ระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน

WIRELESS CAMERA CONTROL BY ANDROID SMARTPHONE

โดย

นายณัฐกูล ภูมิภักดี

นายทิวากร มณีโชติ

นายศิริพงษ์ กลั่นภูมิสี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน

WIRELESS CAMERA CONTROL BY ANDROID SMARTPHONE

โดย

นาย ณัฐกูล ภูมิภักดี

นาย ทิวากร มณีโชติ

นาย ศิริพงษ์ กลั่นภูมิสี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขา วิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน

WIRELESS CAMERA CONTROL BY ANDROID SMARTPHONE

โดย

นาย ณัฐกูล	ภูมิภักดี	53010449
นาย ทิวากร	มณีโชติ	53010597
นาย ศิริพงษ์	รัตนนรเศรษฐ	53011573

อาจารย์ที่ปรึกษา

ผศ.สมภพ แก้วมีชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

...../...../.....

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

...../...../.....

ปริญญานิพนธ์ปีการศึกษา 2556
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน
WIRELESS CAMERA CONTROL BY ANDROID SMARTPHONE

ผู้จัดทำ

1. นาย ณัฐกูล	ภูมิภักดี	53010449
2. นาย ทิวากร	มณีโชติ	53010597
3. นาย ศิริพงษ์	กลั่นภูมิสี	53011573

..... **อาจารย์ที่ปรึกษา**

ผศ.สมภพ แก้วมีชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์เป็นรูปเล่นขึ้นมาได้ เพราะอาศัยความรู้จากบูรพาจารย์หลายๆท่าน และความอนุเคราะห์ของ ผ.ศ.สมภพ แก้วมีชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ได้ให้คำแนะนำ ความช่วยเหลือ และคำชี้แน่ะแก้ปัญหา และตลอดจนประสบการณ์ต่างๆ แก่กลุ่มของข้าพเจ้า

ผู้จัดทำขอขอบคุณ บิดา มารดา ที่ให้การสนับสนุน ในด้านต่างๆ รวมทั้งคอยเป็นกำลังใจให้แก่ผู้จัดทำอย่างเสมอมา

สุดท้ายนี้ กลุ่มของข้าพเจ้าหวังว่า โครงงานนี้จะเป็นประโยชน์ นำไปประยุกต์ หรือใช้งานต่อไปในภายภาคหน้าได้

นายณัฐกูล ภูมิภักดี
นายทิวากร มณีโชติ
นายศิริพงษ์ รัตนนรเศรษฐ

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน

WIRELESS CAMERA CONTROL BY ANDROID SMARTPHONE

โดย นายณัฐกูล ภูมิภักดี 53010449
นายทิวากร มณีโชติ 53010597
นายศิริพงษ์ รัตนนรเศรษฐ 53011573

อาจารย์ที่ปรึกษา ผศ.สมภพ แก้วมีชัย

บทคัดย่อ

ปริญญานิพนธ์นี้ได้ศึกษาและทำการออกแบบ แอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ สำหรับรับภาพจากกล้อง และการแจ้งเตือนจากเซ็นเซอร์ตรวจจับการเคลื่อนไหวที่ติดตั้งอยู่กับราสเบอร์รี่ไพ รวมทั้งยังสามารถควบคุมทิศทางของกล้องให้สามารถหมุนในทิศทางที่กำหนดได้(บน-ล่าง ,ซ้าย-ขาว)

ABSTRACT

THIS PROJECT IS ABOUT STUDYING AND DESIGNING ANDROID SMARTPHONE APPLICATION FOR RECEIVE IMAGE FROM CAMERA AND ALARM BY MOTION SENSOR WHICH IS INSTALLED ON RASPBERRY PI. THIS APPLICATION CAN CONTROL CAMERA TO TURN UP, TURN DOWN, TURN LEFT AND TURN RIGHT.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 ภาษาจาวา	2
	2.2 ระบบปฏิบัติการแอนดรอยด์ (ANDROID)	6
	2.3 ปฏิบัติการลีนุกซ์ (LINUX)	9
	2.4 ไพธอน (PYTHON)	12
	2.5 ราสเบียน (RASPBIAN OS)	17
	2.6 ราสเบอร์รี่ไพ (RASPBERRY PI)	18
	2.7 USER DATAGRAM PROTOCOL (UDP)	21
	2.8 REAL-TIME TRANSPORT PROTOCOL (RTP)	23
	2.9 REAL-TIME TRANSPORT CONTROL PROTOCOL (RTCP)	24
	2.10 REAL-TIME STREAMING PROTOCOL (RTSP)	25
	2.11 เว็บเซอเวอร์ (WEB SERVER)	26
	2.12 สเต็ปปิ้งมอเตอร์ (STEPPINGMOTOR)	27
	2.13 เซนเซอร์ (SENSER)	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.14 ไอซี ULN2003	32
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**34**
	3.1 การออกแบบ	34
	3.2 อุปกรณ์ที่ใช้ในการทดลอง	50
บทที่ 4	**ผลการทดลอง**	**55**
	4.1 ผลการทดลองการใช้ราสเบอร์รี่ไพในการควบคุมการทำงานของอุปกรณ์ต่างๆ	55
	4.2 การทดลองการทำงานของชุดคำสั่งของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	68
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**73**
	5.1 สรุปผล	73
	5.2 ข้อเสนอแนะ	74
บรรณานุกรม		**75**
ภาคผนวก ก	ดาต้าชีส	76
ภาคผนวก ข	โปรแกรมแอนดรอยด์	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	สัญลักษณ์ภาษาจาวา	2
2.2	สัญลักษณ์ระบบปฏิบัติการแอนดรอยด์	6
2.3	วงรอบการทำงานของแอปพลิเคชั่น	7
2.4	สัญลักษณ์ระบบปฏิบัติการลีนุ๊กซ์	10
2.5	สัญลักษณ์ไพธอน	12
2.6	สัญลักษณ์ RASPBIAN OS	17
2.7	สัญลักษณ์ราสเบอร์รี่ไพ	19
2.8	วงจรราสเบอร์รี่ไพ	20
2.9	การจัดเรียงขาของราสเบอร์รี่ไพ	21
2.10	สเต็ปปริ้มมอเตอร์แบบยูนิโพล่าร์	28
2.11	โมดูลตรวจความเคลื่อนไหว (PIR SENSOR)	32
2.12	ขาสัญญาณต่าง ๆ ของ IC ULN2003	25
3.1	บล็อกไดอะแกรมการทำงานของระบบ	34
3.2	บล็อกไดอะแกรมการทำงานของชุดคำสั่งระบบควบคุมกล้องไร้สายบนโทรศัพท์เคลื่อนที่ระบบปฏิบัตการแอนดรอยด์	35
3.3	การประมวลผลของชุดคำสั่งระบบควบคุมกล้องไร้สายผ่านโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	36
3.4	ไดอะแกรมแสดงการเชื่อมต่อเพื่อรับภาพของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์จากราสเบอร์รี่ไพ	37
3.5	ไดอะแกรมของส่วนของการส่ง UDP PACKET จากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ไปยังราสเบอร์รี่ไพเพื่อควบคุมมอเตอร์	38
3.6	ไดอะแกรมของส่วนของการรับการแจ้งเตือนจากราสเบอร์รี่ไพเมื่อมีเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้	40
3.7	โฟล์วชาร์ตของชุดคำสั่ง	43
3.8	โฟล์วชาร์ตของชุดคำสั่งเพื่อตรวจจับการ	48
3.9	ราสเบอร์รี่ไพ	50
3.10	โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์	51
3.11	USB-MINI-WIFI	51
3.12	ชิป ULN2003AN	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.13	เซ็นเซอร์ตรวจจับการเคลื่อนไหว	52
3.14	สเต็ปปิ้งมอเตอร์	53
3.15	กล้อง (MODULE RASPBERRY PI CAMERA)	53
3.16	ออสซิลโลสโคป	54
3.17	คอมพิวเตอร์	54
4.1	คำสั่งที่ใช้ในการสร้างไฟล์และสั่งรันโปรแกรม	55
4.2	สัญญาณที่วัดได้จากจีพีไอโอของราสเบอร์รี่ไพเมื่อสั่งมอเตอร์หมุนทวนเข็มนาฬิกา	56
4.3	สัญญาณที่วัดได้จากจีพีไอโอของราสเบอร์รี่ไพเมื่อสั่งมอเตอร์หมุนตามเข็มนาฬิกา	57
4.4	ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “A” ไปยังราสเบอร์รี่ไพ	58
4.5	ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “D” ไปยังราสเบอร์รี่ไพ	58
4.6	ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “W” ไปยังราสเบอร์รี่ไพ	59
4.7	ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “S” ไปยังราสเบอร์รี่ไพ	59
4.8	แพ็กเกตที่ตอบกลับเมื่อส่ง “A”	60
4.9	แพ็กเกตที่ตอบกลับเมื่อส่ง “D”	60
4.10	แพ็กเกตที่ตอบกลับเมื่อส่ง “W”	61
4.11	แพ็กเกตที่ตอบกลับเมื่อส่ง “S”	61
4.12	ผลของหน้าจอเทอร์มินอลของราสเบอร์รี่ไพเมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้	62
4.13	ผลของสัญญาณที่ได้จากเซ็นเซอร์เมื่อตรวจพบการเคลื่อนไหว	62
4.14	ผลหน้าเทอร์มินอลเมื่อพิมคำสั่ง “LSUSB” ในราสเบอร์รี่ไพ	63
4.15	ผลของหน้าเทอร์มินอลเมื่อพิมคำสั่ง “IFCONFIG”	63
4.16	ผลการสตรีมภาพโดยดูผ่านหน้าเวปบราวเซอร์	64
4.17	ผลที่ได้จากโปรแกรม WIRESHARK ใช้ดักจับข้อมูลทางเวปบราวเซอร์	65
4.18	ไอคอนของแอปพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	65
4.19	หน้าต่างสำหรับผู้ใช้ของแอปพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	66
4.20	หน้าต่างสำหรับตั้งค่าแอปพลิเคชั่นของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่

4.21 หน้าต่างแอปพลิเคชั่นเมื่อทำการกำหนดค่าต่างๆแล้วทำการรัน 67
4.22 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กขึ้นด้านบน 70
4.23 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กลงข้างล่าง 70
4.24 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กไปด้านซ้าย 70
4.25 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กได้ด้านขวา 71
4.26 ผลของแอพพลิเคชั่นเมื่อได้รับการแจ้งเตือนจากเซ็นเซอร์ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	ชนิดข้อมูลของภาษาจาวาแบบอ้างอิง	5
2.2	ลีนุกซ์คอมมานไลน์	11
2.3	แสดงเปรียบเทียบการประกาศตัวแปรระหว่างภาษาต่าง ๆ	14
2.4	ชนิดข้อมูลของภาษาไพธอน	16
2.5	คุณสมบัติของราสเบอร์รี่ไพ	19
2.6	ลำดับการป้อนกระแสแบบเฟสเดียว	29
2.7	ลำดับการป้อนกระแสแบบสองเฟส	29
2.8	ลำดับการป้อนกระแสแบบครึ่งเฟส	30
4.1	บอกลักษณะการหมุนของสเต็ปปิ้งมอเตอร์เมื่อทำการป้อนตัวอักษรในโปรแกรม	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันได้มีการพัฒนาเทคโนโลยีไร้สายเพื่ออำนวยความสะดวกในการดำรงชีวิตของมนุษย์ให้มีความสะดวกสบายมากขึ้น ซึ่งเทคโนโลยีไร้สายที่มนุษย์นิยมใช้พกติดตัวคือโทรศัพท์เคลื่อนที่สมาร์ทโฟน ได้มีการนำมาประยุกต์ใช้กับระบบรักษาความปลอดภัย โดยระบบรักษาความปลอดภัยระบบหนึ่งที่สำคัญภายในบ้านคือระบบกล้องรักษาความปลอดภัย แต่เนื่องจากระบบการแจ้งเตือนของระบบรักษาความปลอดภัยภายในบ้านยังไม่สามารถแจ้งเตือนได้ทันท่วงทีเช่น เมื่อเกิดเหตุการณ์โจรกรรม และนอกจากนี้การตรวจจับของกล้องได้ภาพที่ไม่ครอบคลุมบริเวณที่ทั่วถึง จึงทำให้สิ้นเปลืองค่าใช้จ่ายในการติดตั้งกล้องหลายตัว

1.2 วัตถุประสงค์

1. เพื่อทำการศึกษาและออกแบบระบบควบคุมสเต็ปปิ้งมอเตอร์ผ่านโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์
2. ทำการส่งโนติฟิเคชั่นไปยังโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เพื่อแจ้งเตือนการตรวจจับความเคลื่อนไหว
3. ทำการออกแบบและจำลองแอฟพลิเคชั่นรับภาพจากกล้องผ่านโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์
4. สร้างแอพพลิเคชั่นที่สามารถแจ้งเตือนการตรวจพบการเคลื่อนไหว ไปยังโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

1.3 ขอบเขตของปริญญานิพนธ์

1. ใช้สเต็ปปิ้งมอเตอร์ในการควบคุมการหมุนของฐานกล้อง
2. ใช้ราสเบอร์รี่ไพ ในการควบคุมสเต็ปปิ้งมอเตอร์ ,ตรวจจับการเคลื่อนไหวโดยผ่านเซนเซอร์ตรวจจับการเคลื่อนไหว ,รับภาพจากกล้อง และสื่อสารข้อมูลไปที่โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์
3. ใช้แอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ในการรับภาพจากกล้องและควบคุมสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ภาษาจาวา (Java)

2.1.1 ความหมายของจาวา

Java หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับ ภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม ซึ่งมีสถานะ และรูปพรรณ ประจำพฤติกรรม โดยมีสัญลักษณ์ตามรูปที่ 2.1

รูปที่ 2.1 สัญลักษณ์จาวา
ที่มา http://www.mindphp.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ข้อดีของภาษาจาวา

2.1.2.1 ภาษาจาวา เป็นภาษาโปรแกรมที่ง่ายในการเรียนรู้ ภาษา Java มีคุณลักษณะต่างๆ ดังนี้ เช่น เชื่อมต่อข้ามแพล็ตฟอร์ม (Platforms) ต่างๆ ได้ สามารถเขียนโปรแกรมแบบ OOP (Object-Oriented Programming) ได้ง่ายมาก โปรแกรมมีขนาดเล็ก และมีวิธีการเขียนไม่ยุ่งยากซับซ้อน ดังนั้นโปรแกรมที่เขียนด้วยภาษาจาวา จึงคอมไพล์ได้ง่ายตลอดจนตรวจสอบหาข้อผิดพลาดโปรแกรมได้ง่ายด้วย ภาษาจาวา เป็นภาษาที่ทำความเข้าใจได้ง่ายมาก มีขนาดเล็กและยากที่จะเกิดข้อผิดพลาด เขียนคำสั่งได้ง่าย มีประสิทธิภาพในการทำงานและมีความยืดหยุ่นสูง

2.1.2.2 ภาษาจาวา เป็นการเขียนโปรแกรมเชิงวัตถุ OOP (Object-Oriented Programming) การเขียนโปรแกรมเชิงวัตถุ เป็นเทคนิคการเขียนโปรแกรมให้มีลักษณะเป็นโมดูล (Module) แบ่งโปรแกรมเป็นส่วนๆ ตามสภาวะแวดล้อมการทำงานของโปรแกรมซึ่งเรียกว่า Method โดยทุก Method ก็คือ ระเบียบวิธี หรือการทำงานอย่างใดอย่างหนึ่ง โดยจะถูกรวบรวมอยู่ในคลาส ซึ่งหลักการเขียนโปรแกรมเชิงวัตถุจะมององค์ประกอบของโปรแกรมต่างๆเป็นคลาสหรือวัตถุ เรียกว่า ออบเจ็ค (Object) ตัวอย่าง เช่น วัตถุที่มองเห็นได้ เช่น รถ สินค้า หรือ วัตถุที่ไม่สามารถมองเห็นได้ เช่น เหตุการณ์ต่างๆ ข้อมูลต่างๆของ ออบเจ็ค จะถูกซ่อนไว้ในคลาส(Data Encapsulation) ซึ่งมีประโยชน์ในการแก้ไขข้อมูลหรือ Method ใดๆ ที่อยู่ในคลาส โดยไม่ส่งผลกระทบต่อการทำงานหรือเรียกใช้งานของ Object นั้น นอกจากนั้น จาวา ยังมีคุณสมบัติการสืบทอด (Inheritance) เพื่อส่งผ่านและถ่ายทอดลักษณะต่างๆ ของคลาสแม่ไปยังคลาสลูก ทำให้เขียนโปรแกรมได้ง่ายขึ้น และมีโครงสร้างการทำงานที่เข้าใจง่ายและมีความสัมพันธ์กัน

2.1.2.3 ภาษาจาวา เป็นอิสระต่อแพล็ตฟอร์ม (Java is Platform-Independent) จาวาเป็นอิสระต่อแพล็ตฟอร์ม ทั้งระดับซอร์ซโค้ด (Source Code) และไบนารีโค้ด (Binary Code) ช่วยให้สามารถเคลื่อนย้ายโปรแกรมจากระบบคอมพิวเตอร์หนึ่งไปยังระบบคอมพิวเตอร์อื่นได้อย่างง่ายดาย เพราะว่าโปรแกรมที่เขียนด้วยภาษา จาวา ได้รวบรวมคำสั่งต่างๆไว้ในไลบรารีคลาสพื้นฐานต่างๆ เป็น จาวาPackages ช่วยอำนวยความสะดวกในการเขียนคำสั่ง เมื่อย้ายโปรแกรมไปยังแพล็ตฟอร์มอื่น โดยไม่ต้องเขียนซอร์ซโค้ด (Source Code) ขึ้นใหม่ทำให้ประหยัดเวลามาก เมื่อคอมไพล์ซอร์ซโค้ด จะได้ไฟล์ไบนารีโค้ด ที่เรียกว่า Bytecode การรันโปรแกรมของ Java จะทำงานในลักษณะอินเทอร์พรีเตอร์ (Interpreter) ของไฟล์ Bytecode ซึ่งสามารถรันบนแพล็ตฟอร์มใดๆ ก็ได้ รวมทั้งระบบปฏิบัติการต่างๆ เช่น ระบบ Windows, Solaris, Linux หรือ MacOS โดยการแปลคำสั่งทีละคำสั่ง แพล็ตฟอร์มที่ Java ทำงานได้จะต้องประกอบด้วย 2 ส่วน คือ Java Virtual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Machine (JVM) และ Java Application Programming Interface (Java API) โดย Java Virtual Machine คือเครื่องมือที่รวบรวมคำสั่งคอมไฟล์และรันโปรแกรม Java ส่วน Java API เป็นกลุ่มของคลาส และอินเตอร์เฟส (Interface) ที่รวมอยู่ในไลบรารีที่เรียกว่า Java Package เช่น java.awt, java.util หรือ java.io เป็นต้น ลักษณะการทำงานของ Java ที่เป็นอิสระต่อแพล็ตฟอร์ม โดยการเขียนโปรแกรมเพียงครั้งเดียวแต่สามารถนำไปใช้ทำงานยังเครื่องอื่นๆ ได้

2.1.2.4 ภาษา Java มีระบบการทำงานและมีระบบความปลอดภัยที่ดี Java จะคำสั่งต่างๆ ที่เป็นส่วนประกอบของ Java API โดยมีการรวบรวมเป็นคลาสต่างๆไว้มากมาย ช่วยอำนวยความสะดวกในการเขียนโปรแกรม นอกจากนั้นยังมี Garbage Collectorโดยมีระบบจัดการหน่วยความจำเพื่อเก็บขยะของโปรแกรมและคืนหน่วยความจำให้กับระบบโปรแกรมที่เขียนด้วยภาษาจาวามีระบบจัดการข้อผิดพลาดที่เกิดจากการทำงานของโปรแกรมที่เรียกว่า Exception Handling ด้วยทำให้สามารถตรวจสอบโปรแกรม (Debug) โปรแกรมได้ง่ายขึ้น Java มีระบบความปลอดภัยที่ดี เช่น โปรแกรม Java ที่ทำงานบนเว็บบราวเซอร์ (Web Browser) ที่เรียกว่า Java Applet นั้นจะทำงานเฉพาะบนเครื่องแม่ข่าย (Server) โดยไม่สามารถเข้าถึงเครื่องลูกข่าย (Client) ไปทำลายไฟล์ หรือไฟล์ระบบ (System file) ได้ ทำให้มีระบบความปลอดภัยที่ดี ป้องกันข้อมูลจากไวรัส และโปรแกรมที่เขียนด้วย Java ไม่มีพฤติกรรมเป็นไวรัส ได้

2.1.3 ตัวแปรและชนิดของข้อมูลของภาษาจาวา

2.1.3.1 ตัวแปร หรือ Variable เป็นที่เก็บข้อมูลต่างๆ ที่เราต้องการใช้ในการประมวลผล และตัวแปรยังเป็นสัญลักษณ์ที่ใช้ในการแทนหน่วยความจำ ที่อยู่ในเครื่อง และ คอมไพเลอร์จะเป็นผู้กำหนดว่าอยู่ที่ใด มีขนาดเท่าใด

2.1.3.2 กฎเกณฑ์ในการตั้งชื่อตัวแปร

1) ประกอบด้วยตัวเลข ตัวอักษร(letter), เครื่องหมาย _ (Underscore) หรือเครื่องหมาย $ (Dollar sign)

2) ต้องขึ้นต้นด้วยตัวอักษรและห้ามเว้นวรรค

3) ไม่สามารถใช้สัญลักษณ์พิเศษอื่น เช่น +, -, *, &, %, # ฯลฯ เป็นส่วนประกอบอยู่ภายในชื่อได้

4) สามารถตั้งชื่อให้มีความยาวเท่าใดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ไม่สามารถใช้ คำสงวน(Reserved word)

6) ตัวอักษรตัวใหญ่และตัวเล็ก ถือว่ามีความแตกต่างกัน

2.1.3.3 ชนิดของข้อมูลของภาษาจาวา (Data Type) ชนิดข้อมูลในภาษาจาวาแบ่งเป็นสองประเภทใหญ่ๆ คือ

1) ชนิดข้อมูลแบบพื้นฐาน (primitive data type)

หมายถึงชนิดข้อมูลที่สามารถเก็บข้อมูลที่เป็นข้อมูลทั่วไปหรือข้อมูลพื้นฐาน มีทั้งหมด 8 ตัว ได้แก่ boolean, char, byte, short, int, long, float,double

2) ชนิดข้อมูลแบบอ้างอิง (reference data type)

มีความแตกต่างกับชนิดข้อมูลพื้นฐาน ที่ว่าชนิดข้อมูลชนิดนี้อยู่ในรูปแบบหนึ่ง ซึ่งการเข้าถึง (ใช้งาน)ข้อมูลเป็นการอ้างถึง มากกว่าการเข้าถึงข้อมูลโดยตรง เก็บข้อมูลไว้ 2 ส่วน คือ

- Execution Stack เก็บค่าอ้างอิงที่ชี้ไปยัง Heap memory
- Heap Memory เก็บข้อมูลที่เรียกว่าออปเจ็ค ที่สร้างขึ้นมาจากคลาส

ตารางที่2.1 ชนิดข้อมูลของภาษาจาวาแบบอ้างอิง

ชนิดข้อมูล	ขนาด (bit)	ค่าที่เก็บได้
boolean	JVM กำหนด	true หรือ false
Byte	8	-128 ถึง 127
Short	16	-32768 ถึง 32767
Int	32	-2147483648 ถึง 2147483648
Long	64	-9223372036854775808 ถึง 9223372036854775808
Float	32	-3.4E38 ถึง 3.4E38
double	64	-1.7E308 ถึง 1.7E308
Char	16	ใช้เก็บอักขระที่มีรหัสตั้งแต่ 0 ถึง 65535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ระบบปฏิบัติการแอนดรอยด์ (Android)

แอนดรอยด์ (Android) กูเกิลแอนดรอยด์ (Google Android) หรือ ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) เป็นชื่อเรียกชุดซอฟท์แวร์ หรือแพลตฟอร์ม (Platform) สำหรับอุปกรณ์อีเล็กทรอนิกส์ ที่มีหน่วยประมวลผลเป็นส่วนประกอบ อาทิเช่น คอมพิวเตอร์, โทรศัพท์ (Telephone), โทรศัพท์เคลื่อนที่ (Cell phone), อุปกรณ์เล่นอินเตอร์เน็ตขนาดพกพา (MID) เป็นต้น แอนดรอยด์นั้น ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัท กูเกิล กูเกิลแอนดรอยด์ เป็นชื่อเรียกอย่างเป็นทางการของแอนดรอยด์ เนื่องจากปัจจุบันนี้ บริษัทกูเกิล เป็นผู้ที่ถือสิทธิบัตรในตราสัญญาลักษณ์ ชื่อ และ รหัสต้นฉบับ (Source Code) ของแอนดรอยด์ ภายใต้เงื่อนไขการพัฒนาแบบ GNL โดยเปิดให้นักพัฒนา (Developer) สามารถนำรหัสต้นฉบับ ไปพัฒนาปรับแต่งได้อย่างเปิดเผย (Open source) ทำให้แอนดรอยด์มีผู้เข้าร่วมพัฒนาเป็นจำนวนมาก และพัฒนาไปได้อย่างรวดเร็ว แอนดรอยด์เปิดตัวอย่างเป็นทางการเมื่อวันที่ 5 พฤษภาคม พุทธศักราช 2550 ปัจจุบันมีผู้ร่วมพัฒนากว่า 52 องค์กรประกอบด้วยบริษัทซอฟท์แวร์ บริษัทผู้ผลิตอุปกรณ์ บริษัทผู้ผลิตชิ้นส่วนอิเลคทรอนิกส์ บริษัทผู้ให้บริการเครือข่าย และบริษัทที่เกี่ยวข้องกับการสื่อสาร ฯลฯ โดยมีสัญลักษณ์ตามรูปที่ 2.2

ANDROID

รูปที่ 2.2 สัญลักษณ์ระบบปฏิบัติการแอนดรอยด์
ที่มา http://parewa253607.wordpress.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ประเภทของแอนดรอยด์

ประเภทของชุดซอฟท์แวร์ เนื่องจากแอนดรอยด์นั้นเปิดให้นักพัฒนาเข้าไปชมรหัสต้นฉบับได้ ทำให้มีผู้พัฒนาจากหลายฝ่ายนำเอารหัสต้นฉบับมาปรับแต่ง และสร้างแอนดรอยด์ในแบบฉบับของตนเองขึ้น เราจึงแบ่งประเภทของแอนดรอยด์ออกได้เป็น 3 ประเภท ดังต่อไปนี้

2.2.1.1 Android Open Source Project (AOSP) เป็นแอนดรอยด์ประเภทแรกที่กูเกิลเปิดให้สามารถนำ "ต้นฉบับแบบเปิด" ไปติดตั้งและใช้งานในอุปกรณ์ต่าง ๆ ได้โดยไม่ต้องเสียค่าใช้จ่ายใดๆ

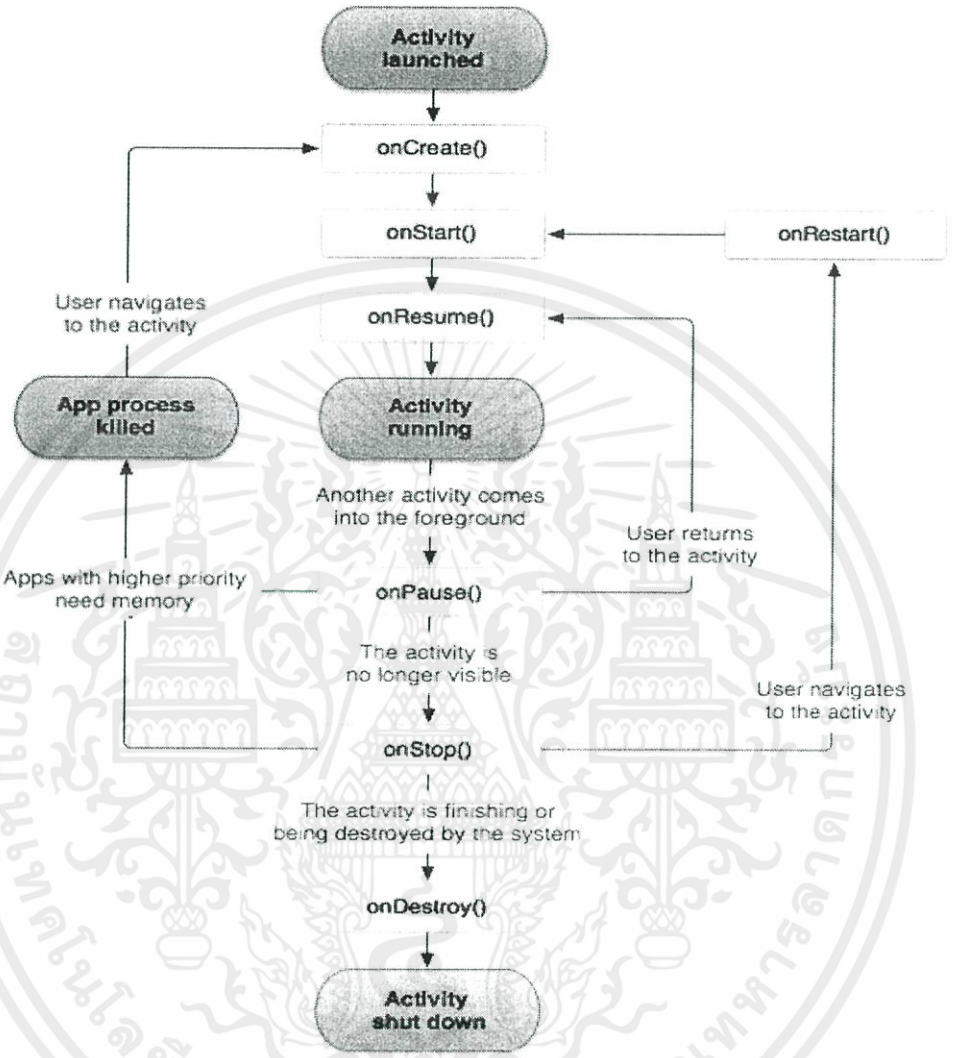
2.2.1.2 Open Handset Mobile (OHM) เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่มบริษัทผู้ผลิตอุปกรณ์พกพา ที่เข้าร่วมกับกูเกิลในนาม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาแอนดรอยด์ในแบบฉบับของตนออกมา โดยรูปร่างหน้าตาการแสดงผล และฟังค์ชั่นการใช้งาน จะมีความเป็นเอกลักษณ์ และมีลิขสิทธิ์เป็นของตน พร้อมได้รับสิทธิ์ในการมีบริการเสริมต่าง ๆ จากกูเกิล ที่เรียกว่า Google Mobile Service (GMS) ซึ่งเป็นบริการเสริมที่ทำให้แอนดรอยด์มีประสิทธิภาพ เป็นไปตามจุดประสงค์ของแอนดรอยด์ แต่การจะได้มาซึ่ง GMS นั้น ผู้ผลิตจะต้องทำการทดสอบระบบ และขออนุญาตกับทางกูเกิลก่อน จึงจะนำเครื่องออกสู่ตลาดได้

