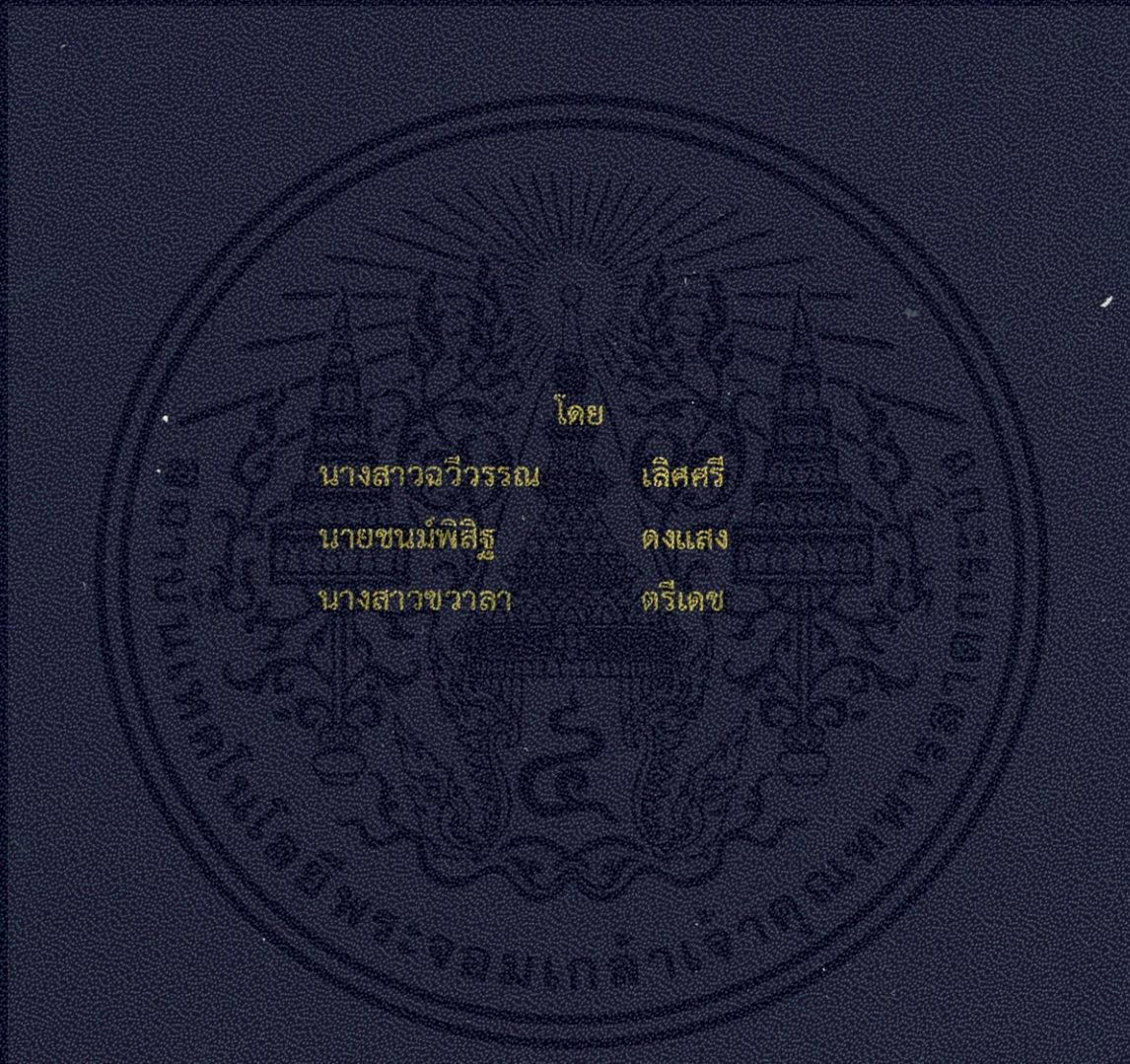


รถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย SMARTPHONE
SURVEILLANCE VEHICLE BASED ON EMBEDDED
SYSTEM CONTROLLED BY SMARTPHONE



โดย

นางสาวฉวีวรรณ

เลิศศรี

นายชนม์พิสิฐ

ดงแสง

นางสาวชวลา

ตรีเดช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

รถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย SMARTPHONE
SURVEILLANCE VEHICLE BASED ON EMBEDDED
SYSTEM CONTROLLED BY SMARTPHONE



โดย
นางสาวฉวีวรรณ เลิศศรี
นายชนม์พิสิฐ ดงแสง
นางสาวชวาลา ตรีเดช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย SMARTPHONE
SURVEILLANCE VEHICLE BASED ON EMBEDDED
SYSTEM CONTROLLED BY SMARTPHONE



โดย

นางสาวฉวีวรรณ	เลิศศรี	53010261
นายชนม์พิสิฐ	ดงแสง	53010293
นางสาวชวาลา	ตรีเดช	53010331

อาจารย์ที่ปรึกษา
รศ.ดร.พิพัฒน์ พรหมมี

อาจารย์ที่ปรึกษาร่วม
ผศ.ดร.พิเชฐ ม่วงนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผ่านการตรวจรูปเล่มแล้ว

(.....)
19 มิถ 57

อาจารย์ที่ปรึกษา

19 มิถ 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปีการศึกษา 2556 นั้น ไม่เอาหน้าไปใช้ประโยชน์ด้านการค้า

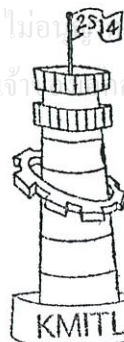
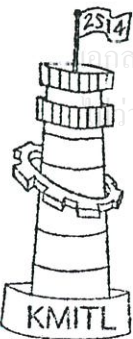
ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

14 มิถ 57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย SMARTPHONE

SURVEILLANCE VEHICLE BASED ON EMBEDDED SYSTEM CONTROLLED BY
SMARTPHONE

ผู้จัดทำ

- | | |
|--------------------------|----------|
| 1. นางสาวฉวีวรรณ เลิศศรี | 53010261 |
| 2. นายชนม์พิสิฐ ดงแสง | 53010293 |
| 3. นายสาวชวลา ตรีเดช | 53010331 |


.....
(รศ.ดร.พิพัฒน์ พรหมมี)

อาจารย์ที่ปรึกษา

.....
(ผศ.ดร.พิเชฐ ม่วงนวล)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ สำเร็จได้ด้วยความกรุณาจาก รศ.ดร.พิพัฒน์ พรหมมี และ ผศ.ดร.พิเชฐ ม่วงนวล อาจารย์ที่ปรึกษาปริญญาบัตร ที่ให้คำปรึกษา คำแนะนำที่เป็นประโยชน์ต่อ ปริญญาบัตรนี้และแนวทางการแก้ไขเมื่อเกิดปัญหาต่างๆ รวมทั้งสนับสนุนเครื่องมือและอุปกรณ์ ต่างๆ ที่จำเป็นต้องใช้ในการดำเนินปริญญาบัตร ผู้จัดทำขอขอบพระคุณท่านทั้งสองเป็นอย่างสูง

นอกจากนี้ยังได้รับความช่วยเหลือจากพี่ๆ ปริญญาโท ที่ให้คำแนะนำและให้ความรู้ เพิ่มเติมเกี่ยวกับปริญญาบัตรนี้ และเพื่อนๆ ที่คอยช่วยเหลือและให้กำลังใจในทุกๆ ด้าน ผู้จัดทำ ขอขอบคุณไว้ ณ โอกาสนี้ด้วย

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดาและครอบครัวผู้เป็นกำลังใจสำคัญ และให้การสนับสนุนช่วยเหลือมาโดยตลอด ผู้จัดทำหวังว่าปริญญาบัตรฉบับนี้จะอำนวยประโยชน์ และสามารถนำไปประยุกต์ใช้สืบต่อไปได้

นางสาวฉวีวรรณ เลิศศรี
นายชนม์พิสิฐ ดงแสง
นางสาวชวลา ตรีเดช
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย
SMARTPHONE
SURVEILLANCE VEHICLE BASED ON EMBEDDED
SYSTEM CONTROLLED BY SMARTPHONE

โดย	นางสาวฉวีวรรณ เลิศศรี	53010261
	นายชนม์พิสิฐ ดงแสง	53010293
	นางสาวชวาลา ตรีเดช	53010331

อาจารย์ที่ปรึกษา รศ.ดร.พิพัฒน์ พรหมมี
อาจารย์ที่ปรึกษาร่วม ผศ.ดร.พิเชฐ ม่วงนวล

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบและสร้างรถตรวจการณ์โดยใช้สมองกลฝังตัวควบคุมด้วย Smartphone มีวัตถุประสงค์เพื่อใช้สำรวจในบริเวณที่มีความเสี่ยงสูงต่อการเกิดอันตราย โดยแบ่งการทำงานออกเป็น 2 ส่วนคือ ส่วนของฮาร์ดแวร์ จะนำสมองกลฝังตัวมาเป็นตัวกลางในการควบคุมวงจรรีเลย์สวิตซ์ให้รถสามารถเคลื่อนที่ได้ รวมทั้งติดตั้งกล้องเพื่อให้รถสามารถส่งภาพผ่านเครือข่ายอินเทอร์เน็ตได้ และส่วนของซอฟต์แวร์ จะทำการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ สำหรับควบคุมทิศทางการเคลื่อนที่ของรถและสามารถแสดงภาพเหตุการณ์จริงบนหน้าแอปพลิเคชัน ซึ่งเป็นการช่วยลดจำนวนของผู้ประสบภัยจากอุบัติเหตุต่างๆ

ABSTRACT

This project to design and fabrication of Surveillance Vehicle based on Embedded System controlled by Smartphone. The purpose of this project is to survey in risk area. The section are divided into hardware and software. For hardware, the embedded in controller of relay circuit in order to drive the vehicle and also install the camera to capture and send data through internet network. For software, application is programmed on Android Operating System to control direction of vehicle and display the situation on application screen in order to decrease the victim from accident.

เอกสารนี้เป็นเอกสารผลงานวิจัยที่ทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกสิ่งนี้ออก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	IX
สารบัญตาราง	XIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 Raspberry Pi	2
2.1.1 ส่วนประกอบของบอร์ด Raspberry Pi	2
2.1.2 คุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi	3
2.1.3 การเริ่มต้นใช้งานบอร์ด Raspberry Pi	3
2.1.4 การจัดขาของ GPIO	4
2.2 ภาษา Python	5
2.2.1 ประวัติภาษา Python	5
2.2.2 หลักการทำงานของภาษา Python	5
2.2.2.1 คอมไพเลอร์ (Compiler)	5
2.2.2.2 อินเทอร์พรีเตอร์ (Interpreter)	6
2.3 ภาษา PHP	6
2.3.1 ประวัติภาษา PHP	7
2.3.2 คุณสมบัติของภาษา PHP	7
2.4 เว็บเซิร์ฟเวอร์ (Web Server)	7
2.4.1 หลักการทำงานของเว็บเซิร์ฟเวอร์	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับดูตัวอย่างเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5 ระบบจัดการฐานข้อมูล MySQL	8
2.5.1 สถาปัตยกรรมของ MySQL	8
2.5.1.1 ส่วนของผู้ให้บริการ (Server)	8
2.5.1.2 ส่วนของผู้ใช้บริการ (Client)	9
2.5.2 การใช้งาน MySQL	9
2.5.3 โปรแกรมช่วยในการจัดการฐานข้อมูล phpMyAdmin	9
2.6 ระบบปฏิบัติการแอนดรอยด์ (Android Operating System)	10
2.6.1 ลักษณะเด่นของแอนดรอยด์ (Android Feature)	10
2.6.2 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)	11
2.6.2.1 Applications	12
2.6.2.2 Application Framework	12
2.6.2.3 Libraries	13
2.6.2.4 Linux Kernel	13
2.6.3 ส่วนประกอบของแอปพลิเคชัน (Application Component)	14
2.6.3.1 Activity	14
2.6.3.2 Service	14
2.6.3.3 Broadcast Receiver	14
2.6.3.4 Content Provider	14
2.7 อีคลิปส์ (Eclipse)	15
2.7.1 แพลตฟอร์มของอีคลิปส์ (Eclipse Platform)	15
2.7.2 Android SDK (Android Software Development Kit)	15
2.8 ภาษา JAVA	15
2.8.1 ประวัติภาษา JAVA	16
2.8.2 การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)	16
2.8.3 คุณลักษณะเด่นของภาษา JAVA	16

สารบัญ (ต่อ)

	หน้า
2.8.4 รูปแบบของภาษา JAVA	16
2.8.5 ซอร์สโค้ด (Source Code)	17
2.9 รีเลย์ (Relay)	17
2.9.1 โครงสร้างของรีเลย์	17
2.9.2 หลักการทำงานของรีเลย์	18
2.9.3 ส่วนประกอบของรีเลย์	18
2.10 Analog to Digital Converter	19
2.10.1 สัญญาณอนาล็อก (Analog Signal)	19
2.10.2 สัญญาณดิจิทัล (Digital Signal)	19
2.10.3 Digital Ramp ADC	19
2.11 ไมโครคอนโทรลเลอร์ Arduino	21
2.11.1 หลักการทำงานของ Arduino	21
2.11.2 โปรแกรม Arduino IDE	22
2.12 Accelerometer Sensor	23
2.12.1 หลักการทำงานของ Accelerometer Sensor	24
2.12.2 คุณสมบัติของ Accelerometer Sensor ADXL335	25
2.13 Wi-Fi	25
2.13.1 มาตรฐาน Wi-Fi	25
2.13.2 ลักษณะการเชื่อมต่อของอุปกรณ์	26
2.13.2.1 โหมด Ad-Hoc หรือ Peer-to-Peer	26
2.14 กล้องเว็บแคม (Webcam)	26
บทที่ 3 การออกแบบและการจัดทำปริญญาานิพนธ์	27
3.1 หลักการทำงานของระบบ	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2 โพล์วชาร์ตการทำงานของระบบ	28
3.3 การออกแบบและสร้างเซิร์ฟเวอร์บนสมองกลฝังตัว (Raspberry Pi)	31
3.4 การออกแบบส่วนติดต่อกับผู้ใช้ (Graphic User Interface)	32
แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์	
3.4.1 ออกแบบและสร้างระบบสมัครสมาชิกและระบบล็อกอินบน	32
แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์	
3.4.1.1 เขียนโปรแกรมกำหนดการทำงานทางฝั่ง Server	32
3.4.1.2 เขียนโปรแกรมกำหนดการทำงานทางฝั่ง Client	33
3.4.2 ออกแบบและสร้างหน้าแอปพลิเคชันสำหรับควบคุมการเคลื่อนที่	34
ของรถและแสดงภาพจากกล้อง	
3.5 การออกแบบและสร้างวงจรรีเลย์สวิตช์	36
3.6 การออกแบบและสร้างวงจ Analog to Digital Converter	36
3.7 ส่วนประกอบของรถตรวจการณ์	37
3.7.1 ส่วนควบคุมการเคลื่อนที่ของรถ	37
3.7.1.1 Raspberry Pi	37
3.7.1.2 วงจรรีเลย์สวิตช์	38
3.7.2 ส่วนของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล	38
3.7.2.1 วงจ Analog to Digital Converter	38
3.7.2.2 เซนเซอร์วัดค่าความเอียง (Accelerometer Sensor)	39
3.8 เครื่องมือที่ใช้ในการทดลอง	40
3.8.1 ฮาร์ดแวร์ (Hardware)	40
3.8.2 ซอฟต์แวร์ (Software)	40

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.9.1 การทดสอบการรับส่งข้อมูลระหว่าง Server และ Client โดย Wireshark	41
3.9.2 การทดสอบความแรงของสัญญาณ Wi-Fi โดยใช้ Wi-Fi Analyzer	41
3.9.3 การทดลองควบคุมการเคลื่อนที่ของรถผ่านสมาร์ทโฟนบน ระบบปฏิบัติการแอนดรอยด์	41
3.9.4 การทดลองการส่งค่าความเอียงของ Accelerometer Sensor จากวงจร Analog to Digital Converter ไปแสดงผลบน แอปพลิเคชันแอนดรอยด์	42
บทที่ 4 ผลการทดลอง	43
4.1 ผลการทดสอบการรับส่งข้อมูลระหว่าง Server และ Client โดย Wireshark	43
4.2 ผลการทดสอบความแรงของสัญญาณ Wi-Fi โดยใช้ Wi-Fi Analyzer	46
4.3 ผลการทดลองควบคุมการเคลื่อนที่ของรถผ่านสมาร์ทโฟนบนระบบ ปฏิบัติการแอนดรอยด์	47
4.3.1 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเคลื่อนที่ไปข้างหน้า	52
4.3.2 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเคลื่อนที่ไปข้างหลัง	56
4.3.3 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเลี้ยวซ้าย	60
4.3.4 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเลี้ยวขวา	64
4.4 ผลการทดลองการส่งค่าความเอียงของ Accelerometer Sensor จากวงจร Analog to Digital Converter ไปแสดงผลบนแอปพลิเคชัน แอนดรอยด์	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะ	72
5.1 สรุปผล	72
5.2 ข้อเสนอแนะ	72
บรรณานุกรม	73
ภาคผนวก ก Datasheet	74
ภาคผนวก ข Source Code	91



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของบอร์ด Raspberry Pi	2
2.2 การจัดขาของ GPIO แบบมาตรฐาน	4
2.3 การจัดขาของ GPIO แบบใช้โปรแกรม Wiring Pi	4
2.4 การทำงานของคอมไพเลอร์	6
2.5 การทำงานของอินเทอร์พรีเตอร์	6
2.6 การทำงานของเว็บเซิร์ฟเวอร์	8
2.7 หน้าจอของโปรแกรม phpMyAdmin	10
2.8 สถาปัตยกรรมแอนดรอยด์	11
2.9 รีเลย์	17
2.10 สัญลักษณ์ของรีเลย์	17
2.11 โครงสร้างของรีเลย์	18
2.12 สภาพะการทำงานของรีเลย์	18
2.13 การทำงานของ Digital Ramp ADC	20
2.14 output voltage ของ Digital Ramp ADC	21
2.15 โปรแกรม Arduino IDE	22
2.16 ระบายที่ใช้วัดค่าความเร่งในการเอียง	23
2.17 โครงสร้างภายในของ Accelerometer Sensor	24
2.18 Accelerometer Sensor ADXL335	24
2.19 กล้องเว็บแคม	26
3.1 บล็อกไดอะแกรมโดยรวมของระบบ	27
3.2 โฟล์วชาร์ตการทำงานของระบบทั้งหมด	28
3.3 โฟล์วชาร์ตการทำงานในส่วนผู้ใช้แอปพลิเคชันบนแอนดรอยด์	29
3.4 โฟล์วชาร์ตการทำงานของสมองกลฝังตัว (Raspberry Pi)	30
3.5 ฐานข้อมูลบนเซิร์ฟเวอร์ในสมองกลฝังตัว (Raspberry Pi)	31
3.6 หน้าจอหลักของแอปพลิเคชัน	32
3.7 หน้าจอสำหรับป้อนข้อมูลสมัครสมาชิก	33
3.8 หน้าจอสำหรับล็อกอิน	34

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.9 หน้าจอสำหรับป้อนไอพีแอดเดรสและพอร์ตของสมองกลฝังตัว	35
3.10 หน้าจอสำหรับควบคุมการเคลื่อนที่ของรถและแสดงภาพจากกล้อง	35
3.11 วงจรรีเลย์สวิตช์ที่ออกแบบโดยใช้โปรแกรม Proteus 7 Professional	36
3.12 วงจร Analog to Digital Converter ที่ออกแบบโดยใช้โปรแกรม Proteus 7 Professional	37
3.13 Raspberry Pi	38
3.14 วงจรรีเลย์สวิตช์	38
3.15 วงจร Analog to Digital Converter	39
3.16 Accelerometer Sensor ADXL335	39
3.17 รถตรวจการณ์	40
4.1 IP Address ของ Server (สมองกลฝังตัว)	43
4.2 IP Address ของ Client (แอปพลิเคชันแอนดรอยด์)	43
4.3 การเชื่อมต่อระหว่าง Server และ Client โดยการ Ping	44
4.4 ตรวจสอบการรับส่งข้อมูลระหว่าง Server และ Client โดยใช้ Wireshark	44
4.5 รายละเอียดของแต่ละ Packet ที่ได้จากการดักจับจาก Wireshark	45
4.6 เมื่ออยู่ใกล้เครื่องปล่อยสัญญาณ Wi-Fi	46
4.7 เมื่ออยู่ไกลเครื่องปล่อยสัญญาณ Wi-Fi	46
4.8 เมื่อรถยนต์ยังไม่มีเคลื่อนที่	47
4.9 LED ที่ใช้แสดงสถานะของแต่ละพอร์ต	47
4.10 ฐานข้อมูลบน Server (สมองกลฝังตัว)	48
4.11 หน้าจอ LXTerminal แสดงสถานะของพอร์ต GPIO	48
4.12 แอปพลิเคชันแอนดรอยด์บนสมาร์ทโฟนและหน้าจอหลักของแอปพลิเคชัน	49
4.13 หน้าจอสำหรับล็อกอินและหน้าจอสำหรับป้อนข้อมูลสมัครสมาชิก	49
4.14 ป้อนข้อมูลสมัครสมาชิก	50
4.15 ข้อมูลสมัครสมาชิกที่บันทึกในฐานข้อมูลบนสมองกลฝังตัว	50
4.16 การล็อกอินเข้าสู่หน้าจอสำหรับการเชื่อมต่อกับกล้องเว็บแคม	51
4.17 ภาพจากกล้องเว็บแคมและปุ่มสำหรับควบคุมการเคลื่อนที่ของรถบนหน้า แอปพลิเคชันแอนดรอยด์	51
4.18 เมื่อทำการกดปุ่ม Forward บนหน้าแอปพลิเคชัน	52

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.19 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 18 และ 23 เป็น ลอจิก 1	53
4.20 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 18 และ 23	53
4.21 สถานะการทำงานของพอร์ต 18 และ 23	54
4.22 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17	54
4.23 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18	54
4.24 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22	55
4.25 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23	55
4.26 รถกำลังเคลื่อนที่ไปข้างหน้า	55
4.27 เมื่อทำการกดปุ่ม Backward บนหน้าแอปพลิเคชัน	56
4.28 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 17 และ 22 เป็น ลอจิก 1	57
4.29 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 17 และ 22	57
4.30 สถานะการทำงานของพอร์ต 17 และ 22	58
4.31 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17	58
4.32 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18	58
4.33 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22	59
4.34 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23	59
4.35 รถกำลังเคลื่อนที่ไปข้างหลัง	59
4.36 เมื่อทำการกดปุ่ม Left บนหน้าแอปพลิเคชัน	60
4.37 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 23 เป็น ลอจิก 1	61
4.38 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 23	61
4.39 สถานะการทำงานของพอร์ต 23	62
4.40 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17	62
4.41 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18	62
4.42 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22	63
4.43 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23	63
4.44 รถกำลังเลี้ยวซ้าย	63
4.45 เมื่อทำการกดปุ่ม Right บนหน้าแอปพลิเคชัน	64

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.46	ค่าที่รับเข้ามาจากแอฟฟลิเคชันแสดงสถานะของพอร์ต 18 เป็น ลอจิก 1	65
4.47	หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 18	65
4.48	สถานะการทำงานของพอร์ต 18	66
4.49	สัญญาณเอาต์พุตที่ออกจากพอร์ต 17	66
4.50	สัญญาณเอาต์พุตที่ออกจากพอร์ต 18	66
4.51	สัญญาณเอาต์พุตที่ออกจากพอร์ต 22	67
4.52	สัญญาณเอาต์พุตที่ออกจากพอร์ต 23	67
4.53	รถกำลังเลี้ยวขวา	67
4.54	พิกัดอ้างอิง	68
4.55	การเอียงในแนวแกนปกติและค่าความเอียงที่แสดงบนแอฟฟลิเคชัน	68
4.56	การเอียงในแนวแกน +X และค่าความเอียงที่แสดงบนแอฟฟลิเคชัน	69
4.57	การเอียงในแนวแกน - X และค่าความเอียงที่แสดงบนแอฟฟลิเคชัน	69
4.58	การเอียงในแนวแกน +Y และค่าความเอียงที่แสดงบนแอฟฟลิเคชัน	70
4.59	การเอียงในแนวแกน - Y และค่าความเอียงที่แสดงบนแอฟฟลิเคชัน	70
4.60	หน้าจอแอฟฟลิเคชัน	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	การอธิบายส่วนประกอบของ Application Framework	12
2.2	การอธิบายส่วนประกอบของ Libraries	13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน เทคโนโลยีการสื่อสารและอินเทอร์เน็ตได้มีการพัฒนาและมีความก้าวหน้าอย่างต่อเนื่อง เพื่ออำนวยความสะดวกอีกทั้งยังช่วยทุ่นแรงของมนุษย์ ทุกวันนี้คงจะปฏิเสธไม่ได้ว่า โทรศัพท์มือถือเข้ามามีบทบาทอย่างมากในชีวิตประจำวัน โดยเฉพาะโทรศัพท์สมาร์ทโฟน (Smartphone) ที่กำลังได้รับความนิยมอย่างมากและก็มีการพัฒนาอย่างต่อเนื่อง ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) บนโทรศัพท์สมาร์ทโฟน ได้รับการพัฒนาให้มีคุณสมบัติในการใช้งานที่หลากหลายและสะดวกสบายเพิ่มมากขึ้น โดยเฉพาะการใช้งาน Mobile internet ที่สามารถใช้งานได้ทุกที่ทุกเวลาและ Raspberry Pi เป็นอุปกรณ์หนึ่งที่กำลังได้รับความสนใจจากนักพัฒนาซอฟต์แวร์ระบบฝังตัว (Embedded System) เนื่องจากใช้พื้นที่ติดตั้งน้อย กินไฟน้อย สามารถพัฒนาซอฟต์แวร์ได้เหมือนพัฒนาบนเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer : PC) จึงเหมาะสำหรับนำไปทำงานเฉพาะทางหลายๆ อย่างที่ไม่ได้ใช้งานหนักมากนัก จากลักษณะเด่นนี้ จึงทำให้มีความสนใจในการนำ Raspberry Pi มาประยุกต์ใช้งานในการสร้างอุปกรณ์ที่ช่วยอำนวยความสะดวกในการค้นหาผู้รอดชีวิตจากการพังทลายของสิ่งปลูกสร้างหรือแม้กระทั่งในสถานที่ที่มีความเสี่ยงสูงต่อการเกิดอันตรายเพื่อช่วยลดจำนวนของผู้ประสบภัยจากอุบัติเหตุต่างๆ ดังนั้น ปรียนุญานิพนธ์ฉบับนี้จึงได้นำเสนอรถตรวจการณ์ที่ใช้สมองกลฝังตัวควบคุมด้วย Smartphone

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาการเขียนโปรแกรมควบคุมสมองกลฝังตัว
- 2) เพื่อศึกษาการเขียนโปรแกรมบนเครือข่ายอินเทอร์เน็ต
- 3) เพื่อศึกษาการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์
- 4) เพื่อออกแบบและสร้างชุดควบคุมรถตรวจการณ์ให้สามารถขับเคลื่อนและส่งภาพได้

1.3 ขอบเขตของปรียนุญานิพนธ์

- 1) ศึกษาการเขียนโปรแกรมให้สมองกลฝังตัวทำหน้าที่เป็นเซิร์ฟเวอร์
- 2) ศึกษาการเขียนโปรแกรมบนเครือข่ายอินเทอร์เน็ต
- 3) ศึกษาการเขียนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสาร 4) ออกแบบและสร้างชุดควบคุมรถตรวจการณ์ให้สามารถขับเคลื่อนและส่งภาพได้ ชั้นด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

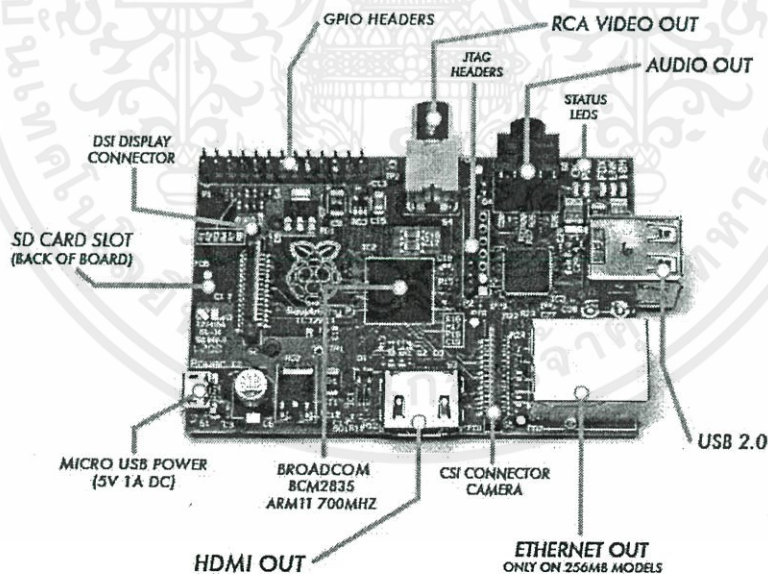
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry Pi

Raspberry Pi คือบอร์ดไมโครคอมพิวเตอร์แบบแผ่นเดียว ถูกพัฒนาขึ้นในประเทศสหรัฐอเมริกาโดย Raspberry Pi Foundation ภายใต้การทำงานขององค์กรไม่หวังผลกำไร สร้างขึ้นเพื่อรองรับระบบปฏิบัติการลินุกซ์ (Linux) ที่ถูกบรรจุลงใน SD การ์ด สำหรับการพัฒนาไปสู่บอร์ด Embedded Linux ซึ่งประกอบด้วยจุดเชื่อมต่ออุปกรณ์อินพุต เอาต์พุต ทั้งผ่านพอร์ต USB LAN HDMI ช่องสัญญาณภาพและ GPIO (General Purpose Input Output) สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์

2.1.1 ส่วนประกอบของบอร์ด Raspberry Pi

แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบของบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่มา : <http://www.pcmag.com/article2/0,2817,2407058,00.asp> ซึ่งประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 คุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

1. ชิพควบคุมหลัก (Broadcom BCM2835) หรือเทียบเท่า ซึ่งรวมซีพียู หน่วยประมวลผลกราฟิกหรือ GPU และหน่วยความจำ SDRAM ไว้ภายในตัวถังเดียวกัน
2. หน่วยประมวลผลกลางหรือ CPU: ARM11 Core ARM1176JZF-S ความเร็ว 700 MHz
3. หน่วยประมวลผลกราฟิกหรือ GPU: Broadcom Video Core IV หรือเทียบเท่า ใช้สำหรับรองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
4. หน่วยความจำ SDRAM: 512 MB
5. จุดต่อ: USB 2.0 (2 พอร์ต) แจ็ค RCA และ HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับโทรทัศน์หรือจอแสดงผล ที่มีจุดต่อแบบ RCA ตัวเมียหรือ HDMI จุดต่อเอาต์พุตเสียงเป็นแจ๊ค หูฟัง 3.5 มิลลิเมตร จุดต่อ internet หรือจุดต่อระบบ LAN connector หรือจุดต่อพอร์ตอินพุตเอาต์พุต (GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus) I2C I2S ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และ socket ของ SD การ์ด สำหรับเสียบ SD การ์ดที่ติดตั้งระบบปฏิบัติการเรียบร้อยแล้ว
6. ความต้องการไฟเลี้ยง: +5V 700 mA เป็นอย่างน้อย
7. ขนาด: 85.60 x 53.98 มิลลิเมตร หรือ 3.370 x 2.125 นิ้ว

