมื่อเชื่อมต่ออินเทอร์เน็ตแล้วให้เปิดโปรแกรม LXTerminal ขึ้นมาแล้วใช้คำสั่ง `sudo apt-get updates` เพื่อทำการอัปเดตโปรแกรมล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โปรแกรม LXTerminal

- ใช้คำสั่ง `sudo apt-get upgrade` กดปุ่ม Y ตามด้วยกดปุ่ม Enter เพื่อยืนยันแล้วรอดาว์โหลดไฟล์ จากนั้นระบบจะถามว่าต้องการทำอะไรกับแพ็คเกจให้กดปุ่ม Y แล้ว Enter เพื่อติดตั้งทับไฟล์เดิม แล้วรอนจนกระทั่งการอัปเดตเสร็จสิ้น



รูปที่ 2.4 การใช้คำสั่ง `sudo apt-get upgrade` บน LXTerminal

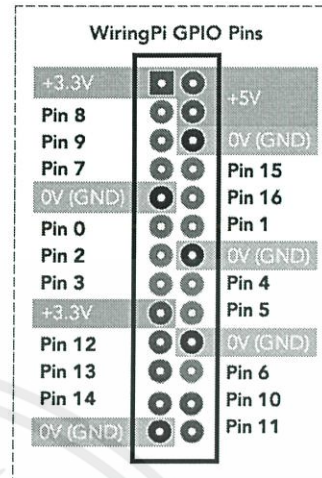
2.1.4 การจัดการของ GPIO

บอร์ด Raspberry Pi มีขาอินพุต เอาต์พุต ดิจิตอลหรือ GPIO สำหรับติดต่อกับอุปกรณ์ภายนอกรวม 17 ขาโดยมีการจัดเรียงขามาตรฐานตามรูปที่ 2.5 และหากใช้โปรแกรม WiringPi จะมีการจัดเรียงขาอินพุต เอาต์พุตใหม่ตามรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การจัดเรียงขา GPIO แบบมาตรฐาน



รูปที่ 2.6 การจัดเรียงขา GPIO เมื่อใช้งานกับโปรแกรม WiringPi

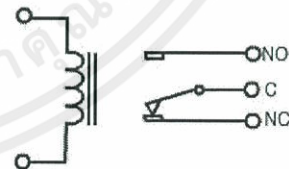
(แหล่งที่มา <http://pi.gadgetoid.com/article/raspberry-pi-gpio-pinout-guide>)

2.2 รีเลย์

รีเลย์ (Relay) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ ตัด-ต่อวงจร คล้ายกับสวิตช์ โดยใช้หลักการหน้าสัมผัส และการที่จะให้มันทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนดเพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้มัน มันก็จะกลายเป็นวงจรเปิดไฟที่ใช้ข้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจากเพาเวอร์ฯของเครื่อง ดังนั้นทันทีที่เปิดเครื่องก็จะทำให้รีเลย์ทำงาน



รูปที่ 2.7 รีเลย์ที่ใช้ในงาน



รูปที่ 2.8 สัญลักษณ์ของรีเลย์

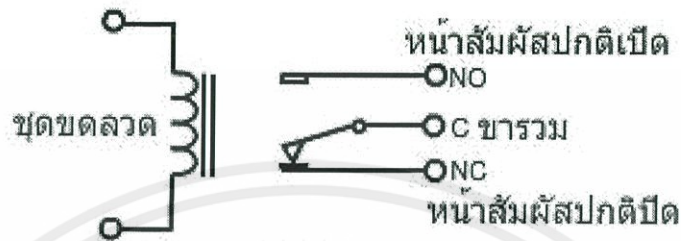
(แหล่งที่มา <http://www.smartlearningweb.com/knowledge/relay/relay.htm>)

2.2.1 โครงสร้างของรีเลย์

ภายในโครงสร้างของ รีเลย์ จะประกอบไปด้วยขดลวด 1 ชุด และ หน้าสัมผัส ซึ่งในหน้าสัมผัส 1 ชุด ซึ่งจะประกอบไปด้วยหน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC.) ซึ่งในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (C) และ หน้าสัมผัสแบบปกติเปิด (Normally Open หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งผู้จัดทำเอกสารหากมีการละเมิดลิขสิทธิ์

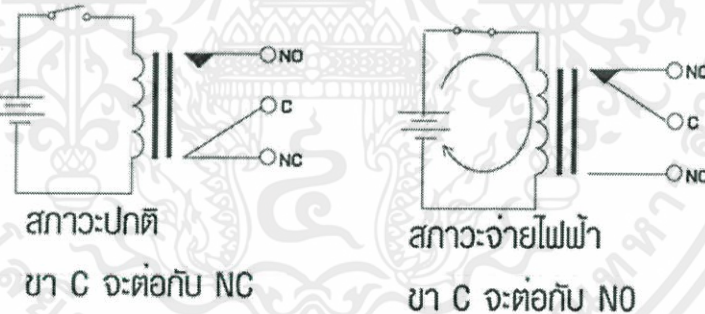
ในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (C) และ หน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO.) ขานี้จะต่อเข้ากับขาร่วม (C) เมื่อขดลวดมีแรงดันตกคร่อม หรือกระแสไหลผ่าน (ในปริมาณที่เพียงพอ) ใน รีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด ซึ่งขึ้นอยู่กับผู้ผลิต



รูปที่ 2.9 สัญลักษณ์ของรีเลย์แทนโครงสร้างรีเลย์

2.2.2 หลักการทำงานของรีเลย์

เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดจะทำให้ขดลวดเกิดสนามแม่เหล็กไปดึงแผ่นหน้าสัมผัสให้ตกลงมา แต่หน้าสัมผัสอีกอันทำให้มีกระแสไหลผ่านหน้าสัมผัสไปได้



รูปที่ 2.10 สถานะการทำงานของรีเลย์

2.3 ภาษา PHP

เป็นโปรแกรมภาษาที่สามารถใช้งานในระบบเครือข่ายอินเทอร์เน็ตได้ตัวหนึ่งที่มีความสามารถสูง มีผู้นิยมใช้จำนวนมาก สามารถใช้ได้ฟรีโดยไม่ผิดกฎหมาย นอกจากนี้ภาษา PHP ยังสามารถทำงานร่วมกับโปรแกรมฐานข้อมูล MySQL ซึ่งเป็นโปรแกรมดาวน์โหลดมาใช้ได้ฟรีเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การนำภาษา PHP ไปใช้งาน

PHP เป็นภาษาสคริปต์ (Scripting Language) คำสั่งต่างๆ จะเก็บในรูปแบบของข้อความ (Text) อาจเขียนแทรกอยู่ภายในภาษา HTML หรือใช้งานอิสระก็ได้ แต่ในการใช้งานจริงมักใช้งานร่วมกับภาษา HTML ดังนั้นการเขียนโปรแกรมนี้ต้องมีความรู้ด้านภาษา HTML เป็นอย่างดี อย่างไรก็ตามเราสามารถใช้อุปกรณ์ประยุกต์มาช่วยอำนวยความสะดวกในการสร้างงานได้ เช่น Macromedia Dreamweaver หรือโปรแกรมประเภท Editor (โปรแกรมที่อำนวยความสะดวกในการเขียนโปรแกรมภาษาต่างๆ) เช่น EditPlus Macromedia Homesite ฯลฯ โปรแกรมเหล่านี้จะช่วยจำแนกคำ เช่น คำสั่ง คำทั่วไป ตัวแปร ฯลฯ ให้มีสีต่างกันเพื่อสะดวกในการสังเกต และมีตัวเลขบอกบรรทัดทำให้สะดวกในการแก้ไข

2.3.2 การทำงานของภาษา PHP กับ Web server

ภาษา PHP นั้น ทำงานอยู่บนฝั่งเซิร์ฟเวอร์ (Server) จะไม่สามารถเห็นโค้ดของภาษา PHP ที่เครื่องคอมพิวเตอร์ของไคลเอ็นท์ได้เลย เพราะวาระหว่างที่ใช้งานเว็บเบราว์เซอร์เพื่อเรียกชมเว็บไซต์นั้น ในฝั่งของเซิร์ฟเวอร์ จะทำการประมวลผลภาษา PHP และส่งผลลัพธ์ที่ได้คือ คำสั่ง HTML หลังจากนั้นจะส่ง HTML ที่ได้จากการประมวลผลแล้วนั้นมาให้กับเว็บเบราว์เซอร์ของไคลเอ็นท์แล้วก็จะแสดงผล HTML ให้เห็นเป็นรูปร่างเว็บ และเมื่อ View Source ในฝั่งไคลเอ็นท์ จะเห็นโค้ดของ HTML เพียงอย่างเดียว แต่ไม่เห็นโค้ด PHP เลย

2.3.3 ข้อดีและข้อจำกัดของภาษา PHP

PHP เป็นภาษาที่มีการประมวลผลทางเครื่องผู้ให้บริการ (Server) โดยเป็นเครื่องมือที่ทำให้เราสามารถสร้างไฟล์เอกสารประเภท HTML ที่มีการเคลื่อนไหวเปลี่ยนแปลงได้หรือที่เรียกว่าพลวัต (Dynamic HTML) การที่เครื่องบริการมีการประมวลผลก่อนส่งข้อมูลมายังเครื่องผู้ใช้เราเรียกว่า Server Side Include (SSI) การทำงานในลักษณะนี้ทำให้ความเร็วในการทำงานสูงขึ้นอีกด้วย

โปรแกรมภาษา PHP มีข้อดีและข้อจำกัดที่พอสรุปได้ดังนี้

1. เป็นโปรแกรมฟรี (Download จากอินเทอร์เน็ตมาใช้ได้โดยไม่ผิดกฎหมาย)
2. ใช้งานร่วมกับฐานข้อมูลได้หลายชนิดเช่น MySQL ซึ่งเป็นของฟรีเช่นเดียวกัน

ประมวลผลโปรแกรมทำได้รวดเร็วกว่าการใช้โปรแกรมอื่น

2.4 ภาษา Python

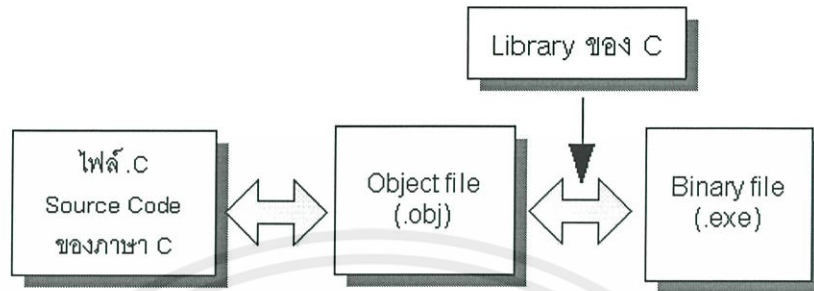
2.4.1 หลักการทำงานของภาษา Python

ภาษา Python สามารถแบ่งการทำงานออกเป็น 2 ส่วนดังนี้

2.4.1.1 คอมไพเลอร์ (Compiler)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้สอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัด เป็นตัวแปลภาษาสำหรับภาษา C, C++, Pascal การทำงานก็คือจะ ตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการคอมไพล์ ถ้าไม่มี

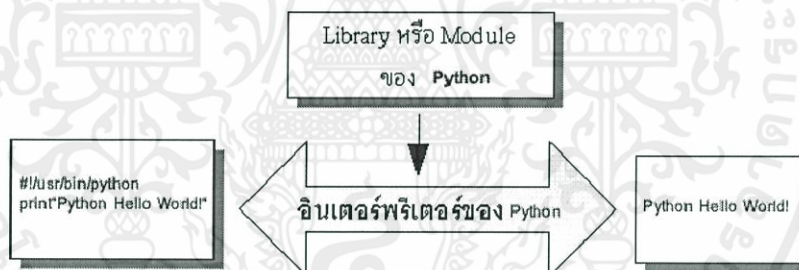
แปลไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ภาษา C ดังรูปที่ 2.11



รูปที่ 2.11 การทำงานคอมไพเลอร์ (Compiler)

(แหล่งที่มา http://python.cmsthailand.com/basic_python.html)

2.4.1.2 อินเทอร์พรีเตอร์ (Interpreter) จะทำงานเป็นบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมาเลย ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 การทำงาน อินเทอร์พรีเตอร์ (Interpreter)

(แหล่งที่มา http://python.cmsthailand.com/basic_python.html)

2.4.2 จุดเด่นของภาษาไพธอน (Python)

ภาษาไพธอน พัฒนาโดยชาวเยอรมันชื่อนาย Guido van Rossum ซึ่งได้ออกแบบมาเพื่อให้ทำงานได้กับ Web Application ที่ลักษณะคล้ายกับภาษา Perl, PHP, JAVA และ ASP เนื่องจากภาษาไพธอน เป็นภาษาที่ใหม่จึงมีคุณสมบัติที่ดีดังต่อไปนี้

- สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือ สามารถทำงานได้ทุก ๆ CPU หลาย ๆ

เอกสารนี้เป็นระบบปฏิบัติการ เพียงแต่ผู้เขียนโปรแกรมเขียนจากแพลตฟอร์มใด ๆ แล้วนำไปรันที่แพลตฟอร์มอื่นก็ทำงานต่างแพลตฟอร์มกันได้ แปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมต้นฉบับ โดยปกติแล้วโปรแกรมภาษาทั่ว ๆ ไปจะต้องจัดซื้อโปรแกรมต้นฉบับเพื่อนำมาติดตั้งในราคาที่แพงมาก แต่โปรแกรมภาษาไพธอนสามารถดาวน์โหลดจาก www.python.org ได้โดยตรง แล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภท Open Source

- ภาษาไพธอนได้นำเอาข้อดีของโปรแกรมในอดีตเข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java และ Perl เป็นต้น

- มีความปลอดภัยสูง เนื่องจากภาษาไพธอนทำงานอยู่ด้าน Server เป็นหลัก เมื่อมีการร้องขอจากเครื่อง Client จะประมวลผลที่เครื่อง Server ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่อง Server ได้โดยตรงจึงมีความปลอดภัยสูงกว่า

- ใช้ในการพัฒนา Web Service ซึ่งในปัจจุบันการพัฒนาซอฟต์แวร์ได้เน้นที่มีการแลกเปลี่ยนข้อมูลซึ่งกันและกันทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน ทำให้เกิดความสะดวกรวดสบาย ไม่ต้องใช้ซอฟต์แวร์อื่น ๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไปเรียนรู้ได้เร็วกว่าโปรแกรมภาษาอื่น ๆ เพราะมีโครงสร้างภาษาที่ไม่ซับซ้อน ซึ่งโครงสร้างภาษาค่อยคลึงกับภาษา C ถ้าโปรแกรมเมอร์ที่เคยใช้ภาษา C มาก่อนจะทำให้เรียนรู้ได้เร็วยิ่งขึ้น นอกจากนี้การเขียนโปรแกรมด้วยภาษาไพธอนจะมีความกระชับและสั้นกว่าภาษาซี

2.5 การจัดการฐานข้อมูล MYSQL

MySQL เป็น SQL (Structured Queries Language) database server ซึ่ง SQL เป็น database language ที่นิยมมากภาษาหนึ่ง เป็น database server ขนาดเล็กซึ่งเหมาะกับ applications ที่มีขนาดเล็กและปานกลางอีกทั้งยังสนับสนุน standard SQL (ANSI) MySQL ถูกเขียนขึ้นมาใช้ได้หลาย platform ของคอมพิวเตอร์ ทั้ง Unix และ Windows

2.5.1 ความรู้พื้นฐานเกี่ยวกับ user-name และ รหัสผ่าน ใน MySQL

- User-name ใน MySQL ไม่ใช่ user-name ที่ใช้ในระบบ UNIX แต่โดยปกติเมื่อมีการ connect ไปยัง MySQL server แล้ว user-name บน UNIX จะถูกใช้เป็นค่า user-name เริ่มต้น โดยผู้ใช้สามารถระบุ user-name ของ MySQL โดยใช้ option `-u username` ได้ User-name ใน MySQL มีความยาวได้สูงสุด 16 ตัวอักษร

- รหัสผ่านของ user ใน MySQL ไม่ใช่รหัสผ่านของ user ที่ใช้ในระบบ UNIX

2.5.2 การใช้งาน MySQL

MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ phpBB นิยมใช้งานร่วมกับภาษาโปรแกรม PHP ซึ่งมักจะได้ชื่อว่าเป็นคู่ จะเห็นได้จากคู่มือคอมพิวเตอร์ต่างๆ ที่จะสอนการใช้งาน MySQL และ PHP ควบคู่กันไป นอกจากนี้ หลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูล MySQL ซึ่งรวมถึง ภาษา C, C++, Pascal, ซีชาร์ป, ภาษา Java, ภาษา Perl, Python Ruby และภาษาอื่น ใช้งานผ่าน API ทุกครั้ง (Application Programming Interface) สำหรับโปรแกรมที่ติดต่อผ่าน ODBC (Open Database Connector)

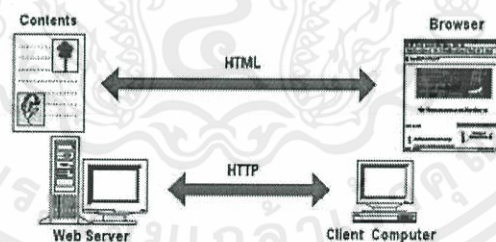
หรือ ส่วนเชื่อมต่อกับภาษาอื่น (database connector) เช่น เอเอสพี สามารถเรียกใช้ MySQL ผ่านทาง MyODBC, ADO, ADO.NET เป็นต้น

2.6 Web Server

โปรแกรมที่อยู่และทำงานบนเครื่องฝั่งเซิร์ฟเวอร์ ทำหน้าที่ในการรับคำสั่งจากการร้องขอของไคลเอ็นท์ (Client) (โดยผู้ใช้จะเรียกชมหน้าเว็บไซต์ได้โดยใช้โปรโตคอล HTTP ผ่านทางเว็บเบราว์เซอร์) และประมวลผลการทำงานจากการร้องขอดังกล่าว แล้วส่งข้อมูลกลับไปยังเครื่องของไคลเอ็นท์ สามารถเขียนได้หลายภาษา หลายรูปแบบ การเขียนภาษา PHP ระบบจะสามารถทำงานได้ก็จะต้องมี เว็บเซิร์ฟเวอร์ เป็นตัว Run Engine ของภาษานั้นๆ อีกทีหนึ่ง ดังนั้นถ้าต้องการให้คอมพิวเตอร์สามารถRunภาษา PHP ได้นั้น จะต้องจำลองเครื่องคอมพิวเตอร์ให้เป็นเซิร์ฟเวอร์ โดยใช้โปรแกรมเว็บเซิร์ฟเวอร์