2.2.1.3 Cooking หรือ Customize เป็นแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่าง ๆ มาปรับแต่ง ในแบบฉบับของตนเอง โดยจะต้องทำการปลดล็อคสิทธิ์การใช้งานอุปกรณ์หรือ Unlock เครื่องก่อน จึงจะสามารถติดตั้งได้ โดยแอนดรอยด์ประเภทนี้ถือเป็นประเภทที่มีความสามารถมากที่สุด เท่าที่อุปกรณ์เครื่องนั้น ๆ จะรองรับได้ เนื่องจากได้รับการปรับแต่งให้เข้ากับอุปกรณ์นั้น ๆ จากผู้ใช้งานจริง

2.2.2 วงรอบการทำงานของแอปพลิเคชั่น (Android Activity Lifecycle)

ระบบปฏิบัติการแอนดรอยด์ถูกออกแบบมาเพื่อใช้งานกับอุปกรณ์บนโทรศัพท์เคลื่อนที่ ซึ่ง จะมีทรัพยากรของระบบที่จำกัด จึงจำเป็นที่ต้องมีกลไกสำหรับจัดการบริหารทรัพยากรเหล่านั้นซึ่งเรียกว่า วงรอบการทำงานของแอปพลิเคชั่น ซึ่งจะกำหนดสถานะที่แอคทิวิตีต้องผ่านตั้งแต่ตอนที่ถูกสร้างขึ้นจนกว่าจะเสร็จสิ้นการทำงาน การทำงานของวงรอบการทำงานจะมีหลักการทำงาน หลักๆ ดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 วงรอบการทำงานของแอปพลิเคชั่น

ที่มา http://www.android-app-market.com/android-activity-lifecycle.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 วงรอบการทำงานของแอปพลิเคชั่นมีการทำงานซึ่งสามารถอธิบายขั้นตอนของการทำงานได้ดังนี้

1) onCreate จะถูกเรียกเมื่อแอ็คทิวิตีถูกเริ่มการทำงาน ซึ่งปกติจะมีการสร้างส่วนติดต่อผู้ใช้หรือวิวต่างๆ สำหรับเปิดไฟล์ข้อมูลถาวร (Persistent Datafiles) ที่ใช้ในแอ็คทิวิตีและกำหนดค่าเริ่มต้นของแอ็คทิวิตี (Activity)หลังจากนั้นจะตามด้วยสถานะของเมธอด onStart

2) onStart จะถูกเรียกก่อนที่แอ็คทิวิตีจะแสดงผลออกมาที่หน้าจอ เมื่อเมธอด onStart ทำงานเสร็จ ถ้าแอ็คทิวิตีสามารถเป็นโฟร์กราวน์แอ็คทิวิตีบนหน้าจอ จะถูกส่งต่อไปยังสถานะของเมธอด onResume

3) onResume จะถูกเรียกถ้าแอ็คทิวิตีนั้นมีการถูกเรียกใช้งานจากผู้ใช้งาน หลังจากที่แอ็คทิวิตีนั้นอยู่ในสถานะของเมธอด onPause

4) onPause จะถูกเรียกเมื่อแอ็คทิวิตี จะถูกเปลี่ยนจากการทำงานเบื้องหน้าไปอยู่การทำงานเบื้องหลัง

5) onStop จะถูกเรียกเมื่อแอ็คทิวิตีนั้นๆ ผู้ใช้งานไม่ต้องการที่จะใช้งานในช่วงระยะเวลาหนึ่งๆ และจะตามด้วยสถานะของเมธอด onRestart ซึ่งจะถูกเรียกเมื่อแอ็คทิวิตีนั้นๆ ถูกเรียกกลับมาใช้งานอีกครั้งและตอบสนองกับผู้ใช้หรืออาจจะตามด้วยสถานะของเมธอด onDestroy จะถูกเรียกเมื่อแอ็คทิวิตีสิ้นสุดการทำงาน

6) onDestroy จะถูกเรียกใช้เมื่อจะปิดการทำงานของแอ็คทิวิตีนั้นๆ

2.3 ปฏิบัติการลีนุกซ์ (Linux)

ลีนุกซ์ (Linux) เป็นชื่อตัวปฏิบัติการระบบ (Operating System) ตัวหนึ่ง เช่นเดียวกับ DOS, Windows 95, Windows NT, OS/2 หรือ ระบบ Unix อื่นๆ Linux ถูกออกแบบมาให้ใช้กับเครื่องคอมพิวเตอร์ ที่ใช้ตัวประมวลผลหรือ CPU ตระกูล x86 (เช่น 80386, 486, Pentium เป็นต้น) แต่ปัจจุบันมีการพัฒนาให้ใช้กับตัวประมวลผลตระกูลอื่นๆ เช่น Alpha Chip ได้ด้วย โดยจะมีลักษณะการทำงานแบบ Unix โดย Unix เป็นระบบปฏิบัติการที่พัฒนาใช้กันมานานหลายปี เป็นระบบปฏิบัติการที่มีประสิทธิภาพมาก โดยมีสัญลักษณ์ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 สัญลักษณ์ระบบปฏิบัติการลีนุกซ์
ที่มา http://xn--ebook-tbr3dxn.blogspot.com/2011/09/unix-linux.html

2.3.1 ข้อดีของระบบปฏิบัติการลีนุกซ์

2.3.1.1 ลีนุกซ์เป็นระบบปฏิบัติการที่ สามารถดาวน์โหลดมาจากศูนย์บริการ FTP ต่างๆ บนอินเตอร์เน็ตหรือคุณอาจจะหาซื้อซีดีรอมลีนุกซ์ได้ในราคาที่ไม่แพงนักตามร้านหนังสือหรือร้านอุปกรณ์คอมพิวเตอร์ทั่วไป

2.3.1.2 ลีนุกซ์เป็นระบบปฏิบัติการ UNIX ที่สามารถรันได้บนเครื่องพีซีทั่วไปที่มีราคาไม่แพงนักโดยสามารถรันได้บนเครื่องที่มีหน่วยประมวลผลกลางตั้งแต่ 80386 ขึ้นไป พร้อมกับสนับสนุนอุปกรณ์ต่อพ่วงต่างๆ เช่น การ์ดควบคุมการแสดงผล ซีดีรอม ฮาร์ดดิสค์ เครื่องพิมพ์ และอีเธอร์เน็ตการ์ด เป็นต้น

2.3.1.3 คุณภาพลีนุกซ์ เป็นความร่วมมือระหว่างโปรแกรมเมอร์นับพันที่ติดต่อกันทางอินเตอร์เน็ต ใน รายงานความผิดพลาดและให้คำแนะนำหรือเสนอความสามารถใหม่ๆ ให้กับผู้พัฒนาตลอดเวลา

2.3.1.4 ความสามารถในการใช้งานไฟล์ร่วมกับระบบปฏิบัติการอื่นๆ ลีนุกซ์สนับสนุนระบบไฟล์ของระบบปฏิบัติการหลายชนิด ไม่ว่าจะเป็น DOS , Windows for Workgroup , Windows 95 , Windows NT เป็นต้น

2.3.1.5 ความต้องการทรัพยากรของระบบ ระบบขั้นต่ำที่ลีนุกซ์สามารถทำงานได้คือเครื่องพีซีที่มีหน่วยประมวลผลกลาง 80386 /SX หน่วยความจำ 2 เมกะไบต์ ฟลอบปี้ดิสค์ขนาด 1.44 หรือ 1.2 เมกะไบต์ การ์ดแสดงผล และจอภาพแบบโมโนโครม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ลีนุกซ์คอมมานไลน์ (Linux command line)

ในการใช้งานระบบปฏิบัติการลินุกซ์ ซึ่งการทำงานหลักจะสั่งการไปที่หน้าคอมมานไลน์ จึงจำเป็นอย่างยิ่งที่จะต้องรู้คำสั่งพื้นฐานของระบบปฏิบัติการลินุกซ์ เพื่อสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ โดยคำสั่งคอมมานไลน์พื้นฐานนั้นแสดงในตารางที่ 2.2 ดังนี้

ตารางที่ 2.2 ลีนุกซ์คอมมานไลน์

คำสั่ง	หน้าที่	รายละเอียด
Uname	Print system information	แสดงรายละเอียดของระบบปฏิบัติการ
Hostname	Print the system's hostname	แสดงกำหนดชื่อโฮสต์/
Ifconfig	Set/display network information	แสดงกำหนดค่าเกี่ยวกับเครือข่าย/
Host	Look up DNS	สืบค้นชื่อและไอพีของโฮสต์ในระบบ DNS
Whois	Lookup domain registrants	สืบค้นข้อมูลการจดทะเบียนโดเมน
คำสั่ง	**หน้าที่**	**รายละเอียด**
Ping	Check if host is reachable	ทดสอบการตอบสนองของโฮสต์ปลายทาง
Traceroute	View network path to a host	ตรวจสอบเส้นทางไปสู่โฮสต์ปลายทาง
Ls	List files	แสดงรายชื่อไฟล์และไดเร็คทอรี่
Cp	Copy files	สำเนาไฟล์
Mv	Rename files	เปลี่ยนชื่อไฟล์
Rm	Delete files	ลบไฟล์
Ln	Link files	สร้างไฟล์เชื่อมโยง
Cd	Change directory	ย้ายไปยังไดเร็คทอรี่ที่ต้องการ
Pwd	Print current directory name	แสดงชื่อไดเร็คทอรี่ปัจจุบัน
Mkdir	Create directory	สร้างไดเร็คทอรี่ใหม่
Rmdir	Delete directory	ลบไดเร็คทอรี่ (ที่ว่างเปล่าเท่านั้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษาไพธอน (Python)

ไพธอน (Python) เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษาไพธอน ได้ทั้งบนระบบ Unix , Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็นภาษาลักษณะ Open Source เหมือนอย่าง PHP

ในปัจจุบันภาษาที่ใช้ในการพัฒนา Web Application มีมากมายหลายภาษา อาทิเช่น ภาษา Perl, PHP, JAVA, ASP, Tcl, Python เป็นต้น สำหรับภาษาไพธอน นับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษาไพธอน ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ โดยมีสัญลักษณ์ดังรูปที่ 2.5

รูปที่ 2.5 สัญลักษณ์ไพธอน

ที่มา http://wiki.it.kmitl.ac.th/Python_info

2.4.1 ข้อดีของภาษาไพธอน

2.4.1.1 ง่ายต่อการเรียนรู้ โดยภาษาไพธอน มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษา C มาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี

2.4.1.2 ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปรภาษาไพธอน อยู่ภายใต้ลิขสิทธิ์ GNU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.3 ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษาไพธอน ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษาไพธอน ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux , Windows 95/98/ME , Windows NT, Windows 2000 , OS/2

2.4.1.4 ภาษาไพธอน ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา C, C++, Java, Perl

2.4.1.5 ภาษาไพธอน เป็นภาษาประเภท Server Side Script คือการทำงานของภาษาไพธอน จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง

2.4.1.6 ใช้พัฒนา Web Service โดยที่ภาษาไพธอน สามารถนำมาพัฒนาเว็บเซอร์วิส รวมทั้งใช้บริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Framework (CMF) ตัวอย่าง CMF ที่มีชื่อเสียงมากและเบื้องหลังทำงานด้วยไพธอน

2.4.2 การประกาศตัวแปร

ตัวแปร (Variable) เป็นการกำหนดชนิดข้อมูลของตัวแปร เพื่อนำไปใช้ในการเขียนโปรแกรม โดยที่โปรแกรมภาษาไพธอนไปจองพื้นที่ในหน่วยความจำ เพื่อใช้เก็บข้อมูลชนิดต่าง ๆ แล้วแต่ชนิดของตัวแปรที่ประกาศเอาไว้ อาจเป็นชนิดตัวเลข ตัวอักขระ หรือสายอักขระ ข้อมูลประเภทเหล่านี้จะถูกนำไปอ้างถึงเมื่อเขียนคำสั่งไปอ้างอิง ภาษาไพธอนมีการประกาศตัวแปรไม่เหมือนเหมือนภาษาซี หรือภาษาปาสกาล ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงเปรียบเทียบการประกาศตัวแปรระหว่างภาษาต่าง ๆ

ชนิดข้อมูล	ภาษาไพธอน	ภาษาซี	ภาษาปาสกาล
Integer	i = 0 j = 5	int i; int j;	i : integer; j : integer
ชนิดข้อมูล	**ภาษาไพธอน**	**ภาษาซี**	**ภาษาปาสกาล**
Float	i = 0.0 j = 5.0	float i; float j;	j : real; j: real;
Character	ch = 'Y'	char ch;	ch : char;
String	text = 'Programming Language'	string text; .	string : text;
Boolean	Boo = True Boo = False		Boo : Boolean; Boo := true; Boo := false;

จากตารางที่ 2.3 แสดงการเปรียบเทียบการประกาศตัวแปรของภาษาต่าง ๆ เปรียบเทียบกับภาษาไพธอน ภาษาซีจะต้องบอกชนิดอย่างชัดเจน เช่น int i; แต่ภาษาไพธอนสามารถกำหนดค่าให้กับตัวแปรนั้น ๆ ได้เลย ตัวแปรของภาษาไพธอนจะเรียนรู้จากข้อมูลที่เรากำหนดให้ เช่น i = 0 หมายถึง ตัวแปร i เก็บข้อมูลประเภทตัวเลขจำนวนเต็ม (Integer) แต่การประกาศตัวแปรมีเงื่อนไขที่ต้องคำนึงตามกฎการตั้งชื่อตัวแปรของภาษาไพธอน มีดังต่อไปนี้

1) ต้องขึ้นต้นด้วยตัวอักษร ห้ามใช้ตัวเลขหรือสัญลักษณ์ใด ๆ

2) ห้ามมีช่องว่าง หรือเว้นวรรค

3) ห้ามใช้เครื่องหมายต่อไปนี้ในการตั้งชื่อตัวแปร !,@, #, $, %, ^, &, *, (,), -, =, \, |, +

4) ห้ามตั้งชื่อตัวแปรซ้ำกับคำสงวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ควรตั้งชื่อตัวแปรที่สื่อความหมายให้ชัดเจน เพื่อผู้อื่นตีความหมายได้เข้าใจ แต่ถ้ามีความยาวมากให้ย่อ เช่น student_name ควรใช้ st_name เป็นต้น

6) ตัวแปรที่มีตัวพิมพ์ใหญ่และตัวพิมพ์เล็กผสมกันจะมีความหมายต่างกับตัวพิมพ์เล็กเพียงอย่างเดียว เช่น St_Id แตกต่างจากตัวแปร st_id เป็นต้น

2.4.3 ชนิดของข้อมูล

ภาษาไพธอนจะมีชนิดข้อมูลอยู่ 9 ชนิดด้วยกัน ได้แก่ ตัวเลข ข้อมูลเรียงลำดับ การจับคู่ Callable มอดูล คลาส อินสแตนซ์คลาส ไฟล์ และ Internal ในแต่ละชนิดจะมีชนิดข้อมูลย่อยอีก แต่ที่จะนำมาศึกษาในเล่มนี้จะนำเอาชนิดของข้อมูลที่เป็นพื้นฐานที่ใช้กันบ่อย ๆ ซึ่งอยู่ด้วยกัน 4 ชนิดได้แก่ Integer, Float, String และ Dictionary ซึ่งขึ้นอยู่กับตัวแปรที่จะใช้ว่าต้องการกำหนดให้เป็นข้อมูลชนิดใด เช่น ถ้าต้องการเก็บข้อมูลของคะแนนผลการสอบที่เป็นจำนวนเต็ม จะต้องกำหนดให้ชนิดตัวแปรเป็นจำนวนเต็ม หรือเรียกว่า Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ชนิดข้อมูลของภาษาไพธอน

ชนิดข้อมูล	ชนิด	คำอธิบาย
ตัวเลข	Boolean	ตรรกะ มีค่าจริงกับเท็จ
	Integer	เลขจำนวนเต็ม
	Long integer	เลขจำนวนเต็มใช้บิตสองเท่า
	Float	เลขทศนิยม
	Complex number	จำนวนจินตภาพ
ข้อมูลแบบเรียงลำดับ	String	สายอักขระ
	Unicode	สายอักขระยูนิโคด
	ListType	รายการ
	TupleType	ทูเพิล
	XRangeType	เอกซ์แรนจ์ คืนค่าจาก xrange()
	BufferType	บัฟเฟอร์ คืนค่าจาก buffer()

2.4.4 รูปแบบการเขียนโปรแกรมไพธอน

เราสามารถเขียนโค้ดภาษาไพธอนได้โดยตรงบน Python Shell หรือจะ เขียนเก็บไว้ในเท็กซ์ไฟล์ธรรมาดาแล้วบันทึกนามสกุลเป็น .py ได้เช่นเดียวกัน ในการเขียนภาษาไพธอนไม่จำเป็นต้องมีฟังก์ชั่น main() เหมือนภาษาอื่นๆ ดังนั้น จุดเริ่มต้นการทำงานของโปรแกรม คือ โค้ดบรรทัดบนสุดในซอร์ซโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ราสเบียน (Raspbian OS)

ราสเบียน เป็นระบบปฏิบัติการที่ใช้บนราสเบอร์รี่ไพ ระบบปฏิบัติการของราสเบอร์รี่ไพในช่วงแรกๆ มีเพียงแค่แบบซอฟต์โฟลต (Soft Float) คือใช้ซอฟต์แวร์ในการคำนวณตัวเลขทศนิยม เช่น ดีเบียน บน อาร์ม หรือ ฟีโดรา รีมิกซ์ (Fedora Remix) และทางค่าย ดีเบียน ก็ไม่มีท่าทีที่จะทำการพัฒนาลีนุกซ์แบบที่ใช้กับคอมพิวเตอร์ที่ใช้หน่วยประมวลผลตระกูลอาร์มวีซิกซ์ (ARMv6) สามารถคำนวณทศนิยมด้วยฮาร์ดแวร์ในตัวชิปได้เลย ซึ่งเป็นแบบเดียวกับชิป BCM2835 ที่ราสเบอร์รี่ไพใช้อยู่ ในเวลาต่อมาได้มีดีเบียนแบบที่รองรับอาร์มฮาร์ดโฟลต (ARM Hard Float : ARMHF)ของตระกูลอาร์มวีเซเว่น (ARMv7)

ในเวลาต่อมา ไมค์ทอมสัน (Mike Thompson) ได้สร้างดีเบียนแบบอาร์มวีซิกซ์ ซึ่งเป็นแบบฮาร์ดโฟลต โดยใช้การคอมไพล์ข้ามเครื่องคอมพิวเตอร์ คือ จะใช้เครื่องคอมพิวเตอร์อื่นที่ไม่ใช่ราสเบอร์รี่ไพมาทำการคอมไพล์ ในการคอมไพล์นั้นถ้าทำบนราสเบอร์รี่ไพจะใช้เวลานาน และในการคอมไพล์จะต้องปรับแต่งคอมไพล์เลอร์ที่ใช้ในการคอมไพล์ ให้สามารถคอมไพล์โปรแกรมออกมาเป็นแบบตระกูลอาร์มวีซิกซ์ด้วย เพื่อให้ได้แพ็กเกจ (Package) ของโปรแกรมที่ทำงานได้บนราสเบียน โดยมีสัญลักษณ์ดังรูปที่ 2.6

รูปที่ 2.6 สัญลักษณ์ Raspbian OS
ที่มา http://discover.greentechnetwork.net/?p=39

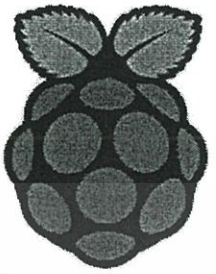
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ราสเบียน เป็นระบบปฏิบัติการที่ได้จากการโคลนมาจากดีเบียน แล้วถูกพัฒนาต่อโดยองค์กรของราสเบอร์รี่ไพ โดยจุดเด่นที่สำคัญที่สุด คือเรื่องของฮาร์ดโฟลต เนื่องจากปัจจุบัน ดีเบียนอาร์มฮาร์ดโฟลต จะให้กรรมสิทธิ์เฉพาะกับซีพียูอาร์ม ซึ่งของราสเบอรี่ไพเป็น อาร์มวีซิกซ์ ความต้องการขององค์กรของราสเบอร์รี่ไพต้องการให้ทำงานแบบฮาร์ดโฟลต และสามารถควบคุมแพ็จเกจได้ง่าย จึงเกิดเป็นราสเบอร์เรียนขึ้นมา ข้อดีของราสเบียนคือ เป็นระบบปฏิบัติการที่เหมาะสมที่สุดสำหรับราสเบอร์รี่ไพ ส่วนข้อเสียนั้นเป็นระบบปฏิบัติการมีขนาดใหญ่เนื่องจากบรรจุฟังก์ชั่นไว้มาก

2.6 ราสเบอร์รี่ไพ (Raspberry Pi)

คอมพิวเตอร์ขนาดจิ๋วตัวนี้มีชื่อว่า Raspberry Pi อ่านว่า “ราสเบอร์รี่ไพ” เป็นคอมพิวเตอร์ในบอร์ดเดียว (Single Board Computer) ถูกพัฒนาขึ้นในประเทศสหรัฐอเมริกา ซึ่งเป็นแผงวงจรที่ถูกพัฒนาขึ้นในปีพุทธศักราช 2549 โดย ดร.อีเบน อัพตัน (Eben Upton) ภายใต้ Raspberry Pi Foundation ซึ่งเป็นองค์กรไม่หวังผลกำไร แรกเริ่มนั้นถูกพัฒนาขึ้นมาให้มีราคาต่ำเพื่อใช้สำหรับการเรียนของนักศึกษาในสาขาวิทยาศาสตร์คอมพิวเตอร์ มหาวิทยาลัยเคมบริดจ์ และต้องการให้เด็ก เยาวชน และบุคคลที่มีกำลังทรัพย์น้อยได้มีโอกาสซื้อได้ แม้ว่าราสเบอร์รี่ไพมีระบบปฏิบัติการเป็นลีนุกซ์ แต่ด้วยคุณสมบัติที่มีขนาดเล็ก และสามารถพกพาได้สะดวก เพียงแค่มีจอภาพ เมาส์ คีย์บอร์ด ราสเบอร์รี่ไพก็จะกลายเป็นคอมพิวเตอร์ดีๆเครื่องหนึ่ง ที่สามารถใช้ในการค้นคว้าหาข้อมูล และฝึกพัฒนาโปรแกรมเบื้องต้นได้เป็นอย่างดี โดย ดร.อีเบน อัพตัน ใช้ระยะเวลาในการพัฒนานานถึง 6 ปี และได้ผลิตออกจำหน่ายครั้งแรกในเดือนกุมภาพันธ์ปีพุทธศักราช 2555 โดยมีสัญลักษณ์ดังรูปที่ 2.7 และ วงจรตามรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RaspberryPi

รูปที่ 2.7 สัญลักษณ์ราสเบอร์รี่ไพ

ที่มา http://www.banym.de/embedded-devices/experiences-with-my-raspberry-pi

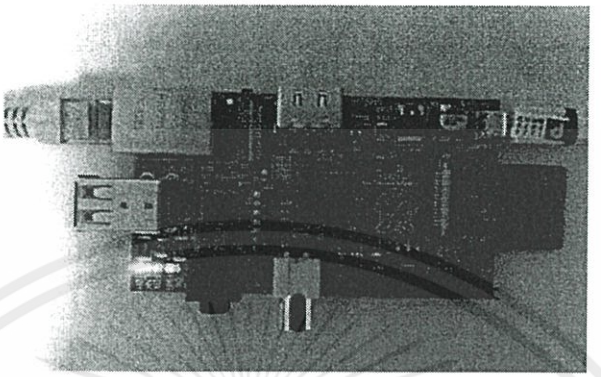
2.6.1 คุณสมบัติของราสเบอร์รี่ไพ

ราสเบอร์รี่ไพ จะมีด้วยกันอยู่สองโมเดลคือ โมเดล A และโมเดล B เราจะกล่าวถึงเพียงโมเดล B เนื่องจากโมเดล A ไม่เป็นที่นิยมในการใช้งานเท่ากับโมเดล B ซึ่งคุณสมบัติของราสเบอร์รี่ไพแสดงได้ ดังตารางที่ 2.5

ตารางที่ 2.5 คุณสมบัติของราสเบอร์รี่ไพ

Feature	Specification Model B
CPU	CPU 700MHz ARM 1176-JZF
Memory	512 MB
Video	Video Out via Composite (PAL and NTSC), HDMI or Raw CD (DSI)
Audio	Audio Out via 3.5mm Jack or Audio over HDMI
USB	2 x USB2.0
Storage	SD/MMC/SDIO
Networking	10/100 Ethernet (RJ45)
Power	5V DC micro USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

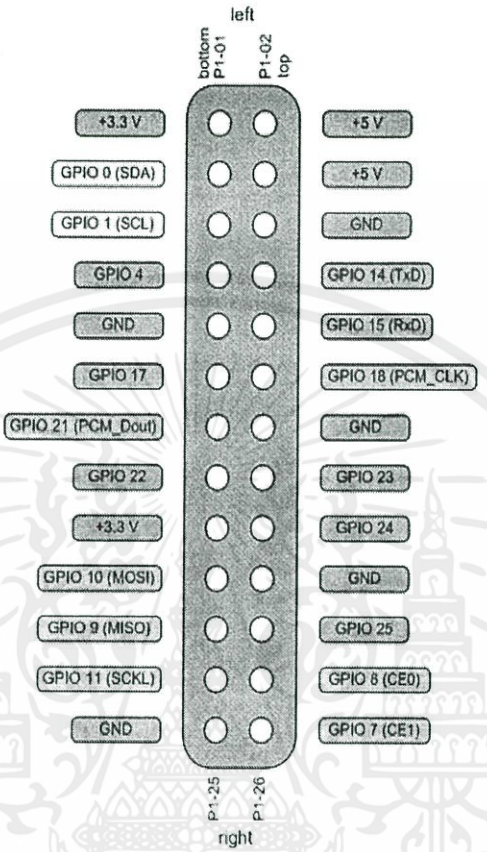


รูปที่ 2.8 บอร์ดวงจรราสเบอร์รี่ไพ

2.6.2 พอร์ตอินพุตเอาต์พุตอเนกประสงค์ (General Purpose Input Output : GPIO)

ด้วยคุณสมบัติที่โดดเด่นของราสเบอร์รี่ไพ นอกจากความสามารถที่สูงกว่าราคาแล้ว นั่นคือ ความยืดหยุ่นในการเชื่อมต่อกับอุปกรณ์ภายนอก ราสเบอร์รี่ไพมีพอร์ตอินพุตเอาต์พุตอเนกประสงค์ไว้รองรับให้ใช้งานถึง 26 ขา (โมเดล B) โดยประกอบด้วยขาพอร์ตอินพุตเอาต์พุตดิจิตอลปกติ, ขาเชื่อมบัสไอสแควร์ซี และ SPI จึงทำให้ราสเบอร์รี่ไพเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์ได้หลากหลาย ตั้งแต่พื้นฐานอย่างหลอดแอลอีดี และสวิตช์ ไปจนถึงไอซีสำหรับแปลงสัญญาณจากอนาล็อกเป็นดิจิตอล ไอซีขยายพอร์ตอินพุตเอาต์พุต และตัวตรวจจับต่างๆที่เชื่อมต่อผ่านระบบบัสไอสแควร์ซี ซึ่งการจัดเรียงขาของพอร์ตอินพุตเอาต์พุตอเนกประสงค์แสดงได้ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การจัดเรียงขาของราสเบอร์รี่ไพ

2.7 User Datagram Protocol (UDP)

User Datagram Protocol (UDP) เป็นวิธีการสื่อสารหรือโปรโตคอลที่จำกัดจำนวนการบริการ เมื่อข่าวสารมีการแลกเปลี่ยน ระหว่างคอมพิวเตอร์ในเครือข่ายที่ใช้ Internet Protocol (IP) โดย UDP เป็นตัวเลือกหนึ่งของ Transmission Control Protocol (TCP) และใช้ร่วมกับ IP บางครั้งเรียกว่า UDP/IP ซึ่ง UDP เหมือนกับ TCP ในการใช้ IP ในการดึงหน่วยข้อมูล (เรียกว่า Datagram) จากคอมพิวเตอร์หนึ่งไปยังอีกเครื่องหนึ่ง แต่ต่างจาก TCP โดย UDP ไม่ให้การบริการสำหรับการแบ่งข่าวสารเป็นแพ็คเกต (Datagram) และประกอบขึ้นใหม่เมื่อถึงปลายหนึ่ง UDP ไม่ให้ชุดของแพ็คเกตที่ข้อมูลมาถึง หมายความว่า โปรแกรมประยุกต์ที่ใช้ UDP ต้องมีความสามารถในการสร้างมั่นใจว่าข่าวสารที่มาถึงอยู่ในลำดับที่ถูกต้อง การประยุกต์เครือข่ายที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประหยัดเวลาในการประมวลผล เพราะมีหน่วยข้อมูลในการแลกเปลี่ยน (ดังนั้น จึงมีข่าวสารน้อยมากในการประกอบขึ้นใหม่) จะชอบ UDP มากกว่า TCP ซึ่ง Trivial File Transfer Protocol (TFTP) ใช้ UDP แทนที่ TCP