2.1.3 การเริ่มต้นใช้งานบอร์ด Raspberry Pi

ต้องเตรียมอุปกรณ์ ดังนี้

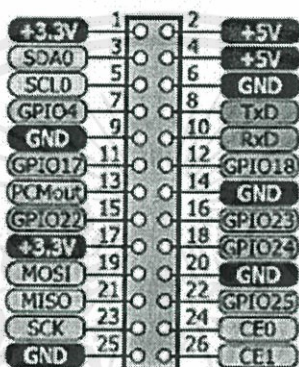
1. บอร์ด Raspberry Pi
2. SD การ์ดสำหรับเก็บข้อมูลรวมไปถึงระบบปฏิบัติการควรมีความจุตั้งแต่ 4GB คลาส 4 ขึ้นไปซึ่งก็คือ SDHC การ์ดจะเป็นแบบ FAT32 หรือ NTFS ก็ได้
3. สาย microUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V 700 mA ที่มีจุดต่อแบบ USB ไม่ควรใช้แหล่งจ่ายไฟจากพอร์ต USB ของคอมพิวเตอร์ เนื่องจากมีความสามารถในการจ่ายกระแสไฟฟ้าไม่เพียงพอ
4. สาย HDMI หรือ RCA สำหรับต่อกับจอแสดงผล
5. คีย์บอร์ดและเมาส์แบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi
6. สาย LAN สำหรับเชื่อมต่อกับอินเทอร์เน็ต
7. ลำโพงที่มีสายต่อเป็นปลั๊กหูฟัง 3.5 มม. เพื่อต่อกับแจ๊คเอาต์พุตสัญญาณเสียงของบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การจัดขาของ GPIO

บอร์ด Raspberry Pi มีขาอินพุต เอาต์พุต ดิจิทัลหรือ GPIO สำหรับติดต่อกับอุปกรณ์ภายนอก รวม 17 ขา โดยมีการจัดเรียงขามาตรฐานดังรูปที่ 2.2 และหากใช้โปรแกรม Wiring Pi จะมีการจัดเรียงขาอินพุต เอาต์พุตใหม่ดังรูปที่ 2.3

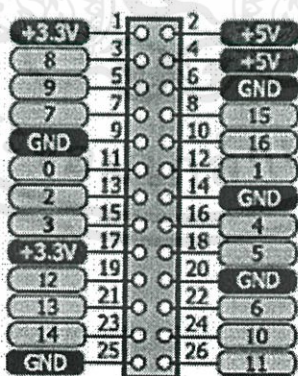
Raspberry Pi (Rev.2)
GPIO pinout
STANDARD



รูปที่ 2.2 การจัดขาของ GPIO แบบมาตรฐาน

ที่มา : www.inex.co.th/store/manual/R-Pi_IO.pdf

Raspberry Pi (Rev.2)
GPIO pinout
wiring-Pi



รูปที่ 2.3 การจัดขาของ GPIO แบบใช้โปรแกรม Wiring Pi

ที่มา : www.inex.co.th/store/manual/R-Pi_IO.pdf

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานภายในเท่านั้นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ภาษา Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งที่มีความสามารถสูง ซึ่งถูกพัฒนาขึ้นมา โดยไม่ยึดติดกับแพลตฟอร์ม ภาษา Python เป็น Open Source ทำให้ทุกคนสามารถที่จะนำมาพัฒนาโปรแกรมได้ฟรีๆ โดยไม่ต้องเสียค่าใช้จ่ายและความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ภาษา Python มีความสามารถสูงขึ้นและใช้งานได้ครอบคลุมกับทุกลักษณะงาน

2.2.1 ประวัติภาษา Python

ภาษา Python ถูกสร้างขึ้นมาจาก Guido Van Rossum ซึ่งโค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ คือจะประมวลผลไปที่ละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันแรกคือ เวอร์ชัน 0.9.0 ออกมาเมื่อปี พ.ศ. 2533 และเวอร์ชันปัจจุบันคือ 2.5.2 ออกเมื่อวันที่ 21 กุมภาพันธ์ พ.ศ. 2551 และได้มีการพัฒนา Python ในรุ่นที่ 3 คือ Python 3000 หรือ Py3k

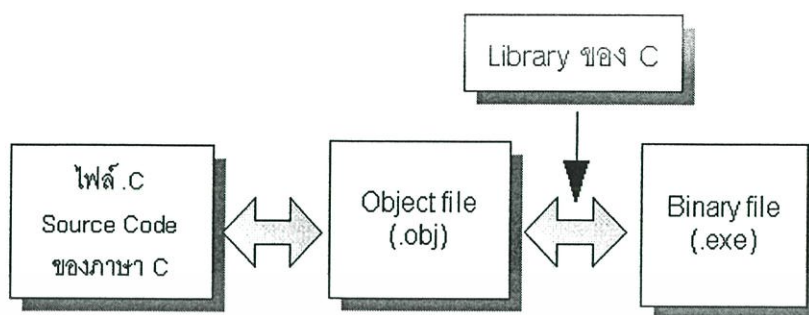
2.2.2 หลักการทำงานของภาษา Python

เมื่อเราได้เขียนโค้ดขึ้นมาตามโครงสร้างของโปรแกรมภาษาใดก็ตามและการจะให้โค้ดคำสั่งเหล่านั้นทำงานได้ก็จะต้องมีตัวแปลภาษามาจัดการแปลโค้ดคำสั่งเพื่อให้ทำงานตามที่เราต้องการ โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภทใหญ่ๆ คือ

2.2.2.1 คอมไพเลอร์ (Compiler)

เป็นตัวแปลภาษาสำหรับภาษา C ภาษา C++ และภาษา Pascal การทำงานก็คือ จะตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้งแต่ต้นจนจบก่อนหรือเรียกว่า การคอมไพล์ (Compile) ถ้าไม่มีข้อผิดพลาดก็จะทำการแปลโค้ดคำสั่งของเราให้เป็นไฟล์นามสกุล .obj (Object File) จากนั้นก็ทำการแปลงไฟล์ .obj ให้เป็นไบนารีไฟล์ .exe เพื่อทำงานต่อไป ตัวอย่างการทำงานของคอมไพเลอร์ภาษา C ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

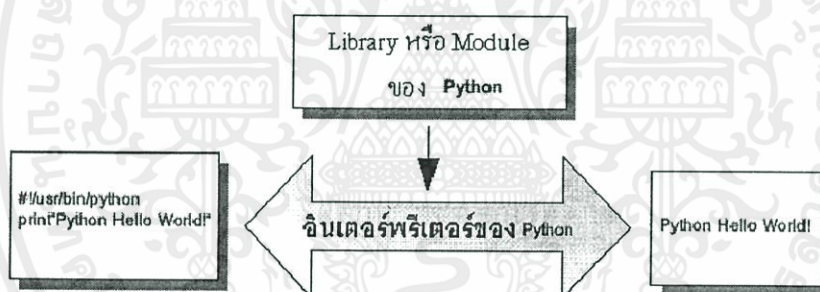


รูปที่ 2.4 การทำงานของคอมไพเลอร์

ที่มา : http://python.cmsthailand.com/basic_python.html

2.2.2.2 อินเทอร์พรีเตอร์ (Interpreter)

จะทำงานเป็นบรรทัดต่อบรรทัด คืออ่านโค้ดคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมาเลยดังรูปที่ 2.5



รูปที่ 2.5 การทำงานของอินเทอร์พรีเตอร์

ที่มา : http://python.cmsthailand.com/basic_python.html

จากรูปที่ 2.5 ในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารี (Library) หรือโมดูล (Module) ของภาษา Python อินเทอร์พรีเตอร์ของภาษา Python ก็จะไปทำการเรียกฟังก์ชันเหล่านั้นให้ทำงานแล้วจึงแสดงผลการทำงานออกมา

2.3 ภาษา PHP

PHP เป็นภาษาจำพวก Scripting Language คำสั่งต่างๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (Script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์ ก็เช่น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JavaScript Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า Server-side หรือ HTML-Embedded Scripting Language เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น อาจกล่าวได้ว่า PHP ได้รับการพัฒนาขึ้นมาเพื่อแทนที่ SSI (Server Side Include) รูปแบบเดิมๆ โดยให้มีความสามารถและมีส่วนเชื่อมต่อกับเครื่องมือชนิดอื่นมากขึ้น เช่น ติดต่อกับฐานข้อมูล (Database) เป็นต้น

2.3.1 ประวัติภาษา PHP

PHP ได้รับการเผยแพร่เป็นครั้งแรกในปี ค.ศ. 1994 จากนั้นก็มีการพัฒนาต่อมาตามลำดับ เป็นเวอร์ชัน 1 ในปี 1995 เวอร์ชัน 2 (ตอนนั้นใช้ชื่อว่า PHP/FI) ในช่วงระหว่าง 1995-1997 และเวอร์ชัน 3 ช่วง 1997 ถึง 1999 จนถึงเวอร์ชัน 4 ในปัจจุบัน

2.3.2 คุณสมบัติของภาษา PHP

การแสดงผลของภาษา PHP จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่ภาษา PHP แตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ (Client-Side Script Language) นอกจากนี้ PHP ยังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยากโดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต

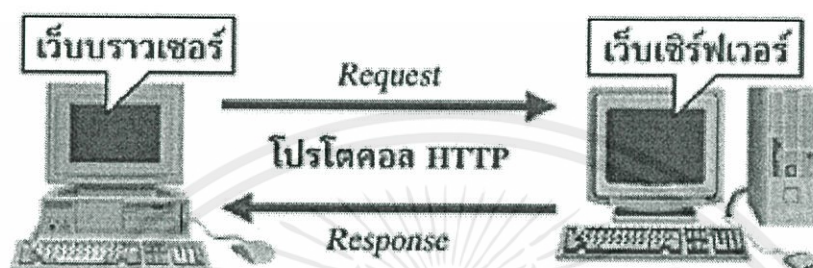
2.4 เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์คือโปรแกรมที่อยู่และทำงานบนเครื่องฝั่งเซิร์ฟเวอร์ ทำหน้าที่ในการรับคำสั่งจากการร้องขอของไคลเอนต์ (Client) และประมวลผลการทำงานจากการร้องขอดังกล่าว แล้วส่งข้อมูลกลับไปยังเครื่องของไคลเอนต์ สามารถเขียนได้หลายภาษา หลายรูปแบบ การเขียนภาษา PHP ระบบจะสามารถทำงานได้ก็จะต้องมีเว็บเซิร์ฟเวอร์ เป็นตัว Run Engine ของภาษานั้นๆ อีกทีหนึ่ง ดังนั้นถ้าต้องการให้คอมพิวเตอร์สามารถ Run ภาษา PHP ได้นั้น จะต้องจำลองเครื่องคอมพิวเตอร์ให้เป็นเซิร์ฟเวอร์ โดยใช้โปรแกรมเว็บเซิร์ฟเวอร์

2.4.1 หลักการทำงานของเว็บเซิร์ฟเวอร์

หลักการทำงานของเว็บเซิร์ฟเวอร์มีลักษณะเช่นเดียวกับบริการอื่นๆของอินเทอร์เน็ตคืออยู่ในรูปแบบไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) โดยมีโปรแกรมเว็บไคลเอนต์ (Web Client) ทำหน้าที่เป็นผู้ร้องขอบริการและมีโปรแกรมเว็บเซิร์ฟเวอร์ ทำหน้าที่เป็นผู้ให้บริการ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเว็บไคลเอนต์ก็คือโปรแกรมเว็บเบราว์เซอร์ (Web Browser) ในเครื่องของผู้ใช้นั่นเอง ซึ่งสำหรับโปรแกรมเว็บเซิร์ฟเวอร์นั้นจะถูกติดตั้งไว้ในเครื่องของผู้ให้บริการเว็บไซต์ การติดต่อระหว่างโปรแกรมเว็บเบราว์เซอร์กับโปรแกรมเว็บเซิร์ฟเวอร์จะกระทำผ่านโปรโตคอล HTTP (Hypertext Transfer Protocol) ดังรูปที่ 2.6



รูปที่ 2.6 การทำงานของเว็บเซิร์ฟเวอร์

ที่มา : <http://www.freebsd.sru.ac.th/course/4012305-Web-Programming/01-Parinya-Book/chapter-02.pdf>

2.5 ระบบจัดการฐานข้อมูล MySQL

MySQL เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL มีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรีและแบบที่ใช้ในเชิงธุรกิจ MySQL สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael “Monty” Widenius. ปัจจุบัน บริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.) เข้าซื้อกิจการของ MySQL AB เรียบร้อยแล้ว ฉะนั้น ผลิตภัณฑ์ภายใต้ MySQL AB ทั้งหมดจะตกเป็นของซัน ชื่อ “MySQL”

2.5.1 สถาปัตยกรรมของ MySQL

โครงสร้างการทำงานของ MySQL เป็นลักษณะการทำงานแบบ client/server ซึ่งประกอบด้วย 2 ส่วนหลักๆ คือ ส่วนของผู้ให้บริการ (Server) และส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนก็จะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน

2.5.1.1 ส่วนของผู้ให้บริการ (Server)

เป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูล ก็คือตัว MySQL Server และเป็นทีจัดเก็บข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 ส่วนของผู้ใช้บริการ (Client)

คือผู้ใช้นั้นเอง โปรแกรมใช้งานในส่วนนี้ได้แก่ MySQL Client Access Web development platform ต่างๆ เช่น Java Perl PHP ASP เป็นต้น

2.5.2 การใช้งาน MySQL

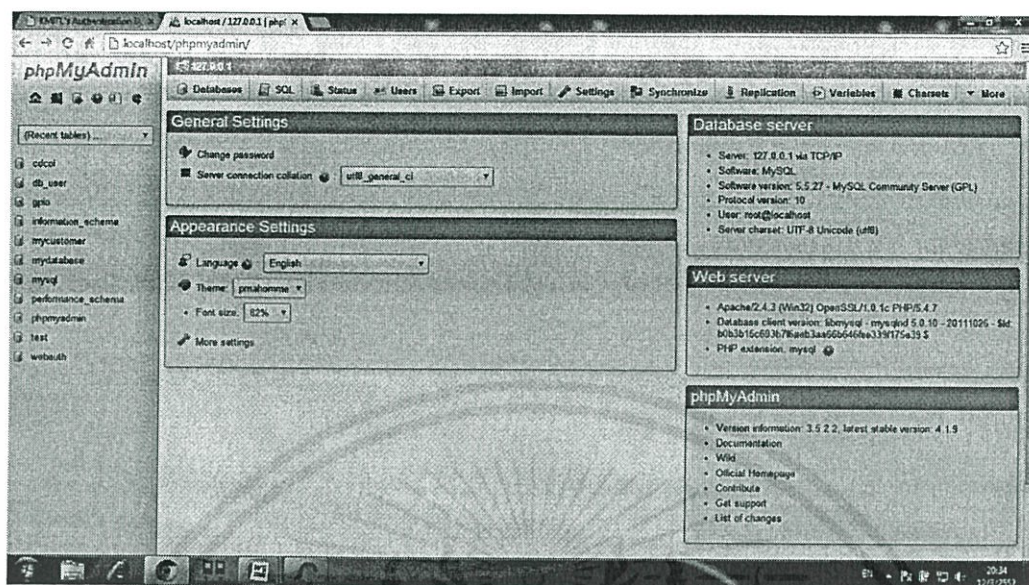
MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ phpBB นิยมใช้งานร่วมกับโปรแกรมภาษา PHP จะเห็นได้จากคู่มือคอมพิวเตอร์ต่างๆ ที่จะสอนการใช้งาน MySQL และ PHP ควบคู่กันไป นอกจากนี้หลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูล MySQL ซึ่งรวมถึงภาษา C C++ Pascal C# Java Perl Python Ruby และภาษาอื่นๆ ใช้งานผ่าน API (Application Programming Interface) สำหรับโปรแกรมที่ติดต่อผ่าน ODBC (Open Database Connector) หรือส่วนเชื่อมต่อกับภาษาอื่น (Database Connector)

2.5.3 โปรแกรมช่วยในการจัดการฐานข้อมูล phpMyAdmin

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ได้โดยตรงซึ่งเป็นโปรแกรมแบบ Open Source เพราะว่าเขียนด้วยภาษา PHP ทั้งหมด แสดงหน้าจอของโปรแกรมได้ดังรูปที่ 2.7 phpMyAdmin จะทำงานบน Web Server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ซึ่งความสามารถของ phpMyAdmin มีดังนี้

1. การสร้าง (Create) และลบ (Delete) ฐานข้อมูล
2. การสร้างและจัดการตาราง (Table) เช่น แทรก record ลบ record แก้ไข record หรือลบ Table เพิ่มหรือแก้ไข field ในตาราง
3. โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
4. ทหาผลสรุป (Query) ด้วยคำสั่ง SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 หน้าจอของโปรแกรม phpMyAdmin

2.6 ระบบปฏิบัติการแอนดรอยด์ (Android Operating System)

แอนดรอยด์ (Android) คือระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ต คอมพิวเตอร์ เน็ตบุ๊ก เป็นต้น การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลินุกซ์ เคอร์เนล (Linux Kernel) เริ่มพัฒนาโดยบริษัทแอนดรอยด์ (Android Inc.) จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล (Google) และนำแอนดรอยด์ไปพัฒนาต่อ กูเกิลได้ทำการเปิดตัวแอนดรอยด์ต่อสาธารณชนเมื่อวันที่ 5 พฤศจิกายน พ.ศ. 2550 โดยทางกูเกิลได้ประกาศก่อตั้ง Open Handset Alliance ของกลุ่มบริษัทฮาร์ดแวร์ ซอฟต์แวร์และการสื่อสาร 48 แห่งที่ร่วมมือกันเพื่อพัฒนาเป็นมาตรฐานเปิดสำหรับอุปกรณ์มือถือและทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขซอร์สโค้ด (Source Code) ต่างๆ ได้ด้วยภาษา Java และควบคุมอุปกรณ์ผ่านทางชุด Java Libraries ที่กูเกิลพัฒนาขึ้น ลิขสิทธิ์ของโค้ดแอนดรอยด์นี้จะใช้ในลักษณะของซอฟต์แวร์เสรี

2.6.1 ลักษณะเด่นของแอนดรอยด์ (Android Feature)

แอนดรอยด์เป็นระบบปฏิบัติการที่เปิดเผยแพร่ซอร์สโค้ดและไม่มีข้อกำหนดฮาร์ดแวร์ ซอฟต์แวร์ตายตัว แต่อย่างไรก็ตามแอนดรอยด์นั้นสนับสนุนลักษณะเด่น ดังนี้

1. การจัดเก็บข้อมูล (Storage) ใช้ SQLite ระบบฐานข้อมูลที่มีขนาดเล็ก เร็ว

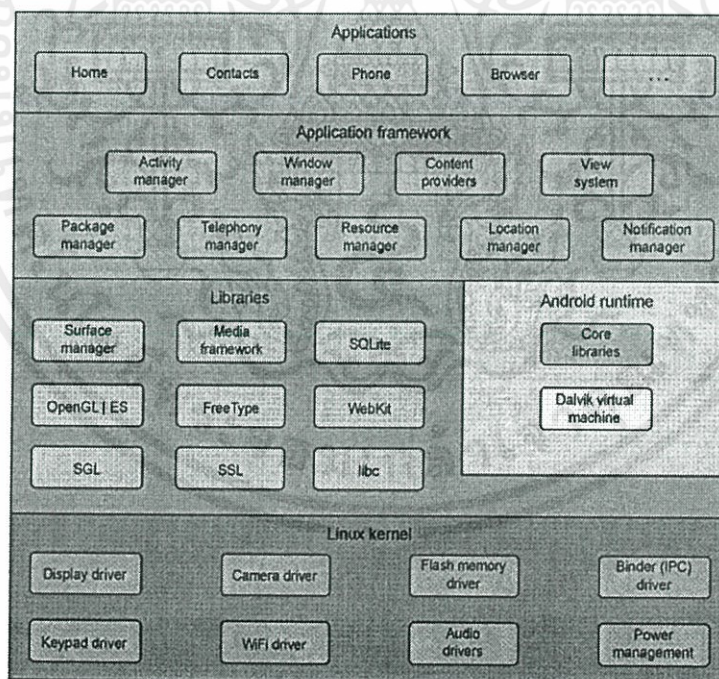
และมีเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเชื่อมโยง (Connectivity) สนับสนุน GSM/EDGE IDEN. CDMA EV-DO UMTS Bluetooth Wi-Fi LTE และ WiMAX
3. ข้อความ (Messaging) สนับสนุนทั้ง SMS และ MMS
4. Web Browser มีพื้นฐานอยู่บน WebKit
5. การสนับสนุนสื่อ (Media Support) H.263 H264 (3GP หรือ MP4) MPEG-4 SP AMR AMR-WB AAC HE-AAC MP3 MIDI WAV JPEG PNG GIF และ BMP
6. ฮาร์ดแวร์ Accelerometer Sensor Camera Digital Compass Proximity Sensor และ GPS
7. Multi-touch สนับสนุนหน้าจอสัมผัส
8. Multitasking สนับสนุนมัลติทาสกิงแอปพลิเคชัน
9. Flash Support แอนดรอยด์ 2.3 สนับสนุน Flash 10.1

2.6.2 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)

สถาปัตยกรรมของระบบปฏิบัติการแอนดรอยด์นั้นถูกแบ่งออกเป็นลำดับชั้น (Layer) ซึ่งโครงสร้างโดยรวมของระบบสามารถแสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 สถาปัตยกรรมแอนดรอยด์

เอกสารนี้เป็นที่มา : <http://kadroiz.blogspot.com/2012/03/android-architecture.html> โฆษณาด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์จะสังเกตได้ว่าการแบ่งออกมาเป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งก็คือ ส่วนของ Applications จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel สามารถอธิบายเป็นส่วนๆ ได้ดังนี้

2.6.2.1 Applications

Applications เป็นส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้เช่น แอปพลิเคชันรับ/ส่งอีเมล SMS ปฏิทิน แผนที่ เว็บเบราว์เซอร์ รายชื่อผู้ติดต่อ เป็นต้น ซึ่งแอปพลิเคชันจะอยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app

2.6.2.2 Application Framework

Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้ผู้พัฒนาสามารถพัฒนาโปรแกรมได้สะดวกและมีประสิทธิภาพมากยิ่งขึ้น ซึ่งสามารถพัฒนาโปรแกรมโดยใช้ภาษา Java ผ่านทาง API (Application Programming Interface) โดยไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งานแล้วนำมาใช้งาน 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

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ถือว่าผิดกฎหมาย

2.6.2.3 Libraries

Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งานแสดงได้ดังตารางที่ 2.2

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

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

Dalvik Virtual Machine ถูกออกแบบมาเพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ (Memory) หน่วยประมวลผลกลาง (CPU) และพลังงาน (Battery) ที่จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาก็คือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษา Java

2.6.2.4 Linux Kernel

Linux Kernel เป็นส่วนที่ทำหน้าที่ในการจัดการกับบริการหลักของระบบปฏิบัติการ โดยส่วนมากแล้วจะเกี่ยวข้องกับฮาร์ดแวร์โดยตรง เช่น หน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์รุ่น 2.6 (Linux 2.6.Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 ส่วนประกอบของแอปพลิเคชัน (Application Component)

คุณลักษณะอย่างหนึ่งของแอนดรอยด์ก็คือเป็นแอปพลิเคชันที่สามารถเป็นส่วนประกอบของแอปพลิเคชันอื่นๆ ได้ซึ่งแบ่งออกเป็น 4 ส่วน คือ

2.6.3.1 Activity

Activity หรือ User Interface คือหน้าจอที่ติดต่อกับผู้ใช้ โดยที่ในแต่ละแอปพลิเคชัน อาจจะมีได้มากกว่า 1 หน้าจอ หรือ 1 Activity ซึ่งแต่ละ Activity จะทำหน้าที่เก็บสถานะการใช้งานในส่วนต่างๆ โดยสิ่งที่อยู่ใน Activity จะถูกเรียกว่า View ซึ่งมีอยู่หลายรูปแบบ เช่น button text fields scroll bars menu items checkboxes และอื่นๆ

2.6.3.2 Service

Service เป็นส่วนที่ไม่มีผลการแสดงผล แต่ถูกเรียกใช้ให้รันอยู่ในลักษณะของ background process โดย service นั้นอาจจะมีการทำอะไรบางอย่าง เช่น ติดต่อรับส่งข้อมูลผ่านเครือข่ายหรือคำนวณค่าต่างๆ แล้วทำการส่งข้อมูลไปแสดงยัง Activity ก็ได้ หรือการเปิดเพลงในขณะที่เรากำลังทำงานบนแอปพลิเคชันอื่น

2.6.3.3 Broadcast Receiver

Broadcast Receiver หรือ Data Provider คือตัวที่ใช้สำหรับคอยรับและตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ การเปลี่ยนภาษา มีการโทรออก มีข้อความเข้าและอื่นๆ ถึงแม้ Broadcast Receiver จะไม่มีส่วนของการแสดงผล แต่ก็สามารถที่จะเรียก Activity ขึ้นมาแสดงผลให้ผู้ใช้ทราบได้ เรียกว่า Notification Manager ซึ่งจะเป็นตัวที่แจ้งเตือนในรูปแบบต่างๆ เช่น การสั่น การแสดงไฟกระพริบที่หน้าจอ หรือการส่งเสียงออกมาโดยจะมี icon แสดงอยู่บน status bar เพื่อแจ้งเหตุการณ์ที่เกิดขึ้น

2.6.3.4 Content Provider

Content Provider หรือ System Event Listener คือส่วนของการให้บริการข้อมูลสำหรับแต่ละแอปพลิเคชัน โดยที่ข้อมูลสามารถเก็บอยู่ในรูปแบบของไฟล์หรือฐานข้อมูลก็ได้ เช่น Google สามารถเข้าใช้งานข้อมูลร่วมกันกับผู้ใช้งานได้ในแอปพลิเคชันที่ต้องการข้อมูลของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 อีคลิปลี่ (Eclipse)

อีคลิปลี่คือโปรแกรมประเภท IDE (Integrated Development Environments) ใช้เป็นเครื่องมือสำหรับการพัฒนาซอฟต์แวร์หลากหลายภาษา นิยมใช้ในการพัฒนาโปรแกรมภาษา Java และภาษาต่างๆ ได้แก่ ภาษา C C++ Perl PHP และ Python ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ อีกทั้งเป็นโปรแกรมฟรีแวร์ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว มี Plug-in ต่างๆ เพื่อใช้เสริมประสิทธิภาพการทำงานมากมาย ใช้ได้ทั้งระบบปฏิบัติการ Window Linux และ Mac OS

2.7.1 แพลตฟอร์มของอีคลิปลี่ (Eclipse Platform)

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกันและมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อ download มาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ Debug โปรแกรมภาษา Java

2.7.2 Android SDK (Android Software Development Kit)

Android SDK เป็นชุดโปรแกรมที่ทาง Google พัฒนาออกมาเพื่อแจกจ่ายให้นักพัฒนาแอปพลิเคชันหรือผู้สนใจทั่วไปดาวน์โหลดไปใช้งานกันได้ โดยไม่มีค่าใช้จ่ายใดๆ จึงเป็นอีกหนึ่งปัจจัยที่ทำให้แอปพลิเคชันบนแอนดรอยด์นั้นเพิ่มขึ้นอย่างรวดเร็ว ในชุด SDK นั้นจะมีโปรแกรมและไลบรารีต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ อย่างเช่น Emulator หรือโปรแกรมจำลอง ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถสร้างแอปพลิเคชันและนำมาทดลองใช้งานบน Emulator ก่อน โดยมีสภาวะแวดล้อมเหมือนกับการนำไปใช้งานโทรศัพท์เคลื่อนที่ที่ใช้ระบบปฏิบัติการแอนดรอยด์จริงๆ เลย โดยสามารถดาวน์โหลด Android SDK ที่มีให้เลือกทั้งบน Windows Mac และ Linux

2.8 ภาษา JAVA

ภาษา Java คือภาษาคอมพิวเตอร์ที่ถูกพัฒนาขึ้นโดยบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้น คลาสคือที่เก็บกระบวนการ (Method) หรือ พฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 ประวัติภาษา JAVA

ภาษา Java เป็นภาษาโปรแกรมเชิงวัตถุที่พัฒนาขึ้นโดยเจมส์ กอสลิงและทีมวิศวกรของเขา ซึ่งบริษัทซันไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษา C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษา Objective-C แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ภายหลังเปลี่ยนไปใช้ชื่อ “จาวา” ซึ่งเป็นชื่อกาแฟแทน

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

2.8.2 การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมที่ประกอบด้วยกลุ่มของวัตถุ (Object) แต่ละวัตถุจะจัดเป็นกลุ่มในรูปของคลาส ซึ่งแต่ละคลาสอาจมีคุณสมบัติการปกป้อง (Encapsulation) การสืบทอด (Inheritance) หรือการพ้องรูป (Polymorphism)

2.8.3 คุณลักษณะเด่นของภาษา JAVA

1. โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกันโดยไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่องชั้น โปรแกรมนั้น ก็สามารถถูกคอมไพล์และรันบนเครื่องคอมพิวเตอร์ส่วนบุคคลได้
2. เมื่อเปรียบเทียบโค้ดของโปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java กับ C++ พบว่าโปรแกรมที่เขียนโดยภาษา Java จะมีจำนวนโค้ดน้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ถึง 4 เท่าและใช้เวลาในการเขียนโปรแกรมน้อยกว่าประมาณ 2 เท่า
3. ภาษา Java มีการรักษาความปลอดภัยทั้งระดับต่ำและระดับสูง ได้แก่ Electronic signature Public and Private key management Access control และ Certification ของภาษา Java

2.8.4 รูปแบบของภาษา JAVA

ภาษา Java เป็นภาษาที่ไม่กำหนดแบบการเขียนโปรแกรมในแต่ละบรรทัด แต่ละบรรทัดสามารถเขียนคำสั่งได้หลายคำสั่ง สามารถแทรกคำอธิบาย (Comment) เป็นภาษาที่บังคับอักขระตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก มีตัวดำเนินการ (Operators) หลายชนิดให้ได้ใช้งาน นอกจากคำสั่งนั้นเป็นคำสั่งที่ผู้ใช้สร้างขึ้นใหม่ อาจกำหนดเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กก็ได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถเขียนชุดคำสั่งที่ประกอบด้วยตัวดำเนินการหลายตัวที่ต่างชนิดกัน ในชุดคำสั่งหนึ่งๆ ได้ โดยภาษา Java จะจัดลำดับการประมวลผลตามลำดับการทำงานของตัวดำเนินการ

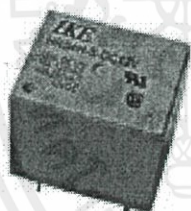
รูปแบบคำสั่ง (Statements) ของภาษา Java คือ ส่วนประมวลผล (Execute) เพื่อให้ได้ผลลัพธ์ออกมาทุกคำสั่งจะต้องจบด้วยเครื่องหมายเซมิโคลอน (;)

2.8.5 ซอร์สโค้ด (Source Code)

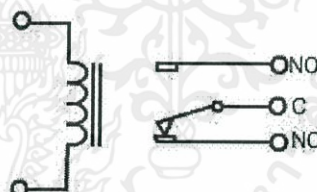
ในภาษา Java จะมีซอร์สโค้ดเป็นไฟล์ที่มีนามสกุล *.java เมื่อผ่านการคอมไพล์แล้วจะมีไฟล์เพิ่มมาเป็นไฟล์ที่มีนามสกุลเป็น *.class

2.9 รีเลย์ (Relay)

รีเลย์คืออุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ตัด-ต่อวงจร คล้ายกับสวิตช์ โดยใช้หลักการหน้าสัมผัสและการที่จะให้รีเลย์ทำงานก็ต้องจ่ายไฟให้ตามที่กำหนดเพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ จะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิดและตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้ก็จะกลายเป็นวงจรเปิด ไฟที่ใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจากแหล่งจ่ายไฟของเครื่อง ดังนั้นทันทีที่เปิดเครื่องก็จะทำให้รีเลย์ทำงาน รีเลย์และสัญลักษณ์ของรีเลย์แสดงได้ดังรูปที่ 2.9 และรูปที่ 2.10