2.6.1 หลักการทำงาน Web server

หลักการทำงานของเว็บเซิร์ฟเวอร์มีลักษณะเช่นเดียวกับกับบริการอื่นๆของอินเทอร์เน็ต คือ อยู่ในรูปแบบ ไคลเอ็นท์-เซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมเว็บไคลเอ็นท์ (Web Client) ทำหน้าที่เป็นผู้ร้องขอบริการ และมีโปรแกรมเว็บเซิร์ฟเวอร์ ทำหน้าที่เป็นผู้ให้บริการ โปรแกรมเว็บไคลเอ็นท์ก็คือโปรแกรมเว็บเบราว์เซอร์ (Web Browser) ในเครื่องของผู้ใช้นั่นเอง ซึ่งสำหรับโปรแกรมเว็บเซิร์ฟเวอร์นั้นจะถูกติดตั้งไว้ในเครื่องของผู้ให้บริการเว็บไซต์ (มักเรียกเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์เพื่อให้บริการ WWW ว่า "เว็บเซิร์ฟเวอร์" เช่นเดียวกัน) การติดต่อระหว่างโปรแกรมเว็บเบราว์เซอร์กับโปรแกรมเว็บเซิร์ฟเวอร์จะกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocol) ดังภาพที่ 2.13



รูปที่ 2.13 การทำงานของ Web Server

(แหล่งที่มา http://www.kmitl.ac.th/~kpteeraw/data_com/datacom_52/WWW/Index.html)

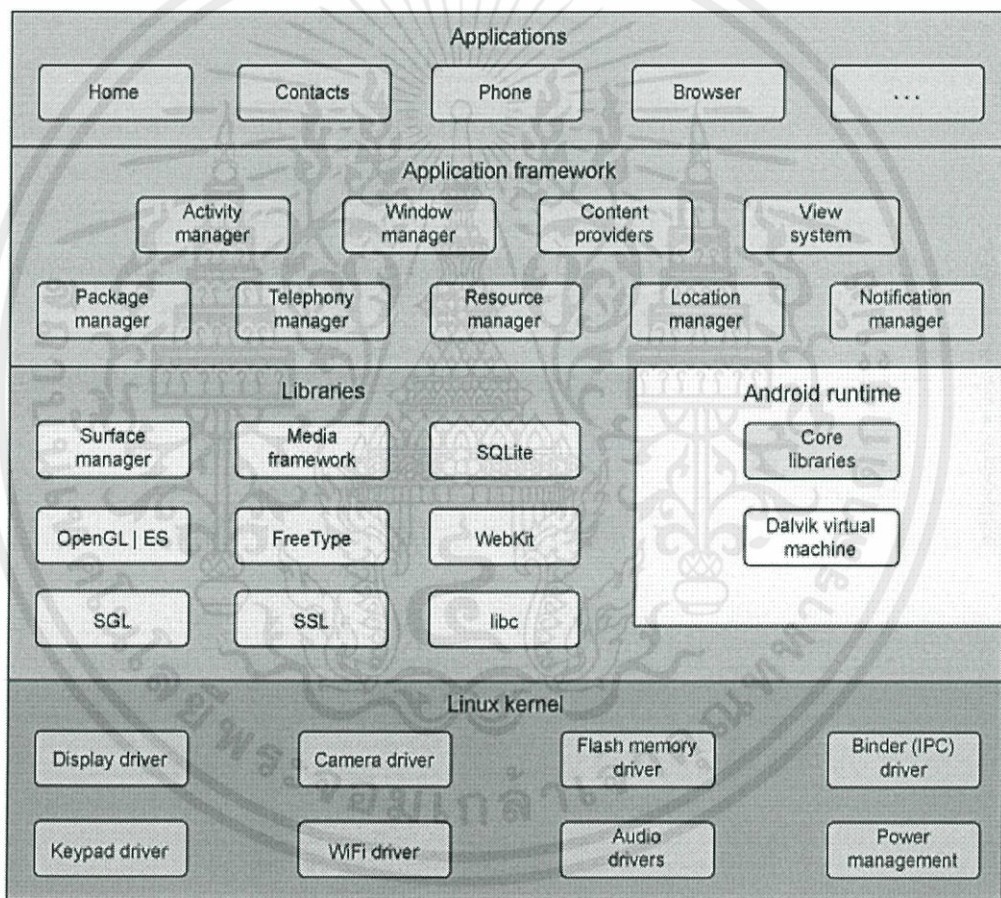
2.7 ระบบปฏิบัติการแอนดรอยด์ (Android)

แอนดรอยด์ (Android) กูเกิลแอนดรอยด์ (Google Android) หรือ ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) เป็นชื่อเรียกชุดซอฟต์แวร์ หรือแพลตฟอร์ม (Platform) สำหรับอุปกรณ์อิเล็กทรอนิกส์ ที่มีหน่วยประมวลผลเป็นส่วนประกอบ อาทิเช่น คอมพิวเตอร์, โทรศัพท์ (Telephone), โทรศัพท์เคลื่อนที่ (Cell phone), อุปกรณ์เล่นอินเทอร์เน็ตขนาดพกพา

(MID) เป็นต้น แอนดรอยด์นั้น ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัท กูเกิล จุดประสงค์ของแอนดรอยด์นั้น มีจุดเริ่มต้นมาจากบริษัท Android Inc. ที่ได้นำเอา ระบบปฏิบัติการลินุกซ์ (Linux) ซึ่งนิยมนำไปใช้งานกับเครื่องแม่ข่าย (Server) เป็นหลัก นำมาลดทอนขนาดตัว (แต่ไม่ลดทอนความสามารถ) เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพา ที่มีขนาดพื้นที่จัดเก็บข้อมูลที่จำกัด โดยหวังว่า แอนดรอยด์ นั้นจะเป็นหุ่นยนต์ตัวน้อย ๆ ที่คอยช่วยเหลืออำนวยความสะดวกแก่ผู้ที่พกพามัน ไปในทุกที่ ทุกเวลา

2.7.1 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)

สถาปัตยกรรมของระบบปฏิบัติการแอนดรอยด์นั้นถูกแบ่งออกเป็นลำดับชั้น (Layer) ซึ่งโครงสร้างโดยรวมของระบบ สามารถแสดงได้ดังรูปที่ 2.14



รูปที่ 2.14 สถาปัตยกรรมแอนดรอยด์

(แหล่งที่มา <http://kadroidz.blogspot.com/2012/03/android-architecture.html>)

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ จะสังเกตได้ว่า มีการแบ่งออกมาเป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งก็คือส่วนของ Applications จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel สามารถอธิบายเป็นส่วนๆ ได้ดังนี้

2.7.1.1 Application เป็นส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้ เช่น แอปพลิเคชันรับ/ส่งอีเมล, SMS, ปฏิทิน, แผนที่, เว็บเบราว์เซอร์, รายชื่อผู้ติดต่อ เป็นต้น ซึ่งแอปพลิเคชันจะอยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app

2.7.1.2 Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้ผู้พัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น ซึ่งสามารถพัฒนาโปรแกรมโดยใช้ภาษา Java ผ่านทาง API (Application Programming Interface) โดยไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน สามารถอธิบายส่วนประกอบของ Application Framework ได้ดังตารางที่ 2.1

ตารางที่ 2.1 การอธิบายส่วนประกอบของ Application Framework

ส่วนประกอบ	หน้าที่
Activity Manager	ควบคุม Lifecycle ของ Activity
Window Manager	ควบคุมหน้าต่างการทำงาน
Content Providers	การเข้าถึงข้อมูลแอปพลิเคชันอื่นๆ
View System	ชุดเครื่องมือของ User Interface
Package Manager	จัดการเกี่ยวกับชุดคำสั่งต่างๆ
Telephony Manager	เกี่ยวกับการจัดการข้อมูลทางเครือข่าย
Resource Manager	เกี่ยวกับข้อมูลที่ไม่ใช่โค้ด เช่น รูปภาพ เสียง วิดีโอ
Location Manager	จัดการเกี่ยวกับข้อมูลทางด้าน GPS
Notification Manager	จัดการเกี่ยวกับการแจ้งเตือนต่างๆ

2.7.1.3 Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน แสดงได้ดังตารางที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การอธิบายส่วนประกอบของ Libraries

ส่วนประกอบ	หน้าที่
Surface Manager	เป็นตัวจัดการระบบแสดงผลและควบคุมบนจอภาพ
Media Framework	การใช้งานไฟล์ฟอร์แมตมัลติมีเดีย
SQLite	ระบบฐานข้อมูลที่มีขนาดเล็ก เร็ว และเสถียร
OpenGL / ES	ชุดคำสั่งประมวลผลภาพสามมิติ
FreeType	สำหรับการประมวลผลตัวอักษรทั้งแบบ Bitmap และ Vector
WebKit	เป็นส่วนของการทำงานของ Web Browser
SGL	สำหรับพัฒนากาฟิกสองมิติ
SSL	เกี่ยวกับความปลอดภัยในการส่งข้อมูลระหว่าง Client กับ Server
Libc	ไลบรารีมาตรฐานของภาษาซีสำหรับอุปกรณ์ที่รันบน Linux

นอกจากนี้ ใน Libraries จะมีการแบ่งไลบรารีย่อยๆ ออกเป็น 2 ไลบรารี ซึ่งจะเรียกว่า Android Runtime จะประกอบไปด้วย Dalvik Virtual Machine และ Core Libraries

2.7.1.4 Linux Kernel เป็นส่วนที่ทำหน้าที่ในการจัดการกับบริการหลักของระบบปฏิบัติการ โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์โดยตรง เช่น หน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6 Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.8 ภาษา JAVA

ภาษา Java คือ ภาษาคอมพิวเตอร์ที่ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ (Sun Microsystems) เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้น คลาส คือ ที่เก็บกระบวนการ (Method) หรือ พฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม

ซึ่งจุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object Oriented Programming มาพัฒนาโปรแกรมของตนได้ ภาษา Java เริ่มเป็นที่นิยมแพร่หลายในปี ค.ศ. 1996 เป็นภาษาที่ไม่ขึ้นกับแพลตฟอร์ม (Platform Independent) เวอร์ชันแรกคือ JDK 1.0

2.8.1 คุณลักษณะเด่นของภาษา JAVA

1) โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกันโดยไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่องหนึ่ง โปรแกรมนั้น ก็สามารถถูกคอมไพล์และรันบนเครื่องพีซีธรรมดาได้

2) เมื่อเปรียบเทียบโค้ดของโปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวนโค้ดน้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ถึง 4 เท่า และใช้เวลาในการเขียนโปรแกรมน้อยกว่าประมาณ 2 เท่า

3) ภาษาJavaมีการรักษาความปลอดภัยทั้งระดับต่ำและระดับสูงได้แก่ Electronic signature, Public and Private key management, Access control และ Certification ของภาษา Java

2.8.2 รูปแบบของภาษา JAVA

ภาษา Java เป็นภาษาที่ไม่กำหนดแบบการเขียนโปรแกรมในแต่ละบรรทัด แต่บรรทัดสามารถเขียนคำสั่งได้หลายคำสั่ง สามารถแทรกคำอธิบาย (Comment) เป็นภาษาที่บังคับอักขระตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก มีตัวดำเนินการ (Operators) หลายชนิดให้ได้ใช้งาน นอกจากคำสั่งนั้นเป็นคำสั่งที่ผู้ใช้สร้างขึ้นใหม่ อาจกำหนดเป็นตัวพิมพ์ใหญ่ หรือตัวพิมพ์เล็กก็ได้ และสามารถเขียนชุดคำสั่งที่ประกอบด้วยตัวดำเนินการหลายตัวที่ต่างชนิดกัน ในชุดคำสั่งหนึ่งๆ ได้ โดยภาษา Java จะจัดลำดับการประมวลผลตามลำดับการทำงานของตัวดำเนินการ

2.8.3 ซอร์สโค้ด (Source Code)

ในภาษา Java จะมีซอร์สโค้ดเป็นไฟล์ที่มีนามสกุล *.java เมื่อผ่านการคอมไพล์ (Compile) แล้วจะมีไฟล์เพิ่มมาเป็นไฟล์ที่มีนามสกุลเป็น *.class

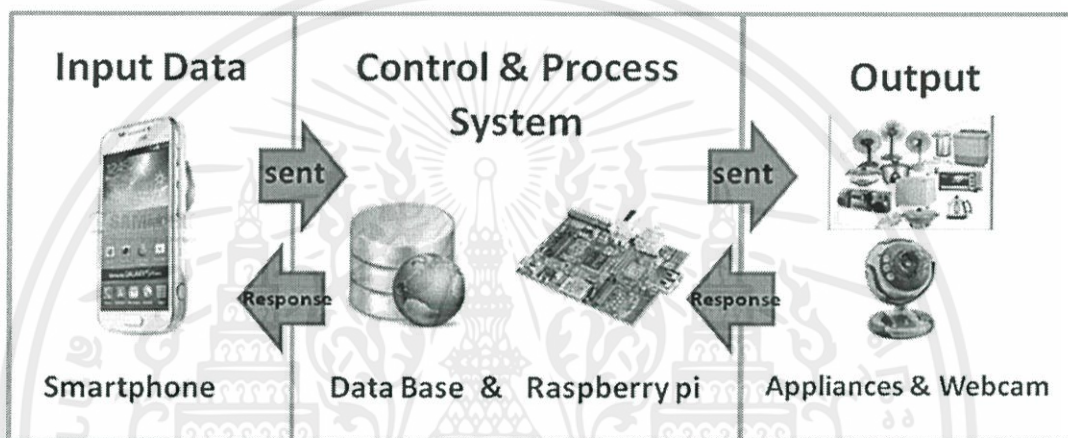
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปฏิญญาพันธ

3.1 การออกแบบ

ในบทนี้จะเป็นการอธิบายการออกแบบการทำงานของระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone (Android) ซึ่งการออกแบบสามารถเขียนเป็นบล็อกไดอะแกรมโดยรวมของระบบการทำงานได้รูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมโดยรวมของระบบการทำงาน

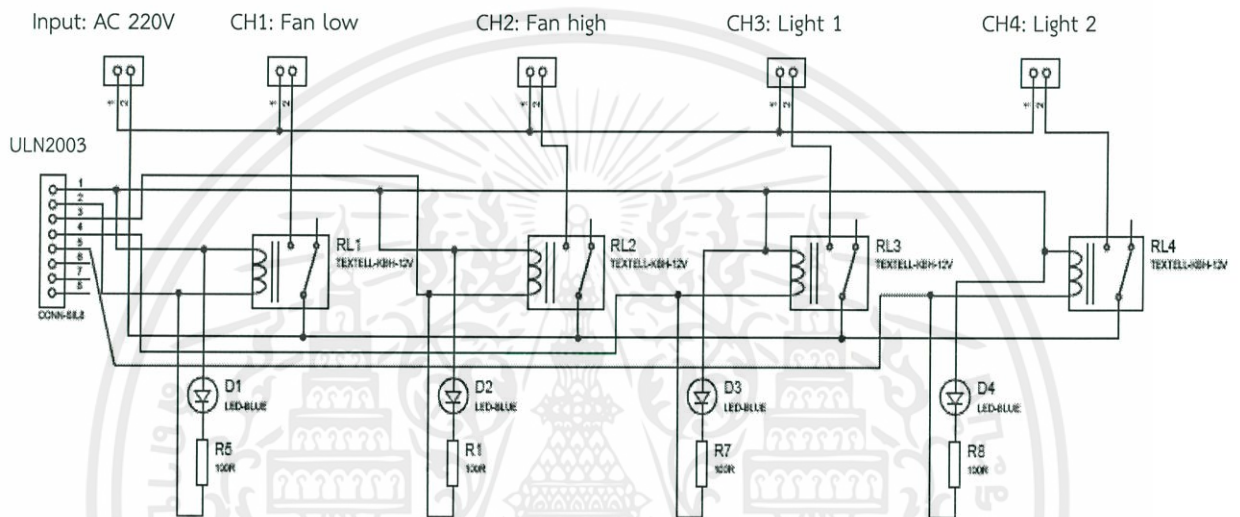
จากบล็อกไดอะแกรมโดยรวมของระบบการทำงานดังรูปที่ 3.1 ซึ่งการทำงานจะเริ่มจากฝั่ง Input Data คือป้อนข้อมูลเข้าทางแอปพลิเคชันแอนดรอยด์บนสมาร์ตโฟน (Client) จากนั้นข้อมูลจะถูกส่งต่อไปยังส่วน Control & Process System คือ ระบบฐานข้อมูลบนระบบอินเทอร์เน็ตซึ่งข้อมูลที่ได้รับมาจากส่วน Input Data จะถูกเก็บค่าไว้ในระบบฐานข้อมูล เพื่อให้ระบบสมองกลฝังตัว (Server) ตรวจสอบค่าที่ได้รับมาบนระบบฐานข้อมูลเพื่อนำไปควบคุม GPIO ของระบบสมองกลฝังตัวเพื่อควบคุมส่วนของ Output คือ ชุดรีเลย์สวิตซ์ที่ทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้าได้ตามต้องการ อีกทั้งยังมีส่วนแสดงสถานะการทำงานของเครื่องใช้ไฟฟ้า ทั้งแบบแสดงภาพต่อเนื่องจากกล้องและแบบแสดงค่าการทำงานด้วยชุดวงจรตรวจสอบแรงดัน 220V. Isolated กลับมาผ่านทาง Smart Phone ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การออกแบบทางฮาร์ดแวร์

การออกแบบทางฮาร์ดแวร์ของระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone (Android) จะประกอบด้วย 2 ส่วนที่สำคัญคือ