UDP ให้ 2 บริการที่ไม่มีโดยเลเยอร์ของ IP คือ Port Number เพื่อช่วยแยกแยะการขอของผู้ใช้ และความสามารถ Checksum เพื่อตรวจสอบการมาถึงข้อมูล ในแบบจำลองการสื่อสาร Open System Interconnection (OSI) UDP เหมือนกัน TCP คือ อยู่ที่เลเยอร์ 4 Transport Layer

2.7.1 วิธีการใช้ User Datagram Protocol (UDP)

ถึงแม้ว่า UDP ไม่ได้จัดการบริการต่างๆ ที่อยู่เหนือกว่า Application Layer Protocol Identification และ Checksum แต่ก็ใช้ในการส่งข้อมูลโดยใช้ UDP ดังประกอบไปด้วย

1) Lightweight Protocol เพื่อที่จะติดต่อหน่วยความจำและแหล่งประมวลผลทรัพยากร บาง Application Layer Protocol ต้องการการใช้โพรโตคอลที่เล็กกว่าที่ทำหน้าที่สำหรับรูปแบบการใช้งานเฉพาะในการแลกเปลี่ยนข้อมูลเบื้องต้น ตัวอย่างที่ดีที่สุดคือการสอบถามชื่อของ Domain Name System (DNS) ที่เป็นแบบฉบับคือ DNS Client จะส่งการร้องขอข้อมูลชื่อของ DNS ไปยัง DNS Server DNS Server ก็จะตอบกลับด้วยการตอบกลับข้อมูลชื่อของ DNS ถ้า DNS Server ไม่มีการตอบสนอง DNS Client ก็จะส่งการร้องขอข้อมูลชื่อของ DNS ไปอีกครั้ง

2) Reliability Provided by the Application Layer Protocol ถ้า Application Layer Protocol จัดการให้บริการการส่งข้อมูลของตัวเองน่าเชื่อถือ ทำให้ Transport Layer ไม่จำเป็นต้องจัดการในส่วนนี้ ยกตัวอย่าง Application Layer Protocol ที่น่าเชื่อถือคือ Trivial File Transfer Protocol (TFTP) และ Network File System (NFS)

3) Reliability not required due to periodic advertisement process ถ้า Application Layer Protocol ประกาศข้อมูลเป็นช่วงๆ การส่งข้อมูลที่น่าเชื่อถือจะไม่เกิดขึ้น ถ้าการประกาศล้มเหลวจะประกาศอีกครั้งเป็นช่วงๆ ตามช่วงเวลา ตัวอย่างของ Application Layer Protocol ที่ใช้ประกาศข้อมูลเป็นช่วงๆ คือ Routing Information Protocol (RIP)

4) One-to-many delivery UDP สามารถใช้เป็น Transport Layer Protocol ได้เมื่อไหร่ก็ตามที่ข้อมูลของ Application Layer จำเป็นต้องส่งไปยังหลายๆ ปลายทางโดยใช้ IP Multicast หรือ Broadcast Address TCP สามารถใช้ได้กับการส่งแบบหนึ่งต่อหนึ่งเท่านั้น ยกตัวอย่างเช่น ต้นทางส่งชื่อ Broadcast NetBIOS เรียกขอข้อความตอบกลับโดยใช้ UDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Real-time Transport Protocol (RTP)

RTP ย่อมาจาก Real Time Protocol เป็นโปรโตคอลหนึ่งในระบบ VoIP ทำหน้าที่รับส่งสัญญา เสียงพูด (Voice) แฟ็กซ์ (Fax) ข้อความ (Message) และวีดีโอ (Video) ระหว่างต้นทางและปลายทาง โปรโตคอลนี้ทำงานบนพื้นฐานของ UDP (User Datagram Protocol) โดยทั่วไป VoIP Server ส่วนใหญ่เราจะตั้ง RTP Ports เองได้ ทุกๆ 1 คอลไม่ว่าจะเป็นระหว่างต้นทางถึงปลายทาง ระหว่างต้นทางถึง VoIP Server ที่อยู่ตรงกลางก็ตาม จะใช้ RTP จำนวน 2 พอร์ต ได้แก่

1. RTP Port หรือจะเรียกว่า Media Port เป็นพอร์ตเลขคู่ สำหรับรับส่งสัญญาณเสียง แฟ็กซ์ ข้อความ หรือวีดีโอ
2. RTCP Port หรือเรียกว่า Media Control Port เป็นพอร์ตเลขคี่ที่อยู่ติดกัน ทำหน้าที่ควบคุม Media เช่นลำดับการรับส่ง และเพื่อควบคุมคุณภาพการบริการ(QoS - Quality of Service)โดยที่ทั้ง 2 พอร์ตเราเรียกรวมกันเป็น 1 RTP Stream

ข้อมูลที่อยู่ใน RTCP มักจะประกอบไปด้วยข้อมูลดังต่อไปนี้ Packet Loss, Jitter, Delay, Signal Level, Call Quality Matrics, Echo Return Loss และบางครั้งอาจจะมีการใช้ RTCP XR (Real Time Control Protocol Extended Reports) ร่วมด้วยเพื่อเพิ่มรายงานเข้าไปด้วย ได้แก่ R Factor, MOS

RTP ถูกใช้โดย SIP และ H.323 เพื่อทำงานด้านของการสื่อสารแบบเรียลไทม์ เช่น ออดิโอและวีดีโอบนเครือข่ายแพ็กเก็ตสวิตชิง มีหน้าที่จัดการเรื่องข้อมูลประเภทเวลาไปยังผู้รับ โดยสามารถแก้ไขค่าดีเลย์ของสัญญาณได้ยอมให้ผู้รับสามารถค้นหาแพ็กเก็ตที่สูญหาย และประเมินเส้นทางในการส่งข้อมูลอีกด้วยนอกจากนี้แล้ว RTP จะมีหน้าที่จัดการในเรื่องของการส่งข้อมูลไปยังผู้รับ ซึ่งสามารถกู้คืนได้ในกรณีที่แพ็กเก็ตสูญหายหรือ jitter โดย RTP ได้รับการประกาศใช้จาก IETF ใน RFC 1889 ซึ่งหน้าที่หลักก็เพื่อให้บริการฟังก์ชันต่าง ๆ เช่น การจัดลำดับ การกำหนด payload และ intra-media synchrinization กับReal-time Transport Control Protocol (RTCP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1หลักการทำงานของ RTP (Real Time Transport Protocol)

RTP จะทำงานในระบบ end-to-end network การทำงานของ RTPจะทำงานไม่คำนึงถึง QoS (Quality of Service) โดยใช้กับการส่งแบบ real-time ข้อมูลที่ส่งจะถูกจัดการโดย RTCP (Real-time Control Protocol) เพื่อที่จะส่งในระบบ network RTP สามารถส่งข้อมูลได้แบบ Unicast และ Multicast RTP ถูกออกแบบมาโดยไม่ขึ้นกับ transport และ network layers ความสามารถใช้ในการส่งข้อมูลผ่านเครือข่ายสำหรับการส่งข้อมูลพวกใช้เวลาจริง (Real Time) เช่นทางวีดิทัศน์ (Video) เป็นโปรโตคอลที่ใช้สำหรับส่งข้อมูล audio และ audio แบบ real-time ซึ่งไม่มีข้อกำหนดที่แน่นอนสามารถใช้ได้ทั้ง TCP และ UDP ซึ่งโดยทั่วไปจะใช้ UDP นอกจากนั้นยังไม่มีช่วงของ port ที่แน่นอน ข้อมูลที่ส่งจะถูกควบคุมด้วยReal Time Control Protocol (RTCP) RTP เป็นแบบ Connectionless ไม่มีการรับประกันคุณภาพของข้อมูลที่ส่ง หมายความว่าไม่ได้มีกลไกใดๆ ในการยืนยันข้อมูลว่าส่งได้สำเร็จหรือไม่ ไม่มีความผิดพลาดในการเรียงลาดับข้อมูล ซึ่งแตกต่างจากโปรโตคอล UDP เมื่อทำการส่งแล้วมี ปัญหาในการลำดับก่อนหลังของเฟรม

2.8.2ประโยชน์ของโปรโตคอล RTP มีดังต่อไปนี้

RTP ใช้กำหนดรูปแบบ packet ในการส่งภาพและเสียงผ่านอินเตอร์เน็ต ถูกพัฒนาโดย Audio-Video Transport Working Groupของ IETF และได้ตีพิมพ์ครั้งแรกในปี 1996 (พ.ศ. 2539) โดย RTP จะไม่มีพอร์ต TCP หรือ UDP มาตรฐานในการสื่อสาร แต่จะใช้พอร์ต UDP ที่เป็นเลขคู่ในการสื่อสารและพอร์ต UDP เลขคี่ถัดไปเป็น RTP Control Protocol (RTCP) เลขพอร์ตมักอยู่ระหว่าง163834-32767 RTP สามารถรับส่งข้อมูลอะไรก็ได้แบบ real time เช่น ภาพและเสียง โดยใช้โปรโตคอล SIP ในการตั้งค่าและยกเลิก

2.9 Real-time Transport Control Protocol (RTCP)

RTCP เป็นโพรโตคอลที่ทำงานร่วมกับ RTP เพื่อควบคุมทำงานในเรื่อง QoS Feedback การควบคุมเซสชัสของการส่งข้อมูล การกำหนดเกี่ยวกับผู้ใช้ และ inter media synchrinization เพื่อเข้าจังหวะ ระหว่างออดิโอและวีดีโอสตรีม RTCP เป็นโปรโตคอลที่ถูกกำหนดอยู่ใน RFC 3550 เช่นเดียวกับ RTP โปรโตคอลตัวนี้ใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ RTCP ใช้ช่องสื่อสารแยกกันกับ RTP โดย TRCP นั้นใช้พอร์ตถัดจาก RTP ไปหนึ่งพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องสื่อสารของ RTCP นั้นใช้สำหรับการรายงานเกี่ยวกับคุณภาพการบริการ เช่น จำนวนแพ็กเก็ตที่ได้รับจำนวนแพ็กเก็ตที่สูญหาย จิตเตอร์ เป็นต้น แพ็กเก็ต RTCP นั้นแบ่งออกเป็น 5 ชนิดได้แก่

- SR (Sender Report) สำหรับบอกสถิติเกี่ยวกับการส่งข้อมูลของผู้ส่ง
- RR (Receive Report) สำหรับบอกสถิติเกี่ยวกับการรับข้อมูลของผู้รับ
- SDES (Source description items) เป็นรายละเอียดต่างๆของผู้ส่ง
- BYE สำหรับแจ้งถึงการจบการทำงาน
- APP (Application Specific Functions) สำหรับการกำหนดการทำงานบางอย่างในส่วนของโปรแกรมประยุกต์ ซึ่งไม่ได้ระบุเอาไว้เป็นมาตรฐาน

แพ็กเก็ต RTCP ที่น่าสนใจก็คือ RR เพราะว่าผู้รับจะรวบรวมสถิติต่างๆของข้อมูลเสียงหรือวีดิทัศน์ที่ได้รับจากแพ็กเก็ต RTP แล้วบรรจุลงในแพ็กเก็ต RR นี้และส่งให้กับผู้ส่ง ผู้ส่งสามารถใช้ข้อมูลในแพ็กเก็ตชนิดนี้ในการประเมินสถานะของเครือข่าย รวมถึงคุณภาพของสื่อที่ผู้รับได้รับ ในการออกแบบวิธีการควบคุมคุณภาพแบบปรับตัวของทั้งการสื่อสารเสียงและวีดิทัศน์ของวิทยานิพนธ์นี้ก็จะมีการใช้งานแพ็กเก็ตนี้เช่นกัน ในรูปที่ 2.9 เป็นโครงสร้างเฮดเดอร์ของแพ็กเก็ต RR ซึ่งฟิลด์ต่างๆที่กำหนดอยู่ในมาตรฐานแล้ว โครงสร้างเฮดเดอร์ของแพ็กเก็ต RTCP ยังอนุญาตให้มีการใส่ข้อมูลเพิ่มเติมลงในแพ็กเก็ตนี้ได้อีก โดยข้อมูลส่วนที่เพิ่มเติมนี้สามารถใส่ได้ในตำแหน่งที่เรียกว่า Profile-Specific Extension

2.10 Real-Time Streaming Protocol (RTSP)

RSTP เป็นโปรโตคอลที่ใช้รับส่งข้อมูลมัลติมีเดียระหว่างเซิร์ฟเวอร์กับคอมพิวเตอร์ปลายทาง ซึ่งจะทำให้การรับส่งข้อมูลต่อเนื่องผ่านอินเตอร์เน็ต โดยตัวเซิร์ฟเวอร์ด้านผู้ส่งสามารถส่งข้อมูลไปให้ผู้รับปลายทางเพียงคนเดียว หรือจะส่งไปให้ผู้รับหลายๆ คนในลักษณะเป็นกลุ่มก็ได้ ซึ่ง RTSP ถูกกำหนดให้เป็นโปรโตคอลที่นำไปใช้ในอินเตอร์เน็ตโดย Internet Engineering Task Force (IETF) ในเดือนเมษายน ปี ค.ศ. 1998 เรียกว่า RFC 2326 RTSP เป็นการกำหนดมาตรฐานโปรโตคอลที่สำคัญมากในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ต เนื่องจากการรับส่งข้อมูลมัลติมีเดียในแบบต่อเนื่องนั้นจะมีส่วนต่างๆ ที่เกี่ยวโยงกันถึง 3 ส่วน คือ เซิร์ฟเวอร์ที่เก็บข้อมูล , Encoder ที่ใช้เข้ารหัสข้อมูล และผู้รับข้อมูลหรือ Player ตัวเข้ารหัสข้อมูลหรือ Encoder นั้นจะต้องเข้ารหัสข้อมูลมัลติมีเดียเก็บลงในไฟล์โดยมีฟอร์แมตที่เซิร์ฟเวอร์ที่เรียกใช้งานได้ และเมื่อเซิร์ฟเวอร์ต้องการส่งข้อมูลนี้ไปให้ผู้รับ ก็จะต้องใช้โปรโตคอลรับส่งข้อมูลอย่างต่อเนื่องที่ผู้รับเข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถรับข้อมูลได้อย่างถูกต้อง จากนั้นเมื่อได้รับข้อมูลมาแล้วก็จะต้องถอดรหัสข้อมูลออกแสดงผลได้ โดยใช้มาตรฐานเดียวกันกับตัวเข้ารหัส การทำงานทั้งหมดจึงผูกกัน ซึ่ง RTSP จะอยู่ในส่วนโปรโตคอลที่ใช้รับส่งข้อมูลจากดซิร์ฟเวอร์ไปให้ผู้รับนั่นเอง แม้ว่า RTSP จะมีความสำคัญในการรับส่งข้อมูลมัลติมีเดียผ่านอินเตอร์เน็ตก็ตาม แต่ก็ไม่ใช่สิ่งเดียวที่ทำให้การรับส่งข้อมูลสมบูรณ์ได้ เรายังต้องการฟอร์แมตมาตรฐานของไฟล์ที่ใช้เก็บข้อมูลมัลติมีเดียอีกด้วย เช่น Active Streaming Format (ASF) ของไมโครซอฟต์ , QuickTime หรืออื่นๆ เพื่อเก็บข้อมูล รวมถึงมาตรฐานการเข้ารหัสข้อมูล เช่น MPEG สำหรับใช้เข้ารหัสข้อมูลมัลติมีเดียเก็บลงไฟล์อีกด้วย

RTSP จะทำหน้าที่ควบคุมการรับส่งข้อมูลมัลติมีเดียอย่างต่อเนื่อง ระหว่างเซิร์ฟเวอร์ที่เก็บข้อมูลกับคอมพิวเตอร์ผู้รับข้อมูล โดยมีไฟล์ฟอร์แมตสำหรับเก็บข้อมูล เช่น ASF , QuickTime หรืออื่นๆ ซึ่งการสร้างไฟล์ข้อมูลมัลติมีเดียนี้ก็ได้มาจากการนำข้อมูลมาทำการเข้ารหัส โดยใช้ตัวเข้ารหัส เช่น MPEG นั่งเอง ซึ่งในปัจจุบันซอฟต์แวร์ที่ใช้เล่นข้อมูลมัลติมีเดียก็ใช้ RTSP ในการรับส่งข้อมูลทั้งหมดทั้งนั้น เช่น QuickTime Version 4 , Real System G2 และ Windows Media Player เป็นต้น เราอาจสงสัยว่าทำไมไม่ใช้ Hyper Text Transfer Protocol หรือ HTTP ในการรับส่งมัลติมีเดียผ่านอินเตอร์เน็ต เหตุผลก็คือ HTTP เป็นโปรโตคอลที่ถูกออกแบบมาให้รับส่งข้อมูลผ่าน TCP ซึ่งเน้นหนักไปทางความเชื่อถือได้ในการรับส่งข้อมูล (คือข้อมูลไม่สูญหายระหว่างทาง) โดยไม่สนใจเวลาที่ใช้รับส่งข้อมูล ดังนั้นเมื่อนำ HTTP มาใช้รับส่งมัลติมีเดียที่ต้องการความต่อเนื่องของข้อมูลก็จะเกิดปัญหาขึ้น คือข้อมูลที่ได้รับอาจหายไปเป็นช่วงๆ จึงต้องใช้ RTSP แทน เพื่อให้รับส่งข้อมูลได้อย่างต่อเนื่องและสามารถรับส่งข้อมูลในลักษณะกระจาย (multicast) ได้ ทำให้ RTSP เหมาะสำหรับการรับส่งข้อมูลภาพ และเสียงผ่านอินเตอร์เน็ตมากกว่า

2.11 เว็บเซิร์ฟเวอร์ (Wed Server)

เว็บเซิร์ฟเวอร์ (Wed Server) หัวใจสำคัญของทุกเว็บไซต์ที่จะต้องมี สำหรับเทคโนโลยีบนโลกไอทีที่ต้องออนไลน์ เว็บไซต์เป็นสิ่งจำเป็นในหลายหน่วยงาน หรือองค์กร เราสามารถถ่ายทอดข่าวสารผ่านเว็บไซต์ ไม่ว่าจะเป็นข้อมูลต่างๆ ความรู้ ความบันเทิง ประกาศ และการประชาสัมพันธ์ เบื้องหลังของเว็บไซต์ต่างๆเหล่านี้ต้องทำงานอยู่บนเว็บเซิฟเวอร์ เพื่อจะรัน Script ให้เราได้ดูและเข้าใจในสิ่งที่เว็บไซต์นั้นๆสื่อให้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.1 ความหมายของ เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์ (Wed Server) คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องบริการเว็บเพจแก่ผู้ร้องขอด้วยโปรแกรมประเภทเว็บบราวเซอร์ (Web Browser) ที่ร้องขอข้อมูลผ่านโปรโตคอลเฮชทีทีพี (HTTP = Hyper Text Transfer Protocol) เครื่องบริการจะส่งข้อมูลให้ผู้ร้องขอในรูปของข้อความ ภาพ เสียง หรือสื่อผสม เครื่องบริการเว็บเพจมักเปิดบริการพอร์ท 80 (HTTP Port) ให้ผู้ร้องขอได้เชื่อมต่อและนำข้อมูลไปใช้ เช่น โปรแกรมอินเทอร์เน็ตเอ็กโพเลอร์ (Internet Explorer) หรือ ฟายฟร็อก (FireFox Web Browser) การเชื่อมต่อเริ่มด้วยการระบุที่อยู่เว็บเพจที่ร้องขอ (Web Address หรือ URL = Uniform Resource Locator) เช่น http://www.google.com หรือ http://www.thaiall.com เป็นต้น โปรแกรมที่นิยมใช้เป็นเครื่องบริการเว็บ คือ อาปาเช่ (Apache Web Server) หรือไมโครซอฟท์ไอไอเอส (Microsoft IIS = Internet Information Server) ส่วนบริการที่นิยมติดตั้งเพิ่ม เพื่อเสริมความสามารถของเครื่องบริการ เช่น ตัวแปลภาษาสคริปต์ ระบบฐานข้อมูล ระบบจัดการผู้ใช้ และระบบจัดการเนื้อหา เป็นต้น

2.12 สเต็ปปริ้งมอเตอร์ (Steppringmotor)

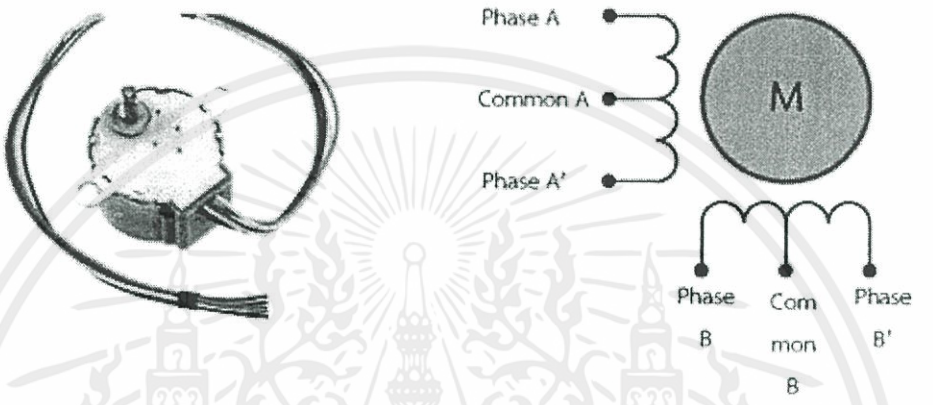
สเต็ปเปอร์มอเตอร์ คือ มอเตอร์ที่มีการหมุนเป็นขั้นๆ โดยแกนของมอตอร์จะหมุนด้วยมุมค่าคงที่เมื่อมีสัญญาณ Pulse มากระตุ้น เช่น 0.9, 1.8, 5, 7.5, 15 หรือ 45 องศาต่อสเต็ป เนื่องจากการควบคุมสเต็ปเปอร์มอเตอร์สามารถใช้สัญญาณดิจิตอลควบคุมได้โดยตรง และสามารถบังคับทิศทางและความเร็วของแกนหมุนสเต็ปเปอร์มอเตอร์เป็นที่นิยมมาใช้ในงานควบคุม เช่น เครื่องปริ้นเตอร์, X-Y Table

2.12.1 ข้อดีของสเต็ปปริ้งมอเตอร์

1) หมุนได้ครบ 360 องศา ต่อเนื่อง
2) การควบคุมตำแหน่งไม่ต้องอาศัยตัวตรวจจับการหมุน
3) ไม่ต้องใช้แปรงถ่านดังนั้นจึงทำให้ไม่มีส่วนที่สึกหรอ และปัญหาจาก ประกายไฟ (ที่เกิดจากหน้าสัมผัสของแปรงถ่านกับแหวนตัวนำในโรเตอร์ที่ทำให้เกิดสัญญาณรบกวน)
4) การควบคุมโดยทางวงจรดิจิตอลหรือไมโครคอนโทรลเลอร์ทำได้ง่าย และสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็ปปิ้งมอเตอร์ที่จะนำมาใช้ในการทดลองนี้ จะใช้สเต็ปปิ้งแบบยูนิโพล่าร์ (Uni -Polar Stepper Motor) ดังรูปที่ 2.10



รูปที่ 2.10 สเต็ปปริ้มมอเตอร์แบบยูนิโพล่าร์

ที่มา http://aimagin.com/blog/wp-content/uploads/2013/11/img1-1.jpg

2.12.2 การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์แบบ Unipolar

สามารถทำได้โดย ควบคุมกระแสไฟที่จ่ายให้กับขดลวดในแต่ละเฟส อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆเป็นสถานะลอจิก “1” การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์สามารถทำได้ 3 วิธีดังนี้

1) การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive เป็นการป้อนกระแสไฟให้กับขดลวดของสเต็ปเปอร์มอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไปตามตารางที่ 2.6 (ตัวเลข 1 หมายถึง มีการจ่ายแรงดันให้กับ ขดลวด 0 คือไม่จ่ายแรงดันให้กับขดลวด) ดังนั้นกระแสที่ไหลในขดลวดจะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเต็ปปิ้งมอเตอร์มีน้อย จึงไม่เป็นที่นิยมใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ลำดับการป้อนกระแสแบบเฟสเดียว

Step	Phase A	Phase B	Phase A'	PhaseB'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
6	Back to step 1			

2)การควบคุมการหมุนแบบสองเฟส(Two Phase) หรือ Full Step Drive เป็นการป้อนกระแสไฟให้กับขดลวดของสเต็ปเปอร์มอเตอร์ทั้ง 2 ขด พร้อมๆ กันไป และป้อนกระแสเรียงตามลำดับกันไป ตามตารางที่ 2.7 ดังนั้นจึงมีกระแสไหลในขดลวดของมอเตอร์มากขึ้น และจะทำให้มอเตอร์มีแรงบิดมากขึ้น

ตารางที่ 2.7 ลำดับการป้อนกระแสแบบสองเฟส

Step	Phase A	Phase B	Phase A'	PhaseB'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
6	Back to step 1			

3) การควบคุมการหมุนแบบครึ่งเฟส (Half Step)

เป็นการป้อนกระแสแบบ Two Phase และ Single Phase สลับการไป ตามตาราง ที่ 2.8 ซึ่งเป็นการเพิ่มความละเอียดของตำแหน่งในการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 ลำดับการป้อนกระแสแบบครึ่งเฟส

Step	Phase A	Phase B	Phase A'	PhaseB'
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	Back to step 1			

ความเร็วการหมุนของแกนสเต็ปเปอร์มอเตอร์ จะขึ้นอยู่กับการหน่วงเวลา (Time Delay) ในแต่ละลำดับ ถ้า Time Delay มีค่าน้อย แกนจะหมุนเร็วขึ้น ถ้า Time Delay มีค่ามาก แกนจะหมุนช้าและหมุนไม่สม่ำเสมอ

2.13 เซนเซอร์ (senser)

เซนเซอร์ (Sensor) คือตัวอุปกรณ์ตรวจรู้ตัวแรกในระบบการวัด ซึ่งใช้ตรวจจับหรือรับรู้การเปลี่ยนแปลงปริมาณทางกายภาพของตัวแปรต่างๆ เช่น ความร้อน แสง สี เสียง ระยะทาง การเคลื่อนที่ ความดัน การไหล เป็นต้น แล้วเปลี่ยนให้อยู่ในรูปของสัญญาณหรือข้อมูลที่สอดคล้องและเหมาะสมกับส่วนของการกำหนดเงื่อนไขทางสัญญาณ ถ้าเป็นการวัดแบบสัมผัสกับตัวแปรโดยตรง เรียกตัวตรวจรู้แบบปฐม (Primary Sensors) หรือตัวตรวจรู้ชั้นต้น หากมีการตรวจรู้โดยผ่านส่วนอื่นก่อน เช่น สเตรนเกจตรวจรับแรงกดที่ต้องรับแรงถ่ายถอดจากแท่งโลหะที่รับแรงโดยตรงอีกทอดโดยใช้เตรนเกจแปะติดกับแท่งโลหะดังกล่าวเพื่อวัดแรงนั้น เราจะเรียก สเตรนเกจในกรณีนี้ว่าเป็น ตัวตรวจรู้ทุติยภูมิ (Secondary Sensor) หรือตัวตรวจจับชั้นรองการตรวจรู้จะอาศัยผลการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงของพารามิเตอร์ในตัวเซนเซอร์เองที่สามารถตรวจวัดได้ซึ่งส่วนใหญ่จะเป็นพารามิเตอร์ทางไฟฟ้า เช่น แรงดัน กระแส ความต้านทาน ความจุ และความเหนี่ยวนำ เป็นต้น
เมื่อค่าตัวแปรเปลี่ยนแปลงแล้วพารามิเตอร์ดังกล่าวจะเปลี่ยนตาม ทำให้สามารถวัดและทราบค่าพารามิเตอร์ทางไฟฟ้าที่เปลี่ยนตามได้ ซึ่งเราอาจวัดได้โดยใช้มิเตอร์หรือวงจรบริดจ์ต่าง ๆ ซึ่งเป็นการวัดตัวแปรด้วยวิธีทางไฟฟ้าโดยเราจะทำการเทียบหรือปรับแต่งปริมาณทางไฟฟ้านี้แทนค่าตัวแปรที่ทำการวัดอีกที เราจึงอาจเรียกว่าเป็นการวัดโดยวิธีอ้อมได้ กระบวนการนี้เรียกว่าการตรวจจับ (Sensing) กรณีนี้คำว่าทรานสดิวเซอร์จะถูกเรียกว่า เซนเซอร์ จะเห็นว่าทรานสดิวเซอร์และเซนเซอร์แท้จริงคืออุปกรณ์ที่ทำหน้าที่เดียวกันต่างตรงที่เราจะกล่าวถึงหลักการทำงานหรือกล่าวถึงลักษณะการใช้งาน

2.13.1 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)

เซ็นเซอร์ตรวจจับความเคลื่อนไหว เป็นอุปกรณ์ที่แปลงการตรวจจับความเคลื่อนไหวเป็นสัญญาณไฟฟ้า โดยทั่วไปเซ็นเซอร์ตรวจจับความเคลื่อนไหวมี 3 ประเภทคือ

1) Passive Infrared Sensors (PIR) เป็นเซ็นเซอร์ที่รับความร้อนจาก ร่างกายเมื่อเคลื่อนที่ ไม่มีการปล่อยพลังงานออกมาจากเซ็นเซอร์

2) Ultrasonic เป็นเซ็นเซอร์ที่มีการปล่อยคลื่นอัลตร้าโซนิกออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่

3)Microwave เป็นเซ็นเซอร์ที่มีการปล่อยคลื่นไมโครเวฟออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่

ในปริญญานิพนธ์นี้ได้เลือกใช้เซ็นเซอร์ตรวจจับความเคลื่อนไหวประเภท Passive Infrared Sensors

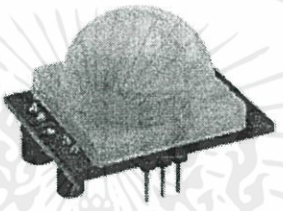
2.13.2 Passive infrared Sensors (PIR Sensor)