รูปที่ 2.9 รีเลย์



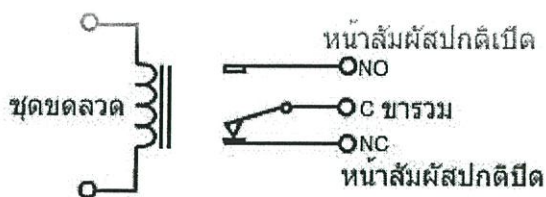
รูปที่ 2.10 สัญลักษณ์ของรีเลย์

ที่มา : <http://www.smartlearningweb.com/knowledge/relay/relay.htm>

2.9.1 โครงสร้างของรีเลย์

ภายในโครงสร้างของรีเลย์ จะประกอบไปด้วยขดลวด 1 ชุดและหน้าสัมผัส ซึ่งในหน้าสัมผัส 1 ชุด จะประกอบไปด้วย หน้าสัมผัสแบบปกติปิด (Normally Close: NC) ในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (C) และหน้าสัมผัสแบบปกติเปิด (Normally Open: NO) ขานี้จะต่อเข้ากับขาร่วม (C) เมื่อขดลวดมีแรงดันตกคร่อมหรือกระแสไหลผ่าน (ในปริมาณที่เพียงพอ) ในรีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด ซึ่งขึ้นอยู่กับผู้ผลิต โครงสร้างของรีเลย์แสดงได้ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปขึ้นประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

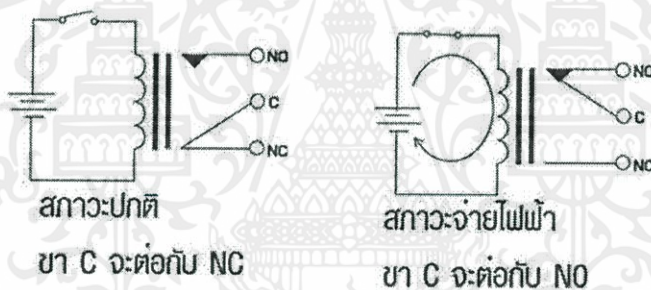


รูปที่ 2.11 โครงสร้างของรีเลย์

ที่มา : <http://www.smartlearningweb.com/knowledge/relay/relay.htm>

2.9.2 หลักการทำงานของรีเลย์

เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดจะทำให้ขดลวดเกิดสนามแม่เหล็กไปดึงแผ่นหน้าสัมผัสให้ดึงลงมาแตะหน้าสัมผัสอีกอันทำให้มีกระแสไหลผ่านหน้าสัมผัสไปได้ดังรูปที่ 2.12



รูปที่ 2.12 สภาวะการทำงานของรีเลย์

ที่มา : <http://www.smartlearningweb.com/knowledge/relay/relay.htm>

2.9.3 ส่วนประกอบของรีเลย์

ขาย้ายแรงดันใช้งาน ซึ่งจะมีอยู่ 2 ขา คือขา C หรือขา COM หรือขาคอมมอน จะเป็นขาคู่ระหว่าง NO และ NC ขา NO โดยปกติขานี้จะเปิดเอาไว้ จะทำงานเมื่อเราป้อนแรงดันให้รีเลย์ ขา NC โดยปกติขานี้จะต่อกับขา C ในกรณีที่เรายังไม่ได้จ่ายแรงดัน หน้าสัมผัสของ C และ NC จะต่อถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Analog to Digital Converter

สัญญาณที่ใช้ในอุปกรณ์อิเล็กทรอนิกส์มี 2 ชนิด คือสัญญาณอนาล็อกและสัญญาณดิจิทัล สัญญาณอนาล็อกจะใช้ในอุปกรณ์ต่างๆ ไปและใช้ในการควบคุมแบบเก่า ในปัจจุบันมีไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์เข้ามาช่วยในการควบคุมอุปกรณ์ต่างๆ มากมาย ซึ่งทำให้การควบคุมนั้นทำได้ง่ายและรวดเร็วยิ่งขึ้น แต่ในการควบคุมนั้นเราจำเป็นต้องใช้สัญญาณดิจิทัลในการติดต่อกับไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ แต่ในความเป็นจริงนั้นเราใช้สัญญาณอนาล็อกในการควบคุม ดังนั้นเราจึงจำเป็นต้องมีการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลแล้วจึงนำสัญญาณนั้นเข้ามาสูไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์เพื่อใช้ควบคุมระบบต่อไป

2.10.1 สัญญาณอนาล็อก (Analog Signal)

สัญญาณอนาล็อกหมายถึงสัญญาณที่เกี่ยวข้องกับข้อมูลแบบต่อเนื่อง (Continuous Data) ที่มีขนาดไม่คงที่ มีลักษณะเป็นเส้นโค้งต่อเนื่องกันไป โดยการส่งสัญญาณแบบอนาล็อกจะถูกรบกวนให้มีการแปลความหมายผิดพลาดได้ง่าย เช่น สัญญาณเสียงในสายโทรศัพท์ เป็นต้น

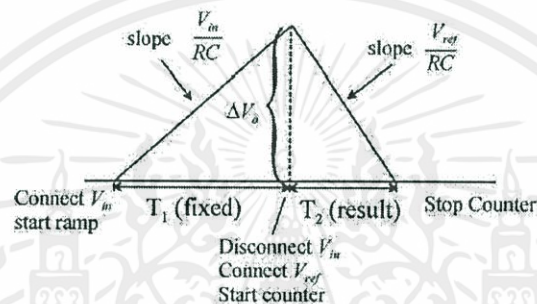
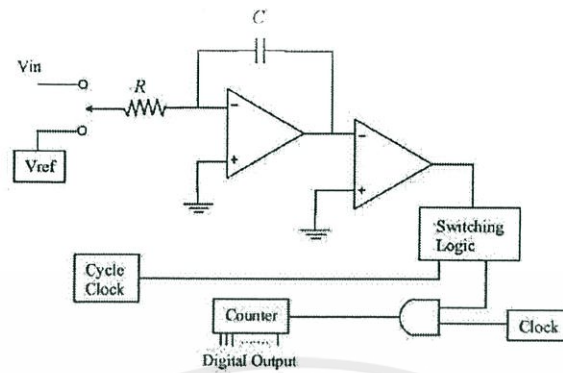
2.10.2 สัญญาณดิจิทัล (Digital Signal)

สัญญาณดิจิทัลหมายถึงสัญญาณที่เกี่ยวข้องกับข้อมูลแบบไม่ต่อเนื่อง (Discrete Data) ที่มีขนาดแน่นอนซึ่งขนาดดังกล่าวอาจกระโดดไปมาระหว่างค่าสองค่า คือสัญญาณระดับสูงสุดและสัญญาณระดับต่ำสุด ซึ่งสัญญาณดิจิทัลนี้เป็นสัญญาณที่คอมพิวเตอร์ใช้ในการทำงานและติดต่อสื่อสารกัน

2.10.3 Digital Ramp ADC

Digital Ramp ADC เป็นวิธีที่ง่ายที่สุดของการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล โดยใช้อัลกอริทึม การนับค่าเพิ่มขึ้นเรื่อยๆ แล้วนำผลที่ได้จากการนับไปเปรียบเทียบกับค่าที่ต้องการที่ตั้งไว้ ลักษณะการทำงานเป็นดังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



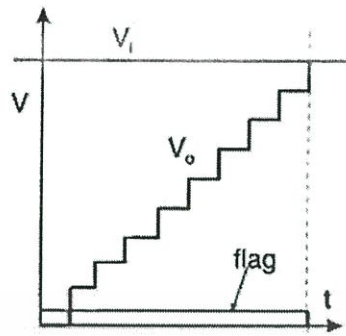
รูปที่ 2.13 การทำงานของ Digital Ramp ADC

ที่มา : <http://dev.emcelettronica.com/analog-to-digital-converter-digital-to-analog-converter-overview>

จากวงจร Counter เป็นอุปกรณ์นับค่าที่เพิ่มขึ้นทีละหนึ่ง แล้วส่งค่าที่ได้ให้ D/A มีขา Reset รับสัญญาณ Reset เมื่อต้องการให้เริ่มนับใหม่ D/A เมื่อรับค่าที่นับเพิ่มขึ้นทีละหนึ่งจากตัวนับ ก็แปลงค่าให้เป็นสัญญาณนาฬิกาที่มีค่าความต่างศักย์ค่าๆ หนึ่ง แล้วส่งต่อเข้าไปที่อุปกรณ์ Comparator จะเป็นอุปกรณ์ตัวเปรียบเทียบค่าความต่างศักย์ของอินพุตและค่าจากที่ตัวนับ ถ้าหากทั้งสองสัญญาณมีค่าเท่ากันส่งค่าความต่างศักย์ 0 โวลต์ออกมา (ลอจิก 0) ถ้าไม่เท่ากันก็จะส่งความต่างศักย์ที่ไม่ใช่ 0 โวลต์ออกมา (ลอจิก 1)

ซึ่งค่าความต่างศักย์ที่ออกมา จะนำมาเข้าลอจิกเกต "และ" กับ สัญญาณนาฬิกาจะได้ค่าลอจิกออกมา ถ้าผลลัพธ์ออกมาเป็นสัญญาณนาฬิกาแสดงว่ายังไม่ได้ผลลัพธ์เท่าที่ต้องการ สัญญาณนาฬิกาก็จะไปทำให้ตัวนับนับเพิ่มขึ้นต่อไปและเมื่อได้ค่าผลลัพธ์ดิจิทัลที่ต้องการแล้วค่าที่ได้จากตัวเปรียบเทียบจะให้ค่าความต่างศักย์เป็น 0 (ลอจิก 0) ซึ่งเมื่อนำมาเข้าลอจิกเกต "และ" กับสัญญาณนาฬิกาแล้วก็จะให้ลอจิก 0 ซึ่งทำให้ตัวนับไม่นับเพิ่มอีก ก็จะได้ค่าดิจิทัลจากตัวนับที่ต้องการ จากคำอธิบายข้างต้นจะได้กราฟของ VO ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 output voltage ของ Digital Ramp ADC

ที่มา : <http://capet1.blogspot.com/2009/02/analog-to-digital-converter.html>

การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล มีประโยชน์มากในการควบคุมอุปกรณ์ สวิตชิง ซึ่งมีลักษณะการแปลงสัญญาณได้หลายวิธี แต่ละวิธีจะมีอัลกอริทึม ความรวดเร็วในการทำงานและการใช้อุปกรณ์ฮาร์ดแวร์ต่างกันด้วย ทำให้ขนาดและราคาต่างกัน ขึ้นกับความต้องการของผู้ใช้ที่จะต้องเลือกให้เหมาะสมกับงานที่ใช้และงบประมาณที่มีอยู่

2.11 ไมโครคอนโทรลเลอร์ Arduino

Arduino คือเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพ ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิปเดี่ยวและมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน เหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์อินพุต/เอาต์พุตต่างๆ ได้มากมาย ทั้งในแบบที่เป็นการทำงานเดี่ยวอิสระหรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นๆ

2.11.1 หลักการทำงานของ Arduino

แพลตฟอร์มของ Arduino จะมีการเขียนโปรแกรมเล็กๆ (Firmware) ฝังไว้ในไมโครคอนโทรลเลอร์บริเวณ Bootloader ของไมโครคอนโทรลเลอร์ตระกูล AVR โดยมีหน้าที่รับข้อมูลจาก Serial Port มาเขียนลงในพื้นที่ว่างของตัวเอง (Flash) ในเวลาต่อมา Serial Port เริ่มหายไป ได้มีการเปลี่ยนจาก Serial Port เป็น USB ด้วยการใช้ชิพ FTDI เข้ามาทำหน้าที่แปลง USB ให้เป็น Serial (USB to Serial Converter) แล้วก็ต่อเข้ากับไมโครคอนโทรลเลอร์โดยตรง ทำให้สามารถใช้ Arduino กับ USB Port ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 โปรแกรม Arduino IDE

ในการเขียนโปรแกรมโดยใช้ภาษาของ Arduino แบ่งได้เป็น 2 ส่วนหลักคือ โครงสร้างภาษา ตัวแปรและค่าคงที่และฟังก์ชัน

ภาษาของ Arduino จะอ้างอิงตามภาษา C/C++ จึงกล่าวได้ว่า การเขียนโปรแกรมสำหรับ Arduino ก็คือการเขียนโปรแกรมภาษาซี โดยเรียกฟังก์ชันและไลบรารีที่ทาง Arduino ได้เตรียมไว้ให้แล้วจึงสะดวกมากสำหรับการใช้งาน โดยโครงสร้างของ Arduino จะแบ่งได้เป็น 2 ส่วนคือ void setup () และ void loop ()

โดยฟังก์ชัน setup () เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชันนี้เพียงครั้งเดียวใช้ในการกำหนดค่าเริ่มต้นของการทำงาน ส่วนฟังก์ชัน loop () เป็นส่วนที่โปรแกรมจะทำคำสั่งในฟังก์ชันนี้ต่อเนื่องกันตลอดเวลา โดยปกติใช้กำหนดโหมดการทำงานของขาต่างๆ กำหนดการสื่อสารแบบอนุกรม ส่วนของ loop เป็นโค้ดโปรแกรมที่ทำงานเช่น อ่านค่าอินพุต ประมวลผลสั่งงานเอาต์พุต โดยส่วนกำหนดค่าเริ่มต้นเช่น ตัวแปร จะต้องเขียนที่ส่วนหัวของโปรแกรมก่อนถึงตัวฟังก์ชัน นอกจากนั้นยังต้องคำนึงถึงตัวพิมพ์เล็ก ตัวพิมพ์ใหญ่ของตัวแปรและชื่อฟังก์ชันให้ถูกต้อง

```

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

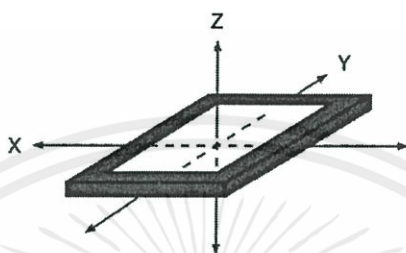
```

รูปที่ 2.15 โปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 Accelerometer Sensor

Accelerometer Sensor เป็นเซนเซอร์ที่สามารถทำการวัดค่าความเร่งได้ในสามระนาบคือ แกน X Y และ Z ตามความเร่งของโลก โดยจะทำการส่งค่าออกมาเป็นตัวเลขจำนวนจริงตามค่าของระนาบต่างๆ ดังรูปที่ 2.16

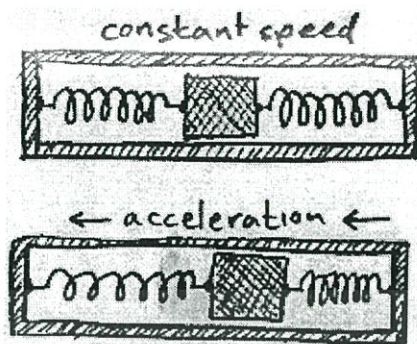


รูปที่ 2.16 ระนาบที่ใช้วัดค่าความเร่งในการเอียง
ที่มา : <http://www.akexorcist.com/2013/03/android-code-gyroscope.html>

1. แนวแกน X วัดค่าความเร่งในการเอียงจากด้านข้าง (ซ้าย, ขวา) โดยค่าบวกแสดงค่าของการเคลื่อนที่ไปทางขวา ส่วนค่าลบแสดงการเคลื่อนที่ไปทางซ้าย
2. แนวแกน Y วัดค่าความเร่งในการเอียงจากการเคลื่อนไปข้างหน้าหรือถอยหลัง โดยค่าบวกแสดงการเคลื่อนที่ไปข้างหน้า ส่วนค่าลบแสดงการเคลื่อนที่ไปข้างหลัง
3. แนวแกน Z วัดค่าความเร่งในการเอียงจากการเคลื่อนที่ขึ้นลง โดยค่าบวกแสดงการเคลื่อนที่ขึ้น ส่วนค่าลบแสดงการเคลื่อนที่ลง

โครงสร้างภายในของ Accelerometer Sensor จะประกอบด้วยสปริงและลูกตุ้มน้ำหนักดังรูปที่ 2.17 เมื่อมีการเคลื่อนที่ด้วยความเร่งลูกตุ้มน้ำหนักจะถูกกดไปอีกฝั่งตรงข้ามกับการเคลื่อนที่ สปริงก็ทำหน้าที่ดึงกลับเข้าที่อีกครั้งเมื่อหยุดการเคลื่อนที่ การเคลื่อนที่ด้วยความเร็วคงที่คือความเร่งเท่ากับศูนย์ ค่าที่วัดได้ก็จะไม่เปลี่ยนแปลง

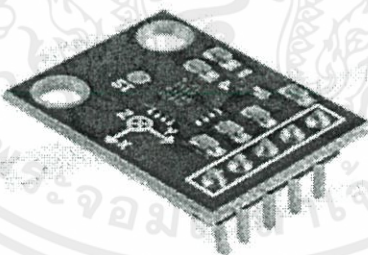
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 โครงสร้างภายในของ Accelerometer Sensor
ที่มา : <https://sites.google.com/site/thaimulticopter/acc>

2.12.1 หลักการทำงานของ Accelerometer Sensor

Accelerometer Sensor เป็นเซนเซอร์วัดค่าความเร่งที่เพิ่มขึ้นหรือลดลง (ในหน่วย m/s^2) ถ้าความเร่งของแรงโน้มถ่วงคือ $9.8 m/s^2$ โดยมีหลักการทำงานคือให้น้ำหนักถึงห้องสี่เหลี่ยมเล็กๆ ที่ทุกด้านของกำแพงจะมีสปริงติดอยู่ เวลาที่ห้องนี้เอียงไปทางใดทางหนึ่ง สปริงก็จะยุบไปด้านนั้นๆ โดยสมมุติว่าแรงดันของสปริงมีน้อยกว่าแรงโน้มถ่วงของโลกและใช้วงจรไฟฟ้าในการดึง Output Analog ออกมาใช้งาน จะใช้ Accelerometer Sensor สำหรับเป็นตัวชี้ว่าอยู่ในสถานะ Static หรือ Dynamic ซึ่งเซนเซอร์วัดค่าความเร่งในการเอียงที่ใช้ในปริญญาโทฉบับนี้คือ Accelerometer Sensor ADXL335



รูปที่ 2.18 Accelerometer Sensor ADXL335
ที่มา : <http://www.amazon.com/Gino-ADXL335-Module-Digital-Gravity/dp/B00COD8UOO>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.2 คุณสมบัติของ Accelerometer Sensor ADXL335

สามารถอ่านค่าความเอียงในแกน X Y และ Z จาก Accelerometer Sensor ADXL335 ที่เชื่อมต่อกับอนาล็อกอินพุต 0 1 และ 2 ของวงจร Analog to Digital Converter แสดงค่าออกทางจอภาพแบบอนุกรมแล้วนำค่าที่ได้ไปคำนวณเพื่อแสดงค่าความเอียง

2.13 Wi-Fi

Wi-Fi (Wireless Fidelity) เป็นองค์กรหนึ่งที่ทำหน้าที่ทดสอบผลิตภัณฑ์ระบบเครือข่ายไร้สาย (Wireless Local Area Network: WLAN) ภายใต้เทคโนโลยีการสื่อสารมาตรฐาน IEEE 802.11 ว่าอุปกรณ์ทุกตัวซึ่งต่างยี่ห้อกันนั้นสามารถติดต่อสื่อสารกันได้โดยไม่มีปัญหาหากว่าอุปกรณ์ตัวนั้นผ่านมาตรฐานจะมีตรา Wi-Fi certified ซึ่งเป็นอันรู้กันว่าอุปกรณ์ชิ้นนั้นสามารถติดต่อสื่อสารกับอุปกรณ์ตัวอื่นที่มีตรา Wi-Fi certified นี้ได้เช่นกัน

ต่อมา Wi-Fi กลายเป็นคำศัพท์สำหรับเรียกอุปกรณ์ต่างๆ ที่สามารถใช้ได้กับระบบเครือข่ายไร้สายซึ่งอยู่ภายใต้มาตรฐาน IEEE 802.11 หรือกล่าวได้ว่า Wi-Fi หมายถึงเครือข่ายไร้สายที่ใช้กับระบบเครือข่ายไม่ว่าจะเป็นในองค์กรหรือในระบบเครือข่ายอินเทอร์เน็ต

ระบบเครือข่ายไร้สายหมายถึงเทคโนโลยีที่ช่วยให้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องหรือกลุ่มของเครื่องคอมพิวเตอร์สามารถสื่อสารกันได้รวมถึงการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่ายคอมพิวเตอร์ด้วยเช่นกัน โดยปราศจากการใช้สายสัญญาณในการเชื่อมต่อ แต่จะใช้คลื่นวิทยุเป็นช่องทางการสื่อสารแทน การรับส่งข้อมูลระหว่างกันจะผ่านอากาศทำให้ไม่ต้องเดินสายสัญญาณและติดตั้งใช้งานได้สะดวกขึ้น

2.13.1 มาตรฐาน Wi-Fi

การสื่อสารผ่านเครือข่ายไร้สายมีมาตรฐาน IEEE 802.11 เป็นมาตรฐานกำหนดรูปแบบการสื่อสาร ซึ่งมาตรฐานแต่ละตัวจะบอกถึงความเร็วและคลื่นความถี่สัญญาณที่แตกต่างกันในการสื่อสารข้อมูล เช่น 802.11b และ 802.11g ที่ความเร็ว 11 Mbps และ 54 Mbps ตามลำดับและขอบเขตของสัญญาณครอบคลุมพื้นที่ประมาณ 100 เมตรในพื้นที่โล่งและประมาณ 30 เมตรในอาคาร ซึ่งระยะทางของสัญญาณมีผลกระทบจากสิ่งรอบข้างหลายๆอย่าง เช่น โทรศัพท์มือถือ ความหนาของกำแพง เครื่องใช้ไฟฟ้า อุปกรณ์อิเล็กทรอนิกส์ต่างๆ รวมถึงร่างกายมนุษย์ด้วยเช่นกัน สิ่งเหล่านี้มีผลกระทบต่อการใช้งานเครือข่ายไร้สายทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.2 ลักษณะการเชื่อมต่อของอุปกรณ์

Wi-Fi ได้กำหนดลักษณะการเชื่อมต่อของอุปกรณ์ภายในระบบเครือข่ายไร้สายไว้ 2 ลักษณะคือโหมด Infrastructure และโหมด Ad-Hoc หรือ Peer-to-Peer โดยลักษณะการเชื่อมต่อของอุปกรณ์ที่ใช้ในปฏิญานินพจน์นี้คือ โหมด Ad-Hoc

2.13.2.1 โหมด Ad-Hoc หรือ Peer-to-Peer

เครือข่าย Wi-Fi ในโหมด Ad-Hoc หรือ Peer-to-Peer เป็นเครือข่ายที่ปิดคือไม่มีสถานีแม่ข่ายและไม่มีการเชื่อมต่อกับเครือข่ายอื่น บริเวณของเครือข่าย Wi-Fi ในโหมด Ad-Hoc จะถูกเรียกว่า Independent Basic Service Set (IBSS) ซึ่งสถานีผู้ใช้หนึ่งสามารถติดต่อสื่อสารข้อมูลกับสถานีผู้ใช้อื่นๆ ในเขต IBSS เดียวกันได้โดยตรงโดยไม่ต้องผ่านสถานีแม่ข่าย แต่สถานีผู้ใช้จะไม่สามารถรับส่งข้อมูลกับเครือข่ายอื่นๆได้

2.14 กล้องเว็บแคม (Webcam)

เว็บแคม (Webcam) หรือชื่อเรียกเต็มๆ ว่า Web Camera อาจเรียกว่า Video Camera หรือ Video Conference ก็ได้ เว็บแคมเป็นอุปกรณ์อินพุตที่สามารถจับภาพเคลื่อนไหวของเราไปปรากฏในหน้าจอมอนิเตอร์และสามารถส่งภาพเคลื่อนไหวนี้ผ่านระบบเครือข่ายเพื่อให้คนอื่นฝั่งหนึ่งสามารถเห็นตัวเราเคลื่อนไหวได้เหมือนอยู่ต่อหน้า ถือได้ว่าเป็นอุปกรณ์ที่มีประโยชน์และเริ่มมีความจำเป็นมากขึ้นเรื่อยๆ



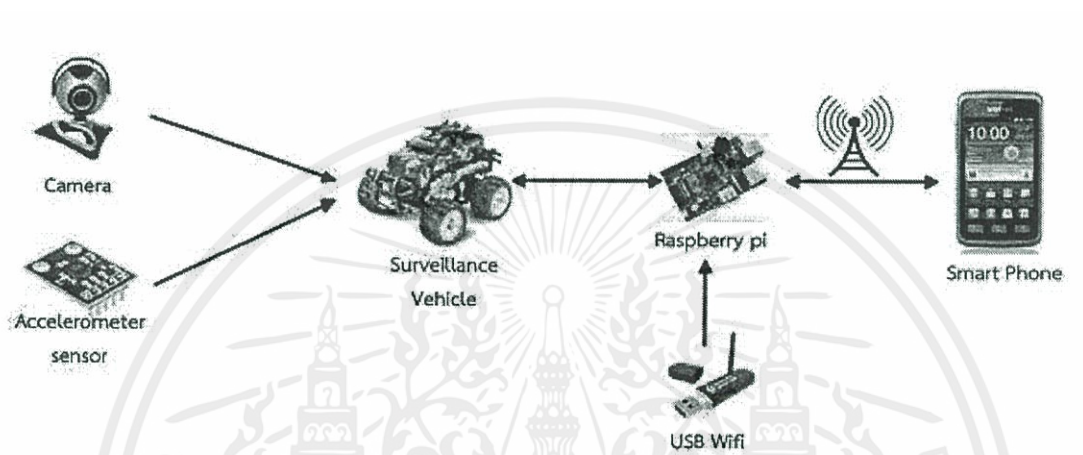
รูปที่ 2.19 กล้องเว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปฏิญานินพนธ์

3.1 หลักการทำงานของระบบ

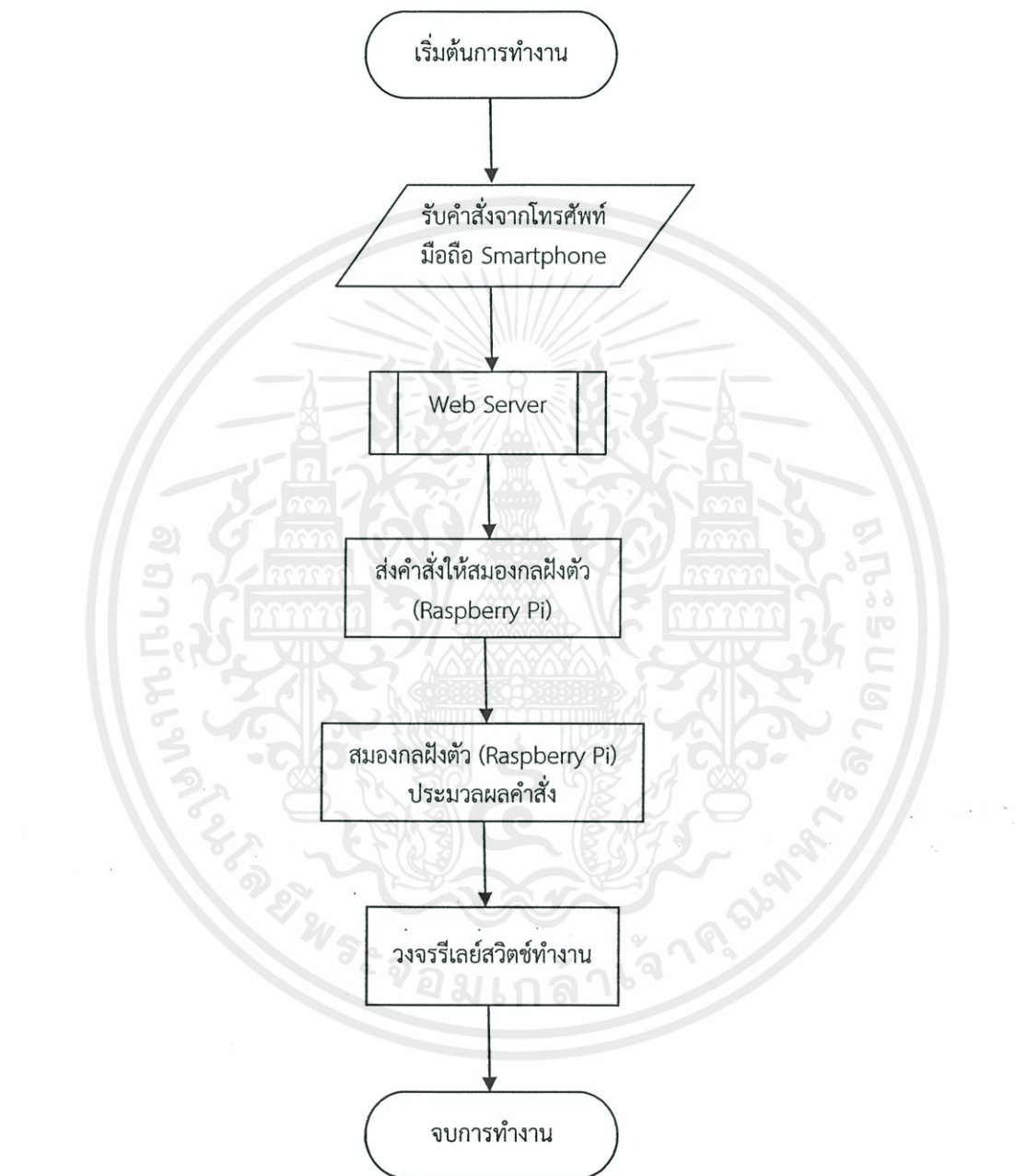


รูปที่ 3.1 บล็อกไดอะแกรมโดยรวมของระบบ

หลักการทำงานของระบบแสดงได้ดังรูปที่ 3.1 ในภาคเรียนที่ 1 จะเป็นการทำงานของระบบ โดยเริ่มต้นเมื่อผู้ใช้แอปพลิเคชันแอนดรอยด์บนสมาร์ตโฟนทำการเชื่อมต่อเข้ากับ Web Server ผ่านทางโครงข่ายอินเทอร์เน็ตแบบไร้สายแล้วส่งค่าจากแอปพลิเคชันแอนดรอยด์เข้าไปยังฐานข้อมูล (Database) บน Web Server ซึ่งให้สมองกลฝังตัว (Raspberry Pi) ทำหน้าที่เป็น Web Server ในตัว ส่วนในภาคเรียนที่ 2 ทำการประมวลผลคำสั่งที่ได้จากภาคเรียนที่ 1 ส่งไปยังส่วนควบคุมวงจรรีเลย์สวิตช์เพื่อให้ขั้วมอเตอร์บังคับล้อของรถให้เคลื่อนที่ ติดตั้งกล้องลงไปบนรถเพื่อให้สามารถแสดงภาพเหตุการณ์จริงขณะที่รถเคลื่อนที่บนหน้าจอแอปพลิเคชัน รวมทั้งติดตั้งเซนเซอร์ในการวัดความเอียงของตัวรถ โดยอาศัยตัวแปลงสัญญาณจากอนาล็อกเป็นดิจิทัลประมวลผลค่าความเอียงจากค่าที่เป็นพิกัดมาคำนวณให้เป็นองศา จากนั้นนำค่าความเอียงไปแสดงผลบนแอปพลิเคชันแอนดรอยด์