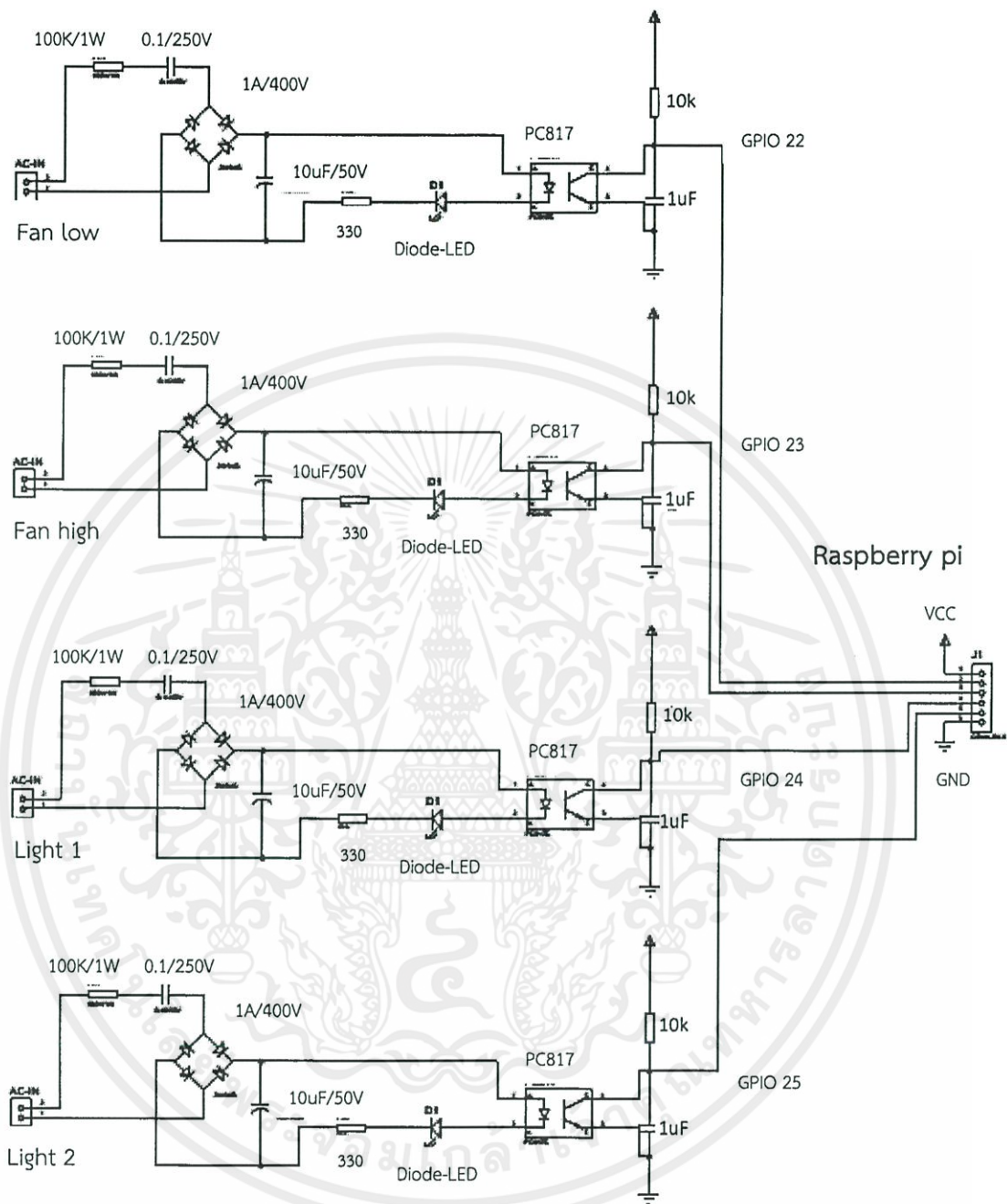
- ส่วนของชุดวงจรรีเลย์สวิตช์ซึ่งสามารถออกแบบวงจรผ่านทางโปรแกรมโปรโตคอลดังรูปที่ 3.2 โดยสัญญาณขาเข้าจะผ่านทางไอซีทรานซิสเตอร์ ULN2003 ซึ่งเป็นส่วนที่ขับแรงดันให้รีเลย์สวิตช์ทำงานและมีส่วนตรวจสอบสถานะด้วยหลอด LED



รูปที่ 3.2 การออกแบบทางฮาร์ดแวร์ส่วนของชุดวงจรรีเลย์สวิตช์

- ส่วนของวงจรตรวจจับแรงดัน 220V Isolated เป็นชุด ตรวจจับแรงดัน 220V Isolated ขนาด 4 ช่อง โดยทำหน้าที่แปลงสัญญาณ Input ชนิดแรงดันไฟสลับแบบ AC Voltage ขนาด 220VAC ให้เป็นสัญญาณ โลจิก “0” หรือ “1” แบบสัญญาณดิจิทัล เพื่อส่งให้กับภาค Input ของไมโครคอนโทรลเลอร์ โดยในการประยุกต์ใช้งานนั้น จะใช้สำหรับตรวจจับการทำงานของอุปกรณ์ Sensor หรือหน้าสัมผัสสวิตช์ ต่างๆที่ให้ผลการทำงานเป็นแรงดันไฟสลับแบบ AC Voltage ขนาด 220VAC โดยการทำงานของวงจรจะให้ผลลัพธ์เป็นโลจิกสัญญาณดิจิทัล โดยถ้าทางด้าน Input ณ จุด Terminal AC-IN ได้รับระดับแรงดันไฟสลับ 220VAC จะให้สถานะทาง Output เป็นโลจิก “0” แต่ถ้า Input ได้รับระดับแรงดันเป็น 0VAC จะให้ สถานะทาง Output เป็นโลจิก “1”

ซึ่งในการทดลอง จะใช้ส่วนของวงจรตรวจจับแรงดัน 220V Isolated เป็นชุด ตรวจจับแรงดัน 220V Isolated ขนาด 4 ช่อง เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า โดยจะมี input เข้ามาเป็นแรงดัน AC 220V. จากนั้นจะได้ output ออกมาเป็นแรงดันสัญญาณดิจิทัล (โลจิก “0” หรือ “1”) เข้าสู่ระบบสมองกลฝังตัวเพื่อตรวจสอบสถานะของอุปกรณ์ไฟฟ้าทั้ง 4 ช่องสัญญาณ โดยจะมีการออกแบบทางฮาร์ดแวร์ดังรูปที่ 3.3



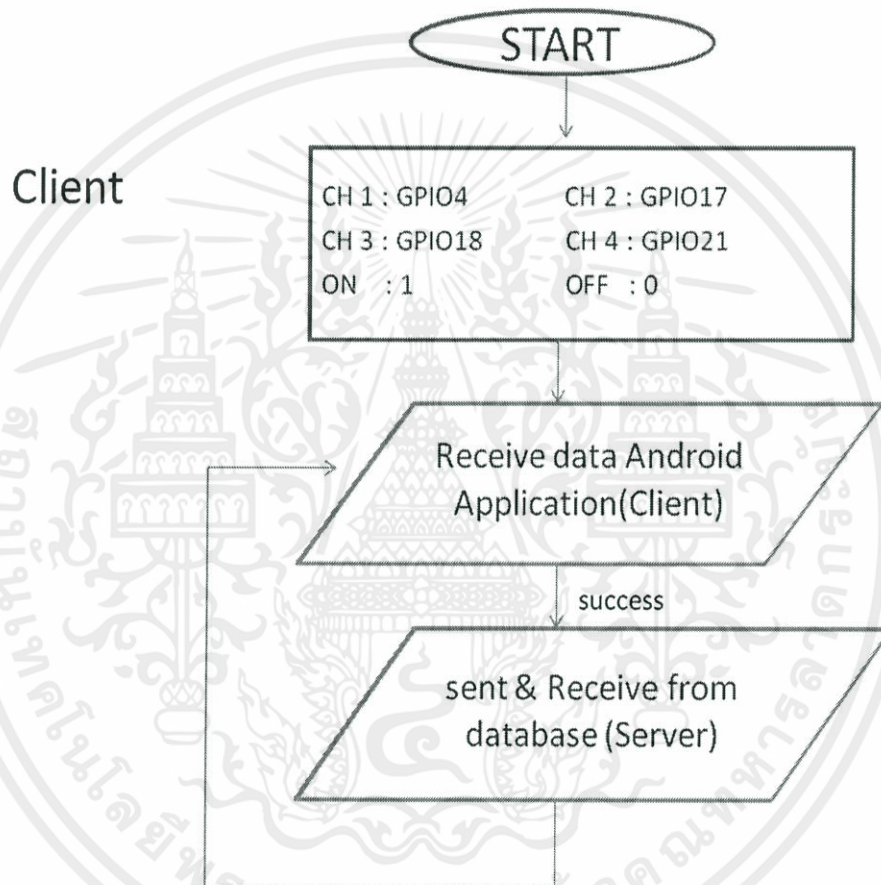
รูปที่ 3.3 การออกแบบทางฮาร์ดแวร์ของวงจรตรวจจับแรงดัน 220V Isolated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบทางซอฟต์แวร์

ในการออกแบบทางซอฟต์แวร์ของระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone (Android) สามารถแบ่งได้ออกเป็น 2 ส่วนดังนี้

3.1.2.1 การออกแบบทางซอฟต์แวร์ของฝั่ง Client โดยหลักๆจะเป็นการออกแบบทางซอฟต์แวร์เพื่อสร้างและพัฒนาแอปพลิเคชันแอนดรอยด์ ให้สามารถส่งข้อมูลไปยังระบบฐานข้อมูลบนระบบอินเทอร์เน็ตได้ ซึ่งสามารถออกแบบทางซอฟต์แวร์ของฝั่ง Client ได้ดังรูปที่ 3.4

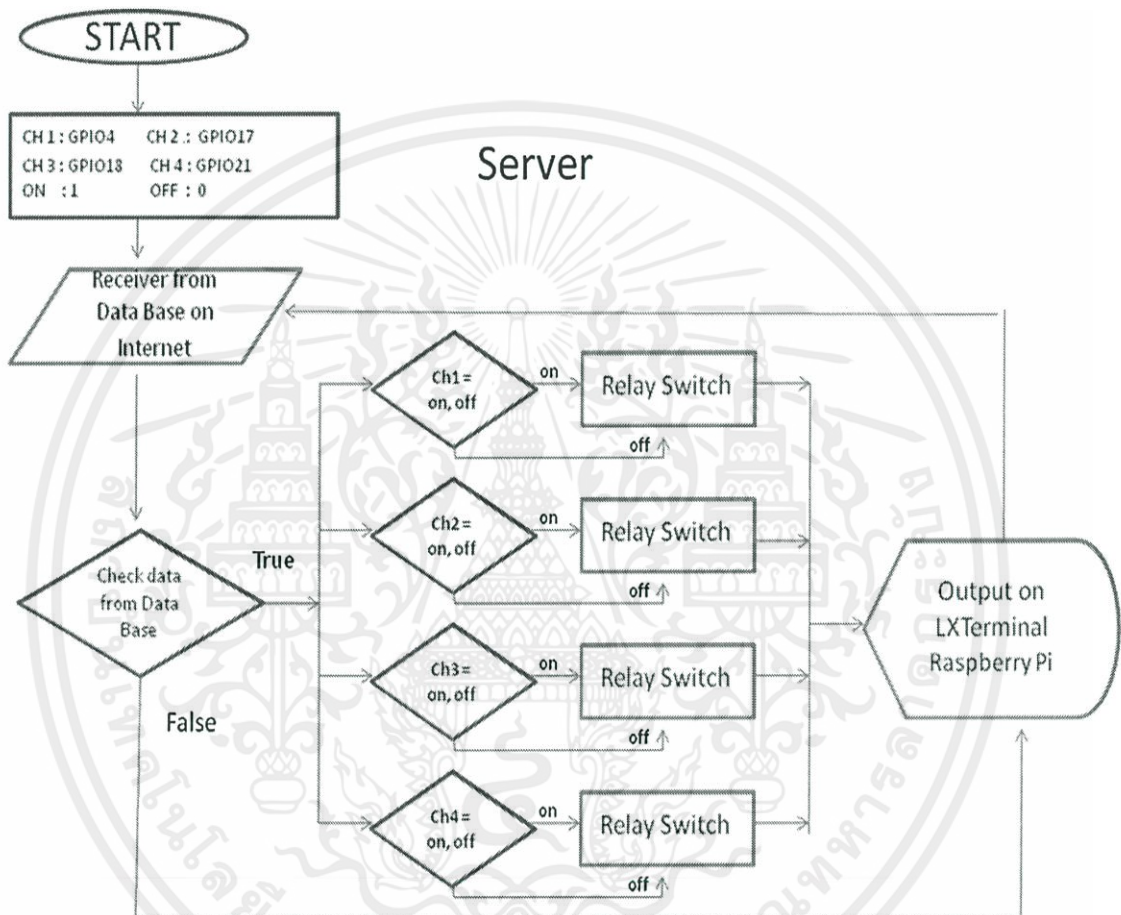


รูปที่ 3.4 การออกแบบทางซอฟต์แวร์ของฝั่ง Client

จากการออกแบบทางซอฟต์แวร์ของฝั่ง Client ดังรูปที่ 3.4 เริ่มต้นจากการกำหนดตัวแปร จากนั้นทำการป้อนข้อมูลเข้าทางแอปพลิเคชันแอนดรอยด์โดยข้อมูลจะถูกเก็บไว้ในระบบฐานข้อมูลบนระบบอินเทอร์เน็ตแล้วจึงกลับมารับค่าข้อมูลทางขาเข้าจากระบบแอปพลิเคชันแอนดรอยด์อีกครั้งสำหรับรับข้อมูลถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การออกแบบทางซอฟต์แวร์ของฝั่ง Server จะเป็นการออกแบบทางซอฟต์แวร์บนระบบสมองกลฝังตัวเพื่อสามารถตรวจสอบข้อมูลที่ได้รับมาจากระบบฐานข้อมูลสำหรับนำไปควบคุมชุดวงจรรีเลย์สวิตช์ผ่านทาง GPIO ได้ ซึ่งสามารถออกแบบทางซอฟต์แวร์ของฝั่ง Server ได้ดังรูปที่ 3.5



รูปที่ 3.5 การออกแบบทางซอฟต์แวร์ของฝั่ง Server

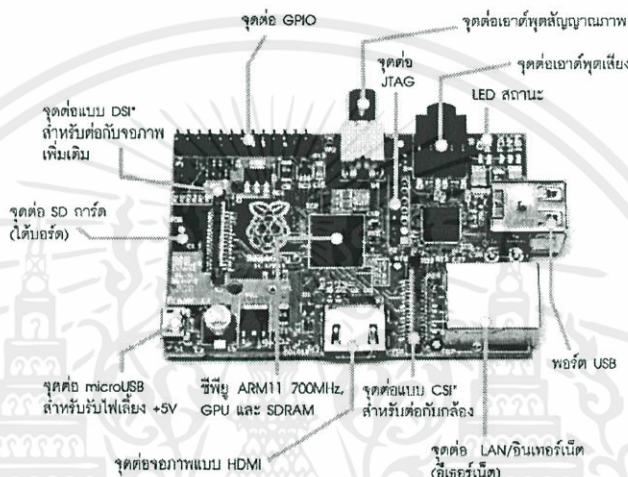
จากการออกแบบทางซอฟต์แวร์ทางฝั่ง Server ดังรูปที่ 3.5 เริ่มต้นจากการกำหนดค่าตัวแปรให้เหมือนกับทางฝั่ง Client จากนั้นจะรับข้อมูลจากระบบฐานข้อมูลบนระบบอินเทอร์เน็ตแล้วตรวจสอบข้อมูลที่ได้รับมาบนระบบฐานข้อมูลเพื่อใช้ในการควบคุมการทำงานของรีเลย์สวิตช์ทั้ง 4 ช่องทางแล้วจึงแสดงผลทาง LXTERMINAL บนระบบสมองกลฝังตัวแล้วทำการวนกลับไปตรวจสอบข้อมูลในระบบฐานข้อมูลสำหรับข้อมูลถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Raspberry pi (บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก)

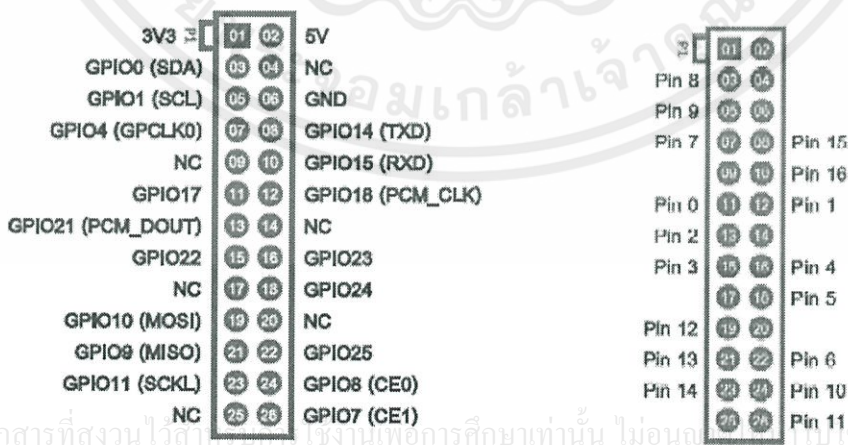
บอร์ดไมโครคอมพิวเตอร์แบบแผ่นเดี่ยวที่บรรจุความสามารถไว้เพียงรองรับระบบปฏิบัติการ Linux บรรจุลงใน SDการ์ดสำหรับการพัฒนาไปสู่บอร์ด Embedded Linux พร้อมจุดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตทั้งผ่านพอร์ต USB, LAN, HDMI, ช่องสัญญาณภาพและ GPIO สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์



รูปที่ 3.6 แสดงส่วนประกอบของบอร์ด Raspberry pi

3.2.2 GPIO

บอร์ด Raspberry Pi มีขา Input , Output ดิจิตอลหรือ GPIO สำหรับติดต่อกับอุปกรณ์ภายนอก รวม 17 ขาโดยมีการจัดเรียงขามาตรฐานตามรูปที่ 3.7 และหากใช้โปรแกรม WiringPi จะมีการจัดเรียงขาอินพุต เอาต์พุตใหม่ตามรูปที่ 3.8



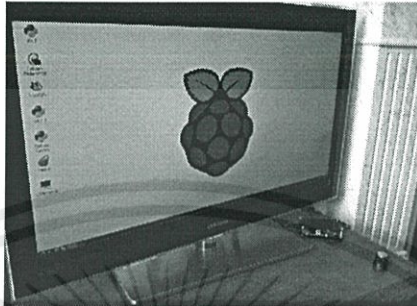
รูปที่ 3.7 การจัดเรียงขา GPIO แบบมาตรฐาน

รูปที่ 3.8 การจัดเรียงขา GPIO เมื่อใช้งานกับโปรแกรม WiringPi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น

3.2.3 จอแสดงผล

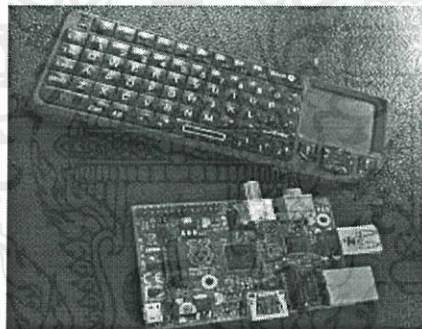
ใช้ในการแสดงผลการทดลองที่ใช้ในการทำงานของ Raspberry pi



รูปที่ 3.9 จอแสดงผล

3.2.4 KEY-BOARD & MOUSE

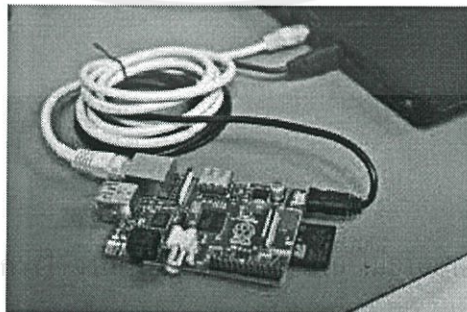
เป็นอุปกรณ์ที่ใช้ในการทำงานร่วมกับ Raspberry เพื่อใช้ในกระบวนการทำงานของ Raspberry Pi



รูปที่ 3.10 KEYBOARD & MOUSE

3.2.5 สาย LAN

เป็นอุปกรณ์ที่ใช้ในการเชื่อมต่อ Raspberry กับ Web server



รูปที่ 3.11 สาย LAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ...ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 กล้อง Web Cam

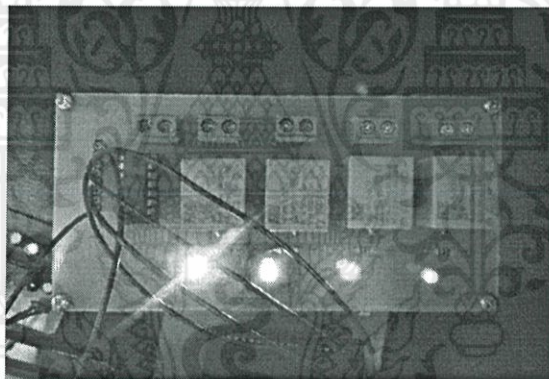
เป็นอุปกรณ์ที่ใช้ในแสดงภาพบนแอปพลิเคชันแอนดรอยด์จาก Raspberry pi อีกทั้งยังสามารถตรวจสอบสถานะการทำงานของเครื่องใช้ไฟฟ้าได้