เป็นอุปกรณ์ที่ตรวจจับความเคลื่อนไหวด้วยการตรวจวัดความร้อนในพื้นที่ที่ต้องการ ความร้อนที่วัดได้จากการเปลี่ยนแปลงระดับของรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ เมื่อวัตถุเคลื่อนที่ สิ่งมีชีวิตทุกชนิดจะแผ่รังสีอินฟราเรดออกมาจากตัวเอง การแผ่รังสีดังกล่าว เกิดจากการเคลื่อนที่ของอิเล็กตรอนในอะตอม ปริมาณรังสีจะมีมากน้อยตามแต่โครงสร้างทางเคมี และอุณภูมิ ของวัตถุหรือสิ่งมีชีวิตนั้นๆ จึงทำให้สามารถตรวจจับสัญญาณลอจิกที่เปลี่ยนแปลงที่ขาเอาต์พุตได้

2.13.2.1ส่วนประกอบที่สำคัญของ PIR Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เลนส์ - สำหรับควบคุมหรือโฟกัสพื้นที่ในการตรวจจับความเคลื่อนไหว
2) เซ็นเซอร์ - เป็นตัวแปลงพลังงานความร้อนจากรังสีอินฟราเรด มาเป็นสัญญาณทางไฟฟ้า



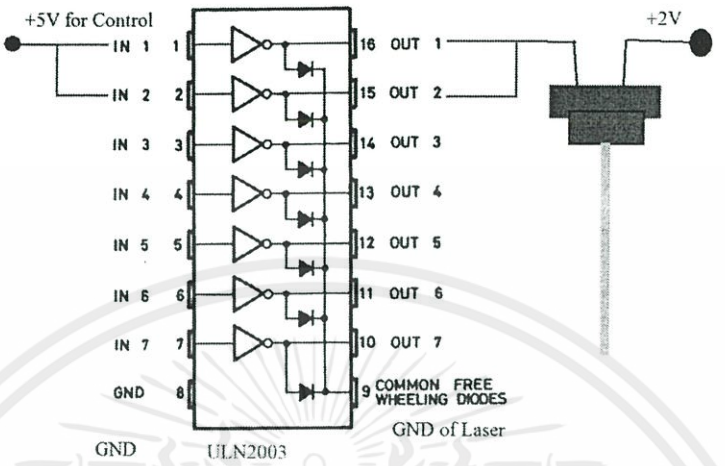
รูปที่ 2.11 โมดูลตรวจความเคลื่อนไหว (PIR sensor)

PIR Sensor เป็นแผงวงจรตรวจจับความเคลื่อนไหวด้วยการตรวจวัดความร้อน สามารถวัดได้ไกลถึง 6 เมตร มีขนาดเล็ก ถูกออกแบบมาให้ใช้งานกับไมโครคอนโทรลเลอร์ได้ง่ายโดยใช้ขาเชื่อมต่อเพียง 1 ขา และสามารถเลือกโหมดสัญญาณเอาต์พุตได้ ดังรูปที่ 2.11

2.14 ไอซี ULN2003

ไอซี ULN2003 เป็นไอซีทำหน้าที่ขยายแรงดันและกระแสแล้วส่งสัญญาณออกทางเอาต์พุตเพื่อขับโหลด ซึ่ง ไอซี ULN2003 จะทำหน้าที่ขับมอเตอร์ ซึ่ง ไอซี ULN2003 มีคุณสมบัติคือ สามารถขับโหลดได้ถึง 500 mA และทนแรงดันออกได้สูงถึง 50 โวลต์ นอกจากนี้ยังมีไดโอดต่อป้องกันไว้ภายในสำหรับขับรีเลย์ได้เลย และใช้สำหรับอินพุทแบบ TTL หรือ CMOS ขนาด 5V วงจรนี้ต้องมีแหล่งจ่ายไฟจากภายนอกจ่ายไฟเลี้ยงให้กับมอเตอร์ ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ขาสัญญาณต่าง ๆ ของ IC ULN2003

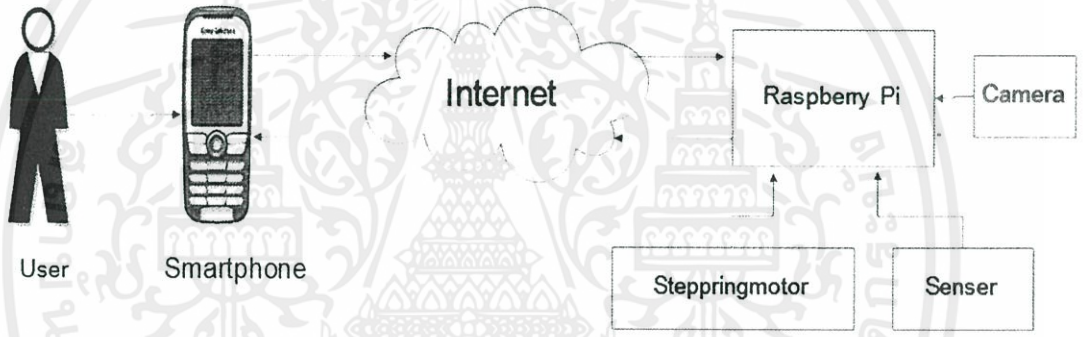
ที่มา http://www.cncroom.com/forum/index.php?topic=1608.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

สำหรับปริญญานิพนธ์นี้มีกระบวนการทำงานของระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์บนสมาร์ทโฟน ดังนี้เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวสามารถตรวจจับการเคลื่อนไหวได้ ตัวราสเบอร์รี่ไพจะทำการส่งแพ็คเกตข้อมูลมาที่ตัวโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เมื่อโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ได้รับแพ็คเกตข้อมูลก็จะทำการสร้างโนติฟิเคชั่น (NOTIFICATION) เพื่อแจ้งเตือนผู้ใช้ เมื่อผู้ใช้ได้รับการแจ้งเตือนก็จะทำการเปิดแอพพลิเคชั่นเพื่อทำการเปิดดูภาพจากกล้อง โดยผู้ใช้สามารถควบคุมทิศทางของกล้องได้จากตัวแอพพลิเคชั่นได้ โดยมีทิศทางการเคลื่อนที่ไปทางด้านบน-ล่าง ซ้าย-ขวา ดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ
ควบคุมกล้องไร้สายผ่านโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์

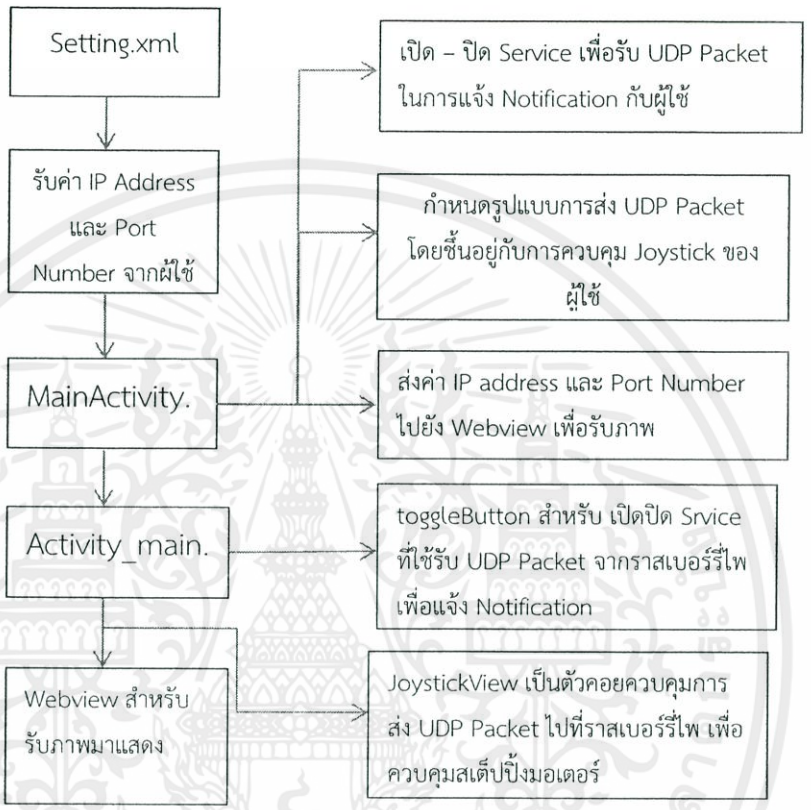
3.1 การออกแบบ

3.1.1 การออกแบบการทำงานการประมวลผลของโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์

3.1.1.1 การออกแบบการทำงานของชุดคำสั่ง

การทำงานของชุดคำสั่งของระบบปฏิบัติการแอนดรอยด์จะทำงานเมื่อมีการเปิดแอพพลิชั่นที่มีชุดคำสั่งการรับแพ็คเก็ตจากราสเบอร์รี่ไพ เพื่อแจ้งโนติฟิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ และชุดคำสั่งเพื่อส่งแพ็คเกตจากโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ไปที่ราสเบอร์รี่ไพ เพื่อควบคุมทิศทางของกล้องผ่านสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



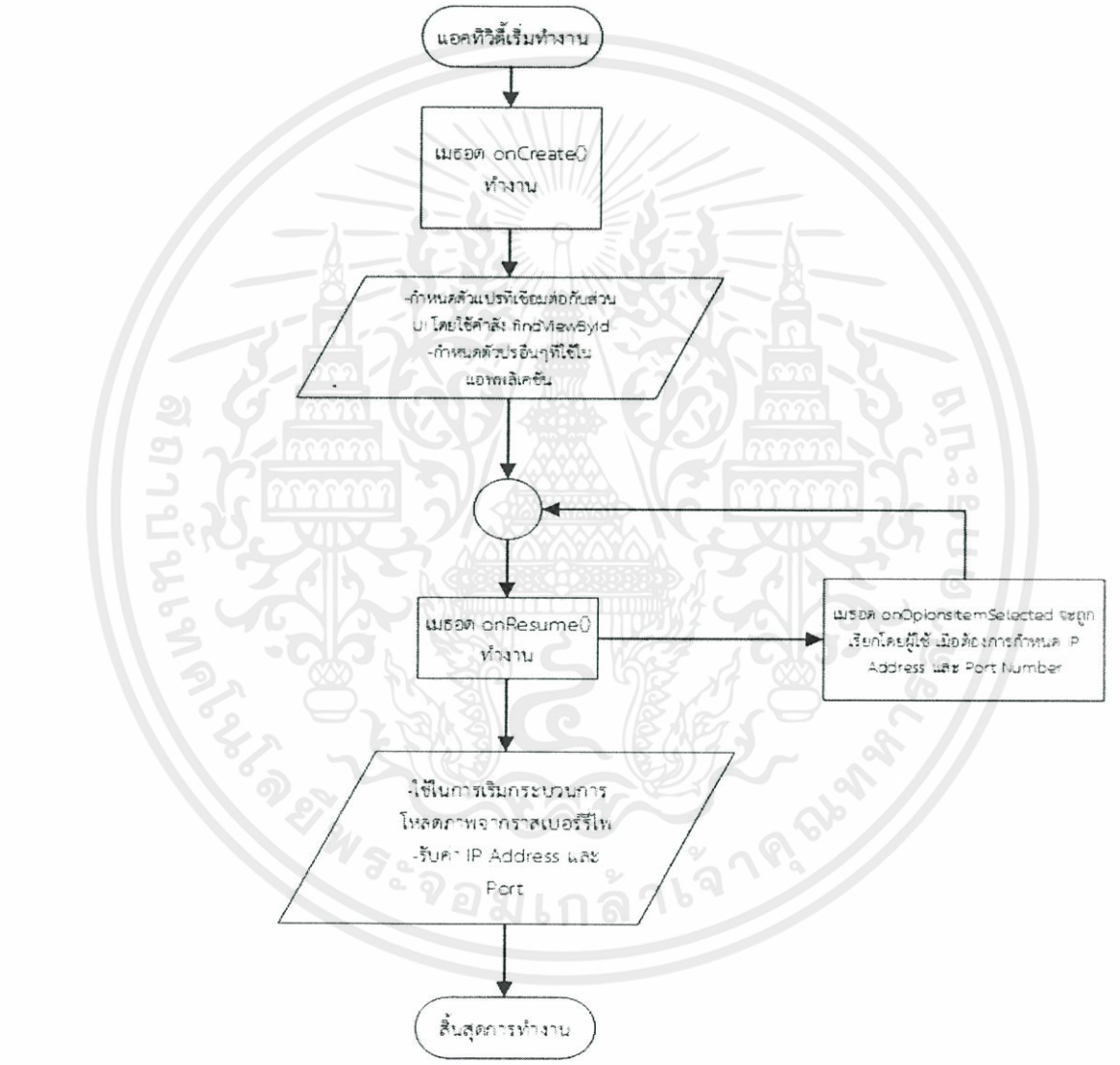
รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของชุดคำสั่ง
ระบบควบคุมกล้องไร้สายบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

จากรูปที่ 3.2 ได้ทำการออกแบบบล็อกไดอะแกรมการทำงานของชุดคำสั่งระบบควบคุมกล้องไร้สายผ่านโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ ซึ่งมีขั้นตอนการทำงานเมื่อเปิดใช้งานแอปพลิเคชั่น แอคทิวิตี (ACTIVITY)จะเริ่มทำงานในทันทีและเริ่มทำงานในเมธอด onCreate ซึ่งในเมธอดนี้ได้กำหนดตัวแปรของผลลัพธ์ที่เป็นส่วนแสดงผลออกทางหน้าจอ ตัวแปรของการส่งแพ็คเกตแบบ UDP และตัวแปรเกี่ยวกับการรับภาพจากราสเบอร์รี่ไพ หลังจากนั้นเมธอด onResume จะทำงานในการจัดการเกี่ยวกับการรับภาพจากราสเบอร์รี่ไพ และรับค่าจากราสเบอร์รี่ไพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 การทำงานชุดคำสั่งของระบบปฏิบัติการแอนดรอยด์

ในส่วนการประมวลผลชุดคำสั่งของระบบบล็อกไดอะแกรมการทำงานของชุดคำสั่งระบบควบคุมกล้องไร้สายผ่านโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ เป็นไปดังรูปดังรูปที่ 3.3



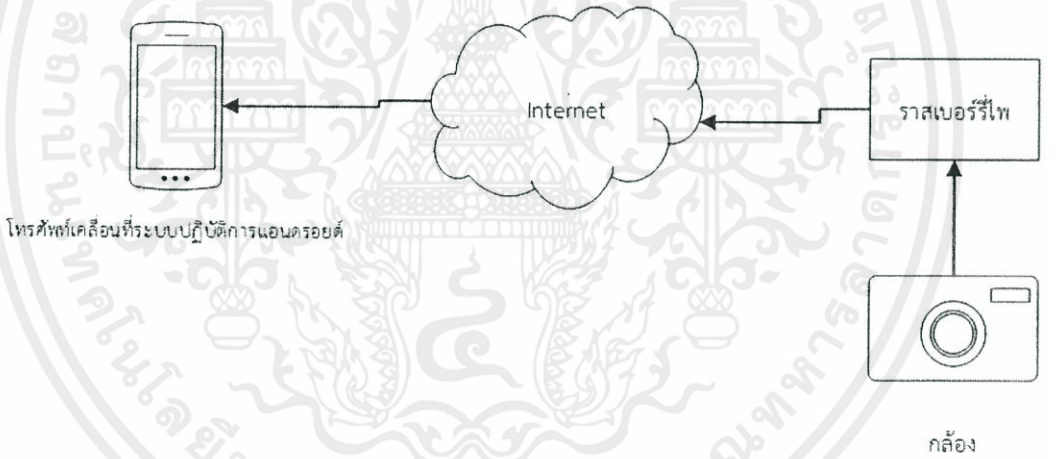
รูปที่ 3.3 การประมวลผลของชุดคำสั่งระบบควบคุมกล้องไร้สายผ่านโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 เป็นการออกแบบการประมวลผลชุดคำสั่งของระบบควบคุมกล้องไร้สายบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ซึ่งสามารถแบ่งออกเป็น 3 ส่วนคือ 1.ส่วนของการรับภาพจากราสเบอร์รี่ไพ โดยมีรับค่าไอพีแอดเดรสและพอร์ตจากผู้ใช้ 2. ส่วนของการส่ง UDP Packet ไปยังราสเบอร์รี่ไพ โดยผ่านตัวจอยสติ๊กเพื่อใช้ควบคุมสเต็ปปิ้งมอตอร์ 3.ส่วนของการเปิด-ปิด Serviceเพื่อรับการแจ้งเตือนจากราสเบอร์รี่ไพเมื่อมีเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้

3.1.1.3 ส่วนของการรับภาพจากราสเบอร์รี่ไพ

สำหรับการทำงานของแอพพลิเคชั่นในส่วนของการรับภาพจากราสเบอร์รี่ไพ นั้นจะทำการรับค่าไอพีแอดเดรสและพอร์ตจากผู้ใช้ ผ่านหน้าแอพพลิเคชั่น โดยจะทำการติดต่อกับราสเบอร์รี่ไพผ่านอินเตอร์เน็ต



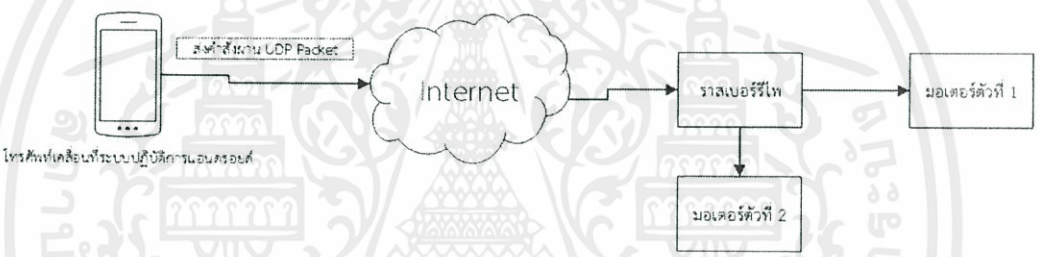
รูปที่ 3.4 ไดอะแกรมแสดงการเชื่อมต่อ
เพื่อรับภาพของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์จากราสเบอร์รี่ไพ

จากไดอะแกรมของระบบควบคุมกล้องไร้สายผ่านระบบปฏิบัติการแอนดรอยด์สมาร์ทโฟนนำมาเขียนชุดคำสั่งในแอพพลิเคชั่นในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เพื่อรับภาพจากราสเบอร์รี่ไพ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.4 ส่วนของการส่ง UDP Packet ไปที่ราสเบอร์รี่ไพ เพื่อควบคุมสเต็ปปิ้งมอเตอร์

สำหรับส่วนของการส่ง UDP Packet นั้นจะรับการควบคุมจากผู้ใช้ ผ่านตัวจอยสติ๊ก เพื่อควบคุมทิศทางการเคลื่อนที่ของกล้อง โดยจะมี 4 ทิศทางคือ หมุนขึ้น หมุนลง หมุนซ้าย และหมุนขวา โดยกำหนดว่า หากต้องการให้หมุนขึ้น จะส่งข้อความ “w” ต้องการให้หมุนลง จะส่งข้อความ “s” ต้องการให้หมุนซ้าย จะส่งข้อความ “a” และต้องการให้หมุนขวา จะส่งข้อความ “d” ผ่าน UDP Packet



รูปที่ 3.5 ไดอะแกรมของส่วนของการส่ง UDP Packet จากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ไปยังราสเบอร์รี่ไพเพื่อควบคุมมอเตอร์

จากไดอะแกรมของส่วนของการส่ง UDP Packet จากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ไปยังราสเบอร์รี่ไพเพื่อควบคุมมอเตอร์ นำมาเขียนชุดคำสั่งในแอพพลิเคชั่นในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เพื่อส่ง UDP Packet จากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ไปยังราสเบอร์รี่ไพเพื่อควบคุมมอเตอร์ ได้ดังนี้

```
private JoystickMovedListener _listener = new JoystickMovedListener() {
        @Override
        public void OnMoved(int pan, int tilt) {
            txtX.setText(Integer.toString(pan));
```

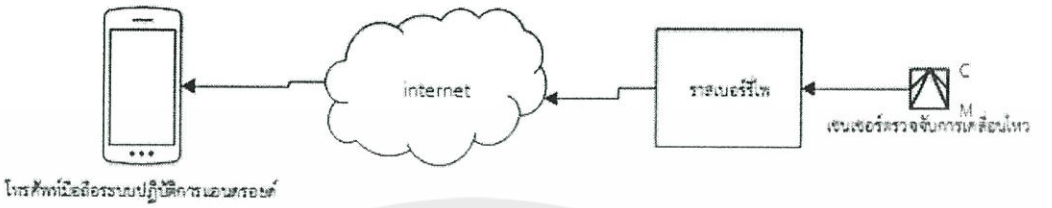
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
txtY.setText(Integer.toString(tilt));
        if (Math.abs(pan) > Math.abs(tilt)) {
            if (pan > 0) {
                RunDetect = new Detect(right);
            } else if (pan < 0) {
                RunDetect = new Detect(left);
            } else{
            }
        } else {
            if (tilt > 0) {
                RunDetect = new Detect(down);
            } else if (tilt < 0) {
                RunDetect = new Detect(up);
            } else {
            }
        }
    }
```

3.1.1.5 ส่วนของการรับการแจ้งเตือนจากราสเบอร์รี่ไพเมื่อมีเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้

สำหรับส่วนของการรับการแจ้งเตือนจากราสเบอร์รี่ไพ จะทำการเปิด Service เพื่อรับ UDP Packet จากราสเบอร์รี่ไพ เมื่อโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์รับ UDP Packet ได้จะทำการสร้างโนติฟิเคชั่นเพื่อแจ้งแก่ผู้ใช้ให้ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ไดอะแกรมของส่วนของการรับการแจ้งเตือนจากราสเบอร์รี่ไพเมื่อมีเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้

จากไดอะแกรมขอส่วนของการรับการแจ้งเตือนจากราสเบอร์รี่ไพเมื่อมีเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้ นำมาเขียนชุดคำสั่งในแอพพลิเคชั่นในโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เพื่อส่ง UDP Packet จากราสเบอร์รี่ไพ ไปยังโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เพื่อแจ้งเตือนผู้ใช้ผ่านโนติฟิเคชั่น ได้ดังนี้

```
public class MyService extends Service {
    String sentence;
    static Boolean isLooping = false;
    int port = 10597;

    byte[] receiveData = new byte[1024];
    byte[] sendData = new byte[1024];
    DatagramSocket serverSocket;

    public MyService() {
        try {
            serverSocket = new DatagramSocket(port);
        } catch (IOException e) {
            e.printStackTrace();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        }
Thread thread = new Thread() {
                @SuppressWarnings("deprecation")
                @Override
                public void run() {
                        try {
                                if (isLooping) {
                                        while (isLooping) {
                                                DatagramPacket receivePacket = new
DatagramPacket(receiveData, receiveData.length);
                                                try {
                                                        serverSocket.receive(receivePacket);
                                                } catch (IOException e) {
                                                        e.printStackTrace();
                                                }
                                                sentence = new
String(receivePacket.getData());
                                                if(receivePacket!=null){
                                                        createNotification();

                                                }
                                                InetAddress IPAddress =
receivePacket.getAddress();
                                                //
                                                int port = receivePacket.getPort();
                                                String capitalizedSentence = "Receive
Deteted";
                                                sendData = capitalizedSentence.getBytes();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    DatagramPacket sendPacket = new
                DatagramPacket(sendData, sendData.length,
                IPAddress, port);
                    try {
                        serverSocket.send(sendPacket);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    sleep(200);
                }
            } else {
                thread.stop();
                thread=null;
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
            thread.stop();
        }
    }
};
}
```

3.1.2 การออกแบบการทำงานการประมวลผลของราสเบอร์รี่ไพ

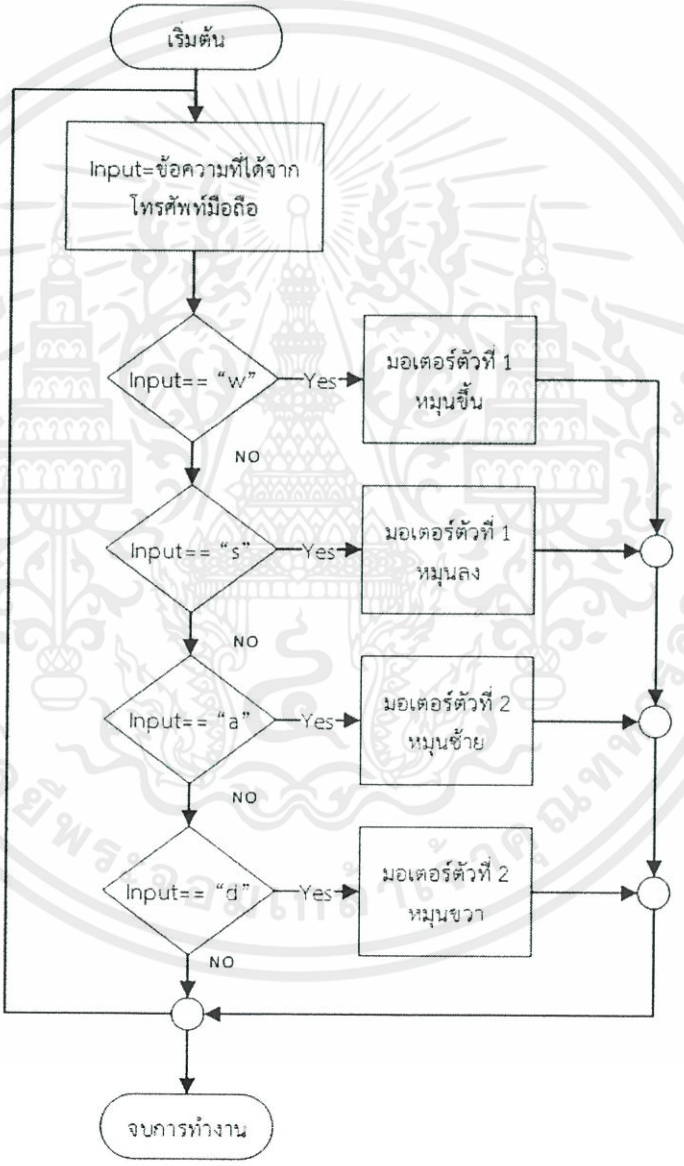
3.1.2.1 การออกแบบการทำงานของสเต็ปปิ้งมอเตอร์

การทำงานของชุดคำสั่งสเต็ปปิ้งมอเตอร์จะทำการสร้างเซิฟเวอร์ เพื่อรอรับข้อความ โดยผ่านการส่งข้อมูลแบบ UDP จากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ เมื่อรับข้อความได้ จะทำการเข้าเงื่อนไข โดยที่หากข้อความที่ได้เป็นตัวอักษร “w” และ “s” จะทำการสั่งให้มอเตอร์ตัวที่หนึ่งหมุน โดยสั่งให้มอเตอร์หมุนกล้องขึ้น และลงตามลำดับ และถ้าตัวอักษรเป็น “a” และ “d” จะทำการสั่งให้มอเตอร์ตัวที่สองหมุน โดยสั่งให้มอเตอร์หมุนกล้องซ้ายและขวา ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การตั้งค่ากล้อง

การทำงานของชุดคำสั่งของระบบปฏิบัติการแอนดรอยด์จะทำงานเมื่อมีการเปิดแอพพลิชั่นที่มีชุดคำสั่งการรับแพ็คเก็ตจากราสเบอรี่ไพ เพื่อแจ้งโนติฟิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ และชุดคำสั่งเพื่อส่งแพ็คเกตจากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ไปที่ราสเบอรี่ไพ เพื่อควบคุมทิศทางของกล้องผ่านสเต็ปปิ้งมอเตอร์ ตามโฟล์วชาร์ตรูปที่ 3.7



รูปที่ 3.7 โฟล์วชาร์ตของชุดคำสั่งเพื่อควบคุมทิศทางของกล้องผ่านสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโฟล์วชาร์ตของชุดคำสั่งเพื่อควบคุมทิศทางของกล้องผ่านสเต็ปปิ้งมอเตอร์ สามารถนำมาเขียนชุดคำสั่งภาษาไพธอนในราสเบอร์รี่ไพ ได้ดังนี้