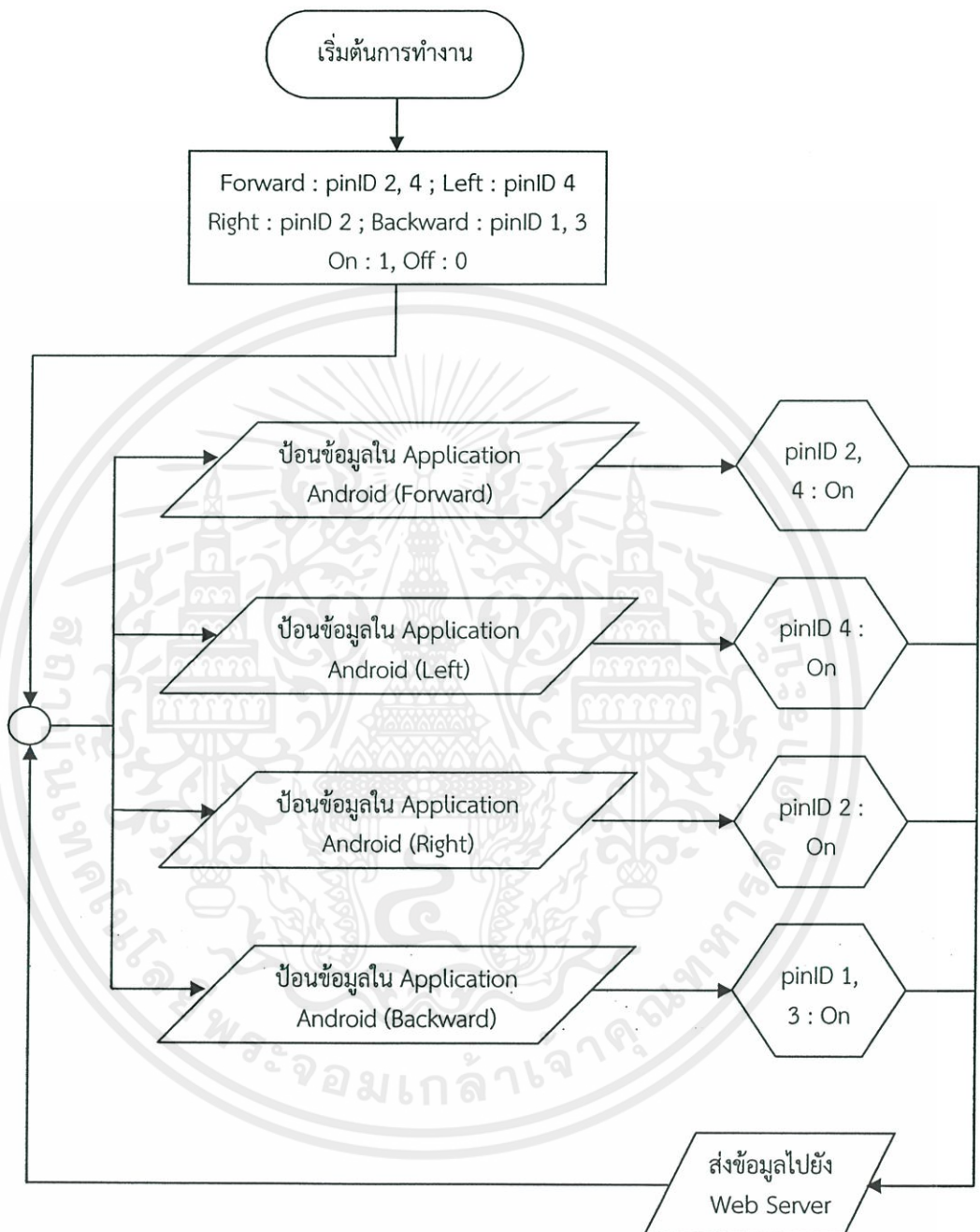
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โฟลว์ชาร์ตการทำงานของระบบ



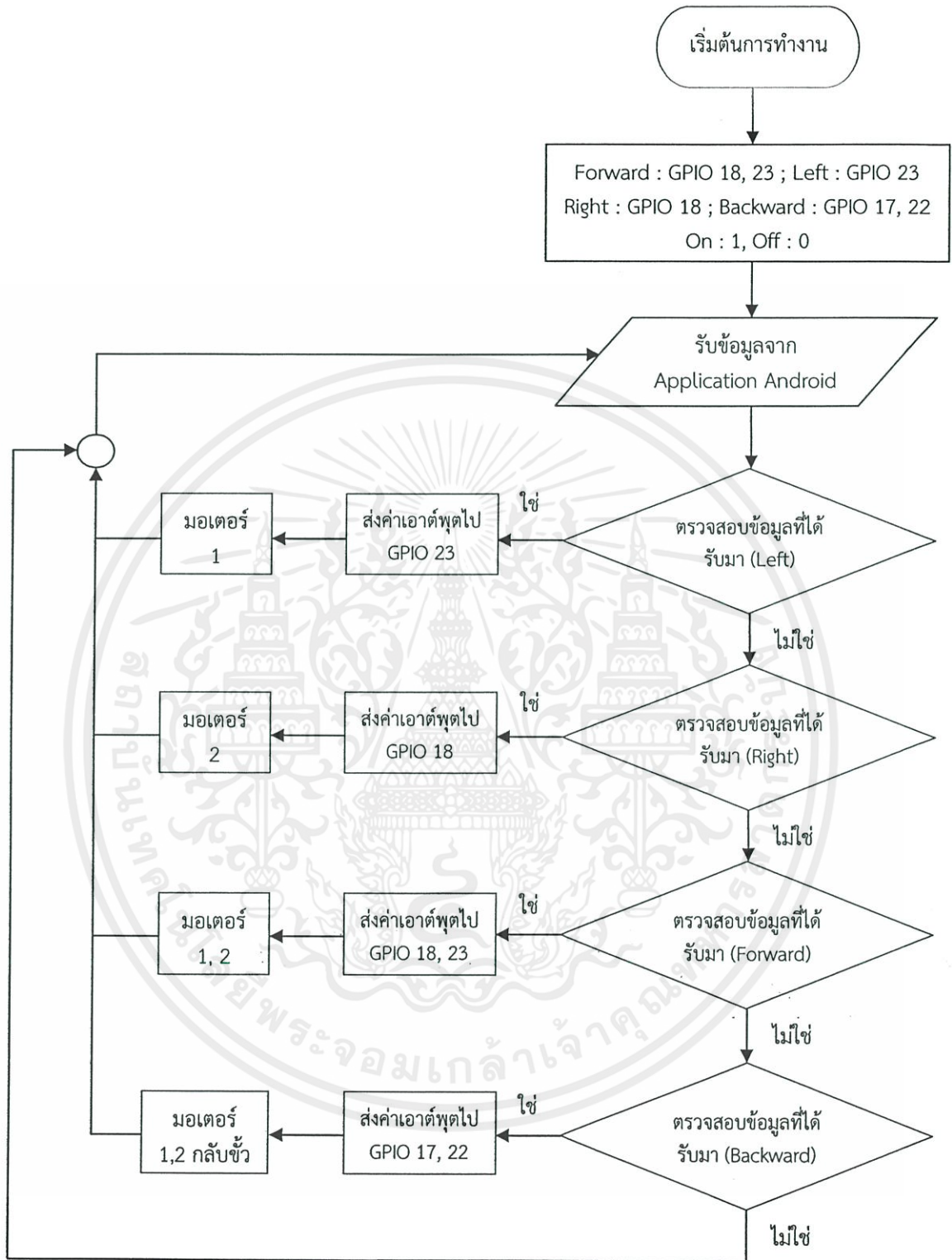
รูปที่ 3.2 โฟลว์ชาร์ตการทำงานของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาค้นคว้าเท่านั้น มิใช่ผู้ใดให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 โฟลว์ชาร์ตการทำงานในส่วนผู้ใช้แอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับเอาไว้ใช้เพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.4 โฟลว์ชาร์ตการทำงานของสมองกลฝังตัว (Raspberry Pi)
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบและสร้างเซิร์ฟเวอร์บนสมองกลฝังตัว (Raspberry Pi)

+ Options

← T →				pinID	pinNumber	pinStatus			
<input type="checkbox"/>		Edit		Copy		Delete	1	17	0
<input type="checkbox"/>		Edit		Copy		Delete	2	18	0
<input type="checkbox"/>		Edit		Copy		Delete	3	22	0
<input type="checkbox"/>		Edit		Copy		Delete	4	23	0

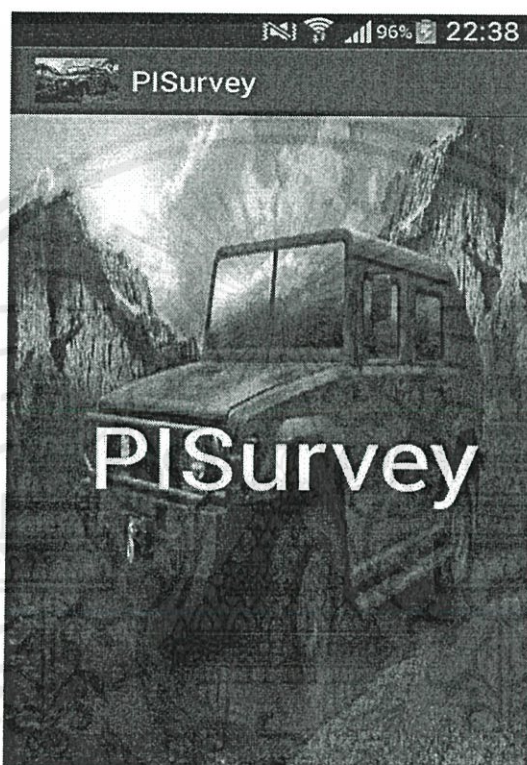
รูปที่ 3.5 ฐานข้อมูลบนเซิร์ฟเวอร์ในสมองกลฝังตัว (Raspberry Pi)

ภายในสมองกลฝังตัวจะทำการออกแบบรวมทั้งสร้างเซิร์ฟเวอร์และฐานข้อมูล เพื่อให้สมองกลฝังตัวกลายเป็นเครื่องเซิร์ฟเวอร์ใช้ในการสื่อสารและรับส่งค่าจากแอปพลิเคชันโดยใช้โปรแกรม phpMyAdmin ในการสร้างฐานข้อมูล MySQL เพื่อใช้อ่านค่าจากการป้อนข้อมูลจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) สำหรับควบคุมการเคลื่อนที่ของรถ แล้วส่งค่าเป็นแรงดันเพื่อขับมอเตอร์ให้ทำงาน จากนั้นเขียนคำสั่งในการเชื่อมต่อกับแอปพลิเคชัน โดยอาศัยภาษา Python และใช้ภาษา PHP ในการสร้างเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบส่วนติดต่อกับผู้ใช้ (Graphic User Interface) แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

ใช้โปรแกรมอีclipse (Eclipse) และภาษา Java ในการเขียนโปรแกรมออกแบบแอปพลิเคชันเพื่อติดต่อกับผู้ใช้แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์



รูปที่ 3.6 หน้าจอหลักของแอปพลิเคชัน

3.4.1 ออกแบบและสร้างระบบสมาชิกและระบบล็อกอินบนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

ส่วนนี้เป็นการสร้างแอปพลิเคชันแอนดรอยด์ให้ทำงานร่วมกับฐานข้อมูล MySQL เพื่อส่งข้อมูลบนแอปพลิเคชันแอนดรอยด์ไปบันทึกที่ฐานข้อมูล MySQL โดยกำหนดการทำงานสองส่วนคือ

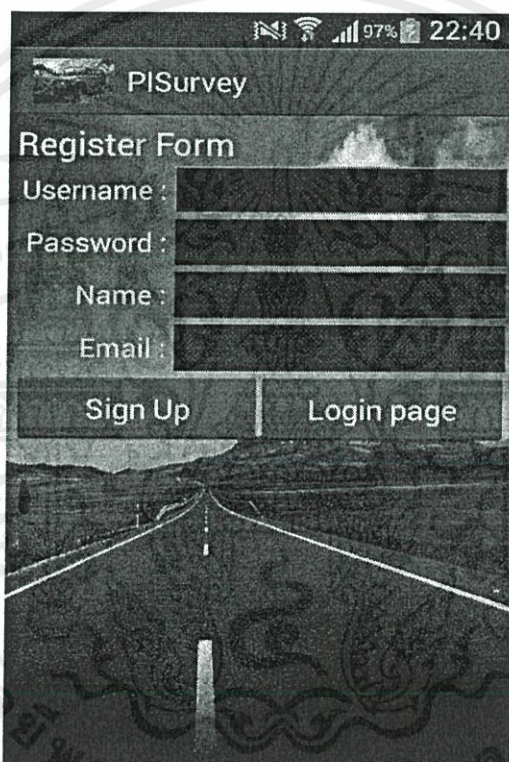
3.4.1.1 เขียนโปรแกรมกำหนดการทำงานทางฝั่ง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะใช้สมองกลฝั่งตัวเป็น Sever โดยใช้โปรแกรมจัดการฐานข้อมูล phpMyAdmin สร้างฐานข้อมูล MySQL และเขียนโปรแกรมภาษา PHP ซึ่งทำหน้าที่ประมวลผลข้อมูลที่ส่งมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) และบันทึกข้อมูลลงในฐานข้อมูล

3.4.1.2. เขียนโปรแกรมกำหนดการทำงานทางฝั่ง Client

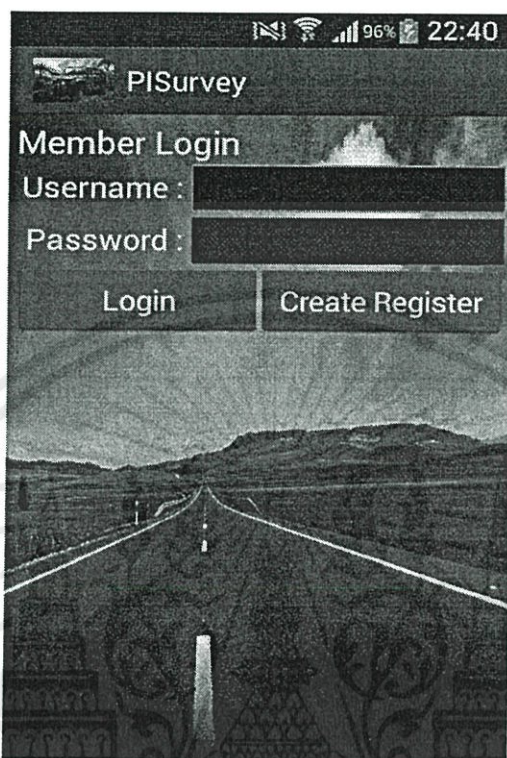
1. สร้างระบบสมัครสมาชิกบนแอปพลิเคชันแอนดรอยด์



รูปที่ 3.7 หน้าจอสำหรับป้อนข้อมูลสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สร้างระบบล็อกอินบนแอปพลิเคชันแอนดรอยด์

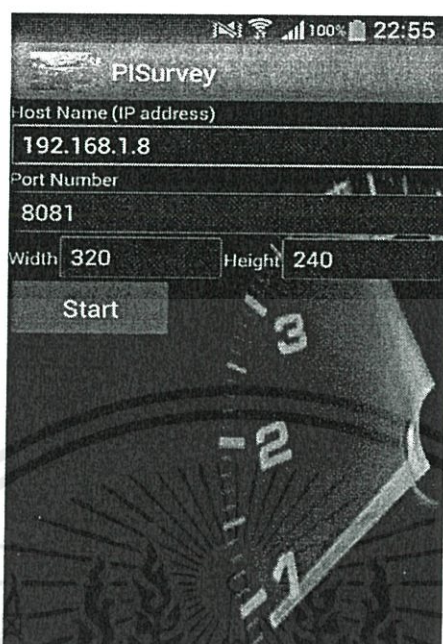


รูปที่ 3.8 หน้าจอสำหรับล็อกอิน

3.4.2 ออกแบบและสร้างหน้าแอปพลิเคชันสำหรับควบคุมการเคลื่อนที่ของรถและแสดงภาพจากกล้อง

ส่วนนี้เป็นการสร้างหน้าแอปพลิเคชันสำหรับป้อนไอพีแอดเดรสและพอร์ตของสมองกล้องตัวในการเชื่อมต่อกับกล้องและหน้าแอปพลิเคชันสำหรับควบคุมการเคลื่อนที่ของรถและแสดงภาพจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 หน้าจอสำหรับป้อนไอพีแอดเดรสและพอร์ตของสมองกล้องตัว

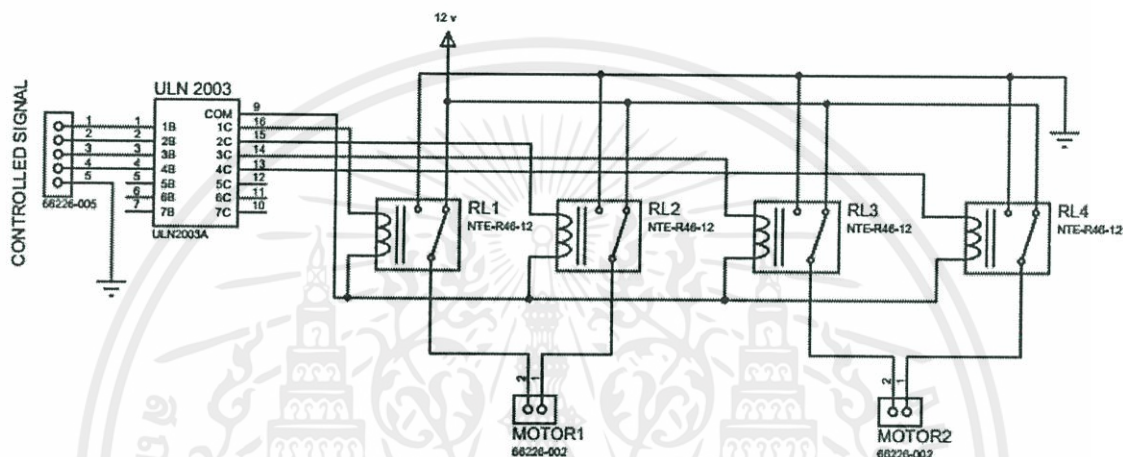


รูปที่ 3.10 หน้าจอสำหรับควบคุมการเคลื่อนที่ของรถและแสดงภาพจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบและสร้างวงจรรีเลย์สวิตช์

การออกแบบวงจรรีเลย์สวิตช์จะใช้โปรแกรม Proteus 7 Professional ในการออกแบบวงจรแสดงดังรูปที่ 3.11 โดยสัญญาณขาเข้าที่ออกจากพอร์ต GPIO จะผ่านทางไอซี ULN 2003 ซึ่งเป็นส่วนที่ขับแรงดันให้รีเลย์สวิตช์ทำงานสับสวิตช์จาก NC ไป NO เกิดแรงดันตกคร่อมตัวมอเตอร์ ทำให้มอเตอร์หมุน ส่งผลให้รถสามารถเคลื่อนที่ได้

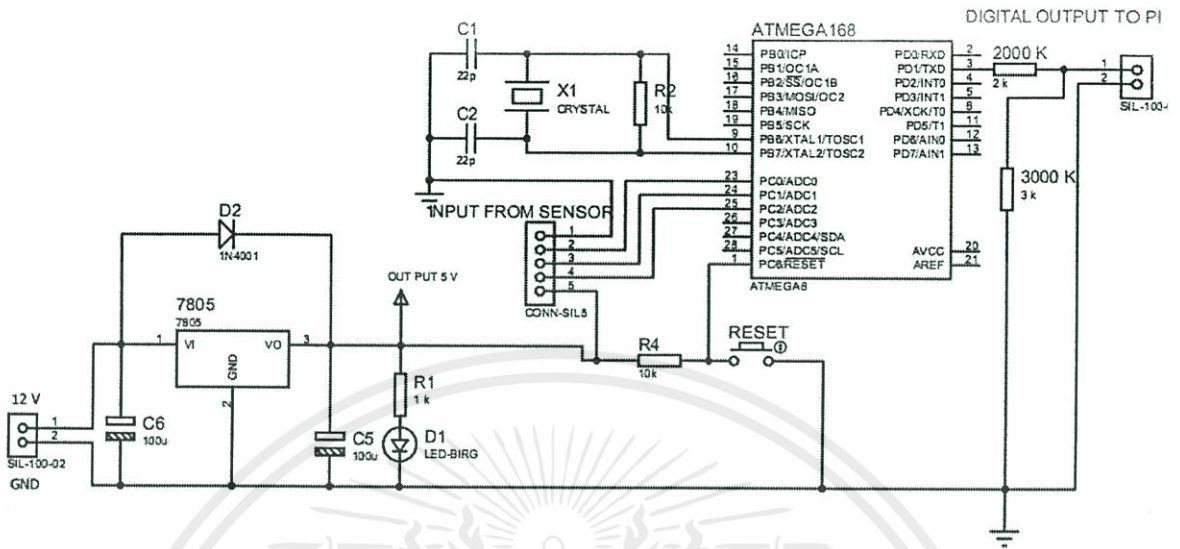


รูปที่ 3.11 วงจรรีเลย์สวิตช์ที่ออกแบบโดยใช้โปรแกรม Proteus 7 Professional

3.6 การออกแบบและสร้างวงจร Analog to Digital Converter

การออกแบบวงจร Analog to Digital Converter โดยใช้ ATmega168 เป็นไมโครคอนโทรลเลอร์ที่ใช้ในการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลและใช้โปรแกรม Proteus 7 Professional ในการออกแบบวงจรแสดงดังรูปที่ 3.12 ในวงจรมีแหล่งจ่ายไฟ 12 โวลต์ ผ่าน IC 7805 ซึ่งทำหน้าที่รักษาระดับแรงดันเอาต์พุตให้คงที่ 5 โวลต์ ป้อนเป็นไฟเลี้ยงให้กับวงจร โดยวงจรรับค่าแรงดันอนาล็อก X Y และ Z จาก Accelerometer Sensor ทางพอร์ต 23 24 และ 25 แล้วทำ Analog to Digital Converter ส่งค่าออกทาง พอร์ต TX (2) ของวงจร แล้วจึงนำสัญญาณนั้นเข้ามาสู่ Raspberry Pi เพื่อใช้ในการส่งค่าไปแสดงผลบนแอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 วงจร Analog to Digital Converter ที่ออกแบบโดยใช้โปรแกรม Proteus 7 Professional

3.7 ส่วนประกอบของรถตรวจการณ์

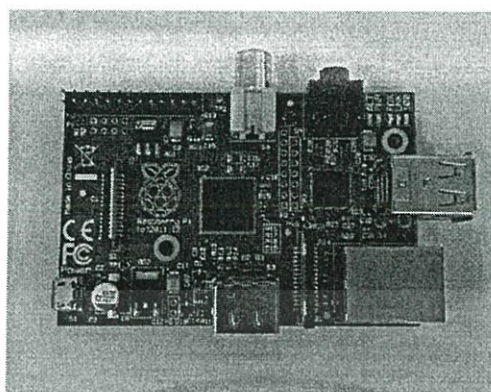
รถตรวจการณ์ที่สร้างขึ้นประกอบด้วย 2 ส่วน คือส่วนควบคุมการเคลื่อนที่ของรถ และส่วนของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

3.7.1 ส่วนควบคุมการเคลื่อนที่ของรถ

3.7.1.1 Raspberry Pi

สมองกลฝังตัว (Raspberry Pi) ทำหน้าที่เป็นหน่วยประมวลผลกลางของระบบทั้งหมด ซึ่งในส่วนของฮาร์ดแวร์จะประมวลผลคำสั่งที่ได้รับจากแอปพลิเคชันแอนดรอยด์ แล้วส่งค่าไปควบคุมวงจรรีเลย์สวิตช์เพื่อให้วงจรขับเคลื่อนมอเตอร์ให้ทำงานและรับค่าจากวงจรการแปลงสัญญาณอนาล็อกเป็นดิจิทัล โดยค่าที่ได้เป็นค่าพิกัด X Y และ Z จากนั้นส่งค่าไปแสดงผลบนแอปพลิเคชันต่อไป

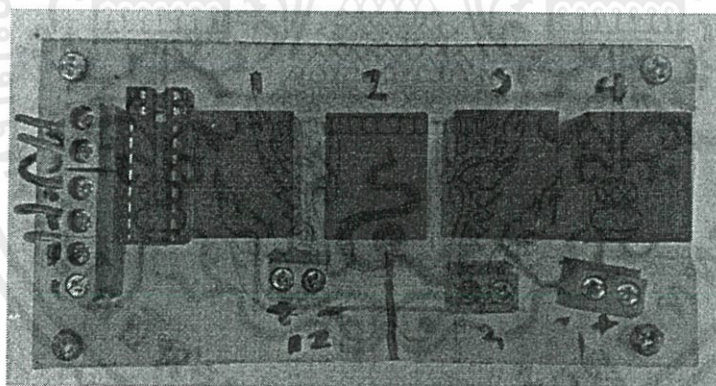
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Raspberry Pi

3.7.1.2 วงจรรีเลย์สวิตช์

จากการออกแบบวงจรรีเลย์สวิตช์ในโปรแกรม Proteus 7 Professional นำมาสร้างเป็นวงจรจริงดังรูปที่ 3.14 ซึ่งวงจรรีเลย์สวิตช์เป็นอุปกรณ์ที่ใช้รับค่าอินพุตที่ได้จากพอร์ต GPIO ของ Raspberry Pi และทำหน้าที่เป็น Switch เพื่อควบคุมการหมุนมอเตอร์ของรถตรวจการณ์



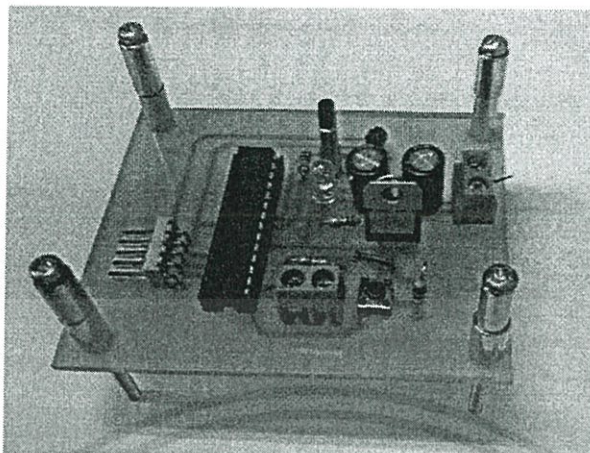
รูปที่ 3.14 วงจรรีเลย์สวิตช์

3.7.2 ส่วนของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

3.7.2.1 วงจร Analog to Digital Converter

จากการออกแบบวงจร Analog to Digital Converter ในโปรแกรม Proteus 7 Professional นำมาสร้างเป็นวงจรจริงดังรูปที่ 3.15

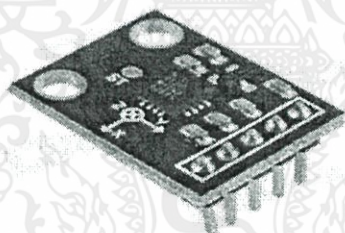
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 วงจร Analog to Digital Converter

3.7.2.2 เซนเซอร์วัดค่าความเอียง (Accelerometer Sensor)

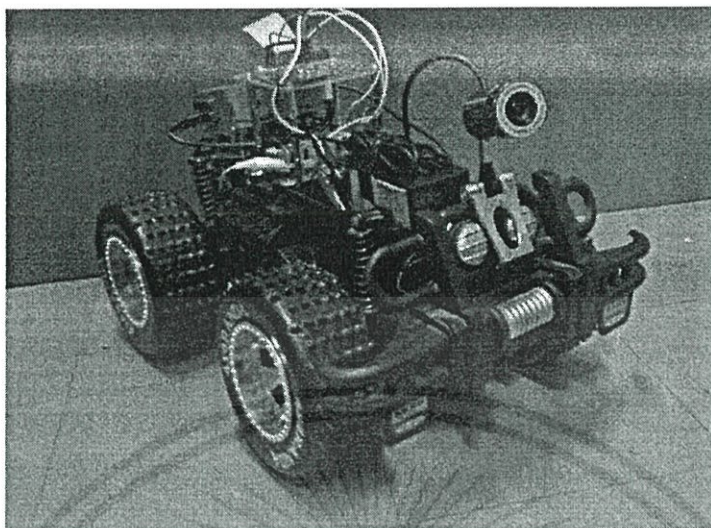
เซนเซอร์วัดค่าความเอียงในการเอียงที่ใช้ในปริญญาโทฉบับนี้คือ Accelerometer Sensor ADXL335 ดังรูปที่ 3.16 สามารถอ่านค่าความเอียงในแกน X Y และ Z ที่เชื่อมต่อกับบอานาล็อกอินพุต 23 24 และ 25 ของวงจร Analog to Digital Converter



รูปที่ 3.16 Accelerometer Sensor ADXL335

จากส่วนประกอบทั้งสองส่วน นำมาประกอบเป็นรถตรวจการณได้ดังรูปที่ 3.17 โดยส่วนประกอบทั้งสองส่วนสามารถควบคุมการเคลื่อนที่ของรถและแสดงค่าความเอียงจากเซนเซอร์ได้พร้อมๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 รถตรวจการณ์

3.8 เครื่องมือที่ใช้ในการทดลอง

3.8.1 ฮาร์ดแวร์ (Hardware) มีดังนี้

1. Raspberry Pi
2. จอแสดงผล
3. คีย์บอร์ดและเมาส์
4. สาย LAN และสาย HDMI
5. วงจรรีเลย์สวิตช์
6. วงจร Analog to Digital Converter
7. เซนเซอร์วัดค่าความเอียง Accelerometer Sensor ADXL335
8. USB Wi-Fi
9. กล้องเว็บแคม
10. โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

3.8.2 ซอฟต์แวร์ (Software) มีดังนี้

1. ระบบปฏิบัติการลินุกซ์ (Linux)
2. โปรแกรมอีคลิปส์ (Eclipse)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือที่สงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรม Proteus 7 Professional
4. โปรแกรม Arduino IDE

3.9 การจัดเก็บผลการทดลอง

3.9.1 การทดสอบการรับส่งข้อมูลระหว่าง Server และ Client โดย Wireshark

1. ทำการตรวจสอบ IP Address ในฝั่ง Server (สมองกลฝังตัว) ด้วยคำสั่ง IF Config บนหน้าจอ LXTerminal
2. ทำการตรวจสอบ IP Address ในฝั่ง Client (แอปพลิเคชันแอนดรอยด์) ด้วยคำสั่ง IP Config บน command prompt
3. ทำการ ping เข้าไปยังเครื่อง Server เพื่อตรวจสอบสถานะการเชื่อมต่อบนโครงข่ายด้วย IP Address ของฝั่ง Server (สมองกลฝังตัว)
4. ทำการตรวจสอบกระบวนการสื่อสารระหว่างเครื่อง Client (แอปพลิเคชันแอนดรอยด์) และเครื่อง Server (สมองกลฝังตัว) ที่มีการรับส่งข้อมูลกันด้วยโปรแกรม Wireshark

3.9.2 การทดสอบความแรงของสัญญาณ Wi-Fi โดยใช้ Wi-Fi Analyzer

1. ใช้โปรแกรม Wi-Fi Analyzer บนสมาร์ตโฟนในการวัดค่าความแรงของสัญญาณ Wi-Fi
2. ทำการตรวจสอบว่าระยะทางที่สัญญาณสามารถเชื่อมต่อกับอุปกรณ์รับสัญญาณนั้นมีระยะทางไกลเท่าไร
3. บันทึกค่าความแรงของสัญญาณ

3.9.3 การทดลองควบคุมการเคลื่อนที่ของรถผ่านสมาร์ตโฟนบนระบบปฏิบัติการแอนดรอยด์

1. ทำการต่อวงจรรีเลย์สวิตช์เข้ากับ Raspberry Pi ผ่านทางพอร์ต GPIO โดยต่อพอร์ต 17 เข้ากับสวิตช์ควบคุมทิศทางในการถอยหลัง ต่อพอร์ต 18 เข้ากับสวิตช์ควบคุมทิศทางในการเลี้ยวขวา ต่อพอร์ต 22 เข้ากับสวิตช์ควบคุมทิศทางในการเดินหน้าและต่อพอร์ต 23 เข้ากับสวิตช์ควบคุมทิศทางในการเลี้ยวซ้าย
2. ทำการต่อแหล่งจ่ายไฟ 12 โวลต์ ให้กับวงจรรีเลย์สวิตช์
3. เขียนโปรแกรมบน Raspberry Pi เพื่อควบคุมทิศทางและการเคลื่อนที่ของรถ
4. จากนั้นทำการเชื่อมต่อสมาร์ตโฟนและ Raspberry Pi ผ่านเครือข่าย

อินเทอร์เน็ตแบบไร้สายโดยใช้ USB Wi-Fi

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ของรถ
5. เปิดแอปพลิเคชันแอนดรอยด์บนสมาร์ตโฟนเพื่อควบคุมทิศทางการเคลื่อน
 6. บันทึกผลการทดลองการเคลื่อนที่ของรถใน 4 ทิศทาง