รูปที่ 3.12 กล้อง Web Cam

3.2.7 ชุดวงจร Relay Switch

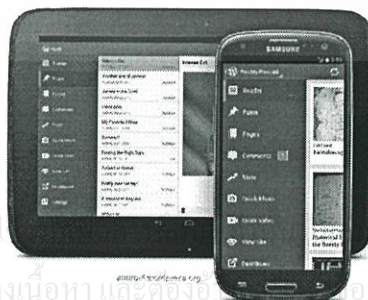
เป็นอุปกรณ์ที่ใช้ในการตรวจสอบค่าที่ได้จาก GPIO ของ Raspberry และเป็น Switch ไปยังเครื่องใช้อุปกรณ์ไฟฟ้า



รูปที่ 3.13 ชุด Relay Switch

3.2.8 Smart Phone

ใช้ในการแสดงผลและควบคุมการทำงานของอุปกรณ์ไฟฟ้าผ่านระบบอินเทอร์เน็ต

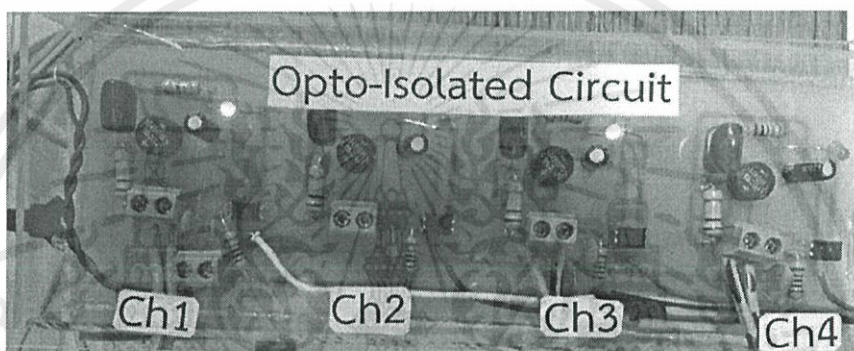


รูปที่ 3.14 Smart Phones

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่ควรเผยแพร่หรือใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงเอกสารทุกครั้งที่มีการนำไปใช้

3.2.9 ชุดวงจรตรวจจับแรงดัน 220V Isolated

วงจรถวายจับแรงดัน 220V Isolated ขนาด 4 ช่อง โดยทำหน้าที่แปลงสัญญาณ Input ชนิดแรงดันไฟสลับแบบ AC Voltage ขนาด 220VAC ให้เป็นสัญญาณ โลกิก “0” หรือ “1” แบบสัญญาณดิจิทัล เพื่อส่งให้กับภาค Input ของไมโครคอนโทรลเลอร์ โดยในการประยุกต์ใช้งาน นั้น จะใช้สำหรับตรวจจับการทำงานของอุปกรณ์เซนเซอร์ หรือหน้าสัมผัสสวิตช์ต่างๆ ที่ให้ผลการ ทำงานเป็นแรงดันไฟสลับแบบ AC Voltage ขนาด 220VAC โดยการทำงานของวงจรถวายจะให้ผลลัพธ์ เป็นโลกิกสัญญาณดิจิทัล โดยถ้าทางด้าน Input ณ.จุดเทอร์มินอลที่ AC-IN ที่ได้รับระดับแรงดันไฟ สลับ 220VAC จะให้สถานะทาง Output เป็นโลกิก “0” แต่ถ้า Input ที่ได้รับระดับแรงดันเป็น 0V AC จะให้ สถานะทาง Output เป็นโลกิก “1”



รูปที่ 3.15 ชุดวงจรถวายจับแรงดัน 220V Isolated

3.3 การจัดเก็บผลการทดลอง

3.3.1 ทำการตรวจสอบ IP Address ทั้งฝั่ง Client (แอปพลิเคชันแอนดรอยด์) และ ฝั่ง Server (ระบบสมองกลฝังตัว) ด้วยคำสั่ง IP config บน Window cmd

3.3.2 ทำการ ping เข้าไปยังเครื่อง Server เพื่อตรวจสอบสถานะการเชื่อมต่อบน โครข่ายด้วย IP Address ของฝั่ง Server (ระบบสมองกลฝังตัว)

3.3.3 ทำการตรวจสอบกระบวนการสื่อสารระหว่างเครื่อง Client (แอปพลิเคชันแอนดรอยด์) และ เครื่อง Server (ระบบสมองกลฝังตัว) ที่มีการรับ-ส่งข้อมูลกัน ด้วยโปรแกรม Wire Shark

3.3.4 เก็บผลการทดลองที่ได้จากหน้าเว็บแอปพลิเคชันแอนดรอยด์ก่อนป้อนข้อมูลเข้าไปในแอปพลิเคชันดังกล่าว

3.3.5 เก็บผลการทดลองที่ได้จากหน้าเว็บแอปพลิเคชันแอนดรอยด์ต่างๆหลังป้อนข้อมูลเข้าไปในแอปพลิเคชันดังกล่าว

3.3.6 เก็บผลการทดลองที่ได้จากระบบฐานข้อมูลบนระบบอินเทอร์เน็ต หลังป้อนข้อมูลเข้าไปในแอปพลิเคชันแอนดรอยด์

3.3.7 เก็บผลการทดลองที่ได้จากหน้าต่าง LXTerminal บนระบบสมองกลฝังตัว

3.3.8 แสดงผลการทดลองที่ได้จาก GPIO ของระบบสมองกลฝังตัวจากสถานะของหลอด

LED

3.3.9 จัดเก็บผลการทดลองที่ได้จากการตรวจสอบสถานะของอุปกรณ์ไฟฟ้าผ่านชุดวงจร Opto-Isolated Relay เข้าระบบสมองกลฝังตัว

3.3.10 วัดสัญญาณดิจิทัลที่ได้หลังจากผ่านวงจรตรวจจับแรงดัน 220V Isolated ก่อนเข้า GPIO ของ Raspberry Pi



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

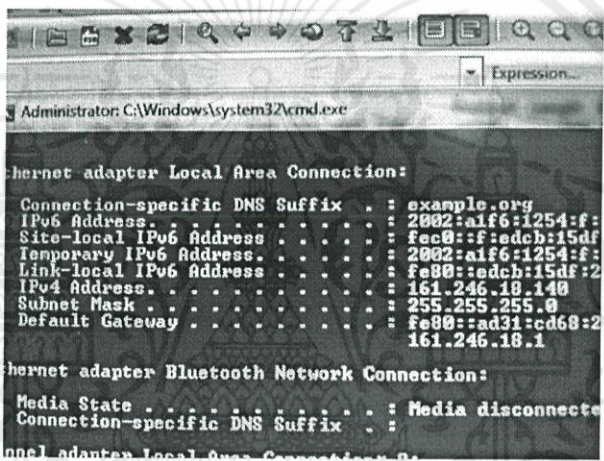
บทที่ 4

ผลการทดลอง

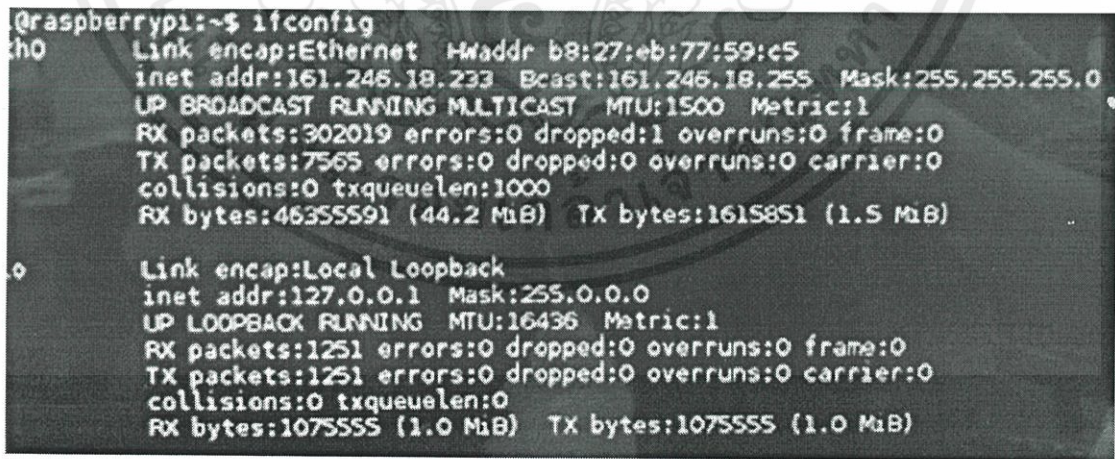
จากการออกแบบในบทที่ 3 เป็นการออกแบบโดยแบ่งการทำงานเป็น 2 ส่วน คือ ส่วนของ Client และส่วนของ Server โดยการเก็บผลการทดลองจะเริ่มต้นดังขั้นตอนต่อไปนี้

4.1 ตรวจสอบ IP Address ทั้งทางฝั่ง Client และ Server

ตรวจสอบ IP Address ทั้งทางฝั่ง client (Android Application) ด้วยคำสั่ง IP configบน Window cmd. แสดงดังรูปที่ 4.1 และฝั่ง Server (ระบบสมองกลฝังตัว) ด้วยคำสั่ง ifconfigบนหน้าจอ Terminal แสดงดังรูปที่ 4.2



รูปที่ 4.1 IP Address ของฝั่ง Client (Android Application)



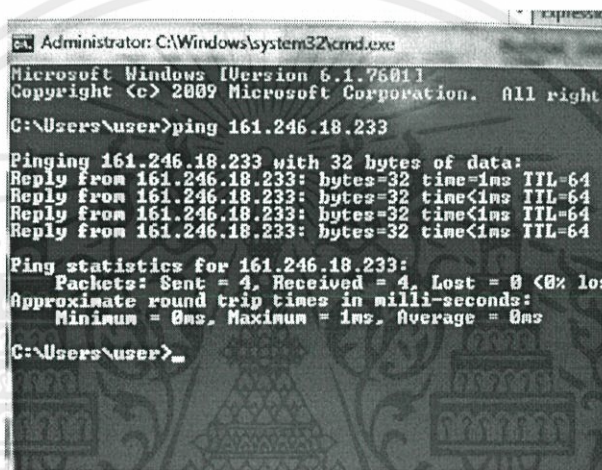
รูปที่ 4.2 IP Address ของฝั่ง Server (สมองกลฝังตัว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4.1 จะเห็นได้ว่า IP Address ของฝั่ง Client (Application Android) คือ 161.246.18.140 และในรูปที่ 4.2 จะเห็นได้ว่า IP Address ของฝั่ง Server (ระบบสมองกลฝังตัว) คือ 161.246.18.233

4.2 ตรวจสอบการเชื่อมต่อกันระหว่าง Server และ Client

ผู้ทำการทดลองทำการตรวจสอบการเชื่อมต่อกันระหว่าง Server (Android Application) และ Client (ระบบสมองกลฝังตัว) ด้วยคำสั่ง ping ตามด้วย IP Address ของฝั่ง Server (ระบบสมองกลฝังตัว) บน cmd. Window จะได้ดังรูปที่ 4.3



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 161.246.18.233

Pinging 161.246.18.233 with 32 bytes of data:
Reply from 161.246.18.233: bytes=32 time=1ms TTL=64
Reply from 161.246.18.233: bytes=32 time<1ms TTL=64
Reply from 161.246.18.233: bytes=32 time<1ms TTL=64
Reply from 161.246.18.233: bytes=32 time<1ms TTL=64

Ping statistics for 161.246.18.233:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\user>
```

รูปที่ 4.3 ผลลัพธ์ที่ได้จากคำสั่ง ping

จากรูปที่ 4.3 จะเห็นได้ว่า ในฝั่งของ Client (ระบบสมองกลฝังตัว) และฝั่งของ Server (Android Application) สามารถเชื่อมต่อถึงกัน พร้อมสำหรับการ รับ ส่ง ข้อมูล

4.3 ตรวจสอบ Protocol สื่อสารระหว่างเครื่อง Client และ เครื่อง Server

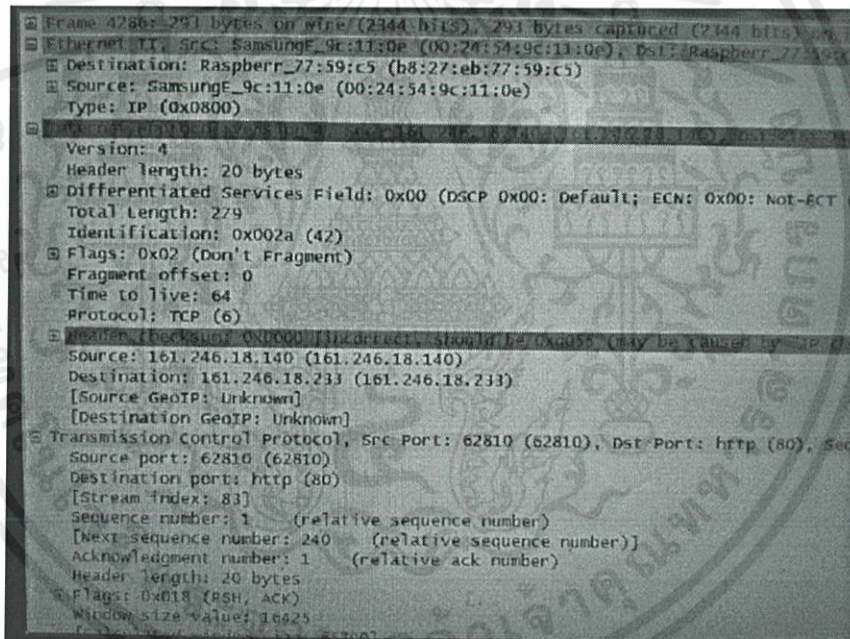
จากนั้นทำการตรวจสอบกระบวนการสื่อสารระหว่างเครื่อง Client (แอปพลิเคชัน แอนดรอยด์) และ เครื่อง Server (ระบบสมองกลฝังตัว) ที่มีการรับ-ส่งข้อมูลถึงกัน ด้วยโปรแกรม Wire Shark โดยการทดลองส่งข้อมูลจากเครื่อง Client ไปยังเครื่อง Server แล้ว capture ไว้ จากนั้น หา Source 161.246.18.140 (Client) และ Destination 161.246.18.233 (Server) ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	Time	Source	Destination	Protocol
4042	20.6123090	161.246.18.206	239.255.255.250	SS
4050	20.7458860	161.246.18.130	239.255.255.250	SS
4125	21.5637090	161.246.18.144	239.255.255.250	SS
4138	21.7487300	161.246.18.31	239.255.255.250	SS
4139	21.7621200	161.246.18.144	239.255.255.250	SS
4160	21.8405420	161.246.18.144	239.255.255.250	SS
4163	21.9033070	161.246.18.31	239.255.255.250	SS
4258	22.3984290	161.246.18.19	239.255.255.250	SS
4290	22.5139250	161.246.18.233	161.246.18.140	HT
4300	22.5884680	161.246.18.31	239.255.255.250	SS
4334	22.7944400	fe80::c456:e910:a99ff02::c		SS
4354	22.9271980	161.246.18.130	239.255.255.250	SS
4376	23.0154560	161.246.18.206	239.255.255.250	SS
4408	23.1835740	fe80::1106:6d8b:888ff02::c		SS
4409	23.1838930	161.246.18.133	239.255.255.250	SS
4410	23.2095790	fe80::1106:6d8b:888ff02::c		SS
4411	23.2095820	161.246.18.133	239.255.255.250	SS
4439	23.2764830	fe80::1106:6d8b:888ff02::c		SS

รูปที่ 4.4 ผลการตรวจจับการรับ-ส่งข้อมูลด้วยโปรแกรม wireshark

ในหน้าต่างโปรแกรม wireshark จะแสดงถึงกระบวนการของการสื่อสารข้อมูลใน network และสามารถแสดงข้อมูล package ต่างๆ ที่ใช้ในการรับ - ส่งข้อมูล ดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างโปรแกรม wiresharkตรวจจับข้อมูล package ต่างๆ

4.4 config ค่าต่างๆ ของกล้อง webcam

ในขั้นตอนต่อมา ผู้ทำการทดลองต้องทำการ config ค่าต่างๆ ของกล้อง webcam ที่ต่ออยู่กับ raspberry pi โดยเข้าไปแก้ไขในไฟล์ gst-server.sh ด้วยคำสั่ง nano gst-server.sh บนหน้าจอ terminal ของ raspberry pi ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารทบทวนวิชาสำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to
permitted by applicable law.
Last login: Mon Feb 3 00:51:47 2014
^[[Api@raspberrypi$ sudo -i
root@raspberrypi:~# nano /var/www/control2.php
root@raspberrypi:~# chmod 755 /var/www/control2.php
root@raspberrypi:~# nano gst-server.sh
root@raspberrypi:~# nano gst-server.sh
```

รูปที่ 4.6 คำสั่ง nano gst-server.sh บนหน้าจอ terminal

ทำการแก้ไข และ config ค่า IP address และ port ที่ใช้ในการเชื่อมต่อกล้อง webcam บน raspberry pi โดยค่า IP address ที่ใช้ เป็นของ raspberry pi (server) ซึ่งคือ "161.246.18.233" และ port ที่ใช้ต่อกับกล้อง webcam คือ 8081 ดังรูปที่ 4.7

```
GNU nano 2.2.6 File: gst-server.sh
#!/bin/tcsh

myip=161.246.18.233
port=8081
width=320
height=240

gst-launch \
v4l2src ! \
ffmpegcolorspace ! \
video/x-raw-yuv,width=${width},height=${height},framerate=${framerate} ! \
jpegenc ! \
tcpserver sink host=${myip} port=${port} sync=false
```

รูปที่ 4.7 การ config ค่าของกล้อง webcam

4.5 เริ่มการใช้งานกล้อง webcam

จากนั้น ทำการเริ่มการใช้งานกล้อง webcam ที่ต่อบน raspberry pi โดยการรันไฟล์ gst-server.sh ด้วยคำสั่ง ./gst-server.sh บนหน้าจอ terminal ของ raspberry pi เมื่อทำการรันไฟล์ gst-server.sh บนหน้าจอ terminal แล้วจะแสดงสถานะของกล้อง webcam ที่ต่อกับ raspberry pi พร้อมใช้งานดังรูปที่ 4.8

```

tv4l2src:tv4l2src0:
error requesting 2 buffers: Device or resource busy
Setting pipeline to NULL ...
Freeing pipeline ...
root@raspberrypi:~# nano gst-server.sh
root@raspberrypi:~# nano /etc/motion/motion.conf
root@raspberrypi:~# nano gst-server.sh
root@raspberrypi:~# ./gst-server.sh
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock

```

รูปที่ 4.8 สถานะของกล้อง webcam พร้อมใช้งาน

4.6 เริ่มต้นการทำงาน android application

ต่อจากนั้น ทดลองรันโปรแกรมสำหรับ android application จะปรากฏหน้าจอ interface บน android application สำหรับกรอกค่าต่างๆที่ใช้ต่อกับกล้อง webcam บน raspberry pi เพื่อให้แสดงผลบน หน้าจอ android application ดังรูปที่ 4.9

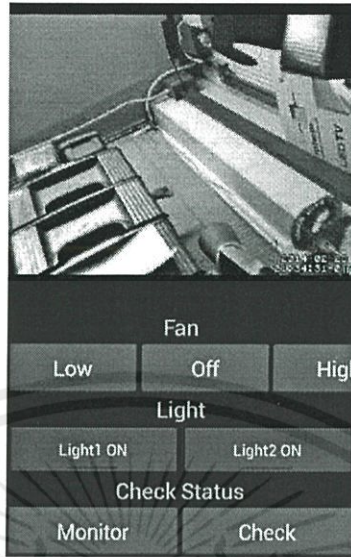


รูปที่ 4.9 หน้าจอ interface บน android application

กรอกค่า IP address และ port ของ raspberry pi และกล้อง webcam ที่ใช้ในการเชื่อมต่อ ซึ่งค่า IP address ของ raspberry pi (server) คือ “161.246.18.233” และค่า port ที่ใช้ต่อกับกล้อง webcam บน raspberry pi คือ 8081

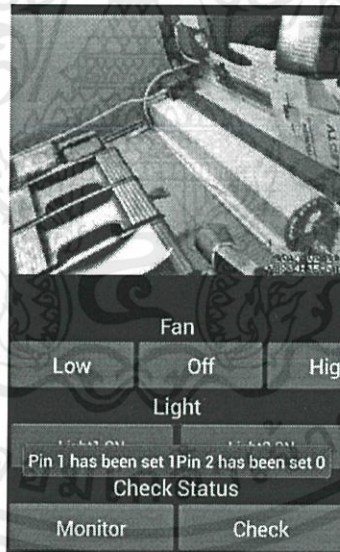
จากนั้นกด start button จะปรากฏหน้าจอ interface ที่ใช้สำหรับควบคุมอุปกรณ์ไฟฟ้าผ่าน android application พร้อมทั้งแสดงภาพแบบ real time ที่ได้จากกล้อง webcam ที่ต่ออยู่บน raspberry pi ดังแสดงในรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าจอ interface สำหรับควบคุมอุปกรณ์ไฟฟ้า

จากนั้นทดลองป้อนค่า input ต่างๆเข้าไป โดยการกดปุ่มต่างๆ เพื่อควบคุมอุปกรณ์ไฟฟ้าตามที่ต้องการ ดังในรูปที่ 4.11



รูปที่ 4.11 การป้อนค่า input ต่างๆเข้าไปบนหน้าจอ interface

จากรูปที่ 4.11 จะเห็นได้ว่า เมื่อเราทำการกดปุ่ม ต่างๆ บนหน้าจอ interface ของ android application จะมีข้อความบอกสถานะของอุปกรณ์ไฟฟ้าที่รับเข้ามาของปุ่มที่ทำการกดปรากฏขึ้นมา ยกตัวอย่างเช่น ในรูปที่ 4.11 เมื่อผู้ทำการทดลองทำการกดปุ่ม Fan Low ก็จะมีข้อความปรากฏขึ้นมาว่า “Pin 1 has been set 1 Pin 2 has been set 0” เป็นต้น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 เก็บผลการทดลองจากระบบฐานข้อมูล

เมื่อทำการเก็บผลการทดลองจากระบบฐานข้อมูลหลังจากป้อนข้อมูลต่างๆเข้าไปทาง Android Application แล้ว จะได้ผลดังรูปที่ (4.12)

+ Options

← T →		pinID	pinNumber	pinStatus	monitor
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		1	4	1	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		2	17	0	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		3	18	0	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		4	21	0	0

รูปที่ 4.12 ผลลัพธ์ที่ได้จากระบบฐานข้อมูล

4.8 จัดเก็บผลการทดลองที่ได้จากหน้าต่าง LXTERMINAL

ผลการทดลองที่ได้จากหน้าต่าง LXTERMINAL บนระบบสมองกลฝังตัวเป็นการแสดงผลลัพธ์ที่ได้จากสถานะของแต่ละ port ของ GPIO ดังรูปที่ (4.13)

```

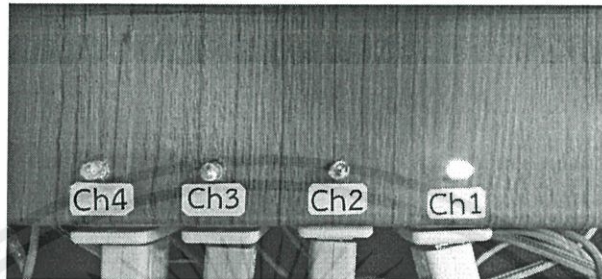
161.246.18.193 - PuTTY
Monitor GPIO17 Turned Off
Monitor GPIO18 Turned Off
Monitor GPIO21 Turned Off
GPIO 4 Turned On
GPIO 17 Turned Off
GPIO 18 Turned Off
GPIO 21 Turned Off
Monitor GPIO4 Turned On
Monitor GPIO17 Turned Off
Monitor GPIO18 Turned Off
Monitor GPIO21 Turned Off
  
```

รูปที่ 4.13 ผลลัพธ์ที่ได้บนหน้าจอ LXTerminal บนสมองกลฝังตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 จัดเก็บผลการทดลองที่ได้จาก GPIO

ผลการทดลองที่ได้จาก GPIO ของระบบสมองกลฝังตัวจากสถานะของหลอด LED ดังรูปที่ 4.14

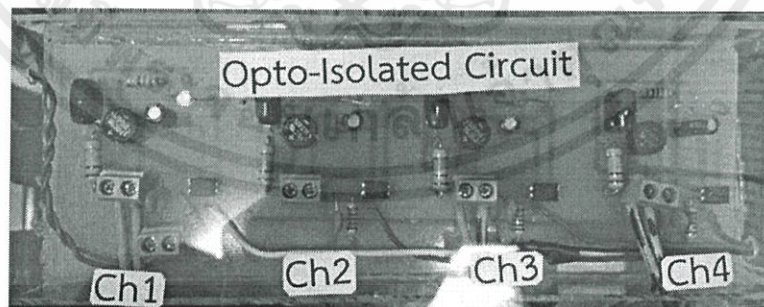


รูปที่ 4.14 ผลลัพธ์ที่ได้จาก GPIO ของสมองกลฝังตัว

จากรูปที่ 4.14 เป็นผลลัพธ์สุดท้ายที่ได้ GPIO ของระบบสมองกลฝังตัวซึ่งแสดงสถานะการทำงานโดยการติดดับของหลอด LED เมื่อป้อนข้อมูลขาเข้าผ่านแอปพลิเคชันแอนดรอยด์เป็นค่าต่างๆเพื่อควบคุมให้ LED ทำงานได้ตามต้องการ

4.10 ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

เมื่อ LED ทำงานสถานะของ LED เปรียบเสมือนสถานะของอุปกรณ์ไฟฟ้าเมื่ออุปกรณ์ไฟฟ้าทำงานแล้วจะมีแรงดัน AC 220V เข้ามาในวงจรตรวจจับแรงดัน 220V Isolated ดังรูปที่ 4.15 จากนั้นจะถูกเปลี่ยนแรงดันเป็นแรงดันดิจิตอล เข้าไปตรวจสอบในระบบสมองกลฝังตัวเพื่อแสดงผลว่าอุปกรณ์ไฟฟ้านั้นทำงาน โดยข้อมูลที่ได้รับมาจะถูกพักไว้ในระบบฐานข้อมูลของระบบสมองกลฝังตัวใน monitor field ดังแสดงในรูปที่ 4.16



รูปที่ 4.15 วงจรตรวจจับแรงดัน 220V Isolated

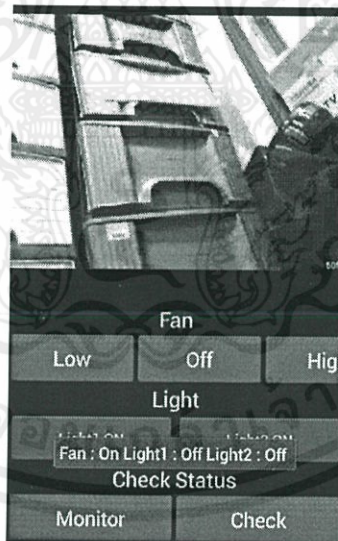
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ Options				pinID	pinNumber	pinStatus	monitor			
<input type="checkbox"/>		Edit		Copy		Delete	1	4	1	1
<input type="checkbox"/>		Edit		Copy		Delete	2	17	0	0
<input type="checkbox"/>		Edit		Copy		Delete	3	18	0	0
<input type="checkbox"/>		Edit		Copy		Delete	4	21	0	0

รูปที่ 4.16 monitor field ในระบบฐานข้อมูล

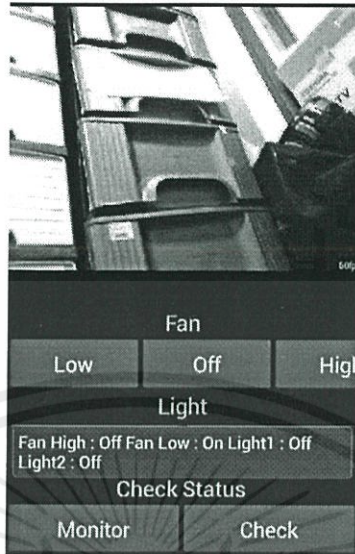
จากรูปที่ 4.16 monitor field จะปรากฏสถานะของอุปกรณ์ไฟฟ้าที่รับเข้ามาเพื่อแสดงว่าถ้าอุปกรณ์ไฟฟ้า On คือโลจิก “1” และ Off คือโลจิก “0”

และเมื่อผู้ทำการทดลองทำการตรวจสอบสถานะของอุปกรณ์ไฟฟ้าจากหน้าจอแอปพลิเคชันแอนดรอยด์บน Smartphone โดยการกดปุ่ม Monitor ก็จะมีปรากฏสถานะของอุปกรณ์ไฟฟ้าชนิดต่างๆ ดังรูปที่ 4.17 และตรวจสอบสถานะล่าสุดที่ผู้ทำการทดลองได้ทำการกดปุ่มเพื่อควบคุมที่ช่องทางต่างๆ โดยการกดปุ่ม Check ก็จะมีปรากฏสถานะล่าสุดที่ได้ทำการกดปุ่มควบคุม ดังรูปที่ 4.18



รูปที่ 4.17 สถานะของอุปกรณ์ไฟฟ้าชนิดต่างๆ

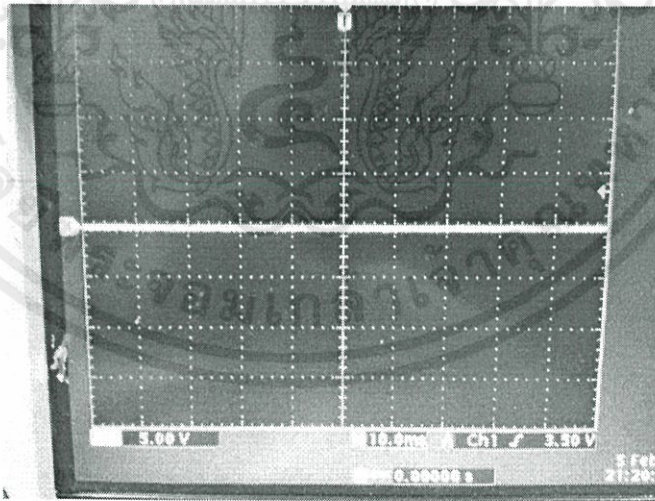
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 สถานะล่าสุดที่กดปุ่มควบคุมช่องทางต่างๆ

4.11 ระดับสัญญาณดิจิตอลที่วัดได้หลังจากผ่านวงจรตรวจจับแรงดัน 220V

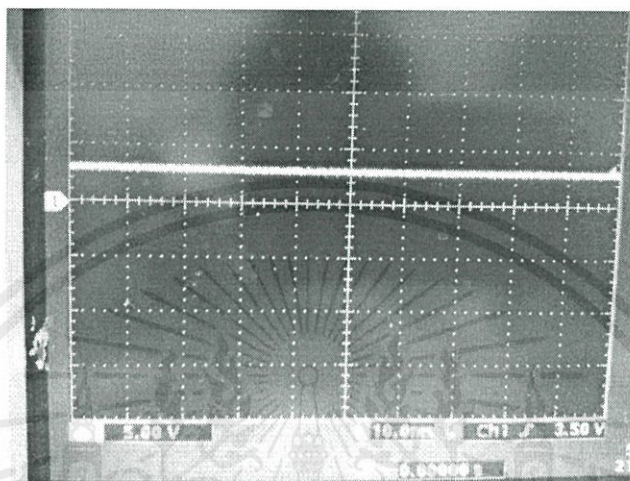
เมื่อวัดแรงดันทางดิจิตอล หลังจากผ่านชุดวงจรตรวจจับแรงดัน 220V Isolated ก่อนเข้าระบบสมองกลฝังตัว จะเห็นได้ว่าเมื่ออุปกรณ์ไฟฟ้าทำงานจะมีแรงดัน AC220V เข้ามายังชุดวงจรตรวจจับแรงดัน 220V Isolated และจะถูกแปลงเป็นแรงดันทางดิจิตอล (บิตที่มีค่าลอจิกเป็น “0”) ที่มีค่า 0V ดังรูปที่ 4.19



รูปที่ 4.19 ระดับสัญญาณดิจิตอลที่วัดได้เมื่ออุปกรณ์ไฟฟ้าทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อวัดแรงดันทางดิจิทัล หลังจากผ่านชุดวงจรตรวจจับแรงดัน 220V Isolated ก่อนเข้าระบบสมองกลฝังตัว จะเห็นได้ว่าเมื่ออุปกรณ์ไฟฟ้าไม่ทำงานจะมีแรงดัน AC220V เข้ามายังชุดวงจรตรวจจับแรงดัน 220V Isolated และจะถูกแปลงเป็นแรงดันทางดิจิทัล (บิตที่มีค่าลอจิกเป็น “1”) ที่มีค่า 3.3V ดังรูปที่ 4.20



รูปที่ 4.20 ระดับสัญญาณดิจิทัลที่วัดได้เมื่ออุปกรณ์ไฟฟ้าไม่ทำงาน

แสดงให้เห็นว่าสามารถที่จะตรวจสอบสถานะของอุปกรณ์ไฟฟ้าที่เชื่อมต่อแต่ละช่องสัญญาณ จากการรับแรงดัน AC 220V. มาจากอุปกรณ์ไฟฟ้า จากนั้นแรงดันถูกแปลงไปเป็นแรงดัน TTL เพื่อรับค่าแรงดัน input เข้าทางระบบสมองกลฝังตัวเพื่อแสดงให้เห็นว่าอุปกรณ์ไฟฟ้าแต่ละช่องสัญญาณนั้นทำงานอยู่จริง

4.12 ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

เมื่อเราดูที่อุปกรณ์ไฟฟ้าที่ต่ออยู่กับ GPIO ของ Raspberry Pi ก็จะได้เห็นว่าอุปกรณ์ไฟฟ้าในแต่ละช่องทางนั้น สามารถทำงานได้ตามที่เราควบคุมจากหน้าแอปพลิเคชันแอนดรอยด์ ดังรูปที่ 4.21 ซึ่งผู้ทำการทดลองได้ทดลองกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดพัดลมในระดับต่ำ (Fan Low) ในข้อที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

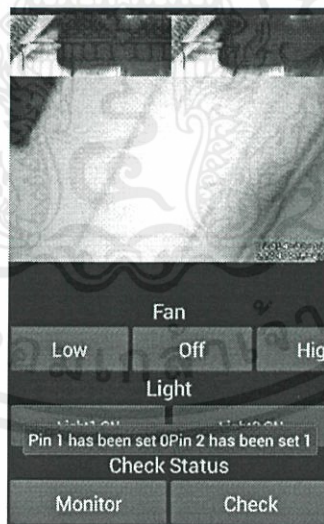


รูปที่ 4.21 สถานะพัดลมทำงานในระดับต่ำ

4.13 เปรียบเทียบผลที่ได้จากการทดลองเมื่อป้อนข้อมูลต่างๆ

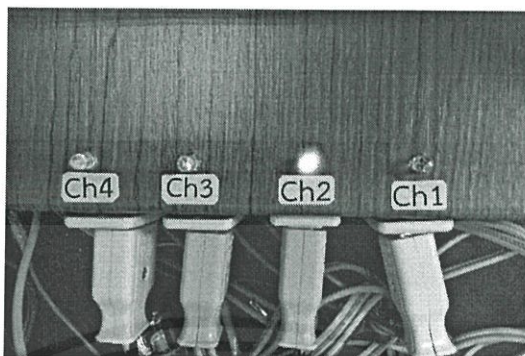
เมื่อผู้ทำการทดลองทดลองป้อนข้อมูลเข้าโดยการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดต่างๆจากแอปพลิเคชันแอนดรอยด์บน Smartphone แล้วทำการเก็บผลการทดลองที่ได้จากส่วนต่างๆดังที่กล่าวมา จะสามารถเปรียบเทียบผลการทดลองในส่วนต่างๆได้ดังนี้

- เมื่อกดปุ่มเพื่อควบคุมให้พัดลมติดในระดับสูง (Fan High)



รูปที่ 4.22 กดปุ่ม Fan High บนหน้าแอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

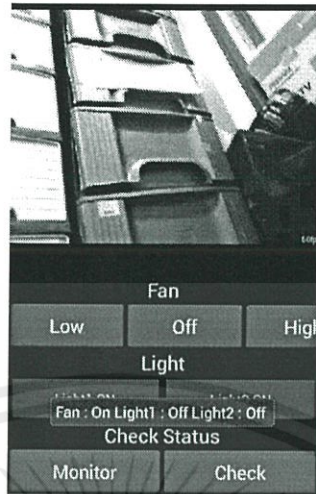


รูปที่ 4.23 LED ช่องทางที่ 2 ติด



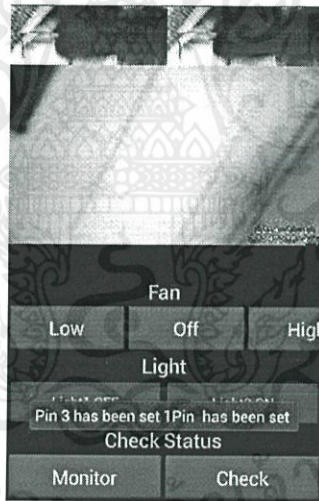
รูปที่ 4.24 พัดลมติดในระดับสูง (Fan High)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



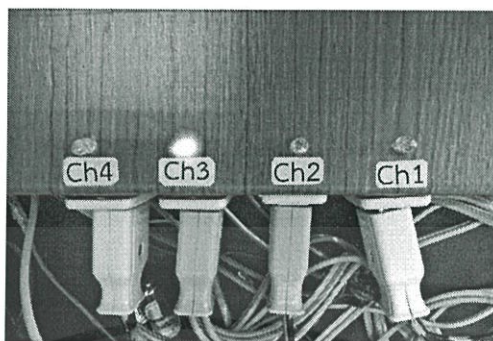
รูปที่ 4.25 ปรากฏสถานะของอุปกรณ์ไฟฟ้าเมื่อกดปุ่ม Monitor