```
# Raspberry pi control motor
from socket import *
from time import ctime
import RPi.GPIO as GPIO
import time
import sys

pin1=23
pin2=24
pin3=25
pin4=8
pin5=17
pin6=18
pin7=9
pin8=11

# set pin directions
GPIO.setmode(GPIO.BCM)
GPIO.setup(pin1,GPIO.OUT)
GPIO.setup(pin2,GPIO.OUT)
GPIO.setup(pin3,GPIO.OUT)
GPIO.setup(pin4,GPIO.OUT)
GPIO.setup(pin5,GPIO.OUT)
GPIO.setup(pin6,GPIO.OUT)
GPIO.setup(pin7,GPIO.OUT)
GPIO.setup(pin8,GPIO.OUT)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#
Apin1=[1,0,0,1]
Apin2=[1,1,0,0]
Apin3=[0,1,1,0]
Apin4=[0,0,1,1]
current=300
target=0
HOST= "ไอพีเครื่อง"
PORT = พอร์ต
BUFSIZ = 1024
ADDR = HOST,PORT
def GO_THERE(target,current):
    if current<target:
        while current<target:
            i=current%4
            GPIO.output(pin1,Apin1[i])
            GPIO.output(pin2,Apin2[i])
            GPIO.output(pin3,Apin3[i])
            GPIO.output(pin4,Apin4[i])
            time.sleep(.003)
            current= current + 1
    else:
        while current>target:
            i=current%4
            GPIO.output(pin1,Apin1[i])
            GPIO.output(pin2,Apin2[i])
            GPIO.output(pin3,Apin3[i])
            GPIO.output(pin4,Apin4[i])
            time.sleep(.003)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                current= current - 1
        print current,target
        return current;
    def GO_THERE1(target,current):
        if current<target:
            while current<target:
                i=current%4
                GPIO.output(pin5,Apin1[i])
                GPIO.output(pin6,Apin2[i])
                GPIO.output(pin7,Apin3[i])
                GPIO.output(pin8,Apin4[i])
                time.sleep(.003)
                current= current + 1
        else:
            while current>target:
                i=current%4
                GPIO.output(pin5,Apin1[i])
                GPIO.output(pin6,Apin2[i])
                GPIO.output(pin7,Apin3[i])
                GPIO.output(pin8,Apin4[i])
                time.sleep(.003)
                current= current - 1
        print current,target
        return current;


udpSerSock=socket(AF_INET, SOCK_DGRAM)
udpSerSock.bind(ADDR)
```

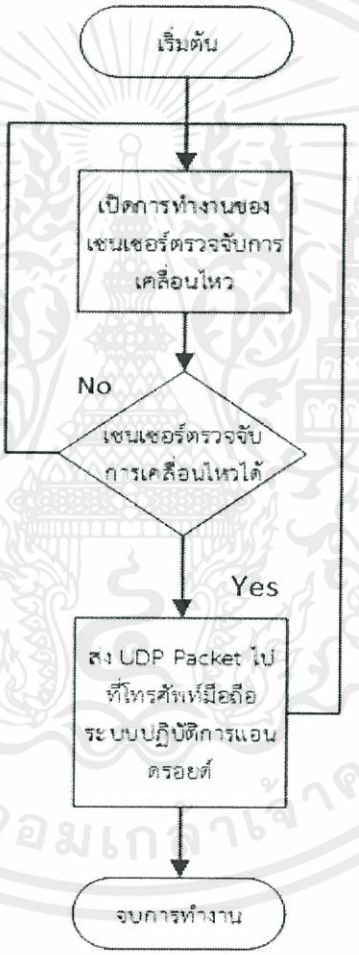
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while True:
        current=300
        print 'waiting for message...'
        data, addr = udpSerSock.recvfrom(BUFSIZ)
        print data
        udpSerSock.sendto('[%s] %s' %(
        ctime(), data), addr)
        print'...received from and returned to:', addr
        if data=="a":
                target=0
                current=GO_THERE(target,current)
                time.sleep(.003)
        elif data=="d":
                target=600
                current=GO_THERE(target,current)
                time.sleep(.003)
        elif data=="w":
                target=0
                current=GO_THERE1(target,current)
                time.sleep(.003)
        elif data=="s":
                target=600
                current=GO_THERE1(target,current)
                time.sleep(.003)
        else:
                time.sleep(2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 การออกแบบการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว

การทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว โดยการเขียนชุดคำสั่งเพื่อตรวจจับการเคลื่อนไหว หากเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหว ก็จะทำการส่ง UDP Packet ไปที่โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ตามโฟล์วชาร์ตรูปที่ 3.8



รูปที่ 3.8 โฟล์วชาร์ตของชุดคำสั่งเพื่อตรวจจับการเคลื่อนไหวและส่ง UDP Packet ไปที่โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโฟล์วชาร์ตของชุดคำสั่งเพื่อตรวจจับการเคลื่อนไหวและส่ง UDP Packet ไปที่โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ สามารถนำมาเขียนชุดคำสั่งภาษาไพธอนในราสเบอร์รี่ไพ ได้ดังนี้

```
#Raspberry pi Sensor
import RPi.GPIO as GPIO
import time
import socket

GPIO_PIR = 7

GPIO.setup(GPIO_PIR,GPIO.IN)

Current_State  = 0
Previous_State = 0
UDP_IP = "หมายเลขไอพีเครื่องปลายทาง"
UDP_PORT = "หมายเลขพอร์ต"

try:
        print "Waiting for PIR to settle ..."

        while GPIO.input(GPIO_PIR)==1:
                Current_State  = 0
                print "  Ready"
        while True :
                Current_State = GPIO.input(GPIO_PIR)
                if Current_State==1 and Previous_State==0:
                        print "  Motion detected!"
                        data = "m"
```

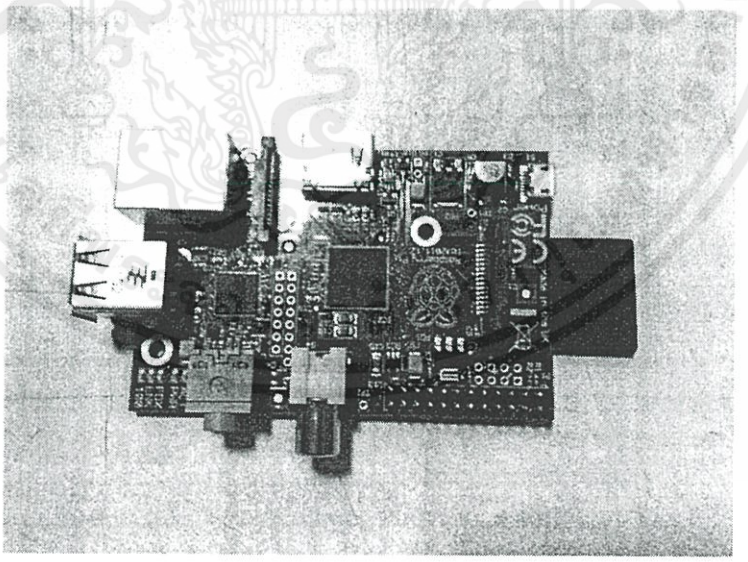
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sock = socket.socket(socket.AF_NET,
                                         socket.SOCK_DGRAM)
                    sock.sendto(data, (UDP_IP, UDP_PORT))
                    Previous_State=1
                elif Current_State==0 and Previous_State==1:
                    print "  Ready"
                    Previous_State=0
                    time.sleep(0.01)

except KeyboardInterrupt:
    print "  Quit"
    GPIO.cleanup()
```

3.2 อุปกรณ์ที่ใช้ในการทดลอง

3.2.1 ราสเบอร์รี่ไพ



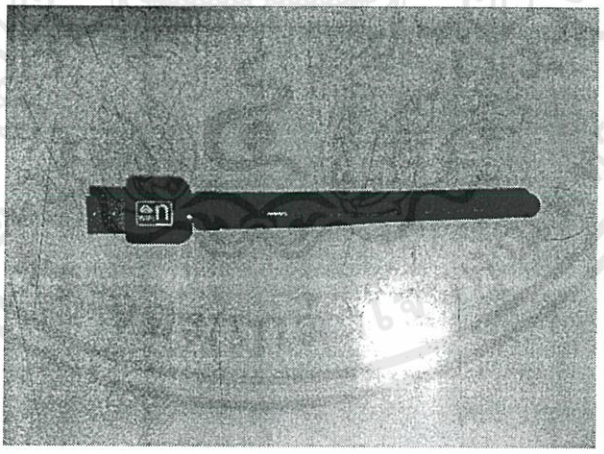
รูปที่ 3.9 ราสเบอร์รี่ไพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

รูปที่ 3.10 โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์

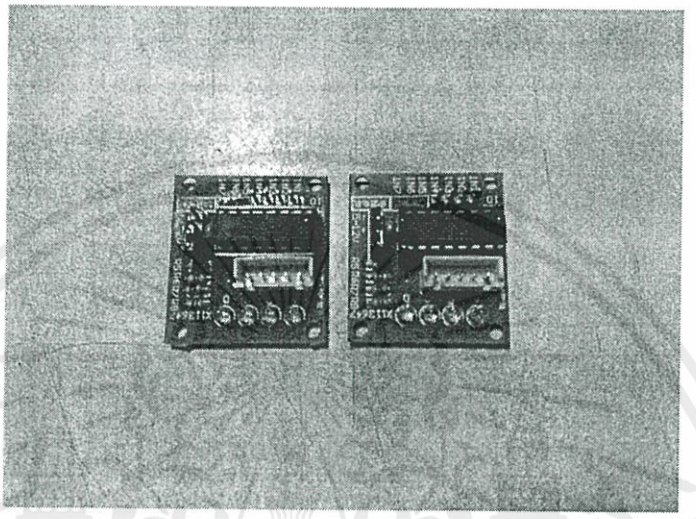
3.2.3 USB – Mini – Wifi



รูปที่ 3.11 USB-Mini-Wifi

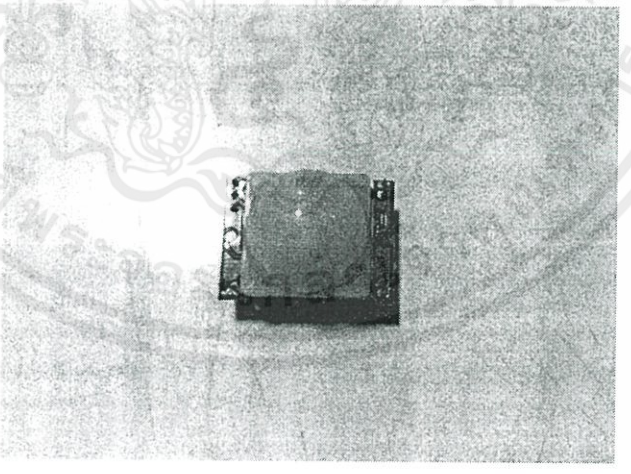
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 ชิป ULN2003an



รูปที่ 3.12 ชิป ULN2003an

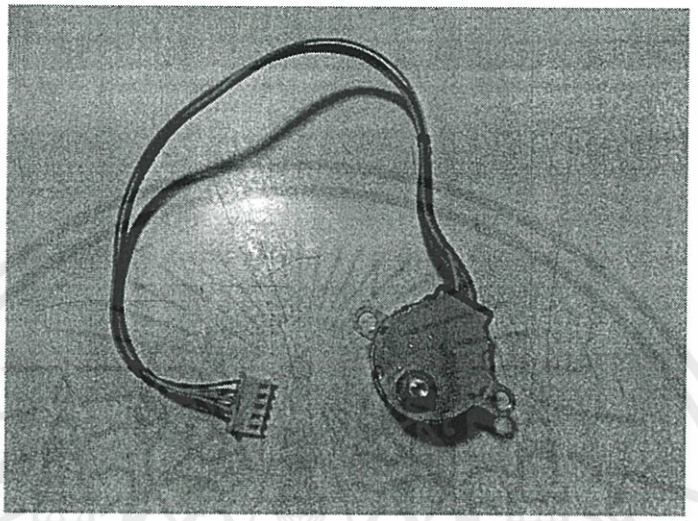
3.2.5 เซ็นเซอร์ตรวจจับการเคลื่อนไหว



รูปที่ 3.13 เซ็นเซอร์ตรวจจับการเคลื่อนไหว

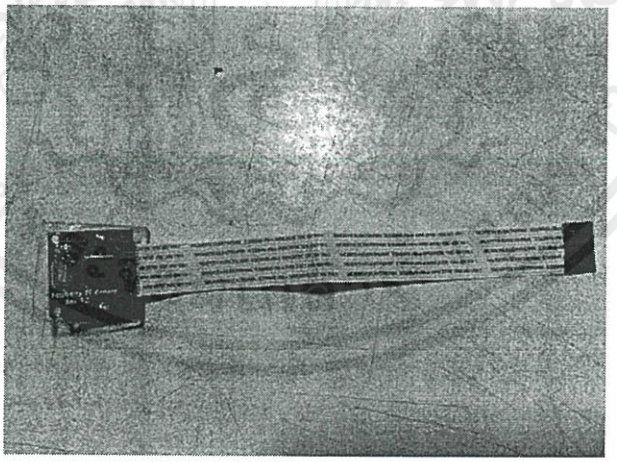
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 สเต็ปปิ้งมอเตอร์



รูปที่ 3.14 สเต็ปปิ้งมอเตอร์

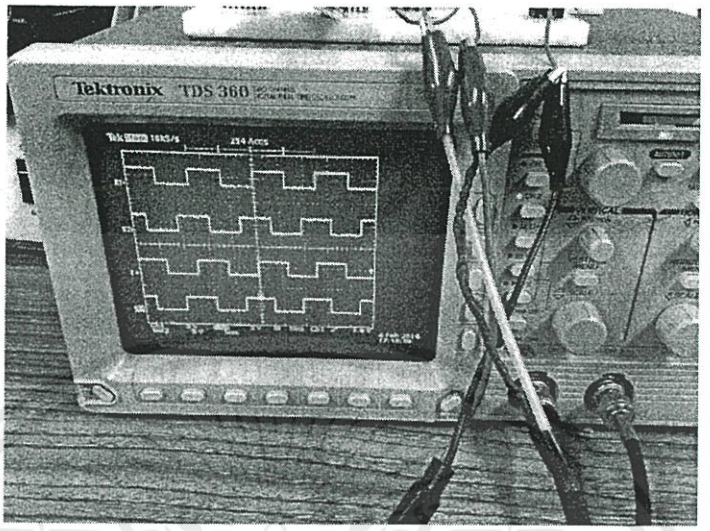
3.2.7 กล้อง



รูปที่ 3.15 กล้อง (Module raspberry pi camera)

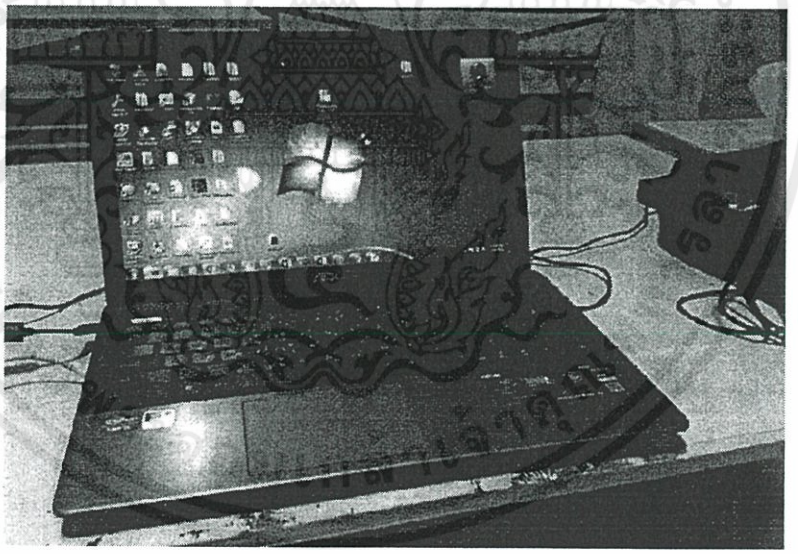
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 ออสซิลโลสโคป



รูปที่ 3.16 ออสซิลโลสโคป

3.2.8 คอมพิวเตอร์



รูปที่ 3.17 คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองการใช้ราสเบอร์รี่ไพในการควบคุมการทำงานของอุปกรณ์ต่างๆ

4.1.1 ผลการทดลองโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์

ก่อนที่จะเริ่มการทำงานของโปรแกรมบนราสเบอร์รี่ไพ จะต้องเชื่อมต่ออุปกรณ์ที่ใช้ในการทดลองซึ่งประกอบไปด้วยคอมพิวเตอร์ เร้าเตอร์ ราสเบอร์รี่ไพ และชุดบอร์ดวงจรขับสเต็ปปิ้งมอเตอร์ หลังจากที่ทำการเชื่อมต่ออุปกรณ์เสร็จเรียบร้อยแล้ว ให้พิมคำสั่งดังรูปที่ 4.1

รูปที่ 4.1 คำสั่งที่ใช้ในการสร้างไฟล์และสั่งรันโปรแกรม

จากรูปที่ 4.1 สามารถอธิบายได้ดังนี้

4.1.1.1 พิมคำสั่ง sudo nano test.py เพื่อสร้างไฟล์เพื่อเข้าไปเขียนชุดคำสั่งหรือเข้าไปแก้ไขชุดคำสั่ง

4.1.1.2 พิมคำสั่ง sudo python test.py เพื่อเริ่มทำงานของโปรแกรม

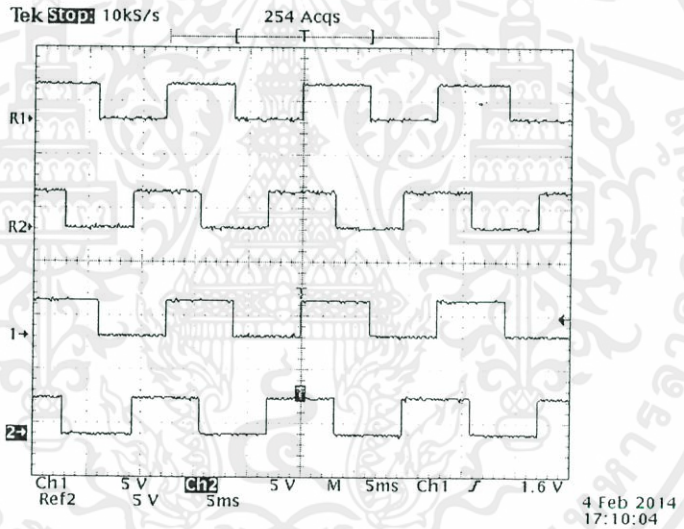
จากการทดลองพบว่า เมื่อเริ่มทำการทดลองโปรแกรมในชุดคำสั่งที่ได้ออกแบบไว้เริ่มต้นการทำงานของโปรแกรม โดยรับค่าตัวอักษรภาษาอังกฤษจาก แป้นพิมพ์ คือ “a” , “d”, “w” หรือ “s” กล่าวคือรับตัวอักษร “a” สเต็ปปิ้งมอเตอร์ตัวที่ 1 จะหมุนทวนเข็มนาฬิกา เมื่อรับตัวอักษร “d” สเต็ปปิ้งมอเตอร์ตัวที่ 1 จะหมุนตามเข็มนาฬิกา เมื่อรับตัวอักษร “w” สเต็ปปิ้งมอเตอร์ตัวที่ 2 จะหมุนทวนเข็มนาฬิกา และ เมื่อรับตัวอักษร “s” สเต็ปปิ้งมอเตอร์ตัวที่ 2 จะหมุนตามเข็มนาฬิกา ดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 บอกลักษณะการหมุนของสเต็ปปิ้งมอเตอร์เมื่อทำการป้อนตัวอักษรในโปรแกรม

ป้อนตัวอักษร	ลักษณะการหมุนของมอเตอร์
A	มอเตอร์ตัวที่ 1 หมุนทวนเข็มนาฬิกา
D	มอเตอร์ตัวที่ 1 หมุนตามเข็มนาฬิกา
W	มอเตอร์ตัวที่ 2 หมุนทวนเข็มนาฬิกา
S	มอเตอร์ตัวที่ 2 หมุนตามเข็มนาฬิกา

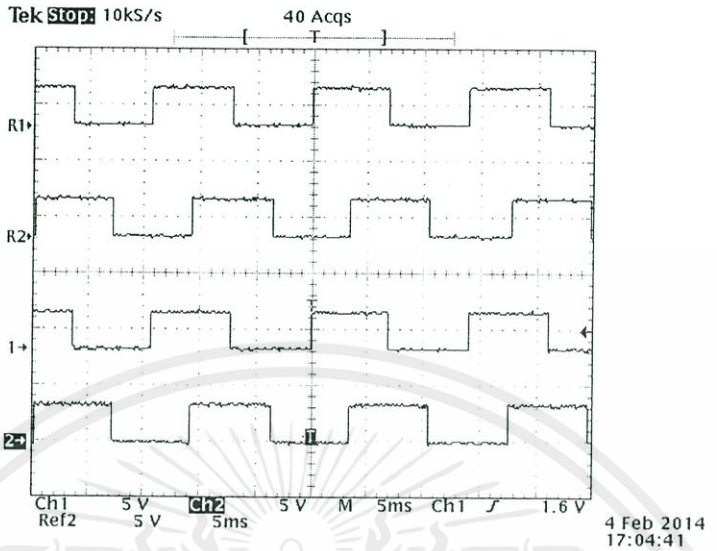
จากนั้นดูรูปสัญญาณที่ออกมาจาก จีพีไอโอ(GPIO)ของราสเบอร์รี่ไพ เพื่อไปควบคุมบอร์ดวงจรเพื่อขับสเต็ปปิ้งมอเตอร์ โดยการต่อออสซิลโลสโคป ทำการวัดสัญญาณขณะมอเตอร์หมุนทวนเข็มนาฬิกาได้ผลดังรูปที่ 4.2 และเมื่อทำการวัดสัญญาณขณะมอเตอร์หมุนตามเข็มนาฬิกาได้ผลดังรูปที่ 4.3



รูปที่ 4.2 สัญญาณที่วัดได้จากจีพีไอโอของราสเบอร์รี่ไพเมื่อสั่งมอเตอร์หมุนทวนเข็มนาฬิกา

จากรูปที่ 4.2 เป็นสัญญาณเกิดจากกระตุ้นให้กับเฟสของสเต็ปปิ้งมอเตอร์ที่ละ 2 เฟสที่อยู่ใกล้กันในเวลาเดียวกันและเรียงถัดกันโดยที่ R1: คือสัญญาณที่วัดจากเฟสที่ 1 ของสเต็ปปิ้งมอเตอร์ , R2 : คือสัญญาณที่วัดจากเฟสที่ 2 ของสเต็ปปิ้งมอเตอร์ , 1 : คือสัญญาณที่วัดจากเฟส 3 ของสเต็ปปิ้งมอเตอร์ , 2 : คือสัญญาณที่วัดจากเฟส 4 ของสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



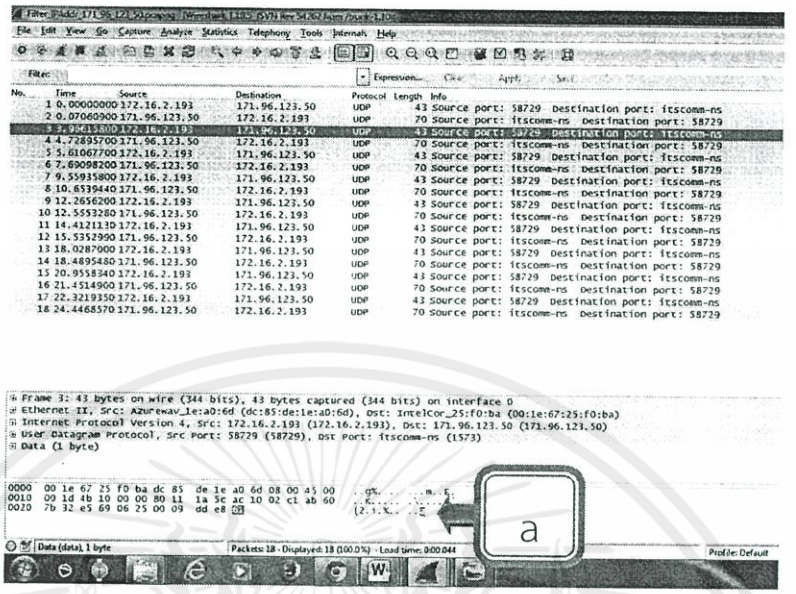
รูปที่ 4.3 สัญญาณที่วัดได้จากจีพีไอโอของราสเบอร์รี่ไพเมื่อสั่งมอเตอร์หมุนตามเข็มนาฬิกา

จากรูปที่ 4.3 เป็นสัญญาณเกิดจากกระตุ้นให้กับเฟสของสเต็ปปิ้งมอเตอร์ที่ละ 2 เฟส ที่อยู่ใกล้กันในเวลาเดียวกันและเรียงถัดกันแต่เป็นการป้อนย้อนกลับเพื่อให้มอเตอร์หมุนตามเข็มนาฬิกา โดยที่ R1: คือสัญญาณที่วัดจากเฟสที่ 1 ของสเต็ปปิ้งมอเตอร์ , R2 : คือสัญญาณที่วัดจากเฟสที่ 2 ของสเต็ปปิ้งมอเตอร์ , 1 : คือสัญญาณที่วัดจากเฟส 3 ของสเต็ปปิ้งมอเตอร์ , 2 : คือสัญญาณที่วัดจากเฟส 4 ของสเต็ปปิ้งมอเตอร์

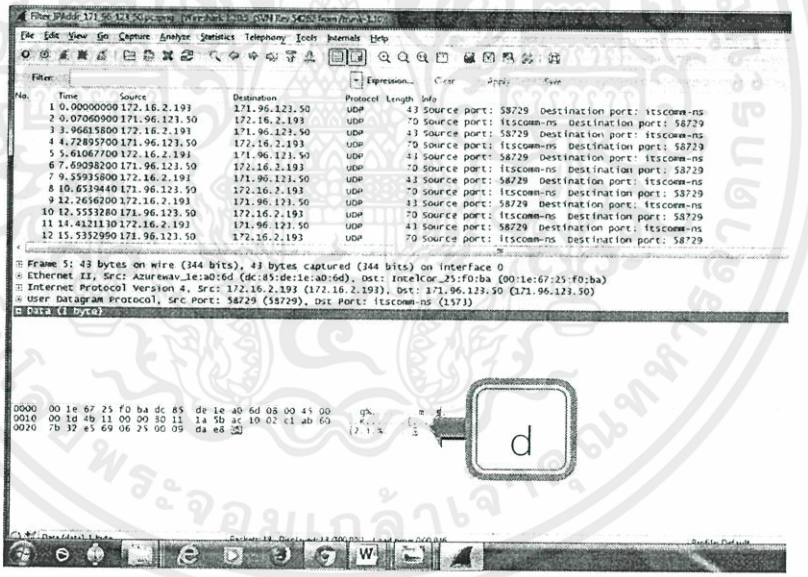
4.1.2 ผลการทดลองการส่งข้อมูลผ่านโปรโตคอล UDP ไปยังราสเบอร์รี่ไพ

เริ่มจากทำการเขียนชุดคำสั่งไพธอนที่คอมพิวเตอร์ที่เชื่อมต่อไวไฟวงแลนเดียวกันกับราสเบอร์รี่ไพจากนั้นทำการรันโปรแกรมเพื่อส่งค่า “a”, “d”, “w”และ “s” ไปยังราสเบอร์รี่ไพ ทำการดักจับข้อมูลที่ส่งไปยังราสเบอร์รี่ไพด้วยโปรแกรม Wireshark บนคอมพิวเตอร์ โดยการทดลองนี้ฝั่งไคลแอนท์จะทำการต่อกับเครือข่ายเน็ตเวิร์คได้ไอพี 172.16.2.193 ซึ่งทำการดักจับแพ็คเกจที่ฝั่งนี้ ส่งไปยังราสเบอร์รี่ซึ่งต่อกับเร้าเตอร์ตั้งค่าไอพีเป็น 192.168.0.222 ต่อกับเร้าเตอร์ซึ่งมีไอพีขา WAN เป็นไอพี 171.96.123.50 จะต้องทำการ Forward Port ที่เร้าเตอร์โดยกำหนดให้ เมื่อแพ็คเกจมาทางพอร์ต 1573 ให้ส่งแพ็คเกจไปที่ราสเบอร์รี่ไพซึ่งมีไอพี 192.168.0.222 พอร์ต 1573 ผลการทดลองการดักจับข้อมูลของ “a” ที่ได้เป็นดังรูปที่ 4.4 ผลการทดลองการดักจับข้อมูลของ “d” ที่ได้เป็นดังรูปที่ 4.5 ผลการทดลองการดักจับข้อมูลของ “w” ที่ได้เป็นดังรูปที่ 4.6 ผลการทดลองการดักจับข้อมูลของ “s” ที่ได้เป็นดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

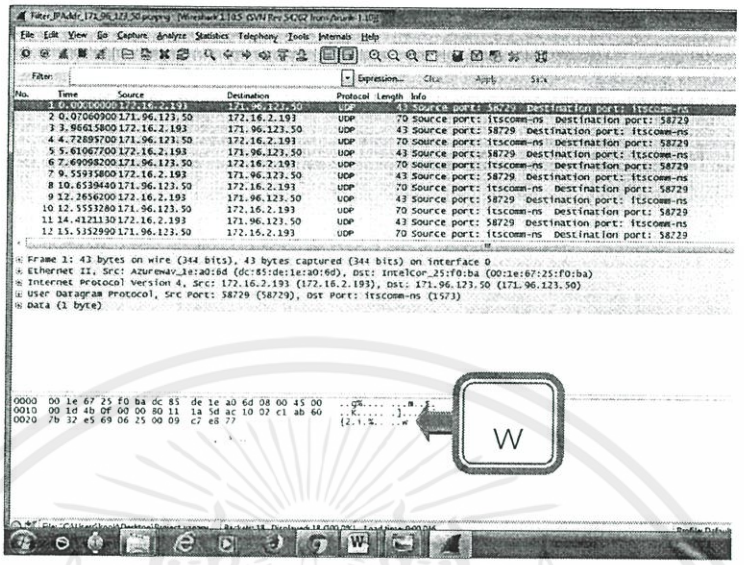


รูปที่ 4.4 ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “a” ไปยังราสเบอร์รี่ไพ

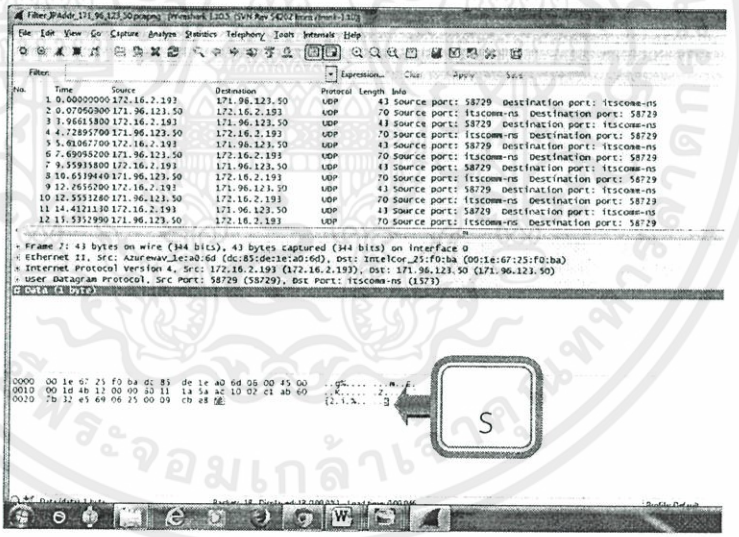


รูปที่ 4.5 ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “d” ไปยังราสเบอร์รี่ไพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

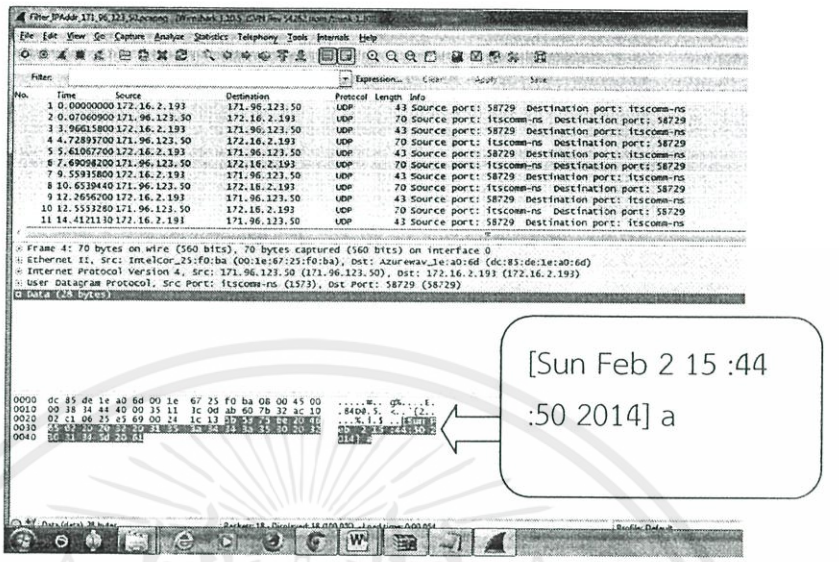


รูปที่ 4.6 ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “w” ไปยังราสเบอร์รี่ไพ