3.9.4 การทดลองการส่งค่าความเอียงของ Accelerometer Sensor จากวงจร Analog to Digital Converter ไปแสดงผลบนแอปพลิเคชันแอนดรอยด์

1. ทำการต่อวงจร Analog to Digital Converter เข้ากับ Raspberry Pi
2. ทำการต่อแหล่งจ่ายไฟ 12 โวลต์ ให้กับวงจร Analog to Digital Converter
3. ทำต่อพอร์ต 2 ซึ่งเป็นพอร์ต TX ของวงจร Analog to Digital Converter เข้ากับ GPIO พอร์ต 15 ซึ่งเป็นพอร์ต RX ของ Raspberry Pi และเชื่อมต่อกราวด์ทั้งสองเข้าด้วยกัน
4. เขียนโปรแกรมบน Raspberry Pi ในการรับค่าความลาดเอียงของพื้นที่ จากวงจร Analog to Digital Converter
5. จากนั้นทำการเชื่อมต่อสมาร์ตโฟนและ Raspberry Pi ผ่านเครือข่ายอินเทอร์เน็ตแบบไร้สายโดยใช้ USB Wi-Fi
6. เปิดแอปพลิเคชันแอนดรอยด์บนสมาร์ตโฟนเพื่อดูค่าความลาดเอียงที่แสดงบนหน้าแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการรับส่งข้อมูลระหว่าง Server และ Client โดย Wireshark

มีวัตถุประสงค์ เพื่อทดสอบการเชื่อมต่อระหว่าง Server และ Client ด้วยการ Ping และเพื่อตรวจสอบการรับส่งข้อมูลระหว่าง Server และ Client ในระหว่างการรับส่งข้อมูลนั้นเกิดเหตุการณ์ใดบ้าง

```
wlan0    Link encap:Ethernet  HWaddr 00:13:ef:40:0b:b8
          inet addr:192.168.66.99  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115165 errors:0 dropped:157 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10901098 (10.3 MiB)  TX bytes:8944 (8.7 KiB)
```

รูปที่ 4.1 IP Address ของ Server (สมองกลฝังตัว)

จากรูปที่ 4.1 จะเห็นว่า IP Address ของฝั่ง Server คือ 192.168.66.99 และในรูปที่ 4.2 จะเห็นว่า IP Address ของฝั่ง Client คือ 192.168.1.6

```
Wireless LAN adapter Wireless Network Connection:
Connection-specific DNS Suffix . . : localdomain
IPv6 Address. . . . . : 2002:a1f6:125e:d:28a3:e87e:30ef:5217
Site-local IPv6 Address . . . . . : fec0::d:28a3:e87e:30ef:5217%2
Temporary IPv6 Address. . . . . : 2002:a1f6:125e:d:584d:cc62:a29a:264b
Link-local IPv6 Address . . . . . : fe80::28a3:e87e:30ef:5217%13
IPv4 Address. . . . . : 192.168.1.6
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 192.168.0.1
```

รูปที่ 4.2 IP Address ของ Client (แอปพลิเคชันแอนดรอยด์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

C:\Users\Administrator>ping 192.168.66.99

Pinging 192.168.66.99 with 32 bytes of data:
Reply from 192.168.66.99: bytes=32 time=1827ms TTL=64
Reply from 192.168.66.99: bytes=32 time=8ms TTL=64
Reply from 192.168.66.99: bytes=32 time=5ms TTL=64
Reply from 192.168.66.99: bytes=32 time=17ms TTL=64

Ping statistics for 192.168.66.99:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 1827ms, Average = 464ms

```

รูปที่ 4.3 การเชื่อมต่อระหว่าง Server และ Client โดยการ Ping

จากรูปที่ 4.3 จะเห็นว่าฝั่ง Server (สมองกลฝั่งตัว) สามารถเชื่อมต่อกับฝั่ง Client (แอปพลิเคชันแอนดรอยด์) เพื่อรับส่งข้อมูลระหว่างกันได้

No.	Time	Source	Destination	Protocol	Length	Info
53	2.48392300	192.168.66.99	192.168.1.6	TCP	60	http > 51624 [SYN, ACK] Seq=1 Ack=1 Win=14600 Len=0 M
54	2.48607200	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
55	2.48704100	192.168.1.6	192.168.66.99	HTTP	478	GET /control.php HTTP/1.1
56	2.52157900	192.168.66.99	192.168.1.6	TCP	54	http > 51624 [ACK] Seq=1 Ack=425 Win=15680 Len=0
59	2.53430000	192.168.66.99	192.168.1.6	HTTP	713	HTTP/1.1 200 OK (text/html)
65	2.73369300	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=425 Ack=660 Win=16860 Len=0
74	3.33593200	192.168.1.6	192.168.66.99	HTTP	478	GET /control.php HTTP/1.1
77	3.38651700	192.168.66.99	192.168.1.6	HTTP	712	HTTP/1.1 200 OK (text/html)
80	3.58673600	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=849 Ack=1318 Win=16200 Len=0
93	4.09530100	192.168.1.6	192.168.66.99	HTTP	478	GET /control.php HTTP/1.1
96	4.12617800	192.168.66.99	192.168.1.6	HTTP	712	HTTP/1.1 200 OK (text/html)
102	4.32674600	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=1273 Ack=1976 Win=17520 Len=0
108	4.55742300	192.168.1.6	192.168.66.99	HTTP	478	GET /control.php HTTP/1.1
109	4.61480600	192.168.66.99	192.168.1.6	HTTP	712	HTTP/1.1 200 OK (text/html)
110	4.81478600	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=1697 Ack=2634 Win=16860 Len=0
116	5.36379700	192.168.1.6	192.168.66.99	TCP	66	51623 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4
117	5.37275200	192.168.66.99	192.168.1.6	TCP	66	http > 51623 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 M
118	5.37287500	192.168.1.6	192.168.66.99	TCP	54	51623 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
126	6.78125500	192.168.66.99	192.168.1.6	TCP	68	[TCP Retransmission] http > 51623 [SYN, ACK] Seq=0 Ac
127	6.78135200	192.168.1.6	192.168.66.99	TCP	66	[TCP Dup ACK 118#1] 51623 > http [ACK] Seq=1 Ack=1 W
158	9.64340400	192.168.66.99	192.168.1.6	TCP	54	http > 51624 [FIN, ACK] Seq=2634 Ack=1697 Win=18944 L
159	9.64363700	192.168.1.6	192.168.66.99	TCP	54	51624 > http [ACK] Seq=1697 Ack=2635 Win=16860 Len=0

รูปที่ 4.4 ตรวจสอบการรับส่งข้อมูลระหว่าง Server และ Client โดยใช้ Wireshark

จากรูปที่ 4.4 จากนั้นทำการตรวจสอบกระบวนการติดต่อสื่อสารระหว่างฝั่ง Client (แอปพลิเคชันแอนดรอยด์) และฝั่ง Server (สมองกลฝั่งตัว) ที่มีการรับส่งข้อมูลระหว่างกัน ด้วยโปรแกรม Wireshark โดยทำการส่งข้อมูลจากแอปพลิเคชันแอนดรอยด์ไปยัง Server แล้วทำการ Capture เพื่อหา Source 192.168.1.6 (Client) และ Destination 192.168.66.99 (Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองแสดงได้ดังรูปที่ 4.5

```

Internet Protocol Version 4, Src: 192.168.1.6 (192.168.1.6), Dst: 192.168.66.99 (192.168.66.99)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 40
  Identification: 0x273a (10042)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x4edc [validation disabled]
  Source: 192.168.1.6 (192.168.1.6)
  Destination: 192.168.66.99 (192.168.66.99)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: 51624 (51624), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
  Source port: 51624 (51624)
  Destination port: http (80)
  [Stream index: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header length: 20 bytes
  Flags: 0x010 (ACK)
  Window size value: 4380
  [Calculated window size: 17520]
  [Window size scaling factor: 4]
  Checksum: 0xb5ce [validation disabled]
  [SEQ/ACK analysis]

```

รูปที่ 4.5 รายละเอียดของแต่ละ Packet ที่ได้จากการดักจับจาก Wireshark

ในรูปที่ 4.5 หน้าต่างโปรแกรม Wireshark จะแสดงถึงรายละเอียดของกระบวนการสื่อสารข้อมูลในเน็ตเวิร์ค รวมทั้งสามารถแสดงรายละเอียดข้อมูลต่างๆ ที่สำคัญในแต่ละ Package ที่ใช้ในการรับส่งข้อมูล

จากผลการทดลองด้วยการ Ping พบว่า Server และ Client สามารถเชื่อมต่อกันได้ และจากการใช้ Wireshark เพื่อตรวจสอบการรับส่งข้อมูลระหว่าง Server และ Client พบว่าเมื่อใส่ชนิดของโปรโตคอลตรงช่อง Filter เป็นชนิด TCP จะบอกถึงการติดต่อระหว่าง Server และ Client ทั้ง IP Address ของ Server IP Address ของ Client Port ที่ใช้ของ Server และ Port ที่ใช้ของ Client และยังบอกถึงความยาวของข้อมูล รวมทั้งข้อมูลต่างๆ ที่อยู่ใน Packet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

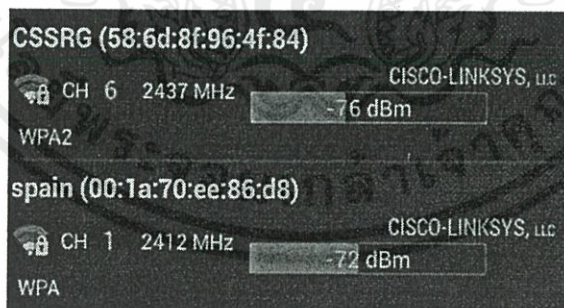
4.2 ผลการทดสอบความแรงของสัญญาณ Wi-Fi โดยใช้ Wi-Fi Analyzer

ในการทดสอบนี้จะใช้โปรแกรม Wi-Fi Analyzer บนสมาร์ตโฟนในการวัดค่าความแรงของสัญญาณ Wi-Fi เพื่อตรวจสอบว่าระยะทางที่สัญญาณสามารถเชื่อมต่อกับอุปกรณ์รับสัญญาณนั้นได้ระยะทางไกลเท่าไร เพื่อให้อุปกรณ์ทั้งฝั่ง Client (แอปพลิเคชันแอนดรอยด์) กับฝั่ง Server (สมองกลฝั่งตัว) ติดต่อสื่อสารกันได้อย่างมีประสิทธิภาพ



รูปที่ 4.6 เมื่ออยู่ใกล้เครื่องปล่อยสัญญาณ Wi-Fi

จากรูปที่ 4.6 จะเห็นว่าค่าความแรงของสัญญาณ Wi-Fi (Spain) มีค่า Received Signal Strength Indication (RSSI) เท่ากับ -41 dBm ในระยะ 1 เมตร ซึ่งมีค่ามากทำให้ทราบว่าเครื่องส่งกับเครื่องรับอยู่ใกล้กัน แต่เมื่อทดลองเดินห่างออกไปจากจุดเริ่มต้น เป็นระยะ 20 เมตร พบว่าค่า Received Signal Strength Indication (RSSI) มีค่าเป็น -72 dBm ซึ่งมีค่าน้อยแสดงว่าเครื่องส่งกับเครื่องรับอยู่ไกลกันแสดงดังรูปที่ 4.7

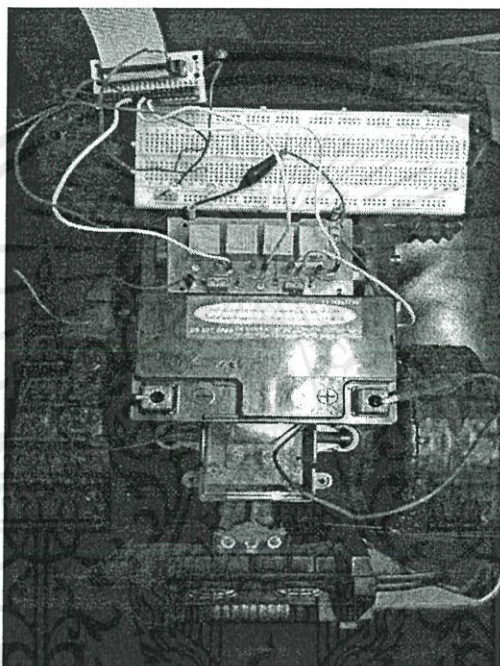


รูปที่ 4.7 เมื่ออยู่ไกลเครื่องปล่อยสัญญาณ Wi-Fi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

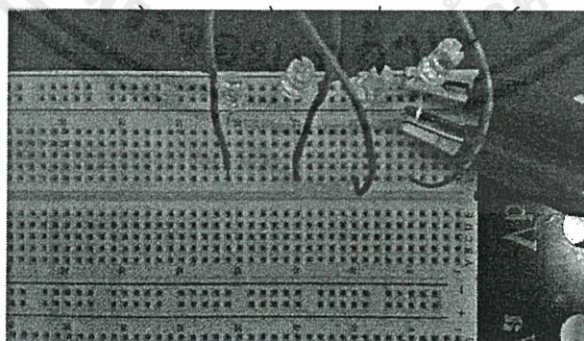
4.3 ผลการทดลองควบคุมการเคลื่อนที่ของรถผ่านสมาร์ทโฟนบนระบบปฏิบัติการแอนดรอยด์

ทำการบันทึกผลการทดลองโดยการนำ LED มาต่อเข้ากับพอร์ต GPIO เพื่อดูสถานะการทำงานของรถ รวมทั้งวัดสัญญาณเอาต์พุตที่ออกจากพอร์ต GPIO โดยใช้ Oscilloscope และควบคุมการเคลื่อนที่ของรถผ่านแอปพลิเคชันแอนดรอยด์บนสมาร์ทโฟน



รูปที่ 4.8 เมื่อรถยังไม่มีการเคลื่อนที่

LED 1 LED 2 LED 3 LED 4



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.9 LED ที่ใช้แสดงสถานะของแต่ละพอร์ต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.9 นำ LED มาต่อเข้ากับพอร์ต GPIO โดย LED ตัวที่ 1 ต่อเข้ากับพอร์ต 17 LED ตัวที่ 2 ต่อเข้ากับพอร์ต 18 LED ตัวที่ 3 ต่อเข้ากับพอร์ต 22 และ LED ตัวที่ 4 ต่อเข้ากับพอร์ต 23

จากนั้นเขียนโปรแกรมกำหนดการทำงานทางฝั่ง Server (สมองกลฝังตัว) โดยใช้โปรแกรมจัดการฐานข้อมูล phpMyAdmin สร้างฐานข้อมูล MySQL และเขียนโปรแกรมภาษา PHP ซึ่งทำหน้าที่ประมวลผลข้อมูลที่ส่งมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) และบันทึกข้อมูลลงในฐานข้อมูลดังรูปที่ 4.10

+ Options

	pinID	pinNumber	pinStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	17	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	18	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	22	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	23	0

รูปที่ 4.10 ฐานข้อมูลบน Server (สมองกลฝังตัว)

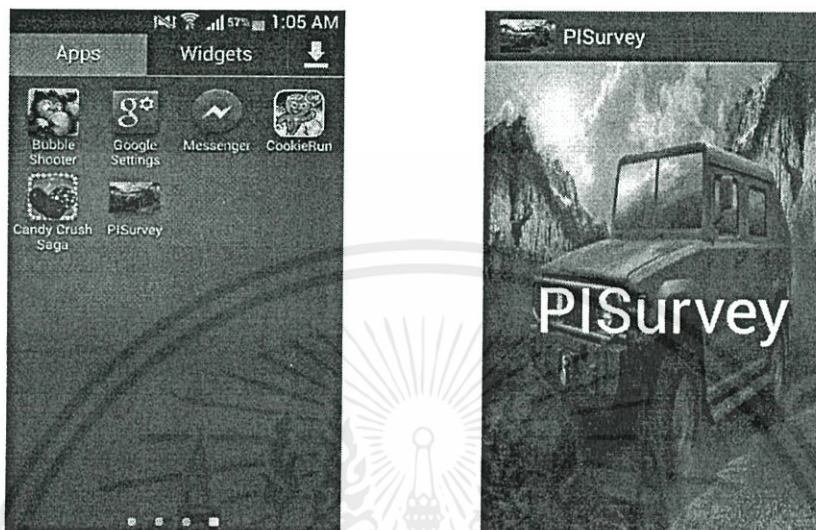
เขียนโปรแกรมควบคุมทิศทางการเคลื่อนที่ของรถบนสมองกลฝังตัวจากนั้นสั่งรันโปรแกรม จะเห็นได้ว่าฝั่ง Server ทำการอัปเดตค่าพอร์ต GPIO ที่ใช้ในการควบคุมการเคลื่อนที่ของรถอย่างต่อเนื่อง โดยแสดงสถานะการทำงานของแต่ละพอร์ตบนหน้าจอ LXTerminal หากข้อมูลที่ได้รับมาเป็นลอจิก 0 จะแสดงสถานะของพอร์ตว่า Turned off ดังรูปที่ 4.11