- เมื่อกดปุ่มเพื่อควบคุมให้หลอดไฟดวงที่ 1 ติด

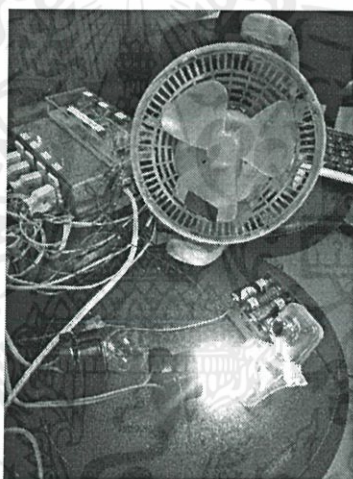


รูปที่ 4.26 เมื่อกดปุ่ม Light1: On บนหน้าแอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

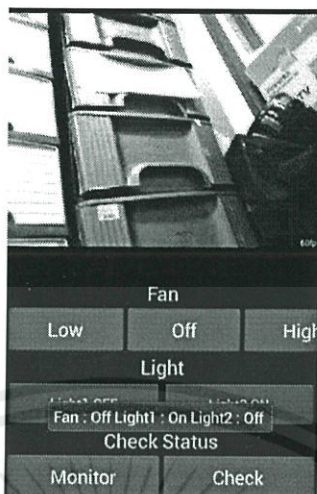


รูปที่ 4.27 สถานะ LED ช่องทางที่ 3 ติด



รูปที่ 4.28 หลอดไฟดวงที่ 1 ติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 ปรากฏสถานะของอุปกรณ์ไฟฟ้าชนิดต่างๆเมื่อกดปุ่ม Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

จากการศึกษาและสร้างระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone พบว่า Raspberry PI เป็นคอมพิวเตอร์ขนาดเล็กที่สามารถสร้างเป็นเซิร์ฟเวอร์ด้วยโปรแกรม Apache PHP และสร้างฐานข้อมูลด้วยโปรแกรม MySQL รวมทั้งใช้โปรแกรม phpMyAdmin เป็นตัวจัดการฐานข้อมูล ในส่วนของการออกแบบและสร้างแอปพลิเคชันแอนดรอยด์จะใช้โปรแกรมอีคลิปส์ (Eclipse) โดยใช้ภาษาในการสร้างด้วยภาษา JAVA เมื่อสร้างองค์ประกอบทั้งสองส่วนสำเร็จ จึงนำมาเชื่อมโยงเพื่อให้สามารถควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเทอร์เน็ตได้

5.1 สรุปผล

จากการทดลองเพื่อศึกษาและสร้างระบบควบคุมอุปกรณ์ไฟฟ้าด้วยระบบไฟฟ้าสมองกลฝังตัว พบว่าเมื่อผู้ทำการทดลองป้อนข้อมูลเข้าผ่านทางแอปพลิเคชันแอนดรอยด์(Client) เป็นค่าต่างๆ ข้อมูลจะถูกส่งเข้าไปเก็บไว้ในระบบฐานข้อมูล จากนั้น ระบบสมองกลฝังตัว (Server) จะทำการตรวจสอบค่าที่ได้รับเข้ามาจากระบบฐานข้อมูล เพื่อนำไปควบคุม GPIO ของระบบสมองกลฝังตัว ให้สามารถทำงานร่วมกับชุดรีเลย์สวิตช์ และสามารถภาพสถานการณ์ทำงานของเครื่องใช้ไฟฟ้าจากกล้อง Web camera พร้อมกับการส่งค่ากลับไปยังแอปพลิเคชันแอนดรอยด์เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

5.2 ข้อเสนอแนะ

ในการทำวิทยานิพนธ์ฉบับนี้ เมื่อทำการวิเคราะห์ผลแล้ว ผลการทดลองที่ได้ข้างต้นยังมีข้อผิดพลาดอยู่ ซึ่งอาจเกิดจากการออกแบบและสร้างที่อาจจะยังไม่ดีเท่าที่ควร และผู้ทดลองขาดประสบการณ์ในการออกแบบและสร้างทั้งซอฟต์แวร์และฮาร์ดแวร์ รวมทั้งความประมาทในการจัดการระยะเวลาในการดำเนินงาน สิ่งเหล่านี้อาจทำให้ผลการทดลองเกิดความผิดพลาดและไม่เป็นไปตามแผนการดำเนินงานที่วางไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Raspberry pi controlling from web browser. [ออนไลน์]. :
<http://iqjar.com/jar/controlling-the-raspberry-pis-gpio-pins-from-a-web-browser>.
- [2] Raspberry pi and android simple remote control. [ออนไลน์]. :
<http://morethanuser.blogspot.com/2013/05/raspberry-pi-and-android-simple-remote.html>.
- [3] Raspberry Pi Web Server. [ออนไลน์]. : <http://www.instructables.com/id/Raspberry-Pi-Web-Server>.
- [4] Webcam Steaming By Raspberry Pi. [ออนไลน์]. :
<http://www.youtube.com/watch?v=Wsm0EvRGCTE&feature=youtu.be>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE PHP (SERVER)

```
//name : control4.php
```

```
<?php
mysql = mysql_connect("localhost","gpio","admin");
mysqlDB = "gpio";
mysql_select_db($mysqlDB,$mysql);

$cid=$_GET['id'];
$status=$_GET['status'];
$cid1=$_GET['id1'];
$status1=$_GET['status1'];

$query_control = sprintf("UPDATE `pinStatus` SET `pinStatus`=%s where
`pinID`=%s",$status,$cid);
mysql_query($query_control,$mysql);
$query_control1 = sprintf("UPDATE `pinStatus` SET `pinStatus`=%s where
`pinID`=%s",$status1,$cid1);
mysql_query($query_control1,$mysql);

echo "Pin ".$cid." has been set ".$status;
echo "Pin ".$cid1." has been set ".$status1;
?>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE PHP (SERVER)

```
//name : control2.php
```

```
<?php
```

```
$mysql = mysql_connect("localhost","gpio","admin");
```

```
$mysqlDB = "gpio";
```

```
mysql_select_db($mysqlDB,$mysql);
```

```
$id=$_GET['id'];
```

```
$status=$_GET['status'];
```

```
$query_control = sprintf("UPDATE `pinStatus` SET `pinStatus`=%s where  
`pinID`=%s",$status,$id);
```

```
mysql_query($query_control,$mysql);
```

```
$SQL = "SELECT * FROM pinStatus WHERE pinID='1'";
```

```
$result = mysql_query($SQL,$mysql);
```

```
$row = mysql_fetch_array($result, MYSQL_ASSOC);
```

```
IF ($row['monitor']=="1"){
```

```
    echo "Fan : On ";
```

```
}
```

```
else IF ($row['monitor']=="0"){
```

```
    echo "Fan : Off ";
```

```
}
```

```
$SQL = "SELECT * FROM pinStatus WHERE pinID='3'";
```

```
$result = mysql_query($SQL,$mysql);
```

```
$row = mysql_fetch_array($result, MYSQL_ASSOC);
```

```
IF ($row['monitor']=="1"){
```

```
    echo "Light1 : On ";
```

```
}
```

```
else IF ($row['monitor']=="0"){
```

```
    echo "Light1 : Off ";
```

```
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
```

```
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
$SQL = "SELECT * FROM pinStatus WHERE pinID='4'";
```

```
$result = mysql_query($SQL,$mysql);  
$row = mysql_fetch_array($result, MYSQL_ASSOC);  
IF ($row['monitor']=="1"){  
    echo "Light2 : On ";  
}  
else IF ($row['monitor']=="0"){  
    echo "Light2 : Off ";  
}  
?>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE CONTROL GPIO SHELL SCRIPT (SERVER)

```
mysqlusername="gpio"  
mysqlpassword="admin"
```

```
#Set Refresh  
echo "How long do you want the wait time to be?"  
read waitTime
```

```
#Invoke GPIO  
echo "4" > /sys/class/gpio/export  
echo "17" > /sys/class/gpio/export  
echo "18" > /sys/class/gpio/export  
echo "21" > /sys/class/gpio/export
```

```
#Start Loop  
while :  
do  
#Read MySQL Data  
#Direction  
direction4=$(mysql -B --disable-column-names --user=$mysqlusername --  
password=$mysqlpassword gpio -e "SELECT pinDirection FROM pinDirection WHERE pinNumber=$  
direction17=$(mysql -B --disable-column-names --user=$mysqlusername --  
password=$mysqlpassword gpio -e "SELECT pinDirection FROM pinDirection WHERE pinNumber=$  
direction18=$(mysql -B --disable-column-names --user=$mysqlusername --  
password=$mysqlpassword gpio -e "SELECT pinDirection FROM pinDirection WHERE pinNumber=$  
direction21=$(mysql -B --disable-column-names --user=$mysqlusername --  
password=$mysqlpassword gpio -e "SELECT pinDirection FROM pinDirection WHERE pinNumber=$
```

```
#Run Commands
```

```
if [ "$direction4" == "out" ]; then  
    echo "out" > /sys/class/gpio/gpio4/direction  
    if [ "$status4" == "1" ]; then  
        echo "1" > /sys/class/gpio/gpio4/value  
        echo "GPIO 4 Turned On"  
    else  
        echo "0" > /sys/class/gpio/gpio4/value  
        echo "GPIO 4 Turned Off"  
    fi  
else  
    echo "in" > /sys/class/gpio/gpio4/direction
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fi

if [ "$direction17" == "out" ]; then
    echo "out" > /sys/class/gpio/gpio17/direction
    if [ "$status17" == "1" ]; then
        echo "1" > /sys/class/gpio/gpio17/value
        echo "GPIO 17 Turned On"
    else
        #echo "0" > /sys/class/gpio/gpio17/value
        echo "GPIO 17 Turned Off"
    fi
else
    echo "in" > /sys/class/gpio/gpio17/direction
fi

if [ "$direction18" == "out" ]; then
    echo "out" > /sys/class/gpio/gpio18/direction
    if [ "$status18" == "1" ]; then
        echo "1" > /sys/class/gpio/gpio18/value
        echo "GPIO 18 Turned On"
    else
        #echo "0" > /sys/class/gpio/gpio18/value
        echo "GPIO 18 Turned Off"
    fi
else
    echo "in" > /sys/class/gpio/gpio18/direction
fi

if [ "$direction21" == "out" ]; then
    echo "out" > /sys/class/gpio/gpio21/direction
    if [ "$status21" == "1" ]; then
        echo "1" > /sys/class/gpio/gpio21/value
        echo "GPIO 21 Turned On"
    else
        echo "0" > /sys/class/gpio/gpio21/value
        echo "GPIO 21 Turned Off"
    fi
else
    echo "in" > /sys/class/gpio/gpio21/direction
fi

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆก็ตามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#Complete Loop
sleep $waitTime
done

```

CODE CAMERA SHELL SCRIPT (SERVER)

```
#!/bin/tcsh
```

```
set myip=161.246.18.233
```

```
set port=8081
```

```
set width=320
```

```
set height=240
```

```
gst-launch\
```

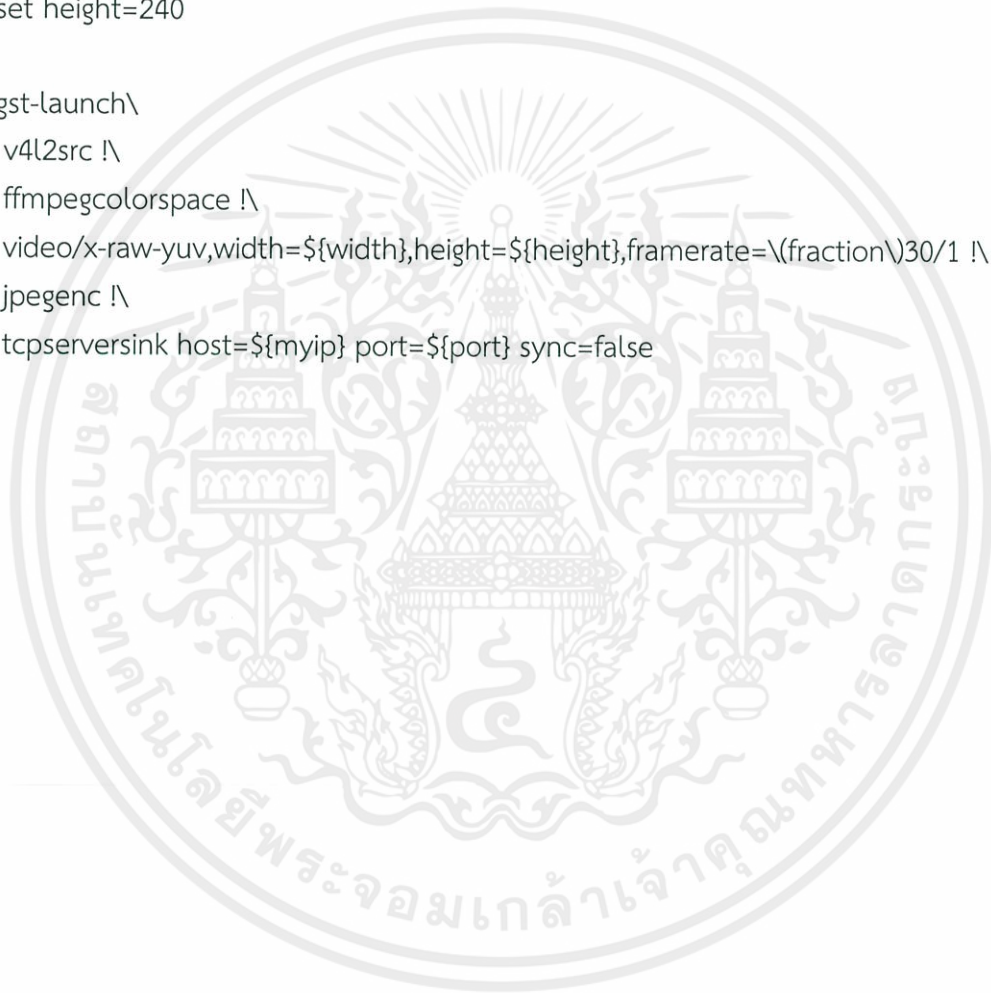
```
v4l2src !\
```

```
ffmpegcolorspace !\
```

```
video/x-raw-yuv,width=${width},height=${height},framerate=\(fraction\)30/1 !\
```

```
jpegenc !\
```

```
tcpserver sink host=${myip} port=${port} sync=false
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE GPIO READ DATA SHELL SCRIPT (SERVER)

```
#!/bin/bash
```

```
mysqlusername="gpio"  
mysqlpassword="admin"
```

```
while :  
do
```

```
echo "22" > /sys/class/gpio/export  
echo "in" > /sys/class/gpio/gpio22/direction  
data=$( cat /sys/class/gpio/gpio22/value)  
if [ "$data" == "0" ]; then  
    echo "GPIO4 Turned On"  
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e  
"UPDATE pinStatus SET monitor='1' WHERE pinID='1';"  
else  
    echo "GPIO4 Turned Off"  
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e  
"UPDATE pinStatus SET monitor='0' WHERE pinID='1';"  
fi  
echo "22" > /sys/class/gpio/unexport
```

```
echo "23" > /sys/class/gpio/export  
echo "in" > /sys/class/gpio/gpio23/direction  
data=$( cat /sys/class/gpio/gpio23/value)  
if [ "$data" == "0" ]; then  
    echo "GPIO17 Turned On"  
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e  
"UPDATE pinStatus SET monitor='1' WHERE pinID='2';"  
else  
    echo "GPIO17 Turned Off"  
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e  
"UPDATE pinStatus SET monitor='0' WHERE pinID='2';"  
fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัย กรุณาติดต่อฝ่ายวิชาการของเอกสารทุกครั้งที่มีการนำไปใช้

```

echo "23" > /sys/class/gpio/unexport

echo "24" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio24/direction
data=$( cat /sys/class/gpio/gpio24/value)
if [ "$data" == "0" ]; then
    echo "GPIO18 Turned On"
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e
"UPDATE pinStatus SET monitor='1' WHERE pinID='3'";
else
    echo "GPIO18 Turned Off"
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e
"UPDATE pinStatus SET monitor='0' WHERE pinID='3'";
fi
echo "24" > /sys/class/gpio/unexport

echo "25" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio25/direction
data=$( cat /sys/class/gpio/gpio25/value)
if [ "$data" == "0" ]; then
    echo "GPIO21 Turned On"
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e
"UPDATE pinStatus SET monitor='1' WHERE pinID='4'";
else
    echo "GPIO21 Turned Off"
    mysql --user=$mysqlusername --password=$mysqlpassword gpio -e
"UPDATE pinStatus SET monitor='0' WHERE pinID='4'";
fi
echo "25" > /sys/class/gpio/unexport

sleep $waitTime
done

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA IP CONFIG (CLIENT)

```
package com.camera.simplemjpeg;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.content.SharedPreferences;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.EditText;
```

```
public class PreferenceActivity extends Activity {  
    public static final String KEY_HOSTNAME = "hostname";  
    public static final String KEY_PORTNUM = "portnum";  
    public static final String KEY_WIDTH = "width";  
    public static final String KEY_HEIGHT = "height";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
  
        this setContentView( R.layout.preference);  
  
        // load stored data  
        SharedPreferences sp = this.getSharedPreferences( MODE_PRIVATE);  
        String hostname = sp.getString( KEY_HOSTNAME, this.getString(  
R.string.defaultHostName));  
        String portnum = sp.getString( KEY_PORTNUM, this.getString(  
R.string.defaultPortNum));  
        String width = sp.getString( KEY_WIDTH, this.getString(  
R.string.defaultWidth));  
        String height = sp.getString( KEY_HEIGHT, this.getString(  
R.string.defaultHeight));  
  
        // set stored parameters to the contents
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EditText et = (EditText)findViewById( R.id.editText_hostname);
et.setText( hostname);
et = (EditText)findViewById( R.id.editText_portnum);
et.setText( portnum);
et = (EditText)findViewById( R.id.editText_width);
et.setText( width);
et = (EditText)findViewById( R.id.editText_height);
et.setText( height);
}

/*
 * This function is called when user clicks start button.
 */
public void onClick( View view){
    // get data from EditText components
    EditText etHost = (EditText)findViewById( R.id.editText_hostname);
    EditText etPort = (EditText)findViewById( R.id.editText_portnum);
    EditText etWidth = (EditText)findViewById( R.id.editText_width);
    EditText etHeight = (EditText)findViewById( R.id.editText_height);
    String hostname = etHost.getText().toString();
    String portnum = etPort.getText().toString();
    String width = etWidth.getText().toString();
    String height = etHeight.getText().toString();

    // store the input data
    SharedPreferences sp = this.getSharedPreferences( MODE_PRIVATE);
    SharedPreferences.Editor editor = sp.edit();
    editor.putString( KEY_HOSTNAME, hostname);
    editor.putString( KEY_PORTNUM, portnum);
    editor.putString( KEY_WIDTH, width);
    editor.putString( KEY_HEIGHT, height);
    editor.commit();