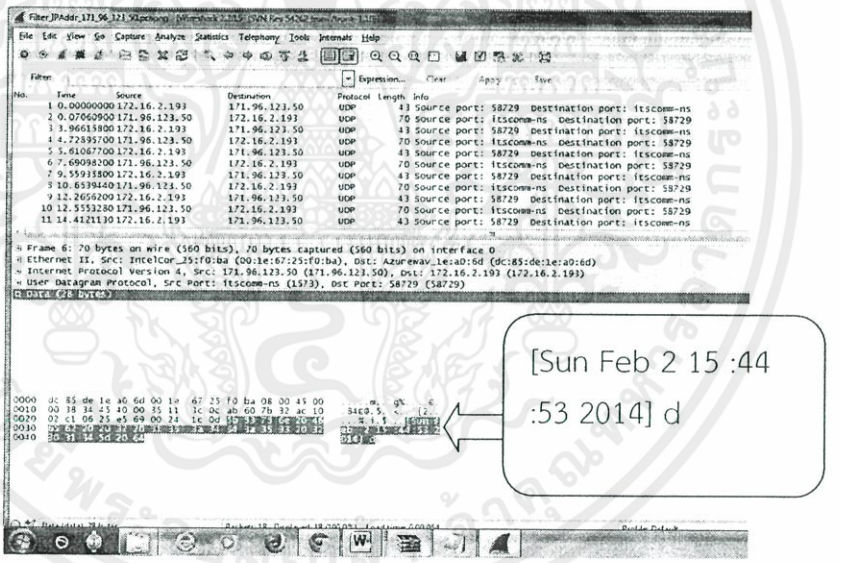


รูปที่ 4.7 ผลการดักจับข้อมูลที่คอมพิวเตอร์ ส่งข้อมูล “s” ไปยังราสเบอร์รี่ไพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

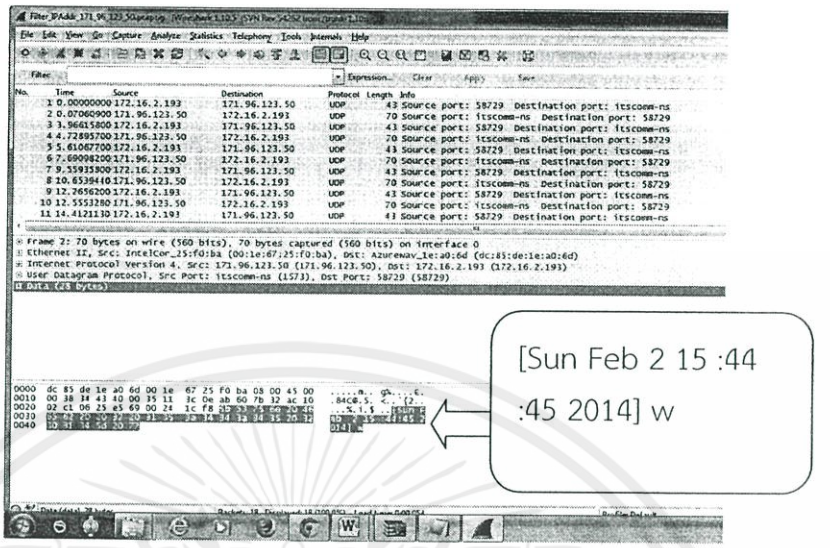


รูปที่ 4.8 แพ็กเกตที่ตอบกลับมาเมื่อส่ง “a”

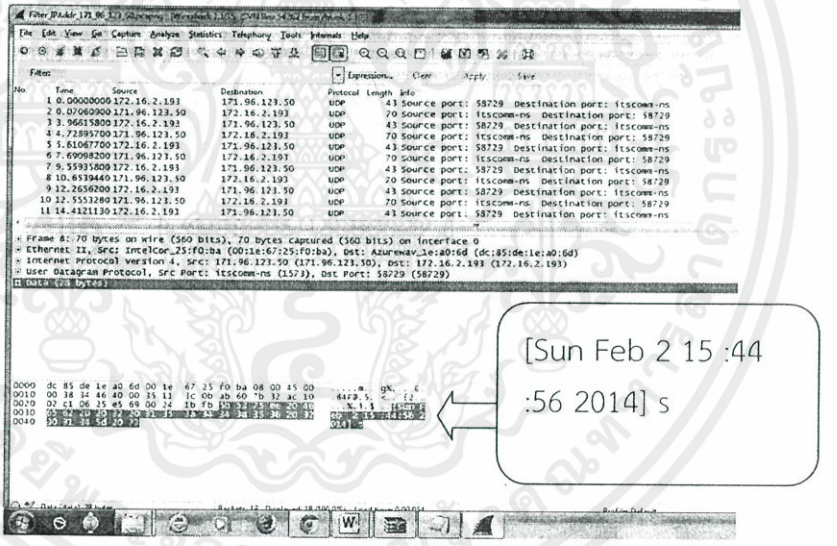


รูปที่ 4.9 แพ็กเกตที่ตอบกลับเมื่อส่ง “d”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แพ็กเกตที่ตอบกลับเมื่อส่ง “w”



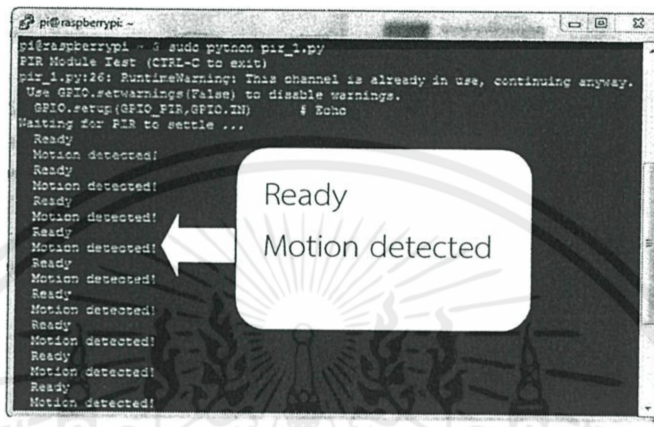
รูปที่ 4.11 แพ็กเกตที่ตอบกลับเมื่อส่ง “s”

เมื่อคอมพิวเตอร์ส่งค่า “a” , “d” , “w” และ “s” จะมีแพ็กเกตตอบกลับตามที่ได้โปรแกรมไว้เป็นข้อมูลเวลาแล้วตามด้วยแพ็กเกตนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

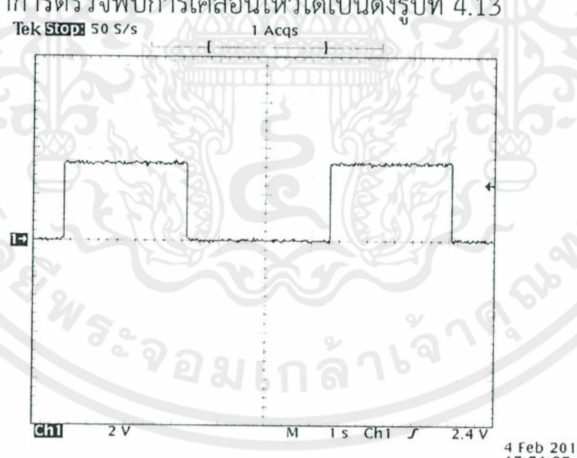
4.1.3 ผลการทดลองโปรแกรมรับค่าการตรวจจับการเคลื่อนไหว (PIR Sensor)

จากการทดลองพบว่าเมื่อเริ่มทำงานของโปรแกรมที่ได้ออกแบบชุดคำสั่งไว้ จะขึ้นผลทางหน้าเทอร์มินอลของราสเบอร์รี่ไพว่า “Ready” หมายถึงพร้อมตรวจจับ เมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้ จะแสดงผลทางหน้าเทอร์มินอลของราสเบอร์รี่ไพว่า “Motion detected” ดังรูปที่ 4.12



รูปที่ 4.12 ผลของหน้าจอเทอร์มินอลของราสเบอร์รี่ไพเมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้

จากนั้นวัดสัญญาณที่ได้รับจากเซ็นเซอร์ตรวจจับการเคลื่อนไหวเข้าราสเบอร์รี่ไพในขณะที่เซ็นเซอร์ทำการตรวจพบการเคลื่อนไหวได้เป็นดังรูปที่ 4.13



รูปที่ 4.13 ผลของสัญญาณที่ได้จากเซ็นเซอร์เมื่อตรวจพบการเคลื่อนไหว

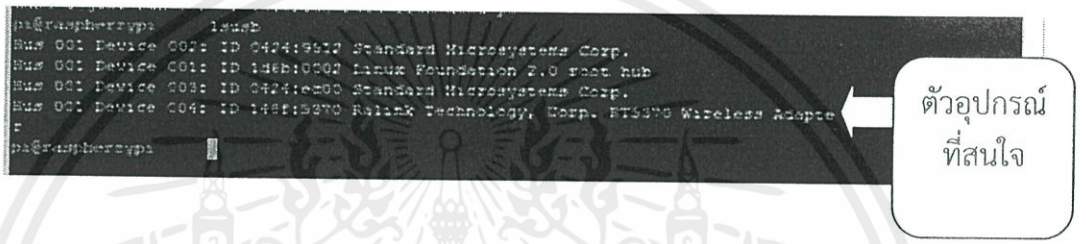
จากรูปที่ 4.13 เป็นสัญญาณที่วัดจากขาเอาต์พุทของเซ็นเซอร์โดยเริ่มแรกระดับสัญญาณที่วัดได้ขณะเซ็นเซอร์ไม่พบการเคลื่อนไหวจะมีค่าแรงดันเป็น 0 โวลต์ แต่เมื่อเซ็นเซอร์ตรวจพบการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนไหวจะมีค่าแรงดันเปลี่ยนแปลงโดยเพิ่มจาก 0 โวลต์ เป็น 3.2 โวลต์ และเมื่อเซ็นเซอร์ตรวจไม่พบการเคลื่อนไหวก็จะมีแรงดันเปลี่ยนแปลงกลับมาเป็น 0 โวลต์ อีกครั้ง และกลับมาเป็น 3.2 โวลต์ อีกครั้งเมื่อเซ็นเซอร์ตรวจพบการเคลื่อนไหวในครั้งถัดไป

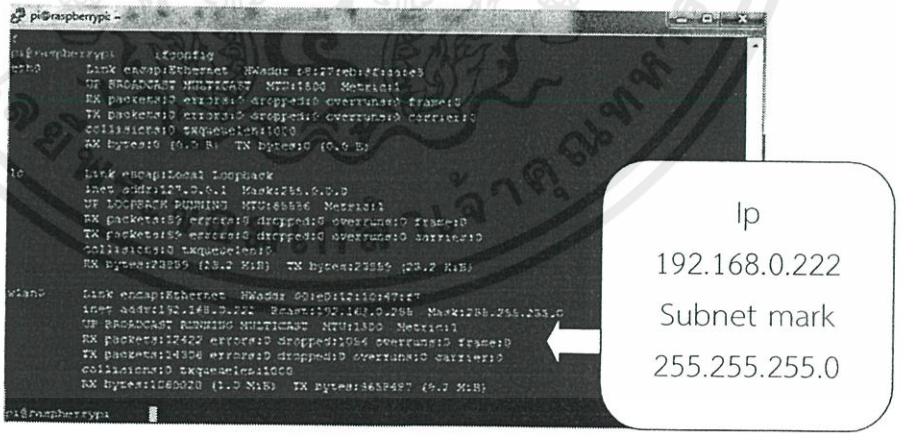
4.1.4 ผลการทดลองติดตั้งยูเอสบีไวไฟ (MINI USB WIFI)

เริ่มต้นจากต่ออุปกรณ์ USB WIFI เข้ากับราสเบอรี่ไพผ่านทางพอร์ตยูเอสบี จากนั้นพิมคำสั่ง “lsusb” ดังรูปที่ 4.14 เป็นคำสั่งที่ใช้ในการดูว่ามีอุปกรณ์อะไรต่ออยู่กับอินเตอร์เฟสใดในราสเบอร์รี่ไพ



รูปที่ 4.14 ผลหน้าเทอร์มินอลเมื่อพิมคำสั่ง “lsusb” ในราสเบอร์รี่ไพ

จากรูปที่ 4.14 คำสั่ง “lsusb” เป็นคำสั่งที่ตรวจดูว่ามีอุปกรณ์ใดต่ออยู่กับราสเบอร์รี่ไพบ้าง โดยเราจะพบว่าในส่วนของ Bus 001 Device 004 เป็นบรรทัดที่แสดงให้เห็นว่ามีอุปกรณ์ Wireless Adapter เชื่อมต่ออยู่ พิมคำสั่ง “sudo nano /etc/network/interfaces เพื่อเข้าไปกำหนดค่าต่างๆ เมื่อเสร็จทำการรีบูทราสเบอร์รี่ไพ แล้วถอดสายแลนออกแล้วรอสักครู่เมื่อไปยูเอสบีไวไฟติดให้ทำการพิมสั่ง “ifconfig” ผลที่ได้เป็นดังรูปที่ 4.15



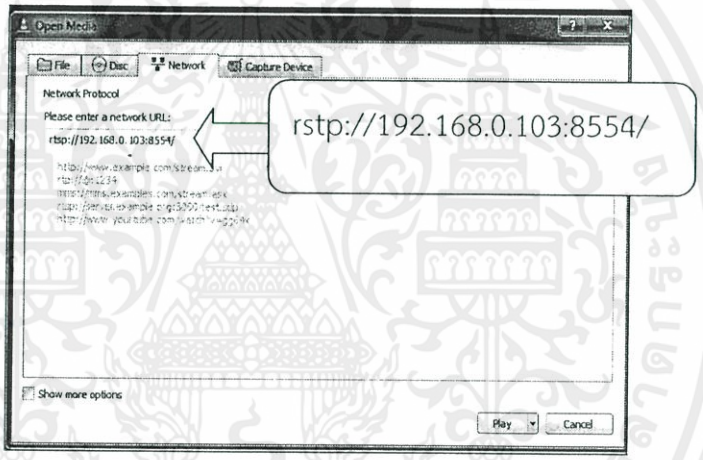
รูปที่ 4.15 ผลของหน้าเทอร์มินอลเมื่อพิมคำสั่ง “ifconfig”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.15 คำสั่ง “ifconfig ” เป็นคำสั่งที่ใช้แสดงข้อมูลเกี่ยวกับเน็ตเวิร์กอินเตอร์เฟส (NETWORK INTERFACE) จะเห็นว่าในส่วนดับบลิวแลนศูนย์ ได้รับไอพีจากเราเตอร์แบบไร้สาย คือ 192.168.0.222 มีบรอร์ดแคสเป็น 192.168.0.255 และมีซับเน็ตมาร์คเป็น 255.255.255.0

4.1.5 ผลการทดลองรับภาพจากกล้องของราสเบอร์รี่ไพ

ทำการติดตั้งโปรแกรมวีแอลซี (VLC) ที่ราสเบอร์รี่ไพเพื่อให้ราสเบอร์รี่เป็นเซริฟเวอร์เพื่อส่งภาพไปยังเครื่องไคลแอนด์ที่ราสเบอร์รี่ไพทำการพิมคำสั่ง “raspivid -o - -t 0 -n -w 600 -h 400 -fps 12 | cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264” เป็นคำสั่งที่ใช้เปิดราสเบอร์รี่ไพเป็นเซริฟเวอร์ และที่ฝั่งไคลแอนด์เปิดโปรแกรมวีแอลซีขึ้นมาพิม “rtsp://192.168.0.103:8554/” ดังรูปที่ 4.16 เพื่อทดลองรับวิดีโอจากราสเบอร์รี่ไพผลที่ได้เป็นดังรูปที่ 4.17

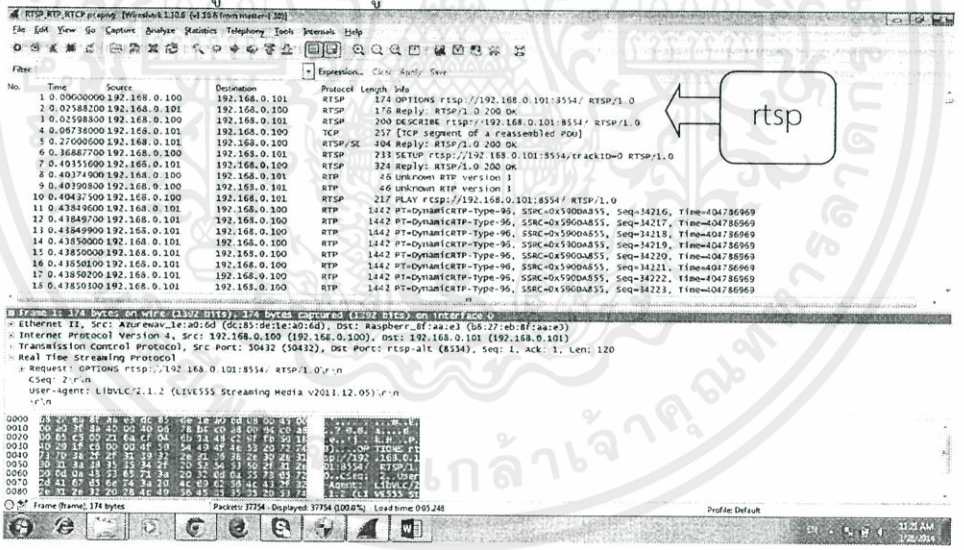


รูปที่ 4.16 หน้าต่างการป้อนค่าการรับภาพจากเน็ตเวิร์ก (Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

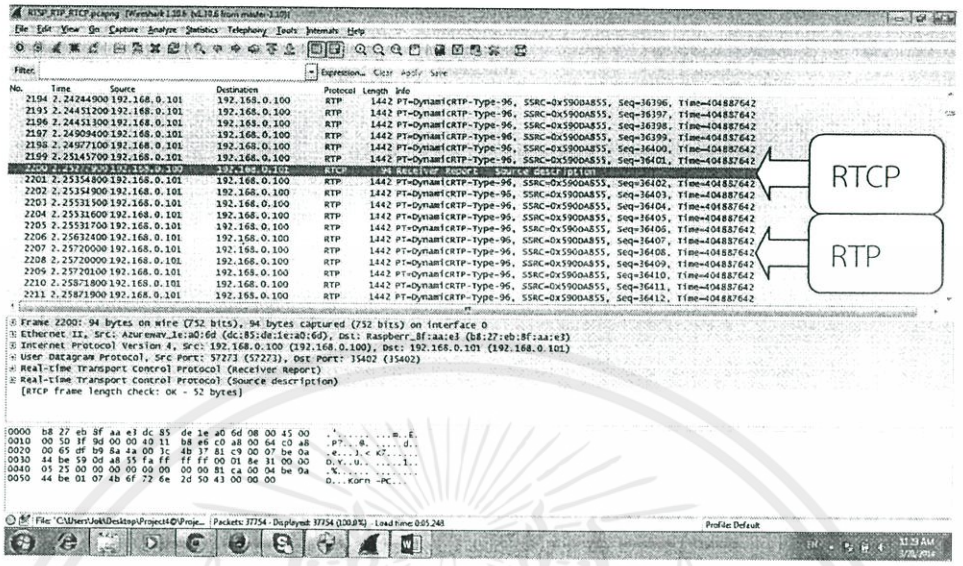
รูปที่ 4.17 ภาพที่ได้จากโปรแกรมวีแอลซี

จากนั้นทำการทำการดักจับแพ็คเก็ตที่ติดต่อกับราสเบอร์รี่ไพโดยการใช้โปรแกรมไวชาร์กผลที่ได้เป็นดังรูปที่ 4.18 และรูปที่ 4.19



รูปที่ 4.18 แพ็กเกต rtsp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แพ็กเกต rtp และ rtcp

เมื่อฝั่นไคลแอนด์ต้องการเรียกดูภาพจากราสเบอร์รี่ไพโดยฝั่นไคลแอนด์มีไอพีเป็น 192.168.0.100 และไอพีฝั่งเซริฟเวอร์เป็น 192.168.0.101 โดยเริ่มแรกฝั่งไคลแอนด์จะส่ง Option ไปยังเซิร์ฟเวอร์ จากนั้น เมื่อมีการร้องขอของฝั่งไคลแอนด์แล้วฝั่งเซิร์ฟเวอร์มีการตอบกลับฝั่งไคลแอนด์จะส่ง Setup เพื่อเป็นการร้องขอให้เปิดพอร์ตที่ใช้ส่งข้อมูลวีดีโอโดยผ่านทางโปรโตคอล rtsp หลังจากนั้นเมื่อกี้การร้องขอและติดต่อกันเสร็จแล้ว ฝั่งเซิร์ฟเวอร์จะส่งข้อมูลวีดีโอไปยังฝั่งไคลแอนด์ผ่านทางโปรโตคอล rtp เป็นดังรูปที่ 4.18 เมื่อเวลาผ่านไประยะหนึ่งจะมีการติดต่อกันของโปรโตคอล rtcp ซึ่งเป็นโปรโตคอลสำหรับการเช็คข้อมูลที่ผิดพลาดต่างๆ ดังรูปที่ 4.19

4.2 การทดลองการทำงานของชุดคำสั่งของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

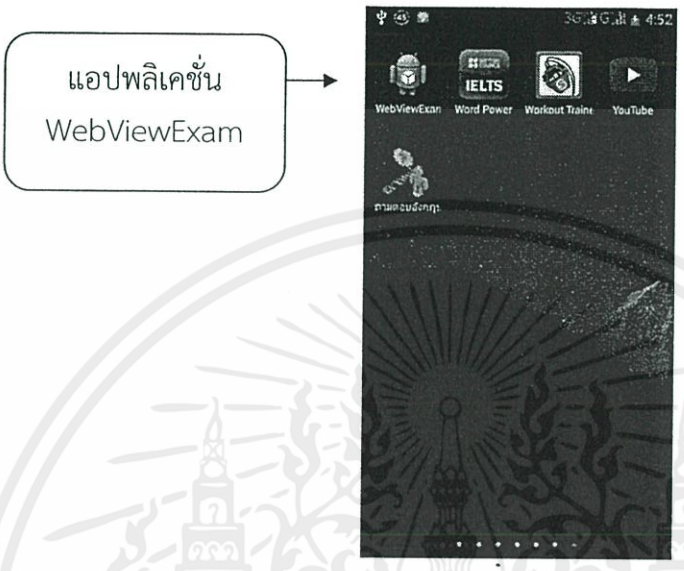
4.2.1 ผลการดูภาพจากกล้องของราสเบอร์รี่ไพผ่านตัวแอพพลิเคชั่นของโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์

ในส่วนของแอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ในการเก็บผลการทดลองเริ่มแรกต้องติดตั้งชุดคำสั่งที่ได้ออกแบบไว้ลงโทรศัพท์เคลื่อนที่บนระบบปฏิบัติการแอนดรอยด์ โดยทำการต่อโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เข้ากับคอมพิวเตอร์ผ่านสายยูเอสบี (Universal Serial Bus : USB)

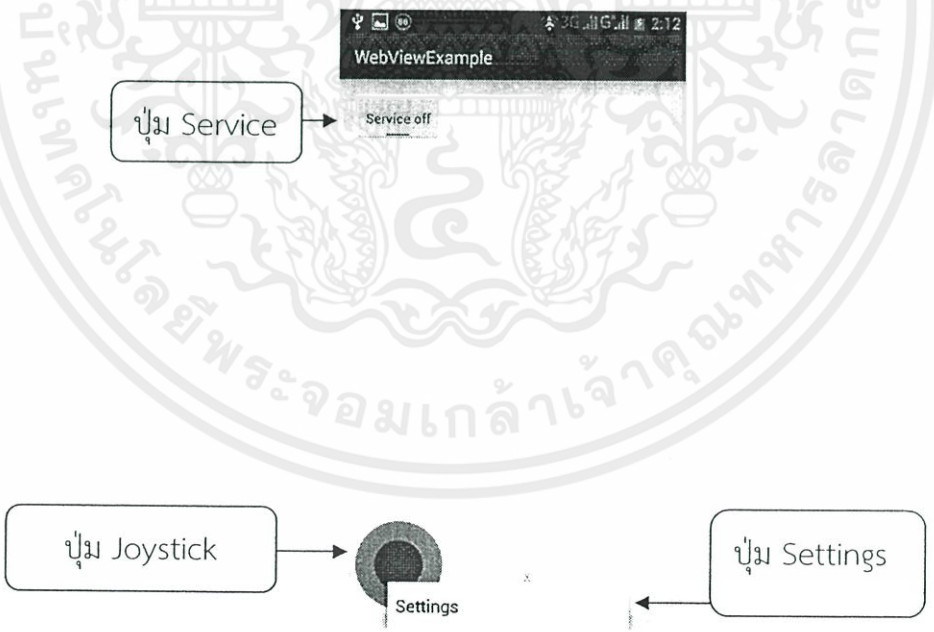
ในส่วนของโปรแกรม eclipse ที่ใช้ในการพัฒนาชุดคำสั่งของแอพพลิเคชั่นเมื่อทำการเชื่อมต่อโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เข้ากับคอมพิวเตอร์แล้ว เมื่อสั่งเริ่มการทำงานของโปรแกรมแล้วเลือกโทรศัพท์เคลื่อนที่ที่ได้ทำการเชื่อมต่อไว้แอพพลิเคชั่นจะทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลงบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์จะได้หน้าไอคอนแอพพลิเคชั่นเป็นรูปที่ 4.20 จะเห็นไอคอนที่ชื่อว่า “ WebViewExam ” ทำการเปิดโปรแกรมได้ผลเป็นรูปที่ 4.21



รูปที่ 4.20 ไอคอนของแอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์



รูปที่ 4.21 หน้าต่างสำหรับผู้ใช้ของแอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

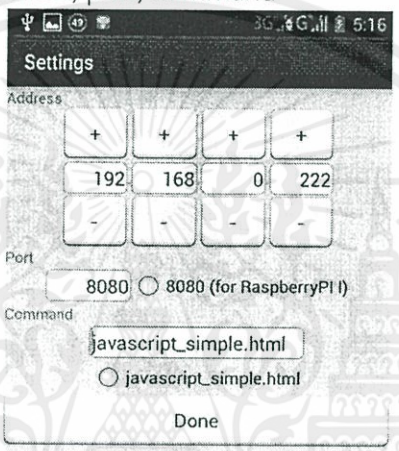
จากรูปที่ 4.21 มีรายละเอียดดังนี้

1) ปุ่ม Service เป็นปุ่มควบคุมการปิด-เปิด การแจ้งเตือนของจากราสเบอร์รี่ไพเมื่อเซ็นเซอร์สามารถตรวจจับการเคลื่อนไหวได้

2) ปุ่ม Joystick เป็นปุ่มควบคุมกล้องหมุนซ้าย-หมุนขวา ขึ้นบน-ลงล่าง

3) ปุ่ม Settings เป็นปุ่มสำหรับเข้าสู่หน้าต่างสำหรับการตั้งค่า

เมื่อเข้าหน้าแอพพลิเคชั่นหลักแล้ว จากนั้นกดที่ปุ่ม Settings จะเข้าสู่หน้าต่างรูปที่ 4.22 เพื่อเข้าไปกำหนด ip address, port, command



รูปที่ 4.22 หน้าต่างสำหรับตั้งค่าแอพพลิเคชั่นของโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

จากรูปที่ 4.22 มีรายละเอียดดังนี้

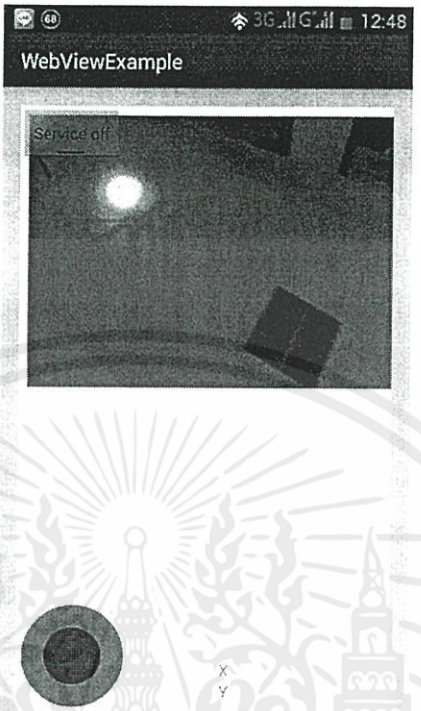
1) Address คือการใส่ค่าไอพีแอดเดสของกล้องในที่นี้จะใช้ 192.168.0.222 เนื่องจากกล้องได้ทำการกำหนดไอพีแอดเดรสเป็นไอพีนี้

2) Port คือ การกำหนดค่าพอร์ต

3) Command กำหนดเป็น javascript_simple.html

เมื่อกำหนดค่าต่างๆเสร็จเรียบร้อยทำการกดปุ่ม Done จะได้ผลดังรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