```
GPIO 17 Turned Off
GPIO 18 Turned Off
GPIO 22 Turned Off
GPIO 23 Turned Off
```

รูปที่ 4.11 หน้าจอ LXTerminal แสดงสถานะของพอร์ต GPIO

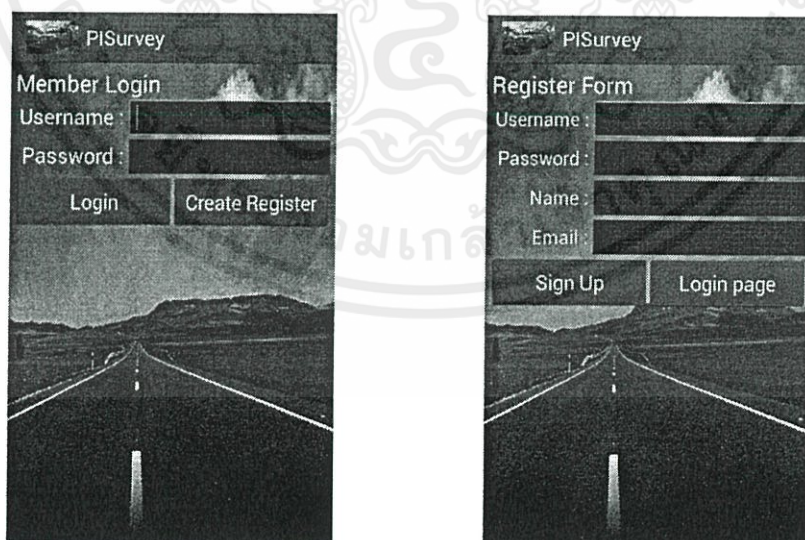
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปิดแอปพลิเคชันแอนดรอยด์ชื่อ PISurvey บนสมาร์ตโฟนจะปรากฏหน้าจอหลักของแอปพลิเคชันแสดงดังรูปที่ 4.12



รูปที่ 4.12 แอปพลิเคชันแอนดรอยด์บนสมาร์ตโฟนและหน้าจอหลักของแอปพลิเคชัน

จากนั้นทำการล็อกอินเข้าสู่ระบบถ้าไม่มี Username และ Password ต้องทำการสมัครสมาชิกก่อน โดยการกดปุ่ม Create Register จะปรากฏหน้าจอสำหรับป้อนข้อมูลสมัครสมาชิกแสดงดังรูปที่ 4.13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.13 หน้าจอสำหรับล็อกอินและหน้าจอสำหรับป้อนข้อมูลสมัครสมาชิก
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการสมัครสมาชิกแล้วข้อมูลที่ป้อนลงไปจะถูกบันทึกลงในฐานข้อมูลบนสมองกลฝังตัวแสดงดังรูปที่ 4.15 กดปุ่ม Login page จะกลับมาที่หน้าจอสำหรับล็อกอิน



รูปที่ 4.14 ป้อนข้อมูลสมัครสมาชิก

+ Options						
←T→	username	password	name	email	register_date	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Android	1234		cha@hotmail.com	2013-12-08 14:46:54	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Oil	555	Cha	cha@hotmail.com	2014-03-12 00:49:07	

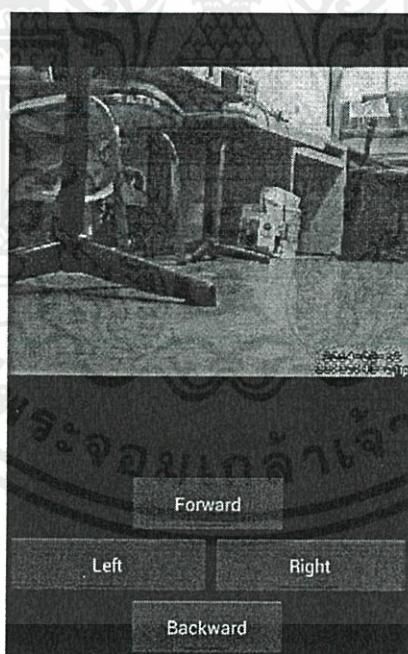
รูปที่ 4.15 ข้อมูลสมัครสมาชิกที่บันทึกในฐานข้อมูลบนสมองกลฝังตัว

หลังจากนั้นจึงทำการล็อกอินเข้าสู่หน้าจอสำหรับการเชื่อมต่อกับกล้อง จะมีช่องสำหรับป้อนไอพีแอดเดรสและพอร์ตของ Raspberry Pi ในการเชื่อมต่อกับกล้องแสดงดังรูปที่ 4.16 เมื่อกดปุ่ม Start ภาพจากกล้องเว็บแคมและปุ่มสำหรับควบคุมการเคลื่อนที่ของรถจะปรากฏบนหน้าแอปพลิเคชันแอนดรอยด์ดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 การล็อกอินเข้าสู่หน้าจอสำหรับการเชื่อมต่อกับกล้องเว็บแคม



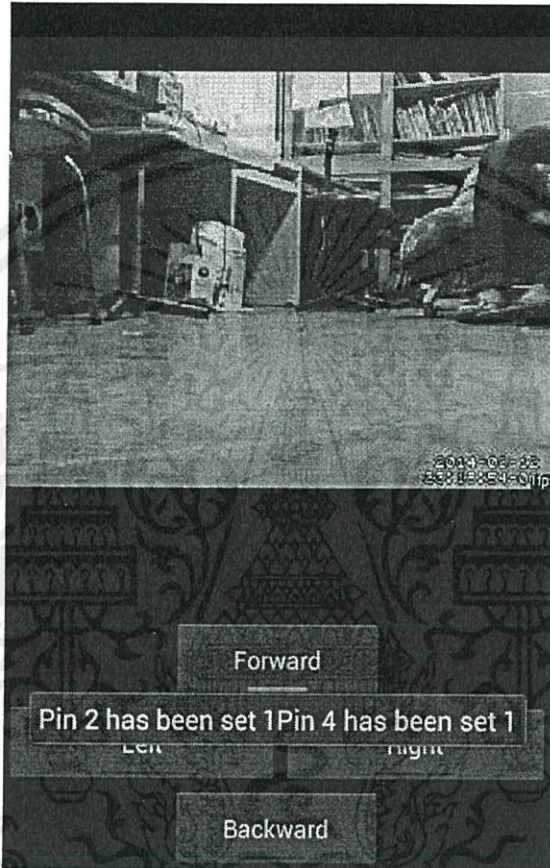
รูปที่ 4.17 ภาพจากกล้องเว็บแคมและปุ่มสำหรับควบคุมการเคลื่อนที่ของรถ

บนหน้าแอปพลิเคชันแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเคลื่อนที่ไปข้างหน้า

เมื่อทำการกดปุ่ม Forward บนหน้าแอปพลิเคชันจะแสดงข้อความขึ้นเตือนว่า “Pin 2 has been set 1 Pin 4 has been set 1” แสดงได้ดังรูปที่ 4.18 จากนั้นข้อมูลจะถูกส่งเข้าไปยังฐานข้อมูลบนสมองกลฝังตัวเป็นลอจิก 1 ซึ่งจะส่งผลให้รถสามารถเคลื่อนที่ไปข้างหน้าได้



รูปที่ 4.18 เมื่อทำการกดปุ่ม Forward บนหน้าแอปพลิเคชัน

ผลการทดลองแสดงได้ดังนี้

ข้อมูลที่รับมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) จะถูกบันทึกลงในฐานข้อมูลบนฝั่ง Server (สมองกลฝังตัว) ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ Options

←T→	pinID	pinNumber	pinStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	17	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	18	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	22	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	23	1

รูปที่ 4.19 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 18 และ 23 เป็นลอจิก 1

จากรูปที่ 4.19 จะเห็นว่าสถานะของ pinID 2 (พอร์ต 18) และ pinID 4 (พอร์ต 23) เปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ซึ่งแสดงถึงการทำงานของพอร์ต GPIO ที่ถูกตั้งค่าไว้เพื่อส่งไปควบคุมการทำงานของวงจรีเลย์สวิตซ์ต่อไป

สถานะการทำงานของพอร์ต GPIO จะแสดงบนหน้าจอ LXTerminal ดังรูปที่ 4.20

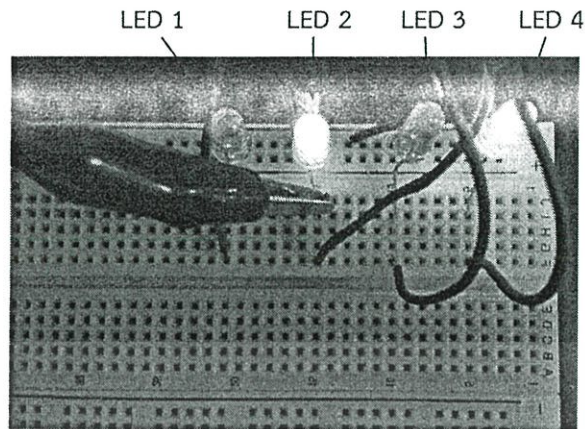
```
GPIO 17 Turned Off
GPIO 18 Turned On
GPIO 22 Turned Off
GPIO 23 Turned On
```

รูปที่ 4.20 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 18 และ 23

จากรูปที่ 4.20 จะเห็นว่าสมองกล้องตัวทำการประมวลผลค่าที่รับมาจากแอปพลิเคชันแอนดรอยด์ แล้วแสดงสถานะการทำงานของพอร์ต GPIO บนหน้าจอ LXTerminal ซึ่งพอร์ต 18 และ 23 เป็นลอจิก 1 แสดงสถานะเป็น Turned On ส่วนพอร์ต 17 และ 22 เป็นลอจิก 0 แสดงสถานะเป็น Turned Off

จากการส่งข้อมูลจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) มายังฝั่ง Server (สมองกล้องตัว) แสดงสถานะการทำงานของพอร์ต GPIO โดยใช้ LED ดังรูปที่ 4.21

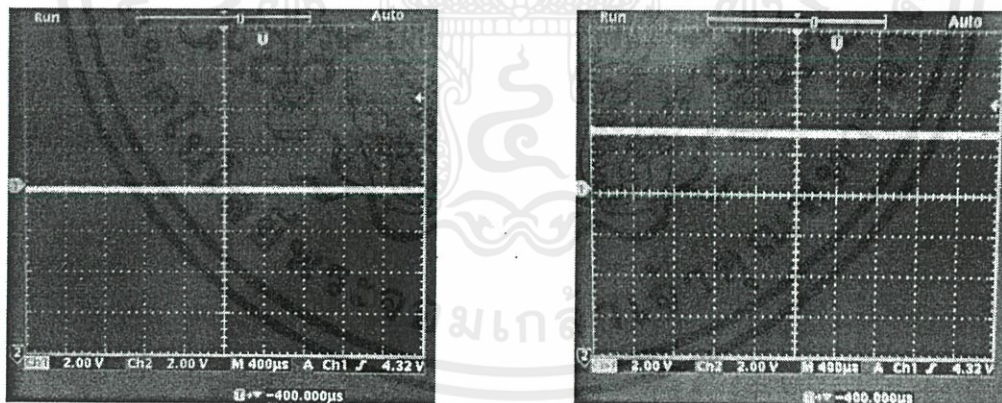
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 สถานะการทำงานของพอร์ต 18 และ 23

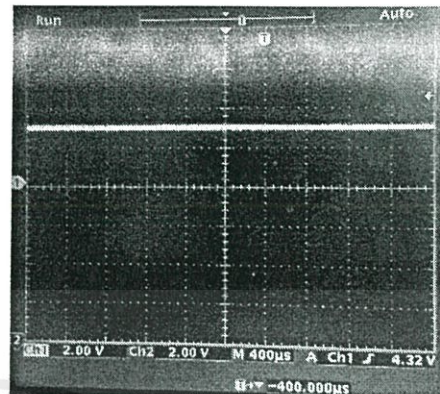
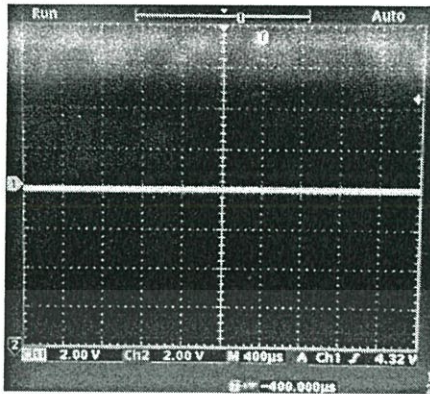
จากรูปที่ 4.21 LED ตัวที่ 2 ที่ต่ออยู่กับพอร์ต 18 และ LED ตัวที่ 4 ที่ต่ออยู่กับพอร์ต 23 ติดทั้งสองดวง แสดงให้เห็นว่ามีสัญญาณเอาต์พุตออกจากพอร์ต GPIO

จากนั้นทำการวัดสัญญาณเอาต์พุตที่ออกจากพอร์ต GPIO โดยใช้ Oscilloscope จะเห็นว่าสัญญาณเอาต์พุตที่ออกจากพอร์ต 18 และ 23 วัดสัญญาณได้ 3 โวลต์ แสดงว่าพอร์ต 18 และ 23 เกิดการทำงานดังรูปที่ 4.23 และรูปที่ 4.25 ส่วนสัญญาณเอาต์พุตที่ออกจากพอร์ต 17 และ 22 วัดสัญญาณได้ 0 โวลต์ แสดงว่าพอร์ต 17 และ 22 ไม่ได้ถูกใช้งานอยู่ดังรูปที่ 4.22 และรูปที่ 4.24



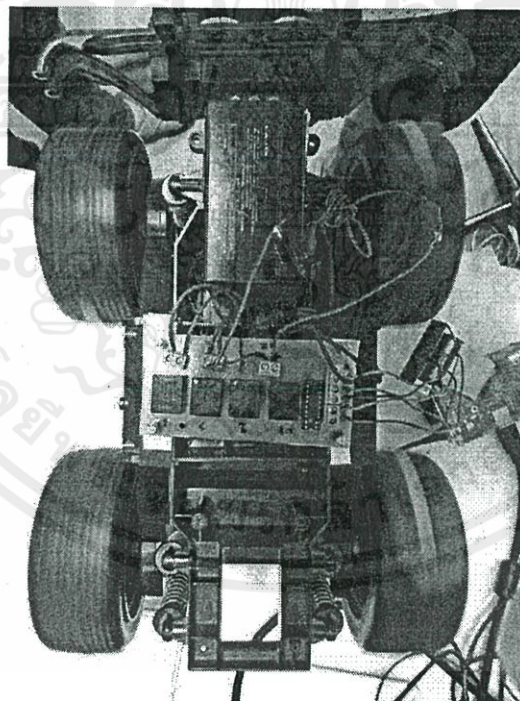
รูปที่ 4.22 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17 รูปที่ 4.23 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22 รูปที่ 4.25 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23

จากสถานะของ LED ตัวที่ 2 ที่ต่ออยู่กับพอร์ต 18 และ LED ตัวที่ 4 ที่ต่ออยู่กับพอร์ต 23 รวมทั้งสัญญาณเอาต์พุตที่ออกจากพอร์ต 18 และ 23 แสดงให้เห็นว่ารถกำลังเคลื่อนที่ไปข้างหน้าดังรูปที่ 4.26

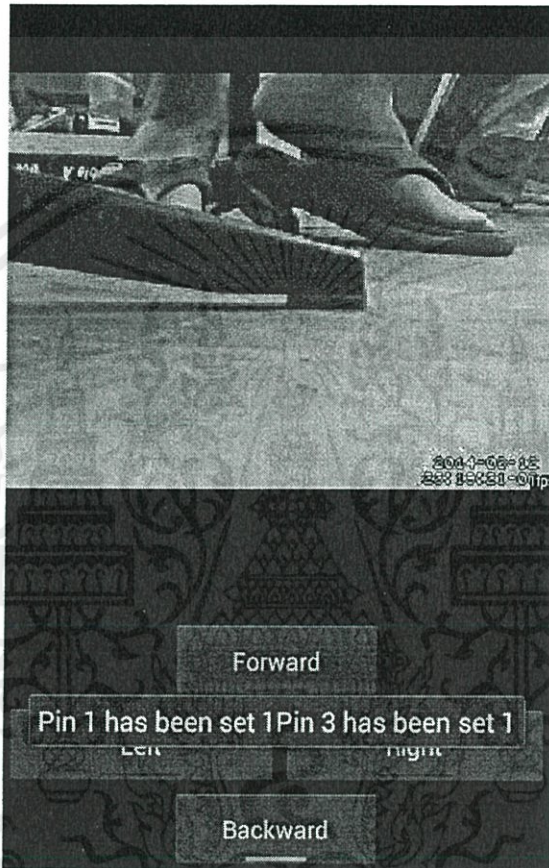


รูปที่ 4.26 รถกำลังเคลื่อนที่ไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเคลื่อนที่ไปข้างหลัง

เมื่อทำการกดปุ่ม Backward บนหน้าแอปพลิเคชันจะแสดงข้อความขึ้นเตือนว่า “Pin 1 has been Set 1 Pin 3 has been set 1” แสดงได้ดังรูปที่ 4.27 จากนั้นข้อมูลจะถูกส่งเข้าไปยังฐานข้อมูลบนสมองกลฝังตัวเป็นลอจิก 1 ซึ่งจะส่งผลให้รถสามารถเคลื่อนที่ไปข้างหลังได้



รูปที่ 4.27 เมื่อทำการกดปุ่ม Backward บนหน้าแอปพลิเคชัน

ผลการทดลองแสดงได้ดังนี้

ข้อมูลที่รับมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) จะถูกบันทึกลงในฐานข้อมูลบนฝั่ง Server (สมองกลฝังตัว) ดังรูปที่ 4.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ Options

	pinID	pinNumber	pinStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	17	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	18	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	22	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	23	0

รูปที่ 4.28 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 17 และ 22 เป็นลอจิก 1

จากรูปที่ 4.28 จะเห็นว่าสถานะของ pinID 1 (พอร์ต 17) และ pinID 3 (พอร์ต 22) เปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ซึ่งแสดงถึงการทำงานของพอร์ต GPIO ที่ถูกตั้งค่าไว้เพื่อส่งไปควบคุมการทำงานของวงจรีเลย์สวิตซ์ต่อไป

สถานะการทำงานของพอร์ต GPIO จะแสดงบนหน้าจอ LXTerminal ดังรูปที่ 4.29

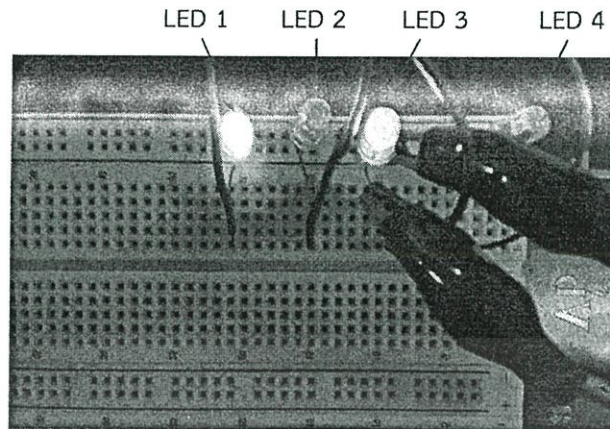
```
GPIO 17 Turned On
GPIO 18 Turned Off
GPIO 22 Turned On
GPIO 23 Turned Off
```

รูปที่ 4.29 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 17 และ 22

จากรูปที่ 4.29 จะเห็นว่าสมองกล้องตัวทำการประมวลผลค่าที่รับมาจากแอปพลิเคชันแอนดรอยด์ แล้วแสดงสถานะการทำงานของพอร์ต GPIO บนหน้าจอ LXTerminal ซึ่งพอร์ต 17 และ 22 เป็นลอจิก 1 แสดงสถานะเป็น Turned On ส่วนพอร์ต 18 และ 23 เป็นลอจิก 0 แสดงสถานะเป็น Turned Off

จากการส่งข้อมูลจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) มายังฝั่ง Server (สมองกล้องตัว) แสดงสถานะการทำงานของพอร์ต GPIO โดยใช้ LED ดังรูปที่ 4.30

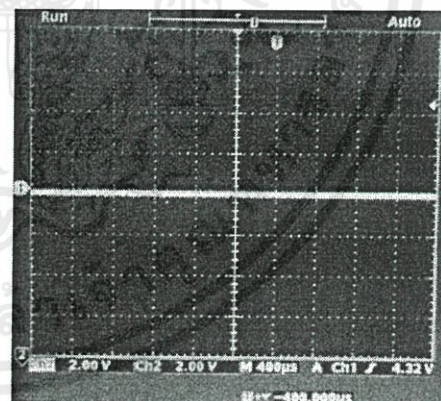
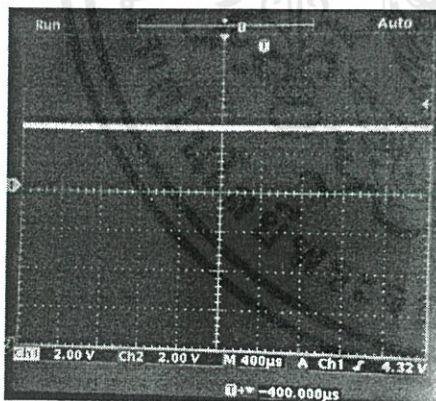
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 สถานะการทำงานของพอร์ต 17 และ 22

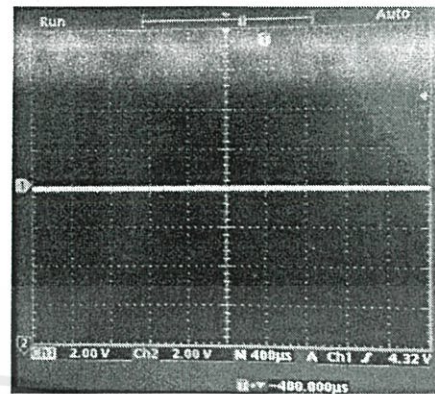
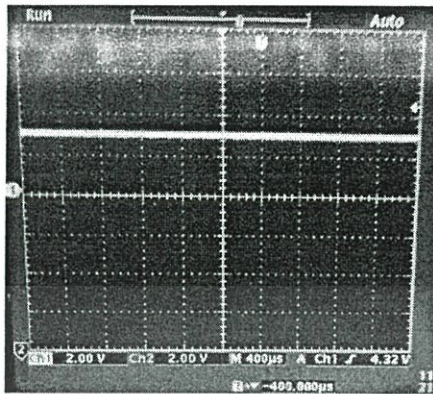
จากรูปที่ 4.30 LED ตัวที่ 1 ที่ต่ออยู่กับพอร์ต 17 และ LED ตัวที่ 3 ที่ต่ออยู่กับพอร์ต 22 ติดทั้งสองดวง แสดงให้เห็นว่ามีสัญญาณเอาต์พุตออกจากพอร์ต GPIO

จากนั้นทำการวัดสัญญาณเอาต์พุตที่ออกจากพอร์ต GPIO โดยใช้ Oscilloscope จะเห็นว่าสัญญาณเอาต์พุตที่ออกจากพอร์ต 17 และ 22 วัดสัญญาณได้ 3 โวลต์ แสดงว่าพอร์ต 17 และ 22 เกิดการทำงานดังรูปที่ 4.31 และรูปที่ 4.33 ส่วนสัญญาณเอาต์พุตที่ออกจากพอร์ต 18 และ 23 วัดสัญญาณได้ 0 โวลต์ แสดงว่าพอร์ต 18 และ 23 ไม่ได้ถูกใช้งานอยู่ดังรูปที่ 4.32 และรูปที่ 4.34



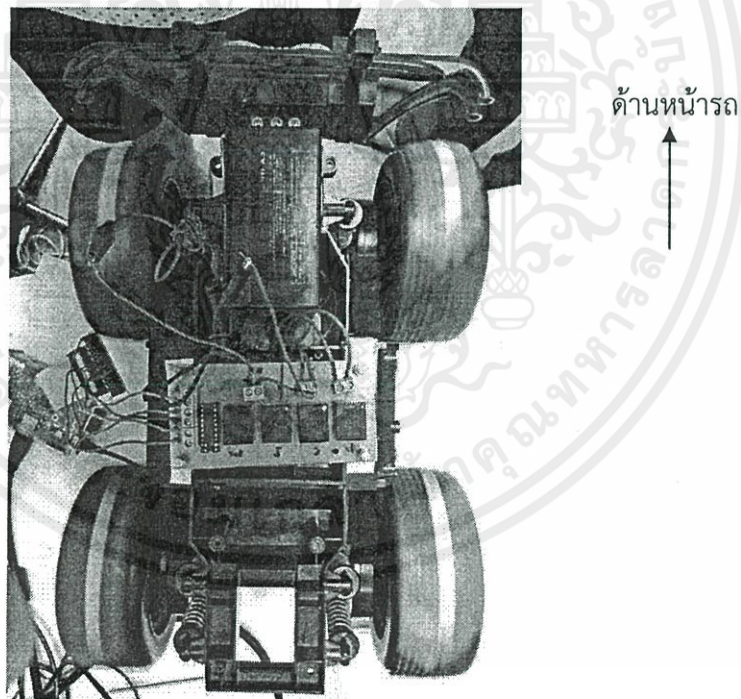
รูปที่ 4.31 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17 รูปที่ 4.32 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22 รูปที่ 4.34 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23

จากสถานะของ LED ตัวที่ 1 ที่ต่ออยู่กับพอร์ต 17 และ LED ตัวที่ 3 ที่ต่ออยู่กับพอร์ต 22 รวมทั้งสัญญาณเอาต์พุตที่ออกจากพอร์ต 17 และ 22 แสดงให้เห็นว่ารถกำลังเคลื่อนที่ไปข้างหลังดังรูปที่ 4.35

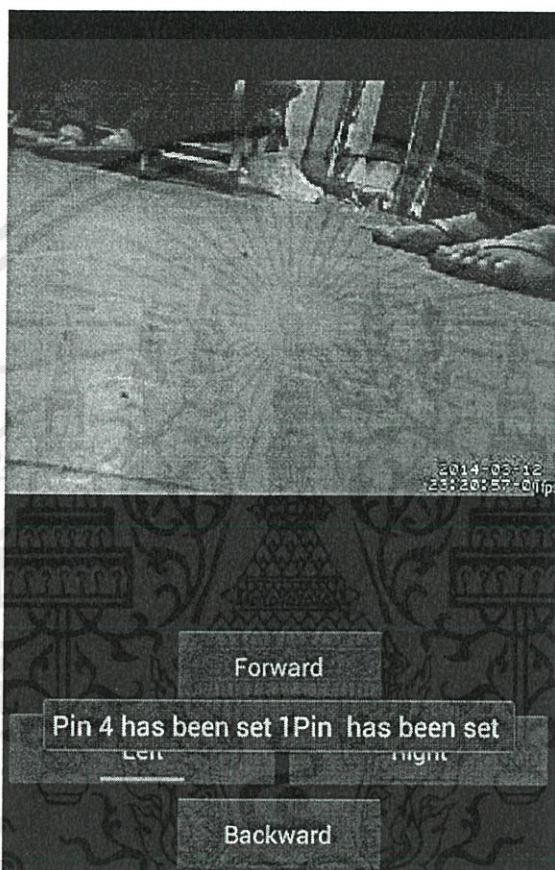


รูปที่ 4.35 รถกำลังเคลื่อนที่ไปข้างหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเคลื่อนซ้าย

เมื่อทำการกดปุ่ม Left บนหน้าแอปพลิเคชันจะแสดงข้อความขึ้นเตือนว่า “Pin 4 has been set 1” แสดงได้ดังรูปที่ 4.36 จากนั้นข้อมูลจะถูกส่งเข้าไปยังฐานข้อมูลบนสมองกลฝั่งตัวเป็นลอจิก 1 ซึ่งจะส่งผลให้รถสามารถเคลื่อนซ้ายได้



รูปที่ 4.36 เมื่อทำการกดปุ่ม Left บนหน้าแอปพลิเคชัน

ผลการทดลองแสดงได้ดังนี้

ข้อมูลที่รับมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) จะถูกบันทึกลงในฐานข้อมูลบนฝั่ง Server (สมองกลฝั่งตัว) ดังรูปที่ 4.37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ Options

	pinID	pinNumber	pinStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	17	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	18	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	22	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	23	1

รูปที่ 4.37 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 23 เป็นลอจิก 1

จากรูปที่ 4.37 จะเห็นว่าสถานะของ pinID 4 (พอร์ต 23) เปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ซึ่งแสดงถึงการทำงานของพอร์ต GPIO ที่ถูกตั้งค่าไว้เพื่อส่งไปควบคุมการทำงานของวงจรรีเลย์สวิตช์ต่อไป

สถานะการทำงานของพอร์ต GPIO จะแสดงบนหน้าจอ LXTerminal ดังรูปที่ 4.38

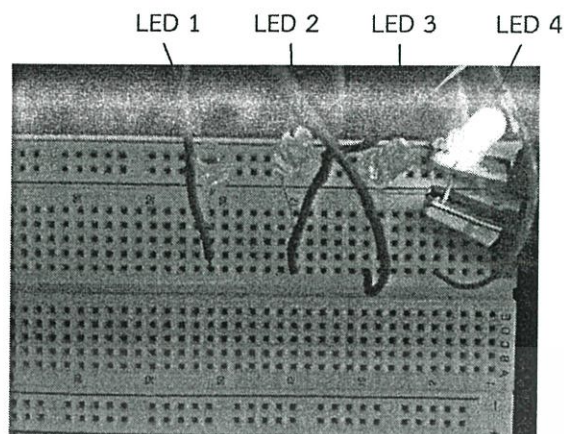
```
GPIO 17 Turned Off
GPIO 18 Turned Off
GPIO 22 Turned Off
GPIO 23 Turned On
```

รูปที่ 4.38 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 23

จากรูปที่ 4.38 จะเห็นว่าสมองกลฝังตัวทำการประมวลผลค่าที่รับมาจากแอปพลิเคชันแอนดรอยด์ แล้วแสดงสถานะการทำงานของพอร์ต GPIO บนหน้าจอ LXTerminal ซึ่งพอร์ต 23 เป็นลอจิก 1 แสดงสถานะเป็น Turned On ส่วนพอร์ต 17 18 และ 22 เป็นลอจิก 0 แสดงสถานะเป็น Turned Off

จากการส่งข้อมูลจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) มายังฝั่ง Server (สมองกลฝังตัว) แสดงสถานะการทำงานของพอร์ต GPIO โดยใช้ LED ดังรูปที่ 4.39

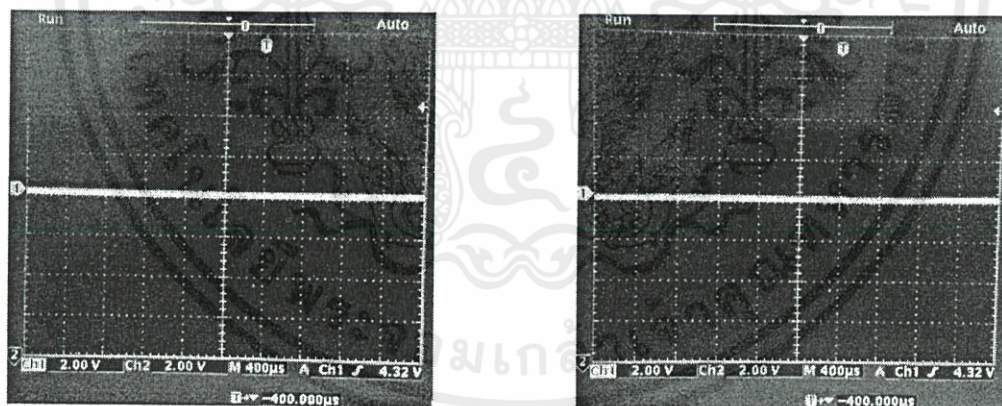
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 สถานะการทำงานของพอร์ต 23

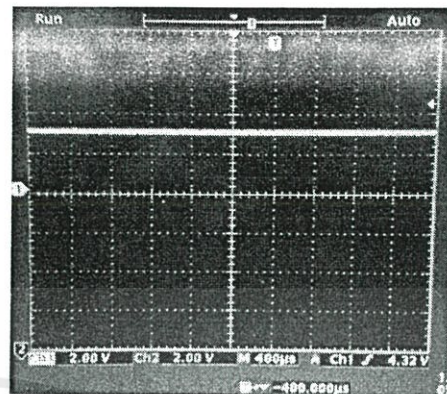
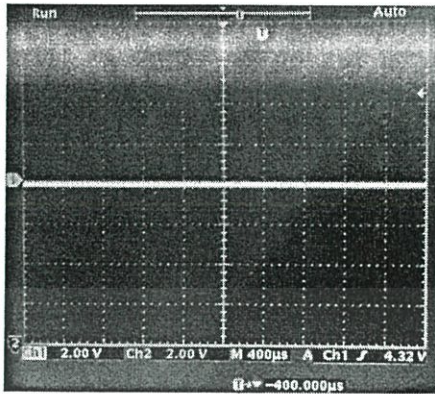
จากรูปที่ 4.39 LED ตัวที่ 4 ที่ต่ออยู่กับพอร์ต 23 ติดเพียงดวงเดียว แสดงให้เห็นว่ามีสัญญาณเอาต์พุตออกจากพอร์ต GPIO

จากนั้นทำการวัดสัญญาณเอาต์พุตที่ออกจากพอร์ต GPIO โดยใช้ Oscilloscope จะเห็นว่าสัญญาณเอาต์พุตที่ออกจากพอร์ต 23 วัดสัญญาณได้ 3 โวลต์ แสดงว่าพอร์ต 23 เกิดการทำงานดังรูปที่ 4.43 ส่วนสัญญาณเอาต์พุตที่ออกจากพอร์ต 17 18 และ 22 วัดสัญญาณได้ 0 โวลต์ แสดงว่าพอร์ต 17 18 และ 22 ไม่ได้ถูกใช้งานอยู่ดังรูปที่ 4.40 รูปที่ 4.41 และรูปที่ 4.42



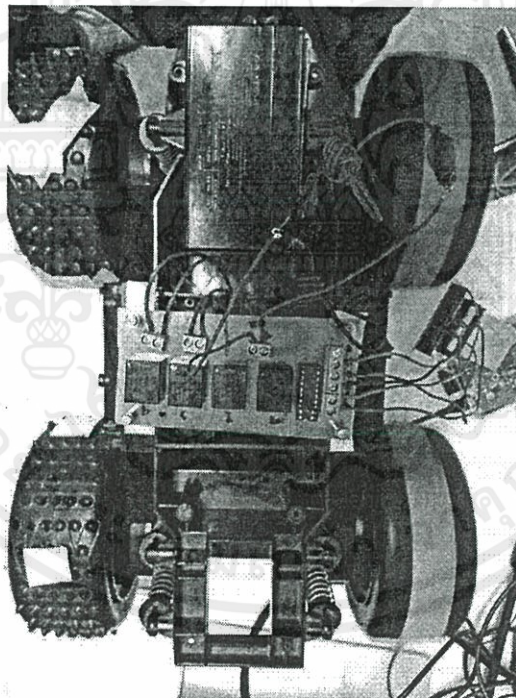
รูปที่ 4.40 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17 รูปที่ 4.41 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22 รูปที่ 4.43 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23

จากสถานะของ LED ตัวที่ 4 ที่ต่ออยู่กับพอร์ต 23 รวมทั้งสัญญาณเอาต์พุตที่ออกจากพอร์ต 23 แสดงให้เห็นว่ารถกำลังเลี้ยวซ้ายดังรูปที่ 4.44

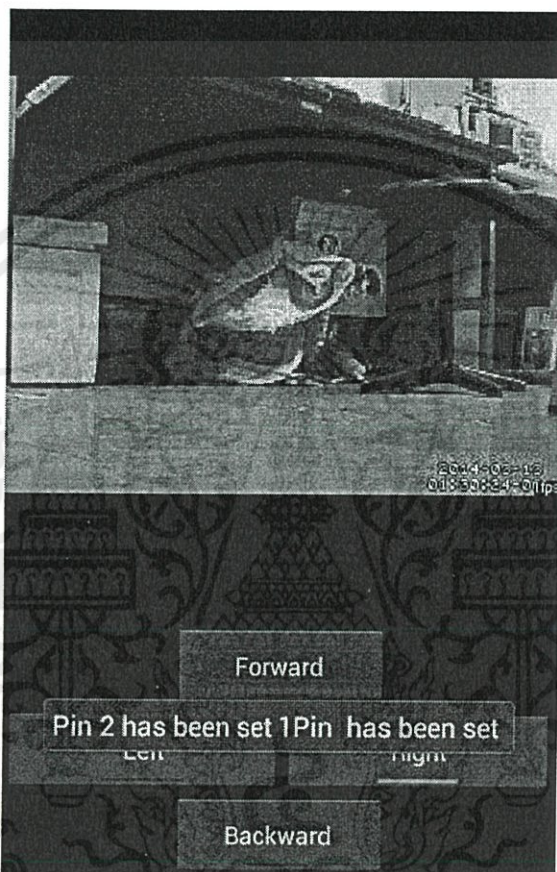


รูปที่ 4.44 รถกำลังเลี้ยวซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การทดลองกดปุ่มบนแอปพลิเคชันให้รถเลี้ยวขวา

เมื่อทำการกดปุ่ม Right บนหน้าแอปพลิเคชันจะแสดงข้อความขึ้นเตือนว่า “Pin 2 has been set 1” แสดงได้ดังรูปที่ 4.45 จากนั้นข้อมูลจะถูกส่งเข้าไปยังฐานข้อมูลบนสมองกลฝังตัวเป็นลอจิก 1 ซึ่งจะส่งผลให้รถสามารถเลี้ยวขวาได้



รูปที่ 4.45 เมื่อทำการกดปุ่ม Right บนหน้าแอปพลิเคชัน

ผลการทดลองแสดงได้ดังนี้

ข้อมูลที่รับมาจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) จะถูกบันทึกลงในฐานข้อมูลบนฝั่ง Server (สมองกลฝังตัว) ดังรูปที่ 4.46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+ Options

	pinID	pinNumber	pinStatus
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	17	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	18	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	22	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	23	0

รูปที่ 4.46 ค่าที่รับเข้ามาจากแอปพลิเคชันแสดงสถานะของพอร์ต 18 เป็นลอจิก 1

จากรูปที่ 4.46 จะเห็นว่าสถานะของ pinID 2 (พอร์ต 18) เปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ซึ่งแสดงถึงการทำงานของพอร์ต GPIO ที่ถูกตั้งค่าไว้เพื่อส่งไปควบคุมการทำงานของวงจรรีเลย์สวิตซ์ต่อไป

สถานะการทำงานของพอร์ต GPIO จะแสดงบนหน้าจอ LXTerminal ดังรูปที่ 4.47

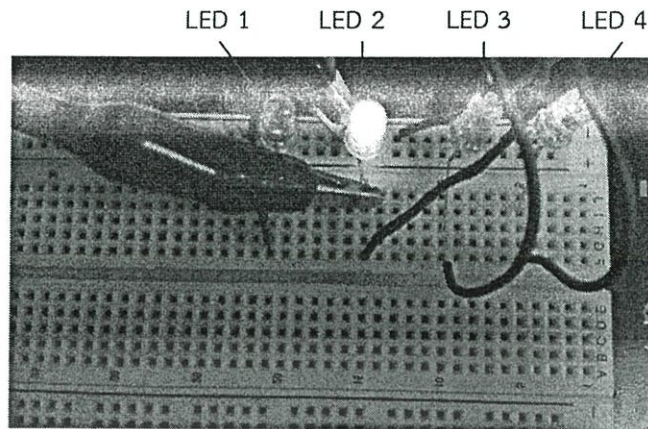
```
GPIO 17 Turned Off
GPIO 18 Turned On
GPIO 22 Turned Off
GPIO 23 Turned Off
```

รูปที่ 4.47 หน้าจอ LXTerminal แสดงสถานะการทำงานของพอร์ต 18

จากรูปที่ 4.47 จะเห็นว่าสมองกล้องฝังตัวทำการประมวลผลค่าที่รับมาจากแอปพลิเคชันแอนดรอยด์ แล้วแสดงสถานะการทำงานของพอร์ต GPIO บนหน้าจอ LXTerminal ซึ่งพอร์ต 18 เป็นลอจิก 1 แสดงสถานะเป็น Turned On ส่วนพอร์ต 17 22 และ 23 เป็นลอจิก 0 แสดงสถานะเป็น Turned Off

จากการส่งข้อมูลจากฝั่ง Client (แอปพลิเคชันแอนดรอยด์) มายังฝั่ง Server (สมองกล้องฝังตัว) แสดงสถานะการทำงานของพอร์ต GPIO โดยใช้ LED ดังรูปที่ 4.48

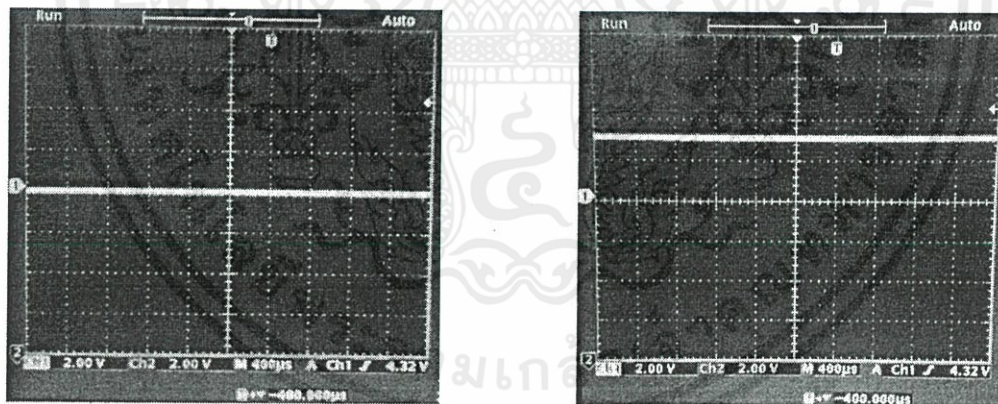
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.48 สถานะการทำงานของพอร์ต 18

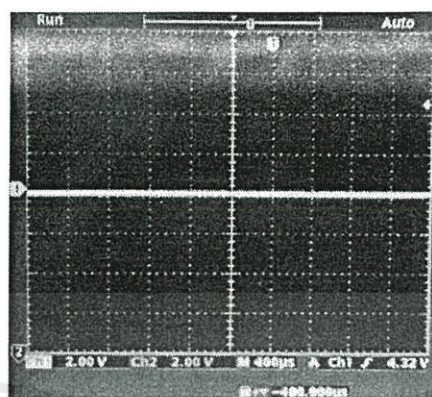
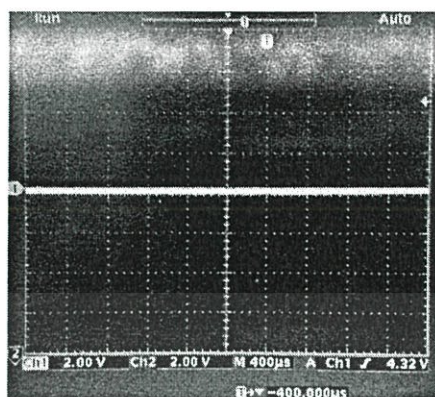
จากรูปที่ 4.48 LED ตัวที่ 2 ที่ต่ออยู่กับพอร์ต 18 ติดเพียงดวงเดียว แสดงให้เห็นว่ามีสัญญาณเอาต์พุตออกจากพอร์ต GPIO

จากนั้นทำการวัดสัญญาณเอาต์พุตที่ออกจากพอร์ต GPIO โดยใช้ Oscilloscope จะเห็นว่าสัญญาณเอาต์พุตที่ออกจากพอร์ต 18 วัดสัญญาณได้ 3 โวลต์ แสดงว่าพอร์ต 18 เกิดการทำงานดังรูปที่ 4.50 ส่วนสัญญาณเอาต์พุตที่ออกจากพอร์ต 17 22 และ 23 วัดสัญญาณได้ 0 โวลต์ แสดงว่าพอร์ต 17 22 และ 23 ไม่ได้ถูกใช้งานอยู่ดังรูปที่ 4.49 รูปที่ 4.51 และรูปที่ 4.52



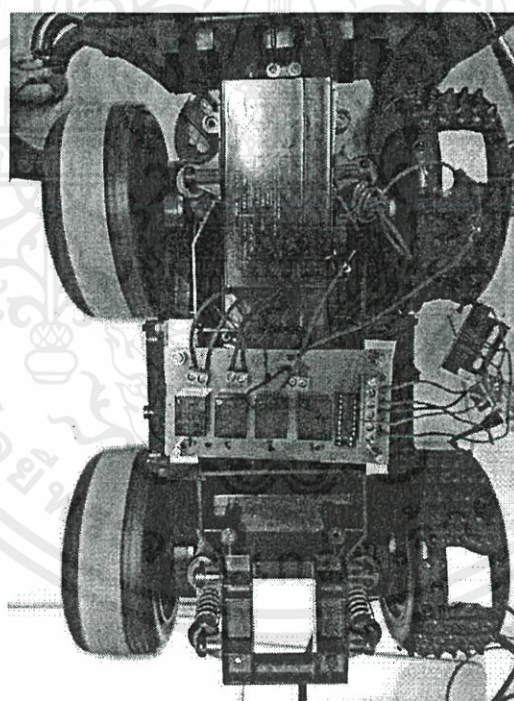
รูปที่ 4.49 สัญญาณเอาต์พุตที่ออกจากพอร์ต 17 รูปที่ 4.50 สัญญาณเอาต์พุตที่ออกจากพอร์ต 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



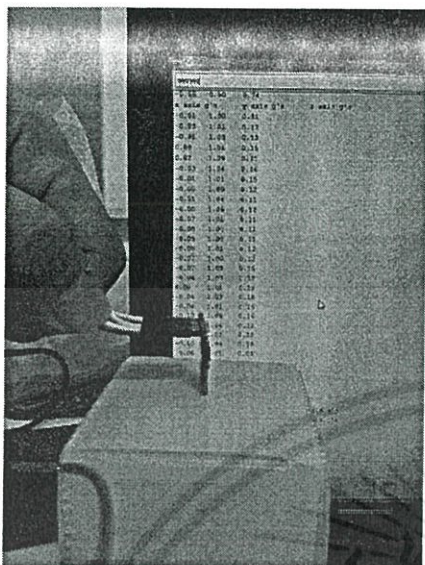
รูปที่ 4.51 สัญญาณเอาต์พุตที่ออกจากพอร์ต 22 รูปที่ 4.52 สัญญาณเอาต์พุตที่ออกจากพอร์ต 23

จากสถานะของ LED ตัวที่ 2 ที่ต่ออยู่กับพอร์ต 18 รวมทั้งสัญญาณเอาต์พุตที่ออกจากพอร์ต 23 แสดงให้เห็นว่ารถกำลังเลี้ยวขวา ดังรูปที่ 4.53



รูปที่ 4.53 รถกำลังเลี้ยวขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



BlueTooth :Device connected

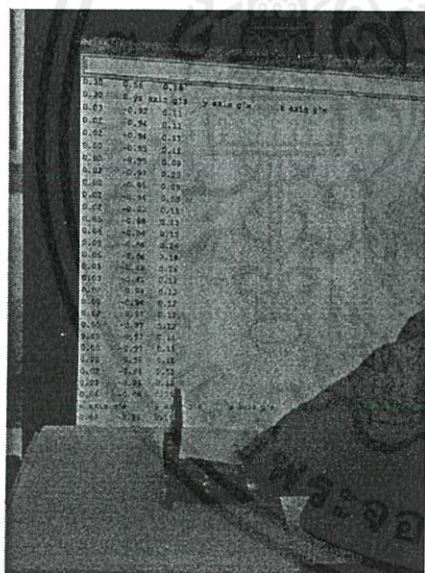
Message:

-0.67,85.93

X axis:-0.67

Y axis:85.93

รูปที่ 4.58 การเอียงในแนวแกน +Y และค่าความเอียงที่แสดงบนแอปพลิเคชัน



BlueTooth :Device connected

Message:

-1.14,-84.82

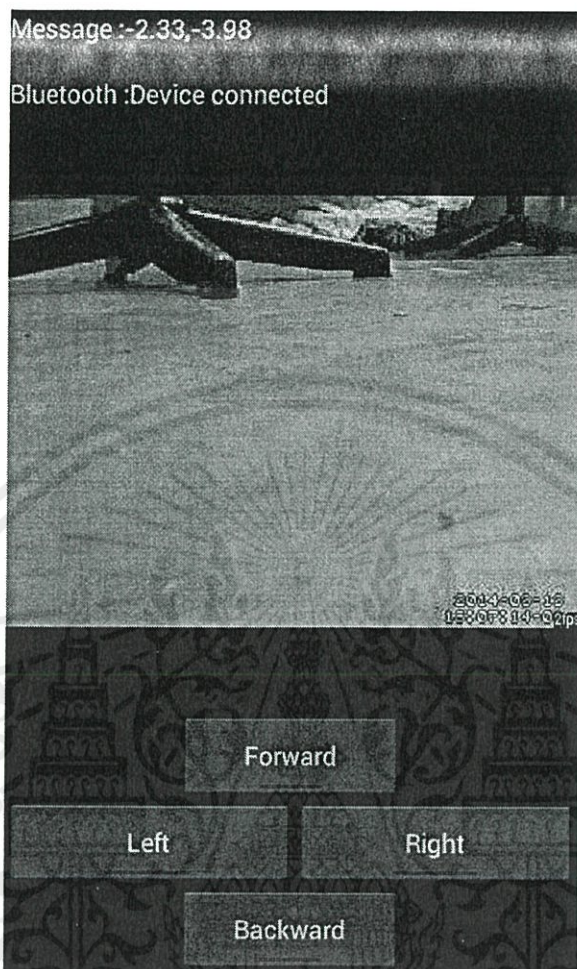
X axis:-1.14

Y axis:-84.82

รูปที่ 4.59 การเอียงในแนวแกน -Y และค่าความเอียงที่แสดงบนแอปพลิเคชัน

จากรูปที่ 4.58 การเอียงในแนวแกน +Y แสดงการเอียงของ Accelerometer Sensor ไปข้างหน้าและรูปที่ 4.59 การเอียงในแนวแกน -Y แสดงการเอียงของ Accelerometer Sensor ไปข้างหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.60 หน้าจอแอปพลิเคชัน

จากรูปที่ 4.60 จะเห็นว่าหน้าจอการทำงานของแอปพลิเคชันการบังคับรถตรวจการณ์ แสดงค่าความเอียงของตัวรถพร้อมกับการบังคับด้วยปุ่ม Forward Backward Left และ Right และยังแสดงภาพจากกล้องที่ติดตั้งบนตัวรถเพื่อให้เห็นเส้นทางการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลและข้อเสนอแนะ

ปริญญานิพนธ์นี้นำเสนอการตรวจการณโดยใช้สมองกลฝังตัวควบคุมด้วย Smartphone โดยศึกษาและออกแบบรถตรวจการณให้สามารถควบคุมการเคลื่อนที่ผ่านทาง แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ สามารถแสดงภาพและค่าความเอียงให้แสดงผ่าน หน้าจอ Smartphone

5.1 สรุปผล

จากการศึกษาเกี่ยวกับการออกแบบและสร้างรถตรวจการณโดยใช้สมองกลฝังตัว (Raspberry Pi) ควบคุมด้วย Smartphone ซึ่งประกอบด้วย 2 ส่วน คือส่วนของซอฟต์แวร์พบว่า Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กที่สามารถสร้างเป็นเซิร์ฟเวอร์ด้วยโปรแกรม Apache PHP และสร้างฐานข้อมูลด้วยโปรแกรม MySQL รวมทั้งใช้โปรแกรม phpMyAdmin เป็นตัวจัดการ ฐานข้อมูล มีการออกแบบและสร้างแอปพลิเคชันด้วยโปรแกรมอีคลิปส์ (Eclipse) โดยใช้ภาษา Java ในส่วนของฮาร์ดแวร์ทำการออกแบบและสร้างวงจรรีเลย์รวมทั้งติดตั้งกล้องและเซนเซอร์วัด ความเอียง เมื่อสร้างองค์ประกอบทั้งสองส่วนเสร็จและนำมาประกอบกันเพื่อให้ได้รถตรวจการณที่ สามารถควบคุมให้เคลื่อนที่โดยที่สามารถแสดงภาพและความเอียงของรถขณะเคลื่อนที่ โดยที่ อุปกรณ์ทั้งหมดสามารถติดต่อสื่อสารกันได้ผ่านเครือข่ายอินเทอร์เน็ตแบบไร้สาย

5.2 ข้อเสนอแนะ

ในการทำปริญญานิพนธ์นี้ได้นำเสนอการตรวจการณโดยใช้สมองกลฝังตัวควบคุมด้วย Smartphone มีส่วนประกอบ 2 ส่วน คือส่วนของซอฟต์แวร์และส่วนฮาร์ดแวร์ ซึ่งส่วนประกอบทั้งสอง สามารถพัฒนาให้มีประสิทธิภาพสูงขึ้นได้โดย

- 1) เชื่อมต่อกับ GPS เพื่อให้สามารถควบคุมรถตรวจการณได้ในระยะทางที่ไกลขึ้น และให้สามารถค้นหาเส้นทางเพื่อไปยังจุดหมายที่ต้องการได้
- 2) ใช้ socket ซึ่งเป็นการเขียนโปรแกรมให้ระบบสมองกลฝังตัวและแอปพลิเคชัน สามารถสื่อสารกันได้ผ่านทางเครือข่ายอินเทอร์เน็ต
- 3) ติดตั้งไมโครโฟนให้รถตรวจการณสามารถส่งเสียงผ่านทางแอปพลิเคชันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ดร.จักรชัย โสอินทร์, และพงษ์ศธร จันทรียอย. *Basic Android App Development*. พิมพ์ครั้งที่ 1. นนทบุรี : บริษัท ไอดีซี พรีเมียร์ จำกัด, 2554.
- [2] ดร.จักรชัย โสอินทร์, พงษ์ศธร จันทรียอย, และณัฐธิดา วีระมงคลเลิศ. *คู่มือพัฒนาแอปพลิเคชัน Android*. พิมพ์ครั้งที่ 1. นนทบุรี : บริษัท ไอดีซี พรีเมียร์ จำกัด, 2555.
- [3] สุรี พงศาสกุลชัย, ภาณุวัฒน์ บุญผาสุก. *คัมภีร์ JAVA เล่ม 2*. พิมพ์ครั้งที่ 2. สำนักพิมพ์ เคทีพี จำกัด, 2548.
- [4] Raspberry Pi Web Server. [ออนไลน์]. เข้าถึงได้จาก:
<http://www.instructables.com/id/Raspberry-Pi-Web-Server/?ALLSTEPS>
- [5] Raspberry Pi Webcam Server Tutorial (Live Stream). [ออนไลน์]. เข้าถึงได้จาก:
<http://techspect.co.uk/2013/how-to-raspberry-pi-webcam-server-stream/>
- [6] arduino ติดต่อ ADXL335. [ออนไลน์]. เข้าถึงได้จาก:
<http://www.circuitshops.com/articles/42086693/arduino-ติดต่อ-ADXL335.htmlg>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES

- 3-axis sensing
- Small, low profile package
4 mm × 4 mm × 1.45 mm LFCSP
- Low power : 350 μ A (typical)
- Single-supply operation: 1.8 V to 3.6 V
- 10,000 g shock survival
- Excellent temperature stability
- BW adjustment with a single capacitor per axis
- RoHS/WEEE lead-free compliant

APPLICATIONS

- Cost sensitive, low power, motion- and tilt-sensing applications
- Mobile devices
- Gaming systems
- Disk drive protection
- Image stabilization
- Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of $\pm 3 g$. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_x , C_y , and C_z capacitors at the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

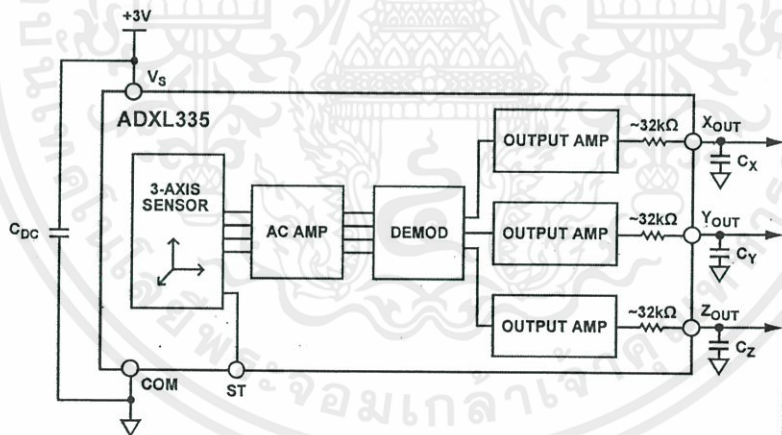


Figure 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rev. 0

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700
Fax: 781.461.3113

www.analog.com

©2009 Analog Devices, Inc. All rights reserved.

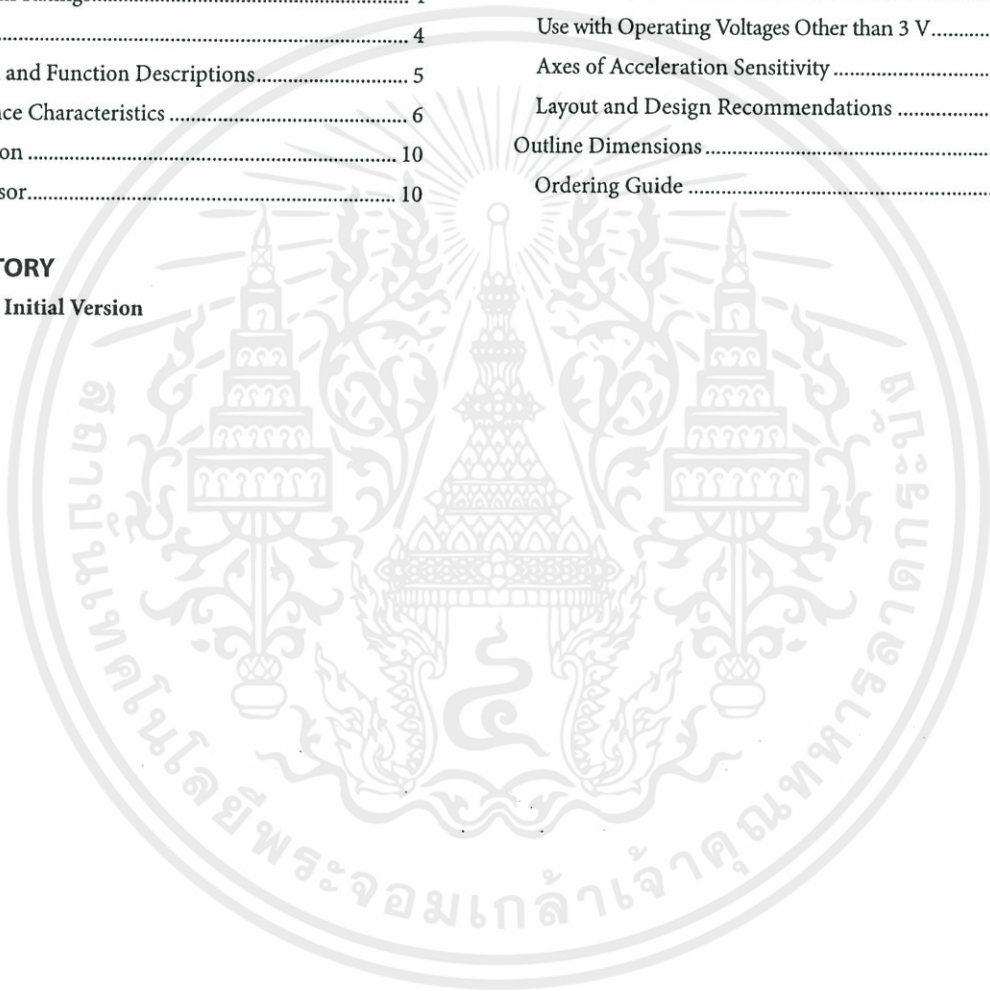
ADXL335

TABLE OF CONTENTS

Features	1	Performance	10
Applications.....	1	Applications Information	11
General Description.....	1	Power Supply Decoupling	11
Functional Block Diagram	1	Setting the Bandwidth Using C_x , C_y , and C_z	11
Revision History	2	Self Test	11
Specifications.....	3	Design Trade-Offs for Selecting Filter Characteristics: The Noise/BW Trade-Off.....	11
Absolute Maximum Ratings.....	4	Use with Operating Voltages Other than 3 V.....	11
ESD Caution.....	4	Axes of Acceleration Sensitivity	12
Pin Configuration and Function Descriptions.....	5	Layout and Design Recommendations	13
Typical Performance Characteristics	6	Outline Dimensions	14
Theory of Operation	10	Ordering Guide	14
Mechanical Sensor.....	10		

REVISION HISTORY

1/09—Revision 0: Initial Version



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 3\text{ V}$, $C_X = C_Y = C_Z = 0.1\ \mu\text{F}$, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis	± 3	± 3.6		g
Nonlinearity	% of full scale		± 0.3		%
Package Alignment Error			± 1		Degrees
Interaxis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ¹			± 1		%
SENSITIVITY (RATIOMETRIC)²					
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	Each axis $V_S = 3\text{ V}$	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	$V_S = 3\text{ V}$		± 0.01		%/ $^\circ\text{C}$
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X_{OUT} , Y_{OUT}	$V_S = 3\text{ V}$	1.35	1.5	1.65	V
0 g Voltage at Z_{OUT}	$V_S = 3\text{ V}$	1.2	1.5	1.8	V
0 g Offset vs. Temperature			± 1		mg/ $^\circ\text{C}$
NOISE PERFORMANCE					
Noise Density X_{OUT} , Y_{OUT}			150		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
Noise Density Z_{OUT}			300		$\mu\text{g}/\sqrt{\text{Hz}}$ rms
FREQUENCY RESPONSE⁴					
Bandwidth X_{OUT} , Y_{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z_{OUT} ⁵	No external filter		550		Hz
R_{FILT} Tolerance			$32 \pm 15\%$		k Ω
Sensor Resonant Frequency			5.5		kHz
SELF-TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3\text{ V}$		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	$^\circ\text{C}$

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S .

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X , C_Y , C_Z).

⁵ Bandwidth with external capacitors = $1/(2 \times \pi \times 32\text{ k}\Omega \times C)$. For C_X , $C_Y = 0.003\ \mu\text{F}$, bandwidth = 1.6 kHz. For $C_Z = 0.01\ \mu\text{F}$, bandwidth = 500 Hz. For C_X , C_Y , $C_Z = 10\ \mu\text{F}$, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S .

⁷ Turn on time is dependent on C_X , C_Y , C_Z and is approximately $160 \times C_X$ or C_Y or $C_Z + 1\text{ ms}$, where C_X , C_Y , C_Z are in microfarads (μF).

ADXL335

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 g
Acceleration (Any Axis, Powered)	10,000 g
V _s	-0.3 V to +3.6 V
All Other Pins	(COM - 0.3 V) to (V _s + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	-55°C to +125°C
Temperature Range (Storage)	-65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

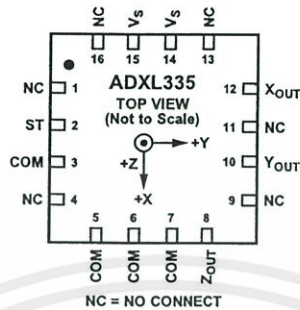


ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NC = NO CONNECT
 NOTES
 1. EXPOSED PAD IS NOT INTERNALLY CONNECTED BUT SHOULD BE SOLDERED FOR MECHANICAL INTEGRITY.

Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect ¹ .
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect ¹ .
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z _{OUT}	Z Channel Output.
9	NC	No Connect ¹ .
10	Y _{OUT}	Y Channel Output.
11	NC	No Connect ¹ .
12	X _{OUT}	X Channel Output.
13	NC	No Connect ¹ .
14	V _S	Supply Voltage (1.8 V to 3.6 V).
15	V _S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect ¹ .
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

¹NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PERFORMANCE CHARACTERISTICS

N > 1000 for all typical performance plots, unless otherwise noted.

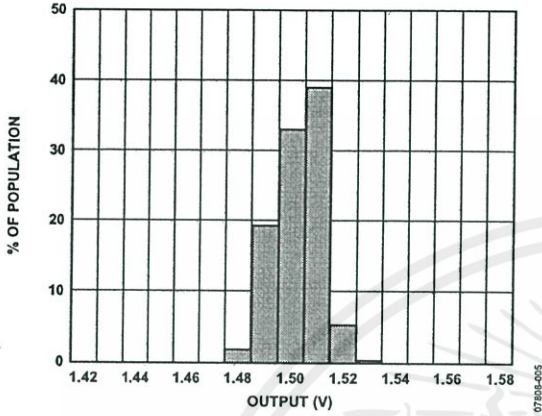


Figure 3. X-Axis Zero g Bias at 25°C, $V_s = 3 V$

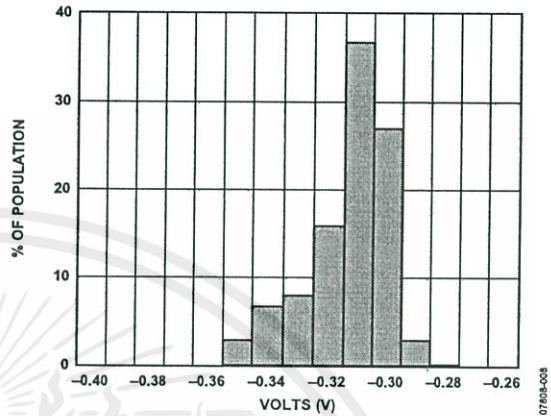


Figure 6. X-Axis Self-Test Response at 25°C, $V_s = 3 V$

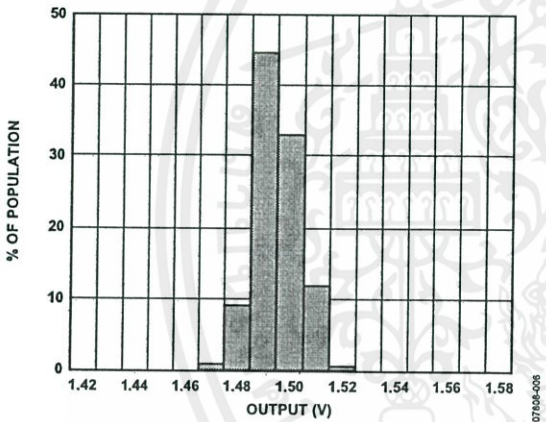


Figure 4. Y-Axis Zero g Bias at 25°C, $V_s = 3 V$

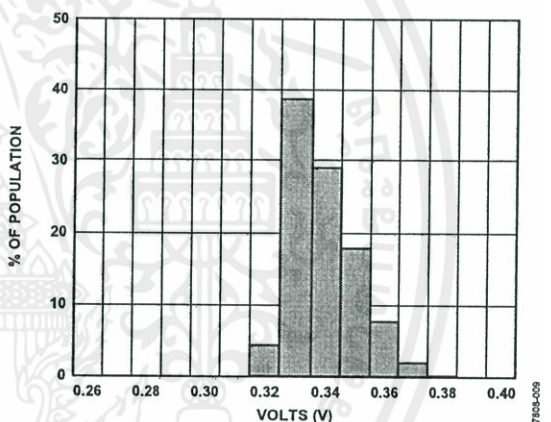


Figure 7. Y-Axis Self-Test Response at 25°C, $V_s = 3 V$

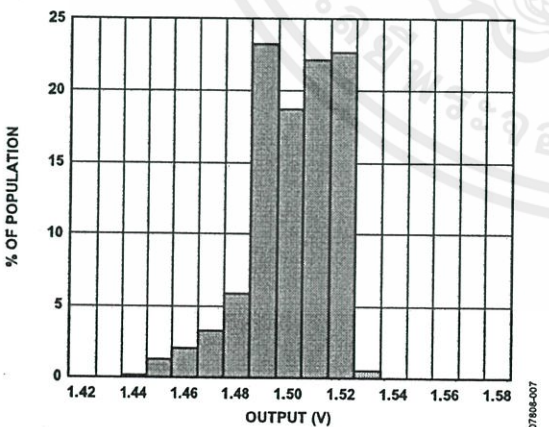


Figure 5. Z-Axis Zero g Bias at 25°C, $V_s = 3 V$

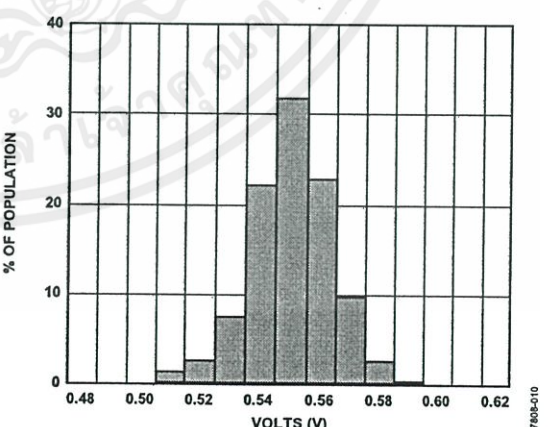


Figure 8. Z-Axis Self-Test Response at 25°C, $V_s = 3 V$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

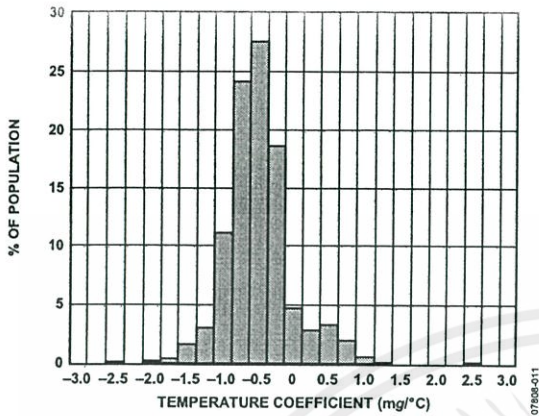


Figure 9. X-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

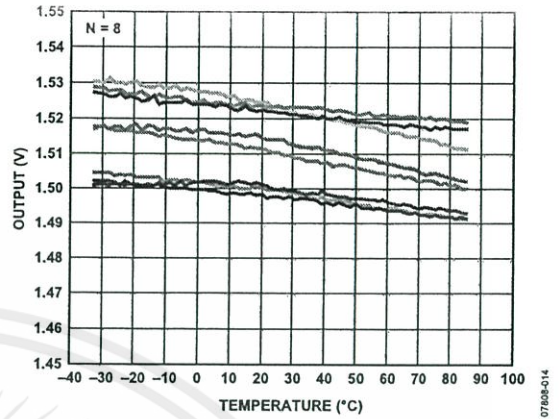


Figure 12. X-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

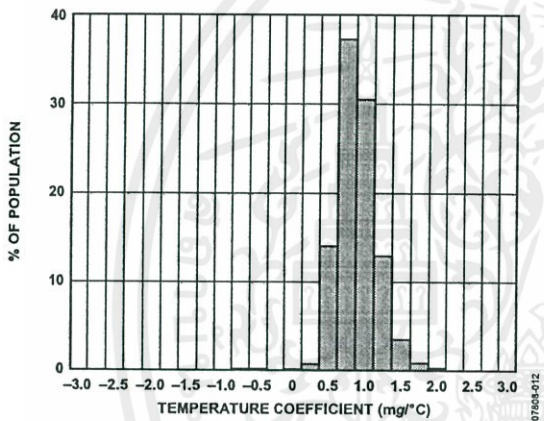


Figure 10. Y-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

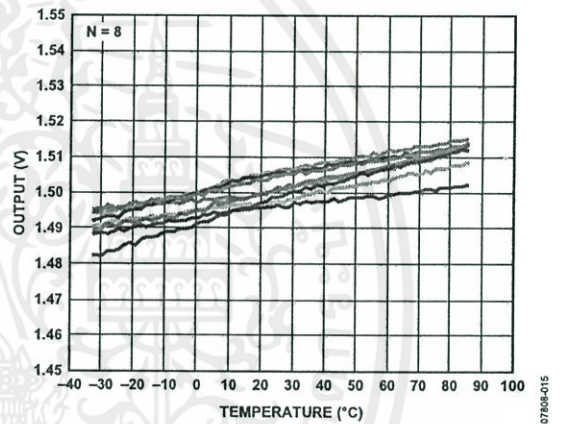


Figure 13. Y-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

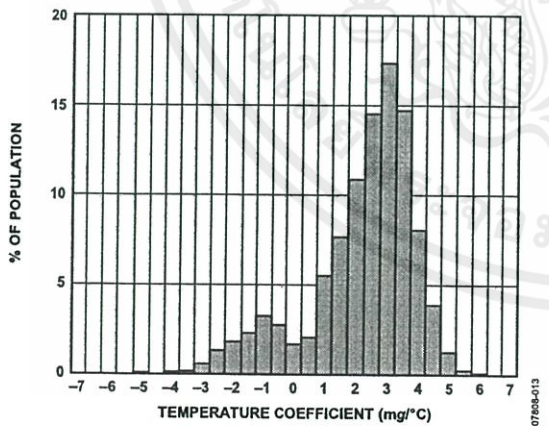


Figure 11. Z-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

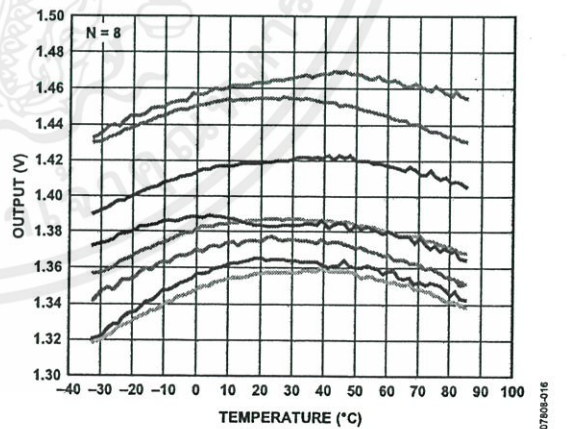


Figure 14. Z-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADXL335

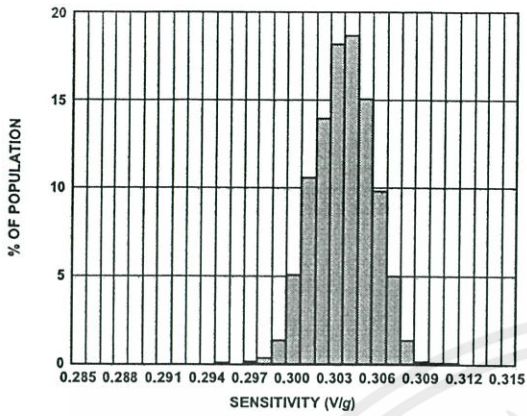


Figure 15. X-Axis Sensitivity at 25°C, $V_S = 3\text{ V}$

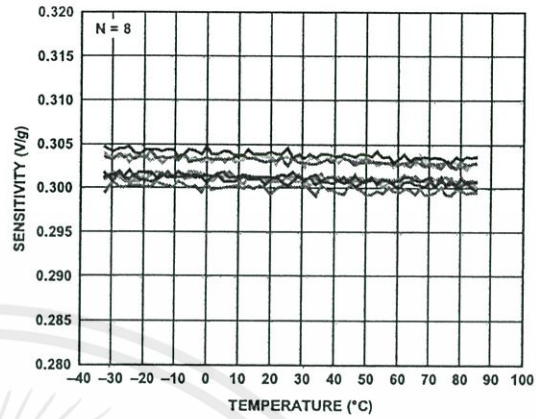


Figure 18. X-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_S = 3\text{ V}$

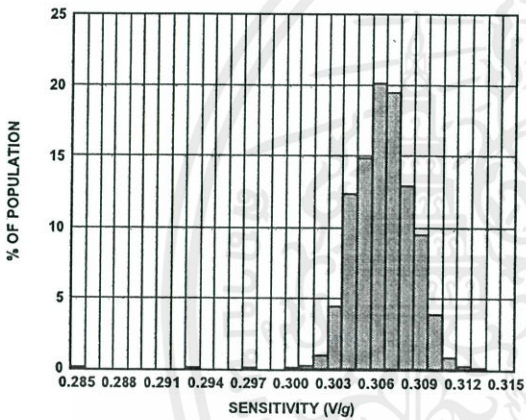


Figure 16. Y-Axis Sensitivity at 25°C, $V_S = 3\text{ V}$

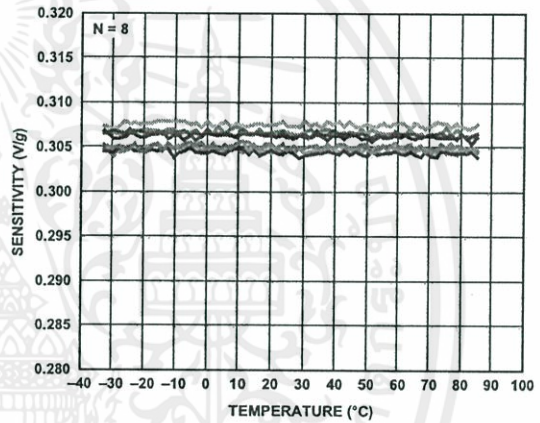


Figure 19. Y-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_S = 3\text{ V}$

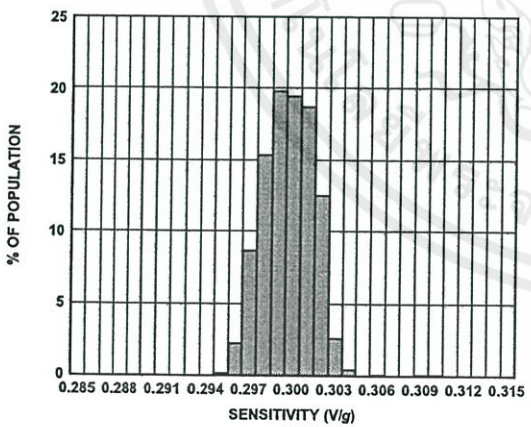


Figure 17. Z-Axis Sensitivity at 25°C, $V_S = 3\text{ V}$

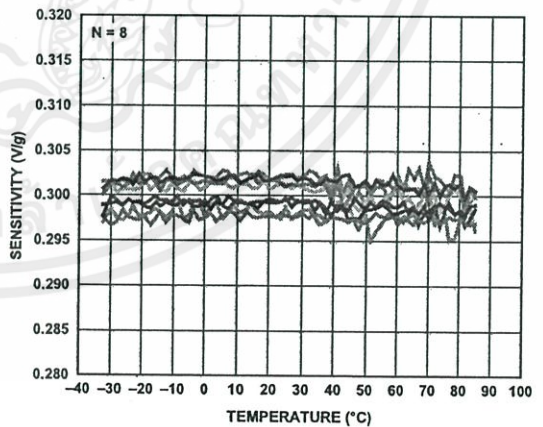


Figure 20. Z-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_S = 3\text{ V}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

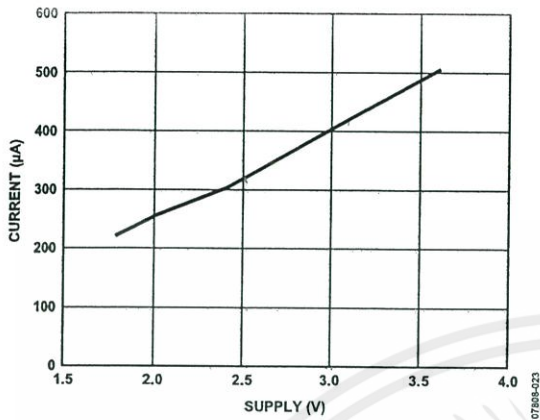


Figure 21. Typical Current Consumption vs. Supply Voltage

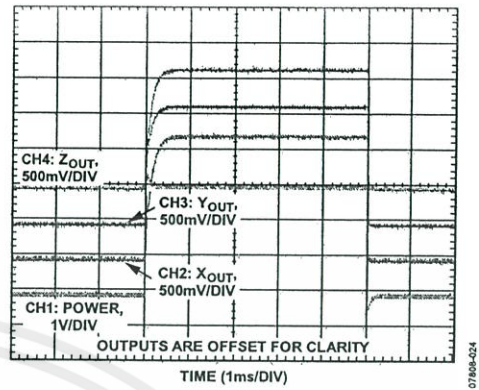
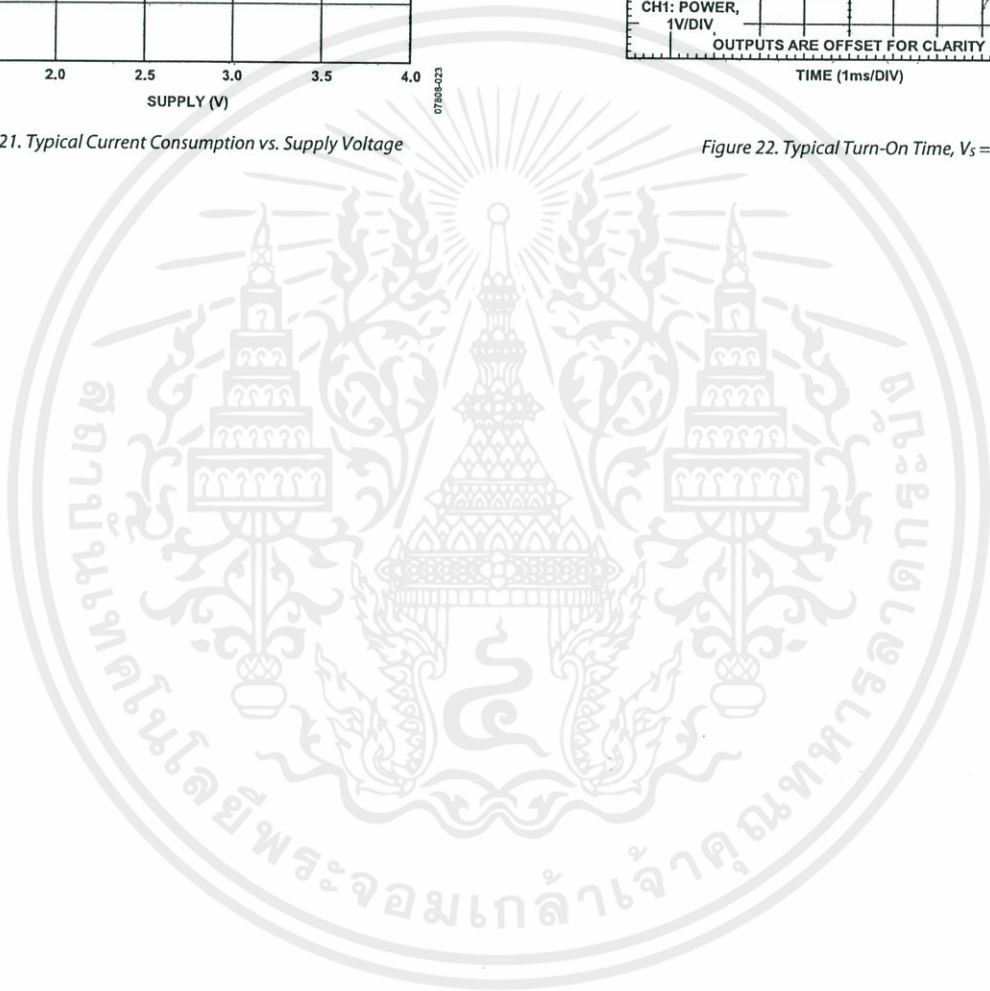


Figure 22. Typical Turn-On Time, $V_s = 3\text{ V}$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADXL335

THEORY OF OPERATION

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ± 3 g minimum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a 32 k Ω resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

MECHANICAL SENSOR

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

PERFORMANCE

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or nonmonotonic behavior, and temperature hysteresis is very low (typically less than 3 mg over the -25°C to $+70^{\circ}\text{C}$ temperature range).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATIONS INFORMATION

POWER SUPPLY DECOUPLING

For most applications, a single 0.1 μF capacitor, C_{DC} , placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply. However, in applications where noise is present at the 50 kHz internal clock frequency (or any harmonic thereof), additional care in power supply bypassing is required because this noise can cause errors in acceleration measurement.

If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite bead can be inserted in the supply line. Additionally, a larger bulk bypass capacitor (1 μF or greater) can be added in parallel to C_{DC} . Ensure that the connection from the ADXL335 ground to the power supply ground is low impedance because noise transmitted through ground has a similar effect to noise transmitted through V_{S} .

SETTING THE BANDWIDTH USING C_{X} , C_{Y} , AND C_{Z}

The ADXL335 has provisions for band limiting the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is

$$F_{-3\text{ dB}} = 1/(2\pi(32\text{ k}\Omega) \times C_{(X, Y, Z)})$$

or more simply

$$F_{-3\text{ dB}} = 5\ \mu\text{F}/C_{(X, Y, Z)}$$

The tolerance of the internal resistor (R_{FILTR}) typically varies as much as $\pm 15\%$ of its nominal value (32 k Ω), and the bandwidth varies accordingly. A minimum capacitance of 0.0047 μF for C_{X} , C_{Y} , and C_{Z} is recommended in all cases.

Table 4. Filter Capacitor Selection, C_{X} , C_{Y} , and C_{Z}

Bandwidth (Hz)	Capacitor (μF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

SELF-TEST

The ST pin controls the self-test feature. When this pin is set to V_{S} , an electrostatic force is exerted on the accelerometer beam. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output is -1.08 g (corresponding to -325 mV) in the X-axis, $+1.08\text{ g}$ (or $+325\text{ mV}$) on the Y-axis, and $+1.83\text{ g}$ (or $+550\text{ mV}$) on the Z-axis. This ST pin can be left open-circuit or connected to common (COM) in normal use.

Never expose the ST pin to voltages greater than $V_{\text{S}} + 0.3\text{ V}$. If this cannot be guaranteed due to the system design (for instance, if there are multiple supply voltages), then a low V_{F} clamping diode between ST and V_{S} is recommended.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The selected accelerometer bandwidth ultimately determines the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor to improve the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at X_{OUT} , Y_{OUT} , and Z_{OUT} .

The output of the ADXL335 has a typical bandwidth of greater than 500 Hz. The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution.

The ADXL335 noise has the characteristics of white Gaussian noise, which contributes equally at all frequencies and is described in terms of $\mu\text{g}/\sqrt{\text{Hz}}$ (the noise is proportional to the square root of the accelerometer bandwidth). The user should limit bandwidth to the lowest frequency needed by the application to maximize the resolution and dynamic range of the accelerometer.

With the single-pole, roll-off characteristic, the typical noise of the ADXL335 is determined by

$$\text{rms Noise} = \text{Noise Density} \times (\sqrt{\text{BW} \times 1.6})$$

It is often useful to know the peak value of the noise. Peak-to-peak noise can only be estimated by statistical methods. Table 5 is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table 5. Estimation of Peak-to-Peak Noise

Peak-to-Peak Value	% of Time That Noise Exceeds Nominal Peak-to-Peak Value
$2 \times \text{rms}$	32
$4 \times \text{rms}$	4.6
$6 \times \text{rms}$	0.27
$8 \times \text{rms}$	0.006

USE WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL335 is tested and specified at $V_{\text{S}} = 3\text{ V}$; however, it can be powered with V_{S} as low as 1.8 V or as high as 3.6 V. Note that some performance parameters change as the supply voltage is varied.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At $V_s = 3.6\text{ V}$, the output sensitivity is typically 360 mV/g . At $V_s = 2\text{ V}$, the output sensitivity is typically 195 mV/g .

The zero g bias output is also ratiometric, thus the zero g output is nominally equal to $V_s/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/g) increases while the noise voltage remains constant. At $V_s = 3.6\text{ V}$, the X-axis and Y-axis noise density is typically $120\text{ }\mu\text{g}/\sqrt{\text{Hz}}$, whereas at $V_s = 2\text{ V}$, the X-axis and Y-axis noise density is typically $270\text{ }\mu\text{g}/\sqrt{\text{Hz}}$.

Self-test response in g is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at $V_s = 3.6\text{ V}$, the self-test response for the ADXL335 is approximately -560 mV for the X-axis, $+560\text{ mV}$ for the Y-axis, and $+950\text{ mV}$ for the Z-axis.

At $V_s = 2\text{ V}$, the self-test response is approximately -96 mV for the X-axis, $+96\text{ mV}$ for the Y-axis, and -163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at $V_s = 3.6\text{ V}$ is $375\text{ }\mu\text{A}$, and typical current consumption at $V_s = 2\text{ V}$ is $200\text{ }\mu\text{A}$.

AXES OF ACCELERATION SENSITIVITY

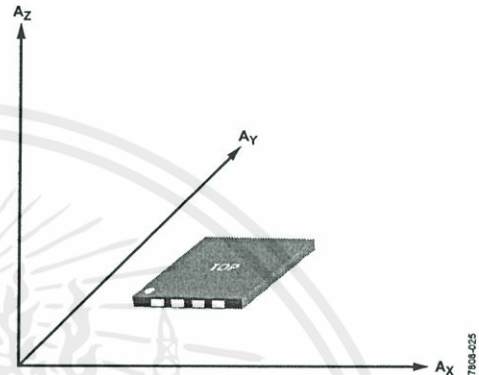


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

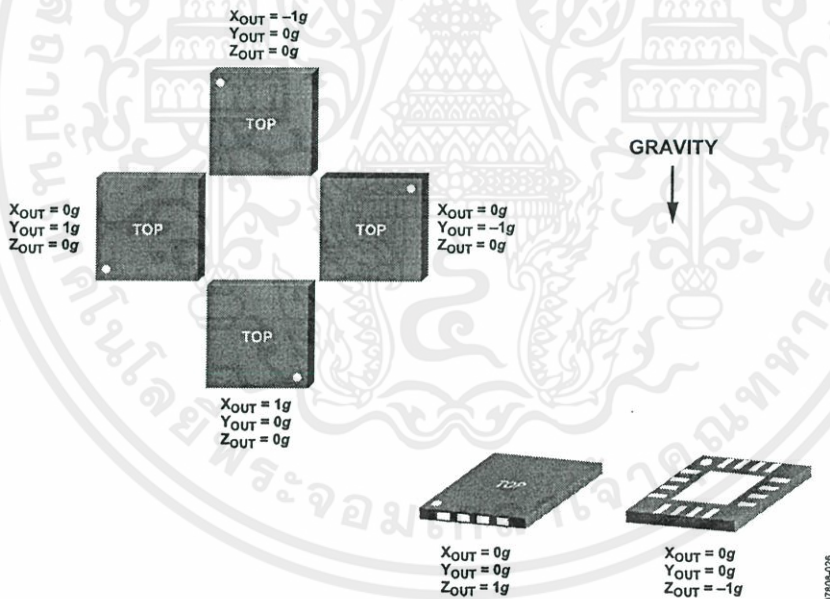


Figure 24. Output Response vs. Orientation to Gravity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LAYOUT AND DESIGN RECOMMENDATIONS

The recommended soldering profile is shown in Figure 25 followed by a description of the profile features in Table 6. The recommended PCB layout or solder land drawing is shown in Figure 26.

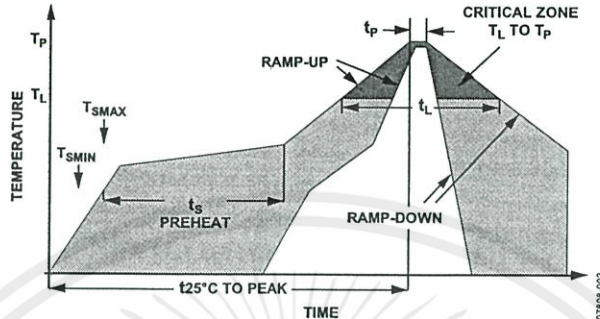


Figure 25. Recommended Soldering Profile

Table 6. Recommended Soldering Profile

Profile Feature	Sn63/Pb37	Pb-Free
Average Ramp Rate (T_L to T_P)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{SMIN})	100°C	150°C
Maximum Temperature (T_{SMAX})	150°C	200°C
Time (T_{SMIN} to T_{SMAX}) (t_s)	60 sec to 120 sec	60 sec to 180 sec
T_{SMAX} to T_L		
Ramp-Up Rate	3°C/sec max	3°C/sec max
Time Maintained Above Liquidous (T_L)		
Liquidous Temperature (T_L)	183°C	217°C
Time (t_L)	60 sec to 150 sec	60 sec to 150 sec
Peak Temperature (T_P)	240°C + 0°C/-5°C	260°C + 0°C/-5°C
Time Within 5°C of Actual Peak Temperature (t_p)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

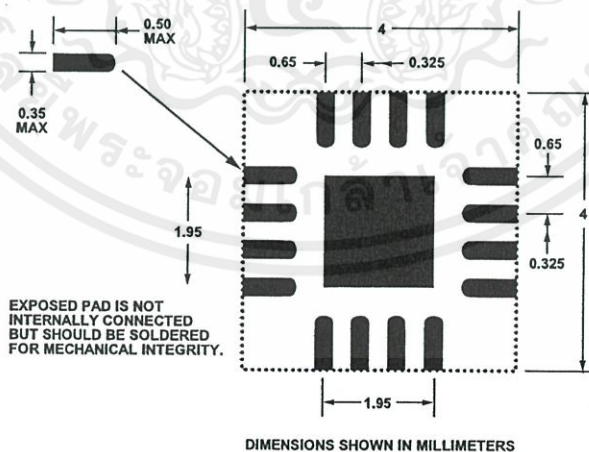


Figure 26. Recommended PCB Layout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADXL335

OUTLINE DIMENSIONS

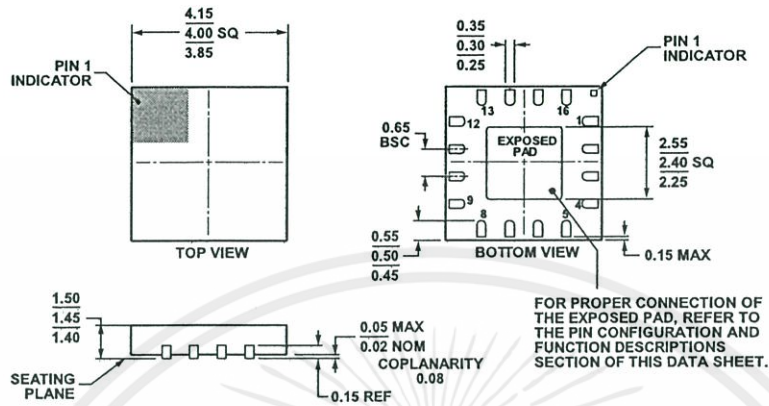


Figure 27. 16-Lead Lead Frame Chip Scale Package [LFCSP_LQ]
4 mm × 4 mm Body, 1.45 mm Thick Quad
(CP-16-14)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Measurement Range	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL335BCPZ ¹	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL ¹	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL7 ¹	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
EVAL-ADXL335Z ¹				Evaluation Board	

¹ Z = RoHS Compliant Part.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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?"
```

```
readwaitTime
```

```
#Invoke GPIO
```

```
echo "17" > /sys/class/gpio/export
```

```
echo "18" > /sys/class/gpio/export
```

```
echo "22" > /sys/class/gpio/export
```

```
echo "23" > /sys/class/gpio/export
```

```
#Start Loop
```

```
while :
```

```
do
```

```
#Read MySQL Data
```

```
#Direction
```

```
direction17=$(mysql -B --disable-column-names --user=$mysqlusername -  
-password=$mysqlpasswordgpio -e "SELECT pinDirection FROM pinDirection WHERE  
pinNumber='17'");
```

```
direction18=$(mysql -B --disable-column-names --user=$mysqlusername -  
-password=$mysqlpasswordgpio -e "SELECT pinDirection FROM pinDirection WHERE  
pinNumber='18'");
```

```
direction22=$(mysql -B --disable-column-names --user=$mysqlusername -  
-password=$mysqlpasswordgpio -e "SELECT pinDirection FROM pinDirection WHERE  
pinNumber='22'");
```

```
direction23=$(mysql -B --disable-column-names --user=$mysqlusername -  
-password=$mysqlpasswordgpio -e "SELECT pinDirection FROM pinDirection WHERE  
pinNumber='23'");
```

```
#Status
```

```
status4=$(mysql -B --disable-column-names --user=$mysqlusername  
-password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE  
pinNumber='4'");
```

```
status17=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='17'");
```

```
status18=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='18'");
```

```
status21=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='21'");
```

```
status22=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='22'");
```

```
status23=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='23'");
```

```
status24=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='24'");
```

```
status25=$(mysql -B --disable-column-names --user=$mysqlusername --password=$mysqlpasswordgpio -e "SELECT pinStatus FROM pinStatus WHERE pinNumber='25'");
```

```
#Right
```

```
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
```

```
#Right
```

```
if [ "$direction18" == "out" ]; then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
echo "out" > /sys/class/gpio/gpio18/direction
if [ "$status18" == "1" ]; then
echo "1" > /sys/class/gpio/gpio18/value
#echo "0" > /sys/class/gpio/gpio22/value
#echo "0" > /sys/class/gpio/gpio23/value
#echo "0" > /sys/class/gpio/gpio17/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
#Backward
if [ "$direction22" == "out" ]; then
#echo "out" > /sys/class/gpio/gpio17/direction
echo "out" > /sys/class/gpio/gpio22/direction
if [ "$status22" == "1" ]; then
#echo "1" > /sys/class/gpio/gpio17/value
echo "1" > /sys/class/gpio/gpio22/value
#echo "0" > /sys/class/gpio/gpio23/value
#echo "0" > /sys/class/gpio/gpio18/value
```

```
echo "GPIO 22 Turned On"
else
#echo "0" > /sys/class/gpio/gpio17/value
echo "0" > /sys/class/gpio/gpio22/value
echo "GPIO 22 Turned Off"
fi
else
echo "in" > /sys/class/gpio/gpio22/direction
fi
```

```
#Forward
if [ "$direction23" == "out" ]; then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกด้วย มิได้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
echo "out" > /sys/class/gpio/gpio23/direction
if [ "$status23" == "1" ]; then
#echo "1" > /sys/class/gpio/gpio18/value
echo "1" > /sys/class/gpio/gpio23/value
#echo "0" > /sys/class/gpio/gpio22/value
#echo "0" > /sys/class/gpio/gpio17/value
```

```
echo "GPIO 23 Turned On"
else
#echo "0" > /sys/class/gpio/gpio18/value
echo "0" > /sys/class/gpio/gpio23/value
echo "GPIO 23 Turned Off"
fi
else
echo "in" > /sys/class/gpio/gpio23/direction
fi
done
```