    // set the image size
    MjpegView.setImageSize( Integer.parseInt( width),
    Integer.parseInt(height));
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ถือว่าผิดกฎหมาย

```
// launch MjpegActivity
Intent intent = new Intent( this, MjpegActivity.class);
intent.putExtra( KEY_HOSTNAME, hostname);
intent.putExtra( KEY_PORTNUM, portnum);
this.startActivity( intent);
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA IP CONFIG LAYOUT (CLIENT)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hostname"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <EditText
        android:id="@+id/editText_hostname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textUri" >

        <requestFocus />
    </EditText>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/portnum"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <EditText
        android:id="@+id/editText_portnum"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textUri" >
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น

อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
android:ems="10"  
android:inputType="number" />
```

```
<LinearLayout
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content" >
```

```
<TextView
```

```
android:id="@+id/textView3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/width"  
android:textAppearance="?android:attr/textAppearanceSmall" />
```

```
<EditText
```

```
android:id="@+id/editText_width"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_weight="1"  
android:ems="10"  
android:inputType="number" />
```

```
<TextView
```

```
android:id="@+id/textView4"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/height"  
android:textAppearance="?android:attr/textAppearanceSmall" />
```

```
<EditText
```

```
android:id="@+id/editText_height"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_weight="1"
```

```
android:ems="10"  
android:inputType="number" />
```

เอกสารนี้เป็นเอกสารที่
ไม่ว่ากรณีใดๆทั้งสิ้น

เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</LinearLayout>
```

```
<Button
```

```
    android:id="@+id/button1"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:onClick="@string/onclick"
```

```
    android:text="@string/start" />
```

```
</LinearLayout>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA CAMERA MJPEG VIEW (CLIENT)

```
package com.camera.simplemjpeg;

import java.io.IOException;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.PorterDuff;
import android.graphics.PorterDuffXfermode;
import android.graphics.Rect;
import android.graphics.Typeface;
import android.util.AttributeSet;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

public class MjpegView extends SurfaceView implements SurfaceHolder.Callback {

    public static final String TAG="MJPEG";

    public final static int POSITION_UPPER_LEFT = 9;
    public final static int POSITION_UPPER_RIGHT = 3;
    public final static int POSITION_LOWER_LEFT = 12;
    public final static int POSITION_LOWER_RIGHT = 6;

    public final static int SIZE_STANDARD = 1;
    public final static int SIZE_BEST_FIT = 4;
    public final static int SIZE_FULLSCREEN = 8;

    SurfaceHolder holder;

    Context saved_context;

    private MjpegViewThread thread;
    private MjpegInputStream mIn = null;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ กรุณาแจ้งผู้จัดทำเอกสารก่อน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private boolean showFps = false;
private boolean mRun = false;
private boolean surfaceDone = false;
```

```
private Paint overlayPaint;
private int overlayTextColor;
private int overlayBackgroundColor;
private int ovlPos;
private int dispWidth;
private int dispHeight;
private int displayMode;
```

```
private boolean suspending = false;
```

```
private Bitmap bmp = null;
// hard-coded image size
private static int imgWidth = 320;
private static int imgHeight = 240;
```

```
public class MjpegViewThread extends Thread {
private SurfaceHolder mSurfaceHolder;
private int frameCounter = 0;
private long start;
private String fps = "";
```

```
public MjpegViewThread(SurfaceHolder surfaceHolder, Context context) {
mSurfaceHolder = surfaceHolder;
}
```

```
private Rect destRect(int bmw, int bmh) {
int tempx;
int tempy;
```

```
if (displayMode == MjpegView.SIZE_STANDARD) {
```

```
tempx = (dispWidth / 2) - (bmw / 2);
```

```
tempy = (dispHeight / 2) - (bmh / 2);
```

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งขอสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการนำไปใช้

```

        return new Rect(tempx, tempy, bmw + tempx, bmh + tempy);
    }
    if (displayMode == MjpegView.SIZE_BEST_FIT) {
        float bmasp = (float) bmw / (float) bmh;
        bmw = dispWidth;
        bmh = (int) (dispWidth / bmasp);
        if (bmh > dispHeight) {
            bmh = dispHeight;
            bmw = (int) (dispHeight * bmasp);
        }
        tempx = (dispWidth / 2) - (bmw / 2);
        tempy = (dispHeight / 2) - (bmh / 2);
        return new Rect(tempx, tempy, bmw + tempx, bmh + tempy);
    }
    if (displayMode == MjpegView.SIZE_FULLSCREEN)
        return new Rect(0, 0, dispWidth, dispHeight);
    return null;
}

public void setSurfaceSize(int width, int height) {
    synchronized(mSurfaceHolder) {
        dispWidth = width;
        dispHeight = height;
    }
}
}

```

```

private Bitmap makeFpsOverlay(Paint p) {
    Rect b = new Rect();
    p.getTextBounds(fps, 0, fps.length(), b);

```

```

    // false indentation to fix forum layout
    Bitmap bm = Bitmap.createBitmap(b.width(), b.height(),
    Bitmap.Config.ARGB_8888);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น Canvas c = new Canvas(bm); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
p.setColor(overlayBackgroundColor);

```

c.drawRect(0, 0, b.width(), b.height(), p);
p.setColor(overlayTextColor);
c.drawText(fps, -b.left, b.bottom-b.top-p.descent(), p);
return bm;
}

```

```

public void run() {
    start = System.currentTimeMillis();
    PorterDuffXfermode mode = new
PorterDuffXfermode(PorterDuff.Mode.DST_OVER);

    int width;
    int height;
    Paint p = new Paint();
    Bitmap ovl=null;
    while (mRun) {
        Rect destRect=null;
        Canvas c = null;
        if(surfaceDone) {
            try {
                if bmp==null){
                    bmp = Bitmap.createBitmap(imgWidth, imgHeight,
Bitmap.Config.ARGB_8888);
                }
                mIn.readMjpegFrame(bmp);

```

```

destRect = destRect(bmp.getWidth(),bmp.getHeight());

```

```

c = mSurfaceHolder.lockCanvas();
synchronized (mSurfaceHolder) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ c.drawRect(bmp, null, destRect, p); เอกสารทุกครั้งที่มีการนำไปใช้


```

}

private void init(Context context) {

    //SurfaceHolder holder = getHolder();
    holder = getHolder();
    saved_context = context;
    holder.addCallback(this);
    thread = new MjpegViewThread(holder, context);
    setFocusable(true);
    overlayPaint = new Paint();
    overlayPaint.setTextAlign(Paint.Align.LEFT);
    overlayPaint.setTextSize(12);
    overlayPaint.setTypeface(Typeface.DEFAULT);
    overlayTextColor = Color.WHITE;
    overlayBackgroundColor = Color.BLACK;
    ovlPos = MjpegView.POSITION_LOWER_RIGHT;
    displayMode = MjpegView.SIZE_STANDARD;
    dispWidth = getWidth();
    dispHeight = getHeight();
}

public void startPlayback() {
    if(mIn != null) {
        mRun = true;
        thread.start();
    }
}
}

```

```

public void resumePlayback() {
    if(suspending){
        if(mIn != null) {

```

เอกสารนี้เป็นเอกสารที่ SurfaceHolder holder = getHolder(); นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีก holder.addCallback(this); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
thread = new MjpegViewThread(holder, saved_context);

```

        thread.start();
    }
    suspending=false;
}
}
public void stopPlayback() {
    if(mRun){
        suspending = true;
    }
    mRun = false;
    boolean retry = true;
    while(retry) {
        try {
            thread.join();
            retry = false;
        } catch (InterruptedException e) {}
    }
}

public static void setImageSize( int width, int height){
    MjpegView.imgWidth = width;
    MjpegView.imgHeight = height;
}

public void freeCameraMemory(){
    if(mIn!=null){
        mIn.freeCameraMemory();
    }
}

public MjpegView(Context context, AttributeSet attrs) {
    super(context, attrs); init(context);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ public void surfaceChanged(SurfaceHolder holder, int f, int w, int h) {มีการนำไปใช้
 thread.setSurfaceSize(w, h);

```

}

public void surfaceDestroyed(SurfaceHolder holder) {
    surfaceDone = false;
    stopPlayback();
}

public MjpegView(Context context) { super(context); init(context); }
public void surfaceCreated(SurfaceHolder holder) { surfaceDone = true; }
public void showFps(boolean b) { showFps = b; }
public void setSource(MjpegInputStream source) {
    mIn = source;
    startPlayback();
}
public void setOverlayPaint(Paint p) { overlayPaint = p; }
public void setOverlayTextColor(int c) { overlayTextColor = c; }
public void setOverlayBackgroundColor(int c) { overlayBackgroundColor = c; }
public void setOverlayPosition(int p) { ovlPos = p; }
public void setDisplayMode(int s) { displayMode = s; }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA CAMERA MJPEG INPUTSTREAM (CLIENT)

```
package com.camera.simplemjpeg;

import java.io.BufferedInputStream;
import java.io.ByteArrayInputStream;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStream;

import java.util.Properties;

import java.net.HttpURLConnection;
import java.net.URL;

import android.util.Log;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

public class MjpegInputStream extends DataInputStream {
    private final byte[] SOI_MARKER = { (byte) 0xFF, (byte) 0xD8 };
    private final byte[] EOF_MARKER = { (byte) 0xFF, (byte) 0xD9 };
    private final String CONTENT_LENGTH = "Content-Length";
    private final static int HEADER_MAX_LENGTH = 100;
    //private final static int FRAME_MAX_LENGTH = 40000 + HEADER_MAX_LENGTH;
    private final static int FRAME_MAX_LENGTH = 200000;
    private int mContentLength = -1;
    byte[] header = null;
    byte[] frameData = null;
    int headerLen = -1;
    int headerLenPrev = -1;
    int skip=1;
    int count=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private static final String TAG="MJPEG";
private static final boolean DEBUG=false;

static {
    System.loadLibrary("ImageProc");
}
public native void pixeltobmp(byte[] jp, int l, Bitmap bmp);
public native void freeCameraMemory();

public static MjpegInputStream read(String url) {
    try {
        URL url = new URL(url);
        HttpURLConnection urlConnection = (HttpURLConnection)
url.openConnection();
        return new MjpegInputStream(urlConnection.getInputStream());
    }catch(Exception e){}
    return null;
}

public MjpegInputStream(InputStream in) {
    super(new BufferedInputStream(in, FRAME_MAX_LENGTH));
}

private int getEndOfSequeunce(DataInputStream in, byte[] sequence)
    throws IOException
{
    int seqIndex = 0;
    byte c;
    for(int i=0; i < FRAME_MAX_LENGTH; i++) {
        c = (byte) in.readUnsignedByte();
        if(c == sequence[seqIndex]) {
            seqIndex++;
        }
        else {
            seqIndex = 0;
        }
        if(seqIndex == sequence.length){

```

เอกสารนี้เป็นเอกสาร if(c == sequence[seqIndex]) { การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีก seqIndex++; และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return i + 1;
    }
} else seqIndex = 0;
}

return -1;
}

private int getStartOfSequence(DataInputStream in, byte[] sequence)
    throws IOException
{
    int end = getEndOfSequence(in, sequence);
    return (end < 0) ? (-1) : (end - sequence.length);
}

private int getEndOfSequenceSimplified(DataInputStream in, byte[] sequence)
    throws IOException
{
    int startPos = (int)(0.9*mContentLength);
    int endPos = (int)(1.1*mContentLength);
    skipBytes(headerLen+startPos);

    int seqIndex = 0;
    byte c;
    for(int i=0; i < endPos-startPos ; i++) {
        c = (byte) in.readUnsignedByte();
        if(c == sequence[seqIndex]) {
            seqIndex++;
            if(seqIndex == sequence.length){
                return headerLen + startPos + i + 1;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

```
    } else seqIndex = 0;
}
```

```
    return -1;
}
```

```
private int parseContentLength(byte[] headerBytes)
    throws IOException, NumberFormatException, IllegalArgumentException
{
    ByteArrayInputStream headerIn = new ByteArrayInputStream(headerBytes);
    Properties props = new Properties();
    props.load(headerIn);
    return Integer.parseInt(props.getProperty(CONTENT_LENGTH));
}
```

```
public Bitmap readMjpegFrame() throws IOException {
    mark(FRAME_MAX_LENGTH);
    int headerLen;
    try{
        headerLen = getStartOfSequence(this, SOI_MARKER);
    }catch(IOException e){
        if(DEBUG) Log.d(TAG, "IOException in betting headerLen.");
        reset();
        return null;
    }
    reset();
```

```
    if(header==null || headerLen != headerLenPrev){
        header = new byte[headerLen];
        if(DEBUG) Log.d(TAG, "header renewed "+headerLenPrev+" -> "+headerLen);
    }
```

```
    headerLenPrev = headerLen;
```

```
    readFully(header);
```

```
    int ContentLengthNew = -1;
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try {
    ContentLengthNew = parseContentLength(header);
} catch (NumberFormatException nfe) {
    ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

    if(ContentLengthNew < 0){
        if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
        reset();
        ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
    }
} catch (IllegalArgumentException e) {
    if(DEBUG) Log.d(TAG,"IllegalArgumentException in parseContentLength");
    ContentLengthNew = getEndOfSequenceSimplified(this, EOF_MARKER);

    if(ContentLengthNew < 0){
        if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
        reset();
        ContentLengthNew = getEndOfSequence(this, EOF_MARKER);
    }
} catch (IOException e) {
    if(DEBUG) Log.d(TAG,"IOException in parseContentLength");
    reset();
    return null;
}
mContentLength = ContentLengthNew;
reset();

```

```

if(frameData==null){
    frameData = new byte[FRAME_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData renewed cl="+FRAME_MAX_LENGTH);
}

```

```

if(mContentLength + HEADER_MAX_LENGTH > FRAME_MAX_LENGTH){
    frameData = new byte[mContentLength + HEADER_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData renewed cl="+mContentLength +
    HEADER_MAX_LENGTH);
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆก็ตาม การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 จะถือว่าผิดกฎหมาย

```

        skipBytes(headerLen);

        readFully(frameData, 0, mContentLength);

        if(count++%skip==0){
            return BitmapFactory.decodeStream(new
ByteArrayInputStream(frameData,0,mContentLength));
        }else{
            return null;
        }
    }
}

public void readMjpegFrame(Bitmap bmp) throws IOException {
    mark(FRAME_MAX_LENGTH);
    int headerLen;
    try{
        headerLen = getStartOfSequence(this, SOI_MARKER);
    }catch(IOException e){
        if(DEBUG) Log.d(TAG,"IOException in betting headerLen.");
        reset();
        return;
    }
    reset();

    if(header==null || headerLen != headerLenPrev){
        header = new byte[headerLen];
        if(DEBUG) Log.d(TAG,"header renewed "+headerLenPrev+" -> "+headerLen);
    }
    headerLenPrev = headerLen;
    readFully(header);

    int ContentLengthNew = -1;
    try {
        ContentLengthNew = parseContentLength(header);
    } catch (NumberFormatException nfe) {

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น

เอกสารทุกครั้งที่มีการนำไปใช้

```

} catch (NumberFormatException nfe) {

```

```

ContentLengthNew = getEndOfSequeunceSimplified(this, EOF_MARKER);

if(ContentLengthNew < 0){
    if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
    reset();
    ContentLengthNew = getEndOfSequeunce(this, EOF_MARKER);
}
}catch (IllegalArgumentException e) {
    if(DEBUG) Log.d(TAG,"IllegalArgumentException in parseContentLength");
    ContentLengthNew = getEndOfSequeunceSimplified(this, EOF_MARKER);

    if(ContentLengthNew < 0){
        if(DEBUG) Log.d(TAG,"Worst case for finding EOF_MARKER");
        reset();
        ContentLengthNew = getEndOfSequeunce(this, EOF_MARKER);
    }
}catch (IOException e) {
    if(DEBUG) Log.d(TAG,"IOException in parseContentLength");
    reset();
    return;
}
mContentLength = ContentLengthNew;
reset();

if(frameData==null){
    frameData = new byte[FRAME_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData newed cl="+FRAME_MAX_LENGTH);
}
if(mContentLength + HEADER_MAX_LENGTH > FRAME_MAX_LENGTH){
    frameData = new byte[mContentLength + HEADER_MAX_LENGTH];
    if(DEBUG) Log.d(TAG,"frameData renewed cl="+mContentLength +
HEADER_MAX_LENGTH));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้ง skipBytes(headerLen); ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
readFully(frameData, 0, mContentLength);

if(count++%skip==0){
    pixeltobmp(frameData, mContentLength, bmp);
}else{
}
}
public void setSkip(int s){
    skip = s;
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA MAIN (CLIENT)

```
package com.camera.simplemjpeg;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
```

```
import java.net.Socket;
```

```
import java.net.UnknownHostException;
```

```
import org.apache.http.HttpEntity;
```

```
import org.apache.http.HttpResponse;
```

```
import org.apache.http.StatusLine;
```

```
import org.apache.http.client.ClientProtocolException;
```

```
import org.apache.http.client.HttpClient;
```

```
import org.apache.http.client.methods.HttpGet;
```

```
import org.apache.http.impl.client.DefaultHttpClient;
```

```
import com.camera.simple.R;
```

```
import com.camera.simplemjpeg.MjpegActivity;
```

```
import android.annotation.TargetApi;
```

```
import android.app.Activity;
```

```
import android.os.AsyncTask;
```

```
import android.os.Build;
```

```
import android.os.Bundle;
```

```
import android.os.StrictMode;
```

```
import android.util.Log;
```

```
import android.view.View;
```

```
import android.view.Window;
```

```
import android.view.WindowManager;
```

```
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
```

```
import android.widget.CompoundButton;
```

```
import android.widget.Toast;
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการ โทร. 0-2942-3000

```
import android.widget.ToggleButton;
```

```
@TargetApi(Build.VERSION_CODES.GINGERBREAD)
```

```
public class MjpegActivity extends Activity {
```

```
    private static final boolean DEBUG=false;
```

```
    private static final String TAG = "MJPEG";
```

```
    private MjpegView mv = null;
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        requestWindowFeature(Window.FEATURE_NO_TITLE);
```

```
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
        setContentView(R.layout.main);
```

```
        mv = (MjpegView) findViewById(R.id.mv);
```

```
        // receive parameters from PreferenceActivity
```

```
        Bundle bundle = getIntent().getExtras();
```

```
        String hostname = bundle.getString( PreferenceActivity.KEY_HOSTNAME);
```

```
        String portnum = bundle.getString( PreferenceActivity.KEY_PORTNUM);
```

```
        new DoRead().execute( hostname, portnum);
```

```
    }
```

```
    public void onResume() {
```

```
        if(DEBUG) Log.d(TAG,"onResume()");
```

```
        super.onResume();
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ไม่ว่ากรณีใดๆทั้งสิ้น