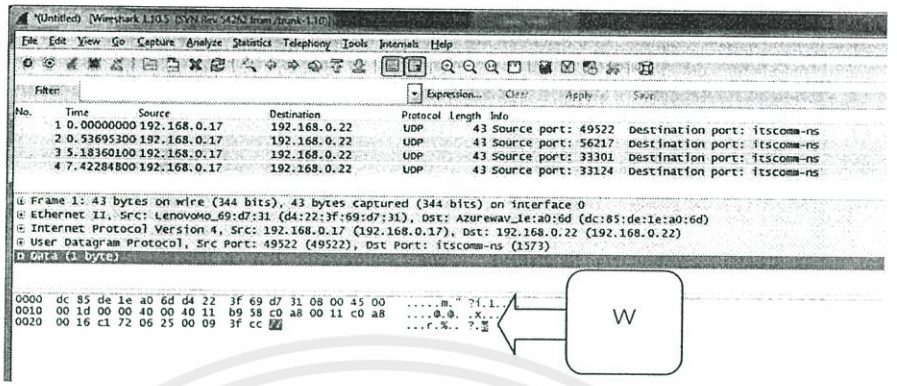


รูปที่ 4.23 หน้าต่างแอพพลิเคชั่นเมื่อทำการกำหนดค่าต่างๆแล้วทำการรัน

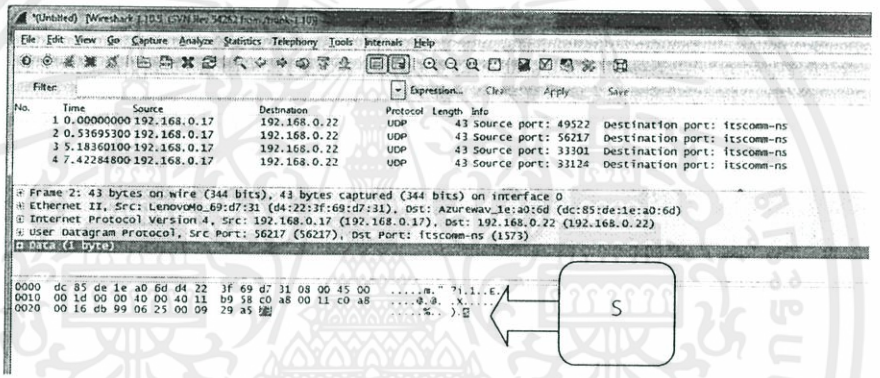
4.2.2 ผลข้อมูลที่ส่งมาจากแอพพลิเคชั่นบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ขณะที่ทำการเลื่อนจอยสติ๊ก

เริ่มจากทำการเขียนโปรแกรมด้วยภาษาไพธอน เพื่อรับค่าข้อมูลที่ส่งมาจากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ขณะที่ทำการเลื่อนจอนสติ๊กไปทาง บน ล่าง ซ้าย ขวา ผลที่ได้จากโปรแกรม Wireshark เป็นดังรูปที่ 4.24 , 4.25 , 4.26 และ 4.27 บนเครื่องคอมพิวเตอร์ที่ใช้รับข้อมูลจากโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

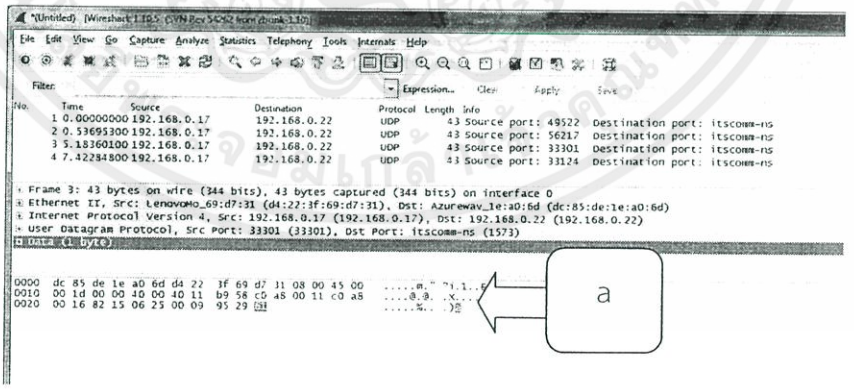
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กขึ้นด้านบน

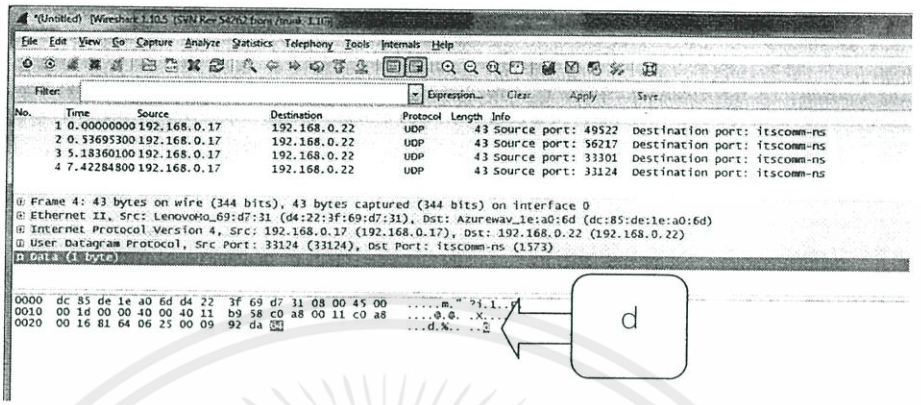


รูปที่ 4.25 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กลงข้างล่าง



รูปที่ 4.26 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กไปด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 แพ็กเกตที่ดักจับได้เมื่อเลื่อนจอยสติ๊กได้ด้านขวา

จากรูปที่ 4.24 ,4.25 ,4.26 และ 4.27 บอกได้ว่าเมื่อเลื่อนจอยสติ๊กขึ้นบนโทรศัพท์เคลื่อนที่จะส่งข้อความ “w” เมื่อเลื่อนจอยสติ๊กลงข้างล่างโทรศัพท์เคลื่อนที่จะส่งข้อความ “s” เมื่อเลื่อนจอยสติ๊กด้านซ้ายโทรศัพท์เคลื่อนที่จะส่งข้อความ “a” และ เมื่อเลื่อนจอยสติ๊กไปด้านขวาโทรศัพท์เคลื่อนที่จะส่งข้อความ “d” ผ่านทางพอร์ต UDP

4.2.3 ผลการแจ้งเตือนโนติฟิเคชั่นเมื่อของแอพพลิเคชั่น

เริ่มจากทำการเชื่อมต่อโทรศัพท์เคลื่อนที่เข้ากับไวเรสราวเตอร์ในวงแลนเดียวกันกับราสเบอร์รี่ เมื่อเซ็นเซอร์ตรวจพบความเคลื่อนไหว ก็จะส่งแจ้งเตือนไปยังโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์จะแจ้งโนติฟิเคชั่นขึ้นมาได้ผลเป็นดังรูปที่ 4.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 ผลของแอพพลิเคชั่นเมื่อได้รับการแจ้งเตือนจากเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากหัวข้อที่ 4.1.1 พบว่าสามารถสั่งให้มอเตอร์หมุนซ้ายขวาได้ โดยการป้อนแรงดันไฟฟ้าไปที่เฟสทั้งสี่เฟสของมอเตอร์ผ่านไอซี ULN2003AN โดยสัญญาณที่ใช้เป็นแบบสองเฟส ในแต่ละไซเคิลของแต่ละเฟส จะมีจำนวนบิตอยู่ 4 บิต โดยแต่ละบิตจะคงสถานะของแต่ละบิตไว้ 3 ms ซึ่งถูกกำหนดไว้ในตัวชุดคำสั่งภาษาไพธอน ผลจากการที่ได้วัดสัญญาณโดยใช้ออสซิลโลสโคป ผลปรากฏว่าสัญญาณที่ใช้ในการควบคุมมอเตอร์เป็นแบบสองเฟส และแต่บิตสามารถวัดช่วงเวลาแต่ละบิตจาก ออสซิลโลสโคปได้ 3.5 ms ซึ่งเป็นไปตามที่ได้เขียนไว้ในชุดคำสั่งในภาษาไพธอนนั่นเอง

จากหัวข้อที่ 4.1.2 พบว่า หลังจากทำการคอมไพล์และรันโปรแกรมเพื่อส่ง UDP Packet ที่มีข้อความ “w”, “s”, “d” และ “a” ไปที่ราสเบอร์รี่ไพ โดยที่ตัวราสเบอร์รี่ไพนั้นกำลังรันโปรแกรมรับค่า UDP Packet แล้วนำข้อความที่ได้ ไปเข้าเงื่อนไขเพื่อสั่งให้มอเตอร์ทั้งสองหมุนไปในทิศทางที่ต้องการ พบว่าสามารถควบคุมมอเตอร์ทั้งสองได้ โดยที่เมื่อรับข้อความเป็น “a” และ “d” จะทำให้มอเตอร์ตัวที่สองหมุนทวนเข็ม และตามเข็มนาฬิกาตาม ลำดับ และเมื่อได้รับข้อความเป็น “w” และ “s” จะทำให้มอเตอร์ตัวที่สองหมุนทวนเข็ม และตามเข็มนาฬิกาตาม ลำดับ อีกทั้งเมื่อทำการวัดสัญญาณที่ใช้ควบคุมมอเตอร์นั้นเป็นแบบสองเฟส ซึ่งเป็นไปตามที่ได้โปรแกรมไว้ในราสเบอร์รี่ไพ รวมทั้งในส่วนของการดักจับแพ็กเกจที่ส่งไปนั้นก็เป็นไปตามที่โปรแกรมไว้ในเครื่องคอมพิวเตอร์ที่ใช้ส่งข้อความไปด้วย UDP Packet

จากหัวข้อที่ 4.1.3 พบว่า เมื่อทำการรันชุดคำสั่ง เพื่อเปิดการทำงานของเซ็นเซอร์ตรวจจับการเคลื่อนไหว พบว่าระดับสัญญาณที่วัดจากขาเอาต์พุทของเซ็นเซอร์ในขณะที่เซนเซอร์ไม่พบการเคลื่อนไหวจะมีค่าแรงดันเป็น 0 โวลต์ แต่เมื่อเซ็นเซอร์ตรวจพบการเคลื่อนไหวจะมีค่าแรงดันเปลี่ยนแปลงโดยเพิ่มจาก 0 โวลต์ เป็น 3.2 โวลต์ และเมื่อเซ็นเซอร์ตรวจไม่พบการเคลื่อนไหวก็จะมีแรงดันเปลี่ยนแปลงกลับมาเป็น 0 โวลต์ อีกครั้ง และกลับมาเป็น 3.2 โวลต์ อีกครั้งเมื่อเซ็นเซอร์ตรวจพบการเคลื่อนไหวในครั้งถัดไป

จากหัวข้อที่ 4.1.5 สามารถรับภาพผ่านเว็บบราวเซอร์ โดยการกำหนด URL:http://192.168.0.222:8080/javascript_simple.html บนเว็บบราวเซอร์ เมื่อทำการดักจับแพ็คเกจโดยใช้ wireshark จะเห็นได้ว่าการรับภาพนั้นจะใช้โปรโตคอล HTTP และภาพที่รับได้ก็เป็นชนิด jpeg

จากหัวข้อที่ 4.2.3 เมื่อทำการเปิดเซอเวอร์บนเครื่องคอมพิวเตอร์ เพื่อดักจับแพ็คเกจที่ส่งมาจากโทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์เมื่อมีการเลื่อนจอยสติ๊ก พบว่า เมื่อทำการเลื่อนจอยสติ๊กไปด้านบน แพ็คเกจที่ได้รับจะเป็นแพ็คเกจที่มีข้อความเป็น “w” หากเลื่อนจอยสติ๊กไปด้านล่าง แพ็คเกจที่ได้รับจะเป็นแพ็คเกจที่มีข้อความเป็น “s” หากเลื่อนจอยสติ๊กไปด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพ็คเกจที่ได้รับจะเป็นแพ็คเกจที่มีข้อความเป็น “a” และเมื่อเลื่อนจอยสติ๊กไปด้านขวา แพ็คเกจที่ได้รับจะเป็นแพ็คเกจที่มีข้อความเป็น “d”

5.2 ข้อเสนอแนะ

1) ต้องทำการตั้งค่าที่ตัวไวเรสเร้าเตอร์ เพื่อกำหนดค่าวงไอพีให้เป็น 192.168.0.x ซับเน็ตมาร์ค 255.255.255.0

2) จำเป็นต้องทำการตั้งค่าราสเบอร์รี่ไพในส่วนของไวเรสแลนทุกครั้ง เมื่อต้องการเปลี่ยน SSID ของไวเรสแลนที่ต้องการเชื่อมต่อ

3) จำเป็นจะต้องทำการเปิด MJPG-Streamer ทุกครั้งเมื่อมีการเปิดราสเบอร์รี่ไพใหม่

4) หากต้องการเชื่อมต่อราสเบอร์รี่ไพ ผ่านอินเทอร์เน็ตจำเป็นจะต้องทำการ Forward Port ที่ตัวเร้าเตอร์ก่อน จึงจะสามารถทำการเชื่อมต่อราสเบอร์รี่ไพได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] เอกชัย มะการ.*เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino.* กรุงเทพมหานคร : บริษัทอีทีที จำกัด,2552.

[2] ดร.วีระศักดิ์ ซึงถาวร .JAVA PROGAMMING Volume 1 . กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัน (มหาชน)

[3] ดร.วีระศักดิ์ ซึงถาวร .JAVA PROGAMMING Volume 2 . กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัน (มหาชน)

[4] ดร.วีระศักดิ์ ซึงถาวร .JAVA PROGAMMING Volume 3 . กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัน (มหาชน)

[5] Mr.Adisak Chinawong "สเตปปิ้งมอเตอร์" http://adisakdiy.com/page22.html.

[6] ATMEL Corporation. AVR Enhanced RISC Microcontrollers Data Book. May 1996.

[7] Michel Margolis . Arduino Cookbook ,Second Edition .2011

[8] ดร.จักรชัย โสอินทร, "Basic Android App Development" พิมพ์ครั้งที่1 นนทบุรี : ไอดีซี, 2554

[9] Wesley, J.Chun . corn PYTHON programming, Second Edition.2006

[10] Reto,Meier. Professional Android 4 Ppplication Developmanet,

[11] Bogdan Ciubotaru , Gabriel Miro Muntean . Advanced Network Pregramming – Principles and Technigues ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

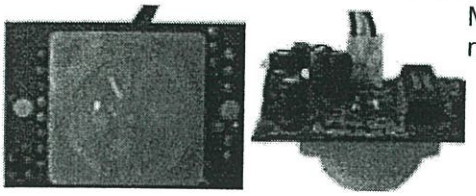
ดาต้าชีส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAX: 86-755-27897072
E-mail:michelle@sbcds.com.cn
Web:www.sbcds.com.cn/en

EAGLE POWER
International Holdings (HK) LTD

PIR Motion Detector Module



Motion detector module uses a motion detector IC and PCB mounted Fresnel lens.

Item: SB0061

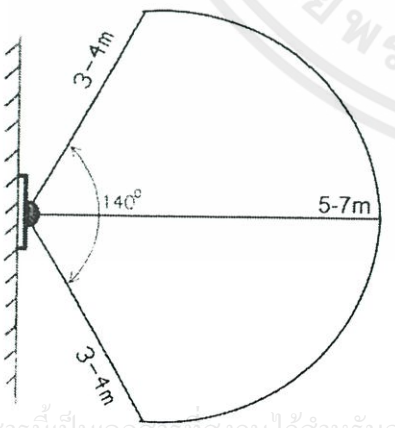
General

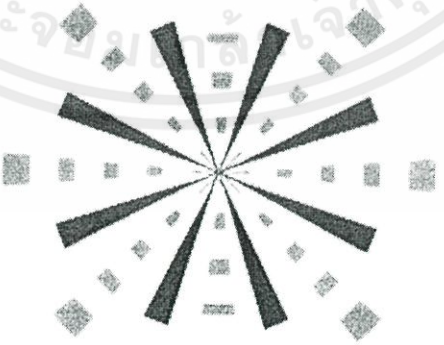
SB0061 is a pyroelectric sensor module which developed for human body detection. A PIR detector combined with a fresnel lens are mounted on a compact size PCB together with an analog IC, SB0061, and limited components to form the module. High level output of variable width is provided.

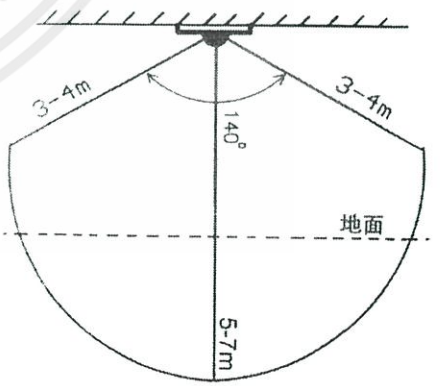
Features and Electrical Specification

Compact size (28 x 38 mm)
Supply current: DC5V-20V(can design DC3V-24V)
Current drain :< 50uA
(Other choice: DC0.8V-4.5V; Current drain: 1.5mA-0.1mA)
Voltage Output: High/Low level signal : 3.3V
(Other choice: Open-Collector Output)
TTL output
High sensitivity
Delay time: 5s-18 minute
Blockade time: 0.5s-50s (acquiescently 0 seconds)
Operation Temperature: -15°C -70Oc
Infrared sensor: dual element, low noise, high sensitivity
Light sensor: CdS photocell (can be add as customer requirement)

Lens information

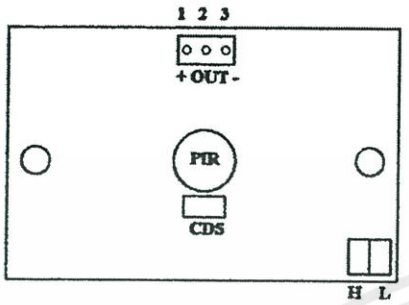






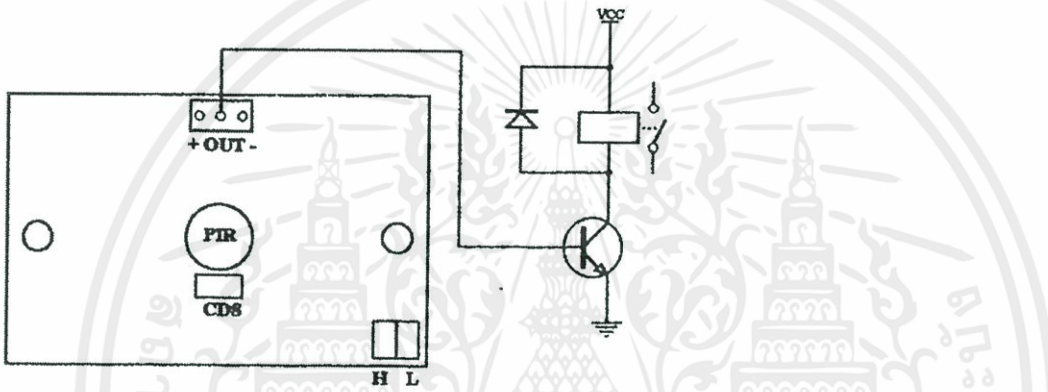
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Note



1. Power anode
2. Output: High level signal
3. Power cathode

H: Can be spring repeatedly
L: Can not be spring repeatedly
CDS: Photocell



Note

Due to the high sensitivity of PIR sensor device, it is not recommended to use the module in the following or similar condition.

A) in rapid environmental changes
B) in strong shock or vibration
C) in a place where there are obstructing material (eg. glass) through which IR cannot pass within detection area.
D) exposed to direct sun light
E) exposed to direct wind from a heater or air condition

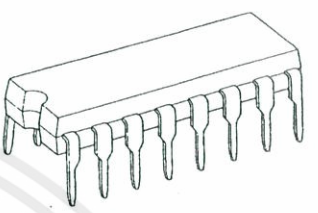
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOSHIBA Bipolar Digital Integrated Circuit Silicon Monolithic

ULN2003APG,ULN2003AFWG
ULN2004APG,ULN2004AFWG

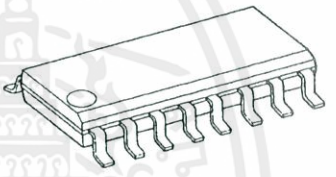
7-ch Darlington Sink Driver

ULN2003APG
ULN2004APG



DIP16-P-300-2.54A

ULN2003AFWG
ULN2004AFWG



SOL16-P-150-1.27A

The ULN2003APG/AFWG Series are high−voltage, high−current darlington drivers comprised of seven NPN darlington pairs.
All units feature integral clamp diodes for switching inductive loads.
Applications include relay, hammer, lamp and display (LED) drivers.

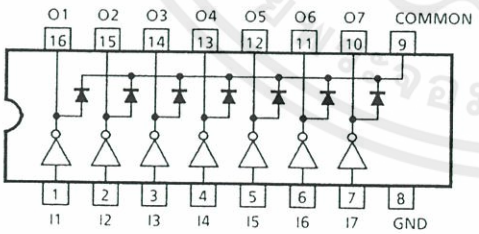
Features

- Output current (single output): 500 mA max
- High sustaining voltage output: 50 V min
- Output clamp diodes
- Inputs compatible with various types of logic
- Package Type-APG: DIP-16pin
- Package Type-AFWG: SOL-16pin

Type	Input Base Resistor	Designation
ULN2003APG/AFWG	2.7 kΩ	TTL, 5 V CMOS
ULN2004APG/AFWG	10.5 kΩ	6 to 15 V PMOS, CMOS

Weight
DIP16−P−300−2.54A : 1.11 g (typ.)
SOL16−P−150−1.27A: 0.15 g (typ.)

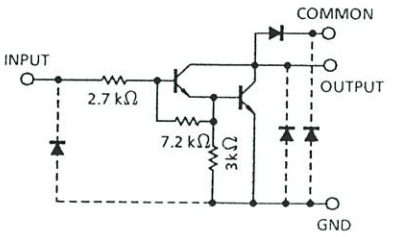
Pin Connection (top view)



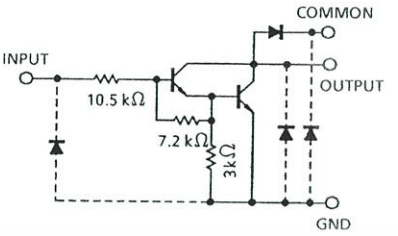
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schematics (each driver)

ULN2003APG/AFWG



ULN2004APG/AFWG



Note: The input and output parasitic diodes cannot be used as clamp diodes.

Absolute Maximum Ratings (Ta = 25°C)

Characteristic		Symbol	Rating	Unit
Output sustaining voltage		$V_{CE\ (SUS)}$	−0.5 to 50	V
Output current		I_{OUT}	500	mA/ch
Input voltage		V_{IN}	−0.5 to 30	V
Clamp diode reverse voltage		V_R	50	V
Clamp diode forward current		I_F	500	mA
Power dissipation	APG	P_D	1.47	W
	AFWG		1.25 (Note)	
Operating temperature		T_{opr}	−40 to 85	°C
Storage temperature		T_{stg}	−55 to 150	°C

Note: On PCB (Test Board: JEDEC 2s2p)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Conditions (Ta = −40 to 85°C)

Characteristic		Symbol	Test Condition		Min	Typ.	Max	Unit
Output sustaining voltage		$V_{CE\ (SUS)}$	—		0	—	50	V
Output current	APG	I_{OUT}	t_{pw} = 25 ms 7 Circuits Ta = 85°C T_j = 120°C	Duty = 10%	0	—	350	mA/ch
				Duty = 50%	0	—	100	
	AFWG			Duty = 10%	0	—	300	
				Duty = 50%	0	—	90	
Input voltage		V_{IN}	—		0	—	24	V
Input voltage (output on)	ULN2003A	$V_{IN\ (ON)}$	I_{OUT} = 400 mA h_{FE} = 800		2.8	—	24	V
	ULN2004A				6.2	—	24	
Input voltage (output off)	ULN2003A	$V_{IN\ (OFF)}$	—		0	—	0.7	V
	ULN2004A		—		0	—	1.0	
Clamp diode reverse voltage		V_R	—		—	—	50	V
Clamp diode forward current		I_F	—		—	—	350	mA
Power dissipation	APG	P_D	Ta = 85°C		—	—	0.76	W
	AFWG		Ta = 85°C	(Note)	—	—	0.65	

Note: On PCB (Test Board: JEDEC 2s2p)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

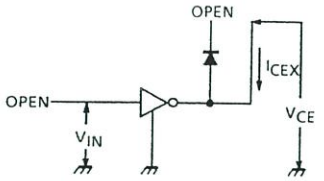
Electrical Characteristics (Ta = 25°C unless otherwise noted)

Characteristic		Symbol	Test Circuit	Test Condition		Min	Typ.	Max	Unit
Output leakage current		I_{CEX}	1	V_{CE} = 50 V, Ta = 25°C		—	—	50	μA
				V_{CE} = 50 V, Ta = 85°C		—	—	100	
Collector−emitter saturation voltage		$V_{CE\ (sat)}$	2	I_{OUT} = 350 mA, I_{IN} = 500 μA		—	1.3	1.6	V
				I_{OUT} = 200 mA, I_{IN} = 350 μA		—	1.1	1.3	
				I_{OUT} = 100 mA, I_{IN} = 250 μA		—	0.9	1.1	
DC Current transfer ratio		h_{FE}	2	V_{CE} = 2 V, I_{OUT} = 350 mA		1000	—	—	—
Input current (output on)	ULN2003A	$I_{IN\ (ON)}$	3	V_{IN} = 2.4 V, I_{OUT} = 350 mA		—	0.4	0.7	mA
	ULN2004A			V_{IN} = 9.5 V, I_{OUT} = 350 mA		—	0.8	1.2	
Input current (output off)		$I_{IN\ (OFF)}$	4	I_{OUT} = 500 μA, Ta = 85°C		50	65	—	μA
Input voltage (output on)	ULN2003A	$V_{IN\ (ON)}$	5	V_{CE} = 2 V, h_{FE} = 800	I_{OUT} = 350 mA	—	—	2.6	V
					I_{OUT} = 200 mA	—	—	2.0	
	ULN2004A				I_{OUT} = 350 mA	—	—	4.7	
					I_{OUT} = 200 mA	—	—	4.4	
Clamp diode reverse current		I_R	6	V_R = 50 V, Ta = 25°C		—	—	50	μA
				V_R = 50 V, Ta = 85°C		—	—	100	
Clamp diode forward voltage		V_F	7	I_F = 350 mA		—	—	2.0	V
Input capacitance		C_{IN}	—	—		—	15	—	pF
Turn−on delay		t_{ON}	8	V_{OUT} = 50 V, R_L = 125 Ω, C_L = 15 pF		—	0.1	—	μs
Turn−off delay		t_{OFF}	8	V_{OUT} = 50 V, R_L = 125 Ω, C_L = 15 pF		—	0.2	—	

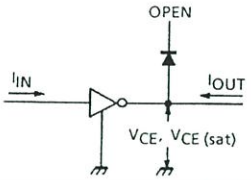
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuit

1. I_{CEX}

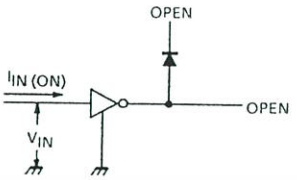


2. $V_{CE\ (sat)}$, h_{FE}

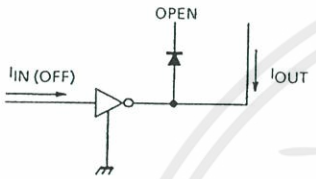


$$h_{FE} = \frac{I_{OUT}}{I_{IN}}$$

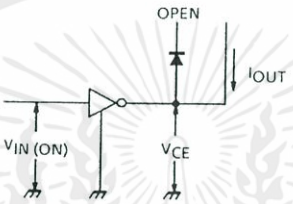
3. $I_{IN\ (ON)}$



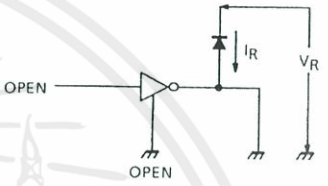
4. $I_{IN\ (OFF)}$



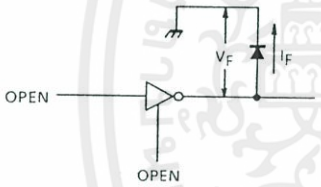
5. $V_{IN\ (ON)}$



6. I_R

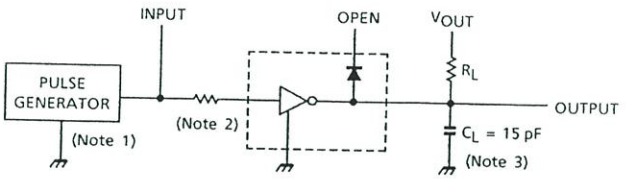


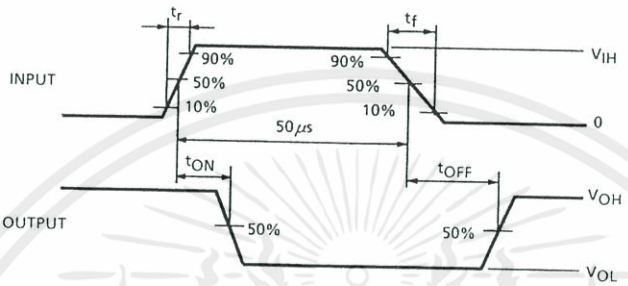
7. V_F



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. t_{ON}, t_{OFF}





Note 1: Pulse width 50 μs, duty cycle 10%
Output impedance 50 Ω, $t_r \leq 5$ ns, $t_f \leq 10$ ns

Note 2: See below

Input Condition

Type Number	R1	V_{IH}
ULN2003A	0	3 V
ULN2004A	0	8 V

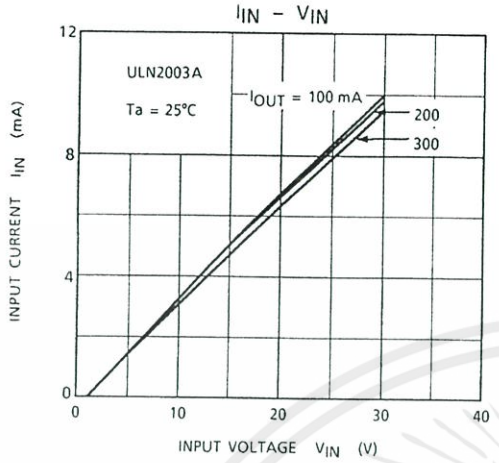
Note 3: C_L includes probe and jig capacitance.