CODE PHP (SERVER)

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

```
$id=$_GET['id'];
$status=$_GET['status'];
$id=$_GET['id'];
$status=$_GET['status1'];
```

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

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

```
$query_control1 = sprintf("UPDATE `pinStatus` SET `pinStatus`=%s where
`pinID`=%s",$status1,$id1);
```

```
mysql_query($query_control1,$mysql);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
echo "Pin ".$id." has been set ".$status;
echo "Pin ".$id1." has been set ".$status1;
?>
```

CODE CAMERA SHELL SCRIPT (SERVER)

```
#!/bin/tcsh

setmyip=192.168.66.99
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
```

CODE ที่ใช้สำหรับควบคุมการเคลื่อนที่ของรถ แสดงภาพจากกล้องและค่าความเอียงจากเซนเซอร์บนแอปพลิเคชันแอนดรอยด์

Code PreferenceActivity (ส่วนที่ใช้ในการเชื่อมต่อกับกล้อง)

```
package com.camera.simplejpeg;

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";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกด้วย

```

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
    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);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 /*
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในข้อมูลและข้อมูลอ้างอิงที่มีผลการนำไปใช้
 * 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.getPreferences( 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));

    // 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 MjpegActivity (ส่วนที่ใช้ควบคุมการเคลื่อนที่ของรถ แสดงภาพจากกล้องและค่าความเอียงจากเซนเซอร์)

```
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 java.util.Set;
import java.util.UUID;

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.simplemjpeg.MjpegActivity;
import com.camera.simplemjpeg.R;

import android.annotation.TargetApi;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงบนสื่อใดๆ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import android.view.Window;
import android.view.WindowManager;
import android.widget.CompoundButton;
import android.widget.TextView;
import android.widget.Toast;
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;
```

```
    //===== BlueTooth=====
```

```
    volatile boolean stopWorker;
```

```
    int readBufferPosition;
```

```
    byte[] readBuffer;
```

```
    Thread workerThread;
```

```
    String temp;
```

```
    InputStream tmpIn = null;
```

```
    OutputStream tmpOut = 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);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยราชภัฏ

```

        // 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);

        //=====Blue Tooth=====

        BluetoothAdapter mBluetoothAdapter = BluetoothAdapter
            .getDefaultAdapter();
        if (mBluetoothAdapter == null) {
            Toast.makeText(MjpegActivity.this, "Bluetooth null",
                Toast.LENGTH_SHORT).show();
        }
        if (!mBluetoothAdapter.isEnabled()) {
            Toast.makeText(MjpegActivity.this, "Bluetooth enable",
                Toast.LENGTH_SHORT).show();
            Intent enableintent = new Intent(
                BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableintent, 1);
        }

        Set<BluetoothDevice> pairedDevices = mBluetoothAdapter
            .getBondedDevices();
        // If there are paired devices
        if (pairedDevices.size() > 0) {

            Toast.makeText(MjpegActivity.this, "pair device enable",
                Toast.LENGTH_SHORT).show();
            // Loop through paired devices
            for (BluetoothDevice device : pairedDevices) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ซ้ำไปมาเพื่อเป็นการกีดกันผู้อื่นในการนำข้อมูลไปใช้

```

// Add the name and address to an array adapter to
show in a

// ListView
// mAdapter.add(device.getName() + "\n" +
// device.getAddress());
String name = device.getName();
Log.v("LogTxt", name);

// Toast.makeText(Bluetooth_Main.this, name,
// Toast.LENGTH_SHORT).show();

if (name.equals("BT UART")) {
    Toast.makeText(MjpegActivity.this, "Connected
BT UART",
    Toast.LENGTH_SHORT).show();
    // Connect
    final BluetoothSocket mmSocket;
    final BluetoothDevice mmDevice;
    final UUID MY_UUID = UUID
    .fromString("00001101-0000-1000-8000-
00805f9b34fb");
    BluetoothSocket tmp = null;
    mmDevice = device;
    try {
        // MY_UUID is the app's UUID string, also
        used by the
        // server code
        tmp =
device.createRfcommSocketToServiceRecord(MY_UUID);

    } catch (IOException e) {
    }
    mmSocket = tmp;
    try {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น โปรดอย่าให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

socket. This will
exception

// Connect the device through the
// block until it succeeds or throws an

mmSocket.connect();
Log.v("LogTxt", "Connect!");

Toast.makeText(MjpegActivity.this,
               "Bluetooth has Connected",
Toast.LENGTH_SHORT) .show();

// #### Connect!!!!!!!!!!!!!!
// Start manage socket
try {
    tmpIn = mmSocket
            .getInputStream();
    tmpOut = mmSocket
            .getOutputStream();
    Log.v("LogTxt", "Can create
            stream");

    TextView btstatusbox = (TextView) findViewById(R.id.btstatus);
    btstatusbox.setText("Device connected");
    Toast.makeText(getApplicationContext(),
                    "Device connected", Toast.LENGTH_SHORT)
            .show();

    // Try to recieve data
    beginListenForData();

} catch (IOException connectException) {
    // Unable to connect; close the socket and get out
    TextView btstatusbox = (TextView) findViewById(R.id.btstatus);
    btstatusbox.setText("Device cannot connect");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติไปรษณีย์โทรเลขโทรสารโทรคมนาคม
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือเผยแพร่ข้อมูลใดๆโดยไม่ได้รับอนุญาต

```

        // Toast.makeText(getApplicationContext(),
        // "Device cannot connect",
        // Toast.LENGTH_LONG).show();
        Log.v("LogTxt", "Cannot connect");
        try {
            mmSocket.close();
        } catch (IOException closeException) {
        }
        return;
    }

} catch (IOException e) {
}
}
}
//=====\
}

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

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับคุณใช้เพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        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();
    }

    void beginListenForData() {

        final Handler handler = new Handler();

        stopWorker = false;
        readBufferPosition = 0;
        readBuffer = new byte[1024];

        workerThread = new Thread(new Runnable() {
            String temp="a";

            public void run() {
                // Toast.makeText(Bluetooth_Main.this, "Thread run",
                // Toast.LENGTH_SHORT).show();
                TextView textbox = (TextView) findViewById(R.id.temp);
                // textbox.setText("run th");
            }
        });
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือคัดลอกเอกสารนี้ไปเผยแพร่ในที่สาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Log.v("BT respond", "Thread run");
while (!Thread.currentThread().isInterrupted() && !stopWorker) {
```

```
    try {
```

```
        // Log.v("BT_respond", "Try recieve");
        int bytesAvailable = tmpIn.available();
```

```
        if (bytesAvailable > 1) {
```

```
            Log.v("BT_count", Integer.toString(bytesAvailable));
```

```
            byte[] packetBytes = new byte[bytesAvailable];
```

```
            tmpIn.read(packetBytes);
```

```
            //for (int i = 0; i < bytesAvailable; i++) {
```

```
                // byte b = packetBytes[i];
```

```
                // Log.v("BT_packet", Byte.toString(b));
```

```
                final String data2 = new String(packetBytes,"US-ASCII");
```

```
                Log.v("BT_read", data2);
```

```
                //Log.v("BT_Int", Integer.toString(packetBytes[0]));
```

```
                //String temp = new String(tempbyte, "US-ASCII");
```

```
                //Log.v("BT_temp", temp);
```

```
                //if (temp.equals("\r")) {
```

```
                    //Log.v("BT_read", "ok");
```

```
                //}
```

```
            // if (b == delimiter) {
```

```
                /*
```

```
                * byte[] encodedBytes = new
```

```
                * byte[readBufferPosition];
```

```
                * //System.arraycopy(readBuffer, 0,encodedBytes,
```

```
                * 0,encodedBytes.length);
```

```
                * System.arraycopy(packetBytes, 0,encodedBytes,
```

```
                * 0,encodedBytes.length);
```

```
                *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำหรือดัดแปลงในลักษณะใดๆที่อาจก่อให้เกิดความเสียหาย
หรือเป็นการละเมิดลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

```

* String data2 = new
* String(packetBytes,"US-ASCII"); Log.v("BT_read",
* data2);
*
*
* //final String data = new String(encodedBytes,
* "US-ASCII"); final String data = new
* String(encodedBytes, "US-ASCII");
* //Log.v("BT_respond", data); readBufferPosition =
* 0; // TextView textbox = (TextView) //
* findViewById(R.id.temp); //
* textbox.setText("ddd");
*
* // String[] splitdata = data.split(","); // //
* splitdata[0] // = // temp // .....
*
* // temp = splitdata[0]; temp = data;
*/
handler.post(new Runnable() {
public void run() {
TextView textbox = (TextView) findViewById(R.id.temp);
//TextView Ecg_text = (TextView) findViewById(R.id.ECG);
//TextView rate_text = (TextView) findViewById(R.id.Hrt_rate);
temp = temp+data2;
Log.v("BT_temp", temp);
if(temp.contains("\r")){
Log.v("BT_check", "enter");
textbox.setText(temp);
String[] separated = temp.split(",");
//Ecg_text.setText(separated[0]);
//rate_text.setText(separated[1]);
temp="";
}
// mydata.setText(data);
//TextView textbox = (TextView) findViewById(R.id.temp);
//textbox.setText(data2);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีการนำไปใช้

```

    }
    });
    //else {}
} catch (IOException ex) {
    stopWorker = true;
    Log.v("BT respond", "Stop worker");
    }
}
}
});

workerThread.start();
// workerThread.run();
Toast.makeText(MjpegActivity.this, "Thread Start", Toast.
LENGTH_SHORT).show();
}

public class DoRead extends AsyncTask<String, Void, MjpegInputStream> {
    protected MjpegInputStream doInBackground( String... params){
        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;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในเนื้อหาเอกสารฉบับนี้ที่มีการนำไปใช้

```

protected void onPostExecute(MjpegInputStream result) {
    mv.setSource(result);
}

```

```
if(result!=null) result.setSkip(1);
mv.setDisplayMode(MjpegView.SIZE_BEST_FIT);
mv.showFps(true);
```

```
ToggleButton btn1 = (ToggleButton)findViewById(R.id.toggleButton1);
ToggleButton btn2 = (ToggleButton)findViewById(R.id.toggleButton2);
ToggleButton btn3 = (ToggleButton)findViewById(R.id.toggleButton3);
ToggleButton btn4 = (ToggleButton)findViewById(R.id.toggleButton4);
```

```
final String URL = "http://192.168.66.99/";
```

```
// Permission StrictMode
if (android.os.Build.VERSION.SDK_INT > 9) {
    StrictMode.ThreadPolicy policy = new StrictMode.
ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
}

//forward
btn1.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {

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

        if(isChecked){
            String out1 = getHttpGet(URL+"control2.php?id=2&status=1&
id1=4&status1=1");
            Toast.makeText(MjpegActivity.this,out1, Toast.
LENGTH_SHORT).show();
        }else{
            String out0 = getHttpGet(URL+"control2.php?id=2&
status=0&id1=4&status1=0"); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น "ก่อนเผยแพร่ให้ผู้อื่น" ฝ่าฝืนจะถือว่าผิดนโยบายด้านความปลอดภัย

```

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

//left
    btn2.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
            // TODO Auto-generated method stub
            if(isChecked){
                String out1 = getHttpGet(URL+"control2.php?id=4&status=1");
                Toast.makeText(MjpegActivity.this,out1, Toast.
LENGTH_SHORT).show();
            }else{
                String out0 = getHttpGet(URL+"control2.php?id=4&status=0");
                Toast.makeText(MjpegActivity.this,out0, Toast.
LENGTH_SHORT).show();
            }
        }
    });

//right
    btn3.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {

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

```

```

        if(isChecked){
            String out1 = getHttpGet(URL+"control2.php?id=2&status=1");
            Toast.makeText(MjpegActivity.this,out1, Toast.
LENGTH_SHORT).show();
        }else{
            String out0 = getHttpGet(URL+"control2.php?id=2&status=0");
            Toast.makeText(MjpegActivity.this,out0, Toast.
LENGTH_SHORT).show();
        }
    }
});

```

```

//backward
btn4.setOnCheckedChangeListener(new CompoundButton.
OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        // TODO Auto-generated method stub
        if(isChecked){
            String out1 = getHttpGet(URL+"control2.php?id=1&status=1&
id1=3&status1=1");
            Toast.makeText(MjpegActivity.this,out1, Toast.
LENGTH_SHORT).show();
        }else{
            String out0 = getHttpGet(URL+"control2.php?id=1&status=0&
id1=3&status1=0");
            Toast.makeText(MjpegActivity.this,out0, Toast.
LENGTH_SHORT).show();
        }
    }
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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();
} catch (IOException e) {
    e.printStackTrace();
}

return str.toString();
}
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้