```
    }
```

```

}

public void onStart() {
    if(DEBUG) Log.d(TAG,"onStart()");
    super.onStart();
}

public void onPause() {
    if(DEBUG) Log.d(TAG,"onPause()");
    super.onPause();
    if(mv!=null){
        mv.stopPlayback();
    }
}

public void onStop() {
    if(DEBUG) Log.d(TAG,"onStop()");
    super.onStop();
}

public void onDestroy() {
    if(DEBUG) Log.d(TAG,"onDestroy()");

    if(mv!=null){
        mv.freeCameraMemory();
    }

    super.onDestroy();
}

```

```

public class DoRead extends AsyncTask<String, Void, MjpegInputStream> {

```

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยไม่หวังผลตอบแทนและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง Socket socket = null;

```

try {

```

```

        socket = new Socket( params[0], Integer.valueOf(
params[1]));

        return (new MjpegInputStream(socket.getInputStream()));
    } catch (UnknownHostException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    }
    return null;
}
}

```

```

protected void onPostExecute(MjpegInputStream result) {
    mv.setSource(result);
    if(result!=null) result.setSkip(1);
    mv.setDisplayMode(MjpegView.SIZE_BEST_FIT);
    mv.showFps(true);

    Button btn1 = (Button) findViewById(R.id.button1);
    Button btn2 = (Button) findViewById(R.id.button2);
    Button btn3 = (Button) findViewById(R.id.button3);
    Button btn4 = (Button) findViewById(R.id.button4);
    Button btn5 = (Button) findViewById(R.id.button5);
    ToggleButton btn6 = (ToggleButton) findViewById(R.id.toggleButton3);
    ToggleButton btn7 = (ToggleButton) findViewById(R.id.toggleButton4);

```

```

final String URL = "http://161.246.18.233/";

```

```

// Permission StrictMode

```

```

    if (android.os.Build.VERSION.SDK_INT > 9) {

```

```

        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
        .permitAll().build();

```

```

        StrictMode.setThreadPolicy(policy);
    }

    //Low
    btn1.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            String out0 =
                getHttpGet(URL+"control4.php?id=1&status=1&id1=2&status1=0");
            Toast.makeText(MjpegActivity.this,out0,
                Toast.LENGTH_SHORT).show();
        }
    });

    //Off
    btn2.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            String out0 =
                getHttpGet(URL+"control4.php?id=1&status=0&id1=2&status1=0");
            Toast.makeText(MjpegActivity.this,out0,
                Toast.LENGTH_SHORT).show();
        }
    });

    //High
    btn3.setOnClickListener(new OnClickListener() {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// TODO Auto-generated method stub

        String out0 =
        getHttpGet(URL+"control4.php?id=1&status=0&id1=2&status1=1");
        Toast.makeText(MjpegActivity.this,out0,
        Toast.LENGTH_SHORT).show();
    }
});

//Light1
btn6.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

    @Override
    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        // TODO Auto-generated method stub

        if(isChecked){
            String out1 =
            getHttpGet(URL+"control4.php?id=3&status=1");
            Toast.makeText(MjpegActivity.this,out1,
            Toast.LENGTH_SHORT).show();
        }else{
            String out1 =
            getHttpGet(URL+"control4.php?id=3&status=0");
            Toast.makeText(MjpegActivity.this,out1,
            Toast.LENGTH_SHORT).show();
        }
    }
});

```

```
//Light2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@Override
public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    // TODO Auto-generated method stub

    if(isChecked){
        String out1 =
getHttpGet(URL+"control4.php?id=4&status=1");

        Toast.makeText(MjpegActivity.this,out1, Toast.LENGTH_SHORT).show();
    }else{
        String out1 =
getHttpGet(URL+"control4.php?id=4&status=0");
        Toast.makeText(MjpegActivity.this,out1, Toast.LENGTH_SHORT).show();
    }
    });

//Monitor
btn4.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub

    String out0 =
getHttpGet(URL+"control2.php");

    Toast.makeText(MjpegActivity.this,out0, Toast.LENGTH_SHORT).show();
    }
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีสิทธิ์เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//Check
btn5.setOnClickListener(new OnClickListener() {

```

```

@Override
public void onClick(View v) {
    // TODO Auto-generated method stub

    String out0 = getHttpGet(URL+"control6.php");
    Toast.makeText(MjpegActivity.this,out0,
Toast.LENGTH_SHORT).show();
    }
});
}
}

```

```

public String getHttpGet(String url) {
    StringBuilder str = new StringBuilder();
    HttpClient client = new DefaultHttpClient();
    HttpGet httpGet = new HttpGet(url);
    try {
        HttpResponse response = client.execute(httpGet);
        StatusLine statusLine = response.getStatusLine();
        int statusCode = statusLine.getStatusCode();
        if (statusCode == 200) { // Status OK
            HttpEntity entity = response.getEntity();
            InputStream content = entity.getContent();
            BufferedReader reader = new BufferedReader(
                new InputStreamReader(content));
            String line;
            while ((line = reader.readLine()) != null) {
                str.append(line);
            }
        } else {
            Log.e("Log", "Failed to download result..");
        }
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

```
} catch (IOException e) {  
    e.printStackTrace();  
  
}  
return str.toString();  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE JAVA MAIN LAYOUT (CLIENT)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <com.camera.simplejpeg.MjpegView
        android:id="@+id/mv"
        android:layout_width="match_parent"
        android:layout_height="221dp"
        android:layout_weight="0.10" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Fan"
        android:textSize="@dimen/Login" />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/button1"
            android:layout_width="121dp"
            android:layout_height="wrap_content"
            android:text="Low" />

        <Button
            android:id="@+id/button2"
            android:layout_width="121dp"
            android:layout_height="wrap_content"
            android:text="High" />

    </LinearLayout>

</LinearLayout>
```

เอกสารนี้เป็นเอกสาร <Button ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
android:layout_width="121dp"
```

```
android:layout_height="wrap_content"
android:text="Off" />
```

```
<Button
android:id="@+id/button3"
android:layout_width="121dp"
android:layout_height="wrap_content"
android:text="High" />
```

```
</LinearLayout>
```

```
<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="Light"
android:textSize="@dimen/Login" />
```

```
<LinearLayout
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="horizontal" >
```

```
<ToggleButton
android:id="@+id/toggleButton3"
android:layout_width="121dp"
android:layout_height="wrap_content"
android:layout_weight="0.11"
android:text="ToggleButton"
android:textOff="Light1 ON"
android:textOn="Light1 OFF" />
```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อย่างไรก็ตามขอสงวนสิทธิ์ในข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
android:id="@+id/toggleButton4"
android:layout_width="121dp"
```

```
android:layout_height="wrap_content"
android:layout_weight="0.11"
android:text="ToggleButton"
android:textOff="Light2 ON"
android:textOn="Light2 OFF" />
```

```
</LinearLayout>
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Check Status"
    android:textSize="@dimen/Login" >
```

```
</TextView>
```

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
```

```
<Button
    android:id="@+id/button4"
    android:layout_width="121dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_weight="0.03"
    android:text="Monitor" />
```

```
<Button
    android:id="@+id/button5"
    android:layout_width="121dp"
```

```
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_weight="0.03"
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นกรณีศึกษาวิจัยเฉพาะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
android:text="Check" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

ระบบควบคุมระบบไฟฟ้าด้วยสมองกลฝังตัวผ่าน Smart Phone

1.เปิด Android Application

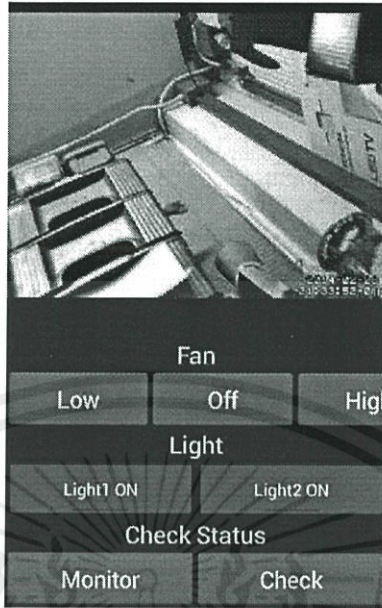
เมื่อผู้ใช้ทำการเปิดหน้า Android Application ขึ้นมา จะปรากฏหน้า Interface สำหรับกรอกค่าต่างๆ เช่น IP Address ของตัว Raspberry Pi และค่าต่างๆ สำหรับการตั้งค่ากล้อง Webcam ที่ต่ออยู่กับตัว Raspberry Pi เช่น ความกว้าง ความยาว ของหน้าต่างกล้อง Webcam หรือแม้กระทั่ง Port ของกล้อง Webcam ที่ใช้ต่ออยู่กับตัว Raspberry Pi ดังรูปที่ 1



รูปที่ 1 หน้าจอ Interface เมื่อเปิด Android Application ขึ้นมา

เมื่อผู้ใช้ทำการกรอกค่าต่างๆ เสร็จเรียบร้อยแล้ว ให้กดที่ปุ่ม Start Button เพื่อเข้าสู่หน้า Interface สำหรับควบคุมอุปกรณ์ไฟฟ้าผ่านทาง Android Application ดังรูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



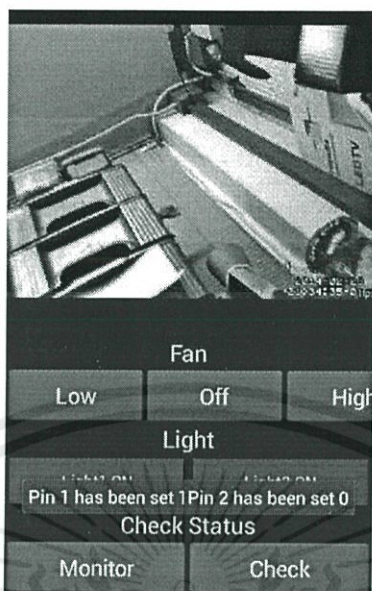
รูปที่ 2 หน้าจอ Interface สำหรับควบคุมอุปกรณ์ไฟฟ้า

จากรูปที่ 2 คือหน้า Interface สำหรับควบคุมอุปกรณ์ไฟฟ้าผ่าน Android Application ในหน้า Interface นี้ ยังสามารถแสดงภาพที่ได้จากกล้อง Webcam ที่ต่ออยู่กับ Raspberry Pi และมีปุ่มสำหรับควบคุมอุปกรณ์ไฟฟ้าชนิดต่างๆ ในช่องทางต่างๆกัน

2.ทดลองกดปุ่มต่างๆเพื่อควบคุม

ผู้ใช้สามารถทดลองกดปุ่มที่ช่องทางต่างๆ เพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดต่างๆ ได้ ยกตัวอย่างเช่น เมื่อผู้ใช้ต้องการควบคุมอุปกรณ์ไฟฟ้าชนิด “พัดลม ในระดับ Low” ให้ผู้ใช้กดปุ่ม Fan Low เมื่อผู้ใช้ทำการกดปุ่ม Fan Low แล้ว ก็จะมีข้อความปรากฏขึ้นมาว่า “Pin 1 has been set 1 Pin 2 has been set 0” นั้นหมายความว่า ผู้ใช้ได้ทำการกดปุ่ม Fan Low ไปแล้ว เป็นต้น ดังรูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 ปรากฏข้อความขึ้น เมื่อกดปุ่ม Fan Low

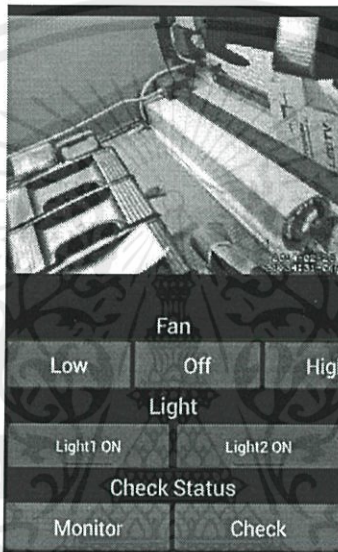
ในกรณีที่มีผู้ใช้หลายคนที่ต้องการควบคุมอุปกรณ์ไฟฟ้าผ่าน Android Application นี้ ในช่วงเวลาเดียวกัน เมื่อผู้ใช้คนที่ทำการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดหนึ่งไปแล้ว ผู้ใช้อีกคนหนึ่ง หรือคนอื่นๆ (ในกรณีที่มีผู้ใช้มากกว่า 2 คน) ก็จะสามารถกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดอื่นๆ ที่เหลืออยู่ได้

แต่ในกรณีที่มีผู้ใช้หลายคน(มากกว่า 1 คน) กดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดเดียวกัน ในช่วงเวลาที่ใกล้เคียงกัน ผล หรือสถานะของอุปกรณ์ไฟฟ้าจะเป็นไปตามที่ผู้ใช้คนสุดท้ายทำการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ตรวจสอบสถานะของอุปกรณ์ไฟฟ้า

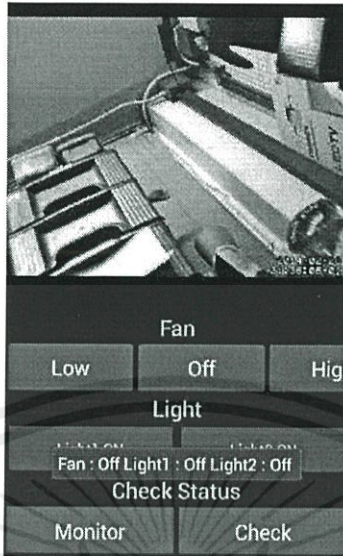
นอกจากนี้ ในกรณีที่ผู้ใช้ไม่ทราบว่าผู้ใช้อื่นได้ทำการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดไหน หรือช่องทางใดไปแล้วบ้าง หรือยังไม่ทราบสถานะล่าสุดของอุปกรณ์ไฟฟ้าที่ถูกควบคุม ผู้ใช้สามารถเข้าไปดูสถานะล่าสุดของอุปกรณ์ไฟฟ้าแต่ละชนิด แต่ละช่องทางได้ โดยกดที่ปุ่ม Monitor เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า ดังรูปที่ 4



รูปที่ 4 monitor button

จากรูปที่ 4 เมื่อผู้ใช้ทำการกดปุ่ม Monitor แล้ว จะปรากฏหน้าจอสำหรับแสดงสถานะของอุปกรณ์ไฟฟ้าล่าสุด เพื่อตรวจสอบสถานะล่าสุดของอุปกรณ์ไฟฟ้าที่กำลังทำงานอยู่ได้ ดังรูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 สถานะของอุปกรณ์ไฟฟ้าเมื่อกดปุ่ม Monitor

4. ตรวจสอบสถานะล่าสุดที่ได้กดปุ่มเพื่อควบคุม

และถ้าผู้ใช้ต้องการตรวจสอบสถานะล่าสุด ที่ผู้ใช้ได้ทำการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าชนิดต่างๆ ผู้ใช้สามารถตรวจสอบโดยการกดปุ่ม Check เพื่อตรวจสอบได้ ดังรูปที่ 6



รูปที่ 6 Check button

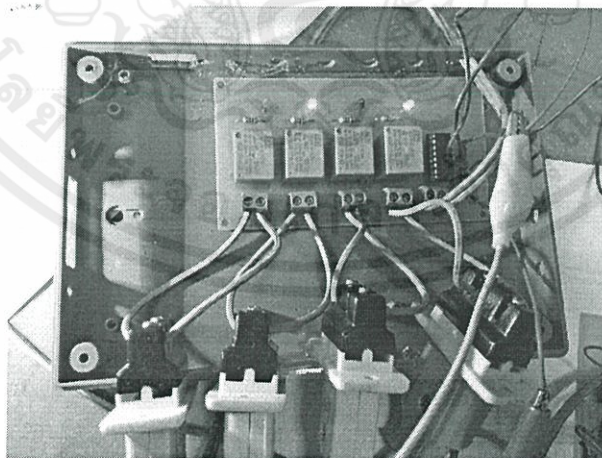
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ได้ทำการกดปุ่ม Check เพื่อตรวจสอบสถานะล่าสุดแล้ว จะมีข้อความแสดงสถานะต่างๆล่าสุด ที่ผู้ใช้ได้ทำการกดปุ่มเพื่อควบคุม ดังรูปที่ 7



รูปที่ 7 สถานะล่าสุดที่ทำการกดปุ่มเพื่อควบคุม

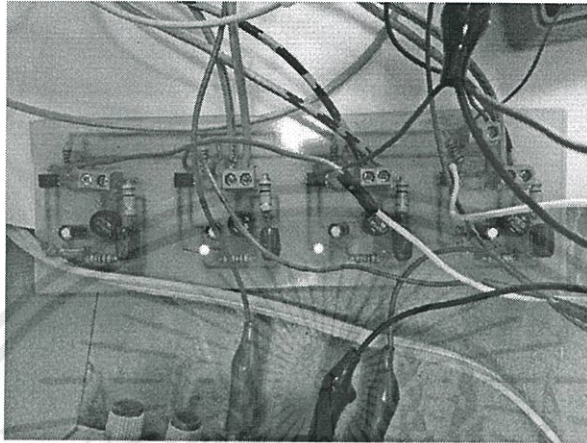
ทั้งหมดนี้ ผู้ใช้สามารถตรวจสอบสถานะของการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าจากสถานะของหลอด LED ในแต่ละช่องทางได้ ยกตัวอย่างเช่น เมื่อผู้ใช้ทำการกดปุ่มเพื่อควบคุมอุปกรณ์ไฟฟ้าช่องทางที่ 1 สถานะของหลอด LED ช่องทางที่ 1 ก็จะติด ดังรูปที่ 8



รูปที่ 8 สถานะของ LED เมื่อกดปุ่มควบคุมช่องทางต่างๆ

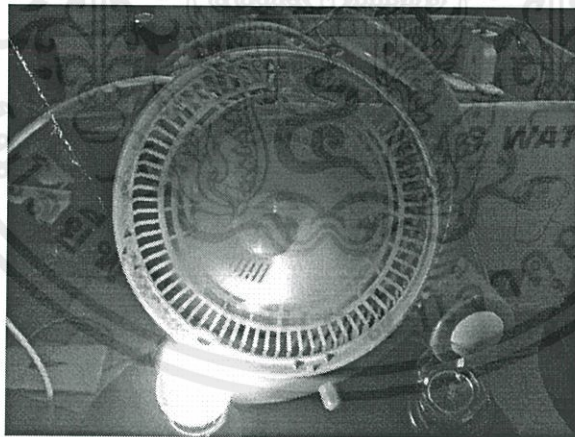
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษามาก่อนนั้น เป็นอนุสัญญาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อผู้ใช้ต้องการตรวจสอบสถานะของอุปกรณ์ไฟฟ้าที่กำลังทำงานอยู่ ผู้ใช้สามารถตรวจสอบได้จากสถานะของหลอด LED ได้เช่นกัน ในส่วนของการตรวจสอบสถานะอุปกรณ์ไฟฟ้า ดังรูปที่ 9



รูปที่ 9 LED แสดงสถานะของอุปกรณ์ไฟฟ้าว่าทำงานจริงหรือไม่

ทั้งหมดนี้ จะสังเกตได้ว่าอุปกรณ์ไฟฟ้านั้นสามารถทำงานได้ตามที่ผู้ใช้ได้ทำการกดปุ่มเพื่อควบคุมจาก Android Application บน Smartphone ในขั้นตอนที่ 2. ดังรูปที่ 10



รูปที่ 10 อุปกรณ์ไฟฟ้าสามารถทำงานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้