Precautions for Using

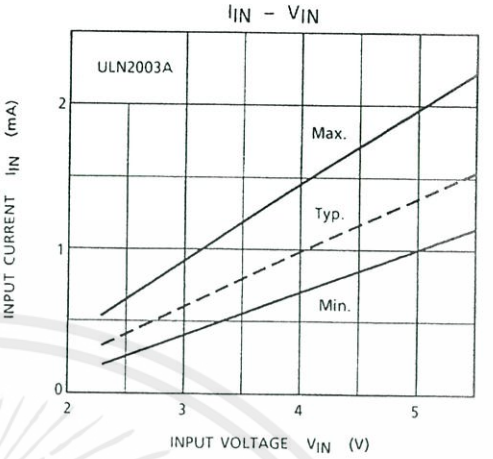
This IC does not include built-in protection circuits for excess current or overvoltage.
If this IC is subjected to excess current or overvoltage, it may be destroyed.
Hence, the utmost care must be taken when systems which incorporate this IC are designed.
Utmost care is necessary in the design of the output line, COMMON and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

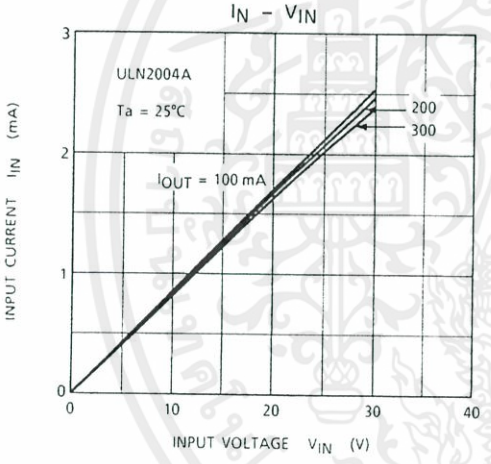
I_{IN} – V_{IN}



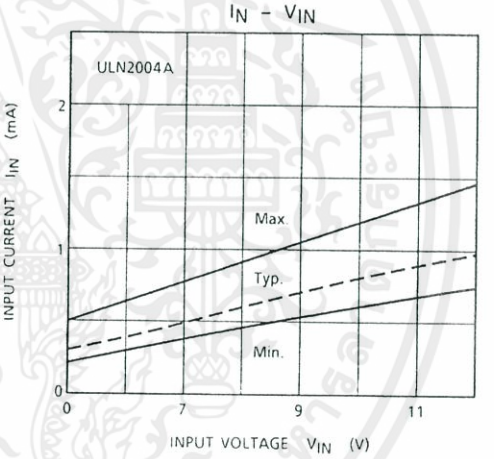
I_{IN} – V_{IN}



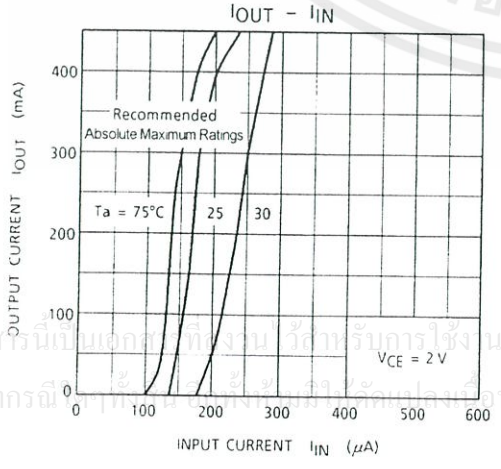
I_N – V_{IN}



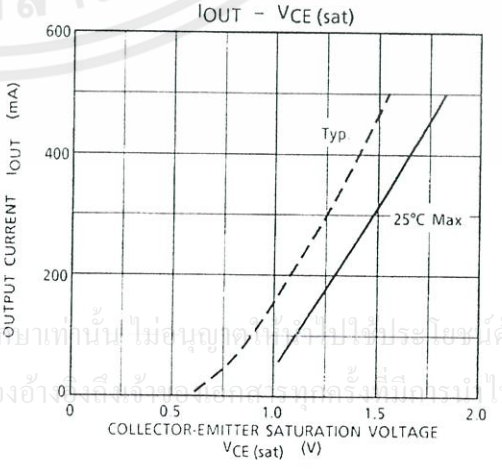
I_N – V_{IN}



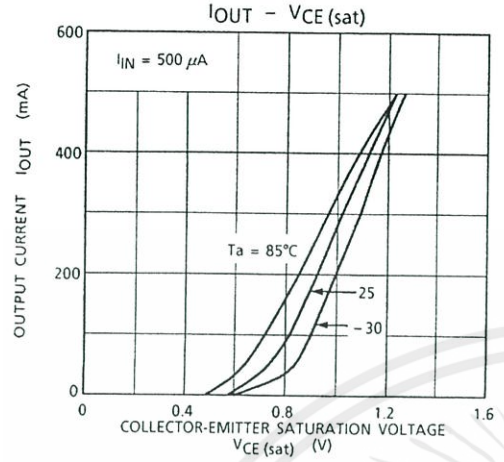
I_{OUT} – I_{IN}

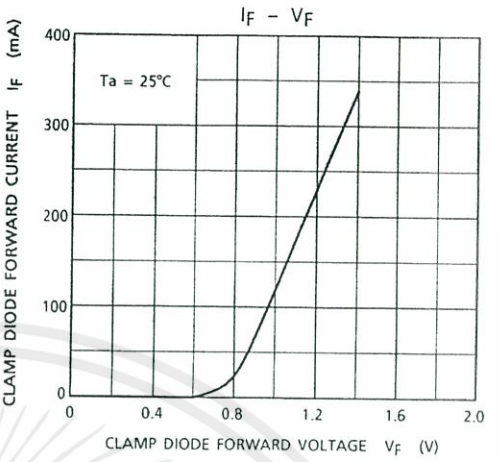


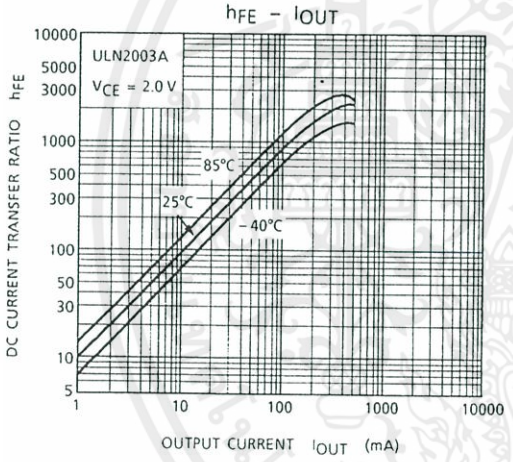
I_{OUT} – $V_{CE\,(sat)}$

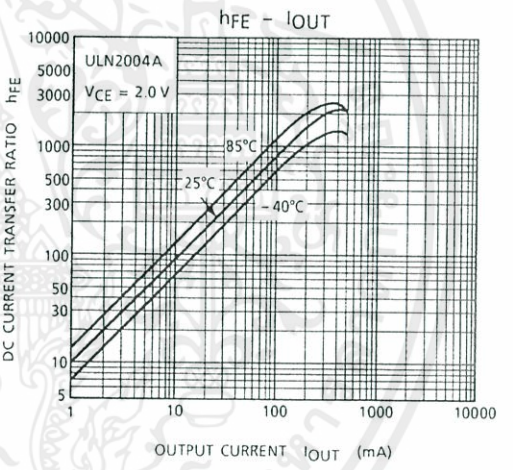


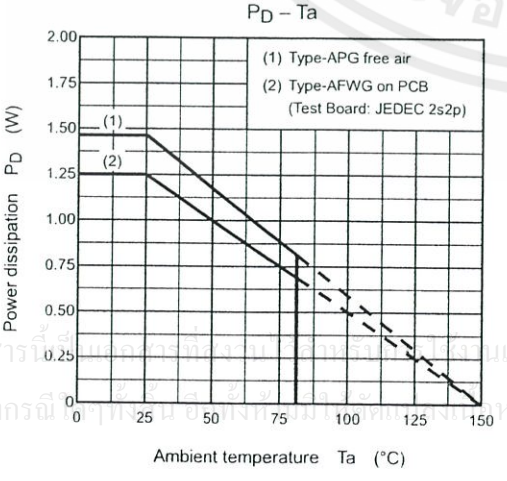
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น พร้อมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้









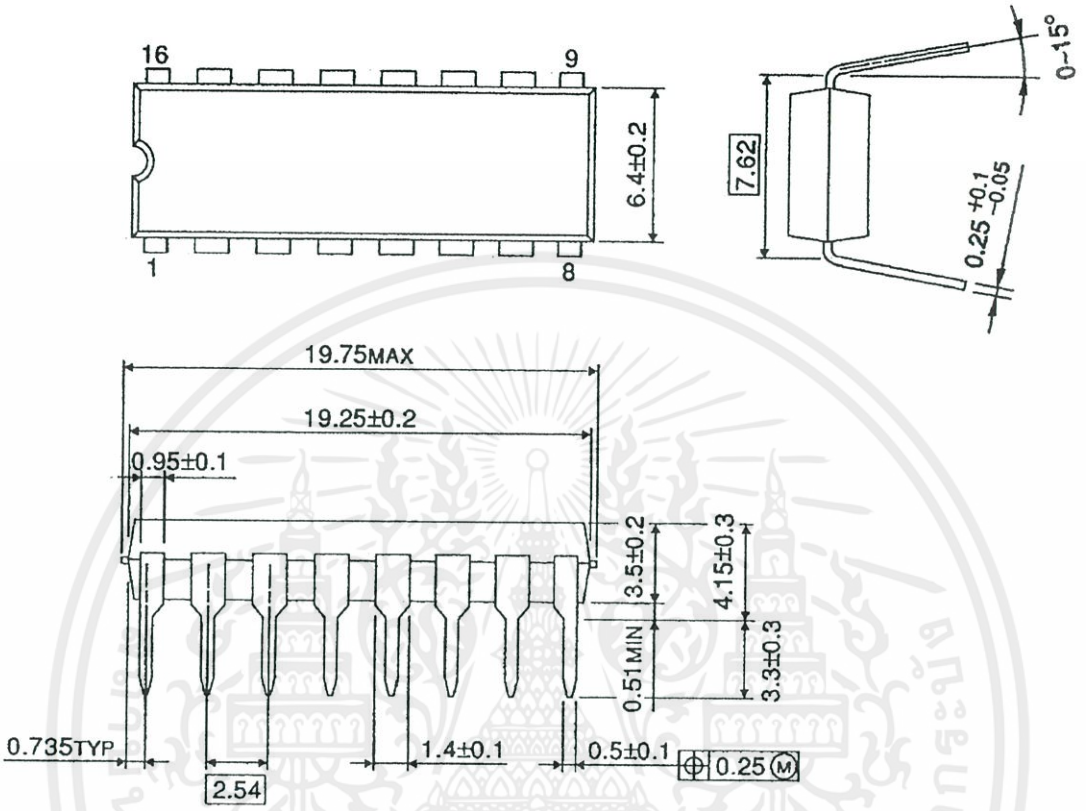


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package Dimensions

DIP16-P-300-2.54A

Unit : mm



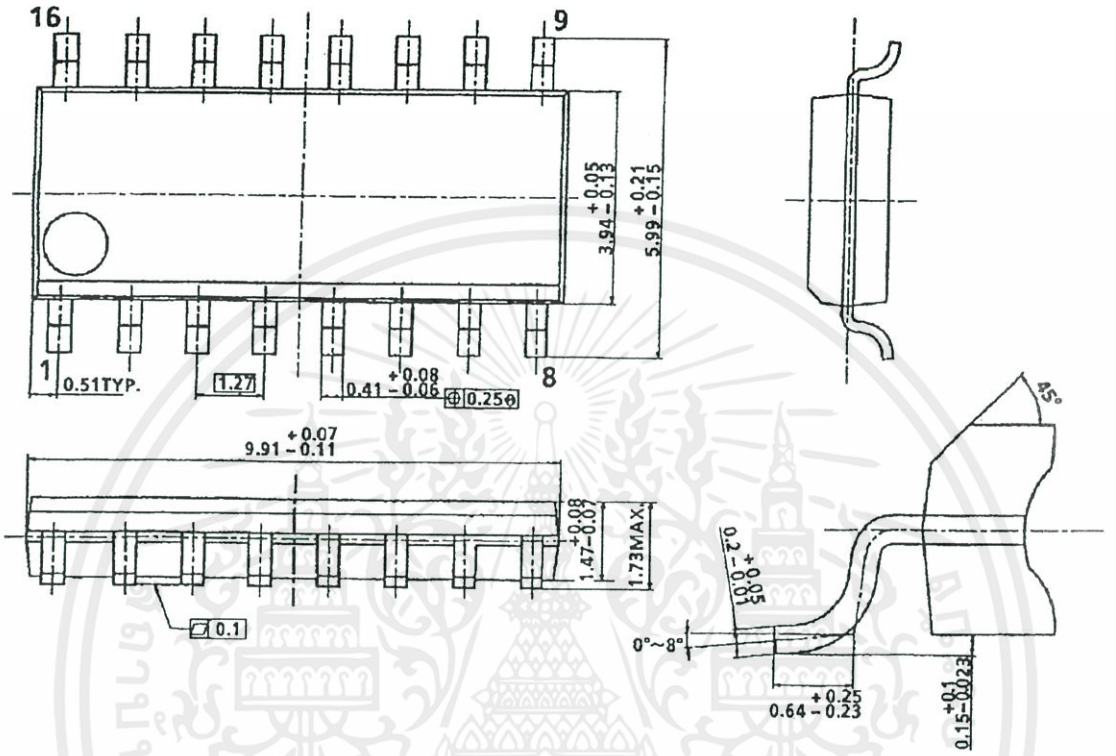
Weight: 1.11 g (typ.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package Dimensions

SOL16-P-150-1.27A

Unit : mm



Weight: 0.15 g (typ.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Notes on Contents

1. Equivalent Circuits

The equivalent circuit diagrams may be simplified or some parts of them may be omitted for explanatory purposes.

2. Test Circuits

Components in the test circuits are used only to obtain and confirm the device characteristics. These components and circuits are not guaranteed to prevent malfunction or failure from occurring in the application equipment.

IC Usage Considerations

Notes on Handling of ICs

(1) The absolute maximum ratings of a semiconductor device are a set of ratings that must not be exceeded, even for a moment. Do not exceed any of these ratings.
Exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion.

(2) Use an appropriate power supply fuse to ensure that a large current does not continuously flow in case of over current and/or IC failure. The IC will fully break down when used under conditions that exceed its absolute maximum ratings, when the wiring is routed improperly or when an abnormal pulse noise occurs from the wiring or load, causing a large current to continuously flow and the breakdown can lead smoke or ignition. To minimize the effects of the flow of a large current in case of breakdown, appropriate settings, such as fuse capacity, fusing time and insertion circuit location, are required.

(3) If your design includes an inductive load such as a motor coil, incorporate a protection circuit into the design to prevent device malfunction or breakdown caused by the current resulting from the inrush current at power ON or the negative current resulting from the back electromotive force at power OFF. IC breakdown may cause injury, smoke or ignition.
Use a stable power supply with ICs with built-in protection functions. If the power supply is unstable, the protection function may not operate, causing IC breakdown. IC breakdown may cause injury, smoke or ignition.

(4) Do not insert devices in the wrong orientation or incorrectly.
Make sure that the positive and negative terminals of power supplies are connected properly. Otherwise, the current or power consumption may exceed the absolute maximum rating, and exceeding the rating(s) may cause the device breakdown, damage or deterioration, and may result injury by explosion or combustion.
In addition, do not use any device that is applied the current with inserting in the wrong orientation or incorrectly even just one time.

(5) Carefully select external components (such as inputs and negative feedback capacitors) and load components (such as speakers), for example, power amp and regulator.
If there is a large amount of leakage current such as input or negative feedback condenser, the IC output DC voltage will increase. If this output voltage is connected to a speaker with low input withstand voltage, overcurrent or IC failure can cause smoke or ignition. (The over current can cause smoke or ignition from the IC itself.) In particular, please pay attention when using a Bridge Tied Load (BTL) connection type IC that inputs output DC voltage to a speaker directly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Points to Remember on Handling of ICs

(1) Heat Radiation Design

In using an IC with large current flow such as power amp, regulator or driver, please design the device so that heat is appropriately radiated, not to exceed the specified junction temperature (Tj) at any time and condition. These ICs generate heat even during normal use. An inadequate IC heat radiation design can lead to decrease in IC life, deterioration of IC characteristics or IC breakdown. In addition, please design the device taking into considerate the effect of IC heat radiation with peripheral components.

(2) Back-EMF

When a motor rotates in the reverse direction, stops or slows down abruptly, a current flow back to the motor's power supply due to the effect of back-EMF. If the current sink capability of the power supply is small, the device's motor power supply and output pins might be exposed to conditions beyond absolute maximum ratings. To avoid this problem, take the effect of back-EMF into consideration in system design.

About solderability, following conditions were confirmed

- Solderability

(1) Use of Sn-37Pb solder Bath
- solder bath temperature = 230°C
- dipping time = 5 seconds
- the number of times = once
- use of R-type flux

(2) Use of Sn-3.0Ag 0.5Cu solder Bath
- solder bath temperature = 245°C
- dipping time = 5 seconds
- the number of times = once
- use of R-type flux

เอกสารนี้จัดทำขึ้นสำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น พร้อมห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact ("Unintended Use"). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Main Activity

```
package com.example.webviewexample;


import java.lang.ref.WeakReference;


import org.videolan.libvlc.EventHandler;
import org.videolan.libvlc.IVideoPlayer;
import org.videolan.libvlc.LibVLC;
import org.videolan.libvlc.Media;
import org.videolan.libvlc.MediaList;


import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.res.Configuration;
import android.graphics.PixelFormat;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.widget.Toast;
import android.widget.ToggleButton;


public class MainActivity extends Activity implements
SurfaceHolder.Callback,IVideoPlayer {


    Intent intent;
    private final String up = "w";
    private final String down = "s";
    private final String left = "a";
    private final String right = "d";


    private JoystickView joystick;
    String URL;
    public static String IP;
    public Detect RunDetect, Right, Left, Up, Down;


    public static final int REQUEST_SETTINGS = 0;


    private int ip_ad1 = 192;
    private int ip_ad2 = 168;
    private int ip_ad3 = 0;
    private int ip_ad4 = 1;
    private int ip_port = 8554;
    //
=============================================
    private SurfaceView mSurface;
    private SurfaceHolder holder;


    // media player
    private LibVLC libvlc;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	private int mVideoWidth;
	private int mVideoHeight;
	private final static int VideoSizeChanged = -1;
	private Intent i;

	public void onCreate(Bundle savedInstanceState)
{
		super.onCreate(savedInstanceState);

	setContentView(R.layout.activity_main);
		i = new Intent(this, MyService.class);

		joystick = (JoystickView)
findViewById(R.id.joystickView);
	joystick.setOnJostickMovedListener(_listener);


	}
	public void onToggleClicked(View view) {
	  // Is the toggle on?
	  boolean on = ((ToggleButton)
view).isChecked();

	  if (on) {
	    startService(i);
	  } else {
		stopService(i);
	  }
	}

	@Override
	protected void onResume() {
		// TODO Auto-generated method stub
		super.onResume();
		SharedPreferences preferences =
getSharedPreferences("SAVED_VALUES",
				MODE_PRIVATE);

		ip_ad1 = preferences.getInt("ip_ad1",
ip_ad1);
		ip_ad2 = preferences.getInt("ip_ad2",
ip_ad2);
		ip_ad3 = preferences.getInt("ip_ad3",
ip_ad3);
		ip_ad4 = preferences.getInt("ip_ad4",
ip_ad4);
		ip_port = preferences.getInt("ip_port",
ip_port);

		//
####################################
		StringBuilder sb = new StringBuilder();
		String s_http = "rtsp://";
		String s_dot = ".";
		String s_colon = ":";
		String s_slash = "/";
		sb.append(s_http);
		sb.append(ip_ad1);
		sb.append(s_dot);
		sb.append(ip_ad2);
		sb.append(s_dot);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
        sb.append(ip_ad3);
        sb.append(s_dot);
        sb.append(ip_ad4);
        sb.append(s_colon);
        sb.append(ip_port);
        sb.append(s_slash);
        URL = new String(sb);

        // ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

        StringBuilder ip = new StringBuilder();
        ip.append(ip_ad1);
        ip.append(s_dot);
        ip.append(ip_ad2);
        ip.append(s_dot);
        ip.append(ip_ad3);
        ip.append(s_dot);
        ip.append(ip_ad4);
        IP = new String(ip);


        Right = new Detect(right);
        Left = new Detect(left);
        Up = new Detect(up);
        Down = new Detect(down);
        mSurface = (SurfaceView)
findViewById(R.id.surface);
        holder = mSurface.getHolder();
        holder.addCallback(this);
                createPlayer(URL);

    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
        releasePlayer();
    }

    private JoystickMovedListener _listener = new
JoystickMovedListener() {
        @Override
        public void OnMoved(int pan, int tilt) {

            if (Math.abs(pan) >
Math.abs(tilt)) {
                if (pan > 7) {

                    RunDetect = new Detect(right);
                    //
Right.start();

                } else if (pan < -7) {

                    RunDetect = new Detect(left);
                    //
Left.start();
                } else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
            } else {
                if (tilt > 7) {
        RunDetect = new Detect(down);
                    //
Down.start();
                } else if (tilt < -7) {
        RunDetect = new Detect(up);
                    //
Up.start();
                } else {
                }
            }
        }

        @Override
        public void OnReleased() {
        }

        public void OnReturnedToCenter() {
        };
    };

    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {
        MenuInflater inflater =
getMenuInflater();
        inflater.inflate(R.layout.option_menu,
menu);
        return true;
    }

    @Override
    public boolean
onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case R.id.settings:
            Intent settings_intent = new
Intent(MainActivity.this,
            SettingsActivity.class);
            settings_intent.putExtra("ip_ad1", ip_ad1);
            settings_intent.putExtra("ip_ad2", ip_ad2);
            settings_intent.putExtra("ip_ad3", ip_ad3);
            settings_intent.putExtra("ip_ad4", ip_ad4);
            settings_intent.putExtra("ip_port", ip_port);
            startActivityForResult(settings_intent,
REQUEST_SETTINGS);
            return true;
        }
        return false;
    }

    public void onActivityResult(int requestCode, int
resultCode, Intent data) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                switch (requestCode) {
                case REQUEST_SETTINGS:
                        if (resultCode ==
Activity.RESULT_OK) {
                                ip_ad1 =
data.getIntExtra("ip_ad1", ip_ad1);
                                ip_ad2 =
data.getIntExtra("ip_ad2", ip_ad2);
                                ip_ad3 =
data.getIntExtra("ip_ad3", ip_ad3);
                                ip_ad4 =
data.getIntExtra("ip_ad4", ip_ad4);
                                ip_port =
data.getIntExtra("ip_port", ip_port);
                                SharedPreferences
preferences = getSharedPreferences(
                "SAVED_VALUES", MODE_PRIVATE);
                SharedPreferences.Editor editor =
preferences.edit();
                editor.putInt("ip_ad1", ip_ad1);
                editor.putInt("ip_ad2", ip_ad2);
                editor.putInt("ip_ad3", ip_ad3);
                editor.putInt("ip_ad4", ip_ad4);
                editor.putInt("ip_port", ip_port);
                                editor.commit();
                        }
                        break;
                }
        }

        /*************
         * Surface
         *************/

        public void surfaceCreated(SurfaceHolder
holder) {
        }

        public void surfaceChanged(SurfaceHolder
surfaceholder, int format,
                        int width, int height) {
                if (libvlc != null)
                libvlc.attachSurface(holder.getSurface(), this);
        }

        public void surfaceDestroyed(SurfaceHolder
surfaceholder) {
        }

        private void setSize(int width, int height) {
                mVideoWidth = width;
                mVideoHeight = height;
                if (mVideoWidth * mVideoHeight <= 1)
                        return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                // get screen size
                int w =
getWindow().getDecorView().getWidth();
                int h =
getWindow().getDecorView().getHeight();

                // getWindow().getDecorView()
doesn't always take orientation into
                // account, we have to correct the
values
                boolean isPortrait =
getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_PORTRAIT;
                if (w > h && isPortrait || w < h &&
!isPortrait) {
                        int i = w;
                        w = h;
                        h = i;
                }
                float videoAR = (float) mVideoWidth /
(float) mVideoHeight;
                float screenAR = (float) w / (float) h;

                if (screenAR < videoAR)
                        h = (int) (w / videoAR);
                else
                        w = (int) (h * videoAR);

                // force surface buffer size
                holder.setFixedSize(mVideoWidth,
mVideoHeight);
                // set display size
                LayoutParams lp =
mSurface.getLayoutParams();
                lp.width = w;
                lp.height = h;
                mSurface.setLayoutParams(lp);
                mSurface.invalidate();
        }

        @Override
        public void setSurfaceSize(int width, int height,
int visible_width,
                        int visible_height, int
sar_num, int sar_den) {
                Message msg =
Message.obtain(mHandler, VideoSizeChanged, width,
height);
                msg.sendToTarget();
        }

        /*************
        * Player
        *************/
        private void createPlayer(String media) {
                releasePlayer();
                try {
                        if (media.length() > 0) {
                                Toast toast =
Toast.makeText(this, media, Toast.LENGTH_LONG);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            toast.setGravity(Gravity.BOTTOM |
Gravity.CENTER_HORIZONTAL, 0,

            0);
                        toast.show();
                }

                // Create a new media player
                libvlc = LibVLC.getInstance();
        libvlc.setHardwareAcceleration(LibVLC.HW_ACC
ELERATION_DISABLED);
        libvlc.setTimeStretching(true);
                libvlc.frameSkipEnabled();
        libvlc.setNetworkCaching(230);
                libvlc.setFrameSkip(true);

                libvlc.setChroma("RV32");
                libvlc.setVerboseMode(true);
                LibVLC.restart(this);

        EventHandler.getInstance().addHandler(mHandl
er);

        holder.setFormat(PixelFormat.RGBX_8888);
        holder.setKeepScreenOn(true);
                MediaList list =
libvlc.getMediaList();
                list.clear();
                list.add(
                        new
Media(libvlc, LibVLC.nativeToURI(media)), false);

                libvlc.playIndex(0);
            } catch (Exception e) {
                Toast.makeText(this, "Error
creating player!", Toast.LENGTH_LONG)
                        .show();
            }
        }

    private void releasePlayer() {
        if (libvlc == null)
            return;

        EventHandler.getInstance().removeHandler(mHa
ndler);

        libvlc.stop();
        libvlc.detachSurface();
        holder = null;
        libvlc.closeAout();
        libvlc.destroy();
        libvlc = null;

        mVideoWidth = 0;
        mVideoHeight = 0;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /*************
         * Events
         *************/

        private Handler mHandler = new
MyHandler(this);

        private static class MyHandler extends Handler {
            private WeakReference<MainActivity>
mOwner;

            public MyHandler(MainActivity owner)
{
                mOwner = new
WeakReference<MainActivity>(owner);
            }

            @Override
            public void handleMessage(Message
msg) {
                MainActivity player =
mOwner.get();

                // SamplePlayer events
                if (msg.what ==
VideoSizeChanged) {

        player.setSize(msg.arg1, msg.arg2);
                    return;
                }
                // Libvlc events
                Bundle b = msg.getData();
                switch (b.getInt("event")) {
                case
EventHandler.MediaPlayerEndReached:

                    player.releasePlayer();
                    break;
                case
EventHandler.MediaPlayerPlaying:
                case
EventHandler.MediaPlayerPaused:
                case
EventHandler.MediaPlayerStopped:
                default:
                    break;
                }
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

My Service

```
package com.example.webviewexample;


import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;


import android.annotation.TargetApi;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.IBinder;
import android.util.Log;
import android.widget.Toast;


@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
public class MyService extends Service {
        String sentence;
        static Boolean isLooping = false;
        int port = 10597;


        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        DatagramSocket serverSocket;
        String IPSource;


        public MyService() {
                try {
                        serverSocket = new
DatagramSocket(port);
                } catch (IOException e) {
                        e.printStackTrace();
                }
                //thread.start();
        }
        @Override
        public IBinder onBind(Intent arg0) {
                return null;
        }

        Thread thread = new Thread() {

        @SuppressWarnings("deprecation")
                @Override
                public void run() {
                        try {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    while (isLooping)                                   }
{

        DatagramPacket receivePacket = new
DatagramPacket(                                                 } catch
                                                    (InterruptedException e) {
        receiveData, receiveData.length);
                                                        e.printStackTrace();
                                                                Thread.yield();
                                    try {
                                                                thread.stop();
        serverSocket.receive(receivePacket);                        }
                                    } catch                     }
(IOException e) {                                               };
        // TODO Auto-generated catch block

        e.printStackTrace();                                    @Override
                                    }
                                                    public int onStartCommand(Intent intent,int
                                                    flags, int startId) {
        sentence = new
String(receivePacket.getData());                                    super.onStartCommand(intent,
                                                    flags, startId);
                                    IPSource
= serverSocket.getLocalAddress().getHostAddress();                  isLooping = true;

                                                                    thread.start();

                                    if                              Toast.makeText(getApplication(),
(receivePacket != null) {                           "Start Service", Toast.LENGTH_SHORT).show();

                                                                    Log.d("TAG", "Service is started");
        createNotification();
                                                                    return START_STICKY;
                                    }

                                                    }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
        sleep(200);                                             @Override
```

```
public void onDestroy() {
        super.onDestroy();

        isLooping = false;
        thread.stop();
        Toast.makeText(getApplication(),
"Stop Service", Toast.LENGTH_SHORT)
                .show();
        Log.d("TAG", "Service is stoped");

    }

    @TargetApi(Build.VERSION_CODES.JELLY_B
EAN) public void createNotification() {

        Intent intent = new Intent(this,
MainActivity.class);
        PendingIntent pIntent =
PendingIntent.getActivity(this, 0, intent, 0);

        Uri soundUri =
RingtoneManager.getDefaultUri(RingtoneManager.T
YPE_NOTIFICATION);

        Notification noti = new
Notification.Builder(getApplicationContext())

    .setContentTitle("Motion Sensor Detected
")

    .setContentText("Subject").setSmallIcon(R.d
rawable.ic_launcher)
                .setContentIntent(pIntent)

                .setSound(soundUri)

                .addAction(R.drawable.ic_launcher, "Call",
pIntent).build();
                NotificationManager
notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                // hide the notification after its
selected
                noti.flags |=
Notification.FLAG_AUTO_CANCEL;

                notificationManager.notify(0,
noti);
            }

}
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้